สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

เครื่องเขียนแบบขนาดเล็ก

SMALL X-Y PLOTTER

นายกฤษณ์ เสมอพิทักษ์

นายสมบัติ บุญมาก

เลขหมู่.....
เลขทะเบียน.....42540
วัน, เดือน, ปี.....24 พ.ค. 2545

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

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมการวัดคุม

ภาควิชาเทคโนโลยีการวัดคุมทางอุตสาหกรรม คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2543

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

SMALL X-Y PLOTTER

KRIT SMERPITAK

SOMBUT BOONMAK

A THESS SUBMITTED IN PARTIAL FULFILLMENT

OF THE REQUIREMENT FOR THE DEGREE OF

BACHELOR OF ENGINEERING IN INSTRUMENTATION ENGINEERING

DEPARTMENT OF INDUSTRIAL INSTRUMENTATION TECHNOLOGY

FACULTY OF ENGINEERING

KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

2000

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาควิชาเทคโนโลยีการวัดคุมทางอุตสาหกรรม

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ใบรับรองปริญญานิพนธ์

หัวข้อปริญญานิพนธ์ เครื่องเขียนแบบขนาดเล็ก

SMALL X-Y PLOTTER

นักศึกษาผู้จัดทำ นายกฤษณ์ เสมอพิทักษ์ **รหัสประจำตัว** 41012080

นายสมบัติ บุญมาก **รหัสประจำตัว** 41012103

ปริญญา วิศวกรรมศาสตรบัณฑิต

สาขาวิชา วิศวกรรมการวัดคุม

ปีการศึกษา 2543

อาจารย์ผู้ควบคุมปริญญานิพนธ์	ลายมือชื่อ
รศ.วิทยา ทิพย์สุวรรณพร	

วัน/เดือน/ปี ที่สอบ วันพฤหัสบดีที่ 15 มีนาคม พ.ศ. 2544

สถานที่สอบ ณ. ห้องสอบปริญญานิพนธ์ ภาควิชาเทคโนโลยีการวัดคุมทางอุตสาหกรรม

ภาควิชารับรองแล้ว

(ผศ.ประสิทธ์ จุลเสรีวงศ์)

หัวหน้าภาควิชา ฯ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อปริญญานิพนธ์	เครื่องเขียนแบบขนาดเล็ก
	SMALL X-Y PLOTTER
นักศึกษาผู้จัดทำ	นายกฤษณ์ เสมอพิทักษ์
	นายสมบัติ บุญมาก
อาจารย์ที่ปรึกษา	รศ.วิทยา ทิพย์สุวรรณพร
ปีการศึกษา	2543

บทคัดย่อ

ปริญญานิพนธ์ฉบับนี้เสนอการออกแบบและสร้างต้นแบบเครื่องวาดภาพ ขนาด 75*75 มิลลิเมตร ใช้คอมพิวเตอร์ส่วนบุคคลประมวลข้อมูลตัวอักษรซึ่งประกอบด้วย ระยะ X ระยะ Y และสถานภาพของปลายปากกาที่สร้างจากโปรแกรม Auto CAD ในรูปไฟล์ข้อมูล PLT และส่งข้อมูลพิกัด การเคลื่อนที่ สถานภาพของปลายปากกา ผ่านระบบเชื่อมต่อมาตรฐาน RS232 ให้กับชุดควบคุมการเคลื่อนที่บนแกน X -Y และชุดควบคุมปากกา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Thesis Title	SMALL X-Y POLTTER
Authers	Mr. Krit Smerpitak
	Mr. Sombut Boonmak
Thesis advisor	Assoc. Prof . Vitaya Tipsuwanporn
Year	2000

ABSTRACT

This Project presents designing and prototype to the SMALL X-Y PLOTTER which has dimension 75 * 75 mm. It uses personal computer to compute data to letters with distance X distance Y and state of pen. It built by AutoCAD in file type PLT then sent data of coordinate and state of pen through RS-232 to motion controller board.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

ปริญญานิพนธ์ฉบับนี้สำเร็จลุล่วงได้เพราะได้รับความเมตาจาก รศ. วิทยา ทิพย์สุวรรณพร และ ผศ. ประภาษ อุคคกิมาพันธ์ ที่ได้ให้คำแนะนำ อีกทั้งยังเอื้อเฟื้ออุปกรณ์และเครื่องมือต่างๆ ในการทำปริญญานิพนธ์นี้

ขอขอบพระคุณอาจารย์ภาควิชาเทคโนโลยีการวัดคุมทางอุตสาหกรรมทุกท่านที่ให้คำแนะนำอันเป็นประโยชน์ต่อการทำปริญญานิพนธ์ฉบับนี้

คุณค่าและประโยชน์อันพึงมีจากปริญญานิพนธ์ฉบับนี้ ขอมอบแด่ผู้มีพระคุณทุกท่าน

คณะผู้จัดทำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

หน้า

บทคัดย่อภาษาไทย.....I

บทคัดย่อภาษาอังกฤษ.....II

กิตติกรรมประกาศ.....III

สารบัญ.....IV

สารบัญตาราง.....VII

สารบัญภาพ.....IX

บทที่ 1 บทนำ.....1

1.1 ความเป็นมาและเหตุจูงใจ.....1

1.2 วัตถุประสงค์ของปริญญานิพนธ์.....1

1.3 ขอบเขตของปริญญานิพนธ์.....1

1.4 ขั้นตอนการศึกษา.....1

บทที่ 2 ความรู้เบื้องต้นเกี่ยวกับเครื่อง X-Y Plotter.....2

2.1 กล่าวนำ.....2

บทที่ 3 หลักการและทฤษฎีที่เกี่ยวข้อง.....4

3.1 ไมโครคอนโทรลเลอร์ MCS-51.....4

3.1.1 คุณสมบัติทางเทคนิคของไมโครคอนโทรลเลอร์ตระกูล AT89xx 4

3.1.2 โครงสร้างและการทำงานของพอร์ต.....7

3.1.3 การใช้งานเป็นพอร์ตอินพุต.....8

3.1.4 การใช้งานพอร์ตเอาต์พุต.....8

3.1.5 พอร์ตอนุกรมของไมโครคอนโทรลเลอร์ MC5-51.....9

3.2 การสื่อสารข้อมูลแบบอะซิงโครนัส.....9

3.2.1 อัตราความเร็วในการรับและส่งข้อมูลของการรับส่งข้อมูลแบบอะซิงโครนัส หรือบอดเรต.....10

3.2.2 การเขียนหรือส่งข้อมูลออกจากพอร์ตอนุกรม.....10

3.2.3 การอ่านหรือรับข้อมูลจากพอร์ตอนุกรม.....11

3.2.4 มาตรฐานพอร์ตอนุกรมแบบ RS-232 11

3.2.5 คอนเน็กเตอร์สำหรับพอร์ต คอนRS-232 และการเชื่อมต่อ.....12

3.3 โปรแกรม AutoCAD R14.....14

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

หน้า

3.4 การเขียนโปรแกรมด้วย Visual Basic.....15

3.4.1 คอนโทรลภายในของ Visual Basic.....15

3.4.2 คุณสมบัติแสดงค่าของคอนโทรลที่สำคัญ.....16

3.4.3 การแบ่งกลุ่มของคอนโทรลภายใน.....17

3.4.4 คอนโทรลภายใน.....18

3.4.4.1 คอนโทรล Command Button.....18

3.4.4.2 คอนโทรล TextBox.....18

3.4.4.3 คอนโทรล Label.....19

3.4.4.4 คอนโทรล Timer.....20

3.4.5 คอนโทรลด้านฐานข้อมูล.....21

3.4.6 Microsoft JET Database Engine.....22

3.4.7 คอนโทรล Data.....23

3.4.8 คอนโทรล MsflexGrid.....24

3.4.9 คอนโทรลคอนโทรล MSComm (Communication).....25

3.5 สเต็ปเปอร์มอเตอร์.....27

3.5.1 การกระตุ้นและควบคุมการหมุนของสเต็ปเปอร์มอเตอร์.....28

บทที่ 4 การออกแบบและสร้างเครื่องเขียนแบบ.....31

4.1 การออกแบบและสร้างส่วนการเคลื่อนที่.....31

4.2 การออกแบบและสร้างส่วนของ MCS-51.....35

4.2.1 บอร์ด MCS-51.....35

4.2.2 โปรแกรม MCS-5.....16

4.3 โปรแกรมคอมพิวเตอร์.....37

4.3.1 โปรแกรม Visual Basic.....38

4.3.2 โปรแกรม AutoCAD R14.....39

4.3.3 วิธีการตั้งเครื่อง PLOT ในโปรแกรม AutoCAD R14.....40

4.3.4 วิธีการเริ่มต้นเขียนแบบใน โปรแกรม Auto CAD.....42

4.3.5 วิธีการตั้งค่า File.PLT ของโปรแกรม AutoCAD (หลังจากเขียนภาพ).....43

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

หน้า

4.4 การคำนวณหาค่าระยะเวลาการหน่วงของ Stepping Motor.....44
4.4.1 การหาค่ามุมที่เอียงที่แคบที่สุดที่สามารถทำได้ 45

บทที่ 5 ผลการทดลอง.....46
5.1 ภาพหน้าต่างของโปรแกรม Visual Basic.....46
5.2 ภาพหน้าต่างของโปรแกรม Visual Basic ใช้เปิด file plot.....47
5.3 การทดลอง.....48
5.3.1 การทดลองเขียนภาพโดยโปรแกรม Visual Basic ภาพที่ 1 48
5.3.2 การทดลองเขียนภาพโดยโปรแกรม Auto CAD R14 ภาพที่ 1 49
5.3.3 การทดลองเขียนภาพโดยโปรแกรม Auto CAD R14 ภาพที่ 2 50
5.3.4 การทดลองเขียนภาพโดยโปรแกรม Auto CAD R14 ภาพที่ 3 51

บทที่ 6 สรุปผลและข้อจำกัดของโครงงาน 52
6.1 สรุปผล 52
6.2 ข้อจำกัดของโครงงาน 52

บรรณานุกรม 53

ภาคผนวก 54

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

ตารางที่ หน้า

3.1 หน้าที่พิเศษของพอร์ต 1 ในไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลช 7

3.2 แสดงการจัดขาคอนเน็กเตอร์พอร์ตอนุกรมตามมาตรฐาน RS-232 12

3.3 แสดงชื่อคอนโทรลที่สำคัญ 16

3.4 รูปแบบการขับสเต็ปเปอร์มอเตอร์แบบฟูลสเต็ป 1 เฟส 29

3.5 รูปแบบการขับสเต็ปเปอร์มอเตอร์แบบฟูลสเต็ป 2 เฟส 29

3.6 รูปแบบการขับสเต็ปเปอร์มอเตอร์แบบฮาล์ฟสเต็ป 30

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญภาพ

ภาพที่ หน้า

2.1 ระบบโดยรวม.....3

3.1 โครงสร้างพื้นฐานของไมโครคอนโทรลเลอร์แบบแฟลชเบอร์ AT89Sxx 5

3.2 แสดงการวางตำแหน่งขาของไมโครคอนโทรลเลอร์ตระกูล AT89xxx 5

3.3 แสดงรูปแบบของข้อมูลอนุกรมแบบอะซิงโครนัส 9

3.4 แสดงหน้าต่างของโปรแกรม Auto CAD R14 15

3.5 แสดงคอนโทรล ControlButton ในขณะออกแบบ 18

3.6 แสดงคอนโทรล TextBox 19

3.7 แสดงคอนโทรล Label 19

3.8 แสดงคอนโทรล Timer 20

3.9 แสดงคอนโทรล Data 24

3.10 แสดงคอนโทรล MsflexGrid 24

3.11 แสดงคอนโทรล MSComm 27

3.12 โครงสร้างพื้นฐานของสเต็ปเปอร์มอเตอร์แบบยูนิโพล่าร์ แบบ 5 และ 6 สาย.....28

4.1 ระบบการทำงานของส่วนเคลื่อนที่ 31

4.2 ส่วนเคลื่อนที่ 1 แกน 32

4.3 ส่วนเคลื่อนที่ 2 แกน 33

4.4 เครื่องเขียนแบบขนาดเล็ก 34

4.5 ระบบการทำงานของ MCS-51 35

4.6 Flow chart โปรแกรมการทำงานของ MCS-51 36

4.7 ระบบการทำงานโดยรวมของโปรแกรม 37

4.8 Flow chart การทำงานของโปรแกรม Visual Basic 38

4.9 Flow chart การทำงานของ โปรแกรม AutoCAD 39

4.10 การติดตั้งเครื่อง PLOT ในโปรแกรม AutoCAD R14 40

4.11 การเลือกเครื่อง PLOT 41

4.12 กรอบสี่เหลี่ยมแสดงถึงพื้นที่ที่วาดภาพได้ 42

4.13 แสดง File → Print 43

4.12 แสดงการตั้งค่าใน File.PLT 44

5.1 แบบฟอร์มโปรแกรมการเขียนรูปจาก Visual Basic 46

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญภาพ (ต่อ)

ภาพที่ หน้า

5.2 แบบฟอร์มโปรแกรมเปิดภาพจาก file plot ของ Auto CAD R14 โดย Visual Basic 47
5.3 ภาพวาดจากโปรแกรม Visual Basic 48
5.4 ภาพที่ได้จากเครื่องเขียนเขียนแบบภาพที่ 1 48
5.5 ภาพวาดจากโปรแกรม Auto CAD R14 49
5.6 ภาพที่ได้จากเครื่องเขียนเขียนแบบภาพที่ 2 49
5.6 ภาพวาดจากโปรแกรม Auto CAD R14 เปิดจาก Visual Basic 50
5.7 ภาพที่ได้จากเครื่องเขียนเขียนแบบ ภาพที่ 3 50
5.8 ภาพวาดจากโปรแกรม Auto CAD R14 51
5.9 ภาพที่ได้จากเครื่องเขียนเขียนแบบ 51
1. แสดงวงจรโดยรวมของชุดควบคุมการเคลื่อนที่ของแกน X และ Y 55
2. แสดงรูปร่างของเครื่องเขียนแบบขนาดเล็กด้านข้าง 56
3. แสดงรูปร่างของเครื่องเขียนแบบขนาดเล็กด้านบน 56
4. แสดงด้านหน้าของกล่องชุดควบคุมการเคลื่อนที่ 57
5. แสดงด้านหลังของกล่องชุดควบคุมการเคลื่อนที่ 57

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1
บทนำ

1.1 ความเป็นมาและเหตุจูงใจ

การเขียนแบบในงานไฟฟ้า งานเครื่องกล งานสถาปนิก งานโยธา ฯลฯ ถ้าใช้มนุษย์เป็นผู้เขียนแบบก็จะต้องใช้ความสามารถ ความพยายามและระยะเวลา เพื่อให้ได้งานออกมา แต่ก็จะมีข้อผิดพลาดในเรื่องระยะห่าง ความถูกต้องของมุมที่วาด ฯลฯ เมื่อนำเครื่อง X-Y Plotter มาใช้ การเขียนแบบก็จะได้ขนาดที่ถูกต้อง ใช้เวลาสั้นในการเขียนแบบ

1.2 วัตถุประสงค์ของปริญญานิพนธ์

ปริญญานิพนธ์นี้จะเป็นการศึกษาเครื่องเขียนแบบและออกแบบสร้างเครื่องเขียนแบบเพื่อนำไปใช้งานเขียนแบบ และนำไปประยุกต์ใช้งานในทางอื่นๆ

1.3 ขอบเขตของปริญญานิพนธ์

ปริญญานิพนธ์เล่มนี้จะกล่าวถึงการสร้างเครื่องเขียนแบบขนาด 75*75 มิลลิเมตร และการเขียนโปรแกรมคอมพิวเตอร์ โดยรูปที่วาดจากโปรแกรม Auto CAD จะเป็นข้อมูลในรูปของไฟล์ POLT (plt) แล้วประมวลข้อมูลไฟล์ POLT โดยโปรแกรม Visual Basic แล้วส่งข้อมูลที่เป็นพิกัดและสถานะภาพของปลายปากกา ผ่าน RS-232 เข้า Board MSC-51 เพื่อควบคุมเครื่องเขียนแบบ

1.4 ขั้นตอนการศึกษา

การทำปริญญานิพนธ์ฉบับนี้เริ่มจากการศึกษาการเคลื่อนที่ในแนวแกน การศึกษาส่วนเคลื่อนที่ไมโครคอนโทรลเลอร์ MCS-51 โปรแกรม Auto CAD โปรแกรม Visual Basic และส่วนที่ประกอบอื่นๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2
ความรู้เบื้องต้นเกี่ยวกับเครื่อง X-Y Plotter

2.1 กล่าวนำ

เครื่องเขียนแบบขนานเล็กก็มีหลักการทำงานเช่นเดียวกับ X-Y Plotter ที่มีการเลื่อนแกน X และ Y โดยความสัมพันธ์ในทั้งสองแกนจะต้องมีความสัมพันธ์กันในด้านของระยะเคลื่อนที่กับความเร็วที่ใช้ของแต่ละแกน อย่างเช่น ต้องการไปที่ตำแหน่ง x=10 y=30 ก็จะต้องมีการเคลื่อนที่ของแกน X ช้ากว่าแกน Y แต่ระยะเวลาของทั้งสองแกนจะต้องหยุดพร้อมกันไม่เช่นนั้นมุมที่เกิดขึ้นก็จะไม่ได้ขนาดตามที่ต้องการ

ในการนำ Stepping Motor มาใช้งานก็จะมีข้อดีในด้านระยะการเคลื่อนที่ของมอเตอร์จะเป็นแบบ สเต็ป (Step) ที่สามารถนำมาใช้เพื่อให้ได้ระยะที่ถูกต้อง อย่างเช่น Stepping Motor มีความละเอียด 200 สเต็ปต่อรอบ ถ้าเราต้องการให้หมุนได้ 450 สเต็ป ก็จะได้จำนวนรอบเท่ากับ 2.25 รอบ และความเร็วของ Stepping Motor ก็จะใช้การหน่วงเวลาในแต่ละสเต็ป โดยระยะเวลาการหน่วงจะขึ้นอยู่กับระยะของการเคลื่อนที่ แกน X และ แกน Y ซึ่งจะต้องมีความสัมพันธ์กัน

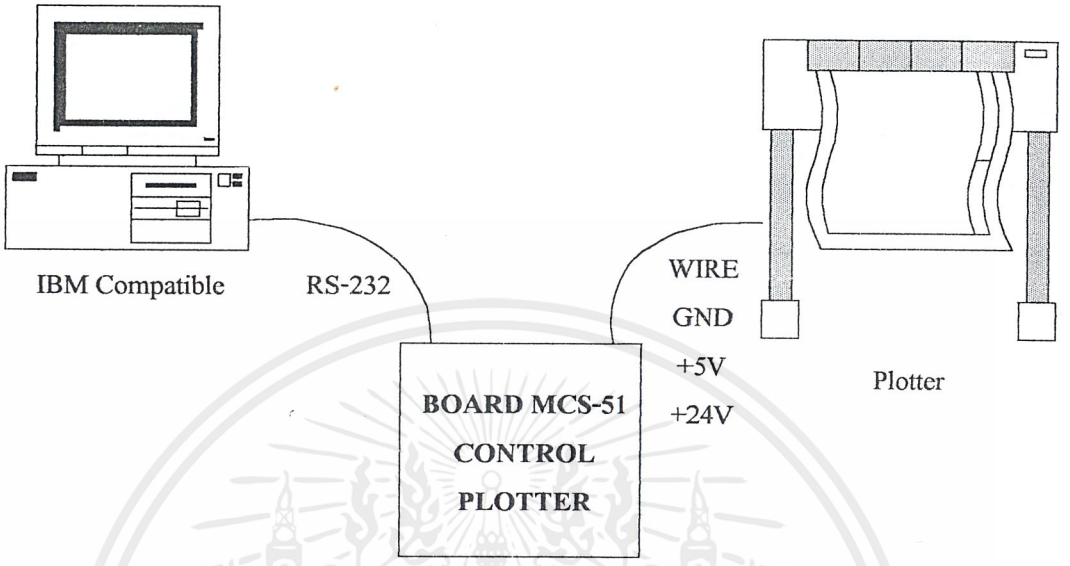
ส่วนที่ใช้ในการเคลื่อนที่ก็จะเป็นการออกแบบหลักการง่ายๆ คือการใช้ Stepping Motor หมุนเกลียวยาวตลอดแล้วให้ไปดันรางเลื่อนให้มีการเคลื่อนที่ไปได้ตามการหมุนของ Stepping Motor

ในด้านการเขียนโปรแกรมคอมพิวเตอร์ก็จะแยกเป็น 2 ส่วน โดยส่วนแรกจะเป็นโปรแกรม Visual BASIC ที่จะเป็นตัวสร้างทางเดินของการเคลื่อนที่ในคอมพิวเตอร์เพื่อส่งให้กับส่วนการเคลื่อนที่จริง และอีกส่วนจะเป็นการสร้างทางเดินจากโปรแกรม AutoCAD ที่มีรูปแบบเป็นข้อมูล File Plot แล้วนำมาจัดเรียงด้วยโปรแกรม Visual BASIC ก่อนที่จะส่งข้อมูลลงเครื่องเขียนแบบขนาดเล็ก

เมื่อได้ภาพที่ต้องการจะวาดแล้วก็จะถูกส่งให้กับ Board MCS-51 ผ่านทางการสื่อสารอนุกรม เพื่อให้ Board MCS-51 ควบคุมการทำงานของ Stepping Motor ของแกน X และแกน Y ในการควบคุมตำแหน่งความเร็วและการควบคุมชุดยกปากกาให้ลากหรือไม่ลากให้ได้ตามที่คอมพิวเตอร์ได้คำนวณไว้

จากที่ได้กล่าวมาเบื้องต้นจะได้ระบบโดยรวม ดังภาพที่ 1.1 ซึ่งจะประกอบด้วยส่วนของคอมพิวเตอร์ ส่วนไมโครคอนโทรลเลอร์ และส่วนการเคลื่อนที่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 2.1 ระบบโดยรวม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3
หลักการและทฤษฎีที่เกี่ยวข้อง

3.1 ไมโครคอนโทรลเลอร์ MCS-51

โครงสร้างและสถาปัตยกรรมของไมโครคอนโทรลเลอร์ตระกูล MCS-51 แบบแฟลช ของ Atmel ที่มีเบอร์ขึ้นต้นด้วย AT89 ซึ่งมีส่วนที่น่าสนใจดังนี้

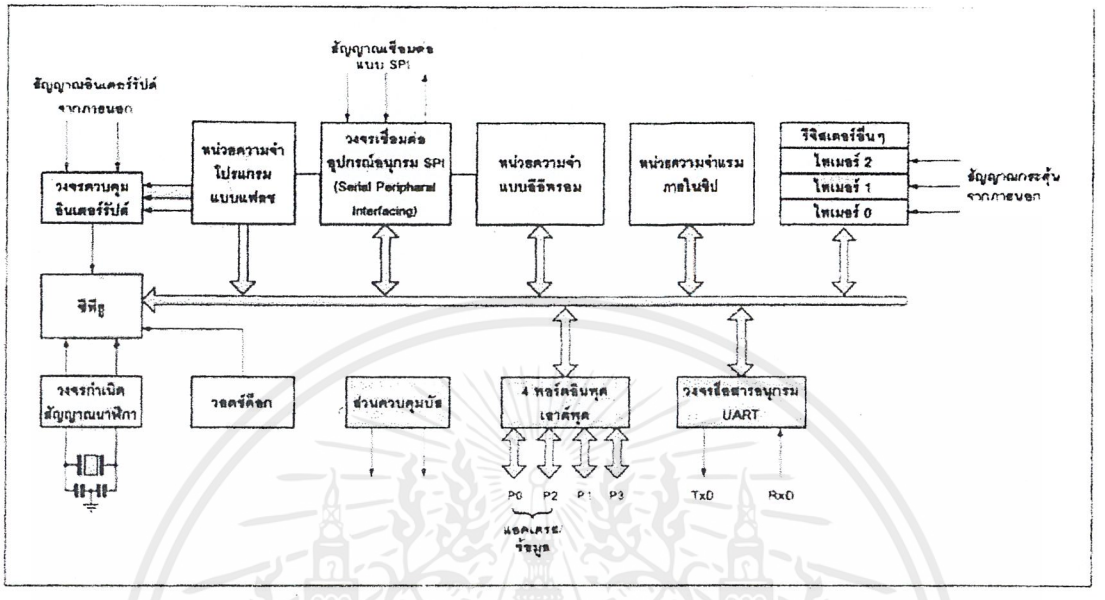
1. หน่วยความจำภายในตัวไมโครคอนโทรลเลอร์นี้สามารถที่จะลบและเขียนใหม่ได้เป็นพันครั้งไม่จำเป็นต้องใช้หน่วยความจำภายนอกจึงสามารถใช้พอร์ตอินพุตเอาท์พุตได้เต็มประสิทธิภาพ
2. ต้นทุนและเวลาในการพัฒนาโปรแกรมลดลงมากเนื่องจากไม่จำเป็นต้องใช้เครื่องมือพัฒนาจำพวกอีพรอมอีมูเลเตอร์และเครื่องโปรแกรมอีพรอม
3. ไมโครคอนโทรลเลอร์ตระกูลนี้ถูกผลิตออกมาให้ใช้หลายเบอร์มีลักษณะพิเศษทำให้มีทางเลือกในการใช้งานแตกต่างกันไปตามความสามารถของแต่ละเบอร์
4. เนื่องจากใช้หน่วยความจำภายในช่วยให้สามารถป้องกันการคัดลอกข้อมูลของหน่วยความจำโปรแกรมได้เป็นอย่างดี
5. ในบางเบอร์สามารถแก้ไขโปรแกรมภายในวงจรได้เลยเพียงต่อส่วนในการแข้ไขโปรแกรมแบบ SPI (Serial Peripheral Interface) ทำให้สะดวกต่อการแก้ไขและปรับปรุงโปรแกรม
6. ชุดคำสั่งและสถาปัตยกรรมพื้นฐานก็เหมือนกันกับไมโครคอนโทรลเลอร์ MCS-51 ตระกูลอื่นๆ

3.1.1 คุณสมบัติทางเทคนิคของไมโครคอนโทรลเลอร์ตระกูล AT89xx

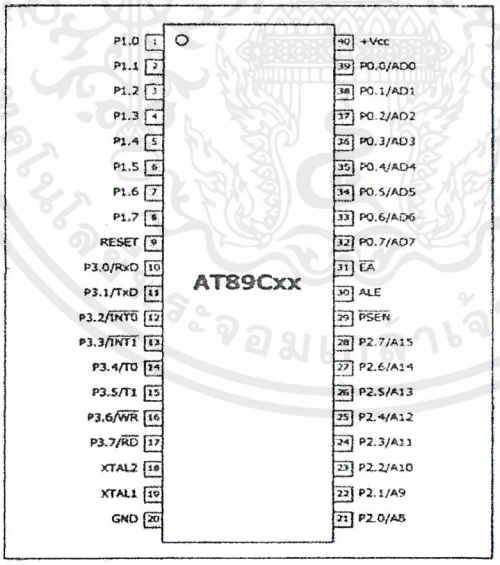
- เป็นไมโครคอนโทรลเลอร์ที่ใช้ซีพียูขนาด 8 บิต
- ภายในมีหน่วยความจำโปรแกรมเป็นแบบแฟลชสามารถลบและเขียนได้พันครั้ง
- หน่วยความจำข้อมูลพื้นฐานเป็นหน่วยความจำแบบแรมในบางเบอร์จะมีหน่วยความจำแบบ EEPROM เพิ่มเติม
- ขาพอร์ตเป็นแบบสองทิศทางสามารถเป็นได้ทั้งอินพุตและเอาท์พุต
- มีวงจรสื่อสารอนุกรมแบบฟูลดูเพล็กซ์
- ไทเมอร์/เคาน์เตอร์ขนาด 16 บิต อย่างน้อย 2 ตัว
- สามารถรองรับแหล่งกำเนิดอินเตอร์รัปต์ได้ 6 ประเภท
- สามารถขยายหน่วยความจำภายนอกเพิ่มเติมได้สูงสุด 64 กิโลไบต์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- มีวงจรกำเนิดสัญญาณนาฬิกาภายในชิป
- มีวงจรสื่อสารอนุกรมแบบ SPI สำหรับในตระกูล 89Sxx
- มีวอตช์ด็อกไทเมอร์ในตัว สำหรับในตระกูล 89Sxx



ภาพที่ 3.1 โครงสร้างพื้นฐานของไมโครคอนโทรลเลอร์แบบแฟลชเบอร์ AT89Sxx



ภาพที่ 3.2 แสดงการวางตำแหน่งขาของไมโครคอนโทรลเลอร์ตระกูล AT89xxx

ขา Vcc ใช้สำหรับต่อไฟเลี้ยง +5V

ขา GND เป็นขากราวด์ สำหรับต่อกราวด์ของระบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขาพอร์ต 0 (P0.0 – P0.7) มี 8 ขาแต่ละขาสามารถกำหนดให้เป็นได้ทั้งอินพุตและเอาท์พุต สำหรับใช้งานทั่วไปหากต้องการให้ขาหนึ่งขาใดในพอร์ต 0 เป็นอินพุตสามารถทำได้โดยการเขียนข้อมูล "1" ไปยังพอร์ตที่ต้องการติดต่อด้วย

ขาพอร์ต 1 (P1.0 – P1.7) เป็นขาใช้งานทั่วไปมี 8 ขาแต่ละขาสามารถกำหนดให้เป็นได้ทั้งอินพุตและเอาท์พุต

ขาพอร์ต 2 (P2.0 – P2.7) หน้าที่เหมือนกับพอร์ต 1 และพอร์ต 2

ขาพอร์ต 3 (P3.0 – P3.7) ทำหน้าที่เหมือนกับพอร์ตอื่นๆ แต่ยังเป็นขาที่มีหน้าที่ใช้งานพิเศษมีรายละเอียดดังนี้

- P3.0 ใช้เป็นขาอินพุต สำหรับรับข้อมูลจากการสื่อสารแบบอนุกรมหรือขา RxD
- P3.1 ใช้เป็นขาอินพุต สำหรับส่งข้อมูลจากการสื่อสารแบบอนุกรมหรือขา TxD
- P3.2 ใช้เป็นขาอินพุต สำหรับสัญญาณอินเตอร์รัปต์ภายนอกช่องที่ 0 หรือ ขา $\overline{INT0}$
- P3.3 ใช้เป็นขาอินพุต สำหรับสัญญาณอินเตอร์รัปต์ภายนอกช่องที่ 1 หรือ ขา $\overline{INT1}$
- P3.4 ใช้เป็นขาอินพุต สำหรับรับสัญญาณไทเมอร์จากภายนอกช่องที่ 0 หรือขา T0
- P3.5 ใช้เป็นขาอินพุต สำหรับรับสัญญาณไทเมอร์จากภายนอกช่องที่ 1 หรือขา T1
- ใช้เป็นขาสัญญาณ $\overline{WR}$ ในกรณีที่ใช้เชื่อมกับหน่วยความจำภายนอก
- ใช้เป็นขาสัญญาณ $\overline{RD}$ ในกรณีที่ใช้เชื่อมกับหน่วยความจำภายนอก

ขารีเซต ใช้ในการรีเซตการทำงานของไมโครคอนโทรลเลอร์ โดยในการป้อนสัญญาณเพื่อรีเซตสถานะที่ขานี้ต้องอยู่ในระดับรีเซตอย่างน้อย 2 แมชีนไซเกิล โดยที่วงจรกำเนิดสัญญาณนาฬิกายังคงทำงานต่อเนื่องอย่างเป็นปกติ

ขา $ALE/\overline{PROG}$ **(Address Latch Enable/Program pulse input)** เป็นขาที่ใช้ในการควบคุมการแลตซ์ของขาพอร์ต 0 เมื่อมีการใช้งานหน่วยความจำภายนอก นอกจากนั้นขานี้ยังใช้เป็นขาสำหรับรับพัลส์ของการทำโปรแกรมสำหรับใช้โปรแกรมข้อมูลที่จะใส่ลงในไมโครคอนโทรลเลอร์ในรุ่นที่มีหน่วยความจำแบบ EEPROM

ขา $\overline{PSEN}$ **(Program Store Enable)** ขานี้ใช้ในการส่งสัญญาณเพื่อร้องขอติดต่อกับหน่วยความจำโปรแกรมภายนอกตัวไมโครคอนโทรลเลอร์จะต้องส่งสัญญาณมาที่ขานี้ 2 ครั้งในแต่ละแมทชีนไซเกิล หากติดต่อกับหน่วยความจำภายในขานี้จะไม่มีการส่งสัญญาณใดๆ ออกมา

ขา $\overline{EA}/Vpp$ **(External Access Enable/Programming voltage input)** ใช้สำหรับเลือกการติดต่อหน่วยความจำจากภายในหรือภายนอกตัวไมโครคอนโทรลเลอร์ถ้าหากขานี้เป็น "0" เป็นการเลือกใช้หน่วยความจำโปรแกรมภายนอกแต่ถ้าหากขานี้เป็น "1" เป็นการเลือกให้ติดต่อกับหน่วยความจำภายในตัวไมโครคอนโทรลเลอร์นอกจากนี้ขานี้ยังสามารถที่จะใช้เป็นขาอินพุตสำหรับรับแรงดันไฟสูงสำหรับการโปรแกรมหน่วยความจำภายในตัวของไมโครคอนโทรลเลอร์สำหรับไมโครคอนโทรลเลอร์ MCS-51 แบบ แฟลชต้องการแรงดันสำหรับการโปรแกรม +12 V

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขา XTAL1 และ XTAL2 ขานี้ใช้สำหรับต่อคริสตอลเพื่อสร้างสัญญาณนาฬิกาในการกำหนดจังหวะการทำงานของไมโครโปรคอนโทรลเลอร์

3.1.2 โครงสร้างและการทำงานของพอร์ต

ไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลชมีพอร์ตใช้งานทั้งสิ้น 4 พอร์ตคือ พอร์ต 0 ถึงพอร์ต 3 แต่ละพอร์ตมีขนาด 8 บิต เป็นพอร์ตแบบ 2 ทิศทาง กล่าวคือสามารถเป็นได้ทั้งอินพุตและเอาต์พุตทุกพอร์ตของไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลชมีวงจรแลตช์และวงจรขับตลอดจนบัฟเฟอร์อินพุตดังแสดงให้เห็นในสถาปัตยกรรม

ตารางที่ 3.1 หน้าที่พิเศษของพอร์ต 1 ในไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลช

ขา	เบอร์ของไมโครคอนโทรลเลอร์	หน้าที่พิเศษ
P1.1	AT89C52/AT89Sxx	ขาT2EX เป็นขาอินพุตทริกเกอร์สำหรับการแคปเจอร์/รีโหลดและควบคุมทิศทางของสัญญาณ
P1.4	AT89Sxx	ขาT2 เป็นขาอินพุตนับค่าของไทเมอร์/เคาน์เตอร์ 2 และเป็นขาเอาต์พุตของการกำเนิดสัญญาณนาฬิกาโดยไทเมอร์ 2
P1.5	AT89Sxx	ขา $\overline{SS}$ (Slave Select) เป็นขาเลือกการติดต่อในกรณีที่ไมโครคอนโทรลเลอร์เป็นอุปกรณ์สเลฟ ในระบบการติดต่อแบบ SPI
P1.6	AT89Sxx	ขา MOSI (Master data output , Slave data output) ใช้ในการติดต่อกับพอร์ต SPI
P1.7	AT89Sxx	ขา MISO (Master data input , Slave data input) ใช้ในการติดต่อกับพอร์ต SPI
P1.0	AT89C52/AT89Sxx	ขา SCK (Master clock output) เป็นขาสัญญาณนาฬิกาของการติดต่อพอร์ต SPI

ที่พอร์ต 0 และพอร์ต 2 จะใช้งานเป็นพอร์ตอินพุตและเอาต์พุตสำหรับงานทั่วไป และใช้ในการติดต่อกับหน่วยความจำภายนอก สำหรับพอร์ต 3 ทั้งพอร์ตและพอร์ต 1 บางขานอก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากจะใช้เป็นขาพอร์ตอินพุตและเอาต์พุตตามปกติแล้วยังสามารถใช้งานในหน้าที่พิเศษได้อีกขึ้นอยู่กับว่าเป็นไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลชเบอร์ใด ดังสรุปในตารางที่ 3.1

3.1.3 การใช้งานเป็นพอร์ตอินพุต

เนื่องจากพอร์ตทั้งหมดของไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลชสามารถเป็นได้ทั้งอินพุตและเอาต์พุตดังนั้นจึงมีความจำเป็นอย่างยิ่งที่ต้องทำความเข้าใจถึงการกำหนดลักษณะการทำงานให้แก่พอร์ตการทำงานของไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลช

ในการกำหนดให้เป็นพอร์ตอินพุตต้องเริ่มด้วยการเขียนข้อมูล "1" มาที่แต่ล่ะบิตของพอร์ตที่ต้องการใช้งานเป็นอินพุตเพื่อหยุดการทำงานของเฟตที่ใช้ในการขับสัญญาณเอาต์พุตของบิตนั้นๆ ทำให้ขาสัญญาณของพอร์ตเชื่อมต่อเข้ากับวงจรพูลอัฟโดยตรงส่งผลให้ขาพอร์ตนั้นมีลอจิกเป็น "1" ซึ่งสามารถรับลอจิก "0" จากอุปกรณ์ภายนอกได้ง่าย สัญญาณข้อมูลจากอุปกรณ์ภายนอกถูกส่งเข้ามาแล้วเก็บไว้ภายในวงจรบัฟเฟอร์ภายในพอร์ตแล้วรอให้ CPU มาอ่านค่าเข้าไป เมื่อเป็นเช่นนี้อุปกรณ์ภายนอกที่เชื่อมต่อกับพอร์ตอินพุตของไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลชควรกำหนดให้ทำงานในสภาวะลอจิก "0" จะดีและสะดวกที่สุด (ซึ่งในปัจจุบันอุปกรณ์อินพุตที่เชื่อมต่อกับไมโครคอนโทรลเลอร์แทบทั้งหมดทำงานที่ลอจิก "0" แล้ว)

3.1.4 การใช้งานพอร์ตเอาต์พุต

โดยปกติแล้วขาพอร์ตจะกำหนดให้มีลักษณะเป็นเอาต์พุตอยู่แล้ว ดังนั้นจึงสามารถส่งข้อมูลได้อย่างง่ายดายและตรงไปตรงมากล่าวคือ เมื่อต้องการส่งข้อมูล "0" ออกไปทางเอาต์พุตก็ให้เขียนข้อมูล "0" ไปยังวงจรแลตช์ซึ่งจะส่งผลต่อไปขับเฟตทำให้เฟตทำงานที่ขาพอร์ตที่กำหนดให้ทำงานก็จะเกิดลอจิก "0" ขึ้นในทางตรงข้ามหากส่งข้อมูล "1" ออกไปก็ให้เขียนข้อมูล "1" ในวงจรแลตช์ วงจรขับก็จะหยุดทำงานทำให้ที่ขาพอร์ตเชื่อมต่อกับวงจรพูลอัฟภายในเกิดเป็นลอจิก "1" ที่ขาพอร์ตนั้น ซึ่งจะคล้ายกับการกำหนดให้เป็นขาอินพุตมากเพียงแต่ต่างกันที่กระบวนการเคลื่อนย้ายข้อมูลโดยถ้าเป็นอินพุตจะมีสัญญาณมาอ่านข้อมูลที่บัฟเฟอร์ แต่ถ้าเป็นเอาต์พุตไม่มีการอ่านข้อมูลที่บัฟเฟอร์แต่อย่างใด เว้นแต่ในกรณีที่ต้องการตรวจสอบข้อมูลที่ส่งออกมาทางเอาต์พุต

เมื่อใช้งานพอร์ตของไมโครคอนโทรลเลอร์ของ MCS-51 แบบแฟลชเป็นแบบเอาต์พุตแต่ละขา (หรือแต่ละบิต) ของแต่ละพอร์ตมีความสามารถในการจ่ายกระแสหรือเรียกว่ากระแสซอร์ส(source current) ได้สูงสุด 10 mA และทุกขารวมกันในแต่ล่ะพอร์ต(ทั้ง 8 บิต) สูงสุด 26 mA สำหรับพอร์ต 0 และ 15 mA สำหรับพอร์ต 1-3 ในกรณีที่ใช้งานทุกพอร์ตเอาต์พุตสามารถจ่ายกระแสได้รวมกันสูงสุด 71 mA ดังนั้นในการใช้งานเป็นพอร์ตเอาต์พุตเพื่อไม่ให้เกิดปัญหากับความสามารถในการจ่ายกระแสจึงควรต่อวงจรบัฟเฟอร์ทางเอาต์พุตเพื่อช่วยขับกระแสอีกทางหนึ่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1.5 พอร์ตอนุกรมของไมโครคอนโทรลเลอร์ MC5-51

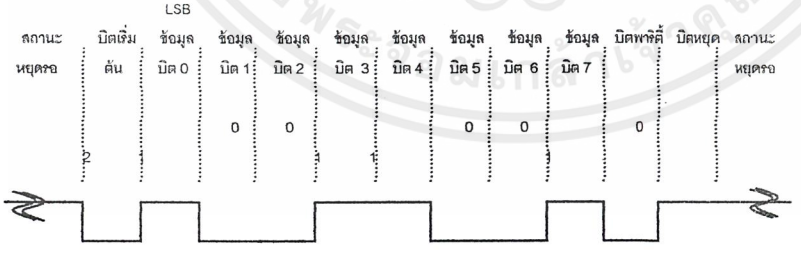
ไมโครคอนโทรลเลอร์ตระกูล MCS-51 มีวงจรสื่อสารอนุกรมแบบฟูลดูเพล็กซ์ 1 ชุด (วงจรสื่อสารแบบฟูลดูเพล็กซ์ หมายถึง วงจรสื่อสารที่สามารถทำการรับและส่งข้อมูลในลักษณะ 2 ทิศทางได้ในเวลาเดียวกัน) โดยใช้ขาสัญญาณของพอร์ต 3 คือ ขา P 3.0 เป็นขาข้อมูลเข้าหรือ RXD และขา P3.1 เป็นขาส่งข้อมูลออกหรือ TXD โดยวงจรสื่อสารข้อมูลแบบอนุกรมของไมโครคอนโทรลเลอร์ตระกูล MCS-51 แบบแฟลชเป็นแบบอะซิงโครนัสปกติแล้วพอร์ตอนุกรมของไมโครคอนโทรลเลอร์ MCS-51 จะใช้ในการติดต่อสื่อสารกับพอร์ตอนุกรมของคอมพิวเตอร์ โดยใช้มาตรฐาน RS-232 แต่ในปัจจุบันสามารถติดต่อกันในมาตรฐาน RS-422 หรือ RS-485 ได้แล้วโดยใช้ไอซีพิเศษที่ทำหน้าที่ในการแปลงสัญญาณการสื่อสารดังกล่าว

3.2 การสื่อสารข้อมูลแบบอะซิงโครนัส

การสื่อข้อมูลแบบอซิงโครนัสคือการรับและส่งข้อมูลโดยไม่จำเป็นต้องมีสัญญาณนาฬิการ่วมด้วยแต่จะใช้การกำหนดค่าอัตราเร็วในการรับและส่งข้อมูลให้มีค่าเท่ากัน ซึ่งเรียกอัตราเร็วนี้ว่า อัตราบอดเรต หรือ บอดเรต(Baud rate) มีหน่วยเป็น บิตต่อวินาที (bps)

รูปแบบของข้อมูลที่ใช้ในการรับส่งแบบอะซิงโครนัสประกอบด้วย 4 ส่วนด้วยกันคือ

1. บิตเริ่มต้น (start bit) มีขนาด 1 บิต
2. บิตข้อมูลแบบอนุกรมมีขนาด 8 บิต
3. บิตตรวจสอบพาริตี้(parity bit) มีขนาดหนึ่งบิตหรือไม่มี
4. บิตปิดท้ายหรือบิตหยุด(stop bit) มีขนาด 1 บิต



ภาพที่ 3.3 แสดงรูปแบบของข้อมูลอนุกรมแบบอะซิงโครนัส

เมื่อไม่มีการส่งข้อมูลขา DATA จะมีสถานะลอจิก “1” เรียกสถานะนี้ว่า สถานะหยุดรอ (wait state) การเริ่มต้นการส่งข้อมูลจะเริ่มจากการให้ขา DATA มีลอจิก “0” ด้วยช่วงระยะเวลา 1 บิต เรียกบิตนี้ว่า บิตเริ่มต้น(start bit) จากนั้นข้อมูลจะถูกส่งออกไป โดยเริ่มจากบิตนัยสำคัญต่ำสุดหรือ LSB ก่อนซึ่งข้อมูลที่ต้องการส่งมีจำนวน 8 บิต จากนั้นตามด้วย บิตพาริตี้ (paraty bit)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ซึ่งใช้ตรวจสอบความผิดพลาดที่เกิดอาจจะเกิดขึ้นจากการส่งข้อมูบิตสุดท้ายที่จะส่งคือ บิตปิดท้าย หรือบิตหยุด (stop bit) โดยจะเป็นการทำให้ขา DATA มีลอจิก "1" อีกครั้งด้วยระยะเวลาอย่างน้อย 1 บิต 1.5 บิต หรือ 2 บิต เพื่อเป็นการแสดงว่าสิ้นสุดข้อมูลแล้ว

3.2.1 อัตราความเร็วในการรับและส่งข้อมูลของการรับส่งข้อมูลแบบอะซิงโครนัส หรือบอดเรต

สำหรับพอร์ตอนุกรม RS-232 มีด้วยกันหลายค่าได้แก่ 110, 150, 300, 600, 1200, 2400, 4800, 9600 และ 19200 บิตต่อวินาที โดยมีค่าเพิ่มมากขึ้นตามเทคโนโลยีของคอมพิวเตอร์ เนื่องจากอัตราคือค่าของจำนวนบิตที่ส่งได้ภายใน 1 วินาที สมมติว่าข้อมูลอนุกรมมีขนาด 8 บิต ไม่มีการตรวจสอบพารีตี้ มีบิตเริ่มต้น 1 บิตและบิตปิดท้าย 1 บิต ความยาวของข้อมูล 1 ไบต์จะมีขนาดเท่ากับ 10 บิต ถ้าใช้บอดเรตในการส่งข้อมูลเท่ากับ 9600 บิตต่อวินาทีก็จะสามารถรับส่งข้อมูลได้ด้วยความเร็ว 960 ไบต์ต่อวินาที การตรวจสอบพาริตี้สามารถกำหนดให้เป็นแบบคี่(odd) แบบคู่(even) หรือไม่มีการตรวจสอบพาริตี้ก็ได้ พาริตี้คี่หรือคู่ แสดงถึงจำนวนลอจิก "1" ทั้งหมดภายในข้อมูลที่ส่งไป 1 ไบต์รวมบิตพาริตี้ว่ามีจำนวนเลขคี่หรือเลขคู่ ยกตัวอย่างข้อมูลที่จะทำการส่งมีขนาด 8 บิต มีค่าเท่ากับ 99h หรือ 10011001bจะเห็นว่าข้อมูลในไบต์นี้มีจำนวนลอจิก "1" จำนวน 4 ตัวซึ่งเป็นเลขคู่ ดังนั้นถ้ากำหนดค่าพาริตี้เป็นคู่ ค่าของบิตพาริตี้จะต้องมีลอจิกเป็น "0" แต่ถ้ากำหนดพาริตี้เป็นคี่ ค่าของพาริตี้จะต้องเป็น "1" เพื่อให้ข้อมูล 1 ไบต์ รวมทั้งบิตพาริตี้เป็นคี่

บิตพาริตี้ถูกสร้างขึ้นจากภาคส่งข้อมูลของ UART (Universal Asynchronous Receiver Transmitter):เป็นอุปกรณ์ที่ใช้ในการรับส่งข้อมูลอนุกรม ซึ่งทางภาครับจะต้องกำหนดคุณสมบัติการตรวจสอบพาริตี้ที่ตรงกันเอาไว้ว่าจะตรวจสอบพาริตี้คู่หรือพาริตี้คี่ จากนั้นภาครับของ UART จะทำการตรวจสอบค่าพาริตี้ที่เกิดขึ้นว่าเป็นคู่หรือคี่ โดยการนับจำนวนลอจิก 1 ทั้งหมดรวมทั้งบิตพาริตี้ด้วย ถ้ากำหนดพาริตี้ไว้เป็นคู่แต่อ่านค่าตัวเลขในการนับค่าออกมาได้เป็นคี่ทางภาครับจะแสดงข้อผิดพลาดออกมาให้ผู้ใช้ทราบ กระบวนการดังกล่าวเป็นวิธีการตรวจสอบความผิดพลาดที่เกิดขึ้นในการรับส่งข้อมูลที่ง่ายที่สุด แต่มันสามารถตรวจสอบได้เมื่อมีบิตข้อมูลที่ทำการรับส่งข้อมูลผิดพลาดเพียงบิตเดียวเท่านั้น ถ้าข้อมูลที่ส่งมีบิตผิดพลาดมากกว่าหนึ่งบิตการตรวจสอบด้วยวิธีนี้จะไม่ได้ผลสำหรับการตั้งพาริตี้บิตเป็น NONE ทั้งทางภาครับและส่งไม่มีการตรวจสอบพาริตี้

3.2.2 การเขียนหรือส่งข้อมูลออกจากพอร์ตอนุกรม

ข้อมูลที่ต้องการส่งออกทุกค่าต้องนำไปเก็บไว้ที่รีจิสเตอร์บัฟเฟอร์ของพอร์ตอนุกรม ซึ่งก็คือ รีจิสเตอร์ SBUF ดังตัวอย่าง

```
MOV     SBUF, # 'A'
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากคำสั่งข้างต้นเป็นการส่งข้อมูลของตัวอักษร A ไปยังพอร์ตอนุกรมของไมโครคอนโทรลเลอร์อย่างไรก็ตามก่อนทำการส่งข้อมูลทุกครั้ง ต้องแน่ใจว่าบิต TI มีค่าเป็น 0 และเมื่อทำการส่งข้อมูลแล้ว ก็จะเกิดการเซ็ท TI เพื่อแจ้งให้ทราบ ดังตัวอย่างโปรแกรมต่อไปนี้

```
CLR     TI          ; เคลียร์บิต TI เพื่อเตรียมการส่งข้อมูลออก
MOV     SBUF,# 'A'  ; ส่งข้อมูลตัวอักษร A ไปยังพอร์ตอนุกรม
JNB     TI ,$       ; รอการเซตบิต TI เพื่อแจ้งการส่งข้อมูลที่
                      เสร็จสมบูรณ์
```

3.2.3 การอ่านหรือรับข้อมูลจากพอร์ตอนุกรม

การรับข้อมูลจากพอร์ตอนุกรมสามารถกระทำได้ง่ายมาก เพียงทำการตรวจว่าบิต RI เกิดการเซตขึ้นหรือไม่ ถ้าพบว่ามีการเซตเกิดขึ้นแล้ว ให้ทำการอ่านค่าจากรีจีสเตอร์ SBUF โดยต้องทำการโอนย้ายข้อมูลผ่านทาง แอกคิวมูเลเตอร์(A) ดังตัวอย่าง

```
CLR     RI          ; เคลียร์บิต RI เพื่อการรับข้อมูล
JNB     RI,$        ; รอคอยการเซตบิต RI อันเป็นการแจ้ง ให้ทราบ
                      ว่า การรับข้อมูลเสร็จสมบูรณ์
MOV     A,SBUF ; อ่านค่าจากรีจีสเตอร์โดยทำการ โอนย้ายมาที่ A
CLR     RI          ; หลังจากทำการรับข้อมูล ต้อง clear RI เสมอ
```

3.2.4 มาตรฐานพอร์ตอนุกรมแบบ RS-232

มาตรฐานการเชื่อมต่อแบบอนุกรมแบบ RS-232 เป็นมาตรฐานอุตสาหกรรมที่ออกแบบมาเพื่อใช้ในการส่งผ่านข้อมูลอนุกรมแบบอะซิงโครนัส 2 ทิศทาง โดยมาตรฐาน RS-232 ในอดีตนั้นถูกออกแบบมาเพื่อการส่งผ่านข้อมูลจากคอมพิวเตอร์ไปยังโมเด็มเพียงอย่างเดียว เพื่อที่จะนำข้อมูลจากโมเด็มนี้สื่อสารผ่านสายโทรศัพท์ไปยังคอมพิวเตอร์อีกชุดซึ่งอยู่ห่างไกลกัน โดยคณะกรรมการที่เรียกว่าสมาคมอุตสาหกรรมอิเล็กทรอนิกส์ (Electronic Industries Association : EIAX) ได้ว่ามาตรฐานที่มีชื่อเรียกกันว่า EIA RS-232 มาตรฐานนี้ในช่วงแรกจะใช้คอนเน็กเตอร์เป็นแบบ DB-25 โดยกำหนดความยาวสูงสุดของสายสัญญาณไว้ที่ 50 ฟุตมีระดับสัญญาณตั้งแต่ –3 ถึง –12 V แสดงว่ามีข้อมูล (Mark) และ +3 - +12 V แสดงว่าเป็นช่องว่าง (Space)

มาตรฐาน RS-232 กำหนดรูปแบบอุปกรณ์เชื่อมต่อ (Data Terminal Equipment : DTE) กับวงจรข้อมูลปลายทาง (Data Circuit Terminating : DCE) ไว้ว่าอุปกรณ์ DTE จะต้องเป็นอุปกรณ์ที่มีการประมวลผลในตัว เช่น ไมโครคอนโทรลเลอร์หรือไมโครคอมพิวเตอร์ ซึ่งมีความสามารถในการสร้างบิตข้อมูลแบบอนุกรมได้ ส่วนอุปกรณ์ DCE จะทำหน้าที่เป็นตัวรับข้อมูลที่ส่งมาจาก DTE เท่านั้น โดยการรับส่งข้อมูลระหว่างอุปกรณ์ทั้งสองจะกระทำผ่านมาตรฐาน RS-232

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ข้อแตกต่างของอุปกรณ์ DTE และอุปกรณ์ DCE อย่างหนึ่งที่เห็นได้ชัดคือ คอนเน็กเตอร์ ของ DTE จะเป็นตัวผู้ ส่วนคอนเน็กเตอร์ของ DCE จะเป็นตัวเมีย ซึ่งพอร์ตอนุกรมของคอมพิวเตอร์ที่ใช้กันอยู่ทั่วไปจะเป็นแบบ DTE ส่วนคอนเน็กเตอร์ที่อยู่ที่โมเด็มจะเป็นแบบ DCE สำหรับการใช้งานบนคอมพิวเตอร์ พอร์ตอนุกรม RS-232 มักถูกใช้เชื่อมต่อกับโมเด็มหรือเมาส์ โดยสามารถรับส่งข้อมูลได้ที่ความยาวของสายสัญญาณสูงสุด ถึง 20 เมตร

3.2.5 คอนเน็กเตอร์สำหรับพอร์ต คอนRS-232 และการเชื่อมต่อ

มาตราฐานการเชื่อมต่อแบ RS-232 จะใช้คอนเน็กเตอร์แบบ DB-25 ตัวผู้หรือ DB-9 ตัวผู้ ซึ่งคอนเน็กเตอร์แบบ DB-25 จะมีขาต่อใช้งานเพียง 9 เส้น เช่นเดียวกับคอนเน็กเตอร์แบบDB-9 เนื่องจากขาอื่น ๆ ที่ใช้งานอดีตไม่ได้ใช้ ปัจจุบันมีการใช้งานไม่มากนัก จึงถูกยกเลิกไป

ตารางที่ 3.2 แสดงการจัดขาคอนเน็กเตอร์พอร์ตอนุกรมตามมาตรฐาน RS-232

คอนเน็กเตอร์DB-9	คอนเน็กเตอร์DB-25	ชื่อของสายสัญญาณ	ชนิดของสายสัญญาณ
1	8	Data carrierDetect: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 SetReady: DSR	อินพุต
7	4	Request To Send:RTS	เอาต์พุต
8	5	Clear To Send: CTS	อินพุต
9	22	Ring Indicator:RI	อินพุต

สำหรับการเชื่อมต่อคอมพิวเตอร์กับอุปกรณ์ภายนอก เป็นการเชื่อมต่อแบบ Null modem หรือการเชื่อมต่อโดยตรงโดยไม่ต้องผ่านโมเด็ม โดยมีการตรวจสอบหรือแฮนด์เช็กเต็มรูปแบบ ส่วนอีกแบบหนึ่งเป็นการเชื่อต่อแบบ Null modem ในลักษณะที่ใช้สายสัญญาณเพียง 3 เส้น โดยเส้นหนึ่งสำหรับส่งข้อมูลอีกเส้นสำหรับรับข้อมูลและเส้นสุดท้ายกราวด์ สำหรับรายละเอียดหน้าที่การทำงานในแต่ละขาของพอร์ตอนุกรม RS-232

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Data Carrier Detect : DCD หรืออาจเรียกว่า Carrier Detect : CD ขานี้จะแอกทีฟเมื่อมีการส่งสัญญาณพาห์จากอุปกรณ์สื่อสารข้อมูล เช่น โมเด็ม สำหรับการใช้งานปกติขานี้จะไม่ได้ถูกใช้งานมากนัก

Receive Data : RD หรือ RxD ขานี้ใช้เพื่อรับสัญญาณอนุกรมเข้ามายังคอมพิวเตอร์ โดยนำข้อมูลที่อ่านได้เก็บไว้ในรีจีสเตอร์ บัฟเฟอร์

Transmitted Data : TD หรือ TxD ขานี้ใช้เพื่อส่งข้อมูลออกจากคอมพิวเตอร์ โดยนำข้อมูลที่เก็บอยู่ในบัฟเฟอร์สำหรับส่งข้อมูลออกไป

Data Terminal Ready: DTR เป็นขาสัญญาณที่ส่งออกจากคอมพิวเตอร์เพื่อให้อุปกรณ์ปลายทางรับรู้ว่าต้องการติดต่อด้วยโดยขา DTR นี้จะต้องเชื่อมต่อกับขา DSR ของอุปกรณ์ปลายทาง และขา DTR ของอุปกรณ์ปลายทางจะต้องเชื่อมต่อกับขา DSR ของคอมพิวเตอร์ ถ้าใช้การเชื่อมต่อเป็นแบบ Null Modem ซึ่งใช้สายในการเชื่อมต่อเพียง 3 เส้น จะต้องต่อขา DTR และ DSR ของตัวมันเองเข้าด้วยกันและต้องต่อกับขา DCD ด้วยในกรณีที่โปรแกรมสื่อสารที่ใช้มีการตรวจจับสัญญาณพาห์

Signal Ground: GND ขากราวด์ของระบบ

Data Set Ready : DSR ขานี้จะใช้คู่กับขา DTR เพื่อตรวจสอบการเชื่อมต่อกันระหว่างคอมพิวเตอร์กับอุปกรณ์ปลายทางซึ่งขา DSR นี้จะเป็นขาสำหรับใช้รับข้อมูล์ที่ส่งจากภายนอกซึ่งถูกส่งมาจากขา DTR

Request To Send : RTS เป็นขาสำหรับส่งสัญญาณร้องขอให้ทางอุปกรณ์ปลายทางส่งข้อมูลกลับมายังคอมพิวเตอร์โดยขาที่รับสัญญาณ RTS ก็คือขา CTS ในกรณีที่ใช้การเชื่อมต่อแบบ Null modem 3 สาย จะต้องเชื่อมต่อขา RTS และ CTS ของตัวมันเองเข้าด้วยกัน เพื่อจะให้การรับและส่งข้อมูลสามารถเกิดขึ้นได้ตลอดเวลา

Clear To Send: CTS ขานี้จะคอยรับสัญญาณจากขา RTS เมื่อรับสัญญาณได้ข้อมูลที่ขา TXD จะถูกส่งออกไป ดังนั้นขานี้จึงถูกใช้เพื่อตรวจสอบอุปกรณ์ต่อพ่วงว่าพร้อมที่จะรับข้อมูลหรือไม่

Ring Indicator : RI ใช้แสดงสถานะสัญญาณเรียกจากสายโทรศัพท์ ปกติในการสื่อสารโดยทั่วไปสายนี้จะไม่ถูกใช้งานจะใช้งานก็ต่อเมื่อมีการเชื่อมต่อกับโมเด็มและโปรแกรมมีการตรวจสอบสัญญาณนี้เท่านั้นผิดพลาดที่เกิดขึ้นระหว่างการถ่ายทอดข้อมูล (ผิดพลาดจากพาริตี้เฟรมข้อมูลโอเวอร์รัน) เป็นต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3 โปรแกรม AutoCAD R14

การใช้คอมพิวเตอร์ช่วยเขียนแบบและออกแบบ

การเขียนแบบและออกแบบโดยใช้คอมพิวเตอร์เป็นเครื่องมือช่วย (Computer Aided Design-CAD) จะเพิ่มขีดความสามารถทำงานได้มากกว่าการเขียนแบบด้วยมือเพราะคอมพิวเตอร์สารถเก็บบันทึกตำแหน่ง พิกัด ขนาด สี และแบบลายเส้นของวัตถุที่เขียนขึ้นโดยจัดเก็บไว้ในระบบฐานข้อมูล เพื่อนำแบบที่เขียนไว้ไปใช้งานต่อ หรือเพื่อการแก้ไขเพิ่มเติมในอนาคต

การใช้คอมพิวเตอร์ช่วยในเขียนแบบทำให้สามารถสร้างเส้นแบบต่างๆ ได้ตามต้องการ และแก้ไขดัดแปลงเพิ่มเติมได้เมื่อมีความผิดพลาดโดยไม่ต้องเสียเวลาเขียนแบบใหม่ทั้งหมด แบบที่เขียนด้วยคอมพิวเตอร์จะให้ความละเอียดถูกต้องและนำไปพิมพ์ออกมาได้ตามต้องการ สรุปแล้วข้อได้เปรียบของการใช้คอมพิวเตอร์ช่วยในการเขียนแบบก็คือ ความสะดวกรวดเร็ว และความถูกต้องแม่นยำกว่า

แต่ปัญหาของการเรียนรู้เพื่อนำคอมพิวเตอร์มาใช้งานในการเขียนแบบด้วยคอมพิวเตอร์ก็คือการขาดความมั่นใจที่จะนั่งหน้าเครื่องคอมพิวเตอร์และใช้แป้นพิมพ์หรือเม้าส์ในการเริ่มต้นเขียนแบบ แต่แท้จริงแล้วเครื่องคอมพิวเตอร์ก็คือเครื่องคิดเลข (Calculator) แบบหนึ่งเพียงแต่มีแป้นที่จะให้กดใช้งานได้มากมายและมีขั้นตอนที่แน่นอนในการสั่งงาน จึงไม่มีผู้ใดที่จะสามารถใช้งานเครื่องคอมพิวเตอร์ได้โดยไม่เรียนรู้จากหนังสือคู่มือหรือมีผู้แนะนำ ดังนั้นผู้ที่ต้องการใช้เครื่องคอมพิวเตอร์ช่วยเขียนแบบจนถึงขั้นที่เกิดความมั่นใจจะต้องมีคู่มือหรือผู้ชี้แนะที่ดีและหัวใจสำคัญก็คือจะต้องมีการฝึกปฏิบัติอยู่เสมอ

การเขียนแบบด้วย AutoCAD นั้นจะต้องเรียนรู้การออกคำสั่งให้โปรแกรมเขียนเส้นตรง เส้นโค้ง วงกลม หรือตัวหนังสือ ผู้ใช้งานจะต้องรู้จักการป้อนข้อมูลและพารามิเตอร์ต่างๆ ให้คอมพิวเตอร์ตีความ โดยใช้แป้นพิมพ์หรือเลือกจากเมนูโดยใช้อุปกรณ์ชี้พารามิเตอร์ที่โปรแกรมคอมพิวเตอร์ต้องการทราบ คือ ตำแหน่ง ขนาด หรือ สี ของเส้นต่างๆ เป็นต้น หลังการออกคำสั่งและป้อนข้อมูลต่างๆ แล้ว คอมพิวเตอร์ก็จะวาดเส้นตามที่ต้องการให้ปรากฏบนจอภาพ

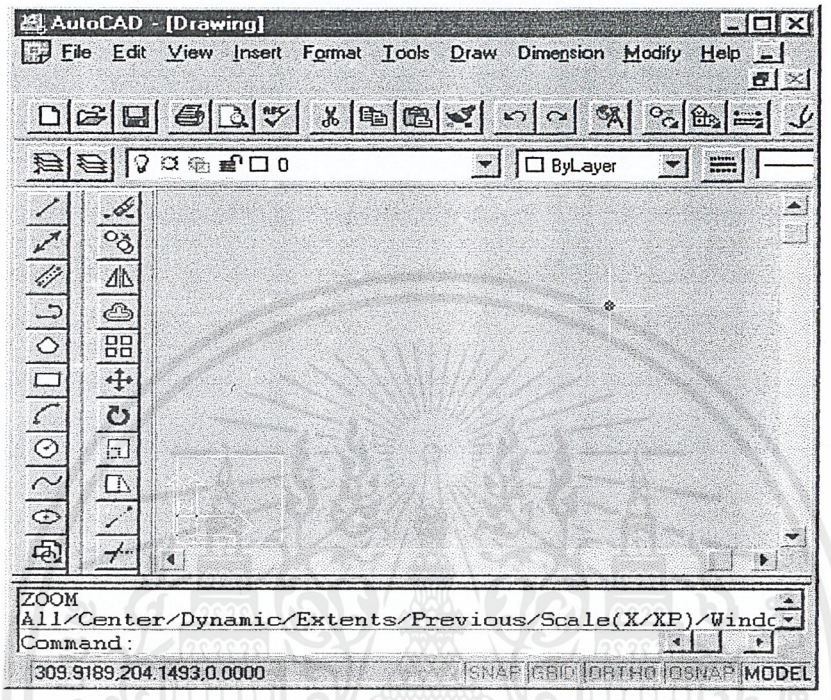
นอกจากการเขียนเส้นต่างๆ แล้ว AutoCAD ยังมีความสามารถในการแก้ไขเส้นที่วาดไว้ เช่น การลบ เคลื่อนย้ายหรือลอกแบบได้รวดเร็วกว่าการเขียนด้วยมือรวมทั้งการแสดงภาพหรือขยายภาพให้สามารถมองได้ชัดเจนขึ้น การพล๊อตแบบที่เขียนไว้ด้วยคอมพิวเตอร์ก็สามารถส่งข้อมูลออกมาที่เครื่อง พล๊อต (Plotter) ให้ได้ขนาดและสีสันตามต้องการ

AutoCAD เป็นโปรแกรมคอมพิวเตอร์ที่มีความเหมาะสมสำหรับงานเขียนแบบลายเส้นดังต่อไปนี้

- งานเขียนแบบด้านสถาปัตยกรรม
- งานออกแบบตกแต่งภายในและแบบลายเส้นด้านศิลปกรรม
- การเขียนแผนผัง flowchart และเขียนกราฟ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- งานเขียนแบบทางวิศวกรรมทั่วไป
- งานเขียนแบบด้านสำรวจภูมิประเทศและภูมิทัศน์



ภาพที่ 3.4 แสดงหน้าต่างของโปรแกรม Auto CAD R14

3.4 การเขียนโปรแกรมด้วย Visual Basic

3.4.1 คอนโทรลภายในของ Visual Basic

Visual Basic เป็นเครื่องมือที่ช่วยพัฒนาโปรแกรมแอพพลิเคชั่นสำหรับวินโดว์ตัวแรกที่ประสบความสำเร็จเป็นอย่างมากทั้งนี้เนื่องมาจากแนวความคิดที่จะนำเอาความสามารถของคอนโทรลมาใช้ในการออกแบบโปรแกรม เพราะคอนโทรลเป็นเครื่องมือที่ช่วยลดความซับซ้อนในการเขียนโค้ดลงไปได้มากทีเดียวและนอกจากนี้คอนโทรลยังมีส่วนที่แสดงผลเพื่อสื่อความหมายของการทำงานระหว่างคอนโทรลและผู้ใช้ได้อีกด้วย ส่วนการใช้งานก็ไม่มีความซับซ้อนเพียงแต่ผู้อ่านทำการเชื่อมต่อคอนโทรลเข้ากับสภาพแวดล้อมของ Visual Basic จากนั้นก็สามารถที่จะนำมาเพิ่มลงในฟอร์มได้ทันที สำหรับ Visual Basic ได้มีการแบ่งคอนโทรลออกเป็น 4 กลุ่มหลักๆ ที่สำคัญดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1. คอนโทรลภายใน (Intrinsic Control) เช่น ComboBox , Command Button หรือ PictureBox เป็นต้น ซึ่งเป็นคอนโทรลที่ถูกสร้างลงในสภาพแวดล้อมของ vb.exe ดังนั้นทุกครั้งที่ผู้อ่านโหลด Visual Basic คอนโทรลเหล่านี้ออกจากแถบกล่องเครื่องมือได้เลย ดังนั้นจึงจัดได้ว่าเป็นคอนโทรลมาตรฐาน(Standard Control) กลุ่มหนึ่งของ Visual Basic
2. คอนโทรลมาตรฐาน (Standard Control) เป็นคอนโทรล ActiveX ที่ถูกสร้างเป็นไฟล์ .ocx ที่แยกออกมาต่างหาก เช่น Dbgrid (Apex data-bound grid) , MSFlexGrid หรือ CommonDialog เป็นต้น ดังนั้นก่อนที่จะสามารถใช้งานคอนโทรลในกลุ่มนี้ได้ เราต้องทำการเชื่อมต่อไฟล์ .ocx เหล่านี้เข้ากับสภาพแวดล้อมของ Visual Basic เสียก่อนโดยใช้คำสั่ง Components ในเมนู Project เช่นเดียวกัน
3. คอนโทรลร่วมวินโดวส์ (Windows Common Control) เป็นคอนโทรล ActiveX ที่ถูกสร้างเป็นไฟล์ .ocx ที่ต้องใช้ร่วมกับไฟล์ .dll ของวินโดวส์ เช่น RichTextBox , Slider หรือ Statusbar เป็นต้น เช่นเดียวกับคอนโทรลมาตรฐาน เพียงแต่คอนโทรลในกลุ่มนี้ได้ถูกจัดเป็นคอนโทรลพื้นฐานของวินโดวส์ 95 โดยที่คอนโทรลร่วมกับวินโดวส์จะถูกจัดเก็บลงในไฟล์ conctl32.ocx และ conct232.ocx
4. คอนโทรล ActiveX รุ่นมืออาชีพ (Professional ActiveX Control) เป็นคอนโทรล ActiveX ที่ถูกสร้างเป็นไฟล์ .ocx เช่นเดียวกับคอนโทรลมาตรฐาน เช่น MSComm (Communication) , MapiMessages (MAPI message) หรือ MMControl (Multimedia MCI) เป็นต้น แต่คอนโทรลในกลุ่มนี้ได้ถูกสร้างและแจกจ่ายมากับ Visual Basic รุ่น Professional และ Enterprise เท่านั้น

3.4.2 คุณสมบัติแสดงค่าของคอนโทรลที่สำคัญ

คอนโทรลทั้งหมดที่มากับ Visual Basic ไม่ว่าจะเป็นคอนโทรลภายในหรือ ActiveX จะมีคุณสมบัติตัวหนึ่งที่ถูกนำมาใช้สำหรับการกำหนดค่า (Value) หรืออ่านจากค่าคอนโทรล และคุณสมบัตินี้ได้ถูกกำหนดให้เป็นคุณสมบัติปกติ (Default) ของคอนโทรล โดยในการเขียนโค้ดเราสามารถใช้เพียงชื่อของคอนโทรล (Control Name) โดยไม่ต้องกำหนดคุณสมบัติปกติของทุกๆ คอนโทรลได้โดยไม่เกิดข้อผิดพลาด เช่น คอนโทรล TextBox ก็จะมีคุณสมบัติ Text เป็นคุณสมบัติปกติของคอนโทรล สำหรับคุณสมบัติปกติของคอนโทรลที่สำคัญมีดังต่อไปนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำนักหอสมุดกลาง จุฬาลงกรณ์มหาวิทยาลัย

ตารางที่ 3.3 แสดงชื่อคอนโทรลที่สำคัญ

คอนโทรล	คุณสมบัติ
Command Button	Value
DATA	Caption
MsflexGrid	Text
Label	Caption
TextBox	Text
Timer	Enable

3.4.3 การแบ่งกลุ่มของคอนโทรลภายใน

เราจะแยกตามวัตถุประสงค์ของการใช้งานได้ทั้งหมด 4 กลุ่ม

1. คอนโทรลภายในทั่วไป ประกอบด้วยคอนโทรลที่แสดงผลในลักษณะของการเลือกตอบหรือเลือกรายการ เช่น CheckBox , OptionBox หรือ ListBox เป็นต้น
2. คอนโทรลภายในด้านระบบไฟล์ ประกอบไปด้วยคอนโทรลที่ทำหน้าที่ในการติดต่อหรือ แสดงผลระบบไฟล์ (รวมทั้งไดรฟ์ และ ไดเร็คทอรี่ด้วย) ของวินโดวส์ เช่น FileListBox หรือ DirListBox เป็นต้น
3. คอนโทรลภายในด้านกราฟิก ประกอบด้วยคอนโทรลที่ทำหน้าที่ด้านการแสดงผลกราฟิกด้วยวิธีกราฟิกคอนโทรล หรือ ฟังก์ชันวินโดวส์ API หรือไฟล์กราฟิกในรูปแบบต่างๆ เช่น PictureBox , Shape หรือ Image เป็นต้น
4. คอนโทรลภายในด้านเวลา ซึ่งจะมีอยู่คอนโทรลเดียวได้แก่ Timer ซึ่งมีหน้าที่สร้างเหตุการณ์ที่ตอบสนองเป็นครั้งๆตามช่วงเวลาที่ถูกกำหนด

3.4.4 คอนโทรลภายใน

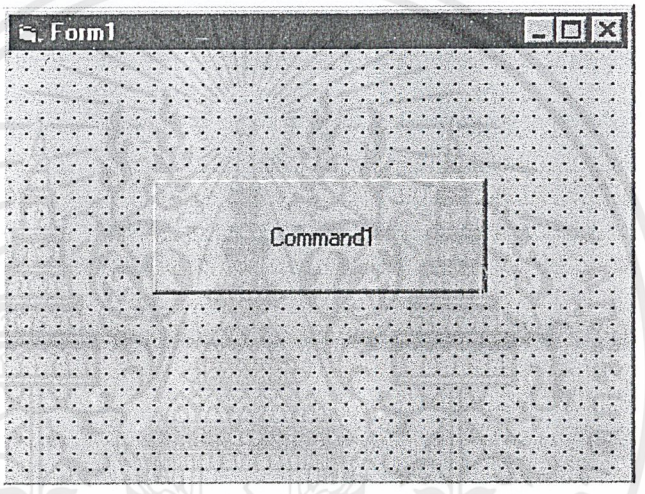
คอนโทรลภายในก็จะเป็นคอนโทรลพื้นฐานที่ถูกนำไปใช้งานมากที่สุดเพราะจะเป็นกลุ่มคอนโทรลที่ช่วยในการสื่อสารแบบสองทางหรือรับเลือกเงื่อนไขจากผู้ใช้ดังเช่นทุกๆ แอพลิเคชั่นจะใช้คอนโทรล Command Button สำหรับให้ผู้ใช้เลือกที่ยอมรับ (OK) ยกเลิก (Cancle) หรือ อื่นๆ ตามข้อกำหนดของแต่ละแอพลิเคชั่นเป็นต้นซึ่งคอนโทรลภายในทั่วไปจะประกอบไปด้วยคอนโทรลต่างๆ ดังต่อไปนี้ คอนโทรล Command Button

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

42540

3.4.4.1 คอนโทรล Command Button

คอนโทรล Command Button จะเป็นคอนโทรลที่ถูกนำไปใช้งานมากที่สุด เพราะกำหนดให้ผู้ใช้งานเลือก OK หรือ Cancle นั้น เรามักจะใช้คอนโทรล Command Button เป็นส่วนใหญ่ ดังนั้นจึงถือว่าเป็นคอนโทรลที่พื้นฐานที่สุดของ Visual basic เนื่องจากคอนโทรลนี้เป็นปุ่มสำคัญที่ใช้งานในรูปแบบของการคลิกเพื่อยืนยัน ดังนั้นเรียกคอนโทรล Command Button ได้อีกอย่างว่า PushButton ในขณะออกแบบคอนโทรล Command Button ที่วางลงบนฟอร์มจะมีลักษณะดังตัวอย่างในภาพนี้



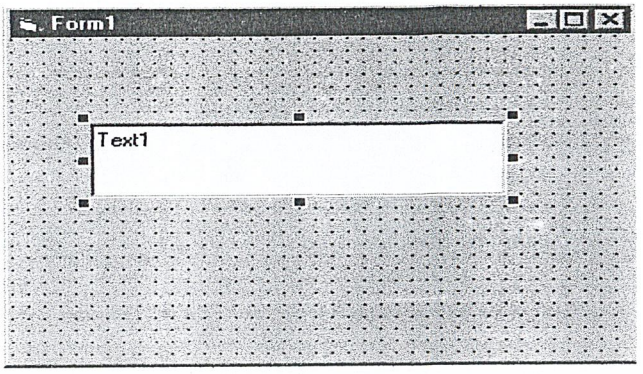
ภาพที่ 3.5 แสดงคอนโทรล ControlButton ในขณะออกแบบ

เราสามารถแก้ไขข้อความที่แสดงผลในคอนโทรลนี้โดยการแก้ไขข้อความในคุณสมบัติ Caption ของคอนโทรลในหน้าต่างคุณสมบัติหรือแก้ไขโค้ดในแอพพลิเคชั่นก็ได้

3.4.4.2 คอนโทรล TextBox

คอนโทรล TextBox มักจะถูกนำไปใช้ทุกๆ ฟอร์มที่มีการรับการกรอกข้อความจากผู้ใช้เนื่องจากคอนโทรลนี้ทำหน้าที่แสดงข้อมูล (โดยผ่านทางคุณสมบัติ Text) ในคอนโทรล และยังอนุญาตให้ผู้ใช้สามารถแก้ไขตัวอักษรต่างๆ ของคุณสมบัตินี้ได้ด้วยเช่นกันนอกจากนี้แล้ว คอนโทรล TextBox ยังได้รวมเอาความสามารถหลายๆด้านของคอนโทรล Label เช่น สามารถแสดงข้อความมากกว่า 1 บรรทัด ความสามารถด้าน DDE (Dynamic Data Exchange) นอกจากนี้ยังสามารถถูกนำไปใช้ในลักษณะของการกรอกรหัสผ่าน (Password) ได้อีกด้วยดังภาพ

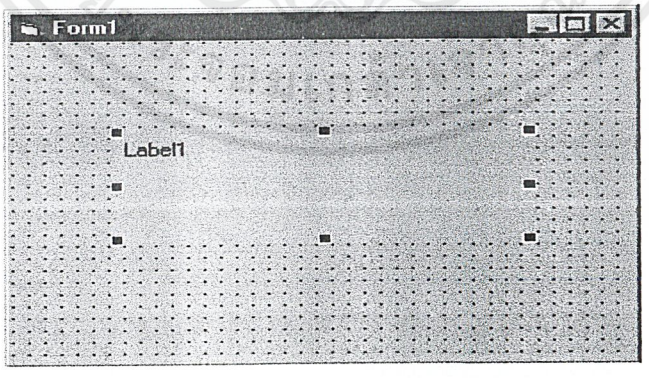
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 3.6 แสดงคอนโทรล TextBox

3.4.4.3 คอนโทรล Label

คอนโทรล Label เป็นคอนโทรลในลักษณะของกราฟิกที่ถูกใช้งานด้านการแสดงผลข้อความบนฟอร์มเหมือนกับผู้อ่านได้นำป้ายข้อความอย่างหนึ่งไปวางไว้บนฟอร์ม เพื่อใช้ในการสื่อข้อความกับผู้ใช้และคอนโทรลนี้ผู้ใช้ไม่สามารถแก้ไขได้โดยตรงด้วยวิธีการคีย์หรือใช้เมาส์ในขณะรันแอพพลิเคชั่น นอกเสียจากภายในแอพพลิเคชั่นจะมีการเขียนโค้ดสำหรับ แก้ไขข้อความในคอนโทรลโดยการแก้ไขค่าคุณสมบัติ Caption เท่านั้น และนอกจากนี้ Label ยังเป็นคอนโทรลที่มีความสามารถด้าน DDE (Dynamic Data Exchange) อีกด้วย ในขณะออกแบบเราสามารถเพิ่มคอนโทรลลงในฟอร์ม หรือตัวบรรจุอื่นๆ ก็จะปรากฏหน้าตาของคอนโทรลดังภาพ



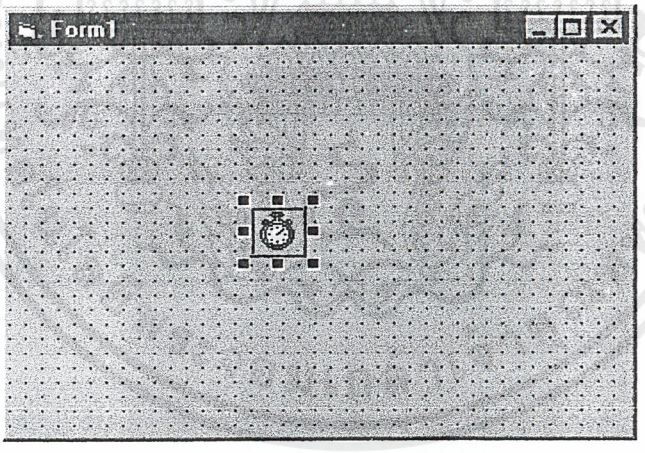
ภาพที่ 3.7 แสดงคอนโทรล Label

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในการควบคุมพฤติกรรมของคอนโทรล เราสามารถกระทำได้โดยการกำหนดค่าต่างๆ ให้กับคุณสมบัติของคอนโทรล ซึ่งการแก้ไขของคุณสมบัติเราสามารถกระทำได้ทั้งในขณะออกแบบโดยการแก้ไขค่าในหน้าต่างคุณสมบัติ และรับแอพพลิเคชั่นโดยการเขียนโค้ดแก้ไขค่าคุณสมบัติแต่ก็มีบางคุณสมบัติที่ไม่สามารถแก้ไขได้ในขณะรันแอพพลิเคชั่น เช่น Name เป็นต้น

3.4.4.4 คอนโทรล Timer

คอนโทรล Timer เป็นคอนโทรลที่ใช้ในการควบคุมและจัดการเหตุการณ์ด้านเวลา ซึ่งเทียบได้กับประโยค ON TIME GOTO ของ QuickBasic โดยเราสามารถเขียนโค้ดเพื่อทำงานใดๆ เมื่อช่วงเวลาผ่านไปตามค่าที่กำหนด เช่น ทำการปรับการแสดงผลของฟอร์มทุกๆ 1 นาที เป็นต้น โดยที่คอนโทรลนี้จะตอบสนองเหตุการณ์เพียงเหตุการณ์เดียวเท่านั้น แต่เราสามารถกำหนดให้แต่ละฟอร์มมีคอนโทรล Timer มากกว่า 1 คอนโทรล เนื่องจากคอนโทรล Timer เป็นคอนโทรลที่ทำงานตามนาฬิกาของระบบดังนั้นมันจึงถูกควบคุมโดยตัวของระบบเอง สำหรับวินโดวส์ 95 และ NT ในทางปฏิบัติจะไม่มีการจำกัดจำนวนของคอนโทรล Timer ในแต่ละฟอร์ม ดังนั้นเราจึงสามารถใช้งานคอนโทรล Timer พร้อมๆ กันครั้งละหลายๆคอนโทรล ได้อย่างไม่จำกัด



ภาพที่ 3.8 แสดงคอนโทรล Timer

ในขณะออกแบบคอนโทรล Timer ที่วางลงบนฟอร์มให้กับฟอร์ม ก็จะมีลักษณะดังภาพที่ 3.8 และเมื่อรันแอพพลิเคชั่น คอนโทรลนี้จะไม่ถูกแสดงผลแต่จะมีการทำให้เกิดเหตุการณ์ Timer ทุกครั้งที่ช่วงเวลาครบตามค่าที่ได้กำหนดให้กับคุณสมบัติ Interval ของคอนโทรล Timer

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.4.5 คอนโทรลด้านฐานข้อมูล

แอพพลิเคชั่นที่จะต้องมีการเข้าถึงไฟล์ฐานข้อมูล ตัวแปลภาษาที่เหมาะสมกับการสร้างแอพพลิเคชั่นเหล่านี้ จึงต้องมีเครื่องมือที่สามารถสนับสนุนการจัดการฐานข้อมูลอย่างง่ายและมีประสิทธิภาพ Visual Basic เป็นตัวแปรภาษาที่มีการสนับสนุนระบบจัดการฐานข้อมูลในรูปแบบ Microsoft Access โดยอาศัย JET Datbase Engine ซึ่งเป็นเครื่องมือที่ให้โปรแกรมเมอร์สามารถจัดการกับฐานข้อมูลได้ 2 วิธีดังนี้

1. คอนโทรลด้านฐานข้อมูล (Data Control)

การเข้าถึงฐานข้อมูลด้วยคอนโทรลด้านฐานข้อมูลจะเป็นวิธีที่ง่ายที่สุดในการเขียนโปรแกรมเพราะเราเพียงแต่กำหนดไฟล์ฐานข้อมูลฟิลด์สำหรับแต่ละคอนโทรลและคอนโทรล Data เท่านั้น คอนโทรลเหล่านี้จะจัดการสร้าง การแสดงผล การแก้ไขฟิลด์ต่างๆ ของฐานข้อมูลให้โดยอัตโนมัติหรือตามที่ถูกกำหนดในคุณสมบัติต่างๆ สำหรับความสามารถโดยทั่วไปของคอนโทรลด้านฐานข้อมูลมีรายละเอียดดังต่อไปนี้

- สามารถสร้างตัวแปร Recordset โดยอ้างอิงกับตัวแปร Recordset ที่สนับสนุนโดยคอนโทรลด้านฐานข้อมูล
- แก้ไขโครงสร้างของฐานข้อมูล เช่นการแก้ไขตาราง (Table) ฟิลด์ (Field) หรือ ดัชนี (Index) เป็นต้น
- ค้นหาหรือสืบค้นข้อมูลจากฟิลด์ที่ถูกกำหนดของฐานข้อมูล

คอนโทรลที่สนับสนุนการติดต่อกับฐานข้อมูล (Bound Control) นั้น เราสามรถสังเกตุได้จากคอนโทรลที่มีคุณสมบัติ Datafield , Dtatchange หรือ DataSource เป็นต้น ซึ่งคอนโทรลที่มีคุณสมบัติเหล่านี้ก็จะเป็นคอนโทรลด้านฐานข้อมูลทั้งสิ้น เช่น CheckBox , PictureBox , Data , ProgressBar , RichTextBox หรือ MaskedEdit เป็นต้น

2. Data Access Object (DAO)

DAO เป็นโมเดลของคลาสของออบเจ็กต์ที่สนับสนุนการจัดการในเรื่องฐานข้อมูลในระบบ RelationalDatabase ซึ่งก็จะประกอบด้วยคุณสมบัติ โพรซีเยอร์เหตุการณ์และก็วิธีเพื่อช่วยในการสร้าง แก้ไข จัดเก็บค้นหาและลบฐานข้อมูลโดยในการจัดการกับฐานข้อมูลในทางปฏิบัติ Visual Basic ก็จะอาศัยความสามารถของ JET Database Engine (JET) ซึ่งเป็นเอนจิ้นท์ที่ช่วยในการจัดการฐานข้อมูลด้วยกายภาพสำหรับ Visual Basic ฐานข้อมูลในรูปแบบ Microsoft Access จะถูกกำหนดให้เป็นฐานข้อมูลแม่แบบ (Native Database Format) ที่ถูกจัดการโดย JET ซึ่งกลุ่มชนิดของฐานข้อมูลที่ Visual Basic สนับสนุนโดยผ่านทาง DAO และ JET สามารถแบ่งออกได้เป็น 3 กลุ่มดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ฐานข้อมูลของ Visual Basic หมายถึง ฐานข้อมูลในรูปแบบของ Microsoft Access (.mdb)ซึ่ง JET สามารถที่จะจัดการได้โดยตรง ดังนั้นจึงเป็นรูปแบบของฐานข้อมูลที่เอ็นจิ้นท์ JET สามารถจัดการได้รวดเร็วที่สุด
- ฐานข้อมูลภายนอก หมายถึง ฐานข้อมูลในรูปแบบ Indexed Sequential Access Method (ISAM) ซึ่งเป็นรูปแบบฐานข้อมูลสำหรับโปรแกรมด้านฐานข้อมูลระดับเดสก์ทอปในปัจจุบันนั่นเอง เช่น dBase IV, Foxpro, Paradox เป็นต้น ซึ่งผู้อ่านสามารถจัดการกับฐานข้อมูลในรูปแบบ ISAM ได้โดยตรงจาก Visual Basic เท่านั้น
- ฐานข้อมูล ODBC หมายถึง ฐานข้อมูลที่ทำการติดติดโดยผ่านทาง Open Database Connectivity (ODBC) ซึ่งเป็น มาตรฐานการจัดการกับฐานข้อมูลในรูปแบบที่แตกต่างกันของไมโครซอฟท์ เช่น ฐานข้อมูลชนิด ไคลเอนต์ เซิร์ฟเวอร์ Microsoft SQL Server เป็นต้น ดังนั้นเราจึงสามารถที่จะสร้าง แอพพลิเคชั่น ชนิด ไคลเอนต์ เซิร์ฟเวอร์ ด้านฐานข้อมูลได้โดยตรงจาก Visual Basic โดยมาตรฐาน ODBC นี้เอง

3.4.6 Microsoft JET Database Engine

ในการเขียนโปรแกรมเพื่อจัดการกับฐานข้อมูลของ Visual Basic นั้นเราสามารถใช้อินเตอร์เฟสของ DAO ในการกำหนดวิธีการจัดการกับฐานข้อมูลที่ต้องการแต่ในทางปฏิบัติจริงๆ DAO ก็จะต้องมีการส่งคำสั่งต่อเนื่องไปยัง JET ที่มากับ Visual Basic 5.0 ก็จะเป็นเวอร์ชั่น 3.5 ซึ่งจะเป็นเอ็นจิ้นท์ระดับ 32 บิทที่ทำงานภายใต้ระบบปฏิบัติการวินโดวส์ 32 บิทเท่านั้นโดยจะมีความเสถียร ประสิทธิภาพและสมรรถนะการประมวลผลรวดเร็วกว่า JET เวอร์ชั่นที่ผ่านๆ มาก

- User Interface ผู้อ่านต้องเขียนโค้ดโดยอาศัย DAO หรือ คอนโทรลด้านฐานข้อมูลเพื่อจัดการเก็บฐานข้อมูลที่ต้องการโดยแสดงผลฟิลด์ต่างๆ ของฐานข้อมูลนั้น ผู้อ่านก็ต้องกระทำโดยอาศัยคอนโทรล หรือ ฟอร์มสำหรับการสร้าง แก้ไข ลบทิ้ง จัดเก็บ หรือ ค้นหาข้อมูลในฐานข้อมูล Visual Basic ก็กระทำผ่านทางเอ็นจิ้นท์ JET อีกทอดหนึ่ง

- Database Engine JET เป็นไฟล์ไลบรารี่ (.dll) ที่ประกอบด้วยฟังก์ชันที่ช่วยในการจัดเก็บกับไฟล์ฐานข้อมูลในระดับกายภาพ ซึ่งในการทำงาน JET จะแปลงการร้องขอจาก Visual Basic เพื่อที่จะดำเนินการกับฟิลด์ต่างๆ ของไฟล์ฐานข้อมูล และรายงานข้อมูลกลับไปยังโค้ดส่วนที่ร้องขอ (ถ้าหากมี) และนอกจากนี้ JET ยังสนับสนุนการค้นหาหรือประมวลผลฐานข้อมูลโดยอาศัยภาษา SQL(Structured Query Language) อีกด้วย

- Data Store ไฟล์ที่จัดเก็บฐานข้อมูลและรายละเอียดทั้งหมดของฐานข้อมูล โดยที่ไฟล์ข้อมูลแม่แบบของ Visual Basic ก็จะจัดเก็บในรูปแบบของ Microsoft Access โดยมีนามสกุล .mdb โดยที่ข้อมูลที่จัดเก็บในไฟล์ฐานข้อมูลจะถูกจัดระเบียบโดยรูปแบบของฐานข้อมูลแต่ละประเภท ซึ่งเอ็นจิ้นท์ JET จะทำหน้าที่จัดการกับข้อมูลให้โดยอัตโนมัติ ซึ่งเราสามารถที่จะจัดการกับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ฐานข้อมูลหลากหลายชนิดโดยไม่จำเป็นต้องทราบถึงโครงสร้างของไฟล์ฐานข้อมูลชนิดนั้นๆ เลยซึ่งก็เป็นจุดเด่นอีกประการของระบบฐานข้อมูลของ Visual basic

คอนโทรลด้านฐานข้อมูลเป็นคอนโทรลที่ใช้ในการเข้าถึงฐานข้อมูลคอมพิวเตอร์ ซึ่งหลักๆ สามารถแบ่งออกได้เป็น 3 กลุ่มดังนี้

1. คอนโทรล Data เป็นคอนโทรลหลักที่ใช้ในการควบคุมการติดต่อระหว่างคอนโทรลด้านฐานข้อมูลกับฐานข้อมูลโดยที่คอนโทรล Data จะทำหน้าที่ควบคุมการเข้าถึงฐานข้อมูล เช่น การเคลื่อนที่ไปยังเรคคอร์ด การเปิดปิด การจัดเก็บฐานข้อมูล เป็นต้น

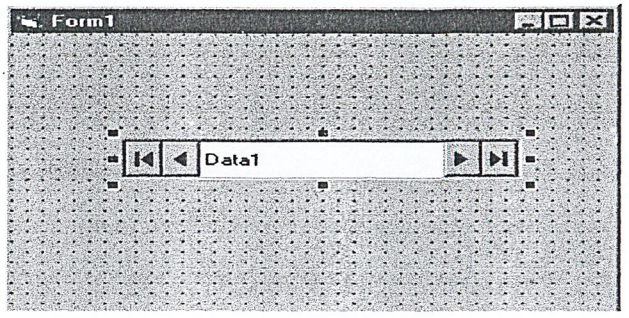
2. คอนโทรลภายใน Data-Aware เป็นคอนโทรลภายในของ visual basic ที่สนับสนุนคุณสมบัติการแสดงผลข้อมูลของฟิลด์หนึ่งๆ ของฐานข้อมูล เช่น CheckBox , PictureBox หรือ TextBox เป็นต้น ซึ่งคอนโทรลภายใน Dtat-Aware จะแตกต่างกับคอนโทรล Data-Bound ที่คอนโทรลภายในData-Awareจะเชื่อมต่อเข้ากับฟิลด์ของฐานข้อมูลคอนโทรลละ 1 ฟิลด์เท่านั้น

3. คอนโทรล Data-Bound เป็นคอนโทรล ActiveX ที่ถูกออกแบบพิเศษเพื่อให้สามารถเชื่อมต่อกับ Record (ประกอบด้วยตั้งแต่ 1 ฟิลด์ขึ้นไป) ของฐานข้อมูล เช่น Dblist , DbCombo , DbGrid และ MSFrexGrid ซึ่งคอนโทรล Dtat-Bound จะสามารถเชื่อมต่อกับหลายๆ เรคคอร์ดได้ในเวลาเดียวกันทั้งนี้เพื่อขยายขีดความสามารถในการจัดการฐานข้อมูลนั่นเอง

3.4.5 คอนโทรล Data

คอนโทรล data เป็นคอนโทรลที่ใช้ในการเข้าถึงฐานข้อมูลคอมพิวเตอร์และทำการเชื่อมต่อการแสดงผลข้อมูลของแต่ละฟิลด์ในฐานคอมพิวเตอร์ข้อมูลเข้ากับคอนโทรลด้านฐานข้อมูล (Data-Bound or Data-Aware Control) โดยเมื่อมีการเคลื่อนที่ไปยังเรคคอร์ดใดๆ ในฐานข้อมูลด้วยคอนโทรล Data ข้อมูลที่ถูกแก้ไขในคอนโทรลด้านฐานข้อมูลที่เชื่อมต่อกับคอนโทรล Data ซึ่งเป็นเรคคอร์ดปัจจุบันในขณะนั้นก็จะถูกจัดเก็บลงในฐานข้อมูลโดยอัตโนมัติแล้วจึงเคลื่อนที่ไปยังเรคคอร์ดถัดไปทันที แต่เนื่องจากคอนโทรล Data สามารถที่จะจัดเก็บข้อมูลที่ถูกแก้ไขให้โดยอัตโนมัติ ดังนั้นถ้าหากเราต้องการตรวจสอบความถูกต้องของข้อมูลก่อนที่จะถูกจัดเก็บโดยคอนโทรล Data ก็สามารถกระทำได้โดยการเขียนโค้ดเพื่อตรวจสอบความถูกต้องของข้อมูลในโพรซีเยอร์ของเหตุการณ์ Validate ในขณะออกแบบคอนโทรล Data ที่วางลงบนฟอร์มจะมีลักษณะดังภาพ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

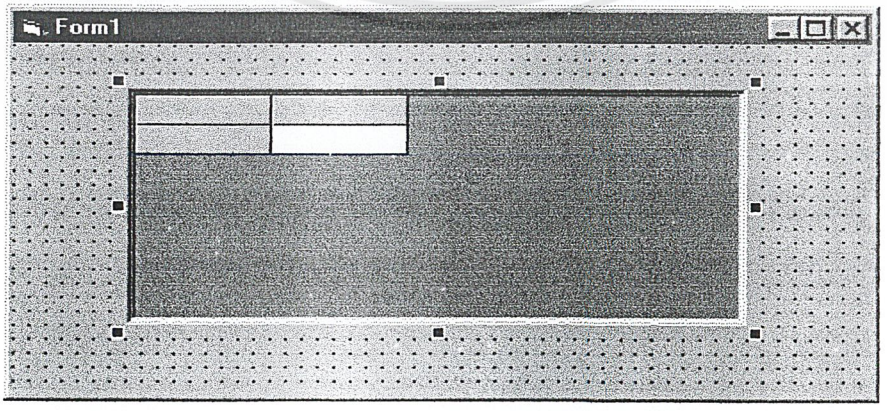


ภาพที่ 3.9 แสดงคอนโทรล Data

3.4.6 คอนโทรล MsflexGrid

คอนโทรล MsflexGrid เป็นวัตถุที่แสดงผลค่อนข้างมีประสิทธิภาพ เมื่อเข้าสู่โปรแกรม ถ้ามองไม่เห็นวัตถุควบคุมเอ็มเอสเฟลกกริดในกล่องเครื่องมือ ให้เพิ่มวัตถุควบคุมนี้เข้าไปโดยใช้เมนู Project เป็นวัตถุที่สามารถแสดงผลข้อมูลออกมาเป็นแถวและคอลัมน์ เช่นเดียวกับตาราง การทำงานของเอกซ์เซล ข้อมูลอาจจะเป็นข้อความหรือตัวเลขก็ได้ และยังมีความสามารถที่จะเชื่อมเซลล์เข้าด้วยกัน เรียงลำดับข้อมูลตามแนวคอลัมน์ได้อีกด้วย ความสามารถของคอนโทรล FlexGrid

- ข้อมูลที่แสดงจะสามารถอ่านได้อย่างเดียวไม่สามารถแก้ไขได้
- สามารถย้ายคอลัมน์และแถวได้
- สามารถจัดกลุ่มข้อมูลที่เหมือนกันเข้าเป็นเซลเดียวได้
- แต่ละเซลในคอนโทรลสามารถแสดงได้ทั้งข้อความและรูปภาพ
- ใช้งานร่วมกับ Data Control ในการแสดงข้อมูลฐานข้อมูลได้
- มีฟังก์ชันในการค้นหา และแทนที่ข้อความในคอนโทรล
- สามารถแก้ไขค่าของข้อมูลได้จากคำสั่งในโปรแกรม



ภาพที่ 3.10 แสดงคอนโทรล MsflexGrid

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.4.7 คอนโทรลคอนโทรล MSComm (Communication)

เนื่องจากในปัจจุบันมีการใช้งานมาตรฐานการเชื่อมต่อแบบ RS 232Cกันอย่างแพร่หลาย ซึ่งเป็นมาตรฐานถูกกำหนดโดย EIA ซึ่งเป็นองค์กรอุตสาหกรรมอิเล็กทรอนิกส์ ของสหรัฐอเมริกา โดยแบ่งการเชื่อมต่อออกเป็น 2 ลักษณะ คือ DTE (Data Terminal Equipment) และ DCE (Data Communication Equipment) ซึ่งโดยปกติ DTE จะต้องต่อเข้ากับ DCE เสมอ เช่น การต่อเครื่องคอมพิวเตอร์ (อุปกรณ์ DTE) เข้ากับอุปกรณ์โมเด็ม (อุปกรณ์ DCE)เป็นต้น

พอร์ทอนุกรม RS 232C จะเป็นพอร์ทของเครื่องคอมพิวเตอร์ที่มีขาต่อ (Connector) ทั้งประเภท 9 และ 25 ขาและเรียกว่า พอร์ท COM1: และ COM2: นั่นเอง ในความจริงพอร์ทอนุกรมไม่ได้ถูกควบคุมโดยตรงจาก CPU บนเมนบอร์ด แต่การสื่อสารทั้งหมดจะถูกเก็บโดยชิป UART (Universal Asynchronous Receiver/Transmitter) อีกทีหนึ่ง ซึ่งในปัจจุบันเบอร์ที่ใช้กันมากที่สุดก็คือ เบอร์ 16550C ซึ่งเป็นเวอร์ชั่นที่ได้รับการแก้ไขข้อผิดพลาดแล้ว ซึ่งชิป UART นี้จะทำหน้าที่ในการรับและส่งข้อมูลดังต่อไปนี้

การส่งข้อมูล(Data transmission)

- รับตัวอักษรจากเครื่องคอมพิวเตอร์
- แปลงตัวอักษรให้เป็นสายข้อมูลแบบบิต (เราเรียกว่าขบวนการ Serialization)
- สร้างเฟรมข้อมูล โดยการเพิ่มบิตที่จำเป็นสำหรับการสื่อสารและการตรวจสอบเช่น บิต Start , Stop และ Parity เป็นต้น
- ส่งผ่านเฟรมข้อมูลที่สร้างขึ้นมาแล้วจากขั้นตอนที่ผ่านมาด้วยความเร็วของโมเด็มหรือพอร์ทอนุกรม(Baud Rate)
- แสดงสถานะความพร้อมที่จะรับข้อมูลตัวอักษรถัดไปให้กับเครื่องคอมพิวเตอร์ในการรับข้อมูล (Data Receiver)
- รับตัวอักษรจากตัวอินเตอร์เฟส
- ตรวจสอบความถูกต้องจองเฟรมข้อมูลตามมาตรฐานเฟรมที่กำหนดโดยถ้าหากเฟรมข้อมูลมีรูปแบบที่มไถูกต้องก็จะมีการแจ้งข้อผิดพลาดทันที
- ตรวจสอบความถูกต้องของพาริตี้
- แปลงสายข้อมูลแบบบิตให้เป็นตัวอักษร
- ส่งตัวอักษรให้กับเครื่องคอมพิวเตอร์
- แสดงสถานะความพร้อมที่รับข้อมูลตัวอักษรถัดไปให้กับอินเตอร์เฟส

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คอนโทรล MSComm เป็นคอนโทรลตัวหนึ่งที่ช่วยในการติดต่อกับพอร์ทอนุกรม (Serial Port) ซึ่งผู้อ่านสามารถรับส่งข้อมูลผ่านทางพอร์ทอนุกรมได้ด้วยคอนโทรลนี้ เช่น การติดต่อผ่านทางโมเด็ม หรือ ติดต่อโดยตรงกับบอร์ดอิเล็กทรอนิกส์ เป็นต้น ซึ่งคอนโทรล MSComm ที่มากับ Visual Basic จะเป็นคอนโทรลที่ทำงานโดยมีการตอบสนองต่อเหตุการณ์แบบ Event-Driven นั่นก็คือ คอนโทรลจะทำหน้าที่ตรวจสอบการเกิดขึ้นหรือร้องขอให้เกิดเหตุการณ์ต่างๆ กับพอร์ทอนุกรมโดยอัตโนมัติและจะมีการแจ้งเตือนให้ผู้อ่านได้รับทราบโดยผ่านโพรซีเยอร์เหตุการณ์ เช่นเดียวกับคอนโทรลทั่วไปของ Visual Basic นั่นเอง ดังนั้นในการเขียนโค้ดเราจึงไม่จำเป็นต้องสร้างโพซีเยอร์ที่ทำหน้าที่คอยตรวจสอบเหตุการณ์ต่างๆ ของพอร์ทอนุกรมซึ่งจะทำให้ง่ายต่อการทำงานเป็นอย่างมากคอนโทรลMSCommจะมีหน้าที่สำหรับการสื่อสารผ่านพอร์ทอนุกรม 3 ประการ ดังต่อไปนี้

- หมุนหมายเลขติดต่อกับโทรศัพท์ปลายทางที่กำหนด
- ตรวจสอบการเข้ามาของข้อมูลยังพอร์ทอนุกรมโดยอัตโนมัติ
- ส่งข้อมูลตามที่กำหนดจากโปรแกรมไปยังพอร์ทอนุกรม

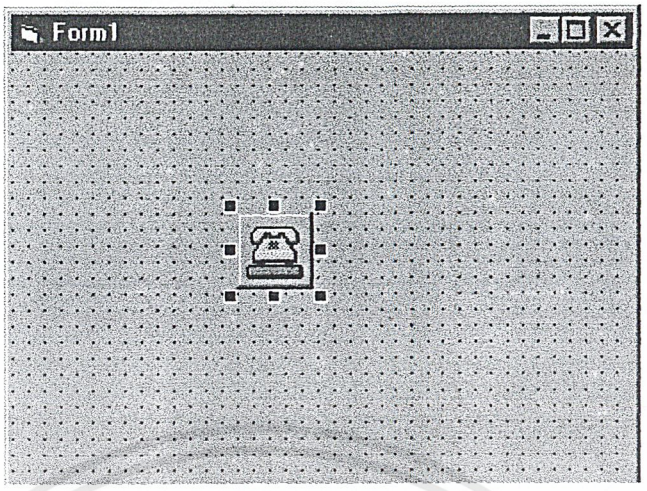
ในความจริงคอนโทรล MSComm ไม่ได้ทำหน้าที่ติดต่อกับพอร์ทอนุกรมโดยตรงแต่มันจะทำหน้าที่เรียกใช้ฟังก์ชันวินโดวส์ API ซึ่งวินโดวส์จะทำการส่งหรือรับข้อมูลผ่านพอร์ทอนุกรมโดยอาศัยไดร์วเวอร์ Comm.drv อีกทอดหนึ่ง ดังนั้นจึงสามารถสรุปสั้นๆ ได้ว่าทุกครั้งที่เรามีการเรียกใช้คอนโทรล Mscomm ก็หมายถึงเรียกใช้ฟังก์ชันวินโดวส์ API ซึ่งจะถูกตีความอีกทอดหนึ่งโดยไดร์เวอร์ Comm.drv จากนั้นก็จะส่งผ่านข้อมูลที่ถูกจัดรูปแบบตามมาตรฐานการสื่อสาร (ทั้งนี้ขึ้นกับอุปกรณ์ที่ต่อเข้ากับพอร์ทอนุกรม) ให้กับดีไวซ์ไดรเวอร์อีกทอดหนึ่งนั่นเอง

การกำหนดคุณสมบัติของคอนโทรล MSComm ในขณะออกแบบเราสามารถกระทำได้อย่างสะดวกโดยการคลิ๊กไปที่ปุ่มของรายการ (Custom) ในหน้าต่างคุณสมบัติซึ่งก็จะปรากฏไดอะล็อกบ็อกซ์ Property Pages เพื่อให้เราได้ปรับแต่งค่าของคุณสมบัติของคอนโทรล MSComm สนับสนุน ซึ่งปุ่มต่างๆ ของไดอะล็อกบ็อกซ์ Property Pages มีความหมายดังนี้

- ปุ่มคำสั่ง OK ยอมรับการแก้ไขคุณสมบัติของคอนโทรล MSComm
- ปุ่มคำสั่ง Cancel ยกเลิกการแก้ไขคุณสมบัติของคอนโทรล MSComm
- ปุ่มคำสั่ง Apply อัปเดตคุณสมบัติที่ถูกแก้ไขของคอนโทรล MSComm
- ปุ่มคำสั่ง Help แสดงผล Help ของคอนโทรล MSComm

สำหรับฟอร์มหนึ่งๆ เราสามารถเพิ่มได้หลายๆ คอนโทรลMSComm ทั้งนี้ขึ้นอยู่กับความต้องการของเราในการติดต่อกับพอร์ทอนุกรมใดบ้าง สำหรับวินโดวส์ 95 และ NT 4.0 เราสามารถติดตั้งพอร์ทอนุกรมได้มากกว่า 4 พอร์ท โดยเราสามารถเพิ่มคอนโทรล MSComm ลงในฟอร์มก็จะปรากฏดังภาพ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 3.11 แสดงคอนโทรล MSComm

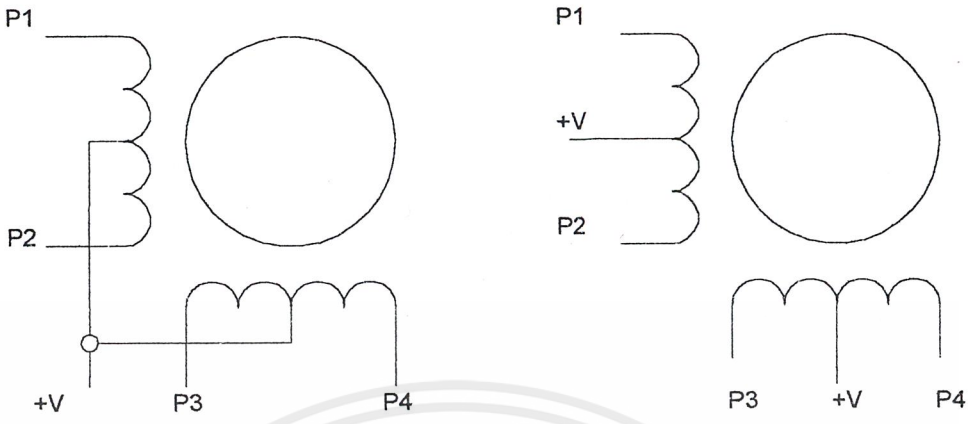
3.5 สเต็ปเปอร์มอเตอร์

สเต็ปเปอร์มอเตอร์เป็นมอเตอร์ที่มีลักษณะการทำงานแตกต่างจากมอเตอร์ทั่วไป เพราะจะต้องป้อนสัญญาณเป็นพลัส์ให้แก่ขดลวดมอเตอร์เป็นจังหวะอย่างเหมาะสม และการหมุนของมอเตอร์ชนิดนี้จะหมุนเป็นจังหวะตามพัลส์ที่ป้อนเข้ามา ไม่หมุนต่อเนื่องเหมือนกับมอเตอร์ธรรมดาทำให้ผู้ควบคุมสามารถเลือกตำแหน่งที่ต้องการให้มอเตอร์หยุดหมุนได้ จังหวะการหมุนของสเต็ปเปอร์มอเตอร์เรียกว่า สเต็ป (step) ความละเอียดของมอเตอร์กำหนดเป็นองศาที่หมุนไปในหนึ่ง สเต็ป หากมอเตอร์มีจำนวนองศาต่อสเต็ปมาก หมายความว่ามอเตอร์ตัวนี้มีความละเอียดของการหมุนต่ำ ยกตัวอย่าง การหมุนควบ 1 รอบเท่ากับ 360 องศา หากมอเตอร์มีสเต็ปการหมุนเท่ากับ 7.5 องศาต่อสเต็ป มอเตอร์นี้ความละเอียดของการหมุนเท่ากับ 48 ตำแหน่ง แต่ถ้าหากว่าสเต็ปการหมุนเท่ากับ 1.8 องศาต่อสเต็ป ความละเอียดของการหมุนเท่ากับ 200 ตำแหน่ง จะเห็นว่ามอเตอร์จะมีความละเอียดสูง ทำให้การนำไปใช้งานที่ต้องการกำหนดตำแหน่งได้ดีกว่า แม่นยำกว่า ผนวกเข้ากับวงจรขับแบบฮาล์ฟสเต็ป ความละเอียดของการหมุนจะเพิ่มขึ้นอีก 2 เท่า ทำให้มีความละเอียดของการหมุนกลายเป็น 400 ตำแหน่ง

ขนาดของสเต็ปเปอร์มอเตอร์ที่มีการผลิตและจำหน่ายในท้องตลาดมีตั้งแต่ขนาดแรงดันต่ำ 3 V ไปถึง 24 V ส่วนขนาดของกระแสมีตั้งแต่ไม่กี่สิบมิลลิแอมป์อันเป็นสเต็ปเปอร์มอเตอร์ตัวเล็ก ไปจนถึงเป็นสิบแอมป์ซึ่งจะเป็นขนาดใหญ่

สเต็ปเปอร์มอเตอร์ได้รับการพัฒนามาอย่างต่อเนื่องจนในปัจจุบันสเต็ปเปอร์มอเตอร์ที่ได้นิยมใช้อย่างแพร่หลายมากที่สุดและหาได้ง่าย คือ สเต็ปเปอร์มอเตอร์แบบยูนิโพล่าร์ (uni-polar stepper motor) มีลักษณะการพันขดลวดของมอเตอร์ดังภาพ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



แบบ 5 สาย 4 เฟส แบบ 6 สาย 4 เฟส

ภาพที่ 3.12 โครงสร้างพื้นฐานของสเต็ปเปอร์มอเตอร์แบบยูนิโพล่าร์ แบบ 5 และ 6 สาย

สเต็ปเปอร์มอเตอร์แบบนี้มีการพันขดลวด 2 ขดบนแต่ละขั้วแม่เหล็กของสเตเตอร์ แต่ละขดลวดแบ่งเป็น 2 เฟส รวมมอเตอร์ทั้งตัวจะมี 4 เฟสคือ เฟส 1,2,3 และ 4 มีการต่อสายออกมาจากขดลวดแต่ละขดเพื่อจ่ายไฟเลี้ยง ทำให้สเต็ปเปอร์มอเตอร์แบบนี้มีทั้งแบบ 5 สาย และ 6 สาย ถ้าเป็นแบบ 5 สาย จะเป็นการนำสายไฟเลี้ยงของขดลวดทั้งสองมาต่อรวมเป็นสายเดียว

3.5.1 การกระตุ้นและควบคุมการหมุนของสเต็ปเปอร์มอเตอร์

การกระตุ้นและควบคุมการหมุนของมอเตอร์ให้เคลื่อนที่ไปแต่ละสเต็ปทำได้โดยจ่ายกำลังไฟฟ้าไปยังขดลวดแต่ละขดบนสเตเตอร์ ซึ่งต้องป้อนเป็นซีเควนเชียลในรูปแบบที่ถูกต้องด้วย สามารถแบ่งได้เป็น 3 รูปแบบคือ แบบฟูลสเต็ป 1 เฟส (full step 1 phase) , แบบฟูลสเต็ป 2 เฟส และ แบบฮาล์ฟสเต็ป (half step)

แบบฟูลสเต็ป 1 เฟสเป็นการกระตุ้นที่มีรูปแบบง่ายที่สุด โดยทำการกระตุ้นขดลวดทีละขดในเวลาหนึ่งไล่เรียงถัดกันไป เช่น เริ่มต้นที่ขด 1,2,3,4 แล้ววนกลับมาขดที่ 1 วนไปเรื่อยๆ หรือเริ่มที่ขดที่ 1 แล้วย้อนไปยังขดที่ 4,3,2 แล้วกลับมายังขดที่ 1 อีกครั้ง ซึ่งทำให้ทิศทางการหมุนสวนกัน ในการกระตุ้นด้วยรูปแบบนี้จึงมีขดลวดเพียงขดเดียวในเวลาหนึ่งที่จะถูกกระตุ้นเท่านั้น วงจรกระตุ้นแบบเวฟจึงมีราคาถูกและหาง่าย ขั้นตอนการทำงานต่างๆ แสดงดังในตาราง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 3.4 รูปแบบการขับสเต็ปเปอร์มอเตอร์แบบฟูลสเต็ป 1 เฟส

สเต็ปที่	เฟสที่1	เฟสที่2	เฟสที่3	เฟสที่4
1	ทำงาน	-	-	-
2	-	ทำงาน	-	-
3	-	-	ทำงาน	-
4	-	-	-	ทำงาน

แบบฟูลสเต็ป 2 เฟสเป็นการกระตุ้นซึ่งคล้ายกับแบบ 1 เฟส แต่การกระตุ้นแบบนี้จะทำการกระตุ้นโดยจ่ายกำลังไฟฟ้าให้ไปที่ขด 2 ขด ที่อยู่ใกล้กันในเวลาเดียวกันและเรียงถัดกันไป เช่นเดียวกับแบบเวฟ ดังตัวอย่าง ขดลวดชุดแรกที่ถูกกระตุ้นจะเป็นขดที่ 1 และ 2 ตามด้วยการกระตุ้นขดที่ 2 และ 3 ต่อไปเป็นขดที่ 3 และ 4 ถัดไปเป็นขดที่ 4 และ 1 แล้วกลับมาที่ขดที่ 1 และ 2 วนไปตามลำดับเช่นนี้ หรือเริ่มขดที่ 1 และ 4 ทิศทางหารหมุนจะสวนทางกัน การกระตุ้นสเต็ปเปอร์มอเตอร์แบบนี้สามารถเพิ่มแรงบิดได้มากกว่าแบบฟลูสเต็ป 1 เฟส โรเตอร์จะเคลื่อนที่ด้วยแรงดึงอย่างเต็มแรง จาก 2 ขด ขดลวดที่ถูกกระตุ้นพร้อมกัน และต่อไปด้วยแรงดึงจากอีก 2 ขดลวดถัดไป สำหรับข้อเสียคือการกระตุ้นต้องใช้กำลังไฟฟ้ามากขึ้นขั้นตอนการทำงานต่างๆ แสดงในตาราง

ตารางที่ 3.5 รูปแบบการขับสเต็ปเปอร์มอเตอร์แบบฟูลสเต็ป 2 เฟส

สเต็ปที่	เฟสที่ 1	เฟสที่ 2	เฟสที่ 3	เฟสที 4
1	ทำงาน	ทำงาน	-	-
2	-	ทำงาน	ทำงาน	-
3	-	-	ทำงาน	ทำงาน
4	ทำงาน	-	-	ทำงาน

แบบฮาล์ฟสเต็ปเป็นรูปแบบที่ผสมผสานระหว่างการกระตุ้นแบบฟูลสเต็ป 1 และ 2 เฟส เพื่อเพิ่มจำนวนของสเต็ปต่อรอบอีกเท่าตัวหนึ่ง ในระบบนี้จะทำการกระตุ้นขดลวดเรียงกันไปเป็นลำดับดังนี้ เริ่มจากขดลวดที่ 1,1 และ 2,2,2 และ 3,3,3 และ 4,4,4 และ 1 และวกหลับมายังขดลวดที่ 1 แรงบิดที่ได้จาการกระตุ้นแบบนี้จะเพิ่มมากขึ้นอีก เพราะช่วงสเต็ปมีระยะสั้นลง แต่ละสเต็ปเกิดแรงดึงจากขดลวด 2 ขดที่ถูกกระตุ้นพร้อมกัน ความถูกต้องของตำแหน่งมีเพิ่มมากขึ้นแต่ต้องพึ่งระวังไว้อีกประการหนึ่งว่าเมื่อกระตุ้นให้ทำงานในรูปแบบนี้จะต้องใช้การหมุนถึง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2 สเต็ป จึงจะได้เท่ากับระยะทางเท่ากับ 1 สเต็ปเต็มของการควบคุมใน 2 แบบแรก สำหรับแหล่งจ่ายกำลังไฟฟ้าต้องใช้ขนาดเท่ากับแบบที่ 2 เฟสเป็นอย่างน้อย จึงจะเพียงพอ ขั้นตอนการทำงานต่างๆ แสดงในตาราง

ตารางที่ 3.6 รูปแบบการขับสเต็ปเปอร์มอเตอร์แบบฮาล์ฟสเต็ป

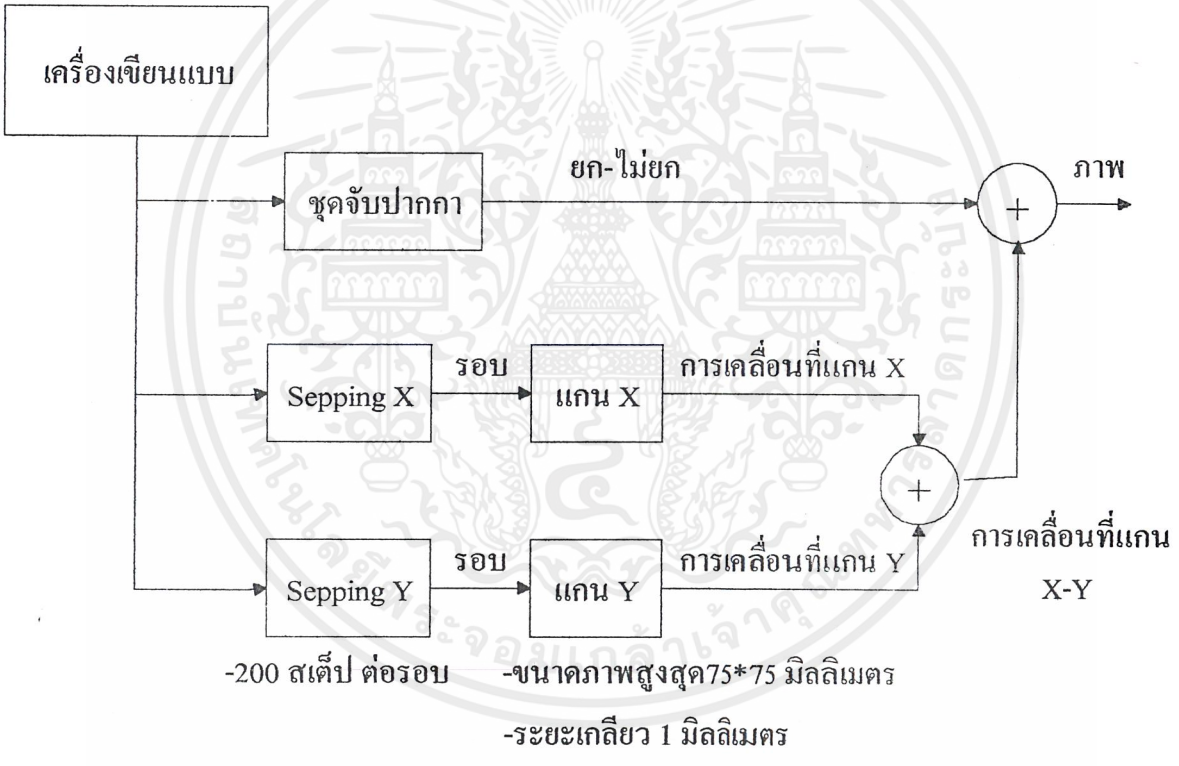
สเต็ปที่	เฟสที่ 1	เฟสที่ 2	เฟสที่ 3	เฟสที่ 4
1	ทำงาน	-	-	-
2	ทำงาน	ทำงาน	-	-
3	-	ทำงาน	-	-
4	-	ทำงาน	ทำงาน	-
5	-	-	ทำงาน	-
6	-	-	ทำงาน	ทำงาน
7	-	-	-	ทำงาน
8	ทำงาน	-	-	ทำงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4
การออกแบบและสร้างเครื่องเขียนแบบ

4.1 การออกแบบและสร้างส่วนการเคลื่อนที่

จะเป็นส่วนที่ทำงานตามการสั่งงานของ Board MCS-51 ระบบการทำงานของเครื่องจะเริ่มจากการสั่งงานของ MCS-51 ให้มอเตอร์เกิดการหมุนเป็นรอบแล้วทำให้เกิดการเคลื่อนที่ของแต่ละแกน เมื่อทั้งสองแกนต่างก็หมุนการเคลื่อนที่ของชุดจับปากกาก็จะเป็นการเคลื่อนที่ในแนวแกน X-Y รวมกับการสั่งให้ยกหรือไม่ยกปากกา สิ่งที่ได้ก็จะเป็นรูปภาพที่ต้องการ ดังภาพที่ 4.1

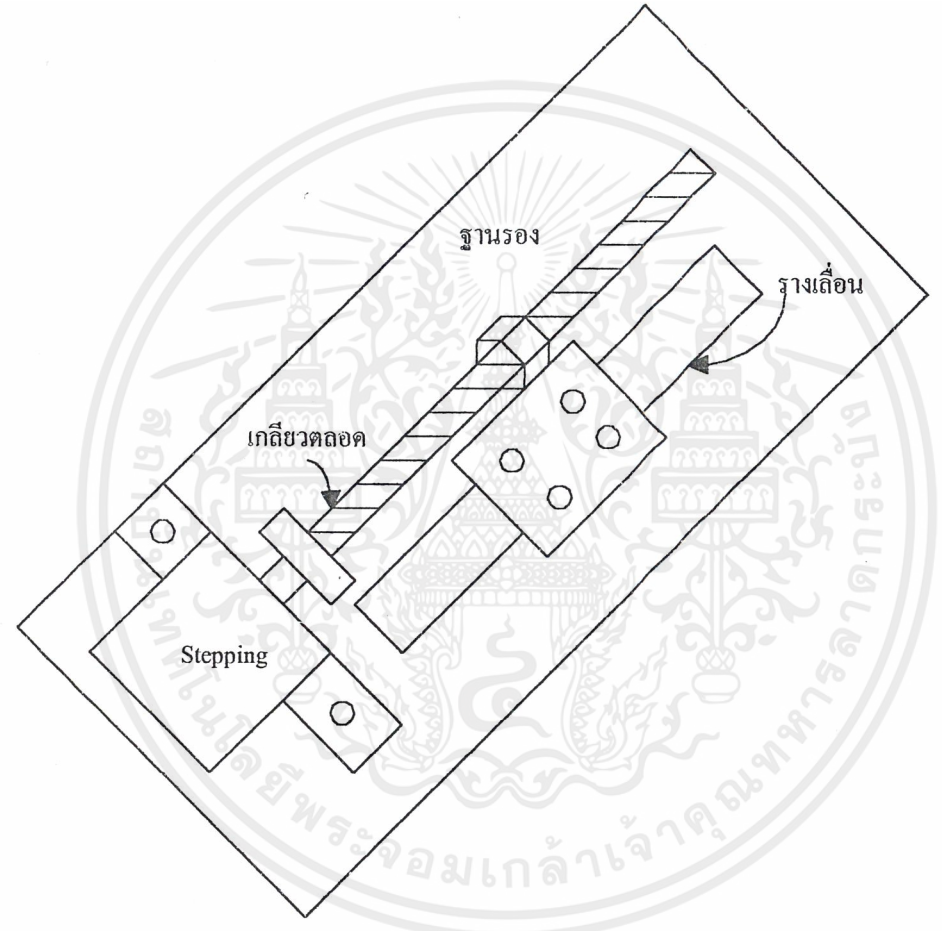


ภาพที่ 4.1 ระบบการทำงานของส่วนเคลื่อนที่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ส่วนการเคลื่อนที่จะเป็นส่วนที่ใช้ในการลากรูปที่เขียนโดยโปรแกรมคอมพิวเตอร์ โดยจะมีลักษณะเหมือนกับ X-Y Plotter ที่ประกอบด้วยส่วนประกอบหลัก 5 ส่วนที่สำคัญ คือ

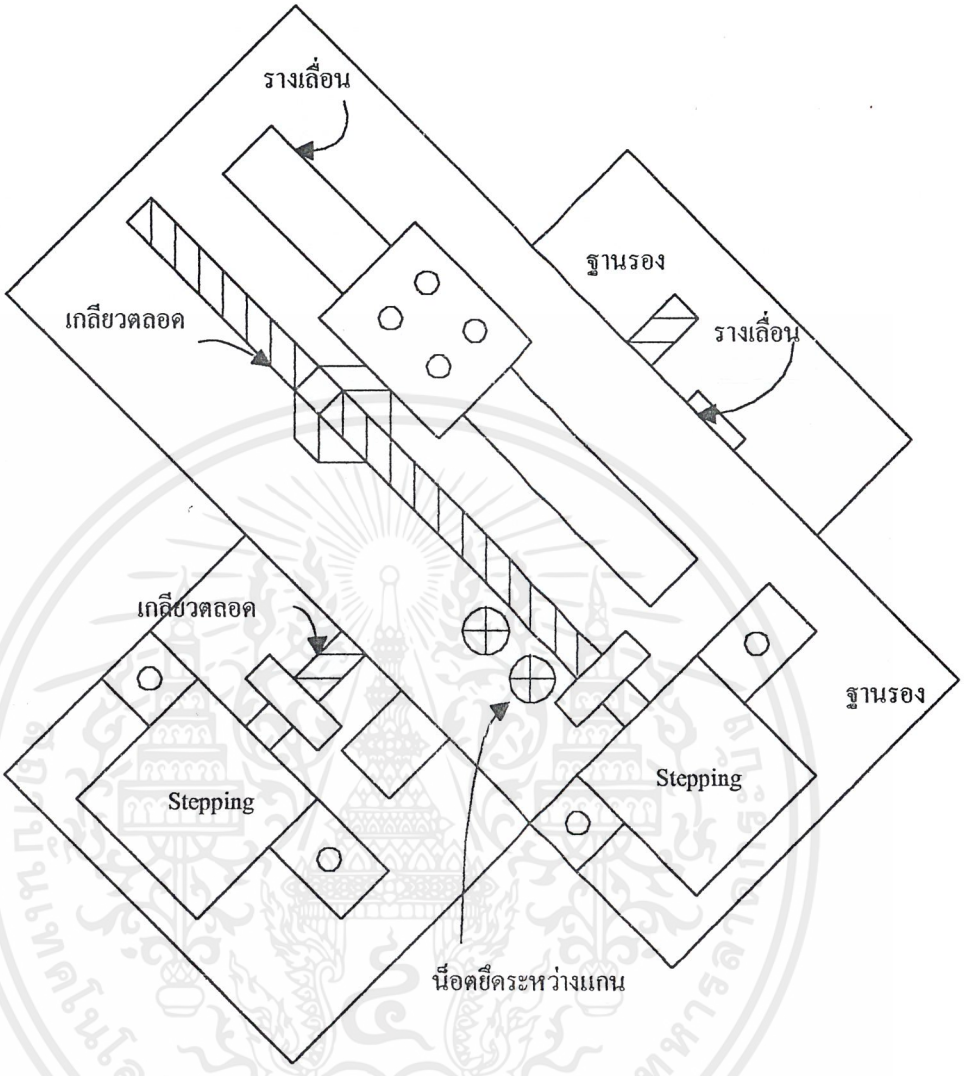
1. Stepping Motor
2. เกลียวยาวตลอด
3. รางเลื่อน
4. ฐานรอง
5. ชุดยกปากกา



ภาพที่ 4.2 ส่วนเคลื่อนที่ 1 แกน

นำส่วนประกอบหลักทั้ง 4 มาประกอบเป็น 1 แกน จะได้ดังรูปที่ 4.2 แล้วนำอีกแกนมาวางลงอีกแกนแล้วยึดไว้ด้วยน็อตจะได้ดังภาพที่ 4.3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

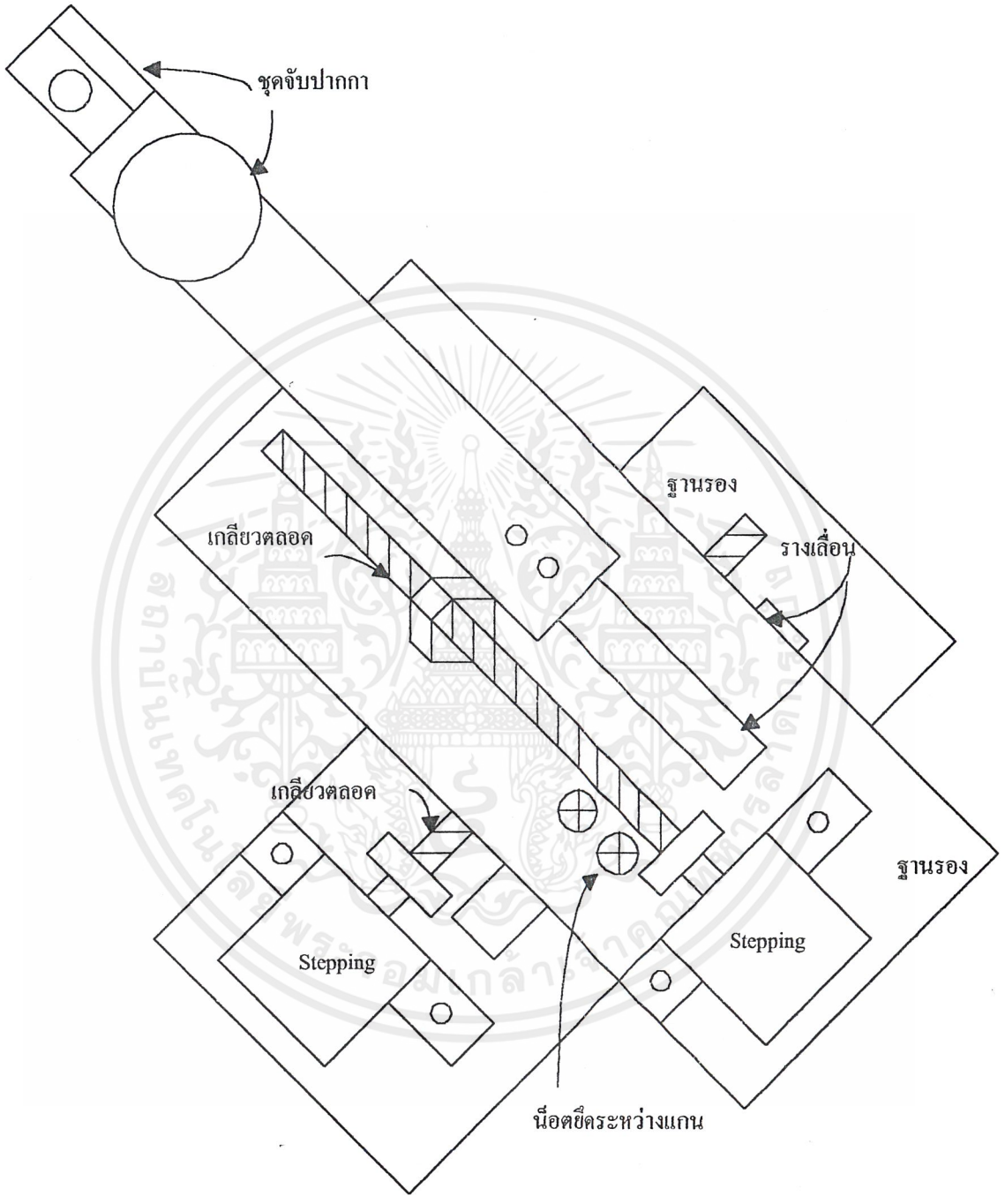


ภาพที่ 4.3 ส่วนเคลื่อนที่ 2 แกน

เกลียวยาวตลอดมีระยะ pitch เท่ากับ 1 มิลลิเมตร Stepping Motor มีสเต็ป เท่ากับ 200 สเต็ปต่อรอบ เมื่อต้องการเคลื่อนที่ไป 1 มิลลิเมตร ก็จะต้องปล่อยสัญญาณสเต็ปมา 200 สเต็ป เมื่อต้องการเคลื่อนที่ 1 เซนติเมตร ก็จะต้องปล่อยสัญญาณสเต็ป 2000 สเต็ป

ชุดยกปากกาจะเป็นชุดที่ทำหน้าที่ในการลากเส้นหรือไม่ลากของการเขียนแบบซึ่งจะถูกใส่ไว้บนแกนที่อยู่ด้านบน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



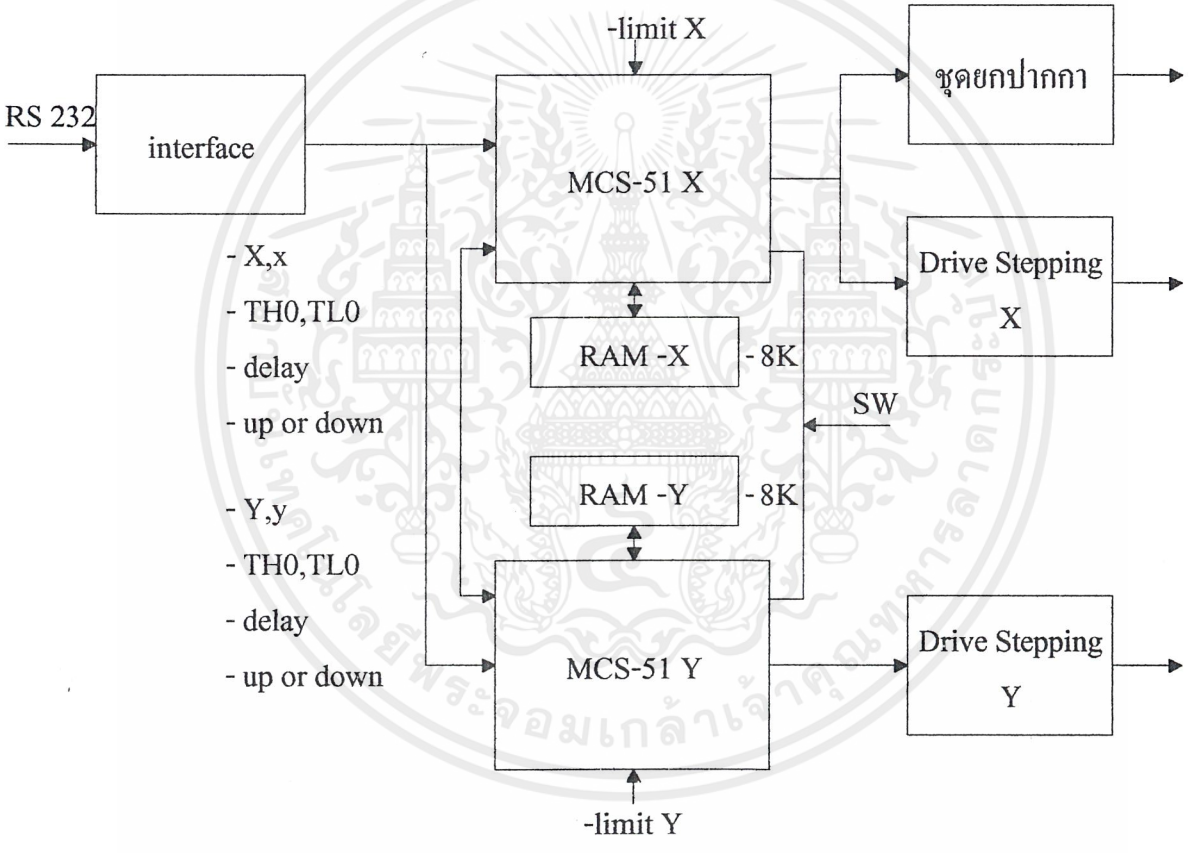
ภาพที่ 4.4 เครื่องเขียนแบบขนาดเล็ก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2 การออกแบบและสร้างส่วนของ MCS-51

4.2.1 บอร์ด MCS-51

บอร์ดที่ใช้จะเป็นตัวที่ใช้สำหรับรับข้อมูลจากคอมพิวเตอร์แล้วนำข้อมูลมาประมวนผล แล้วส่งค่าสเต็ปให้กับมอเตอร์ทำงานตามข้อมูลที่ได้รับ โดยจะมี MCS-51 สองตัวแต่ละตัว จะควบคุมแยกกันคนละแกน ข้อมูลที่ส่งมาจากคอมพิวเตอร์ก็จะเป็นข้อมูลฐานสิบหกในการตั้งค่า Register ก็จะมีทิศทางของการหมุน จำนวนสเต็ปของการหมุน ระยะหน่วงเวลา และการยกหรือไม่ยกปากกา ดังระบบการทำงานดังภาพ

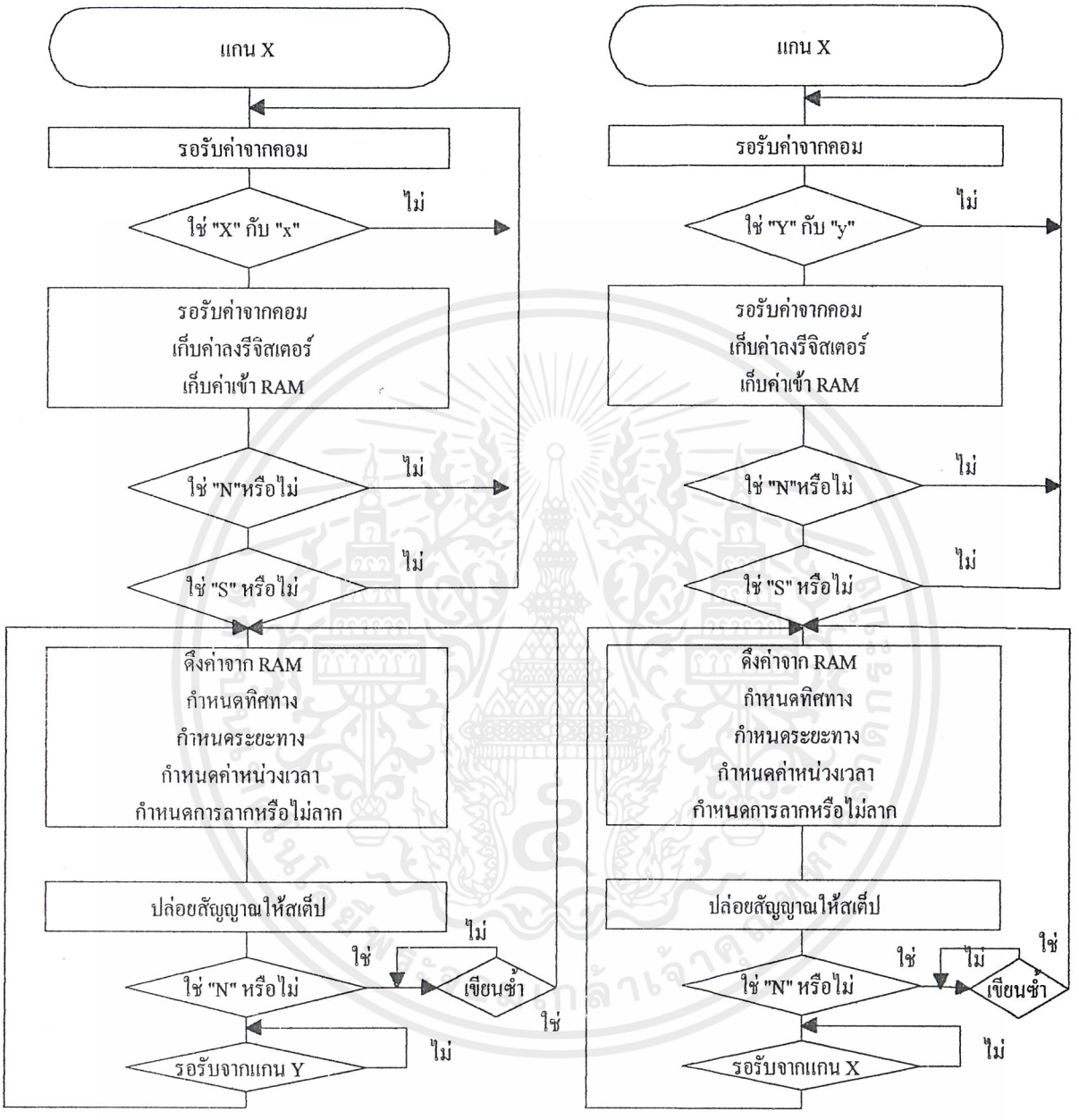


ภาพที่ 4.5 ระบบการทำงานของ MCS-51

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2.2 โปรแกรม MCS-51

การทำงานของโปรแกรม MCS-51 มีดังนี้

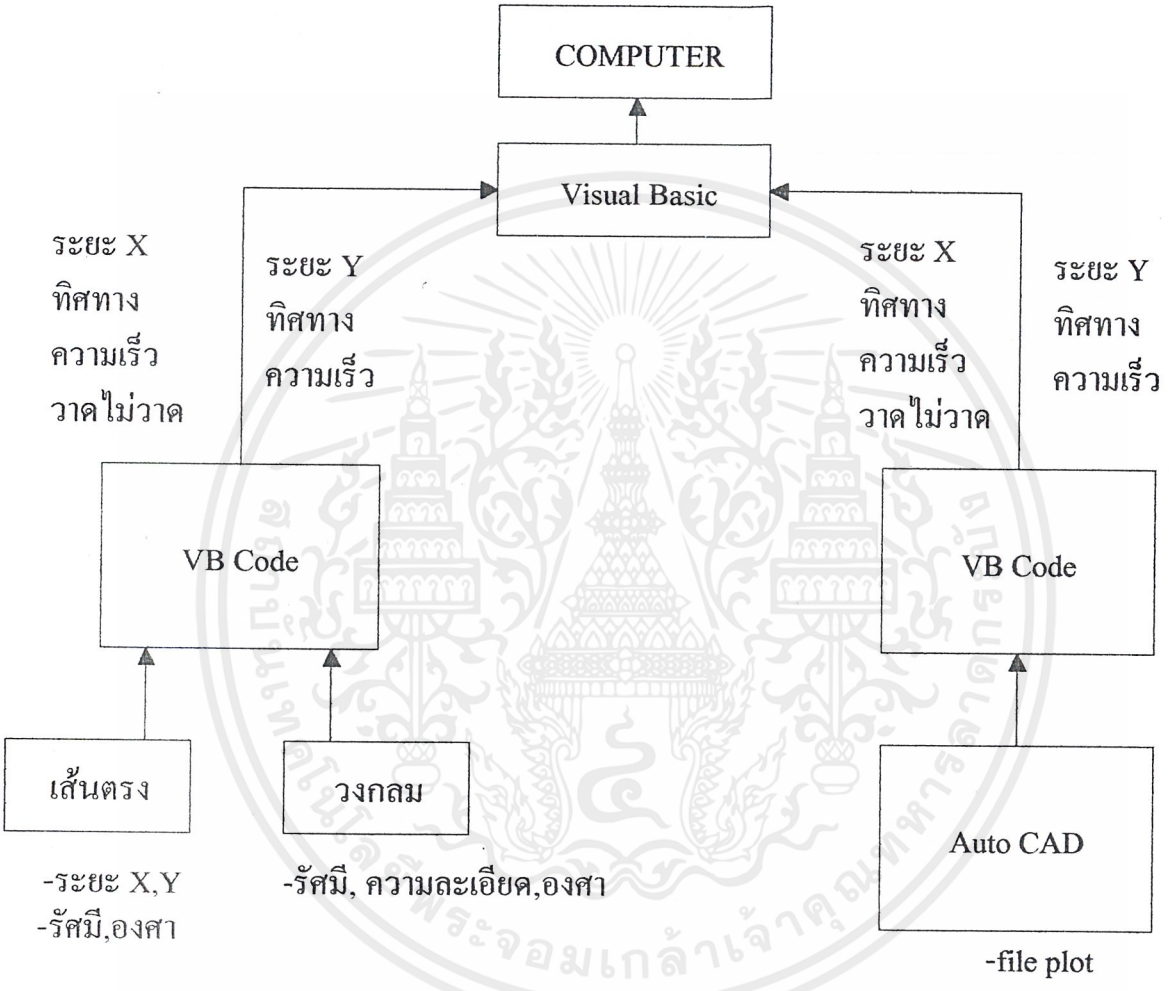


ภาพที่ 4.6 Flow chart โปรแกรมการทำงานของ MCS-51

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3 โปรแกรมคอมพิวเตอร์

โปรแกรมคอมพิวเตอร์ก็จะเป็นตัวกำหนดทางเดินของการเคลื่อนที่โดยจะแยกเป็นโปรแกรม Auto CAD และ โปรแกรม Visual Basic ทั้งสองโปรแกรมนี้สามารถที่จะสร้างทางเดินให้กับเครื่องเขียนแบบขนาดเล็กนี้ได้ ระบบการทำงานโดยรวมก็เป็นดังภาพ

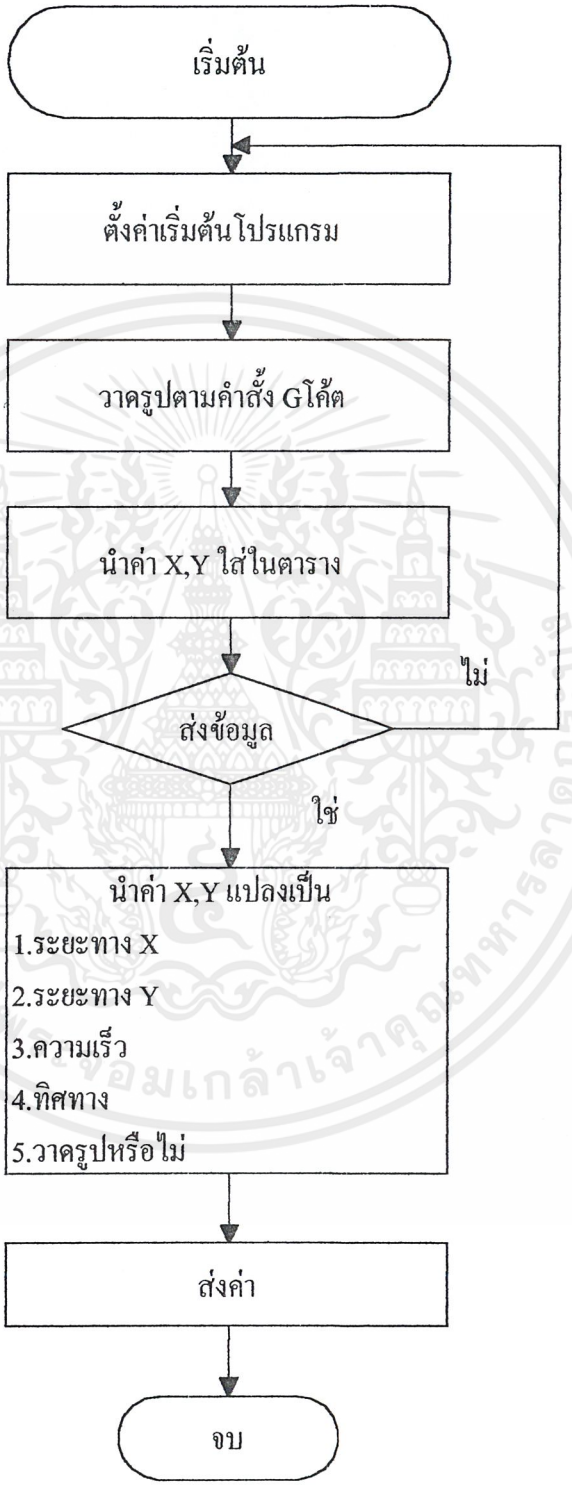


ภาพที่ 4.7 ระบบการทำงานโดยรวมของโปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3.1 โปรแกรม Visual Basic

โปรแกรม Visual Basic จะเป็นตัวที่ใช้ในการเขียนแบบที่จอคอมพิวเตอร์แล้วจะเปลี่ยนเป็นระยะทาง ทิศทาง ระยะเวลาหน่วง การยกปากกา แล้วส่งลงบอร์ด MCS-51 จะมีการทำงานดังนี้



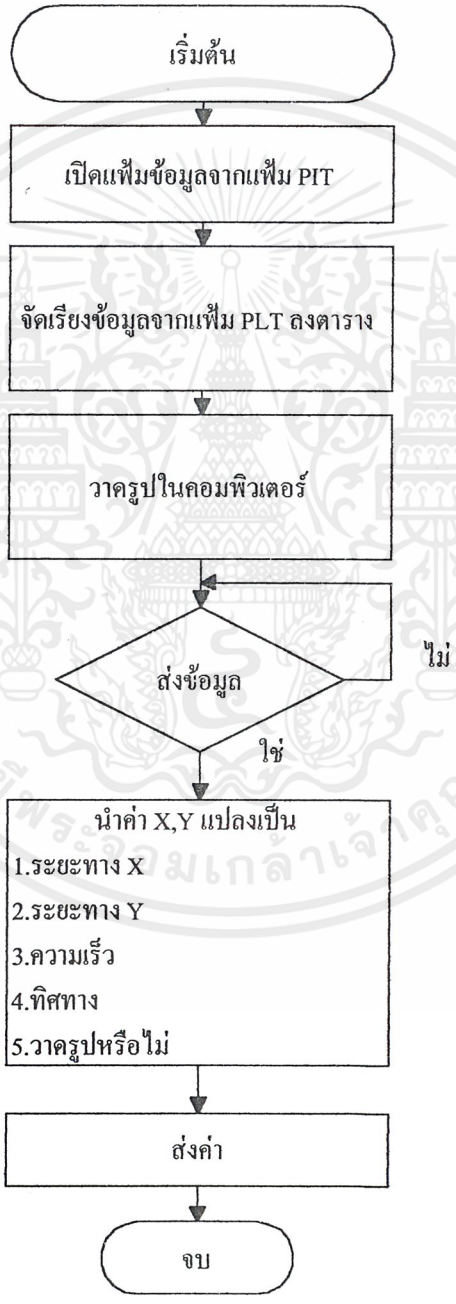
ภาพที่ 4.8 Flow chart การทำงานของโปรแกรม Visual Basic

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3.2 โปรแกรม AutoCAD R14

โปรแกรม AutoCAD R14 จะเป็นโปรแกรมที่ใช้ในการเขียนแบบ ซึ่งโปรแกรมนี้สามารถที่จะเปลี่ยนข้อมูลที่เป็นภาพมาเป็นข้อมูลที่สามารถใช้ในโปรแกรมอื่นได้จะมีลักษณะเป็น Text File ในโครงงานนี้จะใช้ แฟ้มนามสกุล PLT ตัวอย่างเช่น รูปสี่เหลี่ยมจัตุรัส 75.3 มิลลิเมตร จะได้รูปแบบของ file plot คือ

□.(;□.I81;;17:□.N;19:IN;SC;PU;PU;SP1;LT;VS36;PU;PA0,0;PD;PA753,0;PA753,753;PA0,753;PA0,0;PU;PA0,0;SP;



ภาพที่ 4.9 Flow chart การทำงานของ โปรแกรม AutoCAD

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปแบบการจัดเรียงข้อมูลของไฟล์แบ่งได้เป็น 3 ส่วน คือ

- ส่วนข้อมูลทั่วไป (Header Section)
- ส่วนข้อมูลในการเขียน (Block Section)
- ส่วนการจบ (EOF Section)

ส่วนข้อมูลทั่วไปและส่วนการจบจะไม่ถูกทำมาใช้จะใช้เฉพาะส่วนข้อมูลการเขียนซึ่งจะมีตัวอักษรและตัวเลขในการแสดงความหมาย จะมีลักษณะดังนี้

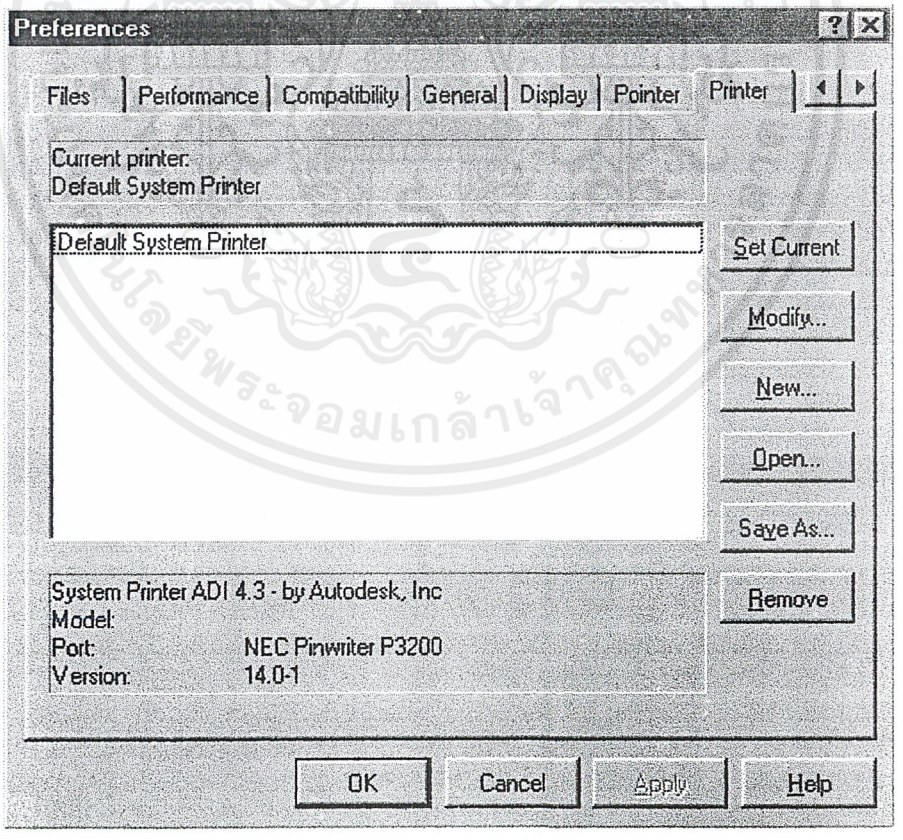
PU จะหมายถึง การยกปากกาขึ้น (Pan Up)

PA0,0 จะหมายถึง พิกัดที่ต้องการจะเขียน

PD จะหมายถึง การยกปากกาลง (Pan Down)

4.3.3 วิธีการตั้งเครื่อง PLOT ในโปรแกรม AutoCAD R14

1. เปิดโปรแกรม AutoCAD R14
2. เลือกที่ TOOLS →Preferences จะได้ดังภาพที่ 4.10



ภาพที่ 4.10 การติดตั้งเครื่อง PLOT ในโปรแกรม AutoCAD R14

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. เลือก Printer →New → Hewlett-Packard(HP-GL)ADI 4-2 by Autodesk ดังภาพ

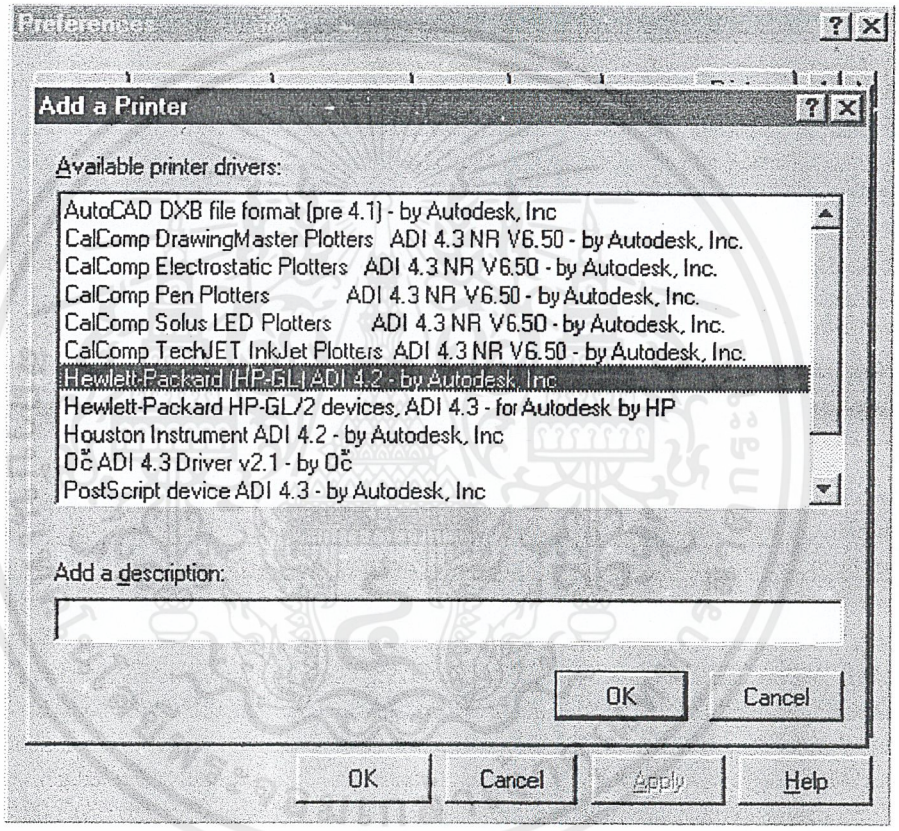
4. เลือก 1

5. เลือก <S>srial port(local)

6. เลือก <30>

7. เลือก <com1> หรือ <com2>

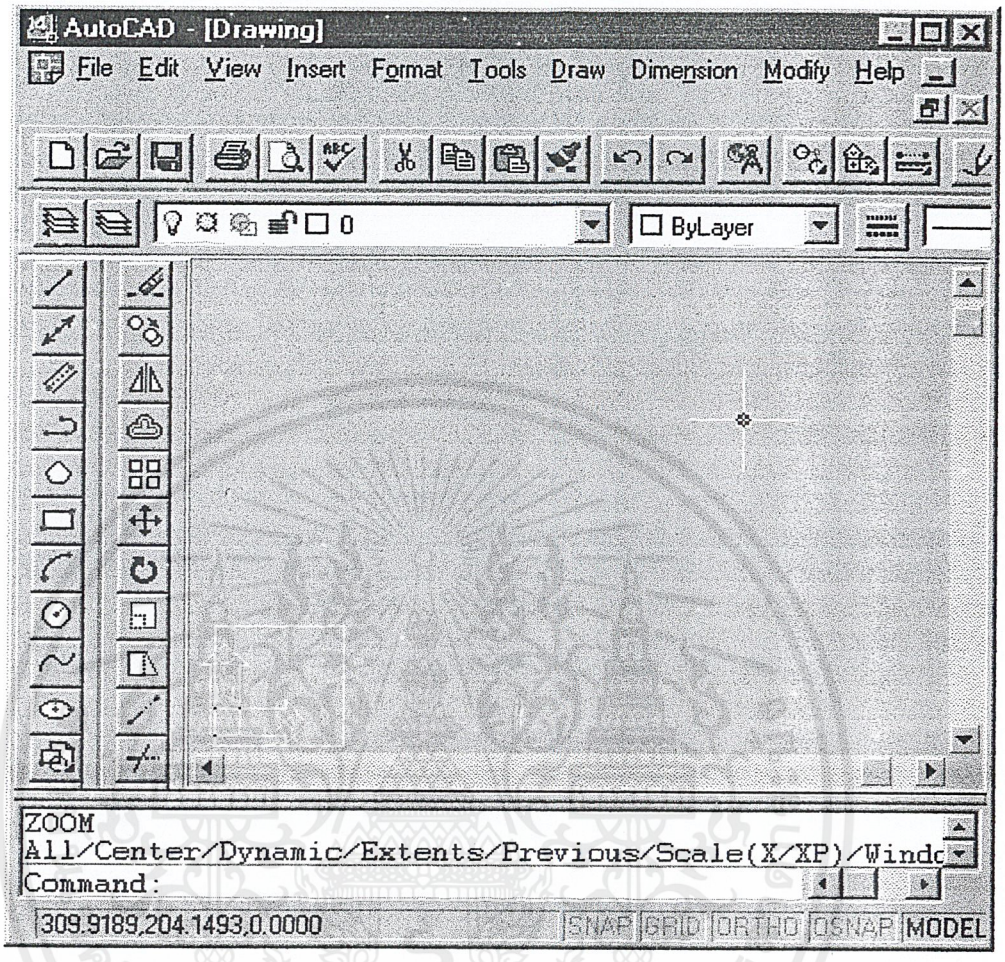
8. เลือก <n>



ภาพที่ 4.11 การเลือกเครื่อง PLOT

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3.4 วิธีการเริ่มต้นเขียนแบบใน โปรแกรม Auto CAD



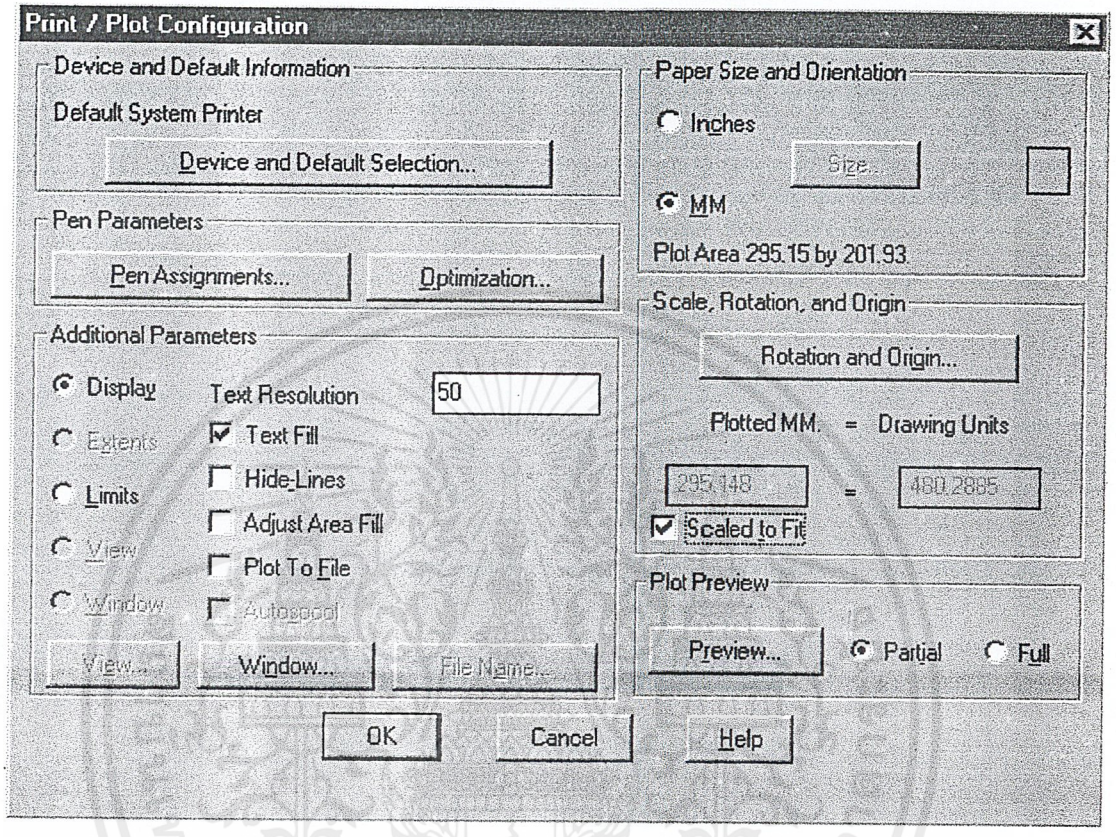
ภาพที่ 4.12 กรอบสี่เหลี่ยมแสดงถึงพื้นที่ที่วาดภาพได้

1. เปิดโปรแกรมเลือก Metric
2. พิมพ์ใน Command: rectangle ↵ ใส่ค่า 0,0 ↵ ใส่ 75,75 ↵ จะปรากฏกรอบสี่เหลี่ยมที่แสดงถึงพื้นที่ที่สามารถเขียนแบบได้ โดยจะต้องไม่มีส่วนใดอยู่นอกกรอบภาพนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3.5 วิธีการตั้งค่า File.PLT ของโปรแกรม AutoCAD (หลังจากเขียนภาพ)

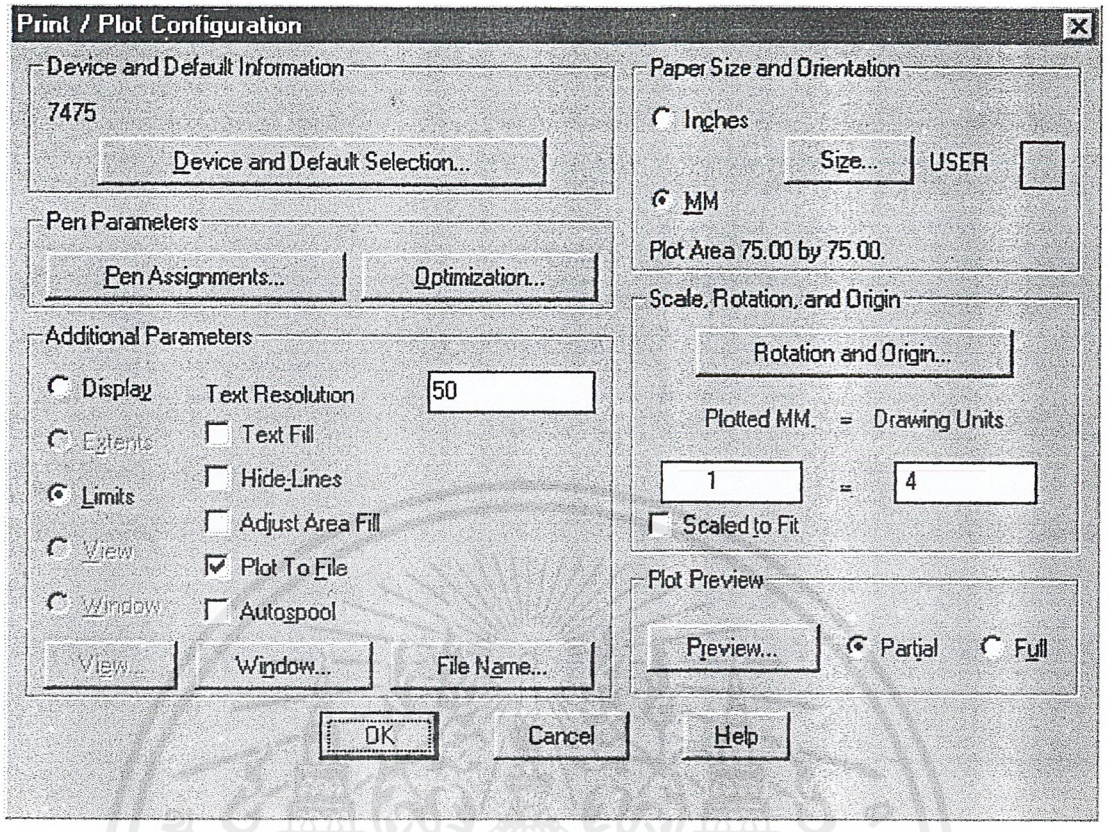
1. เลือก File → Print จะได้ดังภาพที่ 4.13



ภาพที่ 4.13 แสดง File → Print

2. เลือก Device and Default Selection เลือก 7475
3. ที่ Additional Parameters เลือก Limits เลือก Plot to File
4. Paper Size and Orientation เลือก MM, ขนาด USER 75,75
5. Scale ,Rotation and Origin 1=4 จะได้ดังภาพที่ 4.14

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 4.14 แสดงการตั้งค่าใน File.PLT

4.4 การคำนวณหาค่าระยะเวลาการหน่วงของ Stepping Motor

เมื่อต้องการให้การเคลื่อนที่ของแกน X และ แกน Y มีการเคลื่อนที่แตกต่างกันก็จะต้องมีการคำนวณหาค่าเวลาการหน่วง โดยจะต้องขึ้นอยู่กับ

- ฐานเวลาของ MCS-51
- ระยะทางของแกน X และแกน Y
- ความเร็วสูงสุดที่ Stepping Motor เคลื่อนที่ได้
- จำนวนสเต็ปต่อรอบ

จากความสัมพันธ์เหล่านี้นำมาเขียนเป็นสมการได้ว่า

(ระยะทาง*200) * (ความเร็วสูงสุดที่ทำได้) * ฐานเวลาของ MCS-51= เวลาที่ใช้ (1)

การหาค่าความเร็วสูงสุดของมอเตอร์ได้จากการทดลอง ได้เท่ากับ FA00H จำนวน แมชีนไซเคิล ของ MCS-51 เท่ากับ 10000H-FA00H =600H (1536)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตัวอย่างเช่น ต้องการเขียนไปที่ตำแหน่ง X=10 mm , Y=1 mm

ระยะการหน่วงของแกน X =FA00H แทนค่าใน (1)

(10 * 200) * (1000H-FA00H) * (12/11.0592M) = 3.33333333 วินาที

ต้องการหาค่าระยะเวลาการหน่วงของแกน Y จะได้ดังนี้

(1*200) * (10000H-DelayY) * (12/11.0592M) = 3.33333333 วินาที

DelayY=10000H-(3.33333333 * 11.0592M/(1 * 200 * 12)) =CA00H

4.4.1 การหาค่ามุมที่เอียงที่แคบที่สุดที่สามารถทำได้

จากตัวอย่างจะได้ ค่าการหน่วงของ X=F900H,Y=CA00H

ค่าการหน่วงที่ต่ำสุดของ MCS-51 คือ 0001H

(Y*200) * (10000H-0001H) * (12/11.0592M) = 3.3333333 วินาที

Y= 3.3333333 / ((200) * (10000H-FFFFH) * (12/11.0592M)) =0.23437mm

X= 10mm

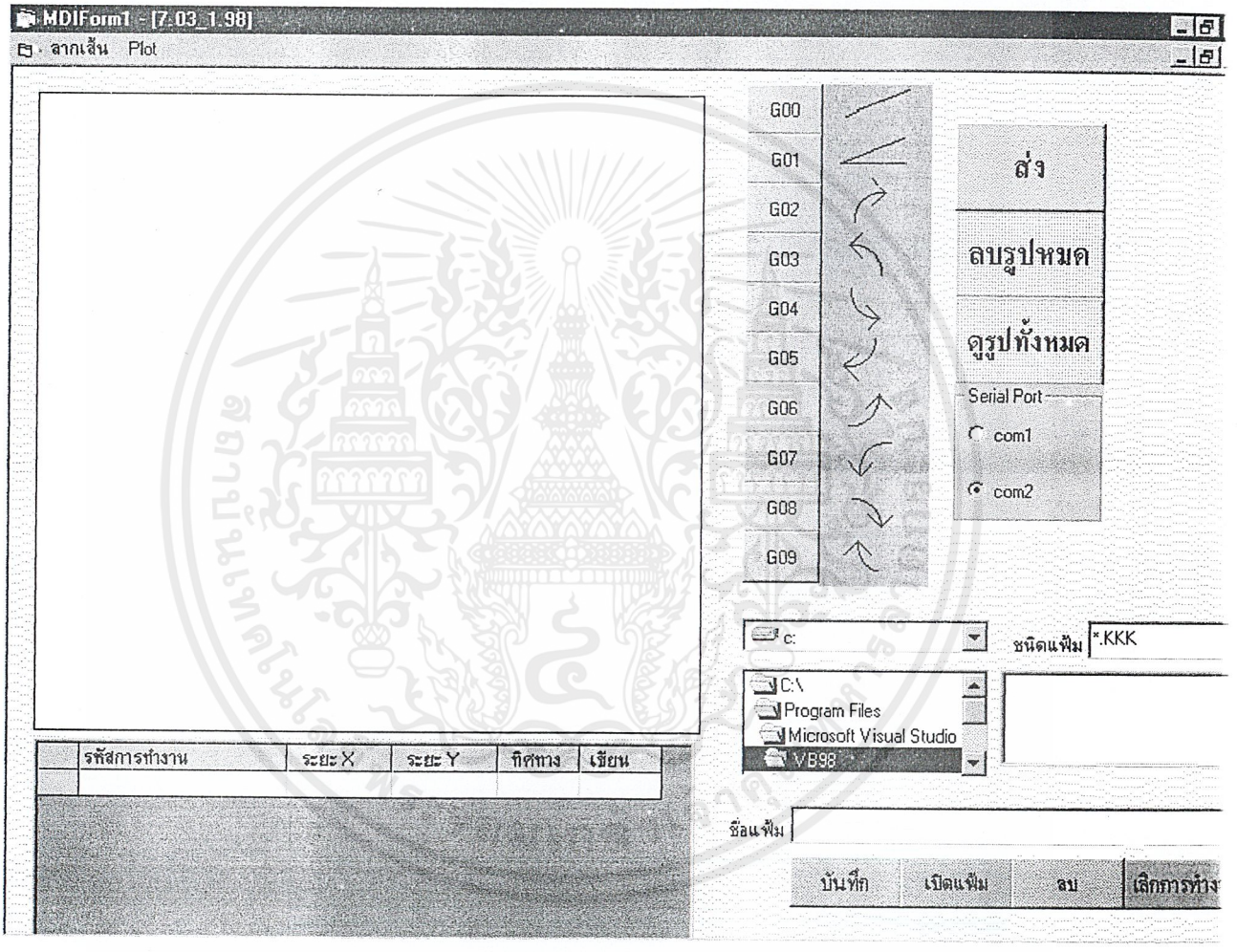
จะได้ว่า มุมที่แคบที่สุดคือ $\tan^{-1}(Y/X)$=1.342 องศา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

ผลการทดลอง

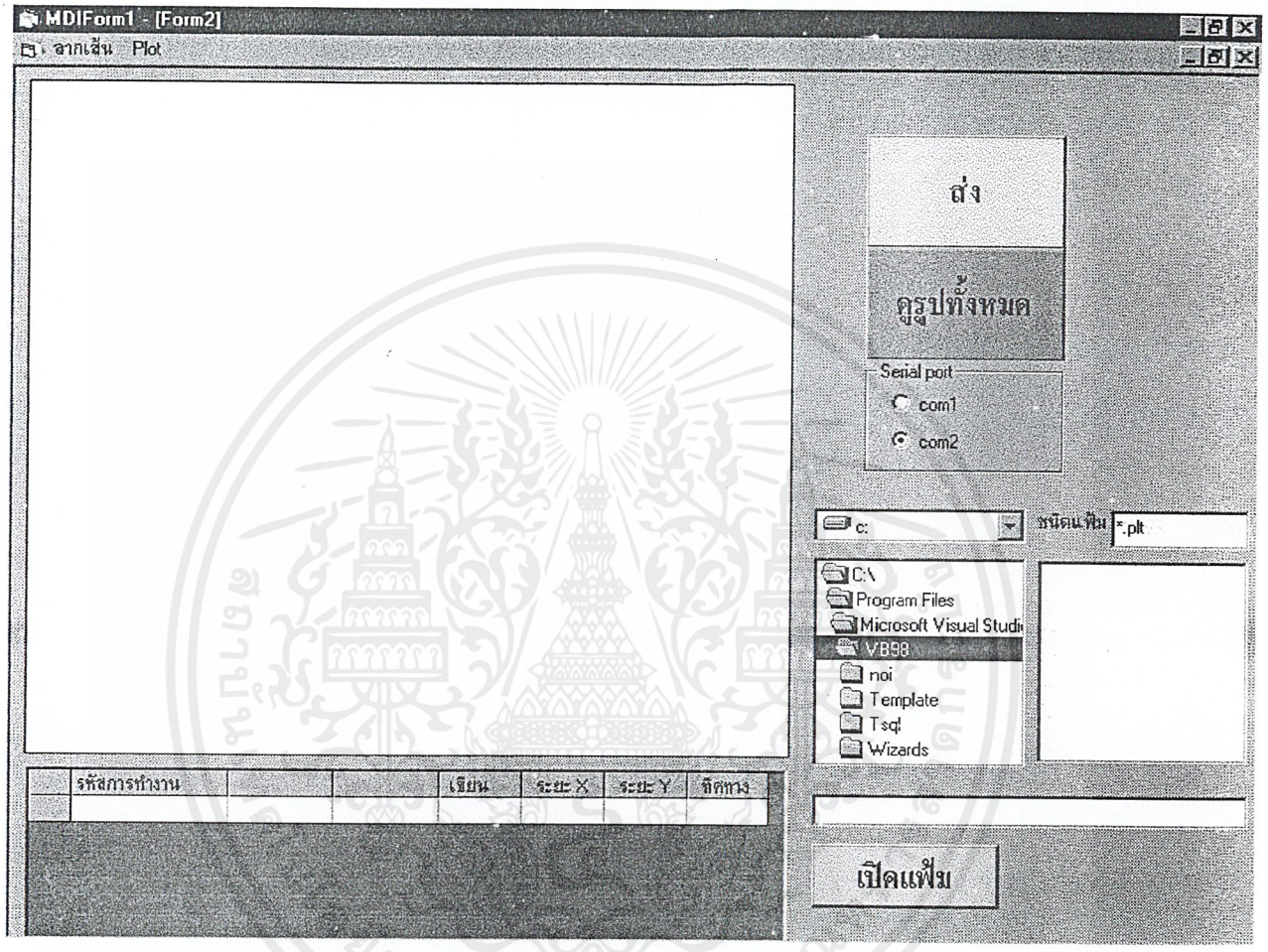
5.1 ภาพหน้าต่างของโปรแกรม Visual Basic



ภาพที่ 5.1 แบบฟอร์มโปรแกรมการเขียนรูปจาก Visual Basic

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.2 ภาพหน้าต่างของโปรแกรม Visual Basic ใช้เปิด file plot

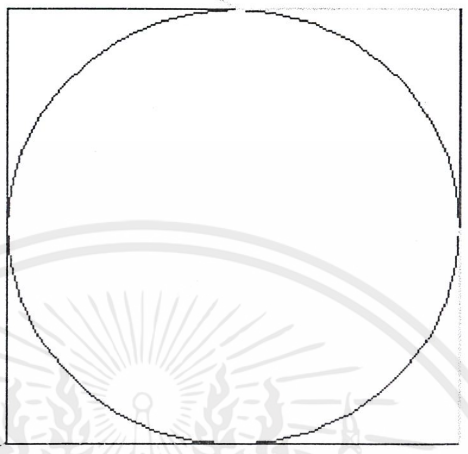


ภาพที่ 5.2 แบบฟอร์มโปรแกรมเปิดภาพจาก file plot ของ Auto CAD R14 โดย Visual Basic

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.3 การทดลอง

5.3.1 การทดลองเขียนภาพโดยโปรแกรม Visual Basic ภาพที่ 1



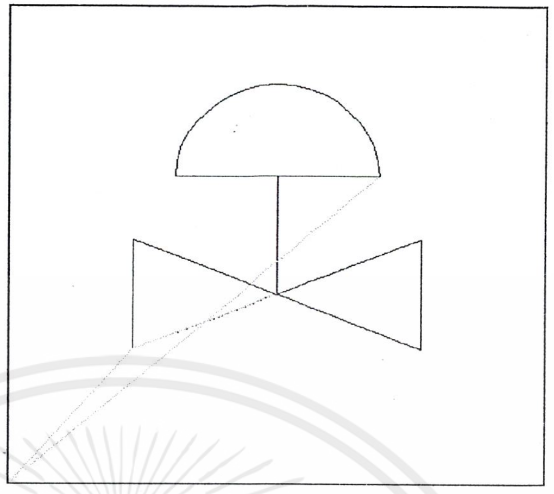
ภาพที่ 5.3 ภาพวาดจากโปรแกรม Visual Basic

ผลที่ได้จากการวาดของเครื่องเขียนแบบ

ภาพที่ 5.4 ภาพที่ได้จากเครื่องเขียนเขียนแบบภาพที่ 1

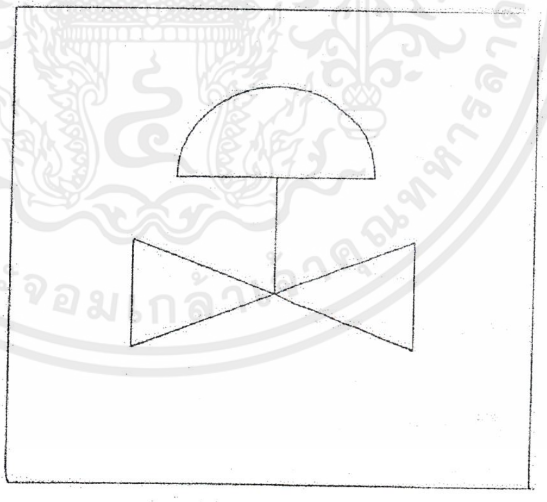
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.3.2 การทดลองเขียนภาพโดยโปรแกรม Auto CAD R14 ภาพที่ 1



ภาพที่ 5.5 ภาพวาดจากโปรแกรม Auto CAD R14

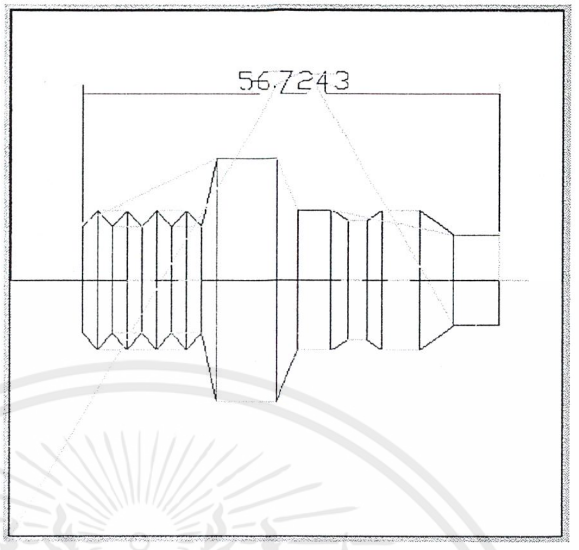
ผลที่ได้จากการวาดของเครื่องเขียนแบบ



ภาพที่ 5.6 ภาพที่ได้จากเครื่องเขียนเขียนแบบภาพที่ 2

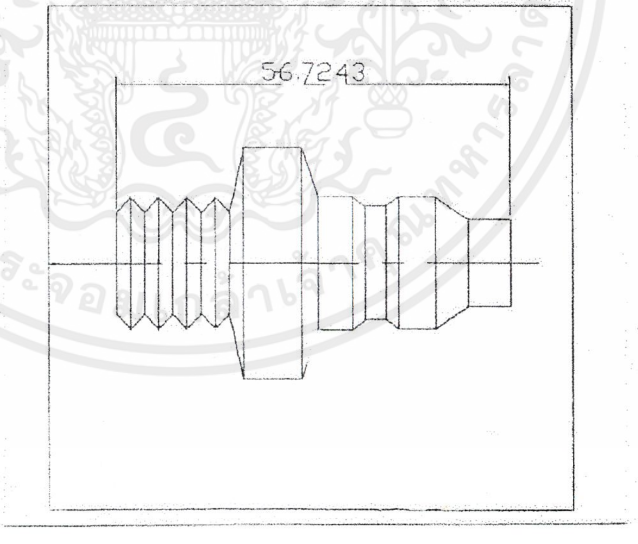
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.3.3 การทดลองเขียนภาพโดยโปรแกรม Auto CAD R14 ภาพที่ 2



ภาพที่ 5.7 ภาพวาดจากโปรแกรม Auto CAD R14 เปิดจาก Visual Basic

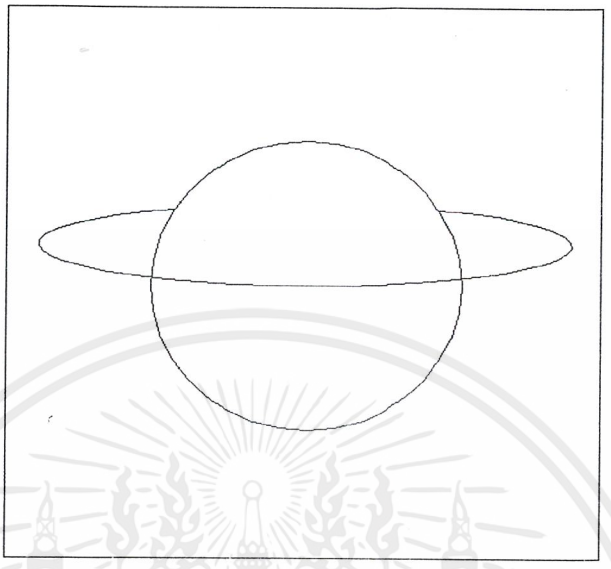
ผลที่ได้จากการวาดของเครื่องเขียนแบบ



ภาพที่ 5.8 ภาพที่ได้จากเครื่องเขียนเขียนแบบ ภาพที่ 3

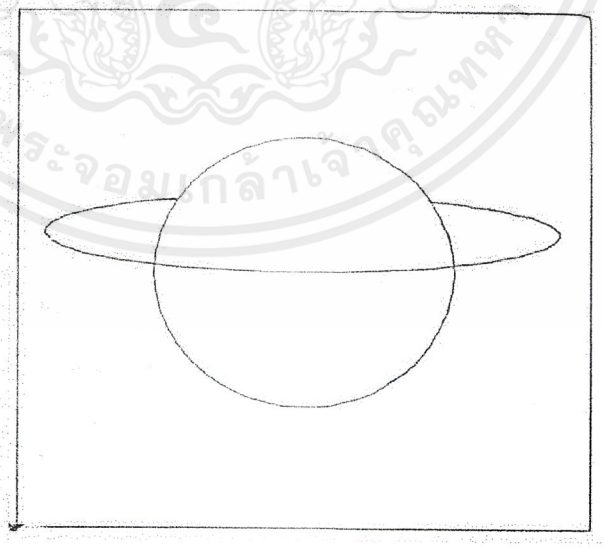
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.3.4 การทดลองเขียนภาพโดยโปรแกรม Auto CAD R14 ภาพที่ 3



ภาพที่ 5.9 ภาพวาดจากโปรแกรม Auto CAD R14

ผลที่ได้จากการวาดด้วยเครื่องเขียนแบบ



ภาพที่ 5.10 ภาพที่ได้จากเครื่องเขียนเขียนแบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 6
สรุปผลและข้อจำกัดของโครงงาน

6.1 สรุปผล

รูปภาพที่เกิดขึ้นจะเกิดจากการเคลื่อนที่ของแกน X แกน Y และ ชุดควบคุมการยกปากกา ซึ่งถูกควบคุมโดย Board MCS-51 เพื่อให้ได้ระยะทาง ความเร็ว การยกปากกา จากข้อมูลที่ส่งมาโดยคอมพิวเตอร์ ข้อมูลนี้จะเป็นข้อมูลที่ได้มาจากภาพที่เขียนจากโปรแกรมแล้วเปลี่ยนเป็นระยะทาง ความเร็ว โดยการใช้โปรแกรมที่เขียนขึ้น

ภาพที่ได้จากการวาดด้วยเครื่องเขียนแบบขนาดเล็ก จะเป็นเดียวกับภาพที่เขียนขึ้นโดยคอมพิวเตอร์และขนาดก็จะได้ตามที่ต้องการ ส่วนในด้านความคมชัดก็จะขึ้นกับปากกาที่ใช้และระยะการยกปากกาซึ่งก่อนวาดก็จะต้องมีการตั้งระยะปากกาให้พอดีกับการยกขึ้นยกลง

6.2 ข้อจำกัดของโครงงาน

ส่วนการเคลื่อนที่

- ระยะการเคลื่อนที่มีขนาดไม่เกิน 75 * 75 มิลลิเมตร
- ความเร็วในการเขียน
- การวาดมุมต่ำสุดจะประมาณที่ 2 องศา
- ไม่มีระบบตรวจสอบระยะทาง

ส่วนของโปรแกรม

- สเต็ปการเขียนที่ใช้ไม่เกิน 500 สเต็ป
- การปัดเศษทศนิยม 3 ตำแหน่ง ของมิลลิเมตร
- ข้อมูล file plot ของ Auto CAD R14 จะต้องตั้งเครื่อง PLOT ตามที่กำหนด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

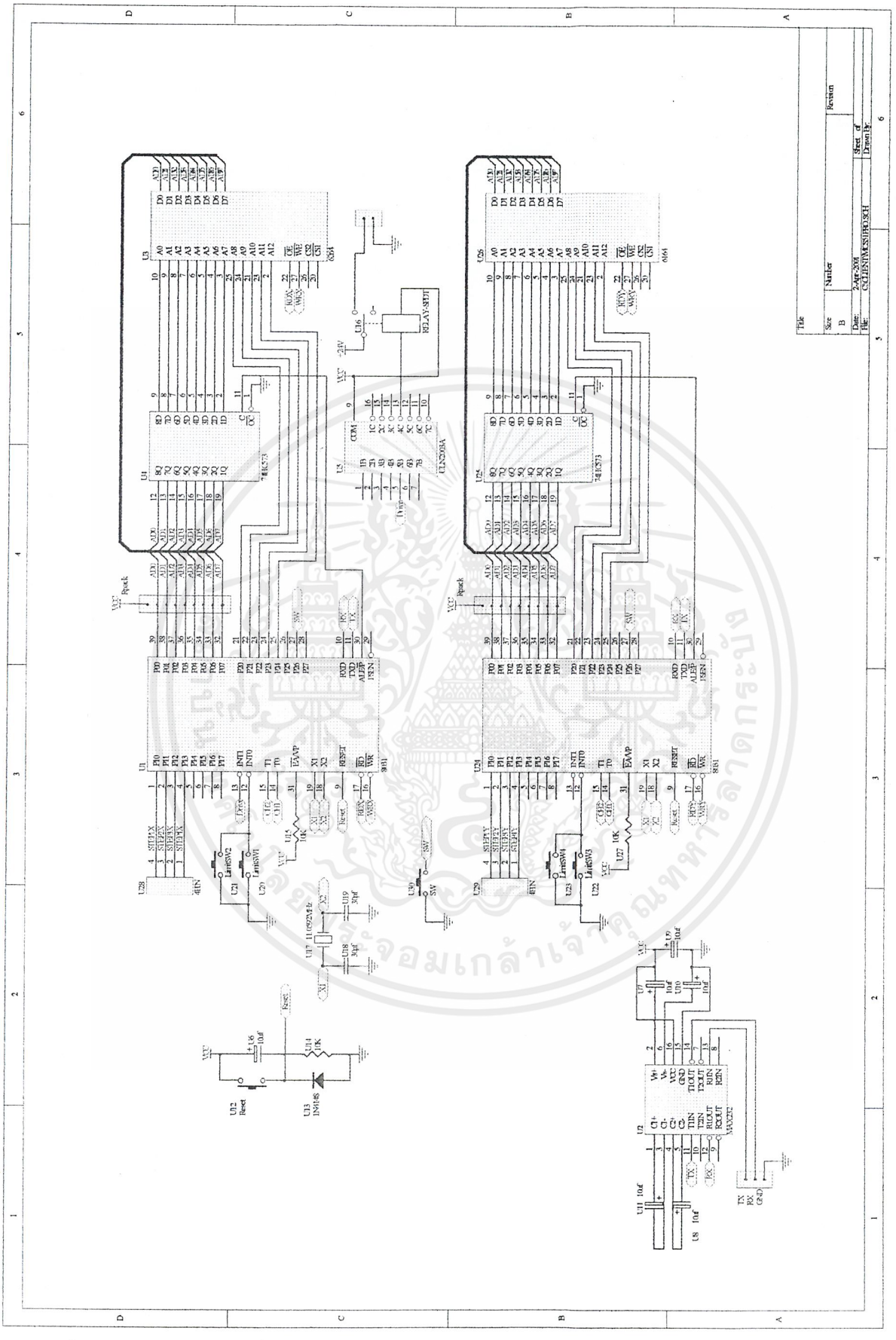
บรรณานุกรม

1. “การประยุกต์ใช้งานไมโครคอนโทรลเลอร์ ตระกูล MCS-51” , สมยศ จุณณะปิยะ คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง , 2541
2. “เรียนรู้และปฏิบัติการ ไมโครคอนโทรลเลอร์ MSC-51” , Innovative Experiment Co.,Ltd
3. “หลักการโปรแกรม วิเชวลเบสิก” , รศ. เดือน สิริพัธ์ประทุม ภาควิชาวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์ มหาลัยบูรพา
4. “Visual Basic 6 ฉบับโปรแกรมเมอร์” , กิตติ ภักดีวัฒนะกุล-จำลอง ครูอุตสาหะ
5. “คู่มือการใช้โปรแกรม Auto CAD 2000:3D Modeling” , ภานุพงษ์ ปัตติสิงห์
6. “Advanced Auto CAD” , กอบเกียรติ สระอุบล กรุงเทพ:ซีเอ็ดยูเคชั่น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

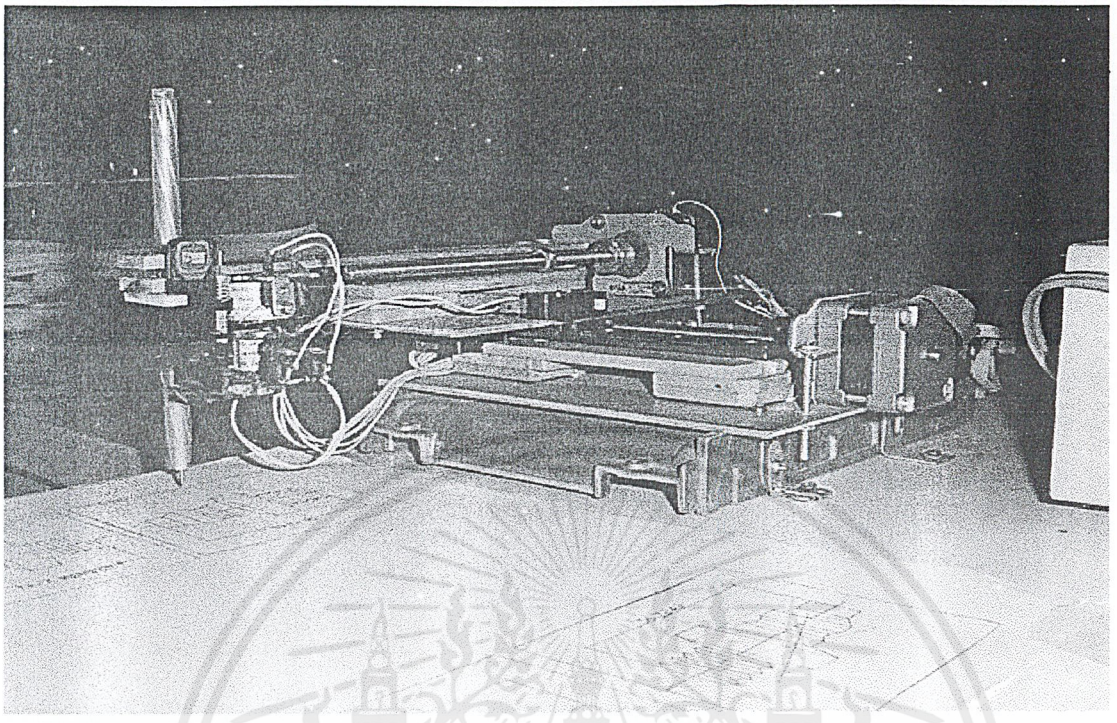
ภาคผนวก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

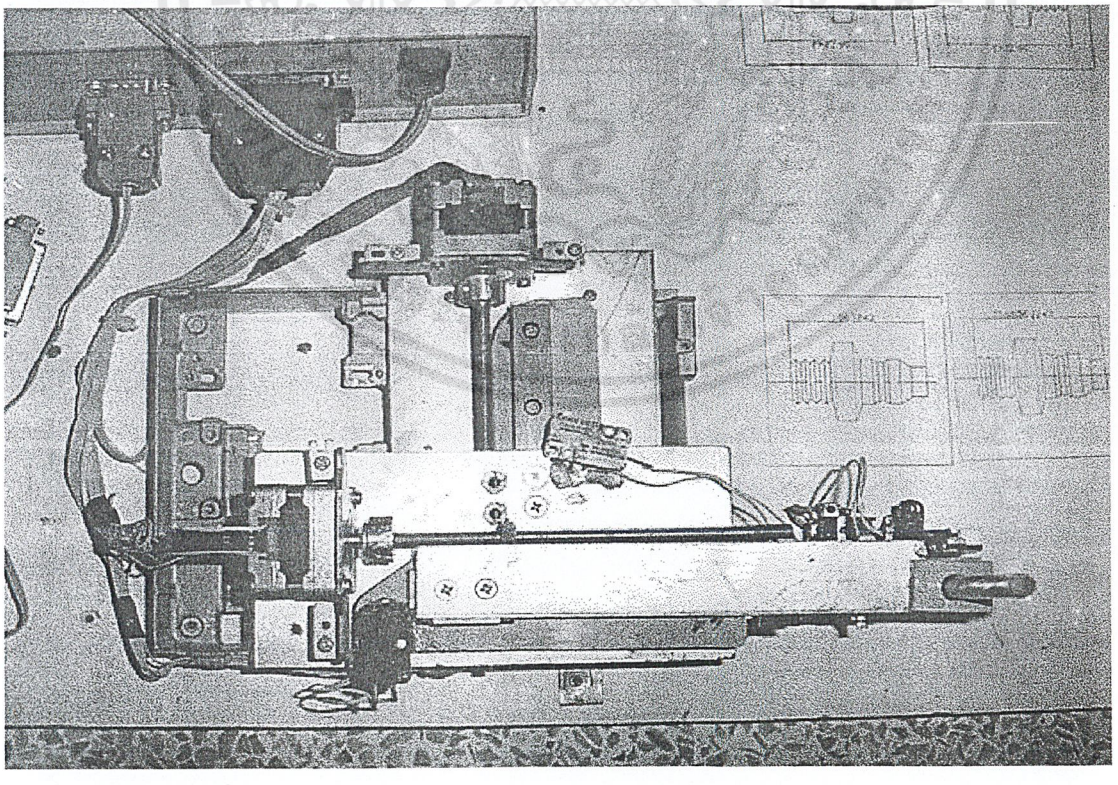


ภาพที่ 1 แสดงวงจรโดยรวมของชุดควบคุมการเคลื่อนที่ของแกน X แกน Y

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

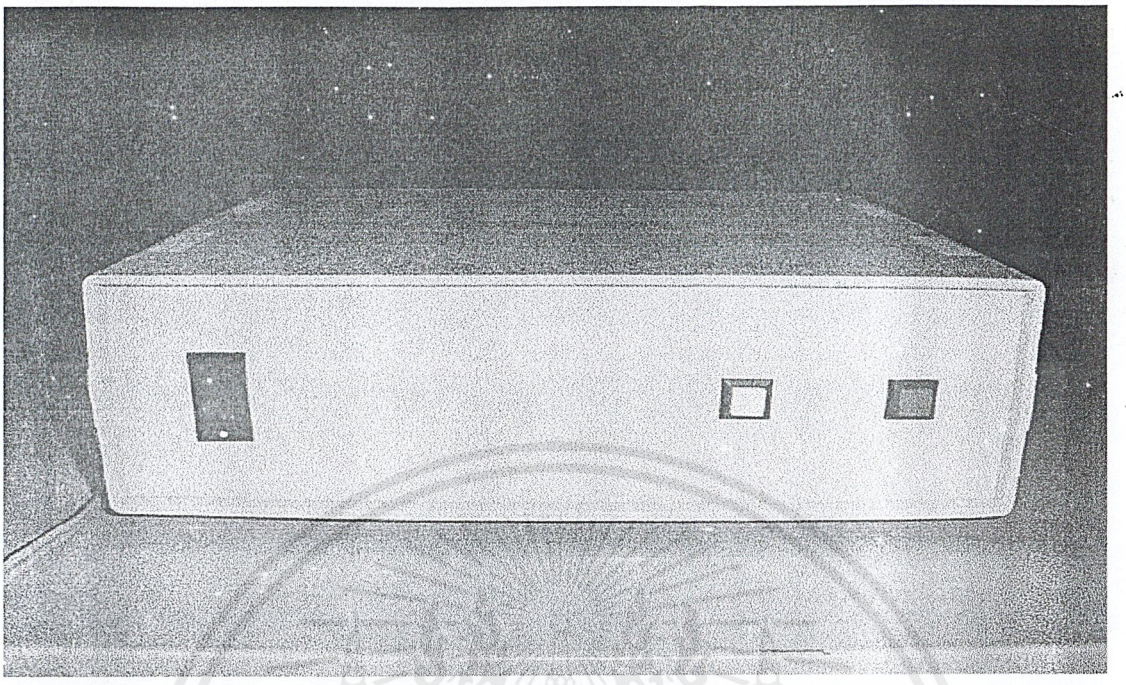


ภาพที่ 2 แสดงรูปร่างของเครื่องเขียนแบบขนาดเล็กด้านข้าง

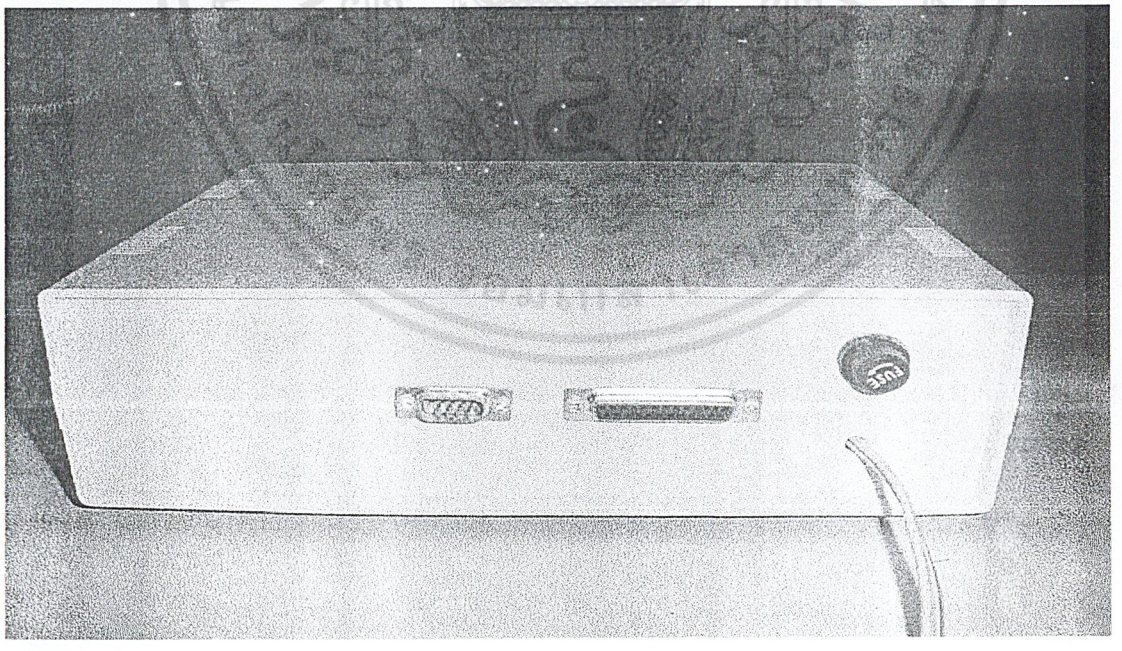


ภาพที่ 3 แสดงรูปร่างของเครื่องเขียนแบบขนาดเล็กด้านบน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 4 แสดงด้านหน้าของกล่องชุดควบคุมการเคลื่อนที่



ภาพที่ 5 แสดงด้านหลังของกล่องชุดควบคุมการเคลื่อนที่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้