

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

ฐานข้อมูลสำหรับแบบจำลองคลังจัดเก็บสินค้าและการบริหารคงคลัง
(Database For Warehouse Model And Inventory Management)



นางสาวบังอรศิริ หุ่นจันทร์
นายอรินทน์ ไม้รอด

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
สาขาวิศวกรรมการวัดคุม
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2542

เลขหน้.....
เลขทะเบียน 36818
วัน, เดือน, ปี 29 ส.ค. 2543

เอกสารนี้เป็นเอกสารสงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
โดยไม่ได้รับอนุญาต หากฝ่าฝืนให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญานิพนธ์ปีการศึกษา 2542

ภาควิชาเทคโนโลยีการวัดคุมทางอุตสาหกรรม

สาขาวิชาวิศวกรรมการวัดคุม

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง ฐานข้อมูลสำหรับแบบจำลองคลังจัดเก็บสินค้าและการบริหารคงคลัง

DATABASE FOR WAREHOUSE MODEL AND INVENTORY MANAGEMENT

ผู้จัดทำ

นางสาวบังอรศิริ

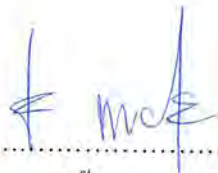
หุ่ณจันทร์

รหัสประจำตัว 39014276

นายอรินทม์

ไม้รอด

รหัสประจำตัว 39014662


.....อาจารย์ที่ปรึกษา
(อาจารย์เชื้อ นกอยู่)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ฐานข้อมูลสำหรับแบบจำลองคลังจัดเก็บสินค้าและการบริหารคลัง

บังอรศิริ หุ่นจันทร์

อรินทน์ ไม้รอด

อ. เชื้อ นกอยู่ อาจารย์ที่ปรึกษา

ปีการศึกษา 2542

บทคัดย่อ

ในปฏิญานิพนธ์ฉบับนี้ เป็นการนำเสนอการประยุกต์การใช้งานของเครื่องควบคุมแบบโปรแกรมได้ (Programmable Logic Controller) ใช้ต่อควบคุม แบบจำลองคลังจัดเก็บสินค้า (Warehouse Model) โดยมีการจัดการฐานข้อมูล (Database) ด้วยคอมพิวเตอร์ ซึ่งจะใช้โปรแกรมไมโครซอฟท์แอคเซส (Microsoft Access) ควบคู่ไปกับการใช้โปรแกรมไมโครซอฟท์ทวิซเวลเบสิก (Microsoft Visual Basic) เพื่อสร้างหน้าจอติดต่อสื่อสารกับผู้ใช้งาน ในการเก็บสินค้าแต่ละครั้งจะเป็นการเก็บ ประเภทของสินค้า วัน เวลาที่จัดเก็บ ซึ่งจะบันทึกลงฐานข้อมูล และการจัดเก็บสินค้าเหล่านี้จะทำการจัดเก็บแบบ “เข้าก่อน-ออกก่อน” (First In - First Out) เพื่อเป็นการป้องกันสินค้าที่เข้ามาเก็บก่อนถูกเก็บไว้นานจนเสียหาย และที่สำคัญเป็นการนำหลักการบริหารคลังสินค้า (Inventory Management) มาใช้ในเรื่องของพื้นที่ในการใช้สอยให้เกิดประโยชน์สูงสุด เพื่อลดต้นทุนและค่าใช้จ่ายในการจัดเก็บสินค้า

Database For Warehouse Model And Inventory Management

MissBungornsiri Hunchan 39014276

Mr. Arin Myerod 39014662

Mr. Chuae Nokyoo

1999

ABSTRACT

This thesis presents an application of programmable logic controller to control the warehouse. The database can be managed by microsoft access program. The presented graphic of warehouse on computer monitor can be achieved by microsoft visual basic. When the product is stored in the warehouse, the program will be updated the database. This system will work with the standard of FIFO (First In-First Out) in order to protect the expiring period of products. The principle objective of the system is to save the warehouse area and decrease the cost of product storage.

สารบัญ

	หน้า
บทที่ 1 บทนำ	
1.1 ที่มาของโครงการ	1
1.2 วัตถุประสงค์ของโครงการ	1
1.3 ขอบเขตของโครงการ	1
1.4 รายละเอียดของโครงการ	2
บทที่ 2 หลักการและทฤษฎี	
2.1 เครื่องควบคุมแบบโปรแกรมได้ (Programmable Logic Control)	3
2.1.1 Introduction to Programmable Logic Control	3
2.1.2 Programmable Logic Control Modicon 512/00 Micro	4
2.2 โปรแกรมควบคุม Modicon 512/00 Micro PLC	9
2.3 การจัดการฐานข้อมูลโดย MS-Visual Basic และ MS-Access	18
2.3.1 Microsoft Visual Basic	18
2.3.2 Microsoft Access	23
2.3.3 การจัดการฐานข้อมูล	23
2.4 การติดต่อสื่อสาร	30
บทที่ 3 รายละเอียดในการสร้างและประกอบโครงการ	
3.1 ฮาร์ดแวร์	41
3.1.1 คลังจัดเก็บสินค้าจำลอง	41
3.1.2 PLC Modicon 512/00 Micro PLC	49
3.2 ซอร์ฟแวร์	49
3.2.1 แผนผังขั้นบันได (PLC Ladder Program)	49
3.2.2 โปรแกรมสำหรับจัดการฐานข้อมูลและติดต่อสื่อสารกับผู้ใช้งาน	52
บทที่ 4 การทดลองโครงการ	53
บทที่ 5 สรุปผลการทำงานของโครงการและแนวทางในการพัฒนา	60
บรรณานุกรม	
กิตติกรรมประกาศ	
ภาคผนวก	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1

บทนำ

1.1 ที่มาของโครงการ

การบริหารขององค์กรนับได้ว่าเป็นสิ่งที่มีความสำคัญมากในเกือบจะทุกๆ ประเภทของธุรกิจ ไม่ว่าจะเป็นธุรกิจที่เกี่ยวข้องกับสินค้า หรือการให้บริการ หน่วยงานจะต้องรับผิดชอบในการควบคุมต้นทุนค่าใช้จ่ายในการดำเนินงาน ซึ่งต้นทุนชนิดหนึ่งซึ่งเพิ่มขึ้นมาอย่างหนึ่ง ก็คือ ค่าแรง ค่าสวัสดิการคนงาน ค่าใช้จ่ายสำหรับพื้นที่ในการจัดเก็บ เมื่อมีการใช้พื้นที่มากเกินไปในการดูแลรักษาขององค์กร ซึ่งค่าใช้จ่ายเหล่านี้สามารถลดหรือตัดออกได้บางส่วน ด้วยการนำเทคโนโลยีสมัยใหม่เข้ามาช่วย

ในปัจจุบันเทคโนโลยีทางด้านคอมพิวเตอร์มีความเจริญก้าวหน้ามาก ดังนั้นจึงได้ถูกนำมาประยุกต์ใช้แทนแรงงานคนอย่างแพร่หลายกับระบบโรงงานอัตโนมัติ ซึ่งจะมีผลทำให้ระบบควบคุมมีความผิดพลาดน้อยลงประหยัดต้นทุนการผลิต รวดเร็ว สามารถทำงานภายใต้สภาวะที่เป็นพิษได้ ในคลังจัดเก็บพัสดุก็เช่นกัน เพื่อจัดการปัญหาการต้องจ้างแรงงานขนถ่ายสินค้า การเสียเวลาในการบันทึกและค้นหาตำแหน่งสินค้า และยังรวมไปถึงการบริหารพื้นที่ใช้สอยในการจัดเก็บให้เกิดประโยชน์สูงสุด เครื่องควบคุมแบบโปรแกรมได้จึงถูกนำมาใช้ในโครงการนี้ ซึ่งเป็นตัวควบคุมการเคลื่อนที่ของคลังจัดเก็บสินค้าจำลองคอมพิวเตอร์ถูกนำมาเป็นตัวจัดการกับฐานข้อมูล และติดต่อกับผู้ใช้งาน

1.2 วัตถุประสงค์ของโครงการ

1. เพื่อศึกษาและประยุกต์ใช้งานเครื่องควบคุมแบบโปรแกรมได้ ควบคุมไปกับการจัดการฐานข้อมูล โดยอาศัยแบบจำลองคลังจัดเก็บสินค้า (Warehouse Model) เป็นตัวอย่างในการเก็บข้อมูล
2. เพื่อศึกษาการจัดการฐานข้อมูล (Database) โดยใช้โปรแกรมไมโครซอฟท์เอกเซล (Microsoft Access) ควบคู่กับไมโครซอฟท์วิชวลเบสิก (Microsoft Visual Basic)

1.3 ขอบเขตของโครงการ

ในโครงการนี้ได้ศึกษาเกี่ยวกับการจัดการฐานข้อมูล โดยใช้โปรแกรม Microsoft Access ควบคู่กับการใช้โปรแกรม Microsoft Visual Basic เพื่อสร้างหน้าจอคอมพิวเตอร์เพื่อติดต่อกับผู้ใช้งาน และยังได้ศึกษาเกี่ยวกับเครื่องควบคุมแบบโปรแกรมได้ และโครงสร้างของคลังจัดเก็บสินค้า เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จำลองเพื่อนำมาใช้ร่วมกัน เพื่อควบคุมการเคลื่อนที่ รับ-ส่ง พัดจุของคลังจัดเก็บสินค้าจำลอง ตามการใช้งานของผู้ใช้ซึ่งส่งงานจากหน้าจอคอมพิวเตอร์ส่วนบุคคล และพัฒนาการจัดเก็บสินค้าลงในฐานข้อมูลเป็นแบบ เข้าก่อน-ออกก่อน (First In-First Out) และข้อมูลที่จัดเก็บเป็น ประเภทของผลิตภัณฑ์ และวัน เวลา ที่จัดเก็บ

1.4 รายละเอียดของโครงการ

ในรายงานฉบับนี้ได้แบ่งเนื้อหาออกเป็นส่วนๆ ดังนี้คือ

บทที่ 1 บทนำ เป็นรายละเอียดเกี่ยวกับ ที่มาของโครงการ, วัตถุประสงค์ของโครงการ, ขอบเขตของโครงการ และรายละเอียดของโครงการ

บทที่ 2 หลักการและทฤษฎี

บทที่ 3 การสร้างและประกอบโครงการ

บทที่ 4 การทดลองโครงการ

บทที่ 5 สรุปผลการทำงานและแนวทางในการพัฒนา

บทที่ 2

เครื่องควบคุมแบบโปรแกรมได้และคลังจัดเก็บสินค้าจำลอง

2.1 เครื่องควบคุมแบบโปรแกรมได้ (PROGRAMMABLE LOGIC CONTROL: PLC)

2.1.1 INTRODUCTION TO PROGRAMMABLE LOGIC CONTROL

ระบบควบคุมแบบซีควেনซ์ (Sequence Control) แต่เดิมประกอบด้วยอุปกรณ์ไฟฟ้าเชิงกล (Electromechanical Device) ซึ่งได้แก่ วงจรรีเลย์ (Relay), ตัวตั้งเวลา (Timer), ตัวนับ (Counter) แต่เดิมประกอบด้วยอุปกรณ์ไฟฟ้าการทำงานของระบบควบคุมนี้มีอยู่ 2 สถานะด้วยกันคือ เปิดกับปิด (On, Off) ระบบซีควেনซ์ที่ประกอบด้วยอุปกรณ์ดังกล่าวนี้มีขนาดใหญ่ใช้กำลังงานสูง และเมื่อมีการเปลี่ยนแปลงการทำงานก็ต้องทำการสร้างวงจรควบคุมใหม่ ซึ่งไม่สามารถประยุกต์ใช้กับระบบควบคุมที่มีความยุ่งยากและซับซ้อน เหล่านี้เป็นต้น

เนื่องจากเทคโนโลยีทางด้านไมโครโพรเซสเซอร์ (Microprocessor) ในปัจจุบันได้เจริญก้าวหน้าไปมาก มีการนำไมโครโพรเซสเซอร์มาใช้ในการควบคุมแบบต่อเนื่อง ซึ่งก็คือเครื่องควบคุมแบบโปรแกรมได้ ถูกนำมาใช้ในการควบคุมสำหรับงานด้านอุตสาหกรรมโดยเฉพาะ

เครื่องควบคุมแบบโปรแกรมได้ จะมีส่วนที่เป็นอินพุตที่สามารถต่อใช้งานได้กับ ตัวตรวจจับต่างๆ (Sensor) และส่วนเอาต์พุตจะต่อไปควบคุมการทำงานของอุปกรณ์ หรือเครื่องจักรกลโดยสามารถสร้างวงจรหรือเงื่อนไข การทำงานของเครื่องจักรเหล่านี้ได้จากการป้อนโปรแกรมลงงานเป็นภาษาที่เรียกว่า แผนผังขั้นบันได (Ladder Diagram) เข้าไป โปรแกรมนี้จำทำหน้าที่เหมือนกับวงจรรีเลย์ ตัวตั้งเวลา ตัวนับ และอื่นๆอีก

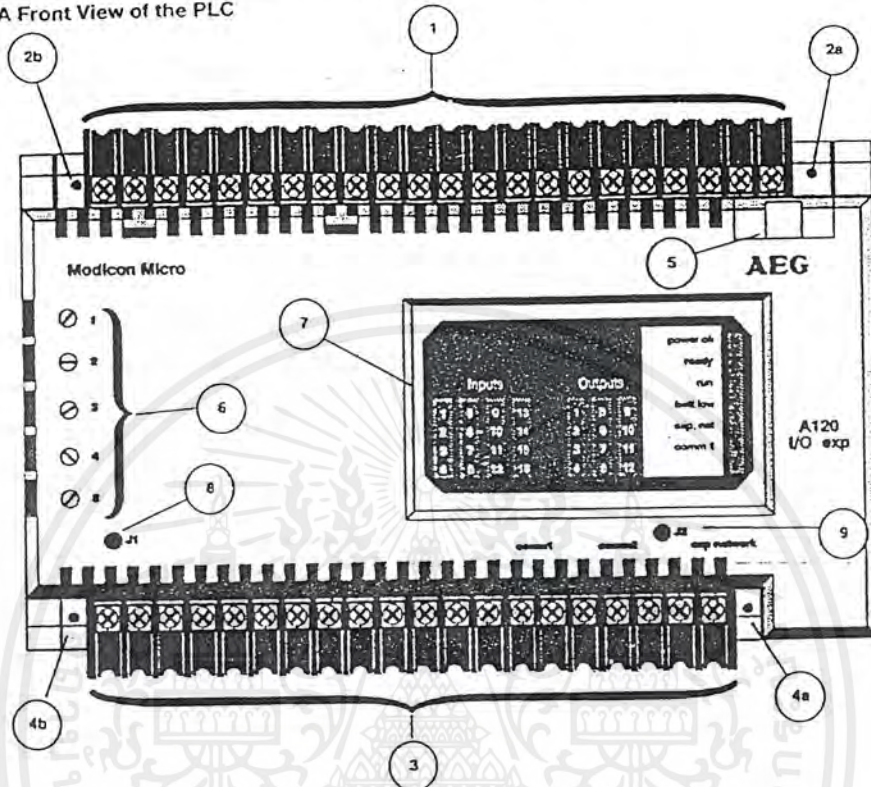
นอกจากนี้ เครื่องควบคุมแบบโปรแกรมได้ที่มีให้เลือกใช้อยู่ในปัจจุบัน มีการออกแบบระบบให้มีความยืดหยุ่นสูงให้สามารถจะต่อร่วมกับอุปกรณ์ภายนอกต่างๆ ทั้งนี้ก็เพื่อเป็นการเพิ่มประสิทธิภาพการทำงานของเครื่องควบคุมแบบโปรแกรมได้ให้มีสมรรถนะสูงขึ้น

ในโครงงานนี้ได้ใช้เครื่องควบคุมแบบโปรแกรมได้เป็น MODICON 512/00 MICRO PLC มีส่วนต่างๆแสดงไว้เห็นดังนี้

2.1.2 PROGRAMMABLE LOGIC CONTROL MODICON 512/00 MICRO

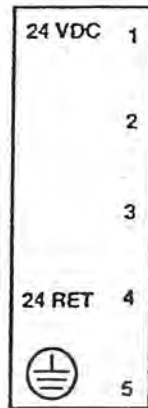
ลักษณะทั่วไปของ MODICON 512/00 MICRO PLC

A Front View of the PLC



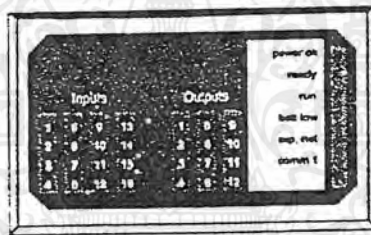
รูปที่ 2.1 ลักษณะทั่วไปของ MODICON 512/00 MICRO PLC

1. ส่วนขั้วต่อทางด้าน INPUT จะประกอบด้วยขั้วต่อ INPUT 16 ขั้ว และขั้วต่อ COMMON 2 ขั้ว ใช้ไฟ 24 VDC ทางด้าน INPUT
2. A/B ส่วนขั้วต่อทางด้าน INPUT
3. ส่วนขั้วต่อทางด้าน OUTPUT จะประกอบด้วยขั้วต่อ OUTPUT 12 ขั้ว ซึ่งต่อ COMMON 2 ขั้ว ใช้ไฟ 8 VDC ทางด้าน OUTPUT
4. A/B ส่วนขั้วต่อทางด้าน OUTPUT
5. ส่วนที่ใส่ BATTERY
6. ส่วนขั้วต่อไฟเลี้ยง ประกอบด้วย ไฟ +24, -24 VDC และขั้วต่อกราวด์ดังรูปที่ 2.2



รูปที่ 2.2 ขั้วต่อไฟ SUPPLY

7. ส่วนแสดงผลของ CPU จะเป็น LED ที่ใช้แสดงผลเหมือน PLC โดยทั่วไป ความหมายต่างๆจะได้ทราบดังต่อไปนี้



รูปที่ 2.3 แสดง LED ในตำแหน่งต่างๆ

หมวดแสดงผล

POWER OK

จะติดเมื่อจ่ายไฟให้กับ CPU

READY

จะติดเมื่ออุปกรณ์ภายใน PLC พร้อมทุกส่วน

RUN

จะติดเมื่อ CPU ทำงาน

BATT LOW

จะติดเมื่อ BATTERY ต้องการเก็บพลังงานเพิ่ม

EXP.NET

จะติดเมื่อมีการติดต่อทาง PORT I/O EXP LINK

COMM1

จะติดเมื่อมีการติดต่อทาง PORT COMM1

COMM2

จะติดเมื่อมีการติดต่อทาง PORT COMM2

INPUT

จะติดเมื่อมีไฟเลี้ยงที่ INPUT

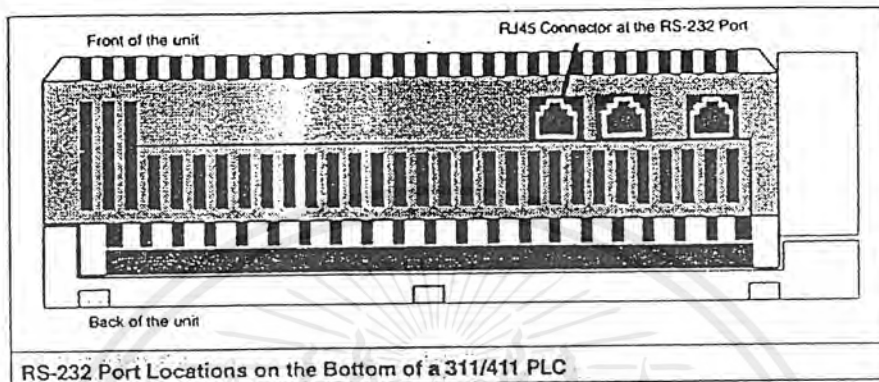
OUTPUT

จะติดเมื่อมีการสั่งงานให้ OUTPUT ทำงาน

8. ส่วนปรับระดับ GROUND TO CHASSIS GROUND

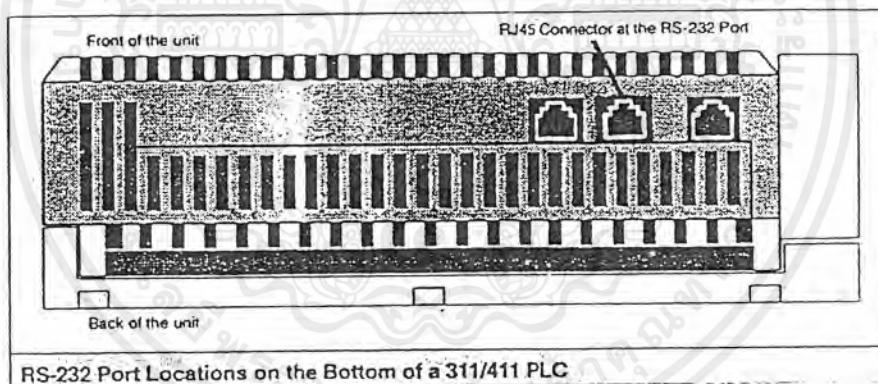
9. ส่วนปรับความต้านทานของ PLC

10. PORT COMM1 ใช้ RS-232 ในการติดต่อและใช้หัว CONNECT แบบ RJ45 ดังรูปที่ 2.4



รูปที่ 2.4 แสดง PORT COMM1

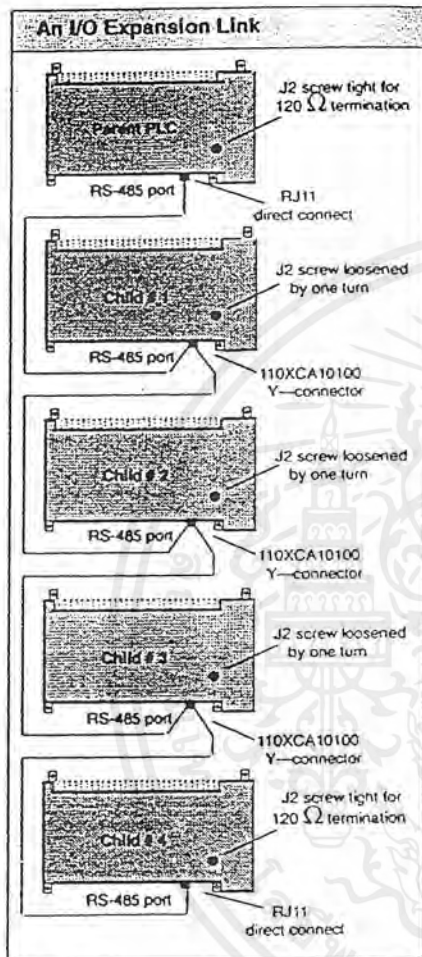
11. PORT COMM2 ใช้ RS-232 ในการติดต่อและใช้หัว CONNECT แบบ RJ45 ดังรูปที่ 2.5



รูปที่ 2.5 แสดง PORT COMM2

12. PORT EXP NETWORK

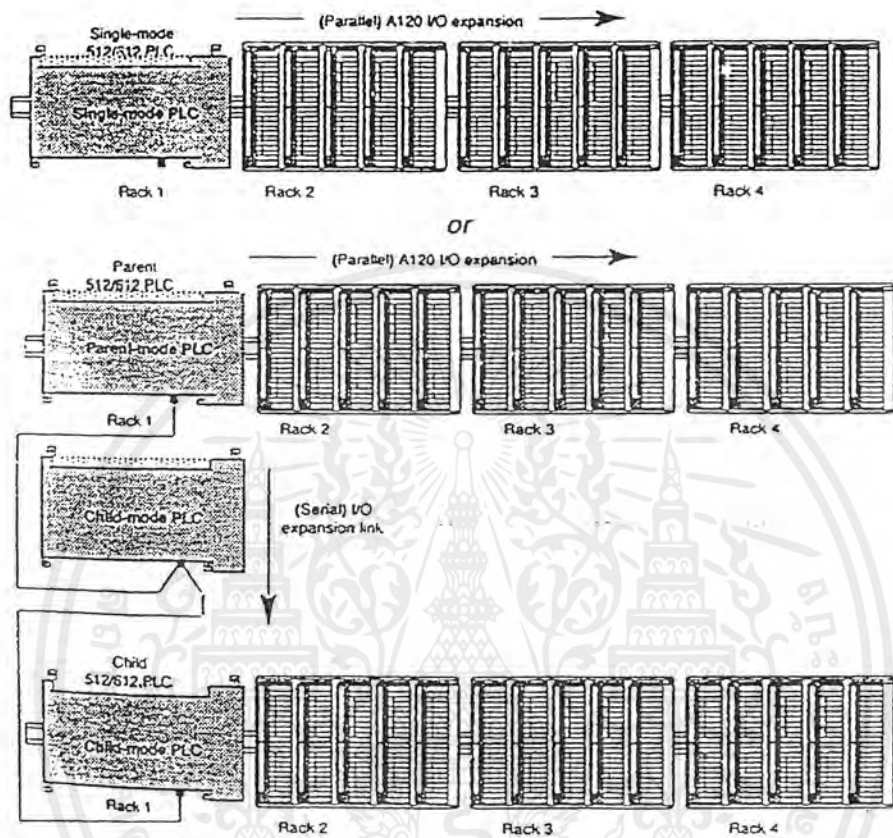
I/O EXPANSION LINK มีไว้สำหรับต่อ
PARENT PLC กับ CHILD 1-4 ตัว ไว้สำหรับใช้
งานด้วยกันและใช้หัว CONNECT แบบ RJ 11 ใน
การต่อกับ PORT โดยใช้สาย RS-485 ในการใช้งาน
ดังรูปที่ 2.6



รูปที่ 2.6 แสดง I/O EXPANSION LINK

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

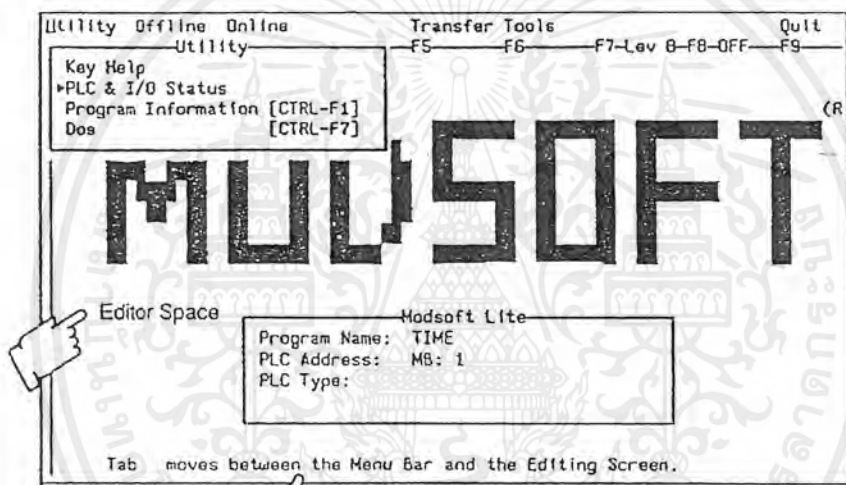
13. PORT A120 I/O EXP



รูปที่ 2.7 แสดง PORT A102 I/O EXP

2.2 โปรแกรมควบคุม MODICON 512/00 MICRO PLC

โดยทั่วไปการที่เราจะใช้งานตัว PLC เราจะต้องทำการเขียน แผนผังขั้นบันได (Ladder Diagram) ให้กับตัว PLC โดยการคีย์โปรแกรมเข้าไป ซึ่งสามารถที่จะทำได้ทั้งการเขียนโดยเครื่อง คีย์บันทึกภายนอก หรือโดยเขียนทางโปรแกรมโดยเฉพาะของ PLC ของแต่ละผู้ผลิตที่ต่างกัน แต่ สำหรับ MODICON PLC นั้นจะทำได้โดยการเขียนทางโปรแกรมโดยเฉพาะของตัวเอง ซึ่งจัดได้ว่าเป็นโปรแกรมมาตรฐานตัวหนึ่งที่สามารถนำมาใช้ได้กับ PLC ของ MODICON ทุกรุ่น ซึ่งโปรแกรม ซึ่งโปรแกรมนั้นถูกเรียกว่า MODSOFT และในปัจจุบันก็มีเวอร์ชันด้วยกันในรายงานนี้ใช้ VERSION 2.5 ในการเขียนโปรแกรม ซึ่งมีหน้าจอดังรูป 2.8



รูปที่ 2.8 โปรแกรม MODSOFT VERSION 2.5

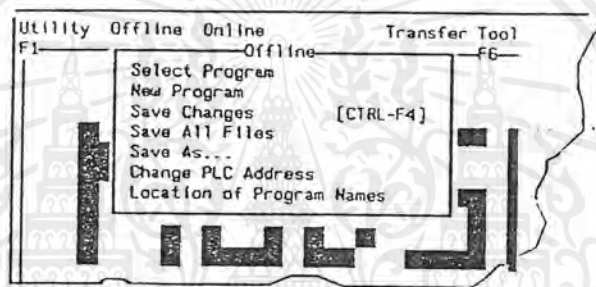
จากรูปหน้าจอจะแบ่งเป็น 3 ส่วน คือ

1. MAIN MENU LINE คือ ส่วนของเมนูหลักในการใช้ระบบของโปรแกรม
2. EDITOR SPACE คือ ส่วนของพื้นที่ในการเขียนโปรแกรม
3. TRACKING HELP LINE คือ ส่วนที่คอยบอกลักษณะใช้งานในแต่ละเมนู

ในการใช้งานหลักของโปรแกรม MODSOFT มี 3 MAIN MENU คือ

1. OFFLINE คือ ใช้ในการเขียนหรือแก้ไขโปรแกรมต่างๆ ที่เราต้องการนำไปใช้งานให้กับตัว PLC ซึ่งจะเปิดไว้ในตัวโปรแกรม MODSOFT
2. ONLINE คือ ใช้ในการติดต่อกับ PLC โดยต่อกับคอมพิวเตอร์ เราสามารถที่จะดูโปรแกรมในตัว PLC ที่เราติดต่อยู่ได้และยังสามารถแก้ไขได้ทันทีตามที่เราต้องการ
3. TRANSFER คือ ใช้ในการจัดบันทึกโปรแกรมทั้งจาก PLC มาสู่โปรแกรม MODSOFT หรือจะบันทึกจากโปรแกรม MODSOFT ไปสู่ PLC ก็ได้

ดังนั้นในการเขียนโปรแกรมที่เราต้องการ เราจะต้องใช้งานจะเริ่มเขียนในเมนู OFFLINE โดยในเมนู OFFLINE ประกอบด้วยดังรูปที่ 2.9



รูปที่ 2.9เมนู OFFLINE

SELECT PROGRAM	มีไว้สำหรับเลือกโปรแกรมที่เราเก็บไว้ในโปรแกรม MODSOFT
NEW PROGRAM	มีไว้สำหรับสร้างโปรแกรมใหม่
* SAVE CHANGE	มีไว้สำหรับบันทึกโปรแกรมที่มีการแก้ไข
* SAVE ALL FILE	มีไว้สำหรับบันทึกทั้งโปรแกรมใหม่หรือที่มีการแก้ไข
* SAVE AS	มีไว้สำหรับบันทึกโปรแกรมใหม่พร้อมตั้งชื่อใหม่
*CHANGE PLC ADDRESS	มีไว้สำหรับ SET ค่าต่างๆ ในการติดต่อสื่อสารกับ PLC โดยจะมีค่าต่างๆ ให้ SET ดังรูปที่ 2.10
* LOCATION OF PROGRAM NAME	มีไว้สำหรับบอกที่อยู่ของโปรแกรมที่เลือกใช้อยู่ ดังนั้นการเริ่มใช้งานเราจะต้องสร้างโปรแกรมใหม่ก่อนโดยใช้เมนูย่อย NEW PROGRAM ดังรูปที่ 2.11

Address = 1-247

Protocol = Modbus 1

Modbus Plus

Default *

Mode = ASCII (7 data bits) RTU (8 data bits)

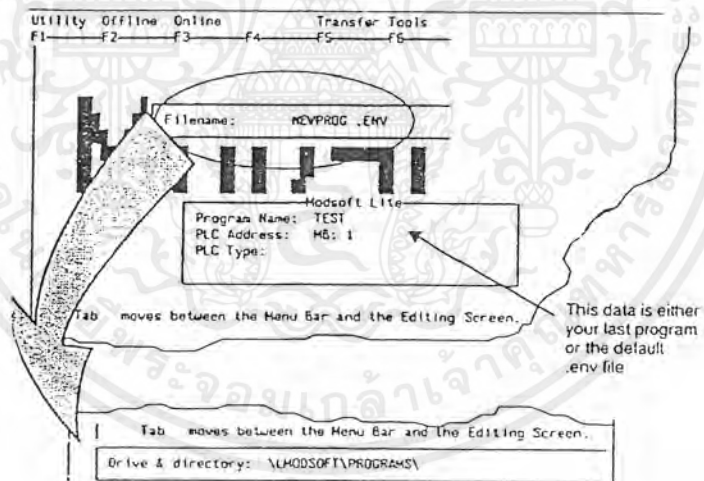
Parity = NONE EVEN ODD

Stop Bits = 1 or 2

Baud	=	50	75	110	134
		150	300	600	1200
		1800	2000	2400	3600
		4800	7200	9600	19200

Device COM1 through COM8

รูปที่ 2.10 แสดง CHANGE PLC ADDRESS



รูปที่ 2.11 การสร้างโปรแกรมใหม่โดยใช้เมนูย่อย NEW PROGRAM

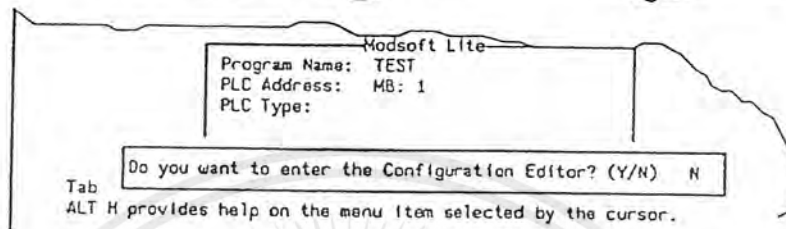
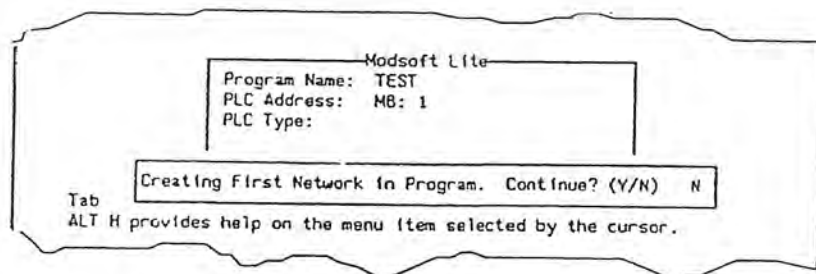
เราจะเริ่มต้นโดยการสร้างชื่อโปรแกรมใหม่ เมื่อสร้างชื่อของโปรแกรมเสร็จแล้วโปรแกรมจะให้เรา SET ค่าต่างๆดังรูปที่ 2.12

The screenshot shows the MODSOFT utility menu with the following options: Utility, Offline, Online, Transfer Tools, and Quit. The menu is navigated using function keys F1 through F9. The 'Program Name' is set to 'NEWPROG' and the 'PLC Address' is 'M8: 1'. The 'COMMUNICATIONS PARAMETERS' section shows: Address 1, Protocol MODBUS I, Mode RTU, Data Bits 8, Parity EVEN, Stop Bits 1, Baud 9600, and Device COM1. A 'Dip Switches' section shows a barcode and the number 12345678.

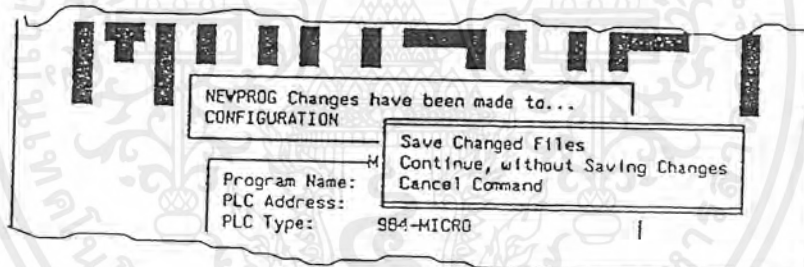
รูปที่ 2.12 แสดงค่าพารามิเตอร์

The screenshot shows the 'CONFIGURATION OVERVIEW' screen. It displays the following information: PLC Type: 984 - MICRO-S, Model: S12/88, Memory: 3.1K, Micro Child ID: None. The 'I/O' section shows: Size of Full Logic Area: 82298, No. of I/O Map Words: 88849, Number of Segments: 2, Number of Children: 8, and I/O Locations: 5. The 'Ranges' section shows: 0xxxx: 00001 - 00512, 1xxxx: 10001 - 10512, 3xxxx: 30001 - 30512, and 4xxxx: 40001 - 40512. The 'Specials' section shows: Battery Coil: 0----, Timer Register: 4----, and Time of Day Clock: 4----. A note at the bottom states: 'I/O Map is the traffic cop which links the I/O modules to program logic.'

รูปที่ 2.13 แสดงค่า CONFIGURATION



รูปที่ 2.14 แสดงการยืนยันการ SET ค่าใหม่



รูปที่ 2.15 การ SAVE โปรแกรมใหม่

เมื่อ SET ค่าต่างๆตามที่เรากำลังต้องการแล้ว เราก็สามารถใช้ LADDER LOGIC เพื่อนำไปควบคุมการทำงานตามที่เรากำลังต้องการได้

การ SET ค่า CONFIGURATION ดังรูปที่ 2.16

Utility	F1	F2	F3	F4	F5	F6	F7-Lev	F8-Off	F9	Quit
OVERVIEW I/Q MAP Ports Loadable										
CONFIGURATION OVERVIEW										
PLC :	Size of Full Logic Area						82258			
PLC Type	984 - MICRO-S						I/O Map Words		88049	
Model	S12/88						COMPACT		MICRO	
Memory	3.1K						Number of Segments		2	
Micro Child I/O	None						Number of Children		8	
							I/O Locations		5	
Specials :										
Battery Call							8----			
Timer Register							4----			
Time of Day Clock							4----			
Ranges :										
Bknw : 00001 - 01024										
1K: CONFIGURATION OVERVIEW										
Size of Full I/O K										
2.1K										
K										
MICRO										
311/88										
311/81										
311/82										
311/83										
411/88										
411/81										
411/82										
411/83										
512/88										
512/81										
512/82										
512/83										
612/88										
612/83										
81284										
18256										
38832										
48488										
None										
CONFIGURATION OVERVIEW										
Size of Full I/O K										
-COMPACT										
3.1K										
K										
8128										
8138										
8145										
4128										
413X										
4141										
4145										
8128										
8138										
8145										
4128										
413X										
4141										
4145										
8128										
8138										
8145										
4128										
413X										
4141										
4145										
8128										
8138										
8145										
4128										
413X										
4141										
4145										
8128										
8138										
8145										
4128										
413X										
4141										
4145										
8128										
8138										
8145										
4128										
413X										
4141										
4145										
8128										
8138										
8145										
4128										
413X										
4141										
4145										
8128										
8138										
8145										
4128										
413X										
4141										
4145										
8128										
8138										
8145										
4128										
413X										
4141										
4145										
8128										
8138										
8145										
4128										
413X										
4141										
4145										
8128										
8138										
8145										
4128										
413X										
4141										
4145										
8128										
8138										
8145										
4128										
413X										
4141										
4145										
8128										
8138										
8145										
4128										
413X										
4141										
4145										
8128										
8138										
8145										
4128										
413X										
4141										
4145										
8128										
8138										
8145										
4128										
413X										
4141										
4145										
8128										
8138										
8145										
4128										
413X										
4141										
4145										
8128										
8138										
8145										
4128										
413X										
4141										
4145										
8128										
8138										
8145										
4128										
413X										
4141										
4145										
8128										
8138										
8145										
4128										
413X										
4141										
4145										

รูปที่ 2.16 การ SET ค่า CONFIGURATION

คือ จะต้อง SET ค่า ชนิดของ PLC จะมี 2 ชนิด คือ 1. COMPACT, 2. MICRO โดยในแต่ละชนิดจะมีให้เลือก CPU แต่ละชนิดดังรูป ในรายงานนี้เราต้องเลือกชนิด MICRO และรุ่น CPU 512/00 และยังจะต้อง SET ค่า PORT อีกดังรูปที่ 2.17

```

Utility      Default Bridge      Quit
F1-----F2-----F3-----F4-----F5-----F6-----F7-Lev B-F8-OFF-----F9-----

MICRO PORTS

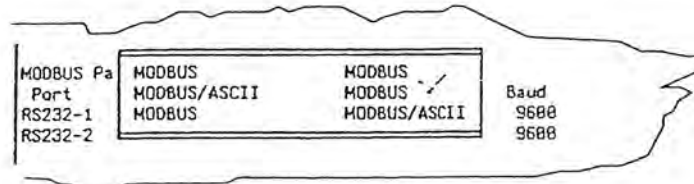
MODE          RS232-1          RS232-2          RS485
Parent        MODBUS

Port          Mode    Data Bits  Parity  Stop Bits  Baud    Address  Delay
RS232-1      RTU       8          EVEN     1          9600    1         10 ms
RS232-2      RTU       8          EVEN     1          9600    1         10 ms

          ASCII Parameters
          8          EVEN     1          9600
  
```

รูปที่ 2.17 การ SET ค่า PORT

ดังรูปที่ 2.17 จะมีทั้งการ SET ค่าการติดต่อแต่ละ PORT รวมทั้งรูปแบบการติดต่อสื่อสารในขณะที่ใช้ FUNCTION COMM ซึ่งการ SET ค่าต่างๆ อยู่ในรูปที่ 2.18

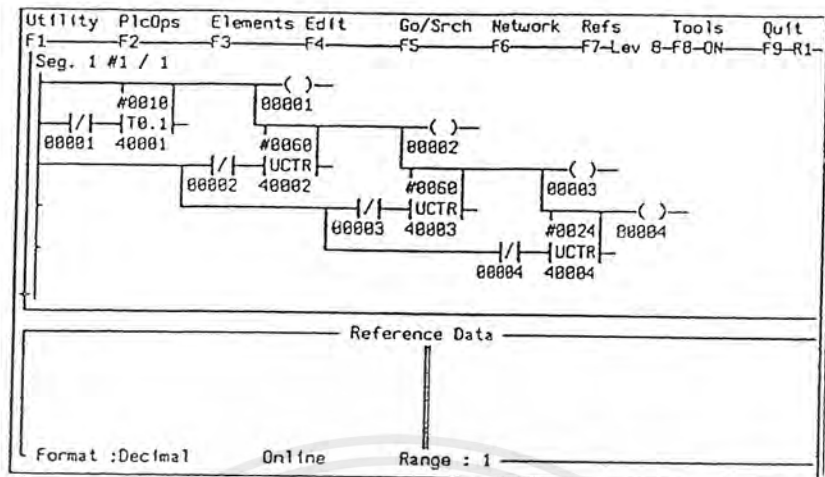


รูปที่ 2.18 รูปแบบการ SET ค่า PORT

Autoconfiguration Parameters for a Single Mode Micro PLC		
Parameter	110CPU Models	
	311 / 411	512 / 612
Number of 0x references	1024	1536
Number of 1x references	256	512
Number of 3x references	32	48
Number of 4x references	400	1872
Number of ladder logic segments	2 (the first for control logic and the second for subroutines)	2 (the first for control logic and the second for subroutines)
RS-232 port (comm 1)	Dedicated Modbus mode: 8-bit RTU communications, 9600 baud, even parity, 1 STOP bit, Modbus address #1	Dedicated Modbus mode: 8-bit RTU communications, 9600 baud, even parity, 1 STOP bit, Modbus address #1
RS-232 port (comm 2)	N/A	Dedicated Modbus mode: 8-bit RTU communications, 9600 baud, even parity, 1 STOP bit, Modbus address #1
RS-485 port (exp. link)	Dedicated ASCII 8-bit ASCII communications, 9600 baud, even parity, 1 STOP bit	Dedicated ASCII 8-bit ASCII communications, 9600 baud, even parity, 1 STOP bit

รูปที่ 2.19 AUTOCONFIGURATION PARAMETERS

เมื่อเราสร้างโปรแกรมใหม่แล้วเข้ามาอยู่ในโปรแกรมที่เราสร้างจะมีเมนูต่างๆ ดังรูปที่ 2.20



รูปที่ 2.20 แสดง LADDER DIAGRAM ของโปรแกรม
เราสามารถเขียน LADDER LOGIC ได้ในเมนู ELEMENTS ในเมนู ELEMENTS จะมี
LADDER LOGIC มากมายดังรูปที่ 2.21

Standard Ladder Logic Instructions (available on all Micro PLCs)	
Instruction	Description
Relay Logic	
	A normally open (N.O.) contact
	A normally closed (N.C.) contact
	A positive transitional contact
	A negative transitional contact
	A normal coil
	A memory-retentive coil
Counters	
UCTR	An up counter from 0 to a specified preset
DCTR	A down counter to 0 from a specified preset
Timers	
T1.0	A timer that increments in seconds
T0.1	A timer that increments in tenths of a second
T.01	A timer that increments in hundredths of a second
T1MS	A timer that increments in ms
Integer Math	
ADD	Addition
SUB	Subtraction or greater than/less than operations
MUL	Multiplication
DIV	Division
Data Move	
R→T	A register-to-table move
T→R	A table-to-register move
T→T	A table-to-table move
BLKM	A block move
FIN	A first-in operation to a queue
FOUT	A first-out operation from a queue
SRCH	A table search for a bit pattern in one of the registers

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Standard Ladder Logic Instructions (continued)	
Instruction	Description
Data Matrix	
AND	A logical AND of two matrices
OR	A logical OR of two matrices
XOR	A logical exclusive OR of two matrices
COMP	A logical complement of the bit pattern in a matrix
CMPR	A logical compare of the bit patterns in two matrices
MBIT	A bit modify—i.e., changing the current (1, 0) value of the bit
SENS	A bit sense—i.e., reporting the current (1, 0) value of the bit
BROT	A bit rotation—i.e., shifting the bit positions left or right in a matrix
ASCII	
COMM	An ASCII read or write communication operation
Sequencing	
SCIF	Drum sequencing and input comparison operations
Subroutines	
JSR	Jumps the logic scan from control logic to a ladder logic subroutine programmed in the last segment
LAB	Labels the entry location for the called subroutine in the last segment
RET	Returns the logic scan to its previous place in logic prior to the JSR
CTIF	Sets up the high-speed inputs for interrupt and counter/timer operations
Other	
STAT	Checks and reports the health of the PLC and its I/O
SKP	Causes the logic scan to skip specified networks in the program

Enhanced Ladder Logic Instructions (available in specified 110CPU512 and 110CPU612 Models only)	
Instruction	Description
BLKT	A block-to-table move
TBLK	A table-to-block move
CKSM	Performs CRC-16, LRC, straight, or binary checksum operations
PID2	Performs proportional-integral-derivative control functions
EMTH	Performs extended math functions such as square root, process square root, log, antilog, and floating point operations

รูปที่ 2.21 STANDARD LADDER LOGIC INSTRUCTIONS

เมื่อเราเขียน LADDER LOGIC ต่างๆ ตามที่เราต้องการได้แล้ว เราจะต้องนำโปรแกรมนี้นั้นที่กลงสู่ MODICON 521/00 MICRO PLC โดยใช้เมนู TRANSFER โดยในเมนูจะประกอบด้วย 3 เมนูย่อย คือ PLC TO FILE, VERIFY ALL และ FILE TO PLC เราสามารถส่งถ่ายข้อมูลจากโปรแกรมสู่ PLC ได้โดยเมนู FILE TO PLC โดยให้เราเลือกชื่อโปรแกรมนั้นที่เราต้องการ และเรา

จะบันทึกโปรแกรมสู่ PLC ได้ตามต้องการ PLC ก็จะเก็บข้อมูลที่เรานำบันทึกไว้ในหน่วยความจำ ทำให้เราสามารถนำ PLC ไปใช้ตามการทำงานที่โปรแกรมไว้ภายในได้

2.3 การจัดการฐานข้อมูลโดย MS-Visual Basic และ MS-Access

2.3.1 Microsoft Visual Basic

โปรแกรม Microsoft Visual Basic 6.0 เป็นเครื่องมือในการพัฒนาโปรแกรมที่ได้รับความนิยมเป็นอย่างมาก เนื่องจากมีความง่ายและสะดวกในการใช้งาน อีกทั้งยังมีเครื่องมือในการเขียนโปรแกรมเป็นจำนวนมาก Microsoft Visual Basic 6.0 เป็นเครื่องมือในการสร้างโปรแกรมบนระบบปฏิบัติการ Windows ที่ใช้งานง่าย โดยการสร้างโปรแกรมนั้นจะเป็นการเลือกเครื่องมือต่างๆ มาออกแบบหน้าจอของโปรแกรมที่จะสร้างขึ้น ซึ่งเรียกการเขียนโปรแกรมในลักษณะนี้ว่า Visual Programming การเขียนโปรแกรมแบบนี้ไม่จำเป็นต้องเขียนคำสั่งต่างๆ มากนัก จึงสามารถสร้างโปรแกรมได้อย่างรวดเร็ว นอกจากนี้ยังสนับสนุนการจัดการกับฐานข้อมูลได้เป็นอย่างดี คำศัพท์ที่ใช้ใน MS-Visual Basic

คอนโทรล(Control)

เป็นเครื่องมืออย่างหนึ่งที่ใช้สร้างออบเจกต์บนฟอร์มของ MS-Visual Basic เลือกคอนโทรลจากทูลบ็อกซ์แล้ววาดคอนโทรลนั้นไว้บนฟอร์ม โดยการใช้เมาส์ เราจะใช้คอนโทรลส่วนใหญ่ในการสร้างยูสเซอร์อินเตอร์เฟซ เช่น Imagebox, Commandbutton

ออบเจกต์ (Object)

ออบเจกต์เป็นชื่อหนึ่งขององค์ประกอบของยูสเซอร์อินเตอร์เฟซที่เราสร้างไว้บนฟอร์มของ MS-Visual Basic โดยใช้คอนโทรลในทูลบ็อกซ์ เราสามารถเคลื่อนย้าย ปรับขนาดและปรับแต่งออบเจกต์ได้โดยการกำหนดคุณสมบัติ ออบเจกต์ยังมีสิ่งที่เรา Inherent Functionality ซึ่งหมายถึงการที่ออบเจกต์รู้ว่าต้องทำอะไรบ้างและสามารถตอบสนองต่อสถานะการณ์ต่างๆ ได้เอง (เช่น Listbox รู้จักวิธีเลื่อน Scrollbox เป็นต้น) และสามารถเขียนโปรแกรม MS-Visual Basic ให้กับออบเจกต์ โดยใช้ไวยกรณ์โปรแกรมที่ปรับแต่งไว้ตามสถานะการณ์ต่างๆ ในโปรแกรมได้ ฟอร์มในวิซวลเบสิกเองก็เป็นออบเจกต์อย่างหนึ่งเหมือนกัน

คุณสมบัติ (Property)

คุณสมบัติเป็นคำหรือลักษณะที่มีอยู่ในออบเจกต์ของวิซวลเบสิก ตัวอย่างเช่น คุณสมบัติ Caption และ Forecolour ซึ่งจะสามารถกำหนดคุณสมบัติในระหว่างออกแบบได้ โดยใช้หน้าต่าง Properties หรือระหว่างที่รันโปรแกรม โดยใช้สเตตเมนต์ (statement) ต่างๆ ในโค้ดโปรแกรม ในการเขียนโค้ดโปรแกรมจะใช้รูปแบบการกำหนดคุณสมบัติดังนี้

`Object.Property = Value`

โดยที่ object เป็นชื่อของออบเจกต์ที่ต้องการปรับแต่ง Property เป็นคุณลักษณะที่ต้องการเปลี่ยน และ Value เป็นค่าที่กำหนดขึ้นใหม่ ตัวอย่างเช่น `Command.Caption = "Hello"` สามารถใช้โค้ดโปรแกรมเพื่อกำหนดคุณสมบัติ Caption ของออบเจกต์ Command1 เป็น "Hello"

อีเวนต์โพรซีเจอร์ (Eventprocedure)

เป็นบล็อกของโค้ดที่ทำงานเมื่อออบเจกต์นั้นถูกใช้งานในโปรแกรม เช่น หากมีการคลิกที่ปุ่ม CommandButton ที่ชื่อ Command1 จะมีการเรียกโพรซีเจอร์ `Command1_Click` ทันที โปรแกรมสเตตเมนต์ (Statement)

โปรแกรมสเตตเมนต์เป็นคีย์เวิร์ดในโค้ดที่ใช้ทำงานของโปรแกรมสเตตเมนต์ของวิซวลเบสิก ใช้สร้างแหล่งเก็บข้อมูล เปิดไฟล์ คำนวณตัวเลข ชื่อหรือคุณสมบัติต่างๆ เก็บไว้ในโปรแกรมได้วิธีการ (Method)

เป็นสเตตเมนต์พิเศษที่ใช้ทำงานอย่างใดอย่างหนึ่งให้กับออบเจกต์เฉพาะอย่างในโปรแกรม ในการเขียนโค้ดโปรแกรมจะใช้รูปแบบการกำหนดวิธีการดังนี้

`Object.Method Value`

โดยที่ Object เป็นชื่อของออบเจกต์ที่ต้องการเปลี่ยน Method เป็นคำสั่งที่ต้องการใช้เปลี่ยนแปลงออบเจกต์นั้น และ Value เป็นอาร์กิวเมนต์เพิ่มเติมที่ใช้กับวิธีการ ตัวอย่างเช่น

`List1.AddItem "Check"`

ใช้วิธีการ AddItem เพิ่มคำว่า Check ไว้ใน List1

อุปกรณ์ที่ใช้ในการสร้างยูสเซอร์อินเตอร์เฟซ (User Interface)

-คอนโทรล Command Button

เป็นคอนโทรลที่ส่วนใหญ่นำไปใช้ในฟอร์มต่างๆ ฟอร์ม เนื่องจากเป็นคอนโทรลที่ใช้ในการตอบรับการทำงานจากผู้ใช้ เช่นเดียวกับปุ่มคำสั่ง OK, Cancel ทั่วไป

-คอนโทรล Text Box

มักจะถูกนำไปใช้ในทุกรูปแบบที่มีการรับกรอกข้อความจากผู้ใช้นี้ เนื่องจากคอนโทรลนี้ทำหน้าที่แสดงข้อมูล และยังสามารถแก้ไขตัวอักษรต่างๆ ได้อีกด้วย

-คอนโทรล Label

เป็นคอนโทรลในลักษณะของกราฟฟิกที่ถูกใช้ในงานด้านการแสดงผลข้อความบนฟอร์ม เพื่อใช้ในการสื่อสารข้อความกับผู้ใช้ และคอนโทรลนี้ไม่สามารถแก้ไขได้โดยตรง

-คอนโทรล Option Button

เป็นคอนโทรลที่ถูกออกแบบเพื่อนำมาใช้ในลักษณะของการเลือกอย่างใดอย่างหนึ่ง และคอนโทรลนี้ผู้อ่านสามารถนำมาใช้โดยแยกออกเป็นกลุ่มๆ ได้โดยอาศัยคอนโทรลชนิดตัวบรรจุบอกลักษณะเป็น true หรือ false

-คอนโทรล Check Box

เป็นคอนโทรลที่ถูกนำไปใช้ในลักษณะของการกำหนดสถานะเป็น True หรือ False คอนโทรลนี้ไม่มีลักษณะการรวมกลุ่ม ผู้ใช้สามารถเลือกที่เดียวพร้อมๆ กันได้คราวละหลายๆ ตัว

-คอนโทรล List Box

โดยที่ลักษณะการใช้งานจะเป็นในรูปแบบการแสดงผลข้อมูลหลายๆ บรรทัดภายในคอนโทรล List Box

-คอนโทรล Line และ Shape

มีเฉพาะคุณสมบัติและวิธีการสำหรับใช้ควบคุมการแสดงผลตำแหน่งและควบคุมพฤติกรรมของคอนโทรลเท่านั้น แต่จะไม่มีการตอบสนองเหตุการณ์กับผู้ใช้ เนื่องจากคอนโทรลนี้ไม่มีเหตุการณ์เป็นคอนโทรลเพิ่มกราฟฟิกให้กับฟอร์มเท่านั้น

-คอนโทรล Image

เป็นคอนโทรลที่ช่วยในการแสดงผลกราฟฟิก ในไฟล์กราฟฟิกต่างๆ เช่นเดียวกับคอนโทรล Picture Box แต่คอนโทรล Image จะสามารถทำงานหรือวาดกราฟฟิกภายในคอนโทรลได้รวดเร็วกว่า และที่สำคัญมีการใช้ทรัพยากรระบบน้อยกว่าอีกด้วย แต่มีการสนับสนุนคุณสมบัติเหตุการณ์ และวิธีน้อยกว่าคอนโทรล Picture Box

-คอนโทรล Picture Box

เป็นคอนโทรลแสดงกราฟฟิกและภาพกราฟฟิกจากไฟล์

-คอนโทรล Timer

เป็นคอนโทรลที่ใช้ในการควบคุม และจัดการเหตุการณ์ด้านเวลา สามารถเขียนโค้ดเมื่อทำงานใดๆ เมื่อ ช่วงเวลาผ่านไปตามค่าที่กำหนด คอนโทรลนี้จะตอบสนองเหตุการณ์เพียงเหตุการณ์เดียวเท่านั้น

-คอนโทรล Common Dialogue

เป็นคอนโทรลที่ช่วยในการแสดงผลไดอะล็อกบ็อกซ์ Open, Save As, Colour, Font, Print Setup หรือ Print ซึ่งสามารถจะควบคุมการแสดงผลส่วนต่างๆ ของๆ ไดอะล็อกบ็อกซ์ ได้โดยผ่านทางคุณสมบัติของคอนโทรล CommonDialogue ทั้งนี้เพื่อให้การใช้งานเป็นไปด้วยความยืดหยุ่นมากยิ่งขึ้น

-คอนโทรล Communications

เป็นเครื่องมือที่ช่วยในการติดต่อกับพอร์ตอนุกรม (serial port) ทำให้สามารถรับ-ส่งข้อมูลผ่านทางพอร์ตอนุกรมได้ด้วยคอนโทรลนี้ เช่นการติดต่อผ่านโมเด็ม

-ออบเจกต์ OLE

หมายถึง ข้อมูลที่ได้รับจากแอปพลิเคชัน OLE ซึ่งข้อมูลนี้สามารถเป็นไปได้หลายๆ ชนิด เช่น Microsoft Access สนับสนุนออบเจกต์ OLE ดังต่อไปนี้ worksheet, macro, sheet, chart, cell เป็นต้น และ ออบเจกต์ OLE ที่สร้างขึ้นใหม่ ผู้อ่านสามารถกำหนดให้เป็นออบเจกต์ OLE ชนิดเชื่อมโยง (Linked) หรือชนิดฝัง (Embedded) ได้

-ออบเจกต์ Printer

ใช้ในการควบคุมการพิมพ์ข้อความเอกสาร ฟอนต์หรือกราฟฟิก ออกทางเครื่องพิมพ์ โดยที่ข้อมูลทั้งหมดที่ถูกส่งพิมพ์โดยอาศัย ออบเจกต์นี้ จะถูกนำไปเข้าคิวรอจัดพิมพ์อีกครั้งโดย Print Manager ของวินโดวส์อีกครั้ง

คุณสมบัติ (Property)

Alignment	กำหนดตำแหน่งการวางข้อความในคอนโทรล คือ ค่า 0 วางข้อความชิดซ้าย ค่า 1 วางข้อความชิดขวา ค่า 2 วางข้อความไว้ตรงกลาง
BackColor	และ ForeColor รายงานหรือกำหนดสีพื้นและสีตัวอักษรของข้อความในคอนโทรล
Backstyle	รายงานหรือกำหนดคุณลักษณะของพื้นหลังของคอนโทรลให้เป็นแบบโปร่งใส (Transparent) หรือแบบทึบ (Opaque)
Caption	กำหนดข้อความสำหรับแสดงภายในคอนโทรล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Enabled	รายงานหรือกำหนดให้คอนโทรลสามารถมีหรือไม่มีการตอบสนองต่อเหตุการณ์ที่ถูกทำให้เกิดขึ้นโดยผู้ใช้ เช่น Click เป็นต้น โดยจะมีค่าเป็น True หรือ False
Fond	ใช้ในการกำหนดคุณสมบัติของออบเจกต์ฟอนต์ หรือรายงานคุณสมบัติของออบเจกต์ Font ของคอนโทรล
Height, Width, Left, Top	รายงานหรือกำหนดขนาดความกว้าง ความสูงและตำแหน่ง (Left,Top) การวางคอนโทรลลงในตัวบรรจุ
Name	กำหนดชื่อของคอนโทรลสำหรับใช้ในการอ้างอิงโดยโค้ดในการเขียนโปรแกรม
Visible	รายงานหรือกำหนดให้เกิดเหตุการณ์ Click ของคอนโทรลโดยการกำหนดให้คุณสมบัตินี้มีค่าเท่ากับ True
Value	รายงานหรือกำหนดให้เกิดเหตุการณ์ Click ของคอนโทรล โดยการกำหนดให้คุณสมบัตินี้มีค่าเท่ากับ True
List Index	รายงานหรือกำหนดค่าลำดับของรายการที่ถูกเลือก (สำหรับรายการแรกคุณสมบัติ List Index มีค่าเท่ากับ 0)
BorderStyle	รายงานหรือกำหนดการแสดงผลเส้นขอบคอนโทรลหรือกำหนดรูปแบบในการวาดเส้นตรงหรือเส้นรอบรูปในคอนโทรล Shape
BorderWidth	รายงานหรือกำหนดความหนาของเส้นตรงหรือเส้นรอบรูปทรง Shape
FillColor	รายงานหรือกำหนดสีที่ใช้ในการระบายพื้นที่ภายในรูปทรง Shape
FillStyle	รายงานหรือกำหนดรูปแบบของการระบายพื้นที่ภายในรูปทรง Shape
Shape	รายงานหรือกำหนดรูปทรงของคอนโทรล Shape เช่น สี่เหลี่ยม วงกลม
Picture	รายงานหรือกำหนดกราฟฟิกหรือไฟล์กราฟฟิกจะแสดงผลภายในคอนโทรล
Text	รายงานหรือกำหนดข้อความให้กับคอนโทรล ซึ่งข้อความที่อยู่ในคุณสมบัตินี้จะถูกนำไปแสดงในคอนโทรล Text box
Stretch	รายงานหรือข้อกำหนดให้กราฟฟิกมีขนาดเปลี่ยนแปลงตามขนาดของคอนโทรล
Interval	รายงานหรือกำหนดช่วงเวลาในหน่วยของมิลลิวินาที (1 ใน 1000 วินาที) สำหรับทำให้เกิดเหตุการณ์ Timer ซึ่งการตรวจสอบเวลาจะถูกกระทำโดย Vb/Win

2.3.2 Microsoft Access

Microsoft Access เป็นโปรแกรมจัดการฐานข้อมูล (Database Management Program) ที่มีขีดความสามารถสูง และมีการออกแบบให้ง่ายต่อการใช้ แต่ก็ยังคงมีปัญหาเกิดขึ้นบ้างในการใช้งานเนื่องจากคุณสมบัติต่างๆ ที่มีอยู่มากมายในโปรแกรม

นิยามต่างๆ ที่ควรรู้

Database	คือไฟล์เดี่ยวซึ่งประกอบด้วยข้อมูลและ tools ทั้งหมด (เช่น form และ report) สำหรับใช้ในการดำเนินการกับข้อมูล
Tables	คือสิ่งที่ต้องใช้ในการจัดเก็บข้อมูลดิบ ซึ่งมีการจัดเรียงเช่นเดียวกับ Spreadsheet แต่ละคอลัมน์หรือฟิลด์จะประกอบด้วยรายละเอียดแต่ละชนิด เช่น ZIP Code ในแต่ละแถว (row) หรือ (Records) จะประกอบไปด้วยรายละเอียดในแต่ละรายการหรือเหตุการณ์ เช่น ลูกค้า คนไข้ หรือ ข้อมูลต่างๆ ด้านเครื่องมือ เป็นต้น
Queries	จะเป็นส่วนที่เรียกใช้และรวบรวมข้อมูลจาก Tables ต่างๆ โดยขึ้นกับหลักเกณฑ์ที่กำหนดไว้ ตัวอย่างเช่น สามารถใช้ Query ในการเรียกใช้ record จาก table ชื่อ employee สำหรับพนักงานทุกคนซึ่งทำงานล่วงเวลาในสัปดาห์ที่ผ่านมา Queries ยังสามารถเรียกใช้ข้อมูลเป็นกลุ่มได้เช่นกัน
Forms	เป็น window ที่ปรากฏขึ้นบน screen และใช้ในการแสดงข้อมูลหรือบันทึกข้อมูลโดยตรง
Reports	จะทำงานคล้ายกับ Forms เว้นแต่ว่า Report ใช้สำหรับพิมพ์ออกมา ไม่แสดงบน screen และไม่ได้ใช้ในการบันทึกข้อมูล
Macro	เป็นส่วนที่ดำเนินการตามที่กำหนดโดยอัตโนมัติ และใช้เพื่อทำให้การทำงานใน Access ง่ายขึ้น
Modules	คือ sheet ต่างๆ ซึ่งประกอบด้วย Access Basic code ที่มีการโปรแกรมขึ้นใน Access

2.3.3 การจัดการฐานข้อมูล

ในการจัดการฐานข้อมูลต่างๆ ทำในลักษณะของการจัดเก็บข้อมูลแล้วนำข้อมูลนั้นออกมาใช้หรือแสดงในภายหลังในรูปแบบของรายงานต่างๆ เช่น งานทางด้านบัญชี จัดการกับสินค้าคงคลัง ข้อมูลบุคลากร เป็นต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ความซับซ้อนของกระบวนการต่างๆ ในโปรแกรมส่วนหนึ่งเป็นผลจากความสลับซับซ้อนของข้อมูลที่เกี่ยวข้อง อีกทั้งรูปแบบของการเก็บข้อมูลในแฟ้มข้อมูลในลักษณะพื้นฐาน เช่น แฟ้มข้อมูลแบบ Sequential, Random หรือ Indexed File ก็ล้วนมีจุดอ่อนในการลดทอนประสิทธิภาพการทำงานของโปรแกรมหลายประการ เช่น

ผู้พัฒนาโปรแกรมจะต้องมีความเข้าใจในลักษณะของแฟ้มข้อมูลที่จะใช้นั้นเป็นอย่างดีและจะต้องคิดหาวิธีการที่จะเรียกใช้ข้อมูลดังกล่าวให้เหมาะสมกับลักษณะการจัดเก็บในแฟ้มข้อมูลแบบนั้นๆ โดยเฉพาะทำให้โปรแกรมที่ต้องการจะพัฒนานั้นซับซ้อนขึ้นโดยไม่จำเป็น

ขัดแย้งกับแนวความคิดในการพัฒนาโปรแกรมในลักษณะของ Modular Programming เพราะคำสั่งต่างๆ ที่ใช้ในการจัดการกับฐานข้อมูลนั้นจะขึ้นอยู่กับภาษาหรือเครื่องมือในการพัฒนานั้น เช่น คำสั่งที่ใช้ในการอ่านข้อมูลภาษาซี หรือ เบสิก ก็ไม่เหมือนกัน ไม่สามารถจะใช้สิ่งที่ได้เรียนรู้อะไรและโปรแกรมย่อยต่างๆ ร่วมกันได้ ทำให้สิ้นเปลืองทั้งเวลาและเงินทองในการเรียนรู้และพัฒนา

ตัวข้อมูลเองก็อาจจะไม่สามารถที่จะใช้ร่วมกัน (share) กับโปรแกรมอื่นได้อย่างสะดวก หากในหน่วยงานหรือองค์กรมีระบบงานที่ใช้ข้อมูลเหมือนกัน แต่ไม่สามารถใช้ข้อมูลร่วมกันได้จะทำให้เกิดความซับซ้อนของข้อมูลขึ้นในหน่วยงานหรือองค์กรนั้น ซึ่งผลต่อมาจะทำให้ไม่สามารถบอกได้ว่าข้อมูลใดถูกต้องและใหม่ที่สุด

เราสามารถจะหลีกเลี่ยงปัญหาเหล่านี้ได้โดยการใช้ระบบจัดการฐานข้อมูล (DataBase Management System – DBMS) แทนการใช้แฟ้มข้อมูลทั่วไป ลักษณะของฐานข้อมูลนั้นมีหลักๆ อยู่ 3 ประเภท คือแบบ Hierarchy แบบ Network และแบบ Relational แต่ที่ได้รับความนิยมอย่างมากในปัจจุบันจะเป็นฐานข้อมูลแบบสัมพันธ์ (Relation Database) เพราะว่าพื้นฐานข้อมูลที่จะจัดเก็บจะอยู่ในรูปของตาราง (Table) ซึ่งทำให้ง่ายต่อการเข้าใจและเรียกใช้งาน

ใน MS-Visual Basic 6.0 สามารถติดต่อกับฐานข้อมูลได้หลากหลาย เช่น dBASE, FoxPro และยังสามารถสร้าง ODBC (Open DataBase Connectivity) เพื่อเป็นมาตรฐานในการติดต่อกับฐานข้อมูลภายนอกบนเครื่องเซิร์ฟเวอร์ (Server) ต่างๆ เช่น SQL Server, Oracle ซึ่ง MS-Visual Basic จะเตรียมโปรแกรมสำหรับติดต่อกับฐานข้อมูลชนิดนั้นๆ หรือที่เรียกว่าไดรเวอร์ (Driver) ไว้ให้ หากต้องการใช้กับฐานข้อมูลชนิดอื่น นอกจากนั้นต้องหาไดรเวอร์สำหรับฐานข้อมูลชนิดนั้นมาติดตั้งเพิ่มเติม

ข้อดีของความสามารถทางด้านฐานข้อมูลของ MS-Visual Basic คือ ช่วยลดข้อจำกัดการพัฒนางาน โดยใช้แฟ้มข้อมูลธรรมดาในการจัดเก็บข้อมูล เช่น

-โปรแกรมจะเป็นอิสระจากฐานข้อมูลที่ใช้เพราะเมื่อไม่มีความแตกต่างของฐานข้อมูลหมายความว่าจะเป็นฐานข้อมูลชนิดใด ก็จะใช้ชุดคำสั่งในการอ่านเขียน หรือจัดการต่างๆ ได้เหมือนกัน จะแตกต่างก็ตรงส่วนขยายของคำสั่งสำหรับการเปิด (open) หรือสร้าง (create) ฐานข้อมูลเท่านั้น ทำให้ไม่ต้องเรียนรู้ถึงคำสั่งที่แตกต่างกันหลายแบบ นอกจากนี้การใช้ชุดคำสั่งย่อย (Module) ร่วมกันก็เป็นไปได้มาก และสร้างระบบงานในลักษณะของ Client/Server ก็ง่ายขึ้น สามารถทำการพัฒนาบนระบบเครื่องคอมพิวเตอร์เครื่องเดียวก็ได้ จากนั้นก็ค่อยเปลี่ยนค่าคงที่ที่กำหนดในเรื่องฐานข้อมูลเท่านั้น ก็สามารถนำไปใช้กับข้อมูลบนเครื่องอื่นๆ ได้ทันที

-ข้อมูลเดิมที่มีอยู่แล้วในฐานข้อมูลชนิดอื่นๆ ก็สามารถนำมาใช้ใน MS-Visual Basic ได้โดยไม่ต้องสร้างใหม่อีกชุดให้ซับซ้อน หรือต้องทำการแปลง (convert) ให้เป็นฐานข้อมูล Access เสียก่อน เพราะการทำเช่นนั้นจะเสียเวลาไปพัฒนาระบบงานเดิมที่ใช้กับฐานข้อมูลนั้นอยู่แล้วซ้ำอีกชุดหนึ่งสำหรับฐานข้อมูลที่เปลี่ยนแปลงมาใหม่ซึ่งจะเป็นการสิ้นเปลืองเวลาและเงิน

โครงสร้างฐานข้อมูล

ฐานข้อมูลแบบสัมพัทธ์มีหลักการพื้นฐานมาจากคณิตศาสตร์ในเรื่องทฤษฎีบทของเซต (Theory of Set) โดยที่การจัดเก็บหรือแสดงข้อมูลให้ผู้ใช้เห็นจะเป็นในรูปแบบของตาราง

ข้อมูลต่างๆ จะแสดงในรูปของตาราง โดยในแต่ละตารางจะเป็นการจัดรวบรวมข้อมูลประเภทเดียวกันไว้ด้วยกัน โดยที่แต่ละแถวที่ประกอบขึ้นเป็นตารางนั้น ก็คือ เรคคอร์ด ซึ่งจะเป็นที่เก็บข้อมูลแต่ละชุดของตารางนั้น และในแต่ละแถวก็จะประกอบด้วยฟิลด์หรือคอลัมน์ซึ่งเป็นหน่วยย่อยที่แสดงคุณสมบัติของข้อมูลในแต่ละแถว

ในแต่ละตารางมักจะมีบางคอลัมน์หรืออาจจะหลายคอลัมน์ประกอบกันที่สามารถบอกถึงความแตกต่างกันของฐานข้อมูลในแต่ละแถวได้ คอลัมน์ดังกล่าวจะเรียกว่า Primary Key เช่น คอลัมน์เลขประจำตัวพนักงานที่จะแตกต่างกันทุกคน และในการเชื่อมโยงตารางฐานข้อมูลที่มีหลายตารางนั้นจะทำได้ก็ต่อเมื่อแต่ละตารางมีคอลัมน์ที่เกี่ยวข้องกัน ซึ่งคอลัมน์ที่เกี่ยวข้องกันนี้จะเรียกว่า Foreign Key

ชุดคำสั่งที่ใช้ในการจัดการกับข้อมูลและโครงสร้างของฐานข้อมูล

แบ่งได้เป็น 2 ประเภทหลักๆ คือ

1. จัดการกับโครงสร้างฐานข้อมูล เป็นการสร้าง ลบ หรือแก้ไขฐานข้อมูลและตาราง เช่น สร้างตารางฐานข้อมูลพนักงานขึ้นมา ชุดคำสั่งต่างๆ ที่เกี่ยวข้องกับการสร้างหรือเปลี่ยนแปลงโครงสร้างของฐานข้อมูลมีศัพท์เรียกว่า Data Definition Language หรือ DDL
2. จัดการกับข้อมูลที่อยู่ในตารางต่างๆ ในฐานข้อมูล เป็นการทำงานในเรื่องของการเพิ่ม ลบ แก้ไขข้อมูลในตารางต่างๆ เช่น การเพิ่มข้อมูลพนักงานใหม่เข้าไปในตารางข้อมูลพนักงาน คำสั่งประเภทนี้จะเรียกว่า เป็นคำสั่ง Data Management Language หรือ DML

ปัจจุบันชุดคำสั่งฐานข้อมูลที่นิยมกันมาก เรียกว่า Structure Query Language (SQL) โดยจะประกอบไปด้วยคำสั่งสำหรับการจัดการกับฐานข้อมูล ซึ่งจะมีมาตรฐานกำหนดไว้ทั้งที่เป็นคำสั่งสำหรับการจัดการกับโครงสร้าง (DDL) หรือจัดการกับข้อมูลในฐานข้อมูล (DML) ชุดคำสั่ง SQL ของผู้ผลิตแต่ละรายเกือบจะใกล้เคียงหรือเหมือนกัน แต่อาจจะแตกต่างกันบ้างเล็กน้อย เพราะผู้ผลิตในแต่ละรายมักจะมีการเพิ่มเติมคำสั่งหรือตัวลึกลงให้มากขึ้นกว่าที่กำหนดไว้ในมาตรฐาน เพื่อให้มีความสะดวกหรือมีประสิทธิภาพดีเหมาะกับฐานข้อมูลของตน

การจัดการฐานข้อมูลโดยใช้ Structure Query Language (SQL)

Structure Query Language (SQL) เป็นภาษามาตรฐานที่ใช้ในการจัดการฐานข้อมูลที่เป็นแบบ RDBMS (Relational Database Management System) SQL เป็นภาษาที่มีประสิทธิภาพสูง สามารถเรียนรู้และใช้งานได้อย่างง่ายดาย จึงทำให้เป็นที่นิยมได้อย่างรวดเร็ว นอกจากนี้ยังสามารถใช้ในการจัดการกับฐานข้อมูลโดยผ่าน RDBMS เอง หรือจะตั้งผ่านโปรแกรมก็ได้ เช่น หากเราใช้ MS-Access เราก็สามารถพิมพ์คำสั่ง SQL ได้โดยตรงในโปรแกรม Access หรือจะใช้คำสั่ง SQL ผ่าน VB6 หรือ Compiler อื่นๆ ก็ได้

คำสั่งที่ใช้ใน SQL มีรูปแบบดังต่อไปนี้

SELECT ชื่อฟิลด์ FROM ชื่อตาราง [WHERE เงื่อนไข ORDER BY ชื่อฟิลด์ที่ต้องการจัดเรียง]

ตัวอย่างการใช้งานเช่น

```
SELECT * FROM Product WHERE ProductID >= 10 ORDER BY ProductName
```

คือต้องการเรียกฟิลด์ทุกฟิลด์ (หากกำหนดเป็น *) จากตาราง Product โดยที่ค่า ProductID มีค่ามากกว่าหรือเท่ากับ 10 โดยมีการจัดเรียงค่า ProductName ตามลำดับตัวอักษร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

วิธีการติดต่อกับฐานข้อมูลของ MS-Visual Basic 6.0

โปรแกรม MS-Visual Basic 6.0 คือมีความสามารถในการจัดการกับฐานข้อมูลได้ โดยจะมีวิธีการจัดการอยู่ 2 แบบ คือ

1. การใช้ค่าตัวคอนโทรล (Data Control) เป็นวิธีการที่ง่ายและสะดวกที่สุดในการติดต่อกับฐานข้อมูล เนื่องจากค่าตัวคอนโทรลจะติดต่อกับฐานข้อมูลและจัดการกับข้อมูลในตารางอัตโนมัติ เช่น การเปิดฐานข้อมูล การแสดงและแก้ไขข้อมูลในตาราง เป็นต้น อย่างไรก็ตามการใช้ค่าตัวคอนโทรลยังมีข้อจำกัดอยู่พอสมควร เช่น ไม่มีฟังก์ชันในการลบข้อมูล เป็นต้น

2. การใช้ค่าตัวออบเจกต์ (Data Object) วิธีนี้จะต้องเขียนโปรแกรมเองทั้งหมดเพื่อติดต่อกับฐานข้อมูล โดยการใช้ออบเจกต์ต่างๆ ที่ VB6 มีมาให้ โดยที่การเพิ่ม ลบ หรือแก้ไขข้อมูล จะต้องเขียนโปรแกรมจัดการเอง แต่ข้อดีของวิธีนี้คือสามารถติดต่อกับข้อมูลจากหลายๆ ตารางพร้อมกัน สามารถสร้างคิวรีตอนรันโปรแกรมได้ และสามารถควบคุมความผิดพลาดต่างๆ ได้ดีกว่าใช้ค่าตัวคอนโทรล รวมทั้งสามารถใช้ภาษา SQL เพื่อจัดการกับฐานข้อมูลได้อีกด้วย

การจัดการเกี่ยวกับฐานข้อมูล

โดยปกติวิธีการสร้างฐานข้อมูลเพื่อจะนำมาใช้งานในวิซวลเบสิกนั้น ทำได้ 4 วิธีหลักๆ คือ

1. สร้างโดยอาศัยเครื่องมือของระบบจัดการฐานข้อมูลนั้น

วิธีนี้จะแตกต่างกันไปตามแต่ละรูปแบบของฐานข้อมูลที่ใช้ เช่น หากต้องการใช้งานฐานข้อมูลของ dBASE ก็ใช้เครื่องมือของ dBASE เป็นตัวสร้าง

2. ใช้โปรแกรม Data Manager

โปรแกรม Data Manager จะมีมากับวิซวลเบสิกแล้ว โปรแกรมนี้จะใช้สร้างฐานข้อมูลในรูปแบบของ Jet 1.1 หรือ Access ได้เพียงอย่างเดียวเท่านั้น โดยที่เราไม่จำเป็นต้องมีโปรแกรม MS-Access ตัวโปรแกรม Data Manager นั้นอาจจะมีขีดความสามารถจำกัด ทำได้เพียงการสร้าง ลบ และแสดงรายละเอียดของตารางและดัชนีต่างๆ ในฐานข้อมูลเท่านั้น ไม่สามารถทำการแก้ไขหรือเปลี่ยนแปลงรายละเอียดของตารางเดิมที่มีอยู่แล้วได้

3. ใช้ Data Control

Data Control ก็นับเป็นคอนโทรลประเภทหนึ่งที่สามารถจัดวางลงในฟอร์มของวิซวลเบสิกได้และใช้ในการจัดเก็บข้อมูลต่างๆ ในฐานข้อมูลได้โดยไม่ต้องเขียนโปรแกรม วิธีการนี้ถึงแม้สะดวกแต่ก็ยังข้อจำกัดอยู่บ้าง

4. ใช้ชุดคำสั่งในการจัดการฐานข้อมูล (DDL) กับออบเจกต์ฐานข้อมูลต่างๆ

วิธีการนี้เหมาะสำหรับเมื่อเราต้องการจะสร้างฐานข้อมูลขึ้นมาใหม่ในระหว่างการทำงานของโปรแกรม จัดเป็นวิธีที่ยุงยากมากที่สุด แต่ก็ให้ความยืดหยุ่นดี เพราะสามารถจะสร้าง ลบ หรือแก้ไขรายละเอียดต่างๆ ได้ในระหว่างการทำงานและยังใช้ในการจัดการกับออบเจกต์แบบที่เป็น QueryDef ได้ด้วยเช่นเดียวกัน (เราไม่สามารถจะสร้าง QueryDef โดยใช้โปรแกรม Data anager ได้)

การใช้งานดาต้าคอนโทรล (Data Control)

ดาต้าคอนโทรลเป็นเครื่องมือที่ช่วยให้สามารถจัดการกับข้อมูลในตารางได้ เช่น การเพิ่ม ลบ แก้ไข หรือการค้นหาข้อมูลที่ต้องการได้ เป็นต้น



รูปที่ 2.22 Data Control

คุณสมบัติที่สำคัญของดาต้าคอนโทรล

- DataSource เป็นการกำหนดชื่อของดาต้าคอนโทรลที่ต้องการเชื่อมโยง
- DataField เป็นการกำหนดว่าเราต้องการให้แสดงหรือแก้ไขข้อมูลฟิลด์ใดในตาราง
- DatabaseName เป็นที่อยู่ของไฟล์ฐานข้อมูลที่ต้องการ โดยมีการระบุชื่อไฟล์นั้นๆ
- Connect เป็นคุณสมบัติที่ใช้บอกว่าต้องการติดต่อกับฐานข้อมูลประเภทใด เช่น MS-Access
- RecordSetType เป็นชนิดของเรคอร์ดเซตที่ต้องการเข้าถึงในฐานข้อมูลแบ่งได้เป็น 3 ชนิด คือ
1. Table สามารถใช้ติดต่อกับตารางใดตารางหนึ่งในฐานข้อมูลเท่านั้น ซึ่งการเข้าถึงข้อมูลสามารถเข้าได้อย่างรวดเร็ว เนื่องจากจะใช้อินเด็กซ์เป็นตัวค้นหาข้อมูล ส่วนข้อจำกัดคือ ไม่สามารถดึงข้อมูลบางส่วนจากตารางได้ จะต้องดึงมาทั้งตารางเท่านั้น
 2. Dynaset สามารถใช้ติดต่อกับข้อมูลในหลายๆ ตารางได้ เราสามารถใช้คำสั่ง SQL เพื่อดึงข้อมูลบางส่วนจากตาราง รวมทั้งสามารถแก้ไขค่าของข้อมูลได้ แต่การทำงานช้ากว่าการใช้ Table
 3. Snapshot จะเหมือนกับ Dynaset แต่การทำงานจะเร็วกว่า เนื่องจากจะเก็บข้อมูลอยู่ในหน่วยความจำ และไม่สามารถแก้ไขข้อมูลได้
- RecordSource เป็นการกำหนดชื่อของ Table Dynaset หรือ Snapshot ที่เราต้องการติดต่อ

ออบเจกต์เรคอร์ดเซต Recordset และการจัดเก็บข้อมูล

คุณสมบัติ (Property) ของเรคอร์ดเซต

Connect	โดยปกติจะว่าง แต่ถ้าหากมีค่าจะหมายถึงค่าของ Attached Table ในกรณีที่เป็นตาราง
Datacreated	วันที่ที่ตารางที่เกี่ยวข้องนี้ถูกสร้างขึ้นมา
LastUpdated	วันที่ที่มีการแก้ไขครั้งสุดท้าย
Name	แสดงชื่อของตาราง
SourceTableName	ชื่อของตารางที่อ้างอิงถึงบนฐานข้อมูลอื่น ในกรณีที่เป็นตารางซึ่งกำหนดบนตารางฐานข้อมูลอื่น
Updatable	หากว่ามีค่าเป็นจริง ก็หมายความว่ายอมให้มีการแก้ไขเปลี่ยนแปลงข้อมูลได้
EOF	เป็นคุณสมบัติที่แสดงว่าได้มีการเลื่อนตำแหน่งเรคอร์ดไปข้างหน้าผ่านเรคอร์ดสุดท้ายในไฟล์หรือไม่ โดยจะมีค่าเป็นจริงหรือเท็จ
BOF	เป็นคุณสมบัติที่แสดงว่าได้มีการเลื่อนตำแหน่งเรคอร์ดย้อนหลังผ่านเรคอร์ดแรกในไฟล์หรือไม่ โดยมีค่าเป็นจริงหรือเท็จ

ฟังก์ชันการทำงาน (Method) ของเรคอร์ดเซต

MoveFirst	เป็นการตั้งเลื่อนตัวชี้ตำแหน่งในออบเจกต์เรคอร์ดเซตนั้นให้ไปชี้ตำแหน่งเรคอร์ดแรกสุด
MoveNext	เพื่อเลื่อนตัวชี้ตำแหน่งไปยังเรคอร์ดถัดไปจากตำแหน่งปัจจุบัน
MovePrevious	เพื่อเลื่อนตัวชี้ตำแหน่งไปยังเรคอร์ดก่อนหน้าตำแหน่งปัจจุบัน
MoveLast	เลื่อนไปยังตำแหน่งเรคอร์ดสุดท้ายในเรคอร์ดเซต
FindFirst	เลื่อนไปยังตำแหน่งเรคอร์ดที่มีข้อมูลในลักษณะตามที่ต้องการ โดยเริ่มต้นหาจากเรคอร์ดแรกสุดไปเรคอร์ดสุดท้าย
FindNext	เลื่อนไปยังตำแหน่งของเรคอร์ดถัดไปจากตำแหน่งปัจจุบัน ที่มีข้อมูลตามลักษณะที่ต้องการ
FindPrevious	เหมือนกับฟังก์ชัน FindNext แต่เป็นการค้นหาในทิศทางย้อนกลับ
FindLast	เหมือนกับฟังก์ชัน FindFirst แต่ในทิศทางตรงกันข้าม คือเริ่มจากตำแหน่งเรคอร์ดสุดท้ายก่อนแล้วจึงย้อนกลับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Nomatch

หากพบเรคอร์ดที่มีข้อมูลในลักษณะที่กำหนด ค่าของคุณสมบัตินี้จะเป็น False แต่ถ้าหากค้นหาไม่พบ จะมีค่าเป็น True และตำแหน่งของตัวชี้จะอยู่ที่เดิม

2.4 การติดต่อสื่อสาร

พอร์ตอนุกรม (Serial Port)

พอร์ตอนุกรม RS-232-C จะเป็นพอร์ตของเครื่องคอมพิวเตอร์ที่มีขาต่อ(connector) ทั้งประเภท 9 และ 25 ขา และเราเรียกกันว่า COM1: และ COM2: นั่นเอง ในความเป็นจริงพอร์ตอนุกรมไม่ได้ถูกควบคุมโดยตรงจาก CPU บนเมนบอร์ด แต่การสื่อสารทั้งหมดจะถูกจัดการโดยชิป UART (Universal Asynchronous Receiver/Transmitter) อีกทีหนึ่ง ซึ่งในปัจจุบันบอร์ดที่ใช้กันมากที่สุดคือเบอร์ 16550C ซึ่งเป็นเวอร์ชันที่ได้รับการแก้ไขข้อผิดพลาดแล้ว ชิป UART นี้จะทำหน้าที่ในการรับและการส่งข้อมูลดังต่อไปนี้

การส่งข้อมูล (Data Transmission)

- รับตัวอักษรจากเครื่องคอมพิวเตอร์
- แปลงตัวอักษรให้เป็นสายข้อมูลแบบบิต (เราเรียกว่ากระบวนการ serialization)
- สร้างเฟรมข้อมูลโดยการเพิ่มบิตที่จำเป็นสำหรับการสื่อสารและการตรวจสอบ เช่น บิต START, STOP และ PARITY เป็นต้น
- ส่งผ่านเฟรมข้อมูลที่สร้างขึ้นมาแล้วจากขั้นตอนที่ผ่านมา ด้วยความเร็วของโมเด็มหรือพอร์ตอนุกรม (baud rate)
- แสดงสถานะความพร้อมที่จะรับข้อมูลตัวอักษรถัดไปให้กับเครื่องคอมพิวเตอร์

การรับข้อมูล (Data Receiver)

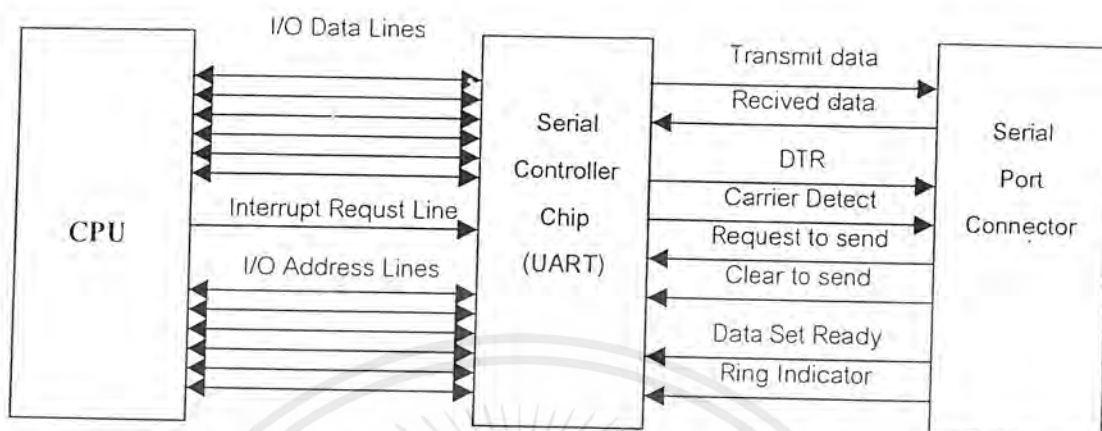
- รับตัวอักษรจากอินเตอร์เฟซ
- ตรวจสอบความถูกต้องของเฟรมข้อมูลตามมาตรฐานที่กำหนด โดยถ้าหากเฟรมข้อมูลมีรูปแบบที่ไม่ถูกต้องก็จะมีการแจ้งข้อผิดพลาดทันที
- ตรวจสอบความถูกต้องของพาริตี
- แปลงสายข้อมูลแบบบิตเป็นตัวอักษร
- ส่งตัวอักษรให้กับเครื่องคอมพิวเตอร์
- แสดงสถานะความพร้อมที่จะรับข้อมูลตัวอักษรถัดไปให้กับอินเตอร์เฟซ

สำหรับการเชื่อมต่อสายสัญญาณต่างๆ ระหว่าง CPU ของเมนบอร์ดของเครื่องคอมพิวเตอร์กับ

พอร์ตอนุกรมนั้น จะต้องกระทำผ่านทางชิป UART ซึ่งมีวิธีการเชื่อมต่อดังในรูปข้างล่างนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่อผู้จัดทำเอกสารนี้ได้รับประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.23 แสดงการเชื่อมต่อสายสัญญาณระหว่าง CPU ของเครื่องคอมพิวเตอร์กับพอร์ตอนุกรม

Transmit Data (TD)

เป็นวงจรที่สร้างสัญญาณ transmitted data ซึ่งถูกส่งจาก DTE ไปยัง DCE ซึ่งสัญญาณที่ส่งออกมาจะเป็นไคด์คำสั่งโมเด็มหรือข้อมูลก็ได้

Received Data (RD)

เป็นวงจรที่สร้างสัญญาณ received data ซึ่งถูกส่งจาก DCE ไปยัง DTE ซึ่งสัญญาณที่ส่งออกมาจะเป็นไคด์คำสั่งของโมเด็มหรือข้อมูลก็ได้ ซึ่งเป็นสัญญาณที่มีทิศทางตรงข้ามกับสัญญาณ transmitted data

Data Terminal Ready (DTR)

สัญญาณ DTR จะถูกส่งจาก DTE ไปยัง DCE เพื่อเป็นการแจ้งความพร้อมในการสื่อสารให้โมเด็มได้ทราบ โดยหากโมเด็ม (อุปกรณ์ DCE) มีความสามารถในการตอบรับแบบอัตโนมัติ (automatically answer) โมเด็มก็จะสามารถตอบรับได้เฉพาะเมื่อมีสัญญาณ DTR อยู่สถานะ on เท่านั้น

Carrier Detect (CD)

สัญญาณ CD จะถูกส่งจาก DCE ไปยัง DTE เพื่อเป็นการแจ้งว่าโมเด็มอยู่ในสถานะกำลังติดต่อกับโมเด็มตัวอื่น หรือโมเด็มกำลังได้รับสัญญาณที่พร้อมสำหรับการติดต่อสื่อสาร สำหรับสัญญาณ Carrier Detect นี้ สามารถเรียกอีกชื่อได้ว่า Received Line Signal Detector

Request To Send (RTS)

สัญญาณ RTS จะถูกส่งออกจาก DTE ไปยัง DCE โดยเมื่อสัญญาณ RTS อยู่ในสถานะ on ก็หมายถึง เครื่องคอมพิวเตอร์พร้อมที่จะรับข้อมูลจากโมเด็ม และในทางกลับกันหากสัญญาณ RTS อยู่ในสถานะ off ก็หมายถึง เครื่องคอมพิวเตอร์ไม่พร้อมที่จะรับข้อมูลจากโมเด็ม

Clear To Send (CTS)

สัญญาณ CTS จะถูกส่งจาก DCE ไปยัง DTE ซึ่งเป็นสัญญาณที่ทำหน้าที่ต้องห้ามกับสัญญาณ RTS โดยเมื่อสัญญาณ CTS อยู่ในสถานะ on ก็หมายถึง โมเด็มพร้อมที่จะรับข้อมูลจากเครื่องคอมพิวเตอร์ และในทางกลับกันหากสัญญาณ CTS อยู่ในสถานะ off ก็หมายถึงโมเด็มไม่พร้อมที่จะรับข้อมูลจากเครื่องคอมพิวเตอร์

Data Set Ready (DSR)

สัญญาณ DSR จะถูกส่งจาก DCE ไปยัง DTE เพื่อเป็นการแจ้งความพร้อมในการสื่อสารจากโมเด็มให้กับเครื่องคอมพิวเตอร์ได้ทราบ โดยสัญญาณ DSR จะอยู่ในสถานะ on ก็ต่อเมื่อโมเด็มได้รับสัญญาณ DTR เท่านั้น

Ring Indicator (RI)

สัญญาณ RI จะถูกส่งจาก DCE ไปยัง DTE เพื่อเป็นการแจ้งให้เครื่องคอมพิวเตอร์ทราบว่าโมเด็มกำลังได้รับสัญญาณกระดิ่ง (ring signal) จากโมเด็มตัวอื่น โดยที่สัญญาณ RI จะอยู่ในสถานะ on โมเด็มได้รับสัญญาณกระดิ่ง และ off เมื่อโมเด็มไม่ได้รับสัญญาณกระดิ่ง เนื่องจากโมเด็มรุ่นใหม่ ๆ ในปัจจุบันมักจะสามารถในการตอบรับแบบอัตโนมัติ (automatic answer) ดังนั้นจึงไม่มีความจำเป็นต้องใช้สัญญาณ RI อีกต่อไป

รูปแบบของเฟรมข้อมูล (Data Format)

เฟรมข้อมูลในการส่งหรือรับจะประกอบด้วยบิตข้อมูลที่มีความหมายดังต่อไปนี้

- บิตข้อมูล (Data Bit)

เมื่อชิป UART ได้รับตัวอักษรที่ส่งมาจากเครื่องคอมพิวเตอร์แล้ว ก็ต้องทำการแปลงตัวอักษรดังกล่าวให้เป็นสายข้อมูลชนิดบิตที่มีความยาวตั้งแต่ 5 ถึง 8 บิต ซึ่งเราเรียกขบวนการแปลงตัวอักษรให้เป็นสายข้อมูลบิตนี้ว่า Serialization จากนั้นโมเด็มก็จะทำการส่งแต่ละบิตไปยังโมเด็มปลายทาง โดยจะเริ่มส่งจากบิตที่มีนัยสำคัญต่ำสุด (least significant bit) ไปยังบิตที่มีนัยสำคัญสูงสุด (most significant bit)

- **บิตเริ่มต้นข้อมูล (START Bit)**

ในการส่งข้อมูลแบบ asynchronous นั้น เราจะต้องมีวิธีการบอกโมเด็มให้ทราบถึงจุดเริ่มต้นของข้อมูลที่ต้องการส่ง ดังนั้นก่อนหน้าข้อมูลทุกๆ เฟรมจะต้องถูกนำหน้าด้วยบิตเริ่มต้นข้อมูล (start bit) เสมอ

- **บิตสิ้นสุดข้อมูล (STOP Bit)**

ในการส่งข้อมูลแบบ asynchronous นั้น ในกรณีที่บิตเริ่มต้นข้อมูลเกิดการสูญหายในระหว่างการส่ง โมเด็มก็จะไม่สามารถทราบถึงจุดสิ้นสุดของสายข้อมูลบิตได้เลย นอกจากโมเด็มจะตรวจบิตเริ่มต้นใหม่อีกครั้งเท่านั้น ดังนั้นจึงมีการเพิ่มบิตสิ้นสุดข้อมูล ต่อท้ายทุกๆ ข้อมูลของแต่ละตัวอักษรเพื่อแจ้งการสิ้นสุดของสายข้อมูลบิต โดยการที่เรียงกันของบิตเริ่มต้นข้อมูล สายข้อมูล และบิตสิ้นสุดข้อมูลนี้เราเรียกว่า เฟรมข้อมูล (data frame)

- **บิตพาริตี (Parity Bit)**

เนื่องจากการส่งผ่านข้อมูลทางสายสัญญาณนั้น สามารถเกิดสัญญาณรบกวนได้ง่าย ด้วยเหตุนี้จึงเป็นไปได้มากที่สถานะของแต่ละบิตของข้อมูลที่ถูกส่งจะมีการเปลี่ยนแปลง เช่น จากบิต 0 เป็นบิต 1 เป็นต้น ในระหว่างการส่งผ่านข้อมูล ด้วยเหตุนี้จึงมีการคิดค้นวิธีที่ง่ายที่สุดที่ช่วยในการตรวจสอบความถูกต้องของสถานะของบิตข้อมูล ซึ่งเรียกกันว่า การตรวจสอบค่าพาริตี (Parity Check) โดยในการส่งผ่านข้อมูลด้วยโปรโตคอล START/STOP นั้น ในการส่งตัวอักษรข้อความทั่วไปจะใช้เพียง 7 บิตข้อมูลเท่านั้น ดังนั้นจึงมีการเพิ่มบิตพาริตีต่อสายเฟรมข้อมูลที่ถูกสร้างขึ้นมา เพื่อตรวจสอบสถานะของผลบวกของบิตที่เป็น 1 ของสายบิตข้อมูลของสายแต่ละตัวอักษร สำหรับการคำนวณค่าของพาริตีมีดังนี้

1. ถ้าหากจำนวนของบิตที่มีค่าเท่ากับ 1 ของสายบิตข้อมูลมีค่าเป็นเลขคู่ (even number) บิตพาริตี จะมีค่าเท่ากับ 0
2. ถ้าหากจำนวนของบิตที่มีค่าเท่ากับ 1 ของสายบิตข้อมูลที่มีค่าเป็นเลขคี่ (odd number) บิตพาริตี จะมีค่าเท่ากับ 1

คอนโทรลที่ใช้ในการติดต่อสื่อสารของโปรแกรม Visual Basic

คอนโทรล MSComm

คอนโทรล MSComm(Communications) เป็นคอนโทรลตัวหนึ่งที่ใช้ช่วยในการติดต่อสื่อสารกับพอร์ตอนุกรม (serial port) ซึ่งเป็นคอนโทรลที่ทำงานโดยมีการตอบสนองต่อเหตุการณ์แบบ even-driven นั่นก็คือคอนโทรลจะทำหน้าที่ตรวจสอบการเกิดขึ้นหรือการร้องขอให้เกิดเหตุการณ์ต่างๆ กับพอร์ตอนุกรมโดยอัตโนมัติ และจะมีกาแจ้งเตือนให้ผู้อ่านได้ทราบโดยผ่านทางโพธิ์เซอร์เหตุการณ์ เช่นเดียวกับคอนโทรลทั่วไปของ Visual Basic นั่นเอง



รูปที่ 2.24 Microsoft Communication Control

การนำ MSComm มาใช้งานมีรายละเอียดดังนี้

1. ต้องมี MSComm ซึ่งจะมียู่แล้วเมื่อตอนติดตั้ง Visual Basic
2. กำหนด Custom Control MSCOMM.OCX (สำหรับ Visual Basic4 ขึ้นไป) โดยการเลือก MSCOMM.OCX จากเมนู Project/Components/ที่ Control เลือก Microsoft Comm Control 6.0 (กรณีที่ใช้ Version 6) ก็จะได้ MSComm รูปโทรศัพท์ปรากฏในทูลบ็อกซ์
3. นำคอนโทรลไปใส่ไว้ใน Form ที่ต้องการใช้งาน
4. กำหนดคุณสมบัติต่างๆ ให้กับ Control เขียนฟังก์ชันที่เกี่ยวข้องกับ MSComm ในการใช้ MSComm ติดต่อกับ Serial Port นั้นมีการติดต่อกัน 2 แบบ คือ
 - 1) การสื่อสารกระตุ้นด้วยเหตุการณ์ (Even-Driven Communication) หรือเป็นการเขียนกระตุ้นด้วยการอินเตอร์รปนั่นเอง โดยเมื่อข้อมูลเข้ามาจะเกิด CommEvent ขึ้นกับ OnComm Event
 - 2) การสื่อสารแบบโพลลิ่ง (Polling) โดยการวนรอบตรวจสอบข้อมูลจาก Serial Port ตลอดเวลา

การกำหนด Property ที่สำคัญ

ComPort คือการกำหนดหมายเลขของ Port ในการติดต่อ RS232 เช่น Com1, Com2
รูปแบบการใช้งาน

Object.ComPort [= value]

MSComm1.Comport = 1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Setting

คือการกำหนดพารามิเตอร์ในการสื่อสารผ่านพอร์ตอนุกรม
รูปแบบการใช้งาน

Object.Setting [=value]

MSComm1.Setting = "9600,e,7,2"

Value หมายถึง ข้อมูลชนิดสตริง ที่กำหนดพารามิเตอร์ในการสื่อสารผ่าน
พอร์ตอนุกรม สำหรับคุณสมบัติ Setting จะต้องเรียงลำดับดังนี้

"BBBB,P,D,S"

- **B** คือความเร็วของการส่งถ่ายข้อมูลในหน่วยของ baud rate
สำหรับค่า baud rate คอนโทรล MSComm สามารถปรับได้มีค่า
ดังต่อไปนี้ 110,300,600,1200,2400,9600,14400,19200,28800,38400,
56000,128000 และ 256000 ซึ่งค่า 9600 จะเป็นค่า default
- **P** คือบิตพาริตี (Parity Bit) สำหรับใช้ในการตรวจสอบ
ความถูกต้องของข้อมูล ซึ่งมีความหมายดังต่อไปนี้

พารามิเตอร์	ความหมาย
E	Even
M	Mark
N	None(default)
O	Odd
S	Space
- **D** คือขนาดของบิตข้อมูล ซึ่งสามารถมีค่าได้ดังต่อไปนี้
4,5,6,7 หรือ 8 (default)
- **S** คือขนาดของบิตหยุด (stop bit) ซึ่งสามารถมีค่าได้ดังต่อไปนี้
1(default) หรือ 2

PortOpen

เป็นการกำหนดสถานะการเปิด(open) หรือปิด (close) ของพอร์ตอนุกรม
ของเครื่องคอมพิวเตอร์

รูปแบบการใช้งาน

Object.PortOpen [=boolean]

MSComm1.PortOpen = true

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Boolean หมายถึง ข้อมูลชนิดบูลีน ที่กำหนดสถานะของพอร์ตอนุกรม ดังต่อไปนี้

True หมายถึง พอร์ตอนุกรมถูกเปิด

False หมายถึง พอร์ตอนุกรมถูกปิด

Input เป็นการย้ายข้อมูลจากบัฟเฟอร์การรับข้อมูล
รูปแบบการใช้งาน

Object.Input

MSComm1.Input = ข้อมูลที่จะรับจากพอร์ต

Output เป็นการส่งข้อมูลให้กับบัฟเฟอร์ส่งข้อมูลส่งออก
รูปแบบการใช้งาน

Object.Output [=value]

MSComm1.Output = ข้อมูลที่จะส่งออกพอร์ต

โพซีเอร์เหตุการณ์ OnComm

เกิดขึ้นเมื่อมีการเปลี่ยนแปลงค่าของคุณสมบัติ CommEvent ซึ่งเป็นการ
บอกถึงการเกิดข้อผิดพลาดหรือมีการสื่อสารเกิดขึ้นก็ได้

รูปแบบโพซีเอร์เหตุการณ์

Private Sub object_OnComm()

Modbus Protocol

Modbus Protocol เป็นโปรโตคอลในการสื่อสารข้อมูลระหว่าง Computer กับ PLC , PLC กับ PLC หรือระบบอื่นๆ ของ Modicon ซึ่งปัจจุบันนอกจากจะใช้กับ PLC แล้วยังใช้ได้กับอุปกรณ์อื่นๆ อีก เช่น SCADA , FLOW COMPUTER เป็นต้น ทั้งนี้เพราะใช้งานง่ายและแพร่หลายกว่าโปรโตคอลอื่นๆ

Modbus เป็นระบบ Master-Slave โดยมาสเตอร์ 1 ตัว สามารถต่อเข้ากับสลาฟ ได้ 246 ตัว และมาสเตอร์เท่านั้นที่จะส่งคำถาม (Request Query) ไปยังสลาฟและสลาฟจะส่งคำตอบกลับมา

การส่งข้อมูลระหว่าง Master-Slave

1. มาสเตอร์จะส่งคำถามที่มีรูปแบบดังนี้

SlaveID	Function	Start	Start	NO.of	NO.of	CRC/LRC
	Code	Addr(HO)	Addr(LO)	Point(HO)	Point(LO)	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. สลอฟ (PLC) ที่มี ID ตรงกับที่มาสเตอร์ส่งคำถาม (Request) มา (จะมีเพียงตัวเดียวเท่านั้น) จะส่งคำตอบกลับไปยังมาสเตอร์ ซึ่งมีรูปแบบทั่วไปดังนี้

SlaveID	FunctionCode	ByteCount	Data.....	CRC/LRC
---------	--------------	-----------	----------------	---------

ตัวอย่างฟังก์ชันของ Modbus

Function Code	Description
01	Read Coil status
02	Read input status
03	Read multiple registers
04	Read input registers
05	Write Coil
06	Write single register
15	Force multiple registers
16	Write multiple registers
23	read/write registers
etc.	

ความหมายของแต่ละ Field มีดังนี้

1. SLAVE ID หมายถึง หมายเลขประจำตัวของอุปกรณ์ที่สลอฟ ซึ่งจะไม่ซ้ำกัน
2. FUNCTION CODE หมายถึง หมายเลขฟังก์ชันของ Modbus ใ้บอก Slave ว่า มาสเตอร์ต้องการทำอะไร หรือต้องการข้อมูลแบบใดจากสลอฟ ดังตัวอย่างฟังก์ชันข้างบน
3. START ADDR.(HO,LO) หมายถึง ตำแหน่งเริ่มต้นของข้อมูลที่ต้องการอ่าน (HO ย่อมาจาก High order ,LO ย่อมาจาก Low order)
4. NO. OF POINT (HO,LO) หมายถึง จำนวนข้อมูลที่ต้องการจะนับจากตำแหน่งเริ่มต้นที่ระบุไปแล้ว
5. BYE COUNT หมายถึง จำนวนข้อมูลที่ส่งมาไว้ว่ามีทั้งสิ้นกี่ไบต์ (จะเป็นส่วนของคำตอบบางฟังก์ชัน)
6. CRC/LRC หมายถึง ข้อมูลที่เพิ่มเข้าไปในการส่งข้อมูลเพื่อใช้ตรวจสอบความผิดพลาด

โหมดการรับ-ส่งข้อมูล

การส่งรับ-ข้อมูลทาง Modbus ทำได้สองแบบคือ โหมด ASCII และโหมด RTU ซึ่งทั้งสองแบบมีข้อแตกต่างกันดังตาราง

ตารางความแตกต่างระหว่างโหมด ASCII กับ RTU

Description	ASCII (7 Bit)	RTU (8 Bit)
Coding	ASCII Code 0-9, A-F	8 Bit Binary
Number of Bits Per Charracter		
* Start bits	1	1
* Data bits	1	1
* Parity	1	1
* Stop bits	1 or 2	1 or 2
Error Check	LRC	CRC

สำหรับข้อดีของโหมด RTU คือ จำนวนไบต์น้อยกว่าทำให้ส่งข้อมูล (Data) ได้เร็ว ส่วนข้อดีของโหมด ASCII คือตรวจสอบได้ง่าย

เฟรมการส่งแบบแอสกี (ASCII Framing)

BEG OF FRAME	ADDRESS	FUNCTION	DATA	ERROR	EOF CHECK	READY TO RECRESP
	2-CHAR	2-CHAR	N*4CHR	2-CHAR	CR	LF
	16-BITS	16-BITS	N*16BIT	16-BITS		

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เฟรมการส่งแบบรีโมทเทอร์มินัลยูนิต (Remote Terminal Unit (RTU) Framing)

T1 T2 T3	ADDRESS	FUNCTION	DATA	CHECK	T1 T2 T3
	8-BITS	8-BITS	N*8-BITS	16-BITS	

การตรวจสอบความผิดพลาด (ERROR DETECTION)

ปกติความผิดพลาด (Error) จะมีอยู่สองลักษณะคือ Communication Error และ Operation Error ในกรณี Communication Error ถ้าสถานีพบว่าข้อมูลผิดพลาดมันจะไม่ส่งคำตอบ (Response) มาให้มาสเตอร์ และมาสเตอร์ก็จะส่งคำถามไปตามใหม่ เมื่อมันรอคำตอบจนถึงเวลาที่ตั้งไว้ หรือเมื่อพบว่ามีความผิดพลาด สถานีอาจจะส่งข้อความแจ้งความผิดพลาดกลับมาให้มาสเตอร์ก็ได้

การตรวจสอบความผิดพลาดมี 2 วิธี

1. Parity จะใช้วิธีเพิ่มจำนวนบิต 1 บิต เข้าไปในเฟรมของข้อมูลในแต่ละไบต์ที่ส่งไปอาจเป็น 1 หรือ 0 แล้วแต่นชนิดของพาริตีที่ใช้ ซึ่งวิธีนี้จะตรวจสอบได้เฉพาะความผิดพลาดที่เกิดเพียง 1 บิต เท่านั้น ถ้าความผิดพลาดเกิดขึ้นมากกว่า 1 บิต จะไม่สามารถบอกได้

2. Redundancy Check วิธีนี้สามารถตรวจสอบความผิดพลาดที่เกิดขึ้นหลายบิตได้ มีอยู่ด้วยกันสองแบบ คือ Cyclical Redundancy Check (CRC) ซึ่งใช้กับโหมด RTU และแบบ Longitudinal Redundancy Check (LRC) ใช้กับโหมด ASCII

ในการส่งข้อมูลระหว่างกัน ตัวส่งจะส่งข้อมูลพร้อมกับ CRC/LRC ออกไป ส่วนตัวรับก็จะรับข้อมูล และคำนวณ CRC/LRC ออกมา แล้วเปรียบเทียบกับ CRC/LRC ที่ส่งมาว่าตรงกันหรือไม่ ถ้าไม่ตรงแสดงว่ามีการผิดพลาด

2.1) Cyclical Redundancy Check (CRC) ที่ได้จะมีขนาด 16 บิต และมีวิธีการคำนวณดังนี้

1. โหลดค่าคงที่ FFFFH (16 บิต) accumulator (ACC)
2. นำข้อมูล 8 บิต (ที่จะส่ง) มา EX-OR
3. Shift Right 1 บิต
4. ถ้าบิตที่ Shift มาเป็น 1 ให้นำ A001H มา EX-OR กับ ACC แต่ถ้าบิตที่ Shift มาเป็น 0 ก็ไม่ต้องทำอะไร
5. ถ้ายังไม่ครบ 8 บิต ให้กลับไปทำข้อ 3

6. ใช้ค่าผลลัพธ์ที่ได้ใน ACC เป็นตัวตั้งแทน FFFFH ในการคำนวณ สำหรับข้อมูล 8 บิต ตัวต่อไปทำเช่นนี้จนครบทุกไบต์ของแพคเกจ

หมายเหตุ การคำนวณ CRC จะใช้ได้เฉพาะข้อมูล 8 บิตเท่านั้น โดยบิต Start, Parity, Stop จะไม่นำมาคำนวณ

2.2) Longitudinal Redundancy Check มีขนาด 8 บิต มีขั้นตอนการคำนวณดังนี้

Transmitter

1. หาค่าผลรวม (SUM) ของข้อมูลทุกไบต์ที่จะส่งไป
2. นำผลลัพธ์จากข้อ 1 มาหาค่า Complement อันดับ 2 จะได้เป็น LRC ส่งไปพร้อมกับข้อมูล

Receiver

1. รับค่าทั้งหมดมาแล้วหาค่าผลรวม (SUM รวม LRC ด้วย)
2. ถ้าผลลัพธ์ที่ได้เป็น 00H แสดงว่าไม่มีการผิดพลาด

บทที่ 3

รายละเอียดการสร้างและประกอบโครงงาน

ในโครงงานนี้แบ่งออกเป็น 2 ส่วน คือ

- ส่วนของฮาร์ดแวร์(Hardware) ได้แก่

Electrical Hardware

Mechanical Hardware (Warehouse Model)

- ส่วนของซอฟต์แวร์(Soft Ware)

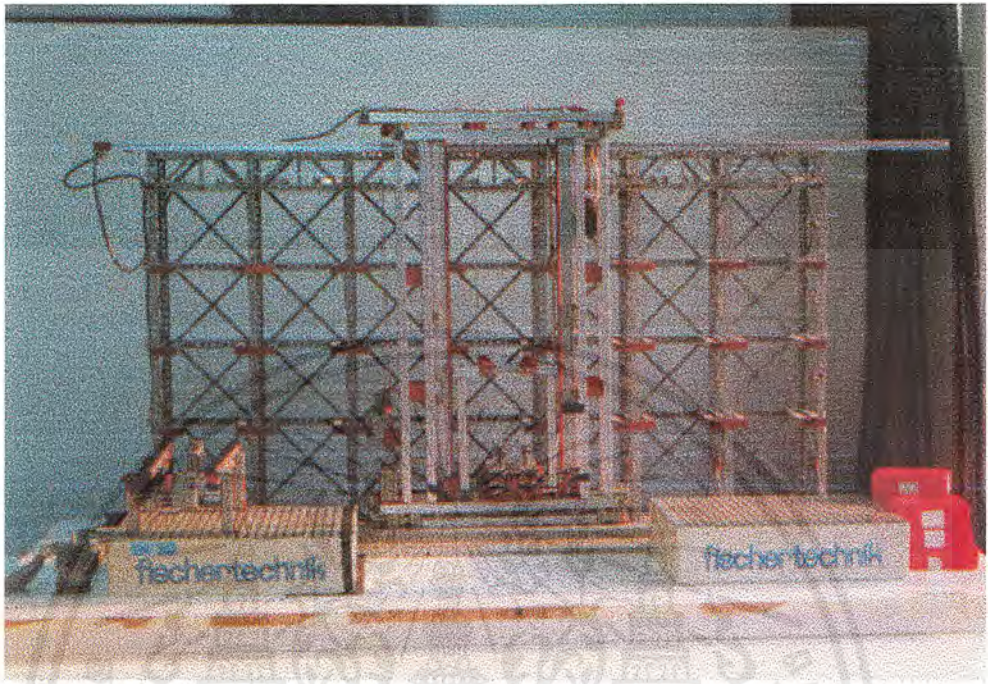
แผนผังขั้นบันได (Ladder Program)

โปรแกรมการจัดการฐานข้อมูล และติดต่อกับผู้ใช้งาน

3.1 ส่วนฮาร์ดแวร์ (Hardware)

3.1.1 คลังจัดเก็บสินค้าจำลอง (Warehouse Model)

จากโครงงานนี้ได้ใช้แบบจำลองคลังจัดเก็บสินค้า ซึ่งมีจำนวน 3 แถว 7 หลัก จะมีสายพานลำเลียง คอยส่งวัตถุเข้าออก และมีคันลำเลียงคอยรับวัตถุเข้าเก็บตามชั้นและหลักการต่างๆ



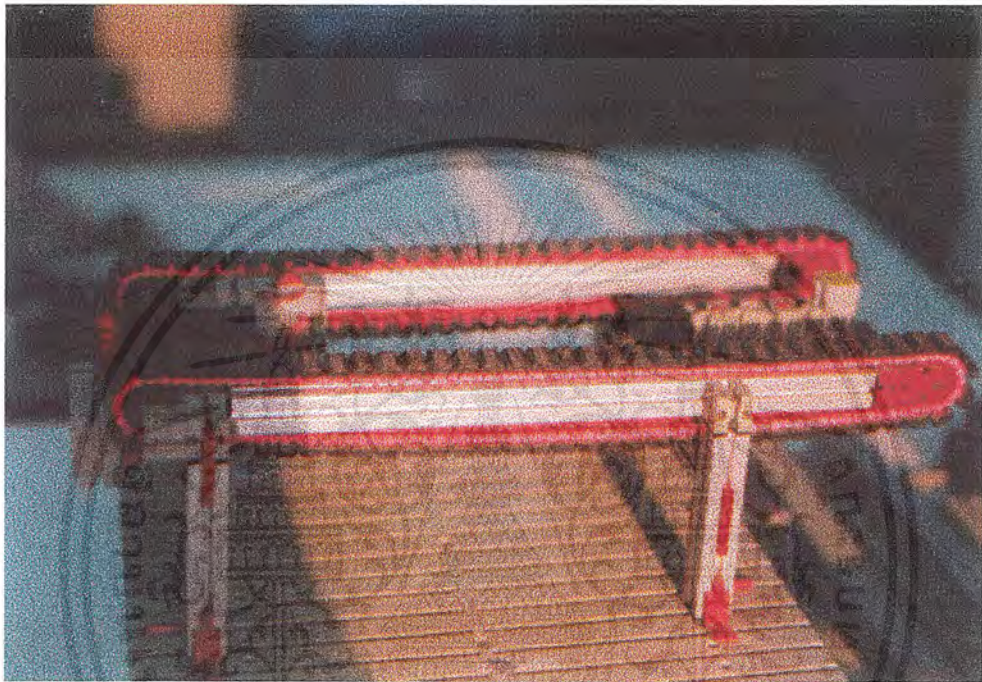
รูปที่ 3.1 แสดงแบบจำลองคลังจัดเก็บสินค้าที่ใช้เป็นตัวอย่างในโครงการนี้

ระบบขับเคลื่อน

ระบบการขับเคลื่อนได้ใช้การขับเคลื่อนโดยมอเตอร์กระแสตรง (DC MOTOR) ส่งกำลังผ่านเฟืองและลูกกรอก ทำให้สายพานและคานละเลียงเคลื่อนที่ได้

อุปกรณ์ในการขับเคลื่อนและการลำเลียง

1. สายพานลำเลียง จะทำหน้าที่เคลื่อนวัสดุเข้าสู่ถาดลำเลียง โดยตัวสายพานลำเลียงจะมีลิมิตสวิทช์ (Limit Switch) คอยตรวจสอบเมื่อมีวัสดุเข้าสู่สายพานลำเลียงและมีฟร็อกซิมีตี-สวิทช์ (Proximity Switch) เป็นตัวคอยตรวจสอบเมื่อวัสดุมาอยู่ตำแหน่งสุดท้ายสายพานลำเลียง Proximity Switch จะเป็นสวิทช์อินพุทของเครื่องควบคุมแบบโปรแกรมได้ คอยสั่งงานให้ ถาดลำเลียง ยื่นมารับวัสดุในลำดับต่อไป การขับเคลื่อนให้สายพานลำเลียงเคลื่อนที่จะขับเคลื่อนด้วย มอเตอร์ (M 1) ผ่านเฟือง



รูปที่ 3.2 แสดงสายพานลำเลียง

2. ถาดลำเลียงในแนวแกน Z ที่หน้าที่คอยยื่นรับวัสดุเข้ามาจากสายพานลำเลียง และเมื่อถาดลำเลียงเคลื่อนที่ถึงตำแหน่งของ หลั๊ก และแถวที่ต้องการแล้ว ถาดลำเลียงในแนวแกน Z ยังมีหน้าที่คอยยื่นส่งวัสดุเข้าไปในช่องที่ต้องการเก็บวัสดุอีกด้วย โดยมีตัวลิมิตสวิทช์ 3 ตัวคอยตรวจสอบตำแหน่งของถาดลำเลียงคือ

1. ยื่นออกมาสุด
2. อยู่ในตำแหน่งตรงกลาง
3. เข้าไปในสุดเพื่อส่งเข้าวัสดุ

การขับเคลื่อนให้ถาดลำเลียงเคลื่อนที่จะขับเคลื่อนด้วย มอเตอร์ (M 2) ผ่านเฟืองตรง



รูปที่ 3.3 แสดงถาดลำเลียง และทิศทางการเคลื่อนที่ (แนวแกน Z)

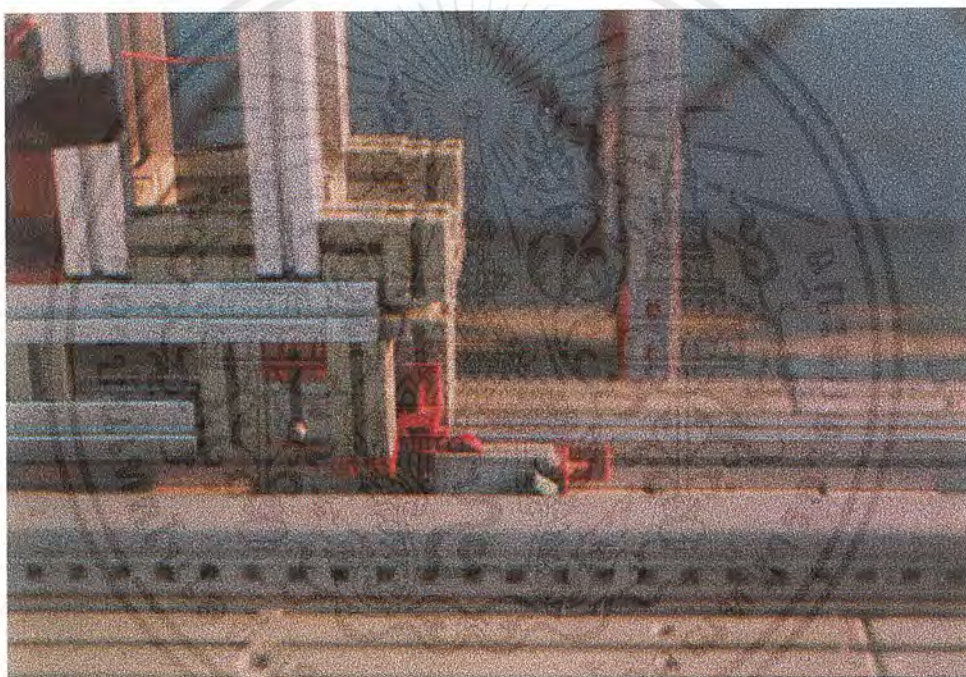
3. รอกยกระดัดด้านแกน Y รอกยกระดัดทางด้านแกน Y จะทำหน้าที่ยกถาดลำเลียงขึ้นและลงสู่ชั้นต่างๆที่เราต้องการ โดยมีลิฟต์สวิทช์ 6 ตัวคอยตรวจสอบตำแหน่งสำหรับการเคลื่อนที่ 3 ชั้น แต่ละชั้นจะมีลิฟต์สวิทช์อยู่ 2 ตัว รอกยกระดัดจะถูกขับเคลื่อนด้วยมอเตอร์ (M3) ผ่านระบบเฟืองและรอกอีกทีหนึ่ง



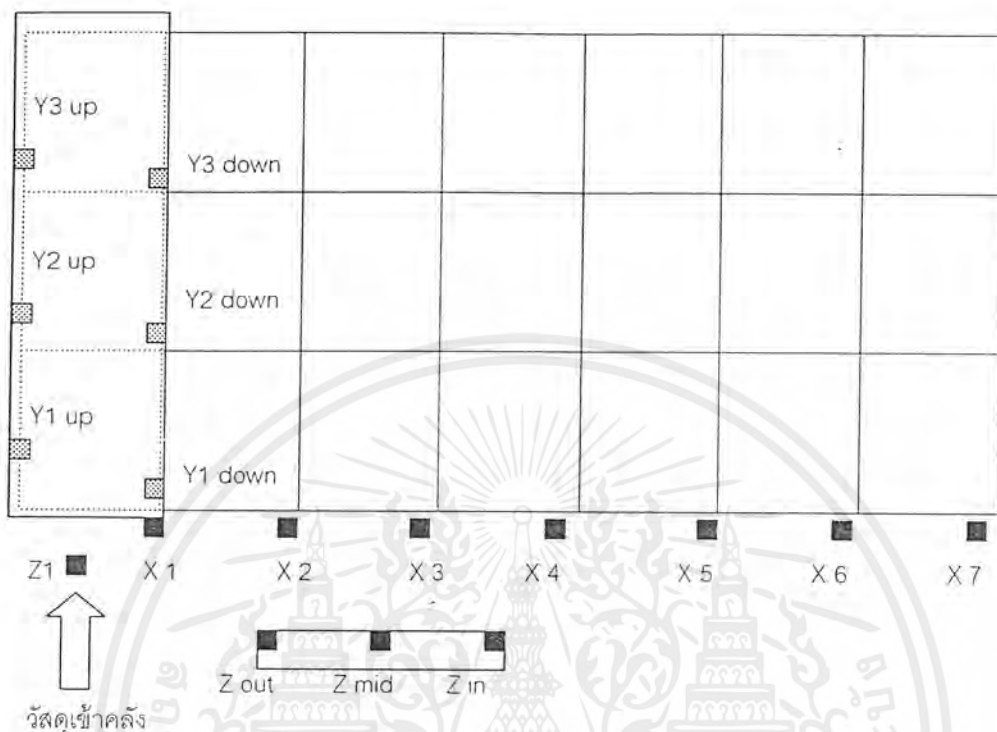
รูปที่ 3.4 แสดงระบบเฟืองที่ใช้ในการทำงานของรอกยกระดับด้านแกน Y

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. คานเลื่อนตามแนวแกน X คานเลื่อนตัวนี้จะคอยเลื่อนถาดดำเตียง และรอกยกกระดานซึ่งติดอยู่ด้วยกันให้เคลื่อนที่ไปตามแกน X โดยมีลิimitsวิตซ์ ด้านล่างคอยตรวจสอบตำแหน่งตามแนวแกน X ทั้งหมด 7 ตัว โดยแต่ละตัวประจำอยู่ตามหลักที่ 1 - 7 คานเคลื่อนที่ได้โดยมีมอเตอร์ (M 4) ขับเคลื่อนผ่านเฟืองทดและตัวถูกล้อ



รูปที่ 3.5 แสดงตำแหน่งการเคลื่อนที่ตามแนวแกน X และตำแหน่งของลิimitsวิตซ์



รูปที่ 3.6 แสดงตำแหน่งของลิมิตสวิตช์ ตามตำแหน่งต่างๆ

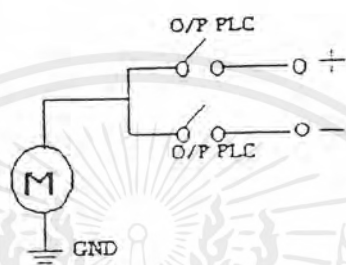
ลิมิตสวิตช์จะตรวจจับตำแหน่งต่างๆ ของแบบจำลองคลังจัดเก็บสินค้า เพื่อส่งสถานะ ON และ OFF ให้แก่เครื่องควบคุมแบบโปรแกรมได้ โดยสามารถสรุปค่าอินพุตที่เป็นลิมิตสวิตช์ และพรีอักษิตีลวิตช์ ได้ดังนี้

X1 – X7	คอยตรวจจับตำแหน่งของคานเลื่อนให้ตรงตามหลักต่างๆ ทั้ง 7 หลัก
Z1	คอยตรวจจับเมื่อมีสินค้าเข้าสู่สายพานลำเลียง
Z out, Z mid, Z in	คอยตรวจสอบตำแหน่งของถาดลำเลียงทั้ง 3 ตำแหน่ง คือ out, mid, in
Y1 – Y3 DOWN	คอยตรวจสอบตำแหน่งของถาดลำเลียงด้านต่ำ
Y1 – Y3 UP	คอยตรวจสอบตำแหน่งของถาดลำเลียงด้านสูง

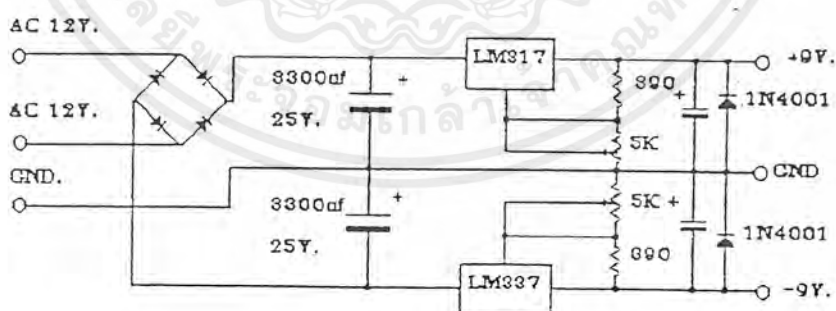
ระบบการขับเคลื่อนมอเตอร์

มอเตอร์ทั้ง 4 ตัว เป็นมอเตอร์กระแสตรง ขนาดเล็ก ทนแรงดันได้ไม่เกิน 9 โวลท์ แรงดันปกติ 5 ถึง 8 โวลต์

ในการทำงานของมอเตอร์นั้นมอเตอร์จะต้องหมุน 2 ทิศทาง การขับเคลื่อนใช้แรงดันไฟเลี้ยงที่ทั้งบวกและลบ เพื่อเปลี่ยนทิศทางการหมุนของมอเตอร์



รูปที่ 3.7 วงจรการกลับทิศของมอเตอร์



รูปที่ 3.8 แสดงภาคจ่ายไฟแรงดันคงที่

3.1.2 PLC Modicon 512/00 Micro PLC

ทำหน้าที่เป็น Electrical Hardware รายละเอียดได้กล่าวไปแล้วในบทที่ 2 ในส่วนของ Configuration, Modsoft, Set port ส่วนการต่อ Input และ Output จะกล่าวรายละเอียดในภาคผนวก

3.2 ซอร์ฟแวร์

3.2.1 แผนผังขั้นบันได (Ladder Program)

การทำงานของ PLC ในโครงงานนี้ได้แบ่งออกเป็น 2 MODE การควบคุมหลักได้แก่

1. โหมดรับ เป็นการเก็บสินค้าเข้าในคลังจัดเก็บสินค้า
2. โหมดส่ง เป็นการเอาสินค้าออกจากคลังจัดเก็บสินค้า

ซึ่งค่าต่างๆ คือค่าของโหมด ค่าของแถว และค่าของหลักที่ต้องการจะถูกคอมพิวเตอร์ป้อนเข้ามาเก็บไว้ในหน่วยความจำของ PLC ในรูปของตัวเลข 4 ตำแหน่งคือ “0ABC” ความหมายของตัวเลขในตำแหน่งดังกล่าวได้แก่

ตำแหน่งที่ 0 จะให้เป็นเลข “0” ตลอดเพราะไม่ได้ใช้งาน

ตำแหน่งที่ A จะเป็นค่าของ MODE ซึ่งมี 2 ค่า ได้แก่ 1 และ 0

1 แทน MODE รับ

0 แทน MODE ส่ง

ตำแหน่งที่ B จะเป็นค่าของแถวที่ต้องการจะไป รับหรือส่ง

ตำแหน่งที่ C จะเป็นค่าของหลักที่ต้องการจะไปรับหรือส่ง

ตัวอย่างเช่น

- ถ้าคอมพิวเตอร์ส่งค่า “0125” มาจะแปลความหมายได้ว่า PLC จะไปรับวัสดุที่เข้ามาไปเก็บไว้ที่แถวที่ 2 หลักที่ 5

- ถ้าคอมพิวเตอร์ส่งค่า “0031” มาจะแปลความหมายได้ว่า PLC จะไปนำวัสดุจากแถวที่ 3 หลักที่ 1 ส่งออกไป

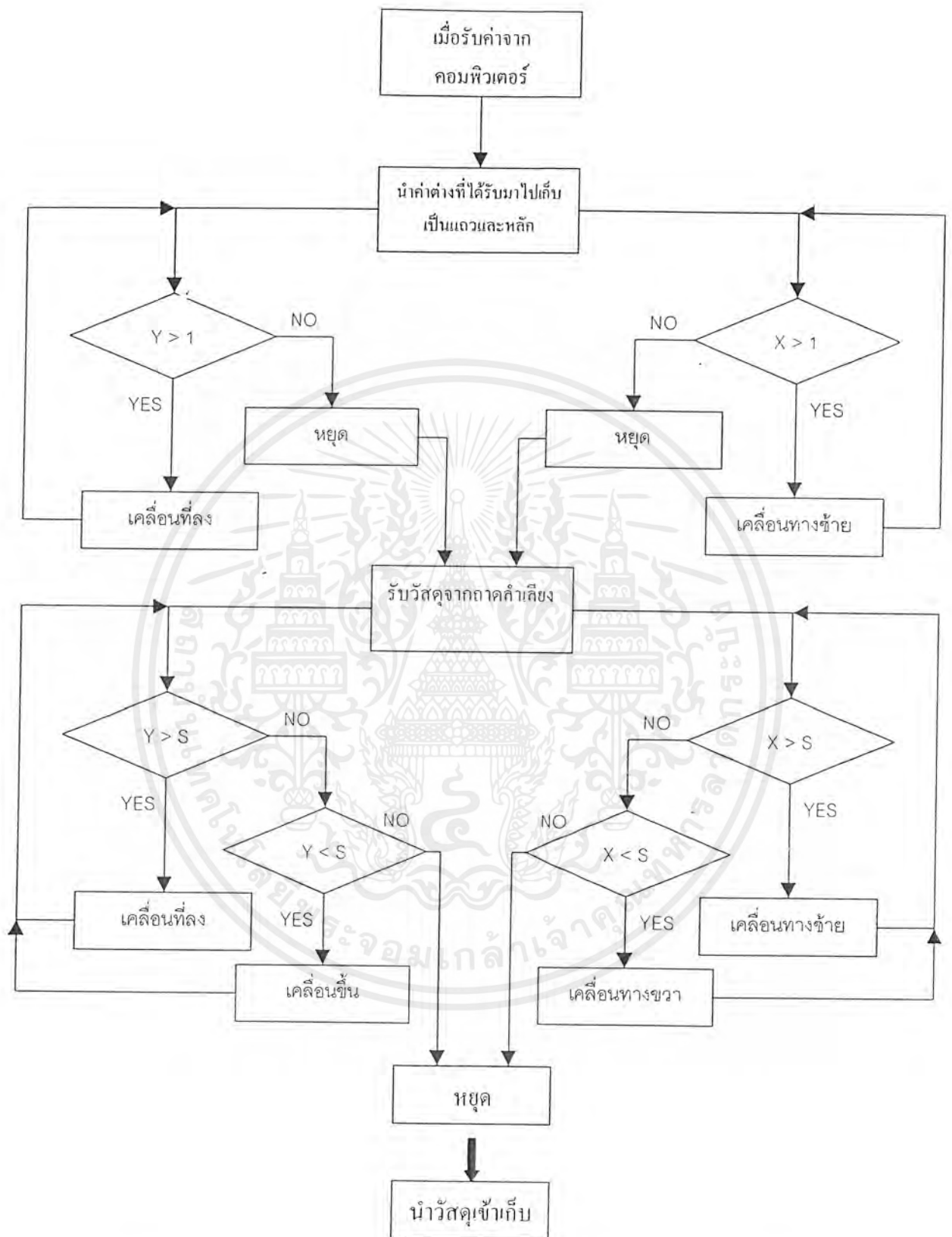
การทำงานของโหมดรับ

ตามโครงงานแล้ว เราจะนำวัสดุซึ่งเป็นตัวอย่างของสินค้าที่จะทำการจัดเก็บวางไว้บนสายพาน ทับลิเมตรสวิทช์ซึ่งเป็นตัวตรวจจับวัสดุเพื่อเริ่มการทำงานจะทำให้สายพานเลื่อนวัสดุนั้นไปเรื่อยๆ จนติดกับฟร็อกซ์มิเตอร์สวิทช์ จะทำให้สายพานหยุดและพร้อมกับเลื่อนเอาลิฟต์วัสดุมารับเอาวัสดุนั้นจากสายพาน แล้วจะดำเนินการจัดเก็บวัสดุตามตำแหน่งที่ได้ระบุไว้

การทำงานของโหมดส่ง

ในโหมดนี้จะวัสดุจะมีอยู่ในคลังสินค้าอยู่แล้วเมื่อคอมพิวเตอร์ส่งค่ามาลิฟต์จะเลื่อนไปนำวัสดุจากแถวและหลักที่ต้องการมาส่งออกที่สายพานลำเลียง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



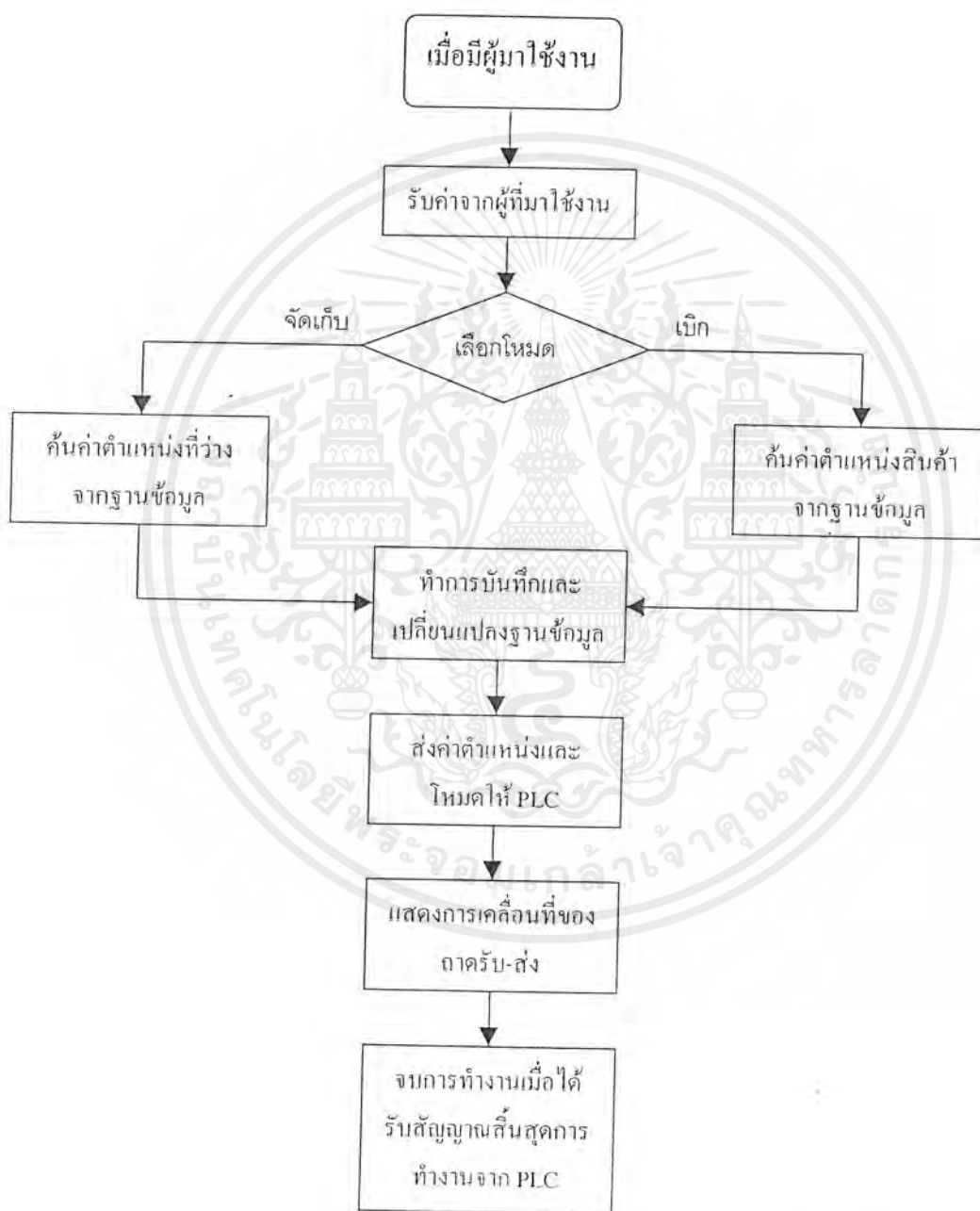
(Y คือตำแหน่งแถวปัจจุบัน, X คือตำแหน่งหลักปัจจุบัน, S คือค่าแอมและหลักที่รับมา)

รูปที่ 3.9 แสดงการทำงานของโหมดรีบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.2 โปรแกรมสำหรับการจัดการฐานข้อมูลและติดต่อสื่อสารกับผู้ใช้งาน

เมื่อมีการนำสินค้าเข้าจัดเก็บ หรือนำเอาออกไปใช้งาน จะต้องมีการบันทึกค่าข้อมูลลงในฐานข้อมูลการทำงานทั้งหมดแสดงให้เห็นดังต่อไปนี้ทุกครั้งที่มีการเปลี่ยนแปลง



รูปที่ 3.11 แสดงการทำงานของโปรแกรมการจัดการฐานข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

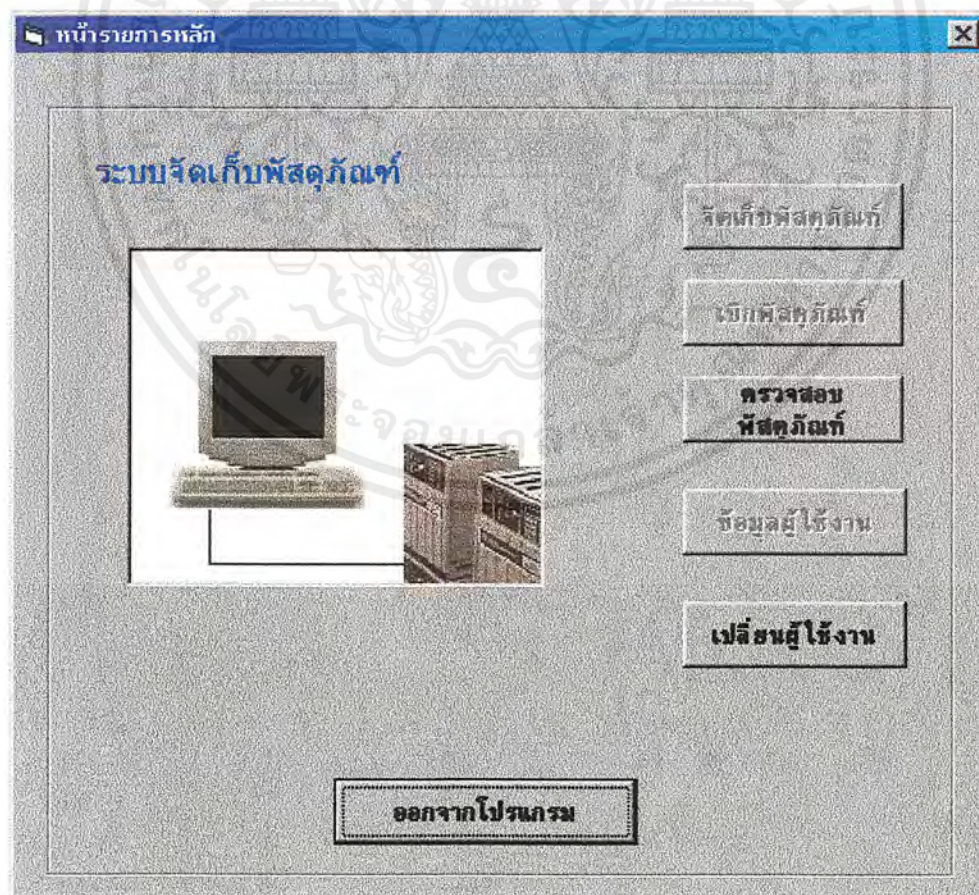
การทดลองโครงงาน

การทำงานในส่วนคอมพิวเตอร์ของโปรแกรม MS –Visual Basicจะมีความทำงานอยู่ 2 ส่วนหลัก คือ

1. ทำการจัดการกับฐานข้อมูลใน MS-Access
2. ทำการติดต่อกับ PLC โดยใช้ Modbus Protocol ในการติดต่อสื่อสาร

ซึ่งการทำงานในส่วนของคอมพิวเตอร์นี้จะเป็นระบบป้องกันการใช้งานจากภายนอก เนื่องจากในความเป็นจริงการนำเข้า-นำออกพัสดุภัณฑ์จะมีการอนุญาตให้สามารถกระทำได้เพียงบางบุคคลเท่านั้น เพื่อป้องกันพัสดุภัณฑ์สูญหาย หรือง่ายในการดูแลรับผิดชอบ

1. หน้าจอการเริ่มทำงานตอนแรก ผู้ใช้ต้องทำการกดปุ่มเปลี่ยนแปลงผู้ใช้งาน (Log in) เพื่อเข้าสู่ระบบ ซึ่งหากผู้ใช้งานมิได้ทำการ Log in เข้าไป การทำงานก็จะสามารถทำได้แค่เพียงตรวจสอบสถานะของ WareHouse ว่ามีพัสดุภัณฑ์อยู่ในช่องไหนบ้าง จำนวนเท่าไร



รูปที่ 4.1 แสดงหน้ารายการหลักก่อนมีการ Log in เข้าสู่ระบบ

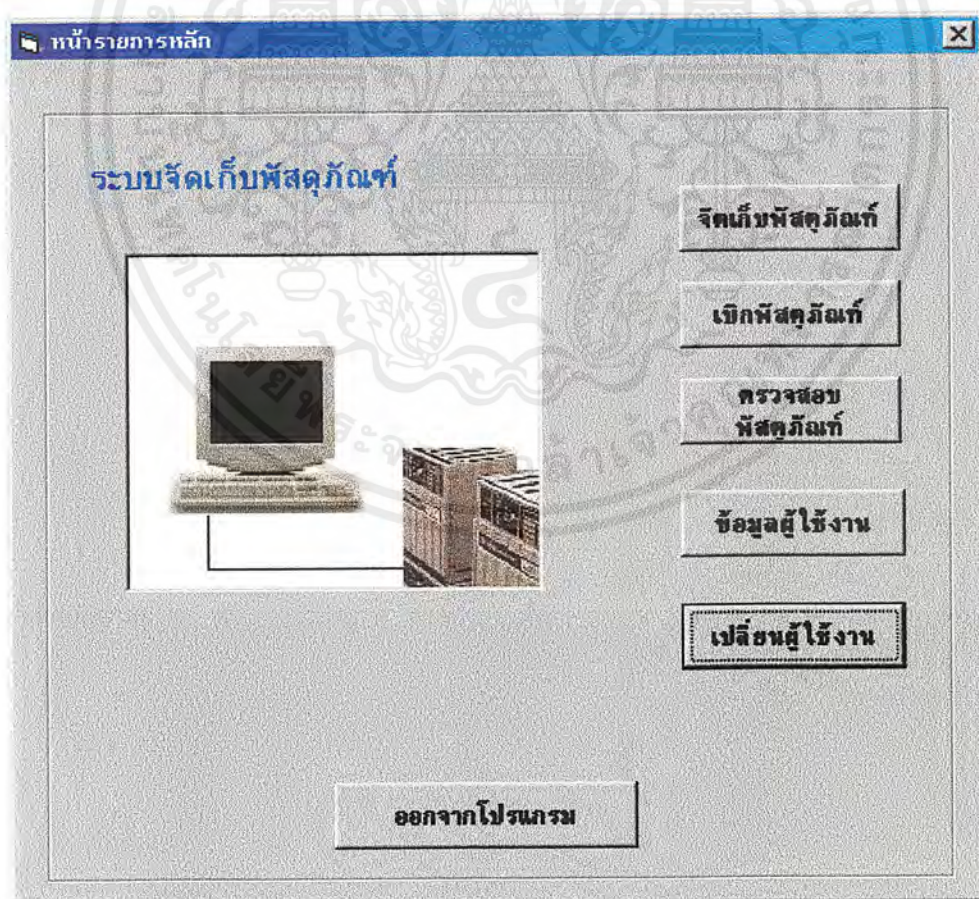
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. หน้าจอการ Log in นี้หากมีการนำเข้า-นำออก จะมีการบันทึกข้อมูลผู้เข้ามาใช้ในฐานะข้อมูลโดยอัตโนมัติ



รูป 4.2 แสดงหน้าจอในการเข้าสู่ระบบ (Log in) ของผู้ใช้งาน

3. เมื่อทำการ Log in เข้าสู่ระบบ ก็จะสมารถนำเข้าพัสดุภัณฑ์ นำออกพัสดุภัณฑ์ ตรวจสอบข้อมูลผู้ใช้งาน และตรวจสอบข้อมูลของพัสดุภัณฑ์ได้



รูปที่ 4.3 แสดงหน้าจอหลักที่สมบูรณ์หลังจากมีการ Log in

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. เมื่อทำการจัดเก็บพัสดุภัณฑ์ จะมีการบันทึกข้อมูลวัน/เวลาที่นำเข้า ผู้นำเข้า รหัสพัสดุภัณฑ์ จำนวน ตำแหน่งซึ่งได้จากการตรวจสอบจากคอมพิวเตอร์ในฐานข้อมูลว่าช่องใดว่างและส่งค่าให้ PLC เพื่อทำการจัดเก็บ

รูปที่ 4.4 แสดงหน้าจอในการนำเข้าพัสดุภัณฑ์

5. เมื่อทำการนำออกพัสดุภัณฑ์ จะมีการบันทึกข้อมูลวัน/เวลาที่นำออก ผู้นำออก รหัสพัสดุภัณฑ์ จำนวน ตำแหน่งซึ่งได้จากการตรวจสอบจากคอมพิวเตอร์ในฐานข้อมูลว่าช่องใดที่มีพัสดุภัณฑ์ที่ต้องการนำออกและการนำออกต้องเป็นระบบเข้าก่อนออกก่อน (first in first out) เพื่อเป็นการป้องกันสินค้าเก่าค้างสต็อก

รูปที่ 4.5 แสดงหน้าจอในการนำออกพัสดุภัณฑ์

6. เมื่อทำการตรวจสอบพัสดุภัณฑ์ที่สามารถตรวจสอบสถานะใน WareHouse ว่ามีพัสดุภัณฑ์ชนิดใดอยู่ในช่องไหน จำนวนเท่าไร และสามารถตรวจสอบรหัสพัสดุภัณฑ์กับชื่อได้โดยไม่ต้องเข้าไปดูใน MS-Access

WAREHOUSE

3	6	9	12	15	18	21
39014023		39014001	39014011			
1	0	1	1	0	0	0
2	5	8	11	14	17	20
39014026	39014013	39014003	39014013	39014006		
1	4	1	10	1	0	0
1	4	7	10	13	16	19
39014004	39014040	39014002	39014004	39014006		
1	1	1	1	1	0	0

☐ ตรวจสอบการนำเข้า-การนำออก
☐ ตรวจสอบชื่อพัสดุภัณฑ์จากรหัส

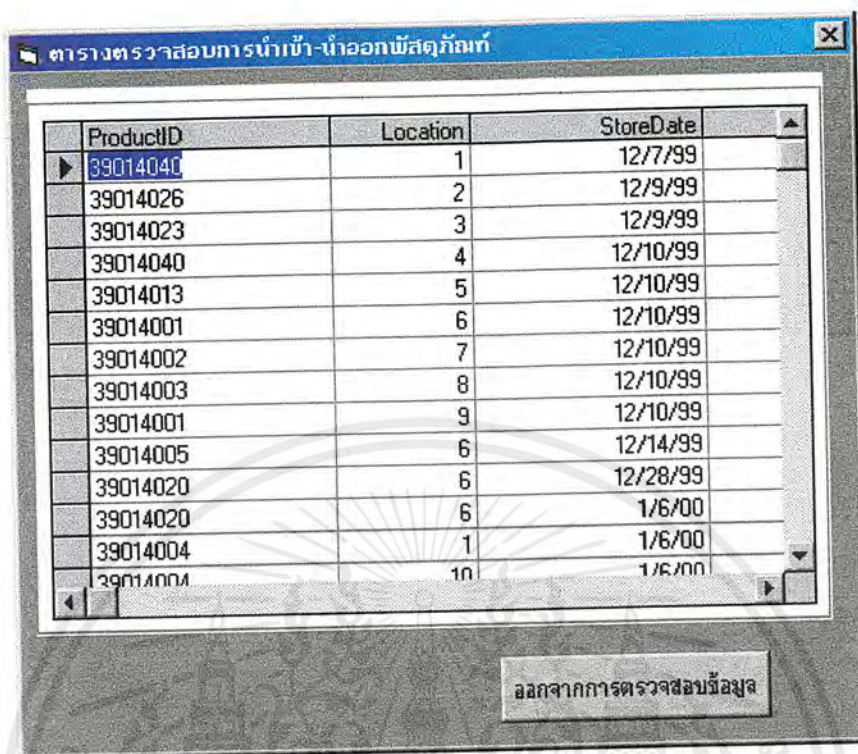
ออกจากการตรวจสอบข้อมูล

รูปที่ 4.6 แสดงสถานะของพัสดุภัณฑ์ใน WareHouse

ProductID	ProductName
39014001	Stainless Steel
39014002	Nut
39014003	R207 Angle
39014004	R209 Angle
39014005	Bolts w/nuts
39014006	R404 Roller
39014007	Lock Washer
39014008	Bracket
39014009	Hex Nut
39014010	PVC PIPE
39014011	Part Number Tag

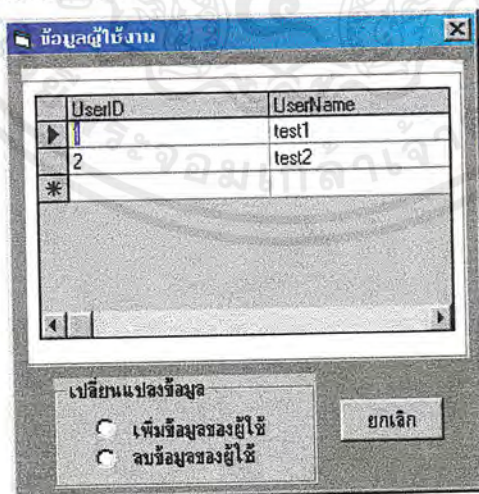
ออกจากการตรวจสอบ

รูปที่ 4.7 แสดงรหัสและชื่อพัสดุภัณฑ์



รูปที่ 4.8 แสดงหน้าจอการตรวจสอบผู้นำเข้า-ออก รหัสสินค้า วัน/เวลา ตำแหน่งที่จัดเก็บ

7. เมื่อทำการตรวจสอบผู้ใช้งาน สามารถตรวจสอบชื่อและรหัสของผู้ที่มีหน้าที่ในการจัดการกับพัสดุภัณฑ์ และสามารถเพิ่มหรือลบผู้ใช้งานออกได้จากหน้าจอนี้ โดยไม่ต้องไปทำการเปลี่ยนแปลงที่ตัวฐานข้อมูลในMS-Access โดยตรง



รูปที่ 4.9 แสดงหน้าจอข้อมูลของผู้ใช้โดยแสดงชื่อ รหัส แต่ไม่แสดงรหัสผ่าน



เพิ่มข้อมูลผู้ใช้งาน

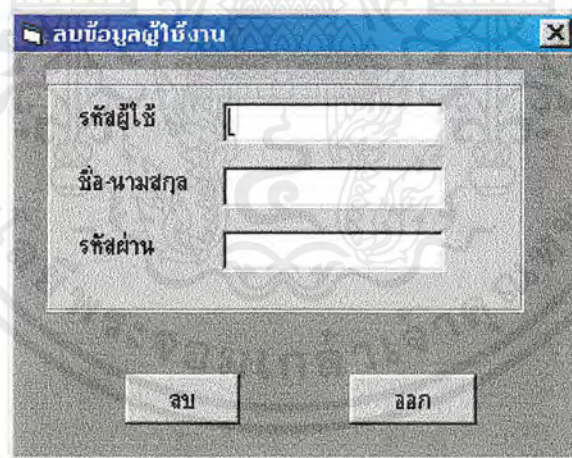
รหัสผู้ใช้

ชื่อ-นามสกุล

รหัสผ่าน

บันทึก ยกเลิก

รูปที่ 4.10 แสดงหน้าจอที่ใช้ในการเพิ่มข้อมูลผู้ใช้งานในฐานข้อมูล



ลบข้อมูลผู้ใช้งาน

รหัสผู้ใช้

ชื่อ-นามสกุล

รหัสผ่าน

ลบ ออก

รูปที่ 4.11 แสดงหน้าจอการลบข้อมูลของผู้ใช้งานในฐานข้อมูล

8. เมื่อมีการจัดเก็บหรือนำออกพัสดุภัณฑ์ หน้าจอจะแสดงการเคลื่อนที่ของตำแหน่งปัจจุบันของ ถาดรับ-ส่ง จนกว่าการจัดเก็บหรือนำออกจะเสร็จสมบูรณ์



รูปที่ 4.12 แสดงหน้าจอสถานะตำแหน่งของถาดรับ-ส่ง ในขณะที่นำเข้าหรือนำออก

บทที่ 5

สรุปผลการทดลองของโครงการและแนวทางในการพัฒนา

จากผลการทดลองเป็นไปตามวัตถุประสงค์ที่ตั้งไว้ แต่โครงการนี้ยังมีข้อจำกัดอยู่โดยที่คลังจัดเก็บสินค้ามีขนาด 3 แถวและ 7 เท่านั้น ถ้าจะต้องมีการนำไปใช้กับคลังสินค้าที่มีขนาดใหญ่หรือแตกต่างจากขนาดนี้ จะต้องมีการแก้ไขโปรแกรมบางส่วน

นอกจากนี้โครงการนี้ยังมีทางพัฒนาไปได้อีกหลายทางเพื่อให้เกิดประสิทธิภาพมากขึ้นในกระบวนการจัดเก็บ เช่น การนำทฤษฎีของคงคลังมาใช้ในโปรแกรม กำหนดจำนวนสินค้าที่ควรมีในคลัง เมื่อสินค้าใดเกิดมีจำนวนน้อยกว่าที่กำหนดไว้ ให้มีการเตือนขึ้นมานที่หน้าจอ หรือมีการบันทึกเวลากำหนดเวลาออก แต่เมื่อถึงเวลายังไม่มีการเอาสินค้าออกก็ให้มีการเตือนขึ้นมา หรืออาจจะจัดให้มีการพัฒนาเป็นการควบคุมและตรวจสอบสินค้าผ่านทาง Internet ได้

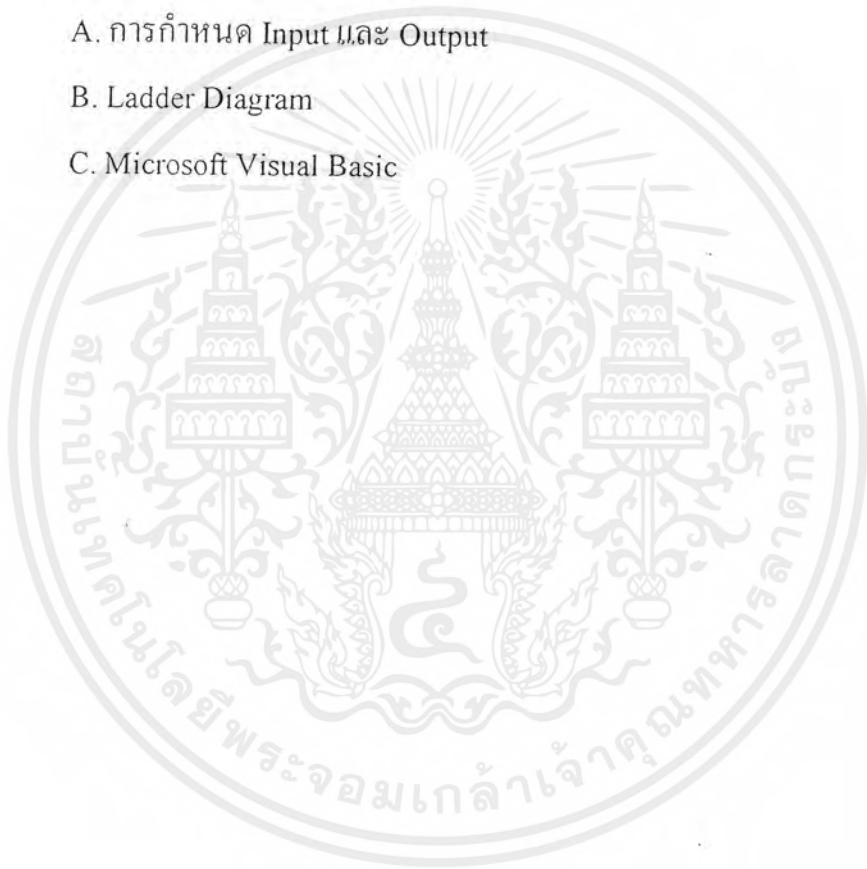
โครงการนี้เป็นเพียงตัวอย่างหนึ่งซึ่งเป็นการควบคุมระบบ ด้วยการทำงานร่วมกันของ PLC กับ คอมพิวเตอร์ และมีการจัดเก็บหรือตรวจสอบฐานข้อมูล เราอาจจะนำไปประยุกต์ใช้กับกระบวนการอื่นได้อย่างไม่ยากนัก

ภาคผนวก

A. การกำหนด Input และ Output

B. Ladder Diagram

C. Microsoft Visual Basic



ภาคผนวก A

การกำหนดอินพุตเอาต์พุต



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การกำหนดอินพุตและเอาต์พุต

Input

Limit switch column	1	10001
Limit switch column	2	10002
Limit switch column	3	10003
Limit switch column	4	10004
Limit switch column	5	10005
Limit switch column	6	10006
Limit switch column	7	10007

Limit switch row	1	up	10008
Limit switch row	1	down	10009
Limit switch row	2	up	10010
Limit switch row	2	down	10011
Limit switch row	3	up	10012
Limit switch row	3	down	10013

Limit switch ถาดรับ - ส่ง	out	10014
Limit switch ถาดรับ - ส่ง	midium	10015
Limit switch ถาดรับ - ส่ง	in	10016

Limit switch ที่สายพาน 10017

Proximity switch 10018

ภาคผนวก B

Ladder Diagram



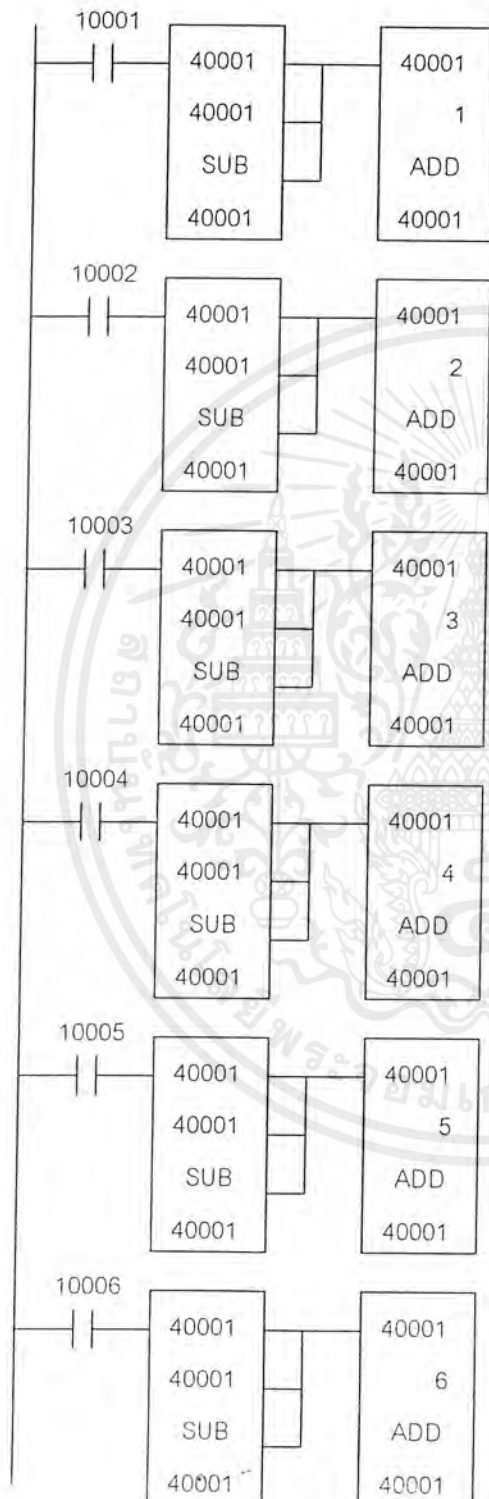
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Output

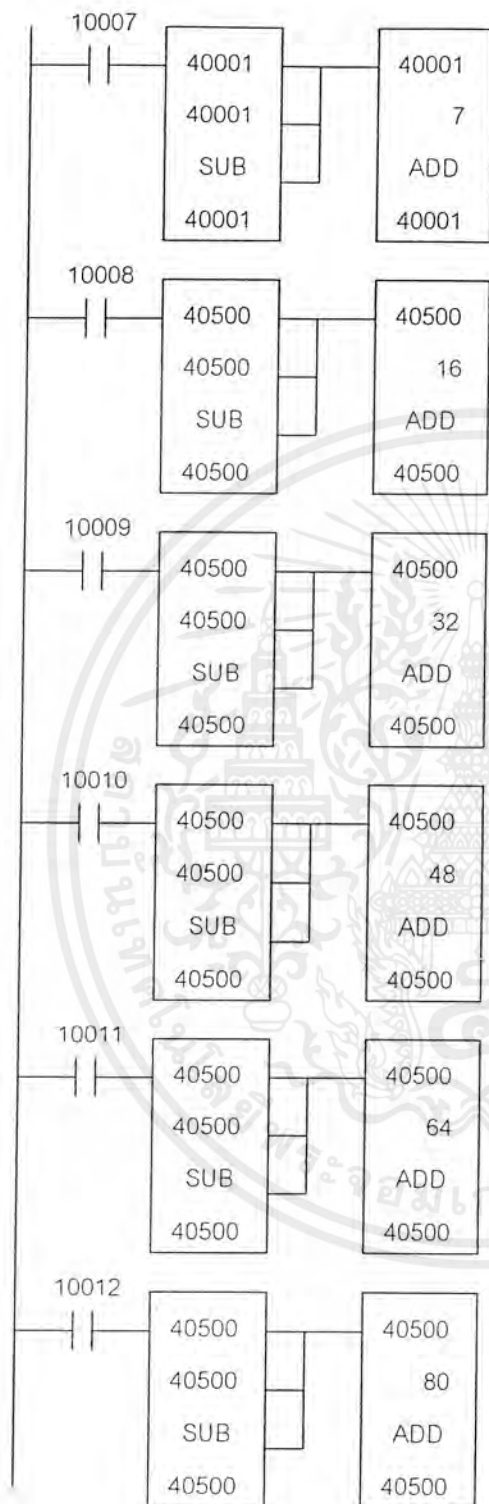
Motor ขับเคลื่อนในแนวแกน X เคลื่อนที่ทางซ้าย	00005
Motor ขับเคลื่อนในแนวแกน X เคลื่อนที่ทางขวา	00009
Motor ขับเคลื่อนในแนวแกน Y เคลื่อนที่ขึ้น	00006
Motor ขับเคลื่อนในแนวแกน Y เคลื่อนที่ลง	00010
Motor ขับเคลื่อนของถาด เคลื่อนที่เข้า	00007
Motor ขับเคลื่อนของถาด เคลื่อนที่ออก	00011
Motor ขับเคลื่อนของสายพาน เคลื่อนที่นำวัสดุเข้า	00008



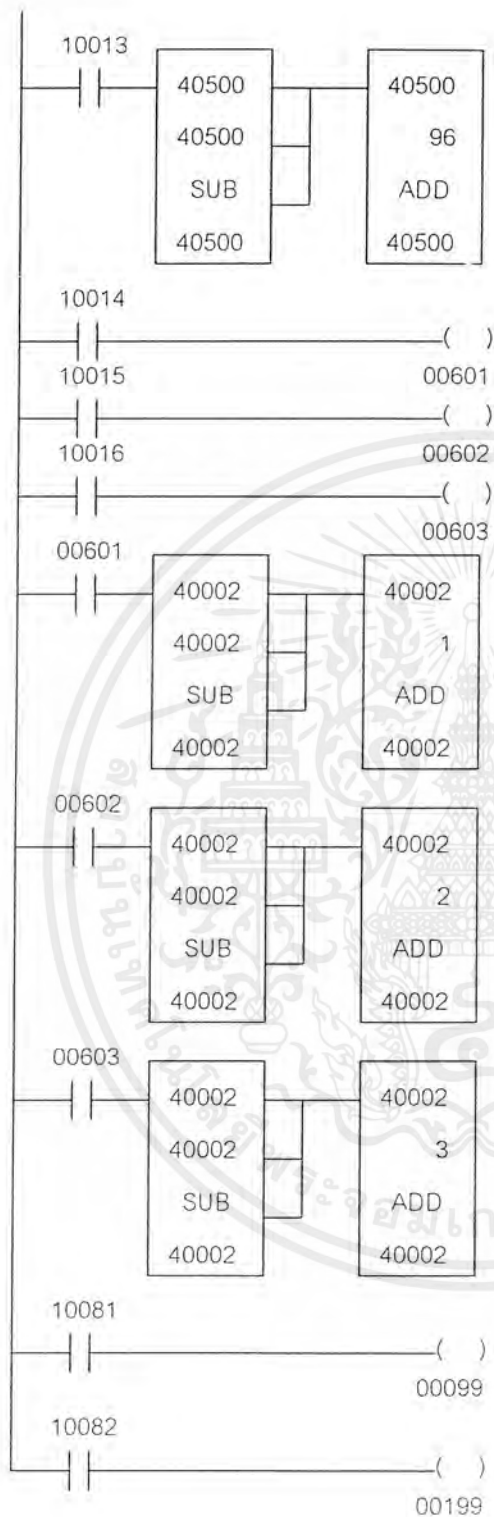
LADDER DIAGRAM



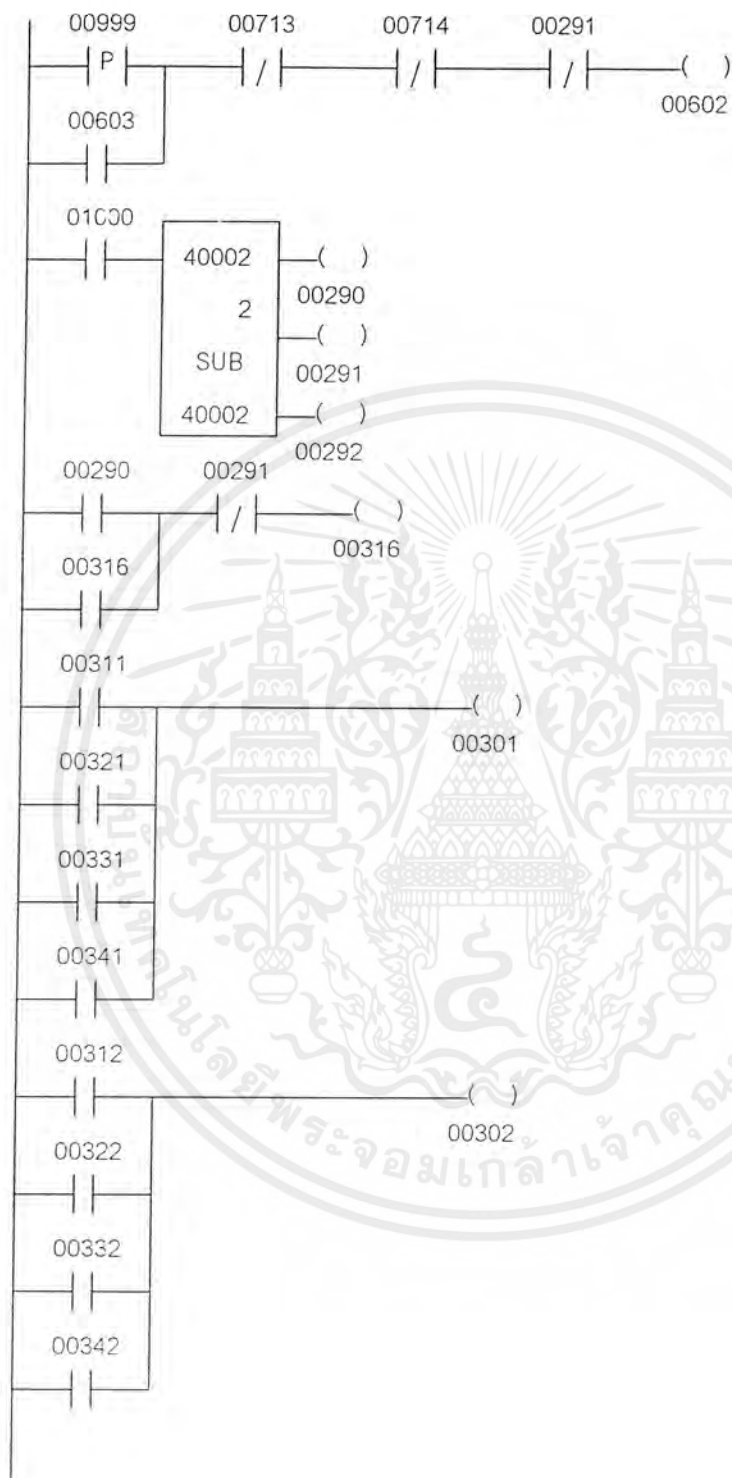
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



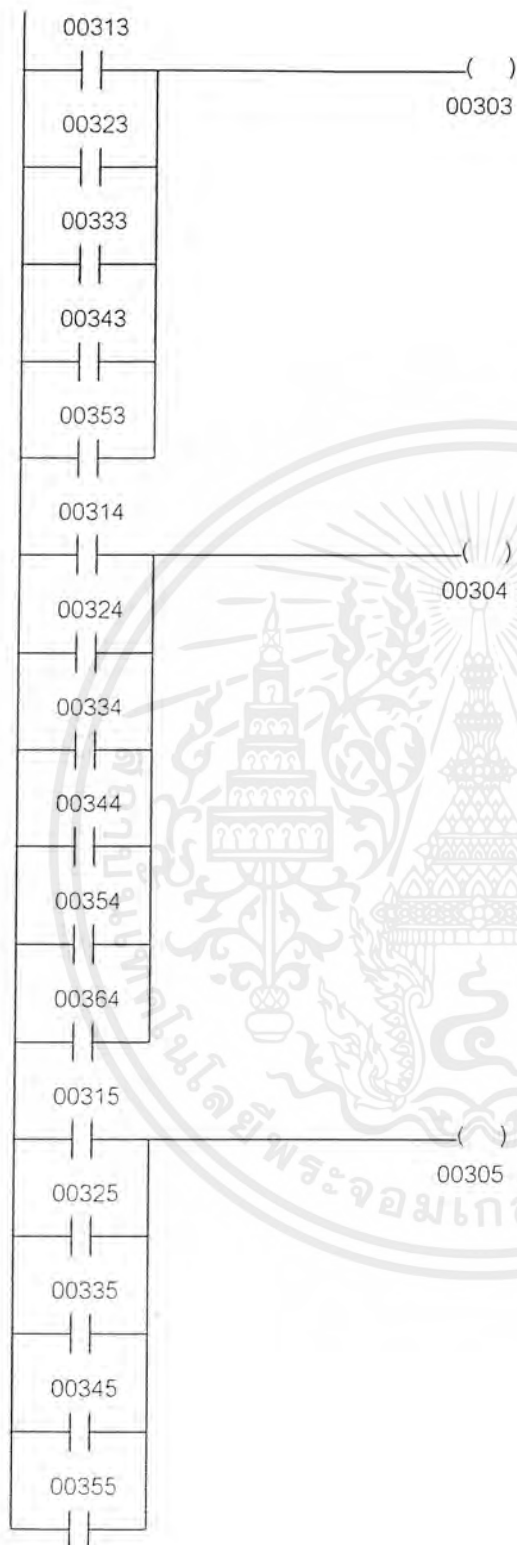
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



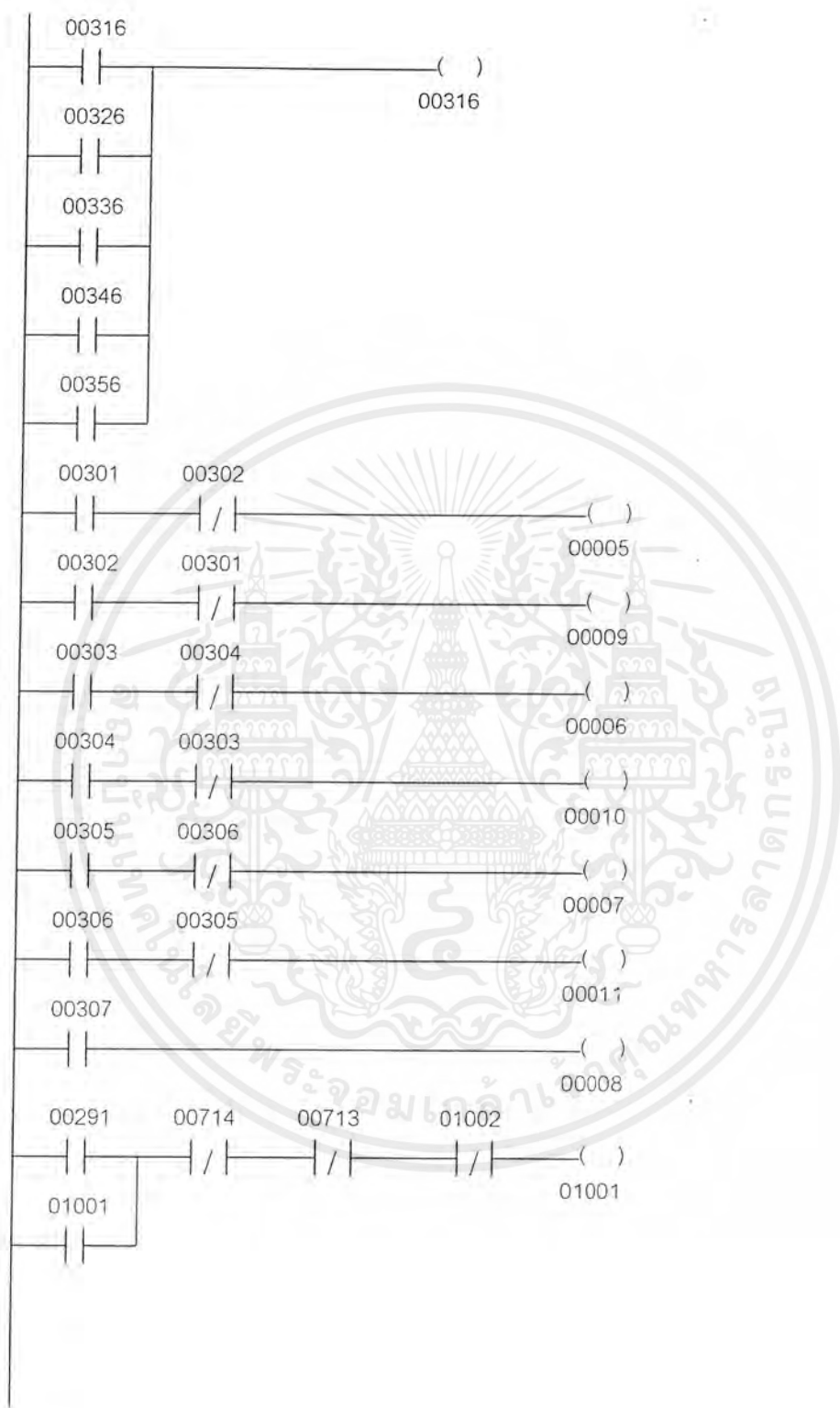
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



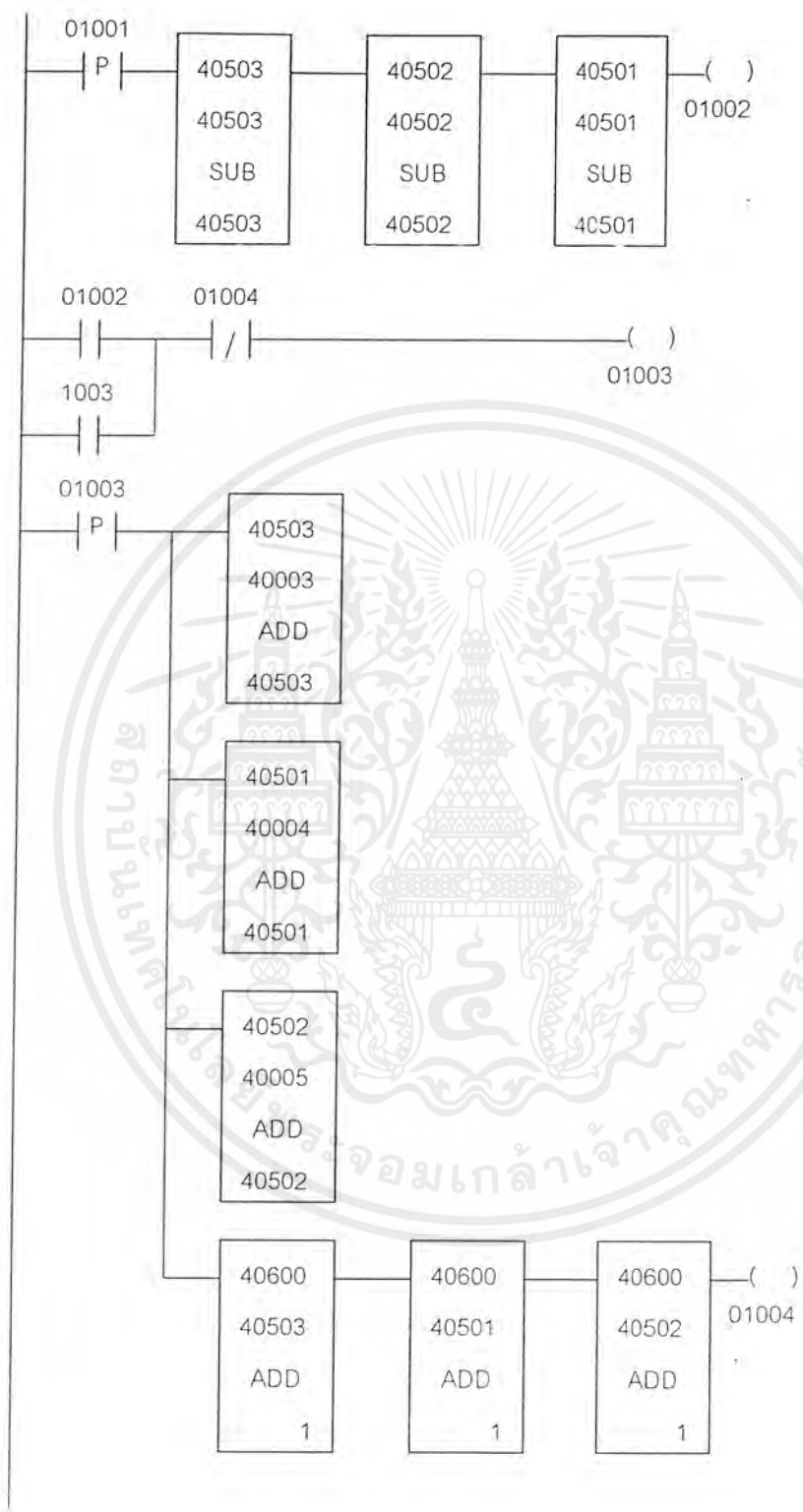
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



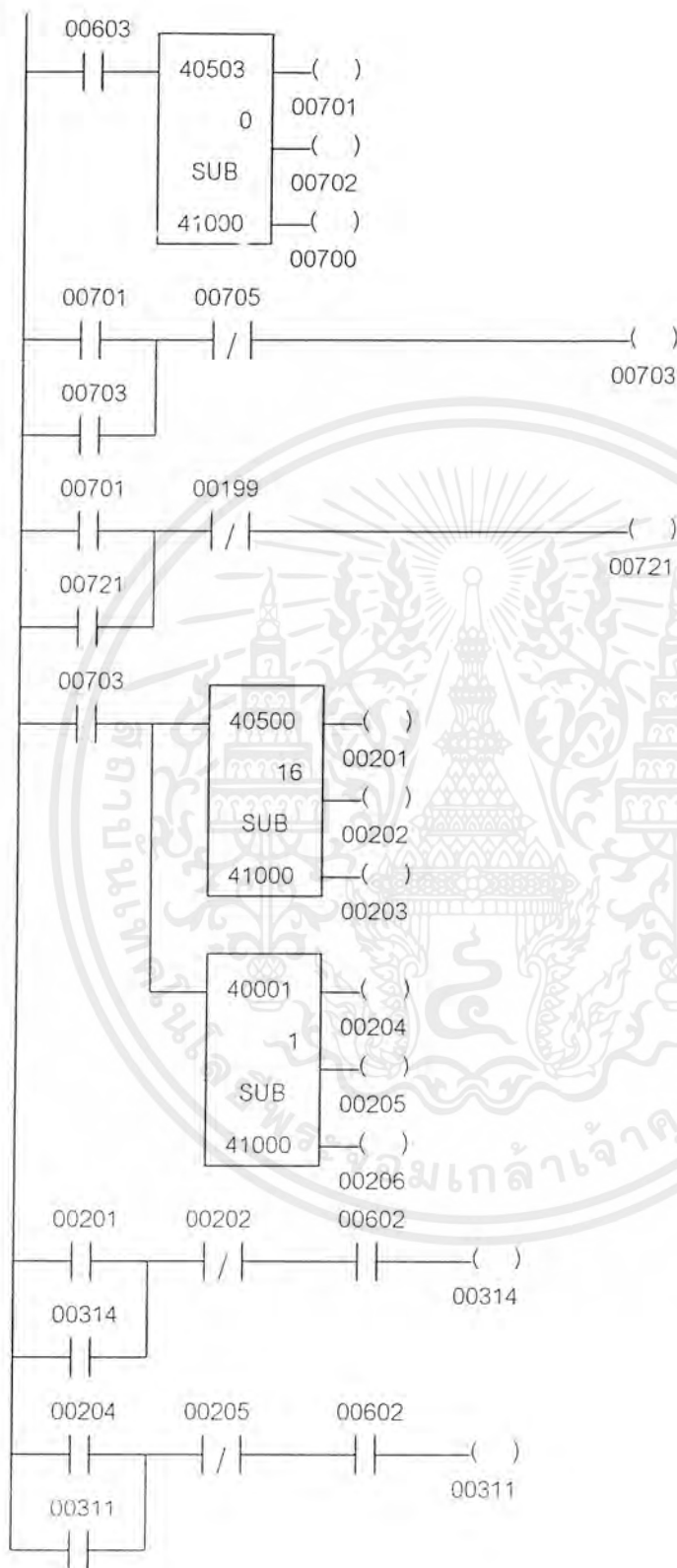
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



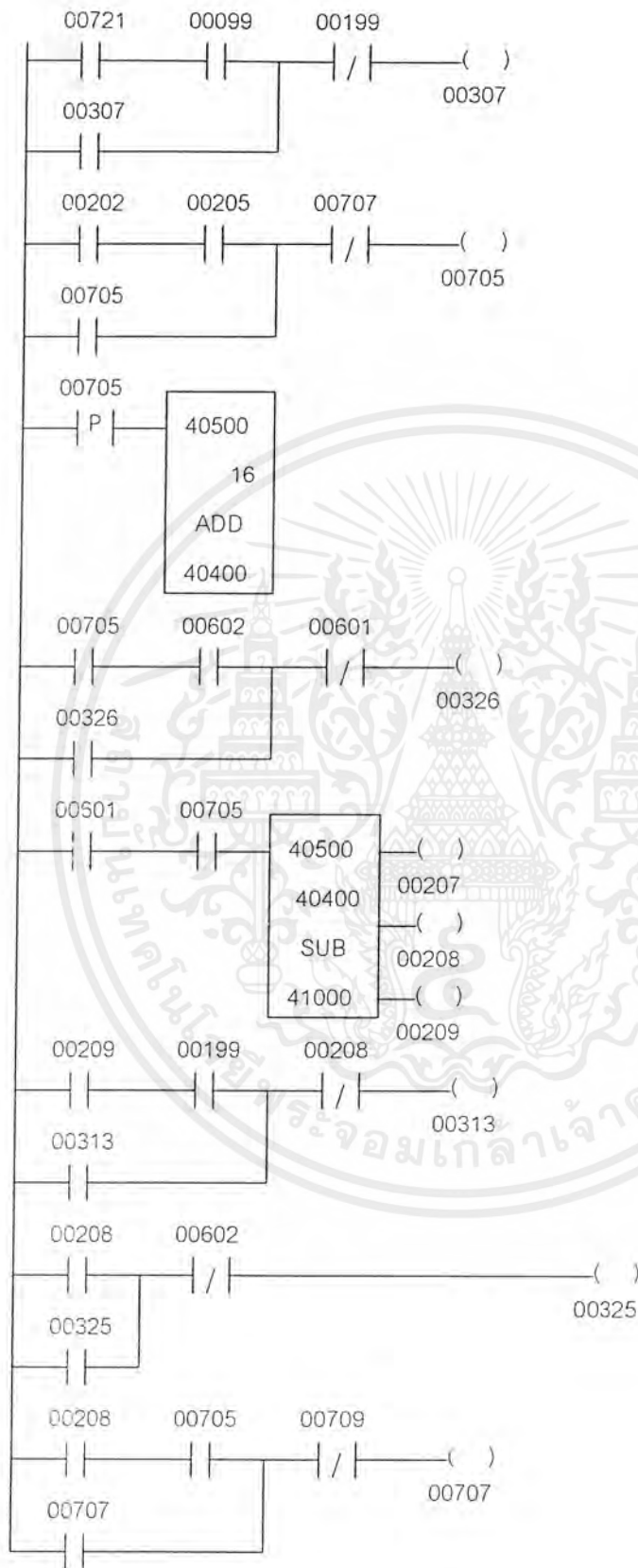
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



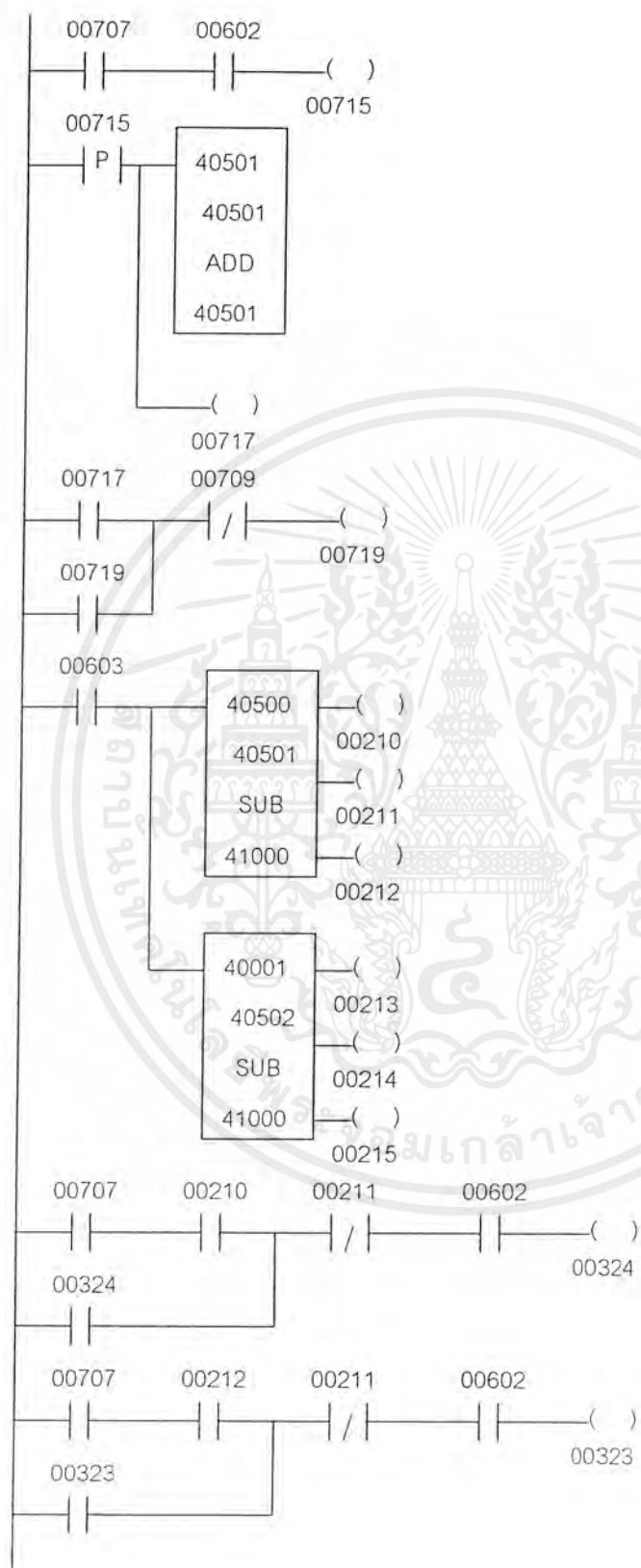
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



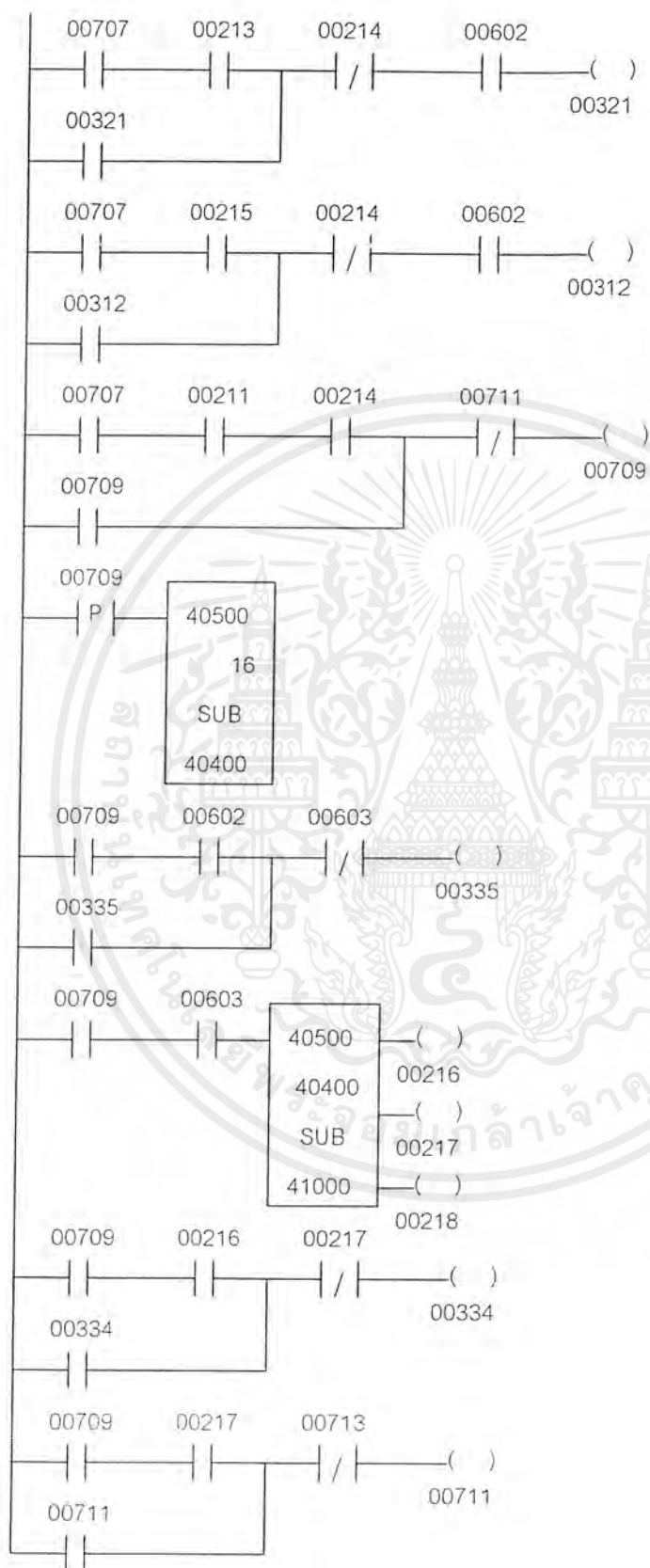
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



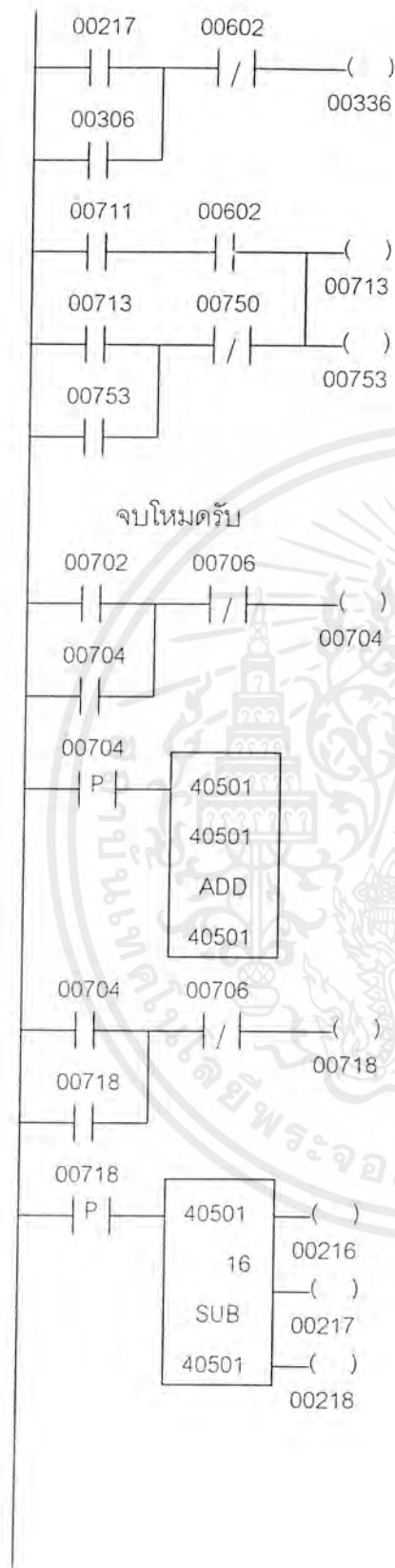
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



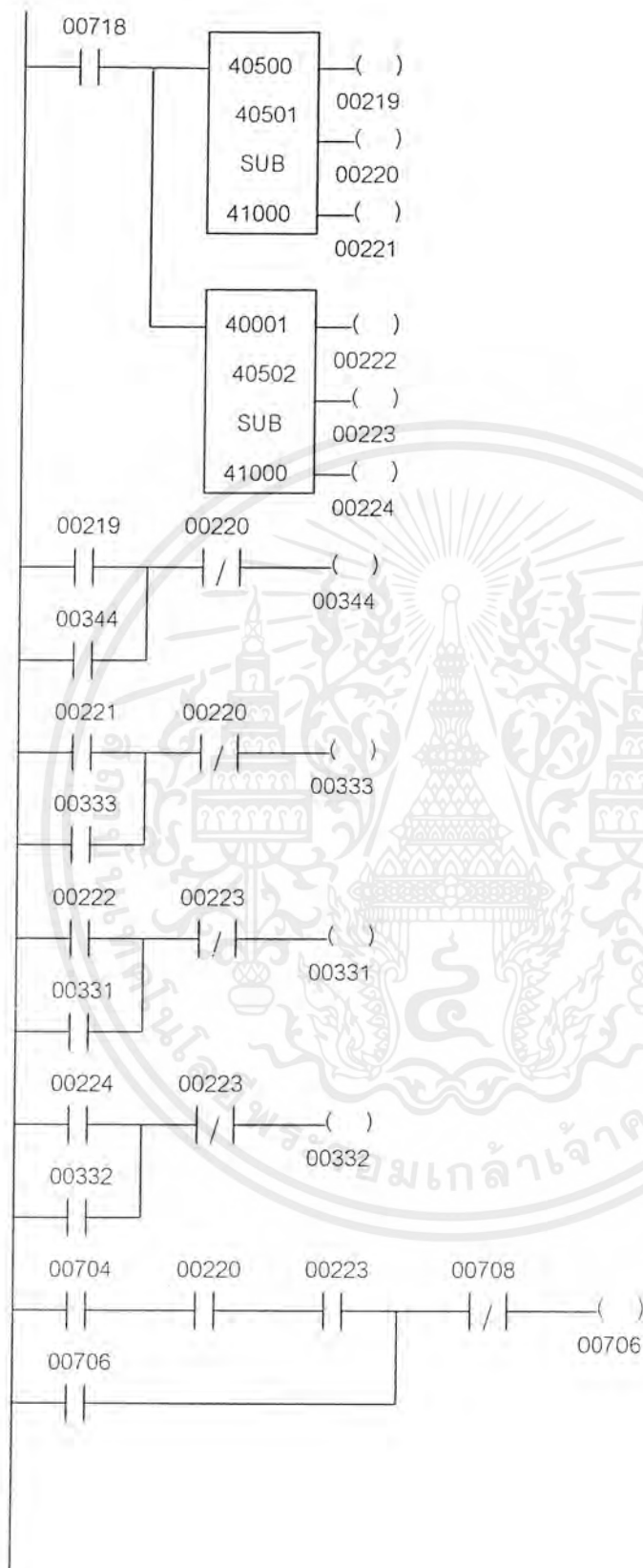
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



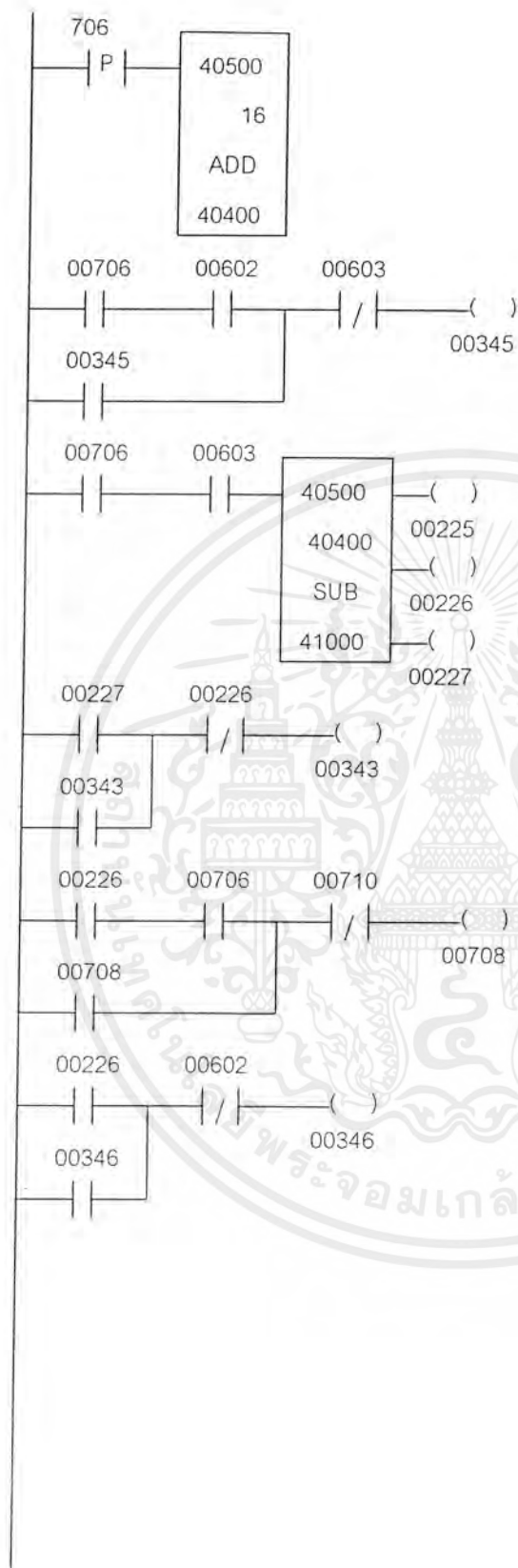
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



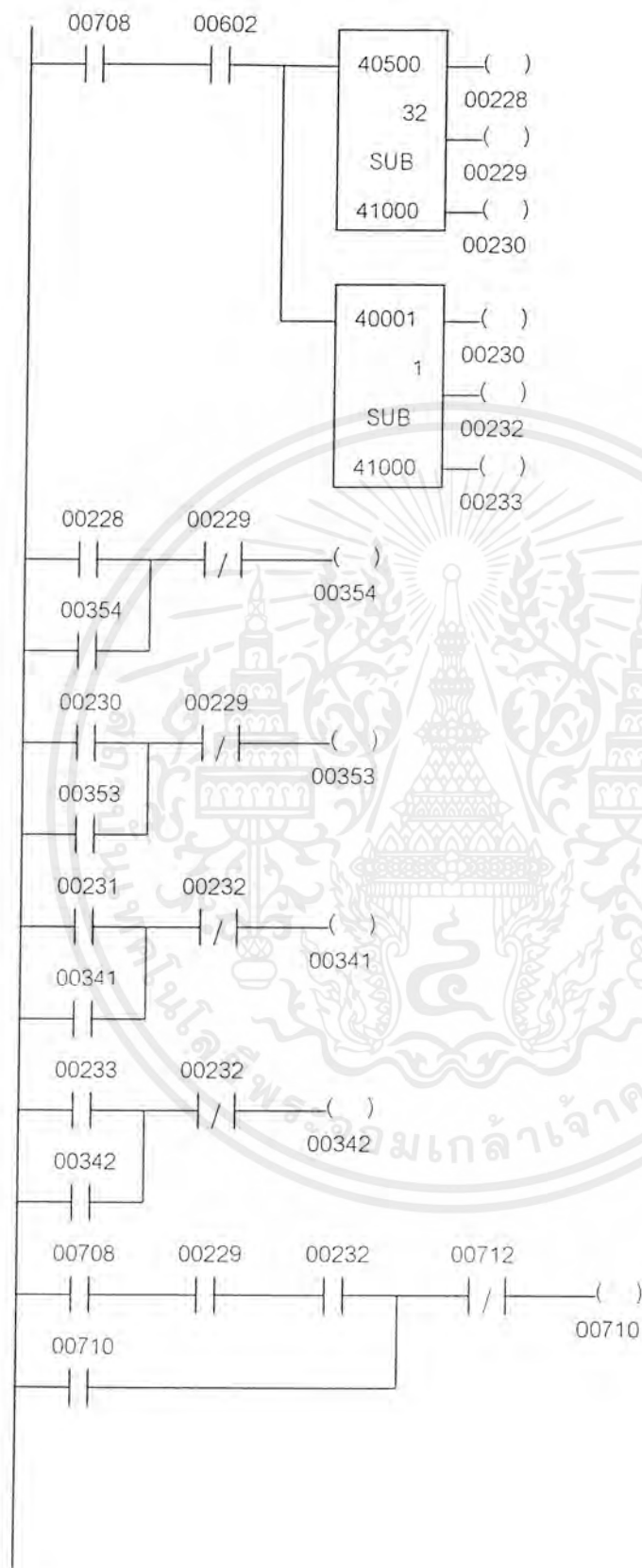
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

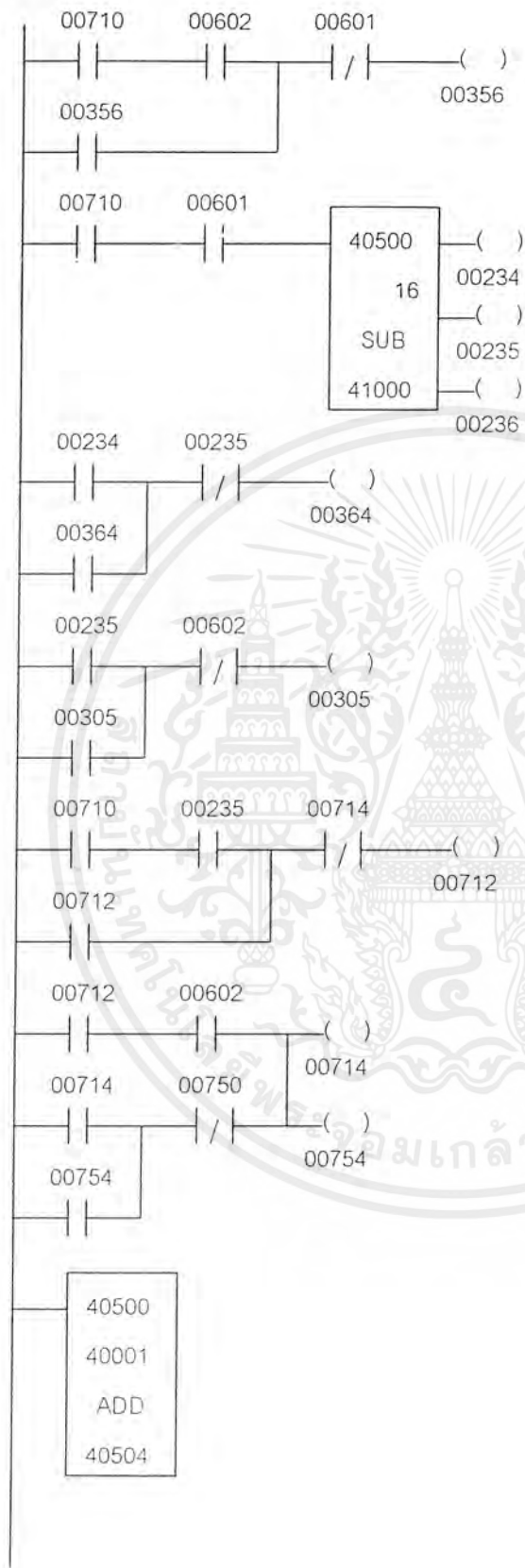


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้





เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก C

Microsoft Visual Basic



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

frmMainMenu

Private Sub ButtonExit_Click()

    Unload Me

End Sub

Private Sub ButtonPassword_Click()

    frmLogin.Show vbModal, Me

End Sub

Private Sub Cmdcheck_Click()

    frmCheck.Show vbModal, Me

End Sub

Private Sub Form_Load()

    CmdStore.Enabled = False

    CmdWithdraw.Enabled = False

    CmdUserdata.Enabled = False

End Sub

Private Sub Cmdstore_Click()

    frmstore.Show vbModal, Me

End Sub

Private Sub Cmduserdata_Click()

    frmUser.Show vbModal, Me

End Sub

Private Sub Cmdwithdraw_Click()

    frmwithdraw.Show vbModal, Me

End Sub

frmLogin

Option Explicit

Public UserID As Variant

Public LoginSucceeded As Boolean

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Private Sub cmdCancel_Click()
    LoginSucceeded = False
    frmMainMenu.Enabled = True
    Me.Hide
End Sub

Private Sub cmdOK_Click()
    Dim strSQL As Variant
    Dim UserName As String
    UserName = Trim(txtUserName.Text)
    If UserName <> "" Then
        strSQL = "select [Password] from [users] where [userID]=''" & UserName & "'"
        Data1.RecordSource = strSQL
        Data1.Refresh
        If Data1.Recordset.RecordCount = 1 Then
            strSQL = Data1.Recordset.Fields("password")
            If Trim(txtPassword.Text) = strSQL Then
                Unload Me
                frmMainMenu.CmdStore.Enabled = True
                frmMainMenu.CmdWithdraw.Enabled = True
                frmMainMenu.CmdUserdata.Enabled = True
                UserID = UserName
            Else: MsgBox "รหัสผ่านไม่ถูกต้อง", vbExclamation, "กรอกรหัสผ่าน"
            End If
        Else: MsgBox "ไม่พบชื่อของผู้ใช้ในรายการ", vbExclamation, "โปรดกรอกข้อมูลใหม่"
        End If
    End If
End Sub

Private Sub Form_Unload(Cancel As Integer)

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

frmMainMenu.Enabled = True
End Sub
Private Sub txtPassword_KeyPress(KeyAscii As Integer)
    If KeyAscii = 13 Then
        cmdOK.SetFocus
    End If
End Sub
Private Sub txtUserName_KeyPress(KeyAscii As Integer)
    If KeyAscii = 13 Then
        txtPassword.SetFocus
    End If
End Sub

frmStore
Private Sub Command1_Click()
    Dim strSQL As Variant
    Dim x,Y , LRC , keep As String
    Dim a ,ep As , ans , ke Variant
    Dim see As Integer
    Dim AllSum , onecom , twocom As Variant
    If (Text1.Text <> "") And (Text2.Text <> "") Then
        strSQL = "select * from [Products] where [ProductID]=''" & Text1.Text & "'"
        DataTableProduct.RecordSource = strSQL
        DataTableProduct.Refresh
        If DataTableProduct.Recordset.RecordCount <> 0 Then
            strSQL = " select * from [Remain] where [Remainder] =0"
            DataTableRemain.RecordSource = strSQL
            DataTableRemain.Refresh
        End If
    End If
End Sub

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

If DataTableRemain.Recordset.RecordCount <> 0 Then

 DataTableMain.Recordset.AddNew

 DataTableMain.Recordset.[ProductID] = Text1.Text

 DataTableMain.Recordset.[StoreDate] = Date

 DataTableMain.Recordset.[StoreTime] = Time

 strSQL = DataTableRemain.Recordset.Fields("Location")

 DataTableMain.Recordset.[Location] = strSQL

 DataTableMain.Recordset.[UserID] = frmLogin.UserID

 DataTableMain.Recordset.Update

 DataTableRemain.Recordset.MoveFirst

 DataTableRemain.Recordset.Edit

 DataTableRemain.Recordset.[Remainder] = Val(Text2.Text)

 DataTableRemain.Recordset.Update

 Select Case strSQL

 Case 1

 x = "1"

 Y = "1"

 Case 2

 x = "2"

 Y = "1"

 Case 3

 x = "3"

 Y = "1"

 Case 4

 x = "1"

 Y = "2"

 Case 5

 x = "2"

$$Y = "2"$$

Case 6

$$x = "3"$$

$$Y = "2"$$

Case 7

$$x = "1"$$

$$Y = "3"$$

Case 8

$$x = "2"$$

$$Y = "3"$$

Case 9

$$x = "3"$$

$$Y = "3"$$

Case 10

$$x = "1"$$

$$Y = "4"$$

Case 11

$$x = "2"$$

$$Y = "4"$$

Case 12

$$x = "3"$$

$$Y = "4"$$

Case 13

$$x = "1"$$

$$Y = "5"$$

Case 14

$$x = "2"$$

$$Y = "5"$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Case 15

x = "3"

Y = "5"

Case 16

x = "1"

Y = "6"

Case 17

x = "2"

Y = "6"

Case 18

x = "3"

Y = "6"

Case 19

x = "1"

Y = "7"

Case 20

x = "2"

Y = "7"

Case 21

x = "3"

Y = "7"

End Select

If MSComm1.PortOpen = False Then

MSComm1.PortOpen = True

End If

keep = "01" + x + Y

Text3.Text = keep

ke = Mid(keep, 3, 1)

```

ep = Mid(keep, 4, 1)
a = Val(ke)
ke = a * 16
a = Val(ep)
ep = a
see = 97 + ke + ep
oncom = 255 Xor see
twocom = oncom + 1
AllSum = Hex(twocom)
LRC = AllSum
a = ":01060257" + keep + LRC + Chr$(13) + Chr$(10)
MSComm1.Output = a
Text4.Text = a
Y = ":010503E6FF0012" + Chr$(13) + Chr$(10)
MSComm1.Output = Y
MSComm1.PortOpen = False
frmCheckLocation.Show vbModal, Me
Else: MsgBox "ไม่มีช่องว่างในการจัดเก็บ", 48, "โปรดตรวจสอบ"
End If
Else: MsgBox "ไม่พบรหัสสินค้าในรายการ", 48, "โปรดตรวจสอบ"
End If
Text1.Text = ""
Text2.Text = ""
Text1.SetFocus
Else
MsgBox "ยังกรอกข้อมูลไม่ครบถ้วน", vbExclamation, "โปรดกรอกข้อมูล "
End If
End Sub

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Private Sub Command2_Click()
    Unload Me
    frmMainiMenu.Enabled = True
End Sub

Private Sub MSComm1_OnComm()
    Dim s As String
    If MSComm1.PortOpen = False Then
        MSComm1.PortOpen = True
    Select Case MSComm1.CommEvent
    Case comEvReceive
        s = MSComm1.Input
        Text5.Text = Text5.Text & s
    End Select
    End Sub

Private Sub Text1_KeyPress(KeyAscii As Integer)
    If KeyAscii = 13 Then
        Text2.SetFocus
    End If
End Sub

Private Sub Text2_KeyPress(KeyAscii As Integer)
    If KeyAscii = 13 Then
        Command1.SetFocus
    End If
End Sub

```

frmWithdraw

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Private Sub Command1_Click()
Dim strSQL, tmp, X, time1, day As Variant
if Text1.Text <> "" And Text2.Text <> "" And Text3.Text <> "" Then
    strSQL = " select Min(Main.[StoreTime])As MinOfStoreTime,Remain.[Location] from
[Remain]Inner Join [Main] on Remain.[Location] = Main.[Location]"
    strSQL = strSQL & "Where Remain.[Remainder] <> 0 and Main.[StoreDate] =
( select Min(StoreDate) from [Main] where [ProductID] = """" & Text2.Text & """" and UserID2 is
null ) "
    strSQL = strSQL & "and Main.ProductID = """" & Text2.Text & """" and UserID2 is null Group
by Remain.[Location] order by 1 "
    Data3.RecordSource = strSQL
    Data3.Refresh
    If Data3.Recordset.RecordCount <> 0 Then
        time1 = Data3.Recordset.Fields("MinOfStoreTime")
        X = Data3.Recordset.Fields("Location")
        strSQL = "select * from Main "
        strSQL =strSQL&"where[ProductID]='"&Text2.Text&"'and [Location] = " & X & " "
        strSQL = strSQL & "and [StoreTime]= " & "#" & time1 & "#" & "and[UserID2] is null"
        Data1.RecordSource = strSQL
        Data1.Refresh
        Data1.Recordset.MoveFirst
        Data1.Recordset.Edit
        Data1.Recordset.[WithdrawDate] = Date
        Data1.Recordset.[WithdrawTime] = Time
        Data1.Recordset.[UserID2] = frmLogin.UserID
        Data1.Recordset.Update
        Data3.Recordset.MoveFirst
        Data3.Recordset.Edit

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Data3.Recordset.[Remainder] = 0

Data3.Recordset.Update

Dim Y , keep , LRC As String

Dim a , ep As , ans, ke As Variant

Dim see As Integer

Dim AllSum , onecom , twocom As Variant

Select Case strSQL

Case 1

x = "1"

Y = "1"

Case 2

x = "2"

Y = "1"

Case 3

x = "3"

Y = "1"

Case 4

x = "1"

Y = "2"

Case 5

x = "2"

Y = "2"

Case 6

x = "3"

Y = "2"

Case 7

x = "1"

Y = "3"

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Case 8

$x = "2"$

$Y = "3"$

Case 9

$x = "3"$

$Y = "3"$

Case 10

$x = "1"$

$Y = "4"$

Case 11

$x = "2"$

$Y = "4"$

Case 12

$x = "3"$

$Y = "4"$

Case 13

$x = "1"$

$Y = "5"$

Case 14

$x = "2"$

$Y = "5"$

Case 15

$x = "3"$

$Y = "5"$

Case 16

$x = "1"$

$Y = "6"$

Case 17

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

x = "2"

Y = "6"

Case 18

x = "3"

Y = "6"

Case 19

x = "1"

Y = "7"

Case 20

x = "2"

Y = "7"

Case 21

x = "3"

Y = "7"

End Select

If MSComm1.PortOpen = False Then

MSComm1.PortOpen = True

End If

keep = "00" + x + Y

ke = Mid(keep, 3, 1)

ep = Mid(keep, 4, 1)

a = Val(ke)

ke = a * 16

a = Val(ep)

ep = a

see = 96 + ke + ep

onecom = 255 Xor see

twocom = onecom + 1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

AllSum = Hex(twocom)
LRC = AllSum
Dim i As Integer
For i = 1 To 5
a = ":01060257" + keep + LRC + Chr$(13) + Chr$(10)
MSComm1.Output = a
Next i
MSComm1.PortOpen = False
Text1.Text = ""
Text2.Text = ""
Text3.Text = ""
End If
Else: MsgBox "คงเหลือพัสดุภัณฑ์ไม่เพียงพอในการนำออก", vbExclamation, "นำออกพัสดุ
ภัณฑ์ไม่ได้"
End If
Else
MsgBox "ข้อมูลยังกรอกไม่ครบถ้วน", 48, "กรอกข้อมูล"
End If
End Sub
Private Sub Command2_Click()
Unload Me
frmMainMenu.Enabled = True
End Sub
Private Sub Text1_KeyPress(KeyAscii As Integer)
If KeyAscii = 13 Then
Command1.SetFocus
End If
End Sub

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Private Sub Text2_KeyPress(KeyAscii As Integer)
    If KeyAscii = 13 Then
        Text3.SetFocus
    End If
End Sub

Private Sub Text2_LostFocus()
Dim strSQL As Variant

    strSQL = "select [ProductID] from [Products] where [ProductID] ="" & Text2.Text & """"
    Data2.RecordSource = strSQL
    Data2.Refresh

    If Data2.Recordset.RecordCount <> 0 Then
        strSQL = "select [ProductName],[ProductID] from [Products] where [ProductID]= "" &
Text2.Text & """"
        Data2.RecordSource = strSQL
        Data2.Refresh
        strSQL = Data2.Recordset.Fields("ProductName")
        Text2.Text = Data2.Recordset.Fields("ProductID")
        Text3.Text = strSQL
    Else: MsgBox " ไม่มีรหัสสินค้าที่ท่านกรอกในรายการ", vbExclamation, " โปรดตรวจสอบ"
    End If
End Sub

Private Sub Text3_KeyPress(KeyAscii As Integer)
    If KeyAscii = 13 Then
        Text1.SetFocus
    End If
End Sub

frmCheck

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Private Sub Command22_Click()

    Unload Me

End Sub

Private Sub Form_Load()

Dim strSQL As Variant

    For i = 1 To ID.Count

        strSQL = "select * from [Main] where UserID2 is Null"

        Data1.RecordSource = strSQL

        Data1.Refresh

        strSQL = "Location = " + CStr(i) 'if no recordset

        Data1.Recordset.FindLast (strSQL)

        If Data1.Recordset.NoMatch Then

            ID(i - 1).Text = ""

            Remain(i - 1).Text = "0"

        Else: strSQL = "select * from [Remain] where [Location]= " + CStr(i)

            Data3.RecordSource = strSQL

            Data3.Refresh

            strSQL = Data3.Recordset.Fields("Remainder")

            If strSQL <> 0 Then

                strSQL = "select * from [Main] where [Location]= " + CStr(i) + " and [UserID2] is null "

                Data1.RecordSource = strSQL

                Data1.Refresh

                ID(i - 1).Text = Data1.Recordset.Fields("ProductID")

                Remain(i - 1).Text = Data3.Recordset.Fields("Remainder")

            End If

        End If

    Next i

End Sub

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Private Sub Command22_Click()
    Unload Me
End Sub

Private Sub Option1_Click()
    frmDataInOut.Show vbModal, Me
End Sub

Private Sub Option1_Click()
    frmDataInOut.Show vbModal, Me
End Sub

Private Sub Option2_Click()
    frmProduct.Show vbModal, Me
End Sub

frmUser
Private Sub Command3_Click()
    Unload Me
    frmMainMenu.Show
End Sub

Private Sub Option1_Click()
    frmplus.Show vbModal, Me
End Sub

Private Sub Option2_Click()
    frmDel.Show vbModal, Me
End Sub

```

frmPlus

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Private Sub Command1_Click()
    If Text1.Text <> "" And Text2.Text <> "" And Text3.Text <> "" Then
        Data1.Recordset.AddNew
        Data1.Recordset.[UserID] = Text1.Text
        Data1.Recordset.[UserName] = Text2.Text
        Data1.Recordset.[Password] = Text3.Text
        Data1.Recordset.Update
        Text1.Text = ""
        Text2.Text = ""
        Text3.Text = ""
    Else
        MsgBox "ข้อมูลยังกรอกไม่ครบถ้วน", 48, "กรอกข้อมูล"
    End If
End Sub

Private Sub Command2_Click()
    Unload Me
    frmUser.Enabled = True
End Sub

Private Sub Text1_KeyPress(KeyAscii As Integer)
    If KeyAscii = 13 Then
        Text2.SetFocus
    End If
End Sub

Private Sub Text2_KeyPress(KeyAscii As Integer)
    If KeyAscii = 13 Then
        Text3.SetFocus
    End If
End Sub

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Private Sub Text3_KeyPress(KeyAscii As Integer)
```

```
    If KeyAscii = 13 Then
```

```
        Command1.SetFocus
```

```
    End If
```

```
End Sub
```

```
frmDel
```

```
Private Sub Command1_Click()
```

```
    Dim strSQL As String
```

```
    If Text1.Text <> "" And Text2.Text <> "" And Text3.Text <> "" Then
```

```
        strSQL = " select [UserID] from [Users] where [UserID]=''" & Text1.Text & "'" & ""
```

```
        Data1.RecordSource = strSQL
```

```
        Data1.Refresh
```

```
        If Data1.Recordset.RecordCount = 1 Then
```

```
            Data1.Recordset.MoveFirst
```

```
            Data1.Recordset.Delete
```

```
            Data1.Refresh
```

```
        Else: MsgBox "ไม่พบข้อมูลของผู้ใช้ในรายการ", vbExclamation, "โปรดกรอกข้อมูลอีกครั้ง"
```

```
        End If
```

```
        Text1.Text = ""
```

```
        Text2.Text = ""
```

```
        Text3.Text = ""
```

```
    Else
```

```
        MsgBox "ยังกรอกข้อมูลไม่ครบถ้วน", 48, "กรอกข้อมูล"
```

```
    End If
```

```
End Sub
```

```
Private Sub Command2_Click()
```

```
    Unload Me
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

frmUser.Enabled = True
End Sub
Private Sub Text1_KeyPress(KeyAscii As Integer)
    If KeyAscii = 13 Then
        Text2.SetFocus
    End If
End Sub
Private Sub Text1_LostFocus()
    Dim sqlcmd As String
    Dim strSQL As String
    strSQL = "UserID = " & Text1.Text & ""
    Data1.Recordset.FindFirst (strSQL)
    If Data1.Recordset.NoMatch Then
        MsgBox "ไม่พบข้อมูลของผู้ใช้ในรายการ", vbExclamation, "กรอกข้อมูลใหม่"
    ElseIf Text2.Text = "" And Text3.Text = "" Then
        sqlcmd = "select [UserName],[Password] from [Users]
            where [UserID]='" & Text1.Text & "'"
        Data1.RecordSource = sqlcmd
        Data1.Refresh
        Data1.Recordset.MoveFirst
        sqlcmd = Data1.Recordset.Fields("UserName")
        strSQL = Data1.Recordset.Fields("Password")
        Text2.Text = sqlcmd
        Text3.Text = strSQL
    End If
End Sub
Private Sub Text2_KeyPress(KeyAscii As Integer)
    If KeyAscii = 13 Then

```

```

        Text3.SetFocus
    End If
End Sub

Private Sub Text2_LostFocus()
    Dim sqlcmd As String
    Dim strSQL As String
    If Text1.Text = "" And Text3.Text = "" Then
        sqlcmd = "select [UserID],[Password] from [Users]
                where [UserName]=''" & Text2.Text & "'"
        Data1.RecordSource = sqlcmd
        Data1.Refresh
        Data1.Recordset.MoveFirst
        sqlcmd = Data1.Recordset.Fields("UserID")
        strSQL = Data1.Recordset.Fields("Password")
        Text1.Text = sqlcmd
        Text3.Text = strSQL
    End If
End Sub

Private Sub Text3_KeyPress(KeyAscii As Integer)
    If KeyAscii = 13 Then
        Command1.SetFocus
    End If
End Sub

Private Sub Text3_LostFocus()
    Dim sqlcmd As String
    Dim strSQL As String
    If Text1.Text = "" And Text2.Text = "" Then
        sqlcmd = "select [UserID],[UserName] from [Users]"

```

```

        where [Password]="'" & Text3.Text & "'"

Data1.RecordSource = sqlcmd
Data1.Refresh
Data1.Recordset.MoveFirst
sqlcmd = Data1.Recordset.Fields("UserID")
strSQL = Data1.Recordset.Fields("UserName")
Text1.Text = sqlcmd
Text2.Text = strSQL
End If
End Sub

frmchecklocation
Dim send As Variant
Dim decdat, onecom, twocom, decsend, hexsend As Variant
Sub Delay()
    Dim i, j As Integer
    For i = 0 To 25
        For j = 0 To 30000
            Next j
        Next i
    End Sub
Sub LEDsend()
    Dim Led As Variant
    Dim block As String
    Dim i As Integer
    For i = 1 To 7
        Led = DataTableRemain.Recordset.Fields("Remainder")
        block = block & Led
    
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    DataTableRemain.Recordset.MoveNext
Next i

    decsend = Mid(block, 1, 1) + Mid(block, 2, 1) * 2 + Mid(block, 3, 1) * 4 + Mid(block, 4, 1) * 8
        + Mid(block, 5, 1) * 16 + Mid(block, 6, 1) * 32 + Mid(block, 7, 1) * 64

    hexsend = Hex(decsend)
End Sub

Private Sub Command1_Click()
    Dim buffer As String
    buffer = ":010503E6000011" + Chr$(13) + Chr$(10)
    MSComm1.Output = buffer
    Delay
    Text22.Text = ""
    MSComm1.PortOpen = False
    Unload Me
    frmstore.Enabled = True
End Sub

Private Sub Command2_Click()
    Dim Scoil, Y As String
    Dim i, n As Integer
    Timer1.Enabled = True
End Sub

Private Sub Form_Load()
    Timer1.Enabled = False
    Command1.Enabled = False
    If MSComm1.PortOpen = False Then
        MSComm1.PortOpen = True
    End If
    Text23.Text = frmstore.cell

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

End Sub

Private Sub MSComm1_OnComm()

Timer1.Enabled = False

Dim i As Integer

Dim a, ab, LRC, strSQL As String

Text1.BackColor = &H80000005

Text2.BackColor = &H80000005

Text3.BackColor = &H80000005

Text4.BackColor = &H80000005

Text5.BackColor = &H80000005

Text6.BackColor = &H80000005

Text7.BackColor = &H80000005

Text8.BackColor = &H80000005

Text9.BackColor = &H80000005

Text10.BackColor = &H80000005

Text11.BackColor = &H80000005

Text12.BackColor = &H80000005

Text13.BackColor = &H80000005

Text14.BackColor = &H80000005

Text15.BackColor = &H80000005

Text16.BackColor = &H80000005

Text17.BackColor = &H80000005

Text18.BackColor = &H80000005

Text19.BackColor = &H80000005

Text20.BackColor = &H80000005

Text21.BackColor = &H80000005

If MSComm1.CommEvent = 2 Then

MSComm1.RThreshold = 15

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

For i = 1 To MSComm1.InBufferCount
a = MSComm1.Input
Text22.Text = Text22.Text & a
Next i
ab = Mid(Text22.Text, 5, 5)
Select Case ab
    Case "10100"
        send = ":010301F7000103" + Chr$(13) + Chr$(10)
        MSComm1.Output = send
        Text22.Text = ""
    Case "10101"
        MSComm1.RThreshold = 0
        send = ":010503E6000011" + Chr$(13) + Chr$(10)
        MSComm1.Output = send
        strSQL = "select [Remainder] from [Remain]"
        DataTableRemain.RecordSource = strSQL
        DataTableRemain.Refresh
        LEDsend
        decdat = 97 + decsend
        onecom = 255 Xor decdat
        twocom = onecom + 1
        LRC = Hex(twocom)
        send = ":0106025800" + hexsend + LRC + Chr$(13) + Chr$(10)
        MSComm1.Output = send
        frmwithdraw.Delay
        LEDsend
        decdat = 98 + decsend
        onecom = 255 Xor decdat

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

twocom = onecom + 1
LRC = Hex(twocom)
send = ":0106025900" + hexsend + LRC + Chr$(13) + Chr$(10)
MSComm1.Output = send
frmwithdraw.Delay
LEDsend
decdat = 99 + decsend
onecom = 255 Xor decdat
twocom = onecom + 1
LRC = Hex(twocom)
send = ":0106025A00" + hexsend + LRC + Chr$(13) + Chr$(10)
MSComm1.Output = send
frmwithdraw.Delay
Command1.Enabled = True
Text22.Text = ""
End Select
ab = Mid(Text22.Text, 5, 1)
If ab = "3" Then
ab = Mid(Text22.Text, 10, 2)
Select Case ab
    Case "11"
        Text1.BackColor = &HFF0000
    Case "21"
        Text1.BackColor = &HFF0000
    Case "31"
        Text2.BackColor = &HFF0000
    Case "41"
        Text2.BackColor = &HFF0000

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Case "51"

Text3.BackColor = &HFF0000

Case "61"

Text3.BackColor = &HFF0000

Case "12"

Text4.BackColor = &HFF0000

Case "22"

Text4.BackColor = &HFF0000

Case "32"

Text5.BackColor = &HFF0000

Case "42"

Text5.BackColor = &HFF0000

Case "52"

Text6.BackColor = &HFF0000

Case "62"

Text6.BackColor = &HFF0000

Case "13"

Text7.BackColor = &HFF0000

Case "23"

Text7.BackColor = &HFF0000

Case "33"

Text8.BackColor = &HFF0000

Case "43"

Text8.BackColor = &HFF0000

Case "53"

Text9.BackColor = &HFF0000

Case "63"

Text9.BackColor = &HFF0000

Case "14"

Text10.BackColor = &HFF0000

Case "24"

Text10.BackColor = &HFF0000

Case "34"

Text11.BackColor = &HFF0000

Case "44"

Text11.BackColor = &HFF0000

Case "54"

Text12.BackColor = &HFF0000

Case "64"

Text12.BackColor = &HFF0000

Case "15"

Text13.BackColor = &HFF0000

Case "25"

Text13.BackColor = &HFF0000

Case "35"

Text14.BackColor = &HFF0000

Case "45"

Text14.BackColor = &HFF0000

Case "55"

Text15.BackColor = &HFF0000

Case "65"

Text15.BackColor = &HFF0000

Case "16"

Text16.BackColor = &HFF0000

Case "26"

Text16.BackColor = &HFF0000

```
Case "36"
Text17.BackColor = &HFF0000
Case "46"
Text17.BackColor = &HFF0000
Case "56"
Text18.BackColor = &HFF0000
Case "66"
Text18.BackColor = &HFF0000
Case "17"
Text19.BackColor = &HFF0000
Case "27"
Text19.BackColor = &HFF0000
Case "37"
Text20.BackColor = &HFF0000
Case "47"
Text20.BackColor = &HFF0000
Case "57"
Text21.BackColor = &HFF0000
Case "67"
Text21.BackColor = &HFF0000
End Select

Timer1.Enabled = True
End If
End If
End Sub
Private Sub Timer1_Timer()
MSComm1.RThreshold = 12
send = ":010102F000010B" + Chr$(13) + Chr$(10)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
MSComm1.Output = send
```

```
Text22.Text = ""
```

```
End Sub
```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

ปริญญานิพนธ์ฉบับนี้คงไม่สามารถสำเร็จลุล่วงไปได้ ถ้าไม่ได้รับความช่วยเหลือจาก อาจารย์เชื้อ นกอยู่ อาจารย์ทวีพล ชื้อสัตว์ และอาจารย์ท่านอื่นๆ ที่คอยให้คำปรึกษาและช่วยแก้ไข ปัญหาต่างๆ ด้วยดีตลอดมา

ขอขอบคุณ พี่เอ็กซ์ พี่ก้อย คิม เบนซ์และเพื่อนๆภาควิชาเทคโนโลยีการวัดคุมทางอุตสาหกรรม ที่คอยให้ความช่วยเหลือและแก้ไขปัญหาดังๆ

สุดท้ายขอขอบคุณ ภาควิชาเทคโนโลยีการวัดคุมทางอุตสาหกรรม สถาบันเทคโนโลยี พระจอมเกล้าเจ้าคุณทหารลาดกระบัง ที่ได้เอื้อเฟื้อในเรื่องเครื่องมือและอุปกรณ์ต่างๆ ในการทำ โครงการนี้ให้สำเร็จลุล่วงไปด้วยดี



บรรณานุกรม

1. ราเบนเดอร์ ศรีกิจจาภรณ์, “คู่มือการใช้งาน Visual Basic สำหรับวินโดวส์”, บริษัทซีเอ็ดยูเคชั่น, 407 หน้า, 2538
2. เอกวิทย์ จิตรดา, “MODBUS PROTOCOL”, เทคนิค165, หน้า 65-68, 2538
3. ชาริน สติธรรมชาวี, “Microsoft Visual Basic Version 5.0”, บริษัท ส.เอเชียเพลส (1989) จำกัด, 2539
4. รศ.พิภพ ลลิตาภรณ์, “การบริหารคลังระบบ MRP”, สมาคมส่งเสริมเทคโนโลยี (ไทย-ญี่ปุ่น), 217 หน้า, 2541
5. อนุชา ลือสกล, “Microsoft Access ถามตอบ”, แมคกรอ-ฮิล อินเทอร์เน็ต เนชั่นแนล เอ็นเตอร์ไพรส์, อิงค์, 311 หน้า, 2538
6. ชาริน สติธรรมชาวี, “คู่มือการเขียนโปรแกรม Microsoft Visual Basic Version 6.0”, บริษัท ชัคเชส มีเดีย จำกัด, 388 หน้า
7. G.M. NYSSSEN & E.D. FALKENBERG, “INTRODUCTION TO IBM SQL”, Nyssen Data Bases Pty.Ltd., 1984, 198 หน้า