สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

แขนกล 3 แกน

SCARA ROBOT

นายนพดล สว่างญาติ
นายนำชัย นาควิทูร
นายสมชัย สายชลไพศาล

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิศวกรรมศาสตรบัณฑิต
สาขาวิชาวิศวกรรมการวัดคุม
ภาควิชาวิศวกรรมการวัดคุม คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2546

เลขหมู่ 5604
เลขทะเบียน
วัน,เดือน,ปี 7 เม.ย. 2548
b
i
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

SCARA ROBOT

NOPADOL SAWANGYATI

NUMCHAI NARKVITUL

SOMCHAI SAICHONPHAISARN

A THESIS SUBMITTED IN PARTIAL FULFILLMENT

OF THE REQUIREMENT FOR THE DEGREE OF

BACHELOR OF ENGINEERING IN INTRUMENTATION ENGINEERING

DEPARTMENT OF INSTRUMENTATION ENGINEERING

FACULTY OF ENGINEERING

KING MONGKUT 'S INSTITUTE OF TECHNOLOGY LADKRABANG

2003

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาควิชาวิศวกรรมการวัดคุม

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ใบรับรองปริญญานิพนธ์

.....

หัวข้อปริญญานิพนธ์ แขนกล 3 แกน

SCARA ROBOT

นักศึกษาผู้จัดทำ นายนพดล สว่างญาติ **รหัสประจำตัว** 43010200

นายนำชัย นาควิทูร **รหัสประจำตัว** 43010217

นายสมชัย สายชลไพศาล **รหัสประจำตัว** 43010445

ปริญญา วิศวกรรมศาสตรบัณฑิต

สาขาวิชา วิศวกรรมการวัดคุม

ปีการศึกษา 2546

อาจารย์ผู้ควบคุมปริญญานิพนธ์	ลายมือชื่อ
รศ.ประภาษ อุคคกิมาพันธุ์	

วัน/เดือน/ปี ที่สอบ วันอังคารที่ 23 มีนาคม พ.ศ. 2547

สถานที่สอบ ณ ห้องสอบปริญญานิพนธ์ ภาควิชาวิศวกรรมการวัดคุม

ภาควิชารับรองแล้ว

(รศ.ประสิทธิ์ จุลเสรีวงศ์)

หัวหน้าภาควิชาวิศวกรรมการวัดคุม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อปริญญานิพนธ์	แขนกล 3 แกน	
	SCARA ROBOT	
นักศึกษาผู้จัดทำ	นายนพดล	สว่างญาติ
	นายนำชัย	นาควิทูร
	นายสมชัย	สายชลไพศาล
อาจารย์ที่ปรึกษา	รศ. ประภาษ	อุดคกิมาพันธุ์
ปีการศึกษา	2546	

บทคัดย่อ

ในปัจจุบันมีการใช้อุปกรณ์ที่สามารถทำงานแทนมนุษย์มากขึ้นเรื่อยๆ แขนกลเป็นอุปกรณ์ประเภทหนึ่งที่ได้รับความนิยม จึงเลือกแขนกลขับตรงเป็นกรณีศึกษา เนื่องจากมีโครงสร้างและการทำงานที่สามารถเข้าใจได้ง่าย โดยการเลือกแบบของโครงสร้างและ ศึกษาสมการการเคลื่อนที่ของวัตถุ จากนั้นทำการสร้างสมการตามโครงสร้างของแขนกลขับตรงที่สร้างขึ้นมา แล้วนำสมการที่ได้มาควบคุมการเคลื่อนที่ผ่านโปรแกรมที่เขียนขึ้นมาให้เป็นไปตามที่ต้องการเพื่อการประยุกต์ใช้ต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Thesis Title	Scara Robot
Authers	Mr. Nopadol Sawangyati
	Mr. Numchai Narkvitul
	Mr. Somchai Saichonphaisarn
Thesis Advisor	Assoc.Prof. Prapart Ukakimaparn
Year	2003

ABSTRACT

Robot, the one of equipment is more efficiently than human, become popular at this time. Scara 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 Scara Robot. The equation was used through the programed-software to control the robot for the advance application.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

การจัดทำปริญญานิพนธ์และโครงงานครั้งนี้ สำเร็จได้ด้วยดี เพราะได้รับการแนะนำข้อมูลเกี่ยวกับการดำเนินงาน การออกแบบ การสร้าง การเขียนตัวโปรแกรมและเครื่องมืออุปกรณ์ต่างๆ ตลอดจนความช่วยเหลือต่างๆ ทั้งในด้านทุนทรัพย์ แรงงาน และกำลังใจ จากบุคคลซึ่งมีรายนามดังต่อไปนี้

1. รศ. ประภาษ อุคคกิมาพันธุ์ (อาจารย์ที่ปรึกษา)
2. นาย กฤษณ์ เสมอพิทักษ์

ทางผู้จัดทำใคร่ขอขอบคุณเป็นอย่างสูง ในความช่วยเหลือครั้งนี้ ซึ่งส่งผลให้โครงงานนี้ประสบความสำเร็จลุล่วงไปได้ด้วยดี ทั้งนี้คุณค่าและประโยชน์อันพึงมีจากปริญญานิพนธ์ฉบับนี้ขอมอบแด่ผู้มีพระคุณทุกท่าน

คณะผู้จัดทำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

หน้า

บทคัดย่อภาษาไทย.....I

บทคัดย่อภาษาอังกฤษ.....II

กิตติกรรมประกาศ.....III

สารบัญ.....IV

สารบัญตาราง..... VII

สารบัญภาพ.....VIII

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

2.1.3 Scara Robot.....3

2.2 ทฤษฎีที่เกี่ยวข้อง(Theory).....4

2.2.1 พื้นฐานกลศาสตร์ที่เกี่ยวกับการเคลื่อนที่ (Basic Kinematics) 4

2.2.1.1 กลศาสตร์ที่เกี่ยวกับการเคลื่อนที่ (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 กลศาสตร์ที่เกี่ยวกับการเคลื่อนที่โดยตรง และ ผันกลับ (Direct & Inverse Kinematics Equation) ……..……8

2.2.3.1 Direct Kinematics Equation……..8

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ(ต่อ)

หน้า

2.2.3.2 Inverse Kinematics Equation.....10

2.2.4 การออกแบบ Link.....11

2.3 การสื่อสารแบบอนุกรม.....11

2.3.1 การสื่อสารข้อมูลแบบอะซิงโครนัส.....12

2.3.2 มาตรฐานพอร์ตอนุกรมแบบ RS-232..... 13

2.3.3 คอนเน็กเตอร์สำหรับพอร์ต RS-232 และการเชื่อมต่อ.....14

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 27

บทที่ 3 การคำนวณและการสร้าง.....31

3.1 Project Flowchart.....31

3.2 Hardware Flowchart.....32

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ(ต่อ)

หน้า

3.3 Software Flowchart 33
3.4 ลักษณะโครงสร้าง..... 34

บทที่ 4 การทดลอง..... 37
4.1 สมการคำนวณค่าพิกัด..... 37
4.1.1 การคำนวณการเคลื่อนที่โดยตรง..... 37
4.1.2 การคำนวณเคลื่อนที่แบบผันกลับ..... 38
4.2 ลักษณะของโปรแกรม..... 38
4.2.1 การใช้โปรแกรมคำนวณการเคลื่อนที่โดยตรง..... 39
4.2.2 การใช้โปรแกรมคำนวณการเคลื่อนที่แบบผันกลับ..... 40
4.2.3 การเคลื่อนที่แบบเส้นตรง..... 41

บทที่ 5 สรุปผลการวิจัยและข้อเสนอแนะ..... 45
5.1 ปัญหาและอุปสรรค..... 45
5.2 ข้อเสนอแนะและแนวทางการพัฒนา..... 45
5.3 สรุป..... 45

บรรณานุกรม..... 47
ภาคผนวก 48

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

ตารางที่ หน้า

2.1 แสดงข้อมูลในแอดเดรส 0000:0411H ที่ใช้แจ้งจำนวนพอร์ตอนุกรม.....19

2.2 แสดงผลของโมชั่นคอนโทรลเมื่อป้อนคำสั่ง H.....28

4.1 การคำควณการเคลื่อนที่โดยตรงจากสมการและโปรแกรม.....37

4.2 การคำควณการเคลื่อนที่แบบผันกลับจากสมการและโปรแกรม.....38

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญภาพ

ภาพที่ **หน้า**

2.1 Scara Robot.....4

2.2 แสดงกรอบอ้างอิง.....5

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.....7

2.8 แสดงความสัมพันธ์กันระหว่าง Direct Kinematics กับ Inverse Kinematics.....8

2.9 ลักษณะของแขน.....9

2.10 การแสดงขนาดของมุมต่างๆ.....9

2.11 รูปแบบอย่างง่ายที่สุดของข้อมูลอนุกรม.....12

2.12 รูปแบบอย่างง่ายที่สุดของข้อมูลอนุกรมแบบอะรสื่อสารข้อมูลแบบอะซิงโครนัส.....12

2.13 การจัดขาคอนเน็กเตอร์ตามมาตรฐาน RS-232 ทั้งแบบ DB9 และ DB25.....14

2.14 การต่ออุปกรณ์ภายนอกกับพอร์ตอนุกรมของคอมพิวเตอร์ในลักษณะต่างๆ.....15

2.15 ไดอะแกรมแสดงโครงสร้างทางฮาร์ดแวร์ของพอร์ตอนุกรม.....18

3.1 แสดงProject Flowchart.....31

3.2 Hardware Flowchart.....32

3.3 Software Flowchart.....33

3.4 ลักษณะโดยทั่วไป.....34

3.5 ขนาดความยามของแต่ละ Link.....34

3.6 ลักษณะการเคลื่อนที่ของแขน.....35

3.7 ลักษณะการเคลื่อนที่เมื่อมองจากด้านบน.....35

3.8 ขอบเขตของการเคลื่อนที่.....36

4.1 แสดงมุมของการเคลื่อนที่.....37

4.2 ลักษณะของโปรแกรมเมื่อเปิด.....38

4.3 การใช้โปรแกรมคำนวณการเคลื่อนที่โดยตรง.....39

4.4 การใช้โปรแกรมคำนวณการเคลื่อนที่แบบผันกลับ.....40

4.5 การเคลื่อนที่แบบกำหนดจุดปลายจุดเดียว.....41

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญภาพ(ต่อ)

ภาพที่ **หน้า**

4.6 การเคลื่อนที่แบบเส้นตรง.....43

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1
บทนำ

1.1 ความเป็นมาและเหตุจูงใจ

เนื่องจากในการทำงานจริง จำเป็นที่จะต้องมีการเรียนรู้เกี่ยวกับระบบการควบคุมวงจรทางอิเล็คทรอนิคส์ การเขียนโปรแกรม รวมถึงความรู้ทางกล เมื่อพิจารณาถึงหัวข้อต่างๆเหล่านี้ จึงได้เลือกแขนกลมาทำการศึกษาและได้เลือกแขนกล 3 แกน (Scara Robot) ขึ้นมาเป็นตัวอย่างในการศึกษาเนื่องจากรับภาระ (Load) ได้มากและมีความแม่นยำสูง สามารถนำไปประยุกต์ใช้งานกับงานที่ต้องทำซ้ำๆและประเภทที่แขนกลแนวแกน X , Y ไม่สามารถเข้าถึงได้

1.2 วัตถุประสงค์ของปริญญานิพนธ์

ปริญญานิพนธ์ฉบับนี้ได้ศึกษาถึงโครงสร้างของแขนกล การควบคุมการเคลื่อนที่ การสร้างโปรแกรมที่เขียนขึ้นโดยอาจจะเป็นควบคุมการเดินจุดต่อจุด หรือป้อนสมการเพื่อให้แขนกลทำการเคลื่อนที่ตามสมการทางเดินและสามารถนำไปประยุกต์ใช้กับงานด้านอื่นๆได้

1.3 ขอบเขต

ปริญญานิพนธ์ฉบับนี้จะกล่าวถึงแนวคิดในการสร้างแขนกล ความสามารถของแขนกลในการควบคุมการเคลื่อนที่ซึ่งประกอบไปด้วย ส่วนโปรแกรมวิชวลเบสิกที่ทำหน้าที่ติดต่อสื่อสารเพื่อจัดการป้อนตำแหน่งให้กับตัววงจรควบคุมการเคลื่อนที่ไปขับสัญญาณเข้าสู่มอเตอร์ผ่านวงจรไดร์ฟ

1.4 ขั้นตอนการศึกษา

การทำปริญญานิพนธ์นี้เริ่มทำการศึกษาจากชนิดของแขนกลชนิดต่างๆ การพิจารณาข้อดีข้อเสียและความเป็นไปได้ในการที่สร้างแขนกลขับตรงในแต่ละแบบและศึกษาสมการการเคลื่อนที่ (Kinematic&Inverse Kinematic) ศึกษาการทำงานของอุปกรณ์ต่างๆในการสร้างแขนกล ศึกษาการเขียนโปรแกรมวิชวลเบสิก ศึกษาการทำงานของชุดวงจรควบคุมการเคลื่อนที่และวงจรขับมอเตอร์และตัวมอเตอร์ ศึกษาการติดต่อสื่อสารผ่านพอร์ตอนุกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2
ทฤษฎีและหลักการ

2.1 ความรู้เบื้องต้นเกี่ยวกับโรบอท

2.1.1 Hydraulic Drives

จะมีความแม่นยำสูงและมีแรงขับค่อนข้างสูง ภายใต้โหลดที่มีขนาดใหญ่ แต่ Electrically Power Robot มักจะใช้กับการทำงานที่มีโหลดเบาๆ จึงมีราคาถูกและสะอาดกว่า Hydraulic Drives ทำให้การบำรุงรักษาง่ายมอเตอร์ไฟฟ้าจะผลิต Maximum Power ที่ความเร็วสูง แต่แขนของโรบอทเคลื่อนที่ด้วยความเร็วต่ำจึงทำให้มีแรงบิดต่ำไปด้วย ดังนั้นจึงต้องมีการใช้เฟืองทดเข้ามาช่วยในการออกแบบ จึงต้องคำนึงถึงอัตราการทดของเฟืองและต้องคำนึงถึงการรักษาความแม่นยำให้คงอยู่ด้วย เพราะการใช้เฟืองจะมีข้อผิดพลาดที่เรียกว่า Backlash ซึ่งเกิดจากเฟืองไม่แนบสนิทกันพอดี จึงทำให้ความแม่นยำลดลง เพราะถึงแม้ว่าจะมี Backlash เพียงเล็กน้อยที่ตัวขับ ก็จะนำไปสู่ค่าผิดพลาดจำนวนมากที่จุดปลาย

อีกปัญหาสำหรับการใช้เฟืองทดก็คือ ความเสียดทาน (Friction) เกิดจากการแก้ Backlash โดยการทำให้เฟืองแน่นหรือพอดีกัน ในทางกลับกันจึงเกิดแรงเสียดทานเป็นจำนวนมาก ซึ่งอาจทำให้กำลังที่ตัวขับจ่ายไปขับ Link สูญเสียไปกับแรงเสียดทานเป็นจำนวนมาก ทำให้ความแม่นยำลดลงและยากที่จะแก้ไข

คุณสมบัติที่โรบอทควรจะมีคือ

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

การออกแบบจึงต้องคำนึงถึงคุณสมบัติต่างๆเหล่านี้ ดังนั้นเมื่อกล่าวถึงการควบคุมตำแหน่งที่แม่นยำซึ่งไม่ได้ให้ความสำคัญกับการควบคุมแรง Hydraulic Drives ซึ่งมีความสามารถในการจ่าย Load ขนาดใหญ่ แต่มีปัญหาเกี่ยวกับการควบคุมตำแหน่งที่ละเอียดอ่อนจึงไม่ถูกนำมาพิจารณา

2.1.2 Direct Drives Robot

คือการควบคุมอย่างง่ายๆจากตัวขับทางไฟฟ้าโดยตรงสู่ Load ซึ่งไม่มีตัวทด (Reducer) มาช่วย ทำให้ขจัดปัญหาด้าน Mechanic ได้เป็นอย่างดี เพราะโครงสร้างง่ายๆของมันประกอบไปด้วย Armlinks, Motors, Baring (ตลับลูกปืน) โดยที่ Rotor จะยึดอยู่ที่จุดเชื่อมของแท่งแขน ดังนั้นแรงบิดที่จุดปลายจะถูกส่งตรงมาจาก Rotor และ Stator ของ Motor ทำให้ขจัดปัญหาเรื่อง Backlash และยัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ลดแรงเสียดทานไปได้เป็นจำนวนมาก แต่ Direct Drive ก็มีข้อเสียเช่นกัน คือมีความฝืด (Stiffness) ในโครงสร้างมากกว่าการขับด้วยระบบเฟือง แต่

ข้อดีของ Direct Drive ก็มีหลายประการด้วยกันคือ

1. มีความแม่นยำสูง เนื่องจากพฤติกรรมของระบบนี้สามารถสร้างขึ้นเป็น Model ได้ ดังนั้นผลตอบสนองทาง Dynamic จึงสามารถคำนวณและทำนายได้
2. มีความสามารถในการทำซ้ำสูง เนื่องจากมีความผิดพลาดทาง Mechanic ต่ำ
3. มีโครงสร้างที่ง่าย ทำให้บำรุงรักษาง่าย

ปัญหาที่ต้องแก้ไขในการออกแบบควบคุม Direct Drive

1. สามารถจ่าย Load ได้น้อยและแขนมีน้ำหนักมาก (Small Pay Load and Heavy Armweight)
2. คุณสมบัติการหน่วง (Damping Characteristic) เนื่องจาก Direct Drive ส่วนใหญ่จะมีค่าของ Damping Ratio น้อยมาก ซึ่งน้อยกว่าค่า Damping ที่จำเป็นต่อการตอบสนอง ดังนั้นเราจึงต้องเพิ่มค่าการหน่วงในการออกแบบแก้ไขได้โดยการเพิ่ม 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 จึงส่งผลต่อ Damping Ratio ทำให้เกิดสภาวะที่ไม่เสถียรขึ้น จึงต้องออกแบบให้มีผลของการตอบสนองที่เร็วพอ เพื่อให้เหมาะสมกับค่า Inductance ที่มีค่ามาก
5. Arm Dynamic Effects เกิดจากการควบคุมที่ไม่ Linear
6. Over Heat เกิดจากการขับ Load อย่างต่อเนื่อง

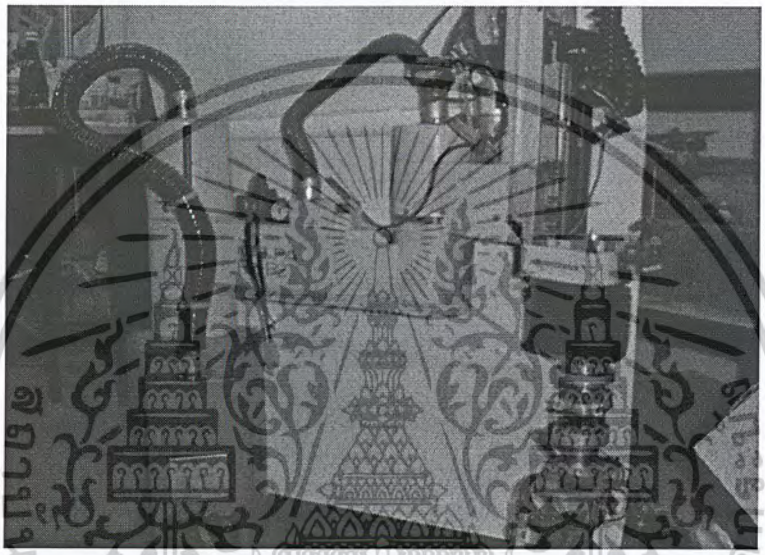
2.1.3 Scara Robot

เป็นแขนกลที่ใช้กันทั่วไปในโรงงานอุตสาหกรรม Scara Robot ซึ่งในทำงานแบบอัตโนมัติ การบอกความสัมพันธ์พื้นฐานของหุ่นยนต์โรบอทที่ทำขึ้นนี้เป็น 4 freedom degree ที่มี 3 ข้อต่อที่อยู่ในระนาบเดียวกันในความหมายนี้คือ โครงสร้างของโรบอทสนับสนุนการเคลื่อนที่ของน้ำหนัก prismatic link ดังนั้นข้อต่อที่1 และ 2 ต้องมีมอเตอร์ที่กำลังมากและผลการทำงานที่รวดเร็ว Scara Robot manipulator สามารถเคลื่อนที่ภายใน 10 วินาทีไม่เกิน 280 องศา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ข้อดีของ Scara Robot
- มีความแม่นยำสูงเนื่องจากมีการป้อนกลับตำแหน่ง
- สามรถรับภาระ (Load) ได้มาก
- เคลื่อนที่ได้รวดเร็ว
- ประยุกต์ใช้กับงานได้หลายประเภท

ข้อเสียของ Scara Robot
- ระบบการควบคุมยุ่งยาก
- ความแม่นยำขึ้นอยู่กับตัวป้อนกลับ
- Arm Dynamic Effects จากการควบคุมที่ไม่เป็นเชิงเส้น



ภาพที่ 2.1 Scara Robot

2.2 ทฤษฎีที่เกี่ยวข้อง (Theory)

2.2.1 พื้นฐานกลศาสตร์ที่เกี่ยวกับการเคลื่อนที่ (Basic Kinematics)

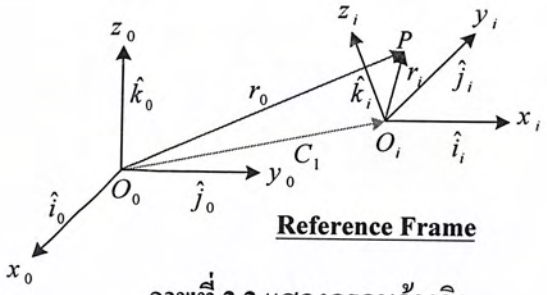
2.2.1.1 กลศาสตร์ที่เกี่ยวกับการเคลื่อนที่ (Kinematics)

ทฤษฎีที่ใช้อธิบายถึงเรขาคณิตของการเคลื่อนที่ของจุด โดยจะพูดถึงตำแหน่งที่จุดอาจจะเคลื่อนที่ผ่านในรูปของการสะสมของจุด (หรือ เส้นโค้ง : Curves) โดยมีองค์ประกอบที่เข้าเกี่ยวข้อง คือ ตำแหน่ง (Position) , ความเร็ว (Velocity) และความเร่ง (Acceleration)

2.2.1.2 กรอบอ้างอิง (Reference Frames)

วัตถุแข็งเกร็งที่มีการเคลื่อนที่ บางครั้งเราอาจจะกำหนดให้โครงสร้างของ Robot เป็นกรอบอ้างอิง (Reference Frames) เพื่อช่วยในการอ้างจุดเทียบกับกรอบอ้างอิงต่างๆ ที่จุด Origin (O_{xyz})$_i$ เมื่อ i = 0,1,2,3,.....,n โดยที่เมื่อ i = 0 จุด (O_{XYZ})$_0$ นี้จะเรียกว่า Fixed Reference Frame หรือ Base Reference Frame เมื่อนำไปเทียบกับ Reference Frame อื่นๆ ดังรูป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 2.2 แสดงกรอบอ้างอิง

โดยที่ r_o = เวคเตอร์ที่บอกตำแหน่งของจุด P เทียบกับ fixed reference frame

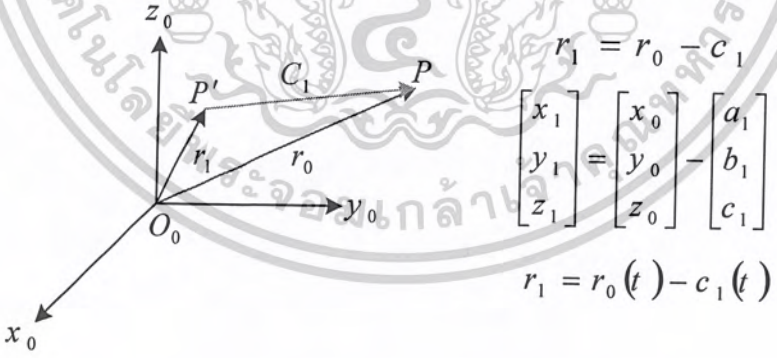
r_i = เวคเตอร์ที่บอกตำแหน่งของจุด P เทียบกับ จุด O_i ที่อาจเป็น Moving Frame หรือไม่ก็ได้

สามารถเขียน r_i ในรูปของ Vector และ Matrix ได้ดังนี้

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

2.2.1.3 การย้าย (Translation)

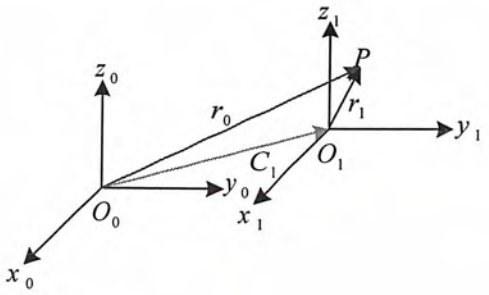
เป็นการศึกษาการเปลี่ยนแปลงของจุด P เมื่อเทียบกับกรอบอ้างอิงเดียวกัน หรือการเปลี่ยนแปลงพิกัดของจุด P เมื่อเทียบกับกรอบอ้างอิงที่แตกต่างกัน 2 กรอบ โดยที่การ Translation แบ่งออกเป็น 2 รูปแบบดังนี้ คือ



ภาพที่ 2.3 แสดงการย้าย (Translation)

1) Point Transformation คือ จุด P' เปลี่ยนเป็น P แต่ Coordinate คงเดิม โดยที่มี r_o เป็นฟังก์ชันของเวลา เนื่องจากจุด P เป็น Moving Point
2) Coordinate Transformation คือ จุด P เดียวกัน แต่เทียบกับกรอบอ้างอิงที่แตกต่างกัน 2 กรอบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



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

$$r_1 = r_0 - c_1$$

$$\begin{bmatrix} x_1 \\ y_1 \\ z_1 \end{bmatrix} = \begin{bmatrix} x_0 \\ y_0 \\ z_0 \end{bmatrix} - \begin{bmatrix} a_1 \\ b_1 \\ c_1 \end{bmatrix}$$

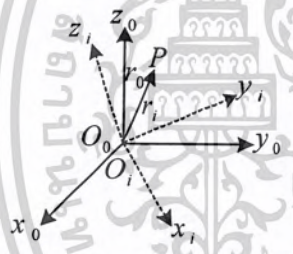
$$r_1(t) = r_0 - c_1(t)$$

ภาพที่ 2.4 แสดงสมการการย้ายแกน

r_1 เป็นฟังก์ชันของเวลาเนื่องจาก O_1 เป็น Moving Frame

2.2.1.4 เมตริกซ์การหมุน (Rotation Matrix)

เป็นเมตริกซ์ (Matrix) ที่ใช้อธิบายการหมุนของกรอบอ้างอิง ที่มีการเคลื่อนที่ (Moving Reference Frame) เมื่อเทียบกับกรอบอ้างอิงที่ไม่มีการเคลื่อนที่ (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

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

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

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

จัดสมการในรูปของ Matrix จะได้ดังนี้

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

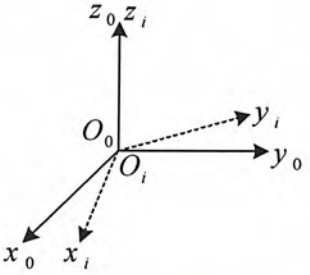
หรือ $r_0 = A_i r_i$

โดยเรียก A ว่า Rotation Matrix

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แต่จากโครงสร้างของ Robot พื้นที่การทำงานจะเป็น 2 มิติ โดยจะหมุนรอบแกน Z ดังนั้น จะได้ A ดังนี้ เมื่อ θ 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

$$A = \begin{bmatrix} \hat{i}_0 \bullet \hat{i}_i & \hat{i}_0 \bullet \hat{j}_i \\ \hat{j}_0 \bullet \hat{i}_i & \hat{j}_0 \bullet \hat{j}_i \end{bmatrix} = \begin{bmatrix} C\theta & -S\theta \\ S\theta & C\theta \end{bmatrix}$$



ภาพที่ 2.6 รูปและสมการ Rotation Matrix

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 เวคเตอร์ที่มี

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

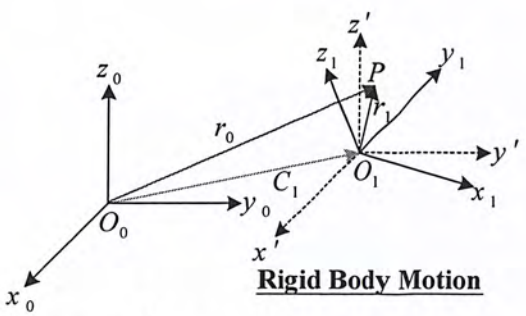
เมื่อนำไปแทนใน Translation จะได้สมการรวมดังนี้

$$r_0 = C_1 + A_1 r_1$$

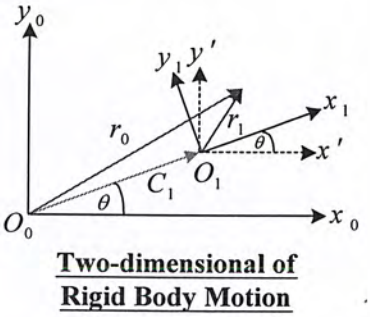
แต่จากการทำงานของ Robot เราจะควบคุมเป็นระนาบ 2 มิติ ที่หมุนรอบ

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

แกน Z ดังนั้นสมการ Matrix ที่ได้จะเป็นดังนี้



Rigid Body Motion



Two-dimensional of Rigid Body Motion

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

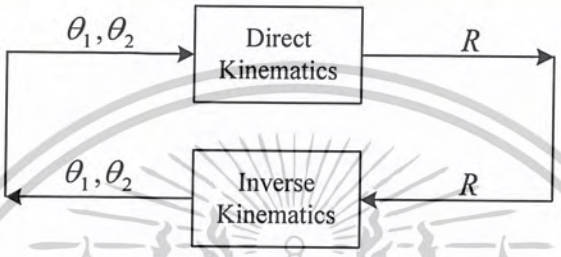
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.3 กลศาสตร์ที่เกี่ยวกับการเคลื่อนที่โดยตรงและผันกลับ

(Direct & Inverse Kinematics Equation)

Direct Kinematics เป็นสมการที่ใช้อธิบายการควบคุมเวคเตอร์ลัพธ์ (R) ของตัว Robot โดยการป้อนพารามิเตอร์ต่างๆ เช่น ค่ามุม (θ_1 , θ_2) ค่าความยาวของแต่ละ Link (L) เป็นต้น หรือ $R = f(\theta_1, \theta_2, L)$ นั่นเอง

Inverse Kinematics เป็นการคำนวณหาพารามิเตอร์ต่างๆ ณ จุดปลาย (R) ต่างๆ เพื่อใช้ในการป้อนค่าไปควบคุม



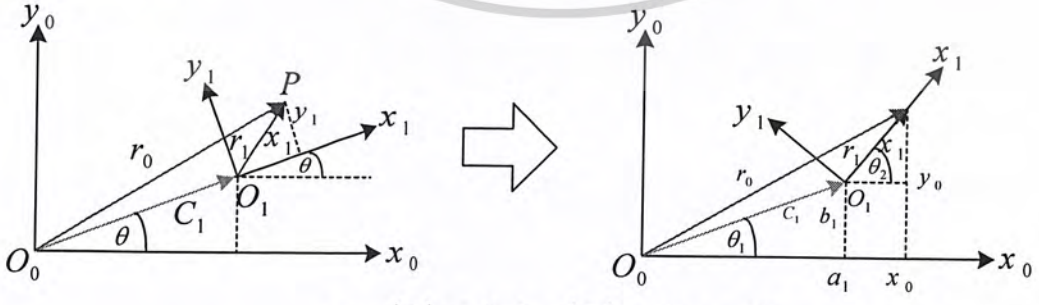
ภาพที่ 2.8 แสดงความสัมพันธ์กันระหว่าง Direct Kinematics กับ Inverse Kinematics

จากหลักการของ Direct & Inverse Kinematics จะเห็นว่า Inverse Kinematics จะเหมาะสมกับการที่เราทราบตำแหน่งของเวคเตอร์ลัพธ์ (R) จากนั้นจึงนำมาคำนวณหาค่าพารามิเตอร์ที่ใช้ควบคุม จากนั้นจึงนำพารามิเตอร์เหล่านี้ไปป้อนให้กับ Robot เพื่อให้ได้ตำแหน่งลัพธ์ตามที่ต้องการ

จากทฤษฎีที่ได้ศึกษามาแล้ว เมื่อนำมาประยุกต์ใช้กับ Robot ที่ได้ศึกษา ทำให้สร้างสมการต่างๆได้ดังนี้

2.2.3.1 Direct Kinematics Equation

จากรูปด้านล่าง เราเลือกที่จะให้จุด P ตกบนแกน x_1 เพื่อง่ายต่อการพิจารณา ทำให้ขนาดของ $r_1 = x_1$ ($y_1 = 0$) และเมื่อพิจารณาตามเรขาคณิต จะได้ค่าดังนี้

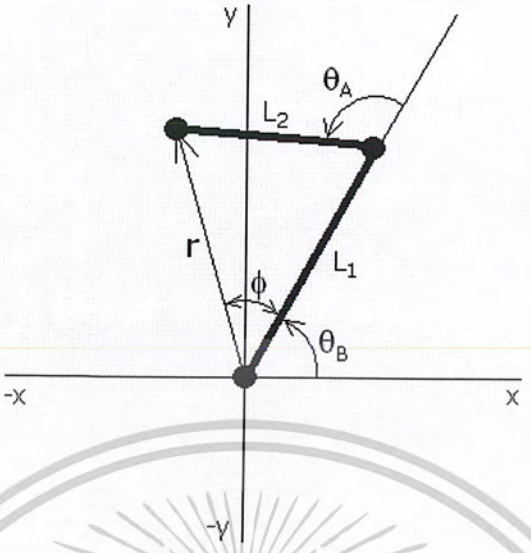


$$a_1 = |C_1|\cos\theta_1, b_1 = |C_1|\sin\theta_1, y_1 = 0$$

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

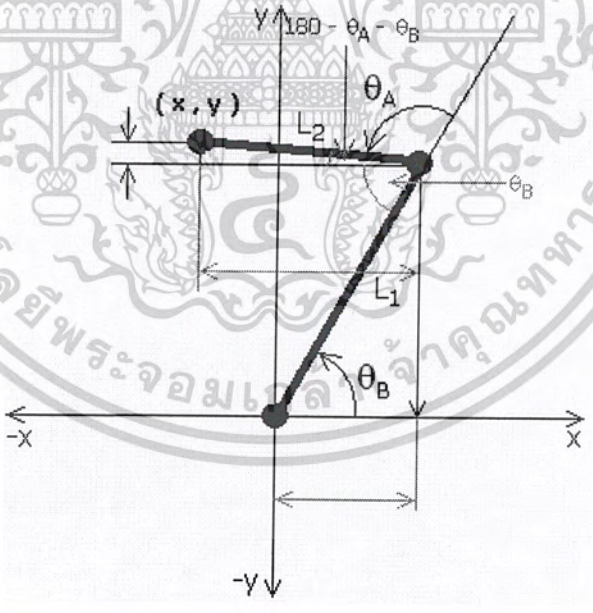
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อนำมาพิจารณาถึง Robot รวมถึงการติดตั้งจะได้ดังนี้



ภาพที่ 2.9 ลักษณะของแขน

จากสมการด้านบนและข้อมูลที่มี ทำให้เราสามารถสร้าง Direct Kinematics Equation ได้ดังนี้



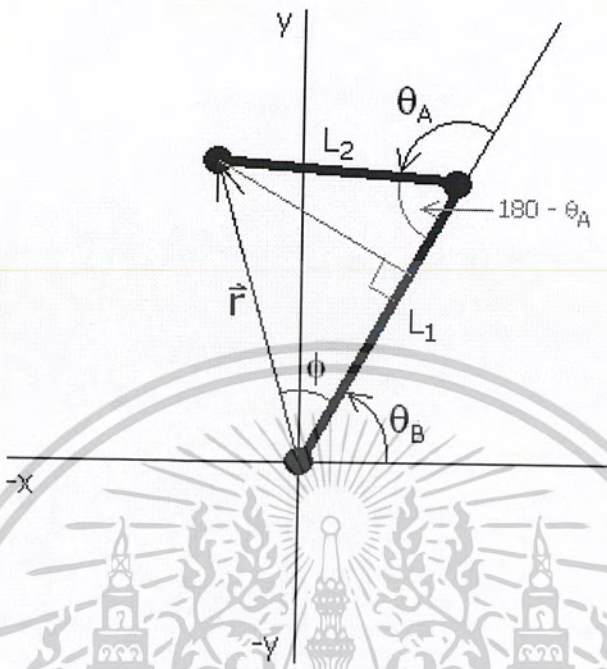
ภาพที่ 2.10 การแสดงขนาดของมุมต่างๆ

$$x = L_1 \cos\theta_B - L_2 \cos(180 - \theta_A - \theta_B)$$

$$y = L_1 \sin\theta_B + L_2 \sin(180 - \theta_A - \theta_B)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.3.2 Inverse Kinematics Equation



การคำนวณหาขนาดของมุม θ_A

$$r^2 = x^2 + y^2$$

$$r^2 = L_1^2 + L_2^2 - 2L_1L_2\cos(180 - \theta_A)$$

$$\theta_A = 180 - \cos^{-1}\left(\frac{r^2 - L_1^2 - L_2^2}{-2L_1L_2}\right)$$

การคำนวณหาขนาดของมุม θ_B

$$r\sin\phi = L_2\sin(180 - \theta_A)$$

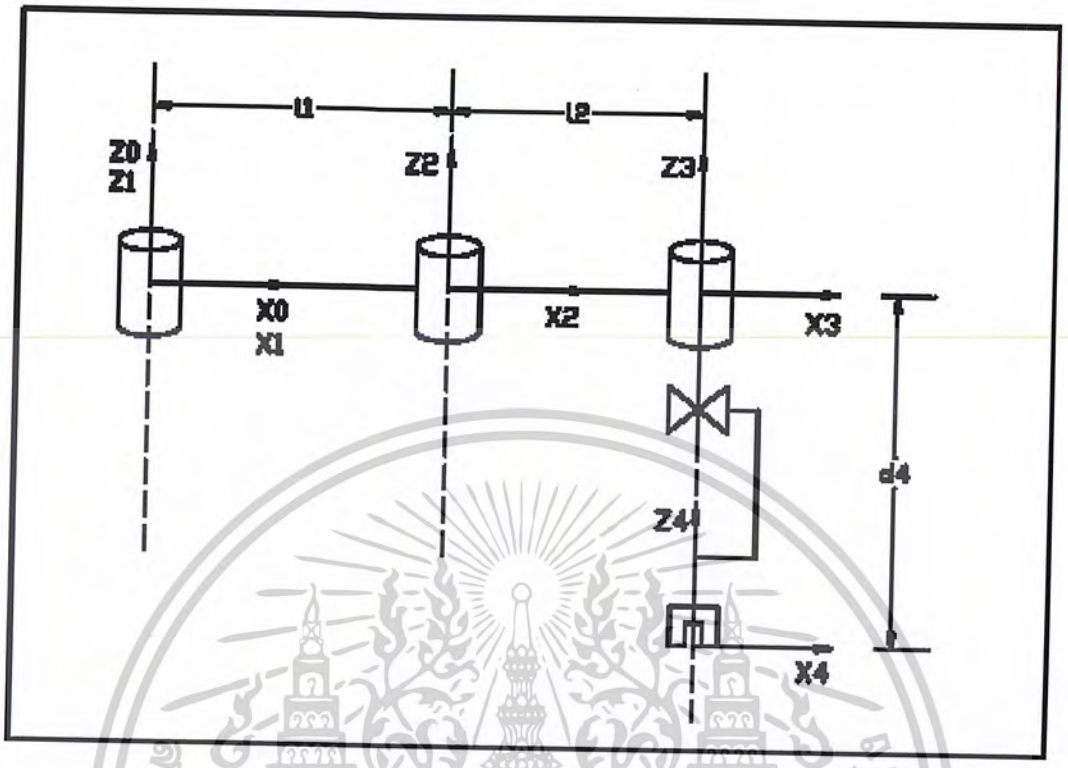
$$\phi = \sin^{-1}\left(\frac{L_2\sin(180 - \theta_A)}{r}\right)$$

$$\theta_B + \phi = \tan^{-1}\frac{y}{x}$$

$$\theta_B = \tan^{-1}\frac{y}{x} - \sin^{-1}\left(\frac{L_2\sin(180 - \theta_A)}{r}\right)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.4 การออกแบบ Link

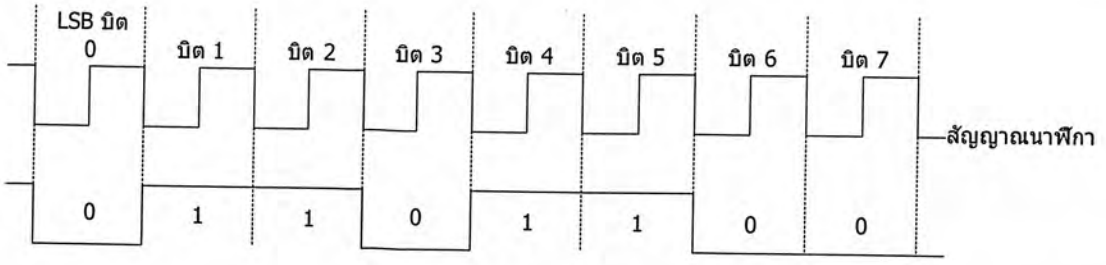


เฟรมแรกเรียกว่าเฟรมศูนย์ เป็นเฟรมฐานที่ใช้ในการยึดของ Scara Robot ทั้งหมด เฟรมที่เหลือ 4 เฟรม 4 ข้อต่อของระบบ Scara โดยข้อต่อที่ 1,2,3 เป็น Revolute และข้อต่อที่ 4 เป็นปริซึม เฟรมที่หนึ่งใช้ร่วมกับออริจินเฟรมของ 0 โดยสามารถเคลื่อนที่รอบแกน Z การยึดเฟรมหนึ่งและการเชื่อมต่อโดยตรงกับความยาว L1 จุดจบของลิงค์ 1 ที่อยู่ในเฟรม2 สามารถหมุนรอบแกน Z เป็นมุม θ_2 จุดเริ่มของลิงค์ 2 ในออริจินของเฟรมสองไปจนถึงทิศทางของ X2 โดยเป็นระยะ L2 เฟรม 3 มีออริจินที่จุดจบของ L2 และเคลื่อนที่เป็นอิสระตามแกน Z โดยมุม θ_3

2.3 การสื่อสารแบบอนุกรม

การสื่อสารแบบอนุกรมนั้นจะแบ่งออกได้เป็น 2 แบบคือการสื่อสารอนุกรมแบบซิงโครนัส และการสื่อสารอนุกรมแบบอะซิงโครนัส การสื่อสารแบบซิงโครนัสจะมีสัญญาณนาฬิการ่วมอยู่กับการรับและส่งสัญญาณด้วย ตัวอย่างการส่งข้อมูลแบบซิงโครนัสก็คือคีร์บอร์ดของคอมพิวเตอร์ ซึ่งสายเส้นหนึ่งจะเป็นสายของนาฬิกา ส่วนสายอีกเส้นจะเป็นสายของข้อมูล ดังนั้นการติดต่อกันแบบซิงโครนัสนี้ จะต้องใช้สายในการเชื่อมต่ออย่างน้อยที่สุด 3 เส้นคือ สัญญาณนาฬิกา ข้อมูลและกราวด์ ภาพที่ 1-1 แสดงให้เห็นถึงไทม์มิ่งไดอะแกรมของการส่งข้อมูลแบบซิงโครนัส

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 2.11 รูปแบบอย่างง่ายที่สุดของข้อมูลอนุกรม

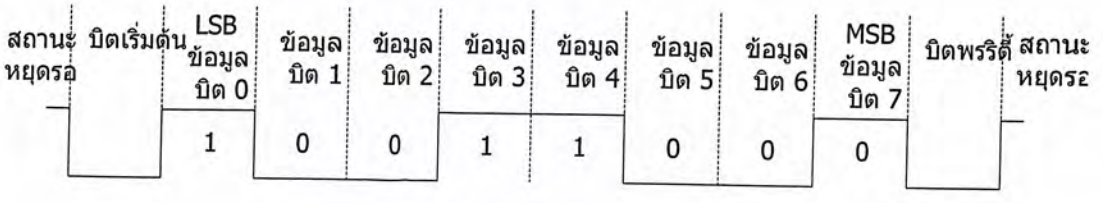
2.3.1 การสื่อสารข้อมูลแบบอะซิงโครนัส

การสื่อสารข้อมูลแบบอะซิงโครนัส คือการรับและส่งข้อมูลไปในสาย โดยไม่จำเป็นต้องมีสัญญาณนาฬิการ่วมด้วยเหมือนกับการรับส่งข้อมูลแบบซิงโครนัส แต่จะใช้การกำหนดค่าสัญญาณนาฬิการทั้งภาครับและภาคส่ง ให้มีค่าเท่ากัน ซึ่งเรียกสัญญาณนาฬิกาที่ใช้ในการกำหนดค่าให้กับภาครับและภาคส่งนี้ว่า อัตราการถ่ายทอดข้อมูลหรือบอดเรต (baudrate) มีหน่วยเป็น บิตต่อวินาที (bit per sec : bps)

รูปแบบของข้อมูลที่ใช้ในการรับส่งแบบอะซิงโครนัสประกอบด้วย 4 ส่วนด้วยกันคือ

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

ภาพที่ 1-2 แสดงรูปแบบของข้อมูลอนุกรมแบบอะซิงโครนัส ซึ่งเมื่อไม่มีข้อมูลที่จะส่งขา DATA จะมีสถานะลอจิก “1” ซึ่งจะเรียกสถานะนี้ว่าสถานะหยุดรอ (waiting stage) การเริ่มต้นจากนั้นบิตข้อมูลจะถูกส่งออกไป โดยเริ่มจากบิตที่มีนับสำคัญต่ำที่สุด (LSB) ก่อน ซึ่งข้อมูลในไบต์ที่จะส่งอาจจะมีจำนวนบิต 5,6,7 หรือ 8 บิตก็ได้ จากนั้นจะตามด้วยบิตพาริตี้บิต ซึ่งใช้เพื่อตรวจสอบความผิดพลาดจากการส่งข้อมูลบิตสุดท้ายที่จะส่ง คือบิตปิดท้าย ซึ่งจะใช้ขาดาต้าที่มีสถานะลอจิกเป็น 1 อีกครั้งด้วยระยะเวลาอย่างน้อย 1 บิต, 1.5 บิต หรือ 2 บิต เพื่อเป็นการแสดงว่าสิ้นสุดข้อมูลแล้ว



ภาพที่ 2.12 รูปแบบอย่างง่ายที่สุดของข้อมูลอนุกรมแบบอะรสื่อสารข้อมูลแบบอะซิงโครนัส

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อุปกรณ์ชนิดพิเศษ ที่ได้รับการออกแบบมาสำหรับการรับและส่งข้อมูล มีการสื่อสารข้อมูลแบบอะซิงโครนัสเรียกว่า Universal Asynchronous Receiver/Transmitter หรือ UART อัตราความเร็วในการรับและส่งข้อมูลของการรับส่งข้อมูลแบบอะรสื่อสารข้อมูล แบบอะซิงโครนัสคือค่าบอดเรต ซึ่งก็คือค่าจำนานบิตต่อวินาทีที่ใช้ในการรับและส่งข้อมูล บอดเรตมาตรฐานที่ใช้สำหรับพอร์ตอนุกรม RS-232 ได้แก่ 110, 150, 300, 600, 1200, 2400, 4800, 9600, และ 19200 บิตต่อวินาที และมีค่าเพิ่มมากขึ้นตามเทคโนโลยีของคอมพิวเตอร์ ซึงการรับส่งแบบอนุกรมโดยไม่ผ่านโมเด็มอาจจะสามารถกำหนดค่าบอดเรตได้สูงถึง 115200 บิตต่อวินาที เนื่องจากบอดเรตคือจำนวนบิตของข้อมูลที่สามารถถ่ายทอดได้ภายใน 1 วินาที ยกตัวอย่าง ข้อมูลอนุกรมถูกส่งในลักษณะ 8 บิต ไม่มีการตรวจสอบพาริตี้ มีบิตเริ่มต้น 1 บิต และบิตปิดท้าย 1 บิต ความยาวของข้อมูลที่รับส่งนี้เท่ากับ 10 บิต ถ้าใช้บอดเรดในการส่งข้อมูลเท่ากับ 9600บิตต่อวินาที ก็จะสามารถรับส่งข้อมูลได้ด้วยความเร็ว 960 ไบต์ต่อวินาที และถ้ามีการใช้พาริตี้ความเร็วในการรับส่งข้อมูลจะเหลือเป็น 872 ไบต์ต่อวินาที

2.3.2 มาตรฐานพอร์ตอนุกรมแบบ RS-232

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

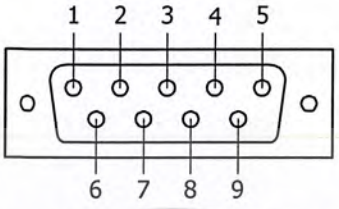
มาตรฐาน RS-232 ได้กำหนดรูปแบบของอุปกรณ์ที่เชื่อมต่อข้อมูล (Data Terminal Equipment: DTE) กับวงจรข้อมูลปลายทาง (Data Circuit Terminating: DCE) ไว้ว่า อุปกรณ์ CTE จะต้องเป็นอุปกรณ์ที่มีการประมวลผลในตัว เช่น ไมโครคอนโทรลเลอร์หรือไมโครคอมพิวเตอร์ ซึ่งมีความสามารถในการสร้างบิตข้อมูลแบบอนุกรมได้ ส่วนอุปกรณ์ DCE จะทำหน้าที่เป็นเพียงตัวรับข้อมูลที่ส่งมาจาก DTE เท่านั้น โดยการรับส่งข้อมูลระหว่างอุปกรณ์ทั้งสองจะกระทำผ่านมาตรฐาน RS-232

ข้อแตกต่างของอุปกรณ์ DTE และอุปกรณ์ DCE อย่างหนึ่งที่เห็นได้ชัดคือ คอนเน็กเตอร์ของ DTE จะเป็นตัวผู้ ส่วนคอนเน็กเตอร์ของ DCE จะเป็นตัวเมีย ซึ่งพอร์ตอนุกรมของคอมพิวเตอร์ที่ใช้กันอยู่ทั่วไปจะเป็นแบบ DTE ส่วนคอนเน็กเตอร์ที่อยู่ที่โมเด็มจะเป็นแบบ DCE สำหรับการใช้งานบนคอมพิวเตอร์ พอร์ตอนุกรม RS-232 มักถูกใช้เชื่อมต่อกับโมเด็มหรือเมาส์ โดยสามารถรับส่งข้อมูลได้ที่ความยาวของสายสัญญาณสูงสุดถึง 20 เมตร

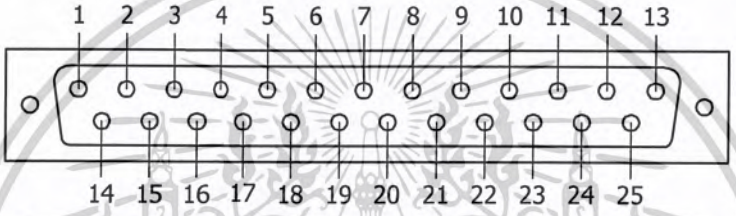
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

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 โดยแสดงรูปร่างและตำแหน่งขาในรูป ที่ 1-3



(ก) คอนเน็กเตอร์อนุกรม 9 ขาหรือแบบ DB-9 (มองจากด้านหลังคอมพิวเตอร์)



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

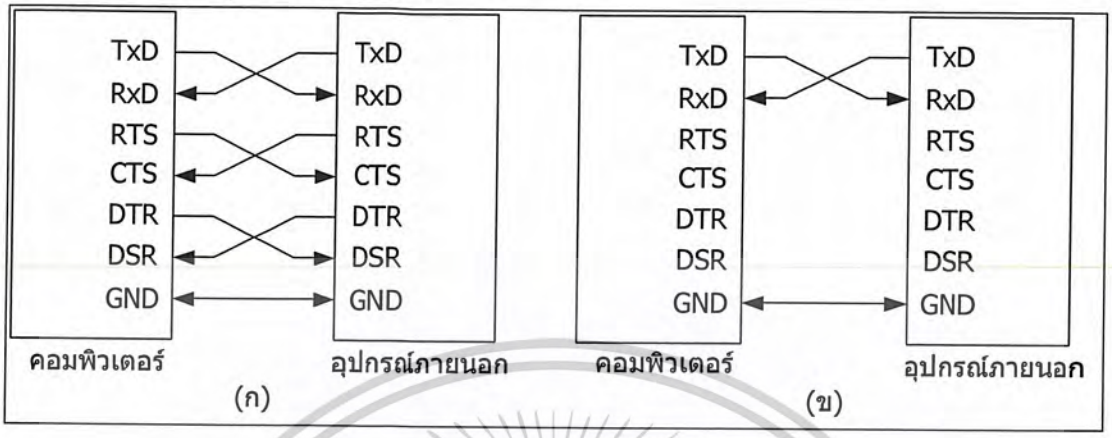
คอนเน็กเตอร์	คอนเน็กเตอร์	ชื่อของสายสัญญาณ	ชนิดของสายสัญญาณ
1	8	Data Carrier Detect: DCD	อินพุต
2	3	Received Data: RxD	อินพุต
3	2	Transmitted Data: TxD	เอาต์พุต
4	20	Data Terminal Ready: DTR	เอาต์พุต
5	7	Signal Ground: GND	-
6	6	Data Set Ready: DSR	อินพุต
7	4	Request To Send: RTS	เอาต์พุต
8	5	Clear To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

ภาพที่ 2.13 การจัดขาของคอนเน็กเตอร์พอร์ตอนุกรมตามมาตรฐาน RS-232 ทั้งแบบ DB-9 และ DB -25

สำหรับการเชื่อมต่อคอมพิวเตอร์กับอุปกรณ์ภายนอก แสดงดังในภาพที่ 1-4 ลูกศรในรูปแสดงถึงทิศทางของข้อมูล ในภาพที่ 1-4 (ก) เป็นการเขื่อมต่อแบบ Null modem หรือการเขื่อมต่อโดยตรงโดยไม่ต้องผ่านโมเด็มโดยมีการตรวจสอหรือแฮนด์เช็กเต็มรูปแบบ ส่วนในภาพที่ 1-4 (ข)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เป็นการเชื่อมต่อแบบ Null modemในลักษณะที่ใช้สายสัญญาณเพียง 3 เส้นโดยเส้นหนึ่งสำหรับส่งข้อมูลอีกเส้นสำหรับรับข้อมูลและเส้นสุดท้ายเป็นกราวด์ สำหรับรายละเอียดหน้าที่การทำงาภายในและขาของพอร์ตอนุกรม RS-232 มีดังนี้



ภาพที่ 2.14 การต่ออุปกรณ์ภายนอกกับพอร์ตอนุกรมของคอมพิวเตอร์ในลักษณะต่างๆ

(ก) การต่ออุปกรณ์ภายนอกเข้ากับคอมพิวเตอร์แบบ Null modem

(ข) การต่ออุปกรณ์ภายนอกเข้ากับคอมพิวเตอร์แบบ RS-232 โดยใช้สายสัญญาณเพียง 3 เส้น

- Data Carrier Detect : DCD หรืออาจเรียกว่า Carrier Detect:: CD ขานี้จะแอกตีฟเมื่อมีการส่งสัญญาณพาหะจากอุปกรณ์สื่อสารข้อมูล เช่น โมเด็ม สำหรับการใช้งานปกตินี้ จะไม่ได้ถูกใช้งานมากนัก
- Receive Data : RD หรือ RxD ขานี้ใช้เพื่อรับสัญญาณอนุกรมเข้ามายังคอมพิวเตอร์ โดยนำข้อมูลที่อ่านได้ เก็บไว้ในรีจิสเตอร์บัฟเฟอร์
- Transmitted Data : TD หรือ TxD ขานี้ใช้เพื่อส่งข้อมูลออกจากคอมพิวเตอร์ โดยนำข้อมูลที่เก็บอยู่ในบัฟเฟอร์สำหรับส่งข้อมูลส่งออกไป
- Data Treminal Ready : DTR เป็นขาสัญญาณที่ส่งออกจากคอมพิวเตอร์ เพื่อให้อุปกรณ์ปลายทาง รับรู้ว่าต้องการติดต่อด้วย โดยขา DTR นี้จะต้องเชื่อมต่อกับขา DSR ของอุปกรณ์ปลายทางและขา DTR ของอุปกรณ์ปลายทาง จะต้องเชื่อมต่อกับขา DSR ของคอมพิวเตอร์ ถ้าใช้การเชื่อมต่อเป็นแบบ Null Modem ซึ่งใช้สายในการเชื่อมต่อเพียง 3 เส้น จะต้องต่อขา DTR และ DSR ของตัวมันเองเข้าด้วยกัน และต้องต่อกับขา DCD ด้วยในกรณีโปรแกรมสื่อสารที่ใช้ มีการตรวจจับสัญญาณพาหะ
- Signal Ground : GND ขากราวด์ของระบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- Data Set Ready : DSR ขานั้นจะใช้คู่กับขา DTR เพื่อตรวจสอบการเชื่อมต่อกันระหว่างคอมพิวเตอร์กับอุปกรณ์ปลายทาง ซึ่งขา DSR จะเป็นขาสำหรับรับข้อมูลจากภายนอก ซึ่งถูกส่งมาจากขา DTR
- Request To Send : RTS เป็นขาสำหรับส่งสัญญาณร้องขอให้ทางอุปกรณ์ปลาทางส่งข้อมูลกลับมายังคอมพิวเตอร์ โดยขาที่รับสัญญาณ RTS ก็คือขา CTS ในกรณีที่ใช้การเชื่อมต่อ Null modem 3 สาย จะต้องเชื่อมต่อขา RTS และ CTS ของตัวมันเองเข้าด้วยกัน เพื่อจะให้การรับและส่งข้อมูลสามารถเกิดขึ้นได้ตลอดเวลา
- Clear To Send : CTS ขานี้จะคอยรับสัญญาณจากขา RTS เมื่อรับสัญญาณได้ข้อมูลที่ขา TxD จะถูกส่งออกไป ดังนั้นขานี้จึงถูกใช้เพื่อตรวจสอบอุปกรณ์ต่อพ่วงว่าพร้อมที่จะรับข้อมูลหรือไม่
- Ring Indicator : RI ใช้แสดงสถานะสัญญาณเรียกจากสายโทรศัพท์ ปกติในการสื่อสารโดยทั่วไปสายนี้จะไม่ถูกใช้งาน จะใช้งานก็ต่อเมื่อมีการเชื่อมต่อกับโมเด็มและโปรแกรมมีการตรจสอบสัญญาณนี้เท่านั้น

2.3.4 UART

UART ย่อจากคำว่า Universal Asynchronous Receiver Transmitter ซึ่งหมายถึงอุปกรณ์ที่ทำหน้าที่รับและส่งข้อมูลแบบอะซิงโครนัสนั่นเอง สำหรับการสื่อสารอนุกรมบนคอมพิวเตอร์แล้ว UART ถือว่าเป็นหัวใจสำคัญของการสื่อสารอนุกรม

หน้าที่หลักของ UART คือทำหน้าที่แปลงข้อมูลที่อยู่ในรูปแบบขนานจากคอมพิวเตอร์ให้อยู่ในรูปแบบอนุกรม แบบอะซิงโครนัส แล้วส่งออกไปและทำหน้าที่แปลงสัญญาณอนุกรมแบบอะซิงโครนัสที่ป้อนเข้ามายัง UART ให้เป็นแบบขนานก่อนที่จะส่งเข้าสู่คอมพิวเตอร์ ซึ่งนอกจาก UART จะส่งข้อมูลไปยังคอมพิวเตอร์แล้ว ยังแจ้งข้อมูลอื่นๆ ให้คอมพิวเตอร์รับทราบด้วย เช่น อัตราเร็วในการรับส่งข้อมูล (ผิดพลาดจากพาริตี้, เฟรมข้อมูล, โอเวอร์รัน) เป็นต้น

ภายใน UART มีส่วนของวงจรสร้างบอดเรตแบบโปรแกรมได้ (programmable buadrate generator) โดยการกำหนดค่าตัวหารให้กับสัญญาณนาฬิกาของ UART โดยตัวหารนี้มีขนาด 16 บิต ดังนั้นจึงสามารถกำหนดตัวหารอยู่ในช่วง 1 - 56,535 UART ซึ่งสามารถรับส่งข้อมูลได้ทั้งแบบฮาล์ฟดูเพล็กซ์ (half duplex) และฟูลดูเพล็กซ์ (full duplex) โดยส่งแบบฮาล์ฟดูเพล็กซ์ เป็นการส่งแบบทิศทางเดียว ส่วนการส่งแบบฟูลดูเพล็กซ์นั้น สามารถรับและส่งข้อมูลได้ในคราวเดียวกัน

ในเครื่องคอมพิวเตอร์ทั่วไปมี UART ที่ใช้งานกันอยู่ 2 เบอร์คือ 8250 ซึ่งเป็น UART มาตราฐานที่มีใช้กันมายานาน UART เบอร์นี้จะมีบัฟเฟอร์สำหรับรับและส่งข้อมูลตำแหน่งเดียวกัน ทำให้การรับและส่งข้อมูล ถูกจำกัดความเร็วอยู่ที่ 57.6 กิโลบิตต่อวินาทีเท่านั้น แต่UART เบอร์นี้ถือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

ว่าเป็นต้นแบบของ UART ที่ใช้ในคอมพิวเตอร์ โดยคอมพิวเตอร์ทุกๆรุ่น จะต้องสนับสนุนการทำงานตามรูปแบบของ UART เบอร์นี้

UART อีกเบอร์หนึ่งคือ 16450 มีความสามารถรับส่งข้อมูลได้ที่ความเร็ว 115,200 บิตต่อวินาที และเพิ่มรีจิสเตอร์สำหรบพักข้อมูลสำหรับ UART นอกจากนั้นยังเพิ่มส่วนของชิฟต์รีจิสเตอร์แบบ FIFO (First In First Out) ขนาด 16 ไบต์เข้าไปทำให้สามารถสนับสนุนความเร็วในการรับส่งข้อมูลของที่ 26 กิโลบิตต่อวินาทีได้ โดยคอมพิวเตอร์ในปัจจุบันใช้ UART เบอร์นี้ หรือใหม่กว่า เช่น เบอร์ TL16C750 ซึ่งมีรีจิสเตอร์แบบ FIFO ขนาด 64 ไบต์ทำงานได้ทีระดับแรงดัน +5V และ +3V มีโหมดประหยัดพลังงาน สามารถรับส่งข้อมูลได้ที่ความเร็ว 1 เมกะบิตต่อวินามี เมื่อใช้สัญญาณนาฬิกา 16 MHz

อย่างไรก็ตาม ความเร็วในการส่งข้อมูลที่มากมายของ UART เบอร์ใหม่ๆ ก็ไม่ได้ช่วยให้การรับส่งข้อมูลของคอมพิวเตอร์เร็วขึ้น เนื่องจากว่าคอมพิวเตอร์ยังใช้ความถี่ของสัญญาณนาฬิกาในการแปลงข้อมูลเพียง 1.8432 MHz เท่านั้น

2.3.5 รีจิสเตอร์ของพอร์ตอนุกรม RS-232

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

ในภาพที่ 1-5 แสดงไดอะแกรมการทำงานภายในของพอร์ตอนุกรม ซึ่งประกอบไปด้วยรีจิสเตอร์ขนาด 8 บิต 8 ตัวที่ใช้งานร่วมกับ UART แอดเดรสของรีจิสเตอร์ภายในพอร์ตอนุกรมสามารถคำนวนได้จากค่ารีจิสเตอร์พื้นฐานของพอร์ตอนุกรม ยกตัวอย่าง พอร์ตอนุกรม COM1 มีแอดเดรสอยู่ที่ 3F8H ตำแหน่งของรีจิสเตอร์ต่างๆจะเป็นตำแหน่งที่บวกเข้าไปกับค่า 3F8H โดยรีจิสเตอร์ที่ใช้งานกับพอร์ตอนุกรมมีดังนี้

00H รีจิสเตอร์บัฟเฟอร์สำหรับเก็บข้อมูลรับเข้ามาหรือเตรียมข้อมูลก่อนที่จะส่งออกไป

01H รีจิสเตอร์อีนาเบิลการอินเตอร์รัปต์ที่ใช้ในการเซตโหมดการอินเตอร์รัปต์ของพอร์ตอนุกรม

02H รีจิสเตอร์แสดงโหมดการอินเตอร์รัปต์ ใช้เพื่อตรวจอบโหมดการอินเตอร์รัปต์เมื่อมีการอินเตอร์รัปต์เกิดขึ้น

03H รีจิสเตอร์กำหนดรูปแบบของข้อมูล

04H รีจิสเตอร์ควาบคุมโมเด็ม ใช้ตรวจสอบิตสำหรับติดต่อกับโมเด็ม เช่น RTS หรือ DTR

05H รีจิสเตอร์แสดงสถานะการรับและการส่งข้อมูลแบอนุกรม

06H รีจิสเตอร์แสดงสถานะของโมเด็ม ซึ่งจะแสดงสถานะของขา DCD, RI, DSR และCTS

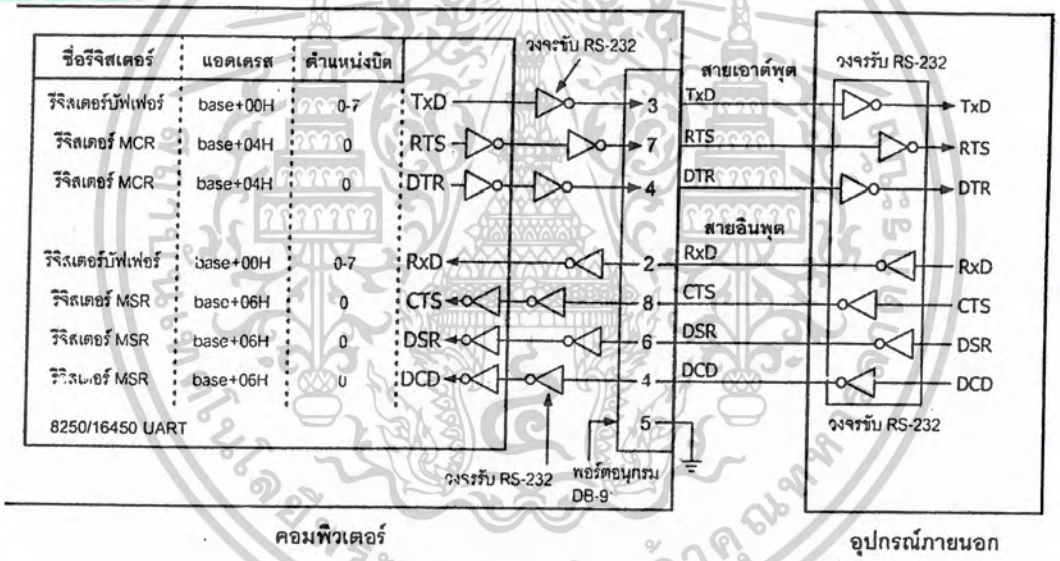
07H รีจิสเตอร์สำหรับการเก็บข้อมูลชั่วคราว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
55047

ลักษณะสัญญาณอินพุตและเอาต์พุตของพอร์ต RS-232

สัญญาณเอาต์พุตที่ใช้ควบคุม (RTS และ DTR) และสัญญาณแสดงสถานะอินพุต (CTS, DSR และ DCD) ของพอร์ตอนุกรม RS-232 จะถูกกลับสถานะภายในตัว UART ส่วนสัญญาณข้อมูลทั้งภาคส่งและรับจะไม่ถูกกลับสถานะ UART จะให้ระดับสัญญาณเอาต์พุตออกมาเป็นแบบทีทีแอลเท่านั้น ดังนั้นเมื่อสัญญาณถูกส่งออกมาจาก UART จึงต้องส่งเข้าสู่วงจรขับเพื่อปรับระดับแรงดันให้ได้ระดับสัญญาณเป็นไปตามมาตรฐาน RS-232 ก่อนส่งออกไปจากคอมพิวเตอร์ สำหรับอุปกรณ์ต่อเชื่อมปลายทางก็จะต้องมีวงจรขับในลักษณะนี้เช่นเดียวกัน เพื่อให้ได้ระดับสัญญาณในระดับเดียวกัน แต่วงจรขับที่ใช้ทั้งภายในคอมพิวเตอร์และอุปกรณ์ต่อเชื่อมปลายทางนั้น จะถูกกลับสถานะ ดังแสดงเป็นบล็อกไดอะแกรมในภาพที่ 1-6

2.3.6 แอดเดรสของพอร์ตอนุกรม



ภาพที่ 2.15 ไดอะแกรมแสดงโครงสร้างทางฮาร์ดแวร์ของพอร์ตอนุกรม

แอดเดรสพื้นฐานของพอร์ตอนุกรมมี 4 ตำแหน่ง ดังนี้คือ

COM1 : 3F8H

COM2 : 2F8H

COM3 : 3E8H

COM4 : 2E8H

เมื่อเริ่มเปิดเครื่องเพื่อใช้งานคอมพิวเตอร์ ไบออสภายในคอมพิวเตอร์จะทำการตรวจสอบแอดเดรสของพอร์ตอนุกรมทั้งหมด ถ้าไบออสตรวจพบแอดเดรสของพอร์ตอนุกรม ไบออสจะนำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แอดเดรสที่ตรวจพบไปเก็บไว้ในหน่วยความจำขนาด 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

นอกจากนี้ ที่หน่วยความจำแอดเดรส 0000:0411H ยังใช้สำหรับแสดงจำนวนของพอร์ตอนุกรมที่มีใช้อยู่ในคอมพิวเตอร์อีกด้วยโดยมีรายละเอียดดังแสดงในตารางที่ 1-2

ตารางที่ 2.1 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

มาตราฐานการสื่อสารข้อมูลของพอร์ตอนุกรม ได้ระบุช่วงระดับแรงดันสำหรับการทำงานของพอร์ตอนุกรมไว้ว่า ที่ลอจิก “0” จะมีระดับสัญญาณ +3V ถึง +15V ส่วนลอจิก ”1” จะมีระดับสัญญาณ –3V ถึง –15V ระดับสัญญาณนี้ทำให้ไม่สามารถที่จะนำขาเอาต์พุตใดๆต่อเข้ากับลอจิกเกตเพื่อใช้งานได้โดยตรง จะต้องผ่านวงจรเพื่อเปลี่ยนระดับแรงดันเสียก่อน โดยปกติ จะใช้ไอซีจำพวก RS-232 transceiver ที่นิยมมากคือ MAX232 หรือ ICL232 ไปใช้ในกลุ่มนี้ จะทำหน้าที่แปลงระดับแรงดันของ RS-232 ให้อยู่ในระดับ ทีทีแอล โดยลอจิก “0” ซึ่งเดิมมีระดับสัญญาณ +3V ถึง +15V จะถูกแปลงเป็น 0V ส่วนลอจิก “1” ซึ่งมีระดับสัญญาณ –3 ถึง –15V จะแปลงเป็น +5V ทั้งนี้เพื่อให้สามารถเชื่อมต่อกับอุปกรณ์ดิจิตอลอื่นที่ใช้ระดับแรงดัน ทีทีแอล ได้

2.4 การเขียนโปรแกรมเพื่อใช้งานพอร์ตอนุกรม

2.4.1 การกำหนดค่าเริ่มต้นให้กับพอร์ตอนุกรม

ก่อนการใช้งานพอร์ตอนุกรมนั้นจะต้องมีการกำหนดค่าเริ่มต้นให้กับตัวมันก่อน ซึ่งก็คือการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กำหนดจำนวนบิตข้อมูลที่ต้องการจะส่ง, จำนวนปิดท้าย, ชนิดของพาริตี้ที่ใช้และบอตเรต

การกำหนดสามารถทำได้หลายวิธี วิธีแรกเป็นการกำหนดจากดอสพร้อมพ์ โดยใช้คำสั่ง MODE ซึ่งมีวิธีในการใช้งานดังนี้

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

หรือ Mode Comm: b, p ,d ,s ,r

ตัวอย่าง MODE COM1 : 9600, n, 8, 1 จะเป็นการกำหนดให้พอร์ตอนุกรม COM1 มีบอตเรตเท่ากับ 9600 บิตต่อวินาที ไม่มีการตรวจสอบพาริตี้รับส่งข้อมูลแบบ 8 บิตและมีบิตปิดท้าย 1 บิต

วิธีที่ 2 เป็นการกำหนดโดยใช้อินเตอร์รัปต์ของดอส 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 ใช้สำหรับกำหนดค่าบอตเรต

"111" บอตเรตเท่ากับ 9,600 บิตต่อวินาที

"110" บอตเรตเท่ากับ 4,800 บิตต่อวินาที

"101" บอตเรตเท่ากับ 2,400 บิตต่อวินาที

"100" บอตเรตเท่ากับ 1,200 บิตต่อวินาที

"011" บอตเรตเท่ากับ 600 บิตต่อวินาที

"010" บอตเรตเท่ากับ 300 บิตต่อวินาที

"001" บอตเรตเท่ากับ 150 บิตต่อวินาที

"000" บอตเรตเท่ากับ 110 บิตต่อวินาที

PAR1,PAR0 ใช้กำหนดค่าพาริตี้โดย

"00" หรือ "10" ไม่มีการตรวจสอบพาริตี้

"01" พาริตี้คี่

"11" พาริตี้คู่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

STOP ใช้กำหนดจำนวนของบิตปิดท้าย

"1" มีบิตปิดท้ายเท่ากับ 2 บิต

"0" มีบิตปิดท้ายเท่ากับ 1 บิต

DA1,DA0 ใช้กำหนดความยาวของข้อมูลโดย

"10" ความยาวข้อมูลเท่ากับ 7 บิต

"11" ความยาวข้อมูลเท่ากับ 8 บิต

2.4.2 การรับและส่งข้อมูลแบบอนุกรม

มีหลากหลายวิธีในการรับและส่งข้อมูลแบบอนุกรมผ่านพอร์ต RS-232 เช่น ใช้คำสั่งพิมพ์ออกทางเครื่องพิมพ์ เรียกอินเทอร์รัปต์ของไบออสหรือของดอส การเขียนหรืออ่านไปยังแอดเดรสของพอร์ตโดยตรง วิธีสุดท้ายเป็นวิธีที่มีความยืดหยุ่นในการใช้งานที่สุด ยกตัวอย่าง ถ้าต้องการส่งข้อมูลไปยังพอร์ตอนุกรม COM1 สามารถเขียนข้อมูลโดยตรงไปที่รีจิสเตอร์บัฟเฟอร์สำหรับส่งข้อมูล (แอดเดรส 3F8H) โดยใช้คำสั่งภาษา QBASIC ง่ายๆดังนี้

OUT &H3F8,X

ค่า X ในที่นี้หมายถึงข้อมูลที่ต้องการส่งมีขนาด 8 บิต

สำหรับการอ่านค่าข้อมูลจากพอร์ตอนุกรม จะเป็นการอ่านข้อมูลมาจากรีจิสเตอร์บัฟเฟอร์สำหรับรับข้อมูล (แอดเดรส 3F8H เช่นเดียวกัน) ซึ่งสามารถเขียนโปรแกรมง่ายๆได้ดังนี้

Y=INP[&3F8]=X

แต่เมื่อมีการใช้คำสั่งนี้ ในขณะที่โปรแกรม ทำงานผ่านระบบปฏิบัติการวินโดวส์ จะไม่สามารถใช้งานได้ เนื่องจากระบบปฏิบัติการวินโดวส์นั้น ได้เข้าฝังตัวพอร์ตอนุกรมเข้าเป็นส่วนหนึ่งของระบบปฏิบัติการ ดังนั้นการใช้งาน ต้องเรียกผ่านเครื่องมือที่ติดต่อผ่านระบบปฏิบัติการ เช่น MSCOMM32.OCX ของโปรแกรม Visual Basic

2.4.3 คอนโทรล MSCOMM

สำหรับการใช้งาน Visual Basic ตั้งแต่เวอร์ชั่น 2 เป็นต้นมาใน Visual Basic จะมีคัสตอมคอนโทรล สำหรับการสื่อสารอนุกรมผ่านทางพอร์ตอนุกรมของคอมพิวเตอร์มาให้ โดยใน Visual Basic เวอร์ชัน 3 จะชื่อว่า MSCOMM.VBX ส่วนเวอร์ชัน 4 ใช้ชื่อว่า MSCOMM16.OCX สำหรับการทำงานกับระบบปฏิบัติการ 16 บิตและ MSCOMM32.OCX สำหรับระบบปฏิบัติการ 32 บิต สำหรับ Visual Basic เวอร์ชันต่อๆมาจะมีเพียง MSCOMM32.OCX เท่านั้น

MSCOMM จัดเตรียมทางเลือกเอาไว้ 2 ทางเพื่อความสะดวกในการสื่อสารข้อมูล ทางแรกคือการสื่อสารข้อมูลที่กระตุ้นด้วยเหตุการณ์ (event-driven communication) เป็นรูปแบบการใช้งานที่มีประสิทธิภาพมาก สำหรับการตอบสนองแบบทันทีทันใด ยกตัวอย่างเช่น เมื่อตัวอักษรถูกส่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

มาที่พอร์ตอนุกรม หรือเกิดการเปลี่ยนแปลงที่ขา Data Carrier Detect (DCD) หรือขา Request To Send (RTS) เหตุการณ์ Oncomm ของ MSCOMM จะสามารถตรวจจับได้ทันที ส่วนทางเลือกที่สอง เป็นการคอยตรวจสอบค่าเหตุการณ์และความผิดพลาดที่เกิดขึ้น ด้วยการดูค่าที่เปลี่ยนแปลงภายในคุณสมบัติ Commevent หลังจากให้โปรแกรมทำงานในฟังก์ชั่นต่างๆไปเรียบร้อยแล้ว ซึ่งวิธีนี้ใช้งานได้ดีในกรณีที่โปรแกรมมีขนาดเล็ก

คอนโทรล MSCOMM 1 ตัว สามารถควบคุมการทำงานของพอร์ตอนุกรมได้ 1 พอร์ต ถ้าในโปรแกรมที่ใช้งาน ต้องการติดต่อกับพอร์อนุกรมในแต่ละพอร์ตแอดเดรสในพอร์อนุกรมและแอดเดรสของการอินเทอร์รัปต์ สามาถเปลี่ยนแปลงได้จากการแก้ไขค่าที่ Control Panel ถึงแม้ว่าคอนโทรล MSCOMM จะมีคุณสมบัติมากมายแต่จะสามารถทำความเข้าใจได้ไม่ยากดังนี้

2.4.3.1 CommPort

ใช้ในการกำหนดอ่านค่าพอร์ตอนุกรมที่ติดต่ออยู่ (COM1,COM2,COM3,COM4) รูปแบบการใช้งาน

Object.Settings[=Value]

ค่า Value เป็นค่าของพอร์ตอนุกรมชนิดของข้อมูลเป็น Integer ค่า Value สามารถกำหนดได้ในช่วง 1 - 16 (ค่าเริ่มต้นกำหนดไว้ที่ 1) เมื่อมีการกำหนดค่าแล้ว ทำการเปิดพอร์ตอนุกรมโดยใช้คุณสมบัติ PortOpen แต่ว่าพอร์ตนั้นไม่มีอยู่ในระบบ MSCOMM จะสร้างสัญญาณแสดงข้อผิดพลาด error 68 ขึ้นมา ซึ่งหมายถึงอุปกรณ์ตัวนี้ไม่มีอยู่ในระบบ ดังนี้นการเขียนโปรแกรมจึงจำเป็นต้องกำหนดตำแหน่งของพอร์ตอนุกรมก่อนที่ใช้คำสั่ง OpenPort

2.4.3.2 Setting

ใช้ในการกำหนดและอ่านค่าอัตราบอตเรต, พาริตี้, จำนวนบิตข้อมูล, จำนวนของบิตปิดท้าย

รูปแบบการใช้งาน

Object.settings[=value]

ค่า value นี้ มีชนิดข้อมูลเป็นแบบ string มีรูปแบบเป็น " BBBB, P, D, S " โดย BBBB เป็นค่าอัตราบอตเรต, P เป็นค่าพาริตี้, D เป็นจำนวนของข้อมูลบิตข้อมูลและ S เป็นจำนวนของบิตปิดท้าย ปกติแล้วค่านี้ถูกกำหนดไว้เป็น " 9600, N, 8, 1 "

ค่าบอตเรตมาตรฐานที่ใช้กับ MSCOMM มีดังนี้

110 บิตต่อวินาที

300 บิตต่อวินาที

600 บิตต่อวินาที

1,200 บิตต่อวินาที

2,400 บิตต่อวินาที

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

9,600 บิตต่อวินาที

14,400 บิตต่อวินาที

19,200 บิตต่อวินาที

28,800 บิตต่อวินาที

38,400 บิตต่อวินาที

56,000 บิตต่อวินาที

128,000 บิตต่อวินาที

256,000 บิตต่อวินาที

สำหรับค่ามาตรฐานในการทำหนดค่าพาริตี้มีดังนี้

สัญลักษณ์	รายละเอียด
E	พาริตี้คู่
M	ลอจิก"1"
N	ไม่ใช้
O	พาริตี้คี่
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 โดยจะเป็นการกำหนดค่าบอดเรตเท่ากับ 9600 ไม่มีพาริตี้ จำนวนข้อมูล 8 บิตและบิตปิดท้าย 1 บิต สามารถเขียนโปรแกรมได้ดังนี้

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

สาเหตุที่ค่าที่กำหนดจะต้องอยู่ภายในเครื่องหมายคำพูด ก็เนื่องจากค่าที่กำหนด อยู่ในรูปตัวแปร String

2.4.3.3 PortOpen

เพื่อใช้ในการกำหนดและอ่านค่าสถานะของพอร์ตอนุกรม เพื่อทำการเปิดและปิดพอร์ตอนุกรม

รูปแบบการใช้งาน

Object.portopen[=value]

ค่า value นี้ มีชนิดของข้อมูลเป็นแบบบูลีน คือมี True กับ False โดย True จะหมายถึงพอร์ตที่ได้ถูกเปิดออกแล้วและ False หมายถึงการปิดพอร์ตอนุกรม สำหรับการปิดพอร์ตนั้นจะมีการเคลียร์บัฟเฟอร์รับข้อมูลและบัฟเฟอร์ส่งข้อมูลด้วย คอนโทรล MSCOMM จะปิดพอร์ตอนุกรมโดยอัตโนมัติเมื่อออกจากโปรแกรม ก่อนที่จะใช้คุณสมบัติ PortOpen ต้องตรวจสอบให้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แน่ใจก่อนว่าคุณสมบัติ Commport นั้นได้ทำการกำหนดตำแหน่งของพอร์ตอนุกรมไว้ได้ถูกต้องหรือไม่ มิฉนั้นแล้ว MSCOMM จะแสดงข้อผิดพลาด error 68 แจ้งแก่ผู้ใช้งาน หรือถ้าพอร์ตอนุกรมนั้นถูกเปิดเอาไว้แล้ว โปรแกรมก็จะแจ้งข้อผิดพลาดออกมาเช่นเดียวกัน

ถ้าคุณสมบัติ DTR Enable หรือ RTS Enable ถูกกำหนดให้เป็น True ก่อนที่จะทำการเปิดพอร์ต ค่าคุณสมบัติของ DTR Enable หรือ RTS Enable จะถูกเซตเป็น False หลังจากปิดพอร์ต แต่ถ้าเซตเป็น False หลังจากปิดโปรแกรมแล้วค่าที่กำหนดไว้จะเป็นค่าเดิม

ตัวอย่างการใช้คำสั่งเปิดพอร์ตเพื่อติดต่อสื่อสารกับพอร์ตอนุกรม COM1 และมีบอตเรต 9,600 บิตต่อวินาที ไม่มีพาริตี้ 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 จะส่งค่าข้อมูลกลับมาในรูปแบบของข้อความ ชนิดข้อมูลเป็นแบบ Variant ถ้า comInputModeBinary คุณสมบัติ Input จะส่งข้อมูลกลับมาในรูปของไบนารี่และชนิดข้อมูลเห็นแบบ Variant

ตัวอย่างโปรแกรมแสดงให้เห็นถึงวิธีการรับข้อมูลจากบัฟเฟอร์ทั้งหมด

```
Private  Sub Command_click()
Dim  Instring as  String
        Mscomm1.inputlen=0
        If  Mscomm1.inbuffercount then
        Instring=Mscomm1.input
        End  if
End sub
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4.3.5 InBufferCount

ส่งค่าจำนวนของตัวอักษรที่อยู่ในบัฟเฟอร์ภาครับ

รูปแบบการใช้งานคำสั่ง

Object.inbuffercount[=value]

คำสั่ง InBufferCount จะแสดงค่าจำนวนของตัวอักษร ซึ่งรับมาจากภายนอกและยังเก็บอยู่ในบัฟเฟอร์ของภาครับ เพื่อให้ผู้ใช้งานอ่านค่าออกไปสำหรับการเคลียร์ค่าบัฟเฟอร์ภาครับทำได้โดยกำหนดให้ InBufferCount มีค่าเป็น 0

หมายเหตุ อย่าสับสนระหว่างคำสั่ง InBuffersize และ Inbuffercount สำหรับคำสั่ง InBuffersize นั้น ใช้เพื่อกำหนดขนาดของบัฟเฟอร์

2.4.3.6 InBufferSize

กำหนดและคืนค่าขนาดของบัฟเฟอร์ภาครับในหน่วยเป็นไบต์

รูปแบบการใช้งานคำสั่ง

Object.inbuffersize[=value]

คำสั่ง Inbuffersize ใช้เพื่อกำหนดขนาดของบัฟเฟอร์ภาครับ โดยค่าเริ่มต้นถูกกำหนดไว้ที่ 1,024 ไบต์

หมายเหตุ การกำหนดค่าบัฟเฟอร์ภาครับขนาดใหญ่ จะทำให้หน่วยความจำที่เหลือสำหรับการใช้งานในส่วนอื่นๆจะเหลือน้อยลง อย่างไรก็ตาม การกำหนดค่าบัฟเฟอร์ภาครับที่น้อยเกินไป จะทำให้เกิดการโอเวอร์โฟลว์หรือข้อมูลล้นบัฟเฟอร์ เว้นแต่จะมีการใช้แฮนด์เช็ก ดังนั้นค่าปานกลางที่เหมาะสมก็คือค่า 1,024 ซึ่งเป็นค่าเริ่มต้นนั่นเอง แต่ถ้าโปรแกรมมีการเกิดโอเวอร์โฟลว์แล้วจึงค่อยปรับเพิ่มค่าขนาดของบัฟเฟอร์ให้มีค่ามากขึ้น

2.4.3.7 InputLen

กำหนดค่าและคืนค่าจำนวนของตัวอักษรที่อ่านจากบัฟเฟอร์ภาครับ

รูปแบบการใช้งานคำสั่ง

Object. InputLen[=value]

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

ถ้าไม่มีข้อมูลอยู่ในบัฟเฟอร์ภาครับมากเท่ากับจำนวน InputLen คำสั่ง Input จะส่งค่าว่างกลับออกมา ผู้ใช้งานสามารถตรวจสอบข้อมูลที่อยู่ในบัฟเฟอร์ภาครับก่อนแล้วจึงค่อยอ่านข้อมูลจากบัฟเฟอร์ภาครับ

คุณสมบัตินี้ มักจะใช้กับการอ่านค่าข้อมูลจากเครื่องมือหรือเครื่องจักรที่กำหนดค่าขนาดความยาวของข้อมูลเอาไว้แล้ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตัวอย่างโปรแกรมการอ่านค่าตัวอักษรออกมา 10 ตัวอักษร

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

2.4.3.8 InputMode

กำหนดค่าและคืนค่าชนิดของข้อมูลที่รับโดยคำสั่ง Input

รูปแบบการใช้งานคำสั่ง

Object.inputmode[=value]

คุณสมบัติ InputMode ใช้กำหนดว่าข้อมูลชนิดไหนที่รับเข้ามาผ่านคำสั่ง Input โดยข้อมูลจะเลือกได้ 2 ประเภทคือ

ComInputModeText สำหรับข้อมูลที่อยู่ในรูปข้อความตัวอักษรตามมาตรฐาน ANSI โดยจะต้องกำหนดค่าเป็น "0" และค่าเริ่มต้นของการรับค่าข้อมูลก็จะเป็นค่านี้

ComInputModeBinary สำหรับข้อมูลอื่นๆ ซึ่งจะเก็บอยู่ในรูปไบนารี่รวมกันอยู่เป็นไบต์ข้อมูล

ตัวอย่าการใช้งาน InputMode ต่อไปนี้จะทำการอ่านค่าข้อมูล 10 ไบต์จากพอร์ตอนุกรมและเก็บข้อมูลไว้ในตัวแปรแบบอาร์เรย์ ชนิดข้อมูลเป็นแบบไบต์

```
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
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4.3.9 Output

ใช้ในการส่งขบวนข้อมูลไปยังบัฟเฟอร์ส่งข้อมูล

รูปแบบการใช้งาน

Object.output[=value]

ค่า value เป็นค่าของตัวอักษรที่เขียนไปยังบัฟเฟอร์ส่งข้อมูล คุณสมบัติ Output สามารถใช้ในการส่งข้อมูลตัวอักษรหรือข้อมูลไบนารีก็ได้ โดยการส่งข้อมูลเป็นรูปแบบตัวอักษรจะต้องกำหนดข้อมูลตัวอักษรเป็นแบบ Variant และมีข้อมูลภายในเป็นแบบ String สำหรับการส่งข้อมูลไบนารี่ จะต้องกำหนดชนิดของข้อมูลเป็นแบบ Variant และข้อมูลภายในแบบ Byte

ตัวอย่างของโปรแกรการส่งค่าที่ป้อนจากคีย์บอร์ดไปยังพอร์ตอนุกรม โดยการใช้คุณสมบัติ Output

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

2.5 คำสั่งที่ใช้ติดต่อกับ Motion Control

ตัวโมชั่นคอนโทรลที่เรานำมาใช้งานนี้เป็นผลงานที่ รศ. ประภาษ อุคคกิมาพันธุ์ ได้ทำการวิจัยสร้างขึ้นมา โดยใช้ตัวไมโครคอนโทรลเลอร์ของ Philips รุ่น 89C51 RD+ เป็นตัวประมวลผล ตัวโมชั่นคอนโทรลนี้ใช้การเชื่อมต่อการสื่อสารผ่านพอร์ตอนุกรมโดยการใช้รหัส ASCII ในการส่งข้อมูล ตัวโมชั่นคอนโทรล สามารถควบคุมการทำงานของมอเตอร์ได้สองแกนและการทำงานจะเป็นแบบเพิ่มพูน (Increment) โดยมีคำสั่งในการใช้งานดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2.2 แสดงผลของโมชั่นคอนโทรลเมื่อป้อนคำสั่ง 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 t0 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 H เป็นคำสั่งที่ป้อนเพื่อดูรูปแบบการใช้งานคำสั่งต่างๆที่ต้องใช้กับตัวโมชั่คอนโทรล เมื่อป้อนคำสั่งก็จะแสดงผลออกมาดังนี้ ตามตารางที่ 2.2

- คำสั่ง T เป็นคำสั่งการป้อนค่าพิกัดและอัตราความเร็วที่ใช้ในการเคลื่อนที่ การป้อนคำสั่งรูปแบบมีดังนี้ [T y z f a]

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยที่ค่า y คือการหมุนของมอเตอร์ A

ค่า z คือการหมุนของมอเตอร์ B

ค่า f คืออัตราความเร็วในการเคลื่อนที่ มีหน่วยเป็น step / clock ของโมชั่น

ค่า a คือค่าของความเร่ง โดยค่าความเร่งที่แท้จริงจะเป็นส่วนกลับกับ a

ตัวอย่าง จะเป็นการใช้งานเพื่อให้มอเตอร์ A เคลื่อนที่ตามเข็มไป 2500 สเต็ป มอเตอร์ B เคลื่อนที่ไปตามเข็ม 2500 สเต็ป มีความเร็วเท่ากับ 10,000 สเต็ปต่อ 1 คล็อกของตัวโมชั่นคอนโทรล มีความเร่งเท่ากับ 500

T 2500 -2500 10000 500

หมายเหตุ เครื่องหมายของ z ติดลบเพราะมอเตอร์ B ถูกติดตั้งกลับด้าน

- คำสั่ง G เป็นคำสั่งที่ป้อนให้โมชั่นคอนโทรลเพื่อเริ่มการเคลื่อนที่ไปตามที่กำหนดไว้
- 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 เป็นคำสั่งในกรณีที่เราต้องการเปลี่ยนแปลงเฉพาะค่าความเร็วหรือความเร่งเท่านั้น

  รูปแบบคำสั่งจะเป็น F f a

  เช่น F 10000 100

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

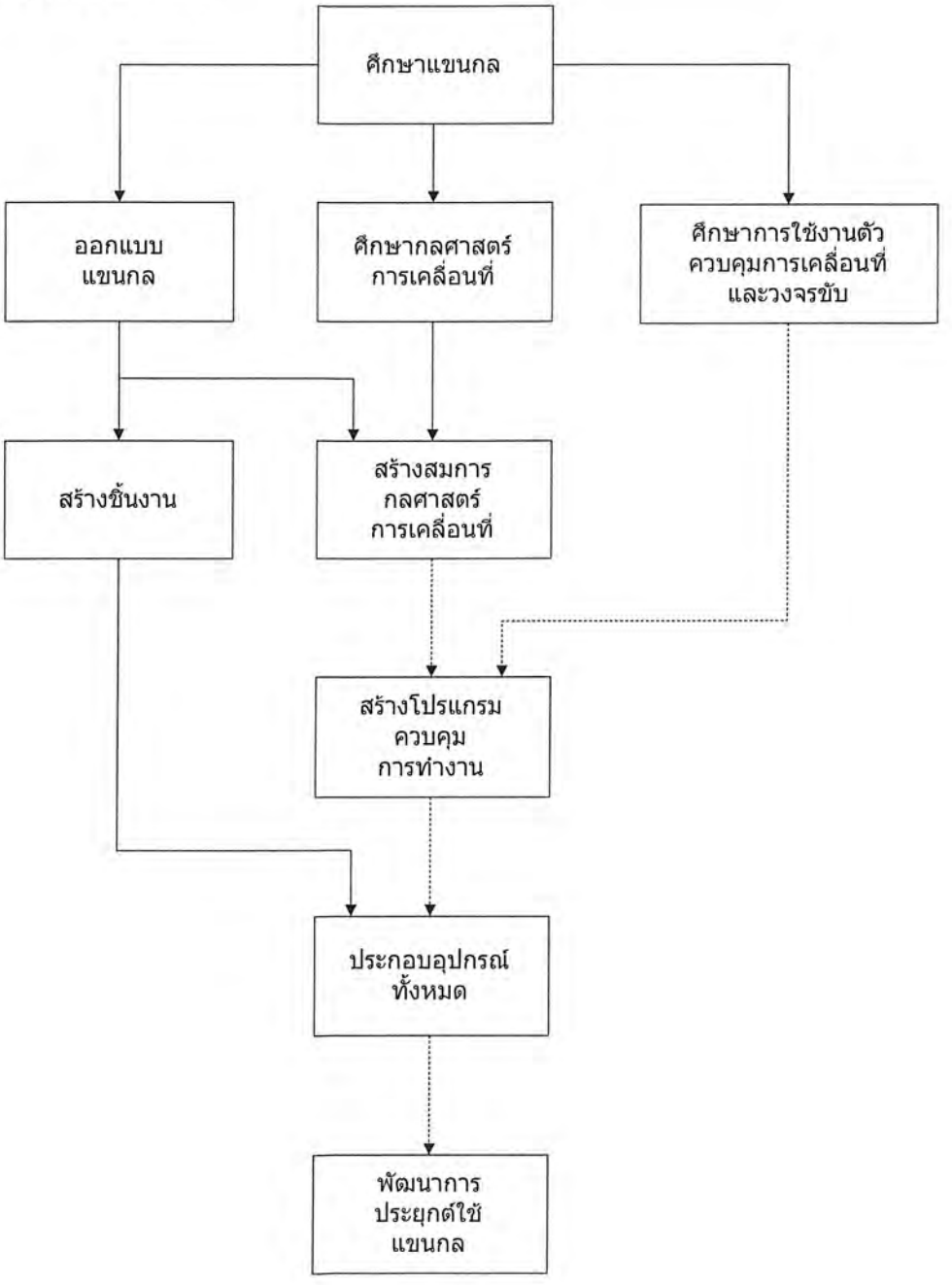
- คำสั่ง I จะเป็นคำสั่งเริ่มต้นที่ใช้ป้อนชุดของคำสั่ง เข้าไปในบัฟเฟอร์ของตัวโมชั่นคอนโทรล และจบชุดคำสั่งด้วยเครื่องหมาย "%" ตัวอย่าง เป็นการป้อนชุดคำสั่งให้มอเตอร์ทั้งสองตัวเคลื่อนที่ไป แล้วกลับมายังตำแหน่งเดิม โดยมีจุดมุ่งหมายเพื่อทำการทดสอบความแม่นยำของตัวแขนกล หากใช้การป้อนทีละคำสั่งย่อมเป็นการลำบาก

```
I
F 20000 200
T 2500 0
G
T 0 2500
G
T –2500 0
G
T 0 –2500
G
%
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3
การคำนวณและการสร้าง

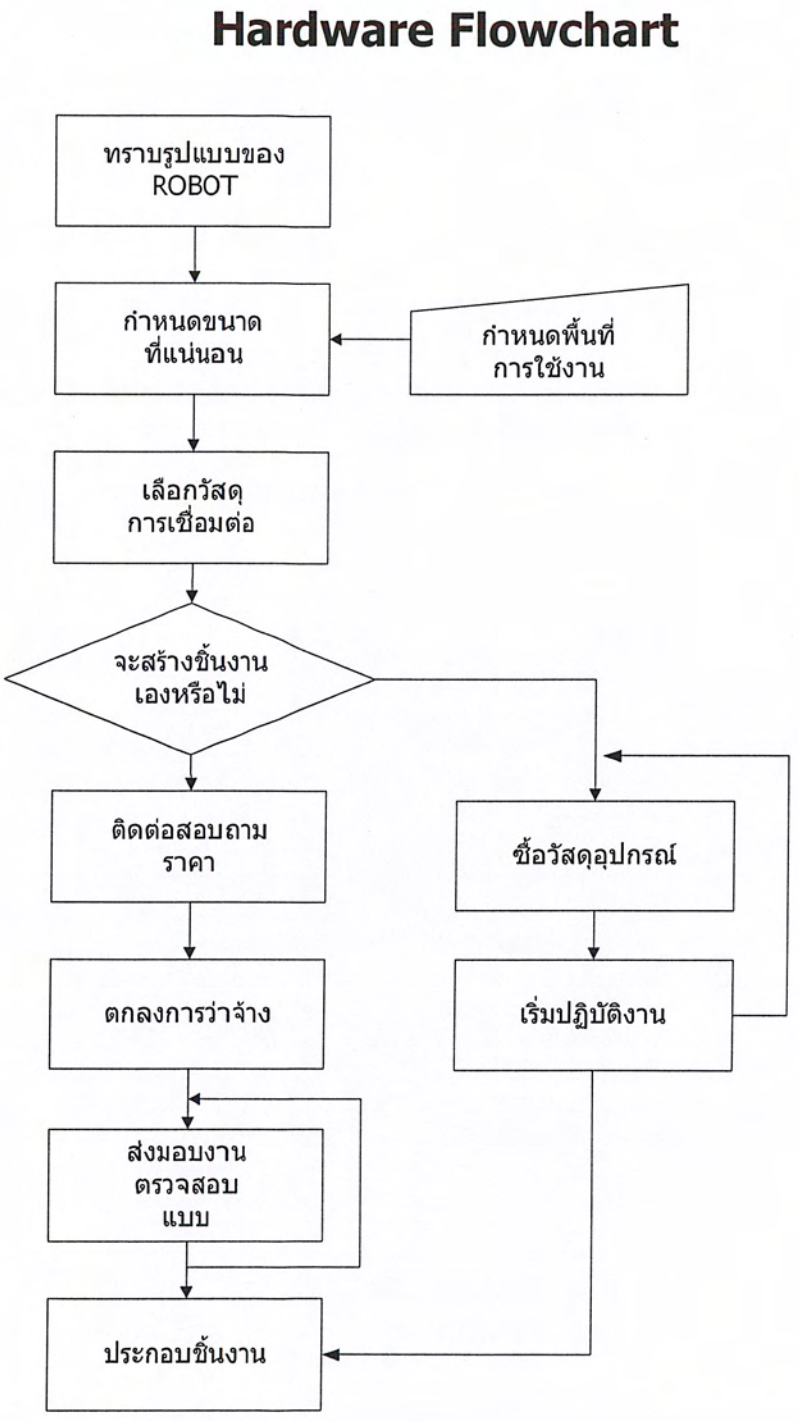
3.1 Project Flowchart



ภาพที่ 3.1 แสดงProject Flowchart

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

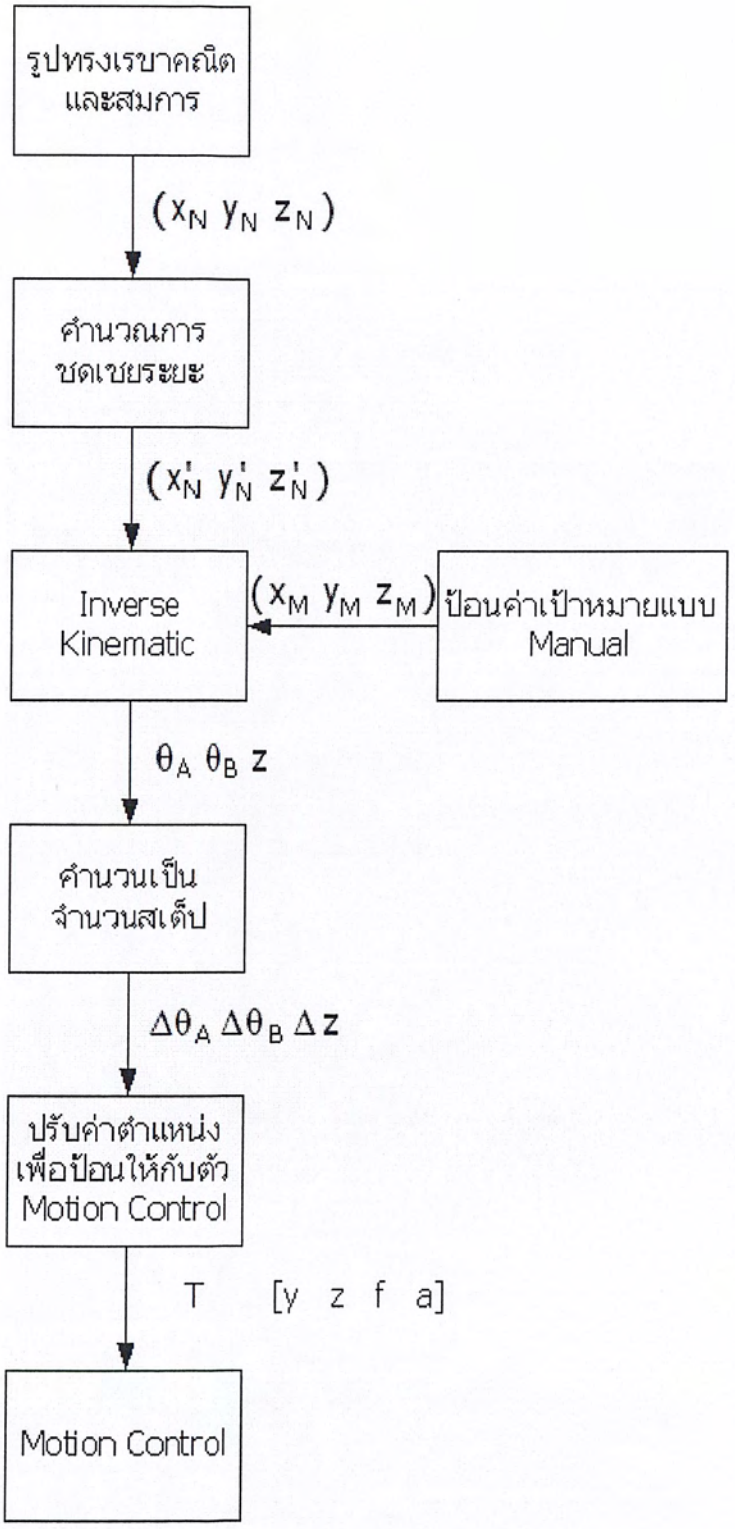
3.2 Hardware Flowchart



ภาพที่ 3.2 Hardware Flowchart

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

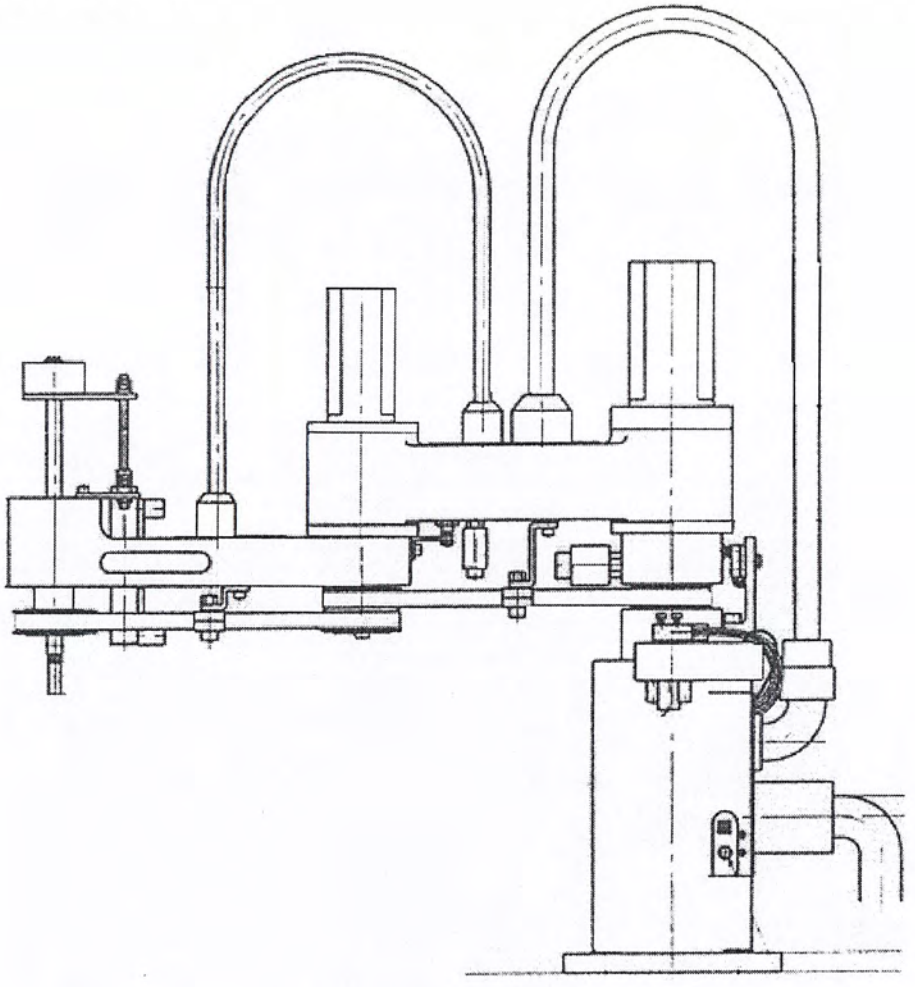
3.3 Software Flowchart



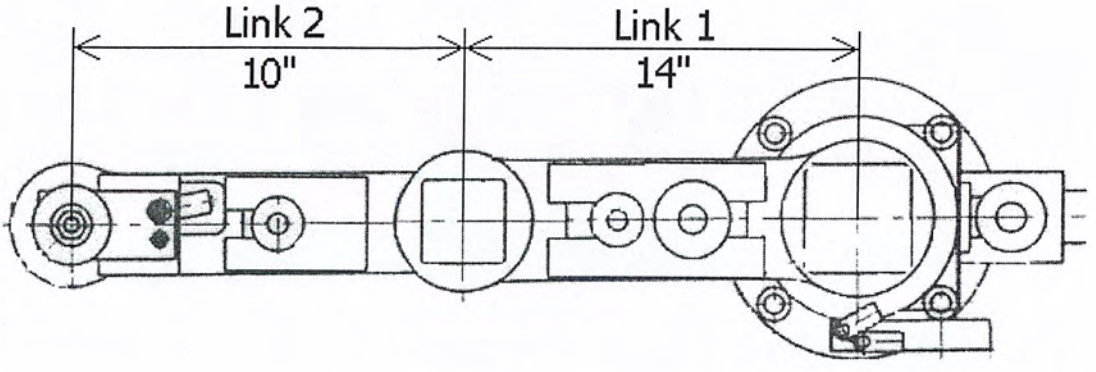
ภาพที่ 3.3 Software Flowchart

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.4 ลักษณะโครงสร้าง

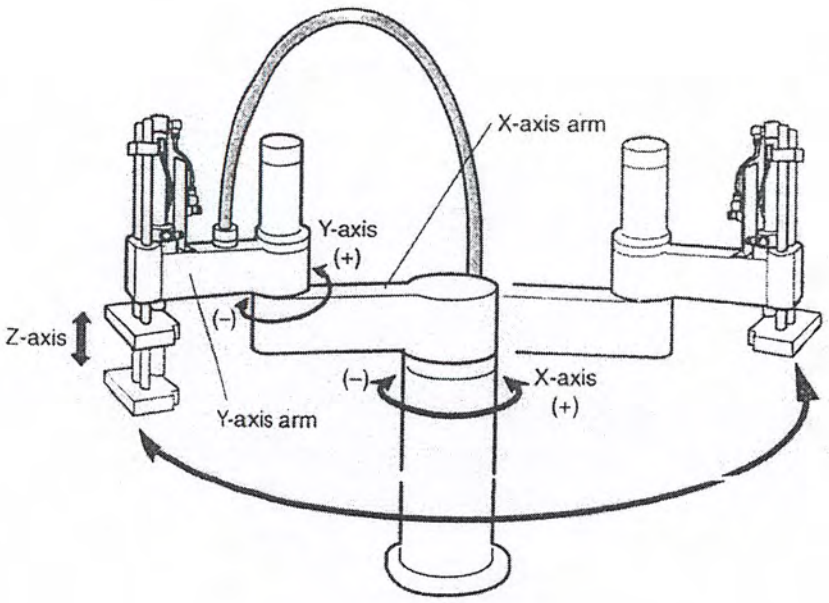


ภาพที่ 3.4 ลักษณะโดยทั่วไป

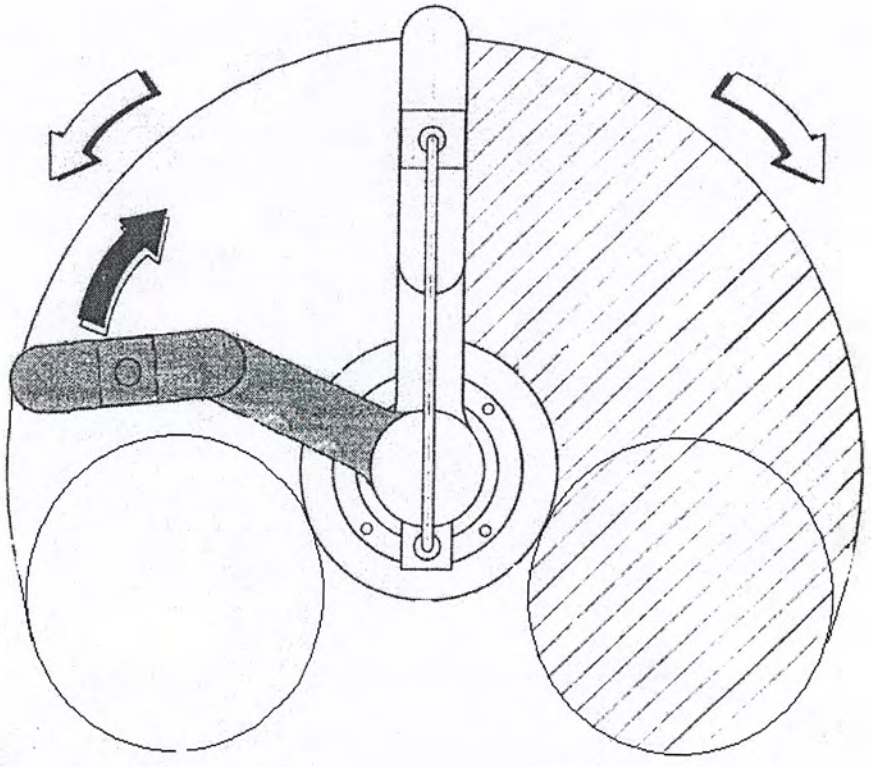


ภาพที่ 3.5 ขนาดความยามของแต่ละ Link

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

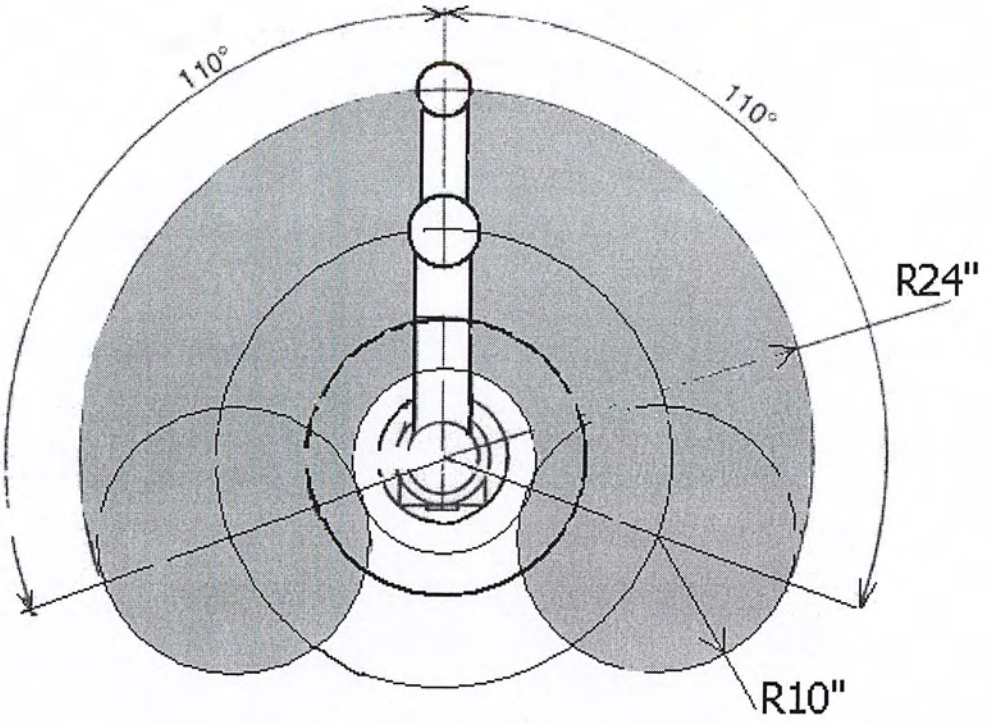


ภาพที่ 3.6 ลักษณะการเคลื่อนที่ของแขน



ภาพที่ 3.7 ลักษณะการเคลื่อนที่เมื่อมองจากด้านบน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



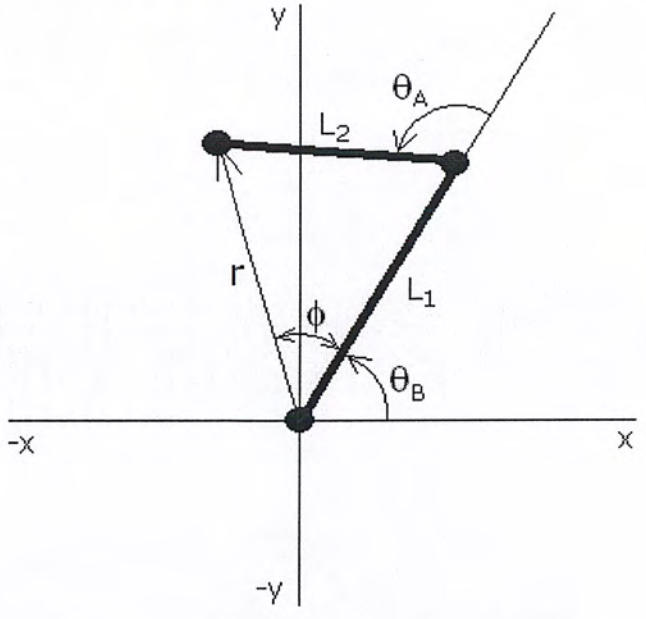
ภาพที่ 3.8 ขอบเขตของการเคลื่อนที่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

การทดลอง

4.1 สมการการคำนวณค่าพิกัด



ภาพที่ 4.1 แสดงมุมของการเคลื่อนที่

4.1.1 การคำนวณการเคลื่อนที่โดยตรง

$$x = L_1 \cos\theta_B - L_2 \cos(180 - \theta_A - \theta_B)$$

$$y = L_1 \sin\theta_B + L_2 \sin(180 - \theta_A - \theta_B)$$

จากสมการที่ได้ สามารถนำมาเขียนโปรแกรมการคำนวณ จากการทดลองการทำงานของโปรแกรม ตัวอย่างการคำนวณได้ค่าดังนี้

ตารางที่ 4.1 การคำควณการเคลื่อนที่โดยตรงจากสมการและโปรแกรม

ค่ามุมที่ป้อน		ค่าพิกัดที่คำนวณจากสมการ		ค่าพิกัดที่คำนวนจากโปรแกรม	
θ_A	θ_B	x	y	x	y
90	90	-10	14	-10.00006	13.99999
60	120	-17	12.124356	-17.00008	12.12431
30	45	12.487685	19.558753	12.487716	19.558962
25	145	-21.316206	9.766552	-21.316268	9.7664663

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.1.2 การคำนวณการเคลื่อนที่แบบผันกลับ

$$\theta_A = 180 - \cos^{-1}\left(\frac{r^2 - L_1^2 - L_2^2}{-2L_1L_2}\right)$$

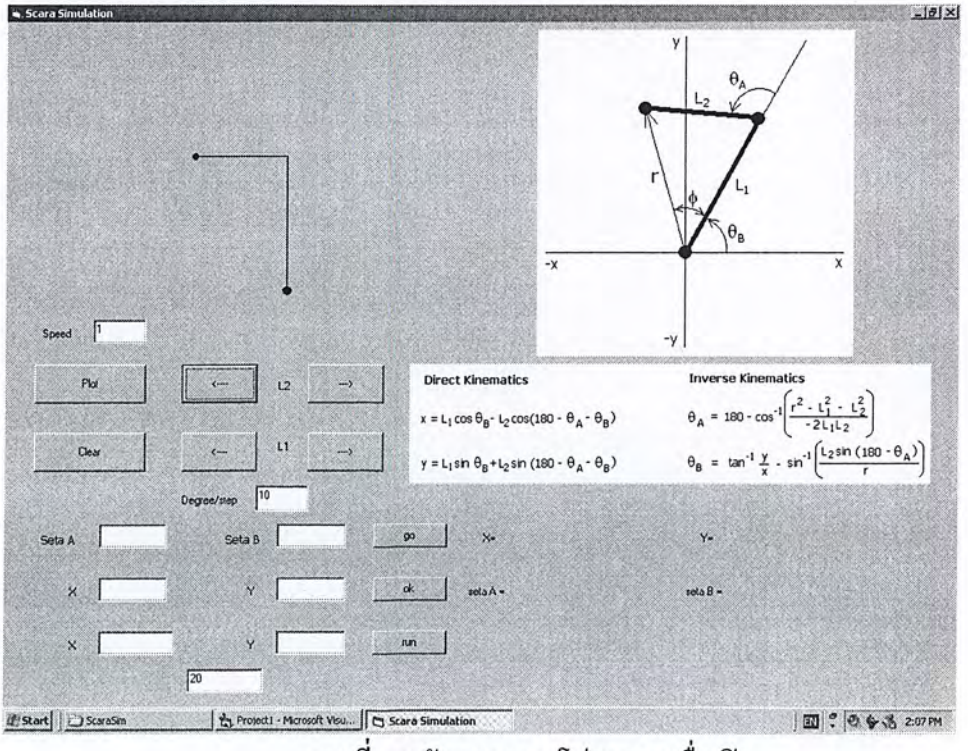
$$\theta_B = \tan^{-1}\frac{y}{x} - \sin^{-1}\left(\frac{L_2 \sin(180 - \theta_A)}{r}\right)$$

จากสมการที่ได้ นำมาเขียนโปรแกรมคำนวณ จากการทดลองการทำงานของโปรแกรมตัวอย่างการคำนวณได้ค่าดังนี้

ตารางที่ 4.2 การคำควณการเคลื่อนที่แบบผันกลับจากสมการและโปรแกรม

ค่าพิกัด		ค่ามุมที่คำนวณจากสมการ		ค่ามุมที่คำนวณจากโปรแกรม	
x	y	θ_A	θ_B	θ_A	θ_B
-10	10	110.0510426	93.375393	110.051206137	93.37561329
-5	20	62.56679134	78.52540772	62.5670659457	78.535728462
8	6	134.427004	-8.73098354	134.42711057	-8.703143516
11.7	17.2	60.76968647	30.97176522	60.7699652778	30.971673534

4.2 ลักษณะของของโปรแกรม

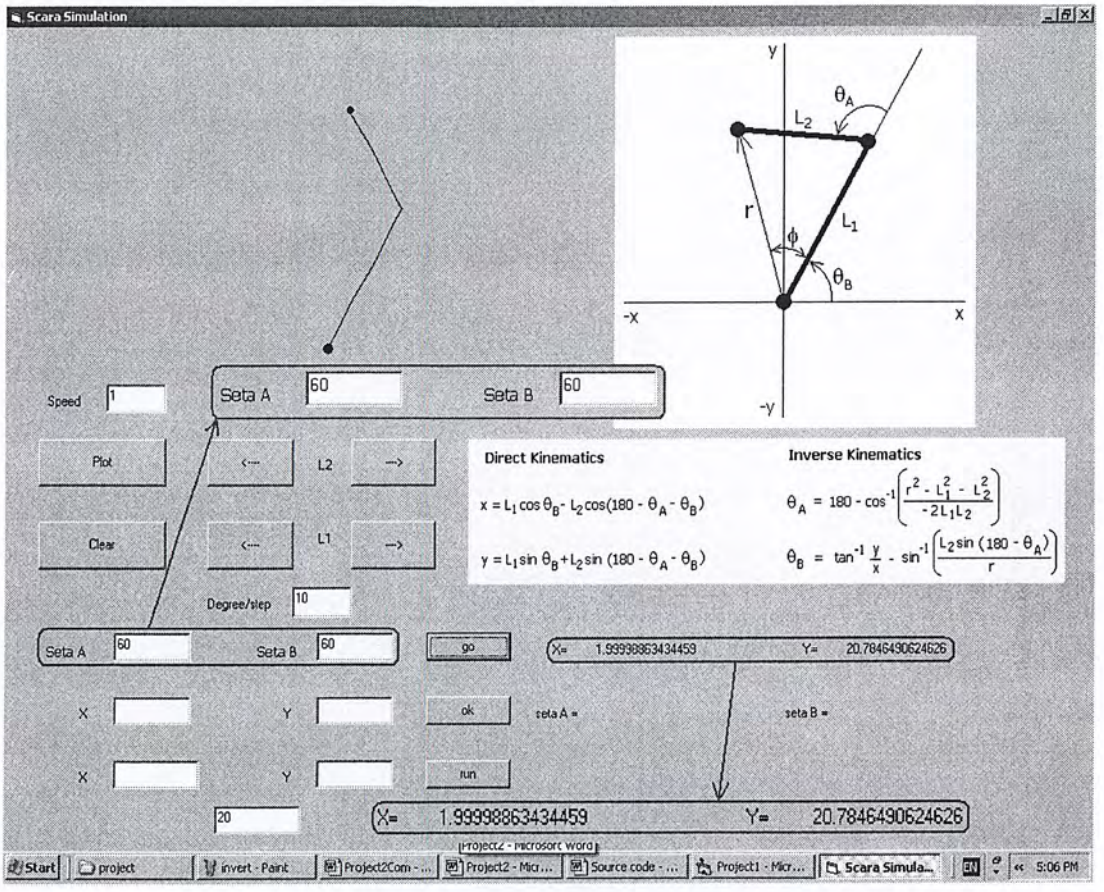


ภาพที่ 4.2 ลักษณะของโปรแกรมเมื่อเปิด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2.1 การใช้โปรแกรมคำนวณการเคลื่อนที่โดยตรง

การกำหนดพิกัด โดยการใช้สมการการเคลื่อนที่โดยตรงนั้น จะป้อนค่าของมุมที่จะให้แขนแต่ละส่วนที่เคลื่อนที่ไป

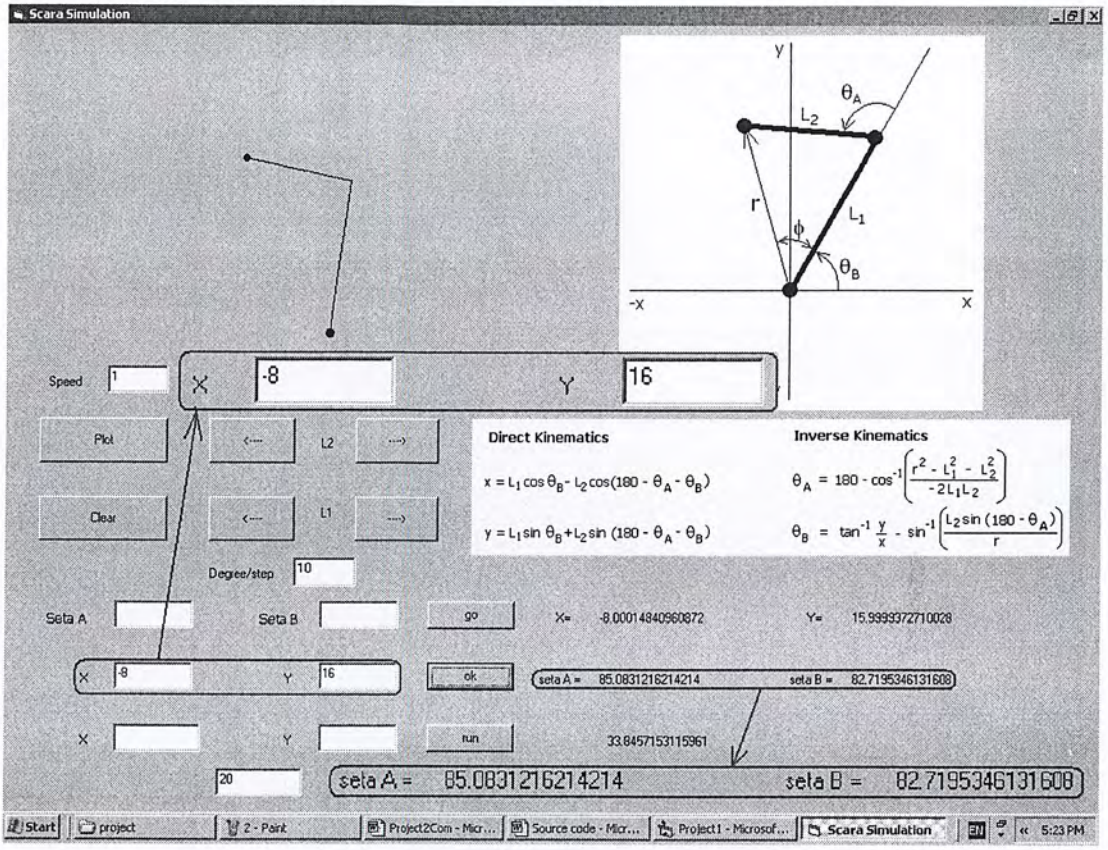


ภาพที่ 4.3 การใช้โปรแกรมคำนวณการเคลื่อนที่โดยตรง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2.2 การใช้โปรแกรมคำนวณการเคลื่อนที่แบบผันกลับ

การเคลื่อนที่แบบผันกลับนั้นเป็นการระบุพิกัดที่จะให้แขนเคลื่อนที่ไปที่จุดนั้นๆ โดยจะป้อนค่าพิกัด x,y แล้วโปรแกรมจะคำนวณให้อยู่ในรูปของมุมที่เคลื่อนที่

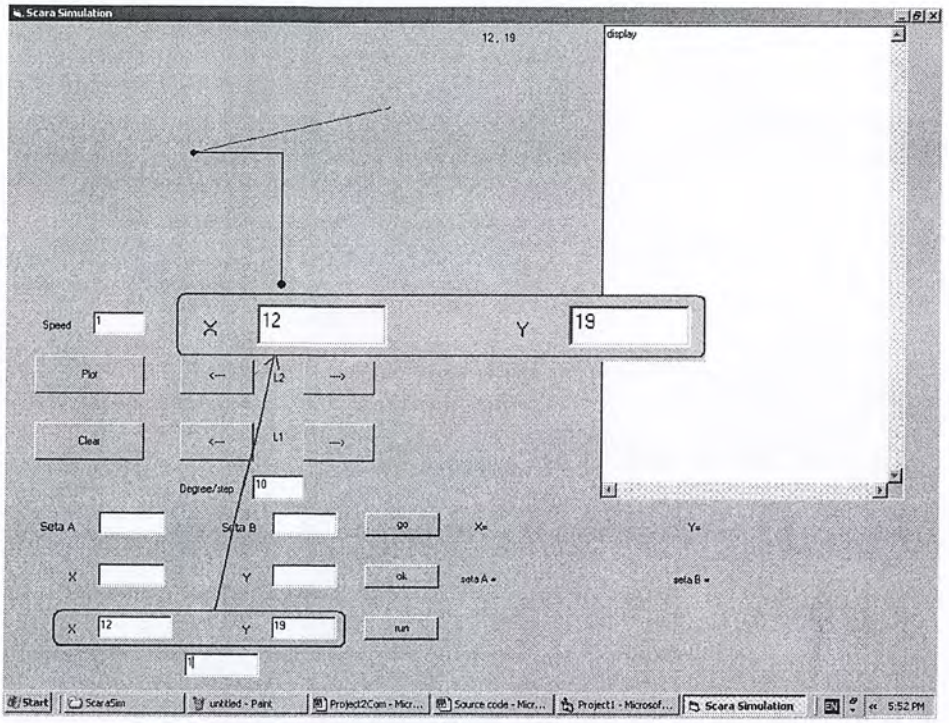


ภาพที่ 4.4 การใช้โปรแกรมคำนวณการเคลื่อนที่แบบผันกลับ

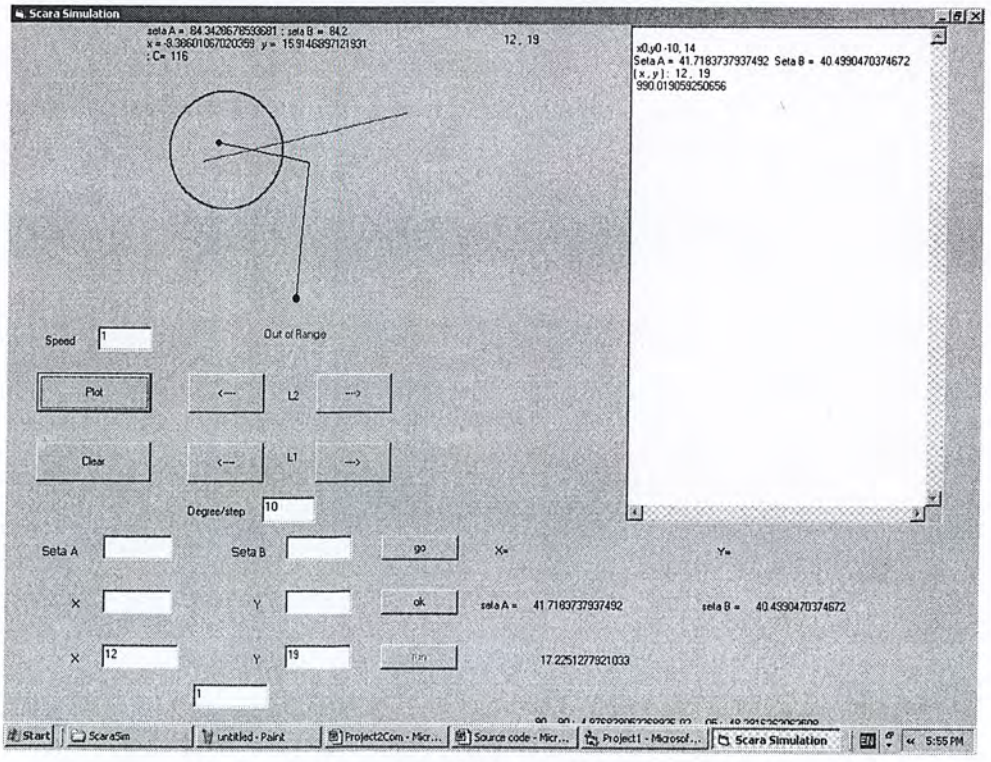
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2.3 การเคลื่อนที่แบบเส้นตรง

จากสมการ Inverse Kinematics จะคำนวณได้ที่ละจุด ถ้าป้อนค่าที่จุดที่ต้องการให้กับแขนเคลื่อนที่ไปยังจุดที่เราต้องการเพียงจุดๆเดียว ลักษณะของการเคลื่อนที่ ก็จะเคลื่อนที่ไปเป็นแนวโค้ง เนื่องจากสมการการเคลื่อนที่ไม่เป็นเส้นตรง

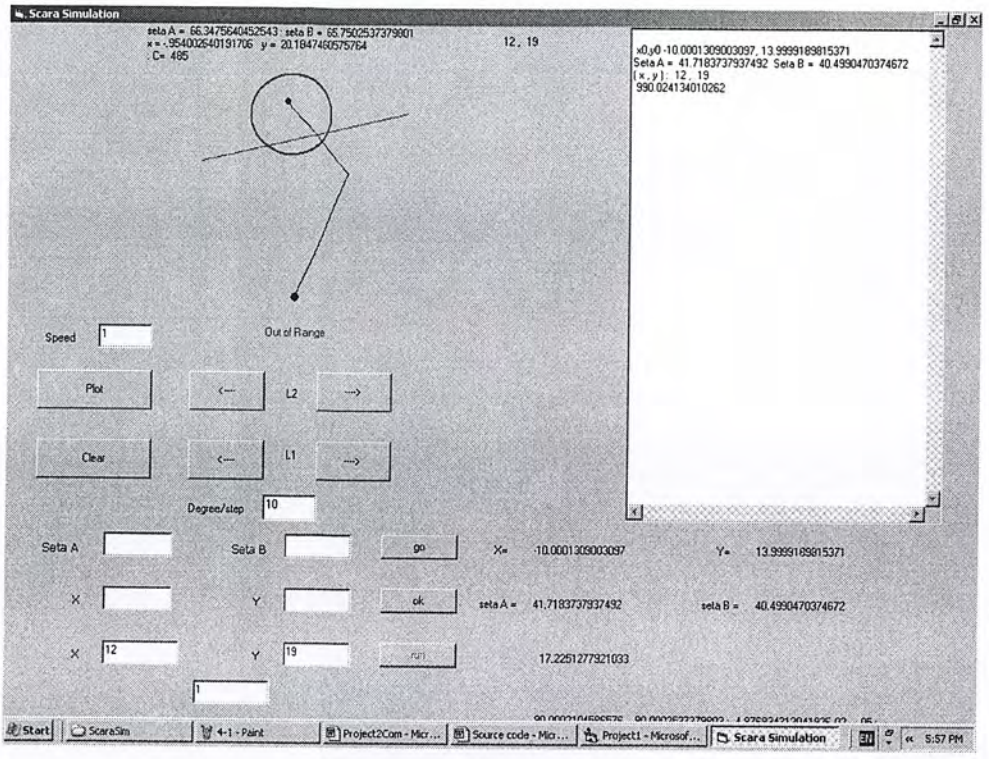


(a)

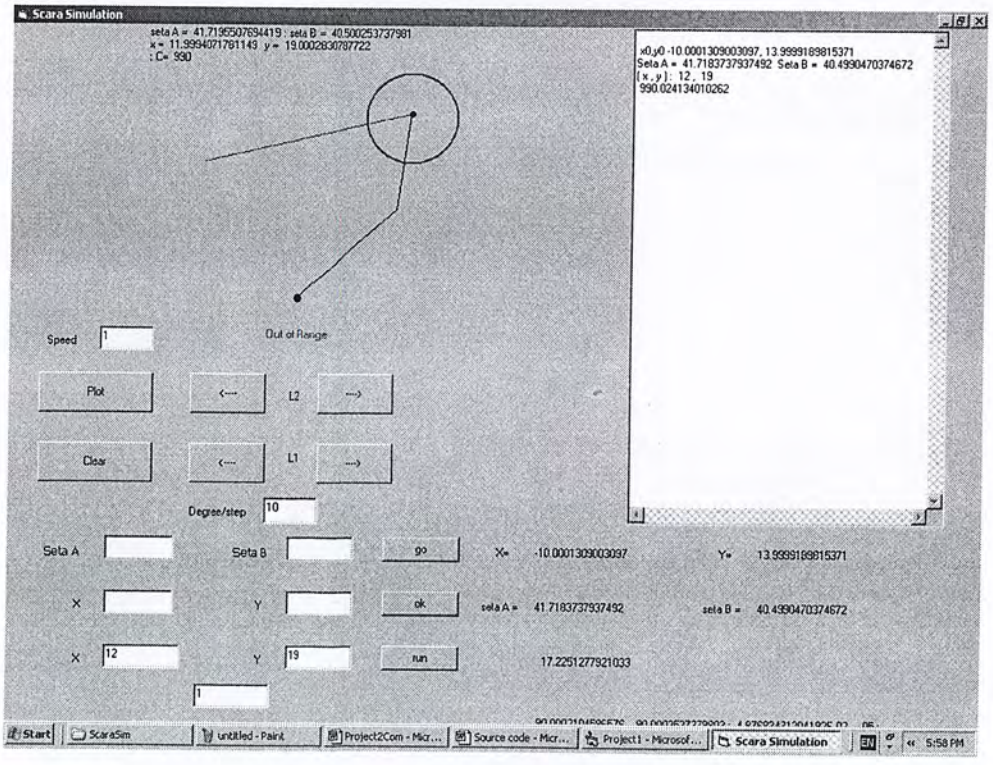


(b)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



(c)

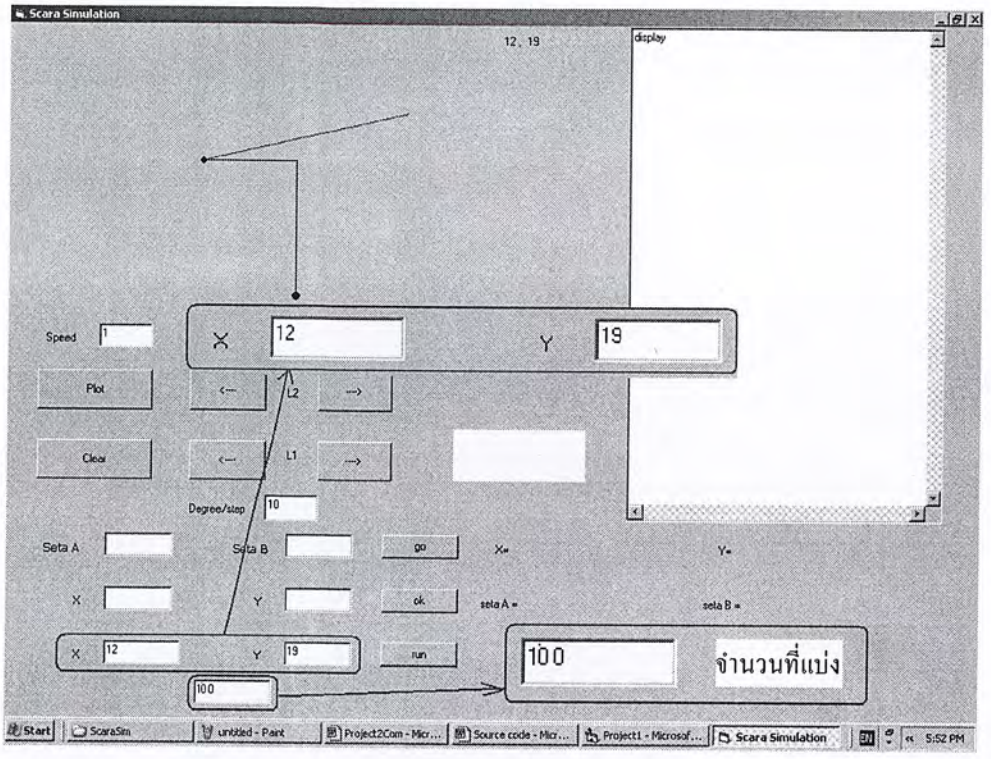


(d)

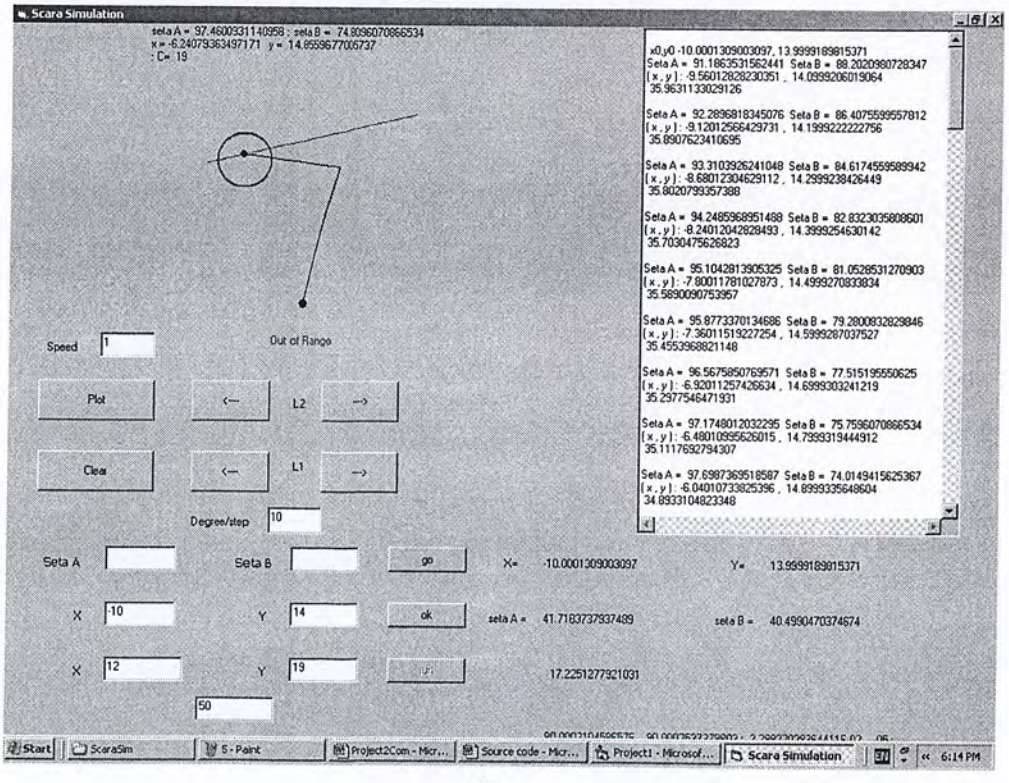
ภาพที่ 4.5 การเคลื่อนที่แบบกำหนดจุดปลายจุดเดียว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

วิธีแก้คือ แบ่งเส้นตรงนั้นให้สั้นๆแล้วคำนวณจุดปลายของแต่ละเส้นแล้วให้แขนเคลื่อนที่ไปยังจุดปลายของแต่ละเส้น จะได้การเคลื่อนที่ที่เป็นเส้นตรง

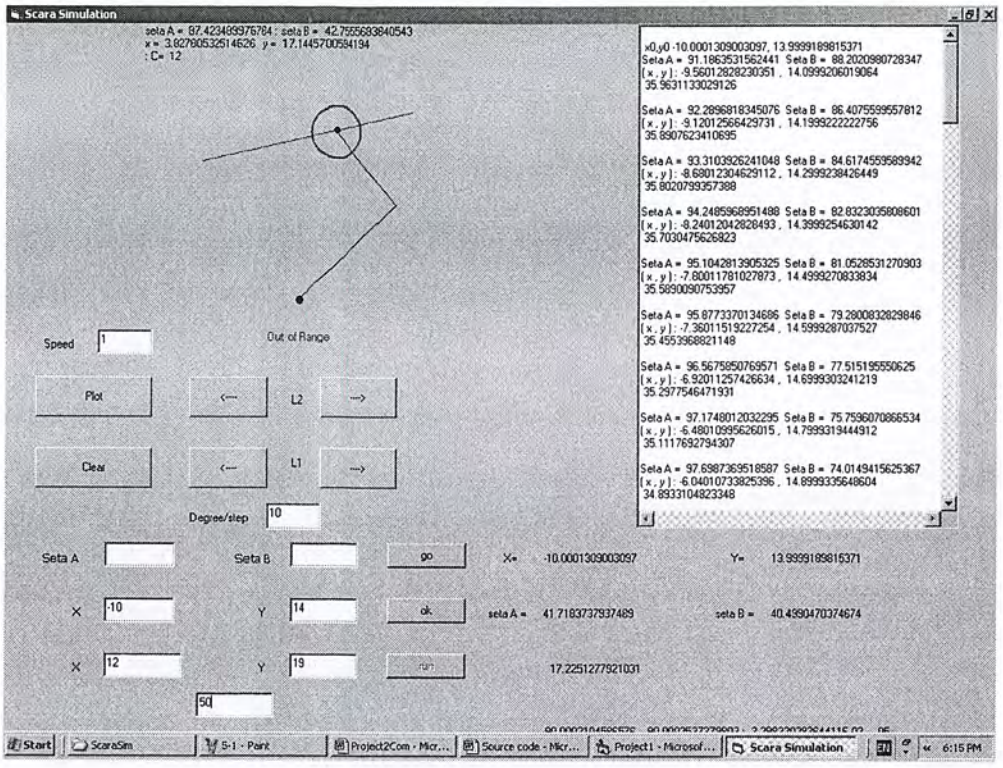


(a)

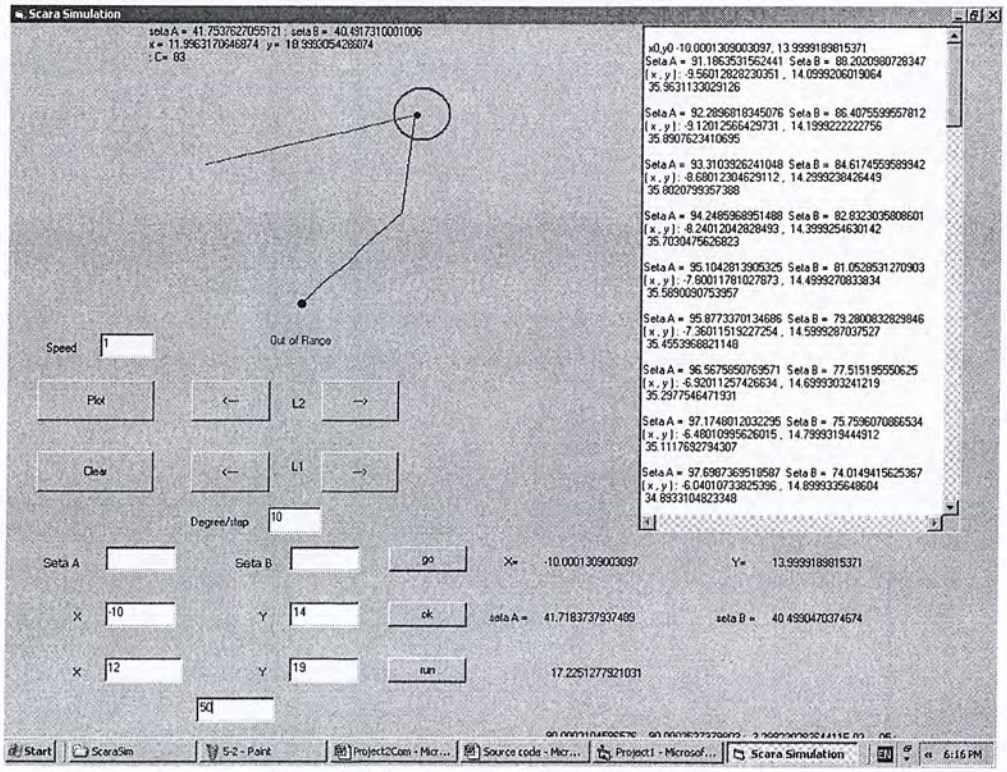


(b)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



(c)



(d)

ภาพที่ 4.6 การเคลื่อนที่แบบเส้นตรง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5
สรุปผลการวิจัยและข้อเสนอแนะ

5.1 ปัญหาและอุปสรรค

1. ขาดประสบการณ์ ในการใช้เครื่องมือและเครื่องจักร
2. ขาดประสบการณ์ ในการออกแบบ ทำให้เกิดความยุ่งยากในการทำชิ้นงาน เช่น การบอกขนาดของแบบโดยไม่อ้างอิงจากมุมใดมุมหนึ่งทำให้ไม่สามารถกำหนดจุดอ้างอิงที่แน่นอนได้
3. เกิดปัญหาการสั่นที่จุดปลายของแขนกล เนื่องจากการสั่นของตัวมอเตอร์ทำให้ไม่สามารถใช้ความเร็วในการเคลื่อนที่ที่มากได้
4. เกิดปัญหาในเรื่องของพื้นที่ในการทำงานที่เกิดมาจากฐานรองมอเตอร์และตัวฐานไปจำกัดการเคลื่อนที่ของแขนกลเอง
5. ขอบเขตการเคลื่อนที่เนื่องจากความยาวของแขน ส่งผลถึงพื้นที่การทำงาน
6. โครงสร้างส่วนฐานยึดมอเตอร์ (Motor Base) และฐานด้านหลัง (Base) ขวางการเคลื่อนที่ของแขนกลและอุปกรณ์ที่ติดตั้งอยู่ที่ส่วนปลายของแขนกล
7. Error จากขนาดของขนาดแขนError จากการปัดเศษในการคำนวณทาง Software

5.2 ข้อเสนอแนะและแนวทางการพัฒนา

1. ควรปรับปรุงโครงสร้างเพื่อให้แขนกลมีพื้นที่ในการทำงานได้เต็มที่
2. ควรเขียนเงื่อนไขให้ครอบคลุมพื้นที่ที่แขนกลไม่เคลื่อนที่ไปยังจุดที่ไม่ต้องการหรืออาจจะเกิดอันตรายจากการเคลื่อนที่ไปยังจุดนั้นๆ
3. แขนกลนี้สามารถนำไปพัฒนาประยุกต์ใช้กับงานที่ควบคุมทางเดิน หรือตำแหน่ง ตัวอย่าง เช่น เป็นเครื่องมือ, เครื่องเชื่อม, เครื่องตัดหรือการจับชิ้นงานแบบอัตโนมัติได้
4. เพิ่มอุปกรณ์หรือเครื่องมือที่ปลายแกน z เพื่อประยุกต์ใช้งานตามที่ต้องการ
5. นำระบบการประมวลผลภาพมาประยุกต์ใช้ในการควบคุม

5.3 สรุป

จากการศึกษาการทำงานของแขนกลแบบต่างๆ ที่ได้รวบรวมมาจากแหล่งข้อมูลจากที่ต่างๆ แล้วนำมาออกแบบการควบคุมแขนกล เพื่อเป็นกรณีศึกษาการควบคุมและสามารถนำมาสร้างได้จริง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยแบ่งการศึกษาออกเป็น 3 ส่วนใหญ่ดังนี้

1. ส่วนโครงสร้าง (Hardware) เป็นแขนกล 3 แกน (Scara Robot) ที่ใช้ในโรงงานอุตสาหกรรม
2. ส่วนการควบคุมการเคลื่อนที่ (Motion Control) ประกอบไปด้วย
    a. ส่วนควบคุมการเคลื่อนที่ (Motion Control)
    b. ชุดขับมอเตอร์
    c. อุปกรณ์ป้อนกลับตำแหน่ง
    d. Servo Motor
3. ส่วนโปรแกรม ใช้โปรแกรมวิชวลเบสิกสร้างโปรแกรมในการประยุกต์การใช้งานเพื่อเพิ่มประสิทธิภาพการใช้งาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

[1] Stadler, Wolfram. "Analytical robotics and mechatronics." New York: McGraw-Hill, c1995

[2] EUROPA LEHRMITTEL." Tabellenbuch Metall." Mourney, Vollmer GmbH & Co. Dusselberger StraBe 23 42781 Haan-Gruiten

[3] กิตติ ภักดีวัฒนะกุล และ จำลอง ครูอุตสาหะ, "Visual Basic ฉบับโปรแกรมเมอร์", กรุงเทพฯ: ไทยเจริญการพิมพ์, 2544

[4] กฤษดา ใจเย็น,อรรถพล บุญยะโภคา, ชัยวัฒน์ ลิ้มพรวิไล, "เรียนรู้และปฏิบัติการเชื่อมต่อคอมพิวเตอร์กับอุปกรณ์ภายนอกผ่านพอร์ตอนุกรม,บริษัท อินโนเวตีฟ เอ็กเพอริเมนต์ จำกัด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Visual Basic Source Code

```
Dim a, b, c, d, e, f, g, h, i, j, r, xout, yout, xin, yin, xgo, ygo, adiv, ydif, astep, bstep As Double
Dim x12, y12, x22, y22, yr, xr, alfa, beta, fie, fies, step, xstep(0 To 100), ystep(0 To 100) As Double
Dim afac(0 To 1000), bfac(0 To 1000), xpdiv, ypdiv, xgot, ygot As Double
Dim maxpoint As Integer
Dim xp(0 To 1000), yp(0 To 1000) As Double
Dim setaa(0 To 1000), setab(0 To 1000), maxstep(0 To 1000) As Double
Private Sub Command1_Click()

  For d = 0 To 40
    Line4(d).Visible = False
  Next
  Line3.Visible = False
  a = a - Val(Text10.Text)
  setaa(0) = 90 - a
  setab(0) = 90 - b
  y12 = Line2.Y2 - 2100 * Sin((90 - b) / 57.2956) - 1500 * Sin((180 - a - b) / 57.2956)
  x12 = Line2.X2 + 2100 * Cos((90 - b) / 57.2956) + 1500 * Cos((180 - a - b) / 57.2956)

  Line1.Y2 = y12
  Line1.X2 = x12
  Shape2.Top = Line1.Y2 - 50
  Shape2.Left = Line1.X2 - 50

  xp(0) = 14 * Cos((90 - b) / 57.2956) - 10 * Cos((-a - b) / 57.2956)
  yp(0) = 14 * Sin((90 - b) / 57.2956) - 10 * Sin((-a - b) / 57.2956)
  Label3.Caption = xp(0)
  Label4.Caption = yp(0)
End Sub
Private Sub Command2_Click()
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    For d = 0 To 40
       Line4(d).Visible = False
    Next
    a = a + Val(Text10.Text)
    Line3.Visible = False
    setaa(0) = 90 - a
    setab(0) = 90 - b
    y12 = Line2.Y2 - 2100 * Sin((90 - b) / 57.2956) - 1500 * Sin((180 - a - b) / 57.2956)
    x12 = Line2.X2 + 2100 * Cos((90 - b) / 57.2956) + 1500 * Cos((180 - a - b) / 57.2956)

    Line1.Y2 = y12
    Line1.X2 = x12
    Shape2.Top = Line1.Y2 - 50
    Shape2.Left = Line1.X2 - 50


    xp(0) = 14 * Cos((90 - b) / 57.2956) - 10 * Cos((-a - b) / 57.2956)
    yp(0) = 14 * Sin((90 - b) / 57.2956) - 10 * Sin((-a - b) / 57.2956)
    Label3.Caption = xp(0)
    Label4.Caption = yp(0)

End Sub

Private Sub Command3_Click()
   For d = 0 To 40
      Line4(d).Visible = False
   Next
   Line3.Visible = False
   b = b + Val(Text10.Text)
   setaa(0) = 90 - a
   setab(0) = 90 - b
   x22 = Line2.X2 + 2100 * Cos((90 - b) / 57.2956)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    y22 = Line2.Y2 - 2100 * Sin((90 - b) / 57.2956)
    Line2.X1 = x22
    Line2.Y1 = y22
    y12 = Line2.Y2 - 2100 * Sin((90 - b) / 57.2956) - 1500 * Sin((180 - a - b) / 57.2956)
    x12 = Line2.X2 + 2100 * Cos((90 - b) / 57.2956) + 1500 * Cos((180 - a - b) / 57.2956)

    Line1.X1 = x22
    Line1.Y1 = y22
    Line1.Y2 = y12
    Line1.X2 = x12
    Shape2.Top = Line1.Y2 - 50
    Shape2.Left = Line1.X2 - 50

    xp(0) = 14 * Cos((90 - b) / 57.2956) - 10 * Cos((-a - b) / 57.2956)
    yp(0) = 14 * Sin((90 - b) / 57.2956) - 10 * Sin((-a - b) / 57.2956)
    Label3.Caption = xp(0)
    Label4.Caption = yp(0)

End Sub

Private Sub Command4_Click()
   For d = 0 To 40
      Line4(d).Visible = False
   Next

   Line3.Visible = False
   b = b - Val(Text10.Text)
   setaa(0) = 90 - a
   setab(0) = 90 - b
   x22 = Line2.X2 + 2100 * Cos((90 - b) / 57.2956)
   y22 = Line2.Y2 - 2100 * Sin((90 - b) / 57.2956)
   Line2.X1 = x22
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    Line2.Y1 = y22
    y12 = Line2.Y2 - 2100 * Sin((90 - b) / 57.2956) - 1500 * Sin((180 - a - b) / 57.2956)
    x12 = Line2.X2 + 2100 * Cos((90 - b) / 57.2956) + 1500 * Cos((180 - a - b) / 57.2956)

    Line1.X1 = x22
    Line1.Y1 = y22
    Line1.Y2 = y12
    Line1.X2 = x12
    Shape2.Top = Line1.Y2 - 50
    Shape2.Left = Line1.X2 - 50

    xp(0) = 14 * Cos((90 - b) / 57.2956) - 10 * Cos((-a - b) / 57.2956)
    yp(0) = 14 * Sin((90 - b) / 57.2956) - 10 * Sin((-a - b) / 57.2956)
    Label3.Caption = xp(0)
    Label4.Caption = yp(0)

End Sub

Private Sub Command5_Click()
    Label18.Caption = ""
    For d = 0 To 40
        Line4(d).Visible = False
    Next
    b = 90 - Val(Text2.Text)
    a = 90 - Val(Text1.Text)
    Line3.Visible = False
    setaa(0) = 90 - a
    setab(0) = 90 - b

    x22 = Line2.X2 + 2100 * Cos((90 - b) / 57.2956)
    y22 = Line2.Y2 - 2100 * Sin((90 - b) / 57.2956)
    Line2.X1 = x22
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    Line2.Y1 = y22
    y12 = Line2.Y2 - 2100 * Sin((90 - b) / 57.2956) - 1500 * Sin((180 - a - b) / 57.2956)
    x12 = Line2.X2 + 2100 * Cos((90 - b) / 57.2956) + 1500 * Cos((180 - a - b) / 57.2956)

    Line1.X1 = x22
    Line1.Y1 = y22
    Line1.Y2 = y12
    Line1.X2 = x12
    Shape2.Top = Line1.Y2 - 50
    Shape2.Left = Line1.X2 - 50

    xp(0) = 14 * Cos((90 - b) / 57.2956) - 10 * Cos((-a - b) / 57.2956)
    yp(0) = 14 * Sin((90 - b) / 57.2956) - 10 * Sin((-a - b) / 57.2956)
    Label3.Caption = xp(0)
    Label4.Caption = yp(0)

    End Sub
Private Sub Command6_Click()
    Label18.Caption = ""
    For d = 0 To 40
      Line4(d).Visible = False
    Next
    xin = Val(Text3.Text)
    yin = Val(Text4.Text)
    Line3.Visible = False

If ((xin ^ 2 + yin ^ 2) >= 16) And ((xin ^ 2 + yin ^ 2) <= 576) And yin >= 0 Then
    r = xin ^ 2 + yin ^ 2
    alfa = -(r - 296) / 280
    setaa(0) = 180 - ((Atn(-alfa / Sqr(-alfa * alfa + 1)) + 2 * Atn(1)) * 180 / 3.1416)
    Label9.Caption = setaa(0)
    fie = (10 / Sqr(r)) * Sin((180 - setaa(0)) / 57.2956)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    fies = (Atn(fie / Sqr(-fie * fie + 1))) * 180 / 3.1416

    If xin <= 0 Then
       setab(0) = (180 - (Atn(yin / (-xin)) * 57.2956)) - fies
    Else
       setab(0) = (Atn(yin / xin) * 57.2956) - fies
    End If

    Label10.Caption = setab(0)
    Label11.Caption = fies

    a = 90 - setaa(0)
    b = 90 - setab(0)
    x22 = Line2.X2 + 2100 * Cos((90 - b) / 57.2956)
    y22 = Line2.Y2 - 2100 * Sin((90 - b) / 57.2956)
    Line2.X1 = x22
    Line2.Y1 = y22
    y12 = Line2.Y2 - 2100 * Sin((90 - b) / 57.2956) - 1500 * Sin((180 - a - b) / 57.2956)
    x12 = Line2.X2 + 2100 * Cos((90 - b) / 57.2956) + 1500 * Cos((180 - a - b) / 57.2956)

    Line1.X1 = x22
    Line1.Y1 = y22
    Line1.Y2 = y12
    Line1.X2 = x12
    Shape2.Top = Line1.Y2 - 50
    Shape2.Left = Line1.X2 - 50

    xp(0) = 14 * Cos((90 - b) / 57.2956) - 10 * Cos((-a - b) / 57.2956)
    yp(0) = 14 * Sin((90 - b) / 57.2956) - 10 * Sin((-a - b) / 57.2956)
    Label3.Caption = xp(0)
    Label4.Caption = yp(0)
  ElseIf MsgBox("Out of Range", vbOKOnly, "Error") = vbOK Then
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    Text3.Text = ""
    Text4.Text = ""
  End If

  End Sub

  Private Sub Command7_Click()
    For d = 0 To 40
      Line4(d).Visible = False
    Next
    Text8.Text = ""
    xgo = Val(Text5.Text)
    ygo = Val(Text6.Text)

    Picture1.Visible = False
    Picture2.Visible = False
    Text8.Visible = True
    Line3.X1 = xp(0) * 150 + 4560
    Line3.Y1 = -yp(0) * 150 + 4200
    Line3.X2 = xgo * 150 + 4560
    Line3.Y2 = -ygo * 150 + 4200
    Line3.Visible = True
    maxpoint = Val(Text7.Text)
    Text8.Text = Text8.Text + Chr(13) + Chr(10) + " x0,y0 " + Str(xp(0)) + "," + Str(yp(0))
    xpdiv = (xgo - xp(0)) / maxpoint
    ypdiv = (ygo - yp(0)) / maxpoint

    h = 0
    For d = 1 To maxpoint
      xp(d) = xp(d - 1) + xpdiv
      yp(d) = yp(d - 1) + ypdiv
If ((xp(d) ^ 2 + yp(d) ^ 2) >= 16) And ((xp(d) ^ 2 + yp(d) ^ 2) <= 576) And yp(d) >= 0 Then
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    r = xp(d) ^ 2 + yp(d) ^ 2
    alfa = -(r - 296) / 280
    setaa(d) = 180 - ((Atn(-alfa / Sqr(-alfa * alfa + 1)) + 2 * Atn(1)) * 180 / 3.1416)
    Label9.Caption = setaa(d)
    fie = (10 / Sqr(r)) * Sin((180 - setaa(d)) / 57.2956)
    fies = (Atn(fie / Sqr(-fie * fie + 1))) * 180 / 3.1416
    If xp(d) < 0 Then
       setab(d) = (180 - (Atn(yp(d) / (-xp(d))) * 57.2956)) - fies
    Else
       setab(d) = (Atn(yp(d) / (xp(d))) * 57.2956) - fies
    End If

    Label10.Caption = setab(d)
    Label11.Caption = fies

    adiv = -(setaa(d) - setaa(d - 1))
    bdiv = -(setab(d) - setab(d - 1))
    afac(d) = 1
    bfac(d) = 1
    If adiv <> 0 Then afac(d) = Abs(adiv) / adiv
    If bdiv <> 0 Then bfac(d) = Abs(bdiv) / bdiv

 If (Abs(adiv)) <= (Abs(bdiv)) Then
    maxstep(d) = Abs(bdiv / 0.05)
    bfac(d) = bfac(d) * 0.05
    afac(d) = adiv / maxstep(d)

 Else
    maxstep(d) = Abs(adiv / 0.05)
    afac(d) = afac(d) * 0.05
    bfac(d) = bdiv / maxstep(d)
 End If
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        Text8.Text = Text8.Text + Chr(13) + Chr(10) + "Seta A = " + Str(setaa(d)) + "  Seta B = " +
    Str(setab(d)) + Chr(13) + Chr(10) + "( x , y ) : " + Str(xp(d)) + " , " + Str(yp(d)) + Chr(13) +
    Chr(10) + Str(maxstep(d)) + Chr(13) + Chr(10)


  Else
    h = 1
  End If


    Next


    If h = 0 Then
      xp(0) = xgo
      yp(0) = ygo


      c = 0
      e = 1
      a = 90 - setaa(0)
      b = 90 - setab(0)


      Label19.Caption = Str(setaa(0)) + " , " + Str(setab(0)) + " : " + Str(afac(1)) + " , " + Str(bfac(1))
+ " : " + Str(adiv) + " , " + Str(bdiv) + " : " + Str(maxstep(1))
      Timer1.Enabled = True
    ElseIf MsgBox("Out of Range", vbOKOnly, "Error") = vbOK Then
      Text5.Text = ""
      Text6.Text = ""
    End If


End Sub


Private Sub Command8_Click()
    Text8.Text = ""
    xgo = xstep(1)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
ygo = ystep(1)

Picture1.Visible = False
Picture2.Visible = False
Text8.Visible = True
Line3.Visible = False
maxpoint = Val(Text7.Text)
Text8.Text = Text8.Text + Chr(13) + Chr(10) + " x0,y0 " + Str(xp(0)) + "," + Str(yp(0))
xpdiv = (xgo - xp(0)) / maxpoint
ypdiv = (ygo - yp(0)) / maxpoint

h = 0
For d = 1 To maxpoint
  xp(d) = xp(d - 1) + xpdiv
  yp(d) = yp(d - 1) + ypdiv

If ((xp(d) ^ 2 + yp(d) ^ 2) >= 16) And ((xp(d) ^ 2 + yp(d) ^ 2) <= 576) And yp(d) >= 0 Then
  r = xp(d) ^ 2 + yp(d) ^ 2
  alfa = -(r - 296) / 280
  setaa(d) = 180 - ((Atn(-alfa / Sqr(-alfa * alfa + 1)) + 2 * Atn(1)) * 180 / 3.1416)
  Label9.Caption = setaa(d)
  fie = (10 / Sqr(r)) * Sin((180 - setaa(d)) / 57.2956)
  fies = (Atn(fie / Sqr(-fie * fie + 1))) * 180 / 3.1416
  If xp(d) < 0 Then
    setab(d) = (180 - (Atn(yp(d) / (-xp(d))) * 57.2956)) - fies
  Else
    setab(d) = (Atn(yp(d) / (xp(d))) * 57.2956) - fies
  End If
  Label10.Caption = setab(d)
  Label11.Caption = fies

adiv = -(setaa(d) - setaa(d - 1))
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        bdiv = -(setab(d) - setab(d - 1))
        afac(d) = 1
        bfac(d) = 1
        If adiv <> 0 Then afac(d) = Abs(adiv) / adiv
        If bdiv <> 0 Then bfac(d) = Abs(bdiv) / bdiv

      If (Abs(adiv)) <= (Abs(bdiv)) Then
        maxstep(d) = Abs(bdiv / 0.05)
        bfac(d) = bfac(d) * 0.05
        afac(d) = adiv / maxstep(d)

      Else
        maxstep(d) = Abs(adiv / 0.05)
        afac(d) = afac(d) * 0.05
        bfac(d) = bdiv / maxstep(d)

      End If

      Text8.Text = Text8.Text + Chr(13) + Chr(10) + "Seta A = " + Str(setaa(d)) + "  Seta B = " +
Str(setab(d)) + Chr(13) + Chr(10) + "( x , y ) : " + Str(xp(d)) + " , " + Str(yp(d)) + Chr(13) + Chr(10)
+ Str(maxstep(d)) + Chr(13) + Chr(10)

Else
    h = 1
End If
  Next

  If h = 0 Then
      xp(0) = xgo
      yp(0) = ygo

      c = 0
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        e = 1
        a = 90 - setaa(0)
        b = 90 - setab(0)

        Label19.Caption = Str(setaa(0)) + " , " + Str(setab(0)) + " : " + Str(afac(1)) + " , " + Str(bfac(1))
        + " : " + Str(adiv) + " , " + Str(bdiv) + " : " + Str(maxstep(1))
        Timer1.Enabled = True
    ElseIf MsgBox("Out of Range", vbOKOnly, "Error") = vbOK Then
        Text5.Text = ""
        Text6.Text = ""
    End If

End Sub

Private Sub Command9_Click()
    Label21.Caption = ""
    i = 0
    For d = 0 To 40
        Line4(d).Visible = False
    Next
End Sub
Private Sub Form_Click()
        xstep(0) = xp(0)
        ystep(0) = yp(0)

    If Label20.Caption <> "Out of Range" Then
        i = i + 1
        If i = 1 Then
            Label21.Caption = ""
            Line3.Visible = False
            For d = 0 To 40
                Line4(d).Visible = False
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        Next
      End If
      xstep(i) = xgot
      ystep(i) = ygot
      xgo = xstep(i)
      ygo = ystep(i)

      Picture1.Visible = False
      Picture2.Visible = False
      Text8.Visible = True
      Line4(i).Visible = True
      Line4(i).X1 = xstep(i - 1) * 150 + 4560
      Line4(i).Y1 = -ystep(i - 1) * 150 + 4200
      Line4(i).X2 = xgo * 150 + 4560
      Line4(i).Y2 = -ygo * 150 + 4200

      Label21.Caption = Label21.Caption + Str(xstep(i)) + " , " + Str(ystep(i)) + Chr(13) + Chr(10)
   End If
End Sub
Private Sub Form_Load()
   a = 0
   b = 0
   c = 0
   i = 0
   j = 0
   seta = 90
   xp(0) = -10
   yp(0) = 14
   setaa(0) = 90
   setab(0) = 90
   Text8.Text = "display"
   Line1.X1 = Line2.X1
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    Line1.Y1 = Line2.Y1
    Line1.X2 = Line1.X1 - 1500
    Line1.Y2 = Line1.Y1
    Shape2.Top = Line1.Y2 - 50
    Shape2.Left = Line1.X2 - 50

End Sub

Private Sub Form_MouseMove(Button As Integer, Shift As Integer, X As Single, Y As Single)
    xgot = (X - 4560) / 150
    ygot = (4200 - Y) / 150

    If ((xgot ^ 2 + ygot ^ 2) >= 16) And ((xgot ^ 2 + ygot ^ 2) <= 576) And ygot >= 0 Then
        Label20.ForeColor = RGB(0, 0, 0)
        Label20.Caption = Str(xgot) + " , " + Str(ygot)
    Else
        Label20.ForeColor = RGB(255, 0, 0)
        Label20.Caption = "Out of Range"

    End If

End Sub

Private Sub Text9_Change()
    Timer1.Interval = Val(Text9.Text)
End Sub

Private Sub Timer1_Timer()
    Command7.Enabled = False
    c = c + 1
    If c <= maxstep(e) Then
        a = a + afac(e)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        b = b + bfac(e)

        Label18.Caption = "seta A = " + Str(90 - a) + " : seta B = " + Str(90 - b) + Chr(13) + Chr(10) +
    "x = " + Str(14 * Cos((90 - b) / 57.2956) - 10 * Cos((-a - b) / 57.2956)) + "   y = " + Str(14 * Sin((90 -
    b) / 57.2956) - 10 * Sin((-a - b) / 57.2956)) + Chr(13) + Chr(10) + ": C= " + Str(c)

        x22 = Line2.X2 + 2100 * Cos((90 - b) / 57.2956)
        y22 = Line2.Y2 - 2100 * Sin((90 - b) / 57.2956)
        Line2.X1 = x22
        Line2.Y1 = y22
        y12 = Line2.Y2 - 2100 * Sin((90 - b) / 57.2956) - 1500 * Sin((180 - a - b) / 57.2956)
        x12 = Line2.X2 + 2100 * Cos((90 - b) / 57.2956) + 1500 * Cos((180 - a - b) / 57.2956)
        Line1.X1 = x22
        Line1.Y1 = y22
        Line1.Y2 = y12
        Line1.X2 = x12
        Shape2.Top = Line1.Y2 - 50
        Shape2.Left = Line1.X2 - 50
    ElseIf e <= maxpoint Then
        e = e + 1
        a = 90 - setaa(e - 1)
        b = 90 - setab(e - 1)
        c = 0

      Else
        Command7.Enabled = True
        Timer1.Enabled = False
        setaa(0) = setaa(maxpoint)
        setab(0) = setab(maxpoint)

j = j + 1
If j <= i Then
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Text8.Text = ""
xgo = xstep(j)
ygo = ystep(j)

Picture1.Visible = False
Picture2.Visible = False
Text8.Visible = True
Line3.X1 = xp(0) * 150 + 4560
Line3.Y1 = -yp(0) * 150 + 4200
Line3.X2 = xgo * 150 + 4560
Line3.Y2 = -ygo * 150 + 4200
Line3.Visible = True
maxpoint = Val(Text7.Text)
Text8.Text = Text8.Text + Chr(13) + Chr(10) + " x0,y0 " + Str(xp(0)) + "," + Str(yp(0))
xpdiv = (xgo - xp(0)) / maxpoint
ypdiv = (ygo - yp(0)) / maxpoint

h = 0
For d = 1 To maxpoint
  xp(d) = xp(d - 1) + xpdiv
  yp(d) = yp(d - 1) + ypdiv

If ((xp(d) ^ 2 + yp(d) ^ 2) >= 16) And ((xp(d) ^ 2 + yp(d) ^ 2) <= 576) And yp(d) >= 0 Then
  r = xp(d) ^ 2 + yp(d) ^ 2
  alfa = -(r - 296) / 280
  setaa(d) = 180 - ((Atn(-alfa / Sqr(-alfa * alfa + 1)) + 2 * Atn(1)) * 180 / 3.1416)
  Label9.Caption = setaa(d)
  fie = (10 / Sqr(r)) * Sin((180 - setaa(d)) / 57.2956)
  fies = (Atn(fie / Sqr(-fie * fie + 1))) * 180 / 3.1416
  If xp(d) < 0 Then
    setab(d) = (180 - (Atn(yp(d) / (-xp(d))) * 57.2956)) - fies
  Else
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            setab(d) = (Atn(yp(d) / (xp(d))) * 57.2956) - fies
        End If

        Label10.Caption = setab(d)
        Label11.Caption = fies

        adiv = -(setaa(d) - setaa(d - 1))
        bdiv = -(setab(d) - setab(d - 1))
        afac(d) = 1
        bfac(d) = 1
        If adiv <> 0 Then afac(d) = Abs(adiv) / adiv
        If bdiv <> 0 Then bfac(d) = Abs(bdiv) / bdiv

    If (Abs(adiv)) <= (Abs(bdiv)) Then
        maxstep(d) = Abs(bdiv / 0.05)
        If maxstep(d) < 1 Then maxstep(d) = 1
        bfac(d) = bfac(d) * 0.05
        afac(d) = adiv / maxstep(d)

    Else
        maxstep(d) = Abs(adiv / 0.05)
        If maxstep(d) < 1 Then maxstep(d) = 1
        afac(d) = afac(d) * 0.05
        bfac(d) = bdiv / maxstep(d)

    End If

    Text8.Text = Text8.Text + Chr(13) + Chr(10) + "Seta A = " + Str(setaa(d)) + "  Seta B = " +
Str(setab(d)) + Chr(13) + Chr(10) + "( x , y ) : " + Str(xp(d)) + " , " + Str(yp(d)) + Chr(13) + Chr(10)
+ Str(maxstep(d)) + Chr(13) + Chr(10)

Else
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    h = 1
  End If

    Next
      If h = 0 Then
      xp(0) = xgo
      yp(0) = ygo

      c = 0
      e = 1
      a = 90 - setaa(0)
      b = 90 - setab(0)
      Label19.Caption = Str(setaa(0)) + " , " + Str(setab(0)) + " : " + Str(afac(1)) + " , " + Str(bfac(1))
+ " : " + Str(adiv) + " , " + Str(bdiv) + " : " + Str(maxstep(1))
      Timer1.Enabled = True
    ElseIf MsgBox("Out of Range", vbOKOnly, "Error") = vbOK Then
      Text5.Text = ""
      Text6.Text = ""
    End If

Else

    i = 0
    j = 0

End If
    End If

End Sub
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้