โปรแกรมแสดงผลกระบวนการผลิตที่ควบคุมโดยเครื่องควบคุมแบบโปรแกรมได้

PROCESS MONITORING SOFTWARE FOR PLC

โดย

นางสาวนัฏพร ฤทธิกุลสิทธิชัย

นายนัทธพงศ์ โมกขะเวส

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมระบบควบคุม

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2543

ปริญญานิพนธ์ปีการศึกษา 2543

ภาควิชา วิศวกรรมระบบควบคุม

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง โปรแกรมแสดงผลกระบวนการผลิตที่ควบคุมโดยเครื่องควบคุมแบบโปรแกรมได้
PROCESS MONITORING SOFTWARE for PLC

ผู้จัดทำ

นางสาวนัฏพร ฤทธิกุลสิทธิชัย รหัสประจำตัว 40010377

นายนัทธพงศ์ โมกขะเวส รหัสประจำตัว 40010379

.....อาจารย์ที่ปรึกษา

(รศ. สุเชียร เกียรติสุนทร)

โปรแกรมแสดงผลกระบวนการผลิตที่ควบคุมด้วยเครื่องคุมแบบโปรแกรมได้

นางสาวนัฏพร ฤทธิกุลสิทธิชัย
นายนัทธพงศ์ โมกขะเวส
รศ. สุเธียร เกียรติสุนทร อาจารย์ที่ปรึกษา
ปีการศึกษา 2543

บทคัดย่อ

ปริญญานิพนธ์ฉบับนี้มีวัตถุประสงค์เพื่อที่จะศึกษาทฤษฎี และการประยุกต์ใช้งานเครื่องควบคุมแบบโปรแกรมได้ซึ่งใช้ในกระบวนการอุตสาหกรรม โดยทำการออกแบบโปรแกรมกราฟฟิกที่ใช้แสดงผลกระบวนการผลิต ลักษณะของโปรแกรมแสดงผลกระบวนการผลิตนี้ จะเป็นโปรแกรมสำเร็จรูปที่ใช้สำหรับออกแบบ และจำลองกระบวนการผลิตซึ่งถูกควบคุมโดยเครื่องควบคุมแบบโปรแกรมได้ และเป็นโปรแกรมที่จะสามารถทำการเปลี่ยนแปลงสภาวะของกระบวนการผลิตตามการควบคุมของเครื่องควบคุมแบบโปรแกรมได้ ซึ่งจะแสดงการเปลี่ยนแปลงนั้นเป็นภาพกราฟิกบนหน้าจอคอมพิวเตอร์ นอกจากนี้ผู้ใช้งานยังสามารถสั่งการทำงานของเครื่องควบคุมแบบโปรแกรมได้ผ่านภาพกราฟิกเหล่านี้ได้ด้วย

การออกแบบและพัฒนาโปรแกรมสำเร็จรูปนี้ ได้ใช้โปรแกรมวิชวลเบสิค 6.0 (Visual Basic 6.0) ในการออกแบบดังนั้นโปรแกรมนี้จึงสามารถทำงานบนระบบปฏิบัติการวินโดวส์ (Windows) ได้ โปรแกรมแสดงผลกระบวนการผลิตนี้จะรอรับการทำงานร่วมกับเครื่องควบคุมแบบโปรแกรมได้รุ่น CQM1 ของบริษัท OMRON การติดต่อสื่อสารระหว่างเครื่องควบคุมแบบโปรแกรมได้กับเครื่องคอมพิวเตอร์นี้ จะใช้การสื่อสารผ่านพอร์ตอนุกรมตามมาตรฐานการสื่อสารผ่านพอร์ตอนุกรมแบบอาร์เอส 232ซี (RS-2323C) และข้อกำหนดการติดต่อสื่อสารของเครื่องควบคุมแบบโปรแกรมได้กับเครื่องคอมพิวเตอร์ (Host Link Communication)

PROCESS MONITORING SOFTWARE for PLC

Miss. NATTAPORN RITTIKOONSITTICHAI

Mr. NUTTHAPONG MOKKHAVESA

Associate Professor. SUTHIAN KRIENSUNTORN Advisor

Academic Year 2000

Abstract

The objective of this thesis is studying in the Programmable Logic Controller and its application for controlling the industrial process by designing the process monitoring software for PLC. This software be able to design and simulate the industrial process controlled by PLC. It be able to display the operation of process followed the operation of PLC on the monitor of computer. And user can control PLC through this software

The Visual Basic 6.0 was used to design and develop the Process Monitoring Software so it can work under the Windows System. This Software supports the OMRON's PLC (model CQM1). The communication between computer with PLC used communications protocol of PLC (Host Link Communication) and the standard RS232-C interface.

สารบัญ

สารบัญ (ต่อ)

สารบัญ (ต่อ)

สารบัญรูป

บทที่ 2

บทที่ 3

บทที่ 4

บทที่ 5

สารบัญรูป (ต่อ)

บทที่ 6

สารบัญตาราง

บทที่ 1
บทนำ

ปัจจุบันการใช้โปรแกรมสำเร็จรูปในงานควบคุมกระบวนการผลิตด้วยคอมพิวเตอร์มีการใช้งานกันอย่างแพร่หลายในงานอุตสาหกรรมและงานระบบควบคุมอัตโนมัติ ซึ่งโปรแกรมสำเร็จรูปที่ใช้ในระบบควบคุมการผลิตมีมากมายหลายชนิด โดยโปรแกรมหนึ่งที่เป็นที่ต้องการมาก ก็คือ โปรแกรมสำเร็จรูปที่สามารถเชื่อมต่อข้อมูลกับเครื่องควบคุมแบบโปรแกรมได้ ซึ่งจะประกอบด้วยภาพกราฟฟิกแสดงการทำงานของกระบวนการผลิต การรายงานผลข้อมูล และการแสดงผลบนหน้าจอคอมพิวเตอร์

โครงงานปริญญานิพนธ์ เรื่องโปรแกรมแสดงผลกระบวนการผลิตที่ใช้กับเครื่องควบคุมแบบโปรแกรมได้นี้ เป็นโปรแกรมที่จำลองการทำงานของกระบวนการผลิตที่ควบคุมด้วยเครื่องควบคุมแบบโปรแกรมได้ ลงบนระบบปฏิบัติการวินโดวส์(Windows)ของเครื่องคอมพิวเตอร์ โดยสามารถออกแบบกระบวนการผลิตและสามารถแสดงผลการเปลี่ยนแปลงของอุปกรณ์ต่างๆ ได้ตามการควบคุมของเครื่องควบคุมแบบโปรแกรมได้ ตลอดจนสามารถทำการสั่งงานเครื่องควบคุมแบบโปรแกรมได้ผ่านทางภาพกราฟฟิก(Graphic) ซึ่งอยู่บนหน้าจอคอมพิวเตอร์ได้

1.1 วัตถุประสงค์ของโครงงาน

1. เพื่อศึกษาการทำงานของเครื่องควบคุมแบบโปรแกรมได้ (PLC) ทั้งทางด้านทฤษฎี และการประยุกต์ใช้งาน

2. เพื่อสร้างโปรแกรมสำเร็จรูปที่สามารถจำลองกระบวนการผลิตเป็นภาพกราฟฟิกบนวินโดวส์ ซึ่งสามารถแสดงผลการเปลี่ยนแปลงที่เกิดจากการควบคุมของPLC และสามารถสั่งงานเครื่อง PLC ผ่านทางหน้าจอคอมพิวเตอร์ได้

1.2 ขอบเขตของโครงงาน

1.เป็นโปรแกรมสำเร็จรูปที่สามารถออกแบบกระบวนการผลิตบนหน้าจอวินโดวส์ของเครื่องคอมพิวเตอร์ได้

2. เป็นโปรแกรมสำเร็จรูปที่สามารถแสดงการเปลี่ยนแปลงของอุปกรณ์ในกระบวนการผลิตที่เกิดจากการควบคุมของเครื่อง PLC ได้

3.เป็นโปรแกรมสำเร็จรูปที่สามารถสั่งการทำงานของเครื่องPLCผ่านทางหน้าจอคอมพิวเตอร์ได้

4. เป็นโปรแกรมแสดงผลกระบวนการผลิต ที่สามารถบันทึกและจัดเก็บข้อมูลเพื่อนำไปใช้ในครั้งต่อไปได้

1.3 ขั้นตอนการพัฒนาโครงงาน

1. ศึกษาโครงสร้าง และการใช้งานเครื่องควบคุมแบบโปรแกรมได้(PLC) ในการควบคุมกระบวนการผลิต

2. ศึกษาวิธีการติดต่อสื่อสารระหว่างเครื่อง PLC กับคอมพิวเตอร์

2.1 ศึกษามาตรฐานการสื่อสารแบบอนุกรม RS-232C ที่ใช้ในการติดต่อสื่อสารระหว่างคอมพิวเตอร์กับอุปกรณ์ภายนอก

2.2 ศึกษารูปแบบการสื่อสารของ PLC ที่ใช้ในการติดต่อกับคอมพิวเตอร์ (Hostlink Communication)

3. ศึกษาโปรแกรมวิชวลเบสิค(Visual Basic) ซึ่งจะใช้ในการพัฒนาโปรแกรมแสดงผลกระบวนการผลิต

4. ออกแบบและพัฒนาโปรแกรมกราฟฟิกที่ใช้ในการแสดงผลกระบวนการผลิต และทำการเชื่อมต่อโปรแกรมที่ออกแบบได้กับเครื่อง PLC

5. ทดลองใช้โปรแกรมจำลองกระบวนการผลิตที่ออกแบบได้ ในการแสดงผลการควบคุมของเครื่อง PLC กับกระบวนการผลิตแบบต่าง ๆ

6. วิเคราะห์ และสรุปผลโครงงาน อธิบายปัญหาและข้อเสนอแนะ เพื่อเป็นแนวทางในการพัฒนาต่อไป

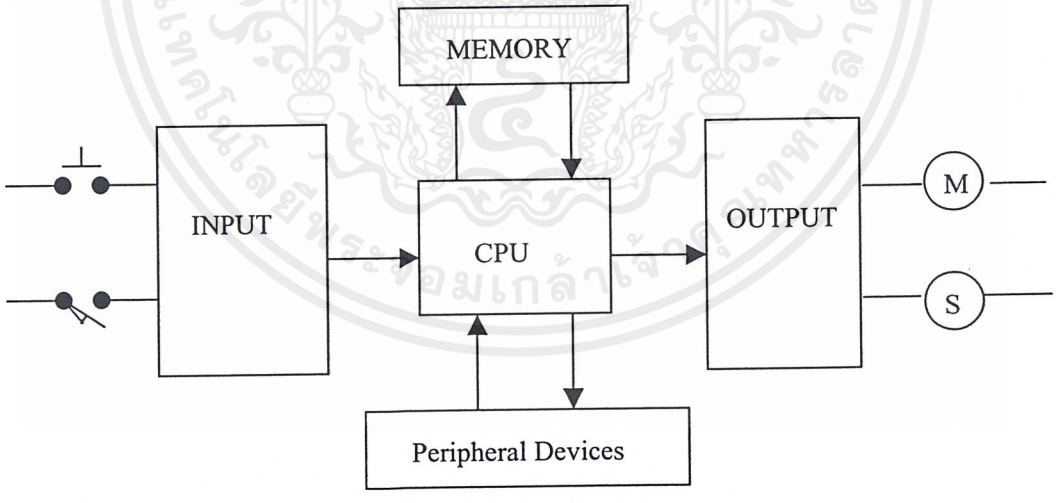
บทที่ 2
โครงสร้าง และ การใช้งาน เครื่องควบคุมแบบโปรแกรมได้

โปรแกรมแสดงผลกระบวนการผลิตนี้ ได้ถูกออกแบบมาให้ใช้งานรวมกับเครื่องควบคุมแบบโปรแกรมได้ยี่ห้อ OMRON SYSMAC CQM1 CPU21 ดังนั้นในบทนี้จะกล่าวถึงโครงสร้างและการใช้งานเครื่อง PLC รุ่นนี้

2.1 โครงสร้างและส่วนประกอบ ของ PLC

PLC เป็นอุปกรณ์คอมพิวเตอร์สำหรับใช้ในงานอุตสาหกรรม PLC ประกอบด้วย หน่วยประมวลผลกลาง หน่วยความจำ หน่วยรับข้อมูล หน่วยส่งข้อมูล และหน่วยป้อนโปรแกรม PLC ขนาดเล็ก ส่วนประกอบทั้งหมดของ PLC จะรวมเป็นเครื่องเดียว แต่ถ้าเป็นขนาดใหญ่สามารถแยกออกเป็นส่วนประกอบย่อย ๆ ได้

หน่วยความจำของ PLC ประกอบด้วยหน่วยความจำชนิด RAM และ ROM หน่วยความจำชนิด RAM ทำหน้าที่เก็บโปรแกรมของผู้ใช้และข้อมูลสำหรับใช้ในการปฏิบัติงานของ PLC ส่วน ROM ทำหน้าที่เก็บโปรแกรมสำหรับใช้ในการปฏิบัติงานของ PLC ตามโปรแกรมผู้ใช้



รูปที่ 2.1 แสดงโครงสร้างของ PLC

PLC แบ่งออกได้เป็น 4 ส่วนด้วยกันคือ

1. ส่วนที่เป็นหน่วยประมวลผลกลาง (Control Processing Unit :CPU)
2. ส่วนที่เป็นหน่วยความจำ (Memory)

3. ส่วนที่เป็นอินพุต/เอาท์พุต (Input Output :I/O)
4. ส่วนที่เป็นอุปกรณ์ติดต่อภายนอก (Peripheral Devices)

2.1.1 หน่วยประมวลผลกลาง (CPU:Central Processing Unit)

หน่วยประมวลผลกลาง เป็นส่วนที่ทำหน้าที่ควบคุมการทำงานของ PLC โดยทั่วไปแล้วมักจะใช้ ไมโครโปรเซสเซอร์ชนิด 8 บิต เป็นตัวประมวลผล ปกติหน้าที่ของ CPU คือรับข้อมูลอินพุทเข้ามาทำการประมวลผล แล้วส่งผลที่ได้ออกไปยังเอาท์พุท จากนั้นก็จะวนกลับไปรับข้อมูลอินพุทเข้ามาอีก ทำซ้ำๆในลักษณะเช่นนี้ไปเรื่อย ๆ ซึ่งเรียกกันว่าการสแกน(scan) การทำงานของ CPU จะอยู่ภายใต้การควบคุมของโปรแกรมที่ผู้ใช้ป้อนเข้าไป

นอกจากนี้ CPU ยังทำหน้าที่รับส่งข้อมูล กับอุปกรณ์ติดต่อ(peripheral device) ตรวจเช็คตัวเองและหน่วยความจำ ,ตรวจเช็คแหล่งจ่ายไฟ

2.1.2 หน่วยความจำ(Memory Unit)

หน่วยความจำเป็นองค์ประกอบหนึ่งที่สำคัญของระบบ เพราะใช้เป็นที่เก็บโปรแกรมและข้อมูล ขนาดของหน่วยความจำจะเป็นตัวกำหนดขีดความสามารถของระบบ โดยปกติแล้วมักจะมีหน่วยวัดขนาดเป็นสเต็ป(Step) หรือบรรทัดของการโปรแกรม PLC แบ่งหน่วยความจำออกเป็น 2 ส่วนที่สำคัญด้วยกัน คือ

1.) หน่วยความจำระบบ(System memory) เก็บโปรแกรมบริหารระบบและข้อมูลของระบบ

2.) หน่วยความจำผู้ใช้(User memory) เก็บโปรแกรมผู้ใช้ ,ข้อมูลของหน่วยอินพุท/เอาท์พุทและอุปกรณ์ภายใน

โปรแกรมบริหารระบบ และข้อมูลระบบไม่อนุญาตให้ผู้ใช้เปลี่ยนแปลง แก้ไขข้อมูลภายในแต่ผู้ใช้อาจสามารถตรวจสอบข้อมูลภายในหน่วยความจำในส่วนของข้อมูลระบบได้หรือตรวจสอบสภาพการทำงานของ PLC

โปรแกรมผู้ใช้ เป็นคำสั่งหรือโปรแกรมที่ผู้เขียนเขียนขึ้น เพื่อควบคุมเครื่องจักร หรือกระบวนการให้ทำงานตามต้องการ

ตารางที่ 2.1 การจัดสรรหน่วยความจำของ PLC

พื้นที่	อักษรย่อ	ช่วงการใช้งาน	หน้าที่
Internal Relay	IR	Words 000 to 246 Bits 00000 to 24615	ใช้ควบคุม I/O port , bitอื่น ๆ , Timer Counter และเก็บข้อมูลชั่วคราว
Special Relay	SR	Words 247 to 255 Bits 24700 to 25507	ใช้เก็บ System Clock , Flag , Control bit และข่าวสารสถานะ
Auxiliary Relay	AR	Words AR 00 to 27 Bits AR 0000 to 2715	ใช้เก็บ Flag, Bit ของฟังก์ชันพิเศษ และคงค่าสถานะระหว่าง power failure
Data Memory	DM	Read/Write DM 0000 to 0999 Read only DM 1000 to 1999	ใช้เก็บข้อมูลภายใน
Holding Relay	HR	Words HR 00 to 99 Bits HR 0000 to 9915	ใช้เก็บค่าข้อมูล และคงค่าข้อมูลไว้เมื่อปิดเครื่อง
Link Relay	LR	Words LR 00 to 63 Bits LR 0000 to 6315	ใช้งานหลากหลาย ใช้งานเป็น work บิต
Timer/Counter	TC	TC 000 to TC511	ใช้กำหนด timer และ counter , ใช้ในการ access compition flag , present value และ set value
Temporary Relay	TR	TR 00 to TR 07 (bits only)	ใช้เก็บข้อมูลชั่วคราว ใช้แยก Output จากจุดเดียวกันออกเป็นหลาย ๆ ส่วน
Program Memory	UM	UM:ขึ้นกับ memory unit ที่ใช้	ใช้เก็บโปรแกรมที่ถูก execute โดย CPU

ในตารางที่ 2.1 นั้นเป็นการจัดสรรหน่วยความจำแบบคร่าวๆของเครื่องPLC สำหรับการจัดสรรหน่วยความจำแบบละเอียดของเครื่อง PLC รุ่น CQM1 ของ OMRON นั้นสามารถดูได้จากภาคผนวกท้ายเล่ม

ในการใช้งานควบคุมโดยทั่วไปนั้น ปกติแล้วจะใช้หน่วยความจำ ในส่วนของ รีเลย์ภายใน (Internal Relay :IR) เป็นบิตควบคุมอินพุทเอาท์พุท(input/output) ตามการโปรแกรมของผู้ใช้ ซึ่งในส่วนของรีเลย์ภายใน(IR) นี้จะแบ่งเป็นส่วนย่อย ๆ ออกไปอีก 3 ส่วน คือ

1.) ส่วนพื้นที่อินพุท (Input Area) words 000 to 015 (bits 00000 to 01515)

2.) ส่วนพื้นที่เอาท์พุท (Output Area) words 100 to 115 (bits 10000 to 11515)

3.) ส่วนพื้นที่ใช้งาน (Work Area) words 016 to 095 and 116 to 195

2.1.3 หน่วยอินพุท/เอาท์พุท (Input/Output Unit)

ส่วนของอินพุท และเอาท์พุท จะต่อร่วมกับชุดควบคุมเพื่อรับสภาวะและสัญญาณต่าง ๆ เช่น หน่วยอินพุทรับสัญญาณหรือสภาวะแล้วส่งไปยัง CPU เพื่อประมวลผล เมื่อ CPU ประมวลผลแล้วจะส่งให้ส่วนของเอาท์พุท เพื่อให้อุปกรณ์ทำงานตามที่โปรแกรมเอาไว้

สัญญาณอินพุทจากภายนอกที่เป็นสวิตช์ และตัวตรวจจับชนิดต่าง ๆ จะถูกแปลงให้เป็นสัญญาณที่เหมาะสมถูกต้องไม่ว่าจะเป็น AC หรือ DC เพื่อส่งให้ CPU ดังนั้น สัญญาณเหล่านี้จึงต้องมีความถูกต้องไม่เช่นนั้นแล้ว CPU จะเสียหายได้

ในส่วนของเอาท์พุท จะทำหน้าที่รับค่าสภาวะที่ได้จากการประมวลผลของ CPU แล้วนำค่าเหล่านี้ไปควบคุมอุปกรณ์ทำงานเช่น รีเลย์ โซลีนอยด์ หรือ หลอดไฟเป็นต้น นอกจากนั้นแล้วยังทำหน้าที่แยกสัญญาณของหน่วยประมวลผลกลาง ออกจากอุปกรณ์เอาท์พุท โดยปกติเอาท์พุทนี้จะมีความสามารถขับโหลดด้วยกระแสไฟฟ้าประมาณ 1-2 แอมแปร์ แต่ถ้าโหลดต้องการกระแสไฟฟ้ามากกว่านี้ จะต้องต่อเข้ากับอุปกรณ์ขับอื่นเพื่อขยายให้รับกระแสไฟฟ้ามากขึ้น เช่น รีเลย์ หรือคอนแทคเตอร์ เป็นต้น

2.1.4 อุปกรณ์ติดต่อภายนอก (Peripheral Devices)

เป็นอุปกรณ์แบบต่างๆ ที่อำนวยความสะดวกในการพัฒนาโปรแกรมสามารถใช้ร่วมกับ PC ชนิดเดียวกันได้หลาย ๆ ตัว หน้าที่ของอุปกรณ์ติดต่อภายนอก ได้แก่

1.) ป้อนโปรแกรมเข้าไปใน หน่วยความจำของระบบ

2.) ใช้ในการแก้ไขโปรแกรม

3.) ใช้ในการเก็บรักษาโปรแกรม

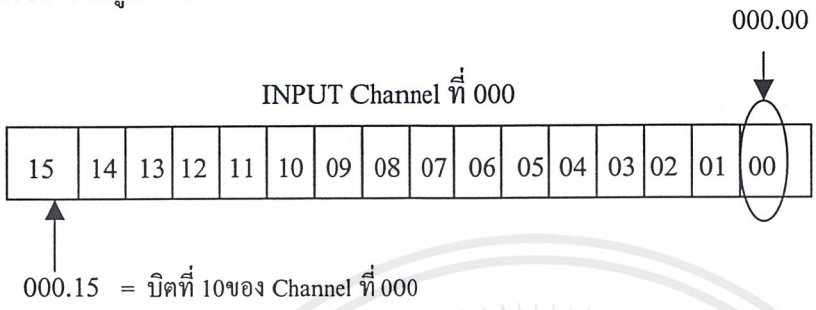
4.) ใช้ในการพิมพ์โปรแกรม

5.) ใช้แสดงสภาวะการควบคุม

อุปกรณ์ติดต่อภายนอกมีหลายชนิด และเหมาะกับงานแต่ละอย่าง เช่น เครื่องป้อนโปรแกรม (Program Panel Unit) ซึ่งใช้ในการป้อนโปรแกรมผู้ใช้ลงในหน่วยความจำของ PLC นอกจากนี้ยังทำหน้าที่ติดต่อระหว่างผู้ใช้กับ PLC เพื่อให้ผู้ใช้สามารถตรวจการปฏิบัติงานของเครื่อง PLC และผลการควบคุมเครื่องจักรและกระบวนการตามโปรแกรมควบคุมที่ผู้ใช้เขียนขึ้นได้อีกด้วย

2.3 การทำงานของเครื่องควบคุมแบบโปรแกรมได้

การกำหนดเบอร์ของรีเลย์ในเครื่อง PLC ของบริษัท OMRON นั้น จะกำหนดพื้นที่รีเลย์ (Relay) เป็นเวิร์ด (Word) หรือ แชนแนล(Channel) ซึ่งแต่ละแชนแนลจะประกอบด้วย 16 บิต ดังตัวอย่างในรูปที่ 2.2



รูปที่ 2.2 การกำหนดเบอร์รีเลย์

สำหรับเครื่อง PLC รุ่น CQM1 นั้น รีเลย์อินพุทจะเริ่มตั้งแต่ แชนแนลที่ 000 ถึง แชนแนลที่ 015 ส่วนในกรณีของเอาท์พุทก็จะเริ่มตั้งแต่ แชนแนลที่ 100 ถึง 115 ซึ่งทั้งรีเลย์อินพุท และเอาท์พุทนี้อยู่ในหน่วยความจำในส่วนของรีเลย์ภายใน (IR Area) นอกจากอินพุทและเอาท์พุทแล้วยังมีรีเลย์ที่ใช้ทำหน้าที่อื่นๆ อยู่ในหน่วยความจำส่วนต่าง ๆ การอ้างถึงจะมีการกำหนดเบอร์เหมือนกัน เช่น ต้องการอ้างถึงบิตที่ 1 ของรีเลย์ตัวที่ 15 ในหน่วยความจำส่วน Holding Relay Area สามารถอ้างถึงได้โดยเขียนว่า HR 15.01 เป็นต้น

2.4 วงจรตรรกะ และ ลอจิกแลดเดอร์ไดอะแกรม

หลักการทำงานของเครื่อง PLC นั้นมีพื้นฐานมาจากวงจรตรรกะ (logic) เพื่อให้เกิดสัญญาณเอาท์พุทที่มีเงื่อนไข ซึ่งนำไปควบคุมการทำงานแบบต่างๆ

วงจรตรรกะ เป็นวงจรไฟฟ้าที่ประกอบด้วยอุปกรณ์อิเล็กทรอนิกส์ หรือระบบรีเลย์ที่มีสัญญาณเพียง 2 ระดับ หรือ 2 สภาวะเท่านั้น PLC ใช้สัญญาณไฟฟ้า 2 ระดับแทน 2 เหตุการณ์ที่ต่างกัน เช่น การปิดเปิดวาล์ว การปิดเปิดสวิตช์ เป็นต้น วงจรตรรกะมี 2 ชนิด คือ แบบบวก (Positive Logic) และแบบลบ (Negative Logic) ลอจิกแบบบวกจะใช้สัญญาณไฟฟ้าระดับสูงแทนสภาวะลอจิก “1” และใช้สัญญาณไฟฟ้าระดับต่ำแทนสภาวะลอจิก “0” ส่วนวงจรลอจิกแบบลบจะใช้สัญญาณไฟฟ้าระดับต่ำแทนสภาวะลอจิก “1” และใช้สัญญาณไฟฟ้าระดับสูงแทนสภาวะลอจิก “0”

สภาวะทางลอจิก คือ สภาวะ “1” หรือ “0” ใช้แทนการทำงานของอุปกรณ์ที่เปลี่ยนแปลง 2 สภาวะ ระบบควบคุมที่ใช้ระบบรีเลย์ และ PLC จะนำเอาสภาวะของอุปกรณ์เหล่านี้มาปฏิบัติลอจิก

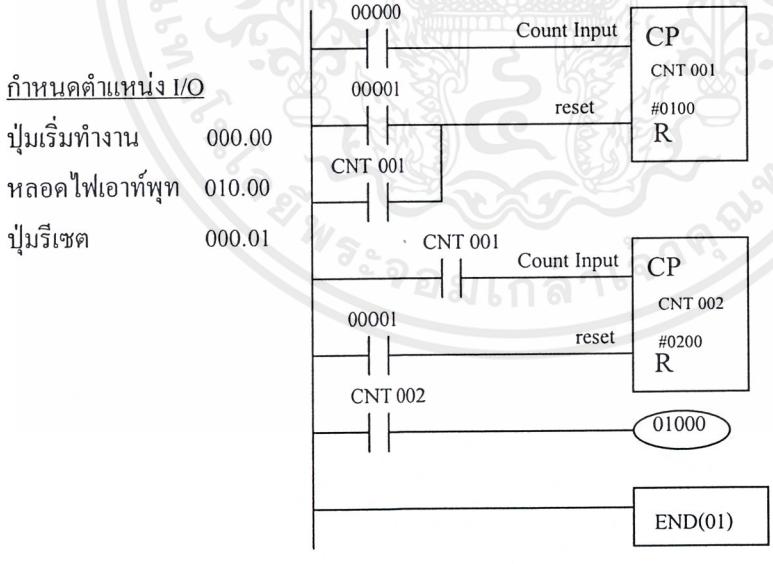
ด้วยกัน เพื่อให้เข้ากันกับเงื่อนไขการควบคุม ปฏิบัติการลอจิกประกอบด้วย AND OR และ NOT เพื่อทำให้เกิดสภาวะอินพุทต่าง ๆ เช่น A,B ทำให้เกิดเอาท์พุท Y เป็นต้น

ภาษาแลดเดอร์(Logic Ladder Diagram)

การควบคุมแบบตรรกะโดยใช้เครื่อง PLC เป็นตัวควบคุมนั้น เราจะต้องทำการเขียนโปรแกรมการทำงานของระบบเป็นภาษาแลดเดอร์ (Logic Ladder Diagram)ให้เครื่อง PLC ซึ่งภาษาแลดเดอร์นี้ เป็นภาษาสัญลักษณ์ ที่ใช้แทนการเชื่อมต่อระหว่างรีเลย์ต่างๆ เพื่อสร้างเงื่อนไขการควบคุม การเขียนวงจรควบคุมจะอยู่ระหว่างเส้นจ่ายไฟในแนวตั้ง 2 เส้นเรียกว่า Rails หรือ Legs

คำสั่งพื้นฐานที่ใช้สร้างแลดเดอร์ให้กับ PLC แบ่งเป็นหลายประเภท ได้แก่ คำสั่งที่ใช้สร้างรีเลย์อินพุท และเงื่อนไขควบคุมการทำงาน เช่น LD , NOT , OR , AND เป็นต้น และคำสั่งที่ใช้สร้างรีเลย์เอาท์พุท คือ OUT ,OUT NOT นอกจากนี้ยังมีคำสั่งที่เป็นฟังก์ชั่นซึ่งทำหน้าที่เฉพาะอย่างในรูปแบบต่าง ๆ อีกด้วย ซึ่งสามารถศึกษาได้จากภาคผนวกท้ายเล่ม

ตัวอย่างการเขียนคำสั่งแลดเดอร์ให้เครื่อง PLC แสดงได้ดังรูปที่2.3 ซึ่งเป็นรูปที่แสดงรูปแบบและวิธีการป้อนโปรแกรมแลดเดอร์ เพื่อทำการนับคนดูคอนเสิร์ตซึ่งสามารถจุคนดูได้ 20,000 คน โดยใช้ปุ่มเริ่มทำงาน (Photo Switch) หลังจากนับได้ 20,000 คนแล้วให้หลอดไฟของเอาท์พุท (Output Lamp) ทำงานเพื่อแสดงว่าคนเต็มแล้ว



00000	LD	00000
00001	LD	00001
00003	OR	CNT001
00004	CNT	001
		#0100
00005	LD	CNT001
00006	LD	00001
00007	CNT	002
		#0200
00008	LD	CNT002
00009	OUT	01000
00010	END(01)	

รูปที่ 2.3 ตัวอย่างการป้อนคำสั่งในภาษาแลดเดอร์

จากรูปที่ 2.3 สามารถอธิบายขั้นตอนการทำงานของเครื่อง PLC จากวงจรแลดเดอร์ได้ ดังนี้ รีเลย์ตำแหน่งที่ 00000 จะทำการนับจำนวนคนที่ผ่านเข้าประตู โดยจะทำการทริกทุกครั้งที่มีคนผ่านเข้าประตู 1 คน รีเลย์ตำแหน่งที่ 00000 นี้จะต่อเข้ากับตัวนับ (Counter) ตัวที่ 1 ซึ่งการทำงานของตัวนับ คือ จะทำการนับลงจากค่าเริ่มต้นที่ตั้งไว้จนกว่าจะเป็นศูนย์ หน้าคอนแทคของรีเลย์ตัวนับก็จะติด ในที่นี้ตัวนับตัวที่ 1 จะทำการกำหนดค่าเริ่มต้นเอาไว้ที่ 100 ดังนั้นเมื่อรีเลย์ตำแหน่งที่ 0000 ทริคครบ 100 ครั้ง จะทำให้ รีเลย์ของตัวนับตัวที่ 1 ติด 1 ครั้ง ซึ่งจะส่งผลให้รีเลย์ตัวที่ 2 ซึ่งกำหนดค่าเริ่มต้นไว้ที่ 200 นับลง 1 ครั้งด้วย(เพราะเอาท์พุทของตัวนับตัวที่ 1 เป็นอินพุทของตัวนับตัวที่ 2) ดังนั้นเมื่อรีเลย์ตัวที่ 2 นับลงครบ 200 ครั้ง นั้นหมายถึงว่า รีเลย์ตัวที่ 1 ได้ทำงานมาแล้ว 200 รอบการทำงาน ซึ่งแต่ละรอบการทำงานนั้นจะนับคนผ่านประตูได้ 100 คน นั้นคือ หน้าคอนแทคของรีเลย์ตัวที่ 2 นี้จะติด ก็ต่อเมื่อนับจำนวนคนผ่านประตูได้ 20000 คน ซึ่งจะทำให้หลอดไฟ ซึ่งต่อกับรีเลย์ตำแหน่งที่ 01000 ติด ส่วนรีเลย์ตำแหน่งที่ 00001 นั้นถูกใช้เป็นรีเลย์สำหรับสั่งยกเลิกการทำงานของตัวนับ

บทที่ 3
การติดต่อสื่อสารระหว่างเครื่องคอมพิวเตอร์ กับ PLC

การติดต่อสื่อสารระหว่างเครื่องคอมพิวเตอร์ กับ PLC นั้น ได้กระทำผ่านพอร์ตอนุกรม โดยทำการสื่อสารตามรูปแบบของมาตรฐานการสื่อสารแบบอนุกรม RS232-C ส่วนข้อมูลที่รับส่งกับเครื่อง PLC นั้น จะต้องถูกจัดให้อยู่ในรูปแบบที่ เครื่อง PLC สามารถเข้าใจได้ สำหรับเครื่อง PLC ของบริษัท OMRON นี้ สามารถทำการติดต่อสื่อสารได้โดยใช้ Host Link Communication เป็นข้อกำหนดในการจัดรูปแบบของชุดข้อมูลและคำสั่งที่ใช้ในการสื่อสาร ก่อนที่จะทำการส่งออกทางพอร์ตอนุกรมด้วยมาตรฐาน RS-232C

3.1 มาตรฐานการสื่อสารแบบอนุกรม RS-232C

มาตราฐานการสื่อสารแบบอนุกรม RS-232C เป็นมาตราฐานที่ถูกกำหนดโดย EIAซึ่งเป็นองค์กรอุตสาหกรรมอิเล็กทรอนิกส์ของอเมริกา โดยแบ่งการเชื่อมต่อออกเป็น 2 ลักษณะคือ DTE (Data Terminal Equipment) และ DCE (Data Communication Equipment) ซึ่งโดยปกติ DTE จะต้องต่อเข้ากับ DCE เสมอ เช่น การต่อเครื่องคอมพิวเตอร์ (อุปกรณ์ DTE) เข้ากับโมเด็ม (อุปกรณ์ DCE) เป็นต้น

พอร์ตอนุกรม RS-232C จะเป็นพอร์ตของเครื่องคอมพิวเตอร์ที่มีขาต่อ (Connector) ทั้ง 9 ขา และ 25 ขา และเราเรียกว่าพอร์ต COM1: และ COM2: นั่นเอง ในความเป็นจริงพอร์ตอนุกรมไม่ได้ถูกควบคุมโดยตรงจาก CPU บนเมนบอร์ด แต่การสื่อสารทั้งหมดจะถูกจัดการโดยชิป UART (Universal Asynchronous Receiver/Transmitter) อีกทีหนึ่ง ซึ่งชิป UART นี้จะทำหน้าที่ในการรับและส่งข้อมูลดังต่อไปนี้

การส่งข้อมูล :

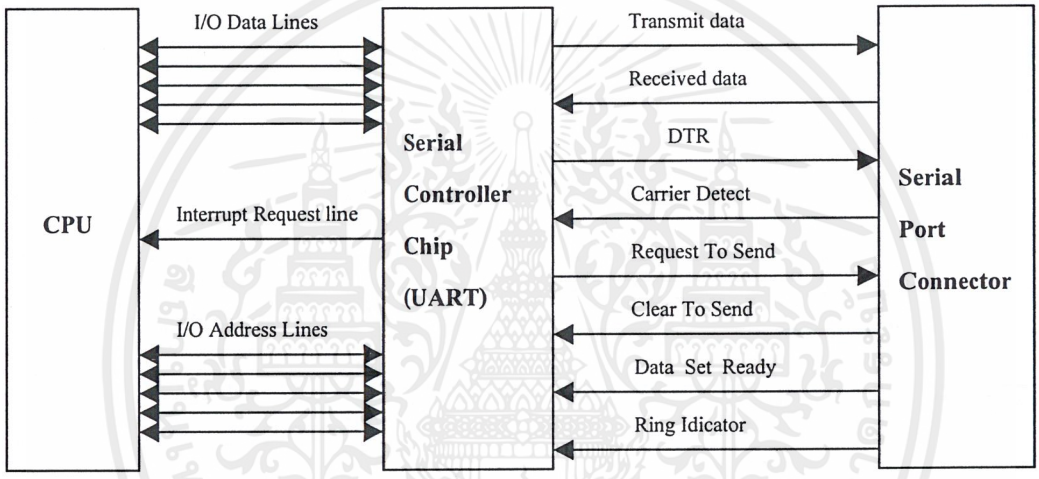
- รับตัวอักษรจากเครื่องคอมพิวเตอร์
- แปลงตัวอักษรให้เป็นสายข้อมูลแบบบิต (เรียกว่าขบวนการ Serialization)
- สร้างเฟรมข้อมูลโดยการเพิ่มบิตที่จำเป็นสำหรับการสื่อสารและการตรวจสอบ เช่น บิต Start , Stop และ Parity เป็นต้น
- ส่งผ่านเฟรมข้อมูลที่สร้างขึ้นมาแล้วจากขั้นตอนที่ผ่านมา ด้วยความเร็วของโมเด็ม หรือพอร์ตอนุกรม (baud rate)

การรับข้อมูล :

- รับตัวอักษรจากอินเทอร์เฟซ (Interface)

- ตรวจสอบความถูกต้องของเฟรมข้อมูลตามมาตรฐานเฟรมที่กำหนด โดยถ้าหากเฟรมข้อมูลมีรูปแบบที่ไม่ถูกต้องก็จะมีการแจ้งข้อผิดพลาดทันที
- ตรวจสอบความถูกต้องของพาริตี้
- แปลงสายข้อมูลแบบบิตให้เป็นตัวอักษร
- ส่งตัวอักษรให้กับเครื่องคอมพิวเตอร์
- แสดงสถานะความพร้อมที่จะรับข้อมูลตัวอักษรต่อไปให้กับอินเทอร์เฟซ

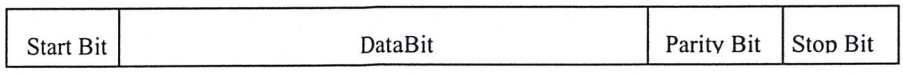
สำหรับการเชื่อมต่อสายสัญญาณต่าง ๆ ระหว่าง CPU ของเมนบอร์ดของเครื่องคอมพิวเตอร์กับพอร์ตอนุกรมนั้น จะต้องกระทำผ่านทางชิป UART ซึ่งจะมีวิธีการเชื่อมต่อดังในรูปที่ 3.1



รูปที่ 3.1 การเชื่อมต่อสายสัญญาณระหว่าง CPU ของเครื่องคอมพิวเตอร์ กับ พอร์ตอนุกรม

3.1.1 รูปแบบของเฟรมข้อมูล (Data Format)ในมาตรฐาน RS-232C

การติดต่อสื่อสารภายใต้มาตรฐานการสื่อสารแบบอนุกรม RS-232C เป็นการส่งข้อมูล แบบ Asynchronous ซึ่งเวลาในการรับส่งจะไม่คงที่ ดังนั้นเครื่องคอมพิวเตอร์ กับ อุปกรณ์ภายในนอกที่ติดต่อด้วยจะต้องจัดรูปแบบกลุ่มของข้อมูลให้อยู่ในรูปแบบเดียวกันเรียกว่า เฟรมข้อมูล(Data Frame) ซึ่งจะช่วยให้จังหวะในการรับส่งข้อมูลถูกต้องยิ่งขึ้น รูปแบบของเฟรมข้อมูลในการสื่อสารแสดงได้ดังรูปที่ 3.2



รูปที่ 3.2 รูปแบบการจัดเฟรมข้อมูลภายใต้มาตรฐานการสื่อสารแบบอนุกรม RS-232C

ใน 1 เฟรมข้อมูลจะประกอบด้วยบิตข้อมูลต่าง ๆ ที่มีความหมายดังต่อไปนี้

- บิตเริ่มต้นข้อมูล (Start bit)

ในการส่งข้อมูลแบบ asynchronous นั้น เราจะต้องมีวิธีการบอกโมเด็มให้ทราบถึงจุดเริ่มต้นของข้อมูลที่ต้องการส่ง ดังนั้นก่อนหน้าข้อมูลในทุก ๆ เฟรมจะต้องถูกนำหน้าด้วยบิตเริ่มต้นข้อมูล (Start bit) จำนวน 1 บิต เสมอ

- บิตข้อมูล (Data bit)

เมื่อชิป UART ได้รับตัวอักษรที่ส่งมาจากเครื่องคอมพิวเตอร์แล้ว ก็ต้องทำการแปลงตัวอักษรดังกล่าวให้เป็นสายข้อมูลชนิดบิตที่มีความยาวตั้งแต่ 5 ถึง 8 บิต ซึ่งเราเรียก ขบวนการแปลงตัวอักษรให้เป็นสายข้อมูลชนิดบิตนี้ว่าขบวนการ Serialization จากนั้นโมเด็มก็จะทำการส่งแต่ละบิตไปยังโมเด็มปลายทาง โดยจะเริ่มต้นส่งจากบิตที่มีนัยสำคัญต่ำสุด (least significant bit) ไปยังบิตที่มีนัยสำคัญสูงสุด (most significant bit)

- บิตพาริตี้ (Parity bit)

เนื่องจากการส่งผ่านข้อมูลทางสายโทรศัพท์นั้น สามารถเกิดสัญญาณรบกวนได้ง่าย ด้วยเหตุนี้จึงเป็นไปได้มากที่สถานะของแต่ละบิตของข้อมูลที่ถูกส่งจะมีการเปลี่ยนแปลง เช่น จากบิต 0 เป็นบิต 1 ในระหว่างการส่งผ่านข้อมูล เป็นต้น ด้วยเหตุนี้จึงต้องนำการตรวจสอบค่าพาริตี้ (Parity Check) มาใช้ โดยในการส่งผ่านข้อมูลด้วยโปรโตคอล START/STOP นั้น ในการส่งตัวอักษรข้อความทั่วไปจะใช้เพียง 7 บิตข้อมูลเท่านั้น ดังนั้นจึงมีการเพิ่มบิตพาริตี้ต่อสายเฟรมข้อมูลที่ถูกสร้างขึ้นมา เพื่อตรวจสอบสถานะของผลบวกของบิตที่เป็น 1 ของสายบิตข้อมูล พาริตี้มีค่า 2 อย่างคือ

พาริตี้คู่ (Even Parity) เมื่อรวมเลข 1 ในสายข้อมูลแล้วได้เป็นจำนวนคู่

พาริตี้คี่ (Odd Parity) เมื่อรวมเลข 1 ในสายข้อมูลแล้วได้เป็นจำนวนคี่

-บิตสิ้นสุดข้อมูล (Stop bit)

ในการส่งข้อมูลแบบอะซิงโครนัส (asynchronous) นั้น บิตสิ้นสุดข้อมูลจะต้องถูกเพิ่มต่อท้ายข้อมูลทุกครั้ง เพื่อแจ้งการสิ้นสุดของสายข้อมูล บิตสิ้นสุดข้อมูลสามารถกำหนดให้มี 1 หรือ 2 บิตก็ได้ แต่ต้องกำหนดให้ตรงกันทั้งในคอมพิวเตอร์ และอุปกรณ์ที่สื่อสารด้วย

ในการส่งข้อมูลผ่านพอร์ตอนุกรมนั้น บิตเริ่มต้นของสายข้อมูลจะทำให้สถานะสายส่งมีค่าต่ำเป็นเวลา 1 รอบ จากนั้นจึงส่งเฟรมข้อมูลออกไป ซึ่งอัตราความเร็วที่ใช้ในการสื่อสารแบบอนุกรมมีหน่วยเป็น บอร์ด (Baud Rate) หรือ บิตต่อวินาที (BPS) สำหรับเครื่องคอมพิวเตอร์ทั่วไปนั้นสามารถใช้อัตราความเร็วในการสื่อสารได้สูงสุด 9600 บอร์ด

3.2 ข้อกำนดในการสื่อสารระหว่าง PLC กับ คอมพิวเตอร์ (Host Link Communication)

การติดต่อสื่อสารระหว่างเครื่อง PLC กับคอมพิวเตอร์ สามารถทำได้โดยการใช้งานในส่วนของ Host Link Communication ซึ่งถูกพัฒนาโดย OMRON เพื่อใช้ในการติดต่อสื่อสารแบบอนุกรมระหว่างเครื่อง PLC กับโฮสต์คอมพิวเตอร์ (Host Computer) ผ่านทางพอร์ตอนุกรมตามมาตราฐาน RS-232C รูปแบบของการจัดเรียงข้อมูลเป็นชุดคำสั่ง และชุดตอบสนอง (Response) ที่ใช้ใน Host Link นี้ จะเป็นชุดคำสั่งที่ PLC สามารถเข้าใจ และทำงานตามได้ โดยกลุ่มของข้อมูลที่ส่งในครั้งหนึ่ง ๆ เรียกว่า เฟรม (Frame) ซึ่งใน 1 เฟรมสามารถบรรจุตัวอักษรได้สูงสุด 131 ตัวอักษร

ในการสื่อสารระหว่างคอมพิวเตอร์กับ PLC นั้น จะเป็นการส่งชุดคำสั่ง(Command Frame) จากคอมพิวเตอร์ไปยังเครื่อง PLC แล้วเครื่อง PLC ก็จะทำการส่งชุดผลตอบสนอง (Response Frame) กลับไปให้เครื่องคอมพิวเตอร์ ซึ่งรูปแบบการจัดเรียงข้อมูลของชุดคำสั่ง และชุดผลตอบสนองนี้ จะแตกต่างกัน ซึ่งจำเป็นที่ต้องศึกษาและจดจำรูปแบบของชุดคำสั่งและรูปแบบชุดผลตอบสนอง เพื่อนำไปใช้ประโยชน์ในการเขียนโปรแกรมรับส่งข้อมูลต่อไป

3.2.1 รูปแบบของชุดข้อมูลที่ใช้ในการติดต่อสื่อสารตามข้อกำหนด Host Link Communication

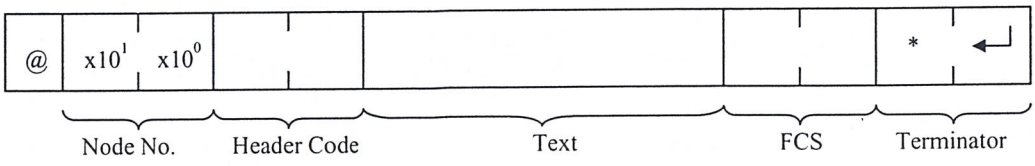
จากที่ได้กล่าวมาแล้วข้างต้นว่า เฟรมข้อมูลที่ใช้ในการสื่อสารตามข้อกำหนดของ Host Link นั้นจะมีตัวอักษรได้ไม่เกิน 131 ตัวอักษรต่อเฟรม ดังนั้นถ้าข้อมูลที่ต้องการจะส่งมีมากเกินที่จะใส่ใน 1 เฟรมได้ก็จะต้องมีการแบ่งข้อมูลออกเป็นหลายเฟรม ซึ่งรูปแบบข้อมูลที่มีการแบ่งเฟรม และไม่มีการแบ่งเป็นหลายเฟรมจะแตกต่างกัน ดังจะกล่าวได้ดังต่อไปนี้

3.2.1.1 รูปแบบของเฟรมข้อมูลที่ไม่มีการแบ่งเป็นหลายเฟรม

สำหรับข้อมูลที่มีขนาดไม่มากสามารถส่งให้หมดได้ภายในเฟรมเดียวนี้ เฟรมนั้นจะทำการลงท้ายด้วยเทอร์มิเนเตอร์ (ตัวอักษร * และ CHR$(13)) ซึ่งจะบอกให้อุปกรณ์รับข้อมูลทราบว่าข้อมูลได้สิ้นสุดลงภายในเฟรมนี้ ซึ่งรูปแบบข้อเฟรมข้อมูลที่ไม่มีการแบ่งเฟรมนี้สามารถแบ่งออกได้เป็นรูปแบบของชุดคำสั่งจากคอมพิวเตอร์ และชุดตอบสนองจากเครื่อง PLC

-รูปแบบชุดคำสั่งจากคอมพิวเตอร์ (Command Format)

ก่อนที่จะทำการส่งคำสั่ง หรือข้อมูลใด ๆ ไปให้เครื่อง PLC ผู้ใช้จะต้องจัดข้อมูลให้อยู่ในรูปแบบที่ PLC สามารถเข้าใจได้ ซึ่งรูปแบบของชุดคำสั่งแสดงได้ดังรูปที่ 3.3



รูปที่ 3.3 รูปแบบชุดคำสั่งที่ส่งจากคอมพิวเตอร์ไปยัง PLC

จากรูปแบบของชุดคำสั่งที่ส่งจากเครื่องคอมพิวเตอร์ไปยัง PLC สามารถอธิบายส่วนต่าง ๆ ของเฟรมได้ดังนี้

@ เป็นสัญลักษณ์ที่จะต้องใส่ที่หน้าเฟรมทุกเฟรมที่ใช้ใน Host Link

Node No ใช้กำหนดเครื่อง PLC ที่เครื่องคอมพิวเตอร์ต้องการจะสื่อสารด้วย ในกรณีที่มีเครื่อง PLC เพียงเครื่องเดียวต่ออยู่กับคอมพิวเตอร์ ให้ใช้ Node No = 00

Header Code เป็นรหัสคำสั่งที่จะใช้สั่งงาน PLC ประกอบด้วยตัวอักษร 2 ตัวอักษรซึ่งเป็นอักษรย่อที่ใช้แทนคำสั่งควบคุมการทำงานของ PLC

Text เป็นกลุ่มของข้อมูลที่ใช้ร่วมกันกับคำสั่งตาม Header Code

FCS Frame Check Sequence เป็นส่วนที่ใช้ตรวจสอบความถูกต้องของข้อมูล

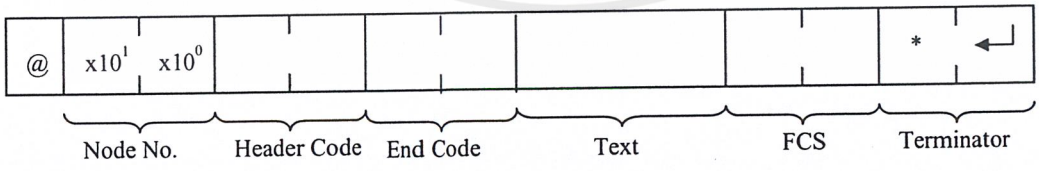
Terminator ประกอบด้วยเครื่องหมาย * และ carriage return (CHR$(13)) ใช้เพื่อบอกอุปกรณ์ตัวรับว่าสิ้นสุดเฟรมข้อมูล

ตัวอย่างของชุดคำสั่งที่ส่งจากคอมพิวเตอร์ไปยัง PLC ที่ไม่มีการแบ่งเฟรมสามารถแสดงได้จากคำสั่งอ่านข้อมูลจากหน่วยความจำในส่วนพื้นที่รีเลย์ภายใน (Inertial Relay) ตั้งแต่เวิร์ดที่ 0 ถึงเวิร์ดที่ 10 สามารถจัดข้อมูลได้ดังนี้

@ 00 RR 00000011 42 * CHR$(13)

- รูปแบบชุดผลตอบสนองจาก PLC

เมื่อ PLC ได้รับชุดคำสั่งจากเครื่องคอมพิวเตอร์แล้ว มันจะทำการส่งชุดผลตอบสนองกลับมาให้คอมพิวเตอร์ ซึ่งชุดตอบสนองนี้อาจจะมีข้อมูลต่างๆมาด้วยซึ่งสามารถนำไปใช้ในการตรวจสอบความผิดพลาด และสถานะในการรับส่งข้อมูลได้ ดังนั้นในคอมพิวเตอร์จึงควรจะมีโปรแกรมที่ใช้แปลความหมายของเฟรมที่ส่งกลับเข้ามาได้ โดยศึกษาจากรูปแบบชุดผลตอบสนองจาก PLC ดังรูปที่ 3.4



รูปที่ 3.4 รูปแบบชุดผลตอบสนองจากเครื่อง PLC

เมื่อลองเปรียบเทียบรูปแบบของชุดคำสั่งในรูปที่ 3.3 กับรูปแบบของชุดผลตอบสนองในรูปที่ 3.4 จะเห็นได้ว่าส่วนประกอบส่วนใหญ่เหมือนกัน แต่จะแตกต่างกันตรงที่รูปแบบชุดผลตอสนองจากเครื่อง PLC นี้จะมีส่วนของ End Code ส่งกลับมาให้ด้วย ซึ่งจะใช้ในการแสดงสถานะในการสื่อสาร และความผิดพลาดที่เกิดขึ้นในการสื่อสาร ในกรณีที่ไม่มีความผิดพลาดเกิดขึ้น End Code จะ

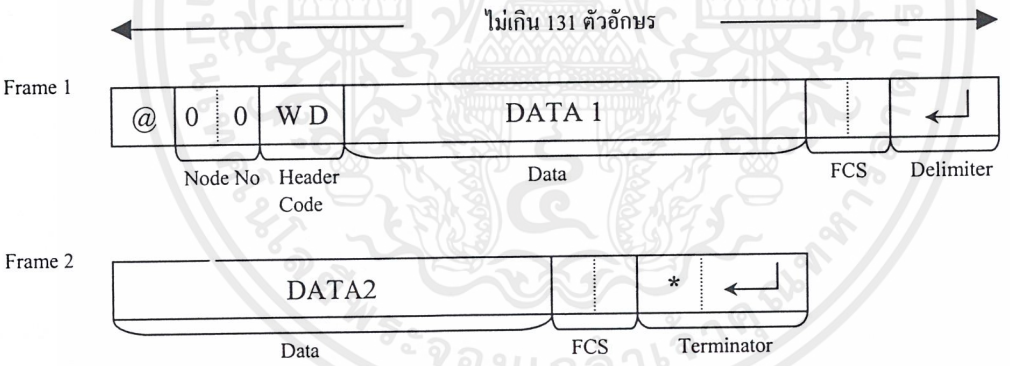
เท่ากับ 00 แต่ถ้า End Code ไม่เท่ากับศูนย์ แสดงว่าการสื่อสารมีความผิดพลาด ซึ่งสามารถตรวจสอบสาเหตุของความผิดพลาดได้จากตารางแสดงค่า End Code กับสาเหตุของความผิดพลาด ในภาคผนวกท้ายเล่ม เช่น End Code = 18 แสดงว่าเกิดความผิดพลาดในเรื่องขนาดความยาวของข้อมูลเกิน 131 ตัวอักษรเป็นต้น

ตัวอย่างชุดผลตอบสนองที่ไม่มีการแบ่งเฟรมข้อมูล เช่นผลตอบสนองจากคำสั่งอ่านหน่วยความจำในส่วนของพื้นที่รีเลย์ภายในจะมีรูปแบบของข้อมูลดังนี้

@ 00 RR 00 011001001100 56 * CHR$(13)

3.2.1.2 รูปแบบของเฟรมข้อมูลที่มีการแบ่งเป็นหลายเฟรม

สำหรับข้อมูลที่มีความยาวเกิน 131 ตัวอักษร จะต้องมีการแบ่งออกเป็นหลาย ๆ เฟรม ซึ่งรูปแบบโดยรวมของชุดคำสั่งที่ส่งจากคอมพิวเตอร์ กับชุดผลตอบสนองจาก PLC จะเหมือนกัน แต่ในการลงท้ายเฟรมแต่ละเฟรมจะลงท้ายด้วยดีลิมิตเตอร์(ตัวอักษร CHR$(13)) เพื่อบอกให้อุปกรณ์ตัวรับทราบว่าข้อมูลยังไม่หมด และจะมีเฉพาะเฟรมสุดท้ายของข้อมูลเท่านั้นที่ลงท้ายด้วยเทอร์มิเนเตอร์เพื่อบอกให้อุปกรณ์ตัวรับรู้ว่าสิ้นสุดข้อมูล รูปแบบนี้แสดงได้ดังรูปที่ 3.5 ซึ่งเป็นคำสั่งในการเขียนข้อมูลลงในหน่วยความจำส่วนพื้นที่ข้อมูล (DATA Memory)



รูปที่ 3.5 รูปแบบข้อมูลที่มีการแบ่งออกเป็นหลายเฟรม

สำหรับข้อกำหนดในการแบ่งข้อมูลระหว่างเฟรมข้อมูลนั้น มีข้อควรระวังอยู่บางประการ คือ ห้ามทำการแบ่งข้อมูลที่อยู่ในเวิร์ดเดียวกันให้อยู่คนละเฟรมข้อมูล ซึ่งจะทำให้เกิดการผิดพลาดในข้อมูลที่รับเข้ามาได้ เนื่องจากข้อมูลในเวิร์ดเดียวกันนั้นจะเรียงลำดับตามเลขฐาน 16 ดังนั้นถ้าแบ่งให้อยู่คนละเฟรมแล้วการเรียงลำดับในเลขฐานจะเปลี่ยนไป

ตัวอย่างการชุดคำสั่งที่มีการแบ่งข้อมูลออกเป็นหลาย ๆ เฟรม ส่วนมากจะเป็นคำสั่งในการเขียนข้อมูลลงในหน่วยความจำของ PLC หรือไม่ก็จะเป็นชุดของผลตอบสนองจากคำสั่งอ่านข้อมูล

เป็นต้น เช่น ชุดคำสั่งเขียนข้อมูลลงในหน่วยความจำในส่วนของพื้นที่ข้อมูล (DATA Memory) จะมีรูปแบบดังนี้

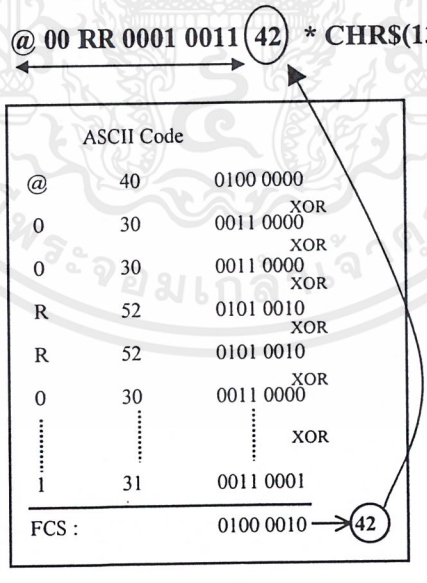
@ 00 WD 0000 1001 1111 1111...1101 58 CHR$(13)

0011 1111 0101 ...1110 42 * CHR$(13)

3.2.2 การคำนวณ FCS (Frame Check Sequence)

FCS (Frame Check Sequence) เป็นส่วนที่ใช้ในการตรวจสอบความถูกต้องของการรับส่งข้อมูลตามข้อกำหนดของ Host Link ซึ่งจะทำหน้าที่คล้ายบิตพาริตี้ในการสื่อสารแบบอนุกรมตามมาตรฐาน RS-232C นั่นเอง

ในการการรับส่งชุดข้อมูลระหว่างเครื่องคอมพิวเตอร์ กับ PLC ทุกครั้งจะต้องมีการคำนวณ FCS เพื่อใส่ให้กับเฟรมที่จะส่งออกไป หรือ นำ FCS ที่คำนวณได้จากเฟรมที่รับเข้ามา มาเปรียบเทียบกับค่า FCS ในเฟรมที่รับเข้ามาว่าตรงกันหรือไม่ ถ้า FCS ที่คำนวณได้ กับ FCS ที่รับเข้ามาไม่ตรงกัน จะแสดงให้เห็นว่าข้อมูลต้นทางกับข้อมูลปลายทางไม่เหมือนกัน ซึ่งอาจเกิดขึ้นได้จากมีสัญญาณรบกวนระหว่างการสื่อสารซึ่งจะต้องทำการแก้ไขเพื่อให้เกิดความถูกต้องในการสื่อสาร หลักการคำนวณค่า FCS คือ การแปลงตัวอักษรที่อยู่หน้าตำแหน่ง FCS ในเฟรมข้อมูลทั้งหมดให้เป็นรหัสแอสกี้ (ASCII) เลขฐาน 16 แล้วนำค่าที่ได้ทั้งหมดมา XOR กันจะได้ค่า FCS สำหรับเฟรมนั้นๆ การคำนวณ FCS แสดงได้ดังรูปที่ 3.6



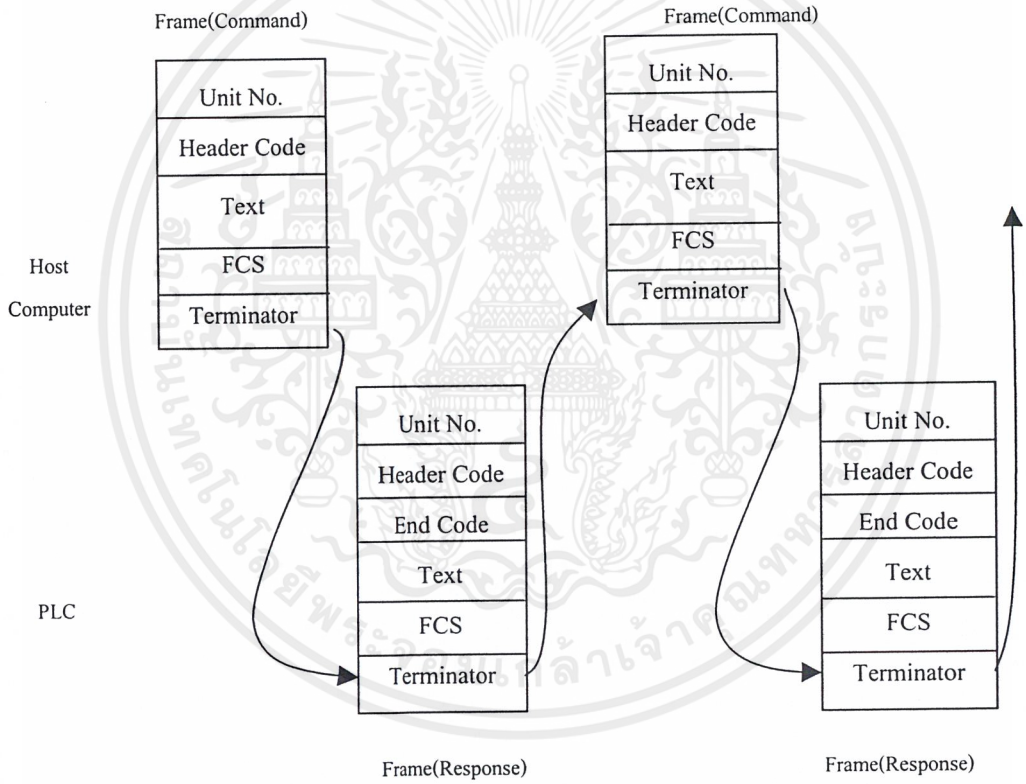
รูปที่ 3.6 การคำนวณ FCS

3.2.3 ขั้นตอนการรับส่งชุดข้อมูลระหว่างเครื่องคอมพิวเตอร์ กับ PLC

จากที่ได้กล่าวมาแล้วข้างต้น จะเห็นได้ว่ารูปแบบของชุดข้อมูลที่ใช้ในการสื่อสารใน Host Link นี้จะแบ่งเป็น 2 รูปแบบ คือ ชุดข้อมูลที่ลงท้ายเฟรมด้วยเทอร์มิเนเตอร์ และ ชุดข้อมูลที่ลงท้ายด้วยดีลิมิตเตอร์ ซึ่งเป็นข้อมูลที่มีการแบ่งออกเป็นหลายเฟรม ดังนั้นการรับส่งชุดข้อมูลระหว่างเครื่องคอมพิวเตอร์กับ PLC จึงต้องแบ่งเป็น 2 แบบเช่นกัน คือ

3.2.3.1 การรับส่งชุดข้อมูลที่ไม่มีการแบ่งเฟรมข้อมูลออกเป็นหลายเฟรม

สำหรับการรับส่งชุดข้อมูลที่ไม่มีการแบ่งข้อมูลออกเป็นหลายเฟรมนั้น เฟรมที่ใช้ในการสื่อสารทุกเฟรมจะมีขนาดไม่เกิน 131 ตัวอักษร และจะต้องลงท้ายเฟรมด้วยเทอร์มิเนเตอร์ ขั้นตอนในการรับส่งแสดงได้ดังรูปที่ 3.7



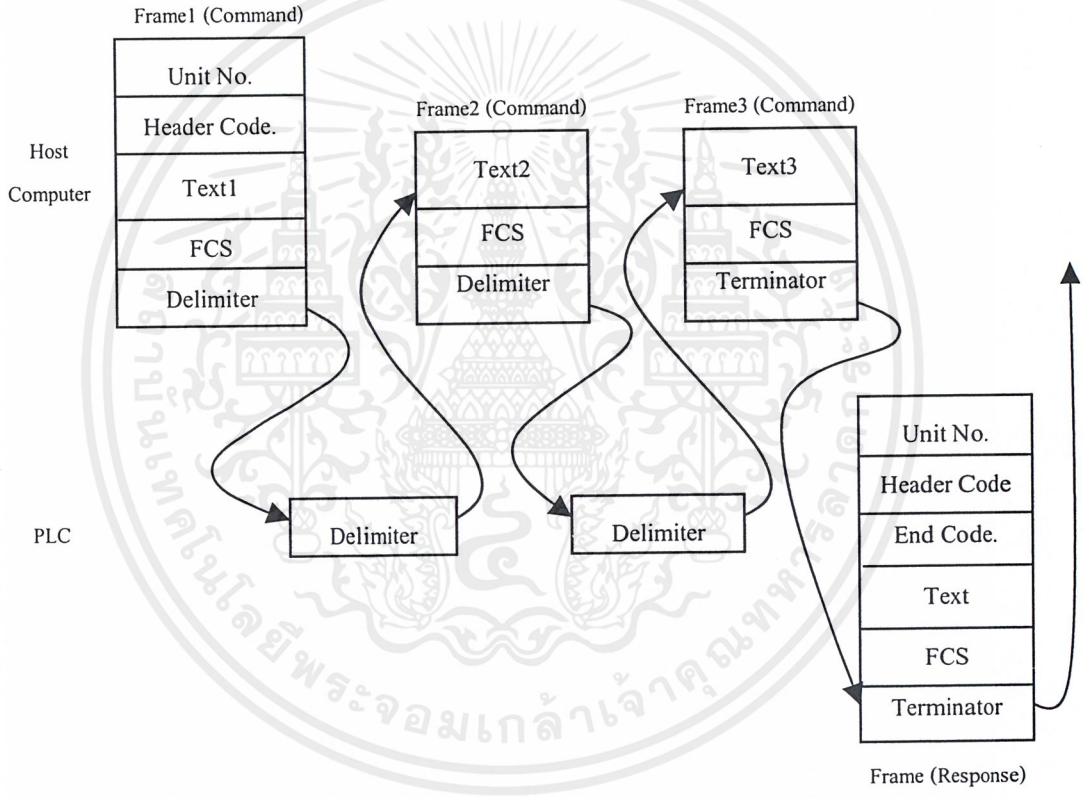
รูปที่ 3.7 การรับส่งชุดข้อมูลที่ไม่มีการแบ่งเฟรมข้อมูล

จากรูปที่ 3.7 สามารถอธิบายขั้นตอนการรับส่งชุดข้อมูลที่ไม่มีการแบ่งข้อมูลออกเป็นหลายชุดได้ คือ เมื่อคอมพิวเตอร์ส่งชุดคำสั่งที่จบด้วยเทอร์มิเนเตอร์มาให้ PLC เครื่อง PLC จะทำการตรวจสอบความถูกต้องของเฟรมที่รับเข้ามาจาก FCS จากนั้นจึงตรวจสอบว่าเฟรมที่รับเข้ามานั้นจบเฟรมด้วยเทอร์มิเนเตอร์หรือดีลิมิตเตอร์ ถ้าเฟรมนั้นจบเฟรมด้วยเทอร์มิเนเตอร์ PLC ก็จะนำข้อมูลนั้นไปประมวลผลได้ทันที แล้วทำการจัดเฟรมชุดผลตอบสนองกลับไปให้คอมพิวเตอร์ เมื่อรับข้อ

มูลเข้ามาแล้ว คอมพิวเตอร์ก็จะต้องทำการตรวจสอบว่าข้อมูลในชุดผลตอบสนองนั้นไม่ความถูกต้องหรือไม่โดยดูจาก FCS และ END Code แล้วจึงตรวจสอบว่าข้อมูลสิ้นสุดหรือยังโดยตรวจสอบว่าเฟรมจบด้วยเทอร์มิเนเตอร์หรือไม่ ถ้าเฟรมนั้นจบด้วยเทอร์มิเนเตอร์ คอมพิวเตอร์ก็สามารถนำข้อมูลนั้นมาประมวลผล และแสดงให้ผู้ใช้เห็นได้ทันที และเริ่มการทำงานรอบต่อไป

3.2.3.2 การรับส่งชุดข้อมูลที่มีการแบ่งข้อมูลออกเป็นหลายเฟรม

สำหรับชุดคำสั่งหรือชุดผลตอบสนองที่มีความยาวมากเกิน 131 ตัวอักษร (มักจะเป็นชุดคำสั่งในการอ่านเขียนหน่วยความจำ) เฟรมข้อมูลจะถูกแบ่งออกเป็นหลาย ๆ เฟรม และคั่นแต่ละเฟรมด้วยดีลิมิตเตอร์ ซึ่งจะมีขั้นตอนการรับส่งแสดงได้ดังรูปที่ 3.8



รูปที่ 3.8 การรับส่งข้อมูลที่มีการแบ่งเฟรมข้อมูล

จากรูปที่ 3.8 จะเห็นได้ว่าเมื่อชุดคำสั่งที่ส่งมาให้ PLC นั้นจบเฟรมด้วยดีลิมิเตอร์ PLC จะไม่สร้างชุดผลตอบสนองทันที แต่จะส่งดีลิมิตเตอร์กลับไปยังคอมพิวเตอร์เพื่อให้ทำการส่งชุดข้อมูลชุดต่อไปมาให้ก่อน และจะทำแบบนี้จนกว่าชุดคำสั่งที่ส่งมาจะจบเฟรมด้วยเทอร์มิเนเตอร์ซึ่งแสดงการสิ้นสุดข้อมูล PLCจึงจะนำข้อมูลที่ได้จากแต่ละเฟรมมารวมกันแล้วนำไปประมวลผล จากนั้นจึงทำการส่งชุดผลตอบสนองกลับไปยังเครื่องคอมพิวเตอร์ ในกรณีที่ชุดผลตอบสนองมีความยาวมาก

PLC ก็จะทำการแบ่งชุดผลตอบสนองออกเป็นหลายๆ เฟรมเช่นเดียวกัน ซึ่งขั้นตอนการรับส่งก็จะเหมือนกับการรับส่งชุดคำสั่งที่มีการแบ่งข้อมูลนั้นเอง ดังนั้นจึงมีความจำเป็นที่จะต้องทำการเขียนโปรแกรมให้คอมพิวเตอร์ทำการส่งดีลิมิตเตอร์กลับไปให้ PLC โดยอัตโนมัติเพื่อสั่งให้ PLC ส่งข้อมูลส่วนที่เหลือ เมื่อตรวจสอบพบว่าชุดผลตอบสนองนั้นมีการแบ่งข้อมูล

บทที่ 4
การใช้งานโปรแกรมวิชวลเบสิค

โครงงานปริญญานิพนธ์ เรื่อง โปรแกรมแสดงผลกระบวนการผลิตที่ควบคุมด้วย PLC นี้ ได้ใช้โปรแกรมวิชวลเบสิค 6.0 (Visual Basic 6.0) เป็นพื้นฐานในการออกแบบและพัฒนาโปรแกรม ดังนั้นจึงต้องเข้าในส่วนประกอบ และการใช้งานโปรแกรมวิชวลเบสิคอย่างลึกซึ้งจึงจะสามารถสร้างโปรแกรมสำเร็จรูปที่สามารถใช้งานบนวินโดวส์ได้อย่างมีประสิทธิภาพ

4.1 โปรแกรมวิชวลเบสิค 6.0 (Visual Basic 6.0)

วิชวลเบสิค นับได้ว่าเป็นตัวแปลภาษาตัวแรกของวงการคอมพิวเตอร์สมัยใหม่ที่ประสบความสำเร็จเป็นอย่างสูง สำหรับงานด้านการสร้างแอปพลิเคชันภายใต้ระบบปฏิบัติการวินโดวส์ เพราะเป็นตัวแปลภาษาที่ถูกออกแบบสภาพแวดล้อมให้ง่ายต่อการใช้งาน โดยเน้นหนักด้านการใช้กราฟิกเป็นสื่อแทนเมนูหรือคำสั่งต่าง ๆ จนเป็นที่มาของคำว่า “Visual” วิชวลเบสิกสามารถออกแบบหน้าตาโปรแกรมสำหรับผู้ใช้งานแบบกราฟิก หรือที่เรียกว่า กราฟิกยูสเซอร์อินเตอร์เฟส (GUI) ได้ทันที โดยใช้เครื่องมือที่เตรียมไว้แล้ว ทำให้มีโอกาสเห็นหน้าตาของแอปพลิเคชันตั้งแต่ตอนพัฒนา ซึ่งจะง่ายต่อการจัดรูปแบบ และการแก้ไขหากความต้องการของผู้ใช้งานเปลี่ยนไป

วิชวลเบสิค สนับสนุนการเขียนโปรแกรมแบบ OOP (Object Oriented Programming) ซึ่งมีหลักสำคัญในการเขียนคือ การได้แบ่งส่วนต่าง ๆ ของโปรแกรมออกเป็นส่วนย่อยที่สุดเท่าที่จะสามารถทำงานเสร็จได้ภายในตัวเอง ไม่สามารถแบ่งย่อยได้อีก ที่เรียกว่า ออบเจ็กต์ (Object) แล้วกำหนดการทำงานให้กับออบเจ็กต์เหล่านั้นด้วยพรอพเพอร์ตี้ (Properties) หรือ เมธอด (Method) มีข้อดีคือสามารถกำหนดการทำงานและแก้ไขออบเจ็กต์ได้ง่าย เพราะโดยออบเจ็กต์เองแล้วการเปลี่ยนแปลงดังกล่าวจะไม่กระทบส่วนอื่น ทำให้สะดวกในการตรวจสอบการทำงาน และมีความน่าเชื่อถือมากอีกด้วย คำศัพท์ที่ควรรู้เกี่ยวกับการใช้งานวิชวลเบสิค ได้แก่

- กราฟฟิกยูสเซอร์อินเตอร์เฟส (GUI) คือ รูปแบบการทำงานกับผู้ใช้คอมพิวเตอร์โดยการใช้รูปภาพ ทำให้ง่ายในการสื่อความหมาย ซึ่งจำเป็นต้องมีออบเจ็กต์ เช่น ปุ่ม ,กรอบความ , สกรอลบาร์ ซึ่งคอยรับคำสั่งจากผู้ใช้ และตอบสนองการทำงานคำสั่งเหล่านั้นในรูปแบบต่าง ๆ เช่น เสียง สี สัน ข้อความ เป็นต้น

- ออบเจ็กต์ (Object) คือ หน่วยย่อยที่สุดของโปรแกรม ซึ่งเป็นพื้นฐานของแอปพลิเคชัน โดยเราจะใช้งาน หรือติดต่อกับออบเจ็กต์ผ่านทางพรอพเพอร์ตี้ และ เมธอด และเขียนโปรแกรมเพื่อให้เกิดการทำงานที่สัมพันธ์กันระหว่าออบเจ็กต์ ตัวอย่างออบเจ็กต์ ฟอร์ม คอนโทรล เป็นต้น

- คอนโทรล (Control or ActiveX Control) คือ ออบเจ็กต์ที่เตรียมไว้สำหรับใช้งานวิชวลเบสิก ซึ่งจะกำหนดการทำงานให้กับคอนโทรลด้วยพรอพเพอร์ตี้ และ เมธอด

- พรอพเพอร์ตี้ (Properties) คือ คุณสมบัติ หรือลักษณะของออบเจ็กต์ ได้แก่ ชื่อออบเจ็กต์ ,ความสูง , ความกว้าง , การแสดงรูปแบบตัวชี้บนออบเจ็กต์ ,การซ่อนออบเจ็กต์ เป็นต้น

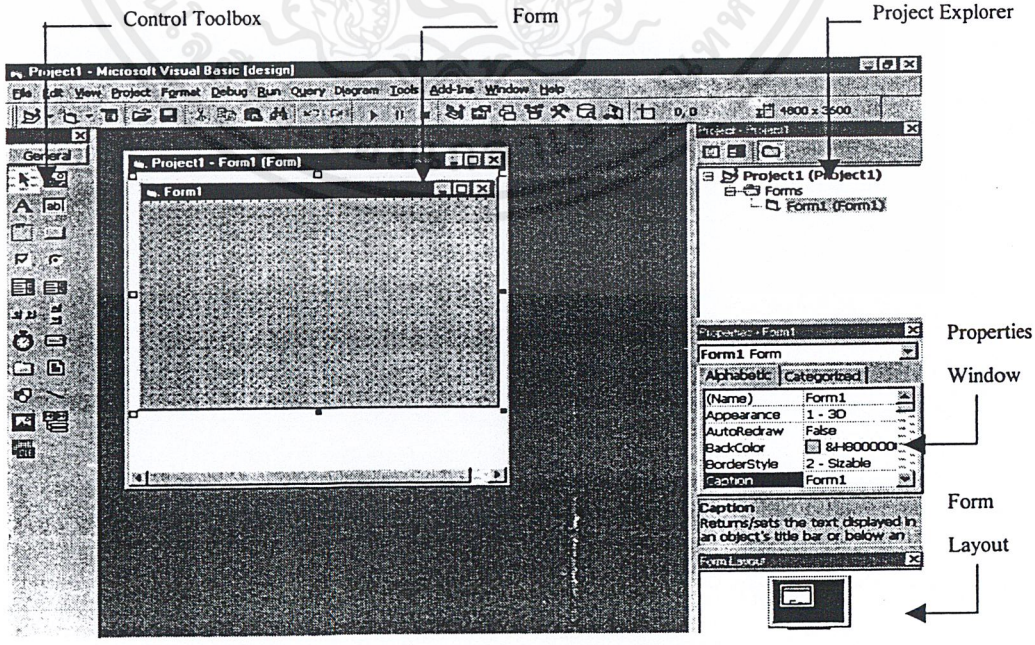
- วิธี หรือ เมธอด (Method) คือ ความสามารถในการทำงานของออบเจ็กต์ เป็นลักษณะเฉพาะของแต่ละออบเจ็กต์ (อาจจะเหมือนกันได้)

- เหตุการณ์ (Event) คือ สิ่งที่เกิดขึ้น ซึ่งเหตุการณ์ที่เกิดขึ้น อาจมาจากผู้ใช้งาน หรืออาจมาจากระบบปฏิบัติการ เมื่อมีเหตุการณ์ต่าง ๆ เกิดขึ้นเราจะต้องเขียนโปรแกรมเพื่อรองรับกับเหตุการณ์ที่กำลังจะเกิดขึ้น โดยการเปลี่ยนค่าพรอพเพอร์ตี้ของออบเจ็กต์ หรือ เรียกใช้เมธอดของออบเจ็กต์ที่มีอยู่ เช่น เมื่อมีการคลิกส์เมาส์ที่ปุ่มขวา , เมื่อผู้ใช้กรอกข้อความ , เมื่อแอปพลิเคชันเริ่มใช้งาน เป็นต้น ซึ่งเป็นสิ่งที่ดำเนินไปเมื่อมีการใช้งานแอปพลิเคชัน

-โพรซิเยอร์ (Procedure) คือโปรแกรมย่อยที่ผู้พัฒนาแอปพลิเคชันเขียนขึ้นสำหรับทำงานชิ้นหนึ่ง ซึ่งเป็นส่วนที่ถูกเรียกใช้งานจากส่วนต่าง ๆ ของแอปพลิเคชัน

4.2 สภาพแวดล้อมของวิชวลเบสิค 6.0

ก่อนการเริ่มใช้งานโปรแกรมวิชวลเบสิค เราจะต้องทำการศึกษาถึงส่วนประกอบต่าง ๆ ของวิชวลเบสิคเสียก่อน หรือที่เราเรียกว่า สภาพแวดล้อม(Environment) ของวิชวลเบสิคนั่นเอง ซึ่งสามารถแสดงได้ดังรูปที่ 4.1



รูปที่ 4.1 แสดงสภาพแวดล้อมของวิชวลเบสิค 6.0

จากรูปที่ 4.1 องค์ประกอบต่างๆ สำหรับสภาพแวดล้อมของวิชวลเบสิค 6.0 จะขออธิบายเฉพาะส่วนที่สำคัญต่อการพัฒนาโปรแกรม ดังต่อไปนี้

ฟอร์ม (Form)

เป็นส่วนที่ใช้ติดต่อกับผู้ใช้โปรแกรม โดยจะบรรจุคอนโทรลต่าง ๆ ที่ต้องการ เพื่อจัดหน้าตาของฟอร์มให้ใช้งานได้สะดวกและใช้งานได้ตามต้องการ ฟอร์มที่ใช้ในการสร้างแอปพลิเคชันมาตรฐานมี 2 รูปแบบ คือ SDI (Single Document Interface) ซึ่งเป็นฟอร์มที่ทำงานเพียงตัวเดียวได้ และ MDI (Multi-Document Interface) เป็นฟอร์มหลักที่ทำงานร่วมกับฟอร์มย่อยอื่น ๆ

หน้าต่างโปรเจ็กต์ (Project Explorer Window)

เป็นหน้าต่างที่แสดงรายการโปรเจ็กต์ และส่วนของไฟล์ที่เกี่ยวข้องทั้งหมดในวิชวลเบสิค ซึ่งกำลังเปิดใช้งานอยู่ในขณะนั้น ซึ่งการรวมเอาไฟล์ต่าง ๆ เข้าด้วยกันเพื่อสร้างแอปพลิเคชันภายใต้วิชวลเบสิค เราเรียกว่า โปรเจ็กต์ (Project) ซึ่งโครงสร้างของไฟล์ต่าง ๆ ที่ใช้ในโปรเจ็กต์ ถูกจำแนกออกเป็นหลายแบบดังตารางที่ 4.1

ตารางที่ 4.1 การจำแนกลักษณะของไฟล์ต่าง ๆ ในวิชวลเบสิค

โปรเจ็กต์ไฟล์	นามสกุลไฟล์
EXE Project	.VBP
Form	.FRM
Module	.BAS
User Control	.CTL

หน้าต่างคุณสมบัติ (Properties Window)

เป็นหน้าต่างที่รวบรวมคุณสมบัติทั้งหมดของฟอร์มหรือคอนโทรลเอาไว้ ซึ่งคุณสมบัติทั้งหมดที่ปรากฏในหน้าต่างนี้จะเป็นคุณสมบัติที่ผู้เขียนโปรแกรมสามารถกำหนดค่าได้ในขณะกำลังออกแบบ เมื่อผู้เขียนโปรแกรมทำการแก้ไขค่าคุณสมบัติต่าง ๆ ในหน้าต่างคุณสมบัตินี้จะส่งผลต่อคอนโทรลตัวนั้นทันทีซึ่งบางคุณสมบัติสามารถแสดงให้เห็นการเปลี่ยนแปลงคุณลักษณะได้ทันที เช่น คุณสมบัติเกี่ยวกับขนาด หรือ สี เป็นต้น ส่วนคุณสมบัติบางอย่างจะแสดงผลให้เห็นก็ต่อเมื่อผู้อ่านมีการรับแอปพลิเคชันเท่านั้น เช่น การซ่อนคอนโทรล เป็นต้น สำหรับการแก้ไขคุณสมบัติของคอนโทรลนั้นผู้เขียนโปรแกรมสามารถแก้ไขได้ทั้งจากการเขียนโค้ด หรือแก้ไขโดยตรงจากหน้าต่างคุณสมบัติ

ฟอร์มเลย์เอาท์ (Form Layout)

เป็นหน้าต่างที่ใช้สำหรับแสดงรูปแบบตำแหน่งของการวางฟอร์มทั้งหมดที่อยู่ในโปรแกรม เราสามารถใช้ฟอร์มเลย์เอาท์เพื่อบอกตำแหน่งของฟอร์มต่าง ๆ ในหน้าจอคอมพิวเตอร์ โดยสามารถเคลื่อนย้ายฟอร์มภายในฟอร์มเลย์เอาท์ได้โดยการคลิกฟอร์มที่ต้องการค้างไว้แล้วจัดวางไว้ที่ตำแหน่งอื่น ๆ ทำให้สามารถจัดวางตำแหน่งของฟอร์มต่างๆ ให้สัมพันธ์กันกับฟอร์มหลักได้อย่างเหมาะสม

แถบกล่องเครื่องมือ (Control ToolBox)

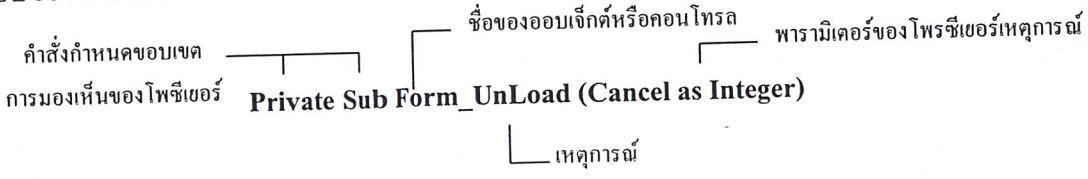
เป็นหน้าต่างที่มีไว้สำหรับบรรจุคอนโทรลต่าง ๆ (ActiveX Control) ซึ่งคอนโทรลที่อยู่ในแถบกล่องเครื่องมือนี้ เป็นเครื่องมือที่ใช้ติดต่อกับผู้ใช้งานทำให้เกิดความสะดวกในการทำงาน ดังนั้นจึงสามารถเรียกชื่ออีกอย่างได้ว่า “กล่องเครื่องมือ (ToolBox)” การใช้งานแถบกล่องเครื่องมือ ทำได้โดยการคลิกเพื่อเลือกคอนโทรลที่ต้องการ แล้วนำมาวางลงในฟอร์มโดยวิธีลากแล้วปล่อย (DnD) หรือทำการดับเบิ้ลคลิกที่ไอคอนของคอนโทรลที่ต้องการในหน้าต่างคอนโทรลทูลบ็อกซ์โดยตรงเลยก็ได้

4.3 การใช้งานออบเจ็กต์ฟอร์ม และคอนโทรลในวิชวลเบสิค

ก่อนที่จะอธิบายคุณสมบัติ วิธี และ เหตุการณ์ของออบเจ็กต์หรือคอนโทรลในวิชวลเบสิค จะต้องทำความเข้าใจถึงรูปแบบของโค้ดในการเรียกใช้งานหรือการแก้ไขคุณสมบัติของออบเจ็กต์หรือคอนโทรลนั้นๆ เสียก่อน

หลักเกณฑ์ในการเขียนโค้ดสำหรับกำหนดคุณสมบัติ (Properties) หรือวิธี (Method) ที่ใช้ในโปรแกรมวิชวลเบสิคสามารถแสดงดังนี้

สำหรับโพซีเยอร์เหตุการณ์ (Event Procedure) ของออบเจ็กต์ และคอนโทรล สามารถอธิบายรูปแบบของโค้ดได้ดังนี้



4.3.1 ออบเจ็กต์ Form และ MDI Form

ฟอร์ม (Form) เป็นออบเจ็กต์พื้นฐานของวิชวลเบสิค ที่ถูกสร้างขึ้นมาเพื่อทำหน้าที่เป็นหน้าต่าง (Window) หรือ ไดอะล็อกบ็อกซ์ (Dialog Box) สำหรับการสื่อการใช้านหรือการทำงานกับผู้ใช้ ซึ่งอาจจะเป็นแบบทางเดียวหรือโต้ตอบสองทางก็ได้ ดังนั้นเพื่อให้โปรแกรมเมอร์สามารถที่จะควบคุมพฤติกรรมการแสดงผลหรือการตอบสนองต่อการกระทำใด ๆ กับออบเจ็กต์ฟอร์ม วิชวลเบสิคจึงได้กำหนดคุณสมบัติโฟรซีเยอร์เหตุการณ์และวิธีสำหรับออบเจ็กต์ฟอร์ม นอกจากออบเจ็กต์ฟอร์มแล้ว ยังมีฟอร์มอีกประเภทหนึ่งที่ทำหน้าที่เป็นตัวบรรจุออบเจ็กต์ฟอร์ม นั่นคือ MDI Form ซึ่งออบเจ็กต์ฟอร์มที่จะสามารถบรรจุภายใน MDI Form ได้นั้นจะต้องถูกกำหนดคุณสมบัติ MDIChild ให้เท่ากับ True เสมอ

นอกจากผู้เขียนโปรแกรมสามารถสร้างออบเจ็กต์ฟอร์มในณะออกแบบได้โดยใช้คำสั่ง Add Form ของวิชวลเบสิคแล้ว ผู้เขียนโปรแกรมยังสามารถสร้างฟอร์มขณะรันแอปพลิเคชันได้ด้วยโดยการเขียนโค้ดโดยใช้คำสงวน New ร่วมกับประโยคในการประกาศตัวแปร Dim หรือ Set เป็นต้น

คุณสมบัติ (Properties) ของฟอร์ม

ฟอร์มมีคุณสมบัติหลาย ๆ อย่าง สามารถมีผลต่อลักษณะ และพฤติกรรมของฟอร์มในเวลาที่แสดงแบบฟอร์มนั้นออกมา ในที่นี้จะอธิบายถึงคุณสมบัติบางคุณสมบัติที่น่าสนใจ และมีการนำมาใช้ในการออกแบบโปรแกรมแสดงผลกระบวนการผลิตนี้

- AutoRedraw

รายงานหรือกำหนดการแสดงผลด้วยวิธีกราฟิกของฟอร์ม ซึ่งจะมีผลต่อการวาดฟอร์มใหม่อีกครั้งในกรณีที่ขนาดของฟอร์มมีการเปลี่ยนแปลง หรือฟอร์มซ้อนโดยฟอร์มอื่น ๆ

รูปแบบการใช้งาน : Object.AutoRedraw = Boolean

ถ้าค่าบูลีนเป็นจริง หมายถึง การกำหนดให้วิชวลเบสิคสามารถทำการวาดฟอร์มและกราฟิกต่าง ๆ ของฟอร์มใหม่อีกครั้งโดยอัตโนมัติ ทุกครั้งที่ฟอร์มมีการเปลี่ยนแปลง แต่ถ้าค่าบูลีนเป็นเท็จ วิชวลเบสิคจะยกเลิกการวาดฟอร์มและคอนโทรลของฟอร์มใหม่

- BackColor และ ForeColor

ใช้ในการกำหนดสีพื้น (BackColor) หรือสีตัวอักษรและเส้นที่ใช้ในการวาดกราฟฟิกต่าง ๆ (ForeColor) ของฟอร์มหรือออบเจ็คต์

รูปแบบการใช้งาน : Object.BackColor = Value หรือ Object.ForeColor = Value

- MDIChild

ใช้กำหนดให้ฟอร์มแสดงผลในรูปแบบของฟอร์มลูก ภายในฟอร์มหลัก MDI Form

รูปแบบการใช้งาน : Object.MDIChild = Boolean

- Tag

ใช้จัดเก็บข้อมูลพิเศษ โดยที่ข้อมูลที่ถูกจัดเก็บในคุณสมบัตินี้จะไม่มีผลใด ๆ กับฟอร์ม หรือออบเจ็กต์นั้น โดยทั่วไปเราจะใช้ในการเก็บข้อมูลชั่วคราวของฟอร์มหรือออบเจ็กต์

รูปแบบการใช้งาน : Object.Tag = String

โพรซีเยอร์เหตุการณ์ของฟอร์ม

- Load และ Unload

เหตุการณ์ Load จะเกิดเมื่อภายในแอปพลิเคชันมีการโหลดฟอร์มมายังหน่วยความจำ ส่วนเหตุการณ์ Unload จะเกิดขึ้นเมื่อมีการยกเลิกฟอร์มออกจากหน่วยความจำในขณะรันแอปพลิเคชันโดยปกติแล้ว การเขียนโค้ดเพื่อกำหนดค่าเริ่มต้นให้กับตัวแปรต่างๆ ของแอปพลิเคชันนั้นจะเขียนโค้ดลงในโพรซีเยอร์เหตุการณ์ Form_Load เพราะเหตุการณ์ Form_Load มักจะถูกเรียกก่อนเสมอ ส่วนเหตุการณ์ Form_Unload สามารถทำให้เกิดขึ้นได้หลายวิธีเช่น การใช้โค้ดคำสั่ง Unload Me หรือเมื่อเกิดการสิ้นสุดการรันแอปพลิเคชัน เป็นต้น

- Activate และ Deactivate

เหตุการณ์ Form_Activate จะเกิดขึ้นเมื่อฟอร์มถูกกระตุ้น และมีสถานะเป็นฟอร์มแอกทีฟ สำหรับฟอร์มที่จะมีสถานะเป็นฟอร์มแอกทีฟได้ จะต้องเป็นฟอร์มที่กำลังแสดงผลเท่านั้น หรือเป็นฟอร์มที่คุณสมบัติการมองเห็น (Visible) มีค่าเป็นจริง เท่านั้น ซึ่งอาจทำได้โดยใช้เมธอด (Method) SetFocus หรือ Show ของฟอร์มก็ได้ ส่วนเหตุการณ์ Form_Deactivate จะเกิดขึ้นเมื่อฟอร์มสูญเสียสภาวะแอกทีฟให้กับฟอร์มอื่น

- Click และ DblClick

เหตุการณ์ Form_Click จะเกิดขึ้นเมื่อผู้ใช้คลิกเมาส์ ในพื้นที่แสดงผลของฟอร์ม 1 ครั้ง ส่วน Form_DblClick จะเกิดขึ้นเมื่อผู้ใช้คลิกเมาส์ในพื้นที่แสดงผลของฟอร์ม 2 ครั้งติดต่อกัน

- DragDrop และ DragOver

เหตุการณ์ Form_DragDrop จะเกิดขึ้นเมื่อผู้ใช้มีการลากและวาง (Drag and Drop) ออบเจ็กต์จากต้นทางมาวางลงในปลายทาง หรือโดยการใช้โค้ดเมธอด (Method) Drag

เหตุการณ์ Form_DragOver จะเกิดขึ้นเมื่อผู้ใช้มีการลากเมาส์ของขบวน DnD บนฟอร์ม ซึ่งภายในโพรซีเยอร์เหตุการณ์นี้จะทำให้ผู้อ่านสามารถทราบถึงสถานะ DnD ของฟอร์มที่เมาส์ลากผ่านได้ทันที

- KeyUp , KeyDown และ KeyPress

เหตุการณ์ KeyDown เกิดขึ้นเมื่อผู้ใช้มีการกด (press) คีย์ใด ๆ

เหตุการณ์ KeyUp เกิดขึ้นเมื่อผู้ใช้มีการปล่อย (release) คีย์ใด ๆ

เหตุการณ์ KeyUp เกิดขึ้นเมื่อผู้ใช้มีการปล่อย (release) คีย์ใด ๆ

เหตุการณ์ KeyPress เกิดขึ้นเมื่อผู้ใช้มีการกด และปล่อยคีย์ ใด ๆ

เหตุการณ์ทั้งสามนี้จะสามารถเกิดขึ้นกับออบเจ็กต์ฟอร์มได้ ก็ต่อเมื่อฟอร์มนั้น ๆ ไม่มีคอนโทรลใด ๆ เลยที่คุณสมบัติ Enable และ Visible มีค่าเท่ากับ True โดยปกติเราจะใช้เหตุการณ์ KeyDown หรือ KeyUp ในการอ่านค่าสแกนโค้ดของแต่ละปุ่มบนคีย์บอร์ด ซึ่งจะทำให้ผู้อ่านสามารถแยกความแตกต่างระหว่างตัวอักษรพิมพ์ใหญ่กับตัวอักษรพิมพ์เล็กได้ ส่วนเหตุการณ์ KeyPress เราจะใช้ในการอ่านหรือรับตัวอักษร (ANSI Key) ตามปกติ

- MouseDown , MouseMove และ MouseUp

เหตุการณ์ MouseDown เกิดขึ้นเมื่อมีการกดปุ่มของเมาส์ภายในพื้นที่แสดงผลของฟอร์ม

เหตุการณ์ MouseUp เกิดขึ้นเมื่อมีการปล่อยปุ่มของเมาส์ภายในพื้นที่แสดงผลของฟอร์ม

เหตุการณ์ MouseDown เกิดขึ้นเมื่อมีการกดและปล่อยปุ่มของเมาส์ภายในพื้นที่แสดงผลของฟอร์ม

วิธี (Method) ของฟอร์ม

- Circle ใช้วาดรูปวงกลมหรือส่วนของวงกลมลงในฟอร์ม

รูปแบบการใช้งาน :

Object.Circle [Step] (x,y) , radius, [color,strart,end,espect]

Step หมายถึง คำสงวนที่ใช้กำหนดตำแน่งการวางจุดศูนย์กลางของวงกลม (x,y) โดยจะถูกวัดสัมพันธ์กับตำแหน่งโคออร์ดิเนตปัจจุบันของเมาส์

(x,y) หมายถึง เลขที่ใช้กำหนดตำแหน่งโคออร์ดิเนตของจุดศูนย์กลางของวงกลม

color หมายถึง เลขที่ใช้บอกสีของเส้นที่ใช้ในการวาดวงกลม (ForeColor)

radius หมายถึง เลขที่ใช้กำหนดระยะรัศมีของวงกลม

start หมายถึง เลขจำนวนจริงที่ใช้บอกมุมของจุดเริ่มต้นในการวาดส่วนโค้งของวงกลมในหน่วยเรเดียน โดยปกติจะมีค่าเท่ากับ 0 เรเดียน

end หมายถึง เลขจำนวนจริงที่ใช้บอกมุมของจุดเริ่มสิ้นสุดในการวาดส่วนโค้งของวงกลมในหน่วยเรเดียน โดยปกติจะมีค่าเท่ากับ 2π เรเดียน

aspect หมายถึง ข้อมูลเลขจำนวนจริง single ที่ใช้กำหนดอัตราส่วนในการวาดวงรี ซึ่งโดยปกติในการวาดรูปวงกลมค่าของ aspect จะมีค่าเท่ากับ 1.0 เสมอ ดังนั้นถ้าต้องการรูปวงรีก็ให้กำหนดค่าของ aspect ให้เป็นค่าอื่นที่ไม่เท่ากับ 1.0

- Cls ใช้กำหนดให้มีการลบกราฟฟิกหรือความความใด ๆที่ปรากฏในฟอร์ม

- Line ใช้วาดเส้นตรงหรือกล่องสีเหลี่ยมด้วยวิธีกราฟิกลงในพื้นที่แสดงผลของฟอร์ม

รูปแบบการใช้งาน

Object.Line [Step] (x1,y1) [Step] (x2,y2) , [color] , [B] , [F]

(x1,y1) เป็นโคออดิเนตที่ใช้บอกจุดเริ่มต้นของเส้นที่ต้องการวาด

(x2,y2) เป็นโคออดิเนตที่ใช้บอกจุดสิ้นสุดของเส้นที่ต้องการวาด

B หมายถึง กำหนดให้วิธีนี้วาดกล่องสี่เหลี่ยม

F หมายถึง กำหนดให้วิธีนี้วาดกล่องสี่เหลี่ยมและระบายสีที่ถูกกำหนดด้วยคุณสมบัติ FillColor ลงในพื้นที่ของกล่องสี่เหลี่ยมนั้น

- **Refresh** ใช้กำหนดให้มีการปรับปรุงการแสดงผลของข้อมูลของฟอร์มให้ทันสมัยทันที

รูปแบบการใช้งาน

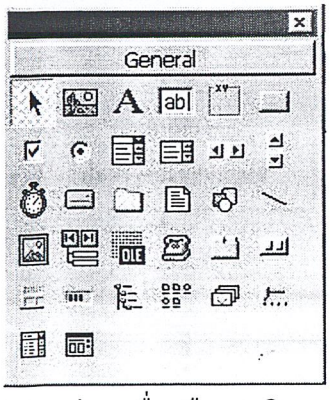
Object.Refresh

วิธี Refresh นี้เป็นวิธีที่มีประโยชน์มาก เพราะเราสามารถที่จะปรับปรุงการแสดงผลข้อมูลของฟอร์ม ให้มีความทันสมัยได้อย่างทันทีทันใด เนื่องจากวินโดวส์มีสภาพแวดล้อมการทำงานแบบมัลติทาสกิงส์ ดังนั้นในบางครั้งข้อมูลที่เราแก้ไขให้กับฟอร์มอาจจะยังไม่ได้รับการปรับปรุงการแสดงผลในทันทีทันใดก็ได้ ดังนั้นผู้เขียนโปรแกรมจึงสามารถใช้วิธี Refresh เพื่อบังคับให้วิชวลเบสิคทำการบริบปรุงการแสดงผลพื้นที่ของฟอร์มทันที ซึ่งจะทำให้ข้อความหรือกราฟฟิกต่าง ๆ ที่ปรากฏในพื้นที่ของฟอร์มจะมีความทันสมัยตลอดเวลา

- **Show** ใช้สั่งให้แสดงผลฟอร์มที่กำหนด

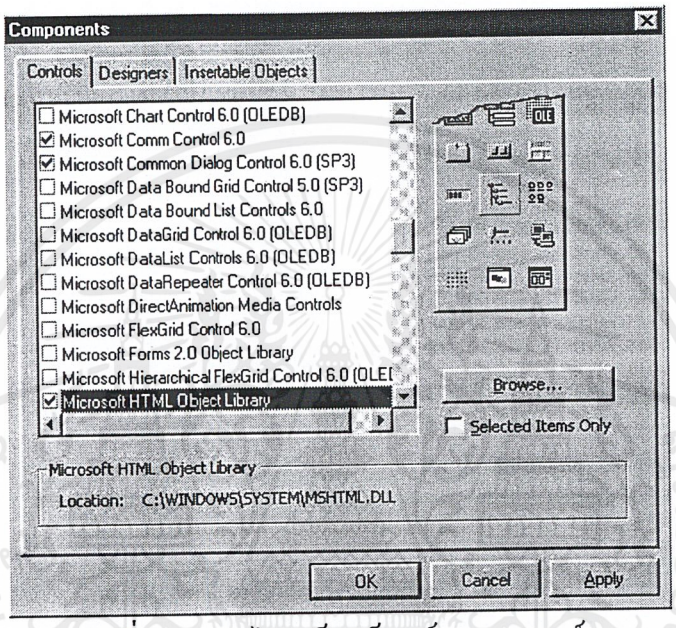
4.3.2 การใช้งานคอนโทรลในวิชวลเบสิค

ในวิชวลเบสิคมีคอนโทรลให้เลือกใช้มากมายเพื่อสนับสนุนการติดต่อสื่อสารระหว่างผู้ใช้กับตัวโปรแกรม คอนโทรลที่แสดงอยู่ในแถบกล่องเครื่องมือนั้นเป็นคอนโทรลภายใน (Intrinsic Control) ที่มีการใช้งานบ่อย ๆ ดังนั้นทุกครั้งที่มีการเปิดวิชวลเบสิค คอนโทรลเหล่านี้ก็จะปรากฏในแถบกล่องเครื่องมือโดยอัตโนมัติ ดังแสดงในรูปที่ 4.2



รูปที่ 4.2 แถบกล่องเครื่องมือของวิชวลเบสิค 6.0

นอกจากคอนโทรลภายในแล้ว วิชวลเบสิคยังมีคอนโทรลมาตราฐานแบบอื่น ๆ อีกที่ใช้สำหรับการออกแบบแอปพลิเคชันที่มีความซับซ้อน ซึ่งคอนโทรลต่าง ๆ นี้เรียกว่า คอนโทรล ActiveX ในการเรียกใช้งานคอนโทรลพิเศษเหล่านี้เราจะต้องนำมันมาเพิ่มในแถบกล่องเครื่องมือเสียก่อน ซึ่งขั้นตอนในการเพิ่มคอนโทรลพิเศษเหล่านี้ทำได้โดยการคลิกที่คำสั่งคอมพาเนนท์ (Components) ของเมนูโปรเจ็ค(Project) จะปรากฏไดอะล็อกบ็อกซ์ คอมพาเนนซ์ ดังรูปที่ 4.3



รูปที่ 4.3 แสดงไดอะล็อกบ็อกซ์คอมพาเนนซ์

จากนั้นก็เลือกคอนโทรลที่ต้องการจากไดอะล็อกบ็อกซ์ โดยคลิกที่กล่องสี่เหลี่ยมหน้าชื่อของคอนโทรล แล้วคลิกปุ่มคำสั่ง O.K. หรือ Apply เพื่อเพิ่มคอนโทรลนั้นลงในแถบเครื่องมือ

ดังที่ได้กล่าวมาแล้วว่าวิชวลเบสิคมีคอนโทรลให้เลือกใช้มากมาย ดังนั้นในที่นี้จะขออธิบายถึงคอนโทรลบางตัวที่ต้องใช้ในการพัฒนาโครงงานโปรแกรมแสดงผลกระบวนการผลิตที่ถูกควบคุมด้วย PLC

4.3.2.1 คอนโทรลภายในทั่วไป

คอนโทรล Image

 เป็นคอนโทรลสำหรับแสดงบิตแมป ไอคอนหรือไฟล์เมต้า หรือแสดงข้อความก็ได้ โดยจะทำหน้าที่คล้ายปุ่มๆหนึ่ง คอนโทรลนี้จะมีความสามารถคล้ายกับคอนโทรล PictureBox แต่มีวิธี(Method)ในการควบคุมน้อยกว่า

คุณสมบัติ(Property) ของคอนโทรล Image ที่ใช้บ่อย ๆ ในการเขียนโปรแกรมกราฟิก คือ

- Picture ใช้ในการกำหนดกราฟิกหรือไฟล์กราฟฟิกที่จะแสดงผลภายในคอนโทรล Image มีรูปแบบการใช้งานดังนี้

Image1.Picture = LoadPicture("c:\windows\256color.bmp")

- **Stretch** ใช้ในการกำหนดให้กล่องรูปภาพ มีการปรับขนาดตามรูปภาพที่แสดงโดยอัตโนมัติ มีรูปแบบการใช้งานดังนี้

Object.Stretch = boolean

ถ้าคุณสมบัติ Stretch มีค่าเป็นจริง หมายถึง ยอมให้ขนาดของกล่องรูปภาพ มีการปรับขนาดตามรูปภาพที่แสดงอยู่โดยอัตโนมัติ แต่ถ้าคุณสมบัติ Stretch มีค่าเป็นเท็จ กล่องรูปภาพจะมีขนาดเท่ากับขนาดที่กำหนดไว้ในหน้าต่างพรอพเพอร์ตี้ ไม่ว่ารูปภาพที่โหลดเข้ามาแสดงจะมีขนาดเท่าใดก็ตาม

คอนโทรล Line

 คอนโทรล Line เป็นคอนโทรลที่ใช้ในการวาดเส้นตรงลงบนฟอร์ม เท่านั้น จึงไม่มีความจำเป็นที่จะต้องมีการตอบสนองต่อเหตุการณ์เกิดขึ้น คุณสมบัติที่สำคัญของคอนโทรล Line คือ คุณสมบัติ BorderStyle ซึ่งใช้กำหนดรูปแบบการแสดงผลของเส้นตรงของคอนโทรล Line ในรูปแบบต่างๆ เช่น เส้นตรงทึบ , เส้นประ , เส้นจุด เป็นต้น

คอนโทรล Shape

 คอนโทรล Shape ใช้ในการวาดกราฟิกรูปสี่เหลี่ยม และวงกลม ให้กับฟอร์ม คุณสมบัติที่สำคัญของคอนโทรล Shape คือ คุณสมบัติ Shape ซึ่งใช้ในการกำหนดรูปทรงของคอนโทรล Shape ว่าจะให้แสดงผลในรูปแบบ สี่เหลี่ยม สี่เหลี่ยมปลายมน วงกลม หรือ วงรี เป็นต้น

คอนโทรล

คอนโทรล Timer

 คอนโทรล Timer เป็นคอนโทรลที่ใช้ในการควบคุมและจัดการเหตุการณ์ด้านเวลา ซึ่งเทียบได้กับประโยค ON TIMER GOTO ของ QuickBASIC โดยผู้เขียนโปรแกรมสามารถเขียนโค้ดเพื่อทำงานใด ๆ เมื่อช่วงเวลาผ่านไปตามค่าที่กำหนด เช่น ทำการปรับปรุงการแสดงผลของฟอร์มทุก ๆ 1 นาที เป็นต้น โดยที่คอนโทรลนี้จะตอบสนองต่อเหตุการณ์เพียงเหตุการณ์เดียวเท่านั้น แต่ผู้เขียนโปรแกรมสามารถกำหนดให้แต่ละฟอร์มมีคอนโทรล Timer มากกว่า 1 คอนโทรลได้ เนื่องจากคอนโทรล Timer เป็นคอนโทรลที่ทำงานตามนาฬิกาของระบบ ดังนั้นมันจึงถูกควบคุมด้วยตัวของระบบเอง

คุณสมบัติที่สำคัญของ Timer ได้แก่

- **Interval** ใช้กำหนดช่วงการจับเวลาให้ Timer ซึ่งมีหน่วยเป็นมิลลิวินาที
- **Enabled** ใช้สั่งให้ Timer เริ่มทำงานหรือหยุดทำงาน

โพซีเยอร์เหตุการณ์ของ Timer มีเหตุการณ์เดียว คือ Timer ซึ่งมีรูปแบบดังนี้

Private Sub Timer1_Temer()

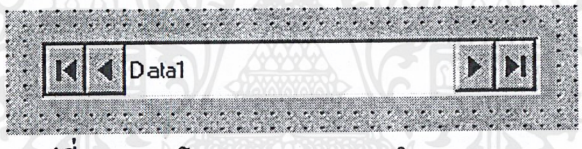
เหตุการณ์ Timer จะเกิดทุกครั้งที่ Timer ทำการนับเวลาครบช่วงเวลาที่กำหนดในคุณสมบัติ Interval ดังนั้นถ้าผู้เขียนโปรแกรมต้องการสั่งงานให้แอปพลิเคชันทำงานเมื่อถึงเวลาที่กำหนดก็สามารถเขียนโค้ดคำสั่งลงในโพซีเยอร์เหตุการณ์นี้

4.3.2.2 คอนโทรลด้านฐานข้อมูล และไดอะล็อกบ็อกซ์

คอนโทรล Data

 คอนโทรล Data เป็นคอนโทรลที่ใช้ในการเข้าถึงฐานข้อมูล และทำการเชื่อมต่อการแสดงผลข้อมูลของแต่ละฟิล์ดในฐานข้อมูลเข้ากับคอนโทรลด้านฐานข้อมูล โดยที่ 1 คอนโทรล Data จะใช้อ้างถึงตารางฐานข้อมูลได้ 1 ตาราง ข้อมูลที่ถูกอ่านจากตารางฐานข้อมูลเข้ามาเก็บไว้ในคอนโทรล Data จะเรียกว่า "Recordset"

ในขณะออกแบบ คอนโทรล Data ที่ถูกวางลงบนฟอร์ม จะมีลักษณะดังรูปที่ 4.4



รูปที่ 4.4 คอนโทรล DATA ขณะทำการออกแบบ

ในการทำงานกับ Data Control จะต้องอาศัยคุณสมบัติ (Properties) ต่าง ๆ ดังนี้

- **Connect** ใช้สำหรับกำหนดประเภทของฐานข้อมูล ในโครงงานนี้ใช้ไมโครซอฟท์ Access เป็นฐานข้อมูลของระบบ ดังนั้นในคุณสมบัตินี้จะต้องกำหนดค่าให้เป็น dBase

- **DatabaseName** ใช้สำหรับกำหนดพาร์ท(Path) และชื่อของฐานข้อมูล(Databese) สำหรับไมโครซอฟท์ Access นี้จะมีพาร์ทเป็น file.MDB

- **RecordsetType** ใช้กำหนดประเภทของ Recordset ซึ่งประกอบด้วย

Table เป็น Recordset ที่ทำงานกับตารางเพียงตารางเดียว

Dynaset เป็น Recordset ที่กระทำกับตารางตั้งแต่ 1 ตารางขึ้นไป Recordset ประเภทนี้จะสามารถแก้ไขข้อมูลได้ โดยข้อมูลที่แก้ไขจะถูกส่งไปยังตารางที่อ้างถึงโดยอัตโนมัติ

Snapshot เป็น Recordset ที่กระทำกับตารางตั้งแต่ 1 ตารางขึ้นไป Recordset ประเภทนี้จะไม่สามารถแก้ไขข้อมูลได้

- **Recordsource** ใช้สำหรับกำหนดชื่อของตารางฐานข้อมูลที่ต้องการติดต่อ

ในการแสดงผลข้อมูลจากฐานข้อมูลที่อ้างถึงนั้น เรากระทำผ่านคอนโทรลที่มีความสามารถในการใช้งานร่วมกับคอนโทรล Data ได้ ซึ่งเรียกว่า Bound Control ตัวอย่างของคอนโทรลเหล่านี้ได้แก่ TextBox , ListBox , Image เป็นต้น

วิธี (Method) ที่ใช้ในการจัดการข้อมูลในฐานข้อมูล โดยทั่วไปจะเกี่ยวข้องกับการเพิ่ม แก้ไข ลบ และค้นหาข้อมูล ซึ่งแต่ละการทำงานจะอาศัยวิธีที่แตกต่างกันไป ได้แก่

recordset.**AddNew** ใช้ในการเพิ่มข้อมูลเข้าไปในตารางข้อมูล

recordset.**Update** ใช้ในการบันทึกข้อมูลที่มีการแก้ไข โดยจะบันทึกข้อมูลที่ปรากฏอยู่ใน Object ที่เป็น Bound Control กลับลงไปยัง Record ปัจจุบันใน Recordset

recordset.**Delete** ใช้ในสำหรับลบข้อมูลใน Record ปัจจุบันออกไปจาก Recordset

recordset.{**MoveFirst** | **MoveLast** | **MoveNext** | **MovePrevious**}ใช้ในการสั่งให้คอนโทรล Data เลื่อนตัวชี้ไปยังเรคอร์ดที่กำหนด

recordset.{**FindFirst** | **FindLast** | **FindNext** | **FindPrevious**} criteria ใช้ในการหาเรคอร์ดที่ต้องการจากประโยคเงื่อนไขในการค้นหา (Criteria)

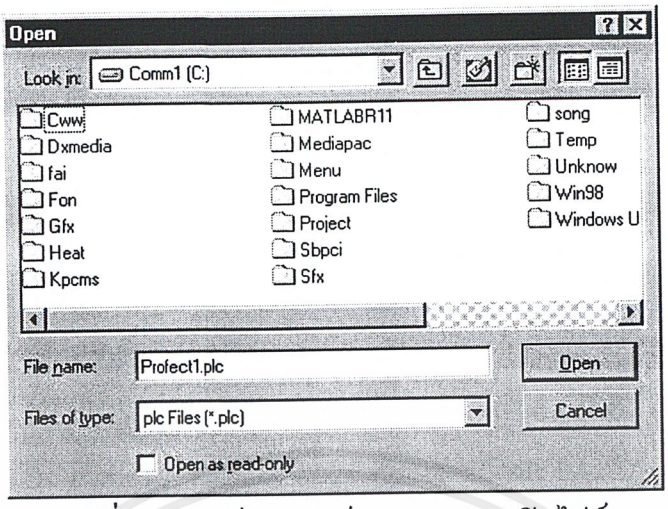
คอนโทรล CommonDialog

 คอนโทรล CommonDialogBox เป็นคอนโทรล ActiveX ตัวหนึ่ง ดังนั้นผู้เขียนโปรแกรมจะต้องทำการเพิ่มเข้าไปในหน้าต่างแถบกล่องเครื่องมือเอง โดยเลือกคอนโทรลชื่อ Microsoft Common Dialog6.0 จากหน้าต่างคอมพาเนนท์ แล้วจึงคลิกO.K.เพื่อทำการเพิ่มคอนโทรลในหน้าต่างคอนโทรลบ็อกซ์

คอนโทรล CommonDialog เป็นคอนโทรลที่แสดงผลไดอะล็อกบ็อกซ์มาตรฐานของวินโดวส์ 95 หลาย ๆ ประเภท เช่น การเปิดไฟล์ การจัดเก็บไฟล์ การจัดพิมพ์ และการตั้งสีและฟอนต์ เป็นต้น ในที่นี้จะอธิบายเฉพาะการใช้งานไดอะล็อกบ็อกซ์สำหรับการเปิด (Open) และ เซฟ (Save) ไฟล์เท่านั้น ซึ่งการที่จะกำหนดให้แสดงไดอะล็อกบ็อกซ์นั้น สามารถกำหนดได้โดยใช้วิธี (Method) ดังนี้

ShowOpen กำหนดให้แสดงไดอะล็อกบ็อกซ์มาตราฐาน Open

ShowSave กำหนดให้แสดงไดอะล็อกบ็อกซ์มาตราฐาน Save



รูปที่ 4.5 ไดอะล็อกบอกซ์มาตรฐาน การเปิดไฟล์

ไดอะล็อกซ์บอกซ์เหล่านี้ ไม่ได้ใช้ในการเปิดหรือเซฟไฟล์จริง ๆ แต่เป็นการให้ผู้ใช้สามารถเลือกชื่อไฟล์หรือสร้างชื่อไฟล์ใหม่ได้ เท่านั้น คุณสมบัติที่สำคัญได้แก่

- **FileName** ใช้ในการกำหนดชื่อของไฟล์ที่ถูกเลือกในขณะรันแอปพลิเคชัน ซึ่งมีรูปแบบการใช้งานคือ

Object.FileName [=String]

ถ้าหากไม่มีการเลือกไฟล์จากไดอะล็อกบ็อกซ์มาตรฐาน คุณสมบัติ FileName ก็จะมีค่าเท่ากับสตริงว่าง แต่ถ้าหากผู้อ่านกำหนดชื่อไฟล์ให้กับคุณสมบัติ FileName ในขณะออกแบบ ชื่อของไฟล์ที่ถูกกำหนดก็จะปรากฏในช่อง File name ของไดอะล็อกบ็อกซ์มาตรฐาน

- **Filter** ใช้ในการกำหนดประเภทหรือนามสกุลของไฟล์ ที่จะแสดงผลในไดอะล็อกบ็อกซ์มาตรฐาน ซึ่งมีรูปแบบการใช้งานดังนี้

Object.Filter [=description1 |filter1 |description2 |filter2…]

เช่น CommonDialog1.Filter = “Bitmap Files(*.bmp)|*.bmp|GIF File(*.gif)|*.gif”

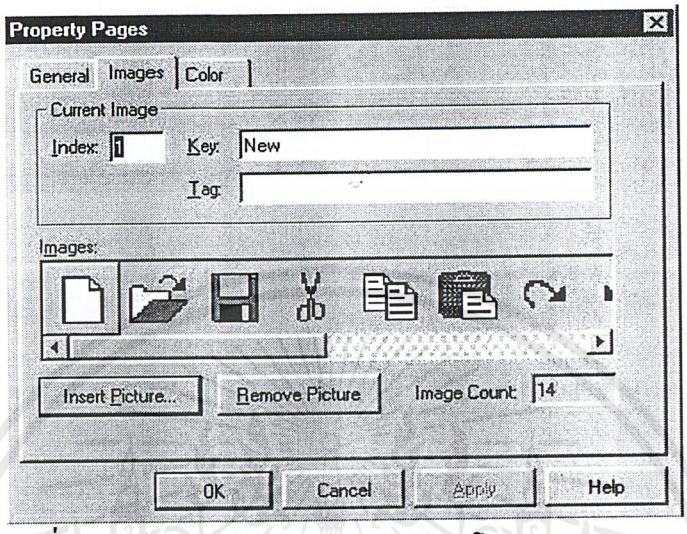
4.3.2.3 คอนโทรลร่วม ImageList และ Toolbar

คอนโทรล ImageList และ Toolbar นี้เป็นคอนโทรล Active X ดังนั้นต้องทำการเพิ่มมันเข้าไปในหน้าต่างคอนโทรลบ็อกซ์ก่อน ซึ่งทำได้โดยการเลือก ชื่อคอนโทรล Microsoft Windows Common Controls 6.0 จากหน้าต่างคอมพาเนนซ์

คอนโทรล ImageList


คอนโทรล Image List เป็นคอนโทรลตัวหนึ่งในชุดของคอนโทรลร่วมวินโดวส์ ที่ทำหน้าที่จัดเก็บอิมเมจสำหรับนำไปใช้งานร่วมกับคอนโทรลอื่น ๆ ที่มีคุณสมบัติ Picture เช่น Image หรือ PictureBox เป็นต้น หรือนำไปใช้งานร่วมกับคอนโทรลร่วมวินโดวส์อื่น ๆ เช่น Toolbar การ

กำหนดคุณสมบัติ และรูปภาพที่จะจัดเก็บในคอนโทรล ImageList นั้นสามารถทำได้โดยการเรียกใช้ Properties Page ของมัน ซึ่งใน Properties Page นี้จะประกอบด้วย 3 แท็บดังรูปที่ 4.6

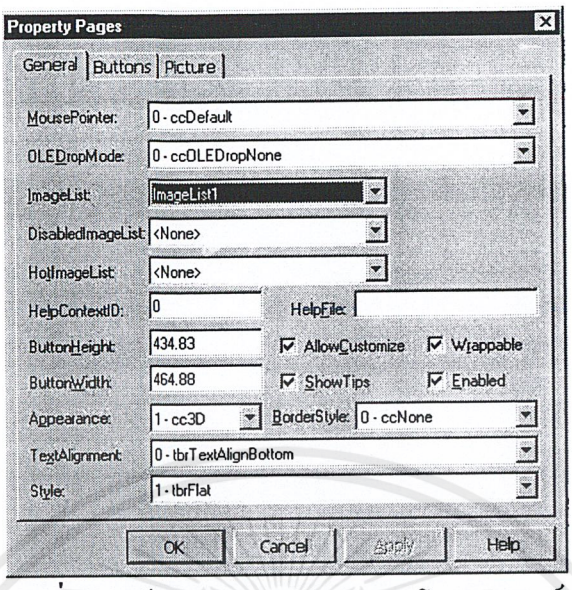


รูปที่ 4.6 หน้าต่าง Property Page ของคอนโทรล ImageList

ออบเจ็กต์ **ListImage** เป็นออบเจ็กต์ที่แสดงเป็นรูปภาพในคอนโทรล ImageList คุณสมบัติของออบเจ็กต์ ListImage เช่น Key , Enable , index , Tag , Value เป็นต้น ซึ่งคุณสมบัติเหล่านี้ใช้ในการอ้างถึงรูปภาพต่าง ๆ ที่แสดงอยู่ในคอนโทรล ImageList นั้นเอง

คอนโทรล Toolbar

คอนโทรลทูลบาร์ เป็นคอนโทรลที่ใช้ในการสร้างตัวบรรจุคำสั่งในรูปแบบของบิตแมป หรือรูปภาพที่เรียกว่า ทูลบาร์ ปุ่มคำสั่งที่อยู่บนทูลบาร์นี้จะถูกกำหนดให้เป็นออบเจ็กต์Button สำหรับการกำหนดบิตแมปให้กับออบเจ็กต์ Button ของคอนโทรลทูลบาร์นั้น เราจะใช้คอนโทรล ImageList เป็นตัวจัดเก็บรูปภาพที่ต้องการแสดงไว้ก่อน จากนั้นจึงทำการอ้างอิงออบเจ็กต์ ListImage ของคอนโทรล ImageList ไปยังแต่ละออบเจ็กต์ Button ในคอนโทรลทูลบาร์ผ่านทางหน้าต่างคุณสมบัติของคอนโทรลทูลบาร์ ดังแสดงในรูปที่ 4.7



รูปที่ 4.7 หน้าต่างคุณสมบัติของคอนโทรลทูลบาร์

คุณสมบัติที่สำคัญของคอนโทรลทูลบาร์ได้แก่

- **ShowTips** ใช้ในการกำหนดความสามารถในการแสดงผล ToolTip ของอ็อบเจ็กต์ในคอนโทรลทูลบาร์ มีรูปแบบการใช้งาน คือ Object.ShowTips = boolean

- **Wrappable** ใช้กำหนดความสามารถในการปรับขนาด หรือเพิ่มแถวของทูลบาร์ เพื่อที่จะให้สามารถแสดงผล Button ได้ทั้งหมด ในขณะที่พื้นที่แสดงผลของฟอร์มถูกสับเปลี่ยน

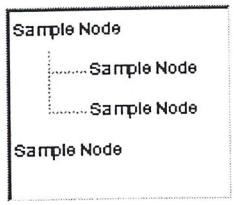
โพรซีเยอร์เหตุการณ์ ที่สำคัญของคอนโทรลทูลบาร์ได้แก่

- Private Sub object_**ButtonClick**(ByVal button As Botton) เป็นเหตุการณ์ที่เกิดขึ้นเมื่อมีการคลิกออบเจ็กต์ Button ของคอนโทรล Toolbar

ออบเจ็กต์ **Button** เป็นออบเจ็กต์ที่แสดงเป็นปุ่มในคอนโทรลทูลบาร์ คุณสมบัติของออบเจ็กต์ Button เช่น Key , Enable , index , Tag , Value เป็นต้น ซึ่งคุณสมบัติเหล่านี้ใช้ในการอ้างถึงปุ่มคำสั่งต่างๆที่แสดงอยู่บนทูลบาร์

4.3.2.4 คอนโทรลร่วม TreeView

คอนโทรลทรีวิว เป็นคอนโทรลที่ทำหน้าที่แสดงผลข้อมูลในรูปแบบของแผนภูมิโครงสร้าง (hierarchical list) ซึ่งแต่ละโหนดที่ปรากฏในแผนภูมิโครงสร้างจะหมายถึงออบเจ็กต์ Node ซึ่งประกอบด้วยข้อความ และรูปภาพ โหนดที่ปรากฏในคอนโทรลทรีวิวจะถูกจัดการในรูปแบบของโหนดแม่ และโหนดลูก (parent-children relation)



รูปที่ 4.8 คอนโทรลทรีวิวขณะออกแบบ

ออบเจ็กต์ Node เป็นออบเจ็กต์ของคอนโทรลทรีวิว ที่ใช้สำหรับแสดงผลข้อความ และรูปภาพของออบเจ็กต์ ListImage โดยที่การอ้างถึงสมาชิกลำดับใด ๆ ในคอนโทรลทรีวิวสามารถทำได้โดยใช้ค่าตัวเลขของคุณสมบัติ Index หรือค่าสตริงของคุณสมบัติคีย์ก็ได้ นอกจากนี้ยังสามารถกำหนดอิมเมจสำหรับ Node ที่อยู่ในสถานะต่างๆ กันได้อีกด้วย

โพรซีเยอร์เหตุการณ์ที่สำคัญเกี่ยวกับคอนโทรลทรีวิว ได้แก่

Expend เป็นเหตุการณ์ที่เกิดเมื่อออบเจ็กต์โหนดในคอนโทรลทรีวิวถูกขยาย เพื่อให้แสดงผลโหนดลูกภายใน เหตุการณ์ Expend นี้จะเกิดก่อนเหตุการณ์ Click เสมอ

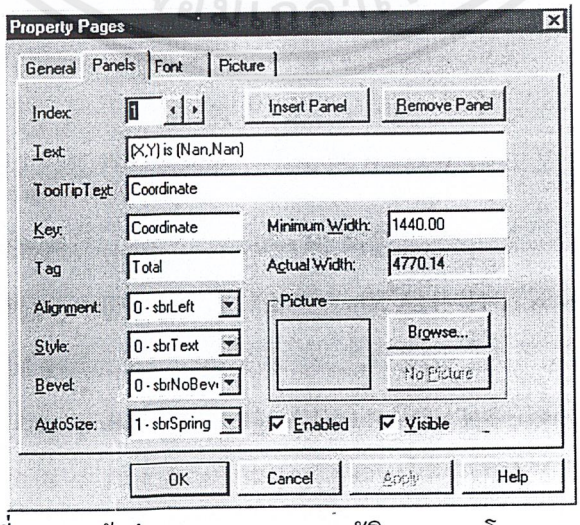
NodeClick เป็นเหตุการณ์ที่เกิดเมื่อออบเจ็กต์โหนดในคอนโทรลทรีวิวถูกคลิก

4.3.2.5 คอนโทรลร่วม StatusBar

 คอนโทรล StatusBar เป็นคอนโทรลที่ใช้ในการแสดงผลข้อมูลหรือสถานะต่าง ๆ ภายในพื้นที่ของตัวเอง (Panel)

รูปที่ 4.9 แถบสถานะ (Status Bar)

การสร้างพาเนล (Panel) ผู้เขียนโปรแกรมสามารถทำได้ขณะออกแบบโดยใช้หน้าต่างคุณสมบัติของคอนโทรล StatusBar หรือ อ้างถึงออบเจ็กต์พาเนล (panel) ในขณะเขียนโปรแกรมก็ได้



รูปที่ 4.10 หน้าต่างออกแบบคุณสมบัติของคอนโทรล Status

การอ้างถึงพาเนลใด ๆ ในแถบสถานะทำได้โดยการเขียนโปรแกรมโดยใช้ออบเจ็กต์ Panel ของคอนโทรล Status เช่น ถ้าต้องการเขียนสตริง"(X,Y) is (NaN,NaN)"ลงในแถบสถานะในพาแนลที่ 1 สามารถเขียนคำสั่งได้ ดังนี้

Me.StatusBar1.**Panel(1)**.Text = "(X,Y) is (NaN,NaN)"

4.2.3.6 คอนโทรล MSComm

 คอนโทรล MSComm (Communication) เป็นคอนโทรล ActiveX ตัวหนึ่ง ดังนั้นจึงต้องทำการเพิ่มเข้าไปในคอนโทรลทูลบ็อกซ์ก่อนโดยเลือกชื่ออุปกรณ์ MSComm Control 6.0 ในหน้าต่างคอมพาเนนท์ MSComm นี้ช่วยในการติดต่อสื่อสารกกับพอร์ตอนุกรม ในโปรแกรมแสดงผลกระบวนการผลิตนี้จะใช้คอนโทรล MSComm ในการเปิดพอร์ตติดต่อสื่อสารแบบอนุกรมเมื่อปุ่มคำสั่ง Link ถูกกด

คุณสมบัติที่สำคัญที่ใช้ในการตั้งค่าเริ่มต้นการสื่อสารตามมาตรฐานการสื่อสารแบบอนุกรม และ ใช้ในการเปิด ปิด พอร์ตได้แก่

- object.**CommPort** = value ใช้ในการกำหนดหมายเลขพอร์ตอนุกรมของเครื่องที่ต้องการติดต่อ ซึ่งจะต้องกำหนดหมายเลขพอร์ตให้กับคุณสมบัติ CommPort ก่อนที่จะทำการเปิดพอร์ตเพื่อการสื่อสาร

- object.**PortOpen** = boolean ใช้ในการเปิดปิดพอร์ตเพื่อการใช้งาน

- object.**Setting** = value ใช้ในการตั้งค่าเริ่มต้นของมาตรฐานการสื่อสารแบบอนุกรม ซึ่งค่าที่กำหนดให้คุณสมบัติ Settings นี้ได้แก่ อัตราบอร์ด , จำนวนบิตพาริตี้ , จำนวนของข้อบิตข้อมูล และจำนวนของบิตจบ ซึ่งเราจะต้องเรียงลำดับค่าต่างดังนี้ บอร์ด,บิตพาริตี้,บิตข้อมูล,บิตจบ ตัวอย่าง เช่น ถ้าต้องการกำหนดอัตราบอร์ดเป็น 9600 บอร์ด ,พาริตี้คู่ , 7 บิตข้อมูล และ 2 บิตจบ เราสามารถเขียนโปรแกรมได้ดังนี้

Me.MSComm1.Setting = "9600,E,7,2"

- object.**Input** = value ใช้ในการอ่านค่าต่างๆ ที่อยู่ในบัฟเฟอร์รับข้อมูลของคอมพิวเตอร์

- object.**Output** = value ใช้ในการส่งค่าที่ต้องการออกไปยังบัฟเฟอร์ส่งข้อมูล

- object.**InBufferSize** และ object.**OutBufferSize** ใช้ในการกำหนดขนาดของบัฟเฟอร์รับข้อมูล และบัฟเฟอร์ส่งข้อมูล

-object.**InBufferCount** และ object.**OutBufferCount** ใช้ในการกำหนด หรือรายงานจำนวนข้อมูลที่อยู่ในบัฟเฟอร์รับข้อมูล หรือบัฟเฟอร์ส่งข้อมูลตามลำดับ

บทที่ 5
การออกแบบและเขียนโปรแกรมแสดงผลกระบวนการผลิต

5.1 แนวความคิดเกี่ยวกับโปรแกรมกราฟฟิกแสดงผลกระบวนการผลิต

โปรแกรมแสดงผลกระบวนการผลิตที่ควบคุมด้วย PLC นี้ จะเป็นโปรแกรมที่สามารถแสดงภาพกราฟิกเพื่อใช้ในการแสดงผลกระบวนการผลิต โดยภาพกราฟิกเหล่านี้จะต้องมีการเปลี่ยนแปลงที่สัมพันธ์กันกับการเปลี่ยนแปลงของรีเลย์เอาท์พุตของเครื่อง PLC ดังนั้นจึงควรจะมีหน้าต่างที่ใช้สำหรับแสดงสถานะของอุปกรณ์ตัวนั้นให้ผู้ใช้ได้เห็นได้โดยง่ายอีกด้วย และโปรแกรมนี้ควรจะเป็นโปรแกรมสำเร็จรูปที่มีการจัดหาอุปกรณ์ต่างๆในกระบวนการผลิตไว้ให้ผู้ใช้เรียบร้อยแล้ว โดยสามารถดึงมาใช้ได้โดยง่าย นอกจากนี้ยังควรที่จะต้องมีเมนู และปุ่มต่าง ๆ ที่อำนวยความสะดวกในการทำงานกับโปรแกรมให้กับผู้ใช้ตามมาตรฐานของโปรแกรมที่ทำงานบนวินโดวส์ทั่วๆ ไป

5.2 ขั้นตอนในการดำเนินงาน

หลังจากที่ทราบความต้องการต่างๆ ของผู้ใช้ และได้วิเคราะห์และออกแบบระบบเป็นที่เรียบร้อยแล้ว ขั้นตอนการพัฒนาโปรแกรมด้วยวิชวลเบสิคจะประกอบด้วย

-การสร้างจอภาพของโปรแกรมต่าง ๆ เป็นขั้นตอนในการออกแบบ และสร้างจอภาพที่จะใช้สื่อสารกับผู้ใช้ ซึ่งสามารถทำได้โดยวาดคอนโทรลต่าง ๆ ลงบนฟอร์มโดยตรง

-การออกแบบฐานข้อมูล ในที่นี้จะใช้ไฟล์นามสกุล MDB เป็นตัวฐานข้อมูล ดังนั้นจึงสามารถใช้ไมโครซอฟท์เอ็กเซส (Microsoft Access) เป็นโปรแกรมช่วยในการสร้างฐานข้อมูลได้

-การเขียนโปรแกรมเพื่อควบคุมการทำงานของแต่ละออบเจ็กต์บนจอภาพ เป็นขั้นตอนที่สำคัญและยากที่สุด ในการที่จะสั่งใช้โปรแกรมที่ออกแบบไว้ทำงานได้ตามความประสงค์ นอกจากนี้ยังต้องให้ทำงานสัมพันธ์กับฟอร์มอื่น ๆ ในโปรเจคท์อีกด้วย ซึ่งจะต้องอาศัยการศึกษา และประสบการณ์ในการเขียนโปรแกรมอย่างมากเพื่อที่จะให้โปรแกรมทำงานได้อย่างมีประสิทธิภาพ

5.3 ภาพกราฟิก ที่ใช้สำหรับออกแบบกระบวนการผลิต

ส่วนสำคัญส่วนหนึ่งของโปรแกรมกราฟิกที่ใช้จำลองกระบวนการผลิต คือรูปภาพอุปกรณ์ต่าง ๆ ที่จะใช้เป็นภาพจำลองกระบวนการผลิต ซึ่งจะต้องออกแบบให้มีจำนวนมากพอที่จะทำให้เกิดความสะดวกในการเรียกใช้งาน สำหรับภาพอุปกรณ์ในโปรแกรมจำลองกระบวนการผลิตนี้ ผู้เขียนโปรแกรมได้ใช้โปรแกรม Paint สร้างขึ้นมาโดยจะมีนามสกุล .GIF อุปกรณ์ต่าง ๆ ในโปรแกรม

แสดงผลกระบวนการผลิตได้แสดงรูปภาพไว้แล้วในภาคผนวก ซึ่งสามารถจำแนกออกได้เป็น 6 ประเภท ดังนี้

5.3.1 อุปกรณ์ควบคุมทั่วไป (General Item) เป็นรูปที่แสดงอุปกรณ์ที่มีการใช้งานในระบบควบคุมทางอุตสาหกรรมโดยทั่วๆไป โดยอุปกรณ์เหล่านี้จะสามารถทำการปิด เปิด ได้ตามการออกแบบ หรือตามคำสั่งจากเครื่อง PLC สำหรับโปรแกรมจำลองกระบวนการผลิตนี้ได้ออกแบบอุปกรณ์ทั่วไปเหล่านี้ ไว้ทั้งสิ้น 14 ตัวด้วยกัน เช่น Blower , Motor , Pump , Lamp ,Turbine เป็นต้น

5.3.2 อุปกรณ์เก็บวัตถุดิบ (Storage Item) เป็นรูปที่แสดงอุปกรณ์เก็บของเหลว หรือก๊าซที่ใช้ในกระบวนการผลิต อุปกรณ์เหล่านี้ไม่ได้ถูกให้ควบคุมด้วย PLC ภาพแสดงอุปกรณ์เหล่านี้มีทั้งสิ้น 9 ชิ้นคือ เช่น Vessel , Bin , Tower , Pressure Storage เป็นต้น

5.3.3 อุปกรณ์แลกเปลี่ยนความร้อน (Exchanger Item) เป็นรูปแสดงอุปกรณ์ที่เกี่ยวข้องกับการแลกเปลี่ยนความร้อน มีทั้งสิ้น 5 ชิ้น ได้แก่ Exchanger , Evaporator , Cooling Tower , Scriber และ Spay Dryer

5.3.4 ท่อแบบต่าง ๆ (Pipe Item) เป็นรูปที่แสดงท่อแบบต่าง ๆ ที่ใช้ในการวาดรูปจำลองกระบวนการผลิต ได้แก่ PipeDirect , PipeL ,PipeT และ PipeFourWay

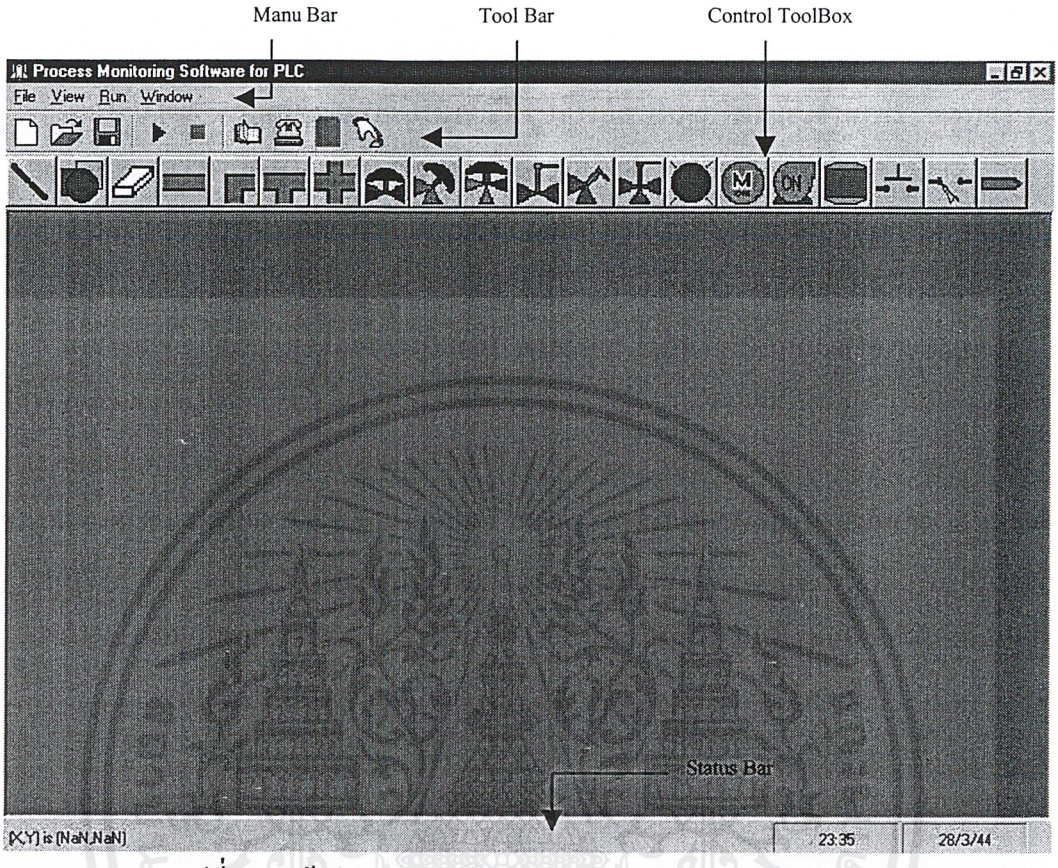
5.3.5 วาล์วแบบต่าง ๆ (Valve Item) เป็นรูปแสดงวาล์วแบบต่าง ๆ ที่ใช้ในกระบวนการผลิต ซึ่งแบ่งเป็น คอนโทรลวาล์ว เช่นวาล์วแบบปกติเปิด , วาล์วแบบปกติปิด , วาล์วหักมุม และวาล์วสามทาง วาล์วเหล่านี้สามารถควบคุมการปิดเปิดได้จากเครื่อง PLC นอกจากนี้ยังมีรูปภาพที่ใช้แสดงวาล์วที่ปิดเปิดด้วยมือ (Manual Valve) ด้วย

5.3.6 อุปกรณ์สวิตช์ (Switching Item) อุปกรณ์ประเภทนี้ได้แก่ ปุ่มกด , ลิมิตสวิตช์ และ ตัวตรวจวัด(Sensor) ซึ่งอุปกรณ์เหล่านี้จะใช้ในการสั่งการทำงานของกระบวนการผลิตผ่านทางเครื่อง PLC

5.4 หน้าต่างสำหรับติดต่อกับผู้ใช้ และ การทำงาน

สำหรับโปรแกรมจำลองกระบวนการผลิตที่ถูกควบคุมด้วย PLC นี้ ได้ทำการออกแบบให้มีs หน้าต่างสำหรับติดต่อกับผู้ใช้ทั้งหมด 7 หน้าต่างซึ่งแต่ละหน้าต่างมีหน้าที่และความสัมพันธ์กัน ดังจะอธิบายต่อไปนี้

5.4.1 หน้าต่างหลัก (Main Windows)



รูปที่ 5.1 หน้าต่างหลักของโปรแกรมจำลองกระบวนการผลิต

หน้าต่างหลักนี้เป็นฟอร์มชนิด MDI ซึ่งจะใช้เป็นฟอร์มหลักที่ใช้ติดต่อกับผู้ใช้และใช้ในการเรียกฟอร์มย่อยอื่น ๆ ส่วนประกอบต่าง ๆ ของหน้าต่างหลักของโปรแกรมจำลองกระบวนการผลิตประกอบด้วยส่วนหลัก ๆ 4 ส่วนด้วยกันคือ เมนูบาร์ , ทูลบาร์ , คอนโทรล ทูลบ็อกซ์ และสเตตัสบาร์ แต่ละส่วนสามารถอธิบายได้ดังนี้

เมนูบาร์ (Manu Bar)

เป็นส่วนที่เก็บคำสั่งทั่วไปของโปรแกรม ซึ่งจะประกอบด้วย เมนูหลัก 4 เมนูด้วยกันคือ

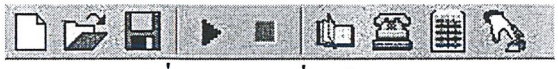
-เมนู File ประกอบด้วยคำสั่ง New (ใช้สำหรับสร้างหน้าต่างออกแบบอันใหม่) , คำสั่ง Save (ใช้ในการเซฟหน้าต่างออกแบบที่ทำการออกแบบไว้แล้ว) ,คำสั่งOpen (ใช้สำหรับเปิดหน้าต่างออกแบบที่เซฟไว้) และคำสั่ง Exit (ใช้เลิกการทำงานของโปรแกรม)

-เมนู View เป็นเมนูที่ใช้ในการเรียกดูหน้าต่างคุณสมบัติ หน้าต่างไลบารี่ หน้าต่างสถานะการติดต่อสื่อสาร และหน้าต่างเครื่องป้อนโปรแกรมแลดเดอร์

-เมนู Run เป็นเมนูที่เก็บคำสั่งในการเปิด หรือปิดพอร์ท ในการติดต่อสื่อสารกับ PLC ซึ่งได้แก่คำสั่ง Link และ Stop

-เมนู Windows เป็นเมนูที่เก็บคำสั่งดำเนินการเกี่ยวกับการจัดรูปแบบหน้าต่างออกแบบ ในกรณีที่มีหน้าต่างออกแบบหลายหน้าต่าง ได้แก่คำสั่ง Cascade , Vertical , Herizotal และคำสั่งในการปิดหน้าต่างออกแบบ

แถบเครื่องมือ (Toolbar)



รูปที่ 5.2 แถบเครื่องมือ (Toolbar)

แถบเครื่องมือเป็นปุ่มคำสั่งที่อำนวยความสะดวกให้กับผู้ใช้ เพื่อให้ผู้ใช้สามารถเรียกใช้คำสั่งที่ใช้บ่อย ๆ ได้อย่างสะดวก แถบเครื่องมือของโปรแกรมจำลองกระบวนการผลิตประกอบด้วยปุ่มคำสั่ง New , Open , Save , Run , Stop , Library , Communication , Properties และ Console ตามลำดับดังรูปที่ 5.2

กล่องเครื่องมือ (Control ToolBox)

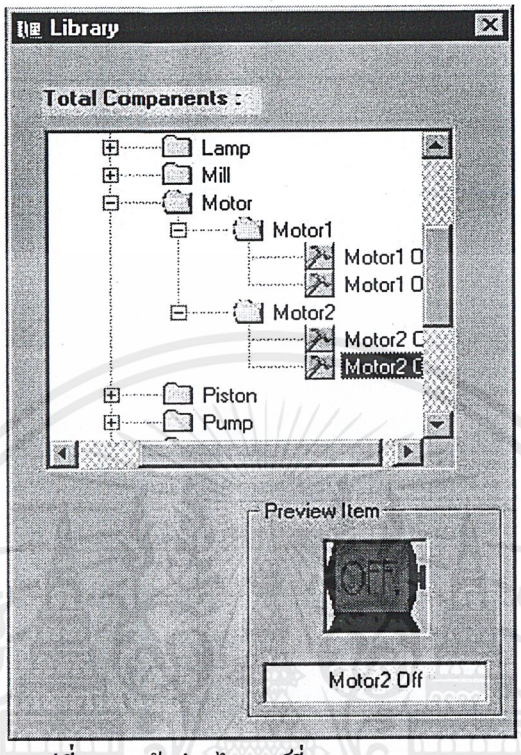
รูปที่ 5.3 กล่องเครื่องมือ (Control ToolBox)

กล่องเครื่องมือ เป็นแถบทูลบาร์ที่รวบรวมเอาอุปกรณ์ที่ใช้บ่อย ๆ ในการออกแบบหรือจำลองกระบวนการผลิตเข้าไว้ด้วยกัน เช่น วาล์วแบบต่าง ๆ ,หลอดไฟ ,มอเตอร์ ,ลูกสูบ ,ปุ่มกด เป็นต้น นอกจากนี้ยังมีคอนโทรลที่ใช้ในการวาดภาพอิสระในการออกแบบด้วยคือ Line , Shape และ Eraser การใช้งานกล่องเครื่องมือในการออกแบบทำได้โดยการคลิกเลือกอุปกรณ์ที่ต้องการจะวาดในกล่องเครื่องมือ แล้ววาดลงในหน้าต่างออกแบบด้วยวิธีลากแล้วปล่อย (DnD)

แถบสถานะ (Status Bar)

เป็นแถบที่แสดงสถานะปัจจุบันของโปรแกรมจำลองกระบวนการผลิต เช่น วันเดือนปี และเวลาปัจจุบัน นอกจากนี้ยังแสดงจำนวนอุปกรณ์การออกแบบที่ถูกวาดลงในหน้าต่างออกแบบในขณะนั้น และแสดงตำแหน่งปัจจุบันของเมาส์ด้วย

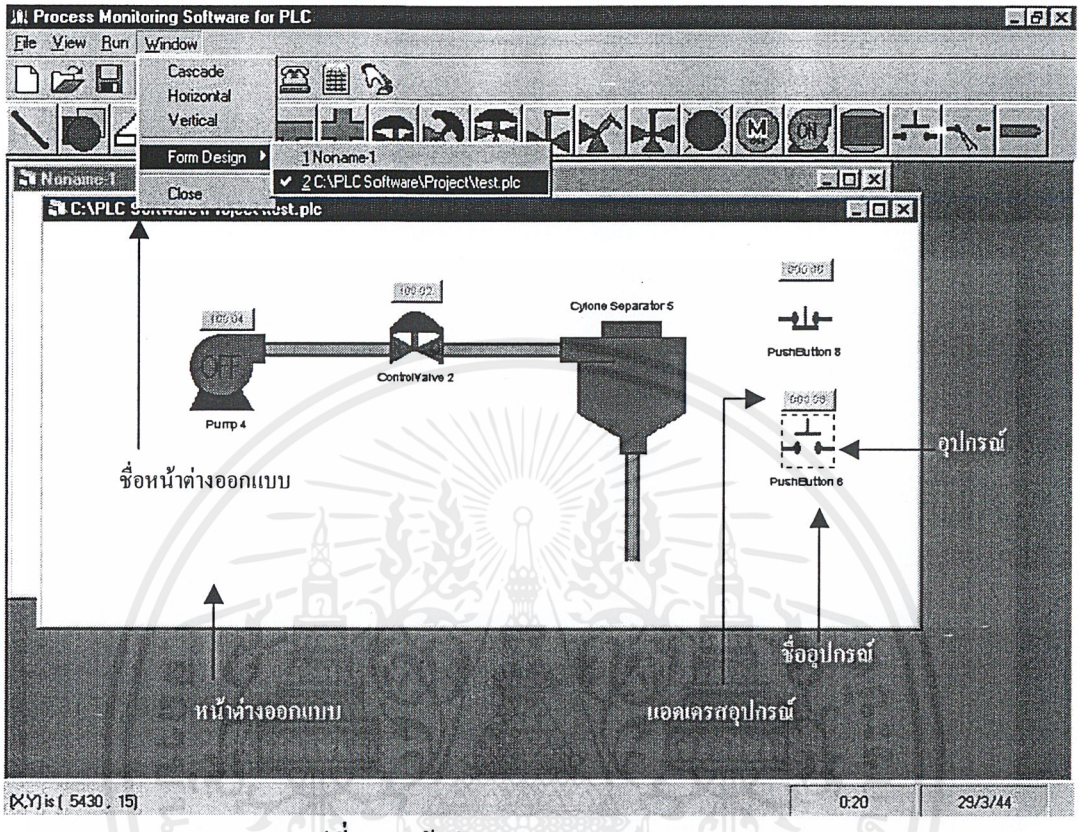
5.4.2 หน้าต่างไลบารี่ (Library Windows)



รูปที่ 5.4 หน้าต่างไลบารี่ (Library Window)

หน้าต่างไลบารี่ เป็นหน้าต่างที่รวบรวมอุปกรณ์ (Item) ต่าง ๆ ที่จัดเตรียมไว้สำหรับวาดภาพกราฟิกจำลองกระบวนการผลิตทั้งหมดไว้ในหน้าต่างเดียว เปรียบเสมือนเป็นห้องสมุดอุปกรณ์ของโปรแกรม ซึ่งผู้ใช้สามารถเข้ามาเลือกอุปกรณ์ที่ต้องการนำไปสร้างกระบวนการผลิตจำลองได้ทันที การวาดภาพกราฟิกโดยใช้อุปกรณ์ในหน้าต่างไลบารี่ ทำได้โดยคลิกเลือกอุปกรณ์ที่ต้องการ ซึ่งจะปรากฏภาพตัวอย่างของอุปกรณ์นั้นในช่อง Preview Item จากนั้นผู้ใช้ก็สามารถที่จะคลิกที่ภาพตัวอย่างนั้นแล้วนำไปวางบนหน้าต่างออกแบบได้ทันที

5.4.3 หน้าต่างออกแบบ (Form Design)

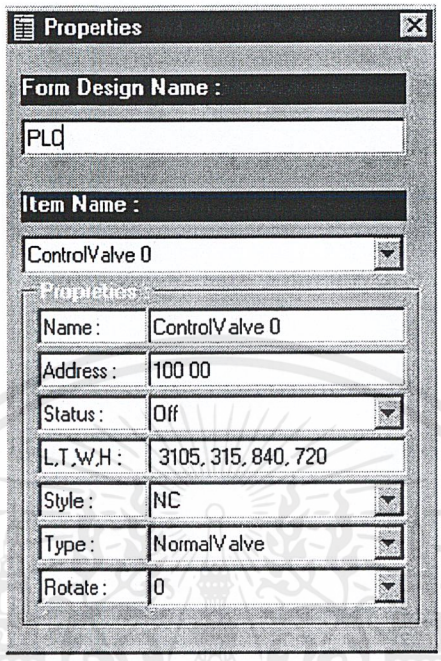


รูปที่ 5.5 หน้าต่างออกแบบ (Form Design)

หน้าต่างออกแบบนี้ สามารถเรียกได้โดยใช้คำสั่ง New จากเมนูไฟล์ หรือคลิกปุ่ม New จากทูลบาร์ หน้าต่างออกแบบนี้มีไว้สำหรับทำการวาดภาพกราฟิกจำลองกระบวนการผลิต โดยภาพอุปกรณ์ต่างๆ นั้นสามารถหาได้จากแถบกล่องเครื่องมือ หรือดึงมาจากหน้าต่างไลเบอร์รี่ก็ได้ ฟอร์มสำหรับออกแบบนี้ สามารถเปิดขึ้นมาได้ที่เดียวหลาย ๆ ฟอร์ม เพื่อความสะดวกในการออกแบบกระบวนการที่มีหลายลักษณะ

สำหรับภาพอุปกรณ์ที่แสดงบนหน้าต่างออกแบบนั้น เมื่อวาดลงบนฟอร์มออกแบบแล้ว จะมีบล็อกแสดงชื่อ หรือ บล็อกแสดงแอดเดรสของอุปกรณ์นั้นขึ้นมาด้วยโดยอัตโนมัติ ซึ่งชื่อ หรือแอดเดรสเหล่านี้สามารถแก้ไขได้โดยทำการแก้ไขคุณสมบัติของมันจากหน้าต่างคุณสมบัติ

5.4.4 หน้าต่างคุณสมบัติ (Properties Windows)



รูปที่ 5.6 หน้าต่างคุณสมบัติ (Properties Window)

หน้าต่างคุณสมบัติ เป็นหน้าต่างที่แสดงคุณสมบัติของอุปกรณ์ที่อยู่ในหน้าต่างสำหรับออกแบบทั้งหมด โดยในหน้าต่างนี้จะประกอบด้วย ชื่อของฟอร์มการออกแบบ ชื่อของอุปกรณ์ที่อยู่ในฟอร์มการออกแบบนั้น และคุณสมบัติต่าง ๆ ของมัน ซึ่งคุณสมบัติพื้นฐานของอุปกรณ์ในโปรแกรมจำลองกระบวนการผลิตนี้ถูกกำหนดให้มี 7 คุณสมบัติ ได้แก่

Name : ใช้กำหนด หรือตั้งชื่อของอุปกรณ์

Address : ใช้กำหนด หรือแก้ไขแอดเดรสของอุปกรณ์นั้น ซึ่งแอดเดรสนี้จะถูกนำไปติดต่อกับเครื่อง PLC เมื่อมีการติดต่อสื่อสารเกิดขึ้น

Status : ใช้กำหนด หรือแสดงสถานะการเปิดปิดของอุปกรณ์ในขณะนั้น

L,T,W,H : ใช้กำหนด หรือแสดงขนาดของภาพอุปกรณ์ที่แสดงในฟอร์มการออกแบบ ในการแก้ไขจะต้องกำหนดค่าเรียงตามลำดับ Left ,Top ,Width และ Height

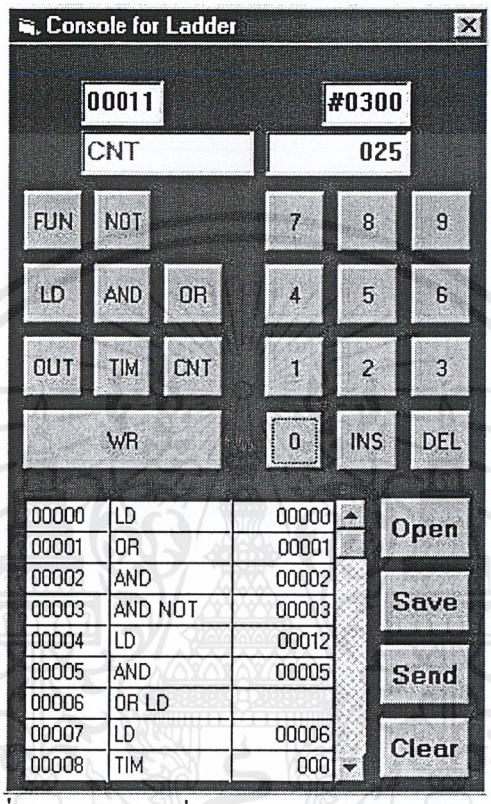
Style : ใช้กำหนด หรือแสดงรูปแบบของอุปกรณ์นั้น ๆ สำหรับคุณสมบัตินี้จะมีเฉพาะในบางอุปกรณ์เท่านั้น

Type : ใช้กำหนด หรือ แสดงชนิดของอุปกรณ์นั้น ๆ ในกรณีมีหลายชนิด ซึ่งคุณสมบัตินี้จะมีเฉพาะในบางอุปกรณ์เท่านั้น

Rotate : ใช้ในการหมุนอุปกรณ์ ในขณะที่อุปกรณ์อยู่ในหน้าต่างออกแบบ สามารถหมุนได้ 90 , 180 หรือ 270 องศา

การแก้ค่าใด ๆในหน้าต่างคุณสมบัติสามารถทำได้เมื่อยังไม่มีการติดต่อสื่อสารกับ PLC เกิดขึ้นเท่านั้น และค่าที่แก้ไข จะส่งผลต่อลักษณะของอุปกรณ์นั้นในหน้าต่างการออกแบบทันที

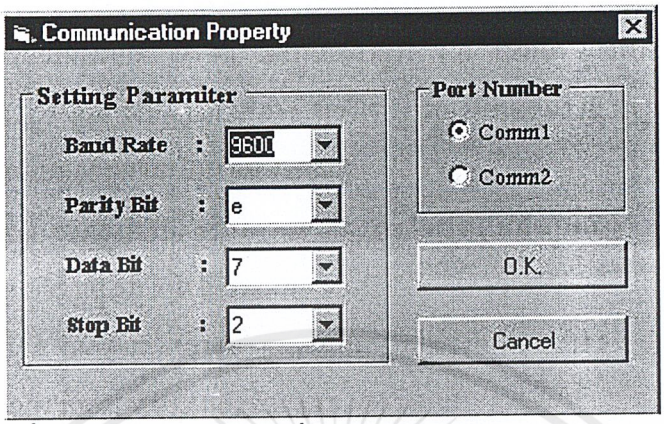
5.4.5 หน้าต่างเครื่องมือป้อนโปรแกรมแลดเดอร์ (Console Window)



รูปที่ 5.7 หน้าต่างเครื่องมือป้อนโปรแกรมแลดเดอร์

หน้าต่างเครื่องมือป้อนโปรแกรมแลดเดอร์ เป็นหน้าต่างที่จำลองลักษณะและการทำงานของเครื่องมือป้อนโปรแกรมแลดเดอร์ (Console) ของเครื่อง PLC หน้าต่างนี้จะใช้ในการป้อนโปรแกรมแลดเดอร์ให้กับเครื่อง PLC โดยตรงผ่านทางหน้าจอคอมพิวเตอร์ ซึ่งการใช้งานหน้าต่างนี้ทำได้โดยการกดปุ่มที่ใช้แทนลอจิกต่างๆ และกดตำแหน่งแอดเดรสของรีเลย์ที่ต้องการใช้ในแต่ละคำสั่ง โดยทุกคำสั่งที่เขียนไว้แล้วจะแสดงให้เห็นในตารางแสดงข้อความด้านล่างดังรูปที่ 5.7 เมื่อทำการเขียนโปรแกรมแลดเดอร์เรียบร้อยแล้ว ก็สามารถทำการบันทึกเก็บเอาไว้ได้โดยกดปุ่ม Save และเมื่อต้องการใช้ก็กด Open แล้วเลือกโปรแกรมที่ต้องการ เราสามารถป้อนโปรแกรมแลดเดอร์ที่เขียนไว้แล้วให้กับเครื่อง PLC ได้โดยกดปุ่มคำสั่ง Send เมื่อเครื่อง PLC ได้บันทึกโปรแกรมลงหน่วยความจำแล้ว ก็จะมีไดอะล็อกบอกซ์แสดงข้อความ “Send already” ขึ้นเพื่อบอกผู้ใช้ว่าเครื่อง PLC ได้รับคำสั่งแล้ว

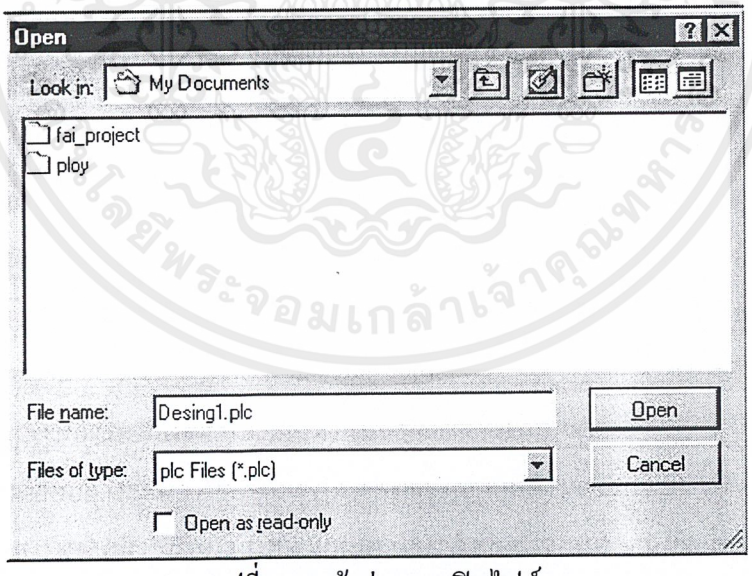
5.4.6 หน้าต่างการติดต่อสื่อสาร (Communication Window)



รูปที่ 5.8 หน้าต่างการติดต่อสื่อสาร (Communication Window)

หน้าต่างการติดต่อสื่อสารถูกออกแบบมาเพื่ออำนวยความสะดวกแก่ผู้ใช้ในการเลือกค่าพารามิเตอร์สำหรับการสื่อสารตามมาตรฐานการสื่อสารแบบอนุกรม RS-232C ให้ตรงกับค่าที่ตั้งไว้ในเครื่อง PLC แต่ละเครื่อง

5.4.7 หน้าต่างการเปิด และการบันทึกไฟล์ (Common Dialog Open / Save)



รูปที่ 5.9 หน้าต่างการเปิดไฟล์

หน้าต่างการเปิดไฟล์ และหน้าต่างการบันทึกไฟล์ เป็นหน้าต่างมาตรฐานของระบบปฏิบัติการวินโดวส์ ซึ่งสามารถเรียกใช้ผ่านคอนโทรล Common Dialog สำหรับโปรแกรมแสดงผลกระบวนการผลิตนี้ สามารถบันทึกฟอร์มที่ได้ทำการออกแบบไว้แล้วได้ โดยไฟล์ที่บันทึกนี้จะมี

นามสกุล .PLC โดยใช้คำสั่ง Save จากเมนูไฟล์ และสามารถเปิดไฟล์ที่บันทึกเพื่อทำการแก้ไขได้ โดยใช้คำสั่ง Open จากเมนูไฟล์เช่นกัน

5.5 โครงสร้างฐานข้อมูลของโปรแกรมแสดงผลกระบวนการผลิต

โครงงานโปรแกรมแสดงผลกระบวนการผลิตนี้ ใช้โปรแกรมไมโครซอฟท์แอกเซส (Microsoft Access) ในการสร้างฐานข้อมูลเกี่ยวกับอุปกรณ์ และคุณสมบัติของอุปกรณ์ที่ใช้ภายในโปรแกรม โดยจะจัดทำเป็นตารางข้อมูล ซึ่งจะจัดเก็บอยู่ในไฟล์นามสกุล .MDB ไฟล์นี้จะถูกเรียกใช้โดยคอนโทรล Data ที่ออกแบบไว้ในตัวโปรแกรมหลักของโปรแกรมแสดงผลกระบวนการผลิต ฐานข้อมูลที่ออกแบบให้กับโปรแกรมแสดงผลกระบวนการผลิตนี้จะแบ่งออกเป็น 3 ตาราง คือ ตารางข้อมูลอุปกรณ์ทั้งหมดในโปรแกรม ตารางข้อมูลอุปกรณ์ที่อยู่ในหน้าต่างออกแบบในขณะโปรแกรมกำลังทำงาน และตารางข้อมูลอุปกรณ์ และฟอร์มที่ถูกบันทึก ซึ่งรายละเอียดของแต่ละตารางอธิบายได้ดังนี้

5.51 ฐานข้อมูล 1 ตารางข้อมูลอุปกรณ์รวม (Total Component Table)

TotalItem : ตาราง

ID	KeyItem	NameItem	Status	TypeItem	Style	Rotate
69	100	Pipe	1	PipeT	1	90
70	101	Pipe	1	PipeT	1	180
71	102	Pipe	1	PipeT	1	270
72	103	Pipe	1	PipeFour	1	1
73	104	Fan	1	1	1	1
74	108	ControlValve	On	NormalValve	NO	0
75	109	ControlValve	On	NormalValve	NO	90
76	110	ControlValve	On	NormalValve	NO	270
77	111	ControlValve	Off	NormalValve	NO	0
78	112	ControlValve	Off	NormalValve	NO	90
79	113	ControlValve	Off	NormalValve	NO	270
80	115	ControlValve	On	NormalValve	NC	0
81	116	ControlValve	On	NormalValve	NC	90
82	117	ControlValve	On	NormalValve	NC	270
83	118	ControlValve	Off	NormalValve	NC	0
84	119	ControlValve	Off	NormalValve	NC	90
85	120	ControlValve	Off	NormalValve	NC	270
86	122	ControlValve	On	CornerValve	1	0
87	123	ControlValve	On	CornerValve	1	90

รูปที่ 5.10 รูปแสดงตารางฐานข้อมูลอุปกรณ์ทั้งหมดที่เก็บไว้ใน Microsoft Access

ตารางข้อมูลอุปกรณ์รวม เป็นตารางที่ใช้เก็บอุปกรณ์ทั้งหมดที่เตรียมไว้สำหรับการออกแบบภาพจำลองกระบวนการ ซึ่งเป็นตารางที่ใช้อ่านอย่างเดียวไม่สามารถแก้ไขได้ ซึ่งฐานข้อมูลนี้จะประกอบด้วยค่าคุณสมบัติพื้นฐานต่าง ๆ ของอุปกรณ์แต่ละตัว ซึ่งฐานข้อมูลอุปกรณ์รวมนี้จะมีประโยชน์เมื่อเกิดการแก้ไขเปลี่ยนแปลงค่าต่าง ๆ ในหน้าต่างคุณสมบัติ ซึ่งจะสามารถหาออบเจ็กต์ที่มีคุณสมบัติพื้นฐานตามต้องการได้จากตารางอุปกรณ์รวมนี้ ทำให้สามารถแสดงภาพได้ถูกต้อง

5.5.2 ฐานข้อมูล 2 ตารางข้อมูลปัจจุบัน (Item Table)

ID	FormName	KeyItem	Item	ItemName	Address	L,T,W,H	ImageTag
(AutoNumber)		0					

รูปที่ 5.11 ตารางข้อมูลปัจจุบันขณะออกแบบกระบวนการ

ตารางข้อมูลอุปกรณ์ปัจจุบันนี้ เป็นตารางที่ใช้เก็บชื่อ และคุณสมบัติของอุปกรณ์ในขณะที่กำลังทำการออกแบบกระบวนการผลิต เป็นตารางฐานข้อมูลที่สามารถอัพเดต(Update) ข้อมูลได้โดยอัตโนมัติ เมื่อมีการเพิ่มหรือลดรูปในฟอร์มการออกแบบ หรือมีการแก้ไขค่าต่าง ๆในหน้าต่างคุณสมบัติ ซึ่งค่าคุณสมบัติที่แสดงในตารางข้อมูลนี้ จะถูกนำไปแสดงผลในหน้าต่างคุณสมบัติ

5.5.3 ฐานข้อมูล 3 ตารางข้อมูลอุปกรณ์สำหรับเปิดและบันทึกไฟล์ (Open-Save Table)

SaveOpen : ตาราง

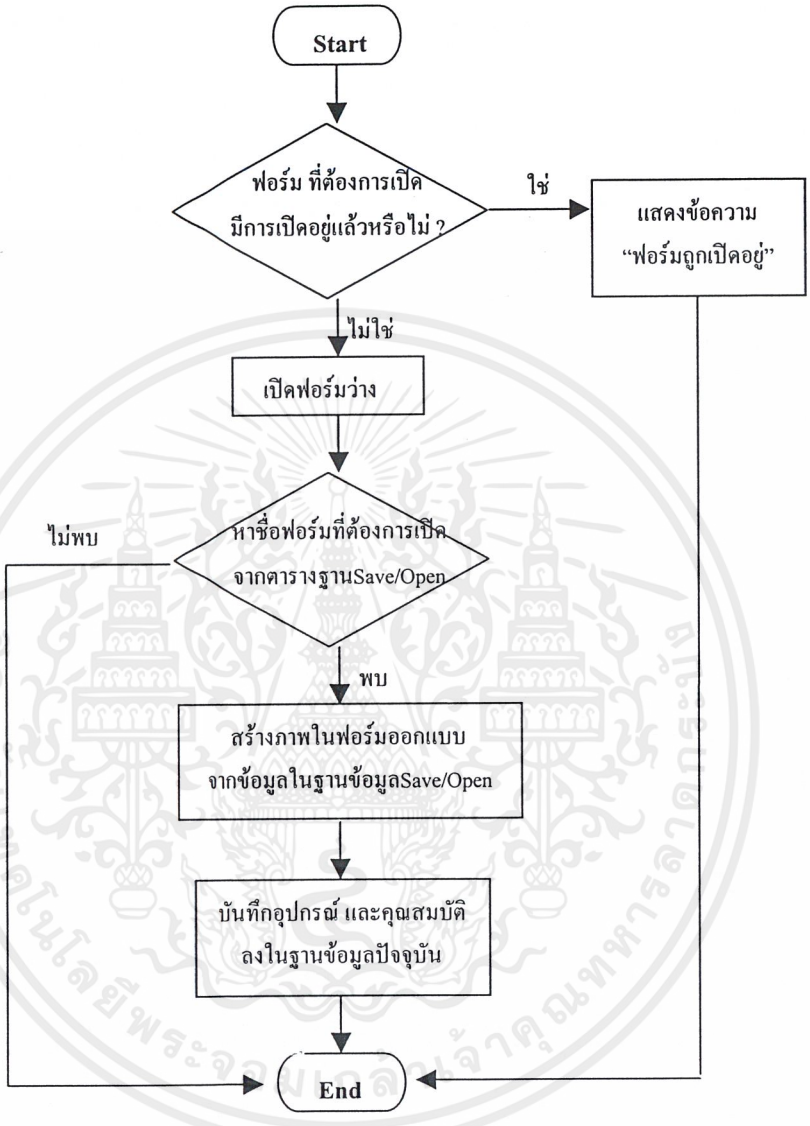
ID	FormName	KeyItem	Item	ItemName	Address	L,T,W,H	ImageTag	Imag	Save
27	PLC	118	ControlValve	ControlValve 0	100 00	3105, 31	NotPipeOutpu	1	☑
28	PLC	95	Pipe	Pipe 1		5835, 79	Pipe	2	☑
29	PLC	91	Pipe	Pipe 2		3900, 79	Pipe	3	☑
30	PLC	65	Reactor	Reactor 3		5700, 12	NotPipe	4	☑
31	PLC	45	Pump	Pump 4	100 04	960, 765	NotPipeOutpu	5	☑
32	PLC	91	Pipe	Pipe 5		2010, 79	Pipe	6	☑
33	PLC	164	PushButton	PushButton 6	000 06	8310, 18	NotPipeInput	7	☑
34	PLC	167	PushButton	PushButton 7	000 07	8370, 51	NotPipeInput	8	☑
40	Piston	174	LimitSwitch	Limit1	000 00	3480, 12	NotPipeInput	2	☑
41	Piston	22	Lamp	Lamp1	100 00	8235, 10	NotPipeOutpu	3	☑
42	Piston	176	LimitSwitch	Limit2	000 03	1095, 17	NotPipeInput	4	☑
43	Piston	22	Lamp	Lamp2	100 01	8325, 30	NotPipeOutpu	5	☑
44	Piston	36	Piston	Piston 6	100 06	1245, 13	NotPipeOutpu	7	☑
(Auto		0						0	☐

รูปที่ 5.12 ตารางข้อมูลอุปกรณ์สำหรับเปิดและบันทึกไฟล์

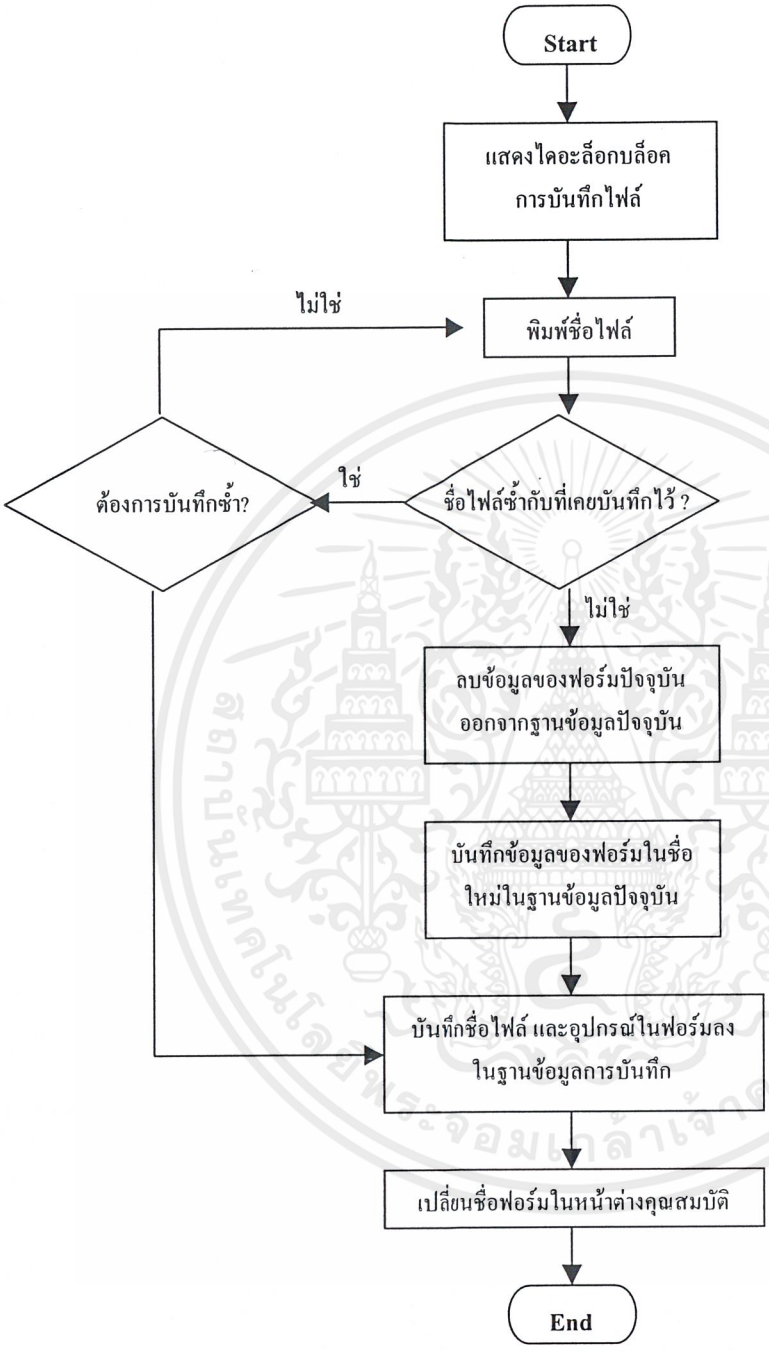
ตารางข้อมูลอุปกรณ์สำหรับเปิดและบันทึกไฟล์ จะใช้ในการเก็บชื่อของฟอร์ม และคุณสมบัติอุปกรณ์ในฟอร์มที่ถูกบันทึกเรียบร้อยแล้ว เมื่อคำสั่ง Open ในโปรแกรมหลักถูกเรียกใช้ โปรแกรมจะทำการตรวจสอบชื่อฟอร์มที่เลือกนั้นจากตารางฐานข้อมูลตารางนี้ และเมื่อพบชื่อฟอร์มที่ต้องการ โปรแกรมก็จะทำการสร้างฟอร์มนั้นขึ้นมาใหม่ตามคุณสมบัติของออบเจ็กต์ที่พบในตารางและฟอร์มที่ถูกสร้างขึ้นใหม่นี้ก็จะถูกนำไปเพิ่มในตารางข้อมูลอุปกรณ์ขณะทำงานตามลำดับ ส่วนเมื่อมีการ Save เกิดขึ้น โปรแกรมจะทำการอัพเดตข้อมูลในตารางข้อมูลสำหรับเปิด/บันทึกไฟล์โดยอัตโนมัติ

5.6 โฟว์ชาร์ตแสดงขั้นตอนการทำงานของ โปรแกรม

5.6.1 โฟว์ชาร์ตการเปิดไฟล์

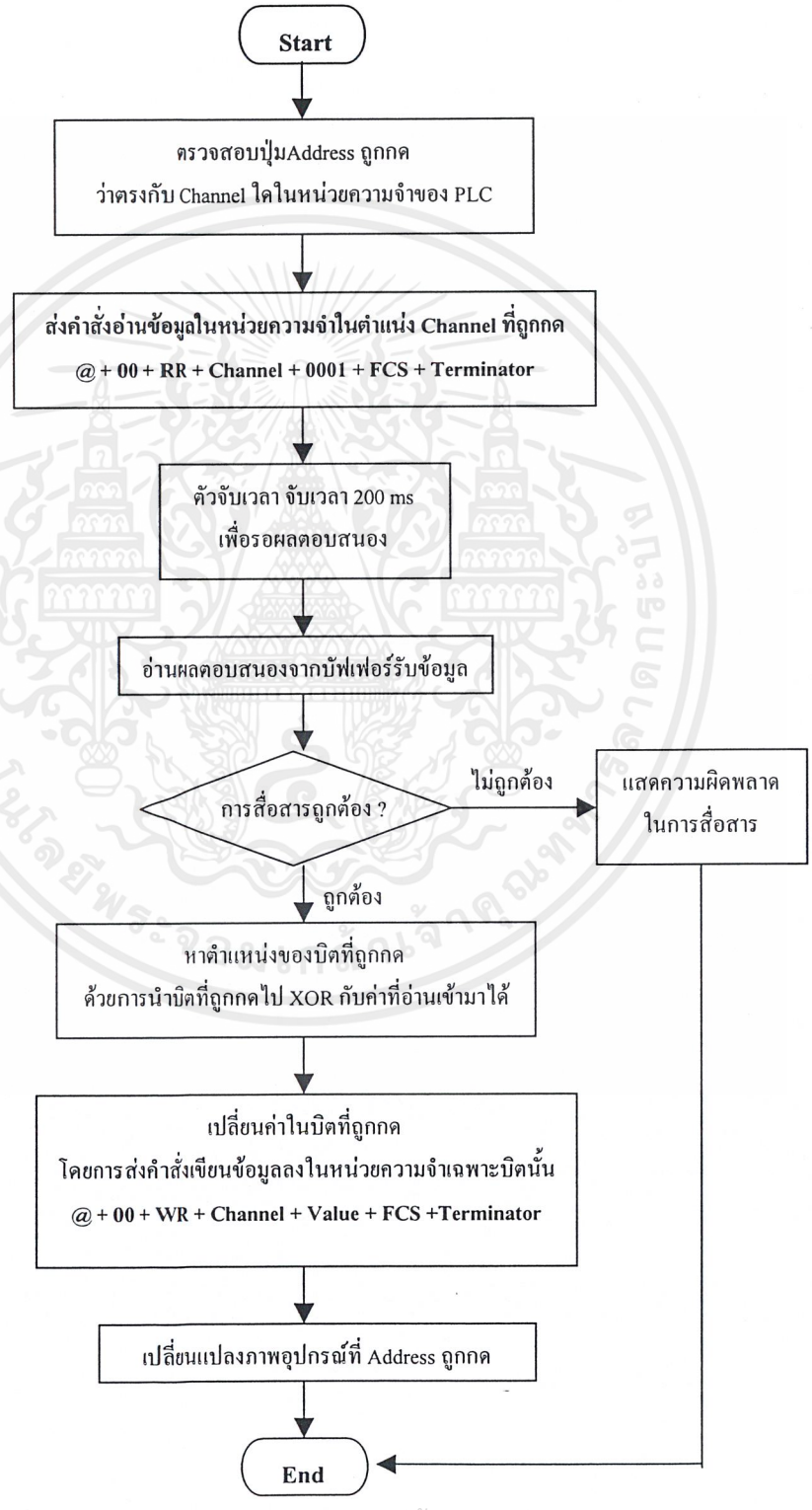


5.6.2 โฟว์ชาร์ตการบันทึก (Save)



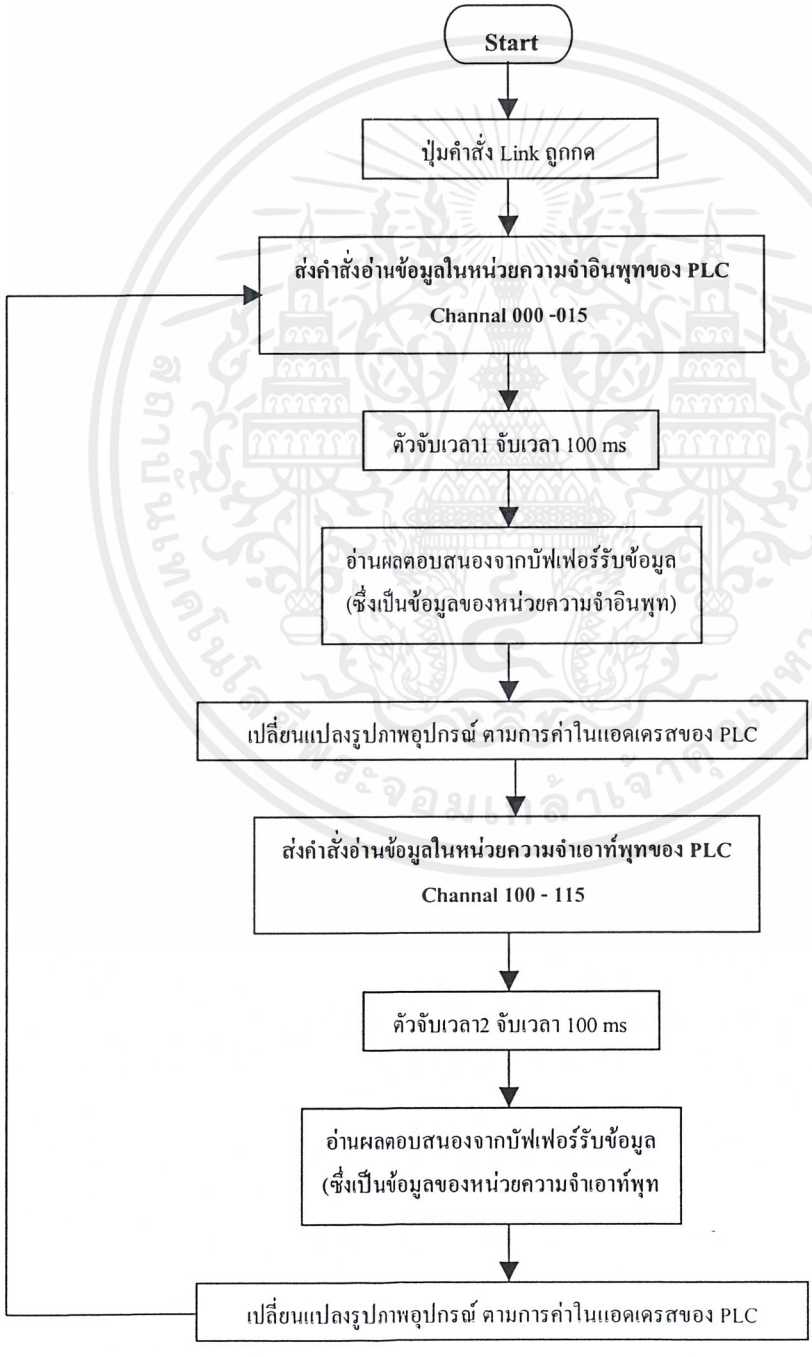
5.6.3 โฟว์ชาร์ตการสั่งงาน PLC ผ่านทางจอภาพ

การใช้งานโปรแกรมแสดงผลกระบวนการผลิตนี้ จะสามารถทำการสั่งงานเครื่อง PLC ผ่านทางจอภาพได้ โดยทำการกดปุ่มแอดเดรสของรูปภาพแต่ละภาพ ซึ่งแอดเดรสที่ถูกกดนี้ก็จะทำให้ตำแหน่งแอดเดรสจริงในหน่วยความจำของ PLC เกิดการเปลี่ยนแปลงด้วย ขั้นตอนการทำงาน คือ



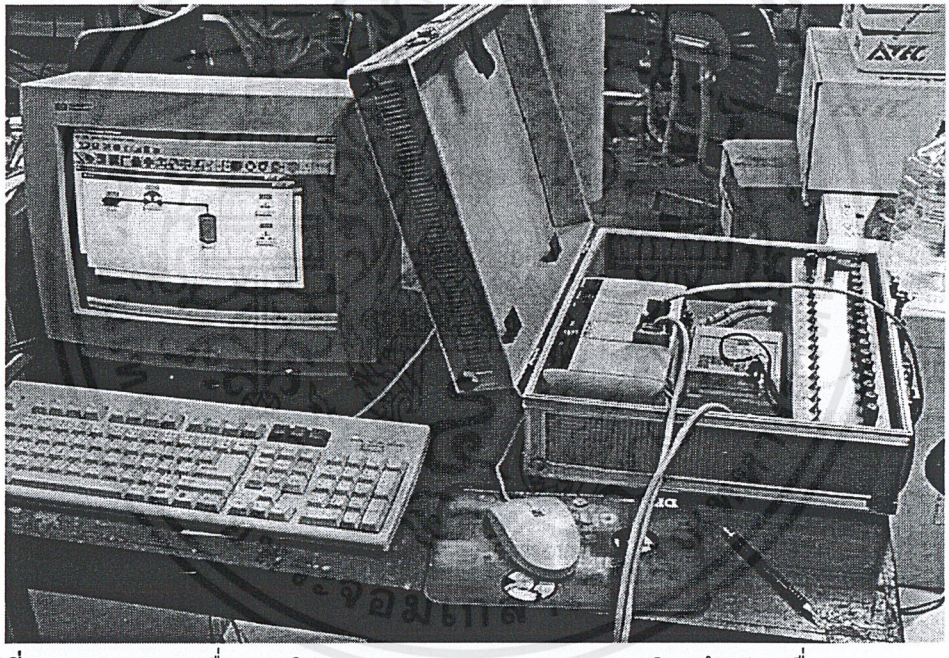
5.6.4 โฟว์ชาร์ตในการวนรอบ ตรวจสอบแอดเดรสของ PLC

ในโปรแกรมแสดงผลกระบวนการผลิตนี้ เมื่อมีการลิงค์ระหว่างเครื่องคอมพิวเตอร์กับ PLC เกิดขึ้น เมื่อมีการเปลี่ยนแปลงค่าต่างๆในหน่วยความจำของ PLC โปรแกรมแสดงผลกระบวนการผลิตจะต้องรับรู้ได้ว่าเกิดการเปลี่ยนแปลงในหน่วยความจำของ PLC และทำการเปลี่ยนแปลงรูปภาพในหน้าจอการออกแบบให้เหมาะสม ซึ่งเราจะใช้การวนรอบส่งคำสั่งอ่านข้อมูลจากหน่วยความจำของ PLCทุก ๆ 0.1 วินาที เพื่อติดตามการเปลี่ยนแปลงค่าในหน่วยความจำ สามารถขั้นตอนการตรวจสอบได้ดังนี้ (ขั้นตอนการตรวจสอบนี้จะสิ้นสุดเมื่อปุ่มคำสั่ง Link ถูกยกเลิกการกด)



บทที่ 6
การทดสอบโปรแกรมแสดงผลกระบวนการผลิต

โปรแกรมแสดงผลกระบวนการผลิต เป็นเพียงโปรแกรมที่ใช้ในการแสดงผลของกระบวนการผลิต ออกทางหน้าจอคอมพิวเตอร์ให้ผู้ใช้ได้เห็นเท่านั้น ส่วนการควบคุมจะถูกต้องหรือไม่นั้นขึ้นอยู่กับประสบการณ์ของผู้ใช้ในการที่จะเขียนโปรแกรมแลดเดอร์ให้กับเครื่อง PLC ดังนั้นโปรแกรมแสดงผลกระบวนการผลิตนี้อาจจะนำไปใช้ในการทดสอบโปรแกรมแลดเดอร์ที่ถูกเขียนขึ้นมาก่อนที่จะนำไปใช้ควบคุมกระบวนการผลิตจริง ๆ ทำให้สามารถปรับปรุงแก้ไขข้อผิดพลาดในการเขียนโปรแกรมควบคุมให้กับเครื่อง PLC ได้



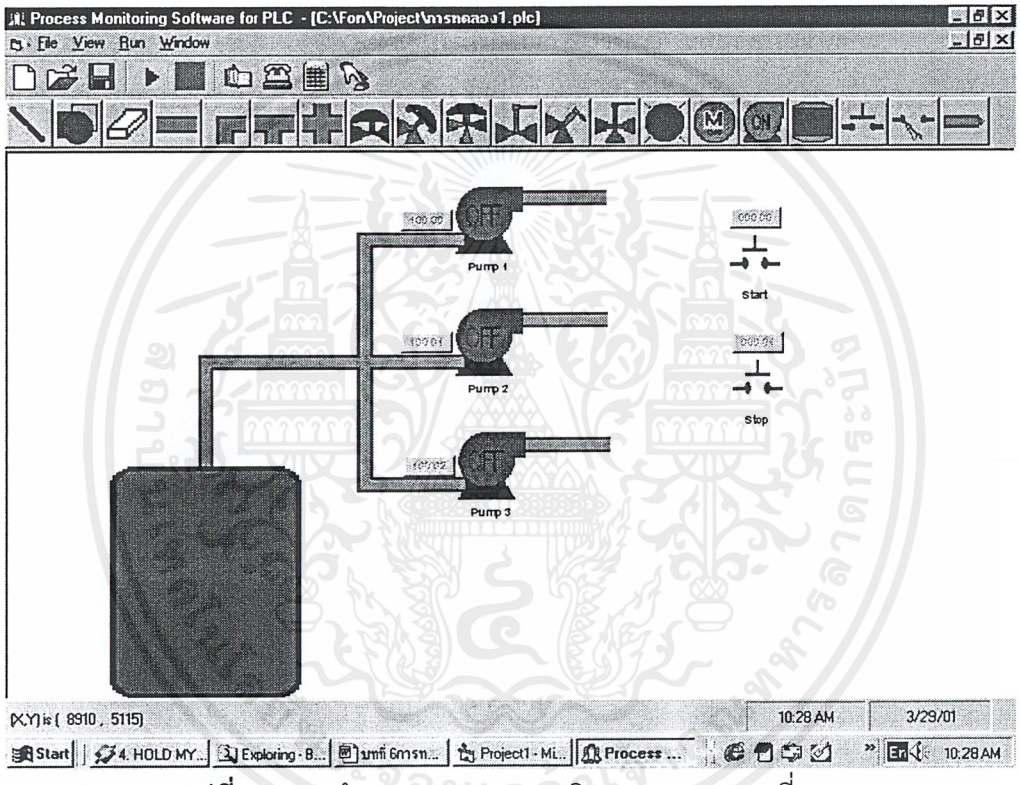
รูปที่ 6.1 การทดลองเชื่อมต่อโปรแกรมแสดงผลกระบวนการผลิต เข้ากับเครื่อง PLC CQM1

ในบทนี้จะกล่าวถึงการทดสอบโปรแกรมแสดงผลกระบวนการผลิต ซึ่งจะแสดงถึงขั้นตอนการใช้งานโปรแกรมทั้งหมด โดยการยกตัวอย่างโจทย์การควบคุมกระบวนการด้วย PLC ที่มีข้อกำหนดต่าง ๆ กัน นำมาทดลองสร้างเป็นกระบวนการจำลองในหน้าจอคอมพิวเตอร์ และทำการควบคุม และแสดงผลได้ตามการเปลี่ยนแปลงของ PLC

การทดลองที่ 6.1 การทำงานของไทเมอร์ (Timer)

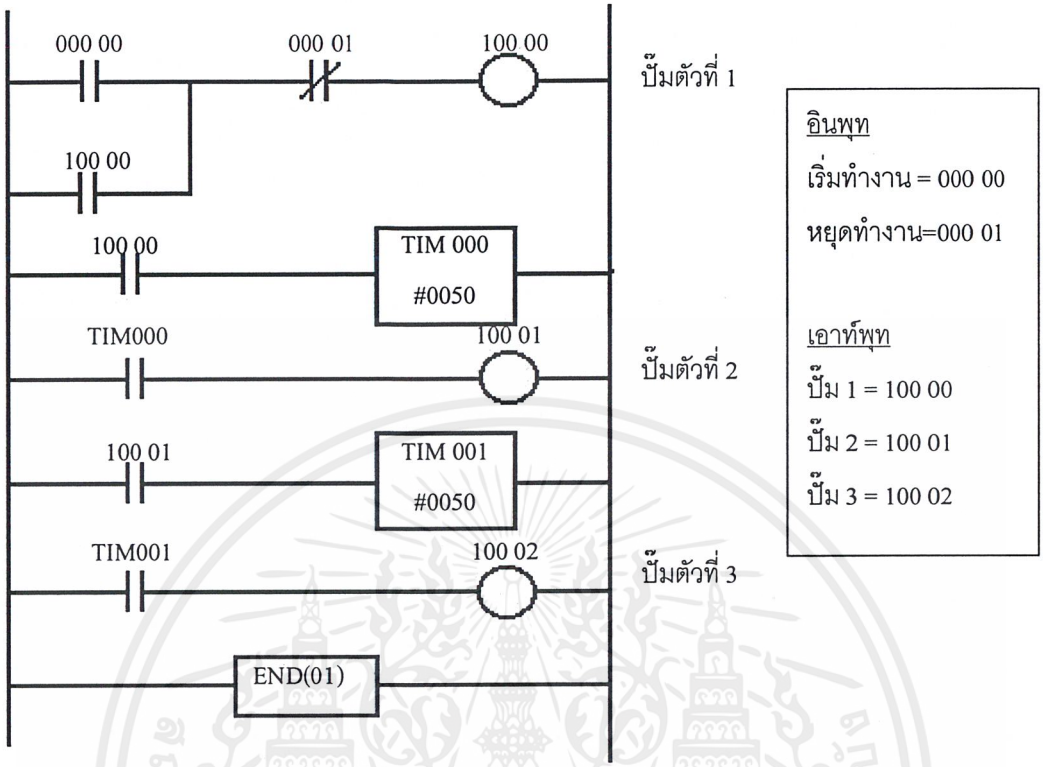
มีปั๊มอยู่ 3 ตัวทำหน้าที่ส่งน้ำมันไปยังเครื่องจักร การทำงานของระบบคือ เมื่อกดปุ่มสตาร์ท ปั๊มตัวที่ 1 จะทำงานทันที และหลังจากนั้น 5 วินาที ปั๊มตัวที่ 2 ทำงาน และอีก 5 วินาที ปั๊มตัวที่ 3 จะทำงาน และเมื่อกดปุ่มสต็อป ปั๊มทั้งหมดจะหยุดทำงาน

ขั้นตอนที่ 1 ออกแบบภาพแสดงระบบจำลองกระบวนการผลิต และกำหนดแอดเดรสให้กับอุปกรณ์ที่ทำหน้าที่เป็นอินพุท/เอาท์พุทของระบบ ดังรูปที่ 6.2



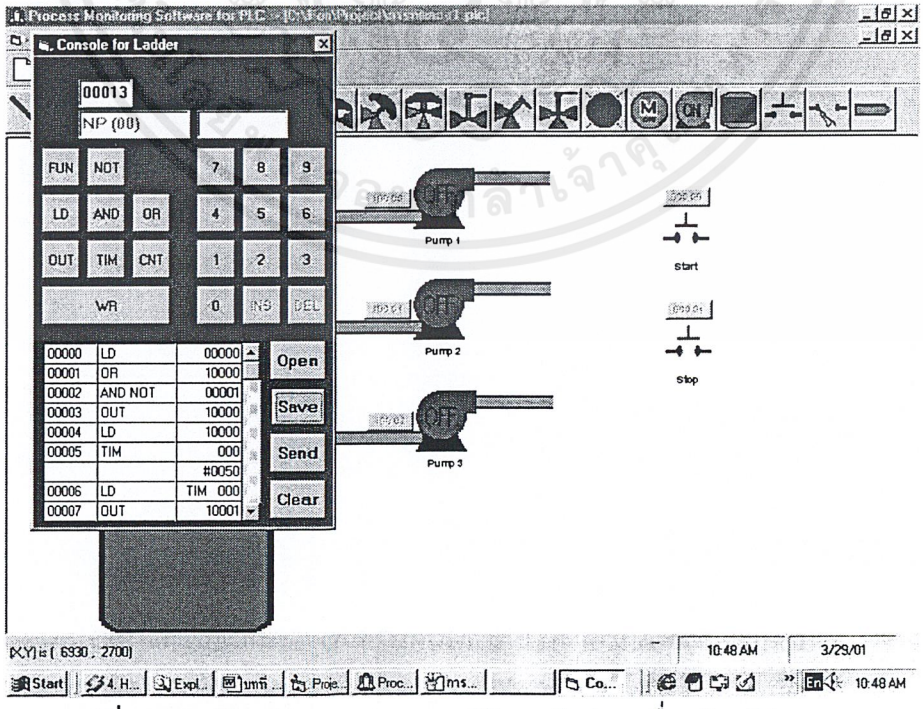
รูปที่ 6.2 ภาพจำลองกระบวนการผลิตของการทดลองที่ 6.1

ขั้นตอนที่ 2 ออกแบบโปรแกรมแลดเดอร์ และป้อนโปรแกรมให้กับเครื่อง PLC เพื่อให้เครื่อง PLC ทำหน้าที่ควบคุมอุปกรณ์ต่าง ๆ ได้ตามเงื่อนไขที่กำหนด ซึ่งในการทดลองที่ 1 นี้เราออกแบบโปรแกรมแลดเดอร์ได้ดังนี้



รูปที่ 6.3 แลดเดอร์ไดอะแกรมของการทดลองที่ 6.1

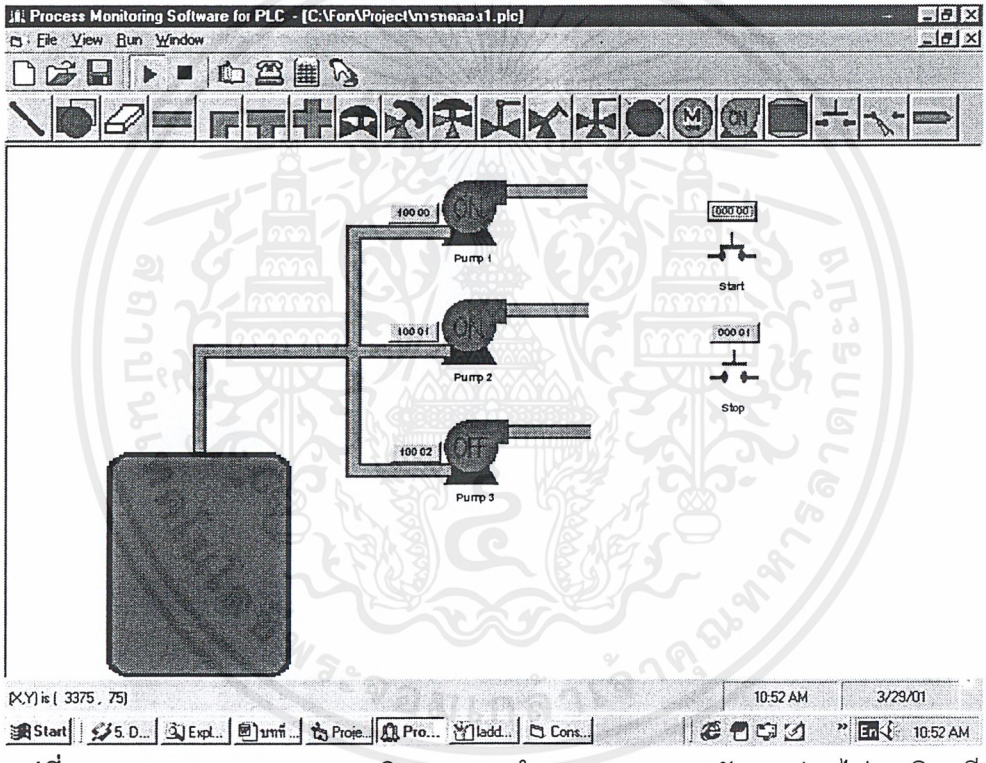
ในขั้นตอนนี้เราจะป้อนโปรแกรมแลดเดอร์ที่ออกแบบไว้แล้วด้วยหน้าต่างเครื่องป้อนโปรแกรม ดังรูปที่ 6.4



รูปที่ 6.4 การป้อนโปรแกรมแลดเดอร์ด้วยหน้าต่างเครื่องป้อนโปรแกรม

ขั้นตอนที่ 3 ทำการทดสอบการทำงานของโปรแกรมแสดงผลกระบวนการผลิต โดยการกำหนดค่าเริ่มต้นในการสื่อสารจากหน้าต่างการสื่อสาร และกดปุ่มคำสั่ง Link แล้วเริ่มทดสอบการควบคุมของ PLC ดังนี้

- ทดลองติดตามการเปลี่ยนแปลงของ PLC ใช้เครื่องป้อนโปรแกรมทำการเซตบิต 00000 ของเครื่อง PLC ให้เป็น 1 หน้าต่างจำลองกระบวนการผลิตที่ออกแบบไว้จะมีการเปลี่ยแปลงตามการเปลี่ยนแปลงของบิตอินพุท/เอาท์พุทในเครื่อง PLC คือ ภาพปุ่มกดแอดเดรส 00000 จะถูกกดและภาพปั๊ม 1 จะ On (กลายเป็นสีเขียว) หลังจากนั้น 5 นาที ภาพปั๊ม 2 ก็จะ On และหลังจากนั้นอีก 5 นาที ปั๊ม 3 ก็จะ On ภาพการเปลี่ยนแปลงหลังจากปั๊ม 2 On แสดงได้ดังรูปที่ 6.5

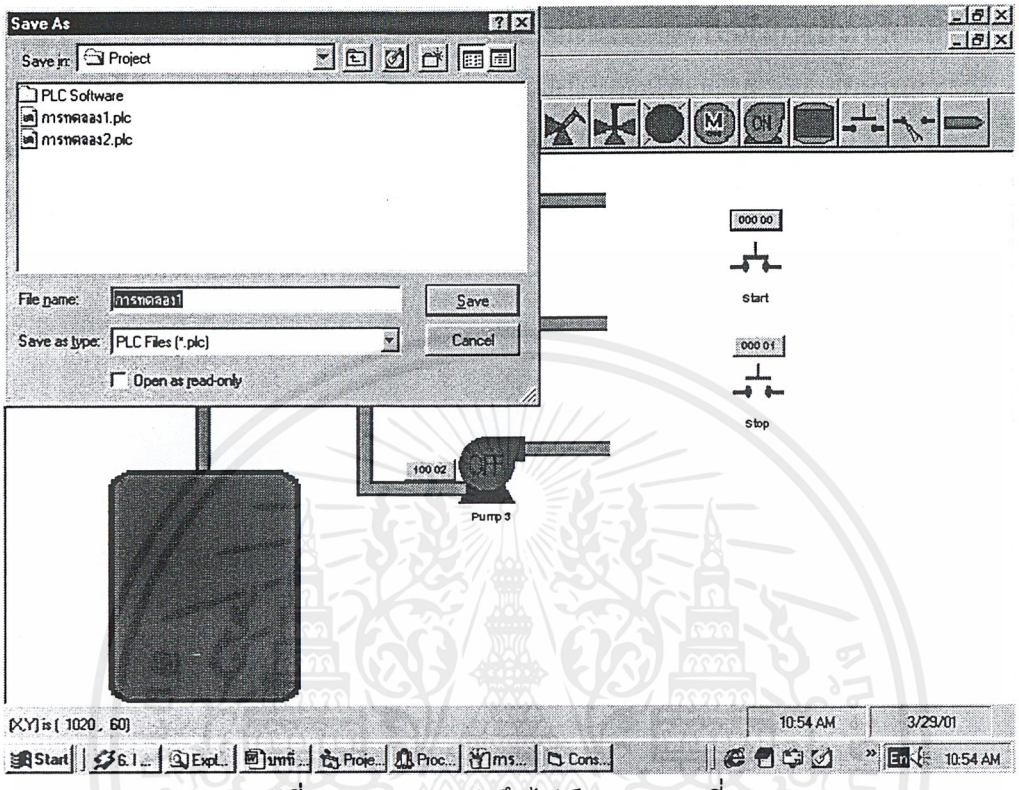


รูปที่ 6.5 ภาพแสดงกระบวนการผลิตตามการทำงานของ PLCหลังจากผ่านไป 10 วินาที

-ทดลองสั่งการทำงานให้กับเครื่อง PLC โดยการกดปุ่มสต๊อปตำแหน่ง 00001 ในภาพจำลองกระบวนการผลิต ผลที่ได้คือ บิตอินพุทตำแหน่ง 00001ในเครื่อง PLC ติด และบิตเอาท์พุตตำแหน่ง 10000 , 10001 และ10002 Off และรูปภาพปั๊มทั้ง 3 ในภาพจำลองกระบวนการผลิตจะเปลี่ยนเป็น Off (เปลี่ยนเป็นสีแดง)

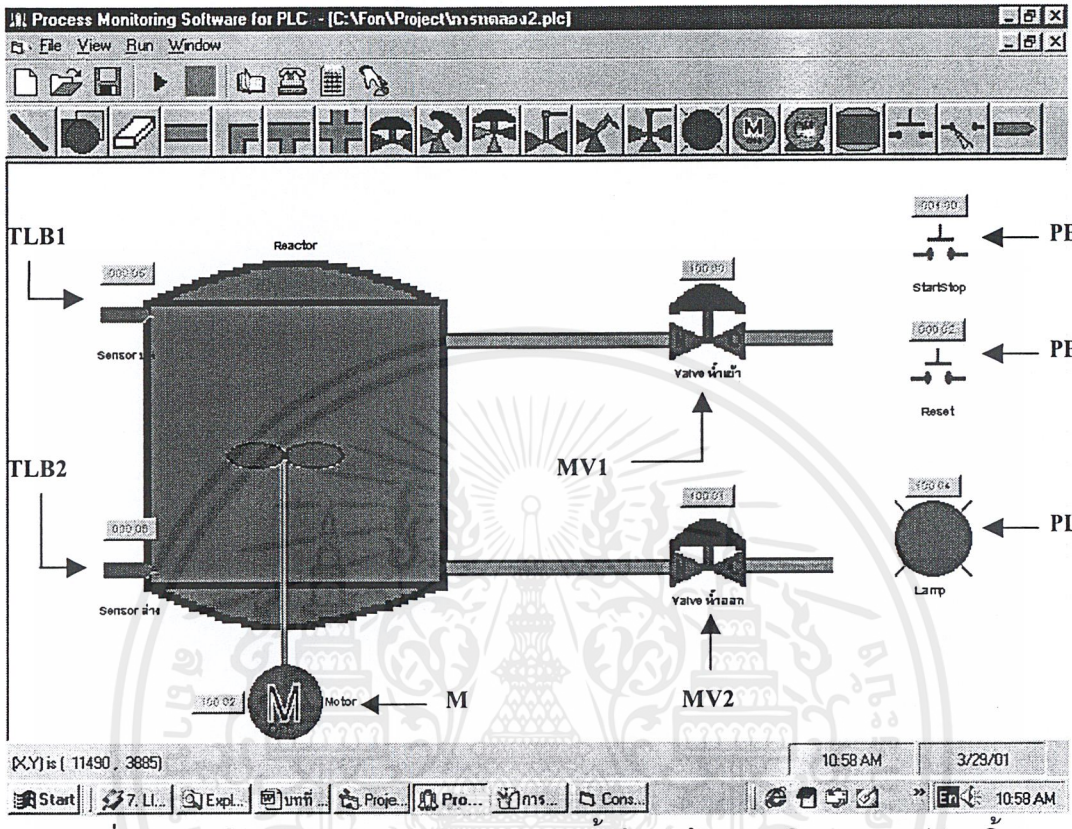
- ทดลองการบันทึก และการเปิดไฟล์ เมื่อออกแบบภาพจำลองกระบวนการผลิตเสร็จเรียบร้อยแล้ว ก็ทดลองทำการบันทึกไฟล์ ในการทดลองนี้ได้ทดลองบันทึกในชื่อ "การทดลองที่1.plc" แล้วปิดโปรแกรม จากนั้นจึงเปิดโปรแกรมแล้วทำการเปิดไฟล์ขึ้นมาเพื่อใช้งาน หรือทำการแก้ไขรูป

ภาพ ผลปรากฏว่าโปรแกรมแสดงผลกระบวนการผลิตสามารถทำการเปิดไฟล์ที่ถูกบันทึกไว้ได้ตามต้องการ



รูปที่ 6.6 แสดงการบันทึกไฟล์การทดลองที่ 6.1

การทดลองที่ 6.2 การควบคุมระดับน้ำในถังด้วยการเติม และการปล่อยทิ้ง



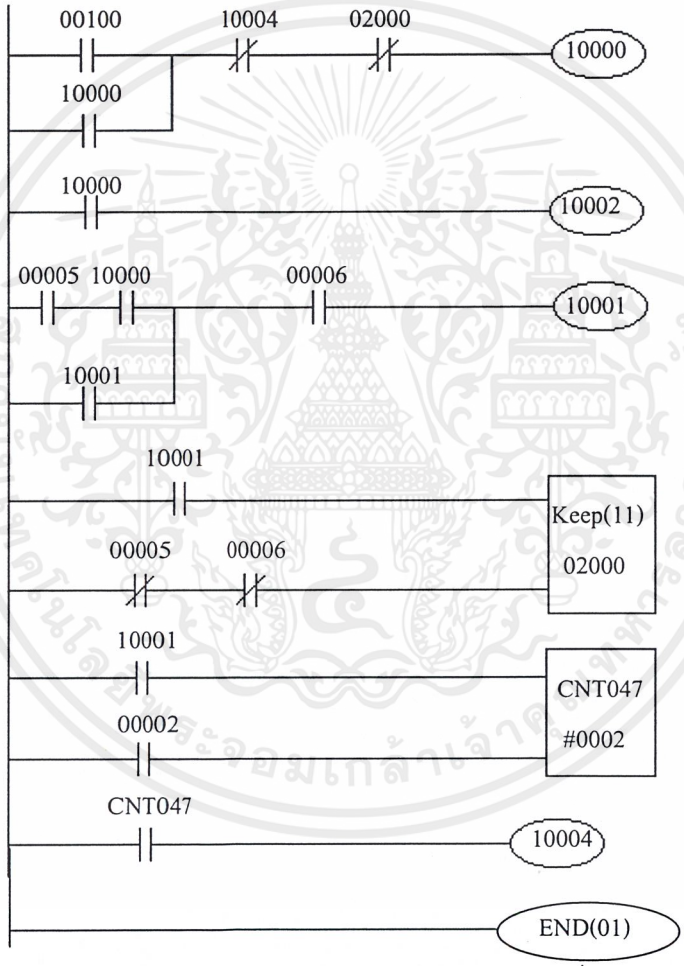
รูปที่ 6.7 ภาพจำลองกระบวนการควบคุมระดับนั้นในถังด้วยการเติมและการปล่อยทิ้ง

เงื่อนไขการทำงาน : เมื่อกดปุ่มสตาร์ท/สต็อปครั้งแรก (PB1) จะทำให้วาล์วอินพุท (MV1) เปิดให้น้ำเติมลงในถัง ขณะเดียวกันมอเตอร์ก็จะทำการหมุนใบพัดหมุนกวนน้ำในถัง เมื่อน้ำมีระดับสูงขึ้นมาผ่านเซ็นเซอร์ TLB2 จนถึง TLB1 แล้ว วาล์ว MV1 ปิด และมอเตอร์กวนน้ำก็จะหยุด หลังจากนั้น วาล์ว MV2 จะเปิดเพื่อระบายน้ำออกจากถังจนกระทั่งระดับน้ำต่ำกว่าเซ็นเซอร์ TLB2 จะทำให้วาล์ว MV2 ปิดเมื่อรอบการทำงานครบ 4 ครั้งแล้ว หลอดไฟแสดงว่าจบการทำงานจะติด และหลอดไฟจะดับเมื่อมีการกดปุ่มตั้งค่าใหม่ (Reset) หรือหยุดการทำงานด้วยการกดปุ่มสตาร์ท/สต็อปอีกครั้ง

ออกแบบโปรแกรมแลดเดอร์ : สำหรับการทดลองนี้เราจะกำหนดตำแหน่งแอดเดรสของปุ่มอินพุท และเอาท์พุทต่าง ๆ ดังนี้

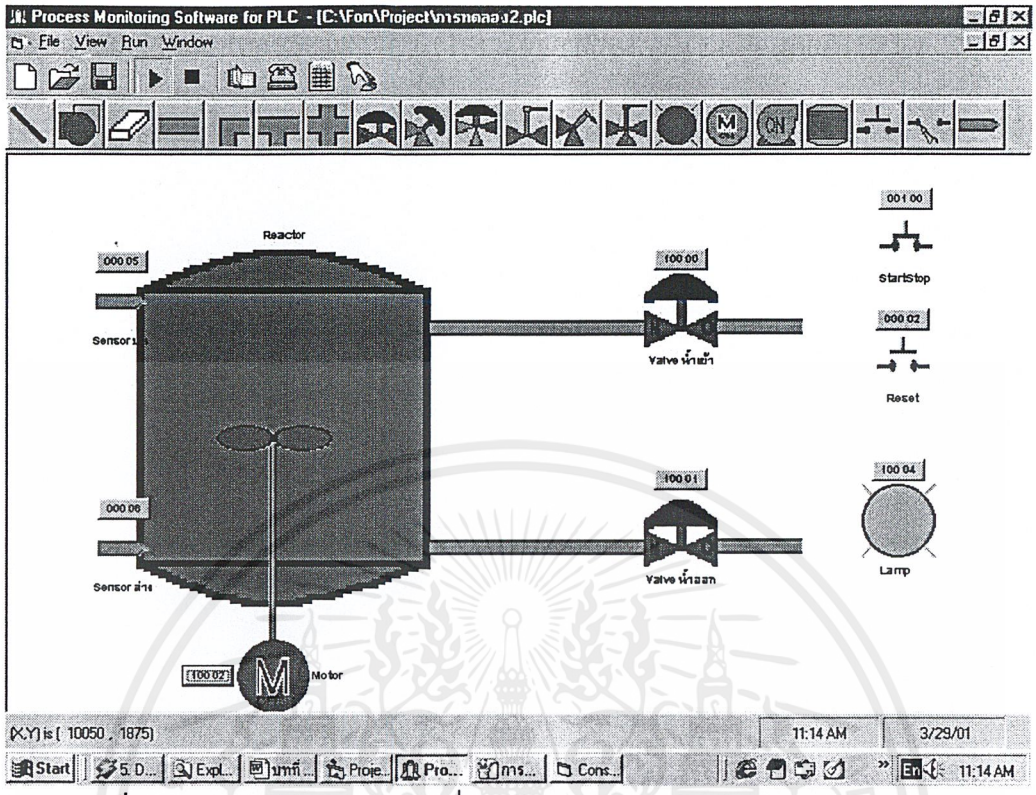
อินพุต :
00100 ปุ่มสตาร์ท/สต็อป (PB1)
00002 ปุ่มตั้งค่าใหม่ (PB3)
00005 เซ็นเซอร์วัดระดับบน (TLB1)
00006 เซ็นเซอร์วัดระดับล่าง (TLB2)

เอาท์พุท :
10000 วาล์วจ่ายน้ำ (MV1)
10001 วาล์วระบายน้ำ (MV1)
10002 มอเตอร์กวนน้ำ (M)
10004 หลอดไฟแสดงจบการทำงาน



รูปที่ 6.8 แลดเดอร์ไดอะแกรมของการทดลองที่ 6.2

ป้อนโปรแกรมแลดเดอร์ที่ออกแบบได้ให้กับเครื่อง PLC แล้วทำการทดสอบการทำงานของ PLC จากโปรแกรมแลดเดอร์ที่ออกแบบได้ โดยการเซตบิตอินพุทต่าง ๆ ด้วยเครื่องป้อนโปรแกรม แล้วดูผลได้จากโปรแกรมแสดงผลกระบวนการผลิต ซึ่งสามารถแสดงการเปลี่ยนแปลงที่เกิดขึ้นได้ ดังรูปที่ 6.9



รูปที่ 6.9 การแสดงผลการทดลองที่ 6.2 ด้วยโปรแกรมแสดงผลกระบวนการผลิต

บทที่ 7
บทวิจารณ์ และ สรุป

บทวิจารณ์

โปรแกรมแสดงผลกระบวนการผลิตที่สามารถออกแบบได้นี้ ยังมีข้อจำกัดในการใช้งานหลายประการ กล่าวคือ แม้ว่าตัวโปรแกรมจะจัดหาอุปกรณ์ที่ใช้ในกระบวนการผลิตแบบต่าง ๆ มารองรับการออกแบบของผู้ใช้อย่างมากแล้วก็ตาม แต่อุปกรณ์ที่จัดหาไว้ให้เหล่านี้ก็เป็นเพียงอุปกรณ์มาตรฐานที่พบเห็นในกระบวนการผลิตทั่ว ๆ ไป ทำให้เมื่อต้องการจะออกแบบกระบวนการผลิตที่มีประกอบด้วยอุปกรณ์แปลก ๆ ที่พบเห็นได้ไม่บ่อยนัก โปรแกรมนี้จะไม่สามารถออกแบบได้ ข้อจำกัดอีกประการหนึ่งของโปรแกรมแสดงผลกระบวนการนี้ก็คือ เป็นโปรแกรมที่แสดงผลได้เฉพาะสถานะปิด หรือเปิดเท่านั้น หรือกล่าวอีกนัยหนึ่งคือ โปรแกรมนี้จะสนับสนุนการควบคุมแบบซีเควนซ์เท่านั้น ไม่สามารถที่จะแสดงผลของกระบวนการที่มีการควบคุมแบบต่อเนื่อง (แบบอะนาล็อก) ได้ อย่างไรก็ตามเมื่อนำโปรแกรมนี้มาทดลองแสดงผลกระบวนการผลิตที่ควบคุมแบบซีเควนซ์ด้วย PLC ตามการทดลองในบทก่อนหน้านี้แล้ว แสดงให้เห็นว่าโปรแกรมแสดงผลกระบวนการผลิตนี้ สามารถทำงานร่วมกับเครื่อง PLC ได้อย่างมีประสิทธิภาพ

สรุปผลโครงงาน

โปรแกรมแสดงผลกระบวนการผลิตที่ออกแบบได้นี้ เป็นโปรแกรมที่ใช้ต้องใช้ความรู้เกี่ยวกับโปรแกรมสำเร็จรูปหลายโปรแกรมด้วยกันในการออกแบบ ได้แก่ โปรแกรมวิชวลเบสิค6.0 (Visual Basic) ซึ่งใช้เป็นโปรแกรมหลักในการออกแบบลักษณะ และการทำงานของหน้าต่างที่ใช้ติดต่อกับผู้ใช้งาน ,โปรแกรมไมโครซอฟท์เอ็กเซส (Microsoft Access) ซึ่งใช้ในการออกแบบระบบฐานข้อมูลให้กับโปรแกรมหลัก และโปรแกรมเกี่ยวกับการสร้างภาพกราฟิกต่าง ๆ เช่น โปรแกรมโฟโต้ช้อป(Photoshop 5.0) , โปรแกรมเพนท์ (Paint) เป็นต้น ส่วนในด้านการติดต่อสื่อสารระหว่างเครื่องคอมพิวเตอร์กับ PLC นั้นจะใช้การสื่อสารผ่านพอร์ตอนุกรมด้วยมาตราฐาน RS-232C และใช้รูปแบบของข้อมูลในการสื่อสารตามข้อกำหนด Host Link Communication ของเครื่อง PLC ซึ่งเป็นรูปแบบของชุดคำสั่งที่เครื่อง PLC สามารถเข้าใจและปฏิบัติตามได้

คุณสมบัติของโปรแกรมแสดงผลกระบวนการผลิตที่ออกแบบได้นี้คือ สามารถแสดงผลการเปลี่ยนแปลงบิตในหน่วยความจำในส่วนพื้นที่รีเลย์ภายในของเครื่อง PLC ได้โดยใช้ภาพกราฟิก

แสดงการเปลี่ยนแปลงที่เกิดขึ้นให้ผู้ใช้เห็น และสามารถสั่งการทำงานของเครื่อง PLC ได้ผ่านทางภาพกราฟฟิกที่ออกแบบไว้แล้วเช่นกัน

ปัญหา และข้อเสนอแนะ

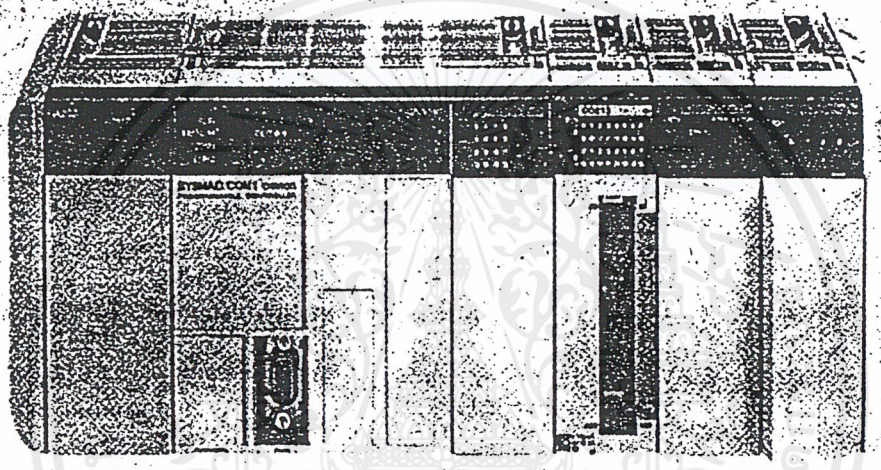
โปรแกรมแสดงผลกระบวนการผลิตที่ออกแบบได้นี้ มีขนาดใหญ่ และมีการทำงานซับซ้อน ทำให้การทำงานโดยรวมของโปรแกรมค่อนข้างช้า และควรจะปรับปรุงให้มีความสะดวกในการใช้งานมากขึ้น เช่น อาจจะออกแบบให้สามารถวาดอุปกรณ์ขึ้นมาใช้งานเองได้สำหรับอุปกรณ์ที่ไม่ค่อยพบเห็นในกระบวนการผลิตทั่ว ๆ ไป และอาจจะออกแบบโปรแกรมให้สามารถเลือกสีของอุปกรณ์ หรือทำให้อุปกรณ์สามารถเคลื่อนที่ได้อย่างต่อเนื่อง เป็นต้น ซึ่งการออกแบบให้ได้ตามข้อกำหนดเหล่านี้จะต้องใช้ทักษะในการเขียนโปรแกรม และอาศัยประสบการณ์ในการเขียนโปรแกรมอย่างมาก

ภาคผนวก

เครื่องควบคุมแบบโปรแกรมได้ (PLC) รุ่น CQM1 ของ OMRON

CQM1
PLC สมรรถนะสูงจาก OMRON

บริษัท ออมรอน อีเลคทรอนิคส์ จำกัด



บริษัท ออมรอน อีเลคทรอนิคส์ จำกัด เป็นบริษัทสาขาจากประเทศญี่ปุ่น ก่อตั้งบริษัทในประเทศไทยมาเป็นเวลาถึง 2 ปี เพื่อให้บริการต่างๆ ดังต่อไปนี้

1. ระบบการให้บริการด้านวิศวกรรม
2. ระบบสินค้าคงคลัง
3. ระบบการซ่อมแซมและบำรุงรักษา
4. การให้บริการด้านการฝึกอบรมเกี่ยวกับผลิตภัณฑ์
5. ระบบการให้บริการแก่ลูกค้าญี่ปุ่นในประเทศไทย
6. การให้บริการด้านเอกสารและคู่มือเกี่ยวกับผลิตภัณฑ์

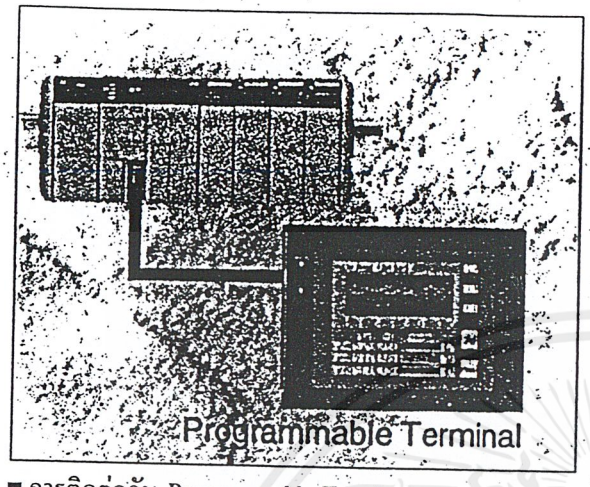
ผลิตภัณฑ์ของบริษัท ออมรอน อีเลคทรอนิคส์ จำกัด เป็นผลิตภัณฑ์ในระบบ Factory Automation ซึ่งสินค้าที่นำเข้ามาจำหน่ายเป็นผลิตภัณฑ์ของออมรอนที่ผลิตในหลายๆ ประเทศอาทิ ญี่ปุ่น, มาเลเซีย, อินโดเนเซีย, ไต้หวัน, จีน, เยอรมัน, และ ฮอลแลนด์ ส่วนในประเทศไทยนั้นยังไม่มีโรงงานผลิต

ในปัจจุบันระบบ Factory Automation ได้เข้ามามีบทบาทสำคัญมากขึ้นในงานอุตสาหกรรม ดังนั้นออมรอนจึงนำเสนอผลิตภัณฑ์ PLC รุ่น CQM1 ซึ่งเป็น PLC ขนาดเล็กที่มีสมรรถนะสูง มีความยืดหยุ่นในการใช้งานและใช้ในระบบ CompoBus ได้ ซึ่งเป็นระบบที่ถูกพัฒนาขึ้นเพื่อประหยัดต้นทุนและเวลาในการเดินสายไฟจากอุปกรณ์อินพุตและเอาต์พุตทำให้ง่ายและสะดวกในการออกแบบระบบควบคุม

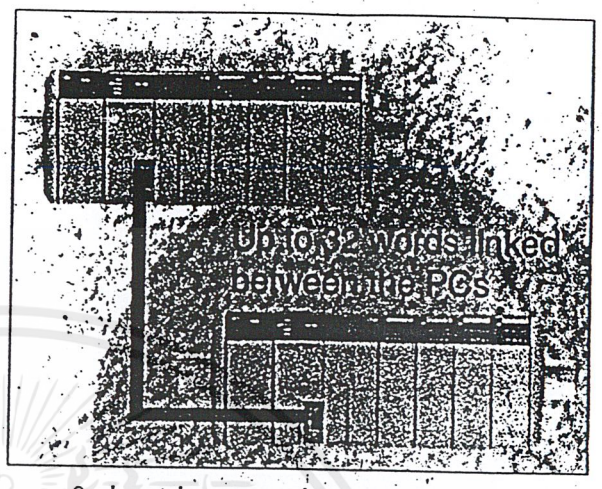
คุณสมบัติของ CQM1

ขนาดอินพุต/เอาต์พุตมากที่สุด	256 จุด
หน่วยความจำโปรแกรม	77.2k Words
หน่วยความจำข้อมูล	6k Words
รีเลย์ภายใน	2,720 ตัว
จำนวนคำสั่ง	Basic 14, Special103
ความเร็วในการคำสั่งพื้นฐาน	0.5 µs
ไทม์เมอร์/เคาน์เตอร์	512 ตัว
ระบบการติดต่อสื่อสาร	*CompoBus/D *CompoBus/S *Host Link *NT Link *1:1 Link
ฟังก์ชันควบคุมพิเศษ	*Anolog I/O *Sensor *Temperature *Linear Sensor Interface *B7A Interface

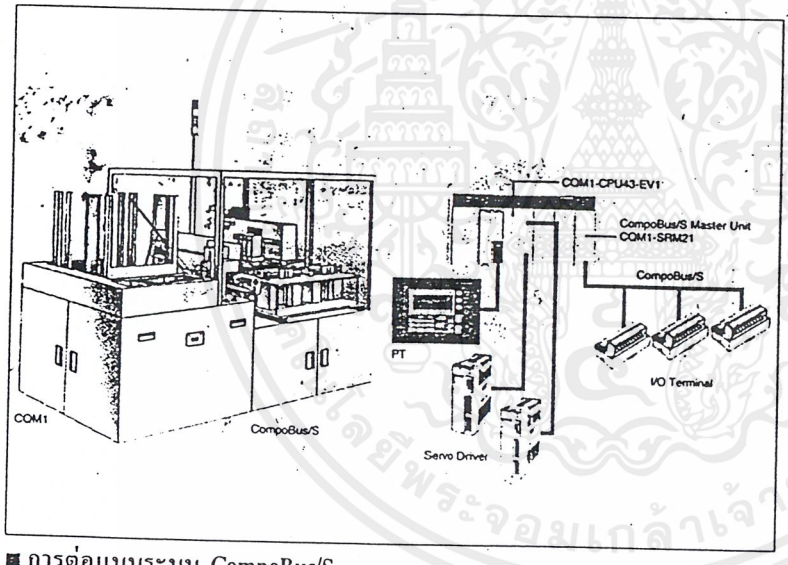
ระบบการติดต่อสื่อสาร



■ การติดต่อกับ Programmable Terminal



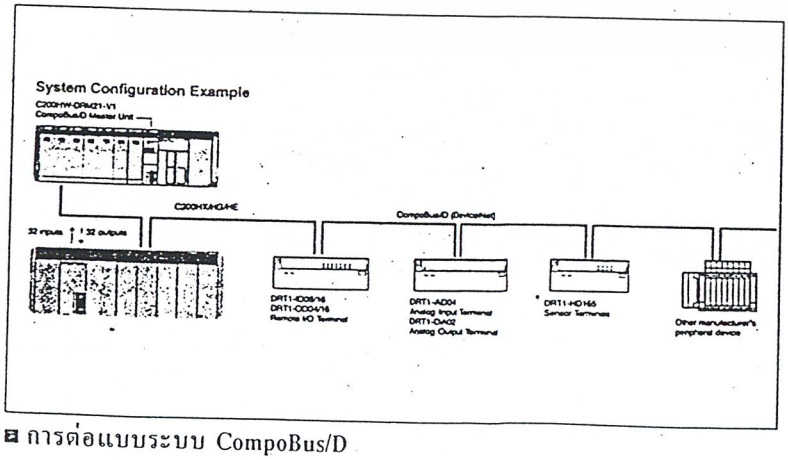
■ การติดต่อระหว่าง CQM1 ผ่าน RS-232



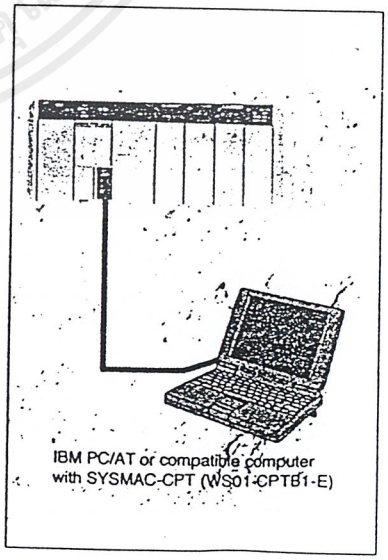
■ การต่อแบบระบบ CompoBus/S

การเขียน Program

CQM1 สามารถใช้อุปกรณ์ในการเขียนโปรแกรมชนิดเดียวกับที่ใช้ใน PLC ตระกูล Sysmac ซึ่งใช้ได้ทั้ง Programming Console และ SSS (DOS), Syswin (Window) และ Software ใหม่ที่ได้ถูกพัฒนาขึ้น สามารถใช้ง่าย เพราะมีฟังก์ชันการทำงานต่าง ๆ ให้เขียนและแก้ไขได้ง่าย เป็นโปรแกรม Ladder บน Windows คือ Sysmac-CPT

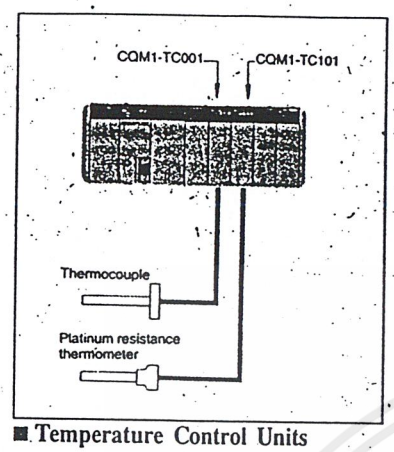


■ การต่อแบบระบบ CompoBus/D

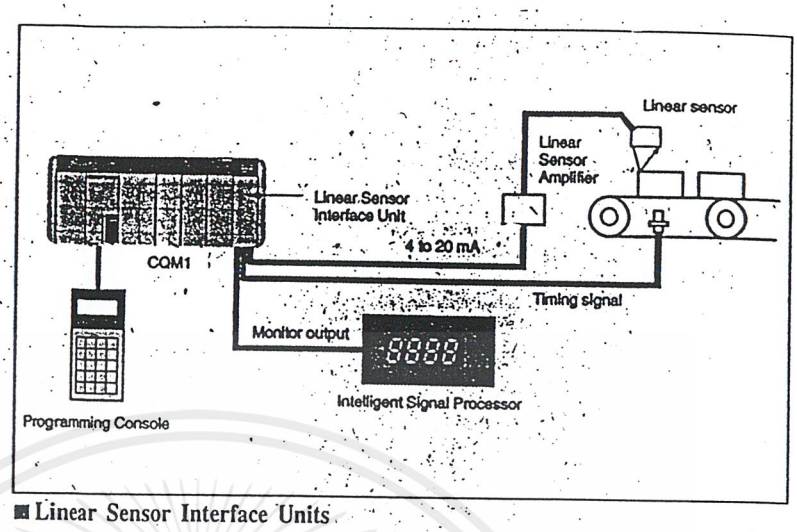


■ การเขียนโปรแกรมเข้า CQM1 ผ่านคอมพิวเตอร์

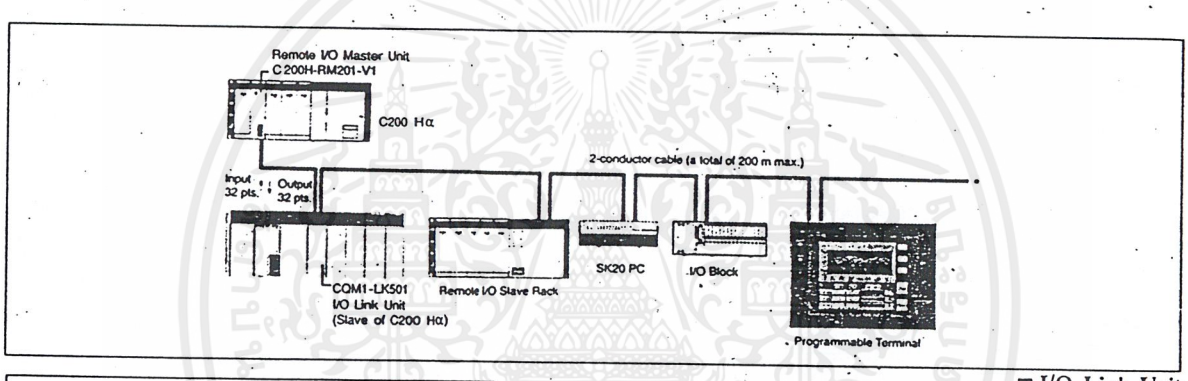
โมดูลควบคุมฟังก์ชันพิเศษ



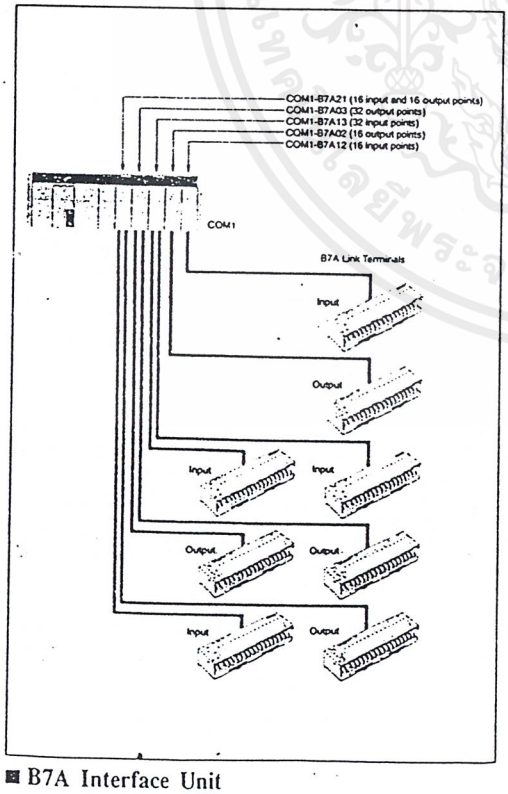
■ Temperature Control Units



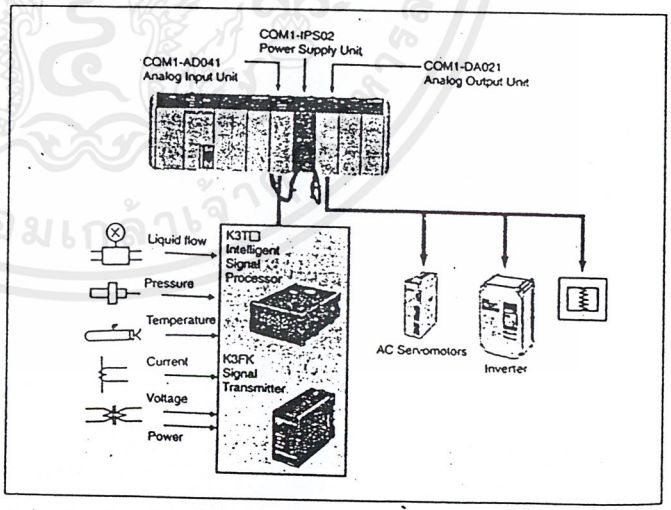
■ Linear Sensor Interface Units



■ I/O Link Unit



■ B7A Interface Unit



■ Analog Input and Output Units

สนใจตัวสินค้าและข้อมูลเพิ่มเติม
ติดต่อ บริษัท ออมรอน อีเลคทรอนิคส์ จำกัด
โทรศัพท์ 937-0500

ตารางภาคผนวกที่ 1 การจัดสรรพื้นที่หน่วยความจำของ PLC CQM1

Data area		Size	Words	Bits	Function
IR area[1]	Input area	128, or 256 bits	IR 000 to IR 015	IR 00000 to IR 01515	CQM1-CPU11/21-E: Up to 8 words (128 bits) can be used for I/O bits. Up to 7 Units can be connected.
	Output area		IR 100 to IR 115	IR 10000 to IR 11515	CQM1-CPU4□-EV1: Up to 16 words (256 bits) can be used for I/O bits. Up to 11 Units can be connected.
	Work areas	2,720 bits min.[2]	IR 016 to IR 095	IR 01600 to IR 09515	Work bits do not have any specific function, and they can be freely used within the program.
			IR 116 to IR 195	IR 11600 to IR 19515	
			IR 216 to IR 219	IR 21600 to IR 21915	
			IR 224 to IR 229	IR 22400 to IR 22915	
MACRO operand area[1]	Input area	64 bits	IR 096 to IR 099	IR 09600 to IR 09915	Used when the MACRO instruction, MCRO(99), is used. When the MACRO instruction is not used, these bits may be used as work bits.
	Output area	64 bits	IR 196 to IR 199	IR 19600 to IR 19915	
Analog SV area[1]		64 bits	IR 220 to IR 223	IR 22000 to IR 22315	CQM1-CPU42-EV1: Used to store the analog set values. (Cannot be used as work bits.) Can be used as work bits in other CPU Units.
High-speed Counter 0 PV[1]		32 bits	IR 230 to IR 231	IR 23000 to IR 23115	Used to store the present values of high-speed counter 0.
Port 1 and 2 Pulse Output PVs[1]		64 bits	IR 236 to IR 239	IR 23600 to IR 23915	CQM1-CPU43-EV1: Used to store the present values of pulse outputs for ports 1 and 2. (Cannot be used as work bits.) CQM1-CPU44-EV1: Used by the system. (Cannot be used as work bits.) Can be used as work bits in other CPU Units.
High-speed Counter 1 and 2 PVs[1]		64 bits	IR 232 to IR 235	IR 23200 to IR 23515	CQM1-CPU43/44-EV1: Used to store the present values of high-speed counters 1 and 2 for ports 1 and 2. (Cannot be used as work bits.) Can be used as work bits in other CPU Units.
Expansion Areas[1]		320 bits	IR 200 to IR 215 IR 240 to IR 243	IR 20000 to IR 21515 IR 24000 to IR 24315	These bits are expected to be used in planned function expansion.
SR area		184 bits	SR 244 to SR 255	SR 24400 to SR 25507	These bits serve specific functions such as flags and control bits. Can be used as work bits.
TR area		8 bits	---	TR 0 to TR 7	These bits are used to temporarily store ON/OFF status at program branches.
HR area		1,600 bits	HR 00 to HR 99	HR 0000 to HR 9915	These bits store data and retain their ON/OFF status when power is turned off.
AR area		448 bits	AR 00 to AR 27	AR 0000 to AR 2715	These bits serve specific functions such as flags and control bits.
LR area[1]		1,024 bits	LR 00 to LR 63	LR 0000 to LR 6315	Used for 1:1 data link through the RS-232 port.
Timer/Counter area[3]		512 bits	TC 000 to TC 511 (timer/counter numbers)		The same numbers are used for both timers and counters. TC 000 to TC 002 are used for interval timers.

Data area		Size	Words	Bits	Function
DM area	Read/write	1,024 words	DM 0000 to DM 1023	---	DM area data can be accessed in word units only. Word values are retained when the power is turned off.
		5,120 words	DM 1024 to DM 6143	---	Available in CQM1-CPU4□-EV1 CPU Units only.[4]
	Read-only[5]	425 words	DM 6144 to DM 6568	---	Cannot be overwritten from program.
	Error history area[5]	31 words	DM 6569 to DM 6599	---	Used to store the time of occurrence and error code of errors that occur.
	PC Setup[5]	56 words	DM 6600 to DM 6655	---	Used to store various parameters that control PC operation.
User program area (UM area)		3,200 or 7,200 words	---		Used to store the program. Retained when the power is turned off. CQM1-CPU11/21-E: 3,200 words CQM1-CPU4□-EV1: 7,200 words

ตารางภาคผนวกที่ 2 แสดงคำสั่งต่าง ๆ(Header Code)ที่ใช้ในการติดต่อสื่อสารแบบ Host Link

Header code	Name	PC mode			Applicable PCs	Page
		RUN	MON	PRG		
RR	IR/SR AREA READ	Valid	Valid	Valid	All	356
RL	LR AREA READ	Valid	Valid	Valid	All	356
RH	HR AREA READ	Valid	Valid	Valid	All	357
RC	PV READ	Valid	Valid	Valid	All	357
RG	TC STATUS READ	Valid	Valid	Valid	All	358
RD	DM AREA READ	Valid	Valid	Valid	All	358
RJ	AR AREA READ	Valid	Valid	Valid	All	359
WR	IR/SR AREA WRITE	Not valid	Valid	Valid	All	359
WL	LR AREA WRITE	Not valid	Valid	Valid	All	360
WH	HR AREA WRITE	Not valid	Valid	Valid	All	360
WC	PV WRITE	Not valid	Valid	Valid	All	361
WG	TC STATUS WRITE	Not valid	Valid	Valid	All	362
WD	DM AREA WRITE	Not valid	Valid	Valid	All	362
WJ	AR AREA WRITE	Not valid	Valid	Valid	All	363
R#	SV READ 1	Valid	Valid	Valid	All	364
R$	SV READ 2	Valid	Valid	Valid	All	364
R%	SV READ 3	Valid	Valid	Valid	COM1 only	365
W#	SV CHANGE 1	Not valid	Valid	Valid	All	366
W$	SV CHANGE 2	Not valid	Valid	Valid	All	367
W%	SV CHANGE 3	Not valid	Valid	Valid	COM1 only	368
MS	STATUS READ	Valid	Valid	Valid	All	369
SC	STATUS WRITE	Valid	Valid	Valid	All	370
MF	ERROR READ	Valid	Valid	Valid	All	370
KS	FORCED SET	Not valid	Valid	Valid	All	371
KR	FORCED RESET	Not valid	Valid	Valid	All	372
FK	MULTIPLE FORCED SET/RESET	Not valid	Valid	Valid	All	373
KC	FORCED SET/RESET CANCEL	Not valid	Valid	Valid	All	374
MM	PC MODEL READ	Valid	Valid	Valid	All	375
TS	TEST	Valid	Valid	Valid	All	375
RP	PROGRAM READ	Valid	Valid	Valid	All	376
WP	PROGRAM WRITE	Not valid	Not valid	Valid	All	376
QQ	COMPOUND COMMAND	Valid	Valid	Valid	All	376
XZ	ABORT (command only)	Valid	Valid	Valid	All	378
**	INITIALIZE (command only)	Valid	Valid	Valid	All	378
IC	Undefined command (response only)	---	---	---	All	379

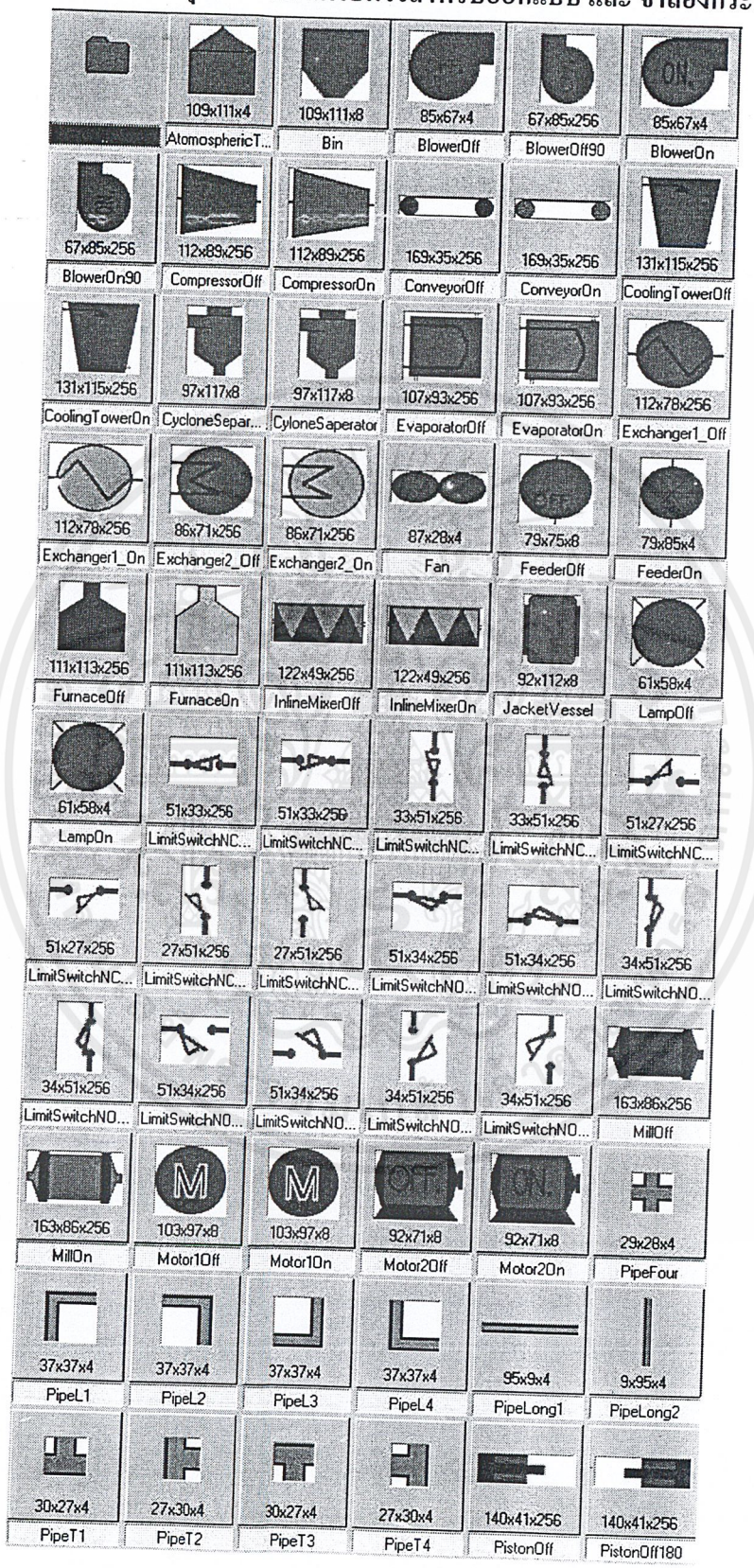
ตารางภาคผนวกที่ 3 แสดงความหมายของ End Code ในการติดต่อสื่อสารแบบ Host Link

End code	Contents	Probable cause	Corrective measures
00	Normal completion	---	---
01	Not executable in RUN mode	The command that was sent cannot be executed when the PC is in RUN mode.	Check the relation between the command and the PC mode.
02	Not executable in MONITOR mode	The command that was sent cannot be executed when the PC is in MONITOR mode.	
04	Address over (CPM1/CPM1A/SRM1 PCs)	The user program area's highest address was exceeded.	Check the program.
0B	Not executable in PROGRAM mode	The command that was sent cannot be executed when the PC is in PROGRAM mode.	This code is not presently being used.
13	FCS error	The FCS is wrong. Either the FCS calculation is mistaken or there is adverse influence from noise.	Check the FCS calculation method. If there was influence from noise, transfer the command again.
14	Format error	The command format is wrong.	Check the format and transfer the command again.
15	Entry number data error	The areas for reading and writing are wrong.	Correct the areas and transfer the command again.
16	Command not supported	The specified command does not exist in the specified address. (Reading the SV, etc.)	Check the address and instruction.
18	Frame length error	The maximum frame length was exceeded.	Divide the command into multiple frames.
19	Not executable	Items to read not registered for composite command (QQ).	Execute QQ to register items to read before attempting batch read.
23	User memory write-protected	CQM1 PCs: Pin 1 on the CQM1 DIP switch is ON. CPM1/CPM1A/SRM1 PCs: The memory is write-protected in the PC Setup.	CQM1 PCs: Turn OFF pin 1. CPM1/CPM1A/SRM1 PCs: Change the setting in the PC Setup (DM 6602).
A3	Aborted due to FCS error in transmit data	The error was generated while a command extending over more than one frame was being executed. Note: The data up to that point has already been written to the appropriate area of the CPU Unit.	Check for corrupted frames, correct if necessary, and try the transfer again.
A4	Aborted due to format error in transmit data		
A5	Aborted due to entry number data error in transmit data		
A8	Aborted due to frame length error in transmit data		
Other	---	Influence from noise was received.	Transfer the command again.

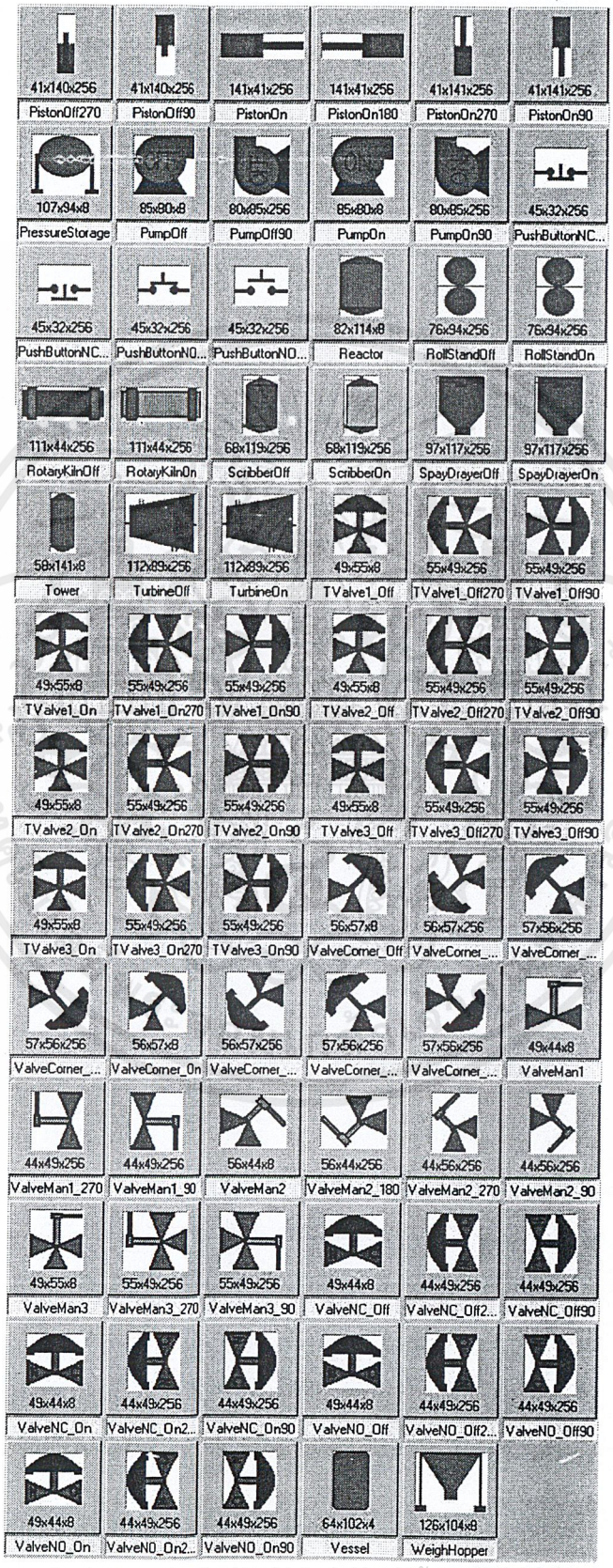
ตารางภาคผนวกที่ 4 แสดงคำสั่งที่เป็นฟังก์ชันที่ใช้เขียนแลดเดอร์ไดอะแกรม

Left digit	Right digit									
	0	1	2	3	4	5	6	7	8	9
0	NOP NO OPERATION	END END	IL INTERLOCK	ILC INTERLOCK CLEAR	JMP JUMP	JME JUMP END	(@)FAL FAILURE ALARM AND RESET	FALS SEVERE FAILURE ALARM	STEP STEP DEFINE	SNXT STEP START
1	SFT SHIFT REGISTER	KEEP KEEP	CNTR REVERS-IBLE COUNTER	DIFU DIFFEREN-TIATE UP	DIFD DIFFEREN-TIATE DOWN	TIMH HIGH-SPEED TIMER	(@)WSFT WORD SHIFT	(@)ASFT ASYNCHRO-NOUS SHIFT REGISTER	-	-
2	CMP COMPARE	(@)MOV MOVE	(@)MVN MOVE NOT	(@)BIN BCD TO BINARY	(@)BCD BINARY TO BCD	(@)ASL SHIFT LEFT	(@)ASR SHIFT RIGHT	(@)ROL ROTATE LEFT	(@)ROR ROTATE RIGHT	(@)COM COMPLE-MENT
3	(@)ADD BCD ADD	(@)SUB BCD SUBTRACT	(@)MUL BCD MULTIPLY	(@)DIV BCD DIVIDE	(@)ANDW LOGICAL AND	(@)ORW LOGICAL OR	(@)XORW EXCLUSIVE OR	(@)XNRW EXCLUSIVE NOR	(@)INC INCREMENT	(@)DEC DECRE-MENT
4	(@)STC SET CARRY	(@)CLC CLEAR CARRY	-	-	-	-	(@)MSG MESSAGE DISPLAY	-	-	-
5	(@)ADB BINARY ADD	(@)SBB BINARY SUBTRACT	(@)MLB BINARY MULTIPLY	(@)DVB BINARY DIVIDE	(@)ADDL DOUBLE BCD ADD	(@)SUBL DOUBLE BCD SUBTRACT	(@)MULL DOUBLE BCD MULTIPLY	(@)DIVL DOUBLE BCD DIVIDE	-	-
6	CMPL DOUBLE COMPARE	(@)INI MODE CON-TROL	(@)PRV HIGH-SPEED COUNTER PV READ	(@)CTBL COMPARI-SON TABLE LOAD	-	-	-	(@)BCNT BIT COUNTER	(@)BCMP BLOCK COMPARE	(@)STIM INTERVAL TIMER
7	(@)XFER BLOCK TRANSFER	(@)BSET BLOCK SET	-	(@)XCHG DATA EXCHANGE	(@)SLD ONE DIGIT SHIFT LEFT	(@)SRD ONE DIGIT SHIFT RIGHT	(@)MLPX 4-TO-16 DECODER	(@)DMPX 16-TO-4 ENCODER	(@)SDEC 7-SEGMENT DECODER	-
8	(@)DIST SINGLE WORD DISTRIBUTE	(@)COLL DATA COLLECT	(@)MOVB MOVE BIT	(@)MOVD MOVE DIGIT	(@)SFTR REVERS-IBLE SHIFT REGISTER	(@)TCMP TABLE COMPARE	(@)ASC ASCII CONVERT	-	-	(@)INT INTERRUPT CONTROL
9	-	(@)SBS SUBROU-TINE ENTRY	(@)SBN SUBROU-TINE DEFINE	RET SUBROU-TINE RETURN	-	-	-	(@)IORF I/O REFRESH	-	(@)MCRO MACRO

ตารางภาคผนวกที่ 5 แสดงอุปกรณ์ที่จัดเตรียมไว้สำหรับออกแบบ และ จำลองกระบวนการผลิต



ตารางภาคผนวกที่ 5(ต่อ) แสดงอุปกรณ์ที่จัดเตรียมไว้สำหรับออกแบบและจำลองกระบวนการผลิต



กิตติกรรมประกาศ

ในการทำโครงงานปริญญานิพนธ์เล่มนี้ หลายครั้งที่ประสบกับปัญหาต่างๆ มากมายทั้งที่เกี่ยวข้องกับการเขียนโปรแกรม การทำงานกับเครื่องควบคุมแบบโปรแกรมได้ และปัญหาเกี่ยวกับเครื่องคอมพิวเตอร์ แต่ก็ได้รับความช่วยเหลือจากอาจารย์หลายท่านที่ได้คอยแนะนำแนวทางในการแก้ไข และพัฒนาโครงงานโดยตลอด

ขอขอบพระคุณ อ.สุเธียร เกียรติสุนทร ซึ่งคอยตรวจสอบงาน ให้แนวคิด และคำแนะนำต่างการในการออกแบบตัวโปรแกรม , อ. เทพจิตร เชยโภคา ที่อธิบายและให้คำปรึกษาเกี่ยวกับการใช้งานเครื่องควบคุมแบบโปรแกรม และ อ.ทวีพล ซื่อสัตย์ อาจารย์ประจำภาควิชาเทคโนโลยีการวัดคุมที่ให้คำปรึกษาเกี่ยวกับการเขียนโปรแกรมด้วยวิชวลเบสิค

และสุดท้ายขอขอบพระคุณผู้ที่เกี่ยวข้องที่คอยให้ความช่วยเหลือต่าง ๆ จนทำให้โครงงานปริญญานิพนธ์เล่มนี้สำเร็จลุล่วงไปได้

คณะผู้จัดทำ

บรรณานุกรม

1. ณรงค์ ตันชีวะวงศ์ , "ระบบ PLC (Programmable Logic Controller)" , สมาคมส่งเสริม-เทคโนโลยีไทยญี่ปุ่น , 405 หน้า , 2542 .
2. กิตติ ภักดีวัฒนะกุล-จำลอง ครูอุตสาหะ , "Visual Basic 6 (ฉบับบโปรแกรมเมอร์)" , บ.ดวงกมลสมัย จำกัด , 621 หน้า , 2542 .
3. สุรศักดิ์ พงศ์ธนาพานิช , "Visual Basic 5.0 (Professional) " , บริษัท ซีเอ็ดยูเคชัน จำกัด , 1144 หน้า , 2541.
4. จิรศักดิ์ เหลืองอุไร , "คัมภีร์การใช้งานการสื่อสารอนุกรมบน PC , บริษัท ซีเอ็ดยูเคชัน จำกัด , 395 หน้า , 2538.
5. OMRON , "SYSMAC CQM1/CPM1/SRM1 Programmable Controller (Programming Mamual)", OMRON,1998