สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

ระบบการติดตามและบันทึกลายเส้นด้วยระบบไมโครคอมพิวเตอร์ สำหรับเครื่องตัดสมัยใหม่

MICROCOMPUTER-BASE PATTERN TRACKING AND RECORDING SYSTEM FOR MODERN CUTTING MACHINE

หนังสืออ้างอิง
ห้ามนำออกนอกห้องสมุด

นายสุมิตร พนาอุดมทรัพย์
MR. SUMIT PANAUDOMSUP

T021207

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต
สาขาวิศวกรรมไฟฟ้า
บัณฑิตวิทยาลัย
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
พ.ศ. 2536

ISBN 974-621-107-2

ลิขสิทธิ์ของบัณฑิตวิทยาลัย สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เลขหมู่
เลขทะเบียน 21207
วัน, เดือน, ปี 3 ส.ค. 2537

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

MICROCOMPUTER-BASE PATTERN TRACKING AND RECORDING SYSTEM FOR MODERN CUTTING MACHINE

MR. SUMIT PANAUDOMSUP

A THESIS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE
MASTER OF ENGINEERING IN ELECTRICAL ENGINEERING
GRADUATE SCHOOL
KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG
1993

ISBN 974-621-107-2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อวิทยานิพนธ์	ระบบการติดตามและบันทึกลายเส้นด้วยระบบไมโครคอมพิวเตอร์สำหรับเครื่องตัดสมัยใหม่
นักศึกษา	นายสุมิตร พนาอุดมทรัพย์
อาจารย์ผู้ควบคุมวิทยานิพนธ์	รศ.ดร.โยธิน เปรมปุราณีรัชต์
ระดับการศึกษา	วิศวกรรมศาสตร์มหาบัณฑิต สาขาวิศวกรรมไฟฟ้า
ภาควิชา	วิศวกรรมระบบควบคุม สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
พ.ศ.	2536

บทคัดย่อ

ในปัจจุบันการตัดชิ้นงานมีความต้องการความแม่นยำและคุณภาพที่แน่นอน ซึ่งกระบวนการตัดดังกล่าวจำเป็นต้องควบคุมด้วยไมโครคอมพิวเตอร์ แต่การใช้งานเครื่องตัดที่ควบคุมด้วยไมโครคอมพิวเตอร์จำเป็นต้องอาศัยนักเขียนโปรแกรมซึ่งเป็นภาระต่อผู้ใช้ซึ่งมักจะไม่รู้เรื่องโปรแกรม งานวิจัยนี้ได้เสนอระบบการติดตามลายเส้นและบันทึกด้วยไมโครคอมพิวเตอร์สำหรับเครื่องตัดสมัยใหม่ เครื่องตัดนี้มีจุดประสงค์ให้ง่ายต่อการใช้งานโดยเพียงแต่ผู้ใช้เขียนรูปลงบนแผ่นกระดาษ แล้วปล่อยให้เครื่องทำการติดตามลายเส้นและบันทึก โดยอาศัยขั้นตอนทางคณิตศาสตร์เก็บอยู่ในรูปของข้อมูลสำหรับคอมพิวเตอร์ ซึ่งจะสามารถนำข้อมูลออกมาใช้กับงานตัดต่อไปได้ ซึ่งเครื่องนี้จะเป็นประโยชน์อย่างยิ่งสำหรับงานตัดในอุตสาหกรรมขนาดเล็กเพราะให้ต้นทุนการผลิตที่ต่ำกว่า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Thesis Title	Microcomputer-Base Pattern Tracking and Recording System for Modern Cutting Machine
Student	Mr.Sumit Panaudomsup
Thesis Advisor	Assoc.Prof.Dr.Yothin Prempraneerach
Level of Study	Master of Engineering in Electrical Engineering
Department	Control Engineering King Mongkut's Institute of Technology Ladgrabang
Year	1993

ABSTRACT

Nowadays, the materials cut in various shape needs a high accuracy and reliable quality, consequently a cutting process is to be controlled by microcomputer. The application of cutting machine controlled by microcomputer,the users or programmers are required to deal with it. Thus non-programmer meet with difficulty to get started. This research project presents a development of pattern tracking and recording system based on microcomputer for modern cutting machine. This cutting machine is easy to be operated and its cutting process will follow the set desired pattern which has firstly drawn on the paper by the user. The new technique is mathematical sequence of the cutting process which any designed pattern is converted to be data for storing in the microcomputer's memory. This data can be taken out for controlling the cutting machine in its process. The system therefore is very useful for small industrial material cutting work with low cost productions.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

วิทยานิพนธ์ฉบับนี้สำเร็จลงได้ด้วยดี ก็เพราะได้รับความเมตตาจาก รองศาสตราจารย์ ดร.โยธิน เปรมปราณีรัชต์ ที่ได้ให้ความกรุณาแนะนำแก่ผู้วิจัยตลอดมา ผู้วิจัยขอกราบขอบพระคุณเป็นอย่างสูงมา ณ. โอกาสนี้ด้วย และขอขอบคุณ คุณอนุตรชัย ณ ถลาง ที่ได้ให้กำลังใจ

เหนือสิ่งอื่นใดคือคุณแม่ที่ได้อุปการะผู้วิจัยโดยลำพังตลอดมา

สุมิตร พนาอุดมทรัพย์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

หน้า

บทคัดย่อภาษาไทย I
บทคัดย่อภาษาอังกฤษ II
กิตติกรรมประกาศ III
สารบัญ IV
สารบัญรูป VII
สารบัญตาราง XI
คำอธิบายสัญญลักษณ์/คำย่อ XII
บทที่ 1 บทนำ 1
1.1 วัตถุประสงค์ 1
1.2 ขอบเขตของงานวิจัย 1
1.3 ประโยชน์ของงานวิจัย 2
บทที่ 2 การเลือกชนิดหัวอ่านและการออกแบบวิธีการติดตามลายเส้น 4
2.1 การเลือกชนิดหัวอ่านให้เหมาะสมกับระบบ 4
2.1.1 หัวอ่านแบบเมกตริกซ์ 4
2.1.2 หัวอ่านแบบเซลเดียว 5
2.2 การออกแบบวิธีการติดตามลายเส้นเพื่อการตัดและการบันทึกที่ดี 6
2.2.1 ใช้ระบบควบคุมแบบตั้งค่าควบคุม PID ได้ 6
2.2.2 ใช้ระบบควบคุมแบบกึ่งตัดสินใจร่วมกับการควบคุมแบบ P 10
2.2.3 ใช้วิธีการติดตามลายเส้นโดยเลียนแบบการสะท้อนแสงร่วมกับการตัดสินใจทางโปรแกรม 11
บทที่ 3 การออกแบบและสร้างชิ้นส่วนต่าง ๆ ของชิ้นงานเพื่อให้เหมาะสมกับระบบ 13
3.1 การออกแบบและสร้างระบบกลไกขับเคลื่อนหัวอ่าน 13
3.2 การออกแบบและสร้างหัวอ่าน 14
3.3 การออกแบบและสร้างระบบอิเลคทรอนิกส์สำหรับขับเคลื่อนระบบกลไก 15
3.4 การออกแบบและสร้างวงจรอิเลคทรอนิกส์สำหรับเชื่อมโยงการติดต่อระหว่างระบบกับคอมพิวเตอร์ 24
3.5 สรุปตำแหน่งช่องผ่านข้อมูล(port address)ระหว่างคอมพิวเตอร์กับข้อมูลของอุปกรณ์ต่างๆ 25
บทที่ 4 การออกแบบขั้นตอนการทำงานของโปรแกรมคอมพิวเตอร์ 27

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.1 การออกแบบโปรแกรมย่อยควบคุมการเคลื่อนที่ของหัวอ่าน 27
4.1.1 สรุปคำสั่งสำหรับการเขียนโปรแกรมย่อยควบคุมการเคลื่อนที่ของหัวอ่าน 27
4.2 การออกแบบโปรแกรมติดตามลายเส้น 28
4.2.1 โปรแกรมติดตามลายเส้นระบบควบคุมแบบตั้งค่าควบคุม PID ได้ 28
4.2.2 โปรแกรมติดตามลายเส้นระบบควบคุมแบบกึ่งตัดสินใจร่วมกับการควบคุมแบบ P .. 32
4.2.3 โปรแกรมติดตามลายเส้นที่ใช้วิธีการวิ่งตามลายเส้นแบบสะท้อนแสงร่วมกับการตัดสินใจทางโปรแกรม 34
4.3 การออกแบบโปรแกรมบันทึกข้อมูลที่ได้จากการติดตามลายเส้น 37
4.3.1 โปรแกรมบันทึกข้อมูลที่ได้จากการติดตามลายเส้นระบบควบคุมแบบตั้งค่าควบคุม PID 37
4.3.2 โปรแกรมบันทึกข้อมูลที่ได้จากการติดตามลายเส้นระบบควบคุมแบบกึ่งตัดสินใจร่วมกับการควบคุมแบบ P 41
4.3.3 โปรแกรมบันทึกข้อมูลที่ได้จากการติดตามลายเส้นที่ใช้วิธีการวิ่งตามลายเส้นแบบ..... การสะท้อนของแสงร่วมกับการตัดสินใจทางโปรแกรม 45
4.4 การออกแบบโปรแกรมย่อยทำหน้าที่อ่านและเก็บข้อมูลจากจานเก็บข้อมูล(HARD DISK)..... 48
4.4.1 โปรแกรมอ่านข้อมูลจากจานเก็บข้อมูล 48
4.4.2 โปรแกรมเก็บข้อมูลลงจานเก็บข้อมูล 48
4.5 การออกแบบโปรแกรมหลักเพื่อรับคำสั่งจากผู้ใช้มาควบคุมโปรแกรมย่อยส่วนต่างๆ 51
4.5.1 การกำหนดขอบเขตเพื่อกำหนดเงื่อนไขและค่าเริ่มต้นสำหรับระบบ 51
4.5.2 การออกแบบโครงสร้างหรือขั้นตอนการทำงานของโปรแกรมหลัก 51
บทที่ 5 วิธีการใช้งานโปรแกรม 56
5.1 แสดงวิธีการใช้งานโปรแกรมหลัก 56
5.2 การใช้งานระบบการบันทึกลายเส้น 57
5.2.1 การใช้งานการบันทึกแบบ PID 58
5.2.2 การใช้งานการบันทึกแบบ P ร่วมกับการตัดสินใจ 59
5.2.3 การใช้งานการบันทึกแบบการสะท้อนของแสงร่วมกับการตัดสินใจ 61
5.3 การตัดขึ้นงานด้วยข้อมูลที่มีอยู่ในหน่วยความจำ 62
5.4 โปรแกรมปรับปรุงข้อมูล 62
5.4.1 การใช้งานโปรแกรมแก้ไขข้อมูล 63

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

บทที่ 6 การทดสอบการทำงาน 79
6.1 การทดสอบความสามารถในการเคลื่อนที่ของระบบขับเคลื่อน 79
6.2 การทดสอบความสามารถในการติดตามลายเส้นสำหรับการควบคุมแบบต่างๆ 89
6.3 การทดสอบความสามารถในการบันทึกข้อมูลที่ได้จากการติดตามลายเส้นด้วยวิธีการต่างๆ... 96
6.4 การทดสอบโปรแกรมแก้ไขข้อมูล 102
6.5 สรุปผลการทดลอง 104
บทที่ 7 **บทสรุป** 105
6.1 สรุปผลการวิจัย 105
เอกสารอ้างอิง 106
ภาคผนวก ก ชิ้นงานของวิทยานิพนธ์ 107
ภาคผนวก ข รายละเอียดของโปรแกรมต่าง ๆโดยสมบูรณ์ 112
ภาคผนวก ค บทความวิธีการเพิ่มความละเอียดเชิงมุมของแกนสเต็ปมอเตอร์ 174
ภาคผนวก ง บทความที่ได้รับการตีพิมพ์ในระหว่างการศึกษาปริญญาโท 187
ประวัติผู้เขียน 196

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป

รูปที่		หน้า
2.1	แสดงหัวอ่านต้นแบบซนิดเมกตริกซ์	4
2.2	แสดงแผนผังการสร้างสมการเส้นตรงจากการวัดแสงแบบเมกตริกซ์	5
2.3	โครงสร้างของระบบควบคุมโดยสังเขป	7
2.4	แสดงโครงสร้างระบบควบคุมที่ดูง่ายขึ้น	8
2.5	แสดงผลต่อระบบควบคุมที่เกิดจากเส้นตรง	8
2.6	แสดงผลต่อระบบควบคุมที่เกิดจากเส้นโค้ง	9
2.7	แสดงผลต่อระบบควบคุมที่เกิดจากเส้นหักมุม	9
2.8	แสดงค่าความผิดพลาดป้อนกลับที่ได้จากหัวอ่าน	10
2.9	แสดงรูปแบบการควบคุมแบบกึ่งตัดสินใจ	11
2.10	แสดงการปรับทิศทางการเดินของหัวอ่านต่อลายเส้น	12
2.11	แสดงรูปแบบการควบคุมแบบการสะท้อนของแสง	12
3.1	รูปถ่ายแสดงโครงสร้างของระบบขับเคลื่อนหัวอ่าน	14
3.2	แสดงการวางตำแหน่งของส่วนต่างๆ ในหัวอ่าน	15
3.3	แสดงวงจรสมบูรณ์ของส่วนแปลงความเข้มแสงเป็นข้อมูลดิจิตอลขนาด 8 บิต	16
3.4	แสดงโครงสร้างพื้นฐานของสเต็ปปิ้งมอเตอร์	17
3.5	ก. แสดงโครงสร้างประยุกต์ของสเต็ปปิ้งมอเตอร์	17
3.5	ข. ภาพผ่าตัดสเต็ปปิ้งมอเตอร์ที่ใช้งานทั่วไป	18
3.6	กราฟแสดงความสัมพันธ์ระหว่างมุมกับการจ่ายกระแสแบบปกติ	19
3.7	กราฟแสดงความสัมพันธ์ระหว่างมุมกับการจ่ายกระแสแบบสัมพันธ์	20
3.8	แสดงวงจรขับกระแสสเต็ปปิ้งมอเตอร์	21
3.9	แสดงวงจรแปลงข้อมูล 16 บิทเป็นความถี่	22
3.10	แสดงวงจรนับความถี่แปลงเป็นตำแหน่งอ้างอิงของมอเตอร์	23
3.11	แสดงวงจรตรวจจับจุดเริ่มต้นและจุดสิ้นสุดสำหรับ 1 แกน	24
3.12	แสดงวงจรเชื่อมต่อระหว่างคอมพิวเตอร์กับส่วนควบคุมต่างๆ	25
3.13	แสดงวงจรควบคุมระบบที่สมบูรณ์	26
4.1	แสดงโครงสร้างโปรแกรมของระบบควบคุมแบบ PID	31
4.2	แสดงโครงสร้างของโปรแกรมควบคุมแบบกึ่งตัดสินใจร่วมกับการควบคุมแบบ P	33
4.3	แสดงโครงสร้างโปรแกรมการติดตามลายเส้นแบบสะท้อนแสงร่วมกับการตัดสินใจ	36
4.4	แสดงโครงสร้างของโปรแกรมติดตามลายเส้นเพื่อบันทึกระบบ P I D	39
4.5	แสดงโครงสร้างโปรแกรมบันทึกสำหรับระบบควบคุม P I D	40

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ **หน้า**

4.6 แสดงโครงสร้างของโปรแกรมติดตามลายเส้นเพื่อบันทึกแบบกึ่งตัดสินใจร่วมกับการควบคุม..... แบบ P..... 43
4.7 แสดงโครงสร้างของโปรแกรมบันทึกข้อมูล..... 44
4.8 แสดงโครงสร้างโปรแกรมบันทึกลายเส้นแบบสะท้อนแสงร่วมกับการตัดสินใจทางโปรแกรม..... 46
4.9 แสดงโครงสร้างโปรแกรมบันทึกข้อมูล..... 47
4.10 แสดงโครงสร้างโปรแกรมอ่านข้อมูล..... 49
4.11 แสดงโครงสร้างโปรแกรมเขียนข้อมูล..... 50
4.12 แสดงโครงสร้างโปรแกรมหลักส่วนที่ 1..... 52
4.13 แสดงโครงสร้างโปรแกรมหลักส่วนที่ 2..... 53
4.14 แสดงโครงสร้างโปรแกรมหลักส่วนที่ 3..... 54
4.15 แสดงโครงสร้างโปรแกรมหลักส่วนที่ 4..... 55
5.1 แสดงรายการของโปรแกรมหลัก..... 56
5.2 แสดงตัวอย่างข้อมูลที่ปรากฏบนจอภาพ..... 57
5.3 แสดงรายการระบบบันทึก..... 58
5.4 แสดงรายการบันทึกแบบ P I D..... 58
5.5 แสดงรายการบันทึกแบบ P ร่วมกับการตัดสินใจ..... 59
5.6 แสดงรายการบันทึกแบบสะท้อนของแสงร่วมกับการตัดสินใจ..... 61
5.7 แสดงรายการการตัดโดยข้อมูลจากหน่วยความจำ..... 62
5.8 แสดงรายการของโปรแกรมแก้ไข..... 63
5.9 การแสดงรูปข้อมูลบนจอ..... 64
5.10 การแสดงรายการแก้ไข..... 64
5.11 แสดงขนาดชิ้นงานเทียบกับขนาดโต๊ะ..... 65
5.12 การแสดงข้อมูลเป็นจุดๆบนจอภาพ..... 66
5.13 การแสดงข้อมูลโดยขยายเพิ่มขึ้นสองเท่า..... 66
5.14 การแสดงข้อมูลโดยย่อลงสองเท่า..... 67
5.15 แสดงการใช้งานคำสั่ง PAN..... 70
5.16 แสดงการใช้งานคำสั่ง MOVE..... 71
5.17 แสดงการใช้งานคำสั่ง ROTATE..... 73
5.18 แสดงการใช้คำสั่ง 2 LINE ADJ..... 74
5.19 แสดงการใช้งานคำสั่ง 3 LINE ADJ..... 75

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ **หน้า**

5.20 แสดงการใช้งานคำสั่ง 3 CIR. ADJ. 76
5.21 แสดงการใช้คำสั่ง SIZE ADJ. 78
6.1 แสดงการเคลื่อนที่เป็นรูปวงกลมที่ความเร็ว 1 cm/sec และอัตราเร่ง 3 cm/sec^2 80
6.2 แสดงการเคลื่อนที่เป็นรูปสี่เหลี่ยมขนมเปียกปูน 80
6.3 แสดงการเคลื่อนที่เป็นรูปวงกลม 10 รอบ ที่ความเร็ว 300 cm/min 81
6.4 แสดงการเคลื่อนที่เป็นรูปสี่เหลี่ยม 10 รอบ ที่ความเร็ว 300 cm/min 81
6.5 แสดงการเคลื่อนที่เป็นรูปขนมเปียกปูน 10 รอบ ที่ความเร็ว 300 cm/min 82
6.6 แสดงการเคลื่อนที่เป็นวงกลมที่ความเร็ว 150 mm/sec อัตราเร่ง 100 mm/sec^2 83
6.7 แสดงการเคลื่อนที่เป็นวงกลมที่ความเร็ว 200 mm/sec อัตราเร่ง 100 m.m./sec^2 83
6.8 แสดงการเคลื่อนที่เป็นวงกลมที่ความเร็ว 220 mm/sec อัตราเร่ง 100 mm/sec^2 84
6.9 แสดงการเคลื่อนที่เป็นวงกลมที่ความเร็ว 250 mm/sec อัตราเร่ง 100 mm/sec^2 84
6.10 แสดงการเคลื่อนที่เป็นรูปสี่เหลี่ยมที่ความเร็ว 200 mm/sec อัตราเร่ง 500 mm/sec^2 85
6.11 แสดงการเคลื่อนที่เป็นรูปสี่เหลี่ยมที่ความเร็ว 200 mm/sec อัตราเร่ง 800 mm/sec^2 85
6.12 แสดงการเคลื่อนที่เป็นรูปสี่เหลี่ยมที่ความเร็ว 200 mm/sec อัตราเร่ง 1200 mm/sec^2 86
6.13 แสดงการเคลื่อนที่เป็นรูปสี่เหลี่ยมที่ความเร็ว 200 mm/secอัตราเร่ง 1500 mm/sec^2 86
6.14 แสดงการเคลื่อนที่เป็นรูปวงกลมกับดาวซ้อนกันแบบที่หนึ่ง ความเร็ว 100 mm/sec และ
อัตราเร่ง 100 mm/sec^2 87
6.15 แสดงการเคลื่อนที่เป็นรูปวงกลมกับดาวซ้อนกันแบบที่สอง ความเร็ว 100 mm./sec และ
อัตราเร่ง 100 mm/sec^2 87
6.16 แสดงการเคลื่อนที่เป็นรูปวงกันหอยที่ความเร็ว 100 mm/sec และอัตราเร่ง 100 mm/sec^2 88
6.17 แสดงรูปที่ใช้ทดสอบทางด้านเส้นตรงและเส้นโค้ง 89
6.18 ผลการติดตามลายเส้นระบบ PID ที่ความเร็ว 100 mm/min 90
6.19 ผลการติดตามลายเส้นระบบ PID ที่ความเร็ว 300 mm/min 90
6.20 ผลการติดตามลายเส้นระบบ PID ที่ความเร็ว 500 mm/min 91
6.21 ผลการติดตามลายเส้นระบบ PID ที่ความเร็ว 700 mm/min 91
6.22 ผลการติดตามลายเส้นระบบ P ร่วมกับการตัดสินใจที่ความเร็ว 100 mm/min 92
6.23 ผลการติดตามลายเส้นระบบ P ร่วมกับการตัดสินใจที่ความเร็ว 300 mm/min 92

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ **หน้า**

6.24 ผลการติดตามลายเส้นระบบ P ร่วมกับการตัดสินใจที่ความเร็ว 500 mm/min 93
6.25 ผลการติดตามลายเส้นระบบ P ร่วมกับการตัดสินใจที่ความเร็ว 700 mm/min 93
6.26 ผลการติดตามลายเส้นระบบสะท้อนแสงร่วมกับการตัดสินใจที่ความเร็ว 100 mm/min 94
6.27 ผลการติดตามลายเส้นระบบสะท้อนแสงร่วมกับการตัดสินใจที่ความเร็ว 300 mm/min 94
6.28 ผลการติดตามลายเส้นระบบสะท้อนแสงร่วมกับการตัดสินใจที่ความเร็ว 500 mm/min 95
6.29 ผลการติดตามลายเส้นระบบสะท้อนแสงร่วมกับการตัดสินใจที่ความเร็ว 700 mm/min 95
6.30 แสดงรูปมงกุฎที่ใช้ทดสอบ 96

กลุ่มรูปการทดลองระบบ PID

6.31 ข้อมูลบนจอภาพที่ได้จากการบันทึกตัวอย่างแรก 96
6.32 ผลการวาดโดยใช้ข้อมูลจากการบันทึกตัวอย่างแรก ที่ความเร็ว 1000 mm/min 97
6.33 ผลของการตามลายเส้นจากตัวอย่างที่ 2 97
6.34 ข้อมูลบนจอภาพที่ได้จากการบันทึกตัวอย่างที่ 2 97

กลุ่มรูปการทดลองระบบ P ร่วมกับการตัดสินใจ

6.35 ข้อมูลบนจอภาพที่ได้จากการบันทึกตัวอย่างแรก 98
6.36 ผลการวาดโดยใช้ข้อมูลจากการบันทึกตัวอย่างแรกที่ความเร็ว 1000 mm/min 98
6.37 ผลของการตามลายเส้นจากตัวอย่างที่ 2 99
6.38 ข้อมูลบนจอภาพที่ได้จากการบันทึกตัวอย่างที่ 2 99

กลุ่มรูปการทดลองระบบการสะท้อนแสงร่วมกับการตัดสินใจ

6.39 ข้อมูลบนจอภาพที่ได้จากการบันทึกตัวอย่างแรก 100
6.40 ผลการวาดโดยใช้ข้อมูลจากการบันทึกตัวอย่างแรกที่ความเร็ว 1000 mm/min 100
6.41 ผลของการตามลายเส้นจากตัวอย่างที่ 2 101
6.42 ข้อมูลบนจอภาพที่ได้จากการบันทึกตัวอย่างที่ 2 101
6.40 ผลการวาดโดยใช้ข้อมูลจากการบันทึกตัวอย่างที่ 2 ที่ความเร็ว 1000 mm/min 101

กลุ่มรูปที่ทำการแก้ไขแล้ว

6.41 ผลบนจอที่ได้หลังจากการแก้ไขแล้ว 102
6.42 ผลการวาดโดยข้อมูลที่แก้ไขแล้ว ที่ความเร็ว 1000 mm/min 103

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

ตารางที่ **หน้า**

1.1 ข้อมูลมาตราฐานสำหรับการตัดแผ่นเหล็กด้วยแก๊ส 2

6.1 สรุปขนาดเส้นผ่าศูนย์กลางของส่วนโค้งที่สามารถติดตามลายเส้นได้ที่ความเร็วต่าง ๆ..... 104

6.2 แสดงผลการสั่นที่เกิดขึ้นที่ความเร็วต่าง ๆ..... 104

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คำอธิบายสัญญลักษณ์/คำย่อ

LT	ตำแหน่งลายเส้นที่อยู่ใกล้หัวอ่านมากที่สุด
LX	ตำแหน่งลายเส้นที่อยู่ใกล้หัวอ่านมากที่สุดในแนวแกนนอน
LY	ตำแหน่งลายเส้นที่อยู่ใกล้หัวอ่านมากที่สุดในแนวแกนตั้ง
de	ระยะห่างระหว่างลายเส้นกับหัวอ่านที่ใกล้ที่สุด
dx	ระยะห่างระหว่างลายเส้นกับหัวอ่านที่ใกล้ที่สุดในแนวแกนนอน
dy	ระยะห่างระหว่างลายเส้นกับหัวอ่านที่ใกล้ที่สุดในแนวแกนตั้ง
eh	ระยะห่างระหว่างลายเส้นกับหัวอ่านที่อ่านได้จากหัวอ่าน
K_M	ค่าคงที่ของระบบขับเคลื่อนสเต็ปมอเตอร์
dvx	ค่าความเร็วที่เปลี่ยนแปลงในแนวแกนนอน
dvy	ค่าความเร็วที่เปลี่ยนแปลงในแนวแกนตั้ง
VX_1, VX_2	ค่าความเร็วในแนวแกนนอน
VY_1, VY_2	ค่าความเร็วในแนวแกนตั้ง
θ_1	มุมกำหนดทิศทางหัวอ่านที่ได้ใหม่
θ_2	มุมที่หัวอ่านเคลื่อนที่ไป
V_o , V_m	ความเร็วคงที่ที่ผู้ใช้กำหนด
θ_{VL}	ทิศทางของลายเส้นที่หัวอ่านกำลังอ่านอยู่
θ_{VT}	ทิศทางเดินของหัวอ่านในขณะนั้น
θ_e	ทิศทางที่แตกต่างระหว่างหัวอ่านกับลายเส้นในขณะนั้น
θ_{eh}	ทิศทางที่แตกต่างที่อ่านได้โดยหัวอ่าน
$d\theta$	ทิศทางที่ต้องเปลี่ยนแปลงไปจากเดิม
θ_M , θ_m	ทิศทางหลักของหัวอ่าน
D.C.MOTOR	มอเตอร์กระแสตรง
สเต็ปต่อรอบ	หมายถึงตำแหน่งที่สเต็ปมอเตอร์สามารถหยุดได้ในหนึ่งรอบการหมุน
$\theta(n)$	ทิศทางที่เปลี่ยนแปลงไป
e(n)	ค่าความผิดพลาดที่ได้จากหัวอ่าน
e(n-1)	ค่าความผิดพลาดเก่าที่ได้จากหัวอ่าน
X[...]	ตัวแปรเก็บตำแหน่งทางแกนนอน
Y[...]	ตัวแปรเก็บตำแหน่งทางแกนตั้ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1
บทนำ

1.1 วัตถุประสงค์

ในปัจจุบัน การตัดชิ้นงานต้องการความแม่นยำและคุณภาพที่ดี รวมทั้งปริมาณการตัดต่อหน่วยเวลาก็ต้องมีค่าสูงด้วย ซึ่งเป็นการยากที่จะอาศัยความสามารถของมนุษย์โดยลำพัง อีกทั้งปัจจุบันราคาคอมพิวเตอร์มีราคาถูกลงอย่างมาก เช่นคอมพิวเตอร์ที่ใช้ CPU 80486DLC-40 ก็มีราคาไม่ถึงสองหมื่นบาท ดังนั้นจึงเห็นว่าเป็นการคุ้มค่าที่จะพัฒนาระบบตัดชิ้นงานอัตโนมัติที่ควบคุมด้วยไมโครคอมพิวเตอร์ โดยมีจุดประสงค์ให้สามารถนำไปใช้งานได้กับงานตัดทั่วไป รวมทั้งพัฒนาขั้นตอนการใช้งานให้ง่ายขึ้นด้วยเพื่อให้บุคคลที่ไม่มีความชำนาญสามารถใช้งานได้โดยเพียงแต่เขียนรูปแบบชิ้นงานที่ต้องการตัดลงในแผ่นกระดาษแล้วนำไปให้เครื่องตัดชิ้นงานอัตโนมัติทำการอ่านลายเส้นแล้วเก็บข้อมูลเข้าคอมพิวเตอร์และเมื่อต้องการตัดชิ้นงานก็เพียงแต่สั่งให้คอมพิวเตอร์เอาข้อมูลที่ได้เก็บไว้นำออกมาใช้ในการตัดชิ้นงานดังกล่าว

นอกจากนี้ การพัฒนาจะมุ่งเน้นให้ระบบโดยรวมมีอุปกรณ์และราคาต่ำที่สุด ในขณะที่มีประสิทธิภาพการทำงานที่ดี และง่ายต่อการซ่อมบำรุง มีความเชื่อถือได้ดี

1.2 ขอบเขตของงานวิจัย

ระบบตัดชิ้นงานอัตโนมัติที่สร้างขึ้นนี้จะสามารถทำการวิ่งติดตามลายเส้นที่วาดไว้ได้โดยอิสระ และสามารถตั้งค่าความเร็วต่างๆ ได้ อีกทั้งยังสามารถทำการบันทึกตำแหน่งต่างๆ ในระหว่างการติดตามลายเส้นได้อีกด้วย สามารถตั้งค่าความละเอียดสำหรับการบันทึกตำแหน่งลายเส้น และเลือกวิธีการติดตามลายเส้นแบบต่างๆ ได้จากสามวิธีที่มีให้เลือก ในวิธีการติดตามลายเส้นแต่ละแบบผู้ใช้สามารถทำหนดตัวแปรสำหรับการติดตามลายเส้นเพื่อให้เหมาะสมกับลายเส้นและความเร็วได้ นอกจากนี้ยังได้มีการเขียนโปรแกรมสำหรับปรับปรุงข้อมูลของลายเส้นที่ได้บันทึกไว้เพื่อให้สามารถทำการแก้ไขข้อมูลลายเส้นได้เป็นอย่างดี

สำหรับความเร็วในการติดตามลายเส้นนั้นจะพยายามทำให้ได้เร็วที่สุด แต่ในงานตัดทั่วไปมักจะเป็นการตัดแผ่นโลหะหนาเป็นส่วนใหญ่และความเร็วก็มักจะต่ำกว่า 70 เซนติเมตรต่อนาที ดูจากตารางที่ 1.1 แต่ความเร็วในการตัดที่ได้จากการบันทึกลายเส้นนั้นจะสามารถทำได้สูงกว่ามาก ดังนั้นจึงขอแนะนำว่าในการตัดชิ้นงานที่ดีนั้นควรเริ่มให้ทำการติดตามลายเส้นเพื่อบันทึกด้วยความเร็วที่เหมาะสมเพื่อให้ได้รูปที่ใกล้เคียงที่สุด จากนั้นก็ทำการปรับปรุงข้อมูลด้วยโปรแกรมแก้ไขถ้าจำเป็น ต่อไปก็ให้ทำการตัดชิ้นงานด้วยข้อมูลที่ได้บันทึกไว้ ซึ่งจะทำให้สามารถตัดชิ้นงานได้คุณภาพที่ดีที่ทุกความเร็วเท่าที่ระบบขับเคลื่อนสามารถตอบสนองได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Plate Thicknes in	Nozzle Size	Gas Pressure p.s.i		Cutting Speed	Gas Flow cf/h		
		Oxygen	Acetylene	in/min	Cutting Oxygen	Preheat Oxygen	Acetylene
~ 3/16	00	21	2.8	~27	24	16	14
3/16 ~ 3/8	0	28	2.8	27~22	42	16	14
3/8 ~ 5/8	1	36	2.8	22~19	74	16	14
5/8 ~1 3/16	2	43	2.8	19~16	120	16	14
1 3/16 ~1 9/16	3	43	2.8	16~14	152	19	17
1 9/16 ~2	4	50	3.6	14~13	229	21	19
2 ~ 4	5	57	4.3	13~8	388	26	24
4 ~ 6	6	57	5.0	8~6	530	29	26
6 ~ 8	7	64	5.7	6~4	777	39	35
10 ~ 12	8	64	5.7	4~3	988	39	35

ตารางที่ 1.1 ข้อมูลมาตราฐานสำหรับการตัดแผ่นเหล็กด้วยแก๊ส

1.3 ประโยชน์ของงานวิจัย

ระบบตัดชิ้นงานนี้ผู้วิจัยเห็นว่ามีประสิทธิ์ภาพที่ดีพอที่จะนำไปใช้ในงานตัดงานทั่วไปเป็นอย่างดี ซึ่งประโยชน์ที่จะได้รับมีดังต่อไปนี้

1. ต้นทุนในการตัดต่อหน่วยจะมีราคาถูกลงเนื่องมาจากประหยัดค่าแรงงานไปได้ส่วนหนึ่ง

2. ปริมาณงานที่ได้จะมากกว่าการใช้แรงงานมนุษย์ เพราะระบบตัดชิ้นงานสามารถทำงานได้อย่างต่อเนื่องเป็นเวลานานๆได้

3. คุณภาพของชิ้นงานสามารถควบคุมได้ดีกว่า เพราะการตัดชิ้นงานโดยมนุษย์มีความไม่แน่นอนในด้านคุณภาพเป็นอย่างมากโดยเฉพาะความสามารถในด้านการควบคุมความเร็วและตำแหน่ง แต่ระบบตัดชิ้นงานสามารถควบคุมความเร็วและตำแหน่งได้เป็นอย่างดี ส่งผลให้ชิ้นงานตัดทุกชิ้นงานมีคุณภาพที่ใกล้เคียงกันมาก

4. การสั่งงานระบบตัดชิ้นงานมีความง่ายสำหรับการใช้งานของบุคคลทั่วไป เพราะระบบตัดชิ้นงานจะทำการเก็บข้อมูลจากลายเส้นเข้าไปไว้ในคอมพิวเตอร์โดยอัตโนมัติ ทำให้ไม่ต้องทำการออกแบบชิ้นงานบนคอมพิวเตอร์เพียงแต่ต้องออกแบบบนกระดาษ ซึ่งเป็นวิธีที่ง่ายกว่าสำหรับผู้ที่ไม่ชำนาญใช้คอมพิวเตอร์

5. ระบบตัดชิ้นงานที่ได้นี้มีราคาถูกอย่างมากเมื่อพิจารณาถึงอุปกรณ์ที่ใช้ซึ่งมีไม่มาก มีผลให้ค่า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บำรุงรักษาต่ำ เหมาะกับร้านประกอบการที่ไม่ต้องการลงทุนมาก ในขณะที่มีประสิทธิ์ภาพในการตัดที่ดีมาก

6. ระบบตัดชิ้นงานนี้มีช่องทางที่สามารถทำการพัฒนาได้ดีกว่านี้อีก โดยการเพิ่มโปรแกรมการออกแบบชิ้นงานบนคอมพิวเตอร์ โปรแกรมคำนวณเส้นทางตัดที่ใช้เวลาน้อยที่สุด โปรแกรมแปลงข้อมูลลายเส้นให้อยู่ในรูปคำสั่งการเคลื่อนที่หรือโปรแกรมวิเคราะห์ข้อมูลลายเส้น ฯลฯ ซึ่งจะทำให้เราสามารถพึ่งตนเองได้อย่างเต็มที่โดยไม่ต้องไปพึ่งพาระบบตัดที่มีราคาแพงจากประเทศอื่น

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

บทที่ 2
การเลือกชนิดหัวอ่านและการออกแบบวิธีการติดตามลายเส้น

2.1 การเลือกชนิดหัวอ่านให้เหมาะสมกับระบบ

ก่อนอื่นจำต้องทราบคุณสมบัติของหัวอ่านหรืออุปกรณ์ตรวจจับลายเส้นก่อนที่จะเลือกใช้ก่อน ดังนั้นเราจึงได้ทำการออกแบบและสร้างหัวอ่านต้นแบบชนิดต่างๆ ขึ้นมาทำการทดลอง เพื่อคัดเลือกชนิดที่เหมาะสมที่สุด ซึ่งสามารถสรุปได้ดังนี้

2.1.1 หัวอ่านแบบเมกตริกซ์ขนาด 4 * 4

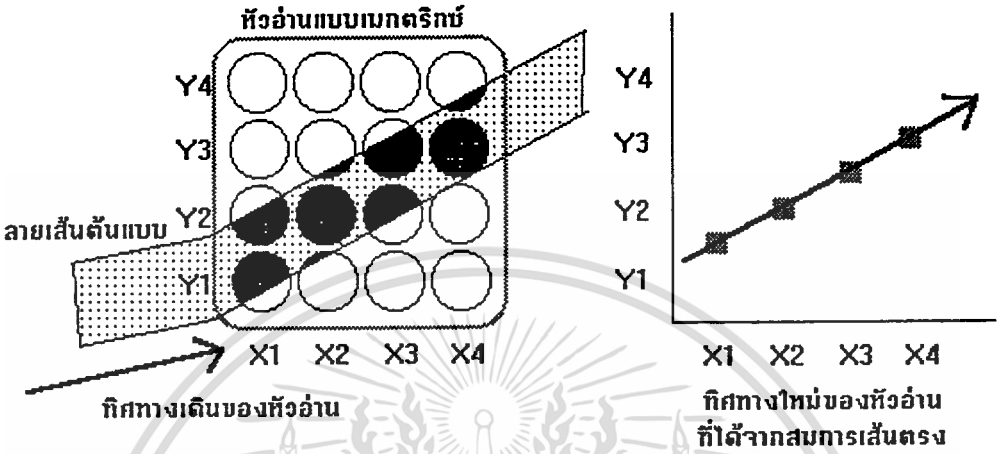
2.1.2 หัวอ่านแบบเซลเดียว

2.1.1 หัวอ่านแบบเมกตริกซ์ขนาด 4 * 4 ประกอบด้วยเซลวัดแสง 16 ตัววางตัวในรูปเมกตริกซ์ มีขนาดหน้าตัด 4 * 4 $cm.^2$ จึงต้องใช้สายไฟเบอร์ออปติกช่วยทำให้พื้นที่หน้าตัดรับแสงเล็กลงเหลือเท่ากับ 4 * 4 $mm.^2$ ดังรูปที่ 2.1 จะเห็นว่าสายไฟเบอร์ถูกมัดเรียงให้รูปหน้าตัดสี่เหลี่ยมที่ปลายโดยใช้เส้นอลูมิเนียมฉากประกบรักษารูปไว้ ส่วนอีกปลายด้านหนึ่งต่อเข้ากับเซงรับแสงแต่ละเซลซึ่งก็ถูกมัดรวมเช่นกัน สัญญาณแสงในรูปไฟฟ้าที่ได้จากเซลรับแสงแต่ละตัวจะถูกแปลงสัญญาณให้อยู่ในรูปข้อมูลดิจิตอลโดยวงจรแปลงสัญญาณที่มีอยู่เพียงวงจรเดียวโดยเลือกสัญญานที่จะแปลงผ่านอุปกรณ์อนาล็อกสวิตช์(I.C.4066)

รูปที่ 2.1 แสดงหัวอ่านต้นแบบชนิดเมกตริกซ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จุดประสงค์ของการออกแบบหัวอ่านชนิดนี้ คือต้องการนำค่าที่ได้จากการวัดแสงของเซลแต่ละเซลมาทำการสร้างสมการเส้นตรงขึ้น พิจารณารูปที่ 2.2



รูปที่ 2.2 แสดงแผนผังการสร้างสมการเส้นตรงจากการวัดแสงแบบเมกตริกซ์

จากสมการที่ได้จะทำให้คอมพิวเตอร์ทราบค่าความชันของลายเส้นในขณะนั้นว่ามีค่าเท่าไร และนำไปใช้กำหนดทิศทางการเคลื่อนที่ของหัวอ่านเพื่อติดตามลายเส้นอย่างต่อเนื่องโดยตลอด

แต่เมื่อได้ทำการทดลองขั้นพื้นฐานโดยการทดสอบความสามารถในวัดแสง พบว่าเกิดปัญหาขึ้นหลายประการ ตัวอย่างได้แก่

ก. ความไวในการรับแสงของเซลแต่ละเซลไม่เท่ากัน ทั้งนี้คงเนื่องจากการใช้ชนิดของเซลรับแสงไม่เหมาะสมเนื่องจากเซลรับแสงที่ใช้ ถูกออกแบบมาสำหรับรับส่งข้อมูลดิจิตอลผ่านสายไฟเบอร์ออปติกรวมถึงการสูญเสียความสว่างภายในสายด้วยเพราะความเค้นที่เกิดจากการดัดสายไฟเบอร์ฯของเซลรับแสงไม่เท่ากัน

ข. ความล่าช้าจากการรอคอยทุกครั้งที่มีการสับสวิตช์เลือกเซลรับแสงตัวถัดไป ทั้งนี้เนื่องมาจากอุปกรณ์ออปเอมป์ต้องการเวลาปรับตัวรับสัญญาณที่เข้ามาใหม่ ทำให้ความเร็วของระบบโดยรวมลดลงอย่างมาก

จากผลดังกล่าวทำให้ต้องมีการปรับปรุงโดยแยกวงจรแปลงสัญญาณให้แต่ละเซล แล้วทำการปรับความไวแสงให้เท่ากันซึ่งจะทำให้ต้นทุนสูงและปรับแต่งลำบากโดยเฉพาะกับบุคคลที่ไม่คุ้นเคยกับระบบดังนั้นจึงสรุปได้ว่า หัวอ่านดังกล่าวไม่เหมาะต่อการพัฒนานี้

2.1.2 หัวอ่านแบบเซลเดียว มีการพัฒนาขึ้นหลังจากแบบแรก เมื่อทำการทดลองใช้งานดู โดยมี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แนวคิดที่จะอาศัยความสามารถของโปรแกรมชดเชยข้อด้อยของการรับแสงแบบเซลเดียว จากการทดลองให้ผลที่น่าพอใจในด้านความเร็วในการแปลงสัญญาณแสงเป็นข้อมูลเลขฐานสองให้แก่คอมพิวเตอร์ วงจรมีความเรียบง่ายไม่ซับซ้อนสามารถสร้างและปรับแต่งได้ง่าย มีความลำบากเพียงการเขียนโปรแกรมเท่านั้นอันเนื่องมาจากช่วงการวัดค่าความผิดพลาดจากลายเส้นอยู่ในช่วงแคบมาก มีช่วงกว้างประมาณ 1.00 m.m. เท่านั้นดังนั้นเมื่อลายเส้นอยู่ห่างจากหัวอ่านเกินระยะดังกล่าวจะก็จะได้ค่าคงที่ค่าหนึ่ง ทำให้การติดตามลายเส้นโดยใช้วิธีการควบคุมแบบป้อนกลับ (FEEDBACK CONTROL) อาจเกิดปัญหาทางด้านค่าความผิดพลาดป้อนกลับที่ไม่ถูกต้อง ทำให้ระบบขาดเสถียรภาพได้ง่าย ทำให้ต้องแก้ไขโดยอาศัยการตัดสินใจทางโปรแกรมเข้าช่วย ซึ่งไม่มีเกี่ยวข้องกับผู้ใช้เลยจึงถือได้ว่าตรงตามจุดประสงค์ที่สุด ดังนั้นจึงได้เลือกหัวอ่านชนิดนี้ทำการพัฒนาต่อไป

2.2 การออกแบบวิธีการติดตามลายเส้นเพื่อการตัดและการบันทึกที่ดี

เมื่อได้เลือกหัวอ่านแบบเซลเดียวแล้วจึงได้คิดวิธีการติดตามลายเส้นต่าง ๆ ที่เหมาะสมได้ดังต่อไปนี้

2.2.1 ใช้ระบบควบคุมแบบตั้งค่าควบคุม PID ได้

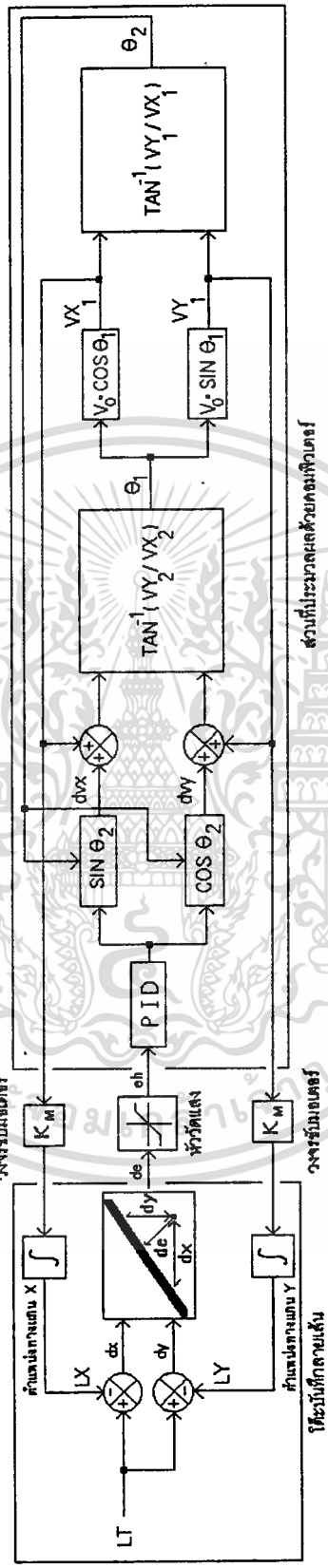
2.2.2 ใช้ระบบโปรแกรมแบบกึ่งตัดสินใจร่วมกับการควบคุมแบบ P

2.2.3 ใช้วิธีการวิ่งตามทางลายเส้นแบบสะท้อนแสงร่วมกับการตัดสินใจทางโปรแกรม

2.2.1 ใช้ระบบควบคุมแบบตั้งค่าควบคุม PID ได้ เพื่อให้ผู้ใช้ที่มีความสามารถทางด้านระบบควบคุมสามารถตั้งค่าควบคุมให้เหมาะสมกับงานต่างๆ ได้ รูปแบบการควบคุมแบบ PID สำหรับระบบนี้กำหนดได้ดังรูปที่ 2.3

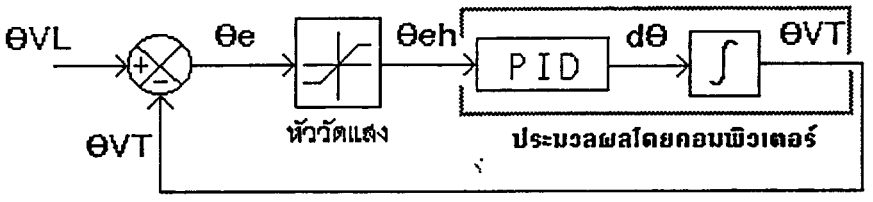
จากโครงสร้างของระบบที่สร้างขึ้นจะเห็นว่าระบบมีการเปลี่ยนทิศทางเดินตามลายเส้นที่เปลี่ยนไป ดังนั้น เราจะสรุปโครงสร้างให้ดูเรียบง่ายยิ่งขึ้นได้ดังรูปที่ 2.4

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.3 โครงสร้างของระบบควบคุมโดยสังเขป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



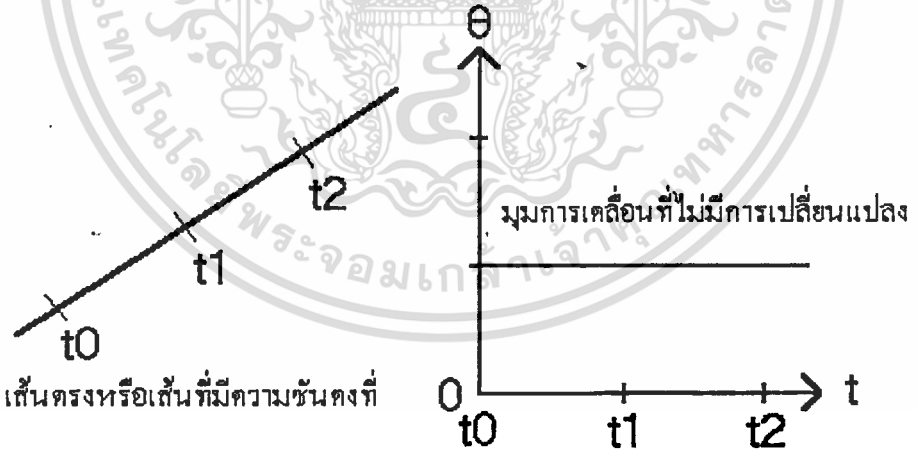
รูปที่ 2.4 แสดงโครงสร้างระบบควบคุมที่ดูง่ายขึ้น

โดยเราจะถือว่าทิศทางของลายเส้นที่เปลี่ยนแปลงนั้นเป็นการอินพุทของระบบ ทำให้ระบบต้องปรับทิศทางเดินของหัววัดให้สมดุลย์กับลายเส้นตลอดเวลา

ก่อนอื่นต้องพิจารณาว่าลายเส้นต่างๆ มีคุณสมบัติเกี่ยวข้องกับการควบคุมอย่างไรบ้าง ซึ่งแยกออกได้ 3 ลักษณะคือ

1. เส้นตรง
2. เส้นโค้ง
3. เส้นหักมุม

1. เส้นตรง หมายถึงเส้นตรงทั่วไป ซึ่งจะมีผลต่อระบบควบคุมดังรูปที่ 2.5

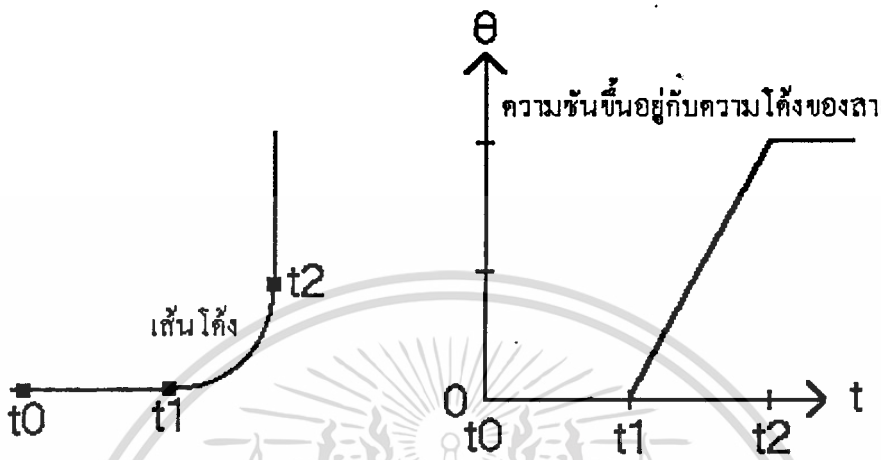


รูปที่ 2.5 แสดงผลต่อระบบควบคุมที่เกิดจากเส้นตรง

จากรูปจะเห็นว่าไม่มีผลกระทบต่อระบบควบคุมแต่อย่างไร แสดงว่าระบบจะมีเสถียรภาพต่อเส้นตรงเสมอ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

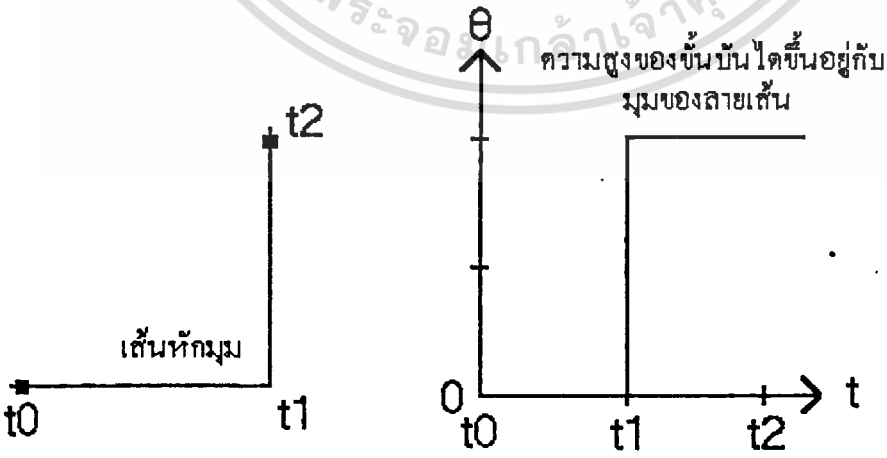
2. เส้นโค้ง หมายถึงเส้นที่มีความโค้งต่างๆ ไม่ว่าจะเป็นโค้งวงกลมหรือวงรี ซึ่งจะมีผลต่อระบบควบคุมดังรูปที่ 2.6



รูปที่ 2.6 แสดงผลต่อระบบควบคุมที่เกิดจากเส้นโค้ง

จากรูปจะเห็นว่ามีผลกระทบต่อระบบควบคุมในลักษณะที่มีความชัน(ramp)เกิดขึ้น ซึ่งเรียกได้ว่ามีผลแบบ RAMP INPUT

3.เส้นหักมุมหมายถึงเส้นตรงสองเส้นที่ไม่ขนานกันลากมาบรรจบกันแล้วเกิดเส้นหักมุมต่างๆ เช่น เส้นมุมป้าน มุมฉาก และ มุมแหลม เป็นต้น ซึ่งจะมีผลต่อระบบควบคุมดังรูปที่ 2.7



รูปที่ 2.7 แสดงผลต่อระบบควบคุมที่เกิดจากเส้นหักมุม

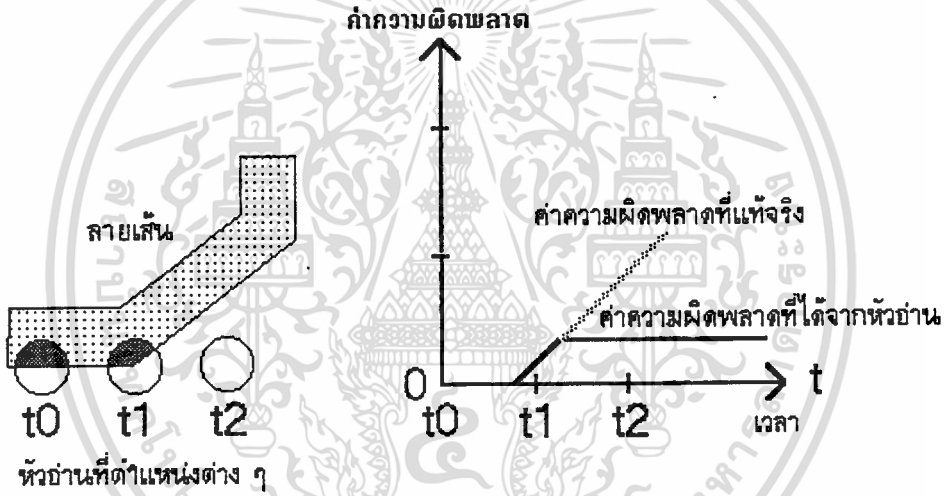
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปจะเห็นว่ามีผลกระทบต่อระบบควบคุมในลักษณะที่เป็นขั้นบันได(STEP) ซึ่งเรียกได้ว่ามีผลแบบ STEP INPUT

ต่อไปพิจารณาถึงการตอบสนองของหัวอ่านแบบเซลเดียวที่มีต่อระบบควบคุมในผลกระทบที่เกิดขึ้นในแต่ละแบบ

1. เส้นตรง ในทางอุดมคติ เมื่อหัวอ่านอยู่ในตำแหน่งที่เหมาะสมและมีทิศทางการเคลื่อนที่ในทางเดียวกับเส้นตรงจะไม่เกิดผลใดๆ ต่อระบบควบคุม เพราะหัวอ่านจะไม่มีโอกาสหลุดออกนอกลายเส้นทำให้ไม่เกิดค่าผิดพลาดป้อนกลับให้แก่ระบบควบคุม

2.เส้นโค้งและเส้นหักมุม ในขณะที่หัวอ่านกำลังวิ่งไปตามลายเส้นตรงและพบกับเส้นโค้งหรือเส้นหักมุม จะทำให้เกิดค่าความผิดพลาดจากหัวอ่านป้อนให้แก่ระบบควบคุม ดังรูปที่ 2.8



รูปที่ 2.8 แสดงค่าความผิดพลาดป้อนกลับที่ได้จากหัวอ่าน

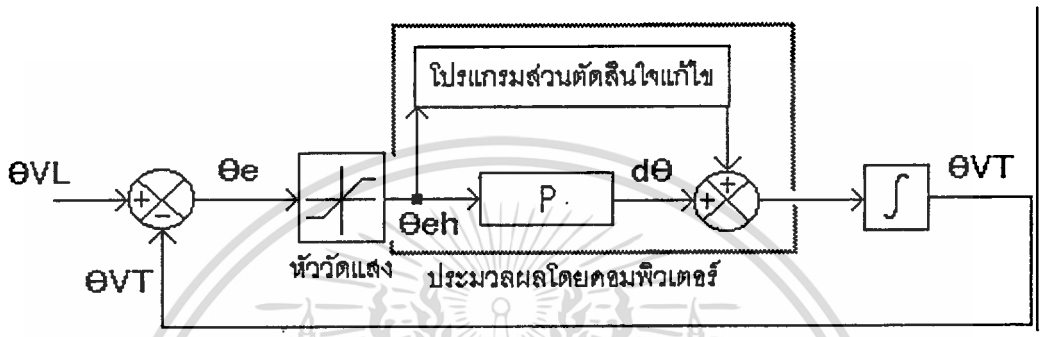
จากรูปที่แสดงจะเห็นได้ว่าค่าความผิดพลาดป้อนกลับที่ได้จะอยู่ในช่วงแคบๆ อันเนื่องมาจากพื้นที่วัดแสงของหัวอ่านมีขนาดเล็กมาก ค่าความผิดพลาดที่ได้จะมีค่าสูงสุดและต่ำสุดอยู่ในช่วง+/- 128 หน่วยเท่านั้นเอง ทำให้ระบบขาดเสถียรภาพความเป็นเชิงเส้น เป็นปัญหาในการให้ค่าตัวแปรควบคุม(PID)แก่ระบบดังนั้นจึงควรที่จะอาศัยโปรแกรมสร้างค่าความผิดพลาดชดเชยในกรณีที่หัวอ่านไม่สามารถให้ค่าความผิดพลาดที่แท้จริงได้(ค่าความผิดพลาดชดเชยไม่ได้มีการพัฒนาในงานวิจัยนี้)

2.ใช้การเขียนโปรแกรมแบบกึ่งตัดสินใจร่วมกับการควบคุมแบบ P เป็นการทำงานร่วมกันระหว่างระบบการควบคุมแบบPกับการควบคุมแบบเงื่อนไขโดยโปรแกรมคอมพิวเตอร์ อธิบายการทำงานได้ว่าในกรณีที่ลายเส้นมีความโค้งหรือมุมเล็กน้อยการควบคุมแบบP จะสามารถทำงานได้ดี แต่เมื่อลายเส้นมี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ความโค้งหรือมุมมากขึ้นจะทำให้การควบคุมแบบ P ทำงานไม่ทัน ทำให้หัวอ่านหลุดออกนอกลายเส้นไป ดังนั้นต้องอาศัยการตัดสินใจด้วยโปรแกรมที่ได้ตั้งเงื่อนไขไว้โดยเงื่อนไขที่ใช้ได้จากการทดลองโดยอาศัยการคาดคะเนและการลองผิดลองถูกจนกว่าจะได้เงื่อนไขสำหรับการตัดสินใจที่ดี ซึ่งจากผลการทดลองที่ได้ พบว่าลายเส้นที่ได้จะมีความราบเรียบและต่อเนื่องดีสำหรับลายเส้นที่มีการเปลี่ยนทิศทางเป็นมุมมน

รูปแบบการควบคุมของระบบดังกล่าวแสดงได้ดังรูปที่ 2.9 .

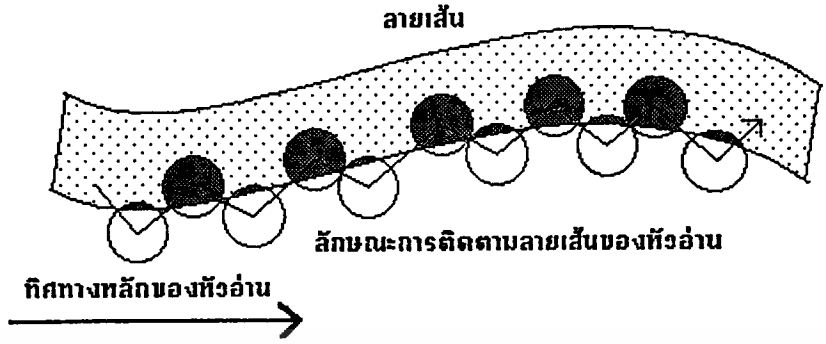


รูปที่ 2.9 แสดงรูปแบบการควบคุมแบบกึ่งตัดสินใจ

จากรูปที่ 2.9 ส่วนตัดสินใจแก้ไขมีหน้าที่ตรวจสอบค่าความผิดพลาดที่ได้จากหัวอ่านเพื่อประมาณสถานการณ์การติดตามลายเส้นของหัวอ่านว่ายังสามารถติดตามลายเส้นได้ดีอยู่หรือไม่ โดยจะคอยดูว่าค่าความผิดพลาดที่ได้ว่ามีการเปลี่ยนแปลงอยู่ในช่วงที่เหมาะสมหรือไม่ ถ้าหากว่าค่าความผิดพลาดเกินค่าที่กำหนดไว้ในช่วงเวลาหนึ่ง ก็แสดงว่าหัวอ่านเริ่มมีปัญหาในการติดตามลายเส้นที่มีการเปลี่ยนแปลงทิศทางที่มากเกินความสามารถในปรับตัวของหัวอ่าน ดังนั้นส่วนตัดสินใจแก้ไขก็จะให้ค่าป้อนกลับเสริม เพื่อช่วยให้หัวอ่านสามารถติดตามลายเส้นได้ตามปกติต่อไป

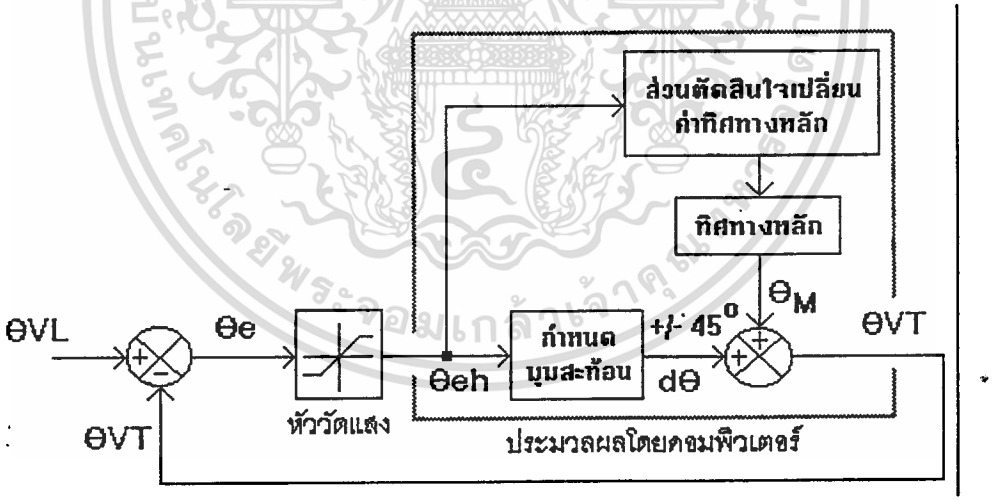
3. **ใช้วิธีการวิ่งตามทางลายเส้นแบบสะท้อนแสงร่วมกับการตัดสินใจทางโปรแกรม** เป็นการเลียนแบบการสะท้อนของแสงในสายตัวนำแสง ซึ่งจะมีทิศทางการเดินแบบสะท้อนกลับไปมาระหว่างช่วงค่าผิดพลาดช่วงหนึ่งซึ่งสามารถกำหนดได้โดยตั้งค่าสูงสุดและต่ำสุดไว้ วิธีนี้ยังมีจุดอ่อนอยู่ที่ว่าจะทำให้เกิดการสั่นในระหว่างการติดตามลายเส้นทำให้ไม่เหมาะแก่การตัดชิ้นงานไปพร้อม ๆ กันกับการติดตามลายเส้น แต่วิธีนี้สามารถติดตามลายเส้นที่ลักษณะโค้งแคบ ๆ หรือมุมหักศอกได้ดีมากที่ความเร็วต่ำ ทำให้เหมาะสมกับการทำงานในลักษณะติดตามลายเส้นเพื่อเก็บข้อมูล และค่าความผิดพลาดที่ได้จากการติดตามลายเส้นก็จะมีค่าอยู่ในขอบเขตที่กำหนดทำให้ผลการบันทึกมีค่าผิดพลาดไม่มากนัก ใกล้เคียงกับต้นฉบับมาก แต่อย่างไรก็ตาม วิธีนี้ก็ต้องอาศัยการตัดสินใจทางโปรแกรมเข้าช่วย เพราะวิธีการสะท้อนทิศทางการเดินของหัวอ่านจะสะท้อนกลับไปมาในช่วงองศาหลักบวกกับ ± 45 องศาสะท้อนเท่านั้น ดูรูปที่ 2.10

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.10 แสดงการปรับทิศทางการเดินของหัวอ่านต่อลายเส้น

ดังนั้นการติดตามลายเส้นวิธีนี้จะมีการตอบสนองต่อลายเส้นที่มีทิศทางที่เปลี่ยนแปลงไม่เกิน ± 45 องศา ดังนั้นเมื่อมีลายเส้นมีการเปลี่ยนทิศทางที่มากกว่า ± 45 องศา จึงจำเป็นที่จะต้องมีการเปลี่ยนทิศทางเดินหลักของหัวอ่านโดยอาศัยการตัดสินใจของโปรแกรมเพื่อให้หัวอ่านสามารถติดตามลายเส้นต่อไปได้ ดังนั้นรูปแบบที่ใช้ในการควบคุมวิธีนี้สามารถกำหนดได้ดังรูปที่ 2.11



รูปที่ 2.11 แสดงรูปแบบการควบคุมแบบการสะท้อนของแสง

จากรูปที่ 2.11 ส่วนตัดสินใจเปลี่ยนค่าทิศทางหลักทำหน้าที่สังเกตค่าความผิดพลาดว่าหัวอ่านยังสามารถติดตามลายเส้นได้หรือไม่ ถ้าหากว่าค่าความผิดพลาดมีค่าเกินช่วงที่กำหนดเป็นระยะเวลาหนึ่งก็จะมีการตัดสินใจเปลี่ยนทิศทางหลัก โดยจะมีการเปลี่ยนแปลงมุมครั้งละ ± 20 องศา ทำให้หัวอ่านสามารถติดตามลายเส้นได้อย่างต่อเนื่องตลอดเวลา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2 ความเป็นมาของการพัฒนาทั้ง 3 วิธี

จากการติดตามลายเส้นทั้งสามวิธีดังกล่าวที่จริงมีการพัฒนาแบบต่อเนื่องกันมาโดยเริ่มที่การควบคุมแบบ PID จะมีปัญหาเรื่องลายเส้นที่ได้จากการติดตามเกิดการสั่นที่ความเร็วสูงซึ่งคาดว่าจะเป็นลักษณะของการสั่นอันเนื่องมาจากความไม่สมดุลย์ของตัวควบคุม D เพราะค่าความผิดพลาดมีการเปลี่ยนแปลงที่มากเกินไป ซึ่งจำเป็นต้องปรับให้มีค่าน้อยลง สำหรับค่าตัวควบคุมแบบ I นั้นมีประโยชน์ในด้านการปรับทิศทางการที่ให้สามารถติดตามลายเส้นต่อไปได้ แต่ความสามารถของการควบคุมแบบ PID สามารถติดตามลายเส้นที่มีลักษณะโค้งและตรงได้ดีมากแต่ไม่สามารถติดตามลายเส้นมีมุมหรือโค้งที่มากๆ ได้

สำหรับระบบโปรแกรมแบบกึ่งตัดสินใจร่วมกับการควบคุมแบบ P เป็นการพัฒนาต่อจากระบบ PID เนื่องจากเห็นว่าช่วงกว้างของการวัดลายเส้นมีขนาดที่แคบมาก ดังนั้นค่าที่ได้จากการวัดแสงจึงมักจะอิ่มตัวเมื่อติดตามลายเส้นที่ความโค้งอยู่ตลอดเวลาทำให้ผลของการควบคุมแบบ I และ D ให้ผลที่ไม่ดีนัก (ซึ่งการควบคุมแบบ PID นั้นเหมาะสมกับการควบคุมที่มีช่วงของการป้อนกลับที่กว้างอย่างพอเพียง แต่ไม่เหมาะกับการป้อนกลับที่มีช่วงแคบๆและอิ่มตัวอยู่เสมอ) ดังนั้นจึงได้พิจารณาหาวิธีใหม่ซึ่งได้ข้อสรุปว่า

1. การควบคุมแบบ P สามารถติดตามลายเส้นที่ตรงหรือมีความโค้งเพียงเล็กน้อยได้ดี

2. ในการติดตามลายเส้นที่ความโค้งมากสามารถใช้การประมาณอัตราการเปลี่ยนทิศทางได้โดยอาศัยเงื่อนไขที่เหมาะสม

ดังนั้นจึงได้ออกแบบโปรแกรมแบบกึ่งตัดสินใจร่วมกับการควบคุมแบบ P ขึ้นมาและเมื่อได้ทำการทดสอบพบว่าการตอบสนองต่อลายเส้นสามารถทำได้ราบเรียบกว่าการควบคุมแบบ PID แต่อย่างไรก็ตามในการติดตามลายเส้นที่มีมุมหรือเส้นโค้งมากๆ มักก่อให้เกิดการสั่นขึ้นก่อนจะเข้าสู่ระดับปกติหรือไม่ก็ไม่สามารถติดตามต่อไปได้ จึงได้พยายามหาวิธีใหม่ที่จะทำให้ระบบสามารถติดตามลายเส้นได้ทุกรูปแบบ

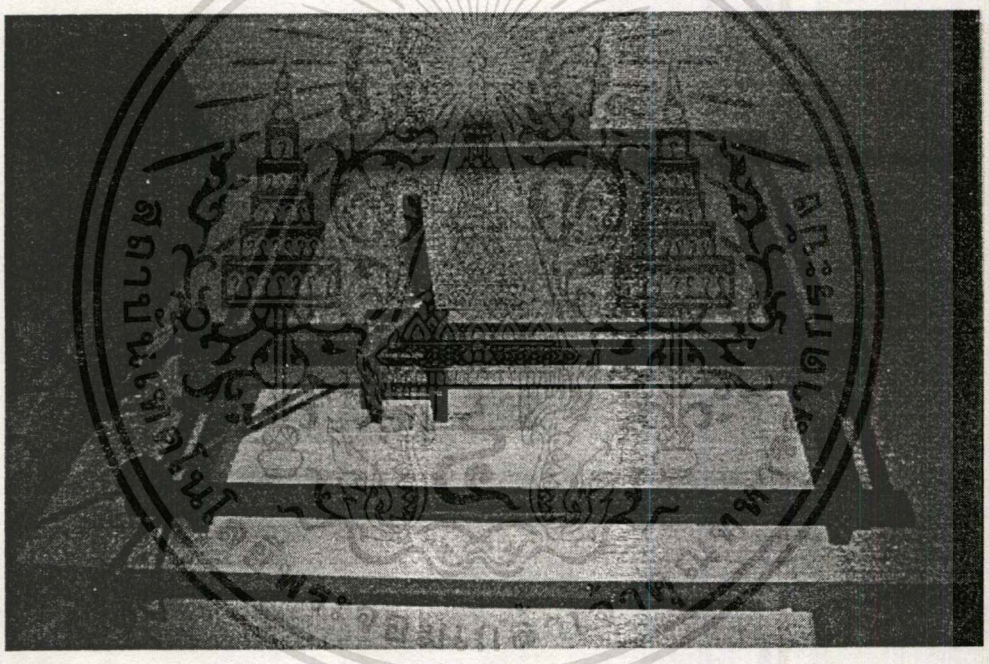
วิธีการวิ่งตามทางลายเส้นแบบสะท้อนแสงร่วมกับการตัดสินใจทางโปรแกรม เป็นวิธีที่เน้นให้สามารถติดตามลายเส้นได้ทุกรูปแบบแต่มีข้อเสียคือจะเกิดการตามลายเส้นแบบฟันเลื่อยทำให้ไม่เหมาะสำหรับการตัดในทันที แต่เหมาะสำหรับการติดตามลายเส้นเพื่อการบันทึกแล้วสั่งตัดทีหลัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3
การออกแบบและสร้างชิ้นส่วนต่าง ๆ ของชิ้นงานเพื่อให้เหมาะสมกับระบบ

3.1 การออกแบบและสร้างระบบกลไกขับเคลื่อนหัวอ่าน

ระบบขับเคลื่อนหัวอ่านมีขนาด 1.2*1.2 ม.^{2} มีโครงสร้างดังรูปที่3.1โครงสร้างทำด้วยอลูมิเนียมทั้งหมด ใช้สเต็ปปิ้งมอเตอร์ขนาด12V ความละเอียด 200 สเต็ปต่อรอบ ขับเคลื่อนโดยอาศัยแกนหมุนที่มีเส้นผ่าศูนย์กลางขนาด 15 mm ขับสายสลิงทั้ง 2 แกน ซึ่งให้ความละเอียดในการเคลื่อนที่สูงมาก ความนิ่มนวลในการเคลื่อนที่ทัดเทียมกับระบบ D.C. MOTOR ทีเดียว มีความคลาดเคลื่อนทางตำแหน่งน้อยมาก ทำให้เหมาะกับการติดตามลายเส้นที่มีขนาดเล็ก



รูปที่ 3.1 รูปถ่ายแสดงโครงสร้างของระบบขับเคลื่อนหัวอ่าน

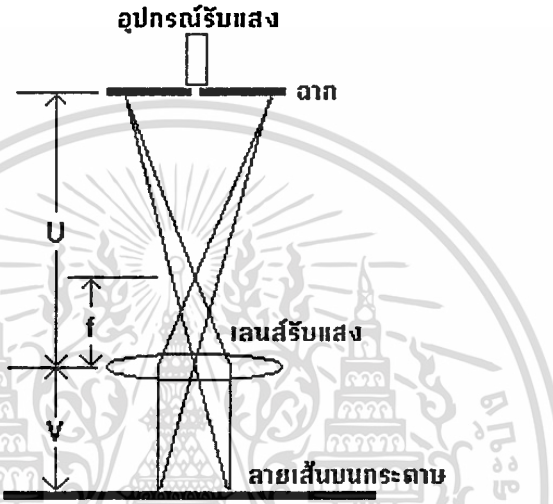
จากการที่ใช้รอกขับเคลื่อนขนาด 15 mm จะได้ว่าหนึ่งรอบการหมุนจะได้ระยะทาง 47.124 mm ต่อ 200 สเต็ป ดังนั้น 1 สเต็ปจะเคลื่อนที่ได้ 0.236 mm ซึ่งยังมีความเป็นสเต็ปอยู่ จึงได้อาศัยระบบทางอิเลคทรอนิกส์เป็นตัวช่วยปรับความละเอียดต่อรอบจาก 200 สเต็ปต่อรอบ เป็น 6400 สเต็ปต่อร์อบ ซึ่งจะได้อธิบายหลักการในบทที่ 3.5

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2 การออกแบบและสร้างหัวอ่าน

หัวอ่านที่ตัดสินใจสร้างใช้งานนี้เป็นแบบมีเซลรับแสงเพียงตัวเดียวทำให้สร้างได้ง่าย(แต่ลำบากในการเขียนโปรแกรมติดตามลายเส้นพอสมควร)ประกอบด้วยส่วนสำคัญ 2 ส่วนคือ

3.2.1 ส่วนรับแสง ประกอบด้วยเลนส์นูนทำหน้าที่ขยายขนาดภาพของลายเส้นขึ้นประมาณ 2 เท่า โดยอยู่ระหว่างลายเส้นและเซลรับแสงดังรูปที่ 3.2



รูปที่ 3.2 แสดงการวางตำแหน่งของส่วนต่างๆ ในหัวอ่าน

จากสูตรของเลนส์ คือ $1/f = 1/u + 1/v$ ——— (1)

โดย f = ความยาวโฟกัสของเลนส์

u = ระยะห่างจากกึ่งกลางเลนส์ถึงเซลรับแสง

v = ระยะห่างจากกึ่งกลางเลนส์ถึงลายเส้น

ขนาดที่ต้องการขยายคือ 2 เท่า ดังนั้นจะได้ว่า

$$u = 2v$$

เลนส์ที่ใช้มีค่า f = 15 m.m. แทนค่าลงในสมการ (1) จะได้ว่า

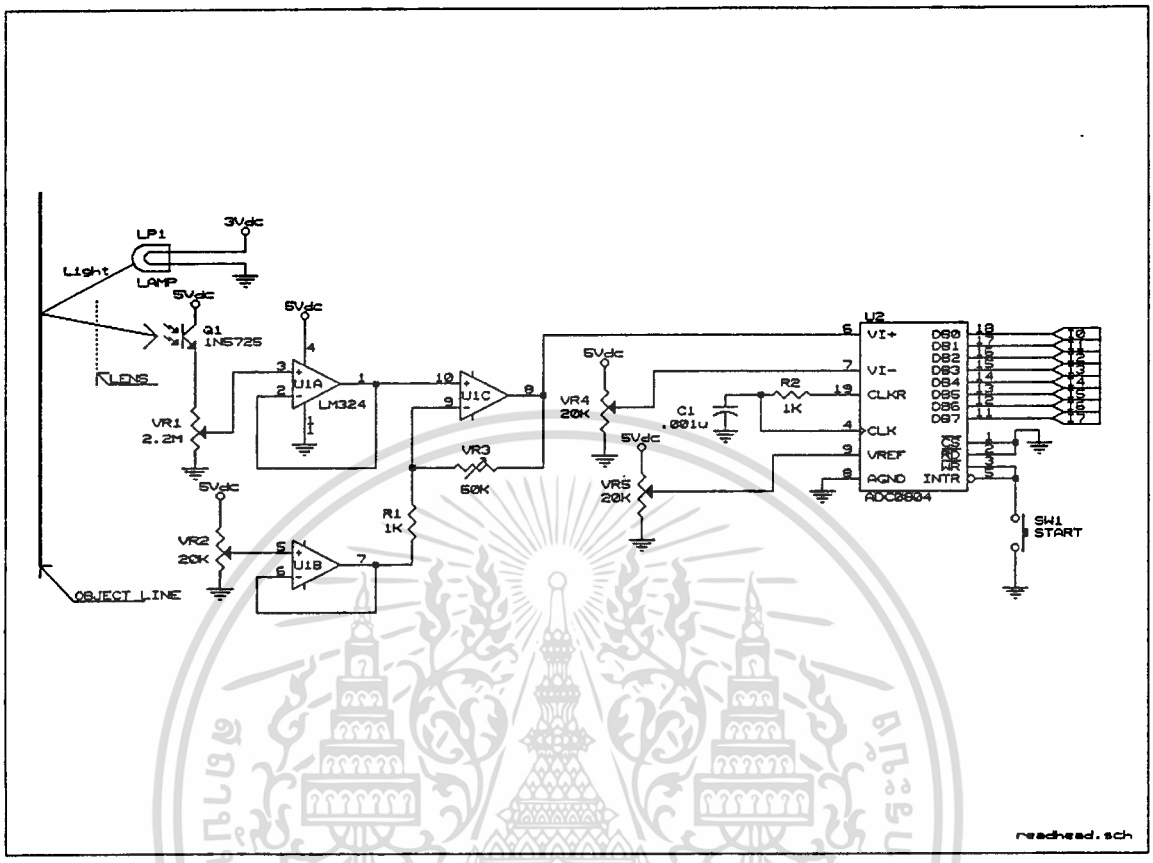
$$1/15 = 1/2v + 1/v$$

$$v = 22.5 \text{ mm.}$$

และจะได้ว่า $u = 45.0$ mm. ตามลำดับ

3.2.2 ส่วนแปลงความเข้มแสงเป็นข้อมูลดิจิตอลขนาด 8 บิต ประกอบด้วยเซลรับแสง ออปแอมป์ และตัวแปลงสัญญาณจากอนาลอกเป็นดิจิตอล(ใช้ไอซีเบอร์ ADC8004) ดังแสดงวงจรสมบูรณ์ในรูปที่ 3.3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.3 แสดงวงจรสมบูรณ์ของส่วนแปลงความเข้มแสงเป็นข้อมูลดิจิตอลขนาด 8 บิต

วงจรดังกล่าวสามารถปรับช่วงการรับแสงได้โดยปรับที่ตัวความต้านทางปรับค่าได้ VR4,V45 และความไวแสงปรับที่ VR1, VR2 และ VR3 ตามลำดับ และที่สำคัญวงจรนี้ควรใช้แหล่งจ่ายไฟแยกต่างหากเพื่อป้องกันการรบกวน โดยใช้ถ่านไฟฉายเป็นแหล่งจ่ายไฟแทนและต้องอยู่ใกล้วงจรด้วยเพื่อเสถียรภาพที่ดี

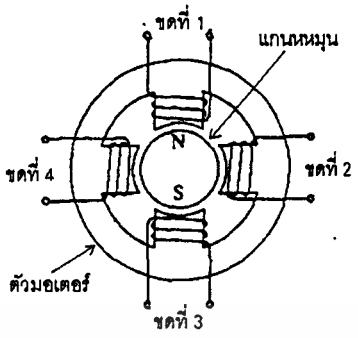
ในการทำงานของโปรแกรมติดตามลายเส้นจะมีการรับข้อมูลขนาด 8 บิตจากหัวอ่านมาทำการลบกับค่ากึ่งกลางซึ่งมีค่า 127 หมายความถ้าหัวอ่านรับแสงได้ครึ่งหนึ่งจะได้อ่านค่าความเข้มแสงได้เท่ากับ 128 ถือว่าหัวอ่านอยู่บนขอบเส้นพอดี แต่ถ้าอ่านค่าได้มากกว่า 128 จะถือว่าหัวอ่านอยู่ห่างลายเส้นมากเกินไป และถ้าอ่านค่าได้น้อยกว่า 128 ก็จะถือว่าหัวอ่านล้ำเข้าไปในเส้นมากเกินไป

3.3 การออกแบบและสร้างระบบอิเลคทรอนิกส์สำหรับขับเคลื่อนระบบกลไก

เนื่องจากระบบขับเคลื่อนที่ใช้ในงานพัฒนานี้ เลือกใช้สเต็ปปิ้งมอเตอร์เป็นตัวขับ ดังนั้นสิ่งที่เป็นอุปสรรค์ในการเคลื่อนที่ คือความเป็นสเต็ปของการเคลื่อนที่ ดังนั้นจึงได้พัฒนาระบบอิเลคทรอนิกส์ขับเคลื่อนสเต็ปปิ้งมอเตอร์แบบพิเศษขึ้นมาโดยเฉพาะ ซึ่งมีความสามารถที่จะเพิ่มความละเอียดของสเต็ปต่อรอบได้มากขึ้นหลายเท่า เป็นผลให้สเต็ปปิ้งมอเตอร์มีลักษณะการหมุนที่นุ่มนวลมากขึ้น ซึ่งต่อไปนี้จะอธิบายหลักการของระบบอิเลคทรอนิกส์ดังกล่าว

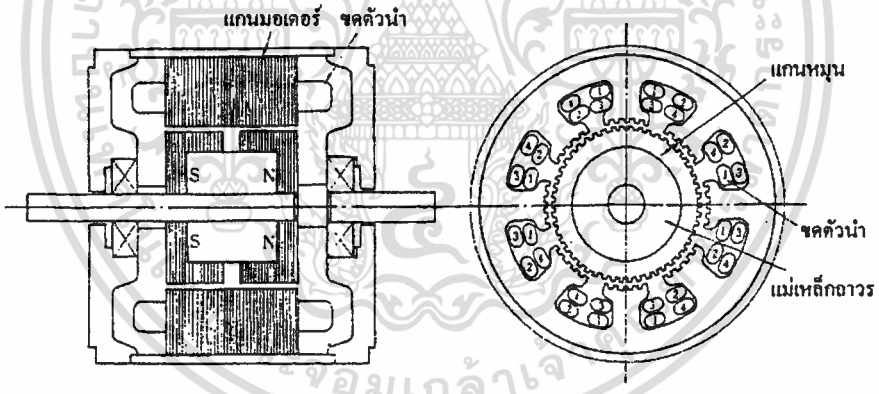
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สเต็ปปิ้งมอเตอร์มาตราฐานจะมีโครงสร้างพื้นฐานดังรูปที่ 3.4



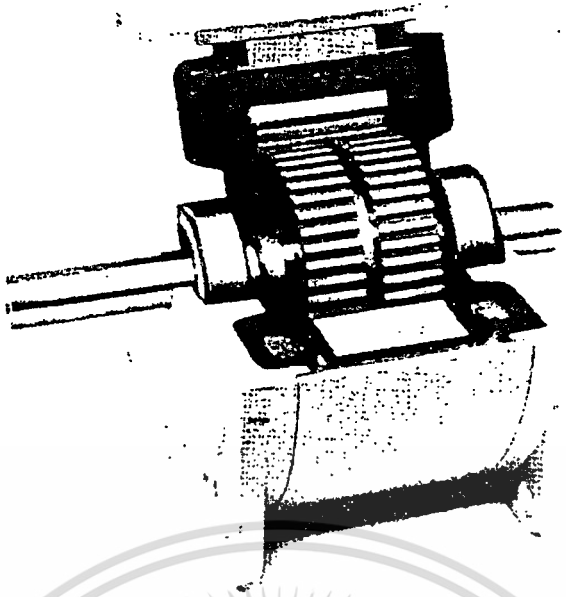
รูปที่ 3.4 แสดงโครงสร้างพื้นฐานของสเต็ปปิ้งมอเตอร์

จากรูปแกนหมุน(ROTOR)จะเป็นแม่เหล็กถาวร และมีขดลวด 4 ชุดอยู่ที่ตัวมอเตอร์(STATOR)จากโครงสร้างแบบนี้จะได้สเต็ปปิ้งมอเตอร์ชนิด 4 สเต็ปต่อรอบ ถ้าต้องการเพิ่มความละเอียดก็สามารถทำได้ดังรูปที่ 3.5



รูปที่ 3.5 ก. แสดงโครงสร้างประยุกต์ของสเต็ปปิ้งมอเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า



รูปที่ 3.5 ข. ภาพผ่าตัดสเต็ปปิ้งมอเตอร์ที่ใช้งานทั่วไป

หลักการทำงานของสเต็ปปิ้งมอเตอร์เริ่มด้วยการป้อนกระแสไฟฟ้าให้แก่ขดลวดชุดใดชุดหนึ่งก่อน ซึ่งจะเกิดแนวสนามแม่เหล็กผ่านแกนมอเตอร์ ทำให้แกนมอเตอร์ซึ่งเป็นแม่เหล็กถาวรมีการวางตัวตามแนวเส้นแรงแม่เหล็กไฟฟ้าที่สร้างขึ้น ดังนั้นถ้าต้องการให้มอเตอร์มีการหมุนก็สามารถทำได้โดยการเปลี่ยนมุมเส้นแรงแม่เหล็กที่ผ่านแกนมอเตอร์ ซึ่งก็คือการป้อนกระแสให้แก่ขดลวดที่อยู่ถัดไปแทนขดลวดเดิมเท่านั้นเอง เมื่อพิจารณาการเคลื่อนที่ที่เกิดขึ้นจากวิธีดังกล่าวจะเห็นว่าความละเอียดของการเคลื่อนที่นั้นจะขึ้นอยู่กับโครงสร้างของตัวมอเตอร์ ถ้าต้องการความละเอียดมากก็ต้องอาศัยการสร้างที่ละเอียดมากขึ้น ซึ่งจะทำให้เกิดผลเสียทางด้านความเร็วสูงสุดที่จะลดลงอันเนื่องมาจากความต้านทานเหนี่ยวนำ รวมทั้งความสามารถในการรักษาตำแหน่งมุมของมอเตอร์ต่อภาระแรงบิดต่างๆก็จะลดลงด้วย ดังนั้นเพื่อการแก้ไขปัญหาดังกล่าวจึงได้คิดวิธีการจ่ายกระแสในแต่ละขดลวดเพื่อทำให้สามารถสร้างแนวเส้นแรงแม่เหล็กในมุมต่างๆให้ละเอียดมากขึ้น ซึ่งวิธีที่ใช้ในงานวิจัยนี้จะใช้วิธีการที่เรียบง่ายเพื่อความสะดวกในการสร้างระบบซึ่งจะอธิบายได้ดังต่อไปนี้

ตามปกติในการจ่ายกระแสให้แก่มอเตอร์เท่าที่พบมีอยู่ 4 แบบคือ

ก.การจ่ายกระแสแบบเฟสเดียว(ONE EXCITE) เป็นการจ่ายกระแสแบบทีละเฟส(ในเฟสหนึ่งอาจจะมีขดลวดมากกว่าหนึ่งขดโดยทำหน้าที่สร้างสนามแม่เหล็กร่วมกัน)

ข.การจ่ายกระแสแบบเฟสคู่(TWO EXCITE) เป็นการจ่ายกระแสแบบทีละสองเฟส มีผลดี-ในแง่ที่ว่าเกิดแรงเหนี่ยวนำ(แรงบิด)สูงกว่าวิธีแรก ทำให้รักษาตำแหน่งได้ดีโดยมีความละเอียดเท่าเดิม

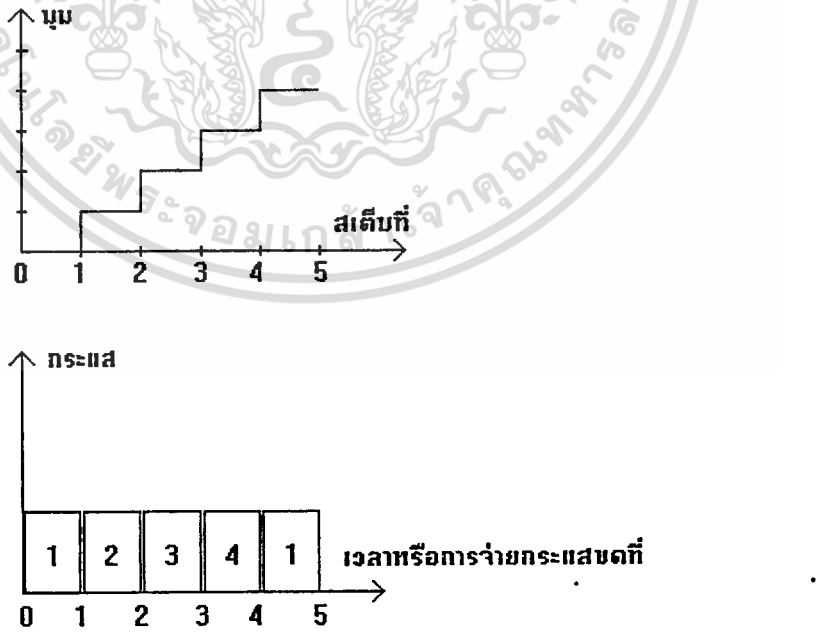
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ค.การจ่ายกระแสแบบผสม(ONE-TWO EXCITE) เป็นการจ่ายกระแสแบบทีละหนึ่งเฟสและสองเฟสสลับกันไป ซึ่งจะทำให้ได้ความละเอียดเพิ่มเป็นสองเท่า แต่แรงบิดจะไม่คงที่

ในการจ่ายกระแสทั้ง 3 แบบข้างต้นเป็นวิธีที่ค่อนข้างจะแพร่หลาย สามารถศึกษาหารายละเอียดได้ในหนังสือที่เกี่ยวกับการขับเคลื่อนสเต็ปปิ้งมอเตอร์ทั่วไป ดังนั้นจึงขอละรายละเอียดไว้ ต่อไปจะเป็นกล่าวการจ่ายการกระแสอีกแบบหนึ่ง ซึ่งมีการนำมาใช้ในงานวิจัยนี้

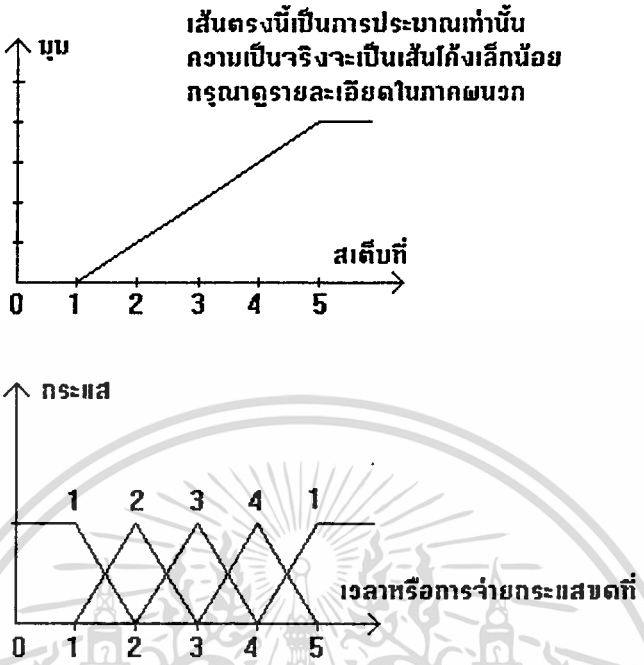
ง.การจ่ายกระแสในแต่ละเฟสให้สัมพันธ์กัน เป็นการจ่ายกระแสในแต่ละเฟสให้สัมพันธ์กันเพื่อให้สามารถควบคุมแนวเส้นแรงแม่เหล็กในมุมต่างๆ ได้ละเอียดมากขึ้น ซึ่งวิธีนี้ยังไม่เป็นที่รู้จักกันทั่วไป แต่มีการนำมาใช้งานหลายปีแล้ว ตัวอย่างเช่นในระบบพล็อตเตอร์(PLOTER)ที่ใช้กับเครื่องคอมพิวเตอร์ตั้งโต๊ะทั่วไป รวมถึงงานด้านอุตสาหกรรม ถ้าท่านมีสนใจในรายละเอียดข้อมูลของมอเตอร์ต่างๆ ที่ถูกแสดงในงานแสดงสินค้าอุตสาหกรรม ท่านอาจพบว่ามีบางบริษัทระบุว่าสเต็ปปิ้งมอเตอร์ของเขาสามารถหมุนได้ด้วยความละเอียดถึง 12800 สเต็ปต่อรอบหรืออาจจะมากกว่านี้ ทั้งที่เมื่อพิจารณาข้อมูลของตัวมอเตอร์เองแล้วพบว่ามีโครงสร้างทางกายภาพที่ละเอียดเพียง 200 สเต็ปต่อรอบเท่านั้น ซึ่งความเป็นจริงแล้วหัวใจของความละเอียดสูงดังกล่าวนั้นอยู่ที่ตัวเครื่องจ่ายกระแสเอง แต่อย่างไรก็ตามความเป็นเชิงเส้นของความละเอียดที่ได้จะดีหรือไม่ ก็ขึ้นอยู่กับหลักการที่ใช้ในการจ่ายกระแสว่าเหมาะสมกับชนิดของสเต็ปปิ้งมอเตอร์นั้นหรือไม่

ต่อไปนี้จะเป็นการกล่าวถึงเทคนิคที่ใช้ในการเพิ่มความละเอียดเชิงมุมของสเต็ปปิ้งมอเตอร์ เพื่อให้เกิดความเข้าใจโปรดพิจารณารูปกราฟต่อไปนี้



รูปที่ 3.6 กราฟแสดงความสัมพันธ์ระหว่างมุมกับการจ่ายกระแสแบบปกติ

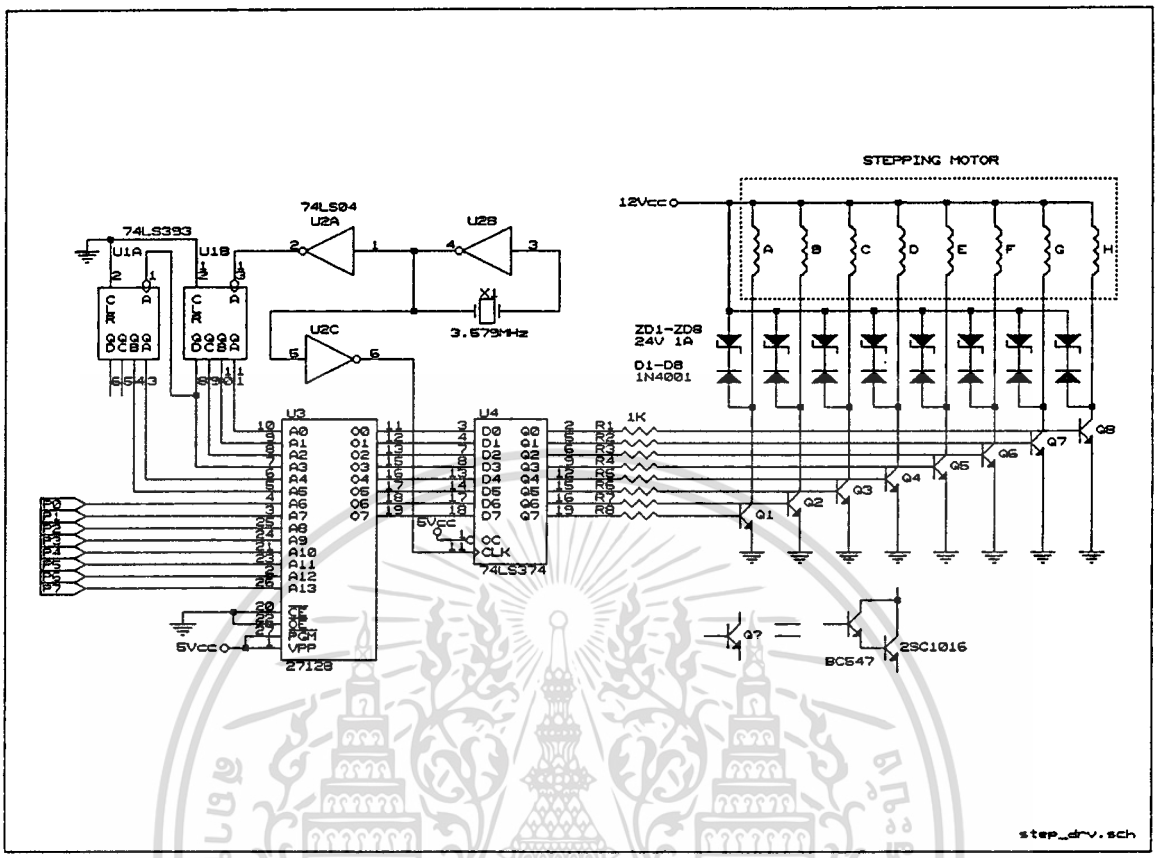
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.7 กราฟแสดงความสัมพันธ์ระหว่างมุมกับการจ่ายกระแสแบบสัมพันธ์

จากรูปที่ 3.6 จะพบว่าการจ่ายกระแสแบบปกติจะทำให้แกนมอเตอร์มีช่วงองศาเปลี่ยนมุมที่กว้าง ทำให้เกิดลักษณะขั้นบันไดที่ชัดเจน ในขณะที่การจ่ายกระแสแบบสัมพันธ์กัน จะให้ช่วงองศาเปลี่ยนมุมที่แคบกว่ากันมาก (ดูรูปที่ 3.7) ทำให้เกิดลักษณะเชิงเส้นที่ดีขึ้น การจ่ายกระแสแบบสัมพันธ์กันแบบง่ายๆ ที่ใช้นี้อธิบายได้ว่า เมื่อจะมีการเปลี่ยนมุมจะมีการลดกระแสในเฟสหนึ่งในขณะเดียวกันก็จะมีการเพิ่มกระแสในอีกเฟสหนึ่งโดยให้กระแสรวมมีค่าคงที่ตลอดเพื่อรักษาแรงบิดเฉลี่ยให้คงที่ ดังนั้นจากหลักการดังกล่าวทำให้เราสามารถอ้างอิงตำแหน่งมุมที่อยู่ระหว่างมุมปกติได้ โดยที่ความละเอียดเชิงมุมจะขึ้นอยู่กับความละเอียดทางด้านการจ่ายกระแสในแต่ละเฟส ในงานนี้เราได้ออกแบบระบบอิเลคทรอนิกส์ที่จ่ายกระแสในลักษณะคาบเวลา(PWM)ที่ปรับได้ 32 ขั้น ดังนั้นแสดงว่าในระบบนี้เราสามารถแบ่งสเต็ปปกติได้เป็น 32 สเต็ปย่อย และมอเตอร์ที่ใช้ในการนี้มีความละเอียดปกติเท่ากับ 400 สเต็ปต่อรอบ สรุปแล้วจะได้ความละเอียดเพิ่มขึ้นเป็น 400*32 = 12800 สเต็ปต่อรอบ ทำให้ได้ความละเอียดในการเคลื่อนที่เท่ากับ 15*3.1416/12800 = 3.682 µm (ค่า 15 mm มาจากเส้นผ่าศูนย์กลางตัวรอกของแกนมอเตอร์) ซึ่งจะเห็นว่าเป็นค่าที่ละเอียดมากแต่ในความเป็นจริงแล้วสเต็ปที่ได้ในแต่ละสเต็ปย่อยมีความไม่เท่ากันอยู่ แต่ก็มีผลเสียเพียงเล็กน้อยเท่านั้น เมื่อเทียบกับความนิ่มนวลที่เพิ่มขึ้นอย่างมากจนเทียบได้กับ D.C.MOTOR เสียเปรียบอยู่ตรงที่หมุนรอบสูงได้ช้ากว่าเท่านั้นเอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



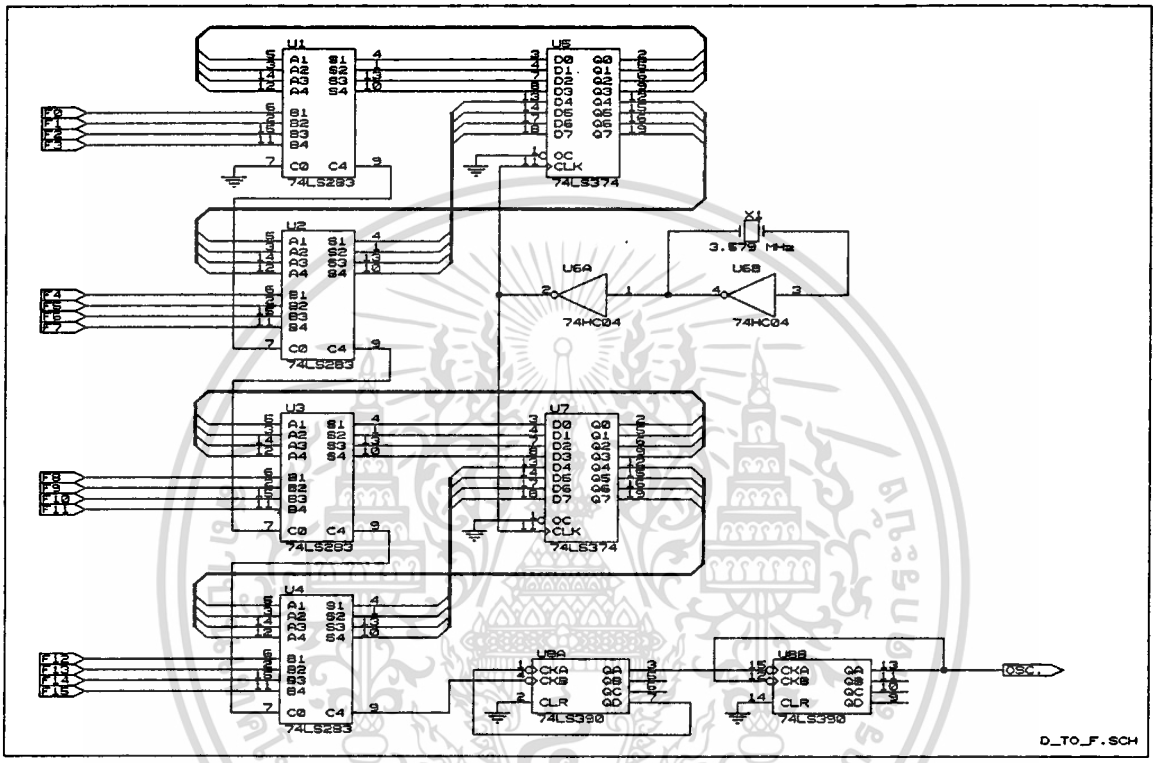
รูปที่ 3.8 แสดงวงจรขับกระแสสเต็ปปิ้งมอเตอร์

วงจรขับกระแสสเต็ปปิ้งที่ออกแบบไว้จะประกอบด้วย หน่วยความจำถาวรที่บรรจุข้อมูลสำหรับการจ่ายกระแสแก่ขดลวดในแต่ละขดที่มุมสเต็ปย่อยต่างๆ หน่วยกำเนิดความถี่ผลิตความถี่ขนาด 3.579 MHz ป้อนให้แก่วงจรนับขนาด 8 บิทแต่จะมีการใช้เพียง 6 บิทเท่านั้น

หลักการทำงานของวงจรจะเริ่มด้วยการรับข้อมูลขนาด 8 บิท เพื่อใช้กำหนดตำแหน่งมุมของสเต็ปปิ้งมอเตอร์และอีก 6 บิทจากวงจรนับ จากผลของการนับจะทำให้มีการเปลี่ยนแปลงข้อมูลที่ได้จากหน่วยความจำในลักษณะวนรอบตลอดเวลา ซึ่งก็คือข้อมูลสำหรับการเปิดปิดกระแสแต่ละขดลวดที่มุมนั้นๆ เพื่อทำให้แกนมอเตอร์อยู่ในมุมที่กำหนด ดังนั้นเมื่อมีการเปลี่ยนแปลงข้อมูลขนาด 8 บิทก็จะทำให้ตำแหน่งมุมมีการเปลี่ยนไปเช่นกัน เมื่อได้วงจรขับกระแสสเต็ปปิ้งมอเตอร์แล้ว ก็จำต้องมีวงจรควบคุมอีกชุดหนึ่งทำหน้าที่รับคำสั่งจากคอมพิวเตอร์มาควบคุมวงจรขับกระแสสเต็ปปิ้งอีกต่อหนึ่ง ทั้งนี้เพราะการควบคุมวงจรขับกระแสสเต็ปปิ้งมอเตอร์จำเป็นต้องมีการควบคุมตำแหน่งมุมอยู่ตลอดเวลา ดังนั้นถ้าให้คอมพิวเตอร์จัดการในส่วนนี้เองแล้วจะทำให้การตอบสนองของระบบโดยรวมลดลงอย่างมากจนอาจจะควบคุมระบบได้ไม่ดี อีกทั้งอาจเป็นปัญหาต่อการเขียนโปรแกรมอีกด้วย ในส่วนของวงจรที่สร้างขึ้นนี้จะทำหน้าที่รับข้อมูลความเร็วรอบของ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

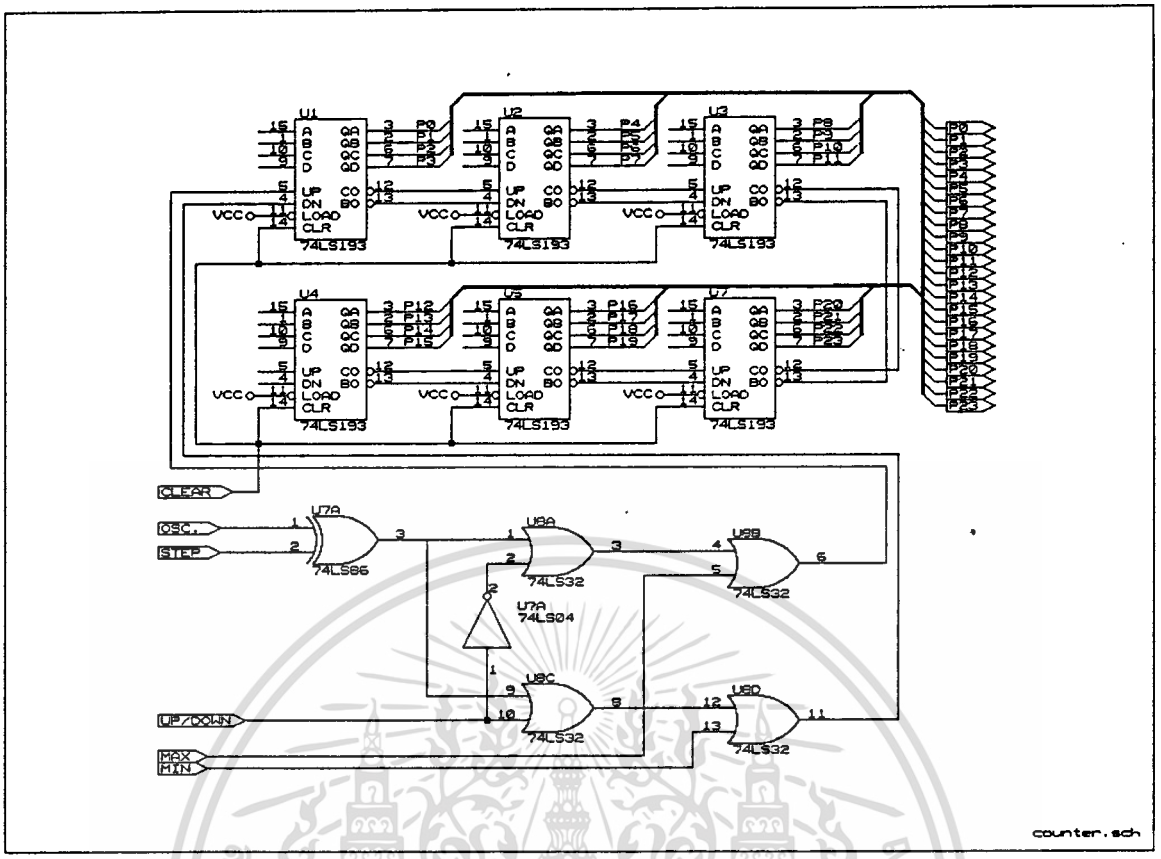
มอเตอร์จากคอมพิวเตอร์มาทำการสร้างความถี่ขึ้น และทำการนับความถี่ที่ได้นี้ส่งให้กับคอมพิวเตอร์รับรู้ซึ่งจะถือเป็นตำแหน่งอ้างอิงของตำแหน่งหัวอ่านลายเส้นได้ รวมทั้งส่งข้อมูลตำแหน่งอ้างอิงนี้ให้กับวงจรขับกระแสมอเตอร์เพื่อใช้ในการกำหนดตำแหน่งมุมแกนหมุนของสเต็ปปิ้งมอเตอร์ต่อไป ซึ่งวงจรดังกล่าวมีรายละเอียดดังรูปที่ 3.9 และ 3.10



รูปที่ 3.9 แสดงวงจรแปลงข้อมูล 16 บิทเป็นความถี่

วงจรแปลงข้อมูลเป็นความถี่จะประกอบด้วยวงจรบวกเลขขนาด 16 บิท ซึ่งรับข้อมูล 16 บิทจากผลลัพธ์เป็นตัวตั้งและข้อมูล 16 บิทจากคอมพิวเตอร์เป็นตัวบวก ทำให้เกิดการบวกในลักษณะสะสมขึ้นเมื่อผลบวกขึ้นถึงจุดสูงสุดก็จะวนกลับสู่ค่าต่ำสุด เป็นการเลียนแบบการสร้างสัญญาณฟันเลื่อนในวงจรเชิงเส้นความถี่ที่ได้จะได้จากบิทหดดังนั้นถ้าข้อมูลมากขึ้นก็จะทำให้ความถี่มีค่าสูงขึ้นด้วย และมีวงจรผลิตความถี่ขนาด 3.579 MHz ป้อนให้กับวงจรบวกเลขเป็นการกำหนดอัตราการบวกเลข ดังนั้นความถี่ที่ได้จากวงจรนี้จะมีค่าต่ำสุดเท่ากับ 0Hz และสูงสุดเท่ากับ 1.7895MHz โดยการป้อนข้อมูล16บิทเท่ากับ 0111 1111 1111 1111 และ 0000 0000 00000000ตามลำดับ อย่างไรก็ตามความเป็นเชิงเส้นระหว่างข้อมูลกับความถี่จะมีค่าที่ดีเยี่ยมเมื่อข้อมูลมีค่าน้อยลง (หมายถึงว่าสัญญาณจะมีค่า DUTY CYCLE ใกล้เคียงกับ 50/50 มาก)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

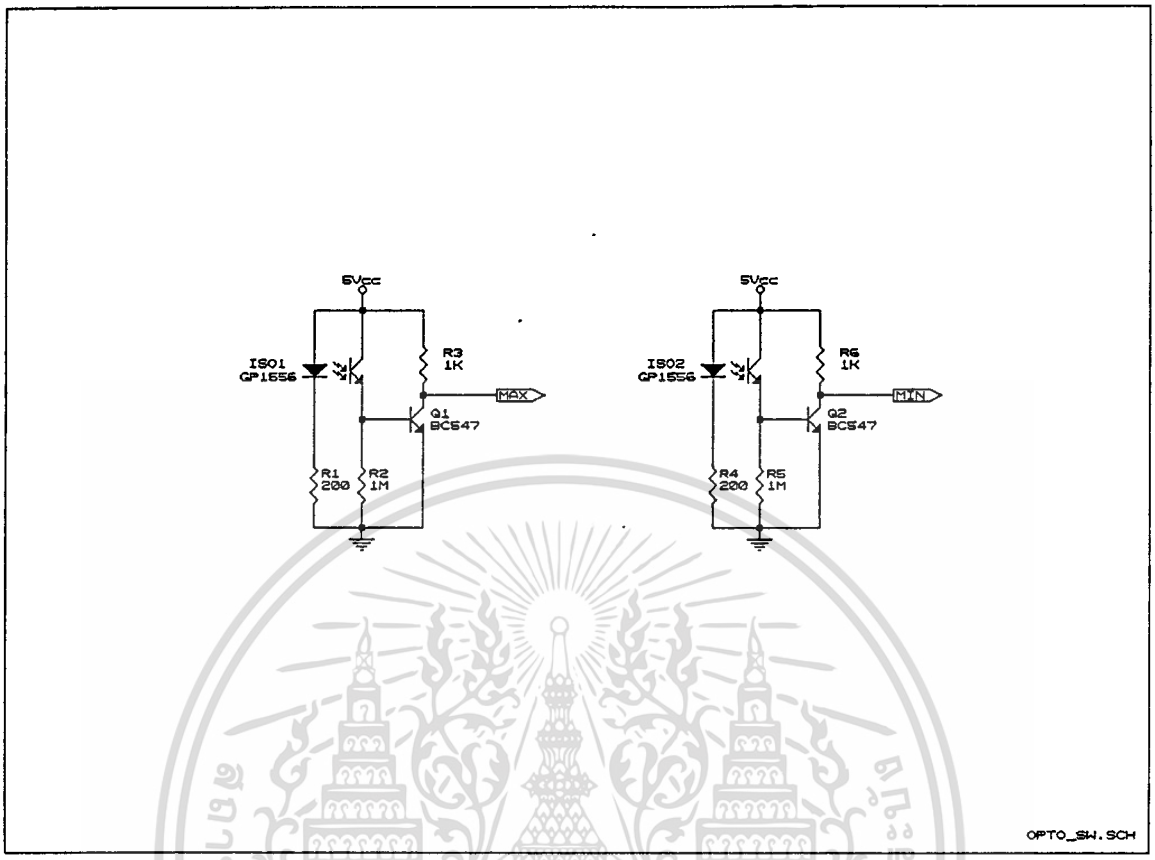


รูปที่ 3.10 แสดงวงจรนับความถี่แปลงเป็นตำแหน่งอ้างอิงของมอเตอร์

วงจรนับความถี่ทำหน้าที่นับสัญญาณที่ได้จากวงจรนับความถี่ ค่าที่นับได้จะใช้สำหรับบอกตำแหน่งอ้างอิงของมอเตอร์ว่าหมุนไปกี่รอบแล้ว ทำให้คอมพิวเตอร์สามารถคำนวณหาตำแหน่งที่แท้จริงของมอเตอร์ได้ นอกจากนั้นวงจรนับความถี่ก็จะส่งผลการนับขนาด 8 บิท(P0-P7) เพื่อใช้ในการอ้างอิงตำแหน่งมุมสำหรับวงจรจ่ายกระแสต่อไป

วงจรตรวจจุดเริ่มต้นและจุดสิ้นสุดของการเคลื่อนที่ของหัวอ่านก็มีความสำคัญมากเพราะต้องใช้สำหรับอ้างอิงการเริ่มต้นระบบรวมทั้งป้องกันความเสียหายที่เกิดจากการเคลื่อนที่เกินขอบเขต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า



รูปที่ 3.11 แสดงวงจรตรวจจับจุดเริ่มต้นและจุดสิ้นสุดสำหรับ 1 แกน

สำหรับวงจรแหล่งจ่ายไฟนั้นจะขอละไว้ในที่นี้เพราะเป็นวงจรพื้นฐาน ซึ่งประกอบด้วยหม้อแปลงไฟ วงจรเรียงกระแสขนาด 2 แอมแปร์ ตัวเก็บประจุขนาด 10000 ไมโครฟารัดสำหรับสำรองกระแสไฟให้แก่มอเตอร์และวงจรไอซีปรับแรงดันคงที่ขนาด 12 โวลต์(LM7812)และ 5 โวลต์(LM 7812)สำหรับไฟเลี้ยงวงจรต่างๆ ตามลำดับ

3.4 การออกแบบและสร้างวงจรอิเลดทรอนิกส์สำหรับเชื่อมโยงการติดต่อระหว่างระบบกับคอมพิวเตอร์

จากหัวข้อที่ผ่านมาสรุปได้ว่าข้อมูลที่ต้องติดต่อกับคอมพิวเตอร์ได้แก่

ก. ข้อมูลเลขฐานสองขนาด 8 บิทที่ได้จากการแปลงสัญญาณความเข้มแสงจากลายเส้น 1 ชุด

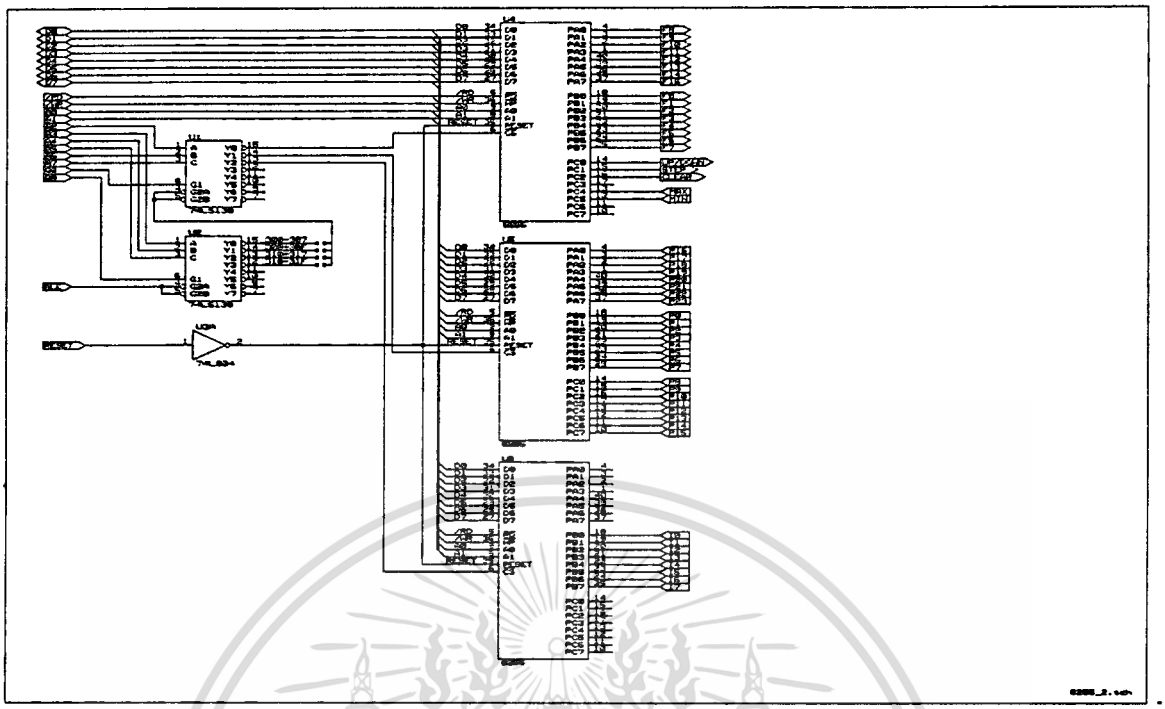
ข. ข้อมูลเลขฐานสองขนาด 24 บิทที่ได้จากวงจรนับเพื่อการอ้างอิงทางตำแหน่งของมอเตอร์ 2 ชุด

ข. วงจรควบคุมวงจรขับกระแสสเต็ปปิ้งมอเตอร์ 2 ชุด

ค. ข้อมูลย่อยที่เกี่ยวกับการบอกตำแหน่งสูงสุด,ต่ำสุดของหัวอ่าน ฯลฯ ทั้ง 2 แกน

ทำการออกแบบวงจรเชื่อมระบบกับคอมพิวเตอร์ได้ดังรูปที่ 3.12

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



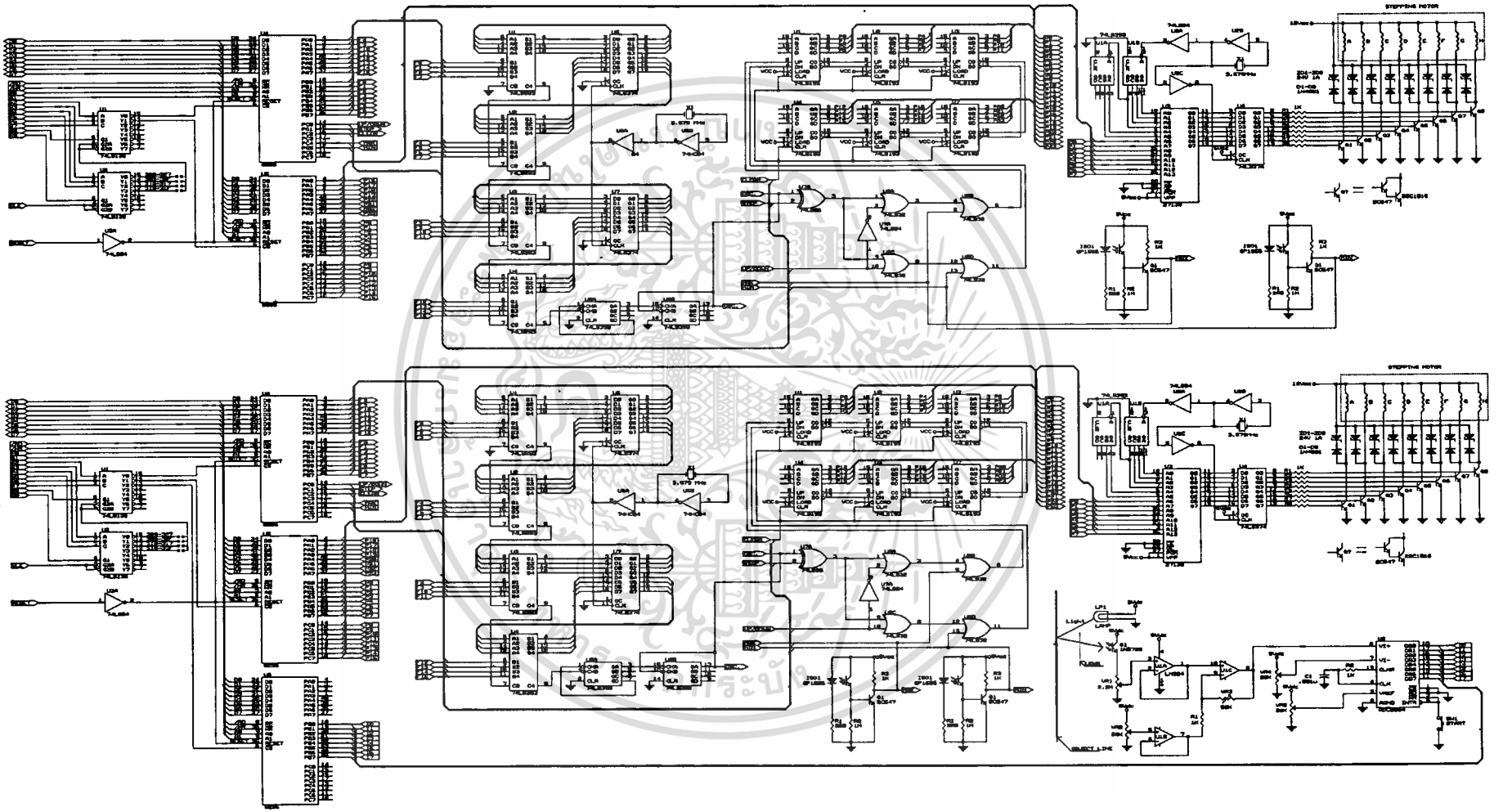
รูปที่ 3.12 แสดงวงจรเชื่อมต่อระหว่างคอมพิวเตอร์กับส่วนควบคุมต่างๆ

วงจรที่ได้จะประกอบด้วยวงจรถอดระหัสตำแหน่งเพื่อเลือกควบคุม 8255 ที่มีอยู่ 2 ตัวโดย 8255 ตัวหนึ่งจะทำหน้าที่ควบคุมการทำงานของสเต็ปปิ้งมอเตอร์ และ 8255 อีกตัวหนึ่งจะทำหน้าที่รับค่าทางตำแหน่งจากวงจรนับขนาด 24 บิท รูปที่ 3.13 เป็นการแสดงถึงการรวมระบบทั้งระบบเข้าด้วยกัน

3.5 สรุปตำแหน่งสำคัญที่ใช้สำหรับติดต่อระหว่างคอมพิวเตอร์กับระบบควบคุม

ตำแหน่งส่งข้อมูลความเร็วในแกนนอน	0x300 - 0x301
ตำแหน่งอ่านข้อมูลตำแหน่งในแนวแกนนอน	0x305 - 0x306
ตำแหน่งส่งข้อมูลความเร็วในแกนนอน	0x308 - 0x309
ตำแหน่งอ่านข้อมูลตำแหน่งในแนวแกนนอน	0x30c - 0x30e
ตำแหน่งอ่านข้อมูลความเข้มแสงจากหัวอ่าน	0x311

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.13 แสดงวงจรควบคุมระบบที่สมบูรณ์

บทที่ 4
การออกแบบขั้นตอนการทำงานของโปรแกรมคอมพิวเตอร์

4.1 การออกแบบโปรแกรมย่อยควบคุมการเคลื่อนที่ของหัวอ่าน

โดยปกติในการเขียนโปรแกรมที่ดีจำเป็นจะต้องพิจารณาว่าระบบควบคุมของเรามีความต้องการการเคลื่อนที่ที่จำเป็นในลักษณะใดบ้าง และการเคลื่อนที่พื้นฐานที่ระบบเราควรจะมีนั้นมีอะไรบ้าง ดังนั้นจากการวิเคราะห์ระบบในการควบคุมแบบต่างๆ สามารถสรุปคำสั่งที่ควรเขียนเป็นโปรแกรมย่อยสำหรับการเรียกใช้งานได้ดังต่อไปนี้

4.1.1 สรุปคำสั่งสำหรับการเขียนโปรแกรมย่อยควบคุมการเคลื่อนที่ของหัวอ่าน

คำสั่ง		หน้าที่
float	deg_to_rad(float deg)	รับค่ามุมที่มีหน่วยเป็นองศาแล้วคืนค่าเดิมในหน่วยของเรเดียน
void	go_home(void)	ทำการเคลื่อนย้ายหัวอ่านให้กลับไปอยู่ในตำแหน่งเริ่มต้น
void	initport(void)	ทำการตั้งค่าเงื่อนไขเริ่มต้นของระบบ
void	stop(void)	ทำการหยุดการเคลื่อนที่ของหัวอ่านในทันที
void	zero_port(void)	ทำการปรับวงจรนับให้เริ่มต้นนับตำแหน่งอ้างอิงที่ศูนย์
int	get_photo(void)	ทำการรับค่าที่อ่านได้จากหัวอ่านหรือหัววัดแสง
void	photo_on(void)	ทำการเปิดสวิตซ์ไฟที่หัวอ่าน
void	photo_off(void)	ทำการปิดสวิตซ์ไฟที่หัวอ่าน
long	get_x(void)	ทำการอ่านค่าตำแหน่งอ้างอิงในแกนนอนขนาด 24 บิต
long	get_y(void)	ทำการอ่านค่าตำแหน่งอ้างอิงในแกนตั้งขนาด 24 บิต
void	set_x_direction(int direct);	ทำการปรับวงจรนับให้นับขึ้นหรือนับลงเพื่อบังคับทิศทางการเคลื่อนที่ของหัวอ่านในแนวแกนนอน
void	set_y_direction(int direct);	ทำการปรับวงจรนับให้นับขึ้นหรือนับลงเพื่อบังคับทิศทางการเคลื่อนที่ของหัวอ่านในแนวแกนตั้ง
void	out_port(int port,int out);	ทำการส่งข้อมูลข้อมูลขนาด 16 บิตให้แก่ตำแหน่งระบบควบคุมต่างๆ
void	put_x_speed(float speed);	ทำการแปลงเลขจุดทศนิยมเป็นเลขฐานสอง 16 บิต แล้วส่งให้วงจรแปลงข้อมูลเป็นความถี่เพื่อปรับความเร็วของหัวอ่านในแนวแกนนอน
void	put_y_speed(float speed);	ทำการแปลงเลขจุดทศนิยมเป็นเลขฐานสอง 16 บิต แล้วส่งให้วงจรแปลงข้อมูลเป็นความถี่เพื่อปรับความเร็วของหัวอ่านในแนวแกนตั้ง
int	check_xmin(void);	ตรวจสอบว่าหัวอ่านอยู่ในตำแหน่งต่ำสุดทางแกนนอนหรือไม่
int	check_xmax(void);	ตรวจสอบว่าหัวอ่านอยู่ในตำแหน่งสูงสุดทางแกนนอนหรือไม่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

int check_ymin(void); ตรวจสอบว่าหัวอ่านอยู่ในตำแหน่งต่ำสุดทางแกนตั้งหรือไม่

int check_ymax(void); ตรวจสอบว่าหัวอ่านอยู่ในตำแหน่งสูงสุดทางแกนตั้งหรือไม่

void set_v(long xi,long yi,float vi); รับค่าตำแหน่งปลายทางของแกนตั้งและแกนนอนมาเปรียบเทียบกับตำแหน่งปัจจุบันของหัวอ่านเพื่อหาทิศทางที่จะวิ่งไป และรับค่าความเร็วจริงมาทำการปรับค่าความเร็วที่ควรจะเป็นในแต่ละแกน

void set_v2(long xi,long yi,long v); รับค่าตำแหน่งปลายทางของแกนตั้งและแกนนอนมาเปรียบเทียบกับตำแหน่งปัจจุบันของหัวอ่านเพื่อทิศทางที่จะวิ่งไป และรับค่าความเร็วมาทำการปรับค่าเป็นความเร็วที่ควรจะเป็นในแต่ละแกน

void go_rel(float xi,float yi,float vi,float ai); คำสั่งให้หัวอ่านเคลื่อนห่างจากตำแหน่งปัจจุบันไปเป็นระยะทางที่กำหนดในแต่ละแกน โดยมีความเร็วและอัตราเร่ง

void go_to(float xi,float yi,float vi,float ai); คำสั่งให้หัวอ่านเคลื่อนที่ไปยังตำแหน่งที่กำหนด โดยมีความเร็วและความเร่ง

void move_to(long xi,long yi,float vi); คำสั่งให้หัวอ่านมีทิศทางการเคลื่อนที่ไปยังจุดที่กำหนดโดยมีความเร็วคงที่

void go_zeta(float vi,float zeta); คำสั่งให้หัวอ่านมีความเร็วและทิศทางการเคลื่อนที่ตามที่กำหนด

void arc_move(float xi,float yi,float rxi,float ryi,float oi1,float oi2,float vi,float ai);
คำสั่งให้หัวอ่านทำการเคลื่อนที่เป็นเศษส่วนของวงกลมโดยมีการกำหนดค่ารัศมี มุมเริ่มต้น มุมสิ้นสุด ความเร็ว และอัตราเร่ง

จากคำสั่งต่างๆ ที่ได้สรุปไว้มีความสำคัญต่อการเขียนโปรแกรมต่อไปในส่วนต่างๆ มาก สำหรับลายละเอียดของโปรแกรมย่อยโดยสมบูรณ์จะแสดงอยู่ในภาคผนวก

4.2 การออกแบบโปรแกรมติดตามลายเส้น

4.2.1 โปรแกรมติดตามลายเส้นระบบควบคุมแบบตั้งค่าควบคุม PID ได้

โปรแกรมนี้จะมีวิธีการทำงานโดยการรรับค่า P , I , D และความเร็วหลังจากนั้นจะทำการย้ายหัวอ่านมายังจุดเริ่มต้น เปิดไฟส่องลายเส้น และเริ่มต้นวิ่งด้วยความเร็วคงที่ 10 เซ็นติเมตรต่อนาทีเป็นมุม 45 องศาจากจุดเริ่มต้นจนกระทั่งพบกับลายเส้นซึ่งหมายถึงค่าที่ได้รับจากหัวอ่านมีค่าน้อยกว่า 255 หรือ 1111 1111 ในเลขฐานสองหลังจากนั้นจะทำการเปลี่ยนทิศทางเริ่มต้นของหัวอ่านเท่ากับ 0 องศา และเริ่มเข้าสู่การติดตามลายเส้นระบบ PID ต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในการทำงานของโปรแกรมติดตามลายเส้นระบบ PID จะเริ่มต้นโดยการรับข้อมูลขนาด 8 บิตจากหัวอ่านมาทำการลบกับค่ากึ่งกลางซึ่งมีค่า 127 (หมายความถ้าหัวอ่านรับแสงได้ครึ่งหนึ่งจะได้อ่านค่าความเข้มแสงได้เท่ากับ 127 ถือว่าหัวอ่านอยู่บนขอบเส้นพอดีจะได้ค่าความผิดพลาดเท่ากับศูนย์ แต่ถ้าอ่านค่าได้มากกว่า 127 จะถือว่าหัวอ่านอยู่ห่างลายเส้นมากเกินไป ได้ค่าความผิดพลาดมากกว่าศูนย์ และถ้าอ่านค่าได้น้อยกว่า 127 ก็จะถือว่าหัวอ่านล้ำเข้าไปในเส้นมากเกินไป ทำให้ได้ค่าความผิดพลาดน้อยกว่าศูนย์ สรุปแล้วค่าความผิดพลาดที่ได้จะมีค่าอยู่ในช่วง +/- 127)ทำให้ได้ค่าผิดพลาดค่าหนึ่ง นำค่าความผิดพลาดที่ได้ใส่ลงในสมการ

$$\theta(n) = Kp * e(n) + Ki \sum_{N=0}^{n} e(n) + Kd[e(n') - e(n'-1)] \quad \text{.....(1)}$$

หลังจากนั้นก็จะนำค่าที่ได้ไปรวมกับค่าองศาของทิศทางหลักเดิมกลายเป็นทิศทางหลักค่าใหม่นำไปคำนวณหาค่าความเร็วสำหรับแกนนอนและแกนตั้ง หลังจากนั้นก็เริ่มรับค่าจากหัวอ่านใหม่และทำตามขั้นตอนที่ได้กล่าวมาแล้วไปเรื่อยๆ จนกว่าจะได้รับการขัดจังหวะจากผู้ใช้จึงจะหยุดทำงาน

อธิบายการทำงานของโปรแกรมระบบควบคุมแบบ P I D

โปรแกรมนี้จะเริ่มต้นทำงานโดยย้ายหัวอ่านไปที่ตำแหน่งเริ่มต้นทำการเปิดไฟส่องลายเส้นแล้ววิ่งออกจากจุดเริ่มต้นเป็นมุม 45 องศาที่ความเร็ว 100 มิลลิเมตรต่อนาที จนกระทั่งหัวอ่านเริ่มตอบสนองต่อลายเส้นได้(ค่าความเข้มแสงมีค่าลดลงถึงจุดหนึ่ง) ต่อจากนั้นก็จะเข้าสู่การทำงานในระบบการติดตามลายเส้น

โปรแกรมติดตามลายเส้นจะมีขั้นตอนโดยละเอียดดังต่อไปนี้

1. เมื่อเริ่มเข้าสู่โปรแกรมติดตามลายเส้นจะมีการตั้งค่าความเร็วตามที่ผู้ใช้กำหนดและให้หัวอ่านมีทิศทางการเคลื่อนที่เป็นมุมเท่ากับศูนย์องศา และตั้งค่าตัวแปรทุกตัวมีค่าเท่ากับศูนย์
2. ทำการตรวจสอบว่ามีการกดคีย์หรือไม่ ถ้ามีก็จะไปทำที่ข้อ 10.
3. ทำการอ่านค่าความเข้มแสงจากหัวอ่านมาลบกับค่ากึ่งกลาง 0x7f ก็จะได้ค่าความผิดพลาดค่าหนึ่งแทนค่าด้วยตัวแปร e
4. กำหนดให้ $\theta k = Kp * e$
5. กำหนดให้ $ei(n) = ei(n-1) + e$

   ถ้า $ei(n) > 1000$ กำหนดให้ $ei(n) = 1000$

   ถ้า $ei(n) < -1000$ กำหนดให้ $ei(n) = -1000$

   Vx กำหนดให้ $\theta i = Ki * ei(n)$
6. กำหนดให้ $ed(n) = e - ed(n-1)$

   กำหนดให้ $\theta d = Kd * ed(n)$
7. นำค่าที่ได้จากข้อ 4. 5. และ 6. มารวมกันจะได้ว่า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้ในการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$\theta m(n) = \theta m(n-1) + \theta p + \theta i + \theta d$$

8. หาค่าความเร็วในแนวตั้งจะได้ว่า

$$Vx = Vm * cos\ \theta m$$

หาค่าความเร็วในแนวนอนจะได้ว่า

$$Vy = Vm * sin\ \theta m$$

9. ทำการส่งค่า Vx ให้แก่แนวนอน

ทำการส่งค่า Vy ให้แก่แนวตั้ง

เริ่มทำที่ข้อ 2. ใหม่

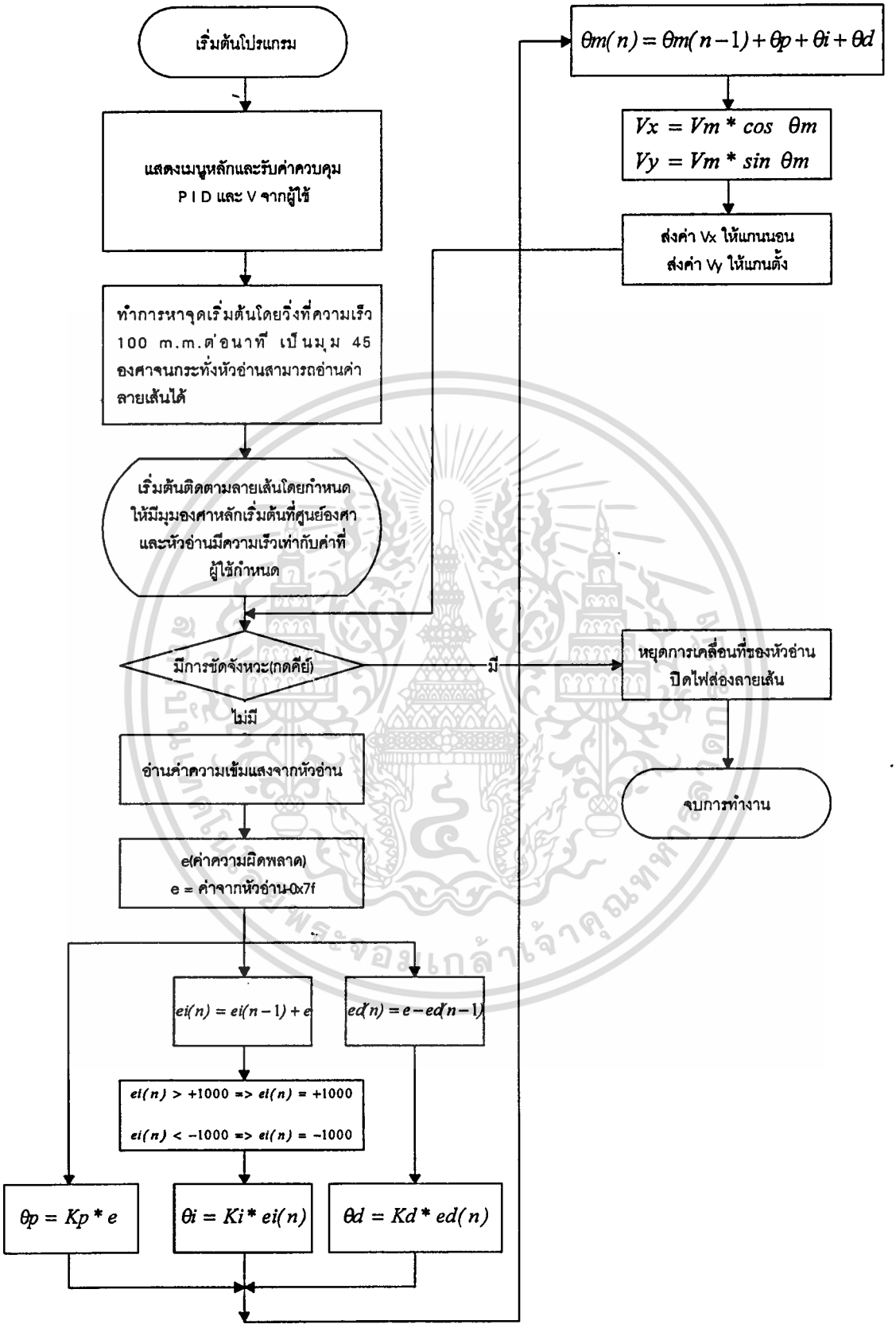
10. ทำการปิดไฟส่องลายเส้น

หยุดการเคลื่อนที่ของหัวอ่าน

หลับสู่โปรแกรมหลัก

จากการออกแบบขั้นตอนการทำงานดังกล่าวสามารถนำมาเขียนเป็นโครงสร้างทางโปรแกรมได้ดังรูปที่ 4.1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.1 แสดงโครงสร้างโปรแกรมของระบบควบคุมแบบ PID

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จุดสังเกต จากโปรแกรมจะเห็นว่าค่าองศาของทิศทางหลักจะมีการเปลี่ยนแปลงตามค่าความผิดพลาดตลอดเวลาทำให้มีลักษณะของการอินทีเกรตในระบบควบคุมด้วย

ความสามารถในการติดตามลายเส้นจะดีหรือไม่ ขึ้นอยู่กับการตั้งค่าตัวแปรควบคุมในสมการความเร็วในการเคลื่อนที่ของหัวอ่าน และความเร็วของคอมพิวเตอร์เป็นสำคัญ

4.2.2 โปรแกรมติดตามลายเส้นระบบควบคุมแบบกึ่งตัดสินใจร่วมกับการควบคุมแบบ P

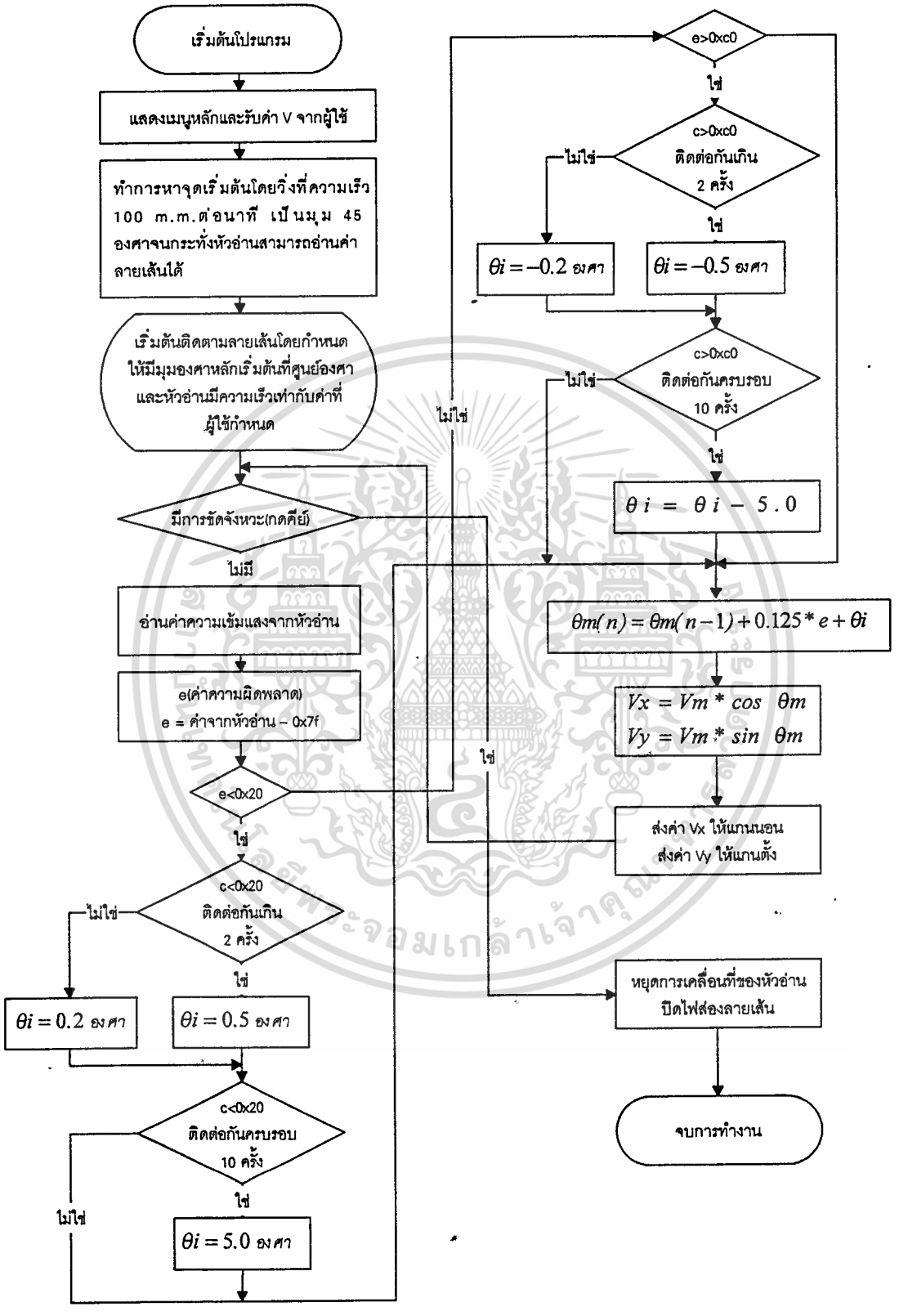
โปรแกรมนี้จะเริ่มต้นทำงานโดยย้ายหัวอ่านไปที่ตำแหน่งเริ่มต้นทำการเปิดไฟส่องลายเส้นแล้ววิ่งออกจากจุดเริ่มต้นเป็นมุม 45 องศาที่ความเร็ว 100 มิลลิเมตรต่อนาที จนกระทั่งหัวอ่านเริ่มตอบสนองต่อลายเส้นได้(ค่าความเข้มแสงมีค่าลดลงถึงจุดหนึ่ง) ต่อจากนั้นก็จะเข้าสู่การทำงานในระบบการติดตามลายเส้น

โปรแกรมติดตามลายเส้นจะมีขั้นตอนโดยละเอียดดังต่อไปนี้

1. เมื่อเริ่มเข้าสู่โปรแกรมติดตามลายเส้นจะมีการตั้งค่าความเร็วตามที่ผู้ใช้กำหนดและให้หัวอ่านมีทิศทางการเคลื่อนที่เป็นมุมเท่ากับศูนย์องศา และตั้งค่าตัวแปรทุกตัวมีค่าเท่ากับศูนย์
2. ทำการตรวจสอบว่ามีการกดคีย์หรือไม่ ถ้ามีก็จะไปทำที่ข้อ 14.
3. ทำการอ่านค่าความเข้มแสงจากหัวอ่านมาลบกับค่ากึ่งกลาง 0x7f ก็จะได้ค่าความผิดพลาดค่าหนึ่งแทนค่าด้วยตัวแปร e และให้ตัวแปร i เก็บค่าความเข้มแสงจากหัวอ่าน
4. ถ้าค่า i มีค่ามากกว่า 0x20 ให้ไปทำงานที่ข้อ 10.
5. ถ้าค่า i มีค่าน้อยกว่า 0x20 ติดต่อกันเกิน 2 ครั้ง กำหนดให้ $\theta i = 0.5$ องศา
   แต่ถ้าไม่เกิน กำหนดให้ θi = 0.2 องศา
6. ถ้าค่า i ยังมีค่าน้อยกว่า 0x20 ติดต่อกันเกิน 10 ครั้ง ให้เพิ่มค่า θi อีก 5.0 องศาทุกๆ 10 ครั้ง เพื่อให้หัวอ่านมีการเปลี่ยนทิศทางที่เร็วขึ้น
7. หาค่ามุมทิศทางหลัก($\theta m(n)$)ในปัจจุบันโดยกำหนดให้
   $\theta m(n) = \theta m(n-1) + 0.125 * e + \theta i$
8. คำนวณหาค่าความเร็วในแนวตั้งและแนวนอนโดยกำหนดให้
   $Vx = Vm * cos\ \theta m$ และ $Vy = Vm * sin\ \theta m$
9. ทำการส่งค่า Vx ให้แกนนอนและ Vy ให้แนวตั้ง และกลับไปทำข้อ 2.
10. ถ้าค่า i มีค่าน้อยกว่า 0xc0 ให้ไปทำงานที่ข้อ 7.
11. ถ้าค่า i มีค่ามากกว่า 0xc0 ติดต่อกันเกิน 2 ครั้ง กำหนดให้ θi = -0.5 องศา
    แต่ถ้าไม่เกินกำหนดให้ θi = -0.2 องศา
12. ถ้าค่า i ยังมีค่ามากกว่า 0xc0 ติดต่อกันเกิน 10 ครั้ง ให้เพิ่มค่า θi อีก 5.0 องศาทุกๆ 10 ครั้ง
13. กลับไปทำข้อที่ 7.
14. หยุดการเคลื่อนที่ของหัวอ่าน ปิดไฟส่องลายเส้น และออกจากโปรแกรม

จากขั้นตอนการทำงานข้างต้นสามารถเขียนโครงสร้างโปรแกรมติดตามลายเส้นระบบควบคุมแบบกึ่งตัดสินใจร่วมกับการควบคุมแบบ P ได้ดังรูปที่ 4.2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.2 แสดงโครงสร้างของโปรแกรมควบคุมแบบกึ่งตัดสินใจร่วมกับการควบคุมแบบ P

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กล่าวโดยสรุปโปรแกรมนี้จะมีการเริ่มต้นหาลายเส้นเหมือนกับวิธีแรก และเมื่อเริ่มเข้าสู่โปรแกรมการติดตามจะมีการทำงานโดยการรับค่าความผิดพลาดมาคูณกับค่า P ที่ได้ตั้งไว้แล้วนำไปรวมกับค่าองศาทิศทางหลัก นำค่าที่ได้ใหม่นี้ไปทำการหาค่าความเร็วทางแกนนอนและแกนตั้งต่อไป อย่างก็ตามโปรแกรมนี้จะมีขั้นตอนการตรวจสอบว่าค่าความผิดพลาดที่ได้มีค่าเกินที่กำหนดเป็นระยะเวลาที่กำหนดหรือไม่ ถ้าไม่เกินก็จะไม่มีผลอย่างไร แต่ถ้าเกินขึ้นมาก็จะทำการเปลี่ยนแปลงให้ค่าองศาทิศทางหลักหมุนได้เร็วกว่าปกติเพื่อให้หัวอ่านเข้าหาลายเส้นได้เร็วยิ่งขึ้น

4.2.3โปรแกรมติดตามลายเส้นที่ใช้วิธีการวิ่งตามลายเส้นแบบสะท้อนแสงร่วมกับการตัดสินใจทางโปรแกรม

โปรแกรมนี้จะเริ่มต้นทำงานโดยย้ายหัวอ่านไปที่ตำแหน่งเริ่มต้นทำการเปิดไฟส่องลายเส้นแล้ววิ่งออกจากจุดเริ่มต้นเป็นมุม 45 องศาที่ความเร็ว 100 มิลลิเมตรต่อนาที จนกระทั่งหัวอ่านเริ่มตอบสนองต่อลายเส้นได้(ค่าความเข้มแสงมีค่าลดลงถึงจุดหนึ่ง) ต่อจากนั้นก็จะเข้าสู่การทำงานในระบบการติดตามลายเส้น

โปรแกรมติดตามลายเส้นจะมีขั้นตอนโดยละเอียดดังต่อไปนี้

1. เมื่อเริ่มเข้าสู่โปรแกรมติดตามลายเส้นจะมีการตั้งค่าความเร็วตามที่ผู้ใช้กำหนดและให้หัวอ่านมีทิศทางการเคลื่อนที่เป็นมุมเท่ากับศูนย์องศา และตั้งค่าตัวแปรทุกตัวมีค่าเท่ากับศูนย์
2. ทำการตรวจสอบว่ามีการกดคีย์หรือไม่ ถ้ามีก็จะไปทำที่ข้อ 14.
3. ทำการอ่านค่าความเข้มแสงจากหัวอ่านมาให้ตัวแปร i
4. ถ้าค่า i มีค่าน้อยกว่า 0x90 ให้ไปทำงานที่ข้อ 10.
5. กำหนดให้ค่า θi = 45 องศา
6. ถ้าค่า i มีค่ามากกว่า 0x90 ติดต่อกันเกิน 20 ครั้งให้เพิ่มค่า θi อีก 20 องศา และให้เพิ่มทุกๆ 20 ครั้ง
7. หาค่ามุมทิศทางหลัก($\theta m(n)$)ในปัจจุบันโดยกำหนดให้

   $\theta m(n) = \theta m(n-1) + \theta i$
8. คำนวณหาค่าความเร็วในแนวตั้งและแนวนอนโดยกำหนดให้

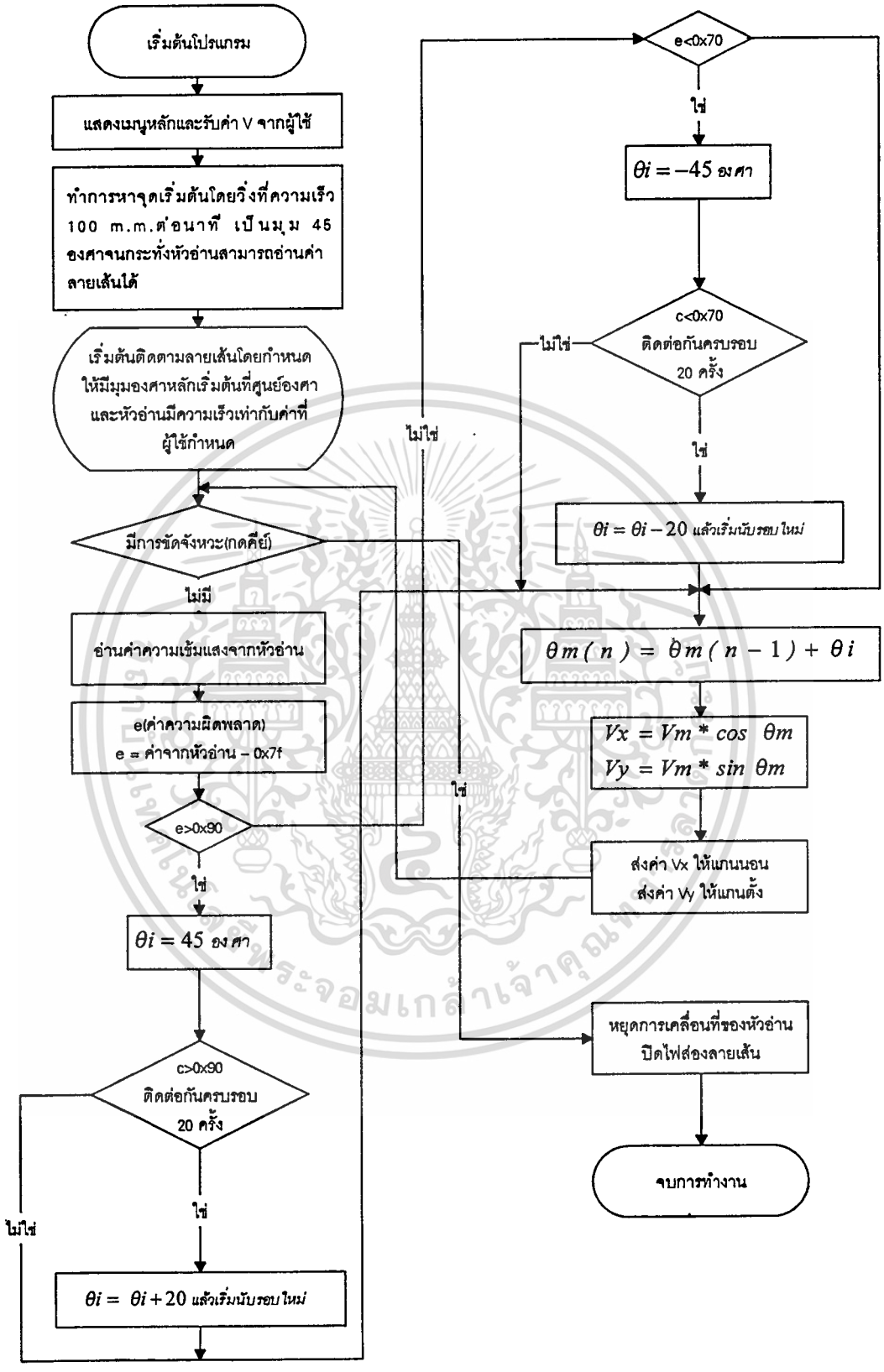
   $Vx = Vm * cos\ \theta m$ และ $Vy = Vm * sin\ \theta m$
9. ทำการส่งค่า Vx ให้แกนนอนและ Vy ให้แนวตั้ง และกลับไปทำข้อ 2.
10. ถ้าค่า i มีค่ามากกว่า 0x70 ให้ไปทำงานที่ข้อ 7.โดยกำหนดให้ θi = 0 องศา
11. กำหนดให้ค่า θi = – 45 องศา
12. ถ้าค่า i มีค่าน้อยกว่า 0x70 ติดต่อกันเกิน 20 ครั้ง ให้เพิ่มค่า θi อีก 20 องศา และให้เพิ่มทุกๆ 20 ครั้ง
13. กลับไปทำข้อที่ 7.
14. หยุดการเคลื่อนที่ของหัวอ่าน ปิดไฟส่องลายเส้น และออกจากโปรแกรม

จากขั้นตอนการทำงานข้างต้นสามารถเขียนโครงสร้างโปรแกรมติตตามลายเส้นได้ดังรูปที่ 4.3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กล่าวโดยสรุปในการเริ่มต้นหาลายเส้นจะเหมือนกับวิธีแรก เมื่อเข้าสู่โปรแกรมติดตามลายเส้นจะมีการตั้งค่าองศาทิศทางหลักเท่ากับศูนย์ และอ่านค่าความผิดพลาดจากหัวอ่านมาตรวจสอบ ถ้าค่าที่อ่านได้มีค่าน้อยกว่าค่ากำหนดต่ำสุดก็จะให้ค่ามุมสะท้อนมีค่าเท่ากับ 45 องศา แต่ถ้ามีค่ามากกว่าค่ากำหนดสูงสุดก็จะให้ค่ามุมสะท้อนมีค่าเท่ากับ – 45 องศา และนำค่ามุมสะท้อนที่ได้ไปรวมกับค่าองศาทิศทางหลักเพื่อคำนวณหาค่าความเร็วในแกนนอนและแกนตั้งต่อไป ในขณะที่หัวอ่านมีการวิ่งแบบสะท้อนไปมาบนลายเส้นนั้น ค่าองศาทิศทางหลักจะไม่มีการเปลี่ยนแปลง จนกระทั่งโปรแกรมตรวจพบว่าการสะท้อนครั้งสุดท้ายไม่สามารถทำให้หัวอ่านติดตามลายเส้นต่อไปได้ ในเวลาหรือจำนวนรอบการประมวลผลที่กำหนดก็จะทำการเปลี่ยนแปลงค่าองศาทิศทางหลักใหม่ โดยจะมีการเพิ่มครั้งละ ± 45 องศา ซึ่งค่าเครื่องหมายบวกลบที่ได้มีค่าขึ้นอยู่กับค่าความผิดพลาดเช่นเดียวกับการหาค่ามุมสะท้อน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.3 แสดงโครงสร้างโปรแกรมการติดตามลายเส้นแบบสะท้อนแสงร่วมกับการตัดสินใจ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3 การออกแบบโปรแกรมบันทึกข้อมูลที่ได้จากการติดตามลายเส้น

4.3.1 โปรแกรมบันทึกข้อมูลที่ได้จากการติดตามลายเส้นระบบควบคุมแบบตั้งค่าควบคุม PID

โปรแกรมนี้จะเริ่มต้นทำงานโดยย้ายหัวอ่านไปที่ตำแหน่งเริ่มต้นทำการเปิดไฟส่องลายเส้นแล้ววิ่งออกจากจุดเริ่มต้นเป็นมุม 45 องศาที่ความเร็ว 100 มิลลิเมตรต่อนาที จนกระทั่งหัวอ่านเริ่มตอบสนองต่อลายเส้นได้(ค่าความเข้มแสงมีค่าลดลงถึงจุดหนึ่ง) ต่อจากนั้นก็จะเข้าสู่การทำงานในระบบการติดตามลายเส้น

โปรแกรมติดตามลายเส้นเพื่อบันทึกข้อมูลจะมีขั้นตอนการทำงานดังต่อไปนี้

1. เมื่อเริ่มเข้าสู่โปรแกรมติดตามลายเส้นจะมีการตั้งค่าความเร็วตามที่ผู้ใช้กำหนดและให้หัวอ่านมีทิศทางการเคลื่อนที่เป็นมุมเท่ากับศูนย์องศา และตั้งค่าตัวแปรทุกตัวมีค่าเท่ากับศูนย์
2. ทำการตรวจสอบว่ามีการกดคีย์หรือไม่ ถ้ามีก็จะไปทำที่ข้อ 11.
3. ทำการอ่านค่าความเข้มแสงจากหัวอ่านมาลบกับค่ากึ่งกลาง 0x7f ก็จะได้ค่าความผิดพลาดค่าหนึ่งแทนค่าด้วยตัวแปร e
4. เข้าสู่โปรแกรมบันทึกลายเส้น
5. กำหนดให้ $\theta k = Kp * e$
6. กำหนดให้ $ei(n) = ei(n-1) + e$

   ถ้า $ei(n) > 1000$ กำหนดให้ $ei(n) = 1000$

   ถ้า $ei(n) < -1000$ กำหนดให้ $ei(n) = -1000$

   Vx กำหนดให้ $\theta i = Ki * ei(n)$
7. กำหนดให้ $ed(n) = e - ed(n-1)$

   กำหนดให้ $\theta d = Kd * ed(n)$
8. นำค่าที่ได้จากข้อ 4. 5. และ 6. มารวมกันจะได้ว่า

   $$\theta m(n) = \theta m(n-1) + \theta p + \theta i + \theta d$$
9. หาค่าความเร็วในแนวตั้งจะได้ว่า

   $$Vx = Vm * cos\ \theta m$$

   หาค่าความเร็วในแนวนอนจะได้ว่า

   $$Vy = Vm * sin\ \theta m$$
10. ทำการส่งค่า Vx ให้แก่แนวนอน

    ทำการส่งค่า Vy ให้แก่แนวตั้ง

    เริ่มทำที่ข้อ 2. ใหม่
11. ทำการปิดไฟส่องลายเส้น

    หยุดการเคลื่อนที่ของหัวอ่าน

    ออกจากโปรแกรม

สำหรับโปรแกรมบันทึกลายเส้นจะมีขั้นตอนการทำงานดังต่อไปนี้

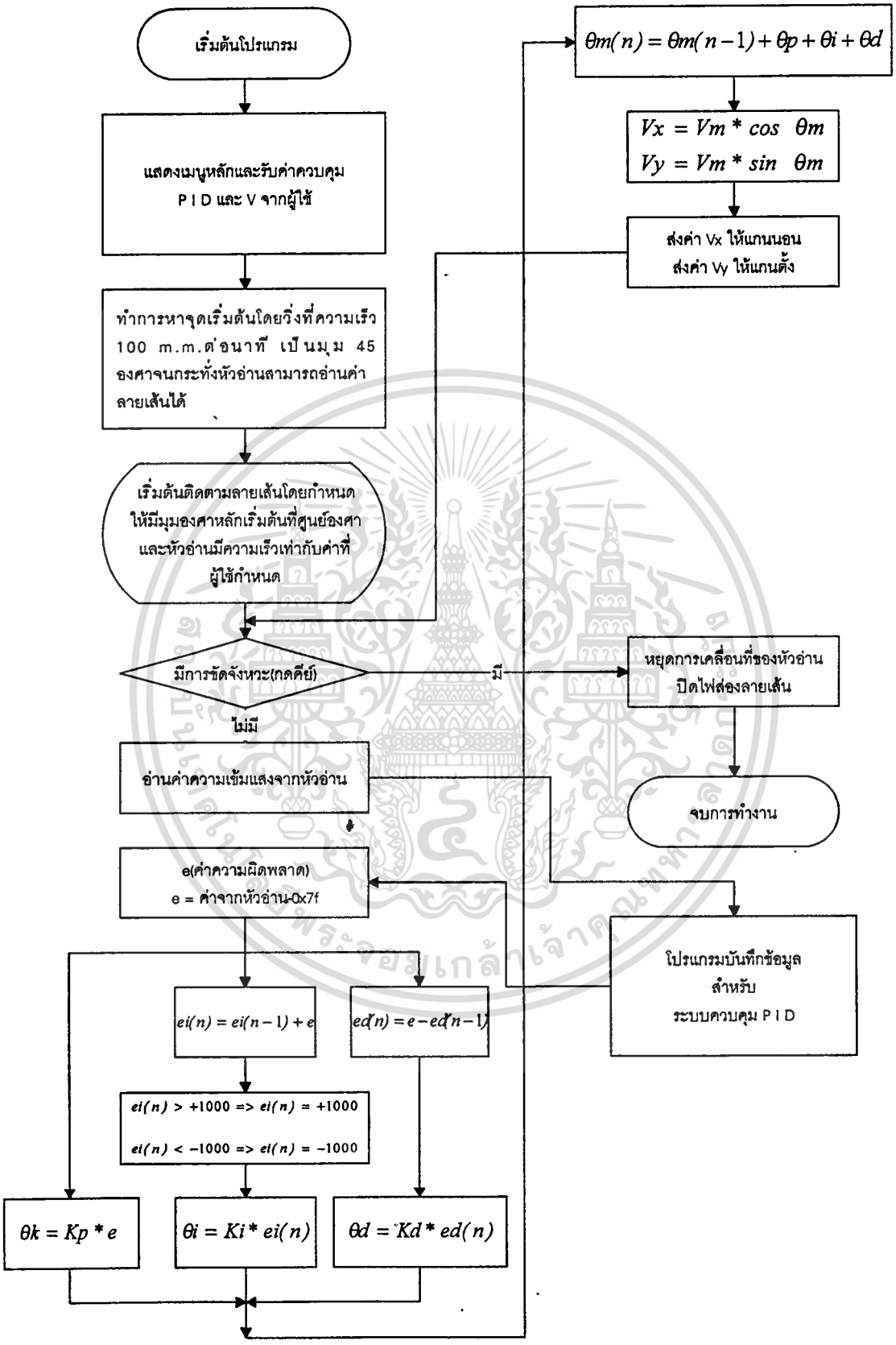
1. ถ้าสภาวะการบันทึกปิดให้ไปทำข้อที่ 6.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. ถ้าค่าที่ได้จากหัวอ่านไม่อยู่ในช่วงระหว่าง 0x73 ถึง 0x77 ให้ไปทำข้อที่ 6.
3. ถ้าหัวอ่านอยู่ห่างจากจุดบันทึกครั้งสุดท้ายเกินกว่า 300 สเต็ปย่อยให้ไปทำข้อที่ 6.
4. อ่านตำแหน่งหัวอ่านในแนวแกนตั้งและแกนนอน เก็บลงในแฟ้มข้อมูล
5. ถ้าแฟ้มข้อมูลมีไม่มีที่ว่างเหลือหรือหัวอ่านอยู่ห่างจุดที่เริ่มทำการบันทึกน้อยกว่า 500 สเต็ปย่อย จะถือว่าการบันทึกเสร็จสมบูรณ์แล้ว ให้ยุติการติดตามลายเส้น
6. ออกจากโปรแกรมกลับสู่โปรแกรมติดตามลายเส้น

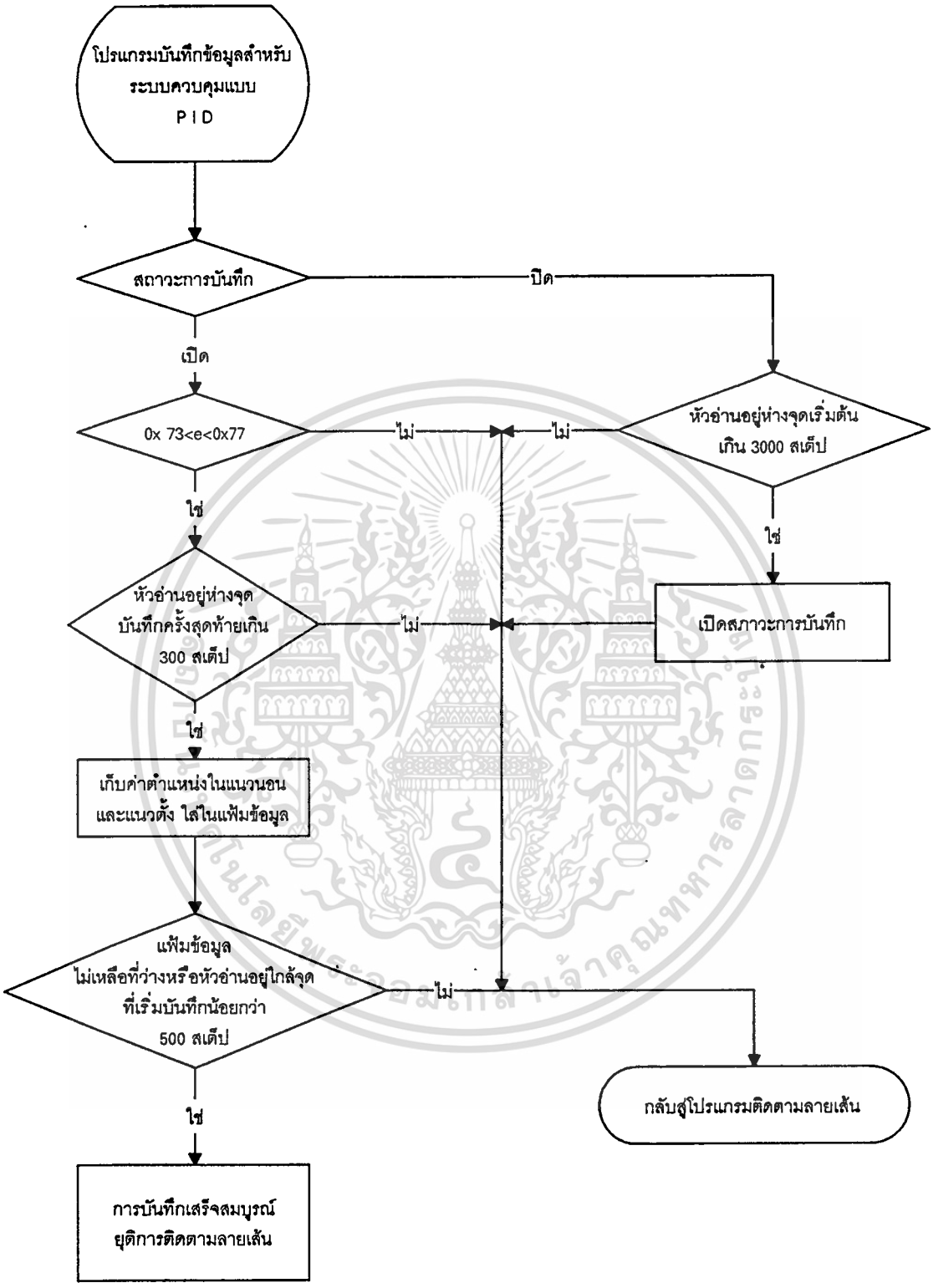
จากการออกแบบขั้นตอนการทำงานขั้นต้นสามารถนำมาเขียนโครงสร้างโปรแกรมสำหรับการติดตามเพื่อบันทึกลายเส้นได้ดังรูปที่ 4.4 และรูปที่ 4.5 ตามลำดับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.4 แสดงโครงสร้างของโปรแกรมติดตามลายเส้นเพื่อบันทึกระบบ P I D

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.5 แสดงโครงสร้างโปรแกรมบันทึกสำหรับระบบควบคุม P I D

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3.2 โปรแกรมบันทึกข้อมูลที่ได้จากการติดตามลายเส้นระบบควบคุมแบบกึ่งตัดสินใจร่วมกับการควบคุมแบบ P

โปรแกรมนี้จะเริ่มต้นทำงานโดยย้ายหัวอ่านไปที่ตำแหน่งเริ่มต้นทำการเปิดไฟส่องลายเส้นแล้ววิ่งออกจากจุดเริ่มต้นเป็นมุม 45 องศาที่ความเร็ว 100 มิลลิเมตรต่อนาที จนกระทั่งหัวอ่านเริ่มตอบสนองต่อลายเส้นได้(ค่าความเข้มแสงมีค่าลดลงถึงจุดหนึ่ง) ต่อจากนั้นก็จะเข้าสู่การทำงานในระบบการติดตามลายเส้น

โปรแกรมติดตามลายเส้นเพื่อบันทึกข้อมูลจะมีขั้นตอนการทำงานดังต่อไปนี้

1. เมื่อเริ่มเข้าสู่โปรแกรมติดตามลายเส้นจะมีการตั้งค่าความเร็วตามที่ผู้ใช้กำหนดและให้หัวอ่านมีทิศทางการเคลื่อนที่เป็นมุมเท่ากับศูนย์องศา และตั้งค่าตัวแปรทุกตัวมีค่าเท่ากับศูนย์
2. ทำการตรวจสอบว่ามีการกดคีย์หรือไม่ ถ้ามีก็จะไปทำที่ข้อ 15.
3. ทำการอ่านค่าความเข้มแสงจากหัวอ่านมาลบกับค่ากึ่งกลาง 0x7f ก็จะได้ค่าความผิดพลาดค่าหนึ่งแทนค่าด้วยตัวแปร e และให้ตัวแปร i เก็บค่าความเข้มแสงจากหัวอ่าน
4. เข้าสู่โปรแกรมบันทึกลายเส้น
5. ถ้าค่า i มีค่ามากกว่า 0x20 ให้ไปทำงานที่ข้อ 11.
6. ถ้าค่า i มีค่าน้อยกว่า 0x20 ติดต่อกันเกิน 2 ครั้ง กำหนดให้ $\theta i = 0.5$ องศา
   แต่ถ้าไม่เกิน กำหนดให้ θi = 0.2 องศา
7. ถ้าค่า i ยังมีค่าน้อยกว่า 0x20 ติดต่อกันเกิน 10 ครั้ง ให้เพิ่มค่า θi อีก 5.0 องศาทุกๆ 10 ครั้ง เพื่อให้หัวอ่านมีการเปลี่ยนทิศทางที่เร็วขึ้น
8. หาค่ามุมทิศทางหลัก($\theta m(n)$)ในปัจจุบันโดยกำหนดให้
   $\theta m(n) = \theta m(n-1) + 0.125 * e + \theta i$
9. คำนวณหาค่าความเร็วในแนวตั้งและแนวนอนโดยกำหนดให้
   $Vx = Vm * \cos\ \theta m$ และ $Vy = Vm * \sin\ \theta m$
10. ทำการส่งค่า Vx ให้แกนนอนและ Vy ให้แนวตั้ง และกลับไปทำข้อ 2.
11. ถ้าค่า i มีค่าน้อยกว่า 0xc0 ให้ไปทำงานที่ข้อ 8.
12. ถ้าค่า i มีค่ามากกว่า 0xc0 ติดต่อกันเกิน 2 ครั้ง กำหนดให้ θi = -0.5 องศา
    แต่ถ้าไม่เกินกำหนดให้ θi = -0.2 องศา
13. ถ้าค่า i ยังมีค่ามากกว่า 0xc0 ติดต่อกันเกิน 10 ครั้ง ให้เพิ่มค่า θi อีก 5.0 องศาทุกๆ 10 ครั้ง
14. กลับไปทำข้อที่ 8.
15. หยุดการเคลื่อนที่ของหัวอ่าน ปิดไฟส่องลายเส้น และออกจากโปรแกรม

สำหรับโปรแกรมบันทึกลายเส้นจะมีขั้นตอนการทำงานดังต่อไปนี้

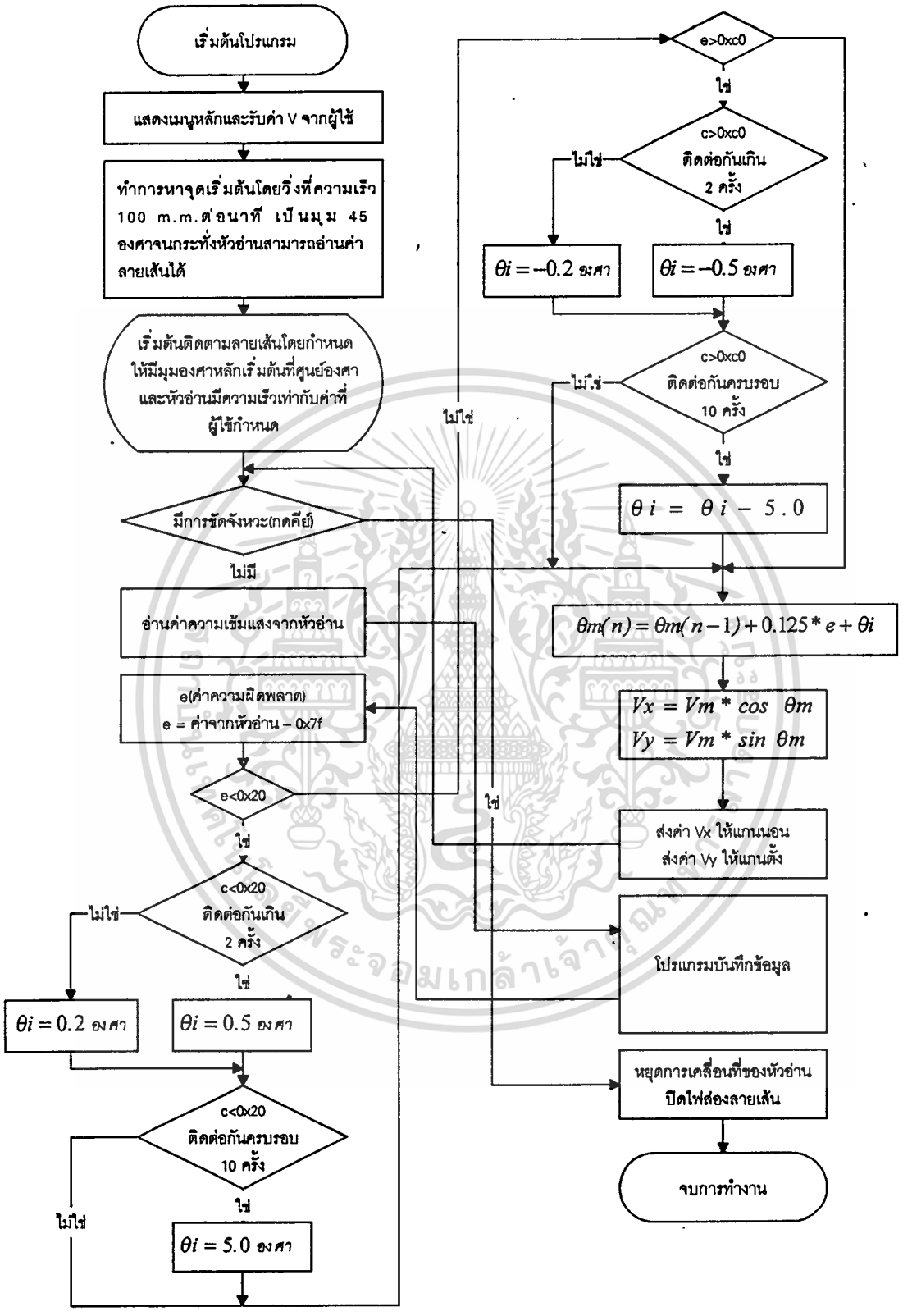
1. ถ้าสภาวะการบันทึกปิดให้ไปทำข้อที่ 6.
2. ถ้าค่าที่ได้จากหัวอ่านไม่อยู่ในช่วงระหว่าง 0x73 ถึง 0x77 ให้ไปทำข้อที่ 6.
3. ถ้าหัวอ่านอยู่ห่างจากจุดบันทึกครั้งสุดท้ายเกินกว่า 300 สเต็ปย่อยให้ไปทำข้อที่ 6.
4. อ่านตำแหน่งหัวอ่านในแนวแกนตั้งและแกนนอน เก็บลงในแฟ้มข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5. ถ้าแฟ้มข้อมูลมีไม่มีที่ว่างเหลือหรือหัวอ่านอยู่ห่างจุดที่เริ่มทำการบันทึกน้อยกว่า 500 สเต็ปย่อย จะถือว่าการบันทึกเสร็จสมบูรณ์แล้ว ให้ยุติการติดตามลายเส้น
6. ออกจากโปรแกรมกลับสู่โปรแกรมติดตามลายเส้น

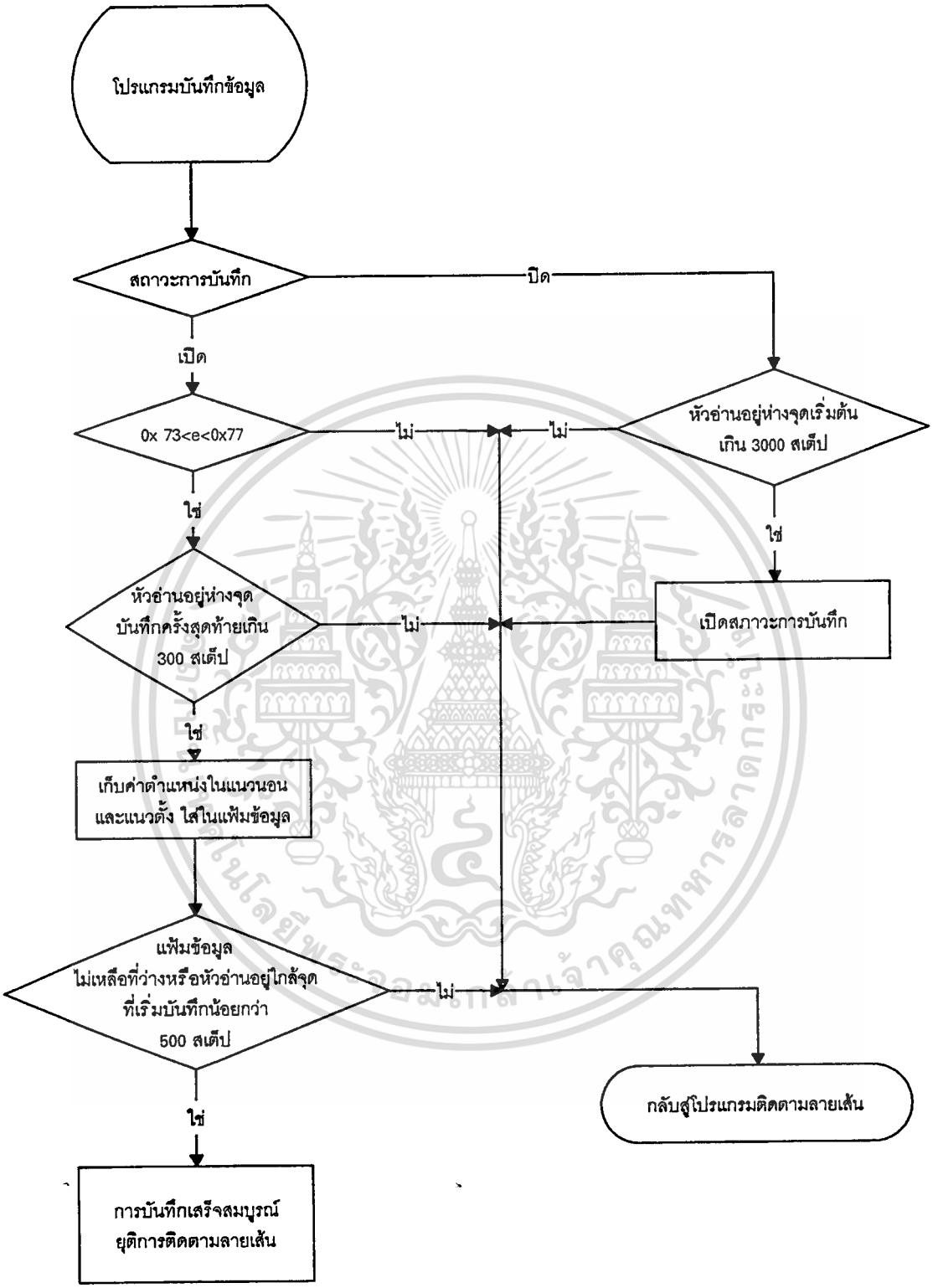
จากการออกแบบขั้นตอนการทำงานขั้นต้นสามารถนำมาเขียนโครงสร้างโปรแกรมสำหรับการติดตามเพื่อบันทึกลายเส้นได้ดังรูปที่ 4.6 และรูปที่ 4.7 ตามลำดับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.6 แสดงโครงสร้างของโปรแกรมติดตามลายเส้นเพื่อบันทึกแบบกึ่งตัดสินใจร่วมกับการควบคุมแบบ P

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.7 แสดงโครงสร้างของโปรแกรมบันทึกข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3.3 โปรแกรมบันทึกข้อมูลที่ได้จากการติดตามลายเส้นที่ใช้วิธีการวิ่งตามลายเส้นแบบการสะท้อนของแสงร่วมกับการตัดสินใจทางโปรแกรม

โปรแกรมนี้จะเริ่มต้นทำงานโดยย้ายหัวอ่านไปที่ตำแหน่งเริ่มต้นทำการเปิดไฟส่องลายเส้นแล้ววิ่งออกจากจุดเริ่มต้นเป็นมุม 45 องศาที่ความเร็ว 100 มิลลิเมตรต่อนาที จนกระทั่งหัวอ่านเริ่มตอบสนองต่อลายเส้นได้(ค่าความเข้มแสงมีค่าลดลงถึงจุดหนึ่ง) ต่อจากนั้นก็จะเข้าสู่การทำงานในระบบการติดตามลายเส้น

โปรแกรมติดตามลายเส้นจะมีขั้นตอนโดยละเอียดดังต่อไปนี้

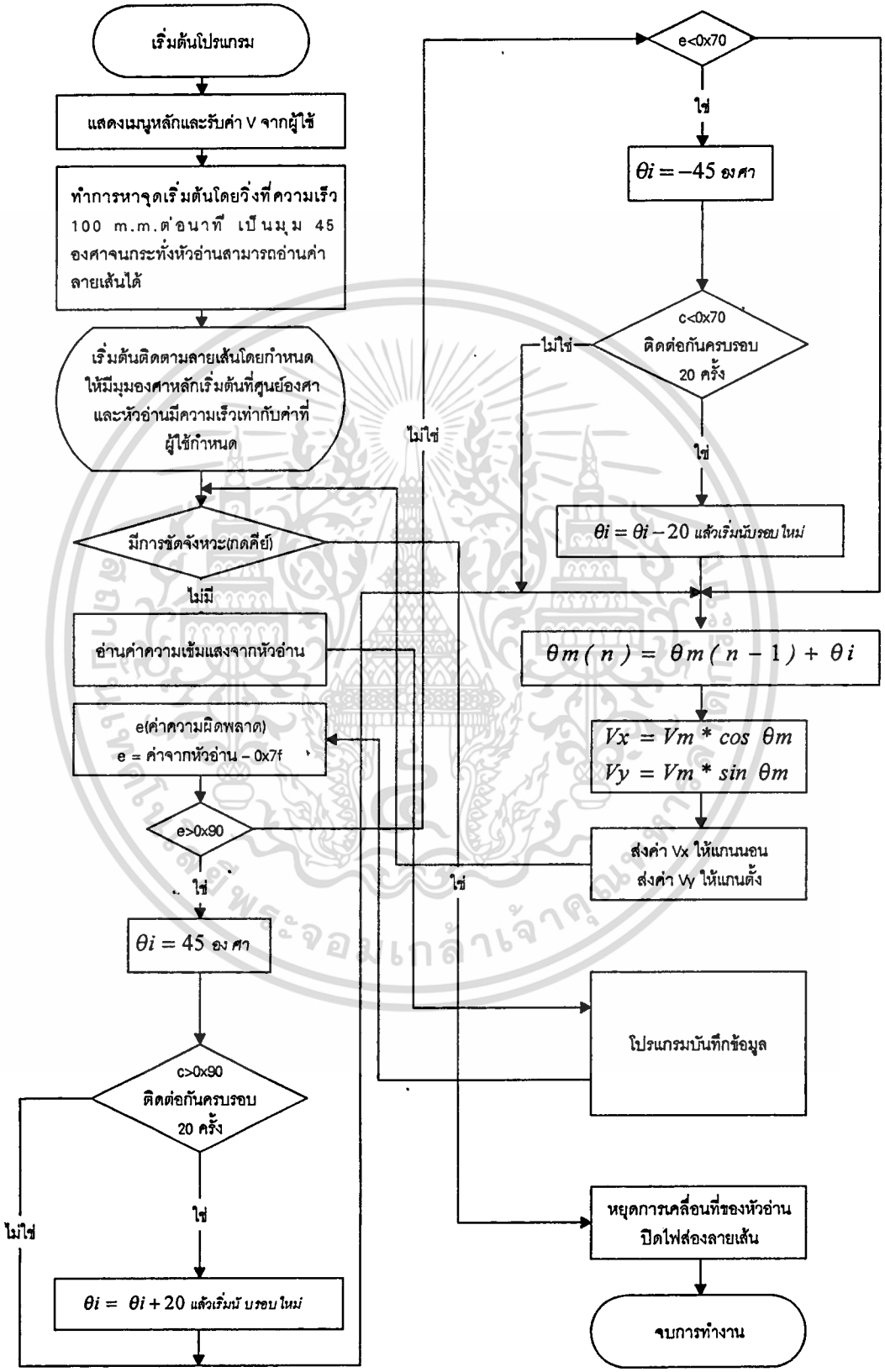
1. เมื่อเริ่มเข้าสู่โปรแกรมติดตามลายเส้นจะมีการตั้งค่าความเร็วตามที่ผู้ใช้กำหนดและให้หัวอ่านมีทิศทางการเคลื่อนที่เป็นมุมเท่ากับศูนย์องศา และตั้งค่าตัวแปรทุกตัวมีค่าเท่ากับศูนย์
2. ทำการตรวจสอบว่ามีการกดคีย์หรือไม่ ถ้ามีก็จะไปทำที่ข้อ 15.
3. ทำการอ่านค่าความเข้มแสงจากหัวอ่านมาให้ตัวแปร i
4. เข้าสู่โปรแกรมบันทึกข้อมูล
5. ถ้าค่า i มีค่าน้อยกว่า 0x90 ให้ไปทำงานที่ข้อ 11.
6. กำหนดให้ค่า θi = 45 องศา
7. ถ้าค่า i มีค่ามากกว่า 0x90 ติดต่อกันเกิน 20 ครั้งให้เพิ่มค่า θi อีก 20 องศา และให้เพิ่มทุกๆ 20 ครั้ง
8. หาค่ามุมทิศทางหลัก($\theta m(n)$)ในปัจจุบันโดยกำหนดให้
   $\theta m(n) = \theta m(n-1) + \theta i$
9. คำนวณหาค่าความเร็วในแนวตั้งและแนวนอนโดยกำหนดให้
   $Vx = Vm * cos\ \theta m$ และ $Vy = Vm * sin\ \theta m$
10. ทำการส่งค่า Vx ให้แกนนอนและ Vy ให้แนวตั้ง และกลับไปทำข้อ 2.
11. ถ้าค่า i มีค่ามากกว่า 0x70 ให้ไปทำงานที่ข้อ 7.โดยกำหนดให้ θi = 0 องศา
12. กำหนดให้ค่า θi = – 45 องศา
13. ถ้าค่า i มีค่าน้อยกว่า 0x70 ติดต่อกันเกิน 20 ครั้ง ให้เพิ่มค่า θi อีก 20 องศา และให้เพิ่มทุกๆ 20 ครั้ง
14. กลับไปทำข้อที่ 8.
15. หยุดการเคลื่อนที่ของหัวอ่าน ปิดไฟส่องลายเส้น และออกจากโปรแกรม

สำหรับโปรแกรมบันทึกลายเส้นจะมีขั้นตอนการทำงานดังต่อไปนี้

1. ถ้าสภาวะการบันทึกปิดให้ไปทำข้อที่ 6.
2. ถ้าค่าที่ได้จากหัวอ่านไม่อยู่ในช่วงระหว่าง 0x73 ถึง 0x77 ให้ไปทำข้อที่ 6.
3. ถ้าหัวอ่านอยู่ห่างจากจุดบันทึกครั้งสุดท้ายเกินกว่า 300 สเต็ปย่อยให้ไปทำข้อที่ 6.
4. อ่านตำแหน่งหัวอ่านในแนวแกนตั้งและแกนนอน เก็บลงในแฟ้มข้อมูล
5. ถ้าแฟ้มข้อมูลมีไม่มีที่ว่างเหลือหรือหัวอ่านอยู่ห่างจุดที่เริ่มทำการบันทึกน้อยกว่า 500 สเต็ปย่อย จะถือว่าการบันทึกเสร็จสมบูรณ์แล้ว ให้ยุติการติดตามลายเส้น
6. ออกจากโปรแกรมกลับสู่โปรแกรมติดตามลายเส้น

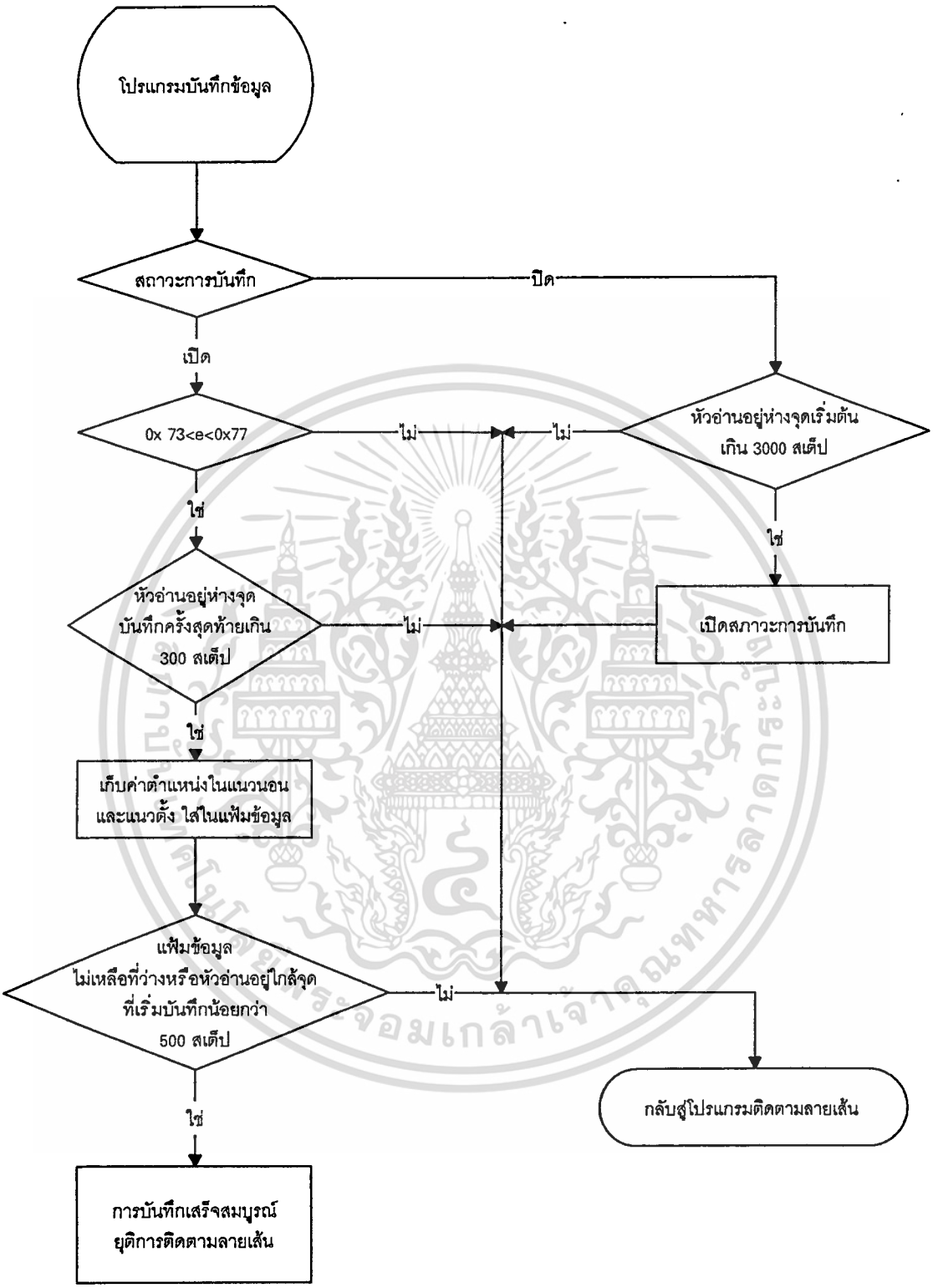
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากการออกแบบขั้นตอนการทำงานขั้นต้นสามารถนำมาเขียนโครงสร้างโปรแกรมสำหรับการติดตามเพื่อบันทึกลายเส้นได้ดังรูปที่ 4.8 และรูปที่ 4.9 ตามลำดับ



รูปที่ 4.8 แสดงโครงสร้างโปรแกรมบันทึกลายเส้นแบบสะท้อนแสงร่วมกับการตัดสินใจทางโปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.9 แสดงโครงสร้างโปรแกรมบันทึกข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.4 การออกแบบโปรแกรมย่อยทำหน้าที่อ่านและเก็บข้อมูลจากจานเก็บข้อมูล(HARD DISK)

ในการเก็บตำแหน่งของหัวอ่านในแกนแนวตั้งและแกนแนวนอนจะใช้ตัวแปรเลขฐานสอง ขนาด 32 บิต เก็บค่าตำแหน่งในแต่ละแกน ซึ่งได้กำหนดให้เก็บตำแหน่งได้สูงสุด 5000 ตำแหน่ง โดยกำหนดให้ตัวแปรมีชื่อว่า X[...] และ Y[...] ตามลำดับ ดังนั้นตำแหน่งต่ำสุดก็คือ X[0] ,Y[0] และตำแหน่งสูงสุดก็คือ X[5000] ,Y[5000] ตามลำดับ

ดังนั้นในการเขียนโปรแกรมสำหรับการอ่านและเก็บข้อมูลสามารถเขียนได้ดังต่อไปนี้

4.4.1 โปรแกรมอ่านข้อมูลจากจานเก็บข้อมูล

โครงสร้างของโปรแกรมสามารถแสดงได้ดังรูปที่ 4.10

โปรแกรมอ่านข้อมูลจะเริ่มรับชื่อข้อมูลจากผู้ใช้งานแล้วทำการตรวจสอบกับชื่อข้อมูลในจานเก็บว่ามีชื่อที่ต้องการหรือไม่ ถ้าไม่มีก็จะหยุดการทำงาน ต่อจากนั้นก็จะทำการลบข้อมูลเก่าออกจากหน่วยความจำ และเริ่มอ่านข้อมูลจากจานเก็บลงสู่หน่วยความจำจนกระทั่งสิ้นสุดข้อมูล ซึ่งหมายถึงข้อมูลชุดสุดท้ายมีค่าเท่ากับศูนย์ หรือจำนวนข้อมูลมีค่าเท่ากับค่าสูงสุดแล้ว หลังจากนั้นก็จะออกจากโปรแกรม

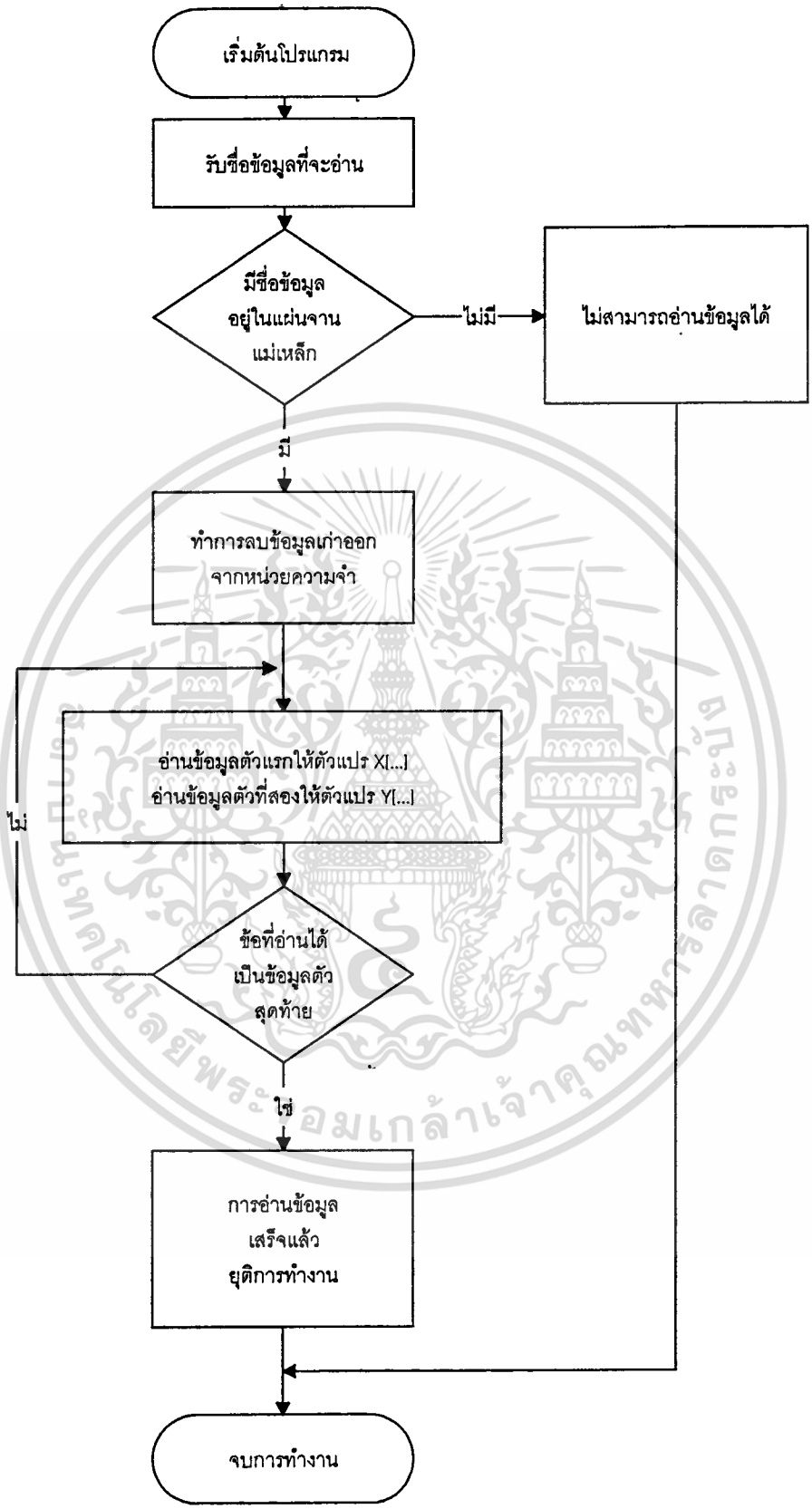
4.4.2 โปรแกรมเขียนข้อมูลลงจานเก็บข้อมูล

โครงสร้างของโปรแกรมสามารถแสดงได้ดังรูปที่ 4.11

โปรแกรมเขียนข้อมูลจะรับชื่อข้อมูลที่ต้องการจัดเก็บจากผู้ใช้ หลังจากนั้นก็จะตรวจสอบว่ามีข้อมูลสำหรับการจัดเก็บหรือไม่ ถ้าไม่มีก็จะออกจากโปรแกรม ต่อจากนั้นก็จะเริ่มเก็บค่า X[...] ก่อนและตามด้วยค่า y[...] เป็นการเก็บค่า X[...] และ Y[...] สลับกันไปเรื่อยจนกระทั่งหมดข้อมูล หลังจากนั้นก็จะสิ้นสุดการทำงานเก็บข้อมูล

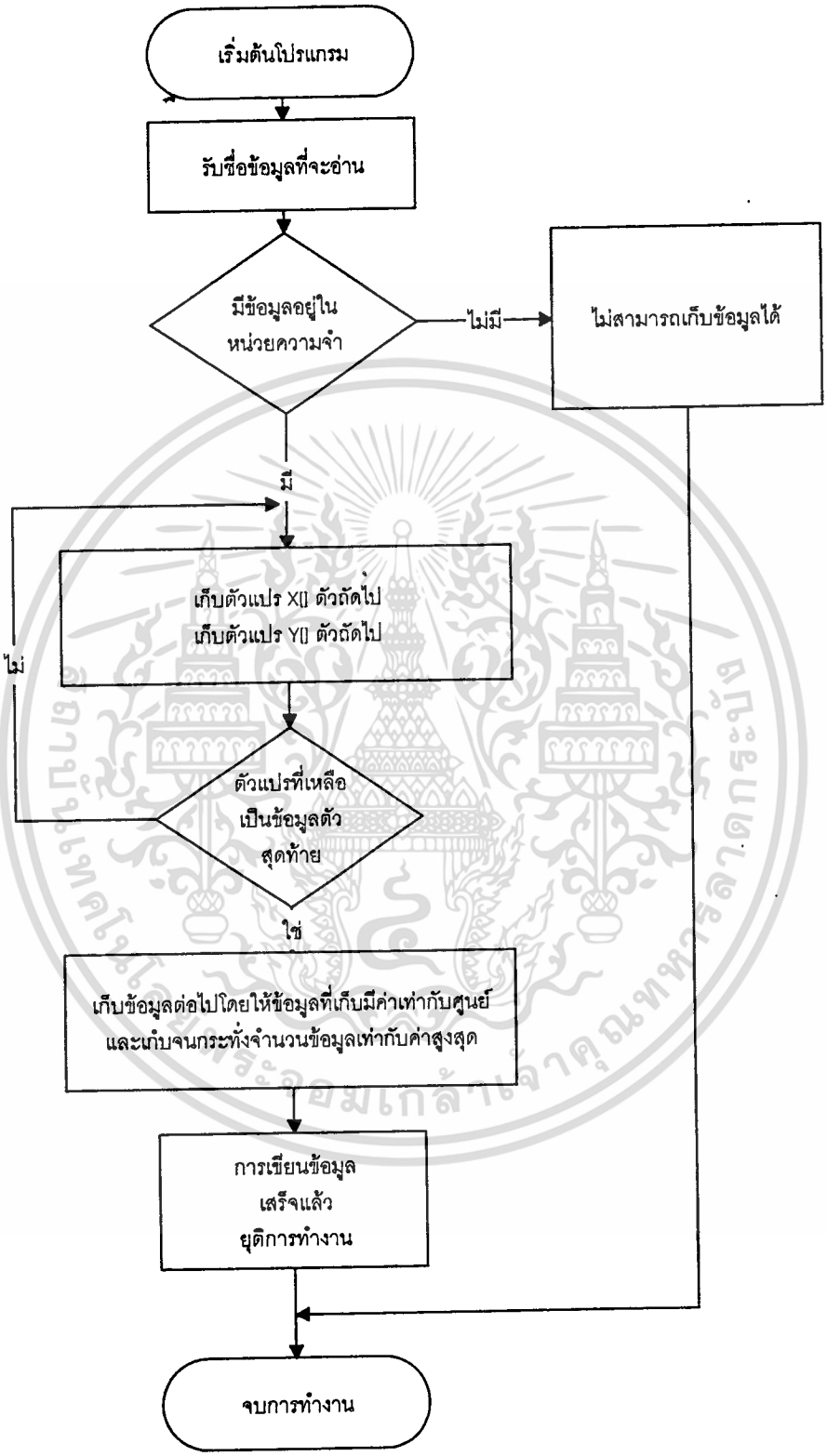
จุดสังเกต โปรแกรมนี้จะไม่มีการตรวจสอบว่าชื่อมีซ้ำอยู่ในจานเก็บดังนั้นข้อมูลในจานเก็บจะถูกเขียนทับไป จึงต้องระวังในเรื่องนี้ด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 4.10 แสดงโครงสร้างโปรแกรมอ่านข้อมูล



รูปที่ 4.11 แสดงโครงสร้างโปรแกรมเขียนข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.5 การออกแบบโปรแกรมหลักเพื่อรับคำสั่งจากผู้ใช้งานมาควบคุมโปรแกรมย่อยส่วนต่างๆ

ในการออกแบบระบบรวมกันเป็นโปรแกรมขนาดใหญ่จะต้องมีการกำหนดสิ่งแวดล้อมที่สำคัญของระบบให้เรียบร้อย เพื่อความเป็นระเบียบในการใช้ข้อมูลร่วมกัน ระหว่างโปรแกรมย่อยส่วนต่างๆ ดังนั้นจึงได้กำหนดสิ่งแวดล้อมของระบบดังต่อไปนี้

4.5.1 การกำหนดขอบเขตเพื่อกำหนดเงื่อนไขและค่าเริ่มต้นสำหรับระบบ

จากการพิจารณาระบบขับเคลื่อนสรุปได้ว่า

1. หัวอ่านสามารถเคลื่อนที่ได้ในช่วงมิติ 0.750 * 0.750 เมตร^2
2. หัวอ่านสามารถเคลื่อนที่ได้ด้วยความเร็วสูงสุดไม่เกิน 0.50 เมตรต่อวินาที

จากการพิจารณาระบบอิเลคทรอนิกส์สรุปได้ว่า

1. หัวอ่านมีอัตราการสุ่มข้อมูลความเข้มแสงประมาณ 3000 ครั้งต่อวินาที
2. การผลิตความถี่เพื่อกำหนดความเร็วของหัวอ่านสามารถทำได้ตั้งแต่ 0 ถึง 1.7895 เมกะเฮิตร์ แสดงสามารถทำความเร็วในทางอุดมคติได้เท่ากับ 0 ถึง 13.176 เมตรต่อวินาที โดยปรับได้ละเอียดขั้นละ 1 มิลลิเมตรต่อวินาที

จากการพิจารณาการเขียนโปรแกรมสรุปได้ว่า

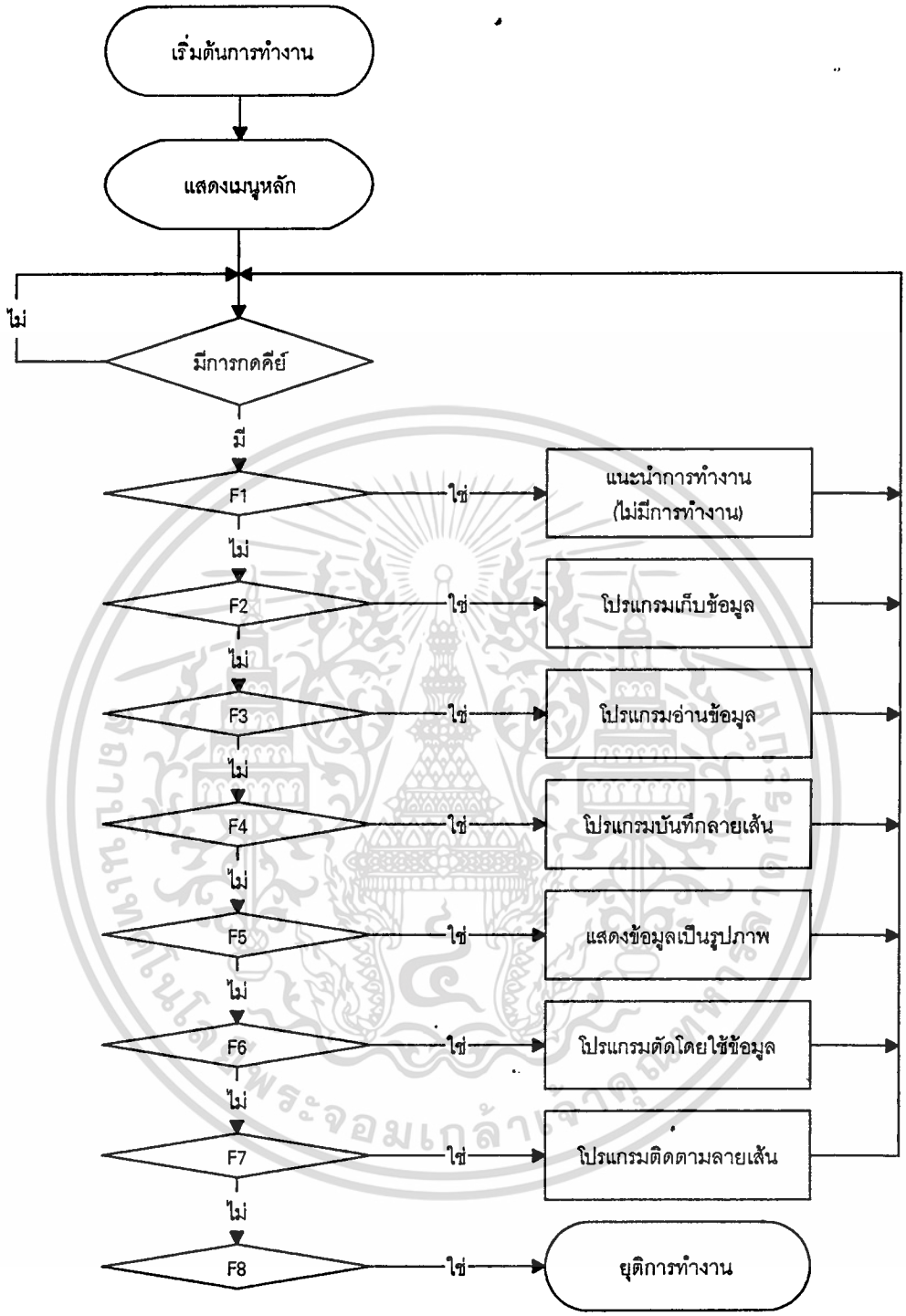
1. ความเร็วรอบการทำงานของระบบขึ้นอยู่กับชนิดของหน่วยประมวลผล ซึ่งหน่วยประมวลผลที่ใช้อยู่นี้ได้ใช้ CPU 80486 SX-33 ซึ่งไม่มีตัวประมวลผลทางคณิตศาสตร์ร่วมด้วย ทำให้ได้ความเร็วรอบการทำงานอย่างน้อยประมาณ 150 รอบต่อวินาที ค่าที่ได้นี้ได้จากการประมาณเท่านั้น โดยการทดสอบด้านการคำนวณของโปรแกรมได้ว่าสามารถหาค่า sin θ ได้ในอัตรา 8000 ครั้งต่อวินาทีและประมาณว่าในการทำงานจะมีการหาค่าทางตรีโกณประมาณ 30 ครั้งและโปรแกรมเขียนด้วยภาษาสูงทำให้มีการทำงานช้าลง เมื่อคิดระบบโดยรวมจึงคาดว่าได้ความเร็วรอบการทำงานประมาณ 150 ครั้งต่อวินาที ซึ่งก็เป็นความเร็วที่ใช้ได้ดีทีเดียว

2. ขนาดของหน่วยความจำที่ใช้เก็บข้อมูลมีขนาดรวมกันได้ไม่เกิน 65536 ไบต์(BYTE) หรือ 16384 ดับเบิลเวิดส์ (DOUBLE WORD)ซึ่งเป็นขนาดข้อมูล 32 บิต และเนื่องจากจะต้องเก็บค่าในแกนนอนและแกนตั้ง อีกทั้งจะต้องแบ่งปันให้แก่ตัวแปรต่างๆ ด้วย จึงได้ว่าสามารถเก็บข้อมูลขนาด 32 บิตได้แกนละ 5000 ค่าเท่านั้น เนื่องโปรแกรมมีการบันทึกค่าช่วงละประมาณ 100 สเต็ป ดังนั้นจึงได้ว่าสามารถบันทึกลายเส้นได้ยาวขนาด 1.80 เมตร

4.5.2 การออกแบบโครงสร้างหรือขั้นตอนการทำงานของโปรแกรมหลัก

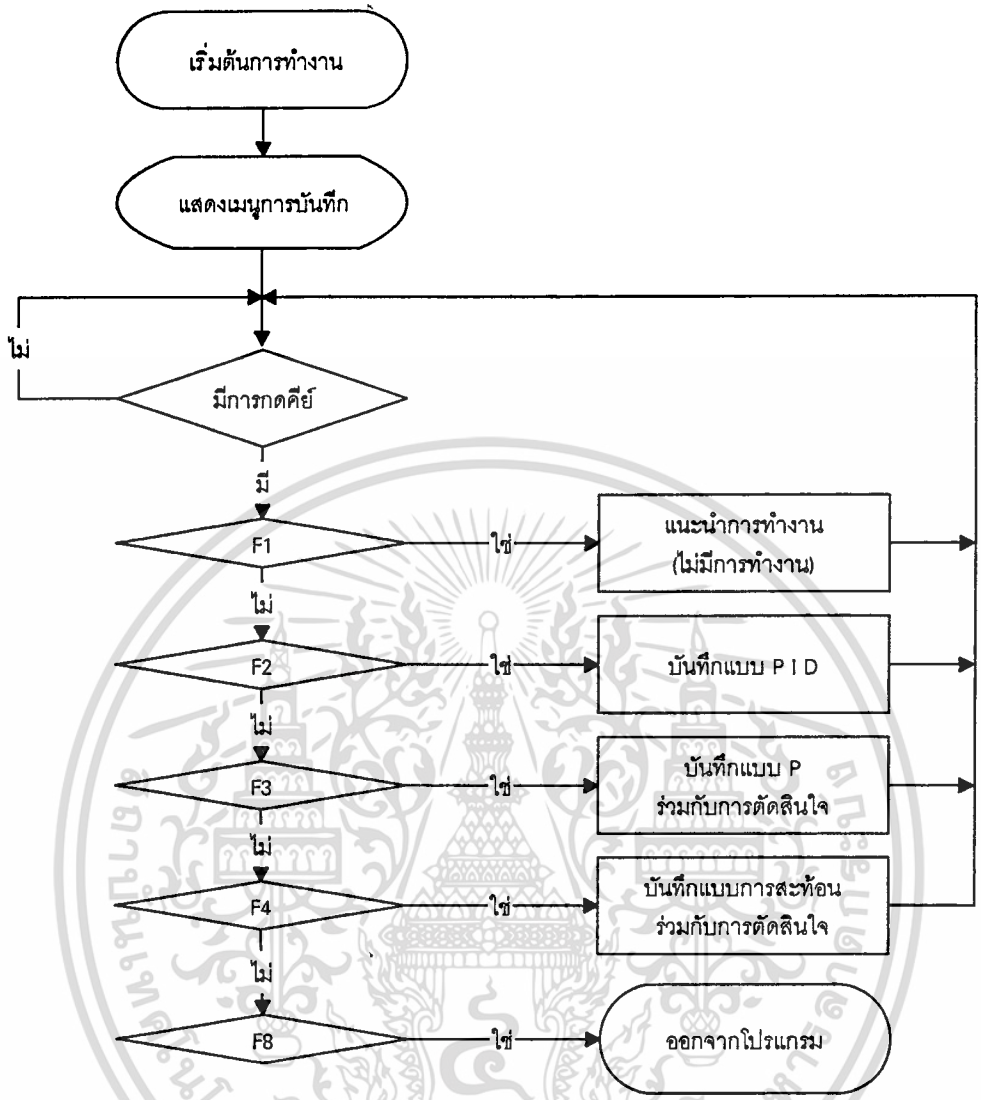
ขั้นตอนการทำงานของโปรแกรมหลักได้ออกแบบไว้เป็นโครงสร้างโปรแกรมดังรูปต่อไปนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.12 แสดงโครงสร้างโปรแกรมหลักส่วนที่ 1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

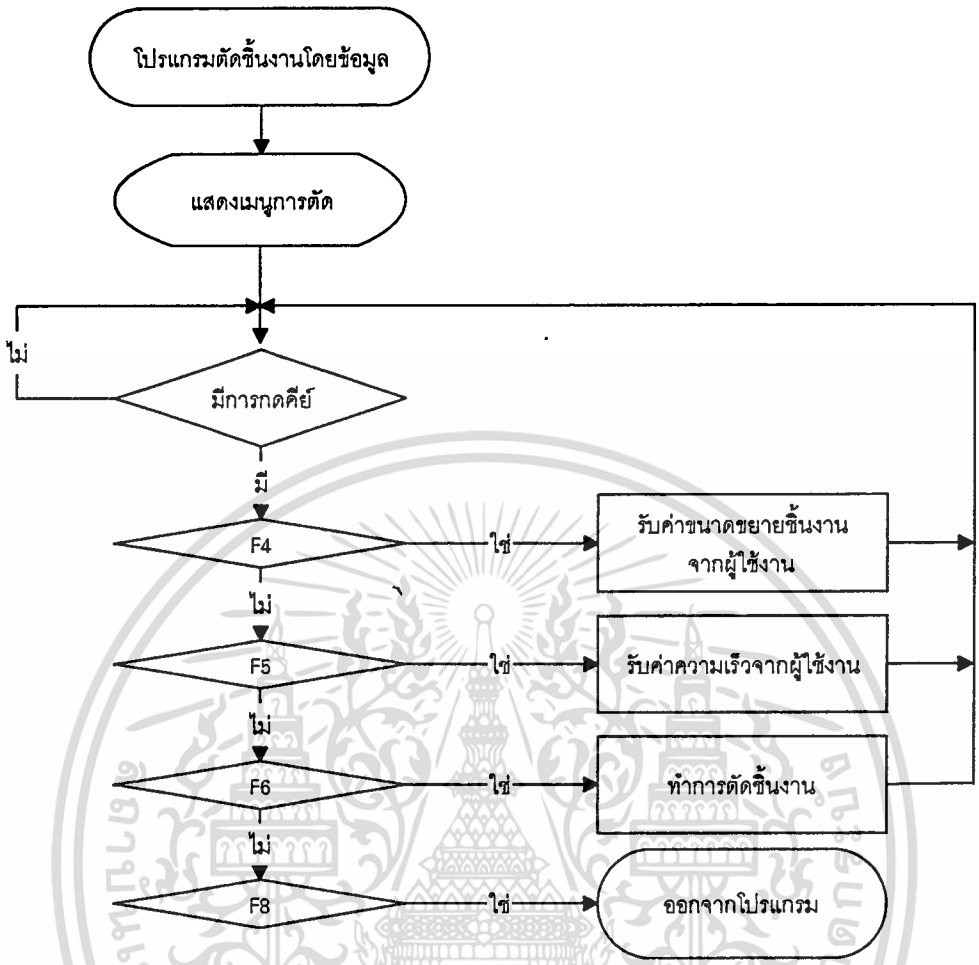


รูปที่ 4.13 แสดงโครงสร้างโปรแกรมหลักส่วนที่ 2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 4.14 แสดงโครงสร้างโปรแกรมหลักส่วนที่ 3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.15 แสดงโครงสร้างโปรแกรมหลักส่วนที่ 4

ในโครงสร้างโปรแกรมหลักจะมีบางโปรแกรมที่ไม่มีรายละเอียดเนื่องเห็นว่าได้มีการกล่าวถึงมาก่อนหน้าแล้วและบางโปรแกรมควรจะพิจารณาจากตัวโปรแกรมที่เขียนใช้งานโดยตรงจะดูเข้าใจได้ดีกว่า

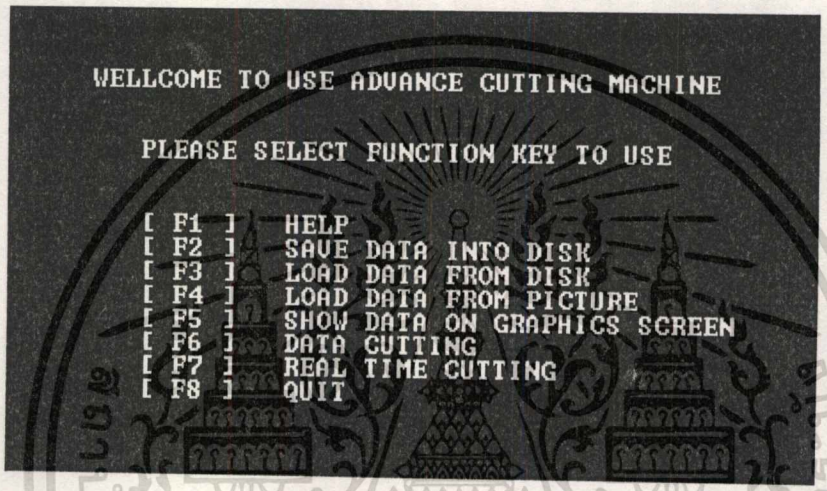
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5 วิธีการใช้งานโปรแกรม

หลังจากที่ได้เขียนโปรแกรมเสร็จสมบูรณ์แล้ว ก็ได้ทำการทดลองใช้โปรแกรมในส่วนต่างๆ ซึ่งสามารถแสดงวิธีใช้ได้ดังต่อไปนี้

5.1 แสดงวิธีการใช้งานโปรแกรมหลัก

หลังจากเรียกโปรแกรมให้ทำงานจะมีการแสดงรายการดังรูปที่ 5.1



รูปที่ 5.1 แสดงรายการของโปรแกรมหลัก

จากรายการจะเห็นคำสั่งต่างๆ ซึ่งมีหน้าที่ดังต่อไปนี้

[F1] เป็นคำสั่งขอข้อแนะนำการใช้โปรแกรม ซึ่งยังไม่มีการทำงานใดๆ เนื่องจากเห็นว่ายังไม่จำเป็น

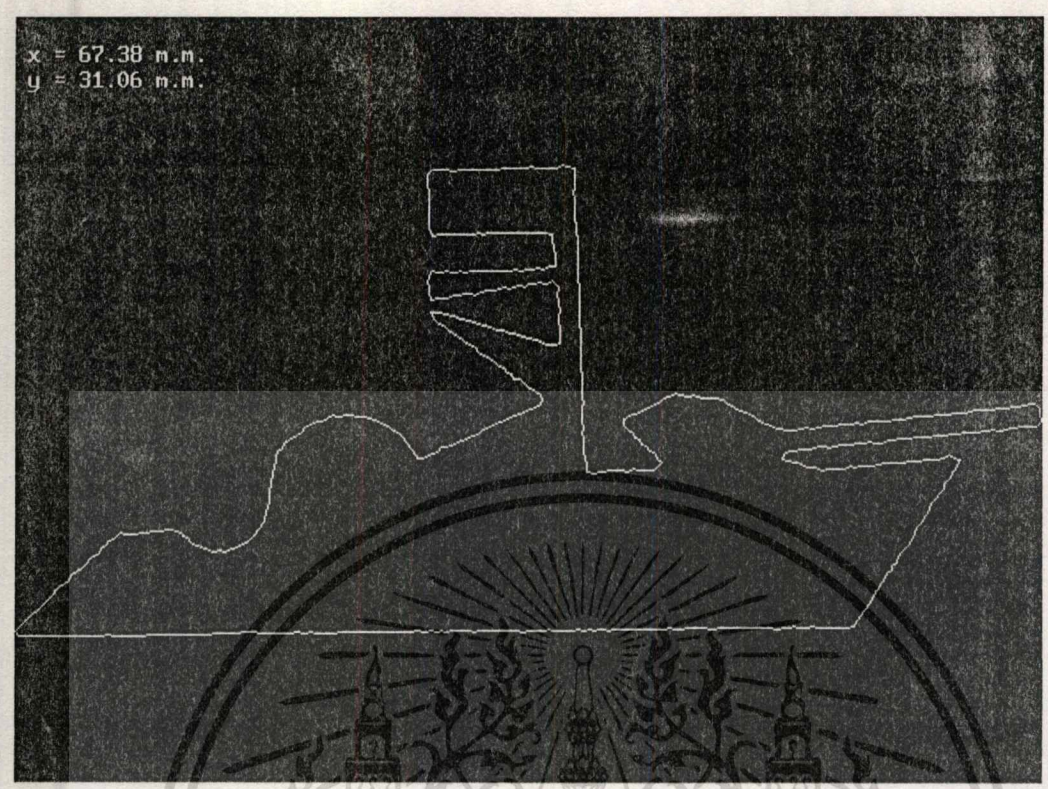
[F2] เป็นคำสั่งเก็บข้อมูลที่มีอยู่ลงบนจานเก็บข้อมูล เมื่อเรียกใช้งานจะมีการให้ผู้ใช้ตั้งชื่อแฟ้มข้อมูล ถ้าไม่มีข้อมูลสำหรับการจัดเก็บจะมีเสียงเตือนด้วย

[F3] เป็นคำสั่งอ่านจากจานเก็บข้อมูล เมื่อเรียกใช้งานจะมีการถามชื่อแฟ้มข้อมูลที่ต้องการอ่าน ถ้าหาข้อมูลไม่พบจะมีเสียงเตือน แต่ในการอ่านจะไม่มีการตรวจสอบว่าเป็นข้อมูลของระบบหรือไม่

[F4] เป็นคำสั่งให้ทำการบันทึกข้อมูลจากลายเส้น เมื่อเรียกใช้งานจะมีการทำงานตามหัวข้อที่ 5.2

[F5] เป็นคำสั่งให้แสดงข้อมูลที่มีอยู่บนจอภาพเป็นรูปวัตถุ โดยจะมีการปรับขนาดของรูปให้มีขนาดเต็มจอทุกกรณี ดังตัวอย่างในรูปที่ 5.2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5.2 แสดงตัวอย่างข้อมูลที่ปรากฏบนจอภาพ

[F6] เป็นคำสั่งให้ทำการตัดชิ้นงานโดยใช้ข้อมูลที่มีอยู่ เมื่อเรียกใช้งานจะมีการทำงานตามหัวข้อที่ 5.3

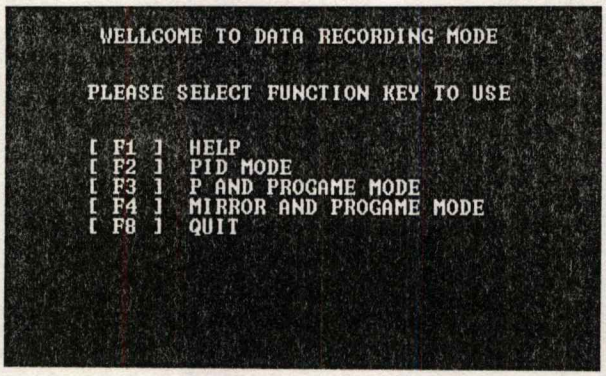
[F7] เป็นคำสั่งให้ทำการติดตามลายเส้น โดยไม่มีการบันทึกข้อมูลเก็บไว้ จะมีการกำหนดขั้นตอนการทำงานและตัวแปรต่างเช่นเดียวกับคำสั่ง [F4] แต่จะมีการทำงานติดตามลายเส้นไปเรื่อยๆ จนกระทั่งมีการกดคีย์ใดคีย์หนึ่งจากผู้ใช้งาน และจะไม่มีการตั้งค่าตัวแปรสำหรับการบันทึก

[F8] เป็นคำสั่งยกเลิกการทำงานทั้งหมด ในกรณีที่มีข้อมูลอยู่ในหน่วยความจำที่ยังไม่ได้จัดเก็บ ก็จะถูกลบโดยไม่มีการเตือนเช่นกัน ดังนั้นต้องระวังในจุดนี้ด้วย

5.2 การใช้งานระบบการบันทึกลายเส้น

เมื่อเรียกใช้งานจะแสดงรายการดังรูปที่ 5.3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5.3 แสดงรายการระบบบันทึก

จากรายการระบบบันทึกจะเห็นว่ามีวิธีการบันทึกให้เลือกหลายแบบตามความประสงค์ โดยสรุปได้ว่า

[F1] เป็นคำสั่งขอความช่วยเหลือ แต่ยังไม่สามารถได้เพราะไม่มีข้อมูลเนื่องจากเห็นว่ายังไม่จำเป็นในตอนนี้

[F2] เป็นการเลือกบันทึกในระบบ P I D

[F3] เป็นการเลือกบันทึกในระบบ P ร่วมกับการตัดสินใจ

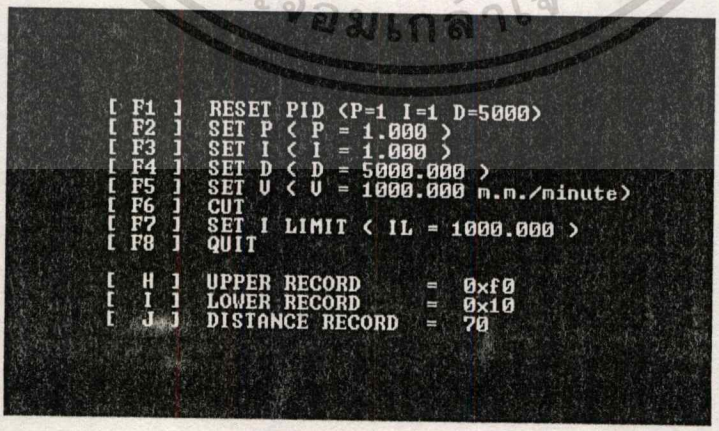
[F4] เป็นการเลือกบันทึกในระบบการสะท้อนของแสงร่วมกับการตัดสินใจ

[F5] เป็นคำสั่งยกเลิกการทำงาน

ซึ่งในรายละเอียดของการใช้งานในแต่ละวิธีมีดังต่อไปนี้

5.2.1 การใช้งานการบันทึกแบบ P I D

เมื่อเรียกใช้งานจะแสดงรายการดังรูปที่ 5.4



รูปที่ 5.4 แสดงรายการบันทึกแบบ P I D

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สามารถอธิบายหน้าที่ในแต่ละรายการได้ว่า

[F1] เป็นคำสั่งตั้งค่าตัวแปรต่างๆ ให้มีค่าเท่ากับค่าตอนเริ่มระบบ

[F2] เป็นคำสั่งตั้งค่าควบคุม P

[F3] เป็นคำสั่งตั้งค่าควบคุม I

[F4] เป็นคำสั่งตั้งค่าควบคุม D

[F5] เป็นคำสั่งตั้งค่าความเร็ว V

[F6] เป็นคำสั่งให้ทำการติดตามลายเส้นเพื่อทำการบันทึกข้อมูล และเมื่อหัวอ่านมีการเคลื่อนที่ไปตามลายเส้นจนครบรอบก็จะหยุดการทำงานโดยอัตโนมัติ

[F7] เป็นคำสั่งกำหนดค่าสูงสุดและต่ำสุดของการอินทีเกรต มีผลทำให้ค่าที่ทำการสะสมจะมีค่าไม่เกินหรือน้อยกว่าค่าดังกล่าว

[F8] เป็นคำสั่งยุติการทำงาน

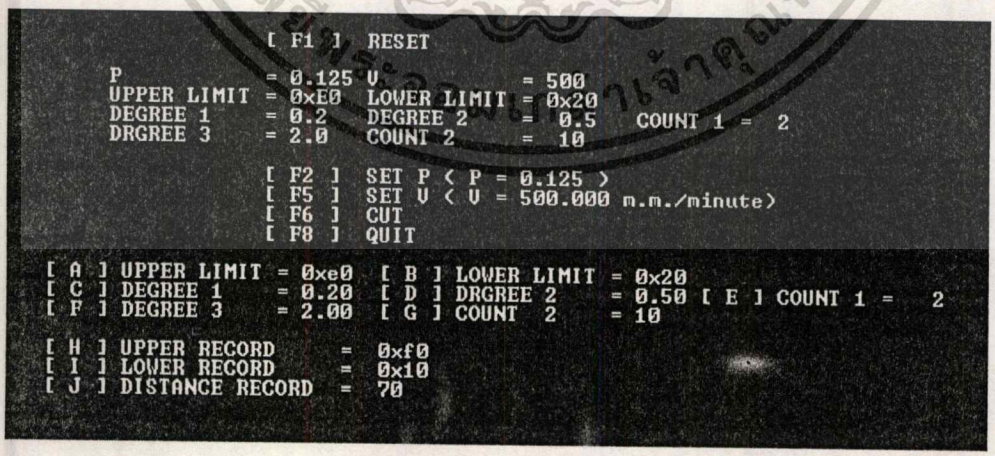
[H] เป็นคำสั่งกำหนดค่าสูงสุดของการบันทึก ถ้าค่าที่อ่านได้จากหัวอ่านมีค่าเกินค่าดังกล่าวจะไม่มีการบันทึก

[I] เป็นคำสั่งกำหนดค่าต่ำสุดของการบันทึก ถ้าค่าที่อ่านได้จากหัวอ่านมีค่าน้อยกว่าค่าที่กำหนดก็จะไม่มีการบันทึก

[J] เป็นคำสั่งกำหนดช่วงการบันทึกตำแหน่งลายเส้นถ้าต้องการความละเอียดมากก็ต้องกำหนดช่วงให้น้อย

5.2.2 การใช้งานการบันทึกแบบ P ร่วมกับการตัดสินใจ

เมื่อเรียกใช้งานจะแสดงรายการดังรูปที่ 5.5



รูปที่ 5.5 แสดงรายการบันทึกแบบ P ร่วมกับการตัดสินใจ

สามารถอธิบายหน้าที่ในแต่ละรายการได้ว่า

[F1] เป็นคำสั่งตั้งค่าตัวแปรต่างๆ ให้มีค่าเท่ากับค่าตอนเริ่มระบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

[F2] เป็นคำสั่งตั้งค่าควบคุม P

[F5] เป็นคำสั่งตั้งค่าความเร็ว V

[F6] เป็นคำสั่งให้ทำการติดตามลายเส้นเพื่อทำการบันทึกข้อมูล และเมื่อหัวอ่านมีการเคลื่อนที่ไปตามลายเส้นจนครบรอบก็จะหยุดการทำงานโดยอัตโนมัติ

[F8] เป็นคำสั่งยุติการทำงาน

[A] เป็นคำสั่งกำหนดค่าสูงสุดที่ระบบตัดสินใจเริ่มทำงาน ถ้าอ่านค่าจากหัวอ่านได้ค่าเกินค่าที่กำหนดนี้จะทำให้ระบบตัดสินใจเริ่มทำงานโดยการเริ่มนับจำนวนรอบการทำงานของระบบเพื่อใช้ค่าที่นับได้สำหรับการตัดสินใจต่อไป

[B] เป็นคำสั่งกำหนดค่าต่ำสุดที่ระบบตัดสินใจเริ่มทำงาน ถ้าอ่านค่าจากหัวอ่านได้ค่าน้อยกว่าค่าที่กำหนดนี้จะทำให้ระบบตัดสินใจเริ่มทำงานโดยการเริ่มนับจำนวนรอบการทำงานของระบบเพื่อใช้ค่าที่นับได้สำหรับการตัดสินใจต่อไป

[C] เป็นคำสั่งกำหนดอัตราการเปลี่ยนแปลงทิศทางการเคลื่อนที่ที่เกิดจากการตัดสินใจในขั้นแรก ถ้าหากอ่านค่าที่ได้จากหัวอ่านเกิดขอบเขตที่กำหนดไว้ก็จะทำให้เกิดการนับจำนวนรอบขึ้นและในขณะเดียวกันก็จะมีการเสริมอัตราการเปลี่ยนแปลงทิศทางการเคลื่อนที่ด้วยค่ามุมที่กำหนดนี้

[D] เป็นคำสั่งกำหนดอัตราการเปลี่ยนแปลงทิศทางการเคลื่อนที่ที่เกิดจากการตัดสินใจในขั้นที่สอง ถ้าหากอ่านค่าที่ได้จากหัวอ่านเกิดขอบเขตที่กำหนดไว้ก็จะทำให้เกิดการนับจำนวนรอบขึ้นและเมื่อนับถึงค่าที่กำหนดไว้ใน [E] ก็จะมีการเสริมอัตราการเปลี่ยนแปลงทิศทางการเคลื่อนที่ด้วยค่ามุมที่กำหนดนี้

[E] เป็นคำสั่งกำหนดค่านับรอบการทำงานเพื่อให้ระบบเพิ่มอัตราการเปลี่ยนแปลงทิศทางการเคลื่อนที่ในขั้นที่สอง

[F] เป็นคำสั่งกำหนดอัตราการเปลี่ยนแปลงทิศทางการเคลื่อนที่ที่เกิดจากการตัดสินใจในขั้นที่สาม เมื่อระบบมีการนับจำนวนรอบต่อจากค่าที่กำหนดไว้ใน [E] และเมื่อนับถึงค่าที่กำหนดไว้ใน [G] ก็จะมีการเสริมอัตราการเปลี่ยนแปลงทิศทางการเคลื่อนที่ด้วยค่ามุมที่กำหนดนี้ ถ้าหากมีการนับมาจนถึงขั้นนี้แสดงว่าระบบเริ่มมีการทำงานที่ไม่น่าไว้วางใจแล้ว และมีโอกาสหลุดออกจากการติดตามลายเส้นได้ ดังนั้นอัตราการเปลี่ยนแปลงทิศทางการเคลื่อนที่จึงควรจะมีค่าที่มากกว่าปกติเพื่อให้ระบบมีการวิ่งเข้าหาลายเส้นโดยเร็ว

[G] เป็นคำสั่งกำหนดค่านับรอบการทำงานเพื่อให้ระบบเพิ่มอัตราการเปลี่ยนแปลงทิศทางการเคลื่อนที่ในขั้นที่สาม เมื่อระบบมีการนับรอบการทำงานครบรอบตามที่กำหนดนี้ก็จะนับใหม่แต่ไม่ใช่เริ่มต้นนับใหม่ตั้งแต่ต้น

[H] เป็นคำสั่งกำหนดค่าสูงสุดของการบันทึก ถ้าค่าที่อ่านได้จากหัวอ่านมีค่าเกินค่าดังกล่าวจะไม่มีการบันทึก

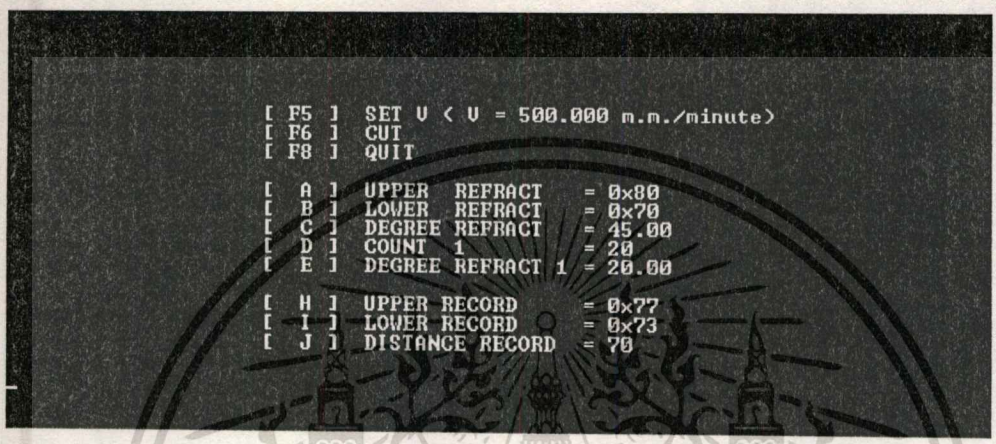
[I] เป็นคำสั่งกำหนดค่าต่ำสุดของการบันทึก ถ้าค่าที่อ่านได้จากหัวอ่านมีค่าน้อยกว่าค่าที่กำหนดก็จะไม่มีการบันทึก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

[J] เป็นคำสั่งกำหนดช่วงการบันทึกตำแหน่งลายเส้นถ้าต้องการความละเอียดมากก็ต้องกำหนดช่วงให้น้อย

5.2.3 การใช้งานการบันทึกแบบการสะท้อนของแสงร่วมกับการตัดสินใจ

เมื่อเรียกใช้งานจะแสดงรายการดังรูปที่ 5.6



รูปที่ 5.6 แสดงรายการบันทึกแบบสะท้อนของแสงร่วมกับการตัดสินใจ

สามารถอธิบายหน้าที่ในแต่ละรายการได้ว่า

[F5] เป็นคำสั่งตั้งค่าความเร็ว V

[F6] เป็นคำสั่งให้ทำการติดตามลายเส้นเพื่อทำการบันทึกข้อมูล และเมื่อหัวอ่านมีการเคลื่อนที่ไปตามลายเส้นจนครบรอบก็จะหยุดการทำงานโดยอัตโนมัติ

[F8] เป็นคำสั่งยุติการทำงาน

[A] เป็นคำสั่งกำหนดตำแหน่งสูงสุดที่ทำให้การเคลื่อนที่เกิดการสะท้อน

[B] เป็นคำสั่งกำหนดตำแหน่งต่ำสุดที่ทำให้การเคลื่อนที่เกิดการสะท้อน

[C] เป็นคำสั่งกำหนดมุมสะท้อนปกติ

[D] เป็นคำสั่งกำหนดค่านับจำนวนรอบการทำงานเพื่อให้ระบบตัดสินใจเริ่มทำงาน หมายถึงว่า เมื่อหัวอ่านอ่านค่าได้เกินขอบเขตตำแหน่งสะท้อนทั้งสูงสุดและต่ำสุดก็จะเริ่มทำการนับรอบการทำงานของระบบ และเมื่อนับรอบการทำงานได้ถึงค่าที่กำหนดก็จะเริ่มนับใหม่พร้อมทั้งเปลี่ยนทิศทางการเคลื่อนที่หลักด้วยค่าที่กำหนดใน [E] เพื่อให้ระบบเข้าสู่การติดตามลายเส้นตามปกติต่อไป

[E] เป็นคำสั่งกำหนดอัตราการเปลี่ยนแปลงทิศทางการเคลื่อนที่หลักสำหรับ [D]

[H] เป็นคำสั่งกำหนดค่าสูงสุดของการบันทึก ถ้าค่าที่อ่านได้จากหัวอ่านมีค่าเกินค่าดังกล่าวจะไม่มีการบันทึก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

[I] เป็นคำสั่งกำหนดค่าต่ำสุดของการบันทึก ถ้าค่าที่อ่านได้จากหัวอ่านมีค่าน้อยกว่าค่าที่กำหนดก็จะไม่มีการบันทึก

[J] เป็นคำสั่งกำหนดช่วงการบันทึกตำแหน่งลายเส้นถ้าต้องการความละเอียดมากก็ต้องกำหนดช่วงให้น้อย

5.3 การตัดชิ้นงานด้วยข้อมูลที่มีอยู่ในหน่วยความจำ

เมื่อเรียกใช้งานจะแสดงรายการดังรูปที่ 5.7

รูปที่ 5.7 แสดงรายการการตัดโดยข้อมูลจากหน่วยความจำ

สามารถอธิบายหน้าที่ในแต่ละรายการได้ว่า

[F4] เป็นคำสั่งกำหนดขนาดขยายชิ้นงานที่ต้องการตัด

[F5] เป็นคำสั่งกำหนดความเร็วสำหรับการตัด

[F6] เป็นคำสั่งให้ระบบเริ่มทำการตัดชิ้นงาน ระหว่างที่ทำการตัดสามารถหยุดได้โดยกดคีย์ใดๆ

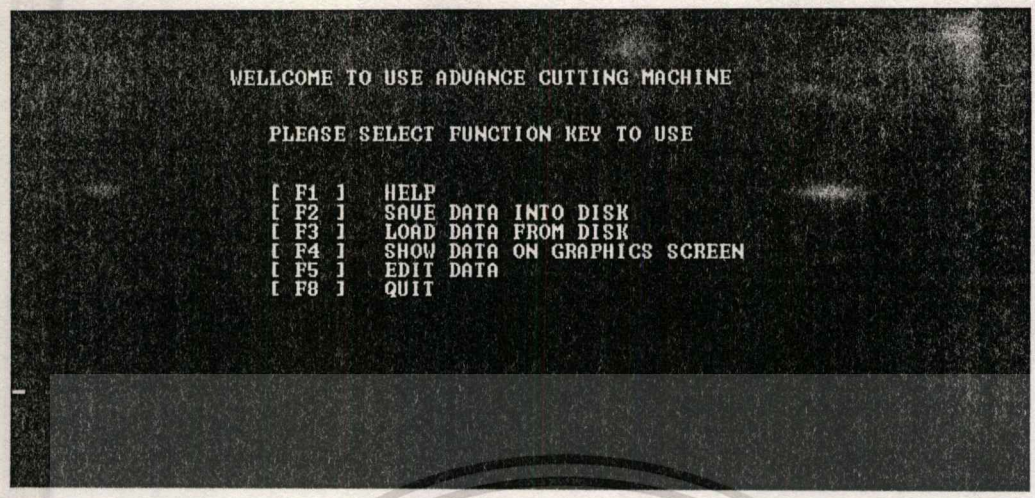
[F8] เป็นคำสั่งยกเลิกการทำงาน

5.4 โปรแกรมปรับปรุงข้อมูล

การทำงานของโปรแกรมติดตามและบันทึกลายเส้นนั้นอาจจะได้ข้อมูลที่ไม่ถูกใจนักดังนั้นจึงได้ทำการพัฒนาโปรแกรมอีกชุดหนึ่งขึ้นมาเพื่อทำการแก้ไขข้อมูลดังกล่าว เพื่อให้ข้อมูลมีลักษณะที่ต้องการ โดยมีการใช้งานดังต่อไปนี้

เมื่อเรียกใช้งานระบบจะมีการแสดงดังรูปที่ 5.8

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5.8 แสดงรายการของโปรแกรมแก้ไข

สามารถอธิบายหน้าที่ในแต่ละรายการได้ว่า

[F1] เป็นคำสั่งขอคำแนะนำ(ยังไม่มีการพัฒนาในส่วนนี้)

[F2] เป็นคำสั่งเก็บข้อมูลจากหน่วยความจำสู่จานเก็บแม่เหล็ก

[F3] เป็นคำสั่งอ่านข้อมูลจากจานแม่เหล็กสู่หน่วยความจำ

[F4] เป็นคำสั่งแสดงข้อมูลในหน่วยความจำเป็นรูปภาพบนจอ

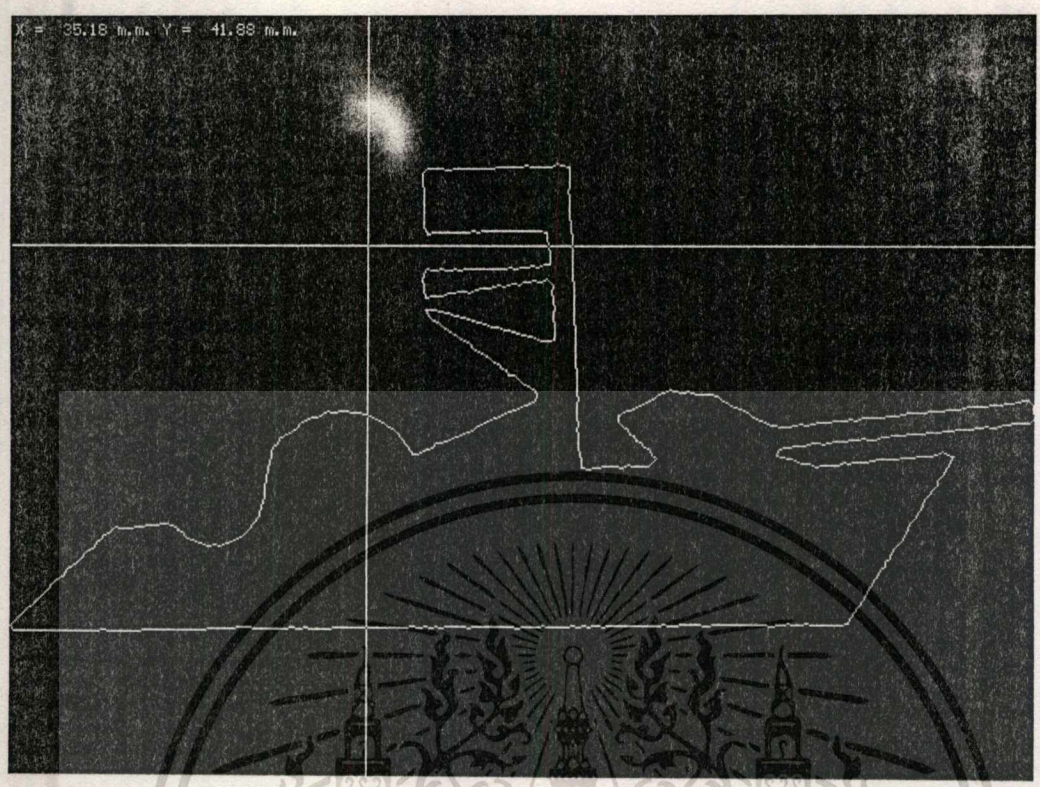
[F5] เป็นคำสั่งเข้าสู่โปรแกรมแก้ไข ซึ่งมีขั้นตอนการใช้งานในหัวข้อที่ 5.4.1

[F8] เป็นคำสั่งออกจากโปรแกรม

5.4.1 การใช้งานโปรแกรมแก้ไขข้อมูล

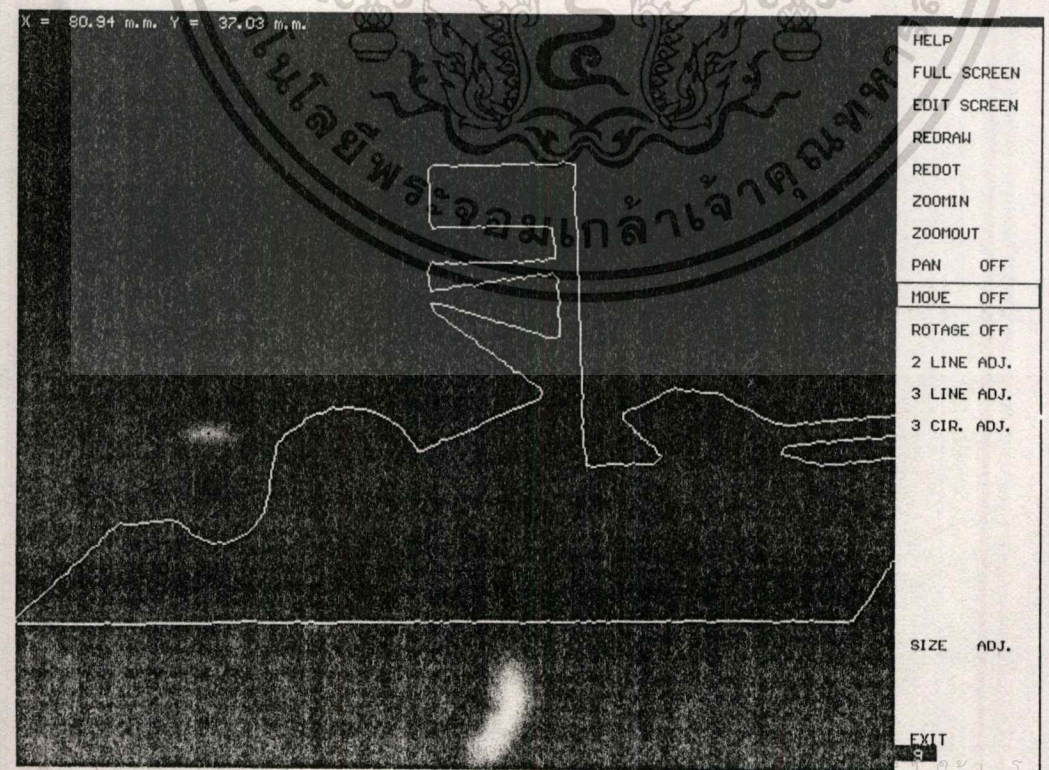
เมื่อสู่โปรแกรมนี้จะมีการแสดงรูปที่มีอยู่ในหน่วยความจำดังรูปที่ 5.9

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5.9 การแสดงรูปข้อมูลบนจอ

จากรูปจะเห็นว่ามีเพียงรูปภาพและตัวเลขแสดงตำแหน่งตัวชี้เท่านั้น แต่เมื่อเลื่อนตัวชี้ไปที่ริมจอภาพทางขวาสุดจะปรากฏรายการต่างๆ ให้เลือกดังรูปที่ 5.10

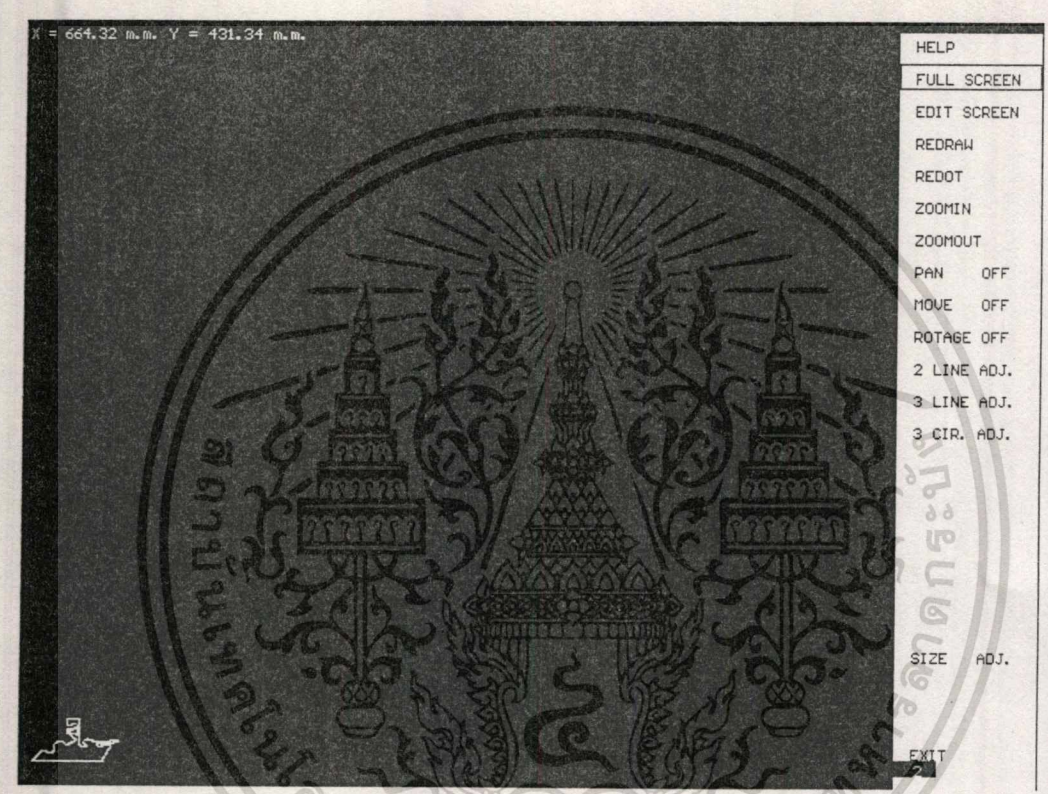


รูปที่ 5.10 การแสดงรายการแก้ไข

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปจะเห็นตัวชี้ตำแหน่งมีการเปลี่ยนจากจุดตัดของเส้นตรงกลายไปเป็นเส้นกรอบบอกตำแหน่งรายการที่จะเลือกแทน ซึ่งรายการที่จะทำการเลือกจะมีหน้าที่ต่างๆ ดังต่อไปนี้

FULL SCREEN เป็นคำสั่งให้แสดงขนาดชิ้นงานที่เป็นจริงเมื่อเทียบกับขนาดโต๊ะ ซึ่งก็คือการแสดงขนาดของโต๊ะให้เต็มจอพอดี และขนาดของชิ้นงานก็จะมีขนาดเป็นสัดส่วนกับจอตามลำดับดังตัวอย่างรูปที่ 5.11



รูปที่ 5.11 แสดงขนาดชิ้นงานเทียบกับขนาดโต๊ะ

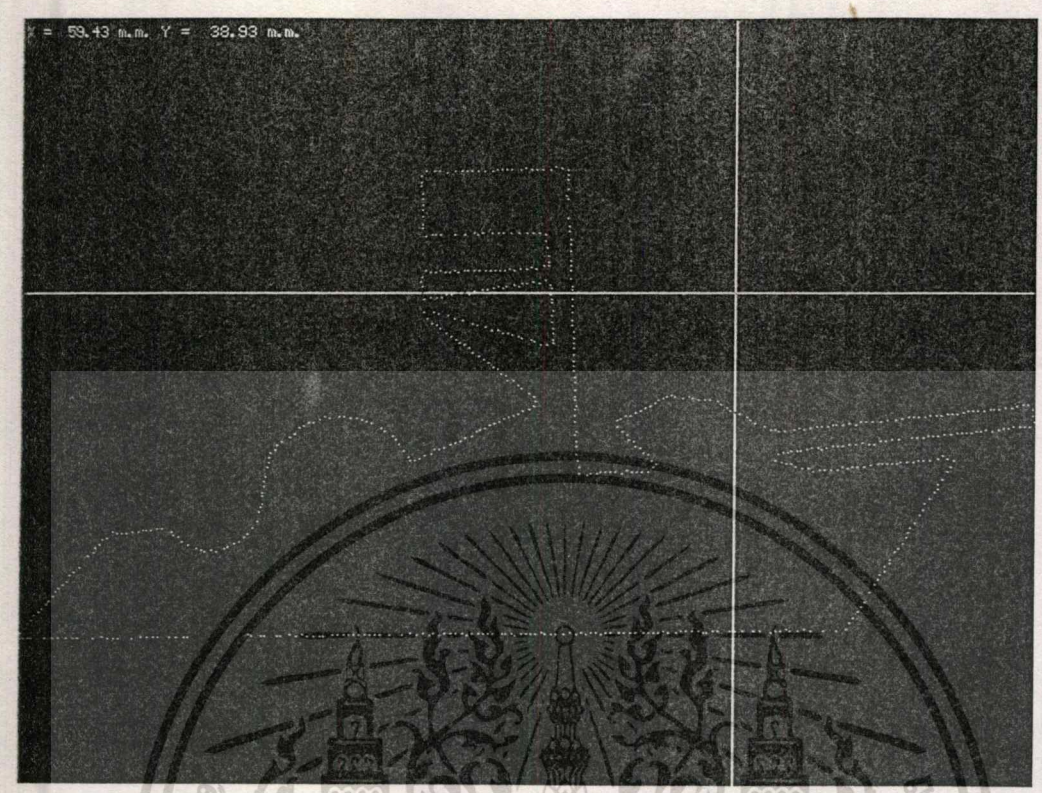
EDIT SCREEN เป็นคำสั่งให้แสดงข้อมูลเต็มจอภาพ กล่าวคือไม่ว่าข้อมูลจะมีขนาดใหญ่หรือเล็ก ก็จะถูกขยายให้มีขนาดเต็มจอภาพ

REDRAW เป็นคำสั่งให้ทำการวาดรูปข้อมูลใหม่ ในการทำงานแก้ไขบางอย่างอาจก่อให้เกิดริ้วรอยบางอย่าง ดังนั้นจึงอาจจำเป็นที่จะต้องทำการวาดรูปใหม่ในเงื่อนไขเดิมเพื่อให้รูปตามความเป็นจริง

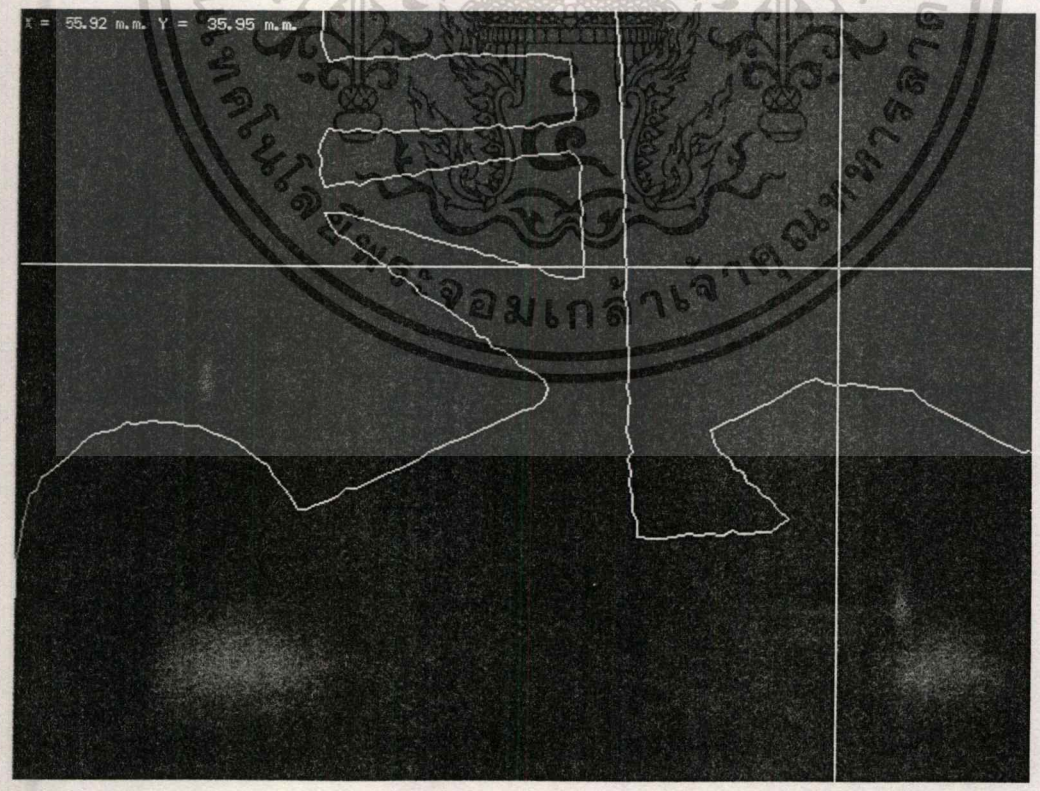
REDOT เป็นคำสั่งให้แสดงข้อมูลในลักษณะเป็นจุดๆ ซึ่งจุดที่ปรากฏบนจอก็คือข้อมูลบ่งตำแหน่งที่ได้เก็บไว้ ซึ่งจะมีความถี่ของจุดเท่าใดก็ขึ้นอยู่กับความละเอียดในการบันทึก ดูตัวอย่างรูปที่ 5.12

ZOOMIN เป็นคำสั่งแสดงรูปขยายของข้อมูลเพิ่มอีกสองเท่าจากรูปปัจจุบัน ดังนั้นขนาดของรูปที่ปรากฏบนจอก็จะมีขนาดใหญ่เพิ่มขึ้น ดังตัวอย่างรูปที่ 5.13

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5.12 การแสดงข้อมูลเป็นจุดๆบนจอภาพ



รูปที่ 5.13 การแสดงข้อมูลโดยขยายเพิ่มขึ้นสองเท่า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ZOOMOUT เป็นคำสั่งย่อรูปภาพบนจอลงสองเท่า ดูตัวอย่างรูปที่ 5.14

รูปที่ 5.14 การแสดงข้อมูลโดยย่อลงสองเท่า

PAN เป็นคำสั่งเกี่ยวกับการเลื่อนรูปภาพ ทำให้สามารถเลื่อนส่วนที่ต้องการแก้ไขให้มาอยู่กึ่งกลางหน้าจอได้พอดี มีวิธีการเรียกใช้ดังนี้

1. เลื่อนตัวชี้ไปทางขวาสุดจะปรากฏรายการขึ้น ให้เลื่อนกรอบไปที่ PAN OFF แล้วกดปุ่มเพื่อเลือก จะสังเกตเห็นว่าตัวอักษรขึ้นคำว่า PAN ON แทน หมายถึงตอนนี้โปรแกรมเลื่อนภาพกำลังทำงาน

2. เลื่อนตัวชี้ไปที่ตำแหน่งที่ต้องการแล้วกดคีย์ ENTER รูปภาพก็เกิดการเลื่อนตามที่ต้องการทันที ตำแหน่งที่ถูกเลือกจะถูกเลื่อนมาไว้ที่กึ่งกลางพอดี

คำสั่งนี้สามารถเรียกใช้ได้ตลอดเวลา

ดูการใช้งานในรูปที่ 5.15

MOVE เป็นคำสั่งเกี่ยวกับการเลื่อนข้อมูล มีวิธีการใช้งานคล้ายกับการเลื่อนรูปภาพ แตกต่างกันตรงที่คำสั่ง MOVE เป็นการเลื่อนย้ายทางตำแหน่งจริงข้อมูลในหน่วยความจำจะมีการเปลี่ยนแปลงไปด้วย คำสั่งนี้มีประโยชน์ในการปรับตำแหน่งข้อมูลให้ไปอยู่ในส่วนใดๆ ของโต๊ะก็ได้

มีการสั่งงานดังต่อไปนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1. เลื่อนตัวชี้ไปทางขวาสุดจะปรากฏรายการขึ้น ให้เลื่อนกรอบไปที่ MOVE OFF แล้วกดปุ่มเลือก จะสังเกตเห็นว่าตัวอักษรจะขึ้นคำว่า MOVE ON แทน หมายถึงตอนนี้โปรแกรมเลื่อนข้อมูลกำลังทำงาน

2. เลื่อนตัวชี้ไปที่ตำแหน่งที่ต้องการเลือกแล้วกดคีย์ ENTER สังเกตดูจะเห็นว่าจะปรากฏเส้นสีแดงลากเชื่อมระหว่างตัวชี้กับตำแหน่งที่เลือกไว้ตลอดเวลา

3. เลื่อนตัวชี้ไปที่ตำแหน่งปลายทางที่ต้องการแล้วกดคีย์ ENTER รูปภาพก็เกิดการเลื่อนตามที่ต้องการทันที

ดูการใช้งานในรูปที่ 5.16

ROTAGE เป็นคำสั่งเกี่ยวกับการหมุนตำแหน่งข้อมูลรอบจุดกึ่งกลางข้อมูล มีวิธีการใช้งานดังต่อไปนี้

1. เลื่อนตัวชี้ไปเลือกตำแหน่งสองจุดบนจอภาพเพื่อกำหนดมุมเริ่มต้น สังเกตว่าจะมีการลากเส้นเชื่อมระหว่างจุดที่เลือกครั้งแรกกับตัวชี้ตลอดเวลา

2. เลื่อนตัวชี้ไปเลือกตำแหน่งที่สามเพื่อกำหนดมุมปลายทาง เมื่อเลือกแล้วจะสังเกตเห็นว่าระนาบเส้นตรงที่เลือกไว้ในครั้งแรกถูกหมุนให้ขนานกับระนาบเส้นตรงที่เลือกไว้ในครั้งที่สอง

ดูการใช้งานในรูปที่ 5.17

2 LINE ADJ. เป็นคำสั่งปรับข้อมูลระหว่างจุดสองจุดให้เป็นเส้นตรง โดยจุดเดิมที่อยู่ระหว่างจุดสองจุดที่เลือกก็จะถูกลบไปด้วย มีประโยชน์ในการปรับข้อมูลที่ได้จากการบันทึกแล้วมีเส้นตรงที่ไม่เรียบให้เรียบขึ้นได้ มีวิธีการใช้งานดังต่อไปนี้

1. เลื่อนตัวชี้เลือกตำแหน่งแรกเป็นจุดเริ่มต้นและต้องเลือกให้อยู่บนจุดข้อมูล

2. เลื่อนตัวชี้เลือกตำแหน่งที่สองเป็นจุดปลายทางและต้องเลือกให้อยู่บนจุดข้อมูลเช่นกัน ก็จะเกิดการลากเส้นตรงเชื่อมระหว่างจุดสองจุดดังกล่าว

ดูการใช้งานในรูปที่ 5.18

3 LINE ADJ. เป็นคำสั่งวาดเส้นตรงสองเส้นที่ทำมุมกัน มีประโยชน์ในการสร้างมุมของรูปให้มีความคมยิ่งขึ้น มีวิธีการใช้งานดังต่อไปนี้

1. เลื่อนตัวชี้เลือกตำแหน่งแรกเป็นจุดเริ่มต้นและต้องเลือกให้อยู่บนจุดข้อมูล

2. เลื่อนตัวชี้เลือกตำแหน่งที่สองเป็นจุดปลายทางและต้องเลือกให้อยู่บนจุดข้อมูลเช่นกัน ก็จะเกิดการลากเส้นตรงเชื่อมระหว่างจุดสามจุดคือจุดแรกกับตัวชี้และตัวชี้กับจุดที่สอง

3. เลื่อนตัวชี้ไปยังจุดที่ต้องการให้เกิดมุม ก็ได้เส้นตรงที่บรรจบกันเกิดมุมขึ้นดังกล่าว

ดูการใช้งานในรูปที่ 5.19

3 CIRCLE ADJ. เป็นคำสั่งวาดวงกลมโดยกำหนดจุดสามจุด มีประโยชน์ในการปรับเส้นโค้งของข้อมูลให้มีความราบเรียบยิ่งขึ้น มีวิธีการใช้งานดังต่อไปนี้

1. เลื่อนตัวชี้เลือกตำแหน่งแรกและสองบนจุดข้อมูล

2. เลื่อนตัวชี้เพื่อกำหนดค่ารัศมีวงกลมที่ต้องการ ก็จะได้ส่วนโค้งวงกลมที่ต้องการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ดูการใช้งานในรูปที่ 5.20

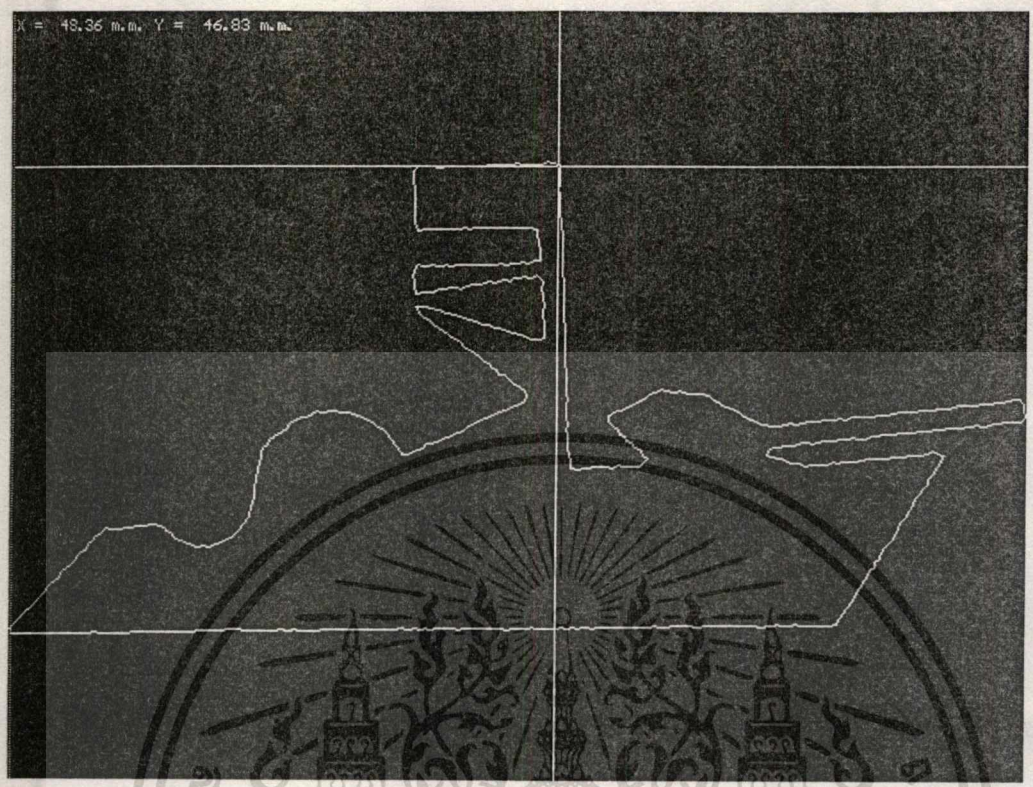
SIZE ADJ. เป็นคำสั่งปรับขนาดข้อมูลให้ขยายใหญ่ขึ้นหรือเล็กลงได้ตามต้องการ มีการสั่งงานเช่นเดียวกับคำสั่ง ROTAGE มีวิธีการใช้งานดังต่อไปนี้

1. เลื่อนตัวชี้กำหนดจุดสองจุดเพื่อกำหนดขนาดปัจจุบัน
2. เลื่อนตัวชี้กำหนดจุดที่สามเพื่อกำหนดขนาดที่ต้องการ

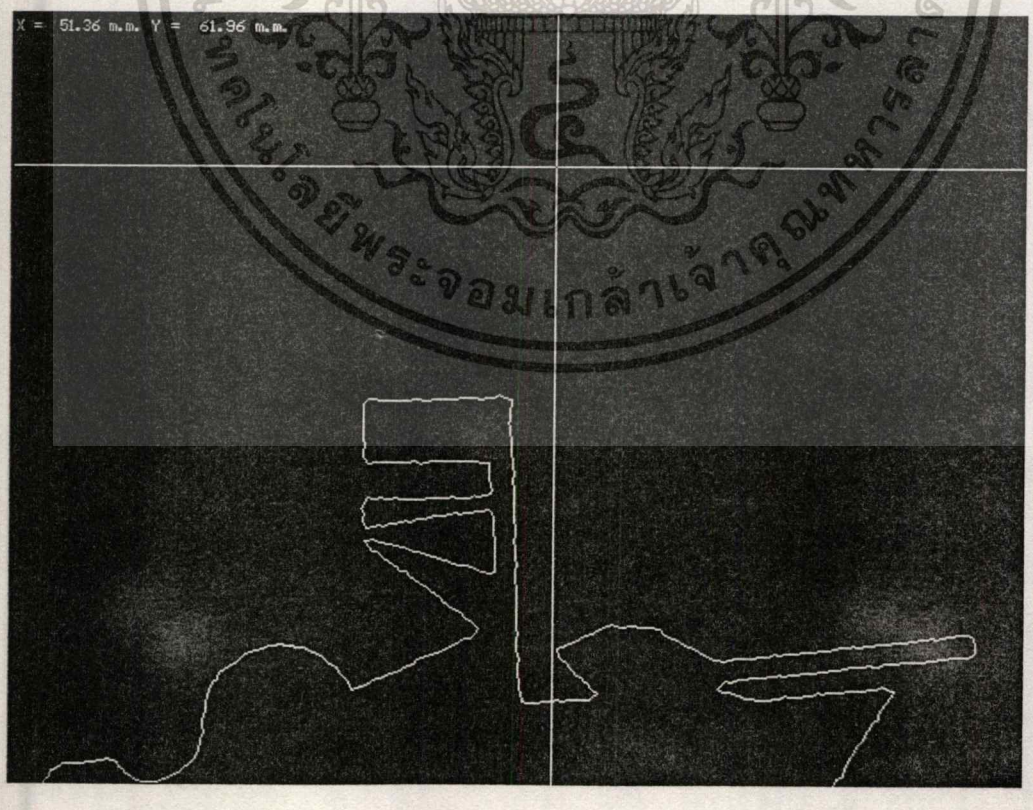
ดูการใช้งานในรูปที่ 5.21

EXIT เป็นคำสั่งออกจากโปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



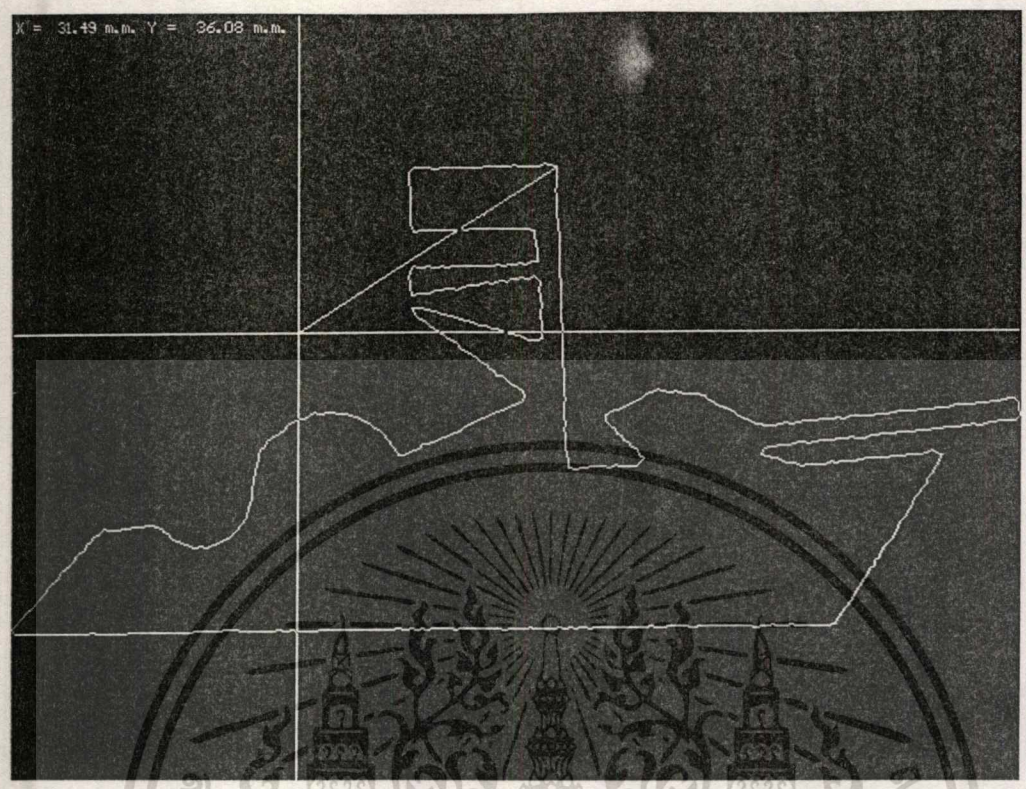
ก. เลือกตำแหน่งที่ต้องการ



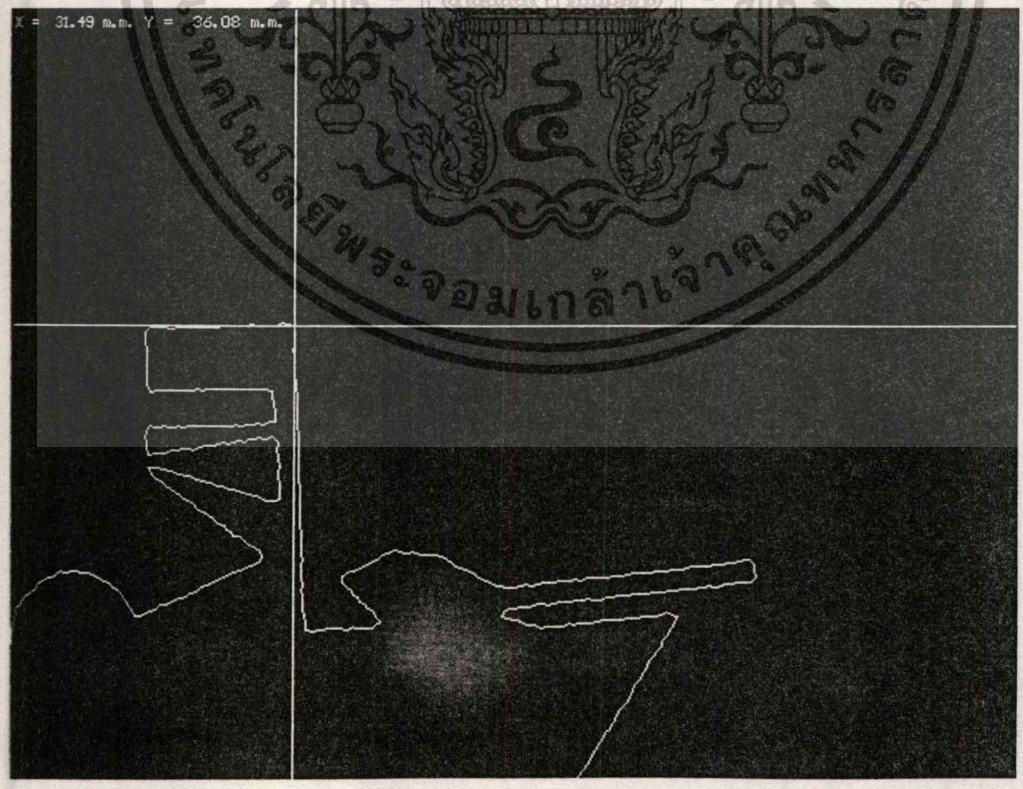
ข. ผลที่เกิดขึ้น

รูปที่ 5.15 แสดงการใช้งานคำสั่ง PAN

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



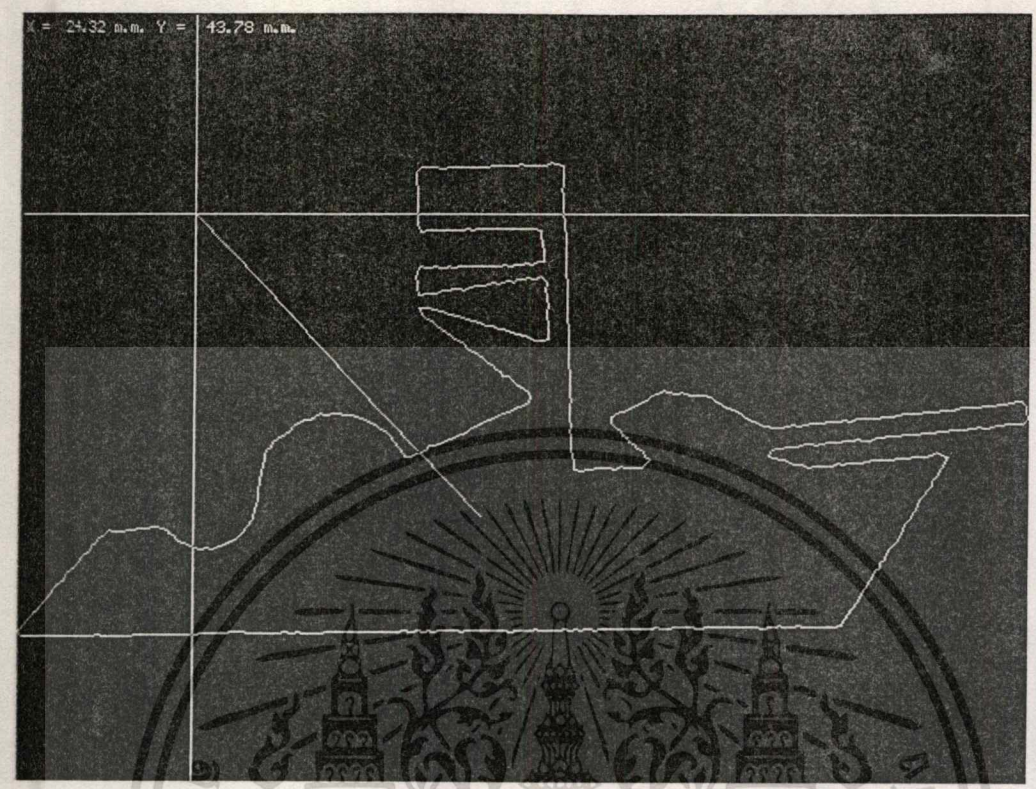
ก. เลือกตำแหน่งที่ต้องการ



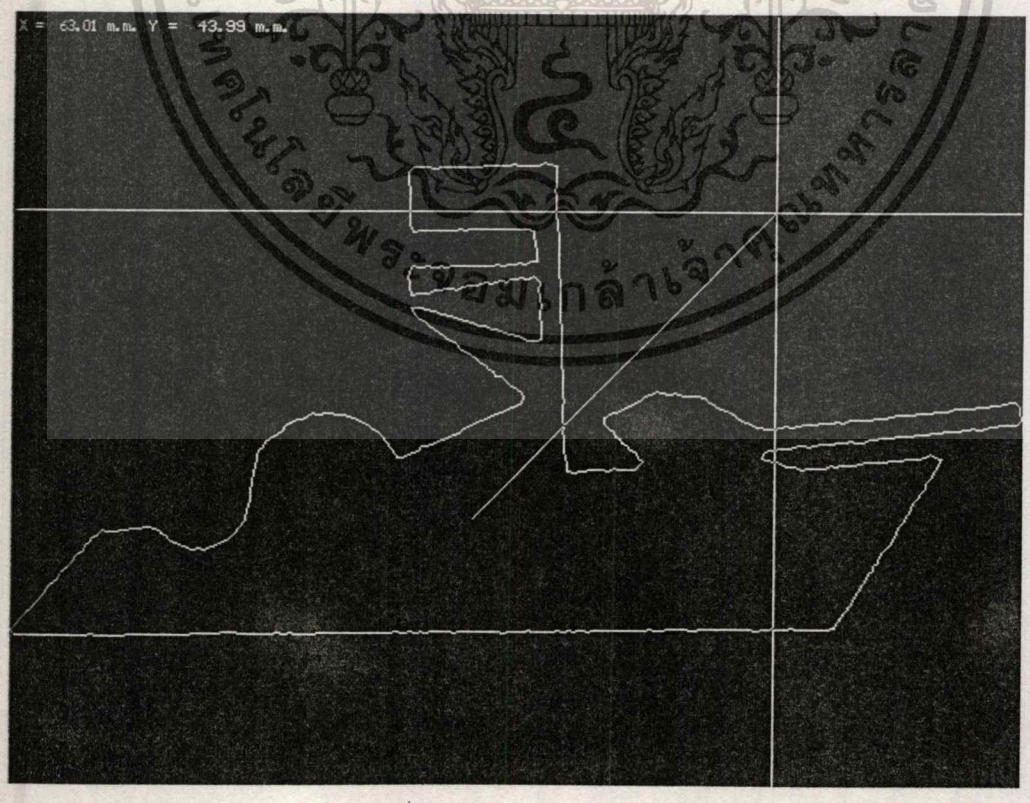
ข. ผลที่เกิดขึ้น

รูปที่ 5.16 แสดงการใช้งานคำสั่ง MOVE

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

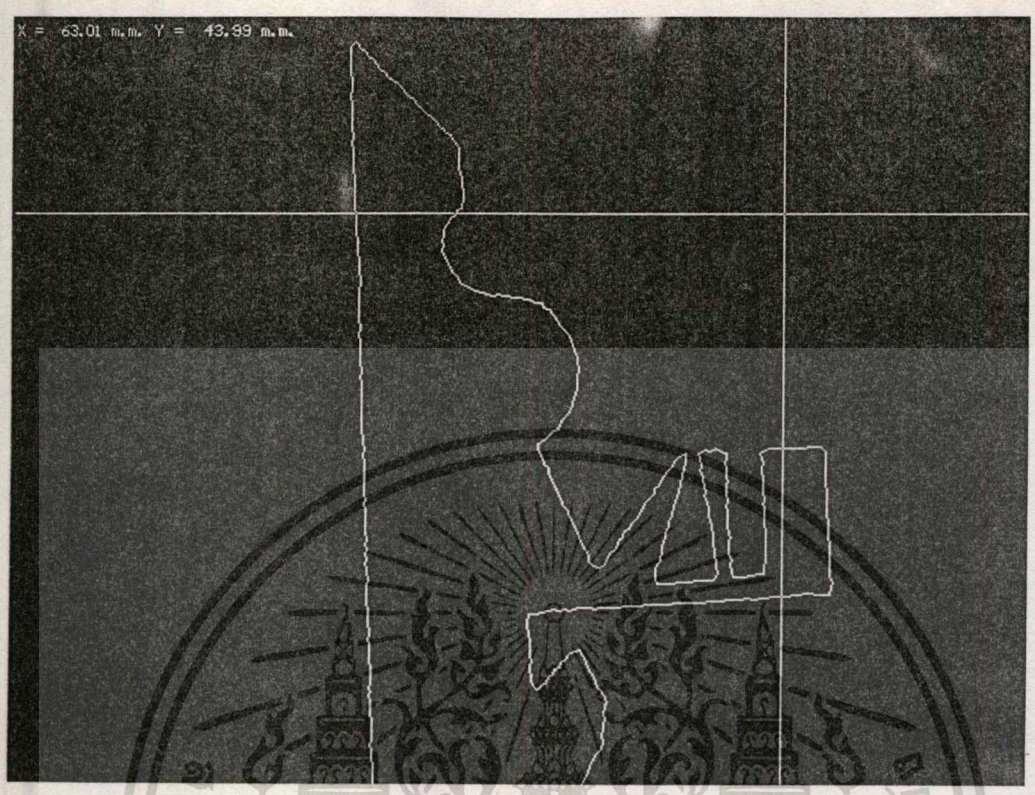


ก. เลือกมุมเริ่มต้น



ข. เลือกมุมปลายทาง

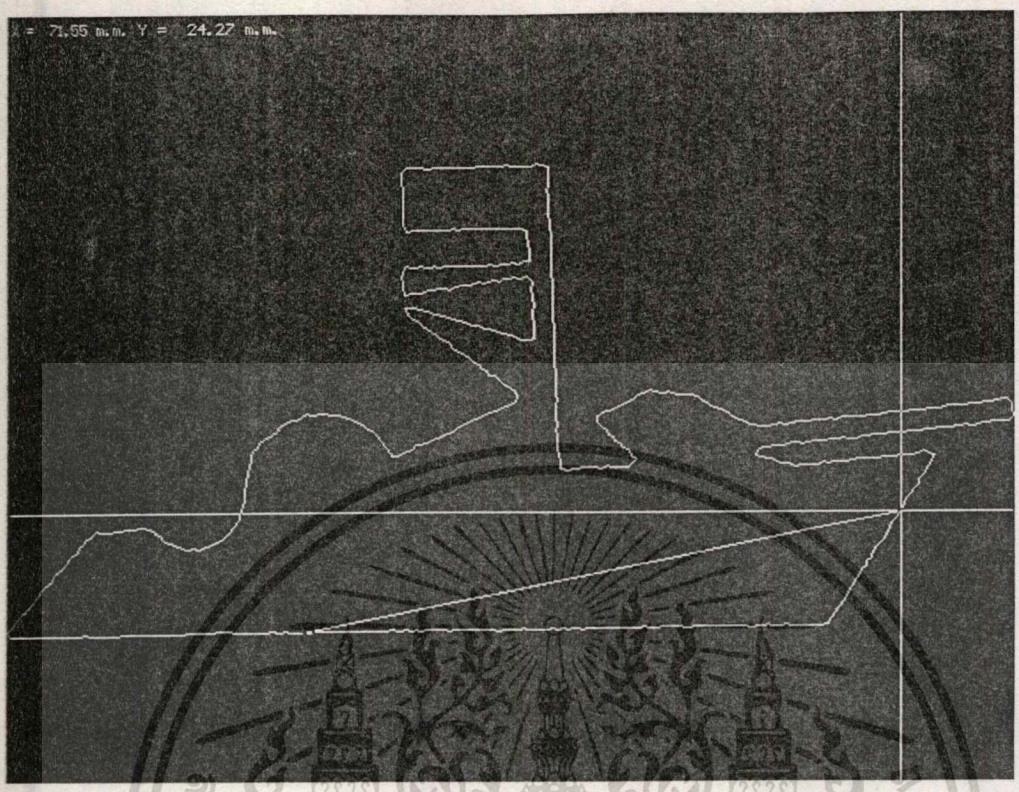
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



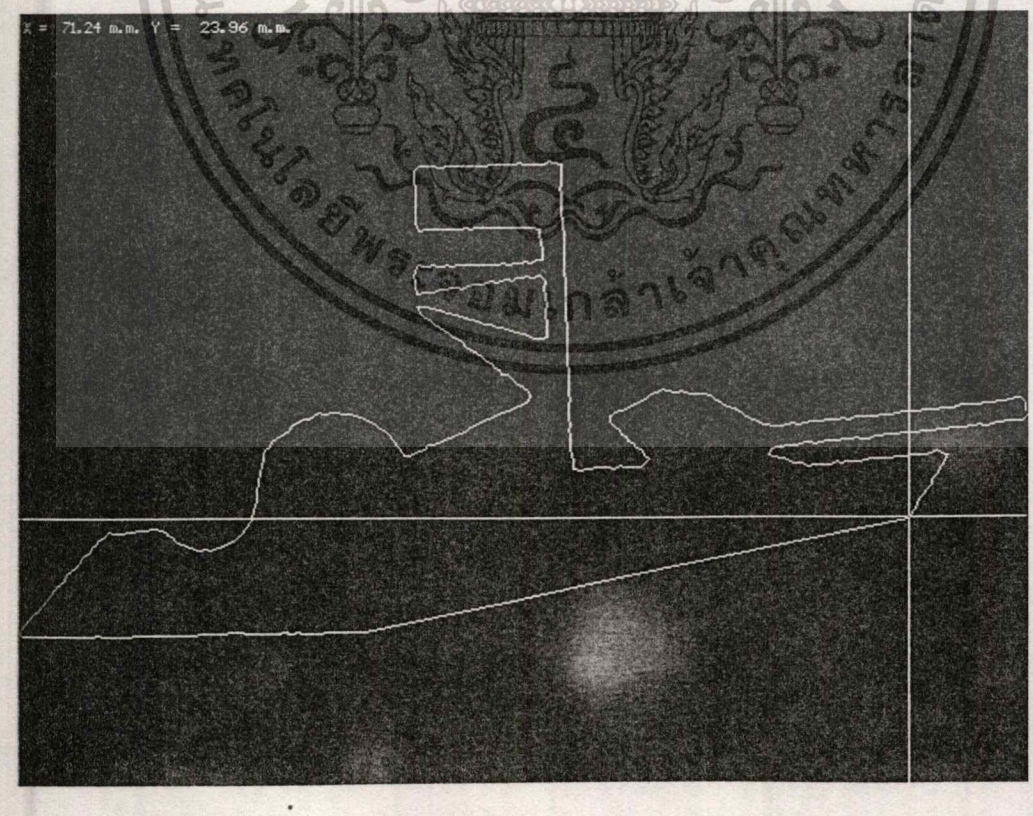
ค. ผลที่เกิดขึ้น

รูปที่ 5.17 แสดงการใช้งานคำสั่ง ROTATE

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



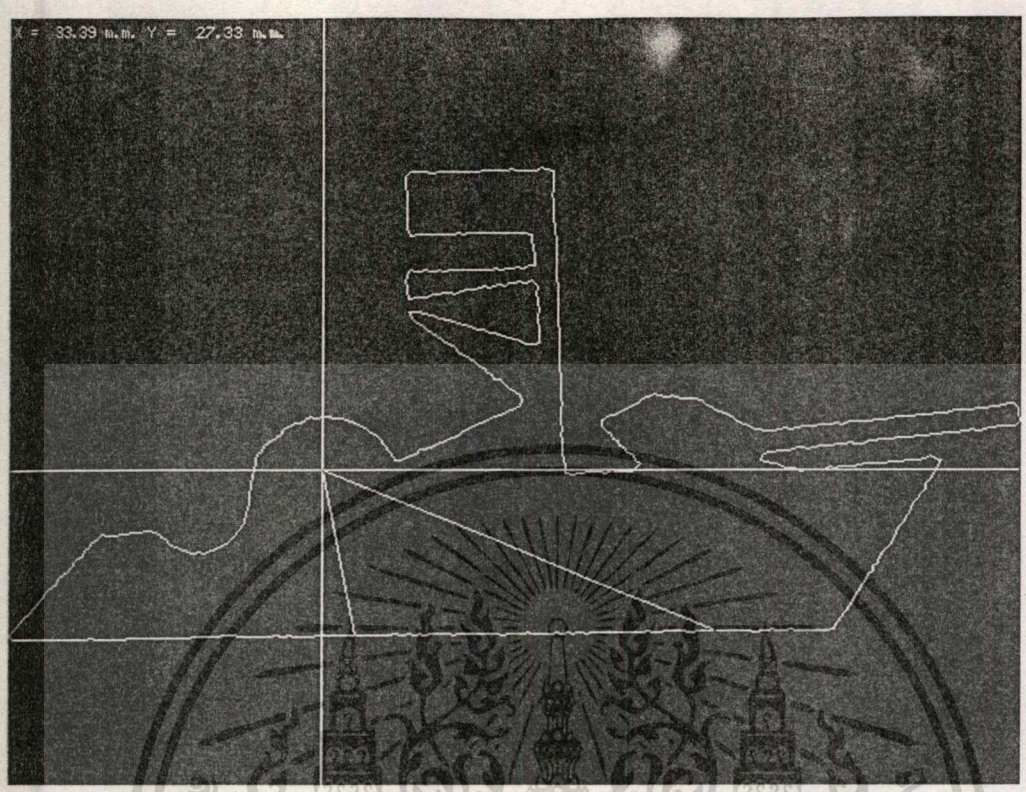
ก. เลือกจุดสองจุดบนจุดข้อมูล



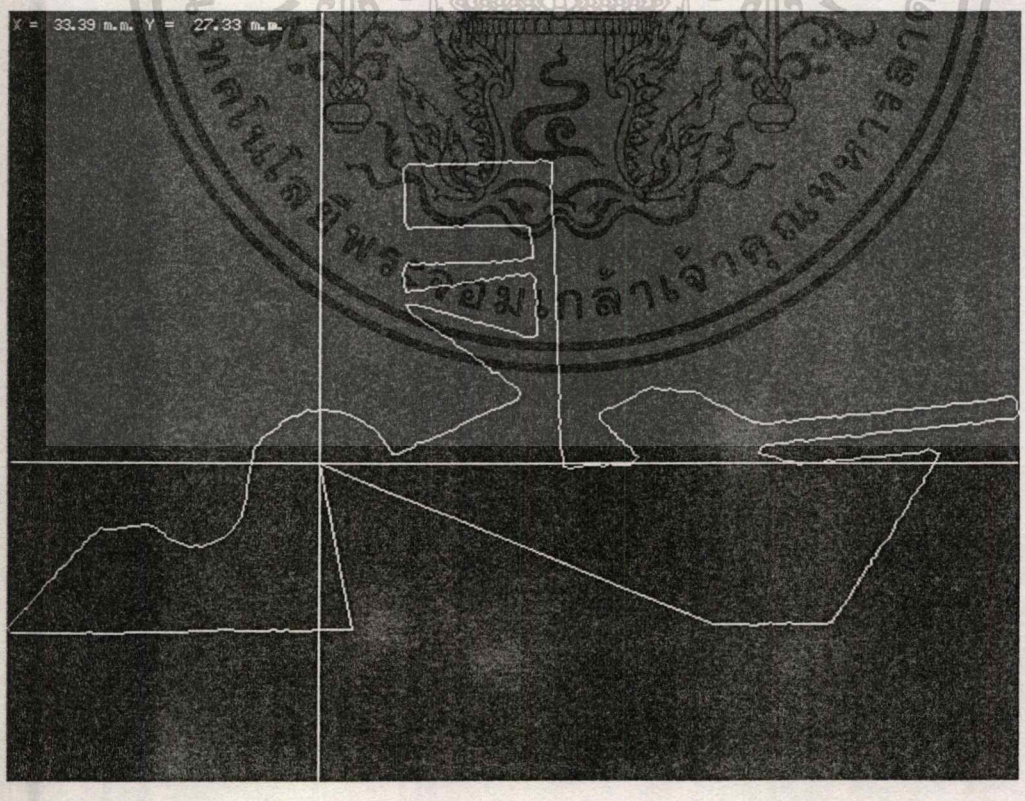
ค. ผลที่เกิดขึ้น

รูปที่ 5.18 แสดงการใช้คำสั่ง 2 LINE ADJ.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



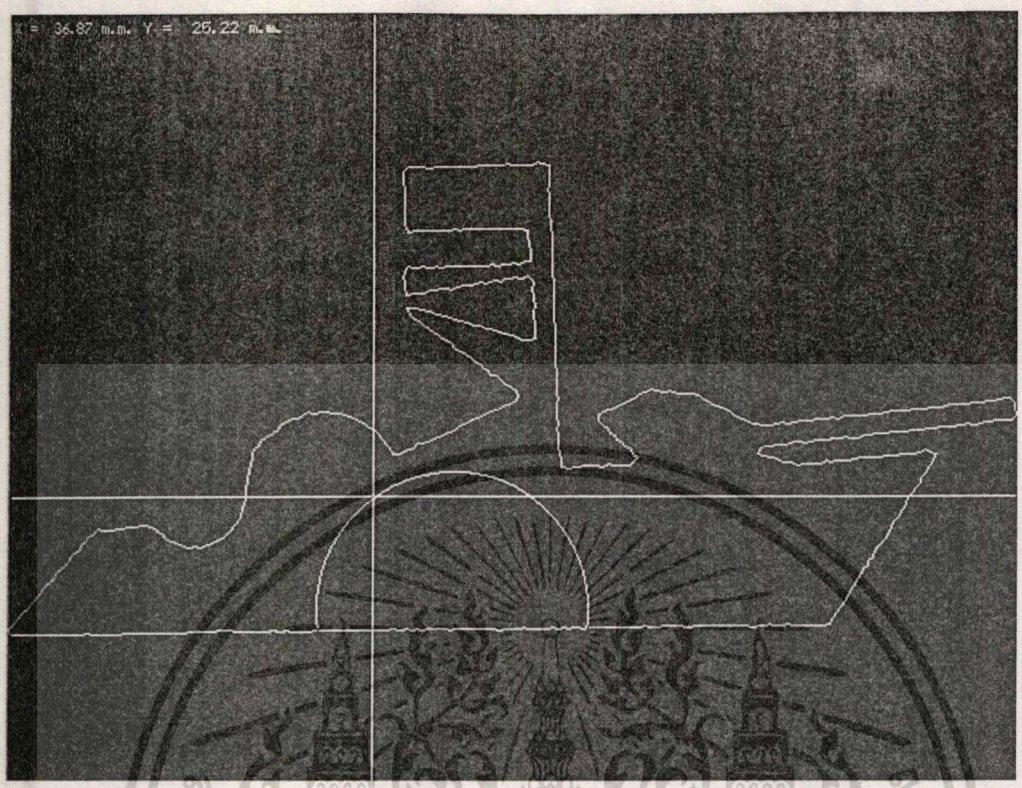
ก. เลือกตำแหน่งและมุมที่ต้องการ



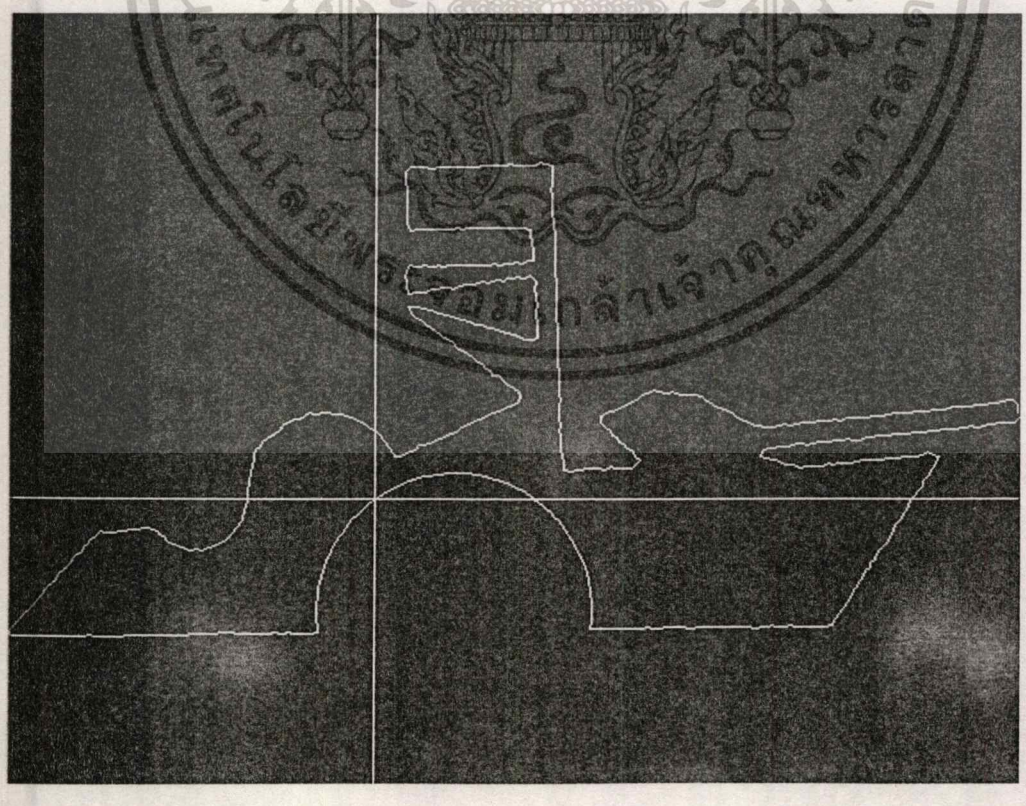
ข. ผลที่เกิดขึ้น

รูปที่ 5.19 แสดงการใช้งานคำสั่ง 3 LINE ADJ.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



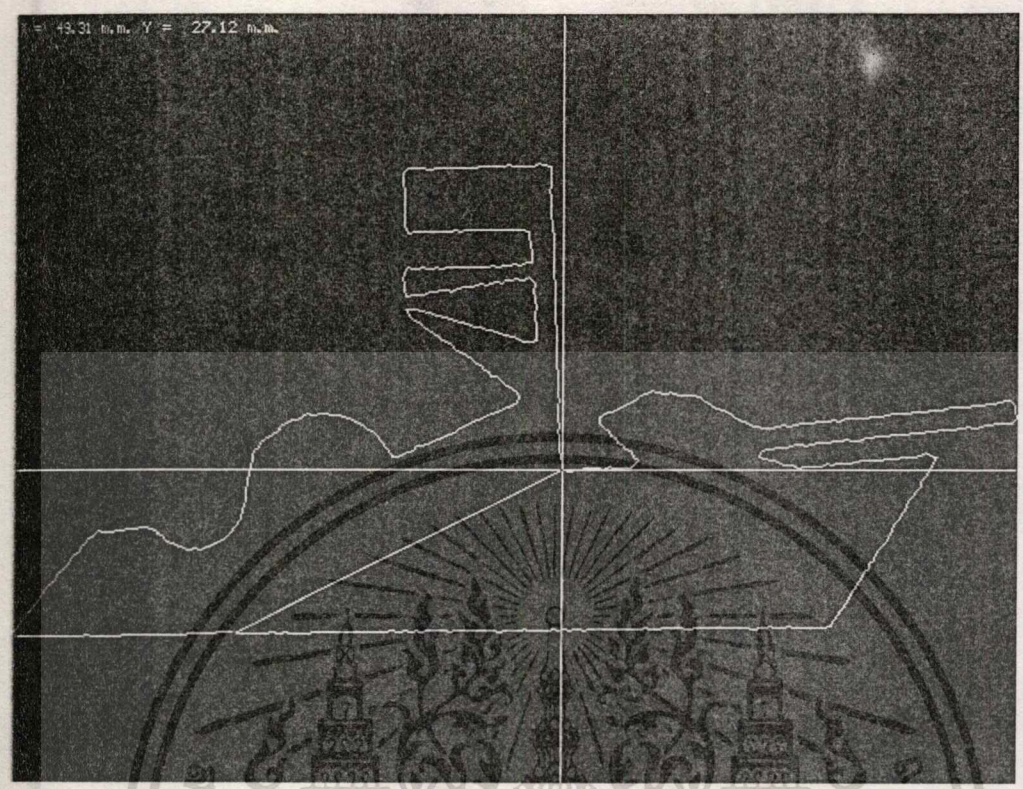
ก. เลือกจุดสองจุดและขนาดรัศมีที่ต้องการ



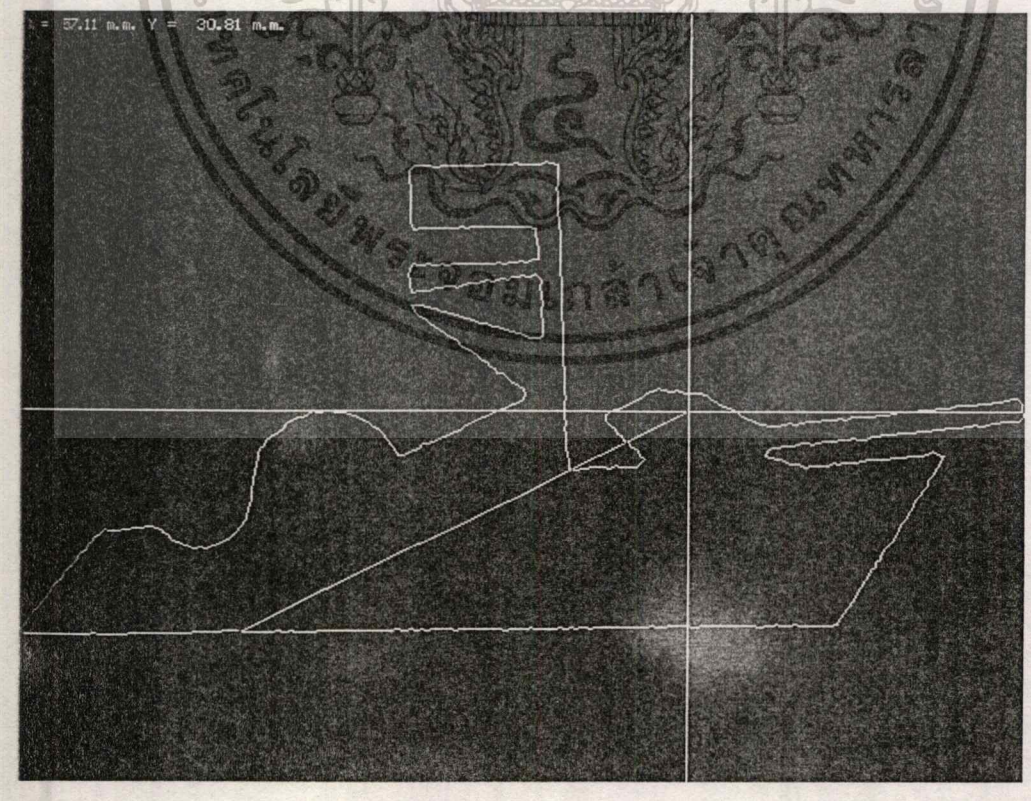
ข. ผลที่เกิดขึ้น

รูปที่ 5.20 แสดงการใช้งานคำสั่ง 3 CIR. ADJ.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

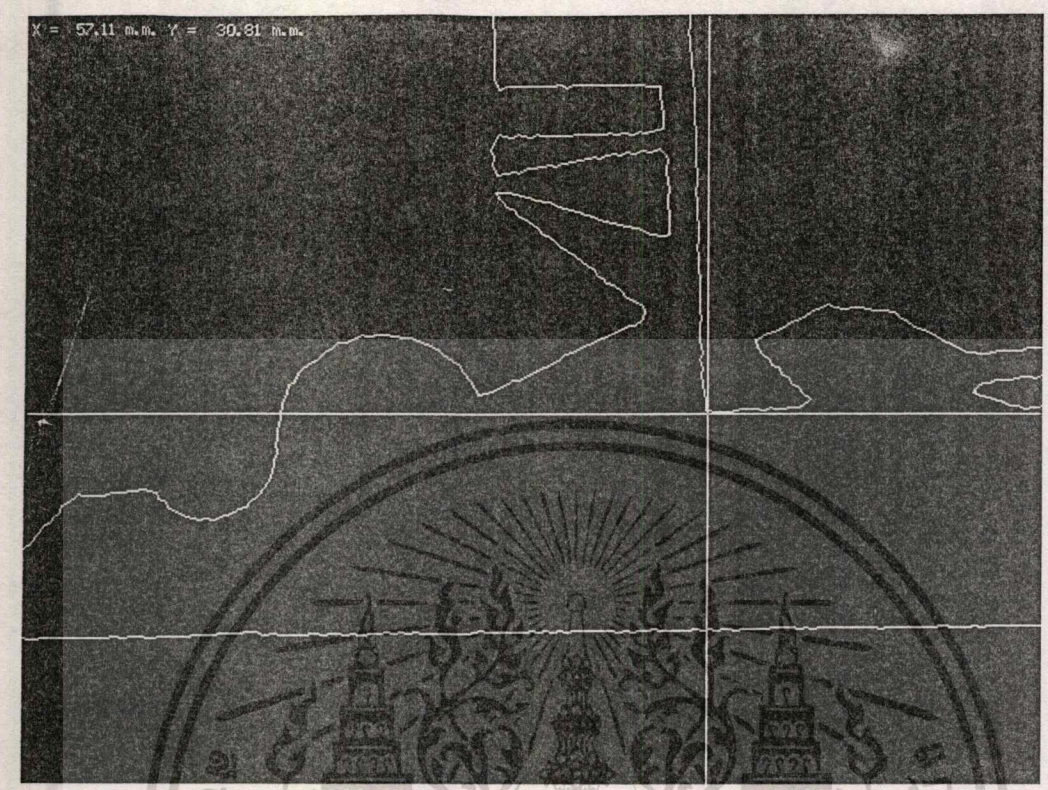


ก. กำหนดขนาดปัจจุบัน



ข. กำหนดขนาดที่ต้องการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ค. ผลที่เกิดขึ้น

รูปที่ 5.21 แสดงการใช้คำสั่ง SIZE ADJ.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 6 การทดสอบการทำงาน

ในการทดสอบนี้จะเป็นการทดสอบใช้งานระบบ โดยจะเริ่มตั้งแต่การทดสอบประสิทธิ์ภาพของการเคลื่อนที่ของหัวอ่าน จนกระทั่งถึงการติดตามลายเส้นเพื่อการตัดและการบันทึก เพื่อศึกษาการตอบสนองของต่อลายเส้นที่มีผลต่อการติดตามลายเส้นด้วยวิธีการต่างๆ ลายเส้นที่ใช้ทดสอบจะมีความโค้งมากบ้างน้อยบ้าง หรือเป็นมุมฉากบ้าง รวมทั้งมีความหนาของลายเส้นในลักษณะต่างๆ บ้าง เพื่อให้เห็นว่าระบบมีประสิทธิ์ภาพโดยรวมเป็นอย่างไรบ้าง

6.1 การทดสอบความสามารถในการเคลื่อนที่ของระบบขับเคลื่อน

ในการทดสอบจะเป็นการสั่งให้หัวอ่านวาดรูปต่างๆ ที่ความเร็วและความเร่งที่ต่างกัน และมีการวาดซ้ำๆ ที่เดิมหลายครั้งด้วย เพื่อทดสอบประสิทธิ์ภาพของระบบดังหัวข้อต่อไปนี้

6.1.1 ทดสอบความนิ่มนวลของการเคลื่อนที่ เพื่อดูว่ายังมีความเป็นสเต็ปในการเคลื่อนที่ที่เกิดจากความเป็นสเต็ปของสเต็ปปิ้งมอเตอร์ลงเหลืออยู่หรือไม่ สังเกตได้โดยพิจารณาในช่วงที่มีการเคลื่อนที่เป็นเส้นโค้งหรือเส้นตรงที่มีความชัน ซึ่งผลของความเป็นสเต็ปจะมีผลมากที่สุด

6.1.2 ทดสอบความแม่นยำในการเคลื่อนที่ เพื่อดูว่ามีความคลาดเคลื่อนทางตำแหน่งในระหว่างการเคลื่อนที่มากน้อยขนาดไหน สังเกตได้โดยพิจารณาการวาดรูปเดิมซ้ำที่ตำแหน่งเดิมหลายๆครั้งถ้าระบบมีความคลาดเคลื่อนทางตำแหน่งมากก็จะมีการวาดแล้วไม่ทับรอยเดิมให้เห็นได้ชัดเจน

6.1.3 ทดสอบความเร็วในการทำงานของระบบ เพื่อดูว่าระบบมีความเร็วในการประมวลที่สลับซับซ้อนได้ดีเพียงไหน โดยจะมีการเคลื่อนที่เป็นรูปวงกลมที่ความเร็วต่างๆ (การเคลื่อนที่เป็นรูปวงกลมหรือเส้นโค้งใดก็ตาม ระบบจะทำงานหนักหรือใช้เวลาในการคำนวณมากกว่าการเคลื่อนที่ในเส้นตรง

6.1.4 ทดสอบผลของแรงเฉื่อยที่มีผลต่อการเคลื่อนที่ของระบบ เพื่อดูว่าที่ความเร็วต่างๆ แรงเฉื่อยมีผลต่อการเคลื่อนที่ หรือทำให้การเคลื่อนที่ผิดพลาดอย่างไรบ้าง

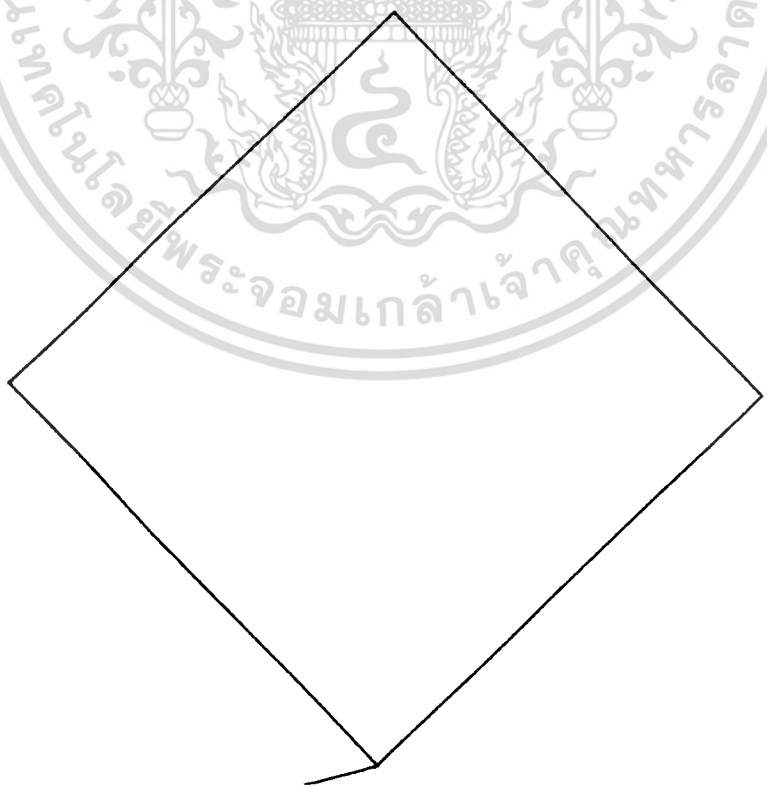
6.1.5 ทดสอบการเคลื่อนที่โดยรวม เป็นสั่งให้ระบบทำการเคลื่อนที่เป็นรูปต่างๆซ้อนกัน เพื่อดูประสิทธิ์ภาพโดยรวม ว่ามีความน่าเชื่อถือหรือไม่

ต่อไปจะเป็นการแสดงวิธีการทดสอบและผลที่ได้จากการทดสอบ

6.1.1 ทดสอบความนิ่มนวลของการเคลื่อนที่ เนื่องจากสเต็ปปิ้งมอเตอร์ทั่วไปเมื่อหมุนด้วยความเร็วต่ำๆ จะให้ผลของการเคลื่อนที่ที่มีลักษณะเป็นสเต็ปที่ชัดเจนมากกว่าการหมุนด้วยความเร็วสูง ดังจึงได้ทดสอบโดยมีการเขียนโปรแกรมสั่งให้ระบบทำการเคลื่อนที่เป็นรูปวงกลม และรูปสี่เหลี่ยมขนมเปียกปูน ที่ความเร็ว 1 เซนติเมตรต่อวินาที และอัตราเร่ง 3 เซนติเมตรต่อวินาที^{2}

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 6.1 แสดงการเคลื่อนที่เป็นรูปวงกลมที่ความเร็ว 1 cm./sec. และอัตราเร่ง 3 cm./sec.^{2}

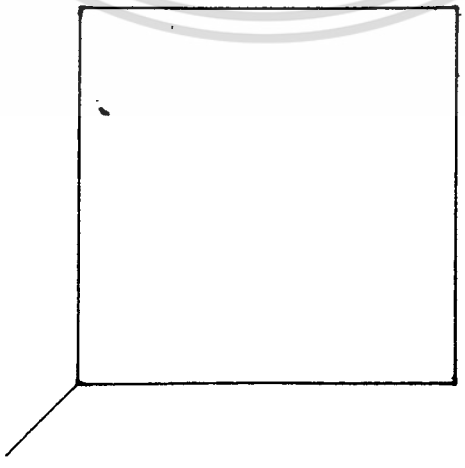


รูปที่ 6.2 แสดงการเคลื่อนที่เป็นรูปสี่เหลี่ยมขนมเปียกปูน
ที่ความเร็ว 1 cm./sec. และอัตราเร่ง 3 cm./sec.^{2}

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

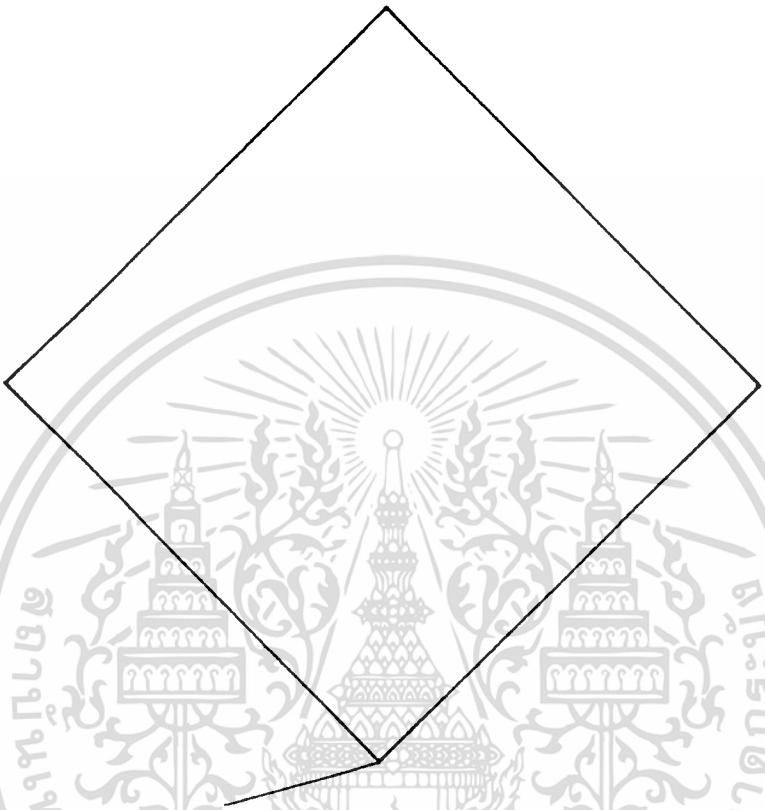
6.1.2 ทดสอบความแม่นยำในการเคลื่อนที่ ทำการทดสอบโดยการเขียนโปรแกรมสั่งให้ระบบทำการวาดรูปวงกลมและรูปสี่เหลี่ยมซ้ำที่เดิมจำนวน 10 ครั้ง ที่ความเร็ว 300 เซนติเมตรต่อนาที

รูปที่ 6.3 แสดงการเคลื่อนที่เป็นรูปวงกลม 10 รอบ ที่ความเร็ว 300 เซนติเมตรต่อนาที



รูปที่ 6.4 แสดงการเคลื่อนที่เป็นรูปสี่เหลี่ยม 10 รอบ ที่ความเร็ว 300 เซนติเมตรต่อนาที

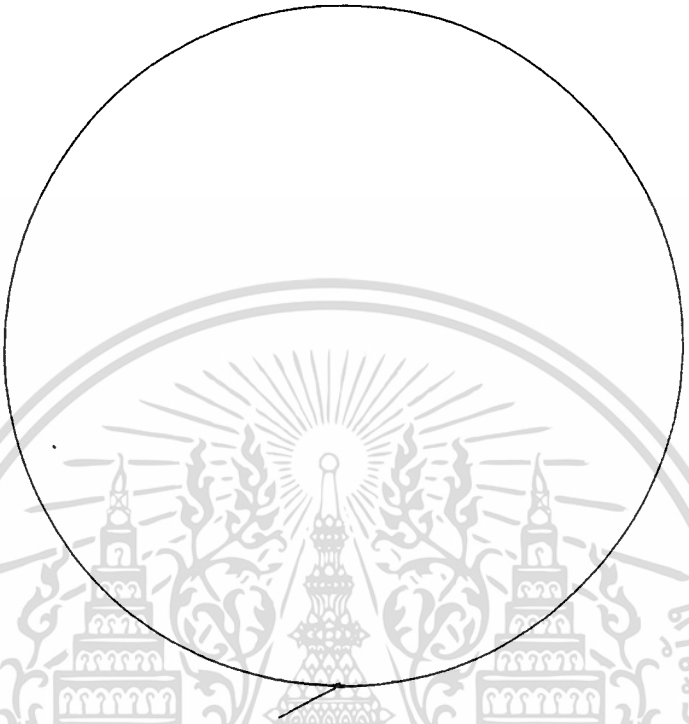
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 6.5 แสดงการเคลื่อนที่เป็นรูปขนมเปียกปูน 10 รอบ ที่ความเร็ว 300 เซนติเมตรต่อนาที

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

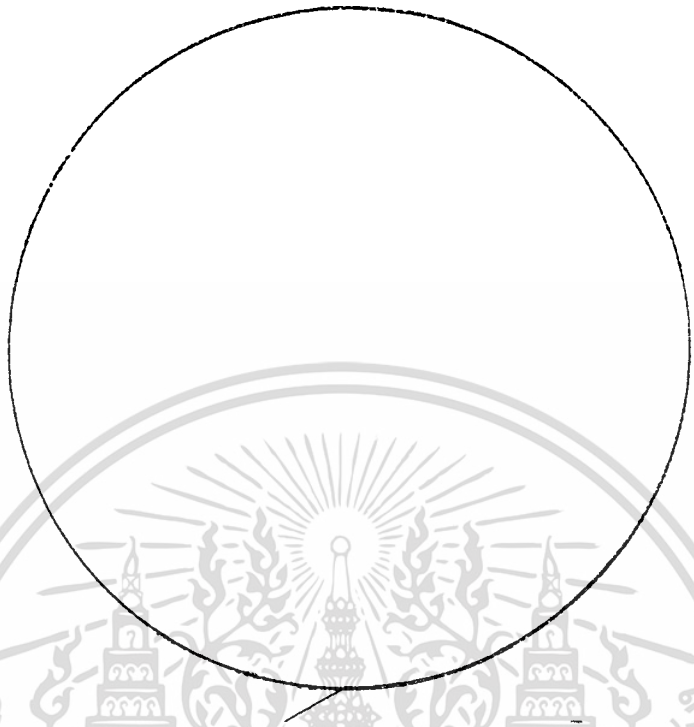
6.1.3 ทดสอบความเร็วในการทำงานของระบบ ทำการทดสอบโดยการเขียนโปรแกรมสั่งให้ระบบทำการวิ่งเป็นวงกลม 10 รอบที่ความเร็วค่าต่าง ๆ ที่อัตราเร่ง 100 m.m./sec.^{2} ดังต่อไปนี้



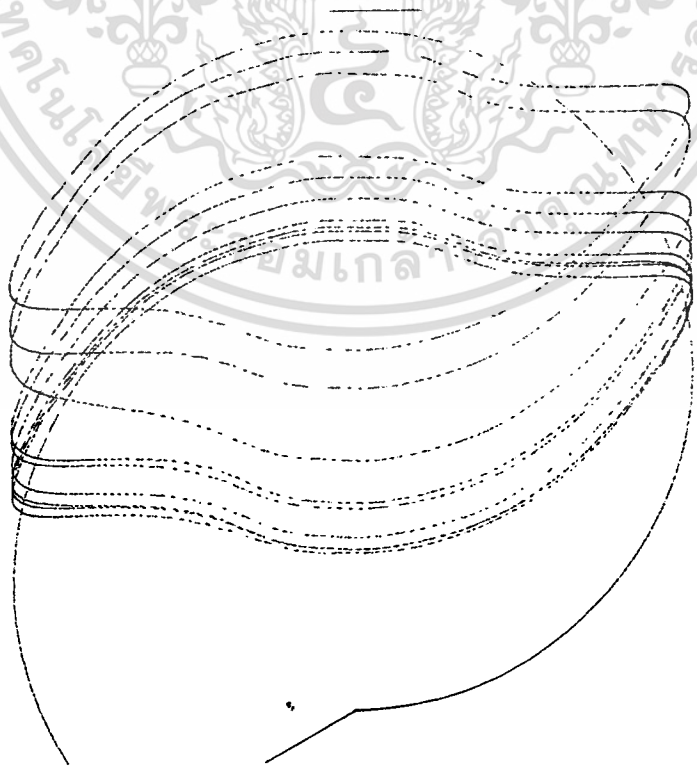
รูปที่ 6.6 แสดงการเคลื่อนที่เป็นวงกลมที่ความเร็ว 150 m.m./sec อัตราเร่ง 100 m.m./sec.^{2}

รูปที่ 6.7 แสดงการเคลื่อนที่เป็นวงกลมที่ความเร็ว 200 m.m./sec อัตราเร่ง 100 m.m./sec.^{2}

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



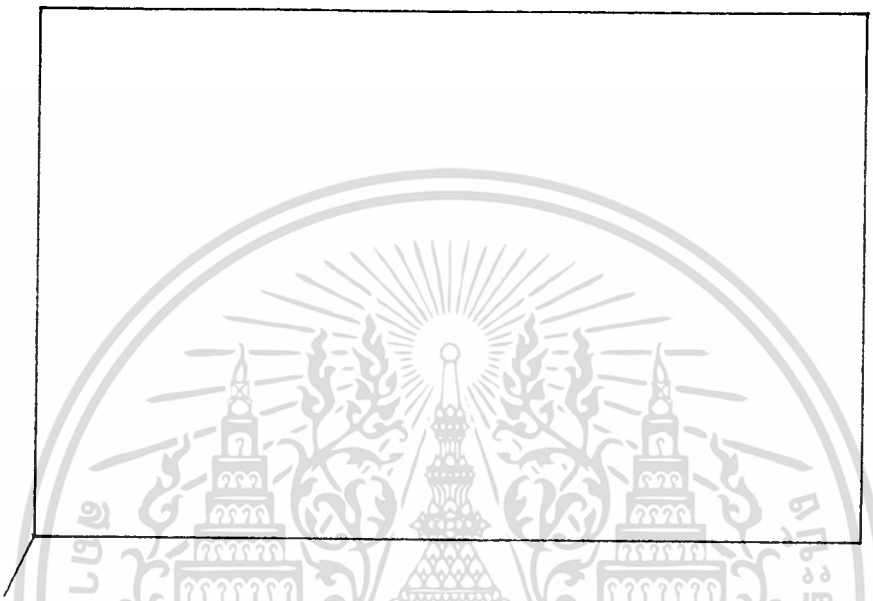
รูปที่ 6.8 แสดงการเคลื่อนที่เป็นวงกลมที่ความเร็ว 220 m.m./sec อัตราเร่ง 100 m.m./sec.^{2}



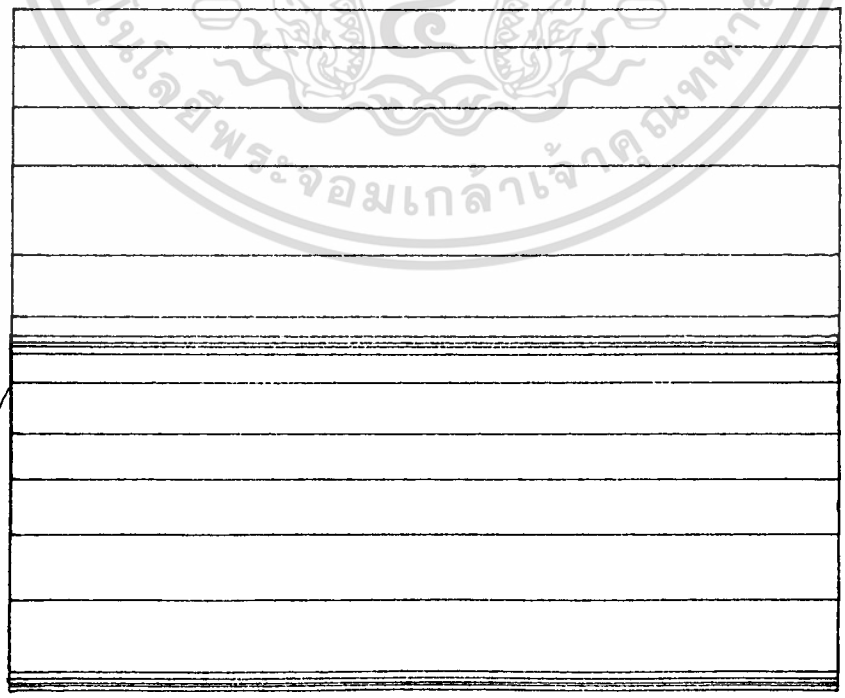
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 6.9 แสดงการเคลื่อนที่เป็นวงกลมที่ความเร็ว 250 m.m./sec อัตราเร่ง 100 m.m./sec.^{2}

6.1.4 ทดสอบผลของแรงเฉื่อยที่มีผลต่อการเคลื่อนที่ของระบบ ทำการทดสอบโดยการเขียนโปรแกรมสั่งให้ระบบทำการวิ่งเป็นรูปสี่เหลี่ยมที่ความเร่งค่าต่าง ๆ ที่ความเร็ว 200 m.m./sec

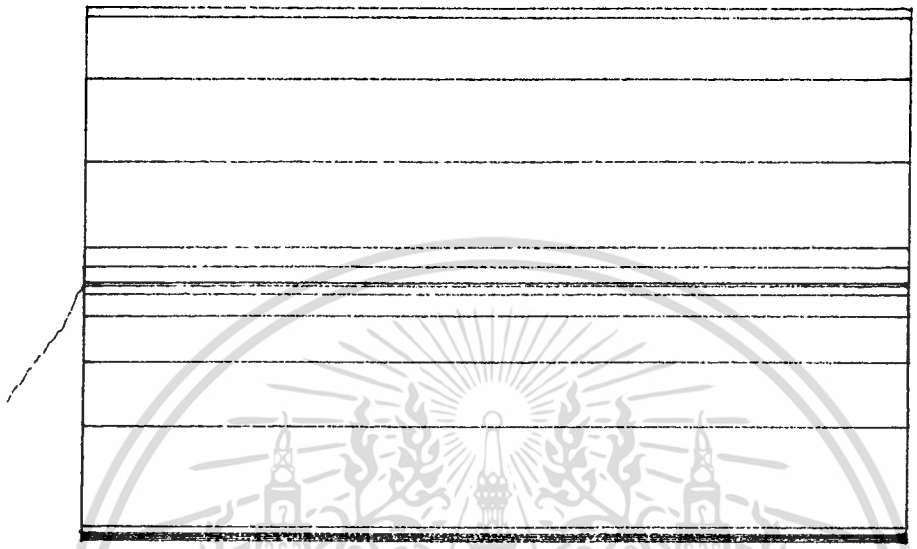


รูปที่ 6.10 แสดงการเคลื่อนที่เป็นรูปสี่เหลี่ยมที่ความเร็ว 200 m.m./sec อัตราเร่ง 500 m.m./sec^2

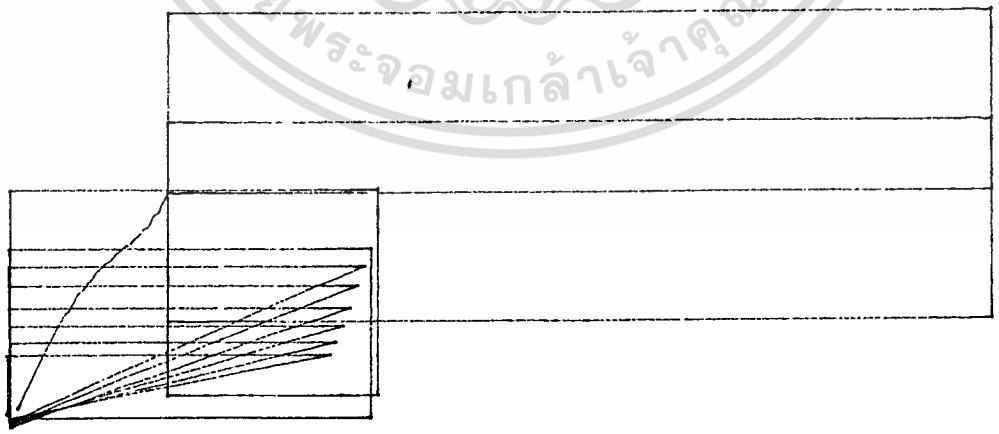


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 6.11 แสดงการเคลื่อนที่เป็นรูปสี่เหลี่ยมที่ความเร็ว 200 m.m./sec อัตราเร่ง 800 m.m./sec^2

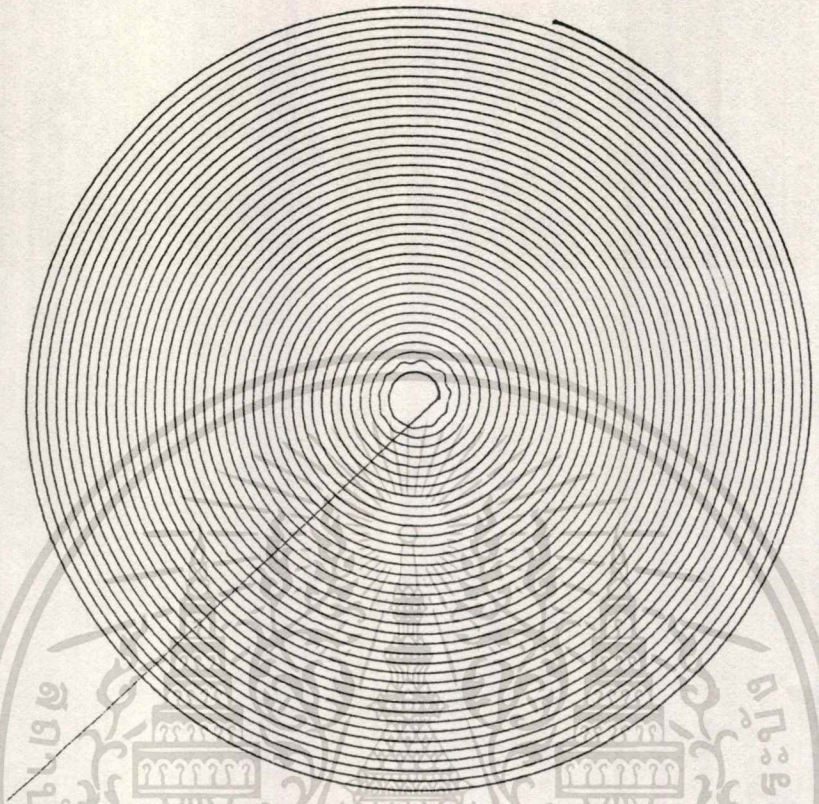


รูปที่ 6.12 แสดงการเคลื่อนที่เป็นรูปสี่เหลี่ยมที่ความเร็ว 200 m.m./sec อัตราเร่ง 1200 m.m./sec^2



รูปที่ 6.13 แสดงการเคลื่อนที่เป็นรูปสี่เหลี่ยมที่ความเร็ว 200 m.m./secอัตราเร่ง 1500 m.m./sec^2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



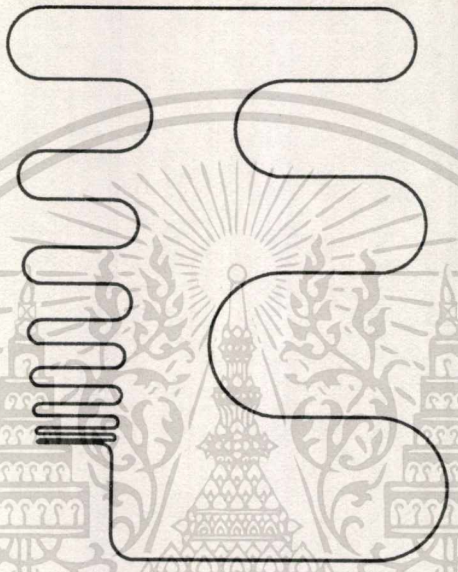
รูปที่ 6.16 แสดงการเคลื่อนที่เป็นรูปวงกันหอยที่ความเร็ว 100 m.m./sec และอัตราเร่ง 100 m.m./sec^2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6.2 การทดสอบความสามารถในการติดตามลายเส้นสำหรับการควบคุมแบบต่าง ๆ

เป็นการติดตามลายเส้นโดยให้มีการวาดลายเส้นไปพร้อมๆกัน เพื่อดูว่าผลการติดตามลายเส้นในวิธีไหนให้ผลดีที่สุดต่อลายเส้นรูปไหนบ้าง และดูความเร็วในการติดตามลายเส้นของวิธีการต่างๆ ว่ามีความเหมาะสมที่ความเร็วในช่วงไหนบ้าง

ต้นแบบที่ใช้ในการทดสอบในขั้นแรกเป็นรูปที่มีส่วนของวงกลมที่มีรัศมีต่าง ๆ ดังรูปที่ 6.17

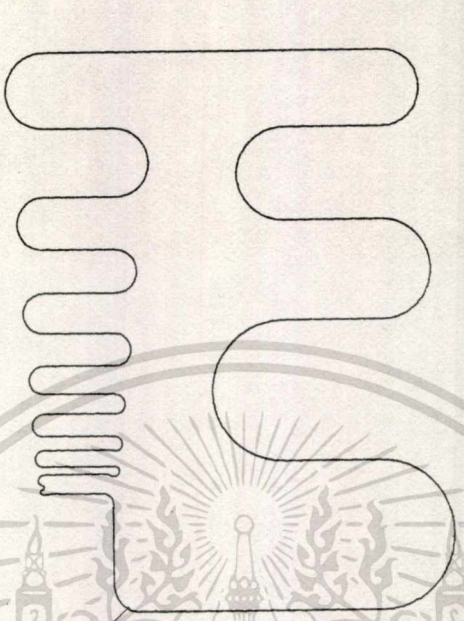


รูปที่ 6.17 แสดงรูปที่ใช้ทดสอบทางด้านเส้นตรงและเส้นโค้ง

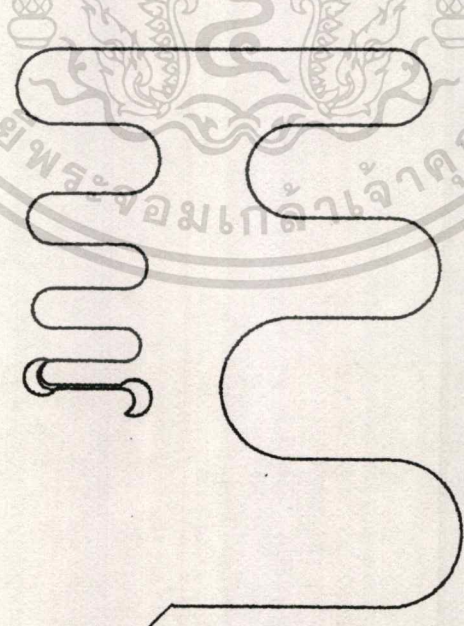
จากรูปที่ใช้ทดสอบนี้จะทำให้เรารู้ว่าวิธีการติดตามลายเส้นในแต่ละแบบให้ผลตอบสนองต่อลายเส้นอย่างไรบ้าง โดยจะทดสอบที่ความเร็ว 100 , 300 , 500 และ 700 m.m./min ตามลำดับ

6.2.1 การทดสอบการติดตามลายเส้นระบบ P I D

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

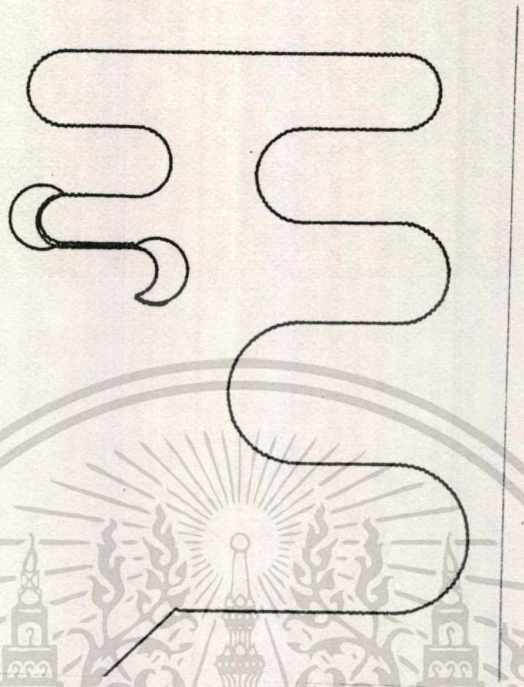


รูปที่ 6.18 ผลการติดตามลายเส้นที่ความเร็ว 100 m.m./min

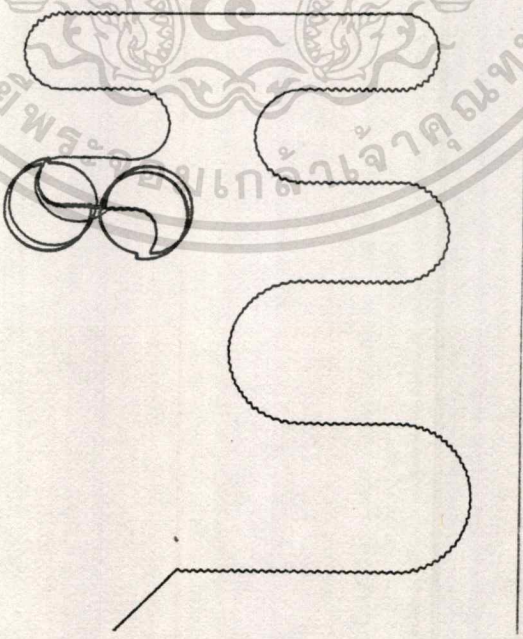


รูปที่ 6.19 ผลการติดตามลายเส้นที่ความเร็ว 300 m.m./min

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



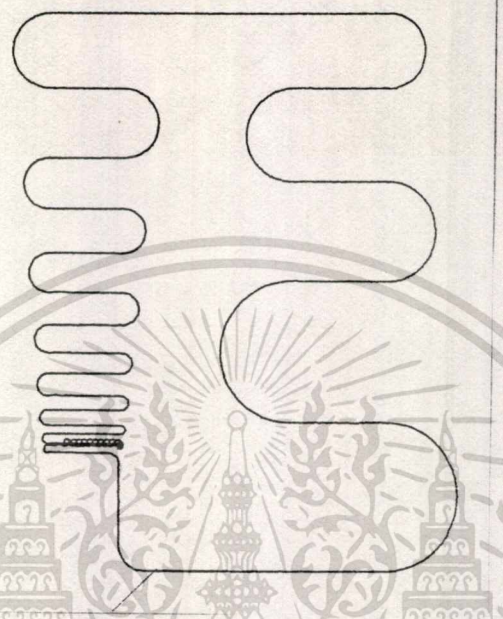
รูปที่ 6.20 ผลการติดตามลายเส้นที่ความเร็ว 500 m.m./min



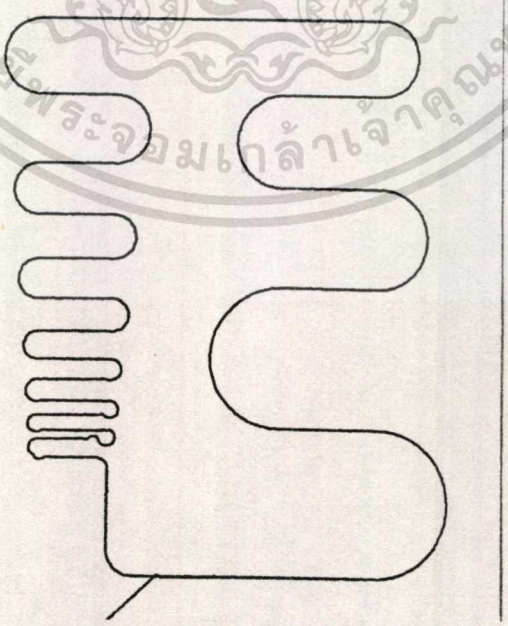
รูปที่ 6.21 ผลการติดตามลายเส้นที่ความเร็ว 700 m.m./min

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6.2.2 การทดสอบการติดตามลายเส้นระบบ P ร่วมกับการตัดสินใจ

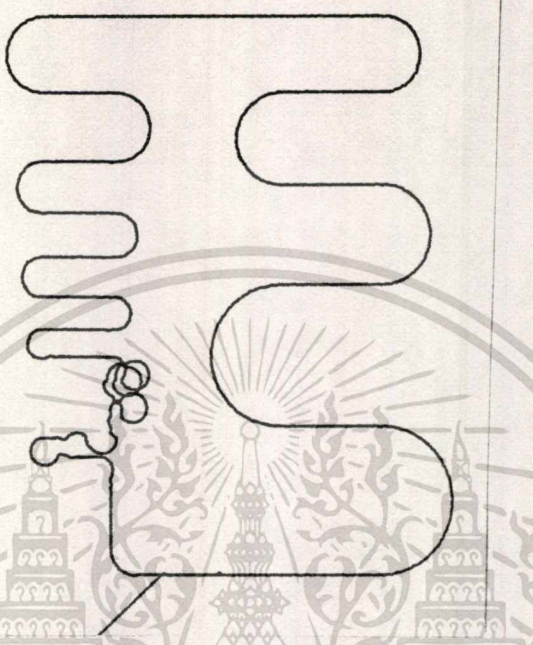


รูปที่ 6.22 ผลการติดตามลายเส้นที่ความเร็ว 100 m.m./min

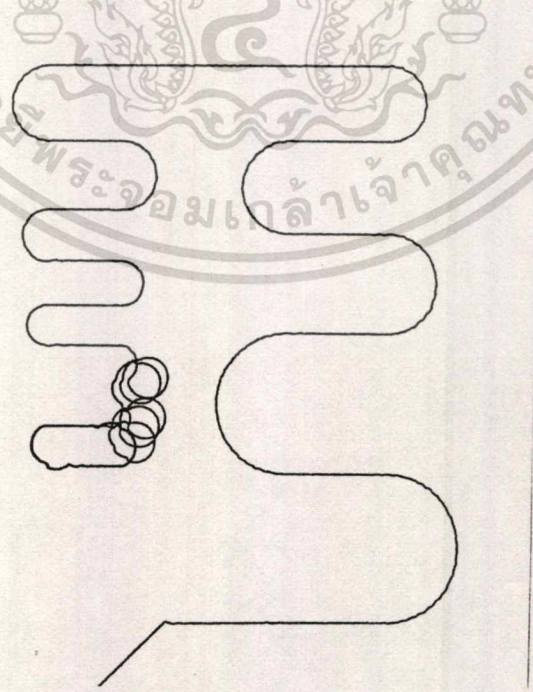


รูปที่ 6.23 ผลการติดตามลายเส้นที่ความเร็ว 300 m.m./min

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



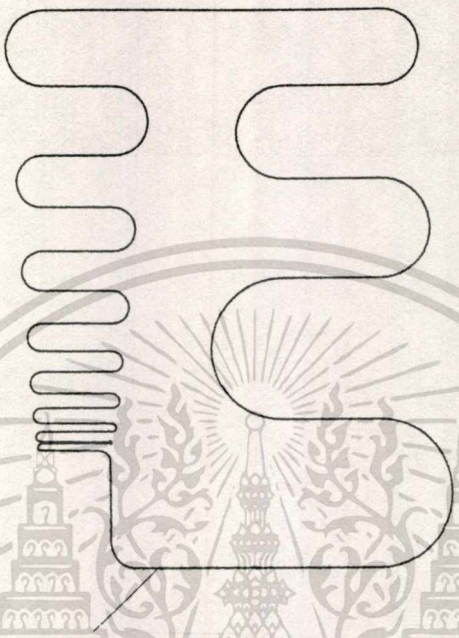
รูปที่ 6.24 ผลการติดตามลายเส้นที่ความเร็ว 500 m.m./min



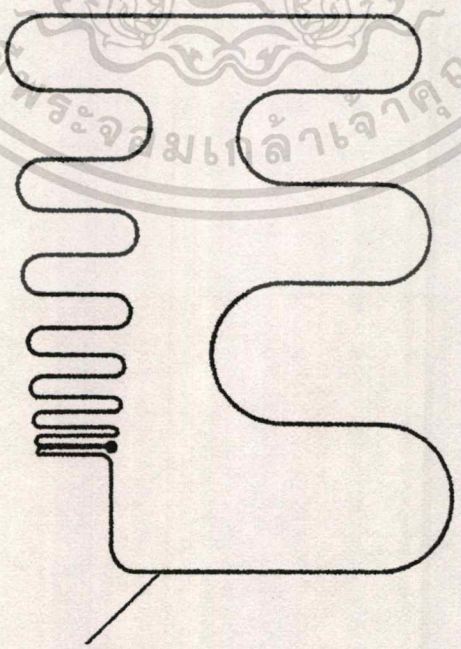
รูปที่ 6.25 ผลการติดตามลายเส้นที่ความเร็ว 700 m.m./min

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6.2.3 การทดสอบการติดตามลายเส้นระบบสะท้อนแสงร่วมกับการตัดสินใจ

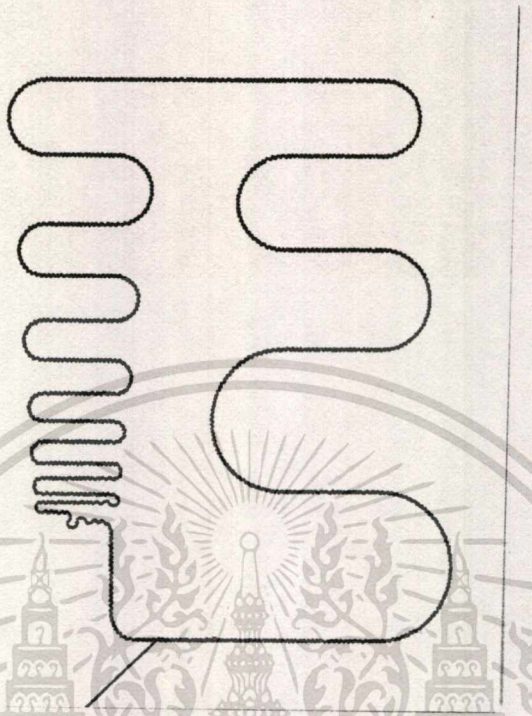


รูปที่ 6.26 ผลการติดตามลายเส้นที่ความเร็ว 100 m.m./min

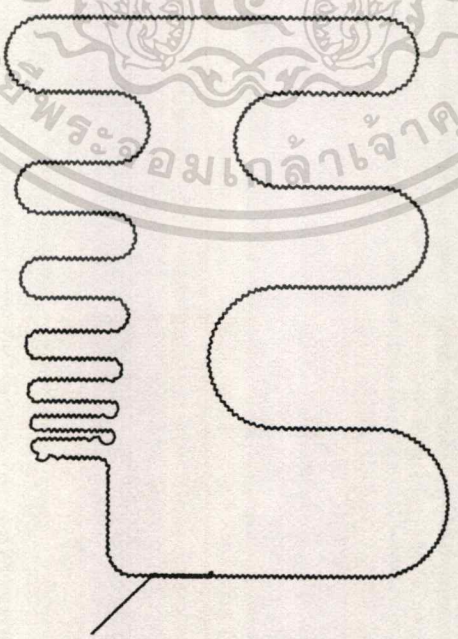


รูปที่ 6.27 ผลการติดตามลายเส้นที่ความเร็ว 300 m.m./min

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 6.28 ผลการติดตามลายเส้นที่ความเร็ว 500 m.m./min

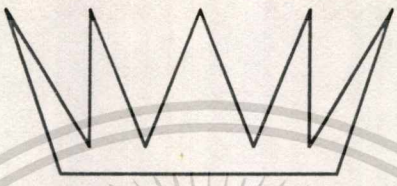


รูปที่ 6.29 ผลการติดตามลายเส้นที่ความเร็ว 700 m.m./min

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

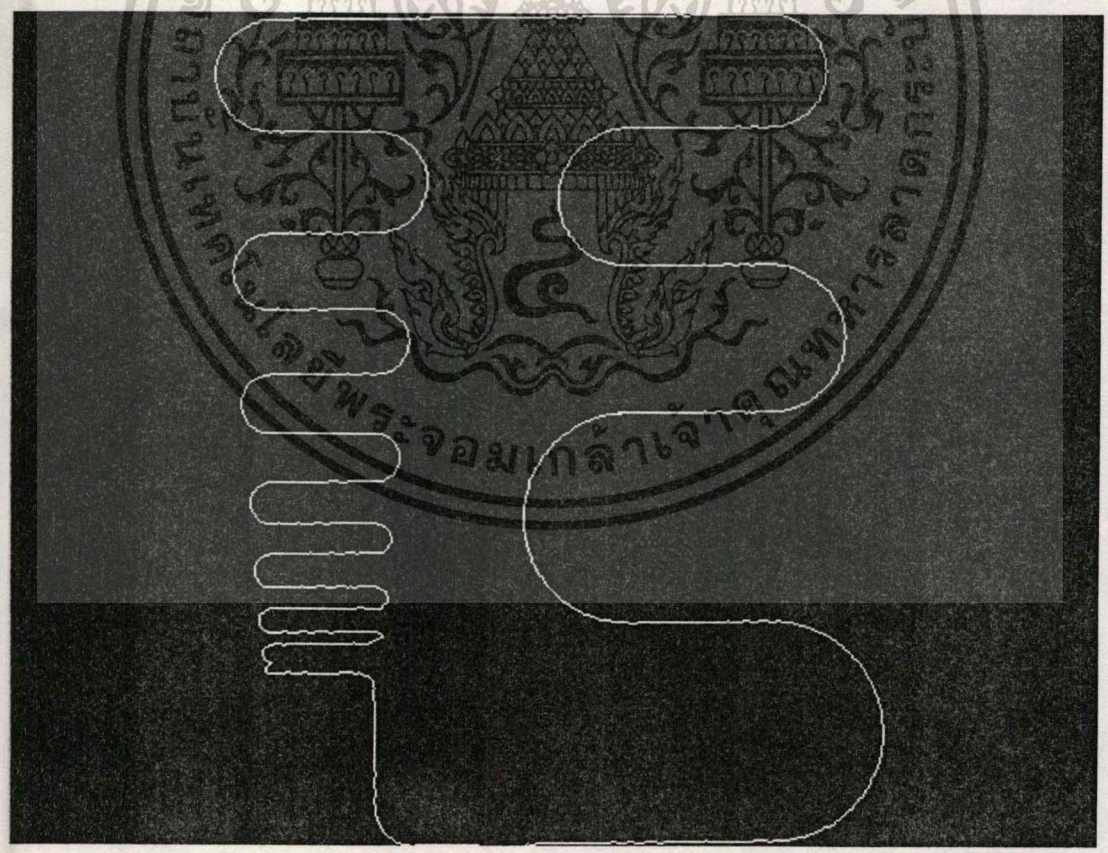
6.3 การทดสอบความสามารถในการบันทึกข้อมูลที่ได้จากการติดตามลายเส้นด้วยวิธีการต่างๆ

ในการทดสอบนี้ทำการทดสอบที่ความเร็ว 100 m.m./min ซึ่งเป็นความเร็วเหมาะสมที่สุดสำหรับการบันทึกข้อมูล สำหรับตัวอย่างที่ใช้จะมี 2 แบบ แบบแรกเป็นรูปเดียวกับที่สำหรับการติดตามลายเส้น ส่วนแบบที่ 2 จะเป็นรูปมงกุฎดังรูปที่ 6.30



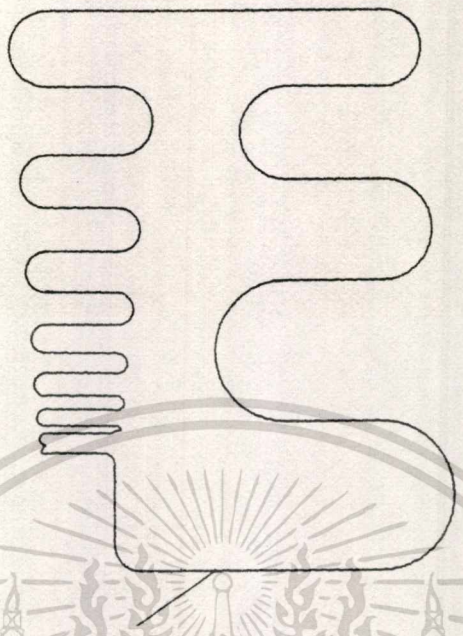
รูปที่ 6.30 แสดงรูปมงกุฎที่ใช้ทดสอบ

6.3.1 ผลการทดสอบการบันทึกลายเส้นระบบ P I D

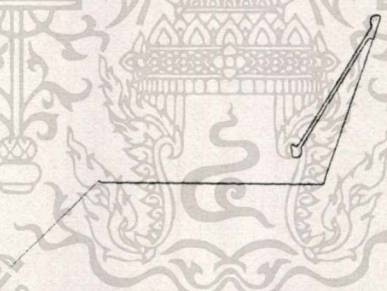


รูปที่ 6.31 ข้อมูลบนจอภาพที่ได้จากการบันทึกตัวอย่างแรก

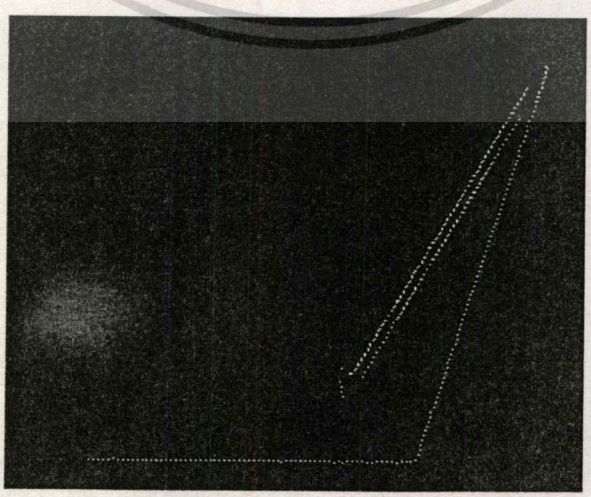
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 6.32 ผลการวาดโดยใช้ข้อมูลจากการบันทึกตัวอย่างแรก ที่ความเร็ว 1000 m.m./min.



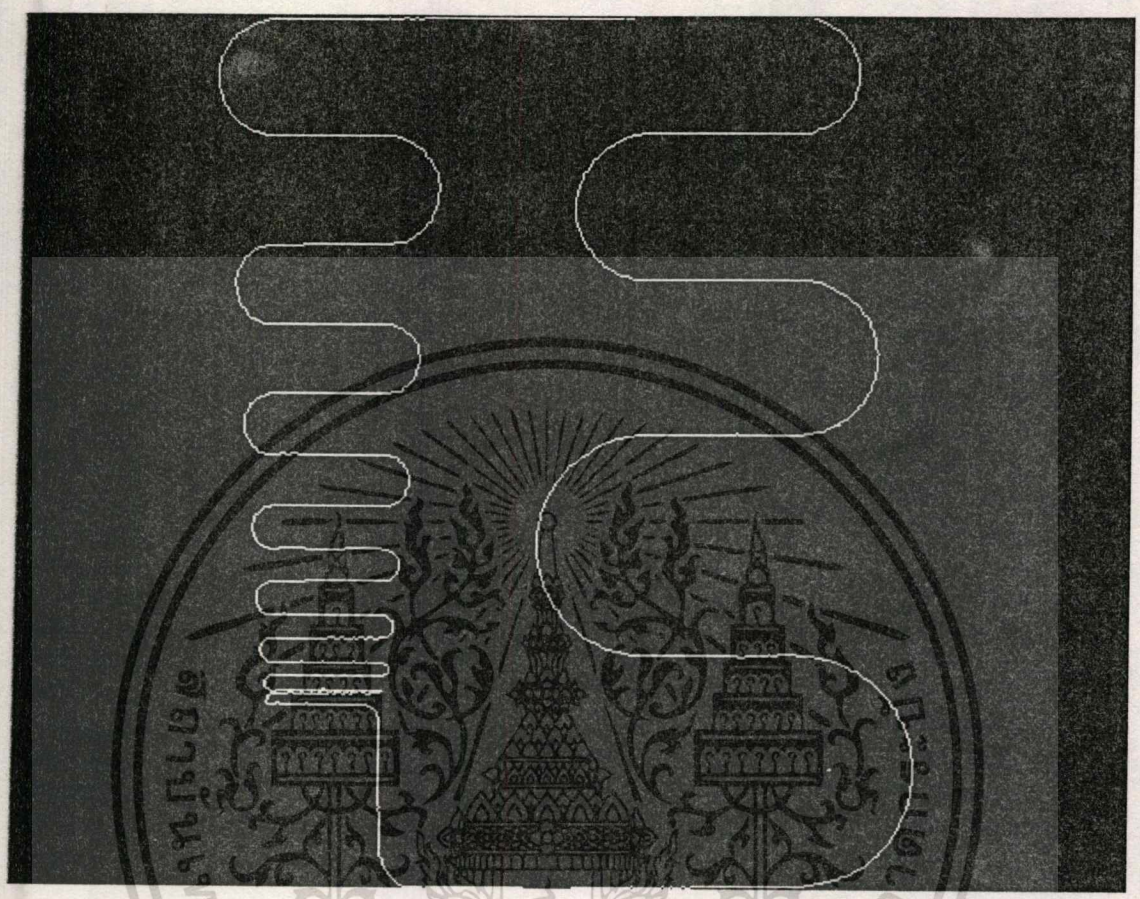
รูปที่ 6.33 ผลของการตามลายเส้นจากตัวอย่างที่ 2



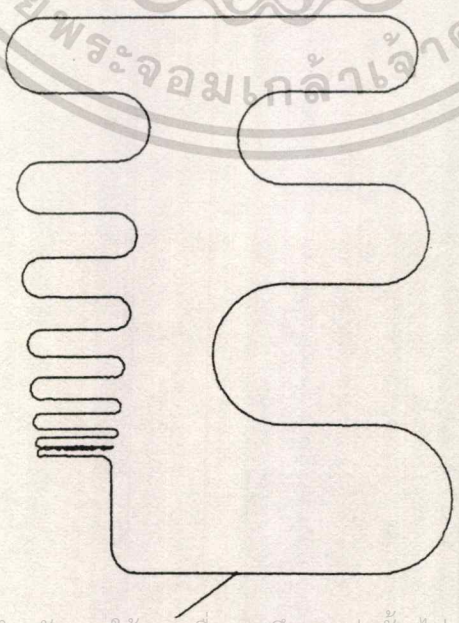
รูปที่ 6.34 ข้อมูลบนจอภาพที่ได้จากการบันทึกตัวอย่างที่ 2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเอกสารทุกครั้งที่มีการนำไปใช้

6.3.2 ผลการทดสอบการบันทึกลายเส้นระบบ P ร่วมกับการตัดสินใจ

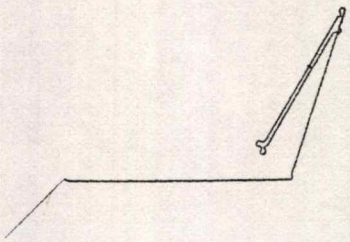


รูปที่ 6.35 ข้อมูลบนจอภาพที่ได้จากการบันทึกตัวอย่างแรก

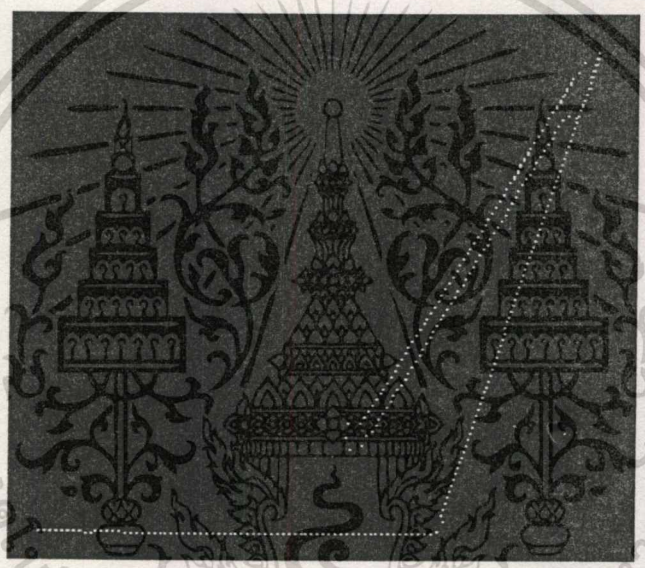


รูปที่ 6.36 ผลการวาดโดยใช้ข้อมูลจากการบันทึกตัวอย่างแรกที่ความเร็ว 1000 m.m./min.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเอกสารทุกครั้งที่มีการนำไปใช้



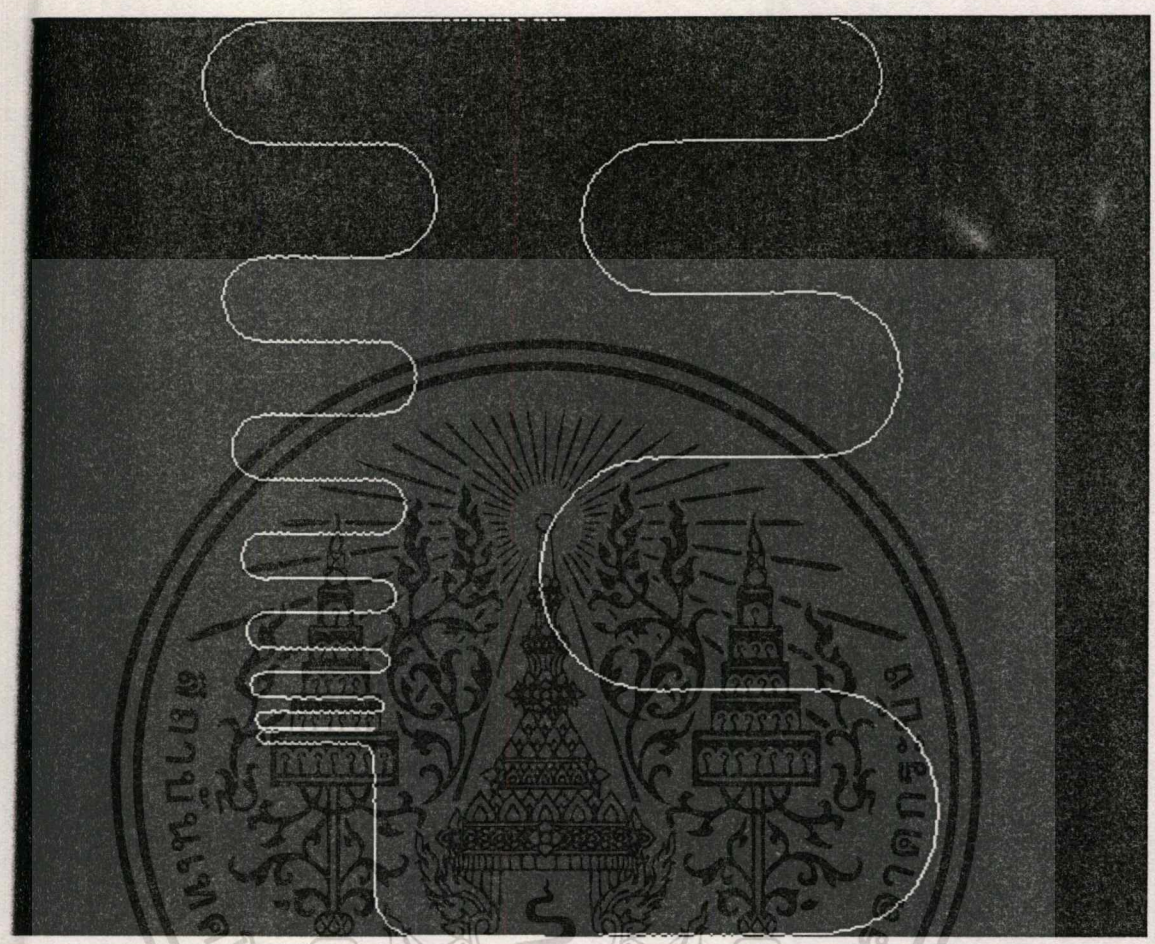
รูปที่ 6.37 ผลของการตามลายเส้นจากตัวอย่างที่ 2



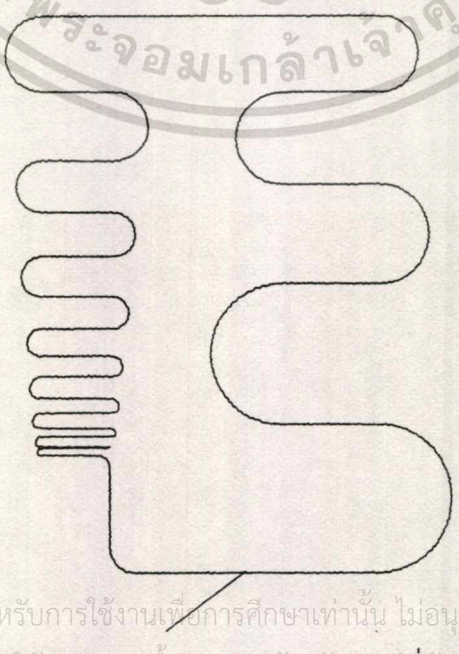
รูปที่ 6.38 ข้อมูลบนจอภาพที่ได้จากการบันทึกตัวอย่างที่ 2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6.3.3 ผลการทดสอบการบันทึกลายเส้นระบบสะท้อนแสงร่วมกับการตัดสินใจ

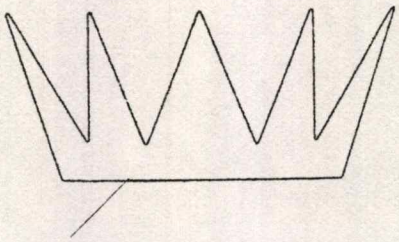


รูปที่ 6.39 ข้อมูลบนจอภาพที่ได้จากการบันทึกตัวอย่างแรก

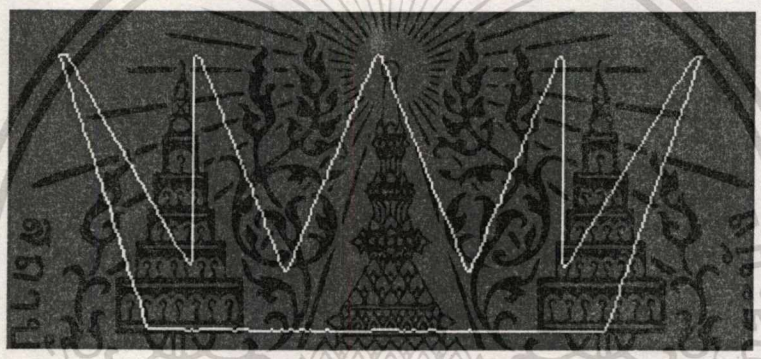


รูปที่ 6.40 ผลการวาดโดยใช้ข้อมูลจากการบันทึกตัวอย่างแรกที่ความเร็ว 1000 m.m./min.

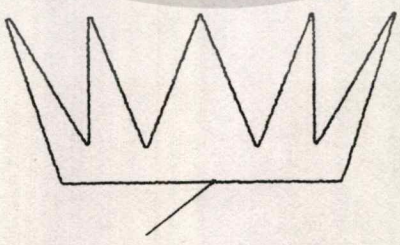
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารนี้ทุกครั้งที่มีการนำไปใช้



รูปที่ 6.41 ผลของการตามลายเส้นจากตัวอย่างที่ 2



รูปที่ 6.42 ข้อมูลบนจอภาพที่ได้จากการบันทึกตัวอย่างที่ 2

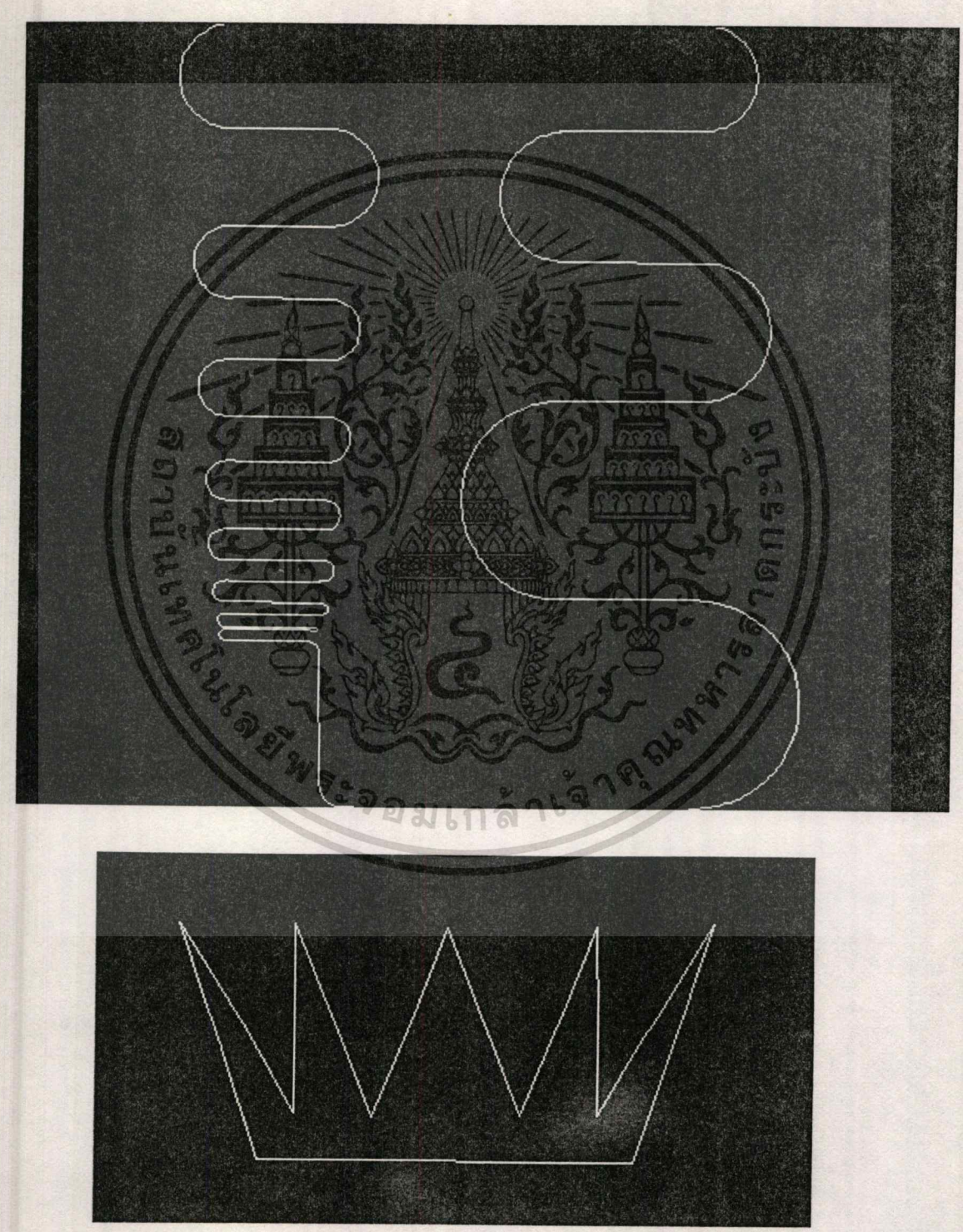


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 6.40 ผลการวาดโดยใช้ข้อมูลจากการบันทึกตัวอย่างที่ 2 ที่ความเร็ว 1000 m.m./min.

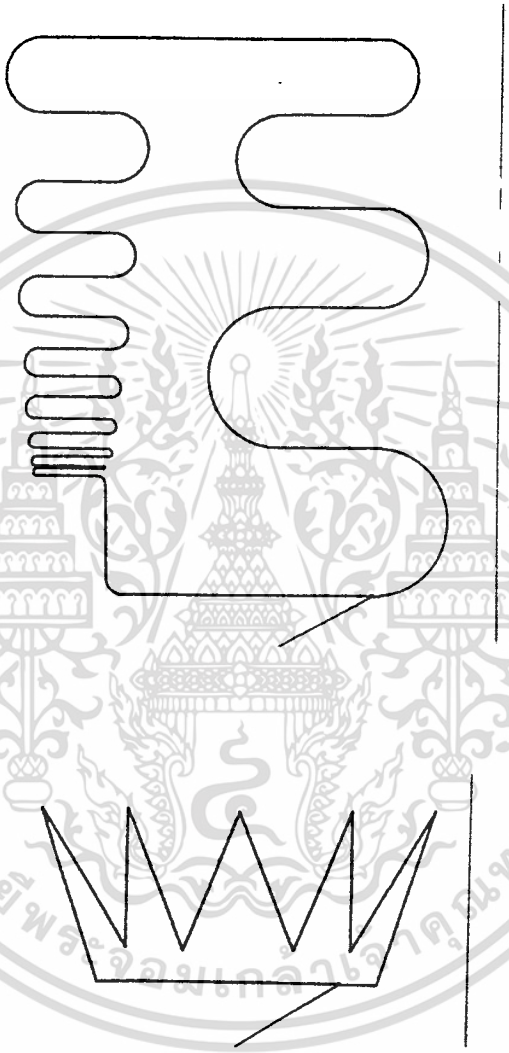
6.4 การทดสอบโปรแกรมแก้ไขข้อมูล

ในการทดสอบนี้จำใช้ข้อมูลที่ดีที่สุดจากบันทึกลายเส้น ซึ่งจะเห็นว่าการบันทึกแบบสะท้อนแสงร่วมกับการตัดสินใจนั้นให้รายละเอียดในบันทึกที่ดีที่สุด ซึ่งหลังจากแก้ไขแล้วได้ผลดังต่อไปนี้



รูปที่ 6.41 ผลบนจอที่ได้หลังจากการแก้ไขแล้ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 6.42 **ผลการวาดโดยข้อมูลที่แก้ไขแล้ว ที่ความเร็ว** 1000 m.m./min.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6.5 สรุปผลการทดลอง

จากทดลองที่ผ่านสรุปการทำงานของระบบขับเคลื่อนได้ว่า

1. ระบบขับเคลื่อนมีความนิ่มนวลการเคลื่อนที่ดีมาก

2. ระบบขับเคลื่อนมีความแม่นยำในการเคลื่อนที่ที่ดีมาก

3. ความเร็วในการทำงานของคอมพิวเตอร์สูงกว่าความสามารถในการตอบสนองการเคลื่อนที่ของระบบขับเคลื่อน

4. แรงเฉื่อยของระบบขับเคลื่อนจะเริ่มมีผลต่อการเคลื่อนที่ เมื่อความเร่งเกิน 500 mm./sec^2

5. ระบบขับเคลื่อนสามารถทำการวาดเป็นรูปต่างๆ ได้เป็นอย่างดี รูปที่ได้มีความถูกต้องทางตำแหน่งที่ดี

สำหรับการทดลองทางด้านการติดตามลายเส้นและการบันทึกลายเส้นสรุปเป็นตารางได้ว่า

การทดสอบ ด้วยวิธีการต่าง ๆ	ขนาดเส้นผ่าศูนย์กลางเล็กสุดที่ยังใช้งานได้ดีที่ความเร็วต่าง ๆ			
	100 mm/min	300 mm/min	500 mm/min	700 mm/min
ระบบ P I D	∅ 1.5 mm	∅ 5 mm	∅ 8.5 mm	∅ 12 mm
ระบบ P ร่วมกับ การตัดสินใจ	∅ 1 mm	∅ 2 mm	∅ 4 mm	∅ 5.5 mm
ระบบสะท้อนแสง กับการตัดสินใจ	∅ 0.1 mm	∅ 1 mm	∅ 1.5 mm	∅ 2 mm

ตารางที่ 6.1 สรุปขนาดเส้นผ่าศูนย์กลางของส่วนโค้งที่สามารถติดตามลายเส้นได้ที่ความเร็วต่าง ๆ

การทดสอบ ด้วยวิธีการต่าง ๆ	ผลการสั่นที่เกิดขึ้นที่ความเร็วต่าง ๆ			
	100 mm/min	300 mm/min	500 mm/min	700 mm/min
ระบบ P I D	น้อยมาก	น้อย	ปานกลาง	มาก
ระบบ P ร่วมกับ การตัดสินใจ	น้อยมาก	น้อย	น้อย	ปานกลาง
ระบบสะท้อนแสง กับการตัดสินใจ	น้อย	ปานกลาง	มาก	มาก

ตารางที่ 6.2 แสดงผลการสั่นที่เกิดขึ้นที่ความเร็วต่าง ๆ

สำหรับผลที่ได้หลังจากการแก้ไขแล้ว ให้ผลการเคลื่อนที่ที่ดีกว่าเดิม

ดังนั้นจึงสามารถสรุปได้ว่า การใช้งานระบบที่ง่ายที่สุด คือการปล่อยให้ระบบทำการติดตามลายเส้นที่ความเร็ว 100 m.m./min ด้วยระบบสะท้อนแสงร่วมกับการตัดสินใจ หลังจากนั้นก็สั่งให้ระบบทำการตัดชิ้นงานโดยใช้ข้อมูลที่ได้บันทึกไว้แล้วที่ความเร็วที่ต้องการต่อไป

เอกสารนี้เป็นเอกสารสงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 7 บทสรุป

7.1 สรุปผลการวิจัย

ระบบการขับเคลื่อนและติดตามลายเส้นนี้สรุปได้ว่ามีประสิทธิ์ที่ดีมาก เพราะสามารถติดตามลายเส้นได้อย่างละเอียด ทั้งนี้ก็เนื่องมาจากมีหัวอ่านที่มีขนาดเล็กมาก ผลความผิดพลาดที่ได้จากการติดตามลายเส้นถือได้ว่ามีน้อยมากเมื่อเทียบกับขนาดของรูที่เกิดจากการตัดแผ่นเหล็กด้วยแก๊ส

ระบบการขับเคลื่อนที่สร้างขึ้นนี้ เป็นระบบที่ยังไม่สมบูรณ์อยู่อย่างเดียวคือระบบควบคุมการตัด ซึ่งยังไม่ได้ติดตั้งระบบตัดให้แก่ระบบดังนั้นจึงยังใช้งานจริงไม่ได้ในตอนนี้ อย่างไรก็ตามระบบนี้ก็มีอนาคตที่จะสามารถพัฒนาต่อไปได้เป็นอย่างดี

ในการพัฒนาเพื่อให้เป็นสินค้านั้นสามารถทำได้ เพราะเมื่อพิจารณาถึงอุปกรณ์ที่ใช้จะเห็นว่ามีน้อยมากๆ ซึ่งจากการประเมินระบบคร่าวๆ โดยไม่รวมราคาคอมพิวเตอร์ ก็จะประมาณได้ว่า 4000 ถึง 8000 บาท ซึ่งขึ้นอยู่กับแหล่งจำหน่ายอุปกรณ์ สำหรับระบบขับเคลื่อนนี้เท่าที่ทดลองการรับน้ำหนักดูพบว่าสามารถรองรับน้ำหนักได้อย่างน้อย 2 กิโลกรัม ดังนั้นในการพัฒนาเพื่อเป็นสินค้าจริงอาจจะต้องปรับปรุงทางด้านโครงสร้างให้แข็งแรงมากขึ้น

สำหรับการพัฒนาต่อไปนั้น

1.ควรจะพัฒนาทางด้านการเขียนแบบบนคอมพิวเตอร์เพื่อให้ระบบมีความมาตราฐานสูงขึ้น

2.ควรจะพัฒนาทางด้านการรวมข้อมูลที่ให้สามารถนำมาเชื่อมกันได้ เพื่อให้รองรับชิ้นงานที่ซับซ้อนได้

3.ควรจะพัฒนาระบบให้สามารถใช้มอเตอร์กระแสสลับธรรมดาได้ เพื่อให้ระบบมีความยืดหยุ่นในการผลิตเพราะสามารถหาได้ง่ายกว่าสเต็ปปิ้งมอเตอร์ แต่ถ้าระบบมีอะไหล่ที่หาง่ายมากเกินไปอาจถูกลอกเลียนแบบได้ง่ายเช่นกัน

4.ควรเพิ่มแกนการเคลื่อนที่ในแนวดิ่งเพื่อการพัฒนาเป็นระบบกัดชิ้นงานต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เอกสารอ้างอิง

[1] ยืน ภู่วรรณ,เทคนิคการประยุกต์และใช้งานไอซีทีทีแอล, บริษัท ซีเอ็ดยูเคชั่น จำกัด, พิมพ์ครั้งที่ 4 พ.ค. 2528 , หน้า 113 - 114

[2] SCIENCE,ENGINEERING & EDUCATION CO.,LTD.,THE WORLD TTL IC DATA & CROSS_REFERENCE GUTDE , พิมพ์ครั้งที่ 2 พ.ค. 2527

[3] ดร.บรรจบ ยศสมบัติ ,วงจรขับสเตปปิ้งมอเตอร์คู่ , วารสาร คอมพิวเตอร์ อิเลคทรอนิกส์ เวิลด์ ฉบับที่ 100 ประจำเดือน มีนาคม-พฤษาคม, หน้า 141-147

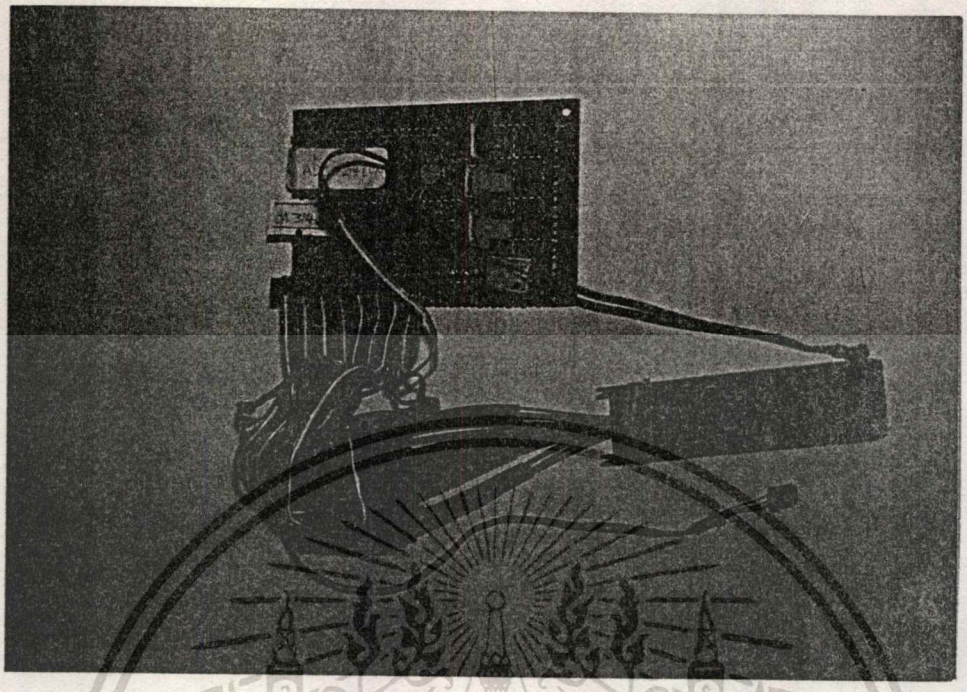
[4] BENJAMIN C.KUO," THEORY AND APPLICATIONS OF STEP MOTORS ", WEST PUBLISHING CO. 1974

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

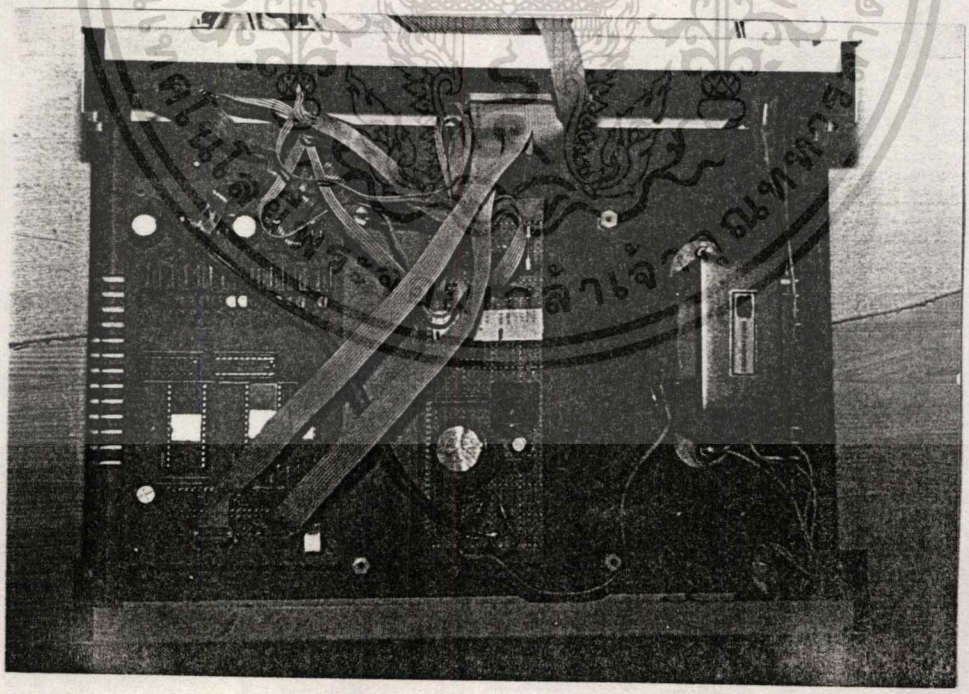
ภาคผนวก ก.

ชิ้นงานของวิทยานิพนธ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

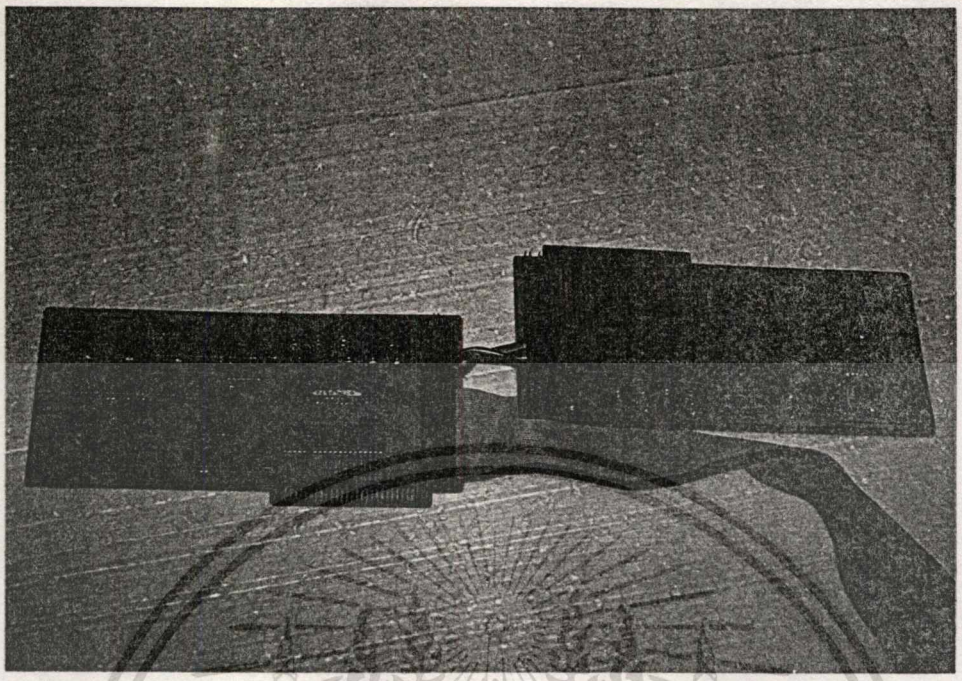


รูปที่ 7.1 หัวอ่านแบบเมกตริกซ์ 4 * 4



รูปที่ 7.2 การติดตั้งวงจรขับเคลื่อนสเต็ปปิ้งมอเตอร์

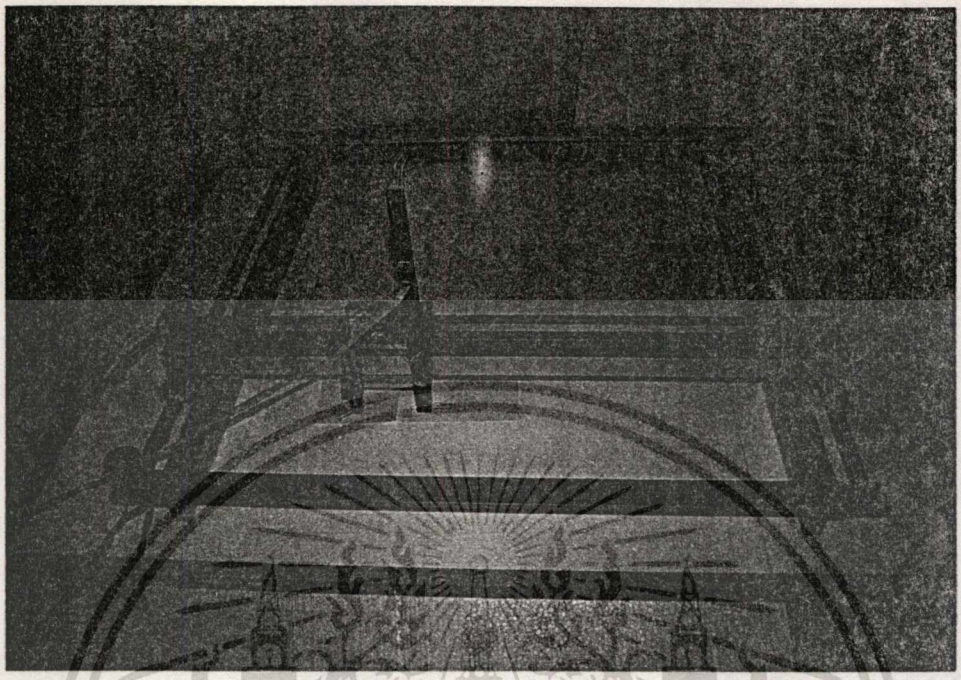
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



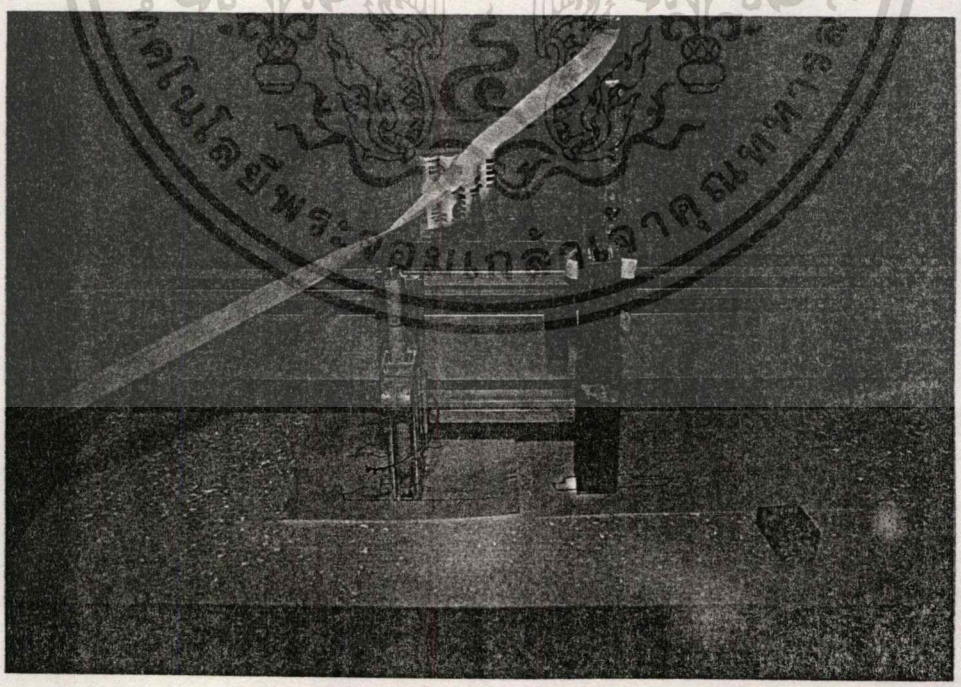
รูปที่ 7.3 แผงวงจรที่สมบูรณ์

รูปที่ 7.4 การติดตั้งแผงวงจรลงในคอมพิวเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

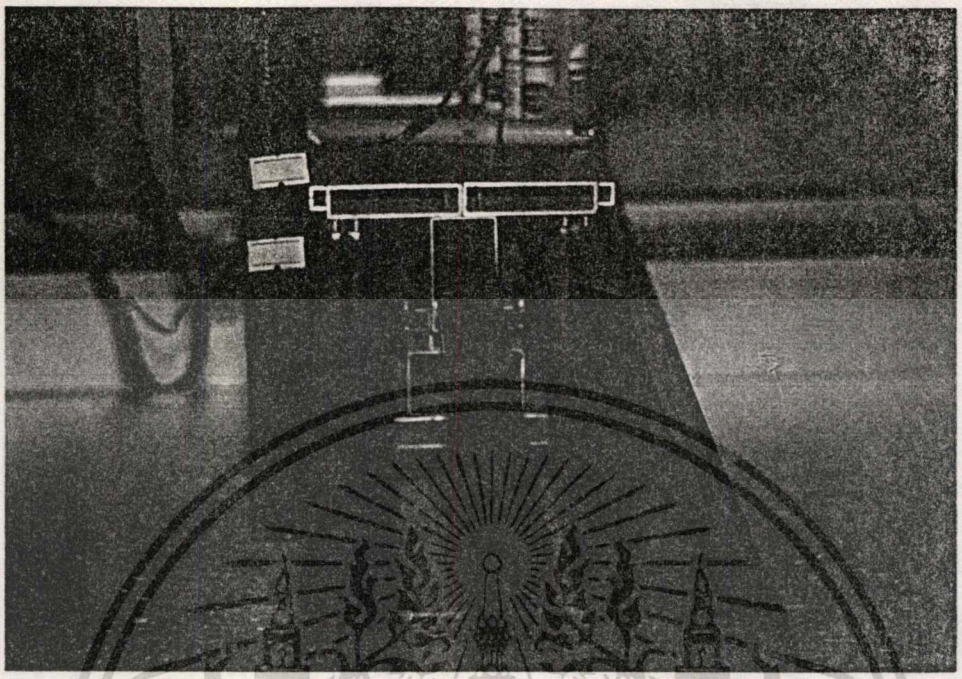


รูปที่ 7.5 ระบบกลไกขับเคลื่อนหัวอ่าน

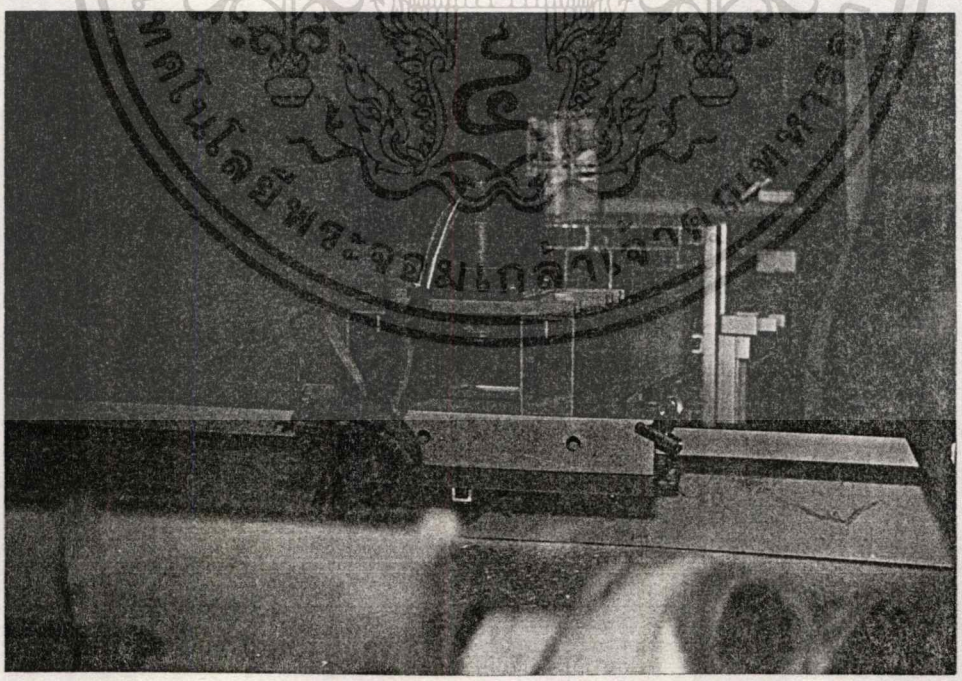


รูปที่ 7.6 ลักษณะการใช้งาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

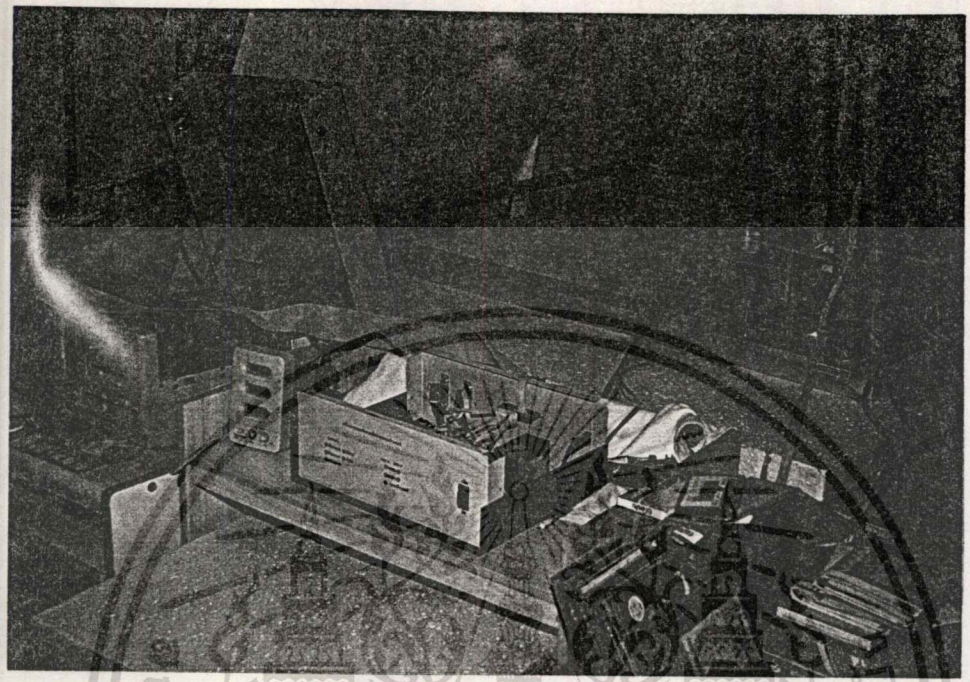


รูปที่ 7.7 การติดตั้งแท่นเลื่อนหัวอ่าน



รูปที่ 7.8 แสดงลักษณะการติดตั้งสายพานผ่านแกนมอเตอร์โดยตรง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 7.9 แสดงระบบที่ต่อถึงกันทั้งหมด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ข

รายละเอียดของโปรแกรมต่างๆ โดยสมบูรณ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรมหลักสำหรับการตัดและบันทึกชิ้นงาน

```
/*   CUTMAIN.C    */
#define MENU_ON        1
#define MENU_OFF       0
#define OK             1
#define  NO            0
#define ON             1
#define OFF            0
#define MOVE           2
#define MAX            5000 /* limit at 7000 */

#include<key_def.c>
#include<define.c>
#include<control.c>
#include<init.c>
#include<graph.c>

int  get_key();
void show_data(void);
void print_menu(void);

long X[MAX],Y[MAX];
int  end_data;
int  update_data_code;
int  data_exist_code;
#include<lo_save.c>
#include<display.c>
#include<cut.c>
#include<realtime.c>
#include<record.c>

int main()
{
   int process,i;
   update_data_code=NO;
   data_exist_code=NO;
   X[0]=Y[0]=0;
   clrscr();
   print_menu();
   process = ON;
   while(process==ON)
   switch(get_key())
   {
      case F1 : show_data();        print_menu(); break;
      case F2 : save_data();        print_menu(); break;
      case F3 : load_data();        show_data();  print_menu(); break;
      case F4 : record_menu();      show_data();  print_menu(); break;
      case F5 : show_data();        print_menu(); break;
      case F6 : cut_menu();         print_menu(); break;
      case F7 : realtime_menu();    print_menu(); break;
      case F8 : process=OFF;
   }
}

int get_key()
{
   int i;
   if(kbhit()==0x00)return 0;
   i=getch();
   if(i!=0x00)return i;
   i=0x0100+getch();
   return i;
}

void show_data(void)
{
   long x,y;
   int i;
   float ratiopicture,ratioscreen,ratio;
   long minx,miny;
   long cenx,ceny;
   long maxx,maxy;
   if(X[0]==0||Y[0]==0){putch(0x07);return;}
   init_graph();
   getmaxmin_data(&minx,&miny,&maxx,&maxy);
   cenx=(minx+maxx)/2;ceny=(miny+maxy)/2;
/*  x = start_x; y = start_y;*/
   x = X[0]; y = Y[0];
   ratiopicture = (float)(maxy-miny)/(float)(maxx-minx);
   ratioscreen  = (float)MaxY/(float)MaxX;
   if(ratiopicture>ratioscreen) ratio = (float)MaxY/(float)(maxy-miny);
   else     ratio = (float)MaxX/(float)(maxx-minx);
   setcolor(BLACK);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
/* lineto(g_x(cenx,start_x,ratio),g_y(ceny,start_y,ratio));*/
   lineto(g_x(cenx,X[0],ratio),g_y(ceny,Y[0],ratio));
   setcolor(YELLOW);
   printf("\n x = %5.2f m.m.",(float)(maxx-minx)/step_ratio);
   printf("\n y = %5.2f m.m.",(float)(maxy-miny)/step_ratio);
   rectangle(0,0,MaxX,MaxY);
/* i = 0;*/
   i = 1;
   setcolor(WHITE);
   while((X[i]!=0||Y[i]!=0)&&get_key()!=0x1b)
   {
      x=X[i];y=Y[i];
      lineto(g_x(cenx,x,ratio),g_y(ceny,y,ratio));
      i++;
   }
   setcolor(LIGHTRED);
   lineto(g_x(cenx,X[0],ratio),g_y(ceny,Y[0],ratio));
   i=getch();
   closegraph();
}
void print_menu(void)
{
   clrscr();
   printf("\n\n");
   printf("\n              WELLCOME TO USE ADVANCE CUTTING MACHINE ");
   printf("\n\n");
   printf("\n               PLEASE SELECT FUNCTION KEY TO USE ");
   printf("\n\n");
   printf("\n               [ F1 ]  HELP  ");
   printf("\n               [ F2 ]  SAVE DATA INTO DISK ");
   printf("\n               [ F3 ]  LOAD DATA FROM DISK ");
   printf("\n               [ F4 ]  LOAD DATA FROM PICTURE ");
   printf("\n               [ F5 ]  SHOW DATA ON GRAPHICS SCREEN ");
   printf("\n               [ F6 ]  DATA CUTTING ");
   printf("\n               [ F7 ]  REAL TIME CUTTING  ");
   printf("\n               [ F8 ]  QUIT     ");
   printf("\n\n\n\n\n\n");
}
```

โปรแกรมกำหนดค่าคงที่ของการกดแป้นพิมพ์

```
/************ KEY_DEF.C ***************/
#define ESC          0x1b
#define F1           0x013b
#define F2           0x013c
#define F3           0x013d
#define F4           0x013e
#define F5           0x013f
#define F6           0x0140
#define F7           0x0141
#define F8           0x0142
#define F9           0x0143
#define F10          0x0144
#define R_NARROW     0x014d
#define L_NARROW     0x014b
#define U_NARROW     0x0148
#define D_NARROW     0x0150
```

โปรแกรมกำหนดตัวแปรและตำแหน่งการรับส่งข้อมูล

```
/**************** DEFINE.C ***************/
#include<stdio.h>
#include<dos.h>
#include<conio.h>
#include<math.h>
#include<graphics.h>

#define  speedx1     0x300
#define  speedx2     0x301
#define  controlx    0x302
#define  func_x1     0x303 /* 0x88 */
#define  addxh       0x304
#define  addxl       0x305
#define  addxm       0x306
#define  func_x2     0x307 /* 0x9b */
#define  speedy1     0x308
#define  speedy2     0x309
#define  controly    0x30a
#define  func_y1     0x30b /* 0x88 */
#define  addyh       0x30c
#define  addyl       0x30d
#define  addym       0x30e
#define  func_y2     0x30f /* 0x9b */
#define  data_p11    0x310
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
#define  data_p12        0x311    /* photo i/p */
#define  data_p13        0x312
#define  func_p1         0x313 /* 0x9b */
#define  pi              3.1415926536
#define  step_ratio      271.6244362  /* 1 m.m = 271.6244 step*/
#define speed_ratio      2.730976105  /* ? m.m./speed_ratio = ? m.m./sec */
#define  speed_rate      99.46056859  /* ? step/speed_rate  = ? m.m./sec */
#define  up              1
#define  down            0
```

โปรแกรมย่อยสำหรับการควบคุมการเคลื่อนที่ของหัวอ่าน

```
/*************** CONTROL.C ***************/
void  test(void);
float  deg_to_rad(float deg);
void  go_home(void);
void  initport(void);
void  stop(void);
void  zero_port(void);
int     get_photo(void);
void  photo_on(void);
void  photo_off(void);
long  get_x(void);
long  get_y(void);
void  set_x_direction(int direct);
void  set_y_direction(int direct);
void  out_port(int port,int out);
void  put_x_speed(float speed);
void  put_y_speed(float speed);
int check_xmin(void);
int check_xmax(void);
int check_ymin(void);
int check_ymax(void);
void  set_v(long xi,long yi,float vi);
void  set_v2(long xi,long yi,long v);
void  go_rel(float xi,float yi,float vi,float ai);
void  go_to(float xi,float yi,float vi,float ai);
void  move_to(long xi,long yi,float vi);
void  go_zeta(float vi,float zeta);
void  arc_move(float xi,float yi,float rxi,float ryi,float oi1,float oi2,float vi,float ai);
void  arc_move1(float xi,float yi,float rxi,float ryi,float oi1,float oi2,float vi,float ai);

void  test()
{
   outportb(speedx1,0x10);
   outportb(speedx2,0x00);
   outportb(controlx,0x01);
   outportb(speedy1,0x10);
   outportb(speedy2,0x00);
   outportb(controly,0x01);
}

float  deg_to_rad(float deg)
{
   deg=deg*0.01745329252;
   return deg;
}

void  go_home()
{
   int flag_x=0;
   int flag_y=0;
   long  count=0;
   outportb(speedx1,0x20);
   outportb(speedx2,0x00);
   outportb(controlx,0x00);
   outportb(speedy1,0x20);
   outportb(speedy2,0x00);
   outportb(controly,0x00);
   while(flag_x==0||flag_y==0)
   {
      flag_x=inportb(controlx)&0x10;
      flag_y=inportb(controly)&0x10;
      count++;
      if(count>20000000){flag_x=flag_y=1;}
      if(kbhit()!=0)
      {
         if(getch()==0x1b)
         flag_x=flag_y=1;
      }
   }
   if(flag_x==1||flag_y==1)
      printf("\n go_home() error! "),exit();
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
void  initport()
{
    outportb(func_x1,0x88); /* AOBOHILO */
    outportb(func_x2,0x9b); /* AIBICI */
    outportb(func_y1,0x88); /* AOBOHILO */
    outportb(func_y2,0x9b); /* AIBICI */
    outportb(func_p1,0x9a); /* AIBIHILO */
}
void  stop()
{
    outportb(speedx1,0x00);
    outportb(speedx2,0x00);
    outportb(controlx,0x01);
    outportb(speedy1,0x00);
    outportb(speedy2,0x00);
    outportb(controly,0x01);
}
void  zero_port()
{
    int x,y;
    x=inportb(controlx);
    y=inportb(controly);
    outportb(controlx,0x00);
    outportb(controlx,0x04);
    outportb(controlx,0x00);
    outportb(controlx,x);
    outportb(controly,0x00);
    outportb(controly,0x04);
    outportb(controly,0x00);
    outportb(controly,y);
}
int    get_photo(void)
{
    int photo1;
    photo1 = inportb(data_p12);
    return photo1;
}

void  photo_on(void)
{
    short data;
    data = inportb(data_p13)&0xfe;
    outportb(data_p13,data);
    sleep(1);
    photo_off();
    delay(5);
    outportb(data_p13,data);
}
void  photo_off(void)
{
    short data;
    data = inportb(data_p13)|0x01;
    outportb(data_p13,data);
}

long  get_x(void)
{
    long  x=0;
    int    seg,off;
    seg=FP_SEG(&x);off=FP_OFF(&x);
    pokeb(seg,off,inportb(0x305));off++;
    pokeb(seg,off,inportb(0x306));off++;
    pokeb(seg,off,inportb(0x304));
    returnx;
}

long  get_y(void)
{
    long  y=0;
    int    seg,off;
    seg=FP_SEG(&y);off=FP_OFF(&y);
    pokeb(seg,off,inportb(0x30d));off++;
    pokeb(seg,off,inportb(0x30e));off++;
    pokeb(seg,off,inportb(0x30c));
    returny;
}

void  set_x_direction(int direct)
{
    char  new,old;
    if(direct == 1)new=0x01;
    else new=0x00;
    outportb(controlx,new);
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
void  set_y_direction(int direct)
{
   char  new,old;
   if(direct == 1)new=0x01;
   else new=0x00;
   outportb(controly,new);
}
void out_port(int port,int out)
{
   int seg,off;
   seg=FP_SEG(&out);
   off=FP_OFF(&out);
   port++;
   outportb(port,(peekb(seg,off)));off++;port--;
   outportb(port,(peekb(seg,off)));
}
void put_x_speed(float speed)
{
   int seg,off,out,port;
   port=speedx1;
   out=(int)fabs(speed);
   if(out>32765)  out=0x7ff0;
   if(speed>0)set_x_direction(up);
      else       set_x_direction(down);
   seg=FP_SEG(&out);
   off=FP_OFF(&out);
   port++;
   outportb(port,(peekb(seg,off)));off++;port--;
   outportb(port,(peekb(seg,off)));
}

void put_y_speed(float speed)
{
   int seg,off,out,port;
   port=speedy1;
   out=(int)fabs(speed);
   if(out>32765)  out=0x7ff0;
   if(speed>0)set_y_direction(up);
      else       set_y_direction(down);
   seg=FP_SEG(&out);
   off=FP_OFF(&out);
   port++;
   outportb(port,(peekb(seg,off)));off++;port--;
   outportb(port,(peekb(seg,off)));
}
int check_xmin(void)
{
   int check;
   check=inportb(controlx)&0x10;
   return check;
}
int check_xmax(void)
{
   int check;
   check=inportb(controlx)&0x20;
   return check;
}
int check_ymin(void)
{
   int check;
   check=inportb(controly)&0x10;
   return check;
}
int check_ymax(void)
{
   int check;
   check=inportb(controly)&0x20;
   return check;
}

void  set_v(long xi,long yi,float vi)
{
   long  x0,y0,x1,y1;
   float  dx,dy,dyx,dz,vx_max,vy_max;
   int    vx,vy;
   x0=(long)get_x();
   y0=(long)get_y();
   x1=(long)xi;
   y1=(long)yi;
   dx=(float)(x1-x0);
   dy=(float)(y1-y0);
   if(dy==0)dy=1e-7;
   if(dx==0)dx=1e-7;
```

เอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    dyx=(fabs)(dy/dx);
    dz=(float)atan(dyx);
    vi=vi/speed_ratio;
    vx_max=(fabs)(vi*cos(dz));
    vy_max=(fabs)(vi*sin(dz));
    if(vx_max>32766)vx=0x7ffe;
      else  vx=(int)vx_max;
    if(vy_max>32766)vy=0x7ffe;
      else  vy=(int)vy_max;
    if(dx>=0)  set_x_direction(up);
      else     set_x_direction(down);
    if(dy>=0)  set_y_direction(up);
      else     set_y_direction(down);
    out_port(speedx1,vx);
    out_port(speedy1,vy);
/*  printf(" spx= %d spy= %d",vx,vy);*/
 }

 void  set_v2(long xi,long yi,long v)
 {
    long  x0,y0,x1,y1;
    float  dx,dy,dyx,dz,vi,vx_max,vy_max;
    int    vx,vy;
    x0=(long)get_x();
    y0=(long)get_y();
    x1=(long)xi;
    y1=(long)yi;
    dx=(float)(x1-x0);
    dy=(float)(y1-y0);
    if(dy==0)dy=1e-7;
    if(dx==0)dx=1e-7;
    dyx=(fabs)(dy/dx);
    dz=(float)atan(dyx);
    vi=(float)v;
    vx_max=(fabs)(vi*cos(dz));
    vy_max=(fabs)(vi*sin(dz));
    if(vx_max>32766)vx=0x7ffe;
      else  vx=(int)vx_max;
    if(vy_max>32766)vy=0x7ffe;
      else  vy=(int)vy_max;
    if(dx>=0)  set_x_direction(up);
      else     set_x_direction(down);
    if(dy>=0)  set_y_direction(up);
      else     set_y_direction(down);
    out_port(speedx1,vx);
    out_port(speedy1,vy);
 }

 void  go_rel(float xi,float yi,float vi,float ai)
 {
    long  x0,y0,x1,y1,x_cen,y_cen;
    float  dx,dy,dxy,dz,vxy,a2i,s_cen,sxy,sx,sy,s_limit;
    x0=(long)get_x();
    y0=(long)get_y();
    x1=(long)(xi*step_ratio)+x0;
    if(x1<0)x1=0;
    y1=(long)(yi*step_ratio)+y0;
    if(y1<0)y1=0;
    vi=(vi*step_ratio);
    ai=(ai*step_ratio);
    a2i=ai*2;
    x_cen=(x0+x1)/2;y_cen=(y0+y1)/2;
    dx=(float)(x1-x0);dy=(float)(y1-y0);
/*  s_cen=sqrt((float)(x_cen*x_cen)+(float)(y_cen*y_cen));*/
    s_limit=vi*vi/(2*ai);
    if(fabs(dx)>=fabs(dy))
    {
      if(dx>=0)
      while(x1>(long)get_x()||x1>(long)get_x())
      {
        if(get_x()<x_cen)
        {
          sx=(float)(get_x()-x0);sy=(float)(get_y()-y0);
          sxy=sqrt((sx*sx+sy*sy));
          if(sxy>s_limit)vxy=vi;
          else  vxy=sqrt((fabs)(a2i*sxy));
        }
        else
        {
          sx=(float)(x1-get_x());sy=(float)(y1-get_y());
          sxy=sqrt((sx*sx+sy*sy));
          if(sxy>s_limit)vxy=vi;
          else  vxy=sqrt((fabs)(a2i*sxy));
/*        printf("ok");*/
```

เอกสารนี้ใช้ได้เพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            }
            if(vxy==0)vxy=50;
            set_v(x1,y1,vxy);
            if(check_xmax()>0)return ;
  /*        printf("\n x+ x= %8lx y= %8lx vxy %f ",get_x(),get_y(),vxy);*/
          }
          else
          while(x1<(long)get_x()||x1<(long)get_x())
          {
            if(get_x()>x_cen)
            {
              sx=(float)(get_x()-x0);sy=(float)(get_y()-y0);
              sxy=sqrt((sx*sx+sy*sy));
              if(sxy>s_limit)vxy=vi;
              else   vxy=sqrt((a2i*sxy));
            }
            else
            {
              sx=(float)(x1-get_x());sy=(float)(y1-get_y());
              sxy=sqrt((sx*sx+sy*sy));
              if(sxy>s_limit)vxy=vi;
              else   vxy=sqrt((a2i*sxy));
            }
            if(vxy==0)vxy=50;
            set_v(x1,y1,vxy);
            if(check_xmin()>0)return ;
  /*        printf("\n x- x= %8lx y= %8lx vxy %f ",get_x(),get_y(),vxy);*/
          }
        }
        else
        {
          if(dy>=0)
          while(y1>get_y()||y1>get_y())
          {
            if(get_y()<y_cen)
            {
              sx=(float)(get_x()-x0);sy=(float)(get_y()-y0);
              sxy=sqrt((sx*sx+sy*sy));
              if(sxy>s_limit)vxy=vi;
              else   vxy=sqrt((fabs)(a2i*sxy));
            }
            else
            {
              sx=(float)(x1-get_x());sy=(float)(y1-get_y());
              sxy=sqrt((sx*sx+sy*sy));
              if(sxy>s_limit)vxy=vi;
              else   vxy=sqrt((fabs)(a2i*sxy));
  /*          printf("ok");*/
            }
            if(vxy==0)vxy=50;
            set_v(x1,y1,vxy);
            if(check_ymax()>0)return ;
  /*        printf("\n y+ x= %8lx y= %8lx vxy %f ",get_x(),get_y(),vxy);*/
          }
          else
          while(y1<get_y()||y1<get_y())
          {
            if(get_y()>y_cen)
            {
              sx=(float)(get_x()-x0);sy=(float)(get_y()-y0);
              sxy=sqrt((sx*sx+sy*sy));
              if(sxy>s_limit)vxy=vi;
              else   vxy=sqrt((a2i*sxy));
            }
            else
            {
              sx=(float)(x1-get_x());sy=(float)(y1-get_y());
              sxy=sqrt((sx*sx+sy*sy));
              if(sxy>s_limit)vxy=vi;
              else   vxy=sqrt((a2i*sxy));
            }
            if(vxy==0)vxy=50;
            set_v(x1,y1,vxy);
            if(check_ymin()>0)return ;
  /*        printf("\n y- x= %8lx y= %8lx vxy %f ",get_x(),get_y(),vxy);*/
          }
        }
        stop();
        printf("\n stop at x=%ld y=%ld ",get_x(),get_y());
        printf("\n stop at x=%ld y=%ld ",get_x(),get_y());
        printf("\n goal at x=%ld y=%ld ",x1,y1);
  /*    sleep(2);*/
      }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
void  go_to(float xi,float yi,float vi,float ai)
{
   long  x0,y0,x1,y1,x_cen,y_cen;
   float  dx,dy,dxy,dz,vxy,a2i,s_cen,sxy,sx,sy,s_limit;
   x0=(long)get_x();
   y0=(long)get_y();
   x1=(long)(xi*step_ratio);
   if(x1<0)x1=0;
   y1=(long)(yi*step_ratio);
   if(y1<0)y1=0;
   vi=(vi*step_ratio);
   ai=(ai*step_ratio);
   a2i=ai*2;
   x_cen=(x0+x1)/2;y_cen=(y0+y1)/2;
   dx=(float)(x1-x0);dy=(float)(y1-y0);
/* s_cen=sqrt((float)(x_cen*x_cen)+(float)(y_cen*y_cen));*/
   s_limit=vi*vi/(2*ai);
   if(fabs(dx)>=fabs(dy))
   {
      if(dx>=0)
      while(x1>(long)get_x()||x1>(long)get_x())
      {
         if(get_x()<x_cen)
         {
            sx=(float)(get_x()-x0);sy=(float)(get_y()-y0);
            sxy=sqrt((sx*sx+sy*sy));
            if(sxy>s_limit)vxy=vi;
            else   vxy=sqrt((fabs)(a2i*sxy));
         }
         else
         {
            sx=(float)(x1-get_x());sy=(float)(y1-get_y());
            sxy=sqrt((sx*sx+sy*sy));
            if(sxy>s_limit)vxy=vi;
            else   vxy=sqrt((fabs)(a2i*sxy));
/*          printf("ok");*/
         }
         if(vxy==0)vxy=50;
         set_v(x1,y1,vxy);
         if(check_xmax()>0)return ;
/*       printf("\n x+ x= %8lx y= %8lx vxy %f ",get_x(),get_y(),vxy);*/
      }
      else
      while(x1<(long)get_x()||x1<(long)get_x())
      {
         if(get_x()>x_cen)
         {
            sx=(float)(get_x()-x0);sy=(float)(get_y()-y0);
            sxy=sqrt((sx*sx+sy*sy));
            if(sxy>s_limit)vxy=vi;
            else   vxy=sqrt((a2i*sxy));
         }
         else
         {
            sx=(float)(x1-get_x());sy=(float)(y1-get_y());
            sxy=sqrt((sx*sx+sy*sy));
            if(sxy>s_limit)vxy=vi;
            else   vxy=sqrt((a2i*sxy));
         }
         if(vxy==0)vxy=50;
         set_v(x1,y1,vxy);
         if(check_xmin()>0)return ;
/*       printf("\n x- x= %8lx y= %8lx vxy %f ",get_x(),get_y(),vxy);*/
      }
   }
   else
   {
      if(dy>=0)
      while(y1>get_y()||y1>get_y())
      {
         if(get_y()<y_cen)
         {
            sx=(float)(get_x()-x0);sy=(float)(get_y()-y0);
            sxy=sqrt((sx*sx+sy*sy));
            if(sxy>s_limit)vxy=vi;
            else   vxy=sqrt((fabs)(a2i*sxy));
         }
         else
         {
            sx=(float)(x1-get_x());sy=(float)(y1-get_y());
            sxy=sqrt((sx*sx+sy*sy));
            if(sxy>s_limit)vxy=vi;
            else   vxy=sqrt((fabs)(a2i*sxy));
/*          printf("ok");*/
```

เอกสารนี้สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            }
            if(vxy==0)vxy=50;
            set_v(x1,y1,vxy);
            if(check_ymax()>0)return ;
/*          printf("\n y+ x= %8lx y= %8lx vxy %f ",get_x(),get_y(),vxy);*/
         }
         else
         while(y1<get_y()||y1<get_y())
         {
            if(get_y()>y_cen)
            {
               sx=(float)(get_x()-x0);sy=(float)(get_y()-y0);
               sxy=sqrt((sx*sx+sy*sy));
               if(sxy>s_limit)vxy=vi;
               else   vxy=sqrt((a2i*sxy));
            }
            else
            {
               sx=(float)(x1-get_x());sy=(float)(y1-get_y());
               sxy=sqrt((sx*sx+sy*sy));
               if(sxy>s_limit)vxy=vi;
               else   vxy=sqrt((a2i*sxy));
            }
            if(vxy==0)vxy=50;
            set_v(x1,y1,vxy);
            if(check_ymin()>0)return ;
/*          printf("\n y- x= %8lx y= %8lx vxy %f ",get_x(),get_y(),vxy);*/
         }
      }
      stop();
      printf("\n stop at x=%ld y=%ld ",get_x(),get_y());
      printf("\n stop at x=%ld y=%ld ",get_x(),get_y());
      printf("\n goal at x=%ld y=%ld ",x1,y1);
}

void  move_to(long xi,long yi,float vi)
{
      long  x0,y0,x1,y1,x_cen,y_cen;
      float  dx,dy,dxy,dz,vxy;
      x0=(long)get_x();
      y0=(long)get_y();
      x1=xi;
      if(x1<0)x1=0;
      y1=yi;
      if(y1<0)y1=0;
      dx=(float)(x1-x0);dy=(float)(y1-y0);
      vi=vi*step_ratio;
      if(fabs(dx)>=fabs(dy))
      {
         if(dx>=0)
         while(x1>(long)get_x()||x1>(long)get_x())
         {
            set_v(x1,y1,vi);
            if(check_xmax()>0)return ;
         }
         else
         while(x1<(long)get_x()||x1<(long)get_x())
         {
            set_v(x1,y1,vi);
            if(check_xmin()>0)return ;
         }
      }
      else
      {
         if(dy>=0)
         while(y1>get_y()||y1>get_y())
         {
            set_v(x1,y1,vi);
            if(check_ymax()>0)return ;
         }
         else
         while(y1<get_y()||y1<get_y())
         {
            set_v(x1,y1,vi);
            if(check_ymin()>0)return ;
         }
      }
}

void  go_zeta(float vi,float zeta)
{
      float  vx,vy;
      vi=vi*speed_rate;
      zeta=deg_to_rad(zeta);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    vx=vi*cos(zeta);
    vy=vi*sin(zeta);
    put_x_speed(vx);
    put_y_speed(vy);
}

void  arc_move(float xi,float yi,float rxi,float ryi,float oi1,float oi2,float vi,float ai)
{
    float  x0,y0,x1,y1,x2,y2,x3,y3,x4,y4,dx,dy,dyx,dz,d1;
    float  oa,oc,ot,os,ov,ov_new,ov_old,vxy,r0,ri,count,p2i;
    float  vx_max,vy_max;
    float  og;
    int    vx,vy,flat;
    int    counter,test;
    float  zeta,sinn,coss;

    p2i=2*pi;
    while(oi1<0){oi1=oi1+p2i;}
    oi2=oi1+oi2;
    oc=(oi1+oi2)/2;
    ov_new=oi1;
    ov_old=ov_new-1;
    ot=pi;
    rxi=rxi*step_ratio;
    ryi=ryi*step_ratio;
    rxi=12000;ryi=10000;
    if(rxi>ryi)ri=ryi;
    else  ri=rxi;
    vi=vi*step_ratio;
    ai=ai*step_ratio;
    x0=(xi*step_ratio);
    y0=(yi*step_ratio);
    x1=(rxi*cos(oi1))+x0;
    y1=(rxi*sin(oi1))+y0;
    go_to(x1/step_ratio,y1/step_ratio,300,200);
    flat=0;
    count=0;
    test=0;
    counter=0;
    zeta=-2;
    if(oi2>=oi1)
    {
       x2=(float)(get_x());
       y2=(float)(get_y());
       dx=x2-x0;dy=y2-y0;
       if(dx==0)dx=1e-9;
       ov=atan(dy/dx);
       if(dx<0) ov=pi+ov;
       if(dy<0&&dx>0) ov=p2i+ov;
       if(fabs(oi1-ov)>1)
       {
          if(oi1<pi) count=-1;
          else  count=0;
       }
       ov_new=ov+p2i*count;
       ov_old=ov_new-1;
       while(ov_new<oi2)
       {
          x2=(float)(get_x());
          y2=(float)(get_y());
          dx=x2-x0;dy=y2-y0;
          if(dx==0)dx=1e-9;
          ov=atan(dy/dx);
          if(dx<0) ov=pi+ov;
          if(dy<0&&dx>0) ov=p2i+ov;
          ov_new=ov+p2i*count;
          if(ov_new<ov_old){count++;ov_new=ov+p2i*count;ot=ot+p2i;}
          printf("\nst%7.3f ",ov_new*57.25978);
          oa=ov_old-ov_new+.2;
          ov_old=ov_new-.1;
          if(ov_new<oc)os=ov_new-oi1;
          else  os=oi2-ov_new;
          vxy=sqrt(fabs(4*ri*ai*os));
          if(vxy>vi)vxy=vi;
          if(vxy<=5)vxy=50;
          oa=.1;
          og=oa+ov_new;
          coss=cos(og);sinn=sin(og);
          x3=(rxi*(coss))+x0;
          y3=(rxi*(sinn))+y0;
/*        set_v((long)x3,(long)y3,(float)vxy);*/
          dx=(float)(x3-x2);
          dy=(float)(y3-y2);
          if(dx==0)dx=1e-7;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            dyx=(fabs)(dy/dx);
            dz=(float)atan(dyx);
            vx_max=(fabs)((vxy*cos(dz))/speed_ratio);
            vy_max=(fabs)((vxy*sin(dz))/speed_ratio);
            if(vx_max>32765)vx=0x7ff0;
            else  vx=(int)vx_max;
            if(vy_max>32765)vy=0x7ff0;
               else  vy=(int)vy_max;
            if(dx>=0)  set_x_direction(1);
               else  set_x_direction(0);
            if(dy>=0)  set_y_direction(1);
               else  set_y_direction(0);
            dx=x3-x2;dy=y3-y2;
            if(zeta>og)test++;
            printf("ze%7.3f og%7.4f x%8.4f y%8.4f dx%8.3f dy%8.3f",zeta*57.29578,og*57.29578,x3-x0,y3-y0,dx,dy);
            zeta=og;
            if(test>3)
            {counter++;if(counter>5){stop();return;}}
            out_port(speedx1,vx);
            out_port(speedy1,vy);
         }
      }
      else
      {
         x2=(float)(get_x());
         y2=(float)(get_y());
         dx=x2-x0;dy=y2-y0;
         if(dx==0)dx=1e-9;
         ov=atan(dy/dx);
         if(dx<0) ov=pi+ov;
         if(dy<0&&dx>0) ov=p2i+ov;
         if(fabs(oi1-ov)>1)
         {
            if(oi1<pi) count=-1;
            else  count=0;
         }
         ov_new=ov-p2i*count;
         ov_old=ov_new+1;
         while(ov_new>oi2)
         {
            x2=(float)(get_x());
            y2=(float)(get_y());
            dx=x2-x0;dy=y2-y0;
            if(dx==0)dx=1e-9;
            ov=atan(dy/dx);
            if(dx<0) ov=pi+ov;
            if(dy<0&&dx>0) ov=p2i+ov;
            ov_new=ov-p2i*count;
            if(ov_new>ov_old)
            {if(count<flat){count++;ov_new=ov-p2i*count;ot=ot-p2i;}}
            if(flat==0){if(ov_new>6&&oi1<1)ov_new=ov_new-p2i;
            if(ov_new<6&&oi1>6)ov_new=ov_new+p2i;}
            if(ov_new<ot)
               if(count==flat)flat++;
            ov_old=ov_new+1;
            if(ov_new<oc)os=ov_new-oi2;
               else  os=oi1-ov_new;
            vxy=sqrt(fabs(4*ri*ai*os));
            if(vxy>vi)vxy=vi;
            if(vxy<=5)vxy=50;
               oa=0.05;
            og=(float)(ov_new-oa);
            x3=rxi*(float)cos(og)+x0;
            y3=ryi*(float)sin(og)+y0;
            printf("\nov_new= %3.6f x3= %7.1f y3= %7.1f vxy= %7.1f f=%d c=%2.0f",ov_new,x3-x0,y3-y0,vxy,flat,count);
            set_v((long)x3,(long)y3,(float)vxy);
         }
      }
      stop();
}

spiral_move(float xi,float yi,float rxi,float ryi,float spiral_x,float spiral_y,float oi1,float oi2,float vi,float ai)
{
   float  x0,y0,x1,y1,x2,y2,x3,y3,x4,y4,dx,dy,dyx,dz,d1;
   float  oa,oc,ot,os,ov,ov_new,ov_old,vxy,r0,ri,rx,ry,count,p2i;
   float  vx_max,vy_max;
   float  og;
   int    vx,vy,flat;
   int    counter,test;
   float  zeta,sinn,coss;

   p2i=2*pi;
   while(oi1<0){oi1=oi1+p2i;}
   oi2=oi1+oi2;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    oc=(oi1+oi2)/2;
    ov_new=oi1;
    ov_old=ov_new-1;
    ot=pi;
    rxi=rxi*step_ratio;
    ryi=ryi*step_ratio;
    spiral_x=spiral_x*step_ratio;
    spiral_y=spiral_y*step_ratio;
/*  rxi=12000;ryi=10000;*/
    if(rxi>ryi)ri=ryi;
    else  ri=rxi;
    vi=vi*step_ratio;
    ai=ai*step_ratio;
    x0=(xi*step_ratio);
    y0=(yi*step_ratio);
    x1=(rxi*cos(oi1))+x0;
    y1=(ryi*sin(oi1))+y0;
    go_to(x1/step_ratio,y1/step_ratio,300,200);
    flat=0;
    count=0;
    test=0;
    counter=0;
    zeta=-2;
    if(oi2>=oi1)
    {
      x2=(float)(get_x());
      y2=(float)(get_y());
      dx=x2-x0;dy=y2-y0;
      if(dx==0)dx=1e-9;
      ov=atan(dy/dx);
      if(dx<0) ov=pi+ov;
      if(dy<0&&dx>0) ov=p2i+ov;
      if(fabs(oi1-ov)>1)
      {
        if(oi1<pi) count=-1;
        else  count=0;
      }
      ov_new=ov+p2i*count;
      ov_old=ov_new-1;
      while(ov_new<oi2)
      {
        x2=(float)(get_x());
        y2=(float)(get_y());
        dx=x2-x0;dy=y2-y0;
        if(dx==0)dx=1e-9;
        ov=atan(dy/dx);
        if(dx<0) ov=pi+ov;
        if(dy<0&&dx>0) ov=p2i+ov;
        ov_new=ov+p2i*count;
        if(ov_new<ov_old){count++;ov_new=ov+p2i*count;ot=ot+p2i;}
        printf("\nst%7.3f ",ov_new*57.25978);
        oa=ov_old-ov_new+.2;
        ov_old=ov_new-.1;
        if(ov_new<oc)os=ov_new-oi1;
        else  os=oi2-ov_new;
        vxy=sqrt(fabs(4*ri*ai*os));
        if(vxy>vi)vxy=vi;
        if(vxy<=5)vxy=50;
        oa=.1;
        og=oa+ov_new;
        coss=(float)cos(og);sinn=(float)sin(og);
        rx=rxi+spiral_x*(fabs)(og/(oi2-oi1));
        ry=ryi+spiral_y*(fabs)(og/(oi2-oi1));
        x3=(rx*coss)+x0;
        y3=(ry*sinn)+y0;
/*      set_v((long)x3,(long)y3,(float)vxy);*/
        dx=(float)(x3-x2);
        dy=(float)(y3-y2);
        if(dx==0)dx=1e-7;
        dyx=(fabs)(dy/dx);
        dz=(float)atan(dyx);
        vx_max=(fabs)((vxy*cos(dz))/speed_ratio);
        vy_max=(fabs)((vxy*sin(dz))/speed_ratio);
        if(vx_max>32765)vx=0x7ff0;
        else  vx=(int)vx_max;
        if(vy_max>32765)vy=0x7ff0;
          else  vy=(int)vy_max;
        if(dx>=0)  set_x_direction(1);
          else  set_x_direction(0);
        if(dy>=0)  set_y_direction(1);
          else  set_y_direction(0);
        dx=x3-x2;dy=y3-y2;
        if(zeta>og)test++;
/*      printf("ze%7.3f og%7.4f x%8.4f y%8.4f dx%8.3f dy%8.3f",zeta*57.29578,og*57.29578,x3-x0,y3-y0,dx,dy);*/
```

เอกสารนี้สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            zeta=og;
            if(test>3)
            {counter++;if(counter>5){stop();return;}}
            out_port(speedx1,vx);
            out_port(speedy1,vy);
         }
      }
      else
      {
         x2=(float)(get_x());
         y2=(float)(get_y());
         dx=x2-x0;dy=y2-y0;
         if(dx==0)dx=1e-9;
         ov=atan(dy/dx);
         if(dx<0) ov=pi+ov;
         if(dy<0&&dx>0) ov=p2i+ov;
         if(fabs(oi1-ov)>1)
         {
            if(oi1<pi) count=-1;
            else   count=0;
         }
         ov_new=ov-p2i*count;
         ov_old=ov_new+1;
         while(ov_new>oi2)
         {
            x2=(float)(get_x());
            y2=(float)(get_y());
            dx=x2-x0;dy=y2-y0;
            if(dx==0)dx=1e-9;
            ov=atan(dy/dx);
            if(dx<0) ov=pi+ov;
            if(dy<0&&dx>0) ov=p2i+ov;
            ov_new=ov-p2i*count;
            if(ov_new>ov_old)
            {if(count<flat){count++;ov_new=ov-p2i*count;ot=ot-p2i;}}
            if(flat==0){if(ov_new>6&&oi1<1)ov_new=ov_new-p2i;
            if(ov_new<6&&oi1>6)ov_new=ov_new+p2i;}
            if(ov_new<ot)
               if(count==flat)flat++;
            ov_old=ov_new+1;
            if(ov_new<oc)os=ov_new-oi2;
               else   os=oi1-ov_new;
            vxy=sqrt(fabs(4*ri*ai*os));
            if(vxy>vi)vxy=vi;
            if(vxy<=5)vxy=50;
            oa=0.1;
            og=(float)(ov_new-oa);
            rx=rxi+spiral_x*(fabs)(og/(oi2-oi1));
            ry=ryi+spiral_y*(fabs)(og/(oi2-oi1));
            x3=rx*(float)cos(og)+x0;
            y3=ry*(float)sin(og)+y0;
/*          printf("\nov_new= %3.6f x3= %7.1f y3= %7.1f vxy= %7.1f f=%d c=%2.0f",ov_new,x3-x0,y3-y0,vxy,flat,count);*/
            set_v((long)x3,(long)y3,(float)vxy);
         }
      }
      stop();
}
```

โปรแกรมเริ่มต้นระบบ

```
/******** INIT.C **********/
void init_port()
{
   int i;
   initport();
   photo_off();
   go_home();
   stop();
   delay(500);
   zero_port();
}
```

โปรแกรมจัดการด้านรูปภาพบนจอ

```
/************ GRAPH.C **************/
int GraphDriver;
int GraphMode;
double AspectRatio;
int MaxX,MaxY;
int MaxColors;
int ErrorCode;
struct palettetype palette;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
void init_graph(void)
{
   int xasp, yasp;                          /* Used to read the aspect ratio*/
   GraphDriver = DETECT;                    ./* Request auto-detection */
  initgraph( &GraphDriver, &GraphMode, "" );
   ErrorCode = graphresult();               /* Read result of initialization*/
   if( ErrorCode != grOk ){                 /* Error occured during init */
   printf(" Graphics System Error: %s\n", grapherrormsg( ErrorCode ) );
   exit( 1 );
 }
   getpalette( &palette );                  /* Read the palette from board*/
   MaxColors = getmaxcolor() + 1;           /* Read maximum number of colors*/
   MaxX = getmaxx();
   MaxY = getmaxy();                        /* Read size of screen       */
   getaspectratio( &xasp, &yasp );          /* read the hardware aspect*/
   AspectRatio = (double)xasp / (double)yasp;   /* Get correction factor  */
}
```

โปรแกรมเกี่ยวกับการอ่านและเขียนข้อมูลกับจานเก็บข้อมูล

```
/************** LO_SAVE.C **************/
void load_data();
void save_data();
void clear_data(void);
void clear_data(void)
{
   int i;
   for(i=0;i<=MAX;i++)
   {  X[i]=Y[i]=0;   }
}
void load_data()
{
   FILE *fp2;
   int i;
   float dat;
   char filename[50];
   char data[50];
   i=0;
   printf("\rplease enter file's name\n ");
   scanf("%s",filename);
   if((fp2=fopen(filename,"r"))!=NULL)
   {
      clear_data();
      fscanf(fp2,"%s",data);dat=atof(data);X[0]=(long)dat;
      getc(fp2);
      fscanf(fp2,"%s",data);dat=atof(data);Y[0]=(long)dat;
      getc(fp2);getc(fp2);
      i=0;
      while(i<=MAX&&(X[i]!=0||Y[i]!=0))
      {
         i++;
         fscanf(fp2,"%s",data);dat=atof(data);X[i]=(long)dat;
         getc(fp2);
         fscanf(fp2,"%s",data);dat=atof(data);Y[i]=(long)dat;
         getc(fp2);getc(fp2);
      }
      end_data=i;
      putc(EOF,fp2);
      fclose(fp2);
      printf("\rLoad file %s have complete                    ",filename);
      data_exist_code=OK;
      update_data_code=OK;
   }
   else
   {
      printf("\n no data ");
      putch(0x07);
      return;
   }
}
void save_data()
{
   FILE *fp2;
   int i;
   char filename[50];
   char ch=0x00;
   i=0;
   if(data_exist_code==NO)
   {
      printf("\rNot data to save ");
      putch(0x07);
      while(kbhit()==0x00);
      return;
   }
```

เอกสารนี้สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    printf("\rplease enter the file's name\n");
    scanf("%s",filename);
    if((fp2=fopen(filename,"r"))!=NULL)
    {
       printf("\nOver write on file %s ? (Y/N)",filename);
       fclose(fp2);ch=0;
       while(ch!='y'&&ch!='n'&&ch!='Y'&&ch!='N'&&ch!=0x1b)
       {
          putch(0x07);
          ch=getch();
          if(ch==0x1b)return;
       }
       if(ch=='n'||ch=='N')return;
       putch(0x07);
       printf("\rsave file %s please wait          ",filename);
    }
    if((fp2=fopen(filename,"w"))!=NULL)
    {
       i=0;
       while(i<=MAX&&(X[i]!=0||Y[i]!=0))
       {
          fprintf(fp2,"%g",(float)X[i]);
          putc(NULL,fp2);
          fprintf(fp2,"%g",(float)Y[i]);
          putc(NULL,fp2);putc(0x0a,fp2);
          i++;
       }
       fprintf(fp2,"%g",0.0);putc(NULL,fp2);
       fprintf(fp2,"%g",0.0);putc(NULL,fp2);
       putc(EOF,fp2);
       fclose(fp2);
       printf("\rsave file %s have complete                ",filename);
    }
    else
    {
       printf("\rCan't save file %s              ",filename);
       putch(0x07);
    }
    while(kbhit()==0);
}
```

โปรแกรมควบคุมการตัดชิ้นงานโดยอาศัยข้อมูลที่มีอยู่ในหน่วยความจำ

```
/************ CUT.C ************/
void cut_menu(void);
void print_cut_menu(float v,float size_cut);
void set_size_cut(float *size);
void set_v_cut(float *v);
void cut_process(float cut_speed,float size1);

void cut_menu(void)
{
   int process;
   static float v,size_cut;
   update_data_code=NO;
   data_exist_code=NO;
   clrscr();
   v = 1000;
   size_cut = 100;
   print_cut_menu(v,size_cut);
   process = ON;
   while(process==ON)
   switch(get_key())
   {
      case F4 : set_size_cut(&size_cut); print_cut_menu(v,size_cut); break;
      case F5 : set_v_cut(&v);  print_cut_menu(v,size_cut);  break;
      case F6 : cut_process(v,size_cut);  print_cut_menu(v,size_cut); break;
      case F8 : process=OFF;
   }
}
void print_cut_menu(float v,float size_cut)
{
   clrscr();
   printf("\n\n");
   printf("\n\n");
   printf("\n\n");
   printf("\n              [ F4 ]  SET SIZE ( SIZE = %2.2f %)",size_cut);
   printf("\n              [ F5 ]  SET V ( V = %4.3f m.m./minute)",v);
   printf("\n              [ F6 ]  CUT  ");
   printf("\n              [ F8 ]  QUIT     ");
   printf("\n\n\n\n\n\n");
}
void set_size_cut(float *size)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
void set_size_cut(float *size)
{
   printf("\r size = ");scanf("%f",size);
}
void set_v_cut(float *v)
{
   printf("\r v = ");scanf("%f",v);
}
void cut_process(float cut_speed,float size1)
{
   long x,y;
   int i;
   float cut_speed2;
   cut_speed2 = (cut_speed/(speed_ratio*22.55));
   size1 = size1/100;
   init_port();
   getch();
   x = (long)((float)X[0]*size1); y = (long)((float)Y[0]*size1);
   go_to((float)(x)/step_ratio,(float)(y)/step_ratio,100,100);
   i=1;
   while((X[i]!=0||Y[i]!=0)&&get_key()!=0x1b)
   {
      x=(long)((float)X[i]*size1);y=(long)((float)Y[i]*size1);
      move_to(x,y,cut_speed2);
      i++;
   }
   x=(long)((float)X[0]*size1);y=(long)((float)Y[0]*size1);
   move_to(x,y,cut_speed2);
   stop();   printf("%d",i);
}
```

โปรแกรมแสดงข้อมูลในหน่วยความจำขึ้นบนจอภาพ

```
/************** DISPLAY.C ****************/
long g_xl(long center,int xi,float ratio);
long g_yl(long center,int yi,float ratio);
int  g_x(long center,long xl,float ratio);
int  g_y(long center,long yl,float ratio);
void getmaxmin_data(long *minx,long *miny,long *maxx,long *maxy);
void redraw_edit(long cenx,long ceny,float ratio,int start_edit,int end_edit);
void dot_edit(long cenx,long ceny,float ratio,int start_edit,int end_edit);

long g_xl(long center,int xi,float ratio)
{
   long l;
   l = (long)( ((float)(xi)-((float)MaxX)/2.0)/ratio )+center;
   return l;
}
long g_yl(long center,int yi,float ratio)
{
   long l;
   l = center-(long)( ((float)(yi+1)-((float)MaxY)/2.0)/ratio );
   return l;
}
int g_x(long center,long xl,float ratio)
{
   int i;
   i = (int)((((float)(xl-center))*ratio)+((((float)MaxX)/2.0)));
   return i;
}
int g_y(long center,long yl,float ratio)
{
   int y;
   y = (int)((((float)MaxY)/2.0)-(((float)(yl-center))*ratio));
   return y;
}
void getmaxmin_data(long *minx,long *miny,long *maxx,long *maxy)
{
   long x,y;
   int i=0;
   x=*minx=*miny=X[0];
   y=*maxx=*maxy=Y[0];
   while((X[i]!=0||Y[i]!=0)&&get_key()!=0x1b)
   {
      x=X[i];y=Y[i];
      if(x>*maxx)*maxx=x;
      if(x<*minx)*minx=x;
      if(y>*maxy)*maxy=y;
      if(y<*miny)*miny=y;
      i++;
   }
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
void redraw_edit(long cenx,long ceny,float ratio,int start_edit,int end_edit)
{
   int i,x_dot,y_dot,color;
   clearviewport();
   color=WHITE;
   setwritemode(COPY_PUT);
   setcolor(BLACK);
   x_dot = g_x(cenx,X[0],ratio);
   y_dot = g_y(ceny,Y[0],ratio);
   lineto(x_dot,y_dot);
   setfillstyle(SOLID_FILL,DARKGRAY);
   bar(1,g_y(ceny,0,ratio),639,479);
   bar(1,1,g_x(cenx,0,ratio),479);
   setcolor(YELLOW);
   gotoxy(2,2);
/* printf("\r x = %6ld y = %6ld ",start_x,start_y);*/
/* printf("\r x = %5.2f m.m.",(float)(((float)MaxX/ratio)/step_ratio));
   printf("\n y = %5.2f m.m.",(float)(((float)MaxY/ratio)/step_ratio));*/
   rectangle(0,0,MaxX,MaxY);
   i=0;
   setcolor(color);
   while((X[i]!=0||Y[i]!=0)&&get_key()!=0x1b)
   {
      if(i>start_edit&&i<end_edit)setcolor(LIGHTGREEN);
      else setcolor(color);
      x_dot = g_x(cenx,X[i],ratio);
      y_dot = g_y(ceny,Y[i],ratio);
      lineto(x_dot,y_dot);
      i++;
   }
   setcolor(LIGHTRED);
   x_dot = g_x(cenx,X[0],ratio);
   y_dot = g_y(ceny,Y[0],ratio);
   lineto(x_dot,y_dot);
}

void redot_edit(long cenx,long ceny,float ratio,int start_edit,int end_edit)
{
   int i,x_dot,y_dot,dot_color,line_color;
   clearviewport();
   dot_color=WHITE;
   line_color=BROWN;
   x_dot = start_edit;x_dot=end_edit;
   setcolor(BLACK);
   x_dot = g_x(cenx,X[0],ratio);
   y_dot = g_y(ceny,Y[0],ratio);
   lineto(x_dot,y_dot);
   putpixel(x_dot,y_dot,dot_color);
   setfillstyle(SOLID_FILL,DARKGRAY);
   bar(1,g_y(ceny,0,ratio),639,479);
   bar(1,1,g_x(cenx,0,ratio),479);
   setcolor(YELLOW);
   rectangle(0,0,MaxX,MaxY);
   i=0;
   setcolor(line_color);
   while((X[i]!=0||Y[i]!=0)&&get_key()!=0x1b)
   {
      if(i>start_edit&&i<end_edit)setcolor(LIGHTGREEN);
      else setcolor(line_color);
      x_dot = g_x(cenx,X[i],ratio);
      y_dot = g_y(ceny,Y[i],ratio);
      lineto(x_dot,y_dot);
      i++;
   }
   setcolor(LIGHTRED);
   x_dot = g_x(cenx,X[0],ratio);
   y_dot = g_y(ceny,Y[0],ratio);
   lineto(x_dot,y_dot);
   i=0;
   while((X[i]!=0||Y[i]!=0)&&get_key()!=0x1b)
   {
      x_dot = g_x(cenx,X[i],ratio);
      y_dot = g_y(ceny,Y[i],ratio);
      putpixel(x_dot,y_dot,dot_color);
      i++;
   }
   x_dot = g_x(cenx,X[0],ratio);
   y_dot = g_y(ceny,Y[0],ratio);
   putpixel(x_dot,y_dot,dot_color);
/* lineto(g_x(cenx,start_x,ratio),g_y(ceny,start_y,ratio));*/
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรมตามลายเส้นพร้อมกับตัดชิ้นงาน

```
/*   realtime.c      */

void realtime_menu();
void print_realtime_menu(void);
#include<pid_cut.c>
#include<pro_cut.c>
#include<mir_cut.c>

void realtime_menu()
{
   int process,i;
   update_data_code=NO;
   data_exist_code=NO;
   clrscr();
   print_realtime_menu();
   process = ON;
   while(process==ON)
   switch(get_key())
   {
      case F1 : break;
      case F2 : pid_main();  print_realtime_menu();  break;
      case F3 : pro_main();  print_realtime_menu();  break;
      case F4 : mir_main();   print_realtime_menu();  break;
      case F8 : process=OFF;
   }
}
void print_realtime_menu(void)
{
   clrscr();
   printf("\n\n");
   printf("\n              WELLCOME TO REAL TIME CUTTING MODE ");
   printf("\n\n");
   printf("\n              PLEASE SELECT FUNCTION KEY TO USE ");
   printf("\n\n");
   printf("\n              [ F1 ]  HELP  ");
   printf("\n              [ F2 ]  PID MODE ");
   printf("\n              [ F3 ]  P AND PROGAME MODE ");
   printf("\n              [ F4 ]  MIRROR AND PROGAME MODE ");
   printf("\n              [ F8 ]  QUIT      ");
   printf("\n\n\n\n\n\n");
}

/************ PID_MAIN.C ***********/

void pid_main(void);
void print_pid_menu(float *p,float *i,float *d,float *v,float *il);
set_p_pid(float *p);
set_i_pid(float *i);
set_d_pid(float *d);
set_v_pid(float *v);
set_il_pid(float *il);
cut_pid(float p,float i,float d,float v,float il);
photo_tracking_pid1(float kp,float ki,float kd,float v,float il);

void pid_main(void)
{
   int process;
   static float p,i,d,v,il;
   update_data_code=NO;
   data_exist_code=NO;
   clrscr();
   p=10;i=1;d=5000;v=100,il=1000;
   print_pid_menu(&p,&i,&d,&v,&il);
   process = ON;
   while(process==ON)
   switch(get_key())
   {
      case F1 : p=1;i=1;d=5000;v=1000;il=1000;print_pid_menu(&p,&i,&d,&v,&il);break;
      case F2 : set_p_pid(&p);    print_pid_menu(&p,&i,&d,&v,&il);  break;
      case F3 : set_i_pid(&i);    print_pid_menu(&p,&i,&d,&v,&il);  break;
      case F4 : set_d_pid(&d);    print_pid_menu(&p,&i,&d,&v,&il);  break;
      case F5 : set_v_pid(&v);    print_pid_menu(&p,&i,&d,&v,&il);  break;
      case F6 : cut_pid(p,i,d,v,il); print_pid_menu(&p,&i,&d,&v,&il);  break;
      case F7 : set_il_pid(&il);   print_pid_menu(&p,&i,&d,&v,&il);  break;
      case F8 : process=OFF;
   }
}
void print_pid_menu(float *p,float *i,float *d,float *v,float *il)
{
   clrscr();
   printf("\n\n");
   printf("\n\n");
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    printf("\n\n");
    printf("\n              [ F1 ]  RESET PID (P=10 I=1 D=5000)");
    printf("\n              [ F2 ]  SET P ( P = %4.3f )",*p);
    printf("\n              [ F3 ]  SET I ( I = %4.3f )",*i);
    printf("\n              [ F4 ]  SET D ( D = %4.3f )",*d);
    printf("\n              [ F5 ]  SET V ( V = %4.3f m.m./minute)",*v);
    printf("\n              [ F6 ]  CUT  ");
    printf("\n              [ F7 ]  SET I LIMIT ( IL = %4.3f )",*il);
    printf("\n              [ F8 ]  QUIT     ");
    printf("\n\n\n\n\n\n");
}
set_p_pid(float *p)
{
   printf("\r p = ");scanf("%f",p);
}
set_i_pid(float *i)
{
   printf("\r i = ");scanf("%f",i);
}
set_d_pid(float *d)
{
   printf("\r d = ");scanf("%f",d);
}
set_v_pid(float *v)
{
   printf("\r v = ");scanf("%f",v);
}
set_il_pid(float *il)
{
   printf("\r il = ");scanf("%f",il);
}

cut_pid(float p,float i,float d,float v,float il)
{
   int count=0;
   init_port();
   photo_on();
   printf("\n\n\n");
   while(get_photo()<250&&kbhit()==0x00&&++count<=21);
   go_to(10,10,100,100);
   go_zeta(10,45);
   while(get_photo()>0xe7&&kbhit()==0x00)
   {}
   photo_tracking_pid1(p,i,d,v,il);
   stop();
   photo_off();
}

photo_tracking_pid1(float kp,float ki,float kd,float v,float il)
{
   float  tracking_speed;
   int photo_center=0x7f;
   int photo_data;
   float  main_zeta1;
   float  er_new,er_old,er_sum,er_sum_max;
   float  fb;
   er_sum_max=il;
/* kp = 1;ki = 1;kd = 5000;*/
   kp =  kp/10000;
   ki  = ki/10000;
   kd =  kd/10000;
   tracking_speed = (v/(speed_ratio*22.55));
   main_zeta1=er_sum=er_new=er_old=0;
   while(kbhit()==0x00)
   {
      er_new=(float)(get_photo()-photo_center);
      er_sum=er_sum+er_new;
      if(er_sum> er_sum_max)er_sum =  er_sum_max;
      if(er_sum<-er_sum_max)er_sum = -er_sum_max;
      fb=(kp*er_new+ki*er_sum+kd*(er_new-er_old));
      main_zeta1=main_zeta1+fb;
      go_zeta(tracking_speed,main_zeta1);
      er_old=er_new;
   }
}

/************ PRO_CUT.C ***********/

void pro_main(void);
void print_pro_cut_menu(float p,float v,int ul,int ll,float dg1,float dg2,int c1,float dg3,int c2);
void set_p_pro(float *r);
void set_v_pro(float *v);
void set_ul_pro(int *ul);
void set_ll_pro(int *ll);
```

เอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
void set_dg1_pro(float *dg1);
void set_dg2_pro(float *dg2);
void set_c1_pro(int *c1);
void set_dg3_pro(float *dg3);
void set_c2_pro(int *c2);
void cut_pro(float p,float v,int ul,int ll,float dg1,float dg2,int c1,float dg3,int c2);
void photo_tracking_pro1(float p,float v,int ul,int ll,float dg1,float dg2,int c1,float dg3,int c2);

void pro_main(void)
{
   int process;
   static float p,v,dg1,dg2,dg3;
   static int  ul,ll,c1,c2;
   update_data_code=NO;
   data_exist_code=NO;
   clrscr();
   v=100;p=0.250;ul=0xe0;ll=0x20;dg1=0.2;dg2=0.5;c1=2;dg3=2.0;c2=10;
   print_pro_cut_menu(p,v,ul,ll,dg1,dg2,c1,dg3,c2);
   process = ON;
   while(process==ON)
   switch(get_key())
   {
      case F1 : v=100;p=0.250;ul=0xe0;ll=0x20;dg1=0.2;dg2=0.5;c1=2;dg3=2.0;c2=10;
                print_pro_cut_menu(p,v,ul,ll,dg1,dg2,c1,dg3,c2);break;
      case F2 : set_p_pro(&p);  print_pro_cut_menu(p,v,ul,ll,dg1,dg2,c1,dg3,c2);  break;
      case F5 : set_v_pro(&v);  print_pro_cut_menu(p,v,ul,ll,dg1,dg2,c1,dg3,c2);  break;
      case F6 : cut_pro(p,v,ul,ll,dg1,dg2,c1,dg3,c2);  print_pro_cut_menu(p,v,ul,ll,dg1,dg2,c1,dg3,c2); break;
      case F8 : process=OFF;break;
      case 'a': ;case 'A': set_ul_pro(&ul);
      print_pro_cut_menu(p,v,ul,ll,dg1,dg2,c1,dg3,c2);break;
      case 'b': ;case 'B': set_ll_pro(&ll);
      print_pro_cut_menu(p,v,ul,ll,dg1,dg2,c1,dg3,c2);break;
      case 'c': ;case 'C': set_dg1_pro(&dg1);
      print_pro_cut_menu(p,v,ul,ll,dg1,dg2,c1,dg3,c2);break;
      case 'd': ;case 'D': set_dg2_pro(&dg2);
      print_pro_cut_menu(p,v,ul,ll,dg1,dg2,c1,dg3,c2);break;
      case 'e': ;case 'E': set_c1_pro(&c1);
      print_pro_cut_menu(p,v,ul,ll,dg1,dg2,c1,dg3,c2);break;
      case 'f': ;case 'F': set_dg3_pro(&dg3);
      print_pro_cut_menu(p,v,ul,ll,dg1,dg2,c1,dg3,c2);break;
      case 'g': ;case 'G': set_c2_pro(&c2);
      print_pro_cut_menu(p,v,ul,ll,dg1,dg2,c1,dg3,c2);break;
   }
}
void print_pro_cut_menu(float p,float v,int ul,int ll,float dg1,float dg2,int c1.float dg3,int c2)
{
   clrscr();
   printf("\n\n\n\n\n");
   printf("\n                [ F1 ]  RESET  ");
printf("\n\n      P       = 0.250 V       = 100  ");
   printf("\n      UPPER LIMIT = 0xE0  LOWER LIMIT = 0x20 ");
   printf("\n      DEGREE 1    = 0.2   DEGREE 2    =  0.5   COUNT 1 =  2");
   printf("\n      DRGREE 3    = 2.0   COUNT 2     =  10  ");
printf("\n\n\n               [ F2 ]  SET P ( P = %4.3f )",p);
   printf("\n                [ F5 ]  SET V ( V = %4.3f m.m./minute)",v);
   printf("\n                [ F6 ]  CUT  ");
   printf("\n                [ F8 ]  QUIT ");
printf("\n\n   [ A ] UPPER LIMIT = 0x%2x  [ B ] LOWER LIMIT = 0x%2x ",ul,ll);
printf("\n\n   [ C ] DEGREE 1    = %2.2f  [ D ] DRGREE 2    = %2.2f [ E ] COUNT 1 = %3d ",dg1,dg2,c1);
printf("\n\n   [ F ] DEGREE 3    = %2.2f  [ G ] COUNT  2    = %2d ",dg3,c2);
   printf("\n\n\n");
}

void set_p_pro(float *p)
{ printf("\r p = ");scanf("%f",p);}
void set_v_pro(float *v)
{ printf("\r v = ");scanf("%f",v);}
void set_ul_pro(int *ul)
{ printf("\r ul = ");scanf("%x",ul);}
void set_ll_pro(int *ll)
{ printf("\r ll = ");scanf("%x",ll);}
void set_dg1_pro(float *dg1)
{ printf("\r dg1 = ");scanf("%f",dg1);}
void set_dg2_pro(float *dg2)
{ printf("\r dg2 = ");scanf("%f",dg2);}
void set_c1_pro(int *c1)
{  printf("\r c1 = ");scanf("%d",c1);}
void set_dg3_pro(float *dg3)
{ printf("\r dg3 = ");scanf("%f",dg3);}
void set_c2_pro(int *c2)
{ printf("\r c2 = ");scanf("%d",c2);}
void cut_pro(float p,float v,int ul,int ll,float dg1,float dg2,int c1,float dg3,int c2)
{
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    int count=0;
    init_port();
    photo_on();
    printf("\n\n\n");
    while(get_photo()<250&&kbhit()==0x00&&++count<=21);
    go_to(10,10,100,100);
    go_zeta(10,45);
    while(get_photo()>0xe7&&kbhit()==0x00)
    {}
    photo_tracking_pro1(p,v,ul,ll,dg1,dg2,c1,dg3,c2);
    stop();
    photo_off();
}
void photo_tracking_pro1(float p,float v,int ul,int ll,float dg1,float dg2,int c1,float dg3,int c2)
{
    float  tracking_speed;
    int photo_center=0x7f;
    int photo_data;
    int inc1,inc2,dec1,dec2,count1;
    float  main_zeta1,main_zeta2;

    main_zeta1=main_zeta2=0;
    tracking_speed = (v/(speed_ratio*22.55));
    inc1=inc2=dec1=dec2=count1=0;
    while(kbhit()==0x00)
    {
       photo_data=get_photo();
       if(photo_data<ll)
       {
          inc1=0;
          dec1++;
          if(dec1>c1)
             main_zeta2=main_zeta2-dg2,count1=0,dec1=1,dec2++;
          else
             main_zeta2=main_zeta2-dg1;
          if(dec2>c2)
             main_zeta2=main_zeta2-dg3,dec1=0,dec2=0;
       }
       if(photo_data>ul)
       {
          inc1++;
          dec1=0;
          if(inc1>c1)
             main_zeta2=main_zeta2+dg2,count1=0,inc1=1,inc2++;
          else
             main_zeta2=main_zeta2+dg1;
          if(inc2>c2)
             main_zeta2=main_zeta2+dg3,inc1=0,inc2=0;
       }
       if(photo_data>5f&&photo_data<9f)
          if(count1++>5)inc1=dec1=inc2=dec2=0;
       main_zeta1=main_zeta2-((float)(photo_center-photo_data)*p);
       go_zeta(tracking_speed,main_zeta1);
    }
}
□
/************ MIR_MAIN.C ***********/

void mir_main(void);
void print_mir_menu(float v,int upr,int lor,float dgr,int c1,float dg1);
void set_v_mir(float *v);
void set_upr_mir(int *upr);
void set_lor_mir(int *lor);
void set_dgr_mir(float *dgr);
void set_c1_mir(int *c1);
void set_dg1_mir(float *dg1);
void cut_mir(float v,int upr,int lor,float dgr,int c1,float dg1);
void photo_tracking_mir1(float v,int upr,int lor,float dgr,int c1,float dg1);

void mir_main(void)
{
    int process;
    static float v,dgr,dg1;
    static int upr,lor,c1;
    update_data_code=NO;
    data_exist_code=NO;
    clrscr();
    if(v==0){v=100;upr=0x75;lor=0x73;dgr=45;c1=20;dg1=20;}
    print_mir_menu(v,upr,lor,dgr,c1,dg1);
    process = ON;
    while(process==ON)
    switch(get_key())
    {
       case F5 : set_v_mir(&v);  print_mir_menu(v,upr,lor,dgr,c1,dg1);  break;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        case F6 : cut_mir(v,upr,lor,dgr,c1,dg1);print_mir_menu(v,upr,lor,dgr,c1,dg1); break;
        case F8 : process=OFF; break;
        case 'a': ;case 'A': set_upr_mir(&upr);print_mir_menu(v,upr,lor,dgr,c1,dg1); break;
        case 'b': ;case 'B': set_lor_mir(&lor);print_mir_menu(v,upr,lor,dgr,c1,dg1); break;
        case 'c': ;case 'C': set_dgr_mir(&dgr);print_mir_menu(v,upr,lor,dgr,c1,dg1); break;
        case 'd': ;case 'D': set_c1_mir(&c1);  print_mir_menu(v,upr,lor,dgr,c1,dg1); break;
        case 'e': ;case 'E': set_dg1_mir(&dg1);print_mir_menu(v,upr,lor,dgr,c1,dg1); break;
    }
}
void print_mir_menu(float v,int upr,int lor,float dgr,int c1,float dg1)
{
    clrscr();
    printf("\n\n");
    printf("\n\n");
    printf("\n\n");
    printf("\n              [ F5 ]  SET V ( V = %4.3f m.m./minute)",v);
    printf("\n              [ F6 ]  CUT  ");
    printf("\n              [ F8 ]  QUIT    ");
printf("\n\n              [ A ]  UPPER  REFRACT   = 0x%2x ",upr);
    printf("\n              [ B ]  LOWER  REFRACT   = 0x%2x ",lor);
    printf("\n              [ C ]  DEGREE REFRACT   = %3.2f ",dgr);
    printf("\n              [ D ]  COUNT  1        = %d   ",c1);
    printf("\n              [ E ]  DEGREE REFRACT 1 = %3.2f ",dg1);
    printf("\n\n");
}

void set_v_mir(float *v)
{ printf("\r v = ");scanf("%f",v);}
void set_upr_mir(int *upr)
{ printf("\r upper refract = ");scanf("%x",upr);}
void set_lor_mir(int *lor)
{ printf("\r lower refract = ");scanf("%x",lor);}
void set_dgr_mir(float *dgr)
{ printf("\r degree refract = ");scanf("%x",dgr);}
void set_c1_mir(int *c1)
{  printf("\r count 1 = ");scanf("%d",c1);}
void set_dg1_mir(float *dg1)
{ printf("\r degree refract 1 = ");scanf("%f",dg1);}

void cut_mir(float v,int upr,int lor,float dgr,int c1,float dg1)
{
    int count=0;
    init_port();
    photo_on();
    printf("\n\n\n");
    while(get_photo()<250&&kbhit()==0x00&&++count<=21);
    go_to(10,10,100,100);
    go_zeta(10,45);
    while(get_photo()>0xe7&&kbhit()==0x00)
    {}
    photo_tracking_mir1(v,upr,lor,dgr,c1,dg1);
    stop();
    photo_off();
}

void photo_tracking_mir1(float v,int upr,int lor,float dgr,int c1,float dg1)
{
    float  tracking_speed;
    int photo_data,count,process;
    long old_x,old_y,now_x,now_y,dl;
    float  main_zeta1,main_zeta2;
    tracking_speed = (v/(speed_ratio*22.55));
    printf("speed_ratio = %f tracking_speed = %d ",v,tracking_speed);
    main_zeta1=main_zeta2=0;
    count =  0;
    process = ON;
    while(kbhit()==0x00&&process==ON)
    {
        photo_data=get_photo();
        if(photo_data<lor)
        {
            main_zeta2=main_zeta1-dgr;
            count--;
        }
        if(photo_data>upr)
        {
            main_zeta2=main_zeta1+dgr;
            count++;
        }
        if(count> c1)main_zeta1=main_zeta1+dg1,count=0;
        if(count<-c1)main_zeta1=main_zeta1-dg1,count=0;
        go_zeta(tracking_speed,main_zeta2);
    }
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรมบันทึกลายเส้น

```
/****************** RECOED.C ********************/
void record_menu(void);
void print_record_menu(void);
int  check_start_point(long now_x,long now_y);
int  check_record_point(long old_x,long old_y,long now_x,long now_y,int dr);
int  check_end_point(long old_x,long old_y.long now_x,long now_y);

#include<pid_reco.c>
#include<pro_reco.c>
#include<mir_reco.c>

void record_menu()
{
   int process,i;
   update_data_code=NO;
   data_exist_code=NO;
   clrscr();
   print_record_menu();
   process = ON;
   while(process==ON)
   switch(get_key())
   {
      case F1 : break;
      case F2 : pid_rec_menu();  print_record_menu();  break;
      case F3 : pro_rec_menu();  print_record_menu();  break;
      case F4 : mir_rec_menu();  print_record_menu();  break;
      case F8 : process=OFF;
   }
}
void print_record_menu(void)
{
   clrscr();
   printf("\n\n");
   printf("\n              WELLCOME TO DATA RECORDING MODE ");
   printf("\n\n");
   printf("\n              PLEASE SELECT FUNCTION KEY TO USE ");
   printf("\n\n");
   printf("\n              [ F1 ]  HELP  ");
   printf("\n              [ F2 ]  PID MODE ");
   printf("\n              [ F3 ]  P AND PROGAME MODE ");
   printf("\n              [ F4 ]  MIRROR AND PROGAME MODE ");
   printf("\n              [ F8 ]  QUIT      ");
   printf("\n\n\n\n\n\n");
}

int  check_start_point(long now_x,long now_y)
{
   long dx,dy;
   dx = fabs((X[0]-now_x));
   dy = fabs((Y[0]-now_y));
   if(dx>3000||dy>3000)
   {
      X[0]=now_x;
      Y[0]=now_y;
      return ON;
   }
   else
      return OFF;
}

int check_record_point(long old_x,long old_y,long now_x,long now_y,int dr)
{
   long dx,dy;
   dx=labs(old_x-now_x);
   dy=labs(old_y-now_y);
   /*dr=70;*/
   if(dx>dr||dy>dr)
      return OK;
   return NO;
}

int check_end_point(long old_x,long old_y,long now_x,long now_y)
{
   long dx,dy;
   dx=labs(old_x-now_x);
   dy=labs(old_y-now_y);
   if(dx>1000||dy>1000)
      return OK;
   return NO;
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
/************ PID_RECO.C ************/

void pid_rec_menu(void);
void print_pid_rec_menu(float p,float i,float d,float v,float il,int ur,int lr,int dr);
set_p_pid_rec(float *p);
set_i_pid_rec(float *i);
set_d_pid_rec(float *d);
set_v_pid_rec(float *v);
set_il_pid(float *il);
set_ur_pid_rec(int *ur);
set_lr_pid_rec(int *lr);
set_dr_pid_rec(int *dr);
process_record_pid(float p,float i,float d,float v,float il,int ur,int lr,int dr);
photo_record_pid(float kp,float ki,float kd,float v,float il,int ur,int lr,int dr);

void pid_rec_menu(void)
{
   int process;
   static float p,i,d,v,il;
   static int ur,lr,dr;
   update_data_code=ON;
   data_exist_code=ON;
   clrscr();
   if(v==0){
   p=10;i=1;d=5000;v=100;il=1000;
   ur=0xf0;lr=0x10;dr=70;}
   print_pid_rec_menu(p,i,d,v,il,ur,lr,dr);
   process = ON;
   while(process==ON)
   switch(get_key())
   {
      case F1 : p=1;i=1;d=5000;v=100;il=1000;ur=0xf0;lr=0x10;dr=70;
                print_pid_rec_menu(p,i,d,v,il,ur,lr,dr);break;
      case F2 : set_p_pid_rec(&p); print_pid_rec_menu(p,i,d,v,il,ur,lr,dr): break;
      case F3 : set_i_pid_rec(&i); print_pid_rec_menu(p,i,d,v,il,ur,lr,dr): break;
      case F4 : set_d_pid_rec(&d); print_pid_rec_menu(p,i,d,v,il,ur,lr,dr): break;
      case F5 : set_v_pid_rec(&v); print_pid_rec_menu(p,i,d,v,il,ur,lr,dr): break;
      case F6 : process_record_pid(p,i,d,v,il,ur,lr,dr);
                print_pid_rec_menu(p,i,d,v,il,ur,lr,dr); break;
      case F7 : set_il_pid_rec(&il);print_pid_rec_menu(p,i,d,v,il,ur,lr,dr): break;
      case F8 : process=OFF; break;
      case 'h': ;case 'H': set_lr_pid_rec(&ur);print_pid_rec_menu(p,i,d,v,il,ur,lr,dr);break;
      case 'i': ;case 'I': set_ur_pid_rec(&lr);print_pid_rec_menu(p,i,d,v,il,ur,lr,dr);break;
      case 'j': ;case 'J': set_dr_pid_rec(&dr);print_pid_rec_menu(p,i,d,v,il,ur,lr,dr);break;
   }
}
void print_pid_rec_menu(float p,float i,float d,float v,float il,int ur,int lr,int dr)
{
   clrscr();
   printf("\n\n");
   printf("\n\n");
   printf("\n\n");
   printf("\n              [ F1 ] RESET PID (P=10 I=1 D=5000)");
   printf("\n              [ F2 ] SET P ( P = %4.3f )",p);
   printf("\n              [ F3 ] SET I ( I = %4.3f )",i);
   printf("\n              [ F4 ] SET D ( D = %4.3f )",d);
   printf("\n              [ F5 ] SET V ( V = %4.3f m.m./minute)",v);
   printf("\n              [ F6 ] CUT  ");
   printf("\n              [ F7 ] SET I LIMIT ( IL = %4.3f )",il);
   printf("\n              [ F8 ] QUIT     ");
printf("\n\n            [ H ] UPPER RECORD    = 0x%2x ",ur);
   printf("\n              [ I ] LOWER RECORD    = 0x%2x ",lr);
   printf("\n              [ J ] DISTANCE RECORD  =  %d ",dr);
   printf("\n\n\n\n\n\n");
}
set_p_pid_rec(float *p)
{ printf("\r p = ");scanf("%f",p);}
set_i_pid_rec(float *i)
{ printf("\r i = ");scanf("%f",i);}
set_d_pid_rec(float *d)
{ printf("\r d = ");scanf("%f",d);}
set_v_pid_rec(float *v)
{ printf("\r v = ");scanf("%f",v);}
set_il_pid_rec(float *il)
{ printf("\r il = ");scanf("%f",il);}
set_ur_pid_rec(int *ur)
{ printf("\r ur = ");scanf("%x",ur);}
set_lr_pid_rec(int *lr)
{ printf("\r lr = ");scanf("%x",lr);}
set_dr_pid_rec(int *dr)
{ printf("\r dr = ");scanf("%x",dr);}
process_record_pid(float p,float i,float d,float v,float il,int ur,int lr,int dr)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
{
    int count=0;
    clear_data();
    init_port();
    photo_on();
    printf("\n\n\n");
    while(get_photo()<250&&kbhit()==0x00&&++count<=21);
    go_to(10,10,100,100);
    go_zeta(10,45);
    while(get_photo()>0xe7&&kbhit()==0x00)
    {}
    photo_record_pid(p,i,d,v,il,ur,lr,dr);
    stop();
    photo_off();
}

photo_record_pid(float kp,float ki,float kd,float v,float il,int ur,int lr,int dr)
{
    float  tracking_speed;
    int photo_center=0x7f;
    int photo_data;
    float  main_zeta1;
    float  er_new,er_old,er_sum,er_sum_max;
    float  fb;
    long old_x,old_y,now_x,now_y;
    int  record,i,process;
    er_sum_max=il;
/*  kp = 1;ki = 1;kd = 5000;*/
    kp =  kp/10000;
    ki  = ki/10000;
    kd =  kd/10000;
    tracking_speed = (v/(speed_ratio*22.55));
    main_zeta1=er_sum=er_new=er_old=0;
    record = OFF; i = 1;  process=ON;
    old_x = now_x = X[0] =    get_x();
    old_y = now_y = Y[0] =    get_y();
    printf("\n %ld %ld\n",now_x,now_y);
    while(kbhit()==0x00&&process==ON)
    {
       photo_data=get_photo();
       now_y=get_y();
       now_x=get_x();
       if(record==OFF)
       {
          record = check_start_point(now_x,now_y);
          old_x=now_x;
          old_y=now_y;
       }
       else
       {
          if((photo_data>lr)&&(photo_data<ur))
          if(check_record_point(old_x,old_y,now_x,now_y,dr)==OK)
          {
             X[i]   = now_x;
             Y[i]   = now_y;
             Y[i]   = now_y;
             old_x  = now_x;
             old_y  = now_y;
             i++;
             if(i>MAX)
                printf("\nout of data memory"),process = OFF;
          }
          if(i>50&&check_end_point(X[0],Y[0],now_x,now_y)==NO)
             printf("\nprocess complete"),process = OFF;
       }
       er_new=(float)(photo_data-photo_center);
       er_sum=er_sum+er_new;
       if(er_sum> er_sum_max) er_sum =  er_sum_max;
       if(er_sum<-er_sum_max) er_sum = -er_sum_max;
       fb=(kp*er_new+ki*er_sum+kd*(er_new-er_old));
       main_zeta1=main_zeta1+fb;
       go_zeta(tracking_speed,main_zeta1);
       er_old=er_new;
    }
}

/************ PRO_RECO.C ***********/

void pro_rec_menu(void);
void rec_pro_menu(float p,float v,int ul,int ll,float dg1,float dg2,int c1,float dg3,int c2,int ur,int lr,int dr);
void set_p_rec_pro(float *r);
void set_v_rec_pro(float *v);
void set_ul_rec_pro(int *ul);
void set_ll_rec_pro(int *ll);
```

เอกสารนี้สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
void set_dg1_rec_pro(float *dg1);
void set_dg2_rec_pro(float *dg2);
void set_c1_rec_pro(int *c1);
void set_dg3_rec_pro(float *dg3);
void set_c2_rec_pro(int *c2);
void set_ur_rec_pro(int *ur);
void set_lr_rec_pro(int *lr);
void set_dr_rec_pro(int *dr);
void rec_pro(float p,float v,int ul,int ll,float dg1,float dg2,int c1,float dg3,int c2,int ur,int lr,int dr);
void photo_recording_pro1(float p,float v,int ul,int ll,float dg1,float dg2,int c1,float dg3,int c2,int ur,int lr,int dr);

void pro_rec_menu(void)
{
   int process;
   static float p,v,dg1,dg2,dg3;
   static int  ul,ll,c1,c2,ur,lr,dr;
   update_data_code=ON;
   data_exist_code=ON;
   clrscr();
   v=100;p=0.250;ul=0xe0;ll=0x20;dg1=0.2;dg2=0.5;c1=2;dg3=2.0;c2=10;lr=0x10,ur=0xf0;dr=70;
   rec_pro_menu(p,v,ul,ll,dg1,dg2,c1,dg3,c2,ur,lr,dr);
   process = ON;
   while(process==ON)
   switch(get_key())
   {
      case F1 : v=100;p=0.250;ul=0xe0:ll=0x20;dg1=0.2;dg2=0.5;c1=2;dg3=2.0;c2=10;lr=0x10,ur=0xf0;dr=70;
                rec_pro_menu(p,v,ul,ll,dg1,dg2,c1,dg3,c2,ur,lr,dr);break;
      case F2 : set_p_rec_pro(&p);  rec_pro_menu(p,v,ul,ll,dg1,dg2,c1,dg3,c2,ur,lr,dr);  break;
      case F5 : set_v_rec_pro(&v);  rec_pro_menu(p,v,ul,ll,dg1,dg2,c1,dg3,c2,ur,lr,dr);  break;
      case F6 : rec_pro(p,v,ul,ll,dg1,dg2,c1,dg3,c2,ur,lr,dr);  rec_pro_menu(p,v,ul,ll,dg1,dg2,c1,dg3,c2,ur,lr,dr); break;
      case F8 : process=OFF;break;
      case 'a': ;case 'A': set_ul_rec_pro(&ul);
      rec_pro_menu(p,v,ul,ll,dg1,dg2,c1,dg3,c2,ur,lr,dr);break;
      case 'b': ;case 'B': set_ll_rec_pro(&ll);
      rec_pro_menu(p,v,ul,ll,dg1,dg2,c1,dg3,c2,ur,lr,dr);break;
      case 'c': ;case 'C': set_dg1_rec_pro(&dg1);
      rec_pro_menu(p,v,ul,ll,dg1,dg2,c1,dg3,c2,ur,lr,dr);break;
      case 'd': ;case 'D': set_dg2_rec_pro(&dg2);
      rec_pro_menu(p,v,ul,ll,dg1,dg2,c1,dg3,c2,ur,lr,dr);break;
      case 'e': ;case 'E': set_c1_rec_pro(&c1);
      rec_pro_menu(p,v,ul,ll,dg1,dg2,c1,dg3,c2,ur,lr,dr);break;
      case 'f': ;case 'F': set_dg3_rec_pro(&dg3);
      rec_pro_menu(p,v,ul,ll,dg1,dg2,c1,dg3,c2,ur,lr,dr);break;
      case 'g': ;case 'G': set_c2_rec_pro(&c2);
      rec_pro_menu(p,v,ul,ll,dg1,dg2,c1,dg3,c2,ur,lr,dr);break;
      case 'h': ;case 'H': set_lr_rec_pro(&lr);
      rec_pro_menu(p,v,ul,ll,dg1,dg2,c1,dg3,c2,ur,lr,dr);break;
      case 'i': ;case 'I': set_ur_rec_pro(&ur);
      rec_pro_menu(p,v,ul,ll,dg1,dg2,c1,dg3,c2,ur,lr,dr);break;
      case 'j': ;case 'J': set_dr_rec_pro(&dr);
      rec_pro_menu(p,v,ul,ll,dg1,dg2,c1,dg3,c2,ur,lr,dr);break;
   }
}
void rec_pro_menu(float p,float v,int ul,int ll,float dg1,float dg2,int c1,float dg3,int c2,int ur,int lr,int dr)
{
   clrscr();
   printf("\n\n");
   printf("\n                [ F1 ]  RESET  ");
printf("\n\n      P       = 0.250 V        = 100  ");
   printf("\n      UPPER LIMIT = 0xE0  LOWER LIMIT = 0x20 ");
   printf("\n      DEGREE 1    = 0.2   DEGREE 2    =  0.5   COUNT 1 =  2");
   printf("\n      DRGREE 3    = 2.0   COUNT 2     =  10  ");
printf("\n\n                [ F2 ]  SET P ( P = %4.3f )",p);
   printf("\n                [ F5 ]  SET V ( V = %4.3f m.m./minute)",v);
   printf("\n                [ F6 ]  CUT  ");
   printf("\n                [ F8 ]  QUIT ");
printf("\n\n   [ A ] UPPER LIMIT = 0x%2x  [ B ] LOWER LIMIT = 0x%2x ",ul,ll);
   printf("\n   [ C ] DEGREE 1    = %2.2f  [ D ] DRGREE 2    = %2.2f [ E ] COUNT 1 = %3d ",dg1,dg2,c1);
   printf("\n   [ F ] DEGREE 3    = %2.2f  [ G ] COUNT 2     = %2d ",dg3,c2);
printf("\n\n   [ H ] UPPER RECORD     = 0x%2x ",ur);
   printf("\n   [ I ] LOWER RECORD     = 0x%2x ",lr);
   printf("\n   [ J ] DISTANCE RECORD  =  %d ",dr);
   printf("\n\n");
}

void set_p_rec_pro(float *p)
{ printf("\r p = ");scanf("%f",p);}
void set_v_rec_pro(float *v)
{ printf("\r v = ");scanf("%f",v);}
void set_ul_rec_pro(int *ul)
{ printf("\r ul = ");scanf("%x",ul);}
void set_ll_rec_pro(int *ll)
{ printf("\r ll = ");scanf("%x",ll);}
void set_dg1_rec_pro(float *dg1)
```

เอกสารนี้สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
{ printf("\r dg1 = ");scanf("%f",dg1);}
void set_dg2_rec_pro(float *dg2)
{ printf("\r dg2 = ");scanf("%f",dg2);}
void set_c1_rec_pro(int *c1)
{  printf("\r c1 = ");scanf("%d",c1);}
void set_dg3_rec_pro(float *dg3)
{ printf("\r dg3 = ");scanf("%f",dg3);}
void set_c2_rec_pro(int *c2)
{ printf("\r c2 = ");scanf("%d",c2);}
void set_ur_rec_pro(int *ur)
{ printf("\r ur = ");scanf("%d",ur);}
void set_lr_rec_pro(int *lr)
{ printf("\r lr = ");scanf("%d",lr);}
void set_dr_rec_pro(int *dr)
{ printf("\r dr = ");scanf("%d",dr);}

void rec_pro(float p,float v,int ul,int ll,float dg1,float dg2,int c1,float dg3,int c2,int ur,int lr,int dr)
{
   int count=0;
   clear_data();
   init_port();
   photo_on();
   printf("\n\n\n");
   while(get_photo()<250&&kbhit()==0x00&&++count<=21);
   go_to(10,10,100,100);
   go_zeta(10,45);
   while(get_photo()>0xe7&&kbhit()==0x00)
   {}
   photo_recording_pro1(p,v,ul,ll,dg1,dg2,c1,dg3,c2,ur,lr,dr);
   stop();
   photo_off();
}

void photo_recording_pro1(float p,float v,int ul,int ll,float dg1,float dg2,int c1,float dg3,int c2,int ur,int lr,int dr)
{
   float  tracking_speed;
   int photo_center=0x7f;
   int photo_data;
   int inc1,inc2,dec1,dec2,count1;
   int record,i,process;
   long old_x,old_y,now_x,now_y;
   float  main_zeta1,main_zeta2;
   record = OFF; i = 1; process = ON;
   main_zeta1=main_zeta2=0;
   tracking_speed = (v/(speed_ratio*22.55));
   inc1=inc2=dec1=dec2=count1=0;
   old_x = now_x = X[0] =   get_x();
   old_y = now_y = Y[0] =   get_y();
   printf("\n %ld %ld\n",now_x,now_y);
   while(kbhit()==0x00&&process==ON)
   {
      photo_data=get_photo();
      now_y=get_y();
      now_x=get_x();
      if(record==OFF)
      {
         record = check_start_point(now_x,now_y);
         old_x=now_x;
         old_y=now_y;
      }
      else
      {
         if(photo_data>lr/*0x10*/&&photo_data<ur/*0xf0*/)
         if(check_record_point(old_x,old_y,now_x,now_y,dr)==OK)
         {
            X[i]   = now_x;
            Y[i]   = now_y;
            Y[i]   = now_y;
            old_x  = now_x;
            old_y  = now_y;
            i++;
            if(i>MAX)
               printf("\nout of data memory"),process = OFF;
         }
         if(i>50&&check_end_point(X[0],Y[0],now_x,now_y)==NO)
            printf("\nprocess complete"),process = OFF;
      }
      if(photo_data<ll)
      {
         inc1=0;
         dec1++;
         if(dec1>c1)
            main_zeta2=main_zeta2-dg2,count1=0,dec1=1,dec2++;
         else
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            main_zeta2=main_zeta2-dg1;
          if(dec2>c2)
            main_zeta2=main_zeta2-dg3,dec1=0,dec2=0;
        }
        if(photo_data>ul)
        {
          inc1++;
          dec1=0;
          if(inc1>c1)
            main_zeta2=main_zeta2+dg2,count1=0,inc1=1,inc2++;
          else
            main_zeta2=main_zeta2+dg1;
          if(inc2>c2)
            main_zeta2=main_zeta2+dg3,inc1=0,inc2=0;
        }
        if(photo_data>5f&&photo_data<9f)
          if(count1++>5)inc1=dec1=inc2=dec2=0;
        main_zeta1=main_zeta2-((float)(photo_center-photo_data)*p);
        go_zeta(tracking_speed,main_zeta1);
      }
  }
  ❑
  /************ MIR_RECO.C ***********/

  void mir_rec_menu(void);
  void print_mir_rec_menu(float v,int upr,int lor,float dgr,int c1,float dg1,int ur,int lr,int dr);
  void set_v_mir_rec(float *v);
  void set_upr_mir_rec(int *upr);
  void set_lor_mir_rec(int *lor);
  void set_dgr_mir_rec(float *dgr);
  void set_c1_mir_rec(int *c1);
  void set_dg1_mir_rec(float *dg1);
  void set_ur_mir_rec(int *ur);
  void set_lr_mir_rec(int *lr);
  void set_dr_mir_rec(int *dr);
  void cut_mir_rec(float v,int upr,int lor,float dgr,int c1,float dg1,int ur,int lr,int dr);
  void photo_tracking_mir_rec(float v,int upr,int lor,float dgr,int c1,float dg1,int ur,int lr,int dr);

  void mir_rec_menu(void)
  {
    int process;
    static float v,dgr,dg1;
    static int upr,lor,c1,ur,lr,dr;
    update_data_code=ON;
    data_exist_code=ON;
    clrscr();
    if(v==0){v=100;upr=0x75;lor=0x73;dgr=45;c1=20;dg1=20;ur=0x75;lr=0x73;dr=70;}
    print_mir_rec_menu(v,upr,lor,dgr,c1,dg1,ur,lr,dr);
    process = ON;
    while(process==ON)
    switch(get_key())
    {
      case F5 : set_v_mir_rec(&v);
                print_mir_rec_menu(v,upr,lor,dgr,c1,dg1,ur,lr,dr); break;
      case F6 : cut_mir_rec(v,upr,lor,dgr,c1,dg1,ur,lr,dr);
                print_mir_rec_menu(v,upr,lor,dgr,c1,dg1,ur,lr,dr); break;
      case F8 : process=OFF; break;
      case 'a': ;case 'A': set_upr_mir_rec(&upr);
                print_mir_rec_menu(v,upr,lor,dgr,c1,dg1,ur,lr,dr); break;
      case 'b': ;case 'B': set_lor_mir_rec(&lor);
                print_mir_rec_menu(v,upr,lor,dgr,c1,dg1,ur,lr,dr); break;
      case 'c': ;case 'C': set_dgr_mir_rec(&dgr);
                print_mir_rec_menu(v,upr,lor,dgr,c1,dg1,ur,lr,dr); break;
      case 'd': ;case 'D': set_c1_mir_rec(&c1);
                print_mir_rec_menu(v,upr,lor,dgr,c1,dg1,ur,lr,dr); break;
      case 'e': ;case 'E': set_dg1_mir_rec(&dg1);
                print_mir_rec_menu(v,upr,lor,dgr,c1,dg1,ur,lr,dr); break;
      case 'h': ;case 'H': set_ur_mir_rec(&ur);
                print_mir_rec_menu(v,upr,lor,dgr,c1,dg1,ur,lr,dr); break;
      case 'i': ;case 'I': set_lr_mir_rec(&lr);
                print_mir_rec_menu(v,upr,lor,dgr,c1,dg1,ur,lr,dr); break;
      case 'g': ;case 'J': set_dr_mir_rec(&dr);
                print_mir_rec_menu(v,upr,lor,dgr,c1,dg1,ur,lr,dr); break;
    }
  }
  void print_mir_rec_menu(float v,int upr,int lor,float dgr,int c1,float dg1,int ur,int lr,int dr)
  {
    clrscr();
    printf("\n\n");
    printf("\n\n");
    printf("\n\n");
    printf("\n          [ F5 ]  SET V ( V = %4.3f m.m./minute)",v);
    printf("\n          [ F6 ]  CUT ");
    printf("\n          [ F8 ]  QUIT    ");
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    printf("\n\n              [ A ] UPPER  REFRACT  = 0x%2x ",upr);
      printf("\n              [ B ] LOWER  REFRACT  = 0x%2x ",lor);
      printf("\n              [ C ] DEGREE REFRACT  = %3.2f ",dgr);
      printf("\n              [ D ] COUNT  1        = %d   ",c1);
      printf("\n              [ E ] DEGREE REFRACT 1 = %3.2f ",dg1);
    printf("\n\n             [ H ] UPPER RECORD    = 0x%2x ",ur);
      printf("\n              [ I ] LOWER RECORD    = 0x%2x ",lr);
      printf("\n              [ J ] DISTANCE RECORD = %d ",dr);
      printf("\n\n");
    }

    void set_v_mir_rec(float *v)
    { printf("\r v = ");scanf("%f",v);}
    void set_upr_mir_rec(int *upr)
    { printf("\r upper refract = ");scanf("%x",upr);}
    void set_lor_mir_rec(int *lor)
    { printf("\r lower refract = ");scanf("%x",lor);}
    void set_dgr_mir_rec(float *dgr)
    { printf("\r degree refract = ");scanf("%x",dgr);}
    void set_c1_mir_rec(int *c1)
    {  printf("\r count 1 = ");scanf("%d",c1);}
    void set_dg1_mir_rec(float *dg1)
    { printf("\r degree refract 1 = ");scanf("%f",dg1);}
    void set_ur_mir_rec(int *ur)
    { printf("\r upper record = ");scanf("%x",ur);}
    void set_lr_mir_rec(int *lr)
    { printf("\r lower record = ");scanf("%x",lr);}
    void set_dr_mir_rec(int *dr)
    { printf("\r distance record = ");scanf("%x",dr);}

  void cut_mir_rec(float v,int upr,int lor,float dgr,int c1,float dg1,int ur,int lr,int dr)
  {
    int count=0;
    clear_data();
    init_port();
    photo_on();
    printf("\n\n\n");
    while(get_photo()<250&&kbhit()==0x00&&++count<=21);
    go_to(10,10,100,100);
    go_zeta(10,45);
    while(get_photo()>0xe7&&kbhit()==0x00)
    {}
    photo_tracking_mir_rec(v,upr,lor,dgr,c1,dg1,ur,lr,dr);
    stop();
    photo_off();
}

void photo_tracking_mir_rec(float v,int upr,int lor,float dgr,int c1,float dg1,int ur,int lr,int dr)
{
  float  record_speed;
  int photo_data,count,process;
  int record,i;
  long old_x,old_y,now_x,now_y,dl;
  float  main_zeta1,main_zeta2;
  main_zeta1=main_zeta2=0;
  record = count =  0;
  record_speed = (v/(speed_ratio*22.55));
  i = 1;
  process = ON;
  old_x = now_x = X[0] =   get_x();
  old_y = now_y = Y[0] =   get_y();
  printf("\n %ld %ld\n",now_x,now_y);
  while(kbhit()==0x00&&process==ON)
  {
    photo_data=get_photo();
    now_y=get_y();
    now_x=get_x();
    if(record==OFF)
    {
      record = check_start_point(now_x,now_y);
      old_x=now_x;
      old_y=now_y;
    }
    else
    {
      if(photo_data>lr&&photo_data<ur)
      if(check_record_point(old_x,old_y,now_x,now_y,dr)==OK)
      {
        X[i]  = now_x;
        Y[i]  = now_y;
        Y[i]  = now_y;
        old_x  = now_x;
        old_y  = now_y;
        i++;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            if(i>MAX)
                printf("\nout of data memory"),process = OFF;
        }
        if(i>50&&check_end_point(X[0],Y[0],now_x,now_y)==NO)
            printf("\nprocess complete"),process = OFF;
    }
    if(photo_data<lor)
    {
        main_zeta2=main_zeta1-dgr;
        count--;
    }
    if(photo_data>upr)
    {
        main_zeta2=main_zeta1+dgr;
        count++;
    }
    if(count> c1)main_zeta1=main_zeta1+dg1,count=0;
    if(count<-c1)main_zeta1=main_zeta1-dg1,count=0;
    go_zeta(record_speed,main_zeta2);
  }
}
```


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรมหลักสำหรับการแก้ไขข้อมูลที่ได้จากการบันทึกลายเส้น

```
/*    editmain.c     */
#define MENU_ON             1
#define MENU_OFF        0
#define OK                  1
#define  NO                   0
#define ON                  1
#define OFF                 0
#define MOVE                2
#define MAX         6000 /* limit at 7000 */

#include<key_def.c>
#include<define.c>
#include<graph.c>
#include<mouse.c>

int  get_key();
void show_data(void);
void print_menu(void);

long X[MAX],Y[MAX];
int  end_data;
int  update_data_code;
int  data_exist_code;
#include<lo_save.c>
#include<display.c>
#include<edit.c>
#include<menu.c>

int main()
{
    int process,i;
    update_data_code=NO;
    data_exist_code=NO;
    clrscr();
    print_menu();
    process = ON;
    while(process==ON)
    switch(get_key())
    {
        case F1 : show_data();    print_menu();  break;
        case F2 : save_data();    print_menu();  break;
        case F3 : load_data();    edit_data();   print_menu();  break;
/*      case F4 : record();       edit_data();   print_menu();  break;*/
        case F4 : show_data();    print_menu();  break;
        case F5 : edit_data();    print_menu();  break;
/*      case F7 : cut_menu();     print_menu();  break;*/
        case F8 : process=OFF;
    }
}

int get_key()
{
    int i;
    if(kbhit()==0x00)return 0;
    i=getch();
    if(i!=0x00)return i;
    i=0x0100+getch();
    return i;
}

void show_data(void)
{
    long x,y;
    int i;
    float ratiopicture,ratioscreen,ratio;
    long minx,miny;
    long cenx,ceny;
    long maxx,maxy;
    if(X[0]==0||Y[0]==0){putch(0x07);return;}
    init_graph();
    getmaxmin_data(&minx,&miny,&maxx,&maxy);
    cenx=(minx+maxx)/2;ceny=(miny+maxy)/2;
/*  x = start_x; y = start_y;*/
    x = X[0];    y = Y[0];
    ratiopicture = (float)(maxy-miny)/(float)(maxx-minx);
    ratioscreen  = (float)MaxY/(float)MaxX;
    if(ratiopicture>ratioscreen) ratio = (float)MaxY/(float)(maxy-miny);
    else                         ratio = (float)MaxX/(float)(maxx-minx);
    setcolor(BLACK);
    lineto(g_x(cenx,X[0],ratio),g_y(ceny,Y[0],ratio));
    setcolor(YELLOW);
    printf("\n x = %5.2f m.m.",(float)(maxx-minx)/step_ratio);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    printf("\n y = %5.2f m.m.",(float)(maxy-miny)/step_ratio);
    rectangle(0,0,MaxX,MaxY);
    i=0;
    setcolor(WHITE);
    while((X[i]!=0||Y[i]!=0)&&get_key()!=0x1b)
    {
       x=X[i];y=Y[i];
       lineto(g_x(cenx,x,ratio),g_y(ceny,y,ratio));
       i++;
    }
    setcolor(LIGHTRED);
    lineto(g_x(cenx,X[0],ratio),g_y(ceny,Y[0],ratio));
    i=getch();
    closegraph();
}

void print_menu(void)
{
    clrscr();
    printf("\n\n");
    printf("\n            WELLCOME TO USE ADVANCE CUTTING MACHINE ");
    printf("\n\n");
    printf("\n             PLEASE SELECT FUNCTION KEY TO USE ");
    printf("\n\n");
    printf("\n             [ F1 ]  HELP  ");
    printf("\n             [ F2 ]  SAVE DATA INTO DISK ");
    printf("\n             [ F3 ]  LOAD DATA FROM DISK ");
    printf("\n             [ F4 ]  SHOW DATA ON GRAPHICS SCREEN ");
    printf("\n             [ F5 ]  EDIT DATA ");
    printf("\n             [ F8 ]  QUIT     ");
    printf("\n\n\n\n\n\n");
}
```

โปรแกรมคำสั่งควบคุมการใช้ตัวชี้บนจอภาพ

```
/********************* MOUSE.C ********************/
#ifndef _MOUSE_H
#define _MOUSE_H
typedef struct{
            int class_;                         /*class of mouse */
            int x;                                /*pointer x */
            int y;                                /*pointer y */
            float x_coordinate;                 /*coordinate x */
            float y_coordinate;                 /*coordinate y */
            }InputMouseStruc,*InputMouse;
              /* define bit_mask of class of Message */
#define IE_LBUTTONUP      0X0001
#define IE_LBUTTONDOWN  0X0002
#define IE_LDBLCLICK      0X0003
#define IE_MBUTTONUP    0X0004
#define IE_MBUTTONDOWN  0X0005
#define IE_MDBLCLICK    0X0006
#define IE_RBUTTONUP    0X0007
#define IE_RBUTTONDOWN  0X0008
#define IE_RDBLCLICK    0X0009
#define IE_MOUSEMOVE        0X000A

#ifndef TRUE
#define TRUE  1
#define FALSE !TRUE
#endif


int mouse_reset(void);
void show_mouse_cursor(void);
void hide_mouse_cursor(void);

void get_mouse_position(int *button,int *hor,int *ver);
void set_mouse_position(int hor,int ver);

int get_mouse_button_pressed(int *button,int *hor,int *ver);
int get_mouse_button_releases(int *button,int *hor,int *ver);
void read_mouse_motion(int *hor,int *ver);

void set_mouse_horizontal_limit(int min,int max);
void set_mouse_vertical_limit(int min,int max);

void set_mouse_graphics_cursor(int hor,int ver,void far *mark);
void set_mouse_text_cursor(int cursor_type,int start,int stop);

void mouse_lightpen_on(void);
void mouse_lightpen_off(void);
void mouse_condition_off(int x1,int y1,int x2,int y2);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
void set_mouse_pixel_ratio(int hor,int ver);
void set_mouse_double_threshold(int threshold);

/* trap interrupt mouse driver */
void set_mouse_function(int mask,void interrupt(*proc)());

#endif

#include<dos.h>
int mouse_reset()
{
   _AX = 0x0000;
   geninterrupt(0x33);
   return(_AX)?(_BX):(_AX);
}
void show_mouse_cursor()
{
   _AX = 0X0001;
   geninterrupt(0x33);
}
void hide_mouse_cursor()
{
   _AX = 0x0002;
   geninterrupt(0x33);
}
void get_mouse_position(int *button,int *hor,int *ver)
{
   int savebx;
   _AX = 0x0003;
   geninterrupt(0x33);
   savebx = _BX;
   *button = savebx;
   *hor   = _CX;
   *ver    = _DX;
}
void set_mouse_position(int hor,int ver)
{
   _AX = 0x0004;
   _CX = hor;
   _DX = ver;
   geninterrupt(0x33);
}
int get_mouse_button_presser(int *button,int *hor,int *ver)
{
   int saveax,savebx;
   _AX = 0x0005;
   _BX = *button;
   geninterrupt(0x33);
   saveax = _AX;
   savebx = _BX;
   *button = saveax;
   *hor   = _CX;
   *ver    = _DX;
   return(savebx);
}
int get_mouse_button_releases(int *button,int *hor,int *ver)
{
   int saveax,savebx;
   _AX = 0x0006;
   _BX = *button;
  geninterrupt(0x33);
   saveax = _AX;
   savebx = _BX;
   *button = saveax;
   *hor   = _CX;
   *ver    = _DX;
   return(savebx);
}
void set_mouse_horizontal_limit(int min,int max)
{
   _AX = 0x0007;
   _CX = min*2;
   _DX = max*2;
   geninterrupt(0x33);
}
void set_mouse_vertical_limit(int min,int max)
{
   _AX = 0x0008;
   _CX = min;
   _DX = max;
   geninterrupt(0x33);
}
void set_mouse_graphics_cursor(int hor,int ver,void far *mark)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
{
    _BX = hor;
    _CX = ver;
    _DX = FP_OFF(mark);
    _ES = FP_SEG(mark); /* _ES use mov ax,[bp+?],mov ES,ax*/
    _AX = 0x0009;              /* _AX set after */
    geninterrupt(0x33);
}
void set_mouse_text_cursor(int cursor_type,int start,int stop)
{
    _AX = 0x000a;
    _BX = cursor_type;
    _CX = start;
    _DX = stop;
    geninterrupt(0x33);
}
void read_mouse_motion(int *hor,int *ver)
{
    /* ES:BX is used to pointer */
    _AX = 0x000b;
    geninterrupt(0x33);
    *hor = _CX;
    *ver = _DX;
}
void set_mouse_function(int mask,void interrupt(*proc)())
{
    _CX = mask;
    _DX = FP_OFF(proc);
    _ES = FP_SEG(proc);
    _AX = 0x000c;
    geninterrupt(0x33);
}
void mouse_lightpen_on()
{
    _AX = 0x000d;
    geninterrupt(0x33);
}
void mouse_lightpen_off()
{
    _AX = 0x000e;
    geninterrupt(0x33);
}
void set_mouse_pixel_ratio(int hor,int ver)
{
    _AX = 0x000f;
    _CX = hor;
    _DX = ver;
    geninterrupt(0x33);
}
void set_mouse_double_threshold(int threshold)
{
    _AX = 0x0013;
    _DX = threshold;
    geninterrupt(0x33);
}
void mouse_condition_off(int x1,int y1,int x2,int y2)
{
    _AX = 0x0010;
    _CX = x1;
    _DX = y1;
    _SI = x2; /* Turbo C save _SI and _DI autometic */
    _DI = y2;
    geninterrupt(0x33);
}

#include<stdio.h>
#include<dos.h>
#include<conio.h>
#include<graphics.h>
#include<alloc.h>

/*---------------------- internal routine -------------------------*/
int  InitMouse(int startx,int starty,int maxx,int maxy);
void CloseMouse();
void show_pointer(int x_cursor,int y_cursor,int function);
int  getuser(int *x_cursor,int *y_cursor,int *key_code);
int  get_mouse(int *x_cursor,int *y_cursor,int *code_mouse);

int InitMouse(int startx,int starty,int maxx,int maxy)
/*  startx,starty is a location that give pointer to it.
        maxx,maxy is maximum range to mouse moving */
{
    if(!mouse_reset())return(FALSE);
    else{
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        set_mouse_position(startx,starty);
        set_mouse_horizontal_limit(0,(maxx));
        set_mouse_vertical_limit(0,maxy);
        return(TRUE);
      }
  }

  void CloseMouse()
  {
     mouse_reset();
  }

  void show_pointer(int x_cursor,int y_cursor,int function)
  {
     static int oldx_cursor,oldy_cursor;
     setcolor(LIGHTBLUE);
     if(function==ON)
     {
        setwritemode(XOR_PUT);
        line(0,y_cursor,MaxX,y_cursor);
        line(x_cursor,0,x_cursor,MaxY);
        oldx_cursor=x_cursor;
        oldy_cursor=y_cursor;
     }
     if(function==OFF)
     {
        setwritemode(XOR_PUT);
        line(0,oldy_cursor,MaxX,oldy_cursor);
        line(oldx_cursor,0,oldx_cursor,MaxY);
        oldx_cursor=x_cursor;
        oldy_cursor=y_cursor;
     }
     if(function==MOVE)
     {
        setwritemode(XOR_PUT);
        line(0,oldy_cursor,MaxX,oldy_cursor);
        line(oldx_cursor,0,oldx_cursor,MaxY);
        line(0,y_cursor,MaxX,y_cursor);
        line(x_cursor,0,x_cursor,MaxY);
        oldx_cursor=x_cursor;
        oldy_cursor=y_cursor;
     }
     setwritemode(COPY_PUT);
  }

  int getuser(int *x_cursor,int *y_cursor,int *key_code)
  {
     int key,code_mouse;
     static int button=0;
     static int step=1;
     code_mouse=0;
     if(get_mouse(x_cursor,y_cursor,&code_mouse)!=0)
     {
        switch(code_mouse)
        {
           case  IE_LBUTTONDOWN : if(button!=0x0211)button=*key_code = 0x0211;
                                  else *key_code = 0x00 ; break;
           case  IE_LBUTTONUP   : if(button!=0x0210)button=*key_code = 0x0210;
                                  else *key_code = 0x00 ; break;
           case  IE_MBUTTONDOWN : if(button!=0x0221)button=*key_code = 0x0221;
                                  else *key_code = 0x00 ; break;
           case  IE_MBUTTONUP   : if(button!=0x0220)button=*key_code = 0x0220;
                                  else *key_code = 0x00 ; break;
           case  IE_RBUTTONDOWN : if(button!=0x0231)button=*key_code = 0x0231;
                                  else *key_code = 0x00 ; break;
           case  IE_RBUTTONUP   : if(button!=0x0230)button=*key_code = 0x0230;
                                  else *key_code = 0x00 ; break;
           default              : *key_code = 0x0000  ;
        }
  /*    printf("\rkey = %x",*key_code);*/
        return 1;
     }
     if(kbhit()!=0x00)
     {
        key=get_key();
        switch(key)
        {
           case  R_NARROW    :  *x_cursor+=step;key=0;set_mouse_position(*x_cursor,*y_cursor);break;
           case  L_NARROW    :  *x_cursor-=step;key=0;set_mouse_position(*x_cursor,*y_cursor);break;
           case  U_NARROW    :  *y_cursor-=step;key=0;set_mouse_position(*x_cursor,*y_cursor);break;
           case  D_NARROW    :  *y_cursor+=step;key=0;set_mouse_position(*x_cursor,*y_cursor);break;
           case  0x2b        :  step+=5;if(step>50)step=50;key=0;break; /* key + */
           case  0x2d        :  step-=5;if(step< 0)step= 1;key=0;break; /* key - */
        }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        if(*x_cursor>MaxX)*x_cursor=MaxX;
        if(*x_cursor<0)*x_cursor=0;
        if(*y_cursor>MaxY)*y_cursor=MaxY;
        if(*y_cursor<0)*y_cursor=0;
        *key_code = key;
/*      printf("\rkey = %x",key);*/
        return 1;
    }
    return 0;
}

int get_mouse(int *x_cursor,int *y_cursor,int *code_mouse)
{
    #define leftbutton  0
    #define rightbutton  1
    #define middlebutton 2
    int button;
    static int oldx_cursor,oldy_cursor,oldcode_mouse;
    oldx_cursor=*x_cursor;oldy_cursor=*y_cursor;*code_mouse=oldcode_mouse;
    get_mouse_position(code_mouse,x_cursor,y_cursor);
    if((*code_mouse==oldcode_mouse)&&(oldx_cursor==*x_cursor)&&
          (oldy_cursor==*y_cursor))return 0;

    *code_mouse=0;
    button=leftbutton;
    if(get_mouse_button_presser(&button,x_cursor,y_cursor)>0)
    { *code_mouse=IE_LBUTTONDOWN;}
    button=leftbutton;
    if(get_mouse_button_releases(&button,x_cursor,y_cursor)>0)
    { *code_mouse=IE_LBUTTONUP;}
    button=rightbutton;
    if(get_mouse_button_presser(&button,x_cursor,y_cursor)>0)
    { *code_mouse=IE_RBUTTONDOWN;}
    button=rightbutton;
    if(get_mouse_button_releases(&button,x_cursor,y_cursor)>0)
    { *code_mouse=IE_RBUTTONUP;}
    button=middlebutton;
    if(get_mouse_button_presser(&button,x_cursor,y_cursor)>0)
    { *code_mouse=IE_MBUTTONDOWN;}
    button=middlebutton;
    if(get_mouse_button_releases(&button,x_cursor,y_cursor)>0)
    { *code_mouse=IE_MBUTTONUP;}
    get_mouse_position(&button,x_cursor,y_cursor);
/*  printf("\rx= %3d y= %3d",x_cursor,y_cursor);*/
    oldcode_mouse=button;
    return 1;
}
```

โปรแกรมควบคุมการแสดงรายการแก้ไขบนจอภาพ

```
#include<stdio.h>
#include<graphics.h>

int  gprintf( int *xloc, int *yloc, char *fmt, ... );
int  gprintfxy( int xloc, int yloc, int textcolor , char *fmt, ... );
int  barprintf( int *xloc, int *yloc,int color, char *fmt, ... );
int  barprintfxy( int xloc, int yloc, int color, int bkcolor, char *fmt, ... );
int  get_menu(char *menu[],int *x_cursor,int *y_cursor);
void draw_menu(char *menu[],int left_menu);
void clear_menu(char *menu[],int left_menu);
void mark_menu_address(int y_cursor,int process);
int  get_menu_address(int y_cursor,int left_menu,int right_menu,int key_code,int process);

int  get_menu(char *menu[],int *x_cursor,int *y_cursor)
{
    int menu_code,menu_select;
    int top_menu,bellow_menu,left_menu,right_menu;
    int key_code;
    unsigned int size;
    void far *backup_image;

    menu_code=menu_select=0;
    top_menu =  1 ; bellow_menu = MaxY-1;
    left_menu = 550 ; right_menu  = MaxX-1;
    size = imagesize(left_menu,top_menu,right_menu,bellow_menu);
    backup_image = malloc(size);
    getimage(left_menu,top_menu,right_menu,bellow_menu,backup_image);
    clear_menu(menu,left_menu);
    draw_menu(menu,left_menu);
    menu_select=OFF;
    while((*x_cursor>=left_menu)&&(menu_select==OFF))
    {
        while(getuser(x_cursor,y_cursor,&key_code)==0x00);
        menu_code=get_menu_address(*y_cursor,left_menu,right_menu,key_code,MOVE);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        if(menu_code!=0x00)menu_select=ON;
    }
    putimage(left_menu,top_menu,backup_image,COPY_PUT);
    free(backup_image);
    return menu_code;
}

int  get_menu_address(int y_cursor,int left_menu,int right_menu,int key_code,int process)
{
    static int oldy_cursor;
    int i,ip,xx1,yy2;
    int oldcolor;
    if(process==ON)oldy_cursor=0;
    i=(int)((y_cursor+17)/(textheight("H")+12));
    if(i==0)i=1;
    xx1=left_menu;yy2=460;
    barprintf(&xx1,&yy2,BLACK," %2d ",i);
    ip=i*(textheight("H")+12)+1;
    if(i>0&&i<24){
        oldcolor=getcolor();
        if(oldy_cursor!=0){
            setcolor(BLUE);
            rectangle(left_menu,oldy_cursor-15,right_menu,oldy_cursor);
        }
        setcolor(YELLOW);
        oldy_cursor=ip;
        rectangle(left_menu,ip-15,right_menu,ip);
        setcolor(oldcolor);
    }
    if((key_code==0x0d)||(key_code==0x0211))
        return i;
    if(key_code!=0x00)
        return key_code;
    return 0;
}

void draw_menu(char *menu[],int left_menu)
{
    register int i,len,count,menu_address;
    unsigned int *p;
    int y,oldcolor;
    int x=left_menu+4;
    menu_address = atoi(menu[0]);
    count = atoi(menu[1])*2+2;
    oldcolor=getcolor();
    setcolor(YELLOW);
    settextstyle(SMALL_FONT,0,0);
    y = menu_address*(12+textwidth("H"))+0-textwidth("H")-2;
    len = 0;
    for(i=2;i<count;i+=2)
        if(strlen(menu[i])>len)len=strlen(menu[i])*textwidth("H");
    for(i=2;i<count;i+=2,y=y+10)
    {
        gprintf(&x,&y,menu[i]);
    }
    setcolor(oldcolor);
}

int gprintf( int *xloc, int *yloc, char *fmt, ... )
{
  va_list  argptr;          /* Argument list pointer */
  char str[140];        /* Buffer to build sting into */
  int cnt;              /* Result of SPRINTF for return */
  va_start( argptr, format );      /* Initialize va_ functions   */
    cnt = vsprintf( str, fmt, argptr );   /* prints string to buffer */

  outtextxy( *xloc, *yloc, str );/* Send string in graphics mode */
  *yloc += textheight( "H" ) + 2;      /* Advance to next line        */

  va_end( argptr );         /* Close va_ functions       */

  return( cnt );          /* Return the conversion count*/
}
int gprintfxy( int xloc, int yloc, int textcolor, char *fmt, ... )
{
    int oldcolor;
  va_list  argptr;          /* Argument list pointer */
  char str[140];        /* Buffer to build sting into */
  int cnt;              /* Result of SPRINTF for return */
  va_start( argptr, format );      /* Initialize va_ functions   */
    cnt = vsprintf( str, fmt, argptr );   /* prints string to buffer */
    oldcolor=getcolor();
    setcolor(textcolor);
    outtextxy( xloc, yloc, str ); /* Send string in graphics mode */
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    setcolor(oldcolor);
  va_end( argptr );       /* Close va_ functions      */
    return( cnt );        /* Return the conversion count*/
}

int barprintf( int *xloc, int *yloc,int color, char *fmt, ... )
{
  va_list  argptr;        /* Argument list pointer */
  char str[140];        /* Buffer to build sting into */
    int cnt;            /* Result of SPRINTF for return */
    struct fillsettingstype old;
  va_start( argptr, format );     /* Initialize va_ functions   */
    cnt = vsprintf( str, fmt, argptr );   /* prints string to buffer */
    getfillsettings(&old);
    setfillstyle(SOLID_FILL,color);
    bar(*xloc-1,*yloc-1,*xloc+textwidth(str)+1,*yloc+textheight("H"));
    setfillstyle(old.pattern,old.color);
  outtextxy( *xloc, *yloc, str );/* Send string in graphics mode */
  *yloc += textheight( "H" ) + 2;      /* Advance to next line       */

  va_end( argptr );       /* Close va_ functions      */

  return( cnt );        /* Return the conversion count*/
}
{
  va_list  argptr;        /* Argument list pointer */
  char str[140];        /* Buffer to build sting into */
    int cnt;            /* Result of SPRINTF for return */
    struct fillsettingstype old;
  va_start( argptr, format );     /* Initialize va_ functions   */
    cnt = vsprintf( str, fmt, argptr );   /* prints string to buffer */
    getfillsettings(&old);
    setfillstyle(SOLID_FILL,bkcolor);
    bar(xloc-1,yloc-1,xloc+textwidth(str)+1,yloc+textheight("H"));
    setfillstyle(old.pattern,old.color);
    old.color=getcolor();
    setcolor(color);
    outtextxy( xloc, yloc, str ); /* Send string in graphics mode */
    setcolor(old.color);
  va_end( argptr );       /* Close va_ functions      */

  return( cnt );        /* Return the conversion count*/
}

void clear_menu(char *menu[],int left_menu)
{
    struct fillsettingstype old;
    int y,menu_address;
    getfillsettings(&old);
    setfillstyle(SOLID_FILL,BLUE);
    menu_address = atoi(menu[0]);
    y = menu_address*(12+textwidth("H"))+0-textwidth("H")-5;
    bar(left_menu,y,MaxX-1,MaxY-1);
    setfillstyle(old.pattern,old.color);
}
```

โปรแกรมควบคุมการแก้ไขข้อมูลบนจอภาพ

```
/***************** EDIT.C ***************/
void edit_data(void);
void coordinate_display(float x,float y);
void adjust_screen( long *cenx,long *ceny,float *ratio,
                    long minx,long miny,long maxx,long maxy);
int check_point(float ratio,long *x,long *y,int *data1);
void cut_data(int start,int end);
void add_data(int start,int number);

#include<rotate.c>
#include<move.c>
#include<2line.c>
#include<3line.c>
#include<3circle.c>
#include<adjsize.c>


void adjust_screen( long *cenx,long *ceny,float *ratio,
                    long minx,long miny,long maxx,long maxy)
{
    float ratiopicture,ratioscreen;
    *cenx=(minx+maxx)/2;*ceny=(miny+maxy)/2;
    ratiopicture = (float)(maxy-miny)/(float)(maxx-minx);
    ratioscreen  = (float)MaxY/(float)MaxX;
    if(ratiopicture>ratioscreen) *ratio = (float)MaxY/(float)(maxy-miny);
    else                 *ratio = (float)MaxX/(float)(maxx-minx);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
}
void coordinate_display(float x,float y)
{
   barprintfxy(3,3,GREEN,BLACK,"X = %6.2f m.m. Y = %6.2f m.m.",x,y);
}
int check_point(float ratio,long *x,long *y,int *data1)
{
   int i,j;
   long dx1,dy1,dx2,dy2,ratio2;
   float dz1,dz2;
   ratio=(1/ratio)*1.5;
   ratio2=(long)(ratio*2);
   *data1=0;
   dx1=labs(X[0]-*x);dy1=labs(Y[0]-*y);
   dx2=dx1;dy2=dy1;
   dz1=(float)sqrt(dx2*dx2+dy2*dy2);
   i=1;
   while((X[i]!=0||Y[i]!=0)&&get_key()!=0x1b)
   {
      dx2=labs(X[i]-*x);
      dy2=labs(Y[i]-*y);
      if((dx2<ratio2)&&(dy2<ratio2))
      {
         dz2=(float)sqrt(dx2*dx2+dy2*dy2);
         if(dz2<dz1)
         {
            dz1=dz2;
            dx1=dx2;dy1=dy2;
            *data1=i;
         }
      }
      i++;
   }
   printf("\r ratio = %6.2f dz1= %6.2f ",ratio,dz1);
   if(dz1<=ratio)
   {
      *x=X[*data1];*y=Y[*data1];
      return OK;
   }
   return NO;
}
void cut_data(int start,int end)
{
   int i,di;
   if(start>end)return;
   i = start; di = end-start+1;
   while((X[i]!=0)||(Y[i]!=0))
   {
      X[i]=X[i+di];
      Y[i]=Y[i+di];
      i++;
   }
}
void add_data(int start,int number)
{
   int i,di,end;
   long bak1,bak2;
   if(number<=0)return;
   end = start;
   while((X[end]!=0)||(Y[end]!=0))
   {
      end++;
   }
   end--;
   i = start;
   while(end>i)
   {
      X[end+number] = X[end];
      Y[end+number] = Y[end];
      end--;
   }
}

void edit_data(void)
{
   char  *Edit[]={"1","23",
            " HELP       "," 1001 ",
            " FULL SCREEN "," 1002 ",
            " EDIT SCREEN "," 1003 ",
            " REDRAW     "," 1004 ",
            " REDOT      "," 1005 ",
            " ZOOMIN     "," 1006 ",
            " ZOOMOUT    "," 1007 ",
            " PAN   OFF  "," 1008 ",
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        " MOVE  OFF ","  1009 ",
        " ROTAGE OFF ","  1010 ",
        " 2 LINE ADJ. ","  1011 ",
        " 3 LINE ADJ. ","  1012 ",
        " 3 CIR. ADJ. ","  1013 ",
        "          ","  1014 ",
        "          ","  1015 ",
        "          ","  1016 ",
        "          ","  1017 ",
        "          ","  1018 ",
        "          ","  1019 ",
        " SIZE  ADJ. ","  1020 ",
        "          ","  1021 ",
        "          ","  1022 ",
        " EXIT     ","  1023 ",};
int  start_edit,end_edit,EDIT_MODE;
int  code_command1,key_code,code_mouse,menu_code;
int  x_cursor,y_cursor;
long x,y;
int  i,move_mode,pan_mode;
float ratiopicture,ratioscreen,ratio;
long minx,miny;
long cenx,ceny;
long maxx,maxy;
long maxx_edit,maxy_edit;
maxx_edit=(long)(750*step_ratio);
maxy_edit=(long)(554*step_ratio);
if(X[0]==0||Y[0]==0){putch(0x07);return;}
if(InitMouse(0,0,MaxX,MaxY)==0x00)
{
   printf("\r no mouse ");putch(0x07);sleep(1);
}
init_graph();
settextstyle(SMALL_FONT,0,0);
x = X[0];   y = Y[0];
getmaxmin_data(&minx,&miny,&maxx,&maxy);
cenx=(minx+maxx)/2;ceny=(miny+maxy)/2;
ratiopicture = (float)(maxy-miny)/(float)(maxx-minx);
ratioscreen  = (float)MaxY/(float)MaxX;
if(ratiopicture>ratioscreen) ratio = (float)MaxY/(float)(maxy-miny);
else                         ratio = (float)MaxX/(float)(maxx-minx);
setcolor(BLACK);
lineto(g_x(cenx,X[0],ratio),g_y(ceny,Y[0],ratio));
setcolor(YELLOW);
rectangle(0,0,MaxX,MaxY);
i=0;
pan_mode = OFF;
setcolor(WHITE);
while((X[i]!=0||Y[i]!=0)&&get_key()!=0x1b)
{
   x=X[i];y=Y[i];
   lineto(g_x(cenx,x,ratio),g_y(ceny,y,ratio));
   i++;
}
setcolor(LIGHTRED);
lineto(g_x(cenx,X[0],ratio),g_y(ceny,Y[0],ratio));
start_edit=0;end_edit=0;
EDIT_MODE=ON;
x_cursor=0;y_cursor=0;
show_pointer(x_cursor,y_cursor,ON);
while(EDIT_MODE==ON)
{
   while(getuser(&x_cursor,&y_cursor,&key_code)==0x00);
   if(x_cursor==MaxX)
   {
      show_pointer(x_cursor,y_cursor,OFF);
      menu_code=get_menu(Edit,&x_cursor,&y_cursor);
      if(menu_code!=0x00)key_code=atoi(Edit[menu_code*2+1]);
      show_pointer(x_cursor,y_cursor,ON);
   }
   if(key_code!=0x00)
   switch(key_code)
   {
      case  1002 : adjust_screen(&cenx,&ceny,&ratio,0,0,maxx_edit,maxy_edit);
                   redot_edit(cenx,ceny,ratio,start_edit,end_edit);
                   show_pointer(x_cursor,y_cursor,ON);
                   break;
      case  1003 :      getmaxmin_data(&minx,&miny,&maxx,&maxy);
                   adjust_screen(&cenx,&ceny,&ratio,minx,miny,maxx,maxy);
                   redot_edit(cenx,ceny,ratio,start_edit,end_edit);
                   show_pointer(x_cursor,y_cursor,ON);
                   break;
      case  1004 :;
      case 'R'   :;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        case 'R'    :;
        case 'r'    : redraw_edit(cenx,ceny,ratio,start_edit,end_edit);
                      show_pointer(x_cursor,y_cursor,ON);
                      break;
        case 1005   : redot_edit(cenx,ceny,ratio,start_edit,end_edit);
                      show_pointer(x_cursor,y_cursor,ON);
                      break;
        case 1006 :;
        case 'Z'    : ratio=ratio*4.0;     /* Zoom In */
        case 1007 :;
        case 'z'    :   ratio=ratio*0.5;       /* Zoom Out */
                      redot_edit(cenx,ceny,ratio,start_edit,end_edit);
                      show_pointer(x_cursor,y_cursor,ON);
                      break;
        case 1008     : if(pan_mode==OFF){pan_mode=ON;
                        strcpy(Edit[(key_code-1000)*2]," PAN   ON ");}
                      else{pan_mode=OFF;strcpy(Edit[(key_code-1000)*2]," PAN   OFF ");}
                      break;
        case 1009   : move_edit(&x_cursor,&y_cursor,&cenx,&ceny,&ratio,&start_edit,&end_edit);
                      getmaxmin_data(&minx,&miny,&maxx,&maxy);
                      break;
        case 1010   : rotate_edit(&x_cursor,&y_cursor,&cenx,&ceny,&ratio,&start_edit,&end_edit);
                      getmaxmin_data(&minx,&miny,&maxx,&maxy);
                      break;
        case 1011   : twoline_edit(&x_cursor,&y_cursor,&cenx,&ceny,&ratio,&start_edit,&end_edit);
                      getmaxmin_data(&minx,&miny,&maxx,&maxy);
                      break;
        case 1012   : threeline_edit(&x_cursor,&y_cursor,&cenx,&ceny,&ratio,&start_edit,&end_edit);
                      getmaxmin_data(&minx,&miny,&maxx,&maxy);
                      break;
        case 1013   : circle3_edit(&x_cursor,&y_cursor,&cenx,&ceny,&ratio,&start_edit,&end_edit);
                      getmaxmin_data(&minx,&miny,&maxx,&maxy);
                      break;

        case 'C'  :;
        case 'c'    :       cenx=g_xl(cenx,x_cursor,ratio);
                      ceny=g_yl(ceny,y_cursor,ratio);
                      redot_edit(cenx,ceny,ratio,start_edit,end_edit);
                      show_pointer(x_cursor,y_cursor,ON);
                      break;
        case 1020   : adjsize_edit(&x_cursor,&y_cursor,&cenx,&ceny,&ratio,&start_edit,&end_edit);
                      getmaxmin_data(&minx,&miny,&maxx,&maxy);
                      break;
        case 1023 :;
        case 'Q'  :;
        case 'q'        : EDIT_MODE = OFF; break;
        case 0x0231:;
        case ESC    : break;
        case 0x0d :;
        case 0x0211     : if(pan_mode==ON)
                      {cenx=g_xl(cenx,x_cursor,ratio);
                      ceny=g_yl(ceny,y_cursor,ratio);
                      redot_edit(cenx,ceny,ratio,start_edit,end_edit);
                      show_pointer(x_cursor,y_cursor,ON);
                      break;}
                      break;
      }
      else
      {
/*      show_pointer(x_cursor,y_cursor,MOVE);*/
      show_pointer(x_cursor,y_cursor,OFF);
      coordinate_display( g_xl(cenx,x_cursor,ratio)/step_ratio,
                          g_yl(ceny,y_cursor,ratio)/step_ratio);
      show_pointer(x_cursor,y_cursor,ON);
      }
      key_code=0;
   }
   show_pointer(x_cursor,y_cursor,OFF);
   CloseMouse();
   closegraph();
}
```

โปรแกรมคำสั่งหมุนข้อมูลในหน่วยความจำ

```
/************************* ROTATE.C ****************************/
void rotate_edit(int *x_cursor,int *y_cursor,long *cenx,long *ceny,float *ratio,int *start_edit,int *end_edit);
void rotage_line(long x_origin,long y_origin,int x_cursor,int y_cursor,long cenx,long ceny,float ratio,int mode);
void rotage_adjust(long x_origin,long y_origin,long x_second,long y_second,long x_end,long y_end);

void rotate_edit(int *x_cursor,int *y_cursor,long *cenx,long *ceny,float *ratio,int *start_edit,int *end_edit)
{
   char  *Menu[]={"1","23",
          " HELP      "," 1001 ",
          " FULL SCREEN "," 1002 ",
          " EDIT SCREEN "," 1003 ",
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        " REDRAW     "," 1004 ",
        " REDOT      "," 1005 ",
        " ZOOMIN     "," 1006 ",
        " ZOOMOUT    "," 1007 ",
        " PAN   OFF  "," 1008 ",
        "            "," 1009 ",
        " ROTAGE ON  "," 1010 ",
        "            "," 1011 ",
        "            "," 1012 ",
        "            "," 1013 ",
        "            "," 1014 ",
        "            "," 1015 ",
        "            "," 1016 ",
        "            "," 1017 ",
        "            "," 1018 ",
        "            "," 1019 ",
        "            "," 1020 ",
        "            "," 1021 ",
        "            "," 1022 ",
        " EXIT       "," 1023 ",};
int key_code,menu_code;
int pan_mode,current_mode;
long x_origin,y_origin,x_second,y_second,x_end,y_end;
long minx,miny,maxx,maxy;
long maxx_edit,maxy_edit;
maxx_edit=(long)(750*step_ratio);
maxy_edit=(long)(554*step_ratio);
getmaxmin_data(&minx,&miny,&maxx,&maxy);
x_origin=x_second=x_end=y_origin=0;
pan_mode = OFF;
current_mode = ON;
while(current_mode==ON)
{
   while(getuser(x_cursor,y_cursor,&key_code)==0x00);
   if(*x_cursor==MaxX)
   {
      if(x_origin!=0) rotage_line(x_origin,y_origin,*x_cursor,
                                  *y_cursor,*cenx,*ceny,*ratio,OFF);
      show_pointer(*x_cursor,*y_cursor,OFF);
      menu_code=get_menu(Menu,x_cursor,y_cursor);
      if(menu_code!=0x00)key_code=atoi(Menu[menu_code*2+1]);
      show_pointer(*x_cursor,*y_cursor,ON);
      if(x_origin!=0) rotage_line(x_origin,y_origin,*x_cursor,
                                  *y_cursor,*cenx,*ceny,*ratio,ON);
   }
   if(key_code!=0x00)
   switch(key_code)
   {
      case 1002 : adjust_screen(cenx,ceny,ratio,0,0,maxx_edit,maxy_edit);
                  redraw_edit(*cenx,*ceny,*ratio,*start_edit,*end_edit);
                  show_pointer(*x_cursor,*y_cursor,ON);
                  break;
      case 1003 :    getmaxmin_data(&minx,&miny,&maxx,&maxy);
                  adjust_screen(cenx,ceny,ratio,minx,miny,maxx,maxy);
                  redraw_edit(*cenx,*ceny,*ratio,*start_edit,*end_edit);
                  show_pointer(*x_cursor,*y_cursor,ON);
                  break;
      case 1004 :;
      case 'R'  :;
      case 'r'  : redraw_edit(*cenx,*ceny,*ratio,*start_edit,*end_edit);
                  show_pointer(*x_cursor,*y_cursor,ON);
                  break;
      case 1005 : redot_edit(*cenx,*ceny,*ratio,*start_edit,*end_edit);
                  show_pointer(*x_cursor,*y_cursor,ON);
                  break;
      case 1006 :;
      case 'Z'  : *ratio=*ratio*4.0;      /* Zoom In */
      case 1007 :;
      case 'z'  :  *ratio=*ratio*0.5;    /* Zoom Out */
                  redraw_edit(*cenx,*ceny,*ratio,*start_edit,*end_edit);
                  show_pointer(*x_cursor,*y_cursor,ON);
                  break;
      case 1008 : if(pan_mode==OFF){pan_mode=ON;
                     strcpy(Menu[(key_code-1000)*2]," PAN   ON ");}
                  else{pan_mode=OFF;strcpy(Menu[(key_code-1000)*2]," PAN   OFF ");}
                  break;
      case 1010 : current_mode = OFF;
                  break;
      case 1023 :;
      case 'Q'  :;
      case 'q'  : current_mode = OFF; break;
      case 0x0231:;
      case ESC  : if(x_origin!=0)rotage_line(x_origin,y_origin,
                  *x_cursor,*y_cursor,*cenx,*ceny,*ratio,OFF);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                                x_origin=x_second=0;
                                break;
        case 0x0d :;
        case 0x0211     : if(pan_mode==ON)
                                {*cenx=g_xl(*cenx,*x_cursor,*ratio);
                                *ceny=g_yl(*ceny,*y_cursor,*ratio);
                                redraw_edit(*cenx,*ceny,*ratio,*start_edit,*end_edit);
                                show_pointer(*x_cursor,*y_cursor,ON);
                                if(x_origin!=0x00)
                                   rotage_line(x_origin,y_origin,*x_cursor,*y_cursor,*cenx,*ceny,*ratio,ON);
                                break;}
                                if(x_origin==0x00)
                                {
                                   x_origin=g_xl(*cenx,*x_cursor,*ratio);
                                   y_origin=g_yl(*ceny,*y_cursor,*ratio);
                                   show_pointer(*x_cursor,*y_cursor,OFF);
                                   show_pointer(*x_cursor,*y_cursor,ON);
                                   rotage_line(x_origin,y_origin,*x_cursor,*y_cursor,*cenx,*ceny,*ratio,ON);
                                }
                                else
                                if(x_second==0x00)
                                {
                                   x_second=g_xl(*cenx,*x_cursor,*ratio);
                                   y_second=g_yl(*ceny,*y_cursor,*ratio);
                                   show_pointer(*x_cursor,*y_cursor,OFF);
                                   show_pointer(*x_cursor,*y_cursor,ON);
                                }
                                else
                                {
                                   x_end=g_xl(*cenx,*x_cursor,*ratio);
                                   y_end=g_yl(*ceny,*y_cursor,*ratio);
                                   rotage_adjust(x_origin,y_origin,x_second,y_second,x_end,y_end);
                                   redraw_edit(*cenx,*ceny,*ratio,*start_edit,*end_edit);
                                   show_pointer(*x_cursor,*y_cursor,ON);
                                   x_origin=x_second=0;
                                   y_origin=y_second=0;
                                }break;
      }
      else
      {
         if(x_origin!=0)rotage_line(x_origin,y_origin,*x_cursor,*y_cursor,
                                   *cenx,*ceny,*ratio,OFF);
            show_pointer(*x_cursor,*y_cursor,OFF);
            coordinate_display( g_xl(*cenx,*x_cursor,*ratio)/step_ratio,
                              g_yl(*ceny,*y_cursor,*ratio)/step_ratio);
            show_pointer(*x_cursor,*y_cursor,ON);
         if(x_origin!=0)rotage_line(x_origin,y_origin,*x_cursor,*y_cursor,
                                   *cenx,*ceny,*ratio,ON);
      }
      key_code=0;
   }
}
void rotage_line(long x_origin,long y_origin,int x_cursor,int y_cursor,long cenx,long ceny,float ratio,int mode)
{
   static int oldx_cursor,oldy_cursor;
   setcolor(LIGHTRED);
   setwritemode(XOR_PUT);
   if(mode==ON)
   {
      line(g_x(cenx,x_origin,ratio),g_y(ceny,y_origin,ratio),
           x_cursor,y_cursor);
      oldx_cursor=x_cursor;
      oldy_cursor=y_cursor;
   }
   if(mode==MOVE)
   {
      line(g_x(cenx,x_origin,ratio),g_y(ceny,y_origin,ratio),
           oldx_cursor,oldy_cursor);
      line(g_x(cenx,x_origin,ratio),g_y(ceny,y_origin,ratio),
           x_cursor,y_cursor);
      oldx_cursor=x_cursor;
      oldy_cursor=y_cursor;
   }
   if(mode==OFF)
   {
      line(g_x(cenx,x_origin,ratio),g_y(ceny,y_origin,ratio),
           oldx_cursor,oldy_cursor);
   }
}

void rotage_adjust(long x_origin,long y_origin,long x_second,long y_second,long x_end,long y_end)
{
   int i;
   float zeta1,zeta2,zeta3,z;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    long x,y;
    x=x_second-x_origin;
    y=y_second-y_origin;
    zeta1 = (float)atan2(y,x);
    x=x_end-x_origin;
    y=y_end-y_origin;
    zeta2 = (float)atan2(y,x)-zeta1;

/*  x=start_x-x_origin;y=start_y-y_origin;
    zeta3=(float)atan2(y,x)+zeta2;
    z=(float)sqrt((x*x+y*y));
    start_x=x_origin+(long)(z*cos(zeta3));
    start_y=y_origin+(long)(z*sin(zeta3));*/

    i=0;
    while((X[i]!=0||Y[i]!=0)&&get_key()!=0x1b)
    {
       x=X[i]-x_origin;y=Y[i]-y_origin;
       zeta3=(float)atan2(y,x)+zeta2;
       z=(float)sqrt((x*x+y*y));
       X[i]=x_origin+(long)(z*cos(zeta3));
       Y[i]=y_origin+(long)(z*sin(zeta3));
       i++;
    }
}
```

โปรแกรมคำสั่งเลื่อนข้อมูลในหน่วยความจำ

```
void move_edit(int *x_cursor,int *y_cursor,
        long *cenx,long *ceny,float *ratio,int *start_edit,int *end_edit);
void move_line(long x_origin,long y_origin,int x_cursor,int y_cursor,long cenx,long ceny,float ratio,int mode);
void move_adjust(long x_origin,long y_origin,long x_second,long y_second);
void move_edit(int *x_cursor,int *y_cursor,
        long *cenx,long *ceny,float *ratio,int *start_edit,int *end_edit)
{
    char  *Menu[]={"1","23",
              " HELP       "," 1001 ",
              " FULL SCREEN "," 1002 ",
              " EDIT SCREEN "," 1003 ",
              " REDRAW     "," 1004 ",
              " REDOT      "," 1005 ",
              " ZOOMIN     "," 1006 ",
              " ZOOMOUT    "," 1007 ",
              " PAN   OFF  "," 1008 ",
              " MOVE  ON   "," 1009 ",
              "            "," 1010 ",
              "            "," 1011 ",
              "            "," 1012 ",
              "            "," 1013 ",
              "            "," 1014 ",
              "            "," 1015 ",
              "            "," 1016 ",
              "            "," 1017 ",
              "            "," 1018 ",
              "            "," 1019 ",
              "            "," 1020 ",
              "            "," 1021 ",
              "            "," 1022 ",
              " EXIT       "," 1023 ",};
    int key_code,menu_code;
    int pan_mode,current_mode;
    long x_origin,y_origin,x_second,y_second,x_end,y_end;
    long minx,miny,maxx,maxy;
    long maxx_edit,maxy_edit;
    maxx_edit=(long)(750*step_ratio);
    maxy_edit=(long)(554*step_ratio);
    getmaxmin_data(&minx,&miny,&maxx,&maxy);
    x_origin=x_second=y_origin=0;
    pan_mode = OFF;
    current_mode = ON;
    while(current_mode==ON)
    {
       while(getuser(x_cursor,y_cursor,&key_code)==0x00);
       if(*x_cursor==MaxX)
       {
          if(x_origin!=0) move_line(x_origin,y_origin,*x_cursor,
                                          *y_cursor,*cenx,*ceny,*ratio,OFF);
          show_pointer(*x_cursor,*y_cursor,OFF);
          menu_code=get_menu(Menu,x_cursor,y_cursor);
          if(menu_code!=0x00)key_code=atoi(Menu[menu_code*2+1]);
          show_pointer(*x_cursor,*y_cursor,ON);
          if(x_origin!=0) move_line(x_origin,y_origin,*x_cursor,
                                          *y_cursor,*cenx,*ceny,*ratio,ON);
       }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
if(key_code!=0x00)
switch(key_code)
{
   case 1002 : adjust_screen(cenx,ceny,ratio,0,0,maxx_edit,maxy_edit);
                redraw_edit(*cenx,*ceny,*ratio,*start_edit,*end_edit);
                show_pointer(*x_cursor,*y_cursor,ON);
                break;
   case 1003 :   getmaxmin_data(&minx,&miny,&maxx,&maxy);
                adjust_screen(cenx,ceny,ratio,minx,miny,maxx,maxy);
                redraw_edit(*cenx,*ceny,*ratio,*start_edit,*end_edit);
                show_pointer(*x_cursor,*y_cursor,ON);
                break;
   case 1004 :;
   case 'R'  :;
   case 'r'  : redraw_edit(*cenx,*ceny,*ratio,*start_edit,*end_edit);
                show_pointer(*x_cursor,*y_cursor,ON);
                break;
   case 1005  : redot_edit(*cenx,*ceny,*ratio,*start_edit,*end_edit);
                show_pointer(*x_cursor,*y_cursor,ON);
                break;
   case 1006 :;
   case 'Z'      : *ratio=*ratio*4.0;     /* Zoom In */
   case 1007 :;
   case 'z'      :  *ratio=*ratio*0.5;    /* Zoom Out */
                redraw_edit(*cenx,*ceny,*ratio,*start_edit,*end_edit);
                show_pointer(*x_cursor,*y_cursor,ON);
                break;
   case 1008       : if(pan_mode==OFF){pan_mode=ON;
                    strcpy(Menu[(key_code-1000)*2]," PAN   ON  ");}
                else{pan_mode=OFF;strcpy(Menu[(key_code-1000)*2]," PAN   OFF  ");}
                break;
   case 1009    : current_mode = OFF;
                break;
   case 1023 :;
   case 'Q'  :;
   case 'q'       : current_mode = OFF; break;
   case 0x0231:;
   case ESC    : if(x_origin!=0)move_line(x_origin,y_origin,
                    *x_cursor,*y_cursor,*cenx,*ceny,*ratio,OFF);
                x_origin=x_second=0;
                break;
   case 0x0d :;
   case 0x0211    : if(pan_mode==ON)
                {*cenx=g_xl(*cenx,*x_cursor,*ratio);
                *ceny=g_yl(*ceny,*y_cursor,*ratio);
                redraw_edit(*cenx,*ceny,*ratio,*start_edit,*end_edit);
                show_pointer(*x_cursor,*y_cursor,ON);
                if(x_origin!=0x00)
                   move_line(x_origin,y_origin,*x_cursor,*y_cursor,*cenx,*ceny,*ratio,ON);
                break;}
                if(x_origin==0x00)
                {
                   x_origin=g_xl(*cenx,*x_cursor,*ratio);
                   y_origin=g_yl(*ceny,*y_cursor,*ratio);
                   show_pointer(*x_cursor,*y_cursor,OFF);
                   show_pointer(*x_cursor,*y_cursor,ON);
                   move_line(x_origin,y_origin,*x_cursor,*y_cursor,*cenx,*ceny,*ratio,ON);
                }
                else
                if(x_second==0x00)
                {
                   x_second=g_xl(*cenx,*x_cursor,*ratio);
                   y_second=g_yl(*ceny,*y_cursor,*ratio);
                   move_adjust(x_origin,y_origin,x_second,y_second);
                   redraw_edit(*cenx,*ceny,*ratio,*start_edit,*end_edit);
                   show_pointer(*x_cursor,*y_cursor,ON);
                   x_origin=x_second=0;
                   y_origin=y_second=0;
                }break;
}
else
{
   if(x_origin!=0)move_line(x_origin,y_origin,*x_cursor,*y_cursor,
                            *cenx,*ceny,*ratio,OFF);
      show_pointer(*x_cursor,*y_cursor,OFF);
      coordinate_display( g_xl(*cenx,*x_cursor,*ratio)/step_ratio,
                          g_yl(*ceny,*y_cursor,*ratio)/step_ratio);
      show_pointer(*x_cursor,*y_cursor,ON);
   if(x_origin!=0)move_line(x_origin,y_origin,*x_cursor,*y_cursor,
                            *cenx,*ceny,*ratio,ON);
}
key_code=0;
}
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
void move_line(long x_origin,long y_origin,int x_cursor,int y_cursor,long cenx,long ceny,float ratio,int mode)
{
   static int oldx_cursor,oldy_cursor;
   setcolor(LIGHTRED);
   setwritemode(XOR_PUT);
   if(mode==ON)
   {
      line(g_x(cenx,x_origin,ratio),g_y(ceny,y_origin,ratio),
           x_cursor,y_cursor);
      oldx_cursor=x_cursor;
      oldy_cursor=y_cursor;
   }
   if(mode==MOVE)
   {
      line(g_x(cenx,x_origin,ratio),g_y(ceny,y_origin,ratio),
           oldx_cursor,oldy_cursor);
      line(g_x(cenx,x_origin,ratio),g_y(ceny,y_origin,ratio),
           x_cursor,y_cursor);
      oldx_cursor=x_cursor;
      oldy_cursor=y_cursor;
   }
   if(mode==OFF)
   {
      line(g_x(cenx,x_origin,ratio),g_y(ceny,y_origin,ratio),
           oldx_cursor,oldy_cursor);
   }
}

void move_adjust(long x_origin,long y_origin,long x_second,long y_second)
{
   int i;
   long x,y;
   x=x_second-x_origin;
   y=y_second-y_origin;
   i=0;
   while((X[i]!=0||Y[i]!=0)&&get_key()!=0x1b)
   {
      X[i]=X[i]+x;
      Y[i]=Y[i]+y;
      i++;
   }
}
```

โปรแกรมคำสั่งแก้ไขเส้นตรงของข้อมูลแบบอ้างอิง 2 จุด

```
/************* 2LINE.C **************/
void twoline_edit(int *x_cursor,int *y_cursor,
      long *cenx,long *ceny,float *ratio,int *start_edit,int *end_edit);
void twoline_line(long x_origin,long y_origin,int x_cursor,int y_cursor,long cenx,long ceny,float ratio,int mode);
void twoline_adjust(int d1,int d2);

void twoline_edit(int *x_cursor,int *y_cursor,
      long *cenx,long *ceny,float *ratio,int *start_edit,int *end_edit)
{
   char  *Menu[]={"1","23",
           " HELP       "," 1001 ",
           " FULL SCREEN "," 1002 ",
           " EDIT SCREEN "," 1003 ",
           " REDRAW     "," 1004 ",
           " REDOT      "," 1005 ",
           " ZOOMIN     "," 1006 ",
           " ZOOMOUT    "," 1007 ",
           " PAN   OFF  "," 1008 ",
           "            "," 1009 ",
           "            "," 1010 ",
           " 2-LINE ADJ. "," 1011 ",
           "            "," 1012 ",
           "            "," 1013 ",
           "            "," 1014 ",
           "            "," 1015 ",
           "            "," 1016 ",
           "            "," 1017 ",
           "            "," 1018 ",
           "            "," 1019 ",
           "            "," 1020 ",
           "            "," 1021 ",
           "            "," 1022 ",
           " EXIT       "," 1023 ",};
   int  key_code,menu_code;
   int  pan_mode,current_mode;
   long x_origin,y_origin,x_second,y_second,x_end,y_end;
   int  data1,data2;
   long minx,miny,maxx,maxy;
   long maxx_edit,maxy_edit;
```

เอกสารนี้สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
maxx_edit=(long)(750*step_ratio);
maxy_edit=(long)(554*step_ratio);
getmaxmin_data(&minx,&miny,&maxx,&maxy);
x_origin=x_second=y_origin=0;
pan_mode = OFF;
current_mode = ON;
while(current_mode==ON)
{
  while(getuser(x_cursor,y_cursor,&key_code)==0x00);
  if(*x_cursor==MaxX)
  {
    if(x_origin!=0) twoline_line(x_origin,y_origin,*x_cursor,
                                 *y_cursor,*cenx,*ceny,*ratio,OFF);
    show_pointer(*x_cursor,*y_cursor,OFF);
    menu_code=get_menu(Menu,x_cursor,y_cursor);
    if(menu_code!=0x00)key_code=atoi(Menu[menu_code*2+1]);
    show_pointer(*x_cursor,*y_cursor,ON);
    if(x_origin!=0) twoline_line(x_origin,y_origin,*x_cursor,
                                 *y_cursor,*cenx,*ceny,*ratio,ON);
  }
  if(key_code!=0x00)
  switch(key_code)
  {
    case 1002 : adjust_screen(cenx,ceny,ratio,0,0,maxx_edit,maxy_edit);
                redraw_edit(*cenx,*ceny,*ratio,*start_edit,*end_edit);
                show_pointer(*x_cursor,*y_cursor,ON);
                if(x_origin!=0x00)
                  twoline_line(x_origin,y_origin,*x_cursor,*y_cursor,*cenx,*ceny,*ratio,ON);
                break;
    case 1003 :   getmaxmin_data(&minx,&miny,&maxx,&maxy);
                adjust_screen(cenx,ceny,ratio,minx,miny.maxx,maxy);
                redot_edit(*cenx,*ceny,*ratio,*start_edit,*end_edit);
                show_pointer(*x_cursor,*y_cursor,ON);
                if(x_origin!=0x00)
                  twoline_line(x_origin,y_origin,*x_cursor,*y_cursor,*cenx,*ceny,*ratio,ON);
                break;
    case 1004 :;
    case 'R'  :;
    case 'r'  : redraw_edit(*cenx,*ceny,*ratio,*start_edit,*end_edit);
                show_pointer(*x_cursor,*y_cursor,ON);
                if(x_origin!=0x00)
                  twoline_line(x_origin,y_origin,*x_cursor,*y_cursor,*cenx,*ceny,*ratio,ON);
                break;
    case 1005 : redot_edit(*cenx,*ceny,*ratio,*start_edit,*end_edit);
                show_pointer(*x_cursor,*y_cursor,ON);
                if(x_origin!=0x00)
                  twoline_line(x_origin,y_origin,*x_cursor,*y_cursor,*cenx,*ceny,*ratio,ON);
                break;
    case 1006 :;
    case 'Z'  : *ratio=*ratio*4.0;    /* Zoom In */
    case 1007 :;
    case 'z'  :   *ratio=*ratio*0.5;  /* Zoom Out */
                redot_edit(*cenx,*ceny,*ratio,*start_edit,*end_edit);
                show_pointer(*x_cursor,*y_cursor,ON);
                if(x_origin!=0x00)
                  twoline_line(x_origin,y_origin,*x_cursor,*y_cursor,*cenx,*ceny,*ratio,ON);
                break;
    case 1008 : if(pan_mode==OFF){pan_mode=ON;
                  strcpy(Menu[(key_code-1000)*2]," PAN   ON  ");}
                else{pan_mode=OFF;strcpy(Menu[(key_code-1000)*2]," PAN   OFF ");}
                break;
    case 1011 : current_mode = OFF;
                break;
    case 1023 :;
    case 'Q'  :;
    case 'q'  : current_mode = OFF; break;
    case 0x0231:;
    case ESC  : if(x_origin!=0)twoline_line(x_origin,y_origin,
                  *x_cursor,*y_cursor,*cenx,*ceny,*ratio,OFF);
                x_origin=x_second=0;
                break;
    case 0x0d :;
    case 0x0211 : if(pan_mode==ON)
                {*cenx=g_xl(*cenx,*x_cursor,*ratio);
                *ceny=g_yl(*ceny,*y_cursor,*ratio);
                redot_edit(*cenx,*ceny,*ratio,*start_edit,*end_edit);
                show_pointer(*x_cursor,*y_cursor,ON);
                if(x_origin!=0x00)
                  twoline_line(x_origin,y_origin,*x_cursor,*y_cursor,*cenx,*ceny,*ratio,ON);
                break;}
                if(x_origin==0x00)
                {
                  x_origin=g_xl(*cenx,*x_cursor,*ratio);
                  y_origin=g_yl(*ceny,*y_cursor,*ratio);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                                    if(check_point(*ratio,&x_origin,&y_origin,&data1)==OK)
                                    {
                                       show_pointer(*x_cursor,*y_cursor,OFF);
                                       show_pointer(*x_cursor,*y_cursor,ON);
                                       twoline_line(x_origin,y_origin,*x_cursor,*y_cursor,*cenx,*ceny,*ratio,ON);
                                       break;
                                    }
                                    x_origin=0;
                                    y_origin=0;
                                    putch(0x07);break;
                                 }
                                 if(x_second==0x00)
                                 {
                                    x_second=g_xl(*cenx,*x_cursor,*ratio);
                                    y_second=g_yl(*ceny,*y_cursor,*ratio);
                                    if(check_point(*ratio,&x_second,&y_second,&data2)==OK)
                                    {
                                       twoline_adjust(data1,data2);
                                       redot_edit(*cenx,*ceny,*ratio,*start_edit,*end_edit);
                                       show_pointer(*x_cursor,*y_cursor,ON);
                                       x_origin=x_second=0;
                                       break;
                                    }
                                    putch(0x07);
                                    x_second=0;
                                    break;
                                 }break;
      }
      else
      {
         if(x_origin!=0)twoline_line(x_origin,y_origin,*x_cursor,*y_cursor,
                                          *cenx,*ceny,*ratio,OFF);
            show_pointer(*x_cursor,*y_cursor,OFF);
            coordinate_display( g_xl(*cenx,*x_cursor,*ratio)/step_ratio,
                                g_yl(*ceny,*y_cursor,*ratio)/step_ratio);
            show_pointer(*x_cursor,*y_cursor,ON);
         if(x_origin!=0)twoline_line(x_origin,y_origin,*x_cursor,*y_cursor,
                                          *cenx,*ceny,*ratio,ON);
      }
      key_code=0;
   }
}
void twoline_line(long x_origin,long y_origin,int x_cursor,int y_cursor,long cenx,long ceny,float ratio,int mode)
{
   static int oldx_cursor,oldy_cursor;
   setcolor(LIGHTRED);
   setwritemode(XOR_PUT);
   if(mode==ON)
   {
      line(g_x(cenx,x_origin,ratio),g_y(ceny,y_origin,ratio),
           x_cursor,y_cursor);
      oldx_cursor=x_cursor;
      oldy_cursor=y_cursor;
   }
   if(mode==MOVE)
   {
      line(g_x(cenx,x_origin,ratio),g_y(ceny,y_origin,ratio),
           oldx_cursor,oldy_cursor);
      line(g_x(cenx,x_origin,ratio),g_y(ceny,y_origin,ratio),
           x_cursor,y_cursor);
      oldx_cursor=x_cursor;
      oldy_cursor=y_cursor;
   }
   if(mode==OFF)
   {
      line(g_x(cenx,x_origin,ratio),g_y(ceny,y_origin,ratio),
           oldx_cursor,oldy_cursor);
   }
}
void twoline_adjust(int d1,int d2)
{
   int i,i2,di;
   long x,y,s1,s2;
   if(d1==d2)return;
   if(d1>d2)i=d1,d1=d2,d2=i;
   s1=s2=0;
   i=d1;
   while(i<=d2)
   {
      s1=labs(X[i+1]-X[i])+labs(Y[i+1]-Y[i])+s1;
      i++;
   }
   i=d1;
   while(i>0)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    {
       s2=labs(X[i]-X[i-1])+labs(Y[i]-Y[i-1])+s2;
       i--;
    }
    i=d2;
    while((X[i]!=0)&&(Y[i]!=0))
    {
       s2=labs(X[i+1]-X[i])+labs(Y[i+1]-Y[i])+s2;
       i++;
    }
    if(s1<s2)
    {
       i = d1+1;
       di=(d2-d1)-1;
       while(((X[i]!=0)||(Y[i]!=0))&&get_key()!=0x1b)
       {
          X[i]=X[i+di];
          Y[i]=Y[i+di];
          i++;
       }
    }
    else
    {
       i2 = d2+1;
       while((X[i2]!=0)||(Y[i2]!=0))
       {
          X[i2]=0;Y[i2]=0;
          i2++;
       }
       i=0;i2=d1;
       while((X[i]!=0)||(Y[i]!=0))
       {
          X[i]=X[i+i2];Y[i]=Y[i+i2];
          i++;
       }
    }
}
```

โปรแกรมแก้ไขมุมของข้อมูลแบบอ้างอิง 3 จุด

```
/************** 3LINE.C **************/
void threeline_edit(int *x_cursor,int *y_cursor,
        long *cenx,long *ceny,float *ratio,int *start_edit,int *end_edit);
void threeline_line(long x_origin,long y_origin,long x_second,long y_second,int x_cursor,int y_cursor,long cenx,long ceny,float ratio,int mode);
void threeline_adjust(int d1,int d2,long x_third,long y_third);

void threeline_edit(int *x_cursor,int *y_cursor,
        long *cenx,long *ceny,float *ratio,int *start_edit,int *end_edit)
{
   char  *Menu[]={"1","23",
            " HELP       "," 1001 ",
            " FULL SCREEN "," 1002 ",
            " EDIT SCREEN "," 1003 ",
            " REDRAW     "," 1004 ",
            " REDOT      "," 1005 ",
            " ZOOMIN     "," 1006 ",
            " ZOOMOUT    "," 1007 ",
            " PAN   OFF  "," 1008 ",
            "          "," 1009 ",
            "          "," 1010 ",
            "          "," 1011 ",
            " 3 LINE ADJ. "," 1012 ",
            "          "," 1013 ",
            "          "," 1014 ",
            "          "," 1015 ",
            "          "," 1016 ",
            "          "," 1017 ",
            "          "," 1018 ",
            "          "," 1019 ",
            "          "," 1020 ",
            "          "," 1021 ",
            "          "," 1022 ",
            " EXIT       "," 1023 ",};
   int  key_code,menu_code;
   int  pan_mode,current_mode;
   long x_origin,y_origin,x_second,y_second,x_third,y_third;
   int  data1,data2;
   long minx,miny,maxx,maxy;
   long maxx_edit,maxy_edit;
   maxx_edit=(long)(750*step_ratio);
   maxy_edit=(long)(554*step_ratio);
   getmaxmin_data(&minx,&miny,&maxx,&maxy);
   x_origin=x_second=x_third=y_origin=y_second=y_third=0;
   pan_mode = OFF;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
current_mode = ON;
while(current_mode==ON)
{
   while(getuser(x_cursor,y_cursor,&key_code)==0x00);
   if(*x_cursor==MaxX)
   {
      if(x_origin!=0) threeline_line(x_origin,y_origin,x_second,y_second,*x_cursor,
                                          *y_cursor,*cenx,*ceny,*ratio,OFF);
      show_pointer(*x_cursor,*y_cursor,OFF);
      menu_code=get_menu(Menu,x_cursor,y_cursor);
      if(menu_code!=0x00)key_code=atoi(Menu[menu_code*2+1]);
      show_pointer(*x_cursor,*y_cursor,ON);
      if(x_origin!=0) threeline_line(x_origin,y_origin,x_second,y_second,*x_cursor,
                                          *y_cursor,*cenx,*ceny,*ratio,ON);
   }
   if(key_code!=0x00)
   switch(key_code)
   {
      case 1002 : adjust_screen(cenx,ceny,ratio,0,0,maxx_edit,maxy_edit);
                  redraw_edit(*cenx,*ceny,*ratio,*start_edit,*end_edit);
                  show_pointer(*x_cursor,*y_cursor,ON);
                  if(x_origin!=0x00)
                     threeline_line(x_origin,y_origin,x_second,y_second,*x_cursor,*y_cursor,*cenx,*ceny,*ratio,ON);
                  break;
      case 1003 :    getmaxmin_data(&minx,&miny,&maxx,&maxy);
                     adjust_screen(cenx,ceny,ratio,minx,miny,maxx,maxy);
                  redraw_edit(*cenx,*ceny,*ratio,*start_edit,*end_edit);
                  show_pointer(*x_cursor,*y_cursor,ON);
                  if(x_origin!=0x00)
                     threeline_line(x_origin,y_origin,x_second,y_second,*x_cursor,*y_cursor,*cenx,*ceny,*ratio,ON);
                  break;
      case 1004 :;
      case 'R'  :;
      case 'r'  : redraw_edit(*cenx,*ceny,*ratio,*start_edit,*end_edit);
                  show_pointer(*x_cursor,*y_cursor,ON);
                  if(x_origin!=0x00)
                     threeline_line(x_origin,y_origin,x_second,y_second,*x_cursor,*y_cursor,*cenx,*ceny,*ratio,ON);
                  break;
      case 1005 : redot_edit(*cenx,*ceny,*ratio,*start_edit,*end_edit);
                  show_pointer(*x_cursor,*y_cursor,ON);
                  if(x_origin!=0x00)
                     threeline_line(x_origin,y_origin,x_second,y_second,*x_cursor,*y_cursor,*cenx,*ceny,*ratio,ON);
                  break;
      case 1006 :;
      case 'Z'  : *ratio=*ratio*4.0;      /* Zoom In */
      case 1007 :;
      case 'z'  :  *ratio=*ratio*0.5;    /* Zoom Out */
                  redraw_edit(*cenx,*ceny,*ratio,*start_edit,*end_edit);
                  show_pointer(*x_cursor,*y_cursor,ON);
                  if(x_origin!=0x00)
                     threeline_line(x_origin,y_origin,x_second,y_second,*x_cursor,*y_cursor,*cenx,*ceny,*ratio,ON);
                  break;
      case 1008   : if(pan_mode==OFF){pan_mode=ON;
                     strcpy(Menu[(key_code-1000)*2]," PAN   ON  ");}
                  else{pan_mode=OFF;strcpy(Menu[(key_code-1000)*2]," PAN   OFF  ");}
                  break;
      case 1012 : current_mode = OFF;
                  break;
      case 1023 :;
      case 'Q'  :;
      case 'q'  : current_mode = OFF; break;
      case 0x0231:;
      case ESC  : if(x_origin!=0)threeline_line(x_origin,y_origin,x_second,y_second,
                     *x_cursor,*y_cursor,*cenx,*ceny,*ratio,OFF);
                  x_origin=x_second=0;
                  break;
      case 0x0d :;
      case 0x0211 : if(pan_mode==ON)
                  {*cenx=g_xl(*cenx,*x_cursor,*ratio);
                  *ceny=g_yl(*ceny,*y_cursor,*ratio);
                  redraw_edit(*cenx,*ceny,*ratio,*start_edit,*end_edit);
                  show_pointer(*x_cursor,*y_cursor,ON);
                  if(x_origin!=0x00)
                     threeline_line(x_origin,y_origin,x_second,y_second,*x_cursor,*y_cursor,*cenx,*ceny,*ratio,ON);
                  break;}
                  if(x_origin==0x00)
                  {
                     x_origin=g_xl(*cenx,*x_cursor,*ratio);
                     y_origin=g_yl(*ceny,*y_cursor,*ratio);
                     if(check_point(*ratio,&x_origin,&y_origin,&data1)==OK)
                     {
                        show_pointer(*x_cursor,*y_cursor,OFF);
                        show_pointer(*x_cursor,*y_cursor,ON);
                        x_second=y_second=0;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                                threeline_line(x_origin,y_origin,x_second,y_second,*x_cursor,*y_cursor,*cenx,*ceny,*ratio,ON);
                                break;
                            }
                            x_origin=0;
                            y_origin=0;
                            putch(0x07);
                            break;
                        }
                        if(x_second==0x00)
                        {
                            x_second=g_xl(*cenx,*x_cursor,*ratio);
                            y_second=g_yl(*ceny,*y_cursor,*ratio);
                            if(check_point(*ratio,&x_second,&y_second,&data2)==OK)
                            {
                                threeline_line(x_origin,y_origin,0,0,*x_cursor,*y_cursor,*cenx,*ceny,*ratio,OFF);
                                show_pointer(*x_cursor,*y_cursor,OFF);
                                show_pointer(*x_cursor,*y_cursor,ON);
                                threeline_line(x_origin,y_origin,x_second,y_second,*x_cursor,*y_cursor,*cenx,*ceny,*ratio,ON);
                                break;
                            }
                            x_second=0;
                            y_second=0;
                            putch(0x07);
                            break;
                        }
                        if(x_third==0x00)
                        {
                            x_third=g_xl(*cenx,*x_cursor,*ratio);
                            y_third=g_yl(*ceny,*y_cursor,*ratio);
                            threeline_adjust(data1,data2,x_third,y_third);
                            redraw_edit(*cenx,*ceny,*ratio,*start_edit,*end_edit);
                            show_pointer(*x_cursor,*y_cursor,ON);
                            x_origin=x_second=0,x_third=0;
                            break;
                        }break;
                }
                else
                {
                    if(x_origin!=0)threeline_line(x_origin,y_origin,x_second,y_second,*x_cursor,*y_cursor,
                                                  *cenx,*ceny,*ratio,OFF);
                        show_pointer(*x_cursor,*y_cursor,OFF);
                        coordinate_display( g_xl(*cenx,*x_cursor,*ratio)/step_ratio,
                                            g_yl(*ceny,*y_cursor,*ratio)/step_ratio);
                        show_pointer(*x_cursor,*y_cursor,ON);
                    if(x_origin!=0)threeline_line(x_origin,y_origin,x_second,y_second,*x_cursor,*y_cursor,
                                                  *cenx,*ceny,*ratio,ON);
                }
                key_code=0;
            }
        }
        void threeline_line(long x_origin,long y_origin,long x_second,long y_second,int x_cursor,int y_cursor,long cenx,long ceny,float ratio,int mode)
        {
            static int oldx_cursor,oldy_cursor;
            setcolor(LIGHTRED);
            setwritemode(XOR_PUT);
            if(x_second==0)
            {
                if(mode==ON)
                {
                    line(g_x(cenx,x_origin,ratio),g_y(ceny,y_origin,ratio),
                         x_cursor,y_cursor);
                    oldx_cursor=x_cursor;
                    oldy_cursor=y_cursor;
                }
                if(mode==MOVE)
                {
                    line(g_x(cenx,x_origin,ratio),g_y(ceny,y_origin,ratio),
                         oldx_cursor,oldy_cursor);
                    line(g_x(cenx,x_origin,ratio),g_y(ceny,y_origin,ratio),
                         x_cursor,y_cursor);
                    oldx_cursor=x_cursor;
                oldy_cursor=y_cursor;
                }
                if(mode==OFF)
                {
                    line(g_x(cenx,x_origin,ratio),g_y(ceny,y_origin,ratio),
                         oldx_cursor,oldy_cursor);
                }
            }
            else
            {
                if(mode==ON)
                {
                    line(g_x(cenx,x_origin,ratio),g_y(ceny,y_origin,ratio),
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                x_cursor,y_cursor);
            line(g_x(cenx,x_second,ratio),g_y(ceny,y_second,ratio),
                x_cursor,y_cursor);
            oldx_cursor=x_cursor;
            oldy_cursor=y_cursor;
        }
        if(mode==MOVE)
        {
            line(g_x(cenx,x_origin,ratio),g_y(ceny,y_origin,ratio),
                oldx_cursor,oldy_cursor);
            line(g_x(cenx,x_second,ratio),g_y(ceny,y_second,ratio),
                oldx_cursor,oldy_cursor);
            line(g_x(cenx,x_origin,ratio),g_y(ceny,y_origin,ratio),
                x_cursor,y_cursor);
            line(g_x(cenx,x_second,ratio),g_y(ceny,y_second,ratio),
                x_cursor,y_cursor);
            oldx_cursor=x_cursor;
        oldy_cursor=y_cursor;
        }
        if(mode==OFF)
        {
            line(g_x(cenx,x_origin,ratio),g_y(ceny,y_origin,ratio),
                oldx_cursor,oldy_cursor);
            line(g_x(cenx,x_second,ratio),g_y(ceny,y_second,ratio),
                oldx_cursor,oldy_cursor);
        }
    }
}
void threeline_adjust(int d1,int d2,long x_third,long y_third)
{
    int i,i2,di;
    int x_bak1,y_bak1,x_bak2,y_bak2;
    long x,y,s1,s2;
    if(d1>d2)i=d1,d1=d2,d2=i;
    s1=s2=0;
    i=d1;
    while(i<=d2)
    {
        s1=labs(X[i+1]-X[i])+labs(Y[i+1]-Y[i])+s1;
        i++;
    }
    i=d1;
    while(i>0)
    {
        s2=labs(X[i]-X[i-1])+labs(Y[i]-Y[i-1])+s2;
        i--;
    }
    i=d2;
    while((X[i]!=0)&&(Y[i]!=0))
    {
        s2=labs(X[i+1]-X[i])+labs(Y[i+1]-Y[i])+s2;
        i++;
    }
    if(s1<s2)
    {
        i = d1+1;
        di= (d2-d1)-2;
        if(di>=0)
        {
            X[i]=x_third;Y[i]=y_third;
            i++;
            while(((X[i]!=0)||(Y[i]!=0))&&get_key()!=0x1b)
            {
                X[i]=X[i+di];
                Y[i]=Y[i+di];
                i++;
            }
        }
        else
        {
            x_bak1=X[i];
            y_bak1=Y[i];
            X[i]=x_third,Y[i]=y_third;
            while(((X[i]!=0)||(Y[i]!=0))&&get_key()!=0x1b)
            {
                x_bak2=x_bak1;
                y_bak2=y_bak1;
                x_bak1=X[i+1];
                y_bak1=Y[i+1];
                X[i+1]=x_bak2;
                Y[i+1]=y_bak2;
                i++;
            }
        }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    }
    else
    {
       i2 = d2+1;
       while((X[i2]!=0)||(Y[i2]!=0))
       {
          X[i2]=0;Y[i2]=0;
          i2++;
       }
       i=0;i2=d1;
       while((X[i]!=0)||(Y[i]!=0))
       {
          X[i]=X[i+i2];Y[i]=Y[i+i2];
          i++;
       }
       X[0]=x_third;
       Y[0]=y_third;
    }
}
```

โปรแกรมแก้ไขส่วนโค้งวงกลมของข้อมูลแบบอ้างอิง 3 จุด

```
/********************* 3CIRCLE.C ********************/
void circle3_edit(int *x_cursor,int *y_cursor,
        long *cenx,long *ceny,float *ratio,int *start_edit,int *end_edit);
void circle3_line(long x_origin,long y_origin,long x_second,long y_second,int x_cursor,int y_cursor,long cenx,long ceny,float ratio,int mode);
void circle3_adjust(int d1,int d2,long x_third,long y_third);
int  P3circle(long *r0,long *x0,long *y0,long *x1,long *y1,float *zeta1,long *x2,long *y2,float *zeta2,long x3,long y3,float *zeta3);

int  P3circle(long *r0,long *x0,long *y0,long *x1,long *y1,float *zeta1,long *x2,long *y2,float *zeta2,long x3,long y3,float *zeta3)
{
    float a,b,c,s;
    float x,y,r,dl;
    float zeta4;
    long  d;
    a    = sqrt((*x1-*x2)*(*x1-*x2)+(*y1-*y2)*(*y1-*y2));
    b    = sqrt((*x2- x3)*(*x2- x3)+(*y2- y3)*(*y2- y3));
    c    = sqrt((*x1- x3)*(*x1- x3)+(*y1- y3)*(*y1- y3));
    s    = (a+b+c)/2;
    r    = (4*sqrt(s*(s-a)*(s-b)*(s-c)));
    if(r==0)
       return 0;
    r    = (a*b*c)/(4*sqrt(s*(s-a)*(s-b)*(s-c)));
    *zeta1 = atan2((*y2-*y1),(*x2-*x1));
    *zeta2 = (a/(2.0*r));
    if(*zeta2>=1)*zeta2 = 1;
    *zeta2 = acos(*zeta2);
    x    = *x1+r*cos(*zeta1+*zeta2);
    y    = *y1+r*sin(*zeta1+*zeta2);
    dl   = sqrt((x3-x)*(x3-x)+(y3-y)*(y3-y));
    if((dl>(r*1.001))||(dl<(r*.999)))
    {
       x    = *x1+r*cos(*zeta1-*zeta2);
       y    = *y1+r*sin(*zeta1-*zeta2);
    }
    *x0  = (long)(x);
    *y0  = (long)(y);
    *r0  = (long)(r);
    printf("\r*x=%ld *y= %ld *r= %ld ",*x0,*y0,*r0);
    *zeta1 = atan2((*y1-*y0),(*x1-*x0))*57.29577951;
    *zeta2 = atan2((*y2-*y0),(*x2-*x0))*57.29577951;
    *zeta3 = atan2(( y3-*y0),( x3-*x0))*57.29577951;
    while(*zeta3<*zeta2){*zeta3=*zeta3+360;}
    while(*zeta1<*zeta2){*zeta1=*zeta1+360;}
/*  if(*zeta3>*zeta1)
    {
        zeta4 =  *zeta1;
       *zeta1 =  *zeta2;
       *zeta2 =  zeta4;
       d=*x1;*x1=*x2;*x2=d;
       d=*y1;*y1=*y2;*y2=d;
    }*/
    return 1;
}


void  circle3_edit(int *x_cursor,int *y_cursor,
        long *cenx,long *ceny,float *ratio,int *start_edit,int *end_edit)
{
    char  *Menu[]={"1","23",
              " HELP       "," 1001 ",
              " FULL SCREEN "," 1002 ",
              " EDIT SCREEN "," 1003 ",
              " REDRAW     "," 1004 ",
```

เอกสารนี้สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        " REDOT     "," 1005 ",
        " ZOOMIN    "," 1006 ",
        " ZOOMOUT   "," 1007 ",
        " PAN   OFF "," 1008 ",
        "           "," 1009 ",
        "           "," 1010 ",
        "           "," 1011 ",
        "           "," 1012 ",
        " 3 CIR. ADJ. "," 1013 ",
        "           "," 1014 ",
        "           "," 1015 ",
        "           "," 1016 ",
        "           "," 1017 ",
        "           "," 1018 ",
        "           "," 1019 ",
        "           "," 1020 ",
        "           "," 1021 ",
        "           "," 1022 ",
        " EXIT      "," 1023 ",};
int key_code,menu_code;
int pan_mode,current_mode;
long x_origin,y_origin,x_second,y_second,x_third,y_third;
int data1,data2;
long minx,miny,maxx,maxy;
long maxx_edit,maxy_edit;
maxx_edit=(long)(750*step_ratio);
maxy_edit=(long)(554*step_ratio);
getmaxmin_data(&minx,&miny,&maxx,&maxy);
x_origin=x_second=x_third=y_origin=y_second=y_third=0;
pan_mode = OFF;
current_mode = ON;
while(current_mode==ON)
{
   while(getuser(x_cursor,y_cursor,&key_code)==0x00);
   if(*x_cursor==MaxX)
   {
      if(x_origin!=0) circle3_line(x_origin,y_origin,x_second,y_second,*x_cursor,
                                   *y_cursor,*cenx,*ceny,*ratio,OFF);
      show_pointer(*x_cursor,*y_cursor,OFF);
      menu_code=get_menu(Menu,x_cursor,y_cursor);
      if(menu_code!=0x00)key_code=atoi(Menu[menu_code*2+1]);
      show_pointer(*x_cursor,*y_cursor,ON);
      if(x_origin!=0) circle3_line(x_origin,y_origin,x_second,y_second,*x_cursor,
                                   *y_cursor,*cenx,*ceny,*ratio,ON);
   }
   if(key_code!=0x00)
   switch(key_code)
   {
      case 1002 : adjust_screen(cenx,ceny,ratio,0,0,maxx_edit,maxy_edit);
                  redot_edit(*cenx,*ceny,*ratio,*start_edit,*end_edit);
                  show_pointer(*x_cursor,*y_cursor,ON);
                  if(x_origin!=0x00)
                     circle3_line(x_origin,y_origin,x_second,y_second,*x_cursor,*y_cursor,*cenx,*ceny,*ratio,ON);
                  break;
      case 1003 :    getmaxmin_data(&minx,&miny,&maxx,&maxy);
                  adjust_screen(cenx,ceny,ratio,minx,miny,maxx,maxy);
                  redot_edit(*cenx,*ceny,*ratio,*start_edit,*end_edit);
                  show_pointer(*x_cursor,*y_cursor,ON);
                  if(x_origin!=0x00)
                     circle3_line(x_origin,y_origin,x_second,y_second,*x_cursor,*y_cursor,*cenx,*ceny,*ratio,ON);
                  break;
      case 1004 :;
      case 'R'  :;
      case 'r'  : redraw_edit(*cenx,*ceny,*ratio,*start_edit,*end_edit);
                  show_pointer(*x_cursor,*y_cursor,ON);
                  if(x_origin!=0x00)
                     circle3_line(x_origin,y_origin,x_second,y_second,*x_cursor,*y_cursor,*cenx,*ceny,*ratio,ON);
                  break;
      case 1005   : redot_edit(*cenx,*ceny,*ratio,*start_edit,*end_edit);
                  show_pointer(*x_cursor,*y_cursor,ON);
                  if(x_origin!=0x00)
                     circle3_line(x_origin,y_origin,x_second,y_second,*x_cursor,*y_cursor,*cenx,*ceny,*ratio,ON);
                  break;
      case 1006 :;
      case 'Z'  : *ratio=*ratio*4.0;      /* Zoom In */
      case 1007 :;
      case 'z'  :   *ratio=*ratio*0.5;    /* Zoom Out */
                  redot_edit(*cenx,*ceny,*ratio,*start_edit,*end_edit);
                  show_pointer(*x_cursor,*y_cursor,ON);
                  if(x_origin!=0x00)
                     circle3_line(x_origin,y_origin,x_second,y_second,*x_cursor,*y_cursor,*cenx,*ceny,*ratio,ON);
                  break;
      case 1008   : if(pan_mode==OFF){pan_mode=ON;
                  strcpy(Menu[(key_code-1000)*2]," PAN   ON  ");}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                                else{pan_mode=OFF;strcpy(Menu[(key_code-1000)*2]," PAN   OFF ");}
                                break;
        case 1013       : current_mode = OFF;
                                break;
        case 1023 :;
        case 'Q'    :;
        case 'q'            : current_mode = OFF; break;
        case 0x0231:;
        case ESC        : if(x_origin!=0)circle3_line(x_origin,y_origin,x_second,y_second,
                                *x_cursor,*y_cursor,*cenx,*ceny,*ratio,OFF);
                                x_origin=x_second=0;
                                break;
        case 0x0d :;
        case 0x0211     : if(pan_mode==ON)
                                {*cenx=g_xl(*cenx,*x_cursor,*ratio);
                                *ceny=g_yl(*ceny,*y_cursor,*ratio);
                                redot_edit(*cenx,*ceny,*ratio,*start_edit,*end_edit);
                                show_pointer(*x_cursor,*y_cursor,ON);
                                if(x_origin!=0x00)
                                   circle3_line(x_origin,y_origin,x_second,y_second,*x_cursor,*y_cursor,*cenx,*ceny,*ratio,ON);
                                break;}
                                if(x_origin==0x00)
                                {
                                   x_origin=g_xl(*cenx,*x_cursor,*ratio);
                                   y_origin=g_yl(*ceny,*y_cursor,*ratio);
                                   if(check_point(*ratio,&x_origin,&y_origin,&data1)==OK)
                                   {
                                      show_pointer(*x_cursor,*y_cursor,OFF);
                                      show_pointer(*x_cursor,*y_cursor,ON);
                                      x_second=y_second=0;
                                      circle3_line(x_origin,y_origin,x_second,y_second,*x_cursor,*y_cursor,*cenx,*ceny,*ratio,ON);
                                      break;
                                   }
                                   x_origin=0;
                                   y_origin=0;
                                   putch(0x07);
                                   break;
                                }
                                if(x_second==0x00)
                                {
                                   x_second=g_xl(*cenx,*x_cursor,*ratio);
                                   y_second=g_yl(*ceny,*y_cursor,*ratio);
                                   if(check_point(*ratio,&x_second,&y_second,&data2)==OK)
                                   {
                                      circle3_line(x_origin,y_origin,0,0,*x_cursor,*y_cursor,*cenx,*ceny,*ratio,OFF);
                                      show_pointer(*x_cursor,*y_cursor,OFF);
                                      show_pointer(*x_cursor,*y_cursor,ON);
                                      circle3_line(x_origin,y_origin,x_second,y_second,*x_cursor,*y_cursor,*cenx,*ceny,*ratio,ON);
                                      break;
                                   }
                                   x_second=0;
                                   y_second=0;
                                   putch(0x07);
                                   break;
                                }
                                if(x_third==0x00)
                                {
                                   x_third=g_xl(*cenx,*x_cursor,*ratio);
                                   y_third=g_yl(*ceny,*y_cursor,*ratio);
                                   circle3_adjust(data1,data2,x_third,y_third);
                                   redot_edit(*cenx,*ceny,*ratio,*start_edit,*end_edit);
                                   show_pointer(*x_cursor,*y_cursor,ON);
                                   x_origin=x_second=0,x_third=0;
                                   break;
                                }break;
    }
    else
    {
       if(x_origin!=0)circle3_line(x_origin,y_origin,x_second,y_second,*x_cursor,*y_cursor,
                                *cenx,*ceny,*ratio,OFF);
          show_pointer(*x_cursor,*y_cursor,OFF);
          coordinate_display( g_xl(*cenx,*x_cursor,*ratio)/step_ratio,
                                g_yl(*ceny,*y_cursor,*ratio)/step_ratio);
          show_pointer(*x_cursor,*y_cursor,ON);
       if(x_origin!=0)circle3_line(x_origin,y_origin,x_second,y_second,*x_cursor,*y_cursor,
                                *cenx,*ceny,*ratio,ON);
    }
    key_code=0;
  }
}
void circle3_line(long x_origin,long y_origin,long x_second,long y_second,int x_cursor,int y_cursor,long cenx,long ceny,float ratio,int mode)
{
  static int oldx_cursor,oldy_cursor;
  static int oldx,oldy,oldr,oldzeta1,oldzeta2,oldzeta3;
```

เอกสารนี้มีไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    long  r0,x0,y0,x1,y1,x2,y2,x3,y3;
    float zeta1,zeta2,zeta3,zeta4;
    int   x,y,r;
    x1 = x_origin; y1 = y_origin;
    x2 = x_second; y2 = y_second;
    x3 = g_x1(cenx,x_cursor,ratio);
    y3 = g_y1(ceny,y_cursor,ratio);
    setcolor(LIGHTRED);
    setwritemode(XOR_PUT);
    if(x_second==0)
    {
       if(mode==ON)
       {
          line(g_x(cenx,x_origin,ratio),g_y(ceny,y_origin,ratio),
               x_cursor,y_cursor);
          oldx_cursor=x_cursor;
          oldy_cursor=y_cursor;
       }
       if(mode==MOVE)
       {
          line(g_x(cenx,x_origin,ratio),g_y(ceny,y_origin,ratio),
               oldx_cursor,oldy_cursor);
          line(g_x(cenx,x_origin,ratio),g_y(ceny,y_origin,ratio),
               x_cursor,y_cursor);
          oldx_cursor=x_cursor;
       oldy_cursor=y_cursor;
       }
       if(mode==OFF)
       {
          line(g_x(cenx,x_origin,ratio),g_y(ceny,y_origin,ratio),
               oldx_cursor,oldy_cursor);
       }
    }
    else
    if(P3circle(&r0,&x0,&y0,&x1,&y1,&zeta1,&x2,&y2,&zeta2,x3,y3,&zeta3)==OK)
    {
       x = g_x(cenx,x0,ratio);
       y = g_y(ceny,y0,ratio);
       r = (int)((float)(r0)*ratio);
       printf(" x=%d y=%d r=%d",x,y,r);
/*     zeta1 = atan2((y1-y0),(x1-x0))*57.29577951;
       zeta2 = atan2((y2-y0),(x2-x0))*57.29577951;
       zeta3 = atan2((y3-y0),(x3-x0))*57.29577951;
       while(zeta3<zeta2){zeta3=zeta3+360;}
       while(zeta1<zeta2){zeta1=zeta1+360;}*/
       if(zeta3>zeta1)
       {
          zeta4 = zeta1;
          zeta1 = zeta2;
          zeta2 = zeta4;
       }
       if(mode==ON)
       {
          oldx = x; oldy = y; oldr = r;
          oldzeta1 = (int)zeta1; oldzeta2 = (int)zeta2; oldzeta3 = (int)zeta3;
          setcolor(LIGHTRED);
          arc(oldx,oldy,oldzeta2,oldzeta3,oldr);
          arc(oldx,oldy,oldzeta3,oldzeta1,oldr);
       }
       if(mode==MOVE)
       {
          setcolor(BLACK);
          arc(oldx,oldy,oldzeta2,oldzeta3,oldr);
          arc(oldx,oldy,oldzeta3,oldzeta1,oldr);
          oldx = x; oldy = y; oldr = r;
          oldzeta1 = (int)zeta1; oldzeta2 = (int)zeta2; oldzeta3 = (int)zeta3;
          setcolor(LIGHTRED);
          arc(oldx,oldy,oldzeta2,oldzeta3,oldr);
          arc(oldx,oldy,oldzeta3,oldzeta1,oldr);
       }
       if(mode==OFF)
       {
          setcolor(BLACK);
          arc(oldx,oldy,oldzeta2,oldzeta3,oldr);
          arc(oldx,oldy,oldzeta3,oldzeta1,oldr);
       }
    }
}

void circle3_adjust(int d1,int d2,long x3,long y3)
{
   int  i,i2,di;
   int  x_bak1,y_bak1,x_bak2,y_bak2,number;
   long r0,x0,y0,x1,y1,x2,y2;
```

เอกสารนี้สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    float zeta1,zeta2,zeta3,zeta4,dzeta,sr;
    long  x,y,s1,s2;
    if(d1>d2)i=d1,d1=d2,d2=i;
    x1 = X[d1]; y1 = Y[d1];
    x2 = X[d2]; y2 = Y[d2];
    if(P3circle(&r0,&x0,&y0,&x1,&y1,&zeta1,&x2,&y2,&zeta2,x3,y3,&zeta3)==OK);
    if(zeta3<zeta1)
    {
       sr    = 2*3.1415926536*r0*(zeta1-zeta2)/360;
       number = abs((int)(sr/100)+1);
       dzeta  = (zeta2-zeta1)/number;
    }
    if(zeta3>zeta1)
    {
       sr    = 2*3.1415926536*r0*(360-zeta1+zeta2)/360;
       number = abs((int)(sr/100)+1);
       dzeta  = (360-zeta1+zeta2)/(float)number;
    }
/*  printf("\n r0= %ld zeta1= %f zeta2= %f sr = %f num= %d dzeta= %f",r0,zeta1,zeta2,sr,number,dzeta);getch();*/
    s1 = s2 = 0;
    i  = d1;
    while(i<=d2)
    {
       s1=labs(X[i+1]-X[i])+labs(Y[i+1]-Y[i])+s1;
       i++;
    }
    i=d1;
    while(i>0)
    {
       s2=labs(X[i]-X[i-1])+labs(Y[i]-Y[i-1])+s2;
       i--;
    }
    i=d2;
    while((X[i]!=0)&&(Y[i]!=0))
    {
       s2=labs(X[i+1]-X[i])+labs(Y[i+1]-Y[i])+s2;
       i++;
    }
    if(s1>s2)
    {
       i2 = d2+1;
       while((X[i2]!=0)||(Y[i2]!=0))
       {
          X[i2]=0;Y[i2]=0;
          i2++;
       }
       cut_data(1,d1);
       X[0]=x1;
       Y[0]=y1;
       d1=d2-d1;
/*     add_data(d1,number);*/
       dzeta=-dzeta;
       zeta1=zeta2;
    }
    if(s1<s2)
    {
       cut_data(d1+1,d2-1);
       add_data(d1,number);
    }
    zeta4=zeta1*0.01745329252;
    dzeta=dzeta*0.01745329252;
    i=0;
    while(i<number)
    {
       zeta4=zeta4+dzeta;
       X[d1+1+i] = x0+(long)(r0*cos(zeta4));
       Y[d1+1+i] = y0+(long)(r0*sin(zeta4));
       i++;
    }
/*  if(s1<s2)
    {
       cut_data(d1+1,d2-1);
       add_data(d1,number);
       zeta4=zeta1*0.01745329252;
       dzeta=dzeta*0.01745329252;
       i=0;
       while(i<number)
       {
          zeta4=zeta4+dzeta;
          X[d1+1+i] = x0+(long)(r0*cos(zeta4));
          Y[d1+1+i] = y0+(long)(r0*sin(zeta4));
       }
    }*/
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรมปรับขนาดขยายของข้อมูล

```
/************************* ADJSIZE.C ****************************/
void adjsize_edit(int *x_cursor,int *y_cursor,long *cenx,long *ceny,float *ratio,int *start_edit,int *end_edit);
void adjsize_line(long x_origin,long y_origin,int x_cursor,int y_cursor,long cenx,long ceny,float ratio,int mode);
void adjsize_adjust(long x_origin,long y_origin,long x_second,long y_second,long x_end,long y_end);

void adjsize_edit(int *x_cursor,int *y_cursor,long *cenx,long *ceny,float *ratio,int *start_edit,int *end_edit)
{
   char  *Menu[]={"1","23",
           " HELP      "," 1001 ",
           " FULL SCREEN "," 1002 ",
           " EDIT SCREEN "," 1003 ",
           " REDRAW    "," 1004 ",
           " REDOT     "," 1005 ",
           " ZOOMIN    "," 1006 ",
           " ZOOMOUT   "," 1007 ",
           " PAN   OFF "," 1008 ",
           "           "," 1009 ",
           "           "," 1010 ",
           "           "," 1011 ",
           "           "," 1012 ",
           "           "," 1013 ",
           "           "," 1014 ",
           "           "," 1015 ",
           "           "," 1016 ",
           "           "," 1017 ",
           "           "," 1018 ",
           "           "," 1019 ",
           " ADJ SIZE ON "," 1020 ",
           "           "," 1021 ",
           "           "," 1022 ",
           " EXIT      "," 1023 ",};
  int key_code,menu_code;
  int pan_mode,current_mode;
  long x_origin,y_origin,x_second,y_second,x_end,y_end;
  long minx,miny,maxx,maxy;
  long maxx_edit,maxy_edit;
  maxx_edit=(long)(750*step_ratio);
  maxy_edit=(long)(554*step_ratio);
  getmaxmin_data(&minx,&miny,&maxx,&maxy);
  x_origin=x_second=x_end=y_origin=0;
  pan_mode = OFF;
  current_mode = ON;
  while(current_mode==ON)
  {
     while(getuser(x_cursor,y_cursor,&key_code)==0x00);
     if(*x_cursor==MaxX)
     {
        if(x_origin!=0) adjsize_line(x_origin,y_origin,*x_cursor,
                                     *y_cursor,*cenx,*ceny,*ratio,OFF);
        show_pointer(*x_cursor,*y_cursor,OFF);
        menu_code=get_menu(Menu,x_cursor,y_cursor);
        if(menu_code!=0x00)key_code=atoi(Menu[menu_code*2+1]);
        show_pointer(*x_cursor,*y_cursor,ON);
        if(x_origin!=0) adjsize_line(x_origin,y_origin,*x_cursor,
                                     *y_cursor,*cenx,*ceny,*ratio,ON);
     }
     if(key_code!=0x00)
     switch(key_code)
     {
        case  1002  : adjust_screen(cenx,ceny,ratio,0,0,maxx_edit,maxy_edit);
                      redraw_edit(*cenx,*ceny,*ratio,*start_edit,*end_edit);
                      show_pointer(*x_cursor,*y_cursor,ON);
                      break;
        case  1003  :     getmaxmin_data(&minx,&miny,&maxx,&maxy);
                      adjust_screen(cenx,ceny,ratio,minx,miny,maxx,maxy);
                      redraw_edit(*cenx,*ceny,*ratio,*start_edit,*end_edit);
                      show_pointer(*x_cursor,*y_cursor,ON);
                      break;
        case  1004  :;
        case  'R'   :;
        case  'r'   : redraw_edit(*cenx,*ceny,*ratio,*start_edit,*end_edit);
                      show_pointer(*x_cursor,*y_cursor,ON);
                      break;
        case  1005  : redot_edit(*cenx,*ceny,*ratio,*start_edit,*end_edit);
                      show_pointer(*x_cursor,*y_cursor,ON);
                      break;
        case  1006  :;
        case  'Z'   : *ratio=*ratio*4.0;     /* Zoom In */
        case  1007  :;
        case  'z'   : *ratio=*ratio*0.5;     /* Zoom Out */
                      redraw_edit(*cenx,*ceny,*ratio,*start_edit,*end_edit);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                                show_pointer(*x_cursor,*y_cursor,ON);
                                break;
        case 1008       : if(pan_mode==OFF){pan_mode=ON;
                                strcpy(Menu[(key_code-1000)*2]," PAN   ON  ");}
                          else{pan_mode=OFF;strcpy(Menu[(key_code-1000)*2]," PAN   OFF ");}
                          break;
        case 1020     : current_mode = OFF;
                          break;
        case 1023 :;
        case 'Q'   :;
        case 'q'        : current_mode = OFF; break;
        case 0x0231:;
        case ESC      : if(x_origin!=0)adjsize_line(x_origin,y_origin,
                                *x_cursor,*y_cursor,*cenx,*ceny,*ratio,OFF);
                        x_origin=x_second=0;
                        break;
        case 0x0d :;
        case 0x0211     : if(pan_mode==ON)
                          {*cenx=g_xl(*cenx,*x_cursor,*ratio);
                          *ceny=g_yl(*ceny,*y_cursor,*ratio);
                          redraw_edit(*cenx,*ceny,*ratio,*start_edit,*end_edit);
                          show_pointer(*x_cursor,*y_cursor,ON);
                          if(x_origin!=0x00)
                             adjsize_line(x_origin,y_origin,*x_cursor,*y_cursor,*cenx,*ceny,*ratio,ON);
                          break;}
                          if(x_origin==0x00)
                          {
                             x_origin=g_xl(*cenx,*x_cursor,*ratio);
                             y_origin=g_yl(*ceny,*y_cursor,*ratio);
                             show_pointer(*x_cursor,*y_cursor,OFF);
                             show_pointer(*x_cursor,*y_cursor,ON);
                             adjsize_line(x_origin,y_origin,*x_cursor,*y_cursor,*cenx,*ceny,*ratio,ON);
                          }
                          else
                          if(x_second==0x00)
                          {
                             x_second=g_xl(*cenx,*x_cursor,*ratio);
                             y_second=g_yl(*ceny,*y_cursor,*ratio);
                             show_pointer(*x_cursor,*y_cursor,OFF);
                             show_pointer(*x_cursor,*y_cursor,ON);
                          }
                          else
                          {
                             x_end=g_xl(*cenx,*x_cursor,*ratio);
                             y_end=g_yl(*ceny,*y_cursor,*ratio);
                             adjsize_adjust(x_origin,y_origin,x_second,y_second,x_end,y_end);
                             redraw_edit(*cenx,*ceny,*ratio,*start_edit,*end_edit);
                             show_pointer(*x_cursor,*y_cursor,ON);
                             x_origin=x_second=0;
                             y_origin=y_second=0;
                          }break;
      }
      else
      {
         if(x_origin!=0)adjsize_line(x_origin,y_origin,*x_cursor,*y_cursor,
                                     *cenx,*ceny,*ratio,OFF);
         show_pointer(*x_cursor,*y_cursor,OFF);
         coordinate_display( g_xl(*cenx,*x_cursor,*ratio)/step_ratio,
                             g_yl(*ceny,*y_cursor,*ratio)/step_ratio);
         show_pointer(*x_cursor,*y_cursor,ON);
         if(x_origin!=0)adjsize_line(x_origin,y_origin,*x_cursor,*y_cursor,
                                     *cenx,*ceny,*ratio,ON);
      }
      key_code=0;
   }
}
void adjsize_line(long x_origin,long y_origin,int x_cursor,int y_cursor,long cenx,long ceny,float ratio,int mode)
{
   static int oldx_cursor,oldy_cursor;
   setcolor(LIGHTRED);
   setwritemode(XOR_PUT);
   if(mode==ON)
   {
      line(g_x(cenx,x_origin,ratio),g_y(ceny,y_origin,ratio),
           x_cursor,y_cursor);
      oldx_cursor=x_cursor;
      oldy_cursor=y_cursor;
   }
   if(mode==MOVE)
   {
      line(g_x(cenx,x_origin,ratio),g_y(ceny,y_origin,ratio),
           oldx_cursor,oldy_cursor);
      line(g_x(cenx,x_origin,ratio),g_y(ceny,y_origin,ratio),
           x_cursor,y_cursor);
```

เอกสารนี้ใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        oldx_cursor=x_cursor;
        oldy_cursor=y_cursor;
    }
    if(mode==OFF)
    {
        line(g_x(cenx,x_origin,ratio),g_y(ceny,y_origin,ratio),
                oldx_cursor,oldy_cursor);
    }
}

void adjsize_adjust(long x_origin,long y_origin,long x_second,long y_second,long x_end,long y_end)
{
    int i;
    float size1,size2,zeta1,zeta2,zeta3,z;
    long x,y;
    x=x_second-x_origin;
    y=y_second-y_origin;
    size1 = sqrt((x*x)+(y*y));
    x=x_end-x_origin;
    y=y_end-y_origin;
    size2 = sqrt((x*x)+(y*y));
    size2 = size2/size1;
    i=0;
    while((X[i]!=0||Y[i]!=0)&&get_key()!=0x1b)
    {
        x=X[i]-x_origin;y=Y[i]-y_origin;
        X[i]=x_origin+(long)(x*size2);
        Y[i]=y_origin+(long)(y*size2);
        i++;
    }
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรมที่ใช้ในการทดสอบการเคลื่อนที่หัวอ่าน

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

#include<define.c>
#include<control.c>

main()
{
   int i;
   float v,a;
   initport();
   go_home();
   stop();
   delay(1500);
/* getch();*/
   zero_port();
   v=50;
   a=50;
   /* distance test */
/* go_to(50,50,v,a);
   getch();
   go_rel(300,0,v,a);*/
   /* speed test */
/* go_to(50,50,v,a);
   getch();
   set_v(100*step_ratio,50*step_ratio,50*step_ratio);
   sleep(5);
   stop();*/
   /* spiral test */
/* getch();
   spiral_move(100,100,3,3,50,50,0,240,v,a);*/
   /* circle test */
/* getch();
   v=150;a=100;
   arc_move(100,100,45,45,-1.570796327,62.83185307,v,a);*/
/* arc_move(100,100,20,20,0,-12.2832,v,a);
   arc_move(100,100,20,20,1.5,-12.2832,v,a);
   arc_move(100,100,20,20,3,-12.2832,v,a);
   arc_move(100,100,20,20,4,-12.2832,v,a);
   arc_move(100,100,20,20,6,-12.2832,v,a);*/
   /* star & circle */
   getch();
   v=100;a=100;
/* arc_move(100,100,50,50,4.084070450,+1.056637061,v,a);
   arc_move(100,100,50,50,2.827433388,+1.056637061,v,a);
   arc_move(100,100,50,50,1.570796327,+1.056637061,v,a);
   arc_move(100,100,50,50,0.314159265,+1.056637061,v,a);
   arc_move(100,100,50,50,-.942477796,+1.056637061,v,a);
   arc_move(100,100,50,50,4.084070450,+1.056637061,v,a);*/
/* arc_move(100,100,50,50,4.084070450,+1.256637061,v,a);
   arc_move(100,100,50,50,2.827433388,+1.256637061,v,a);
   arc_move(100,100,50,50,1.570796327,+1.256637061,v,a);
   arc_move(100,100,50,50,0.314159265,+1.256637061,v,a);
   arc_move(100,100,50,50,-.942477796,+1.256637061,v,a);
   arc_move(100,100,50,50,4.084070450,+1.256637061,v,a);*/
   /* square */
/* getch();
   v=200;a=15 00;
   go_to(20,40,v,a);
   for(i=0;i<10;i++)
   {
      go_to(130,40,v,a);
      go_to(130,110,v,a);
      go_to(20,110,v,a);
      go_to(20,40,v,a);
   }*/
   /* rhombus */
/* getch();
   v=50;a=50;
   go_to(70,20,v,a);
   for(i=0;i<10;i++)
   {
      go_to(120,70,v,a);
      go_to(70,120,v,a);
      go_to(20,70,v,a);
      go_to(70,20,v,a);
   }*/
   stop();
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ค

บทความวิธีการเพิ่มความละเอียดเชิงมุมของสเต็ปปิงมอเตอร์

ในประชุมวิชาการทาง วิศวกรรมไฟฟ้า 9 สถาบัน ครั้งที่ 11 พ.ศ. 2531

ณ คณะวิศวกรรมเทคโนโลยี สถาบันเทคโนโลยีราชมงคล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

วิธีการเพิ่มความละเอียดเชิงมุมของแกนสเต็ปมอเตอร์

Method of Increasing the Shaft Angle Resolution of the Stepping Motors

สุมิตร พนาอุดมทรัพย์ *
ดร.โยธิน เปรมปราณีรัชต์ **

ภาควิชาวิศวกรรมระบบควบคุม
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้า
เจ้าคุณทหาร ลาดกระบัง

บทคัดย่อ

บทความนี้เป็นการศึกษาการทำงานของสเต็ปมอเตอร์ เพื่อเพิ่มความละเอียดเชิงมุมให้แก่มอเตอร์ โดยอาศัยการเปลี่ยนแปลงกระแสที่ไหลผ่านเฟสต่าง ๆ ให้มีการเปลี่ยนค่าไปทีละน้อย เพื่อให้จุดสมดุลย์ทางไฟฟ้าหรือแรงลัพธ์ ภายในมอเตอร์มีการเคลื่อนมุมไปทีละน้อย ซึ่งจะทำให้แกนของสเต็ปมอเตอร์เคลื่อนที่เชิงมุมเป็นสเต็ปที่มีความละเอียดมากขึ้น ผลการทดลองแสดงการเคลื่อนที่เชิงมุมที่อัตราส่วนต่าง ๆ เช่น 1/2 1/4, 1/8, 1/16 ของสเต็ปเดิม

Abstract

This paper presents the results of the studying the operation behaviour of the stepping motor in order to increasing the shaft angle resolution of a stepping motor. The procedure is successful by gradually changing the value of the current which flows through the coil of the motor phases. Consequently the equilibrium electrical point or the resultional torque in the stepping motor will rotate the motor in small angle. Therefore the stepping motor can rotate the shaft angle at each step with better resolution. The experimental results of controlling the stepping motor in each step angle with various ratio such as 1/2, 1/4, 1/8 and 1/16 of the former step presented.

---

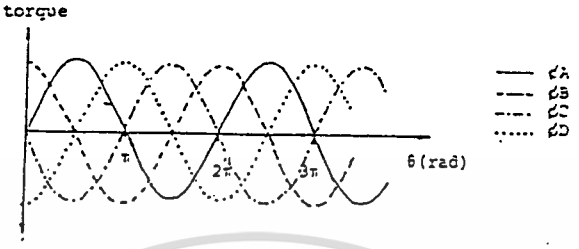
* นักศึกษาทดลองวิจัย คณะวิศวกรรมศาสตร์ ลาดกระบัง

** รองศาสตราจารย์ คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้า เจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

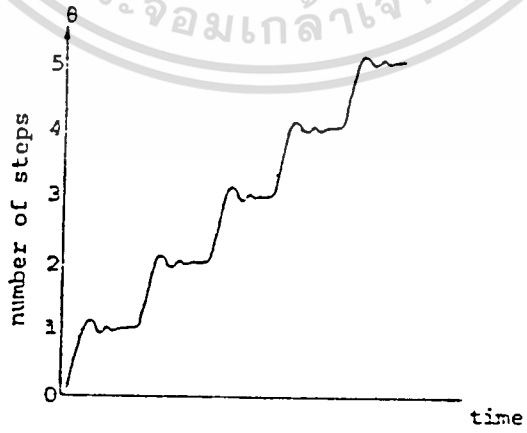
1. บทนำ

สำหรับ stepping motor เมื่อให้สัญญาณไฟฟ้าผ่านขดลวดเส้นใดเส้นหนึ่ง จะเกิด torque ขึ้น ซึ่ง torque ที่ได้จะมีการเปลี่ยนแปลงไปตามมุมของแกนมอเตอร์ ดังรูปที่ 1.1



จากรูปจะพบว่าเมื่อจ่ายกระแสให้เฟส A,B,C และ D ตามลำดับจะเกิดจุดสมดุลย์ที่มุม (2n-1)π ,(2n-1/2)π ,2nπ และ (2n+1/2)π ตามลำดับเช่นกัน จากผลดังกล่าวทำให้แกนมอเตอร์มีการหมุนไปทีละ step และมีโอกาสเกิด osoillate ได้ดังแสดงดังรูปที่ 1.2

รูปที่ 1.2 แสดง single - step response



รูปที่ 1.2 แสดง single -step response

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ผลลัพธ์ที่ได้จากรูปที่ 1.2 จะสังเกตุได้ง่ายเมื่อ stepping motor มีขนาดแรงเฉื่อยของโหลดสูงขึ้น โดยเฉพาะ stepping motor ที่มีค่าองศาต่อ step มาก เช่น 15 /step

จากผลดังกล่าว สามารถสรุปผลเบื้องต้นได้ว่า

1. stepping motor มีการหมุนของแกนที่ไม่ต่อเนื่อง เนื่องจากเป็น step

2. stepping motor เกิดการ damp ได้ง่ายและคาบการ damp จะยาวขึ้นเมื่อมีแรงเฉื่อยมาก

ในการแก้ปัญหานี้ สามารถแก้ได้โดยการลดขนาด องศา/step ของ stepping motor ให้น้อยลง ซึ่งสามารถกระทำได้ 2 วิธี คือ

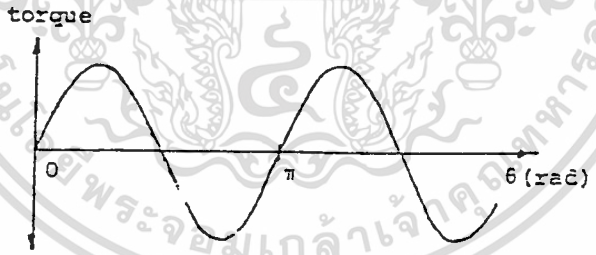
1. พัฒนาทางด้านโครงสร้างของ stepping motor

2. พัฒนาทางด้าน control system

ในกรณีนี้การพัฒนาทางด้าน control system จะสะดวกกว่าอีกทั้งสามารถนำไปประยุกต์ใช้กับ stepping motor ขนาดและชนิดอื่น ๆ ได้

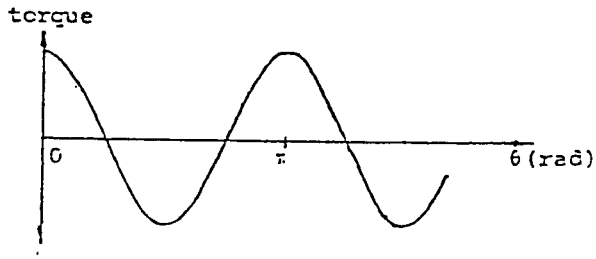
3. หลักการ

สำหรับ permanent and hybrid stepping motor นี้เมื่อมีการจ่ายกระแสไฟฟ้า I_M ผ่านเฟส A จะได้ความสัมพันธ์ดังรูปที่ 2.1



รูปที่ 2.1 แสดงความสัมพันธ์ระหว่าง torque กับ degree เมื่อมีกระแสไหลผ่านเฟส A

เช่นเดียวกันเมื่อมีกระแส I_M ผ่านเฟส B จะได้ความสัมพันธ์ใหม่ดังรูปที่ 2.2



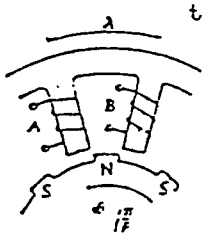
รูปที่ 2.2 แสดงความสัมพันธ์ระหว่าง torque กับ degree เมื่อมีกระแส IM ผ่านเฟส B

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากความสัมพันธ์ดังกล่าวสามารถเขียนรูปแบบสมการเบื้องต้นได้ว่า

สำหรับเฟส A $\quad torque\ A = -\ pn\ \Phi_m\ i_a \sin p\theta$ -------(1)

สำหรับเฟส B $\quad torque\ B = -\ pn\ \Phi_m\ i_b \sin p(\theta - \lambda)$ -------(2)



torque A,B = torque ที่เกิดจากการแส ia , ib

pn = number of pairs of magnetic poles

Φ_m = peak flux linkage

i_a, i_b = กระแสที่ไหลผ่าน เฟส A และ B ตามลำดับ

θ = ตำแหน่งมุมปัจจุบันของแกนโรเตอร์

λ = มุมระหว่างเฟส A และ B

จากข้อมูลที่ได้จากกราฟในรูปที่ 2.1 และ 2.2 นั้นจะพบว่าจุดสมดุลย์จะมีการเคลื่อนตำแหน่งไปทีละ λ ซึ่งเป็นผลให้เกิดการเคลื่อนที่เป็น step ขึ้น ดังนั้นการที่จะทำให้แกนมอเตอร์ มีการหมุนที่ต่อเนื่องมากขึ้นจึงต้องหาทางปรับปรุง ช่วงกว้างระหว่างจุดสมดุลย์ที่อยู่ใกล้กัน 2 จุด ให้มีขนาดแคบลง ซึ่งจะเป็นการลดความกว้างของ step เช่นกัน ผลที่ได้จะทำให้จำนวน step/รอง ของมอเตอร์มีมากขึ้น ทำให้มอเตอร์สามารถอ้างอิงตำแหน่งของมอเตอร์ได้ละเอียดขึ้น

พิจารณา สมการ 1 และ 2 นำมาประยุกต์เขียนรูปแบบสมการง่าย ๆ ขึ้นใหม่โดยกำหนดค่า p = 1 และ λ = 90องศา จะได้ว่า

$$torque\ A = K\ i_a \sin\theta \qquad \text{--------(3)}$$

$$torque\ B = K\ i_b \sin(\theta - \pi/2) \qquad \text{--------(4)}$$

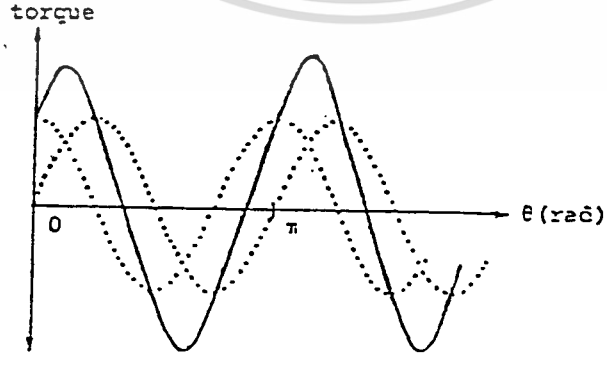
$K = -pn\ \Phi_m$ (ซึ่งเป็นค่าคงที่)

ถ้าให้ i_a และ i_b มีค่าเท่ากับกระแส I_{max} จะได้ผลลัพธ์ของ torque ตังสมการ

$$torque_{sum} = torque\ A + torque\ B$$

$$= K(i_a \sin\theta + i_b \sin(\theta - \pi/2)) \qquad \text{---(5)}$$

นำสมการ 5 มาเขียนกราฟจะได้ดังรูปที่ 2.3



รูปที่ 2.3 แสดงความสัมพันธ์ระหว่าง torque กับ deree เมื่อ $i_a = i_b = I_{max}$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 2.3 จะพบว่า torque จะมีค่า peak สูงขึ้นและจุดสมดุลย์จะอยู่ระหว่าง $\frac{\pi}{2}$ กับ π คือ $\frac{3\pi}{4}$ ซึ่งเป็นตำแหน่งกึ่งกลางพอดี ดังนั้นแสดงว่าการขับมอเตอร์แบบ one-two excitation จะเพิ่มจำนวน step/รอบ ขึ้นเป็น 2 เท่า ซึ่งเรียกกันว่า " การขับแบบ half step "

พิจารณา สมการที่ 5 จะพบว่า ถ้า i_a มีค่าคงที่เท่ากับ I_{max} และ i_b มีค่าระหว่าง 0 ถึง I_{max} จะพบว่า จุดสมดุลย์ย่อมจะมีการเปลี่ยนแปลงระหว่าง θ กับ $\theta-\frac{\pi}{2}$ ซึ่งสามารถหาจุดสมดุลย์ที่เปลี่ยนแปลงตาม i_b ได้ดังนี้

จากสมการที่ 5 เมื่อพิจารณาที่ torque มีค่าเท่ากับศูนย์จะได้ว่า

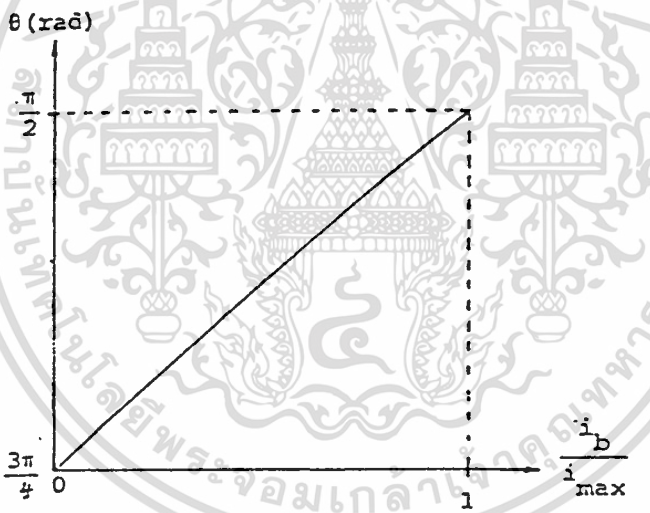
$$i_a \sin\theta = -i_b \sin(\theta - \pi/2)$$

$$= -i_b \cos\theta$$

$$i_b / i_a = -\tan\theta$$

$$\theta = -\tan^{-1}(i_b / i_a) \qquad \text{------(6)}$$

นำสมการที่ 5 มาเขียนกราฟโดยปรับปรุงค่า θ ที่ได้ให้อยู่ในช่วง $\pi/2$ ถึง $3\pi/2$ จะได้ดังรูปที่ 2.4



รูปที่ 2.4 แสดงความสัมพันธ์ระหว่าง ib / imax กับ θ

จากกราฟจะพบว่าจุดสมดุลย์มีการเปลี่ยนแปลงตำแหน่งมุมตาม i_b / i_{max} แสดงว่าในทางปฏิบัติ หากสามารถควบคุมค่า i_b / i_{max} ได้ ย่อมมีทางที่จะควบคุมตำแหน่งจุดสมดุลย์ได้เช่นกัน

จากหลักการดังกล่าวจึงได้ออกแบบวงจรจ่ายกระแสที่สามารถปรับค่าได้ โดยอาศัยการจ่ายแรงดันไฟฟ้าแบบเป็นพัลส์ที่สามารถปรับค่า t_{on} / t_{on+off} ได้เป็นช่วง ๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 2.3 จะพบว่า torque จะมีค่า peak สูงขึ้นและจุดสมดุลย์จะอยู่ระหว่าง $\frac{\pi}{2}$ กับ π คือ $\frac{3\pi}{4}$ ซึ่งเป็นตำแหน่งกึ่งกลางพอดี ดังนั้นแสดงว่าการขับมอเตอร์แบบ one-two excitation จะเพิ่มจำนวน step/รอบ ขึ้นเป็น 2 เท่า ซึ่งเรียกกันว่า " การขับแบบ half step "

พิจารณา สมการที่ 5 จะพบว่า ถ้า i_a มีค่าคงที่เท่ากับ I_{max} และ i_b มีค่าระหว่าง 0 ถึง I_{max} จะพบว่า จุดสมดุลย์ย่อมจะมีการเปลี่ยนแปลงระหว่าง θ กับ $\theta-\frac{\pi}{2}$ ซึ่งสามารถหาจุดสมดุลย์ที่เปลี่ยนแปลงตาม i_b ได้ดังนี้

จากสมการที่ 5 เมื่อพิจารณาที่ torque มีค่าเท่ากับศูนย์จะได้ว่า

$$i_a \sin\theta = -i_b \sin(\theta - \pi/2)$$

$$= -i_b \cos\theta$$

$$i_b / i_a = -\tan\theta$$

$$\theta = -\tan^{-1}(i_b / i_a) \qquad \text{------(6)}$$

นำสมการที่ 6 มาเขียนกราฟโดยปรับปรุงค่า θ ที่ได้ให้อยู่ในช่วง $\pi/2$ ถึง $3\pi/2$ จะได้ดังรูปที่ 2.4

รูปที่ 2.4 แสดงความสัมพันธ์ระหว่าง ib / imax กับ 0

จากกราฟจะพบว่าจุดสมดุลย์มีการเปลี่ยนแปลงตำแหน่งมุมตาม i_b / i_{max} แสดงว่าในทางปฏิบัติ หากสามารถควบคุมค่า i_b / i_{max} ได้ ย่อมมีทางที่จะควบคุมตำแหน่งจุดสมดุลย์ได้เช่นกัน

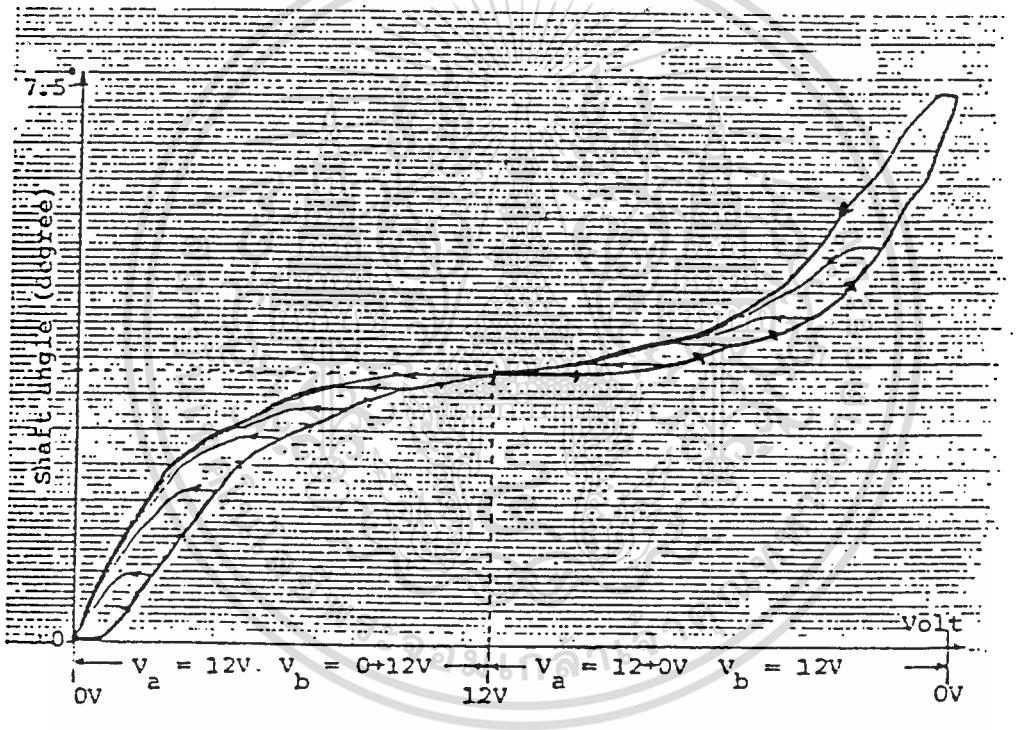
จากหลักการดังกล่าวจึงได้ออกแบบวงจรจ่ายกระแสที่สามารถปรับค่าได้ โดยอาศัยการจ่ายแรงดันไฟฟ้าแบบเป็นพัลส์ที่สามารถปรับค่า t_{on} / t_{on+off} ได้เป็นช่วง ๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คือ 0, 1/16, 1/8, 1/4, 1/2 และ 1/1 ตามลำดับ ซึ่งค่ากระแส output ที่ผ่านสเต็ปปิ้ง มอเตอร์ แม้จะมีค่าเฉลี่ยที่ผิดพลาดบ้างแต่ก็สามารถใช้ศึกษาหลักการเบื้องต้นได้เป็นอย่างดี

3. การทดลอง

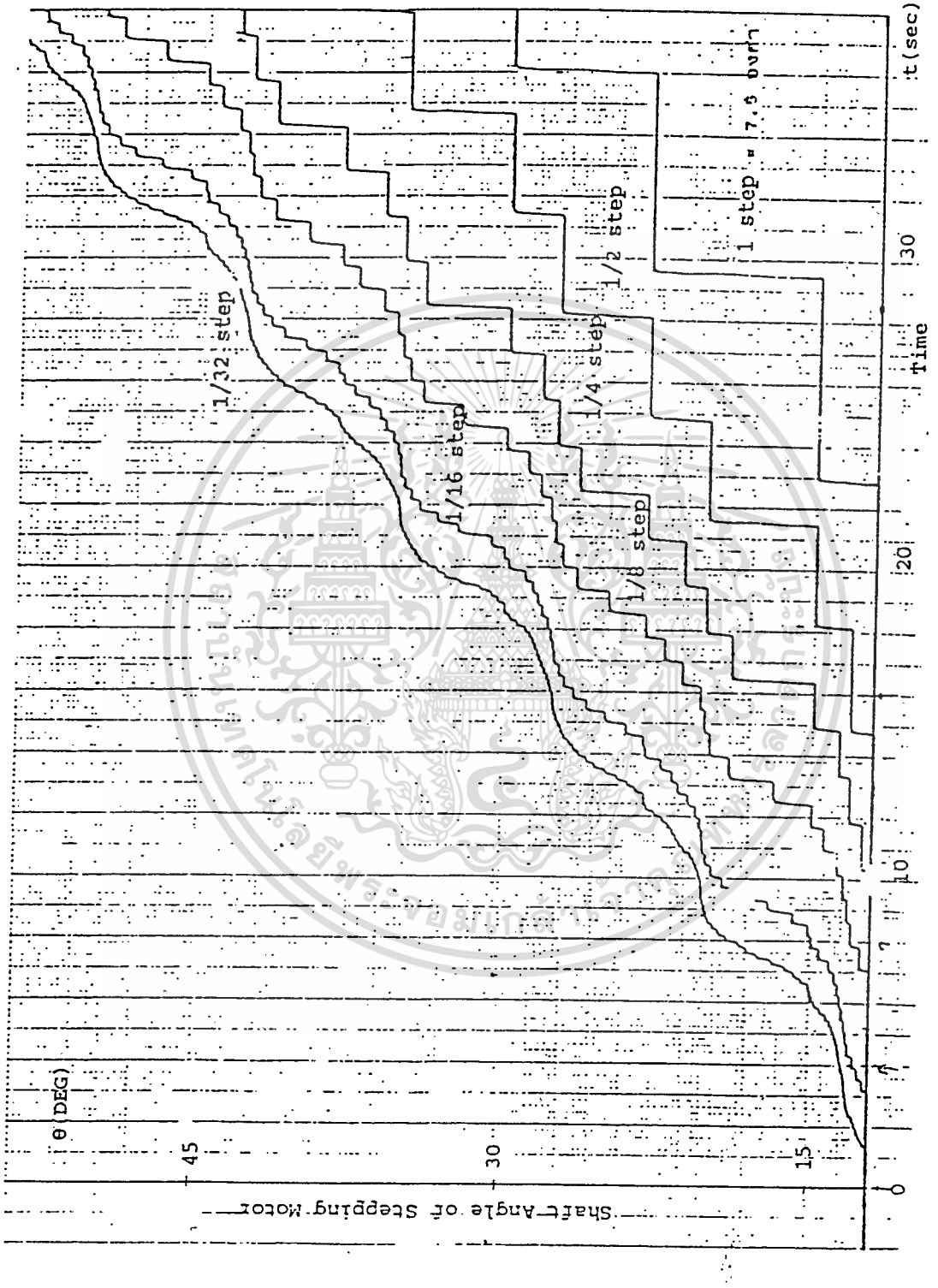
เมื่อทำการทดลองใช้ stepping motor มีค่า 7.5 องศา/step มีเฟส 4 เฟส และใช้ไฟเลี้ยงต่อเฟสเท่ากับ 12 โวลต์ โดยการต่อเฟส A เข้ากับ 12 V(d.c.) และต่อเฟส B เข้ากับแหล่งจ่ายไฟปรับค่ากระแสได้ตั้งแต่ 0 - 12 โวลต์ พบว่าได้ผลดังรูปที่ 3.1



รูปที่ 3.1 แสดงความสัมพันธ์ระหว่าง θ กับ i_b/i_{max} โดย $i_a=i_{max}=12V(d.c.)$

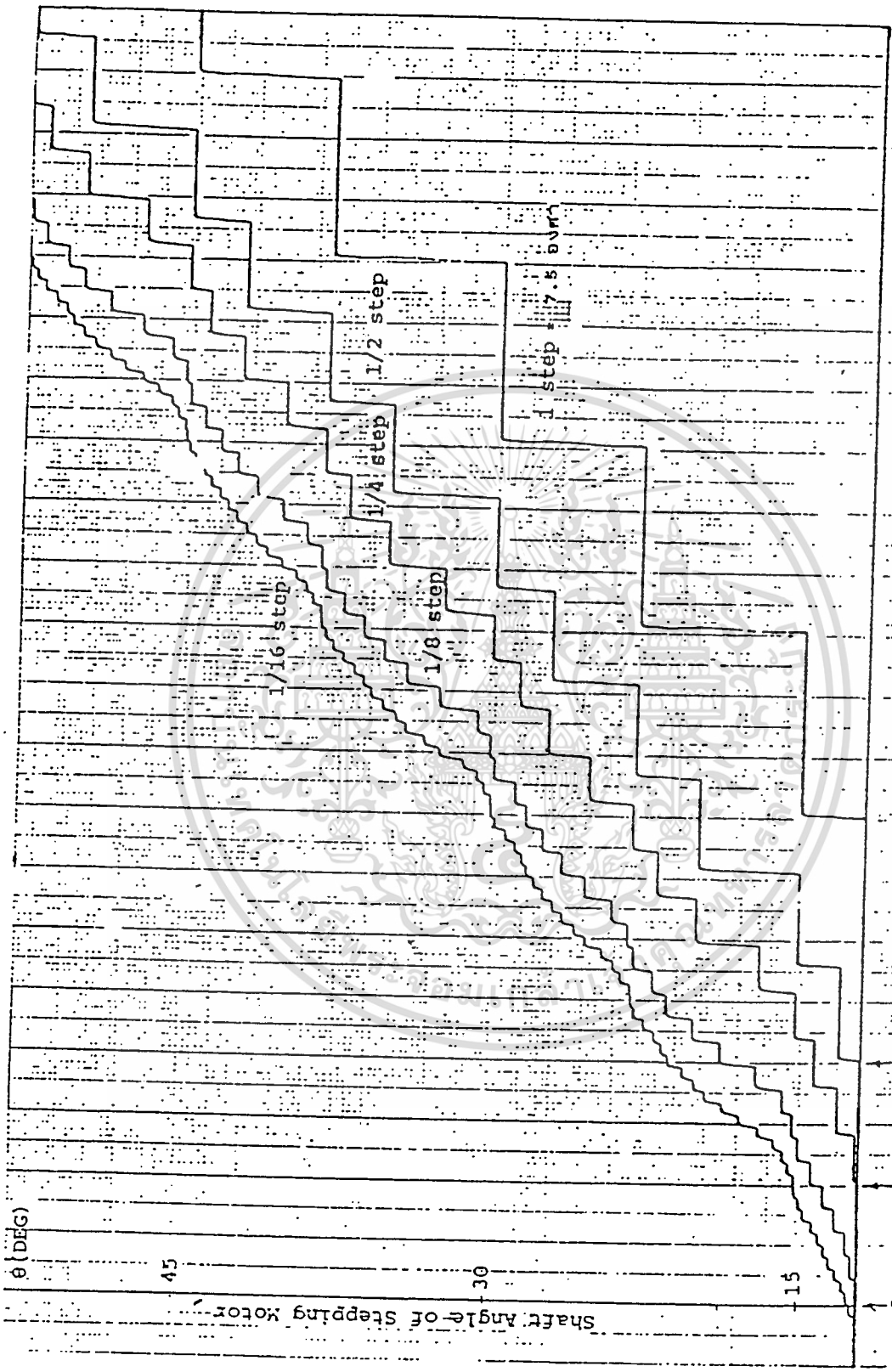
ต่อไปเป็นการขับ stepping motor ให้มีการหมุนอย่างต่อเนื่องเพื่อเปรียบเทียบข้อดีและข้อเสียระหว่าง one excitation , one-two excitation และการขับโดยอาศัยหลักการดังกล่าวซึ่งจะขอเรียกว่าการขับแบบ 1/N step ซึ่งได้ทำการบันทึกกราฟไว้แล้วดังรูปที่ 3.2 และ 3.3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.2 แสดงการเปรียบเทียบมุมการเคลื่อนที่แต่ละสเตประหว่างการขับแบบ 1 excitation, 1-2 excitation และแบบ 1/n step (เมื่อแกนของมอเตอร์หมุน 1 รอบ)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.3 แสดงการเปรียบเทียบมุมการเลื่อนที่แต่ละสเตประหว่างการขับแบบ 1 excitation, 1-2 excitation และแบบ 1/n step (เมื่อกระแสคงตัวและระหว่างคู่เฟสให้สัมพันธ์กัน)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

step No \ เฟส	A	B	C	D	
0-(N-1)	1	↑	0	0	A
N-(2N-1)	↓	1	0	0	B
2N-(3N-1)	0	1	↑	0	
3N-(4N-1)	0	↓	1	0	
4N-(5N-1)	0	0	1	↑	
5N-(6N-1)	0	0	↓	1	
6N-(7N-1)	↑	0	0	1	
7N-(8N-1)	1	0	0	↓	

step No \ เฟส	A	B
0	1	$\frac{0}{N}$
1	1	$\frac{1}{N}$
2	1	$\frac{2}{N}$
•	•	•
•	•	•
•	•	•
N-3	1	$\frac{N-3}{N}$
N-2	1	$\frac{N-2}{N}$
N-1	1	$\frac{N-1}{N}$
	A	

step No \ เฟส	A	B
N	1	1
N+1	$\frac{N-1}{N}$	1
N+2	$\frac{N-2}{N}$	1
•	•	•
•	•	•
•	•	•
2N-3	$\frac{3}{N}$	1
2N-2	$\frac{2}{N}$	1
2N-1	$\frac{1}{N}$	1
	B	

ตารางแสดงการขับแบบ $\frac{1}{N}$ step โดยเพิ่มและลดกระแสทีละเฟสตามรูปที่ 3.2

step No \ เฟส	A	B	C	D	
0-(N-1)	↓	↑	0	0	C
N-(2N-1)	0	↓	↑	0	
2N-(3N-1)	0	0	↓	↑	
3N-(4N-1)	↑	0	0	↓	

step No \ เฟส	A	B
0	1	0
1	$\frac{N-1}{N}$	$\frac{1}{N}$
2	$\frac{N-2}{N}$	$\frac{2}{N}$
•	•	•
•	•	•
•	•	•
N-3	$\frac{3}{N}$	$\frac{N-6}{N}$
N-2	$\frac{2}{N}$	$\frac{N-2}{N}$
N-1	$\frac{1}{N}$	$\frac{N-1}{N}$
	C	

ตารางแสดงการขับแบบ $\frac{1}{N}$ step โดยเพิ่มและลดกระแสระหว่างคู่เฟสให้สัมพันธ์กัน ตามรูปที่ 3.3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในรูปที่ 3.3 นั้นเป็นการขับมอเตอร์ในลักษณะที่มีการควบคุมกระแส และ ให้มีความสัมพันธ์กันเพื่อให้เกิดการชดเชยกันขึ้นเพื่อปรับลักษณะการเคลื่อนที่ให้มีความเป็นเส้นตรงมากขึ้น ซึ่งสามารถเปรียบเทียบถึงความแตกต่างได้กับรูปที่ 3.2

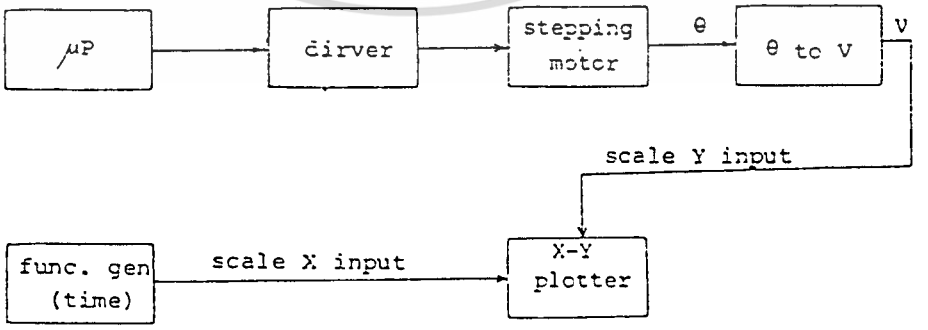
4. สรุป

ผลที่ได้จากการทดลองนี้จะพบว่ามีความแตกต่างไปจากผลที่คิดว่าจะได้ในทางอุดมคติ ทั้งนี้เนื่องมาจากโครงสร้างของสเต็ปมอเตอร์ ทั่วไปจะมีแรงลัพธ์ที่เกิดจากแรงเสียดทาน , แรงเฉื่อย และแรงกระทำจากภายนอก เป็นองค์ประกอบ ซึ่งจำเป็นจะต้องนำมาคิดในการศึกษาเพื่อพัฒนาให้สเต็ปมอเตอร์มีการทำงานที่ดีขึ้น

แต่อย่างไรก็ตามผลที่ได้นี้ ชี้ให้เห็นว่าสามารถเพิ่มความละเอียดในแต่ละสเต็ปของมอเตอร์ได้ ถึงแม้ผลที่ได้ยังไม่ดีนัก แต่ก็อยู่ในขั้นที่น่าพอใจและมีทางปรับปรุงให้ดีขึ้นได้

5. ภาคผนวก

ในการทดลองนี้ได้ใช้ stepmotor ขนาด 12 V/phase มีขนาดเส้นผ่าศูนย์กลาง 7 เซนติเมตร หนา 4.5 เซนติเมตร มีสภาพที่เก่าพอสมควร ตัวถังเป็นพลาสติก จากการทดลองพบว่า ค่าองศาต่อสเต็ปมีค่าไม่คงที่ แต่ทุก 2 สเต็ปจะมีค่า 15 องศา และแกนหมุนมีความฝืดไม่มาก ซึ่งความผิดนี้สามารถบอกได้ว่า torque ที่ได้มีค่าคงที่หรือไม่ โดยสังเกตได้จากกราฟในรูปที่ 3.1 , 3.2 และ 3.3 ตามลำดับ



รูปที่ 5.1 แสดงไดอะแกรมของระบบที่ใช้ในการทดลอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6. เอกสารอ้างอิง

(1.) TAKASHI KENJO, "STEPPING MOTOR AND THEIR MICROPROCESSOR CONTROLS", OXFORD 1984

(2.) B.C. KUO ,"STEPPING MOTORS THEORY AND APPLICATION", SRL PUBLISHING Co., 1985

ภาคผนวก ง

บทความที่ได้รับการตีพิมพ์ในระหว่างการศึกษาปริญญาโท

ในประชุมวิชาการทาง วิศวกรรมไฟฟ้า 9 สถาบัน ครั้งที่ 13 พ.ศ. 2533
ณ ภาควิชาวิศวกรรมไฟฟ้า คณะวิศวกรรมศาสตร์ มหาวิทยาลัยเชียงใหม่

เรื่อง

วิธีการวัดแรงบิดที่แปรตามมุมของสเต็ปมอเตอร์ เพื่อประยุกต์ใช้เลื่อนตำแหน่งจุดสมดุลย์ของแรงบิด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

วิธีการวัดแรงบิดที่แปรตามมุมของสเต็ปมอเตอร์เพื่อประยุกต์ใช้เลื่อนตำแหน่งจุดสมดุลย์ของแรงบิด

Measuring the Variation of Torque Depending on Shaft Angles of Stepping Motor Application for Shift the Equilibrium Position of Torque

* ลุมิตร พนาอุดมทรัพย์
** โยธิน เปรมปรานีรัชต์

บทคัดย่อ

บทความนี้เสนอวิธีการวัดความสัมพันธ์ระหว่างแรงบิดที่เปลี่ยนไปตามมุมของแกนสเต็ปมอเตอร์กับกระแสที่จ่ายให้ในแต่ละเฟส และเอานำข้อมูลของกระแสซึ่งมีผลต่อแรงบิดที่ได้นี้มาทำใช้ในการควบคุมกระแสในแต่ละเฟสอย่างสัมพันธ์กันเพื่อควบคุมความสัมพันธ์ระหว่างแรงบิดกับมุมของแกนมอเตอร์ให้มีลักษณะตามที่ต้องการ ซึ่งเป็นผลให้สามารถกำหนดจุดสมดุลย์ทางแรงบิดให้อยู่ที่มุมใด ๆ ระหว่างมุมสเต็ปปกติของแต่ละเฟสได้ตามที่ต้องการ ทำให้สามารถอ้างสเต็ปที่เป็นเศษส่วนได้ เป็นการเพิ่มจำนวนสเต็ป(จุดสมดุลย์ทางแรงบิด)ต่อรอบได้มากเท่าที่ต้องการ แต่ในการทดลองนี้จะแสดงการขับสเต็ปมอเตอร์ครั้งละ 1/8 ของสเต็ปเดิมเพื่อเป็นตัวอย่างให้เห็นถึงการประยุกต์ใช้งาน

ABSTRACT

This research paper presents a measurement method of the relation between torque which varies with shaft angle of the stepping motor and the supplied current in each phase. The obtained measurement data is applied to control the current in each phase in a correction ratio for controlling the relation of torque and the shaft angle of stepping motor as prediction. Conquently, the equilibrium positions of torque can be determined at any position angle within the normal step angle, Therefore we can control the step angles in form of microstepping. thus the numbers of steps per revalution can be increased as desirecd. As an example; the practical application of the measured results is used to drive the stepping motor of the 1/8 normal step angle.

---

* นิสิตปริญญาโท ภาควิชาวิศวกรรมระบบควบคุม คณะวิศวกรรมศาสตร์
** รองศาสตราจารย์ สถาบันเทคโนโลยีพระจอมเกล้า เจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

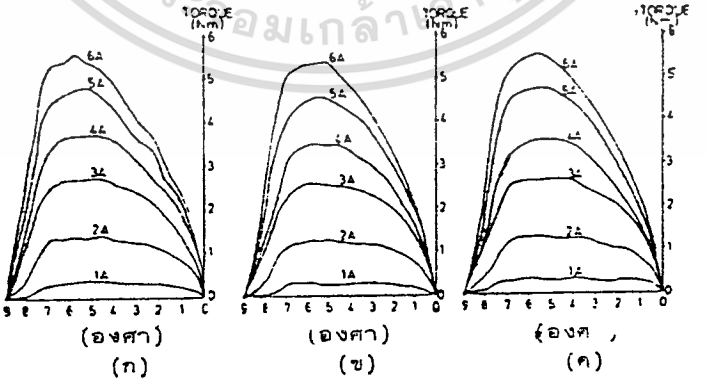
1. บทนำ

สเตปมอเตอร์จะหมุนเป็นมุมสเตปปกติ เมื่อได้รับการกระตุ้นให้ขดลวดหนึ่งของมอเตอร์ให้มีกระแสไหล โรเตอร์จะเคลื่อนที่ไปยังตำแหน่งที่เกิดค่าริลัคแต้นท์ต่ำที่สุด เราเรียกว่าตำแหน่งนี้ว่าตำแหน่งสมดุลย์ของแรงบิด ขนาดของกระแสในแต่ละขดลวดของมอเตอร์ตามปกติจะมีค่าคงที่อยู่ที่ค่านิกัดและค่ามุมสเต็ปของมอเตอร์ก็จะมีค่าคงที่ปกติตามการออกแบบทางเชิงกลของ profile ของชิ้นสเตเตอร์-โรเตอร์

เพื่อที่ทำให้การหมุนของสเตปมอเตอร์มีขนาดสเตปเล็กลงกว่าปกติ[1][2][5] เราจะต้องทำให้เกิดจุดสมดุลย์ของแรงบิดอยู่ระหว่างจุดสมดุลย์เดิมเหล่านั้น เราจะต้องควบคุมกระแสในขดลวดสองขดของมอเตอร์ให้ไหลพร้อมๆ กันด้วยขนาดที่ต่างกัน ดังนั้นจุดประสงค์หลักของการออกแบบตัวควบคุมสเตปมอเตอร์ให้สามารถหมุนเป็นสเต็ปเล็กๆ ลงกว่าปกติ(microstepping) ก็คือการออกแบบตัวควบคุมกระแสหรือแรงบิดในขดลวดของแต่ละขด นั่นหมายถึงเราจะต้องรู้ถึงความสัมพันธ์ระหว่างแรงบิดที่เปลี่ยนไปตามมุมของแกนสเต็ปมอเตอร์กับกระแสที่จ่ายให้ในแต่ละเฟสนั่นเอง.

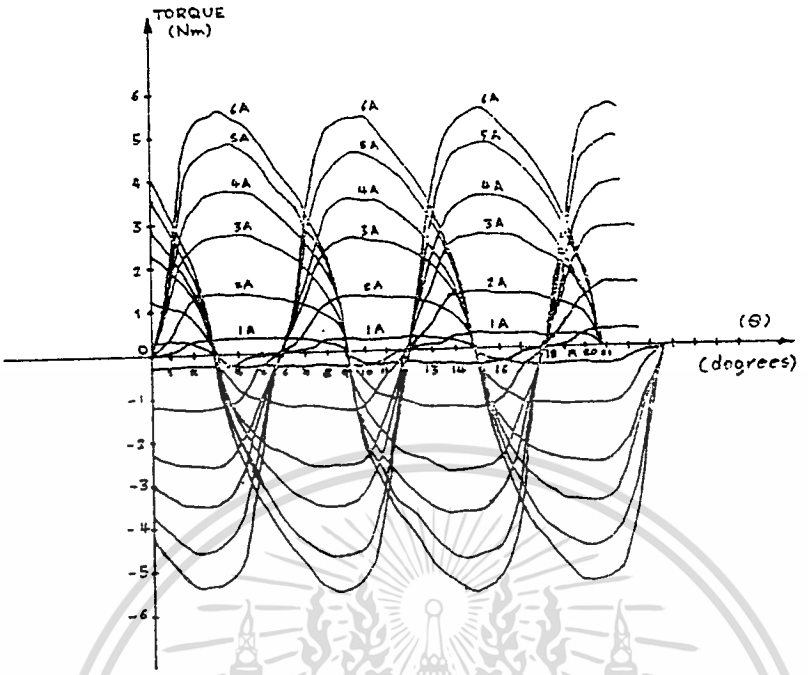
ผลงานวิจัยนี้เสนอวิธีการวัดความสัมพันธ์ระหว่างแรงบิดที่เปลี่ยนไปตามมุมของแกนสเต็ปมอเตอร์กับกระแสที่จ่ายให้ในแต่ละเฟส และนำเอาข้อมูลของกระแสซึ่งมีผลต่อแรงบิดที่ได้นี้มาใช้ควบคุมกระแสในแต่ละเฟสอย่างสัมพันธ์กันเพื่อควบคุมความสัมพันธ์ระหว่างแรงบิดกับมุมของแกนสเต็ปมอเตอร์ให้มีลักษณะตามที่ต้องการ ซึ่งเป็นผลให้สามารถกำหนดจุดสมดุลย์ทางแรงบิดให้อยู่ที่ตำแหน่งใดๆ ระหว่างมุมสเต็ปปกติได้ และทำให้สามารถกำหนดสเต็ปที่เป็นเศษส่วนได้เป็นการเพิ่มจำนวนสเต็ป(จุดสมดุลย์ของแรงบิด)ต่อรอบได้มากเท่าที่ต้องการ. ผลการทดลองนี้จะแสดงการขับสเต็ปมอเตอร์สเต็ปละ 1/8 ของสเต็ปปกติเพื่อเป็นตัวอย่างให้เห็นถึงประโยชน์ของผลงานวิจัยนี้.

โดยปกติ เมื่อมีกระแสไฟฟ้าไหลผ่านขดลวดของสเต็ปมอเตอร์เฟสใดเฟสหนึ่งจะเกิดความสัมพันธ์ระหว่างแรงบิดกับมุมของแกนสเต็ปมอเตอร์ขึ้นมาซึ่งมีลักษณะเป็นคลื่นคล้ายรูปซายน์ ดังรูปกราฟตัวอย่าง(รูป 1.)ซึ่งได้จากคุณสมบัติของสเต็ปมอเตอร์ทั่วๆ ไป[3][4]ส่วนรูปที่ 2. เป็นการนำเอาผลของแรงบิดในแต่เฟสมาแสดงให้อยู่ร่วมแกนกัน



รูปที่ 1. แสดงความสัมพันธ์ระหว่างแรงบิดกับตำแหน่งมุมของสเต็ปมอเตอร์
(ก) เฟส A (ข) เฟส B (ค) เฟส C

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า



รูปที่ 2. แสดงผลของแรงบิดในแต่เฟสบนแกนเดียวกัน

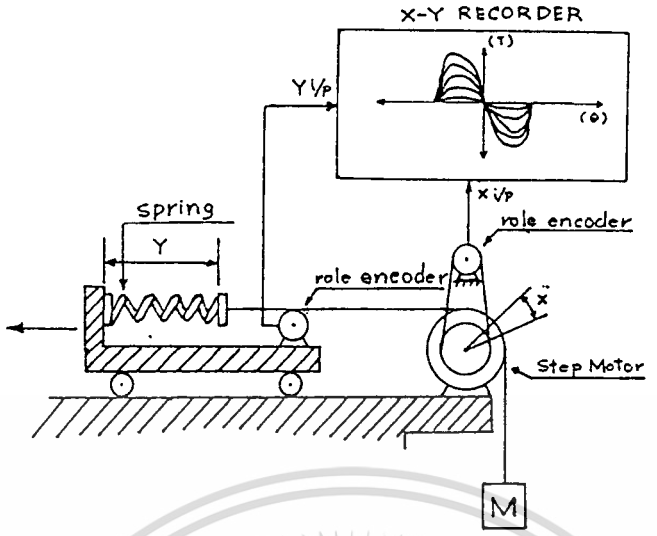
จากรูปที่ 2. จะอธิบายได้ว่าการจ่ายกระแสในเฟสใดเฟสหนึ่งจะเกิดผลแรงบิดที่มีลักษณะเฉพาะ และมีลักษณะคล้ายคลึงกัน แต่จุดสมดุลย์ยังคงอยู่ที่เดิมไม่ขึ้นกับกระแสที่จ่ายให้ แต่ถ้ามีการจ่ายกระแส 2 เฟสพร้อมกัน แรงบิดที่เกิดก็จะเกิดการเสริมและหักล้างกันตามแอมปลิจูด ซึ่งได้จะได้ผลของแรงบิดที่เลื่อนไปทำให้จุดสมดุลย์เลื่อนไปด้วย ดังนั้นถ้าเรารู้ความสัมพันธ์ระหว่างแรงบิดกับกระแสของสเต็ปมอเตอร์ เราก็สามารถเพิ่มประสิทธิภาพให้สเต็ปมอเตอร์ตัวนั้นได้ตามความต้องการ เช่น ควบคุมจุดสมดุลย์ให้อยู่ที่ตำแหน่งต่างๆ กลายเป็นสเต็ปย่อยที่แฝงตัวอยู่ระหว่างมุมของสเต็ปปกติ

2. วิธีการวัดแรงบิด

เนื่องจากเรารู้ว่าแรงบิดของมอเตอร์มีการเปลี่ยนแปลงตามมุมของแกนมอเตอร์ ดังนั้นอุปกรณ์ที่เราจะใช้ต้องสามารถวัดแรงบิดควบคู่ไปกับมุมของสเต็ปมอเตอร์ได้พร้อม ๆ กัน แต่อุปกรณ์ดังกล่าวเป็นอุปกรณ์เฉพาะงานซึ่งไม่สามารถซื้อได้โดยง่ายและมักมีราคาแพง เพื่อความประหยัดเราจึงได้ออกแบบระบบอุปกรณ์ดังกล่าวซึ่งมีราคาถูกแต่คุณภาพพอใช้ได้ขึ้นมาแทน ซึ่งมีระบบดังรูป 3.

จากรูปที่ 3. จะพบว่าประกอบด้วยอุปกรณ์ที่ทำการวัดมุมและแรงบิดซึ่งใช้ role encoder 2 ตัวโดยตัวแรกจะมีหน้าวัดมุมของสเต็ปมอเตอร์ ส่วนอีกตัวจะใช้ร่วมกับสปริงซึ่งจะได้หน่วยของแรงดึงออกมาเป็นนิวตัน ซึ่งสามารถนำมาแปรค่าให้อยู่รูปของแรงบิดได้โดยการคูณค่ารัศมีของวงล้อที่สปริงดึงอยู่ จะได้หน่วยออกมาเป็นนิวตัน.เมตร ส่วนภาคจ่ายกระแสใช้แหล่งจ่ายกระแสที่ปรับค่าได้ 2 ตัว และสเต็ปมอเตอร์ตัวอย่างที่ใช้เป็นชนิด 2 เฟสทนกระแสได้ 1.0 Amp.

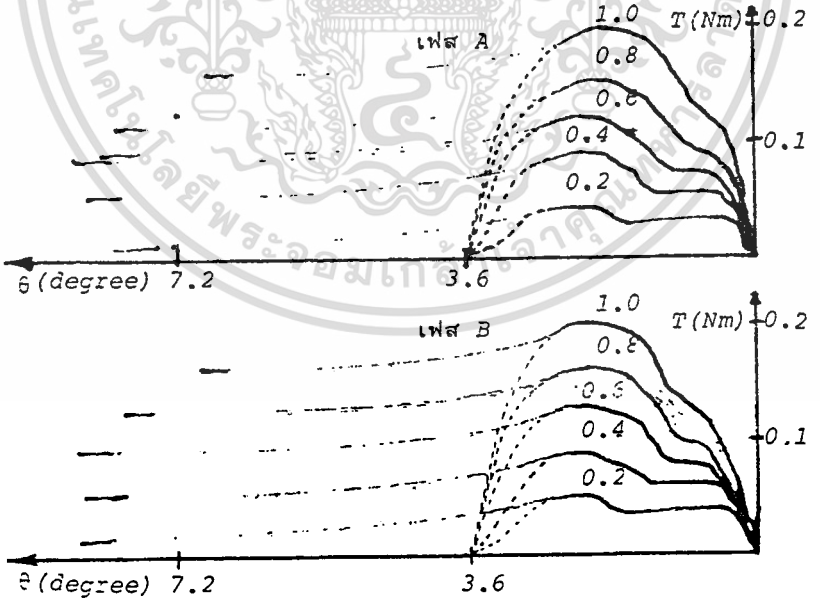
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า



รูปที่ 3. แสดงระบบการวัดแรงบิดที่แปรตามมุมของสเต็ปมอเตอร์

3. ผลการวัด

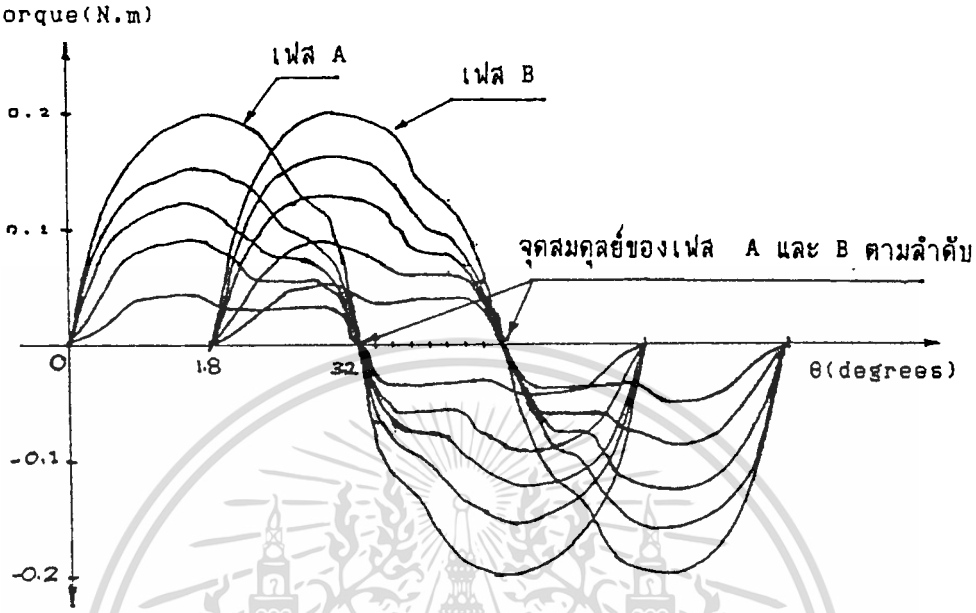
เมื่อเราได้ระบบทดลองที่สมบูรณ์แบบแล้ว เราจะทำการเก็บข้อมูลทีละเฟส โดยในแต่ละเฟสจะมีการเปลี่ยนค่ากระแสไปเรื่อย ๆ ซึ่งผลของการวัดจะได้ดังรูปที่ 4.



รูปที่ 4. แสดงผลของการวัดที่กระแสค่าต่าง ๆ ในแต่ละเฟส

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

จากผลที่วัดได้ดังกล่าวจะพบว่าแรงบิดช่วงขาลงไม่สามารถวัดได้ชัดเจน แต่สามารถตัดทิ้งได้เพราะยังไม่มีผลในตอนนี้ซึ่งจะได้อธิบายต่อไป ในตอนนี้เราจะนำเอาผลที่ได้นี้มาทำการรวบรวมให้อยู่แกนร่วมกันได้ดังรูปที่ 5.



รูปที่ 5. แสดงผลของแรงบิดในแต่ละเฟสบนแกนเดียวกัน

จากรูปจะเห็นว่าเราได้ข้อมูลของแรงบิดมาแล้ว ซึ่งต่อไปเราจะนำผลของแรงบิดในแต่เฟสนี้ไปประยุกต์ใช้เพื่อลองเปลี่ยนตำแหน่งมุมสมดุลย์ให้อยู่ในตำแหน่งที่การจ่ายกระแสแบบปกติไม่สามารถทำได้

4. การประยุกต์ใช้

เพื่อให้เห็นแนวทางการพัฒนาต่อไป เราจะแสดงตัวอย่างการเพิ่มจำนวนสเต็ปของสเต็ปมอเตอร์ให้เพิ่มขึ้นอีก 8 เช่า (โดยจะยังไม่มีการควบคุมค่าแรงบิดสูงสุดแต่อย่างไร จึงอาจเกิด ripple torque ได้ ซึ่งอาจมีผลต่อการใช้งานจริงบ้าง แต่ก็ไม่ค่อยมีความสำคัญมากเท่าไร) ในขั้นแรกเรารู้ว่ามอเตอร์ตัวนี้ต้องการกระแสปกติ 1.0 Amp ในแต่ละเฟสและมีเนียง 2 เฟส สมมติเป็นเฟส A และ B ตามลำดับ เราจะเริ่มต้นจ่ายกระแสในกระแสในแต่ละเฟสดังตารางที่ 1.

ค่า X ที่แสดงจะได้จากการประมาณค่าจากกราฟความสัมพันธ์ของแรงบิดกับมุมแกนสเต็ปมอเตอร์ ซึ่งจะมีวิธีหาได้ง่ายๆ คือ ดูตัวอย่างสเต็ปที่ 1.125 สมมุติสเต็ปที่ 1 อยู่มุม 0 องศา แต่มอเตอร์ที่ใช้มีขนาดองศาต่อสเต็ปเท่ากับ 1.8 องศา ดังนั้นที่สเต็ปที่ 1.125 มุมสมดุลย์จะต้องอยู่ที่ 0.125*1.8 = 0.225 องศา และเราให้กระแสเฟส A มีค่า 1-0.125=0.875 Amp จากค่ากระแสดังกล่าว เราจะรู้ว่าที่มุม 0.225 องศา มีขนาดแรงบิดเท่าใดโดยดูจากกราฟที่ 3.1-3.2 จากกราฟเดียวกันนี้เราจะต้องหาค่ากระแสทางเฟส B ที่ทำให้เกิดแรงบิดมาหักล้างกับแรงบิดจากเฟส A จนแรงบิดลัพธ์ที่ได้มีค่าเป็นศูนย์ที่มุม 0.225 องศา สำหรับการหากระแส X ที่สเต็ปอื่น ๆ ก็ใช้หลักการนี้ไปจนครบ ซึ่งจะได้ตารางที่สมบูรณ์ในตารางที่ 2.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

สเต็ปปกติที่	มุมที่(องศา)	สเต็ปย่อยที่	กระแสในเฟส A (Amp)	กระแสในเฟส B (Amp)
1.0	0.0	0	1.0*1.0 = 1.0	0.0
1.125	0.225	1	1.0*0.875 = 0.875	X
1.25	0.45	2	1.0*0.75 = 0.75	X
1.375	0.675	3	1.0*0.625 = 0.625	X
1.5	0.9	4	1.0*0.5 = 0.50	1.0*0.5 = 0.50
1.625	1.125	5	X	1.0*0.625 = 0.625
1.75	1.35	6	X	1.0*0.75 = 0.75
1.875	1.575	7	X	1.0*0.875 = 0.875
2.0	1.8	8	0.0	1.0*1.0 = 1.00

ตารางที่ 1. แสดงค่ากระแสเริ่มต้นในแต่ละเฟสสำหรับสเต็ปต่าง ๆ

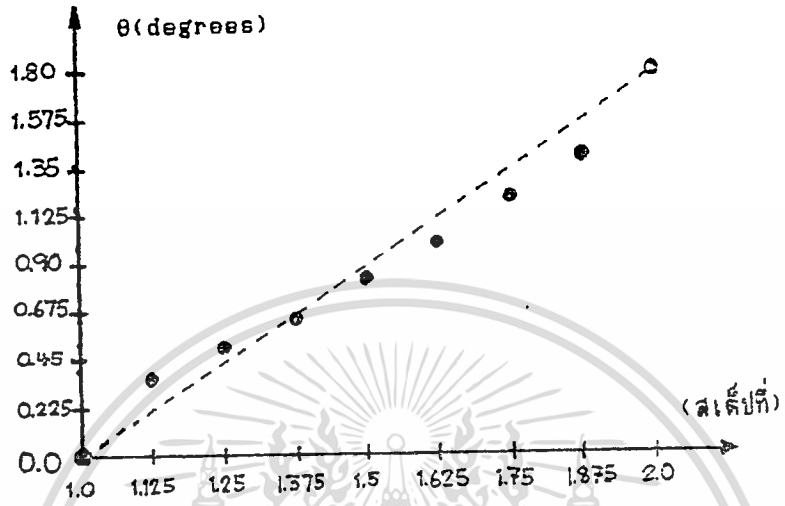
สเต็ปปกติที่	มุมที่(องศา)	สเต็ปย่อยที่	กระแสในเฟส A (Amp)	กระแสในเฟส B (Amp)
1.0	0.0	0	1.0*1.0 = 1.0	0.0
1.125	0.225	1	1.0*0.875 = 0.875	~ 0.32
1.25	0.45	2	1.0*0.75 = 0.75	~ 0.41
1.375	0.675	3	1.0*0.625 = 0.625	~ 0.48
1.5	0.9	4	1.0*0.5 = 0.50	1.0*0.5 = 0.50
1.625	1.125	5	~ 0.45	1.0*0.625 = 0.625
1.75	1.35	6	~ 0.40	1.0*0.75 = 0.75
1.875	1.575	7	~ 0.30	1.0*0.875 = 0.875
2.0	1.8	8	0.0	1.0*1.0 = 1.00

ตารางที่ 2. แสดงค่ากระแสในแต่ละเฟสสำหรับที่สเต็ปย่อยต่างๆ โดยสมบูรณ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

5. ผลที่ได้

หลังจากที่ได้ค่ากระแสสำหรับในแต่ละเฟสที่สเต็ปย่อยต่างๆ ครบแล้ว เราจะนำค่ากระแสดังกล่าวมาป้อนให้แก่เฟส A และ B ตามลำดับเพื่อดูผลว่าสเต็ปย่อยที่ได้จะอยู่ในตำแหน่งที่ถูกต้องหรือไม่ โดยจะแสดงผลที่ได้ในรูปของกราฟดังรูปที่ 6.



รูปที่ 6. แสดงมุมสมดุลย์ที่ได้จากการป้อนให้แก่เฟส A และ B ตามค่าในตารางที่ 2

ในรูปที่ 6. จะเห็นว่าเราสามารถที่จะกำหนดมุมสมดุลย์ให้ระหว่างเฟสได้ เป็นผลให้สเต็ปมอเตอร์ตัวนี้มีประสิทธิภาพในการอ้างมุมได้ขึ้นจากเดิมที่มีเพียง 48 สเต็ปต่อรอบ ขึ้นเป็น 384 สเต็ปต่อรอบ ซึ่งเมื่อนำไปใช้งานจะเห็นผลความละเอียดที่ชัดเจนกว่าปกติมาก

6. สรุป

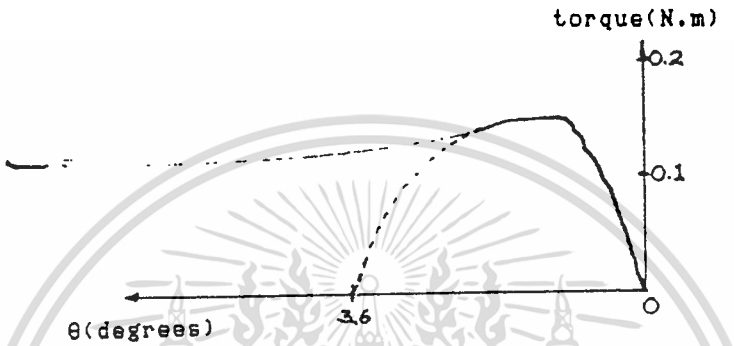
จากที่ผ่านมาเราสามารถสรุปได้ว่า การศึกษาแรงบิดที่แปรตามค่ามุมมีความสำคัญต่อการศึกษาสเต็ปมอเตอร์เพื่อการพัฒนาเป็นอย่างมาก เพราะสามารถที่จะอาศัยความสัมพันธ์ระหว่างแรงบิดกับมุมที่ได้นี้มาพัฒนาประสิทธิภาพของมอเตอร์ได้อย่างกว้างขวาง และจากการเก็บข้อมูลการทดลองต่างๆ ที่ผ่านมา เราจะพบว่าตำแหน่งสเต็ปย่อยที่เราสามารถจะอ้างอิงได้ยังไม่ละเอียดนัก ทั้งนี้อาจจะเป็นเพราะอุปกรณ์การวัดของเราคงจะดีไม่เพียงพอ แต่ก็ทำให้เราเห็นแนวทางที่จะพยายามพัฒนาต่อไป ข้อดีอีกอย่างหนึ่งเราสามารถนำเอาระบบนี้ไปใช้แทนระบบเซอร์โวมอเตอร์ในบางงานได้เช่น ในระบบ X-Y ploter และยังสามารถควบคุมได้ง่ายและราคาระบบถูกกว่าระบบเซอร์โวมอเตอร์ เพราะระบบของสเต็ปมอเตอร์เป็น ระบบเปิด อีกทั้งสามารถควบคุมความเร็วและอ้างอิงตำแหน่งได้โดยตัวเองไม่ต้องอาศัยอุปกรณ์ภายนอกช่วย ซึ่งเป็นสิ่งจำเป็นมากสำหรับระบบเซอร์โวมอเตอร์

7. ภาคผนวก

เนื่องจากบทความนี้มีจุดประสงค์เพื่อแนะนำหลักการที่สำคัญๆ เท่านั้นซึ่งความจริงยังสามารถปรับปรุงผล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

การทดลองที่ผ่านมาให้ดีขึ้นอีก เช่นควบคุมให้รูปกราฟของแรงบิดที่แปรตามมุมมีรูปร่างลักษณะเหมือนเดิมให้มากที่สุดพร้อมกับให้มีการอ้างอิงตำแหน่งได้แม่นยำ เพราะผลที่ผ่านมาพบว่าขณะที่มอเตอร์ชี้ตำแหน่งที่สเต็ป 1.5 ดูตารางที่ 2. ซึ่งกระแสที่ไหลในแต่ละเฟสมีค่า 0.50 Amp เมื่อทำการเก็บข้อมูลของแรงบิดจะได้ผลดังรูปที่ 7. ซึ่งจะเห็นว่าค่าสูงสุดของแรงบิดที่ได้มีค่าต่างจากค่าปกติ ซึ่งถ้าจะให้ดีที่สุดนอกจะอ้างอิงตำแหน่งสเต็ปย่อยได้แล้วก็ควรจะรักษาลักษณะของแรงบิดที่แปรตามมุมให้คงที่ด้วย เนื่องจากผลที่เกิดจากแรงบิดไม่คงที่กลายเป็น riple torque ซึ่งจะทำให้เกิดการสั่นได้ในขณะใช้งานที่โหลดค่ามาก



รูปที่ 7. แสดงความสัมพันธ์ระหว่างแรงบิดกับมุมของแกนสเต็ปมอเตอร์

สำหรับแรงบิดช่วงขาลงนั้นเราจะสามารถทำการวัดได้ถ้า หากเปลี่ยนอุปกรณ์จากเดิมซึ่งเป็นสปริงไปใช้อุปกรณ์ที่สามารถบอกขนาดแรงดึงได้โดยมีการยืดตัวน้อยมากเช่น load cell ซึ่งมีราคาแพงมาก ทั้งนี้เพราะการใช้อุปกรณ์แรงดึงที่มีการยืดตัวมากดังเช่นสปริงจะทำให้ในขณะแรงบิดของมอเตอร์เริ่มลดลงนั้นสเต็ปมอเตอร์จะเริ่มแพ้แรงที่สปริงดึงและเริ่มหมุนตามแรงสปริง ซึ่งถ้าหากว่าสเต็ปมอเตอร์เริ่มหมุนตามแรงสปริงไปแล้วและสปริงเริ่มหดตัวลงเพื่อลดแรงดึงของตัวเองแต่ถ้าแรงดึงนั้นยังขณะแรงบิดของสเต็ปมอเตอร์อยู่ ก็จะทำให้สเต็ปมอเตอร์ยังคงหมุนตามแรงดึงของสปริงไปจนกระทั่งหลุดออกจากมุมสมดุลย์เดิมไปยังมุมสมดุลย์ใหม่ ที่ซึ่งสามารถสู้กับแรงดึงของสปริงได้ ด้วยเหตุนี้การใช้สปริงจึงไม่สามารถวัดแรงบิดของสเต็ปมอเตอร์ได้สมบูรณ์แบบจึงต้องหาทางหันไปใช้อุปกรณ์ที่มีการยืดตัวน้อยมากๆ แทน

8. เอกสารอ้างอิง

[1] M.F.Rahman"Approaches to Design of Ministepping Step Motor Controllers" IEEE Trans.on indust. Electro., Vol. IE-32, No.3,August,pp.229-233,1985

[2] สุมิตร พนาอุดมพรัทย์, ดร.โยธิน เปรมปราณีรัชต์, " วิธีการเพิ่มความละเอียดเชิงมุมของสแกนสเต็ปมอเตอร์",การประชุมทางวิชาการ วิศวกรรมไฟฟ้า 9 สถาบัน ครั้งที่ 11 ที่คณะวิศวกรรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เทคโนโลยี สถาบันเทคโนโลยีราชมงคล ระหว่าง 16-17 ธันวาคม 2531,เล่มที่ 2 หน้า 4-1-1

[3] TAKASHI KENJO," STEEPING MOTOR AND THEIR MICROPROCESSOR CONTROLS ", OXFORD 1984.

[4] B.C. KUD,"STEEPING MOTORS; THEORY AND APPLICATION",SRL PUBLISHING Co., 1985.

[5] MUHAMMED F.RAHMAN," Approaches to Design of Ministepping Step Motor Controlles and Their Accuracy Considerations ", IEEE TRANSACTIONS ON INDUSTRIAL ELECTRONICS,VOL.IE-32,NO.3,AUGUST 1985,pp.229-233.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ประวัติผู้เขียน

ชื่อผู้เขียน	นายสุมิตร พนาอุดมทรัพย์
วุฒิการศึกษาระดับปริญญาตรี	วิทยาศาสตรบัณฑิต สาขาวิชาฟิสิกส์
สถานที่สำเร็จการศึกษา	สถาบันเทคโนโลยีพระจอมเกล้า วิทยาเขตธนบุรี
ปีที่สำเร็วการศึกษา	ปีการศึกษา 2531

ผลงานทางวิชาการที่ได้รับการตีพิมพ์ (1) เรื่อง" วิธีการเพิ่มความละเอียดเชิงมุมของสเต็ปมอเตอร์ "

(2) เรื่อง "วิธีการวัดแรงบิดที่แปรตามมุมของสเต็ปมอเตอร์เพื่อประยุกต์ใช้เลื่อนตำแหน่งจุดสมดุลย์ของแรงบิด "

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า