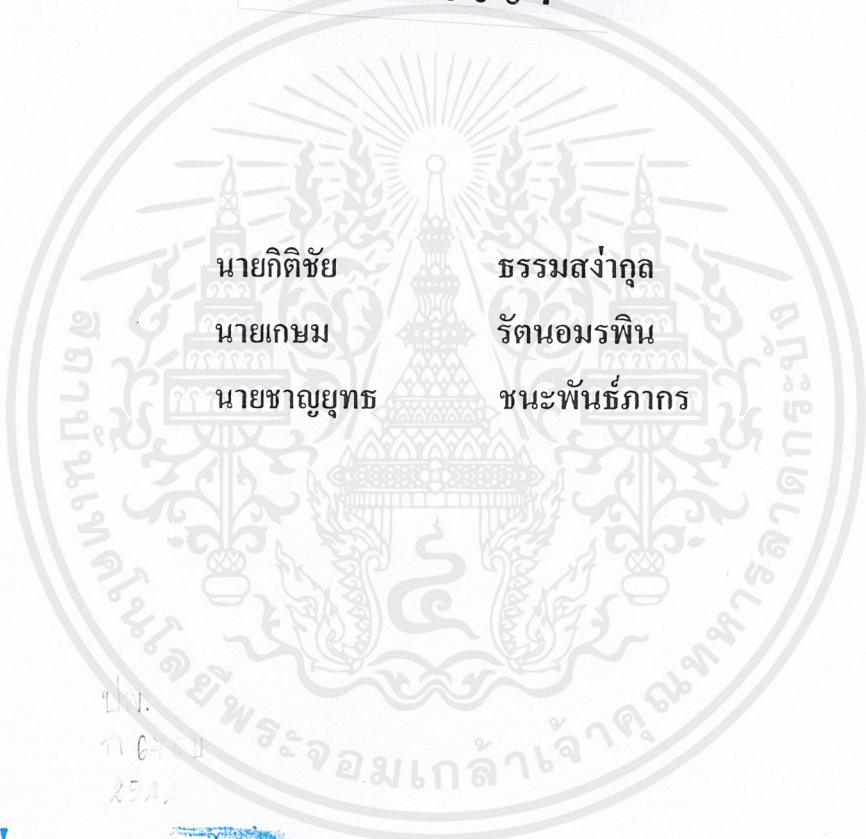


แขนกลควบคุมตรง  
DIRECT DRIVE ROBOT



นายกิตติชัย

ธรรมสง่ากุล

นายเกษม

รัตนอมรพิน

นายชาญยุทธ

ชนะพันธ์ภากร

ปี  
๒๕๔๖  
๒๕๔๖

เลขหมู่.....  
เลขทะเบียน **45868**  
วัน, เดือน, ปี **19 ก.พ. 2546**

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

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมการวัดคุม

ภาควิชาวิศวกรรมการวัดคุม คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2544

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# **DIRECT DRIVE ROBOT**



**A THESIS SUBMITTED IN PARTIAL FULFILLMENT  
OF THE REQUIREMENT FOR THE DEGREE OF  
BACHELOR OF ENGINEERING IN INSTRUMENTATION ENGINEERING  
DEPARTMENT OF INSTRUMENTATION ENGINEERING  
FACULTY OF ENGINEERING  
KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG**

**2001**

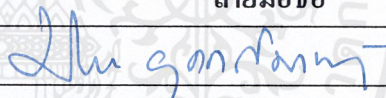
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาควิชาวิศวกรรมการวัดคุม  
คณะวิศวกรรมศาสตร์  
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง  
ใบรับรองปริญญาโท

หัวข้อปริญญาโท แขนกลขับเคลื่อน  
DIRECT DRIVE ROBOT

นักศึกษาผู้จัดทำ นายกิตติชัย ชรรณสง่ากุล รหัสประจำตัว 41014034  
นายเกษม รัตนอมรพิน รหัสประจำตัว 41014042  
นายชาญยุทธ ชนะพันธ์ภากร รหัสประจำตัว 41014105

ปริญญา วิศวกรรมศาสตรบัณฑิต  
สาขาวิชา วิศวกรรมการวัดคุม  
ปีการศึกษา 2544

อาจารย์ผู้ควบคุมปริญญาโท	ลายมือชื่อ
ผศ. ประภาส อุกคภิมาพันธุ์	

วัน/เดือน/ปี ที่สอบ วันจันทร์ที่ 25 มีนาคม พ.ศ. 2545  
สถานที่สอบ ณ.ห้องสอบปริญญาโท ภาควิชาวิศวกรรมการวัดคุม

ภาควิชารับรองแล้ว



( ผศ.ประสิทธิ์ จุลเสรีวงศ์ )

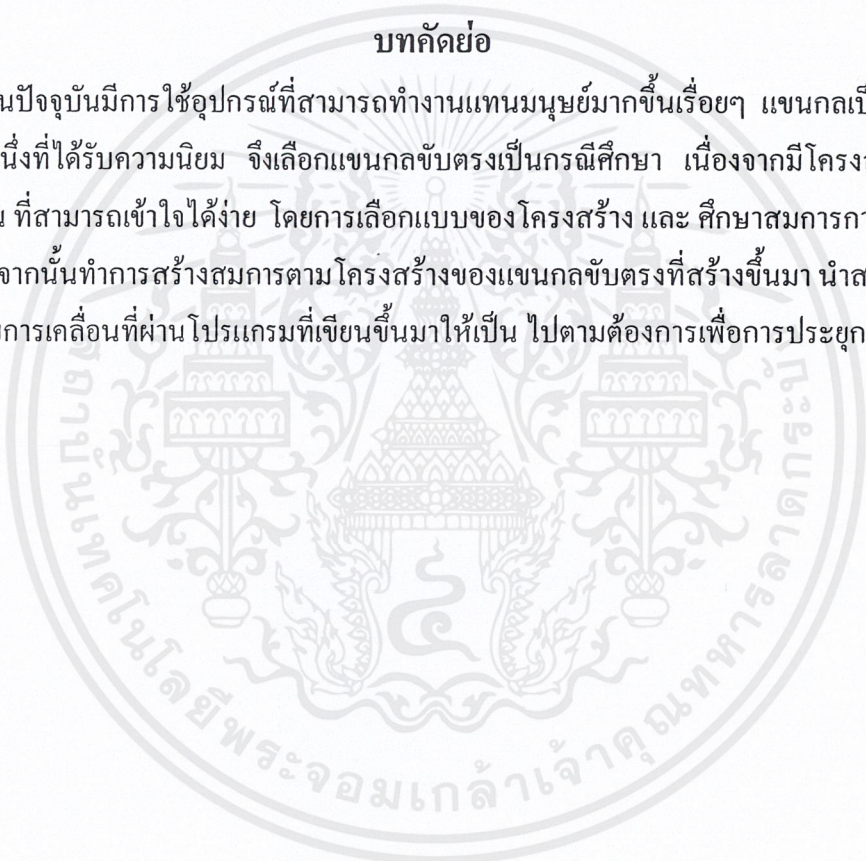
หัวหน้าภาควิชา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อปริญญานิพนธ์	แขนกลขับเคลื่อน	
	Direct Drive Robot	
นักศึกษาผู้จัดทำ	นายกิตติชัย	ธรรมสง่ากุล
	นายเกษม	รัตนอมรพิน
	นายชาญยุทธ	ชนะพันธ์ภากร
อาจารย์ที่ปรึกษา	ผศ. ประภาส	อุคคกิมพันธ์
ปีการศึกษา	2544	

### บทคัดย่อ

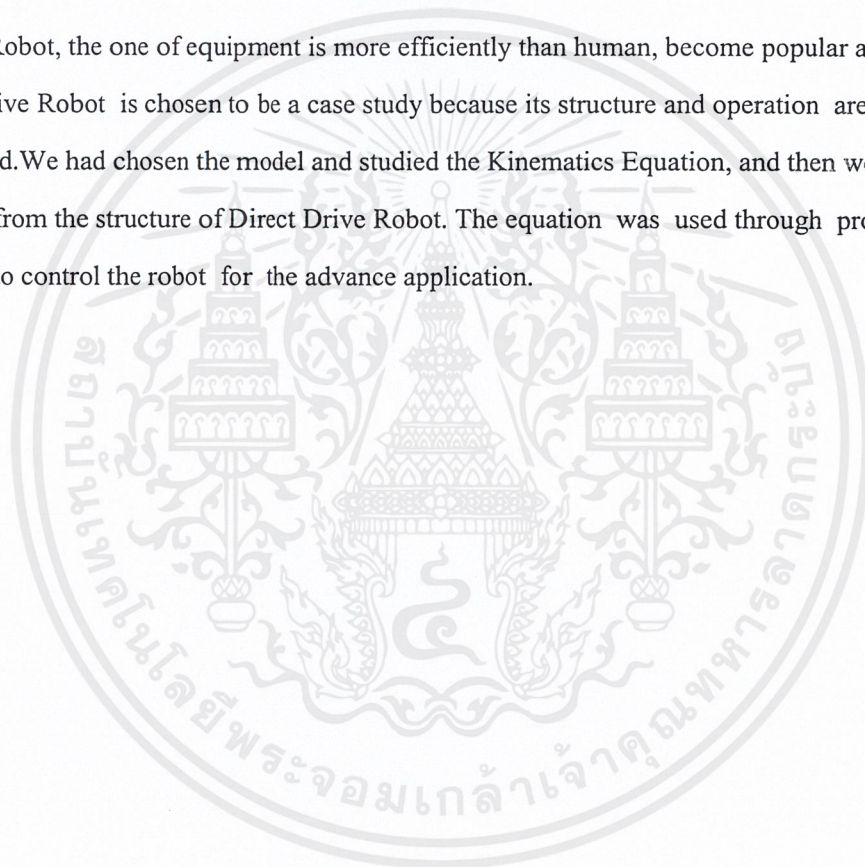
ในปัจจุบันมีการใช้อุปกรณ์ที่สามารถทำงานแทนมนุษย์มากขึ้นเรื่อยๆ แขนกลเป็นอุปกรณ์ประเภทหนึ่งที่มีความนิยม จึงเลือกแขนกลขับเคลื่อนเป็นกรณีศึกษา เนื่องจากมีโครงสร้างและการทำงาน ที่สามารถเข้าใจได้ง่าย โดยการเลือกแบบของโครงสร้าง และ ศึกษาสมการการเคลื่อนที่ของวัตถุ จากนั้นทำการสร้างสมการตามโครงสร้างของแขนกลขับเคลื่อนที่สร้างขึ้นมานำสมการที่ได้มาควบคุมการเคลื่อนที่ผ่านโปรแกรมที่เขียนขึ้นมาให้เป็นไปตามต้องการเพื่อการประยุกต์ใช้ต่อไป



<b>Thesis Title</b>	Direct Drive Robot	
<b>Authors</b>	Mr. Kitichai	Thamsangakul
	Mr. Kasem	Ratana-amornpin
	Mr. Chanyut	Chanpanpakorn
<b>Thesis Advisor</b>	Asst.Prof. Prapart	Ukakimaparn
<b>Year</b>	2001	

## ABSTRACT

Robot, the one of equipment is more efficiently than human, become popular at this time. Direct Drive Robot is chosen to be a case study because its structure and operation are easily to understand. We had chosen the model and studied the Kinematics Equation, and then we made the equation from the structure of Direct Drive Robot. The equation was used through programmed software to control the robot for the advance application.



## กิตติกรรมประกาศ

การจัดทำปฏิญานិพนธ์และโครงการครั้งนี้สำเร็จได้ด้วยดี เพราะได้รับการแนะนำข้อมูลเกี่ยวกับ การดำเนินงาน การออกแบบ และ การสร้าง, การเขียนโปรแกรม, เครื่องมืออุปกรณ์ต่างๆ ตลอดจนความช่วยเหลือทั้งในด้านทุนทรัพย์ แรงงาน และกำลังใจ จากบุคคลซึ่งมีรายนามดังต่อไปนี้

1. ผศ. ประภาส อุกคกิมพันธ์ (อาจารย์ที่ปรึกษา)
2. นาย กฤษณ์ เสมอพิทักษ์

ทางกลุ่มผู้จัดทำใคร่ขอขอบคุณเป็นอย่างสูงในความช่วยเหลือครั้งนี้ ซึ่งส่งผลให้โครงการนี้ประสบความสำเร็จลุล่วงไปได้ด้วยดี  
คุณค่าและประโยชน์อันพึงมีจากปฏิญานิพนธ์ฉบับนี้ ขอมอบแด่ผู้มีพระคุณทุกท่าน

คณะผู้จัดทำ

# สารบัญ

	หน้า
บทคัดย่อภาษาไทย.....	I
บทคัดย่อภาษาอังกฤษ.....	II
กิตติกรรมประกาศ.....	III
สารบัญ.....	IV
สารบัญตาราง.....	VIII
สารบัญภาพ.....	IX
บทที่ 1 บทนำ.....	1
1.1 ความเป็นมาและเหตุจูงใจ.....	1
1.2 วัตถุประสงค์ของปริญญานิพนธ์.....	1
1.3 ขอบเขตของปริญญานิพนธ์.....	1
1.4 ขั้นตอนการศึกษา.....	1
บทที่ 2 ทฤษฎีและหลักการ.....	2
2.1 ความรู้เบื้องต้นเกี่ยวกับโรบอท.....	2
2.1.1 Hydraulic Drives.....	2
2.1.2 Direct Drives Robot.....	2
2.2 ทฤษฎีที่เกี่ยวข้อง (Theory).....	4
2.2.1 พื้นฐานกลศาสตร์ที่เกี่ยวข้องกับการเคลื่อนที่ (Basic Kinematics).....	4
2.2.1.1 กลศาสตร์ที่เกี่ยวข้องกับการเคลื่อนที่ (Kinematics).....	4
2.2.1.2 กรอบอ้างอิง (Reference Frames).....	4
2.2.1.3 การย้าย (Translation).....	5
2.2.1.4 เมตริกซ์การหมุน (Rotation Matrix).....	6
2.2.2 การเคลื่อนที่ของวัตถุแข็งเกร็ง (Rigid Body Motion).....	7
2.2.3 กลศาสตร์ที่เกี่ยวข้องกับการเคลื่อนที่โดยตรง และ ผันกลับ (Direct & Inverse Kinematics Equation).....	8
2.2.3.1 Direct Kinematics Equation.....	9

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# สารบัญ(ต่อ)

	หน้า
2.2.3.2 Inverse Kinematics Equation.....	10
2.3 การสื่อสารแบบอนุกรม.....	10
2.3.1 การสื่อสารข้อมูลแบบอะซิงโครนัส.....	11
2.3.2 มาตรฐานพอร์ตอนุกรมแบบ RS-232.....	12
2.3.3 คอนเน็กเตอร์สำหรับพอร์ต RS-232 และการเชื่อมต่อ.....	13
2.3.4 UART.....	16
2.3.5 รีจิสเตอร์ของพอร์ตอนุกรม RS-232.....	17
2.3.6 แอดเดรสของพอร์ตอนุกรม.....	18
2.3.7 ระดับแรงดันที่ใช้งานสำหรับพอร์ตอนุกรม RS-232.....	19
2.4 การเขียนโปรแกรมเพื่อใช้งานพอร์ตอนุกรม.....	19
2.4.1 การกำหนดค่าเริ่มต้นให้กับพอร์ตอนุกรม.....	19
2.4.2 การรับและส่งข้อมูลแบบอนุกรม.....	21
2.4.3 คอนโทรล MSCOMM .....	21
2.4.3.1 CommPort.....	22
2.4.3.2 Setting.....	22
2.4.3.3 PortOpen.....	23
2.4.3.4 Input.....	24
2.4.3.5 InBufferCount.....	25
2.4.3.6 InBufferSize.....	25
2.4.3.7 InputLen.....	25
2.4.3.8 InputMode.....	26
2.4.3.9 Output.....	27
2.5 คำสั่งที่ใช้ติดต่อกับ Motion Control .....	28
2.6 สเต็ปปีงมอเตอร์.....	30
2.6.1 หลักการทำงานของสเต็ปปีงมอเตอร์.....	30
2.6.2 โครงสร้างและหลักการพื้นฐานของสเต็ปปีงมอเตอร์.....	31
2.6.3 สเต็ปปีงมอเตอร์ชนิดต่างๆ.....	33
2.6.4 สัญลักษณ์ต่างๆ ของสเต็ปปีงมอเตอร์.....	37

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# สารบัญ(ต่อ)

	หน้า
2.6.5 พารามิเตอร์ต่างๆ ของสเต็ปปีงมอเตอร์.....	37
2.6.6 การควบคุมการทำงานของสเต็ปปีงมอเตอร์.....	38
2.6.7 การตั้งให้สเต็ปปีงมอเตอร์หมุน.....	39
2.6.8 หลักการออกแบบวงจรขับสเต็ปปีงมอเตอร์.....	40
2.7 Motor Drive.....	40
2.7.1 การติดตั้ง.....	41
2.7.2 รายละเอียดการเชื่อมต่อของวงจรไดรฟ์.....	45
2.7.3 การเชื่อมต่อสัญญาณเข้าและสัญญาณออก.....	48
2.7.4 การปรับแต่งวงจร ไดรฟ์.....	49
2.7.5 รายละเอียดคุณสมบัติของตัววงจรไดรฟ์.....	49
<b>บทที่ 3 การคำนวณและการสร้าง.....</b>	<b>50</b>
3.1 Project Flowchart.....	50
3.2 Hardware Flowchart.....	51
3.2.1 ลำดับความสำคัญในการสร้างชิ้นงาน.....	52
3.3 Software Flowchart .....	53
3.4 Terminal Program Flowchart.....	54
3.5 Point Program Flowchart.....	55
3.6 Main Plot Program Flowchart.....	56
3.7 Line Plot Flowchart.....	58
3.8 Polygon Plot Flowchart.....	59
3.9 Circle Plot Flowchart.....	60
3.10 การพิสูจน์สมการ Inverse Kinematics Equation..	61
3.11 การชดเชย (Compensate) ...	63
<b>บทที่ 4 ผลการทดลอง.....</b>	<b>65</b>
4.1 ตัวอย่างการคำนวณเทียบกับค่าที่คำนวณได้จากโปรแกรมในพื้นที่การทำงานที่ 1...65	
4.2 ตัวอย่างการคำนวณเทียบกับค่าที่คำนวณได้จากโปรแกรมในพื้นที่การทำงานที่ 2...68	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## สารบัญ(ต่อ)

	หน้า
4.3 การทดลองPlot เส้นตรง จากจุด (-10,10) ไปยังจุด (-5,15) .....	72
4.4 การทดลองPlot วงกลม รัศมีเท่ากับ 3 โดยมีจุดศูนย์กลางที่จุด (-5,15) .....	73
4.5 การทดลองPlot หกเหลี่ยม ที่จุด(-7.5,15), (-6.25,17.16), (-3.75,17.16), (-2.5,15), (-3.75,12.84), (-6.25,12.84) .....	74
4.6 การทดลองPlot ตัวอักษร “ฐ” โดยรับ File Plot จากAuto CAD... ..	75
<b>บทที่ 5 สรุปผลการวิจัยและข้อเสนอแนะ.....</b>	<b>78</b>
5.1 ปัญหาและอุปสรรค.....	78
5.2 ข้อเสนอแนะและแนวทางการพัฒนา .....	78
5.3 สรุป.....	79
<b>บรรณานุกรม.....</b>	<b>80</b>
<b>ภาคผนวก .....</b>	<b>81</b>
<b>ภาคผนวก ก. แบบ Drawing.....</b>	<b>81</b>
<b>ภาคผนวก ข. Source Code.....</b>	<b>95</b>

# สารบัญตาราง

ตารางที่	หน้า
2.1 การจัดขาของคอนเน็กเตอร์พอร์ตอนุกรมตามมาตรฐาน RS-232 ทั้งแบบ DB-9 และDB-25...14	
2.2 แสดงข้อมูลในแอดเดรส 0000:0411H ที่ใช้แจ้งจำนวนพอร์ตอนุกรม.....19	
2.3 แสดงผลของโมชันคอนโทรลเมื่อป้อนคำสั่ง H.....28	
2.4 การจ่ายกระแสไฟให้กับสเต็ปปีงมอเตอร์ แบบ Full Step.....39	
2.5 ก) แสดงถึงค่าความต้านทานที่นำมาใช้เมื่อติดตั้งจัมเปอร์ขาที่ 11.....44	
2.5 ข) แสดงถึงค่าความต้านทานที่นำมาใช้เมื่อติดตั้งจัมเปอร์ขาที่ 11.....45	
2.6 แสดงถึงการเซ็ตตัวจัมเปอร์เพื่อกำหนดความละเอียดของวงจรไครฟ์.....46	
2.7 แสดงถึงการเซ็ตจัมเปอร์ขาที่ 6-8.....46	
2.8 แสดงถึงการเซ็ตจัมเปอร์ขาที่ 9-10.....47	
4.1 เปรียบเทียบค่าที่คำนวณได้กับค่าที่ได้จากProgram.....68	
4.2 เปรียบเทียบค่าที่คำนวณได้กับค่าที่ได้จากProgram.....71	

# สารบัญภาพ

ภาพที่	หน้า
2.1 The M.I.T Direct-Drive robot Model IV.....	4
2.2 แสดงกรอบอ้างอิง.....	4
2.3 แสดงการย้าย (Translation).....	5
2.4 รูปแสดงสมการการย้ายแกน.....	6
2.5 แสดงสมการการหมุน.....	6
2.6 รูปและสมการ Rotation Matrix.....	7
2.7 แสดง Rigid Body Motion และ Two-dimensional of Rigid Body Motion.....	8
2.8 แสดงความสัมพันธ์กันระหว่าง Direct Kinematics กับ Inverse Kinematics.....	8
2.9 รูปแบบอย่างง่ายที่สุดของข้อมูลอนุกรม.....	11
2.10 รูปแบบอย่างง่ายที่สุดของข้อมูลอนุกรมแบบอะตอมข้อมูลแบบอะซิงโครนัส.....	12
2.11 การจัดขาของคอนเน็คเตอร์พอร์ตอนุกรมตามมาตรฐาน RS-232 ทั้งแบบ DB-9 และ DB-25.....	13
2.12 การต่ออุปกรณ์ภายนอกกับพอร์ตอนุกรมของคอมพิวเตอร์ในลักษณะต่างๆ.....	14
2.13 ไดอะแกรมแสดงโครงสร้างทางฮาร์ดแวร์ของพอร์ตอนุกรม.....	18
2.14 แสดงคอนโทรลเลอร์ สำหรับควบคุมการทำงานของสเต็ปปีงมอเตอร์.....	30
2.15 (ก) แสดงสเต็ปปีงมอเตอร์ที่มีการต่อวงจรขดลวดภายในเพื่อกระตุ้นให้เกิดขั้วแม่เหล็กขึ้น 1 ขั้ว ในทิศทางตรงกันข้าม ส่วนขดลวดอื่นๆ จะไม่ถูกกระตุ้น.....	31
2.15 (ข) แสดงการต่อวงจรขดลวดแบบกระตุ้นให้เกิดขั้วแม่เหล็กพร้อมกัน 2 ขั้ว ที่อยู่ใกล้กัน.....	31
2.16 แสดงวงจรจ่ายกำลังไฟฟ้าให้กับสเต็ปปีงมอเตอร์ทั้งสองแบบ ก) สำหรับชนิดยูนิโพลาร์ ซึ่งใช้ทรานซิสเตอร์สวิตช์เพียงตัวเดียวต่อ 1 คอยล์.....	32
ข) สำหรับชนิดไบโพลาร์ ซึ่งใช้ทรานซิสเตอร์สวิตช์ 4 ตัวต่อ 1 คอยล์.....	32
2.17 แสดงแรงดึงดูด ทำให้เกิดทอร์กที่หมุนโรเตอร์ให้ไปอยู่ในตำแหน่งที่สมดุล.....	32
2.18 แสดงโครงสร้างของสเต็ปปีงมอเตอร์แบบแม่เหล็กถาวร.....	33
2.19 แสดงการทำงานในสเต็ปปีงมอเตอร์แบบแม่เหล็กถาวรขนาดสี่เฟส.....	33
2.20 สเต็ปปีงมอเตอร์แบบแปรค่ารีล็กแทนซ์ได้.....	34
2.21 แสดงตำแหน่งสมดุลเมื่อเฟลไดเฟสหนึ่งของสเต็ปปีงมอเตอร์ถูกกระตุ้น.....	34
2.22 แสดงการเปลี่ยนลำดับสเต็ปปีงในการเคลื่อนที่ของสเต็ปปีงมอเตอร์ได้.....	35
2.23 แสดงการเปลี่ยนลำดับสเต็ปปีงในการเคลื่อนที่ของสเต็ปปีงมอเตอร์ได้.....	36

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## สารบัญภาพ(ต่อ)

ภาพที่	หน้า
2.24 แสดงหลักการทำงานของสตีปิ้งมอเตอร์แบบไฮบริด.....	37
2.25 แสดงบล็อกไดอะแกรมของระบบควบคุมการขับเคลื่อนมอเตอร์.....	39
2.26 แสดงวงจรสมมูลย์ของสตีปิ้งมอเตอร์.....	40
2.27 แสดงถึงโครงสร้างของชุดไดร์ฟ Compumotor รุ่น OEM 650.....	41
2.28 แสดงถึงการเชื่อมต่อของสายมอเตอร์เข้าวงจรไดร์ฟเมื่อต่อขดลวดมอเตอร์แบบขนาน.....	42
2.29 แสดงถึงการเชื่อมต่อตัวต้านทานเพื่อจำกัดกระแสขับเคลื่อนมอเตอร์.....	43
2.30 แสดงถึงการเชื่อมต่อแหล่งจ่ายไฟเข้ากับชุดวงจรไดร์ฟ.....	47
2.31 แสดงถึงรูปแบบการเชื่อมต่อภายในพอร์ต DB-25 ภายในภาครับของชุดวงจรไดร์ฟ.....	48
2.32 แสดงถึงคุณสมบัติแรงบิดของมอเตอร์เมื่อทำงานที่ความเร็วต่างๆกัน.....	49
3.1 แสดงProject Flowchart.....	50
3.2 Hardware Flowchart.....	51
3.3 Software Flowchart.....	53
3.4 Terminal Program Flowchart.....	54
3.5 Point Program Flowchart.....	55
3.6 Main Plot Flowchart.....	56
3.7 Line Plot Flowchart.....	58
3.8 Polygon Plot Flowchart.....	59
3.9 Circle Plot Flowchart.....	60
3.10 แสดงบริเวณการทำงานของแกนกลขับเคลื่อน ในพื้นที่ที่ 1 และพื้นที่ที่ 2.....	61
3.11 แสดงค่ามุมในพื้นที่ที่ 1 มีทิศวนเข็มนาฬิกา.....	61
3.12 แสดงค่ามุมในพื้นที่ที่ 2 มีทิศตามเข็มนาฬิกา.....	62
3.13 แสดงการแบ่งระยะการเคลื่อนที่ โดย dx และ dy.....	63
3.14 แสดงการชดเชยระยะการเคลื่อนที่ของแกนกล.....	64
4.1 แสดงทิศทางของ $\theta_A$ และ $\theta_B$ ซึ่งจะมีค่าเป็นบวกเมื่อมีทิศวนเข็มนาฬิกา.....	65
4.2 แสดงการหาค่า $\theta_A$ และ $\theta_B$ ในพื้นที่ ที่1.....	65
4.3 แสดงโปรแกรมการคำนวณในพื้นที่ที่1.....	67
4.4 แสดงค่าที่คำนวณได้จากโปรแกรม.....	67

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## สารบัญภาพ(ต่อ)

ภาพที่	หน้า
4.5 แสดงการหาค่า $\theta_A$ และ $\theta_B$ ในพื้นที่ที่ 2.....	68
4.6 แสดงโปรแกรมการคำนวณในพื้นที่ที่ 2.....	70
4.7 แสดงค่าที่คำนวณได้จากโปรแกรม.....	70
4.8 แสดงโปรแกรมการคำนวณเกี่ยวกับเรขาคณิต.....	71
4.9 แสดงProgram การทดลองPlot เส้นตรง จากจุด (-10,10) ไปยังจุด (-5,15) ... ..	72
4.10 แสดงการเคลื่อนที่ของ แขนกล Plot เส้นตรงจากจุด (-10,10) ไปยังจุด (-5,15) ... ..	72
4.11 แสดงProgram การทดลองPlot วงกลม รัศมีเท่ากับ 3 โดยมีจุดศูนย์กลางที่จุด (-5,15) ... ..	73
4.12 แสดงการเคลื่อนที่ของ แขนกล Plot วงกลม รัศมีเท่ากับ 3 โดยมีจุดศูนย์กลางที่จุด (-5,15)	73
4.13 แสดงProgram การทดลองPlot หกเหลี่ยม ที่จุด(-7.5,15), (-6.25,17.16), (-3.75,17.16), (-2.5,15), (-3.75,12.84), (-6.25,12.84) ... ..	74
4.14 แสดงการเคลื่อนที่ของ แขนกล Plot หกเหลี่ยม ที่จุด(-7.5,15), (-6.25,17.16) (-3.75,17.16), (-2.5,15), (-3.75,12.84), (-6.25,12.84) ... ..	74
4.15 แสดงตัวอย่างการเขียนตัวอักษรผ่านทาง AutoCAD... ..	75
4.16 แสดงProgram ที่ทำการรับ FilePlot จาก AutoCAD.....	75
4.17 แสดงProgram ที่ทำการรับเวคเตอร์จาก FilePlot มาคำนวณแล้วทำการแปลงเป็น โปรแกรมย่อยส่งให้ตัว Motion Control.....	76
4.18 แสดงผลการของแขนกลที่รับค่าทาง AutoCAD.....	76
4.19 แสดงโปรแกรมการติดต่อสื่อสารกับตัวควบคุมการเคลื่อนที่.....	77
ก.1 แสดงแบบ Drawing Arm A&B.....	82
ก.1 แสดงแบบ Drawing Arm C.....	83
ก.1 แสดงแบบ Drawing Arm DE.....	84
ก.1 แสดงแบบ Drawing Arm Lock .....	85
ก.1 แสดงแบบ Drawing Back Base.....	86
ก.1 แสดงแบบ Drawing Base.....	87
ก.1 แสดงแบบ Drawing Buch.....	88
ก.1 แสดงแบบ Drawing Catch Motor.....	89
ก.1 แสดงแบบ Drawing Front Base.....	90

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## สารบัญภาพ(ต่อ)

ภาพที่	หน้า
ก.1 แสดงแบบ Drawing Motor Base.....	91
ก.1 แสดงแบบ Drawing Motor Up & Down.....	92
ก.1 แสดงแบบ Drawing ภาพสามมิติ .....	93
ก.1 แสดงแบบ Drawing ภาพสามมิติ.....	94
ข.1 แสดงหน้าจอ Main (หน้าหลัก).....	96
ข.2 แสดงหน้าจอ Terminal .....	96
ข.3 แสดงหน้าจอ Line (รูปเส้นตรง) .....	97
ข.4 แสดงหน้าจอ Triangle (รูปสามเหลี่ยม).....	97
ข.5 แสดงหน้าจอ Rectangular (รูปสี่เหลี่ยม).....	98
ข.6 แสดงหน้าจอ Polygon (รูปหลายเหลี่ยม).....	98
ข.7 แสดงหน้าจอ Circle (รูปวงกลม) .....	99
ข.8 แสดงหน้าจอ Auto CAD.....	99

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# บทที่ 1

## บทนำ

### 1.1 ความเป็นมาและเหตุจูงใจ

เนื่องจากการทำงานจริงจำเป็นที่จะต้องเรียนรู้เกี่ยวกับระบบการควบคุมวงจรถืออิเล็กทรอนิกส์การเขียนโปรแกรมรวมถึงความรู้ทางกล เมื่อพิจารณาถึงหัวข้อต่างๆเหล่านี้ จึงได้เลือกแขนกลมาศึกษา และได้เลือกแขนกลซับซ้อนขึ้นมาเป็นตัวอย่างในการศึกษาเนื่องจากการควบคุมที่ง่ายและมีความแม่นยำสูง สามารถนำไปประยุกต์ใช้งานกับงานบางประเภทที่แขนกลแนวแกน X,Y ซึ่งไม่สามารถเข้าถึงได้

### 1.2 วัตถุประสงค์ของปริญญานิพนธ์

ปริญญานิพนธ์นี้ได้ศึกษาถึงโครงสร้างของแขนกล การควบคุมการเคลื่อนที่ การสร้างโปรแกรมที่เขียนขึ้น โดยอาจเป็นการควบคุมทางเดินจุดต่อจุด หรือ การป้อนสมการเพื่อให้แขนกลทำการเคลื่อนที่ตามสมการทางเดิน และสามารถนำไปประยุกต์ใช้กับงานด้านอื่นๆได้

### 1.3 ขอบเขตของปริญญานิพนธ์

ปริญญานิพนธ์ฉบับนี้ จะกล่าวถึงแนวคิดในการสร้างแขนกล ความสามารถของแขนกล การควบคุมการเคลื่อนที่ซึ่งประกอบไปด้วย ส่วนโปรแกรมวิซวลเบสิกที่ทำหน้าที่ติดต่อสื่อสารเพื่อจัดการป้อนตำแหน่งให้กับตัววงจรควบคุมการเคลื่อนที่ซึ่งไปขับสัญญาณเข้าสู่ตัวสเต็ปมอเตอร์ผ่านวงจรไดรฟ์

### 1.4 ขั้นตอนการศึกษา

การทำปริญญานิพนธ์นี้เริ่มทำการศึกษาจากชนิดของแขนกลชนิดต่างๆพิจารณาข้อดีข้อเสีย และความเป็นไปได้ในการสร้างแขนกลซับซ้อนแต่ละแบบศึกษาสมการการเคลื่อนที่ (Kinematic & Inverse Kinematic) ศึกษาการทำงานของอุปกรณ์ต่างๆในการสร้างแขนกล ศึกษาการเขียนและใช้โปรแกรมวิซวลเบสิก ศึกษาการทำงานของชุดวงจรควบคุมการเคลื่อนที่ วงจรขับสเต็ปมอเตอร์ ตัวสเต็ปมอเตอร์ และ ศึกษาการติดต่อสื่อสารผ่านพอร์ตอนุกรม

## บทที่ 2

# ทฤษฎีและหลักการ

## 2.1 ความรู้เบื้องต้นเกี่ยวกับโรบอท

### 2.1.1 Hydraulic Drives

จะมีความแม่นยำสูงและมีแรงขับเคลื่อนข้างสูงภายใต้โหลดที่มีขนาดใหญ่ แต่ Electrically Power Robot มักจะใช้กับการทำงานที่มีโหลดเบาๆ จึงมีราคาถูกและสะดวกกว่า Hydraulic Drives ทำให้การบำรุงรักษาย่างมอเตอร์ไฟฟ้าจะผลิต Maximum Power ที่ความเร็วสูงแต่ แขนของโรบอทเคลื่อนที่ด้วยความเร็วต่ำจึงทำให้มีแรงบิดต่ำไปด้วย ดังนั้นจึงต้องมีการใช้เฟืองทดเข้ามาช่วย การออกแบบจึงต้องคำนึงถึงอัตราทดของเฟืองและต้องคำนึงถึงการรักษาความแม่นยำให้คงอยู่ด้วย เพราะการใช้เฟืองจะมีข้อผิดพลาดที่เรียกว่า Backlash ซึ่งเกิดจากเฟืองไม่แนบสนิทกันพอดี จึงทำให้ความแม่นยำลดลงเพราะแม้ว่าจะมี Backlash เพียงเล็กน้อยที่ตัวขับก็จะนำไปสู่ค่าผิดพลาดจำนวนมากที่จุดปลาย

อีกปัญหาหนึ่งสำหรับการใช้เฟืองทดก็คือ ความเสียดทาน (Friction) เกิดจากการ แก่ Backlash โดยการทำให้เฟืองแน่นหรือพอดีกัน ในทางกลับกันจึงเกิดแรงเสียดทานเป็นจำนวนมาก ซึ่งอาจทำให้กำลังที่ตัวขับจ่ายไปขับ Link สูญเสียไปกับแรงเสียดทานเป็นจำนวนมาก ทำให้ความแม่นยำลดลงและยากที่จะแก้ไข

คุณสมบัติที่โรบอทควรมีคือ

1. ความยืดหยุ่น ( Flexibility )
2. ความคล่องแคล่ว ( Dexterity )
3. ความเฉลียวฉลาด ( Intelligent )
4. ความสามารถในการทำซ้ำ ( Repeatability )
5. ความแม่นยำ ( Accuracy )

การออกแบบจึงต้องคำนึงถึงคุณสมบัติเหล่านี้ ดังนั้นเมื่อก้าวถึงการควบคุมตำแหน่งที่แม่นยำซึ่งไม่ได้ให้ความสำคัญกับการควบคุมแรง Hydraulic Drives ซึ่งมีความสามารถในการจ่าย Load ขนาดใหญ่ แต่มีปัญหาเกี่ยวกับการควบคุมตำแหน่งที่ละเอียดอ่อนจึงไม่ถูกนำมาพิจารณา

### 2.1.2 Direct Drives Robot

คือการควบคุมอย่างง่ายจากตัวขับเคลื่อนไฟฟ้าโดยตรงสู่ Load ซึ่งไม่มีตัวทด (Reducer) มาช่วย ทำให้ขจัดปัญหาด้าน Mechanic ได้เป็นอย่างดี เพราะโครงสร้างง่ายๆของมัน ประกอบไปด้วย Armlinks, Motors, Baring (ตลับลูกปืน) โดยที่ Rotor จะยึดอยู่ที่จุดเชื่อมของแท่ง แขน ดังนั้นแรงบิดที่จุดปลายจะถูกส่งตรงมาจาก Rotor และ Stator ของ Motor ทำให้ขจัดปัญหา เรื่อง Backlash และยังคงแรงเสียดทาน ไปได้เป็นจำนวนมาก แต่ Direct Drive ก็มีข้อเสียเช่นกัน คือ มีความฝืด (Stiffness) ในโครงสร้างมากกว่าการขับเคลื่อนด้วยระบบเฟือง แต่

#### ข้อดีของ Direct Drive ก็มีหลายประการด้วยกันคือ

1. มีความแม่นยำสูง เนื่องจากพลวัตของระบบนี้สามารถสร้างขึ้นเป็น Model ได้ ดังนั้นผลตอบสนองทาง Dynamic จึงสามารถคำนวณและทำนายได้
2. มีความสามารถในการทำซ้ำสูง เนื่องจากมีความผิดพลาดทาง Mechanic ต่ำ
3. มีโครงสร้างที่ง่าย ทำให้บำรุงรักษาได้ง่าย

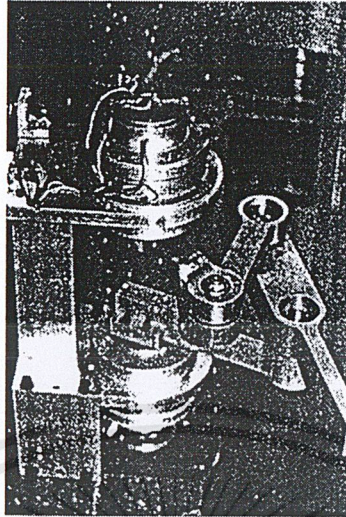
#### ปัญหาที่ต้องแก้ไขในการออกแบบควบคุม Direct Drive

1. ง่าย Load ได้น้อย และแขนที่มีน้ำหนักมาก ( Small Pay Load and Heavy Armweight )
2. คุณสมบัติการหน่วง (Damping Characteristic ) เนื่องจาก Direct Drive ส่วนใหญ่ จะมี Damping Ratio น้อยมาก ซึ่งน้อยกว่าค่า Damping ที่จำเป็นต่อการตอบสนอง ดังนั้นเราจึงต้องเพิ่มการหน่วงในการออกแบบแก้ไขได้โดยการเพิ่ม Velocity Feedback Compensation
3. ความฝืด ( Servo Stiffness ) แก้ไขได้โดยการเพิ่มค่า Gain ของ Controller เพื่อชดเชย
4. Motor Inductance Effects เนื่องจากแรงบิด ( Torque ) จะเป็นสัดส่วนกับ  $T = L/R$  ดังนั้น การเพิ่มแรงบิดจึงต้องเพิ่ม L ส่งผลให้เกิด Inductance เพิ่มขึ้น จึงส่งผลกระทบต่อ Damping Ratio ทำให้เกิดสถานะไม่เสถียรขึ้น จึงต้องออกแบบให้มีการตอบสนองที่เร็วพอ เพื่อให้เหมาะสมกับค่า Inductance ที่มีค่ามาก
5. Arm Dynamic Effects เกิดจากการควบคุมที่ไม่ Linear
6. Over Heat เกิดจากการขับ Load อย่างต่อเนื่อง

จากการได้ศึกษาชนิด และ รูปแบบต่างของแขนกลแล้ว ได้เลือก The M.I.T Direct-Drive robot

Model IV มาเป็นกรณีศึกษา

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 2.1 The M.I.T Direct-Drive robot Model IV

## 2.2 ทฤษฎีที่เกี่ยวข้อง (Theory)

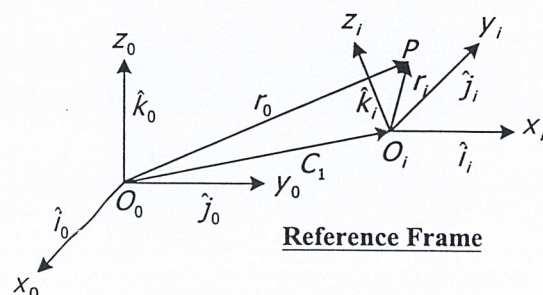
### 2.2.1 พื้นฐานกลศาสตร์ที่เกี่ยวข้องกับการเคลื่อนที่ (Basic Kinematics)

#### 2.2.1.1 กลศาสตร์ที่เกี่ยวข้องกับการเคลื่อนที่ (Kinematics)

ทฤษฎีที่อธิบายถึงเรขาคณิตของการเคลื่อนที่ของจุด โดยพูดถึงตำแหน่งที่จุดอาจจะเคลื่อนที่ผ่านในรูปของการสะสมของจุด (หรือ เส้นโค้ง: Curves) โดยมีองค์ประกอบที่เกี่ยวข้อง คือ ตำแหน่ง (Position) , ความเร็ว ( Velocity ) และ ความเร่ง ( Acceleration )

#### 2.2.1.2 กรอบอ้างอิง (Reference Frames)

วัตถุแข็งเกร็งที่มีการเคลื่อนที่บางครั้งเราอาจกำหนดให้โครงสร้างของ Robot เป็นกรอบอ้างอิง (Reference Frames) เพื่อช่วยในการอ้างอิงจุดเทียบกับกรอบอ้างอิงต่างๆ ที่จุด origin ( $O_{xyz}$ )<sub>i</sub> เมื่อ  $i = 0, 1, 2, 3, \dots, n$  โดยที่เมื่อ  $i = 0$  จุด ( $O_{xyz}$ )<sub>0</sub> นี้จะเรียกว่า Fixed Reference Frame หรือ Base Reference Frame เมื่อนำไปเทียบกับ Reference Frame อื่นๆ ดังรูป



ภาพที่ 2.2 แสดงกรอบอ้างอิง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยที่  $r_0 =$  เวกเตอร์ที่บอกตำแหน่งของจุด P เทียบกับ fixed reference frame

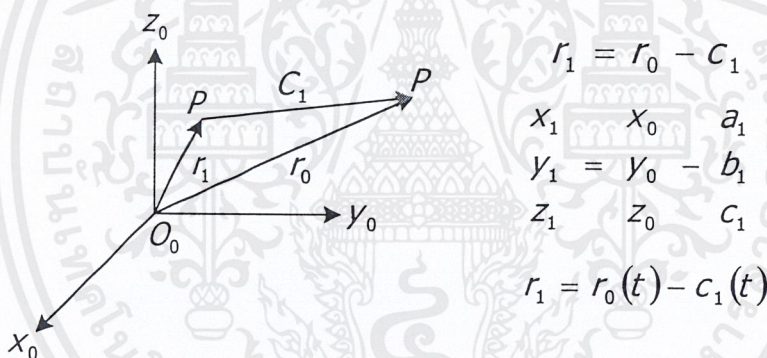
$r_i =$  เวกเตอร์ที่บอกตำแหน่งของจุด P เทียบกับ จุด  $O_i$  ที่อาจเป็น Moving Frame หรือไม่ได้

สามารถเขียน  $r_i$  ในรูปของ Vector และ Matrix ได้ดังนี้

$$r_i = x_i \hat{i}_i + y_i \hat{j}_i + z_i \hat{k}_i \quad \underline{\text{or}} \quad r_i = \begin{bmatrix} x_i \\ y_i \\ z_i \end{bmatrix} = [x_i \ y_i \ z_i]^T \quad (2.1)$$

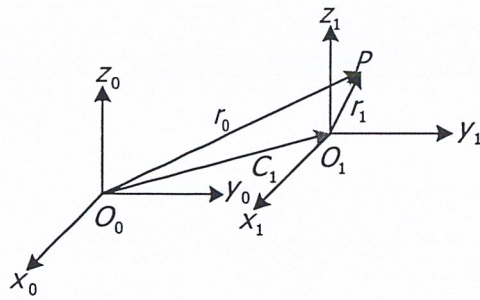
### 2.2.1.3 การย้าย ( Translation )

เป็นการศึกษาการเปลี่ยนแปลงของจุด P เมื่อเทียบกับกรอบอ้างอิงเดียวกัน หรือ การเปลี่ยนแปลงพิกัดของจุด P เมื่อเทียบกับกรอบอ้างอิงที่แตกต่างกัน 2 กรอบ โดยที่การ Translation แบ่งออกเป็น 2 รูปแบบดังนี้ คือ



ภาพที่ 2.3 แสดงการย้าย (Translation)

- 1) Point Transformation คือ จุด P เปลี่ยนเป็น P แต่ Coordinate คงเดิม  $r_0$  เป็นฟังก์ชันของเวลา เนื่องจากจุด P เป็น Moving Point
- 2) Coordinate Transformation คือ จุด P เดียวกัน แต่เทียบกับกรอบอ้างอิงที่แตกต่างกัน 2 กรอบ



$$\hat{i}_0 = \hat{i}_1, \hat{j}_0 = \hat{j}_1, \hat{k}_0 = \hat{k}_1$$

$$r_1 = r_0 - c_1$$

$$x_1 = x_0 - a_1$$

$$y_1 = y_0 - b_1$$

$$z_1 = z_0 - c_1$$

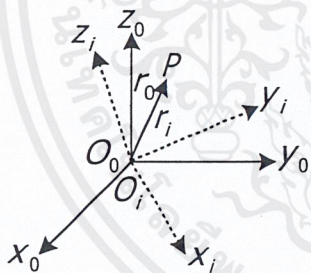
$$r_1(t) = r_0 - c_1(t)$$

ภาพที่ 2.4 รูปแสดงสมการการย้ายแกน

$r_1$  เป็นฟังก์ชันของเวลาเนื่องจาก  $O_1$  เป็น Moving Frame

### 2.2.1.4 เมตริกซ์การหมุน (Rotation Matrix)

เป็น Matrix ที่ใช้ในการอธิบายการหมุนของ กรอบอ้างอิงที่มีการเคลื่อนที่ (Moving Reference Frame) เมื่อเทียบกับ กรอบอ้างอิงที่ไม่มีการเคลื่อนที่ (Fixed Reference Frame) ในหัวข้อนี้เราจะให้ความสนใจกับการหมุนเท่านั้นดังนั้นเราจึงกำหนดให้  $(O_{xyz})_0$  เป็นจุดเดียวกับ  $(O_{xyz})_i$  ดังนั้นจากรูปเราจะเห็นได้ว่า



$$r_0 = r_i$$

$$x_0 \hat{i}_0 + y_0 \hat{j}_0 + z_0 \hat{k}_0 = x_i \hat{i}_i + y_i \hat{j}_i + z_i \hat{k}_i$$

ภาพที่ 2.5 แสดงสมการการหมุน

จากสมการด้านบนนำมาหา dot product จะได้สมการดังนี้

$$x_0 = \hat{i}_0 \cdot (x_i \hat{i}_i + y_i \hat{j}_i + z_i \hat{k}_i) \text{ --- take } \hat{i}_0$$

$$y_0 = \hat{j}_0 \cdot (x_i \hat{i}_i + y_i \hat{j}_i + z_i \hat{k}_i) \text{ --- take } \hat{j}_0$$

$$z_0 = \hat{k}_0 \cdot (x_i \hat{i}_i + y_i \hat{j}_i + z_i \hat{k}_i) \text{ --- take } \hat{k}_0$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จัดสมการในรูปของ Matrix จะได้ดังนี้

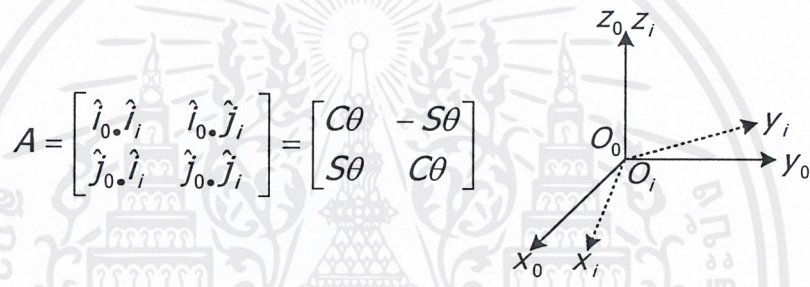
$$\begin{bmatrix} x_0 \\ y_0 \\ z_0 \end{bmatrix} = \begin{bmatrix} \hat{i}_0 \cdot \hat{i}_i & \hat{i}_0 \cdot \hat{j}_i & \hat{i}_0 \cdot \hat{k}_i \\ \hat{j}_0 \cdot \hat{i}_i & \hat{j}_0 \cdot \hat{j}_i & \hat{j}_0 \cdot \hat{k}_i \\ \hat{k}_0 \cdot \hat{i}_i & \hat{k}_0 \cdot \hat{j}_i & \hat{k}_0 \cdot \hat{k}_i \end{bmatrix} \begin{bmatrix} x_i \\ y_i \\ z_i \end{bmatrix}$$

หรือ

$$r_0 = A_i r_i$$

โดยเรียก A ว่า Rotation Matrix

แต่จากโครงสร้างของ Robot พื้นที่การทำงานจะเป็น 2 มิติ โดยจะหมุน



ภาพที่ 2.6 รูปและสมการ Rotation Matrix

รอบแกน Z ดังนั้นจะได้ A ดังนี้ เมื่อ  $\theta$  คือ มุมระหว่าง แกน  $x_0$  กับ  $x_i$  และ  $y_0$  กับ  $y_i$  ที่เกิดขึ้นจากการหมุนรอบแกน Z และ  $C\theta$  กับ  $S\theta$  คือ  $\cos\theta$  กับ  $\sin\theta$  ตามลำดับ

### 2.2.2 การเคลื่อนที่ของวัตถุแข็งเกร็ง ( Rigid Body Motion )

ในหัวข้อนี้จะเป็นการอธิบาย การเคลื่อนที่ของวัตถุแข็งเกร็งโดยพิจารณาทั้ง Rotation และ Translation ในการเคลื่อนที่ของวัตถุแข็งเกร็ง

จากหัวข้อ Rotation Matrix จะได้

$$r' = A_1 r_1$$

โดยที่  $r'$  คือ เวกเตอร์ที่มี

$$\hat{i}_0 = \hat{i}', \hat{j}_0 = \hat{j}', \hat{k}_0 = \hat{k}'$$

เมื่อนำไปแทนใน Translation จะได้สมการรวมดังนี้

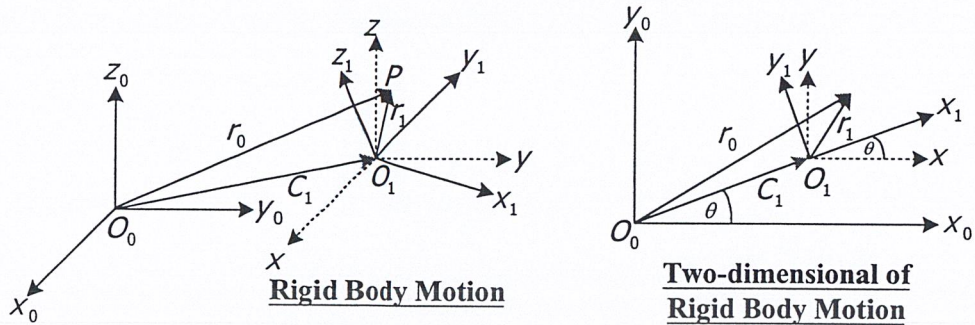
$$r_0 = C_1 + A_1 r_1$$

แต่จากการทำงานของ Robot เราจะควบคุมเป็นระนาบ 2 มิติ ที่หมุนรอบ

$$\begin{bmatrix} X_0 \\ Y_0 \end{bmatrix} = \begin{bmatrix} a_1 \\ b_1 \end{bmatrix} + \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix} \begin{bmatrix} X_1 \\ Y_1 \end{bmatrix}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้เพื่อการศึกษาค้นคว้าเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แกน Z ดังนั้นสมการ Matrix ที่ได้จะเป็นดังนี้

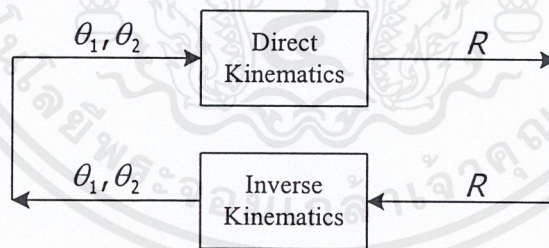


ภาพที่ 2.7 แสดง Rigid Body Motion และ Two-dimensional of Rigid Body Motion

### 2.2.3 กลศาสตร์ที่เกี่ยวข้องกับการเคลื่อนที่โดยตรง และ ผันกลับ ( Direct & Inverse Kinematics Equation )

**Direct Kinematics** เป็นสมการที่ใช้อธิบายการควบคุมเวกเตอร์ลัพธ์ (R) ของ robot โดยการป้อนพารามิเตอร์ต่างๆเช่น มุม ( $\theta_1, \theta_2$ ), ความยาวของ แต่ละ Link (L) เป็นต้น หรือ  $R=f(\theta_1, \theta_2, L)$  นั่นเอง

**Inverse Kinematics** เป็นการคำนวณหา พารามิเตอร์ต่างๆ ณ จุดปลาย (R) ต่างๆ เพื่อใช้ในการป้อนค่าไปควบคุม



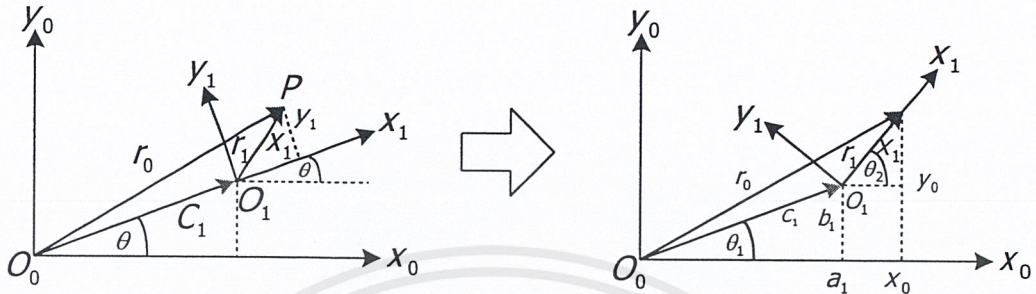
ภาพที่ 2.8 แสดงความสัมพันธ์กันระหว่าง Direct Kinematics กับ Inverse Kinematics

จากหลักการของ Direct & Inverse Kinematics จะเห็นว่า Inverse Kinematics จะเหมาะสมกับการที่เราทราบตำแหน่งของเวกเตอร์ลัพธ์ (R)จากนั้นจึงนำมาคำนวณหา พารามิเตอร์ที่ใช้ควบคุม จากนั้นจึงนำ พารามิเตอร์เหล่านี้ไปป้อนให้กับ robot เพื่อให้ได้ตำแหน่งลัพธ์ตามที่ต้องการ

จากทฤษฎีที่ได้ศึกษามา เมื่อนำมาประยุกต์ใช้กับ Robot ที่ได้ศึกษา ทำให้สร้างสมการต่างๆได้ดังนี้

2.2.3.1 Direct Kinematics Equation

จากรูปด้านล่าง เราเลือกที่จะให้จุด P ตกบนแกน  $x_1$  เพื่อง่ายต่อการพิจารณา ทำให้ขนาดของ  $r_1 = x_1$  ( $y_1 = 0$ ) และ เมื่อพิจารณาตามเรขาคณิต จะได้อ่างดังนี้



$$a_1 = |C_1| \cos \theta_1, b_1 = |C_1| \sin \theta_1, y_1 = 0$$

$$\begin{aligned} x_0 &= |C_1| \cos \theta_1 + |r_1| \cos \theta_2 \\ y_0 &= |C_1| \sin \theta_1 + |r_1| \sin \theta_2 \end{aligned}$$

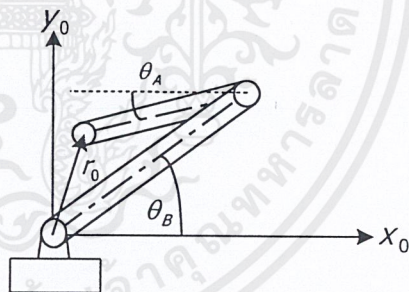
เมื่อนำมาพิจารณาถึง robot รวมถึงการติดตั้งจะได้อ่างดังนี้

จากการออกแบบจะได้

$$|C_1| = |r_1| = L$$

$$\theta_1 = \theta_B$$

$$\theta_2 = 180 + \theta_A$$



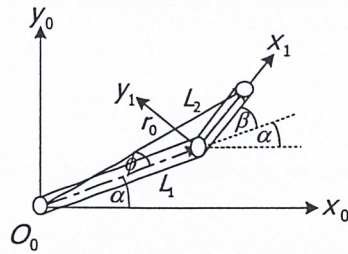
จากสมการด้านบน และข้อมูลที่มีทำให้เราสามารถสร้าง Direct Kinematics Equation ได้อ่างดังนี้

$$y_0 = L[\sin \theta_B + \sin(180 + \theta_A)]$$

$$x_0 = L[\cos \theta_B + \cos(180 + \theta_A)]$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 2.3.2 Inverse Kinematics Equation



$$r_0^2 = L_1^2 + L_2^2 + 2L_1L_2 \cos \beta$$

$$\beta = \cos^{-1} \frac{r_0^2 - L_1^2 - L_2^2}{2L_1L_2}$$

$$r_0 \sin \phi = L_2 \sin \beta$$

$$\phi = \sin^{-1} \frac{L_2}{r_0} \sqrt{1 - \cos^2 \beta}$$

และ จาก

$$\tan(\phi + \alpha) = \frac{y_0}{x_0}$$

$$\alpha = \tan^{-1} \frac{y_0}{x_0} - \sin^{-1} \frac{L_2}{r_0} \sqrt{1 - \cos^2 \beta}$$

เมื่อนำมาเปรียบเทียบกับrobotที่ทำการติดตั้ง แล้วแทนค่า  $L_2 = L_1 = L$  และ  $r_0 = \sqrt{(x_0^2 + y_0^2)}$  ลงในสมการด้านบน จะได้ Inverse Kinematics Equation ดังนี้

$$\theta_2 = 180 + \theta_A = \beta + \alpha$$

$$\theta_A = \cos^{-1} \frac{r_0^2 - 2L^2}{2L^2} + \tan^{-1} \frac{y_0}{x_0} - \sin^{-1} \frac{L}{r_0} \sqrt{1 - \frac{r_0^2 - 2L^2}{2L^2}^2} - 180$$

และ

$$\theta_1 = \theta_B = \alpha$$

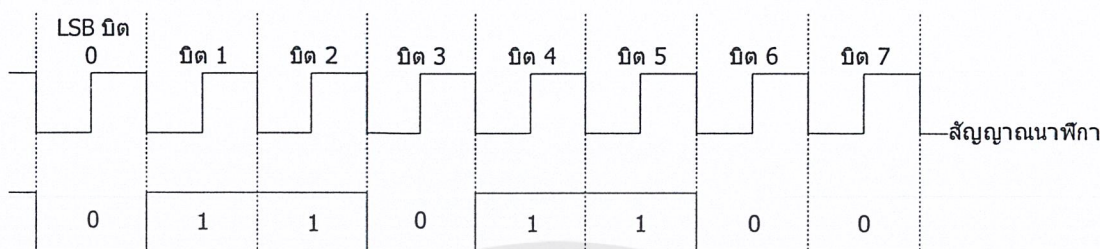
$$\theta_B = \tan^{-1} \frac{y_0}{x_0} - \sin^{-1} \frac{L}{r_0} \sqrt{1 - \frac{r_0^2 - 2L^2}{2L^2}^2}$$

### 2.3 การสื่อสารแบบอนุกรม

การสื่อสารแบบอนุกรมนี้จะแบ่งออกได้เป็น 2 แบบคือการสื่อสารอนุกรมแบบซิงโครนัส และการสื่อสารอนุกรมแบบอะซิงโครนัส การสื่อสารแบบซิงโครนัสจะมีสัญญาณนาฬิกา ร่วมอยู่กับการรับและส่งสัญญาณด้วย ตัวอย่างการส่งข้อมูลแบบซิงโครนัสก็คือรีซีปอร์ตของคอมพิวเตอร์ ซึ่งสายเส้นหนึ่งจะเป็นสายของนาฬิกา ส่วนสายอีกเส้นจะเป็นสายของข้อมูล ดังนั้นการติดต่อกันแบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ซิงโครนัสนี้จะต้องใช้สายในการเชื่อมต่ออย่างน้อยที่สุด 3 เส้นคือ สัญญาณนาฬิกา, ข้อมูล และกราวด์ ภาพที่ 2.9 แสดงให้เห็นถึงไทมิงไคอะแกรมของการส่งข้อมูลแบบซิงโครนัส



ภาพที่ 2.9 รูปแบบอย่างง่ายที่สุดของข้อมูลอนุกรม

### 2.3.1 การสื่อสารข้อมูลแบบอะซิงโครนัส

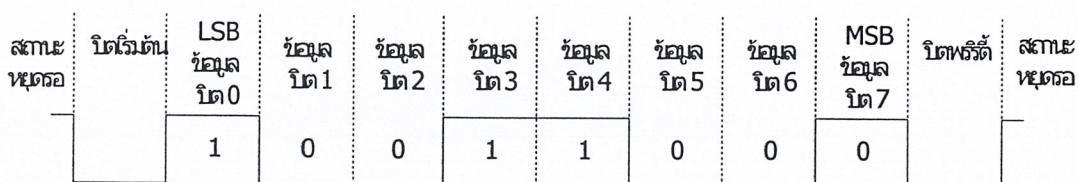
การสื่อสารข้อมูลแบบอะซิงโครนัสคือการรับ และ ส่งข้อมูลไปในสายโดยไม่จำเป็นต้องมีสัญญาณนาฬิกาพร้อมด้วยเหมือนกับการรับส่งข้อมูลแบบซิงโครนัส แต่จะใช้การกำหนดค่าสัญญาณนาฬิกาทั้งภาครับและภาคส่งให้มีค่าเท่ากัน ซึ่งเรียกสัญญาณนาฬิกาที่ใช้ในการกำหนดค่าให้ภาครับ และภาคส่งนี้ว่า อัตราการถ่ายทอดข้อมูล หรือ บอดเรต (baudrate) มีหน่วยเป็น บิตต่อวินาที (bit per secone: bps)

รูปแบบของข้อมูลที่ใช้ในการรับส่งแบบอะซิงโครนัสประกอบด้วย 4 ส่วนด้วยกันคือ

1. บิตเริ่มต้น (Start Bit) ซึ่งจะมีขนาด 1 บิต
2. บิตข้อมูลแบบอนุกรมจะมีขนาด 5,6,7 หรือ 8 บิต
3. บิตตรวจสอบพาริตี (Parity Bit) จะมีขนาด 1 บิตหรือไม่มี
4. บิตปิดท้าย (Stop Bit) จะมีขนาด 1,1.5 หรือ 2 บิต

ภาพที่ 2.10 แสดงรูปแบบของข้อมูลอนุกรมแบบอะซิงโครนัส ซึ่งเมื่อไม่มีข้อมูลที่ส่ง ขา DATA จะมีสถานะลอจิก “1” ซึ่งจะเรียกสถานะนี้ว่าสถานะหยุดรอ (waiting stage) การเริ่มต้นจากนั้นบิตข้อมูลจะถูกส่งออกไป โดยเริ่มจากบิตที่มีนัยสำคัญต่ำที่สุด (LSB) ก่อน ซึ่งข้อมูลในไบต์ที่จะส่ง อาจจะมีจำนวนบิต 5,6,7 หรือ 8 บิตก็ได้ จากนั้นจะตามด้วยบิตพาริตี ซึ่งใช้เพื่อตรวจสอบความผิดพลาดที่เกิดขึ้นจากการส่งข้อมูล บิตสุดท้ายที่จะส่งคือบิตปิดท้าย ซึ่งจะใช้เวลาตามสถานะลอจิก ‘1’ อีกครั้งด้วยระยะเวลาอย่างน้อย 1 บิต, 1.5 บิต หรือ 2 บิต เพื่อเป็นการแสดงว่าสิ้นสุดข้อมูลแล้ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 2.10 รูปแบบอย่างง่ายที่สุดของข้อมูลอนุกรมแบบอะซิงโครนัส

อุปกรณ์พิเศษที่ได้รับการออกแบบมาสำหรับการรับและส่ง และ สื่อสารข้อมูลแบบอะซิงโครนัสเรียกว่า Universal Asynchronous Receiver/Transmitter หรือ UART อัตราความเร็วในการรับและส่งข้อมูลของการรับส่งข้อมูลแบบอะซิงโครนัสคือ ค่าบอดเรต ซึ่งก็คือค่าจำนวนบิตต่อวินาทีที่ใช้ในการรับและส่งข้อมูล บอดเรตมาตรฐานที่ใช้สำหรับพอร์ตอนุกรม RS-232 ได้แก่ 110, 150, 300, 600, 1200, 2400, 4800, 9600, และ 19200 บิตต่อวินาที และมีค่าเพิ่มมากขึ้นตามเทคโนโลยีของคอมพิวเตอร์ ซึ่งการรับส่งแบบอนุกรม โดยไม่ผ่าน โมเด็มอาจจะสามารถกำหนดค่าบอดเรตได้สูงถึง 115200บิตต่อวินาทีเนื่องจากบอดเรตคือจำนวนบิตของข้อมูลที่สามารถ ถ่ายทอดได้ภายใน 1 วินาที ยกตัวอย่าง ข้อมูลอนุกรมถูกส่งในลักษณะ 8 บิต ไม่มีการตรวจสอบ พริตต์ มีบิตเริ่มต้น 1 บิต และบิตปิดท้าย 1 บิต ความยาวของข้อมูลที่รับส่งนี้เท่ากับ 10 บิต ถ้าใช้บอดเรตในการส่งข้อมูลเท่ากับ 9600 บิตต่อวินาที ก็จะสามารถรับส่งข้อมูลได้ด้วยความเร็ว 960 ไบต์ต่อวินาที และถ้ามีการใช้พริตต์ความเร็วในการรับส่งข้อมูลจะเหลือเป็น 872 ไบต์ต่อวินาที

### 2.3.2 มาตรฐานพอร์ตอนุกรมแบบ RS-232

มาตรฐานการเชื่อมต่อแบบอนุกรม RS-232 เป็นมาตรฐานอุตสาหกรรมที่ออกแบบมาเพื่อใช้ในการส่งข้อมูลอนุกรมแบบอะซิงโครนัส 2 ทิศทาง โดยมาตรฐาน RS-232 ในอดีตนั้นถูกออกแบบมาเพื่อการส่งผ่านข้อมูลจากคอมพิวเตอร์ไปยัง โมเด็มเพียงอย่างเดียวเพื่อที่จะนำข้อมูลจากโมเด็มนี้สื่อสารผ่านสายโทรศัพท์ไปยังคอมพิวเตอร์อีกชุดซึ่งอยู่ห่างไกลกัน โดยคณะกรรมการที่เรียกว่า สมาคมอุตสาหกรรมอิเล็กทรอนิกส์ (Electronic Industries Association: EIA) ได้วางมาตรฐานที่มีชื่อเรียกว่า EIA RS-232 มาตรฐานนี้ในช่วงแรกจะใช้คอนเน็คเตอร์เป็นแบบ DB-25 โดยกำหนดความยาวสูงสุดของสายสัญญาณไว้ที่ 50 ฟุต มีระดับสัญญาณตั้งแต่ -3 ถึง -12V แสดงว่ามีข้อมูล (Mark) และ +3 ถึง +12V แสดงว่าเป็นช่องว่าง (Space)

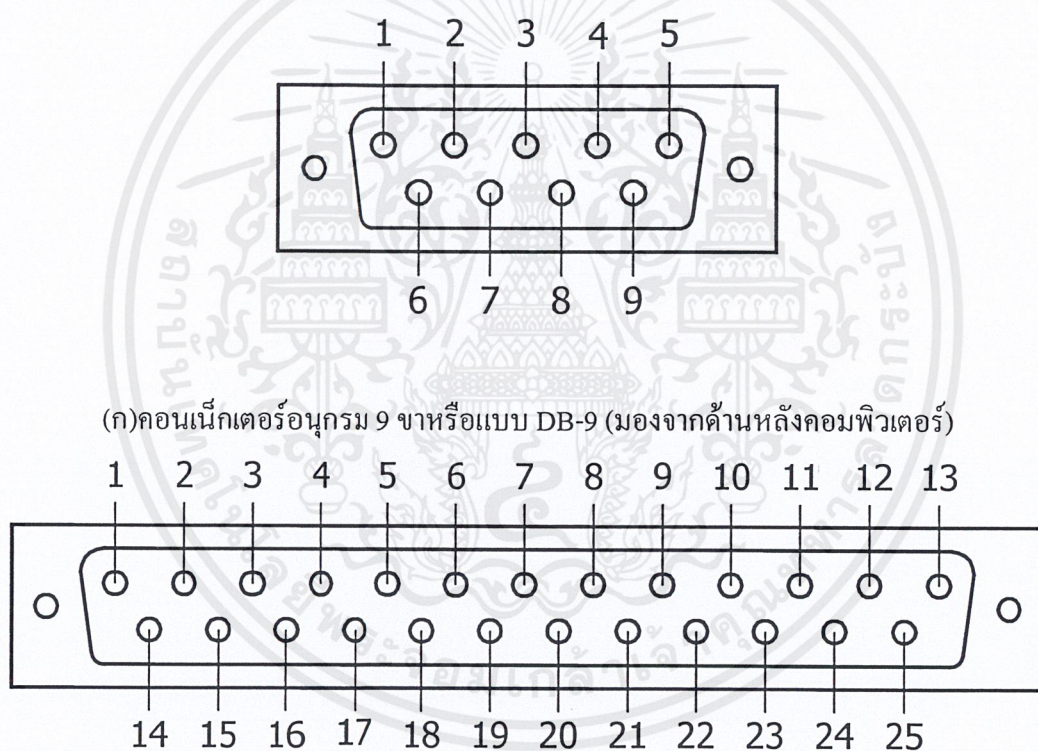
มาตรฐาน RS-232 ได้กำหนดรูปแบบของอุปกรณ์ที่เชื่อมต่อข้อมูล (Data Terminal Equipment: DTE) กับวงจรข้อมูลปลายทาง (Data Circuit Terminating: DCE) ไว้ว่า อุปกรณ์ DTE จะต้องเป็นอุปกรณ์ที่มีการประมวลผลในตัวเช่น ไมโครคอนโทรลเลอร์หรือไมโครคอมพิวเตอร์ ซึ่งมีความสามารถในการสร้างบิตข้อมูลแบบอนุกรมได้ ส่วนอุปกรณ์ DCE จะทำหน้าที่เป็นเพียง

ตัวรับข้อมูลที่ส่งมาจาก DTE เท่านั้น โดยการรับส่งข้อมูลระหว่างอุปกรณ์ทั้งสองจะกระทำผ่านมาตรฐาน RS-232

สำหรับการใช้งานบนคอมพิวเตอร์ พอร์ตอนุกรม RS-232 มักถูกใช้เชื่อมต่อกับโมเด็มหรือเมาส์ โดยสามารถรับส่งข้อมูลได้ที่ความยาวของสายสัญญาณสูงสุดถึง 20 เมตร

### 2.3.3 คอนเน็กเตอร์สำหรับพอร์ต RS-232 และการเชื่อมต่อ

มาตรฐานการเชื่อมต่อแบบ RS-232 จะใช้คอนเน็กเตอร์แบบ DB-25 ตัวผู้หรือ DB-9 ตัวผู้ ซึ่งคอนเน็กเตอร์แบบ DB-25 จะมีขาต่อใช้งานเพียง 9 เส้นเช่นเดียวกับคอนเน็กเตอร์แบบ DB-9 เนื่องจากขาอื่น ๆ ที่เคยใช้งานในอดีต ปัจจุบันมีการใช้งานไม่มากนัก จึงถูกยกเลิกไป โดยแสดงรูปร่างและตำแหน่งขาในภาพที่ 2.11



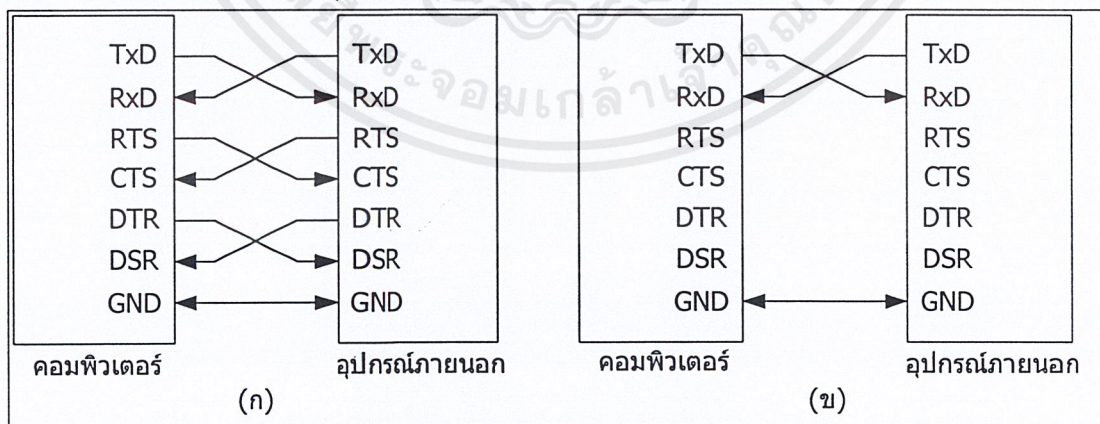
(ข) คอนเน็กเตอร์อนุกรม 25 ขาหรือแบบ DB-25 (มองจากด้านหลังคอมพิวเตอร์)

ภาพที่ 2.11 การจัดขาของคอนเน็กเตอร์พอร์ตอนุกรมตามมาตรฐาน RS-232 ทั้งแบบ DB-9 และ DB-25

ตารางที่ 2.1 การจัดขาของคอนเน็กเตอร์พอร์ตอนุกรมตามมาตรฐานRS-232 ทั้งแบบ DB-9 และ DB-25

คอนเน็กเตอร์	คอนเน็กเตอร์	ชื่อของสายสัญญาณ	ทิศของสายสัญญาณ
1	8	Data Carrier Detect: DCD	อินพุต
2	3	Received Data: RxD	อินพุต
3	2	Transmitted Data: TxD	เอาต์พุต
4	20	Data Terminal Ready: DTR	เอาต์พุต
5	7	Signal Ground: GND	-
6	6	Data Set Ready: DSR	อินพุต
7	4	Request To Send: RTS	เอาต์พุต
8	5	Clear To Send: CTS	อินพุต
9	22	Ring Indicator: RI	อินพุต

สำหรับการเชื่อมต่อคอมพิวเตอร์กับอุปกรณ์ภายนอกแสดงดังในภาพที่ 1-4 ลูกศรในรูปแสดงถึงทิศทางของข้อมูล ในภาพที่ 1-4 (ก) เป็นการเชื่อมต่อแบบ Null modem หรือการเชื่อมต่อโดยตรงโดยไม่ต้องผ่านโมเด็ม โดยมีการตรวจสอบหรือแฮนด์เช็กเต็มรูปแบบ ส่วนในภาพที่ 1-4 (ข) เป็นการเชื่อมต่อแบบ Null modem ในลักษณะที่ใช้สายสัญญาณเพียง 3 เส้น โดยเส้นหนึ่งสำหรับส่งข้อมูล อีกเส้นสำหรับรับข้อมูล และเส้นสุดท้ายเป็นกราวด์ สำหรับรายละเอียดหน้าที่การทำงานในและขาของพอร์ตอนุกรม RS-232 มีดังนี้



ภาพที่ 2.12 การต่ออุปกรณ์ภายนอกกับพอร์ตอนุกรมของคอมพิวเตอร์ในลักษณะต่างๆ

(ก) การต่ออุปกรณ์ภายนอกเข้ากับคอมพิวเตอร์แบบ Null modem

(ข) การต่ออุปกรณ์ภายนอกเข้ากับคอมพิวเตอร์แบบ RS-232 โดยใช้สายสัญญาณเพียง 3 เส้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- Data Carrier Detect: DCD หรืออาจเรียกว่า Carrier Detect:: CD ขานี้จะแอกติฟเมื่อมีการส่งสัญญาณพาห้จากอุปกรณ์สื่อสารข้อมูล เช่น โมเด็ม สำหรับการใช้งานปกติ นี้จะไม่ได้ถูกใช้งานมากนัก
- Receive Data: RD หรือ RxD ขานี้ใช้เพื่อรับสัญญาณอนุกรมเข้ามายังคอมพิวเตอร์ โดยนำข้อมูลที่อ่านได้เก็บไว้ในรีจิสเตอร์ บัฟเฟอร์
- Transmitted Data: TD หรือ TxD ขานี้ใช้เพื่อส่งข้อมูลออกจากคอมพิวเตอร์ โดยนำข้อมูลที่เก็บอยู่ในบัฟเฟอร์สำหรับส่งข้อมูลส่งออกไป
- Data Terminal Ready: DTR เป็นขาสัญญาณที่ส่งออกจากคอมพิวเตอร์เพื่อให้อุปกรณ์ปลายทางรับรู้ว่า ต้องการติดต่อด้วย โดยขา DTR นี้จะต้องเชื่อมต่อกับขา DSR ของอุปกรณ์ปลายทาง และขา DTR ของอุปกรณ์ปลายทางจะต้องเชื่อมต่อกับขา DSR ของคอมพิวเตอร์ ถ้าใช้การเชื่อมต่อเป็นแบบ Null Modem ซึ่งใช้สายในการเชื่อมต่อเพียง 3 เส้น จะต้องต่อขา DTR และ DSR ของตัวมันเองเข้าด้วยกัน และต้องต่อกับขา DCD ด้วยในกรณีที่ใช้โปรแกรมสื่อสารที่ใช้มีการตรวจจับสัญญาณพาห้
- Signal Ground: GND ขากราวด์ของระบบ
- Data Set Ready: DSR ขานี้จะใช้คู่กับขา DTR เพื่อตรวจสอบการเชื่อมต่อกันระหว่างคอมพิวเตอร์กับอุปกรณ์ปลายทาง ซึ่ง DSR นี้จะเป็นขาสำหรับรับข้อมูลจากภายนอกซึ่งถูกส่งมาจากขา DTR
- Request To Send: RTS เป็นขาสำหรับส่งสัญญาณร้องขอให้ทางอุปกรณ์ปลายทางส่งข้อมูลกลับมายังคอมพิวเตอร์ โดยขาที่รับสัญญาณ RTS ก็คือขา CTS ในกรณีที่ใช้การเชื่อมต่อ Null modem 3 สาย จะต้องเชื่อมต่อกับขา RTS และ CTS ของตัวมันเองเข้าด้วยกัน เพื่อจะให้การรับและส่งข้อมูลสามารถเกิดขึ้นได้ตลอดเวลา
- Clear To Send: CTS ขานี้จะคอยรับสัญญาณจากขา RTS เมื่อรับสัญญาณได้ข้อมูลที่ขา TxD จะถูกส่งออกไป ดังนั้นขานี้จึงถูกใช้เพื่อตรวจสอบอุปกรณ์ต่อพ่วงว่าพร้อมที่จะรับข้อมูลหรือไม่
- Ring Indicator: RI ใช้แสดงสถานะสัญญาณเรียกจากสายโทรศัพท์ ปกติในการสื่อสาร โดยทั่วไปขานี้จะไม่ถูกใช้งาน จะใช้งานก็ต่อเมื่อมีการเชื่อมต่อกับโมเด็มและโปรแกรมมีการตรวจสอบสัญญาณนี้เท่านั้น

### 2.3.4 UART

UART มาจากคำว่า Universal Asynchronous Receiver Transmitter ซึ่งหมายถึง อุปกรณ์ที่ทำหน้าที่รับและส่งข้อมูลแบบอะซิงโครนัสนั่นเอง สำหรับการสื่อสารอนุกรมบน คอมพิวเตอร์แล้ว UART ถือว่าเป็นหัวใจสำคัญของการสื่อสารอนุกรม

หน้าที่หลักของ UART คือทำหน้าที่แปลงข้อมูลที่อยู่ในรูปแบบขนานจาก คอมพิวเตอร์ให้อยู่ในรูปแบบอนุกรมแบบอะซิงโครนัส แล้วส่งออกไป และทำหน้าที่แปลง สัญญาณอนุกรมแบบ อะซิงโครนัสที่ป้อนเข้ามายัง UART ให้เป็นแบบขนานก่อนที่จะส่งเข้าสู่ คอมพิวเตอร์ ซึ่งนอกจาก UART จะส่งข้อมูลไปยังคอมพิวเตอร์แล้ว ยังแจ้งข้อมูลอื่นๆ ให้ คอมพิวเตอร์รับทราบด้วย เช่น อัตราเร็วในการรับส่งข้อมูล (ผิดพลาดจากพาริตี, เฟรมข้อมูล, โอเวอร์รัน) เป็นต้น

ภายใน UART จะมีส่วนของวงจรสร้างบอดเรตแบบโปรแกรมได้ (programmable buadrate generator) โดยการกำหนดค่าตัวหารให้กับสัญญาณนาฬิกาของ UART โดยตัวหารนี้มี ขนาด 16 บิต ดังนั้นจึงสามารถกำหนดตัวหารอยู่ในช่วง 1- 56,535 UART สามารถรับส่งข้อมูลได้ ทั้งแบบฮาล์ฟดูเพล็กซ์ (half duplex) และฟูลดูเพล็กซ์ (full duplex) โดยการส่งแบบฮาล์ฟดูเพล็กซ์ เป็นการส่งแบบทิศทางเดียว ส่วนการส่งแบบฟูลดูเพล็กซ์นั้นสามารถรับและส่งข้อมูลได้ในคราว เดียวกัน

ในเครื่องคอมพิวเตอร์ทั่วไปมี UART ที่ใช้งานกันอยู่ 2 เบอร์คือ 8250 ซึ่งเป็น UART มาตรฐานที่มีใช้กันมานาน UART เบอร์นี้จะมีบัฟเฟอร์สำหรับรับและส่งข้อมูลตำแหน่ง เดียวกัน ทำให้การรับและส่งข้อมูลถูกจำกัดความเร็วอยู่ที่ 57.6 กิโลบิตต่อวินาทีเท่านั้น แต่ UART เบอร์นี้ก็ถือว่าเป็นต้นแบบของ UART ที่ใช้ในคอมพิวเตอร์โดยคอมพิวเตอร์ทุกๆ รุ่นจะต้อง สนับสนุนการทำงานตามรูปแบบของ UART เบอร์นี้

UART อีกเบอร์หนึ่งคือ 16450 มีความสามารถรับส่งข้อมูลได้ที่ความเร็ว 115,200 บิตต่อวินาที และเพิ่มรีจิสเตอร์สำหรับพักข้อมูลสำหรับ UART นอกจากนั้นยังเพิ่มส่วนของชิพดีรี จิสเตอร์แบบ FIFO (First In First Out) ขนาด 16 ไบต์เข้าไปทำให้สามารถสนับสนุนความเร็วใน การรับส่งข้อมูลของที่ 26 กิโลบิตต่อวินาทีได้ โดยคอมพิวเตอร์ในปัจจุบันใช้ UART เบอร์นี้หรือ ใหม่กว่า เช่น เบอร์ TL16C750 ซึ่งมีรีจิสเตอร์แบบ FIFO ขนาด 64 ไบต์ทำงานได้ที่ระดับแรงดัน +5V และ +3V มีโหมดประหยัดพลังงาน สามารถรับส่งข้อมูลได้ที่ความเร็ว 1 เมกะบิตต่อวินาทีเมื่อ ใช้สัญญาณ นาฬิกา 16 MHz

อย่างไรก็ตาม ความเร็วในการส่งข้อมูลที่มากมายของ UART เบอร์ใหม่ๆ ก็ไม่ได้ช่วยให้ การรับส่งข้อมูลของคอมพิวเตอร์เร็วขึ้น เนื่องจากว่าคอมพิวเตอร์ยังใช้ความถี่ของสัญญาณนาฬิกา ในการแปลงข้อมูลเพียง 1.8432 MHz เท่านั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 2.3.5 รีจิสเตอร์ของพอร์ตอนุกรม RS-232

เครื่องคอมพิวเตอร์โดยทั่วไปสามารถต่อพอร์ตอนุกรม RS-232 สูงสุดได้ 4 พอร์ต ซึ่งจะมีชื่อเรียกเป็น COM1, COM2, COM3, และ COM4 ซึ่งพอร์ตอนุกรมแต่ละตัวต่างก็ใช้งาน UART ภายในคอมพิวเตอร์ในการติดต่อกับอุปกรณ์ภายนอกเช่นเดียวกัน

ในภาพที่ 1-5 แสดงไดอะแกรมการทำงานภายในของพอร์ตอนุกรมซึ่งประกอบไปด้วยรีจิสเตอร์ขนาด 8 บิต 8 ตัวที่ใช้งานร่วมกับ UART แอแดปเตอร์ของรีจิสเตอร์ภายในพอร์ตอนุกรม สามารถคำนวณได้จากค่ารีจิสเตอร์พื้นฐานของพอร์ตอนุกรม ยกตัวอย่าง พอร์ตอนุกรม COM1 มีแอแดปเตอร์อยู่ที่ 3F8H ตำแหน่งของรีจิสเตอร์ต่างๆ จะเป็นตำแหน่งที่บวกเข้าไปกับค่า 3F8H โดยรีจิสเตอร์ที่ใช้งานกับพอร์ตอนุกรมมีดังนี้

00H เป็นรีจิสเตอร์บัฟเฟอร์สำหรับเก็บข้อมูลรับเข้ามาหรือเตรียมข้อมูลก่อนที่จะส่งออกไป

01H รีจิสเตอร์อีนานาเบิ้ลการอินเตอร์รัปต์ ใช้ในการเซตโหมดการอินเตอร์รัปต์ของพอร์ตอนุกรม

02H รีจิสเตอร์แสดงโหมดการอินเตอร์รัปต์ ใช้เพื่อตรวจสอบโหมดการอินเตอร์รัปต์เมื่อมีการอินเตอร์รัปต์เกิดขึ้น

03H รีจิสเตอร์กำหนดรูปแบบของข้อมูล

04H รีจิสเตอร์ควบคุมโมเด็ม ใช้ตรวจสอบบิตสำหรับติดต่อกับโมเด็ม เช่น RTS หรือ DTR

05H รีจิสเตอร์แสดงสถานะการรับและการส่งข้อมูลแบบอนุกรม

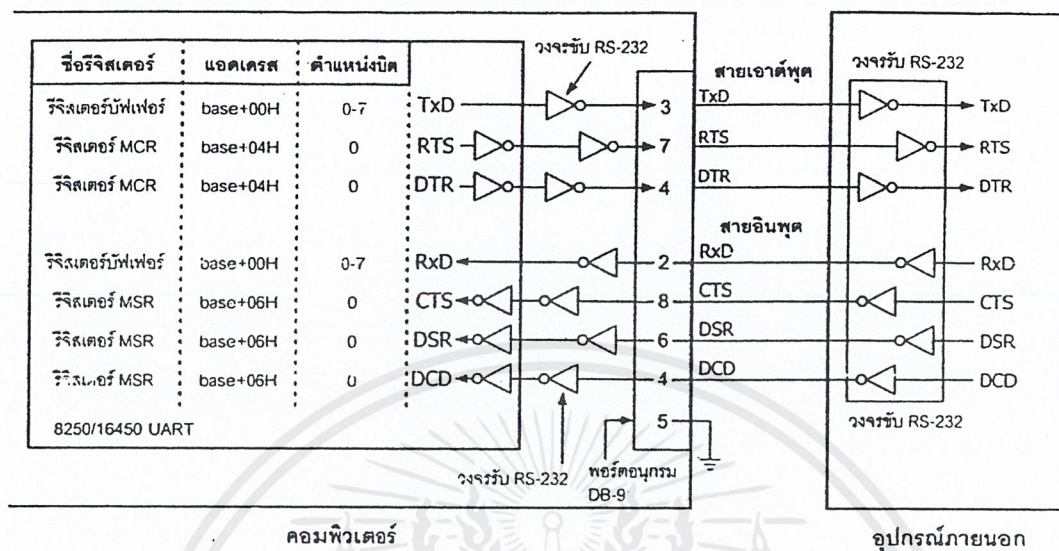
06H รีจิสเตอร์แสดงสถานะของโมเด็ม ซึ่งจะแสดงสถานะของขา DCD, Ri, DSR และ CTS

07H รีจิสเตอร์สำหรับการเก็บข้อมูลชั่วคราว

ลักษณะสัญญาณอินพุตและเอาต์พุตของพอร์ต RS-232

สัญญาณเอาต์พุตที่ใช้ควบคุม (RTS และ DTR) และสัญญาณแสดงสถานะอินพุต (CTS, DSR และ DCD) ของพอร์ตอนุกรม RS-232 จะถูกกลับสถานะภายในตัว UART ส่วนสัญญาณข้อมูลทั้งภาคส่งและรับจะไม่ถูกกลับสถานะ UART จะให้ระดับสัญญาณเอาต์พุตออกมาเป็นแบบ ทีทีแอล เท่านั้น ดังนั้นเมื่อสัญญาณถูกส่งออกมาจาก UART จึงต้องส่งเข้าสู่วงจรขับเพื่อปรับระดับแรงดันให้ได้ระดับสัญญาณเป็นไปตามมาตรฐาน RS-232 ก่อนส่งออกไปจากคอมพิวเตอร์สำหรับอุปกรณ์ต่อเชื่อมปลายทางก็จะต้องมีวงจรขับในลักษณะนี้เช่นเดียวกัน เพื่อให้ได้ระดับสัญญาณในระดับเดียวกัน แต่วงจรขับที่ใช้ทั้งภายในคอมพิวเตอร์และอุปกรณ์ต่อเชื่อมปลายทางนั้นจะถูกกลับสถานะ ดังแสดงเป็นบล็อกไดอะแกรมในภาพที่ 2.13

### 2.3.6 แอดเดรสของพอร์ตอนุกรม



ภาพที่ 2.13 ไคอะแกรมแสดงโครงสร้างทางฮาร์ดแวร์ของพอร์ตอนุกรม

แอดเดรสพื้นฐานของพอร์ตอนุกรมมี 4 ตำแหน่งดังนี้คือ

COM1: 3F8H

COM2: 2F8H

COM3: 3E8H

COM4: 2E8H

เมื่อเริ่มเปิดเครื่องเพื่อใช้งานคอมพิวเตอร์ ไบออสภายในคอมพิวเตอร์จะทำการตรวจสอบแอดเดรสของพอร์ตอนุกรมทั้งหมด ถ้าไบออสตรวจพบแอดเดรสของพอร์ตอนุกรม ไบออสจะนำแอดเดรสที่ตรวจพบไปเก็บไว้ในหน่วยความจำขนาด 2 ไบต์ สำหรับพอร์ตอนุกรม COM1 จะเก็บไว้ที่แอดเดรส 0000:0400H และ 0000:0401H ส่วนตำแหน่งอื่นๆ มีรายละเอียดดังนี้

COM2 = 0000:0402H – 0000:0403H

COM3 = 0000:0404H – 0000:0405H

COM4 = 0000:0406H – 0000:0407H

นอกจากนี้ที่หน่วยความจำแอดเดรส 0000:0411H ยังใช้สำหรับแสดงจำนวนของพอร์ตอนุกรมที่มีอยู่ในคอมพิวเตอร์อีกด้วย โดยมีรายละเอียดดังแสดงในตารางที่ 2.2

## ตารางที่ 2.2 แสดงข้อมูลในแอดเดรส 0000:0411H ที่ใช้แจ้งจำนวนพอร์ตอนุกรม

บิต 3	บิต 2	บิต 1	จำนวนพอร์ต
0	0	0	ไม่มีพอร์ตอนุกรม
0	0	1	มีพอร์ตอนุกรม 1 พอร์ต
0	1	0	มีพอร์ตอนุกรม 2 พอร์ต
0	1	1	มีพอร์ตอนุกรม 3 พอร์ต
1	0	0	มีพอร์ตอนุกรม 4 พอร์ต

### 2.3.7 ระดับแรงดันที่ใช้งานสำหรับพอร์ตอนุกรม RS-232

มาตรฐานการสื่อสารข้อมูลของพอร์ตอนุกรม ได้ระบุช่วงระดับแรงดันสำหรับการทำงานของพอร์ตอนุกรมไว้ว่า ที่ลอจิก “0” จะมีระดับสัญญาณ +3 ถึง +15V ส่วนลอจิก “1” จะมีระดับสัญญาณ -3 ถึง -15V ระดับสัญญาณนี้ทำให้ไม่สามารถที่จะนำเอาตัวพุดใดๆ ต่อเข้ากับลอจิกเกตเพื่อใช้งาน ได้โดยตรง จะต้องผ่านวงจรเพื่อเปลี่ยนระดับแรงดันเสียก่อน โดยปกติจะใช้ไอซีจำพวก RS-232 transceiver ที่นิยมมากคือ MAX232 หรือ ICL232 ไปซีในกลุ่มนี้จะทำหน้าที่แปลงระดับแรงดันของ RS-232 ให้อยู่นะดับ ทีทีแอล โดยลอจิก “0” ซึ่งเดิมมีระดับสัญญาณ +3 ถึง +15V จะถูกแปลงเป็น 0V ส่วนลอจิก “1” ซึ่งมีระดับสัญญาณ -3 ถึง -15V จะแปลงเป็น +5V ทั้งนี้เพื่อให้สามารถเชื่อมต่อกับอุปกรณ์ดิจิทัลอื่นที่ใช้ระดับแรงดัน ทีทีแอล ได้

## 2.4 การเขียนโปรแกรมเพื่อใช้งานพอร์ตอนุกรม

### 2.4.1 การกำหนดค่าเริ่มต้นให้กับพอร์ตอนุกรม

ก่อนการใช้งานพอร์ตอนุกรมนั้นจะต้องมีการกำหนดค่าเริ่มต้นให้กันตัวมันก่อน ซึ่งก็คือการกำหนดจำนวนบิตข้อมูลที่ต้องการจะส่ง,จำนวนอนปิดท้าย,ชนิดของพาราริตี้ที่ใช้ และบอดเรต

การกำหนดสามารถทำได้หลายวิธี วิธีแรกเป็นการกำหนดคาคออสพร้อมพ์ โดยใช้คำสั่ง MOIDE ซึ่งมีวิธีในการใช้งานดังนี้

Mode Comm: baud=b, parity=p, data=d, stop=s, retry=r

หรือ

Mode Comm: b, p, d, s, r

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตัวอย่าง MODE COM1 : 9600 , n , 8 , 1 จะเป็นการกำหนดให้พอร์ตอนุกรม COM1 มีบอดเรตเท่ากับ 9600 บิตต่อวินาที ไม่มีการตรวจสอบพาริตี รับส่งข้อมูลแบบ 8 บิต และมีบิตติดท้าย 1 บิต

วิธีที่ 2 เป็นการกำหนดโดยใช้อินเตอร์รัปต์ของคอส ตำแหน่งที่ 14H ซึ่งการใช้งานจะต้องกำหนดค่าต่างๆลงในรีจิสเตอร์ด้วย โดยจะต้องกำหนดให้รีจิสเตอร์ AH มีค่าเท่ากับ 0 รีจิสเตอร์ DX เก็บค่าของพอร์ตอนุกรมที่ต้องการกำหนดค่าเริ่มต้น โดย

DX=0 จะกำหนดให้พอร์ตอนุกรม COM1

DX=1 จะกำหนดให้พอร์ตอนุกรม COM2

DX=2 จะกำหนดให้พอร์ตอนุกรม COM3

DX=3 จะกำหนดให้พอร์ตอนุกรม COM4

รีจิสเตอร์ AL ซึ่งมีขนาด 8 บิตใช้เก็บค่าเริ่มต้นต่างๆมีรายละเอียดดังนี้

BD2	BD1	BD0	PAR1	PAR0	STOP	DA1	DA0
-----	-----	-----	------	------	------	-----	-----

BD2,BD1,BD0 ใช้สำหรับกำหนดค่าบอดเรต

“111” บอดเรตเท่ากับ 9,600 บิตต่อวินาที

“110” บอดเรตเท่ากับ 4,800 บิตต่อวินาที

“101” บอดเรตเท่ากับ 2,400 บิตต่อวินาที

“100” บอดเรตเท่ากับ 1,200 บิตต่อวินาที

“011” บอดเรตเท่ากับ 600 บิตต่อวินาที

“010” บอดเรตเท่ากับ 300 บิตต่อวินาที

“001” บอดเรตเท่ากับ 150 บิตต่อวินาที

“000” บอดเรตเท่ากับ 110 บิตต่อวินาที

PAR1,PAR0 ใช้กำหนดค่าพาริตีโดย

“00” หรือ “10” ไม่มีการตรวจสอบพาริตี

“01” พาริตีคู่

“11” พาริตีคี่

STOP ใช้กำหนดจำนวนของบิตปิดท้าย

“1” มีบิตปิดท้ายเท่ากับ 2 บิต

“0” มีบิตปิดท้ายเท่ากับ 1 บิต

DA1,DA0 ใช้กำหนดความยาวของข้อมูลโดย

“10” ความยาวข้อมูลเท่ากับ 7 บิต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับบริการเชิงงานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## “11” ความยาวข้อมูลเท่ากับ 8 บิต

### 2.4.2 การรับและส่งข้อมูลแบบอนุกรม

มีหลากหลายวิธีในการรับและส่งข้อมูลแบบอนุกรมผ่านพอร์ต RS-232 เช่น ใช้คำสั่งพิมพ์ออกทางเครื่องพิมพ์ เรียกอินเทอร์รับต์ของไบออสหรือของคอส การเขียนหรืออ่านไปยังแอดเดรสของพอร์ตโดยตรง วิธีสุดท้ายเป็นวิธีที่มีความยืดหยุ่นในการใช้งานที่สุด ยกตัวอย่างถ้าต้องการส่งข้อมูลไปยังพอร์ตอนุกรม COM1 สามารถเขียนข้อมูลโดยตรงไปที่รีจิสเตอร์บัฟเฟอร์สำหรับส่งข้อมูล (แอดเดรส 3F8H) โดยใช้คำสั่งภาษา QBASIC ง่ายๆดังนี้

```
OUT &H3F8,X
```

ค่า X ในที่นี้หมายถึงข้อมูลที่ต้องการส่ง มีขนาด 8 บิต

สำหรับการอ่านค่าข้อมูลจากพอร์ตอนุกรม จะเป็นการอ่านข้อมูลมาจากรีจิสเตอร์บัฟเฟอร์สำหรับรับข้อมูล (แอดเดรส 3F8H เช่นเดียวกัน) ซึ่งสามารถเขียนโปรแกรมง่ายๆได้ดังนี้

```
Y=INP[&3F8]=X
```

แต่เมื่อใช้คำสั่งนี้ในขณะที่โปรแกรมทำงานผ่านระบบปฏิบัติการวินโดวส์ จะไม่สามารถใช้งานได้เนื่องจากระบบปฏิบัติการวินโดวส์ได้เข้าฝั่งตัวพอร์ตอนุกรมเข้าเป็นส่วนหนึ่งของระบบปฏิบัติการ ดังนั้นการใช้งานต้องเรียกผ่านเครื่องมือที่ติดต่อผ่านระบบปฏิบัติการ เช่น MSCOMM32.OCX ของโปรแกรม Visual Basic

### 2.4.3 คอนโทรล MSCOMM

สำหรับการใช้งาน Visual Basic ตั้งแต่เวอร์ชัน 2 เป็นต้นมา ใน Visual Basic จะมีคัสตอมคอนโทรลสำหรับการสื่อสารอนุกรมผ่านทางพอร์ตอนุกรมของคอมพิวเตอร์ของคอมพิวเตอร์มาให้ โดยใน Visual Basic เวอร์ชัน 3 จะชื่อว่า MSCOMM.VBX ส่วนเวอร์ชัน 4 ใช้ชื่อว่า MSCOMM

16.OCX สำหรับทำงานกับระบบปฏิบัติการ 16 บิตและ MSCOMM32.OCX สำหรับระบบปฏิบัติการ 32 บิต สำหรับ Visual Basic เวอร์ชันต่อมาจะมีเพียง MSCOMM32.OCX เท่านั้น

MSCOMM จัดเตรียมทางเลือกเอาไว้ 2 ทาง เพื่อความสะดวกในการสื่อสารข้อมูลทางแรกคือ การสื่อสารข้อมูลที่กระตุ้นด้วยเหตุการณ์ (event-driven communication) เป็นรูปแบบการใช้งานที่มีประสิทธิภาพมากสำหรับการตอบสนองแบบทันทีทันใด เช่นเมื่อตัวอักษรถูกส่งมาที่พอร์ตอนุกรมหรือเกิดการเปลี่ยนแปลงที่ขา Data Carrier Detect(DCD) หรือขา Request To Send (RTS) เหตุการณ์ Oncomm ของ MSCOMM จะสามารถตรวจจับได้ทันที ส่วนทางเลือกที่สองเป็นการคอยตรวจสอบค่าเหตุการณ์และความผิดพลาดที่เกิดขึ้นด้วยการดูค่าที่เปลี่ยน

แปลงภายในคุณสมบัติ Commevent หลังจากให้โปรแกรมทำงานในฟังก์ชันต่างๆไปเรียบร้อยแล้ว ซึ่งวิธีนี้ใช้งานได้ดีในกรณีที่โปรแกรมมีขนาดเล็ก

คอนโทรล MSCOMM 1 ตัวสามารถควบคุมการทำงานของเออร์ตอนุกรได้ 1 พอร์ต ถ้าในโปรแกรมที่ใช้งานต้องการติดต่อกับพอร์ตอนุกรมในแต่ละพอร์ต แอดเดรสใน พอร์ตอนุกรม และแอดเดรสของการอินเทอร์รัปต์สามารถเปลี่ยนแปลงได้จากการแก้ไขค่าที่ Control Panel

ถึงแม้ว่า คอนโทรล MSCOMM จะมีคุณสมบัติมากมายแต่จะสามารถทำความเข้าใจได้ไม่ยากดังนี้

#### 2.4.3.1 CommPort

ใช้ในการกำหนดและอ่านค่าพอร์ตอนุกรมที่ติดต่อกันอยู่ (COM1, COM2, COM3, COM4) รูปแบบการใช้งาน

Object.Settings[=Value]

ค่า Value เป็นค่าของพอร์ตอนุกรม ชนิดของข้อมูลเป็น Integer ค่า Value สามารถกำหนดได้ในช่วง 1-16 (ค่าเริ่มต้นกำหนดไว้ที่ 1) เมื่อมีการกำหนดค่าแล้วทำการเปิดพอร์ตอนุกรมโดยใช้คุณสมบัติ PortOpen แต่ว่าพอร์ตนั้นไม่มีอยู่ในระบบ MSCOMM จะสร้างสัญญาณแสดงข้อผิดพลาด error 68 ขึ้นมา ซึ่งหมายถึงอุปกรณ์ตัวนี้ไม่มีอยู่ในระบบ ดังนั้นการเขียนโปรแกรมจึงจำเป็นต้องกำหนดตำแหน่งของพอร์ตอนุกรมก่อนที่ใช้คำสั่ง OpenPort

#### 2.4.3.2 Setting

ใช้ในการกำหนดและอ่านค่าอัตราบอดเรต, พาริตี, จำนวนบิตข้อมูล, จำนวนของบิตปิดท้าย

รูปแบบการใช้งาน

Object.settings[=value]

ค่า value นี้มีชนิดข้อมูลเป็นแบบ string มีรูปแบบเป็น “BBBB,P,D,S” โดย BBBB เป็นค่าอัตราบอดเรต, P เป็นค่าพาริตี, D เป็นจำนวนของข้อมูลบิตข้อมูล และ S เป็นจำนวนของบิตปิดท้าย ปกติแล้วค่านี้ถูกกำหนดไว้เป็น “9600,N,8,1”

ค่าบอดเรตมาตรฐานที่ใช้กับ MSCOMM มีดังนี้

110 บิตต่อวินาที

300 บิตต่อวินาที

600 บิตต่อวินาที

1,200 บิตต่อวินาที

2,400 บิตต่อวินาที

9,600 บิตต่อวินาที

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

14,400 บิตต่อวินาที

19,200 บิตต่อวินาที

28,800 บิตต่อวินาที

38,400 บิตต่อวินาที

56,000 บิตต่อวินาที

128,000 บิตต่อวินาที

256,000 บิตต่อวินาที

สำหรับค่ามาตรฐานในการกำหนดค่าพาริตีมีดังนี้

สัญลักษณ์      รายละเอียด

E                      พาริตีคู่

M                      ลอจิก "1"

N                      ไม่ใช่

O                      พาริตีคี่

S                      ลอจิก "0"

ค่าที่ใช้ในการกำหนดจำนวนบิตมี 5 ค่าคือ 4, 5, 6, 7 และ 8 (เป็นค่าปกติ)

ค่าที่ระบุจำนวนบิตปิดท้ายมี 3 ค่า คือ 1 (เป็นค่าปกติ), 1.5 และ 2

ตัวอย่างการใช้งานคำสั่ง Settings โดยจะเป็นการกำหนดค่าอัตราเท่ากับ 9600 ไม่มีพาริตี จำนวนข้อมูล 8 บิต และบิตปิดท้าย 1 บิต สามารถเขียนโปรแกรมได้ดังนี้

```
MSCOMM1.settings="9600,N,8,1"
```

สาเหตุที่ค่าที่กำหนดจะต้องอยู่ภายในเครื่องหมายคำพูด เนื่องจากค่าที่กำหนดอยู่ในรูปตัวแปร String

#### 2.4.3.3 PortOpen

ใช้ในการกำหนดและอ่านค่าสถานะของพอร์ตอนุกรม เพื่อเปิดและปิดพอร์ตอนุกรม

รูปแบบการใช้งาน

```
Object.portopen[=value]
```

ค่า value นี้มีชนิดของข้อมูลเป็นแบบบูลีนคือมี True กับ False โดย True จะหมายถึงพอร์ตได้ถูกเปิดออกแล้ว และ False หมายถึงการปิดพอร์ตอนุกรมสำหรับการปิดพอร์ตนั้นจะมีการเคลียร์บัฟเฟอร์รับข้อมูลและบัฟเฟอร์ส่งข้อมูลด้วย คอนโทรล MSCOMM จะปิดพอร์ตอนุกรมโดยอัตโนมัติเมื่อออกจากโปรแกรม ก่อนที่จะใช้คุณสมบัติ PortOpen ต้องตรวจสอบให้แน่ใจก่อนว่าคุณสมบัติ Commport นั้นได้ทำการกำหนดตำแหน่งของพอร์ตอนุกรมไว้ถูก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ต้องหรือไม่ มิเช่นนั้นแล้ว MSCOMM จะแสดงข้อผิดพลาด error 68 แจ้งแก่ผู้ใช้งาน หรือถ้าพอร์ตอนุกรมนั้นถูกเปิดเอาไว้แล้ว โปรแกรมก็จะแจ้งข้อผิดพลาดออกมาเช่นเดียวกัน

ถ้าคุณสมบัติ DTREnable หรือ RTSEnable ถูกกำหนดให้เป็น True ก่อนที่จะทำการเปิดพอร์ต ค่าคุณสมบัตินี้ของ DTREnable หรือ RTSEnable จะถูกเซตเป็น False หลังจากเปิดพอร์ตแต่ถ้าเซตเป็น False หลังจากปิดโปรแกรมแล้วค่าที่กำหนดไว้จะเป็นค่าเดิม

ตัวอย่างการใช้คำสั่งเปิดพอร์ต เพื่อติดต่อสื่อสารกับพอร์ตอนุกรม COM1 และมีบอดเรต 9,600 บิตต่อวินาที ไม่มีพาริตี จำนวนบิตข้อมูล 8 บิตและบิตปิดท้าย 1 บิตมีดังนี้

```
Mscomm.settings= "9600,n,8,1"
```

```
Mscomm1.comport = 1
```

```
Mscomm1.portopen = true
```

#### 2.4.3.4 Input

อ่านค่าและลบค่าขบวนข้อมูลจากบัฟเฟอร์ภาครับ

รูปแบบการใช้งาน

```
Object.input
```

คุณสมบัติ InputLen เป็นตัวกำหนดจำนวนของตัวอักษรที่จะอ่านโดยคุณสมบัติ Input การกำหนดค่าให้ InputLen เท่ากับ 0 เป็นการกำหนดให้คุณสมบัติ Input ทำการอ่านค่าข้อมูลในบัฟเฟอร์รับข้อมูลทั้งหมด

คุณสมบัติ InputMode เป็นตัวกำหนดชนิดของข้อมูลที่คุณสมบัติ Input รับเข้ามา ถ้า InputLen ถูกกำหนดเป็น comInputModeText คุณสมบัติ Input จะส่งค่าข้อมูลกลับมาในรูปแบบของข้อความชนิดข้อมูลเป็นแบบ Variant ถ้า comInputModeBinary คุณสมบัติ Input จะส่งข้อมูลกลับมาในรูปแบบของไบนารีและชนิดข้อมูลเห็นแบบ Variant

ตัวอย่างโปรแกรมแสดงให้เห็นถึงวิธีการในการรับข้อมูลจากบัฟเฟอร์รับข้อมูลทั้งหมด

```
Private Sub Command_click()
```

```
Dim Instring as String
```

```
    Mscomm1.inputlen=0
```

```
    If Mscomm1.inbuffercount then
```

```
        Instring=Mscomm1.input
```

```
    End if
```

```
End sub
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

#### 2.4.3.5 InBufferCount

ส่งค่าจำนวนของตัวอักษรที่อยู่ในบัฟเฟอร์ภาครับ

รูปแบบการใช้งานคำสั่ง

Object.inbuffercount[=value]

คำสั่ง InBufferCount จะแสดงค่าจำนวนของตัวอักษร ซึ่งรับมาจากภายนอกและยังเก็บอยู่ในบัฟเฟอร์ของภาครับ เพื่อให้ผู้ใช้งานอ่านค่าออกไป สำหรับการเคลียร์ค่าบัฟเฟอร์ภาครับทำได้โดยกำหนดให้ InBufferCount มีค่าเป็น 0

หมายเหตุ อย่าสับสนระหว่างคำสั่ง InBuffersize และ Inbuffercount คำสั่ง InBuffersize นั้นใช้เพื่อกำหนดขนาดของบัฟเฟอร์

#### 2.4.3.6 InBufferSize

กำหนดและคืนค่าขนาดของบัฟเฟอร์ภาครับในหน่วยเป็นไบต์

รูปแบบการใช้งานคำสั่ง

Object.inbuffersize[=value]

คำสั่ง Inbuffersize ใช้เพื่อกำหนดขนาดของบัฟเฟอร์ภาครับ โดยค่าเริ่มต้นถูกกำหนดไว้ที่ 1,024 ไบต์

หมายเหตุ การกำหนดค่าบัฟเฟอร์ภาครับขนาดใหญ่จะทำให้หน่วยความจำที่เหลือสำหรับการใช้งานส่วนอื่นๆจะเหลือน้อย อย่างไรก็ตามการกำหนดค่าบัฟเฟอร์ภาครับที่น้อยเกินไปจะทำให้เกิดการโอเวอร์โฟลวหรือข้อมูลล้นบัฟเฟอร์ เว้นแต่จะมีการใช้แฮนด์เช็ก ดังนั้นค่าปานกลางที่เหมาะสมก็คือค่า 1,024 ซึ่งเป็นค่าเริ่มต้นนั่นเอง แต่ถ้าโปรแกรมมีการเกิดโอเวอร์โฟลวแล้วจึค่อยปรับเพิ่มค่าขนาดของบัฟเฟอร์ให้มีความมากขึ้น

#### 2.4.3.7 InputLen

กำหนดค่าและคืนค่าจำนวนขอตัวอักษรที่อ่านจากบัฟเฟอร์ภาครับ

รูปแบบการใช้งานคำสั่ง

Object. InputLen[=value]

ค่าเริ่มต้นของคุณสมบัติ InputLen มีค่าเท่ากับ “0” การกำหนดค่าเท่ากับ “0” จะทำให้คำสั่ง Input ของ MSCOMM อ่านค่าข้อมูลที่อยู่ในบัฟเฟอร์ภาครับทั้งหมด

ถ้าไม่มีข้อมูลอยู่ในบัฟเฟอร์ภาครับมากเท่ากับจำนวน InputLen คำสั่ง Input จะส่งค่าว่างกลับออกมา ผู้ใช้งานสามารถตรวจสอบข้อมูลอยู่ในบัฟเฟอร์ภาครับก่อนแล้วจึงค่อยอ่านข้อมูลจากบัฟเฟอร์ภาครับ

คุณสมบัตินี้มักใช้กันการอ่านค่าข้อมูลจากเครื่องมือหรือเครื่องจักรที่มีการกำหนดค่าขนาดความยาวของข้อมูลเอาไว้แล้ว

ตัวอย่าง โปรแกรมการอ่านค่าตัวอักษรออกมา 10 ตัวอักษร

```

Private Command_click()
Dim Commdata as string
Mscomm1.inputlen=10
Commdata = mscomm1 .input
End sub

```

#### 2.4.3.8 InputMode

กำหนดค่าและคืนค่าชนิดของข้อมูลที่ได้รับ โดยคำสั่ง Input  
รูปแบบการใช้งานคำสั่ง

```
Object.inputmode[=value]
```

คุณสมบัติ InputMode ใช้กำหนดว่าข้อมูลชนิดไหนที่รับเข้ามาผ่านคำสั่ง

Input โดยข้อมูลจะเลือกได้ 2 ประเภทคือ

ComInputModeText สำหรับข้อมูลที่อยู่ในรูปข้อความตัวอักษรตามมาตรฐาน  
ANSI โดยจะต้องกำหนดค่าเป็น "0" และค่าเริ่มต้นของการรับค่าข้อมูลก็จะเป็นค่านี้

ComInputModeBinary สำหรับข้อมูลอื่นๆ ซึ่งจะเก็บอยู่ในรูปไบนารีรวมกันอยู่  
เป็นไบนารีข้อมูล

ตัวอย่างการใช้งาน InputMode ต่อไปนี้จะทำการอ่านค่าข้อมูล 10 ไบนารีจากพอร์ต  
อนุกรมและเก็บข้อมูลไว้ในตัวแปรแบบอาร์เรย์ ชนิดข้อมูลเป็นแบบไบนารี

```

Private sub
Dim Buffer as Variant
Dim Arr() as Byte
MsComm1.CommPort = 1
MsComm1.PortOpen = True
MSComm1.InputMode = comInputModeBinary
Do Until MSComm1.InBufferCount < 10
Input buffer
Doevent
Loop
Buffer =MSComm1.Input
Arr = Buffer
End Sub

```

### 2.4.3.9 Output

ใช้ในการส่งขบวนข้อมูลไปยังบัพเฟอร์ส่งข้อมูล

รูปแบบการใช้งาน

Object.output[=value]

ค่า value เป็นค่าของตัวอักษรที่เขียนไปยังบัพเฟอร์ส่งข้อมูล คุณสมบัติ

Output สามารถใช้ในการส่งข้อมูลตัวอักษรหรือข้อมูลไบนารีก็ได้ โดยการส่งข้อมูลเป็นรูปแบบตัวอักษรจะต้องกำหนดข้อมูลตัวอักษรเป็นแบบ Variant และมีข้อมูลภายในเป็นแบบ String สำหรับการส่งข้อมูลไบนารี จะต้องกำหนดชนิดของข้อมูลเป็นแบบ Variant และข้อมูลภายในแบบ Byte

ตัวอย่าง โปรแกรมการส่งค่าที่ป้อนจากคีย์บอร์ดไปยังพอร์ตอนุกรม โดยใช้

คุณสมบัติ Output

```
Dim Buffer as Variant
MSComm1.Commport = 1
MSComm1.portOpen = True
Buffer = chr$(KeyASCII)
MSComm1.Output = Buffer
```

End Sub

## 2.5 คำสั่งที่ใช้ติดต่อกับ Motion Control

ตัวโมชันคอนโทรลที่เรานำมาใช้งานนี้เป็นผลงานที่ ผศ. ประภาส อุดคภิมาพันธุ์ วิจัยสร้างขึ้นมา ใช้ตัวไมโครคอนโทรลเลอร์ของ Philips รุ่น 89C51 RD+ เป็นตัวประมวลผลซึ่งตัวโมชันคอนโทรลนี้ใช้การเชื่อมต่อผ่านพอร์ตอนุกรมโดยใช้รหัส ASCII ในการส่งข้อมูลไปยังตัวโมชันคอนโทรลสามารถควบคุมการทำงานของมอเตอร์ได้สองแกนและการทำงานจะเป็นแบบเพิ่มพูน(Increment) โดยมีคำสั่งในการใช้งานดังนี้

- คำสั่ง H เป็นคำสั่งที่ป้อนเพื่อจตุรูปแบบการใช้งานคำสั่งต่างๆที่ต้องใช้กับตัวโมชันคอนโทรล เมื่อป้อนคำสั่งก็จะแสดงผลออกมาดังนี้ ตามตารางที่ 2.3

### ตารางที่ 2.3 แสดงผลของโมชันคอนโทรลเมื่อป้อนคำสั่ง H

Mechatronic laboratory division. Department of Instrumentation Engineering. Faculty of Engineering King Mongkuit Institute of Technology Ladkrabang. *** MOTION CONTROLLER MENU CHARACTER COMMAND**** This program is a simple of two and half axis Motion controller operating on the Philips 89C51 RD+ CPU. State of result of the motion signal on Port 1 (bit 10 to bit 17)		
Command	Syntax	Function
Read	R	report motion parameter.
Down pen	D	move pen downward.
Up pen	U	move pen upward.
Total para	T y z s a	initial set motion parameter.
Input buff	I cr..percen	input program replay.
Motion	M	replay motion command in buffer.
Go	G	execute motion or start of motion.
Pen	P u1 u2 u3 d1 d2 d3	defind pen parameter.
F	F s a	defind rapid speed in case of pen up.
Datum	K k1 k2 k3	k1 k2 k3 defind work datum.
Work datum	W	goto works datum.
Remove datum	Q	open datum break point.
Open datum	O	open datum break point.
Fix datum	C	fix datum break point.

- คำสั่ง T เป็นคำสั่งป้อนค่าพิคัดและอัตราความเร็วที่ใช้ในการเคลื่อนที่

การป้อนคำสั่งมีรูปแบบดังนี้ [T y z f a]

โดยที่ค่า y คือการหมุนของมอเตอร์ A

ค่า z คือการหมุนของมอเตอร์ B

ค่า f คืออัตราความเร็วในการเคลื่อนที่มีหน่วยเป็น step / clock ของโมชัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ค่า  $a$  คือค่าของความเร่ง โดยค่าความเร่งที่แท้จริงจะเป็นส่วนกลับกับ  $a$  ตัวอย่าง จะเป็นการใช้งานเพื่อให้มอเตอร์ A เคลื่อนที่ตามเข็มไป 2500 สเต็ป มอเตอร์ B เคลื่อนที่ไปตามเข็ม 2500 สเต็ป มีความเร็วเท่ากับ 10,000 สเต็ปต่อ 1 คล็อกของตัวโมชันคอนโทรล มีความเร่งเท่ากับ 500

T 2500 -2500 10000 500

หมายเหตุ เครื่องหมายของ  $z$  ตีกลับเพราะมอเตอร์ B ถูกติดตั้งกลับด้าน

- คำสั่ง G เป็นคำสั่งที่ป้อนให้โมชันคอนโทรลเพื่อเริ่มการเคลื่อนที่ไปตามที่กำหนดไว้
- คำสั่ง R เป็นคำสั่งที่สั่งให้ตัวโมชันรายงานค่าพารามิเตอร์ต่างๆ แสดงผลออกมา

```
position_r =3535
brake_dist=10
ar =1000
vr =100000
Z=-2500 Y=-2500 Speeds=100000 Accel=100 >
u1=0 u2=0 u3=0
d1=0 d2=0 d3=0
k1=0 k2=0 k3=0
f1=100000 f2=100
Feeds speed=0
Feeds acceleration=0
Pen status now pen up
Pen reference without Works Datum
```

- คำสั่ง F เป็นคำสั่งในกรณีที่เราต้องการเปลี่ยนแปลงเฉพาะค่าความเร็วหรือ ความเร่งเท่านั้น

รูปแบบคำสั่งจะเป็น F f a

เช่น F 10000 100

- คำสั่ง I เป็นคำสั่งเริ่มต้นที่ใช้ป้อนชุดของคำสั่งเข้าไปในบัพเฟอร์ของตัวโมชันคอนโทรล และจบชุดคำสั่งด้วยเครื่องหมาย “%” ตัวอย่าง เป็นการป้อนชุดคำสั่งให้มอเตอร์ทั้งสองตัวเคลื่อนที่ไปแล้วกลับมายังตำแหน่งเดิม มีจุดมุ่งหมายเพื่อทำการทดสอบความแม่นยำของตัวแขนกล หากใช้การป้อนทีละคำสั่งย่อมเป็นการลำบาก

I  
 F 20000 200  
 T 2500 0  
 G  
 T 0 2500  
 G  
 T -2500 0  
 G  
 T 0 -2500  
 G  
 %

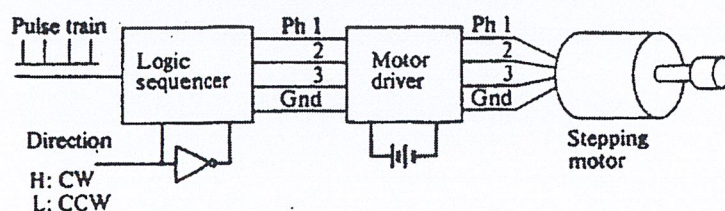
## 2.6 สเต็ปป์มอเตอร์

สเต็ปป์มอเตอร์ แบ่งออกได้เป็นหลายชนิดด้วยกัน เช่น สเต็ปป์มอเตอร์แบบแปรค่ารีลักแตนซ์ (Variable-reluctance), สเต็ปป์มอเตอร์แบบแม่เหล็กถาวร (Permanent magnet), สเต็ปป์มอเตอร์แบบไฮบริด (Hybrid) และสเต็ปป์มอเตอร์แบบลิเนียร์ (Linear) เป็นต้น แต่ที่ใช้งานกันบ่อยส่วนใหญ่เป็นแบบสเต็ปป์มอเตอร์แบบแปรค่ารีลักแตนซ์ และ , สเต็ปป์มอเตอร์แบบไฮบริด

สเต็ปป์มอเตอร์มีความแตกต่างจากมอเตอร์โดยทั่วไป คือเมื่อป้อนไฟฟ้าให้กับมันๆ จะหมุนเพียงเล็กน้อยตามเส้นรอบวง ซึ่งแตกต่างจากมอเตอร์โดยทั่วไป ที่จะหมุน โดยทันที และตลอดเวลาที่จ่ายไฟฟ้า, สเต็ปป์มอเตอร์ยังสามารถกำหนดตำแหน่งการหมุนด้วยตัวเลขได้อย่างละเอียด โดยการใช้อินเตอร์เฟสเป็นตัวกำหนดและ จับเก็บตัวเลขเหล่านั้นไว้

### 2.6.1 หลักการทำงานของสเต็ปป์มอเตอร์

หลักการทำงานของสเต็ปป์มอเตอร์ แสดงได้ดังภาพที่ 1 ซึ่งประกอบด้วยวงจรรีเลย์ วงจรขับสัญญาณความถี่อินพุท และสัญญาณควบคุมทิศทางการหมุนคอนโทรลสเต็ปเปอร์ของสเต็ปป์มอเตอร์จะอินเตอร์เฟสได้โดยตรงกับไมโครโปรเซสเซอร์ หรือ ไมโครคอมพิวเตอร์



ภาพที่ 2.14 แสดงคอนโทรลเลอร์ สำหรับควบคุมการทำงานของสเต็ปป์มอเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 2.6.2 โครงสร้างและหลักการพื้นฐานของสเต็ปิ่งมอเตอร์

โครงสร้างสเต็ปิ่งมอเตอร์ โดยทั่วไปการที่จะทำให้เกิดการหมุนของโรเตอร์ (Rotor) ได้ต้องมีการกระทำของสนามแม่เหล็กที่เกิดขึ้นระหว่าง โรเตอร์และสเตเตอร์ (Stator) ซึ่งขึ้นอยู่กับการจัดวางขั้วแม่เหล็ก (Pole) โครงสร้างของขั้วแม่เหล็กบนสเตเตอร์ ประกอบขึ้นจากแท่นเหล็กวงแหวนที่มีซี่ยื่นออกมา แต่ละซี่เหล่านั้นจะมีคอยล์พันสวมอยู่ ดังนั้น เมื่อป้อนกระแสไฟฟ้าผ่านคอยล์ทำให้เกิดสนามแม่เหล็กไฟฟ้า (Electromagnetic) ขึ้นด้านตรงข้ามของแต่ละขั้วแม่เหล็กจะได้รับกระแสไฟฟ้าในขณะเดียวกัน แต่ว่าจะไหลวนในทิศทางตรงข้ามทำให้เกิดสนามแม่เหล็กไฟฟ้าในทิศทางตรงกันข้ามขึ้นดังภาพที่ 2.2 (ก)



ภาพที่ 2.15 (ก) แสดงสเต็ปิ่งมอเตอร์ที่มีการต่อวงจรขดลวดภายในเพื่อกระตุ้นให้เกิดขั้วแม่เหล็กขึ้น 1 ขั้วในทิศทางตรงกันข้าม ส่วนขดลวดอื่นๆ จะไม่ถูกกระตุ้น

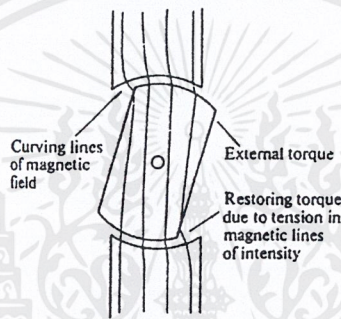
(ข) แสดงการต่อวงจรขดลวดแบบกระตุ้นให้เกิดขั้วแม่เหล็กพร้อมกัน 2 ขั้วที่อยู่ใกล้กัน ทำให้โรเตอร์เคลื่อนที่มาหยุดหมุนอยู่ระหว่างขั้วแม่เหล็กทั้งสอง การเพิ่มจำนวนสเต็ปสามารถทำได้โดยไม่ต้องเพิ่มจำนวนของโครงสร้างภายในได้ โดยการจ่ายกำลังไฟฟ้าไปยังขั้วแม่เหล็ก 2 ขั้วที่อยู่ใกล้กันในเวลาเดียวกัน ซึ่งจะทำให้โรเตอร์หยุดหมุนอยู่ระหว่างกลางของ 2 ขั้วแม่เหล็กนั้น และวิธีการนี้ยังช่วยให้เกิดแรงบิด (Torque) มากขึ้นด้วยดังแสดงในภาพที่ 2.2 (ข) การพันขดลวดหรือคอยล์บนสเต็ปิ่งมอเตอร์มีอยู่ 2 วิธีคือ แบบไดโพลาร์ (Dipolar) และแบบยูนิโพลาร์ (Unipolar) ดังแสดงในรูป 2.3 แบบไบโพลาร์มีการพันขดลวด 1 ขดบนแต่ละขั้วของแม่เหล็กของสเตเตอร์ ขั้วแม่เหล็กที่เกิดขึ้นบนสเตเตอร์จะถูกกำหนดโดยทิศทางของกระแสไฟฟ้า และสามารถทำให้เกิดขั้วแม่เหล็กในทิศทางตรงกันข้ามได้โดยการกลับทิศทางกระแสไฟฟ้า แบบยูนิโพลาร์จะมีการพันขดลวด 2 ขด บนแต่ละขั้วแม่เหล็กของสเตเตอร์ ซึ่งแต่ละขดจะทำให้เกิดขั้วแม่เหล็กในทิศทางตรงกันข้ามกัน การกลับขั้วแม่เหล็กเปลี่ยนไปมาทำได้โดยการสวิตซ์กระแสไฟฟ้าจากขดลวดหนึ่ง ไปยังอีกขดลวดหนึ่งแทนเท่านั้น โดยปกติของขดลวดทั้งสองจะมีการ เชื่อมต่อกัน หรือมีจุกร่วมเพื่อลดจำนวนของสายไฟที่ต่อจากมอเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 2.16 แสดงวงจรจ่ายกำลังไฟฟ้าให้กับสตีปีงมอเตอร์ทั้งสองแบบ

- (ก) สำหรับชนิดยูนิโพลาร์ ซึ่งใช้ทรานซิสเตอร์สวิตช์เพียงตัวเดียวต่อ 1 คอยล์  
 (ข) สำหรับชนิดไบโพลาร์ ซึ่งใช้ทรานซิสเตอร์สวิตช์ 4 ตัวต่อ 1 คอยล์



ภาพที่ 2.17 แสดงแรงดึงดูด ทำให้เกิดทอร์กที่หมุน โรเตอร์ให้ไปอยู่ในตำแหน่งที่สมดุล  
 จากภาพที่ 2.17 แสดงรูปแท่งแม่เหล็กถาวรติดอยู่บนเพลลา และสามารถหมุนได้อย่างอิสระในขณะที่  
 ไม่มี ขั้วไฟฟ้าที่สเตเตอร์ ถ้าก่อนมีขั้วไฟฟ้า ตำแหน่งของ โรเตอร์แม่เหล็กอยู่ตามแกน a-a' ซึ่งทำมุม  
 0 องศา กับแกนของขั้วแม่เหล็กไฟฟ้า

ดังนั้นจะได้แรงแม่เหล็กที่เกิดจากการดึงดูดของขั้วแม่เหล็กที่ต่างกันทำให้เกิดส่วนของแรง  
 บิตปกติ

$$F_n = F \cos \alpha$$

จะได้ทอร์กรวม

$$T = F_n R$$

(ทำให้โรเตอร์หมุนไปในทิศทาง CW จนกว่าแกนของโรเตอร์ a-a' จะอยู่ในแนวเดียวกับแกนของ  
 สเตเตอร์)

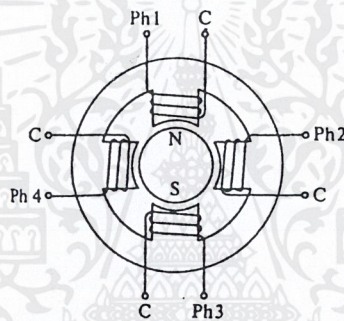
ถ้าหากมีขั้วแม่เหล็กไฟฟ้าหลายๆขั้ว รอบๆสเตเตอร์และถ้าหากขั้วเหล่านั้นถูกกระตุ้นด้วย  
 กระแสพัลส์ ในรูปแบบที่เรียงลำดับกันไป โรเตอร์ก็จะหมุนในรูปลักษณะของสตีปีที่เป็นไปตาม  
 การหมุนของสนามแม่เหล็กที่เกิดจากการสวิตช์ที่เรียงลำดับของขดลวดแม่เหล็กไฟฟ้าของสเตเตอร์

### 2.6.3 สเต็ปป์มอเตอร์ชนิดต่างๆ

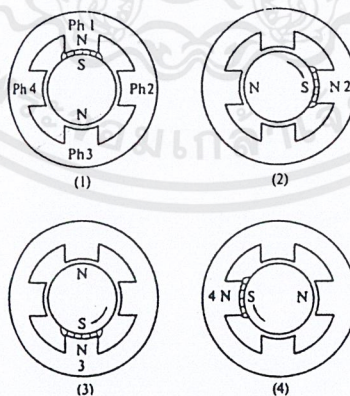
#### 1) สเต็ปป์มอเตอร์แบบแม่เหล็กถาวร

สเต็ปป์มอเตอร์ชนิดนี้ใช้แม่เหล็กถาวรเป็น โรเตอร์ และมีซี่ฟันของสเตเตอร์ล้อมรอบ ซี่ฟันของสเตเตอร์ถูกพันด้วยขดลวดสำหรับสร้างสนามแม่เหล็ก เมื่อต้องการให้สเต็ปป์มอเตอร์แบบแม่เหล็กถาวรมีขนาดมุมสเต็ปเล็กลงจะต้องเพิ่มจำนวนขั้วแม่เหล็กของโรเตอร์และจำนวนซี่ฟันของสเตเตอร์แต่ก็มีขีดจำกัดในการเพิ่มจำนวนขั้วแม่เหล็กของโรเตอร์ เนื่องจากการสร้างแม่เหล็กถาวรให้มีโครงสร้างแบบมีขั้วแม่เหล็กหลายขั้วทำได้ยาก

สมมติว่าสเต็ปป์มอเตอร์แบบแม่เหล็กถาวรขนาดสี่เฟสมีโรเตอร์เป็นแม่เหล็กถาวรทรงกระบอกและสเตเตอร์มีซี่ฟันรอบๆพันด้วยขดลวด มีรูปแบบพื้นฐานการทำงาน คือ เมื่อสร้างสัญญาณกระตุ้นตามลำดับเฟสโรเตอร์ก็จะหันไปตามทิศทางของการกระตุ้นดังแสดงในภาพที่ 2.18



ภาพที่ 2.18 แสดงโครงสร้างของสเต็ปป์มอเตอร์แบบแม่เหล็กถาวร



ภาพที่ 2.19 แสดงการทำงานในสเต็ปป์มอเตอร์แบบแม่เหล็กถาวรขนาดสี่เฟส

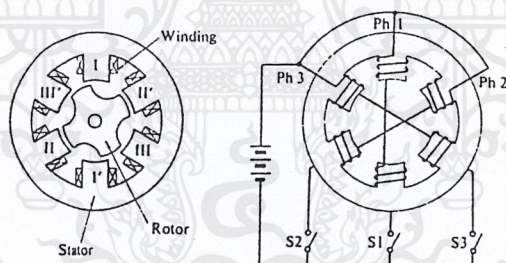
ข้อเสียของสเต็ปป์มอเตอร์แบบแม่เหล็กถาวรคือมีขนาดมุมสเต็ปใหญ่ทำให้มีความละเอียดของสเต็ปต่อรอบน้อยเนื่องจากโครงสร้างของโรเตอร์เป็นแม่เหล็กถาวร การสร้างแม่เหล็กถาวรให้มีหลายขั้วนั้นทำได้ยาก ทำให้ไม่สามารถสร้างสเต็ปป์มอเตอร์ขนาดเล็กได้ สเต็ปป์-เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

มอเตอร์แบบแม่เหล็กถาวรส่วนใหญ่จะมีโครงสร้างขนาดเล็กทำค่าทอร์กที่ได้ต่อหน่วยปริมาตรมีค่าต่ำ ถ้าต้องการปรับปรุงประสิทธิภาพในเรื่องของทอร์กแม่เหล็กถาวรที่ใช้ต้องทำจากสารแม่เหล็กที่มีสภาพความเป็นแม่เหล็กสูง สารแม่เหล็กราคาถูก เช่น เฟอร์ไรต์ มีสภาพความเป็นแม่เหล็กน้อยจึงให้ค่าทอร์กต่ำ

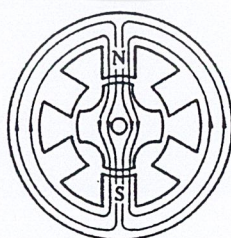
## 2) สเต็ปป์มอเตอร์แบบแปรค่ารีลักแตนซ์ได้

โครงสร้างของสเต็ปป์มอเตอร์แบบแปรค่ารีลักแตนซ์ได้มีโรเตอร์และสเตเตอร์ทำจากโลหะซิลิกอนซึ่งเป็นสารแม่เหล็กที่มีสภาพซึมซับได้ทางเหล็กสูงและยอมให้สนามแม่เหล็กจำนวนมากไหลผ่านได้

ในรูป 2.7 เป็นสเต็ปป์มอเตอร์แบบแปรค่ารีลักแตนซ์ได้ขนาดสามเฟส โดยโครงสร้างของโรเตอร์มี 4 ซี่ฟัน และ สเตเตอร์มี 6 ซี่ฟัน และแต่ละเฟสของสเตเตอร์มี 2 ซี่ฟันที่อยู่ตรงกับข้ามกันและอยู่ห่างกัน 180 องศาต่ออนุกรมกันอยู่ ทั้งสองซี่ฟันของเฟสหนึ่งของสเตเตอร์จะมีขั้วตรงกันข้ามกัน ดังนั้น จากซี่ฟัน I, II, III จะเป็นขั้วเหลือ และ I', I'1', I'1'1' จะเป็นขั้วได้ เมื่อมอเตอร์ได้รับการกระตุ้น เมื่อเฟสหนึ่งถูกกระตุ้นสนามแม่เหล็กจะเกิดขึ้นดังภาพที่ 2.20 ดังนั้น ถ้าโรเตอร์ I และ I' จะวางตัวอยู่ในแนวเดียวกันซึ่งมีค่ารีลักแตนซ์ที่น้อยที่สุดสถานะนี้เป็นสถานะที่จุดสมดุลเป็นจุดซึ่งเส้นแรงแม่เหล็กสั้นที่สุด แสดงดังในรูป 2.21



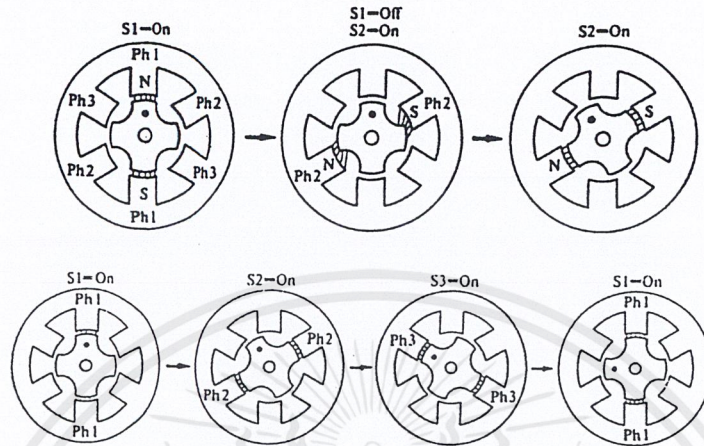
ภาพที่ 2.20 สเต็ปป์มอเตอร์แบบแปรค่ารีลักแตนซ์ได้



ภาพที่ 2.21 แสดงตำแหน่งสมดุลเมื่อเฟสใดเฟสหนึ่งของสเต็ปป์มอเตอร์ถูกกระตุ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากภาพที่ 2.8 สามารถเห็นว่าโรเตอร์และสเตเตอร์ไม่ได้อยู่ในแนวเดียวกัน เส้นแรงของเฟสที่ถูกกระตุ้นมีค่ารีลักแตนซ์มาก สเต็ปป์มอเตอร์แบบแปรค่ารีลักแตนซ์ได้จะทำตัวให้มีค่ารีลักแตนซ์ต่ำที่สุด โดยวางตัวอยู่ในแนวเดียวกันกับแนวของเส้นแรงแม่เหล็ก



ภาพที่ 2.22 แสดงการเปลี่ยนลำดับสเต็ปในการเคลื่อนที่ของสเต็ปป์มอเตอร์ได้

- (1) ช่องว่างอากาศระหว่างซี่ฟันของโรเตอร์และสเตเตอร์ต้องเล็กที่สุดเท่าที่จะเป็นไปได้ เพื่อให้ค่าของทอร์กมากขึ้น และมีความถูกต้องแม่นยำทางตำแหน่งดีขึ้น เนื่องจากที่ระดับแมกนีโตโมทีฟฟอร์ชระดับเดียวกันช่องอากาศขนาดเล็กจะให้ค่าฟลักซ์แม่เหล็กสูงกว่า ซึ่งมีผลทำให้มีค่าทอร์กสูงกว่า และทำให้มีความคลาดเคลื่อนในการเคลื่อนที่จากจุดสมดุลมีน้อยกว่าช่องอากาศขนาดใหญ่
- (2) สำหรับโครงสร้างที่ต้องการมุมสเต็ปขนาดเล็กความสัมพันธ์ระหว่างมุมสเต็ป ( $\theta_n$ ) มุมเฟส ( $m$ ) จำนวนซี่ฟันของโรเตอร์ ( $N_r$ ) และจำนวนสเต็ป ( $S$ ) หาได้จากสมการ

$$S = 360/\theta_n - mN_r$$

เพื่อจะลดขนาดมุมสเต็ปลง จำนวนซี่ฟันของโรเตอร์จะต้องเพิ่มขึ้น ซี่ฟันของสเตเตอร์จะต้องเพิ่มขึ้นด้วย โดยโครงสร้างแต่ละขั้วของเฟสใดๆ จะมีหลายซี่ฟัน แต่จำนวนซี่ฟันของสเตเตอร์ไม่ใช่ชองค์ประกอบโดยตรงที่จะกำหนดมุมสเต็ปของสเต็ปป์มอเตอร์

- (3) การสร้างสเต็ปป์มอเตอร์ให้มีโครงสร้างแบบหลายสเต็ปเพื่อประสิทธิภาพในเรื่องของทอร์กการสร้างสเต็ปป์มอเตอร์แบบแปรค่ารีลักแตนซ์ได้ให้มีโครงสร้างแบบหลายสเต็ปนี้แต่ละสเต็ปจะถูกกำหนดเป็นหนึ่งเฟส โดยโรเตอร์และสเตเตอร์จะมีซี่ฟันเหมือน

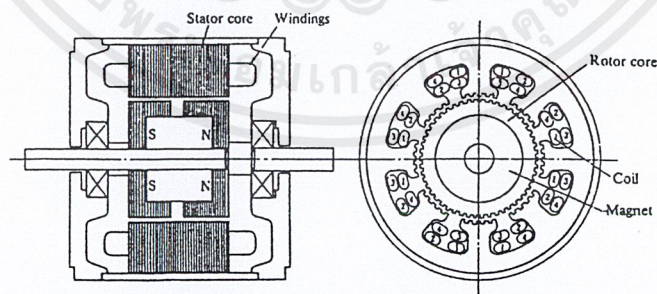
ระหว่างโรเตอร์และสเตเตอร์เพื่อช่วยเพิ่มทอร์กต่อหน่วยปริมาตรของโรเตอร์ ซึ่งเป็นการเพิ่มประสิทธิภาพของแมชชีน(machine)

### 3) สเต็ปป์มอเตอร์แบบไฮบริด

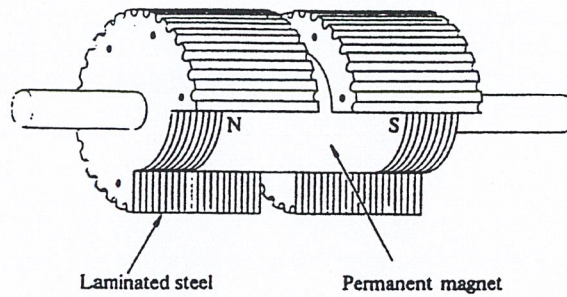
เป็นรูปแบบหนึ่งของสเต็ปป์มอเตอร์ซึ่งมีโรเตอร์เป็นแม่เหล็กถาวร ไฮบริดหมายถึงการทำงานร่วมกันของมอเตอร์แบบถาวร และมอเตอร์แบบแปรค่ารีลักแตนซ์ได้ สเต็ปป์มอเตอร์แบบไฮบริดมีโครงสร้างของสเตเตอร์คล้ายกับโครงสร้างของสเต็ปป์มอเตอร์แบบแปรค่ารีลักแตนซ์ได้ แต่การต่อขดลวดต่างกัน โดย หนึ่งในเฟสของสเต็ปป์มอเตอร์แบบแปรค่ารีลักแตนซ์ได้จะมีสองขดลวดของสองเฟสถูกพันอยู่ที่ขั้วเดียวกัน โดยที่ขั้วหนึ่งจะไม่ใช่ของเฟสใดเฟสหนึ่ง

คุณสมบัติที่สำคัญของสเต็ปป์มอเตอร์แบบไฮบริด คือ โครงสร้างของมอเตอร์มีแม่เหล็กถาวรอยู่ที่กึ่งกลางระหว่างเฟสทั้งสอง การเหนี่ยวนำสนามแม่เหล็กทำได้โดยใช้สนามแม่เหล็กซึ่งสร้างจากสเตเตอร์ซึ่งเป็นสนามแม่เหล็กแบบเฮเทอโพลาร์ (heteropolar field) ดังนั้นทอร์กเกิดจากการทำงานร่วมกันของสนามแม่เหล็กสองชนิดคือ สนามจากแม่เหล็กถาวร และสนามแม่เหล็กเหนี่ยวนำที่เกิดจากการกระตุ้นของขดลวดแต่ละขด โครงสร้างของขั้วฟันของสเตเตอร์จะใหญ่กว่าขั้วฟันของโรเตอร์เล็กน้อย เพื่อเพิ่มความถูกต้องแม่นยำทางตำแหน่งการเคลื่อนที่

หลักการการทำงานของสเต็ปป์มอเตอร์แบบไฮบริดที่แตกต่างจากสเต็ปป์มอเตอร์แบบแปรค่ารีลักแตนซ์ได้ คือแรงบิดที่เกิดจากสนามแม่เหล็กจะไม่ขึ้นอยู่กับกระแสที่ผ่านขดลวดอย่างเดียวแต่ขึ้นอยู่กับโครงสร้างของขั้วฟันด้วย ซึ่งขั้วฟันถูกออกแบบเพื่อให้ได้โครงสร้างขนาดเล็กและใช้แม่เหล็กถาวรเป็นแกนกลางเพื่อลดผลของการออสซิลเลตทางแมคคานิกส์



ภาพที่ 2.23 แสดงการเปลี่ยนลำดับสเต็ปในการเคลื่อนที่ของสเต็ปป์มอเตอร์ได้



ภาพที่ 2.24 แสดงหลักการทำงานของสเต็ปิ่งมอเตอร์แบบไฮบริด

ข้อดีของสเต็ปิ่งมอเตอร์แบบไฮบริด คือ มีขนาดสเต็ปขนาดเล็กมีความละเอียดของสเต็ปต่อรอบสูงมีค่าทอร์คสูงกว่าสเต็ปิ่งมอเตอร์แบบแปรค่ารีลักแตนซ์ได้ แต่สเต็ปิ่งมอเตอร์แบบแปรค่ารีลักแตนซ์ได้มีแรงเฉื่อยแบบแมคคานิกส์น้อยกว่าสเต็ปิ่งมอเตอร์แบบไฮบริด

นอกจากสเต็ปิ่งมอเตอร์ทั้งสามชนิดที่กล่าวมาแล้ว ยังมีสเต็ปิ่งมอเตอร์ชนิดอื่นๆที่ไม่ได้กล่าวถึงอีกเช่น ลิเนียร์สเต็ปิ่งมอเตอร์ ซึ่งเป็นมอเตอร์ที่ได้รับการออกแบบให้มีการเคลื่อนที่แบบเป็นเชิงเส้น อีเล็ก โตร ไฮดรอลิกสเต็ปิ่งมอเตอร์กำลังสูงสำหรับใช้ในอุตสาหกรรม เป็นต้น

#### 2.6.4 สัญลักษณ์ต่างๆ ของสเต็ปิ่งมอเตอร์

- $N_r$  = จำนวนซี่ฟันของ โรเตอร์
- $N_s$  = จำนวนซี่ฟันของสเตเตอร์
- $N_p$  = จำนวนเฟส
- $P_r$  = ความห่างระหว่างปลายซี่ฟัน โรเตอร์ (องศา)
- $P_s$  = ความห่างระหว่างปลายซี่ฟันของสเตเตอร์ (องศา)
- $\theta_n$  = มุมสเต็ป (องศา)
- $R_s$  = อัตราการสเต็ปหรือความเร็วในการสเต็ป (สเต็ป/รอบ)
- $X = N_r/N_p =$  จำนวนซี่ฟันของสเตเตอร์ต่อเฟส

#### 2.6.5 พารามิเตอร์ต่างๆ ของสเต็ปิ่งมอเตอร์

(1) ความห่างระหว่างปลายซี่ฟัน โรเตอร์และสเตเตอร์

$$P_r = 360/N_r \quad P_s = 360/N_s$$

(2) มุมสเต็ป (step angle)

เมื่อโรเตอร์เคลื่อนไหวในขนาดมุม  $P_r$  ได้เท่ากับ  $N_p$  สเต็ป ดังนั้นเราจะหามุมสเต็ปได้

$$\theta_n = P_r/N_p = 360/N_r N_p \quad \text{องศา/สเต็ป}$$

เอกสารนี้เป็นเอกสารลิขสิทธิ์สงวนลิขสิทธิ์หรือสงวนลิขสิทธิ์โดย บริษัท อีอีซี จำกัด (มหาชน) ขอสงวนสิทธิ์ในสิ่งที่ปรากฏและไม่รับผิดชอบต่อความเสียหายใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$\Theta_n = \left| P_r - P_s \right| \text{ องศา/สเต็ป}$$

(3) อัตราการสเต็ป (stepping rate)

ความเร็วในการสเต็ปต่อรอบ (360 องศา) หาได้เป็น

$$R_s = 360 / \Theta_n = N_r N_s \text{ สเต็ปต่อรอบ}$$

(4) ความเร็วของสเต็ปปิ้งมอเตอร์ (speed of step motor)

เมื่อเราป้อนอินพุตพัลส์ที่มีความถี่ (f) สเต็ปต่อพัลส์ให้กับสเต็ปปิ้งมอเตอร์จะสเต็ปไปด้วยความเร็ว (สเต็ป/พัลส์) \* f (พัลส์/วินาที)

$$1/R_x \text{ (รอบ/สเต็ป)} * (\text{พัลส์/วินาที}) (\text{สเต็ป/พัลส์}) * 60 \text{ (วินาที/นาทีย)}$$

$$\text{ความเร็วของมอเตอร์ } (\omega) = f/R_s = 60 * f / N_r N_p$$

$$\text{หรือ } X = R_s / N_p (N_p + 1) = N_r / (N_p + 1)$$

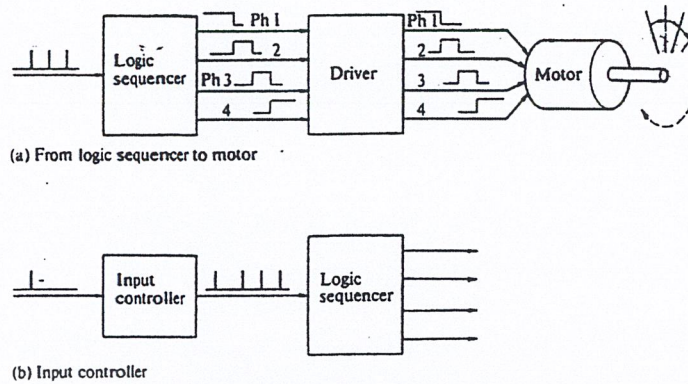
จำนวนโพลของสเตเตอร์ต่อเฟส (X) สัมพันธ์กับอัตราการสเต็ปหรือจำนวนซี่ฟันของโรเตอร์

## 2.6.6 การควบคุมการทำงานของสเต็ปปิ้งมอเตอร์

วงจรที่ใช้ควบคุมสเต็ปปิ้งมอเตอร์ส่วนใหญ่เป็นวงจรลอจิก และวงจรภาคกำลัง ซึ่งใช้สำหรับสเต็ปปิ้งมอเตอร์แบบแปรค่ารีลัคแตนซ์ดี สเต็ปปิ้งมอเตอร์แบบแม่เหล็กถาวรและสเต็ปปิ้งมอเตอร์แบบไฮบริดประสิทธิภาพของมอเตอร์ขึ้นอยู่กับตัวแปรต่อไปนี้ เช่น ไดนามิกส์ทอร์ก ความเร็ว การตอบสนองต่อความถี่ และการแดมป์ (damping) เป็นต้น นอกจากนี้ยังมีพารามิเตอร์บางอย่าง เช่น แรเสถียรภาพ และการสูญเสียกำลัง ก็ขึ้นอยู่กับวงจรขับเช่นกัน

รูปแบบการควบคุมสเต็ปปิ้งมอเตอร์โดยทั่วไป เป็นการควบคุมภาคกำลัง และลำดับการจ่ายพลังงานให้กับขดลวดในเฟสต่างๆ ของสเต็ปปิ้งมอเตอร์ วงจรกำลังจะทำงานในทิศทางเดียวหรือ 2 ทิศทาง ขึ้นอยู่กับวงจรจัดลำดับลอจิก ซึ่งใช้เป็นสัญญาณขับเสตของทรานซิสเตอร์กำลัง ซึ่งจะเพิ่มขนาดสัญญาณเพื่อจ่ายกระแสให้กับขดลวดของมอเตอร์และต้องใช้วงจรจำกัดกระแส เมื่อใช้การกระตุ้นแบบใช้แรงดันเกินเพื่อรักษาเสถียรภาพการทำงานของสเต็ปปิ้งมอเตอร์และใช้วงจรจำกัดแรงดันเกินเพื่อจำกัดสไปค์โวลท์เตจ (spike voltage) ให้อยู่ในระดับที่ปลอดภัยของการใช้งานทรานซิสเตอร์กำลัง

วงจรจัดลำดับลอจิก (sequencer logic circuit) จะมีการทำงานเป็นรลัลักษณะของวงจรมันแบบวงแหวนซึ่งมีขนาดเท่ากับจำนวนเฟสของสเต็ปปิ้งมอเตอร์ การใช้วงจรจัดลำดับลอจิกจะกระตุ้นวงจรขับได้ครั้งละครั้งเฟสเรียงตามลำดับกันไป



ภาพที่ 2.25 แสดงบล็อกไดอะแกรมของระบบควบคุมการขับเคลื่อนมอเตอร์

สัญญาณจากวงจรลำดับลจิกจะต้องถูกขยายให้เพียงพอที่จะจ่ายให้กับสเต็ปมอเตอร์ ดังนั้น จึงต้องป้อนสัญญาณนี้ให้กับวงจรทรานซิสเตอร์กำลังอีกทีหนึ่งเพื่อจะเพิ่มค่ากระแสให้เพียงพอที่จะจ่ายให้ขดลวดของสเต็ปมอเตอร์

### 2.6.7 การสั่งให้สเต็ปมอเตอร์หมุน

เราสามารถสั่งให้สเต็ปมอเตอร์หมุนได้ 3 วิธี คือ

2.1.7.1 แบบจ่ายกระแสไฟฟ้าให้เฟสเดียววนเวียนกันไปเรียก One-Excitation หรือ Half Drive คือ 0.1, 0.2, 0.3, 0.4 การ Excitation แบบนี้แรงบิดจะน้อย

2.1.7.2 แบบจ่ายกระแสไฟฟ้าให้พร้อมกันทีละ 2 เฟส เรียก Two-Excitation หรือ Full Step คือ 0102, 0203, 0304, 0401 หมุนเวียนกันไป

2.1.7.3 แบบจ่ายกระแสไฟฟ้าให้ทีละ 1 เฟส สลับกับ 2 เฟสเรียก One-Two Excitation หรือ Half Step เหมือนรูปแสดงมุมของโรเตอร์ในตารางที่ 2.4 แต่แบบนี้จำนวน Step จะเพิ่มขึ้นเป็น 2 เท่าของสองแบบแรกแต่แรงบิดเฉลี่ยจะน้อย

จากการจ่ายกระแสไฟฟ้าเฟสทั้ง 3 วิธี เราก็สามารถสั่งให้สเต็ปมอเตอร์ หมุนวนซ้ำได้โดยมองการจ่ายกระแสให้เฟสย้อนกลับ เช่น ตามเข็มนาฬิกาแบบ Full Step

ตารางที่ 2.4 การจ่ายกระแสไฟฟ้าให้กับสเต็ปมอเตอร์ แบบ Full Step

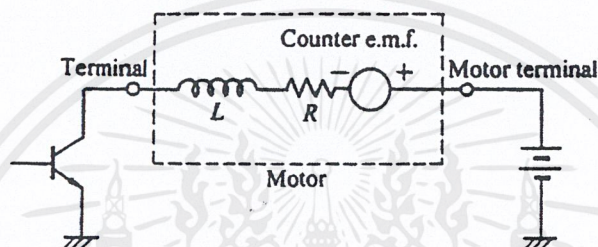
04	03	02	01
1	0	0	1
0	0	1	1
0	1	1	0
1	1	0	0

หมุนตามเข็มนาฬิกา จะเป็น 0104, 1002, 0203, 0304

หมุนทวนเข็มนาฬิกา จะเป็น 0403, 0302, 0201, 0104

### 2.6.8 หลักการออกแบบวงจรขับสเต็ปมอเตอร์

การพิจารณาหลักการออกแบบวงจรขับกำลัง ได้คำนึงถึงหลายแฟกเตอร์ด้วยกัน เนื่องจากการหมุนของ สเต็ปมอเตอร์จะทำให้เกิดแรงดันไฟฟ้าย้อนกลับซึ่งจะมีทิศทางตรงกันข้ามกับแหล่งจ่ายไฟเลี้ยง ซึ่งจะเขียนวงจรสมมุทธ์ 1 เฟสของสเต็ปมอเตอร์ได้ ดังภาพที่ 2.26



ภาพที่ 2.26 แสดงวงจรสมมุทธ์ของสเต็ปมอเตอร์

สเต็ปมอเตอร์ถูกออกแบบให้ทนความร้อนได้สูงถึง 100 องศาเซลเซียสแต่ในการใช้งานจริงจะถูกใช้งานที่เงื่อนไขต่ำกว่าจุดอิมิตัวที่กำหนดมา แต่ในการขับสเต็ปมอเตอร์ต้องเลือกใช้อุปกรณ์ที่กินกระแสสูงเนื่องจากแต่ละเฟสต้อง on และ off อยู่ตลอดเวลา ดังนั้น จึงต้องออกแบบป้องกันทรานซิสเตอร์กำลัง โวลต์เตจที่เกิดจากการเหนี่ยวนำของกระแสในขดลวดและโวลต์เตจเบรคดาวน์ เพื่อป้องกันการเสียหายของทรานซิสเตอร์กำลัง

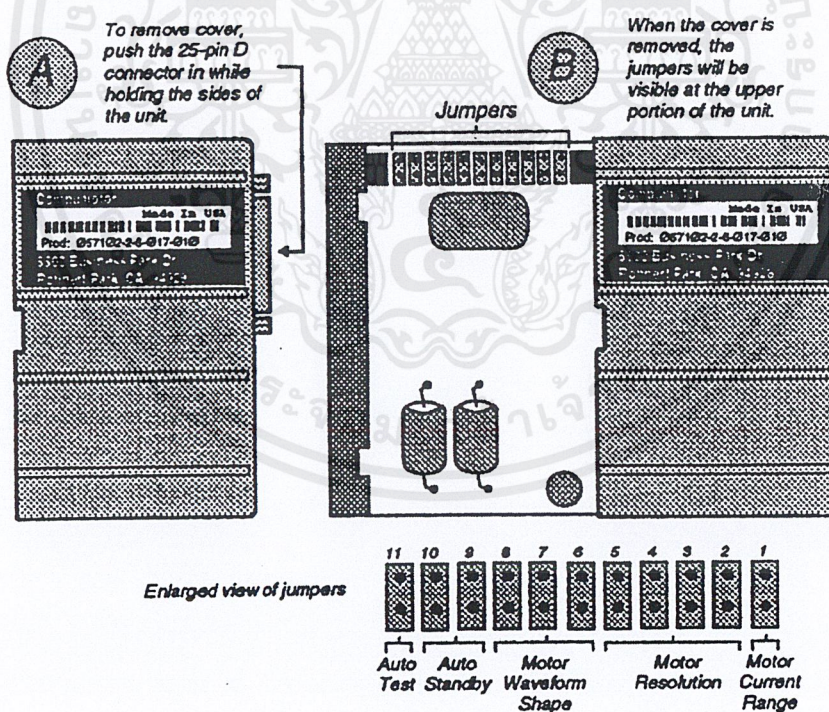
### 2.7 Motor Drive

เราได้เลือกชุดวงจร ไลร์ฟสำเร็จรูปขึ้นมาซึ่งเป็นของ Parker รุ่น Compumotor OEM 650 ขึ้นมาใช้งาน ชุดไลร์ฟ OEM 650 นี้ใช้แหล่งจ่ายไฟจากภายนอกโดยใช้แหล่งจ่ายไฟกระแสตรงขนาด 24-75 โวลต์ โดยใช้ขับกับมอเตอร์ชนิด ไฮบริดจ์ ซึ่งอาจจะเป็นมอเตอร์ชนิด สอง สาม หรือสี่เฟสก็ได้ โดยในการใช้งานนั้นเราอาจเชื่อมต่อในแบบของขดลวดอนุกรมกันหรือว่าขนานกันก็ได้

ข้อจำกัดของขดลวดในแต่ละเฟสของมอเตอร์ที่จะนำมาใช้งานร่วมกับชุดไลร์ฟชุดนี้นั้นจะต้องมีค่าเหนี่ยวนำไม่น้อยกว่า 0.5 มิลลิเฮนรี เพื่อประสิทธิภาพสูงสุดขดลวดในแต่ละเฟสควรมีค่าความเหนี่ยวนำอยู่ในช่วงระหว่าง 1-10 มิลลิเฮนรี คุณสมบัติโดยทั่วไปของวงจรไลร์ฟมีดังนี้ เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- สามารถนำไปใช้งานในการควบคุมการเคลื่อนที่แม้เป็นการเคลื่อนที่จุดเล็กๆ ได้ดีในหลายช่วงความเร็ว
- มีระบบป้องกันการลัดวงจรทั้งการลัดวงจรแบบ เฟสถึงเฟส และเฟสถึงกราวด์
- มีระบบป้องกันการที่มอเตอร์มีสภาพเป็นแหล่งจ่ายไฟฟ้า
- มีระบบป้องกันความร้อนของวงจรไคร์ฟสูงเกิน
- มีพอร์ตติดต่อกับคอมพิวเตอร์ผ่านทางพอร์ตขนาน
- ใช้แหล่งจ่ายไฟกระแสตรงขนาด 24-75 โวลต์
- มีตัวจัมเปอร์ 16 ตัวในการปรับความละเอียดของมอเตอร์
- สัญญาณเข้าและสัญญาณออกสำหรับ ตัวควบคุมการเคลื่อนที่

### 2.7.1 การติดตั้ง

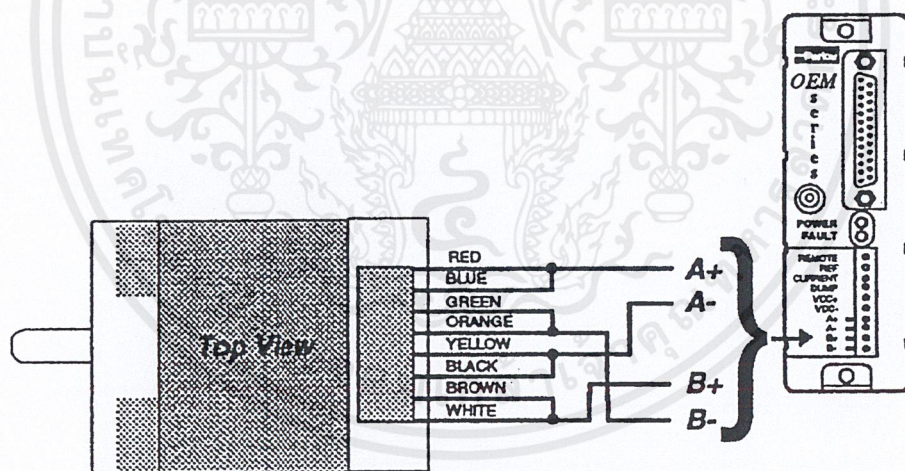


ภาพที่ 2.27 แสดงถึงโครงสร้างของชุดไคร์ฟ Compumotor รุ่น OEM 650

#### 1. นำตัวครอบชุดวงจรไคร์ฟออก

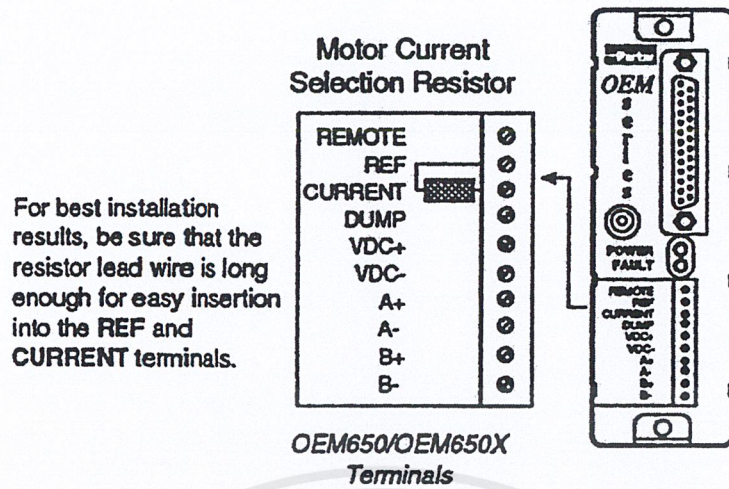
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. เราสามารถทำการทดสอบระบบได้โดยการนำจัมเปอร์ที่ขา 11 ออก เมื่อจ่ายไฟและต่อมอเตอร์ได้ถูกต้อง ระบบจะทำการทดสอบตัวมอเตอร์โดยจะหมุนประมาณ 6 รอบ ที่ความเร็ว 1 รอบต่อนาที เมื่อทุกอย่างปกติทำการปิดฝาครอบ
3. ทำการติดตั้งมอเตอร์ โดยการเชื่อมต่อสายไฟเข้ากับ (ในกรณีที่ใช้การเชื่อมต่อแบบขนาน ควรใช้แหล่งจ่ายไฟขนาดอยู่ในช่วงระหว่าง 24-48 โวลต์ เนื่องจากการใช้แหล่งจ่ายไฟที่มีขนาดเกินกว่านี้จะทำให้เกิดปัญหาด้านความร้อนขึ้นกับวงจรไดรฟ์)
4. ตามตารางที่ 5 ระบุถึงขนาดของกระแสที่เลือกใช้งานและขนาดของตัวต้านทานที่จะใช้เชื่อมต่อระหว่าง ขา REF และ ขา CURRENT เพื่อเป็นตัวจำกัดกระแสใช้งานโดยที่จัมเปอร์ขาที่ 1 บนตัววงจรไดรฟ์เป็นตัวกำหนดขั้วของกระแสที่เลือกใช้ ถ้าถัดวงจรจัมเปอร์ที่ขา 1 ในวงจรตัวไดรฟ์จะสามารถจ่ายกระแสได้ตั้งแต่ 2.5-7.5 แอมป์ ถ้าจัมเปอร์เปิดวงจรจะสามารถจ่ายกระแสได้ 0.3-2.5 แอมป์



ภาพที่ 2.28 แสดงถึงการเชื่อมต่อของสายมอเตอร์เข้าวงจรไดรฟ์เมื่อต่อขดลวดมอเตอร์แบบขนาน

5. ต่อแหล่งจ่ายแรงดันกระแสตรง 24-75 โวลต์ เข้าที่ขา VDC+ และ VDC- ทดสอบการหมุนตามเข็มนาฬิกาและทวนเข็มนาฬิกาตามสัญญาณที่ป้อน



ภาพที่ 2.29 แสดงถึงการเชื่อมต่อตัวต้านทานเพื่อจำกัดกระแสขั้วมอเตอร์



ตารางที่ 2.5 ก) แสดงถึงค่าความต้านทานที่นำมาใช้เมื่อติดตั้งจัมเปอร์ขาที่ 11

urrent Amps)	esistance Ohms)	urrent Amps)	esistance Ohms)
.5		.9	.32 k
.4	05	.8	.68 k
.3	2	.7	.06 k
.2	19	.6	.45 k
.1	25	.5	.87 k
.0	.02 k	.4	.53 k
.9	.27 k	.3	0.0 k
.8	.54 k	.2	0.5 k
.7	.78 k	.1	1.0 k
.6	.05 k	.0	1.5 k
.5	.26 k	.9	2.1 k
.4	.55 k	.8	2.7 k
.3	.80 k	.7	3.3 k
.2	.09 k	.6	3.7 k
.1	.32 k	.5	4.3 k
.0	.57 k	.4	5.0 k
.9	.92 k	.3	5.8 k
.8	.22 k	.2	6.5 k
.7	.53 k	.1	7.4 k
.6	.87 k	.0	8.2 k
.5	.11 k	.9	9.1 k
.4	.49 k	.8	0.0 k
.3	.76 k	.7	0.5 k
.2	.19 k	.6	1.5 k
.1	.49 k	.6	2.6 k
.0	.81 k		

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2.5 ข) แสดงถึงค่าความต้านทานที่นำมาใช้เมื่อติดตั้งจัมเปอร์ขาที่ 11

Current (Amps)	Resistance (Ohms)
2.5	0
2.4	619
2.3	1.27 k
2.2	2.05 k
2.1	2.80 k
2.0	3.57 k
1.9	4.53 k
1.8	5.49 k
1.7	6.49 k
1.6	7.68 k
1.5	8.87 k
1.4	10.5 k
1.3	12.1 k
1.2	13.7 k
1.1	15.8 k
1.0	18.2 k
0.9	20.5 k
0.83	22.6 k

### 2.7.2 รายละเอียดการเซ็ตจัมเปอร์ของวงจรรีเลย์

- ขาที่ 1 จัมเปอร์นี้ถูกตั้งค่าเพื่อปรับขนาดของกระแสที่จะใช้ขับเคลื่อนมอเตอร์
- ขาที่ 2-5 จัมเปอร์ชุดนี้จะเป็นตัวกำหนดความละเอียดในหนึ่งรอบของมอเตอร์ที่จะใช้งานของตัววงจรรีเลย์ ตามตารางที่ 2.6

ตารางที่ 2.6 แสดงถึงการเซ็ตตัวจัมเปอร์เพื่อกำหนดความละเอียดของวงจรรีเลย์

Resolution (steps/rev)	Ju2	Ju3	Ju4	Ju5
50800	On	On	On	Off
50000	On	On	Off	On

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นิยมนำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

36000	On	On	Off	Off
25600	On	Off	On	On
25400	On	Off	On	Off
25000	On	Off	On	On
21600	On	Off	Off	On
20000	On	Off	Off	Off
18000	Off	On	On	On
12800	Off	On	On	Off
10000	Off	On	Off	On
5000	Off	On	Off	Off
2000	Off	Off	On	On
1000	Off	Off	On	Off
400	Off	Off	Off	On
200	Off	Off	Off	Off

- ขาที่ 6-8 จัมเปอร์ชุดนี้จะเป็นตัวกำหนดถึงรูปร่างของกระแสที่ส่งไปยังตัวมอเตอร์ รูปแบบของกระแสที่ส่งไปที่ตัวมอเตอร์สามารถลดปัญหาที่เกิดจากการสังข์พ้องได้

ตารางที่ 2.7 แสดงถึงการเซตจัมเปอร์ขาที่ 6-8 เพื่อกำหนดถึงรูปร่างกระแสขาออกของวงจรไดร์ฟ

Motor waveform	Ju6	Ju7	Ju8
Pure sine	Off	Off	On
-2 % 3 <sup>rd</sup> Harmonic	Off	On	Off
-4 % 3 <sup>rd</sup> Harmonic	On	On	On
-4 % 3 <sup>rd</sup> Harmonic	Off	Off	Off
-4 % 3 <sup>rd</sup> Harmonic	Off	On	On
-6 % 3 <sup>rd</sup> Harmonic	On	Off	Off
-8 % 3 <sup>rd</sup> Harmonic	On	Off	On
-10 % 3 <sup>rd</sup> Harmonic	On	On	Off

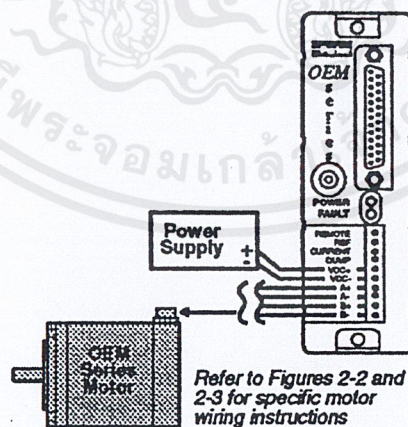
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ขาที่ 9-10 เป็นจัมเปอร์ที่กำหนดถึงขนาดของกระแสที่ส่งไปยังตัวมอเตอร์ขณะที่ไม่ได้เคลื่อนไหว (Standby) ตัวไคร์ฟจะทำการลดกระแสลงตามที่ที่กำหนดไว้เมื่อไม่มีสัญญาณสแต็ปจากตัวควบคุมการเคลื่อนที่ และขับกระแสเต็มอีกครั้งหนึ่งเมื่อได้รับสัญญาณจากตัวควบคุมการเคลื่อนที่อีกครั้ง

ตารางที่ 2.8 แสดงถึงการเซตจัมเปอร์ขาที่ 9-10 เพื่อกำหนดถึงขนาดกระแสที่ขับมอเตอร์เมื่ออยู่ในช่วงไม่ได้ทำงาน

Stanby current	Ju9	Ju10
Full current	On	On
75 % current	Off	On
50 % current	On	Off
25 % current	Off	Off

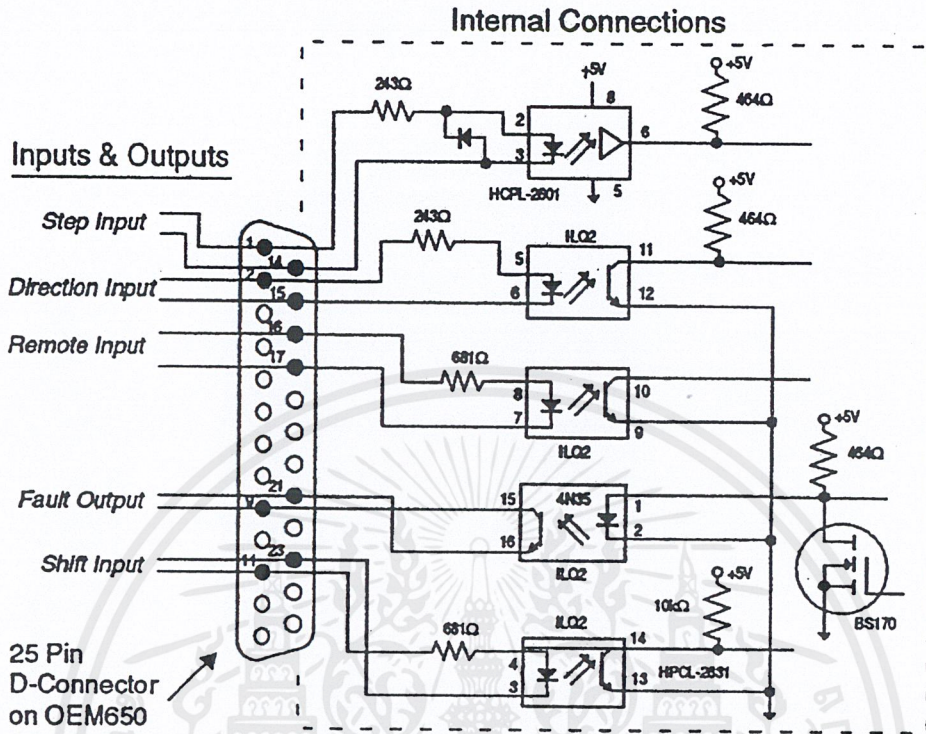
- ขาที่ 11 โดยปกติจะทำการลัดวงจรไว้เมื่อต้องการจะทำการทดสอบการหมุนของมอเตอร์ก็ให้เปิดวงจรออกหากมอเตอร์และวงจรไคร์ฟมีสภาพเป็นปกติ ตัวมอเตอร์ก็จะหมุนประมาณ 6 รอบที่ความเร็วประมาณ 1 รอบต่อวินาที



ภาพที่ 2.30 แสดงถึงการเชื่อมต่อแหล่งจ่ายไฟเข้ากับชุดวงจรไคร์ฟ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.7.3 การเชื่อมต่อสัญญาณเข้าและสัญญาณออก



ภาพที่ 2.31 แสดงถึงรูปแบบการเชื่อมต่อภายในพอร์ต DB-25 ภายในภาครับของชุดวงจรไดรฟ์

- สัญญาณสเต็ปที่ถูกต่อเข้ากับตัวของวงจรไดรฟ์นั้นจะถูกแยกจากตัววงจรอย่างเด็ดขาดโดยใช้แสงในการเชื่อมต่อ แต่สัญญาณที่ส่งเข้ามาที่ตัววงจรไดรฟ์นั้นจะต้องมีกระแสไม่ต่ำกว่า 6.5 มิลลิแอมป์และระดับแรงดันไฟระหว่างขาทั้งสองประมาณ 3.5 โวลต์ และสัญญาณที่ส่งไปนั้นจะต้องมีความกว้างของสัญญาณไม่น้อยกว่า 200 นาโนวินาที และอยู่ในช่วง 40-60 % ของหนึ่งรอบการทำงาน
- สัญญาณที่ควบคุมทิศทางการเคลื่อนที่ที่ถูกต่อเข้ากับวงจรไดรฟ์นั้นจะถูกแยกจากตัววงจรอย่างเด็ดขาดโดยใช้แสงในการเชื่อมต่อ แต่สัญญาณที่ส่งเข้ามาที่ตัววงจรไดรฟ์นั้นจะต้องมีกระแสไม่ต่ำกว่า 10 มิลลิแอมป์และระดับแรงดันไฟระหว่างขาทั้งสองประมาณ 3.5 โวลต์ สัญญาณควบคุมทิศทางนั้นจะสามารถเปลี่ยนทิศทางได้เมื่อหมดสัญญาณสเต็ปลูกสุดท้าย และสัญญาณเปลี่ยนทิศทางนั้นจะต้องอยู่ค้างอย่างน้อย 120 ไมโครวินาทีก่อนที่สัญญาณสเต็ปจะมาถึงตัวไดรฟ์จึงจะสามารถเปลี่ยนทิศทางได้

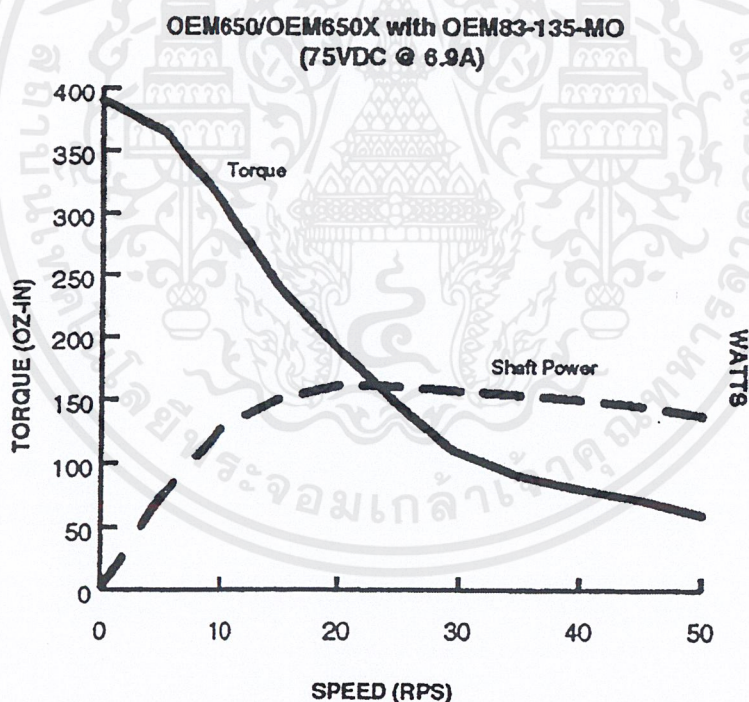
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไมอนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 2.7.4 การปรับแต่งวงจรไครฟ์

สเต็มอเตอร์ทุกตัวจะเกิดความถี่สั่นพ้องเกิดขึ้นเนื่องจาก โครงสร้างของตัวมอเตอร์เองมันจะทำตัวมอเตอร์เกิดการสั่นเมื่อเคลื่อนที่ที่ความเร็วรอบต่ำๆ โดยทั่วไปวงจรไครฟ์จะถูกตั้งค่าให้มีความเร็วรอบมากกว่าค่าความเร็วที่ทำให้เกิดการสั่นพ้อง

## 2.7.5 รายละเอียดคุณสมบัติของตัววงจรไครฟ์

- ความเที่ยงตรง +/- 5 เศษหนึ่งส่วนนาที่เมื่อไม่ได้ต่อโหลด
- ความสามารถการทำซ้ำ +/- 5 เศษหนึ่งส่วนวินาทีเมื่อไม่ได้ต่อโหลด
- ฮิสเทอไลซิส น้อยกว่า 2 เศษหนึ่งส่วนนาที่ -0.0334 องศา
- ความเฉื่อยของโรเตอร์ 1.36 ออนซ์ต่อตารางนิ้ว หรือ 0.25 กิโลกรัมต่อตารางเซนติเมตร
- คุณสมบัติของมอเตอร์ถูกแสดงอยู่ในรูปของความเร็วส่วนด้วยเวลาดังนี้

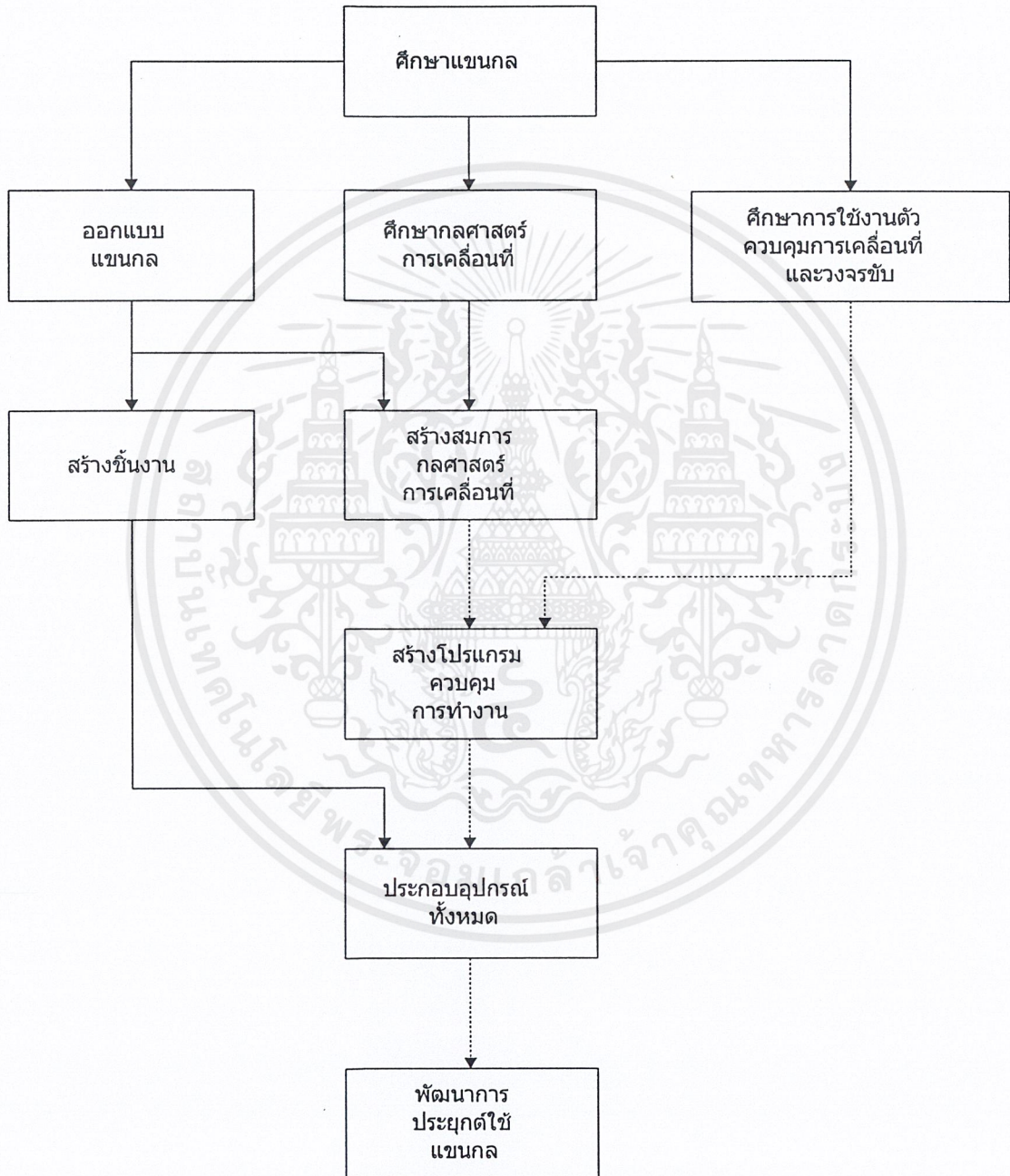


ภาพที่ 2.32 แสดงถึงคุณสมบัติแรงบิดของมอเตอร์เมื่อทำงานที่ความเร็วต่างๆกัน

### บทที่ 3

## การกำหนดและการสร้าง

### 3.1 Project Flowchart

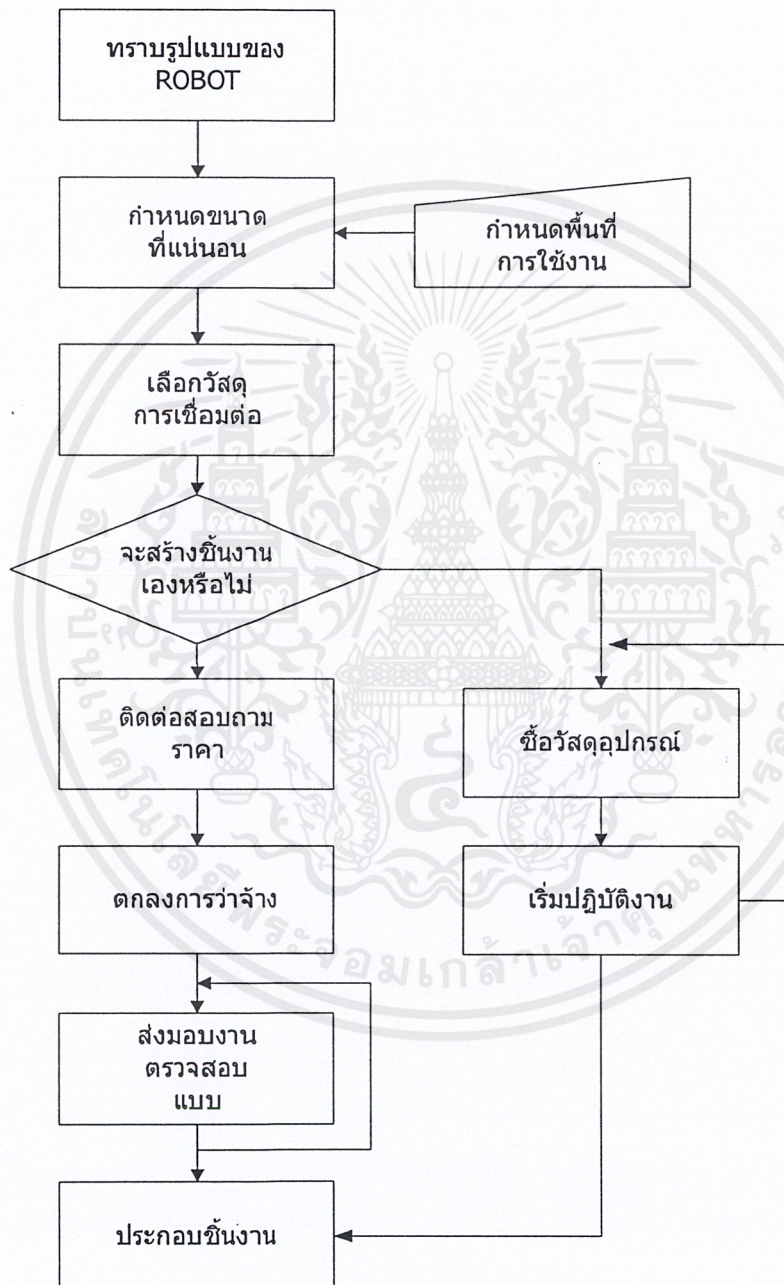


ภาพที่ 3.1 แสดง Project Flowchart

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.2 Hardware Flowchart

## Hardware Flowchart



ภาพที่ 3.2 Hardware Flowchart

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

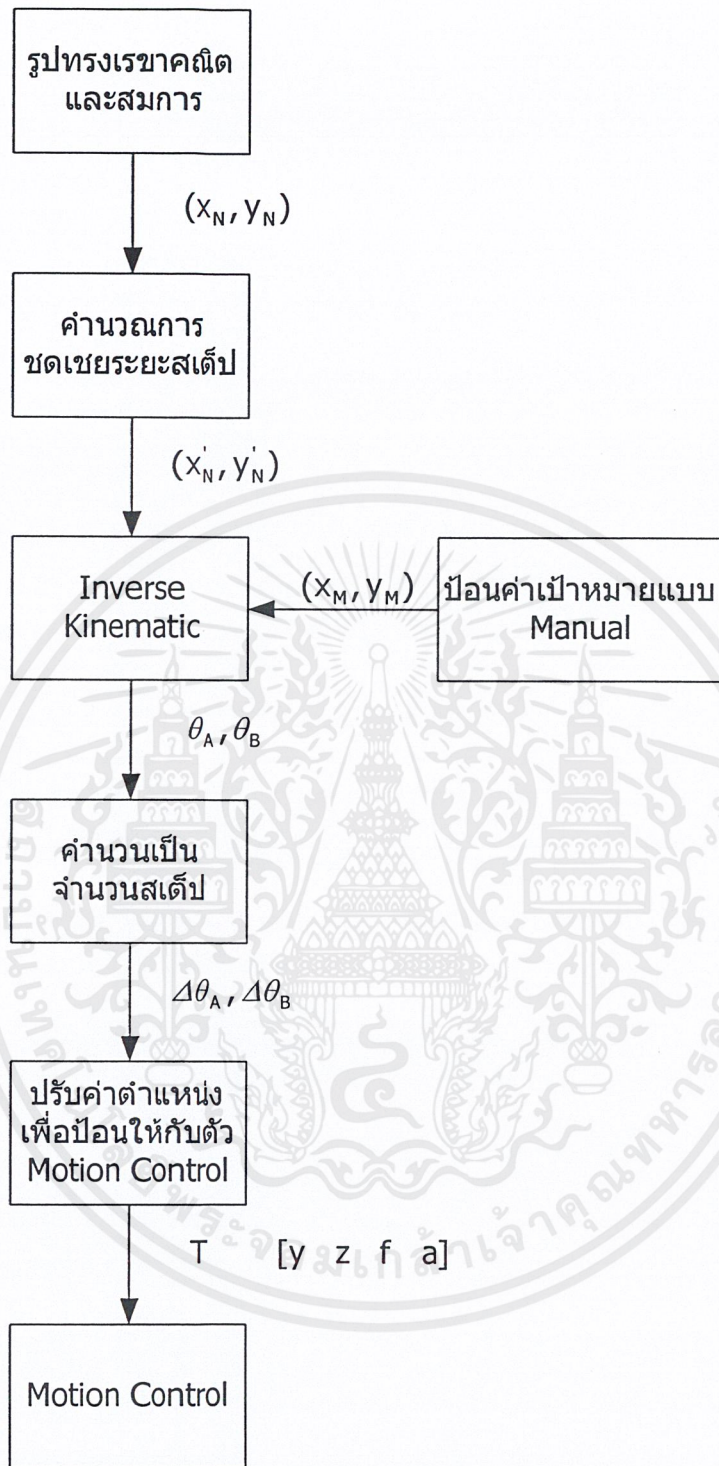
### 3.2.1 ลำดับความสำคัญในการสร้างชิ้นงาน

1. ARM A,B,C,DE เป็นส่วนที่ต้องสร้างก่อนเพื่อดูผลจากการนำแขนมาเชื่อมต่อกันผ่านตัวล็อกและตลับลูกปืน เพื่อดูพื้นที่การทำงานและเงื่อนไขในการหมุนว่าเป็นไปตามที่ต้องการหรือไม่

2 . ตัวล็อก(catch),Catchmotor เป็นชิ้นส่วนที่ดูการเคลื่อนไหวของแขนกลเมื่อต่อเข้ากับมอเตอร์ จากส่วนนี้ ARM A,B,C,DE ก็สามารถทดลองหมุนจากวงจร ใดก็ได้เพื่อดูการเคลื่อนที่ได้แล้ว

Motor up, Motor down ,Motorbase up, Motorbase down, Base, Front-t-base, Back-t-base เป็น ชิ้นส่วนชุดสุดท้ายเมื่อชิ้นส่วนในชุดที่ 1 และ 2 สามารถเคลื่อนที่ได้ตามที่ต้องการแล้ว ส่วนนี้จึงเป็นเพียงการประกอบขึ้นเป็นแขนกลที่สมบูรณ์

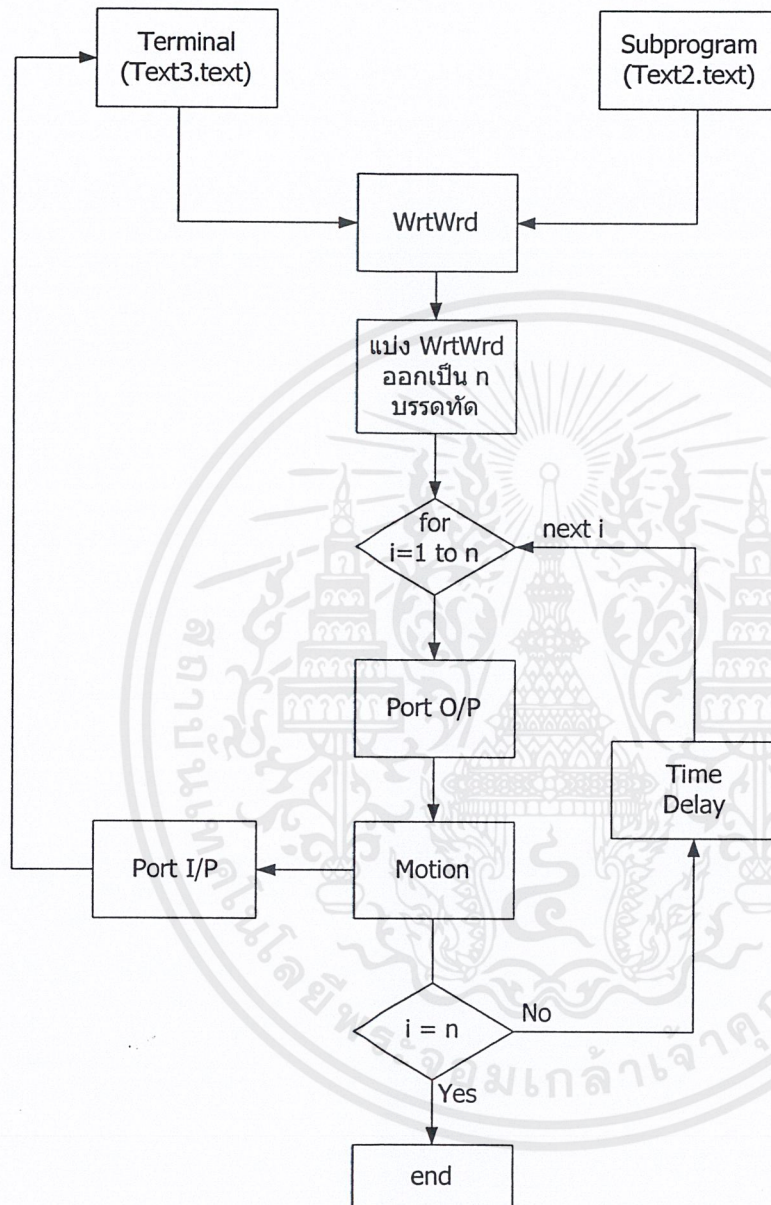




ภาพที่ 3.3 Software Flowchart

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

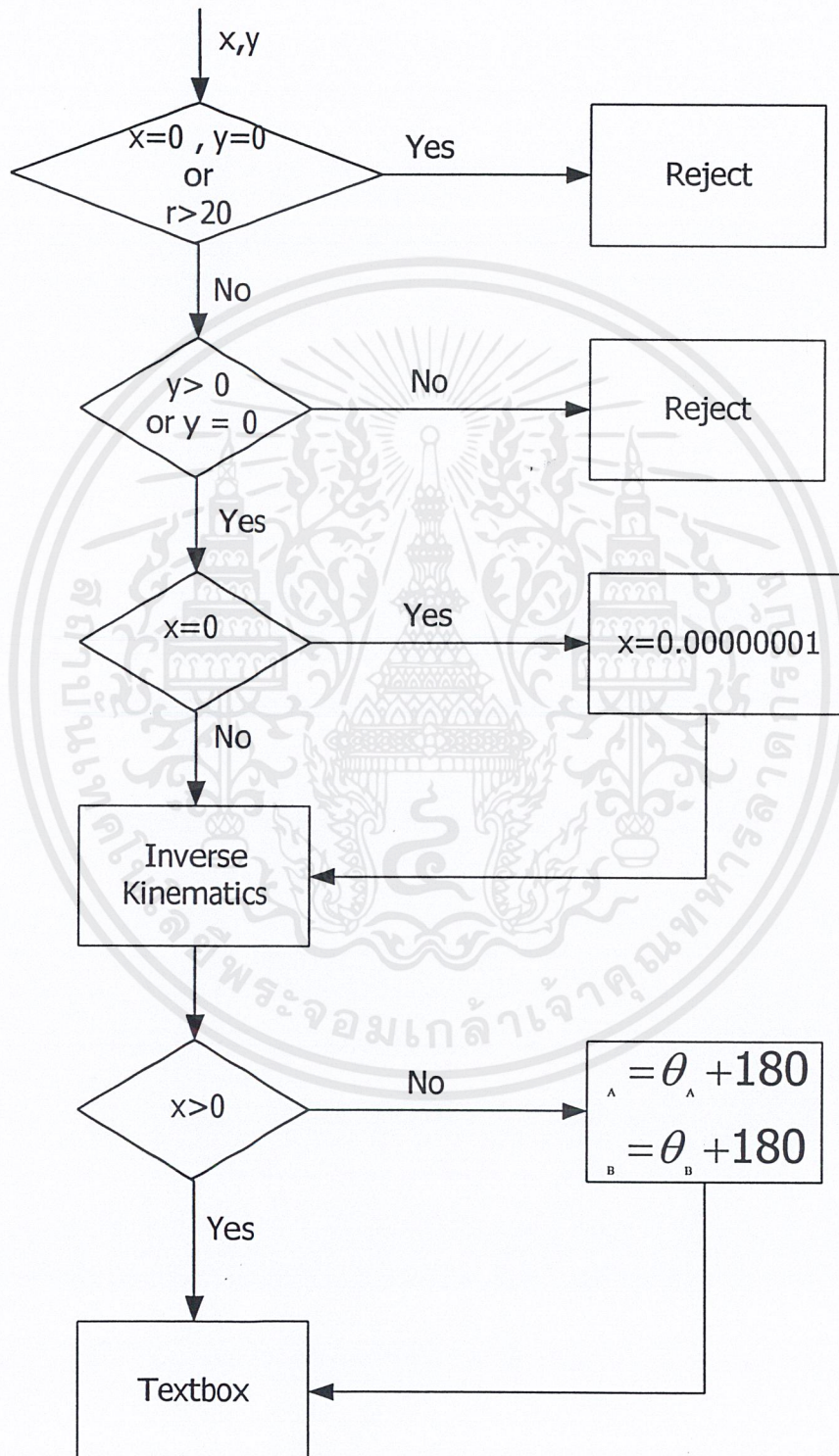
### 3.4 Terminal Program Flowchart



ภาพที่ 3.4 Terminal Program Flowchart

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

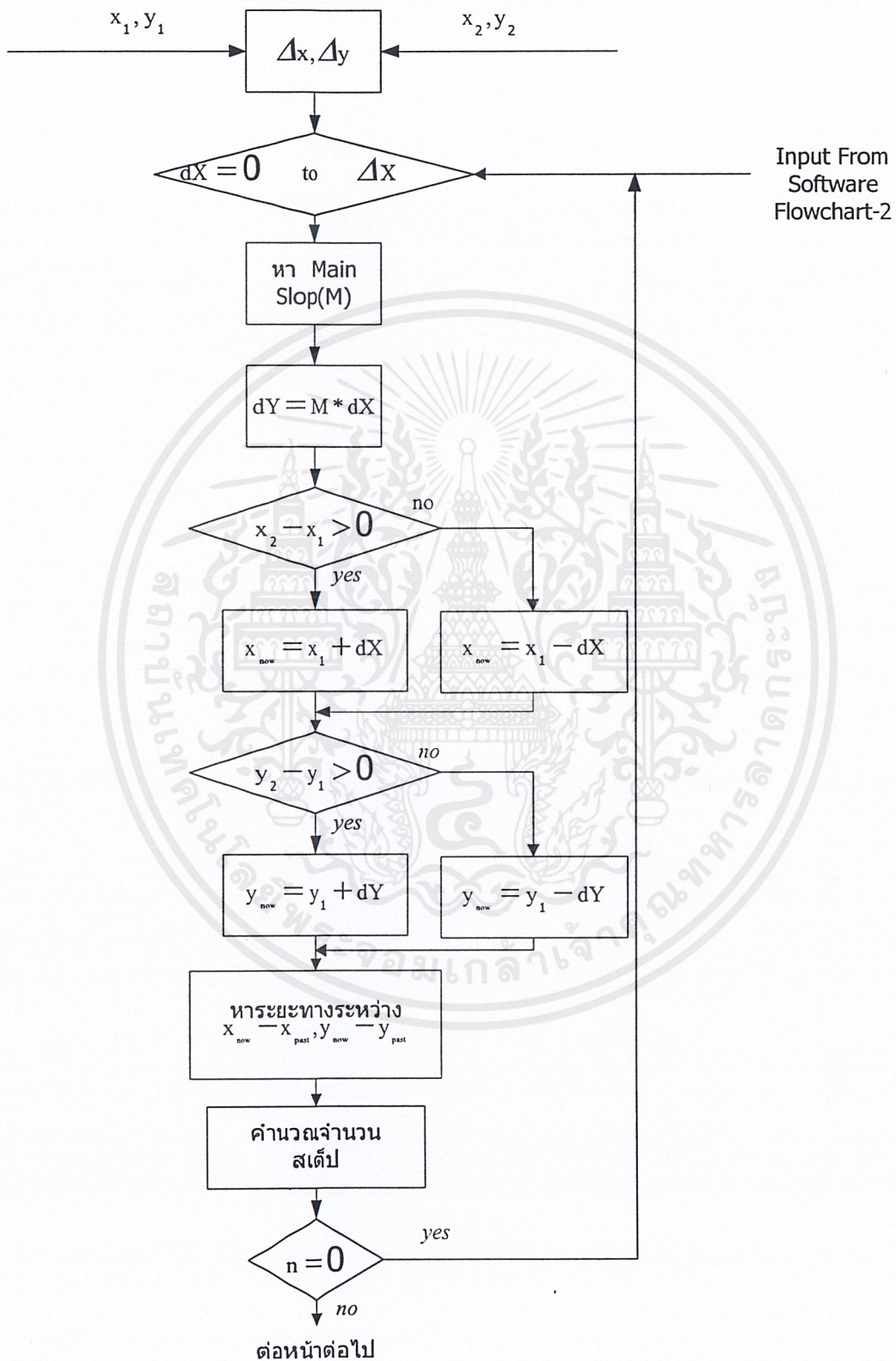
### 3.5 Point Plot Flowchart



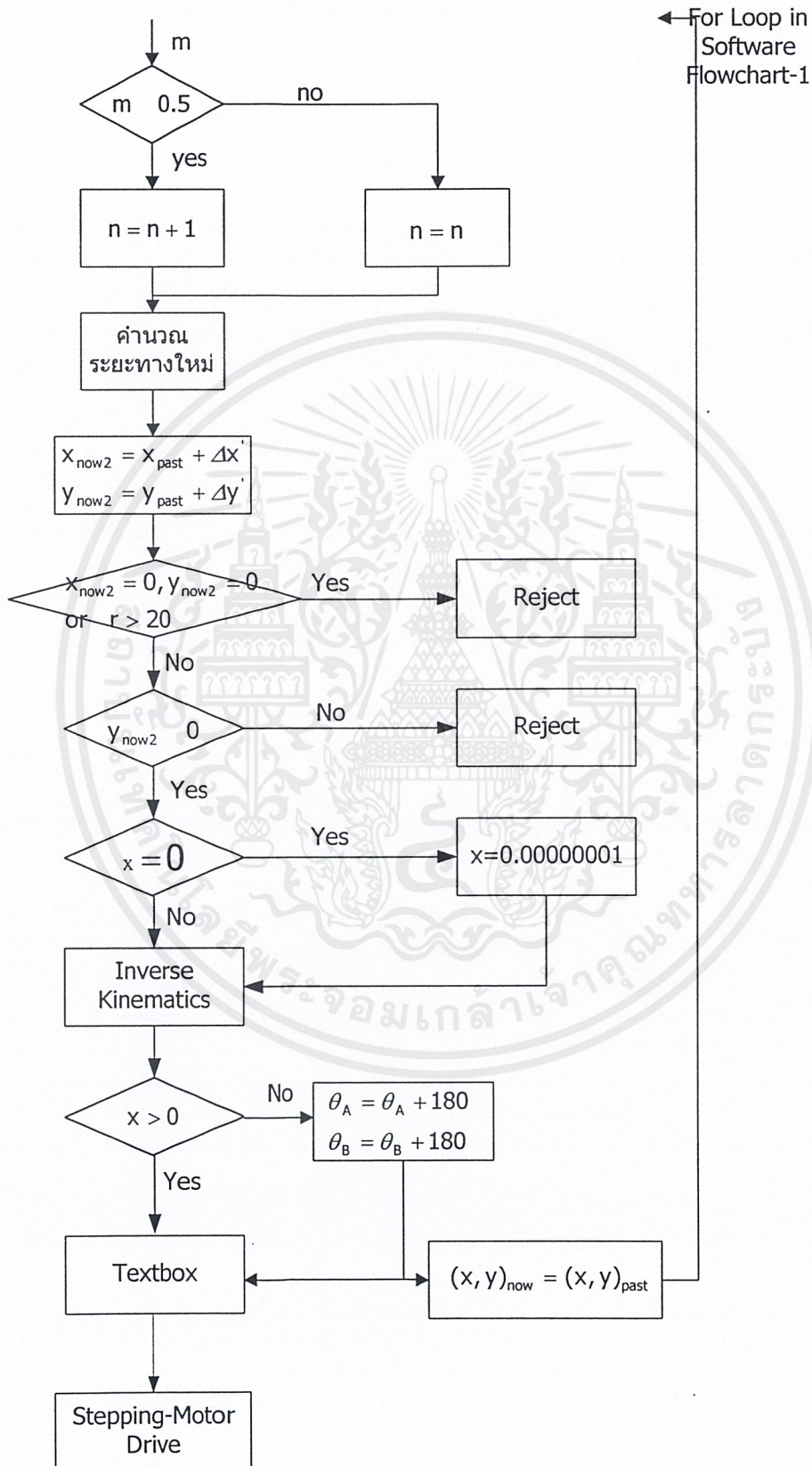
ภาพที่ 3.5 Point Plot Flowchart

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 3.6 Main Plot Flowchart



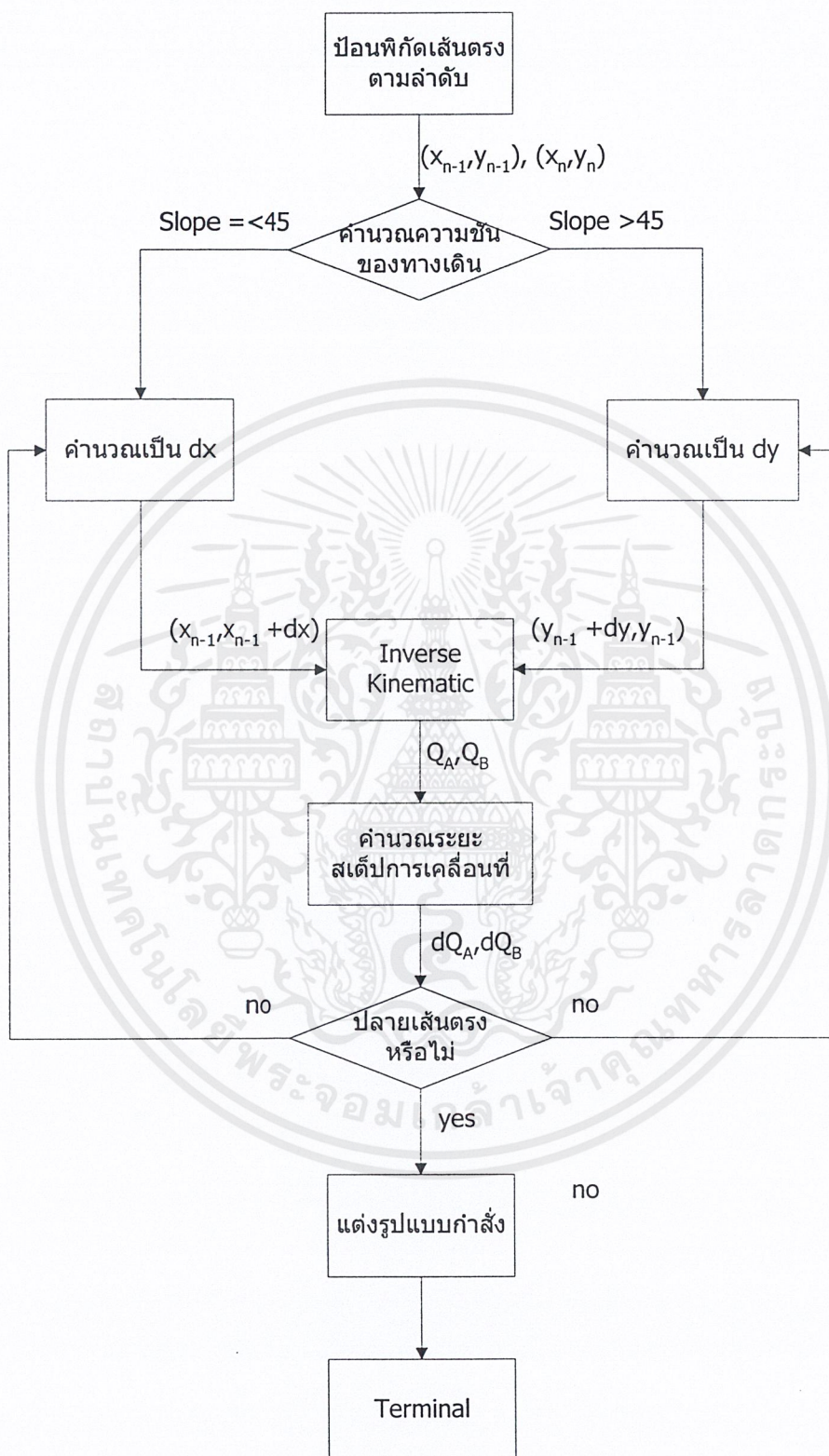
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 3.6 Main Plot Flowchart

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

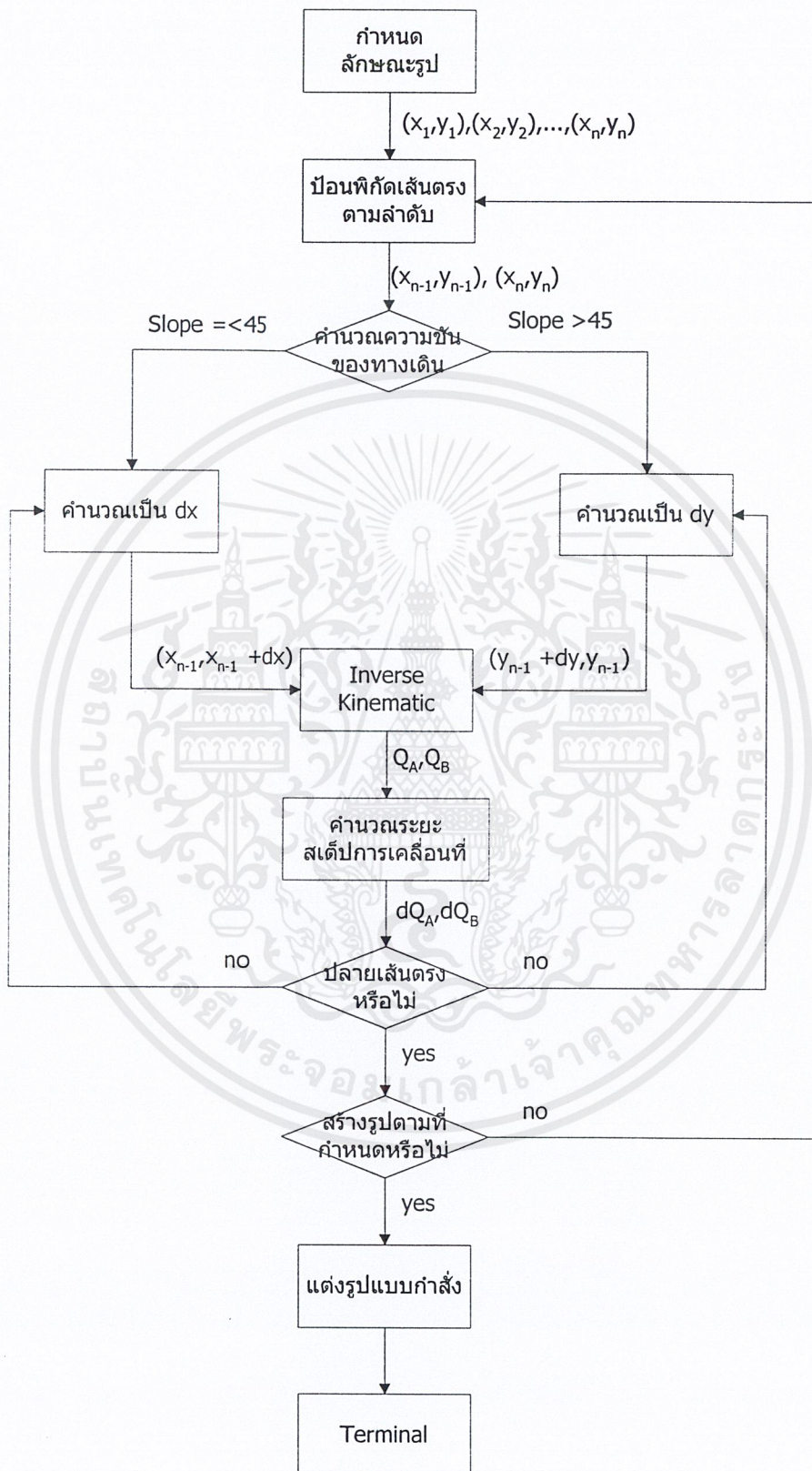
## 3.7 Line Plot Flowchart



ภาพที่ 3.7 Line Plot Flowchart

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

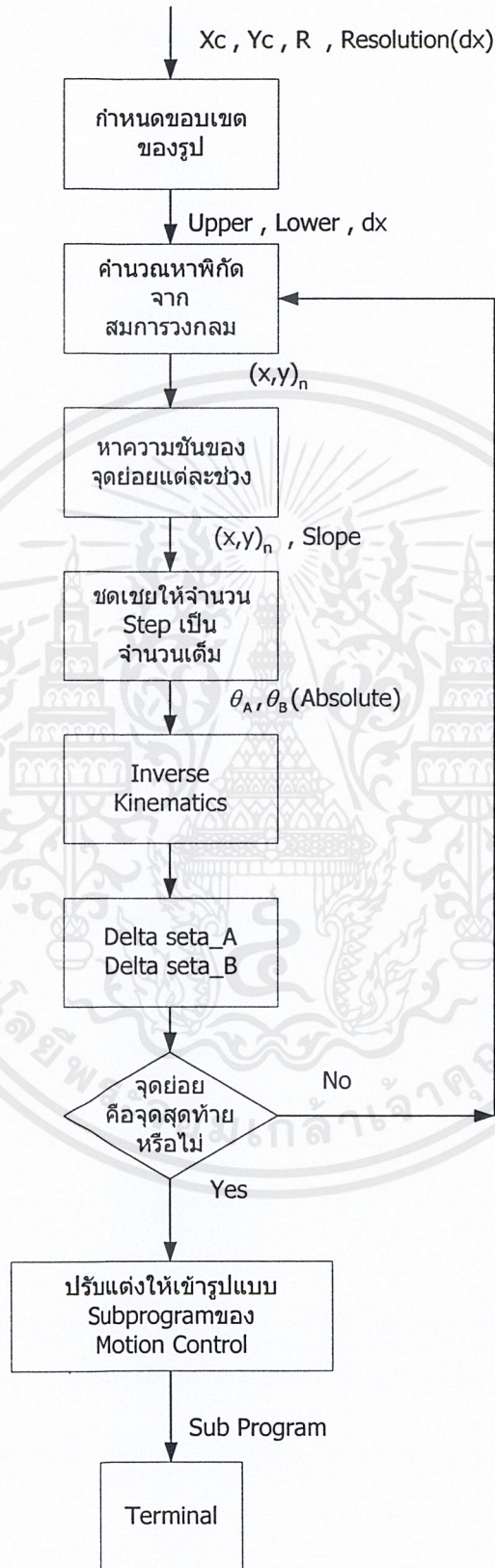
### 3.8 Polygon Plot Flowchart



ภาพที่ 3.8 Polygon Plot Flowchart

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.9 Circle Plot Flowchart

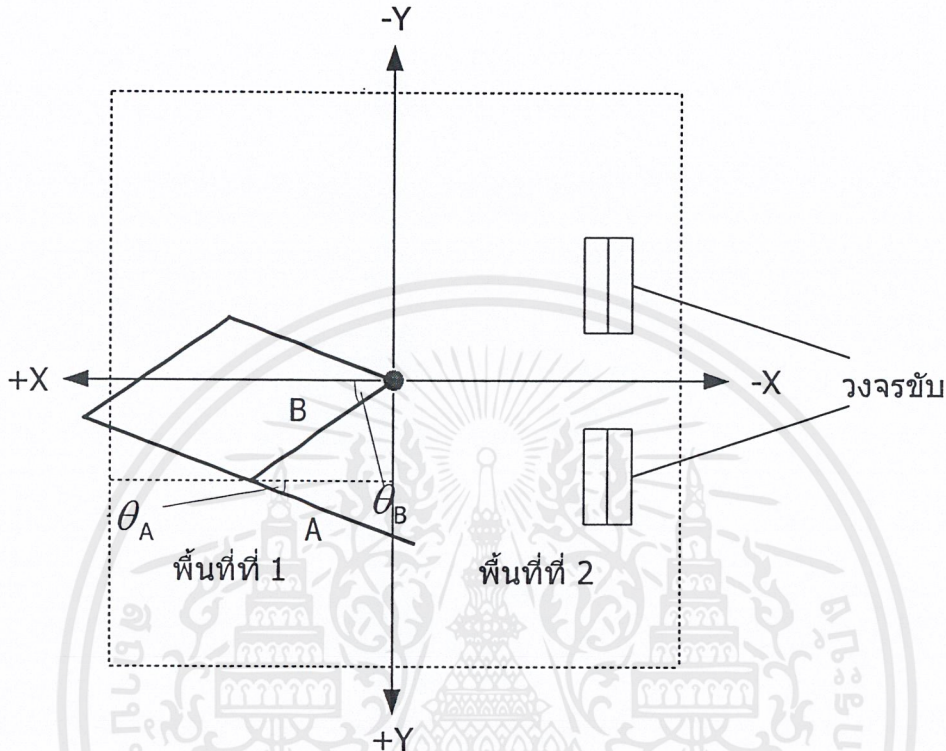


ภาพที่ 3.9 Circle Plot Flowchart

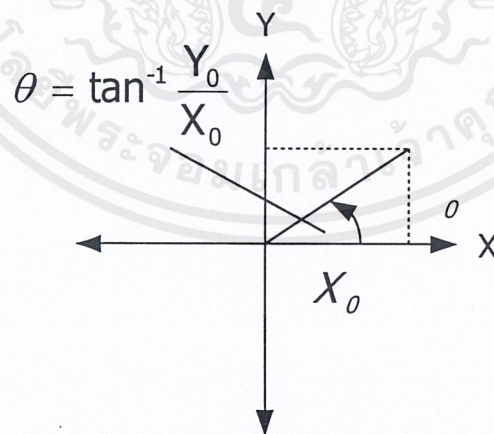
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.10 การพิสูจน์สมการ Inverse Kinematics Equation

เนื่องจากข้อจำกัดของแขนกลที่สร้างขึ้น เราเลือกให้แขนกลทำงานใน พื้นที่ที่ 1 และ 2 ดังในภาพที่ 3.10



ภาพที่ 3.10 แสดงบริเวณการทำงานของแขนกลจับตรง ในพื้นที่ที่ 1 และพื้นที่ที่ 2



ภาพที่ 3.11 แสดงค่ามุมในพื้นที่ที่ 1 มีทิศทวนเข็มนาฬิกา

พิจารณาในพื้นที่ที่ 1 ตามภาพที่ 3.11 เราใช้สมการกลศาสตร์ที่เกี่ยวข้องกับการเคลื่อนที่โดยตรง และตรงข้าม (Direct and Inverse Kinematics Equation) ในการหาค่ามุม และ พิกัดต่างๆ ได้มีสมการดังนี้ เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปเผยแพร่ขึ้นด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

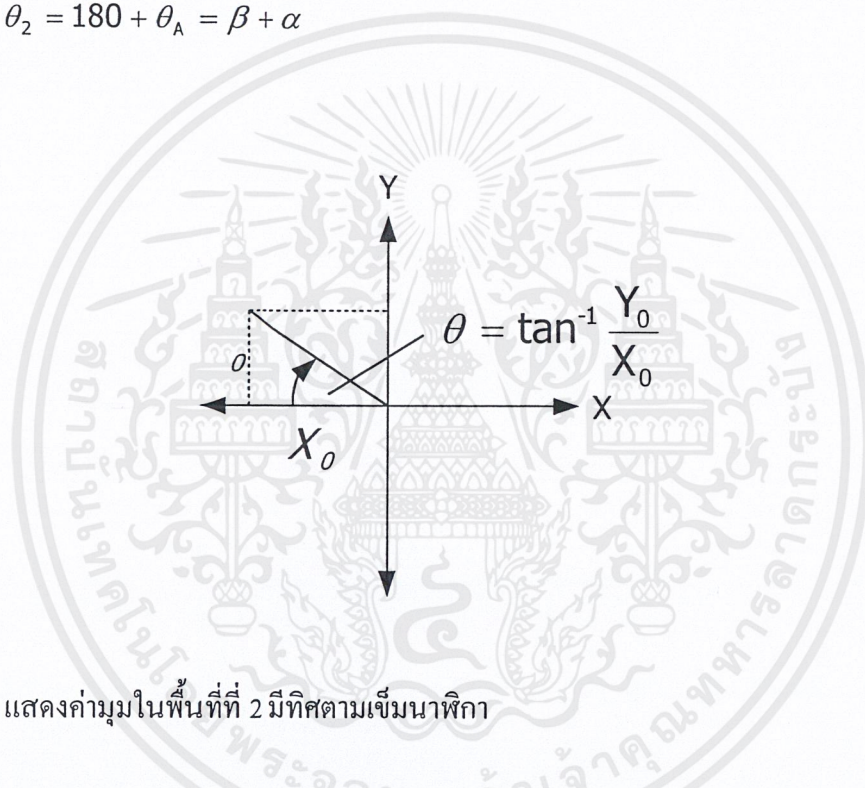
$$\beta = \cos^{-1} \frac{r_0^2 - L_1^2 - L_2^2}{2L_1L_2}$$

$$\phi = \sin^{-1} \frac{L_2}{r_0} \sqrt{1 - \cos^2 \beta}$$

$$\alpha = \tan^{-1} \frac{Y_0}{X_0} - \sin^{-1} \frac{L_2}{r_0} \sqrt{1 - \cos^2 \beta}$$

$$\theta_1 = \theta_B = \alpha$$

$$\theta_2 = 180 + \theta_A = \beta + \alpha$$



ภาพที่ 3.12 แสดงค่ามุมในพื้นที่ที่ 2 มีทิศตามเข็มนาฬิกา

พิจารณาในพื้นที่ที่ 2 เราใช้สมการกลศาสตร์ที่เกี่ยวกับการเคลื่อนที่โดยตรง และ ตรงข้าม (Direct and Inverse Kinematics Equation) ได้เช่นเดียวกับพื้นที่ที่ 1 แต่ค่า  $\alpha$  ที่ได้จากสมการ ต้องนำไปบวกกับ 180 องศาและเนื่องจากค่า  $X_0$  ในพื้นที่ที่ 2 มีค่าเป็นลบ(-)ดังนั้นจะทำให้ค่า  $\tan^{-1} \frac{Y_0}{X_0}$  มีค่าออกมาเป็นลบซึ่งมีทิศตามเข็มนาฬิกาด้วยซึ่งจะได้มุมตามภาพที่ 3.3 แต่มุมที่เราต้องการคือมุมในทิศทวนเข็มนาฬิกา ดังนั้นเราต้องนำค่า 180 ไปบวกกับ  $\tan^{-1} \frac{Y_0}{X_0}$  ซึ่งเป็นเหตุผลว่าเมื่อเราทำงานในพื้นที่ที่ 2 เราจึงต้องใช้สมการดังนี้

$$\beta = \cos^{-1} \frac{r_0^2 - L_1^2 - L_2^2}{2L_1L_2}$$

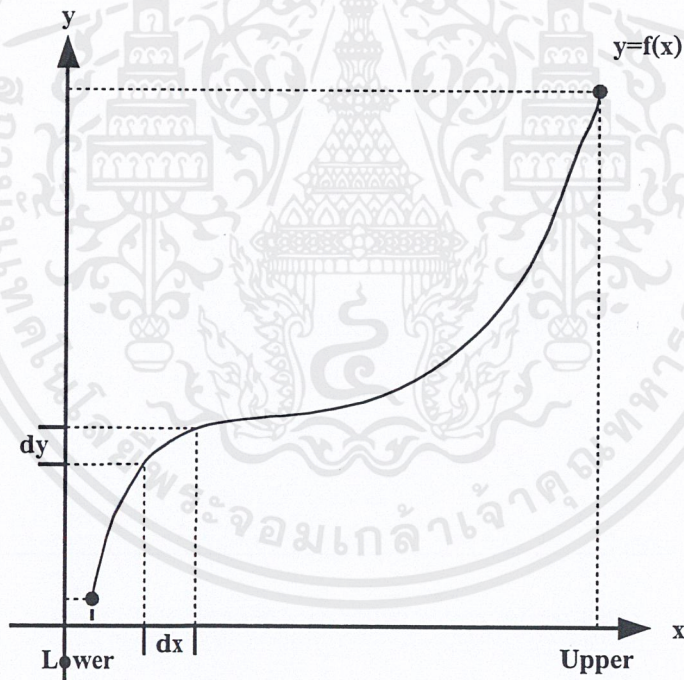
$$\phi = \sin^{-1} \frac{L_2}{r_0} \sqrt{1 - \cos^2 \beta}$$

$$\alpha = 180 + \tan^{-1} \frac{y_0}{x_0} - \sin^{-1} \frac{L_2}{r_0} \sqrt{1 - \cos^2 \beta}$$

$$\theta_1 = \theta_B = \alpha$$

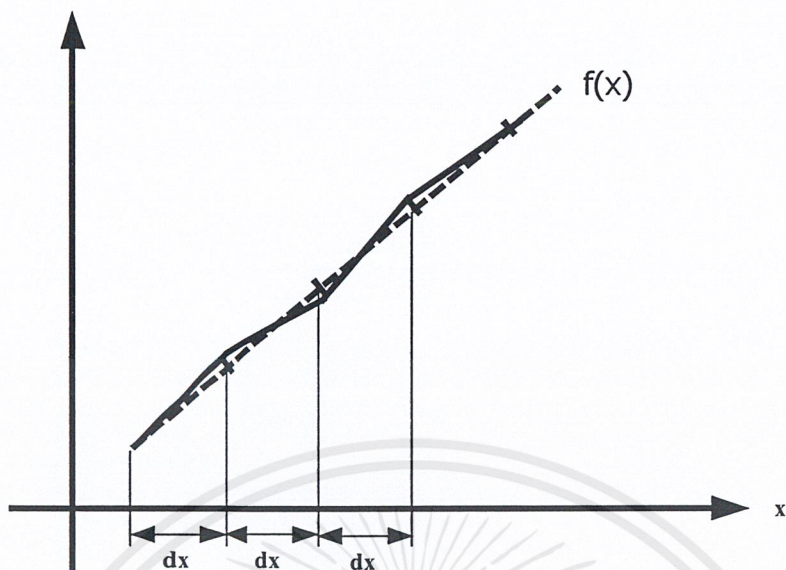
$$\theta_2 = 180 + \theta_A = \beta + \alpha$$

### 3.11 การชดเชย (Compensate)



ภาพที่ 3.13 แสดงการแบ่งระยะการเคลื่อนที่ โดย dx และ dy

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 3.14 แสดงการชดเชยระยะการเคลื่อนที่ของแขนกล

เนื่องจาก Direct Drive Robot ที่สร้างขึ้นมีความสัมพันธ์ระหว่างเวกเตอร์ลิฟท์กับ โครงสร้าง เป็นแบบ Polar form  $(r, \theta)$  ทำให้การเคลื่อนที่จากจุดหนึ่งไปยังอีกจุดหนึ่งมีลักษณะเป็นเส้น โค้ง ทำให้เส้นทางเดินผิดจากที่เราต้องการ โดยเฉพาะเส้นตรงทางเดินที่ได้จะผิดจากเส้นตรงมาก

การแก้ปัญหาเรื่องทางเดินที่เป็นเส้น โค้งนี้ทำได้โดยกำหนดระยะห่างระหว่างจุดให้มีขนาดเล็กที่สุดเท่าที่สามารถจะกำหนดได้ เพราะทำให้เส้น โค้งที่ถูกแบ่งนี้มีใกล้เคียงเส้นตรงมากที่สุด เมื่อทำการแบ่งเส้นทางออกเป็นจุดย่อยหลายจุดแล้ว

ปัญหาต่อมาที่พบ คือระยะการเคลื่อนที่ของแขนกล ไม่ใช่ระยะที่พอดีกับจุดย่อยที่ได้แบ่งไว้ทำให้เราต้องทำการชดเชยเพื่อหาจุดที่ระยะการเคลื่อนที่ลงตัวดัดความผิดพลาดน้อยที่สุด การแก้ไขทำได้โดย

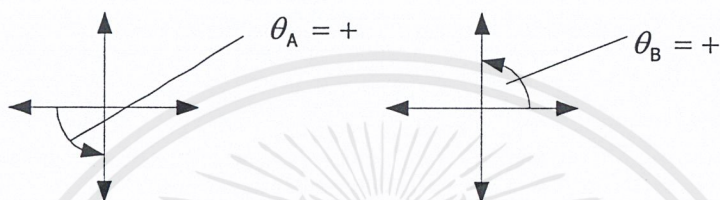
1. หาระยะระหว่างจุดย่อยโดยการประมาณแบบเส้นตรง
2. นำระยะต่อหนึ่ง setp การเคลื่อนที่มาหารระยะระหว่างจุดย่อย
3. นำเศษจากการหารมาพิจารณาว่าควรปรับจำนวน step ให้มากขึ้นหรือเท่าเดิม
4. เคลื่อนที่เท่ากับจำนวน step ที่คำนวณจะได้จุดปลายใหม่
5. จากนั้นคิดระยะระหว่างจุดปลายจากข้อที่ 4 ไปหาจุดปลายจุดต่อไปจากเส้น โค้ง  $f(x)$
6. ทำเช่นเดียวกับข้อ 1 – 5 เช่นนี้ไปจนครบช่วงของ  $dx$  (หรือ  $dy$ ) ที่ต้องการทางเดิน ที่ได้จะมีระยะ Error  $\leq$  ระยะทาง 1 step

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

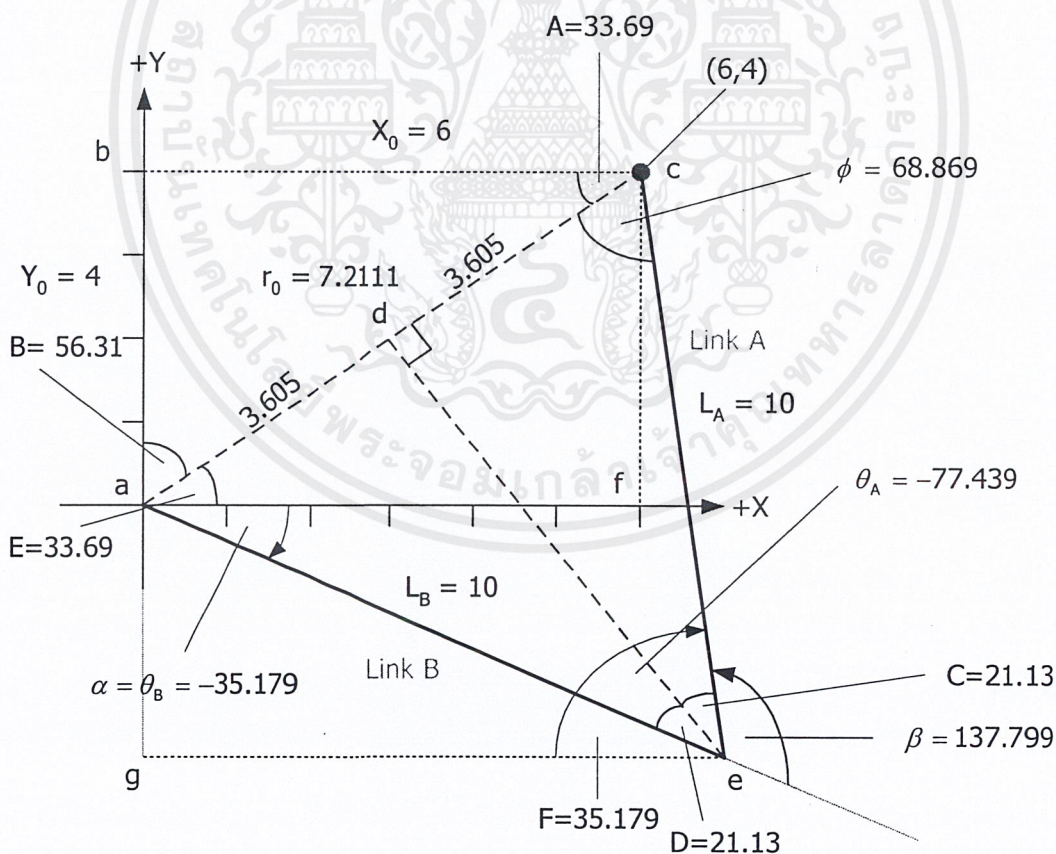
# บทที่ 4

## ผลการทดลอง

### 4.1 ตัวอย่างการคำนวณเทียบกับค่าที่คำนวณได้จากโปรแกรมในพื้นที่การทำงานที่ 1



ภาพที่ 4.1 แสดงทิศทางของ  $\theta_A$  และ  $\theta_B$  ซึ่งจะมีค่าเป็นบวกเมื่อมีทิศทวนเข็มนาฬิกา



ภาพที่ 4.2 แสดงการหาค่า  $\theta_A$  และ  $\theta_B$  ในพื้นที่ที่ 1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากสามเหลี่ยม abc เราจะสามารถหาค่ามุม A ได้จากสมการต่อไปนี้

$$\text{มุม}A = \tan^{-1} \frac{Y_0}{X_0} = \tan^{-1} \frac{4}{6} = 33.69$$

จากมุมภายในสามเหลี่ยมมุมฉาก abc มีค่าเท่ากับ 180 องศา ดังนั้นเราจะได้อ่า

$$\text{มุม}B = 180 - 90 - \text{มุม}A = 56.31$$

$$\text{และค่า } r_0 = \sqrt{X_0^2 + Y_0^2} = \sqrt{6^2 + 4^2} = 7.2111 \text{ ดังนั้นค่า } \frac{r_0}{2} = \frac{7.2111}{2} = 3.6$$

จากสามเหลี่ยม cde เราจะสามารถหาค่ามุม C ได้จากสมการต่อไปนี้ โดยที่  $L_A=10$

$$\text{มุม}C = \sin^{-1} \frac{cd}{L_A} = \sin^{-1} \frac{3.6}{10} = 21.1$$

เนื่องจาก  $L_A$  และ  $L_B$  มีค่าเท่ากับ 10 ทำให้สามเหลี่ยม ace เป็นสามหน้าจั่วทำให้

$$\text{มุม}D = \text{มุม}C = 21.1 \text{ องศา}$$

จากมุมภายในสามเหลี่ยมมุมฉาก cde มีค่าเท่ากับ 180 องศา ดังนั้นเราจะได้อ่า

$$\phi = 180 - 90 - \text{มุม}C = 180 - 90 - 21.1 = 68.9$$

มุม A เป็นมุมแย้ง กับ มุม E ดังนั้น มุม E = 33.69

ดังนั้นเราจะได้อ่ามุม  $\alpha = \theta_B = 68.9 - \text{มุม}E = 68.9 - 33.69 = 35.21$

แต่เนื่องจากค่ามุม  $\alpha$  มีทิศตามเข็มนาฬิกาที่ได้แสดงไว้ในภาพที่ 1 ดังนั้นเราจะได้อ่าจริงๆคือ

$$\alpha = \theta_B = -35.21$$

มุม F เป็นมุมแย้ง กับ มุม  $\alpha$  ดังนั้น มุม F = 35.21

และจะได้  $\theta_A = \text{มุม}C + \text{มุม}D + \text{มุม}F = 21.13 + 21.13 + 35.21 = 77.47$

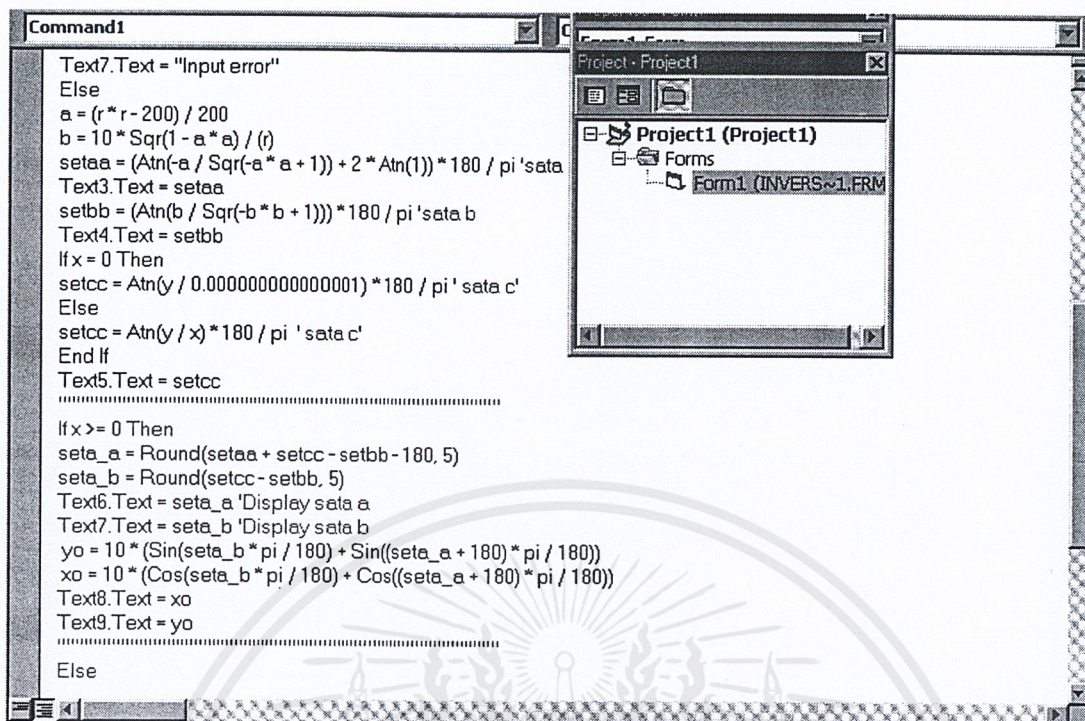
แต่เนื่องจากค่ามุม  $\theta_A$  มีทิศตามเข็มนาฬิกาที่ได้แสดงไว้ในภาพที่ 1 ดังนั้นเราจะได้อ่าจริงๆคือ

$$\theta_A = -77.47$$

จะได้อ่ามุม  $\beta = 180 + \theta_A - \alpha = 180 - 77.47 + 35.21 = 137.74$

จากการคำนวณข้างบนจะได้อ่า  $\theta_A$  และ  $\theta_B$  นี้เราจะนำมาเทียบกับค่า  $\theta_A$  และ  $\theta_B$  ที่ได้จาก โปรแกรมดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 4.3 แสดงโปรแกรมการคำนวณในพื้นที่ที่ 1

X <sub>0</sub>	6	Command1
Y <sub>0</sub>	4	
$\beta$	137.67600105782	
$\phi$	68.8380005289101	
$\alpha$	33.6765127486334	
$\theta_A$	-77.48549	
$\theta_B$	-35.16149	
X <sub>0</sub>	6.00000033530997	
Y <sub>0</sub>	3.9999998076092	

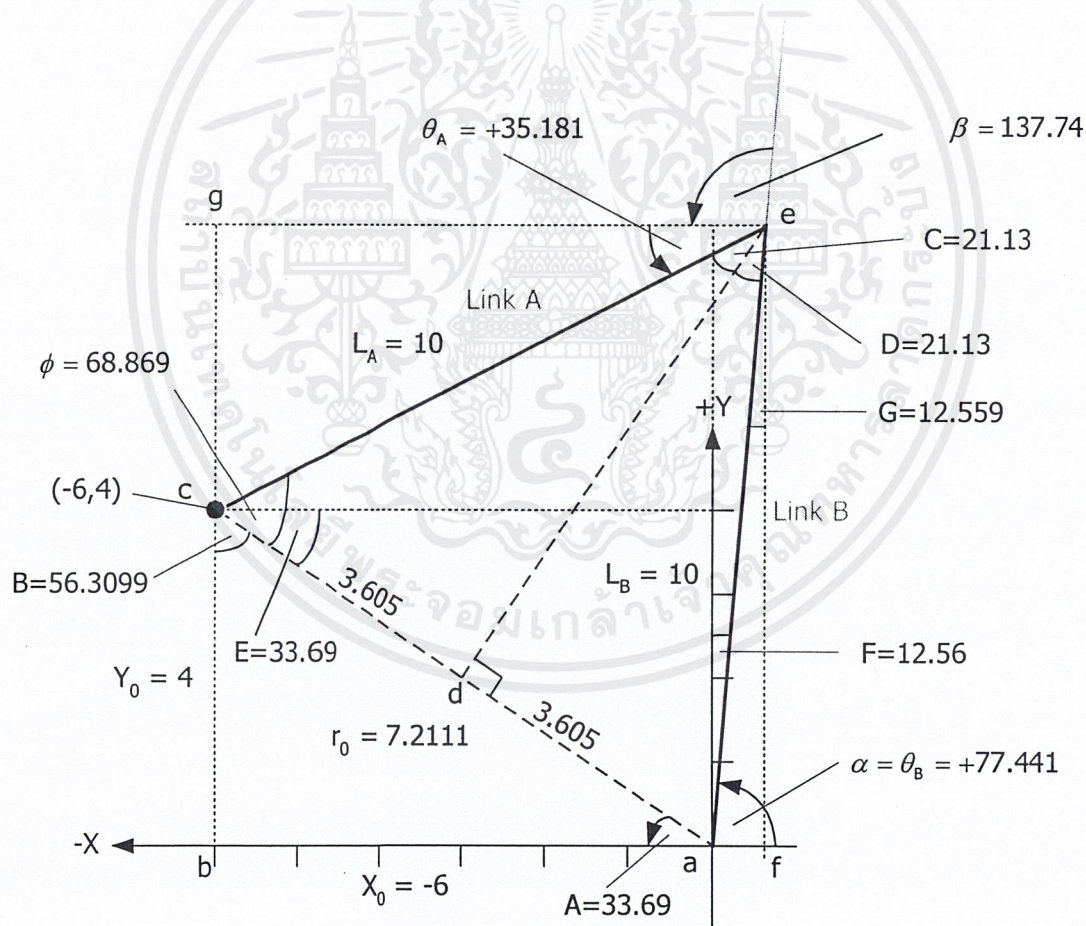
ภาพที่ 4.4 แสดงค่าที่คำนวณได้จากโปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.1 เปรียบเทียบค่าที่คำนวณได้กับค่าที่ได้จากProgram

	ค่าที่ได้จากการคำนวณ	ค่าที่ได้จาก Program
$\beta$	137.74	137.67
$\phi$	68.9	68.83
$\alpha$	35.21	33.67
$\theta_A$	-77.47	-77.485
$\theta_B$	-35.21	-35.16

4.2 ตัวอย่างการคำนวณเทียบกับค่าที่คำนวณที่ได้จากโปรแกรมในพื้นที่การทำงานที่ 2



ภาพที่ 4.5 แสดงการหาค่า  $\theta_A$  และ  $\theta_B$  ในพื้นที่ที่ 2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากสามเหลี่ยม abc เราจะสามารถหาค่ามุม A ได้จากสมการต่อไปนี้

$$\text{มุม}A = \tan^{-1} \frac{Y_0}{X_0} = \tan^{-1} \left| \frac{4}{6} \right| = 33.69$$

จากมุมภายในสามเหลี่ยมมุมฉาก abc มีค่าเท่ากับ 180 องศา ดังนั้นเราจะได้อ่า

$$\text{มุม}B = 180 - 90 - \text{มุม}A = 56.31$$

และค่า  $r_0 = \sqrt{X_0^2 + Y_0^2} = \sqrt{6^2 + 4^2} = 7.2111$  ดังนั้นค่า  $\frac{r_0}{2} = \frac{7.2111}{2} = 3.6$

จากสามเหลี่ยม cde เราจะสามารถหาค่ามุม C ได้จากสมการต่อไปนี้ โดยที่  $L_A=10$

$$\text{มุม}C = \sin^{-1} \frac{\overline{cd}}{L_A} = \sin^{-1} \frac{3.6}{10} = 21.13$$

เนื่องจาก  $L_A$  และ  $L_B$  มีค่าเท่ากับ 10 ทำให้สามเหลี่ยม ace เป็นสามหน้าจั่วทำให้

$$\text{มุม}D = \text{มุม}C = 21.13 \text{ องศา}$$

จากมุมภายในสามเหลี่ยมมุมฉาก cde มีค่าเท่ากับ 180 องศา ดังนั้นเราจะได้อ่า

$$\phi = 180 - 90 - \text{มุม}C = 180 - 90 - 21.13 = 68.87$$

มุมA เป็นมุมแย้ง กับ มุมE ดังนั้น มุมE = 33.69

$$\text{ดังนั้นเราจะได้อ่ามุม } \alpha = \theta_B = 180 - 68.87 - \text{มุม}E = 180 - 68.87 - 33.69 = 77.44$$

จากมุมภายในสามเหลี่ยมมุมฉาก aef มีค่าเท่ากับ 180 องศา ดังนั้นเราจะได้อ่า

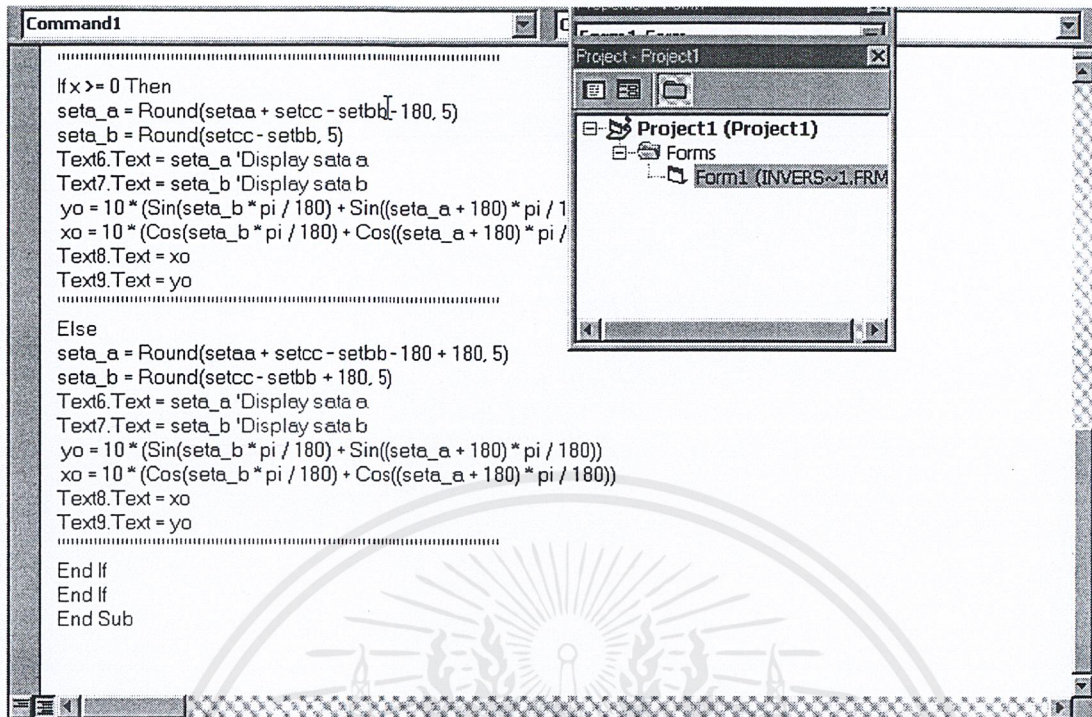
$$\text{มุม}F = 90 - \alpha = 90 - 77.44 = 12.56$$

มุมG เป็นมุมแย้ง กับ มุม F ดังนั้น มุมG = 12.56

$$\text{และจะได้อ่า } \theta_A = 90 - \text{มุม}C - \text{มุม}D - \text{มุม}G = 90 - 21.13 - 21.13 - 12.56 = 35.18$$

$$\text{จะได้อ่ามุม } \beta = 180 + \theta_A - \alpha = 180 - 77.47 + 35.21 = 137.74$$

จากการคำนวณข้างบนจะได้อ่า  $\theta_A$  และ  $\theta_B$  นี้จะนำมาเทียบกับค่า  $\theta_A$  และ  $\theta_B$  ที่ได้จาก โปรแกรมดังนี้



ภาพที่ 4.6 แสดงโปรแกรมการคำนวณในพื้นที่ที่ 2

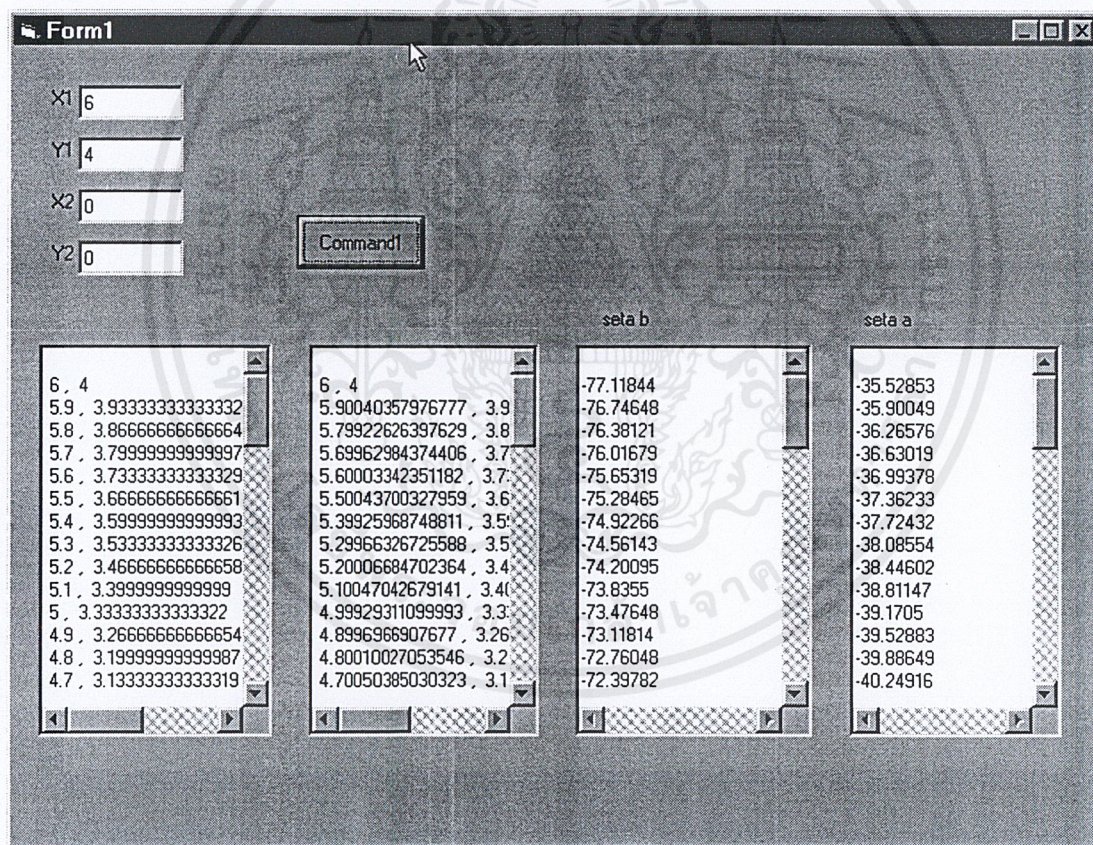
X0	-6	Command1
Y0	4	
$\beta$	137.67600105782	
$\phi$	68.8380005289101	
$\alpha$	-33.6765127486334	
$\theta_A$	35.16149	
$\theta_B$	77.48549	
$X_0$	-6.00505349398925	
$Y_0$	3.9924096757373	

ภาพที่ 4.7 แสดงค่าที่คำนวณได้จากโปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.2 เปรียบเทียบค่าที่คำนวณได้กับค่าที่ได้จากProgram

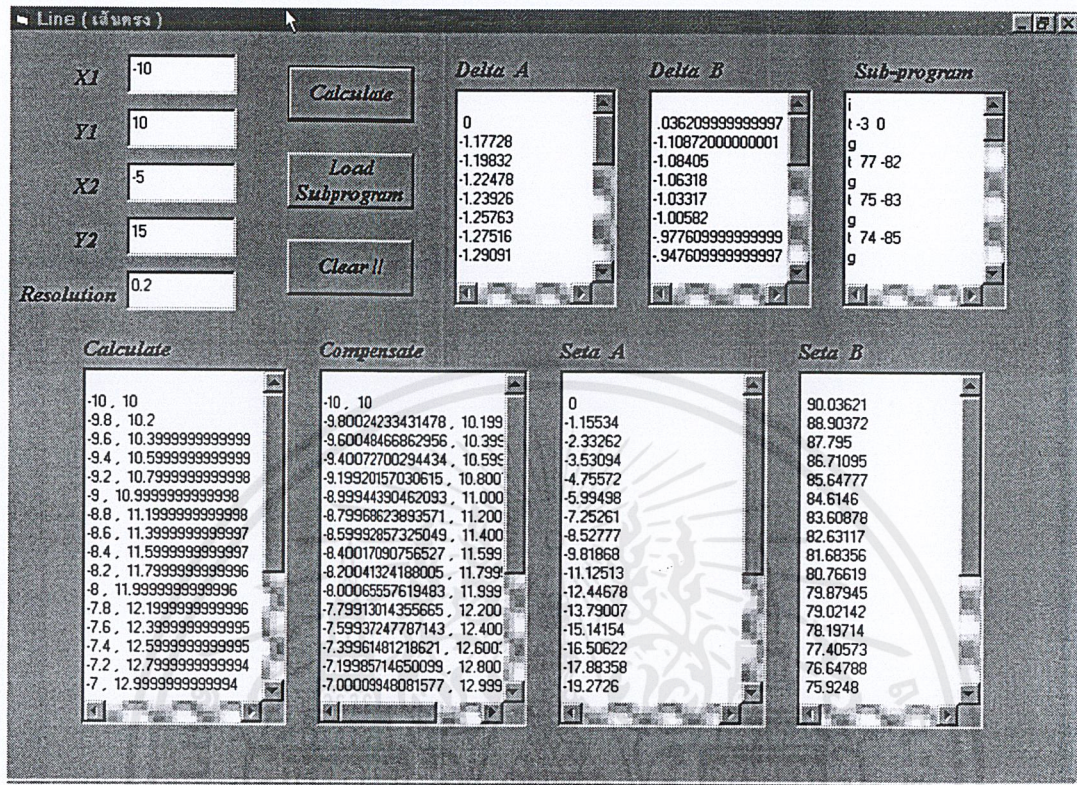
	ค่าที่ได้จากการคำนวณ	ค่าที่ได้จาก Program
$\beta$	137.74	137.67
$\phi$	68.87	68.83
$\alpha$	77.44	33.67
$\theta_A$	35.18	35.16
$\theta_B$	77.44	77.48



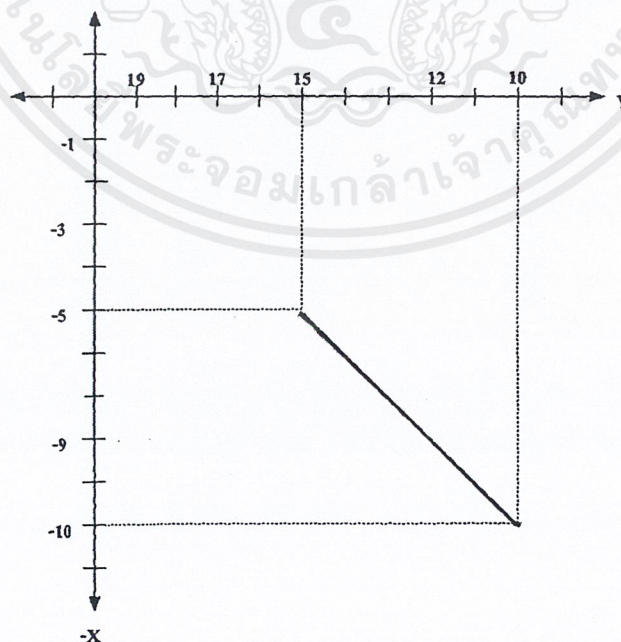
ภาพที่ 4.8 แสดงโปรแกรมการคำนวณเกี่ยวกับเรขาคณิต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 4.3 การทดลองPlot เส้นตรง จากจุด $(-10,10)$ ไปยังจุด $(-5,15)$



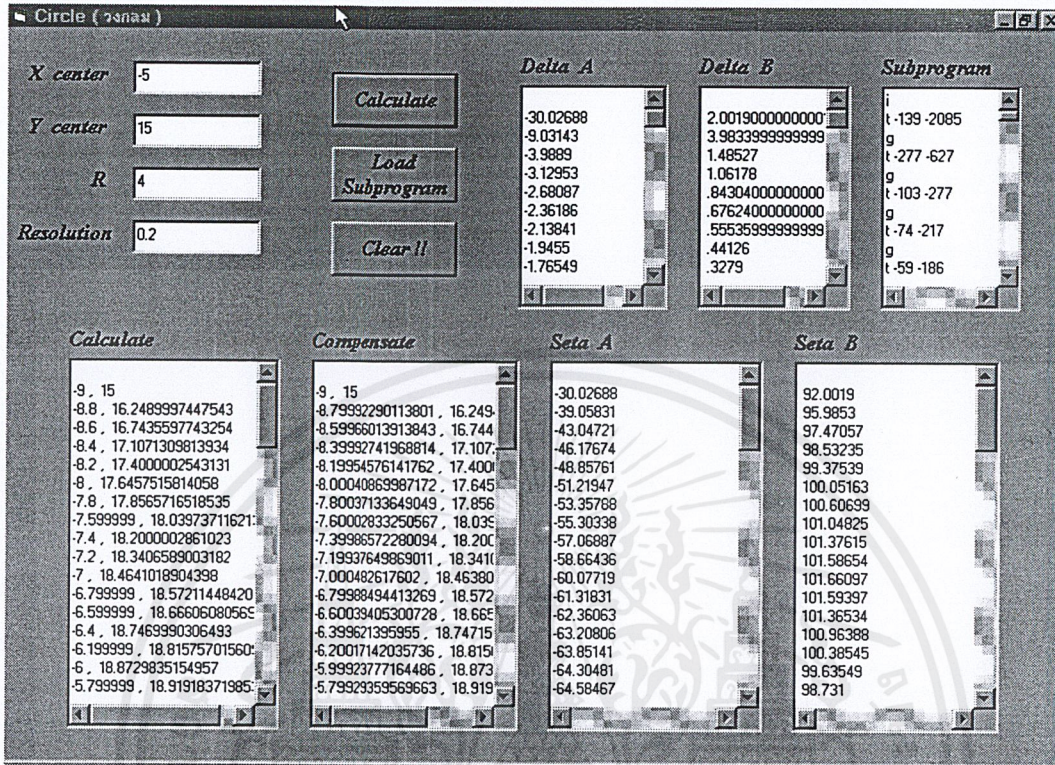
ภาพที่ 4.9 แสดงProgram การทดลองPlot เส้นตรง จากจุด  $(-10,10)$  ไปยังจุด  $(-5,15)$



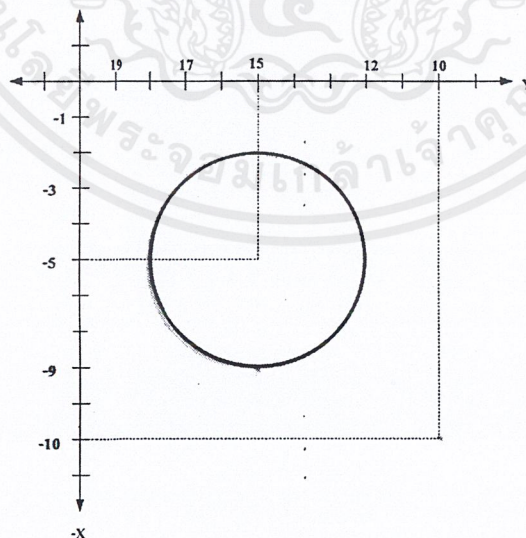
ภาพที่ 4.10 แสดงการเคลื่อนที่ของ แขนก Plot เส้นตรงจากจุด  $(-10,10)$  ไปยังจุด  $(-5,15)$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 4.4 การทดลองPlot วงกลม รัศมีเท่ากับ 3 โดยมีจุดศูนย์กลางที่จุด (-5,15)



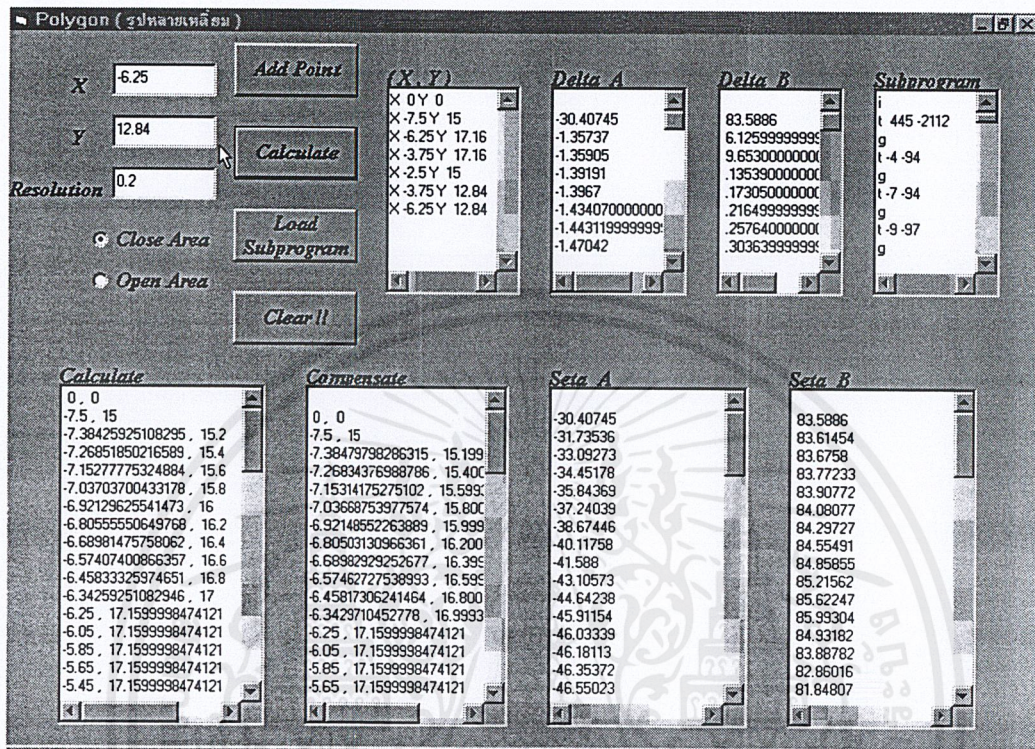
ภาพที่ 4.11 แสดงProgram การทดลองPlot วงกลม รัศมีเท่ากับ 3 โดยมีจุดศูนย์กลางที่จุด (-5,15)



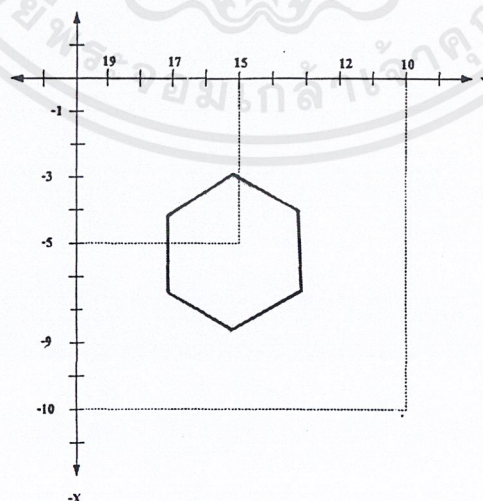
ภาพที่ 4.12 แสดงการเคลื่อนที่ของแกนPlotวงกลมรัศมีเท่ากับ 3 โดยมีจุดศูนย์กลางที่จุด (-5,15)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.5 การทดลองPlot หกเหลี่ยม ที่จุด(-7.5,15), (-6.25,17.16), (-3.75,17.16), (-2.5,15), (-3.75,12.84), (-6.25,12.84)



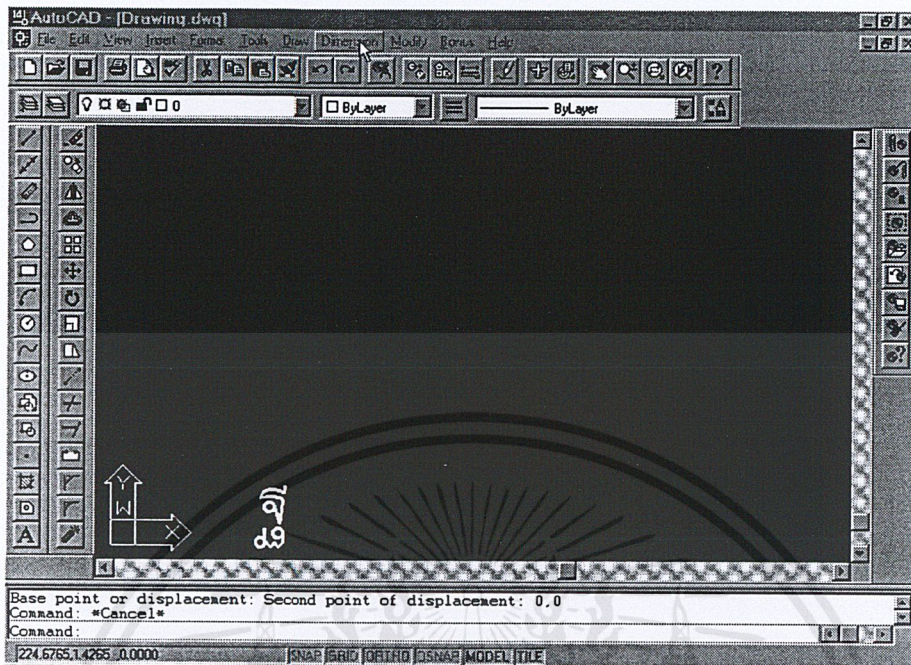
ภาพที่ 4.13 แสดงProgram การทดลองPlotหกเหลี่ยม ที่จุด(-7.5,15), (-6.25,17.16), (-3.75,17.16), (-2.5,15), (-3.75,12.84), (-6.25,12.84)



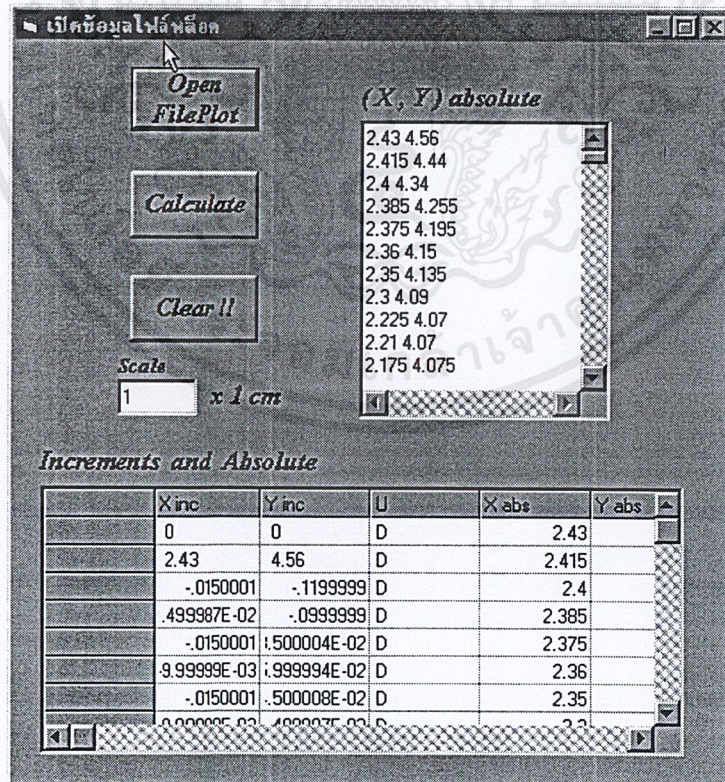
ภาพที่ 4.14 แสดงการเคลื่อนที่ของ แขนกพล Plotหกเหลี่ยม ที่จุด(-7.5,15), (-6.25,17.16), (-3.75,17.16), (-2.5,15), (-3.75,12.84), (-6.25,12.84)

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้สำหรับใช้ในกิจการศึกษเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

#### 4.6 การทดลองPlot ตัวอักษร “ฐ” โดยรับ File Plot จากAuto CAD

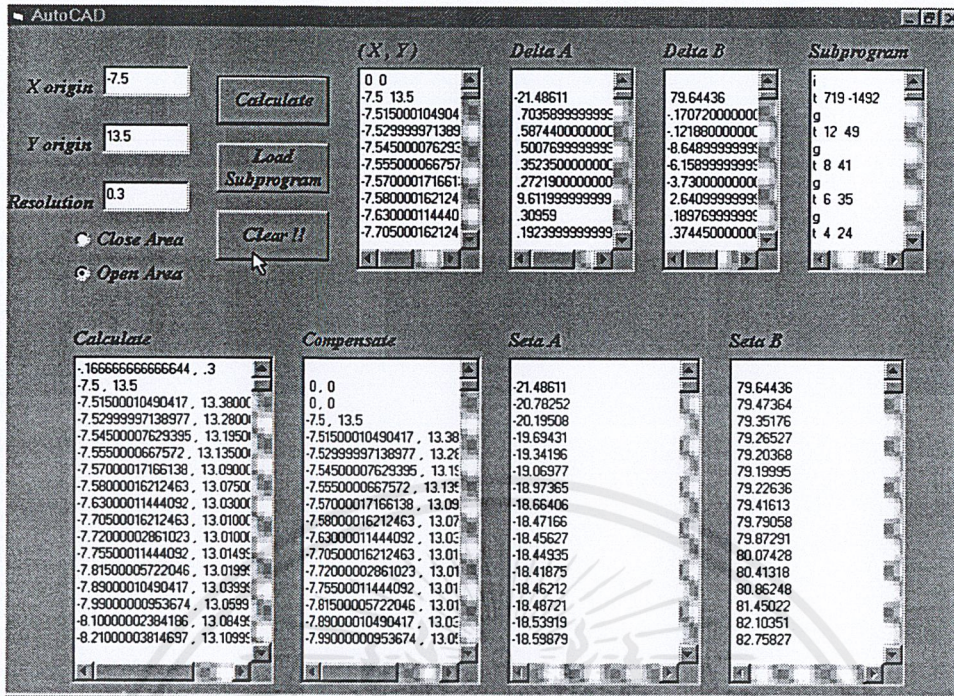


ภาพที่ 4.15 แสดง ตัวอย่างการเขียนตัวอักษรผ่านทาง AutoCAD

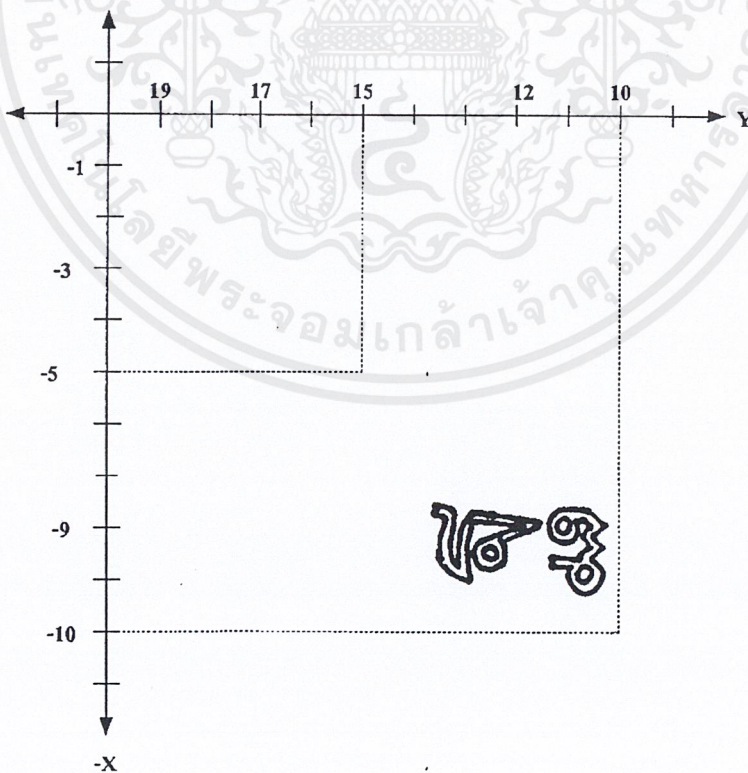


ภาพที่ 4.16 แสดงProgram ที่ทำการรับ FilePlot จาก AutoCAD

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

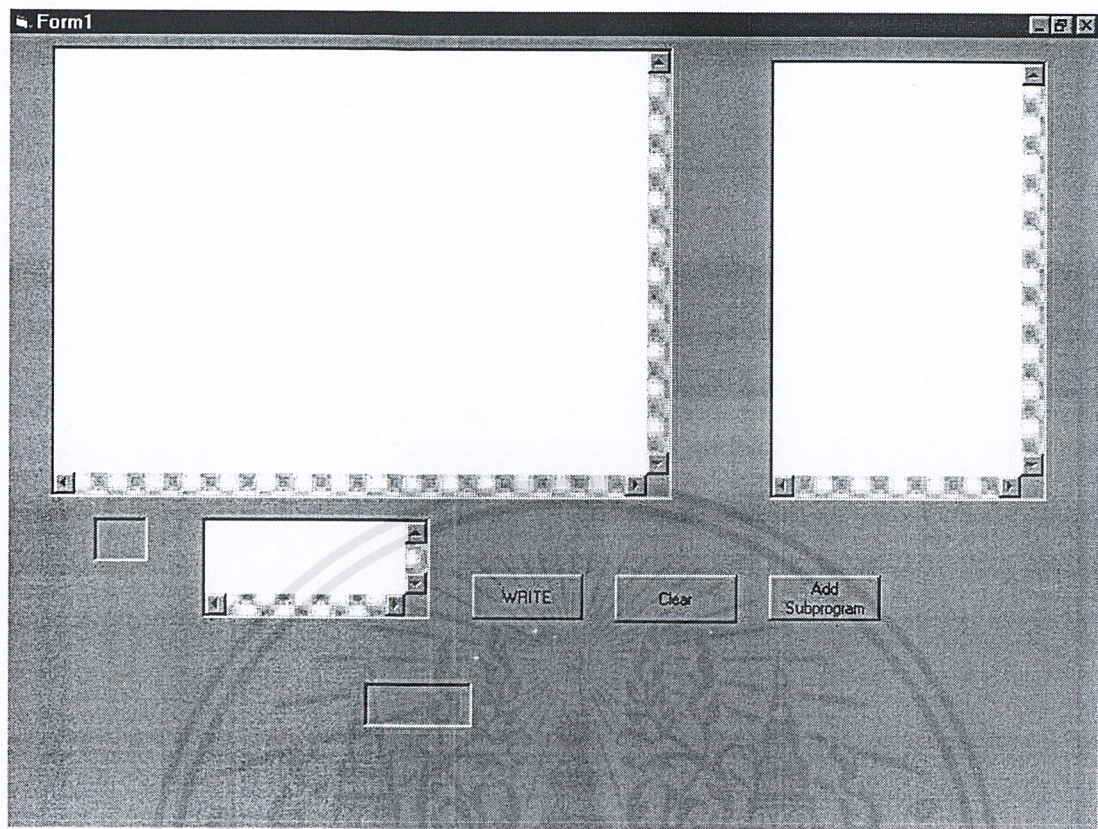


ภาพที่ 4.17 แสดงProgram ที่ทำการรับเวกเตอร์จาก FilePlot มาคำนวณแล้วทำการแปลงเป็นโปรแกรมย่อยส่งให้ตัว Motion Control



ภาพที่ 4.18 แสดงผลการของแกนกลที่รับค่าทาง AutoCAD

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 4.19 แสดงโปรแกรมการติดต่อสื่อสารกับตัวควบคุมการเคลื่อนที่



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บทที่ 5

# สรุปผลการวิจัยและข้อเสนอแนะ

### 5.1 ปัญหาและอุปสรรค

1. ต้องปรับแบบที่ได้ออกแบบไว้ให้เข้ากับสิ่งที่มีอยู่ในท้องตลาด ที่ออกแบบไว้เป็น Aluminums 7075 แต่ไม่สามารถหาซื้อได้จึงจำเป็นต้องเปลี่ยนเป็น Aluminums 5083
2. ขาดประสบการณ์ ในการใช้เครื่องมือ และ เครื่องจักร
3. ขาดประสบการณ์ ในการออกแบบ ทำให้เกิดความยุ่งยากในการทำชิ้นงาน เช่นการบอกขนาดแบบโดยไม่อ้างอิงจากมุมใดมุมหนึ่งทำให้ไม่สามารถกำหนดจุดอ้างอิงที่แน่นอนได้
4. อุปกรณ์ ไม่พร้อม เช่น ต้องรอการย้ายเครื่องมือจากห้องเก่าไม่มีไฟฟ้าป้อนเข้าเครื่อง ไม่มีดอกสว่าน ดอกทำเกลียว ประแจในขนาดที่ต้องการและอุปกรณ์ทำความสะอาดไม่เพียงพอ
5. เกิดความผิดพลาดในการทำชิ้นงานแต่ละชิ้น เนื่องจากเครื่องมือไม่สามารถแสดงค่าในการเคลื่อนที่ในแนวแกน x ได้ทำให้ต้องใช้การตัดสินใจซึ่งก่อให้เกิดความผิดพลาดได้ ดังนั้นเมื่อนำมาประกอบกันจึงไม่สามารถนำชิ้นงานใดมาเป็นระดับอ้างอิงได้
6. เกิดปัญหาการลั่นที่จุดปลายของแขนกลเนื่องจากการลั่นของตัวมอเตอร์ทำให้ไม่สามารถใช้ความเร็วในการเคลื่อนที่ที่มากได้
7. เกิดปัญหาในเรื่องของพื้นที่ในการทำงานที่เกิดมาจากฐานรองมอเตอร์และตัวฐานไปจำกัดการเคลื่อนที่ของแขนกลเอง
8. เกิดปัญหาในการติดตั้งชิ้นงานเนื่องมาจากข้อผิดพลาดในการออกแบบ
9. เนื่องจากเรานำมอเตอร์ขนาด Step 400 Step / รอบ มาใช้งานทำให้ความละเอียดในการเคลื่อนที่ของแขนกลนั้นหยาบ เราจึงต้องเปลี่ยนมอเตอร์เป็นขนาด 58000 Step / รอบ แต่เลือกใช้แค่ 25000 Step / รอบ
10. การตั้งจุด zero ไม่สามารถตั้งได้อย่างเที่ยงตรงทุกครั้ง เนื่องจากไม่มีจุดอ้างอิงเพื่อระบุตำแหน่งของจุดเริ่มต้น
11. ขอบเขตการเคลื่อนที่เนื่องจากความยาวของแขน ส่งผลถึงพื้นที่การทำงาน
12. โครงสร้างส่วนฐานยึดมอเตอร์ ( Motor Base) และ ฐานด้านหลัง (Base) ขวางการเคลื่อนที่ของแขนกล และ อุปกรณ์ที่ติดตั้งอยู่ที่ส่วนปลายของแขนกล
13. Error สะสมจากขนาดของขนาดแขน และ โครงสร้างที่ไม่ได้ขนาดตามที่ออกแบบ รวมถึงการตั้งจุด zero ที่ไม่สามารถอ้างอิงได้
14. Error จากการปิดเศษในการคำนวณทาง software

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 5.2 ข้อเสนอแนะและแนวทางการพัฒนา

1. ควรปรับปรุงโครงสร้างเพื่อให้แขนกลมีพื้นที่ในการทำงานได้เต็มที่
2. ควรสร้างให้ได้ตรงตามขนาด เนื่องจากขนาดที่ผิดพลาดมีผลต่อโครงสร้างโดยรวมของแขนกลทำให้ไม่สามารถควบคุมการเคลื่อนที่ได้ในบางจุด
3. ควรเขียนเงื่อนไขให้ครอบคลุมพื้นที่ที่แขนกลไม่เคลื่อนที่ไปยังจุดที่ไม่ต้องการหรืออาจเกิดอันตรายจากการเคลื่อนที่ไปยังจุดนั้นๆ
4. แขนกลนี้สามารถนำไปพัฒนาประยุกต์ใช้กับงานที่ควบคุมทางเดิน หรือ ตำแหน่งต่อเป็นเครื่องมืออย่างง่ายๆ เช่น เครื่องเชื่อม เครื่องตัด หรือการจับชิ้นงาน ได้
5. เพิ่มแกน Z เพื่อให้อุปกรณ์ที่ติดตั้งอยู่ที่จุดปลายแขนกลสัมผัสกับชิ้นงานเฉพาะทางเดินที่ต้องการ และสามารถควบคุมระดับในแนวดิ่ง

## 5.3 สรุป

จากการศึกษาการทำงานของแขนกลแบบต่างๆ ที่ได้รวบรวมมาจากแหล่งข้อมูลจากที่ต่างๆ แล้วนำมาออกแบบสร้างแขนกลแบบขับเคลื่อนเพื่อเป็นกรณีศึกษา เนื่องจากง่ายต่อการศึกษาค้นคว้าและสามารถนำมาสร้างได้จริง โดยแบ่งการศึกษาออกเป็น 3 ส่วนใหญ่ดังนี้

1. ส่วนโครงสร้าง (Hardware) ได้สร้างแขนกลตามแบบของ M.I.T Direct Drive Robot Model IV (ตามรูปที่ ) ด้วยอลูมิเนียม เบอร์ 7075 และเบอร์ 5083 ติดตั้งอยู่บนแผ่นพลาสติกใส
2. ส่วนการควบคุมการเคลื่อนที่ (Motion Control) ประกอบไปด้วย
  - a. ส่วนควบคุมการเคลื่อนที่ (Motion Control)
  - b. ส่วน ส ัญ ญา ณ ขั บ ส เต็ ป ี่ ง ม อ เต อ ร์ (Stepping-Motor Drive) รุ่น Compumotor OEM 650 ของ Parker
  - c. ส เต็ ป ี่ ง ม อ เต อ ร์ (Stepping-Motor) รุ่น Compumotor OEM 57-83 ของ Parker
3. ส่วนโปรแกรม ใช้โปรแกรมวิชวลเบสิกสร้างโปรแกรมในการประยุกต์การใช้งาน เพื่อเพิ่มประสิทธิภาพการใช้งาน

จากที่ได้ออกแบบ และ ดำเนินการตามกระบวนการคิดที่ได้วางแผนไว้ เราสามารถที่จะควบคุมแขนกลให้เคลื่อนที่ตามที่เราต้องการได้ แต่ก็ยังไม่สามารถควบคุมได้อย่างสมบูรณ์เนื่องจากมีข้อจำกัดทางด้านโครงสร้าง และ จะมีประสิทธิภาพขึ้นถ้าได้รับปรับปรุงแก้ไข

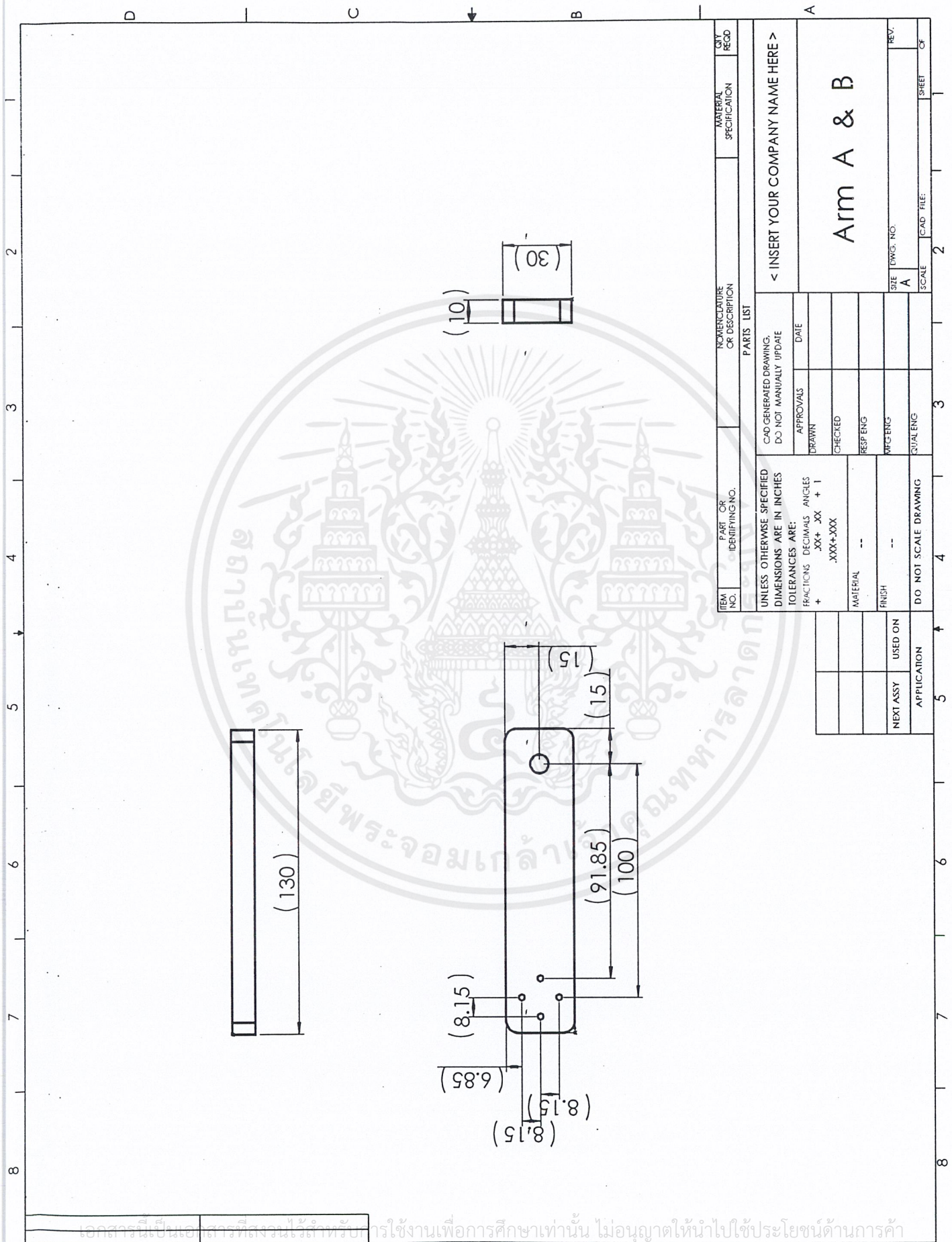
## บรรณานุกรม

- [1] Haruhiko Asada and Kamal Youcef-Toumi, “Direct-Drive Robots *Theory and Practice*.” The MIT Press, Cambridge, Massachusetts, London, England, 1987
- [2] Stadler, Wolfram. “Analytical robotics and mechatronics.” New York: McGraw-Hill, c1995
- [3] EUROPA LEHRMITTEL.“ Tabellenbuch Metall.” Mourney, Vollmer GmbH & Co. Dusselberger StraBe 23 42781 Haan-Gruiten
- [4] จัฑทวุฒิ พีชผล และ พิชิต สันติกุลานนท์, “คู่มือเรียน Visual Basic 6”, กรุงเทพฯ: บริษัท โปรวิชั่น จำกัด, 2542
- [5] กิตติ ภัคดีวัฒน์กุล และ จำลอง ครูอุตสาหะ, “Visual Basic ฉบับ โปรแกรมเมอร์”, กรุงเทพฯ:ไทยเจริญการพิมพ์, 2544
- [6] ลานนา ดวงสิงห์,ผศ. , “COMPUTER:เทคโนโลยีคอมพิวเตอร์และสารสนเทศ”, กรุงเทพฯ: บริษัท เพียร์สัน เอ็ดดูเคชั่น อิน โคไชน่า จำกัด, 2543
- [7] กฤษดา ใจเย็น,อรรรถพล บุญยะโกคา, ชัยวัฒน์ ลิ้มพรวิไล, “เรียนรู้และปฏิบัติการเชื่อมต่อคอมพิวเตอร์กับอุปกรณ์ภายนอกผ่านพอร์ตอนุกรม,บริษัท อิน โนเวตีฟ เอ็กเพอริเมนต์ จำกัด



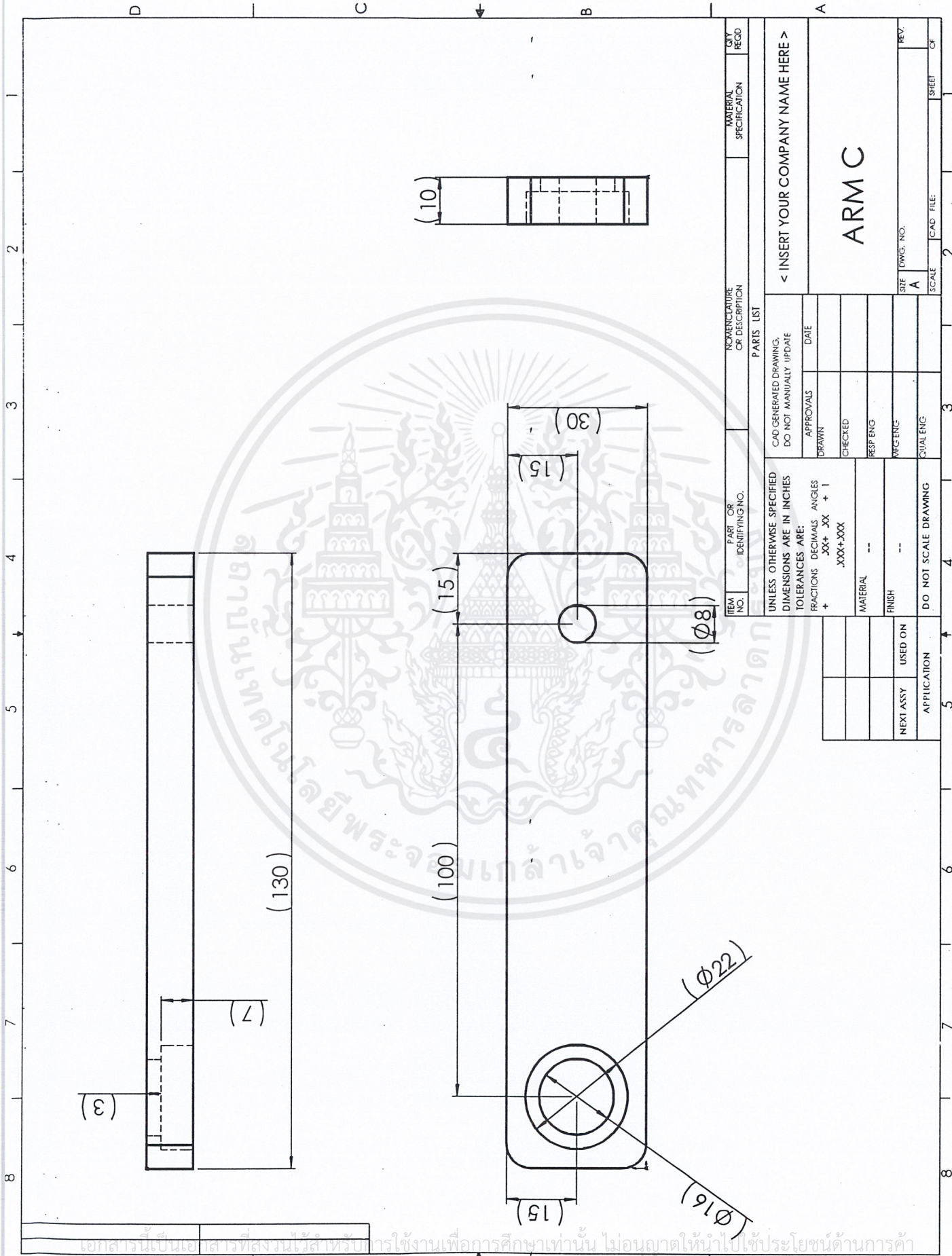
**ภาคผนวก ก. แบบ Drawing**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ITEM NO.	PART OR IDENTIFYING NO.	NUMBER/QUANTITY OR DESCRIPTION	MATERIAL SPECIFICATION	CAD FILE
PARTS LIST				
UNLESS OTHERWISE SPECIFIED DIMENSIONS ARE IN INCHES				
TOLERANCES ARE:				
FRACTIONS	DECIMALS	ANGLES		
+	.XX+ .XX	+ 1		
	.XXX+XXX			
MATERIAL	--			
FINISH	--			
DO NOT SCALE DRAWING				
APPROVALS	DATE	< INSERT YOUR COMPANY NAME HERE >		
DRAWN				
CHECKED				
RESP'G				
MFG'G		Arm A & B		
QUAL'G		SIZE	DWG. NO.	REV.
		A		
		SCALE	CAD FILE:	SHEET
				OF

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้ภายในเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

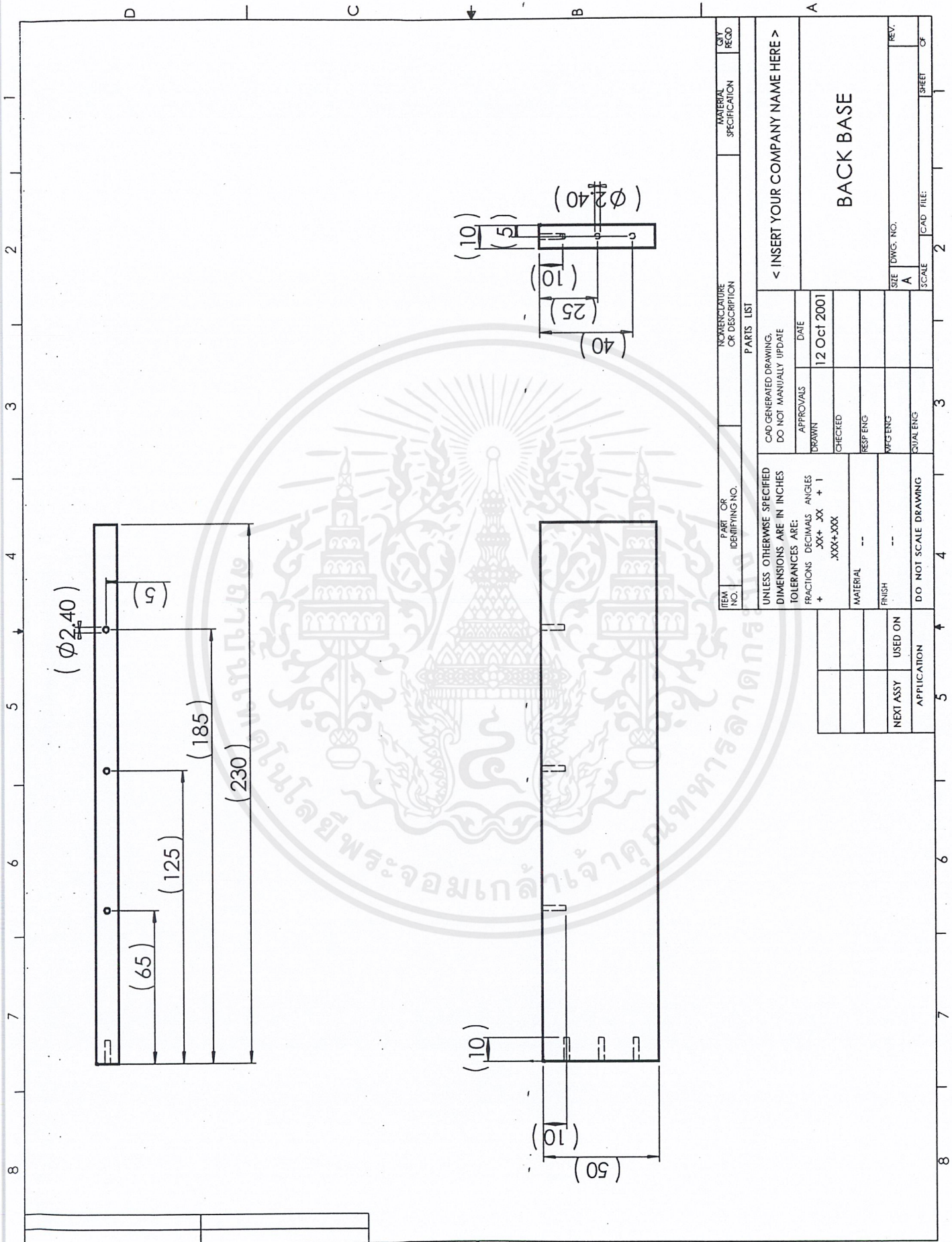


ITEM NO.		PART OR IDENTIFYING NO.		NOMENCLATURE OR DESCRIPTION		MATERIAL SPECIFICATION		QTY REQD	
PARTS LIST									
UNLESS OTHERWISE SPECIFIED DIMENSIONS ARE IN INCHES TOLERANCES ARE: FRACTIONS DECIMALS ANGLES + .XX + .XX + 1 + .XXX + .XXX									
DRAWN		CHECKED		RES'P ENG		MFC ENG		CVAL ENG	
DATE									
CAD GENERATED DRAWING. DO NOT MANUALLY UPDATE									
APPROVALS									
DATE									
< INSERT YOUR COMPANY NAME HERE >									
ARM C									
NEXT ASSY		USED ON		APPLICATION		DO NOT SCALE DRAWING		SCALE	
								A	
								REV:	
								SIZE DWG. NO.	
								SHEET	
								OF	

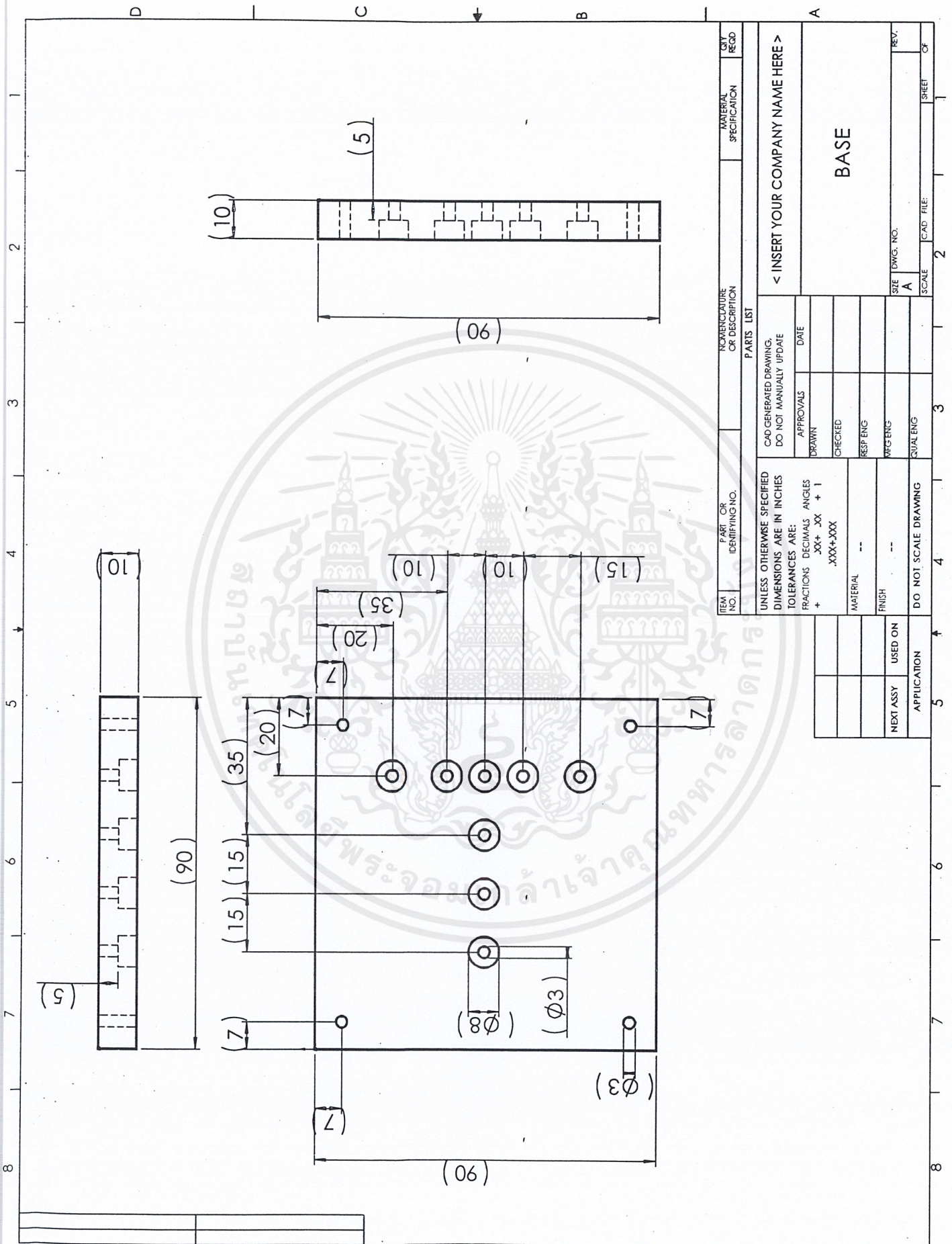
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 ไม่ว่การณใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



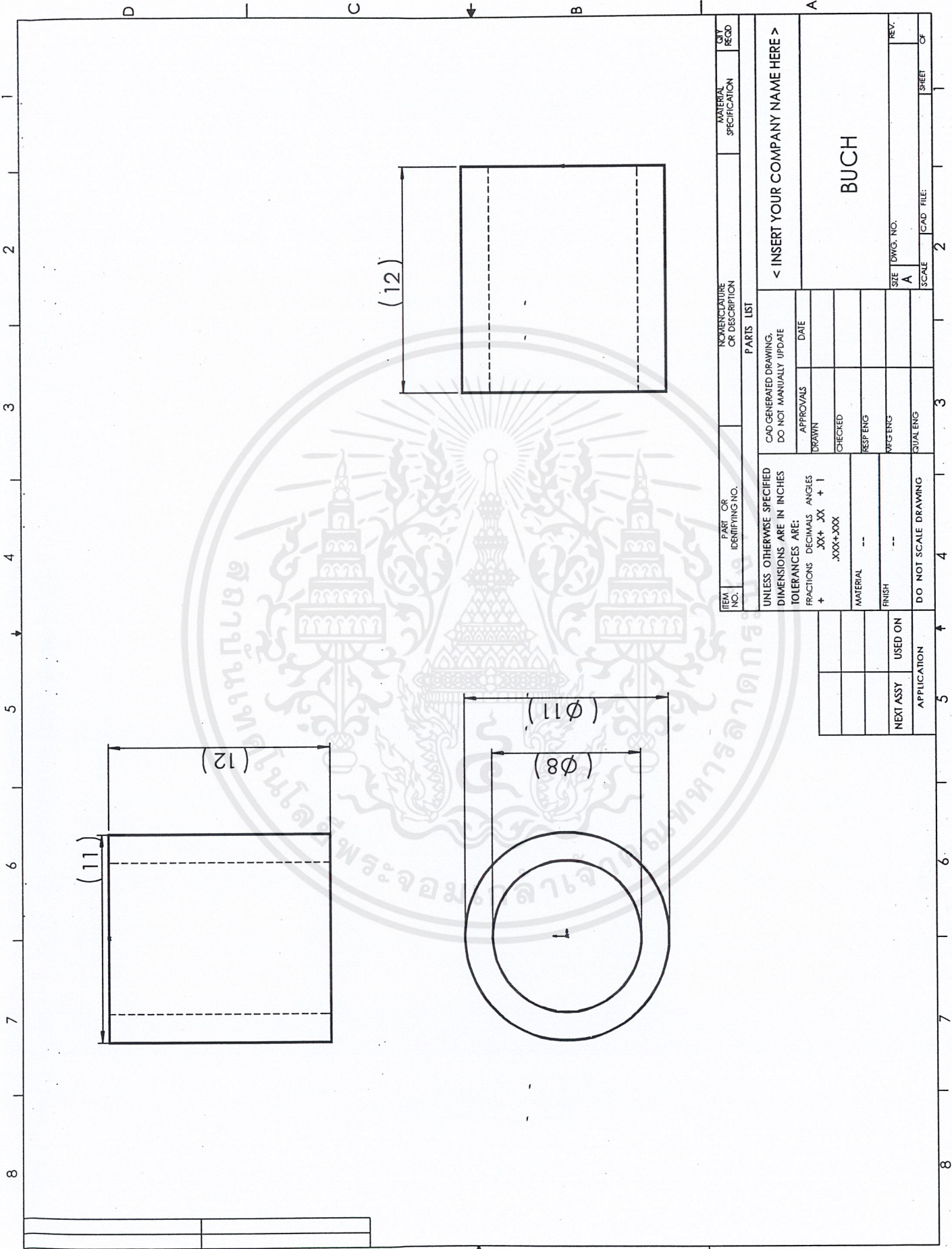




เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

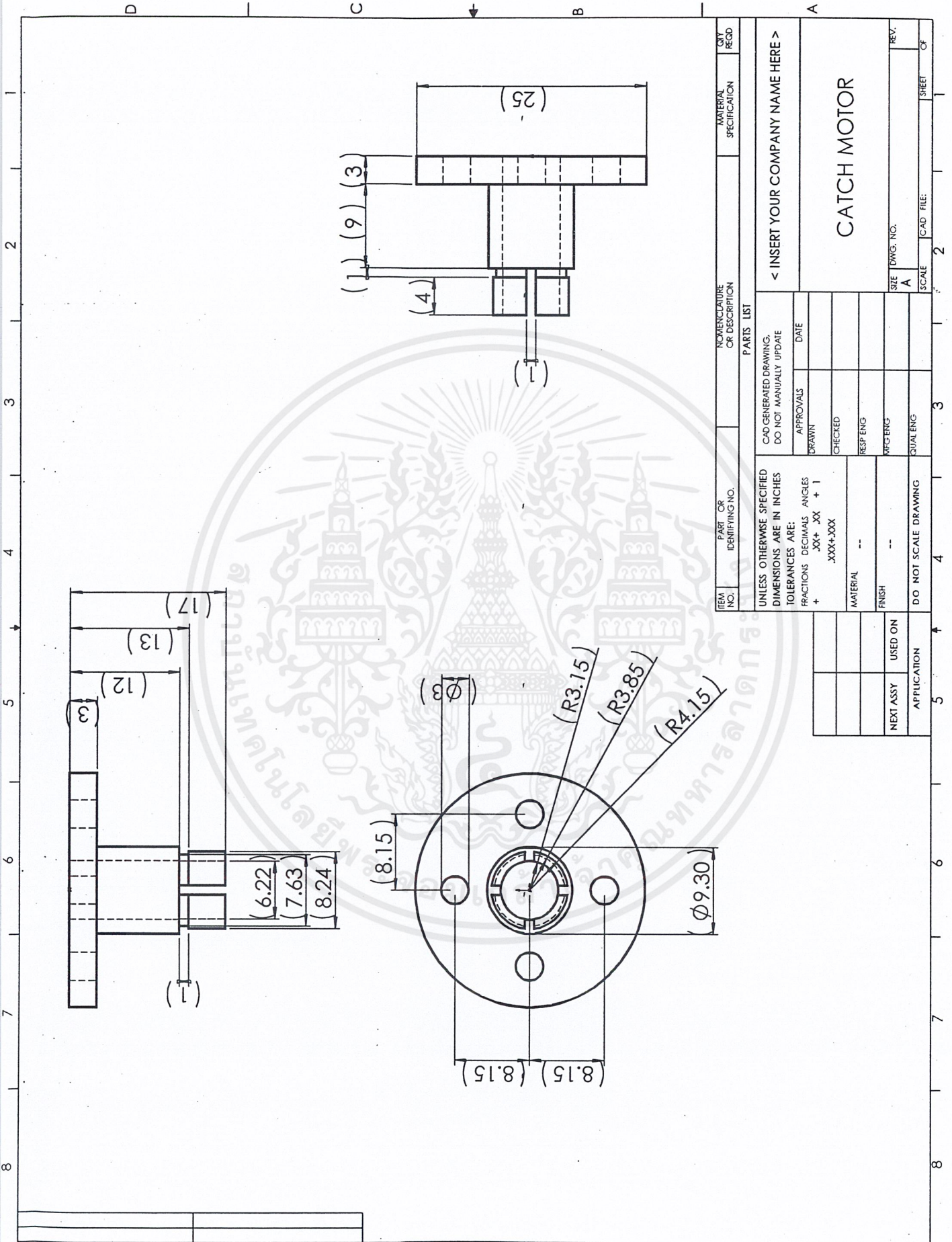


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



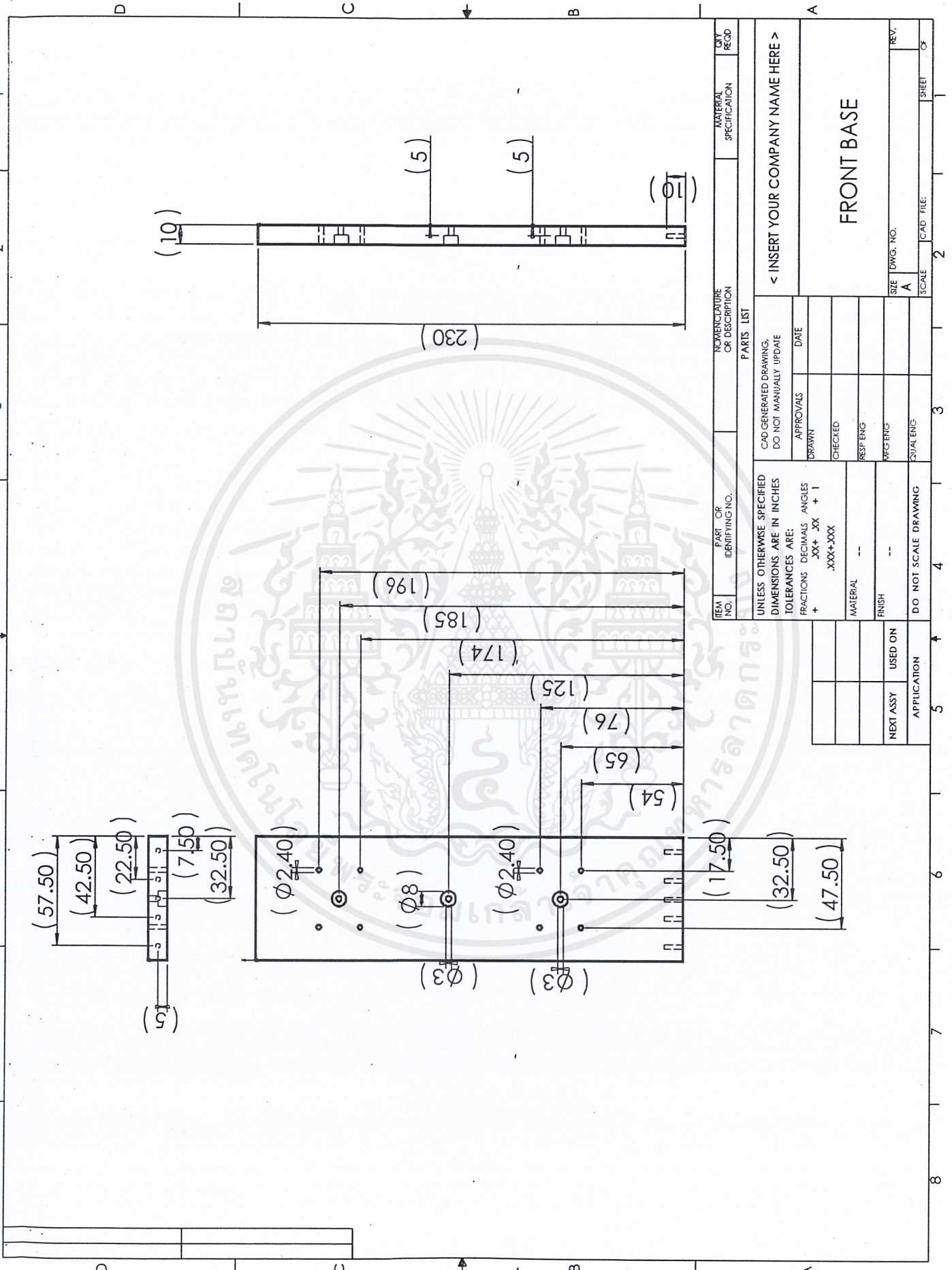
ITEM NO.	PART OR IDENTIFYING NO.	NUMERATURE OR DESCRIPTION	MATERIAL SPECIFICATION	QTY REQD
PARTS LIST				
UNLESS OTHERWISE SPECIFIED DIMENSIONS ARE IN INCHES		CAD GENERATED DRAWING. DO NOT MANUALLY UPDATE		
TOLERANCES ARE:		APPROVALS	DATE	
FRACTIONS .XX+ .XX + 1		DRAWN		
+ .XXX+XXX		CHECKED		
MATERIAL --		RESP ENG		
FINISH --		FIG ENG		
DO NOT SCALE DRAWING		SJAL ENG		
NEXT ASSY APPLICATION		SCALE	DWG. NO.	
USED ON		A	REV.	
APPLICATION		2	BUCH	
		CAD FILE:	SHEET	
			CF	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

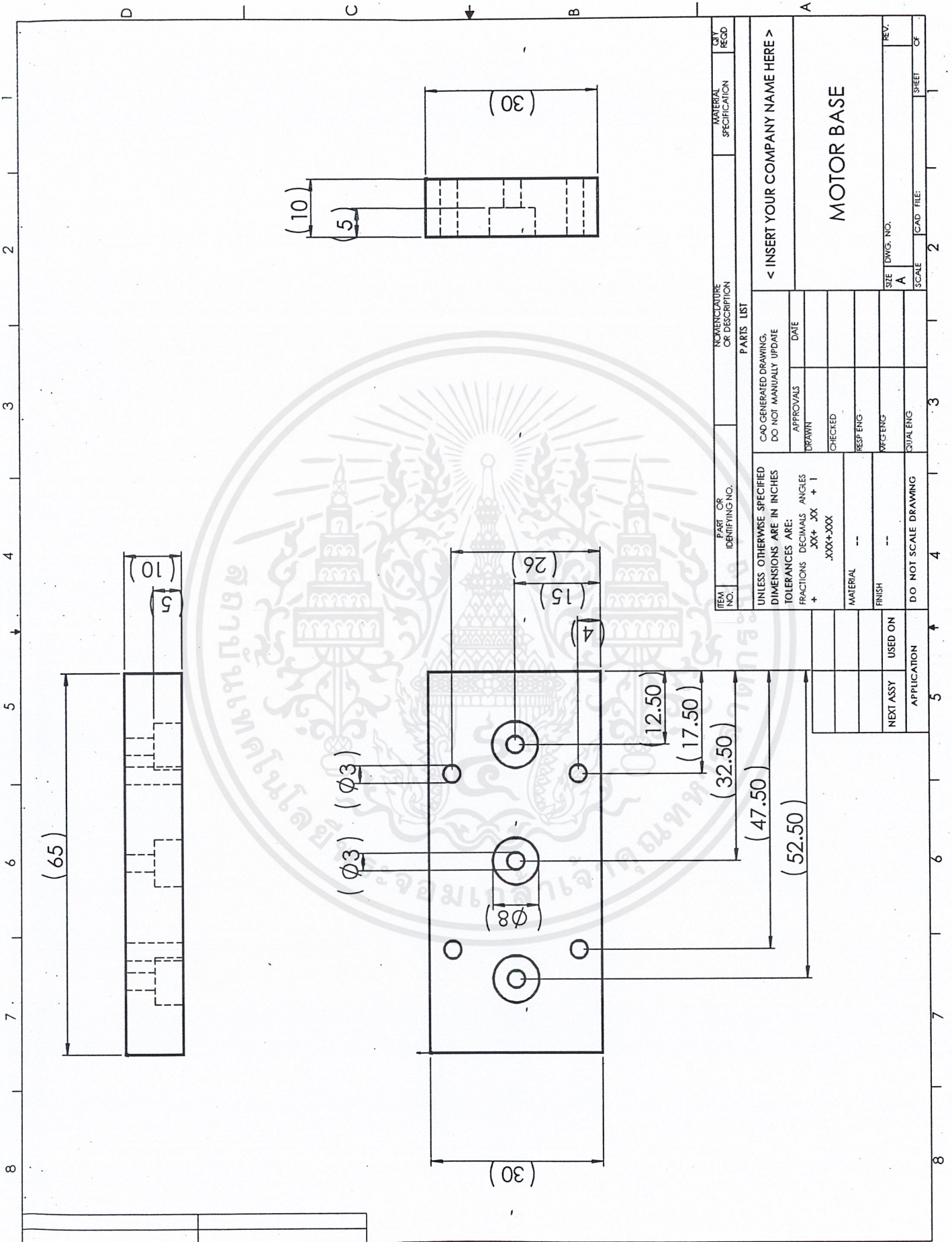


ITEM NO.	PART OR IDENTIFYING NO.	NOMENCLATURE OR DESCRIPTION	MATERIAL SPECIFICATION	QTY REQD
PARTS LIST				
UNLESS OTHERWISE SPECIFIED DIMENSIONS ARE IN INCHES				
TOLERANCES ARE:				
FRACTIONS	DECIMALS	ANGLES		
+ .XX+ .XX	+ 1			
.XXX+XXX				
MATERIAL	---	FINISH	---	---
NEXT ASSY	USED ON	APPLICATION	DO NOT SCALE DRAWING	DO NOT MANUALLY UPDATE
				APPROVALS
				DRAWN
				CHECKED
				RESP ENG
				REC'G ENG
				SUAL ENG
				DATE
< INSERT YOUR COMPANY NAME HERE >				
CATCH MOTOR				
SIZE	DWG. NO.	REV.	SCALE	SHEET
A			2	OF

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

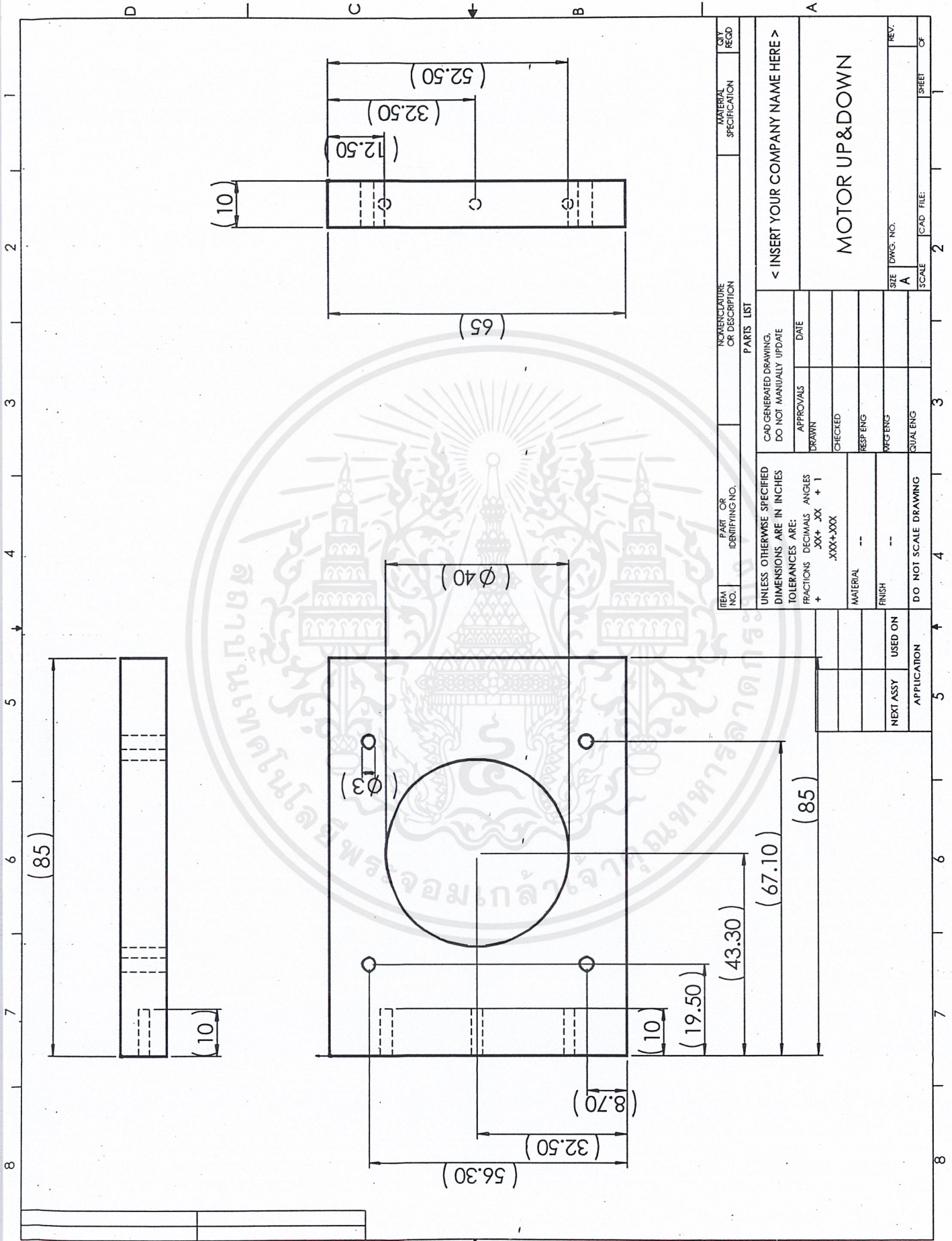


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

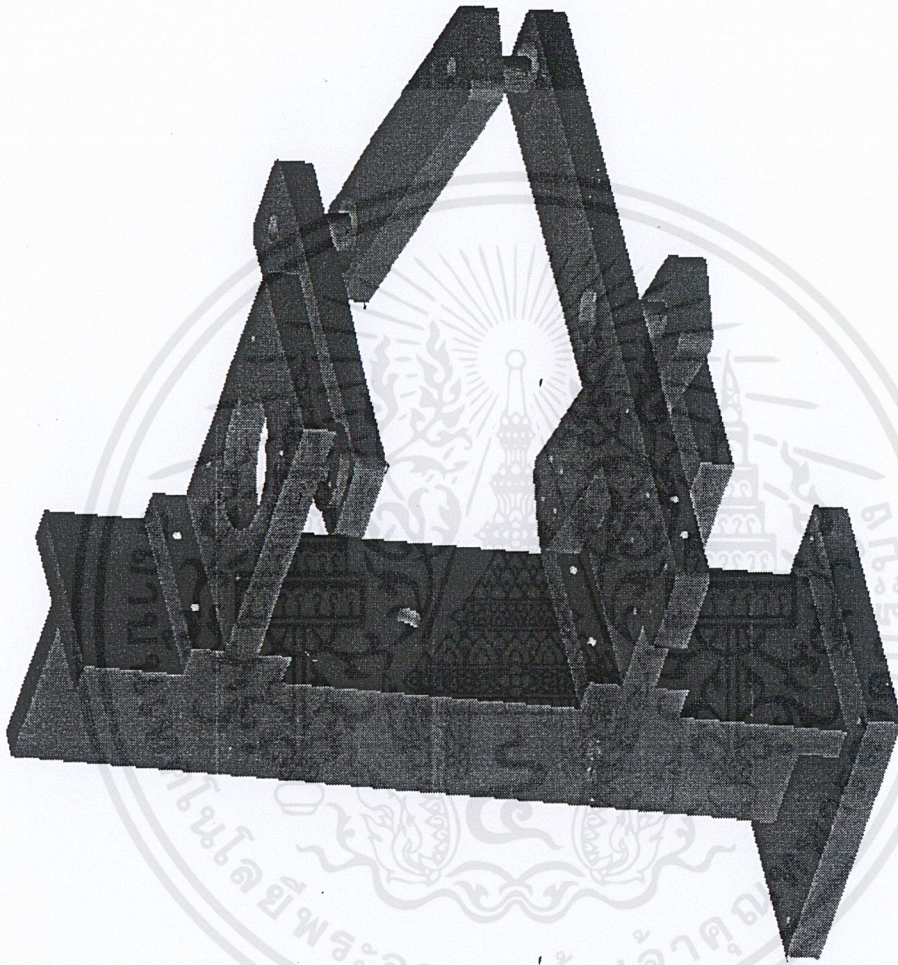


ITEM NO.	PART OR IDENTIFYING NO.	NOMENCLATURE OR DESCRIPTION	MATERIAL SPECIFICATION	CITY REGD.
PARTS LIST				
UNLESS OTHERWISE SPECIFIED DIMENSIONS ARE IN INCHES		CAD GENERATED DRAWING. DO NOT MANUALLY UPDATE		
TOLERANCES ARE:		APPROVALS	DATE	
FRACTIONS DECIMALS ANGLES		DRAWN		
+ .XXX+XXX		CHECKED		
		RESP ENG		
		W/ENG		
		QUAL ENG		
MATERIAL		MOTOR BASE		
FINISH		SIZE DWG. NO.		
DO NOT SCALE DRAWING		A		
NEXT ASSY		SCALE		
APPLICATION		2		
		SHEET		
		OF		

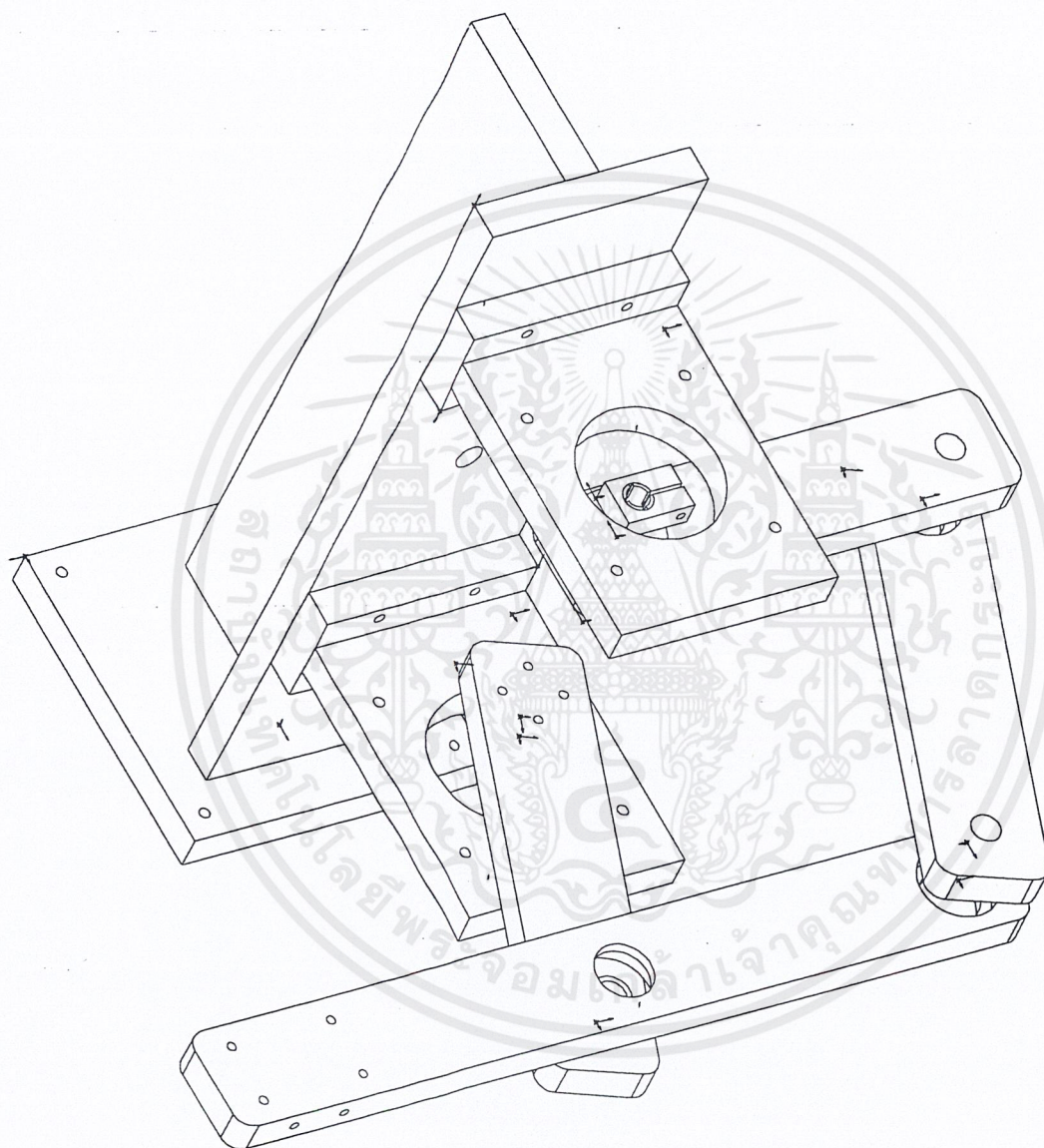
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

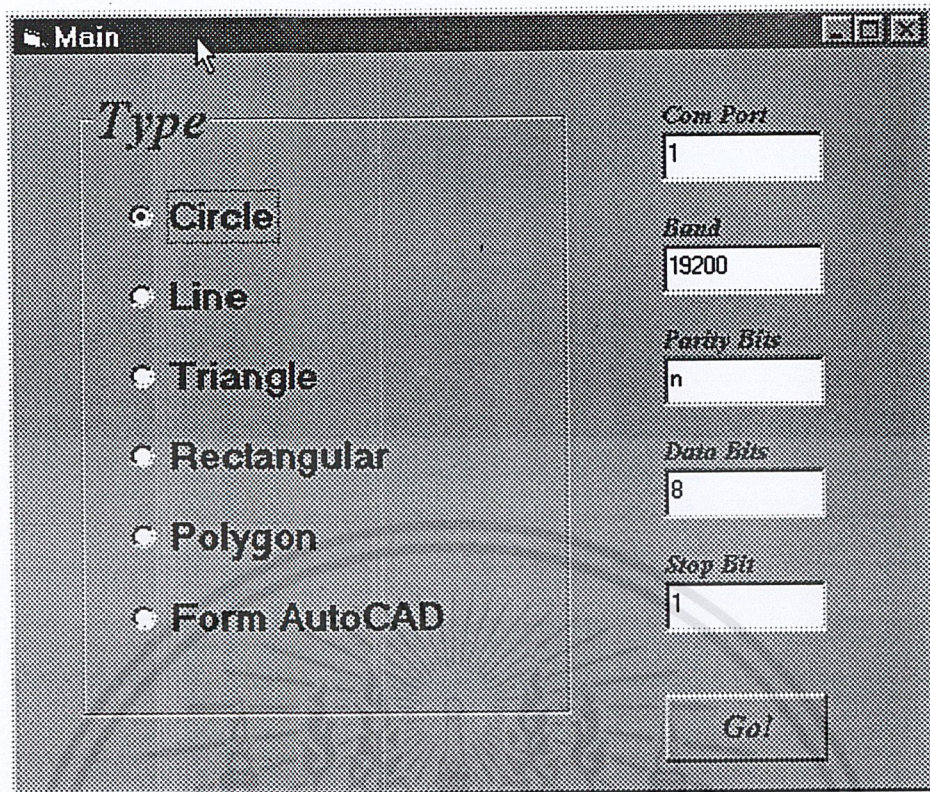


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

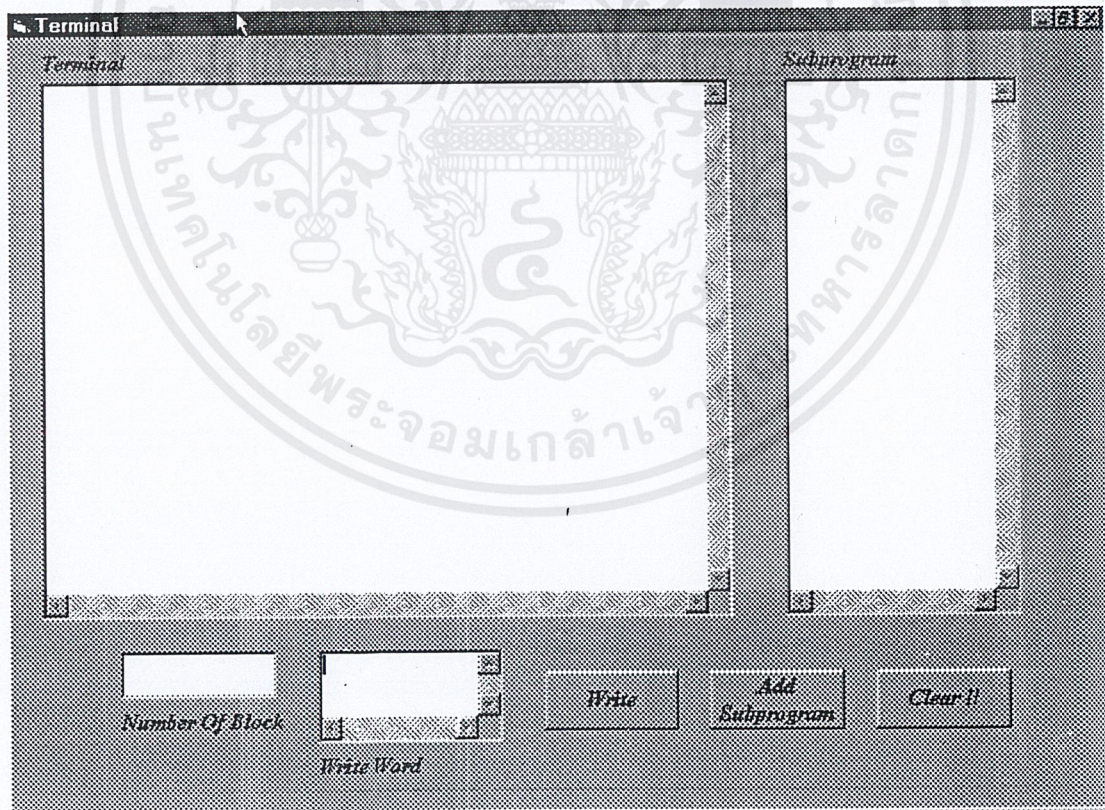


## ภาคผนวก ข. Software Windows

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

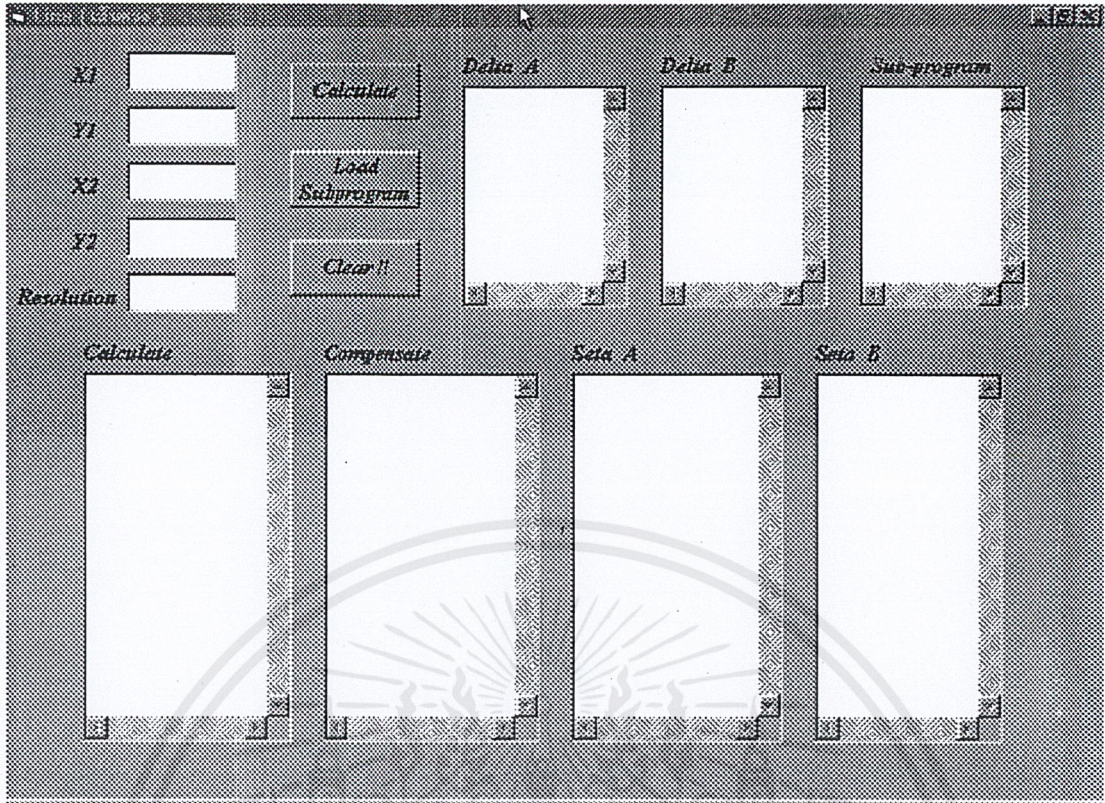


ภาพที่ ข.1 แสดงหน้าจอ Main (หน้าหลัก)

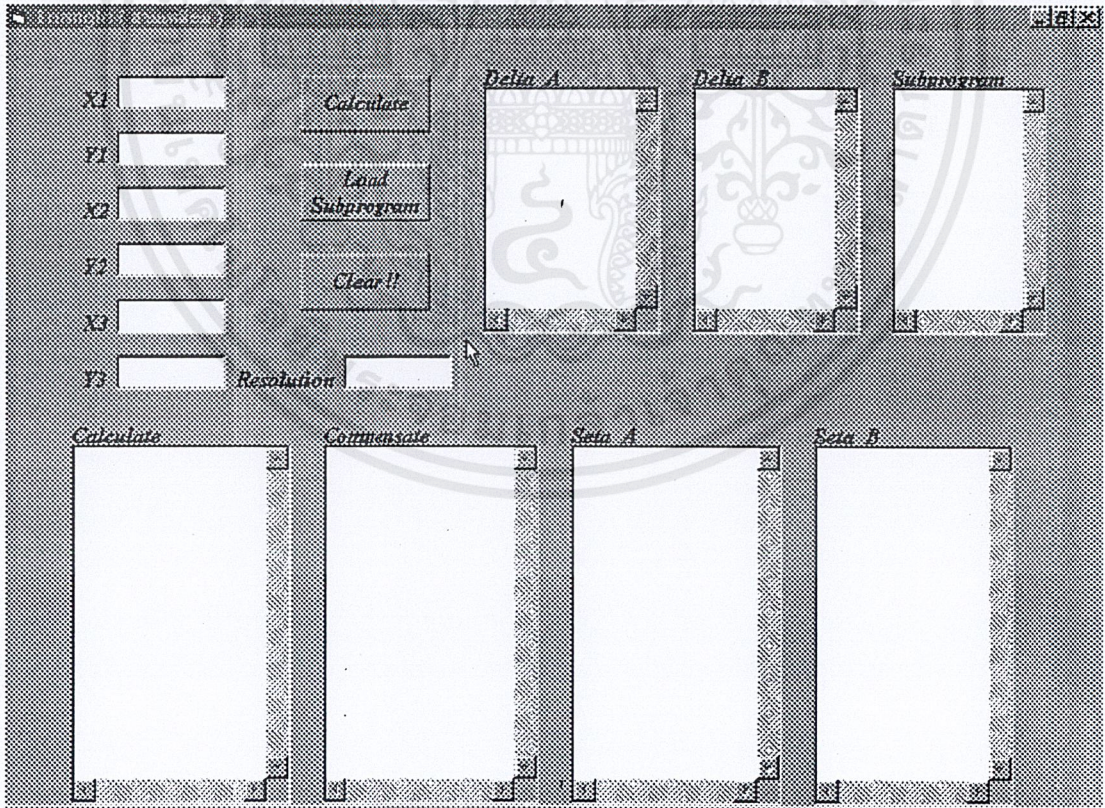


ภาพที่ ข.2 แสดงหน้าจอ Terminal

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

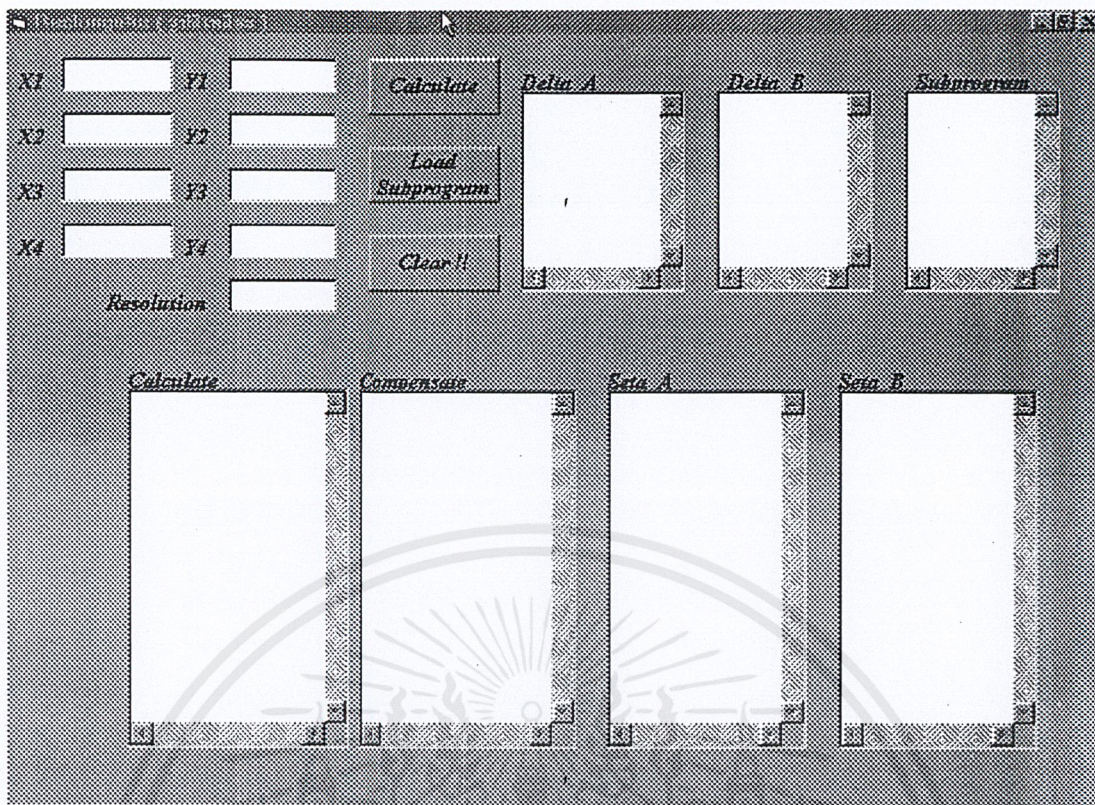


ภาพที่ ข.3 แสดงหน้าจอ Line (รูปเส้นตรง)

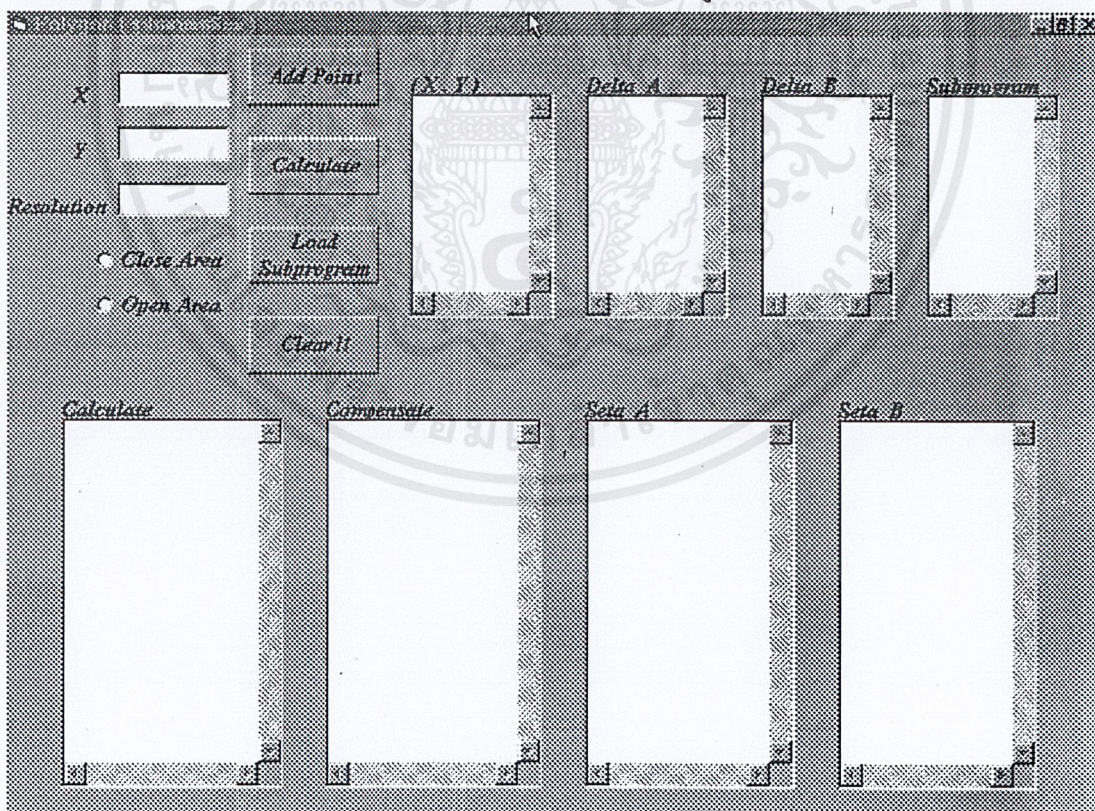


ภาพที่ ข.4 แสดงหน้าจอ Triangle (รูปสามเหลี่ยม)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ ข.5 แสดงหน้าจอ Rectannular (รูปสี่เหลี่ยม)



ภาพที่ ข.6 แสดงหน้าจอ Polygon (รูปหลายเหลี่ยม)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้





# Windows 98 Printer Test Page

Congratulations!

If you can read this information, you have correctly installed your HP LaserJet 6P.

The information below describes your printer driver and port settings.

Printer name: HP LaserJet 6P  
Printer model: HP LaserJet 6P  
Driver name: PCL5EMS2.DRV  
Driver version: 4.00  
Color support: No  
Port name: LPT1:  
Data format: EMF

Files used by this driver:

c:\windows\system\PCL5EMS2.DRV (4.10.1998, GPC 3.02)  
c:\windows\system\PJLMON.DLL (4.10.1998)  
c:\windows\system\UNIDRV.DLL (4.10.1998)  
c:\windows\system\UNIDRV.HLP  
c:\windows\system\ICONLIB.DLL (4.10.1998)  
c:\windows\system\FINSTALL.DLL (4.0.2.01)  
c:\windows\system\FINSTALL.HLP

This is the end of the printer test page.

