สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

เครื่องตัดเหล็กอัตโนมัติด้วยพลาสมา

AUTOMATIC PLASMA CUTTING

นายธิติกร วงษ์งามขำ

นายวิวัฒน์ ครุเสถียร

เลขหมู่.....
เลขทะเบียน 45843
วัน, เดือน, ปี 19 ก.พ. 2546

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

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมการวัดคุม

ภาควิชาวิศวกรรมการวัดคุม คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2544

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

AUTOMATIC PLASMA CUTTING

THITIKORN VONGNGAMKUM

WIWAT KARUSATEAN

A THESIS SUBMITTED IN PARTIAL FULFILLMENT

OF THE REQUIREMENT FOR THE DEGREE OF

BACHELOR OF ENGINEERING IN INSTRUMENTATION ENGINEERING

DEPARTMENT OF INSTRUMENTATION ENGINEERING

FACULTY OF ENGINEERING

KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

2001

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาควิชาวิศวกรรมการวัดคุม

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ใบรับรองปริญญานิพนธ์

หัวข้อปริญญานิพนธ์ เครื่องตัดเหล็กอัตโนมัติด้วยพลาสมา

AUTOMATIC PLASMA CUTTING

นักศึกษาผู้จัดทำ นายฐิติกร วงษ์งามขำ รหัสประจำตัว 41014122

นายวิวัฒน์ คุรุเสถียร รหัสประจำตัว 41014391

ปริญญา วิศวกรรมศาสตรบัณฑิต

สาขาวิชา วิศวกรรมการวัดคุม

ปีการศึกษา 2544

อาจารย์ผู้ควบคุมปริญญานิพนธ์	ลายมือชื่อ
ผศ.ประภาษ อุคคกิมาพันธุ์	

วัน/เดือน/ปี ที่สอบ วันพฤหัสบดีที่ 28 มีนาคม พ.ศ. 2545

สถานที่สอบ ณ ห้องสอบปริญญานิพนธ์ ภาควิชาวิศวกรรมการวัดคุม

ภาควิชารับรองแล้ว

(ผศ.ประสิทธ์ จุลเสรีวงศ์)

หัวหน้าภาควิชาวิศวกรรมการวัดคุม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อปริญญานิพนธ์	**เครื่องตัดเหล็กอัตโนมัติด้วยพลาสมา**	
	AUTOMATIC PLASMA CUTTING	
นักศึกษาผู้จัดทำ	นายฐิติกร	วงษ์งามขำ
	นายวิวัฒน์	คุรุเสถียร
อาจารย์ที่ปรึกษา	ผศ.ประภาษ	อุคคกิมาพันธุ์
ปีการศึกษา	2544	

บทคัดย่อ

ปริญญานิพนธ์ฉบับนี้เสนอการออกแบบและเขียนโปรแกรมสำหรับการตัดแผ่นเหล็กขนาด 1*1 เมตร โดยใช้คอมพิวเตอร์บุคคลประมวลผลข้อมูลที่ผู้ใช้ได้ป้อนเข้าไป ผ่านทางโปรแกรม Visual Basic หรือ โปรแกรม AutoCAD ในรูปไฟล์ข้อมูล PLT และ ส่งคำสั่งและพิกัด X, Y และสถานภาพของหัวตัด ผ่านระบบเชื่อมต่อมาตรฐาน RS232 ให้กับชุดควบคุมการเคลื่อนที่บนแกน X-Y และชุดตัดเหล็กด้วย Plasma

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Thesis Title	**Automatic Plasma Cutting**
Authers	**Mr. Thitikorn Vongngamkum**
	Mr. Wiwat Karusateun
Thesis Advisor	**Asst. Prof. Prapart Ukakimaparn**
Year	**2001**

ABSTRACT

The Project present designing and programming to Automatic Plasma Cutting which has dimension 1*1 Meter. It uses personal computer to compute data from user's input through program Visual Basic or program AutoCAD to orders, 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 อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

ปริญญานิพนธ์ฉบับนี้สำเร็จลุล่วงได้เพราะได้รับความเมตตาจาก ผศ.ประภาษ อุคคกิมาพันธุ์ ที่ได้ให้คำแนะนำ อีกทั้งยังเอื้อเฟื้ออุปกรณ์และเครื่องมือต่างๆ ในการปริญญานิพนธ์นี้

ขอขอบพระคุณอาจารย์ภาควิชาวิศวกรรมการวัดคุมทุกท่านที่ให้คำแนะนำอันเป็นประโยชน์ต่อการทำปริญญานิพนธ์ฉบับนี้

คุณค่าและประโยชน์อันพึงมีจากปริญญานิพนธ์ฉบับนี้ ขอมอบแด่ผู้มีพระคุณทุกท่าน

คณะผู้จัดทำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

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 ความรู้เบื้องต้นเกี่ยวกับเครื่องตัดแผ่นเหล็กด้วยพลาสมา 2

2.1 กล่าวนำ 2

บทที่ 3 หลักการและทฤษฎีที่เกี่ยวข้อง 4

3.1 โปรแกรม Visual Basic 4

3.1.1 ส่วนประกอบจอภาพของ Visual Basic 6

3.1.2 ขั้นตอนในการพัฒนาโปรแกรมของ Visual Basic 8

3.1.3 Object-Oriented Programming กับ Visual Basic 8

3.1.4 การกำหนดคุณสมบัติ (Property) ให้กับ Object 10

3.1.5 การ Run และเลิกงาน Project 10

3.1.6 Control ภายในของ Visual basic 11

3.2 การสื่อสารแบบอนุกรม 20

3.2.1 การสื่อสารข้อมูลแบบอะซิงโครนัส 21

3.2.2 มาตรฐานพอร์ตอนุกรมแบบ RS-232 22

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ(ต่อ)

หน้า

3.2.3 คอนเน็กเตอร์สำหรับพอร์ต RS-232 และการเชื่อมต่อ..... 23

3.2.4 UART..... 26

3.2.5 รีจิสเตอร์ของพอร์ตอนุกรม RS-232..... 27

3.2.6 แอดเดรสของพอร์ตอนุกรม..... 29

3.2.7 ระดับแรงดันที่ใช้งานสำหรับพอร์ตอนุกรม RS-232..... 30

3.3 สเต็ปปิ้งมอเตอร์..... 30

3.3.1 หลักการทำงานของสเต็ปปิ้งมอเตอร์..... 30

3.3.2 โครงสร้างและหลักการพื้นฐานของสเต็ปปิ้งมอเตอร์..... 31

3.3.3 สเต็ปปิ้งมอเตอร์ชนิดต่างๆ..... 33

3.3.3.1 สเต็ปปิ้งมอเตอร์แบบแม่เหล็กถาวร..... 33

3.3.3.2 สเต็ปปิ้งมอเตอร์แบบแปรค่ารีลัคแตนซ์ได้..... 34

3.3.3.3 สเต็ปปิ้งมอเตอร์แบบไฮบริด..... 36

3.3.4 สัญลักษณ์ต่างๆ ของสเต็ปปิ้งมอเตอร์..... 37

3.3.5 พารามิเตอร์ต่างๆ ของสเต็ปปิ้งมอเตอร์..... 37

3.3.6 การควบคุมการทำงานของสเต็ปปิ้งมอเตอร์..... 38

3.3.7 การสั่งให้สเต็ปปิ้งมอเตอร์หมุน..... 39

3.3.7.1 แบบจ่ายกระแสไฟฟ้าให้เฟสเดียวสลับกันไป..... 39

3.3.7.2 แบบจ่ายกระแสไฟฟ้าให้พร้อมกันทีเดียว 2 เฟส..... 39

3.3.7.3 แบบจ่ายกระแสไฟให้ทีละ 1 เฟส สลับกับ 2 เฟส..... 39

3.3.8 หลักการออกแบบวงจรขับสเต็ปปิ้งมอเตอร์..... 39

3.4 โปรแกรม AutoCAD..... 40

3.4.1การทำ File Plot..... 42

3.4.1.1DXF..... 42

3.4.1.2HPGL..... 42

3.4.1.3 เหตุผลที่เลือกใช้ File Plot แบบ HPGL..... 43

3.4.1.4 วิธีการตั้งเครื่อง Plot ในโปรแกรม AutoCAD..... 43

3.4.2วิธีการเริ่มต้นเขียนแบบในโปรแกรม AutoCAD..... 46

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ(ต่อ)

หน้า

3.4.3 วิธีการตั้งค่า File PLT ของโปรแกรม AutoCAD (หลังจากการเขียนภาพ)..48

บทที่ 4 การออกแบบและสร้างเครื่องตัดเหล็กอัตโนมัติด้วยพลาสมา..... 51

4.1 การออกแบบส่วนโปรแกรม..... 51

4.2 โหมดอัตโนมัติ..... 54

4.3 โปรแกรมโหมดManual..... 64

บทที่ 5 ผลการทดลอง..... 68

5.1 ผลจากโหมด Auto..... 69

5.2 ผลการทดลองโหมด Manual..... 76

บทที่ 6 สรุปผลและข้อจำกัดของโครงงาน..... 79

6.1 สรุปผล..... 79

6.2 ข้อจำกัดของโครงงาน..... 79

6.2.1 ส่วนของการเคลื่อนที่..... 79

6.2.2 ส่วนของโปรแกรม..... 79

6.2.3 ส่วนของ Controller..... 79

6.3 ปัญหาของโครงงาน..... 79

บรรณานุกรม..... 80

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

ตารางที่ หน้า

3.1 แสดงคุณสมบัติปกติของคอนโทรลที่สำคัญ....12
3.2 Property ของคอนโทรล Label....14
3.3 เปรียบเทียบ DB-9 และ DB-25....24
3.4 แสดงจำนวนของพอร์ตอนุกรมที่มีใช้อยู่ในคอมพิวเตอร์....29
3.5 การจ่ายกระแสไฟให้กับสเต็ปปิ้งมอเตอร์ แบบ Full Step....39

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญภาพ

ภาพที่ หน้า

2.1 แผนภูมิการทำงานโดยรวมของโปรแกรมทั้งหมด....3

3.1 แสดงหน้าจอตอนเปิดโปรแกรม Visual Basic....4

3.2 แสดงหน้าจอ Visual Basic ตอนเปิด Tab "Existing"....5

3.3 แสดงหน้าจอ Visual Basic ตอนเปิด Tab "Recent"....6

3.4 แสดงส่วนประกอบจอภาพของ Visual basic....6

3.5 แสดงการวาด Control ลงบน Form....9

3.6 แสดงการ Save File ของ Visual Basic....11

3.7 แสดงคอนโทรล CommandButton ที่วางลงบนฟอร์ม....14

3.8 แสดง แสดงคอนโทรล Label ที่วางลงบนฟอร์ม....14

3.9 แสดงคอนโทรล TextBox ที่วางลงบนฟอร์ม....15

3.10 แสดงคอนโทรล Timer ที่วางลงบนฟอร์ม....16

3.11 แสดงคอนโทรล Data ที่วางลงบนฟอร์ม....18

3.12 แสดงคอนโทรล MScomm ที่วางลงบนฟอร์ม....20

3.13 แสดงรูปแบบอย่างง่ายที่สุดของข้อมูลอนุกรม....21

3.14 รูปแบบอย่างง่ายที่สุดของข้อมูลอนุกรมแบบสื่อสารข้อมูลแบบอะซิงโครนัส....22

3.15 แสดงคอนเน็กเตอร์อนุกรม 9 ขาหรือแบบ DB-9....23

3.16 แสดงคอนเน็กเตอร์อนุกรม 25 ขาหรือแบบ DB-25....24

3.17 แสดงการต่ออุปกรณ์ภายนอกเข้ากับคอมพิวเตอร์แบบ Null modem....25

3.18 แสดงการต่ออุปกรณ์ภายนอกเข้ากับคอมพิวเตอร์แบบ RS-232....25

3.19 แสดงบล็อกไดอะแกรมสัญญาณอินพุตและเอาต์พุตของพอร์ต RS-232....28

3.20 แสดงคอนโทรลเลอร์ สำหรับควบคุมการทำงานของสเต็ปปิ้งมอเตอร์....30

3.21 (ก) แสดงสเต็ปปิ้งมอเตอร์ที่มีการต่อวงจรขดลวดภายในเพื่อกระตุ้นให้เกิดขั้วแม่เหล็กขึ้น1 ขั้ว ในทิศทางตรงกันข้าม ส่วนขดลวดอื่นๆ จะไม่ถูกกระตุ้น....31

3.21 (ข) แสดงการต่อวงจรขดลวดแบบกระตุ้นให้เกิดขั้วแม่เหล็กพร้อมกัน 2 ขั้วที่อยู่ใกล้กัน....31

3.22 แสดงวงจรจ่ายกำลังไฟฟ้าให้กับสเต็ปปิ้งมอเตอร์....32

3.23 แสดงแรงดึงดูด ทำให้เกิดทอร์กที่หมุนโรเตอร์ให้ไปอยู่ในตำแหน่งที่สมดุลย์....32

3.24 แสดงโครงสร้างของสเต็ปปิ้งมอเตอร์แบบแม่เหล็กถาวร....33

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญภาพ(ต่อ)

ภาพที่ หน้า

3.25 แสดงการทำงานในสเต็ปปิ้งมอเตอร์แบบแม่เหล็กถาวรขนาดสี่เฟส....33

3.26 สเต็ปปิ้งมอเตอร์แบบแปรค่ารีลัคแตนซ์ได้....34

3.27 แสดงตำแหน่งสมดุลย์เมื่อเฟลไดเฟสหนึ่งของสเต็ปปิ้งมอเตอร์ถูกกระตุ้น....34

3.28 แสดงการเปลี่ยนลำดับสเต็ปในการเคลื่อนที่ของสเต็ปปิ้งมอเตอร์ได้....35

3.29 แสดงโครงสร้างสเต็ปปิ้งมอเตอร์แบบไฮบริด....36

3.30 แสดงหลักการทำดงานของสเต็ปปิ้งมอเตอร์แบบไฮบริด....36

3.31 แสดงบล็อกไดอะแกรมของระบบควบคุมการขับสเต็ปปิ้งมอเตอร์....38

3.32 แสดงวงจรสมสูลย์ของสเต็ปปิ้งมอเตอร์....40

3.33 แสดงหน้าต่างของโปรแกรม Auto CAD R14....41

3.34 แสดงเมนู Tool ของ Auto Cad....44

3.35 แสดงหน้าจอ Tab "Printer"....44

3.36 แสดงหน้าจอ Add Printer....45

3.37 แสดงหน้าจอเริ่มต้นเมื่อเปิด Auto Cad....46

3.38 แสดงหน้าจอเมื่อพิมพ์คำว่า Command : rectangle ↵....46

3.39 แสดงหน้าจอเมื่อพิมพ์คำว่า 50,50 ↵ 100,100 ↵....46

3.40 แสดงรูปสี่เหลี่ยมที่มีมุมที่พิกัด (50,50), (50,100), (100,100), (100,50)....47

3.41 แสดงหน้าจอเมื่อพิมพ์คำว่า Command : circle ↵....47

3.42 แสดงหน้าจอเมื่อพิมพ์ 100,100↵ 20↵....47

3.43 แสดงเมนู File....48

3.44 แสดงหน้าจอ Print/Plot Configuration....49

3.45 แสดงรูปสี่เหลี่ยมที่มีมุมที่พิกัด (50,50), (50,100), (100,100), (100,50)....50

4.1 แสดงFlow Chart การทำงานโหมด Auto....52

4.2 แสดง Flow Chart การทำงานโหมด Manual....53

4.3 แสดงปุ่มวิเคราะห์ทางเดิน....54

4.4 แสดงปุ่มโหลดจาก File Plot....54

4.5 แสดงว่าเมื่อในตารางเดิมมีข้อมูลอยู่....55

4.6 แสดงปุ่มแปลงเป็นพิกัด Increment....56

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญภาพ(ต่อ)

ภาพที่ หน้า

4.7 แสดงสี่เหลี่ยมรูปที่ 1 เป็น พิกัด(50,50) และ พิกัด(100,100) และสี่เหลี่ยมรูปที่ 2 เป็น พิกัด(150,150) และ พิกัด(200,200) 57

4.8 แสดงค่าในตาราง 57

4.9 แสดงปุ่มวิเคราะห์ทางเดินแล้วแปลงเป็นพิกัด Increment 58

4.10 แสดงสี่เหลี่ยม 4 รูป โดยมาการซ้อนกันของสี่เหลี่ยม 60

4.11 แสดงรูปสี่เหลี่ยม 4 รูปที่ซ้อนกัน และมีการจัดเป็นกลุ่มโดยจากรูปที่อยู่ในสุด 61

4.12 แสดงว่า รูป 2 กับ 4 ได้ใช้ไว้แล้ว 62

4.13 แสดง Form ฟังก์ชั่นเส้นตรงของโหมด Manual 65

4.14 แสดงฟังก์ชั่นวงกลมของโหมด Manual 66

4.15 แสดงฟังก์ชั่นวงรีของโหมด Manual 67

4.16 แสดงฟังก์ชั่นเส้นโค้งของโหมด Manual 68

5.1 แสดงหน้าจอของโปรแกรม 69

5.2 แสดงหน้าต่างของโปรแกรมส่วนตาราง เมื่อกดปุ่มดูค่าในตาราง 70

5.3 แสดงภาพในโปรแกรม AutoCAD ที่มาจากค่าในตาราง 71

5.4 แสดงเมื่อตารางมีค่าตัวแปรเก่าอยู่ 72

5.5 แสดงค่าในตารางที่เป็นพิกัดในระบบ Absolute 72

5.6 แสดงค่าในตารางที่อยู่ในระบบ Increment 73

5.7 แสดงการวาดรูปจากค่าในตารางที่ยังไม่ได้วิเคราะห์ 74

5.8 แสดงการวาดภาพจากค่าในตารางที่ผ่านการวิเคราะห์มาแล้ว 75

5.9 แสดงค่าในตารางที่เป็นระบบ Absolute 76

5.10 แสดงค่าในตารางที่เปลี่ยนเป็นระบบ Increment 76

5.11 แสดงกาววาดภาพจากค่าในตารางของฟังก์ชั่นเส้นตรงในโหมด Manual 77

5.12 แสดงการวาดภาพของฟังก์ชั่นวงกลมในโหมด Manual 78

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1
บทนำ

1.1 ความเป็นมาและเหตุจูงใจ

การจะประกอบชิ้นงานชิ้นหนึ่งนั้นจะเริ่มจากการทำส่วนย่อยๆที่ละส่วน แล้วนำมาประกอบกัน ชิ้นงานที่ทำด้วยโลหะจะประกอบจากชิ้นโลหะย่อยๆ ดังนั้นการตัดแผ่นโลหะเพื่อนำมาเป็นส่วนประกอบของชิ้นงานนั้นจึงเป็นจุดเริ่มพื้นฐานของงานหลายๆอย่าง ในโครงงานนี้จึงเลือกการตัดด้วยพลาสม่า ซึ่งเป็นการตัดแผ่นโลหะที่นิยมกันมากอย่างหนึ่งที่ได้ใช้กันอย่างแพร่หลาย และเมื่อนำ Software มาควบคุม การตัดด้วยพลาสมา การตัดชิ้นงานจะได้ขนาดที่ถูกต้องและรวดเร็วยิ่งขึ้น เพราะ Software มีการจัดลำดับการตัดให้มีความเร็วยิ่งขึ้น

1.2 วัตถุประสงค์ของปริญญานิพนธ์

ปริญญานิพนธ์นี้จะเป็นการศึกษาการเขียนโปรแกรมเพื่อจัดเรียงเวคเตอร์ที่ถูกวาดใน AutoCad เพื่อให้ได้ทางเดินที่เร็วเพื่อประหยัดเวลาในการตัดชิ้นงานด้วยพลาสม่า และให้มีความเที่ยงตรงแม่นยำ และเมื่อเรียงเวคเตอร์แล้วจัดเก็บข้อมูลแล้วทำการส่งค่าไปที่ Controller และพลาสมาที่ได้จัดเตรียมไว้แล้ว และสามารถนำโปรแกรมนี้ไปประยุกต์ใช้งานในทางอื่นๆ

1.3 ขอบเขตของปริญญานิพนธ์

ปริญญานิพนธ์เล่มนี้จะกล่าวถึงการเขียนโปรแกรมด้วย Visual Basic เพื่อสั่งการเคลื่อนที่ของมอเตอร์ 2 แกนที่ควบคุมด้วย Controller ที่ควบคุมมอเตอร์ 2 ตัวเพื่อนำไปใช้ในการตัดแผ่นโลหะด้วยพลาสม่าโดยผู้ใช้จะทำการวาดรูปต้นแบบจากโปรแกรม Auto Cad จะเป็นข้อมูลในรูปของไฟล์ Plot (Plt) หรือ สามารถวาดรูปต้นแบบจากโปรแกรมที่เขียนขึ้นมาก็ได้ แล้วประมวลข้อมูลโดยโปรแกรมที่เขียนด้วย Visual Basic แล้วส่งข้อมูลที่เป็นพิกัดและสถานะภาพของตัวตัดพลาสม่า ผ่าน RS-232 เข้า Controller เพื่อควบคุมการตัดชิ้นงานด้วยพลาสม่า

1.4 ขั้นตอนการศึกษา

การทำปริญญานิพนธ์ฉบับนี้เริ่มจากการศึกษาระบบ File Plot ของโปรแกรม Auto Cad เพื่อดูทางเดินของเวคเตอร์ และศึกษาโปรแกรม Visual Basic เพื่อเขียนโปรแกรมมาทำการจัดเรียงเวคเตอร์ให้ได้ทางเดินที่เร็ว และสามารถวาดทางเดินในโปรแกรมได้โดยไม่ต้องผ่าน Auto Cad และ Controller ที่ควบคุมมอเตอร์ 2 แกนว่ามีระบบการรับข้อมูลเพื่อทำการควบคุมมอเตอร์แบบใด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2
ความรู้เบื้องต้นเกี่ยวกับเครื่องตัดแผ่นเหล็กด้วยพลาสมา

2.1 กล่าวนำ

การที่จะตัดชิ้นงานให้ได้มาตราฐานเดิมทุกครั้งนั้นเป็นสิ่งที่สำคัญในงานวิศวกรรมหรือการประกอบชิ้นงานทั่วไป ดังนั้นจึงประยุกต์โดยการนำคอมพิวเตอร์ ระบบควบคุมมอเตอร์และชุดเคลื่อนที่ของมอเตอร์มาประยุกต์ใช้ในการตัดแผ่นโลหะให้ได้มาตราฐาน พร้อมทั้งเขียนโปรแกรมเพื่อใช้สำหรับการใช้งานในการตัดแผ่นโลหะผ่านทางคอมพิวเตอร์

ในการวาดต้นแบบของแผ่นเหล็กนี้จะแบ่งเป็น 2 แบบ คือ

1) วาดต้นแบบของแผ่นเหล็กในโปรแกรม Auto Cad

2) วาดต้นแบบของแผ่นเหล็กในโปรแกรมที่เขียนขึ้นมาด้วย Visual Basic

การวาดด้วยโปรแกรม Auto Cad (ที่อ้างอิงกับโปรแกรม Auto Cad เพราะว่าโปรแกรม Auto Cad เป็นโปรแกรมที่ออกแบบมาเพื่อการเขียนแบบที่ละเอียดและมีผู้ใช้งานเป็นอยู่มากและเป็นแบบแผน) หลังจากนั้นทำการเก็บข้อมูลในระบ File Plotแต่การเก็บข้อมูลแบบ File Plot นั้นจะมีการเรียงเวคเตอร์ที่อาจทำให้การตัดใช้เวลานาน อย่างเช่น เมื่อวาดสี่เหลี่ยมที่ระยะใกล้ 0,0 และมีวงกลมในระยะใกล้กัน และมีสามเหลี่ยมที่ระยะไกลออกไปอีก การวาดของโปรแกรม Auto Cad อาจจะวาดสี่เหลี่ยม แล้วไปที่สามเหลี่ยม แล้วค่อยมาที่วงกลม ซึ่งถ้าการตัดจริงถ้าตัดตามการวาดของโปรแกรม Auto Cad จะทำให้การตัดช้าเพราะว่าต้องมีการวิ่งกลับไปกลับมาของตัวตัดชิ้นงาน (พลาสม่า) ดังนั้นจึงมีการเขียน โปรแกรมขึ้นมาเพื่อทำการจัดเรียงเวคเตอร์ของโปรแกรม Auto Cad ให้มีการจัดเรียงเวคเตอร์ใหม่เพื่อให้การตัดชิ้นงานด้วยพลาสม่าวิ่งเร็วโดยการขจัดการวิ่งกลับไปกลับมาของเวคเตอร์ให้วิ่งไปทีเดียวกลับทีเดียว

และในโปรแกรมที่เขียนขึ้นด้วย Visual Basic นี้ได้มีการเขียนฟังก์ชั่นการวาดแบบของแผ่นเหล็กโดยไม่ต้องวาดผ่านโปรแกรม Auto cad ด้วยเพื่อความสะดวกสำหรับผู้ใช้งานโดยจะใส่ฟังก์ชั่นสำเร็จรูปโดยให้ผู้ใช้งานป้อนค่าที่สำคัญๆ ของฟังก์ชั่นนั้นๆ

หลังจากมีการเรียงเวคเตอร์เสร็จเรียบร้อยจะมีการจัดเก็บข้อมูลที่จะส่งค่าให้ Controller เป็นค่าพิกัด X และ Y และ สถานะของปลายของตัวพลาสม่าว่าจะให้ยกขึ้น (“PU”) , ยกลง (“PD”) , เคลื่อนที่ไปยังพิกัด X,Y (“PA”)

เมื่อ Controller ค่าจาก Computer โดยผ่านทางพอร์ท RS-232 แล้วจะทำการขับชุดไดร์ฟของ Stepping มอเตอร์เพื่อให้มอเตอร์เคลื่อนที่ยังพิกัดตามที่ได้รับจากโปรแกรม และมีการป้อนค่าเพื่อควบคุมความเร็วของตัวมอเตอร์ โดยที่ปลายของชุดเคลื่อนที่ 2 แกนด้วย Stepping มอเตอร์จะถูกขับด้วยตัวขับมอเตอร์เพื่อทำการตัดแผ่นเหล็ก

เอกสารนี้สงวนลิขสิทธิ์ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาพที่ 2.1 แผนภูมิการทำงานโดยรวมของโปรแกรมทั้งหมด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

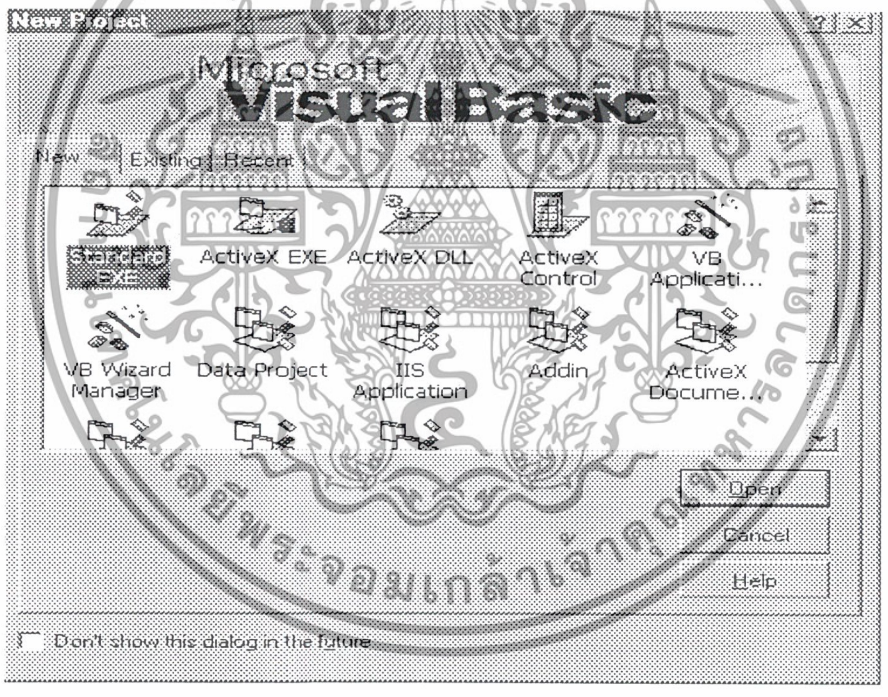
บทที่ 3
หลักการและทฤษฎีที่เกี่ยวข้อง

3.1 โปรแกรม Visual Basic

โปรแกรมที่จะเขียนขึ้นมานี้จะเขียนด้วยโปรแกรม Visual Basic ซึ่งจะมีรายละเอียด ดังต่อไปนี้

เมื่อเปิดโปรแกรม Visual Basic ขึ้นมาจะปรากฏจอสำหรับเปิดProject ซึ่งจะประกอบไปด้วย 3 Tab ดังนี้

1. Tab "New" เป็นจอภาพที่ประกอบไปด้วยiconต่างๆที่ใช้สำหรับเรียกใช้ Project ใหม่ขึ้นมาใช้งานดังรูป



ภาพที่ 3.1 แสดงหน้าจอตอนเปิดโปรแกรม Visual Basic

ซึ่ง icon ต่างๆ มีความหมายดังนี้

-Standard EXE ใช้สร้างโปรแกรมในแบบ GUI โดยทั่วไป

-ActiveX EXE ใช้สร้างโปรแกรมที่ใช้ติดต่อกับโปรแกรมอื่นในรูปแบบของ OLE แบบหนึ่ง ซึ่งเรียกว่า OLE Automation Server

-ActiveX DLL ใช้สร้างโปรแกรมเช่นเดียวกับ ActiveX EXE แต่จะเก็บอยู่ในไฟล์นามสกุล DLL แทนซึ่งไม่สามารถ Run ได้ด้วยตัวเอง จะต้องถูกเรียกใช้โดยโปรแกรมอื่น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

-ActiveX Control ใช้สร้าง ActiveX Control ขึ้นใช้งาน

-VB Application Wizard เป็นเครื่องมือที่ช่วยสร้างโปรแกรมขึ้นใช้งาน

-Addin ใช้เพิ่มเติม Utility อื่นเข้าไว้ใน Visual Basic

-ActiveX Document DLL ใช้สร้างโปรแกรม ActiveX ที่อยู่ในรูปของไฟล์นามสกุล DLL

-ActiveX Document EXE ใช้สร้างโปรแกรมที่อยู่ในรูปของไฟล์นามสกุล EXE

-VB Wizard Manager ใช้สร้างโปรแกรมที่ควบคุมการทำงานของ Wizard

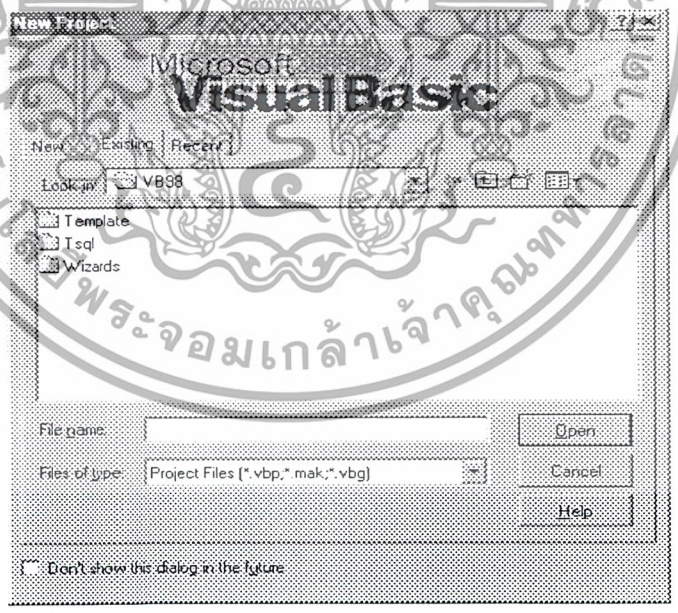
-Data Project ใช้สร้างโปรแกรมที่ใช้ติดต่อกับฐานข้อมูลต่างๆผ่านทาง ODBC หรือ OLEDB

-IIS Application ใช้สร้างโปรแกรมที่ใช้งานบน Internet แบบ IIS

-DHTML Application ใช้สร้างโปรแกรมที่ใช้งานบน Internet แบบ Dynamic HTML

-VB Enterprise Edition Control ใช้สร้างโปรแกรมในแบบ GUI โดยทั่วไป จาก Control ต่างๆ ของ Enterprise Edition

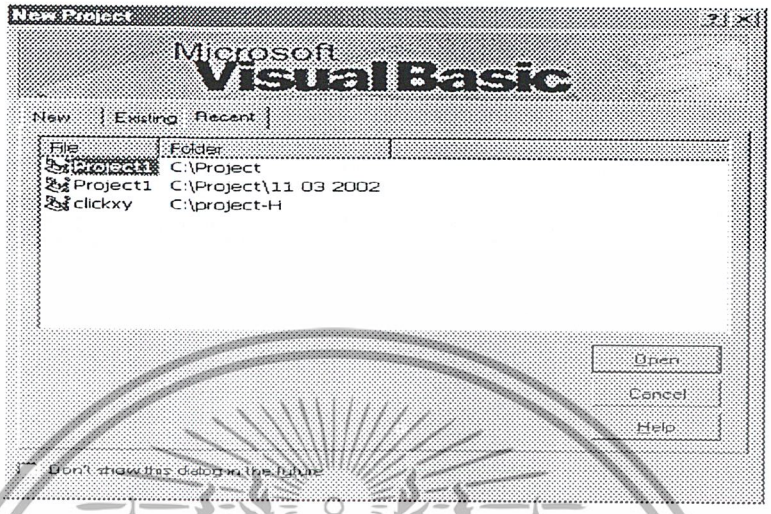
2.Tab " Existing" เป็นจอภาพที่ใช้สำหรับเรียกใช้ Project เดิมที่พัฒนาขึ้นแล้วและเก็บไว้ใน Directory ต่างๆ ขึ้นมาใช้งาน ดังรูป



ภาพที่ 3.2 แสดงหน้าจอ Visual Basic ตอนเปิด Tab "Existing"

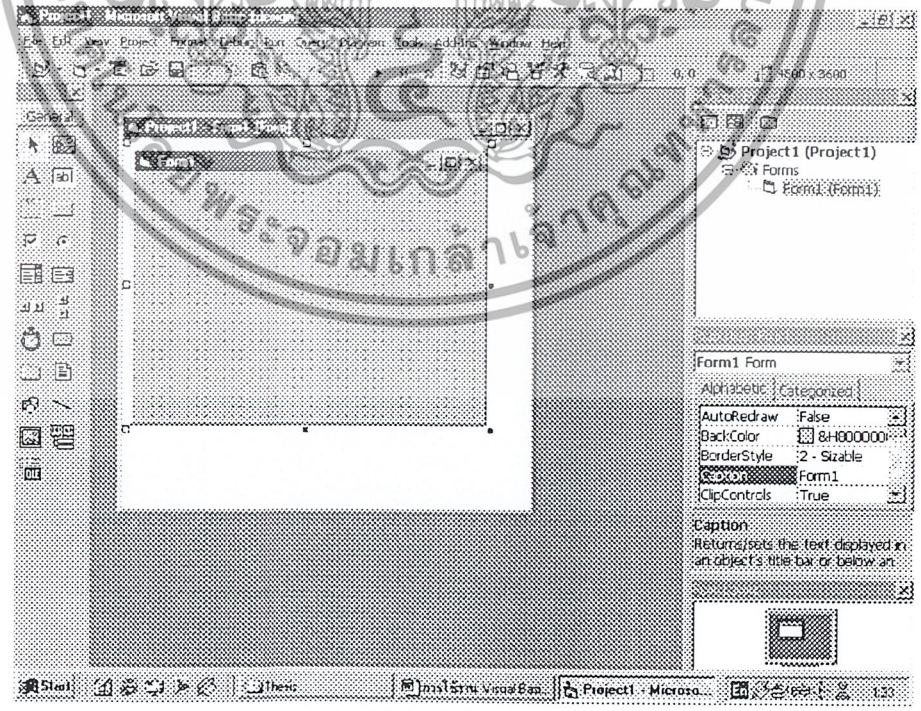
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.Tab "Recent" เป็นจอภาพที่แสดงประวัติของ Project ต่างๆ ที่เคยถูกเรียกขึ้นมาพัฒนา



ภาพที่ 3.3 แสดงหน้าจอ Visual Basic ตอนเปิด Tab "Recent"

3.1.1 ส่วนประกอบของจอภาพ Visual Basic มีดังนี้



ภาพที่ 3.4 แสดงส่วนประกอบจอภาพของ Visual basic

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

-Form เป็นส่วนที่ใช้สำหรับสร้างจอภาพของโปรแกรมขึ้นใช้งาน

-ToolBox เป็นส่วนที่ประกอบด้วย icon ต่าง ซึ่งใน Visual Basic เรียกว่า "Control" ที่จะนำไปใช้งานโดยการนำลงไปวางบน Form สำหรับหน้าที่ของแต่ละ Control

-ToolBar เป็นส่วนที่ประกอบด้วย icon ต่างๆ ที่ใช้ในการพัฒนาโปรแกรม

-Project Explorer Window เป็นจอภาพที่ใช้สำหรับกำหนดคุณสมบัติ (Property) ให้กับ Form และ Object ต่างๆ ที่ปรากฏอยู่บน Form

-Properties Window เป็นจอภาพที่ใช้สำหรับกำหนดคุณสมบัติ (Property) ให้กับ Form และ Object ต่างๆ ที่ปรากฏอยู่บน Form

-Form Layout Window ใช้สำหรับดูตำแหน่งของ Form บนจอภาพ ทำให้จัดตำแหน่งของ Form ได้สะดวกขึ้น

Form

เป็นส่วนที่ใช้สำหรับสร้างจอภาพของโปรแกรม โดยจะทำหน้าที่เป็นพื้น (Background) ของจอภาพ ทุกครั้งของการเปิด Project ใหม่ จะได้ Form

Toolbox

เป็นแถบเครื่องมือที่ประกอบไปด้วย icon ต่างๆ ซึ่งเรียกว่า " Control" ซึ่งจะใช้ร่วมกับ Form เพื่อสร้างจอภาพของ Project แต่ละ Control จะใช้เป็นเครื่องมือที่ใช้สำหรับสร้างส่วนที่ใช้ติดต่อกับผู้ใช้หรือที่เรียกว่า " User Interface" เช่นข้อความต่างๆ ช่องว่างสำหรับรับข้อมูลจากคีย์บอร์ด ปุ่มต่างๆ เป็นต้น และจะถูกนำไปใช้งานด้วย โดยการนำ Control ที่ต้องการไปวางลงบน Form

Project

โดยทั่วไประบบงานหนึ่งๆ มักประกอบด้วยหลายๆจอภาพ เช่น จอภาพสำหรับป้อนข้อมูล (Data Entry) จอภาพสำหรับค้นหาข้อมูล (Data Inquiry) เป็นต้น ดังนั้น ในการพัฒนาโปรแกรม จึงนิยมที่จะแยกแต่ละจอภาพออกเป็นโปรแกรมเพื่อความสะดวกต่อการแก้ไขตามหลักของการเขียนโปรแกรม แล้วจึงนำแต่ละโปรแกรมย่อยมาประกอบกันขึ้นเป็นระบบ โดยการ Compile ไฟล์เหล่านั้นร่วมกันเป็นไฟล์ EXE เพื่อนำไปใช้งาน

ใน Visual Basic ก็เช่นเดียวกัน แต่ละจอภาพที่พัฒนาขึ้นจะได้แก่ Form ต่างๆ และเมื่อนำมารวมกันก็จะกลายเป็นระบบงานระบบหนึ่ง หรือที่เรียกว่า Project และ Form จึงต้องทำงานร่วมกัน จะขาดไฟล์ใดไฟล์หนึ่งไม่ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1.2ขั้นตอนในการพัฒนาโปรแกรมของ Visual Basic

ขั้นตอนที่ 1 สร้างจอภาพของโปรแกรม

ในขั้นตอนนี้ จะนำ Form มาออกแบบเพื่อใช้ในการติดต่อกับผู้ใช้ หรือที่เรียกว่า การออกแบบ "User Interface" ในการพัฒนาโปรแกรมแบบเดิม ขั้นตอนนี้จะใช้เวลาและค่าใช้จ่ายสูง เนื่องจากจะต้องเขียนโปรแกรมเพื่อสร้างจอภาพต่างๆ จากนั้นต้อง Compile โปรแกรมนั้น แล้ว Run จึงจะเห็นจอภาพที่จัดทำขึ้น แต่สำหรับ Visual Basic ปัญหาในลักษณะนี้ได้ถูกแก้ไขโดยใช้เทคนิคของ Visualize ซึ่งเป็นความสามารถส่วนหนึ่งของ Visual basic ขั้นตอนนี้ จึงสามารถทำได้อย่างง่ายดาย เพียงแต่นำเอา Control ต่างๆใน Toolbox ที่ต้องการใช้งานมาวาดไว้บน Form ซึ่งทำให้ประหยัดเวลาและสามารถเห็นลักษณะจอภาพที่ออกแบบได้ในขณะนั้นเลย

ขั้นตอนที่ 2 เขียนโปรแกรม

เมื่อวาง Control ต่างๆลงบน Form เป็นที่เรียบร้อยแล้ว (Control ต่างๆ เมื่อถูกนำมาวาดไว้บน Form จะเรียกว่า "Object") ขั้นตอนต่อมา ได้แก่ การเขียนโปรแกรมเพื่อกำหนดการทำงานให้กับแต่ละ Object ภายใต้เหตุการณ์ต่างๆ (Event) ที่เกิดขึ้นกับจอภาพนั้นๆ

3.1.3 Object-Oriented Programming กับ Visual Basic

Visual Basic เป็นภาษาคอมพิวเตอร์ภาษาหนึ่งที่อาศัยแนวความคิดของ Object-Oriented Programming (OOP) ซึ่งแตกต่างจากการเขียนโปรแกรมในแบบเดิม ที่จะเป็นลักษณะของ Procedural-Oriented กล่าวคือ จะพยายามแบ่งโปรแกรมที่มีขนาดใหญ่ให้เป็นโปรแกรมย่อยหลายๆ โปรแกรม (ตามแนวความคิดของ Top-Down Design) แต่สำหรับ Object-Oriented Programming แล้ว แนวความคิดได้เปลี่ยนไปให้ความสนใจกับสิ่งต่างๆที่อยู่ในโปรแกรม ซึ่งเรียกว่า "Object" แทน

Object ใน Visual Basic ได้แก่ ส่วนของ Control ต่างๆใน Toolbox ที่นำมาวาดลงบน Form ซึ่งจะมีคุณสมบัติบางอย่างเช่นเดียวกับ Object ใน OOP กล่าวคือ แต่ละ Object จะต้องประกอบไปด้วย

1.Data เปรียบเสมือนข้อมูลของ Object สำหรับใน Visual Basic คำว่า "Data" หมายถึง คุณสมบัติ(Property) ประจำตัวของแต่ละ Object เช่น ชื่อ, ความยาว, ความสูง เป็นต้น

2.Code เปรียบเสมือนกับพฤติกรรมของ Object สำหรับใน Visual Basic คำว่า "Code" หมายถึง Method ประจำตัวของแต่ละ Object เช่น Method "Movefirst" ของ Object ชื่อ "Data Control" ที่ใช้สำหรับเลื่อนตัวชี้ (Pointer) ไปยัง Record แรกของข้อมูล เป็นต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ดังนั้น ในการเขียนโปรแกรมแทนที่จะเขียนในแบบเดิมที่เป็นลำดับขั้น ซึ่งเริ่มจากส่วนของ Main Program ที่ทำหน้าที่ Call Subprogram ต่างๆได้เปลี่ยนมาเป็นการเขียนโปรแกรมกับแต่ละ Object แทนโดยอาศัยคำสั่งและ Method ที่เกี่ยวข้องกับแต่ละ Object นั้น

ด้วยหลักการของ OOP นี้ ในการพัฒนาโปรแกรมด้วย Visual Basic จึงแบ่งการทำงานออกเป็น

1.เลือก Object ที่ต้องการใช้งาน ซึ่งได้แก่ การวาด Control ที่ต้องการใช้งานลงบน Form

2.กำหนดข้อมูลในส่วน Data ให้กับแต่ละ Object ที่เลือก ซึ่งได้แก่ การกำหนดคุณสมบัติ (Property) ให้กับ Object

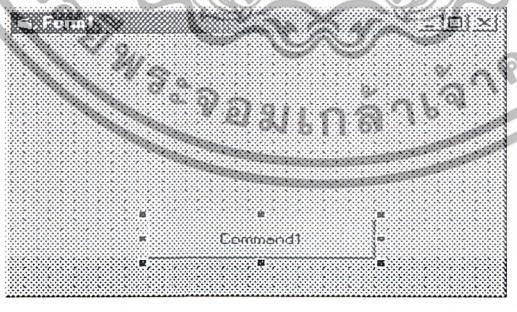
3.นำเอา Method ที่จำเป็นของแต่ละ Object มาเขียนขึ้นเป็นโปรแกรม โดยอาจใช้คำสั่งและฟังก์ชันต่างๆ ของ Visual Basic ประกอบ เพื่อกำหนดการทำงานให้กับ Object นั้นๆ

การวาด Control ลงบน Form

เช่นเราจะวาด Control ชื่อ CommandButton บน Form ซึ่ง Control นี้ลักษณะเป็นรูปปุ่มที่ปรากฏอยู่ในโปรแกรม Windows โดยทั่วไป

1.จาก Form เปล่าที่ปรากฏอยู่บนจอภาพ เลื่อนเม้าส์ไปคลิกที่ Control "CommandButton" ใน Toolbox

2.เลื่อนเม้าส์มายัง Form คลิกค้างไว้พร้อมกับลากเม้าส์แนวทแยงเกิดเป็นกรอบสี่เหลี่ยมประมาณขนาดของปุ่มตามต้องการ แล้วจึงปล่อยเม้าส์ จะปรากฏรูปของปุ่ม Command1 บน Form ดังรูป



ภาพที่ 3.5 แสดงการวาด Control ลงบน Form

3.1.4 การกำหนดคุณสมบัติ (Property) ให้กับ Object

หลังจากที่วาด Control ลงบน Form แล้ว ขั้นตอนต่อไปในการพัฒนาโปรแกรมด้วย Visual Basic ได้แก่ การกำหนดคุณสมบัติ (ต่อไปจะเรียกว่า "Property") ให้กับแต่ละ Object ที่ปรากฏอยู่บน Form ซึ่งแต่ละ Object จะมี Object จะมี Property ทั้งที่เหมือนกับ Object อื่นๆ และ Property

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ประจำตัวที่เป็นส่วนตัวไม่ซ้ำกับ Object ใด การกำหนด Property ให้กับ Object จะกระทำตั้งแต่เริ่มต้นวาด Object ลงบน Form หรืออาจจะเปลี่ยนแปลงโดยการเขียนโปรแกรมก็ได้

การกำหนด Property ให้กับ Object ทำได้ 2 วิธีคือ

1.คลิกที่ Object หรือ Form ที่ต้องการกำหนด Property จะปรากฏจุดสี่เหลี่ยมล้อมรอบ Object นั้น ยกเว้น Form ซึ่งจะมองไม่เห็นจุดดังกล่าว คลิกปุ่มขวาจะปรากฏเมนูขึ้นบนจอภาพ ให้เลือกเมนู Property

2.เลื่อนเม้าส์ไปคลิกยัง Object หรือ Form จนมีจุดล้อมรอบ แล้วกด F4 แทน(ในบางครั้งอาจจะปรากฏจอภาพสำหรับกำหนด Property ของ Form หรือ Object ขึ้นมาบนจอภาพ)

3.1.5 การ Run และเลิกงาน Project

ในการ Run Project ที่พัฒนาขึ้นด้วย Visual Basic สามารถ Run ทั้งโดยใช้ Interpreter และการใช้ Complier กล่าวคือ เราสามารถทดลอง Run สิ่งต่างๆ ที่เราจัดทำขึ้นไปพร้อมๆ กับแก้ไขโปรแกรม จนกระทั่งเสร็จสมบูรณ์ แล้วจึง Compile โปรแกรมให้อยู่ในรูปของ Executed Program เพื่อนำไปใช้งานได้เช่นเดียวกัน

ในการ Run Project ที่เราจัดทำขึ้นนั้น ทำได้ 3 วิธีคือ

1.Run โดยการกด F5

2.คลิกที่ Icon "Run" ใน Toolbar

3.เลือกจากเมนู Run และ Strat ตามลำดับ

สำหรับวิธีเลิกงาน Project ทำได้ 2 วิธีคือ

1.คลิกที่ Icon "End" ใน Toolbar

2.เลือกจากเมนู Run และ End ตามลำดับ

การบันทึก Form และ Project

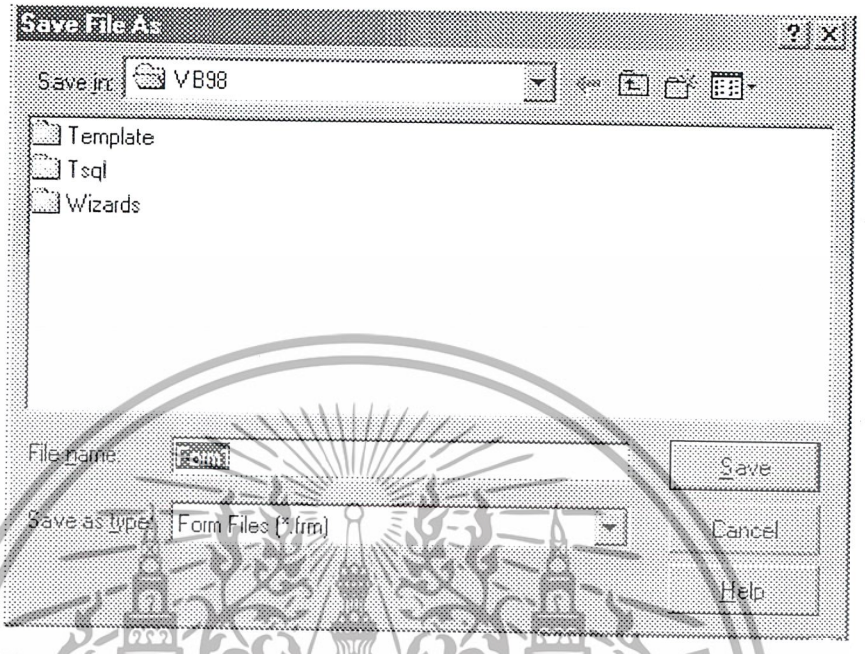
ในการบันทึก Project จะต้องบันทึกทั้งส่วนของ Form และ Project โดย Form จะถูกบันทึกลงในไฟล์นามสกุล FRM ในขณะที่ Project จะถูกบันทึกลงไฟล์นามสกุล VBP ในการบันทึก Form ให้ทำดังนี้

1.คลิกที่ Icon "Save" ใน Toolbar จะปรากฏจอภาพดังรูป

2.ใส่ชื่อ File ที่จะใช้เก็บ Form ซึ่ง Visual Basic จะใช้ชื่อเริ่มต้นเป็นชื่อเดียวกับ Form เสมอ ดังนั้นจึงปรากฏชื่อ Form1 มาให้สามารถเปลี่ยนแปลงชื่อได้ตามต้องการ จากนั้นให้คลิกที่ปุ่ม Save

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จะปรากฏจอภาพสำหรับบันทึก Project ซึ่งก็เช่นเดียวกับ Form จึงปรากฏชื่อ File เป็นชื่อ Project1 ให้เปลี่ยนชื่อตามต้องการแล้วคลิกที่ปุ่ม Save



ภาพที่ 3.6 แสดงการ Save File ของ Visual Basic

3.1.6 Control ภายในของ Visual basic

Visual Basic เป็นเครื่องมือที่ช่วยพัฒนาโปรแกรมแอพพลิเคชั่นสำหรับวินโดว์ตัวแรกที่ประสบความสำเร็จเป็นอย่างมากทั้งนี้เนื่องจากแนวความคิดที่จะนำเอาความสามารถของคอนโทรลมาใช้ในการออกแบบโปรแกรม เพราะคอนโทรลเป็นเครื่องมือที่ช่วยลดความซับซ้อนในการเขียนโค้ดลงไปได้มากทีเดียวและนอกจากนี้คอนโทรลยังมีส่วนที่แสดงผลเพื่อสื่อความหมายของการทำงานระหว่างคอนโทรลและผู้ใช้ได้ดีอีกด้วย ส่วนการใช้งานก็ไม่มีความซับซ้อนเพียงแต่ผู้อ่านทำการเชื่อมต่อคอนโทรลเข้ากับสภาพแวดล้อมของ Visaul Basic จากนั้นก็สามารถที่จะนำมาเพิ่มลงในฟอร์มได้ทันที สำหรับ Visual Basic ได้มีการแบ่งคอนโทรลออกเป็น 4 กลุ่มหลักๆที่สำคัญดังนี้

1.คอนโทรลภายใน (Intrinsic Control) เช่น ComboBox, CommandButton หรือ PictureBox เป็นต้น ซึ่งเป็นคอนโทรลที่ถูกสร้างลงในสภาพแวดล้อมของ Vb.Exe ดังนั้นทุกครั้งที่ผู้อ่านโหลด Visual Basic คอนโทรลเหล่านี้ออกจาก ToolBox ได้เลย ดังนั้นจึงจัดได้ว่าเป็นคอนโทรลมาตราฐาน (Standard Control) กลุ่มหนึ่งของ Visual Basic

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.คอนโทรลมาตราฐาน (Standard Control) เป็นคอนโทรล ActiveX ที่ถูกสร้างเป็นไฟล์ .ocx ที่แยกออกมาต่างหาก เช่น Dbgrid (Apex Data Bound Grid) , MSFlexGrid หรือ CommonDialog เป็นต้น ดังนั้นก่อนที่จะสามารถใช้งานคอนโทรลในกลุ่มนี้ได้เราต้องทำการเชื่อมต่อไฟล์ .ocx เหล่านี้เข้ากับสภาพแวดล้อมของ Visual Basic เสียก่อนโดยคำสั่ง Components ในเมนู Project เช่นเดียวกัน

3.คอนโทรลร่วมวินโดว์ (Windows Common Control) เป็นคอนโทรล ActiveX ที่ถูกสร้างเป็นไฟล์ .ocx ที่ต้องใช้ร่วมไฟล์ .dll ของวินโดว์ เช่น RichTextBox, Slider หรือ Statusbar เป็นต้น เช่นเดียวกับคอนโทรลมาตรฐาน เพียงแต่คอนโทรลในกลุ่มนี้ได้ถูกจัดเป็นคอนโทรลพื้นฐานของวินโดว์ 95 โดยที่คอนโทรลร่วมกับวินโดว์จะถูกจัดเก็บลงในไฟล์ 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 Professinal และ Enterprise เท่านั้น

คุณสมบัติแสดงค่าของคอนโทรลที่สำคัญ

คอนโทรลทั้งหมดที่มากับ Visual Basic ไม่ว่าจะเป็นคอนโทรลภายในหรือ ActiveX จะมีคุณสมบัติตัวหนึ่งที่ถูกนำมาใช้สำหรับการกำหนดค่า (Value) หรืออ่านจากค่าคอนโทรลและคุณสมบัตินี้ได้ถูกกำหนดให้เป็นคุณสมบัติปกติ (Default) ของคอนโทรล โดยในการเขียนโค้ดเราสามารถใช้เพียงชื่อของคอนโทรล (Control Name) โดยไม่ต้องกำหนดคุณสมบัติปกติของทุกๆ คอนโทรลได้โดยไม่เกิดข้อผิดพลาด เช่น คอนโทรล TextBox ก็มีคุณสมบัติ Text เป็นคุณสมบัติปกติของคอนโทรล สำหรับคุณสมบัติปกติของคอนโทรลที่สำคัญมีดังต่อนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 3.1 แสดงคุณสมบัติปกติของคอนโทรลที่สำคัญ

คอนโทรล	คุณสมบัติ
CommandButton	Valve
Data	Caption
Msflexgrid	Text
Label	Caption
TextBox	Text
Timer	Enable

การแบ่งกลุ่มของคอนโทรลภายใน

เราจะแยกตามวัตถุประสงค์ของการใช้งานได้ทั้งหมด 4 กลุ่ม

1.คอนโทรลภายในทั่วไป ประกอบด้วยคอนโทรลที่แสดงผลในลักษณะของการเลือกตอบหรือเลือกรายการ เช่น CheckBox, OptionBox หรือ ListBox เป็นต้น

2.คอนโทรลภายในด้านระบบไฟล์ ประกอบไปด้วยคอนโทรลที่ทำหน้าที่ในการติดต่อหรือ แสดงผลระบบไฟล์ (รวมทั้งไดร์ฟ และ ไดเร็คทอรี่ด้วย) ของวินโดว์ เช่น FileListBox หรือ DirListBox เป็นต้น

3.คอนโทรลภายในด้านกราฟิก ประกอบด้วยคอนโทรลที่ทำหน้าที่ด้านการแสดงผลกราฟิกด้วยวิธีการฟิกคอนโทรล หรือ ฟังก์ชันวินโดว์ API หรือ ไฟล์กราฟิกในรูปแบบต่างๆ เช่น PictureBox, Shape หรือ Image เป็นต้น

4.คอนโทรลภายในด้านเวลา ซึ่งจะมีอยู่คอนโทรลเดียวได้แก่ Timer ซึ่งมีหน้าที่สร้างเหตุการณ์ที่ตอบสนองเป็นครั้งๆตามช่วงเวลาที่ถูกกำหนด

Control มาตราฐานที่ปรากฏอยู่ใน Toolbar จะมีหน้าที่แตกต่างกันไป ซึ่งแต่ละ Control จะมี Property เฉพาะตัว แต่ก็มีบาง Property ที่เหมือนกัน (Property ชื่อเดียวกันจะมีคุณสมบัติเหมือนกัน) เช่น Property “Caption” ของ Object “Label” และ “CommandButton” ต่างก็ใช้สำหรับกำหนดข้อความบน Object เหมือนกัน เป็นต้น

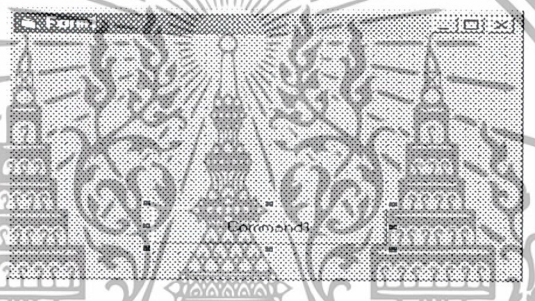
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คอนโทรลภายใน

คอนโทรลภายในก็จะเป็นคอนโทรลพื้นฐานที่ถูกนำไปใช้งานมากที่สุดเพราะจะเป็นกลุ่มคอนโทรลที่ช่วยในการสื่อสารแบบสองทางหรือรับเลือกเงื่อนไขจากผู้ใช้ดังเช่นทุกๆ แอพลิเคชั่นจะใช้คอนโทรล CommandButton

คอนโทรล CommandButton

คอนโทรล CommandButton จะเป็นคอนโทรลที่ถูกนำไปใช้งานมากที่สุดเพราะกำหนดให้ผู้ใช้งานเลือก OK หรือ Cancle นั้น เรามักจะใช้คอนโทรล CommandButton เป็นส่วนใหญ่ ดังนั้นจึงถือว่าเป็นคอนโทรลที่พื้นฐานที่สุดของ Visual Basic เนื่องจากคอนโทรลนี้เป็นปุ่มสำคัญที่ใช้งานในรูปแบบของการคลิกเพื่อยืนยัน ดังนั้นเรียกคอนโทรล CommandButton ได้อีกอย่างว่า PushButton ในขณะออกแบบคอนโทรล CommandButton ที่วางลงบนฟอร์มจะมีลักษณะดังรูป



ภาพที่ 3.7 แสดงคอนโทรล CommandButton ที่วางลงบนฟอร์ม

เราสามารถแก้ไขข้อความที่แสดงผลในคอนโทรลนี้โดยการแก้ข้อความในคุณสมบัติ Caption ของคอนโทรลในหน้าต่างคุณสมบัติหรือแก้ไขโค้ดในแอพพลิเคชั่นก็ได้

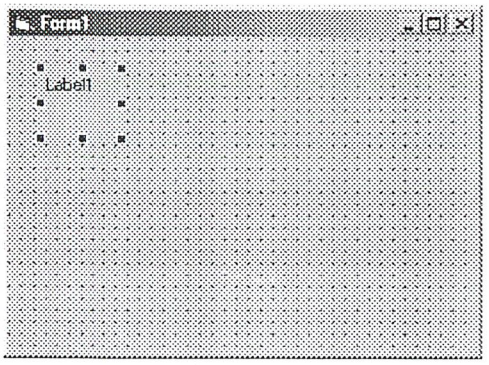
Label

เป็น Control ที่ใช้สำหรับเขียนข้อความบนจอภาพ ซึ่งมี Property ต่างๆที่น่าสนใจดังนี้

ตารางที่ 3.2 Property ของคอนโทรล Label

Property	หน้าที่
Caption	ใช้สำหรับกำหนดข้อความให้แสดงบน Object
Font	ใช้สำหรับกำหนดรูปแบบของตัวอักษรของ Object
ForeColor	ใช้สำหรับกำหนดสีของตัวอักษรของ Object
Alignment	ใช้สำหรับกำหนดรูปแบบในการแสดงผลของข้อความที่กำหนดใน Property "Caption"
BackColor	ใช้สำหรับกำหนดสีฉากหลังของ Object

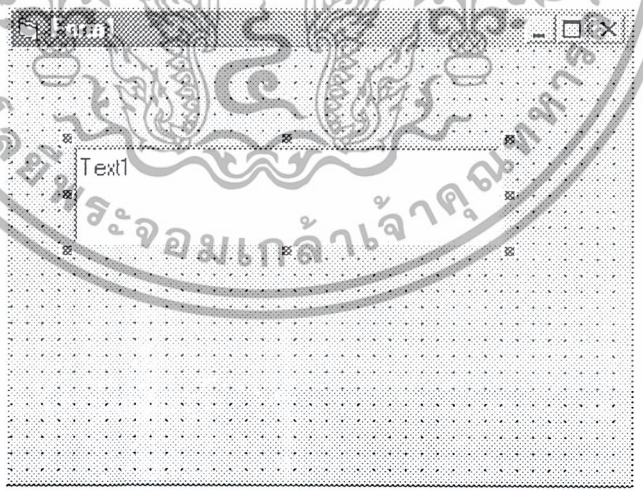
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 3.8 แสดง แสดงคอนโทรล Label ที่วางลงบนฟอร์ม

คอนโทรล TextBox

คอนโทรล TextBox มักจะถูกนำไปใช้ทุกๆฟอร์มที่มีการรับการกรอกข้อความจากผู้ใช้ เนื่องจากคอนโทรลนี้ทำหน้าที่แสดงข้อมูล (โดยผ่านทางคุณสมบัติ Text) ในคอนโทรลและยังอนุญาตให้ผู้ใช้สามารถแก้ไขตัวอักษรต่างๆ ของคุณสมบัตินี้ได้ด้วยเช่นกันนอกจากนี้แล้ว คอนโทรล TextBox ยังได้รวมเอาความสามารถหลายๆด้านของคอนโทรล Label เช่น สามารถแสดงข้อความมากกว่า 1 บรรทัด ความสามารถด้าน DDE (Dynamic Data Exchange) นอกจากนี้ยังสามารถถูกนำไปใช้งานในลักษณะของการกรอกรหัสผ่าน (Password) ได้อีกด้วยดังภาพ

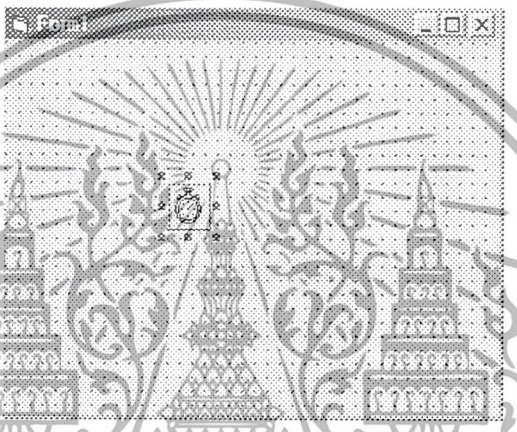


ภาพที่ 3.9 แสดงคอนโทรล TextBox ที่วางลงบนฟอร์ม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คอนโทรล Timer

คอนโทรล Timer เป็นคอนโทรลที่ใช้ในการควบคุมและจัดการเหตุการณ์ด้านเวลา ซึ่งเทียบได้กับประโยค ON TIME GOTO ของ QuickBasic โดยเราสามารถเขียนโค้ดเพื่อทำงานใดๆ เมื่อช่วงเวลาผ่านไปตามค่าที่กำหนด เช่น ทำการปรับการแสดงผลของฟอร์มทุกๆ 1 นาทีเป็นต้น โดยที่คอนโทรลนี้จะตอบสนองเหตุการณ์เพียงเหตุการณ์เดียวเท่านั้น แต่เราสามารถกำหนดให้แต่ละฟอร์มมีคอนโทรล Timer มากกว่า 1 คอนโทรล เนื่องจากคอนโทรล Timer เป็นคอนโทรลที่ทำงานตามนาฬิกาของระบบดังนั้นมันจึงถูกควบคุมโดยตัวของระบบเอง สำหรับวินโดว์ 95 และ NT ในทางปฏิบัติจะไม่มีการจำกัดจำนวนของคอนโทรล Timer ในแต่ละฟอร์มดังนั้นเราจึงสามารถใช้งานคอนโทรล Timer พร้อมๆกันครั้งละหลายๆคอนโทรล ได้อย่างไม่จำกัด



ภาพที่ 3.10 แสดงคอนโทรล Timer ที่วางลงบนฟอร์ม

ในขณะออกแบบคอนโทรล Timer ที่วางลงบนฟอร์มให้กับฟอร์ม ก็จะมีลักษณะดังรูป และเมื่อรันแอพพลิเคชั่น คอนโทรลนี้จะไม่ถูกแสดงผลแต่จะมีการทำให้เกิดเหตุการณ์ Timer ทุกครั้งที่ช่วงเวลาครบตามค่าที่ได้กำหนดให้กับคุณสมบัติ Interval ของคอนโทรล Timer

คอนโทรลด้านฐานข้อมูล

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

1.คอนโทรลด้านฐานข้อมูล (Data Control)

การเข้าถึงฐานข้อมูลด้วยคอนโทรลด้านฐานข้อมูลจะเป็นวิธีที่ง่ายที่สุดในการเขียนโปรแกรมเพราะเราเพียงแต่กำหนดไฟล์ฐานข้อมูลฟิลด์สำหรับแต่ละคอนโทรลและคอนโทรล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

Data เท่านั้น คอนโทรลเหล่านี้จะจัดการสร้าง การแสดงผล การแก้ไขฟิลด์ต่างๆ ของฐานข้อมูลให้โดยอัตโนมัติหรือตามที่ถูกกำหนดในคุณสมบัติต่างๆ สำหรับความสามารถโดยทั่วไปของคอนโทรลด้านฐานข้อมูลมีรายละเอียดดังต่อไปนี้

-สามารถสร้างตัวแปร Recordset โดยอ้างอิงกับตัวแปร Recordset ที่สนับสนุนโดยคอนโทรลด้านฐานข้อมูล

-แก้ไขโครงสร้างของฐานข้อมูล เช่น การแก้ไขตาราง (Table) ฟิลด์ (Field) หรือ ดัชนี (Index) เป็นต้น

-ค้นหาหรือสืบค้นข้อมูลจากฟิลด์ที่ถูกกำหนดของฐานข้อมูล

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

1.Data Acces Object (DAO)

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

-ฐานข้อมูลของ Visual Basic หมายถึง ฐานข้อมูลในรูปแบบของ Microsoft Access (.mdb) ซึ่ง JET สามารถที่จะจัดการ ได้โดยตรง 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 ฐานข้อมูลที่ทำการติดต่อโดยผ่านทาง Open Database Connectivity (ODBC) ซึ่งเป็น มาตรฐานการจัดการกับฐานข้อมูลในรูปแบบที่แตกต่างกันของไมโครซอฟท์ เช่น ฐานข้อมูลชนิด ไคลเอนต์เซิร์ฟเวอร์ Microsoft SQL Server เป็นต้น ดังนั้นเรา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

45843

จึงสามารถที่จะสร้าง แอพพลิเคชั่น ชนิด ไคลเอนต์ เซิร์ฟเวอร์ ด้านฐานข้อมูลได้โดยตรงจาก Visual Basic โดยมาตรฐาน ODBC นี้เอง

คอนโทรล Data

คอนโทรล Data เป็นคอนโทรลที่ใช้ในการเข้าถึงฐานข้อมูลคอมพิวเตอร์และทำการเชื่อมต่อการแสดงผลข้อมูลของแต่ละฟิลด์ในฐานคอมพิวเตอร์ข้อมูลเข้ากับคอนโทรลด้านฐานข้อมูล (Data-Bound or Data-Aware Control) โดยเมื่อมีการเคลื่อนที่ไปยังเรคคอร์ดใดๆในฐานข้อมูลด้วยคอนโทรล Data ข้อมูลที่ถูกแก้ไขในคอนโทรลด้านฐานข้อมูลที่เชื่อมต่อกับคอนโทรล Data ซึ่งเป็นเรคคอร์ดปัจจุบันในขณะนั้นก็จะถูกจัดเก็บลงในฐานข้อมูลโดยอัตโนมัติแล้วจึงเคลื่อนที่ไปยังเรคคอร์ดถัดไปทันที แต่เนื่องจากคอนโทรล Data สามารถที่จะจัดเก็บข้อมูลที่ถูกแก้ไขให้อัตโนมัติ ดังนั้นถ้าหากเราต้องการตรวจสอบความถูกต้องของข้อมูลก่อนที่จะถูกจัดเก็บโดยคอนโทรล Data ก็สามารถกระทำได้โดยการเขียนโค้ดเพื่อตรวจสอบความถูกต้องของข้อมูลในโพรซีเยอร์ของเหตุการณ์ Validate ในขณะออกแบบคอนโทรล Data ที่วางบนฟอร์มจะมีลักษณะดังรูป

ภาพที่ 3.11 แสดงคอนโทรล Data ที่วางลงบนฟอร์ม

คอนโทรล MSComm (Communication)

เนื่องจากในปัจจุบันมีการใช้งานมาตรฐานการเชื่อมต่อแบบ RS 232C กันอย่างแพร่หลาย ซึ่งเป็นมาตรฐานถูกกำนดโดย EIA ซึ่งเป็นองค์กรอุตสาหกรรมอิเล็กทรอนิกส์ ของสหรัฐอเมริกา โดยแบ่งการเชื่อมต่อออกเป็น 2 ลักษณะ คือ DTE (Data Terminal Equipment) และ DCE

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

(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

-รับตัวอักษรจากเครื่องคอมพิวเตอร์

-แปลงตัวอักษรให้เป็นสายข้อมูลแบบบิต (เราเรียกว่าขบวนการ Serialization)

-สร้างเฟรมข้อมูลโดยการเพิ่มบิตที่จำเป็นสำหรับการสื่อสารและการตรวจสอบเช่น บิต Start, Stop และ Parity เป็นต้น

-ส่งผ่านเฟรมข้อมูลที่สร้างขึ้นมาแล้วจากขั้นตอนที่ผ่านมาด้วยความเร็วของโมเด็มหรือพอร์ทอนุกรม (Baud Rate)

-แสดงสถานะความพร้อมที่จะรับข้อมูลตัวอักษรถัดไปให้กับเครื่องคอมพิวเตอร์ในการรับข้อมูล (Data Receiver)

-รับตัวอักษรจากตัวอินเตอร์เฟส

-ตรวจสอบความถูกต้องของเฟรมข้อมูลตามมาตรฐานเฟรมที่กำหนดโดยถ้าหากเฟรมข้อมูลมีรูปแบบที่ไม่ถูกต้องก็จะมีการแจ้งข้อผิดพลาดทันที

-ตรวจสอบความถูกต้องของพาริตี้

-แปลงสายข้อมูลแบบบิตให้เป็นตัวอักษร

-ส่งสายตัวอักษรให้กับเครื่องคอมพิวเตอร์

-แสดงสถานะความพร้อมที่รับข้อมูลตัวอักษรถัดไปให้กับอินเตอร์เฟส

คอนโทรล MSComm เป็นคอนโทรลตัวหนึ่งที่ช่วยในการติดต่อกับพอร์ทอนุกรม (Serial Port) ซึ่งสามารถรับส่งข้อมูลผ่านทางพอร์ทอนุกรมได้ด้วยคอนโทรลนี้ เช่น การติดต่อผ่านทางโมเด็ม หรือ ติดต่อโดยตรงกับบอร์ดอิเล็กทรอนิกส์ เป็นต้น ซึ่งคอนโทรล MSComm ที่มากับ Visual Basic จะเป็นคอนโทรลที่ทำงานโดยมีการตอบสนองต่อเหตุการณ์แบบ Event-Driven นั่นคือ คอนโทรลจะทำหน้าที่ตรวจสอบการเกิดขึ้นหรือร้องขอให้เกิดเหตุการณ์ต่างๆกับพอร์ทอนุกรมโดยอัตโนมัติและจะมีการแจ้งเตือนให้ผู้อ่านได้รับทราบโดยผ่านโพรซีเยอร์เหตุการณ์ เช่นเดียวกับคอนโทรลทั่วไปของ Visual Basic นั่นเอง ดังนั้นในการเขียนโค้ดเราจึงไม่จำเป็นต้องสร้าง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โพรซีเยอร์ที่ทำหน้าที่คอยตรวจสอบเหตุการณ์ต่างๆ ของพอร์ทอนุกรมซึ่งจะทำให้ง่ายต่อการทำงานเป็นอย่างมากคอนโทรล MSComm จะมีหน้าที่สำหรับการสื่อสารผ่านพอร์ทอนุกรม 3 ประการ ดังต่อไปนี้

-หมุนหมายเลขติดต่อกับโทรศัพท์ปลายทางที่กำหนด

-ตรวจสอบการเข้ามาของข้อมูลยังพอร์ทอนุกรมโดยอัตโนมัติ

-ส่งข้อมูลตามที่กำหนดจากโปรแกรมไปยังพอร์ทอนุกรม

ในความจริงคอนโทรล MSComm ไม่ได้ทำหน้าที่ติดต่อกับพอร์ทอนุกรมโดยตรงแต่มันจะทำหน้าที่เรียกใช้ฟังก์ชั่นวินโดว์ API ซึ่งวินโดว์จะทำการส่งหรือรับข้อมูลผ่านพอร์ทอนุกรมโดยอาศัยไดร์วเวอร์ Comm.drv อีกทอดหนึ่ง ดังนั้นจึงสามารถสรุปสั้นๆ ได้ว่าทุกครั้งที่เรามีการเรียกใช้คอนโทรล MSComm ก็หมายถึงเรียกใช้ฟังก์ชั่นวินโดว์ API ซึ่งจะถูกตีความอีกทอดหนึ่งโดยไดร์เวอร์ Comm.drv จากนั้นก็จะส่งผ่านข้อมูลที่ถูกจัดรูปแบบตามมาตรฐานการสื่อสาร (ทั้งนี้ขึ้นกับอุปกรณ์ที่ต่อเข้ากับพอร์ทอนุกรม) ให้กับดีไวซ์ไดรเวอร์อีกทอดหนึ่งนั่นเอง

การกำหนดคุณสมบัติของคอนโทรล MSComm ในขณะออกแบบเราสามารถกระทำได้อย่างสะดวกโดยการคลิกไปที่ปุ่มของรายการ (Custom) ในหน้าต่างคุณสมบัติซึ่งก็จะปรากฏไดอะล็อกบ็อกซ์ Property Pages เพื่อให้เราได้ปรับแต่งค่าของคุณสมบัติของคอนโทรล MSComm สนับสนุน ซึ่งปุ่มต่างๆของไดอะล็อกบ็อกซ์ Property Pages มีความหมายดังนี้

-ปุ่มคำสั่ง OK ยอมรับการแก้ไขคุณสมบัติของคอนโทรล MSComm

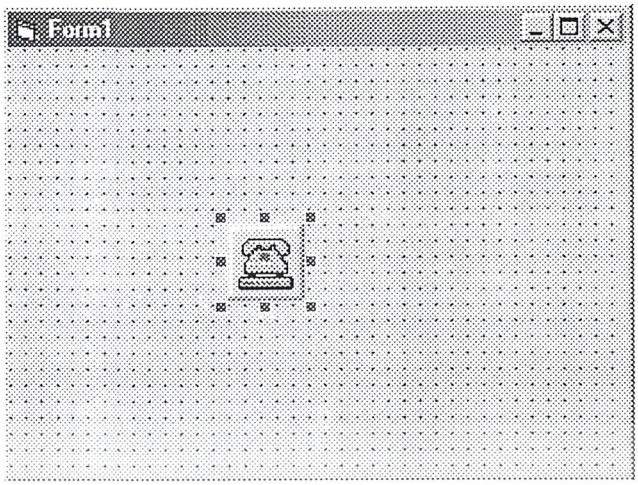
-ปุ่มคำสั่ง Cancel ยกเลิกการแก้ไขคุณสมบัติของคอนโทรล MSComm

-ปุ่มคำสั่ง Apply อัปเดตคุณสมบัติที่ถูกแก้ไขของคอนโทรล MSComm

-ปุ่มคำสั่ง Help แสดงผล Help ของคอนโทรล MSComm

สำหรับฟอร์มหนึ่งๆเราสามารถเพิ่มได้หลายๆคอนโทรล MSComm ทั้งนี้ขึ้นอยู่กับความต้องการของเราในการติดต่อกับพอร์ทอนุกรมใดบ้าง สำหรับวินโดว์ 95 และ NT 4.0 เราสามารถติดตั้งพอร์ทอนุกรมได้มากกว่า 4 พอร์ท โดยเราสามารถเพิ่มคอนโทรล MSComm ลงในฟอร์มก้จะปรากฏดังรูป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

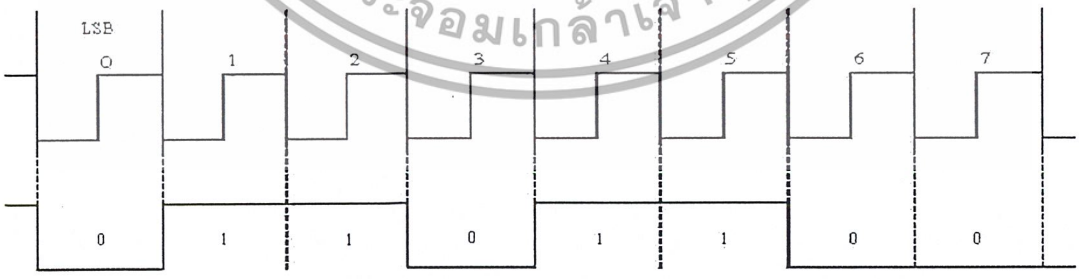


ภาพที่ 3.12 แสดงคอนโทรล MScomm ที่วางลงบนฟอร์ม

3.2 การสื่อสารแบบอนุกรม

การสื่อสารแบบอนุกรมนั้นจะแบ่งออกได้เป็น 2 แบบคือการสื่อสารอนุกรมแบบซิงโครนัส และการสื่อสารอนุกรมแบบอะซิงโครนัส การสื่อสารแบบซิงโครนัสจะมีสัญญาณนาฬิการ่วมอยู่กับการรับและส่งสัญญาณด้วย ตัวอย่างการส่งข้อมูลแบบซิงโครนัสก็คือคีย์บอร์ดของคอมพิวเตอร์ ซึ่งสายเส้นหนึ่งจะเป็นสายของนาฬิกา ส่วนสายอีกเส้นจะเป็นสายของข้อมูล ดังนั้นการติดต่อกันแบบซิงโครนัสนี้จะต้องใช้สายในการเชื่อมต่ออย่างน้อยที่สุด 3 เส้นคือ สัญญาณนาฬิการ, ข้อมูล และกราวด์ รูปที่ 1-1 แสดงให้เห็นถึงไทมิ่งไดอะแกรมของการส่งข้อมูลแบบซิงโครนัส

การสื่อสารข้อมูลแบบอะซิงโครนัส



ภาพที่ 3.13 แสดงรูปแบบอย่างง่ายที่สุดของข้อมูลอนุกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.1 การสื่อสารข้อมูลแบบอะซิงโครนัส

การสื่อสารข้อมูลแบบอะซิงโครนัสคือการรับและส่งข้อมูลไปในสายโดยไม่จำเป็นต้องมีสัญญาณนาฬิการ่วมด้วยเหมือนกับการรับส่งข้อมูลแบบซิงโครนัส แต่จะใช้การกำหนดค่าสัญญาณนาฬิการทั้งภาครับและภาคส่งให้มีค่าเท่ากัน ซึ่งเรียกสัญญาณนาฬิกาที่ใช้ในการกำหนดค่าให้ภาครับ และภาคส่งนี้ว่า อัตราการถ่ายทอดข้อมูล หรือ บอดเรต (baudrate) มีหน่วยเป็น บิตต่อวินาที (bit per secone: bps)

รูปแบบของข้อมูลที่ใช้ในการรับส่งแบบอะซิงโครนัสประกอบด้วย 4 ส่วนด้วยกันคือ

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

รูปที่ 1-2 แสดงรูปแบบของข้อมูลอนุกรมแบบอะซิงโครนัส ซึ่งเมื่อไม่มีข้อมูลที่จะส่ง ขา DATA จะมีสถานะลอจิก "1" ซึ่งจะเรียกสถานะนี้ว่าสถานะหยุดรอ (waiting stage) การเริ่มต้นจากนั้นบิตข้อมูลจะถูกส่งออกไป โดยเริ่มจากบิตที่มีนัยสำคัญต่ำที่สุด (LSB) ก่อน ซึ่งข้อมูลในไบต์ที่จะส่งอาจจะมีจำนวนบิต 5,6,7 หรือ 8 บิตก็ได้ จากนั้นจะตามด้วยบิตพาริตี้ ซึ่งใช้เพื่อตรวจสอบความผิดพลาดที่เกอดขึ้นจากการส่งข้อมูล บิตสุดท้ายที่จะส่งคือบิตปิดท้าย ซึ่งจะใช้ขาดาต้ามีสถานะลอจิก 1 อีกครั้งด้วยระยะเวลาอย่างน้อย 1 บิต, 1.5 บิต หรือ 2 บิต เพื่อเป็นการแสดงว่าสิ้นสุดข้อมูลแล้ว

ภาพที่ 3.14 รูปแบบอย่างง่ายที่สุดของข้อมูลอนุกรมแบบอะรสื่อสารข้อมูลแบบอะซิงโครนัส

อุปกรณ์พิเศษที่ได้รับการออกแบบมาสำหรับการรับและส่งข้อมูลแบบอะรสื่อสารข้อมูลแบบอะซิงโครนัสเรียกว่า 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 ซึ่งการรับส่งแบบอนุกรมโดยไม่ผ่านโมเด็มอาจจะสามารถกำหนดค่าบอดเรตได้สูงถึง 115200 บิตต่อวินาที เนื่องจากบอดเรตคือจำนวนบิตของข้อมูลที่สามารถถ่ายทอดได้ภายใน 1 วินาที ยกตัวอย่าง ข้อมูลอนุกรมถูกส่งในลักษณะ 8 บิต ไม่มี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การตรวจสอบพาริตี้ มีบิตเริ่มต้น 1 บิต และบิตปิดท้าย 1 บิต ความยาวของข้อมูลที่รับส่งนี้เท่ากับ 10 บิต ถ้าใช้บอดเรดในการส่งข้อมูลเท่ากับ 9600บิตต่อวินาที ก็จะสามารถรับส่งข้อมูลได้ด้วยความเร็ว 960 ไบต์ต่อวินาที และถ้ามีการใช้พาริตี้ความเร็วในการรับส่งข้อมูลจะเหลือเป็น 872 ไบต์ต่อวินาที

3.2.2 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 จะต้องเป็นอุปกรณ์ที่มีการประมวลผลในตัวเช่น ไมโครคอนโทรลเลอร์หรือไมโครคอมพิวเตอร์ ซึ่งมีความสามารถในการสร้างบิตข้อมูลแบบอนุกรมได้ ส่วนอุปกรณ์ DCE จะทำหน้าที่เป็นเพียงตัวรับข้อมูลที่ส่งมาจาก DTE เท่านั้น โดยการรับส่งข้อมูลระหว่างอุปกรณ์ทั้งสองจะกระทำผ่านมาตรฐาน RS-232

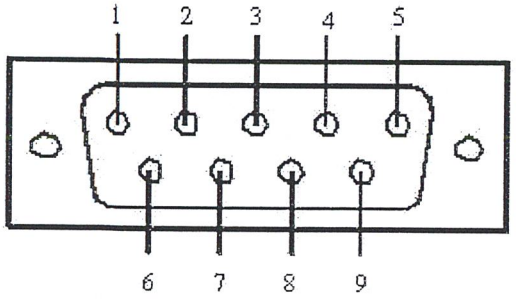
ข้อแตกต่างของอุปกรณ์ DTE และอุปกรณ์ DCE อย่างหนึ่งที่เห็นได้ชัดคือ คอนเน็กเตอร์ของ DTE จะเป็นตัวผู้ ส่วนคอนเน็กเตอร์ของ DCE จะเป็นตัวเมีย ซึ่งพอร์ตอนุกรมของคอมพิวเตอร์ที่ใช้กันอยู่ทั่วไปจะเป็นแบบ DTE ส่วนคอนเน็กเตอร์ที่อยู่ที่โมเด็มจะเป็นแบบ DCE

สำหรับการใช้งานบนคอมพิวเตอร์ พอร์ตอนุกรม RS-232 มักถูกใช้เชื่อมต่อกับโมเด็มหรือเมาส์ โดยสามารถรับส่งข้อมูลได้ที่ความยาวของสายสัญญาณสูงสุดถึง 20 เมตร

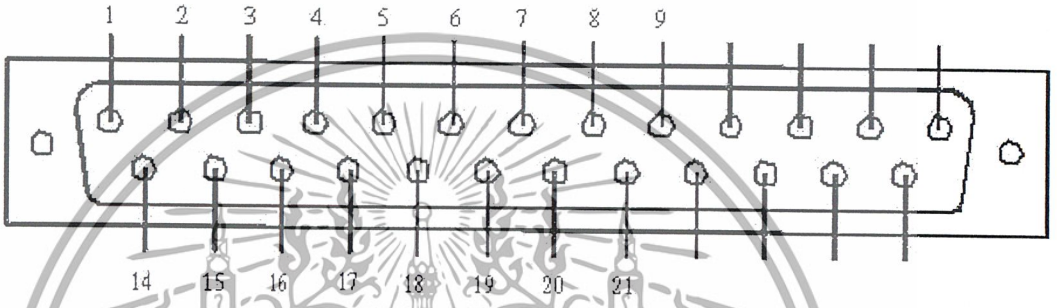
3.2.3 คอนเน็กเตอร์สำหรับพอร์ต RS-232 และการเชื่อมต่อ

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

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 3.15 แสดงคอนเน็กเตอร์อนุกรม 9 ขาหรือแบบ DB-9 (มองจากด้านหลังคอมพิวเตอร์)



ภาพที่ 3.16 แสดงคอนเน็กเตอร์อนุกรม 25 ขาหรือแบบ DB-25 (มองจากด้านหลังคอมพิวเตอร์)

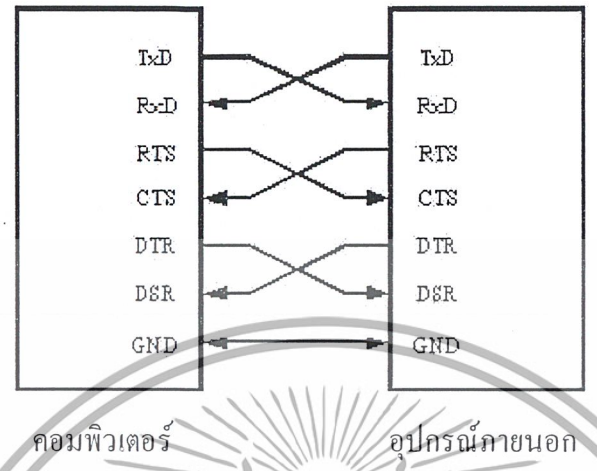
ตารางที่ 3.3 เปรียบเทียบ DB-9 และ DB-25

คอนเน็กเตอร์ DB-9	คอนเน็กเตอร์ DB-25	ชื่อของสายสัญญาณ	ชนิดของสาสัญญาณ
1	8	DataCarrierDetect:DCD	อินพุต
2	3	Received Data : RxD	อินพุต
3	2	Transmitted Data : TxD	เอาต์พุต
4	20	DataTerminalReady: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

สำหรับการเชื่อมต่อคอมพิวเตอร์กับอุปกรณ์ภายนอกแสดงดังในรูป 3.17 ลูกศรในรูปแสดงถึงทิศทางของข้อมูล ในรูป เป็นการเชื่อมต่อแบบ Null modem หรือการเชื่อมต่อโดยตรงโดยไม่ต้องผ่านโมเด็ม โดยมีการตรวจสอบหรือแฮนด์เช็กเต็มรูปแบบ ส่วนในรูปที่ 3.18 เป็นการเชื่อมต่อแบบ Null modem ในลักษณะที่ใช้สายสัญญาณเพียง 3 เส้น โดยเส้นหนึ่งสำหรับส่งข้อมูล อีกเส้นสำหรับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รับข้อมูล และเส้นสุดท้ายเป็นกราวด์ สำหรับรายละเอียดหน้าที่การทำงานในและขาของพอร์ตอนุกรม RS-232 มีดังนี้



ภาพที่ 3.17 แสดงการต่ออุปกรณ์ภายนอกเข้ากับคอมพิวเตอร์แบบ Null modem

ภาพที่ 3.18 แสดงการต่ออุปกรณ์ภายนอกเข้ากับคอมพิวเตอร์แบบ RS-232 โดยใช้สายสัญญาณเพียง 3 เส้น

-Data Carrier Detect: DCD หรืออาจเรียกว่า Carrier Detect:: CD ขานี้จะแอกตีฟเมื่อมีการส่งสัญญาณพาห์จากอุปกรณ์สื่อสารข้อมูล เชน โมเด็ม สำหรับการใช้งานปกติ นี้จะไม่ได้ถูกใช้งานมากนัก

-Receive Data: RD หรือRxD ขานี้ใช้เพื่อรับสัญญาณอนุกรมเข้ามายังคอมพิวเตอร์โดยนำข้อมูลที่อ่านได้เก็บไว้ในรีจิตเตอร์ บัฟเฟอร์

-Transmitted Data: TD หรือ TxDขานี้ใช้เพื่อส่งข้อมูลออกจากคอมพิวเตอร์ โดยนำข้อมูลที่เก็บอยู่ในบัฟเฟอร์สำหรับส่งข้อมูลส่งออกไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

-Data Treminal Ready: DTR เป็นขาสัญญาณที่ส่งออกจากคอมพิวเตอร์เพื่อให้อุปกรณ์ปลายทางรับรู้ว่า ต้องการติดต่อด้วย โดยขา DTR นี้จะต้องเชื่อมต่อกับขา DSR ของอุปกรณ์ปลายทาง และขา DTR ของอุปกรณ์ปลายทางจะต้องเชื่อมต่อกับขา DSR องคอมพิวเตอร์ ถ้าใช้การเชื่อมต่อเป็นแบบ Null Modem ซึ่งใช้สายในการเชื่อมต่อเพียง 3 เส้น จะต้องต่อขา DTR และ DSR ของตัวมันเองเข้าด้วยกันและต้องต่อกับขา DCDด้วยในกรณีที่โปรแกรมสื่อสารที่ใช้มีการตรวจจับสัญญาณพาห์

-Signal Ground: GND ขากราวด์ของระบบ

-Data Set Ready: DSR ขานั้นจะใช้คู่กับขา DTR เพื่อตรวจสอบการเชื่อมต่อกันระหว่างคอมพิวเตอร์กับอุปกรณ์ปลายทาง ซึ่งา DSR นี้จะเป็นขาสำหรับรับข้อมูลจากภาจนอกซึ่งถูกส่งมาจากขา DTR

-Request To Send: RTS เป็นาสำหรับส่งสัญญาณร้องขอให้ทางอุปกรณ์ปลาทางส่งข้อมูลกลับมายังคอมพิวเตอร์ โดยขาที่รับสัญญาณ RTS ก็คือขา CTS ในกรณีที่ใช้การเชื่อมต่อ Null modem 3 สาย จะต้องเชื่อมต่อขา RTS และ CTS ของตัวมันเองเข้าด้วยกัน เพื่อจะให้การรับและส่งข้อมูลสามารถเกิดขึ้นได้ตลอดเวลา

-Clear To Send: CTS ขานี้จะคอยรับสัญญาณจากขา RTS เมื่อรับสัญญาณได้ ข้อมูลที่ขา TxD จะถูกส่งออกไป ดังนั้นขานี้จึงถูกใช้เพื่อตรวจสอบอุปกรณ์ต่อพ่วงว่าพร้อมที่จะรับข้อมูลหรือไม่

-Ring Indicator: RI ใช้แสดงสถานะสัญญาณเรียกจากสายโรศัพท์ ปกติในการสื่อสารโดยทั่วไปสายนี้จะไม่ถูกใช้งาน จะใช้งานก็ต่อเมื่อมีการเชื่อมต่อกับโมเด็มและโปรแกรมมีการตรจสอบสัญญาณนี้เท่านั้น

3.2.4 UART

UART มาจากคำว่า Universal Asynchronous Receiver Transmitter ซึ่งหมายถึงอุปกรณ์ที่หน้าที่รับและส่งข้อมูลแบบอะซิงโครนัสนั่นเอง สำหรับการสื่อสารอนุกรมบนคอมพิวเตอร์แล้ว UART ถือว่าเป็นหัวใจสำคัญของการสื่อสารอนุกรม

หน้าที่หลักของ UART คือทำหน้าที่แปลงข้อมูลที่อยู่ในรูปแบบขนานจากคอมพิวเตอร์ให้อยู่ในรูปแบบอนุกรมแบบอะซิงโครนัส แล้วส่งออกไป และทำหน้าที่แปลงสัญญาณอนุกรมแบบอะซิงโครนัสที่ป้อนเข้ามายัง UART ให้เป็นแบบขนานก่อนที่จะส่งเข้าสู่คอมพิวเตอร์ ซึ่งนอกจาก UART จะส่งข้อมูลไปยังคอมพิวเตอร์แล้ว ยังแจ้งข้อมูลอื่นๆ ให้คอมพิวเตอร์รับทราบด้วย เช่น อัตราเร็วในการรับส่งข้อมูล (ผิดพลาดจากพาริตี้, เฟรมข้อมูล, โอเวอร์รัน) เป็นต้น

ภายใน UART จะมีส่วนของวงจรสร้างบอดเรตแบบโปรแกรมได้ (programmable buadrate generator) โดยการกำหนดค่าตัวหารให้กับสัญญาณนาฬิกาของ UART โดยตัวหารนี้มีขนาด 16 บิต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ดังนั้นจึงสามารถกำหนดตัวหารอยู่ในช่วง 1- 56,535 UART สามารถรับส่งข้อมูลได้ทั้งแบบฮาล์ฟดูเพล็กซ์ (half duplex) และฟูลดูเพล็กซ์ (full duplex) โดยการส่งแบบฮาล์ฟดูเพล็กซ์เป็นการส่งแบบทิศทางเดียว ส่วนการส่งแบบฟูลดูเพล็กซ์นั้นสามารถรับและส่งข้อมูลได้ในคราวเดียวกัน

ในเครื่องคอมพิวเตอร์ทั่วไปมี UART ที่ใช้งานกันอยู่ 2 เบอร์คือ 8250 ซึ่งเป็น UART มาตรฐานที่มีใช้กันมายานาน UART เบอร์นี้จะมีบัฟเฟอร์สำหรับรับและส่งข้อมูลตำแหน่งเดียวกัน ทำให้การรับและส่งข้อมูลถูกจำกัดความเร็วอยู่ที่ 57.6 กิโลบิตต่อวินาทีเท่านั้น แต่UART เบอร์นี้ก็ถือว่าเป็นต้นแบบของ UART ที่ใช้ในคอมพิวเตอร์โดยคอมพิวเตอร์ทุกๆ รุ่นจะต้องสนับสนุนการทำงานตามรูปแบบของ UART เบอร์นี้

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

อย่างไรก็ตาม ความเร็วในการส่งข้อมูลที่มากมายของ UART เบอร์ใหม่ๆ ก็ไม่ได้ช่วยให้การรับส่งข้อมูลของคอมพิวเตอร์เร็วขึ้น เนื่องจากว่าคอมพิวเตอร์ยังใช้ความถี่ของสัญญาณนาฬิกาในการแปลงข้อมูลเพียง 1.8432 MHz เท่านั้น

3.2.5 รีจิสเตอร์ของพอร์ตอนุกรม RS-232

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

ในรูปที่ 1-5 แสดงไดอะแกรมการทำงานภายในของพอร์ตอนุกรมซึ่งประกอบไปด้วยรีจิสเตอร์ขนาด 8 บิต 8 ตัวที่ใช้งานร่วมกับ UART แอดเดรสของรีจิสเตอร์ภายในพอร์ตอนุกรมสามารถคำนวนได้จากค่ารีจิสเตอร์พื้นฐานของพอร์ตอนุกรม ยกตัวอย่าง พอร์ตอนุกรม COM1 มีแอดเดรสอยู่ที่ 3F8H ตำแหน่งของรีจิสเตอร์ต่างๆ จะเป็นตำแหน่งที่บวกเข้าไปกับค่า 3F8H โดยรีจิสเตอร์ที่ใช้งานกับพอร์ตอนุกรมมีดังนี้

00H เป็นรีจิสเตอร์บัฟเฟอร์สำหรับเก็บข้อมูลรับเข้ามาหรือเตรียมข้อมูลก่อนที่จะส่งออกไป

01H รีจิสเตอร์อีนาเบิลการอินเตอร์รัปต์ ใช้ในการเซตโหมดการอินเตอร์รัปต์ของพอร์ตอนุกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

02H รีจิสเตอร์แสดงโหมดการอินเตอร์รัปต์ ใช้เพื่อตรวจอบโหมดการอินเตอร์รัปต์เมื่อมีการอินเตอร์รัปต์เกิดขึ้น

03H รีจิสเตอร์กำหนดรูปแบบของข้อมูล

04H รีจิสเตอร์ควาบคุมโมเด็ม ใช้ตรวจสอบิตสำหรับติดต่อกับโมเด็ม เช่น RTS หรือ DTR

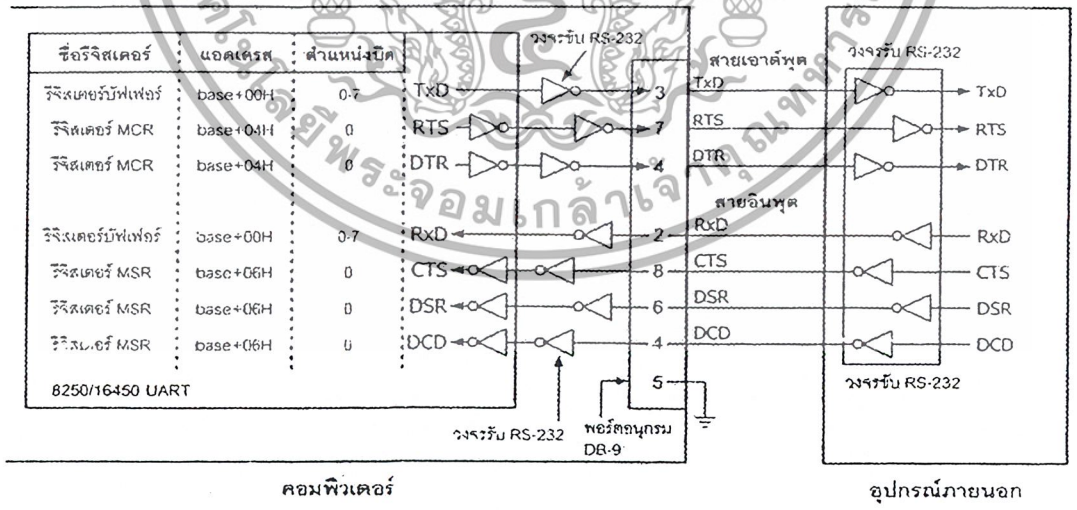
05H รีจิสเตอร์แสดงสถานะการรับและการส่งข้อมูลแบอนุกรม

06H รีจิสเตอร์แสดงสถานะของโมเด็ม ซึ่งจะแสดงสถานะของขา DCD, Ri, DSR และ CTS

07H รีจิสเตอร์สำหรับการเก็บข้อมูลชั่วคราว

ลักษณะสัญญาณอินพุตและเอาต์พุตของพอร์ต RS-232

สัญญาณเอาต์พุตที่ใช้ควบคุม (RTS และ DTR) และสัญญาณแสดงสถานะอินพุต (CTS, DSR และ DCD) ของพอร์ตอนุกรม RS-232 จะถูกกลับสถานะภายในตัว UART ส่วนสัญญาณข้อมูลทั้งภาคส่งและรับจะไม่ถูกกลับสถานะ UART จะให้ระดับสัญญาณเอาต์พุตออกมาเป็นแบบทีทีแอล เท่านั้น ดังนั้นเมื่อสัญญาณถูกส่งออกมาจาก UART จึงต้องส่งเข้าสู่วงจรขับเพื่อปรับระดับแรดันให้ได้ระดับสัญญาณเป็นไปตามมาตรฐาน RS-232 ก่อนส่งออก ไปจากคอมพิวเตอร์สำหรับอุปกรณ์ต่อเชื่อมปลายทางก็จะต้องมีวงจรขับในลักษณะนี้เช่นเดียวกัน เพื่อให้ได้ระดับสัญญาณในระดับเดียวกัน แต่วงจรขับที่ใช้ทั้งภายในคอมพิวเตอร์และอุปกรณ์ต่อเชื่อมปลายทางนั้นจะถูกกลับสถานะ ดังแสดงเป็นบล็อกไดอะแกรม



ภาพที่ 3.19 แสดงบล็อกไดอะแกรมสัญญาณอินพุตและเอาต์พุตของพอร์ต RS-232

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.6 แอดเดรสของพอร์ตอนุกรม

แอดเดรสพื้นฐานของพอร์ตอนุกรมมี 4 ตำแหน่งดังนี้คือ

รูปที่ 1-6 ไดอะแกรมแสดงโครงสร้างทางฮาร์ดแวร์ของพอร์ตอนุกรม

COM1: 3F8H

COM2: 2F8H

COM3: 3E8H

COM4: 2E8H

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

COM2 = 0000:0402H – 0000:0403H

COM3 = 0000:0404H – 0000:0405H

COM4 = 0000:0406H – 0000:0407H

นอกจากนี้ที่หน่วยความจำแอดเดรส 0000:0411H ยังใช้สำหรับแสดงจำนวนของพอร์ตอนุกรมที่มีใช้อยู่ในคอมพิวเตอร์อีกด้วย โดยมีรายละเอียดดังแสดงในตารางที่ 1-2

ตาราง 3.4 แสดงจำนวนของพอร์ตอนุกรมที่มีใช้อยู่ในคอมพิวเตอร์

บิต 3	บิต 2	บิต 1	จำนวนพอร์ท
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

3.2.7 ระดับแรงดันที่ใช้งานสำหรับพอร์ตอนุกรม RS-232

มาตรฐานการสื่อสารข้อมูลของพอร์ตอนุกรม ได้ระบุช่วงระดับแรงดันสำหรับการทำงานของพอร์ตอนุกรมไว้ว่า ที่ลอจิก "0" จะมีระดับสัญญาณ +3 ถึง +15V ส่วนลอจิก"1" จะมีระดับสัญญาณ –3 ถึง –15V ระดับสัญญาณนี้ทำให้ไม่สามารถที่จะนำขาเอาต์พุตใดๆ ต่อเข้ากับลอจิกเกตเพื่อใช้งานได้โดยตรง จะต้องผ่านวงจรเพื่อเปลี่ยนระดับแรงดันเสียก่อน โดยปกติจะใช้ไอซีจำพวก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

RS-232 transceiver ที่สิยมมากคือ MAX232 หรือ ICL232 ไปซีในกลุ่มนี้จะทำหน้าที่แปลงระดับแรงดันของ RS-232 ให้อยู่นะดับ ทีทีแอล โดยลอจิก “0” ซึ่งเดิมมีระดับสัญญาณ +3 ถึง +15V จะถูกแปลงเป็น 0V ส่วนลอจิก “1” ซึ่งมีระดับสัญญาณ –3 ถึง –15V จะแปลงเป็น +5V ทั้งนี้เพื่อให้สามารถเชื่อมต่อกับอุปกรณ์ดิจิตอลอื่นที่ใช้ระดับแรงดัน ทีทีแอล ได้

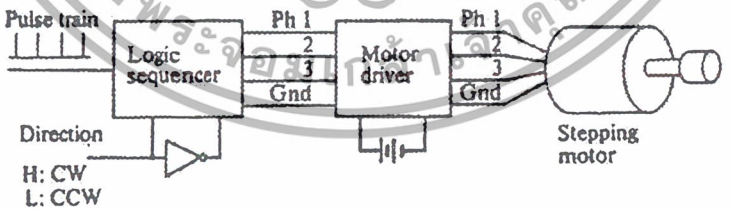
3.3 สเต็ปปิ้งมอเตอร์

สเต็ปปิ้งมอเตอร์ แบ่งออกได้เป็นหลายชนิดด้วยกัน เช่น สเต็ปปิ้งมอเตอร์แบบแปรค่ารีลัคแตนซ์ (Variable-reluctance), สเต็ปปิ้งมอเตอร์แบบแม่เหล็กถาวร (Permanent magnet), สเต็ปปิ้งมอเตอร์แบบไฮบริด (Hybrid) และสเต็ปปิ้งมอเตอร์แบบลิเนียร์ (Linear) เป็นต้น แต่ที่ใช้งานกันบ่อยส่วนใหญ่เป็นแบบสเต็ปปิ้งมอเตอร์แบบแปรค่ารีลัคแตนซ์ และ , สเต็ปปิ้งมอเตอร์แบบไฮบริด

สเต็ปปิ้งมอเตอร์มีความแตกต่างจากมอเตอร์โดยทั่วๆ ไป คือเมื่อป้อนไฟฟ้าให้กับมันๆ จะหมุนเพียงเล็กน้อยตามเส้นรอบวง ซึ่งแตกต่างจากมอเตอร์โดยทั่วไป ที่จะหมุนโดยทันที และตลอดเวลาที่จ่ายไฟฟ้า , สเต็ปปิ้งมอเตอร์ยังสามารถกำหนดตำแหน่งอการหมุนด้วยตัวเลขได้อย่างละเอียดโดยการใช้คอมพิวเตอร์เป็นตัวกำหนดและ จัดเก็บตัวเลขเหล่านั้นไว้

3.3.1 หลักการทำงานของสเต็ปปิ้งมอเตอร์

หลักการทำงานของสเต็ปปิ้งมอเตอร์ แสดงได้ดังรูปที่ 1 ซึ่งประกอบด้วยวงจรซีเควนซ์ วงจรขับสัญญาณความถี่อินพุท และสัญญาณควบคุมทิศทางการหมุนคอนโทรลสเต็ปเปอร์ของสเต็ปปิ้งมอเตอร์จะอินเตอร์เฟสได้โดยตรงกับไมโครโปรเซสซอร์ หรือ ไมโครคอมพิวเตอร์



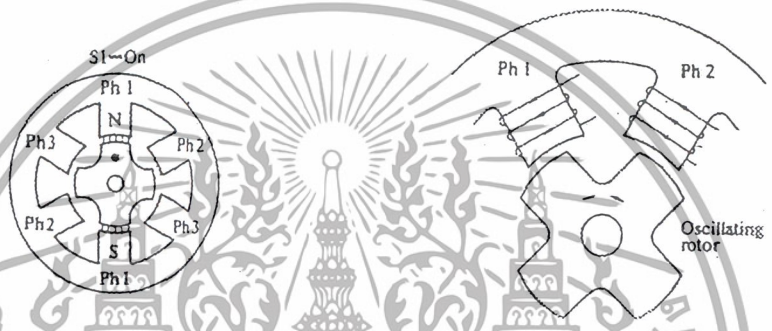
ภาพที่ 3.20 แสดงคอนโทรลเลอร์ สำหรับควบคุมการทำงานของสเต็ปปิ้งมอเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3.2 โครงสร้างและหลักการพื้นฐานของสเต็ปปิ้งมอเตอร์

โครงสร้างสเต็ปปิ้งมอเตอร์

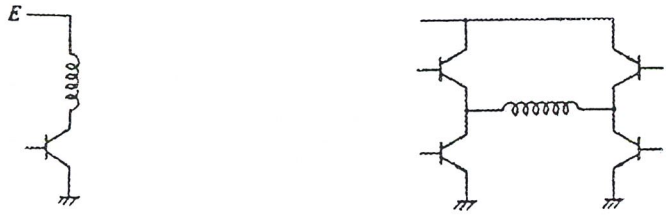
มอเตอร์โดยทั่วไปการที่จะทำให้เกิดการหมุนของโรเตอร์ (Rotor) ได้ต้องมีการกระทำของสนามแม่เหล็กที่เกิดขึ้นระหว่างโรเตอร์และสเตเตอร์ (Stator) ซึ่งขึ้นอยู่กับการจัดวางขั้วแม่เหล็ก (Pole) โครงสร้างของขั้วแม่เหล็กบนสเตเตอร์ประกอบขึ้นจากแท่นเหล็กวงแหวนที่มีซี่ยื่นออกมา แต่ละซี่เหล่านั้นจะมีคอยล์พันสวมอยู่ ดังนั้น เมื่อป้อนกระแสไฟฟ้าผ่านคอยล์ทำให้เกิดสนามแม่เหล็กไฟฟ้า (Electromagnetic) ขึ้นด้านตรงข้ามของแต่ละขั้วแม่เหล็กจะได้รับกระแสไฟฟ้าในขณะเดียวกัน แต่ว่าจะไหลวนในทิศทางตรงข้ามทำให้เกิดสนามแม่เหล็กไฟฟ้าในทิศทางตรงกันข้ามขึ้น ดังรูปที่ 3.21 (ก)



รูปที่ 3.21 (ก) แสดงสเต็ปปิ้งมอเตอร์ที่มีการต่อวงจรขดลวดภายในเพื่อกระตุ้นให้เกิดขั้วแม่เหล็กขึ้น 1 ขั้วในทิศทางตรงกันข้าม ส่วนขดลวดอื่นๆ จะไม่ถูกกระตุ้น

รูปที่ 3.21 (ข) แสดงการต่อวงจรขดลวดแบบกระตุ้นให้เกิดขั้วแม่เหล็กพร้อมกัน 2 ขั้วที่อยู่ใกล้กัน ทำให้โรเตอร์เคลื่อนที่มาหยุดหมุนอยู่ระหว่างขั้วแม่เหล็กทั้งสอง การเพิ่มจำนวนสเต็ปสามารถทำได้โดยไม่ต้องเพิ่มจำนวนของโครงสร้างภายในได้ โดยการจ่ายกำลังไฟฟ้าไปยังขั้วแม่เหล็ก 2 ขั้วที่อยู่ใกล้กันในเวลาเดียวกัน ซึ่งจะทำให้โรเตอร์หยุดหมุนอยู่ระหว่างกลางของ 2 ขั้วแม่เหล็กนั้น และวิธีการนี้ยังช่วยให้เกิดแรงบิต (Torque) มากขึ้นด้วยดังแสดงในรูปที่ 3.21 (ข) การพันขดลวดหรือคอยล์บนสเต็ปปิ้งมอเตอร์มีอยู่ 2 วิธีคือ แบบไดโพลาร์ (Dipolar) และแบบยูนิโพลาร์ (Unipolar) ดังแสดงในรูป 3.22 แบบไบโพลาร์มีการพันขดลวด 1 ขดบนแต่ละขั้วของแม่เหล็กของสเตเตอร์ ขั้วแม่เหล็กที่เกิดขึ้นบนสเตเตอร์จะถูกกำหนดโดยทิศทางของกระแสไฟฟ้า และสามารถทำให้เกิดขั้วแม่เหล็กในทิศทางตรงกันข้ามได้โดยการกลับทิศทางการไหลของกระแสไฟฟ้า แบบยูนิโพลาร์จะมีการพันขดลวด 2 ขด บนแต่ละขั้วแม่เหล็กของสเตเตอร์ ซึ่งแต่ละขดจะทำให้เกิดขั้วแม่เหล็กในทิศทางตรงกันข้ามกัน การกลับขั้วแม่เหล็กเปลี่ยนไปมาทำได้โดยการสวิตซิ่งกระแสไฟฟ้าจากขดลวดหนึ่งไปยังอีกขดลวดหนึ่งแทนเท่านั้น โดยปกติของขดลวดทั้งสองจะมีการเชื่อมต่อกันหรือมีจุดร่วมเพื่อลดจำนวนของสายไฟที่ต่อจากมอเตอร์

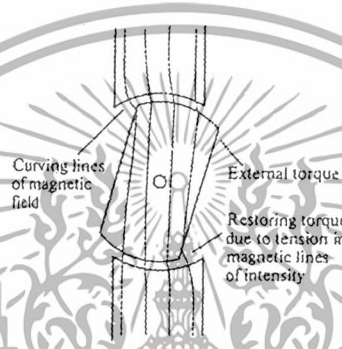
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 3.22 แสดงวงจรจ่ายกำลังไฟฟ้าให้กับสเต็ปปิ้งมอเตอร์ทั้งสองแบบ

(ก) สำหรับชนิดยูนิโพลาร์ ซึ่งใช้ทรานซิสเตอร์สวิตช์เพียงตัวเดียวต่อ 1 คอยล์

(ข) สำหรับชนิดไบโพลาร์ ซึ่งใช้ทรานซิสเตอร์สวิตช์ 4 ตัวต่อ 1 คอยล์



ภาพที่ 3.23 แสดงแรงดึงดูด ทำให้เกิดทอร์กที่หมุนโรเตอร์ให้ไปอยู่ในตำแหน่งที่สมดุล

จากรูปที่ 3.32 แสดงรูปแท่งแม่เหล็กถาวรติดอยู่บนเพลาและสามารถหมุนได้อย่างอิสระ ในขณะที่ไม่มี ขั้วไฟฟ้าที่สเตเตอร์ ถ้าก่อนมีขั้วไฟฟ้า ตำแหน่งของโรเตอร์แม่เหล็กอยู่ตามแกน a-a' ซึ่งทำมุม 0 องศากับแกนของขั้วแม่เหล็กไฟฟ้า

ดังนั้นจะได้แรงแม่เหล็กที่เกิดจากการดึงดูดของขั้วแม่เหล็กที่ต่างกันทำให้เกิดส่วนของแรงบิดปกติ

$$F_n = F\cos\alpha$$

จะได้ทอร์กรวม

$$T = F_n R$$

(ทำให้โรเตอร์หมุนไปในทิศทาง CW จนกว่าแกนของโรเตอร์ a-a' จะอยู่ในแนวเดียวกับแกนของสเตเตอร์)

ถ้าหากมีคู่ขั้วแม่เหล็กไฟฟ้าหลายๆขั้ว รอบๆสเตเตอร์และถ้าหากขั้วเหล่านั้นถูกกตุ้นด้วยกระแสพัลส์ ในรูปแบบที่เรียงลำดับกันไป โรเตอร์ก็จะหมุนในรูปลักษณะของสเต็ปทีเป็นๆไปตามการหมุนของสนามแม่เหล็กที่เกิดจากการสวิตช์ที่เรียงลำดับของขดลวดแม่เหล็กไฟฟ้าของสเตเตอร์

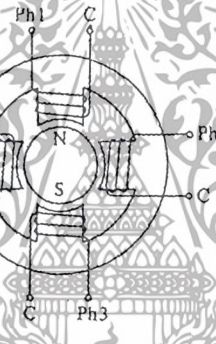
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3.3 สเต็ปปิ้งมอเตอร์ชนิดต่างๆ

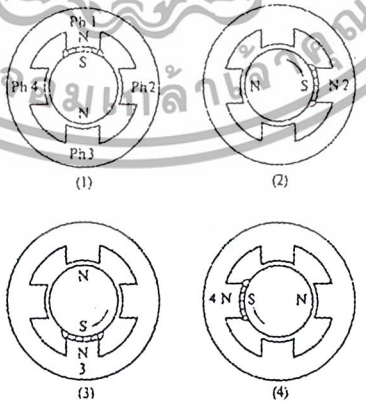
3.3.3.1) สเต็ปปิ้งมอเตอร์แบบแม่เหล็กถาวร

สเต็ปปิ้งมอเตอร์ชนิดนี้ใช้แม่เหล็กถาวรเป็นโรเตอร์ และมีซี่ฟันของสเตเตอร์ล้อมรอบ ซี่ฟันของสเตเตอร์ถูกพันด้วยขดลวดสำหรับสร้างสนามแม่เหล็ก เมื่อต้องการให้สเต็ปปิ้งมอเตอร์แบบแม่เหล็กถาวรมีขนาดมุมสเต็ปเล็กลงจะต้องเพิ่มจำนวนขั้วแม่เหล็กของโรเตอร์และจำนวนซี่ฟันของสเตเตอร์แต่กมีขีดจำกัดในการเพิ่มจำนวนขั้วแม่เหล็กของโรเตอร์ เนื่องจากการสร้างแม่เหล็กถาวรให้มีโครงสร้างแบบมีขั้วแม่เหล็กหลายขั้วทำได้ยาก

สมมติว่าสเต็ปปิ้งมอเตอร์แบบแม่เหล็กถาวรขนาดสี่เฟสมีโรเตอร์เป็นแม่เหล็กถาวรทรงกระบอกและสเตเตอร์มีซี่ฟันรอบๆพันด้วยขดลวด มีรูปแบบพื้นฐานการทำงาน คือ เมื่อสร้างสัญญาณกระตุ้นตามลำดับเฟสโรเตอร์ก็จะหมุนไปตามทิศทางของการกระตุ้น ดังแสดงในรูป



ภาพที่ 3.24 แสดงโครงสร้างของสเต็ปปิ้งมอเตอร์แบบแม่เหล็กถาวร



ภาพที่ 3.25 แสดงการทำงานในสเต็ปปิ้งมอเตอร์แบบแม่เหล็กถาวรขนาดสี่เฟส

ข้อเสียของสเต็ปปิ้งมอเตอร์แบบแม่เห็กถาวรคือมีขนาดมุมสเต็ปใหญ่ทำให้มีความละเอียดของสเต็ปต่อรอบน้อยเนื่องจากโครงสร้างของโรเตอร์เป็นแม่เหล็กถาวร การสร้างแม่เหล็กถาวรให้มีหลายขั้วนั้นทำได้ยาก ทำให้ไม่สามารถสร้างสเต็ปมอเตอร์ขนาดเล็กได้ สเต็ปปิ้งมอเตอร์แบบแม่

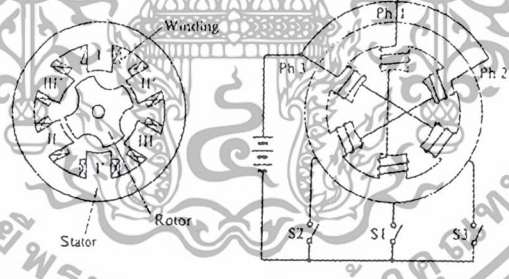
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เหล็กถาวรส่วนใหญ่จะมีโครงสร้างขนาดเล็กทำค่าทอร์กท่ได้ต่อหน่วยปริมาตรมีค่าต่ำ ถ้าต้องการปรับปรุงประสิทธิภาพในเรื่องของทอร์กแม่เหล็กถาวรที่ใช้ต้องทำจากสารแม่เหล็กที่มีสภาพความเป็นแม่เหล็กสูง สารแม่เหล็กราคาถูก เช่น เฟอไรท์ มีสภาพความเป็นแม่เหล็กน้อยจึงให้ค่าทอร์กต่ำ

3.3.3.2) สเต็ปปิ้งมอเตอร์แบบแปรค่ารีลัคแตนซ์ได้

โครงสร้างของสเต็ปปิ้งมอเตอร์แบบแปรค่ารีลัคแตนซ์ได้มีโรเตอร์และสเตเตอร์ทำจากโลหะซิลิกอนซึ่งเป็นสารแม่เหล็กที่มีสภาพซึมซับได้ทางเหล็กสูงและยอมให้สนามแม่เหล็กจำนวนมากไหลผ่านได้

ในรูป 3.25 เป็นสเต็ปปิ้งมอเตอร์แบบแปรค่ารีลัคแตนซ์ได้ขนาดสามเฟสโดยโครงสร้างของโรเตอร์มี 4 ซี่ฟัน และ สเตเตอร์มี 6 ซี่ฟัน และแต่ละเฟสของสเตเตอร์มี 2 ซี่ฟันทีอยู่ตรงกับข้ามกันและอยู่ห่างกัน 180 องศาต่ออนุกรมกันอยู่ ทั้งสองซี่ฟันของเฟสหนึ่งของสเตเตอร์จะมีขั้วตรงกันข้ามกัน ดังนั้น จากซี่ฟัน I, II , III จะเป็นขั้วเหลือ และ I', I'I', I'II' จะเป็นขั้วได้ เมื่อมอเตอร์ได้รับการกระตุ้น เมื่อเฟสหนึ่งถูกกระตุ้นสนามแม่เหล็กจะเกิดขึ้นดังรูปที่ 2.7 ดังนั้น ถ้าโรเตอร์ I และ I' จะวางตัวอยู่ในแนวเดียวกันซึ่งมีค่ารีลัคแตนซ์ที่น้อยที่สุดสถานะนี้เป็นสถานะที่จุดสมดุลย์เป็นจุดซึ่งเส้นแรงแม่เหล็กสั้นที่สุด แสดงดังในรูป 3.26

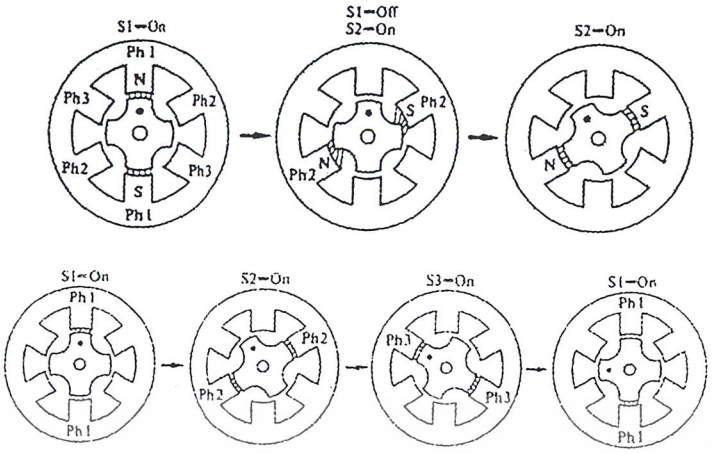


ภาพที่ 3.26 สเต็ปปิ้งมอเตอร์แบบแปรค่ารีลัคแตนซ์ได้

ภาพที่ 3.27 แสดงตำแหน่งสมดุลย์เมื่อเฟลใดเฟสหนึ่งของสเต็ปปิ้งมอเตอร์ถูกกระตุ้น

จากภาพที่ 3.27 สมามติว่าโรเตอร์และสเตเตอร์ไม่ได้อยู่ในแนวเดียวกัน เส้นแรงของเฟสที่ถูกกระตุ้นมีค่ารีลัคแตนซ์มาก สเต็ปปิ้งมอเตอร์แบบแปรค่ารีลัคแตนซ์ได้จะทำตัวให้มีค่ารีลัคแตนซ์ต่ำที่สุด โดยวางตัวอยู่ในแนวเดียวกันกับแนวของเส้นแรงแม่เหล็ก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 3.28 แสดงการเปลี่ยนลำดับสเต็ปในการเคลื่อนที่ของสเต็ปปิ้งมอเตอร์ได้

(1) ช่องว่างอากาศระหว่างซี่ฟันของโรเตอร์และสเตเตอร์ต้องเล็กที่สุดเท่าที่จะเป็นไปได้ เพื่อให้ค่าของทอร์กมากขึ้น และมีความถูกต้องแม่นยำทางตำแหน่งดีขึ้น เนื่องจากที่ระดับแมคนีโตโมทีฟฟอร์ชระดับเดียวกันช่องอากาศขนาดเล็กจะให้ค่าฟลักซ์แม่เหล็กสูงกว่า ซึ่งมีผลทำให้มีค่าทอร์คสูงกว่า และทำให้มีความคลาดเคลื่อนในการเคลื่อนที่จากจุดสมดุลย์มีน้อยกว่าช่องอากาศขนาดใหญ่

(2)สำหรับโครงสร้างที่ต้องการมุมสเต็ปขนาดเล็กความสัมพันธ์ระหว่างมุมสเต็ป (θ_n) มุมเฟส (m) จำนวนซี่ฟันของโรเตอร์ (N_r) และจำนวนสเต็ป (S) หาได้จากสมการ

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

เพื่อจะลดขนาดมุมสเต็ปลง จำนวนซี่ฟันของโรเตอร์จะต้องเพิ่มขึ้น ซี่ฟันของสเตเตอร์จะต้องเพิ่มขึ้นด้วย โดยโครงสร้างแต่ละขั้วของเฟสใดๆ จะมีหลายซี่ฟัน แต่จำนวนซี่ฟันของสเตเตอร์ไม่ใช่องค์ประกอบโดยตรงที่จะกำหนดมุมสเต็ปของสเต็ปปิ้งมอเตอร์

(3)การสร้างสเต็ปปิ้งมอเตอร์ให้มีโครงสร้างแบบหลายสแตค เพื่อประสิทธิภาพในเรื่องของทอร์คการสร้างสเต็ปปิ้งมาเตอร์แบบแปรค่ารีลัคแตนซ์ได้ให้มีโครงสร้างแบบหลายสแตคนี้แต่ละสแตคจะถูกกำหนดเป็นหนึ่งเฟส โดยโรเตอร์และสเตเตอร์จะมีซี่ฟันเหมือนกัน ระหว่างโรเตอร์และสเตเตอร์เพื่อช่วยเพิ่มทอร์คต่อหน่วยปริมาตรของโรเตอร์ ซึ่งเป็นการเพิ่มประสิทธิภาพของแมชชีน(machine)

3.3.3.3) สเต็ปปิ้งมอเตอร์แบบไฮบริด

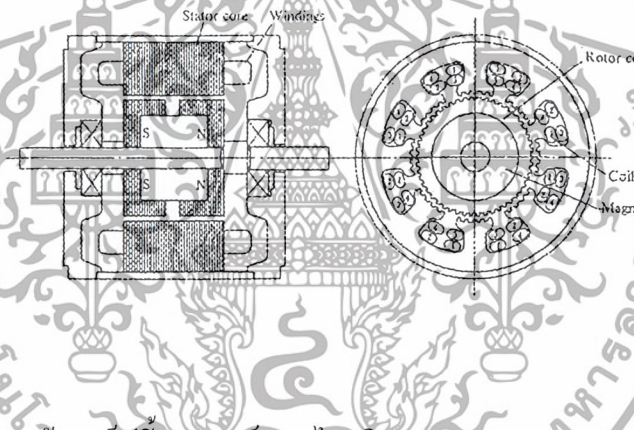
เป็นรูปแบบหนึ่งของสเต็ปปิ้งมอเตอร์ซึ่งมีโรเตอร์เป็นแม่เหล็กถาวร ไฮบริดหมายถึงการทำงานร่วมกันของมอเตอร์แบบถาวร และมอเตอร์แบบแปรค่าลีรัคแตนซ์ได้ สเต็ปปิ้งมอเตอร์แบบไฮบริดมีโครงสร้างของสเตเตอร์คล้ายกับโครงสร้างของสเต็ปปิ้งมอเตอร์แบบแปรค่ารีลัคแตนซ์ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

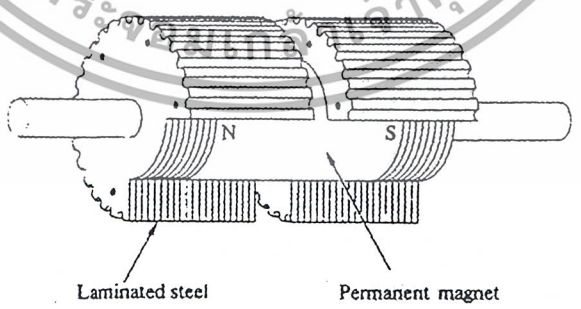
แต่การต่อขดลวดต่างกันโดย หนึ่งในเฟสของสเต็ปปิ้งมอเตอร์แบบแปรค่ารีลัคแตนซ์ได้จะมีสองขดลวดของสองเฟสถูกพันอยู่ที่ขั้วเดียวกันโดยที่ขั้วหนึ่งจะไม่เป็นของเฟสใดเฟสหนึ่ง

คุณสมบัติที่สำคัญของสเต็ปปิ้งมอเตอร์แบบไฮบริด คือ โครงสร้างของมอเตอร์มีแม่เหล็กถาวรอยู่ที่กึ่งกลางระหว่างเฟสทั้งสอง การเหนี่ยวนำสนามแม่เหล็กทำได้โดยใช้สนามแม่เหล็กซึ่งสร้างจากสเตเตอร์ซึ่งเป็นสนามแม่เหล็กแบบเฮทเทอโรโพลาร์ (heteropolar field) ดังนั้น ทอร์คเกิดจากการทำงานร่วมกันของสนามแม่เหล็กสองชนิดคือ สนามจากแม่เหล็กถาวร และสนามแม่เหล็กเหนี่ยวนำที่เกอดจากการกระตุ้นของขดลวดแต่ละขด โครงสร้างของซี่ฟันของสเตเตอร์จะใหญ่กว่าซี่ฟันของโรเตอร์เล็กน้อย เพื่อเพิ่มความถูกต้องแม่นยำทางตำแหน่งของการเคลื่อนที่

หลักการทำงานของสเต็ปปิ้งมอเตอร์แบบไฮบริดที่แตกต่างจากสเต็ปปิ้งมอเตอร์แบบแปรค่ารีลัคแตนซ์ได้ คือแรงบิดที่เกิดจากสนามแม่เหล็กจะไม่ขึ้นอยู่กับกระแสที่ผ่านขดลวดอย่างเดียว แต่ขึ้นอยู่กับโครงสร้างของซี่ฟันด้วย ซึ่งซี่ฟันถูกออกแบบเพื่อให้ได้โครงสร้างขนาดเล็ก และใช้แม่เหล็กถาวรเป็นแกนกลางเพื่อลดผลของการออสซิลเลตทางแมคคานิกส์



ภาพที่ 3.29 แสดงโครงสร้างสเต็ปปิ้งมอเตอร์แบบไฮบริด



ภาพที่ 3.30 แสดงหลักการทำดงานของสเต็ปปิ้งมอเตอร์แบบไฮบริด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ข้อดีของสเต็ปปิ้งมอเตอร์แบบไฮบริด คือ มีขนาดสเต็ปขนาดเล็กมีความละเอียดของสเต็ปต่อรอบสูง มีค่าทอร์คสูงกว่าสเต็ปปิ้งมอเตอร์แบบแปรค่ารีลัคแตนซ์ได้ แต่สเต็ปปิ้งมอเตอร์แบบแปรค่ารีลัคแตนซ์ได้มีแรงเฉื่อยแบบแมคคานิคส์น้อยกว่าสเต็ปปิ้งมอเตอร์แบบไฮบริด

นอกจากสเต็ปปิ้งมอเตอร์ทั้งสามชนิดที่กล่าวมาแล้ว ยังมีสเต็ปปิ้งมอเตอร์ชนิดอื่นๆ ที่ไม่ได้กล่าวถึงอีกเช่น ลิเนียร์สเต็ปปิ้งมอเตอร์ ซึ่งเป็นมอเตอร์ที่ได้รับการออกแบบให้มีการเคลื่อนที่แบบเป็นเชิงเส้น อีเล็กโตรไฮดรอลิกสเต็ปปิ้งมอเตอร์กำลังสูงสำหรับใช้ในอุตสาหกรรม เป็นต้น

3.3.4 สัญลักษณ์ต่างๆ ของสเต็ปปิ้งมอเตอร์

N_r = จำนวนซี่ฟันของโรเตอร์

N_s = จำนวนซี่ฟันของสเตเตอร์

N_p = จำนวนเฟส

P_r = ความห่างระหว่างปลายซี่ฟันโรเตอร์ (องศา)

P_s = ความห่างระหว่างปลายซี่ฟันของสเตเตอร์ (องศา)

θ_n = มุมสเต็ป (องศา)

R_s = อัตราการสเต็ปหรือความเร็วในการสเต็ป (สเต็ป/รอบ)

$X = N_s / N_p$ = จำนวนซี่ฟันของสเตเตอร์ต่อเฟส

3.3.5 พารามิเตอร์ต่างๆ ของสเต็ปปิ้งมอเตอร์

(1)ความห่างระหว่างหลายซี่ฟันโรเตอร์และสเตเตอร์

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

(2)มุมสเต็ป(step angle)

เมื่อโรเตอร์เคลื่อนไหวในขนาดมุม P_r ได้เท่ากับ N_p สเต็ป ดังนั้นเราจะหามุมสเต็ปได้

$\theta_n = P_r/N_p = 360/N_r N_p$ องศา/สเต็ป

มุมสเต็ปจะเท่ากับความแตกต่างระหว่าง P_r และ P_s ดังนั้น เราจะหามุมสเต็ปได้เป็น

$\theta_n = |P_r - P_s|$ องศา/สเต็ป

(3)อัตราการสเต็ป(stepping rate)

ความเร็วในการสเต็ปต่อรอบ (360องศา) หาได้เป็น

$R_s = 360/\theta_n = N_r N_s$ สเต็ปต่อรอบ

(4)ความเร็วของสเต็ปปิ้งมอเตอร์(speed of step motor)

เมื่อเราป้อนอินพุทพัลส์ที่มีความถี่ (f) สเต็ปต่อพัลส์ให้กับสเต็ปปิ้งมอเตอร์จะสเต็ปได้ด้วยความเร็ว (สเต็ป/พัลส์) * f (พลส์/วินาที)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$1/R_x$ (รอบ/สเต็ป) * (พัลส์/วินาที)(สเต็ป/พัลส์) * 60(วินาที/นาที)

ความเร็วของมอเตอร์(ω)=f/R_s = $60*f/ N_r N_p$

หรือ $X= R_s/ N_p(N_p+1)$ = $N_r/(N_p+1)$

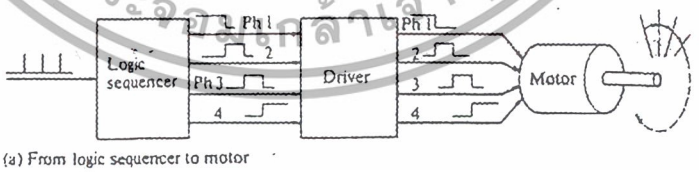
จำนวนโพลของสเตเตอร์ต่อเฟส(X) สัมพันธ์กับอัตราการสเต็ปรือจำนวนซี่ฟันของโรเตอร์

3.3.6 การควบคุมการทำงานของสเต็ปปิ้งมอเตอร์

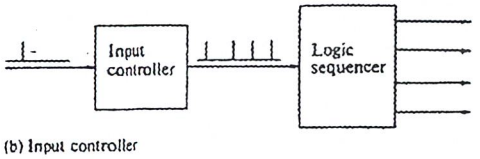
วงจรที่ใช้ควบคุมสเต็ปปิ้งมอเตอร์ส่วนใหญ่เป็นวงจรลอจิก และวงจรภาคกำลัง ซึ่งใช้สำหรับสเต็ปปิ้งมอเตอร์แบบแปรค่ารีลัคแตนซ์ด้ สเต็ปปิ้งมอเตอร์แบบแม่เหล็กถาวรและสเต็ปปิ้งมอเตอร์แบบไฮบริดประสิทธิภาพของมเตอร์ขึ้นอยู่กับตัวแปรต่อไปนี้ เช่น ไดนามิกส์ทอร์ค ความเร็ว การตอบสนองต่อความถี่ และการแดมพ์(damping) เป็นต้น นอกจากนั้นยังมีพารามิเตอร์บางอย่าง เช่น แระสิทธิภาพ และ การสูญเสียกำลัง ก็ขึ้นอยู่กับวงจรขับเช่นกัน

รูปแบบการควบคุมสเต็ปปิ้งมอเตอร์โดยทั่วไป เป็นการคบคุมภาคกำลัง และลำดับการจ่ายพลังงานให้กับขดลวดในเฟสต่างๆ ของสเต็ปปิ้งมอเตอร์ วงจรกำลังจะทำงานในทิศทางเดียวหรือ 2 ทิศทาง ขึ้นอยู่กับวงจรจัดลำดับลอจิก ซึ่งใช้เป็นสัญญาณขับเสสของทรานซิสเตอร์กำลัง ซึ่งจะเพิ่มขนาดสัญญาณเพื่อจ่ายกระแสให้ขดลวดของมอเตอร์และต้องใช้วงจรจำกัดกระแส เมื่อใช้การกระตุ้นแบบใช้แรงดันเกินเพื่อรักษาเสถียรภาพการทำงานของสเต็ปปิ้งมอเตอร์และใช้วงจรจำกัดแรงดันเกอนเพื่อจำกัดสไปค์โวลท์เตจ (spike voltage) ให้อยู่ในระดับที่ปลอดภัยของการใช้งานทรานซิสเตอร์กำลัง

วงจรจัดลำดับลอจิก (sequencer logic circuit) จะมีการทำงานเป็นรลักษณะของวงจรนับแบบวงแหวนซึ่งมีขนาดเท่ากับจำนวนเฟสของสเต็ปปิ้งมอเตอร์ การใช้วงจรจัดลำดับลอจิกจะกระตุ้นวงจรขับได้ครั้งละครึ่งเฟสเรียงตามลำดับกันไป



(a) From logic sequencer to motor



(b) Input controller

ภาพที่ 3.31 แสดงบล็อกไดอะแกรมของระบบควบคุมการขับสเต็ปปิ้งมอเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สัญญาณจากวงจรลำดับลอจิกจะต้องถูกขยายให้เพียงพอที่จะจ่ายให้กับสเต็ปปิ้งมอเตอร์ ดังนั้น จึงต้องป้องสัญญาณนี้ให้กับวงจรทรานซิสเตอร์กำลังอีกทีหนึ่งเพื่อจะเพิ่มค่ากระแสให้เพียงพอที่จะจ่ายให้ขดลวดของสเต็ปปิ้งมอเตอร์

3.3.7 การสั่งให้สเต็ปปิ้งมอเตอร์หมุน

เราสามารถสั่งให้สเต็ปปิ้งมอเตอร์หมุนได้ 3 วิธี คือ

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

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

3.3.7.3 แบบจ่ายกระแสไฟให้ทีละ 1 เฟส สลับกับ 2 เฟสเรียก One-Two Excitation หรือ Half Step เหมือนรูปแสดงมุมของโรเตอร์ในตารางที่ 2 แต่แบบนี้จำนวน Step จะเพิ่มขึ้นเป็น 2 เท่าของสองแบบแรกแต่แรงบิดเฉลี่ยจะน้อย

จากการจ่ายกระแสไฟฟ้าเฟสทั้ง 3 วิธี เราก็สามารถสั่งให้สเต็ปปิ้งมอเตอร์ หมุนทวนเข็มได้โดยมองการจ่ายกระแสให้เฟสย้อนกลับ เช่น ตามเข็มนาฬิการแบบ Full Step

ตารางที่3.5 การจ่ายกระแสไฟให้กับสเต็ปปิ้งมอเตอร์ แบบ Full Step

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

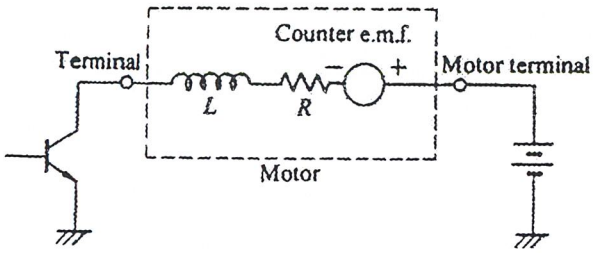
หมุนตามเข็มนาฬิกาจะเป็น 0104, 1002, 0203, 0304

หมุนทวนเข็มนาฬิกาจะเป็น 0403, 0302, 0201, 0104

3.3.8 หลักการออกแบบวงจรขับสเต็ปปิ้งมอเตอร์

การพิจารณาหลักการออกแบบวงจรขับกำลัง ได้คำนึงถึงหลายแฟกเตอร์ด้วยกัน เนื่องจากการหมุนของ สเต็ปปิ้งมอเตอร์จะทำให้เกิดแรงดันไฟฟ้าย้อนกลับซึ่งจะมีทิศทางตรงกันข้ามกับแหล่งจ่ายไฟเลี้ยง ซ่งจะเขียนวงจรสมมูลย์ 1 เฟสของสเต็ปปิ้งมอเตอร์ได้ ดังรูป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 3.32 แสดงวงจรสมสูลย์ของสเต็ปปิ้งมอเตอร์

สเต็ปปิ้งมอเตอร์ถูกออกแบบให้ทนความร้อนได้สูงถึง 100 องศาเซลเซียสแต่ในการใช้งานจริงจะถูกใช้งานที่เงื่อนไขต่ำกว่าจุดอิ่มตัวที่กำหนดมา แต่ในการขับสเต็ปปิ้งมอเตอร์ต้องเลือกใช้อุปกรณ์ที่กินกระแสสูงเนื่องจากแต่ละเฟสต้อง on และ off อยู่ตลอดเวลา ดังนั้น จึงต้องออกแบบป้องกันทรานซิสเตอร์กำลังสไปค์โลวท์เตจที่เกิดจากการเหนี่ยวนำของกระแสในขดลวด และโวลท์เตจเบรคดาวน์ เพื่อป้องกันการเสียหายของทรานซิสเตอร์กำลัง

3.4โปรแกรม AutoCAD

การใช้คอมพิวเตอร์ช่วยเขียนแบบและออกแบบ

การเขียนแบบและออกแบบโดยใช้คอมพิวเตอร์เป็นเครื่องมือช่วย (Computer Aided Design-CAD) จะเพิ่มขีดความสามารถทำงานได้มากกว่าการเขียนแบบด้วยมือ เพราะคอมพิวเตอร์สามารถบันทึกตำแหน่ง พิกัด ขนาด สี และแบบลายเส้นของวัตถุที่เขียนขึ้นมาโดยจัดเก็บไว้ในระบบฐานข้อมูล เพื่อนำแบบที่เขียนไว้ไปใช้ในงานต่อ หรือเพื่อการแก้ไขเพิ่มเติมในอนาคต

การใช้คอมพิวเตอร์ช่วยในการเขียนแบบทำให้สามารถสร้างเส้นแบบต่างๆ ได้ตามต้องการ และแก้ไขดัดแปลงเพิ่มเติมได้เมื่อมีความผิดพลาดโดยไม่ต้องเสียเวลาเขียนแบบใหม่ทั้งหมด แบบที่เขียนด้วยคอมพิวเตอร์จะให้ความละเอียดถูกต้องและนำไปพิมพ์ออกมาได้ตามต้องการ สรุปแล้วข้อได้เปรียบของการใช้คอมพิวเตอร์ช่วยในการเขียนแบบก็คือ ความสะดวกรวดเร็วและความถูกต้องแม่นยำ

แต่ปัญหาของการเรียนรู้เพื่อนำคอมพิวเตอร์มาใช้งานในการเขียนแบบด้วยคอมพิวเตอร์ก็คือ ขาดความมั่นใจที่จะนั่งหน้าคอมพิวเตอร์และใช้แป้นพิมพ์หรือเมาส์ในการเริ่มต้นเขียนแบบ แต่แท้จริงแล้วเครื่องคอมพิวเตอร์ก็คือเครื่องคิดเลข (Calculator) แบบหนึ่งเพียงแต่มีแป้นที่จะให้กดใช้งานได้มากมายและมีขั้นตอนที่แน่นอนในการสั่งงาน ดังนั้นผู้ที่ต้องการใช้เครื่องคอมพิวเตอร์ช่วยเขียนแบบจนถึงขั้นที่เกิดความมั่นใจจะต้องมีคู่มือหรือผู้ชี้แนะที่ดีและหัวใจสำคัญก็คือจะต้องมีการฝึกปฏิบัติอยู่เสมอ

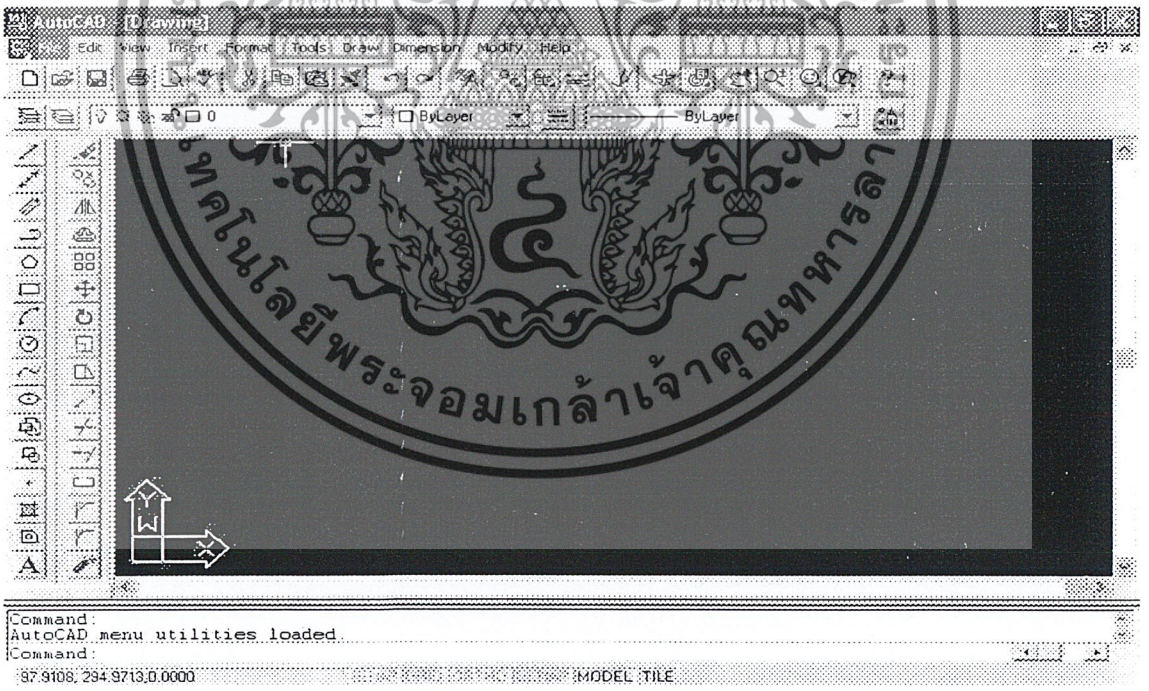
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การเขียนแบบด้วย AutoCAD นั้นจะต้องเรียนรู้การออกคำสั่งให้โปรแกรมเขียนเส้นตรง เส้นโค้ง วงกลม หรือ ตัวหนังสือ ผู้ใช้งานจะต้องรู้จักการป้อนข้อมูลและพารามิเตอร์ต่างๆให้คอมพิวเตอร์ตีความ โดยใช้แป้นพิมพ์หรือเลือกจากเมนู โดยใช้อุปกรณ์ชี้พารามิเตอร์ที่โปรแกรมคอมพิวเตอร์ต้องการทราบ คือ ตำแหน่ง ขนาด หรือ สี ของเส้นต่างๆ เป็นต้น หลังการออกคำสั่งและป้อนข้อมูลต่างๆแล้ว คอมพิวเตอร์ก็จะวาดเส้นตามที่ต้องการให้ปรากฏบนจอภาพ

นอกจากการเขียนเส้นต่างๆแล้ว AutoCAD ยังมีความสามารถในการแก้ไขเส้นที่วาดไว้ เช่น การลบ การเคลื่อนย้ายหรือลอกแบบได้รวดเร็วกว่าการเขียนด้วยมือรวมทั้งการแสดงภาพหรือขยายภาพให้สามารถมองได้ชัดเจนขึ้น

AutoCAD เป็นโปรแกรมคอมพิวเตอร์ที่มีความเหมาะสมสำหรับงานเขียนแบบลายเส้นดังต่อไปนี้

- งานเขียนแบบด้านสถาปัตยกรรม
- งานออกแบบตกแต่งภายในและแบบลายเส้นด้านศิลปกรรม
- การเขียนแผนผัง flowchart และเขียนกราฟ
- งานเขียนแบบทางวิศวกรรมทั่วไป
- งานเขียนแบบด้านสำรวจภูมิประเทศและภูมิทัศน์



ภาพที่ 3.33 แสดงหน้าต่างของโปรแกรม AutoCAD R14

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.4.1การทำ File Plot

เมื่อวาดรูปแบบเหล็กที่จะตัดใน AutoCad แล้วต่อมาจะทำการวิเคราะห์จุดต่างๆของแบบที่วาดออกมาโดยใน AutoCad จะมีรูปแบบของการวิเคราะห์ File Plot หลักๆอยู่ 2 แบบ คือ ระบบ DXF และ ระบบ HPGL ซึ่งรายละเอียดของแต่ละระบบเป็นดังนี้

3.4.1.1DXF

ระบบ DXF นี้การจัดเก็บรูปภาพจะเป็นแบบ ASCII และ Binary Vector โดยการจัดเก็บของ File ใน DXF จะจัดดังนี้

- HEADER Section จะเก็บข้อมูลโดยมากจะบอกถึงตัวอักษร และ มิติ
- TABLES Section จะมีการจัดเก็บที่จำกัดแน่นอนของตัวแปรที่กำหนดไว้ เช่น Layers, Viewing angles และ distances, coordinate system, styles of dimensioning
- BLOCKS Section จะเก็บเฉพาะกลุ่มชื่อที่เป็นเอกลักษณ์
- ENTITIES Section จะระบุว่าเป็นระบบ 3-D หรือ 2-D

ซึ่งเมื่อทำการ Plot แล้วขนาดของ File จะมีข้อมูลยาวมากเพราะจะกำหนดรายละเอียดปลีกย่อยเยอะมากทำให้การนำ File มาวิเคราะห์ทำได้ยุ่งยากและทำให้สับสนในการอ่าน

3.4.1.2HPGL

ระบบ HPGL นี้ File ภาพจะเก็บเป็นระบบ Line Drawing โดยมีรูปแบบคำสั่งที่บรรจุอยู่ใน File ดังต่อไปนี้

- IN (Initialize) เป็นการ SET ค่าพารามิเตอร์ต่างๆให้อยู่ในรูปแบบ Defaults และเคลื่อน ปากกาไปที่ตำแหน่ง Origin เพื่อเตรียมการ Plot ครั้งใหม่
- DF (Defaults) เป็นการ SET ค่าพารามิเตอร์ต่างๆให้อยู่ในรูป Defaults แต่ไม่เปลี่ยน P1 หรือ P2
- SC (Scale) X0,X1,Y0,Y1 เป็นการ SET Scale ของการ Plot โดย X0,Y0 คือมุมล่างซ้าย และ X1,Y1 คือมุมบนขวา
- IP (Input Points) X1,Y1,X2,Y2 เป็นการเปลี่ยนจุดพิกัด (X1,Y1) ให้กลายเป็น P1 อันใหม่ และเป็นการเปลี่ยนจุดพิกัด (X2,Y2) ให้กลายเป็น P2 อันใหม่
- IW (Input Window) X0,Y0,X1,Y1 เป็นการกำหนด Limits ของขอบเขต
- RO (Rotate Angle) การหมุนจะเป็นแบบองศาและทวนเข็มนาฬิกาโดยใน HP-GL จะเป็นมุม 0 ถึง 90 องศา และ ใน HP-GL/2 จะเป็น 180 ถึง 270 องศา
- PG (PAGE) เป็นการเลื่อนกระดาษออกหรือเลื่อนกระดาษใน Frame ถัดไป
- SP (Select Pen) เป็นการเลือกปากกาสำหรับการวาดโดยจะเริ่มที่ตอนต้นของรูปภาพ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- PU (Pen Up) ยกปากกา
- PD (Pen Down) วางปากกา
- PA (Plot Absolute) X1,Y1,X2,Y2,... ; คือการเคลื่อนไปยัง (X1,Y1)และต่อมา ไปที่ตำแหน่ง (X2,Y2)
- PR (Plot Relative) X1,Y1,X2,Y2, ...; คือจากตำแหน่งปัจจุบันเคลื่อนที่โดยบวกเพิ่มไปอีก (X1,Y1) และบวกเพิ่มไปอีก (X2,Y2)
- LT (Line Type) Type ; เป็นการ Set ชนิดของเส้นที่วาด โดย Default เป็น Solid Line , Type 0 เป็น Dot ที่ตอนจบของเส้น , Type 1 เป็นเส้น Dot , Type 2 เป็น Short Dash , Type 3 เป็น Long Dash , Type 4 เป็น Dash และ Dot , Type 5 เป็น Long dash / Short Dash , Type 6 เป็น Long Dash / 2 Short dashes
- SM (Symbol Mode) Character ;
- PE (Plot Encoded , HP-GL/2 Only) Encoded String ;
- AA (Arc Absolute) X,Y,Angle ; วาดส่วนโค้งของวงกลมโดยที่จุดศูนย์กลางเป็นระยะ (X,Y)จากตำแหน่งปัจจุบันและมุมเป็นองศาทวนเข็มนาฬิกา (ถ้าตามเข็มเป็น ลบ) โดยทำในระบบ Absolute
- AR (Arc Relative) X,Y,Angle ; วาดส่วนโค้งของวงกลมโดยจุดศูนย์กลางเป็นระยะ(X,Y) จากตำแหน่งปัจจุบันและมุมเป็นองศาทวนเข็มนาฬิกา (ถ้าตามเข็มเป็น ลบ) โดยทำในระบบ Relative
- CI (Circle) Radius,Angle ; วาดวงกลมโดยจุดศูนย์กลางอยู่ที่ตำแหน่งปัจจุบันและ ให้รัศมีมาให้

3.4.1.3 เหตุผลที่เลือกใช้ File Plot แบบ HPGL

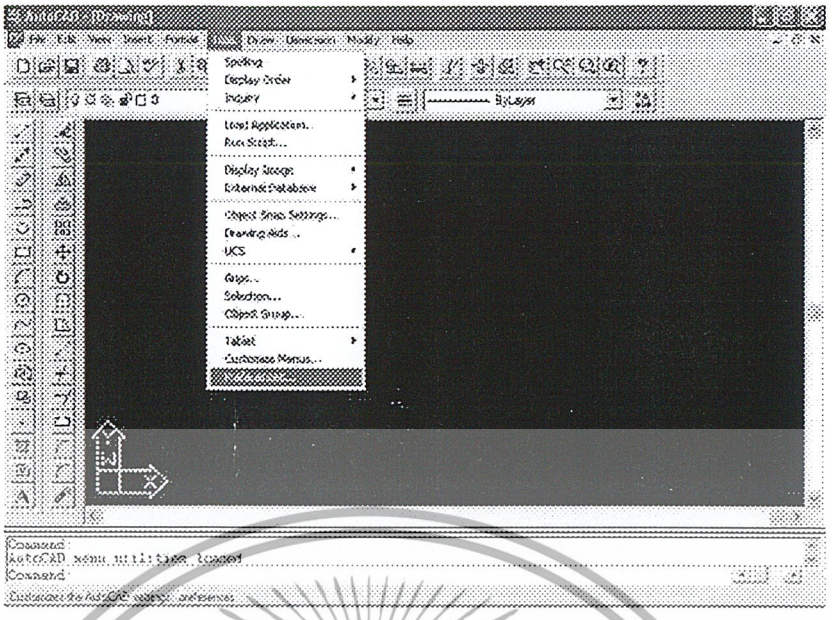
ระบบ File Plot แบบ HPGL จะมี CODE ที่สามารถดูให้เป็นค่าพิกัดออกมาได้ง่ายกว่าระบบ DXF เพราะเมื่อนำรูปข้างต้นมา Plot แบบ DXF File ที่ออกมาจะมีขนาดยาวมากและจะมี CODE เยอะกว่า และอีกผลหนึ่งคือ ระบบ File Plot แบบ HPGL สามารถหาพิกัดได้ง่ายและพิกัดที่ได้ก็สามารถสื่อสารกับตัว Controller ได้

3.4.1.4 วิธีการตั้งเครื่อง Plot ในโปรแกรม AutoCAD

1.เปิดโปรแกรม AutoCAD

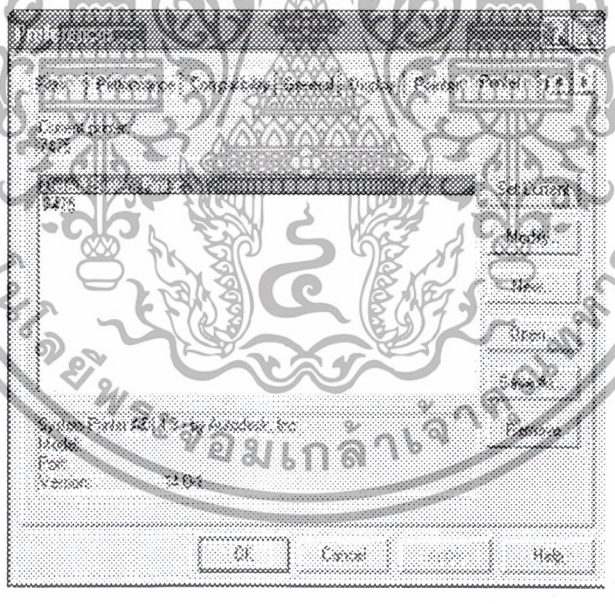
2.เลือกที่ TOOLS → Preferences

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 3.34 แสดงเมนู Tool ของ AutoCAD

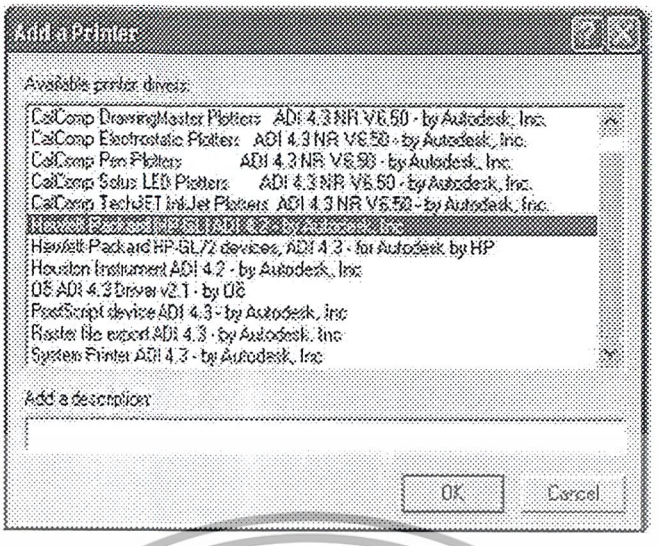
3.เลือกที่ Printer



ภาพที่ 3.35 แสดงหน้าจอ Tab "Printer"

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

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 3.36 แสดงหน้าจอ Add Printer

5.เลือก 1

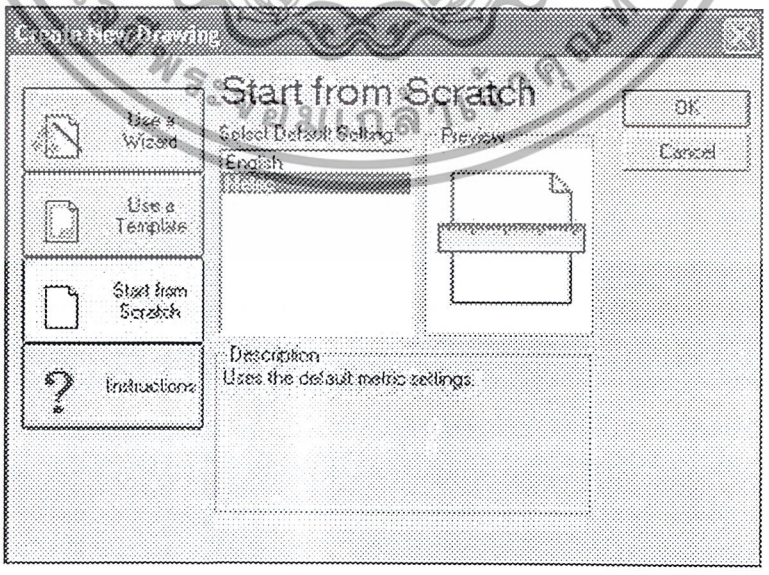
6.เลือก <S>serial port(local)

7.เลือก <com1> หรือ <com2>

8.เลือก <n>

3.4.2วิธีการเริ่มต้นเขียนแบบในโปรแกรม AutoCAD

1.เปิดโปรแกรมเลือก Metric



ภาพที่ 3.37 แสดงหน้าจอเริ่มต้นเมื่อเปิด AutoCAD

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.ในการวาดสี่เหลี่ยม กดที่ Icon รูปสี่เหลี่ยม หรือพิมพ์คำว่า Command : rectangle ↵

```
Command: _rectang
Chamfer/Elevation/Fillet/Thickness/Width/<First corner>:
```

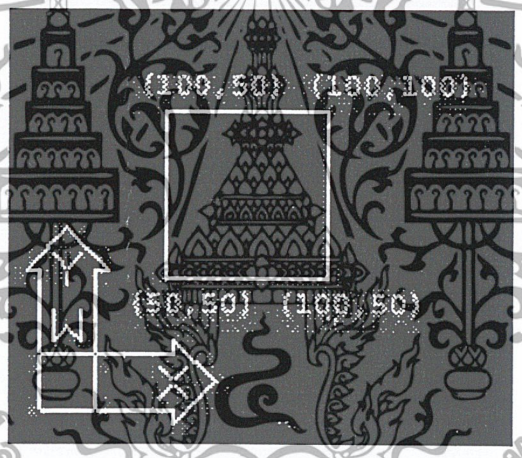
ภาพที่ 3.38 แสดงหน้าจอเมื่อพิมพ์คำว่า Command : rectangle ↵

จากนั้นพิมพ์ค่ามุมทั้งสองที่ทแยงมุมกัน เช่น ใส่ค่า 50,50 ↵ 100,100 ↵

```
Chamfer/Elevation/Fillet/Thickness/Width/<First corner>: 50,50
Other corner: 100,100
Command:
```

ภาพที่ 3.39 แสดงหน้าจอเมื่อพิมพ์คำว่า50,50 ↵ 100,100 ↵

จะได้รูปสี่เหลี่ยมที่มีมุมที่พิกัด (50,50), (50,100), (100,100), (100,50) ดังรูป



ภาพที่ 3.40 แสดงรูปสี่เหลี่ยมที่มีมุมที่พิกัด (50,50), (50,100), (100,100), (100,50)

3.ในการวาดสี่เหลี่ยม กดที่ Icon รูปวงกลม หรือพิมพ์คำว่า Command : circle ↵

```
Command: circle
3P/2P/TTR/<Center point>:
```

ภาพที่ 3.41 แสดงหน้าจอเมื่อพิมพ์คำว่า Command : circle ↵

จากนั้นพิมพ์ค่า จุดศูนย์กลาง และ รัศมี เช่น พิมพ์ 100,100↵ 20↵

```
Command: _circle 3P/2P/TTR/<Center point>: 100,100
Diameter/<Radius>: 20
```

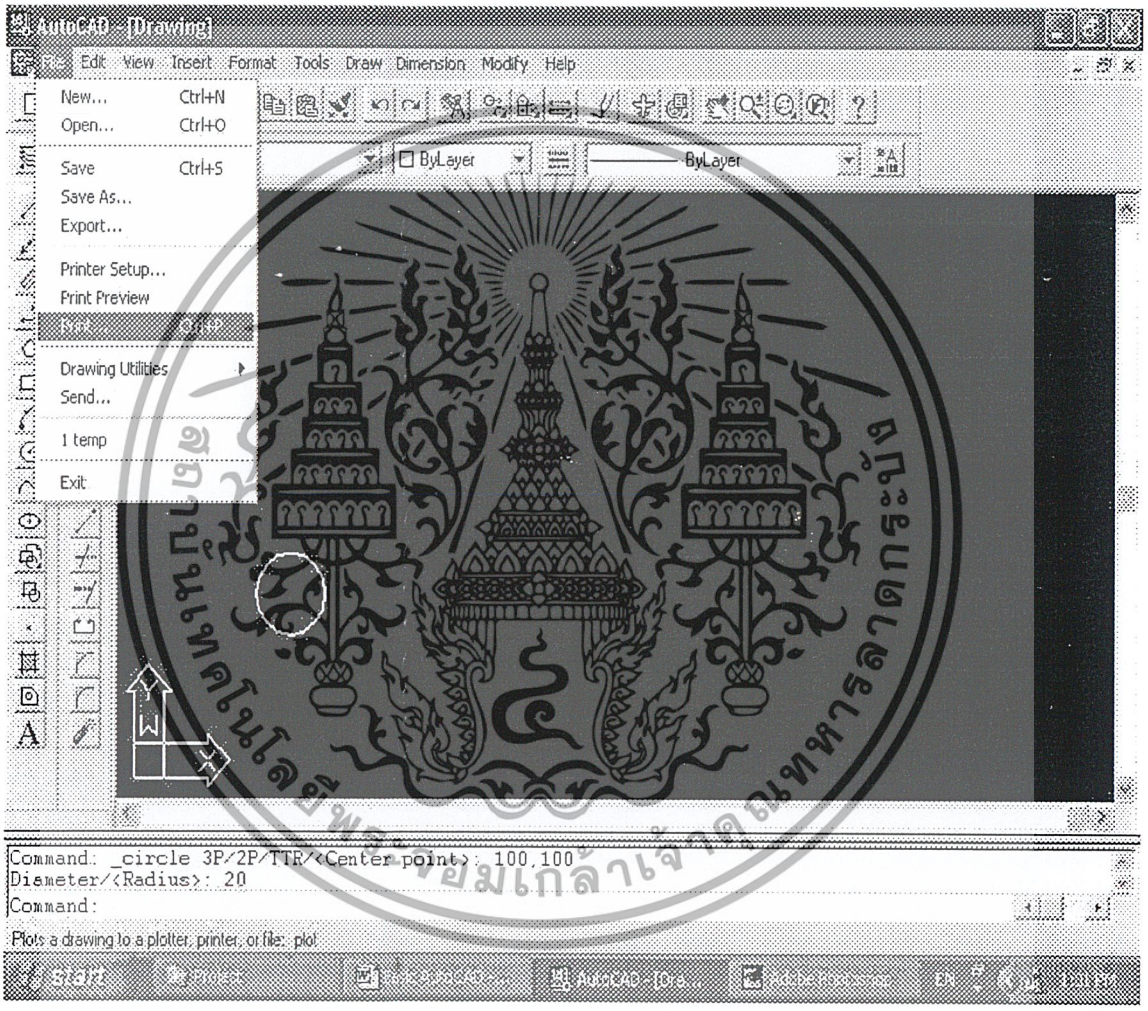
ภาพที่ 3.42 แสดงหน้าจอเมื่อพิมพ์ 100,100↵ 20↵

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จะได้รูปวงกลมที่มีจุดศูนย์กลางอยู่ที่พิกัด (100,100) และมีรัศมี 20 หน่วย

3.4.3 วิธีการตั้งค่า File.PLT ของโปรแกรม AutoCAD (หลังจากการเขียนภาพ)

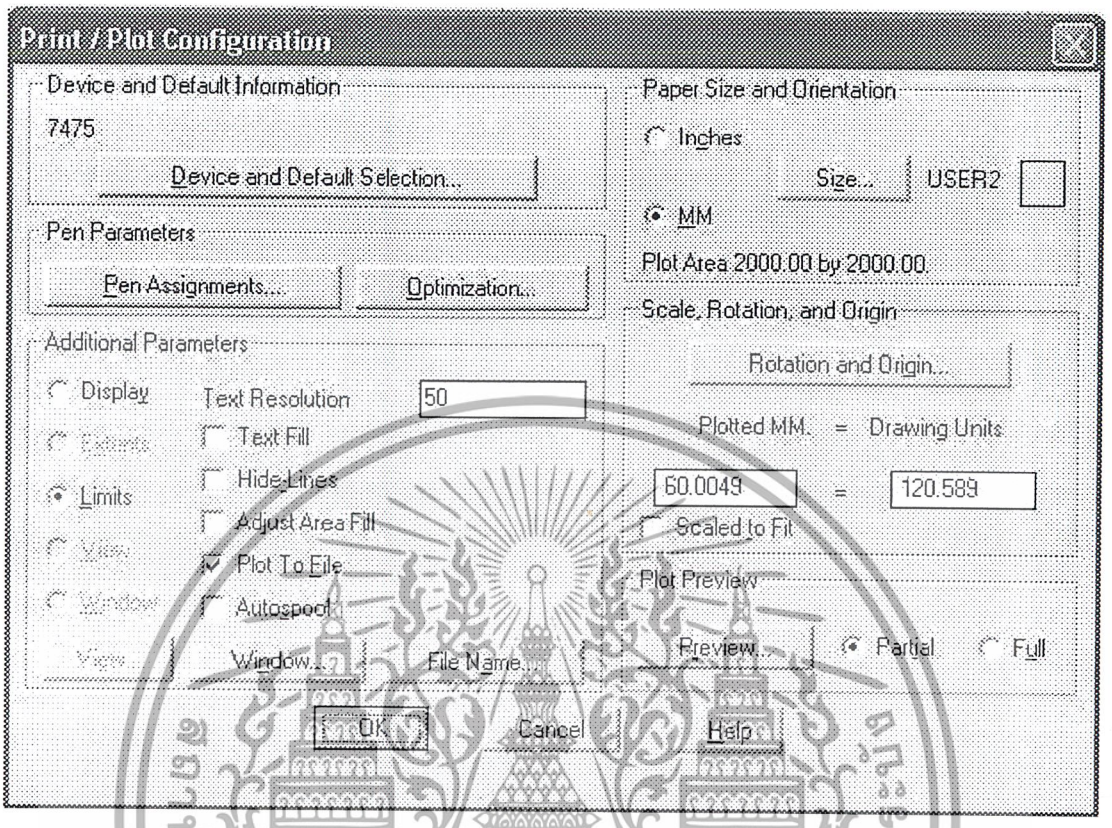
1.เลือก File → Print ดังภาพ



ภาพที่ 3.43 แสดงเมนู File

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จะได้ดังภาพ



ภาพที่ 3.44 แสดงหน้าจอ Print/Plot Configuration

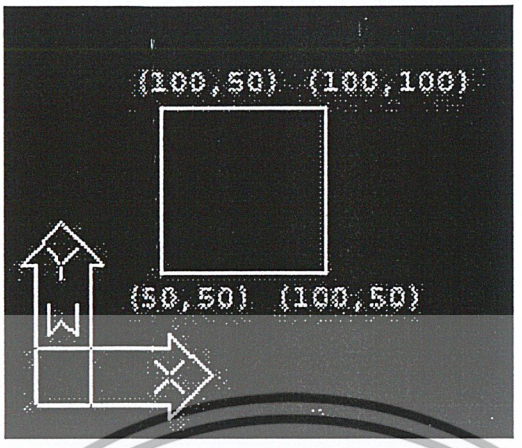
2.เลือก Device and Default Selection เลือก 7475

3.ที่ Additional Parameters เลือก Limit และ เลือก Plot to File

4.Scale, Rotation and Origin 60.0049 = 120.589 จะได้ดังรูป จะได้อัตราส่วน 1 mm ในโปรแกรม AutoCAD จะเท่ากับ 20 ใน File Plot เวลาที่จะส่งเข้า Controller จะต้องคูณ 5 ก่อน เนื่องจาก Motor 4 Phase ที่ใช้นี้เท่ากับ 400 steps ต่อ 1 รอบ, Motor หมุน 1 รอบทำให้ Motion เคลื่อนที่ไป 4 mm ดังนั้น 1 mm จะเท่ากับ 100 steps

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปสี่เหลี่ยมที่มีมุมที่พิกัด (50,50), (50,100), (100,100), (100,50)ดังรูป



ภาพที่ 3.45 แสดงรูปสี่เหลี่ยมที่มีมุมที่พิกัด (50,50), (50,100), (100,100), (100,50)

จะได้ File Plot หลังจากการตั้งค่าตามข้างต้นที่ได้กล่าวมาเป็นดังนี้

□.(;□.I81;;17:□.N;19:IN;SC;PU;PU;SP1;LT;VS36;PU;PA1000,1000;PD;PA2000,1000;PA2000,2000;PA1000,2000;PA1000,1000;PU;PA0,0;SP;

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4
การออกแบบและสร้างเครื่องตัดเหล็กอัตโนมัติด้วยพลาสมา

4.1 การออกแบบส่วนโปรแกรม

ในการออกแบบออกแบบโปรแกรมแบ่งออกเป็น

1 โหมดอัตโนมัติ คือโหมดที่โหลด File Plot มาจาก AutoCAD แล้วทำการตัดค่า File Plot ที่ได้จาก AutoCAD มาเก็บเป็นค่าพิกัด X, Y ในระบบ Absoluteและ ค่าการเคลื่อนที่ของหัวตัด

2. โหมดManual คือโหมดที่ผู้ใช้วาดรูปต้นแบบในโปรแกรมที่เขียนโดย Visaul Basic เลยและแปลงเป็นค่าพิกัด X, Y ในระบบ Absolute และค่าการเคลื่อนที่ของหัวตัด โดยฟังก์ชั่นการวาดมี วงกลม, วงรี , เส้นตรง, เส้นโค้ง

3. การวิเคราะห์ให้ทางเดินของการตัดใช้เวลาเร็วขึ้น

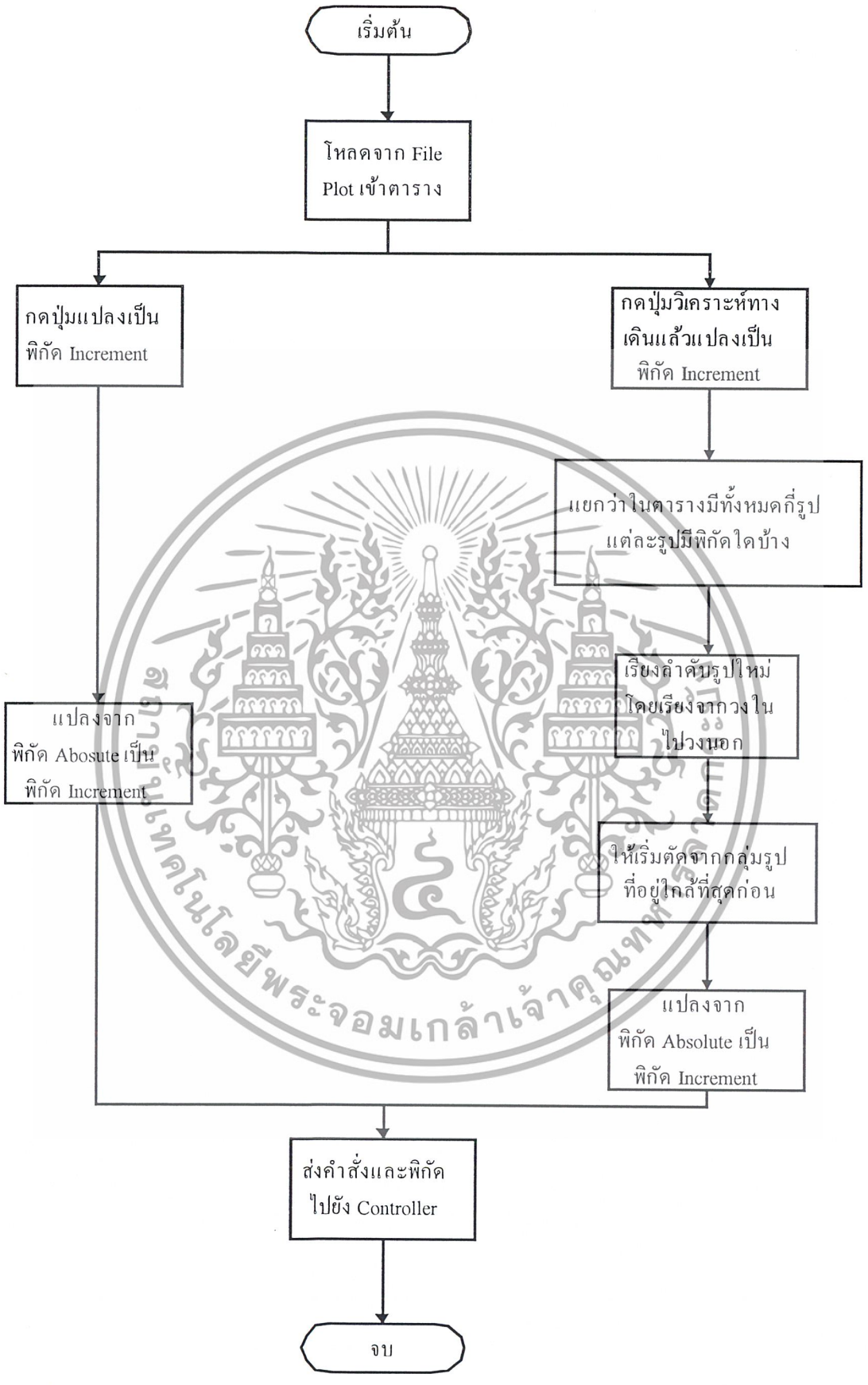
4. การแปลงจาก Absolute เป็นระบบ Increment และเก็บเข้าตารางฐานข้อมูล

5. การส่งค่าจากตารางฐานข้อมูลเข้า Controller

ในการใช้งานจริงนั้นการส่งค่าเข้าให้กับตัว Controller นั้น จะต้องส่งค่าให้ในระบบ Incremnet เป็นแบบ Step ซึ่งเมื่อนำ File Plot จาก โปรแกรม AutoCAD มาเข้าโปรแกรมเพื่อหาค่าออกมาจะได้ค่าเป็นระบบ Absolute ดังนั้นจึงจำเป็นต้องเขียนโปรแกรมขึ้นมาเพื่อแปลงจากระบบพิกัด X, Y ในระบบ Absolute ให้เป็นระบบ Increment แบบ Step

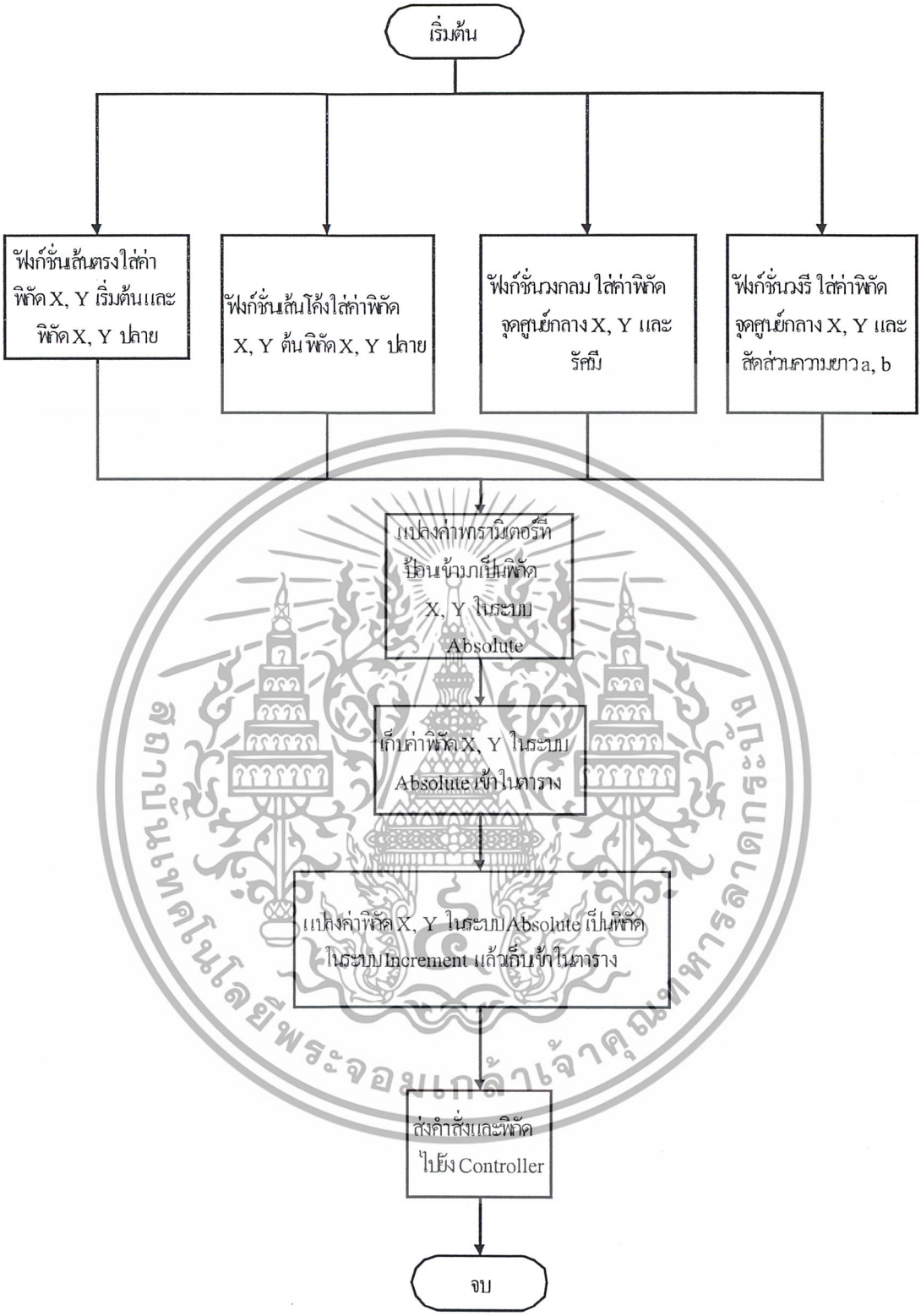
ซึ่งการออก โปรแกรมทั้งในโหมด Auto และ ในโหมด Manual จะแสดงตามขั้นตอนของ FlowChart ของทั้ง 2 โหมด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 4.1 แสดงFlow Chart การทำงานโหมด Auto

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 4.2 แสดง Flow Chart การทำงานโหมด Manual

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2 โหมดอัตโนมัติ

ปุ่มของโปรแกรมที่ใช้สำหรับโหมดการทำงานแบบอัตโนมัติ มีทั้งหมด 3 ปุ่ม คือ

1.ปุ่มโหลด File Plot เข้าตาราง

2.ปุ่มแปลงเป็นพิกัด Increment

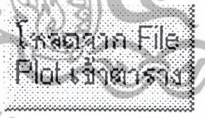
3.ปุ่มวิเคราะห์ทางเดินแล้วแปลงเป็นพิกัด Increment

ดังรูป

ภาพที่ 4.3 แสดงปุ่มวิเคราะห์ทางเดิน

รายละเอียดการทำงานของปุ่มทั้ง 3 ปุ่ม

1.ปุ่มโหลดจาก File Plot เข้าตาราง



ภาพที่ 4.4 แสดงปุ่มโหลดจาก File Plot

เป็นปุ่มกดสำหรับ นำข้อมูลจาก File Plot มาวิเคราะห์ หาทางเดินของรูปที่วาดไว้ในโปรแกรม Autocad มาเก็บไว้ในตารางเพื่อที่จะนำไปใช้ในขั้นตอนต่อไป ตารางที่ใช้ในโปรแกรมนี้ใช้ในรูปแบบของ Microsoft Access โดยโปรแกรมจะกำหนดให้ตารางที่ใช้นี้ ชื่อ Table.mdb ค่าที่เก็บในตารางจะเก็บอยู่ในรูปของตัวแปร 4 ตัว ดังนี้

1.NO หมายถึง ตัวเลขลำดับที่ของคำสั่ง เป็นตัวแปรชนิด Integer

2.ORDER หมายถึง คำสั่งของทางเดินในลำดับนั้นๆของคำสั่ง เป็นตัวแปรชนิด String

3.X หมายถึง พิกัด X ในลำดับนั้นๆของคำสั่ง เป็นตัวแปรชนิด Double

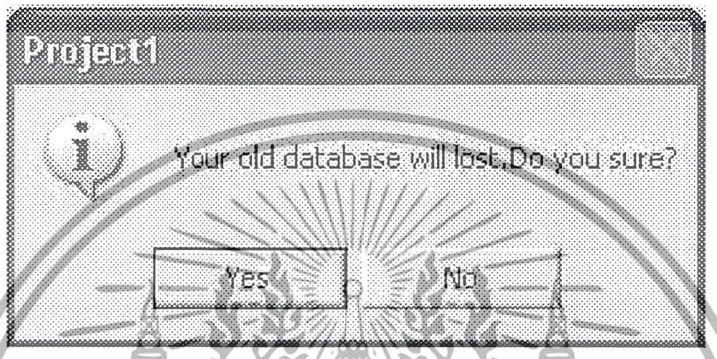
4.Y หมายถึง พิกัด Y ในลำดับนั้นๆของคำสั่ง เป็นตัวแปรชนิด Double

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยตำแหน่งของ File Plot จะถูกกำหนดอยู่ในโปรแกรมส่วนนี้ว่าให้อยู่ใน Folder ไหน โดยในโปรแกรมนี้จะกำหนดให้อยู่ใน C:\Project\drawing.plt

เมื่อทำการคลิ๊กที่ปุ่มนี้ จะทำการโหลดข้อมูลจาก File Plot โดยจะแบ่งเป็น 2 กรณี

1.ถ้าในตารางมีค่าเดิมที่เก็บไว้อยู่ โปรแกรมจะถามว่า ต้องการจะลบค่าตัวแปรเดิมที่บันทึกไว้ในตารางแล้วบันทึกค่าใหม่ที่ได้จากการวิเคราะห์ File Plot ลงไปหรือไม่



ภาพที่ 4.5 แสดงว่าเมื่อในตารางเดิมมีข้อมูลอยู่

ถ้าตอบว่าใช่ โปรแกรมจะลบค่าตัวแปรเดิมที่บันทึกไว้ในตารางทั้งหมด แล้วบันทึกค่าตัวแปรใหม่ที่ได้จากการวิเคราะห์ File Plot ลงไปตาราง

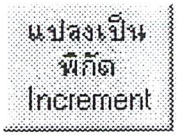
ถ้าตอบว่าไม่ โปรแกรมจะไม่ทำการลบค่าตัวแปรเดิมที่บันทึกไว้ในตารางและจะไม่บันทึกค่าใหม่ที่ได้จากการวิเคราะห์ File Plot ลงไปในตาราง

2.ถ้าในตารางเป็นตารางเปล่า ไม่มีค่าตัวแปรเดิมที่บันทึกไว้ในตาราง โปรแกรมจะบันทึกค่าตัวใหม่ที่ได้จากการวิเคราะห์ File Plot ลงไป โดยจะไม่ถามว่าต้องการลบค่าตัวแปรเก่าที่บันทึกไว้ในตารางแล้วบันทึกค่าใหม่ที่ได้จากการวิเคราะห์ File Plot หรือไม่

โดยค่าตัวแปร X, Y บันทึกลงไปในตารางในขั้นนี้จะเป็นในรูปของพิกัด Absolute ซึ่งจะทำให้ผู้ใช้โปรแกรมสามารถตรวจสอบค่าที่วิเคราะห์ได้จาก File Plot ว่าถูกต้องหรือไม่ในเบื้องต้นได้ แต่การที่จะนำค่าตัวแปรในตารางไปใช้ จะต้องผ่านการวิเคราะห์ในขั้นตอนต่อไปก่อน จึงจะนำไปใช้งานได้ เนื่องจาก Controller จะรับค่าตัวแปร X, Y ในรูปแบบของพิกัด Increment

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.ปุ่มแปลงเป็นพิกัด Increment



ภาพที่ 4.6 แสดงปุ่มแปลงเป็นพิกัด Increment

เป็นปุ่มกดสำหรับนำค่าในตารางหลังจากที่ได้ผ่านการทำงานของปุ่มที่1 ซึ่งค่าตัวแปร X, Y เดิมใน ตารางนั้นจะอยู่ในรูปของพิกัด Absolute นำมาวิเคราะห์แล้วแปลงค่าตัวแปรในตารางให้อยู่ในรูปแบบตัวแปร X, Y ในรูปแบบของพิกัด Increment โดยพิกัดทางเดินจะอยู่ในรูปทางเดินเดิมก่อนที่จะแปลงเป็นพิกัด Increment

ในขั้นตอนการทำงานของปุ่มนี้จะมีการประกาศตัวแปรที่สำคัญดังนี้

ORDER(a,b) จะเป็นตัวแปรชนิด Array of String สำหรับเก็บค่าตัวแปร ORDER ในลำดับของคำสั่ง

PICX(a,b) จะเป็นตัวแปรชนิด Array of Double สำหรับเก็บค่าตัวแปร X ในทุกลำดับของคำสั่ง

PICY(a,b) จะเป็นตัวแปรชนิด Array of Double สำหรับเก็บค่าตัวแปร Y ในทุกลำดับของคำสั่ง

NOPIC จะเป็นตัวแปรชนิด Integer สำหรับเก็บค่าว่าในตารางนี้ แบ่งได้เป็นกี่รูป

SEQUENCEPIC(a) จะเป็นตัวแปรชนิด Array of Integer สำหรับเก็บค่าว่ารูปที่ a นั้นมีลำดับของคำสั่งทั้งหมดกี่คำสั่ง

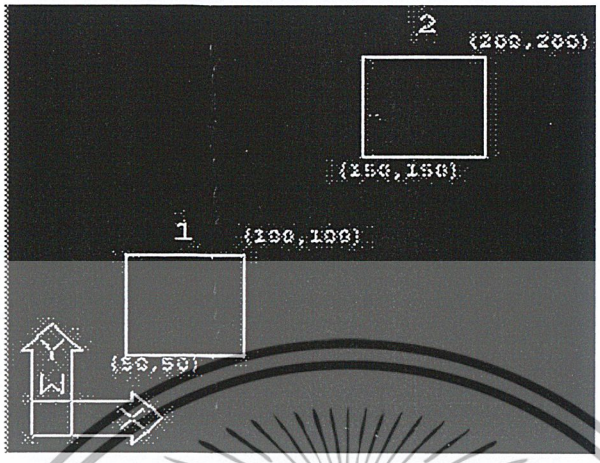
โดย a จะหมายถึง รูปที่ a ของตาราง

b จะหมายถึง ลำดับคำสั่งที่ b ของรูปที่ a

โดยใช้หลักการในการแยกรูปว่าในตารางนั้นมีทั้งหมดกี่รูป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตัวอย่างในการแยกรูปในโปรแกรมวาดรูปสี่เหลี่ยม 2 รูป ในโปรแกรม AutoCADโดยกำหนดพิกัดของสี่เหลี่ยมรูปที่ 1 เป็น พิกัด(50,50) และ พิกัด(100,100)กำหนดพิกัดของสี่เหลี่ยมรูปที่ 2 เป็น พิกัด(150,150) และ พิกัด(200,200) ดังรูป



ภาพที่ 4.7 แสดงสี่เหลี่ยมรูปที่ 1 เป็น พิกัด(50,50) และ พิกัด(100,100) และสี่เหลี่ยมรูปที่ 2 เป็น พิกัด(150,150) และ พิกัด(200,200)

จากนั้น Plot to file เป็น File Plot แล้วนำมาเปิดในโปรแกรม โดยใช้ปุ่มโหลดจาก File Plot เข้าตาราง จากนั้นดูค่าในตาราง โดยใช้ปุ่มดูค่าในตาราง จะได้ข้อมูลดังรูป

	no	order	x	y
▶	1	PU	0	0
	2	PA	5000	5000
	3	PD	0	0
	4	PA	5000	0
	5	PA	0	5000
	6	PA	-5000	0
	7	PA	0	-5000
	8	PU	0	0
	9	PA	10000	10000
	10	PD	0	0
	11	PA	5000	0
	12	PA	0	5000
	13	PA	-5000	0
	14	PA	0	-5000
	15	PU	0	0
	16	PA	-15000	-15000

ภาพที่ 4.8 แสดงค่าในตาราง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูป โดยหลักการในการแบ่งรูป จากรูปจะเห็นว่าทุกๆรูปจะมีรูปแบบที่เหมือนกัน คือ เริ่มต้นรูปจะขึ้นต้นด้วยคำสั่ง PU และจะขึ้นรูปใหม่ด้วยคำสั่ง PU เหมือนกัน จะมี 2 คำสั่งสุดท้าย จะเป็น PU และ PA เพื่อที่จะกำหนดทางเดินไปจบที่จุด (0,0) ทำให้โปรแกรมแยกได้ว่า รูปที่ 1 คือ คำสั่งตั้งแต่บรรทัดที่ 1 ถึงบรรทัดที่ 7

รูปที่ 2 คือคำสั่งตั้งแต่บรรทัดที่ 8 ถึงบรรทัดที่ 14 และ บรรทัดที่ 15 ถึงบรรทัดที่ 16 จะเป็นคำสั่งเพื่อกำหนดให้ทางเดินไปจบที่จุด (0,0)

โดยหลักการเดียวกันนี้โปรแกรมสามารถแยกรูปที่วาดใน AutoCAD ว่ามีกี่รูปได้ แม้ว่าจะมีรูปหลายรูปก็ตาม โดยปุ่มแปลงเป็นพิกัด Increment นี้จะมีขั้นตอนการทำงานของโปรแกรมทั้งหมด 4 ขั้นตอน ดังนี้

2.1.โปรแกรมจะนำค่าตัวแปรทั้ง 4 ตัวในตาราง คือ ORDER, X, Y มาเก็บไว้ในตัวแปร ORDER(a,b), PICX(a,b), PICY(a,b) ตามลำดับ จนครบทุกลำดับคำสั่ง

2.2 โปรแกรมจะนำค่าในตัวแปรมาคำนวณเพื่อที่จะเปลี่ยนจากค่าพิกัด X, Y แบบAbsolute เป็นแบบพิกัด X, Y แบบIncrement โดยจะแบ่งเป็น 3 กรณี ดังนี้

2.2.1 กรณีที่ ORDER เป็น PU จะใช้เป็น พิกัด (0,0) เหมือนเดิม

2.2.2 กรณีที่ ORDER เป็น PD จะใช้เป็น พิกัด (0,0) เหมือนเดิม

2.2.3 กรณีที่ ORDER เป็น PA ครั้งแรก คิดที่พิกัด X, Y ของ ORDER PA คู่แรก ใช้ค่าพิกัดเดิมเลย เนื่องจากเป็นการลบกันระหว่างค่าพิกัด X, Y นั้นกับพิกัด (0,0)

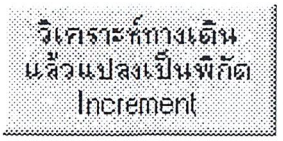
กรณีที่ ORDER เป็น PA ครั้งต่อๆไป คิดที่พิกัด X, Y ของ ORDER PA ใช้ค่าพิกัดตัวนั้นลบด้วยค่าพิกัด X, Yบรรทัดที่มี ORDER PA ก่อนหน้านั้น 1 บรรทัด

โดยโปรแกรมจะนำค่าที่คำนวณได้ ไปต่อท้ายตารางโดยที่จะยังไม่ลบค่าตัวแปร ORDER, X, Y เก่าที่อยู่ในรูปพิกัด Absolute

2.3 โปรแกรมจะทำการเพิ่มคำสั่งเข้าในตารางต่อจากตารางที่ได้มาจากข้อ 2. เพื่อที่จะให้ทางเดินไปจบที่จุด (0,0) ซึ่งจะเป็นการจบคำสั่ง

2.4 โปรแกรมจะทำการลบคำสั่งในตาราง เพื่อลบคำสั่งในส่วนของพิกัด Absolute ออกทั้งหมด ทำให้เหลือค่าตัวแปรในตารางเฉพาะในส่วนของพิกัด Increment

3.ปุ่มวิเคราะห์ทางเดินแล้วแปลงเป็นพิกัด Increment



ภาพที่ 4.9 แสดงปุ่มวิเคราะห์ทางเดินแล้วแปลงเป็นพิกัด Increment

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เป็นปุ่มกดสำหรับนำค่าในตารางหลังจากที่ได้ผ่านการทำงานของปุ่มที่1 ซึ่งค่าตัวแปร X, Y เดิมใน ตารางนั้นจะอยู่ในรูปของพิกัด Absolute นำมาวิเคราะห์ แล้วสร้างทางเดินใหม่ขึ้นมาเพื่อให้สามารถใช้ในการตัดชิ้นงานจริงได้ รวมถึงการเรียงลำดับในการตัด เพื่อให้ใช้เวลาในการตัดน้อยที่สุดด้วย และโปรแกรมจะทำการแปลงค่าตัวแปรในตารางให้อยู่ในรูปแบบตัวแปร X, Y ในรูปแบบของพิกัด Increment

ในขั้นตอนการทำงานของปุ่มนี้จะมีการประกาศตัวแปรที่สำคัญดังนี้

ORDER(a,b) จะเป็นตัวแปรชนิด Array of String สำหรับเก็บค่าตัวแปร ORDER ในทุกลำดับของคำสั่ง

PICX(a,b) จะเป็นตัวแปรชนิด Array of Double สำหรับเก็บค่าตัวแปร X ในทุกลำดับของคำสั่ง

PICY(a,b) จะเป็นตัวแปรชนิด Array of Double สำหรับเก็บค่าตัวแปร Y ในทุกลำดับของคำสั่ง

NOPIC จะเป็นตัวแปรชนิด Integer สำหรับเก็บค่าว่าในตารางนี้ แบ่งได้เป็นกี่รูป

SEQUENCEPIC(a) จะเป็นตัวแปรชนิด Array of Integer สำหรับเก็บค่าว่ารูปที่ a นั้นมีลำดับของคำสั่งทั้งหมดกี่คำสั่ง

XMIN(a) จะเป็นตัวแปรชนิด Array of Double สำหรับเก็บค่าว่ารูปที่ a นั้นมีค่าพิกัด X น้อยที่สุดเท่าไร

XMAX(a) จะเป็นตัวแปรชนิด Array of Double สำหรับเก็บค่าว่ารูปที่ a นั้นมีค่าพิกัด X มากที่สุดเท่าไร

YMIN(a) จะเป็นตัวแปรชนิด Array of Double สำหรับเก็บค่าว่ารูปที่ a นั้นมีค่าพิกัด Y น้อยที่สุดเท่าไร

YMAX(a) จะเป็นตัวแปรชนิด Array of Double สำหรับเก็บค่าว่ารูปที่ a นั้นมีค่าพิกัด Y มากที่สุดเท่าไร

โดย a จะหมายถึง รูปที่ a ของตาราง

b จะหมายถึง ลำดับคำสั่งที่ b ของรูปที่ a

โดยใช้หลักการแยกรูปจากที่กล่าวไว้ข้างต้น

GROUP(a) จะเป็นตัวแปรชนิด Array of String สำหรับเก็บค่าว่ารูปกลุ่มที่ a นั้นมีรูปใดอยู่บ้าง

XMINGROUP(a) จะเป็นตัวแปรชนิด Array of Double สำหรับเก็บค่าว่ากลุ่มที่ a นั้นมีค่าพิกัด X น้อยที่สุดเท่าไร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

XMAXGROUP(a) จะเป็นตัวแปรชนิด Array of Double สำหรับเก็บค่าว่ากลุ่มที่ a นั้นมีค่าพิกัด X มากที่สุดเท่าไร

YMINGROUP(a) จะเป็นตัวแปรชนิด Array of Double สำหรับเก็บค่าว่ากลุ่มที่ a นั้นมีค่าพิกัด Y น้อยที่สุดเท่าไร

YMAXGROUP(a) จะเป็นตัวแปรชนิด Array of Double สำหรับเก็บค่าว่ากลุ่มที่ a นั้นมีค่าพิกัด Y มากที่สุดเท่าไร

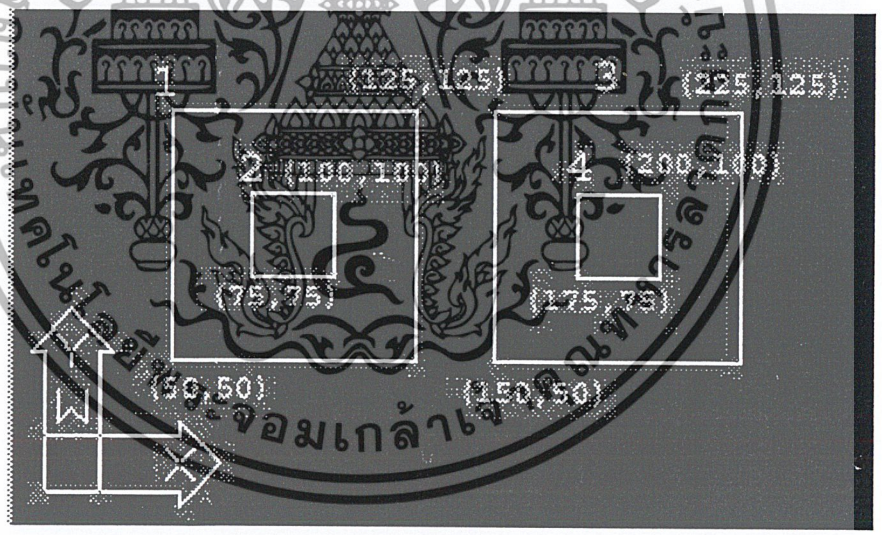
โดยใช้หลักการในการแยกกลุ่มรูปว่าในรูปใดบ้างที่อยู่ในกลุ่ม รูปใดบ้างที่อยู่ล้อมรอบรูปอื่นอยู่

ตัวอย่าง วาดรูปสี่เหลี่ยม 4 รูปโดยใช้โปรแกรม AutoCAD โดยกำหนดพิกัดของสี่เหลี่ยมรูปที่ 1 เป็น พิกัด(50,50) และ พิกัด(125,125)

กำหนดพิกัดของสี่เหลี่ยมรูปที่ 2 เป็น พิกัด(75,75) และ พิกัด(100,100)

กำหนดพิกัดของสี่เหลี่ยมรูปที่ 3 เป็น พิกัด(150,50) และ พิกัด(225,125)

กำหนดพิกัดของสี่เหลี่ยมรูปที่ 4 เป็น พิกัด(175,75) และ พิกัด(200,100) ดังรูป



ภาพที่ 4.10 แสดงสี่เหลี่ยม 4 รูป โดยมาการซ้อนกันของสี่เหลี่ยม

โดยโปรแกรมจะนำตัวแปร ORDER(a,b), PICX(a,b), PICY(a,b), SEQUENCEPIC(a), NOPIC มาใช้ในการวิเคราะห์ เพื่อหาค่าตัวแปร XMIN(a), XMAX(a), YMIN(a), YMAX(a) ของแต่ละรูป จากนั้นโปรแกรมจะนำตัวแปร XMIN(a), XMAX(a), YMIN(a), YMAX(a) มาวิเคราะห์เพื่อที่จะได้ทราบว่ารูปใดอยู่ในสุด รูปใดบ้างที่ล้อมรอบรูปอื่นอยู่ โดยค่าที่เก็บในตัวแปรจะเป็น 20

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เท่าของค่าพิกัด เนื่องจากที่ Set ในโปรแกรม AutoCAD และก่อนที่จะเก็บค่าในตารางจะทำการคูณด้วย 5 อีกทีนึง เนื่องจาก 100 step ของมอเตอร์จะเท่ากับระยะทาง 1 mm

จากรูป ค่า XMIN(1) = 5000

XMAX(1) = 12500

YMIN(1) = 5000

YMAX(1) = 12500

XMIN(2) = 7500

XMAX(2) = 10000

YMIN(2) = 7500

YMAX(2) = 10000

XMIN(3) = 15000

XMAX(3) = 22500

YMIN(3) = 5000

YMAX(3) = 12500

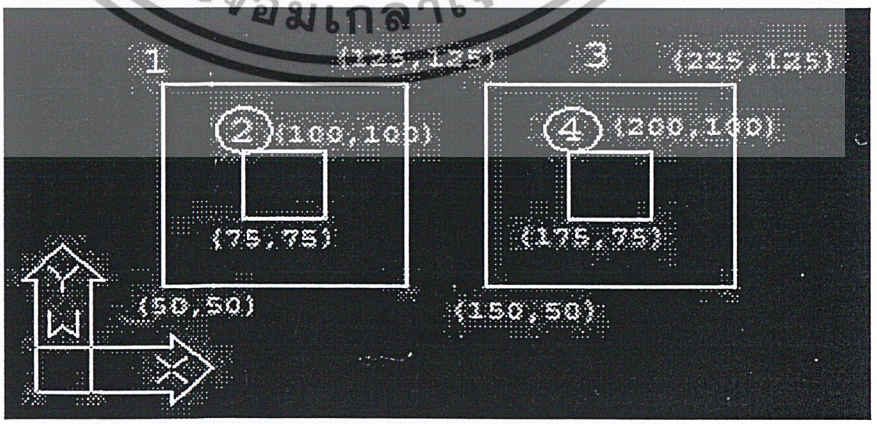
XMIN(4) = 17500

XMAX(4) = 20000

YMIN(4) = 7500

YMAX(4) = 10000

ในขั้นตอนการตัดนั้น เราต้องเริ่มตัดจากรูปวงในก่อนถึงจะตัดรูปในวงถัดมาและจึงจะตัดรูปในวงถัดมาได้ จะทำให้เราควรจะหารูปที่อยู่ในสุดออกมาก่อน โปรแกรมจะนำรูปที่อยู่ในสุดมาจัดเป็นกลุ่มตามจำนวนรูปที่อยู่ในสุด



ภาพที่ 4.11 แสดงรูปสี่เหลี่ยม 4 รูปที่ซ้อนกัน และมีการจัดเป็นกลุ่มโดยจากรูปที่อยู่ในสุด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูป จะได้รูปที่อยู่ในที่สุด ที่ไม่มีรูปอื่นอยู่ข้างในอีก คือ รูปที่ 2 และ รูปที่ 4 จะได้ กลุ่มรูป 2 กลุ่ม คือ กลุ่มที่1 มี รูปที่ 2 และ กลุ่มที่ 2 มีรูปที่ 4

จะทำให้ค่า XMINGROUP(1) = 7500

XMAXGROUP(1) = 10000

YMINGROUP(1) = 7500

YMAXGROUP(1) = 10000

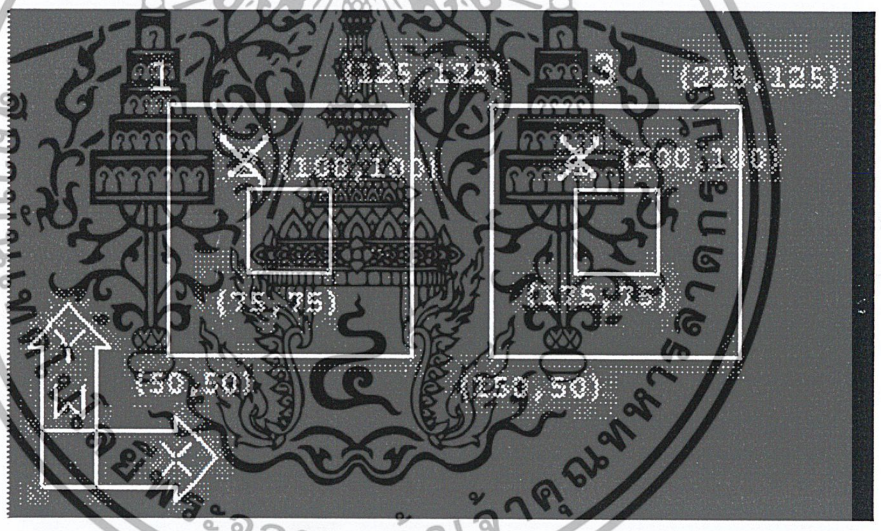
XMINGROUP(2) = 17500

XMAXGROUP(2) = 20000

YMINGROUP(2) = 7500

YMAXGROUP(2) = 10000

จากนั้นโปรแกรมจะวิเคราะห์หารูปที่อยู่ในสุดของรูปที่เหลือจากขั้นตอนก่อนหน้านี้ มาตรวจดูว่าล้อมรอบรูปกลุ่มใดบ้าง



ภาพที่ 4.12 แสดงว่า รูป 2 กับ 4 ได้ใช้ไว้แล้ว

จากรูป รูปที่ 2 และรูปที่ 4 ได้ถูกใช้ไปแล้วในขั้นตอนแรก ดังนั้นโปรแกรมจึงหารูปที่อยู่ในสุดของรูปที่เหลือ คือ รูปที่ 1 และ รูปที่ 3

โปรแกรมจะแยกได้ว่า รูป 1 ล้อมรูป 2 อยู่ และ รูป 3 ล้อมรูป 4 อยู่ ซึ่งจะทำให้เงื่อนไขของ XMIN(a)<XMIN(b) และ YMIN(a)<YMIN(b) และ XMAX(a)>XMAX(b) และ YMAX(a) > YMAX(b) เป็นจริงทำให้เราสรุปได้ว่า รูป a นั้นล้อมรอบรูป b อยู่ ซึ่งถ้าเงื่อนไขเป็นจริงไม่ครบทุกกรณี จะทำให้สรุปไม่ได้ว่ารูป a ล้อมรูป b หรือไม่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากนั้นโปรแกรมจะทำการยุบกลุ่มโดยจะรวมรูปที่ล้อมกลุ่มและรูปกลุ่มที่ถูกล้อมเข้าด้วยกัน รวมเป็นกลุ่มเดียวกัน

โปรแกรมจะทำการรวมรูปที่ 1 เข้ากับกลุ่มที่ 1 ดังนั้นกลุ่มที่ 1 จะมีรูปที่ 2 และรูปที่ 1 และโปรแกรมจะทำการรวมรูปที่ 3 เข้ากับกลุ่มที่ 2 ดังนั้นกลุ่มที่ 2 จะมีรูปที่ 3 และรูป ที่ 4 และโปรแกรมจะทำการยุบกลุ่มรวมไปเรื่อยๆ จนกว่าจะใช้รูปครบทุกรูป

ข้อสังเกต จำนวนกลุ่มมีแต่จะลดลงหรือเท่าเดิม ไม่มีการเพิ่มขึ้น

จะทำให้ค่า XMINGROUP(1) = 5000
XMAXGROUP(1) = 12500
YMINGROUP(1) = 5000
YMAXGROUP(1) = 12500
XMINGROUP(2) = 15000
XMAXGROUP(2) = 22500
YMINGROUP(2) = 5000
YMAXGROUP(2) = 12500

เมื่อได้จำนวนกลุ่มทั้งหมดแล้ว โปรแกรมจะทำการวิเคราะห์ว่า ทางเดินควรจะเป็นอย่างไร ควรจะไปที่กลุ่มไหนก่อน โดย โปรแกรมจะคำนวณระยะทางจากจุดต้นยังจุดปลายต่างๆ ว่าระยะใดเป็นระยะที่ใกล้ที่สุด จึงจะให้ทางเดินไปจุดนั้นก่อน

โดย ระยะทาง^2 = $(X1-X2)^2+(Y1-Y2)^2$

การคิดระยะทางแบ่งได้เป็น 2 กรณี

1.กรณีเริ่มต้นจุดต้นจุดแรกจะเป็นจุด (0,0) จุดปลายจะเป็นจุดอื่นๆของรูปกลุ่มอื่น

2.กรณีที่จุดต้นไม่ใช่จุด (0,0) จะใช้จุดปลายของกลุ่มที่คิดก่อนหน้านี้เป็นจุดต้น และจุดปลายจะเป็นจุดอื่นๆของรูปกลุ่มอื่นที่เหลือ โปรแกรมจะทำการเรียงลำดับกลุ่มจนกระทั่งครบทุกกลุ่ม จากนั้นจะนำกลุ่มที่เรียงลำดับแล้วนี้ ไปแปลงค่าตัวแปร ORDER, X, Y จากพิกัด Absolute เป็น พิกัด Increment ซึ่งจะมีผลเปลี่ยนลำดับของรูปด้วย ตามลำดับของรูปตามลำดับของกลุ่มที่โปรแกรมเรียงมาด้วย

โดยโปรแกรมจะนำตัวแปร ORDER(a,b), PICX(a,b), PICY(a,b) มาจัดลำดับใหม่ตามที่ได้เรียงลำดับกลุ่มไว้แล้ว ซึ่งหมายถึง ได้รูปเรียงตามลำดับจนครบทุกรูป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรมจะนำค่าในตัวแปรมาคำนวณเพื่อที่จะเปลี่ยนจากค่าพิกัด X, Y แบบAbsolute เป็นแบบพิกัด X, Y แบบIncrement โดยจะแบ่งเป็น 3 กรณี ดังนี้

1.กรณีที่ ORDER เป็น PU จะใช้เป็น พิกัด (0,0) เหมือนเดิม

2.กรณีที่ ORDER เป็น PD จะใช้เป็น พิกัด (0,0) เหมือนเดิม

3.กรณีที่ ORDER เป็น PA ครั้งแรก คิดที่พิกัด X, Y ของ ORDER PA คู่แรก ใช้ค่าพิกัดเดิมเลย เนื่องจากเป็นการลบกันระหว่างค่าพิกัด X, Y นั้นกับพิกัด (0,0)

กรณีที่ ORDER เป็น PA ครั้งต่อๆไป คิดที่พิกัด X, Y ของ ORDER PA ใช้ค่าพิกัดตัวนั้นลบด้วยค่าพิกัด X, Yบรรทัดที่มี ORDER PA ก่อนหน้านั้น 1 บรรทัด

โดยโปรแกรมจะนำค่าที่คำนวณได้ ไปต่อท้ายตารางโดยที่จะยังไม่ลบค่าตัวแปร ORDER, X, Y เก่าที่อยู่ในรูปพิกัด Absolute

โปรแกรมจะทำการเพิ่มคำสั่งเข้าในตารางต่อจากตารางที่ได้มาจากข้างต้น เพื่อที่จะให้ทางเดินไปจบที่จุด (0,0) ซึ่งจะเป็นการจบคำสั่ง

โปรแกรมจะทำการลบคำสั่งในตาราง เพื่อลบคำสั่งในส่วนของพิกัด Absolute ออกทั้งหมด ทำให้เหลือค่าตัวแปรในตารางเฉพาะในส่วนของพิกัด Increment

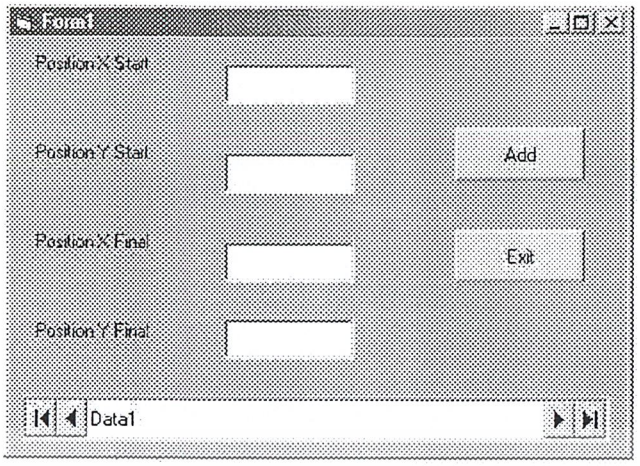
4.3 โปรแกรมโหมดManual

โหมดmanualนี้จะนำไปใช้งานกับการที่วาดแบบของแผ่นเหล็กในตัวโปรแกรมนี้เลยไม่ต้องไปเขียนในโปรแกรม AutoCAD จะช่วยอำนวยความสะดวกแก่ผู้ใช้งาน โดยการทำงานในโหมดนี้จะมีสำหรับการวาดเป็นแบบ เส้นตรง, วงกลม, วงรี

ปุ่มเส้นตรงซึ่งการใช้งานนั้นเมื่อคลิกปุ่ม เส้นตรง จะมีหน้าต่างปรากฏและจะให้ใส่ข้อมูลดังนี้คือ

- Position X Start คือระยะที่พิกัดของแกน X เริ่มต้น
- Position Y Start คือระยะที่พิกัดของแกน Y เริ่มต้น
- Position X Final คือระยะที่พิกัดของแกน X สุดท้าย
- Position Y Final คือระยะที่พิกัดของแกน Y สุดท้าย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 4.13 แสดง Form ฟังก์ชั่นเส้นตรงของโหมด Manual

ซึ่งเมื่อรับค่าจากผู้ใช้มาแล้วก็จะส่งค่าเข้าตารางเมื่อกดปุ่ม Add จะเก็บค่าต่างดังนี้

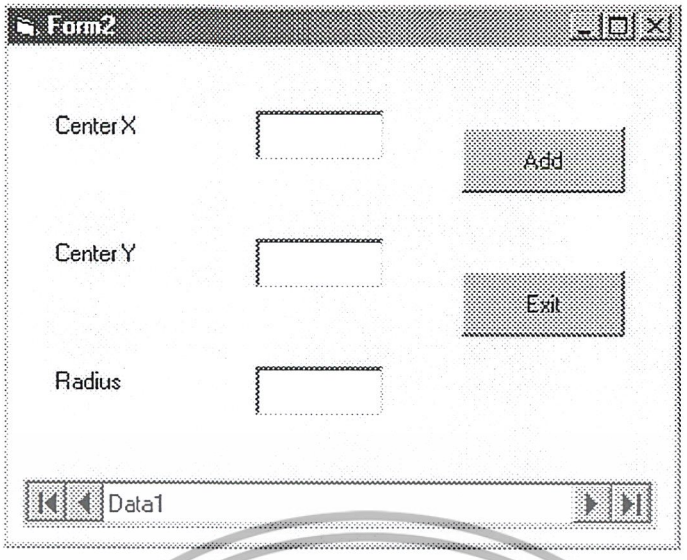
- Position X จะเก็บค่าเข้าตารางในคอลัมน์ X
- Position Y จะเก็บค่าเข้าตารางในคอลัมน์ Y

และโปรแกรมจะทำการใส่ค่าของ No ซึ่งคือลำดับของค่าต่างและจะใส่ค่า Order ซึ่งก็คือค่าที่จะให้ Plasma ยกขึ้น (“PU”) ค่าที่จะให้ Plasma ยกลง (“PD”) ค่าที่ให้ Plasma ทำการวิ่งตามพิกัดที่ได้กำหนดไว้ (“PA”)

ปุ่มวงกลมซึ่งการใช้งานนั้นเมื่อคลิกปุ่ม เส้นตรง จะมีหน้าต่างปรากฏและจะให้ใส่ข้อมูลดังนี้คือ

- Center X คือ จุดศูนย์กลางที่พิกัด Xของวงกลมที่จะทำการวาด
- Center Y คือ จุดศูนย์กลางที่พิกัด Yของวงกลมที่จะทำการวาด
- Radius คือ รัศมีของวงกลมที่จะทำการวาด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพ 4.14 แสดงฟังก์ชั่นวงกลมของโหมด Manual

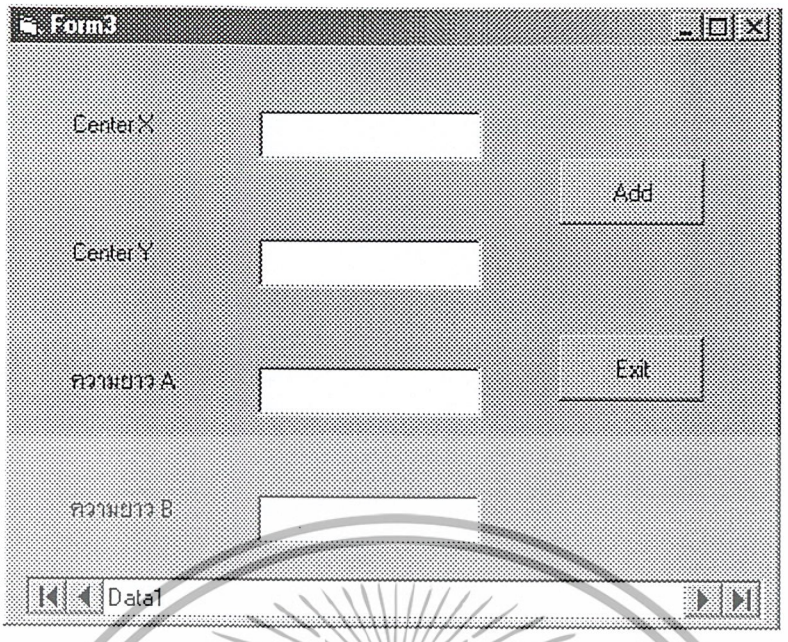
ซึ่งเมื่อรับค่าจากผู้ใช้มาแล้วก็จะส่งค่าเข้าตารางเมื่อกดปุ่ม Add จะเก็บค่าต่างดังนี้

โปรแกรมจะทำการรับค่า Center X, Center Y, Radius แล้วจะเข้าโปรแกรมมาทำการคำนวณออกมาเป็นพิกัด X และ พิกัด Y จากนั้นจะทำการเก็บค่าพิกัด X เข้าตาราง X และพิกัด Y เข้าตาราง Y แล้วโปรแกรมจะทำการใส่ค่าของ No ซึ่งคือลำดับของค่าต่างและจะใส่ค่า Order ซึ่งก็คือค่าที่จะให้ Plasma ยกขึ้น (“PU”) ค่าที่จะให้ Plasma ยกลง (“PD”) ค่าที่ให้ Plasma ทำการวิ่งตามพิกัดที่ได้กำหนดไว้ (“PA”)

ปุ่มวงรีซึ่งการใช้งานนั้นเมื่อคลิ้กปุ่ม เส้นตรง จะมีหน้าต่างปรากฏและจะให้ใส่ข้อมูลดังนี้คือ

- Center X คือ จุดศูนย์กลางที่พิกัด Xของวงรีที่จะทำการวาด
- Center Y คือ จุดศูนย์กลางที่พิกัด Yของวงรีที่จะทำการวาด
- ความยาว A คือ คือระยะห่างของค่า A
- ความยาว B คือ คือระยะห่างของค่า B

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 4.15 แสดงฟังก์ชั่นวงรีของโหมด Manual

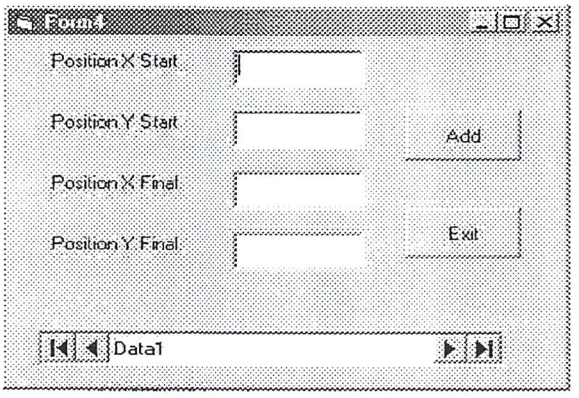
ซึ่งเมื่อรับค่าจากผู้ใช้มาแล้วก็จะส่งค่าเข้าตารางเมื่อกดปุ่ม Add จะเก็บค่าต่างดังนี้

โปรแกรมจะทำการรับค่า Center X, Center Y, ความยาว A, ความยาว B แล้วจะเข้าโปรแกรมมาทำการคำนวณออกมาเป็นพิกัด X และ พิกัด Y จากนั้นจะทำการเก็บค่าพิกัด X เข้าตาราง X และพิกัด Y เข้าตาราง Y แล้วโปรแกรมจะทำการใส่ค่าของ No ซึ่งคือลำดับของค่าต่างและจะใส่ค่า Order ซึ่งก็คือค่าที่จะให้ Plasma ยกขึ้น (“PU”) ค่าที่จะให้ Plasma ยกลง (“PD”) ค่าที่ให้ Plasma ทำการวิ่งตามพิกัดที่ได้กำหนดไว้ (“PA”)

ปุ่มเส้นโค้งซึ่งการใช้งานนั้นเมื่อคลิ้กปุ่ม เส้นตรง จะมีหน้าต่างปรากฏและจะให้ใส่ข้อมูลดังนี้คือ

- Position X Start คือระยะที่พิกัดของแกน X เริ่มต้น
- Position Y Start คือระยะที่พิกัดของแกน Y เริ่มต้น
- Position X Final คือระยะที่พิกัดของแกน X สุดท้าย
- Position Y Final คือระยะที่พิกัดของแกน Y สุดท้าย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 4.16 แสดงฟังก์ชั่นเส้นโค้งของโหมด Manual

4.4 โปรแกรมส่วนทั่วๆไป

ปุ่มของโปรแกรมของส่วนทั่วๆ ไป มีทั้งหมด 3 ปุ่ม คือ

1.ปุ่มดูค่าในตาราง

2.ปุ่มลบค่าในตาราง

3.ปุ่มวาดรูป

4.ปุ่มลบรูป

5.ปุ่มส่งค่าจากตารางไปยังระบบควบคุมมอเตอร์

รายละเอียดการทำงานของปุ่มทั้ง 5 ปุ่ม

1.ปุ่มดูค่าในตาราง

เป็นปุ่มกดสำหรับดูค่าที่เก็บไว้ในตาราง เพื่อตรวจสอบว่าค่าในตารางนั้นถูกต้องหรือไม่

2.ปุ่มลบค่าในตาราง

เป็นปุ่มกดสำหรับลบค่าในตารางเพื่อทำให้เป็นตารางว่าง

3.ปุ่มวาดรูป

เป็นปุ่มกดสำหรับนำค่าในตารางซึ่งอยู่ในรูปพิกัด Increment มาวาดรูปใน Control Picture เพื่อเป็นการจำลองทางเดินของเครื่องมือตัดก่อนการตัดจริง เป็นการตรวจสอบว่าพิกัดในตารางนั้นเป็นไปตามที่ต้องการหรือไม่

4.ปุ่มลบรูป

เป็นปุ่มกดสำหรับลบภาพที่วาดไว้ใน Control Picture

5.ปุ่มส่งค่าจากตารางไปยังระบบควบคุมมอเตอร์

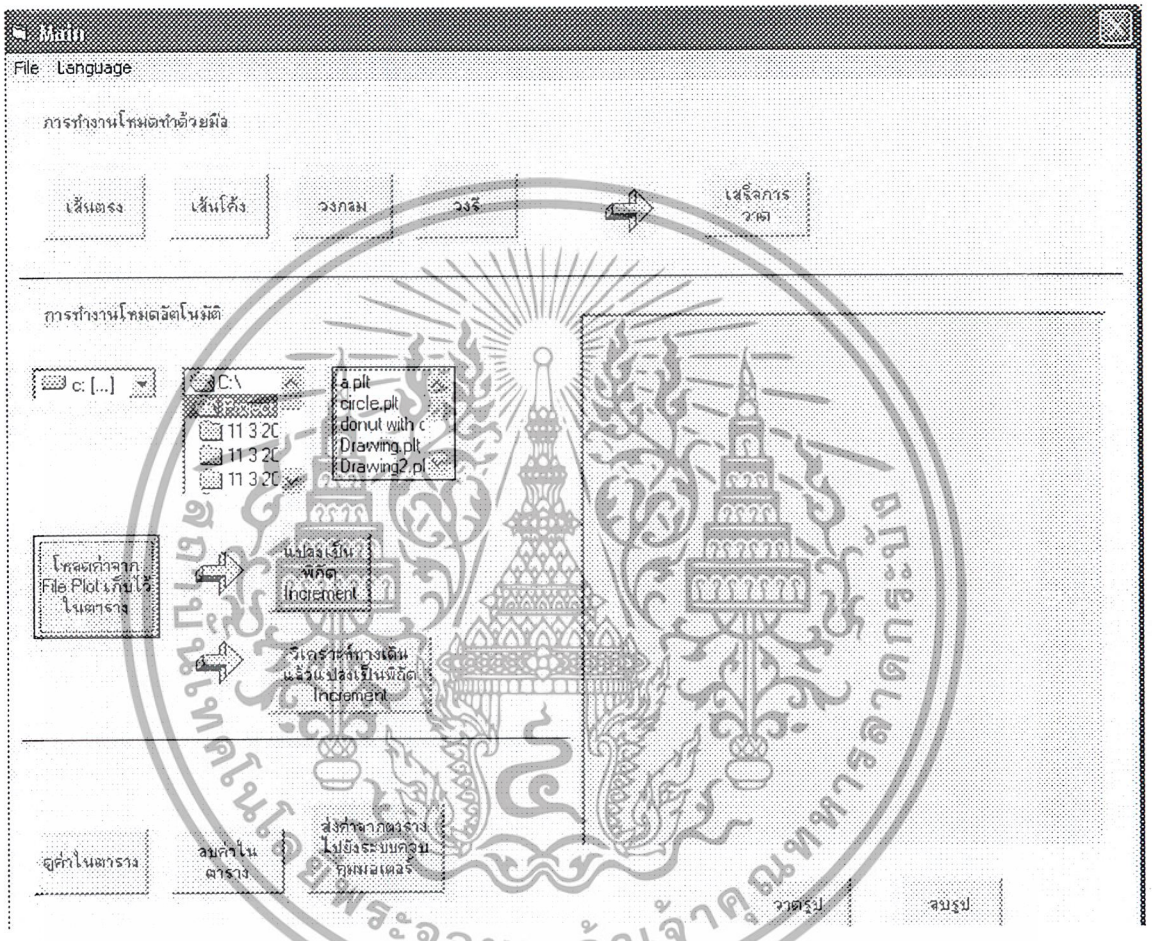
เป็นปุ่มกดสำหรับส่งค่าพิกัด Increment ในตารางไปยังระบบควบคุมมอเตอร์เพื่อบังคับให้ทางเดินของเครื่องมือตัดเป็นไปตามที่ต้อง โดยปกติแล้วควรผ่านการทดสอบทางเดินของเครื่องมือโดยการวาดรูปก่อน เพื่อความถูกต้องในการตัด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

ผลการทดลอง

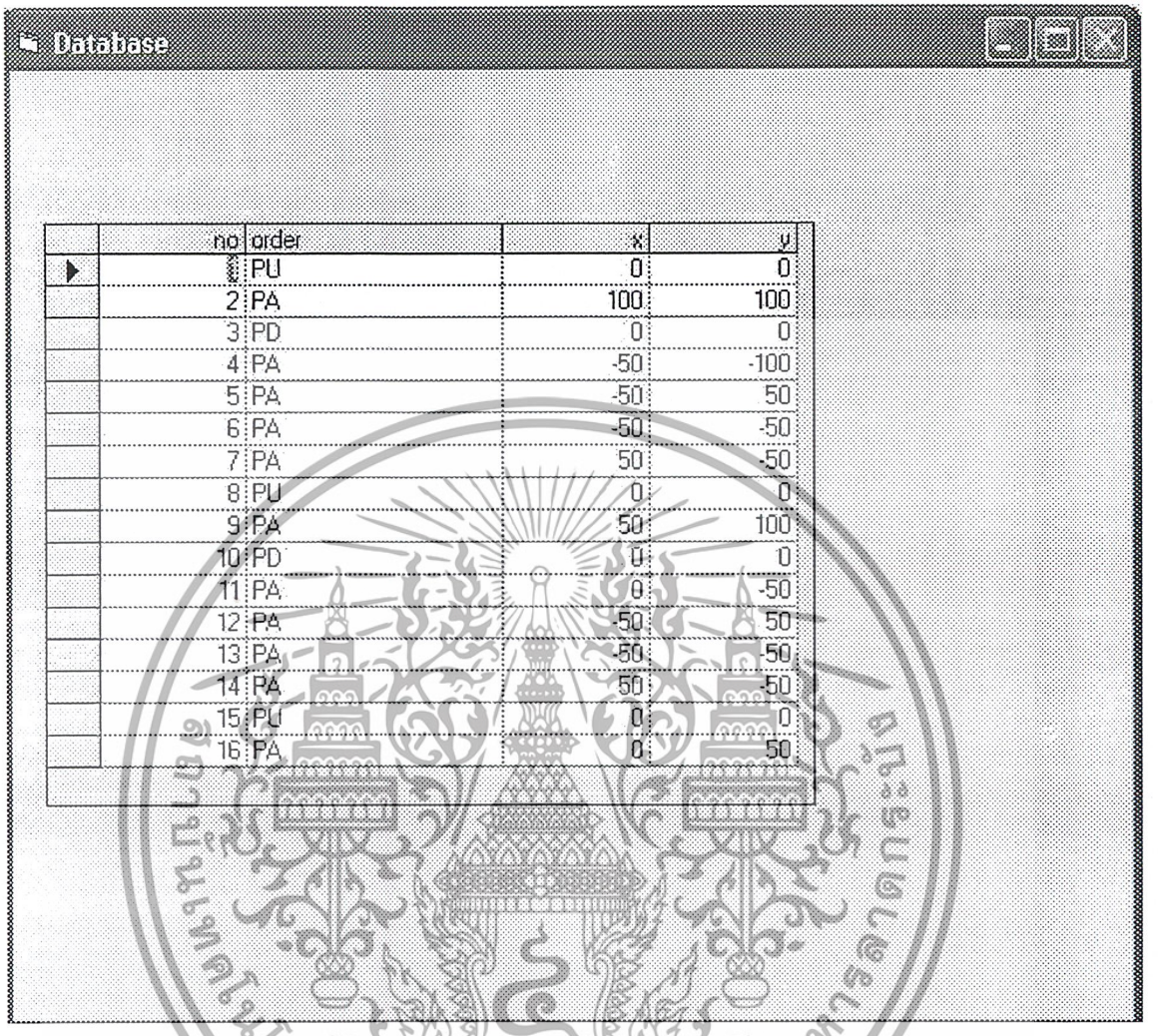
ภาพหน้าต่างของโปรแกรม Visual Basic



ภาพที่ 5.1 แสดงหน้าจอของโปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาพหน้าต่างของโปรแกรมส่วนตาราง เมื่อกดปุ่มดูค่าในตาราง



ภาพที่ 5.2 แสดงหน้าต่างของโปรแกรมส่วนตาราง เมื่อกดปุ่มดูค่าในตาราง

5.1 ผลจากโหมด Auto

วิธีการทดสอบ

1.วาดรูปสี่เหลี่ยมมีมุมที่พิกัด (10,10) และ (290,290)

2.วาดรูปสี่เหลี่ยมมีมุมที่พิกัด (25,75) และ (75,100)

3.วาดรูปวงกลมที่มีจุดศูนย์กลางที่พิกัด (50,200) และ มีรัศมี 25 หน่วย

4.วาดรูปสี่เหลี่ยมมีมุมที่พิกัด (100,25) และ (275,275)

5.วาดรูปวงกลมที่มีจุดศูนย์กลางที่พิกัด (200,200) และ มีรัศมี 50 หน่วย

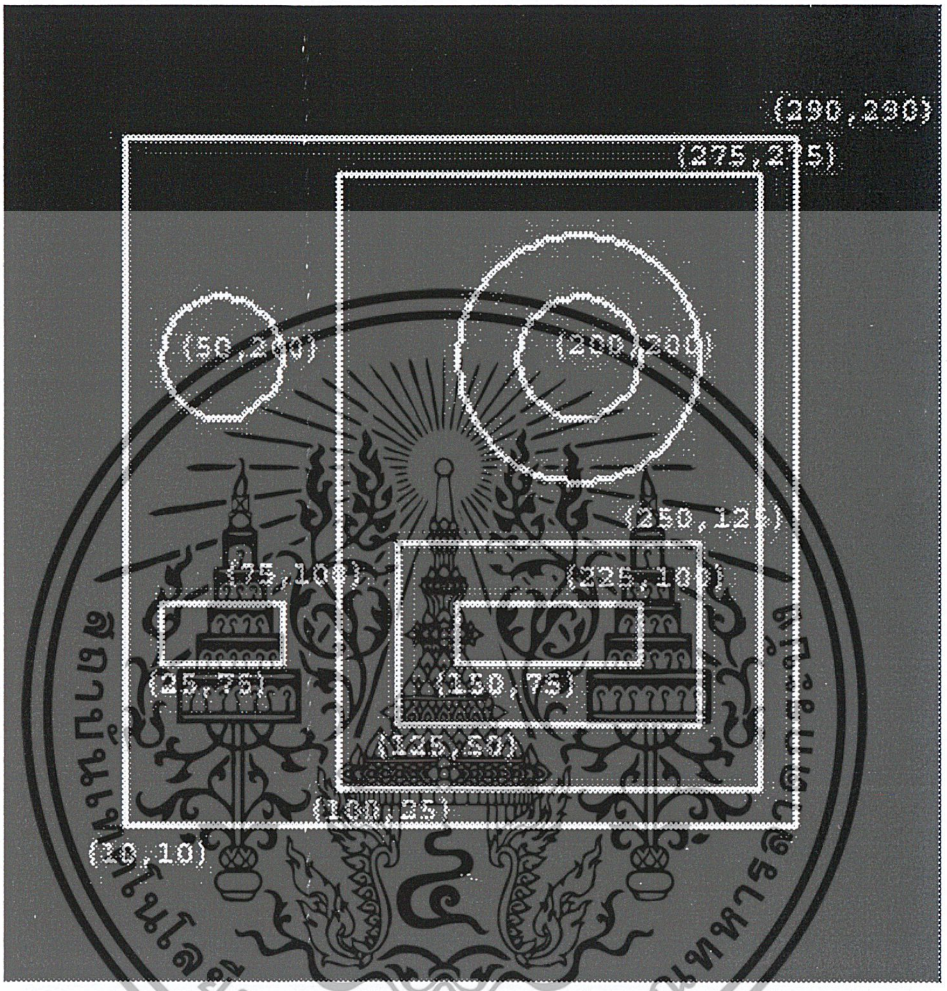
6.วาดรูปสี่เหลี่ยมมีมุมที่พิกัด (125,50) และ (250,125)

7.วาดรูปวงกลมที่มีจุดศูนย์กลางที่พิกัด (200,200) และ มีรัศมี 25 หน่วย

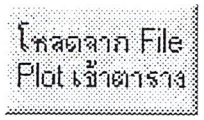
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

8.วาดรูปสี่เหลี่ยมมีมุมที่พิกัด (150,75) และ (225,100)

จะได้ภาพในโปรแกรม AutoCAD ดังรูป

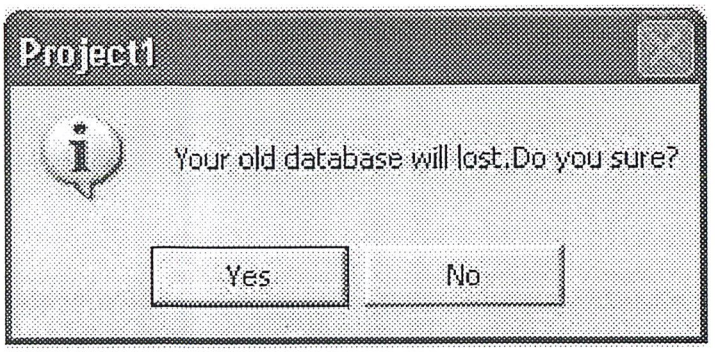


ภาพที่ 5.3 แสดงภาพในโปรแกรม AutoCAD ที่มาจากค่าในตาราง

9.กดปุ่มโหลดจาก File Plot เข้าตาราง 

ถ้าในตารางมีค่าตัวแปรเก่าอยู่ให้กด “Yes”

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 5.4 แสดงเมื่อตารางมีค่าตัวแปรเก่าอยู่

โปรแกรมจะทำการโหลดข้อมูลจาก File Plot เข้ามาเก็บไว้ในตารางในรูปแบบพิกัด Absolute

10.กดปุ่มดูค่าในตาราง

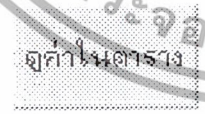
ภาพที่ 5.5 แสดงค่าในตารางที่เป็นพิกัดในระบบ Absolute

จะได้ดังภาพ ซึ่งจะอยู่ในพิกัด Absolute

11.กดปุ่มแปลงเป็นพิกัด Increment

โปรแกรมจะนำข้อมูลในตารางซึ่งเป็นพิกัด Absolute มาแปลงเป็นพิกัด Increment แล้วเก็บไว้ใน

ตารางเดิมโดยค่าพิกัดที่เป็นพิกัด Absolute จะถูกลบทิ้งไป

12.กดปุ่มดูค่าในตาราง 

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

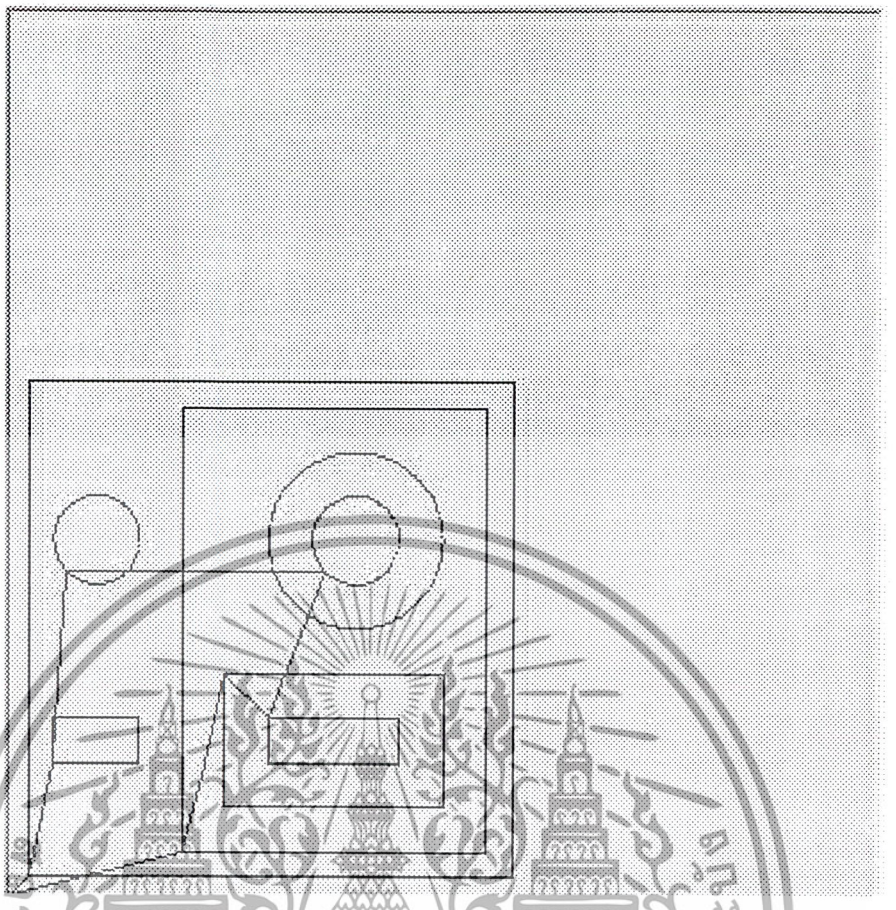
	no	order	x	y
▶	1	PU	0	0
	2	PA	2500	7500
	3	PD	0	0
	4	PA	5000	0
	5	PA	0	2500
	6	PA	-5000	0
	7	PA	0	-2500
	8	PU	0	0
	9	PA	730	10730
	10	PD	0	0
	11	PA	160	-145
	12	PA	175	-135
	13	PA	185	-115
	14	PA	190	-100
	15	PA	205	-85
	16	PA	205	-65

ภาพที่ 5.6 แสดงค่าในตารางที่อยู่ในระบบ Increment

จะได้ดังภาพ ซึ่งอยู่ในพิกัด Increment

13.กดปุ่มวาดรูป วาดรูป โปรแกรมจะนำข้อมูลจากตาราง ซึ่งอยู่ในพิกัด Increment มาวาดบน Control Picture ดังรูป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



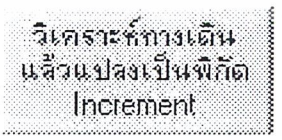
ภาพที่ 5.7 แสดงการวาดรูปจากค่าในตารางที่ยังไม่ได้วิเคราะห์

แต่ลำดับในการวาดยังใช้ในการตัดจริงไม่ได้ เนื่องจาก File Plot ที่ AutoCAD เรียงให้ไม่ได้เรียงลำดับให้เริ่มตัดจากวงในยังวงนอก

14.จากนั้นกดปุ่ม ลบรูป จะทำให้รูปที่วาดไว้บน Control Picture หายไป เพื่อทำการทดลองอีกปุ่มหนึ่งคือ ปุ่มวิเคราะห์ทางเดินแล้วแปลงเป็นพิกัด Increment

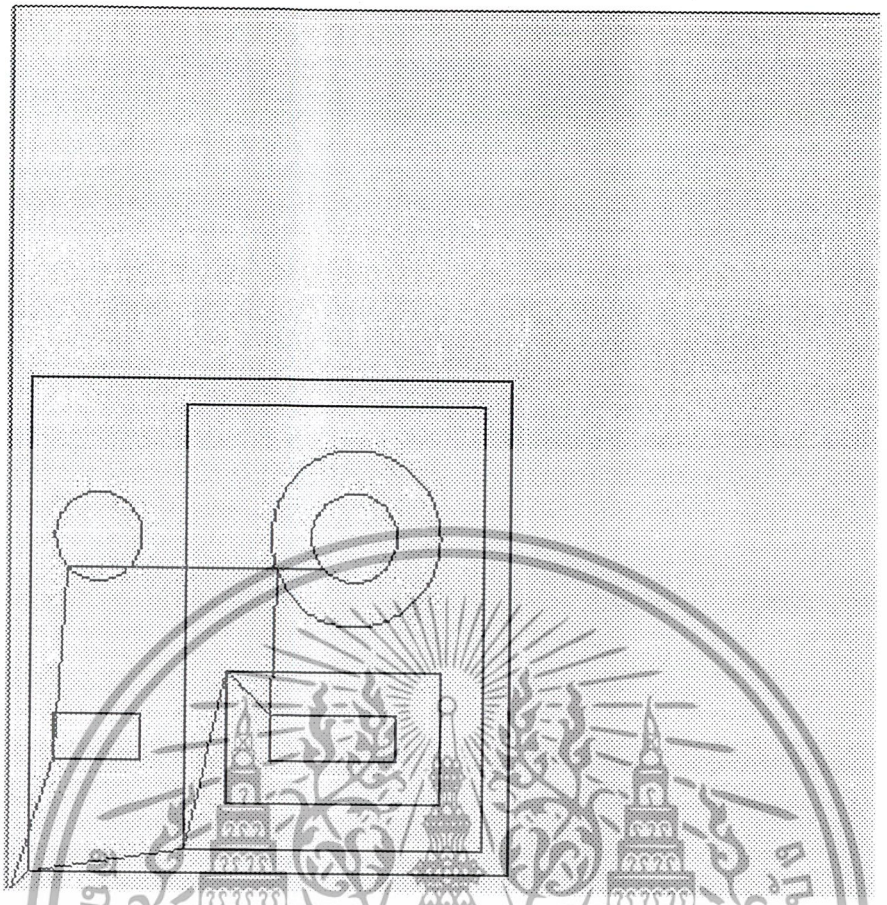
15.กดปุ่มโหลดจาก File Plot เข้าตาราง แล้ว ตอบ "Yes"

16.กดปุ่มวิเคราะห์ทางเดินแล้วแปลงเป็นพิกัด Increment



17.กดปุ่มวาดรูป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 5.8 แสดงการวาดภาพจากค่าในตารางที่ผ่านการวิเคราะห์มาแล้ว

สังเกตได้ว่าลำดับการวาดรูปของ Control Picture จะเปลี่ยนไปเป็นไปตามที่โปรแกรมวิเคราะห์

18.กดปุ่มปิดโปรแกรมที่รูป ด้านขวาบนของโปรแกรม

เพื่อออกจากโปรแกรม

5.2 ผลการทดลองโหมด Manual

จากทดลองโดยใช้ฟังก์ชั่นเส้นตรงเราจะทำการ Plot เส้นตรงจากพิกัด (100,100) ไป (100,150)

และ ไปยัง (150,150) ซึ่งค่าที่เก็บในตารางจะเป็นระบบ Absolute ดังรูป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

no	order	x	y
1	PU	0	0
2	PA	4000	4000
3	PD	0	0
4	PA	4000	6000
5	PA	6000	6000
6	PU	0	0
7	PA	0	0

ภาพที่ 5.9 แสดงค่าในตารางที่เป็นระบบ Absolute

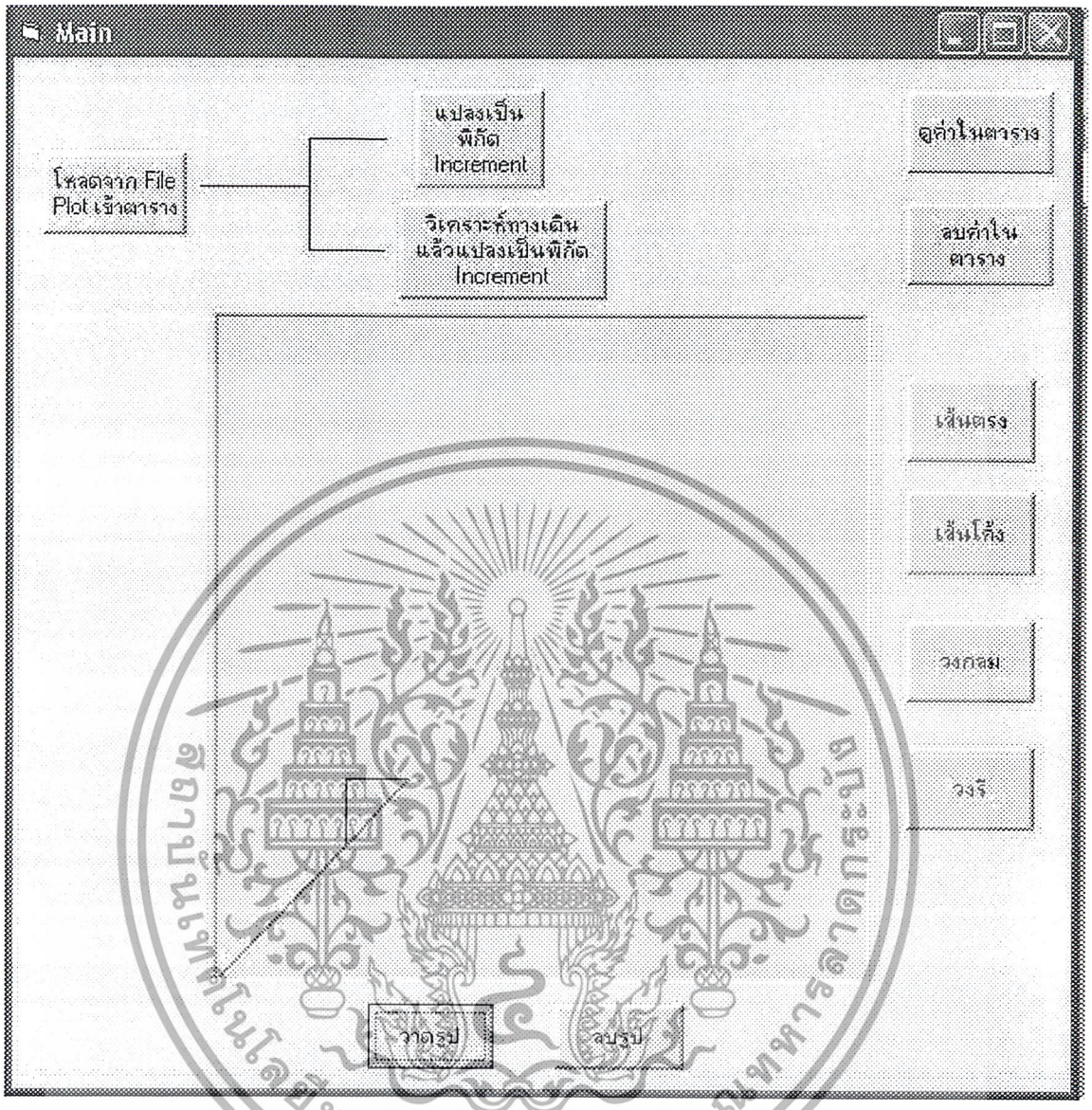
และเมื่อทำการกดปุ่มเพื่อทำการแปลงค่าในตารางจาก Absolute เป็น Increment จะได้ผลลัพธ์ดังรูป

no	order	x	y
1	PU	0	0
2	PA	4000	4000
3	PD	0	0
4	PA	0	2000
5	PA	2000	0
6	PU	0	0
7	PA	-6000	-6000

ภาพที่ 5.10 แสดงค่าในตารางที่เปลี่ยนเป็นระบบ Increment

หลังจากนั้นเมื่อทำการกดปุ่มวาดรูปจะได้ผลลัพธ์ดังรูป

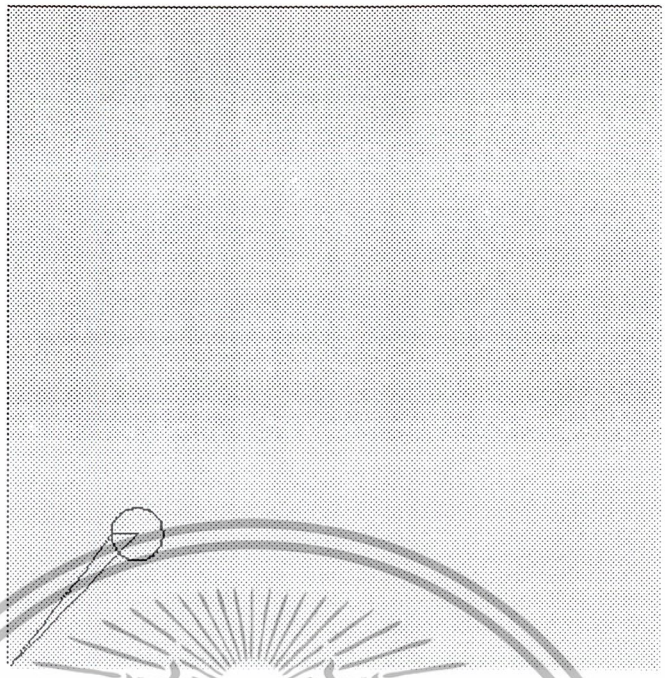
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 5.11 แสดงกาววาดภาพจากค่าในตารางของฟังก์ชั่นเส้นตรงในโหมด Manual

และเมื่อทำการใช้ปุ่มฟังก์ชั่นวงกลมโดยกำหนดพิกัดจุดศูนย์กลาง (100, 100) และมีรัศมี 20 เมื่อทำการใส่ข้อมูลและทำการแปลงข้อมูลในตารางจากระบบ Absolute เป็นระบบ Increment แล้ว ทำการ Plot จะได้ดังรูป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 5.12 แสดงการวาดภาพของฟังก์ชั่นวงกลมใน โหมด Manual

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 6
สรุปผลและข้อจำกัดของปริญญานิพนธ์

6.1 สรุปผล

โปรแกรมสามารถทำงานได้ทั้งโหมดการทำงานแบบ Manual และ Auto

โดยในโหมดการทำงานแบบ Manual เมื่อวาดรูปในโปรแกรมเสร็จแล้ว สามารถเลือกให้ส่งค่าพารามิเตอร์ตามลำดับที่วาดไปยัง Controller หรือจะให้โปรแกรมจัดทางเดินให้ใหม่โดยเรียงลำดับจากวงในไปยังวงนอกและเคลื่อนที่ด้วยระยะทางที่น้อยลง

โดยในโหมดการทำงานแบบ Auto เมื่อโหลดข้อมูลจาก File Plot ซึ่ง Print มาจากโปรแกรม AutoCAD แล้วสามารถเลือกให้ส่งค่าพารามิเตอร์ตามลำดับที่โปรแกรม AutoCAD ตั้งให้ไปยัง Controller หรือจะให้โปรแกรมจัดทางเดินให้ใหม่โดยเรียงลำดับจากวงในไปยังวงนอกและเคลื่อนที่ด้วยระยะทางที่น้อยลง

6.2 ข้อจำกัดของโครงงาน

6.2.1 ส่วนของการเคลื่อนที่

- ระยะการเคลื่อนที่ไม่เกิน 1*1 เมตร
- ความเร็วในการตัด
- ไม่มีระบบตรวจสอบระยะทาง

6.2.2 ส่วนของโปรแกรม

- ข้อมูล File Plot ของโปรแกรม AutoCAD จะต้องตั้งค่าพารามิเตอร์ตามที่กำหนด
- จะมีการปัดเศษทศนิยมได้ 1 ตำแหน่งของมิลลิเมตร

6.2.3 ส่วนของ Controller

- มีการรับอินพุทเป็นจำนวน Step เป็นเลขจำนวนเต็ม

6.3 ปัญหาของโครงงาน

ส่วนของการจำลองวาดทางเดินในโปรแกรมยังไม่ดี เนื่องจากความละเอียดของ File Plot ในโปรแกรม AutoCAD และความละเอียดของ Control Picture ในโปรแกรม Visual Basic

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

1.“ Visual Basic ฉบับโปรแกรมเมอร์” , กิตติ ภักดีวัฒนะกุล – จำลอง ครูอุตสาหะ

2.“ คู่มือเรียน Visual Basic 6”, ฉัททวุฒิ พีชผล - พิชิต สันติกุลานนท์ Provision

3.“คู่มือการใช้โปรแกรม AutoCAD RELEASE 14 (FOR WINDOWS 95 & NT) 3D MODELING”, ภาณุพงษ์ ปัตติสิงห์

4.“เรียนรู้และปฏิบัติการเชื่อมต่อคอมพิวเตอร์กับอุปกรณ์ภายนอกผ่านพอร์ตอนุกรม , อินโนเวตีฟ เอ็กเพอรี่เมนต์ จำกัด, กฤษดา ใจเย็น, อรรถพล บุญยโภดา, ชัยวัฒน์ ลิ้มพรวิไล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้