



โปรแกรมสำหรับการแสดงผลและควบคุมการทำงานของ PLC ที่สามารถออกแบบได้



นาย ธนาภิวัดน์ ยอศรีภักดิ์
นาย ชีระพล แพ่งวิเศษ
นาย สุทธิพงษ์ ฉนวนวุฒิสักดิ์

เลขเรียกหนังสือ ปพ นร ๑๖ ๑๖๓
เลขทะเบียน ๐๕๐๕๑๔
วัน เดือน ปี 18 ตค ๕๖

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิศวกรรมศาสตรบัณฑิต

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้เพื่อการศึกษาเท่านั้น เมื่ออนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปริญญาานิพนธ์ปีการศึกษา 2541

ภาควิชา เทคโนโลยีการวัดคุมทางอุตสาหกรรม

สาขาวิชา วิศวกรรมการวัดคุม

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง โปรแกรมการสำหรับการแสดงผล และ ควบคุมการทำงานของ PLC
ที่สามารถออกแบบได้

Programmable Software For Monitoring And Control With PLC

คณะผู้จัดทำ

นายชนาภิวัฒน์ ยอดรักษ์ รหัสประจำตัว 39012088

นายธีระพล แพงวิเศษ รหัสประจำตัว 39012089

นายสุทธิพงษ์ ถนอมวุฒิศักดิ์ รหัสประจำตัว 39012113

อาจารย์ที่ปรึกษา.....

(อาจารย์ทวีพล ชี้อัตถ์)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อวิทยานิพนธ์ : โปรแกรมสำหรับการแสดงผลและควบคุมการทำงานของ PLC ที่สามารถ
ออกแบบได้

โดย : นาย ธนาภวิวัฒน์ ยอศรีรักษ์
: นาย ชีระพล แพ่งวิเศษ
: นาย สุทธิพงษ์ ฉนวนวุฒิศักดิ์

ภาควิชา : เทคโนโลยีการวัดคุมทางอุตสาหกรรม

อาจารย์ที่ปรึกษา : อ. ทวีพล ชื้อสัตย์

ปีการศึกษา : 2541

บทคัดย่อ

ปัจจุบันได้มีการนำการควบคุมซีเควินซ์ มาใช้ในการควบคุมเครื่องจักร รวมถึงอุปกรณ์
ในสายการผลิตในโรงงานอุตสาหกรรมอย่างมาก โดยเครื่องควบคุมแบบตรรกที่โปรแกรมได้(PLC)
เป็นอุปกรณ์ที่ได้รับความนิยมอย่างมากในระบบโรงงานอัตโนมัติ

โปรแกรมนี้จะเป็โปรแกรมสำหรับการแสดงผลและควบคุมการทำงานของ PLC ที่
สามารถออกแบบได้ ซึ่งเป็นโปรแกรมที่จะสามารถนำมาใช้ในการออกแบบการควบคุมกระบวนการ
การผลิตในทางอุตสาหกรรมโดยผู้ที่ควบคุมกระบวนการผลิตสามารถที่จะควบคุมกระบวนการผลิต
ได้โดยใช้ คอมพิวเตอร์ เป็นตัวควบคุมกระบวนการผลิต และแสดงสถานะของกระบวนการผลิต

โดยได้ใช้โปรแกรม Visual Basic ในการออกแบบโปรแกรมสำหรับการแสดงผลและ
ควบคุมการทำงานของ PLC ที่สามารถออกแบบได้ โปรแกรมนี้จะรองรับการทำงานร่วมกับเครื่อง
ควบคุมแบบตรรก (PLC) การส่งข้อมูลระหว่างคอมพิวเตอร์และเครื่องควบคุมแบบตรรก (PLC)
โดยใช้มาตรฐานการสื่อสาร RS232-C และข้อตกลงการสื่อสาร โปรโตคอล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Project Report Title : Programmable Software For Monitoring And Control With PLC
By : Mr. Tanapiwat Yodruk
 : Mr. Terapol Pangwisate
 : Mr. Suttipong Thanomvootisak
Department : Industrial Instrumentation Technology
Project Report Advisor : Mr. Taweeapol Suesut
Academic : 1998



Abstract

Recently, sequence controller used to control machine included operating equipment within industrial more over. The Programmable Logic Control is device for a automatic plant.

This project presents programmable software for monitoring and control with PLC which use to design control process and display status of process by computer. This program can use in industrial process control.

By used program Visual Basic design this program, this program to support the operating of The Programmable Logic Control. Data transfers between computer and The Programmable Logic Control used communications protocol and using standard RS 232C interface.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

ในปริญญานิพนธ์เล่มนี้จึงขอกราบขอบพระคุณเป็นอย่างยิ่ง ในการทำโครงการปริญญานิพนธ์นี้ กว่าจะสำเร็จลุล่วงไปได้ต้องประสบกับปัญหามากมาย ทั้งที่เกี่ยวข้องกับตัววิทยานิพนธ์ และเกี่ยวกับสภาพแวดล้อมทุกด้าน แต่ก็ได้รับความช่วยเหลือจาก อ. ทวีพล ชื้อศักดิ์ คอยให้คำปรึกษา อีกทั้งยังได้รับความกรุณาในการตรวจทานแก้ไขปริญญานิพนธ์เล่มนี้

และสุดท้ายขอขอบพระคุณผู้ที่เกี่ยวข้องที่คอยให้คำปรึกษาและคอยช่วยเหลือในยามที่ท้อแท้ และช่วยทำให้จิตใจปลอดโปร่งซึ่งทำให้ทำงานต่อไปได้

ผู้จัดทำ



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

เรื่อง	หน้า
บทคัดย่อภาษาไทย	I
บทคัดย่อภาษาอังกฤษ	II
กิตติกรรมประกาศ	III
สารบัญ	IV
สารบัญภาพ	V
บทที่ 1 บทนำ	1
วัตถุประสงค์	
ขอบเขตของโครงการ	
ประโยชน์ที่คาดว่าจะได้รับ	
แผนการดำเนินงาน	
บทที่ 2 ทฤษฎีการทำงาน Visual Basic	3
- วิธีการใช้งานเอนไวรอนเมนต์ ของ Visual Basic	
- เครื่องมือกำหนดการสื่อสาร	18
- ไฟล์สำหรับ Visual Basic	29
- โครงสร้างและส่วนประกอบ PLC	34
- หน่วยความจำชนิดต่าง ๆ	37
- หน่วยอินพุต /เอาต์พุต (Input / output Unit)	38
การติดต่อสื่อสารข้อมูลทั่วไป	41
ข้อตกลงในการติดต่อสื่อสาร(โปรโตคอล : Protocols)	44
การควบคุมความผิดพลาดในการส่งข้อมูล	46
การสื่อสารข้อมูลแบบอนุกรมตามมาตรฐาน RS 232-C	48
ขั้นตอนการควบคุมการสื่อสาร	49
บทที่ 3 การออกแบบโปรแกรม	57
บทที่ 4 รูปแบบของโครงการ และผลการทดลองการทำงาน	65
บทที่ 5 สรุป	77
เอกสารประกอบที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า	78
ใบบรรณานุกรมทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้	

สารบัญภาพ

ภาพที่	หน้า
1. แสดงรูปแบบของ Visual Basic	4
2. เครื่องมือสร้าง ป๊อ๊กซ์ภาพ (PictureBox)	11
3. เครื่องมือสร้างภาพลักษณะ(Image)	11
4. เครื่องมือสร้าง เลเบล(Label)	13
5. เครื่องมือสร้าง เท็กซ์ป๊อ๊กซ์ (TextBox)	14
6. เครื่องมือสร้างคอมมานบัตทอน (CommandButton)	17
7. MSCOMM CONTROL	18
8. เครื่องมือสร้างตัวจับเวลา (Timer)	20
9. เครื่องมือสร้างคอมมอนไดอะล็อกซ์ (CommonDialog)	22
10. ไออะล็อกป๊อ๊กซ์ Open	23
11. ไออะล็อกป๊อ๊กซ์ Save As	25
12. วินโดว์ออกแบบเมนูของ Visual Basic	26
13. แบบฟอร์มพร้อมกับนูบาร์ และเมนูแบบคั้งลง	27
14. ป๊อฝอัปเมนู (Popuption)	29
15. แสดงโครงสร้างของ PLC	34
16. Flow Chart แสดงการทำงานของ CPU	35
17. แสดงความสำพันธ์ของส่วนประกอบหลักในการสื่อสารข้อมูล	41
18. การส่งแบบทิศทางเค็ยว (Simplex)	42
19. การส่งแบบฮาล์วดูเพิ้ล็กซ์ (Half Duplex)	42
20. การส่งข้อมูลแบบสองทิศทาง (Full Duplex)	42
21. การสื่อสารข้อมูลแบบขนาน	43
22. การสื่อสารข้อมูลแบบอนุกรม	44
23. แสดงโครงสร้างของแพ็ทเค็ท	46
24. แสดงโครงสร้างของการสื่อสารข้อมูลแบบอนุกรมมาตรฐาน RS232	48
25. การติดต่อบแบบจุดต่อจุด	49

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญภาพ (ต่อ)

ภาพที่	หน้า
26. แสดงFlow Chart การถ่ายเทข้อมูลจากคอมพิวเตอร์ไปยังเครื่องควบคุมPLC	57
27. แสดงFlow Chart การถ่ายเทข้อมูลจากเครื่องควบคุม PLC ไปยังคอมพิวเตอร์	58
28. แสดงFlow chart การทำงานของการแสดงสถานะ(DISPLAY)	59
29. แสดงFlow chart การทำงานของการจัดเก็บ(Save As)	60
30. แสดงFlow chart การทำงานของการเปิด ไฟล์(Open)	62
31. แสดงรูปแบบการทำงาน	65
32. แสดงการทดลองสื่อสารข้อมูลระหว่างคอมพิวเตอร์กับเครื่องควบคุมPLC	66
33. แสดงการเลือกใช้น้ำยความจำและพอร์ตที่ใช้งาน	67
34. แสดงหน้าต่าง EDIT MEMORY	68
35. แสดงหน้าต่าง Label Properties	69
36. แสดงหน้าต่างการ Save As	70
37. แสดงหน้าต่างการ Open	70
38. โปรแกรมที่ออกแบกับ PLC	71
39. แสดงหน้าต่าง DISPLAY	72
40. การทดลองโปรแกรมที่ออกแบบทางหน้าจคอมพิวเตอร์กับPLC	73
41. แสดงภาพการ RUN PROGRAM ที่ออกแบบระหว่างคอมพิวเตอร์ กับ PLC	75
42. แสดงภาพสถานะการทำงานที่หน้าจคอมพิวเตอร์และที่ PLC	75
43. แสดงสถานะการทำงานทางหน้าต่าง Display	76

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1

บทนำ

สืบเนื่องมาจากทางภาควิชาเทคโนโลยีการวัดคุมทางอุตสาหกรรม ได้มีนโยบายให้นักศึกษาชั้นปีสุดท้ายได้จัดทำโครงการขึ้นเพื่อศึกษาและทดลองโครงการที่จัดทำนั้น โดยโครงการที่จัดทำขึ้นมีเนื้อหาเกี่ยวกับวิชา Programmable Logic Control (PLC) เพื่อศึกษาทฤษฎีนอกเหนือจากที่ได้ศึกษาในหลักสูตรให้สามารถประยุกต์ใช้งานร่วมกับเทคโนโลยีในโลกปัจจุบัน

ดังนั้นจึงได้จัดทำโปรแกรมสำเร็จรูปเพื่อรองรับการทำงานร่วมกับ Programmable Logic Control (PLC) โดยจำลองให้คอมพิวเตอร์มีลักษณะการทำงานและหน้าที่คล้ายกับ Programmable Terminal (PT) โดยโปรแกรมดังกล่าวพยายามเน้นเนื้อหาที่สอดคล้องกับการส่งผ่านข้อมูลระหว่าง PLC กับคอมพิวเตอร์ และการควบคุมกระบวนการทางอุตสาหกรรม

วัตถุประสงค์ของโครงการ

1. เพื่อศึกษา PROGRAMMABLE LOGIC CONTROL (PLC) ทั้งในทางทฤษฎีและการประยุกต์ใช้งาน
2. เพื่อศึกษาลักษณะการใช้งาน โปรแกรม Visual Basic
3. เพื่อศึกษาอุปกรณ์ต่างๆ ที่ใช้เชื่อมต่อกับ PLC เช่น computer , programmable terminal
4. เพื่อให้ได้โปรแกรมที่สามารถออกแบบรูปแบบการควบคุมและแสดงผลของกระบวนการและจัดเก็บเป็นแฟ้มข้อมูลได้
5. เพื่อศึกษาการสื่อสารข้อมูลด้วยมาตรฐาน RS-232C
6. เพื่อศึกษาการสื่อสารที่ใช้ข้อตกลงทางการสื่อสาร (Protocal)

ขอบเขตของโครงการ

1. เป็น โปรแกรมสำเร็จรูปที่สามารถควบคุมการทำงานของ PLC ที่สั่งผ่านทาง computer
2. เป็น โปรแกรมสำเร็จรูปที่สามารถแสดงสถานะการทำงานของ PLC ได้ทางจอ computer
3. เป็น โปรแกรมที่สามารถออกแบบรูปแบบการควบคุมและแสดงผลของกระบวนการ

ผลิตแบบต่างๆตามการประยุกต์ใช้งานของ PLC และจัดเก็บแฟ้มข้อมูล (file) ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. รูปแบบแสดงผล อาทิเช่น การแสดงเป็น Lamp การแสดงเป็นตัวเลข ตลอดจนรูปทรงต่างๆจากกระบวนการผลิตที่จะนำมาแสดงผล

ผลประโยชน์ที่คาดว่าจะได้รับ

1. สามารถที่จะออกแบบรูปแบบการควบคุมกระบวนการผลิตได้
2. สามารถที่จะแสดงผลของกระบวนการผลิตแบบต่างๆได้
3. สามารถนำไปประยุกต์ใช้งานในวงการอุตสาหกรรมต่างๆที่จำเป็นต้องมีการใช้

PLC ในการเป็นส่วนหนึ่งของกระบวนการ



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2

ทฤษฎี

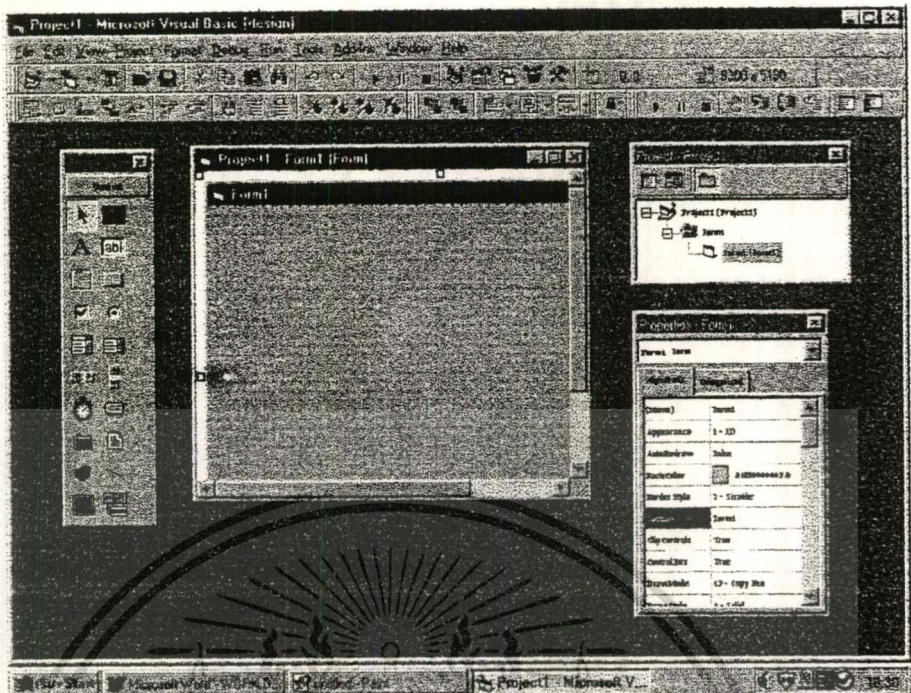
ทฤษฎีการทำงาน Visual Basic

วิธีการใช้งานเอนไวรอนเมนต์ของ VB/WIN

เนื่องจาก VB/Win เป็นเครื่องมือสำหรับพัฒนาแอปพลิเคชันที่ทำงานภายใต้เอนไวรอนเมนต์ของวินโดวส์ ดังนั้นเอนไวรอนเมนต์ของ VB/Win จึงประกอบด้วยคำสั่งหรือหน้าต่างมากมาย ซึ่งแต่ละหน้าต่างก็มีหน้าที่หรือการทำงานที่แตกต่างกัน ดังนั้นก่อนที่จะเริ่มต้นการออกแบบและเขียนโค้ดสำหรับสร้างแอปพลิเคชันหนึ่ง ๆ ขึ้นมา ผู้เขียนก็ขออธิบายถึงความหมายและวิธีการใช้ส่วนต่างๆ ที่เป็นเอนไวรอนเมนต์ของ VB/Win เสียก่อน

รู้จักกับเอนไวรอนเมนต์ของ VB/Win

ผู้อ่านสามารถเริ่มต้นรัน VB/Win ได้โดยการดับเบิลคลิกที่ไอคอน VB/Win ก็จะปรากฏหน้าต่างเอนไวรอนเมนต์ทั้งหมดของ VB/Win ดังในภาพที่ 1 และ 2 แต่โดยส่วนใหญ่เมื่อโหลด VB/Win ก็มักจะแสดงผลเฉพาะส่วนหลักของ VB/Win (Micro soft visual Basic) หน้าต่าง โปรเจกต์ (Project windows) และฟอร์มปรกติ (Form) เท่านั้น ซึ่งความหมายและหน้าที่ของส่วนต่างๆ มีดังนี้



ภาพที่ แสดงรูปแบบของ Visual Basic

หน้าต่างโปรเจกต์ (Project Windows)

เป็นหน้าต่างที่รวบรวมรายชื่อของฟอร์ม โมดูลไฟล์ (custom control) โมดูลคลาส (class module) หรือไฟล์ทรัพยากร (resource file) สำหรับการสร้างแอปพลิเคชันหนึ่ง ๆ ซึ่งการรวมเอาไฟล์เหล่านี้เข้าด้วยกันเพื่อสร้างแอปพลิเคชันภายใต้ VB/Win เราเรียกว่า โปรเจกต์ (Project) หรือโครงการ ซึ่งโดยปรกติ VB/Win จะจัดเก็บไฟล์โปรเจกต์โดยมีนามสกุล .MAK โดยจะมีการจัดเก็บแยกออกจากไฟล์อื่นๆ โดยเด็ดขาด

ฟอร์ม (Form)

ในการแสดงหน้าต่างเพื่อที่จะสื่อการทำงานกับผู้ใช้ ทั้งนี้เนื่องจาก VB/Win ได้รับการออกแบบให้โปรแกรมเมอร์สามารถนำมาใช้ออกแบบแอปพลิเคชันในลักษณะของการสื่อด้วยรูป (Visual) ดังนั้นการสื่อการทำงานต่างๆ ระหว่างแอปพลิเคชันกับผู้ใช้จะต้องกระทำผ่านฟอร์ม โดยในโปรเจกต์หนึ่ง ๆ สามารถมีได้หลายฟอร์ม และภายในฟอร์มต่างๆ ก็จะถูกใช้ในการบรรจุคอนโทรลต่างๆ ดังนั้นฟอร์มจึงทำหน้าที่เป็นตัวบรรจุ (Container) และฟอร์มก็ยังคงเป็นออปเจกต์ตัวหนึ่งของ VB/Win ที่อนุญาตให้ผู้อื่นแก้ไขคุณสมบัติได้ในขณะออกแบบจากหน้าต่างคุณสมบัติ

และสามารถควบคุมพฤติกรรมต่างๆ ของฟอร์มได้โดยผ่านทางวิธีเช่นเดียวกับออปเจกต์อื่นๆ

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หน้าต่างคุณสมบัติ (Property Windows)

เป็นหน้าต่างที่รวบรวมคุณสมบัติทั้งหมดของฟอร์มหรือคอนโทรลเอาไว้ ซึ่งคุณสมบัติทั้งหมดที่ปรากฏในหน้าต่างนี้จะเป็นคุณสมบัติที่ผู้อ่านสามารถกำหนดค่าได้ในขณะกำลังออกแบบ เมื่อผู้อ่านทำการแก้ไขค่าคุณสมบัติต่างๆ ในหน้าต่างคุณสมบัตินี้ก็จะส่งผลต่อคอนโทรลตัวนั้นทันที ซึ่งบางคุณสมบัติสามารถแสดงให้เห็นการเปลี่ยนแปลงคุณลักษณะได้ทันที ส่วนคุณสมบัติบางอย่างจะแสดงผลให้เห็นก็ต่อเมื่อผู้อ่านมีการรันแอปพลิเคชันเท่านั้น

หูลบ็อกซ์ (Toolbox Windows)

เป็นหน้าต่างที่รวบรวมเอาเครื่องมือต่างๆ สำหรับใช้ในการออกแบบฟอร์ม ซึ่งการรวมเอาเครื่องมือ (คอนโทรล) ตัวใดให้กับฟอร์ม ผู้อ่านสามารถทำได้โดยการคลิกที่คอนโทรลนั้น เพื่อเลือกแล้วนำมาวางลงในฟอร์มโดยวิธีการลากแล้ววาง (drag and drop) หรือโดยวิธีการดับเบิลคลิกที่ไอคอนคอนโทรล ในหูลบ็อกซ์ก็ได้

หูลบาร์ (Toolbar)

เป็นส่วนที่รวบรวมเอาไอคอนเล็กๆ มากมายเอาไว้ ซึ่งแต่ละไอคอนเหล่านี้ก็เปรียบเสมือนคำสั่งหนึ่งๆ ของเมนู ดังนั้นไอคอนในส่วนนี้จึงถูกออกแบบมาเพื่อให้การเลือกใช้คำสั่งของเมนูรวดเร็วและมีลักษณะที่สื่อความหมายกับผู้ใช้มากขึ้น

คอนโทรลของ VB/Win (custom control)

คอนโทรลเป็นเครื่องมืออย่างหนึ่งที่จะช่วยให้การสร้างแอปพลิเคชันด้วย VB/Win สามารถกระทำได้ง่ายและรวดเร็วยิ่งขึ้น ซึ่งในความเป็นจริงคอนโทรลก็คือไฟล์ไดนามิกลิงค์ไลบรารีทั่วไปของวินโดวส์ ไฟล์เหล่านี้จะมีนามสกุล .VBX หรือ .OCX ซึ่งเราเรียกว่า custom control ซึ่งไฟล์เหล่านี้จะถูกจัดเก็บในไดเรกทอรีย่อย system ของวินโดวส์

ผู้อ่านสามารถโหลดไฟล์คอนโทรลมารวมกับโปรเจกต์ได้ด้วยการใช้คำสั่ง add file ในเมนู file แต่ถ้าหากโปรเจกต์ไม่มีการใช้คอนโทรลใดๆ ก็ให้ผู้อ่านลบคอนโทรลตัวนั้น ๆ ออกจากโปรเจกต์ ซึ่งสามารถทำได้ด้วยคำสั่ง remove file ในเมนู file เมื่อผู้อ่านสร้างแอปพลิเคชันและแจกจ่ายให้กับผู้ใช้อย่างอื่น ๆ ผู้อ่านก็ต้องคัดลอกไฟล์ custom control ที่ใช้ใน แอปพลิเคชันลงในไดเรกทอรีย่อย system ของวินโดวส์ของเครื่องผู้ใช้ด้วย

แบบฟอร์ม (FORM)

แบบฟอร์มเปรียบเสมือนกับเป็นผ้าใบที่จะใช้สร้างแอปพลิเคชันลงไป แบบฟอร์มแต่ละแบบจะตรงกับวินโดวส์ที่อยู่ในแอปพลิเคชันขณะดำเนินงาน

คุณสมบัติ (Property)

คุณสมบัติหลายๆ อย่าง สามารถมีผลต่อลักษณะปรากฏและพฤติกรรมของแบบฟอร์มในเวลาที่เราแสดงแบบฟอร์มนั้นออกมา แต่ในที่นี้แสดงเฉพาะคุณสมบัติของแบบฟอร์มที่นิยมใช้บ่อยที่สุดการตั้งคุณสมบัติอาจทำได้โดยการใช้วินโดวส์ Properties ในขณะที่กำลังออกแบบแอปพลิเคชันหรืออาจทำได้โดยการเขียนรหัสโปรแกรมที่ตั้งคุณสมบัติในขณะที่โปรแกรมดำเนินงาน

AutoRedraw

คุณสมบัติ autoredraw ทำหน้าที่ควบคุมดูแลว่าจะสร้างภาพลักษณะจอภาพ (screen image) ขึ้นอีกครั้งอย่างไร เมื่อสวิตช์กลับไปยังแบบฟอร์มของ Visual basic หลังจากทำงานในวินโดวส์อื่นๆ ที่ปกคลุมแบบฟอร์มไว้ Visual Basic ก็จะวาดกราฟฟิกใดๆ บนแบบฟอร์มนั้นขึ้นใหม่โดยอัตโนมัติถ้าคุณตั้งคุณสมบัติ autoredraw เป็น true แต่ถ้าตั้งคุณสมบัตินี้เป็น False Visual Basic ก็จะต้องเรียกกระบวนการงานอื่นมาเพื่อทำงานชิ้นนี้แทนที่

BackColor

คุณสมบัตินี้ใช้กำหนดสีของภาวส่วนหลัง (background) เมื่อเลือก BackColor จากรายการคุณสมบัติที่อยู่ในวินโดวส์ Properties จะพบค่ากำหนดโดยปริยายเป็น &H80000005& ซึ่งเป็นค่าคงตัวฐานสิบหกที่นิยามสีของ Visual Basic (ค่าคงตัวเหล่านี้จะถูกบรรจุอยู่ในไฟล์ (constant.txt) จะให้รายละเอียดเกี่ยวกับค่ากำหนดสีนี้โดยละเอียดมากขึ้น

BorderStyle

เราสามารถตั้งคุณสมบัตินี้ให้เป็นค่ากำหนดที่นิยามไว้แล้วอย่างใดอย่างหนึ่งในการตั้งคุณสมบัติ BorderStyle ทำได้เฉพาะในเวลาออกแบบเท่านั้น ค่ากำหนดไม่มีผลต่อการแสดงผลของแบบฟอร์มในเวลาออกแบบ แต่จะเปลี่ยนแปลงการแสดงผลของวินโดวส์ในเวลาดำเนินงานดังนี้

0-None

วินโดวส์ไม่มีขอบเขต

1-Fixed Single วินโดวส์มีขนาดตายตัว (ระบุในเวลาออกแบบ) พร้อมกับมีขอบเขตเส้น

2-Sizable วินโดวส์เปลี่ยนแปลงขนาดได้พร้อมกับมีขอบแบบเส้นคู่มาตรฐาน

3-Fixed Double วินโดวส์มีขนาดตายตัว(ระบุในเวลาออกแบบ)และมีขอบเขตแบบเส้นคู่

Caption

คุณสมบัตินี้กำหนดข้อความที่แสดงในไตเติลบาร์ของแบบฟอร์มหรือข้อความที่แสดงในหรือถัดจากตัวควบคุมตัวใดตัวหนึ่ง

ControlBox

คุณสมบัติ Boolean คุณสมบัตินี้ควรจะต้องให้ True ถ้าต้องการให้แสดงบ็อกซ์สำหรับเมนู (Control) ออกมาตรงมุมด้านบนซ้ายของวินโดวส์ เราสามารถดัดแปลงคุณสมบัตินี้เฉพาะในเวลาออกแบบเท่านั้น การตั้งคุณสมบัติ ControlBox ให้เป็น true จะไม่มีผลใดๆ ถ้าตั้ง BorderStyle ให้เป็น 0-None

Enabled

ออบเจกต์ทุก ๆ ตัวของ Visual Basic มีคุณสมบัติ Enabled ที่สามารถตั้งให้เป็น True หรือ False ซึ่งเป็นการทำออบเจกต์ให้อยู่ในสภาพใช้งานได้ หรืออยู่ในสภาพใช้งานไม่ได้ ในกรณีของแบบฟอร์มหนึ่งๆ ตามปกติตั้งคุณสมบัตินี้ให้เป็น true แต่อาจตั้งคุณสมบัตินี้ให้เป็น False เพื่อป้องกันไม่ให้มีการส่งอีเวนต์ของเมาส์หรือคีย์บอร์ดไปยังแบบฟอร์ม

ForeColor

คุณสมบัตินี้ नियามสีของข้อความหรือกราฟฟิกส่วนหน้า (foreground) การตั้งคุณสมบัตินี้ทำได้โดยการเลือกสีจากชุดอุปกรณ์สีด้วยแนวทางเดียวกันกับที่ใช้ตั้งคุณสมบัติ BackColor ข้อความสั่ง Print ทั้งหมดจะเขียนข้อความ โดยการใช้สีส่วนหน้า

Height, Width

คุณสมบัติเหล่านี้กำหนดความสูงและความกว้างของแบบฟอร์มที่วัดอยู่ในหน่วยที่เรียกว่า ทวิพ หรือเศษหนึ่งส่วนยี่สิบของพอยต์ (ทวิพเทียบเท่ากับ 1/1440 นิ้ว)

Icon

เราสามารถตั้งคุณสมบัตินี้ให้เป็นชื่อของไฟล์ที่บรรจุไอคอน ไอคอนนี้จะไปแทนแบบฟอร์มเมื่อลดขนาดแบบฟอร์ม ในการตั้งคุณสมบัตินี้ในเวลาออกแบบ ให้เลือก Icon จากรายการคุณสมบัติในวินโดวส์ Properties และหลังจากนั้นกดปุ่มเมาส์บนปุ่มไขว้ปลาที่ปรากฏอยู่ด้านข้างของเซตคิงบ็อกซ์ ในไดอะล็อกบ็อกซ์ที่เกิดขึ้น เลือกไฟล์ไอคอนที่ต้องการคุณสมบัติในเวลาดำเนินงาน จะต้องใช้ฟังก์ชัน LoadPicture หรือ กำหนดค่าของคุณสมบัติ Icon ของแบบฟอร์มอื่นๆ

MaxButton, MinButton

เราควรจะต้องตั้งคุณสมบัติ Boolean เหล่านี้เป็น true ถ้าต้องการให้แสดงปุ่มขยายขนาดและปุ่มลดขนาด ที่ตรงมุมด้านบนขวาของวินโดวส์ คุณสมบัติทั้งสองนี้จะถูกมองข้ามถ้าได้ตั้งคุณสมบัติ BorderStyle ให้เป็น 0-None ควรทราบไว้ด้วยว่าปุ่มขยายขนาดและปุ่มลดขนาดไม่ได้หายไปจากแบบฟอร์มในเวลาออกแบบ แม้แต่เมื่อคุณตั้งคุณสมบัติ MaxButton และ MinButton ให้เป็น False ผลของค่ากำหนดเหล่านี้จะไม่ปรากฏในวินโดวส์จนกว่าถึงเวลาดำเนินงาน

Name

คุณสมบัตินี้ยามชื่อของฟอร์มในรหัสโปรแกรม ชื่อนี้ไม่สามารถเปลี่ยนแปลงในเวลาดำเนินงาน

Picture

เราสามารถตั้งคุณสมบัตินี้เพื่อแสดงว่าจะแสดงภาพแบบแผนที่ปิด (bit map) ในวินโดวส์เสมอ ในการตั้งคุณสมบัตินี้ในเวลาออกแบบ ให้เลือกคุณสมบัตินี้จากรายการคุณสมบัติ และกดปุ่มเมาส์บนปุ่มไขว้ปลาที่ปรากฏอยู่ด้านข้างของเซตคิงบ็อกซ์ในวินโดวส์ Properties ในไดอะล็อกบ็อกซ์ที่เกิดขึ้น ให้เลือกไฟล์ที่บรรจุภาพที่ต้องการแสดงในการเปลี่ยนแปลงคุณสมบัตินี้ในเวลาดำเนินงานต้องใช้ฟังก์ชัน LoadPicture

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Top, Left

คุณสมบัติเหล่านี้ควบคุมตำแหน่งของแบบฟอร์มบนจอภาพ โดยการนิยามตำแหน่งส่วนบนสุดของวินโดวส์ และพิกัดด้านซ้ายสุดที่วัดในหน่วยทวิพอยัง โดยปริยาย

Visible

ค่ากำหนดของคุณสมบัติ Boolean นี้แสดงถึงว่าจะให้ฟอร์มวินโดวส์เห็นได้หรือไม่เห็น การตั้งคุณสมบัตินี้ไปเป็น false จะซ่อนไม่ให้เห็นแบบฟอร์ม

WindowState

คุณสมบัตินี้กำหนดค่าให้แสดงวินโดวส์ออกมาในสถานะปกติ (ค่ากำหนดเป็น 0) ในสถานะลดขนาดเป็นไอคอน (ค่ากำหนดเป็น 1) หรือในสถานะขยายขนาด (ค่ากำหนดเป็น 2)

อีเวนต์ (EVEN)

Click, DbClick และ Load เป็นอีเวนต์ที่พบบ่อยที่สุดสำหรับแบบฟอร์มที่จะประมวล (อีเวนต์ KeyPress และการะบวนการลากแล้วปล่อย (Drag and drop))

Click

เมื่อผู้ใช้กดปุ่มเมาส์บนแบบฟอร์ม Visual Basic เรียกกระบวนการ Form_Click ถ้าผู้ใช้กดปุ่มเมาส์บนออบเจกต์ที่บรรจุอยู่ในแบบฟอร์ม อีเวนต์ Click ก็จะไม่ถูกส่งผ่านไปให้แก่แบบฟอร์มแม้แต่เมื่อออบเจกต์ตัวอื่น ๆ ไม่ได้ประมวลอีเวนต์นั้น

DblClick

อีเวนต์นี้จะได้รับถ้าผู้ใช้กดปุ่มเมาส์สองครั้งต่อเนื่องกันอย่างรวดเร็วที่จุดบางจุดบนแบบฟอร์ม โดยแท้จริงแล้วกระบวนการนี้สับเปลี่ยนอีเวนต์สองอีเวนต์ โดยอีเวนต์ Click สำหรับการกดปุ่มเมาส์ครั้งแรก และอีเวนต์ DbClick สำหรับการกดปุ่มเมาส์ครั้งที่สอง

Load

เหตุการณ์นี้เกิดขึ้นอย่างอัตโนมัติ เมื่อใดก็ตามที่บรรจุแบบฟอร์ม เช่น เมื่อเริ่มใช้งานแอปพลิเคชัน กระบวนการอินเวนต์ Load เป็นกระบวนการที่ใช้สะดวกสำหรับการกำหนดค่าเริ่มต้นของคุณสมบัติและตัวแปรในเวลาเริ่มใช้งานโปรแกรม

กระบวนการและกลวิธี (Method)

กลวิธีของแบบฟอร์มที่น่าสนใจมากที่สุดคือ กลวิธีกราฟิก เนื่องจากกลวิธีนี้ค่อนข้างซับซ้อน ดังนั้นจึงได้ใช้บททั้งบทความเกี่ยวกับกราฟิกในหนังสือเล่มนี้ สำหรับในตอนนี้อะจะพิจารณาวิธีเพียงไม่กี่อย่าง เพื่อให้สามารถเริ่มใช้งานได้

Cls

กลวิธีนี้ลบ قالبของกราฟิกและข้อความทั้งหมด วากยสัมพันธ์ อย่างสมบูรณ์ สำหรับกลวิธี Cls เป็นดังต่อไปนี้

```
[formName].Cls
```

LoadPicture

เราสามารถให้ฟังก์ชันนี้เพื่อตั้งคุณสมบัติ Picture ของแบบฟอร์ม วากยสัมพันธ์ของฟังก์ชันเป็นดังนี้

```
LoadPicture ([fileName])
```

ชื่อ ไฟล์จะต้องเป็นนิพจน์สตริงที่ระบุไฟล์ที่ประกอบไปด้วย ไอคอน บิตแมปของ Windows หรือไฟล์เมตาของ Windows (ชื่อไฟล์เหล่านี้ตามปกติมีส่วนขยายเป็น ico,bmp และ wmf) ถ้าไม่ได้กำหนดชื่อไฟล์ไว้ LoadPicture จะส่งคืนภาพว่าง ๆ

Print

กลวิธีนี้เขียนข้อความไปยังแบบฟอร์ม ใช้วากยสัมพันธ์ดังนี้

```
[formName]. Print[[expression][ (;,)]]...
```

แต่ละนิพจน์จะตามหลังด้วยอัมภาค (;) หรือจุดภาค (.) ถ้าใช้อัมภาคจะทำให้ผลส่งออกของนิพจน์อยู่หลังอักขระตัวสุดท้ายที่เขียนในทันที แต่ถ้าใช้จุดภาค ตำแหน่งผลส่งออกบน

เอกสารนี้เป็นเอกสารสงวนไว้สำหรับการใช้ร่วมเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า แบบฟอร์มจะถูกตั้งระยะ ให้อยู่ที่คอลัมน์ถัดไป (คอลัมน์จะนิยามทุกๆ อักขระ 14 ตัว) ถ้านิพจน์สุดท้ายไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ท้ายในข้อความสั่ง Print ลงท้ายด้วยอัฒภาคหรือจุลภาคจะไม่มี การเขียนอักขระปิดแควไปที่แบบฟอร์ม มิฉะนั้นแล้วจะตั้งตำแหน่งผลส่งออกให้อยู่บรรทัดถัดไป ต่อไปนี้เป็นตัวอย่างของการใช้กลวิธี Print ในวินโดวส์ Debug

บ็อกซ์ภาพและภาพลักษณะ(Image and Picture)

ตัวควบคุมบ็อกซ์ภาพและภาพลักษณะทำให้คุณสามารถวางสารสนเทศทางกราฟิกไว้ในตำแหน่งเฉพาะบนแบบฟอร์ม เครื่องมือสร้างบ็อกซ์ภาพและเครื่องมือสร้างภาพลักษณะจากทูลบาร์ได้แสดงเอาไว้ในรูปที่ 2 และ 3 บ็อกซ์ภาพจะมีความยืดหยุ่นมากกว่า และด้วยเหตุนี้จึงต้องการหน่วยความจำและการประมวลผลในประมาณที่มากกว่า บ็อกซ์ภาพเหมาะอย่างยิ่งสำหรับสภาพแวดล้อมไดนามิก เช่น เมื่อคุณกำลังวาดกราฟฟิกเข้าไปที่จอภาพโดยตรงในขณะที่โปรแกรมกำลังดำเนินงานหรือการทำไอคอนให้เคลื่อนไหวได้ ด้วยการโยกย้ายไอคอนไปรอบๆ บนจอภาพ ออบเจ็กต์ภาพลักษณะดีที่สุดสำหรับสถานการณ์สถิต นั่นคือเมื่อไม่มีการคัดแปลงบิตแมปหรือไอคอนที่คุณสร้างและวางเอาไว้บนจอภาพ

ภาพที่ 2 เครื่องมือสร้างบ็อกซ์ภาพ(PictureBox)



ภาพที่ 3 เครื่องมือสร้างภาพลักษณะ(Image)

คุณสมบัติ

ตัวควบคุมบ็อกซ์ภาพและภาพลักษณะมีคุณสมบัติ Enabled, Height, Left, Name, Picture, Top, Visible และ Width คุณสมบัติเหล่านี้ส่วนใหญ่แล้วจะประพฤติปฏิบัติเหมือนกันกับที่ได้บรรยายไว้ในตอนก่อนหน้านี้นบนแบบฟอร์ม ยกเว้นแต่ว่าจะวัดพิกัดสำหรับคุณสมบัติ Height, Left, Top และ Width โดยเทียบกับตำแหน่งของออบเจ็กต์ที่อยู่ภายในแบบฟอร์ม ไม่ได้วัดในรูปของพิกัดไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารที่วางตัวไปใช้

จอภาพสัมผัส (ตัวควบคุมทั้งหมดที่สร้างขึ้นจากทูลบาร์จะมีลักษณะเช่นนี้) คุณสามารถตั้งคุณสมบัติ Picture ให้แสดงบิตแม็พหรือไอคอน โดยในเวลาออกแบบให้ใช้วินโดวส์ Properties และไดอะล็อกบ็อกซ์ Load Picture และในเวลาดำเนินงานคุณจำเป็นต้องใช้ฟังก์ชัน Load Picture บ็อกซ์ภาพ ซึ่งประพติปฏิบัติเกือบเหมือนกันกับแบบฟอร์มเล็กๆ ที่อยู่ภายในแบบฟอร์มหลักมีคุณสมบัติต่อไปนี้ที่ไม่พบอยู่ในตัวควบคุมภาพลักษณะ นั่นคือ AutoRedraw, FontBold, FontItalic, FontName, FontSize, FontUnderline คุณสมบัติเหล่านี้ประพติปฏิบัติในทำนองเดียวกันที่มีไว้สำหรับแบบฟอร์ม ตัวควบคุมภาพลักษณะครอบครองคุณสมบัติอย่างหนึ่งที่ไม่มียู่ในบ็อกซ์ภาพ นั่นคือคุณสมบัติ Stretch โดยปริยายแล้วจะตั้ง Stretch ให้เป็น false ซึ่งทำให้ตัวควบคุมภาพลักษณะเปลี่ยนขนาดตัวมันเองเพื่อนให้พอเหมาะกับภาพที่ตัวควบคุมนั้นบรรจุอยู่ ถ้าคุณตั้ง Stretch ให้เป็น True ภาพจะเปลี่ยนขนาดให้พอเหมาะกับตัวควบคุมตัวควบคุมบ็อกซ์ภาพและภาพลักษณะยังมีลักษณะอีกประการหนึ่งที่แตกต่างกัน จอแสดงผลของแบบฟอร์มนั้นจะประกอบกันขึ้นมาจากชั้นสามชั้นสารสนเทศทั้งหมดที่แสดงบนแบบฟอร์มโดยตรง จะปรากฏอยู่บนชั้นล่างสุด สารสนเทศจากตัวควบคุมกราฟฟิคสามตัว (ตัวควบคุมภาพลักษณะรูปทรง และเส้น) จะปรากฏอยู่ตรงชั้นกลาง และออบเจ็กต์อื่นๆ ทั้งหมดจะอยู่บนชั้นบนสุดด้วยเหตุนี้ภาพลักษณะจึงอาจบังข้อความที่พิมพ์ออกมาบนแบบฟอร์ม ซึ่งโดยค่อเนื่องกันบ็อกซ์ภาพก็อาจบังภาพลักษณะได้เช่นกัน

อีเวนต์

ทั้งตัวควบคุมบ็อกซ์ภาพและภาพลักษณะสามารถรับอีเวนต์ Click และ DblClick ซึ่งเป็นอีเวนต์ที่ง่ายที่สุดและพบบ่อยที่สุด

DragDrop เกิดขึ้นเมื่อผู้ใช้มีการลากแล้ววาง

Mouse Down เกิดขึ้นเมื่อผู้ใช้มีการกด (Press) ปุ่มของเมาส์

Mouse Up เกิดขึ้นเมื่อผู้ใช้ปล่อยปุ่มของเมาส์

Mouse Move จะเกิดขึ้นเมื่อผู้ใช้มีการเลื่อนพอยน์เตอร์ของเมาส์

กระบวนการและกลวิธี

คุณสามารถเรียกฟังก์ชัน LoadPicture เพื่อใช้ตั้งคุณสมบัติ Picture ของตัวควบคุมบ็อกซ์ภาพหรือภาพลักษณะ บ็อกซ์ภาพยังสนับสนุนกลวิธี Cls และ Print รวมทั้งกลวิธีกราฟฟิค

เลเบล (Label)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เลเบลประกอบไปด้วยพื้นที่ซึ่งคุณสามารถนำเสนอสารสนเทศประเภทข้อความที่ผู้ใช้ไม่สามารถจะแก้ไขได้ การตั้งเนื้อหาของเลเบลทำโดยการคัดแปลงคุณสมบัติ Caption ของเลเบล คุณไม่สามารถพิมพ์หรือวาดเข้าไปในเลเบล คุณสร้างเลเบลโดยอาศัยเครื่องมือสร้างเลเบลจากทูลบาร์ของ Visual Basic เช่นที่แสดงในรูปที่ 4



ภาพที่ 4 เครื่องมือสร้างเลเบล(Label)

คุณสมบัติ

เลเบลมีคุณสมบัติ FontBold, FontItalic, FontName, FontSize, FontUnderline, Height, Left, Name, Top, Visible และ Width ซึ่งดำเนินการด้วยแนวทางที่เป็นมาตรฐาน คุณสมบัติเพิ่มเติมบางอย่างได้บรรยายไว้ในที่นี้แล้ว

Alignment

คุณสมบัตินี้กำหนดว่าจะวางข้อความหน้าของเลเบลอย่างไร โดยปริยายแล้วจะตั้งคุณสมบัติ Alignment ให้เป็น 0 ซึ่งกำหนดให้ข้อความหน้าอยู่ชิดซ้าย คุณยังสามารถตั้ง Alignment ให้เป็น 1-Right Justify หรือเป็น 2-Center

AutoSize

ถ้าตั้งคุณสมบัตินี้ให้เป็น True ก็จะเปลี่ยนขนาดเลเบลโดยอัตโนมัติเพื่อให้เหมาะกับข้อความที่กำหนดไว้ด้วยคุณสมบัติ Caption ถ้าตั้ง AutoSize ให้เป็น False เลเบลจะยังคงมีขนาดตามที่นิยามในเวลาออกแบบ ในกรณีนั้นถ้าข้อความหน้ายาวเกินไปข้อความนั้นจะถูกคัดออกเพื่อให้พอเหมาะกับเลเบล

BorderStyle

คุณสมบัตินี้สามารถตั้งให้เป็น 0 (ค่าปริยาย) สำหรับกรณีไม่มีขอบ หรือเป็น 1 สำหรับขอบแบบเส้นเดียว

Caption

คุณสมบัติ Caption ประกอบไปด้วยข้อความที่จะแสดงออกมาในฟิลด์ของเลเบลหรือพื้นที่ของจอภาพ

Enabled

ตามปกติจะตั้งคุณสมบัตินี้ให้เป็น True แต่คุณอาจตั้งให้เป็น False เพื่อทำข้อความให้เป็นสีเทา และทำให้ไอเวนค์เมาส์ต่างๆ ใช้งานไม่ได้

ไอเวนค์

เลเบลออบเจกต์ไอเวนค์ Click และ DblClick ในทำนองเดียวกันกับที่ตัวควบคุมบ็อกซ์ภาพ ภาพลักษณะ และแบบฟอร์มสามารถทำได้ กระบวนการและกลวิธี

เลเบลออบเจกต์มีไว้เพื่อใช้แสดงบ็อกซ์ของข้อความขนาดเล็ก ด้วยเหตุที่สามารถนิยามข้อความ โดยเพียงแต่ตั้งคุณสมบัติ Caption ดังนั้นจึงไม่มีกลวิธีที่เกี่ยวข้องกับเลเบลที่จะมีประโยชน์หรือจำเป็นต่อผู้เริ่มต้น

เท็กซ์บ็อกซ์ (Text Box)

คุณได้ทดลองใช้งานเท็กซ์บ็อกซ์มาแล้ว โดยการใช้เครื่องมือสร้างเท็กซ์บ็อกซ์เช่นที่แสดงไว้ในภาพที่ 5 คุณเพียงวางเท็กซ์บ็อกซ์ไว้บนแบบฟอร์ม ก็จะทำให้สามารถสร้างบริเวณของจอภาพที่ผู้ใช้จะสามารถป้อนข้อความเข้าไปได้



ภาพที่ 5 เครื่องมือสร้างเท็กซ์บ็อกซ์(TextBox)

คุณสมบัติ

เท็กซ์บ็อกซ์มีคุณสมบัติมาตรฐานคือ BorderStyle, Enabled, FontBold, FontItalic, FontName, FontSize, FontUnderline, Height, Left, Name, Top, Visible และ Width บวกกับคุณสมบัติเพิ่มเติมต่อไปนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

MaxLength

คุณสมบัตินี้มักจะตั้งให้เป็น 0 (โดยปริยาย) ซึ่งแสดงว่าเท็กซ์บ็อกซ์จะรับอักขระได้มากเท่าที่ผู้ใช้พิมพ์เข้าไป อย่างไรก็ตาม ถ้าคุณตั้ง MaxLength ให้เป็นค่าที่ไม่ใช่ ศูนย์ Visual Basic ก็จะมีจำกัดสิ่งที่ผู้ใช้ป้อนเข้าไปให้มีจำนวนของอักขระตามที่ระบุไว้

MultiLine

บ่อยครั้งที่คุณจะต้องคุณสมบัติให้เป็น False ซึ่งจำกัดสิ่งที่ผู้ใช้ป้อนเข้าไปให้เป็นข้อความบรรทัดเดียว เมื่อตั้งคุณสมบัติ MultiLine ให้เป็น True ผู้ใช้สามารถกดแป้น Enter เพื่อวางการปิดแคร่ไว้ในข้อความ และป้อนข้อความต่อไปบนบรรทัดถัดไป

PasswordChar

คุณสมบัตินี้ควบคุมว่าเท็กซ์บ็อกซ์เป็นฟิลเตอร์รหัสผ่านหรือไม่ เมื่อผู้ใช้พิมพ์ข้อความเข้าไปในฟิลเตอร์รหัสผ่าน อักขระที่ปรากฏขึ้นบนจอภาพจะแตกต่างไปจากอักขระที่ได้พิมพ์เข้าไป โดยปริยายแล้วจะตั้งคุณสมบัตินี้ให้เป็นสตริงว่าง ซึ่งหมายถึงว่าผู้ใช้จะเห็นอักขระที่พวกเขากำลังป้อนเข้าไป ถ้าตั้งคุณสมบัติ PasswordChar ให้เป็นอักขระเช่นดอกจัน ก็จะแสดงอักขระนั้นออกมาสำหรับอักขระทุก ๆ ตัวที่ผู้ใช้พิมพ์เข้าไปถึงตอนนี้แล้วอย่างเพ็งลึบสน เนื้อหาที่แท้จริงของเท็กซ์บ็อกซ์นั้นจะถูกตั้งให้เป็นข้อความที่ผู้ใช้พิมพ์เข้าไป คุณสมบัตินี้มีผลต่อการแสดงผลเท่านั้น

ScrollBars

คุณสามารถตั้งคุณสมบัตินี้ให้เป็น 0 (ไม่มีสกรอลล์บาร์ในเท็กซ์บ็อกซ์)

1. (มีเฉพาะสกรอลล์บาร์ตามแนวนอน)
2. (มีเฉพาะสกรอลล์บาร์ตามแนวตั้ง) หรือ
3. (มีสกรอลล์บาร์ทั้งแนวนอนและแนวตั้ง)

SelLength

คุณสมบัตินี้ประกอบไปด้วยจำนวนของอักขระที่เลือกเอาไว้ล่าสุด ค่าของคุณสมบัติจะเปลี่ยนแปลงเมื่อผู้ใช้เลือกข้อความในเท็กซ์บ็อกซ์ คุณยังสามารถจะเปลี่ยนการเลือกผ่านทางโปรแกรมด้วยการตั้งคุณสมบัตินี้ให้เป็นค่าจำนวนเต็มในรหัสของคุณถ้า SelLength เป็น 0 นั้นเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตเห็นไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หมายถึงไม่ได้เลือกอักขระไว้ คุณสามารถ เข้าถึงคุณสมบัตินี้และคุณสมบัติ SelStart และ SelText ได้เฉพาะในเวลาดำเนินงานเท่านั้น

SelStart

การตั้งคุณสมบัติบอกให้คุณทราบว่าทางเลือกในขณะนั้นเริ่มต้นที่ใด การตั้งให้เป็น 0 แสดงว่าทางเลือก (หรือจุดแทรก) เริ่มต้นก่อนอักขระตัวแรก การตั้งเป็น 1 แสดงว่าทางเลือกเริ่มต้นก่อนอักขระตัวที่สอง และเป็นเช่นนี้ต่อเนื่องไป คุณสามารถตัดแปลงคุณสมบัตินี้ผ่านทาง โปรแกรมได้เช่นกัน

SelText

คุณสมบัตินี้ประกอบไปด้วยสตริงข้อความที่ได้เลือกไว้ในขณะนั้น ถ้าไม่ได้เลือกข้อความใดๆ ไว้เลย คุณสมบัติก็จะประกอบไปด้วยสตริงว่าง ถ้าคุณตั้ง SelText ในรหัสโปรแกรมของคุณ จะเป็นการแทนที่ข้อความที่เลือกไว้ในเท็กซ์บ็อกซ์ด้วยการตั้งค่าของ SelText เช่น ถ้าเท็กซ์บ็อกซ์หนึ่งชื่อว่า Text1 ประกอบไปด้วย บรรทัด “ Frankly, my deere, I don't give a damn” และเลือกคำว่า deere ไว้ คุณสามารถแก้ไขประโยคได้โดยการเอ็กซิวต์ข้อความสั่ง Yext1.SelText = “dear” การทำเช่นนี้ยังเปลี่ยนแปลงค่าของ SelLength ด้วยเช่นกัน อย่างไรก็ตามด้วยเหตุที่การเลือกเริ่มต้นที่ตำแหน่งเดียวกัน ดังนั้น SelStart จึงไม่มีผลใดๆ ทั้งสิ้น

Text

โปรแกรมอ่านคุณสมบัตินี้เพื่อควาผู้ใช้ป้อนอะไรเข้าไป โปรแกรมยังสามารถตัดแปลงคุณสมบัติ Text เพื่อเปลี่ยนข้อความที่แสดงออกมา

อีเวนต์

เท็กซ์บ็อกซ์ไม่สนับสนุนอีเวนต์เมาส์ แต่จะสามารถตรวจจับการเคาะแป้นที่ผู้ใช้ป้อนเข้าไป เราจะศึกษาอีเวนต์ของคีย์บอร์ดในภายหลัง

Change

การส่งสัญญาณมาที่อีเวนต์นี้จะเกิดขึ้นเมื่อมีการเปลี่ยนแปลงคุณสมบัติ Text ของเท็กซ์ เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บ็อกซ์ ไม่ว่าจะเป็นผู้ใช้ซึ่งป้อนสารสนเทศใหม่เข้าไป หรือจากโปรแกรมซึ่งสามารถตั้งค่าใหม่ให้แก่คุณสมบัติ Text เช่น ถ้าผู้ใช้พิมพ์คำว่า Hello จะส่งสัญญาณไปยังอีเวนต์ Change ห้าครั้ง โดยแต่ละครั้งสำหรับอักขระแต่ละตัวที่พิมพ์ เข้าไป

LostFocus

การส่งสัญญาณมาที่อีเวนต์นี้เกิดขึ้นเมื่อผู้ใช้ตั้งระยะเคลื่อนย้ายออกไปจากเท็กซ์บ็อกซ์ หรือใช้เมาส์เลือกกรอบเจ็ทซ์อื่นๆ ในแบบฟอร์ม ถ้าคุณต้องการตรวจสอบเนื้อหาของคุณสมบัติ Text การทำเป็นส่วนหนึ่งของกระบวนการงานอีเวนต์ LostFocus จะมีประสิทธิภาพมากกว่าการทำเป็นส่วนหนึ่งของกระบวนการอีเวนต์ Change

กระบวนการและกลวิธี

กลวิธีเกี่ยวข้องกับเท็กซ์บ็อกซ์ที่มีประโยชน์ที่สุด SetFocus ซึ่งจะวางเคอร์เซอร์ไว้ในเท็กซ์บ็อกซ์เฉพาะ เมื่อคุณสร้างแบบฟอร์มที่มีเท็กซ์บ็อกซ์หลายๆ บ็อกซ์ คุณสามารถใช้กลวิธีนี้เพื่อชี้ความสนใจของผู้ใช้ไปที่บ็อกซ์ที่สำคัญที่สุด วากยสัมพันธ์จะเป็นดังนี้

[object.]SetFocus

คอมมานด์บัตตอน (Command Button)

เป็นคอนโทรลที่จะถูกนำไปใช้ในฟอร์มทุกๆ ฟอร์มเนื่องจากเป็นคอนโทรลที่ใช้ในการตอบรับการทำงานจากผู้ใช้ เช่นเดียวกับปุ่มคำสั่ง ok, Cancel ทั่วๆ ไป ดังนั้น จึงถือได้ว่าเป็นคอนโทรลที่พื้นฐานที่สุดก็ได้ เนื่องจากคอนโทรลนี้เป็นปุ่มคำสั่งที่ใช้ในการคลิก เครื่องมือสร้างคอมมานด์บัตตอนได้แสดงเอาไว้ในรูปแบบที่



ภาพที่ 6 เครื่องสร้างคอมมานด์บัตตอน(CommamdButton)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



กำหนด	รายละเอียด
E	Even
M	Mark
N	(default)none
O	Odd
S	Space

D คือ จำนวนของ Databits โดยกำหนดให้เป็น 4,5,6,7,8, (default)

S คือ จำนวนของ Stop bits โดยกำหนดให้เป็น 1(default),1.5

PortOpen Property

กำหนดและแจ้งให้ทราบถึงสถานะของ port ที่ใช้ในการรับส่งข้อมูลว่าให้เปิดหรือปิด

Port

รูปแบบ [form.] MSComm.PortOpen [= {True | False}]

โดยที่ True หมายถึง Portเปิด

โดยที่ False หมายถึง Portปิด

และเมื่อกำหนดให้เป็น False จะทำให้ Port ปิดและจะเตรียมในส่วนของ receive และ transmit buffers โดยที่ Communication Control จะปิด โดยอัตโนมัติ เมื่อมีการติดต่อเรียบร้อยแล้ว

Input Property

แจ้งให้ทราบและนำเอาอักขระที่เป็น String จาก receive buffer

รูปแบบ [form.] MsComm.Input

Output Property

จะเป็นการเขียนอักขระที่เป็น String ไปสู่ transmit buffer

รูปแบบ [form.] MSComm.Output [= outString \$]

InputLen Property

กำหนดและแจ้งให้ทราบถึงจำนวนของอักขระที่ Input Property ทำการอ่านจาก Receive

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

buffer

รูปแบบ [form.] MSComm.InputLen [= numchars %]

ตัวจับเวลา (Timer)

ตัวจับเวลาซึ่งสร้างขึ้นโดยอาศัยเครื่องมือสร้างตัวจับเวลาตามที่แสดงอยู่ในภาพที่ 7 เป็นออบเจกต์ที่สามารถทำให้ไอน์เมนต์ขึ้นด้วยช่วงที่สม่ำเสมอ คุณเขียนโปรแกรมตัวจับเวลาด้วยการตั้งคุณสมบัติ Interval ให้เป็นจำนวนของมิลลิวินาที ที่ควรจะสูญเสียไประหว่างไอน์เมนต์แต่ละไอน์เมนต์ของตัวจับเวลา



ภาพที่ 8 เครื่องมือสร้างตัวจับเวลา (Timer)

ตัวจับเวลาจะถูกจำกัดด้วยความสามารถของฮาร์ดแวร์ ระบบคอมพิวเตอร์ส่วนบุคคลส่วนใหญ่จะสนับสนุนอำนาจจำแนกของตัวจับเวลาไม่เกินกว่า 18 ไอน์เมนต์ต่อวินาที หรือประมาณ 56/1000 วินาทีระหว่างไอน์เมนต์ ในทางปฏิบัติอำนาจจำแนกนี้หมายถึงว่า ค่าที่น้อยกว่า 56 จะไม่เป็นประโยชน์เลย หรือกล่าวอีกนัยหนึ่ง ค่าช่วงเวลาที่ต่างกัน 5,10 หรือแม้แต่ 20 จะสร้างผลลัพธ์ที่เหมือนกัน ข้อจำกัดนี้ไม่ได้หมายถึงว่าคุณควรจะคำนวณอย่างสะเพร่า แต่หมายถึงว่าคุณไม่ควรจะไว้วางใจตัวจับเวลาว่ามีความแม่นยำอย่างไม่น่าเชื่อ

นอกจากนี้ไอน์เมนต์ของตัวจับเวลาบางไอน์เมนต์อาจสูญเสียลำดับการทำงาน สมมติว่าโปรแกรมของคุณได้รับไอน์เมนต์ของตัวจับเวลาวินาทีละหนึ่งครั้ง แต่ส่วนประกอบนั้นของโปรแกรมเกี่ยวข้องกับภาระคำนวณที่ซับซ้อนมากที่ต้องใช้เวลา 10 วินาทีเพื่อดำเนินการ เมื่อการคำนวณนั้นเสร็จสิ้นแล้ว คุณอาจคาดหมายว่าไอน์เมนต์ของตัวจับเวลา 10 ไอน์เมนต์อาจกองซ้อนกันขึ้นไปด้วยวิธีการใดวิธีหนึ่ง แต่ตามความเป็นจริงแล้วมีไอน์เมนต์ของตัวจับเวลาเพียง 1 ไอน์เมนต์ที่จะเกิดขึ้นทันทีหลังการคำนวณ ก่อนที่จะก่อกำเนิดไอน์เมนต์ของตัวจับเวลาขึ้นนั้น ระบบจะคอยตรวจสอบก่อนว่าไอน์เมนต์ของตัวจับเวลาก่อนหน้านี้ยังคงค้างอยู่หรือไม่ ถ้ายังคงค้างอยู่ก็จะไม่ก่อกำเนิดไอน์เมนต์ของตัวจับเวลาขึ้น

การที่มีข้อจำกัดเช่นนี้ อาจทำให้คุณสงสัยถึงการใช้ประโยชน์จากตัวจับเวลา แต่จริงๆ แล้วตัวจับเวลาที่มีประโยชน์มากที่สุดคือตัวจับเวลาที่มีความเที่ยงตรง เช่น อาจใช้ตัวจับเวลาคอยค้น โปรแกรมของคุณด้วยช่วงเวลาที่สม่ำเสมอ

คุณสมบัติ (Property)

คุณสมบัติ Name และ Enabled เป็นคุณสมบัติมาตรฐานของตัวจับเวลา คุณสมบัติที่สำคัญที่สุดของออบเจกต์ตัวจับเวลาคือ คุณสมบัติ Interval คุณตั้งคุณสมบัตินี้ให้เป็นจำนวนของมิลลิวินาทีที่อยู่ระหว่างอีเวนต์ของตัวจับเวลา พร้อมกับมีค่าอยู่ในช่วงตั้งแต่ 0 ตลอดจนถึง 65,535 ซึ่งในทางทฤษฎีจะก่อกำเนิดอีเวนต์ด้วยอัตราที่อยู่ในช่วงตั้งแต่วินาทีละ 1000 อีเวนต์ไปจนถึงน้อยกว่า 1 อีเวนต์ทุกๆ นานาที ถ้าตั้งช่วงเวลาเป็น 0 จะเทียบเท่ากับการทำตัวจับเวลาให้ใช้การไม่ได้ ถ้าคุณต้องการก่อกำเนิดอีเวนต์ n ครั้ง ต่อวินาที ให้ใช้สูตร $1000/n$ เพื่อหาการตั้งค่าสำหรับคุณสมบัติ Interval เช่นถ้าคุณต้องการให้อีเวนต์ของตัวจับเวลาเกิดขึ้นทุกๆ ครึ่งวินาที (สองครั้งต่อวินาที) การคำนวณ $1000/2$ แสดงว่าควรตั้งค่า Interval ให้เป็น 500

อีเวนต์

ไม่มีกระบวนการหรือกลวิธีดำเนินการบนออบเจกต์ตัวจับเวลา

คอมมอนไดอะล็อกบ็อกซ์

ไมโครซอฟต์ได้สร้างชุดคอมมอนไดอะล็อกบ็อกซ์สำหรับฟังก์ชันมาตรฐานของ Windows เพื่อให้ผู้ใช้มีอินเตอร์เฟซแบบมาตรฐานสำหรับการดำเนินการร่วมในแอปพลิเคชันที่ดำเนินงานบน Windows รหัสสำหรับฟังก์ชันเหล่านี้จัดจำหน่ายเป็นครั้งแรกมาพร้อมกับ Windows 3.1 แต่สามารถนำมาใช้กับ Windows 3.0 ถ้าเติมไฟล์ commdlg.dll เข้าไปในไดเรกทอรี System Visual Basic ให้แนวทางที่จะเข้าถึงฟังก์ชันเหล่านี้โดยผ่านเครื่องมือสร้างคอมมอนไดอะล็อก การใช้เครื่องมือสร้างคอมมอนไดอะล็อกทำให้มั่นใจว่าแอปพลิเคชันที่สร้างขึ้นจะสอดคล้องกันกับรูปลักษณะมาตรฐานของการดำเนินการที่สนับสนุน

เครื่องมือสร้างคอมมอนไดอะล็อกตามที่แสดงในรูป มีลักษณะคล้ายกับเครื่องมือสร้างตัวจับเวลา กล่าวคืออยู่ในสภาพที่ไม่สามารถเห็นได้ในเวลาดำเนินการ อย่างไรก็ตาม เมื่อบรรจุไอคอนไว้บนแบบฟอร์มแบบใดตัวแบบหนึ่งแล้ว จะสามารถก่อให้เกิดหนึ่งในการทำหน้าที่มาตรฐานห้าอย่างที่เครื่องมือสนับสนุนอยู่ การทำหน้าที่ที่ว่านี้คือ ไดอะล็อกบ็อกซ์มาตรฐานของ Windows เพื่อใช้กับ Open , Save As , Color , Font และ Printer นอกจากนี้เครื่องมือสร้างคอมมอนไดอะล็อกยังสามารถก่อให้เกิดระบบความช่วยเหลือของ Windows การที่จะสามารถเข้าถึงเครื่องมือสร้างคอมมอนไดอะล็อกได้นั้น จะต้องบรรจุไฟล์ Commdeg.vbx เข้าไปในวินโดว์ Project

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้拿去ใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

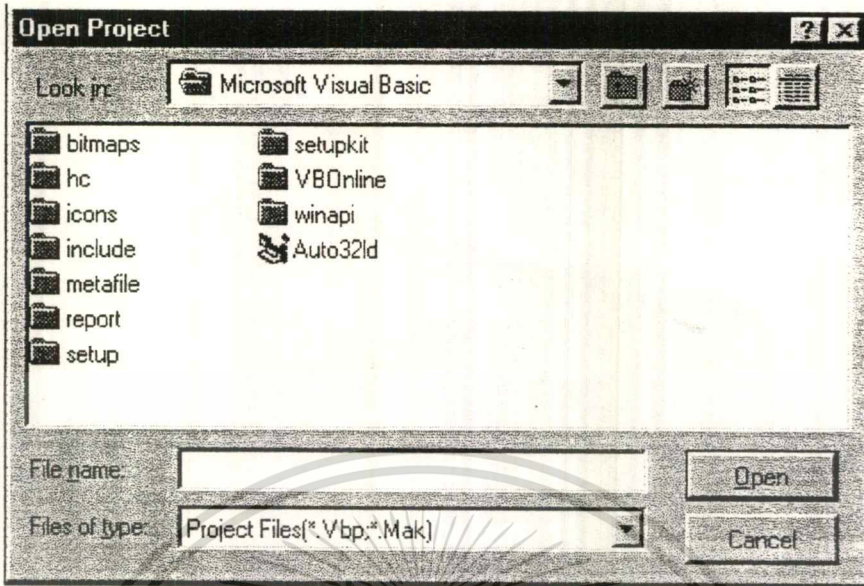


ภาพที่ 9 เครื่องมือสร้างคอมมอนไดอะล็อก(CommonDialog)

การเอ็ชชีควฟังก์ชันที่เลือกไว้เริ่มต้นขึ้นเมื่อได้ตั้งคุณสมบัติ Action ของตัวควบคุมคอมมอนไดอะล็อก ข้อความตั้งของ Visual Basic จะหยุดชะงักจนกว่าจะบอกเลิกไดอะล็อกบ็อกซ์หลังจากนั้นจะสามารถตรวจสอบคุณสมบัติของตัวควบคุมเพื่อหาว่ามีกิจกรรมอะไรเกิดขึ้น ไม่มีอีเวนต์หรือกลวิธีที่เกี่ยวข้องกับตัวควบคุมคอมมอนไดอะล็อก เรากำหนดการดำเนินการของตัวควบคุมโดยการตั้งค่าคุณสมบัติ และหลังจากนั้นกำหนดค่าให้แก่คุณสมบัติ Action นอกเหนือจากคุณสมบัติ Action แล้วฟังก์ชันทั้งหมดของคอมมอนไดอะล็อกจะใช้คุณสมบัติ CancelError ด้วย ค่าโดยปริยายของ CancelError คือ False แต่ถ้าตั้งให้เป็น True Visual Basic จะบ่งบอกความผิดพลาดออกมาถ้าผู้ใช้เลือกปุ่ม Cancel ในไดอะล็อกบ็อกซ์ ถ้ามีแผนจะตั้ง CancelError ให้เป็น True ควรจะใช้คุณลักษณะจัดการความผิดพลาดของ Visual Basic

Open

ไดอะล็อกบ็อกซ์ Open ตาที่แสดงในรูป จะเกิดขึ้นเมื่อกำหนดค่า 1 ให้แก่คุณสมบัติ Action เราสามารถตั้งคุณสมบัติดังต่อไปนี้เพื่อควบคุมลักษณะของไดอะล็อกบ็อกซ์ที่ปรากฏออกมา ควรทราบไว้ด้วยว่าไดอะล็อกบ็อกซ์นี้ไม่ได้เปิดไฟล์จริงๆ แต่เป็นการให้ผู้ใช้สามารถเลือกชื่อไฟล์ได้เท่านั้น



ภาพที่ 10 ไดอะล็อกบ็อกซ์ Open

คุณสมบัติ

DialogTitle

คุณสมบัตินี้สามารถตั้งให้เป็นสตริงใดๆ สตริงจะแสดงอยู่ในไตเติลบาร์ของไดอะล็อกบ็อกซ์โดยปริยายแล้วชื่อเรียกจะเป็น "Open"

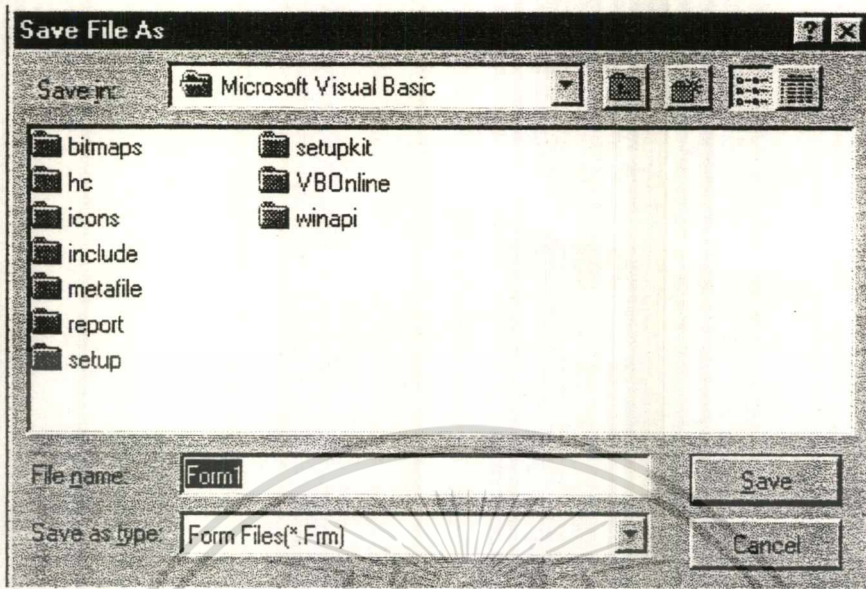
FileName

ใช้คุณสมบัตินี้ตั้งชื่อเริ่มแรกในเท็กซ์บ็อกซ์ File Name ของไดอะล็อกบ็อกซ์หลังจากที่บอกลึกไดอะล็อกบ็อกซ์แล้ว อ่านคุณสมบัตินี้เพื่อหาชื่อของไฟล์ที่เลือก

Filter

เราสามารถตั้งคุณสมบัตินี้เพื่อให้จำกัดชื่อไฟล์ที่จะปรากฏอยู่ในไฟล์ลิสต์บ็อกซ์ คุณสมบัติ Filter ต้องเป็นสตริงข้อความที่มีคู่ส่วนประกอบหนึ่งคู่หรือมากกว่า คู่ส่วนประกอบประกอบไปด้วยคำบรรยายและอักขระวิลด์การ์ด แยกออกจากกันด้วยสัญลักษณ์ไปป์(1) ถ้ามีคู่ส่วนประกอบหลาย ๆ คู่ ให้แยกออกจากกันด้วยสัญลักษณ์ไปป์เช่นเดียวกัน เช่น สตริงดังต่อไปนี้ กำหนดคู่ส่วนประกอบสามคู่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 11 ไดอะล็อกบ็อกซ์ Save As

คุณสมบัติ

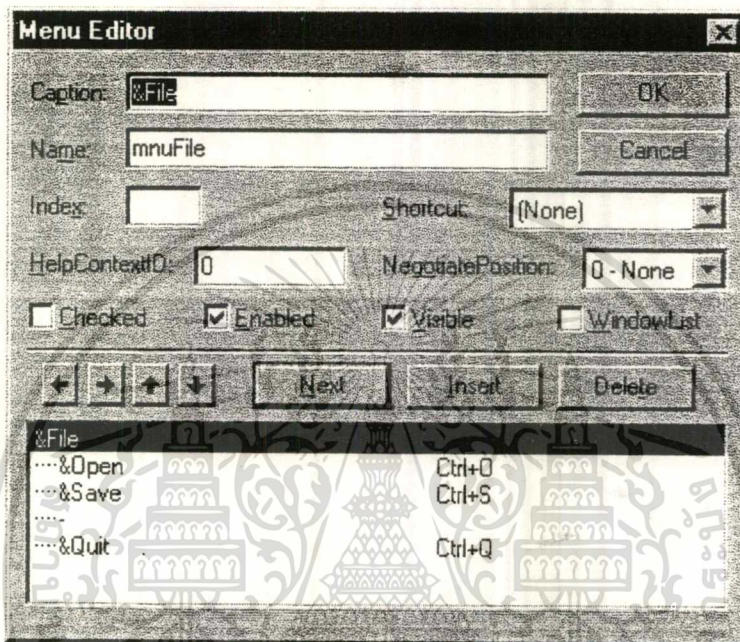
DefaultExt คุณสมบัตินี้สามารถตั้งให้เป็นสตริง 1 ถึง 3 อักขระ ซึ่งนำไปใช้เป็นส่วนขยายโดยปริยายในเวลาที่กำหนดชื่อไฟล์โดยไม่มีส่วนขยาย

เมนู(MENU)

แม้ว่าในตอนนี้นักเรียนได้สร้างแอปพลิเคชันขนาดเล็กหลายๆ โปรแกรมขึ้น แต่คุณก็อาจรู้สึกว่ามีโปรแกรมใดที่ดูเหมือน “แอปพลิเคชันจริงๆ” เหตุใดจึงเป็นเช่นนี้ สาเหตุก็มาจากการที่แอปพลิเคชันที่สร้างขึ้นนั้นไม่มีเมนูบาร์ อย่างไรก็ตาม คุณสามารถแก้ไขสถานการณ์นี้ได้เพราะ Visual Basic ให้แนวทางที่ใช้สร้างเมนูต่างๆ ได้โดยง่ายดาย เมนูดำเนินการด้วยแนวทางแบบอาศัยอีเวนต์คล้ายกับออบเจกต์ที่เราพิจารณามาแล้ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

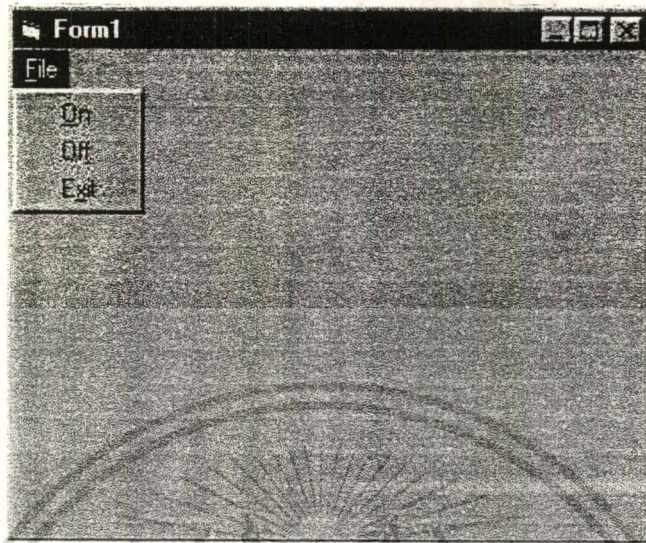
ในการที่จะสร้างเมนูสำหรับแอปพลิเคชันหนึ่งๆ คุณจะต้องใช้วินโดว์ Menu Design ของ Visual Basic ตามที่แสดงอยู่ในรูป สร้างโครงการใหม่ขึ้นหนึ่งโครงการ และเปิดวินโดว์นี้โดยการเลือกคำสั่ง Menu Design จากเมนู Windows หรือโดยการกดปุ่มเมาส์บนปุ่ม Menu Design Windows บนทูลบาร์ของ Visual Basic



ภาพที่ 12 วินโดว์ออกแบบเมนูของ Visual Basic

คุณสร้างเมนูบาร์ของแอปพลิเคชันหนึ่ง ๆ ด้วยการสร้างลำดับชั้นของชั้นข้อมูลในวินโดว์ Menu Design สำหรับชั้นเมนูแต่ละชั้น คุณจะต้องนิยามคุณสมบัติ Caption และคุณสมบัติ Name ข้อความจำเป็นเป็นข้อความที่จะปรากฏขึ้นในเมนูบาร์หรือบนเมนูแบบคิ่งลง ซึ่งของชั้นเมนูจะระบุชั้น เมื่อนั้นไว้ในรหัสโปรแกรมของคุณ การที่จำเป็นต้องใช้ชื่อก็เพราะว่าชั้นเมนูแต่ละชั้นจะมีกระบวนการ Click

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 13 แบบฟอร์มพร้อมกับเมนูบาร์และเมนูแบบดึงลง

ถ้าคุณกดปุ่มเมาส์บนหน่วยเมนูในแบบฟอร์ม Visual Basic จะเปิดวินโดว์รหัสสำหรับกระบวนการงาน Click ที่สอดคล้องกัน ในวินโดว์นี้คุณเขียนรหัสสำหรับการดำเนินการที่จะเกิดขึ้นเมื่อผู้ใช้เลือกขึ้นเมนู เช่น กดปุ่มเมาส์บนขึ้นเมนู Exit ของเมนู File เพื่อเปิดวินโดว์รหัสสำหรับกระบวนการงานอีเวนต์ mnuFileExit_Click คำสั่งของ Visual Basic ที่ใช้ยุติแอปพลิเคชันคือ End ดังนั้นกระบวนการงานสำหรับขึ้นเมนู Exit ควรจะเป็นดังนี้

```
Sub mnuFileExit_Click()
```

```
End
```

```
End Sub
```

บ็อกซ์ใส่ข้อความตั้ง Debug.Print ในกระบวนการงานเมนูต่างๆ และหลังจากนี้ดำเนินการแอปพลิเคชันอีกครั้ง ในขณะที่เลือกขึ้นเมนูต่างๆ นั้นให้จับตาดูที่วินโดว์ เพื่อเป็นการยืนยันว่ากำลังเอ็กซีคิวต์กระบวนการงานเมื่อนั้นอยู่

ออปชันเพิ่มเติมของวินโดว์ Menu Design

วินโดว์ Menu Design ยังประกอบไปด้วยออปชันที่น่าสนใจอีกจำนวนหนึ่ง เช่น ถ้าคุณป้อนไฮเฟน (-) เป็นข้อความจำหน้า Visual Basic ก็จะวางแท่งแยกเมนูเอาไว้ในเมนู ซึ่งเป็นการจัดกลุ่มขึ้นเมนูที่สัมพันธ์กัน (คุณต้องกำหนดชื่อให้แก่อนุมูลไฮเฟน แม้ว่าจะไม่ได้นำไปใช้ในโปรแกรมก็ตาม) คุณยังสามารถกำหนดแป้นลัดให้แก่อนุมูล เปิดวินโดว์ Menu Design อีก

ครั้งหนึ่งและเลือก E&xit หลังจากนั้นกดปุ่มเมาส์บนปุ่มเมาส์บนปุ่มลูกศรซึ่งที่อยู่ด้านข้างบ็อกซ์ Shortcut คุณยังสามารถกำหนดการเคาะแป้นใด ๆ ที่ลงไว้ในเมนูแบบดั่งลงให้แก่ขึ้นเมนู Exit

เช็kb็อกซ์ Checked,Enabled และ Viseble ในวินโดว์ Menu Design จะตรงกันกับคุณสมบัติของขึ้นเมนูที่เลือกไว้ แต่ละคุณสมบัติสามารถตั้งให้เป็น True ตามที่แสดงด้วย X ในเช็kb็อกซ์หรือ False(เช็kb็อกซ์ว่าง ๆ) การตั้งคุณสมบัติ Chedked ให้เป็น True เป็นการวางเครื่องหมายถูกเอาไว้บนเมนูข้างขึ้นเมนู ซึ่งแสดงว่าอปชันอยู่ในสภาพพร้อมใช้งาน (คุณสามารถเห็นเครื่องหมายถูกเช่นนั้นอยู่ถัดจากขึ้น Tool Bar บนเมนู View ของ Visual Basic) คุณสมบัติ Enabled จะถูกตั้งให้เป็น True อย่างโดยปริยาย ถ้าคุณตั้งคุณสมบัตินี้เป็น False จะวาดขึ้นเมนูเป็นสีเทา และไม่สามารถเลือกจากเมนูถ้าคุณตั้งคุณสมบัติ Visible เป็น False ขึ้นเมนูก็จะไม่ปรากฏบนเมนูเลย

จากรูปแสดงให้เห็นว่าเมนู Sound ของแอปพลิเคชันมีลักษณะเช่นใด ถ้าคุณเพิ่มเติมแท่งแยกสองแท่งไปที่ระดับแรก ตั้งคุณสมบัติ Checked ของขึ้นเมนู mnuSouPreLoud ให้เป็น True เพิ่มเติมขึ้นเมนูใหม่ชื่อว่า Really Loud และตั้งคุณสมบัติ Enabled ของขึ้นเมนูใหม่ให้เป็น False

การตั้ง Chedked,Enabled และ Visible ในวินโดว์ Menu Design แสดงถึงการตั้งคุณสมบัติที่เวลาเริ่มใช้งาน คุณสามารถเปลี่ยนแปลงการตั้งค่าต่างๆ เหล่านี้ในรหัสโปรแกรมของคุณด้วยข้อความสั่งที่มีรูปแบบดั่งต่อไปนี้

```
object.property = {True / False}
```

อปชันอื่นๆ ที่ปรากฏในวินโดว์ Menu Design เช่น Windows List, Index และ HelpContextID เป็น คุณลักษณะระดับสูงที่อยู่นอกขอบเขตของหนังสือเล่มนี้ คุณสามารถพบสารสนเทศเกี่ยวกับคุณลักษณะเหล่านี้ได้ในคู่มืออ้างอิงของ Visual Basic

ป๊อปอัปเมนู(Popop menu)

แม้ว่าตามปกติเมนูจะเกิดขึ้นจากเมนูบาร์ แต่ในบางครั้งอาจต้องการแสดงเมนูจากตัวควบคุมมนุษย์ชนิดนี้เรียกว่า ป๊อปอัปเมนู (popup menu) ทั้งนี้เพราะตามปกติจะไม่เห็นแต่จะ “ผุดขึ้น” เมื่อกดปุ่มเมาส์บนตัวควบคุม เราสามารถจัดเตรียมเมนูใดๆ ให้เกิดขึ้นในลักษณะผุดขึ้น โดยการใช้กลวิธี Popop Menu

ตัวอย่างเช่น เราสามารถโยงเมนู View Code ตามที่แสดงในรูป เข้ากับสภาพของปุ่มควบคุม

ควบคุม ใช้เครื่องมือสร้างภาพเพื่อสร้างบ็อกซ์ภาพขึ้นบนจอภาพ และหลังจากนี้ใช้เครื่องมือสร้างรูปทรงเพื่อวาดปุ่มควบคุมทรงกลมไว้ภายในบ็อกซ์ภาพ (ตอนแรกเริ่มรูปทรงจะเป็นสี่เหลี่ยมในการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาก็เป็น เมื่ออยู่ให้เหมาะสมกับเนื้อหาและการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

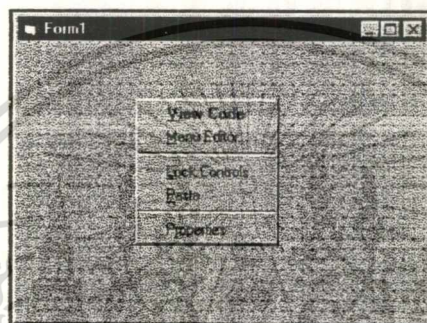
สร้างวงกลมตั้งคุณสมบัติ Shape ให้เป็น 3-Circle) ป้อนรหัสต่อไปนี้อยู่สำหรับอีเวนต์ MouseDown ในบ็อกซ์ภาพ

```
Sub Picture1_MouseDown (Button As Integer, Shift As Integer, X As Single, Y As Single)
```

```
    PopupMenu mnuView
```

```
EndSub
```

เมื่อใช้คอปุมเมาส์ภายในบริเวณของบ็อกซ์ภาพ จะทำให้เกิดเมนู View Code ขึ้นตามทีแสดงไว้ในรูป



ภาพที่ 14 ป๊อปอัพเมนู(Popupmenu)

ไฟล์สำหรับ Visual basic

ก่อนที่ผู้อ่านจะเริ่มต้นเรียนรู้กลวิธีการใช้งานเอนไวรอนเมนต์และวิธีการสร้างแอปพลิเคชัน โดยอาศัย VB/Win ผู้เขียนก็ขอแนะนำให้อ่านได้รู้จักกับไฟล์ชนิดต่างๆ ที่สามารถนำมาใช้งานหรือถูกสร้างโดย VB/Win เสียก่อน ซึ่งการรู้จักกับไฟล์และข้อจำกัดเหล่านี้เป็นสิ่งที่จำเป็นเพราะเมื่อเกิดข้อผิดพลาดขณะกำลังคอมไพล์แอปพลิเคชัน (Compile time error)

ไฟล์โปรเจกต์ (Project File)

เป็นไฟล์ที่มีนามสกุล .MAK ซึ่งใช้ในการจัดเก็บไคเรคทอรีและชื่อของไฟล์ฟอร์ม โมดูล โมดูลคลาส ไฟล์ทรัพยากร ไฟล์ไดนามิกไลบรารี และ custom control ที่มีในโปรเจกต์ และ ภายในไฟล์โปรเจกต์เองยังถูกใช้ในการจัดเก็บค่าตัวเลือกต่างๆ เอาไว้ด้วย

ไฟล์โมดูลฟอร์ม (Form Module File)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เป็นไฟล์ที่ใช้จัดเก็บรายละเอียดต่างๆ ของฟอร์มเอาไว้ ไม่ว่าจะเป็นคุณลักษณะต่างๆ เช่น ตำแหน่งขนาด รูปร่าง ข้อความของคอนโทรลในฟอร์ม การประกาศค่าคงที่ ตัวแปร โพรซีเจอร์ ทัวไป หรือโพรซีเจอร์เหตุการณ์ในระดับฟอร์ม (Form-level) ไฟล์ชนิดนี้จะมีนามสกุล .FRM

ไฟล์ โมดูลมาตรฐาน (Standard Module File)

เป็นไฟล์ที่ใช้กับโพรซีเจอร์ การประกาศตัวแปรโครงสร้าง การประกาศฟังก์ชันวินโดว API หรือ ฟังก์ชัน อื่นๆ ในไฟล์ .DLL จัดเก็บค่าคงที่ประเภทครอบคลุม (Global Constant) หรือตัวแปรประเภทครอบคลุม (Global Variable) ซึ่งโพรซีเจอร์ที่สร้างในไฟล์นี้โดยปกติจะถูกระบุกำหนดให้เป็น โมดูลแบบ Public อยู่แล้ว ไฟล์ชนิดนี้จะมีนามสกุล .BAS

ไฟล์ โมดูลคลาส (Class Module File)

เป็นไฟล์ที่ใช้จัดเก็บการนิยามคุณสมบัติ (property) และวิธี (method) ของคลาสเอาไว้ ซึ่งคลาสนี้หมายถึงการนิยามออบเจกต์อย่างเป็นทางการ ซึ่ง VB/Win จะใช้คลาสทำหน้าที่เป็นแม่แบบสำหรับการสร้างออบเจกต์ในขณะที่กำลังทำงาน (run-time) และคลาสจะนิยามคุณสมบัติและวิธีของออบเจกต์เพื่อให้สามารถนำมาใช้ควบคุมพฤติกรรมการทำงานของออบเจกต์นั้นๆ ไฟล์ชนิดนี้จะมีนามสกุลเป็น .CLS

ไฟล์ โมดูลทรัพยากร (Resource File)

ไฟล์ที่ใช้จัดเก็บทรัพยากรต่างๆ ที่สามารถนำมาใช้แสดงผลใน VB/Win ได้ การอาศัยไฟล์ทรัพยากรนี้ทำให้ผู้อ่านสามารถแก้ไขการแสดงผลโดยการแก้ไขเฉพาะข้อมูลต่างๆ ในไฟล์ทรัพยากรนี้โดยไม่จำเป็นต้องแก้ไขโค้ด สำหรับโปรเจกต์หนึ่ง ๆ ผู้อ่านสามารถรวมเอาไฟล์ทรัพยากรนี้เข้ามาได้เพียงไฟล์เดียวเท่านั้น ซึ่งไฟล์ชนิดนี้จะมีนามสกุล .RES

ไฟล์ custom control (Custom Control File)

ไฟล์คอนโทรลเป็นไฟล์ที่มีนามสกุล .VBX หรือ .OCX ซึ่งใช้เป็นไฟล์ไดนามิกลิงค์ไลบรารีชนิดหนึ่งที่ถูกออกแบบพิเศษสำหรับ VB/Win สำหรับจัดเก็บออบเจกต์เพื่อใช้เป็นเครื่องมือในการพัฒนาแอปพลิเคชัน เมื่อผู้อ่านทำการโหลดไฟล์เหล่านี้เข้ามารวมในโปรเจกต์ก็จะปรากฏเป็น ไอคอนของคอนโทรลในหน้าต่างทูลบ็อกซ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไฟล์ไดนามิกลิงค์ไลบรารี (Dynamic link library File)
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งยังมีเหตุและผลที่ต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ไฟล์ชนิดนี้เป็นไฟล์ที่มีนามสกุล .DLL แบ่งออกเป็น 2 ประเภท คือประเภทที่เขียนขึ้นมาสำหรับใช้กับภาษาซี เป็นไฟล์ที่มักกับวินโดว ซึ่งฟังก์ชันต่างๆ ที่มีในไฟล์เหล่านี้บางฟังก์ชันอาจจะไม่คอมแพททิเบิลกับ VB/Win ร้อยเปอร์เซ็นต์ ดังนั้นในการใช้งานฟังก์ชันเหล่านี้ผู้อ่านควรจะกระทำด้วยความระมัดระวัง เพราะอาจจะทำให้แอฟพลิเคชันหยุดการทำงานและค้างได้ และอีกประเภทหนึ่งคือไฟล์ที่ได้รับการออกแบบเป็นพิเศษสำหรับ VB/Win ก่อนที่ผู้อ่านจะเรียกใช้ฟังก์ชันใดๆ ในไลบรารี จะต้องประกาศฟังก์ชันเหล่านั้นเสียก่อนด้วยประโยค Declare เพื่อให้ VB/Win ทำการต่อเชื่อมเข้ากับฟังก์ชันนั้นๆ ในไฟล์ .DLL โดยที่การประกาศฟังก์ชันของไฟล์ .DLL จะสามารถกระทำได้เฉพาะในส่วนการประกาศของฟอร์ม หรือในส่วนการประกาศของโมดูลเท่านั้น

โมดูล (Module)

สำหรับแอฟพลิเคชันที่มีขนาดโตๆ หรือที่ต้องการข้อมูลชนิดกำหนดโดยผู้ใช้ (user-define type) ที่สามารถถูกอ้างอิงได้จากทุก ๆ ฟอร์มของแอฟพลิเคชัน เราจำเป็นต้องมีการเพิ่มโมดูลเข้ามารวมในโปรเจกต์ ซึ่งโพรซีเจอร์ทั่วไปหรือตัวแปร (ที่ถูกประกาศในส่วนการประกาศของโมดูล) ต่างๆ ที่มีในโมดูลมีข้อจำกัดที่แตกต่างจากโพรซีเจอร์ทั่วไปของฟอร์ม ดังนั้นการสร้างแอฟพลิเคชันหลายๆ แอฟพลิเคชันในบางครั้งเราจะสร้าง โมดูลที่ประกอบด้วยโพรซีเจอร์ทั่วไปที่สามารถใช้งานร่วมกันได้จากหลายๆ แอฟพลิเคชันเพื่อลดเวลาการสร้างแอฟพลิเคชันลงไป

ออปเจกต์ (Object)

ตามที่เคยกล่าวมาแล้วเมื่อผู้อ่านทำการสร้างแอฟพลิเคชันขึ้นมา นั่นก็หมายความว่าเรากำลังมีการใช้งานออปเจกต์ต่างๆ ของ VB/Win ซึ่งออปเจกต์เหล่านี้อาจจะเป็นฟอร์ม คอนโทรล หรือออปเจกต์ของระบบก็ได้ และในการเขียนโค้ดเพื่อจัดการเกี่ยวกับคอนโทรล ผู้อ่านสามารถทำได้โดยการอ้างอิงถึงออปเจกต์โดยตรงหรือโดยการสร้างตัวแปรออปเจกต์ (object variable) ขึ้นมาเพื่อให้ตัวแปรนั้นอ้างอิงถึงออปเจกต์ที่ต้องการได้ ตามที่ได้กล่าวถึงชนิดข้อมูลในบทที่ผ่านมา ผู้อ่านก็จะเห็นว่าตัวแปรออปเจกต์นี้ก็จะมีความสัมพันธ์เป็นไปตามกฎของตัวแปรทั่ว ๆ ไปของ VB/Win ดังนั้นในการใช้งานจึงสามารถทำได้ง่ายๆ เช่นเดียวกับตัวแปรข้อมูลอื่นๆ ในการสร้างตัวแปรออปเจกต์ผู้อ่านสามารถกระทำได้โดยการใช้ประโยคสำหรับประกาศตัวแปรทั่วไปดังนี้

Dim

ใช้ประกาศตัวแปรออปเจ็กต์ ถ้าหากผู้อ่านประกาศตัวแปรออปเจ็กต์ภายในโพธิ์เคอร์ตัวแปรที่ถูกสร้างขึ้นใหม่นี้จะสามารถถูกเรียกใช้งานได้เฉพาะภายในโพธิ์เคอร์นั้นๆ เท่านั้น และตัวแปรนี้จะถูกลบทิ้งทันทีที่โพธิ์เคอร์สิ้นสุดการทำงาน

ReDim

ใช้ประกาศตัวแปรออปเจ็กต์เช่นเดียวกับประ โยค Dim เพียงแต่ประ โยค ReDim จะใช้ในการสร้างตัวแปรออปเจ็กต์แบบอะเรย์เท่านั้น

Static

ใช้ประกาศตัวแปรออปเจ็กต์ภายในโพธิ์เคอร์ทั้งของฟอร์มและ โมดูล ซึ่งตัวแปรออปเจ็กต์ที่ประกาศด้วยประ โยคนี้จะถูกจัดเก็บข้อมูลเอาไว้บนสแต็กเมื่อโพธิ์เคอร์สิ้นสุดการทำงาน

Private

ใช้ประกาศตัวแปรออปเจ็กต์ภายในส่วนการประกาศของฟอร์มหรือ โมดูล เพื่อกำหนดให้ตัวแปรออปเจ็กต์ที่ได้สามารถถูกเรียกใช้งานได้เฉพาะฟอร์มหรือ โมดูลที่ประกาศเท่านั้น สำหรับโมดูลเนื่องจากตัวแปรที่ถูกประกาศในส่วนการประกาศของฟอร์มจะถูกกำหนดให้ Private อยู่แล้ว

Public

ใช้ประกาศตัวแปรออปเจ็กต์ภายในส่วนการประกาศของ โมดูลเท่านั้น เพื่อกำหนดให้ตัวแปรออปเจ็กต์ที่ได้สามารถถูกเรียกใช้งานจากทุกๆ ฟอร์มหรือ โมดูลใน โปรเจ็กต์

Global

ใช้ประกาศตัวแปรออปเจ็กต์ภายในส่วนการประกาศของ โมดูลนั้น เพื่อกำหนดให้ตัวแปรออปเจ็กต์ที่ได้สามารถถูกเรียกใช้งานจากทุกๆ ฟอร์มหรือ โมดูลใน โปรเจ็กต์

ออปเจ็กต์อื่นๆ

สำหรับออปเจ็กต์อื่นๆ นอกเหนือจากฟอร์มและคอนโทรลที่ VB/Win สนับสนุนก็ยังมีอีกหลายๆ ออปเจ็กต์ ซึ่งช่วยให้ผู้อ่านสามารถจัดการกับเอนไวรอนเมนท์ของแอปพลิเคชัน หรือ ไม่รู้ปริณิธใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ออปเจ็กต์ขึ้นมาเองก็ได้ ดังรายการต่อไปนี้

ออปเจ็กต์	รายละเอียด
App	เป็นออปเจ็กต์ระบบที่ช่วยในการจัดการกับข้อมูลต่างของ แอปพลิเคชัน เช่น Path, ชื่อของไฟล์ .EXE เป็นต้น
Clipboard	ออปเจ็กต์ที่ใช้ในการเข้าถึง Clipboard ของระบบ ซึ่งโดย อาศัยออปเจ็กต์ระบบนี้ผู้อ่านก็จะสามารถใช้ความสามารถ Clipboard ของระบบ ได้อย่างเต็มที่
ClassModule	ออปเจ็กต์ที่ VB/Win มีไว้เพื่อให้ผู้อ่านสามารถสร้างออปเจ็กต์ และ กำหนดคุณสมบัติและวิธี ได้เอง
ControlTemplate	ออปเจ็กต์ที่ VB/Win กำหนดให้ผู้อ่านสามารถใช้อ้างอิงถึง คอนโทรลของฟอร์ม ซึ่งผู้อ่านสามารถที่จะแก้ไขคุณสมบัติ ของคอนโทรล แก้ไขค่าของ Z-Order หรือค้นหาออปเจ็กต์ที่บรรจุ คอนโทรลเอาไว้ หรือเพิ่มคอนโทรลลงในฟอร์มก็ได้
Font	ออปเจ็กต์ที่ใช้ในการจัดการฟอนต์ของระบบที่นำมาใช้ใน แอปพลิเคชัน
Picture	ออปเจ็กต์ที่ใช้ในการจัดการกับบิตแมพ ไอคอน และไฟล์เมต้า สำหรับออปเจ็กต์นี้จะไม่เหมือนกับคอนโทรล PictureBox สำหรับออปเจ็กต์นี้มักถูกนำไปใช้งานในกรณีที่ต้องการ ใน อ้างอิงถึงบิตแมพหรือ ไอคอนที่มีการใช้งานบ่อยๆ และผู้อ่าน ยังสามารถกำหนดให้คอนโทรล PictureBox สามารถอ้างอิงถึง ออปเจ็กต์ที่ถูกออปเจ็กต์ Picture อ้างอิงถึง โดยใช้ประ โยค set ได้อีกด้วย
Printer	ออปเจ็กต์ที่ช่วยในการสื่อสารกับเครื่องพิมพ์ที่ติดตั้งให้กับระบบ ซึ่งในการพิมพ์ไม่ว่าจะเป็นข้อความหรือ บิตแมพ ภายใน VB/Win ต้องกระทำผ่านทาง ออปเจ็กต์ Printer นี้เอง
Property	ออปเจ็กต์ที่ใช้จัดการคุณสมบัติของฟอร์มหรือคอนโทรล
Screen	ออปเจ็กต์ที่ใช้ในการจัดการกับจอภาพและพอยน์เตอร์ ของเมาส์ เมื่ออยู่นอกเหนือจากพื้นที่ของฟอร์มของแอปพลิเคชัน
Err	ออปเจ็กต์ที่จัดเก็บข้อมูลต่างๆ ของข้อผิดพลาดในขณะที่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษานานัน ไม่นอนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

รันแอปพลิเคชัน

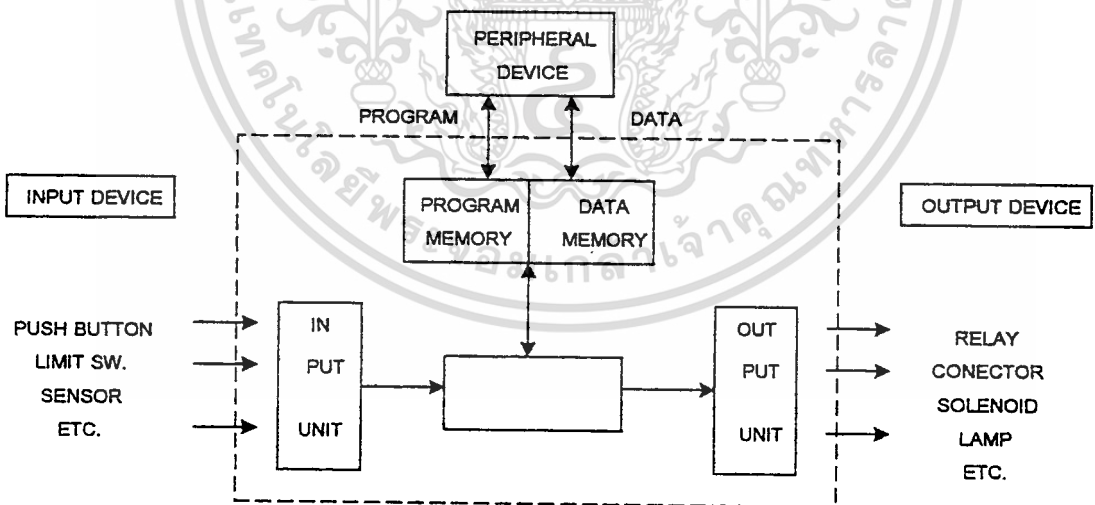
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ฟอร์มชนิด MDI

เป็นฟอร์มชนิด Multiple-Document Interface เป็นฟอร์มชนิดพิเศษของ VB/Win ที่มีคุณสมบัติสามารถใช้เป็นตัวบรรจุฟอร์มอื่นๆ เอาไว้ภายในได้ โดยที่ฟอร์มที่จะถูกแสดงผลภายในฟอร์ม MDI จะต้องเป็นฟอร์มที่มีคุณสมบัติ MDIChild มีค่าเท่ากับ True เท่านั้น และผู้อ่านสามารถกำหนดคอนโทรลต่างๆ ให้กับฟอร์ม MDI ได้เฉพาะเมนูและ คอนโทรล PictureBoX เท่านั้น

โครงสร้างและส่วนประกอบของ PLC

PLC เป็นอุปกรณ์ที่ใช้สำหรับควบคุมเครื่องจักรหรือระบบกระบวนการให้ทำงาน ตามโปรแกรมคำสั่งของผู้ใช้ และข้อมูลต่างๆ ที่ได้รับจากหน่วยอินพุต/เอาต์พุตของ PLC การทำงานของ PLC จะเป็นได้ทั้งการทำงานตามช่วงเวลาตามลำดับขั้นตอนฟังก์ชันทางคณิตศาสตร์และอื่นๆ โครงสร้างของ PLC มีส่วนประกอบคล้ายกับคอมพิวเตอร์หรืออาจกล่าวได้ว่า PLC เป็นคอมพิวเตอร์เฉพาะงานประเภทหนึ่ง ดังนั้น โครงสร้างโดยทั่วไปของ PLC จะประกอบด้วยส่วนที่เป็น ฮาร์ดแวร์และ ซอฟต์แวร์



ภาพที่ 15 แสดงโครงสร้างของ PLC

โครงสร้างทางฮาร์ดแวร์ ของ PLC ที่สำคัญแบ่งออกได้เป็น 4 ส่วนด้วยกันคือ

* หน่วยประมวลผลกลาง (CPU)

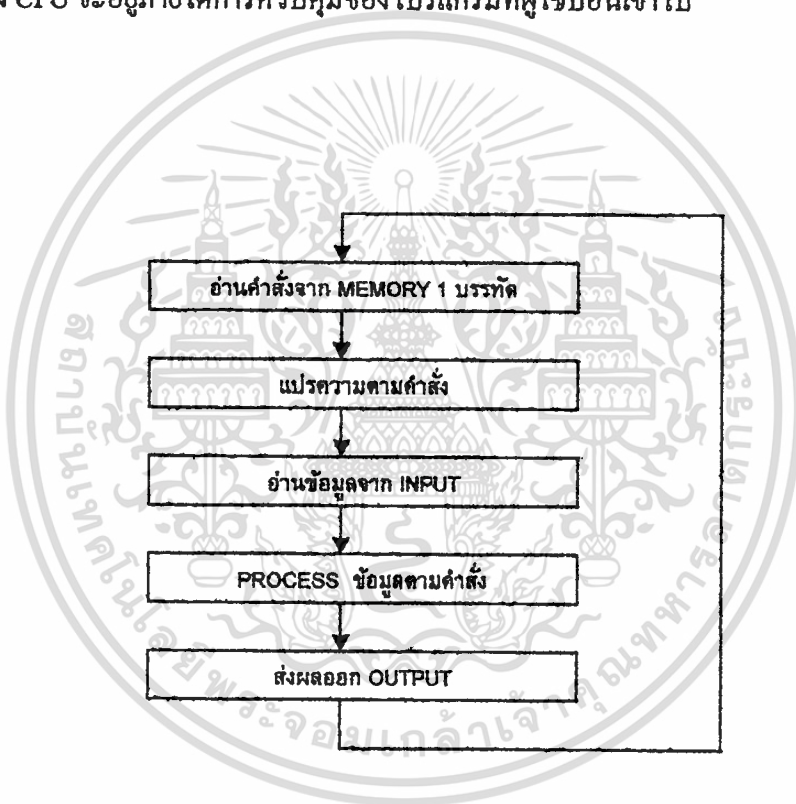
* หน่วยความจำ (MEMORY UNIT) การศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ * หน่วยอินพุต/เอาต์พุต (INPUT/OUTPUT UNIT) ถึงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

* หน่วยติดต่อกับอุปกรณ์ภายนอก (PERIPHERAL DEVICE)

หน่วยประมวลผลกลาง (CPU : Central Processing Unit)

หน่วยประมวลผลกลาง หมายถึง ส่วนที่ทำหน้าที่ควบคุมการทำงานของ PLC โดยทั่วไปแล้วมักจะใช้ ไมโครโปรเซสเซอร์ ชนิด 8 บิต เป็นตัวประมวลผลปคิหน้าที่ของ CPU คือ รับข้อมูลอินพุตเข้ามาทำการประมวลผล แล้วส่งผลที่ได้ออกไปยังเอาต์พุต จากนั้นก็จะวนกลับไปรับข้อมูลอินพุตเข้ามาอีกแล้วทำซ้ำ ๆ ในลักษณะเช่นนี้ไปเรื่อยๆ ซึ่งเรียกกันว่า การสแกน (scan) การทำงานของ CPU จะอยู่ภายใต้การควบคุมของ โปรแกรมที่ผู้ใช้ป้อนเข้าไป



ภาพที่16 Flow Chart แสดงการทำงานของ CPU

นอกจากนี้แล้ว CPU ยังทำหน้าที่รับส่งข้อมูลกับอุปกรณ์ติดต่อ (peripheral device) ตรวจสอบเช็คตัวเองและหน่วยความจำ , ตรวจสอบเช็คแหล่งจ่ายไฟ

หน่วยความจำ (Memory Unit)

เอกสารนี้เป็นเอกสารลิขสิทธิ์ของสถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง การใช้เป็นเอกสารอื่นโดยไม่ได้รับอนุญาตจะถือว่าผิดกฎหมาย

หน่วยความจำเป็นองค์ประกอบหนึ่งที่สำคัญของระบบ เพราะใช้เป็นที่เก็บ โปรแกรมและ ข้อมูล ขนาดของหน่วยความจำ จะเป็นตัวกำหนดขีดความสามารถของระบบปคคิมักจะมีขนาดวัดเป็น

เป็นจำนวน Step หรือบรรทัดของการโปรแกรม PLC แบ่งหน่วยความจำออกเป็น 2 ส่วนที่สำคัญด้วยกันคือ

1. หน่วยความจำระบบ (System memory) เก็บโปรแกรมบริหารระบบและข้อมูลของระบบ
2. หน่วยความจำผู้ใช้ (User memory) เก็บโปรแกรมผู้ใช้, ข้อมูลของหน่วยอินพุท/เอาต์พุท และอุปกรณ์ภายใน

โปรแกรมบริหารระบบและข้อมูลระบบ :

โปรแกรมบริหารระบบ และข้อมูลระบบไม่อนุญาตให้ผู้ใช้เปลี่ยนแปลงแก้ไขข้อมูลภายใน แต่ผู้ใช้สามารถตรวจสอบข้อมูลภายในหน่วยความจำส่วนของผู้ใช้หรือตรวจสอบสภาพการทำงานของ PLC หน่วยความจำผู้ใช้ ตารางข้อมูล และ โปรแกรมผู้ใช้ ตารางข้อมูล :

ข้อมูลของตารางข้อมูลมี 2 ลักษณะ คือ บิตข้อมูล "1" หรือ "0" ซึ่งแทนสถานะเปิดหรือปิดทางไฟฟ้าและข้อมูลแทนค่าตัวเลขสัญญาณอนาล็อก หรือตำแหน่งการควบคุมอุปกรณ์ภายนอก

ตารางข้อมูลแบ่งออกเป็น 4 ส่วนด้วยกันคือ

- * ตารางอินพุท
- * ตารางเอาต์พุท
- * ตารางรีเลย์ภายใน
- * ตารางรีจิสเตอร์

ตารางอินพุท : มีหน้าที่เก็บข้อมูลของสถานะหน่วยอินพุท ขนาดของตารางอินพุทจะเท่ากับจำนวนหน่วยอินพุทที่มากที่สุดของ PLC เสมอ เช่น PLC มีหน่วยอินพุท 64 จุด จะมีตารางอินพุทขนาด 64 บิต ทุกบิตของตารางอินพุทจะแสดงสถานะของอุปกรณ์ภายนอกที่เชื่อมกับหน่วยอินพุท บิตที่มีค่า "1" แสดงสถานะ "ON" และบิตที่มีค่า "0" จะแสดงสถานะ "OFF" ข้อมูลของตารางอินพุทจะเปลี่ยนสถานะตามอุปกรณ์ภายนอกที่ต่อเชื่อม ขณะเริ่มต้นการสแกนทุกครั้ง

ตารางเอาต์พุท : มีหน้าที่เก็บข้อมูลแสดงสถานะของหน่วยเอาต์พุทขนาดของตารางเอาต์พุทจะเท่ากับจำนวนเอาต์พุทที่มากที่สุดของ PLC เช่น PLC มีหน่วยเอาต์พุท 64 จุด จะมีตารางอินพุทขนาด 64 บิต ทุกบิตของตารางอินพุทจะแสดงสถานะของอุปกรณ์ภายนอกที่เชื่อมกับหน่วยอินพุท บิตที่มีค่า "1" แสดงสถานะ "ON" และบิตที่มีค่า "0" จะแสดงสถานะ "OFF" ข้อมูลของตารางอินพุทจะเปลี่ยนสถานะตามอุปกรณ์ภายนอกที่ต่อเชื่อม ขณะเริ่มต้นการสแกนทุกครั้ง

ตารางรีเลย์ภายใน : มีหน้าที่เก็บสถานะของอุปกรณ์ภายใน เช่นเดียวกับหน่วยอินพุท/เอาต์พุท รีเลย์ภายในใช้ประโยชน์ในการเก็บสถานะของการหน่วงเวลา การนับจำนวนและเก็บรักษาสถานะการควบคุมชั่วคราว เช่นเดียวกับรีเลย์ควบคุมของวงจรรีเลย์

ตารางรีจิสเตอร์ : มีหน้าที่เก็บข้อมูลที่ใช้แทนคำสั่งสัญญาณนาฬิกาหรือตำแหน่งการควบคุมอุปกรณ์ภายนอก ที่มีค่าเป็นตัวเลขไมใช่สถานะ ON หรือ OFF ตารางรีจิสเตอร์ของ PLC ประกอบด้วยอินพุทรีจิสเตอร์ และรีจิสเตอร์ภายใน ข้อมูลภายในรีจิสเตอร์อาจมีลักษณะเป็นเลขฐานสอง หรือ BCD (BINARY CODE DECIMAL)

โปรแกรมผู้ใช้ :

เป็นคำสั่งหรือ โปรแกรมที่ผู้ใช้เขียนขึ้น เพื่อควบคุมเครื่องจักรหรือกระบวนการให้ทำงานตามที่ต้องการ

หน่วยความจำชนิดต่างๆ :

ROM (Read Only Memory)

Rom เป็นหน่วยความจำที่ไม่อนุญาตให้ผู้ใช้เปลี่ยนแปลงหรือแก้ไขข้อมูลภายใน แต่สามารถเก็บรักษาข้อมูลไว้ได้แม้ว่าจะไม่มีกระแสไฟฟ้า เหมาะสำหรับเก็บ โปรแกรมบริหารระบบหรือ โปรแกรมผู้ใช้ที่เสร็จสมบูรณ์ไม่ต้องการแก้ไข

RAM (Random Access Memory)

Ram เป็นหน่วยความจำที่ผู้ใช้สามารถเปลี่ยนแปลงแก้ไขข้อมูลได้ เหมาะสำหรับเก็บโปรแกรมผู้ใช้ที่อยู่ในช่วงพัฒนา หรือต้องการเปลี่ยนแปลงบ่อยครั้ง ในการใช้งานจริงๆ แล้วจะต้องมีแหล่งจ่ายไฟสำรองต่อไว้เพื่อป้องกันไม่ให้ข้อมูลสูญหายเมื่อเกิดไฟดับ

EPROM (Erasable Programmable Read Only Memory)

EPROM เป็นหน่วยความจำสำหรับเก็บ โปรแกรม ที่มีการพัฒนาจนใช้งานได้ดีให้เป็นการถาวรและในการอัปเดต โปรแกรมจะทำโดยถ่ายข้อมูลจากหน่วยความจำ RAM ลงมาสู่หน่วยความจำ EPROM โดยอาศัยเครื่องอัตรชนิดพิเศษ (PROM WRITER) ต่อร่วมกับชุดของ PLC หน่วยความจำประเภทนี้ โปรแกรมจะไม่มีผลสูญหาย เมื่อเกิดไฟดับ แต่ถ้ามีความจำเป็นที่จะลบโปรแกรมภายใน ก็สามารถทำได้โดยใช้เครื่องล้างโปรแกรม

EEPROM (Electrically Erasable Programmable Read Only Memory)

เอกสารนี้เป็นเอกสารสงวนลิขสิทธิ์สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

EEPROM มีลักษณะคล้ายกับ EPROM คือเป็นหน่วยความจำที่เหมาะสมสำหรับเก็บโปรแกรม ผู้ใช้ที่เสร็จสมบูรณ์แล้ว การอัปเดตโปรแกรมจะอาศัยเครื่องอัปเดตโปรแกรมชนิดพิเศษเช่นเดียวกัน แต่คอนลบบ ข้อมูลจะใช้วิธีป้อนสัญญาณพัลส์ และการลบข้อมูลก็ไม่จำเป็นต้องลบทั้งหมด ผู้ใช้สามารถแก้ไขเฉพาะตำแหน่งที่ต้องการ

หน่วยอินพุท/เอาต์พุท (Input/Output Unit)

หน่วยอินพุท

หน่วยอินพุททำหน้าที่เชื่อมต่อระหว่าง CPU กับอุปกรณ์ภายนอกโดยรับค่าสถานะหรือปริมาณทางกายภาพต่างๆ จากอุปกรณ์ตรวจเช็ค (Sensor) ของเครื่องจักรหรือกระบวนการ เช่น Limit Switch, Push Button Switch, Proximity Switch, อุณหภูมิ, ระดับแรงดันกระแสหรืออื่นๆ ส่งไปยัง CPU เพื่อประมวลผลตาม โปรแกรมคำสั่งของผู้ใช้ ปกติหน้าที่ของหน่วยอินพุทคือ

- แปลงระดับสัญญาณเข้าให้เป็นระดับสัญญาณที่เหมาะสมกับระบบการทำงานของ CPU
- แบ่งสัญญาณภายนอกและภายในออกจากกัน เพื่อเป็นการป้องกันไม่ให้หน่วยประมวลผลได้รับความเสียหายเมื่อหน่วยอินพุทเกิดการลัดวงจร
- แก้ปัญหาการสั้นสะเทือนของหน้าสัมผัส

หน่วยเอาต์พุท

หน่วยเอาต์พุททำหน้าที่รับค่าสัญญาณที่ได้จากการประมวลผลไปขยายสัญญาณออกให้มีขนาดใหญ่พอที่จะขับอุปกรณ์ภายนอก เช่น มอเตอร์, วาล์ว, ปั๊ม, หลอดไฟ และอื่นๆ นอกจากนั้นแล้วหน่วยเอาต์พุทยังทำหน้าที่แบ่งสัญญาณภายในและสัญญาณภายนอกออกจากกัน เพื่อป้องกันการเสียหายที่อาจเกิดขึ้นได้

ดังที่ได้กล่าวมาแล้วว่า ในช่วงแรก PIC จะถูกใช้แทนรีเลย์สำหรับการควบคุมแบบ ON-OFF หรือ เปิด-ปิด เท่านั้น หน่วยอินพุท/เอาต์พุทในขณะนั้นจึงเป็น แบบที่ใช้สำหรับค่าสถานะลอจิกหรือ ON-Off หรืออาจกล่าวได้ว่า PLC ในช่วงนั้นเป็น PLC นั้นเอง ต่อมา PLC ได้ถูกพัฒนาขึ้นมาเรื่อยๆ จนมีการควบคุมแบบอนาล็อกรวมอยู่ด้วยปัจจุบันหน่วยอินพุท/เอาต์พุทของ PLC แบ่งออกเป็น 3 ประเภทด้วยกันคือ

1. อินพุท/เอาต์พุทแบบสถานะลอจิก
2. อินพุท/เอาต์พุทแบบอนาล็อก
3. อินพุท/เอาต์พุทแบบพิเศษ

อุปกรณ์ติดต่อภายนอก (Peripheral Devices)

เป็นอุปกรณ์แบบต่างๆ ที่อำนวยความสะดวกในการพัฒนาโปรแกรมสามารถใช้ร่วมกับ PC ชนิดเดียวกันได้หลายๆ ตัว หน้าที่ของอุปกรณ์ติดต่อภายนอกได้แก่

1. บื่อนโปรแกรมเข้าไปใน memory ของระบบ
2. ใช้ในการแก้ไข (Debug) โปรแกรม
3. ใช้ในการเก็บรักษา โปรแกรม
4. ใช้ในการพิมพ์โปรแกรม
5. ใช้แสดงสถานะการควบคุม

อุปกรณ์ติดต่อภายนอก มีหลายชนิด และเหมาะสมกับงานแต่ละอย่าง

Program Panel Unit

Crt, Monitor

Audio, Cassette

Eprom Writer

Printer

Graphic Plotter

โครงสร้างของหน่วยลอจิกอินพุท/เอาต์พุท

ในงานควบคุมแบบซีเคิร์นซ์ หน่วยอินพุท/เอาต์พุท PC ก็จำเป็นต้องมีลักษณะการทำงานที่เป็นแบบ ปิด-เปิด เช่นเดียวกับลักษณะของงานด้วย หน่วยอินพุท/เอาต์พุทของการควบคุมแบบนี้มีอยู่หลายแบบและหลายขนาดขึ้นอยู่กับประเภทและการเลือกนำไปใช้งาน

โครงสร้างของหน่วยลอจิกอินพุท

หน่วยอินพุทแบบนี้จะทำหน้าที่เชื่อมต่อระหว่าง CPU กับอุปกรณ์ภายนอกโดยรับค่าสถานะหรือปริมาณทางกายภาพต่างๆ ที่จะต้องอยู่ในลักษณะของการเปิดและปิดเท่านั้น แล้วส่งผลไปยัง CPU เพื่อประมวลผลตาม โปรแกรมคำสั่งของผู้ใช้ปกติหน้าที่ของหน่วยอินพุทแบบนี้คือ

-แปลงระดับสัญญาณเข้าไปให้เป็นระดับสัญญาณที่เหมาะสมกับระบบการทำงานของ

CPU

-แบ่งสัญญาณภายนอกและภายในออกจากกัน เพื่อเป็นการป้องกันไม่ให้หน่วยประมวลผลได้รับความเสียหายเมื่ออินพุทเกิดการลัดวงจร

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์สำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่ออนุญาตให้นำไปเผยแพร่หรือใช้ในการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น นอกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- แก้ปัญหาการสั้นสะพานของหน้าสัมผัส

จากหน้าที่หลักของหน่วยอินพุตดังกล่าวนี้ โครงสร้างภายในของหน่วยอินพุตแบบนี้จะประกอบไปด้วยส่วนสำคัญเนื่องจากระดับแรงดันอินพุตทางด้านอินพุตมีหลายระดับ ดังนั้นการเลือกใช้หน่วยอินพุตแบบนี้จึงต้องทำให้ถูกต้องและเหมาะสมกับชนิดของงาน หน่วยอินพุตแบบนี้สามารถแบ่งออกได้เป็น 2 แบบด้วยกันคือ

- อินพุตที่ใช้กับไฟกระแสตรง ปกติใช้ต่อกับอินพุตที่มีแรงดันกระแสต่ำคอนแทกขนาดเล็ก อยู่ในตู้ควบคุม หรือต่อกับ SENSOR ที่เป็น TRANSISTOR OUTPUT

- อินพุตที่ใช้กับไฟกระแสสลับเหมาะสำหรับต่อกับ CONTACT OUTPUT จากกระบวนการทำงานต่างๆ เช่น LIMIT SWITCH , PUSH BUTTON , CONTACTOR

โครงสร้างของหน่วยลอจิกเอาต์พุต

หน่วยเอาต์พุตจะทำหน้าที่เชื่อมต่อระหว่าง CPU กับ อุปกรณ์ภายนอกโดยให้ค่าสถานะต่างๆ แก่อุปกรณ์ด้านเอาต์พุตซึ่งจะอยู่ในลักษณะของการเปิดและปิดเท่านั้น โดยทั่วไปหน้าที่หลักของหน่วยเอาต์พุตแบบนี้คือ

- แยกสัญญาณระหว่าง CPU กับอุปกรณ์ภายนอก

- ขยายสัญญาณที่ออกจากหน่วย CPU ให้มีค่าสูงพอที่จะขับอุปกรณ์ภายนอกได้

หน่วยเอาต์พุตแบบนี้สามารถแบ่งออกได้เป็น 2 ประเภทเช่นเดียวกับหน่วยอินพุตกล่าวคือ หน่วยเอาต์พุตที่ใช้กับไฟกระแสตรงและกระแสสลับแต่หน่วยลอจิกเอาต์พุตบางประเภทก็สามารใช้ได้กับไฟทั้งสองลักษณะ เช่น RELAY OUTPUT

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การติดต่อสื่อสารข้อมูลทั่วไป

เครื่องควบคุม PLC / PC นอกจากจะใช้ในการควบคุมเครื่องจักรแล้ว ยังได้มีการพัฒนาขีดความสามารถในการสื่อสารข้อมูล การควบคุมระยะไกล การจัดการข้อมูล เพื่อเพิ่มประสิทธิภาพในการควบคุมให้กว้างยิ่งขึ้น ในบทนี้จะกล่าวถึงพื้นฐานการสื่อสารข้อมูลทั่วไป และการสื่อสารข้อมูลของเครื่องควบคุม PLC / PC

ส่วนประกอบเบื้องต้นในการสื่อสารข้อมูลแบ่งได้ออกเป็น 3 ส่วน คือ

1. ฝ่ายกำเนิดข้อมูล (Transmitter)
2. ตัวกลางในการส่งผ่านข้อมูล(Medium)
3. ฝ่ายรับข้อมูล (Receiver)

ซึ่งแต่ละส่วนมีความสัมพันธ์กันดังภาพ



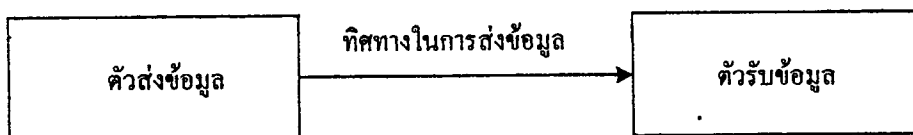
ภาพที่ 17 แสดงความสัมพันธ์ของส่วนประกอบหลักในการสื่อสารข้อมูล

การจำแนกรูปแบบในการสื่อสารข้อมูลอาจจัดแบ่งได้หลายวิธี เช่น แบ่งตามชนิดของสัญญาณของข้อมูล คือการสื่อสารข้อมูล อานาล็อก(Analog) และการสื่อสารข้อมูลดิจิทัล (Digital) หรือแบ่งตามวิธีการส่งข้อมูล ในบทนี้จะกล่าวถึงการสื่อสารข้อมูลดิจิทัล ซึ่งเป็นข้อมูลชนิดที่เกี่ยวข้องกับเครื่องคอมพิวเตอร์และ เครื่องควบคุม PLC / PC

วิธีการส่งข้อมูล

แบ่งออกเป็น 3 แบบคือ

1. การส่งแบบทิศทางเดียว (Simplex) การส่งแบบนี้ทิศทางของการส่งจะคงที่โดยกำหนดในครั้งแรกว่าผ่านใดเป็นฝ่ายส่ง ฝ่ายใดเป็นฝ่ายรับ การส่งแบบนี้แม้ไม่คล่องตัวแต่เหมาะสมในการใช้งานบางประเภท



ภาพที่ 18 การส่งแบบทิศทางเดียว(Simplex)

2.การส่งแบบฮาล์ฟดูเพล็กซ์(Half Duplex) เป็นการส่งข้อมูลในทิศทางใด ก็ได้ เวลาใดก็ได้ คือทั้งสองสถานีสามารถผลัดกันส่งแต่ส่งพร้อมกันไม่ได้



ภาพที่ 19 การส่งแบบฮาล์ฟดูเพล็กซ์ (Half Duplex)

3.การส่งแบบสองทิศทาง (Full Duplex) เป็นการส่งข้อมูลในทิศทางใด ก็ได้ เวลาใดก็ได้ คือสามารถส่งและรับข้อมูลในเวลาเดียวกันได้

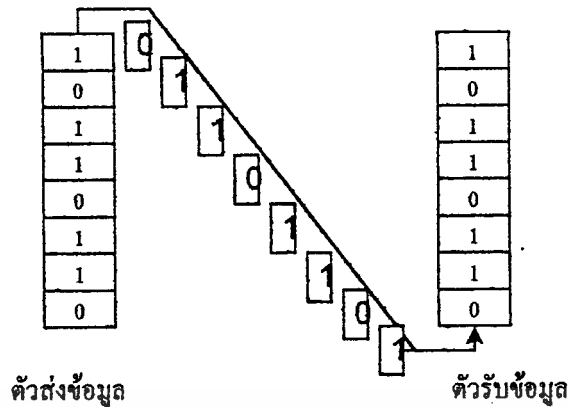


ภาพที่ 20 การส่งแบบสองทิศทาง (Full Duplex)

วิธีการสื่อสารข้อมูลดิจิทัลของคอมพิวเตอร์แบ่งเป็น 2 ชนิด

1. การสื่อสารข้อมูลแบบขนาน (Parallel)

เป็นการติดต่อสื่อสารโดยส่งข้อมูลออกเป็นครั้งละ 1 ไบต์ คือ ครั้งละ 8 บิตจากอุปกรณ์ส่งไปยังอุปกรณ์รับ ดังนั้นตัวกลางระหว่างเครื่องคอมพิวเตอร์ทั้งสองเครื่องจะต้องมีช่องทางให้ข้อมูลเดินทาง อย่างน้อย 8 ช่องทาง โดยมากจะใช้เป็นสายขนานแต่เนื่องจากมีสัญญาณสูญหายไปกับอิมพีแดนซ์ของสาย ซึ่งมีความสัมพันธ์กับระยะทางของสาย ดังนั้นระยะทางระหว่างเครื่องสองเครื่อง ไม่ควรเกิน 100 ฟุต ทำให้การส่งแบบนี้ได้ระยะทางไม่ไกลนักแต่



ภาพที่ 22 การสื่อสารข้อมูลแบบอนุกรม

ข้อตกลงในการติดต่อสื่อสาร (โปรโตคอล : Protocols)

ในการสื่อสารข้อมูลจะต้องมีกฎหรือข้อกำหนดในการสื่อสารข้อมูล หรือที่นิยมเรียกว่า โปรโตคอล (Protocols) ซึ่งเป็นส่วนที่จะกำหนดมาตรฐานในการควบคุมและ จัดการระบบการสื่อสารข้อมูล

สำหรับรายละเอียดที่กล่าวในหัวข้อนี้จะเกี่ยวข้องเฉพาะในส่วนของโปรโตคอลการควบคุมการเชื่อมโยงข้อมูล (Data Link Control Protocols หรือ DLCP) ซึ่งจัดการในส่วนของขั้นตอนและหลักการต่าง ๆ คือ โครงสร้างและรายละเอียดของข้อมูล วิธีในการสื่อสารข้อมูลการตรวจสอบแก้ไขความผิดพลาดของข้อมูล และขบวนการในการควบคุมการติดต่อสื่อสาร โดย (DLCP) แบ่งได้ตามโครงสร้างของข้อมูล 2 แบบ คือ Byte-Oriented Protocols และ Bit-Oriented protocols)

ไบท์โอเรียนโปรโตคอล (Byte-Oriented Protocols)

โปรโตคอลแบบนี้เป็นโปรโตคอลที่การสื่อสารข้อมูล และการควบคุมการทำงานจะทำได้โดยใช้ลักษณะข้อมูลที่เป็นตัวอักษร (character) หรือ ไบท์ (byte) หรือ อาจเรียกว่า Character Oriented Protocols

1.อะซิงโครนัสโปรโตคอล(Asynchronous protocols)

โปรโตคอลในการสื่อสารข้อมูลนี้จะใช้การสื่อสารข้อมูลแบบ Half-Duplex ที่มีลักษณะการสื่อสารข้อมูลแบบอะซิงโครนัส ซึ่งเป็นการสื่อสารข้อมูลแบบพื้นฐานที่ใช้มานานเป็นเวลานานแล้ว จึงมีรายละเอียดและขั้นตอนในการสื่อสารข้อมูลที่ทำให้มีโอกาสเกิดความผิดพลาดได้น้อย และยังมีข้อดีที่การสื่อสารข้อมูลแบบนี้มีใน โครงสร้างการทำงานที่ง่าย อุปกรณ์ที่ใช้ในการไม่วากรมใดๆ ทั้งสิ้น อีกทั้งยังมีเหตุผลเปลี่ยนแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สื่อสารสื่อสารข้อมูลก็ไม่สลับซับซ้อนและมีราคาถูก โปรโตคอลแบบนี้จึงไม่เหมาะสำหรับใช้ในระบบขนาดเล็ก

1. ไบนารีซิงโครนัสโปรโตคอล

โปรโตคอลแบบนี้จะมีลักษณะการทำงานที่ใช้งานข้อมูลเป็นลักษณะไบนารี และยังคงใช้การสื่อสารข้อมูลแบบซิงโครนัส มีรายละเอียดและขั้นตอนในการสื่อสารข้อมูลที่ให้ความน่าเชื่อถือมากกว่า อีกทั้งยังสามารถใช้อัตราเร็วในการสื่อสารข้อมูลที่สูงกว่าโดยตัวอย่าง ของการสื่อสารข้อมูลแบบนี้ที่ได้กำหนดเป็นมาตรฐานแล้วคือ การสื่อสารข้อมูลตามมาตรฐาน BSC (Binary Synchronous Communications) ซึ่งเป็นโปรโตคอลที่มีลักษณะของข้อมูลแบบไบนารีที่ได้รับความนิยมนำไปใช้งาน

บิตโอเรียนโปรโตคอล (Bit-oriented protocols)

โปรโตคอลแบบนี้เป็นโปรโตคอลที่การสื่อสารข้อมูล และการควบคุมการทำงานจะทำได้โดยใช้ลักษณะข้อมูลที่เป็นบิต (bit) โดยมีตัวอย่างของการสื่อสารข้อมูลในลักษณะนี้มีการกำหนดขึ้นเป็นมาตรฐานแล้ว คือ HDLC (High-level Data Link Control) โดยมีโครงสร้างของข้อมูลแบบซิงโครนัสเช่นเดียวกับ BSC แต่ต่างกันที่มีลักษณะของข้อมูลเป็นแบบบิต ซึ่งโปรโตคอลแบบนี้มีข้อดีที่สามารถสื่อสารข้อมูลแบบ full-duplex ได้ทำให้การสื่อสารข้อมูลได้รวดเร็วกว่าแต่โปรโตคอลแบบนี้ก็มีรายละเอียดและโครงสร้างในการสื่อสารข้อมูลที่สลับซับซ้อนมากทำให้การควบคุมการทำงานทำได้ยากและต้องใช้อุปกรณ์ที่มีราคาสูง จึงไม่เหมาะที่จะนำไปใช้งานกับระบบขนาดเล็ก

แพ็กเก็ต (Packet of Information)

รูปแบบของแพ็กเก็ตในระบบโครงข่ายจะประกอบด้วยส่วนต่าง ๆ ดังนี้

1. HEADER จะประกอบด้วย

- Preamble or start of packet indicator เป็นส่วนเริ่มแรกของแพ็กเก็ต และในบางระบบอาจใช้ในการซิงค์กับสัญญาณนาฬิกาของตัวส่งและตัวรับด้วย

- Control information ส่วนนี้เป็นข้อมูลที่บอกถึงวัตถุประสงค์ของแพ็กเก็ตนั้นว่าใช้ทำอะไรทำอะไรเช่นเพื่อการจัดการระบบเพื่อดู Status ของ mode หรืออื่น ๆ

นอกจากส่วนต่าง ๆ เหล่านี้แล้วในส่วน Header อาจจะมีส่วนที่เป็น Sequential Number เป็นส่วนที่บอกให้ทราบถึงลำดับของแพ็กเก็ตในกรณีที่มีข้อมูลมีความยาวหลายแพ็กเก็ต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

2. INFORMATION

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- Data field เป็นส่วนของข้อมูลจริงที่ต้องการจะส่ง

3. TAILER

- Frame Check Sequence (FCS) เป็นส่วนที่ใช้ตรวจสอบความถูกต้องของข้อมูล ซึ่งอาจเป็น parity bit, Check sum หรือ CRC เป็นต้น
- End of Packet Indicator เป็นส่วนที่บอกให้ทราบว่าสิ้นสุดของข้อมูลแล้ว

PREAMBLE	DA	SA	CONTROL	INFORMATION	FCS	STOP
HEADER				TAILER		

ภาพที่ 23 แสดงโครงสร้างของแพ็กเก็ต

การควบคุมความผิดพลาดในการส่งข้อมูล

การที่วงจรส่งข้อมูลขาดชั่วขณะ และผลกระทบต่อสัญญาณรบกวน (Noise) ทำให้ระดับแรงดันไฟฟ้าลดลงเป็นผลให้เกิดความผิดพลาดในการส่ง ดังนั้นต้องทำการค้นหาความผิดพลาดที่เกิดขึ้นระหว่างการส่งข้อมูลและแก้ไขความผิดพลาดให้ถูกต้อง

วิธีการเพิ่มบิตเข้าไปที่ข้อมูล (Parity) เพื่อตรวจสอบความผิดพลาดมี 2 วิธี

1. วิธี Parity แนวนอน

จะเพิ่ม 1 บิตเข้าไปที่แต่ละตัวอักษรที่จะส่ง โดยจะกำหนดว่าเป็นการตรวจสอบแบบ Odd หรือ Even Number แล้วผลรวมของข้อมูลจะเป็น Odd Number ที่ด้านรับจะทำการตรวจสอบว่าผลรวมของข้อมูลจะเป็น Odd number หรือ Even Number

2. วิธี Parity แนวตั้ง

จะตรวจสอบความผิดพลาดโดยการเพิ่ม 1 บิตเข้าไปที่ในแต่ละบล็อก

วิธี Patrol Diffuse Inspection หรือ การใช้ FCS: Frame Check Sequence

สำหรับข้อมูลอันดับหนึ่งจะสร้าง Error Unspecion Sign (CRC Sign) ขนาด 2 Byte ที่คำนวณได้บนพื้นฐานของกฎที่กำหนดไว้ แล้วเพิ่มเครื่องหมายไปที่ข้อมูลอันนั้น ที่ทางด้านรับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จะตรวจสอบข้อมูลโดย Inspection Sign ได้กำหนดไว้โดยสูตรการคำนวณที่เหมือนกัน(Blast Error ความผิดพลาดของข้อมูลที่ต่อเนื่อง) ก็สามารถตรวจสอบได้ทำให้มีความน่าเชื่อถือสูง

อัตราการส่งข้อมูล (Data Transmission Rate)

Transmission Rate นั้นแยกออกได้เป็น "Data Transmission Speed" และ "Modulation Speed"

1.Data transmission Speed

Data Transmission Speed หรือ Bit นั้นจะแสดงในรูปของจำนวน Bit ที่สามารถส่งได้ภายใน 1 วินาที มีหน่วยเป็น บิต ต่อ วินาที โดยเขียนว่า Bit / Sec หรือ Bps จะแสดง Definition ของ Data Transmission Speed จะแสดงด้วยสมการข้างล่างนี้

$$S = (1/T) \log_2 n$$

S : Data Transmission Speed

T : Continue Time ของ 1 pulse ของ Transmission Pulse

N : จำนวนของค่าสถานะใน 1 Pulse

ในกรณีที่ Pulse 1 ลูก มีค่า 2 ค่าคือ 0 และ 1 เป็น สมการจะเป็น $S = 1/T$ โดยที่ T เป็นเวลาต่อเนื่อง (Continue Time) ของ Pulse 1 $1/T$ จึงแสดงถึงจำนวนของ Pulse ใน 1 วินาที ซึ่งก็คือจำนวน Bit ที่สามารถไปได้ใน 1 วินาทีนั่นเอง

2.Modulation Speed

Modulation Speed หรือ Baud rate ก็คือจำนวนครั้งที่สามารถ Modulate ได้ใน 1 วินาที วิธีการ Modulation นั้นมี Amplitude Mod , Frequency Mod , Phase Mod และอื่น ๆ ในกรณีที่ T แสดงถึงช่วงเวลาในการเปลี่ยนแปลงในสภาวะนั้น จะสามารถกำหนดได้ดังสมการข้างล่างนี้โดยหน่วยจะเป็น Baud

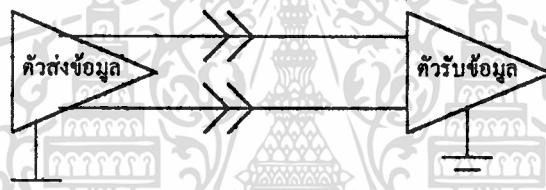
B : $1/T$

B : Modulation Speed (Baud)

T : เวลาในการเปลี่ยนจากสภาวะหนึ่งเป็นอีกสภาวะ

การสื่อสารข้อมูลแบบอนุกรมตามมาตรฐาน RS-232 C

การสื่อสารข้อมูลแบบอนุกรมที่ใช้งานอยู่ในปัจจุบันนั้น ได้มีการกำหนดมาตรฐานการรับส่งข้อมูลไว้หลายแบบด้วยกัน แต่ที่ได้รับความนิยมนำมาใช้งานอย่างมาก คือ การสื่อสารข้อมูลแบบอนุกรมตามมาตรฐาน RS-232 C และที่มาตรฐานเป็นที่นิยมเนื่องจากเป็น ระบบการสื่อสารข้อมูลที่ใช้ในเครื่องไมโครคอมพิวเตอร์ IBM PC ซึ่งเป็นคอมพิวเตอร์ที่ใช้กันอย่างแพร่หลายมากจากอดีตจนถึงปัจจุบันมาตรฐานการสื่อสารนี้ในการออกแบบเบื้องต้นได้ออกแบบ สำหรับการเชื่อมต่อกับเครื่องโมเด็ม (MODEM) ซึ่งเป็นอุปกรณ์ที่ใช้ในการสื่อสารข้อมูลระหว่างคอมพิวเตอร์ผ่านทางสายโทรศัพท์ ซึ่งทำให้อัตราการรับส่งข้อมูลถูกจำกัดให้มีค่าที่ค่อนข้างต่ำ มาตรฐาน RS-232C นี้ได้ออกแบบให้มีโครงสร้างการสื่อสารเป็นแบบจุดต่อจุดเท่านั้น ไฟฟ้าและทางกายภาพ ดังแสดงภาพ



ภาพที่ 24 แสดง โครงสร้างของการสื่อสารข้อมูลแบบอนุกรมมาตรฐาน RS-232C

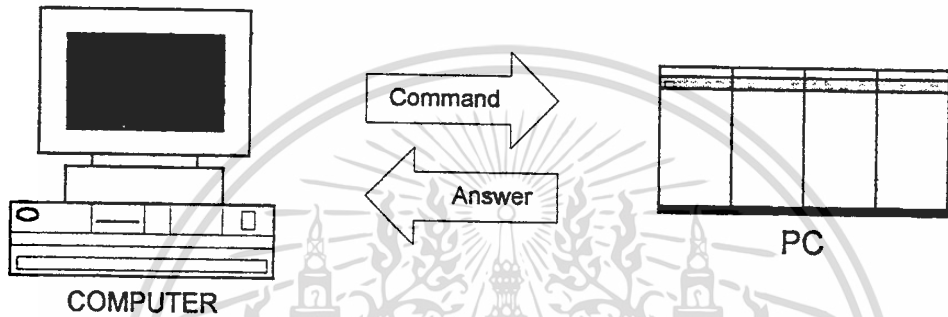
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขั้นตอนควบคุมการสื่อสาร

คำสั่งการสื่อสารและ Response

การสื่อสารของ PLC กับ Computer จะกระทำโดยการให้และรับสั่ง และ Response

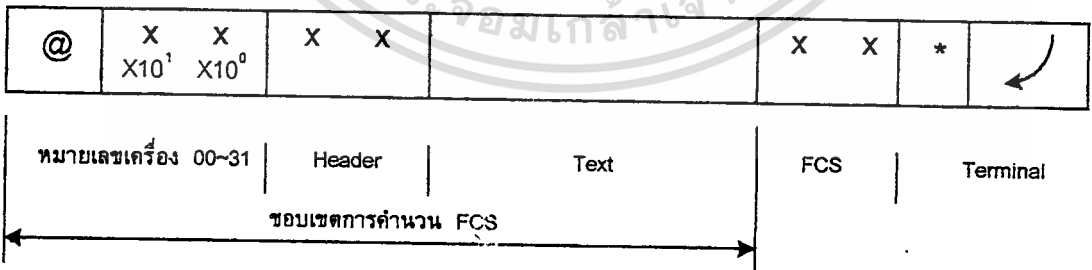
- กับ Command ก็คือ "Computer" "Instrument" ที่ Computer จะส่งไปที่ PLC
- กับ Response ก็คือ "Answer" ซึ่ง PC ที่ได้รับ Command คอบกลับไปที่ Computer



ภาพที่ 25 การติดต่อแบบจุดต่อจุด

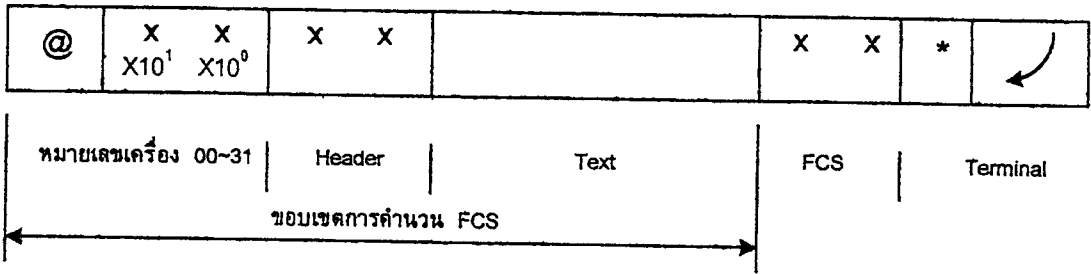
ขั้นตอน Format (1:n) ของ Command Block

ขั้นตอนควบคุมการสั่ง จะเป็นการติดต่อ (สนทนา) ด้วยขั้นตอนและการส่งสัญญาณเริ่มต้น จะอยู่ที่ด้าน Computer จะส่งออกไป



- Block Format (ขั้นตอน 1 : n)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



1.block ของข้อมูลที่จะสื่อสาร จะเริ่มต้นด้วยหมายเลขเครื่องและ Header และสิ้นสุดด้วย FCS กับ terminator

หมายเลขเครื่อง

หมายเลขเครื่อง (00-31ฐาน 10) ที่จะส่ง Command ของ High rank link Unit

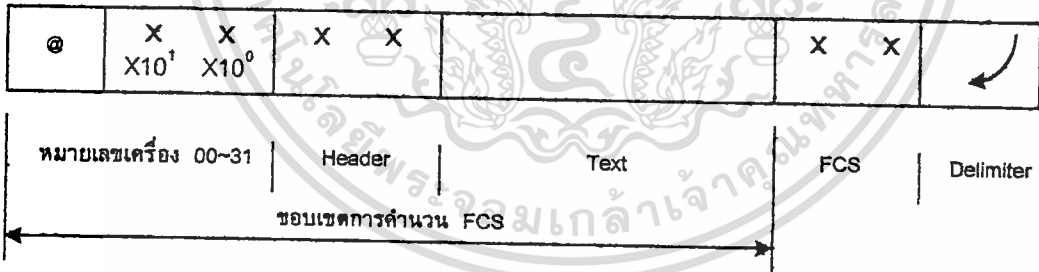
Header Code

กำหนดจากรายละเอียดของ Command Block Format ที่มากกว่า 128 Character

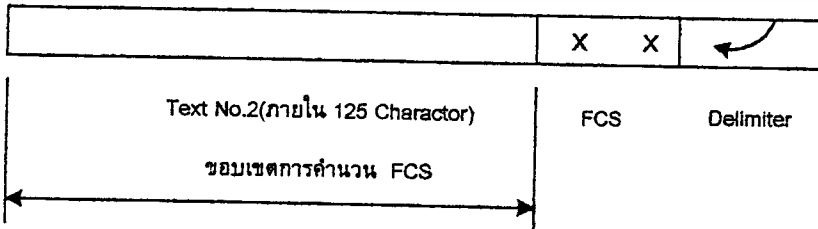
(ขั้นตอน 1:n)

- Block Format ที่มากกว่า 128 Character (ขั้นตอน 1:n)

Head Frame (ภายใน 131 Character)

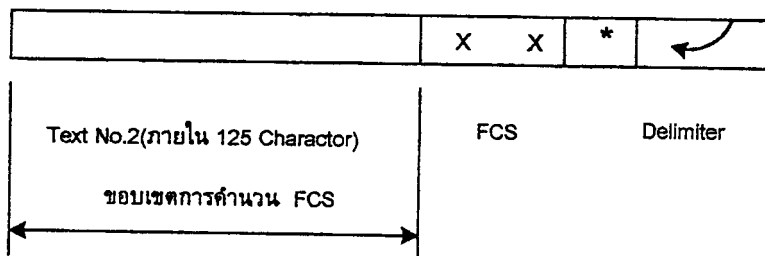


Medium Frame (ภายใน 128 Character)



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Terminate Frame (ภายใน 128 Character)



- ข้อควรระวังขณะแบ่ง Frame

ในกรณีของการส่ง Command ของระบบการเขียน เช่น WR,WL,WH,WD ที่ส่งหลาย ๆ Frame ข้อมูลของ HC อันเดียวกัน จะไม่ทำการแบ่ง Frame หลาย ๆ Frame ออกจากกัน การตัด Frame จะทำให้เป็นการตัด CH Data (ข้อมูล CH)

(1) Frame .Check . Sequence

FCS เป็นสิ่งที่ใช้สำหรับการตรวจเช็คความผิดพลาดของการสื่อสาร เป็นสิ่งที่เปลี่ยนข้อมูล 8 bit เป็น 2 ตัวอักษรข้อมูล ASCII ที่ได้ทั้ง Exclusive OR จาก @ ของหมายเลขเครื่องจนกระทั่งถึงตัวอักษรตัวสุดท้าย Text

รูปแบบของ FCA Code

ตัวอย่าง

@ 0100 0000

@ 10RH00310001

←

XOR

1 0011 0001

←

← ขอบเขตการคำนวณของ FCS ผลการคำนวณ 0111 0001

XOR

0 0011 0000

ผลการคำนวณ 0100 0001

XOR

R 0101 0010

XOR

1 0011 0001

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Response Format...Code เสร็จสิ้นการ Response ขณะปฏิบัติงานปกติจะเป็น " 00 "

@	หมายเลขเครื่อง $X10^1 \times 10^0$	R	R	End Code	Data $x16^3x16^2x16^1x16^0$	Data $x16^3x16^2x16^1x16^0$	FCS	*	↙
---	---------------------------------------	---	---	----------	--------------------------------	--------------------------------	-----	---	---

Header Code

รายละเอียดเริ่ม CH ค้นอ่าน

Terminal

รายละเอียดเริ่ม CH ค้นอ่าน

2)พื้นที่การเขียนของ Input Output Relay / Internal Assistance Relay

- การเขียนข้อมูลที่กำหนดจาก CH ระบุพื้นที่ Input Output Relay , Internal Assistance Relay (การเขียนของ Ch Unit

Command Format

@	หมายเลขเครื่อง $X10^1 \times 10^0$	W	R	CH $x10^3x10^2 \times 10^1x10^0$	จำนวน CH ที่เขียน $x16^3x16^2x16^1x16^0$	จำนวน CH ที่เขียน $x16^3x16^2x16^1x16^0$	FCS	*	↙
---	---------------------------------------	---	---	-------------------------------------	---	---	-----	---	---

Header Code

Terminal
จำนวนข้อมูลที่จะเขียนไปที่ CH เริ่มต้นเขียน
จำนวนข้อมูลที่จะเขียนไปที่ เริ่มต้นการเขียน
CH เริ่มต้นการเขียน : ขอบเขต 0000-0060

และแล้วให้เราทำการอ่านข้อมูลของพื้นที่ Input Output Relay ตามตัวอย่างปัญหาข้างล่าง

Response Format...Code เสร็จสิ้นการ Response ขณะปฏิบัติงานปกติจะเป็น " 00 "

@	หมายเลขเครื่อง $X10^1 \times 10^0$	W	R	End Code $x16^1 \times 16^0$	FCS	*	↙
---	---------------------------------------	---	---	---------------------------------	-----	---	---

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

(2) การอ่าน DM Area

- การอ่านส่วน CH ที่กำหนดจาก Ch ระบุพื้นที่ DM

Command Format

@	หมายเลขเครื่อง $X10^1 \times 10^0$	R	D	CH $\times 10^3 \times 10^2 \times 10^1 \times 10^0$	จำนวน CH ที่อ่าน $\times 10^3 \times 10^2 \times 10^1 \times 10^0$	FCS	*	↩
---	---------------------------------------	---	---	---	---	-----	---	---

Header Code

Terminal

จำนวน CH ที่อ่าน : ขอบเขต 0001~0064

CH เริ่มต้นอ่าน : ขอบเขต 0000~0063

Response Format...Code เสร็จสิ้นการ Response ขณะปฏิบัติงานปกติจะเป็น "00"

@	หมายเลขเครื่อง $X10^1 \times 10^0$	R	D	End Code	Data $\times 16^3 \times 16^2 \times 16^1 \times 16^0$	Data $\times 16^3 \times 16^2 \times 16^1 \times 16^0$	FCS	*	↩
---	---------------------------------------	---	---	----------	---	---	-----	---	---

Header Code

Terminal

รายละเอียดเริ่ม CH ต้นอ่าน

รายละเอียดเริ่ม CH ต้นอ่าน

การเขียน DM Area

- การเขียนข้อมูลที่กำหนด โดย CH Unit จาก CH ระบุพื้นที่ DM

Command Format

@	หมายเลขเครื่อง $X10^1 \times 10^0$	W	D	CH $\times 10^3 \times 10^2 \times 10^1 \times 10^0$	Data ที่จะเขียน $\times 16^3 \times 16^2 \times 16^1 \times 16^0$	Data ที่จะเขียน $\times 16^3 \times 16^2 \times 16^1 \times 16^0$	FCS	*	↩
---	---------------------------------------	---	---	---	--	--	-----	---	---

Header Code

Terminal

จำนวนข้อมูลที่จะเขียนไปที่ CH เริ่มต้นเขียน

จำนวนข้อมูลที่จะเขียนไปที่ เริ่มต้นการเขียน

CH เริ่มต้นการเขียน : ขอบเขต 0000-0060

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่สามารถเผยแพร่หรือใช้
 "ไม่ว่ากรณีใดๆ ทั้งสิ้น" อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่ใช้

Response Format...Code เสร็จสิ้นการ Response ขณะปฏิบัติงานปกติจะเป็น " 00 "

@	หมายเลขเครื่อง $X10^1 \quad X10^0$	W	D	End Code $X16^1 X16^0$	FCS	*	↙
---	---------------------------------------	---	---	---------------------------	-----	---	---

(3) การอ่าน Status

- การอ่านสถานะการทำงานของตัว SYSMAC

Command Format

@	หมายเลขเครื่อง	M	S	FCS	*	↙
---	----------------	---	---	-----	---	---

Header Code

Terminal

Response Format...Code เสร็จสิ้นการ Response ขณะปฏิบัติงานปกติจะเป็น " 00 "

@	หมายเลขเครื่อง	R	D	End Code	Status Data	Message	FCS	*	↙
---	----------------	---	---	----------	-------------	---------	-----	---	---

Header Code

End Code

Terminal

15	14	13	12	11	10	9	8
----	----	----	----	----	----	---	---



0 0 ... Program Mode

0 1 ... Debug Mode

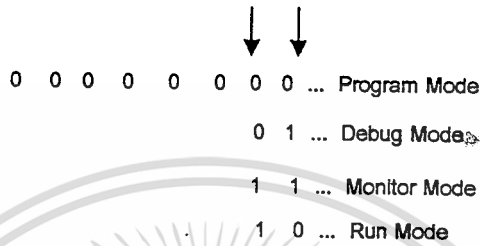
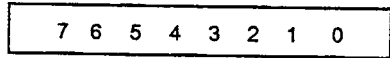
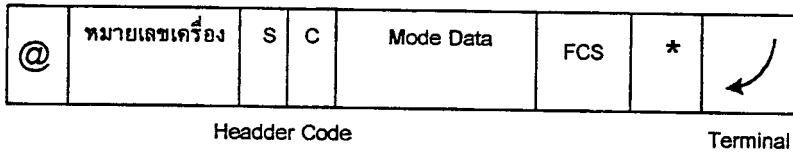
1 0 ... Run Mode

1 1 ... Monitor Mode

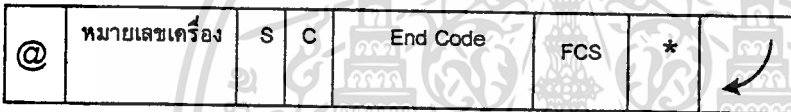
(4) การเขียน Satus

- การเปลี่ยน Mode ของ SYSMAC

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



Response Format...Code เสร็จสิ้นการ Response ขณะปฏิบัติงานปกติจะเป็น " 00 "



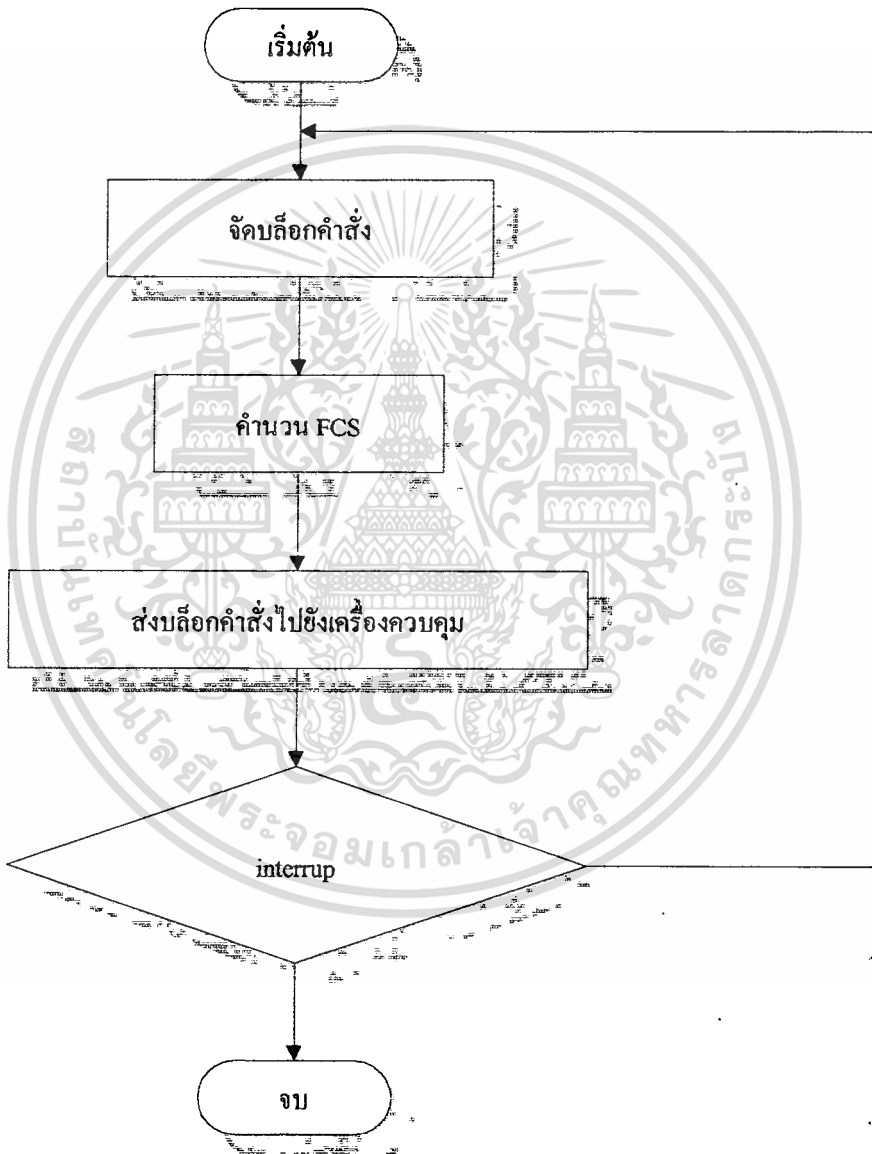
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3

การออกแบบโปรแกรม

การออกแบบโปรแกรมสำหรับการถ่ายเทข้อมูล ระหว่างคอมพิวเตอร์ไปยังเครื่องควบคุมPLC

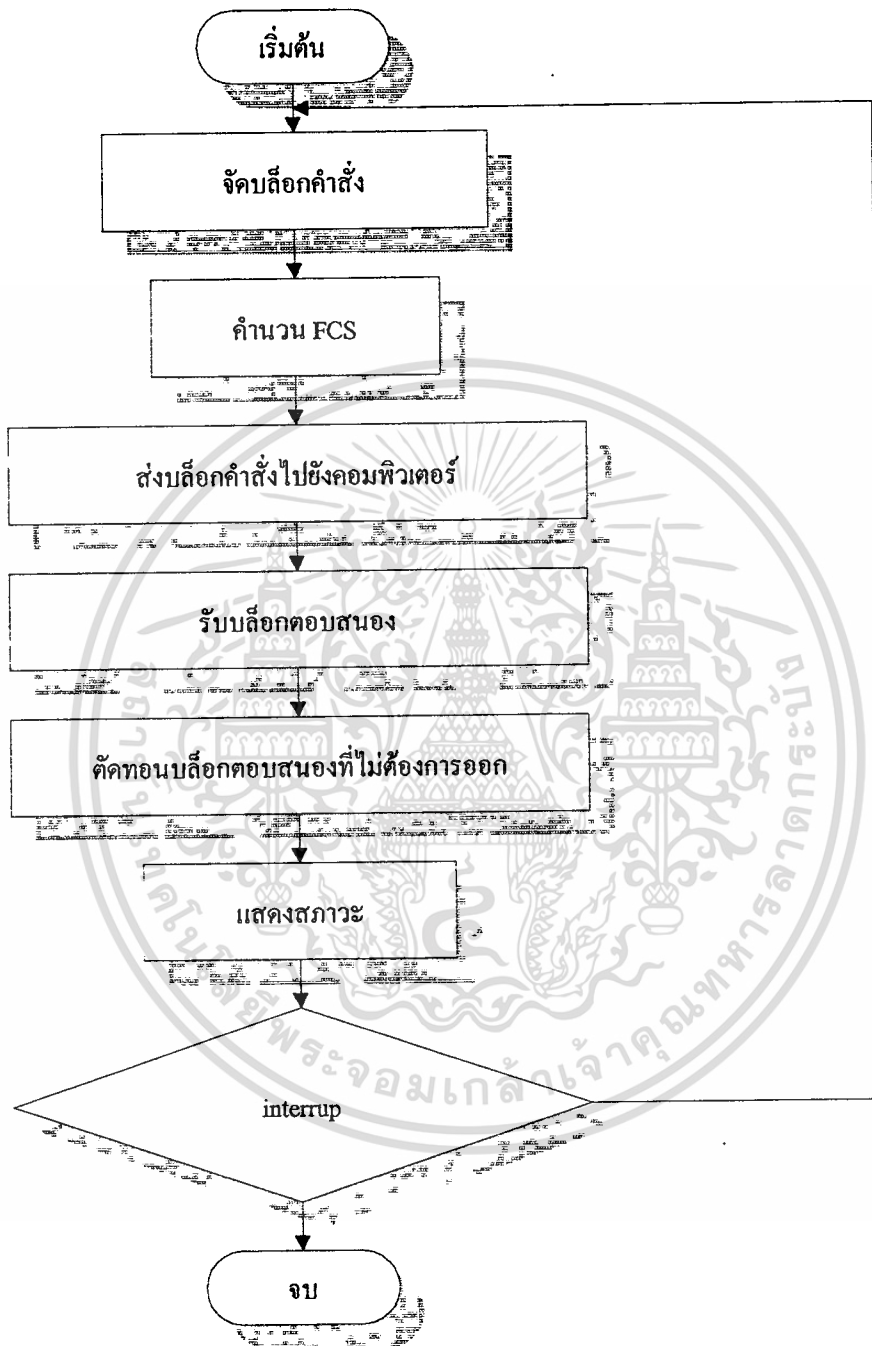
ในการออกแบบโปรแกรมส่วนนี้ได้จัดทำเป็นบล็อกคำสั่งซึ่งพอจะสรุปเป็นผังการทำงานได้ดังนี้



ภาพที่ 26 แสดงFlow Chart การถ่ายเทข้อมูลจากคอมพิวเตอร์ไปยังเครื่องควบคุมPLC

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

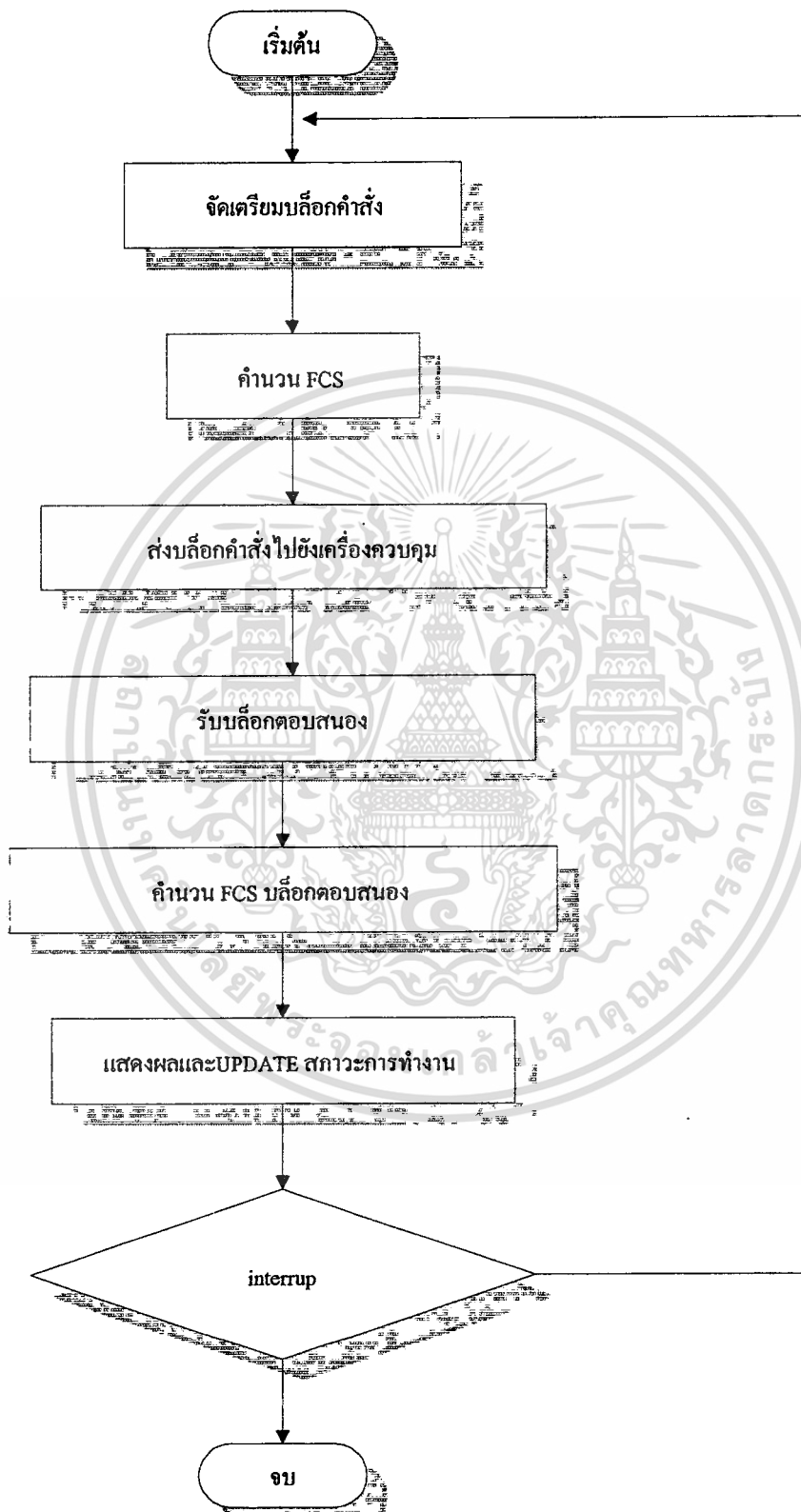
การออกแบบโปรแกรมสำหรับการถ่ายเทข้อมูลจากเครื่องควบคุม PLC ไปยังคอมพิวเตอร์



ภาพที่ 27 แสดง Flow Chart การถ่ายเทข้อมูลจากเครื่องควบคุม PLC ไปยังคอมพิวเตอร์

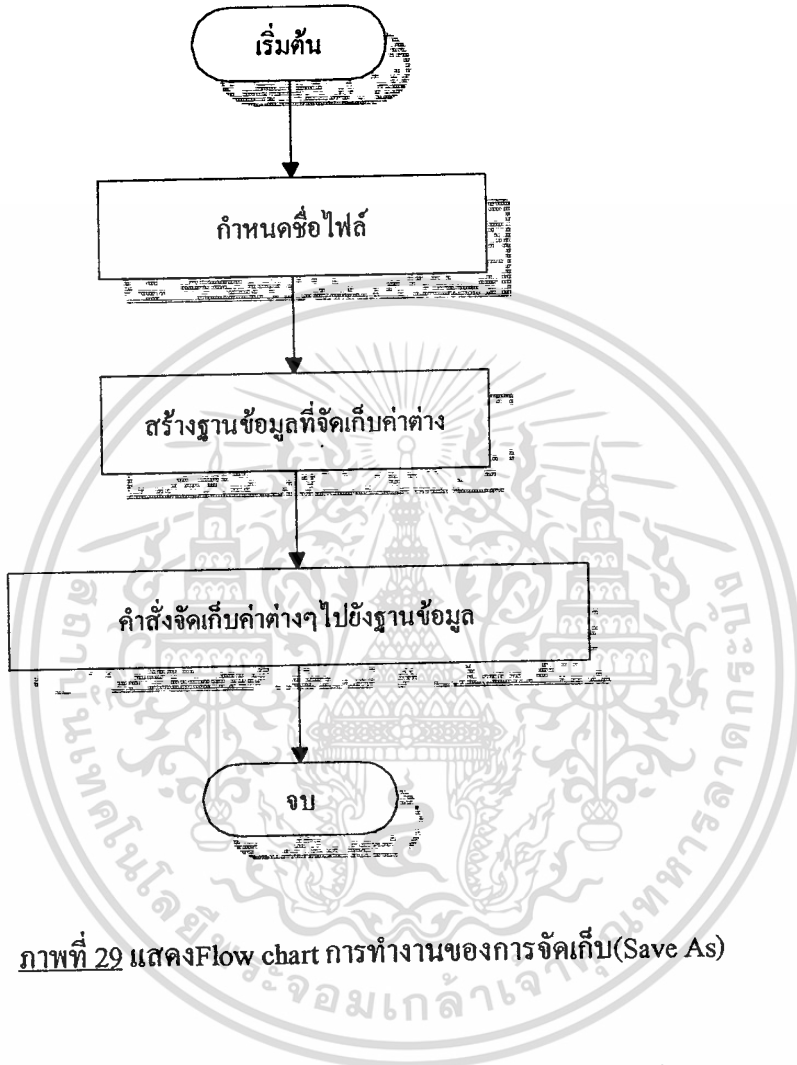
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การออกแบบโปรแกรมแสดงผลการทำงานของเครื่องควบคุม PLC(DISPLAY)



เอกสารนี้เป็นเอกสารสงวนลิขสิทธิ์การใช้งานโดยไม่อนุญาตให้เผยแพร่โดยไม่ได้รับอนุญาตจากเจ้าของเอกสาร
ภาพที่ 28 แสดง Flow chart การทำงานของการแสดงสถานะ(DISPLAY)
 ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

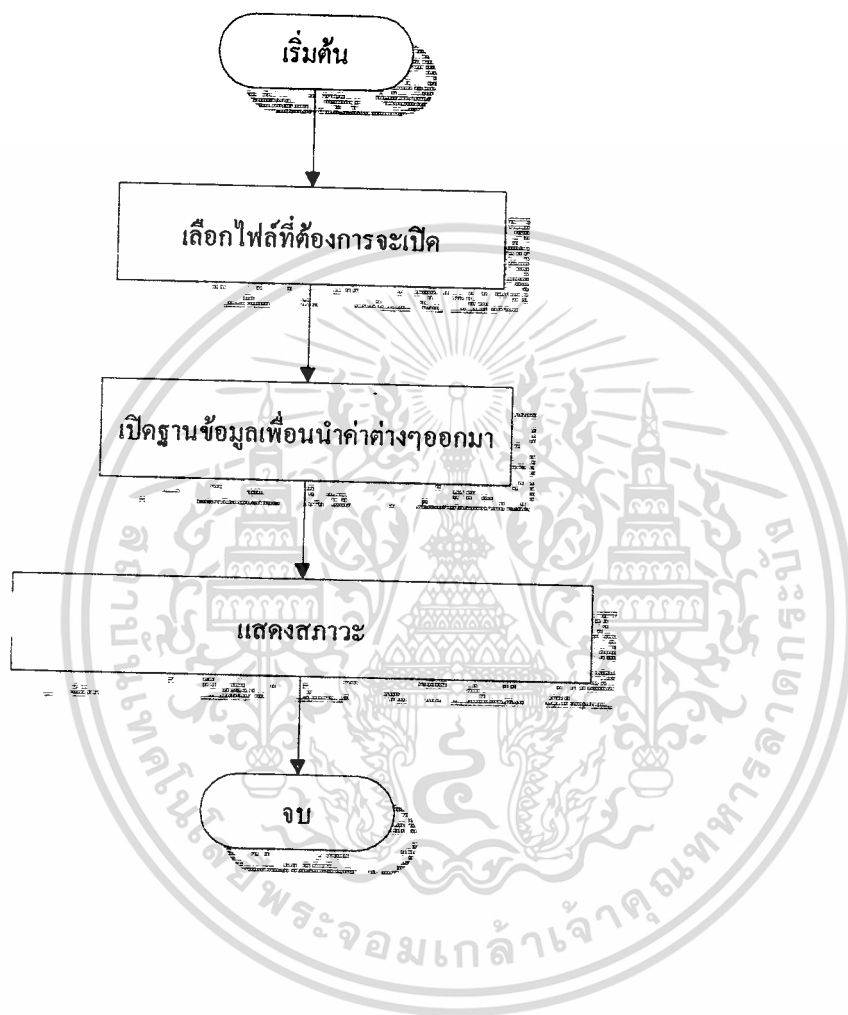
การออกแบบ โปรแกรมการทำงานของ การจัดเก็บ(Save As)



ภาพที่ 29 แสดง Flow chart การทำงานของ การจัดเก็บ(Save As)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การออกแบบ โปรแกรมการทำงานของ การเปิดไฟล์(Open)



ภาพที่ 30 แสดง Flow chart การทำงานของการเปิดไฟล์(Open)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตัวอย่างการเขียนโปรแกรมการสื่อสารข้อมูลระหว่างคอมพิวเตอร์ กับ เครื่องควบคุม PLC

```
Public Function testword(w As String, B As Integer)
```

```
    If w = "CH000" Then
```

```
        kw = ch00
```

```
        dat1 = "0000"
```

```
        a = testbit(kw, bbit)
```

```
    End If
```

```
End Function
```

```
Public Sub interfaceA(text As String)
```

```
    If newpj.MSComm1.PortOpen = True Then
```

```
        newpj.MSComm1.PortOpen = False
```

```
    End If
```

```
    If Not newpj.MSComm2.PortOpen Then
```

```
        newpj.MSComm2.PortOpen = True
```

```
    End If
```

```
    newpj.MSComm2.Output = text + Chr$(13) + Chr$(10)
```

```
    newpj.MSComm2.PortOpen = False
```

```
End Sub
```

```
Public Function inspectword(w1 As String, w2 As String, w3 As String, w4 As String)
```

```
    wordDEC (w1)
```

```
    tb(0) = a(0)
```

```
    tb(1) = a(1)
```

```
    tb(2) = a(1)
```

```
    tb(3) = a(3)
```

```
End Function
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตัวอย่างการเขียนโปรแกรมการจัดเก็บข้อมูล (SAVE AS)

```

Private Sub savep_Click()

    main.CommonDialog1.Filter = "(*.plc)|*.plc|"
    main.CommonDialog1.DialogTitle = "save file"
    main.CommonDialog1.Action = 2
    filenamex = main.CommonDialog1.filename
    If filenamex = "" Then
        Exit Sub
    End If
    myfile = Dir(filenamex)
    If myfile <> "" And vbOK = 1 Then
        Kill filenamex
    End If
    createexam (filenamex)
    Set ws = Workspaces(0)
    Set db = ws.OpenDatabase(filenamex)
    Set rw = db.OpenRecordset("equipment")
    If newpj.tank1(0).Visible = True Then
        rw.AddNew
        rw.Fields("index") = 0
        rw.Fields("nameid") = newpj.tank1(0).Name
        rw.Fields("visiblex") = newpj.tank1(0).Visible
        rw.Fields("topy") = newpj.tank1(0).Top
        rw.Fields("leftx") = newpj.tank1(0).Left
        rw.Fields("channel") = Tank_1
        rw.Fields("bit") = 0
        rw.Update
    End If

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตัวอย่างการเขียน โปรแกรมการเปิดแฟ้มข้อมูล (OPEN)

```
Private Sub openp_Click()
    CommonDialog1.Filter = "(*.plc)|*.plc|"
    CommonDialog1.DialogTitle = "open file"
    CommonDialog1.Action = 1
    filenamex = CommonDialog1.filename

    If filenamex = "" Then
        Exit Sub
    End If

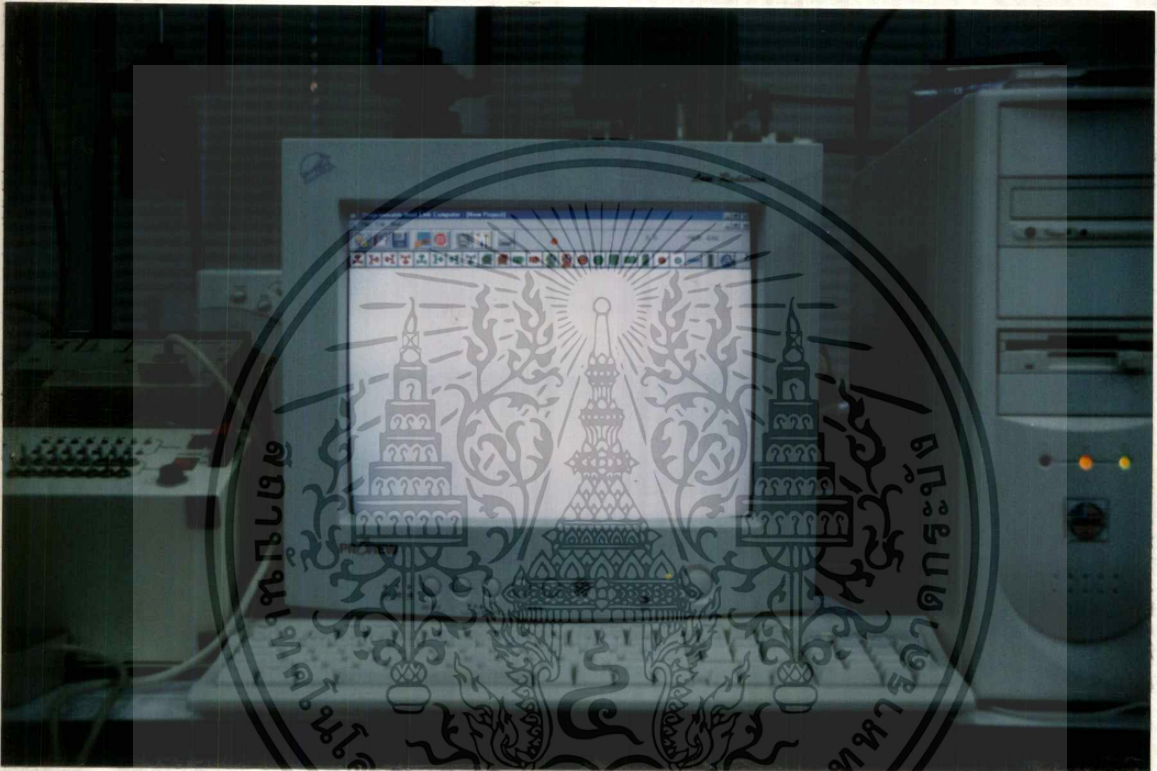
    sav1 = False 'save variable
    myfile = Dir(filenamex)
    If myfile = "" Then Exit Sub
    Set ws = Workspaces(0)
    Set db = ws.OpenDatabase(filenamex)
    Set rw = db.OpenRecordset("equipment", dbOpenDynaset)
    Set Data1.Recordset = rw
    Data1.Recordset.MoveLast
    s = Data1.Recordset.AbsolutePosition
    Data1.Recordset.MoveFirst
    Do Until v > s
    If v = -1 Then GoTo line1
    Select Case Text3.text
        Case "tank1"
            newpj!tank1(main.Text2.text).Visible = Text4.text
            newpj!tank1(main.Text2.text).Top = Text5.text
            newpj!tank1(main.Text2.text).Left = Text6.text
            Tank_1 = Text8.text
            LabelsetGreen Tank_1
            Data1.Recordset.MoveNext
            v = Data1.Recordset.AbsolutePosition
    End Select
    End Do
    line1:

```

บทที่ 4

รูปแบบของโครงการ และผลการทดลองการทำงาน

ในบทนี้เสนอรูปแบบของโครงการและผลการทดสอบการทำงานของโปรแกรมสำหรับออกแบบการควบคุม และแสดงสภาวะการทำงานของเครื่องควบคุมPLC



ภาพที่ 31 แสดงรูปแบบการทำงาน

หน้าที่ของปุ่มคำสั่งในโปรแกรม

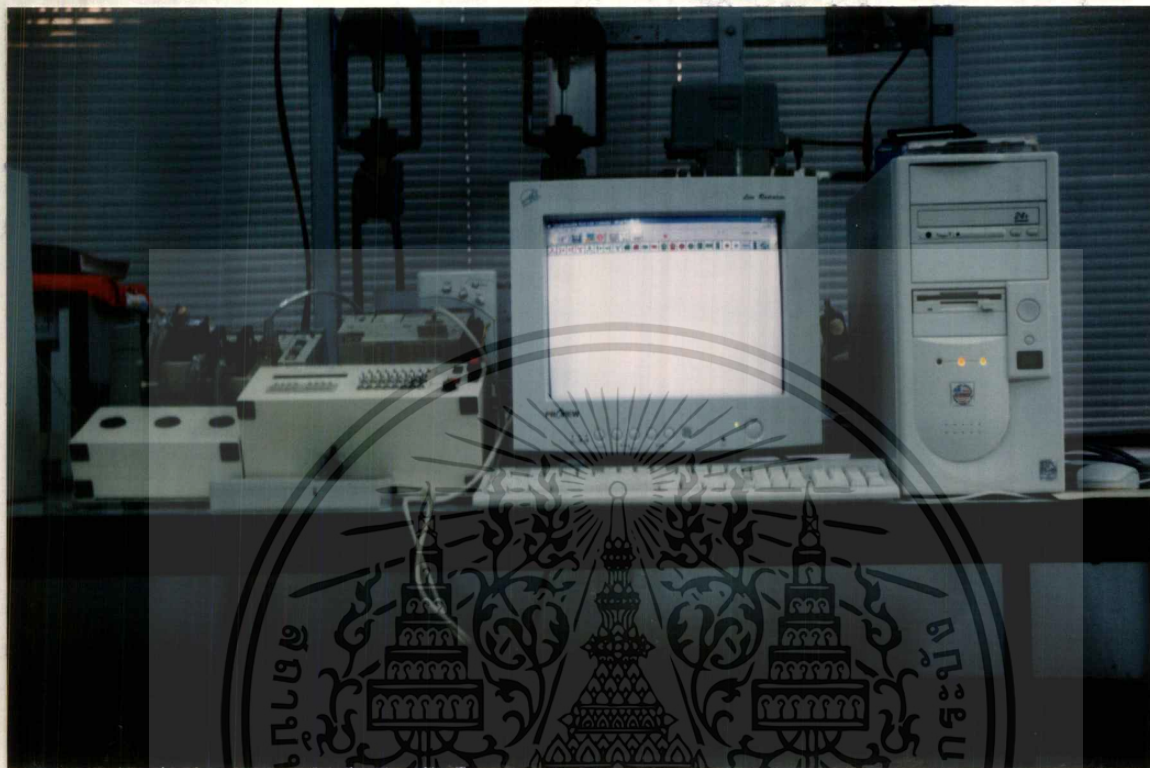
1. NEW การเรียกฟอร์มสำหรับออกแบบการควบคุม
2. OPEN การเปิดไฟล์จากหน่วยความจำ
3. SAVE การบันทึกไฟล์ลงหน่วยความจำ
4. RUN การเข้าสู่กระบวนการทำงานกับ PLC
5. STOP การออกจากกระบวนการทำงานกับPLC
6. DISPLAY การแสดงสภาวะการทำงานของPLC

7. TOOLBOX เครื่องมือที่ใช้ในการออกแบบการควบคุม

8. PICTURE การเปิดไฟล์ของรูปภาพที่จะนำมาใช้เป็นBACKGROUND

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับงานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
สงวนลิขสิทธิ์ พงษ์สิทธิ์พิบูลย์พิมพ์ดีดแบบลงเนื้อที่และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขั้นตอนการทดลอง

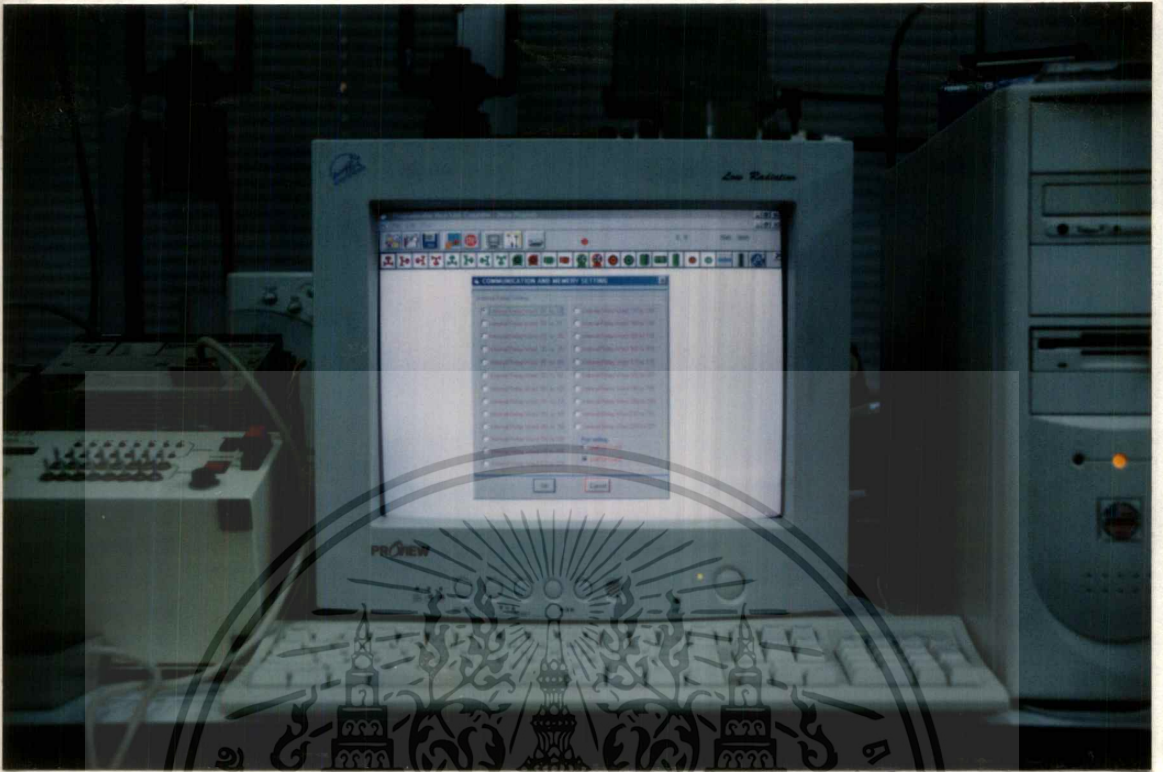


ภาพที่ 32 แสดงการทดลองสื่อสารข้อมูลระหว่างคอมพิวเตอร์กับเครื่องควบคุม PLC

ขั้นตอนที่ 1

เมื่อเริ่มเข้าสู่โปรแกรมจะต้องทำการเซต memory ของ PLC และเลือกพอร์ตใช้งานในการสื่อสารข้อมูลระหว่างคอมพิวเตอร์กับเครื่องควบคุม PLC

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นอนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

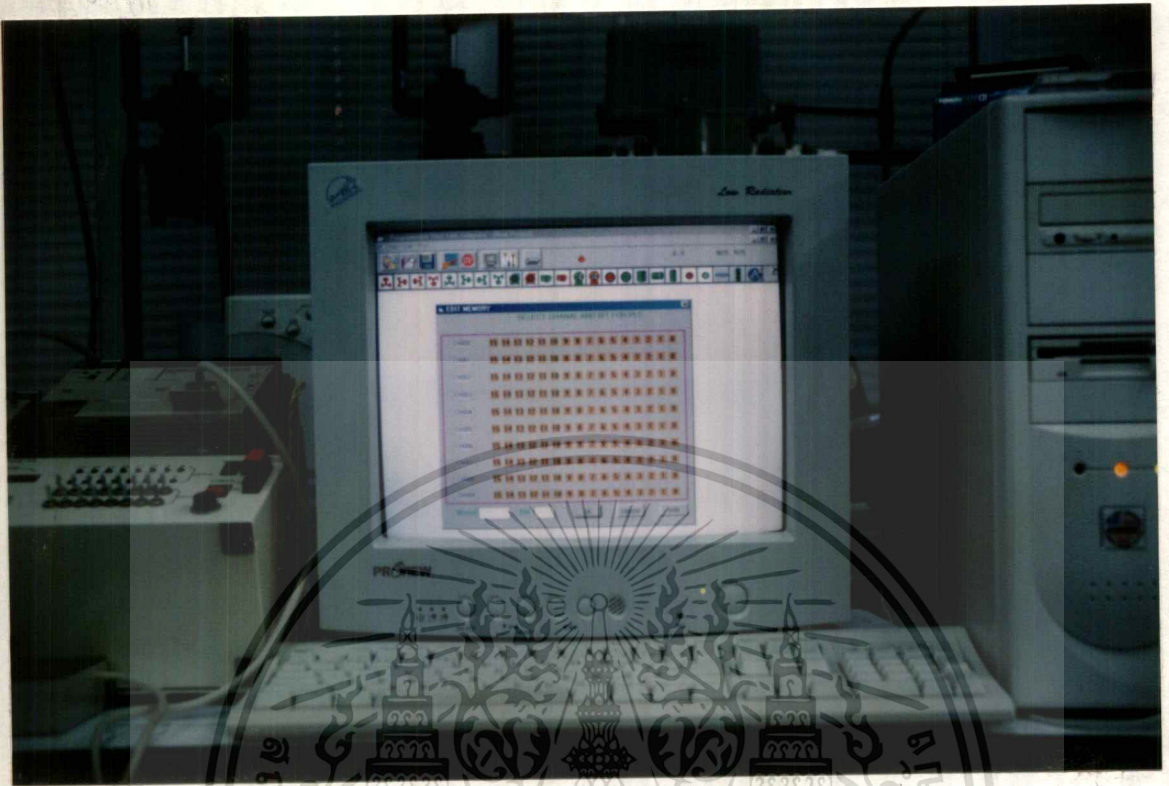


ภาพที่ 33 แสดงการเลือกใช้นหน่วยความจำและพอร์ทที่ใช้งาน

ขั้นตอนที่ 2

เมื่อเริ่มทำการออกแบบ จะนำเมาท์ไปคลิกบนอุปกรณ์ต่างๆบนทูลบาร์ที่ต้องการและลากมาวางไว้บนตำแหน่งที่ต้องการแล้วจะปรากฏหน้าต่างEDITMEMORY เพื่อทำการกำหนดChannel และBitให้กับอุปกรณ์นั้นแต่ถ้าอุปกรณ์เป็นTankและ Bargraph จะทำการกำหนดเป็น Channel อย่างเดียว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

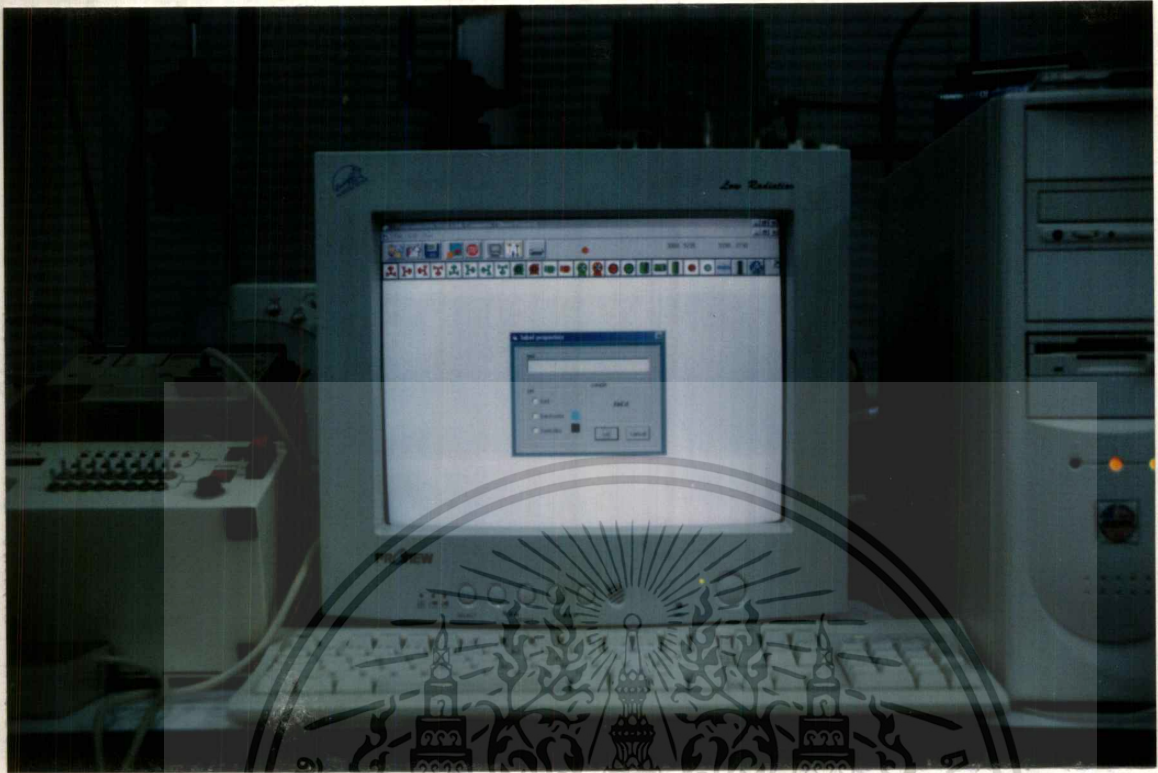


ภาพที่ 34 แสดงหน้าต่าง EDIT MEMORY

ขั้นตอนที่ 3

เมื่อทำการออกแบบอุปกรณ์ต่างๆตามรูปแบบที่ต้องการเสร็จเรียบร้อยแล้วก็อาจทำการกำหนด Label Properties เพื่อทำการอธิบายการทำงานต่างๆของอุปกรณ์ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

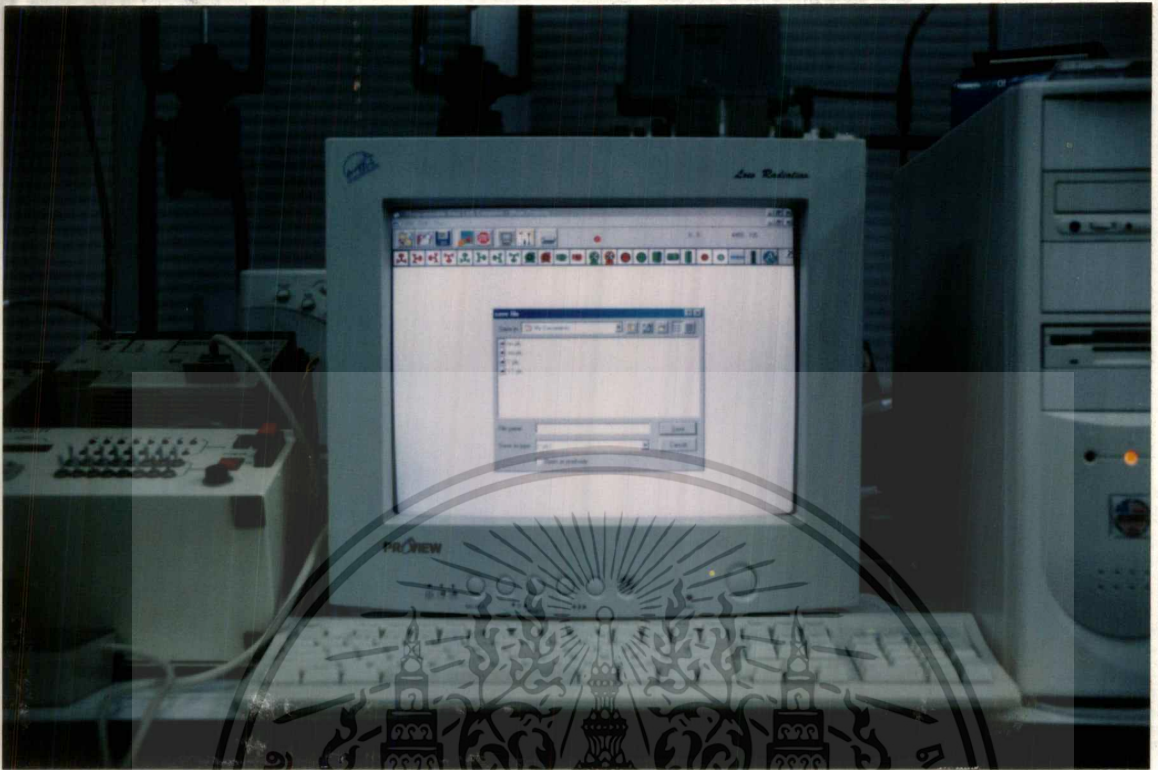


ภาพที่ 35 แสดงหน้าต่าง Label Properties

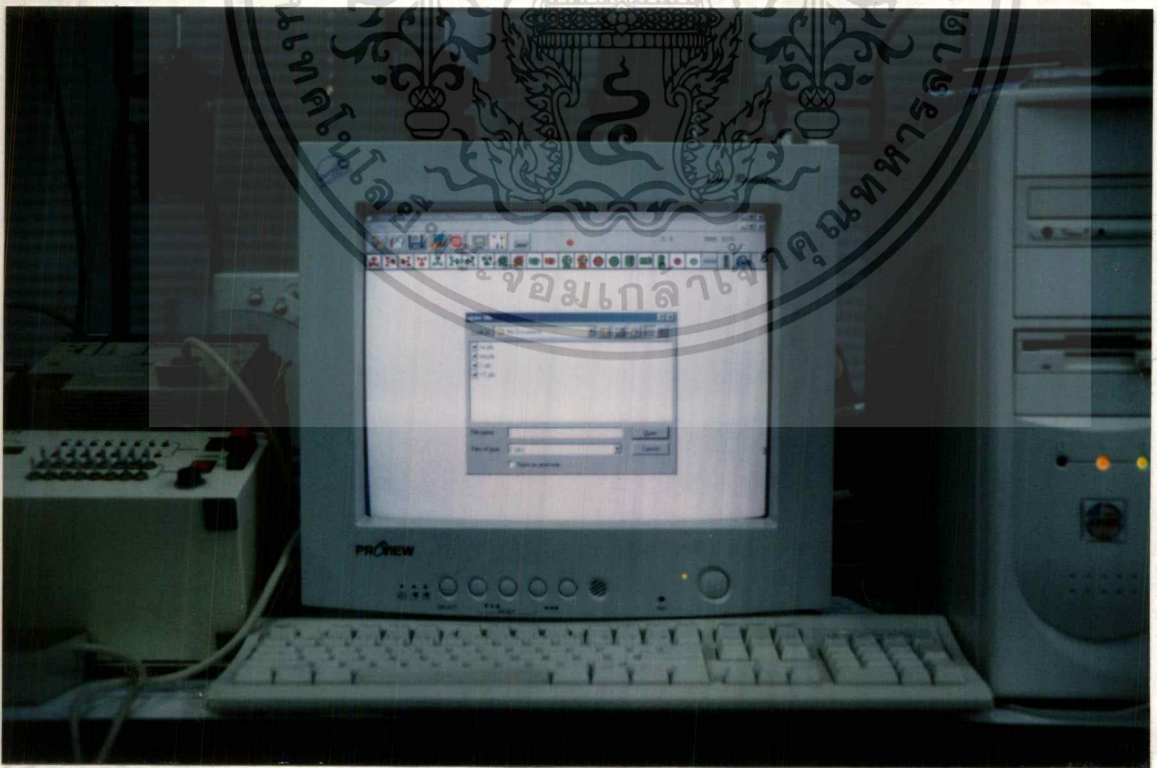
ขั้นตอนที่ 4

เมื่อทำการออกแบบต่างๆตามที่ต้องการเสร็จเรียบร้อยแล้วอาจจะมีการSAVE โปรแกรมที่เราออกแบบไว้เป็นไฟล์เก็บไว้ในหน่วยความจำของคอมพิวเตอร์ (*.plc) หรือโปรแกรมสามารถทำการเปิดแฟ้มที่เราทำการsave ไว้ออกมาได้เช่นเดียวกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 36 แสดงหน้าต่างการ Save As

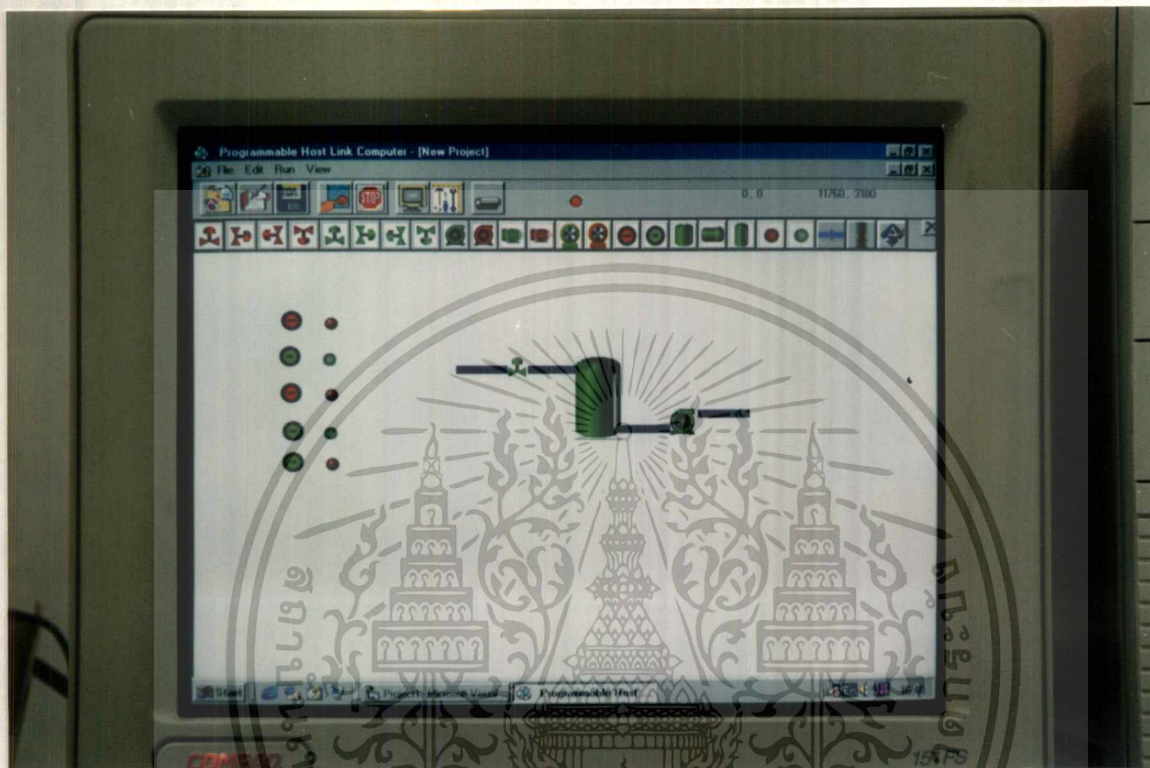


ภาพที่ 37 แสดงหน้าต่างการ Open

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับครูใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมีเหตุดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขั้นตอนที่ 5

เมื่อทำการออกแบบอุปกรณ์ต่างๆตามรูปแบบที่ต้องการเสร็จเรียบร้อยแล้วทำการ RUN โปรแกรมกับเครื่องควบคุม PLC

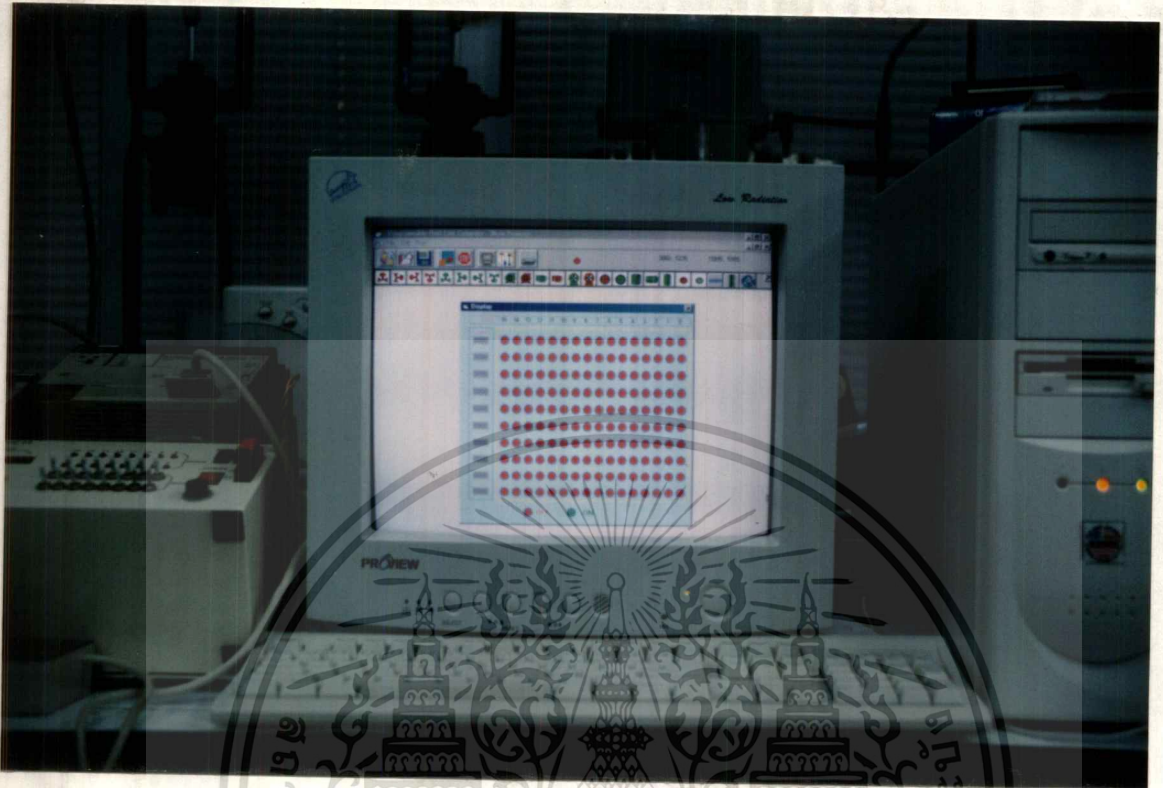


ภาพที่ 38 โปรแกรมที่ออกแบบ

ขั้นตอนที่ 6

โปรแกรมที่ RUN อยู่โปรแกรมสามารถเรียกหน้าต่าง DISPLAY มาตรวจสอบสถานะการทำงานของ PLC ที่เป็นอยู่ในสถานะปัจจุบันได้ และสามารถสั่งการทำงานทางหน้าจอคอมพิวเตอร์ โดยการนำเมาท์ไปคลิกที่ตัวอุปกรณ์ หรือสั่งการทำงานทาง PLC

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



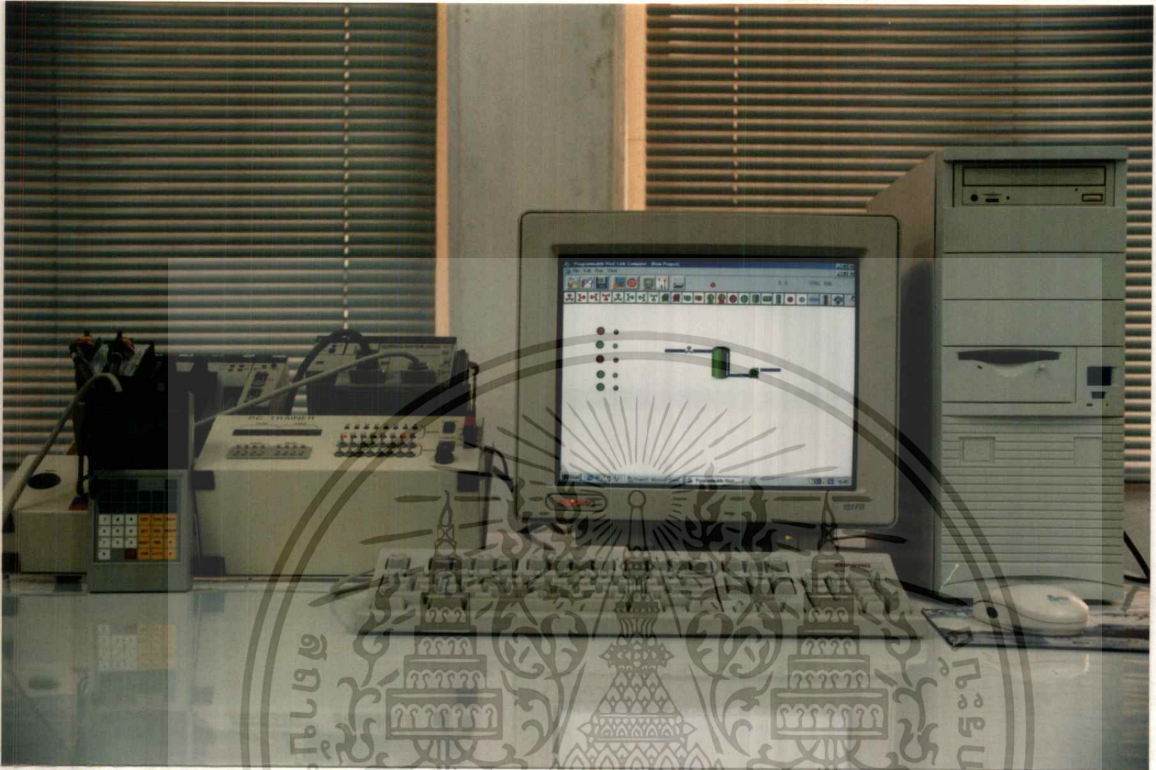
ภาพที่ 39 แสดงหน้าต่าง DISPLAY

ผลการทดลอง

หลังจากที่ทำการทดลองจะพบว่าเมื่อทำการตั้งงานพร้อมกัน ไม่สามารถทำการตั้งงานได้จะต้องทำการสื่อสารข้อมูลให้เสร็จภายในหนึ่งรอบการทำงานของแต่ละอุปกรณ์ ในส่วนของการออกแบบควบคุมพบว่าในการเลือกอุปกรณ์ที่จะนำมาออกแบบ รวมถึงการเดินอุปกรณ์แต่ละครั้ง ต้องคลิกถึง สองครั้งถึงจะทำงานเนื่องจากการเขียน โปรแกรมไม่สามารถรองรับการทำงานพร้อมๆกัน ได้ การสื่อสารข้อมูลระหว่าง PLC กับคอมพิวเตอร์ ถูกต้อง การตรวจสอบสถานะของ PLC สามารถที่จะตรวจสอบสถานะได้สองรูปแบบคือแสดงสถานะด้วยการเปลี่ยนแปลงรูปแบบของตัวอุปกรณ์เอง และตรวจสอบด้วยหน้าต่าง DISPLAY และสามารถจัดเก็บรูปแบบไว้เป็นไฟล์ได้ รวมถึงการเปิดไฟล์ที่ได้จัดเก็บไว้เพื่อนำออกมาใช้งานได้ตามวัตถุประสงค์ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทดลองที่ 1



ภาพที่ 40 การทดลองโปรแกรมที่ออกแบบทางหน้าจคอมพิวเตอร์กับPLC

ขั้นตอนการทำงานของโปรแกรมที่ออกแบบ

1. เมื่อมีการโยกสวิตซ์ทำให้ไฟติดเป็นการตั้งงานทางPLC
2. เมื่อมีการนำมัทไปคริกที่LAMP ไฟจะติด

การกำหนดอินพุท และเอาต์พุทเพื่อทำการควบคุมการทำงาน

อินพุทรีเลย์	อุปกรณ์ทางอินพุท
00000	SWITCH 1
00001	SWITCH 2
00002	SWITCH 3
00003	SWITCH 4
00004	SWITCH 5

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

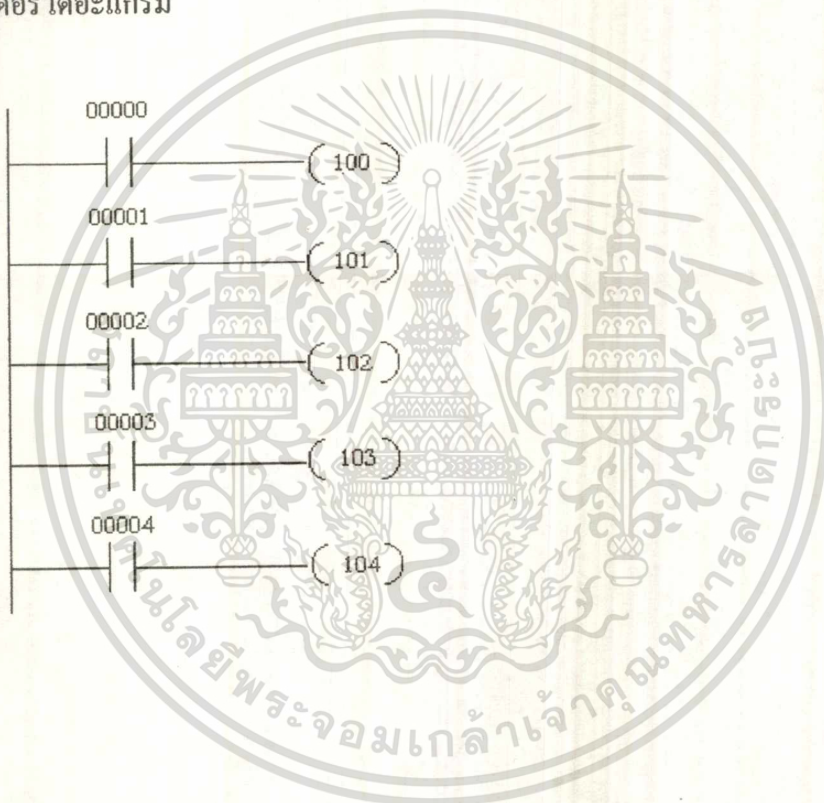
เข้าที่พุทรีเลข

100
101
102
103
104

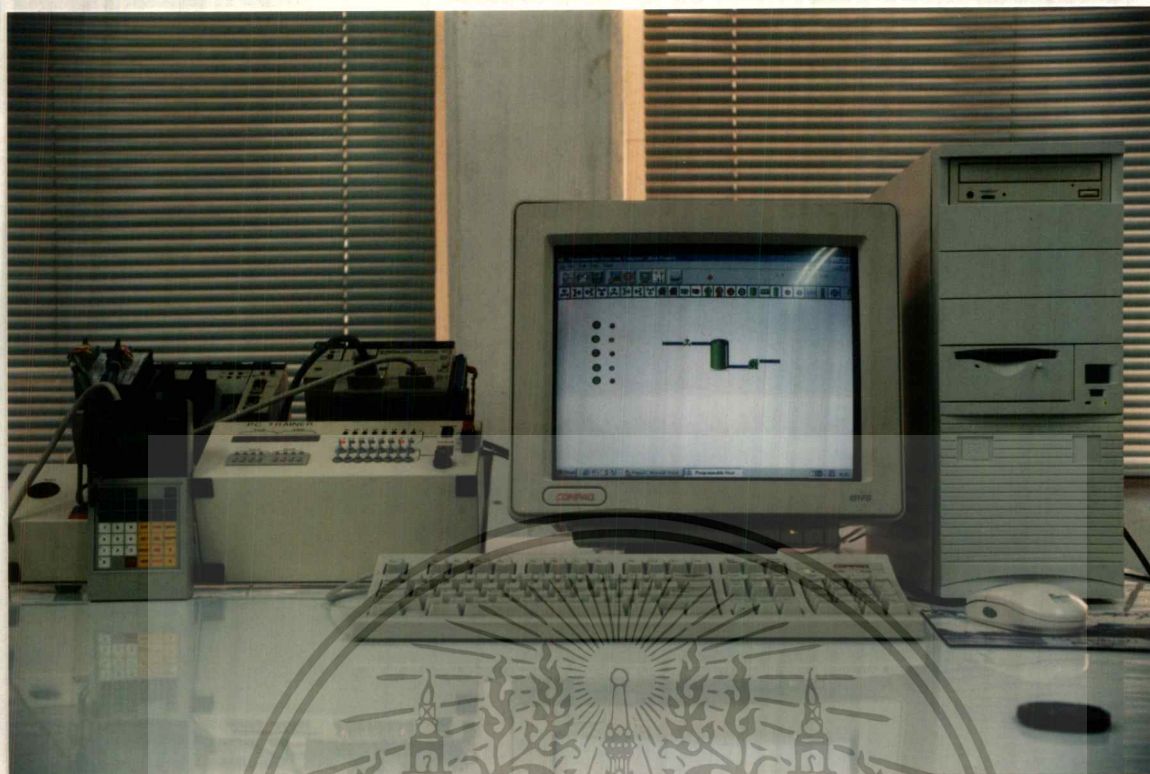
อุปกรณ์ทางเข้าที่พุท

LAMP 1
LAMP 2
LAMP 3
LAMP 4
LAMP 5

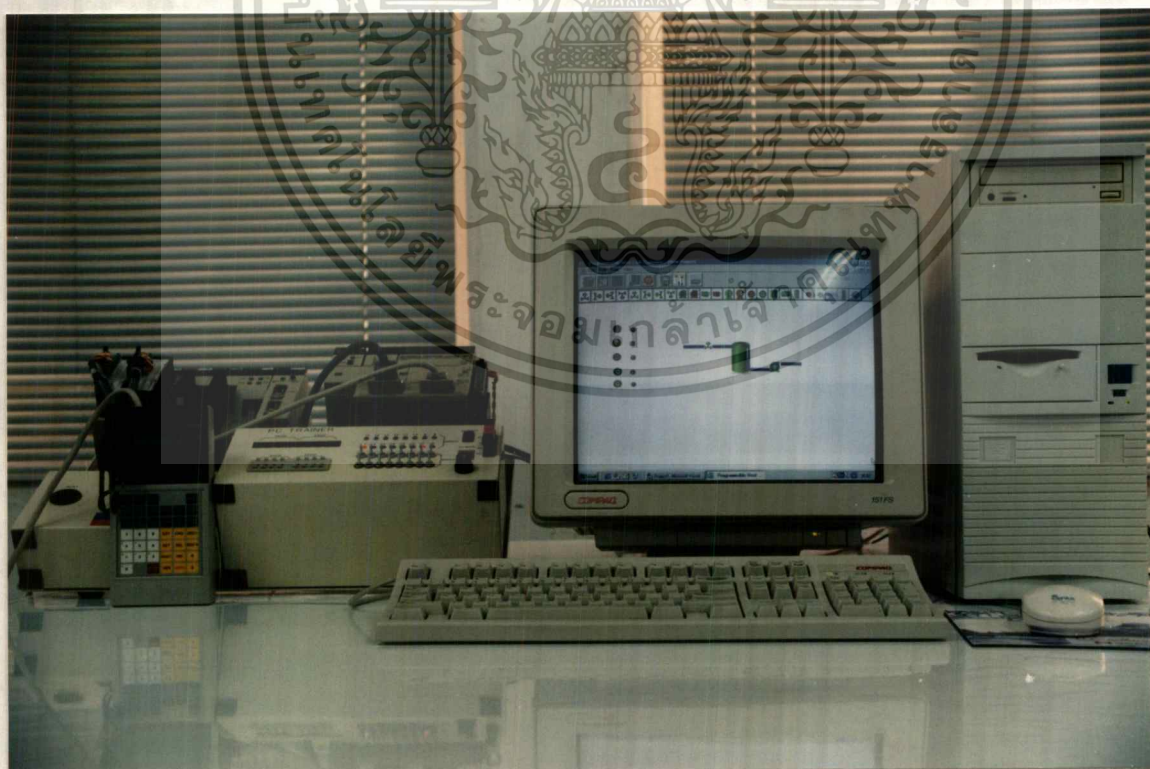
โปรแกรมแลดเดอร์ไดอะแกรม



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

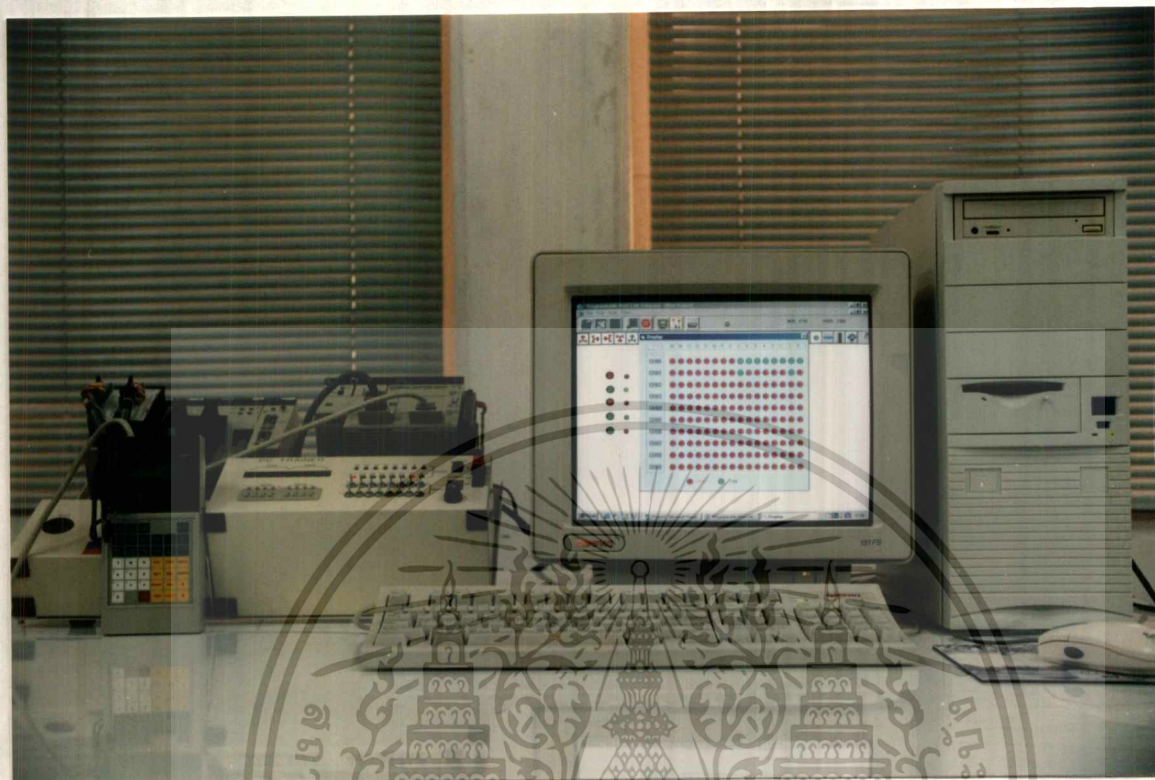


ภาพที่ 41 แสดงภาพการ RUN PROGRAM ที่ออกแบบระหว่างคอมพิวเตอร์ กับ PLC



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ หากมีข้อผิดพลาดประการใดขออภัยเป็นอย่างสูง ขอสงวนสิทธิ์ในสิ่งที่ปรากฏ

ภาพที่ 42 แสดงภาพสถานะการสั่งการทำงานที่หน้าจอคอมพิวเตอร์และที่ PLC



ภาพที่ 43 แสดงสถานะการทำงานทางหน้าต่าง Display

ผลการทดลอง

เมื่อทำการ RUN PROGRAM ได้ทดลองสั่งการทำงานทางหน้าจอคอมพิวเตอร์โดยการนำเมาส์ไปคลิกที่ตัวอุปกรณ์นั้น อุปกรณ์จะเปลี่ยนสถานะจาก OFF เป็น ON ทดลองและตรวจสอบค่าสถานะของ PLC ตรวจสอบเช็คค่า Display ของโปรแกรมว่าได้ตรงกับสถานะของ PLC กรณีของ Tank เรากำหนดค่าเป็น Channel สามารถแสดงสถานะการทำงานได้สิบหกระดับ ทำการกำหนดให้เป็น Channel 000 ทำการตั้งค่าทาง PLC ตรวจสอบค่าทางหน้าจอคอมพิวเตอร์สามารถแสดงผลตรงตามค่าที่ PLC สั่ง แต่อุปกรณ์ Tank ไม่สามารถสั่งสถานะการทำงานได้ทางหน้าจอคอมพิวเตอร์ แต่จะทำหน้าที่เป็น Display เท่านั้น นอกจากนี้ Bargraph จะมีสถานะการทำงานเหมือนกับ Tank และอุปกรณ์อื่นๆจะสถานะการทำงานคล้ายกัน คือ สามารถสั่งการทำงานได้ทั้งทางคอมพิวเตอร์และ PLC และทำการจัดเก็บและเปิดแฟ้มข้อมูลได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

สรุป

การพัฒนาระบบการทำงานของเครื่องควบคุม PLC ส่วนมากจะมุ่งเน้นการควบคุมการทำงานของระบบในรูปแบบต่าง ๆ ลักษณะการควบคุมเป็นแบบจุดต่อจุด คือเครื่องจักร 1 เครื่องต่อเครื่องควบคุม PLC 1 เครื่อง ในวิทยานิพนธ์เล่มนี้ได้นำเสนอเป็นซอฟต์แวร์การออกแบบการควบคุมและแสดงผล PLC ที่สามารถออกแบบการทำงานของระบบและควบคุมการทำงานได้ทางคอมพิวเตอร์โดยการ Interface ผ่านทาง RS-232C และสามารถทำการ Save และ Open ที่มีการออกแบบการทำงานไว้แล้ว

ซอฟต์แวร์ที่ได้ทำการออกแบบนี้ สามารถใช้ควบคุมการทำงานของระบบได้เป็นที่น่าพอใจ รวมทั้งการประยุกต์ใช้งานร่วมกับโปรแกรมสนับสนุนการทำงานได้เป็นอย่างดี แต่อย่างไรก็ตามปัญหาเรื่องการเขียนโปรแกรมควบคุม ก็ยังไม่สามารถเขียนให้มีการควบคุมการทำงานของอุปกรณ์ต่างๆ ได้ครบถ้วน และการรับส่งข้อมูลระหว่างคอมพิวเตอร์กับ PLC ยังไม่เป็นที่เพียงพอต่อระบบการทำงาน

บรรณานุกรม

1. สุทธิศักดิ์ พงษ์ธนาพานิชย์,"Visual Basic 4.0 Professional",ซีเอ็ดยูเคชั่น
2. เอกสารประกอบการอบรมเชิงปฏิบัติการ PROGRAMMABLE LOGIC CONTROLLER, ภาควิชาเทคโนโลยีการวัดคุมทางอุตสาหกรรม คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
3. สุเรีษ เกียรติสุนทร,"หลักการทํางานและเทคนิคการประยุกต์ใช้งาน PC/PLC", ภาควิชาระบบควบคุม คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
4. ชาริน สิทธิธรรมชารี , ธัญชัช จันทร์ศักดิ์,"Microsoft Visual Basic Version 5.0"

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้