# รายงานการวิจัยฉบับสมบูรณ์

## การสร้างส่วนควบคุมเครื่องกัดแนวดิ่งซีเอ็นซี ระยะที่ 1

## Computer Numerical Control for Vertical Milling Machine Phase 1

โดย

นายกวิน สนธิเพิ่มพูน

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ศูนย์เทคโนโลยีอิเล็กทรอนิกส์และคอมพิวเตอร์แห่งชาติ

สำนักงานพัฒนาวิทยาศาสตร์และเทคโนโลยีแห่งชาติ

กระทรวงวิทยาศาสตร์ เทคโนโลยีและสิ่งแวดล้อม

NATIONAL ELECTRONICS AND COMPUTER TECHNOLOGY CENTER

National Science and Technology Development Agency
Ministry of Science Technology and Environment

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# รายงานการวิจัยฉบับสมบูรณ์

## การสร้างส่วนควบคุมเครื่องกัดแนวดิ่งซีเอ็นซี ระยะที่ 1

## Computer Numerical Control for Vertical Milling Machine Phase 1

โดย

นายกวิน สนธิเพิ่มพูน

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# รายงานการวิจัยและพัฒนา

โครงการ การพัฒนาวงจรอิเล็กทรอนิกส์เพื่ออุตสาหกรรม

การสร้างส่วนควบคุมเครื่องกัดแนวดิ่ง CNC ระยะที่ 1

COMPUTER NUMERICAL CONTROL FOR VERTICAL MILLING MACHINE PHASE I


กวิน สนธิเพิ่มพูน

ประภาษ อุคคกิมาพันธ์

พิพัฒน์ เลาหสงคราม

สิงห์ทอง พัฒนเศรษฐานนท์

ณรงค์ สกุลศิริรัตน์

สำนักวิจัยและบริการคอมพิวเตอร์

และ

ภาควิชา การวัดคุมทางอุตสาหกรรม คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง



ได้รับทุนสนับสนุนจากศูนย์เทคโนโลยีอิเล็กทรอนิกส์และคอมพิวเตอร์แห่งชาติ

กระทรวงวิทยาศาสตร์เทคโนโลยีและสิ่งแวดล้อม

ปีงบประมาณ 2534



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# บทคัดย่อ

การวิจัยและพัฒนาต้นแบบส่วนควบคุมเครื่องกัดแนวดิ่งซีเอ็นซี (Computerized Numerical Control for Vertical Milling Machine) การพัฒนาส่วนใหญ่จะอยู่ในส่วนของการพัฒนาฮาร์ดแวร์ และซอฟท์แวร์สำหรับควบคุมการเคลื่อนที่ของระบบมอเตอร์ 3 แกนพร้อมๆ กัน เพื่อใช้ประโยชน์ในการควบคุมเครื่องจักรกล สำหรับการผลิตชิ้นงานตามที่ออกแบบด้วยระบบคอมพิวเตอร์ และงานอีกส่วนของการพัฒนาจะเป็นการสร้างต้นแบบของเครื่องจักรกล ซึ่งต้องอาศัยความสามารถทางด้านเครื่องจักรกล ในส่วนของการพัฒนาโปรแกรมจะอ้างอิงกับการใช้งานที่สะดวกและคำสั่งมาตรฐาน ISO ดังนั้นประโยชน์ของการวิจัยและพัฒนาจะได้ต้นแบบของเครื่องควบคุมดังกล่าว ที่มีราคาลดลงและติดตั้งได้สะดวกและสามารถนำไปใช้ในการควบคุมเครื่องจักรกลประเภทอื่นที่ต้องการตำแหน่งแน่นอน เช่น เครื่องเจียรไน เครื่องตัดกระดาษ เครื่องตัดเหล็ก ฯลฯ

# ABSTRACT

The purpose of this project is to develope and construct a prototype of an computer numerical control (CNC) for vertical milling machine locally in Thailand. The machine consists mainly of the hardware part e.g. a computer, the software part e.g., the PID servo loop algorithm and an user interface.

The development and construction of a CNC in Thailand is seen as only be beneficial to the Industry of Thailand.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# สารบัญเรื่อง

## (Table of Contents)

**หัวข้อ** **หน้า**


**1. บทนำ**

1.1 สถานภาพทางเทคโนโลยีต่างประเทศ 7

1.2 สถานภาพทางเทคโนโลยีภายในประเทศ 7

1.3 สถานภาพทางเทคโนโลยีของหน่วยงานหรือโครงการวิจัย 8

1.4 จุดมุ่งหมายของโครงการวิจัยและพัฒนา 9

1.5 วิธีการและแผนการดำเนินงาน 9

1.6 เนื้อหาที่นำเสนอในรายงานการวิจัย 9

**2. ทฤษฎีการควบคุมมอเตอร์และการออกแบบฮาร์ดแวร์ DSP**

2.1 ทฤษฎีการควบคุมมอเตอร์ 11

2.2 การออกแบบฮาร์ดแวร์ DSP 14

**3. การออกแบบและสร้างเครื่องกัดแนวดิ่งซีเอ็นซี**

3.1 การออกแบบโครงสร้างทางด้านเครื่องกล 26

3.2 คำนวณน้ำหนักของส่วนต่างๆ 28

3.3 การคำนวณขนาดเซอร์โวมอเตอร์ในการขับเคลื่อนแนวแกน x,y และ z 37

**4. การออกแบบและพัฒนาโปรแกรมเครื่องกัดแนวดิ่งซีเอ็นซี**

4.1 กล่าวนำ 43

4.2 การออกแบบในส่วนโปรแกรม 43

4.3 การออกแบบโปรแกรมในส่วนของการจำลองการทำงาน ของโปรแกรม (Graphic Simulation) 50

**5. สรุปและข้อเสนอแนะ**

5.1 ด้านเครื่องจักรกล 52

5.2 ด้านส่วนควบคุม 52

5.3 ด้านส่วนคอมพิวเตอร์ 53

**หนังสืออ้างอิง** 54

**ภาคผนวก** 54


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# สารบัญตาราง

## (List of Tables)

ตารางที่ หน้า

3.1 ข้อมูลของตัวแปรต่างๆ ในการคำนวณขนาดกำลังของเซอร์โวมอเตอร์ 37
4.1 รายละเอียดของ ISO-CODE 43
4.2 รายละเอียดของ M-CODE 45

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# สารบัญภาพ

## (List of Figures)


| รูปที่ | | หน้า |
|---|---|---|
| 1.1 | ส่วนประกอบต่างๆ ของระบบเครื่องกัดแนวดิ่งซีเอ็นซี | 6 |
| 2.1 | Block Diagram ของตัวควบคุมแบบ PID ระบบ Digital | 11 |
| 2.2 | Block Diagram ในรายละเอียดของส่วนควบคุมแบบ PID | 12 |
| 2.3 | ระบบ Digital Signal Processing | 14 |
| 2.4 | วิวัฒนาการของไอซีตระกูล TMS320C25 | 16 |
| 2.5 | Block Diagram ของวงจรทั้งหมด | 18 |
| 2.6 | วงจร Address Selector | 19 |
| 2.7 | วงจร Data Selector | 20 |
| 2.8 | วงจรหน่วยความจำ | 21 |
| 2.9 | วงจร TMS320C25 | 22 |
| 3.1 | โครงสร้างของเครื่องกัดแนวดิ่งซีเอ็นซี ส่วนที่ 1 | 27 |
| 3.2 | โครงสร้างของเครื่องกัดแนวดิ่งซีเอ็นซี ส่วนที่ 2 | 29 |
| 3.3 | โครงสร้างของเครื่องกัดแนวดิ่งซีเอ็นซี ส่วนที่ 3 | 30 |
| 3.4 | โครงสร้างของเครื่องกัดแนวดิ่งซีเอ็นซี ส่วนที่ 4 | 31 |
| 3.5 | Ball Screw ของแกน X และ Y | 32 |
| 3.6 | Ball Screw ของแนวแกน Z | 33 |
| 3.7 | Linear motion ของโครงสร้างเครื่องจักรในแนวแกน X,Y และ Z | 34 |
| 3.8 | ชุดเพลาขับหัวเครื่อง 1 | 35 |
| 3.9 | ชุดเพลาขับหัวเครื่อง 2 | 36 |
| 3.10 | รูปแบบของเครื่องกัดโลหะแนวดิ่งซีเอ็นซี เมื่อประกอบทุกส่วนเข้าด้วยกัน | 36 |
| 3.11 | การติดตั้งมอเตอร์กับ Ball screw แบบตรงในแนวแกน X และ Y | 42 |
| 3.12 | การติดตั้งมอเตอร์กับ Ball screw แบบตรงในแนวแกน Z | 42 |
| 4.1 | ฟังก์ชั่นการทำงาน AUTOMATIC MODE | 47 |
| 4.2 | ฟังก์ชั่นการทำงาน MANUAL MODE | 48 |
| 4.3 | ฟังก์ชั่นการทำงาน JOG MODE | 48 |
| 4.4 | ฟังก์ชั่นการทำงาน EDIT MODE | 50 |
| 4.5 | ฟังก์ชั่นการทำงาน GRAPHICS MODE | 51 |


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# บทที่ 1
# บทนำ

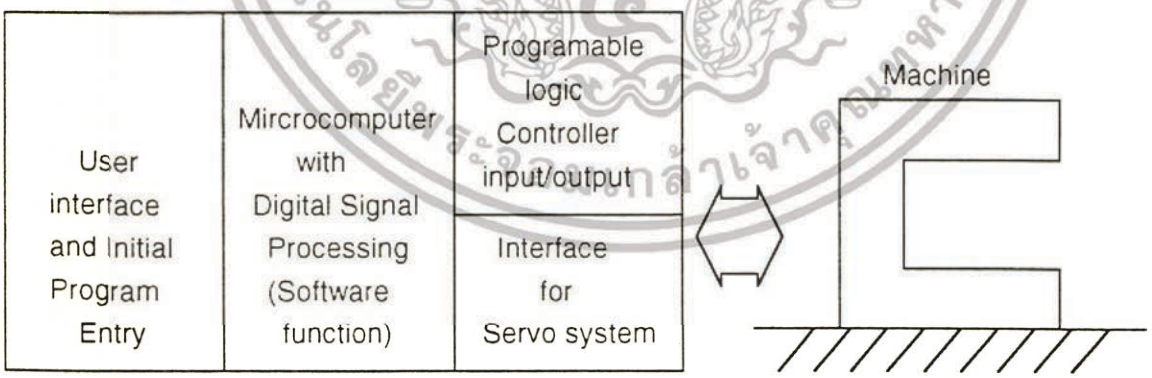
## 1. บทนำ

ความก้าวหน้าทางคอมพิวเตอร์ต่อการพัฒนาอุตสาหกรรม มีบทบาทมากเริ่มตั้งแต่การออกแบบชิ้นส่วนการวิเคราะห์และการควบคุมการผลิตชิ้นส่วน (Computer Aided Design/ Computer Aided Manufacturing : CAD/CAM) ซึ่งทำให้การผลิตมีประสิทธิภาพมากและสามารถลดต้นทุนการผลิตให้ความเร็วในการผลิต หัวใจสำคัญส่วนหนึ่งที่ทำให้การผลิตมีประสิทธิภาพ ได้แก่ ส่วนควบคุมเครื่องจักร (Computer Numerical Control : CNC) เพราะการผลิตจะถูกต้องเสมอในชิ้นงานและความเร็วในการทำงาน ซึ่งส่วนดังกล่าวจะรับข้อมูลจากโปรแกรม CAD/CAM แล้วกระทำตามคำสั่งทางคอมพิวเตอร์ควบคุมเครื่องจักร

ในระบบซีเอ็นซีมีองค์ประกอบสำคัญด้วยกันทั้งหมด 5 ส่วนดังนี้

1. User Interface and Initial Program Entry
2. Microcomputer with Digital Signal Processing (Software function)
3. Programable logic Controller Input/Output
4. Interface for Servo system
5. Machine

สามารถแสดงส่วนประกอบต่างๆ ได้ดังรูปที่ 1




รูปที่ 1.1 ส่วนประกอบต่างๆ ของระบบเครื่องกัดแนวดิ่งซีเอ็นซี

ในการวิจัยและพัฒนาเครื่องควบคุมดังกล่าว จะต้องพัฒนาในส่วนของโปรแกรม User Interface, Digital Signal Processing (Software function), Programable logic Controller input/output และ Interface สำหรับ servo system ซึ่งต้องอาศัย Digital Signal Processing (DSP)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 1.1 สถานภาพทางเทคโนโลยีต่างประเทศ

การพัฒนา Numerical Control ได้รับความสำเร็จอย่างมากจากกองทัพอากาศของสหรัฐอเมริกา ซึ่งมีความต้องการเพิ่มประสิทธิภาพวิธีการผลิตเครื่องบินแบบใหม่ ภายหลังจากสงครามโลกครั้งที่ 2 ชิ้นส่วนต่างๆ ที่เคยใช้ประดิษฐ์เครื่องบินไอพ่นมีความซับซ้อนมาก และต้องการการปรับแต่งมากขึ้น การปรับแต่งด้วยเครื่องจักรที่ใช้ส่วนใหญ่เป็นการกัดโลหะ

ดังนั้น กองทัพอากาศจึงให้ทุนอุดหนุนแก่ Massachusetts Institute of Technology (MIT) ในโครงการวิจัยเพื่อพัฒนาให้ได้เครื่องกัด NC ต้นแบบเครื่องต้นแบบนี้ถูกผลิตขึ้นโดยการประกอบกลไกต้นกำลังระบบ Numerical Control แบบ 3 แกน เข้ากับเครื่องกัดรุ่นเดิมในเดือนมีนาคม ปี ค.ศ. 1952 ห้องปฏิบัติการของ MIT ได้ทำการแสดงเครื่องจักร NC เป็น ครั้งแรกขึ้น ต่อมาผู้ผลิตเครื่องมือเครื่องจักรได้ค่อยๆ เริ่มพัฒนาโครงการของตนเองเพื่อที่จะแนะนำหน่วยงาน NC เพื่อให้พอเพียงต่อความต้องการในการผลิตลำตัวเครื่องบิน กองทัพอากาศได้พยายามพัฒนาระบบ NC ต่อไปโดยการให้ทุนอุดหนุนการวิจัยเพิ่มเติมแก่ MIT ให้ออกแบบภาษาโปรแกรมชิ้นงาน ซึ่งสามารถที่จะควบคุมเครื่องจักร NC และผลลัพธ์ของงานวิจัยครั้งนี้คือ ได้ภาษา APT ซึ่งใช้ในการควบคุมเครื่องจักรอัตโนมัติต่างๆ ต่อมาเป้าหมายของการวิจัย APT นี้ก็เพื่อที่จะช่วยทำให้ผู้จัดโปรแกรมชิ้นงานสามารถที่จะติดต่อสื่อความกับเครื่องมือ เครื่องจักรในรูปประโยคภาษาอังกฤษที่ง่ายๆ แม้ว่าภาษา APT จะทำให้สิ้นเปลืองหน่วยความจำในคอมพิวเตอร์ แต่ก็เป็นผลแห่งความสำเร็จที่ยิ่งใหญ่อันหนึ่ง ทุกวันนี้ภาษา APT ยังคงใช้กันอยู่อย่างกว้างขวาง ในวงการอุตสาหกรรม และภาษาการควบคุมสมัยใหม่ก็ยังมีพื้นฐานมาจากความคิดของ APT (Auto Matically Programmed Tool)

ในปัจจุบัน เทคโนโลยีในต่างประเทศก็ได้พยายามเปลี่ยนแปลงโครงสร้างของส่วนควบคุมเครื่องกัดแนวดิ่งซีเอ็นซีให้มีความสามารถมากขึ้นเช่น สามารถคำนวณคำสั่งล่วงหน้าก่อนการกัดงานจริง สามารถทำฟังก์ชันของการกัดชิ้นงานในรูปแบบต่างๆ ได้มากขึ้น และสามารถรับข้อมูลในระบบ CAD ในขณะเดียวกันโครงสร้างในส่วนควบคุมก็มีขนาดเล็กลง แต่สามารถควบคุมการเคลื่อนที่หลายๆ แกนพร้อมกันได้ ตลอดจนได้พัฒนาอยู่บนพื้นฐานไมโครคอมพิวเตอร์ที่มีจำหน่ายทั่วโลก

## 1.2 สถานภาพทางเทคโนโลยีภายในประเทศ

เครื่องควบคุมเครื่องกัดแนวดิ่งซีเอ็นซีในประเทศไทยเริ่มต้น โดยการนำเข้าเครื่องเก่าจากต่างประเทศซึ่งมีราคาค่อนข้างต่ำเพื่อการเริ่มต้นศึกษาเรียนรู้ถึงความสามารถ เครื่องส่วนใหญ่ที่นำเข้าจะเป็นเครื่องที่ใช้ NC ในการควบคุมจึงมีขนาดใหญ่ในส่วนของคอมพิวเตอร์ในการควบคุม และในเวลาต่อมาก็เริ่มนำเข้าเครื่องจักรที่ควบคุมด้วย CNC จนถึงปัจจุบัน ในเวลาปัจจุบันเครื่องที่นำเข้าสมัยก่อนที่ควบคุมด้วย NC ก็เริ่มประสบปัญหาคือ ใช้งานไม่ได้ในส่วนของคอมพิวเตอร์หรือส่วนของอิเล็กทรอนิกส์ในการควบคุมเครื่องจักร ดังนั้นจึงมีอาจารย์และนักวิจัยหลายกลุ่มหลายสถาบันเริ่มให้ความสนใจในการวิจัยและพัฒนาส่วนควบคุมเครื่องจักรดังกล่าว ซึ่งมีทั้งควบคุมเครื่องจักรที่เรียกว่า "เครื่องกลึง" (Lath machine), เครื่องกัดแนวดิ่ง (Milling หรือ Machining Center Machine) หรือจะเป็นเครื่องจักรประเภทอื่นที่ควบคุม การเคลื่อนที่ โดยคอมพิวเตอร์ ตัวอย่างเช่น สถาบันเทคโนโลยีพระจอมเกล้าธนบุรีได้พัฒนาส่วนควบคุมเครื่องจักร Milling

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยการควบคุมการเคลื่อนที่ 3 แกน คือ แกน X,Y และ Z โดยใช้คอมพิวเตอร์ควบคุม Steping motor และที่มหาวิทยาลัยเชียงใหม่ ก็สร้างส่วนควบคุม เครื่องจักร Lath ก็ยังควบคุม Steping motor เหมือนกัน และสถาบันเทคโนโลยีพระจอมเกล้าพระนครเหนือกำลังสร้างตัวเครื่องจักร Milling แต่อาศัยระบบคอมพิวเตอร์ควบคุมจากต่างประเทศในการวิจัยและพัฒนาดังกล่าวได้ดำเนินการเป็นผลสำเร็จในขั้นหนึ่ง แต่ยังไม่สามารถนำไปควบคุมเครื่องจักรขนาดใหญ่ในอุตสาหกรรมได้ต้องมีการวิจัยและพัฒนาต่อไป

## 1.3 สถานภาพทางเทคโนโลยีของหน่วยงานหรือโครงการวิจัย

สถานภาพทางเทคโนโลยีของโครงการวิจัยเริ่มต้น โดยในสมัยก่อนเริ่มโครงการวิจัยได้ทำการแก้ไขดัดแปลงเครื่องจักรเก่าที่ควบคุมด้วย NC แล้วในส่วนของ NC ใช้การไม่ได้ จึงได้ดัดแปลงนำระบบ CNC และส่วนอิเล็กทรอนิกส์เปลี่ยนใส่เข้าไปแล้วก็แก้ไขตัวแปรต่างๆ จนประสบความสำเร็จด้วยดี ซึ่งในการดำเนินการในระยะแรกได้อาศัยนักวิชาการและวิศวกรประมาณ 4-5 คน ในการดัดแปลงดังกล่าว ในระยะหลังต่อมานักวิจัยเองได้ทำการแก้ไขดัดแปลง โดยอาศัยนักวิชาการและวิศวกรบ้างบางครั้ง ปัญหาที่ตามมาในภายหลังคือ คอมพิวเตอร์ในการควบคุมดังกล่าว ต้องใช้เวลาในการนำเข้าหรือในการจัดหาและถ้ามีปัญหาของระบบก็เสียเวลาในการบำรุงรักษานาน ดังนั้นจากเหตุผลดังกล่าวซึ่งนำไปสู่การวิจัยและพัฒนาโครงการดังกล่าว โดยการพัฒนาโปรแกรมและส่วนอิเล็กทรอนิกส์ในการควบคุม ในส่วน Servo Drive System และมอเตอร์นำเข้าจากต่างประเทศ และในส่วนเครื่องจักรกล Milling ได้รับการสนับสนุนทางเอกชนด้วยส่วนหนึ่งเพื่อจะดำเนินการให้บรรลุถึงระดับเทคโนโลยีที่ต้องการในเชิงอุตสาหกรรมต่อไป

สถานภาพของโครงการวิจัยดังกล่าว ได้ใช้เทคโนโลยีการควบคุมเครื่องกัดแนวดิ่งซีเอ็นซี (CNC Milling Machine) ซึ่งต้องขับเคลื่อน 3 แกน โดยการควบคุมการขับเคลื่อนของระบบ Servo motor อาศัยไมโครคอมพิวเตอร์เป็นพื้นฐานซึ่งอาจจะรวมเอาส่วนในการคำนวณที่เรียกว่า "DSP" เพื่อให้คำนวณรวดเร็ว 3 แกนพร้อมๆ กัน

ในการควบคุม 3 แกนให้ทำงานพร้อมกันในระยะเริ่มต้นได้นำเทคโนโลยีดังกล่าวจากต่างประเทศเข้ามาแล้วศึกษาความเป็นไปได้ในการออกแบบและสร้างต้นแบบ ในการวิเคราะห์ได้ศึกษาถึงในส่วนของโปรแกรมในการควบคุม โดยได้ทำการเขียนโปรแกรมการควบคุมเครื่องจักร 3 แกนขึ้นโดยอาศัยภาษาคอมพิวเตอร์ที่เรียกว่า ภาษา C ร่วมกับภาษาสำหรับควบคุมการเคลื่อนที่ของอุปกรณ์ส่วนควบคุม (Controller Card) หลังจากนั้นได้ทำการทดลองการทำงานจริงสามารถทำงานตามคำสั่งได้ปกติ

ในส่วนทางด้านอุปกรณ์ฮาร์ดแวร์ได้ศึกษาทฤษฎีและหลักการในการควบคุม โดยอาศัย Digital Signal Processing เป็นหลัก และในส่วนการควบคุมอินพุท, เอาท์พุท (Input/Output) ที่เรียกว่า Programable logic controller (PLC) ในส่วนดังกล่าวยังอยู่ในช่วงของการทดลองและออกแบบลายทองแดงเพื่อการสร้างต้นแบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 1.4 จุดมุ่งหมายของโครงการวิจัยและพัฒนา

จุดมุ่งหมายของโครงการวิจัยและพัฒนา แบ่งออกเป็นหัวข้อได้ดังต่อไปนี้

1. พัฒนาและสร้างอุปกรณ์ควบคุมเครื่องกัดแนวดิ่ง ซีเอ็นซี เป็นเครื่องต้นแบบ
2. ออกแบบและสร้างเครื่องกัดแนวดิ่ง (Milling Machine) สำหรับทดสอบอุปกรณ์ควบคุม
3. เพื่อเพิ่มศักยภาพในการวิจัยและพัฒนาความรู้ ทางวิศวกรรมเครื่องกล อิเล็กทรอนิกส์และคอมพิวเตอร์
4. เพื่อกระจายความรู้เทคโนโลยีระดับสูงไปยังวงการอุตสาหกรรมภายในประเทศ

## 1.5 วิธีการและแผนการดำเนินงาน

วิธีการและแผนการดำเนินงาน สามารถแบ่งการดำเนินงานตลอดโครงการได้ดังนี้

1. ออกแบบวงจรอิเล็กทรอนิกส์ เพื่อควบคุมเครื่องจักรที่เหมาะสมกับด้านเครื่องกล
2. จัดหาทางด้านเครื่องจักรทดสอบ และจัดหาวัสดุประกอบเป็นแผนวงจรควบคุม
3. ออกแบบวงจรด้านการรับอินพุทสัญญาณดิจิตอล จากการเคลื่อนที่ของโต๊ะแกน X,Y และ Z
4. ออกแบบวงจรด้านการขับเคลื่อนมอเตอร์ แกน X,Y และ Z
5. ออกแบบวงจรส่วนการสั่งงานดิจิตอลอินพุท และเอาท์พุท (Programmable-logic Controller)
6. ออกแบบแผ่นลายวงจรต่างๆ
7. สร้างและประกอบวงจรต่างๆ ที่ได้ออกแบบเพื่อเป็นต้นแบบ
8. อุปกรณ์ควบคุมเมื่อประกอบเรียบร้อยแล้ว นำทดสอบติดตั้งกับเครื่องจักรที่ทดสอบ
9. ทดสอบการทำงาน แต่ละขั้นตอนทางด้านอิเล็กทรอนิกส์ และการควบคุมการทำงานฟังก์ชั่นต่างๆ
10. ทดสอบการกัดชิ้นงานต่างๆ เพื่อปรับแต่งทางด้านตัวแปรต่างๆ
11. ตรวจสอบอุปกรณ์ควบคุมและเครื่องจักรรวมทั้งชิ้นงานเพื่อปรับแต่งครั้งสุดท้าย

## 1.6 เนื้อหาที่นำเสนอในรายงานการวิจัย

จุดมุ่งหมายของโครงการวิจัยและพัฒนาคือ การสร้างต้นแบบส่วนควบคุมเครื่องกัดแนวดิ่งซีเอ็นซี (Computerized Numerical Control for Vertical Milling Machine) โดยรวมถึงเทคโนโลยีต่างๆ ทางด้าน Control Algorithm, คอมพิวเตอร์, Digital Signal Processing และทางด้านเครื่องจักรกล เนื้อหาของรายงานมีรายละเอียดดังนี้

บทที่ 1 กล่าวถึงบทนำและสถานภาพทางเทคโนโลยีต่างประเทศ, ภายในประเทศ และหน่วยงานหรือโครงการวิจัย

บทที่ 2 ทฤษฎีในส่วนการควบคุมมอเตอร์ การออกแบบและสร้างในส่วนฮาร์ดแวร์ DSP

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3 การออกแบบและสร้างเครื่องกัดแนวดิ่งซีเอ็นซี

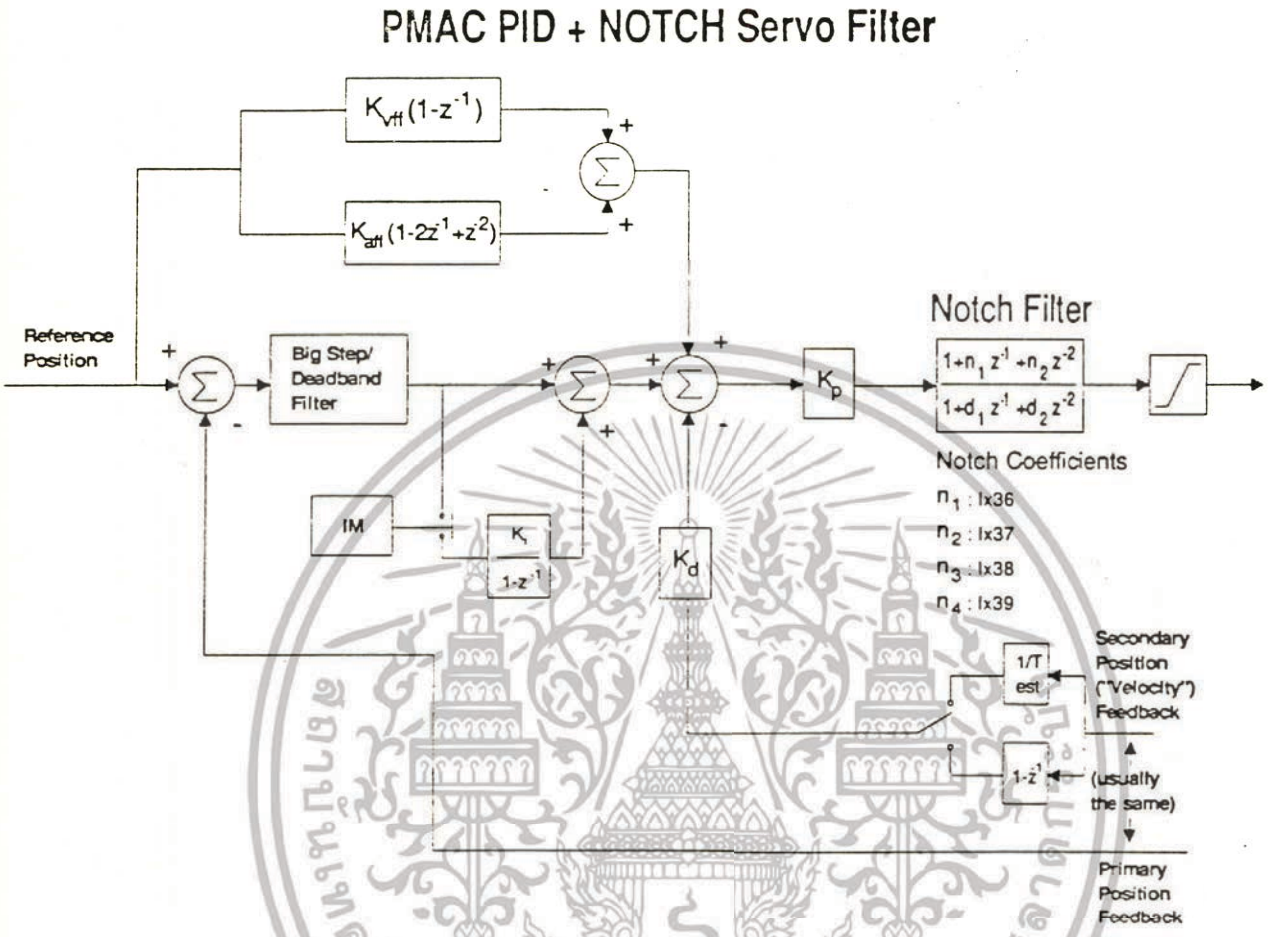
บทที่ 4 การออกแบบและพัฒนาโปรแกรมเครื่องกัดแนวดิ่งซีเอ็นซี

บทที่ 5 สรุปและข้อเสนอแนะ

- เอกสารอ้างอิง
- ภาคผนวก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในรายละเอียดของการคำนวณในส่วนคอมพิวเตอร์ จะพิจารณาโครงสร้างของตัวควบคุมแบบ PID สามารถแสดงโครงสร้างรายละเอียดดังรูปที่ 2.2




รูปที่ 2.2 Block Diagram ในรายละเอียดของส่วนควบคุมแบบ PID

| | | |
|---|---|---|
| Kp | : | Proportional gain |
| Kd | : | Derivative gain |
| Kvff | : | Velocity feedforward gain |
| Ki | : | Integral gain |
| Im | : | Integration mode |
| Kaff | : | Acceleration feedforward gain |

จากสมการของตัวควบคุมแบบ PID ซึ่งอยู่ในรูปของ integro - differential คือ

$$m(t) = Kpe(t) + Ki \int e(t)dt + K_D \, de(t)/dt$$

ในทางปฏิบัติสมการจริงที่ใช้ในการคำนวณของอัลกอริทึม PID เพื่อคำนวณค่าของ Output สำหรับการควบคุมมอเตอร์จะเป็นดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

DACout (n) = Pgain * internal position scaling term for motor * [FE(n) + (-Dgain* AV(n) + velocity feed forward gain *CV(n) + acceleration feed forward gain * CA*(n))/64 + Igain *IE(n)/2 23]/219

โดยที่

DACout (n) = คำสั่ง Output ใน servo cycle n ซึ่งจะเปลี่ยนเป็นสัญญาณ Output ที่เป็น Analog - 10 โวลท์ดีซีถึง +10 โวลท์ดีซี เพื่อส่งต่อไปยังระบบขับเคลื่อน Drive มอเตอร์

FE(n) = Following error หรือค่าผิดพลาดใน Servo cycle n คือ ค่าความแตกต่างระหว่างตำแหน่งที่ต้องการและตำแหน่งจริง สำหรับ cycle n [ตำแหน่งที่ต้องการ (CP(n)) - ตำแหน่งจริง (AP(n))]

AV(n) = actual velocity ใน servo cycle n คือ ความแตกต่างระหว่าง ตำแหน่งจริงสุดท้าย 2 ตำแหน่ง [ตำแหน่งจริงสุดท้าย (AP(n))- ตำแหน่งจริงก่อนสุดท้าย (AP(n-1))]

CV(n) = Command velocity หรือคำสั่งความเร็วใน servo cycle n คือ ความแตกต่างของตำแหน่งที่ต้องการสุดท้าย 2 ตำแหน่ง [ตำแหน่งสุดท้ายที่ต้องการ (CP(n)) - ตำแหน่งสุดท้ายที่ ต้องการก่อนหน้า (CP(n-1))]

CA(n) = Command acceleration หรือคำสั่งความเร่งใน servo cycle n คือ ความแตกต่างระหว่างคำสั่งความเร็วสุดท้าย 2 ตำแหน่ง [คำสั่งความเร็วที่ตำแหน่งสุดท้าย (CV(n)) - คำสั่งความเร็ว ที่ตำแหน่งก่อนหน้า (CV(n-1))]

IE(n) = อินทิเกรทความผิดพลาดใน servo cycle n คือ

ในส่วนของ Notch filter คือ anti-resonance (band-reject) เพื่อช่วยในการแก้ปัญหาด้าน Physical resonance

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 2.2 การออกแบบฮาร์ดแวร์ DSP

### 2.2.1 หลักการเบื้องต้น DSP

Digital Signal Processing เป็นการเปลี่ยนรูปแบบของสัญญาณโดยใช้การคำนวณทางคณิตศาสตร์กับสัญญาณเชิงเลข (Digital) ของสัญญาณอินพุท (Input) ด้วยดิจิตอลซิกแนลโปรเซสเซอร์ ดังนั้นจึงทำให้เราสามารถได้สัญญาณเอาท์พุท (Output) ของวงจรใดๆ ได้เพียงแค่เรานำสัญญาณอินพุท (Input) ผ่านการ Process ด้วยสมการของวงจรนั้นๆ เท่านั้น

การที่ Digital Signal Processing ได้รับความสนใจมากขึ้น เนื่องมาจากเทคโนโลยีในการสร้างวงจรรวม (Integrated Circuit) มีประสิทธิภาพดีขึ้นมากวงจรขนาดใหญ่สามารถนำมาสร้างเป็นวงจรรวมที่ผลิตบนสารกึ่งตัวนำชิ้นเดียวกันได้ และผลิตได้ครั้งละจำนวนมากๆ จึงทำให้ราคาถูกลงเป็นอย่างมาก นอกจากนี้การประมวลเชิงเลข (Digital Signal Processing) ยังให้ความแม่นยำ (Accuracy) และความเชื่อถือ (Reliablility) มากกว่าการประมวล สัญญาณเชิงอุปมานมาก โดยทั่วไประบบ DSP จะประกอบด้วยส่วนต่างๆ ดังรูป

รูปที่ 2.3 ระบบ Digital Signal Processing

LOWPASS FILTER มีไว้เพื่อจำกัดช่วงความถี่สูงของสัญญาณอินพุทจากนั้นสัญญาณอินพุทจะถูกแปลงจาก Analog ให้เป็น Digital โดยวงจร A/D (Analog to Digital Converter) จากนั้นระบบประมวลผลสัญญาณดิจิตอลจะนำอัลกอริทึม (Algorithm) ที่มีอยู่มากระทำกับข้อมูลสัญญาณแล้วทำการส่งข้อมูลสัญญาณที่ถูกเปลี่ยนแปลงแล้วสู่ส่วน D/A (Digital to Analog Converter) เพื่อเปลี่ยนสัญญาณข้อมูลซึ่งเป็น Digital ให้เป็นสัญญาณ Analog ดามเดิม

ในปัจจุบันมีการนำ DSP ไปประยุกต์ในงานต่างๆ มากมายซึ่งสามารถจำแนกออกเป็นประเภทต่างๆ ดังนี้

1) การประยุกต์ในงานทั่วๆ ไป ได้แก่ Digital filter, Adaptive filtering, Wareform generation
2) งานด้านกราฟฟิค (Graphic/Image) ได้แก่ งานด้านการส่งภาพ (image transimission), การเข้ารหัสภาพ (Image coding หรือ Compression), การจำรูปแบบ (Pattern recognition)
3) งานด้านการวัดและเครื่องมือวัด ได้แก่ การวิเคราะห์สเปคตรัม (Spectrum analysis), เครื่องกำเนิดสัญญาณรูปแบบต่างๆ (Function generation), การวิเคราะห์แผ่นดินไหว (Seismic Processing)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4) การประยุกต์ทางด้านเสียงพูด (Voice/Speech) ได้แก่ การจำ (recognition), การยืนยัน (Verification), การเพิ่มคุณภาพ (Enhancement), การสังเคราะห์ (Synthesis) รวมถึงการเปลี่ยนแปลง ข้อความ (Text) เป็นเสียงพูด (Speech)

5) ทางด้านระบบควบคุม (Control) ได้แก่ ระบบควบคุมหุ่นยนต์ (Robotic), การควบคุมเครื่องจักรกลไฟฟ้าต่างๆ ระบบเซอร์โว หรือพวกเฟสล็อคลูพ (Phase Lock Loop (PLL))

6) ทางด้านทหาร (Millitary) ได้แก่ การสื่อสารที่เกี่ยวกับความมั่นคง (Secure communication), ระบบเรโซนา, ระบบนำร่อง (Navigation), ระบบนำวิถี (Missile guidance)

7) ทางระบบโทรคมนาคม (Telecommunication) ได้แก่ การทำจัดเสียงสะท้อน (Echo cancellation), ระบบสื่อสารดิจิตอลโมเด็ม (Modem), โทรศัพท์เคลื่อนที่ (Cellular telephones)

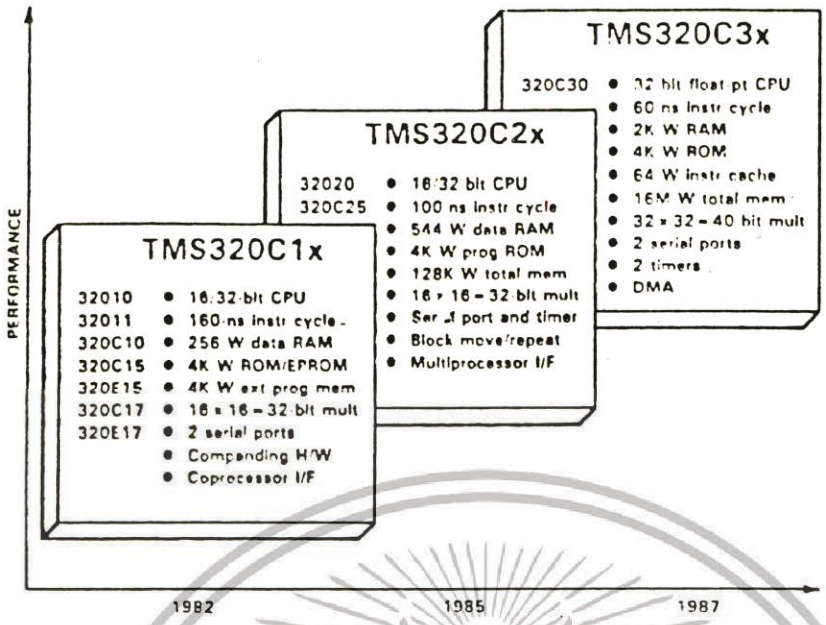
8) ทางด้านการแพทย์ (Medical) ได้แก่ เครื่องช่วยการได้ยิน เครื่องมือวินิจฉัยต่างๆ

9) ทางด้านอุตสาหกรรม (Industrial) ได้แก่ เครื่องควบคุมเครื่องยนต์ (Engine control), ระบบเสียงหรือโทรทัศน์ดิจิตอล (Digital audio/TV), เครื่องมือสังเคราะห์เสียงดนตรี (Music Synthesizer) เครื่องเจาะหรือตัดแบบควบคุมด้วยตัวเลข (Numerical Control)

ดิจิตอลซิกแนลโปรเซสเซอร์ จำเป็นจะต้องมีความเร็วสูงที่ถูกออกแบบมาโดยเฉพาะให้สามารถทำงานได้อย่างมีประสิทธิภาพโดยใช้ข้อดีของการประมวลผลแบบขนาน และการใช้ชุดคำสั่งที่สร้างขึ้นมาเฉพาะทั้งนี้ก็เพื่อการประมวลแบบเวลาจริง (Real Time) ซึ่งหมายถึงเมื่อเราใส่สัญญาณอินพุทเข้าไปก็จะได้สัญญาณเอาท์พุทออกมาเลย โดยจะมีเวลาหน่วง (delay time) น้อยมาก ซึ่งจะมีประโยชน์ในการใช้งานมากกว่าการประมวลผลแบบไม่ใช้เวลาจริง

บริษัทเท็กซัสอินสตรูเมนต์ ได้ผลิตไอซีที่เป็นดิจิตอลซิกแนลโปรเซสเซอร์ซึ่งเป็นตระกูลที่มีชื่อว่า TMS320 ซึ่งมีวิวัฒนาการดังรูป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้




รูปที่ 2.4 วิวัฒนาการของไอซีตระกูล TMS320C25

ซึ่งในงานรายงานการวิจัยฉบับสมบูรณ์นี้เราจะใช้ TMS320C25 มาสร้างเป็นระบบ DSP เพื่อประยุกต์ใช้ในการควบคุมมอเตอร์เพื่อสร้าง CNC 3 มิติ (3 Dimensional Computerized Numerical Control)

### 2.2.2 การออกแบบวงจรแต่ละส่วน

จากรูปที่ 3.1 ประกอบไปด้วยวงจรส่วนต่างๆ เหล่านี้

1. วงจรอินเทอร์เฟส (Interface circuit)
2. วงจรแอดเดรสซิเล็กเตอร์ (Address selector circuit)
3. วงจรดาต้าซีเล็กเตอร์ (Data selector circuit)
4. วงจรหน่วยความจำ (Memory circuit)
5. วงจร TMS320C25 และสัญญาณควบคุม (TMS and Control-Circuit)
6. วงจร D/A และ A/D (Digital to Analog and Analog to Digital Circuit)

วงจรต่างๆ เหล่านี้จะทำงานร่วมกันอย่างสอดคล้องโดยมีสัญญาณควบคุมคอยควบคุมให้การทำงานของแต่ละวงจรทำงานร่วมกันอย่างถูกต้อง

วงจรอินเทอร์เฟส จะทำหน้าที่เป็นตัวอินเทอร์เฟสระหว่างวงจรของ TMS กับ PC ซึ่ง PC จะส่งข้อมูลมาเก็บไว้ในหน่วยความจำภายนอกของ TMS โดยจะส่งข้อมูลที่ต้องการจะส่งผ่านวงจรอินเทอร์เฟสนี้ไปเก็บไว้ที่วงจรหน่วยความจำ

วงจรแอดเดรสซีเล็กเตอร์ จะทำหน้าที่เลือกต่อแอดเดรสบัสของวงจรหน่วยความจำเข้ากับแอดเดรสบัสของ PC หรือแอดเดรสบัสของ TMS ซึ่งขึ้นอยู่กับสัญญาณควบคุมหรืออีกนัย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง



หนึ่งก็คือ วงจรแอดเดรสซีเล็กเตอร์เป็นตัวกำหนดให้ PC หรือ TMS เข้าถึงข้อมูลภายในหน่วยความจำ

วงจรหน่วยความจำ (External Memory of TMS320C25) วงจรนี้จะเป็นวงจรหน่วยความจำภายนอกของ TMS320C25 วงจรหน่วยความจำนี้ประกอบไปด้วย Static RAM เบอร์ SC2568 จำนวน 8 ตัว ซึ่งรวมแล้วมีขนาดทั้งหมด 128 KWord โดยจะแบ่งเป็นหน่วยความจำโปรแกรม 64 KWord

วงจร TMS320C25 และสัญญาณควบคุมจะทำงานโดย TMS จะไปอ่านโปรแกรมการทำงานมาจากหน่วยความจำโปรแกรมในวงจรหน่วยความจำ และดาต้าจากหน่วยความจำข้อมูลในวงจรหน่วยความจำเช่นกัน

เราได้แบ่งวงจรออกเป็น 2 ส่วน โดยส่วนแรกจะเป็นวงจรส่วนอินเทอร์เฟส วงจรแอดเดรสซีเล็กเตอร์ วงจรดาต้าซีเล็กเตอร์ และวงจรหน่วยความจำ

**วงจรอินเทอร์เฟส**

การที่เราให้ PC ติดต่อกับ TMS โดยผ่านทาง I/O Port นั้นก็เพื่อให้การส่งผ่านข้อมูลจาก PC ไป TMS มีความคล่องตัว รวดเร็วและมีเสถียรภาพ พอร์ทของ PC ที่เตรียมไว้มีทั้งแบบ XT และ AT ซึ่งแบบ XT นั้นจะเป็น Slot 8 บิทอยู่บริเวณด้านหลังของเครื่องคอมพิวเตอร์ และในบริเวณเดียวกันนั้นก็จะมี Slot 16 บิท สำหรับพอร์ท AT อยู่ด้วย สัญญาณต่างๆที่อยู่ใน XT Bus ประกอบไปด้วยแอดเดรสบัสจำนวน 16 เส้น (A0-A15) ดาต้าบัสจำนวน 8 เส้น (D0-D7) และสัญญาณควบคุมที่จำเป็นต่างๆ สำหรับสัญญาณของ AT Bus จะเหมือนกับ XT Bus ทุกประการ แต่จะมีส่วนที่เพิ่มมาอีกคือ ดาต้าบัสจำนวน 8 เส้น (D8-D15) และสัญญาณควบคุมที่เป็นของ AT โดยเฉพาะอีกต่างหาก ซึ่งรายละเอียดของสัญญาณและตำแหน่งของสัญญาณบน Slot สามารถค้นคว้าได้จากในหนังสือคอมพิวเตอร์ฮาร์ดแวร์ทั่วไป

วงจรนี้ใช้การอินเทอร์เฟสแบบ AT และใช้การเคลื่อนย้ายข้อมูลทีละ 16 บิท จาก PC ไปยังวงจรหน่วยความจำซึ่งช่วยในการประหยัดเวลา การเคลื่อนย้ายข้อมูลมากกว่าใช้การอินเทอร์เฟสแบบ XT

ภายในวงจรอินเทอร์เฟสประกอบไปด้วย 74LS138 จำนวน 2 ตัว DIP Switch จำนวน 2 ตัว ใช้สำหรับดีโค้ดแอดเดรสของพอร์ทและ 8255 จำนวน 2 ตัว ซึ่งจะทำหน้าที่เป็นพอร์ทแบบขนานให้กับ ข้อมูลและแอดเดรสของข้อมูลดังรูปที่ 4.2

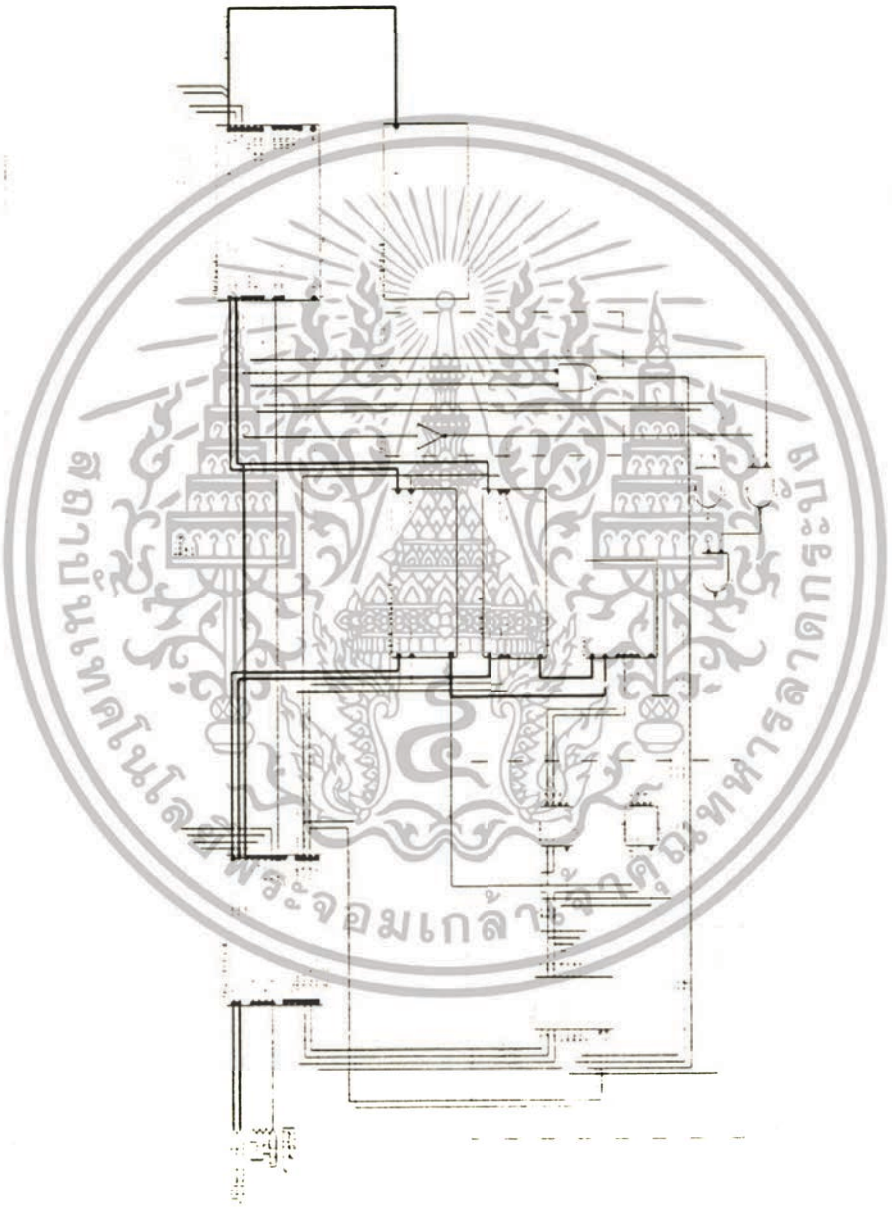
จากรูปที่ 3.2 74LS138 ทั้ง 2 ตัว จะทำหน้าที่ดีโค้ดแอดเดรสของพอร์ทร่วมกับ DIP Switch จำนวน 2 ตัว ซึ่งจะดีโค้ดสัญญาณแอดเดรสที่ส่งมาจาก PC ร่วมกับสัญญาณ AEN (Address Enable) สาเหตุที่ต้องนำสัญญาณ AEN มาดีโค้ดด้วยเพื่อป้องกันไม่ให้วงจรนี้ทำงานขณะที่ PC มีการทำ DMA อยู่ เพราะว่าขณะที่มีการทำ DMA PC จะปล่อยทั้งสัญญาณแอดเดรสลงมาที่แอดเดรสบัสพร้อมกับสัญญาณ IOR หรือ IOW ซึ่งบางครั้งค่าแอดเดรสที่ปล่อยลงม่าอาจจะมีค่ามาตรงกับหมายเลขพอร์ทของวงจรนี้ ซึ่งวงจรนี้จะทำงานอาจจะทำให้เกิดความผิดพลาดของขบวนการ DMA ที่กำลังทำงานอยู่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

26341

ข้อมูลที่ส่งมาจาก PC ทางดาต้าบัสจะประกอบไปด้วย

1. ข้อมูลหรือโปรแกรมที่ต้องการส่งให้ TMS
2. แอดเดรสของข้อมูลนั้นบนวงจรหน่วยความจำ
3. สัญญาณควบคุมการเข้าสู่ข้อมูล



รูปที่ 2.5 Block Diagram ของวงจรทั้งหมด

ดังนั้นข้อมูลที่อยู่ในดาต้าบัสของ PC ถ้าเป็นข้อมูลของ TMS เราจะให้ส่งผ่านทางพอร์ท A ของ 8255 ถ้าเป็นแอดเดรสของข้อมูล เราจะผ่านพอร์ท B ของ 8255 แต่ถ้าเป็นสัญญาณควบคุมการเขียน การอ่าน อินเทอรัพท์ หรือรีเซ็ทพอร์ทจะส่งผ่าน gate

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**วงจรแอดเดรสซีเล็กเตอร์**

วงจรแอดเดรสซีเล็กเตอร์ประกอบด้วย ไอซีเบอร์ 74AS245 จำนวน 4 ตัว สาเหตุที่ต้องใช้ไอซีชนิด AS เพราะว่าต้องการลด Delay time ในตัวให้มีค่าน้อยที่สุดเพราะ TMS320C25 ทำงานด้วยความเร็วสูง ดังนั้นถ้าอุปกรณ์รอบข้างทำงานช้ากว่าตัว TMS320C25 มากทำให้ต้องสร้างสถานะรอ (Wait state)

สัญญาณควบคุมที่คอยควบคุมการทำงานของวงจรแอดเดรสซีเล็กเตอร์นี้ ประกอบด้วย

- IBMEN เมื่อสัญญาณนี้แอคทีฟจะทำให้ 74AS245 ตัวที่ 1 และตัวที่ 3 ทำงานซึ่งเป็นผลให้เกิดการเชื่อมต่อแอดเดรสบัสของ PC เข้ากับแอดเดรสบัสของวงจรหน่วยความจำ
- TMSEN เมื่อสัญญาณนี้แอคทีฟจะทำให้ 74AS245 ตัวที่ 2 และตัวที่ 4 ทำงานซึ่งเป็นผลให้เกิดการเชื่อมต่อแอดเดรสบัสของ TMS เข้ากับแอดเดรสบัสของวงจรหน่วยความจำ และเนื่องจากสัญญาณแอดเดรสจะถูกส่งจากวงจร PC หรือวงจร TMS ไปยังวงจรหน่วยความจำเพียงทิศทางเดียวเท่านั้นจึงต่อขา Direction ของ 74AS245 ทุกตัวลงกราวด์

รูปที่ 2.6 วงจร Address Selector

**วงจรดาต้าซีเล็กเตอร์**

วงจรดาต้าซีเล็กเตอร์ประกอบไปด้วย ไอซีเบอร์ 74AS245 จำนวน 4 ตัวเช่นกัน ซึ่งการทำงานของวงจรคล้ายกับวงจรแอดเดรสซีเล็กเตอร์ ต่างกันตรงที่ข้อมูลในดาต้าบัสสามารถถูกเคลื่อนย้ายไปได้ทั้งสองทิศทาง คือ ทั้งจาก PC ไปยังวงจรหน่วยความจำและจากวงจรหน่วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

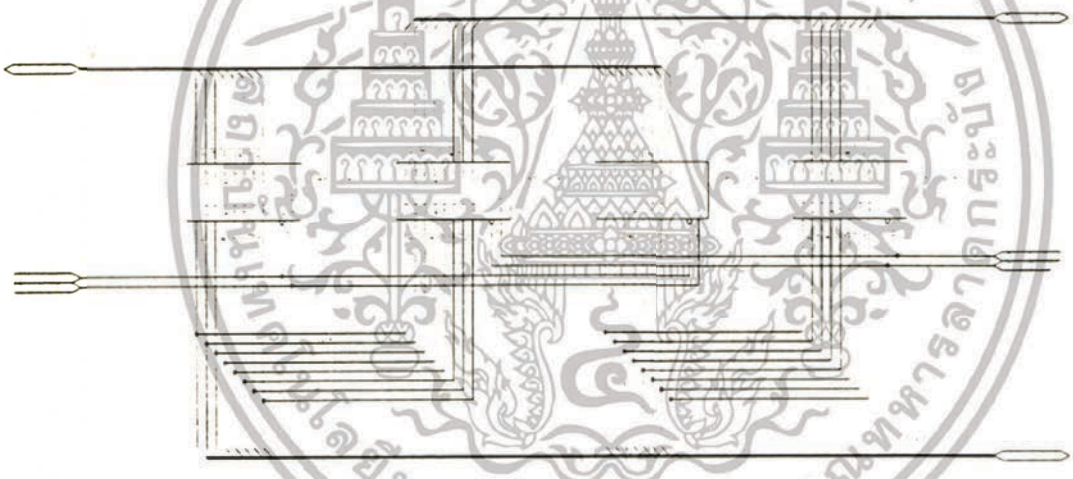
ความจำไปยัง PC ดังนั้นจึงต้องมีสัญญาณควบคุมเพิ่มขึ้นมากอีกสองสัญญาณคือ IBMDIR และ TMSDIR

- IBMEN เมื่อสัญญาณนี้แอคทีฟจะทำให้ 74AS245 ตัวที่ 1 และตัวที่ 3 ทำงานซึ่งเป็นผลให้เกิดการเชื่อมต่อดาต้าบัสของ PC เข้ากับดาต้าบัสของวงจรหน่วยความจำ

- TMSEN เมื่อสัญญาณนี้แอคทีฟจะทำให้ 74AS245 ตัวที่ 2 และตัวที่ 4 ทำงานซึ่งเป็นผลให้เกิดการเชื่อมต่อดาต้าบัสของ TMS เข้ากับดาต้าบัสของวงจรหน่วยความจำ

- IBMDIR เมื่อสัญญาณนี้อยู่ในสถานะ High ข้อมูลจะถูกเคลื่อนย้ายในทิศทางออกจากหน่วยความจำไปยัง PC แต่เมื่ออยู่ในสถานะ Low ข้อมูลจะถูกเคลื่อนย้ายในทิศทางตรงกันข้าม

- TMSDIR เมื่อสัญญาณนี้อยู่ในสถานะ High ข้อมูลจะถูกเคลื่อนย้ายในทิศทางออกจากหน่วยความจำไปยัง TMS แต่เมื่ออยู่ในสถานะ Low ข้อมูลจะถูกเคลื่อนย้ายในทิศทางตรงกันข้าม



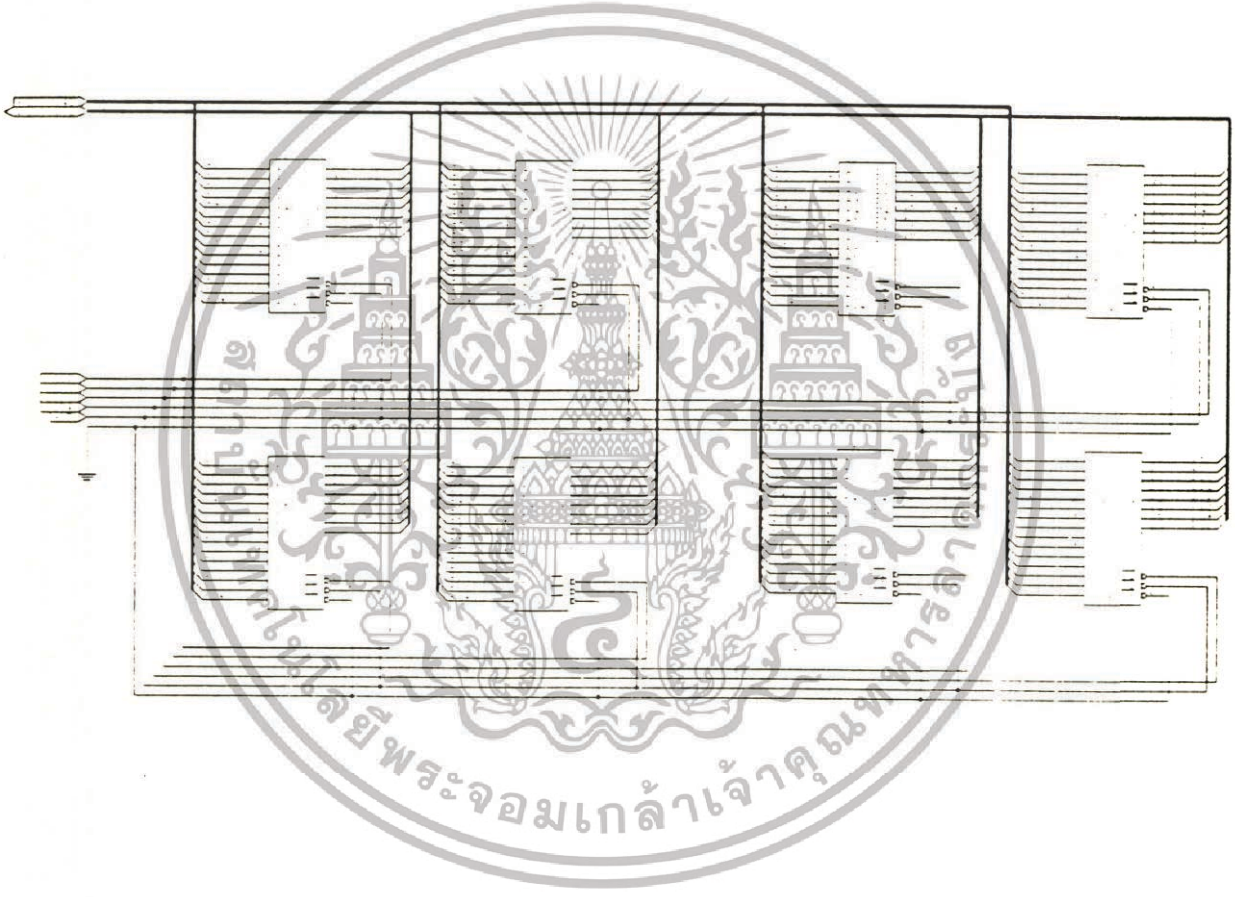
รูปที่ 2.7 วงจร Data Selector

**วงจรหน่วยความจำ**

วงจรหน่วยความจำ จะถูกใช้สำหรับเก็บโปรแกรมและข้อมูลที่จะใช้ในการทำงานของวงจร TMS ซึ่งวงจรหน่วยความจำมีขนาด 128 KWord โดยแบ่งเป็น หน่วยความจำข้อมูล 64 Kword และหน่วยความจำโปรแกรมอีก 64 KWord

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในวงจรหน่วยความจำเราเลือกใช้ Static RAM เบอร์ SC2568 ซึ่งมีขนาด 8x16 K จำนวน 8 ตัว ข้อมูลและโปรแกรมที่นำมาให้ TMS ทำงานได้มาจาก PC โดย PC จะเป็นตัวนำข้อมูลดังกล่าวมาเขียนลงบนวงจรหน่วยความจำ แล้วหลังจากนั้นวงจร TMS จะมาอ่านข้อมูลและโปรแกรมจากหน่วยความจำไปทำงาน



รูปที่ 2.8 วงจรหน่วยความจำ

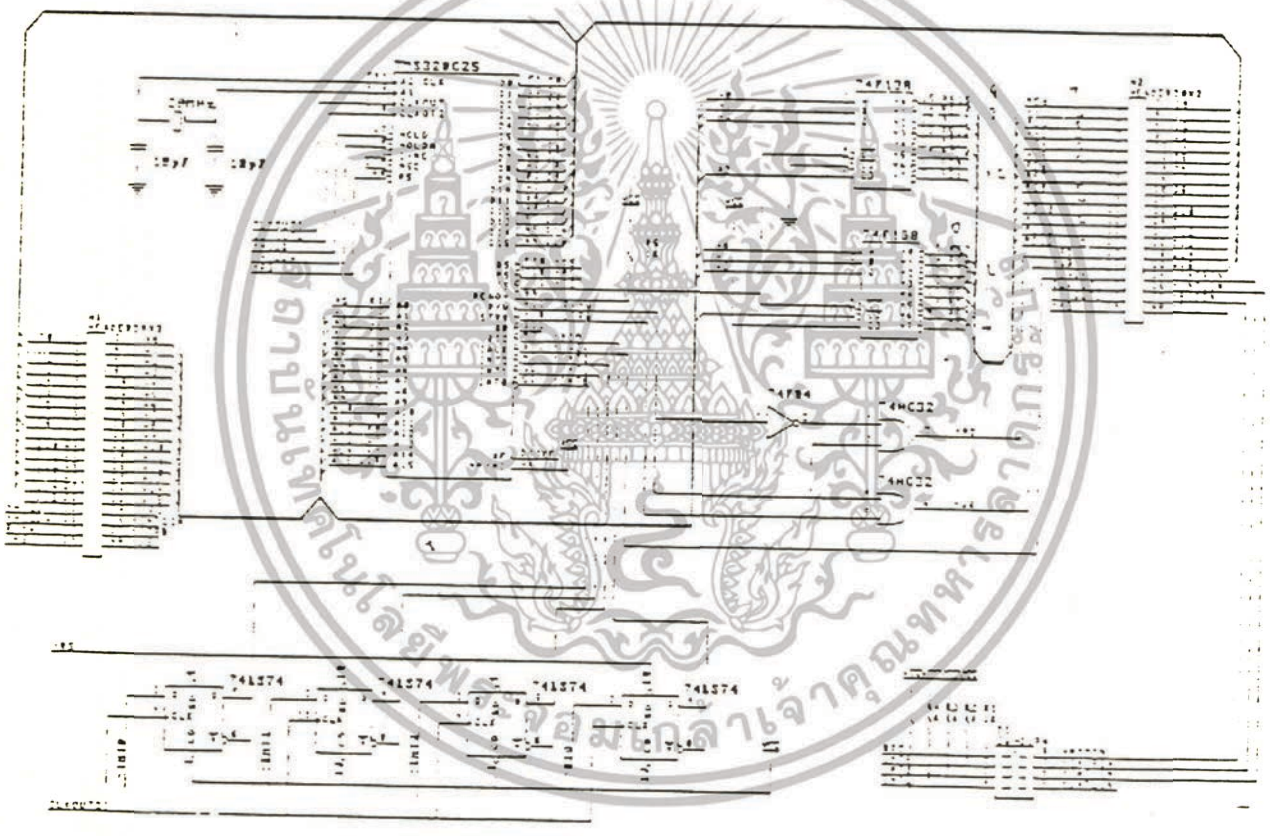
**วงจร TMS320C25 และสัญญาณควบคุม**

ภายในวงจรนี้ประกอบไปด้วย TMS320C25 ทำหน้าที่เป็นหน่วยประมวลผลกลางของวงจรนี้ทั้งหมด 74F138 จำนวน 2 ตัว สำหรับดีโคดพอร์ทของ TMS320C25 วงจร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สร้างสัญญาณ Read และ Write และ D Flip-Flop จำนวน 4 ตัว ใช้สำหรับกำหนดลำดับของสัญญาณ อินเทอร์รัพท์

สำหรับสัญญาณ CLOCK ของ TMS320C25 ในขั้นนี้เราเลือกใช้เพียง 20 MHz เพราะว่าอยู่ในขั้นทดลองวงจร ซึ่งการประกอบวงจรเราใช้การไวร์แลปแทนการออกแบบลายวงจรบนแผ่น PCB เพื่อให้สะดวกในการเปลี่ยนแปลงวงจรได้ตามต้องการ ซึ่งถ้าใช้สัญญาณ CLOCK ด้วยความเร็วสูงสุดที่ TMS ต้องการอาจทำให้เกิดความผิดพลาด อันเนื่องมาจากผลของการไวร์แลปได้




รูปที่ 2.9 วงจร TMS320C25

**การเตรียมการส่วน Processor (Processor Initialization)**

ก่อนที่จะนำเอาอัลกอลิทึ่มของดิจิตอลซิกแนลโปรเซสซิ่ง ที่ออกแบบไว้มาทำงานจะต้องมีการอินนิเชี่ยลโปรเซสเซอร์ก่อนทุกครั้ง ซึ่งโดยทั่วไปแล้วเราจะอินนิเชี่ยลโปรเซสเซอร์หลังจากการรีเซ็ท

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เราสามารถรีเซ็ทโปรเซสเซอร์ได้โดยให้สัญญาณ Low เข้าที่ขา RS ของโปรเซสเซอร์ไม่น้อยกว่าสัญญาณนาฬิกา 3 ลูก ซึ่งจะเป็นผลให้ TMS320C25 หยุดการทำงานและค่าโปรแกรมเคาน์เตอร์จะถูกเปลี่ยนให้เป็นศูนย์ ตำแหน่งของหน่วยความจำโปรแกรมจะไปชี้อยู่ที่ตำแหน่งศูนย์ และนอกจากนี้ยังมีการเปลี่ยนแปลงที่ Status bit

หลังจากที่เกิดการรีเซ็ทแล้ว เราควรที่จะอินนิเชี่ยลโปรเซสเซอร์เพื่อให้โปรเซสเซอร์เริ่มทำงานตามที่ต้องการ ซึ่งสิ่งที่เราควรจะอินนี่เชี่ยลหลังจากการรีเซ็ทคือ

- Memory-mapped rigisters
- Interrupt structure
- Mode control (OVM, SXM, FO, TXM, PM; plua HM and FSM on TMS320C25
- Memory control (CNF)
- Auxiliary registers and the auxiliary register pointer (ARP)
- Data memory page pointer (DP)

Example of Processor Initialization (TMS320C25)

```
TUTL        'PROCESSOR  INITIALIZARTION'
IDT         'EXAMPLE'
DEF         RESET, INTO, INT1, INT2
DEF         TINT, RINT, XINT, USER
REF         TIME, RCV, XMT, PROC

PROCESSOR INITIALIZATION FOR THE TMS320C25
            AORG  > 0000
RESER   B   INIT      ; RS-BEGINS PROCESSING HERE.
INT0    B   ISR0      ; INT0-BEGINS PROCESSING HERE.
INT1    B   ISR1      ; INT1-BEGINS PROCESSING HERE.
INT2    B   ISR2      ; INT2-BEGINS PROCESSING HERE.
            AORG  >  0018
TINT    B   TIMER     ; TIMER INTERRUPT PROCESSING.
RINT    B   RCV       ; SERIAL PORT RECEIVE PROCESSING.
XINT    B   XMT       ; SERIAL PORT TRANSMIT PROCESSING.
USER    B   PROCS     ; TRAP VECTOR PROCESSING BEGINS.
INIT        ROVM      ; DISABJ.E OVERFLOW MODE
            LDPK 0    ; POINT DP REGISTER TO DATA PAGE 0.
            LARP 7    ; POINT TO AUXILIARY REGISTER 7.
            LACK >3F  ; LOAD ACCKUMALATOR WITH > 3F.
            SACL4     ; ENABLE ALL INTERRUPTS VIA IMR.
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
INTERNAL DATA MEMORY INITAILIZATION
ZAC                 ; ZERO THE ACCUMULATOR.
LARK AR7,>60        ; POINT TO BLOCK B2.
RPTK 31
SACL *+             ; STORE ZERO IN ALL 32 LOCATIONS.
LRLK AR7,>200       ; POINT TO BLOCK B0.
RPTK 255
SACL *+             ; ZERO ALL THE PAGER 4 AND 5.
SRLK AR7,>300       ; POINT TO BLOCK B1
RPTK 255
SACL *=             ; ZERO ALL OF PAGES 6 AND 7
THE PROCESSOR ID INITIALIZED.
```

เราได้เขียนโปรแกรมที่มีการเซ็ทและรีเซ็ทค่า RPD Register เพื่อทำให้เกิด Timer interrupt โปรแกรมนี้ถูกเขียนขึ้นมาเพื่อใช้ในการตรวจสอบว่าวงจรที่ได้ออกแบบมานั้นทำงานได้ตามต้องการหรือไม่

ภายใน TMS320C25 มี Timer ขนาด 16 bit และใช้ร่วมกับการอินเทอร์รัพท์ในการปฏิบัติการฟังก์ชั่นต่างๆ Timer ตัวนี้เป็น Down counter ซึ่งได้รับ clock จากสัญญาณ CLKOUT1 ของ TMS320C25 อย่างต่อเนื่องและจะนับเป็นจำนวนเท่ากับ (PRD+1) ไซเคิล ของ CLKOUT1 เราสามารถทำให้เกิด Timer interrupt (TINT) ได้ทุกๆ 2-65536 ไซเคิล ของ TMS320C25 โดยการโปรแกรมค่าใน Period register (PRD)

Memory mapped register 2 ตัวถูกใช้ในการปฏิบัติการของ Timer Timer register (TIM) ซึ่งหมายถึง หน่วยความจำข้อมูลตำแหน่งที่ 2 จะมีค่าของ Timer ที่นับในปัจจุบันอยู่ที่ทุกๆ ไซเคิลของ CLKOUT1 ค่าใน TIM จะถูกลดลงทีละ 1 PRD register จะมีค่าเริ่มต้นในการนับของ Timer อยู่ เมื่อ TIM ถูกลดจนมีค่าเป็นศูนย์จะเกิด Timer interrupt ในไซเคิลต่อมาค่าใน PRD register จะถูกโหลดลงใน TIM ซึ่งในลักษณะนี้จะทำให้เกิด Timer interrupt ทุกๆ PRD+1 ไซเคิล

โปรแกรมนี้เป็นแอสเซมบลี้โค้ดในการประยุกต์ใช้ Timer สมมติเราโหลดค่า X ลงใน PRD register จะทำให้เกิดสัญญาณนาฬิกามีความถี่ตามสมการข้างล่าง คือ

CLKOUT1 / (PRD+1) = 2*Frequency of signal

จากวงจรเราใช้ CLOCKIN 20 MHz เพราะฉนั้น CLKOUT1 = 5 MHz

PRD = X

Frequency of signal = 5 MHz/2*(X+1)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ใน Timer interrupt service routine สัญญาณ XF ของ TMS320C25 จะถูก Toggle สัญญาณที่ขา XF นี้ถูกใช้เป็นสัญญาณอินพุทของ BZ0 ด้วยสัญญาณเอาท์พุทที่ขา XF จะเป็นสัญญาณนาฬิกาที่มี duty cycle 50% ตราบเท่าที่ยังไม่มีการดิสเอเบิ้ลอินเทอร์รัพท์ ซึ่งอินเทอร์รัพท์อาจจะดิสเอเบิลโดยตรง หรือโดยการใช้ DZNT หรือโดยการเอ็กซีคิวส์คำสั่งรีพีทโหมดก็ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# บทที่ 3
# การออกแบบและสร้างเครื่องกัดแนวดิ่งซีเอ็นซี

## 3.1 การออกแบบโครงสร้างทางด้านเครื่องกล

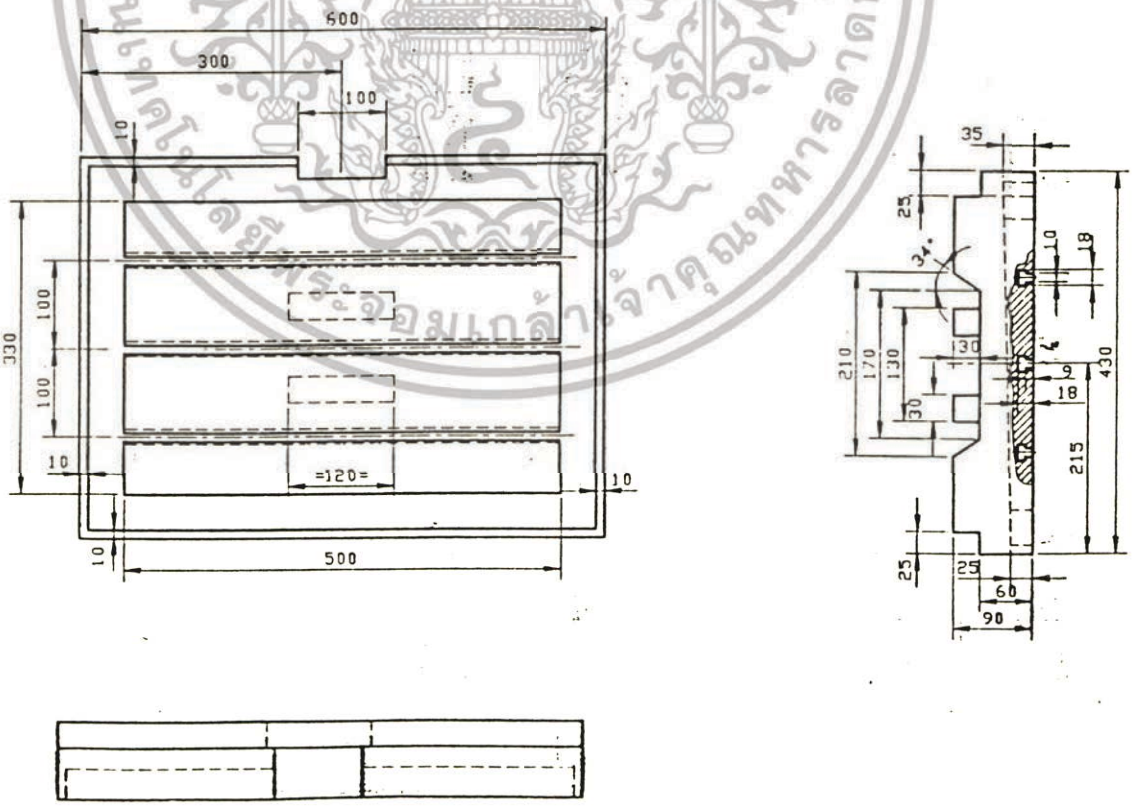
รายละเอียดของเครื่องจักรแนวดิ่งซีเอ็นซี ที่ต้องการและสามารถควบคุมด้วยคอม-พิวเตอร์ มีรายละเอียดดังนี้

1. โต๊ะงานทำด้วยเหล็กหล่อมีขนาด กว้าง x ยาว x หนา ไม่น้อยกว่า 400 x 550 x 60 มิลลิเมตร มีจำนวนร่องตัวทีมีขนาดกว้างไม่น้อยกว่า 10 มิลลิเมตร ตามแนวยาวของโต๊ะงานจำนวน 3 ร่อง
2. ฐานเครื่องทำด้วยเหล็กหล่อมีขนาด กว้าง x ยาว x สูง ไม่น้อยกว่า 650 x 800 x 250 มิลลิเมตร
3. Column ทำด้วยเหล็กหล่อมีขนาด กว้าง x ยาว x สูง ไม่น้อยกว่า 450 x 350 x 1,000 มิลลิเมตร
4. Ball Screw ขนาด 25 มิลลิเมตร ประกอบด้วยรายละเอียดข้างล่างนี้ 2 ชุด
   - ระยะคลาดเคลื่อนไม่เกิน +0.03 มิลลิเมตรต่อความยาว 500 มิลลิเมตร
   - ช่วงเกลียวชุบแข็งมีความแข็งไม่น้อยกว่า HRC 58
   - มีนัดประกอบกับเกลียวนัดเป็นชนิด Single-Flange Double-nut
5. Ball Screw ขนาด 25 มิลลิเมตร ประกอบด้วยรายละเอียดข้างล่างดังนี้ 1 ชุด
   - ขนาดเกลียวโตไม่น้อยกว่า 25 มิลลิเมตร
   - มีความยาวทั้งหมดไม่น้อยกว่า 850 มิลลิเมตร
   - มีความยาวช่วงเกลียวไม่น้อยกว่า 750 มิลลิเมตร
   - ขนาดเม็ดบอลไม่เล็กกว่า 3 มิลลิเมตร
   - ระยะ pitch 5 มิลลิเมตร
   - ระยะ Lead 5 มิลลิเมตร
6. Linear motion มีรายละเอียดดังนี้ 4 ชุด
   - มีขนาดบล๊อกบนรางเลื่อน กว้าง x ยาว x หนา ไม่น้อยกว่า 85 x 110 x 30 มิลลิเมตร
   - มีรางเลื่อน กว้าง x ยาว x หนา ไม่น้อยกว่า 25 x 800 x 25 มิลลิเมตร
   - ความคลาดเคลื่อนของขนาดในแนวราบและแนวดิ่ง ไม่เกิน 0.02 มิลลิเมตร ต่อระยะการเคลื่อนที่ 800 มิลลิเมตร
   - รางเลื่อนชุบแข็งมีความแข็งไม่น้อยกว่า HRC 58
   - มีบล๊อกประกอบบนรางเลื่อนๆ ละ 2 บล๊อก
7. Linear motion มีรายละเอียดดังนี้ จำนวน 2 ชุด
   - ขนาดบล๊อกบนรางเลื่อน กว้าง x ยาว x หนา ไม่น้อยกว่า 55 x 60 มิลลิเมตร
   - รางเลื่อน กว้าง x ยาว x หนา ไม่น้อยกว่า 25 x 600 x 25 มิลลิเมตร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ความคลาดเคลื่อนของขนาดในแนวราบและแนวดิ่ง ไม่เกิน 0.02 มิลลิเมตร ต่อระยะการเคลื่อนที่ 600 มิลลิเมตร
- รางเลื่อนชุบแข็งมีความแข็งไม่น้อยกว่า HRC 58
- มีบล๊อกประกอบบนรางเลื่อนๆ ละ 2 บล๊อก จำนวน 2 ชุด
- ขนาดโตนอกของเกลียวไม่ต่ำกว่า 25 มิลลิเมตร
- ความยาวทั้งหมดไม่น้อยกว่า 620 มิลลิเมตร
- ความยาวช่วงเกลียวไม่น้อยกว่า 550 มิลลิเมตร
- ขนาดของเม็ดบอลที่ใช้ไม่เล็กกว่า 3 มิลลิเมตร
- มีระยะ pitch 5 มิลลิเมตร
- มีระยะ Lead 5 มิลลิเมตร
- มีระยะคลาดเคลื่อนไม่เกิน + 0.03 มิลลิเมตรต่อความยาว 500 มิลลิเมตร
- ช่วงเกลียวชุบแข็งมีความแข็งไม่น้อยกว่า HRC 58
- มีนัดประกอบ ชนิดของนัดเป็น Single-Flang Double-nut

8. ชุดเพลาขับหัวเครื่อง มีรายละเอียดดังนี้ จำนวน 1 ชุด
   - มีขนาดของรูเพลาที่ใช้กับด้ามมีด ขนาดมาตรฐานไม่เล็กกว่า BT30
   - สามารถทนความเร็วรอบสูงสุดไม่ต่ำกว่า 3,500 รอบ/นาที
   - สามารถรองรับการขับเคลื่อนด้วยสายพาน
   - ลูกปืนที่ใช้ต้องเป็นแบบ high precision braring




รูปที่ 3.1 โครงสร้างของเครื่องกัดแนวดิ่งซีเอ็นซีส่วนที่ 1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 3.2 คำนวณน้ำหนักของส่วนต่าง ๆ

3.2.1 โครงสร้างของเครื่องกัดแนวดิ่งซีเอ็นซีส่วนที่ 1 โต๊ะงานทำด้วยเหล็กหล่อ ซึ่งมีขนาดกว้าง 430 ยาว 600 หนา 90 มิลลิเมตร ตามลำดับรูปที่ 3.1

- ส่วนของโต๊ะงานสามารถคำนวณน้ำหนักได้ดังนี้

น้ำหนักโต๊ะงาน = กว้าง x ยาว x หนา x ความหนาแน่นเหล็ก

= 430 x 600 x 90 x 7.5

= 43 x 60 x 0.90 x 7.5

= 17.415 กิโลกรัม

3.2.2 โครงสร้างของเครื่องกัดแนวดิ่งซีเอ็นซีส่วนที่ 2 ฐานเครื่องทำด้วยเหล็กหล่อมีขนาดกว้าง x ยาว x สูง = 650 x 1260 x 315 มิลลิเมตร ดังรูปที่ 3.2

น้ำหนักส่วนฐานเครื่องส่วนที่ 2 = 650 x 1260 x 315 x 7.5

= 65 x 126 x 31.5 x 7.5

= 1,934 กิโลกรัม

3.2.3 โครงสร้างของเครื่องกัดแนวดิ่งซีเอ็นซีส่วนที่ 3 Column ทำด้วยเหล็กหล่อ มีขนาดกว้าง x ยาว x สูง = 380 x 400 x 1200 มิลลิเมตร ตามลำดับดังรูปที่ 3.3

น้ำหนักส่วน Column ส่วนที่ 3

พื้นที่ภายนอก = 380 x 400 x 1200

= 38 x 40 x 120 x 7.5

= 1,368 กิโลกรัม

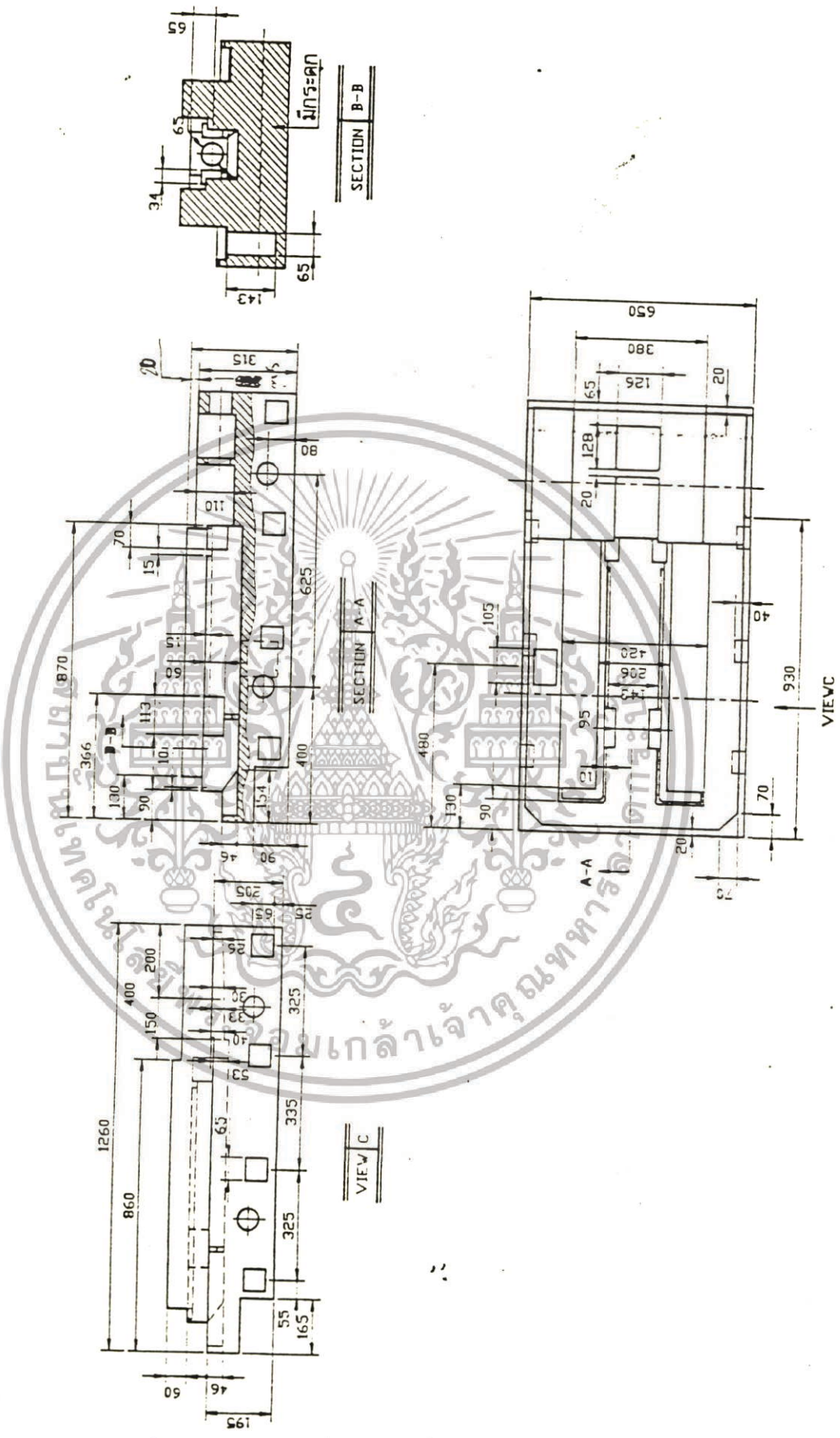
พื้นที่ภายใน = 270 x 270 x 1200

= 27 x 27 x 120 x 7.5

= 656.1 กิโลกรัม

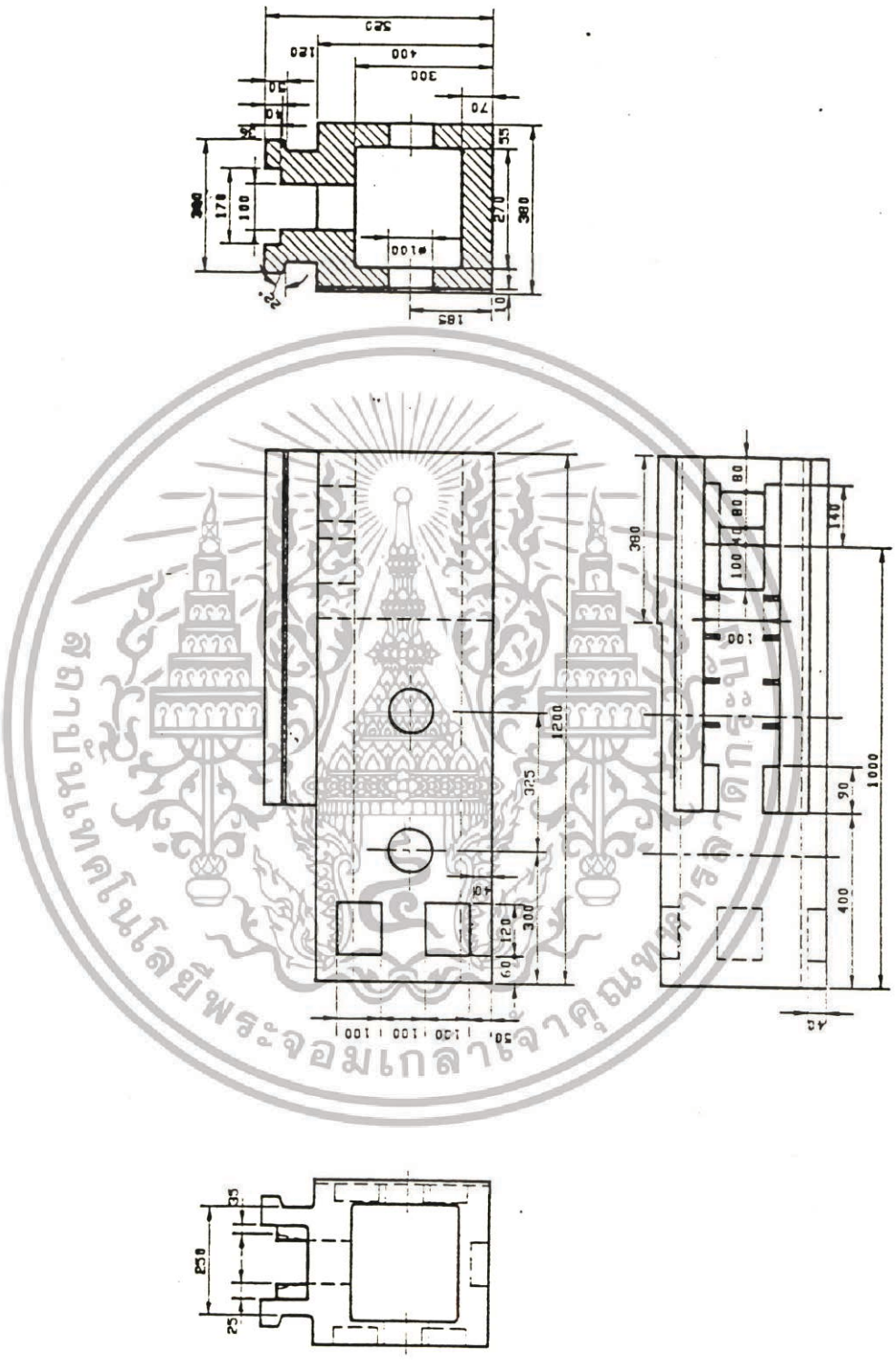
พื้นที่โครงสร้างส่วนที่ 3 = 711.9 กิโลกรัม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้




รูปที่ 3.2 โครงสร้างเครื่องกัดแนวดิ่งซีเอ็นซีส่วนที่ 2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



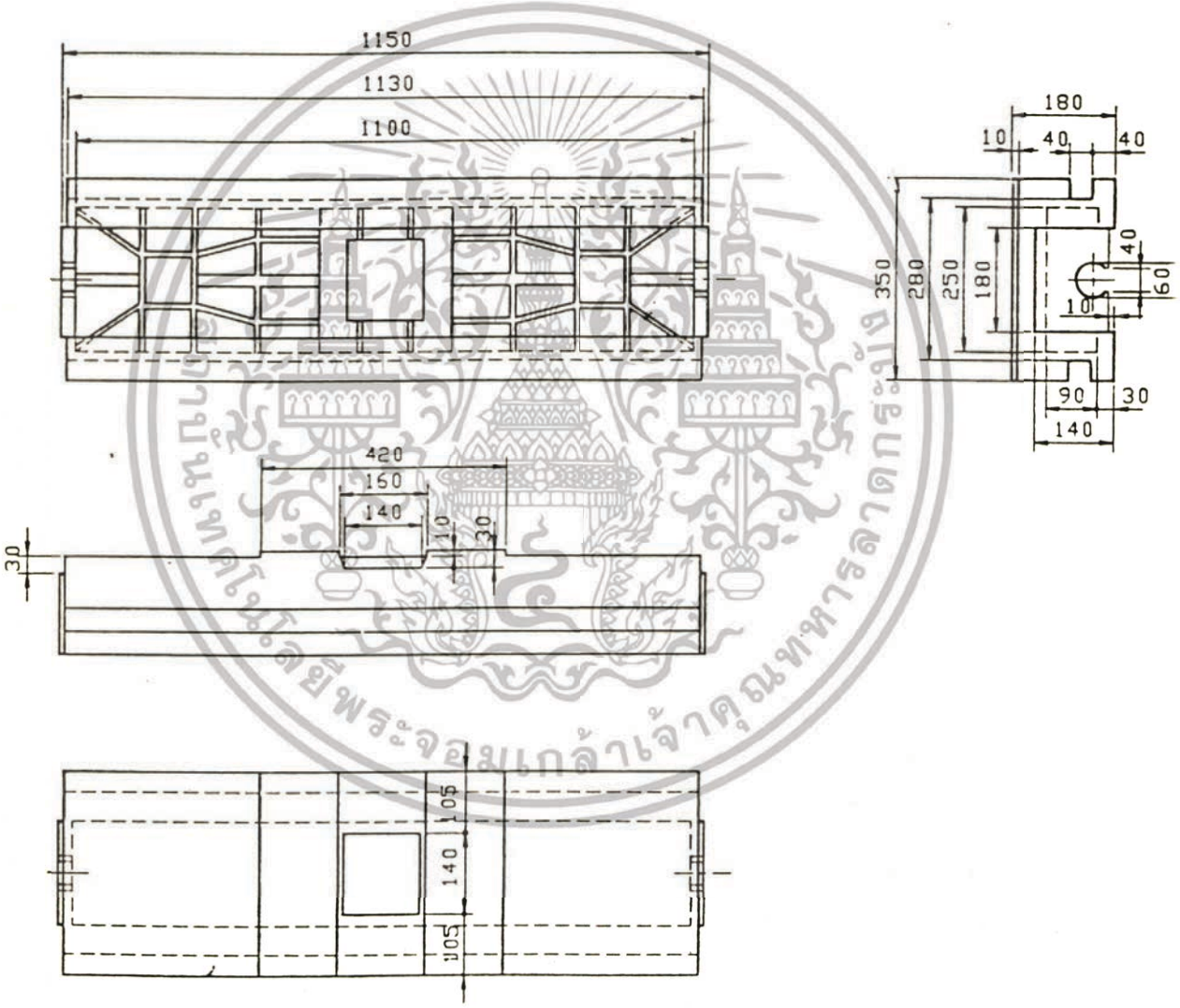
รูปที่ 3.3 โครงสร้างเครื่องกัดแนวดิ่งซีเอ็นซีส่วนที่ 3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.4 โครงสร้างของเครื่องกัดแนวดิ่งซีเอ็นซีส่วนที่ 4 เป็นองค์ประกอบของโต๊ะงาน ซึ่งมีขนาด กว้าง x ยาว x สูง = 350 x 1150 x 180 มิลลิเมตร ตามลำดับดังรูปที่ 3.4

น้ำหนักของส่วนที่ 4 = 35 x 115 x 18 x 7.5

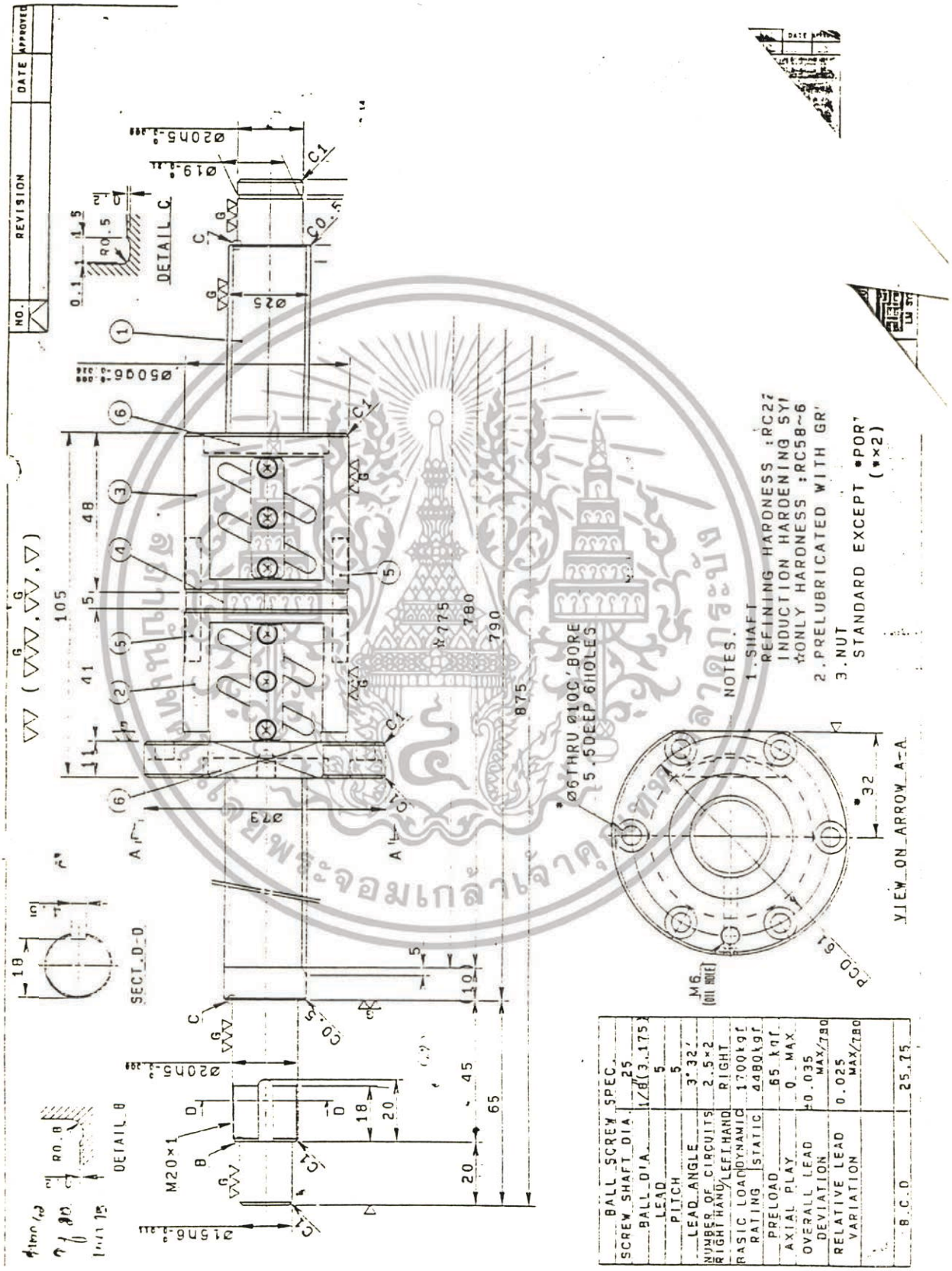
= 543.375 กิโลกรัม




รูปที่ 3.4 โครงสร้างเครื่องกัดแนวดิ่งซีเอ็นซีส่วนที่ 4

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.5 Ball Screw ขนาด 25 มิ ปที่ 3.7 ใน
ความยาว 650 มิลลิเม

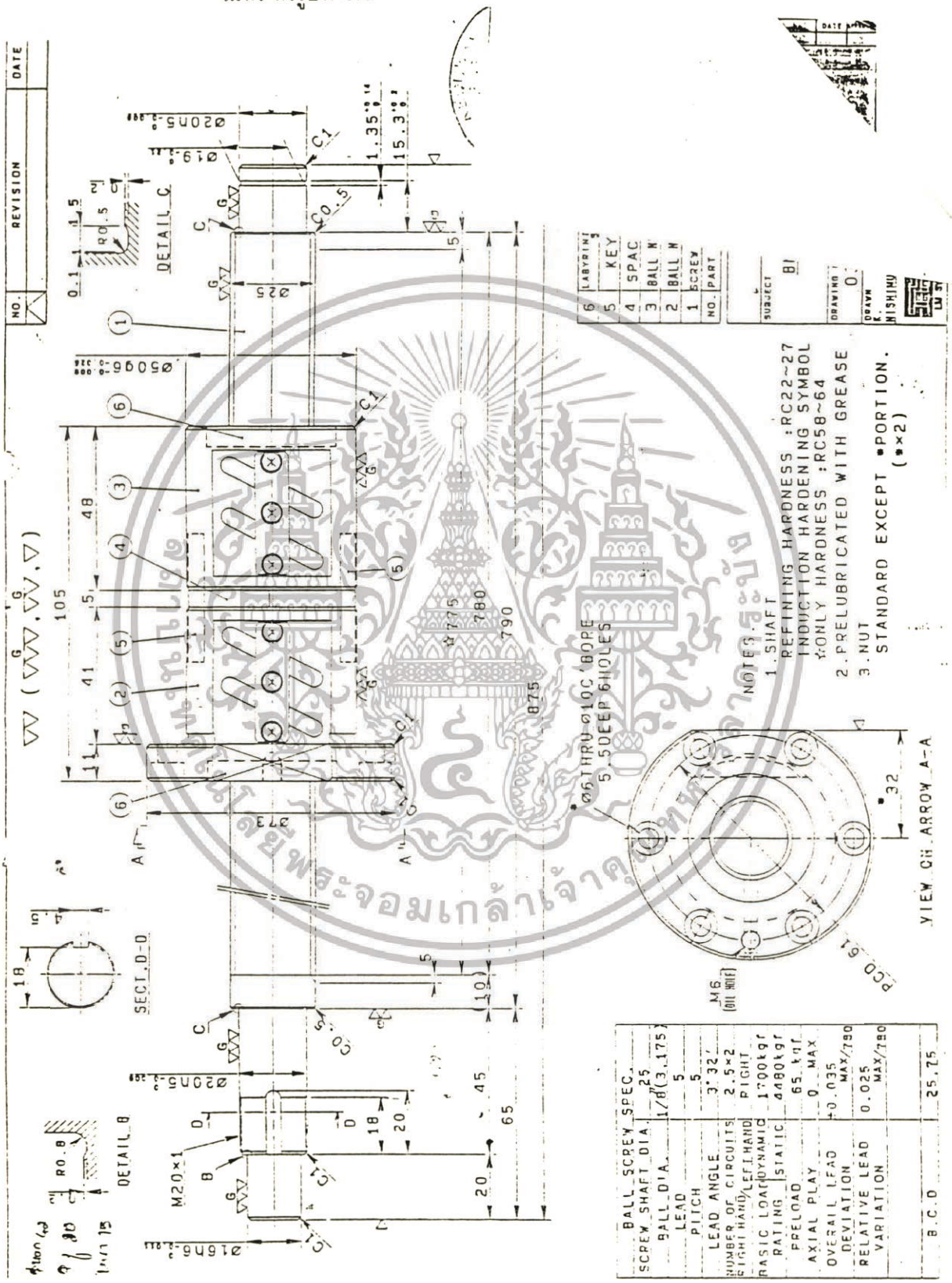



รูปที่ 3.5 Ball Screw ของแกน x และ y

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปที่ 3.7 ใน

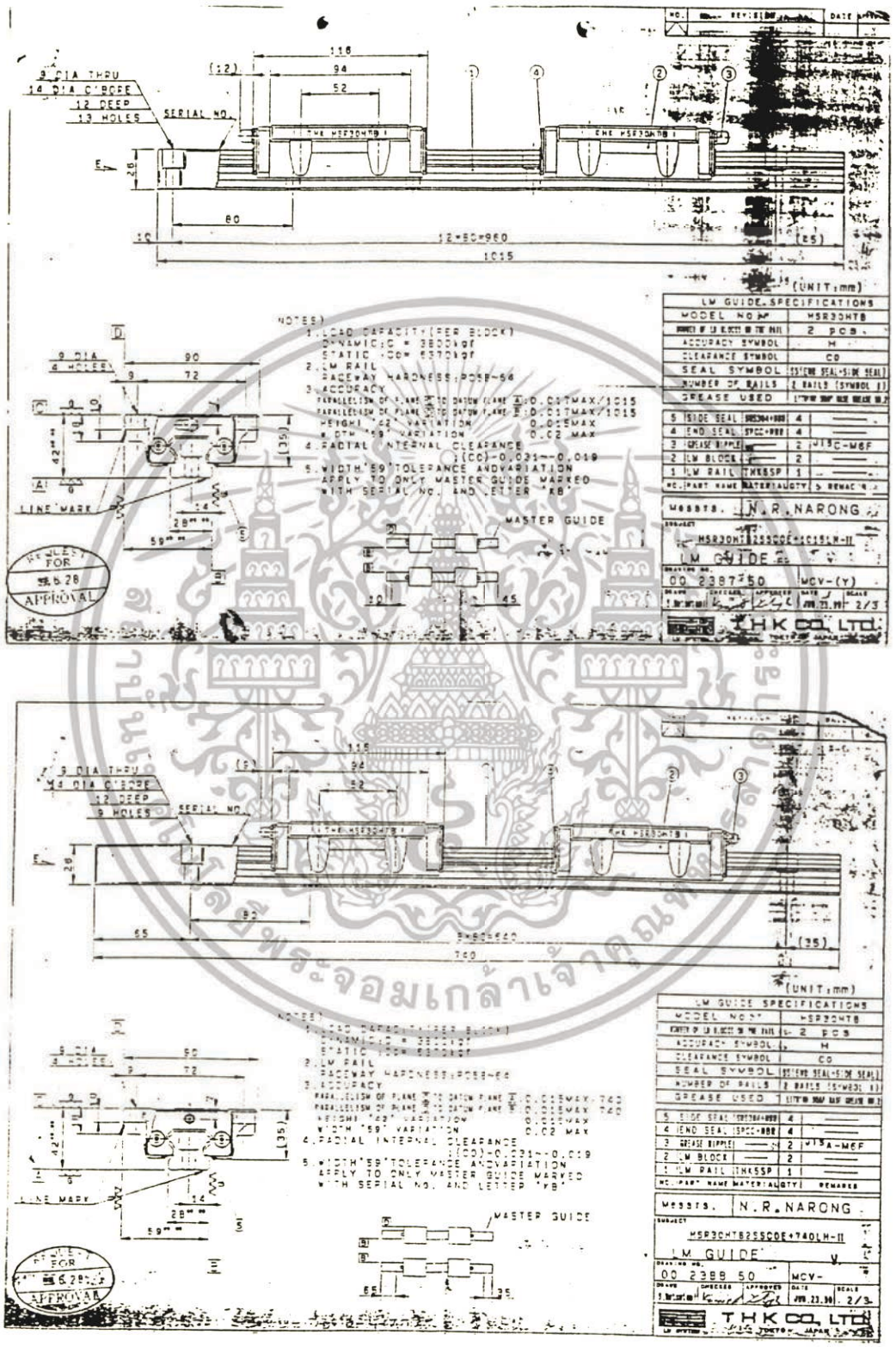
3.2.6 Ball Screw ในแนวแกน z ซึ่งมีข
เมตร ดังรูปที่ 3.6




รูปที่ 3.6 Ball Screw ของแนวแกน z

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.7 Linear motion มีรายละเอียดต่างๆ และโครงสร้างดังรูปที่ 3.7 ในแนวแกน x, y และ z ตามลำดับ

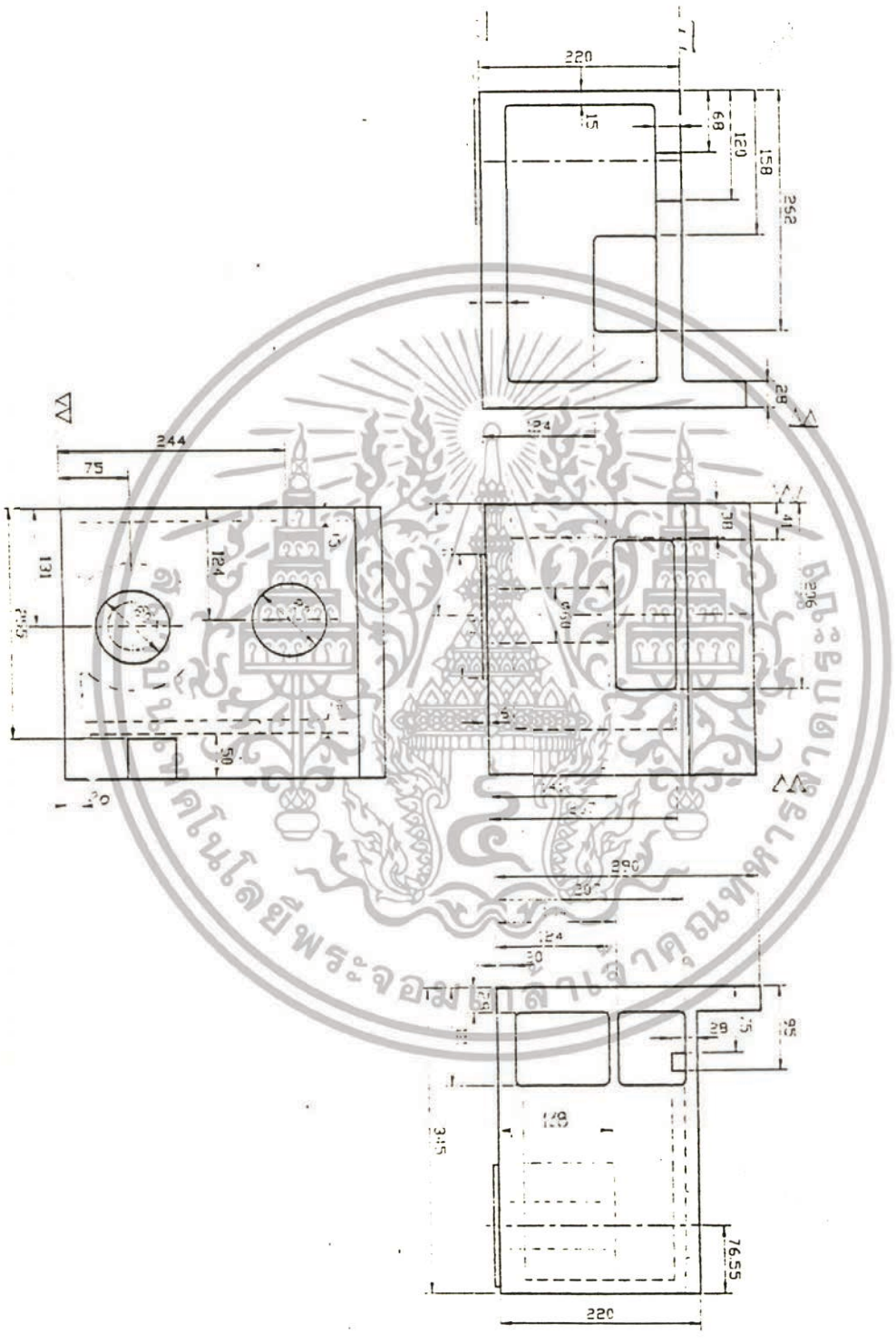



รูปที่ 3.7 Linear motion ของโครงสร้างเครื่องจักรในแนวแกน x,y และ z

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

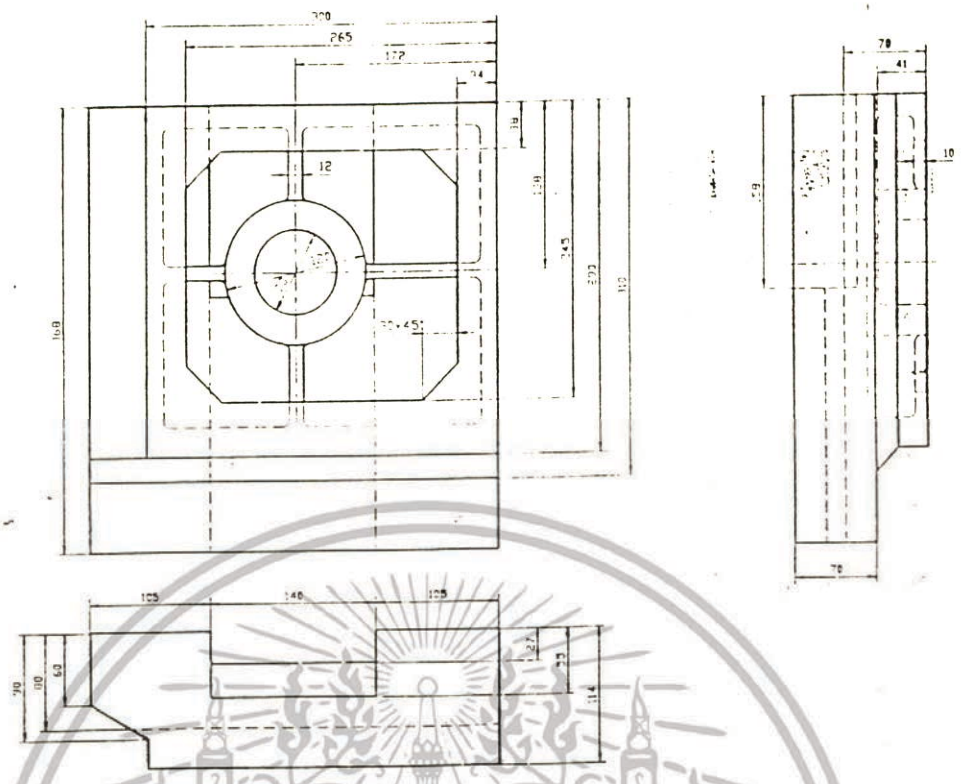
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.8 โครงสร้างเครื่องกัดแนวดิ่งซีเอ็นซี ในส่วนเพลาขับหัวเครื่อง ซึ่งมีรายละเอียด ดังรูปที่ 3.8 และ 3.9

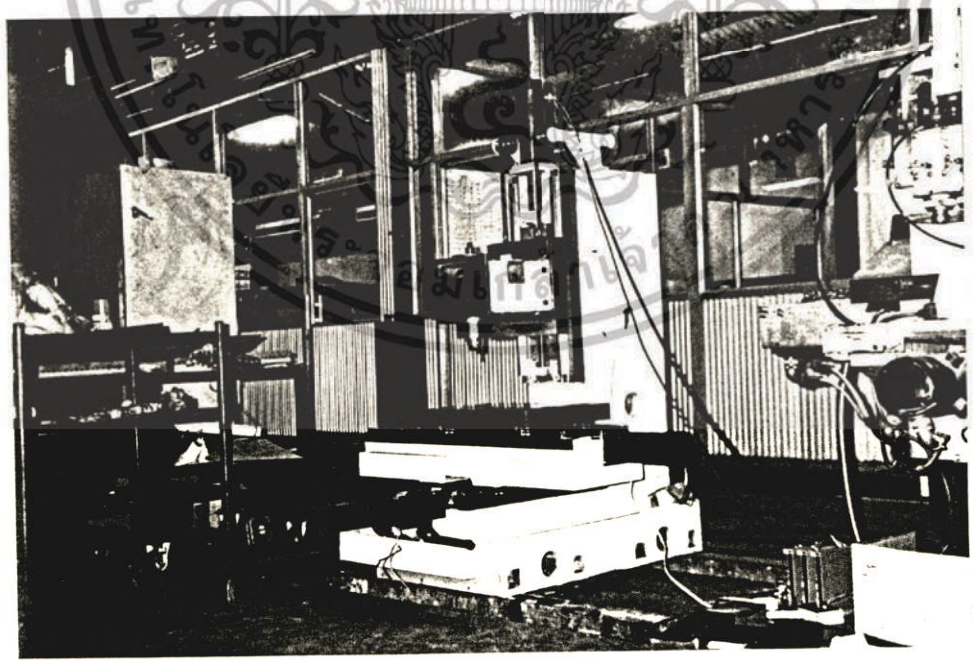



รูปที่ 3.8 ชุดเพลาขับหัวเครื่อง 1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.9 ชุดเพลาขับหัวเครื่อง 2



รูปที่ 3.10 รูปแบบของเครื่องกัดโพรงแนวดิ่งซี.เอ็นซีเมื่อประกอบทุกส่วนเข้าด้วยกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.3 การคำนวณขนาดเซอร์โวมอเตอร์ในการขับเคลื่อนแนวแกน x, y และ z

ในการคำนวณขนาดกำลังของเซอร์โวมอเตอร์ในการขับเคลื่อนแบบ x, y และ z ต้องทราบค่าตัวแปรต่างๆ ที่นำมาคำนวณในตารางที่ 3.1 จะเป็นการใส่ข้อมูลเบื้องต้น

**ตารางที่ 3.1 ข้อมูลของตัวแปรต่าง ๆ ในการคำนวณขนาดกำลังของเซอร์โวมอเตอร์**

| ข้อมูลของเครื่องจักรกล | X-axis | Y-axis | Z-axis |
|---|---|---|---|
| Weight of table (include workpiece) | 100 Kg | 100 Kg | 100 Kg |
| Diameter Ball Screw | 25 mm | 25 mm | 25 mm |
| Lenght of Ball Screw | 875 mm | 650 mm | 650 mm |
| Pitch of Ball Screw | 5 mm | 5 mm | 5 mm |
| a) Pully Motor (or connecting element)<br>DIA of coupling<br>Length of coupling | <br>70 mm<br>80 mm | <br>70 mm<br>70 mm | <br>70 mm<br>80 mm |
| Material | Steel | Stell | Steel |
| Rapid Speed | 10 m/min | 10 m/min | 10 m/min |
| Cutting Force | 10 Kg | 10 Kg | 10 Kg |
| Gear Ratio | 1 : 1 | 1 : 1 | 1 : 1 |
| Friction Factor Steel/Steel | 0.15 | 0.15 | 0.15 |
| % of Cutting time | 60% | 60% | 60% |
| % of Postioning time | 35% | 35% | 35% |
| % of Accelelation time for milling | 5% | 5% | 5% |

**หลักการคำนวณขนาดกำลัง (Torque) ของเซอร์โวมอเตอร์ สำหรับเครื่องกัดเซาะโลหะแนวดิ่งซีเอ็นซี**

$$TORQUE_{total} = T_{friction} + T_{weight} + T_{cutting} + T_{acceleration}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1) $TORQUE_{friction} = (TF_{(tables)} + TF_{(spindle)}) \cdot i$

$$T_{F(table)} = \frac{m \cdot g \cdot \mu \cdot h \cdot i}{2\pi}$$

$$= \frac{100\ kg \times 9.81 \times 0.15 \times 0.05\ m \cdot 1}{6.28}$$

$$= 0.11\ Nm$$

$T_{F(spindle)} = DIA(screw) \cdot i / 10$

$= (25\ mm / 10) \cdot 1$

$= 2.5\ Nm$

$T_F = (0.11\ Nm + 2.5\ Nm) \cdot i$

ที่ $i = 1/1$

$T_F = 2.61\ Nm$

2) $T_{weight} = \frac{m \times g \times \sin\alpha \times h \cdot i}{2\pi}$

$$= \frac{(100\ kg \times 9.18 \times 0 \times 0.15\ m) \cdot 1}{6.28}$$

$= 0\ Nm$

3) $T_{cutting} = \frac{F \times g \times h \cdot i}{2\pi}$

$$= \frac{(10\ kg \times 9.18 \times 0.005\ m) \cdot 1}{6.28}$$

$= 0.49\ Nm$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$J_{total} = (0.000063 + 0.0000003 + 0.0014) \cdot (i)^2$$

$$= 0.00034 \ kgm^2$$

เปรียบเทียบความสัมพันธ์ระหว่างแรงเฉื่อย (Inertia) มอเตอร์กับแรงเฉื่อย (Inertia) ของระบบ ($J_{total}$) จะเห็นว่า -

$$J_{system} < J_{motor}$$

$$0.00034 \ kgm^2 < 0.0005 \ kgm^2$$

ผลสรุปในการเปรียบเทียบอัตราส่วนระหว่างแรงเฉื่อยทั้งสองคือ

| $J_{system}$ | : | $J_{motor}$ | ผลสรุป |
|---|---|---|---|
| <1 | : | 1 | ดีมาก |
| 1 | : | 1 | ดี |
| 1.5 | : | 1 | ยอมรับได้ |
| 2 | : | 1 | ต้องระวัง |

มอเตอร์สำหรับแกน X ต้องมีขนาด คือ

Torque = 3.1 Nm

rpm = 2000 รอบ/นาที

Inertia motor = 0.0005 $kgm^2$

ทำนองเดียวกันหลักการคำนวณแกน y และแกน z จะคำนวณลักษณะเดียวกันและได้ขนาดกำลังคือ

แกน y Torque = 2.7 Nm

rpm = 2000 รอบ/นาที

Inertia motor = 0.0005 $kgm^2$

แกน z Torque = 2.7 Nm

rpm = 2000 รอบ/นาที

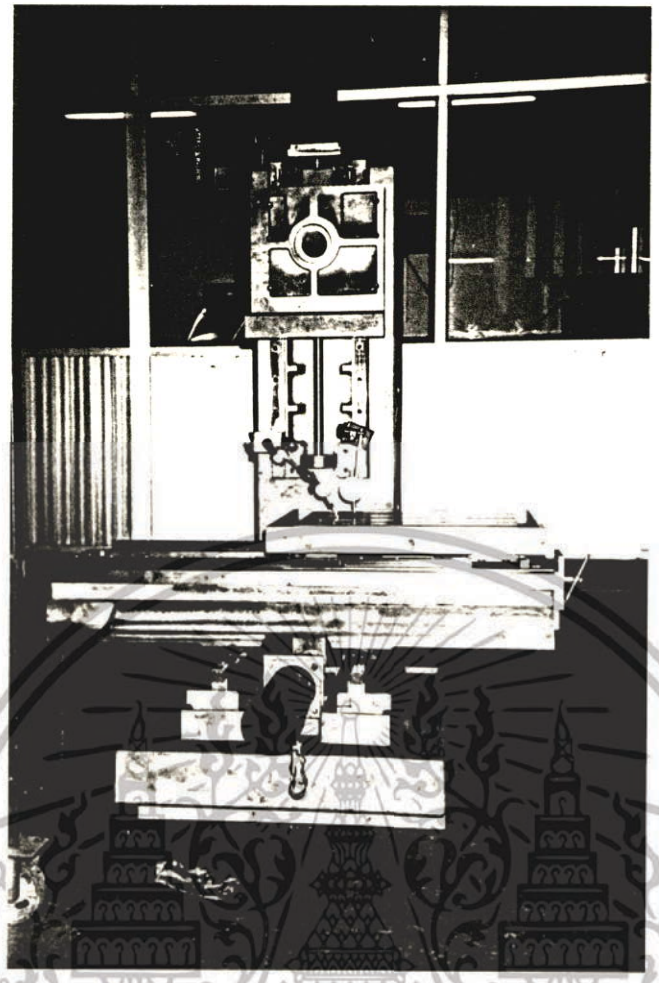
Inertia motor = 0.0005 $kgm^2$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

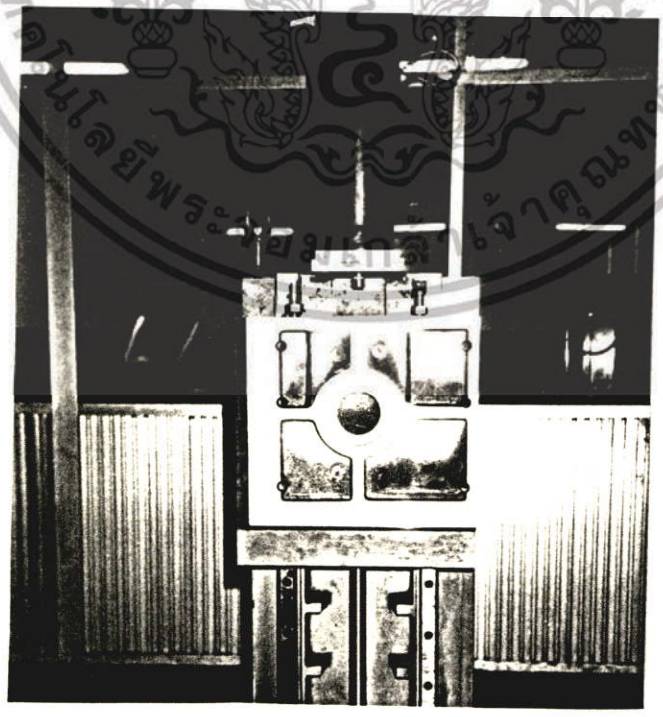
## ค่าของตัวแปรต่าง ๆ ในการคำนวณมอเตอร์

1. $T_F$ = Torque friction
2. $T_{F(table)}$ = Torque friction table
3. $T_{F(Screw)}$ = Torque friction screw
4. i = Gear Ratio
5. m = Mass of axis including workpiece
6. g = 9.81 m/sec
7. $\mu$ = Friction factor (Steel/Steel 0.15, Steel/turcite 0.1)
8. h = Pitch of screw
9. n = Efficiency Factor
10. Tw = Torque weight
11. Sind = Angle of axis to horizontal
    Horizontal axis = sin $\alpha$ = 0
    Vertical axis = sin $\alpha$ = 0
12. $T_C$ = Torque cutting
13. F = Cutting force
14. $V_{max}$ = Rapid speed
15. $J_T$ = Inertia Table including Work piece
16. $J_S$ = Inertia screw
17. $J_{PS}$ = Inertia Pully screw (or connecting element)
18. $J_{PM}$ = Inertia pully motor
19. $J_M$ = Inertia motor
20. DIA = Diameter of screw (or pully)
21. l = Length of screw (or pully)
22. d = Density (Steel 7700 $Kg/m^3$)
23. $T_{acc}$ = Acceleration Torque
24. RPM = RPM of motor (max. speed)
25. $t_{acc}$ = Acceleration time from 0-max. speed
26. $T_{acc}$ = Acceleration Torque
    Torque max. of motor - $(T_F + T_W)$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.11 การติดตั้งมอเตอร์กับ Ball screw แบบตรงในแนวแกน x และ y



รูปที่ 3.12 การติดตั้งมอเตอร์กับ Ball screw แบบตรงในแนวแกน z

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# บทที่ 4
# การออกแบบและพัฒนาโปรแกรมเครื่องกัดแนวดิ่งซีเอ็นซี

## 4.1 กล่าวนำ

ในส่วนของโปรแกรมเครื่องกัดแนวดิ่งซีเอ็นซีเป็นส่วนสำคัญอีกส่วนหนึ่ง ที่ต้องอาศัยความรู้ ความสามารถทางด้านวิศวกรรมคอมพิวเตอร์ผนวกกับความรู้ ความสามมรถทางด้านเครื่องกลหรืออุตสาหการเพื่อสร้าง และพัฒนาโปรแกรมควบคุมเครื่องจักรกลซีเอ็นซีที่เหมาะสม และถูกต้องมากที่สุด

## 4.2 การออกแบบในส่วนโปรแกรม

ในการออกแบบโปรแกรมเครื่องกัดแนวดิ่งซีเอ็นซี จะต้องทราบถึงรายละเอียดของโปรแกรมมาตรฐานสากล (ISO-CODE) ดังมีรายละเอียดดังนี้

**ตารางที่ 4.1 รายละเอียดของ ISO-CODE**

| G-CODE | กลุ่ม | รายละเอียดฟังก์ชั่น |
|---|---|---|
| G00 | 01 | เข้าสู่ตำแหน่งที่กำหนดอย่างรวดเร็ว |
| G01 | | ตัดตามระยะทางที่เป็นเส้นตรง |
| G02 | | ตัดตามแนวโค้งตามเข็มนาฬิกา CW |
| G03 | | ตัดตามแนวโค้งทวนเข็มนาฬิกา CCW |
| G04 | 00 | หยุดชั่วขณะหนึ่งแล้วถอยกลับ |
| G10 | | ตั้งข้อมูลอยู่กับที่แล้วจึงถอยกลับ |
| G17 | 02 | เลือกระนาบ xy |
| G18 | | เลือกระนาบ zx |
| G19 | | เลือกระนาบ yz |
| G20 | 06 | ป้อนหน่วยเป็นนิ้ว |
| G21 | | ป้อนหน่วยเป็นมิลลิเมตร |
| G27 | 00 | กลับไปตรวจสอบที่จุดอ้างอิง |
| G28 | | กลับไปที่จุดอ้างอิง |
| G29 | | กลับไปจากจุดที่อ้างอิง |
| G30 | | กลับไปที่จุดอ้างอิงที่ 2 |
| G31 | | กระโดดข้ามฟังก์ชั่น |
| G39 | | เลื่อนตัดมุมโค้ง |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

| G-CODE | กลุ่ม | รายละเอียดฟังก์ชั่น |
|---|---|---|
| G40 | 07 | ยกเลิกการชดเชยรัศมีดอกกัด |
| G41 | | ชดเชยรัศมีดอกกัดจากซ้ายไปขวา |
| G42 | | ชดเชยรัศมีดอกกัดจากขวาไปซ้าย |
| G43 | 08 | ชดเชยระยะความสูงของดอกกัด (+) |
| G44 | | ชดเชยระยะความสูงของดอกกัด (-) |
| G45 | | ยกเลิกการชดเชยระยะความสูงของดอกกัด |
| G65 | 00 | คำสั่งคัชดอมมาโคร |
| G66 | 01 | เรียกใช้คัชดอมมาโคร |
| G67 | | ยกเลิกเรียกใช้คัชดอมมาโคร |
| G73 | 09 | เจาะ กระแทกเป็นจังหวะ |
| G74 | | ทำเกลียวเป็นจังหวะ |
| G76 | | คว้านรูละเอียด |
| G80 | | ยกเลิกจังหวะที่กำหนดตายตัว |
| G81 | | เจาะเป็นจังหวะ, คว้านรูเป็นจุดๆ |
| G82 | | เจาะเป็นจังหวะ, การคว้านผายปากรู |
| G83 | | เจาะกดเป็นจังหวะ |
| G84 | | จังหวะการทำเกลียว |
| G85 | | คว้านรูเป็นจังหวะ |
| G86 | | คว้านรูเป็นจังหวะ |
| G87 | | จังหวะคว้านถอยหลังกลับ |
| G88 | | คว้านรูเป็นจังหวะ |
| G89 | | คว้านรูเป็นจังหวะ (เจาะคายเศษ) |
| G90 | 03 | โปรแกรมคิดระยะจุดแรกไปถึงจุดสุดท้าย |
| G91 | | โปรแกรมคิดระยะจุดหนึ่งไปถึงอีกจุดหนึ่ง |
| G92 | 05 | กำหนดระนาบแกน xy, zx, yz |
| G94 | | อัตราการป้อนเป็นนาที |
| G95 | | ไม่ใช้ |
| G98 | 04 | ถอยกลับจังหวะสูงสุด |
| G99 | | กลับถึงจุดอ้างอิง (จุด R) |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.2 รายละเอียดของ M-CODE

| M-CODE | รายละเอียดฟังก์ชั่น |
|---|---|
| M00 | หยุดโปรแกรม |
| M01 | โปรแกรมหยุดเพิ่มเติม |
| M02 | เริ่มต้นโปรแกรมใหม่ |
| M03 | หัวกัดหมุนตามเข็มนาฬิกา CW |
| M04 | หัวกัดหมุนทวนเข็มนาฬิกา CCW |
| M05 | หัวกัดหยุดหมุน |
| M08 | เปิดน้ำหล่อเย็น |
| M09 | ปิดน้ำหล่อเย็น |
| M10 | หัวจับ Tool ยึดแน่น |
| M11 | หัวจับปล่อย Tool |
| M19 | หัวกัดหมุนทดลองการหมุน |
| M20 | เปิดลม |
| M21 | ปิดลม |
| M32 | เปิดสมมาตรตามแกน x |
| M33 | เปิดสมมาตรตามแกน y |
| M35 | เปิดไม่ให้สมมาตร |
| M36 | เปิดยกเลิกความเร็วเกินพิกัด |
| M37 | ปิดยกเลิกความเร็วเกินพิกัด |
| M52 | เปิดคลายล็อคโต๊ะชิ้นงาน |
| M53 | ปิดคลายล็อคโต๊ะชิ้นงาน |
| M81 | โต๊ะหัวแบ่ง |
| M98 | เรียกโปรแกรมย่อย |
| M99 | จบการเรียกโปรแกรมย่อย |

ในรายละเอียดดังกล่าวเป็นโปรแกรมการสั่งงานของเครื่องจักรกลซีเอ็นซี ดังนั้นในระบบของโปรแกรมเครื่องกัดแนวดิ่งซีเอ็นซีจำเป็นต้องมีโปรแกรมการสั่งงานดังกล่าว แต่ไม่รวมถึงคำสั่งทั้งหมด ดังนั้นโครงสร้างการออกแบบโปรแกรมจะต้องมีรายละเอียดดังนี้

1. โปรแกรมส่วนกลางที่มีหน้าที่เชื่อมต่อระหว่างคอมพิวเตอร์ กับอุปกรณ์ฮาร์ดแวร์ที่ได้ออกแบบในการควบคุมมอเตอร์ หรือเรียกว่า "DRIVER Software"

2. โปรแกรมส่วนที่มีหน้าที่ในการควบคุมระบบฮาร์ดแวร์ และติดต่อกับผู้ใช้โดยผ่าน "DRIVER Software" ประกอบด้วยโหมดของการทำงานต่างๆ ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1. โหมดอัตโนมัติ (Automatic mode)
2. โหมดธรรมดา (Manual mode)
3. โหมดแบบช่วง (Jog mode)
4. โหมดการเขียนและแก้ไขโปรแกรม (Edit mode)

3. โปรแกรมส่วนของการจำลองการทำงานตามโปรแกรมที่เขียน (Simulation) โดยจำลองการทำงานในเชิงทางด้านกราฟฟิกส์

4. โปรแกรมในส่วนการติดต่อสื่อสารกับข้อมูลภายนอกที่อยู่ในรูปของ Text file หรือแบบเทปกระดาษ

**4.2.1 รายละเอียดฟังก์ชั่นต่าง ๆ ในโหมดอัตโนมัติ (Automatic Mode)**

ในรายละเอียด Automatic Mode สามารถกำหนดฟังก์ชั่นการทำงานต่างๆ ได้ดังนี้

- หยุดโปรแกรม (Abort Program) เพื่อการหยุดโปรแกรมและส่งสัญญาณไปบอกระบบฮาร์ดแวร์ในการควบคุมมอเตอร์หยุดการทำงาน ถ้าจะเริ่มทำงานใหม่ก็จะต้องโหลดโปรแกรมเข้าไปใหม่

- หยุดการทำงานของมอเตอร์ที่ควบคุมการเคลื่อนที่ของแกนต่างๆ (Hold Axis) เมื่อสั่งให้มีการ Hold ก็จะกำหนดให้คำว่า "HOLDING" แสดงผลออกมาในส่วนแสดงสถานะการทำงาน

- การสั่งให้โปรแกรมทำงานทีละ Block โปรแกรม (Single Block หรือ Single Step) ในโหมดดังกล่าว โปรแกรมจะกำหนดให้ฮาร์ดแวร์มีการสั่งการทำงานทีละบรรทัด และเมื่อโหมดดังกล่าวทำงานก็จะให้คำว่า "SINGLE" แสดงสถานะ

- สั่งให้โปรแกรมทำงาน (Program Start) โปรแกรมจะทำงานทั้งหมด และติดต่อกับฮาร์ดแวร์เพื่อควบคุมการทำงานให้สอดคล้องกับโปรแกรม

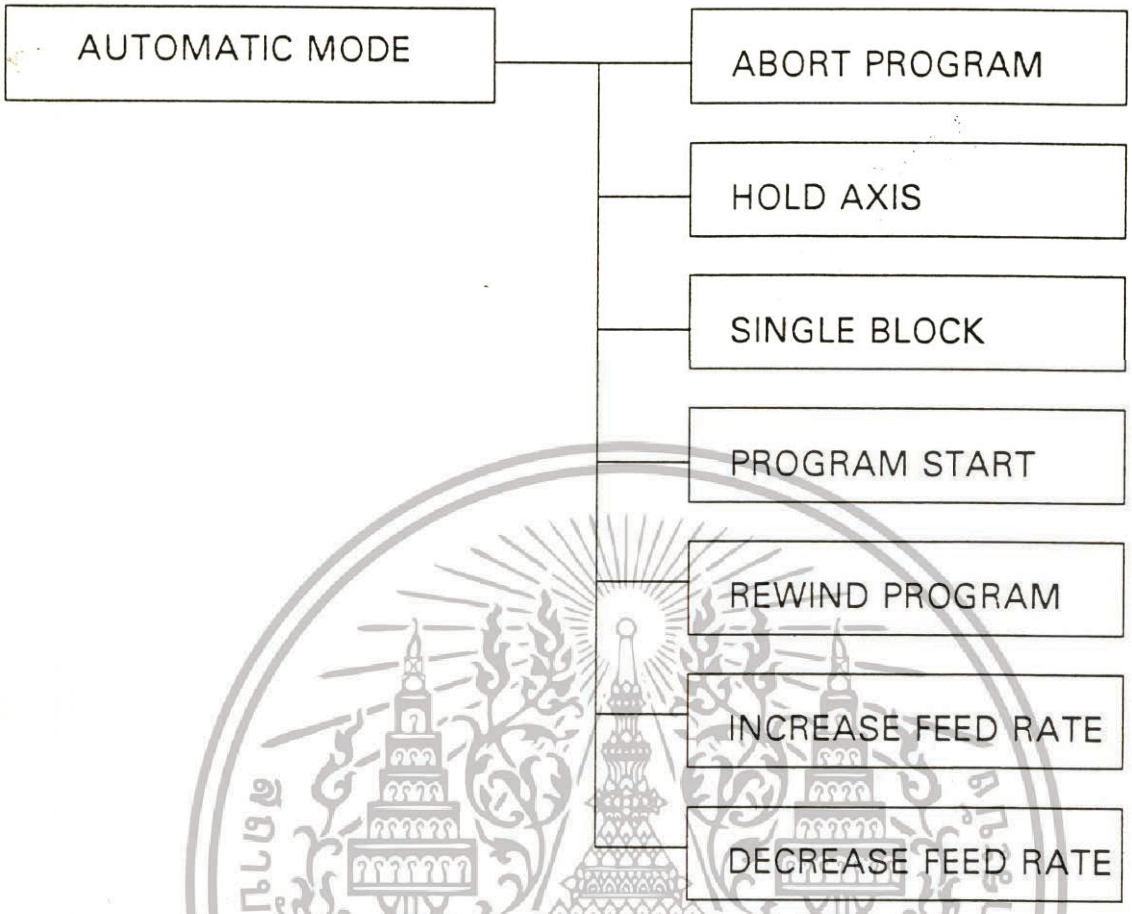
- เริ่มที่จุดเริ่มต้นของโปรแกรม (Rewind Program) ให้เริ่มการทำงานที่ต้นโปรแกรม

- การเพิ่มความเร็วของการเคลื่อนที่ (Increase feed rate override) เป็นการเพิ่มความเร็วของการเคลื่อนที่การกัดชิ้นงานของเครื่องกัดแนวดิ่งซีเอ็นซีทั้ง 3 แกน

- การลดความเร็วของการเคลื่อนที่ (Decrease feed rate override) เป็นการลดความเร็วของการเคลื่อนที่การกัดชิ้นงานของเครื่องกัดแนวดิ่งซีเอ็นซีทั้ง 3 แกน

ในรูปที่ 4.1 แสดงถึงฟังก์ชั่นต่างๆ ของการทำงานใน Automatic Mode

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้




รูปที่ 4.1 ฟังก์ชันการทำงาน AUTOMATIC MODE

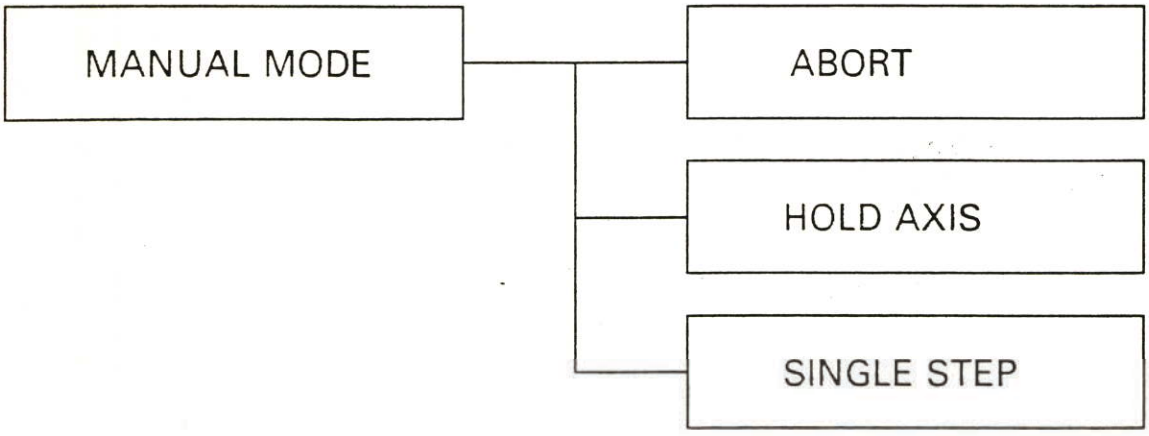
4.2.2 โหมดธรรมดา (Manual Mode (MDI)

ในรายละเอียดของโหมดธรรมดาได้แบ่งฟังก์ชั่นการทำงานต่างๆ ดังนี้

- หยุดการทำงาน (Abort) คือ หยุดการเคลื่อนที่
- หยุดพักการเคลื่อนที่ของแกน (Hold Axis) ส่งสัญญาณไปฮาร์ดแวร์แล้วหยุดการเคลื่อนที่ของทุกแกนมอเตอร์และแสดงสถานะคำว่า "HOLDING"
- การสั่งงานเป็น Block ของโปรแกรม (Single Step) ในโหมดธรรมดา โดยกำหนดให้เครื่องจักรทำงานทีละ Step

ในรูปที่ 4.2 แสดงฟังก์ชั่นต่างๆ ในโหมดของการทำงานใน Manual Mode

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



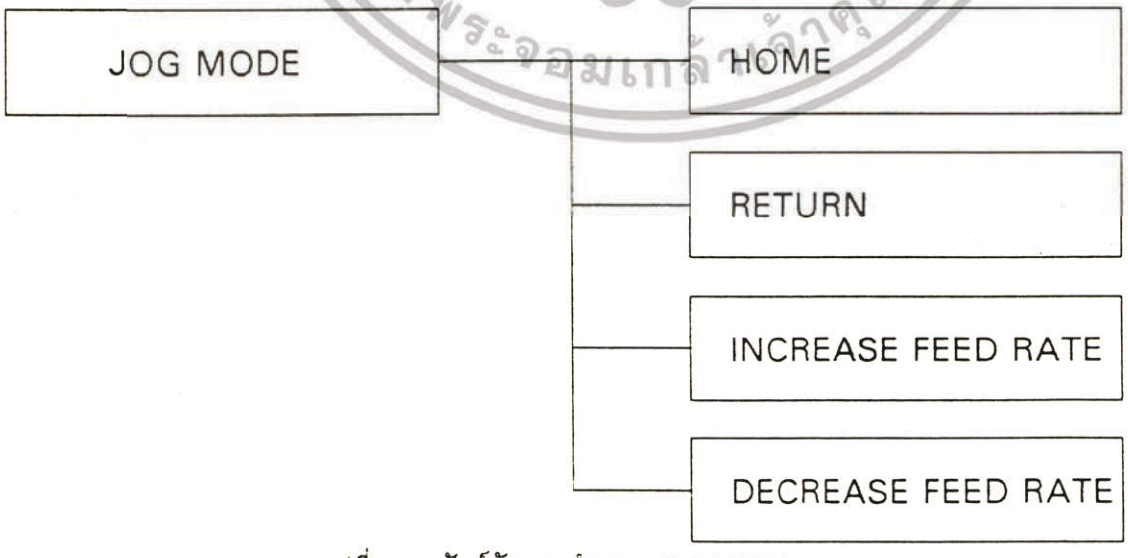

รูปที่ 4.2 ฟังก์ชันการทำงาน MANUAL MODE

**4.2.3 โหมดแบบช่วง (Jog mode)**

ในโหมดดังกล่าวจะเป็นการเคลื่อนที่ของเครื่องจักรในแบบธรรมดา โดยอาศัยกดปุ่ม (Keyboard) Handwheel หรือปุ่มกดภายนอกที่เพิ่มเข้าไป และสามารถสั่งการเคลื่อนที่ได้ทีละแกน ในโหมดนี้ประกอบด้วยฟังก์ชั่นต่างๆ ดังนี้

- กลับสู่ตำแหน่งเริ่มต้น (Home) มอเตอร์ทั้ง 3 แกนขึ้นกับการเลือกให้เคลื่อนที่กับไปที่ตำแหน่ง Home
- กลับไปที่ตำแหน่งก่อนใช้โหมด Job (Return) มอเตอร์หมุนกลับไปที่ตำแหน่งเดิมเข้าโหมด Jog
- เพิ่มความเร็วการเคลื่อนที่ (Increase Feed rate)
- ลดความเร็วการเคลื่อนที่ (Decrease Feed rate)

ในรูปที่ 4.3 แสดงฟังก์ชั่นต่างๆ ใน Jog mode




รูปที่ 4.3 ฟังก์ชันการทำงาน JOG MODE

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

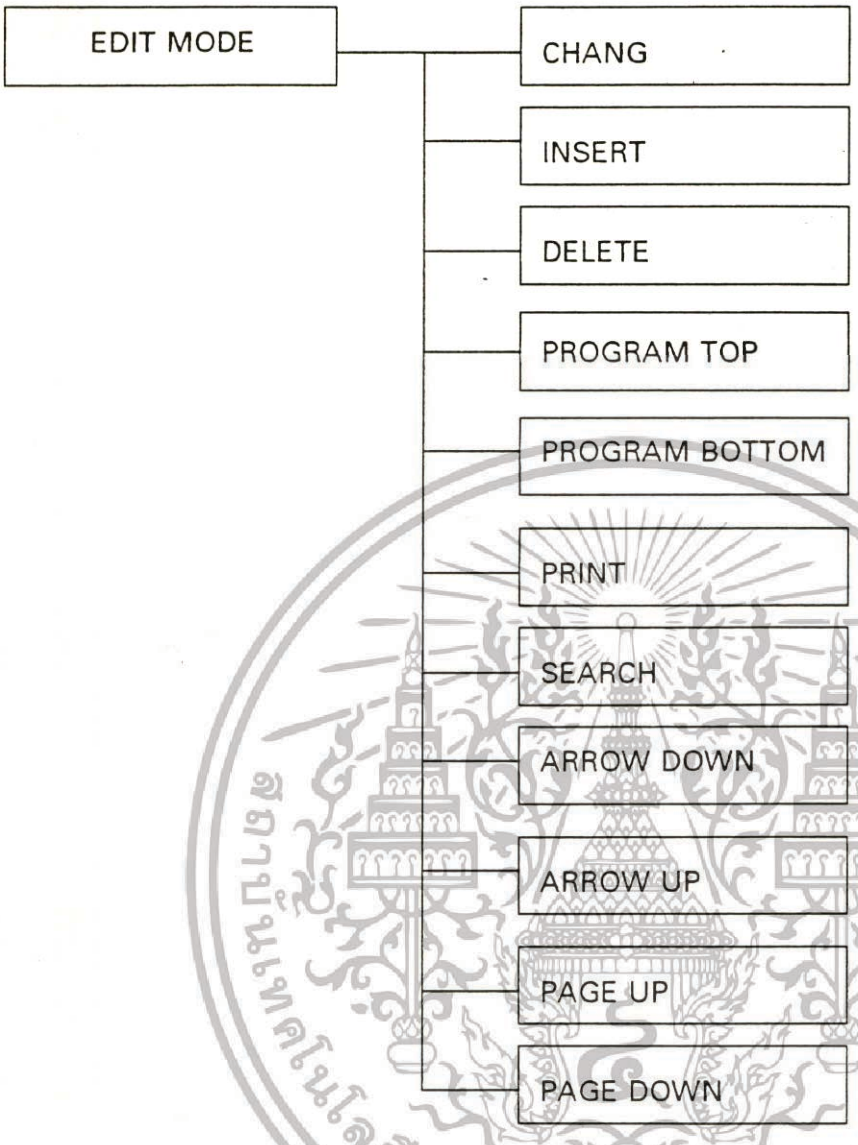
### 4.2.4 โหมดการเขียนและแก้ไขโปรแกรม (Edit mode)

ในโหมดดังกล่าวเป็นส่วนของการเขียนโปรแกรมเพื่อควบคุมเครื่องจักรกลให้ทำงานตามที่ต้องการ โดยการเขียนโปรแกรมต้องอ้างอิงกับรหัสมาตรฐานสากลตามตารางที่ 4.1 (ISO-Code) ซึ่งมีฟังก์ชั่นการทำงานต่าง ๆ ดังนี้

- การเปลี่ยนแปลงข้อมูล (Chang) เป็นฟังก์ชั่นการแก้ไขโปรแกรมที่ตำแหน่งต่าง ๆ ที่ต้องการเปลี่ยนแปลงโปรแกรม
- การแทรกข้อมูล (Insert) เป็นฟังก์ชั่นในการเพิ่มโปรแกรมที่ตำแหน่งต่าง ๆ ที่ต้องการเพิ่มเข้าไป
- การลบข้อมูล (Delete) เป็นฟังก์ชั่นในการลบข้อมูลหรือบรรทัดที่ต้องการลบ
- เลื่อนตัวชี้ไปที่ตำแหน่งบรรทัดแรก (Program Top) เป็นฟังก์ชั่นในการเลื่อนตำแหน่งตัวชี้ไปที่บรรทัดแรก
- เลื่อนตัวชี้ไปที่ตำแหน่งบรรทัดสุดท้าย (Program Bottom) เป็นฟังก์ชั่นในการเลื่อนตำแหน่งตัวชี้ไปที่บรรทัดสุดท้าย
- การพิมพ์โปรแกรม (Print) เป็นฟังก์ชั่นการพิมพ์โปรแกรมที่เขียนออกทางเครื่องพิมพ์ โดยเลือกโปรแกรมที่จะพิมพ์ได้
- การค้นหาข้อมูล (Search) เป็นฟังก์ชั่นในการค้นหาหรือคำที่ต้องการตรวจสอบ
- การเลื่อนตัวชี้ไปหนึ่งบรรทัด (Arrow down) เป็นฟังก์ชั่นในการเลื่อนตัวชี้ไปหนึ่งบรรทัดเพื่อใช้ในการแก้ไขหรือตรวจสอบ
- การเลื่อนตัวชี้ขึ้นไปหนึ่งบรรทัด (A row up) เป็นฟังก์ชั่นในการเลื่อนตัวชี้ไปหนึ่งบรรทัดเพื่อใช้ในการแก้ไขหรือตรวจสอบ
- การเลื่อนข้อมูลไปหนึ่งบรรทัด (Page up) เป็นฟังก์ชั่นการเลื่อนข้อมูลลงไปครั้งละ 1 หน้า เพื่อใช้ในการแก้ไขหรือตรวจสอบ
- การเลื่อนข้อมูลขึ้นไปหนึ่งบรรทัด (Page down) เป็นฟังก์ชั่นการเลื่อนข้อมูลขึ้นไปครั้งละ 1 หน้า เพื่อใช้ในการแก้ไขหรือตรวจสอบ

ในรูปที่ 4.4 แสดงรายละเอียดของฟังก์ชั่นต่าง ๆ ใน Edit mode

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

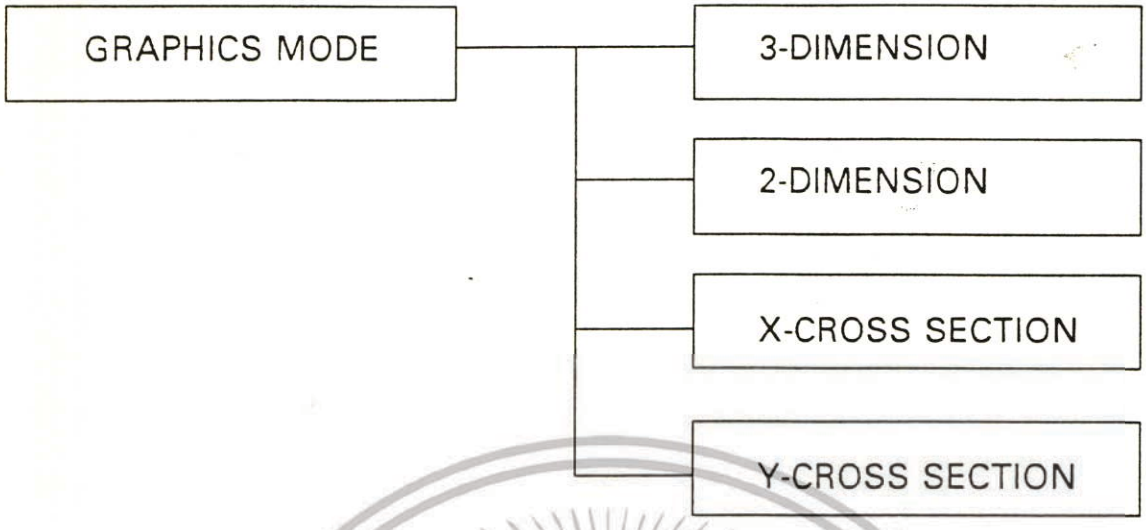



รูปที่ 4.4 ฟังก์ชันการทำงานของ EDIT MODE

## 4.3 การออกแบบโปรแกรมในส่วนของการจำลองการทำงานของโปรแกรม (Graphic Simulation)

ในส่วนของ Graphic Simulation เป็นอีกส่วนหนึ่งในองค์ประกอบของเครื่องกัดแนวดิ่งซีเอ็นซี เพื่อจำลองการกัดชิ้นงานบนคอมพิวเตอร์เพื่อตรวจสอบความถูกต้องอีกครั้งก่อนที่จะมีการกัดชิ้นงานจริง โดยมีรายละเอียดของฟังก์ชั่นต่างๆ ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้




รูปที่ 4.5 ฟังก์ชันการทำงานของ GRAPHICS MODE

ในส่วนของโปรแกรมดังกล่างอยู่ระหว่างการสร้างและพัฒนา โดยเริ่มจากส่วนของโปรแกรม DRIVER ที่เชื่อมต่อระหว่างไมโครคอมพิวเตอร์กับฮาร์ดแวร์ ส่วนควบคุม (MOTION CONTROLLER) ต่อจากนั้นจะเริ่มสร้างและพัฒนาส่วน Edit mode ตามไปด้วย และในการสร้างและพัฒนาดังกล่าวมีการทดลองตรวจสอบและแก้ไขตลอดเวลา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# บทที่ 5
# สรุปและข้อเสนอแนะ

รายงานฉบับนี้กล่าวถึง การสร้างต้นแบบส่วนควบคุมเครื่องกัดแนวดิ่งซีเอ็นซีระยะที่ 1 (Computer Numerical Control for Vertical Milling Machine Phase 1) การวิจัยและพัฒนาได้มีการสร้างและพัฒนาหลายๆ ส่วนประกอบกัน เช่น ส่วนเครื่องกัดแนวดิ่งซีเอ็นซี ส่วนไมโครคอมพิวเตอร์ทั้งฮาร์ดแวร์และซอฟท์แวร์และส่วน Motion controller card โดยอาศัยเทคโนโลยีระบบควบคุม (Control Engineering) ในระยะที่ 1 ของโครงการดังกล่าวได้สร้างและพัฒนาในส่วน Motion controller card และส่วนเครื่องกัดแนวดิ่งซีเอ็นซีได้เป็นผลสำเร็จภายใต้ข้อจำกัดทางด้านคอมพิวเตอร์ในการควบคุมและเครื่องจักรกล ดังนั้นสามารถสรุปและเสนอแนะวิธีในการแก้ปัญหาของการสร้างและพัฒนาดังนี้

## 5.1 ด้านเครื่องจักรกล

ในส่วนเครื่องจักรกลปัญหาส่วนใหญ่อยู่ที่การสร้างหัวกัด (Spindle Head) เพราะต้องอาศัยความละเอียดสูง เพราะเมื่อขณะกัดชิ้นงานจะไม่มีการแกว่งหรือ Vibration ของมีดกัด เพื่อให้ได้แนวกัดที่ถูกต้อง ดังนั้นแนวทางในการแก้ปัญหาอาจจะต้องพิถีพิถันในการสร้างหรือจัดซื้อหัว Spindle จากต่างประเทศ

## 5.2 ด้านส่วนควบคุม (Motion Controller Card)

ในส่วนดังกล่าวมีรายละเอียดซับซ้อนเพราะประกอบด้วยชิ้นส่วนต่างๆ บนระบบดังกล่าวเช่น โครงสร้างของภาค Digital Signal Processing ภาคหน่วยความจำ RAM และ ROM ภาคส่วนรับข้อมูลจากการเคลื่อนที่ของตำแหน่ง (Encoder feedback) ภาคการเปลี่ยนสัญญาณ Analog to Digital และ Digital to Analog Converter และภาคอินพุทและเอาท์พุท (PLC Input and PLC Output) ในระยะที่ 1 ยังมีปัญหาติดขัดด้าน Digital Signal Processing ในส่วนเรื่อง Compiler เพราะปัจจุบันยังใช้ Assembly compiler DSP ของ TI เบอร์ TMS32010 ในการพัฒนาแนวทางแก้ไขปัญหากำลังอยู่ในระหว่างการติดต่อของซื้อจาก TI ในส่วนของภาษาซี (C Compiler) สำหรับ TMS320C25 และ TMS320C30 หรืออาจมีการเปลี่ยนแปลงส่วนของ DSP ไปใช้ไอซีสำเร็จรูปด้าน MOTION CONTROL เช่น ของ Hewlett Packard หรือ National Semiconductor และอีกส่วนที่ยังติดปัญหาคือการติดต่อของซื้ออุปกรณ์ไปซีบางอย่างจากต่างประเทศ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# เอกสารอ้างอิง


1. M.E. Mortenson, "Computer Graphics an Introduction to the mathematics and Geometry" Industrial Press Inc., New York, 1989.

2. Yasuhiko Dote, "Servo Motor and Motion Control Using Digital Signal Processor",Texas Instruments, New Jersey, 1990.

3. Chan S. Park, "Interactive Micro computer Graphics", Addison-Weslay Publishing Company, 1985.

4. Chao-Hwa Chang, "NC Machine programming and Software design", Prentice-Hall International editions, 1989.

5. C.B. Besant "Computer-Aided Design and manufacture", Ellis horwood limited, 1980.

6. V RAMAMURTI "Computer Aided design in mechanical engineering", Tata McGraw-Hill Publishing company limited, 1987.

7. Panos Papamichalis "Digital Signal Processing Applications with the TMS320 Family Volume 3", Prentice Hall, New Jersey, 1990.



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 5.3 ด้านส่วนคอมพิวเตอร์ (Microcomputer)

ในส่วนดังกล่าวปัจจุบันต้องมีการพัฒนาส่วนของโปรแกรม DSP และส่วนกราฟฟิกส์ ดังนั้นต้องมีการพัฒนาและแก้ไขตลอดเวลาการ Compi Program ต้องใช้เวลามากในแต่ละครั้ง และส่วนของหน่วยความจำคอมพิวเตอร์มีฮาร์ดดิสก์ 40MB ยังไม่เพียงพอต่อความต้องการ ต้องเพิ่มความเร็วในการ Compi โดยเปลี่ยนแปลงด้านคอมพิวเตอร์และฮาร์ดดิสก์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# ภาคผนวก

1. รายละเอียดของโปรแกรมคอมพิวเตอร์ในการสร้างและพัฒนา

   1.1 โปรแกรมคอมพิวเตอร์บางส่วนของ DRIVER

```
/*
        MODULE :UTILITY.C

        UTILITY ROUTINE FOR DSP SOFTWARE

        6/4/93      BY KAWIN SONTHIPERMPOON
*/

#include <string.h>
#include <stdio.h>
#include <stdlib.h>
#include <alloc.h>
#include <fcntl.h>
#include <io.h>
#include <stdarg.h>
#include <conio.h>
#include <ctype.h>
#include <dos.h>
#include <dir.h>
#include <graphics.h>
#include "asynch_1.h"
#include "keys.h"
#include "pmaccomm.h"
#include "configur.h"
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
#include "command.h"
#include "m2.h"


/* ---- Remote & screen printf ------ */
int bprintf(const char *fmt,...)
{
        va_list argptr;
        int cnt;
        char buffer[256];

        va_start(argptr,fmt);
        vsprintf(buffer,fmt,argptr);
        va_end(argptr);

        cprintf(buffer);
        if(cfg.remote && (status.mode == command))
                wrtbk_a1(cfg.comport,255,'\0',buffer,&cnt);

        return(cnt);
}


int DisplayMessage(char *msg)
{
        gotoxy(1,MSGLINE1);
        cprintf("%s",msg);
        if(cfg.remote && (status.mode == command))
                comprintf(cfg.comport,"\n\r%s",msg);
        return 0;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
}

int ClearMessage(void)
{
        gotoxy( 1,MSGLINE1);
        gclreol();

        return 0;
}

void gclreol(void)      // graphics cleareol
{
        int ll,xx,yy;

        xx = wherex();
        yy = wherey();

        for(ll = xx; ll<=80; ll++)
                putchar(' ');
        gotoxy(xx,yy);
}

void beep(void)
{
   sound(1000);
   delay(50);
   nosound();
}

void click(void)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
{
        sound(700);
        delay(5);
        nosound();
}

/* ---- Set all home flags to zero ---------- */
void zero_home_flags(void)
{
        int ll;

        for(ll=0; ll<NUM_MOTORS; ll++) {
                home[ll] = 0;
        }
}

/* ---- Check to see if all axis have homed ---- */
int all_homed(void)
{
        int ll;

        for(ll=0; ll<NUM_MOTORS; ll++) {
                if(cfg.axis_enable[ll] && !home[ll] && !cfg.slave[ll])
                        return 0;
        }
        return 1;
}

void toggle_pen(void)
{
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        char str[5];

        if(pen==0)
                pen = 1;
        else
                pen = 0;
        sprintf(str,"M1=%d",pen);
        sendline(str);
}

char *current_directory(char *path)
{
   strcpy(path, "X:\\");     /* fill string with form of response: X:\ */
   path[0] = 'A' + getdisk();    /* replace X with current drive letter */
   getcurdir(0, path+3);  /* fill rest of string with current directory */
   if(path[3] == '\0') path[2] = '\0';
   return(path);
}

void parsefilename(char *out,char *s,char *infile,char *inext)
{
        static char path[MAXPATH];
        static char drive[MAXDRIVE];
        static char dir[MAXDIR];
        static char file[MAXFILE];
        static char ext[MAXEXT];

        strcpy(path,s);
        strcat(path,"\\");
        fnsplit(path,drive,dir,file,ext);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        strcpy(file,infile);
        strcpy(ext,inext);
        fnmerge(out,drive,dir,file,ext);
}

int getstr(char *source,char *dest)
{
        int cnt = 0, done = 0;
        char *dp, *sp;

        dp = dest;
        sp = source;
        while(!done) {
                if(*sp == '\n' || *sp == '\0' || cnt >= MAXLINE) {
                        *dp++ = '\0';
                        done = 1;
                        cnt++;
                }
                else if (*sp == '\r') {
                        ++sp;
                        cnt++;
                }
                else {
                        *dp++ = *sp++;
                        cnt++;
                }
        };

        return cnt;
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
int fgetstr(FILE *fp,char *dest)
{
        int cnt = 0, done = 0;
        char ch, *dp;

        dp = dest;

        while(!done) {
                ch = fgetc(fp);
                if(ch == '\n' || feof(fp) || cnt >= MAXLINE) {
                        *dp = '\0';
                        done = 1;
                }
                else if(isprint(ch)) {
                        *dp++ = ch;
                        ++cnt;
                }
        };

        return cnt;
}

/* ------ Display Program point ---------- */
void prog_point_display(void)
{
        gotoxy(67,1);
        textcolor(high1_color);
        cprintf("O");
        cprintf(cfg.o_format,current_program);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        cprintf(" N");
        cprintf(cfg.n_format,command_block[current]->n);
}

/* ——— Display Program ———— */
void displayprog(void)
{
        int ll,jj,kk,tmp, next;
        char *cp;
        static struct text_info ww;

        gettextinfo(&ww);
        window(1,10,80,19);

        textcolor(high2_color);
        next = current + 1;

        cprintf("F%-6.2f                    S%-5d",command_block[current]-
>f,command_block[current]->s);
        if(command_block[current]->g[1] > 99) {
                tmp = command_block[current]->g[1] % 100;
                cprintf("G%02d.%d      ",tmp,(command_block[current]->g[1]      -
tmp)/100);
        }
        else
                cprintf(" G%02d  ",command_block[current]->g[1]);
        cprintf(" G%02d G%02d",command_block[current]->g[2],
                command_block[current]->g[3]);
        cprintf(" G%02d G%02d G%02d",command_block[current]->g[5],
                command_block[current]->g[6],command_block[current]->g[8]);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        cprintf("       G%02d       G%02d       \n\r",command_block[current]-
>g[7],command_block[current]->g[14]);
        cprintf("           D%02d H%02d\n\r",
                command_block[current]->d,command_block[current]->h);

        for ( ll = 0,jj=4; ll < NUM_BUF_BLOCKS; ll++,jj++) {
                cp = command_str[ll];
                kk = 0;
                gotoxy(1,jj);
                putch(' ');
                if(ll == next) {
                        textcolor(high1_color);
                        while(*cp > 31 && *cp < 127 && kk < 77) {
                                putch(*cp++);
                                ++kk;
                        }
                        while(kk < 77) {
                                putch(' ');
                                ++kk;
                        }
                        textcolor(norm1_color);
                }
                else {
                        while(*cp > 31 && *cp < 127 && kk < 77) {
                                putch(*cp++);
                                ++kk;
                        }
                        while(kk < 77) {
                                putch(' ');
                                ++kk;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                }
            }
        }

        textcolor(high1_color);
        gotoxy(1,5+current);
        putch('>');

        window(67,1,80,1);
        cprintf("O");
        cprintf(cfg.o_format,current_program);
        cprintf(" N");
        cprintf(cfg.n_format,command_block[current]->n);

        textattr(ww.attribute);
        window(ww.winleft,ww.wintop,ww.winright,ww.winbottom);
}

/* ── Send position string to Pmac ──────── */
int set_position(char ax, float num)
{
        static char str[26], str1[12];

        if(strchr("XYZUVWABC",ax) == NULL) {
                return 0;
        }
        sprintf(str,"%c=",ax);
        sprintf(str1,cfg.linear_format,num);
        strcat(str,str1);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        sendline(str);

        return 1;
}

/* — Set rapid rate and Override ———— */
int set_rapid(void)
{
        int ll;
        float cts_msec;

        for (ll=0; ll<NUM_MOTORS; ll++) {
                if(cfg.axis_enable[ll]) {
                        cts_msec = cfg.unit_count[ll] * cfg.motor_rapid[ll] /
60000.0;
                        cts_msec *= (rapid_override * 0.01);
                        pprintf("l%d16=%.2f",ll+1,cts_msec); // set speed limits
                }
        }
        return 1;
}

/* — Clear waiting flag and display ——— */
void clearwaiting(void)
{
        if(waiting) {
                gotoxy(1,MSGLINE1);
                cprintf("          ");
                gotoxy(1,COMLINE-2);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                gclreol();
                waiting = 0;                    // set not waiting
        }
}

/* — Save current program to program directory — */
Int save_program(void)
{
        statlc char str[81];

        If((program[current_program]->handle = open(program[current_program]-
>name,O_BINARY | O_WRONLY)) == -1) {
                sprintf(str,"Unable to open program file
%s..",program[current_program]->name);
                setcommanderror(str);
                return 0;
        }
        If (write(program[current_program]->handle,
                        program[current_program]->base, program[current_program]-
>size) == -1L) {
                close(program[current_program]->handle);
                sprintf(str,"Unable to write file..");
                setcommanderror(str);
                return 0;
        }

        close(program[current_program]->handle);

        return 1;
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
/* — Free program pointer and buffer ——— */
int freeprogram(int pnum)
{
        if(pnum == current_program)
                current_program = 0;

        if(farheapchecknode(program[pnum]->base) == _USEDENTRY) {
                farfree(program[pnum]->base);
                program[pnum]->base = NULL;
        }
        if(farheapchecknode(program[pnum]) == _USEDENTRY) {
                farfree(program[pnum]);
                program[pnum] = NULL;
        }
        return 1;
}

/* — Check to see if a file exists ——— */
int file_exists(char *filename)
{
  return (access(filename, 0) == 0);
}

/* — Free all program pointer & buffers —— */
int freeallprograms(void)
{
        int ii;

        for(ii=1; ii< MAXPROGRAMS; ii++) {
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            if(!freeprogram(ll))
                return 0;
        }
        if(farheapcheck() == _HEAPCORRUPT) {
            setcommanderror("Memory heap corrupted in freeallprograms...");
            return 0;
        }
        return 1;
}

int memcheck( void )
{
  struct farheapinfo hi;
  int entries = 0;
  unsigned long tsz = 0L;

  hi.ptr = NULL;
  fprintf( stdprn,"  Size  Status  Address\n" );
  fprintf( stdprn,"  ----  ------  -------\n" );
  while( farheapwalk( &hi ) == _HEAPOK ) {
      ++entries;
      if(hi.in_use) {
        fprintf(stdprn, "%7u    Used    %p\n", hi.size,hi.ptr);
        tsz += hi.size;
      }
      else
        fprintf(stdprn, "%7u    Free    %p\n", hi.size,hi.ptr);
  }
      fprintf(stdprn, "\n%7u    Total Used\n", tsz);
      fprintf(stdprn, "%7d    Entries\n", entries);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        fprintf(stdprn, "%7lu    Core left\n",farcoreleft());

    return 1;
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.2 โปรแกรมคอมพิวเตอร์บางส่วนของ ISO-CODE

```
CLOSE DELETE TRACE  DELETE GATHER
OPEN PROG 1000      ; G-CODE SUBROUTINE PROGRAM
CLEAR
RAPID RET           ; G00
N1000 LINEAR RET    ; G01
N2000 NORMAL J-1    ; G02 (ALWAYS ZX PLANE)
CIRCLE1 RET         ; CLOCKWISE ARCS
N3000 NORMAL J-1    ; G03 (ALWAYS ZX PLANE)
CIRCLE2 RET         ; COUNTERCLOCKWISE ARCS
N4000 READ(D)       ; G04 - D IS DWELL TIME IN SECS
DWELL(Q104*1000)    ; D VALUE PLACED IN Q104
RET
; Note: It is not clear exactly what needs to be done here.
N33000 READ(X,Z,I,K)  ; G33 - GET ARGUMENTS
IF (Q100 & 8388608 > 0) X(Q124) ; MOVE TO STARTING DIAMETER
FRAX(Z)             ; ONLY Z-AXIS USED IN FEEDRATE CALCS
F(P97*Q111)         ; Z FEEDRATE = (REV/MIN)*(IN/REV)
INC(X,Z)            ; THREAD MOVE SPECIFIED INCREMENTAL
IF (Q100 & 65536 = 0) Q117=0   ; NO Q VALUE IS ZERO TAPER ANGLE
Z(Q126)X(Q126*TAN(Q117))       ; MAKE THREAD MOVE
IF (M90=1) ABS      ; RESTORE ABSOLUTE IF APPROPRIATE
FRAX(X,Z)           ; BOTH AXES USED IN FEEDRATE CALCS
RET
N70000              ; G70 - INCH MODE
M191=25400          ; X AXIS SCALE FACTOR 25400 CTS/IN
M393=25400          ; Z AXIS SCALE FACTOR 25400 CTS/IN
GOSUB 100000        ; SUBROUTINE TO RE-ALIGN COORDINATE SYS
M70=0               ; FLAG TO NOTE NOT METRIC
RET
N71000              ; G71 - MILLIMETER MODE
M191=1000           ; X AXIS SCALE FACTOR 1000 CTS/MM
M393=1000           ; Z AXIS SCALE FACTOR 1000 CTS/MM
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
GOSUB 100000          ; SUBROUTINE TO RE-ALIGN COORDINATE SYS
M70=1                 ; FLAG TO NOTE METRIC
RET
N90000 ABS            ; G90 - ABSOLUTE MODE
M90=1                 ; FLAG FOR MODE
RET
N91000 INC            ; G91 - INCREMENTAL MODE
M90=0                 ; CLEAR ABSOLUTE FLAG
RET
N92000                ; G92 - AXIS PRESET
READ(X,Z,S,R)         ; READ POSSIBLE ARGUMENTS
IF (Q100 & 8388608 > 0)     ; X ARGUMENT PASSED IN Q124?
 Q224=Q224+Q124-M165        ; NEW X POSITION OFFSET
 PSET X(Q124)               ; DO PRESET
ENDIF
IF (Q100 & 33554432 > 0)    ; Z ARGUMENT PASSED IN Q124?
 Q224=Q224+Q124-M165        ; NEW Z POSITION OFFSET
 PSET Z(Q126)
ENDIF
IF (Q100 & 262144 > 0) P92=Q119      ; STORE S VALUE
IF (Q100 & 131072 > 0) P98=M165-Q118 ; STORE XPOS-RVALUE
RET
N94000 I190=60000     ; G94 - FEEDRATE IS PER MINUTE
I193=2054             ; USE INTERNAL TIME BASE
RET
N95000 I193=10        ; G95 - FEEDRATE PER REV (NOM. 10 MSEC)
I193=1833             ; USE FREQUENCY TIME BASE
RET
N96000 READ(S)        ; G96 - SPINDLE SURFACE SPEED
P96=Q119              ; STORE SPINDLE SPEED
M96=1                 ; FLAG TO MARK CSS MODE
RET
N97000 READ(S)        ; G97 - SPINDLE ANGULAR SPEED
IF (Q100 & 262144 > 0) P97=Q119   ; STORE SPINDLE SPEED
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
M96=0                ; TO MARK NOT IN CSS MODE
RET
N99000               ; G99 - CANCEL PRESET
PSET X(M165+Q224)Z(M365+Q226)    ; ELIMINATE OFFSETS
Q224=0               ; NO MORE X OFFSET
Q226=0               ; NO MORE Z OFFSET
RET

N100000              ; COORD SYS RE-ALIGN ROUTINE
P163=M163/(I108*32)  ; SCALE MOTOR 1 POS INTO COUNTS
M165=(P163-M194)/M191  ; CHANGE X AXIS POSITION
P363=M363/(I308*32)  ; SCALE MOTOR 3 POS INTO COUNTS
M365=(P363-M394)/M391  ; CHANGE Z AXIS POSITION
RET

CLOSE
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.3 โปรแกรมคอมพิวเตอร์บางส่วนของ EDIT MODE

```
/*
        MODULE : EDIT.C

        EDITOR FOR MILLING.C

        06/24/93    BY KAWIN   SONTHIPERMPOON
*/

#define FIRSTLINE           3
#define LASTLINE            COMLINE -3
#define LINELENGTH          81

#include <stdio.h>
#include <conio.h>
#include <alloc.h>
#include <string.h>
#include <ctype.h>
#include "keys.h"
#include "block.h"
#include "ptimer.h"
#include "command.h"
#include "m2.h"

static int updatedisplay;       // flag for display update
static char *edit_line;                 // index into lines for edit
static char *top_page;
static char *bottom_page;
static char *top_program;
static char *bottom_program;
static char str[MAXLINE];               // general purpose string
static BLOCK block;                             // general purpose block
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
int  readpage(void);
int  writepage(void);
void clearpage(void);
void displaypage(void);
char *displayline(int xx,int yy,char *cp);
int  pageup(void);
int  pagedown(void);
int  lineup(void);                          // moves up one line
int  linedown(void);                        // move down one line
int  filetop(void);                         // position at file top
int  filebottom(void);                      // position at last line
int  alterline(void);          // alter current line
int  program_change(BLOCK *blk);    // check for 0 address change
int  edit_search(void);                     // moves pointer to search string
int  insertline(char *str);                 // inserts a new line
int  deleteline(void);                      // deletes current line
char *nextline(char *cp);                   // moves pointer down one line
char *lastline(char *cp);                   // moves pointer up one line
int  adjustalloc(unsigned adjust);     // reallocate up or down

void DoEditKeyboard(void)
{
        switch(getcommand()) {
                case EMPTY:                 // no character
                        break;
                case ESCAPE:                // first char a escape key
                        save_program();
                        DoCommandInit();
                        break;
                case RETURN:                // first char a return key
                        break;
                case FIRSTCHAR:                 // first single character
                        break;
                case NEXTCHAR:                  // character - not first
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                break;
            case COMMAND:                       // string command
                DoEditCommand();
                break;
            case FUNCTION:                      // function command
                DoEditFunction();
                break;
            case STRING:                // not a keyword
            case INTEGER:               // a integer number
            case FLOAT:                         // a real number
                setcommanderror("Unknown command..");
                break;
        }
}

int DoEditFunction(void)
{
        switch(funct) {
            case FN1:           // CHANGE
                alterline();
                break;
            case FN2:           // INSERT
                if(getstring("Insert line",str)) {
                    insertline(str);
                }
                break;
            case FN3:           // DELETE
                deleteline();
                break;
            case FN4:           // FILE TOP
                filetop();
                break;
            case FN5:           // FILE BOTTOM
                filebottom();
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                break;
        case FN6:               // SEARCH
                edit_search();
                break;
        case FN7:               // PRINT
                print();
                break;
        case UP:                // LINE UP
                lineup();
                break;
        case DOWN:              // LINE DOWN
                linedown();
                break;
        case PG_UP:             // PAGE UP
                pageup();
                break;
        case PG_DOWN:           // PAGE DOWN
                pagedown();
                break;
        default:
                beep();
                return 0;
    }
    return 1;
}

int DoEditCommand(void)
{
    switch(parameter[0].num) {
        case HELP:
                help(status.mode);
                break;
        default:
                setcommanderror("Command not available under Edit Mode..");
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                    break;
            }

            return 1;
}

void DoEditInit(void)
{
            top_program = edit_line = top_page = program[current_program]->base;
            bottom_program = top_program + program[current_program]->size;
            _setcursortype(_NOCURSOR);
            stoptimer();
            status.mode = edit;
            DisplayFunctions(status.mode);

            display  = DoEditDisplay;
            action   = NULL;
            keyboard = DoEditKeyboard;

            updatedisplay = 1;
            window(1,1,80,25);
}

void DoEditDisplay(void)
{
            if(updatedisplay) {
                        displaypage();
                        updatedisplay = 0;
            }
}

int readpage(void)
{
            return 1;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
}

int writepage(void)
{
        return 1;
}

void clearpage(void)
{
        int ii;

        for(ii=FIRSTLINE; ii<=LASTLINE; ii++) {
                gotoxy(1,ii);
                clreol();
        }
}

void displaypage(void)
{
        int ii;
        char *cp;

        for(ii=FIRSTLINE,cp=top_page; ii<=LASTLINE; ii++) {
                cp = displayline(1,ii,cp);
        }
        bottom_page = cp;
}

char *displayline(int xx, int yy, char *cp)
{
        char str[MAXLINE];

        textcolor(norm1_color);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        gotoxy(xx,yy);
        if(cp >= bottom_program || cp == NULL) {
                clreol();
                return bottom_program;
        }
        if(cp == edit_line) {
                textcolor(high1_color);
                cp += getstr(cp,str);
                cprintf("%c%s",0x1a,str);
                textcolor(norm1_color);
        }
        else {
                cp += getstr(cp,str);
                cprintf(" %s",str);
        }
        clreol();

        return cp;
}

int pageup(void)
{
        int ii;

        ii = LASTLINE;
        while(ii > FIRSTLINE && top_page > top_program) {
                top_page = lastline(top_page);
                edit_line = lastline(edit_line);
                --ii;
        }
        updatedisplay = 1;
        return 1;
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
int pagedown(void)
{
        int ii;

        if(bottom_page >= bottom_program)
                return 0;
        ii = FIRSTLINE;
        while(ii < LASTLINE && top_page < bottom_program) {
                top_page = nextline(top_page);
                edit_line = nextline(edit_line);
                ++ii;
        }
        updatedisplay = 1;
        return 1;
}

int lineup(void)
{
        edit_line = lastline(edit_line);
        if(edit_line < top_page)
                top_page = edit_line;
        updatedisplay = 1;
        return 1;
}

int linedown(void)
{
        edit_line = nextline(edit_line);
        if(edit_line >= bottom_page && bottom_page < bottom_program)
                top_page = nextline(top_page);
        updatedisplay = 1;
        return 1;
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
int filetop(void)
{
        top_page = edit_line = top_program;
        updatedisplay = 1;
        return 1;
}

int filebottom(void)
{
        top_page = bottom_program;
        pageup();
        edit_line = bottom_program;
        linedown();
        updatedisplay = 1;
        return 1;
}

int alterline(void)
{
        static char *cp, *sp;
        static char instr[MAXLINE],outstr[MAXLINE];
        static int change = 0;

        for(cp=edit_line,sp=outstr; *cp!='\n'; cp++) {
                *sp++ = *cp;
        }
        *sp = '\0';

        block = default_block;
        gotoxy( 1,COMLINE - 1);
        cprintf("%s",outstr);
        while(getstring("Change line",instr)) {
                change++;
                strcat(outstr,instr);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                line_to_block(outstr,&block);
                program_change(&block);
                block_to_line(outstr,&block);
                gotoxy( 1,COMLINE - 1);
                cprintf("%s",outstr);
                clreol();
        }
        if(change) {
                deleteline();
                insertline(outstr);
        }
        gotoxy( 1,COMLINE - 1);
        clreol();

        return change;
}

int program_change(BLOCK *blk)
{
        if(blk->flag.change[o]) {
                blk->flag.change[o] = 0;
                return 1;
        }
        return 0;
}

int edit_search(void)
{
        static char *sp;
        static char instr[MAXLINE];

        if(getstring("Search for",instr)) {
                if((sp = strstr(edit_line,instr)) != NULL) {
                        while(*sp != '\n' && sp > top_program)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                        --sp;
                    if(*sp == '\n')
                        ++sp;
                    edit_line = sp;
                    if(edit_line >= bottom_page && bottom_page < bottom_program)
                        top_page = edit_line;
                }
        }
        updatedisplay = 1;
        return 1;
}

int insertline(char *str)
{
        static int expand;
        static char *cp, *sp;
        static char instr[MAXLINE];

        line_to_block(str,&block);
        program_change(&block);
        block_to_line(str,&block);
        expand = sprintf(instr,"%s\n",str);

        if(!adjustalloc(expand)) {
                setcommanderror("Not enough memory to insert line..");
                return 0;
        }

        cp = program[current_program]->base + program[current_program]->size;
        while(cp > edit_line) {
                *cp = *(cp - expand);
                --cp;
        }
        sp = edit_line;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        cp = instr;
        while(*cp != '\0') {
                *sp++ = *cp++;
        }

        updatedisplay = 1;
        return 1;
}

int deleteline(void)
{
        int length;
        char *cp, *sp;

        block = default_block;
        length = (line_to_block(edit_line,&block) - edit_line);
        if(program_change(&block))
                return 0;

        for(cp=edit_line,sp=(edit_line+length); *cp!='\0'; cp++,sp++) {
                *cp = *sp;
        }
        *cp = '\0';
        if(*edit_line == '\0')
                edit_line = lastline(edit_line);

        updatedisplay = 1;
        if(!adjustalloc(-length)) {
                setcommanderror("Allocation fault in delete line..");
                return 0;
        }
        return 1;
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
char *nextline(char *cp)
{
        char *sp;

        sp = cp;
        while(*sp != '\n' && sp < bottom_program) {
                ++sp;
        }
        if(sp >= bottom_program)
                return lastline(sp);

        return ++sp;
}

char *lastline(char *cp)
{
        char *sp;

        sp = cp;
        --sp;
        if(*sp == '\n')
                --sp;
        while(*sp != '\n' && sp >= top_program) {
                --sp;
        };

        return ++sp;

}

int remove_address(char *sp, char addr)
{
        char *cp;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
edit_line      = top_program + editoffset;
bottom_program = top_program + program[current_program]->size - 2;

return 1;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้