เครื่องกัดแนวดิ่งซีเอ็นซีบนพื้นฐานไมโครคอมพิวเตอร์

CNC VERTICAL MILLING MACHINE BASED ON MICROCOMPUTER

โดย

นาย ทรงพล สิริรวมทรัพย์ 36014155

นาย รังสรรค์ เสาวพุทธสุเวช 36014346

อาจารย์ที่ปรึกษา

อ.กวิน สนธิเพิ่มพูน

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
สาขาวิชาวิศวกรรมเครื่องกล
สถาบันเทคโนโลยีพระจอมเกล้า เจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2539

ปีการศึกษา 2539

เครื่องกัดแนวดิ่งซีเอ็นซีบนพื้นฐานไมโครคอมพิวเตอร์

โดย

นายทรงพล สิริรวมทรัพย์ 36014155

นายรังสรรค์ เสาวพุทธสุเวช 36014346

อาจารย์ที่ปรึกษา

(อ.กวิน สนธิเพิ่มพูน)

เครื่องกัดแนวดิ่งซีเอ็นซีบนพื้นฐานไมโครคอมพิวเตอร์

(CNC VERTICAL MILLING MACHINE BASED ON MICROCOMPUTER)

บทคัดย่อ

ปริญญานิพนธ์ฉบับนี้ได้ศึกษาถึงการพัฒนาโปรแกรมที่ใช้ในการควบคุมเครื่องกัดแนวดิ่งซีเอ็นซีบนพื้นฐานไมโครคอมพิวเตอร์ (CNC VERTICAL MILLING MACHINE BASED ON MICROCOMPUTER) โดยอาศัยการควบคุมผ่านทางการ์ดควบคุมสำเร็จรูป (PC SERVO-MOTOR CONTROLLER MODEL 5650) ซึ่งสามารถทำงานและสั่งงานบนเครื่องไมโครคอมพิวเตอร์โดยตรงได้ โดยได้ทำการพัฒนาเพิ่มเติมในส่วนของฟังก์ชันที่ยังไม่สมบูรณ์ รวมทั้งการเขียนโปรแกรมเพิ่มในส่วนการอ่านค่าคำสั่ง G-Code และ M-Code ในส่วนที่เหลือ จากผลงานที่ทำจะได้ต้นแบบของเครื่องกัดแนวดิ่งซีเอ็นซี ที่มีราคาไม่แพง ใช้ต้นทุนในการผลิตต่ำ แต่สามารถใช้งานได้อย่างมีประสิทธิภาพทัดเทียมกับเครื่องกัดที่สั่งเข้ามาจากต่างประเทศ และสามารถพัฒนาโปรแกรมใช้งานให้อยู่ในรูปแบบที่เหมาะสมกับงานเฉพาะทางได้ อีกทั้งยังนำไปประยุกต์ใช้งานในเชิงอุตสาหกรรมได้อย่างกว้างขวางโดยเฉพาะอย่างยิ่งในอุตสาหกรรมการผลิตชิ้นส่วนรถยนต์และแม่พิมพ์ เป็นต้น

ABSTRACT

This thesis is studying to develope program which applied for CNC vertical milling machine based on microcomputer , by using the Servo-motor controlling card model 5650 which can function and be operated directly via microcomputer. This program is developed in the uncomplete function and program in function G-Code&M-Code reading. From the result of the tedious work, we acquired the prototype of the CNC vertical milling machine, which is inexpensive, low operating cost, and perform machining job efficiently as well as other import CNC vertical milling machines from abroad. We also have developed the compact multi-purpose program for all highly-specific application. It can be used in wide range industry fields ,for example: car assembly manufacturing and die casting and so on.

สารบัญ

บทที่ 1

บทนำ

ในสภาวะที่เศรษฐกิจเจริญเติบโตขึ้นมาเรื่อยๆ และจำนวนประชากรที่เพิ่มมากขึ้น ความต้องการทางด้านปัจจัย 4 ก็มีเพิ่มมากขึ้นตามลำดับ การแข่งขันทางการค้าก็ยิ่งทวีสูงขึ้นเรื่อยๆ เหตุต่างๆ เหล่านี้ ทำให้มนุษย์มีความจำเป็นที่จะต้องคิดค้นและพัฒนาการผลิตให้รวดเร็วและประหยัด เพื่อตอบสนองต่อความต้องการที่เพิ่มมากขึ้น เครื่องจักรกลอัตโนมัติได้ถูกออกแบบและพัฒนาสร้างขึ้นมาให้สามารถทำงานซ้ำๆ กันได้ทุกเวลาที่ต้องการ ซึ่งระบบการทำงานอัตโนมัติเป็นที่รู้จักกันอย่างแพร่หลาย เช่น เครื่องเล่นเปียโนอัตโนมัติซึ่งทำงานโดยระบบแมคคานิคควบคุม เครื่องกลึงอัตโนมัติที่ควบคุมการทำงานด้วยลูกเบี้ยว แต่เครื่องจักรเหล่านี้มีข้อเสียตรงที่ การเปลี่ยนผลิตภัณฑ์หรือชิ้นงานใหม่ต้องใช้เวลามาก และการเปลี่ยนลักษณะงานมีขีดจำกัด

ในปี ค.ศ. 1948 นักวิทยาศาสตร์ในสถาบัน MIT (Massachusetts Institute of Technology) ได้ริเริ่มทำโครงการพัฒนาเครื่องจักรกลที่ควบคุมด้วยระบบคอมพิวเตอร์ขึ้น โดยได้รับการสนับสนุนโครงการจากกองทัพอากาศของสหรัฐอเมริกา (U.S. Air Force)

เครื่องจักรระบบเอ็นซีเครื่องแรก คือ CINCINNATIC HYDROTEL VERTICAL-SPINDLE MACHINE และนำออกใช้งานในปี ค.ศ. 1957

สำหรับในประเทศไทยนั้น เริ่มต้นโดยการนำเข้าเครื่องเก่าจากต่างประเทศ ซึ่งมีราคาค่อนข้างต่ำเพื่อการเริ่มต้นศึกษาเรียนรู้ถึงความสามารถ เครื่องส่วนใหญ่ที่นำเข้าจะเป็นเครื่องที่ใช้ NC ในการควบคุม จึงมีขนาดใหญ่ในส่วนของคอมพิวเตอร์ในการควบคุม และในเวลาต่อมาก็เริ่มนำเข้าเครื่องจักรที่ควบคุมด้วย CNC จนถึงปัจจุบัน ในเวลาปัจจุบันเครื่องที่นำเข้าสมัยก่อนที่ควบคุมด้วย NC ก็เริ่มประสบปัญหาคือ ใช้งานไม่ได้ในส่วนของคอมพิวเตอร์หรือส่วนของอิเล็กทรอนิคส์ในการควบคุมเครื่องจักร ดังนั้นจึงมีอาจารย์และนักวิจัยหลายกลุ่มหลายสถาบันเริ่มให้ความสนใจในการวิจัยและพัฒนาส่วนควบคุมเครื่องจักรดังกล่าว

บทที่ 2
ทฤษฎีเบื้องต้นเกี่ยวกับเตรื่องจักรกลซีเอ็นซี

2.1 พัฒนาการของเครื่องจักรกลเอ็นซีและซีเอ็นซี

2.1.1 ความหมายของเอ็นซีและซีเอ็นซี

เอ็นซี (NC) ย่อมาจากคำว่า Numerical Control หมายถึง การควบคุมเครื่องจักรกลด้วยระบบตัวเลขและตัวอักษร ซึ่งคำจำกัดความนี้ได้จากประเทศสหรัฐอเมริกา กล่าวคือ การเคลื่อนที่ต่างๆ ตลอดจนการทำงานอื่นๆ ของเครื่องจักรกล จะถูกควบคุมโดยรหัสคำสั่งที่ประกอบด้วยตัวเลข ตัวอักษร และสัญลักษณ์อื่นๆ ซึ่งจะถูกแปลงเป็นคลื่นสัญญาณ (pulse)ของกระแสไฟฟ้าหรือสัญญาณออกอื่นๆ ที่จะไปกระตุ้นมอเตอร์หรืออุปกรณ์อื่นๆ เพื่อทำให้เครื่องจักรกลทำงานตามขั้นตอนที่ต้องการ

ซีเอ็นซี (CNC) ย่อมาจากคำว่า Computerized Numerical Control ระบบควบคุมเอ็นซีแบบนี้จะมีคอมพิวเตอร์ที่มีความสามารถสูงเพิ่มเข้าไปภายในระบบ ทำให้สามารถจัดการกับข้อมูลที่ป้อนเข้าไปในระบบเอ็นซี และประมวลผลข้อมูลเพื่อนำผลลัพธ์ที่ได้ไปควบคุมการทำงานของเครื่องจักรกล

2.1.2 ความแตกต่างระหว่างเครื่องจักรกลเอ็นซีกับเครื่องจักรกลทั่วไป

ความแตกต่างในการใช้เครื่องจักรกลเอ็นซี เมื่อเปรียบเทียบกับเครื่องจักรกลที่ใช้ทั่วไปก็คือ การตัดสินใจในการกำหนดขั้นตอนการทำงานต่างๆ จะกระทำเพียงครั้งเดียว กล่าวคือ จะกระทำในขั้นตอนการวางแผนและสร้างโปรแกรมสำหรับควบคุมเครื่องจักรกลเท่านั้น ต่อจากนั้น โปรแกรมก็จะถูกนำไปใช้ในการควบคุมการทำงานของเครื่องจักรกล สำหรับการผลิตชิ้นงานที่ต้องการ โดยสามารถทำการผลิตซ้ำๆกันกี่ครั้งก็ได้ตามต้องการ

			เครื่องจักรกลทั่วไป	เครื่องจักรกลเอ็นซี
1	การป้อนโปรแกรม		ไม่มี	มี
2	การจับยึดชิ้นงาน	ขั้น	มือ	มือ
3	การจับยึดเครื่องมือตัด	เ ต ร ี ย ม	มือ	มือ หรือชุดควบคุม
4	การตั้งจุดอ้างอิง	งาน	มือ	มือ
5	การตั้งความเร็วรอบ		มือ	ระบบควบคุม
6	การเลื่อนแท่นเลื่อน	ขั้น	มือหมุน	ระบบควบคุม

7	การเปรียบเทียบระยะ	ตัด	สายตา	ระบบควบคุม
8	การตรวจสอบขนาด	เฉือน	เครื่องมือวัด	ใช้เวลาน้อยกว่า

ตาราง 2.1 ตารางเปรียบเทียบการทำงานระหว่างเครื่องจักรกลทั่วไปกับเครื่องจักรกลเอ็นซี

2.1.3 ความแตกต่างระหว่างระบบเอ็นซีกับระบบซีเอ็นซี

ระบบซีเอ็นซีเป็นระบบที่พัฒนาต่อเนื่องมาจากระบบเอ็นซี ดังนั้น ความแตกต่างระหว่างระบบเอ็นซีกับระบบซีเอ็นซี ก็จะอยู่ที่ความสามารถของระบบควบคุม นั่นคือคอมพิวเตอร์ เมื่อนำระบบซีเอ็นซีไปควบคุมเครื่องจักรกล ความสามารถในการทำงานต่างๆ จะเพิ่มมากขึ้นเมื่อเปรียบเทียบกับเครื่องจักรกลเอ็นซีดังนี้

1. การแสดงภาพจำลอง (Simulation) การทำงานตามโปรแกรมที่ป้อนเข้าในระบบทางจอภาพ
2. ความจุของหน่วยความจำเพิ่มมากขึ้น สามารถเก็บข้อมูลโปรแกรมได้มาก
3. การแก้ไขและลบโปรแกรมสามารถกระทำได้ที่เครื่องจักรโดยตรง
4. สามารถส่งข้อมูลไปเก็บไว้ในหน่วยความจำภายนอกได้
5. ระบบความปลอดภัยเพิ่มมากขึ้น
6. มีการชดเชยความผิดพลาดที่เกิดจากการวัดและการส่งกำลัง
7. มีโปรแกรมสำเร็จสำหรับการคำนวณค่าต่างๆ เช่น ความเร็วรอบ อัตราป้อน เป็นต้น

2.1.4 ข้อดีและข้อเสียของเครื่องจักรกลเอ็นซีและซีเอ็นซี

เครื่องจักรกลเอ็นซีและซีเอ็นซี เป็นเครื่องจักรกลสมัยใหม่ที่มีประสิทธิภาพการทำงานสูง แต่ในขณะเดียวกันราคาก็สูงตามด้วย ดังนั้น ก่อนที่จะพิจารณาจัดซื้อเครื่องจักรกลประเภทนี้มาใช้ในกระบวนการผลิต จำเป็นที่จะต้องศึกษารายละเอียดต่างๆ เกี่ยวกับขีดความสามารถของเครื่อง ตลอดจนข้อดีและข้อเสียของเครื่องจักรกลประเภทนี้ก่อน

ข้อดีของเครื่องจักรกลเอ็นซีและซีเอ็นซี เมื่อเปรียบเทียบกับเครื่องจักรกลอัตโนมัติประเภทอื่นๆ พอจะสรุปได้ดังนี้

1. มีความยืดหยุ่นในการทำงานสูง การเปลี่ยนงานใหม่จะแก้ไขหรือเปลี่ยนแปลงเฉพาะโปรแกรมเท่านั้น
2. ความเที่ยงตรง (Accuracy) จะอยู่ระดับเดียวกันตลอดช่วงความเร็วรอบและอัตราป้อนที่ใช้ทำการผลิต
3. ใช้เวลาในการผลิต (Production Time) สั้นกว่า
4. สามารถใช้ผลิตชิ้นงานที่มีรูปทรงซับซ้อนได้ง่าย

5. การปรับตั้งเครื่องจักรกระทำได้ง่าย ใช้เวลาน้อยกว่าการผลิตด้วยวิธีอื่น ๆ

6. หลีกเลี่ยงความจำเป็นที่ต้องใช้ช่างควบคุมที่มีทักษะและประสบการณ์สูง

7. ช่างควบคุมเครื่องมีเวลาว่างจากการควบคุมเครื่อง สามารถที่จะจัดเตรียมงานอื่นๆไว้ล่วงหน้าได้

8. การตรวจสอบคุณภาพไม่จำเป็นต้องกระทำทุกขั้นตอนและทุกชิ้น

ส่วน**ข้อเสีย**ของเครื่องจักรกลเอ็นซีและซีเอ็นซีมีดังนี้

1. ราคาของเครื่องจักรค่อนข้างสูง

2. การบำรุงรักษามีความซับซ้อนมาก

3. จำเป็นต้องใช้ช่างเขียนโปรแกรม (Part Programmer) ที่มีทักษะสูงและฝึกอบรมมาโดยเฉพาะ

4. ชิ้นส่วนหรืออะไหล่ที่ใช้ในการซ่อมบำรุงไม่สามารถผลิตได้ในประเทศ จำเป็นต้องสั่งซื้อหรือนำเข้าจากต่างประเทศ

5. การซ่อมบำรุงจะต้องใช้ช่างที่มีประสบการณ์สูงและผ่านการฝึกอบรมมาโดยเฉพาะ

6. ราคาของเครื่องมือต่างๆ ที่ใช้ในกระบวนการตัดเฉือน เช่น แกนเพลายึดมีดกัด มีดกลึงแบบใช้อินเสิร์ท (Insert) เป็นต้น มีราคาสูง

7. พื้นที่ติดตั้งเครื่องจักร จะต้องควบคุมระดับอุณหภูมิ ความชื้น และฝุ่นละออง

ข้อมูลเหล่านี้ จำเป็นจะต้องพิจารณาอย่างรอบคอบก่อนที่จะพิจารณาจัดซื้อ ซึ่งสามารถสอบถามได้จากบริษัทผู้ผลิตหรือตัวแทนจำหน่ายได้โดยตรง

2.2 เครื่องจักรกลเอ็นซีและซีเอ็นซี

2.2.1 การทำงานของเครื่องจักรกลเอ็นซี

หลักการทำงานของเครื่องจักรกลเอ็นซีหรือซีเอ็นซี จะคล้ายคลึงกับเครื่องจักรกลทั่วไป กล่าวคือโดยพื้นฐานเบื้องต้นแล้วเครื่องจักรกลเอ็นซีก็จะทำการผลิตชิ้นงานเหมือนกับเครื่องจักรกลทั่วไป เช่น เครื่องกัดเอ็นซี ก็จะทำงานเหมือนกับเครื่องกัดทั่วไป เพียงแต่ว่าระบบควบคุมเอ็นซีของเครื่องจะทำงานในขั้นตอนต่างๆ แทนช่างควบคุมเครื่อง อย่างไรก็ตาม ก่อนที่เครื่องจักรกลเอ็นซีจะสามารถทำงานได้นั้น ระบบควบคุมของเครื่องจะต้องได้รับการบอกกล่าวเสียก่อนว่าจะให้ทำอะไร และจะต้องบอกกล่าวเป็นภาษาที่ระบบควบคุมสามารถเข้าใจได้ นั่นคือ จะต้องป้อนโปรแกรมเข้าไปในระบบควบคุมของเครื่องผ่านแป้นพิมพ์(keyboard)หรือเทปแม่เหล็ก(magnetic tape) ก็ได้

เมื่อระบบควบคุมอ่านโปรแกรมที่ป้อนเข้าไปแล้ว ก็จะนำไปควบคุมให้เครื่องจักรกลทำงาน แต่เนื่องจากเครื่องจักรกลเอ็นซีไม่มีมือสำหรับหมุนมือหมุนให้แท่นเลื่อนเคลื่อนที่ได้

ดังนั้น แท่นเลื่อนต่างๆ จะต้องมีมอเตอร์ป้อน (feed motor) ประกอบอยู่ เช่น เครื่องกัดซีเอ็นซีจะมีการเคลื่อนที่ 3 แนวแกน ก็จะมีมอเตอร์ป้อน 3 ตัว

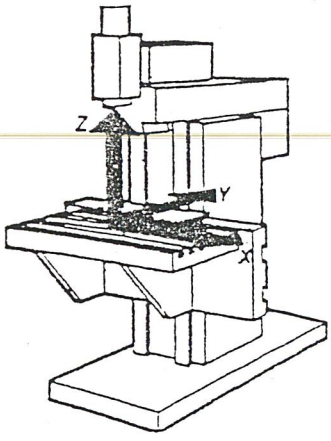
เมื่อระบบควบคุมอ่านโปรแกรมแล้ว ก็จะเปลี่ยนรหัสโปรแกรมนั้นให้เป็นสัญญาณทางไฟฟ้า เพื่อไปควบคุมให้มอเตอร์ทำงาน แต่เนื่องจากสัญญาณที่ออกจากระบบควบคุมนี้มีกำลังน้อย ไม่สามารถไปหมุนขับให้มอเตอร์ทำงานได้ ดังนั้น จึงต้องส่งสัญญาณนี้เข้าไปในภาคขยายสัญญาณของระบบขับ (drive amplified) และส่งต่อไปยังมอเตอร์ป้อนของแนวแกนที่ต้องการเคลื่อนที่

ความเร็วและระยะทางการเคลื่อนที่ของแท่นเลื่อน จะต้องกำหนดให้ระบบควบคุมรู้ ช่างควบคุมเครื่องอาศัยสายตามองดูตำแหน่งของคมตัดกับชิ้นงาน ก็จะรู้ว่าจะต้องเลื่อนแท่นเลื่อนไปอีกเป็นระยะทางเท่าใด แต่ระบบควบคุมเอ็นซีมองไม่ได้ ดังนั้น จึงต้องออกแบบอุปกรณ์หรือเครื่องมือที่สามารถจะบอกตำแหน่งของแท่นเลื่อนให้ระบบควบคุมรู้ได้ อุปกรณ์ชุดนี้เรียกว่า **ระบบวัดขนาด (Measuring System)** ซึ่งประกอบด้วยสเกลแนวตรง (Linear Scale) มีจำนวนเท่ากับจำนวนแนวแกนในการเคลื่อนที่ของเครื่องจักรกล ทำหน้าที่ส่งสัญญาณไฟฟ้าที่สัมพันธ์กับระยะทางที่แท่นเลื่อนเคลื่อนที่กลับไปยังระบบควบคุม ทำให้ระบบควบคุมรู้ว่าแท่นเลื่อนเคลื่อนที่ไปเป็นระยะทางเท่าใดแล้ว

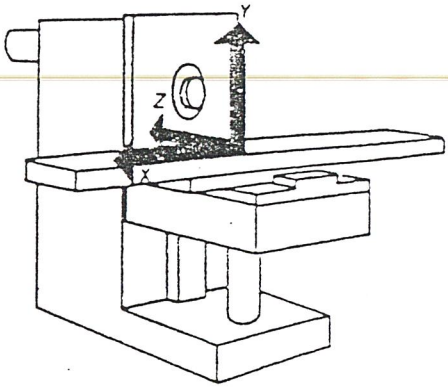
จากหลักการควบคุมการทำงานดังกล่าว ทำให้เครื่องจักรกลซีเอ็นซีสามารถผลิตชิ้นงานให้มีรูปทรงและขนาดที่ต้องการได้ จากลักษณะสร้างและการทำงานที่เหนือกว่าเครื่องจักรกลทั่วไป ทำให้เครื่องจักรกลเอ็นซีและซีเอ็นซี เป็นปัจจัยหนึ่งที่มีความสำคัญมากในอุตสาหกรรมอัตโนมัติ และมีปริมาณความต้องการใช้เพิ่มมากขึ้นเรื่อยๆ

2.2.2 เครื่องกัดเอ็นซี (NC Milling Machines)

เครื่องกัดเอ็นซีเป็นเครื่องจักรกลประเภทหนึ่งที่มีขอบข่ายการทำงานค่อนข้างกว้าง กล่าวคือ นอกจากจะสามารถทำงานกัดเช่นเดียวกับเครื่องกัดทั่วไปแล้ว ยังสามารถทำงานอื่นๆ เช่น เจาะรู ทำเกลียว คว้านรู ได้อีกด้วย โดยทั่วไปเครื่องกัดเอ็นซีจะแบ่งออกเป็น 2 ประเภทใหญ่ๆ คือ เครื่องกัดเอ็นซีเพลาตั้ง กับเครื่องกัดเอ็นซีเพลานอน ซึ่งขึ้นอยู่กับการวางตำแหน่งของเพลาหัวเครื่อง เครื่องกัดเอ็นซีจะมีแนวแกนการควบคุมตั้งแต่ 3 แกน 4 แกน 5 แกน และมากกว่า ดังแสดงในรูป 2.1 และ 2.2

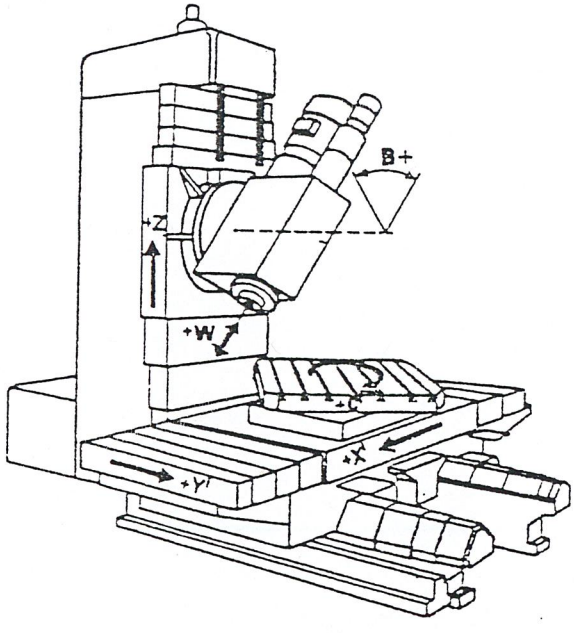


ก) เพลาตั้ง



ข) เพลานอน

รูปที่ 2.1 เครื่องกัด CNC แบบ 3 แกน

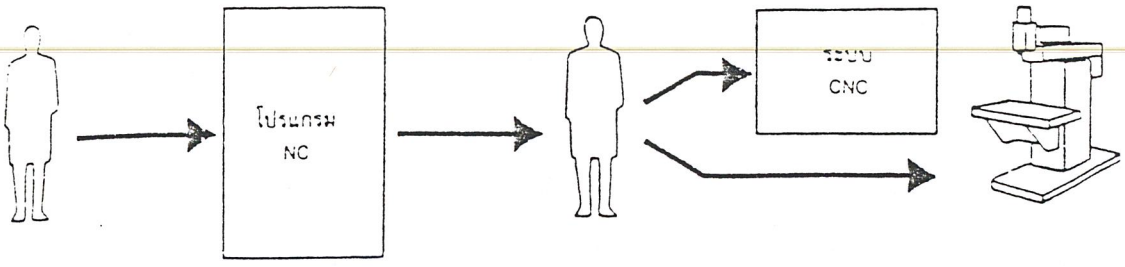


รูปที่ 2.2 เครื่องกัด CNC แบบ 5 แกนที่เอียงเพลามีดได้และมีโต๊ะงานหมุน

2.3 ระบบควบคุมเครื่องจักรกลด้วยตัวเลข

เครื่องจักรกลซีเอ็นซี จะประกอบด้วยองค์ประกอบใหญ่ๆ อยู่ 2 ส่วน คือ

1. **เครื่องจักรกล** เป็นส่วนที่ทำหน้าที่ตัดเฉือนชิ้นงานตามขั้นตอนการทำงานที่กำหนดไว้
2. **ระบบซีเอ็นซี** เป็นส่วนที่ทำหน้าที่ควบคุมขั้นตอนการตัดเฉือนทั้งหมด

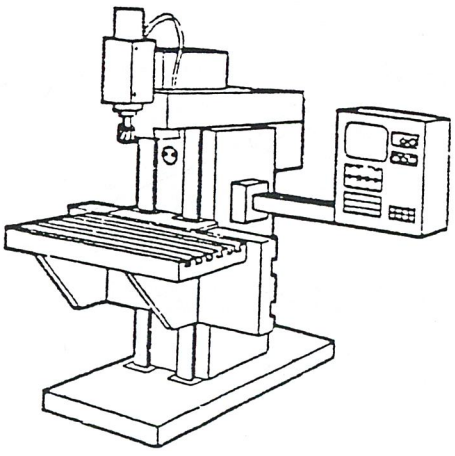


รูปที่ 2.3 องค์ประกอบของเครื่องจักรกล CNC

ข้อมูลเดิมที่อธิบายรายละเอียดของขั้นตอนที่ใช้ในการตัดเฉือนชิ้นงาน จะถูกป้อนเข้าไปในระบบควบคุมของเครื่องจักรกลก่อน ในรูปแบบของโปรแกรมเอ็นซี ซึ่งถูกจัดเตรียมโดยช่างเขียนโปรแกรม ช่างควบคุมเครื่องจะเป็นผู้ป้อนโปรแกรมเข้าไปในระบบควบคุม ซึ่งอาจป้อนด้วยมือผ่านแป้นพิมพ์โดยตรง หรือใช้แถบกระดาษเจาะรู (punched tape) ก็ได้ หลังจากนั้น ก็จะเดินเครื่องทดลองโปรแกรม และสังเกตสภาวะการตัดเฉือนชิ้นงานในแต่ละขั้นตอน บ่อยครั้งที่ช่างควบคุมเครื่องจะต้องจัดเตรียมโปรแกรม หรือเขียนโปรแกรมด้วยตนเอง หรือแก้ไขปรับปรุงโปรแกรมให้มีประสิทธิภาพในการตัดเฉือนสูงสุด ดังนั้น จึงเป็นสิ่งจำเป็นที่ช่างควบคุมเครื่องจะต้องมีความรู้ทั้งระบบควบคุมของเครื่องจักรกลและการเขียนโปรแกรมเอ็นซีด้วย

2.3.1 องค์ประกอบของเครื่องจักรกลที่ควบคุมได้

องค์ประกอบหรือชิ้นส่วนของเครื่องจักรกล ที่ทำหน้าที่เคลื่อนที่เข้าตัดเฉือนชิ้นงาน และองค์ประกอบอื่นๆ ที่ช่วยเสริมการทำงานตัดเฉือนให้สมบูรณ์ยิ่งขึ้น จะถูกควบคุมโดยโปรแกรมเอ็นซี ด้วยวิธีการควบคุมแบบต่าง ๆ กัน



รูปที่ 2.4 เครื่องกัด CNC

ช่างชำนาญงานที่ทำหน้าที่ควบคุมการทำงานของเครื่องจักรกลเอ็นซีหรือซีเอ็นซี จะต้องมีความคุ้นเคยกับหน้าที่การทำงานและขีดจำกัดในการทำงานของเครื่องจักรกลซีเอ็นซีนั้นเป็นอย่างดี ช่างจะใช้วิธีการทำงานแบบง่ายๆ โดยการจับยึดชิ้นงานเข้ากับโต๊ะงานและคาดว่าจะได้วิธีการตัดเฉือนที่ดีที่สุดไม่ได้ ในทางตรงข้ามช่างจะต้องจัดวางแผนขั้นตอนการทำงานไว้ล่วงหน้าเพื่อให้ได้ผลผลิตที่ดี ดังนั้นจึงเป็นสิ่งจำเป็นที่ช่างจะต้องรู้ว่าองค์ประกอบส่วนใดของเครื่องจักรกลซีเอ็นซีที่สามารถควบคุมได้และมีวิธีการควบคุมอย่างไร องค์ประกอบของเครื่องจักรกลเอ็นซีและซีเอ็นซีที่สามารถควบคุมได้ และจะกล่าวถึงในที่นี้ได้แก่

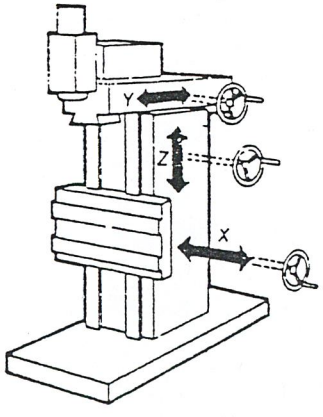
- แนวแกนป้อน (Feed axes)
- การขับป้อน (Feed drives)
- อุปกรณ์วัดขนาด (Measuring devices)
- เพลางาน (Work Spindle)
- อุปกรณ์จับยึดชิ้นงาน (Workpiece holding devices)

1) แนวแกนป้อน (Feed axes)

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

สำหรับเครื่องจักรกลทั่วไป การเคลื่อนที่ในแนวแกนต่างๆ จะเกิดจากการหมุนมือหมุนหรือโยกคันโยกป้อนอัตโนมัติ (Feed levers)

เครื่องจักรกลซีเอ็นซีมีแนวแกนป้อนรวมกันอยู่หลายแนวแกนทำให้ สามารถ ตัดเฉือนชิ้นงานให้เป็นรูปทรงต่างๆ ที่ต้องการได้ การกำหนดแนวแกนต่างๆของเครื่องจักรกลซีเอ็นซีจะกำหนดตามมาตรฐานสากลภายใต้หัวเรื่อง Coordinate axes and direction of movement for numerically controlled machinery ซึ่งจะกำหนดแนวแกนเหล่านี้โดยใช้ตัวอักษร x,y และ z ดังแสดงในรูป 2.5



รูปที่ 2.5 แท่นเลื่อนแบบ 3 แนวแกน

แนวแกนทั้ง 3 แนวแกนที่แสดงในรูป จะทำให้เกิดการเคลื่อนที่ต่างๆ ดังนี้

แนวแกน x : โต๊ะงานเคลื่อนที่ไปทางซ้ายและขวา

แนวแกน y : เพลาหัวเครื่องเคลื่อนที่เข้าและออก

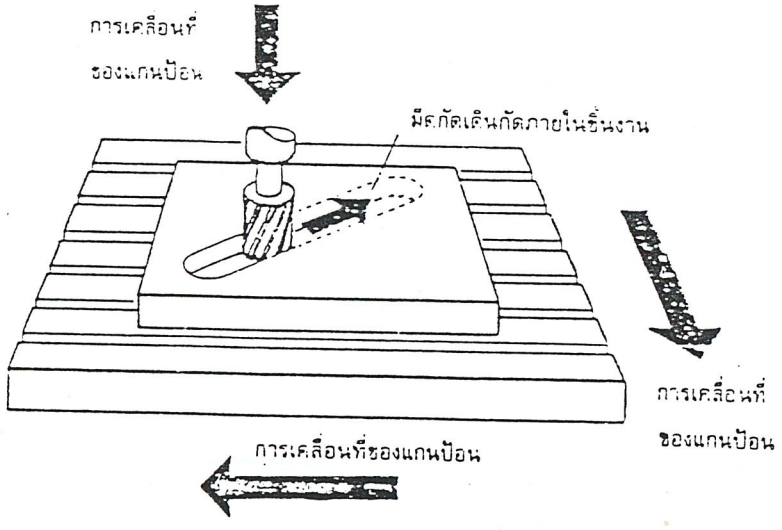
แนวแกน z : โต๊ะงานเคลื่อนที่ขึ้นและลง

เครื่องกัดจะมีแนวแกนป้อนอยู่ 3 แนวแกนด้วยกัน คือ แกน x , y และ z โดยทั่วไปจะมี 2 แกน สำหรับการเคลื่อนที่ของโต๊ะงาน ส่วนแกนที่ 3 จะเป็นการเคลื่อนที่ของเพลา หัวเครื่อง (เพลางาน) ถ้าเครื่องกัดนั้นเป็นแบบโต๊ะงานอยู่กับที่เพลาหัวเครื่องจะเคลื่อนที่ทั้ง 3 แนวแกน

สำหรับเครื่องจักรกลซีเอ็นซี ที่ใช้ผลิตชิ้นงานที่มีรูปทรงซับซ้อนมาก จะมีจำนวนแนวแกนป้อนเพิ่มมากขึ้น

2) การขับป้อน (Feed drives)

การเคลื่อนที่เรียงลำดับกันหรือพร้อม ๆ กันอย่างต่อเนื่องของแนวแกนป้อน จะทำให้เกิดการตัดเฉือนของเครื่องมือในชิ้นงาน ดังรูป 2.6

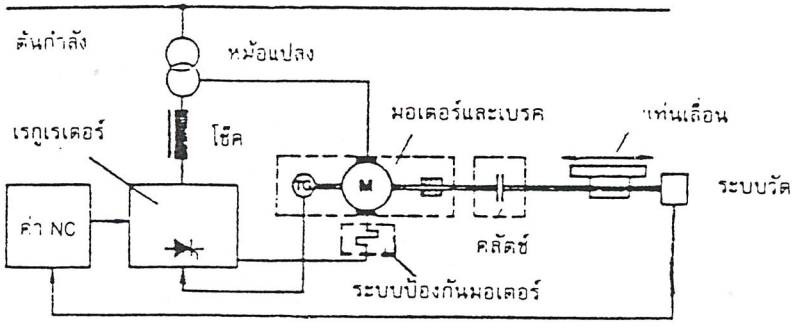


รูปที่ 2.6 การเคลื่อนที่ตัดเฉือนของเครื่องมือตัด

การขับป้อนจะทำให้เกิดการเคลื่อนที่ของแท่นเลื่อน ในขณะตัดเฉือนแท่นเลื่อนอาจพาให้ชิ้นงานเคลื่อนที่หรือคมตัดเคลื่อนที่ก็ได้

ระบบขับป้อนโดยทั่วไปจะใช้มอเตอร์กระแสตรงในการหมุนขับและควบคุมการทำงาน ด้วยวงจรอิเล็กทรอนิกส์จากภายนอก มอเตอร์ชนิดนี้จะสามารถหมุนและเบรคให้หยุดได้ทั้งสองทิศทางขณะตัดเฉือนชิ้นงาน การเคลื่อนที่ป้อนจะต้องเป็นไปอย่างสม่ำเสมอและสามารถต้านแรงกระทำจากภายนอกได้ เช่น แรงตัดเฉือน เป็นต้น ด้วยเหตุนี้ ระบบขับป้อนจึงต้องได้รับการ

ออกแบบให้มีความแข็งแกร่งสูง มีการเคลื่อนที่คงที่และสม่ำเสมอ สามารถตอบสนองต่อการเปลี่ยนอัตราป้อนได้อย่างรวดเร็ว นอกจากนี้ใน ขณะทำงานคมตัดอาจทื่อ หรือการเคลื่อนที่ของแท่นเลื่อนถูกกีดขวาง หรือการเร่งอัตราป้อนให้เคลื่อนที่เร็วและหยุดโดยทันทีทันใด สาเหตุเหล่านี้จะทำให้มอเตอร์รับภาระมากเกินไป (over loading) ซึ่งอาจทำให้มอเตอร์เสียหายได้ ดังนั้น จึงต้องมีการป้องกันอุบัติเหตุเหล่านี้ โดยทั่วไปแล้วจะใช้คลัตช์แบบลูกกลิ้ง (Over running clutch) ร่วมกับ วงจรอิเล็กทรอนิกส์ ปัจจัยหนึ่งที่จะทำให้ระบบขับป้อนสามารถทำงานได้อย่างมีประสิทธิภาพก็คือ การเลือกใช้อุปกรณ์ในระบบขับป้อนให้เหมาะสมกับการทำงานของเครื่องจักรและการออกแบบวงจรควบคุมการทำงานที่มีประสิทธิภาพดังแสดงในรูป2.7

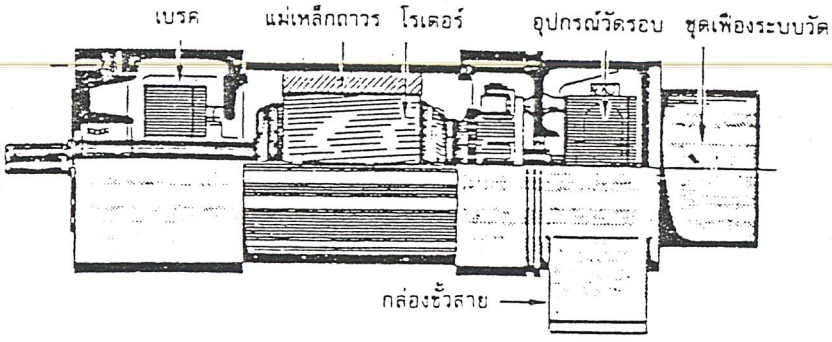


รูปที่ 2.7 Diagram ระบบขับป้อน

2.1) มอเตอร์

เครื่องจักรกลเอ็นซีสมัยใหม่จะออกแบบใช้ระบบขับป้อนแบบเซอโว (servo drives) ทำให้สามารถปรับอัตราป้อนและความเร็วรอบได้โดยไม่มีขีดจำกัดของขั้นความเร็วและอัตราป้อน มอเตอร์ที่ใช้ในระบบขับป้อนโดยทั่วไปจะมีอยู่ 3 ชนิดด้วยกันคือ

ก) มอเตอร์กระแสตรง (DC motors) ลักษณะสร้างของมอเตอร์กระแสตรงจะใช้เป็นแม่เหล็กถาวรที่มี 4 , 6 หรือ 8 ขั้ว ประกอบด้วยระบบเบรค (brake) แกนมอเตอร์(Rotor) อุปกรณ์วัดรอบ (Tachogenerator) และอุปกรณ์วัด (Measuring box) ดังแสดงในรูป 2.8

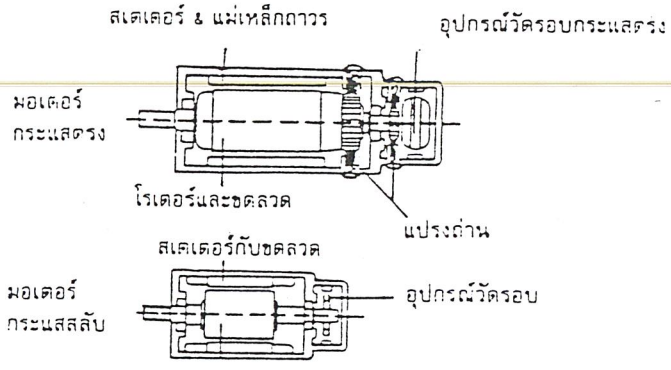


รูปที่ 2.8 ส่วนประกอบของมอเตอร์กระแสตรง

การใช้มอเตอร์กระแสตรง ทำให้สามารถปรับอัตราป้อนได้ละเอียดและมีวงจรควบคุมที่ไม่ซับซ้อน แต่ก็มีข้อเสียตรงที่มอเตอร์ชนิดนี้ต้องใช้แปรงถ่าน ซึ่งจะต้องคอยทำความสะอาดและเปลี่ยนเมื่อแปรงถ่านหมด นอกจากนี้ แปรงถ่านยังทำให้แกนมอเตอร์สึกหรออันเป็นผลทำให้กำลังมอเตอร์ลดลง ข้อเสียอีกประการหนึ่งก็คือ หากต้องการ กำลังขับสูง มอเตอร์ก็จะมีขนาดใหญ่ด้วย และเมื่อใช้ความเร็วรอบสูงๆ จะทำให้แรงบิดลดลง ดังนั้น จึงมักใช้กับเครื่องจักรกลเอ็นซีขนาดเล็กและขนาดกลาง

ข) มอเตอร์แบบเป็นขั้น (Stepping motors) เป็นมอเตอร์ที่ทำงานแบบต่อเนื่องโดยการแปลงคลื่นสัญญาณที่ป้อนเข้าไปในระบบให้เป็นการเคลื่อนที่เชิงมุม การหมุนในแต่ละมุมหรือขั้นที่เปลี่ยนไป 1 ขั้นจะเท่ากับ 1 คลื่นสัญญาณ ดังนั้น ตำแหน่งของเพลาจะถูกกำหนดโดยจำนวนคลื่นสัญญาณที่ป้อนเข้าไปในระบบ และความเร็วในการหมุนของเพลาจะวัดเป็นจำนวนขั้นต่อวินาที (Steps per second) ซึ่งจะเท่ากับความถี่ของคลื่น สัญญาณที่ป้อนเข้าในระบบที่วัดเป็นจำนวนคลื่นสัญญาณต่อวินาที (pulses per second) ความเที่ยงตรงของระบบจะขึ้นอยู่กับความสามารถของมอเตอร์ในการแบ่งขั้นการหมุนตามจำนวนคลื่นสัญญาณที่ป้อนเข้าในระบบแรงบิดของมอเตอร์ชนิดนี้จะลดลงเมื่อความเร็วในการหมุนแบ่งเพิ่มขึ้น ดังนั้น จึงเหมาะสำหรับเครื่องจักรกลเล็กๆ ที่ไม่ต้องใช้กำลังขับมาก เช่น เครื่องพลอตเตอร์ (Plotter machine) เป็นต้น

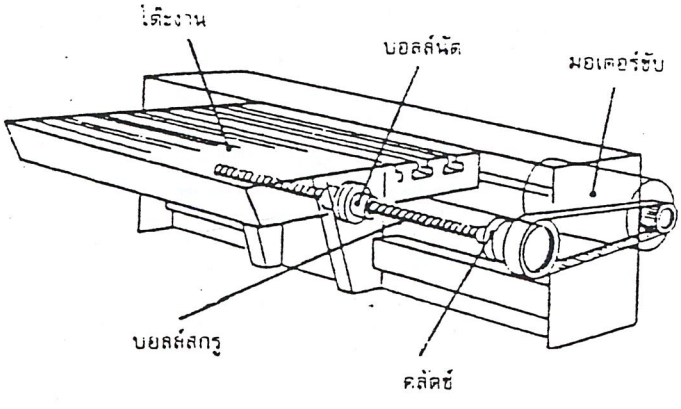
ค) มอเตอร์กระแสสลับ (Alternate-current motor) ส่วนมากจะเป็นมอเตอร์แบบซิงโครนัส (Synchronous motor) ข้อดีของมอเตอร์ชนิดนี้คือ ไม่ต้องใช้แปรงถ่าน ทำให้สามารถลดงานบำรุงรักษาได้มาก และมอเตอร์ขนาดเดียวกันเมื่อเปรียบเทียบกับมอเตอร์กระแสตรงจะสามารถให้แรงบิดได้ดีกว่า และมีขนาดเล็กกว่าด้วย ดังแสดงในรูป 2.9 ส่วนข้อเสียของมอเตอร์แบบนี้คือวงจรควบคุมจะมีความซับซ้อนมากกว่าวงจรควบคุมมอเตอร์กระแสตรง



รูปที่ 2.9 การเปรียบเทียบลักษณะสร้างและขนาดของ
มอเตอร์กระแสตรงกับมอเตอร์กระแสสลับแบบ 3 เฟส

2.2) บอลล์สกรู (Ball screws)

หัวใจของระบบขับป้อนของเครื่องจักรกลซีเอ็นซี ก็คือ การส่งกำลังขับด้วย บอลล์สกรู ซึ่งจะมีลูกบอลล์ไหลหมุนเวียนอยู่ตลอดเวลา บอลล์สกรูจะประกอบด้วยสกรูกับนัตที่มีลักษณะเป็นเกลียวกลม ร่องเกลียวกลมบนสกรูและในนัตจะชุบแข็งและเจียระไนผิวเรียบมันเพื่อลดความฝืดและเพิ่มความเที่ยงในการเคลื่อนที่ ดังแสดงในรูป 2.10

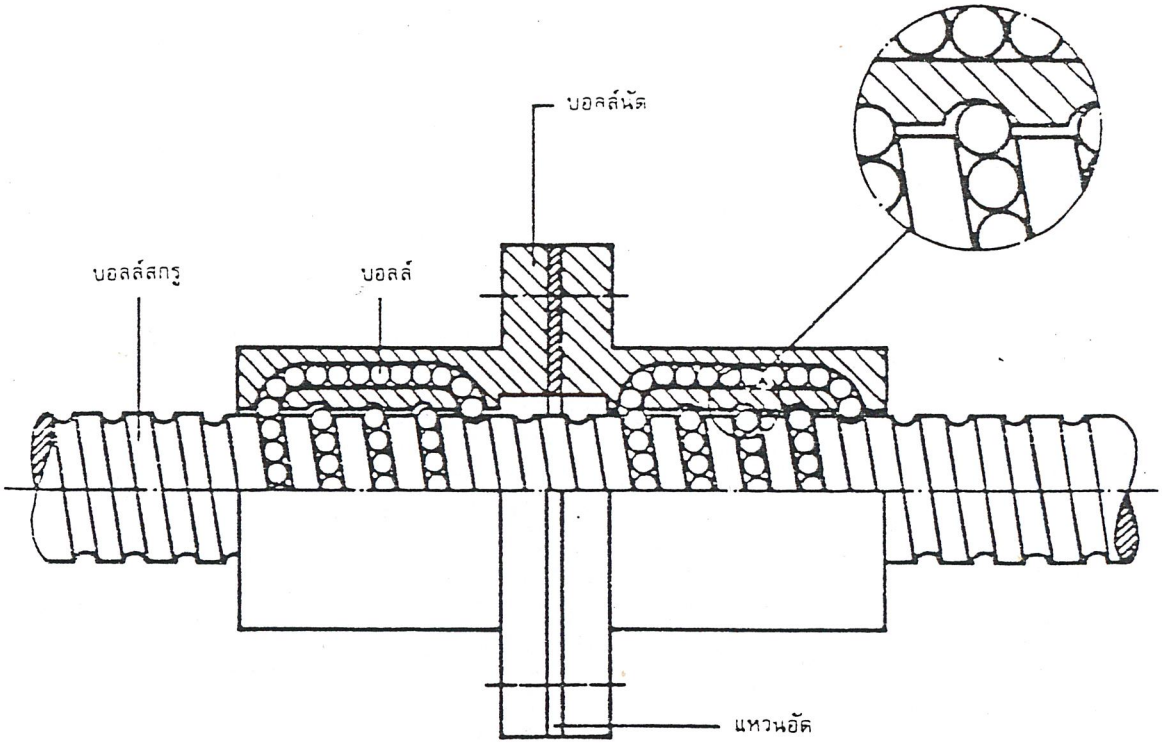


รูปที่ 2.10 การขับป้อนของโต๊ะงาน

เมื่อมอเตอร์หมุนขับสกรู นัตก็จะเคลื่อนที่ไปตลอดความยาวของสกรูพาให้แท่นเลื่อนและโต๊ะงานเคลื่อนที่ไปตามรางเลื่อน

ภายในของตัวนัตจะประกอบไปด้วยชุดของลูกบอลล์จำนวนมาก ดังแสดงในรูป2.11 ทำให้มั่นใจได้ว่าความเสียดทานในการส่งกำลังขับจากสกรูไปยังแท่นเลื่อนจะมีน้อยมาก

นัตจะถูกแบ่งออกเป็นสองซีก และ ขันประกอบยึดเข้าด้วยกัน โดยมีการเตรียมอัดแรงไว้ก่อน (preloaded) ทำให้สามารถลดระยะคลอน (backlash) ให้เหลือน้อยที่สุดจนแทบจะไม่มีเลยได้ทำให้การเคลื่อนที่ของแท่นเลื่อนมีความเที่ยงตรงสูงสามารถทำงานซ้ำๆกันได้



รูปที่ 2.11 ลักษณะสร้างภายในของชุดบอลล์สกรู

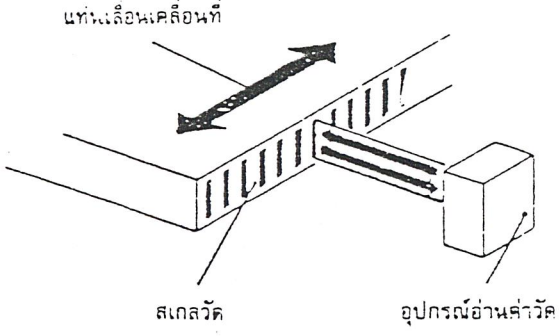
การต่อกำลังระหว่างมอเตอร์กับบอลล์สกรู จะมีชุดคลัตช์ความฝืดเป็นตัวเชื่อม ซึ่งนอกจากจะมีหน้าที่ต่อกำลังขับแล้ว ยังมีหน้าที่ป้องกันอุบัติเหตุที่เกิดจากแท่นเลื่อนหรือโต๊ะงานชนหรือกระแทกกับสิ่งกีดขวางไม่ให้เครื่องจักรกลซีเอ็นซีเกิดความเสียหายมากเกินไป กล่าวคือ เมื่อมีการชน หรือกระแทกกันขึ้นจนแรงมากถึงค่าหนึ่ง ชุดคลัตช์ก็จะตัดระบบการส่งกำลังขับระหว่างมอเตอร์กับตัวบอลล์สกรูทันที

3) ระบบวัดขนาด (Measuring System)

การเคลื่อนที่ไปที่ตำแหน่งต่างๆ ในแต่ละแนวแกนของแท่นเลื่อน จะถูกส่งไปยังระบบควบคุมโดยระบบวัดขนาด การวัดตำแหน่งของแท่นเลื่อนสามารถที่จะวัดได้ทั้งโดยตรง (Direct Measurement) และโดยทางอ้อม (Indirect Measurment)

3.1) การวัดตำแหน่งโดยตรง

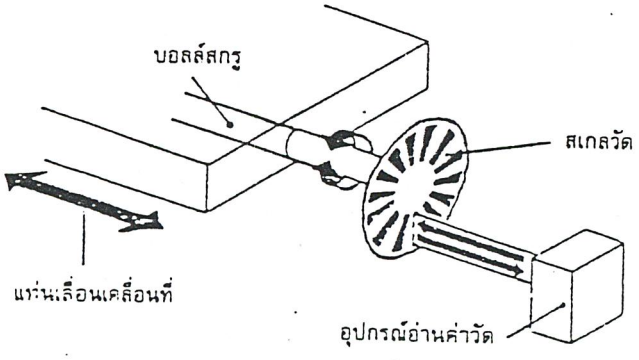
วิธีนี้จะใช้สเกลวัด (measuring scale) ยึดติดกับแท่นเลื่อนหรือโต๊ะงานโดยตรง ดังแสดงในรูป 2.12 ข้อดีของวิธีวัดแบบนี้ก็คือ ความไม่เที่ยงตรงของสกรูนำเลื่อน(leadscrew) ระบบขับจะไม่มีผลกระทบต่อค่าที่อ่านได้ อุปกรณ์อ่านค่าวัด(Measuring valve resolver) จะอ่านข้อมูลในการวัดจากขีดสเกลวัด (Measuring scale grid) และแปลงข้อมูลนี้เป็นสัญญาณไฟฟ้าและส่งกลับไปยังระบบควบคุม



รูปที่ 2.12 การวัดตำแหน่งโดยตรง

3.2) การวัดตำแหน่งทางอ้อม

การเคลื่อนที่ของแท่นเลื่อนจะได้รับกำลังขับมาจากการหมุนของบอลล์สกรู อุปกรณ์เปลี่ยนค่าวัด (Resolver) จะบันทึกการเคลื่อนที่หมุนของแผ่นจานสัญญาณ (pulse disc) ที่ต่อติดอยู่กับบอลล์สกรู และส่งต่อไปยังระบบควบคุมของเครื่อง ระบบควบคุมก็จะใช้สัญญาณ ได้รับนี้ไปคำนวณหาระยะทางการเคลื่อนที่ของแท่นเลื่อนจากสัญญาณการหมุน (rotation pulses) ของแผ่นจานสัญญาณ ดังแสดงในรูป 2.13



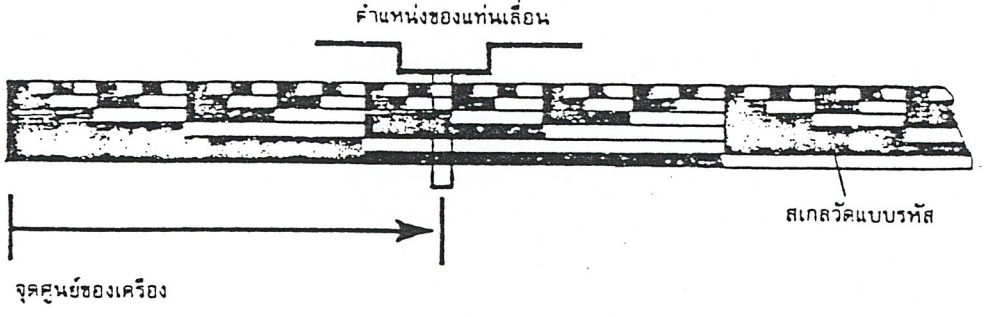
รูปที่ 2.13 การวัดตำแหน่งทางอ้อม

นอกจากการวัดตำแหน่งทางตรงและทางอ้อมแล้ว ในระบบการวัดขนาดของเครื่องจักรกลเอ็นซีที่ต้องการให้การวัดตำแหน่งมีความเที่ยงตรงตลอดแนวแกนป้อนจะต้องต่อระบบขับป้อนเข้ากับอุปกรณ์วัดที่เหมาะสม อุปกรณ์วัดโดยทั่วไปจะประกอบด้วยสเกลกับอุปกรณ์อ่านค่าวัดที่สามารถอ่านสเกลได้ สเกลที่ใช้ในอุปกรณ์วัดมีอยู่ด้วยกัน 2 ชนิด คือ สเกลวัดแบบรหัส (coded measuring scale) กับสเกลวัดแบบช่อง (division grid) การใช้สเกลวัดทั้ง 2 ชนิดนี้จะขึ้นอยู่กับวิธีการวัดตำแหน่ง (position measurement) วิธี การวัดตำแหน่งที่นิยมใช้กันทั่วไปมีอยู่ 2 วิธีคือ การวัดตำแหน่งแบบสัมบูรณ์ absolute position measurement) กับการวัดตำแหน่งแบบต่อเนื่อง หรือแบบลูกโซ่ (incremental or chain position measurement) ซึ่งมีความแตกต่างกันดังรายละเอียดต่อไปนี้

3.3) การวัดตำแหน่งแบบสัมบูรณ์

คำว่า **สัมบูรณ์** (absolute) ที่ใช้ร่วมกับการวัดตำแหน่งนี้จะหมายความว่าค่าตำแหน่งต่างๆ สามารถวัดได้ตลอดเวลาและเป็นอิสระจากสถานะของเครื่องและระบบควบคุม ทั้งนี้เพราะค่าต่างๆเหล่านี้จะวัดอ้างอิงจากจุดศูนย์อ้างอิง (fixed zero datum) เสมอ

การวัดตำแหน่งแบบสัมบูรณ์ ดังแสดงในรูป 2.14 จะใช้สเกลวัดแบบรหัส (coded measuring scale) ซึ่งจะชี้ตำแหน่งของแท่นเลื่อนที่ถูกต้องตลอดเวลา โดยอ้างอิงจากตำแหน่งจุดศูนย์ของเครื่อง (Machine zero point) ซึ่งเป็นตำแหน่งศูนย์ที่มีจุดอ้างอิงที่แน่นอนและถาวรของเครื่องจักรกลเอ็นซี



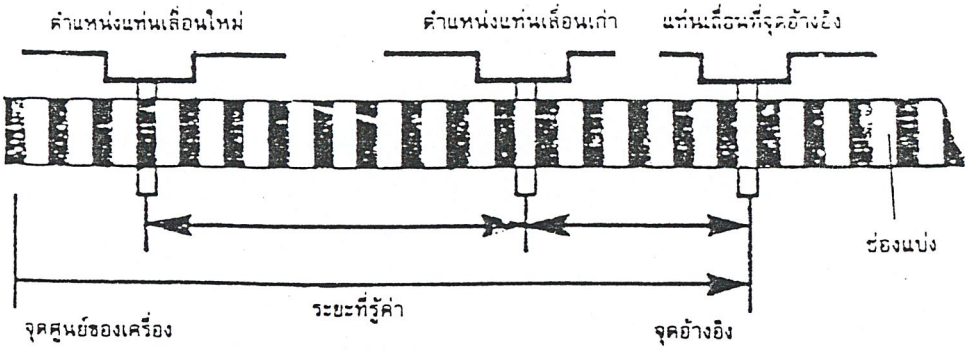
รูปที่ 2.14 การวัดตำแหน่งแบบสัมบูรณ์

ข้อสำคัญของการใช้วิธีการวัดตำแหน่งแบบนี้ก็คือ ความยาวของช่วงอ่านค่าวัดของสเกลจะต้องยาวกว่าระยะเลื่อนทำงานของแท่นเลื่อนเพื่อให้ระบบควบคุมของเครื่องสามารถอ่านค่าวัดได้ทุกตำแหน่งสเกลนี้จะใช้รหัสเป็นระบบตัวเลขฐานสอง (Binary system)

3.4) การวัดตำแหน่งแบบต่อเนื่อง

คำว่า **ต่อเนื่อง** (incremental) แปลว่า ระยะเลื่อนสั้นๆ ตามความยาวที่กำหนดดังนั้นในการวัดตำแหน่งอาจจะเป็นการเพิ่มหรือลดขนาดความยาวในการเคลื่อนที่ที่วัดอยู่ก็ได้ ในระหว่างการเคลื่อนที่ของแท่นเลื่อนระบบควบคุมจะทำการนับจำนวนส่วนแบ่ง(divisions)ที่ตำแหน่งใหม่แตกต่างจากตำแหน่งก่อนหน้านี้เสมอ

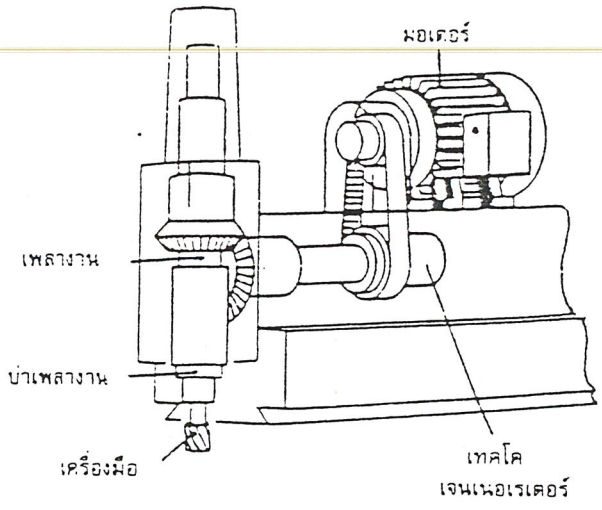
วิธีการวัดตำแหน่งแบบต่อเนื่องที่แสดงในรูป 2.15 สเกลวัดจะแบ่งเป็นช่อง (grid) แบบง่ายๆโดยที่แต่ละช่องจะมีพื้นที่สว่างกับมืดสลับกันไป เมื่อแท่นเลื่อนเคลื่อนที่ช่องนี้ก็จะวิ่งผ่านอุปกรณ์เปลี่ยนค่าวัด (resolver) ซึ่งจะทำหน้าที่นับจำนวนช่องพื้นที่สว่างและมืด จากนั้นก็จะส่งเป็นสัญญาณไฟฟ้าไปยังระบบควบคุมของเครื่องระบบควบคุมก็จะนำสัญญาณนี้มาคำนวณหาตำแหน่งสุดท้ายของแท่นเลื่อนที่แตกต่างจากตำแหน่งก่อนหน้านี้ ในทางปฏิบัติที่ต้องการให้วิธีการวัดแบบนี้ทำงานได้อย่างถูกต้อง เมื่อเริ่มเปิดสวิตช์ระบบควบคุมของเครื่องควรจะเลื่อนไปยังจุดที่ทราบค่าระยะห่างจากจุดศูนย์ของเครื่องจุดนี้จะเรียกว่า "จุดอ้างอิง" (Reference Point) หลังจากที่แท่นเลื่อนในแนวแกนต่างๆ เลื่อนไปยังจุดอ้างอิงแล้ว อุปกรณ์อ่านค่าวัดก็จะสามารถทำหน้าที่วัดตำแหน่งด้วยช่องสเกลได้



รูปที่ 2.15 การวัดตำแหน่งแบบต่อเนื่อง

4) เพลางาน (Work Spindle)

เพลางานเป็นชิ้นส่วนหรือองค์ประกอบของเครื่องจักรกลที่มีความสำคัญมากมีหน้าที่หลักในการทำงาน คือ จะทำหน้าที่ขับพาให้เครื่องมือ เช่น มีดกัด ดอกสว่านเป็นต้น หมุนตัดเฉือนชิ้นงาน

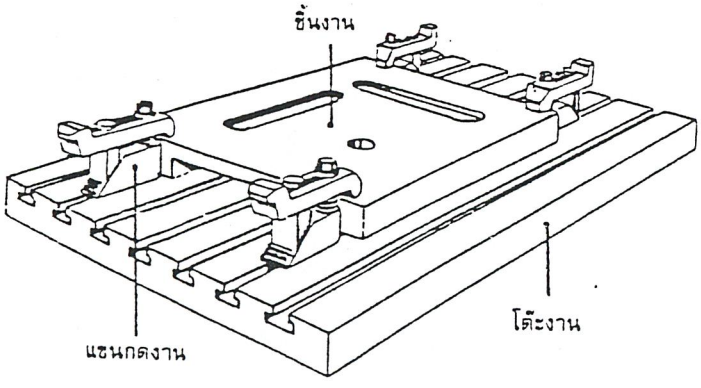


รูปที่ 2.16 เพลางานของเครื่องกัด

5) อุปกรณ์จับยึดชิ้นงาน (Workpiece holding devices)

อุปกรณ์จับยึดชิ้นงานจะจัดเตรียมไว้สำหรับยึดชิ้นงานเข้ากับโต๊ะงานในงานกัด (Milling) สามารถเลือกใช้อุปกรณ์จับยึดชิ้นงานแบบต่างๆ กันได้ดังนี้

- แขนกดชิ้นงาน
- แองเกิล เพลท (angle plate)
- ปากกาจับชิ้นงาน
- แท่นแม่เหล็ก
- อุปกรณ์จับชิ้นงานที่ออกแบบเฉพาะงาน



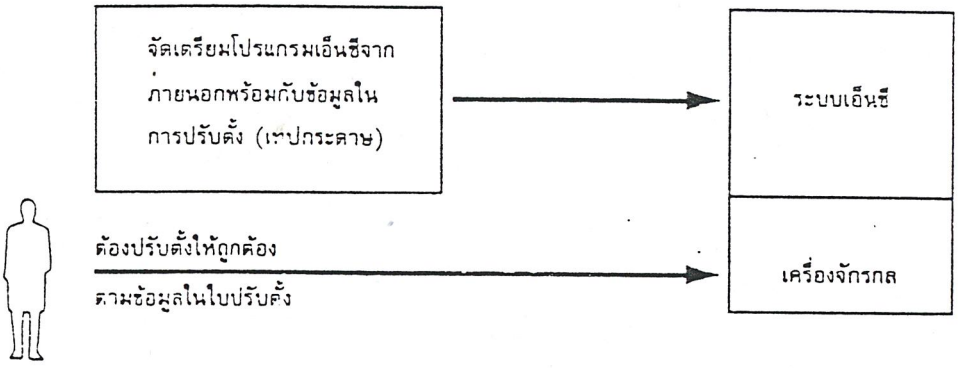
รูปที่ 2.17 แขนกดจับชิ้นงานกัด

2.4 ระบบควบคุมซีเอ็นซี (CNC Control System)

2.4.1 หน้าที่การทำงานที่โปรแกรมได้ (Programmable Functions)

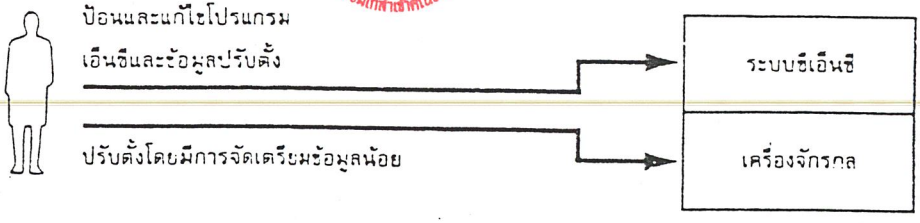
ในปัจจุบัน ระบบควบคุมการทำงานของเครื่องจักรกลสมัยใหม่เกือบทั้งหมดจะควบคุมด้วยระบบซีเอ็นซี แต่เนื่องจากยังคงอ้างอิงถึงโปรแกรมเอ็นซี (NC Program) และเทคโนโลยีเอ็นซี (NC Technology) อยู่ ดังนั้น จึงเป็นเรื่องสำคัญที่จะต้องรู้ถึงความแตกต่างในการทำงานระหว่างระบบเอ็นซีกับระบบซีเอ็นซี

ระบบเอ็นซี (NC System) ดังแสดงในรูป 2.18 จะมีระบบควบคุมประกอบอยู่กับเครื่องจักรกลซึ่งจะต้องจัดเตรียมโปรแกรมเอ็นซีจากภายนอกก่อน แล้วจึงป้อนเข้าไปในระบบควบคุม โดยอาศัยสื่อข้อมูล (data carries) เช่น เทปกระดาษ (Punched tape)เป็นต้น โปรแกรมเอ็นซีที่ป้อนเข้าไปในระบบควบคุมของเครื่องจะถูกนำไปใช้เพื่อสั่งให้เครื่องเริ่มทำงานและหยุดชั่วคราวได้ แต่จะไม่สามารถแก้ไขโปรแกรมโดยช่างควบคุมเครื่องได้ ขนาดของเครื่องมือและอุปกรณ์จับยึดชิ้นงานจะถูกเลือกใช้ในขณะเขียนโปรแกรมไว้ก่อน และกำหนดไว้ในใบปรับตั้ง (set-up sheet) ซึ่งช่างควบคุมเครื่องจะต้องจัดเตรียมและประกอบยึดเครื่องมือ ตลอดจนอุปกรณ์จับยึดชิ้นงานให้ถูกต้องตามข้อมูลที่กำหนดไว้ในใบปรับตั้ง



รูปที่ 2.18 ระบบ NC

ระบบซีเอ็นซี (CNC system) จะมีคอมพิวเตอร์ประกอบอยู่ด้วยดังนั้น ช่างควบคุมเครื่องไม่เพียงแต่จะสามารถใช้โปรแกรมเอ็นซีสั่งให้เครื่องจักรทำงานได้เท่านั้น แต่จะยังสามารถเขียนและป้อนโปรแกรมด้วยตนเอง ตลอดจนการแก้ไขโปรแกรมได้หลังจากป้อนเข้าไปในระบบควบคุมของเครื่องแล้ว ดังแสดงในรูป 2.19



รูปที่ 2.19 ระบบ CNC

ขนาดต่างๆของเครื่องมือตัดและอุปกรณ์จับยึดชิ้นงานสามารถที่จะเลือกใช้และป้อนเข้าไปในระบบควบคุมซีเอ็นซี ขณะทำการปรับตั้ง (setting-up) และเป็นอิสระจากตัวโปรแกรมเอ็นซี ขนาดต่างๆ ของเครื่องมือจะถูกนำไปใช้โดยอัตโนมัติในขณะทำการตัดเฉือน ด้วยเหตุนี้ช่างควบคุมเครื่องจึงไม่จำเป็นต้องมีข้อมูลในการปรับตั้งมากและสามารถที่จะเลือกใช้เครื่องมือและอุปกรณ์จับยึดชิ้นงานได้ด้วยตนเอง

หากพิจารณาถึงภาษาโปรแกรม (Programming language) และเทคโนโลยีทางด้านการตัดเฉือนของเครื่องจักรกลที่ใช้ในระบบเอ็นซีกับซีเอ็นซีแล้วจะไม่แตกต่างกัน

2.4.2 ชนิดของการควบคุม (Control modes)

ลักษณะการควบคุมการเคลื่อนที่ทำงานของแท่นเลื่อนต่าง ๆ ในเครื่องจักรกล เอ็นซีและซีเอ็นซี จะมีการเคลื่อนที่อยู่ 2 ลักษณะ คือ

การเคลื่อนที่ในแนวเส้นตรง (Linear Interpolation หรือ straightlineInter polation) การเคลื่อนที่ลักษณะนี้ ระบบซีเอ็นซีจะคำนวณหาตำแหน่งของจุดต่างๆที่ต่อกันเป็นลูกโซ่ในแนวเส้นตรงระหว่างตำแหน่งของเครื่องมือ 2 ตำแหน่งในขณะที่เครื่องมือเคลื่อนที่จากจุดหนึ่งไปยังอีกจุดหนึ่งนั้น ระบบควบคุมซีเอ็นซีจะตรวจสอบและแก้ไขแนวแกนในการเคลื่อนที่ให้ถูกต้องอยู่ตลอดเวลาทำให้การเคลื่อนที่ของเครื่องมือ ไม่ผิดพลาดหรือคลาดเคลื่อนออกจากจุดต่อของเส้นตรงมากกว่าค่าพิกัดความเผื่อของเครื่องที่กำหนดไว้

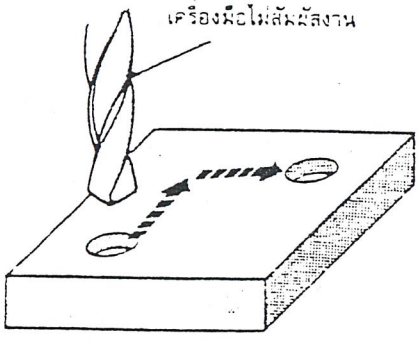
การเคลื่อนที่ในแนวเส้นโค้ง (Circular Interpolation) ระบบควบคุมซีเอ็นซี จะคำนวณหาตำแหน่งของจุดต่างๆ ที่ต่อกันเป็นเส้นโค้งตามขนาดรัศมีที่กำหนด ระหว่างตำแหน่งของเครื่องมือที่กำหนดไว้ 2 ตำแหน่ง ระบบควบคุมจะอาศัยจุดเหล่านี้ในการตรวจสอบและแก้ไขแนวการเคลื่อนที่ของเครื่องมือให้ถูกต้องและอยู่ภายใน พิกัดความเผื่อของเครื่องจักรกลที่กำหนด

ในระบบควบคุมซีเอ็นซีจะแบ่งการควบคุมการเคลื่อนที่ทั้งสองลักษณะตามลักษณะการเคลื่อนที่ป้อนออกเป็น 3 ชนิด คือ

1) การควบคุมจุดต่อจุด (Point to point control)

การควบคุมแบบนี้จะควบคุมการเคลื่อนที่ของเครื่องมือระหว่างจุดสองจุดที่โปรแกรมไว้ในลักษณะการเคลื่อนที่เร็ว (Rapid traverse) โดยที่เครื่องมือจะต้องไม่สัมผัสชิ้นงาน ดังแสดงในรูป 2.20

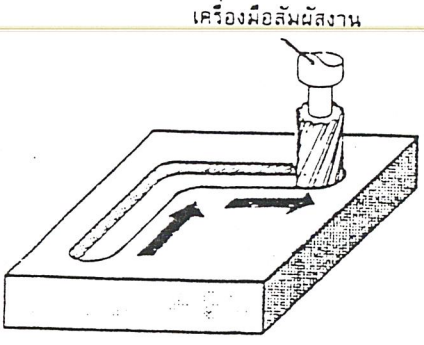
แนวแกนในการเคลื่อนที่ขึ้นอยู่กับชนิดของระบบควบคุม กล่าวคือมอเตอร์ขับของระบบป้อนอาจจะเริ่มทำงานหลายๆ แนวแกนพร้อมกัน หรือทำงานทีละแนวแกนจนกว่าจะเคลื่อนที่ถึงตำแหน่งของเครื่องมือที่โปรแกรมไว้ ทำให้ไม่สามารถควบคุมทางเดินของเครื่องมือ (Tool path) ได้ การควบคุมแบบจุดต่อจุดมักจะใช้กับเครื่องเจาะ (drilling machine) เครื่องเชื่อมจุด(spot drilling) เป็นต้น



รูปที่ 2.20 การควบคุมจุดต่อจุด

2. การควบคุมการตัดเฉือนแนวเส้นตรง (Straight-cut control)

การควบคุมชนิดนี้ นอกจากจะสามารถควบคุมการเคลื่อนที่ของเครื่องมือแบบเคลื่อนที่เร็วได้แล้ว ยังสามารถควบคุมการเคลื่อนที่ของเครื่องมือในแนวขนานกับแนวแกนของเครื่องจักรกล ตามค่าอัตราป้อนที่ต้องการได้อีกด้วย แต่จะสามารถควบคุม การเคลื่อนที่ได้ครั้งละ 1 แนวแกนเท่านั้น การเคลื่อนที่ของเครื่องมือจะถูกควบคุมด้วย อัตราป้อนและความยาวในการเคลื่อนที่ ดังแสดงในรูป 2.21 ระบบการควบคุมการตัดเฉือนแนวเส้นตรงชนิดนี้จะใช้กับเครื่องกัดและเครื่องกลึงแบบง่ายๆ

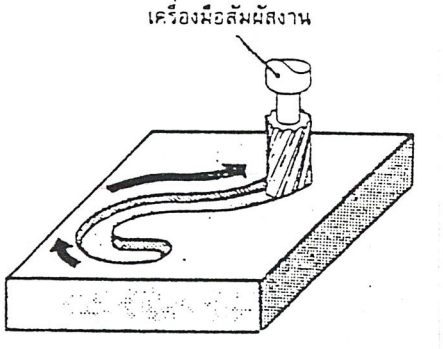


รูปที่ 2.21 การควบคุมแบบเส้นตรง

3) การควบคุมตามเส้นขอบรูป (Contouring control)

การควบคุมแบบนี้จะสามารถควบคุมการเคลื่อนที่ทำงานได้ดังนี้

- ควบคุมเครื่องมือให้เคลื่อนที่ไปยังตำแหน่งที่ต้องการแบบเคลื่อนที่เร็วได้
- ควบคุมเครื่องมือให้เคลื่อนที่ขนานกับแนวแกนไปยังตำแหน่งที่ต้องการตามค่าอัตรา ป้อนได้
- ควบคุมเครื่องมือให้เคลื่อนที่ไปยังตำแหน่งใดๆ บนชิ้นงานที่กำหนดในแนวเส้นตรงและ เส้นโค้งตามค่าอัตราป้อนได้



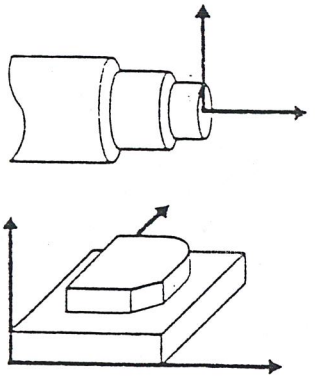
รูปที่ 2.22 การควบคุมตามเส้นขอบรูป

การควบคุมตามเส้นขอบรูปนี้ ยังสามารถแยกย่อยได้อีกเป็น 3 ระดับ ซึ่งขึ้นอยู่กับความสามารถของระบบควบคุมคือ ความสามารถในการควบคุมการเคลื่อนที่ของเครื่องมือได้ 2 หรือ 3 แกน พร้อมๆ กัน โดยไม่คำนึงถึงว่าเครื่องจักรกลซีเอ็นซีนั้นๆ จะมีกี่แนวแกน จุดสำคัญอยู่

ที่ว่าจะสามารถควบคุมแนวแกนป้อนได้พร้อมๆกันกี่แนวแกน ระดับความสามารถทั้งสามระดับของระบบควบคุมมีรายละเอียดดังนี้

ก) การควบคุมตามเส้นขอบรูปแบบ 2 แกน (2D Contouring control) ระบบควบคุมจะสามารถควบคุมเครื่องมือให้เคลื่อนที่ในระนาบ (plane) ที่กำหนดเฉพาะได้ 2 แนวแกนพร้อมๆ กัน ทำให้สามารถเคลื่อนที่ได้ทั้งในแนวเส้นตรงและเส้นโค้ง แต่จะไม่สามารถเปลี่ยนระนาบในการทำงานได้ นั่นหมายความว่า 2 แนวแกนที่เคลื่อนที่พร้อมกันได้นั้น จะถูกกำหนดตายตัวจากบริษัทผู้ผลิตไม่สามารถเปลี่ยนแนวแกนได้

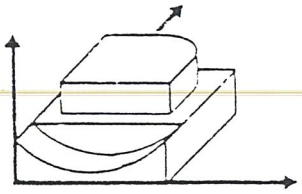
ถ้าเครื่องจักรกลซีเอ็นซีนั้นมี 3 แนวแกน และระบบควบคุมเป็นแบบการควบคุมตามเส้นขอบรูป 2 แกนแล้ว แนวแกนที่ 3 จะถูกควบคุมเป็นอิสระจาก 2 แนวแกนข้างต้น เช่น เครื่องกัด (Milling Machine) จะใช้แนวแกนหนึ่งสำหรับการเคลื่อนที่ป้อนกินลึก ส่วนอีก 2 แนวแกนจะใช้สำหรับการเดินกัดตามเส้นขอบรูป เป็นต้น ดังแสดงในรูป 2.23



รูป 2.23 การควบคุมตามเส้นขอบรูปแบบ 2 แกน

ข) การควบคุมตามเส้นขอบรูปแบบ 2 แกนครึ่ง ($2^{1}/_{2}$-DContouring control) ระบบควบคุมแบบนี้จะควบคุมเครื่องมือให้เคลื่อนที่ในแนวเส้นตรงและเส้นโค้งบนระนาบใดๆ ที่ต้องการก็ได้ แต่จะสามารถควบคุมการเคลื่อนที่ได้พียง 2 แนวแกนพร้อมกันเท่านั้น

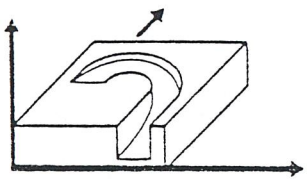
สำหรับเครื่องจักรกลซีเอ็นซีที่มี 3 แนวแกนคือ X,Y และ Z แนวแกนคู่ใดคู่หนึ่ง คือ X/Y หรือ Y/Z หรือ X/Z จะสามารถควบคุมให้เคลื่อนที่พร้อมกันได้นั่นหมายความว่า สำหรับเครื่องกัด การเคลื่อนที่ป้อนกินลึก (In-feed)สามารถทำในแนวแกนใดๆ ได้ทั้ง 3 แนวแกน ส่วนอีก 2 แนวแกนที่เหลือจะใช้สำหรับเดินกัดตามเส้นขอบรูป ดังแสดงในรูป 2.24



รูปที่ 2.24 การควบคุมเส้นขอบรูปแบบ 2 แกนครึ่ง

ค) การควบคุมตามเส้นขอบรูปแบบ 3 แกน (3-D Contouring control)

ระบบควบคุมจะสามารถควบคุมเครื่องมือให้เคลื่อนที่ในแนวเส้นตรงและเส้นโค้งได้พร้อมกันทั้ง 3 แนวแกน เป็นลักษณะ 3 มิติได้ ดังแสดงในรูปที่ 2.25



รูปที่ 2.25 การควบคุมเส้นขอบรูปแบบ 3 แกน

การควบคุมตามเส้นขอบรูปจะสามารถใช้เป็นการควบคุมการตัดเฉือนแนวเส้นตรงได้และการควบคุมการตัดเฉือนแนวเส้นตรงจะใช้เป็นการควบคุมแบบจุดต่อจุดได้ แต่ในทางกลับกันจะกระทำไม่ได้

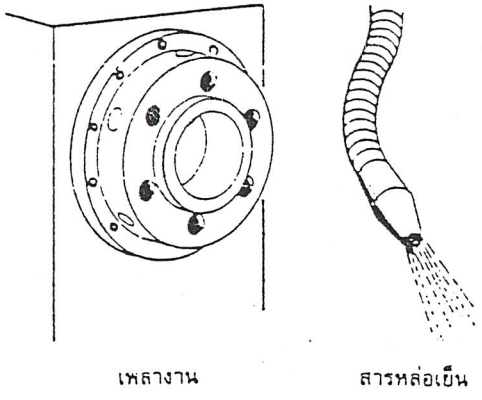
2.4.3 การควบคุมหน้าที่การทำงานของเครื่องจักรกล(Control of machine function)

กระบบควบคุมซีเอ็นซีนอกจากจะสามารถควบคุมการเคลื่อนที่ของเครื่องมือ ตามรูปทรงเรขาคณิตของชิ้นงานแล้ว ยังสามารถควบคุมหน้าที่การทำงานอื่นๆที่ช่วยเสริมการทำงานตัดเฉือนของเครื่องจักรกลให้เหมาะสมกับสภาวะการทำงานในขณะนั้นได้อีกด้วยจำนวนหน้าที่การทำงานและวิธีการควบคุมจะไม่ขึ้นอยู่กับตัวเครื่องจักรกลเพียงอย่างเดียว แต่ยังขึ้นอยู่กับชนิดระบบควบคุมอีกด้วย

ตัวอย่างหน้าที่การทำงานต่างๆ ที่จำเป็นจะต้องโปรแกรมเพื่อช่วยในการทำงานดังแสดงในรูป 2.26 มีดังนี้

- การเริ่มหมุนของเพลางาน ทิศทางการหมุนและการเปลี่ยนความเร็วรอบ
- การกำหนดตำแหน่งของเพลางาน

- การเปิดสารหล่อเย็น และความดันของสารหล่อเย็น
- การรักษาอัตราป้อนให้คงที่
- การเปลี่ยนตำแหน่งของเครื่องมือ
- การรักษาความเร็วตัดให้คงที่
- การเริ่มทำงานหรือควบคุมการทำงานของอุปกรณ์ช่วยงานอื่นๆ เช่น อุปกรณ์เปลี่ยนชิ้นงาน ได้แก่ โต๊ะเปลี่ยนงาน (pallet shuttle) เป็นต้น
- ชุดยันศูนย์ท้ายแท่น (Tail-stock)
- อุปกรณ์ใส่และถอดชิ้นงาน (loader and unloader)
- แท่นประคองศูนย์ (Steady rest)
- อุปกรณ์ลำเลียงเศษ (chip conveyor)
- Sorter

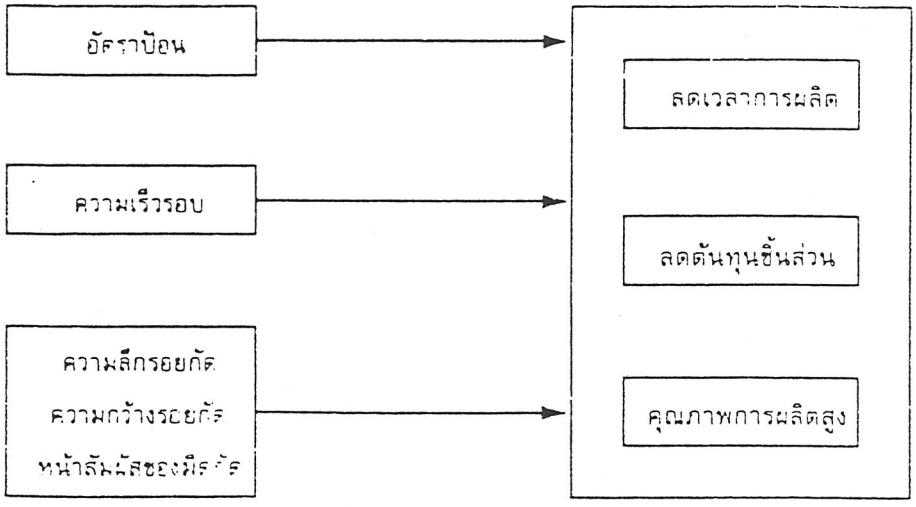


รูปที่ 2.26 หน้าที่การทำงานของเครื่องจักรกล

เครื่องจักรกลซีเอ็นซีที่สามารถใช้ระบบควบคุมสั่งการทำงานในหน้าที่ต่างๆ ได้ยิ่งมากเท่าใดก็จะเป็นระบบที่มีความเหมาะสมสำหรับการควบคุมแบบอัตโนมัติมากยิ่งขึ้น

2.5 การตัดเฉือนโลหะด้วยเครื่องจักรกลซีเอ็นซี

2.5.1 ข้อมูลการตัดเฉือนโลหะสำหรับงานกัด



รูปที่ 2.27 ความสัมพันธ์ของข้อมูลการตัดเฉือนโลหะสำหรับงานกัด

ข้อมูลที่ช่างเขียนโปรแกรมจะต้องจัดเตรียมสำหรับการทำงานกัดได้แก่ ความเร็วรอบของเพลามีดกัด อัตราป้อน ความลึกหรือความกว้างรอยกัดและหน้าสัมผัสของมีดกัด แฟคเตอร์เหล่านี้จะต้องนำมาพิจารณาร่วมกันเพื่อให้บรรลุเป้าหมาย 3 ประการ คือ

เป้าหมายที่ 1 : ลดเวลาการผลิต (Short cycle time)

แฟคเตอร์สำคัญเกี่ยวกับเวลาการผลิต (Cycle time) ที่สามารถควบคุมได้ โดยช่างเขียนโปรแกรมเอ็นซี ได้แก่ ปริมาตรการตัดเฉือนเนื้อโลหะออกต่อนาที (Stock removal rate per minute) ซึ่งเป็นผลคูณของค่าอัตราป้อนกับหน้าสัมผัสของมีดกัด (Cut engagement) กับความลึกหรือความกว้างรอยกัด ค่าแฟคเตอร์ค่าหนึ่งค่าใดใน 3 ค่านี้ หากมีค่าใดสูงขึ้นก็จะเป็นผลให้ปริมาตรการตัดเฉือนเนื้อโลหะออกมีอัตราที่สูงขึ้นก็จะเป็นผลให้เครื่องมือตัดสึกหรอเร็วขึ้นด้วย ทำให้ค่าเฉลี่ยของเวลาการผลิตสูงขึ้นด้วย เนื่องจากการเปลี่ยนเครื่องมือตัด หรือเปลี่ยนคมตัด

เป้าหมายที่ 2 : ลดต้นทุนชิ้นส่วน (Lower part costs)

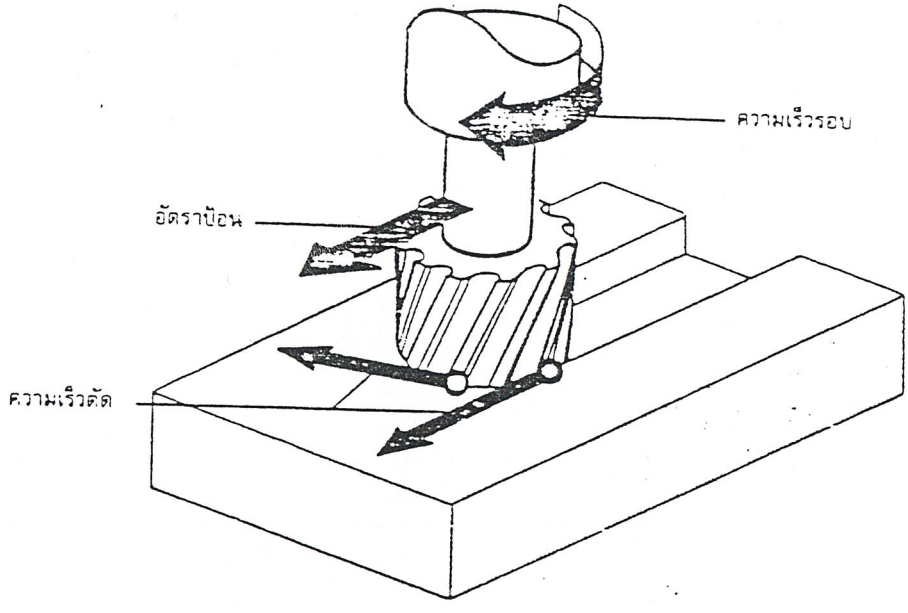
การเพิ่มขึ้นของค่าข้อมูลการตัดเฉือนใดๆ ก็ตามจะมีผลทำให้เวลาการผลิตต่อชิ้นลดลง ซึ่งจะเป็นการช่วยลดค่าแรงงานและค่าเครื่องจักรลงด้วยแต่ค่าเครื่องมือตัดจะสูงขึ้นเนื่องจากมีการสึกหรอสูง ดังนั้นจึงไม่ควรเลือกใช้ค่าข้อมูลการตัดเฉือนที่จะมีผลทำ ให้ค่าเครื่องมือตัดที่เกิดจากการสึกหรอสูงเกินระดับหนึ่ง ซึ่งอาจพิจารณาเลือกใช้สารหล่อเย็นเพื่อเพิ่มอายุคมมีดให้ยาวขึ้นด้วยก็ได้

เป้าหมายที่ 3 : คุณภาพการผลิตสูง (High production quality)

การเลือกใช้ข้อมูลการตัดเฉือนจะถูกจำกัดด้วยผลิตภัณฑ์ที่ต้องการคุณภาพสูง ซึ่งจะเกี่ยวข้องกับผิวสำเร็จและพิกัดความเผื่อของขนาดชิ้นงานสำเร็จ การเลือกใช้ข้อมูลการตัดเฉือนจะต้องพิจารณาให้สัมพันธ์กันกับข้อมูลอื่นๆ ได้แก่

- หมวดงานกัด (Milling mode) เช่น งานกัดตาม งานกัดทวน งานกัดปาดหน้า เป็นต้น
- รูปทรงของมีดกัด
- ชนิดของขอบคมตัดที่ใช้ เช่น รูปทรงของขอบคมตัด วัสดุมีดกัด เป็นต้น
- ภาระงานของเครื่องจักร เช่น ความสามารถในการรับความเค้น (Stressability)
- คุณสมบัติการสั่นสะเทือนของเครื่องจักร เครื่องมือตัด และวัสดุงาน

2.5.2 ความเร็วรอบและอัตราป้อน



รูปที่ 2.28 ความเร็วรอบ ความเร็วตัด และอัตราป้อน

1. ความเร็วรอบของมีดกัด

สามารถป้อนค่าในโปรแกรมเอ็นซีเป็นค่าความเร็วรอบโดยตรงหน่วยเป็น จำนวนรอบต่อนาที (r.p.m.)

ตัวอย่าง : S = 630 r.p.m. หมายถึง ค่าความเร็วรอบ 630 รอบต่อนาที

ค่าความเร็วรอบที่เลือกใช้จะเป็นตัวกำหนดค่าความเร็วตัดในงานกัดด้วย (ดังรูปที่ 2.28) ความเร็วตัดจะมีค่าเท่ากับความเร็วขอบของมีดกัด การเปลี่ยนแปลงของค่าความเร็วตัดนี้จะไม่ขึ้นอยู่กับค่าความเร็วรอบเพียงอย่างเดียว แต่ยังขึ้นอยู่กับขนาดเส้น ผ่านศูนย์กลางของมีดกัดด้วย กล่าวคือ ถ้าความเร็วรอบมีค่าสูงและมีดกัดมีขนาดเส้นผ่านศูนย์กลางโตจะเป็นผลให้ความเร็วตัดมีค่าสูงด้วย

ในการป้อนค่าความเร็วรอบ จะต้องมั่นใจว่าได้กำหนดทิศทางการหมุนของเพลามีดกัดไว้ถูกต้องด้วย

2. อัตราป้อน

อัตราป้อนเป็นการเคลื่อนที่ของมีดกัดในทิศทางทำงานตัดเฉือน (ดูรูปที่2.29) โดยทั่วไปอัตราป้อนจะกำหนดเป็นระยะทางการเคลื่อนที่ต่อนาที นอกจากนี้อาจกำหนดอัตราป้อนเป็นระยะทางการเคลื่อนที่ต่อรอบการหมุนของมีดกัด หรือต่อฟันมีดก็ได้

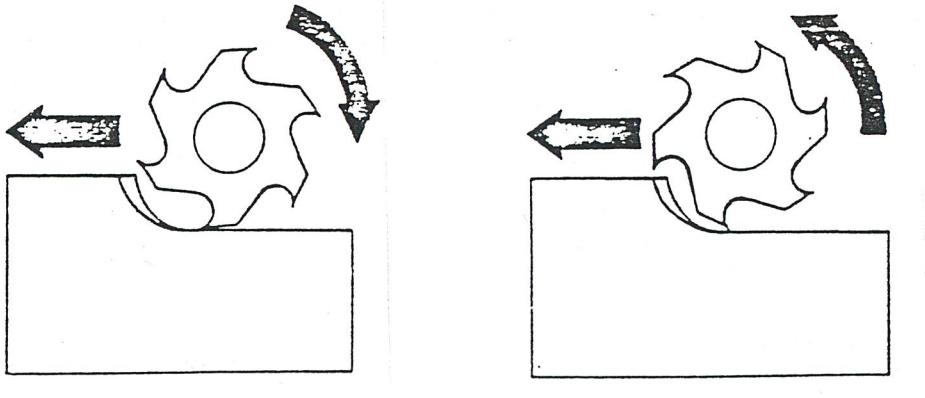
ตัวอย่าง : F = 100 mm/min หมายถึง ระยะทางการเคลื่อนที่ของมีดกัด 100 มม.ในเวลา 1 นาที

F = 0.1 mm/rev หมายถึง ระยะทางการเคลื่อนที่ของมีดกัด 0.1 มม .เมื่อมีดกัดหมุนครบ 1 รอบ

F = 0.02 mm/T หมายถึง ระยะทางการเคลื่อนที่ของมีดกัด 0.02 มม .ต่อ 1 ฟันมีดกัด

โดยทั่วไป การเคลื่อนที่ป้อนในงานกัด จะเกิดจากการเคลื่อนที่ของโต๊ะงานอย่างต่อเนื่องสม่ำเสมอ และการเคลื่อนที่หมุนของมีดกัด แต่ในการเขียนโปรแกรมเอ็นซี สำหรับงานกัดจะโปรแกรมในลักษณะที่สมมุติให้โต๊ะงานอยู่กับที่และมีดกัดเคลื่อนที่ (ตามหลักความสัมพันธ์ในการเคลื่อนที่)

เมื่อกำหนดให้ความเร็วรอบของมีดกัดมีค่าคงที่ การเลือกใช้อัตราป้อนจะมีผลต่อความหนาของเศษและผิวสำเร็จของชิ้นงาน



ก) งานกัดตาม ข) งานกัดทวน

รูปที่ 2.29 ลักษณะการเคลื่อนที่กัด

การเลือกใช้ลักษณะงานกัดระหว่างงานกัดตามกับงานกัดทวนจะมีผลกระทบ ต่อการเปลี่ยนรูปของเศษ(chip formation)และความดันตัดเฉือน (cutting pressure)

ก) งานกัดตาม (Conventional milling) ในงานกัดตาม ความหนาของเศษและความดันในการตัดเฉือนจะเพิ่มขึ้นเรื่อยๆ ที่ฟันมีดกัด และจะมีค่าสูงสุดก่อนที่ฟันมีดกัดจะเลื่อนพ้นวัสดุงานเล็กน้อย เมื่อฟันมีดกัดเลื่อนพ้นวัสดุงานแล้ว จะเกิดสภาวะที่ติดตามมาคือ ความดันตัดเฉือนจะหมดไปทันที ทำให้มีดกัดเคลื่อนที่ไปข้างหน้าโดยเร็ว และฟันมีดกัดถัดไปจะเลื่อนเข้ากัดวัสดุงานในลักษณะการกระตุก (jerking) เป็นผลให้เกิด เป็นรอยสั่น (chatter marks) ขึ้นที่ผิวงาน

ข) งานกัดทวน (Climb milling) ในงานกัดทวน ลักษณะการเกิดเศษจะกลับกันกับงานกัดตาม กล่าวคือ เมื่อฟันมีดกัดเริ่มเข้าตัดเฉือนชิ้นงาน ความหนาของเศษและความดันตัดเฉือนจะมีค่าสูงสุด และเมื่อฟันมีดกัดเลื่อนออกจากวัสดุงาน เศษจะมีขนาด บางที่สุดและความดันตัดเฉือนมีค่าน้อยสุด ดังนั้น จึงทำให้เกิดรอยสั่นสะเทือนน้อยและชิ้นงานมีผิวสำเร็จที่ดีกว่าเมื่อเปรียบเทียบกับงานกัดตาม งานกัดทวนจะใช้เครื่องกัดที่มี กำลังน้อยกว่าได้ แต่ต้องการความแข็งแรงของเครื่องกัดมากกว่า และมีโต๊ะงานที่ปราศจากระยะคลอน (backlash)

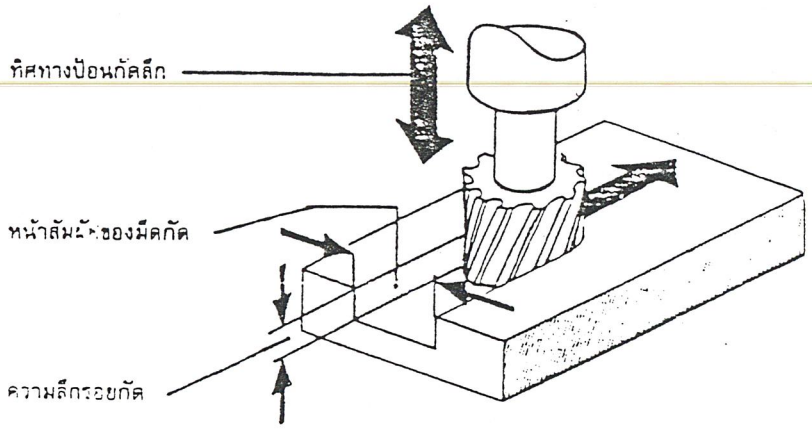
เนื่องจากเครื่องจักรกลซีเอ็นซีจะมีคุณสมบัติเหล่านี้อยู่แล้ว ดังนั้น จึงมักนิยมเลือกใช้งานกัดทวนมากกว่า

2.5.3 ความลึกหรือความกว้างรอยกัดและหน้าสัมผัสของมีดกัด

1. ความลึกหรือความกว้างรอยกัด (Depth or width of cut)

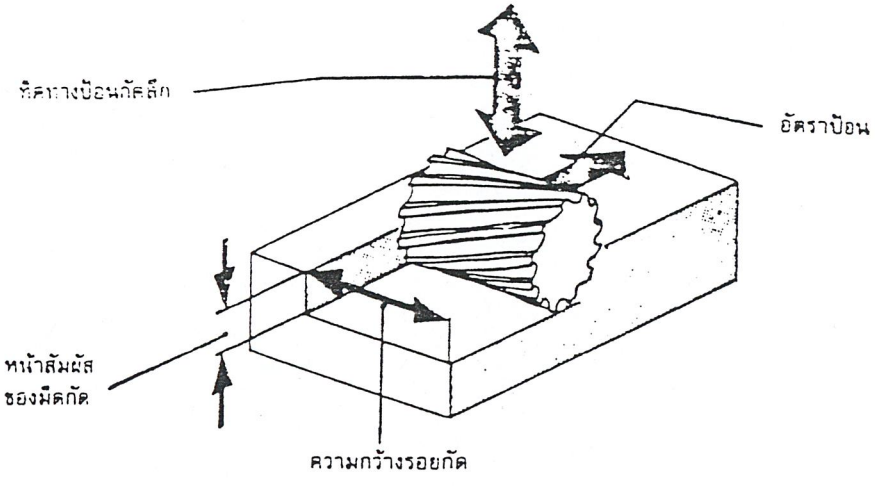
ความลึกหรือความกว้างรอยกัด หมายถึง ระยะทางที่มีดกัดจมลึกเข้าไปในผิวงานในทิศทางป้อนกัด (Infeed direction)

ก) ความลึกรอยกัด (Depth of cut) จะเรียกใช้สำหรับงานกัดด้วยเครื่องกัดเพลาตั้ง เช่น งานกัดปาดหน้า (face milling) เป็นต้น ดังรูปที่ 2.30



รูปที่ 2.30 ความลึกของรอยกัดและหน้าสัมผัสของมีดกัดในงานกัดปาดหน้า

ข) ความกว้างรอยกัด (Width of cut) จะเรียกใช้สำหรับงานกัดด้วยเครื่องกัดเพลานอน เช่น งานกัดราบ (peripheral or plain or slab milling) เป็นต้น ดังรูปที่ 2.31



รูปที่ 2.31 ความกว้างของรอยกัดและหน้าสัมผัสของมีดกัดในงานกัดราบ

2. หน้าสัมผัสของมีดกัด (Cutter engagement)

หน้าสัมผัสของมีดกัด คือ ความกว้างของมีดกัดที่สัมผัสอยู่กับชิ้นงาน โดยวัดในระนาบการทำงาน (working plane) ในทิศทางที่ตั้งฉากกับอัตราป้อน (ดูรูปที่ 2.30 และ 2.31) ความลึกหรือความกว้างของรอยกัด และหน้าสัมผัสของมีดกัด จะเป็นผลมาจาก

ก) การโปรแกรมการเคลื่อนที่ของมีดกัด และ

ข) ขนาดและรูปทรงของมีดกัด

มีดกัดที่เลือกใช้จะต้องมีขนาดไม่ยาวเกินความจำเป็นในการทำงานกัดนั้นๆมีดกัดยิ่งมีขนาดยาว จะทำให้เกิดการเปลี่ยนแปลงของขนาดชิ้นงานมากขึ้น ทั้งนี้เนื่องจาก การโก่งงอของด้ามมีดกัด

ในการเขียนโปรแกรมเส้นทางเดินของมีดกัด (cutter path) ในชิ้นงานจะต้องพิจารณาเลือกใช้ความลึกหรือความกว้างรอยกัดและหน้าสัมผัสของมีดกัดที่สัมพันธ์กับข้อมูลอื่นๆด้วย ได้แก่

- ความเร็วในการตัดเฉือนของมีดกัดที่ใช้ ที่เป็นไปได้ กับวัสดุชิ้นงานที่กัด
- ขนาดผิวสำเร็จที่ต้องการ

2.6 เรขาคณิตเบื้องต้นสำหรับการทำโปรแกรมเอ็นซี

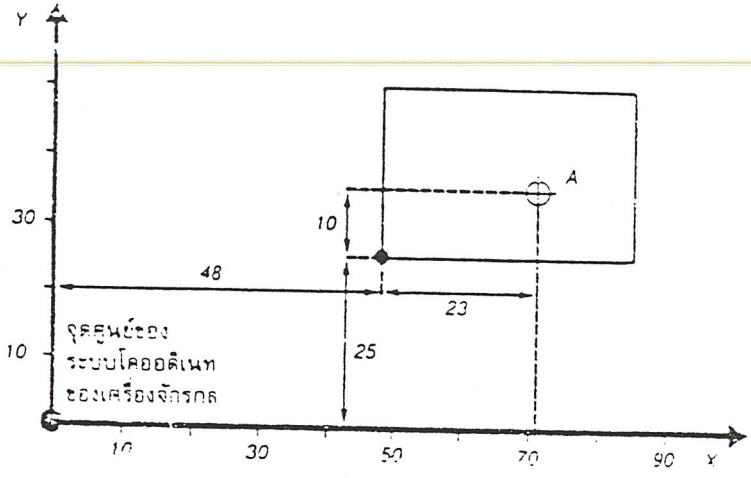
2.6.1 จุดศูนย์และจุดอ้างอิง (Zero points and Reference points)

การเคลื่อนที่ต่างๆ ของเครื่องจักรกลซีเอ็นซี จะถูกควบคุมด้วยระบบโคออดิเนท ตำแหน่งต่างๆที่ถูกต้องภายในพื้นที่ทำงานของเครื่องจักรกลจะวัดระยะมาจากจุดศูนย์ (Zero points) นอกจากจุดศูนย์แล้ว เครื่องจักรกลซีเอ็นซียังมีจุดอ้างอิง (Reference points)อื่นๆอีก เพื่อช่วยเสริมการทำงานและการทำโปรแกรม

1) จุดศูนย์ของเครื่อง (Machine zero point , M)

การจับยึดชิ้นงานเข้ากับเครื่องจักรกลซีเอ็นซีจะต้องให้สัมพันธ์กันระหว่างขนาดที่กำหนดในแบบงานกับระบบโคออดิเนทของเครื่องเพื่อให้สามารถเปรียบเทียบขนาดซึ่งกันและกันได้ เครื่องจักรกลซีเอ็นซีทุกเครื่องจะมีระบบโคออดิเนทประกอบอยู่ ระบบนี้จะกำหนดจากการเคลื่อนที่กับระบบวัดระยะการเคลื่อนที่ของเครื่องจักรกลที่มีอยู่

รูป 2.32 แสดงให้เห็นแบบชิ้นงานที่วางอยู่บนระบบโคออดิเนทของเครื่องจักรกล รู A ที่มีขนาดระยะกำหนดในแบบงาน คือ 23 และ 10 มม. จะมีค่าโคออดิเนทที่สัมพันธ์กับระบบโคออดิเนทของเครื่องจักรกล คือ โคออดิเนท X = 71 และ Y = 35 มม.



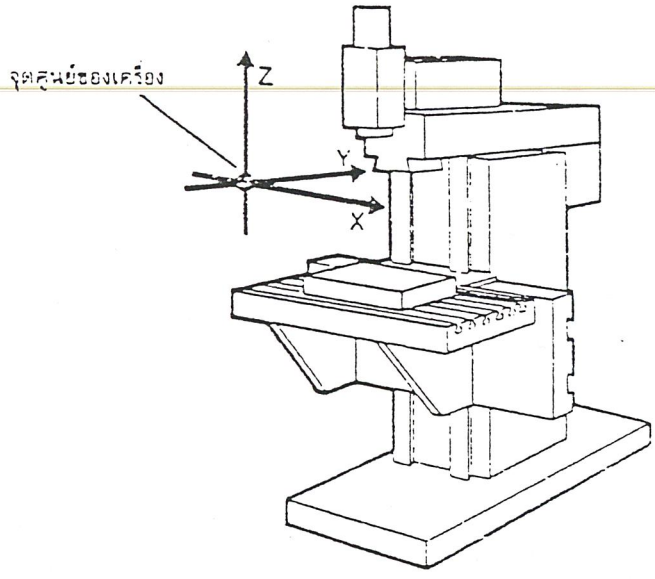
รูปที่ 2.32 แบบชิ้นงานในระบบโคออดิเนทของเครื่องจักรกล CNC

จุดเริ่มต้นของการกำหนดขนาดในแบบงาน (มุมซ้ายมือด้านล่าง) จะมีระยะเยื้องศูนย์จากระบบโคออดิเนทของเครื่องจักรกลซีเอ็นซี คือ X = 48 และ Y = 25 มม.

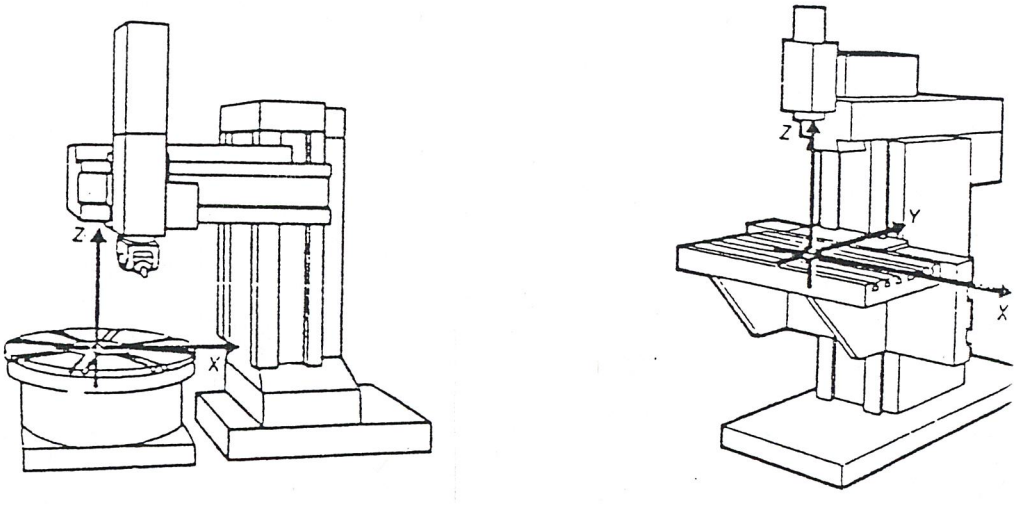
จุดศูนย์ของเครื่อง (M) โดยทั่วไปจะใช้แทนด้วยสัญลักษณ์ ตำแหน่งจุดศูนย์ของเครื่องจะถูกกำหนดโดยบริษัทผู้ผลิตเครื่องจักรกลซีเอ็นซี จุดศูนย์ของเครื่องจะใช้เป็นจุดศูนย์ของระบบโคออดิเนทของเครื่องจักรกล และใช้เป็นจุดเริ่มต้นสำหรับระบบโคออดิเนทอื่นๆ และยังใช้เป็นจุดอ้างอิงในเครื่องจักรกลด้วย

สำหรับเครื่องกัดจะมีตำแหน่งจุดศูนย์ของเครื่องที่แตกต่างกัน ซึ่งขึ้นอยู่กับบริษัทผู้ผลิตเครื่องกัด ดังนั้น ตำแหน่งจุดศูนย์ของเครื่องและทิศทางของแนวแกนที่ถูกต้อง จึงต้องศึกษาจากคู่มือการปฏิบัติงานที่จัดเตรียมโดยบริษัทผู้ผลิตแต่ละบริษัท

ตัวอย่างตำแหน่งจุดศูนย์ของเครื่องของเครื่องจักรกลซีเอ็นซีอื่นๆ ที่เป็นไปได้ พร้อมด้วยระบบโคออดิเนทของเครื่อง ได้แสดงไว้ในรูปที่ 2.34 สำหรับเครื่องกัด ตำแหน่งจุดศูนย์ของเครื่องอาจอยู่ที่จุดศูนย์กลางของโต๊ะงาน หรือที่จุดตรงขอบของช่วงที่มีการเคลื่อนที่ หรือแท่นเลื่อนก็ได้



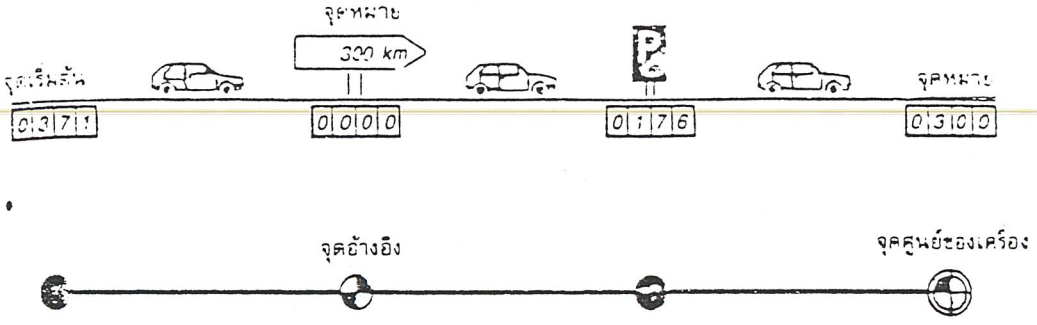
รูปที่ 2.33 ตำแหน่งจุดศูนย์ของเครื่องกัด



รูปที่ 2.34 ตัวอย่างตำแหน่งจุดศูนย์ของเครื่องจักรกล CNC อื่นๆ

2) จุดอ้างอิง (Reference point , R)

วัตถุประสงค์เบื้องต้นของการใช้จุดอ้างอิง สามารถอธิบายให้เข้าใจได้ง่ายโดยเปรียบเทียบกับการใช้หลักกิโลเมตรตามเส้นทางหลวงต่างๆ ดังแสดงใน รูป 2.35



รูปที่ 2.35 วัตถุประสงค์ของการใช้จุดอ้างอิง

จากรูปที่ 2.35 สมมุติว่าขณะที่ท่านขับรถยนต์อยู่ มีเหตุการณ์ต่างๆ เกิดขึ้นตามลำดับขั้นตอนดังต่อไปนี้

1. ณ จุดเริ่มต้นการเดินทาง ท่านจะยังไม่ทราบว่า จุดหมายที่จะไปนั้นอยู่ห่างไกลออกไปเป็นระยะทางเท่าใด

2. ตลอดเส้นทาง ท่านจะคอยสังเกตดูหลักกิโลเมตร (จุดอ้างอิง) ซึ่งจะบอกว่าจุดหมายที่จะไปนั้นอยู่ห่างออกไปเท่าไร ท่านก็จะปรับตั้งมิเตอร์วัดระยะทางให้ตรงขีดศูนย์

3. จากนี้ไปท่านก็จะสามารถบอกได้ว่า ณ จุดต่างๆ ที่ท่านไปถึงนั้นอยู่ห่างจากจุดหมายของท่านเท่าไร หรืออยู่ห่างจากจุดอ้างอิงเท่าไร (หลักกิโลเมตร)

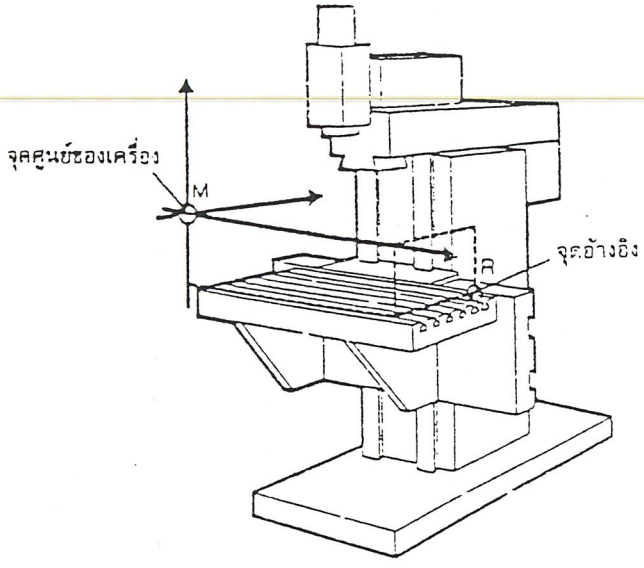
ต่างๆ ดังนี้จากรูปที่ 2.35 จะมีสถานีอยู่ 3 แห่งด้วยกัน ซึ่งเมื่อเปรียบเทียบกับเครื่องจักรกลซีเอ็นซีแล้ว จะมีความหมาย

สถานีที่ 1 หมายถึง การเปิดสวิตช์เครื่อง

สถานีที่2 หมายถึง การเลื่อนไปยังจุดอ้างอิงและปรับตั้งระบบวัดระยะเลื่อนให้เริ่มต้นที่ศูนย์

สถานีที่ 3 หมายถึง ตำแหน่งของเครื่องมือที่เคลื่อนที่ไปอย่างต่อเนื่อง

จุดอ้างอิง (R) มักจะใช้แทนด้วยสัญลักษณ์เป็นจุดที่ใช้ช่วยในการปรับค่าและควบคุมระบบวัดขนาดระยะการเคลื่อนที่ของแท่นเลื่อนและเครื่องมือ ตำแหน่งของจุดอ้างอิงจะถูกกำหนดไว้ก่อนล่วงหน้าอย่างเที่ยงตรงในทุก แนวแกนของการเคลื่อนที่ด้วยสวิตช์จำกัดระยะ หรือสวิตช์ขาเตะ (Limit switches หรือ Trip dogs) ดังนั้น ค่าโคออดิเนทของจุดอ้างอิงจะมีค่าเท่าเดิมเสมอ และรู้ค่าตัวเลขที่แน่นอนที่สัมพันธ์กับจุดศูนย์ของเครื่อง

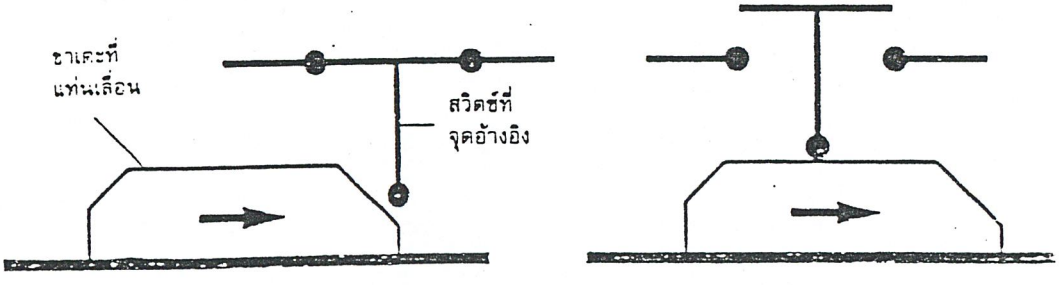


รูปที่ 2.36 ตำแหน่งจุดอ้างอิงของเครื่องกัด

ข้อสำคัญ : *หลังจากเปิดสวิตช์ระบบควบคุมแล้ว แนวแกนทั้งหมดจะต้องเลื่อนไปยังจุดอ้างอิงก่อนเสมอ เพื่อปรับค่าระบบวัดระยะการเคลื่อนที่*

ถ้าเกิดเหตุขัดข้องขึ้นจนทำให้ข้อมูลของตำแหน่งแท่นเลื่อนและเครื่องมือในปัจจุบัน สูญหายไปจากระบบควบคุม ซึ่งอาจมีสาเหตุมาจากไฟฟ้าดับ เป็นต้น จะต้องเลื่อนแท่นเลื่อนต่างๆ กลับไปหาจุดอ้างอิงก่อนเริ่มทำงานใหม่เสมอ เพื่อปรับค่าของระบบวัดระยะการเคลื่อนที่ให้ถูกต้อง

เครื่องจักรกลซีเอ็นซีที่มีระบบวัดระยะการเคลื่อนที่แบบสัมบูรณ์(Absolute traverse measurement) อาจจะไม่ต้องใช้จุดอ้างอิง ทั้งนี้เพราะว่าสามารถอ่านค่าโคออดิเนทการเคลื่อนที่ของแนวแกนต่างๆ ได้โดยตรงและทุกเวลาที่ต้องการ อย่างไรก็ตาม สำหรับเครื่องจักรกลซีเอ็นซีที่ใช้ระบบวัดระยะการเคลื่อนที่แบบต่อเนื่อง (Incremental traverse measuring system) ยังต้องการใช้จุดอ้างอิงสำหรับการปรับค่าระบบวัดระยะการเคลื่อนที่

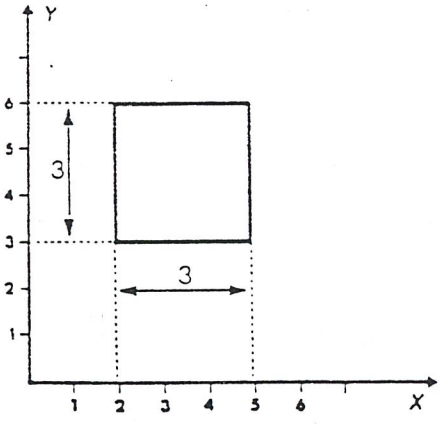


รูปที่ 2.37 การเคลื่อนที่ไปยังตำแหน่งของจุดอ้างอิง

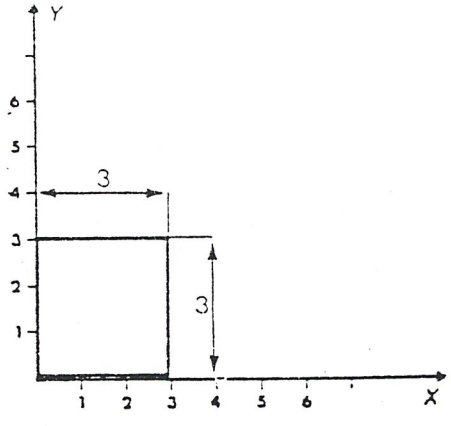
เมื่อขาเตะ (trip dog) เลื่อนสวิตช์ที่จุดอ้างอิงให้เปิดออกแล้ว ระบบวัดระยะการเคลื่อนที่ก็จะปรับค่าไปที่ศูนย์หรือค่าที่กำหนดไว้ล่วงหน้า เพื่อให้ได้ระดับความถูกต้องที่ต้องการ ในขณะที่ทำการปรับค่าของระบบวัดขนาดการเคลื่อนที่ของแท่นเลื่อนไปยังจุดอ้างอิงควรจะลดความเร็วลงและเลื่อนไปในทิศทางเดียวกันตลอด สำหรับเครื่องจักรบางแบบ เช่น เครื่องกัด ระบบวัดขนาดสามารถที่จะปรับค่าโดยการเลื่อนแท่นเลื่อนไปที่จุดศูนย์ของเครื่องก็ได้ อย่างไรก็ตาม ในกรณีทั่วไป การเคลื่อนที่ไปยังจุดศูนย์ของเครื่องจะไม่สามารถทำได้ เมื่อเครื่องมือและชิ้นงานอยู่ในตำแหน่งทำงานแล้ว ดังนั้น จึงต้องเพิ่มจุดอ้างอิงเข้าไปด้วย

3) จุดศูนย์ของชิ้นงาน (Workpiece zero point)

ในระบบโคออดิเนทสามารถที่จะเลือกข้อมูลโคออดิเนทได้ด้วยวิธีง่ายๆ โดยการกำหนดตำแหน่งที่จะวางชิ้นงานลงในระบบโคออดิเนทที่สะดวกต่อการอ่านค่าจุดโคออดิเนท รูปที่ 2.38 แสดงให้เห็นการวางแบบชิ้นงานสี่เหลี่ยมตรงจุดใดๆ ในระบบโคออดิเนท ส่วนรูปที่2.39 เป็นแบบชิ้นงานเดิมที่วางให้ขอบงานสองด้านซ้อนทับแกน X และแกน Y ซึ่งวิธีหลังนี้จะสามารถใช้ค่าขนาดที่กำหนดในแบบชิ้นงานเป็นค่าโคออดิเนทได้เลย และสามารถตรวจสอบค่าได้ง่ายกว่า และยังช่วยหลีกเลี่ยงการคำนวณหาค่าโคออดิเนทเพิ่มเติมได้



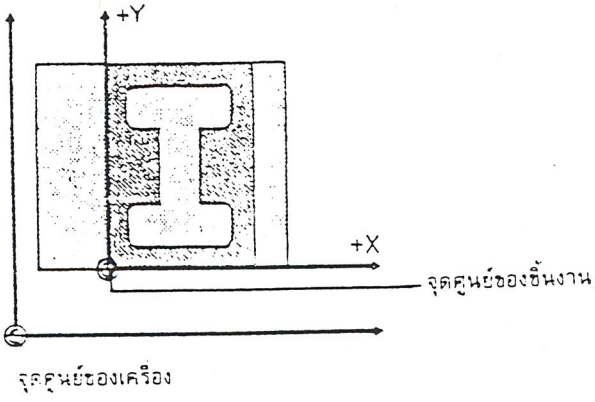
รูปที่ 2.38 การวางแบบชิ้นงานที่จุดใดๆ ในระบบโคออดิเนท



รูปที่ 2.39 การวางแบบชิ้นงานโดยใช้แนวแกนของระบบโคออดิเนทเป็นหลัก

จุดศูนย์ของชิ้นงาน (W) มักจะแสดงด้วยสัญลักษณ์จะเป็นจุดที่ช่วย ในการกำหนดระบบโคออดิเนทของชิ้นงานที่สัมพันธ์กับจุดศูนย์ของเครื่อง จุดศูนย์ของ ชิ้นงานจะถูกเลือกใช้โดยผู้เขียนโปรแกรม และป้อนเข้าไปในระบบซีเอ็นซีในขั้นตอน ของการปรับตั้งตำแหน่งของจุดศูนย์ของชิ้นงานสามารถที่จะกำหนดเลือกใช้ได้อย่างอิสระโดยผู้เขียนโปรแกรม แต่ต้องอยู่ภายในขอบเขตการทำงานของเครื่องจักรกล โดยมีหลักเกณฑ์ง่ายๆ คือ การกำหนดตำแหน่งจุดศูนย์ของชิ้น

งานควรจะกำหนดไว้ในตำแหน่งที่เป็นจุดอ้างอิงต่างๆ ที่กำหนดไว้ในแบบชิ้นงานอยู่แล้ว กล่าวคือ เมื่อกำหนดตำแหน่งจุดศูนย์ของชิ้นงานแล้ว สามารถที่จะเปลี่ยนขนาดที่กำหนดในแบบชิ้นงานให้เป็นค่าโคออดิเนทได้โดยสะดวก และหลีกเลี่ยงการคำนวณค่าโคออดิเนทเพิ่มเติมได้ สำหรับชิ้นงานกัดที่มีรูปทรงของชิ้นงานไม่สมมาตร มักนิยมใช้ขอบมุมของชิ้น งานด้านใดด้านหนึ่งเป็นจุดศูนย์ของชิ้นงาน ส่วนชิ้นงานกัดที่สมมาตร มักจะใช้จุดศูนย์กลางของชิ้นงานเป็นตำแหน่งเป็นตำแหน่งจุดศูนย์ของชิ้นงาน



รูปที่ 2.40 ตัวอย่างการกำหนดจุดศูนย์ของชิ้นงานกัด

โดยทั่วไปแล้ว ตำแหน่งจุดศูนย์ของชิ้นงานมักนิยมใช้เป็นตำแหน่งเดียวกันกับจุดศูนย์ของโปรแกรม (Program zero point)

สรุป : ตำแหน่งจุดศูนย์ของชิ้นงาน มีข้อพิจารณาในการเลือกใช้ดังนี้

1. สามารถกำหนดค่าโคออดิเนทจากขนาดกำหนดในแบบงานได้มากที่สุด

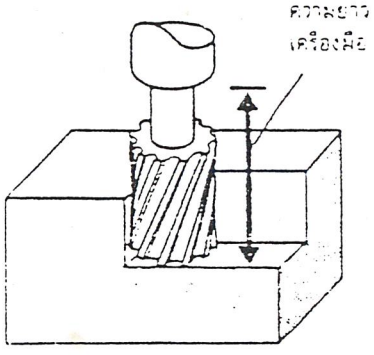
2. สามารถจับยึดชิ้นงาน ปรับตั้ง และตรวจสอบ ตลอดจนการควบคุมด้วยระบบวัดระยะการเคลื่อนที่ได้สะดวก

4) จุดอ้างอิงของเครื่องมือ (Tool reference points)

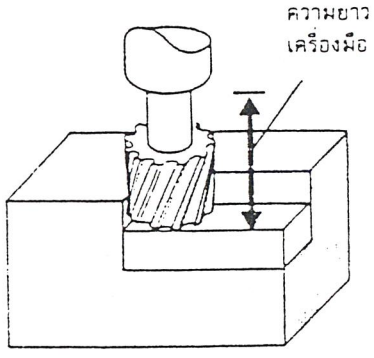
ในการเขียนโปรแกรมสำหรับการตัดเฉือนชิ้นงานตามเส้นขอบรูป สิ่งสำคัญ ที่จะต้องคำนึงถึง คือ การเคลื่อนที่ของขอบคมตัดของเครื่องมือที่ยึดอยู่กับชุดพาเครื่องมือ (Tool carrier) เช่น เพลางานของเครื่องกัด เป็นต้น จะต้องเคลื่อนที่ในลักษณะที่ทำให้ขอบคมตัดของมีดกัดเคลื่อนที่ตามเส้นขอบรูปของชิ้นงานอย่างถูกต้อง

การเขียนโปรแกรมขนาดความยาวของเครื่องมือ จะต้องตรงกับขนาดความยาวของเครื่องมือที่เป็นจริง (รูปที่ 2.41) ถ้าระบบควบคุมเริ่มทำงานด้วยขนาดความยาวของเครื่องมือที่ไม่ถูกต้องก็จะทำให้ไม่ได้ขนาดของเส้นขอบรูปที่ต้องการ ถ้าขนาดของเครื่องมือสั้นเกินไป (รูปที่ 2.42) ก็จะ

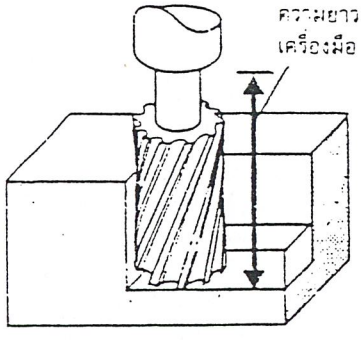
ตัดเฉือนเนื้อวัสดุออกไม่หมด และถ้าเครื่องมือยาวเกินไป (รูปที่ 2.43) ก็จะทำให้เกิดการตัดเฉือนเนื้อวัสดุชิ้นงานออกมากเกินไป



รูปที่ 2.41 ความยาวของเครื่องมือที่ถูกต้อง



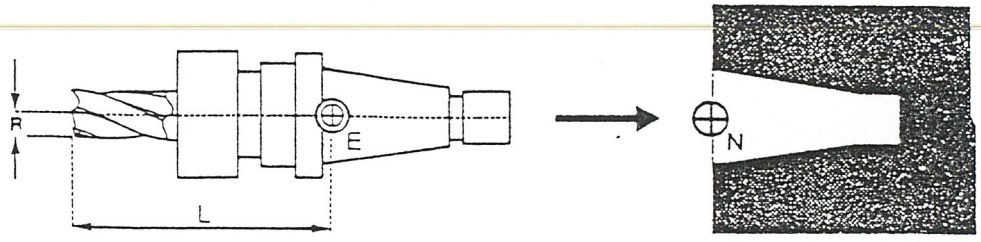
รูปที่ 2.42 เครื่องมือสั้นเกินไป



รูปที่ 2.43 เครื่องมือยาวเกินไป

ดังนั้น ขนาดความยาวของเครื่องมือ จึงต้องทำการวัดขนาดก่อนที่เครื่องจักรกลจะเริ่มทำงานตามโปรแกรมและข้อมูลที่วัดได้จะถูกป้อนเข้าไปไว้ในระบบความจำส่วนที่ทำหน้าที่เก็บข้อมูลของเครื่องมือ (Tool data storage) ของระบบควบคุมซีเอ็นซี

ในการตัดเฉือนชิ้นงาน จุดสำคัญ คือ จะต้องสามารถควบคุมจุดปลายเครื่องมือหรือขอบคมตัดที่สัมพันธ์กับขนาดของเครื่องมือตลอดเส้นทางการเคลื่อนที่ตัดเฉือน(Machining path) เนื่องจากเครื่องมือมีรูปทรงและขนาดที่แตกต่างกัน ดังนั้น ขนาดของเครื่องมือที่ถูกต้องจะต้องหาให้ได้ก่อนและป้อนเข้าไปในระบบควบคุม ซึ่งขนาดของเครื่องมือจะต้องปรับตั้งให้สัมพันธ์กับจุดปรับตั้งเครื่องมือ (Tool setting point) ที่อยู่คงที่



รูปที่ 2.44 จุดปรับแต่งเครื่องมือของเพลามีดกัด

1. จุดปรับตั้งเครื่องมือ(Tool setting point , E)

แสดงด้วยสัญลักษณ์ ซึ่งจะเป็นจุดที่มีตำแหน่งที่แน่นอนบนด้ามจับยึดเครื่องมือดังแสดงในรูป 2.44 จุดปรับตั้งเครื่องมือนี้จะอยู่ในตำแหน่งที่ทำให้สามารถวัดขนาดความยาวของเครื่องมือว่าห่างจากปลายเพลายึดเครื่องมือนั้นเท่าไร โดยทั่วไปค่าขนาดของเครื่องมือที่จะต้องป้อนเข้าไปในระบบควบคุมจะประกอบด้วย

- ความยาวของเครื่องมือ จะใช้ค่าโคออดิเนท Z หรือ L
- ระยะเยื้องศูนย์ของจุดปลายเครื่องมือหรือรัศมีของเครื่องมือ ซึ่งจะใช้ค่าโคออดิเนท X,R หรือ Q

ตำแหน่งที่จุดปรับตั้งเครื่องมือสัมผัสกับเพลายึดเครื่องมือ คือ จุดปลายรูสวมยึดเครื่องมือ (Tool socket point , N) ซึ่งมักจะแสดงด้วยสัญลักษณ์เมื่อสวมเครื่องมือหรือด้ามยึดเครื่องมือเข้าไปในรูของเพลายึดเครื่องมือของชุดพาเครื่องมือ เช่น ชุดเทอเรท เป็นต้น จุดปรับตั้งเครื่องมือจะซ้อนทับกับจุดปลายรูสวมยึดเครื่องมือพอดี

สรุป : จุดอ้างอิงของเครื่องมือมีความสำคัญสำหรับการปรับตั้งเครื่องมือมาก ข้อมูลเกี่ยวกับเครื่องมือจะต้องป้อนเข้าไปและเก็บไว้ในหน่วยความจำที่ทำหน้าที่เก็บข้อมูลของเครื่องมือก่อนที่จะเริ่มทำงานตัดเฉือนตามโปรแกรมที่เตรียมไว้

ระบบควบคุมจะใช้จุดปลายรูสวมยึดเครื่องมือร่วมกับขนาดของเครื่องมือที่ป้อนข้อมูลไว้ ทำให้สามารถกำหนดตำแหน่งของจุดปลายเครื่องมือที่สัมพันธ์กับจุดศูนย์ของเครื่องได้อย่างถูกต้อง สำหรับเครื่องจักรกลที่มีชุดพาเครื่องมือที่ซับซ้อน เช่น ชุดป้อมมีด เทอเรทของเครื่องกลึง แม็กกาซีนเครื่องมือของแมชชีนนิ่งเซนเตอร์ เป็นต้น นอกจากจะต้องคำนวณจุดอ้างอิงของเครื่องมือที่สัมพันธ์กับจุดปลายรูสวมยึดเครื่องมือแล้ว ยังต้อง พิจารณาให้สัมพันธ์กับจุดอ้างอิงการเคลื่อนที่ของชิ้นส่วนต่างๆ ของเครื่องจักรกลด้วย เช่น จุดอ้างอิงของชุดพาเครื่องมือ จุดอ้างอิงของแท่นเลื่อนต่างๆ เป็นต้น

2. จุดอ้างอิงของชุดพาเครื่องมือ (Tool carrier reference point , T)

มักจะใช้แทนด้วยสัญลักษณ์ ซึ่งเป็นจุดที่อยู่คงที่บนชุดพาเครื่องมือ เช่นชุดเทอเรท ชุดแม็กกาซีน เป็นต้น และโดยทั่วไปตำแหน่งของจุดอ้างอิงชุดพาเครื่องมือจะอยู่ที่ตำแหน่งจุดศูนย์กลางของการหมุน หรือจุดศูนย์กลางของการเคลื่อนที่ของชุดพาเครื่องมือ

3. จุดอ้างอิงของแท่นเลื่อน (Slide reference point , F)

จะใช้แทนด้วยสัญลักษณ์ เป็นจุดที่อยู่ในพื้นที่ทำงานของเครื่องจักรกลที่ ถูกกำหนดตำแหน่งไว้อย่างถูกต้อง

บทที่ 3

เครื่องมือและอุปกรณ์

ในระบบของเครื่องกัดแนวดิ่งซีเอ็นซีควบคุมโดยไมโครคอมพิวเตอร์นั้น จะมีส่วนประกอบหลักๆ อยู่ 2 ส่วนดังนี้

1. ส่วนของเครื่องกัดแนวดิ่งซีเอ็นซี
2. ส่วนของระบบควบคุมเซอร์โวมอเตอร์

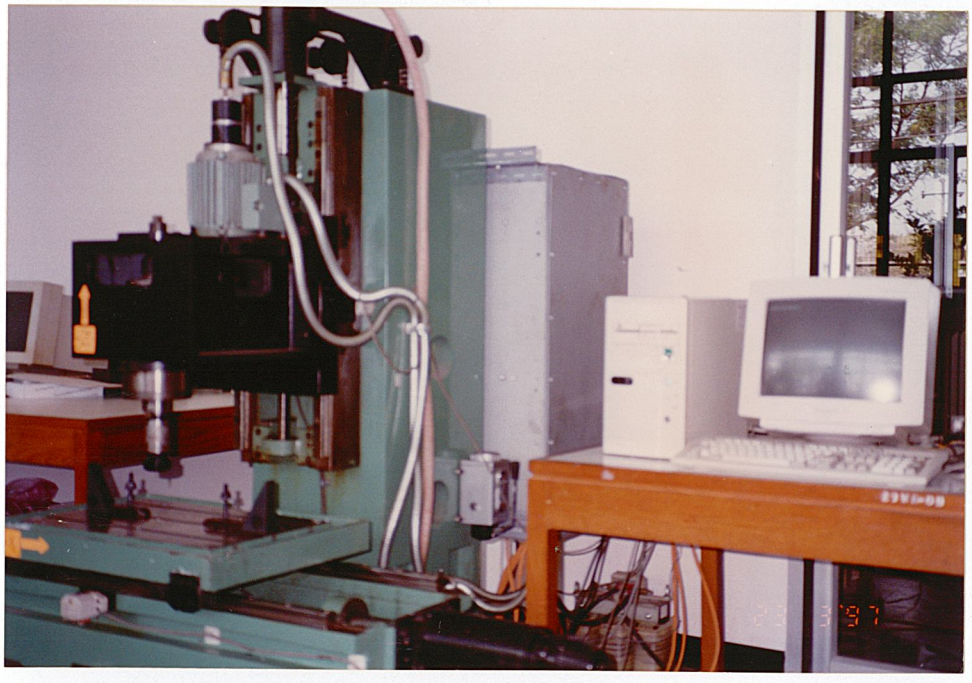
โดยในแต่ละส่วน ประกอบไปด้วยอุปกรณ์และเครื่องมือต่างๆ ต่อไปนี้

3.1 ส่วนของเครื่องกัดแนวดิ่งซีเอ็นซี

เครื่องกัดแนวดิ่งซีเอ็นซีที่ใช้ในการทดลอง มีรายละเอียดทางด้านเทคนิคดังนี้

1. โต๊ะงานทำด้วยเหล็กหล่อ ขนาดกว้าง 430 ยาว 600 หนา 90 มม. มีร่องตัวทีขนาด 10 มม ตามแนวยาวของโต๊ะงานจำนวน 3 ร่อง
2. องค์ประกอบของโต๊ะงานทำด้วยเหล็กหล่อ ขนาดกว้าง 350 ยาว 1150 สูง 180 มม.
3. ฐานเครื่องทำด้วยเหล็กหล่อขนาดกว้าง 650 ยาว 1260 สูง 315 มม.
4. Column ทำด้วยเหล็กหล่อขนาดกว้าง 380 ยาว 400 สูง 1200 มม.
5. Ball screw ขนาด 25 มม. ยาว 650 มม. ระยะ pitch 5 มม. ระยะ lead 5 มม. ในแนวแกน X,Y
6. Ball screw ขนาด 25 มม. ยาว 875 มม. ระยะ pitch 5 มม. ระยะ lead 5 มม. ในแนวแกนZ
7. Linear Motion ในแนวแกน X,Y และ Z
8. ชุดเพลาขับหัวเครื่อง มีรายละเอียดดังนี้
   - มีขนาดของรูเพลาที่ใช้กับด้ามมีด ขนาดมาตรฐานไม่เล็กกว่า BT30
   - ทนความเร็วรอบสูงสุดไม่ต่ำกว่า 3500 รอบ / นาที
   - ขับเคลื่อนหัว Spindle ด้วยสายพาน
   - ลูกปืนที่ใช้เป็นแบบ high precision bearing

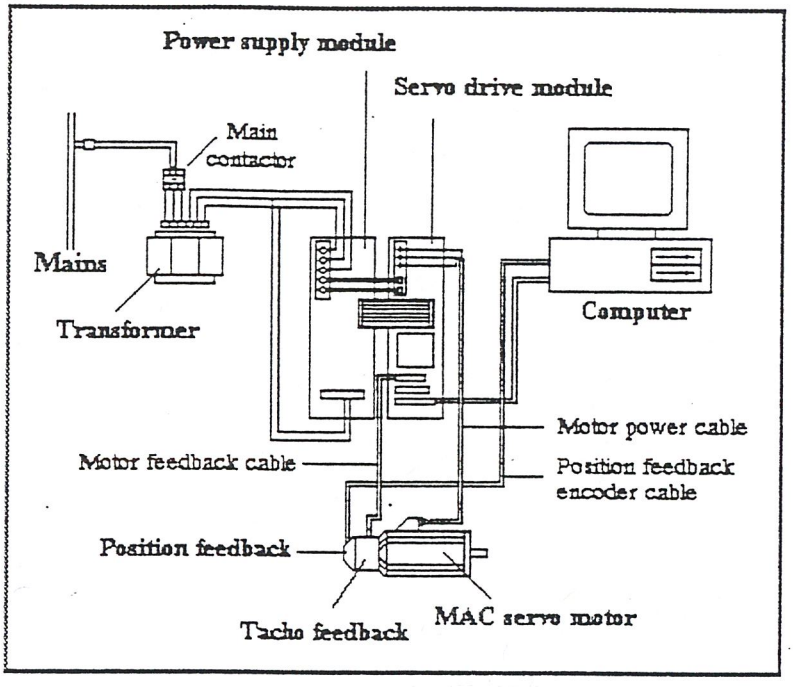
รูปที่ 3.1 ก แสดงโครงสร้างของเครื่องกัดแนวดิ่งซีเอ็นซี



รูปที่ 3.1 ข แสดงโครงสร้างของเครื่องกัดแนวดิ่งซีเอ็นซี (ต่อ)

3.2 ส่วนของระบบควบคุมเซอร์โวมอเตอร์

ประกอบด้วยอุปกรณ์ต่างๆ ดังแสดงในรูปที่ 3.2



รูปที่ 3.2 แสดงระบบควบคุมเซอร์โวมอเตอร์

จากภาพจะพบว่าในระบบควบคุมมีอุปกรณ์ที่สำคัญๆ ดังนี้

3.2.1. การ์ดควบคุมสำเร็จรูป Model 5650

3.2.1.1 ลักษณะและความสามารถ

1) ใช้ชุดชิพของ PMD รุ่น MC1401
2) รีจิสเตอร์เก็บค่าตำแหน่ง,ความเร็ว,ความเร่ง และเจิร์ค ขนาด 32 บิต
3) สามารถรับค่าจากลิมิตสวิตช์ ป้องกันการเคลื่อนที่เกินขอบเขตได้ 2 ตัว ต่อ 1 แกน
4) มีรูปแบบการเคลื่อนที่ 3 แบบ คือ S-curve,Trapezoidal และ Velocity contouring
5) ใช้เวลา 100 µS ต่อการ update ข้อมูลหนึ่งครั้ง
6) มี Electronic Gearing
7) มีรูปแบบของ Digital Filter 2 แบบ คือ PID และ PIVFF
8) สามารถรับสัญญาณพัลซ์ที่มาจากเอนโคดเดอร์ได้สูงถึง 1 ล้านพัลซ์ต่อวินาที
9) มีรีจิสเตอร์ที่ใช้เก็บค่าตำแหน่ง Home และ Index ความเร็วสูง
10) มีสัญญาณควบคุม 2 แบบให้เลือก คือ 16-bit + 10V DAC และ 10-bit PWM

11) สามารถเปลี่ยนแปลงรูปแบบการเคลื่อนที่และพารามิเตอร์ได้ขณะที่มีการทำงานอยู่

12) มี counter แบบ 1/T สำหรับการเคลื่อนที่ที่ความเร็วต่ำๆ

13) มีการหยุดมอเตอร์โดยอัตโนมัติเมื่อมีข้อผิดพลาดเกิดขึ้น

14) ผู้ใช้สามารถกำหนดอินเตอร์รัพได้

15) ใช้พอร์ต I/O ขนาด 8 บิต

3.2.1.2 การใช้งานการ์ดควบคุมสำเร็จรูป Model 5650

1) การติดตั้ง

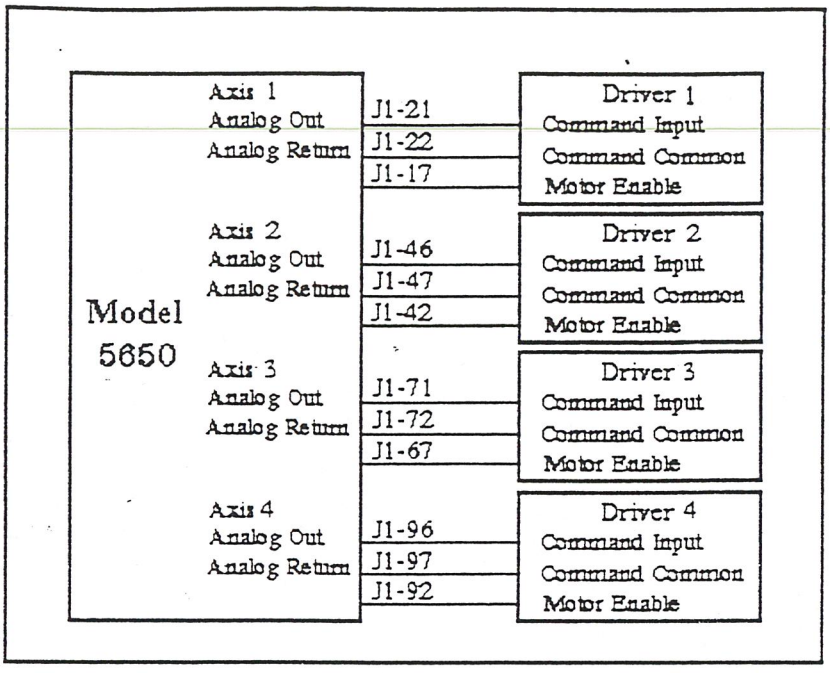
การ์ดควบคุม ต้องการใช้ไฟเลี้ยงขนาด +5 Vdc และ +12 Vdc จาก ISA busในเครื่องคอมพิวเตอร์ จึงต้องทำการเสียบการ์ดควบคุมลงในช่องเสียบของ ISA bus ที่มีอยู่บน Main Board ของคอมพิวเตอร์ แล้วทำการเซ็ต Address ของการ์ดควบคุมให้อยู่ในตำแหน่งที่ยังว่างอยู่ ในที่นี้เซ็ตค่า Address เป็น 300H สำหรับการเซ็ตค่า Address ของการ์ดควบคุมนั้นสามารถทำได้โดย การเซ็ต Jumper W2 เสียใหม่

2) การเชื่อมต่อ

การเชื่อมต่อระหว่างการ์ดควบคุมกับอุปกรณ์ภายนอกต่างๆเช่น Driver,Encoder และ Limit Switchs สามารถทำได้โดย วิธีการต่อไปนี้

2.1) Driver - สัญญาณควบคุมที่ะส่งจากการ์ดควบคุมไปยัง Driver นั้นเราจะใช้สัญญาณไฟฟ้าขนาด 0 ถึง + 10V ในการควบคุมความเร็วและทิศทางการหมุนของ Survo Motor ดังนั้นสายสัญญาณที่เชื่อมระหว่างการ์ดควบคุม และ Driver จะใช้ 2 เส้นต่อการควบคุม Survo Motor 1แกน โดยเส้นหนึ่งเป็นสายของสัญญาณไฟฟ้าแบบAnalogส่วนอีกเส้นหนึ่งจะเป็นสายสัญญาณอ้างอิง(Analog Return)สำหรับสายสัญญาณทั้ง Analog Output และ Analog Return นั้นจะต่อออกมาจากConnector J1

สัญญาณอีกตัวหนึ่งที่ส่งจากการ์ดควบคุมไปยัง Driver ก็คือ Motor Enable Signal ซึ่งทำหน้าที่ในการยกเลิกการใช้งาน (Disable) Driver ชั่งคราวในช่วงที่การ์ดควบคุมเริ่มทำงาน (Power-up) เพื่อป้องกันไม่ให้เกิดความผิดพลาดขึ้นในระหว่างเริ่มเปิดเครื่องหลังจากที่การ์ดควบคุมพร้อมที่จะทำงานแล้วจึงทำการเรียกใช้(Enable) Driver ตามปกติ โดยในแต่ละแกนจะมีสัญญาณนี้ 1 เส้น โดยทำการต่อออกมา จาก Connector J1 ในกรณีที่ต้องการเปลี่ยนสัญญาณ Output ที่สายสัญญาณนี้ จากเดิมในสถานะ Power-up (+ 5 Vdc) และ สถานะพร้อมทำงาน (0 Vdc) ไปเป็นสถานะ Power-up(0 Vdc)และสถานะพร้อมทำงาน(+ 5 Vdc) ก็สามารถทำได้โดยการเซ็ต Jumper W3 ใหม่



รูปที่ 3.3 แสดงการเชื่อมต่อระหว่าง Model 5650 กับ Driver

2.2) Encoder - สำหรับ Encoder ที่ใช้นั้นจะเป็นแบบ Differential Encoder ซึ่งสัญญาณจะมีทั้งหมด 3-เฟสคือ เฟสA,เฟสB,และ Index แต่ละเฟสก็จะมีสายสัญญาณ 2 เส้นคือสายสัญญาณจริงหนึ่งเส้นและสัญญาณ Completement อีกหนึ่งเส้นรวม 6 เส้นอีก 2 เส้นจะเป็นสายไฟเลี้ยงที่จ่ายให้กับ Encoder รวมทั้งหมดจะมีสายสัญญาณอยู่ 8 เส้นต่อ Encoder 1 ตัว ในส่วนของการต่อเข้ากับการ์ดควบคุมนั้นสามารถทำได้โดยต่อสายเข้ากับConnector J1

2.3) Limit Switchs - ในแต่ละแกนของ Survo Motor จะต้องใช้ Limit Switch จำนวน 3 ตัว โดยแบ่งหน้าที่เป็น

- Over-travel Limit Input จำนวน 2 ตัว ทำหน้าที่ในการส่งสัญญาณไปยังการ์ดควบคุมเพื่อบอกให้ทราบว่าขณะนั้นโต๊ะงานได้เคลื่อนที่ไปจนสุดขอบเขตของการเคลื่อนที่แล้ว
- Home Inpuut จำนวน 1 ตัวทำหน้าที่ในการส่งสัญญาณไปยังการ์ดควบคุมเพื่อเพิ่มหรือลดค่าใน Position Capture Register ซึ่งสามารถนำผลจากการเปลี่ยนแปลงนี้ไปประยุกต์ได้

สำหรับสัญญาณจาก Limit Switchที่จะต่อเข้าการ์ดควบคุมนั้น จะต้องต่อเข้ากับ Connector J1

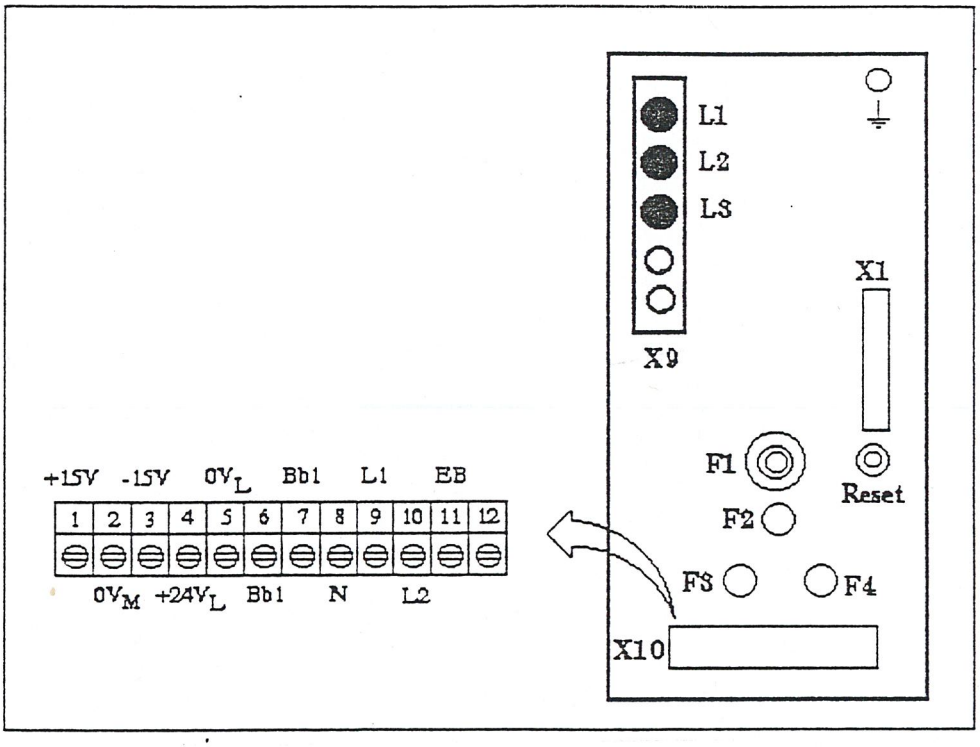
3) การควบคุม

การควบคุมการทำงานของการ์ดควบคุมให้สามารถทำงานได้ตามที่เราต้องการนั้น สามารถทำได้โดยการเขียนโปรแกรมสั่งงานการ์ดผ่านทางคอมพิวเตอร์ โดยภาษาที่จะใช้การเขียนโปรแกรมนั้น ในที่นี้จะใช้ภาษา C เนื่องจาก เราจำเป็นจะต้องใช้ชุดคำสั่งที่ให้มาพร้อมกับการ์ดควบคุม ชุดคำสั่งนี้ปรากฏอยู่ในโปรแกรม host_io.c ซึ่งเขียนขึ้นด้วยภาษา C จึงเป็นการง่ายที่จะใช้ภาษา C ในการเขียนโปรแกรมควบคุม

ในส่วนรายละเอียดของโปรแกรมควบคุมที่จะเขียนนั้น ก็จะเป็นคำสั่งที่เรียกใช้งานชุดคำสั่งในโปรแกรม host_io.c เสียเป็นส่วนใหญ่ ซึ่งจะอยู่ในคู่มือการใช้งานของการ์ดอยู่แล้ว จึงไม่ขอกล่าวถึงในที่นี้

3.2.2. Power Supply Module

ใช้ในการสร้างกระแส D.C. จากกระแส A.C. ที่ได้รับ เพื่อป้อนให้กับ Servo Drive Module สำหรับ Power Supply Module ที่ใช้ในการทดลองเป็นของ INDRAMAT รุ่น TVM 1.2 เป็น Power Supply ซึ่งสามารถจ่ายไฟให้กับ Servo Drive Module ได้ถึง 4 ตัว มีกำลังขับสูงสุด 4.1 KW มีลักษณะภายนอก ดังรูป 3.3



รูปที่ 3.4 แสดงลักษณะของ Power Supply Module

การใช้งาน

Power Supply Module (PSM) จะต้องเชื่อมต่อกับอุปกรณ์ 2 ตัว คือ Transformer และ Servo Drive Module (SM) Transformer จะต่อกับ PSM ผ่านสาย cable 3 เฟส โดยแต่ละเฟสจาก Transformer จะต่อเข้ากับ PSM ที่ Connector X9 ที่ตำแหน่ง L1 , L2 และ L3 ตามลำดับ SDM จะต่อกับ PSM ผ่าน Busbar และ Bus Connection X1 โดย Busbar จะต่อระหว่าง Connector X9 ของ PSM ที่ตำแหน่ง L- , L+ กับ Connector X8 ที่ตำแหน่ง L- , L+ ตามลำดับ ส่วน Bus Connection X1 จะต่อระหว่าง Connector X1 ของ PSM กับ Connector X1 ของ SDM และสำหรับในส่วนของ PSM เอง จะต้องมีการต่อไฟเลี้ยงขนาด 220 VAC จาก Connector L1 และ L2 ลงมายัง Connector X10 ของ PSM ที่ตำแหน่ง 8 และ 9 ตามลำดับด้วย

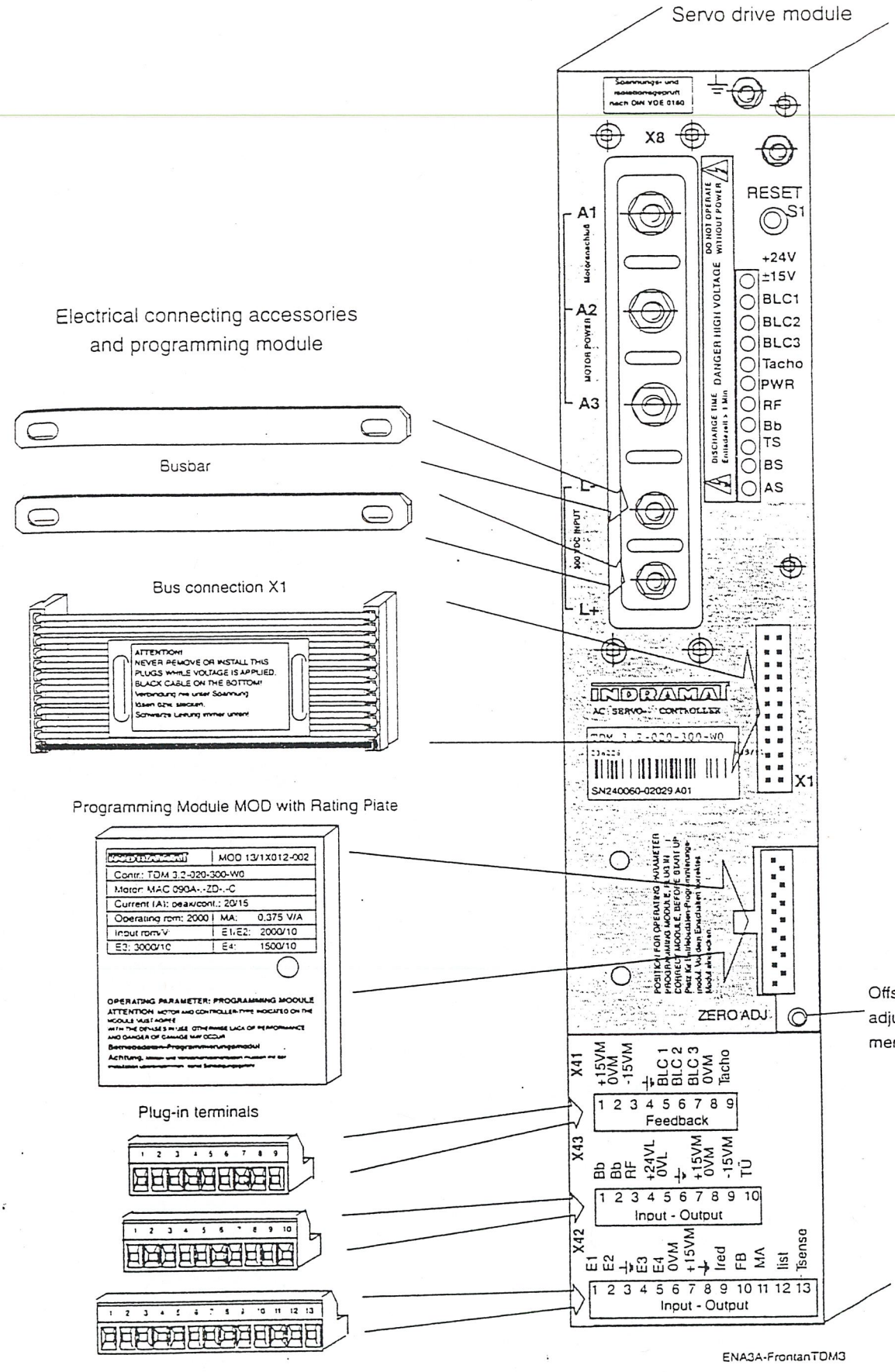
3.2.3. Servo Drive Module

ใช้ในการขับเซอร์โวมอเตอร์โดยจะรับไฟเลี้ยงจาก Power Supply Module และรับคำสั่งจากการ์ดควบคุม สำหรับ Servo Drive Module ที่ใช้ในการทดลองเป็นของ INDRAMAT รุ่น TDM 3.2 ซึ่งใช้สำหรับขับเซอร์โวมอเตอร์ตั้งแต่รุ่น MAC 63 ถึง MAC 112B มีลักษณะภายนอกดังรูป 3.4

การใช้งาน

Servo Drive Module (SDM) จะต้องต่อเชื่อมกับอุปกรณ์อีก 2 ตัว คือ การ์ดควบคุมสำเร็จรูป Model 5650 และเซอร์โวมอเตอร์ ในส่วนของการต่อเชื่อมกับการ์ดควบคุมนั้นได้กล่าวไปแล้วในตอนต้น ดังนั้น ในที่นี้จะกล่าวถึงเฉพาะการเชื่อมต่อกับ เซอร์โวมอเตอร์เท่านั้น

เซอร์โวมอเตอร์จะต่อกับ SDM ผ่านทางสาย Cable 2 เส้น โดยเส้นแรกเป็น Motor Power Cable (Indramat IN250) ต่อระหว่างเซอร์โวมอเตอร์กับ Connector X8 ของ SDM ที่ตำแหน่ง A1,A2 และ A3 ตามลำดับ (ต้องต่อให้ตรงกับอักษรที่เขียนไว้บนสายไฟ) ส่วนอีกเส้นหนึ่งเป็น Motor Feedback Cable (Indramat IN208) ต่อระหว่าง Tacho Feedback กับ Connector X41 ที่ตำแหน่ง 1 ถึง 8



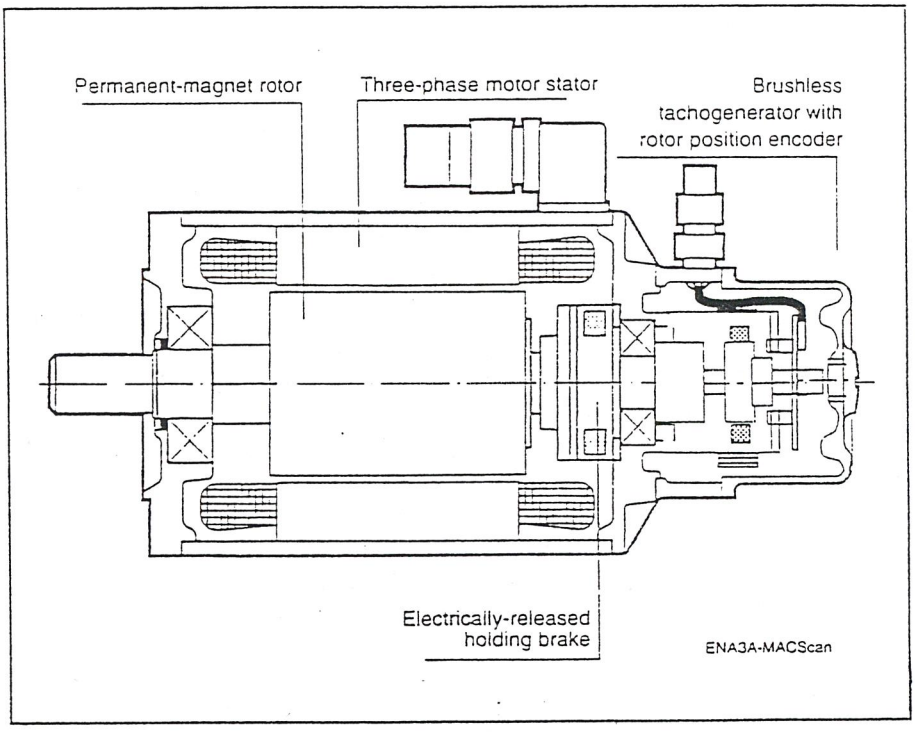
รูปที่ 3.5 แสดงลักษณะของ Servo Drive Module

3.2.4.. MAC Servo Motor

เป็นเอซีเซอร์โวมอเตอร์แบบ 3 เฟส ที่มีลักษณะดังนี้

- สเตเตอร์แบบ 3 เฟส
- โรเตอร์แบบแม่เหล็กถาวร
- มี Electrical-release Brake

โดยมีโครงสร้างและส่วนประกอบต่างๆ ดังรูป 3.5



รูปที่ 3.6 แสดงลักษณะโครงสร้างของ MAC Servo Motor

สำหรับเซอร์โวมอเตอร์ที่ใช้ในการทดลองเป็นเซอร์โวมอเตอร์ของ Indramat รุ่นMAC 63

การใช้งาน

MAC Servo Motor จะต้องเชื่อมต่อกับอุปกรณ์ 2 ตัว คือ SDM และการ์ดควบคุม Model 5650 ซึ่งการเชื่อมต่อกับอุปกรณ์แต่ละตัวนั้นได้กล่าวมาแล้วข้างต้น

บทที่ 4

กลุ่มคำสั่งและความหมายของชุดคำสั่

4.1 ชุดคำสั่ง G (G Function หรือ Preparatory Function)

ตัวเลขที่ตามชุดคำสั่ง G จะเป็นตัวกำหนดความหมายและการทำงานของเครื่อง CNC ซึ่งแบ่งเป็น 2 ประเภท ได้แก่

1) One-shot G code G code ชนิดนี้จะมีผลให้ CNC ทำงานเฉพาะบรรทัดที่ กำหนด เท่านั้น ได้แก่ G code ในกลุ่ม 00

2) Model G code G code ชนิดนี้จะมีผลต่อเนื่องไปยังบรรทัดต่อๆ ไป จน กว่าจะมี G code ในกลุ่มเดียวกันมากำหนด G code ใหม่

ชุดคำสั่ง G ที่ได้พัฒนาขึ้นมาสำหรับโครงงานนี้ (รูปแบบมาตรฐานจากเครื่องFanuc) ได้แก่

G code	กลุ่ม	ความหมาย
G00	01	เข้าสู่ตำแหน่งที่กำหนดอย่างรวดเร็ว
G01		ตัดตามระยะทางที่เป็นเส้นตรง
G02		ตัดตามแนวเส้นโค้งตามเข็มนาฬิกา CW
G03		ตัดตามแนวโค้งทวนเข็มนาฬิกา CCW
G04	00	หยุดชั่วขณะหนึ่งแล้วถอยกลับ
G20*	06	ป้อนหน่วยเป็นนิ้ว
G21		ป้อนหน่วยเป็นมิลลิเมตร
G40*	07	ยกเลิกการชดเชยรัศมีดอกกัด
G41		ชดเชยรัศมีดอกกัดจากซ้ายไปขวา
G42		ชดเชยรัศมีดอกกัดจากขวาไปซ้าย

G80*		ยกเลิกจังหวะที่กำหนดตายตัว
G81		เจาะเป็นจังหวะ , คว้านรูเป็นจุดๆ
G82		เจาะเป็นจังหวะ , การคว้ายผายปากรู
G83	09	เจาะกดเป็นจังหวะ
G85		คว้านรูเป็นจังหวะ
G86		คว้านรูเป็นจังหวะ
G89		คว้านรูเป็นจังหวะ (เจาะคายเศษ)
G90*		โปรแกรมคิดระยะจุดแรกไปถึงจุดสุดท้าย
G91	03	โปรแกรมคิดระยะจุดหนึ่งไปถึงอีกจุดหนึ่ง
G98*		ถอยกลับจังหวะสูงสุด
G99	04	กลับถึงจุดอ้างอิง (จุด R)

หมายเหตุ *1. ที่มี '*' จะเป็น G code ที่กำหนดหลังจากเปิดเครื่อง สามารถใช้ได้ทันทีโดยไม่ต้องอ้างถึง G code ก่อน*

2. G00 แม้จะเป็นคำสั่งในกลุ่ม 01 แต่ก็เป็น One-shot G code

4.2 รายละเอียดเกี่ยวกับ G code

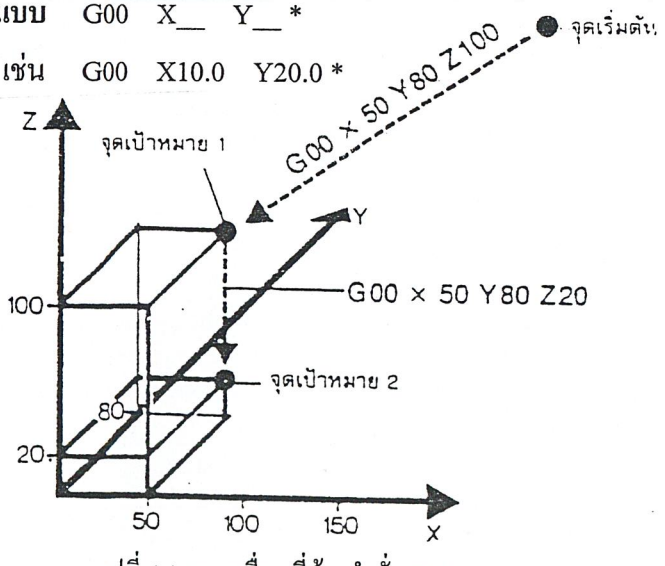
G cede สามารถแบ่งเป็นกลุ่มต่างๆ ซึ่งแต่ละกลุ่มมีความหมายต่างๆ กันดังนี้

4.2.1 กลุ่ม 01 (Interpolation Functions) แบ่งเป็น

1. เข้าสู่ตำแหน่งที่กำหนดอย่างรวดเร็ว (Positioning , G00) การเคลื่อนที่ไปจุดที่กำหนดด้วยความเร็วสูงสุด (2.5 m/min)

รูปแบบ G00 X__ Y__ *

เช่น G00 X10.0 Y20.0 *

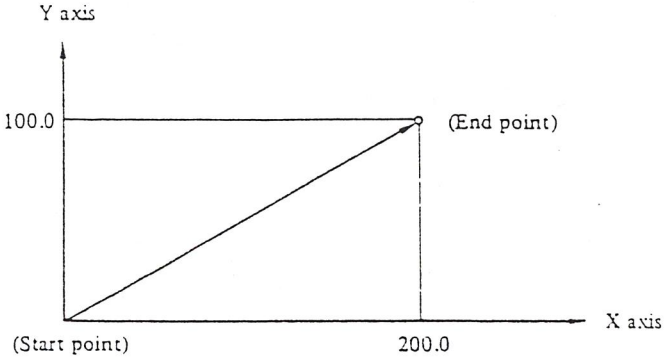


รูปที่ 4.1 การเคลื่อนที่ด้วยคำสั่ง G00

2. ตัดตามระยะทางที่เป็นเส้นตรง (Linear-Interpolation , G01) การเคลื่อนที่เป็นเส้น ตรงไปยังจุดที่กำหนดด้วยความเร็วตาม Feed Rate (Feed Function) ที่กำหนดซึ่งตำแหน่งที่กำหนดขึ้นอยู่กับการกำหนดว่าเป็น G91 หรือ G90 (Incremental หรือ Absolute)

รูปแบบ G01 X__ Y__ * หรือ G01 X__ Z__ *

เช่น (G91) G01 X200.0 Y100.0 F200.0 *



รูปที่ 4.2 การเคลื่อนที่ด้วยคำสั่ง G01

3. ตัดตามแนวโค้งตามเข็มนาฬิกา และ ทวนเข็มนาฬิกา (Circular Interpolation CW and CCW , G02 & G03) การเคลื่อนที่เป็นเส้นโค้งตามเข็มนาฬิกา และทวนเข็ม นาฬิกา ไปตามรัศมี R (ตามรัศมีที่กำหนด) ไปยังจุดที่กำหนด

รูปแบบ G02 X__ Y__ R__ F__ *

เช่น (G91) G02 X60.0 Y20.0 R50.0 F300.0 *

สำหรับ arc (1) (น้อยกว่า 180°)

G91 G02 X60.0 Y20.0 R50.0 F300.0 *

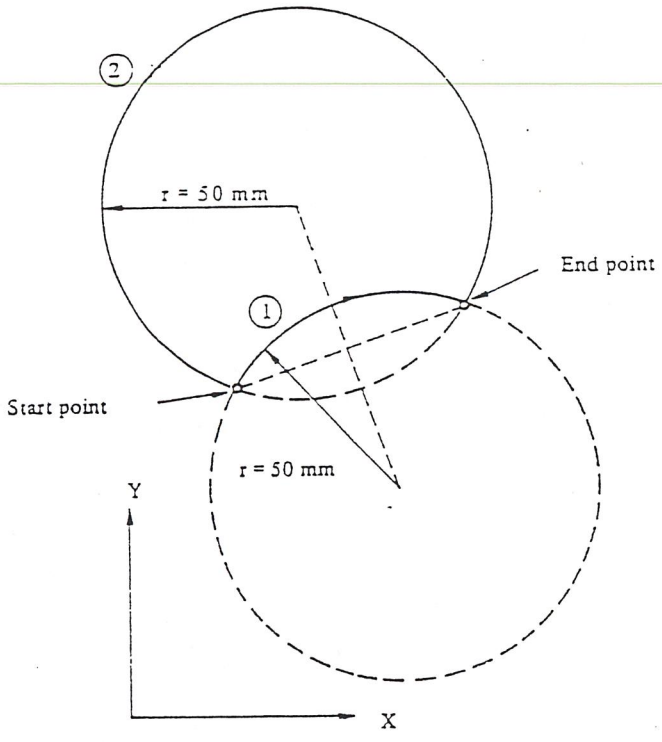
สำหรับ arc (2) (มากกว่า 180°)

G91 G02 X60.0 Y20.0 R50.0 F300.0 *

G02 จะเป็นการเคลื่อนที่ตามเข็มนาฬิกา (CW)

ส่วน G03 จะเป็นการเคลื่อนที่ทวนเข็มนาฬิกา (CCW)

หมายเหตุ *ในคำสั่ง G02 และ G03 จะมีคำสั่ง R เป็นตัวกำหนดขนาดของรัศมีด้วย*



รูปที่ 4.3 การเคลื่อนที่ด้วยคำสั่ง G02

4.2.2 กลุ่ม 00 ได้แก่

1. หยุดชั่วขณะหนึ่งแล้วถอยกลับ (Dwell or Exact Stop , G04) ทำให้การเคลื่อนที่ของ Cutter หยุดชั่วขณะตามเวลา P (หน่วยเป็น sec) ก่อนที่จะทำในบรรทัดถัดไป

รูปแบบ G04 P__ * (0.001 sec - 999.999 sec)

เช่น G04 P20.0 *

4.2.3 กลุ่ม 06 ได้แก่

1. ป้อนหน่วยเป็นนิ้ว (Input in inch , G20)
2. ป้อนหน่วยเป็นมิลลิเมตร (Input in mm. , G21)

Unit systems	G code	Least input icrement
Inch	G20	0.0001 inch
Millimeter	G21	0.001 mm

หมายเหตุ *โดยธรรมดาถ้าไม่ใส่ G20 / G21 จะถือว่าเป็น G21 ก่อน*

4.2.4 กลุ่ม 07 ได้แก่

1. ยกเลิกการชดเชยรัศมีดอกกัด (Cutter Compensation cancel , G40)

2. ชดเชยรัศมีดอกกัดด้านซ้าย (Cutter Compensation left , G41)

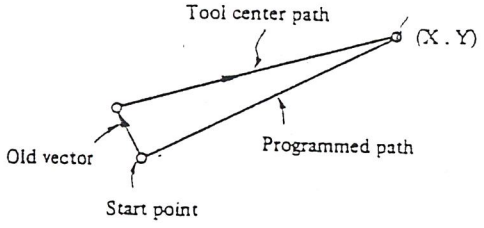
3 ชดเชยรัศมีดอกกัดด้านขวา (Cutter Compensation right , G42)

หมายเหตุ *เมื่อเปิดเครื่องใหม่ จะถือว่าไม่มีการชดเชยรัศมี (G40) อยู่แล้ว โดยไม่ต้องกำหนด G40*

โดยรายละเอียดของ G code ในกลุ่มนี้มีดังนี้

4.2.4.1. ยกเลิกการชดเชยรัศมีดอกกัด (Cutter Compensation , G40) ใช้ร่วมกับ G00 หรือ G01 จะกำหนดในแต่ละแกน เคลื่อนที่ตรงจากจุดของ old vector บน start point ไปยัง end point โดยตรง ตามที่กำหนด (ดูรูปประกอบ)

รูปแบบ G40 D__ *



รูปที่ 4.4 Cutter compnsation cancel

4.2.4.2. การชดเชยรัศมีดอกกัดด้านซ้าย (Cutter Compensation left , G41) สามารถแบ่งรูปแบบการใช้เป็น 2 กรณี ได้ดังนี้

1) ใช้ร่วมกับ G00 , G01 การเคลื่อนที่จะเคลื่อนที่ขนานกับแนวการเคลื่อนที่ของ Program (Programmed path) ไปยังจุดบน new vector ซึ่ง new vector จะตั้งฉากกับทิศทางที่จะเคลื่อนที่ในคำสั่งต่อไป

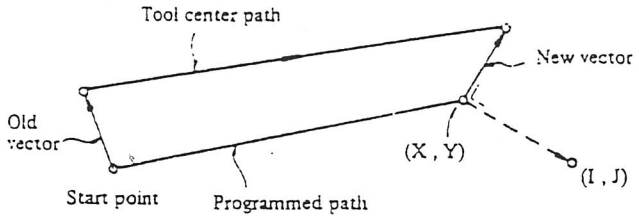
รูปแบบ G41 D__ *

Gxx

(Gxx = G00, G01, G02, G03)

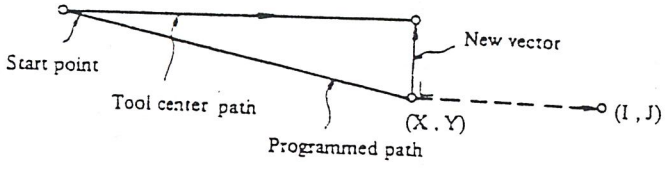
เช่น G41 D6.00 *

G01 X10.00 Y10.00 Z0.00 F100 *



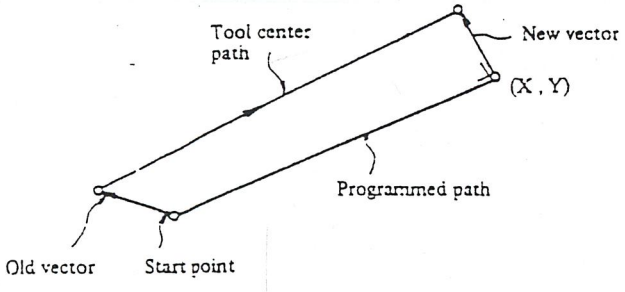
รูปที่ 4.5 Cutter compensate left

ถ้า old vector = 0 การเคลื่อนที่จะเริ่มจาก start point ไปยังจุดบน new vector ทันที ดูรูปที่



รูปที่ 4.6

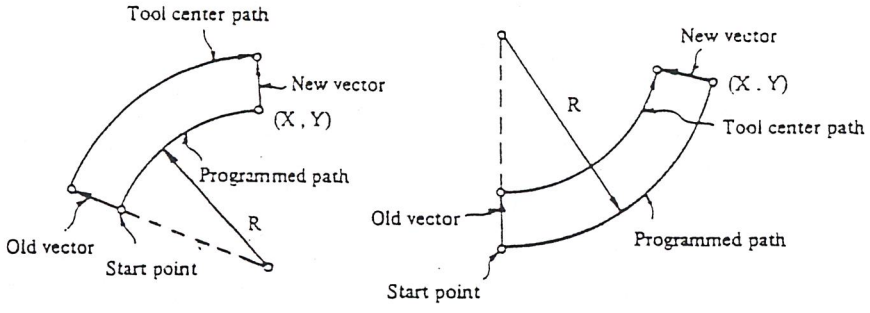
ถ้าไม่มีคำสั่งต่อไป new vector จะตั้งฉากกับ Programmed path ดูรูปที่



รูปที่ 4.7

หมายเหตุ *ขนาดของ new vector จะกำหนด Tool center path ซึ่งการคำนวณได้กล่าว ถึงในส่วนทฤษฎีทางคณิตศาสตร์สำหรับ CNC*

2) ใช้ร่วมกับ G02 , G03 การพิจารณา new vector จะเหมือนกับกรณีแรกคือ พิจารณาทิศทางการเคลื่อนที่เป็นหลัก ซึ่งการเคลื่อนที่จะขนานไปตาม Programmed path แต่มีรัศมีการเคลื่อนที่เพิ่มขึ้นตามรัศมี Cutter และ new vector จะมีทิศทางขนานกับทิศทางจาก arc center ไปยัง end point ดูรูปที่ 4.8

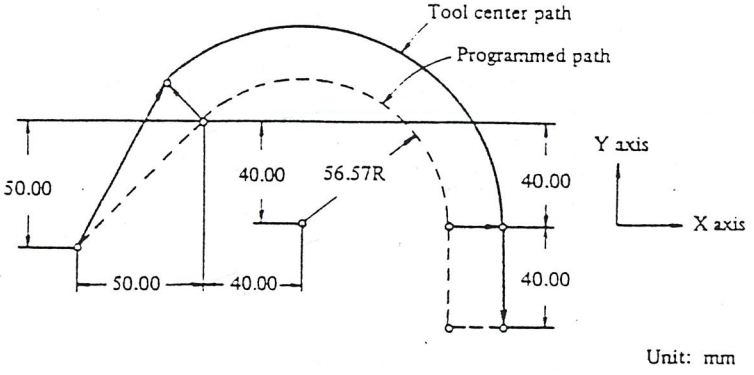


รูปที่ 4.8

4.2.4.3 การชดเชยรัศมีดอกกัดด้านขวา (Cutter Compensation right , G42) การชดเชยทางด้านขวา รูปแบบจะเหมือนกับการชดเชยทางด้านซ้าย แต่ทิศทางของ old vector และ new vector จะเป็นในทิศทางตรงกันข้าม จึงไม่ขอกล่าวรายละเอียด

หมายเหตุ *การชดเชยดอกกัดทั้งซ้ายและขวาในโครงงานนี้จัดทำเฉพาะในแกน X-Y เท่านั้น*

Example of Program of cutter compensation



รูปที่ 4.9 ตัวอย่างการใช้โปรแกรม Compensate

```
N1  G41  D6.00 *
N2  G01  X50.0  F150.0 *
N3  G02  X96.57  Y40.0  R56.57 *
```

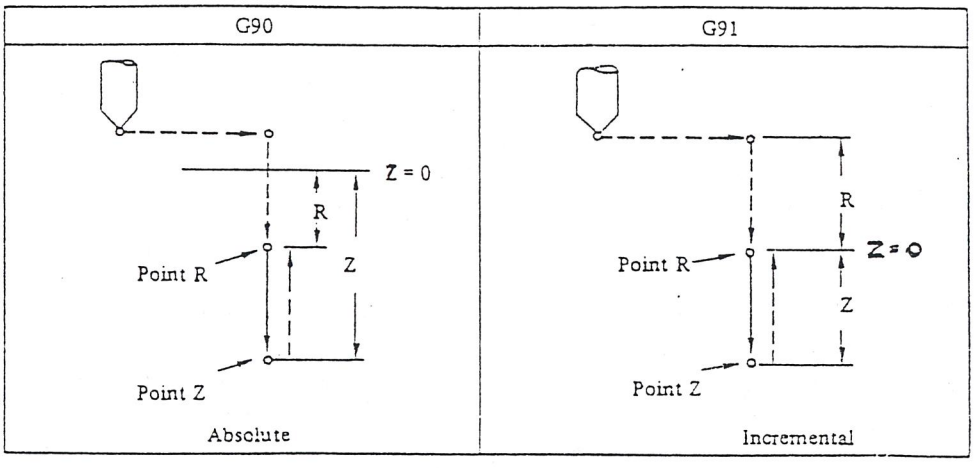
N4 G01 Y40.0 * (Incremental programming)

** หมายเหตุ D หมายถึง เส้นผ่านศูนย์กลางดอกกัด (Cutter)*

4.2.5 กลุ่ม 03 เป็นการกำหนดวิธีการให้ขนาดของแบบในชิ้นงาน แบ่งเป็น

1. การให้ขนาดแบบสัมบูรณ์ (Absolute Dimensions , G90) จะใช้การอ้างอิงจากจุดคงที่จุดหนึ่งในแบบของชิ้นงาน

2. การให้ขนาดแบบต่อเนื่อง (Incremental Dimensions , G91) การวัดขนาดต่างๆ จะอ้างอิงจากตำแหน่งการให้ขนาดครั้งสุดท้าย

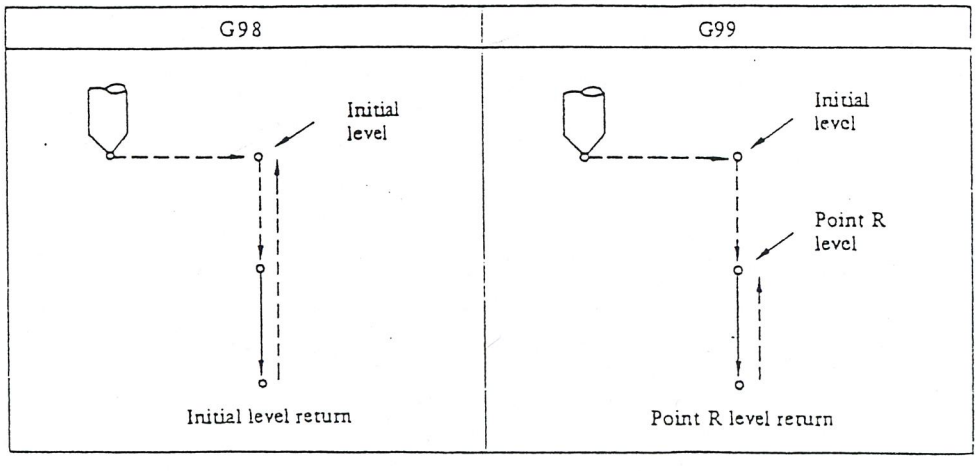


รูปที่ 4.10 Absolute and Incremental Programming

4.2.6 กลุ่ม 04 เป็นการกำหนดรูปแบบของการถอยกลับของ Cutter หลังจากทำการขุดเจาะเสร็จ (รูปแบบของการขุดเจาะจะอยู่ในกลุ่มคำสั่ง 09 ซึ่งจะกล่าวถึงต่อไป) แบ่งเป็น

1. ถอยกลับสู่ความสูงเริ่มต้น (Initial Level , G98)

2. ถอยกลับสู่ความสูงอ้างอิง (Reference Level , G99)



รูปที่ 4.11 Initial level and Point R level

4.2.7 กลุ่ม 09 เป็นกลุ่มคำสั่งกำหนดรูปแบบการขุดเจาะลงบนชิ้นงาน ในลักษณะต่างๆ กัน ซึ่งสำหรับโครงงานนี้จะมีรูปแบบของการขุดเจาะ ดังนี้

การทำการขุดเจาะจะแบ่งรายละเอียดปลีกย่อยของคำสั่งที่ใช้เป็น

1.รูปแบบข้อมูล (Data format) ได้แก่ G90 หรือ G91 (Absolute หรือ Incremental)

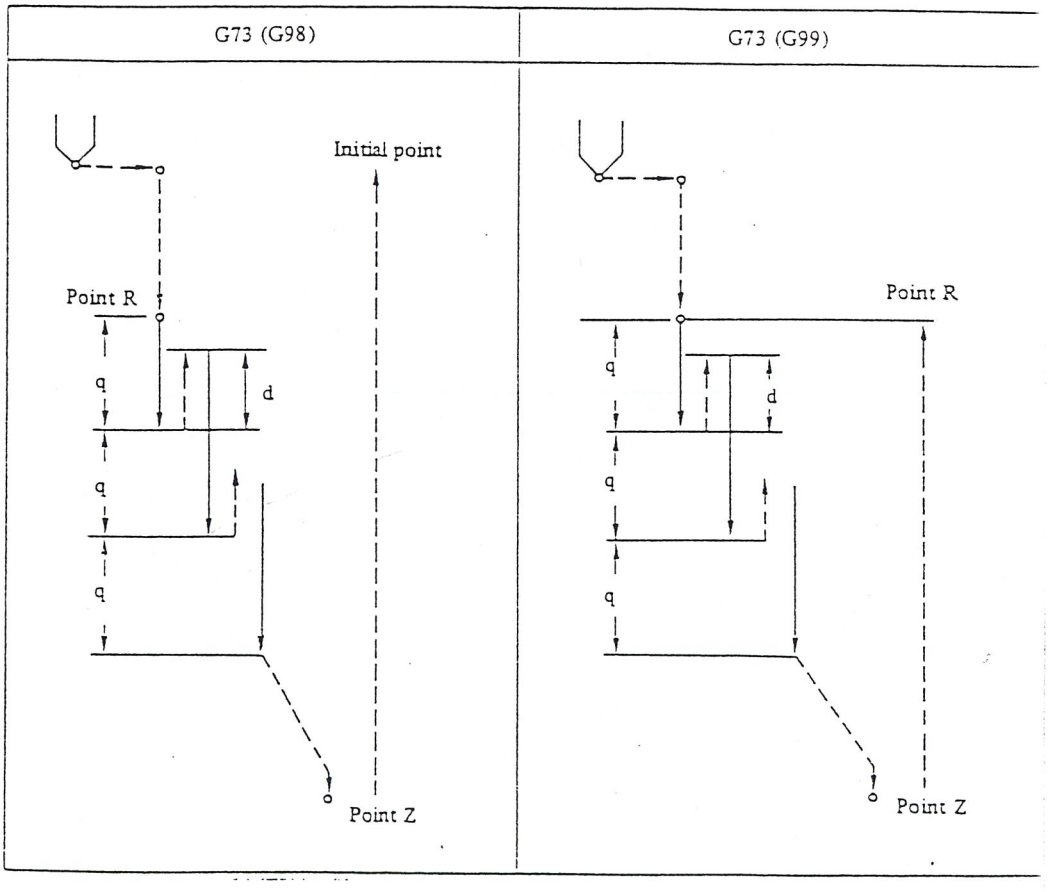
2.ตำแหน่ถอยกลับหลังจากขุดเจาะเสร็จ (Return point level) ได้แก่ G98 หรือ G99

3. รูปแบบการขุดเจาะ (Drilling Mode)

4.2.7.1 การขุดเจาะกระแทกเป็นจังหวะ (High-Speed Peck-Drilling Canned Cycle , G73) ขั้นตอนการทำงานของ G73 จะเป็นดังรูป และเมื่อเจาะเสร็จจะถอยกลับไป R level หรือ Initial level ขึ้นอยู่กับว่าจะใช้ร่วม G98 หรือ G99

รูปแบบ G73 X__ Y__ Z__ Q__ R__ F__ *

* โดยที่ Q หมายถึง ระยะ Cut-in



รูปที่ 4.12

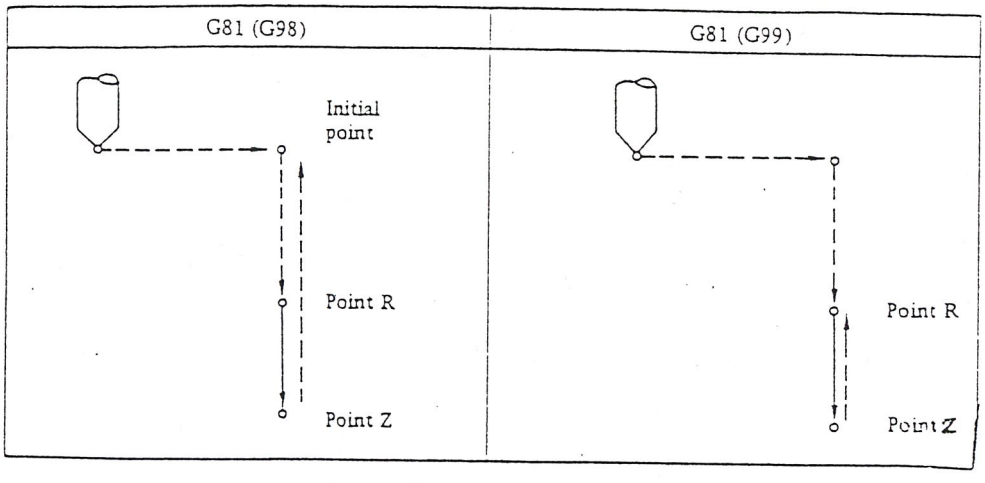
เช่น G91 G98 G73 X10.0 Y15.0 Z-1.72 Q0.61 R-6.3 F30 *

Q = ระยะ cut-in

F = อัตราเร็วการขุด

4.2.7.2 การขุดเจาะเป็นจังหวะ คว้านรูเป็นจุดๆ (Drilling Canned Cycle , SpoyBoring Cycle , G81) ขั้นตอนการทำงาน จะเคลื่อนที่ด้วยความเร็วสูงสุดไปที่ R point และเจาะรูด้วย Feed rate และเมื่อเจาะเสร็จ จะถอยกลับไปที่ R level หรือ Initial level ขึ้นอยู่กับว่าจะใช้ร่วมกับ G99 หรือ G98 ด้วยความเร็วสูงสุด (Rapid traverse)

รูปแบบ G81 X__ Y__ Z__ R__ F__ *



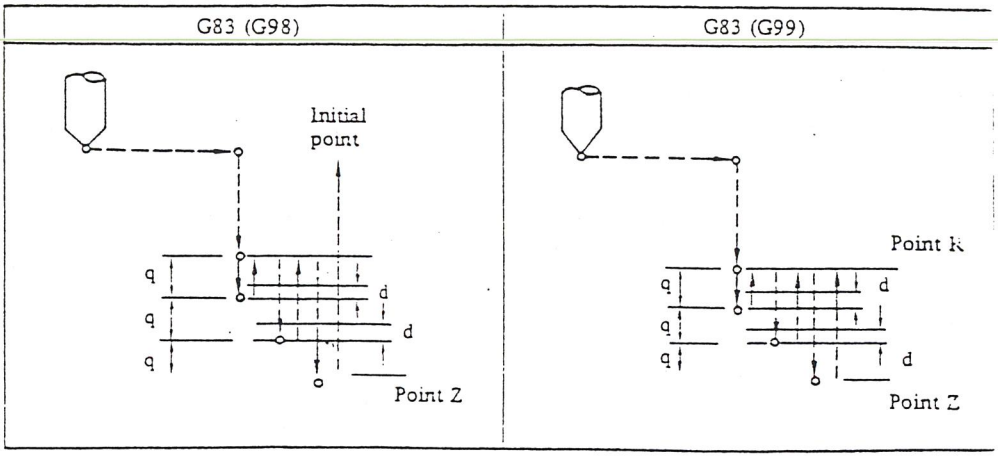
รูปที่ 4.13

4.2.7.3 การขุดเจาะเป็นจังหวะ การคว้านผายปากรู (Drilling Cycle , Counter Boring Cycle , G82) การทำงานจะเหมือนกับ G81 แต่จะหยุด (เป็นเวลา P sec) เมื่อถึงจุดล่างสุดของรู และถอยกลับด้วยความเร็วสูงสุด

รูปแบบ G82 X__ Y__ Z__ R__ P__ F__ *

4.2.7.4 การขุดเจาะกดเป็นจังหวะ (Peck drilling Canned Cycle , G83) ขั้นตอนการขุดเจาะจะคล้ายกับ G73 แต่จะแตกต่างตรงที่ หลังจาก Cut-in ระยะ Q จะถอยกลับมาที่ R level ทุกครั้ง ซึ่งต่างกับ G73 ซึ่งจะถอยกลับมาเป็นระยะ D เท่านั้น

รูปแบบ G83 X__ Y__ Z__ R__ Q__ F__ *



รูปที่ 4.14

4.2.7.5 การคว้านรูเป็นจังหวะ (Boring Cycle , G85) การทำงานจะคล้ายกับ G81 แต่จะถอยกลับด้วยความเร็ว Feed rate

4.2.7.6 การคว้านรูเป็นจังหวะ (Boring Cycle , G86) การทำงานจะคล้าย G81 แต่ดอกัดจะหยุดเมื่อถึงจุดล่างสุดของรู และจะถอยกลับด้วยความเร็วสูงสุด

4.2.7.7 การคว้านรูเป็นจังหวะ , เจาะคายเศษ (Boring Cycle , G89) การทำงานจะ คล้าย G85 แต่จะหยุด(เป็นเวลา P sec) เมื่อถึงจุดล่างสุดของรูและถอยกลับด้วยความเร็ว Feed rate

4.3 ชุดคำสั่ง M (M Function หรือ Miscellaneous Function)

ชุดคำสั่ง M จะใช้ควบคุม Cutter หรือ Spindle ชุดคำสั่งในโครงงานนี้ที่ได้จัดทำ (มาตรฐานจาก Fanuc)

คำสั่งรหัส	ความหมาย
M02	จบโปรแกรม
M03	ให้ Cutter หมุนตามเข็มนาฬิกา
M04	ให้ Cutter หมุนทวนเข็มนาฬิกา
M05	ให้ Cutter หยุดหมุน

นอกจากชุดคำสั่ง G และ M แล้ว ยังมีคำสั่งย่อย ซึ่งใช้ร่วมกับชุดคำสั่ง G และ M ได้แก่

4.4 คำสั่ง F (Feed Function)

จะเป็นตัวกำหนดความเร็วของการ Cut-in (Cutting Feed Rate) ซึ่งค่าป้อนเข้าไปจะเป็น inch / minute หรือ mm. / minute

4.5 คำสั่ง S (Spinle Speed Function)

เป็นคำสั่งกำหนดความเร็วของ Spinle มีหน่วยเป็น rpm.

บทที่ 5
ทฤษฎีทางคณิตศาสตร์ที่นำมาประยุกต์ใช้

ในการเขียนโปรแกรมควบคุมเครื่องกัด CNC แนวดิ่งบนพื้นฐานไมโครคอมพิวเตอร์นี้ ได้มีการนำเอาทฤษฎีทางคณิตศาสตร์มาประยุกต์ใช้เพื่อแก้ปัญหาในหลาย ๆ ด้าน ซึ่งสามารถแบ่งเป็นหัวข้อย่อยๆได้ดังนี้

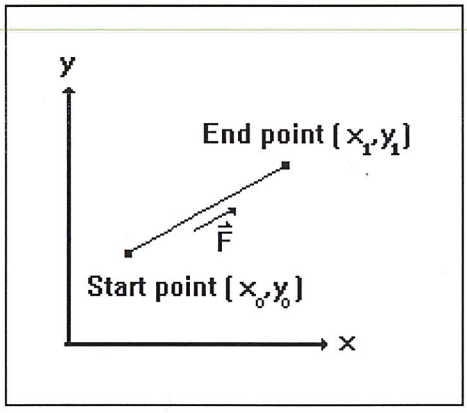
ทฤษฎีในการคำนวณเกี่ยวกับการเคลื่อนที่

1 . Linear Interpolation

เมื่อมี Straight cut linear interpolation (G01) ส่วนควบคุมจะควบคุมแกนของเครื่องจักร 2 แกนหรือมากกว่านั้นพร้อมๆกัน ในส่วนการตัดเชิงมุมส่วนควบคุมจะใช้ข้อมูลที่ได้รับการโปรแกรมเข้ามาเก็บไว้แล้วเพื่อคำนวณ องศาหรือความชันของการตัดชิ้นส่วนของเส้นตรง ความยาวที่เปลียนแปลงจากจุดเริ่มต้น จนถึงจุดสุดท้ายจะเป็นตัวกำหนด การแบ่งเส้นและความชันของแต่ละแกน เพื่อทำการควบคุมการเคลื่อนที่ของ Cutter ให้เคลื่อนที่เสมือนกับ การเคลื่อนที่ไปในทิศทางเดียว ในโครงงานนี้กำหนดให้ Linear interpolation เคลื่อนที่ในระนาบเดียวเท่านั้น

การใช้งานอย่างอื่นของ Linear interpolation คือใช้ในการประมาณเส้นโค้ง หรือวงกลม (Circular interpolation) ซึ่งจะกล่าวถึงวิธีการประมาณเส้นโค้งในช่วงต่อไป

การคำนวณใน Linear interpolation



รูปที่ 5.1 การเคลื่อนที่แบบ Linear Interpolation

F - อัตราป้อนในแนวการเคลื่อนที่

F_x - อัตราป้อนในแนวแกน X

F_y - อัตราป้อนในแนวแกน Y

$$F_x = \frac{F(\text{ระยะทางที่เคลื่อนที่ในแนวแกนx})}{(\text{ระยะทางที่เคลื่อนที่ทั้งหมด})} = \frac{F\,(x_1-x_0)}{\sqrt{(x_1-x_0)^2+(y_1-y_0)^2}}$$

$$F_y = \frac{F(\text{ระยะทางที่เคลื่อนที่ในแนวแกนy})}{(\text{ระยะทางที่เคลื่อนที่ทั้งหมด})} = \frac{F\,(y_1-y_0)}{\sqrt{(x_1-x_0)^2+(y_1-y_0)^2}}$$

2 . Circular Interpolation

Circular interpolation เป็นความสามารถในการตัดส่วนโค้งของวงกลม,จำนวนที่ตัดแปรผันตามขนาดของส่วนโค้งซึ่งขึ้นอยู่กับรัศมีของส่วนโค้งส่วนควบคุมจะคำนวณทิศทางของ Cutter จากข้อมูลที่โปรแกรมมา

ข้อมูลที่ต้องการได้แก่

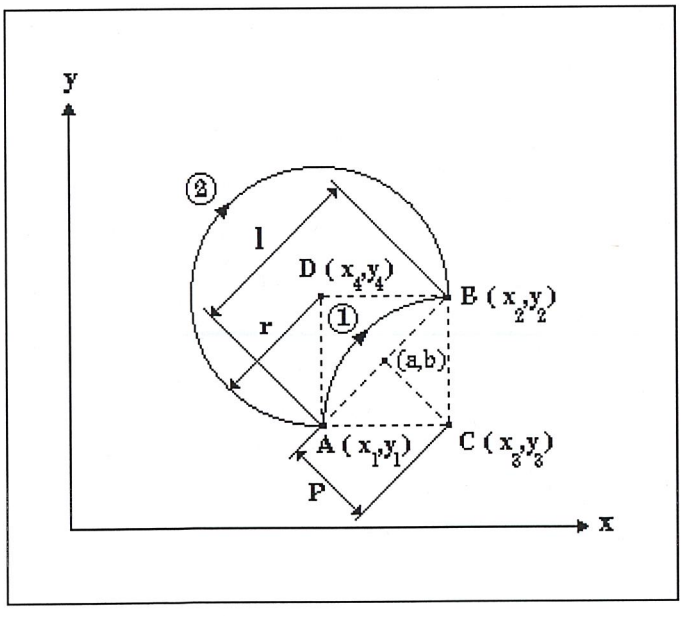
1. ทิศทางของการเคลื่อนที่ (CW หรือ CCW)
2. จุดเริ่มต้นของส่วนโค้ง (arc)
3. จุดสิ้นสุดของส่วนโค้ง (arc)
4. จุดศูนย์กลางของส่วนโค้ง (arc)

ข้อมูลที่ทราบจากการโปรแกรมแบบ Radius method (เป็นการโปรแกรมแบบที่ใช้ในโครงงานนี้)ได้แก่

1. ทิศทาง
2. จุดเริ่มต้น
3. จุดสิ้นสุด
4. รัศมี

ต้องนำมาคำนวณหา จุดศูนย์กลางของส่วนโค้ง

วิธีการคำนวณเพื่อหาจุดศูนย์กลางส่วนโค้ง



รูปที่ 5.2 การเคลื่อนที่แบบ Circular Interpolation

$$a = (x1+x2)/2$$

$$b = (y1+y2)/2$$

$$l = \sqrt{(x2-x1)2+(y2-y1)2}$$

$$n = -(x2 - x1)/(y2 - y1)$$

$$p = \sqrt{R2 - (l/2)2}$$

จะได้

$$x3 = a + \sqrt{p2/(1+n2)}$$

$$y3 = b + (n\sqrt{p2/(1+n2)}$$

และ

$$x4 = a - \sqrt{p2/(1+n2)}$$

$$y4 = b - (n*\sqrt{p2/(1+n2)}$$

ในการเลือกว่าจะใช้จุด (x3,y3) หรือ (x4,y4) เป็นจุดศูนย์กลางของวงกลมนั้นจะต้องพิจารณาค่ารัศมีของวงกลมว่าเป็น + หรือ -

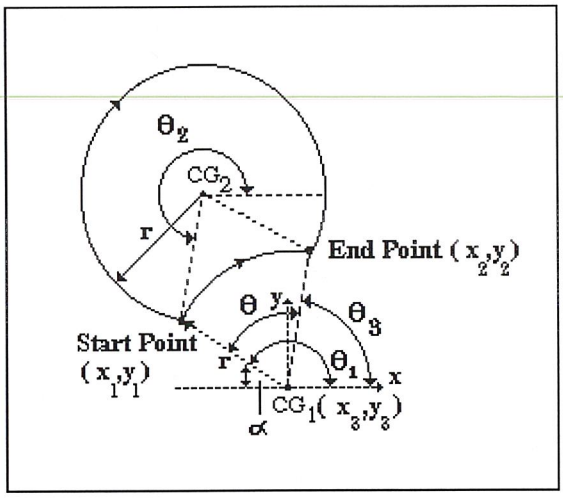
- ถ้าเป็น + จะใช้จุด (x3,y3) เป็นจุดศูนย์กลาง
- ถ้าเป็น - จะใช้จุด (x4,y4) เป็นจุดศูนย์กลาง

โดยจะใช้ผลของการ Cross Vector ระหว่าง Vector 2 คู่ คือ AB*AC กับ AB*AD เป็นตัวชี้ คือ

- ถ้ารัศมีของวงกลมเป็น + ให้ใช้จุดศูนย์กลางที่ให้ผลของการ Cross Vector มีค่าสัมประสิทธิ์ของ k เป็น -
- ถ้ารัศมีของวงกลมเป็น - ให้ใช้จุดศูนย์กลางที่ให้ผลของการ Cross Vector มีค่าสัมประสิทธิ์ของ k เป็น +

ก็จะได้จุดศูนย์กลางของวงกลมตามที่เราต้องการ หลังจากที่เราได้ข้อมูลที่ต้องการนำมาทำการประมาณจุดบนโค้งด้วยการใช้ Linear interpolation

การหามุมเริ่ม



รูปที่ 5.3 การหามุมเริ่ม มุมสุดท้าย และมุมเคลื่อนที่

จากรูป 2.47 มุมเริ่ม คือมุม θ_1 และ θ_2 ของจุดศูนย์กลาง CG_1 และ CG_2 ตามลำดับ โดยการหามุมเริ่มนี้จะมีวิธีการหาเหมือนกัน ในที่นี้จะยกตัวอย่างการหามุมเริ่ม θ_1 ของจุดศูนย์กลาง CG_1 ดังมีขั้นตอนดังต่อไปนี้

1. ตั้งแกน X , Y ที่จุด CG_1 โดยให้จุด Origin อยู่ที่จุด CG_1
2. มุมเริ่มจะวัดจากแกน X ในทิศทวนเข็มนาฬิกา
3. หาค่ามุม α จาก $\alpha = \tan^{-1} (y1 - y3)/(x1 - x3)$
4. ตรวจสอบว่าจุด Start point อยู่ในควอดแดรนท์ใดของแกน X,Y ที่สร้างขึ้น โดยการดูเครื่องหมายของ (x1 - x3) และ (y1 - y3)
5. ถ้าจุด start point อยู่ควอดแดรนท์ที่ 1 ; $\theta_1 = \alpha$

   ถ้าจุด start point อยู่ควอดแดรนท์ที่ 2 ; $\theta_1 = \pi - \alpha$

   ถ้าจุด start point อยู่ควอดแดรนท์ที่ 3 ; $\theta_1 = \pi + \alpha$

   ถ้าจุด start point อยู่ควอดแดรนท์ที่ 4 ; $\theta_1 = 2\pi - \alpha$

   ก็จะได้ค่าของมุมเริ่มตามต้องการ

การหามุมสุดท้าย

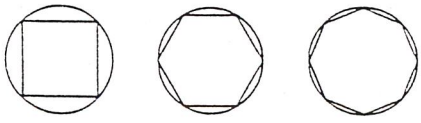
จากรูปที่ 2.47 มุมสุดท้าย คือมุม θ_3 ซึ่งวิธีการหาก็จะคล้ายกับการหามุม θ_1 เพียงแต่เปลี่ยนค่ามุม α เป็น $\alpha = \tan^{-1} (y2 - y3)/(x2 - x3)$ และใช้จุด end point แทน

การหามุมเคลื่อนที่

จากรูปที่ 2.47 มุมเคลื่อนที่คือมุม θ ซึ่งมีค่าเท่ากับ

$$\theta = \theta_1 - \theta_3$$

Linear approximations of A Circle



รูปที่ 5.4 Linear Approximation of A Circle

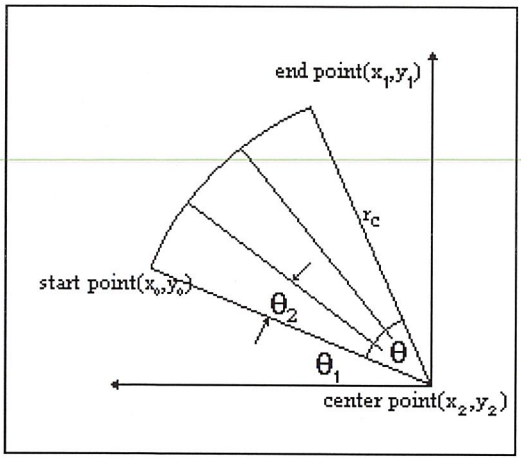
จำนวนช่องที่แบ่ง จะใช้ความยาวของส่วนโค้งและ cord เป็นตัวกำหนดโดยที่chord จะให้ยาว = 0.01 mm

$$\text{จำนวนช่องที่แบ่ง(n)} = \frac{\text{ความยาวของส่วนโค้ง}}{\text{ความยาวของ chord.}}$$

$$n = \theta R / 0.01$$

* โดย n จะต้องเป็นจำนวนเต็มที่ปัดขึ้นเสมอ *

$\therefore$ องศาที่เปลี่ยนไปในการเคลื่อนที่แต่ละchord $\theta_2 = \theta/n$



รูปที่ 5.5 Use of chord segments to approximate arc

และ $x_{1,0} = x_0$ $y_{1,0} = y_0$

$n = 1$ $x_{1,1} = x_2 + \cos(1\theta_2+\theta_1)$ $y_{1,1} = y_2 + \sin(\theta_2+\theta_1)$

$n = 2$ $x_{1,2} = x_2 + \cos(2\theta_2+\theta_1)$ $y_{1,2} = y_2 + \sin(2\theta_2+\theta_1)$

$n = 3$ $x_{1,3} = x_2 + \cos(3\theta_2+\theta_1)$ $y_{1,3} = y_2 + \sin(3\theta_2+\theta_1)$

. .

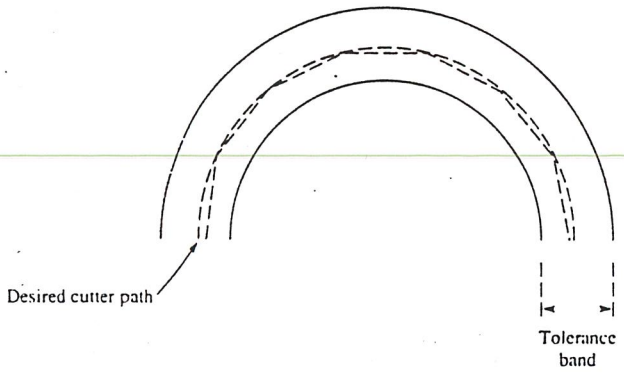
. .

. .

$x_{1,n} = x_1$ $y_{1,n} = y_1$

$$x_{1,n} = x_3 + \cos(n\theta_2+\theta_1) \qquad n = 1.....n$$

$$y_{1,n} = y_3 + \sin(n\theta_2+\theta_1) \qquad n = 1.....n$$



รูปที่ 5.6 Circular Interpolation Using Chord Segments Method

วิธีนี้เรียกว่า **"Chord Segments Method"**

3. การคำนวณการชดเชยขนาด

การเขียนโปรแกรมงานวัดตามเส้นรอบรูปในแนวเส้นตรงและโค้งจะมีการทำงานพร้อมกัน 2 แนวแกน ดังนั้นในการกำหนดจุดโคออดิเนทต่างๆจะต้องคำนวณเผื่อขนาดดอกกัดด้วย ทำให้โปรแกรมมีความยุ่งยาก

ระบบควบคุมในการทำงานของเครื่องจักร CNC จะสามารถคำนวณเผื่อขนาดของดอกกัดได้โดยอัตโนมัติ ซึ่งเรียกว่า การชดเชยขนาดดอกกัดตามเส้นรอบรูปในการโปรแกรมงานกัด จะใช้คำสั่ง

"G40" = การยกเลิกการชดเชยรัศมี
"G41" = การชดเชยรัศมีทางด้านซ้าย
"G42" = การชดเชยรัศมีทางด้านขวา

และสามารถเขียนโปรแกรมจากขนาดกำหนดในแบบงานได้โดยตรงและเปลี่ยนขนาดของ ดอกกัดแทนทำให้สามารถใช้ดอกกัดได้หลายขนาดโดยใช้โปรแกรมเดิม

เส้นตรงตัดกับเส้นตรง

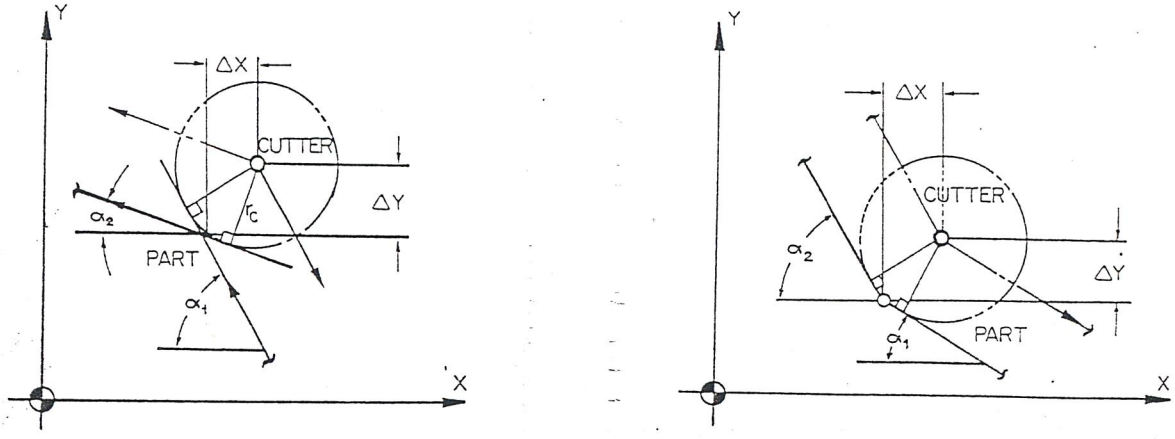
ตัวอย่าง

G41 D__ * (D คือ ขนาดเส้นผ่านศูนย์กลางดอกกัด)

G01 X__ Y__ F__ *

G01 X__ Y__ F__ *

โดยขนาดของตำแหน่งที่เปลี่ยนไปสามารถดูได้จากรูปที่ 5.7



รูปที่ 5.7

จากรูปที่ 5.7 จะได้ว่า

$$\Delta x = r_c * \frac{\sin\left(\frac{\alpha_2+\alpha_1}{2}\right)}{\cos\left(\frac{\alpha 2-\alpha 1}{2}\right)} \qquad \Delta y = r_c * \frac{\cos\left(\frac{\alpha_2+\alpha_1}{2}\right)}{\cos\left(\frac{\alpha 2-\alpha 1}{2}\right)}$$

เมื่อ $r_c = D/2$

เมื่อได้ขนาดของตำแหน่งที่เปลี่ยนไปแล้วต่อไปก็จะพิจารณาเครื่องหมายโดยจะดูจากทิศของ Vector (j,k) คือ Vector จากจุดเริ่มถึงจุดสุดท้ายในเส้นทางเดิน และลักษณะการชดเชย

ขนาดว่าเป็นด้านซ้าย (G41) หรือ ขวา (G42) ซึ่งจะแบ่งได้เป็นกรณี ต่างๆ ดังนี้ (ดูรูปที่ 5.8, 5.9 ประกอบ)

กรณีที่ 1 j + , k +

	G41	G42
Δx	−	+
Δy	+	−

กรณีที่ 2 j + , k -

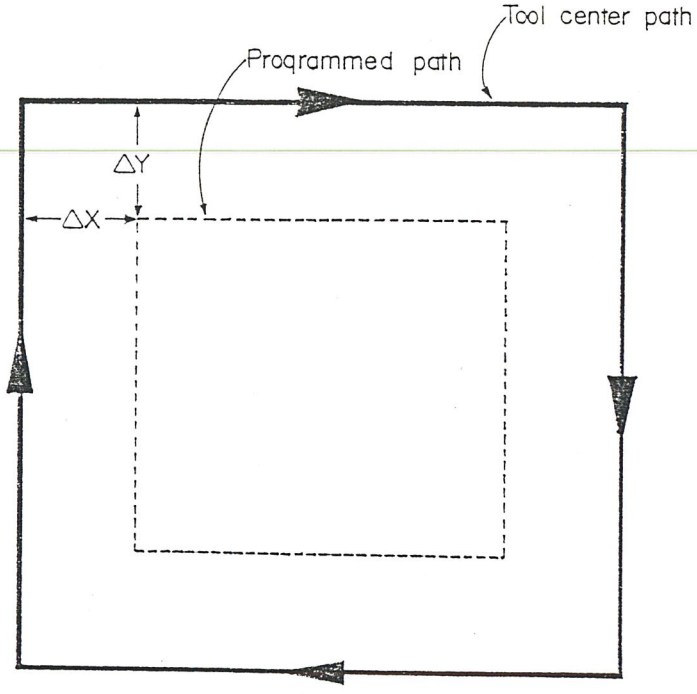
	G41	G42
Δx	+	−
Δy	+	−

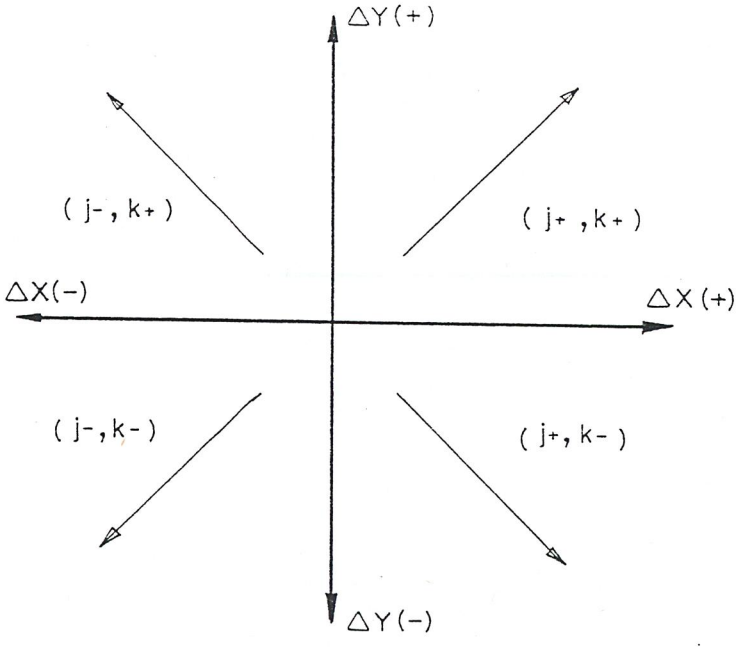
กรณีที่ 3 j - , k +

	G41	G42
Δx	−	+
Δy	−	+

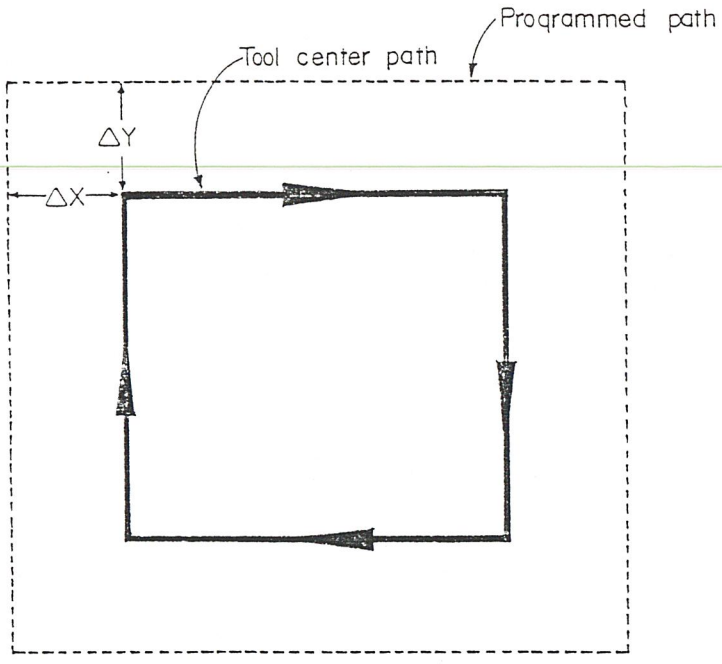
กรณีที่ 4 j - , k -

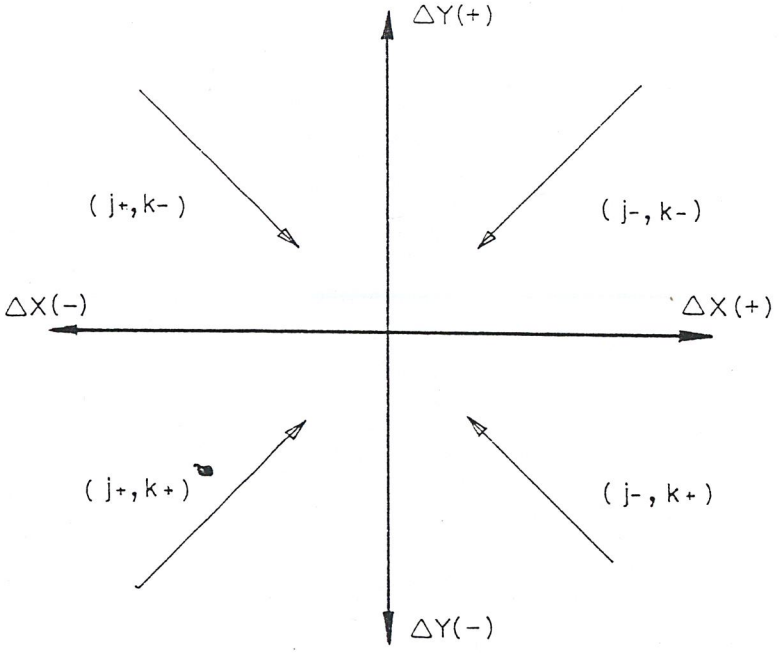
	G41	G42
Δx	+	−
Δy	−	+





รูปที่ 5.8





รูปที่ 5.9

เส้นตรงตัดกับเส้นโค้ง

ทฤษฎีที่จะนำมาใช้จะเป็นการหาจุดตัดกันของสมการเส้นตรงและสมการวงกลม

ตัวอย่าง

G41 D__ * (D คือ ขนาดเส้นผ่านศูนย์กลางดอกกัด)

G01 X__ Y__ F__ *

G02 X__ Y__ R__ F__ *

สมการเส้นตรง $Y = AX + B$ ---------- **(1)**

สมการวงกลม(ส่วนโค้ง) มีจุดศูนย์กลางที่ (X_1 , Y_1)

$$(X-X_1)^2 + (Y-Y_1)^2 = R^2 \quad \text{---------- (2)}$$

นำค่า Y จากสมการที่ **(1)** มาแทนในสมการที่ **(2)** จะได้

$$(X-X_1)^2 + (AX + B - Y_1)^2 = R^2$$

$$(X^2-2XX_1+X_1^2) + (A^2X^2+2AX(B-Y_1)+(B-Y_1)^2) - R^2 = 0$$

$$(1+A^2)X^2 + 2(A(B-Y_1)-X_1)X + (B-Y_1)^2 - R^2 = 0$$

จะได้ผลเฉลยของ **X** เป็น

$$X_{1,2} = \frac{-2(A(B-Y_1)-X_1) \pm \sqrt{(A(B-Y_1)-X_1)^2 - 4(1+A^2)((B-Y_1)^2 - R^2)}}{2(1+A^2)}$$

ได้ค่า **X** สองค่า คือ X_1 และ X_2

นำ **X** สองค่านี้ไปแทนในสมการเส้นตรง $Y = AX+B$ จะได้ Y_1 และ Y_2

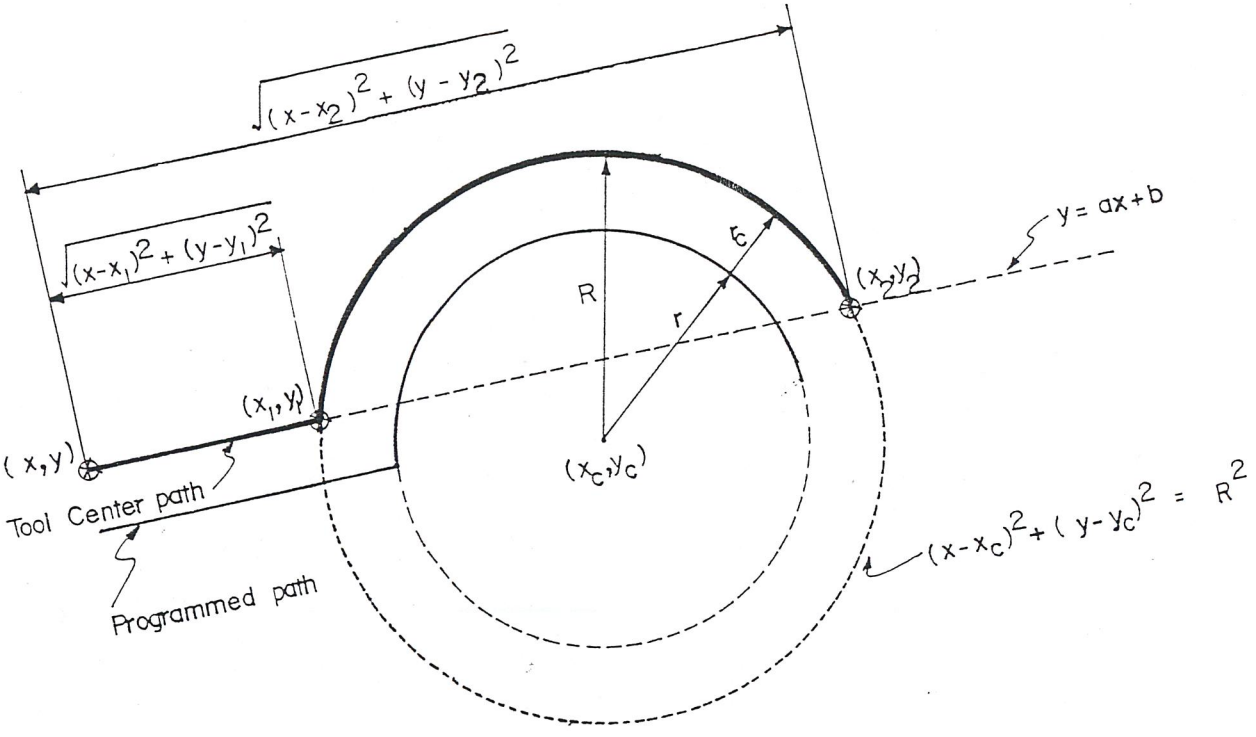
ในการที่จะเลือกว่าจุดใดระหว่าง (X_1,Y_1) และ (X_2,Y_2) ที่จะเป็นจุดที่นำไปใช้งาน จะพิจารณาจากระยะห่างระหว่างจุดทั้งสองนั้นกับจุดเริ่มต้นของ PATH (ดูรูปประกอบ) โดยจุดที่จะนำไปใช้งานจะมีระยะห่างน้อยกว่า นั่นคือ

ถ้า $\sqrt{(X-X_1)^2+(Y-Y_1)^2} < \sqrt{(X-X_2)^2+(Y-Y_2)^2}$

จุดที่นำไปใช้เป็นจุด Compensate คือจุด (X_1,Y_1)

ถ้า $\sqrt{(X-X_1)^2+(Y-Y_1)^2} > \sqrt{(X-X_2)^2+(Y-Y_2)^2}$

จุดที่นำไปใช้เป็นจุด Compensate คือจุด (X_2,Y_2)



รูปที่ 5.10 กรณีเส้นตรงตัดกับส่วนโค้ง

ส่วนโค้งตัดกับส่วนโค้ง

ทฤษฎีที่จะนำมาใช้จะเป็นการหาจุดตัดกันของสมการวงกลมสองวง

ตัวอย่าง

G41 D__ * (D คือ ขนาดเส้นผ่านศูนย์กลางดอกกัด)

G01 X__ Y__ F__ *

G01 X__ Y__ F__ *

สมการวงกลม(ส่วนโค้ง) มีจุดศูนย์กลางที่ (X_1, Y_1)

$$(X-X_1)^2 + (Y-Y_1)^2 = R_1^2 \quad \text{---------- (1)}$$

สมการวงกลม(ส่วนโค้ง) มีจุดศูนย์กลางที่ (X_2, Y_2)

$$(X-X_2)^2 + (Y-Y_2)^2 = R_2^2 \quad \text{---------- (2)}$$

นำสมการที่ (1) - (2) จะได้สมการเส้นตรงที่ผ่านจุดตัดกันของวงกลมทั้งสอง

$$2(X_2-X_1)X + 2(Y_2-Y_1)Y + Y_1^2 - Y_2^2 + R_2^2 - R_1^2 = 0$$

$$Y = \frac{(X_1-X_2)}{(Y_2-Y_1)} X + \frac{(X_2^2 - X_1^2 + Y_2^2 - Y_1^2 + R_1^2 - R_2^2)}{2(Y_2-Y_1)} \quad \text{---------- (3)}$$

นำค่า Y จากสมการที่ (3) ไปแทนใน (1) หรือ (2) จะได้ว่า

$$X_{1,2} = \frac{-2(AB-AY_1-X_1) \pm \sqrt{4(AB-AY_1-X_1)^2 - 4(1+A^2)(B^2+X_1^2+Y_1^2-R_1^2-2BY)}}{2(1+A^2)}$$

$$Y_{1,2} = AX_{1,2} + B$$

โดย $A = (X_2-X_1) / (Y_2-Y_1)$

$B = Y_2^2 - Y_1^2 + R_1^2 - R_2^2$

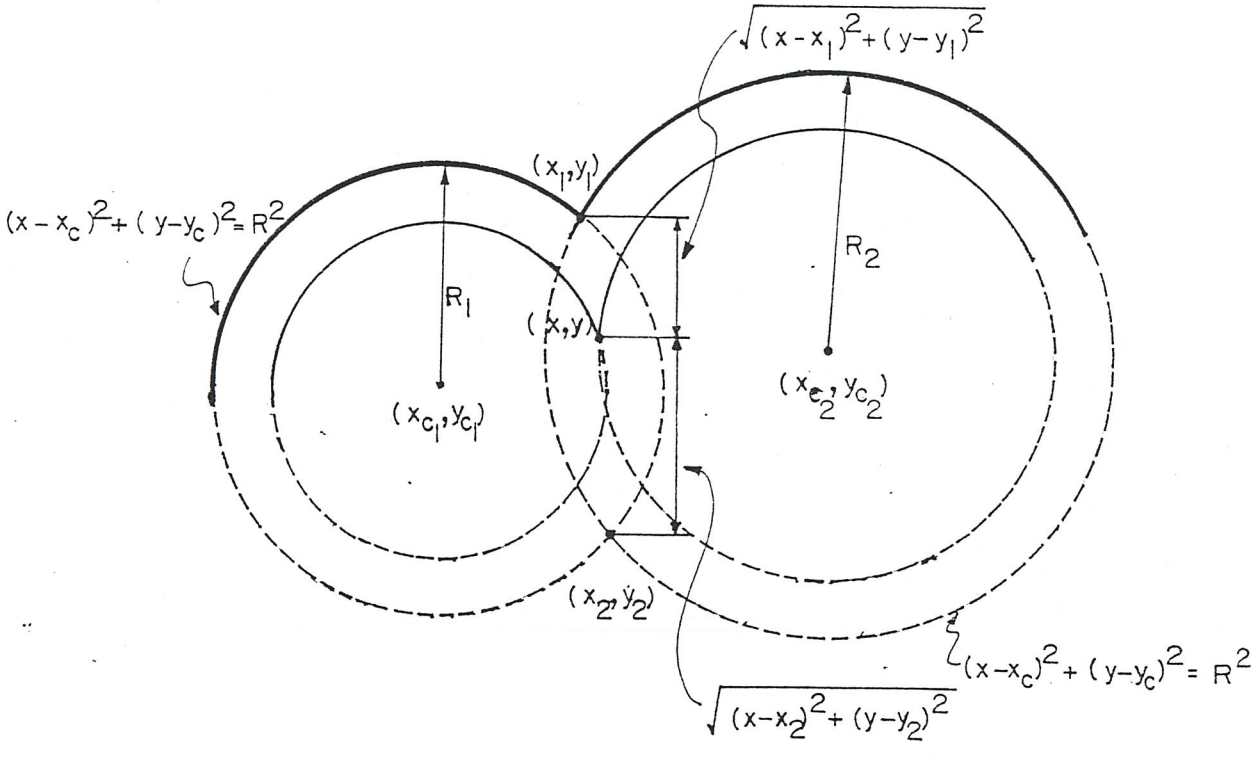
จะได้จุดตัดของวงกลมทั้งสองวงนี้คือจุด (X_1,Y_1) และ (X_2,Y_2) ในการจะเลือกว่าจะนำจุดใดไปใช้จะพิจารณาจากระยะห่างของจุดตัดของวงกลมทั้งสองใน Programmed Path ไปยังจุดทั้งสอง ถ้าระยะใดมีค่าน้อยกว่าก็ให้นำจุดนั้นมาคิดเป็นจุด Compensate (ดูรูปประกอบ)

ถ้า $\sqrt{(X-X_1)^2+(Y-Y_1)^2} < \sqrt{(X-X_2)^2+(Y-Y_2)^2}$

จุดที่นำไปใช้เป็นจุด Compensate คือจุด (X_1,Y_1)

ถ้า $\sqrt{(X-X_1)^2+(Y-Y_1)^2} > \sqrt{(X-X_2)^2+(Y-Y_2)^2}$

จุดที่นำไปใช้เป็นจุด Compensate คือจุด (X_2,Y_2)



รูปที่ 5.11 กรณีส่วนโค้งตัดกับส่วนโค้ง

บทที่ 6

การออกแบบและวิธีการใช้โปรแกรม

6.1 การออกแบบโปรแกรม

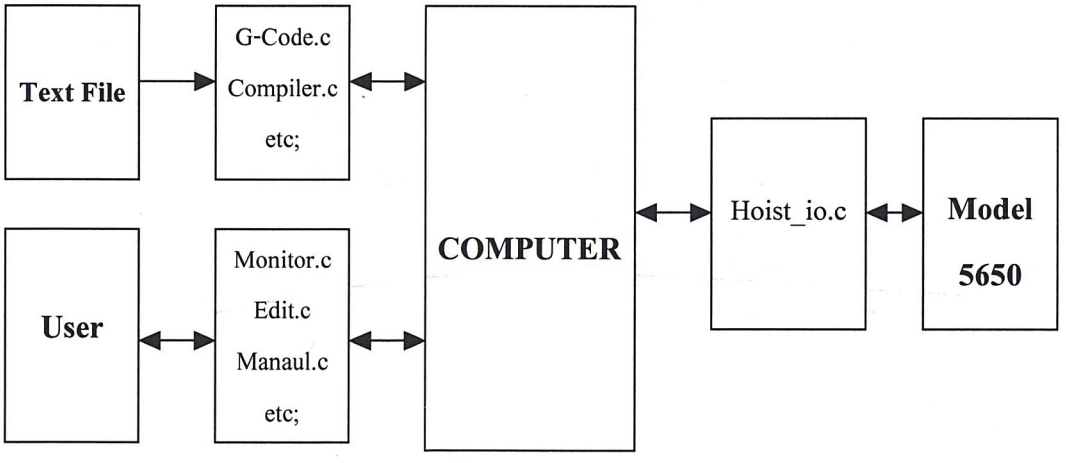
ในการออกแบบโปรแกรมสำหรับการควบคุมเครื่องกัดแนวดิ่งซีเอ็นซีนั้น เราได้ออกแบบโดยใช้ภาษา C ในการเขียน และทำการ Run บน Borland C++ ซึ่งตัวโปรแกรมที่ทำการออกแบบจะแบ่งเป็นส่วนสำคัญ ๆ 3 ส่วนดังต่อไปนี้

1. ส่วนที่ติดต่อระหว่างผู้ใช้กับโปรแกรม (User Interface) ได้แก่ โมดูล Monitor.C โมดูล Edit.C โมดูล Manual.C

2. ส่วนที่เชื่อมต่อระหว่างคอมพิวเตอร์กับการ์ดควบคุม ได้แก่ โมดูล Host_io.C

3. ส่วนการติดต่อสื่อสารกับข้อมูลภายนอกที่อยู่ในรูปของ Text File ได้แก่ โมดูล G-Code.C โมดูล Compiler.C

นอกจากนี้ก็เป็นส่วนประกอบปลีกย่อยซึ่งช่วยในการทำงานของโปรแกรม โดยทั้งหมดทุกส่วนที่กล่าวมาสามารถแสดงดังในรูป 6.1



รูป 6.1 แสดงส่วนการทำงานของโปรแกรม

6.2 รายระเอียดในแต่ละส่วนของโปรแกรม

6.2.1.ส่วนที่ติดต่อกับผู้ใช้โปรแกรม

สำหรับในส่วนที่ติดต่อกับผู้ใช้โปรแกรมนั้น ในการออกแบบโปรแกรมในแต่ละส่วน มีการออกแบบโดยแบ่งออกเป็นโหมดการทำงานต่าง ๆ ดังนี้

1. โหมดการเขียนและแก้ไขโปรแกรม (Edit mode)

2. โหมดอัตโนมัติ (Automatic mode)

3. โหมดความจำ (Memory mode)

4. โหมดแบบบังคับเอง (Manual mode)

โดยแต่ละโหมดดังกล่าวข้างต้น มีรายละเอียดของฟังก์ชันต่างๆ ดังนี้

1. โหมดการเขียนและแก้ไขโปรแกรม (Edit mode)

ในโหมดนี้เป็นส่วนของการเขียนและแก้ไขโปรแกรมเพื่อควบคุมเครื่องกัดแนวดิ่งซีเอ็นซีให้ทำงานตามที่ต้องการ โดยการเขียนโปรแกรม ต้องอ้างอิงกับ G-code และ M-code มาตรฐาน ซึ่งในโหมดดังกล่าวนี้มีความสามารถในการทำงานดังนี้

- สามารถเขียนข้อมูล (Newprogram) โดยในระหว่างการเขียนข้อมูลจะมีการเขียนเลขบรรทัด และรูปแบบคำสั่งของ G-code หรือ M-code ที่เลือกใช้ให้โดยอัตโนมัติ เพื่อสะดวกต่อการเขียนข้อมูล

- สามารถเรียกข้อมูลที่มีอยู่แล้วขึ้นมาทำการแก้ไข (Edit) ได้ โดยในระหว่างการทำการแก้ไขข้อมูลจะมีรูปแบบคำสั่งของ G-code หรือ M-code ที่จะทำการแก้ไขให้เพื่อความสะดวกในการแก้ไข

- สามารถลบข้อมูล (Delete) ในตำแหน่งบรรทัดที่ต้องการได้ โดยการเลื่อนตัวชี้ขึ้นหรือลงเพื่อเลือกบรรทัดที่ต้องการ และหลังจากทำการลบข้อมูลในบรรทัดนั้นแล้ว ตัวเลขแสดงบรรทัดของบรรทัดต่อไปจะทำการเลื่อนขึ้นมาให้โดยอัตโนมัติ

- สามารถแทรกข้อมูล (Insert) ในตำแหน่งบรรทัดที่ต้องการเพิ่มข้อมูลเข้าไปได้ โดยการเลื่อนตัวชี้ขึ้นหรือลงเพื่อเลือกบรรทัดที่ต้องการ และหลังจากทำการแทรกบรรทัดแล้ว ตัวเลขในบรรทัดต่อ ๆ ไปจะทำการเลื่อนลงให้โดยอัตโนมัติ

- สามารถเลื่อนตัวชี้ขึ้นไปหนึ่งบรรทัด(Arrow up) หรือ หนึ่งหน้าจอ (Page up) เพื่อใช้ในการแก้ไขหรือตรวจสอบโปรแกรมได้

- สามารถเลื่อนตัวชี้ลงไปหนึ่งบรรทัด (Arrow down) หรือ หนึ่งหน้าจอ (Page down) เพื่อใช้ในการแก้ไขหรือตรวจสอบโปรแกรมได้

- สามารถเก็บข้อมูล (Save) ที่เขียนหรือแก้ไข ลงในเครื่อง Computer ได้

โปรแกรมสำหรับโหมดนี้จะอยู่ในโมดูล Edit.C ซึ่งมีผังการทำงาน (Flow Chart) ดังรูปที่ 1 และ 2 ในภาคผนวก ก.

2. โหมดอัตโนมัติ (Automatic mode)

ในโหมดดังกล่าวเป็นส่วนของการอ่านข้อมูล การแปลข้อมูล และการทำงานตามข้อมูลที่ได้รับ โดยโปรแกรมจะทำการอ่านข้อมูลที่มาในรูปเทกซ์ไฟล์ทีละบรรทัด แปลความหมายและปฏิบัติตามคำสั่งทีละบรรทัดจนกว่าจะทำเสร็จ และจะวนรอบกการทำงานเช่นนี้ไปเรื่อยๆ จนจบไฟล์ข้อมูล ซึ่งในโหมดนี้จะมีความสามารถในการทำงานดังนี้

- สั่งให้โปรแกรมทำงาน (Cycle start) โปรแกรมจะทำงานทั้งหมดและติดต่อกับ ฮาร์ดแวร์เพื่อควบคุมการทำงานให้สอดคล้องกับโปรแกรม

- สั่งให้โปรแกรมหยุดทำงาน (Cycle stop) เพื่อการหยุดโปรแกรมและส่งสัญญาณไปบอกระบบฮาร์ดแวร์ในการควบคุมมอเตอร์หยุดการทำงาน ถ้าจะเริ่มการทำงานใหม่ก็จะต้องโหลดโปรแกรมเข้าไปใหม่

- สั่งให้โปรแกรมทำงานทีละช่วง (Single block) โดยการกำหนดบรรทัดเริ่มต้นและสิ้นสุดให้แก่โปรแกรม ซึ่งโปรแกรมส่วนนี้จะมีประโยชน์ในการที่บางทีเราอาจต้องการทำการ machine ชิ้นงานแค่บางส่วนเท่านั้น ทำให้ไม่ต้องรอให้โปรแกรมทำงานทั้งหมดก็ได้

โปรแกรมสำหรับโหมดนี้จะอยู่ในโมดูล Monitor.C ซึ่งมีผังการทำงาน (Flow Chart) ดังรูปที่ 4 ในภาคผนวก ก.

3. โหมดความจำ (Memory mode)

ในโหมดดังกล่าวเป็นส่วนของการปรับเปลี่ยนค่าต่าง ๆ ของความเร็วทั้งในระหว่างการทำการ Machine และเวลาบังคับเอง โดยในโหมดนี้จะมีความสามารถต่าง ๆ ดังนี้

- การเปลี่ยนค่า Feed rate คือการเปลี่ยนค่าความเร็วในการเดินของหัวกัดในการ Machine ชิ้นงานได้ สำหรับในกรณีที่ Code ที่นำมาทำการ Machine นั้นไม่มีการตั้งค่า Feed rate เอาไว้

- การเปลี่ยนค่า Spindle rate คือการเปลี่ยนค่าความเร็วในการหมุนของหัว Spindel ได้ ใช้ทั้งในโหมดอัตโนมัติ กรณีที่ Code ที่นำมาทำการ Machine นั้นไม่มีการตั้งค่า Spindel rate เอาไว้ และในโหมดแบบบังคับเอง เพื่อตั้งค่า Spindel rate ตามความต้องการของผู้ใช้โปรแกรม

- การเปลี่ยนค่า Jog speed คือการเปลี่ยนค่าความเร็วในการเดินของหัวกัดในโหมดการบังคับเองโดยผู้ใช้โปรแกรม

โปรแกรมสำหรับโหมดนี้จะอยู่ในโมดูล Monitor.C ซึ่งมีผังการทำงาน (Flow Chart) ของโหมดนี้ดังรูปที่ 3 ในภาคผนวก ก.

4. โหมดแบบบังคับเอง (Manual mode)

ในโหมดดังกล่าวจะเป็นการบังคับการเคลื่อนที่ของเครื่องจักรในแบบธรรมดา โดยอาศัยการกดปุ่ม (Keyboard) และสามารถสั่งการเคลื่อนที่ได้ทีละแกน ในโหมดนี้จะมีความสามารถต่างๆ ดังนี้

- ควบคุมการเคลื่อนที่ของแกนต่าง ๆ (Jog) โดยสามารถควบคุมมอเตอร์ได้ทีละ 1 แกน ด้วยความเร็วที่ได้กำหนดไว้ในโหมดความจำ (Memory mode)

- กลับสู่ตำแหน่งเริ่มต้น (Home) โดยจะสั่งให้มอเตอร์ทั้ง 3 แกนเคลื่อนที่กลับไปที่ตำแหน่ง Home (ตำแหน่ง (0,0,0) ของเครื่อง) ทันที

- เซ็ตศูนย์โปรแกรม (Set Zero) โดยจะทำการเซ็ตค่า ณ ตำแหน่งนั้น ๆ ให้เป็นตำแหน่งศูนย์ของโปรแกรม

- สั่งให้ Spindel ทำการหมุน (Spindel On) ด้วยความเร็วที่ได้กำหนดไว้ในโหมดความจำ (Memory mode)

- หยุดฉุกเฉิน (Emergency Stop) เป็นการหยุดการทำงานของเครื่องในกรณีฉุกเฉิน โดยได้มีการทำไว้ในลักษณะ Hardware อยู่ภายนอก โดยมีการควบคุมในแนวแกน และ Spindel

โปรแกรมสำหรับโหมดนี้จะอยู่ในโมดูล Manual.C , Monitor.C และ Home.C โดยมีผังการทำงาน (Flow Chart) ดังรูปที่ 6 ในภาคผนวก ก.

6.2.2. ส่วนที่เชื่อมต่อระหว่างคอมพิวเตอร์กับการ์ดควบคุม

โมดูล Host_io.c จะทำหน้าที่เป็นตัวควบคุม โดยจะมีชุดคำสั่งง่ายๆ สำหรับบอกการ์ดว่าจะให้ทำอะไรบ้าง เช่น กำหนดแกน, กำหนดตำแหน่ง, กำหนดความเร็ว เป็นต้น ซึ่งโปรแกรมควบคุมที่สร้างขึ้นจะต้องอ้างอิงกับคำสั่งเหล่านี้ การออกแบบโปรแกรมควบคุมก็ทำโดยการนำชุดคำสั่งต่างๆ มารวมกันให้ทำหน้าที่ได้ตามที่ต้องการ

6.2.3 ส่วนการติดต่อสื่อสารกับข้อมูลภายนอกที่อยู่ในรูปของ Text File

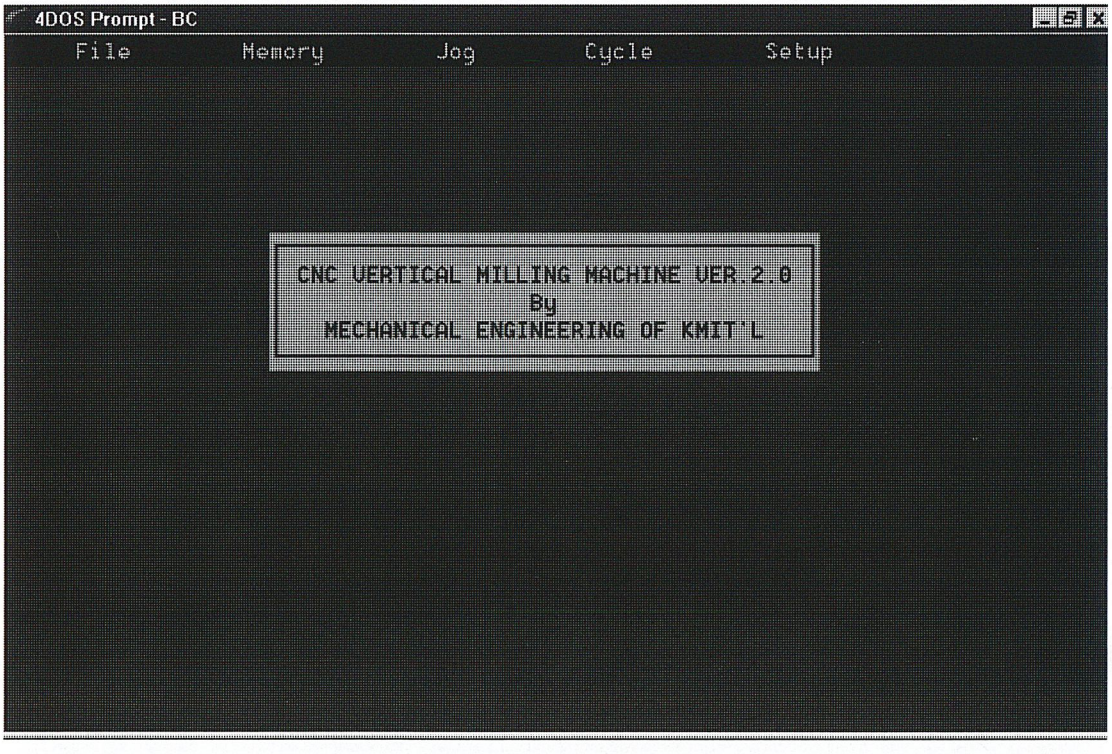
ในส่วนของการติดต่อสื่อสารกับข้อมูลภายนอกนี้จะเกิดขึ้นต่อเมื่อทำงานในโหมดอัตโนมัติเท่านั้นโมดูล Compiler.c จะทำหน้าที่อ่าน Text File ซึ่งมี Code คำสั่งในการกัดชิ้นงานเข้ามา และนำ Code เหล่านี้ไปแปลงเป็นชุดคำสั่งง่ายๆ ใน Host_io.c และ G-code.cเพื่อสั่งการ์ดควบคุมให้ทำงานตามที่ต้องการ

การทำงานของโมดูลนี้จะต้องสามารถอ่าน Code มาตรฐานเครื่อง FANUC ซึ่งเป็นเครื่องต้นแบบเข้ามาทำงานได้อย่างถูกต้องในทุกๆ Code โปรแกรมสำหรับส่วนนี้จะอยู่ในโมดูล Compiler.C โดยมีผังการทำงาน (Flow Chart) แสดงในรูปที่ 5 ในภาคผนวก ก.

6.3 วิธีการใช้โปรแกรม

ต่อไปนี้จะกล่าวถึงการใช้โปรแกรม CNC VERTICAL MILLING MACHINE VER.2.0 ซึ่งเป็นโปรแกรมที่ใช้ในการควบคุมเครื่องกัดแนวดิ่งซีเอ็นซี ดังมีขั้นตอนดังต่อไปนี้

1. นำแผ่นโปรแกรมที่แนบมากับรายงานเล่มนี้ใส่ใน Drive A หรือ B
2. พิมพ์ชื่อโปรแกรม CNC.EXE แล้วกด Enter
3. เมื่อเครื่องโหลดโปรแกรม CNC ขึ้นแล้ว จะมีหน้าต่างแสดงชื่อโปรแกรมปรากฏที่กลางจอภาพ ดังรูปที่ 6.2
4. กด Enter เพื่อเข้าสู่ Menu
5. เลือก Menu หลัก โดยใช้คีย์ลูกศร เลื่อนแถบสว่างไปยัง Menu หลักที่ต้องการ



รูปที่ 6.2

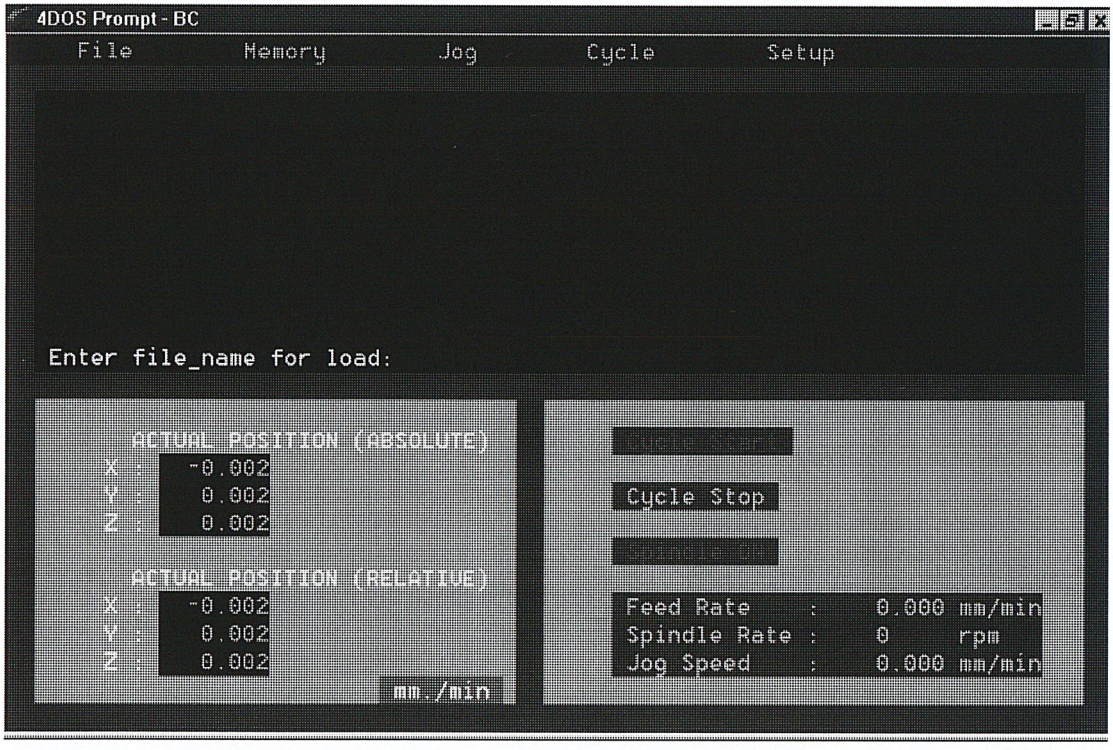
จะเห็นได้ว่าใน Menu หลัก จะประกอบด้วย 5 ส่วน คือ File , Memoy , Jog , Cycle และ Set up โดยแต่ละ Menu หลัก ประกอบด้วย Menu ย่อยๆ ซึ่งมีวิธีการใช้งานดังนี้

6.3.1 File

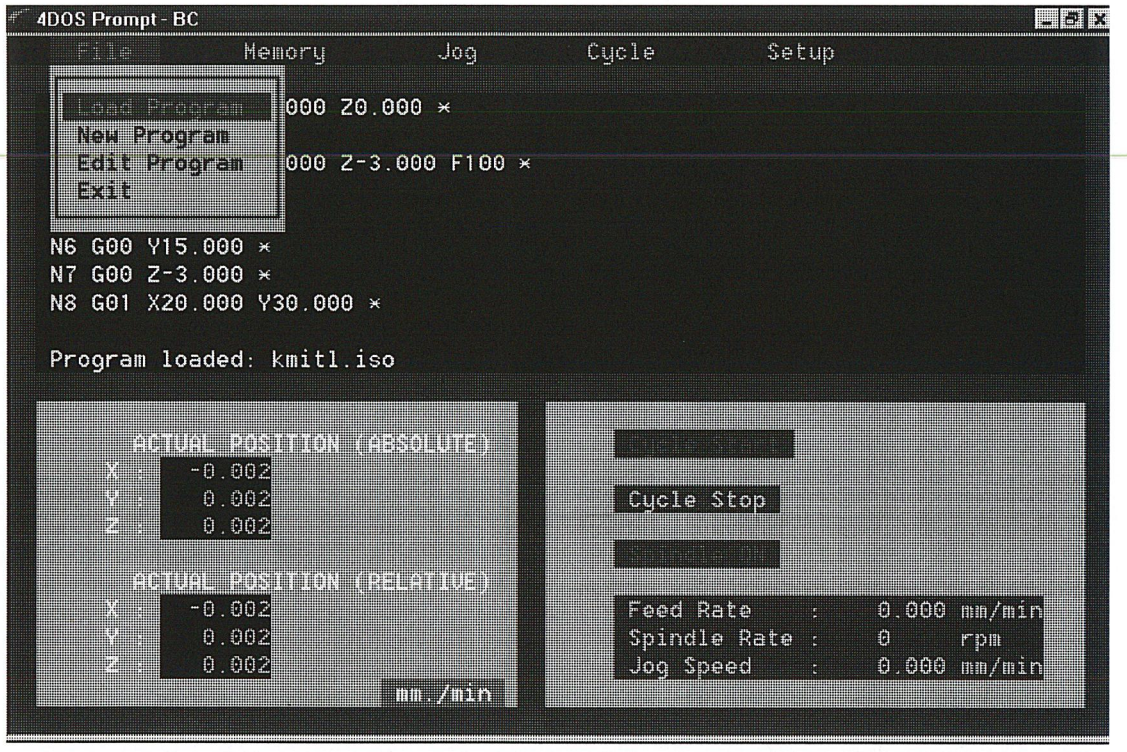
ประกอบด้วย Menu ย่อย 4 ส่วน คือ

1. Load Program

เลื่อนแถบสว่างไปยัง Load Program แล้วกด Enter จะปรากฏข้อความให้ใส่ชื่อโปรแกรมที่ต้องการโหลด ดังรูปที่ 6.3 ซึ่งต้องเป็นโปรแกรมที่อยู่ในรูปเท็กซ์ไฟล์เท่านั้น และต้องมีชื่ออยู่ในไดเรกเตอรีเดียวกันกับโปรแกรม CNC.EXE ด้วย แล้วกด Enter จอภาพจะแสดงชื่อโปรแกรมที่โหลดขึ้นมา พร้อมด้วยรายละเอียดของข้อมูลในโปรแกรมให้เห็นดังรูปที่ 6.4



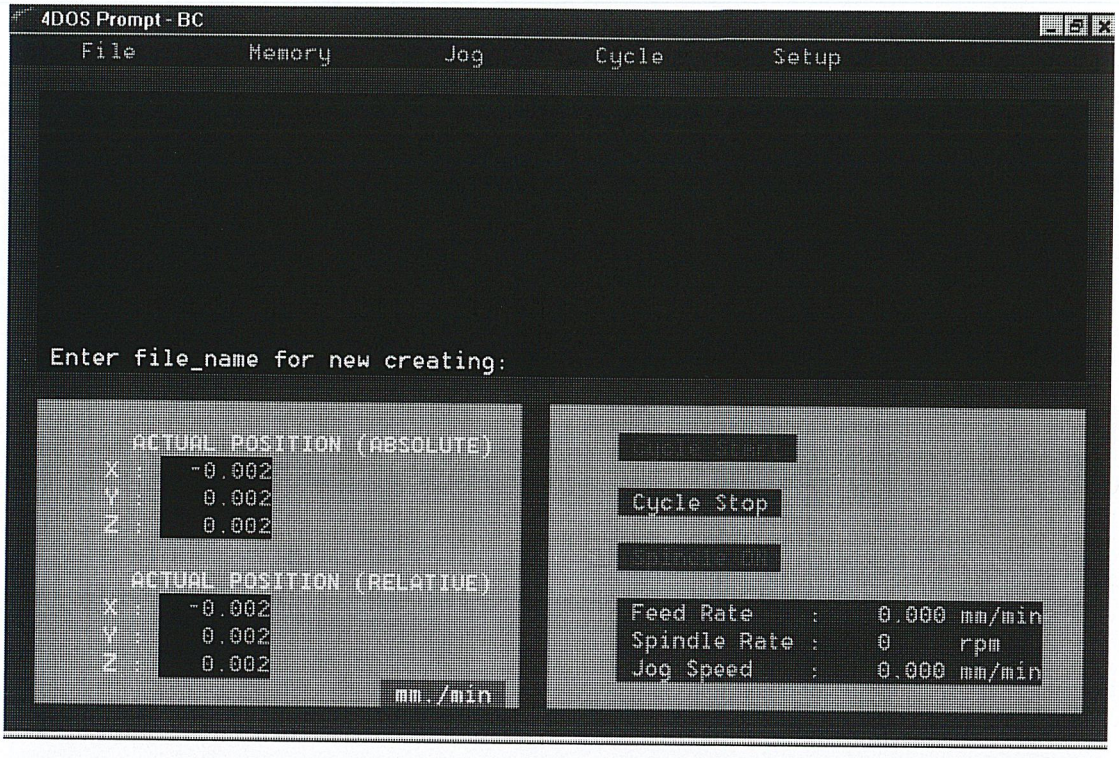
รูปที่ 6.3



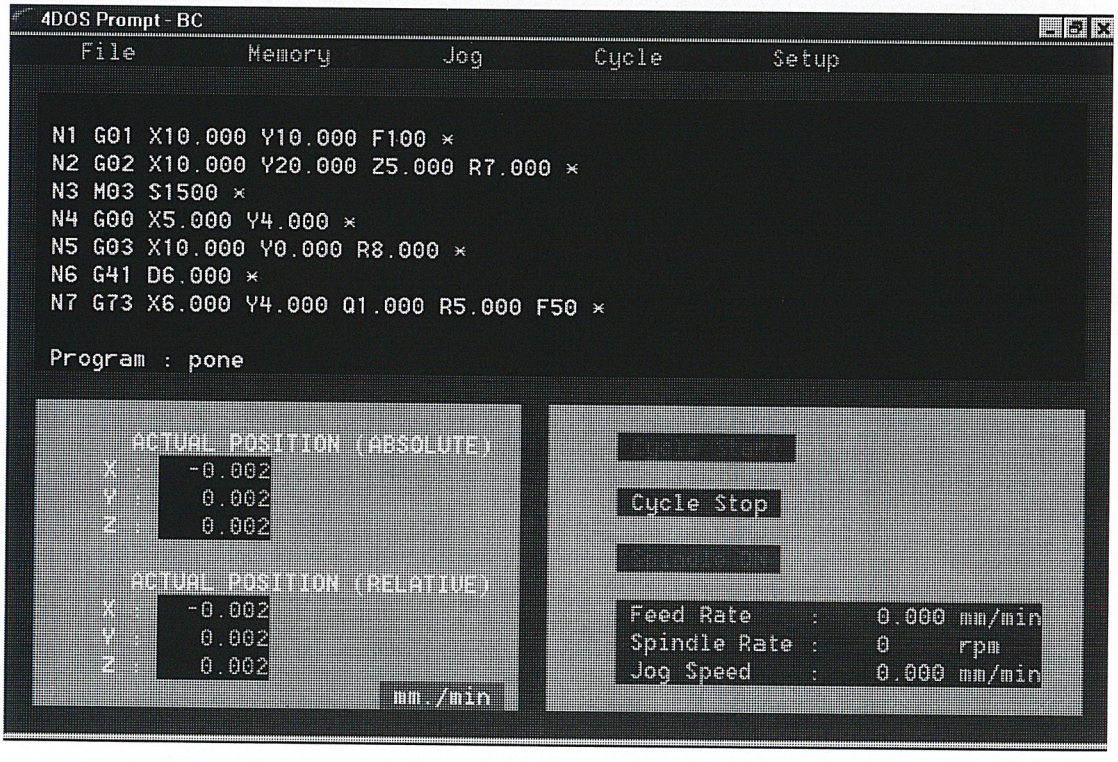
รูปที่ 6.4

2. New program

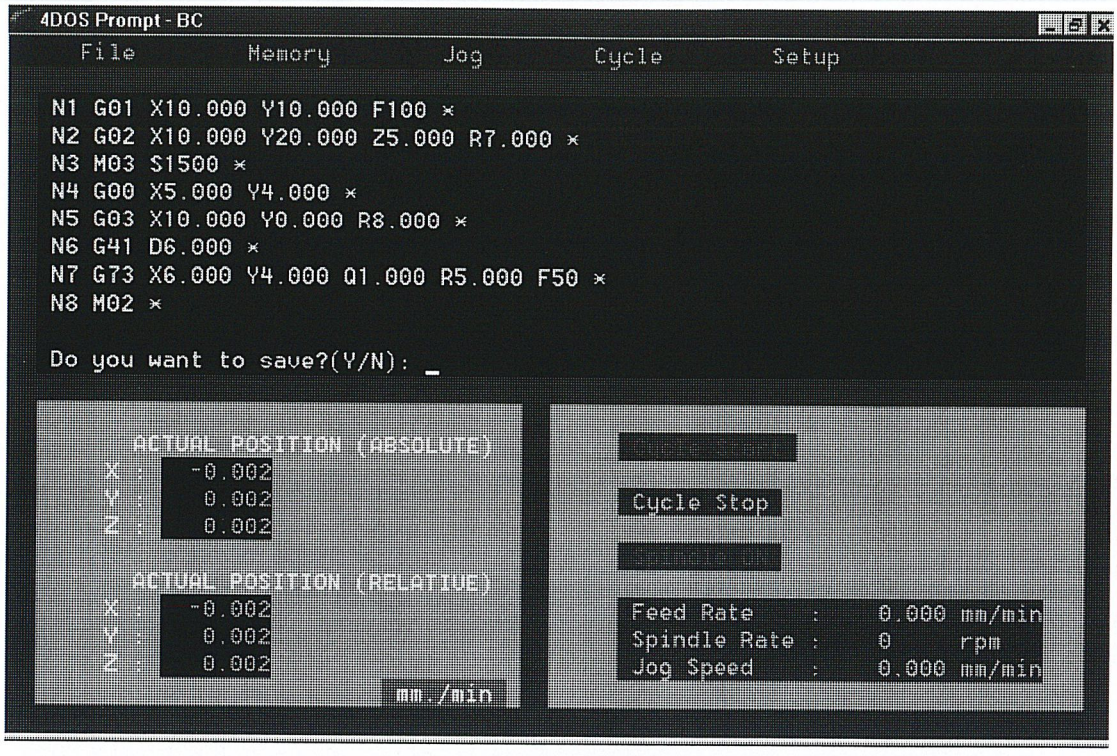
เลื่อนแถบสว่างไปยัง New program แล้วกด Enter จะปรากฏข้อความให้ใส่ชื่อโปรแกรมที่ต้องการจะเขียนใหม่ ดังรูปที่ 6.5 โดยชื่อโปรแกรมที่จะใส่ต้องไม่ซ้ำกับชื่อโปรแกรมเดิมที่เคยมีอยู่ มิฉะนั้น โปรแกรมจะไม่สามารถเขียนโปรแกรมใหม่ได้ จากนั้นกด Enter หน้าจอจะเข้าสู่ Edit mode โดยการเขียนโปรแกรมแต่ละบรรทัด จะมีการขึ้นเลขบรรทัดให้โดยอัตโนมัติ ส่วน G-code และ M-code ที่จะเขียน จะมีรูปแบบของคำสั่งแต่ละตัวให้ เพื่อป้องกันการเขียนคำสั่งผิดรูปแบบมาตรฐาน การใส่ค่าตัวเลขจะต้องมีการกด Enter ตามทุกครั้ง ถ้าไม่ใส่ให้กด Enter ผ่าน เมื่อจบบรรทัดจะมีเครื่องหมาย * ปิดท้าย ดังรูปที่ 6.6 หลังจากนั้นถ้าต้องการยืนยันว่าบรรทัดที่เพิ่งเขียนมาถูกต้องให้กด Enter ถ้าต้องการแก้ไขใหม่ให้กด Del จากนี้จะสามารถใช้คีย์ลูกศรเลื่อนดูโปรแกรมในลักษณะขึ้นลงได้ หากต้องการแก้ไขบรรทัดใด หรือแทรกบรรทัดใดก็สามารถทำได้ด้วยการกด Del และ Insert ตามลำดับ และดำเนินวิธีการเขียนคำสั่งดังที่ได้กล่าวมาแล้วข้างต้นหากกด Enter ในขณะที่อยู่ในช่วงการเลื่อนโปรแกรมขึ้นลง ไม่ว่าตัวชี้จะอยู่ที่บรรทัดใดก็ตาม จะมีการขึ้นบรรทัดใหม่เพื่อการเขียนคำสั่งทันที การออกจาก Edit mode สามารถทำได้ด้วยการกด Escหรือเขียนบรรทัดสุดท้ายด้วยคำสั่ง M02 แล้วจะปรากฏข้อความในการ Save ข้อมูล ดังรูปที่ 6.7 ซึ่งเราจะทำการ Save หรือไม่ก็ได้ โดยถ้าเรา Save ข้อมูลจะถูกเก็บลงในโปรแกรม แต่ถ้าไม่ Save ข้อมูลจะไม่ถูกเก็บลงไป และโปรแกรมก็จะออกจาก Edit mode มายังหน้าจอปกติเลย


4DOS Prompt - BC
File
Memory
Jog
Cycle
Setup
Enter file_name for new creating:
ACTUAL POSITION (ABSOLUTE)
X : -0.002
Y : 0.002
Z : 0.002
ACTUAL POSITION (RELATIVE)
X : -0.002
Y : 0.002
Z : 0.002
mm./min
Cycle Stop
Feed Rate : 0.000 mm/min
Spindle Rate : 0 rpm
Jog Speed : 0.000 mm/min

รูปที่ 6.5


4DOS Prompt - BC
File
Memory
Jog
Cycle
Setup
N1 G01 X10.000 Y10.000 F100 *
N2 G02 X10.000 Y20.000 Z5.000 R7.000 *
N3 M03 S1500 *
N4 G00 X5.000 Y4.000 *
N5 G03 X10.000 Y0.000 R8.000 *
N6 G41 D6.000 *
N7 G73 X6.000 Y4.000 Q1.000 R5.000 F50 *
Program : pone
ACTUAL POSITION (ABSOLUTE)
X : -0.002
Y : 0.002
Z : 0.002
ACTUAL POSITION (RELATIVE)
X : -0.002
Y : 0.002
Z : 0.002
mm./min
Cycle Stop
Feed Rate : 0.000 mm/min
Spindle Rate : 0 rpm
Jog Speed : 0.000 mm/min

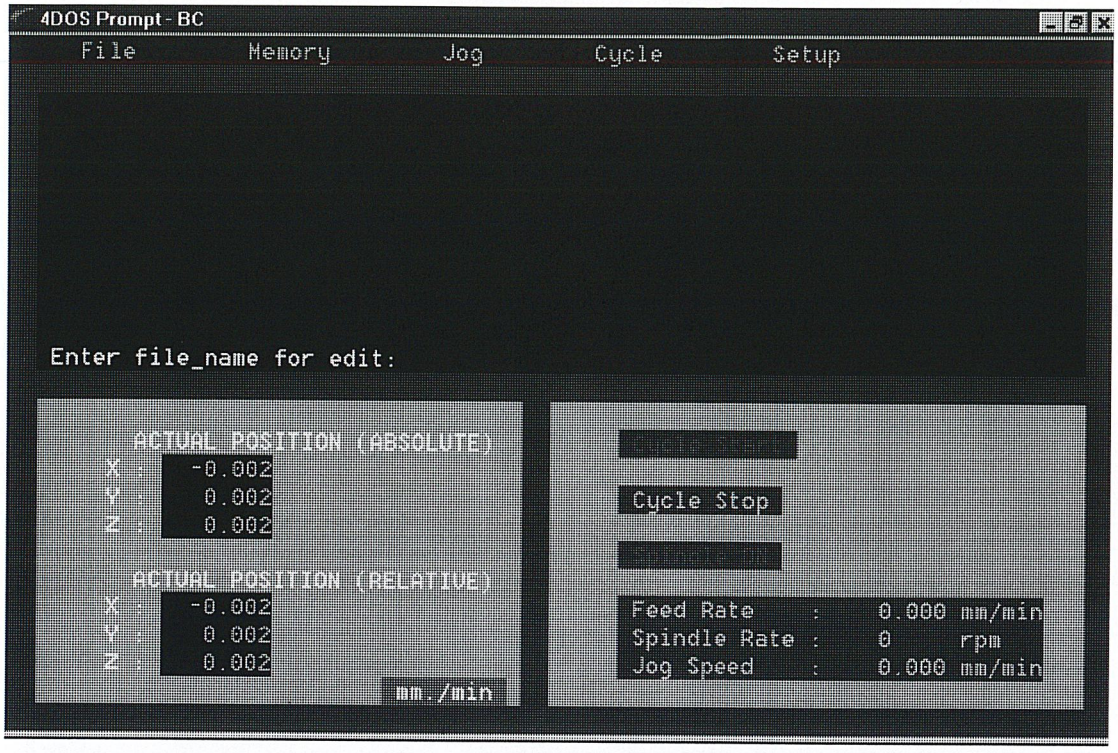
รูปที่ 6.6



รูปที่ 6.7

3. Edit program

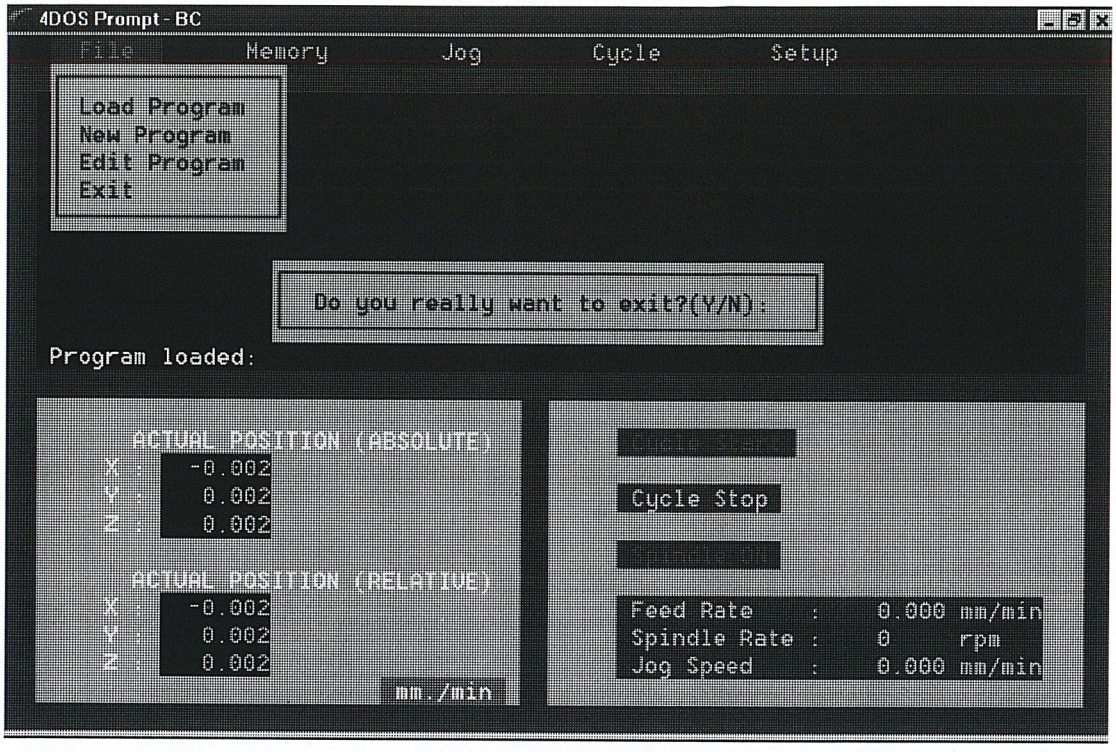
เลื่อนแถบสว่างไปยัง Edit program แล้วกด Enter จะปรากฏข้อความให้ใส่ชื่อโปรแกรมที่ต้องการจะทำการ Edit ดังรูปที่6.8โดยชื่อโปรแกรมที่จะใส่ต้องเป็นชื่อโปรแกรมที่มีอยู่แล้ว มิฉะนั้นโปรแกรมจะไม่สามารถทำการ Edit ได้ จากนั้นกด Enter หน้าจอจะเข้าสู่ Edit mode โดยจะทำการ Load โปรแกรมนั้นขึ้นมาบนหน้าจอเพื่อทำการแก้ไข โดยวิธีการแก้ไขข้อมูลของโปรแกรมนั้นมีวิธีเหมือนกับในโหมด New program การออกจาก Edit mode ในส่วนนี้ ทำได้เช่นเดียวกับในโหมด New Program โดยอาจจะทำการกด Esc หรือเลื่อนตัวชี้มายังบรรทัดสุดท้าย ซึ่งเป็นคำสั่ง M02 แล้วกด Enter จะปรากฏข้อความในการทำการ Save ข้อมูล ซึ่งเราจะ Save การเปลี่ยนแปลงของโปรแกรมหรือไม่ก็ได้ โดยถ้าเราทำการ Save ข้อมูลจะถูกเก็บลงในโปรแกรม แต่ถ้าไม่ Save ก็จะเป็นการออกจาก Edit mode และกลับสู่หน้าจอปกติเลย โดยข้อมูลจะไม่ถูกเก็บลงในโปรแกรม



รูปที่ 6.8

4. Exit

เลื่อนแถบสว่างไปยัง Exit แล้วกด Enter จะปรากฏข้อความดังรูปที่ 6.9 ถ้ากด Y ก็จะออกจากโปรแกรม CNC และ กลับไปบนระบบปฏิบัติการอีกครั้ง ถ้ากด N ก็จะกลับเข้าสู่การใช้งานโปรแกรม CNC ตามเดิม



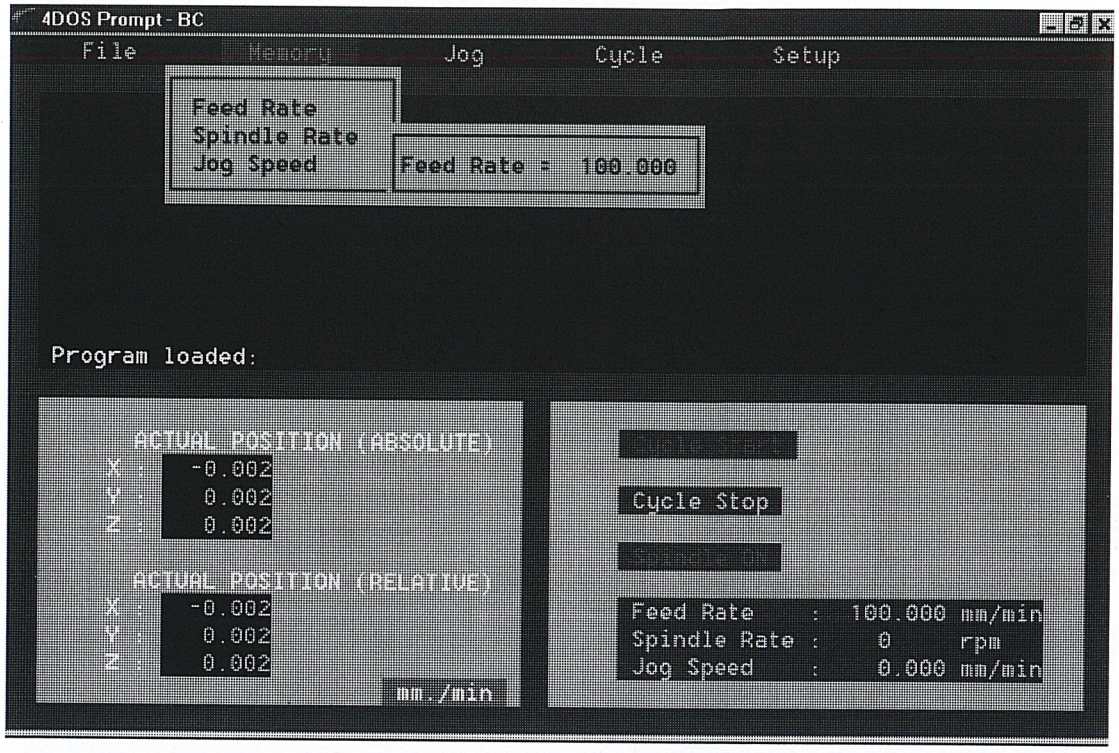
รูปที่ 6.9

6.3.2 Memory

ประกอบด้วย Menu ย่อยอีก 3 ส่วนคือ

1. Feed rate

เมื่อมีการเลือก Menu นี้จะปรากฏข้อความให้ใส่ค่าของ Feed rate ใหม่ ซึ่งจะมีค่า Feedrate เดิมแสดงให้ห็นด้วย ถ้าไม่ต้องการเปลี่ยนให้กด Enter 2 ครั้ง ถ้าต้องการเปลี่ยนให้กด Enter แล้วพิมพ์ค่าที่ต้องการ ตามด้วยการกด Enter อีก 1 ครั้ง ดังรูปที่ 6.10 โดยค่าของ Feed rate ที่เปลี่ยนใหม่นี้จะมีผลต่อทุกโหมดในโปรแกรม CNC



รูปที่ 6.10

2. Spindle speed

การใช้งานและหน้าที่จะเหมือนกับการใช้งานของ Feed rate ซึ่งค่า Spindle speed นี้คือค่าอัตราเร็วในการหมุนของหัว Spindle โดยค่าของ Spindle speed ที่เปลี่ยนใหม่นี้จะมีผลต่อทุกโหมดในโปรแกรม CNC

3. Jog speed

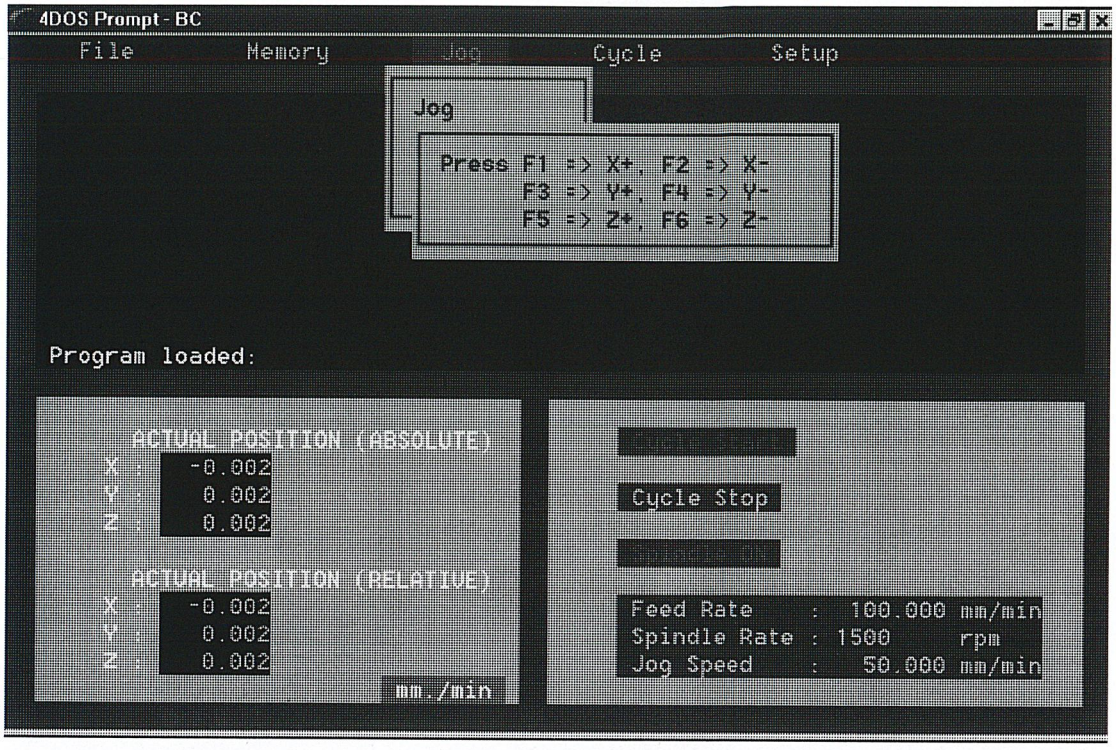
การใช้งานและหน้าที่จะเหมือนกับการใช้งานของ Feed rate และ Spindel speed ซึ่งค่า Jog speed นี้คือค่าอัตราเร็วในการเดินของแกนต่าง ๆ ในโหมด Manual เพราะฉะนั้น ค่าที่เปลี่ยนใหม่นี้จะมีผลต่อการบังคับแบบ Manual mode เท่านั้น

6.3.3 Jog

ประกอบด้วย Menu ย่อย 4 ส่วนคือ

1. Jog

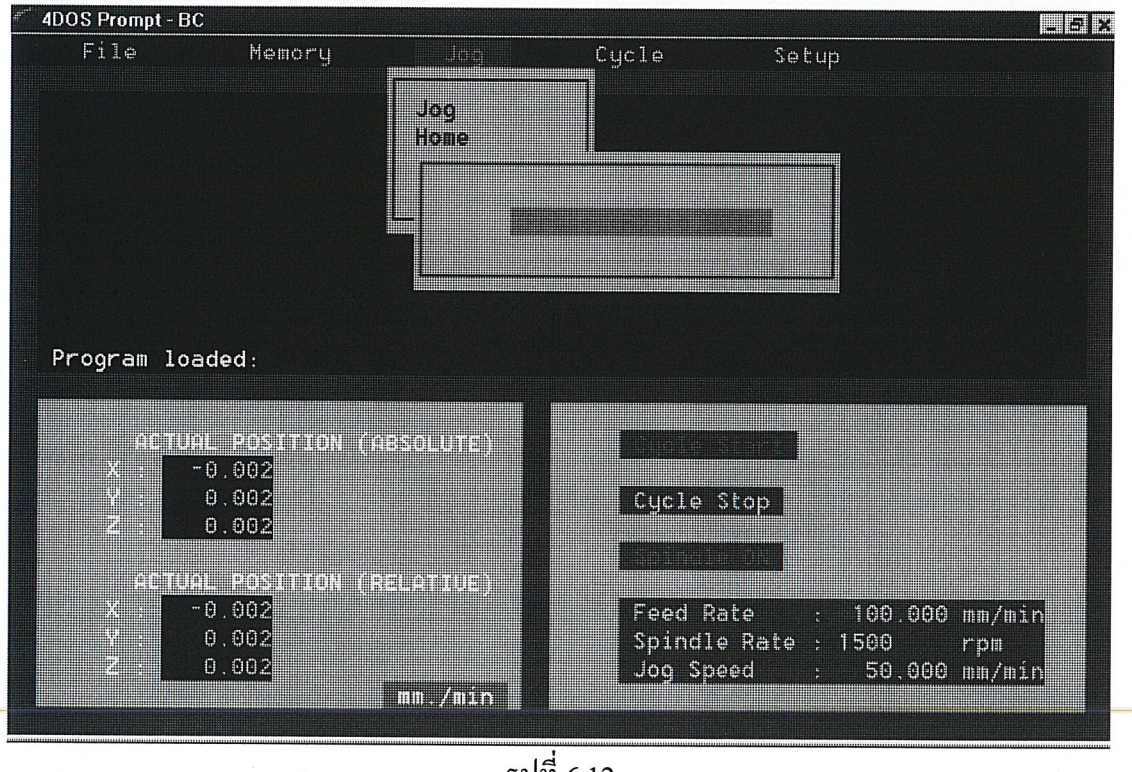
เมื่อมีการเลือก Menu นี้ จะปรากฏข้อความถึงปุ่ม (Keyboard) ที่จะใช้ควบคุม การเคลื่อนที่ของ มอเตอร์ให้ไปในทิศทางที่เราต้องการ ดังรูปที่ 6.11 การควบคุมสามารถกระทำได้โดยการกดปุ่มโดยตรง



รูปที่ 6.11

2. Home

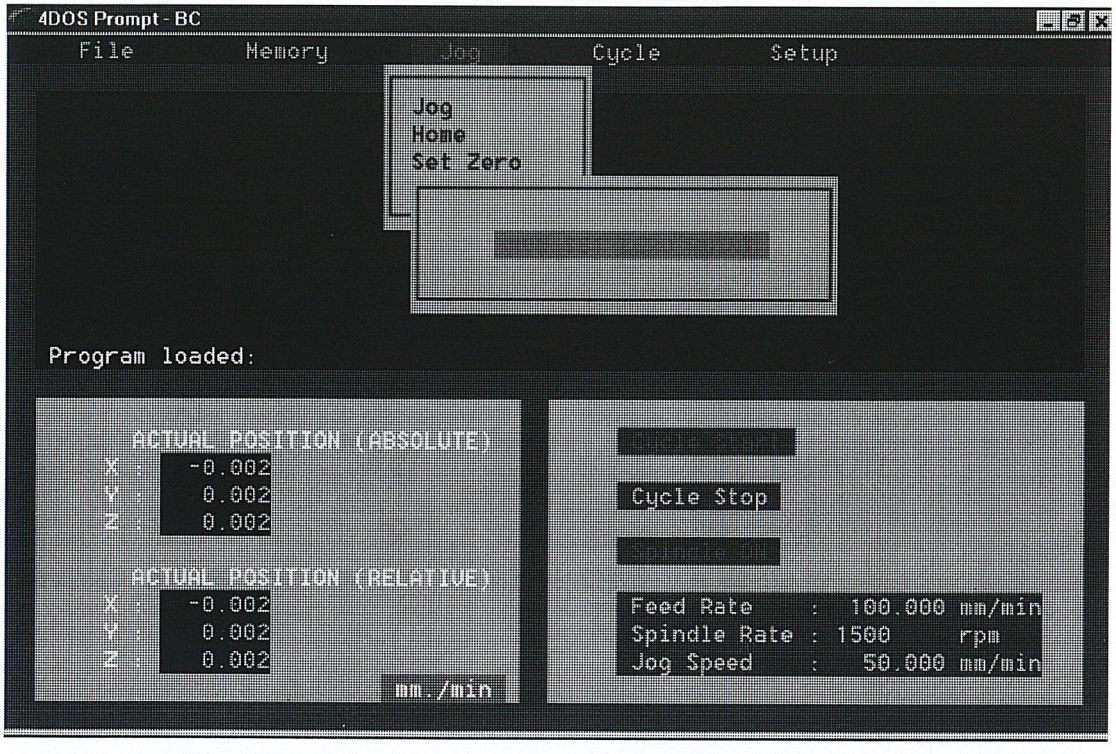
เมื่อมีการเลือก Menu นี้จะปรากฏข้อความบนจอภาพ ดังรูปที่ 6.12 และในแต่ละแกนจะมีการเคลื่อนที่กลับไปยังตำแหน่ง Home



รูปที่ 6.12

3. Set Zero

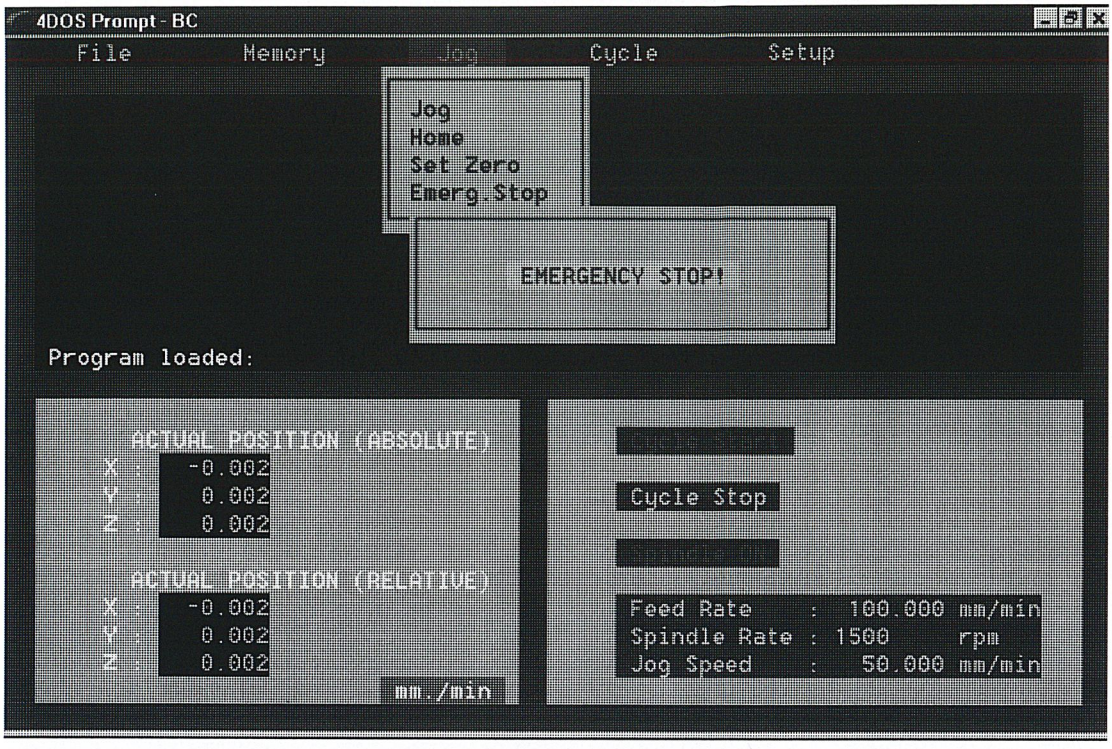
เมื่อมีการเลือก Menu นี้จะปรากฏข้อความบนจอภาพ และค่าตำแหน่งที่แสดงบนจอภาพ (Actual Position <Relative>) จะถูกรีเซ็ตให้เป็นศูนย์ทั้งหมด ดังรูปที่ 6.13



รูปที่ 6.13

4. Emergency Stop

เมื่อมีการเลือก Menu นี้ จะปรากฏข้อความบนจอภาพ ดังรูปที่ 6.14 และเครื่องจักรจะหยุดการทำงานทันที โดยในโหมดนี้ยังไม่ได้ทำการเขียนขึ้นมา เนื่องจากต้องทำให้โปรแกรมสามารถทำงานในลักษณะ Muti-tasking ให้ได้เสียก่อน แต่เราได้แก้ไขด้วยการทำ Emergency Stop นี้ให้เป็นลักษณะของ Switch ปิด-เปิด อยู่ภายนอกในลักษณะของ Hardware ไปก่อน



รูปที่ 6.14

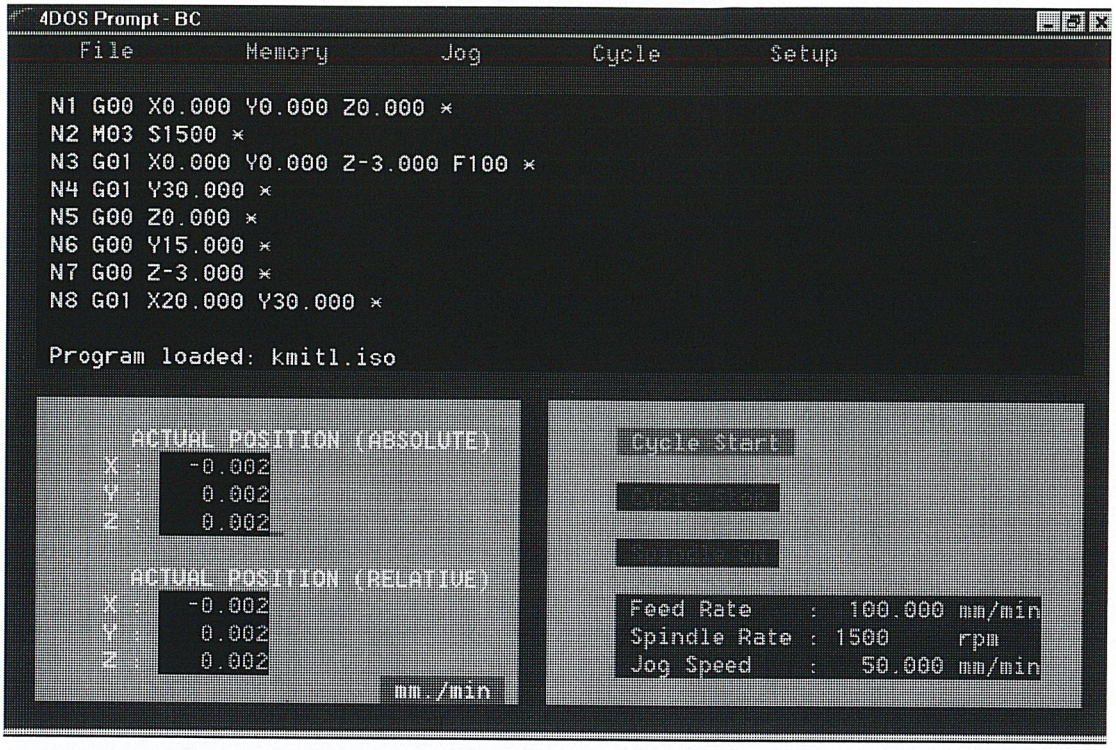
สำหรับทุกๆ Menu ย่อย ดังที่ได้กล่าวมาแล้วนี้ การที่จะยกเลิกการทำงานหรือออกจาก Menu ย่อยนั้นๆ สามารถทำได้โดยการกดปุ่ม ESC

6.3.4 Cycle

ประกอบด้วย Menu ย่อย 4 ส่วน คือ

1. Cycle Start

เลื่อนแถบสว่างไปยัง Cycle Start แล้วกด Enter จะปรากฏข้อความกระพริบ “Cycle Start” ขึ้นบนหน้าจอ แล้วเครื่องจักรจะทำตามโปรแกรมที่ได้โหลดขึ้นมาโดยคำสั่ง Load Program ก่อนหน้านี้ โดยจะมีการแสดงข้อมูลของ Program ที่กำลังทำการ Machine ขึ้นแสดงบนหน้าจอด้วยดังรูปที่ 6.15



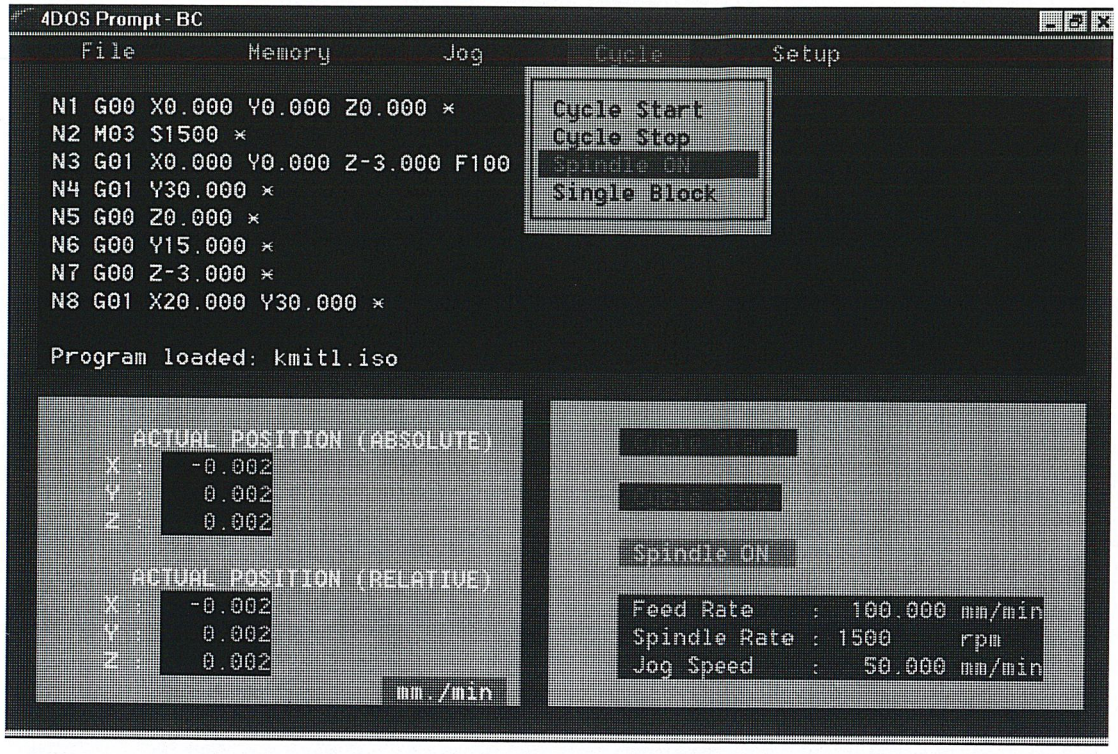
รูปที่ 6.15

2. Cycle Stop

เลื่อนแถบสว่างไปยัง Cycle Stop แล้วกด Enter จะปรากฏแถบสว่างบนข้อความ “Cycle Stop” แล้วเครื่องจักรจะหยุดทำงาน ถ้าต้องการเริ่มการทำงานใหม่ให้เลือก Menu Cycle Start

3. Spindel On

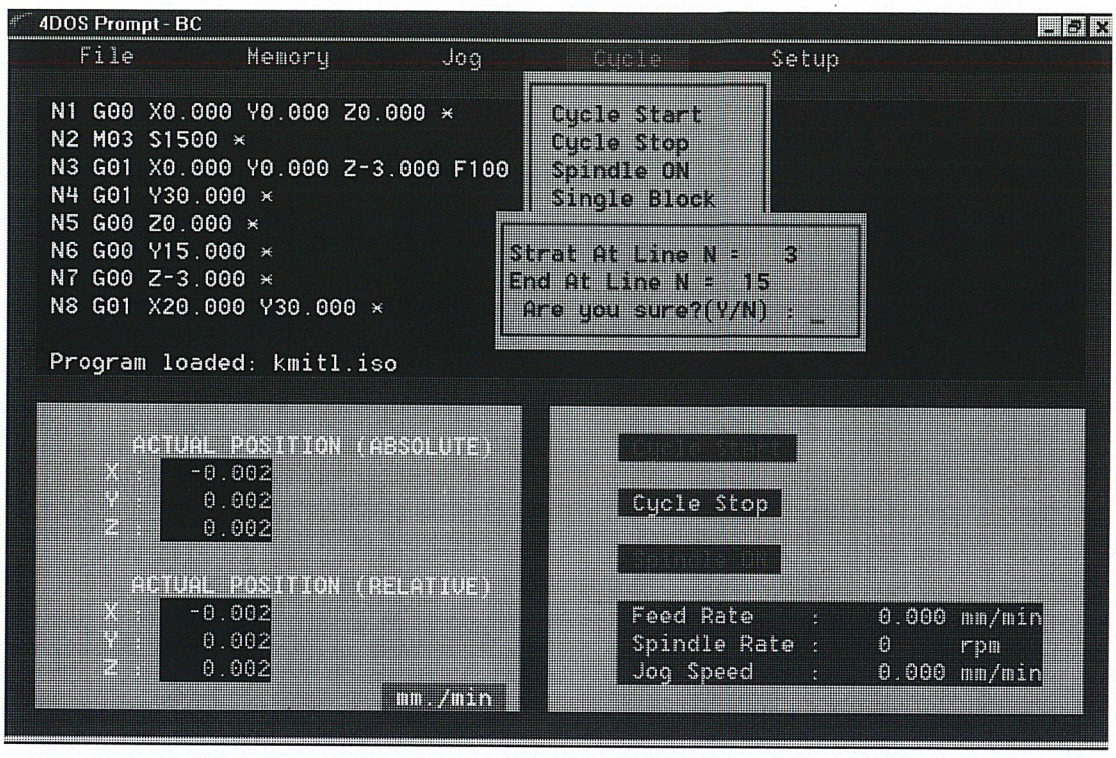
เลื่อนแถบสว่างไปยัง Spindel On แล้วกด Enter จะปรากฏข้อความกระพริบ “Spindel On” บนหน้าจอ ใช้ในการควบคุมการหมุนของ Spindel ในการควบคุมแบบบังคับเองดังรูปที่ 6.16



รูปที่ 6.16

4.Single Block

เลื่อนแถบสว่างไปยัง Single Block แล้วกด Enter จะปรากฏข้อความให้ใส่ค่าบรรทัด N ซึ่งใช้เป็นบรรทัดเริ่มต้นในการ Run แบบ Single Block และค่า N ซึ่งเป็นบรรทัดจบ Single Block โดยถ้าไม่ต้องการเปลี่ยนแปลงค่าให้กด Enter 2 ครั้ง หลังจากใส่ค่า N แล้วจะปรากฏข้อความยืนยันการเปลี่ยนแปลง โดยถ้าถูกต้องก็กด Y ถ้าไม่ถูกต้องก็กด N ดังรูปที่ 6.17



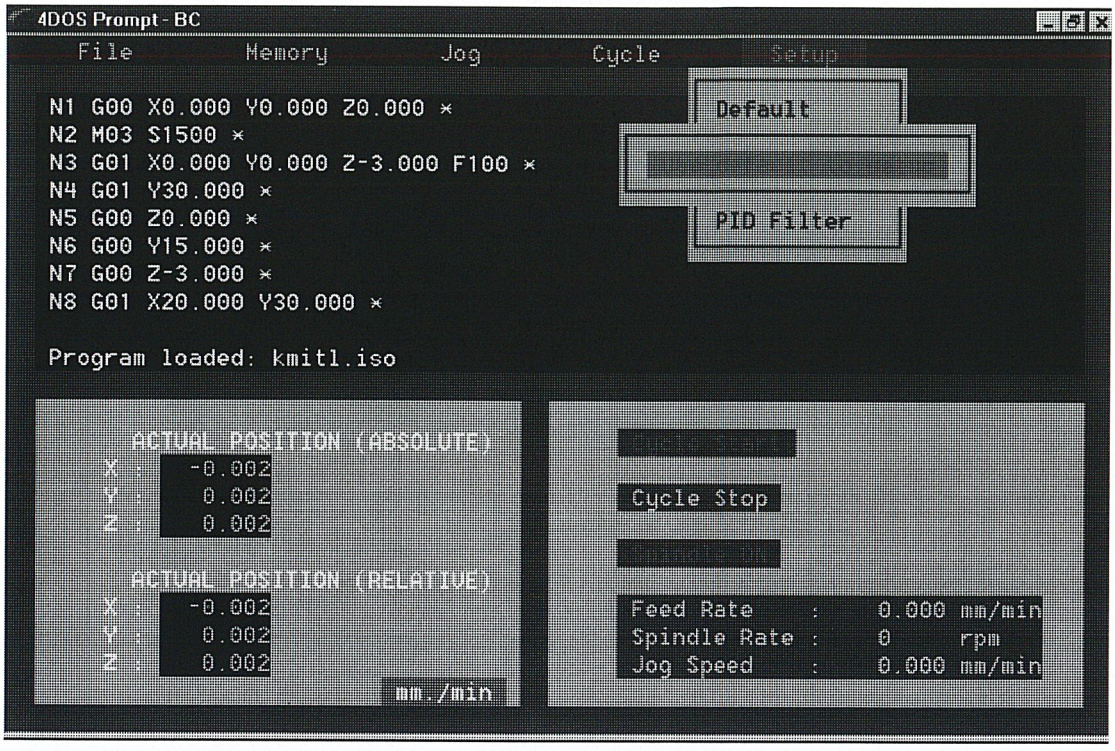
รูปที่ 6.17

6.3.5 Setup

ประกอบด้วย Menu ย่อย 5 ส่วน คือ

1. Default

เมื่อ Menu นี้ถูกเลือก จะปรากฏข้อความดังรูปที่ 6.18 และจะมีการเซ็ตค่าพารามิเตอร์ต่างๆ ดังต่อไปนี้



รูปที่ 6.18

1.1 ค่า PID Digital Filter

1.2 ค่า pulse / rev. ของ Encoder ที่ใช้

1.3 ค่า Pitch ของ Ball Screw ที่ใช้ให้เป็นค่าดั้งเดิม ซึ่งเป็นค่าเฉพาะของเครื่องต้นแบบนี้เท่านั้น

2. In. / mm.

เมื่อ Menu นี้ถูกเลือก ค่าของตำแหน่ง (ทั้ง Abs. และ Rel.) , Feed Rate และ Jog Speed จะถูกเปลี่ยนให้อยู่ในหน่วยของ มิลลิเมตร หรือ นิ้ว ได้ตามต้องการ และจะมีผลต่อหน่วยของข้อมูลที่อยู่ในโปรแกรมที่ได้ทำการโหลดไว้ก่อนหน้านี้ด้วย

3. Ball Screw

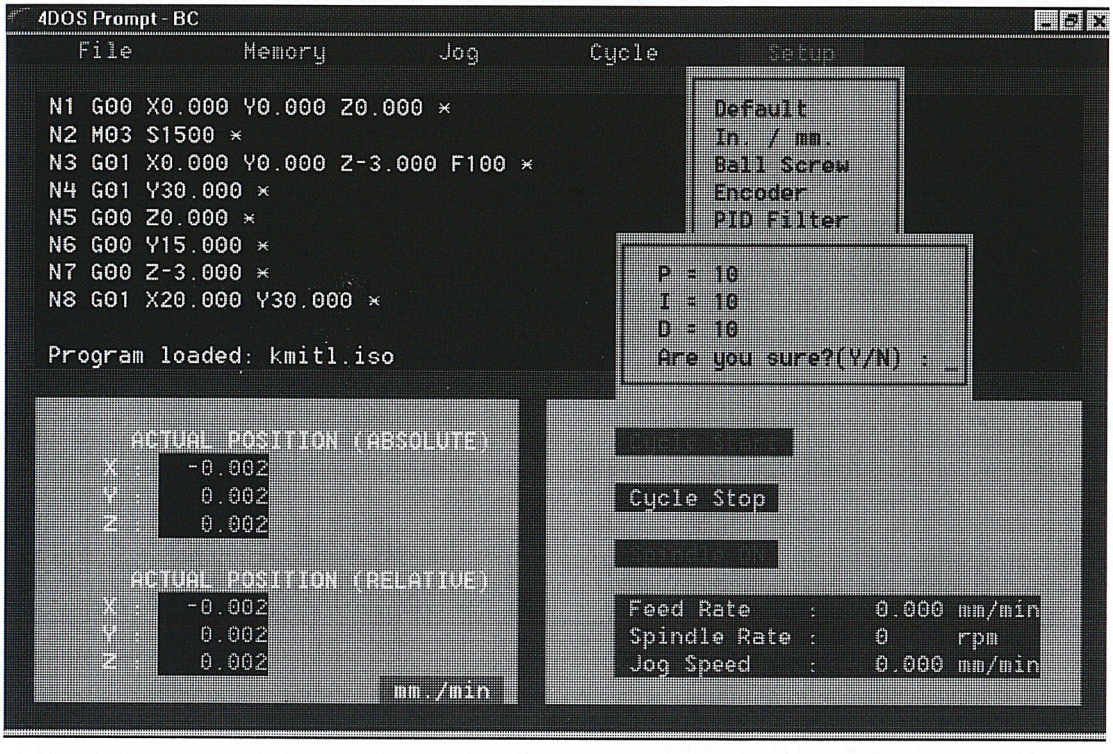
เมื่อ Menu นี้ถูกเลือก จะปรากฏข้อความให้ใส่ค่า Pitch ในหน่วย มิลลิเมตร ของ Ball Screw ลงไปโดยวิธีการเปลี่ยนค่าจะคล้ายกับ FeedRate

4. Encoder

เมื่อ Menu นี้ถูกเลือก จะปรากฏข้อความให้ใส่ค่า pulse / rev. ของ Encoder ลงไป โดยวิธีการเปลี่ยนค่าจะคล้ายกับ Feed Rate

5. PID

เมื่อ Menu นี้ถูกเลือก จะปรากฏข้อความให้ใส่ค่าพารามิเตอร์ P , I และ D โดยจะแสดงค่าเดิมที่ใช้อยู่ด้วย หากไม่ต้องการเปลี่ยนค่าให้กด Enter 2 ครั้ง หากต้องการเปลี่ยน ให้กด Enter ตามด้วยค่าที่ต้องการเปลี่ยนแล้วกด Enter ทำเช่นนี้ไปเรื่อยๆจนครบทั้ง 3 ค่าก็จะปรากฏข้อความยืนยันการเปลี่ยนแปลง ถ้ากด Y หมายถึง ถูกต้อง ถ้ากด N หมายถึงไม่ถูกต้อง ดังรูปที่ 6.19



รูปที่ 6.19

บทที่ 7

ผลการทดลอง

การทดลองได้จัดทำขึ้นโดยนำ Gcode ที่เป็น Code ที่ใช้ในการควบคุมการกัดชิ้นงาน อันได้แก่

G00 เคลื่อนที่ไปยังตำแหน่งที่กำหนดเป็นเส้นตรงด้วย Feed rate สูงสุด

G01 เคลื่อนที่ไปยังตำแหน่งที่กำหนดเป็นเส้นตรงด้วย Feed rate ที่กำหนด

G02 เคลื่อนที่ไปยังตำแหน่งที่กำหนดเป็นเส้นโค้ง ตามเข็มนาฬิกา ด้วย รัศมี และ Feed rate ที่กำหนด

G03 เคลื่อนที่ไปยังตำแหน่งที่กำหนดเป็นเส้นโค้ง ทวนเข็มนาฬิกา ด้วย รัศมี และ Feed rate ที่กำหนด

G40 ยกเลิกการชดเชยรัศมีดอกกัด

G41 ชดเชยรัศมีดอกกัดด้านซ้าย

G42 ชดเชยรัศมีดอกกัดด้านขวา

G73 การขุดเจาะกระแทกเป็นจังหวะ

G81 การขุดเจาะเป็นจังหวะ คว้านรูเป็นจุดๆ

G82 การขุดเจาะเป็นจังหวะ คว้านผายปากรู

G83 การเจาะกดเป็นจังหวะ

G85 การคว้านรูเป็นจังหวะ ถอยกลับด้วยความเร็วที่กำหนด

G86 การคว้านรูเป็นจังหวะ ถอยกลับด้วยความเร็วสูงสุด

G89 การคว้านรูเป็นจังหวะ เจาะคายเศษ

โดยทำการกัดชิ้นงานจริง ใช้ค่าพารามิเตอร์ต่างๆดังนี้

- ดอกกัดแบบ Ball Nose เส้นผ่านศูนย์กลาง 3 mm
- Spindle Rate 1500 rpm
- ค่า PID Filter P=10, I=10, D=10
- Chord 0.1 mm

ผลการทดลองที่ได้มีดังนี้

1. ทดลองกัด Path เส้นตรง ตาม Code ที่กำหนด

N1 G00 Z10.000 *

N2 G00 X-40.000 Y-40.000 *

N3 G01 Z-3.000 F100 *

N4 G01 Y40.000 *

N5 G01 X40.000 *

N6 G01 Y-40.000 *

N7 G01 X-40.000 *

N8 G00 Z10.000 *

N9 G00 X0.000 Y0.000 *

N10 G73 Z-5.000 Q1.000 R0.000 F100 *

N11 M02 *

ได้ผลดังนี้

2. ทดลองกัด Path วงกลมตาม Code ที่กำหนด

N1 G00 X0.000 Y0.000 Z10.000 *

N2 M03 S1500 *

N3 G00 X-40.000 *

N4 G01 Z-3.000 F100 *

N5 G02 X40.000 Y0.000 R40.000 F100 *

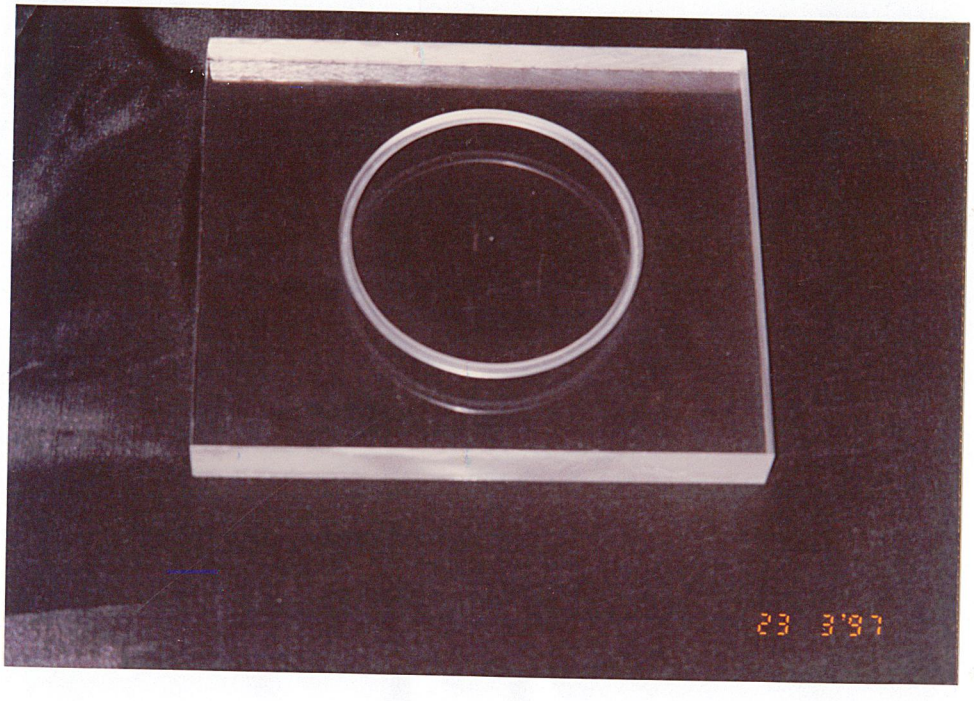
N6 G02 X-40.000 Y0.000 R40.000 F100 *

N7 G00 Z50.000 *

N8 G00 X0.000 Y0.000 *

N9 M02 *

ได้ผลดังนี้



3. ทดลองกัด Path รวมตาม Code ที่กำหนด

N1 G00 Z10.000 *

N2 M03 S1500 *

N3 G01 X0.000 Y0.000 Z-3.000 F100 *

N4 G01 X0.000 Y25.000 *

N5 G01 X10.000 Y40.000 *

N6 G01 X0.000 Y55.000 *

N7 G01 X0.000 Y80.000 *

N8 G01 X20.000 *

N9 G01 Y50.000 *

N10 G03 X40.000 Y50.000 R12.000 *

N11 G02 X60.000 Y50.000 R15.000 *

N12 G01 Y40.000 *

N13 G01 X80.000 Y80.000 *

N14 G01 Y0.000 *

N15 G01 X60.000 *

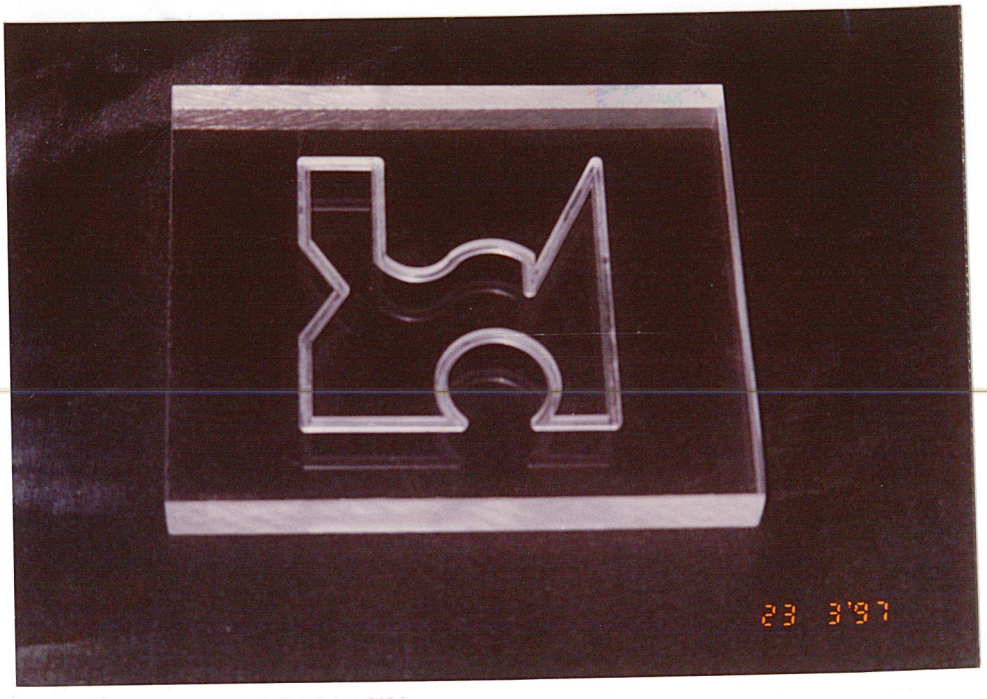
N16 G03 X40.000 Y0.000 R-15.000 F100 *

N17 G01 X0.000 Y0.000 *

N18 G00 Z20.000 *

N19 M02 *

ได้ผลดังนี้



4. ทดลองกัด Pocket ตาม Code ที่กำหนด

N1 G00 Z10.000 *
N2 M03 S1500 *
N3 G00 X-20.000 Y-20.000 *
N4 G01 Z-2.000 F100 *
N5 G01 Y20.000 *
N6 G01 X-18.500 *
N7 G01 Y-20.000 *
N8 G01 X-17.000 *
N9 G01 Y20.000 *
N10 G01 X-15.500 *
N11 G01 Y-20.00 *
N12 G01 X-14.000 *
N13 G01 Y20.000 *
N14 G01 X-12.500 *
N15 G01 Y-20.000 *
N16 G01 X-11.000 *
N17 G01 Y20.000 *
N18 G01 X-9.500 *
N19 G01 Y-20.000 *
N20 G01 X-8.000 *
N21 G01 Y20.000 *
N22 G01 X-6.500 *
N23 G01 Y-20.000 *
N24 G01 X-5.000 *
N25 G01 Y20.000 *
N26 G01 X-3.500 *
N27 G01 Y-20.000 *
N28 G01 X-2.000 *
N29 G01 Y20.000 *
N30 G01 X-0.500 *
N31 G01 Y-20.000 *
N32 G01 X1.000 *
N33 G01 Y20.000 *
N34 G01 X2.500 *
N35 G01 Y-20.000 *
N36 G01 X4.000 *
N37 G01 Y20.000 *
N38 G01 X5.500 *
N39 G01 Y-20.000 *
N40 G01 X7.000 *
N41 G01 Y20.000 *
N42 G01 X8.500 *
N43 G01 Y-20.000 *
N44 G01 X10.000 *
N45 G01 Y20.000 *
N46 G01 X11.500 *
N47 G01 Y-20.000 *
N48 G01 X13.000 *
N49 G01 Y20.000 *
N50 G01 X14.500 *
N51 G01 Y-20.000 *
N52 G01 X16.000 *
N53 G01 Y20.000 *
N54 G01 X17.500 *
N55 G01 Y-20.000 *
N56 G01 X19.000 *
N57 G01 Y20.000 *
N58 G01 X20.000 *
N59 G01 Y-20.000 *
N60 G00 Z10.000 *

N61 G00 X-20.0 Y-20.0 *

N62 G01 Z-4.0 F100 *

N63 G01 Y20.000 *

N64 G01 X-18.500 *

N65 G01 Y-20.000 *

N66 G01 X-17.000 *

N67 G01 Y20.000 *

N68 G01 X-15.500 *

N69 G01 Y-20.000 *

N70 G01 X-14.000 *

N71 G01 Y20.000 *

N72 G01 X-12.500 *

N73 G01 Y-20.000 *

N74 G01 X-11.000 *

N75 G01 Y20.000 *

N76 G01 X-9.500 *

N77 G01 Y-20.000 *

N78 G01 X-8.000 *

N79 G01 Y20.000 *

N80 G01 X-6.500 *

N81 G01 Y-20.000 *

N82 G01 X-5.000 *

N83 G01 Y20.000 *

N84 G01 X-3.500 *

N85 G01 Y-20.000 *

N86 G01 X-2.000 *

N87 G01 Y20.000 *

N88 G01 X-0.500 *

N89 G01 Y-20.000 *

N90 G01 X1.000 *

N91 G01 Y20.000 *

N92 G01 X2.500 *

N93 G01 Y-20.000 *

N94 G01 X4.000 *

N95 G01 Y20.000 *

N96 G01 X5.500 *

N97 G01 Y-20.000 *

N98 G01 X7.000 *

N99 G01 Y20.000 *

N100 G01 X8.500 *

N101 G01 Y-20.000 *

N102 G01 X10.000 *

N103 G01 Y20.000 *

N104 G01 X11.500 *

N105 G01 Y-20.000 *

N106 G01 X13.000 *

N107 G01 Y20.000 *

N108 G01 X14.500 *

N109 G01 Y-20.000 *

N110 G01 X16.000 *

N111 G01 Y20.000 *

N112 G01 X17.500 *

N113 G01 Y-20.000 *

N114 G01 X19.000 *

N115 G01 Y20.000 *

N116 G01 X20.000 *

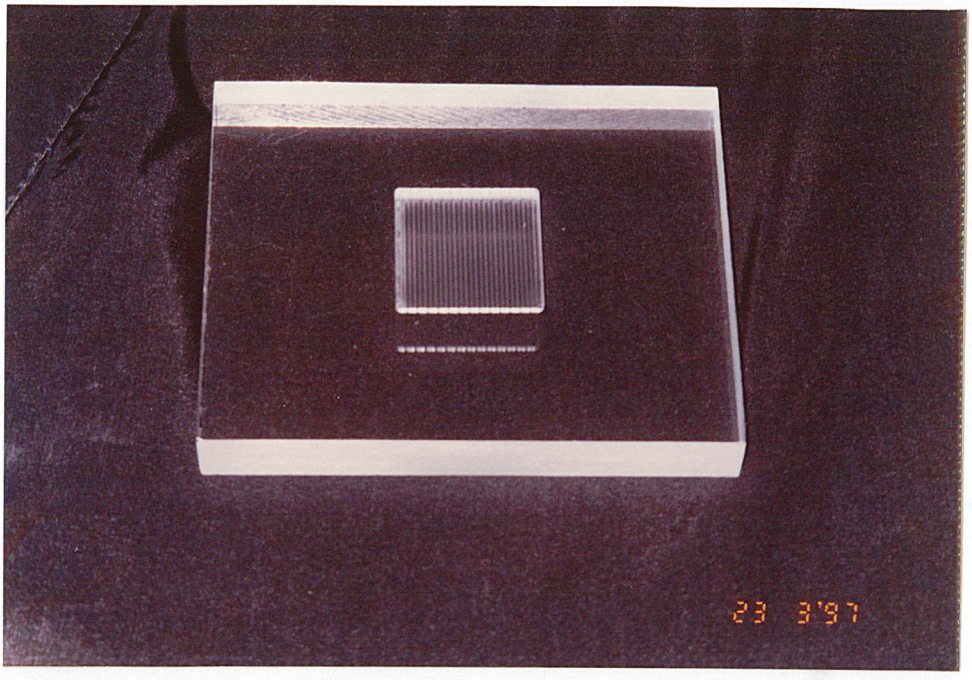
N117 G01 Y-20.000 *

N118 G00 Z10.0 *

N119 G00 X0.0 Y0.0 *

N120 M02 *

ได้ผลดังนี้



จากผลการทดลองที่ได้เมื่อนำมาทำการวิเคราะห์ สามารถสรุปได้ว่าเครื่องกัดแนวดิ่ง CNC ที่ทำขึ้นนี้มีค่า Error ทั้งหมดประมาณ ± 15 ถึง ± 17 ไมครอน ซึ่งค่า Error ที่หามาได้นี้มาจากการอ่านค่าตำแหน่งที่ส่งมาจากการ์ดควบคุมว่าผิดพลาดจากที่ได้กำหนดไว้เท่าใด

สาเหตุของการเกิด Error สามารถแบ่งออกได้เป็น

1. Error จากส่วน Software ซึ่งมาจากการที่โปรแกรมมีการปัดเศษทิ้งไป ระหว่างการคำนวณ รวมไปถึงการแปลงหน่วย ยกตัวอย่างเช่น 1 inch $\approx$ 25.4 mm เป็นต้น ซึ่งค่า Error เหล่านี้จะสะสมไปเรื่อย ๆ ขณะที่มีการเรียกใช้

2. Error จากส่วน Hardware เกิดขึ้นจากสาเหตุหลายประการ ดังนี้

- การสั่นของ Spindle อันเนื่องมาจากสายพานที่ใช้เกิดการหย่อน หรือ ค่า PID Filter ที่ใช้ไม่ดีพอ ต้องทำการเปลี่ยนค่าเหล่านี้ โดยทดลองไปเรื่อยๆ จนได้ค่าที่ดีที่สุด
- ค่า Backlash ของ Ball Screw ที่นำมาใช้

3. Error จากสภาพแวดล้อม เช่น แรงสั่นสะเทือนที่เกิดจากรถบรรทุก , รถไฟ ที่วิ่งผ่าน

บทที่ 8

สรุปและข้อเสนอแนะ

ในส่วนของการทำวิทยานิพนธ์ในปีการศึกษานี้ ได้ทำการพัฒนาในส่วนของโปรแกรมเป็นหลัก โดยสามารถพัฒนาให้มีขีดความสามารถในการใช้งานเท่าเทียมกับการใช้เครื่อง CNC ทั่วไป ข้อดีของการใช้เครื่อง CNC บนพื้นฐานไมโครคอมพิวเตอร์ก็คือ ความยืดหยุ่นในการใช้งาน โดยในเครื่อง CNC ทั่วไป สามารถรับ Code ที่จะทำการกัดได้เพียงรูปแบบใดรูปแบบหนึ่ง ยกตัวอย่างเช่น เครื่อง Fanuc ก็สามารถใช้ได้แต่ G-Code ของเครื่อง Fanuc เท่านั้น เครื่อง Heidenhain ก็สามารถใช้ได้แต่ G-Code ของเครื่อง Heidenhain เท่านั้น ไม่สามารถใช้ Code ต่างรูปแบบกันได้ ในขณะที่เครื่อง CNC บนพื้นฐานไมโครคอมพิวเตอร์ สามารถแก้จุดด้อยตรงส่วนนี้ได้ นอกจากนี้ตัวโปรแกรมที่จัดทำขึ้นยังสามารถพัฒนาต่อไปได้ ไม่หยุดอยู่กับที่เหมือนเครื่อง CNC ทั่วไป

ปัญหาและอุปสรรคที่พบระหว่างการทำงาน

1.Breaker ที่ใช้ในการควบคุม input ไฟ AC 380 V 3 เฟส ถูกต่อพ่วงเข้ากับอุปกรณ์หลายตัวทำให้กระแสไฟที่ไหลเข้าการ์ดควบคุมไม่คงที่ในบางครั้งเนื่องจากอุปกรณ์บางตัวมีการดึงกระแสไปใช้มากเกินไป เมื่อกระแสไม่คงที่ทำให้อุปกรณ์บางตัว เช่น Driver ซึ่งไวต่อการเปลี่ยนแปลงของกระแสมากเกิดการทำงานผิดพลาด จนถึงขั้นชำรุดเสียหายได้

2.ถ้ากระแสไฟที่จ่ายให้คอมพิวเตอร์เกิดดับขณะที่ตัวเครื่อง CNC กำลังทำงานอยู่ จะทำให้เครื่อง CNC อยู่ในสภาวะที่ขาดการควบคุม ทำให้อาจเกิดอุบัติเหตุขึ้นมาได้

3.เนื่องจากข้อจำกัดของโปรแกรม ซึ่งเป็นโปรแกรมที่ทำการ Run บน Dos ทำให้ไม่สามารถพัฒนาโปรแกรมในส่วนของการทำงานแบบ Multitasking ได้ เช่น การเปลี่ยนค่า Feed rate หรือ Spindel rate ระหว่างทำการกัดชิ้นงาน

แนวทางการแก้ไขปัญหา

1.ควรจะมีการแยกตัว Breaker ของอุปกรณ์แต่ละตัวออกจากกัน เพื่อไม่ให้การทำงานของอุปกรณ์ตัวใดตัวหนึ่งมีผลกระทบต่อการทำงานของอุปกรณ์ตัวอื่น

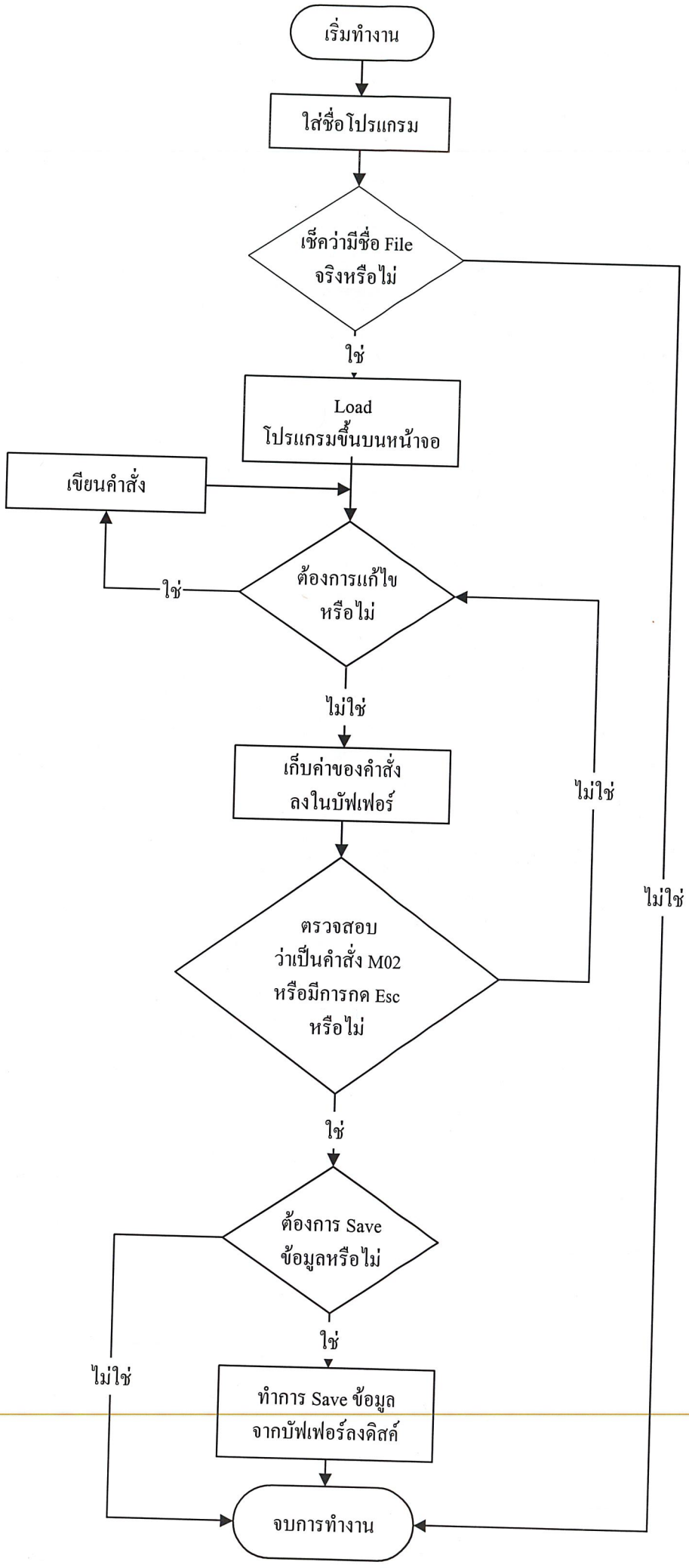
2.ควรมีการติดตั้งอุปกรณ์สำรองไฟฉุกเฉินเอาไว้ใช้ เพื่อป้องกันการเสียหายที่จะเกิดขึ้นกับอุปกรณ์ต่าง ๆ ได้

3.ถ้าต้องการทำการพัฒนาโปรแกรมมากไปกว่านี้ อาจจะต้องมีการเขียนโปรแกรมขึ้นมาใหม่โดยทำการเขียนโปรแกรมบน Window ซึ่งจะทำให้การพัฒนาในส่วน Multitasking นี้ทำได้ง่ายขึ้น

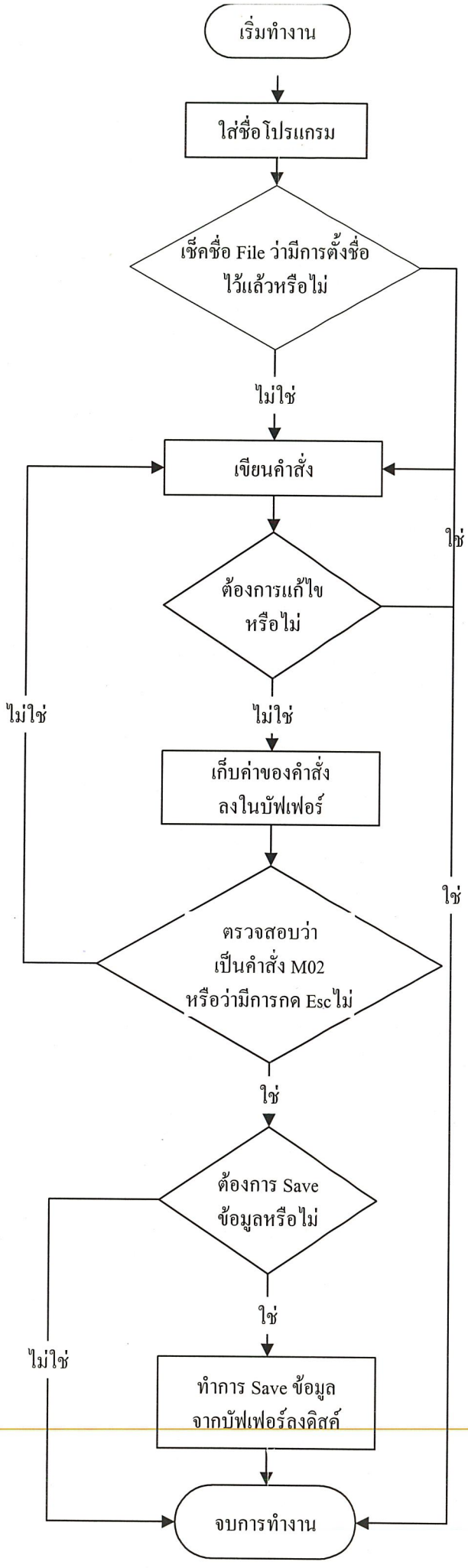
เอกสารอ้างอิง

1. กวิน สนธิเพิ่มพูน , รายงานการวิจัยฉบับสมบูรณ์ เรื่อง การสร้างส่วนควบคุมเครื่องกัดแนวดิ่ง ซีเอ็นซี ระยะที่ 1 , ศูนย์เทคโนโลยีอิเล็กทรอนิกส์และคอมพิวเตอร์แห่งชาติ , 2534.
2. Larry Horath , Computer Numerical Control Programming of Machines , 1993.
3. FANUC LTD , FANUC OM-MODEL A operator's manual , FANUC LTD , 1985.
4. PMD , 5650 Board Technical Reference Version 0.2 , Technology 80 Inc. , 1994.
5. INDRAMAT , MAC Servo Drives with TDM and KDS servo drive modules , INDRAMAT , 1994.
6. ชาลี ตระการกูล , เทคโนโลยีซีเอ็นซี , สมาคมส่งเสริมเทคโนโลยี่ (ไทย-ญี่ปุ่น) , 2538.
7. ธันวา ศรีประโมง , การเขียนโปรแกรมภาษาซี สำหรับวิศวกรรม พิมพ์ครั้งที่ 2 , โครงการตำราวิชาการ มหาวิทยาลัยเทคโนโลยีมหานคร

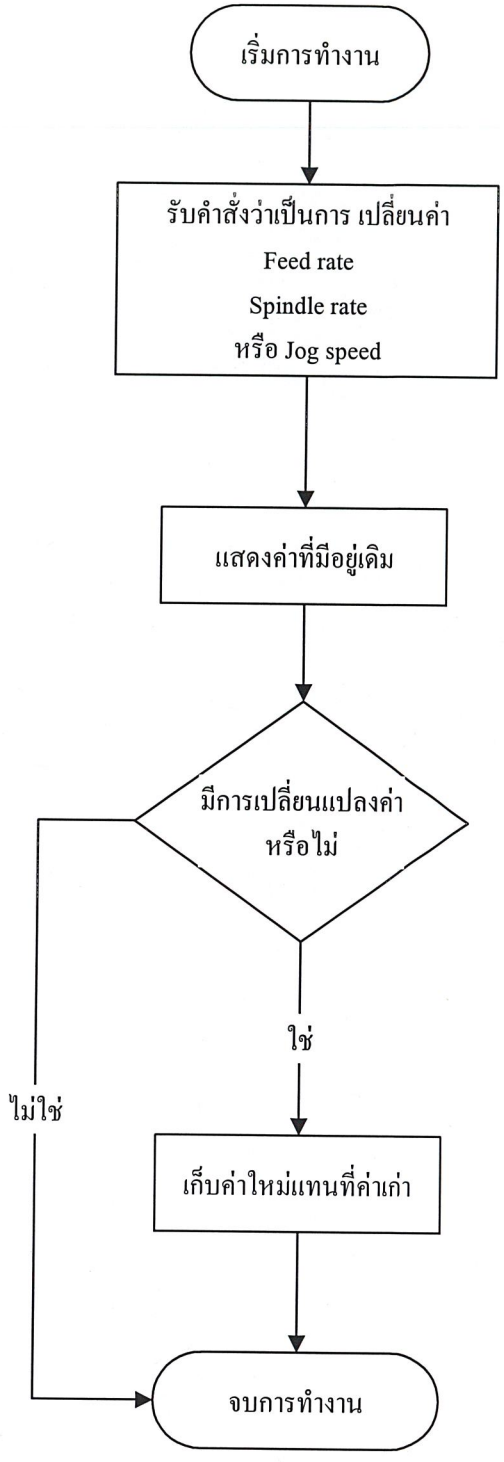
ภาคผนวก ก.



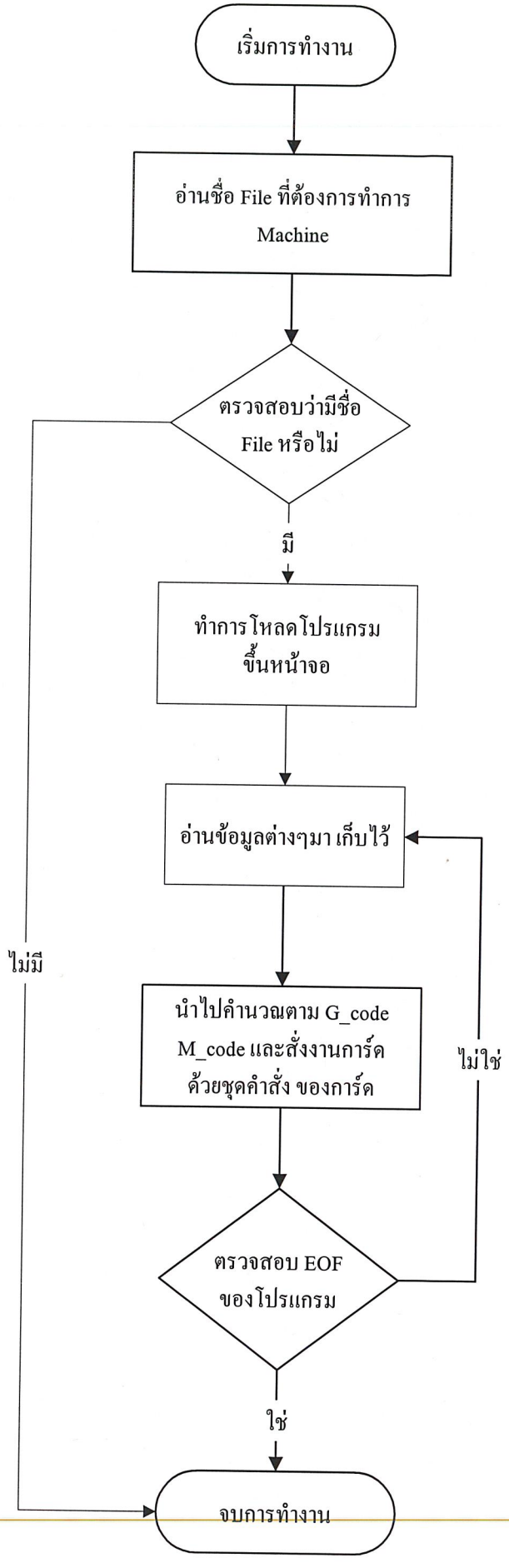
รูปที่ 1 ผังแสดงการทำงานของโหมดการ Edit program



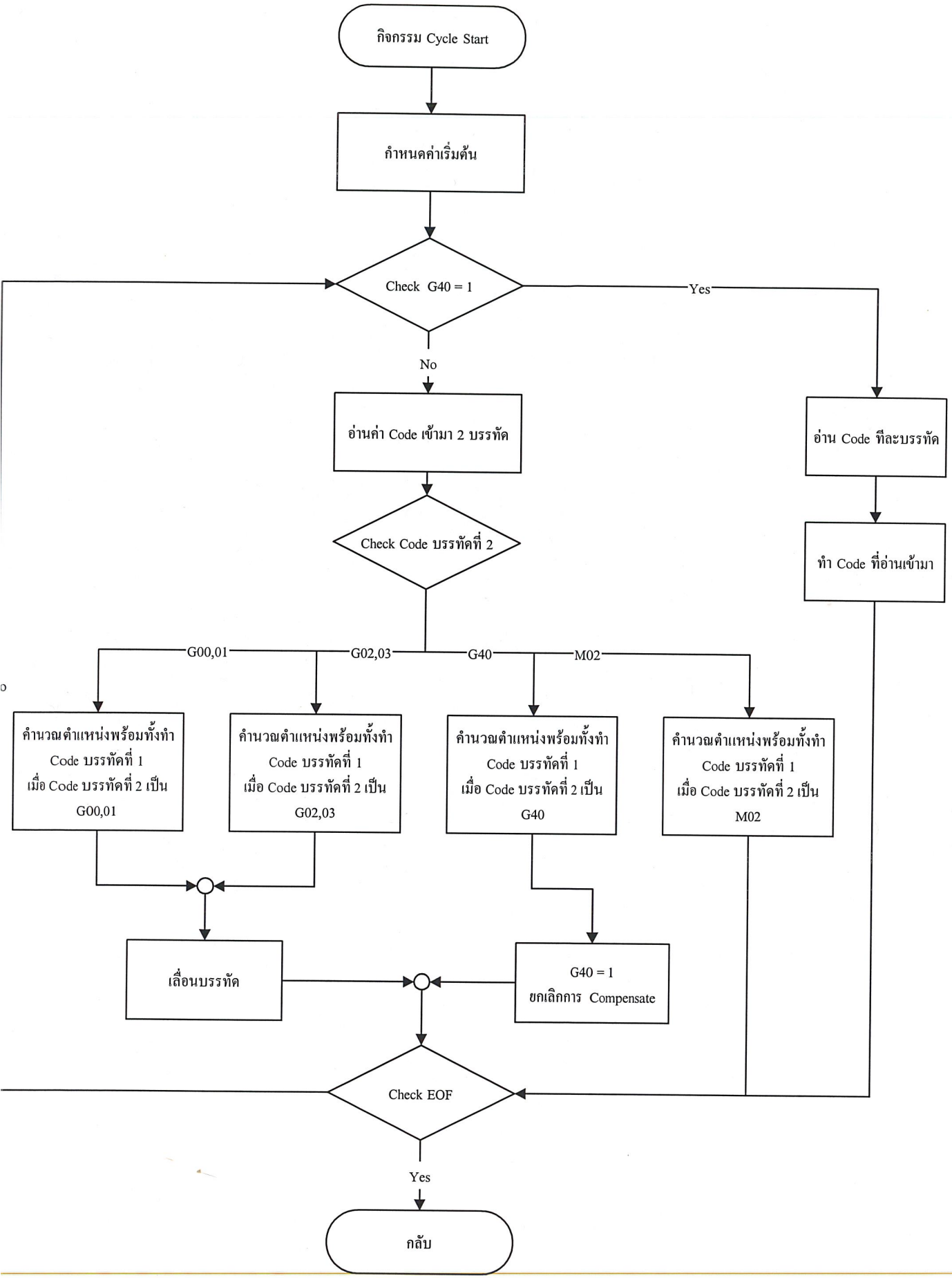
รูปที่ 2 ผังแสดงการทำงานของโหมดการ Edit program



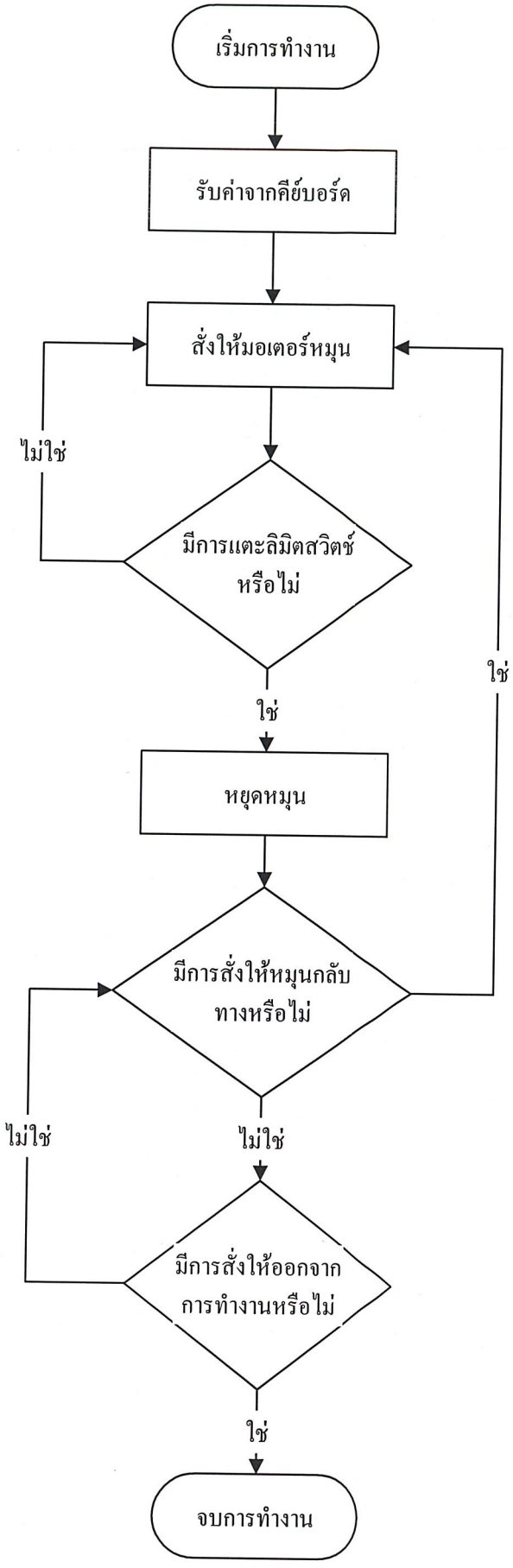
รูปที่ 3 ผังแสดงการทำงานของ โหมดความจำ



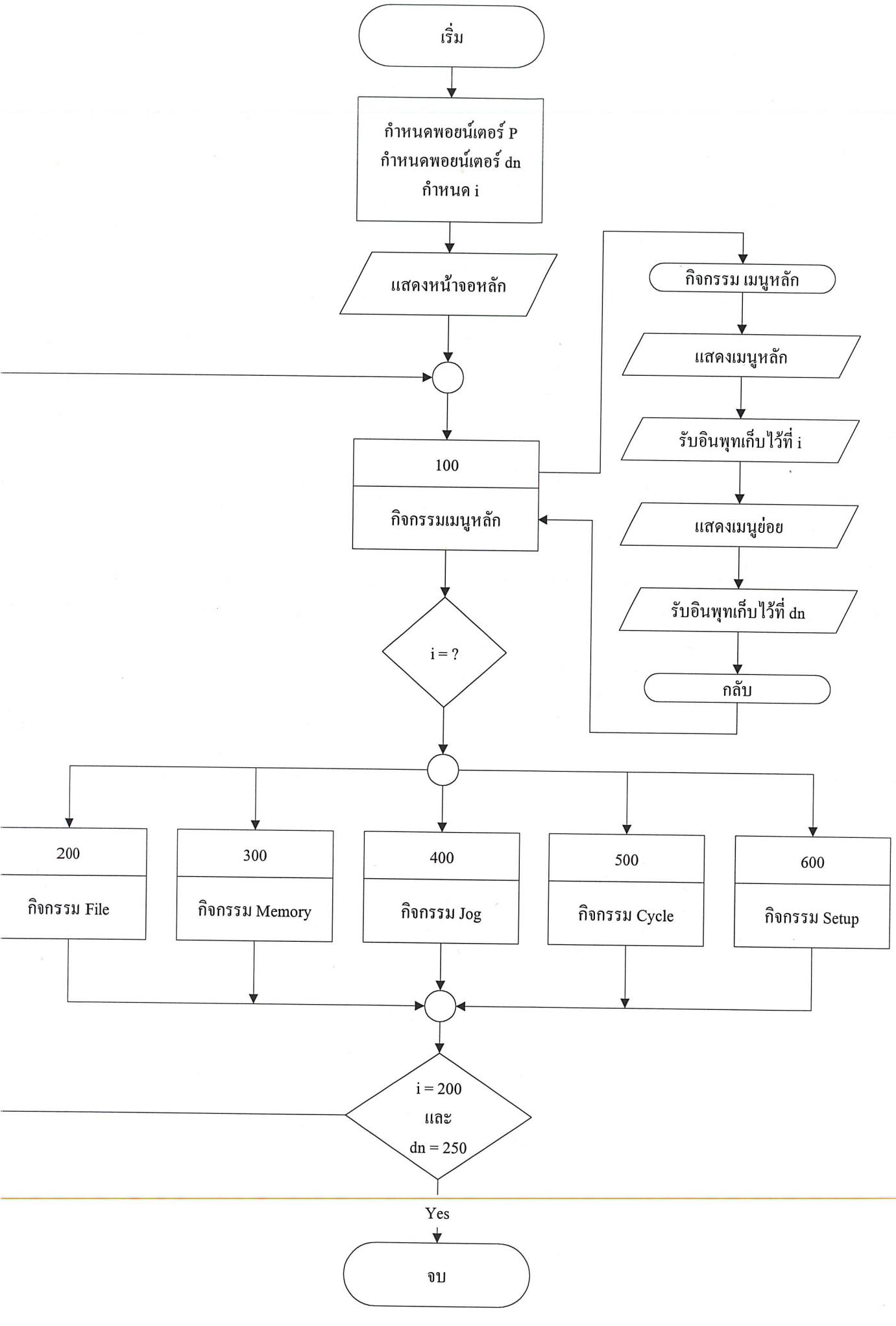
รูปที่ 4 ผังแสดงการทำงานของโหมดอัตโนมัติ



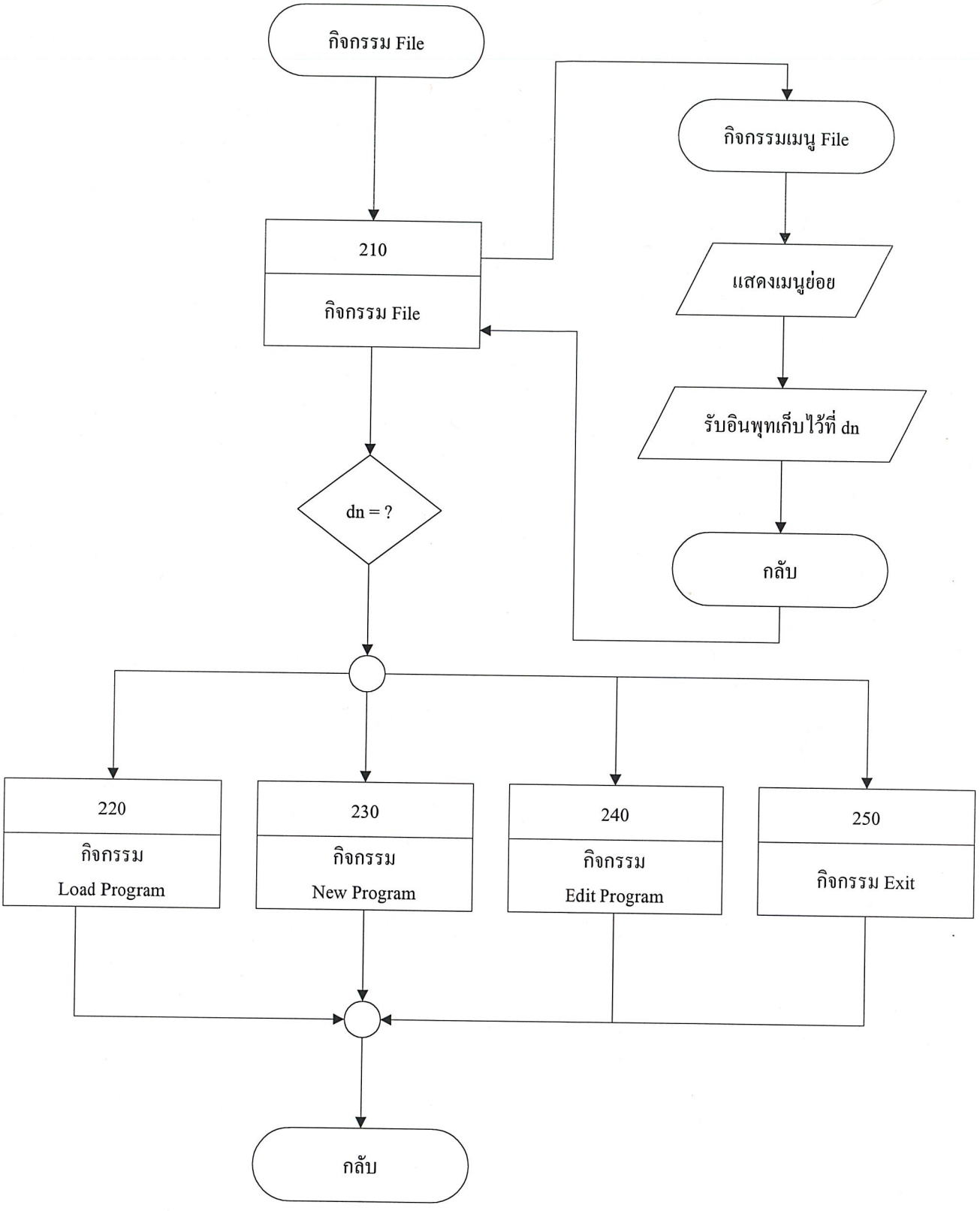
รูปที่ 5 ผังแสดงการทำงานของ Compiler.C



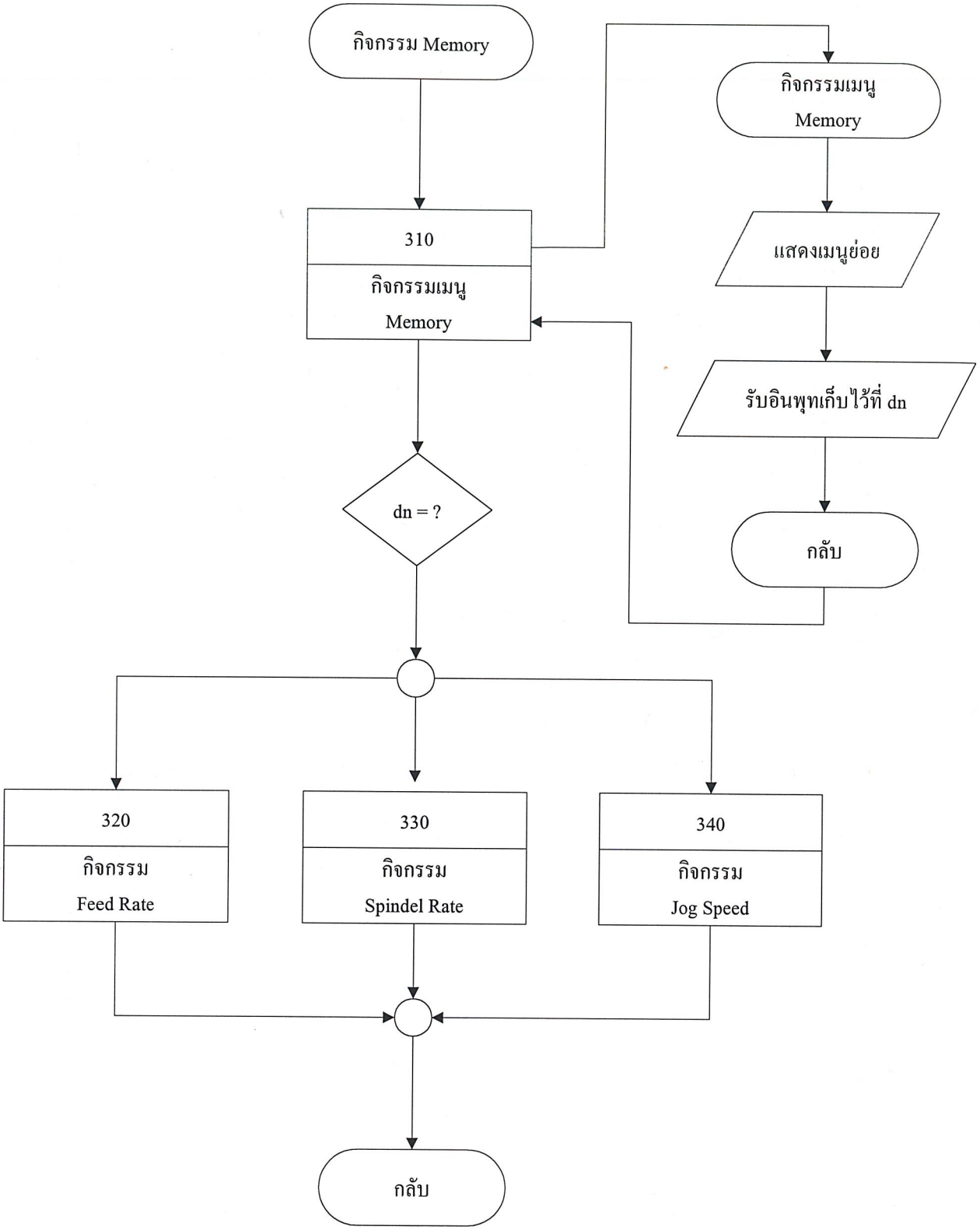
รูปที่ 6 ผังแสดงการทำงานของโหมดแบบบังคับเอง



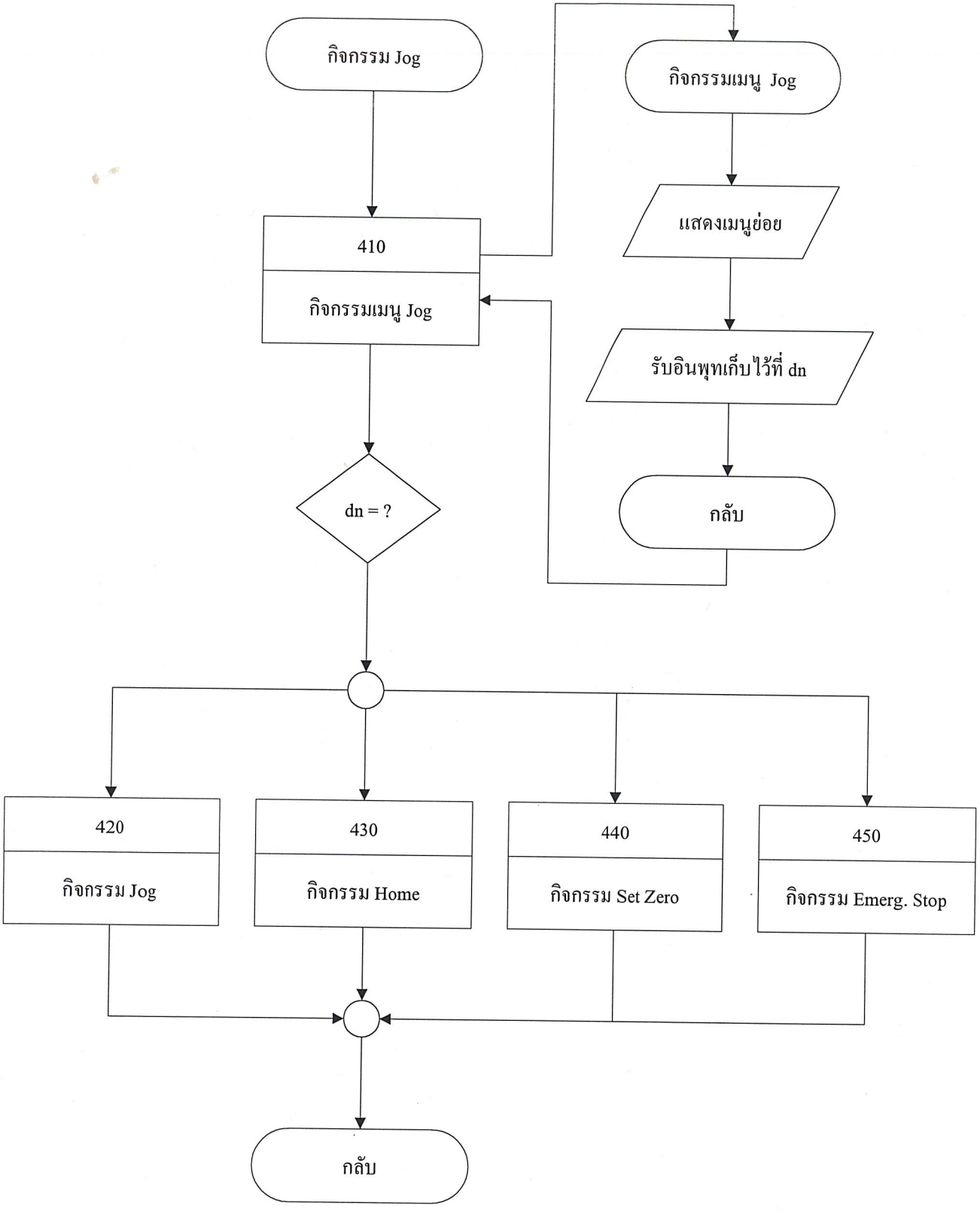
รูปที่ 7 ผังแสดงการทำงานของ Monitor.C



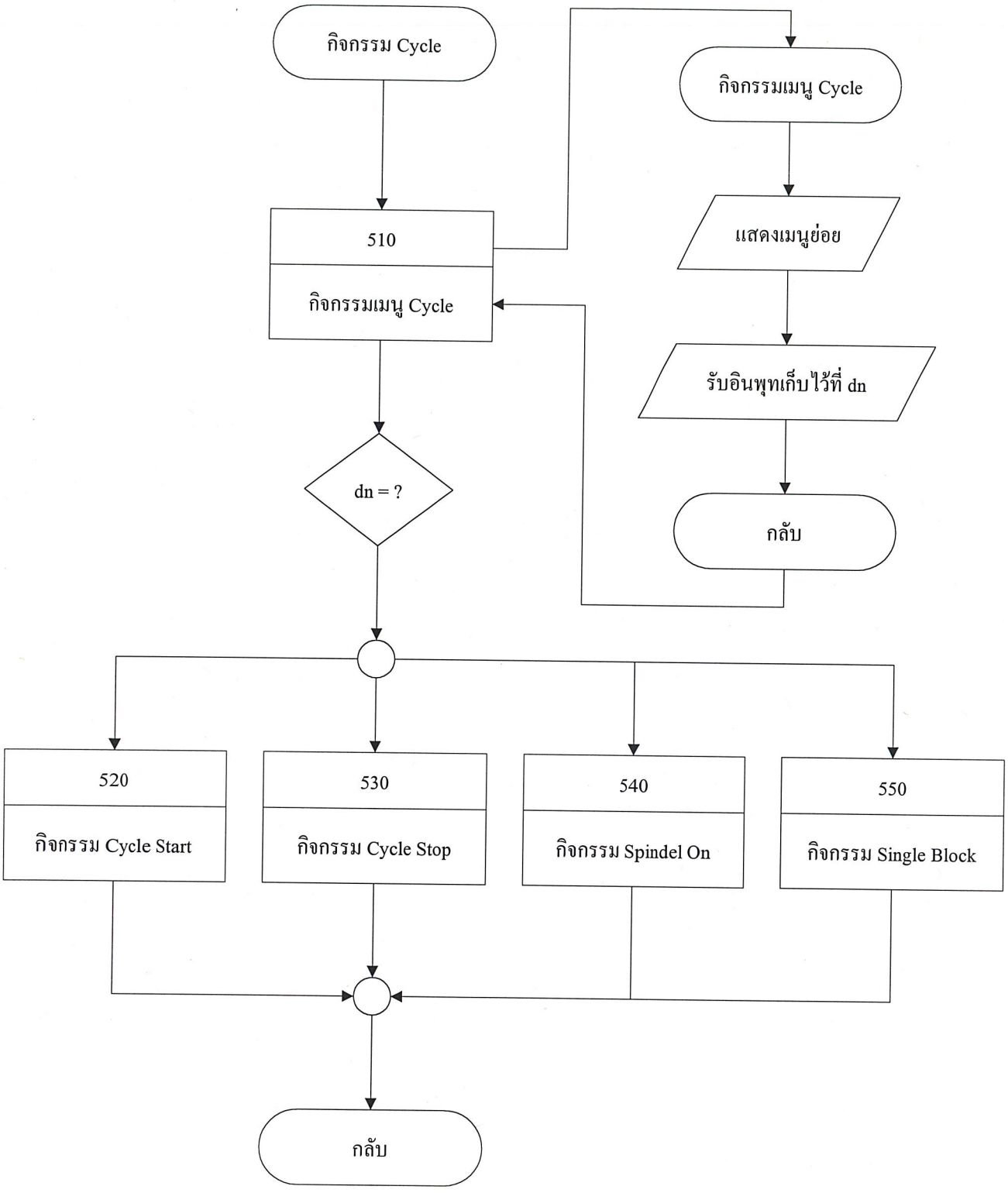
รูปที่ 8 ผังแสดงการทำงานของ Monitor.C



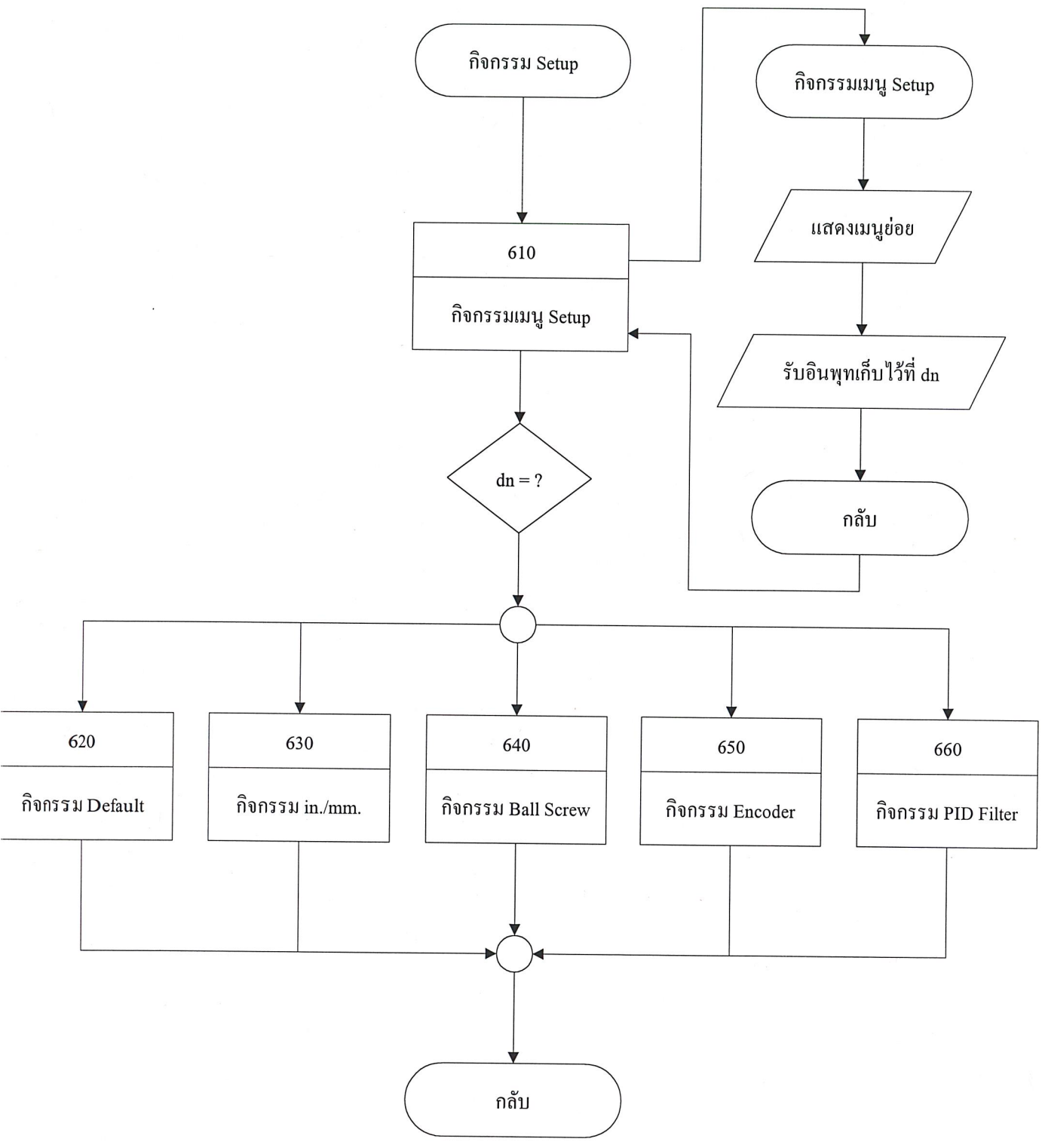
รูปที่ 9 ผังแสดงการทำงานของ Monitor.C



รูปที่ 10 ผังแสดงการทำงานของ Monitor.C



รูปที่ 11 ผังแสดงการทำงานของ Monitor.C



รูปที่ 12 ผังแสดงการทำงานของ Monitor.C

ภาคผนวก ข.

COMPILER.C

```
#include<conio.h>
#include<stdio.h>
#include<math.h>
#include"mc1401.h"
#include"hrdwr.h"
#include"host_io.h"
#include<dos.h>
#include<graphics.h>

#define norm 113
extern int tog,startn,endn,single;
void change(int axis)
{
   switch(axis)
   {
    case 0:
          set_1(0);
          break;
    case 1:
          set_2(0);
          break;
    case 2:
          set_3(0);
          break;
    case 3:
          set_4(0);
          break;
    default:
          break;
   }
}

void compiler(char nme[12])
{       FILE    *fp;
        char    ch,xi,yi; /* nme[12] */
        int     show,axis1=0,axis2=1,axis3=2,plane=1,i,line[4],speed[4];
        int     G40=1,G41=0,G42=0,Shift_ch=0,G91=0,G98=1,Group[4],Gcode[4],Mcode[4];
        int   ctr=1,start=0,No_comp=1;
        float   pos1[4],pos2[4],pos3[4],loc1,loc2,loc3,cut_in,r_level,i_level,
```

```
            cntr1,cntr2,m,arc,feed[4],radius[4],d_spindle;
        double  temp1,temp2,a,b,c,d,j,k,dpos1,dpos2,angle1,angle2,rc,rc1,rc2,
            a1,b1,abs_ab,abs_1,abs_2;
        double  x1,y1,x2,y2,x3,y3,x4,y4,X,Y,XX,YY,X1,X2,Y1,Y2,
            tempX,tempY,s,t,r,r1,r2,n,l,p;
        float   multiplier=1,deep,P;
        float   refx=0,refy=0,refz=0;
        long    feedrate[4];

        pos1[0] = 0 ;
        pos2[0] = 0 ;
        pos3[0] = 0 ;
        Group[1] = -1;
        Group[2] = -1;
        Gcode[1] = -1;
        Gcode[2] = -1;
        Mcode[1] = -1;
        Mcode[2] = -1;

window(3,3,78,11);
textcolor(14);
textbackground(0);
clrscr();
fp=fopen(nme,"r");
showfile(fp);
fclose(fp);
show=15;
fp=fopen(nme,"r");

if(single==1)
{while( fscanf(fp," %c",&ch) != EOF)
 {switch(ch)
 {
 case 'N' :fscanf(fp,"%5d",&line[0]);
            break;
 case '*' :show++;
}
 if(startn==line[0])break;
}
```

```
 refresh(show);
}

while( fscanf(fp," %c",&ch) != EOF)
 {
        if (No_comp == 1) /* Compensate Off */
        {
        while(ch != '*')
        {
        switch (ch)
        {
          case 'N':
                fscanf(fp,"%5d",&line[0]);
                break;
          case 'R':
                if (Gcode[0] == 2 || Gcode[0] == 3)
                 fscanf(fp,"%7f",&radius[0]);
                else
                 fscanf(fp,"%7f",&r_level);
                break;
          case 'F':
                fscanf(fp,"%7f",&feed[0]);
                feedrate[0] = (long)(feed[0]*100000/558);
                window(45,21,78,21);
                textattr(norm);
                clrscr();
                textcolor(LIGHTGREEN);
                textbackground(BLUE);
                gotoxy(1,1);
                if( tog== 0 )cprintf(" Feed Rate    : %8.3f mm/min",feed[0]);
                else cprintf(" Feed Rate    : %8.3f in/min",feed[0]);
                textattr(norm);
                cursoroff();
                break;
        case 'S':
                fscanf(fp,"%4d",&speed[0]);
                M03(speed[0]);
                window(45,22,78,22);
                textattr(norm);
```

```
            clrscr();
            textcolor(LIGHTGREEN);
            textbackground(BLUE);
            gotoxy(1,1);cprintf(" Spindle Rate : %4d    rpm   ",speed[0]);
            textattr(norm);
            cursoroff();
            break;
     case 'D':
            fscanf(fp,"%7f",&d_spindle);
            break;
     case 'P':
            fscanf(fp,"%7f",&P);
            break;
     case 'Q':
            fscanf(fp,"%7f",&cut_in);
            break;
     case 'X':
            axis1 = 0;
            fscanf(fp,"%8f",&pos1[0]);
            break;
     case 'Y':
            axis2 = 1;
            fscanf(fp,"%8f",&pos2[0]);
            break;
     case 'Z':
            axis2 = 2;
            i_level = pos3[0];
            fscanf(fp,"%8f",&pos3[0]);
            break;
     case 'G':
            fscanf(fp,"%2d",&Gcode[0]);
            switch(Gcode[0])
            {
            /*  group 04  */
             case 98: /* Back to Initial Level */
                    G98 = 1;
                    break;
             case 99: /* Back to Reference Level */
                    G98 = 0;
```

```
            break;
        /*  group 07  */
         case 41:  /* Compensate Left */
                ctr = 1;
                Shift_ch=1;
                G41 = 1;
                G42 = 0;
                G40 = 0;
                break;
         case 42:  /* Compensate Right */
                ctr = 1;
                Shift_ch=1;
                G41 = 0;
                G42 = 1;
                G40 = 0;
                break;
        /*  group 06  */
         case 20:  /* Inch */
                multiplier = 25.4*400;
                break;
         case 21:  /* Mill */
                multiplier = 400;
                break;
        /*  group 03  */
         case 90:  /* Absolute */
                G91 = 0;
                break;
         case 91:  /* Incremental */
                G91 = 1;
                break;
        /*  group  02  */
         case 17:
                plane = 1;
                break;
         case 18:
                plane = 2;
                break;
         case 19:
                plane = 3;
```

```
                    break;
                } /* end of switch Gcode */
                No_comp = G40; /* If G41 or G42 not be used G40 always = 1 */
                break;
          case 'M':
                fscanf(fp," %2d",&Mcode[0]);
                switch(Mcode[0])
                {
                 case 2:
                        M02(0);
                        break;
                 case 3:
                        break;
                 case 4:
                        break;
                 case 5:
                        break;
                } /* end of switch Mcode */
                break;
          default:
                window(3,12,78,12);
                clrscr();
                fscanf(fp," %5d",&line[0]);
                /*printf("Error on Program Line %d\n",line);
                exit(2);*/
                break;
         } /* end of switch ch */
          fscanf(fp," %c",&ch);
        } /* end of while(ch =! '*') */
        } /* end of No_comp = 1 */

/*************************************************************************/

          if (No_comp == 0)  /* Compensate On */
          {
           if (Shift_ch==1)
            {
             fscanf(fp," %c",&ch);
             Shift_ch = 0;
```

```
 show++;
 refresh(show);
 }
while (ctr != 3)
{
 switch (ch)
 {
 case '*':
        ctr = ctr+1;
        break;
 case 'N':
        fscanf(fp,"%5d",&line[ctr]);
        break;
 case 'R':
        fscanf(fp,"%7f",&radius[ctr]);
        break;
 case 'F':
        fscanf(fp,"%7f",&feed[ctr]);
        feedrate[ctr] = (long)(feed[ctr]*100000/558);
        window(45,21,78,21);
        textattr(norm);
        clrscr();
        textcolor(LIGHTGREEN);
        textbackground(BLUE);
        gotoxy(1,1);
        if( tog== 0 )cprintf(" Feed Rate    : %8.3f mm/min",feed[1]);
        else cprintf(" Feed Rate    : %8.3f in/min",feed[1]);
        textattr(norm);
        cursoroff();
        break;
case 'S':
        fscanf(fp,"%4d",&speed[ctr]);
        M03(speed[ctr]);
        window(45,22,78,22);
        textattr(norm);
        clrscr();
        textcolor(LIGHTGREEN);
        textbackground(BLUE);
        gotoxy(1,1);cprintf(" Spindle Rate : %4d     rpm    ",speed[1]);
```

```
            textattr(norm);
            cursoroff();
            break;
    case 'D':
            fscanf(fp,"%7f",&d_spindle);
            break;
    case 'X':
            axis1 = 0;
            fscanf(fp,"%8f",&pos1[ctr]);
            break;
    case 'Y':
            axis2 = 1;
            fscanf(fp,"%8f",&pos2[ctr]);
            break;
    case 'Z':
            axis2 = 2;
            fscanf(fp,"%8f",&pos3[ctr]);
            break;
    case 'G':
            fscanf(fp,"%2d",&Gcode[ctr]);
             switch(Gcode[ctr])
             {
             /*  Group 06  */
              case 0:
                      Group[ctr] = 1;
                      break;
              case 1:
                      Group[ctr] = 1;
                      break;
              case 2:
                      Group[ctr] = 2;
                      break;
              case 3:
                      Group[ctr] = 2;
                      break;
              case 20:  /* Inch */
                      multiplier = 25.4*400;
                      break;
              case 21:  /* Mill */
```

```
                multiplier = 400;
                break;
            case 40:
                ctr=3;
                Shift_ch=0;
                G40=1;
                G41=0;
                G42=0;
                break;
            /*  Group 03  */
            case 90:  /* Absolute */
                G91 = 0;
                break;
            case 91:  /* Incremental */
                G91 = 1;
                break;
           /*  Group  02  */
            case 17:
                plane = 1;
                break;
            case 18:
                plane = 2;
                break;
            case 19:
                plane = 3;
                break;
           } /* End of Switch Gcode */
        break;
  case 'M':
        fscanf(fp," %2d",&Mcode[ctr]);
        break;
  default:
        break;
 } /* End of Switch ch */
 fscanf(fp," %c",&ch);
} /* End of While (ctr != 3) */
ctr = 2;
} /* End of No_comp = 0 */
```

```
/*****************************************************************************/

    /* When (ch) = '*' or ctr = 3 Execute this part */

    if (No_comp == 1) /* Compensate Off */
    {
    switch(Gcode[0]) /* group 01 for movement */
          {
          case 0:
                    axis1=0;
                    axis2=1;
                    axis3=2;
                    if (G91)
                    {
                     change(axis1);
                     pos1[0] = pos1[0] + (get_pos(0)/multiplier);
                     change(axis2);
                     pos2[0] = (get_pos(0)/multiplier)-pos2[0];
                     change(axis3);
                     pos3[0] = (get_pos(0)/multiplier)-pos3[0];
                    }
                    G00(axis1,axis2,axis3,pos1[0]+refx,pos2[0]+refy,pos3[0]+refz);
                    break;

          case 1:
                    axis1=0;
                    axis2=1;
                    axis3=2;
                    if (G91)
                    {
                     change(axis1);
                     pos1[0] = pos1[0] + (get_pos(0)/multiplier);
                     change(axis2);
                     pos2[0] = (get_pos(0)/multiplier)-pos2[0];
                     change(axis3);
                     pos3[0] = (get_pos(0)/multiplier)-pos3[0];
                    }
                    G01(axis1,axis2,axis3,pos1[0]+refx,pos2[0]+refy,pos3[0]+refz,feedrate[0]);
                    break;
```

```
case 2:
        if(plane == 1 )
        { axis1 = 0;
          axis2 = 1;
          xi='X';
          yi='Y';
        }
        if(plane == 2 )
        { axis1 = 2;
          axis2 = 0;
          pos2[0] = pos1[0];
          pos1[0] = pos3[0];
          xi='Z';
          yi='X';
        }
        if(plane == 3)
        { axis1 = 1;
          axis2 = 2;
          pos1[0] = pos2[0];
          pos2[0] = pos3[0];
          xi='Y';
          yi='Z';
        }

        if (G91)
        {
         change(axis1);
         pos1[0] = pos1[0] + (get_pos(0)/multiplier);
         change(axis2);
         pos2[0] = (get_pos(0)/multiplier)-pos2[0];
         change(axis3);
         pos3[0] = (get_pos(0)/multiplier)-pos3[0];
        }

        G02(axis1,axis2,pos1[0],pos2[0],radius[0],feedrate[0]);
        break;

case 3 :
```

```
if(plane == 1 )
{ axis1 = 0;
  axis2 = 1;
  xi='X';
  yi='Y';
}
if(plane == 2 )
{ axis1 = 2;
  axis2 = 0;
  pos2[0] = pos1[0];
  pos1[0] = pos3[0];
  xi='Z';
  yi='X';
}
if(plane == 3)
{ axis1 = 1;
  axis2 = 2;
  pos1[0] = pos2[0];
  pos2[0] = pos3[0];
  xi='Y';
  yi='Z';
}

if (G91)
{
 change(axis1);
 pos1[0] = pos1[0] + (get_pos(0)/multiplier);
 change(axis2);
 pos2[0] = pos2[0] + (get_pos(0)/multiplier);
}

G03(axis1,axis2,pos1[0],pos2[0],radius[0],feedrate[0]);
break;

case 73 :
G01(axis1,axis2,axis3,pos1[0]+refx,pos2[0]+refy,i_level+refz,feedrate[0]);
G01(axis1,axis2,axis3,pos1[0]+refx,pos2[0]+refy,r_level+refz,feedrate[0]);
pos3[1] = r_level;
do
```

```
{
pos3[3] = pos3[1] - pos3[0];
pos3[2] = pos3[1] - (cut_in/3);
pos3[1] = pos3[1] - cut_in;
 if (pos3[3] >= cut_in)
 {
  G01(axis1,axis2,axis3,pos1[0]+refx,pos2[0]+refy,pos3[1]+refz,feedrate[0]);
  G01(axis1,axis2,axis3,pos1[0]+refx,pos2[0]+refy,pos3[2]+refz,feedrate[0]);
 }
 if (pos3[3] < cut_in)
  G01(axis1,axis2,axis3,pos1[0]+refx,pos2[0]+refy,pos3[0]+refz,feedrate[0]);
}
while (pos3[3] != 0);
switch(G98)
{
 case 0:
        G01(axis1,axis2,axis3,pos1[0]+refx,pos2[0]+refy,r_level+refz,feedrate[0]);
        break;
 case 1:
        G01(axis1,axis2,axis3,pos1[0]+refx,pos2[0]+refy,i_level+refz,feedrate[0]);
        break;
}
break;

case 81 :
        G00(axis1,axis2,axis3,pos1[0]+refx,pos2[0]+refy,i_level+refz);
        G00(axis1,axis2,axis3,pos1[0]+refx,pos2[0]+refy,r_level+refz);
        G01(axis1,axis2,axis3,pos1[0]+refx,pos2[0]+refy,pos3[0]+refz,feedrate[0]);
        switch(G98)
        {
         case 0:
                G00(axis1,axis2,axis3,pos1[0]+refx,pos2[0]+refy,r_level+refz);
                break;
         case 1:
                G00(axis1,axis2,axis3,pos1[0]+refx,pos2[0]+refy,i_level+refz);
                break;
        }
        break;
```

```
case 82 :
        G00(axis1,axis2,axis3,pos1[0]+refx,pos2[0]+refy,i_level+refz);
        G00(axis1,axis2,axis3,pos1[0]+refx,pos2[0]+refy,r_level+refz);
        G01(axis1,axis2,axis3,pos1[0]+refx,pos2[0]+refy,pos3[0]+refz,feedrate[0]);
        P=P*1000;
        delay(P);
        switch(G98)
        {
         case 0:
                G00(axis1,axis2,axis3,pos1[0]+refx,pos2[0]+refy,r_level+refz);
                break;
         case 1:
                G00(axis1,axis2,axis3,pos1[0]+refx,pos2[0]+refy,i_level+refz);
                break;
        }
        break;

case 83 :
        G01(axis1,axis2,axis3,pos1[0]+refx,pos2[0]+refy,i_level+refz,feedrate[0]);
        G01(axis1,axis2,axis3,pos1[0]+refx,pos2[0]+refy,r_level+refz,feedrate[0]);
        pos3[1] = r_level;
        do
        {
         pos3[3] = pos3[1] - pos3[0];
         pos3[2] = r_level;
         pos3[1] = pos3[1] - cut_in;
          if (pos3[3] >= cut_in)
          {
           G01(axis1,axis2,axis3,pos1[0]+refx,pos2[0]+refy,pos3[1]+refz,feedrate[0]);
           G01(axis1,axis2,axis3,pos1[0]+refx,pos2[0]+refy,pos3[2]+refz,feedrate[0]);
          }
          if (pos3[3] < cut_in)
           G01(axis1,axis2,axis3,pos1[0]+refx,pos2[0]+refy,pos3[0]+refz,feedrate[0]);
        }
        while (pos3[3] != 0);
        switch(G98)
        {
         case 0:
                G01(axis1,axis2,axis3,pos1[0]+refx,pos2[0]+refy,r_level+refz,feedrate[0]);
```

```
                break;
            case 1:
                G01(axis1,axis2,axis3,pos1[0]+refx,pos2[0]+refy,i_level+refz,feedrate[0]);
                break;
        }
        break;

case 85 :
        G00(axis1,axis2,axis3,pos1[0]+refx,pos2[0]+refy,i_level+refz);
        G00(axis1,axis2,axis3,pos1[0]+refx,pos2[0]+refy,r_level+refz);
        G01(axis1,axis2,axis3,pos1[0]+refx,pos2[0]+refy,pos3[0]+refz,feedrate[0]);
        switch(G98)
        {
            case 0:
                G01(axis1,axis2,axis3,pos1[0]+refx,pos2[0]+refy,r_level+refz,feedrate[0]);
                break;
            case 1:
                G01(axis1,axis2,axis3,pos1[0]+refx,pos2[0]+refy,i_level+refz,feedrate[0]);
                break;
        }
        break;

case 86 :
        G00(axis1,axis2,axis3,pos1[0]+refx,pos2[0]+refy,i_level+refz);
        G00(axis1,axis2,axis3,pos1[0]+refx,pos2[0]+refy,r_level+refz);
        G01(axis1,axis2,axis3,pos1[0]+refx,pos2[0]+refy,pos3[0]+refz,feedrate[0]);
        M05(0);
        switch(G98)
        {
            case 0:
                G00(axis1,axis2,axis3,pos1[0]+refx,pos2[0]+refy,r_level+refz);
                break;
            case 1:
                G00(axis1,axis2,axis3,pos1[0]+refx,pos2[0]+refy,i_level+refz);
                break;
        }
        break;

case 89 :
```

```
                G00(axis1,axis2,axis3,pos1[0]+refx,pos2[0]+refy,i_level+refz);
                G00(axis1,axis2,axis3,pos1[0]+refx,pos2[0]+refy,r_level+refz);
                G01(axis1,axis2,axis3,pos1[0]+refx,pos2[0]+refy,pos3[0]+refz,feedrate[0]);
                delay(P*1000);
                switch(G98)
                {
                 case 0:
                        G01(axis1,axis2,axis3,pos1[0]+refx,pos2[0]+refy,r_level+refz,feedrate[0]);
                        break;
                 case 1:
                        G01(axis1,axis2,axis3,pos1[0]+refx,pos2[0]+refy,i_level+refz,feedrate[0]);
                        break;
                }
                break;

        case 92 :

                refx = pos1[0];
                refy = pos2[0];
                refz = pos3[0];
        } /* End switch Gcode group 01 */
        show++;
        refresh(show);
        if(endn==line[0])break;
    } /* End G40 = 1 No Compensate */

/*************************************************************************/

    if (No_comp == 0) /* Compensate On */
    {
    if ((Gcode[2] != 40) && (Mcode[2] != 2))
    {
    switch(Group[1]) /* Group 01 for Movement */
    {
        case 1: /* Group 1 : G00,G01 */
                axis1=0;
                axis2=1;
                axis3=2;
                line[0]   =   line[1];
```

```
Gcode[0]  =  Gcode[1];
feed[0]   =  feed[1];
feedrate[0] =  feedrate[1];
rc = d_spindle/2;

if (start == 1)
{
temp1 = pos1[3];
temp2 = pos2[3];
}
if (start == 0)
{
temp1 = pos1[0];
temp2 = pos2[0];
start=1;
}

if ((Gcode[2] == 0) || (Gcode[2] == 1))
{
/* Programmed1 Path */
a = pos1[1]-temp1;
b = pos2[1]-temp2;
if (a==0)
 angle1 = 90;
else
 angle1 = atan2(b,a);

/* Programmed2 Path */
c = pos1[2]-pos1[1];
d = pos2[2]-pos2[1];
if (c==0)
 angle2 = 90;
else
 angle2 = atan2(d,c);

/* Signed Vector */
j = pos1[2]-temp1;
k = pos2[2]-temp2;
```

```
if (fabs((angle2-angle1)) != 90)
{
 dpos1 = fabs(rc*((sin(angle2+angle1))/2)/((cos(angle2-angle1))/2));
 dpos2 = fabs(rc*((cos(angle2+angle1))/2)/((cos(angle2-angle1))/2));
}

if (fabs((angle2-angle1)) == 90)
{
 dpos1 = rc;
 dpos2 = rc;
}

if (G41 == 1)  /* Compensate Left */
{
 if ((j >= 0) && (k >= 0))
 {
  dpos1 = -dpos1;
  dpos2 = dpos2;
 }
 if ((j >= 0) && (k < 0))
 {
  dpos1 = dpos1;
  dpos2 = dpos2;
 }
 if ((j < 0) && (k >= 0))
 {
  dpos1 = -dpos1;
  dpos2 = -dpos2;
 }
 if ((j < 0) && (k < 0))
 {
  dpos1 = dpos1;
  dpos2 = -dpos2;
 }
}

if (G42 == 1)  /* Compensate Right */
{
 if ((j >= 0) && (k >= 0))
```

```
  {
   dpos1 = dpos1;
   dpos2 = -dpos2;
  }
  if ((j >= 0) && (k < 0))
  {
   dpos1 = -dpos1;
   dpos2 = -dpos2;
  }
  if ((j < 0) && (k >= 0))
  {
   dpos1 = dpos1;
   dpos2 = dpos2;
  }
  if ((j < 0) && (k < 0))
  {
   dpos1 = -dpos1;
   dpos2 = dpos2;
  }
 }

 pos1[0] = pos1[1] + dpos1;
 pos2[0] = pos2[1] + dpos2;

 } /* End if ((Gcode[2] == 0) || (Gcode[2] == 1)) */

 if ((Gcode[2] == 2) || (Gcode[2] == 3 ))
 {
 /* Find Center of Circle */
 r  = radius[2];
 x1 = pos1[1];
 y1 = pos2[1];
 x2 = pos1[2];
 y2 = pos2[2];

 if ((y2-y1) == 0)
 {
  s = (x1+x2)/2;
  t = (y1+y2)/2;
```

```
    l = x2-x1;
    p = pow(r,2)-pow((l/2),2);
    x3= s;
    y3= t-sqrt(p);
    x4= s;
    y4= t+sqrt(p);
   }
   else
   {
    n = -((x2-x1)/(y2-y1));
    s = (x1+x2)/2;
    t = (y1+y2)/2;
    l = sqrt((pow((x2-x1),2)+pow((y2-y1),2)));
    p = pow(r,2)-pow((l/2),2);
    x3 = sqrt(p/(1+pow(n,2)));
    y3 = n*x3;
    x4 = -x3;
    y4 = -y3;
    x3 = s+x3;
    y3 = t+y3;
    x4 = s+x4;
    y4 = t+y4;
   }

  if (Gcode[2] == 2)
  {
  if ( r > 0 )
  {
  if ( (((x2-x1)*(y3-y1))-((y2-y1)*(x3-x1))) < 0 )
   {
    X = x3;
    Y = y3;
   }
   else
   {
    X = x4;      /* find center finish*/
    Y = y4;
   }
  }
```

```
 if ( r < 0)
 {
 if ( (((x2-x1)*(y3-y1))-((y2-y1)*(x3-x1))) >= 0 )
 {
  X = x3;
  Y = y3;
 }
 else
 {
  X = x4;     /* find center finish*/
  Y = y4;
 }
 }
 } /* End Gcode[2] == 2 */

 if (Gcode[2] == 3)
 {
 if ( r > 0 )
 {
 if ( (((x2-x1)*(y3-y1))-((y2-y1)*(x3-x1))) >= 0 )
 {
  X = x3;
  Y = y3;
 }
 else
 {
  X = x4;     /* find center finish*/
  Y = y4;
 }
}
if ( r < 0)
{
 if ( (((x2-x1)*(y3-y1))-((y2-y1)*(x3-x1))) < 0 )
 {
  X = x3;
  Y = y3;
}
else
{
```

```
  X = x4;     /* find center finish*/
  Y = y4;
 }
}
 rc = -rc;
} /* End Gcode[2] == 3 */

if (G41 == 1)  /* Compensate Left */
 r = fabs(r)+rc;
if (G42 == 1)  /* Compensate Right */
 r = fabs(r)-rc;

if (start == 1)
{
 if ((temp1==pos1[1])||(temp2==pos2[1]))
 {
  if (temp1==pos1[1])
  {
   X1 = pos1[0];
   X2 = pos1[0];
   Y1 = (2*Y+sqrt(pow((-2*Y),2)-4*(pow(Y,2)+pow((X1-X),2)-pow(r,2))))/2;
   Y2 = (2*Y-sqrt(pow((-2*Y),2)-4*(pow(Y,2)+pow((X1-X),2)-pow(r,2))))/2;
  } /* End temp1==pos1[1] */

  if (temp2==pos2[1])
  {
   Y1 = pos2[0];
   Y2 = pos2[0];
   X1 = (2*X+sqrt(pow((-2*X),2)-4*(pow(X,2)+pow((Y1-Y),2)-pow(r,2))))/2;
   X2 = (2*X-sqrt(pow((-2*X),2)-4*(pow(X,2)+pow((Y1-Y),2)-pow(r,2))))/2;
  } /* End temp2==pos2[1] */
 } /* End(temp1==pos1[1])||(temp2==pos2[1]) */
 else
 {
  a = (pos2[1]-temp2)/(pos1[1]-temp1); /* dy/dx */
  b = pos2[0]-(a*pos1[0]);  /* y-ax */

  X1 = (-2*(a*b-a*Y-X)+sqrt(pow((2*(a*b-a*Y-X)),2)-4*(1+pow(a,2))
     *(pow(b,2)+pow(X,2)+pow(Y,2)-pow(r,2)-2*b*Y) ) )
```

```
        /(2*(1+pow(a,2)));
    Y1 = (a*X1)+b;

    X2 = (-2*(a*b-a*Y-X)-sqrt(pow((2*(a*b-a*Y-X)),2)-4*(1+pow(a,2))
        *(pow(b,2)+pow(X,2)+pow(Y,2)-pow(r,2)-2*b*Y) ) )
        /(2*(1+pow(a,2)));
    Y2 = (a*X2)+b;
   }
  } /* End start == 1 */

  if (start == 0)
  {
   a = pos1[1]-pos1[0];
   b = pos2[1]-pos2[0];
   abs_ab = sqrt(pow(a,2)+pow(b,2));
   a = a/abs_ab;
   b = b/abs_ab;

   if (G41 == 1)  /* Compensate Left */
   {
    a1 =  b*rc ;
    b1 = -a*rc ;
   }

   if (G42 == 1)  /* Compensate Right */
   {
    a1 = -b*rc ;
    b1 =  a*rc ;
   }

  temp1   = pos1[0];
  temp2   = pos2[0];
  pos1[0] = pos1[0] + a1;
  pos2[0] = pos2[0] + b1;

  if ((a1==0)||(b1==0))
  {
   if (a1==0)
   {
```

```
  X1 = temp1;
  X2 = temp1;
  Y1 = (2*Y+sqrt(pow((-2*Y),2)-4*(pow(Y,2)+pow((X1-X),2)-pow(r,2))))/2;
  Y2 = (2*Y-sqrt(pow((-2*Y),2)-4*(pow(Y,2)+pow((X1-X),2)-pow(r,2))))/2;
 } /* End a1==0 */

 if (b1==0)
 {
  Y1 = temp2;
  Y2 = temp2;
  X1 = (2*X+sqrt(pow((-2*X),2)-4*(pow(X,2)+pow((Y1-Y),2)-pow(r,2))))/2;
  X2 = (2*X-sqrt(pow((-2*X),2)-4*(pow(X,2)+pow((Y1-Y),2)-pow(r,2))))/2;
 }
} /* End a1==0 || b1==0 */
else
{
 /* y=ax+b */
 a = (pos2[1]-temp2)/(pos1[1]-temp1); /* dy/dx */
 b = pos2[0]-(a*pos1[0]);  /* y-ax */

 X1 = (-2*(a*b-a*Y-X)+sqrt(pow((2*(a*b-a*Y-X)),2)-4*(1+pow(a,2))
     *(pow(b,2)+pow(X,2)+pow(Y,2)-pow(r,2)-2*b*Y) ) )
     /(2*(1+pow(a,2)));
 Y1 = (a*X1)+b;

 X2 = (-2*(a*b-a*Y-X)-sqrt(pow((2*(a*b-a*Y-X)),2)-4*(1+pow(a,2))
     *(pow(b,2)+pow(X,2)+pow(Y,2)-pow(r,2)-2*b*Y) ) )
     /(2*(1+pow(a,2)));
 Y2 = (a*X2)+b;
}
} /* End start == 0 */

 abs_1 = sqrt(pow((pos1[0]-X1),2)+pow((pos2[0]-Y1),2));
 abs_2 = sqrt(pow((pos1[0]-X2),2)+pow((pos2[0]-Y2),2));

 if (abs_1 < abs_2)
 {
 pos1[0] = X1;
 pos2[0] = Y1;
```

```
  }
  else
  {
   pos1[0] = X2;
   pos2[0] = Y2;
  }
} /* End (Gcode[2] == 2 , 3) */

switch (Gcode[1])
{
case 0:
        if (G91)
        {
        change(axis1);
        pos1[0] = pos1[0] + (get_pos(0)/multiplier);
        change(axis2);
        pos2[0] = (get_pos(0)/multiplier)-pos2[0];
        change(axis3);
        pos3[0] = (get_pos(0)/multiplier)-pos3[0];
        }
        G00(axis1,axis2,axis3,pos1[0]+refx,pos2[0]+refy,pos3[0]+refz);
        break;
case 1:
        if (G91)
        {
        change(axis1);
        pos1[0] = pos1[0] + (get_pos(0)/multiplier);
        change(axis2);
        pos2[0] = (get_pos(0)/multiplier)-pos2[0];
        change(axis3);
        pos3[0] = (get_pos(0)/multiplier)-pos3[0];
        }
        G01(axis1,axis2,axis3,pos1[0]+refx,pos2[0]+refy,pos3[0]+refz,feedrate[0]);
        break;
}

line[1]   =   line[2];
Group[1]  =       Group[2];
Group[2]  =   -1;
```

```
        Gcode[1]  =  Gcode[2];
        Gcode[2]  =  -1;
        Mcode[1]  =  Mcode[2];
        Mcode[2]  =  -1;
        pos1[3]   =  pos1[1];
        pos2[3]   =  pos2[1];
        pos1[1]   =  pos1[2];
        pos2[1]   =  pos2[2];
        speed[1]  =  speed[2];
        radius[1] =  radius[2];
        feed[1]   =  feed[2];
        feedrate[1] =  feedrate[2];
        break;

case 2: /* Group 2 : G02,G03 */
        axis1 = 0;
        axis2 = 1;
        axis3 = 2;
        line[0]   =  line[1];
        Gcode[0]  =  Gcode[1];
        feed[0]   =  feed[1];
        feedrate[0] =  feedrate[1];
        radius[0] =  radius[1];
        rc = d_spindle/2;

        if (start == 1)
        {
        temp1 = pos1[3];
        temp2 = pos2[3];
        }
        if (start == 0)
        {
        temp1 = pos1[0];
        temp2 = pos2[0];
        start=1;
        }

        /* Find Center of Circle 1 */
        r1 = radius[1];
```

```
x1 = temp1;
y1 = temp2;
x2 = pos1[1];
y2 = pos2[1];

if ((y2-y1) == 0)
{
 s = (x1+x2)/2;
 t = (y1+y2)/2;
 l = x2-x1;
 p = pow(r1,2)-pow((l/2),2);
 x3= s;
 y3= t-sqrt(p);
 x4= s;
 y4= t+sqrt(p);
}
else
{
 n = -((x2-x1)/(y2-y1));
 s = (x1+x2)/2;
 t = (y1+y2)/2;
 l = sqrt((pow((x2-x1),2)+pow((y2-y1),2)));
 p = pow(r1,2)-pow((l/2),2);
 x3 = sqrt(p/(1+pow(n,2)));
 y3 = n*x3;
 x4 = -x3;
 y4 = -y3;
 x3 = s+x3;
 y3 = t+y3;
 x4 = s+x4;
 y4 = t+y4;
}

if (Gcode[1] == 2)
{
if ( r1 > 0 )
{
if ( (((x2-x1)*(y3-y1))-((y2-y1)*(x3-x1))) < 0 )
{
```

```
  X = x3;
  Y = y3;
  }
  else
  {
  X = x4;     /* find center finish*/
  Y = y4;
  }
 }
 if ( r1 < 0)
 {
 if ( (((x2-x1)*(y3-y1))-((y2-y1)*(x3-x1))) >= 0 )
 {
  X = x3;
  Y = y3;
 }
 else
 {
  X = x4;     /* find center finish*/
  Y = y4;
 }
}
rc1 = rc;
} /* End Gcode[1] == 2 */

if (Gcode[1] == 3)
{
if ( r1 > 0 )
{
 if ( (((x2-x1)*(y3-y1))-((y2-y1)*(x3-x1))) >= 0 )
 {
  X = x3;
  Y = y3;
}
else
{
 X = x4;     /* find center finish*/
 Y = y4;
}
```

```
 }
 if ( r1 < 0)
 {
  if ( (((x2-x1)*(y3-y1))-((y2-y1)*(x3-x1))) < 0 )
  {
   X = x3;
   Y = y3;
  }
  else
  {
   X = x4;      /* find center finish*/
   Y = y4;
  }
 }
 rc1 = -rc;
} /* End Gcode[1] == 3 */

if ((Gcode[2] == 0) || (Gcode[2] == 1))
{
 a = pos1[1]-pos1[2];
 b = pos2[1]-pos2[2];
 abs_ab = sqrt(pow(a,2)+pow(b,2));
 a = a/abs_ab;
 b = b/abs_ab;

 if (G41 == 1)  /* Compensate Left */
 {
  a1 =  b*fabs(rc1)  ;
  b1 = -a*fabs(rc1)  ;
  r1 =  fabs(r1)+rc1 ;
 }

 if (G42 == 1)  /* Compensate Right */
 {
  a1 = -b*fabs(rc1)  ;
  b1 =  a*fabs(rc1)  ;
  r1 =  fabs(r1)-rc1 ;
 }
 pos1[3] = pos1[2] + a1;
```

```
 pos2[3] = pos2[2] + b1;

 if ((pos1[2]==pos1[1])||(pos2[2]==pos2[1]))
 {
  if (pos1[2]==pos1[1])
  {
   X1 = pos1[3];
   X2 = X1;
   Y1 = (2*Y+sqrt(pow((-2*Y),2)-4*(pow(Y,2)+pow((X1-X),2)-pow(r1,2))))/2;
   Y2 = (2*Y-sqrt(pow((-2*Y),2)-4*(pow(Y,2)+pow((X1-X),2)-pow(r1,2))))/2;
  } /* End pos1[2]==pos1[1] */

  if (pos2[2]==pos2[1])
  {
   Y1 = pos2[3];
   Y2 = Y1;
   X1 = (2*X+sqrt(pow((-2*X),2)-4*(pow(X,2)+pow((Y1-Y),2)-pow(r1,2))))/2;
   X2 = (2*X-sqrt(pow((-2*X),2)-4*(pow(X,2)+pow((Y1-Y),2)-pow(r1,2))))/2;
  } /* End pos2[2]==pos2[1] */
 } /* End(pos1[2]==pos1[1])||(pos2[2]==pos2[1]) */
 else
 {
 /* y=ax+b */
 a = (pos2[2]-pos2[1])/(pos1[2]-pos1[1]); /* dy/dx */
 b = pos2[3]-(a*pos1[3]);  /* y-ax */

  X1 = (-2*(a*b-a*Y-X)+sqrt(pow((2*(a*b-a*Y-X)),2)-4*(1+pow(a,2))
     *(pow(b,2)+pow(X,2)+pow(Y,2)-pow(r1,2)-2*b*Y) ) )
     /(2*(1+pow(a,2)));
  Y1 = (a*X1)+b;

  X2 = (-2*(a*b-a*Y-X)-sqrt(pow((2*(a*b-a*Y-X)),2)-4*(1+pow(a,2))
     *(pow(b,2)+pow(X,2)+pow(Y,2)-pow(r1,2)-2*b*Y) ) )
     /(2*(1+pow(a,2)));
  Y2 = (a*X2)+b;
 }
 abs_1 = sqrt(pow((pos1[3]-X1),2)+pow((pos2[3]-Y1),2));
 abs_2 = sqrt(pow((pos1[3]-X2),2)+pow((pos2[3]-Y2),2));
```

```
    if (abs_1 < abs_2)
    {
     pos1[0] = X1;
     pos2[0] = Y1;
    }
    else
    {
     pos1[0] = X2;
     pos2[0] = Y2;
    }
  } /* End Gcode[2] == 0 || 1 */

  if ((Gcode[2] == 2) || (Gcode[2] == 3 ))
  {
  /* Find Center of Circle 2 */
  r2 = radius[2];
  x1 = pos1[1];
  y1 = pos2[1];
  x2 = pos1[2];
  y2 = pos2[2];

  if ((y2-y1) == 0)
  {
   s = (x1+x2)/2;
   t = (y1+y2)/2;
   l = x2-x1;
   p = pow(r2,2)-pow((l/2),2);
   x3= s;
   y3= t-sqrt(p);
   x4= s;
   y4= t+sqrt(p);
  }
  else
  {
   n = -((x2-x1)/(y2-y1));
   s = (x1+x2)/2;
   t = (y1+y2)/2;
   l = sqrt((pow((x2-x1),2)+pow((y2-y1),2)));
   p = pow(r2,2)-pow((l/2),2);
```

```
    x3 = sqrt(p/(1+pow(n,2)));
    y3 = n*x3;
    x4 = -x3;
    y4 = -y3;
    x3 = s+x3;
    y3 = t+y3;
    x4 = s+x4;
    y4 = t+y4;
   }

   if (Gcode[2] == 2)
   {
   if ( r2 > 0 )
   {
   if ( (((x2-x1)*(y3-y1))-((y2-y1)*(x3-x1))) < 0 )
   {
    XX = x3;
    YY = y3;
   }
   else
   {
    XX = x4;      /* find center finish*/
    YY = y4;
   }
  }
  if ( r2 < 0)
  {
   if ( (((x2-x1)*(y3-y1))-((y2-y1)*(x3-x1))) >= 0 )
   {
    XX = x3;
    YY = y3;
   }
   else
   {
    XX = x4;      /* find center finish*/
    YY = y4;
   }
  }
 rc2 = rc;
```

```
 } /* End Gcode[2] == 2 */

 if (Gcode[2] == 3)
 {
 if ( r2 > 0 )
 {
 if ( (((x2-x1)*(y3-y1))-((y2-y1)*(x3-x1))) >= 0 )
 {
  XX = x3;
  YY = y3;
 }
 else
 {
  XX = x4;     /* find center finish*/
  YY = y4;
 }
}
if ( r2 < 0)
{
 if ( (((x2-x1)*(y3-y1))-((y2-y1)*(x3-x1))) < 0 )
 {
  XX = x3;
  YY = y3;
 }
 else
 {
  XX = x4;     /* find center finish*/
  YY = y4;
 }
}
 rc2 = -rc;
} /* End Gcode[2] == 3 */

if (G41 == 1)  /* Compensate Left */
{
r1 = fabs(radius[1])+rc1;
r2 = fabs(radius[2])+rc2;
}
```

```
if (G42 == 1)  /* Compensate Right */
{
 r1 = fabs(radius[1])-rc1;
 r2 = fabs(radius[2])-rc2;
}

if ((sqrt(pow((XX-X),2)+pow((YY-Y),2)))==(r1+r2))
{
 a = XX-X;
 b = YY-Y;
 abs_ab = sqrt(pow(a,2)+pow(b,2));
 a = a/abs_ab*r1;
 b = b/abs_ab*r1;
 pos1[0] = X+a;
 pos2[0] = Y+b;
}
else
{
 if ((XX==X)||(YY==Y))
 {
  if (XX==X)
  {
   Y1 = (pow(YY,2)-pow(Y,2)+pow(r1,2)-pow(r2,2))/(2*(YY-Y));
   Y2 = Y1;
   X1 = (2*X+sqrt(pow((-2*X),2)-4*(pow(X,2)+pow((Y1-Y),2)-pow(r1,2))))/2;
   X2 = (2*X-sqrt(pow((-2*X),2)-4*(pow(X,2)+pow((Y1-Y),2)-pow(r1,2))))/2;
  } /* End XX==X */

  if (YY==Y)
  {
   X1 = (pow(XX,2)-pow(X,2)+pow(r1,2)-pow(r2,2))/(2*(XX-X));
   X2 = X1;
   Y1 = (2*Y+sqrt(pow((-2*Y),2)-4*(pow(Y,2)+pow((X1-X),2)-pow(r1,2))))/2;
   Y2 = (2*Y-sqrt(pow((-2*Y),2)-4*(pow(Y,2)+pow((X1-X),2)-pow(r1,2))))/2;
  }
 } /* End XX==X || YY==Y */
 else
 {
 /* y=ax+b */
```

```
a = ((X-XX)/(YY-Y));
b = (pow(r1,2)-pow(r2,2)+pow(XX,2)+pow(YY,2)-pow(X,2)-pow(Y,2))/(2*(YY-Y));

X1 = (-2*(a*b-a*Y-X)+sqrt(pow((2*(a*b-a*Y-X)),2)-4*(1+pow(a,2))
    *(pow(b,2)+pow(X,2)+pow(Y,2)-pow(r1,2)-2*b*Y) ) )
    /(2*(1+pow(a,2)));
Y1 = (a*X1)+b;

X2 = (-2*(a*b-a*Y-X)-sqrt(pow((2*(a*b-a*Y-X)),2)-4*(1+pow(a,2))
    *(pow(b,2)+pow(X,2)+pow(Y,2)-pow(r1,2)-2*b*Y) ) )
    /(2*(1+pow(a,2)));
Y2 = (a*X2)+b;
}

abs_1 = sqrt(pow((pos1[1]-X1),2)+pow((pos2[1]-Y1),2));
abs_2 = sqrt(pow((pos1[1]-X2),2)+pow((pos2[1]-Y2),2));

if (abs_1 < abs_2)
{
 pos1[0] = X1;
 pos2[0] = Y1;
}
else
{
 pos1[0] = X2;
 pos2[0] = Y2;
}
} /* End Else */
} /* End Gcode[2] == 2 || 3 */

switch (Gcode[1])
{
case 2:
        if (G91)
        {
        change(axis1);
        pos1[0] = pos1[0] + (get_pos(0)/multiplier);
        change(axis2);
        pos2[0] = (get_pos(0)/multiplier)-pos2[0];
```

```
                change(axis3);
                pos3[0] = (get_pos(0)/multiplier)-pos3[0];
                }
                G02(axis1,axis2,pos1[0],pos2[0],r1,feedrate[0]);
                break;
        case 3:
                if (G91)
                {
                change(axis1);
                pos1[0] = pos1[0] + (get_pos(0)/multiplier);
                change(axis2);
                pos2[0] = (get_pos(0)/multiplier)-pos2[0];
                change(axis3);
                pos3[0] = (get_pos(0)/multiplier)-pos3[0];
                }
                G03(axis1,axis2,pos1[0],pos2[0],r1,feedrate[0]);
                break;
        }

        line[1]    =   line[2];
        Group[1]   =        Group[2];
        Group[2]   =   -1;
        Gcode[1]   =   Gcode[2];
        Gcode[2]   =   -1;
        Mcode[1]   =   Mcode[2];
        Mcode[2]   =   -1;
        pos1[3]    =   pos1[1];
        pos2[3]    =   pos2[1];
        pos1[1]    =   pos1[2];
        pos2[1]    =   pos2[2];
        speed[1]   =   speed[2];
        radius[1]  =   radius[2];
        feed[1]    =   feed[2];
        feedrate[1] =   feedrate[2];
        break;

} /* End Switch Gcode Group 01 for Movement */
} /* End ((Gcode[2] != 40) && (Mcode[2] != 2)) */
```

```
if ((Gcode[2]==40)||(Mcode[2] == 2))
{
switch(Group[1]) /* Group 01 for Movement */
{
    case 1: /* G00,G01 */
            axis1=0;
            axis2=1;
            axis3=2;
            line[0]    =   line[1];
            Gcode[0]   =   Gcode[1];
            feed[0]    =   feed[1];
            feedrate[0] =   feedrate[1];
            rc = d_spindle/2;

            if (start == 1)
            {
            temp1 = pos1[3];
            temp2 = pos2[3];
            }
            if (start == 0)
            {
            temp1 = pos1[0];
            temp2 = pos2[0];
            start=1;
            }

            /* Programmed1 Path */
            a = temp1-pos1[1];
            b = temp2-pos2[1];
            abs_ab = sqrt(pow(a,2)+pow(b,2));
            a = a/abs_ab;
            b = b/abs_ab;

            if (G41 == 1)  /* Compensate Left */
            {
             a1 =  b*rc ;
             b1 = -a*rc ;
            }
```

```
if (G42 == 1)  /* Compensate Right */
{
 a1 = -b*rc ;
 b1 =  a*rc ;
}

pos1[0] = pos1[1] + a1;
pos2[0] = pos2[1] + b1;

switch (Gcode[1])
{
case 0:
        if (G91)
        {
        change(axis1);
        pos1[0] = pos1[0] + (get_pos(0)/multiplier);
        change(axis2);
        pos2[0] = (get_pos(0)/multiplier)-pos2[0];
        change(axis3);
        pos3[0] = (get_pos(0)/multiplier)-pos3[0];
        }
        G00(axis1,axis2,axis3,pos1[0]+refx,pos2[0]+refy,pos3[0]+refz);
        break;
case 1:
        if (G91)
        {
        change(axis1);
        pos1[0] = pos1[0] + (get_pos(0)/multiplier);
        change(axis2);
        pos2[0] = (get_pos(0)/multiplier)-pos2[0];
        change(axis3);
        pos3[0] = (get_pos(0)/multiplier)-pos3[0];
        }
        G01(axis1,axis2,axis3,pos1[0]+refx,pos2[0]+refy,pos3[0]+refz,feedrate[0]);
        break;
}
break;

case 2: /* G02,G03 */
```

```
axis1 = 0;
axis2 = 1;
axis3 = 2;
line[0]    =   line[1];
Gcode[0]   =   Gcode[1];
feed[0]    =   feed[1];
feedrate[0] =  feedrate[1];
radius[0]  =   radius[1];
rc = d_spindle/2;

if (start == 1)
{
temp1 = pos1[3];
temp2 = pos2[3];
}
if (start == 0)
{
temp1 = pos1[0];
temp2 = pos2[0];
start=1;
}

/* Find Center of Circle */

r  = radius[0];
x1 = temp1;
y1 = temp2;
x2 = pos1[1];
y2 = pos2[1];

if ((y2-y1) == 0)
{
 s = (x1+x2)/2;
 t = (y1+y2)/2;
 l = x2-x1;
 p = pow(r,2)-pow((l/2),2);
 x3= s;
 y3= t-sqrt(p);
 x4= s;
```

```
    y4= t+sqrt(p);
   }
   else
   {
    n = -((x2-x1)/(y2-y1));
    s = (x1+x2)/2;
    t = (y1+y2)/2;
    l = sqrt((pow((x2-x1),2)+pow((y2-y1),2)));
    p = pow(r,2)-pow((l/2),2);
    x3 = sqrt(p/(1+pow(n,2)));
    y3 = n*x3;
    x4 = -x3;
    y4 = -y3;
    x3 = s+x3;
    y3 = t+y3;
    x4 = s+x4;
    y4 = t+y4;
   }

  if (Gcode[1] == 2)
  {
  if ( r > 0 )
  {
   if ( (((x2-x1)*(y3-y1))-((y2-y1)*(x3-x1))) < 0 )
   {
    X = x3;
    Y = y3;
   }
   else
   {
    X = x4;     /* find center finish*/
    Y = y4;
   }
  }
  if ( r < 0)
  {
   if ( (((x2-x1)*(y3-y1))-((y2-y1)*(x3-x1))) >= 0 )
   {
    X = x3;
```

```
  Y = y3;
 }
 else
 {
  X = x4;     /* find center finish*/
  Y = y4;
 }
}
} /* End Gcode[1] == 2 */

if (Gcode[1] == 3)
{
if ( r > 0 )
{
 if ( (((x2-x1)*(y3-y1))-((y2-y1)*(x3-x1))) >= 0 )
 {
  X = x3;
  Y = y3;
 }
 else
 {
  X = x4;     /* find center finish*/
  Y = y4;
 }
}
if ( r < 0)
{
 if ( (((x2-x1)*(y3-y1))-((y2-y1)*(x3-x1))) < 0 )
 {
  X = x3;
  Y = y3;
 }
 else
 {
  X = x4;     /* find center finish*/
  Y = y4;
 }
}
rc = -rc;
```

```
} /* End Gcode[1] == 3 */

a = pos1[1]-X;
b = pos2[1]-Y;
abs_ab = sqrt(pow(a,2)+pow(b,2));
a = a/abs_ab;
b = b/abs_ab;

if (G41 == 1)  /* Compensate Left */
{
 a = a*(fabs(r)+rc);
 b = b*(fabs(r)+rc);
 r = fabs(r)+rc;
}

if (G42 == 1)  /* Compensate Right */
{
 a = a*(fabs(r)-rc);
 b = b*(fabs(r)-rc);
 r = fabs(r)-rc;
}

pos1[0] = X + a;
pos2[0] = Y + b;

switch (Gcode[1])
{
case 2:
        if (G91)
        {
        change(axis1);
        pos1[0] = pos1[0] + (get_pos(0)/multiplier);
        change(axis2);
        pos2[0] = (get_pos(0)/multiplier)-pos2[0];
        change(axis3);
        pos3[0] = (get_pos(0)/multiplier)-pos3[0];
        }
        G02(axis1,axis2,pos1[0],pos2[0],r,feedrate[0]);
        break;
```

```
                case 3:
                        if (G91)
                        {
                        change(axis1);
                        pos1[0] = pos1[0] + (get_pos(0)/multiplier);
                        change(axis2);
                        pos2[0] = (get_pos(0)/multiplier)-pos2[0];
                        change(axis3);
                        pos3[0] = (get_pos(0)/multiplier)-pos3[0];
                        }
                        G03(axis1,axis2,pos1[0],pos2[0],r,feedrate[0]);
                        break;
                }
        }
                Group[2] = -1;
                Gcode[2] = -1;
                No_comp = G40;
                ctr = 1;
                start=0;
                G41 = 0;
                G42 = 0;
                G40 = 1;
                show++;
                refresh(show);
                if (Mcode[2] == 2)
                M02(0);

        } /* End of Gcode[2]==40 || Mcode[2] = 2 */
        show++;
        refresh(show);
        if(single==2)break;
        if(endn==line[1])single=2;
       } /* End Compensate On (No_comp == 0) */

/*************************************************************************/

        } /* End of fscanf */
        fclose(fp);
        window(3,3,78,11);
```

```
    clrscr();
} /* End compiler */

/*****************************************************************************/
```

G_CODE.C

```
#include<stdio.h>
#include<math.h>
#include<dos.h>
#include"mc1401.h"
#include"hrdwr.h"
#include"host_io.h"

#define pi 3.1415926
#define acc 1000000

void linear(int axis_1,int axis_2,double x1,double x2,long feedrate);
void G00(int axis_1,int axis_2,int axis_3,double x1,double x2,double x3);
void G01(int axis_1,int axis_2,int axis_3,double x1,double x2,double x3,long feedrate);
void G02(int axis_1,int axis_2,double pos1,double pos2,double radius,long feedrate);
void G03(int axis_1,int axis_2,double pos1,double pos2,double radius,long feedrate);
void G04(float dwell);
void G17(void);/* XY plane */
void G18(void);/* ZX plane */
void G19(void);/* YZ plane */
void G20(void);/* inch */
void G21(void);/* mm   */
void G81(double pos1,double pos2,double deep,double R_point,int k,long feedrate);
void G90(void);/* absolute */
void G91(void);/* increment */
void G92(int axis_1,int axis_2,int axis_3,double x1,double x2,double x3);
void change_axis(int axis);

void M02(void);/* End of program */
void M03(int speed);/* Spindle CW */
void M04(void);/* Spindle CCW */
void M05(void);/* Spindle Stop */
long mask(long number);
double X1,X2;
int Axis_1=0,Axis_2=1;
int mm=1,Absolute=1,CW=1;
long st,t,u;
/*main()*/
void g_code(char file[12])
{
```

```
  int i;
/*   clrscr();
  reset(0);
  update(0);*/
  for (i=0;i<=1;i++)
    { change_axis(i);
          axis_on(0);
          mtr_on(0);
          set_output_dac16(0);
          set_prfl_trap(0);
          set_acc(acc);
          set_fltr_pid(0);
          set_kp(10L);
          set_ki(10L);
          set_kd(10L);
     set_capt_home(0);
          synch_prfl(0);
          update(0);
  }
  for (i=2;i<=3;i++)
    { change_axis(i);
          axis_on(0);
          mtr_on(0);
          set_output_dac16(0);
          set_prfl_trap(0);
          set_acc(acc);
          set_fltr_pid(0);
          set_kp(15L);
          set_ki(5L);
          set_kd(10L);
    set_capt_home(0);
          synch_prfl(0);
          update(0);
  }
 /* home();
  manual(); */
  compiler(file);
```

```
/*    printf("READY TO GO\n");
    getch();
    G01(2,0,20,20,-20,200000);
    getch();
    G02(0,1,10,-10,-50,300000);
    G01(0,1,2,-10,-10,-20,100000);
    linear(2,0,0,0,100000);*/
}

long mask(long number)
{  long temp;
   temp = 0x000f&number;
   return(temp);
}

void linear(
int axis_1,
int axis_2,
double X1,
double X2,
long feedrate)
{  double    length,loc1,loc2,v1,v2;
   double    pos1,pos2;
   change_axis(axis_1);
   loc1 = get_pos(0);
   clr_status(0);
   update(0);
   change_axis(axis_2);
   loc2 = get_pos(0);
   clr_status(0);
   update(0);
   if( (axis_1==0)&&(axis_2==1) ){
   pos1 = (long)(400*X1);
   pos2 = -(long)(400*X2);}
   if( (axis_1==2)&&(axis_2==0) ){
   pos1 = -(long)(400*X1);
   pos2 = (long)(400*X2);}
   if( (axis_1==1)&&(axis_2==2) ){
   pos1 = -(long)(400*X1);
```

```
pos2 = -(long)(400*X2);}
if( ((pos1-loc1)!=0)||((pos2-loc2)!=0) )
length = (long)(sqrt(pow((pos1-loc1),2)+pow((pos2-loc2),2)));
else length = 1 ;
v1 = (long)(feedrate*(pos1-loc1)/length);
v2 = (long)(feedrate*(pos2-loc2)/length);
if ( v1 < 0 ) v1 = -v1;
if ( v2 < 0 ) v2 = -v2;

change_axis(axis_1);
set_pos(pos1);
set_vel(v1);
change_axis(axis_2);
set_pos(pos2);
set_vel(v2);
multi_update(0x7);
if( ((pos1-loc1)!=0)&&((pos2-loc2)!=0) ){
  do{
  change_axis(axis_1);
  st=get_status(0);
  change_axis(axis_2);
  t=get_status(0);
  st = mask(st);
  t = mask(t);
  online(1);
  online(2);
  }while(
(((st!=0x1)&&(st!=0x9))||((t!=0x1)&&(t!=0x9)))&&(((st!=0x1)&&(st!=0x9))||((t!=0x1)&&(t!=0x9)))&&(((st!
=0x1)&&(st!=0x9))||((t!=0x1)&&(t!=0x9))) );}
 else if((pos2-loc2)!=0){
        change_axis(axis_2);
        do{ t=get_status(0);
      t = mask(t);
    online(2);
        }while( (t!=1)&&(t!=0x9)&&(t!=0x1)&&(t!=0x9)&&(t!=0x1)&&(t!=0x9) );}
 else if((pos1-loc1)!=0){
        change_axis(axis_1);
        do{ st=get_status(0);
      st = mask(st);
```

```
        online(1);
          }while( (st!=0x1)&&(st!=0x9)&&(st!=0x1)&&(st!=0x9)&&(st!=0x1)&&(st!=0x9) );}
}

void   change_axis(int axis)
{ switch(axis)
  {
   case 0:
          set_1(0);
          break;
   case 1:
          set_2(0);
          break;
   case 2:
          set_3(0);
          break;
   case 3:
          set_4(0);
          break;
   default:
          break;
   }
}

void G00(
int axis_1,
int axis_2,
int axis_3,
double X1,
double X2,
double X3)

{ double   length,loc1,loc2,loc3,v1,v2,v3;
  double   pos1,pos2,pos3;
  long     feedrate;
  change_axis(axis_1);
  loc1 = get_pos(0);
  clr_status(0);
  update(0);
```

```
change_axis(axis_2);
loc2 = get_pos(0);
clr_status(0);
update(0);
change_axis(axis_3);
loc3 = get_pos(0);
clr_status(0);
update(0);
pos1 = (long)(400*X1);
pos2 = -(long)(400*X2);
pos3 = -(long)(400*X3);
if( ((pos1-loc1)!=0)||((pos2-loc2)!=0)||((pos3-loc3)!=0) )
length = (long)(sqrt(pow((pos1-loc1),2)+pow((pos2-loc2),2)+pow((pos3-loc3),2)));
else length = 1 ;
feedrate = 50000;
v1 = (long)(feedrate*(pos1-loc1)/length);
v2 = (long)(feedrate*(pos2-loc2)/length);
v3 = (long)(feedrate*(pos3-loc3)/length);
if ( v1 < 0 ) v1 = -v1;
if ( v2 < 0 ) v2 = -v2;
if ( v3 < 0 ) v3 = -v3;
change_axis(axis_1);
set_pos(pos1);
set_vel(v1);
change_axis(axis_2);
set_pos(pos2);
set_vel(v2);
change_axis(axis_3);
set_pos(pos3);
set_vel(v3);
multi_update(0x7);
if( ((pos1-loc1)!=0)&&((pos2-loc2)!=0)&&((pos3-loc3)!=0) ){
  do{
  change_axis(axis_1);
  st=get_status(0);
  change_axis(axis_2);
  t=get_status(0);
  change_axis(axis_3);
  u=get_status(0);
```

```
    st = mask(st);
    t = mask(t);
    u = mask(u);
    online(1);
    online(2);
    online(3);
    }while( ((st!=0x1)&&(st!=0x9))||((t!=0x1)&&(t!=0x9))||((u!=0x1)&&(u!=0x9)) );}
 else if( ((pos1-loc1)!=0)&&((pos2-loc2)!=0) ){
       do{
        change_axis(axis_1);
        st=get_status(0);
        change_axis(axis_2);
        t=get_status(0);
        st = mask(st);
        t = mask(t);
        online(1);
        online(2);
            }while( ((st!=0x1)&&(st!=0x9))||((t!=0x1)&&(t!=0x9)) );}
 else if( ((pos1-loc1)!=0)&&((pos3-loc3)!=0) ){
      do{
        change_axis(axis_1);
        st=get_status(0);
        change_axis(axis_3);
        u=get_status(0);
        st = mask(st);
        u = mask(u);
        online(1);
        online(3);
            }while( ((st!=0x1)&&(st!=0x9))||((u!=0x1)&&(t!=0x9)) );}
else if( ((pos2-loc2)!=0)&&((pos3-loc3)!=0) ){
     do{
       change_axis(axis_2);
       t=get_status(0);
       change_axis(axis_3);
       u=get_status(0);
       t = mask(t);
       u = mask(u);
       online(2);
       online(3);
```

```
            }while( ((t!=0x1)&&(t!=0x9))||((u!=0x1)&&(t!=0x9)) );}
    else if( (pos1-loc1)!=0 ){
        set_1(0);
        do{
          st=get_status(0);
          st = mask(st);
          online(1);
         }while( (st!=0x1)&&(st!=0x9) );
       }
    else if( (pos2-loc2)!=0 ){
        set_2(0);
        do{
          t=get_status(0);
          t = mask(t);
          online(2);
         }while( (t!=0x1)&&(t!=0x9) );
       }
    else if( (pos3-loc3)!=0 ){
        set_3(0);
        do{
          u=get_status(0);
          u= mask(u);
          online(3);
         }while( (u!=0x1)&&(u!=0x9) );
        /* while( (u!=0x6301)&&(u!=0x6309) );*/
       }
}

void G01(
int axis_1,
int axis_2,
int axis_3,
double X1,
double X2,
double X3,
long feedrate)
{ double   length,loc1,loc2,loc3,v1,v2,v3;
  double   pos1,pos2,pos3;
  change_axis(axis_1);
```

```
loc1 = get_pos(0);
clr_status(0);
update(0);
change_axis(axis_2);
loc2 = get_pos(0);
clr_status(0);
update(0);
change_axis(axis_3);
loc3 = get_pos(0);
clr_status(0);
update(0);
pos1 = (long)(400*X1);
pos2 = -(long)(400*X2);
pos3 = -(long)(400*X3);
if( ((pos1-loc1)!=0)||((pos2-loc2)!=0)||((pos3-loc3)!=0) )
length = (long)(sqrt(pow((pos1-loc1),2)+pow((pos2-loc2),2)+pow((pos3-loc3),2)));
else length = 1 ;
v1 = (long)(feedrate*(pos1-loc1)/length);
v2 = (long)(feedrate*(pos2-loc2)/length);
v3 = (long)(feedrate*(pos3-loc3)/length);
if ( v1 < 0 ) v1 = -v1;
if ( v2 < 0 ) v2 = -v2;
if ( v3 < 0 ) v3 = -v3;
change_axis(axis_1);
set_pos(pos1);
set_vel(v1);
change_axis(axis_2);
set_pos(pos2);
set_vel(v2);
change_axis(axis_3);
set_pos(pos3);
set_vel(v3);
multi_update(0x7);

if( ((pos1-loc1)!=0)&&((pos2-loc2)!=0)&&((pos3-loc3)!=0) ){
  do{
  change_axis(axis_1);
  st=get_status(0);
  change_axis(axis_2);
```

```
    t=get_status(0);
    change_axis(axis_3);
    u=get_status(0);
    st = mask(st);
    t = mask(t);
    u = mask(u);
    online(1);
    online(2);
    online(3);
    }while( ((st!=0x1)&&(st!=0x9))||((t!=0x1)&&(t!=0x9))||((u!=0x1)&&(u!=0x9)) );}
  else if( ((pos1-loc1)!=0)&&((pos2-loc2)!=0) ){
       do{
        change_axis(axis_1);
        st=get_status(0);
        change_axis(axis_2);
        t=get_status(0);
        st = mask(st);
        t = mask(t);
        online(1);
        online(2);
            }while( ((st!=0x1)&&(st!=0x9))||((t!=0x1)&&(t!=0x9)) );}
  else if( ((pos1-loc1)!=0)&&((pos3-loc3)!=0) ){
       do{
        change_axis(axis_1);
        st=get_status(0);
        change_axis(axis_3);
        u=get_status(0);
        st = mask(st);
        u = mask(u);
        online(1);
        online(3);
            }while( ((st!=0x1)&&(st!=0x9))||((u!=0x1)&&(t!=0x9)) );}
  else if( ((pos2-loc2)!=0)&&((pos3-loc3)!=0) ){
      do{
        change_axis(axis_2);
        t=get_status(0);
        change_axis(axis_3);
        u=get_status(0);
        t = mask(t);
```

```
        u = mask(u);
        online(2);
        online(3);
            }while( ((t!=0x1)&&(t!=0x9))||((u!=0x1)&&(t!=0x9)) );}
    else if( (pos1-loc1)!=0 ){
        set_1(0);
        do{
          st=get_status(0);
          st = mask(st);
          online(1);
        }while( (st!=0x1)&&(st!=0x9) );
      }
    else if( (pos2-loc2)!=0 ){
        set_2(0);
        do{
          t=get_status(0);
          t = mask(t);
          online(2);
        }while( (t!=0x1)&&(t!=0x9) );
      }
    else if( (pos3-loc3)!=0 ){
        set_3(0);
        do{
          u=get_status(0);
          u= mask(u);
          online(3);
        }while( (u!=0x1)&&(u!=0x9) );
        /* while( (u!=0x6301)&&(u!=0x6309) );*/
      }


/*  if( ((pos1-loc1)!=0)&&((pos2-loc2)!=0)&&((pos3-loc3)!=0) ){
    do{
    change_axis(axis_1);
    st=get_status(0);
    change_axis(axis_2);
    t=get_status(0);
    change_axis(axis_3);
    u=get_status(0);
```

```
    }while( ((st!=0x4301)&&(st!=0x4309))||((t!=0x5301)&&(t!=0x5309))||((u!=0x6301)&&(u!=0x6309)) );}
  else if( ((pos1-loc1)!=0)&&((pos2-loc2)!=0) ){
       do{
        change_axis(axis_1);
        st=get_status(0);
        change_axis(axis_2);
        t=get_status(0);
            }while( ((st!=0x4301)&&(st!=0x4309))||((t!=0x5301)&&(t!=0x5309)) );}
  else if( ((pos1-loc1)!=0)&&((pos3-loc3)!=0) ){
       do{
        change_axis(axis_1);
        st=get_status(0);
        change_axis(axis_3);
        u=get_status(0);
            }while( ((st!=0x4301)&&(st!=0x4309))||((u!=0x6301)&&(t!=0x6309)) );}
  else if( ((pos2-loc2)!=0)&&((pos3-loc3)!=0) ){
       do{
        change_axis(axis_2);
        t=get_status(0);
        change_axis(axis_3);
        u=get_status(0);
            }while( ((t!=0x5301)&&(st!=0x5309))||((u!=0x6301)&&(t!=0x6309)) );}
  else if( (pos1-loc1)!=0 ){
       set_1(0);
       do{
         st=get_status(0);
       }while( (st!=0x4301)&&(st!=0x4309) );
     }
  else if( (pos2-loc2)!=0 ){
       set_2(0);
       do{
         t=get_status(0);
       }while( (t!=0x5301)&&(t!=0x5309) );
     }
  else if( (pos3-loc3)!=0 ){
       set_3(0);
       do{
         u=get_status(0);
       }while( (u!=0x6301)&&(u!=0x6309) );
```

```
            }*/
}

void G02(
int   axis_1,
int   axis_2,
double pos1,
double pos2,
double radius,
long   feedrate)

{  double x1,y1,x2,y2,x3,y3,x4,y4,X,Y,tempX,tempY,a,b,r,n,l,p,feed;
   double start,step,arc,angle;
   long i,loop;
 if( (axis_1==0)&&(axis_2==1) )
 { change_axis(axis_1);
   x1 = (get_pos(0)/400);
   clr_status(0);
   change_axis(axis_2);
   y1 = -(get_pos(0)/400);
   clr_status(0);
   x2 = pos1;
   y2 = pos2;
 }
 if( (axis_1==2)&&(axis_2==0) )
 { change_axis(axis_1);
   x1 = (get_pos(0)/400);
   clr_status(0);
   change_axis(axis_2);
   y1 = (get_pos(0)/400);
   clr_status(0);
   x2 = pos1;
   y2 = pos2;
 }
 if( (axis_1==1)&&(axis_2==2) )
 { change_axis(axis_1);
   x1 = (get_pos(0)/400);
   clr_status(0);
   change_axis(axis_2);
```

```
 y1 = (get_pos(0)/400);
 clr_status(0);
 x2 = pos1;
 y2 = pos2;
}
 r  = radius;
 feed = feedrate;
 if( (y2-y1)==0 )
   { a = (x1+x2)/2;
     b = (y1+y2)/2;
     l = x2-x1;
     p = pow(r,2)-pow((l/2),2);
     x3= a;
     y3= b-sqrt(p);
     x4= a;
     y4= b+sqrt(p);}
 else{n = -((x2-x1)/(y2-y1));
 a = (x1+x2)/2;
 b = (y1+y2)/2;
 l = sqrt((pow((x2-x1),2)+pow((y2-y1),2)));
 p = pow(r,2)-pow((l/2),2);
 x3 = sqrt(p/(1+pow(n,2)));
 y3 = n*x3;
 x4 = -x3;
 y4 = n*x4;
 x3 = x3+a;
 y3 = y3+b;
 x4 = x4+a;
 y4 = y4+b;}
 if ( r > 0 ){
 if ( (((x2-x1)*(y3-y1))-((y2-y1)*(x3-x1))) < 0 ){
          X = x3;
          Y = y3;}
  else{X = x4;      /* find center finish*/
           Y = y4;};}
 else{
 if ( (((x2-x1)*(y3-y1))-((y2-y1)*(x3-x1))) >= 0 ){
          X = x3;
          Y = y3;}
```

```
  else{X = x4;     /* find center finish*/
        Y = y4;};
 }
 /*find start angle*/
 if( r > 0 )  start = asin((y1-Y)/r);
 else start = asin((y1-Y)/(-r));
 if(start < 0)start=-start;
 if( ((y1-Y) >= 0)&&((x1-X)>= 0) )
    start = start;
 if( ((y1-Y) > 0)&&((x1-X)< 0) )
    start = pi-start;
 if( ((y1-Y) <= 0)&&((x1-X)<= 0) )
    start = pi+start;
 if( ((y1-Y) < 0)&&((x1-X)> 0) )
    start = (2*pi)-start;

 if ( r > 0 ){  arc = 2*asin(l/2/r);
                    if( arc < 0 ) arc = -arc;}
 else { arc = 2*asin((-l)/2/r);
        if( arc < 0 ) arc = -arc;
        arc = 2*pi-arc;}
 /*find step angle*/
 loop = (long)(abs(50*r*arc));
 if( loop > 0 )  step = arc/loop;
 else step = 0;
 for(i=1;i<=loop;i++)
   { angle = start-(i*step);
        if ( r > 0 ){
        tempX = X + (r*cos(angle));
        tempY = Y + (r*sin(angle));}
        else{
        tempX = X - (r*cos(angle));
        tempY = Y - (r*sin(angle));}
        X1=tempX ;
        X2=tempY ;
        linear(axis_1,axis_2,X1,X2,feed);
  }
X1 = x2;
X2 = y2;
```

```
  linear(axis_1,axis_2,X1,X2,feed);
}

void G03(
int   axis_1,
int   axis_2,
double pos1,
double pos2,
double radius,
long   feedrate)

{  double x1,y1,x2,y2,x3,y3,x4,y4,X,Y,tempX,tempY,a,b,r,n,l,p,feed;
   double start,step,arc,angle;
   long i,loop;
   change_axis(axis_1);
   x1 = (get_pos(0)/400);
   clr_status(0);
   change_axis(axis_2);
   y1 = -(get_pos(0)/400);
   clr_status(0);
   x2 = pos1;
   y2 = pos2;
   r  = radius;
   feed = feedrate;
   if( (y2-y1)==0 )
     { a = (x1+x2)/2;
       b = (y1+y2)/2;
       l = x2-x1;
       p = pow(r,2)-pow((l/2),2);
       x3= a;
       y3= b-sqrt(p);
       x4= a;
       y4= b+sqrt(p);}
   else{n = -((x2-x1)/(y2-y1));
   a = (x1+x2)/2;
   b = (y1+y2)/2;
   l = sqrt((pow((x2-x1),2)+pow((y2-y1),2)));
   p = pow(r,2)-pow((l/2),2);
   x3 = sqrt(p/(1+pow(n,2)));
```

```
y3 = n*x3;
x4 = -x3;
y4 = n*x4;
x3 = x3+a;
y3 = y3+b;
x4 = x4+a;
y4 = y4+b;}
if ( r > 0 ){
if ( (((x2-x1)*(y3-y1))-((y2-y1)*(x3-x1))) >= 0 ){
        X = x3;
        Y = y3;}
 else{X = x4;     /* find center finish*/
        Y = y4;};}
else{
if ( (((x2-x1)*(y3-y1))-((y2-y1)*(x3-x1))) < 0 ){
        X = x3;
        Y = y3;}
 else{X = x4;     /* find center finish*/
        Y = y4;};
}
/*find start angle*/
if( r > 0 )  start = asin((y1-Y)/r);
else start = asin((y1-Y)/(-r));
if(start < 0)start=-start;
if( ((y1-Y) >= 0)&&((x1-X)>= 0) )
   start = start;
if( ((y1-Y) > 0)&&((x1-X)< 0) )
   start = pi-start;
if( ((y1-Y) <= 0)&&((x1-X)<= 0) )
   start = pi+start;
if( ((y1-Y) < 0)&&((x1-X)> 0) )
   start = (2*pi)-start;

if ( r > 0 ){  arc = 2*asin(l/2/r);
        if( arc < 0 ) arc = -arc;}
else {  arc = 2*asin((-l)/2/r);
     if( arc < 0 ) arc = -arc;
     arc = 2*pi-arc;}
/*find step angle*/
```

```
    loop = (long)(abs(50*r*arc));
    if( loop > 0 )  step = arc/loop;
    else step = 0;
    for(i=1;i<=loop;i++)
      { angle = start+(i*step);
            if ( r > 0 ){
            tempX = X + (r*cos(angle));
            tempY = Y + (r*sin(angle));}
            else{
            tempX = X - (r*cos(angle));
            tempY = Y - (r*sin(angle));}
            X1=tempX ;
            X2=tempY ;
            linear(axis_1,axis_2,X1,X2,feed);
      }
   X1 = x2;
   X2 = y2;
   linear(axis_1,axis_2,X1,X2,feed);
 }

 void G04(
 float dwell)
 {
   delay(dwell*1000);
 }

 void G17(void)
 {
   Axis_1 = 0;
   Axis_2 = 1;
 }

 void G18(void)
 {
   Axis_1 = 2;
   Axis_2 = 0;
 }

 void G19(void)
```

```
{
  Axis_1 = 1;
  Axis_2 = 2;
}

void G20(void)
{
  mm = 0;
}

void G21(void)
{
  mm = 1;
}

void G90(void)
{
  Absolute = 1;
}

void G91(void)
{
  Absolute = 0;
}

/****************** M-Code **********************/

void M02(void)
{
  int i;
  for (i=0;i<=2;i++)
     { change_axis(i);
          axis_on(0);
          mtr_on(0);
          set_output_dac16(0);
          set_prfl_trap(0);
          set_acc(acc);
     set_vel(0L);
     set_fltr_pid(0);
```

```
            set_kp(10L);
            set_ki(10L);
            set_kd(10L);
        set_capt_home(0);
            synch_prfl(0);
            update(0);
      }
   set_4(0);
   set_mtr_cmd(0);
   update(0);
}

void M03(int speed)
{  int spindle;
   CW = 1;
   set_4(0);
   mtr_off(0);
   spindle = (int)((float)speed*32767/1500);
   if (spindle < 0 ) spindle = -spindle;
   set_mtr_cmd(spindle);
}

void M04(void)
{
  CW = 0;
}

void M05(void)
{
      change_axis(3);
          axis_on(0);
          mtr_on(0);
          set_output_dac16(0);
          set_prfl_vel(0);
          set_acc(acc);
      set_vel(0L);
      set_fltr_pid(0);
          set_kp(10L);
          set_ki(10L);
```

```
        set_kd(10L);
    set_capt_home(0);
        synch_prfl(0);
        update(0);
}
```

HOME.C

```
#include <stdio.h>
#include <math.h>
#include <conio.h>
#include <dos.h>
#include "mc1401.h"
#include "hrdwr.h"
#include "host_io.h"

#define KBD_PORT 0x60

/*void main()*/
void home(void)
{
 long a,speed,trip,check;
 int i;
/*clrscr();
 reset(a);
 update(a);*/

 /*AXIS#1*/
 set_1(a);
 axis_on(a);
 mtr_on(a);
 set_output_dac16(a);
 set_prfl_vel(a);
 set_fltr_pid(a);
 set_kp(10L);
 set_ki(10L);
 set_kd(10L);
/* zero_pos(a);*/
 synch_prfl(a);
 set_lmt_sense(0L);
 lmts_on(a);
 update(a);

/*AXIS#2*/
set_2(a);
axis_on(a);
mtr_on(a);
```

```
  set_output_dac16(a);
  set_prfl_vel(a);
  set_fltr_pid(a);
  set_kp(10L);
  set_ki(10L);
  set_kd(10L);
 /*  zero_pos(a);*/
  synch_prfl(a);
  update(a);

  /*AXIS#3*/
  set_3(a);
  axis_on(a);
  mtr_on(a);
  set_output_dac16(a);
  set_prfl_vel(a);
  set_fltr_pid(a);
  set_kp(15L);
  set_ki(5L);
  set_kd(10L);
 /*  zero_pos(a);*/
  synch_prfl(a);
  update(a);

  /*HOME*/
  /*printf("Now,Homing\n");*/
  speed = 250000;
  /*getch();*/

  /*HOME AXIS_Z*/
        set_3(a);
             mtr_on(a);
        set_prfl_vel(a);
        set_acc(-1000000L);
        set_vel(speed);
       /* synch_prfl(a);*/
        set_lmt_sense(0L);
        lmts_on(a);
        set_auto_stop_on(a);
```

```
        set_capt_index(a);
        update(a);
        do{ online(3);
           trip = get_lmt_swtch(a);

}while(!((trip==32)||(trip==37)||(trip==38)||(trip==41)||(trip==42)||(trip==36)||(trip==40)||(trip==34)||(trip==33)
) );
        set_acc(1000000L);
        set_vel(50000L);
        update(a);
        clr_status(a);
        i=0;
        do{
           trip = get_capt(a);
           do{ check = get_capt(a);
           }while( trip == check );
           i = i + 1;
        }while( i != 2 );
        clr_prfl(a);
        synch_prfl(a);
        zero_pos(a);
        update(a);
        delay(2000);
        synch_prfl(a);
        zero_pos(a);
        update(a);
        online(3);

/*HOME AXIS_X*/
       set_1(a);
            mtr_on(a);
       set_prfl_vel(a);
       set_acc(-1000000L);
       set_vel(speed);
       /*synch_prfl(a);*/
       set_lmt_sense(0L);
       lmts_on(a);
       set_auto_stop_on(a);
       set_capt_index(a);
```

```
        update(a);
        do{ online(1);
            trip = get_lmt_swtch(a);
          }while(
!((trip==2)||(trip==6)||(trip==10)||(trip==22)||(trip==38)||(trip==26)||(trip==42)||(trip==18)||(trip==34)) );
        set_acc(1000000L);
        set_vel(50000L);
        update(a);
        clr_status(a);
        i=0;
        do{
           trip = get_capt(a);
           do{ check = get_capt(a);
           }while( trip == check );
           i = i + 1;
        }while( i != 2 );
        clr_prfl(a);
        synch_prfl(a);
        zero_pos(a);
        update(a);
        delay(2000);
        synch_prfl(a);
        zero_pos(a);
        update(a);
        online(1);

/*HOME AXIS_Y*/
      set_2(a);
           mtr_on(a);
      set_prfl_vel(a);
      set_acc(1000000L);
      set_vel(speed);
      /*synch_prfl(a);*/
      set_lmt_sense(0L);
      lmts_on(a);
      set_auto_stop_on(a);
      set_capt_index(a);
      update(a);
      do{ online(2);
```

```
            trip = get_lmt_swtch(a);

}while(!((trip==4)||(trip==5)||(trip==6)||(trip==21)||(trip==37)||(trip==22)||(trip==38)||(trip==20)||(trip==36)) );
        set_acc(-1000000L);
        set_vel(50000L);
        update(a);
        clr_status(a);
        i=0;
        do{
           trip = get_capt(a);
           do{ check = get_capt(a);
           }while( trip == check );
           i = i + 1;
        }while( i != 2 );
        clr_prfl(a);
        synch_prfl(a);
        zero_pos(a);
        update(a);
        delay(2000);
        synch_prfl(a);
        zero_pos(a);
        update(a);
        online(2);

}
```

MANUAL.C

```
#include <stdio.h>
#include <math.h>
#include <conio.h>
#include <dos.h>
#include "mc1401.h"
#include "hrdwr.h"
#include "host_io.h"

#define KBD_PORT 0x60
void online(int sel);
volatile long count;
long keystroke,a,c,speed,pos,trip,check;

/*void main()*/

void manual(long vel)
{
/*  long per;
 clrscr();
 reset(a);
 update(a);*/

 /*AXIS#1*/
 set_1(a);
 axis_on(a);
 mtr_on(a);
 set_output_dac16(a);
 set_prfl_vel(a);
 set_fltr_pid(a);
 set_kp(10L);
 set_ki(10L);
 set_kd(10L);
/* zero_pos(a);*/
 synch_prfl(a);
 set_lmt_sense(0L);
 lmts_on(a);
 update(a);

 /*AXIS#2*/
```

```
  set_2(a);
  axis_on(a);
  mtr_on(a);
  set_output_dac16(a);
  set_prfl_vel(a);
  set_fltr_pid(a);
  set_kp(10L);
  set_ki(10L);
  set_kd(10L);
 /*  zero_pos(a);*/
  synch_prfl(a);
  update(a);

  /*AXIS#3*/
  set_3(a);
  axis_on(a);
  mtr_on(a);
  set_output_dac16(a);
  set_prfl_vel(a);
  set_fltr_pid(a);
  set_kp(10L);
  set_ki(10L);
  set_kd(10L);
 /*  zero_pos(a);*/
  synch_prfl(a);
  update(a);

  /*MANUAL CONTROL*/
 /* clrscr();
  printf("set max speed(NOT MORE THAN 500000) = ");scanf("%ld",&vel);*/
  speed = vel;
  keystroke = inportb(KBD_PORT);
  do{

    keystroke = inportb(KBD_PORT);
    switch( keystroke ){
            case 0x3b:

        set_1(a);
```

```
        mtr_on(a);
  set_prfl_vel(a);
  set_acc(1000000L);
  synch_prfl(a);
  set_lmt_sense(0L);
  lmts_on(a);
  set_auto_stop_on(a);
  update(a);
  trip = get_lmt_swtch(a);
  if( (trip==2)||(trip==6)||(trip==10)||(trip==22)||(trip==38)||(trip==26)||(trip==42)||(trip==18)||(trip==34) )
  {
    set_vel(speed);
    update(a);
    clr_status(a);
    if( speed != 0 )
    delay(200000000/speed);
  }
        do{
    online(1);
    keystroke = inportb(KBD_PORT);
    set_vel(speed);
    update(a);
        }while( keystroke != 0xbb );
 set_vel(0L);
 update(a);
       break;

      case 0x3c:

 set_1(a);
       mtr_on(a);
 set_prfl_vel(a);
 set_acc(-1000000L);
 synch_prfl(a);
 set_lmt_sense(0L);
 lmts_on(a);
 set_auto_stop_on(a);
 update(a);
 trip = get_lmt_swtch(a);
```

```
 if( (trip==1)||(trip==5)||(trip==9)||(trip==21)||(trip==37)||(trip==25)||(trip==41)||(trip==17)||(trip==33) )
 {
   set_vel(speed);
   update(a);
   clr_status(a);
   if( speed != 0 )
   delay(200000000/speed);
 }
 do{
    online(1);
    keystroke = inportb(KBD_PORT);
    set_vel(speed);
    update(a);
  }while( keystroke != 0xbc );
 set_vel(0L);
 update(a);
      break;

case 0x3d:

 set_2(a);
      mtr_on(a);
 set_prfl_vel(a);
 set_acc(-1000000L);
 synch_prfl(a);
 set_lmt_sense(0L);
 lmts_on(a);
 set_auto_stop_on(a);
 update(a);
 trip = get_lmt_swtch(a);
 if( (trip==4)||(trip==5)||(trip==6)||(trip==21)||(trip==37)||(trip==22)||(trip==38)||(trip==20)||(trip==36) )
 {
  set_vel(speed);
  update(a);
  clr_status(a);
  if( speed != 0 )
  delay(200000000/speed);
}
do{ online(2);
```

```
        keystroke = inportb(KBD_PORT);
        set_vel(speed);
        update(a);
          }while( keystroke != 0xbd );
    set_vel(0L);
    update(a);
         break;

   case 0x3e:

    set_2(a);
         mtr_on(a);
    set_prfl_vel(a);
    set_acc(1000000L);
    synch_prfl(a);
    set_lmt_sense(0L);
    lmts_on(a);
    set_auto_stop_on(a);
    update(a);
    trip = get_lmt_swtch(a);
    if( (trip==8)||(trip==9)||(trip==10)||(trip==25)||(trip==41)||(trip==26)||(trip==42)||(trip==24)||(trip==40) )
    {
      set_vel(speed);
      update(a);
      clr_status(a);
      if( speed != 0 )
      delay(200000000/speed);
   }
        do{ online(2);
      keystroke = inportb(KBD_PORT);
      set_vel(speed);
      update(a);
          }while( keystroke != 0xbe );
   set_vel(0L);
   update(a);
        break;

  case 0x40:
```

```
        set_3(a);
            mtr_on(a);
        set_prfl_vel(a);
        set_acc(1000000L);
        synch_prfl(a);
        set_lmt_sense(0L);
        lmts_on(a);
        set_auto_stop_on(a);
        update(a);
        trip = get_lmt_swtch(a);
        if(
(trip==32)||(trip==37)||(trip==38)||(trip==41)||(trip==42)||(trip==36)||(trip==40)||(trip==34)||(trip==33) )
        {
          set_vel(speed);
          update(a);
          clr_status(a);
          if( speed != 0 )
          delay(200000000/speed);
        }
            do{ online(3);
          keystroke = inportb(KBD_PORT);
          set_vel(speed);
          update(a);
            }while( keystroke != 0xc0 );
       set_vel(0L);
       update(a);
            break;

          case 0x3f:

      set_3(a);
            mtr_on(a);
      set_prfl_vel(a);
      set_acc(-1000000L);
      synch_prfl(a);
      set_lmt_sense(0L);
      lmts_on(a);
      set_auto_stop_on(a);
      update(a);
```

```
            trip = get_lmt_swtch(a);
            if(
 (trip==16)||(trip==20)||(trip==24)||(trip==21)||(trip==22)||(trip==25)||(trip==26)||(trip==18)||(trip==17) )
            {
              set_vel(speed);
              update(a);
              clr_status(a);
              if( speed != 0 )
              delay(200000000/speed);
            }
            do{ online(3);
               keystroke = inportb(KBD_PORT);
               set_vel(speed);
               update(a);
                  }while( keystroke != 0xbf );
            set_vel(0L);
            update(a);
                 break;

          default:break;}
     clr_prfl(a);
     update(a);
     synch_prfl(a);
     update(a);
     mtr_on(a);
     update(a);
     keystroke = inportb(KBD_PORT);
     count = getch();
   }while ( count != 0x1b );
}

void online(int sel)
{
   float sent;
   long get,initx,inity,initz;
   if ( sel == 0 )
   {
    set_1(0);
    initx = get_actl_pos(0);
```

```
        set_2(0);
        inity = get_actl_pos(0);
        set_3(0);
        initz = get_actl_pos(0);
      }
      window(3,14,37,24);
      switch(sel){
       case 1:set_1(0);
            get = get_actl_pos(0);
            sent  = (float)get/400;
            gotoxy(10,3);
            printf("%8.3f",sent);
            gotoxy(10,8);
            printf("%8.3f",sent);
            break;
       case 2:set_2(0);
            get = get_actl_pos(0);
            sent  = (float)get/400;
            if( sent != 0 ) sent = -sent;
            gotoxy(10,4);
            printf("%8.3f",sent);
            gotoxy(10,9);
            printf("%8.3f",sent);
            break;
      case 3:set_3(0);
            get = get_actl_pos(0);
            sent  = (float)get/400;
            if( sent != 0 )sent = -sent;
            gotoxy(10,5);
            printf("%8.3f",sent);
            gotoxy(10,10);
            printf("%8.3f",sent);
            break;
      default:break;}
      sel = 100;

   }
```

MONITOR.C

```
#include<conio.h>
#include<stdio.h>
#include<dos.h>
#include<math.h>
#include<ctype.h>
#include<graphics.h>

#define MAXMAINMENU  5
#define high       18  /*define color*/
#define norm       113
#define color1     07
#define color2     112
#define XMAX       319
#define YMAX       199
#define BLUE       1
#define GREEN      2
#define RED        4
#define INTENSE    8
#define BLUE_BACK  16
#define GREEN_BACK 32
#define RED_BACK   64
#define BLINK      128
#define KBD_PORT   0x60
#define UP   0
#define DOWN 1

char *menu[MAXMAINMENU]={" File  ",
                         " Memory  ",
                         " Jog  ",
                         " Cycle  ",
                         " Setup  "}; /*title menu bar*/

typedef struct{
                int l,t,r,b,max;
                char *text,*save;
              }PULLDOWN;  /*STRUCTURE OF PULLDOWN MENU*/
long   sendjog;
float  feed,jog;
float  ax,ay,az,rx,ry,rz;
```

```
long   gx,gy,gz;
int    prop,i,d;
int    startn=1,endn=999,startx[30],starty[30],endx[30],endy[30];
int    single=0,start=0,block=0;
int    con1=1,con2=1,con3=1;
/*int    tog=1,quit,ch;*/
int    tog=1;
int    quit,ch;
int    ball,pulse,spindle;
int    file,testget,press;
char   file_load[12];
FILE   *lf;
cursoroff()
{
        union REGS reg;

        reg.h.ah=1;
        reg.x.cx=0x2000;
        int86(0x10,&reg,&reg);
}

cursoron()
{
        union REGS reg;

        reg.h.ah=1;
        reg.h.ch=6;
        reg.h.cl =7;
        int86(0x10,&reg,&reg);
}

/*-------------------------------------------------------------------------
   FUNCTION  BORDER(*PULLDOWN)
    - create the box of menu that pass into function
        input : pointer of PULLDOWN
        output: draw box on screen
-------------------------------------------------------------------------*/

void Border(PULLDOWN *p)
```

```
 {
        register i;

        textattr(norm);
        for(i=p->l;i<p->r;i++)
        {
         gotoxy(i,p->t);
         cprintf("%c",196);
         gotoxy(i,p->b);
         cprintf("%c",196);
        }
        for(i=p->t;i<p->b;i++)
        {
         gotoxy(p->l,i);
         cprintf("%c",179);
         gotoxy(p->r,i);
         cprintf("%c",179);
        }
         gotoxy(p->l,p->t);cprintf("%c",218);
         gotoxy(p->r,p->t);cprintf("%c",191);
         gotoxy(p->l,p->b);cprintf("%c",192);
         gotoxy(p->r,p->b);cprintf("%c",217);
}

/*---------------------------------------------------------------------------
    Function Pull(*PULLDOWN)
    - save old screen
    - write text into box
        input  : pointer of PULLDOWN
        output : box with text
---------------------------------------------------------------------------*/

void Pull(PULLDOWN *p)
{
        int buffer;

        buffer=(p->r-p->l+4)*
            (p->b-p->t+4)*2;
        p->save=(char *)malloc(buffer);
```

```
        gettext(p->l,p->t,p->r+2,p->b+2,p->save);
        Border(p);
        window(p->l+1,p->t+1,p->r-1,p->b-1);
        textattr(norm);
        clrscr();
        if(p->text != NULL) cputs(p->text);
}

/*---------------------------------------------------------------------------
        Function Restor(*PULLDOWN)
        - restore old screen from Function Pull
          input : pointer of PULLDOWN
          output: old screen
---------------------------------------------------------------------------*/

void Restor(PULLDOWN *p)
{
        puttext(p->l,p->t,p->r+2,p->b+2,p->save);
        free(p->save);
        p->save=NULL;
}

/*---------------------------------------------------------------------------
        Function Rewrite(*PULLDOWN,int,char)
        - write text with attribute
          input : 1.pointer of PULLDOWN
                  2.row
                  3.attribute (63=highlight,127=normal)
          output: text with attribute row
---------------------------------------------------------------------------*/

void Rewrite(PULLDOWN *p,int row,char attri)
{
        int c,chars;
        union REGS reg;

        chars=p->r-p->l;
        for(c=1;c<=chars;c++)
        {
```

```
            gotoxy(c,row);
            reg.h.ah=8;
            reg.h.bh=0;
            int86(0x10,&reg,&reg);
            reg.h.ah=9;
            reg.h.bl=attri;
            reg.h.bh=0;
            reg.x.cx=1;
            int86(0x10,&reg,&reg);
           }
}

/*---------------------------------------------------------------------------
          Function Menubar()
          - write main menu bar at the top of screen
           input : none
           output: main menu
---------------------------------------------------------------------------*/

void Menubar()
{
        int i,s=4;

        textattr(color1);cprintf("%80c",32);
        for(i=0;i<MAXMAINMENU;i++)
        {
          gotoxy(s,1);
          s=s+strlen(menu[i])+4;
          cprintf(menu[i]);
        }
}

/*---------------------------------------------------------------------------
        Function Selmenubar(int)
        - write highlight menubar
        input : order of menubar
        output: highlight menubar
---------------------------------------------------------------------------*/
```

```
void Selmenubar(int s)
{
        int i,m=4;

        for(i=0;i!=s-1;i++)
        m=m+strlen(menu[i])+4;
        gotoxy(m,1);
        textattr(high);
        cprintf(menu[i]);
        textattr(norm);
}

/*-------------------------------------------------------------------------
        Function Normalmenubar(int)
        - write normal menubar
        input : order of menubar
        output: normal attribute of menubar
-------------------------------------------------------------------------*/

void Normalmenubar(int s)
{
        int i,m=4;

        for(i=0;i!=s-1;i++)
        m=m+strlen(menu[i])+4;
        gotoxy(m,1);
        textattr(color1);
        cprintf(menu[i]);
        textattr(norm);
}

/*-------------------------------------------------------------------------
        Function (int)Selection(*PULLDOWN,int *)
        - draw pull down menu and wait for keyboard input
        - process about all pull down menu
        input : 1.pointer of PULLDOWN
                2.pointer of int
        output: 1.return choice of menu bar
                2.pass back by value of int*  that is choice of pull down menu
```

```
----------------------------------------------------------------------*/

int Selection(PULLDOWN *p,int *down,int *vtab)
{
        int se,mainmenu=*down,old=mainmenu,row,oldrow;

        row=oldrow=*vtab;
        window(1,1,80,25);
        Selmenubar(*down);
        Pull(p);
        do
        {
          Rewrite(p,row,high);
          se=getch();
          if(se==0)
          {
            se=getch();
            switch(se)
            { case 72 :  row--;
                         if(row<1) row=p->max;
                         break;
              case 80 :  row++;
                         if(row>p->max) row=1;
                         break;
              case 75 :  mainmenu--;
                         if(mainmenu<1) mainmenu=MAXMAINMENU;
                         break;
              case 77 :  mainmenu++;
                         if(mainmenu>MAXMAINMENU) mainmenu=1;
                         break;
              default :  break;
            }
            Rewrite(p,oldrow,norm);
          }
          oldrow=row;
        }while(se != 13 && old==mainmenu);
        window(1,1,80,25);
        Restor(p);
        *vtab=row;
```

```
            if (se==13)
            {
              *down=row;
              return(mainmenu);
            }
            else
            {
              *down=mainmenu;
              Normalmenubar(old);
              return(0);
            }
}

void About()
{
            PULLDOWN *p;

            p=(PULLDOWN *)malloc(sizeof(PULLDOWN));
            p->l=20;
            p->t=8;
            p->r=59;
            p->b=12;
            p->text=" CNC VERTICAL MILLING MACHINE VER.2.0\r\n            By\r\n  MECHANICAL
ENGINEERING OF KMIT'L";
            p->save=NULL;
            Pull(p);
            getch();
            Restor(p);
}

float get_option() /*read characters and return value of characters*/
{
           char s[20],c;
           int i;

           for(i=0; (c=getchar()) != '\n';i=i+1) s[i] = c;
           s[i] = '\0';
           if( i>=1) testget = 1;
           else testget=0;
```

```
        return(atof(s));
}

void cycle_start()
{
        if( start == 1)
        {
          window(45,15,57,16);
          textattr(norm);
          clrscr();
          textcolor(LIGHTGREEN|BLINK);
          textbackground(BLUE);
          M03(0);
          gotoxy(1,1);cprintf(" Cycle Start ");
          window(45,17,57,18);
          textattr(norm);
          clrscr();
          textcolor(RED);
          textbackground(BLUE);
          gotoxy(1,1);cprintf(" Cycle Stop ");
          textattr(norm);
          start = 0;
          con1=1;
          con2=1;
          con3=0;
          g_code(file_load);
        }
}

void cycle_stop()
{int j;
        window(45,17,57,18);
        textattr(norm);
        clrscr();
        textcolor(LIGHTGREEN);
        textbackground(BLUE);
        M03(0);
        gotoxy(1,1);cprintf(" Cycle Stop ");
        window(45,15,57,16);
```

```
            textattr(norm);
            clrscr();
            textcolor(RED);
            textbackground(BLUE);
            gotoxy(1,1);cprintf(" Cycle Start ");
            window(45,19,59,20);
            textattr(norm);
            clrscr();
            textcolor(RED);
            textbackground(BLUE);
            gotoxy(1,1);cprintf(" Spindle ON ");
            textattr(norm);
            single = 0;
            start = 1;
            block = 1;
            con1=1;
            con2=1;
            con3=1;
            for(j=0;j<=8;j++)file_load[j] = 0;
            window(45,21,78,22);
            textattr(norm);
            clrscr();
            textcolor(LIGHTGREEN);
            textbackground(BLUE);
            gotoxy(1,1);
            if(tog==0)cprintf(" Feed Rate   :   0.000 mm/min");
            else cprintf(" Feed Rate   :   0.000 in/min");
            gotoxy(1,2);cprintf(" Spindle Rate :   0    rpm   ");
            textattr(norm);
            cursoroff();
}

void spindel_on()
{
          if( block == 1 )
          {
            window(45,19,59,20);
            textattr(norm);
            clrscr();
```

```
                textcolor(LIGHTGREEN|BLINK);
                textbackground(BLUE);
                M03(spindle);
                gotoxy(1,1);cprintf(" Spindle ON  ");
                window(45,17,57,18);
                textattr(norm);
                clrscr();
                textcolor(RED);
                textbackground(BLUE);
                gotoxy(1,1);cprintf(" Cycle Stop ");
                textattr(norm);
                block = 0;
                con1=0;
                con2=1;
                con3=1;
              }
}

void real()  /*display the value of position that return
                from controller card on the screen*/
{
        window(3,14,37,24);
             textcolor(YELLOW);
             textbackground(WHITE);
        set_brk_off(0);
        update(0);
             set_1(0);
             gx= get_actl_pos(0);
             ax= (float)gx/400;
       set_2(0);
             gy= get_actl_pos(0);
             ay= (float)gy/400;
             set_3(0);
       gz= get_actl_pos(0);
             az= (float)gz/400;
       set_1(0);
             gotoxy(10,3);cprintf("%8.3f",ax);
             gotoxy(10,4);cprintf("%8.3f",ay);
             gotoxy(10,5);cprintf("%8.3f",az);
```

```
        gotoxy(10,8);cprintf("%8.3f",rx);
        gotoxy(10,9);cprintf("%8.3f",ry);
        gotoxy(10,10);cprintf("%8.3f",rz);
        textattr(norm);

}

void show()  /*display the value of variables in memory menu
              in form of in. or mm. on the screen*/

{
        window(45,21,78,21);
        textattr(norm);
        clrscr();
        textcolor(LIGHTGREEN);
        textbackground(BLUE);
        gotoxy(1,1);
        if( tog == 1 ) cprintf(" Feed Rate    : %8.3f mm/min",feed);
        else cprintf(" Feed Rate    : %8.3f in/min",feed);
        textattr(norm);
        cursoroff();
        window(45,22,78,22);
        textattr(norm);
        clrscr();
        textcolor(LIGHTGREEN);
        textbackground(BLUE);
        gotoxy(1,1);cprintf(" Spindle Rate : %4d    rpm   ",spindle);
        textattr(norm);
        cursoroff();
        window(45,23,78,23);
        textattr(norm);
        clrscr();
        textcolor(LIGHTGREEN);
        textbackground(BLUE);
        gotoxy(1,1);
        if( tog == 1 ) cprintf(" Jog Speed    : %8.3f mm/min",jog);
        else cprintf(" Jog Speed    : %8.3f in/min",jog);
        textattr(norm);
        cursoroff();
```

```
}

void display() /* print pattern of position displaying */
{
        window(3,14,37,24);
        textattr(norm);
        clrscr();
        textcolor(YELLOW);
        textbackground(WHITE);
        gotoxy(1,2);cprintf("      ACTUAL POSITION (ABSOLUTE) ");
        gotoxy(1,3);cprintf("    X : ");
        gotoxy(1,4);cprintf("    Y : ");
        gotoxy(1,5);cprintf("    Z : ");
        gotoxy(1,7);cprintf("      ACTUAL POSITION (RELATIVE) ");
        gotoxy(1,8);cprintf("    X : ");
        gotoxy(1,9);cprintf("    Y : ");
        gotoxy(1,10);cprintf("    Z : ");
        real();
        window(28,24,37,24);
        textcolor(YELLOW);
        textbackground(GREEN);
        if( tog == 1 )
        {
          gotoxy(1,1);
          cprintf(" mm./min ");
        }
        else
        {
          gotoxy(1,1);
          cprintf(" in./min ");
        }
        textattr(norm);
}

void toggle()  /*toggle between in. and mm.*/
{
        window(28,24,37,24);
        textcolor(YELLOW);
        textbackground(GREEN);
```

```
            if( tog == 1 )
            {
              gotoxy(1,1);cprintf(" mm./min ");
              feed = feed*25.4;
              jog = jog*25.4;
              ax = ax*25.4;
              ay = ay*25.4;
              az = az*25.4;
              rx = rx*25.4;
              ry = ry*25.4;
              rz = rz*25.4;
              show();
              display();
              tog=0;
            }
            else
            {
              gotoxy(1,1);
              cprintf(" in./min ");
              feed = feed/25.4;
              jog = jog/25.4;
              ax = ax/25.4;
              ay = ay/25.4;
              az = az/25.4;
              rx = rx/25.4;
              ry = ry/25.4;
              rz = rz/25.4;
              show();
              display();
              tog=1;
            }
}



void initial() /* initializing the parameters of the program */
{
          prop = 10;
          i = 10;
```

```
            d = 10;
            ball = 5;
            pulse = 500;
            feed = 0;
            spindle = 0;
            jog = 0;
}

void Pull_for_file(PULLDOWN *p)
{
            int buffer;
            float get_option(),test;

            buffer=(p->r-p->l+4)*
                 (p->b-p->t+4)*2;
            p->save=(char *)malloc(buffer);
            gettext(p->l,p->t,p->r+2,p->b+2,p->save);
            Border(p);
            window(p->l+1,p->t+1,p->r-1,p->b-1);
            textattr(norm);
            clrscr();
            cursoron();
            gotoxy(1,1);cprintf("  Do you really want to exit?(Y/N): ");
            test = getch();
            if( (test==0x59)||(test==0x79) ) quit = 4;
            else quit = 5;
}

void Pull_for_memory(PULLDOWN  *p)
{
      int buffer,Num;
            float get_option(),test;

            buffer=(p->r-p->l+4)*
                 (p->b-p->t+4)*2;
           p->save=(char *)malloc(buffer);
           gettext(p->l,p->t,p->r+2,p->b+2,p->save);
           Border(p);
           window(p->l+1,p->t+1,p->r-1,p->b-1);
```

```
textattr(norm);
clrscr();
Num = p->t - 4;
if(Num==0)
{
  cursoroff();
  clrscr();gotoxy(1,1);cprintf("Feed Rate = %8.3f",feed);
  getch();
  cursoron();
  if( tog == 0 )
  {
    do
    {
          clrscr();
          gotoxy(1,1);cprintf("Feed Rate = ");
          test=get_option();
          if( (testget==1)&&( test>=0)&&( test<=1200) ) feed=test;
          else
          {
            if( testget == 1)
            {
              clrscr();
              gotoxy(1,1);
              cprintf(" Value out of limit!");
              getch();
            }
          }
   }while( !(( test>=0)&&( test<=1200)) );
}
else
{
  do
  {
          clrscr();
          gotoxy(1,1);cprintf("Feed Rate = ");
          test=get_option();
          if( (testget==1)&&( test>=0)&&( test<=47.244) ) feed=test;
          else
          {
```

```
                if( testget == 1 )
                {
                  clrscr();
                  gotoxy(1,1);
                  cprintf(" Value out of limit!");
                  getch();
                }
              }
    }while( !(( test>=0)&&( test<=47.244)) );
  }
 window(45,21,78,21);
 textattr(norm);
 clrscr();
 textcolor(LIGHTGREEN);
 textbackground(BLUE);
 gotoxy(1,1);
 if( tog== 0 )cprintf(" Feed Rate    : %8.3f mm/min",feed);
 else cprintf(" Feed Rate    : %8.3f in/min",feed);
 textattr(norm);
 cursoroff();
 }
 if(Num==1)
 {
  cursoroff();
  clrscr();
  gotoxy(1,1);cprintf("Spindle Rate = %4d",spindle);
  getch();
  cursoron();
  do
  {
    clrscr();
    gotoxy(1,1);cprintf("Spindle Rate = ");
    test=get_option();
    if( (testget==1)&&( test>=0)&&( test<=1500) )spindle=test;
    else
    {
          if(testget==1)
          {
            clrscr();
```

```
                gotoxy(1,1);
                cprintf(" Value out of limit!");
                getch();
              }
        }
    }while( !(( test>=0)&&( test<=1500)) );
    window(45,22,78,22);
    textattr(norm);
    clrscr();
    textcolor(LIGHTGREEN);
    textbackground(BLUE);
    gotoxy(1,1);cprintf(" Spindle Rate : %4d     rpm   ",spindle);
    textattr(norm);
    cursoroff();
}
if(Num==2)
{
  cursoroff();
  clrscr();
  gotoxy(1,1);cprintf("Jog Speed = %8.3f",jog);
  getch();
  cursoron();
  if( tog == 0 )
  {
    do
    {
          clrscr();
          gotoxy(1,1);cprintf("Jog Speed = ");
          test=get_option();
          if( (testget==1)&&( test>=0)&&( test<=1250) )jog=test;
          else
          {
            if( testget==1 )
            {
              clrscr();
              gotoxy(1,1);
              cprintf(" Value out of limit!");
              getch();
            }
```

```
                }
            }while( !(( test>=0)&&( test<=1250)) );
        }
        else
        {
          do
          {
                clrscr();
                gotoxy(1,1);cprintf("Jog Speed = ");
                test=get_option();
                if( (testget==1)&&( test>=0)&&( test<=50) )jog=test;
                else
                {
                  if( testget==1)
                  {
                    clrscr();
                    gotoxy(1,1);
                    cprintf(" Value out of limit!");
                    getch();
                  }
                }
            }while( !(( test>=0)&&( test<=50)) );
        }
        window(45,23,78,23);
        textattr(norm);
        clrscr();
        textcolor(LIGHTGREEN);
        textbackground(BLUE);
        gotoxy(1,1);
        if ( tog == 0 )cprintf(" Jog Speed    : %8.3f mm/min",jog);
        else cprintf(" Jog Speed    : %8.3f in/min",jog);
        textattr(norm);
        cursoroff();
      }
}

void Pull_for_jog(PULLDOWN *p)
{
      int buffer,Num;
```

```
float get_option();

buffer=(p->r-p->l+4)*
     (p->b-p->t+4)*2;
p->save=(char *)malloc(buffer);
gettext(p->l,p->t,p->r+2,p->b+2,p->save);
Border(p);
window(p->l+1,p->t+1,p->r-1,p->b-1);
textattr(norm);
clrscr();
Num = p->t - 4;
if(Num==0 || Num==1 || Num==2 || Num==3 || Num==4)
{
 if(Num==0)
 {
  cprintf(" Press F1 => X+, F2 => X-\r\n");
  cprintf("      F3 => Y+, F4 => Y-\r\n");
  cprintf("      F5 => Z+, F6 => Z- ");
if( tog == 0 )sendjog = (long)((jog*100000)/558);
   else sendjog = (long)((jog*25.4*100000)/558);
   manual(sendjog);
/*while( getch()!= 0x1b )*/;
 }
 if(Num==1)
 {
  textcolor(GREEN|BLINK);
  textbackground(BLUE);
  gotoxy(7,2);cprintf(" Press <H> to HOME ");
  getch();
  press = inportb(KBD_PORT);
 if( (press == 0x23) ){
          textattr(norm);
          gotoxy(7,2);clreol();
          textcolor(GREEN|BLINK);
      textbackground(BLUE);
          gotoxy(11,2);cprintf(" HOMING...");
 textattr(norm);
 home();}
  /*while( press != 0x1b );*/
```

```
                }
                if(Num==2)
                {
                  textcolor(GREEN|BLINK);
                  textbackground(BLUE);
                  gotoxy(6,2);cprintf(" NOW,SET ZERO POINT ");
                  set_1(0);
                  zero_pos(0);
                  update(0);
              set_2(0);
              zero_pos(0);
              update(0);
                  set_3(0);
                  zero_pos(0);
                  update(0);
                  textattr(norm);
                  online(1);
             online(2);
             online(3);
                  while( getch()!= 0x1b );
                }
                if(Num==3)
                {
                  textcolor(RED|BLINK);
                  textbackground(GREEN);
                  gotoxy(7,2);cprintf(" EMERGENCY STOP! ");
            M03(0);
                  textattr(norm);
                  while( getch()!= 0x1b );
                }
            }
            cursoroff();
}
void Pull_for_cycle(PULLDOWN *p)
{
            int buffer,Num;
            float get_option(),test;

            buffer=(p->r-p->l+4)*
```

```
            (p->b-p->t+4)*2;
p->save=(char *)malloc(buffer);
gettext(p->l,p->t,p->r+2,p->b+2,p->save);
Border(p);
window(p->l+1,p->t+1,p->r-1,p->b-1);
textattr(norm);
clrscr();
Num = p->t - 4;
if(Num == 3)
{
  cursoroff();
  do
  {
    clrscr();
    gotoxy(1,1);clreol();
    cprintf("Strat N = %3d",startn);
    getch();
    cursoron();
    do
    {
          gotoxy(1,1);
          clreol();
          cprintf("Strat N = ");
          test = (int)get_option();
          if( (test > 0)&&(test < 1000)) startn=test;
          else
          {
            gotoxy(1,1);
            if ( test == 0 )break;
            cprintf("  Value out of limit!");
            getch();
          }
  }while( !((test > 0)&&(test<1000)) );
  gotoxy(1,1);clreol();
  cprintf("Strat N = %3d",startn);
  cursoroff();
  gotoxy(1,2);clreol();
  cprintf("Finish N = %3d",endn);
  getch();
```

```
                cursoron();
                do
                {
                        gotoxy(1,2);
                        clreol();
                        cprintf("Finish N = ");
                        test = (int)get_option();
                        if( (test > 0)&&(test<1000)) endn=test;
                        else
                        {
                          gotoxy(1,2);
                          if ( test == 0 )break;
                          cprintf("  Value out of limit!");
                          getch();
                        }
                 }while( !((test > 0)&&(test<1000)) );
                 gotoxy(1,2);clreol();
                 cprintf("Finish N = %3d",endn);
                gotoxy(1,3);cprintf(" Are you sure?(Y/N) : ");
                 test = getch();
               }while( (test!=0x59)&&(test!=0x79) );
              }

}

void Pull_for_setup(PULLDOWN *p)
{
          int buffer,Num;
          float get_option(),test;

          buffer=(p->r-p->l+4)*
               (p->b-p->t+4)*2;
         p->save=(char *)malloc(buffer);
         gettext(p->l,p->t,p->r+2,p->b+2,p->save);
         Border(p);
         window(p->l+1,p->t+1,p->r-1,p->b-1);
         textattr(norm);
         clrscr();
         Num = p->t - 4;
```

```
 if(Num == 0)
 {
   textcolor(GREEN|BLINK);
   textbackground(BLUE);
   gotoxy(2,1);
   cprintf(" Use default to setup ");
   initial();
   while( getch() != 0x1b );
   textattr(norm);
 }
 if(Num == 2)
 {
   cursoroff();
   clrscr();
   gotoxy(1,1);cprintf(" Pitch(mm./rev) = %d",ball);
   getch();
   cursoron();
   do
   {
     clrscr();
     gotoxy(1,1);cprintf(" Pitch(mm./rev) = ",ball);
     test = (int)get_option();
     if( test > 0) ball=test;
     else
     {
          gotoxy(1,1);
          if( test == 0 )break;
          cprintf("  Value out of limit!");
          getch();
     }
   }while(  test<=0 );
}
if(Num == 3)
{
  cursoroff();
  clrscr();
  gotoxy(1,1);cprintf("  Pulse/rev = %d",pulse);
  getch();
  cursoron();
```

```
    do
    {
      clrscr();
      gotoxy(1,1);cprintf("  Pulse/rev = ");
      test = (int)get_option();
      if(  test > 0) pulse=test;
      else
      {
            clrscr();
            if( test == 0 )break;
            gotoxy(1,1);
            cprintf("  Value out of limit!");
            getch();
      }
    }while(  test<=0 );
}
if(Num == 4)
{
  cursoroff();
  do
  {
    clrscr();
    gotoxy(1,1);clreol();
    cprintf("  P = %d",get_kp(0));
    getch();
    cursoron();
    do
    {
          gotoxy(1,1);
          clreol();
          cprintf("  P = ");
          test = (int)get_option();
          if( (test > 0)&&(test < 20)) prop=test;
          else
          {
            gotoxy(1,1);
            if ( test == 0 )break;
            cprintf("  Value out of limit!");
            getch();
```

```
        }
    }while( !((test > 0)&&(test<20)) );
    gotoxy(1,1);clreol();
    cprintf("  P = %d",prop);
    cursoroff();
    gotoxy(1,2);clreol();
    cprintf("  I = %d",get_ki(0));
    getch();
    cursoron();
    do
    {
          gotoxy(1,2);
          clreol();
          cprintf("  I = ");
          test = (int)get_option();
          if( (test > 0)&&(test<20)) i=test;
          else
          {
            gotoxy(1,2);
            if ( test == 0 )break;
            cprintf("  Value out of limit!");
            getch();
          }
    }while( !((test > 0)&&(test<20)) );
    gotoxy(1,2);clreol();
    cprintf("  I = %d",i);
    cursoroff();
    gotoxy(1,3);
    clreol();
    cprintf("  D = %d",get_kd(0));
    getch();
    cursoron();
    do
    {
          gotoxy(1,3);
          clreol();
          cprintf("  D = ");
          test = (int)get_option();
          if( (test > 0)&&(test<20)) d=test;
```

```
                    else
                    {
                      gotoxy(1,3);
                      if( test == 0)break;
                      cprintf("  Value out of limit!");
                      getch();
                    }
                }while( !((test > 0)&&(test<20)) );
                gotoxy(1,3);
                clreol();
                cprintf("  D = %d",d);
                gotoxy(1,4);cprintf("  Are you sure?(Y/N) : ");
                test = getch();
              }while( (test!=0x59)&&(test!=0x79) );
            }
  }



  void Select_file(int Num)
  {
          PULLDOWN *p;

          p=(PULLDOWN *)malloc(sizeof(PULLDOWN));
          p->l=20;
          p->t=9;
          p->r=59;
          p->b=11;
          p->save=NULL;
          Pull_for_file(p);
          cursoroff();
          Restor(p);
  }

  void Select_memory(Num)
  int Num;
  {
          PULLDOWN *p;
```

```
        p=(PULLDOWN *)malloc(sizeof(PULLDOWN));
        p->l=28;
        p->t=4+(Num-1);
        p->r=50;
        p->b=6+(Num-1);
        p->save=NULL;
        Pull_for_memory(p);
        cursoroff();
        Restor(p);
}

void Select_jog(Num)
int Num;
{
        PULLDOWN *p;

        p=(PULLDOWN *)malloc(sizeof(PULLDOWN));
        p->l=30;
        p->t=4+(Num-1);
        p->r=60;
        p->b=8+(Num-1);
        p->save=NULL;
        Pull_for_jog(p);
        Restor(p);
}

void Single_Block(Num)
{       PULLDOWN *p;

        p=(PULLDOWN *)malloc(sizeof(PULLDOWN));
        p->l=36;
        p->t=4+(Num-1);
        p->r=62;
        p->b=8+(Num-1);
        p->save=NULL;
        Pull_for_cycle(p);
        cursoroff();
        Restor(p);
}
```

```
void Select_setup(Num)
int Num;
{
        PULLDOWN *p;

        p=(PULLDOWN *)malloc(sizeof(PULLDOWN));
        if(Num == 5)
        {
          p->l=45;
          p->t=4+(Num-1);
          p->r=70;
          p->b=9+(Num-1);
        }
        else
        {
          p->l=45;
          p->t=4+(Num-1);
          p->r=70;
          p->b=6+(Num-1);
        }
        p->save=NULL;
        Pull_for_setup(p);
        cursoroff();
        Restor(p);
}

/*************************************************************************/

main()
{
        PULLDOWN *p[MAXMAINMENU+1];
        int i,j,*dn,mainmenu=1,row[MAXMAINMENU+1];

        /* START UP CONTROLLER CARD */
        startup();

        cursoroff();
        dn=(int *)malloc(sizeof(int));
        for(i=1;i<MAXMAINMENU+1;i++)
```

```
p[i]=(PULLDOWN *)malloc(sizeof(PULLDOWN));
for(i=1;i<MAXMAINMENU+1;i++) row[i]=1;

p[1]->l=4;
p[1]->t=2;
p[1]->r=20;
p[1]->max=4;
p[1]->b=p[1]->t+p[1]->max+1;
p[1]->text=" Load Program \r\n New Program \r\n Edit Program \r\n Exit";
p[1]->save=NULL;

p[2]->l=12;
p[2]->t=2;
p[2]->r=28;
p[2]->max=3;
p[2]->b=p[2]->t+p[2]->max+1;
p[2]->text=" Feed Rate \r\n Spindle Rate \r\n Jog Speed ";
p[2]->save=NULL;

p[3]->l=28;
p[3]->t=2;
p[3]->r=42;
p[3]->max=4;
p[3]->b=p[3]->t+p[3]->max+1;
p[3]->text=" Jog \r\n Home\r\n Set Zero\r\n Emerg.Stop ";
p[3]->save=NULL;

p[4]->l=38;
p[4]->t=2;
p[4]->r=55;
p[4]->max=4;
p[4]->b=p[4]->t+p[4]->max+1;
p[4]->text=" Cycle Start\r\n Cycle Stop\r\n Spindle ON\r\n Single Block ";
p[4]->save=NULL;

p[5]->l=50;
p[5]->t=2;
p[5]->r=65;
p[5]->max=5;
```

```
    p[5]->b=p[5]->t+p[5]->max+1;
    p[5]->text=" Default\r\n In. / mm.\r\n Ball Screw\r\n Encoder\r\n PID Filter";
    p[5]->save=NULL;

    textattr(high);
    clrscr();
    Menubar();
    About();
    initial();
    window(3,14,37,24);
    textattr(norm);clrscr();
    window(3,3,78,12);
    textcolor(WHITE);
    textbackground(BLACK);clrscr();
    window(40,14,78,24);
    textattr(norm);clrscr();
    window(1,1,80,25);
    textattr(norm);
    cycle_stop();
    display();
    toggle();
    online(0);
    online(1);
online(2);
    online(3);
    do
    {
     window(3,3,78,12);
     textcolor(14);
     textbackground(0);
     if(file_load[0]==0)clrscr();
     gotoxy(1,10);clreol();
     cprintf(" Program loaded: %ls",file_load);
     file = 1;
     *dn=mainmenu;
     do
     {
      i=Selection(p[mainmenu],dn,&row[mainmenu]);
      if(i==0) mainmenu=*dn;
```

```
}while(i==0);
window(1,1,80,25);
Pull(p[i]);
window(1,1,80,25);
switch(i)
{
  case 1 : switch(*dn)
           {
              case 1 : window(1,1,80,1);
                       clrscr();
                       Restor(p[i]);
                       Menubar();
                       window(3,3,78,12);
                       textcolor(14);
                       textbackground(0);
                       clrscr();
                       gotoxy(1,10);clreol();
                              cprintf(" Enter file_name for load: ");
                       gets(file_load);
                       if((lf=fopen(file_load,"r")) == NULL)
                       {
                       clrscr();
                       gotoxy(1,10);clreol();
                       cprintf(" Error in loading program!");
                       for(j=0;j<=8;j++)file_load[j] = 0;
                       getch();
                       gotoxy(1,10);clreol();
                      }
                      else
                      {
                       clrscr();
                       showfile(lf);
                       window(3,3,78,12);
                       gotoxy(1,10);clreol();
                       cprintf(" Program loaded: %ls",file_load);
                       fclose(lf);
                       cursoroff();
                      }
                      file = 0;
```

```
                    break;
                case 2  : window(1,1,80,1);
                    clrscr();
                    Restor(p[i]);
                    Menubar();
                    Newedit();
                    file = 0;
                    break;
                case 3  : window(1,1,80,1);
                    clrscr();
                    Restor(p[i]);
                    Menubar();
                    Edit();
                    file = 0;
                    break;
                case 4  : Select_file(*dn);
                    *dn=quit;
                    break;
                default : break;
            }
            break;
        case 2 : switch(*dn)
            {
                case  1 : Select_memory(*dn);
                    break;
                case  2 : Select_memory(*dn);
                    break;
                case  3 : Select_memory(*dn);
                    break;
                default : break;
            }
            break;
        case 3 : switch(*dn)
            {
                case  1 : Select_jog(*dn);
                    break;
                case  2 : Select_jog(*dn);
                    break;
                case  3 : Select_jog(*dn);
```

```
                    break;
                case  4 : Select_jog(*dn);
                    break;
                default : break;
            }
            break;
    case 4 : switch(*dn)
            {
                case  1 : window(1,1,80,1);
                    clrscr();
                    Restor(p[i]);
                    Menubar();
                    if((con1)&&(file_load[0] !=0))
                    cycle_start();
                    cursoroff();
                    file=0;
                    break;
                case  2 : if(con2)cycle_stop();
                    break;
                case  3 : if(con3)spindel_on();
                    break;
                case  4 : Single_Block(*dn);
                    single=1;
                    break;
                default : break;
            }
            break;
    case 5 : switch(*dn)
            {
                case  1 : Select_setup(*dn);
                    break;
                case  2 : toggle();
                    break;
                case  3 : Select_setup(*dn);
                    break;
                case  4 : Select_setup(*dn);
                    break;
                case  5 : Select_setup(*dn);
                    break;
```

```
                        default : break;
                    }
                }
                if (file == 1)Restor(p[i]);
                mainmenu=i;
            }while(i!=1 || *dn!=4  );
            textattr(07);
            window(1,1,80,25);
            cursoron();
            clrscr();

}
```

NEWEDIT.C

```
#include <conio.h>
#include <stdio.h>
#include <stdlib.h>
#include <dos.h>

#define KBD_PORT 0x60

float  X[1000],Y[1000],Z[1000],D[1000],R[1000],P[1000],Q[1000];
long   N[1000],G[1000],M[1000],F[1000],S[1000];
char   x,y,z,n,g,m,r,p,q,k,f,s,dim,star;
int    j=8,line,col,enteri,enterf;
FILE   *fp;

void go() /* return value of position of cursor */
{
        col  = wherex();
        line = wherey();
}

int get_int() /* return integer value of characters which user enter */
{
        char s[10],c;
        int i;

        while(kbhit())getch();
        for(i=0; (c=getchar()) != '\n';i=i+1) s[i]=c;
        if( i < 1 )enteri=1;
        else enteri = 0;
        s[i]='\0';
        return(atol(s));
}

float get_float() /* return floating point value */
{
        char st[10],c;
        int i;

        while(kbhit())getch();
        for(i=0; (c=getchar()) != '\n';i=i+1) st[i]=c;
```

```
        if( i < 1 )enterf=1;
        else enterf=0;
        st[i]='\0';
        return(atof(st));
}

void set_data(int k)
{
        int i;

        for(i=k;i<=1000;i++)N[i]=10000;
        for(i=k;i<=1000;i++)G[i]=10000;
        for(i=k;i<=1000;i++)M[i]=10000;
        for(i=k;i<=1000;i++)X[i]=10000;
        for(i=k;i<=1000;i++)Y[i]=10000;
        for(i=k;i<=1000;i++)Z[i]=10000;
        for(i=k;i<=1000;i++)D[i]=10000;
        for(i=k;i<=1000;i++)R[i]=10000;
        for(i=k;i<=1000;i++)S[i]=10000;
        for(i=k;i<=1000;i++)P[i]=10000;
        for(i=k;i<=1000;i++)Q[i]=10000;
        for(i=k;i<=1000;i++)F[i]=10000;
        j=8;
        n ='N';
        g ='G';
        m ='M';
        x ='X';
        y ='Y';
        z ='Z';
        dim ='D';
        r ='R';
        p ='P';
        q ='Q';
        s ='S';
        f ='F';
        star ='*';
}

void print_data(int k) /* print data which user enter on screen */
```

```
{
        cursoroff();
        if( (N[k] >= 1)&&(N[k] != 10000) ) cprintf(" N%d",N[k]);
        if( (G[k] < 10000)&&( N[k] >= 1) ) cprintf(" G%02d",G[k]);
        if( (M[k] < 10000)&&( N[k] >= 1) ) cprintf(" M%02d",M[k]);
        if( X[k] != 10000 ) cprintf(" X%.3f",X[k]);
        if( Y[k] != 10000 ) cprintf(" Y%.3f",Y[k]);
        if( Z[k] != 10000 ) cprintf(" Z%.3f",Z[k]);
        if( D[k] != 10000 ) cprintf(" D%.3f",D[k]);
        if( Q[k] != 10000 ) cprintf(" Q%.3f",Q[k]);
        if( R[k] != 10000 ) cprintf(" R%.3f",R[k]);
        if( S[k] != 10000 ) cprintf(" S%d",S[k]);
        if( P[k] != 10000 ) cprintf(" P%.3f",P[k]);
        if( F[k] != 10000 ) cprintf(" F%ld",F[k]);
        if( (N[k] >= 1)&&(N[k]!=10000) ) cprintf(" *");
        clreol();
}

void shift_data(int k) /* shift data from one line to another */
{
        N[k+1] = N[k]+1;
        G[k+1] = G[k];
        M[k+1] = M[k];
        X[k+1] = X[k];
        Y[k+1] = Y[k];
        Z[k+1] = Z[k];
        D[k+1] = D[k];
        R[k+1] = R[k];
        S[k+1] = S[k];
        P[k+1] = P[k];
        Q[k+1] = Q[k];
        F[k+1] = F[k];
}

void del_data(int k)
{
        N[k] = N[k+1]-1;
        G[k] = G[k+1];
        M[k] = M[k+1];
```

```
            X[k] = X[k+1];
            Y[k] = Y[k+1];
            Z[k] = Z[k+1];
            D[k] = D[k+1];
            R[k] = R[k+1];
            S[k] = S[k+1];
            P[k] = P[k+1];
            Q[k] = Q[k+1];
            F[k] = F[k+1];
    }

    void save_data(int k) /* save data into file */
    {
            if( (N[k] >= 1)&&(N[k] != 10000) ) fprintf(fp,"%c%d",n,N[k]);
            if( (G[k] < 10000)&&( N[k] >= 1) ) fprintf(fp," %c%02d",g,G[k]);
            if( (M[k] < 10000)&&( N[k] >= 1) ) fprintf(fp," %c%02d",m,M[k]);
            if( X[k] != 10000 ) fprintf(fp," %c%.3f",x,X[k]);
            if( Y[k] != 10000 ) fprintf(fp," %c%.3f",y,Y[k]);
            if( Z[k] != 10000 ) fprintf(fp," %c%.3f",z,Z[k]);
            if( D[k] != 10000 ) fprintf(fp," %c%.3f",dim,D[k]);
            if( Q[k] != 10000 ) fprintf(fp," %c%.3f",q,Q[k]);
            if( R[k] != 10000 ) fprintf(fp," %c%.3f",r,R[k]);
            if( S[k] != 10000 ) fprintf(fp," %c%d",s,S[k]);
            if( P[k] != 10000 ) fprintf(fp," %c%.3f",p,P[k]);
            if( F[k] != 10000 ) fprintf(fp," %c%d",f,F[k]);
            if( (N[k] >= 1)&&(N[k]!=10000) ) fprintf(fp," %c\n",star);

    }

    void  refresh(int direct) /* display data on screen */
    {
            int i;

            window(3,3,78,11);
            textcolor(14);
            textbackground(0);
            clrscr();
            for(i=1;i<=8;++i)
            {
```

```
                gotoxy(1,i);
                print_data(direct-8+i);
              }
              cursoron();
    }

    void clear(int l) /* clear data in buffer array*/
    {
              G[l] = 10000;
              M[l] = 10000;
              X[l] = 10000;
              Y[l] = 10000;
              Z[l] = 10000;
              D[l] = 10000;
              R[l] = 10000;
              S[l] = 10000;
              P[l] = 10000;
              Q[l] = 10000;
              F[l] = 10000;
    }

    float getpoint(FILE *fp)
    {
              char ch;
              char a[10];
              int i=0;
              float x;
              do
              {
              ch=fgetc(fp);
              a[i]=ch;
              i=i+1;
              }while(ch!=0x20&ch!=0x0a);
              x=atof(a);
              return x;
    }

    int getnum(FILE *fp)
    {
```

```
        char ch;
        char a[10];
        int i=0,x;
        do
        {
         ch=fgetc(fp);
         a[i]=ch;
         i=i+1;
        }while(ch!=0x20&ch!=0x0a);
        x=atoi(a);
        return x;
}

void showfile(FILE *fp)
{
        int j=8;
        char st;

        set_data(0);
        while(!feof(fp))
        {
                st=fgetc(fp);
         switch(st)
         {
          case 0x4e: N[j]=getnum(fp);
                     break;
          case 0x47: G[j]=getnum(fp);
                     break;
          case 0x4d: M[j]=getnum(fp);
                     break;
          case 0x58: X[j]=getpoint(fp);
                     break;
          case 0x59: Y[j]=getpoint(fp);
                     break;
          case 0x5a: Z[j]=getpoint(fp);
                     break;
          case 0x44: D[j]=getpoint(fp);
                     break;
          case 0x50: P[j]=getpoint(fp);
```

```
                    break;
                case 0x51: Q[j]=getpoint(fp);
                    break;
                case 0x52: R[j]=getpoint(fp);
                    break;
                case 0x53: S[j]=getnum(fp);
                    break;
                case 0x46: F[j]=getnum(fp);
                    break;
                case 0x0a: j++;
                    break;
              }
             }
            j=15;
            refresh(j);

}
/* PATTERN OF G_code FUNCTION */
void g00(void)
{
        float temp;

        cprintf(" X");
        go();
        temp=get_float();
        if( enterf!= 1 )X[j]=temp;
        gotoxy(col+8,line);
        cprintf(" Y");
        go();
        temp=get_float();
        if( enterf!=1)Y[j]=temp;
        gotoxy(col+8,line);
        cprintf(" Z");
        go();
        temp=get_float();
        if(enterf!=1)Z[j]=temp;
        gotoxy(col+8,line);
        cprintf("*");
}
```

```
void g01()
{
        float tempf;
        int tempi;

        cprintf(" X");
        go();
        tempf=get_float();
        if( enterf!= 1 )X[j]=tempf;
        gotoxy(col+8,line);
        cprintf(" Y");
        go();
        tempf=get_float();
        if( enterf!=1)Y[j]=tempf;
        gotoxy(col+8,line);
        cprintf(" Z");
        go();
        tempf=get_float();
        if(enterf!=1)Z[j]=tempf;
        gotoxy(col+8,line);
        cprintf(" F");
        go();
        tempi=get_int();
        if(enteri!=1)F[j]=abs(tempi);
        gotoxy(col+8,line);
        cprintf("*");
}

void g02()
{
        float tempf;
        int tempi;
        cprintf(" X");
        go();
        tempf=get_float();
        if( enterf!= 1 )X[j]=tempf;
        gotoxy(col+8,line);
        cprintf(" Y");
```

```
            go();
            tempf=get_float();
            if( enterf!=1)Y[j]=tempf;
            gotoxy(col+8,line);
            cprintf(" Z");
            go();
            tempf=get_float();
            if(enterf!=1)Z[j]=tempf;
            gotoxy(col+8,line);
            cprintf(" R");
            go();
            tempf=get_float();
            if(enterf!=1)R[j]=tempf;
            gotoxy(col+8,line);
            cprintf(" F");
            go();
            tempi=get_int();
            if(enteri!=1)F[j]=abs(tempi);
            gotoxy(col+8,line);
            cprintf("*");
 }

 void g04()
 {
            float tempf;

            cprintf(" P");
            go();
            tempf=get_float();
            if( enterf!= 1 )P[j]=abs(tempf);
            gotoxy(col+8,line);
            cprintf("*");
}
void g40()
{
           go();
           gotoxy(col+8,line);
           cprintf("*");
}
```

```
void g41()
{
        float tempf;

        cprintf(" D");
        go();
        tempf=get_float();
        if( enterf!= 1 )D[j]=abs(tempf);
        gotoxy(col+8,line);
        cprintf("*");
}

void g73()
{
        float tempf;
        int tempi;

        cprintf(" X");
        go();
        tempf=get_float();
        if( enterf!= 1 )X[j]=tempf;
        gotoxy(col+8,line);
        cprintf(" Y");
        go();
        tempf=get_float();
        if( enterf!=1)Y[j]=tempf;
        gotoxy(col+8,line);
        cprintf(" Z");
        go();
        tempf=get_float();
        if(enterf!=1)Z[j]=tempf;
        gotoxy(col+8,line);
        cprintf(" Q");
        go();
        tempf=get_float();
        if(enterf!=1)Q[j]=tempf;
        gotoxy(col+8,line);
        cprintf(" R");
```

```
        go();
        tempf=get_float();
        if(enterf!=1)R[j]=tempf;
        gotoxy(col+8,line);
        cprintf(" F");
        go();
        tempi=get_int();
        if(enteri!=1)F[j]=abs(tempi);
        gotoxy(col+8,line);
        cprintf("*");
}


void g82()
{
        float tempf;
        int tempi;

        cprintf(" X");
        go();
        tempf=get_float();
        if( enterf!= 1 )X[j]=tempf;
        gotoxy(col+8,line);
        cprintf(" Y");
        go();
        tempf=get_float();
        if( enterf!=1)Y[j]=tempf;
        gotoxy(col+8,line);
        cprintf(" Z");
        go();
        tempf=get_float();
        if(enterf!=1)Z[j]=tempf;
        gotoxy(col+8,line);
        cprintf(" R");
        go();
        tempf=get_float();
        if(enterf!=1)R[j]=tempf;
        gotoxy(col+8,line);
        cprintf(" P");
```

```
        go();
        tempf=get_float();
        if(enterf!=1)P[j]=abs(tempf);
        gotoxy(col+8,line);
        cprintf(" F");
        go();
        tempi=get_int();
        if(enteri!=1)F[j]=abs(tempi);
        gotoxy(col+8,line);
        cprintf("*");
}

/* PATTERN OF M_code FUNCTION */
void m03()
{
        int tempi;

        cprintf(" S");
        go();
        tempi=get_int();
        if(enteri!=1)S[j]=abs(tempi);
        gotoxy(col+8,line);
        cprintf("*");
}

void build(int tempj)
{
        int t;

        cursoron();
        window(3,3,78,12);
        do
        {
          gotoxy(1,9);clreol();
          N[tempj]=tempj-7;
          cprintf(" N%d",N[tempj]);
          while(kbhit())getch();
          cprintf(" G");
          go();
```

```
        t=get_int();                    /* wait for Gxx */
        if((enteri != 1)
&&((t==00)||(t==01)||(t==02)||(t==03)||(t==04)||(t==20)||(t==21)||(t==40)||(t==41)||(t==42)||(t==73)||(t==80)||(t
==81)||(t==82)||(t==83)||(t==85)||(t==86)||(t==89)||(t==90)||(t==91)||(t==92)||(t==98)||(t==99)))
        {
          G[tempj]=t;
          gotoxy(col+2,line);
        }
        else
        {
          gotoxy(col-1,line);
          clreol();
          cprintf("M");
          go();
          t=get_int();
          if( (enteri != 1)&&((t==02)||(t==03)||(t==04)||(t==05)) ) M[tempj]=t;
          gotoxy(col+2,line);
        }
      }while( (G[tempj]==10000)&&(M[tempj]==10000) );
      switch(G[tempj])
      {
        case 00: g00();
                 break;
        case 01: g01();
                 break;
        case 02: g02();
                 break;
        case 03: g02();
                 break;
        case 04: g04();
                 break;
        case 40: g40();
                 break;
        case 41: g41();
                 break;
        case 42: g41();
                 break;
        case 73: g73();
                 break;
```

```
                case 81: g02();
                         break;
                case 82: g82();
                         break;
                case 83: g73();
                         break;
                case 85: g02();
                         break;
                case 86: g02();
                         break;
                case 89: g82();
                         break;
                case 92: g00();
                         break;
                default: break;
              }
              switch(M[tempj])
              {
                case 03: m03();
                         break;
                case 04: m03();
                         break;
                default: break;
              }
}

void Newedit()
{
        char file_name[8];
        int  i,c,test,move,tempj;

        set_data(0);
        window(3,3,78,12);
        textcolor(14);
        textbackground(0);
        clrscr();
        gotoxy(1,10);
        cprintf(" Enter file_name for new creating: ");
        gets(file_name);
```

```
            test = 0;

            if((fp=fopen(file_name,"r")) != NULL )
            {
              gotoxy(1,10);clreol();
                    cprintf(" Duplicate file_name!");
              getch();
              gotoxy(1,10);clreol();
            }
            else{
            if((fp=fopen(file_name,"w")) == NULL )
            {
              gotoxy(1,10);clreol();
              cprintf(" Error in creating program!");
              getch();
              gotoxy(1,10);clreol();
            }
            else test = 1;}
if( test == 1 )
{/*0 level*/
/*section 0.1 : input file name*/
gotoxy(1,10);clreol();
cprintf(" Program : %s",file_name);
do
{/*1st level*/
  i = 0x1c;
  do
  {/*2nd level*/
  /* secton 2.1 to input something that is g code line */
        if( i == 0x1c)
        {
         build(j);
         do
         {
          getch();
          i = inportb(KBD_PORT);    /* wait enter end of g code line*/
         }while( (i!= 0x1c)&&(i!= 0x53) );
        }
```

```
/* section 2.2 to check input something -> while (i != 0x53) do til 0x53*/
        do
        {/*3rd level : to check input something*/
         if( i == 0x53 )
         {
          go();
          if( line != 9 )
          {
           gotoxy(1,8);
           clreol();
          }
          else
          {
           gotoxy(1,9);
           clreol();
          }/*if (line !=9)*/
          clear(j);
          build(j);
          getch();
          while((inportb(KBD_PORT)!=0x1c)&&(inportb(KBD_PORT)!=0x53));
          i = inportb(KBD_PORT);
         }/*if (i==0x53)*/
        }while( i == 0x53 );

/*section 2.3 : refresh and input something after i == 0x53*/

        refresh(j);
        getch();
        i = inportb(KBD_PORT);   /* wait up,down,enter after entered line */

/*secton 2.4 : check input something that i != 0x1c*/

        if( i != 0x1c )
        {/*3rd level*/
         tempj = j;
         move = j;
         getch();
         i = inportb(KBD_PORT);
         do
```

```
{
 switch(i)
 { case 0x01:window(3,3,78,12);
                    gotoxy(1,10);clreol();
                    cprintf(" Do you want to save?(Y/N): ");
                    getch();
                    i = inportb(KBD_PORT);
                    switch(i)
                    {
                     case 0x31:fclose(fp);
                               goto no;
                     case 0x15:fclose(fp);
                               gotoxy(1,10);clreol();
                               fp=fopen(file_name,"w");
                               i = 8;
                               while( N[i] != 10000 )
                               {
                               save_data(i);
                               i = i+1;
                               }
                               fclose(fp);
                               goto no;
                     default: fclose(fp);
                              goto no;
                    }
                    break;
 case 0x47: move=8;
                    refresh(move);
                    break;
 case 0x4F: move=j;
                    refresh(move);
                    break;
 case 0x49: if(move > 16) move-=7;
                    else move=8;
                    refresh(move);
                    delay(100);
                    break;
 case 0x51: if(move <j-8) move+=7;
                    else move=j;
```

```
                    refresh(move);
                    delay(100);
                    break;
case 0x48: if(move > 8)refresh(--move);
                    delay(100);
                    break;
case 0x50: if(move < j)refresh(++move);
                    delay(100);
                    break;
case 0x53: do
                    {
                     tempj = j;
                     j = move;
                     go();
                     if( line != 9 )
                     {
                      gotoxy(1,8);
                      clreol();
                     }
                     else
                     {
                      gotoxy(1,9);
                      clreol();
                     }
                     clear(j);
                     build(j);
      while((inportb(KBD_PORT)!=0x1c)&&(inportb(KBD_PORT)!=0x53))
                     getch();
                     i = inportb(KBD_PORT);
                    }while( i == 0x53 );
                    j = tempj;
                    break;
case 0x52: refresh(move-1);
                    c = 8;
                    while( N[c] != 10000 ) c = c+1;
                    for(i=c-1;i>=move;i--) shift_data(i);
                    j = i+1;
                    do
                    {
```

```
                                clear(j);
                                build(j);
                                /*getch();*/
                        while((inportb(KBD_PORT)!=0x1c)&&(inportb(KBD_PORT)!=0x53))
                                getch();
                                i = inportb(KBD_PORT);
                               }while( i == 0x53 );
                               j = tempj+1;
                               tempj = j;
                               break;
                  default  : break;
                 }/*switch*/
                 refresh(move);
                 getch();
                 i = inportb(KBD_PORT);  /* wait up,down,enter after up,down*/
                }while( i != 0x1c);
               }/*if (i!=0x1c*/

  /* section 2.5 : after check that while (i!=0x1c) do til 0x1c */
          refresh(j);
          j=j+1;
    }while( (M[j-1] != 02)&&( M[move]!= 02) );

  /* section 1.3 */

    window(3,3,78,12);
    gotoxy(1,10);clreol();
    cprintf(" Do you want to save?(Y/N): ");
    getch();
    i = inportb(KBD_PORT);
    if( i != 0x15 )
    {
                fclose(fp);
                goto no;
                }
  }while( i != 0x15 );

 /* section 0.2 */
  /* SAVE DATA INTO FILE */
```

```
    i = 8;
    while( N[i] != 10000 )
    {
      save_data(i);
      i = i+1;
    }
    fclose(fp);
   }/*if (test==1)*/
   no:window(3,3,78,12);
   textcolor(14);
   textbackground(0);
   clrscr();
   cursoroff();
}

Edit()
{
        char file_name[10];
        char st;
        int i,c,test,move,tempj;

        set_data(0);
        window(3,3,78,12);
        textcolor(14);
        textbackground(0);
        clrscr();
        gotoxy(1,10);
        cprintf(" Enter file_name for edit: ");
        gets(file_name);
        test = 0;
        if((fp=fopen(file_name,"r")) == NULL )
        {
         gotoxy(1,10);clreol();
         cprintf(" File not found!");
         getch();
         gotoxy(1,10);clreol();
        }
       else test = 1;
if( test == 1 )
```

```
{
        while(!feof(fp))
        {
                st=fgetc(fp);
        switch(st)
        {
        case 0x4e: N[j]=getnum(fp);
                        break;
        case 0x47: G[j]=getnum(fp);
                        break;
        case 0x4d: M[j]=getnum(fp);
                        break;
        case 0x58: X[j]=getpoint(fp);
                        break;
        case 0x59: Y[j]=getpoint(fp);
                        break;
        case 0x5a: Z[j]=getpoint(fp);
                        break;
        case 0x44: D[j]=getpoint(fp);
                        break;
        case 0x50: P[j]=getpoint(fp);
                        break;
        case 0x51: Q[j]=getpoint(fp);
                        break;
        case 0x52: R[j]=getpoint(fp);
                       break;
        case 0x53: S[j]=getnum(fp);
                        break;
        case 0x46: F[j]=getnum(fp);
                        break;
        case 0x0a: j++;
                        break;
        }
        }
       do
       { i=0x1c;
                do{
       refresh(j);
getch();
```

```
i = inportb(KBD_PORT);

if( i != 0x1c )
 {
 tempj = j;
 move = j;
 getch();
 i = inportb(KBD_PORT);
 do
 {
  switch(i)
  {
   case 0x01:window(3,3,78,12);
                    gotoxy(1,10);clreol();
                    cprintf(" Do you want to save?(Y/N): ");
                    getch();
                    i = inportb(KBD_PORT);
                    switch(i)
                    {
                     case 0x15:fclose(fp);
                               gotoxy(1,10);clreol();
                               fp=fopen(file_name,"w");
                               i = 8;
                               while( N[i] != 10000 )
                               {
                               save_data(i);
                               i = i+1;
                               }
                               fclose(fp);
                               goto no;
                     default: fclose(fp);
                               goto no;
                    }
  case 0x47: move=8;
                    refresh(move);
                    break;
  case 0x4F: move=j;
                    refresh(move);
                    break;
```

```
case 0x49: if(move > 16) move-=7;
                else move=8;
                refresh(move);
                delay(100);
                break;
case 0x51: if(move <j-8) move+=7;
                else move=j;
                refresh(move);
                delay(100);
                break;
case 0x48: if(move > 8)refresh(--move);
                delay(100);
                break;
case 0x50: if(move < j)refresh(++move);
                delay(100);
                break;

case 0x53: tempj=j;
                        j=move;
                        c=8;
                        while(N[c]!=10000) c=c+1;
                        for(i=j;i<=c-1;i++) del_data(i);
                        set_data(c-1);
                        j=tempj-1;
                        tempj=j;
                        getch();
                        break;


case 0x52: refresh(move-1);
          c = 8;
          while( N[c] != 10000 ) c = c+1;
          for(i=c-1;i>=move;i--) shift_data(i);
          j = i+1;
          do
          {
            clear(j);
            build(j);
            /*getch();*/
```

```
                    while((inportb(KBD_PORT)!=0x1c)&&(inportb(KBD_PORT)!=0x53))
                        getch();
                        i = inportb(KBD_PORT);
                      }while( i == 0x53 );
                      j = tempj+1;
                      tempj = j;
                      break;
                default  : break;
               }/*switch*/
               refresh(move);
               getch();
                    i = inportb(KBD_PORT);
             }while( i != 0x1c);
           }refresh(j);
           j=j+1;

           /*getch();
           i=inportb(KBD_PORT);
           }while(i!=0x4f);  */
          }while((M[j-1]!=02)&&(M[move]!=02));

   window(3,3,78,12);
   gotoxy(1,10);clreol();
   cprintf(" Do you want to save?(Y/N): ");
   getch();
   i = inportb(KBD_PORT);
                    if( i != 0x15 )
                    {
          fclose(fp);
          goto no;
                    }
         }while( i != 0x15 );
fclose(fp);
fp=fopen(file_name,"w");

 /* SAVE DATA INTO FILE */
i = 8;
while( N[i] != 10000 )
{
```

```
        save_data(i);
        i = i+1;
      }
            fclose(fp);
    }/*if (test==1)*/
    no:window(3,3,78,12);
    textcolor(14);
    textbackground(0);
    clrscr();
    cursoroff();
}
```

STARTUP.C

```
#include<stdio.h>
#include<math.h>
#include<dos.h>
#include<conio.h>
#include"mc1401.h"
#include"hrdwr.h"
#include"host_io.h"
/*void change_axis(int axis)
{  switch(axis)
   {
    case 0:
          set_1(0);
          break;
    case 1:
          set_2(0);
          break;
    case 2:
          set_3(0);
          break;
    case 3:
          set_4(0);
          break;
    default:
          break;
   }
}     */


void startup(void)
{  int i;
   reset(0);
   update(0);
   for (i=0;i<=1;i++)
    { change_axis(i);
          axis_on(0);
          mtr_on(0);
          set_output_dac16(0);
          set_prfl_trap(0);
          set_acc(0L);
```

```
    set_vel(0L);
    set_fltr_pid(0);
        set_kp(10L);
        set_ki(10L);
        set_kd(10L);
        synch_prfl(0);
        update(0);
  }
  for (i=2;i<=3;i++)
  { change_axis(i);
        axis_on(0);
        mtr_on(0);
        set_output_dac16(0);
        set_prfl_trap(0);
        set_acc(0L);
    set_vel(0L);
    set_fltr_pid(0);
        set_kp(10L);
        set_ki(10L);
        set_kd(10L);
        synch_prfl(0);
        update(0);
  }


}

□
```