ปีการศึกษา 2533

Programmable logic controller

โดย

นายวิทยา ชีสกุล 32.6325

นายสมบุญ เลื่อนตามผล 32.6328

อาจารย์ที่ปรึกษา

อาจารย์ภากร หุตะสังกาศ

027871

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

12 ก.ค. 2534

ปริญญานิพนธ์ปีการศึกษา 2533

เรื่อง PROGRAMMABLE LOGIC CONTROLLER

ผู้จัดทำ

1. นายวิทยา ชีสกุล 32.6325

2. นายสมบุญ เลื่อนตามผล 32.6328

อาจารย์ภากร หุตะสังกาศ อาจารย์ที่ปรึกษา

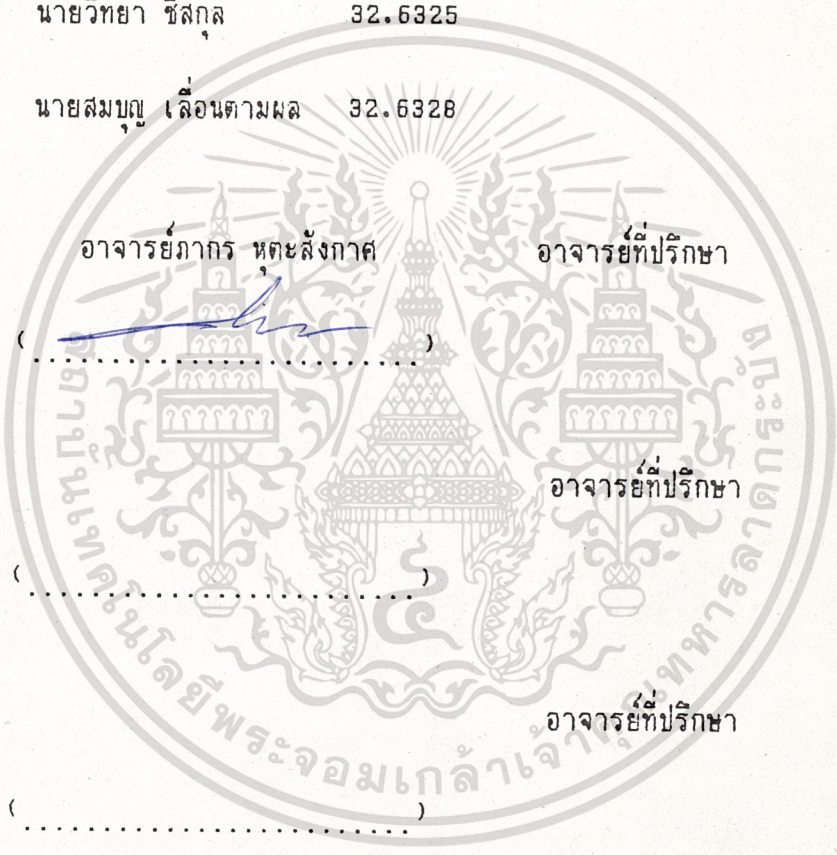
(.....)

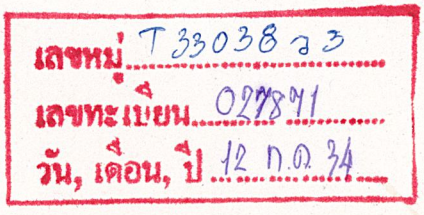
อาจารย์ที่ปรึกษา

(.....)

อาจารย์ที่ปรึกษา

(.....)





เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

027871

PROGRAMMABLE LOGIC CONTROLLER

ชื่อนักศึกษา

นายวิทยา ชีสกุล 32.6325
นายสมบุญ เลื่อนตามผล 32.6328

ชื่ออาจารย์ที่ปรึกษา

อาจารย์ภากร หุตะสังกาศ

ปีการศึกษา 2533

บทคัดย่อ

โครงงานนี้เป็นการสร้าง PLC ขนาดเล็กมีส่วนประกอบคือ Microcomputer PC,บอร์ดควบคุม, หน่วยอินพุตและเอาต์พุต บอร์ดควบคุมติดต่อกับ Microcomputer ได้โดยการส่งข้อมูลแบบอนุกรมผ่านทาง Serial Port(RS232) ของเครื่อง Computer การส่งสามารถเลือกอัตราการส่งข้อมูลได้เช่น 1200, 4800, 9600 เป็นต้น มีการทำงานคือ ป้อนคำสั่งเป็นรูปกราฟฟิกของภาษาแลดเดอร์ลงใน Editor ของตัว Complier ที่ทำงานอยู่บนเครื่อง Microcomputer (PC) แล้วเปลี่ยนจากคำสั่งรูปกราฟฟิกภาษาแลดเดอร์นั้นให้เป็นภาษาแอสแซมบลี้ของ CPU เบอร์ 8051 และเป็น Intel Hex Format ตามลำดับ บอร์ดควบคุมติดต่อกับหน่วยอินพุตและเอาต์พุตโดยการส่งข้อมูลแบบขนาน ซึ่งหน่วยอินพุตและเอาต์พุตมีอย่างละ 8 ตำแหน่ง แต่ละตำแหน่งขยายได้เป็น 64 ตำแหน่ง เราจะส่งคำสั่งไปให้บอร์ดควบคุมทำการประมวลผลตามคำสั่งที่ส่งมา โดยรับเงื่อนไขจากหน่วยอินพุต และส่งสภาวะที่ประมวลผลได้ไปยังหน่วยเอาต์พุตเพื่อขับโหลดต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

PROGRAMMABLE LOGIC CONTROLLER

STUDENT

WITTAYA CHEESAKUL 32.6325

SOMBUN LUANTAMPOL 32.6328

ADVISOR

PHAKORN HUTASANGKAS

Academic Year 1990

Abstract

In this thesis a small PLC was constructed which consist of personal microcomputer, main control board CPU 8051, I/O unit, interface board which serial communication with another computer via RS232 in bit rate 1200,4800,9600. A complier program which can complie graphic ladder program in to 8051 assembly language in intel format have been designed. Parallel communication are used beteween 8051 and I/O board this has 8 input and 8 output was constructed, an I/O can extention maximum with 64 unit.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบาญ

		หน้าที่
บทคัดย่อ		
บทที่ 1	บทนำ	1
บทที่ 2	รายละเอียดของ Programmable Logic Controller	2
บทที่ 3	การออกแบบและการสร้าง	23
บทที่ 4	การทดลองและผลการทดลอง	36
บทที่ 5	บทวิจารณ์และสรุป	55
หนังสืออ้างอิง		56
ภาคผนวก		57

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1

บทนำ

การควบคุมในงานอุตสาหกรรมในปัจจุบันได้มีการนำเอาโปรแกรมเมเบิลลอจิกคอนโทรลเลอร์ (Programmable Logic Control) หรือ PLC มาใช้ในการควบคุมเพิ่มมากขึ้น PLC มีหลายขนาด ตั้งแต่ขนาดเล็กไปจนถึงขนาดใหญ่ PLC ตัวเล็กมีความสามารถในการใช้งานน้อยกว่า PLC ตัวใหญ่ ความสามารถรวมถึงขนาดของช่องอินพุต (Input) และเอาต์พุต (Output) ว่าจะต่อกับอุปกรณ์ภายนอกได้มากเท่าใด และรวมไปถึงความสะดวกในการเขียนโปรแกรมควบคุม(การนำ PLC มาใช้งานจะต้องมีการเขียนโปรมกรมควบคุมด้วย) PLC ตัวใหญ่มีความสามารถที่จะเขียนโปรแกรมได้หลายภาษาซึ่ง PLC ตัวเล็กจะไม่มี ในปริญญานิพนธ์นี้ได้ศึกษาและทดลองสร้าง PLC ขนาดเล็กที่สามารถเขียนโปรมแกรม(คำสั่ง)ด้วยภาษาแลดเดอร์ (Leader Diagram) ที่มีแต่ใน PLC ขนาดใหญ่เท่านั้นและนับว่าเป็นการอำนวยความสะดวกในการเขียนโปรแกรมอีกวิธีหนึ่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2

แนะนำเกี่ยวกับ PLC/PC

2.1 ความเป็นมา

ระบบควบคุมในโรงงานอุตสาหกรรมทั่วๆไป สามารถแบ่งออกได้เป็น 2 ประเภทใหญ่ ๆ คือ

1. ระบบควบคุมแบบลูปปิด (Close loop)
2. ระบบควบคุมแบบซีเควนซ์ (Sequence control)

ระบบควบคุมแบบซีเควนซ์ เป็นการควบคุมที่ต้องการให้ระบบทำงานตามลำดับขั้นในช่วงเวลาและเงื่อนไขที่กำหนดไว้ โดยมีลักษณะการควบคุมเป็นแบบ ON หรือ OFF ระยะแรกๆในรอบซีเควนซ์ส่วนใหญ่ตัวควบคุมจะประกอบด้วยรีเลย์ (Relay) ต่อกันเป็นวงจรมากมาย รีเลย์แต่ละตัวประกอบด้วยขดลวดอาร์เมเจอร์,ขั้วแม่เหล็กและคอนแทค เมื่อมีกระแสไหลในขดลวดแม่เหล็ก อาร์เมเจอร์ก็จะถูกเหนี่ยวนำทำให้เกิดสนามแม่เหล็กซึ่งมีผลทำให้คอนแทคปิดหรือเปิดได้ รีเลย์แต่ละอันจะมีขนาดใหญ่ เมื่อนำรีเลย์จำนวนมากๆมาประกอบกันเป็นวงจรตัวควบคุมในระบบซีเควนซ์ จึงทำให้ตัวควบคุมมีขนาดใหญ่

เมื่อเทคโนโลยีทางด้านอุปกรณ์สารกึ่งตัวนำ (Solid state device) ได้เจริญก้าวหน้าขึ้น จึงได้มีการนำเอา Logic gate ต่าง ๆ มาประกอบเป็นวงจรอิเลคทรอนิคส์ เพื่อใช้เป็นตัวควบคุมแทนรีเลย์ ทำให้วงจรควบคุมของระบบมีขนาดเล็กลง และสิ้นเปลืองพลังงานน้อยลงอีกด้วย สามารถเปรียบเทียบตัวควบคุมที่ใช้รีเลย์กับวงจรอิเลคทรอนิคส์เกทต่าง ๆ ได้ดังนี้

	แบบที่ใช้รีเลย์	แบบวงจรเกท
1. ขนาด	ขนาดใหญ่	ขนาดเล็ก
2. การใช้พลังงาน	มาก	น้อย
3. ราคา : ระบบใหญ่	แพง	ถูก
ระบบเล็ก	ถูก	แพง

เอกสารนี้จัดทำขึ้นไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. ความสลับซับซ้อนในการควบคุม	ระบบที่มีความสลับซับซ้อนมาก ๆ จะทำไม่ได้	เหมาะสำหรับระบบที่สลับซับซ้อน
5. อายุการใช้งาน	ขึ้นกับรีเลย์ที่ใช้เพราะรีเลย์มีอายุการใช้งานจำกัด	ใช้งานได้นาน
6. สภาพแวดล้อมในการใช้งาน	ไม่ขึ้นอยู่กับสภาพแวดล้อมแต่ไม่ควรใช้ในบริเวณที่มีฝุ่นมาก	ต้องการสภาพแวดล้อมที่ดี ไม่มีฝุ่น อุณหภูมิไม่สูง ไม่มี noise และความชื้นไม่สูง
7. การซ่อมบำรุง	ทำได้ง่าย ต้องการ Periodic maintenance	ในกรณีที่เปลี่ยนแผงวงจรจะทำได้ง่าย อาจจะเกิดความยุ่งยากเมื่อต้องเปลี่ยน IC
8. การขยายระบบ	ทำได้ยาก	ทำได้ง่าย
9. การใช้กับคอมพิวเตอร์	ทำได้ยาก	ทำได้ง่าย

ถึงแม้ว่าจะนำเอาเกทมาทำเป็นวงจรควบคุมแล้วก็ตามแต่ก็ยังไม่ทำให้ระบบสมบูรณ์เพียงพอ เนื่องจากระบบวงจรเกทรวมทั้งระบบที่ใช้รีเลย์ โดยทั่วไปเป็นระบบที่สร้างขึ้นมาตายตัวได้เฉพาะอย่างเท่านั้น เมื่อมีการเปลี่ยนแปลงแก้ไขลำดับขั้นตอนต่างๆ หรือมีการขยายระบบ จะทำได้ยาก อาจจะต้องออกแบบวงจรใหม่ หรือทำการเดินสายใหม่ ซึ่งไม่สะดวก ดังนั้นเมื่อมีการผลิตไมโครโปรเซสเซอร์ (Microprocessor) ขึ้นมาใช้อย่างแพร่หลายและวิวัฒนาการของไมโครโปรเซสเซอร์ ทั้งทางด้านฮาร์ดแวร์และซอฟแวร์เจริญรุดหน้ามาก จึงได้มีการนำเอาไมโครโปรเซสเซอร์มาประกอบกับอุปกรณ์ต่างๆ ทำเป็นเครื่องควบคุม ที่สามารถกำหนดโปรแกรมการทำงานได้ซึ่งก็คือ PLC (Programmable Logic Controller) และ PC (Programmable Controller) นั่นเอง PLC/PC เป็นเครื่องควบคุม ที่สามารถแก้ปัญหาอันเกิดจากการใช้เครื่องควบคุมแบบรีเลย์และแบบวงจรเกทและสามารถนำมาใช้กับระบบควบคุมแบบซีเควนซ์ได้ทุกระบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ข้อดีของ PLC/PC สามารถพิจารณาได้ดังนี้

1.ขนาดของระบบ

ภายใน PLC ประกอบด้วยรีเลย์ (Relay) ตัวตั้งเวลา (Timer) ตัวนับ (Counter) และอุปกรณ์ประกอบวงจรซีเควนซ์อื่น ๆ ซึ่งจำนวนองค์ประกอบต่าง ๆ เหล่านี้ไม่ขึ้นอยู่กับขนาดของ PLC/PC ทำให้ระบบมีขนาดเล็กลง

2.ใช้โปรแกรมแทนการเดินสาย

ตัวควบคุมที่ใช้รีเลย์ในการประกอบเป็นวงจรต้องใช้การเดินสายระหว่างรีเลย์ แต่ระบบที่ใช้ PLC/PC จะใช้โปรแกรมแทนการเดินสาย การทำงานของวงจรต่างๆ จะเป็นไปตามโปรแกรม ดังนั้นจึงไม่มีการเดินสายระหว่างอุปกรณ์จริง ๆ ให้ยุ่งยาก

3.การเปลี่ยนวงจรและขยายระบบ

ถ้าต้องการเปลี่ยนลำดับขั้นการทำงาน หรือขยายระบบก็สามารถทำได้โดยง่าย โดยการเปลี่ยนแปลงแก้ไขโปรแกรมใน PLC/PC ใหม่ซึ่งในระบบที่ใช้รีเลย์หรือระบบที่ใช้วงจรเกทจะทำการเปลี่ยนแปลงได้โดยการแก้ไขการเดินสาย หรือออกแบบวงจรใหม่ซึ่งไม่สะดวก

4.ลดเวลาในการออกแบบและสร้าง

อุปกรณ์ต่างๆ ของ PLC เป็นมาตรฐาน สามารถออกแบบวงจรและโปรแกรมได้รวดเร็ว การติดตั้งทำได้สะดวกและง่าย นอกจากนี้ในระบบ PC ใหญ่ๆ ยังสามารถทดสอบวงจรได้ด้วย การตรวจสอบแก้ไขเมื่อมีปัญหาจึงทำได้รวดเร็ว

5.ความน่าเชื่อถือดี

ชิ้นส่วนต่างๆ ภายใน PLC/PC เป็นอุปกรณ์สารกึ่งตัวนำ และภายในระบบก็ไม่ได้ใช้การเดินสาย จึงไม่มีปัญหาเรื่องสายขาด คอนแทคหลวม หน้าสัมผัสไม่ดี และไม่จำเป็นต้องมี Periodic maintenance อย่างระบบที่ใช้รีเลย์ ดังนั้น PLC/PC จึงมีความน่าเชื่อถือดีกว่า

6.ราคาของระบบทั้งหมดจะถูกลดลง

รีเลย์ภายในระบบซอฟแวร์ จึงลดค่าใช้จ่ายในการเดินสาย ถ้าระบบควบคุมที่ใช้ PC ขนาดใหญ่ขึ้นถึงขนาดหนึ่ง จะมีราคาถูกกว่าระบบรีเลย์มาก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

7.การซ่อมบำรุงทำได้ง่าย

ใน PC ขนาดใหญ่ๆ จะมี Self Diagnostic Programme สำหรับทดสอบด้วยตนเอง มีเมื่อมีปัญหาเกิดขึ้น จึงสามารถทราบได้รวดเร็วและแก้ไขได้ทันที การเก็บ spare part มีน้อย และการเก็บเป็น module

2.2 การจำแนกตามขนาดของ PC

สามารถจำแนกตามมาตรฐานทั่ว ๆ ไปออกได้เป็น 4 ขนาดคือ

1. PCขนาดเล็ก ซึ่งก็คือ PLC นั่นเอง PC ขนาดเล็กจะไม่มีฟังก์ชั่นทางคณิตศาสตร์และจะใช้ควบคุมเครื่องจักรหรือระบบการทำงานในลักษณะของการควบคุมแบบ ON OFF เท่านั้นอินพุต/เอาต์พุต ขยายได้ถึง 128 จุด และเป็นแบบดิจิตอล ใช้ไมโครโปรเซสเซอร์ขนาด 4 บิทหรือ 8 บิท มีหน่วยความจำได้ถึง 2 กิโลเวิร์ด ใช้งานแทนรีเลย์สำหรับการควบคุมแบบซีเควนซ์เท่านั้น

2. PCขนาดกลางอินพุต/เอาต์พุต ขยายได้ถึง 1024 จุด และเป็นได้ทั้งแบบดิจิตอลและอนาล็อค ใช้ไมโครโปรเซสเซอร์ขนาด 8 บิท มีหน่วยความจำ 4 กิโลเวิร์ด และขยายได้ถึง 8 กิโลเวิร์ด มีฟังก์ชั่นทางคณิตศาสตร์ ใช้งานแทนรีเลย์สำหรับการควบคุมแบบซีเควนซ์ และใช้สำหรับการควบคุมแบบอนาล็อคได้ด้วย

3. PC ขนาดใหญ่ PC แบบนี้จะใช้กับระบบที่ยุ่งยากซับซ้อนมีอินพุต/เอาต์พุตที่ขยายได้ถึง 2048 จุด เป็นได้ทั้งดิจิตอลและอนาล็อค ใช้ไมโครโปรเซสเซอร์ขนาด 8 บิทหรือ 16 บิท มีหน่วยความจำ 12 กิโลเวิร์ด ถึง 22 กิโลเวิร์ด มีฟังก์ชั่นทางคณิตศาสตร์เพิ่มขึ้น

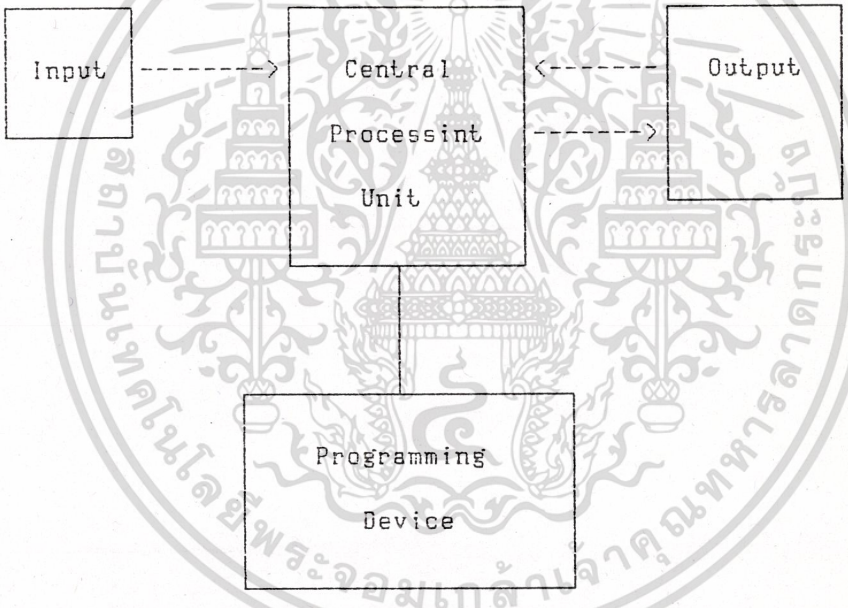
4. PC ขนาดใหญ่พิเศษ PC แบบนี้ จะใช้ในงานที่มีอินพุต/เอาต์พุตมาก ๆ อินพุต/เอาต์พุตที่ใช้ขยายได้ถึง 8192 จุด และเป็นได้ทั้งแบบดิจิตอลและอนาล็อค ใช้ไมโครโปรเซสเซอร์ขนาด 16 บิท หรือไมโครโปรเซสเซอร์หลาย ๆ ตัว มีหน่วยความจำ 64 กิโลเวิร์ด และขยายได้ถึง 128 กิโลเวิร์ด มีฟังก์ชั่นทางคณิตศาสตร์มากมาย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3 โครงสร้างโดยทั่วไปของ PLC/PC

PLC/PC เป็นเครื่องมือควบคุมที่ประกอบด้วยอุปกรณ์สารกึ่งตัวนำ ประกอบกันเป็นวงจรเพื่อใช้งานกับระบบควบคุมซีเควนซ์ โดยให้ทำงานตามโปรแกรมคำสั่งของผู้ใช้และข้อมูลต่าง ๆ ที่ได้รับจาก อินพุต และ เอาต์พุต ของ PLC/PC โครงสร้างของ PLC/PC สามารถแบ่งออกได้เป็น 3 ส่วนคือ

1. หน่วยประมวลผลกลาง (Central Processing Unit)
2. หน่วยอินพุต/เอาต์พุต (Input/Output Unit)
3. หน่วยป้อนโปรแกรม (Programming Unit)



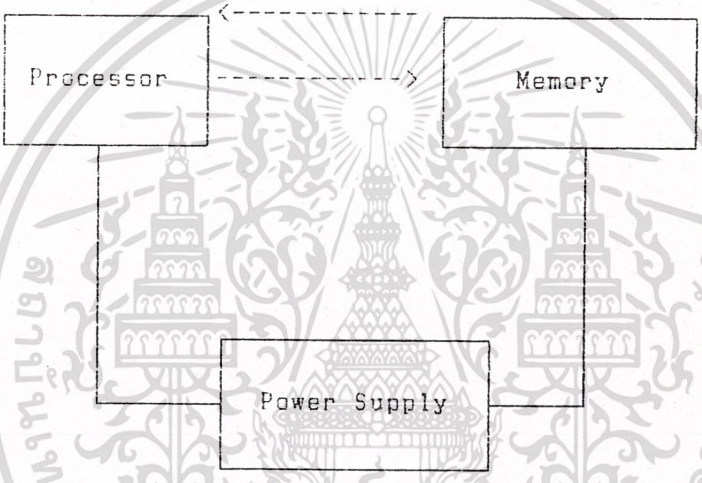
หน่วยประมวลผลกลาง

หน่วยประมวลผลกลางหรือ CPU ประกอบด้วยส่วนต่าง ๆ 3 ส่วนคือ

1. หน่วยประมวลผล (Processor) ทำหน้าที่ควบคุมและดูแลการทำงานของระบบทั้งหมด โดยรับข้อมูลจากหน่วยอินพุต มาทำการประมวลตามโปรแกรม และส่งผลที่ได้ออกไปยังหน่วย เอาต์พุต นอกจากนี้ยังทำหน้าที่ติดต่อกับระบบย่อย (subsystem) ตรวจสอบความผิดพลาดต่าง ๆ หน่วยประมวลผลของ CPU จะใช้ไมโครโปรเซสเซอร์ 1 ตัว หรือหลาย ๆ ตัวมาทำงานร่วมกัน ทำให้มีขีดความสามารถสูงขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทำงานของ CPU 1 รอบ เรียกว่า SCAN TIME โดยเริ่มนับเวลาตั้งแต่การรับข้อมูลจากหน่วยอินพุต ประมวลผลตามโปรแกรมคำสั่งผู้ใช้ ส่งผลไปยังหน่วยเอาต์พุต จนถึงการตรวจสอบการทำงานของระบบทั้งหมดและการติดต่อกับผู้ใช้ เวลา SCAN จะอ้างถึงขนาดของหน่วยความจำที่เก็บโปรแกรมของผู้ใช้เสมอ เวลา SCAN เป็นองค์ประกอบที่สำคัญอย่างหนึ่งในการเลือกใช้ PLC/PC ให้เหมาะสมกับงาน เพราะว่าช่วงเวลา SCAN ของ CPU นี้จะทำให้ทราบถึงความสามารถในการตอบสนองต่อการเปลี่ยนของ อินพุต/เอาต์พุต ว่ารวดเร็วเพียงใด



รูปแสดงโครงสร้างของ CPU

2. หน่วยความจำ (Memory) ทำหน้าที่เก็บโปรแกรมคำสั่ง และข้อมูลที่หน่วยประมวลผลใช้ในการควบคุมการทำงานของ PLC/PC โดยที่ข้อมูลที่เก็บภายในหน่วยความจำนี้มีค่าทางตรรกเป็น "0" หรือ "1" ข้อมูลที่เคลื่อนย้ายและออกจากหน่วยความจำแต่ละครั้งจะเป็นจำนวนบิท (bit) ที่แน่นอนจำนวนหนึ่ง อาจจะเป็นข้อมูลขนาด 8 บิท เรียกว่า ไบท์ (byte) หรือ 16 บิทที่เรียกว่า คำ เราสามารถจำแนกหน่วยความจำออกเป็น 2 ประเภทคือ

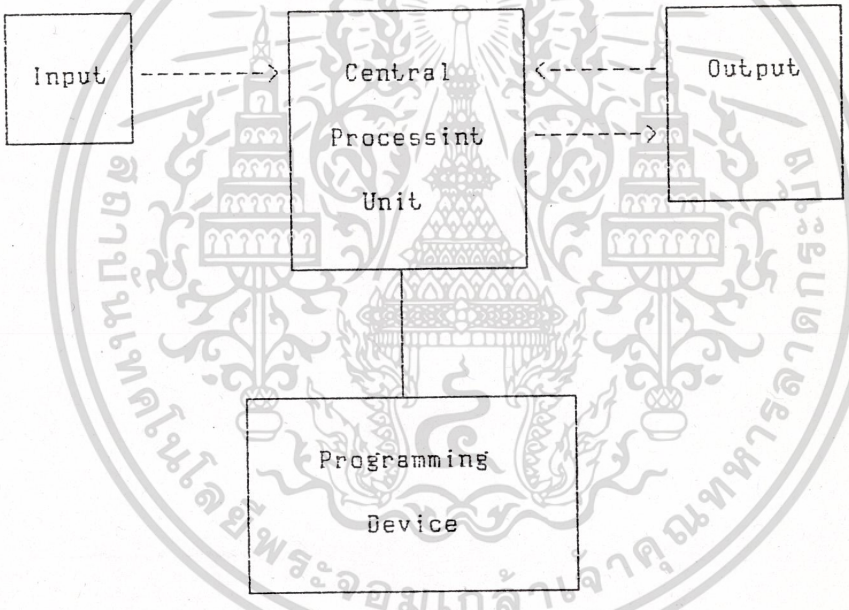
- หน่วยความจำชนิด RAM (Ramdom Access Memory) การเข้าถึงข้อมูลจะเป็นแบบ แรนด้อม สามารถอ้างถึงตำแหน่งที่เก็บข้อมูลได้โดยตรงทุกตำแหน่งทำให้เวลาในการเข้าถึงข้อมูลนั้นเท่ากันเสมอ โครงสร้าง จะประกอบไปด้วยวงจรหน่วยความจำที่มีขนาดเล็กที่สุดคือ 1 บิท อยู่จำนวนมากมายเรียงกันอยู่ภายใน และมีวงจรเลือกตำแหน่งหรือถอดรหัสตำแหน่งรวมอยู่ด้วย ข้อมูลในหน่วยความจำนี้จะถูกลบออกไปเมื่อไม่มีไฟเลี้ยงหรือเรียกอีกอย่างหนึ่งเป็นหน่วยความจำชั่วคราว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3 โครงสร้างโดยทั่วไปของ PLC/PC

PLC/PC เป็นเครื่องมือควบคุมที่ประกอบด้วยอุปกรณ์สารกึ่งตัวนำ ประกอบกันเป็นวงจรเพื่อใช้งานกับระบบควบคุมซีเควนซ์ โดยให้ทำงานตามโปรแกรมคำสั่งของผู้ใช้และข้อมูลต่าง ๆ ที่ได้รับจาก อินพุต และ เอาต์พุต ของ PLC/PC โครงสร้างของ PLC/PC สามารถแบ่งออกได้เป็น 3 ส่วนคือ

1. หน่วยประมวลผลกลาง (Central Processing Unit)
2. หน่วยอินพุต/เอาต์พุต (Input/Output Unit)
3. หน่วยป้อนโปรแกรม (Programming Unit)



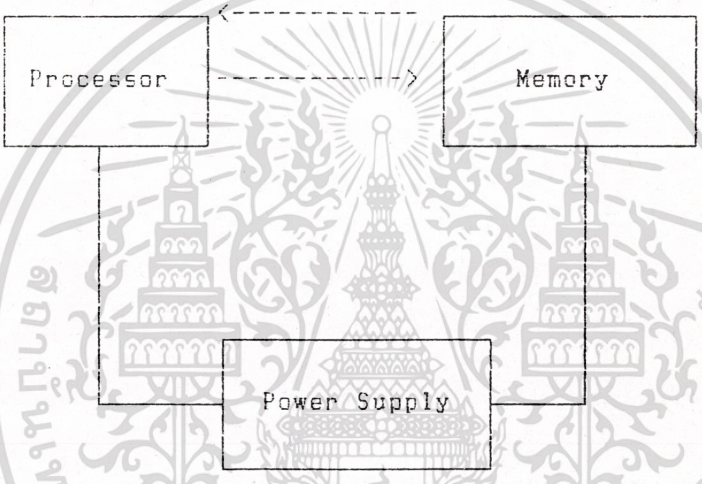
หน่วยประมวลผลกลาง

หน่วยประมวลผลกลางหรือ CPU ประกอบด้วยส่วนต่าง ๆ 3 ส่วนคือ

1. หน่วยประมวลผล (Processor) ทำหน้าที่ควบคุมและดูแลการทำงานของระบบทั้งหมด โดยรับข้อมูลจากหน่วยอินพุต มาทำการประมวลตามโปรแกรม และส่งผลที่ได้ออกไปยังหน่วย เอาต์พุต นอกจากนี้ยังทำหน้าที่ติดต่อกับระบบย่อย (subsystem) ตรวจสอบความผิดพลาดต่าง ๆ หน่วยประมวลผลของ CPU จะใช้ไมโครโปรเซสเซอร์ 1 ตัว หรือหลาย ๆ ตัวมาทำงานร่วมกัน ทำให้มีขีดความสามารถสูงขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทำงานของ CPU 1 รอบ เรียกว่า SCAN TIME โดยเริ่มนับเวลาตั้งแต่การรับข้อมูลจากหน่วยอินพุต ประมวลผลตามโปรแกรมคำสั่งผู้ใช้ ส่งผลไปยังหน่วยเอาต์พุต จนถึงการตรวจสอบการทำงานของระบบทั้งหมดและการติดต่อกับผู้ใช้ เวลา SCAN จะอ้างถึงขนาดของหน่วยความจำที่เก็บโปรแกรมของผู้ใช้เสมอ เวลา SCAN เป็นองค์ประกอบที่สำคัญอย่างหนึ่งในการเลือกใช้ PLC/PC ให้เหมาะสมกับงาน เพราะว่าช่วงเวลา SCAN ของ CPU นี้จะทำให้ทราบถึงความสามารถในการตอบสนองต่อการเปลี่ยนของ อินพุต/เอาต์พุต ว่ารวดเร็วเพียงใด



รูปแสดงโครงสร้างของ CPU

2. หน่วยความจำ (Memory) ทำหน้าที่เก็บโปรแกรมคำสั่ง และข้อมูลที่หน่วยประมวลผลใช้ในการควบคุมการทำงานของ PLC/PC โดยที่ข้อมูลที่เก็บภายในหน่วยความจำนี้มีค่าทางตรรกเป็น "0" หรือ "1" ข้อมูลที่เคลื่อนย้ายและออกจากหน่วยความจำแต่ละครั้งจะเป็นจำนวนบิท (bit) ที่แน่นอนจำนวนหนึ่ง อาจจะเป็นข้อมูลขนาด 8 บิท เรียกว่า ไบท์ (byte) หรือ 16 บิทที่เรียกว่า คำ เราสามารถจำแนกหน่วยความจำออกเป็น 2 ประเภทคือ

- หน่วยความจำชนิด RAM (Ramdom Access Memory) การเข้าถึงข้อมูลจะเป็นแบบ แรนด้อม สามารถอ้างถึงตำแหน่งที่เก็บข้อมูลได้โดยตรงทุกตำแหน่งทำให้เวลาในการเข้าถึงข้อมูลนั้นเท่ากันเสมอ โครงสร้าง จะประกอบไปด้วยวงจรหน่วยความจำที่มีขนาดเล็กที่สุดคือ 1 บิท อยู่จำนวนมากมายเรียงกันอยู่ภายใน และมีวงจรเลือกตำแหน่งหรือถอดรหัสตำแหน่งรวมอยู่ด้วย ข้อมูลในหน่วยความจำนี้จะถูกลบออกไปเมื่อไม่มีไฟเลี้ยงหรือเรียกอีกอย่างหนึ่งเป็นหน่วยความจำชั่วคราว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- หน่วยความจำชนิด ROM (Read Only Memory) หน่วยความจำชนิดนี้จะเป็นชนิดเขียนได้แบบถาวร กล่าวคือข้อมูลจะไม่สูญหายไปเมื่อไม่มีไฟเลี้ยงหรือไฟฟ้าดับก็ตาม เวลาใช้งานก็อ่านออกมาได้เลย การนำข้อมูลเข้าไปเก็บใน ROM เรียกว่า การโปรแกรมรอม(programming) หลังจากโปรแกรมแล้ว หน่วยความจำชนิดนี้จะใช้อ่านได้อย่างเดียว

3. หน่วยจ่ายกำลังงาน(Power Supply)

หน่วยจ่ายกำลังงานทำหน้าที่จ่ายกระแสไฟฟ้าให้กับส่วนต่าง ๆ ของ PC คือ หน่วยประมวลผล หน่วยความจำ และอินพุต/เอาต์พุต โดยรักษาแรงดันไฟฟ้าให้คงที่และแจ้งให้หน่วยประมวลผลทราบเมื่อการทำงานผิดปกติ ความน่าเชื่อถือของ PC ขึ้นอยู่กับการทำงานของหน่วยจ่ายกำลังงานนี้ด้วย

แรงดันไฟฟ้าอินพุต

หน่วยจ่ายกำลังงานของ PC มักจะรับกระแสไฟฟ้าสลับที่มีแรงดัน 110 VAC หรือ 220 VAC แต่มี PC เฉพาะงานที่ใช้แรงดันไฟฟ้าอื่น เช่น แท่นขุดเจาะและสำรวจปิโตเลียม ซึ่งส่วนใหญ่ใช้แรงดันไฟฟ้าขนาด 24 VDC จะใช้ PC ที่ต้องการแรงดันไฟฟ้ากระแสตรงขนาด 24 VDC แทน 110/220 VAC

PC ถูกสร้างขึ้นใช้ในโรงงานอุตสาหกรรมซึ่งแรงดันไฟฟ้ามักไม่คงที่ และมีการรบกวนจากสนามแม่เหล็กไฟฟ้า หน่วยจ่ายกำลังงานของ PC จึงต้องถูกออกแบบเป็นพิเศษ เพื่อให้สามารถปรับแรงดันไฟฟ้าที่มีการเปลี่ยนแปลงจาก 10 ถึง 15 เปอร์เซ็นต์ให้คงที่ตลอดเวลา เช่น PC ที่ใช้แรงดันไฟฟ้า 220 VAC ต้องทำงานตามปกติเมื่อแรงดันมีค่า 194 ถึง 250 VAC ถึงแม้ว่าแรงดันไฟฟ้าจะเปลี่ยนแปลงมากหน่วยจ่ายกำลังงานก็ต้องทนต่อสภาพผิดปกติดังกล่าวและแจ้งให้ CPU ทราบก่อนตัดวงจรออกโดยอัตโนมัติ สาเหตุที่ทำให้แรงดันไฟฟ้าผิดปกติที่พบเสมอคือ

1. การเปิดหรือปิดเครื่องจักรไฟฟ้าขนาดใหญ่ เช่น มอเตอร์ไฟฟ้าปั๊มเครื่องเชื่อม และ เครื่องปรับอากาศ
2. แรงดันไฟฟ้าตกเนื่องจากการเดินสายไฟฟ้าระยะไกล
3. การติดตั้งอุปกรณ์ไฟฟ้าที่ไม่ถูกต้องทำให้กระแสไฟฟ้ารั่ว

โรงงานอุตสาหกรรมที่แรงดันไฟฟ้าเปลี่ยนแปลงอยู่เสมอ ต้องใช้อุปกรณ์พิเศษเพื่อรักษาแรงดันไฟฟ้าให้คงที่ อุปกรณ์เหล่านี้ประกอบด้วยหม้อแปลงรักษาระดับแรงดันไฟฟ้า (constant

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

voltage transformer) หม้อแปลงไฟฟ้า Sola CVS และหม้อแปลงกำจัดสัญญาณรบกวน (isolation transformer)

หม้อแปลงรักษาระดับแรงดันไฟฟ้า ทำหน้าที่ปรับแรงดันไฟฟ้าให้เปลี่ยนแปลงไม่เกิน 1 เปอร์เซ็นต์ ถึงแม้แรงดันอินพุตจะเปลี่ยนแปลงมากกว่า 15 เปอร์เซ็นต์

หม้อแปลงไฟฟ้า Sola CVS ผลิตโดยบริษัท Sola Basic Industry ทำหน้าที่คล้ายหม้อแปลงรักษาระดับแรงดันไฟฟ้าแต่มีคุณสมบัติพิเศษในการกำจัดสัญญานฮาร์มอนิก(Harmonic)

หม้อแปลงกำจัดสัญญาณรบกวนทำหน้าที่ตัดการรบกวนจากคลื่นแม่เหล็กไฟฟ้าออกจาก PC ทำให้การทำงานของ PC และแหล่งจ่ายกำลังงานดีขึ้น เหมาะสำหรับติดตั้งให้ PC ที่อยู่ใกล้เครื่องจักรไฟฟ้าขนาดใหญ่

อัตราจ่ายกระแสไฟฟ้าของหน่วยจ่ายกำลังงาน

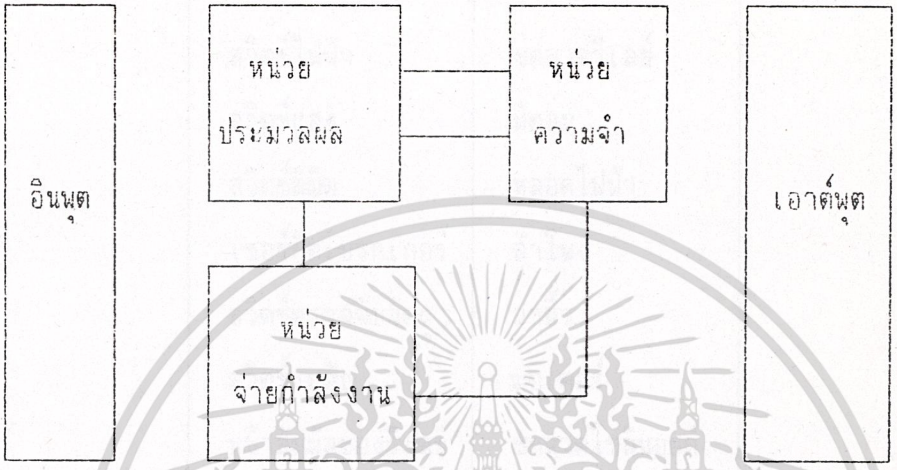
ขนาดของหน่วยจ่ายกำลังงานต้องสามารถจ่ายกระแสไฟฟ้าให้อุปกรณ์อื่น ๆ ได้ เช่น CPU และหน่วยอินพุต/เอาต์พุตทำงานตามปกติการคำนวณขนาดของหน่วยจ่ายกำลังงานต้องทราบจำนวนกระแสไฟฟ้าที่อุปกรณ์ทั้งหมดต้องการโดยเฉพาะหน่วยอินพุต/เอาต์พุตต้องทราบอัตราการใช้กระแสไฟฟ้าทั้งในสภาวะ ON และสภาวะ OFF หน่วยจ่ายกำลังงานของ PC จากบริษัทผู้ผลิตจะถูกออกแบบให้สามารถจ่ายกระแสไฟฟ้าให้กับหน่วยประมวลผล หน่วยความจำ และหน่วยอินพุต/เอาต์พุตทั้งหมดที่ต้องการ แต่บางกรณีที่มีหน่วยอินพุต/เอาต์พุตแบบพิเศษซึ่งต้องการกำลังไฟฟ้าสูงเป็นจำนวนมากอาจต้องเพิ่มหน่วยจ่ายกำลังพิเศษ

หน่วยจ่ายกำลังที่มีขนาดเล็กเกินไปจะทำให้อุปกรณ์ต่างๆ ทำงานผิดปกติโดยไม่ทราบสาเหตุเช่น หน่วยเอาต์พุตเปลี่ยนสภาวะ ON/OFF ไม่ได้ตามเวลาที่กำหนด หรือ CPU และหน่วยความจำทำงานผิดพลาดบ่อยครั้ง

<u>หน่วยอินพุต/เอาต์พุต</u>

หน่วยอินพุต/เอาต์พุตทำหน้าที่ติดต่อระหว่าง PC กับอุปกรณ์ภายนอก รูปข้างล่างจะแสดงหน่วยอินพุตเอาต์พุตของ PC หน่วยอินพุตทำหน้าที่รับสภาวะและค่าวัดจากอุปกรณ์ภายนอก เช่น การ ON/OFF ของสวิตช์ ตำแหน่งเครื่องจักร ระดับของของเหลว อุณหภูมิ ความดัน ระดับแรงดันและกระแสไฟฟ้าส่งต่อให้ PC CPU จะใช้ค่าหรือสภาวะจากหน่วยอินพุต/เอาต์พุต เป็นข้อมูลในการประมวลผลตามโปรแกรมผู้ใช้ และส่งผลที่ได้ไปที่หน่วยเอาต์พุตเพื่อควบคุมอุปกรณ์ภายนอก เช่น ริเลย์มอเตอร์ไฟฟ้า ปั๊ม และวาล์ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
027871



หน่วยอินพุต/เอาต์พุตของ PC

PC ที่มีหน่วยอินพุต/เอาต์พุตแบบสภาวะตรรก (discrete input/output) เพียงชนิดเดียว ทำให้ PC ถูกใช้ในการควบคุมได้เพียงแบบ ON/OFF ซึ่งเป็นส่วนหนึ่งของระบบควบคุมทั้งหมดเท่านั้น ปัจจุบันมีการเพิ่มหน่วยอินพุต/เอาต์พุตแบบตัวเลข (numerical data input /output) ทำให้ขอบเขตการใช้ PC ในการควบคุมกว้างขึ้น ทั้งการควบคุมเครื่องจักรและกระบวนการอุตสาหกรรม แต่ในโครงงานนี้ใช้หน่วยอินพุต/เอาต์พุตเพียงแบบสภาวะตรรกเท่านั้น

หน่วยอินพุต/เอาต์พุตแบบสภาวะตรรก

หน่วยอินพุต/เอาต์พุตชนิดนี้ ทำหน้าที่ติดต่อกับอุปกรณ์ภายนอกที่มีการเปลี่ยนแปลงเพียง 2 สภาวะ เช่น การ ON/OFF ของสวิตช์ไฟฟ้าหรือหน้าสัมผัสของรีเลย์ ตารางที่ 1 ได้แสดงตัวอย่างอุปกรณ์อินพุตและเอาต์พุตแบบสภาวะตรรก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หน่วยอินพุตแบบสภาวะตรรกทำหน้าที่รับค่าทางไฟฟ้าของอุปกรณ์ภายนอกและเปลี่ยนเป็นสภาวะทางตรรกเพื่อเก็บไว้ในหน่วยความจำ ส่วนตารางอินพุตของ CPU บิตที่มีค่า "1" หมายถึงการ ON หรือปิดวงจรไฟฟ้า และบิตที่มีค่า "0" หมายถึงการ OFF หรือเปิดวงจรไฟฟ้าของอุปกรณ์อินพุต หน่วยเอาต์พุตจะรับสภาวะตรรกจากตารางเอาต์พุตของ CPU และเปลี่ยนเป็นค่าทางไฟฟ้าเพื่อควบคุมอุปกรณ์ภายนอกอีกทีหนึ่ง บิตที่มีค่า "1" หมายถึงการ ON หรือการต่อวงจรไฟฟ้า และบิตที่มีค่า "0" หมายถึงการ OFF หรือการตัดวงจรไฟฟ้า การเลือกใช้หน่วยอินพุต/เอาต์พุตชนิดที่ถูกต้องและเหมาะสมจำเป็นจะต้องเข้าใจคุณลักษณะและการทำงานของหน่วยอินพุต/เอาต์พุตแต่ละชนิดด้วย

ในปัจจุบันยังมีการพัฒนาหน่วยหน่วยอินพุต/เอาต์พุตของ PC เพิ่มขึ้นอีกหลายชนิดเช่น

1. หน่วยอินพุต/เอาต์พุตแบบสภาวะตรรก

หน่วยอินพุต AC/DC

ทำหน้าที่รับสัญญาณไฟฟ้าจากอุปกรณ์อินพุต เปรียบเทียบระดับแรงดันไฟฟ้าที่รับเข้ามาทางอินพุตแล้วเปลี่ยนเป็นสภาวะทางลอจิกส่งไปยัง CPU เพื่อประมวลผลต่อไป หน่วยอินพุตแบบนี้ต้องใช้เวลาในการตรวจสอบสภาวะของอุปกรณ์อินพุตประมาณ 9 ถึง 25 มิลลิวินาที

หน่วยอินพุตแบบ TTL

ทำหน้าที่รับสัญญาณอินพุตจากอุปกรณ์อิเล็กทรอนิกส์ชนิด TTL (Transister Transister Logic) หรืออุปกรณ์ที่ใช้สัญญาณไฟฟ้า 5 VDC หน่วยอินพุตชนิดนี้ทำงานได้เร็วกว่าหน่วยอินพุตแบบ AC/DC คือใช้เวลาเพียง 1 ถึง 3 มิลลิวินาที

หน่วยอินพุตแบบหน้าสัมผัส (contact input)

ทำหน้าที่ตรวจสอบสภาวะของอุปกรณ์อินพุตที่ไม่มีแรงดันไฟฟ้าในตัว เช่น สวิตช์ไฟฟ้าและหน้าสัมผัสของรีเลย์ โดยหน่วยอินพุตชนิดนี้จะใช้แรงดันไฟฟ้า 12 หรือ 24 VDC จ่ายให้อุปกรณ์ภายนอก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หน่วยเอาต์พุตแบบ AC

ทำหน้าที่รับสภาวะการควบคุมจาก CPU แล้วเปลี่ยนให้เป็นระดับแรงดันไฟฟ้ากระแสสลับเพื่อควบคุมการทำงานของอุปกรณ์เอาต์พุต หน่วยเอาต์พุตแบบ AC ประกอบด้วยวงจรตรรกซึ่งทำหน้าที่ติดต่อกับ CPU วงจรเชื่อมต่อแบบออปติค และวงจรสวิตช์อิเล็กทรอนิกส์ (electronics switch) ซึ่งใช้ไทร์แอค(Triac)หรือ SCR เป็น Switch On/Off เป็นต้น

หน่วยเอาต์พุตแบบ DC

ทำหน้าที่ควบคุมอุปกรณ์เอาต์พุตที่ทำงานด้วยสัญญาณไฟฟ้ากระแสตรง ลักษณะการทำงานของวงจรคล้ายกับหน่วยเอาต์พุตแบบ AC แต่ใช้ทรานซิสเตอร์เปิดและปิดวงจรไฟฟ้าแทนไทร์แอคและ SCR

หน่วยเอาต์พุตแบบหน้าสัมผัส (contact output)

ทำหน้าที่แทนหน้าสัมผัสของรีเลย์ในการควบคุมอุปกรณ์ที่มีแรงดันไฟฟ้าของตนเอง ลักษณะและการทำงานของวงจรคล้ายกับหน่วยเอาต์พุตแบบ AC และ DC แต่ใช้หน้าสัมผัสและรีเลย์เปิดและปิดวงจรไฟฟ้าแทนอุปกรณ์อิเล็กทรอนิกส์ หน่วยเอาต์พุตชนิดนี้ใช้กับอุปกรณ์ไฟฟ้ากระแสตรงหรือไฟฟ้ากระแสสลับก็ได้

หน่วยเอาต์พุตแบบ TTL

ทำหน้าที่ควบคุมการทำงานของอุปกรณ์อิเล็กทรอนิกส์ที่เป็น TTL หรืออุปกรณ์ที่ใช้สัญญาณไฟฟ้า 5 VDC หน่วยเอาต์พุตชนิดนี้มักต้องการหน่วยจ่ายกำลังงานขนาด 5 VDC จากภายนอก

2. หน่วยอินพุต/เอาต์พุตแบบตัวเลข

ไมโครโปรเซสเซอร์ทำให้ PC สามารถทำการคำนวณทางคณิตศาสตร์ ตรวจสอบและเคลื่อนย้ายข้อมูลที่เป็นตัวเลข หน่วยอินพุต/เอาต์พุตแบบจำนวนเลขทำให้ PC สามารถปรับข้อมูลจากอุปกรณ์วัดและควบคุมเครื่องจักรหรือกระบวนการอุตสาหกรรม ระบบควบคุมที่ใช้ PC จึงครบวงจรทั้งการควบคุมแบบ ON/OFF และแบบอะนาลอก

หน่วยอินพุต/เอาต์พุตแบบตัวเลขมี 2 ชนิดคือ แบบรหัสเลขฐานสอง และแบบอะนาลอก หน่วยอินพุต/หน่วยเอาต์พุตแบบรหัสเลขฐานสองทำงานคล้ายกับแบบสภาวะตรรก แต่การรับและส่งข้อมูลทำได้พร้อมกันครั้งละหลายบิต ข้อมูลอาจอยู่ในรูปรหัส ASCII รหัส BCD

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หรือรหัส GRAY ติดต่อกับอุปกรณ์วัดและควบคุมในกระบวนการอุตสาหกรรม โดยในตารางที่ 3 ได้แสดงตัวอย่างอุปกรณ์อินพุต/เอาต์พุตแบบจำนวนเลขทั้ง 2 ชนิด

อุปกรณ์อินพุตแบบอนาลอก	อุปกรณ์เอาต์พุตแบบอนาลอก
เครื่องวัดอุณหภูมิ	วาล์วควบคุม
เครื่องวัดความดัน	เครื่องบันทึกสัญญาน
สเตรนเกจ	ระบบเซอร์โว(Servo)
เครื่องวัดความชื้น	มิเตอร์แบบต่างๆ
เครื่องวัดอัตราการไหล	
เครื่องวัดแรงดันไฟฟ้า	

อุปกรณ์อินพุตแบบรหัสเลขฐานสอง	อุปกรณ์เอาต์พุตแบบรหัสเลขฐานสอง
สวิตช์รหัส(Thumbwheel Switch)	ตัวเลขแสดงผลเจ็ดส่วน
เครื่องอ่านรหัสบาร์(Bar Code Reader)	จอภาพ
วงจรเข้ารหัส(Encoder)	

ตารางที่ 3 แสดงอุปกรณ์อินพุต/เอาต์พุตแบบตัวเลข

สัญญาณมาตรฐานของอุปกรณ์อินพุต/เอาต์พุตแบบตัวเลข

สัญญาณไฟฟ้าของอุปกรณ์อินพุต/เอาต์พุตแบบอนาลอกอาจมีทั้งค่าบวกและลบหรือค่าบวกหรือลบอย่างใดอย่างหนึ่ง ซึ่งผู้ใช้สามารถปรับเลือกขนาดและชนิดของสัญญาณได้ โดยใช้วิธีทางซอฟต์แวร์หรือฮาร์ดแวร์ได้ ตารางที่ 4 ได้แสดงระดับสัญญาณไฟฟ้ามาตราฐานของหน่วยอินพุต/เอาต์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

พุตชนิดนี้ สำหรับหน่วยอินพุต/เอาต์พุตแบบรหัสเลขฐานสองมักจะใช้สัญญาณชนิดเดียวกับอุปกรณ์อินพุต/เอาต์พุตแบบสภาวะตรรก

อุปกรณ์อินพุต	อุปกรณ์อินพุต
4-20 mADC	4-20 mADC
0-1 VDC	10-50 mADC
0-5 VDC	0-5 VDC
0-10 VDC	0-10 VDC
1-5 VDC	25 VDC
5 VDC	5 VDC
10 VDC	10 VDC

ตารางที่ 4 แสดงสัญญาณไฟฟ้าของอุปกรณ์อินพุต/เอาต์พุตแบบอนาลอก

หน่วยอินพุตแบบอะนาลอก

ทำหน้าที่รับสัญญาณอินพุตจากอุปกรณ์ภายนอกปรับระดับให้เหมาะสม และใช้วงจรแปลงสัญญาณอะนาลอก/ดิจิตอล หรือ ADC (analog to digital converter) เปลี่ยนสัญญาณอะนาลอกเป็นค่าทางดิจิตอลส่งให้ CPU วงจรส่วนแรกของหน่วยอินพุตคือวงจรกันชนและวงจรกรองความถี่เพื่อลดสัญญาณรบกวนจากภายนอก วงจรกันชนทำหน้าที่เพิ่มค่าอิมพิแดนซ์ด้านอินพุต (input impedancd) ให้สูงขึ้นเพื่อเชื่อมต่อเข้ากับอุปกรณ์อินพุตที่มีอิมพิแดนซ์สูง สายส่งที่เชื่อมต่อระหว่างหน่วยอินพุตและอุปกรณ์ภายนอกควรเป็นสายโคแอคเชียลเพื่อป้องกันการรบกวนจากสนามไฟฟ้าและรักษาค่าอิมพิแดนซ์ให้คงที่

หน่วยเอาต์พุตแบบอะนาลอก

ทำหน้าที่รับข้อมูลจาก CPU เปลี่ยนเป็นระดับสัญญาณไฟฟ้าโดยใช้วงจรแปลงสัญ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ญาณดิจิตอล/อะนาลอกหรือ DAC (Digital to Analog Donverter) เพื่อควบคุมอุปกรณ์ภายนอก หน่วยเอาต์พุตชนิดนี้ใช้การเชื่อมต่อแบบออปติคเพื่อป้องกันการรบกวน CPU จากอุปกรณ์ภายนอก

คุณลักษณะของหน่วยอินพุต/เอาต์พุต

ผู้ใช้ PC ควรทราบวิธีการกำหนดคุณลักษณะของ PC และหน่วยอินพุต/เอาต์พุตของบริษัทผู้ผลิต ทั้งคุณลักษณะทางไฟฟ้าและการติดตั้ง และทำความเข้าใจกับรายละเอียดเหล่านี้ก่อนการใช้งานทุกครั้ง

1. คุณลักษณะทางไฟฟ้า มีลักษณะที่สำคัญดังต่อไปนี้คือ

1.1 แรงดันอินพุต (input voltage rating) หมายถึงระดับสัญญาณอินพุตที่หน่วยอินพุตใช้แยกความแตกต่างระหว่างสภาวะ ON และสภาวะ OFF ของอุปกรณ์ภายนอก สัญญาณอินพุตสามารถเปลี่ยนแปลงได้ประมาณ 10 ถึง 15 เปอร์เซ็นต์

1.2 กระแสอินพุต (input current rating) หมายถึง จำนวนกระแสไฟฟ้าที่หน่วยอินพุตต้องการจากอุปกรณ์อินพุตเมื่อมีสภาวะ ON และ OFF

1.3 ระดับเทรสโฮลด์ (input thresholf voltage) หมายถึงช่วงการเปลี่ยนแปลงระดับแรงดันอินพุตที่หน่วยอินพุตสามารถแยกความแตกต่างระหว่างสภาวะ ON และ OFF

1.4 ช่วงเวลาอินพุต (input delay) หมายถึงช่วงเวลาที่หน่วยอินพุตรับสภาวะของอุปกรณ์อินพุต เช่น หน่วยอินพุตแบบ AC/DC มีช่วงเวลาอินพุตประมาณ 9 ถึง 25 มิลลิวินาที และ 1 ถึง 3 มิลลิวินาทีสำหรับหน่วยอินพุตแบบ TTL

1.5 แรงดันเอาต์พุต (output voltage rating) หมายถึง ระดับสัญญาณไฟฟ้าของหน่วยเอาต์พุตเมื่อมีสภาวะ ON และ OFF แรงดันเอาต์พุตนี้อาจเปลี่ยนแปลงประมาณ 10 เปอร์เซ็นต์ เช่น หน่วยเอาต์พุตที่มีแรงดันเอาต์พุต 48 VDC จะส่งแรงดันไฟฟ้า 42 ถึง 56 VDC

1.6 กระแสเอาต์พุต (output current rating) หมายถึงจำนวนกระแสไฟฟ้าสูงสุดที่หน่วยเอาต์พุตสามารถจ่ายให้อุปกรณ์ภายนอกหรืออนุญาตให้กระแสไฟฟ้าไหลผ่านเมื่อมีสภาวะ ON

1.7 กำลังเอาต์พุต (output power rating) หมายถึงขนาดกำลังไฟฟ้าสูงสุดที่หน่วยเอาต์พุตจ่ายให้อุปกรณ์ภายนอก

1.8 อัตราการใช้กระแส (current requirement) หมายถึงจำนวนกระแสไฟฟ้าที่หน่วยอินพุตและเอาต์พุตต้องการจากหน่วยจ่ายกำลังงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.9 อัตราการทนกระแสสูงสุด (maximum surge current) หมายถึงจำนวนกระแสไฟฟ้าสูงสุดที่มีค่ามากกว่ากระแสเอาต์พุต และหน่วยเอาต์พุตสามารถจ่ายให้อุปกรณ์ภายนอกได้โดยไม่เกิดความเสียหายชั่วระยะเวลาหนึ่ง เช่น 20 A @ 1 s หมายถึงหน่วยเอาต์พุตสามารถจ่ายกระแสเอาต์พุตได้ 20 แอมแปร์ เป็นเวลา 1 วินาที ก่อนที่หน่วยเอาต์พุตจะเสียหาย

1.10 กระแสรั่วซึม (off state leakage current) หมายถึงจำนวนกระแสไฟฟ้าที่รั่วจากหน่วยเอาต์พุตเมื่อมีสภาวะ OFF ปกติจะมีค่าประมาณ 2 หรือ 3 ไมโครแอมแปร์ ซึ่งกระแสไฟฟ้าจำนวนนี้อาจทำให้อุปกรณ์เอาต์พุตความไวสูงทำงานผิดพลาดได้

1.11 ช่วงเวลาเอาต์พุต ON (output ON-delay) หมายถึงช่วงเวลาที่หน่วยเอาต์พุตใช้ในการเปลี่ยนจากสภาวะ OFF เป็นสภาวะ ON

1.12 ช่วงเวลาเอาต์พุต OFF (output OFF-delay) หมายถึงช่วงเวลาที่หน่วยเอาต์พุตใช้ในการเปลี่ยนจากสภาวะ ON เป็นสภาวะ OFF

1.13 ช่วงแรงดัน/กระแสเอาต์พุต (output voltage/current range)หมายถึงช่วงสัญญาณแรงดันหรือกระแสไฟฟ้าของหน่วยเอาต์พุตแบบอะนาลอก เช่น 0 ถึง 10 VDC หรือ 4 ถึง 200 mADC

1.14 ช่วงแรงดัน/กระแสอินพุต (input voltage/current range) หมายถึงช่วงวัดแรงดันหรือกระแสไฟฟ้าของหน่วยอินพุตแบบอะนาลอก

1.15 ความละเอียด (digital resolution) หมายถึงความละเอียดหรือความแม่นยำในการแปลงสัญญาณของหน่วยอินพุต/เอาต์พุตแบบอะนาลอก ซึ่งขึ้นกับจำนวนบิตของวงจร ADC และ DAC เช่น หน่วยอินพุต/เอาต์พุตแบบอะนาลอกขนาด 8 บิต ที่มีช่วงสัญญาณ 0 ถึง 10 VDC จะมีความละเอียด 39.06 มิลลิโวลต์/บิต

1.16 ฟิวส์เอาต์พุต (output fuse rating) หมายถึงขนาดฟิวส์ที่ใช้ป้องกันการลัดวงจรของหน่วยเอาต์พุต

1.17 ความเป็นฉนวนไฟฟ้า (electrical isolation) หมายถึงความต้านทานหรือเป็นฉนวนไฟฟ้าของหน่วยอินพุต/เอาต์พุต

2. **คุณลักษณะการติดตั้ง** มีลักษณะที่สำคัญดังต่อไปนี้คือ

2.1 ขนาดหน่วยเชื่อมต่อ (point per module) หมายถึงจำนวนอุปกรณ์หน่วยอินพุต/เอาต์พุตสูงสุดที่สามารถเชื่อมต่อกับหน่วยอินพุตหรือเอาต์พุตของแต่ละโมดูล

2.2 ขนาดสาย (wire size) หมายถึงขนาดของเส้นลวดตัวนำที่เชื่อมต่อระหว่าง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หน่วยอินพุตและเอาต์พุตกับอุปกรณ์ภายนอก

2.3 ช่วงอุณหภูมิใช้งาน (ambient temperature rating) หมายถึงอุณหภูมิสูงสุดและต่ำสุดที่หน่วยอินพุต/เอาต์พุตทำงานได้ปกติ ถ้าจำเป็นต้องติดตั้ง PC หรือหน่วยอินพุต/เอาต์พุตในเขตที่อุณหภูมิร้อนจัดหรือหนาวจัดผู้ใช้ต้องจัดหาระบบปรับอุณหภูมิหรือระบบระบายความร้อนให้กับPC

2.4 ความชื้น (humidity rating) หมายถึงความชื้นอากาศสูงสุดที่หน่วยอินพุต/เอาต์พุตทำงานได้ตามปกติ PC ทั่วไปสามารถทำงานได้เมื่ออากาศมีความชื้นสัมพทธ์ 0 ถึง 95 เปอร์เซ็นต์

หน่วยป้อนโปรแกรม

หน่วยป้อนโปรแกรมของ PLC/PC ทำหน้าที่ติดต่อระหว่างผู้ใช้ และ PLC/PC โดยที่ผู้ใช้ทำการป้อนโปรแกรมคำสั่งเข้าสู่หน่วยความจำของ CPU เพื่อเป็นการตรวจสอบสภาพการทำงานการ รับ-ส่งข้อมูลระหว่าง CPU กับหน่วยป้อนโปรแกรมของ PLC/PC แบ่งออกได้หลายชนิดคือ

1. เครื่องป้อนโปรแกรมแบบ CRT ประกอบด้วยจอภาพและแป้นพิมพ์ เป็นแบบที่สะดวกที่สุด แบ่งออกเป็น 2 ชนิดคือ แบบ Dump CRT ซึ่งเป็นแบบที่ไม่มีหน่วยประมวลผลอยู่ภายในตัวการทำงานทุกอย่างจะถูกควบคุมจาก CPU ของ PLC/PC และแบบ Intelligent CRT ซึ่งจะมีหน่วยประมวลผลอยู่ภายใน ตัวการทำงานเป็นอิสระจาก CPU จึงมีประสิทธิภาพดีกว่าแบบแรก

2. เครื่องป้อนโปรแกรมขนาดเล็ก (Mini Programmer) มีขนาดเล็กสามารถเคลื่อนย้ายไปมาได้สะดวก ใช้ LED หรือ LCD ในการแสดงผล เครื่องป้อนโปรแกรมแบบนี้แบ่งออกได้เป็น 2 ชนิด คือ Dump และ Intelligent เช่นเดียวกัน แต่ประสิทธิภาพจะด้อยกว่าเครื่องป้อนโปรแกรมแบบ CRT

3. เครื่องป้อนโปรแกรมลงในหน่วยความจำของ PLC/PC หรือเทป เครื่องป้อนโปรแกรมแบบนี้สามารถใช้ป้อนหรือแก้ไขโปรแกรมคำสั่งของผู้ใช้ เข้าไปในหน่วยความจำของ PLC/PC ได้โดยตรง หรือใช้ในการรับส่งโปรแกรม และข้อมูลต่าง ๆ ระหว่าง PLC/PC และเทป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. เครื่องป้อนโปรแกรมลงในหน่วยความจำ (Memory Burner) เครื่องป้อนโปรแกรมแบบนี้ ทำหน้าที่ถ่ายโปรแกรมที่มีอยู่ลงเก็บไว้ในหน่วยความจำแบบ ROM เพื่อให้โปรแกรมต่างๆ คงอยู่ตลอดเวลาเมื่อโปรแกรมต่าง ๆ ได้ถูกตรวจสอบ และแก้ไขจนเรียบร้อยสมบูรณ์แล้ว

5. คอมพิวเตอร์ PLC/PC บางเครื่องออกแบบมาเพื่อให้ใช้ติดต่อกับเครื่องคอมพิวเตอร์ได้ ทำให้สามารถเขียนโปรแกรมทุกอย่างโดยใช้คอมพิวเตอร์

นอกจากนี้ PLC/PC บางแบบ ยังสามารถใช้ร่วมกับอุปกรณ์เชื่อมต่อพิเศษเพื่อเพิ่มประสิทธิภาพการทำงานของ PLC/PC ให้ดีขึ้น

ปัจจุบัน PLC/PC ได้ถูกพัฒนาให้มีประสิทธิภาพ และขีดความสามารถสูงขึ้น โดยเฉพาะอย่างยิ่ง PC ขนาดเล็กหรือ PLC นั้น ได้ถูกพัฒนาให้มีคำสั่งพิเศษเพิ่มขึ้น ขยายอินพุต/เอาต์พุตได้ หน่วยความจำก็เพิ่มมากขึ้น ต่อเข้ากับอุปกรณ์อื่น ๆ ได้ หรือมีฟังก์ชั่นทางคณิตศาสตร์อย่างง่าย ๆ และที่สำคัญที่สุดก็คือสามารถนำไปต่อใช้ร่วมกับ PC ขนาดใหญ่ได้อีกด้วย

ภาษา PC

คำสั่งที่ใช้เขียนโปรแกรม PC มี 4 ภาษาคือ ภาษาแลดเดอร์ ภาษาบูลลีน ภาษาบล็อก และคำสั่งข้อความภาษาอังกฤษ (English statement language) ซึ่งแต่ละภาษามีวิธีการใช้แตกต่างกัน ภาษาแลดเดอร์และภาษาบูลลีนเป็นภาษาพื้นฐานที่ใช้กับ PC ขนาดเล็กแทนอุปกรณ์หน่วงเวลา และนับจำนวน ในการควบคุมแบบ ON/OFF ภาษาบล็อกและคำสั่งข้อความภาษาอังกฤษเป็นภาษาระดับสูงมักใช้กับการควบคุมที่ซับซ้อนหรือมีการคำนวณทางคณิตศาสตร์
เกี่ยวข้อง เช่น การควบคุมแบบอะนาลอกและการควบคุมตำแหน่ง โดยใช้ร่วมกับภาษาแลดเดอร์และภาษาบูลลีนใน PC ขนาดกลางและขนาดใหญ่ ปัจจุบันภาษาที่ใช้เขียนโปรแกรม PC มีลักษณะดังนี้

1. ภาษาแลดเดอร์เพียงภาษาเดียว
2. ภาษาบูลลีนเพียงภาษาเดียว
3. ภาษาแลดเดอร์และภาษาบล็อก
4. ภาษาแลดเดอร์และคำสั่งข้อความภาษาอังกฤษ
5. ภาษาบูลลีนและภาษาบล็อก
6. ภาษาบูลลีนและคำสั่งข้อความภาษาอังกฤษ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ชุดคำสั่งภาษา PC

คำสั่งในภาษา PC แบ่งเป็น 6 กลุ่มคือ

1. วงจรรีเลย์และการปฏิบัติตรรก
2. การหน่วงเวลาและนับจำนวน
3. การคำนวณทางคณิตศาสตร์
4. การจัดการข้อมูล (data manipulation)
5. การเคลื่อนย้ายข้อมูล และ
6. คำสั่งควบคุมโปรแกรม (control flow instruction)

ภาษาแลดเดอร์

ภาษาแลดเดอร์ประกอบด้วยสัญลักษณ์หน้าสัมผัส มีลักษณะคล้ายวงจรรีเลย์ การเขียนโปรแกรมภาษาแลดเดอร์จากวงจรรีเลย์จึงทำได้ง่าย PC ที่ใช้ภาษาแลดเดอร์คือ PC ขนาดเล็กและขนาดกลางปัจจุบันคำสั่งภาษาแลดเดอร์มีครบทั้ง 6 กลุ่มคำสั่ง คือ วงจรรีเลย์และปฏิบัติตรรก การหน่วงเวลาและนับจำนวน การคำนวณทางคณิตศาสตร์ การจัดการข้อมูล การเคลื่อนย้ายข้อมูล และคำสั่งควบคุมโปรแกรม

โปรแกรมภาษาแลดเดอร์ที่ประกอบขึ้นเพื่อควบคุมอุปกรณ์เอาต์พุต 1 จุด เรียกว่ารังก์ บางครั้งรังก์อาจมีเอาต์พุตมากกว่า 1 จุด แต่เอาต์พุตเหล่านี้จะต้องถูกควบคุมจากสภาวะตรรกจุดเดียวกันเสมอ

โปรแกรมภาษาแลดเดอร์ที่ประกอบด้วยหน้าสัมผัสต่าง ๆ จะทำงานร่วมกันเพื่อส่งสภาวะการควบคุมไปยังอุปกรณ์เอาต์พุต 1 จุด หรือที่เรียกว่ารังก์ บางครั้งโปรแกรมภาษาแลดเดอร์ 1 รังก์ อาจมีอุปกรณ์เอาต์พุตมากกว่า 1 จุด แต่อุปกรณ์เอาต์พุตเหล่านี้ต้องได้รับสภาวะการควบคุมจากจุดเดียวกันเสมอ

สัญลักษณ์ภาษาแลดเดอร์ คำสั่งภาษาแลดเดอร์ประกอบด้วยสัญลักษณ์หน้าสัมผัสและขดลวด เพื่อแสดงเงื่อนไขการควบคุมระหว่างอุปกรณ์อินพุต/เอาต์พุตและอุปกรณ์ภายในการเขียนโปรแกรมต้องระบุตำหรือหมายเลขของอุปกรณ์เหล่านี้ให้ถูกต้องและตรงกันทุกครั้งหมายเลขของหน้าสัมผัสและขดลวดจะสัมพันธ์กับตำแหน่งภายในหน่วยความจำส่วนตารางข้อมูลสัญลักษณ์หน้าสัมผัสหมายถึงการรับสภาวะของหน่วยอินพุต/เอาต์พุตและอุปกรณ์ภายในเพื่อปฏิบัติลอจิกตามเงื่อนไขการควบคุม สัญลักษณ์ขดลวดหมายถึงการส่งคำสั่งหรือผลการควบคุมไปยังหน่วยเอาต์พุตและอุปกรณ์ภายในรูปแบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

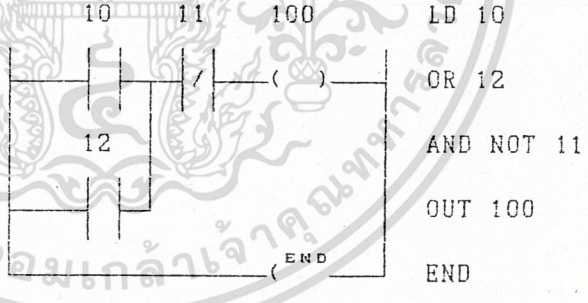
การจัดเรียงหน้าสัมผัสทำให้เกิดสภาวะการควบคุมที่เรียกว่า *สภาวะรังก์* เอาต์พุตแบบปกติดจะทำงานเมื่อรังก์เป็นจริงหรือมีสภาวะ "1" หรือมีเส้นทางผ่านหน้าสัมผัสจะปิดจากด้านซ้ายของรังก์มาที่ขดลวดเอาต์พุต และสิ้นสุดที่ปลายด้านขวาของรังก์เกิดขึ้นอย่างน้อยหนึ่งเส้นทาง เอาต์พุตแบบปกติดับจะหยุดการทำงานเมื่อรังก์เป็นเท็จหรือมีสภาวะเป็น "0" หรือไม่มีเส้นทางเชื่อมต่อระหว่างปลายทั้งสองด้านของรังก์เกิดขึ้น สำหรับเอาต์พุตแบบปกติดับจะมีสภาวะตรงกันข้ามกับเอาต์พุตแบบปกติ ตัวอย่างสัญญลักษณของภาษาแลดเดอร์แสดงในตารางที่ 6

ภาษาบูลีน

ภาษาบูลีนเป็นภาษาพื้นฐานของ PC เช่นเดียวกับภาษาแลดเดอร์คำสั่งในภาษาบูลีนมีลักษณะคล้ายสัญลักษณ์ของนิพจน์บูลีน ซึ่งประกอบด้วย 6 กลุ่มคำสั่งดังตารางที่ 5 และตารางที่ 6 แสดงการเขียนโปรแกรมภาษาแลดเดอร์และภาษาบูลีนเป็นตัวอย่าง

คำสั่งบูลีน	คำสั่งแลดเดอร์
LD/STR	─┤ ├─
LD/STR NOT	─┤/├─
AND	─┤ ├─┤ ├─
AND NOT	─┤ ├─┤/├─
OR	┌─┤ ├─┐
	└─┤ ├─┘
OR NOT	┌─┤ ├─┐
	└─┤/├─┘
OUT	─( )─
OUT NOT	─(/)─
AND	─(ADD)─
SUB	─(SUB)─
.	.

ตารางที่ 5



ตารางที่ 6

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาษาบล็อก

ภาษาบล็อกมักใช้ร่วมกับภาษาแลดเดอร์ในการควบคุมที่ค่อนข้างซับซ้อนหรือมีข้อมูลที่เป็นตัวเลขเกี่ยวข้อง เช่นการควบคุมที่มีการคำนวนทางคณิตศาสตร์และการควบคุมตำแหน่งเครื่องจักร รูปข้างล่างนี้แสดงลักษณะของคำสั่งบล็อกซึ่งประกอบด้วยคำสั่งหรือสัญญาณควบคุมใช้รับสภาวะควบคุมจากรังก์หรือหน้าสัมผัสของหน่วยอินพุต/เอาต์พุต และอุปกรณ์ภายใน สัญญาณเอาต์พุตหรือหน้าสัมผัสของคำสั่งบล็อกซึ่งทำหน้าที่ควบคุมและแสดงผลการปฏิบัติตามคำสั่ง ข้อมูลหรือรีจีสเตอร์ที่ใช้เก็บข้อมูลหรือผลลัพธ์จากการปฏิบัติคำสั่งบล็อก

รูปแสดงคำสั่งบล็อก

ภาษาบล็อกแบ่งออกเป็น 4 กลุ่มคือ

- คำสั่งหน่วงเวลาและนับจำนวน
- การคำนวนทางคณิตศาสตร์
- การจัดการข้อมูล
- การเคลื่อนย้ายข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

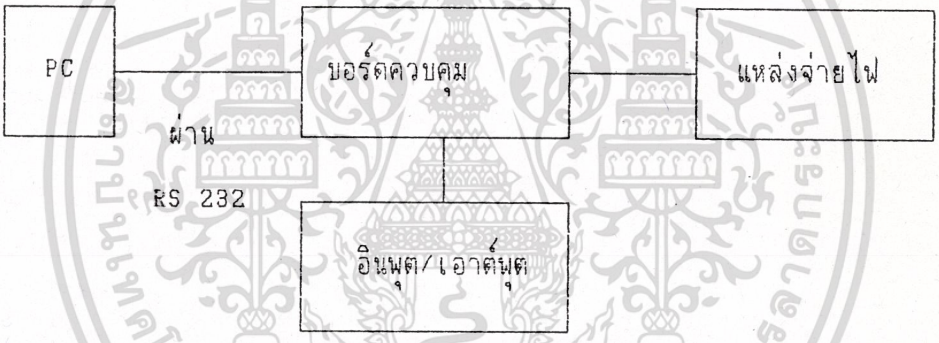
บทที่ 3

การออกแบบและการสร้าง

โครงงานนี้เป็นการสร้าง PLC ขนาดเล็ก โดยป้อนคำสั่งเป็นภาษา แลดเดอร์ (Ladder Diagram) ที่เขียนคำสั่งจากเครื่อง PC แล้วส่งไปยังบอร์ดควบคุม โครงงานนี้แบ่งการทำงานออกเป็นสองส่วนดังนี้

การทำงานทางด้าน Hardware

HARDWARE ประกอบด้วย บล็อกดังรูป



บล็อกแสดงส่วนประกอบของโครงงาน

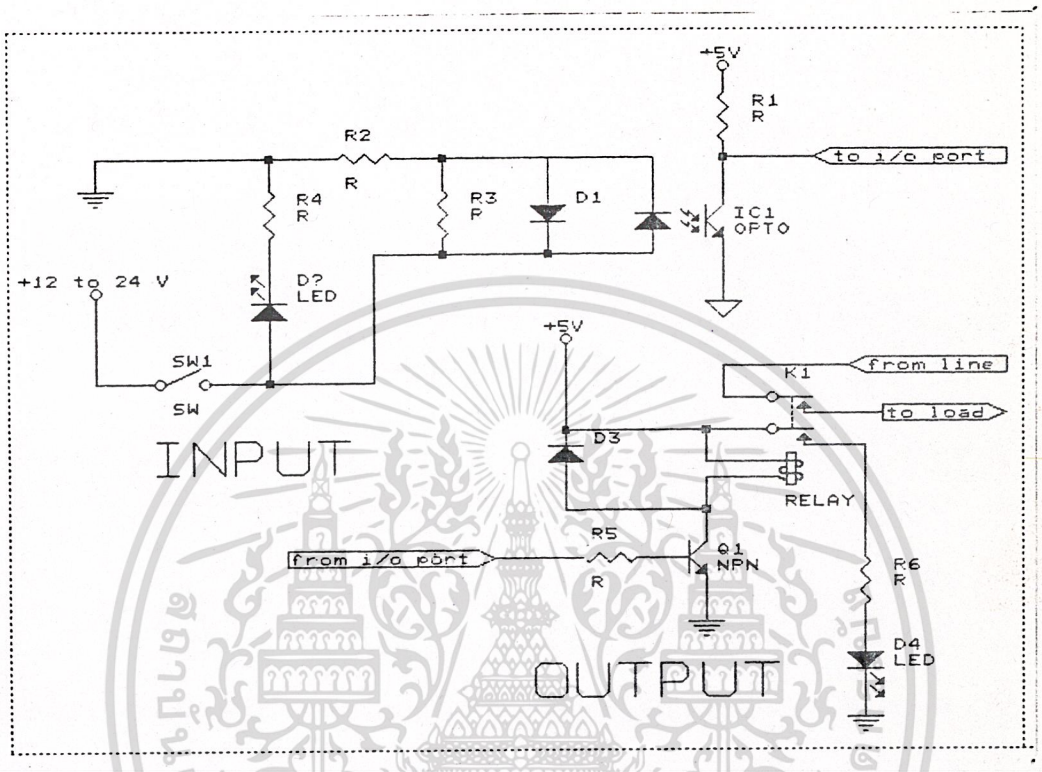
การทำงานของแต่ละส่วนดังนี้

บอร์ดควบคุมจะรับสัญญาณจากหน่วยอินพุต ซึ่งสัญญาณอินพุตมีอยู่ทั้งหมด 8 จุด (สามารถขยายได้อีก) เข้าไปประมวลผลตามโปรแกรมที่ส่งมาจาก PC แล้วส่งผลการประมวลผลนั้นไปออกที่หน่วยเอาต์พุต ซึ่งมี 8 จุด (สามารถขยายได้อีก)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การออกแบบด้านอินพุต

หน่วยอินพุต ใช้ Opto Isolater เพื่อแยกกันทางไฟฟ้าระหว่างอินพุตกับเอาต์พุตแล้วส่งลอจิกที่ได้ไปให้บอร์ดควบคุม หน่วยอินพุต/เอาต์พุตมีวงจรดังรูป



รูปแสดงส่วนอินพุต/เอาต์พุต

การออกแบบส่วนอินพุต

LED ด้านอินพุต มี I_F = 10 mA. V_F = 1.5 V. ตามที่ระบุในคู่มือ เราจะหาค่าอุปกรณ์ต่าง ๆ ได้ดังนี้

R_3

ให้แรงดัน V_F จะให้ LED = 1.5 V.

เราให้ค่า R_2 = 1.2K (IF = 10 mA.) R3 จะมีค่า

จาก $V_F = (R_3 \cdot 12)/(R_2+R_3)$

$R_3 = (V_F.R_2/12) / (1-V_F/12)$

แทนค่าจะได้ R_3 = 171 ohm (เราใช้ 200 ohm)

R_4

ความต้านทานของ LED ทางด้าน Forward ประมาณ 100 ohm เราให้กระแสไหลผ่าน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

10 mA.

จาก I=V/R

$R=12/10x10^{-3}$

จะได้ R_4 มีค่า 1.2 K.

D_1 ใส่ไว้เพื่อป้องกัน การจ่ายแรงดันผิดขั้วอันจะเป็นเหตุทำให้ Opto เสียหาย

D_2 ใส่ไว้เพื่อแสดงสภาวะของอินพุต ว่า ON หรือ OFF

Transister ทางเอาต์พุต มี $V_{CEO(MAX)}$ = 80 V. $I_{C(MAX)}$ = 150 mA.

เราให้ I_C ไหลผ่าน Transister ประมาณ 10 mA.

R_1 = $12/10x10^{-3}$ =12.0

ทางด้านเอาต์พุตประกอบด้วยอุปกรณ์ดังนี้

- Opto Isolator ของ Motorola เบอร์ H11A4
- R_1 = 1 Kohm
- R_2 = 1.2 Kohm
- R_3 = 200 Ohm
- R_4 = 1.2 Kohm
- D_1 = 1N4001
- D_2 = LED

การออกแบบทางด้านเอาต์พุต

ทางด้านเอาต์พุตเราใช้ Relay เพื่อให้จ่ายกระแสได้สูงสุด 5 แอมป์ โดยใช้ Transister ขับให้คอยล์ของ Relay ทำงาน Transister จะรับกระแสที่ขาเบส มาจากส่วนบอร์ดควบคุม

คอยล์ รีเลย์ มีความต้านทานประมาณ 200 ohm ดังนั้นเราจึงเลือกใช้ Transister ที่มีกระแส I_C สูงสุดประมาณ 1 แอมป์ (5/200) เราจะเลือกใช้ Transister เบอร์ 2N3904 มี PD ประมาณ 1/2 W. ใช้ R1 จำกัดกระแสเบสมีค่า 1K

- Relay 5 v 2 Contact

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- Transister 2N3904

- R_1 = 1Kohm

บอร์ดควบคุม

บอร์ดควบคุมประกอบด้วย CPU 8 บิท ตระกูล MCS-51 อาจใช้ได้ตั้งแต่ 8031 8032 เป็นต้น ใช้หน่วยความจำแบ่งเป็นส่วนของโปรแกรม Monitor 32 Kbyte และส่วนของ RAM 32 Kbyte 8255 เป็น อินพุต เอาต์พุตพอร์ต ซึ่ง MCS-51 มีคุณสมบัติคร่าว ๆ ดังนี้

โครงสร้างทั่วไปของไมโครคอนโทรลเลอร์ เบอร์ 8031

1.โครงสร้างภายนอก ประกอบด้วย

1. อินพุต/เอาต์พุตพอร์ต (Input/Output Port)มีทั้งหมด 4 พอร์ คือ พอร์ต0, พอร์ต1,พอร์ต2 และ พอร์ต3 แต่ละพอร์ตประกอบด้วย 8 บิทโอเพ่นเดรน (Opendrain) 2 ทิศทาง นอกจากนี้ พอร์ต0 และพอร์ต2 ยังร่วมกันทำหน้าที่เป็นแอดแดรส (Address) ขนาด 16 บิท โดยการมัติเพล็ก (Multiplex) และ แต่ละขาของพอร์ต3 มีหน้าที่พิเศษ (Alternative Function) ดังแสดงไว้ในภาคผนวก

2. รีเซ็ต(Reset) หรือ อาร์เอสที(RST) เมื่อต้องการรีเซ็ตต้องให้โลจิก 1 เป็นเวลา 2 แม็ชชีนไซเคิล (Machine Cycle)

3. เอเอลอี (ALE) หรือ แอดแดรสแล็ทช์อีนาเบิล (Address Latch Enable) ใช้ในการแล็ทช์แอดแดรสไบท์ต่ำ ระหว่างที่มีการเฟ็ทช์(Fetch) โปรมแกรมภายนอก (External Program Memory) สัญญาณเอเอลอีจะเกิดเป็นพัลซ์(Pulse) จำนวน 2 พัลซ์ ทุกๆแม็ชชีนไซเคิล

4. พีเอสอีเอ็น (PSEN) เป็นสัญญาณการอ่าน (Read Strobe) ใช้ควบคุมการอ่านโปรแกรมภายนอก หรืออีพรอม (EPROM) สัญญาณนี้จะเกิดขึ้น 2 พัลซ์ทุกๆแม็ชชีนไซเคิล

5. อีเอ (EA) เป็นสัญญาณควบคุมให้ซีพียูทำการเฟ็ทช์โปรแกรมภายใน (Internal Program Memory) ซึ่งซีพียูเบอร์ 8031 ที่ใช้ ไม่มีรอมภายในจึงต่อขานี้ลงกราวด์ (Ground)

6. เอ็กซ์ทีเอเอล 1 และ เอ็กซ์ทีเอล 2 (XTAL1,XTAL2) เป็นอินพุตและเอาต์พุตของอินเวอติ้งออสซิลเลเตอร์ (Inverting Oscillator) ใช้ต่อกับส่วนสร้างความถี่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. โครงสร้างภายใน ประกอบด้วย

1. แอ็คคิวมูเลเตอร์รีจิสเตอร์ (Accumulater Register) ใช้เก็บข้อมูลที่ได้จากการกระทำทางตรรก และโอเปอเรชั่น (Operation) ต่างๆยกเว้นการคูณและหาร

2. บีรีจิสเตอร์ (B Register) ใช้เก็บข้อมูลจากการคูณหาร

3. โปรแกรมสเตตัสเวิรด์ (Program Status Ward)เป็นรีจิสเตอร์แสดงสถาวะของโอเปอเรชั่นต่างๆหรือแฟล็กรีจิสเตอร์ (Flag Register)

4. สแต็กพอยน์เตอร์ (Stack Pointer) ใช้ในการเก็บดาต้าพอยน์เตอร์ (Data Pointer) ขณะทำการพุช (Push) คอล (Call) หรือเกิดการอินเทอรัพท์ (Interrupt) โดยค่าสแต็กพอยน์เตอร์จะเพิ่มก่อนการกระทำดังกล่าวและค่านี้เป็นค่าทางฮาร์ดแวร์ (Hardware) เริ่มที่ค่า 07H (แอดแดรสของแรมภายใน)แต่การใช้จะเริ่มที่แอดแดรส 08H

5. ดาต้าพอยน์เตอร์ (Data Pointer) เป็นตัวชี้แอดแดรส

6. ไทม์เมอร์เคาน์เตอร์ (Timer/Counter) จะใช้เป็นไทม์เมอร์หรือเคาน์เตอร์ขึ้นอยู่กับการกำหนดค่าในรีจิสเตอร์ทีเอ็มโอดี (TMOD) กรณีเป็นไทม์เมอร์จะใช้ความถี่ของซีพียูหารด้วย 12 กรณีเป็นเคาน์เตอร์จะใช้ขาที่ 0 และขาที่ 1 (T0,T1) ของพอร์ต3 เป็นขาอินพุต

7. ซีเรียลดาต้าบัฟเฟอร์ (Serial Data Buffer)เป็นบัฟเฟอร์ในการส่งและรับข้อมูลทางซีเรียลพอร์ต (Serial Port)

8. คอนโทรลรีจิสเตอร์ต่างๆ (Control Register) เช่น ไอพี(IP),ไออี(IE), ทีเอ็มโอดี(TMOD),ทีซีโอเอ็น(TCON),เอสซีโอเอ็น(SCON) และพีซีโอเอ็น (PCON) ซึ่งมีรายละเอียดดังนี้

TMOD Timer 1 Timer2

Gate	C/T	M1	M0	Gate	C/T	M1	M0

MSB LSB

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Gate ถ้ามีค่าเป็น 0 จะเป็นตัวที่จะทำให้ Timer หรือ Counter ทำงานได้

C/T เป็นตัวเลือกว่าจะใช้งานเป็น Timer หรือ Counter โดย 0 เป็น Timer และ 1 เป็น Counter

M1,M2 เป็นตัวเลือกโหมดการทำงาน

M1,M2=00 เป็นการใช้ Timer/Counter ขนาด 13 บิท

M1,M2=01 เป็นการใช้ Timer/Counter ขนาด 16 บิท

M1,M2=10 เป็นการใช้ Timer/Counter ขนาด 8 บิท โดยเป็นลักษณะเรียกค่าซ้ำเข้า มาโดยอัตโนมัติ (Auto-reload)

M1,M2=11 ใน (Timer0) เป็นการใช้ Timer,Counter ขนาด 8 บิท โดย TL0 จะ ถูกควบคุมตามมาตรฐาน ส่วน TH0 จะถูกควบคุมจาก TR1 ใน (Timer 1)จะ ถูกหยุดใช้

IE

EA	X	X	ES	ET1	EX1	ET0	EX0

MSB LSB

EA เป็นการควบคุมการรับ INT ของทั้งระบบ ถ้า EA=0 จะไม่รับ INT ถ้า EA=1 การรับ INT จะขึ้นอยู่กับการพร้อมที่จะรับ INT ของแต่ละตัว ซึ่งจะเป็นอิสระต่อกัน

ES เป็นตัวควบคุมการรับ INT ของ Serial Port

ET1 เป็นตัวควบคุมการรับ INT ของ Time1

EX1 เป็นตัวควบคุมการรับ INT ของ INT1

ET0 เป็นตัวควบคุมการรับ INT ของTimer1

EX0 เป็นตัวควบคุมการรับ INT ของ INT0

X สำรอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

IP

X	X	X	PS	PT1	PX1	PT0	PX0

MSB LSB

PS เป็นการทำให้ INT ของ Serial Port มีความสำคัญกว่าอันอื่น

PT1 เป็นการทำให้ INT ของ Timer1 มีความสำคัญกว่าอันอื่น

PX1 เป็นการทำให้ INT ของ INT1 มีความสำคัญกว่าอันอื่น

PT0 เป็นการทำให้ INT ของ Timer0 มีความสำคัญกว่าอันอื่น

PX0 เป็นการทำให้ INT ของ INT0 มีความสำคัญกว่าอันอื่น

X สำรอง

SCAN

SM0	SM1	SM2	REN	TB8	RB8	T1	R1

MSB LSB

SM0,SM1 เป็นตัวเลือกโหมดควบคุม

00 โหมด 0 เป็น Shift Regiter มีอัตราการส่ง = Fosc/12

01 โหมด 1 เป็นการส่ง 8 บิท มีอัตราการส่งที่แปลงค่าได้

10 โหมด 2 เป็นการส่ง 9 บิท มีอัตราการส่ง =Fosc/64 หรือ Fosc/32

11 โหมด 3 เป็นการส่ง 9 บิท มีอัตราการส่งที่แปรค่าได้

SM2 เป็นตัวทำให้มีการติดต่อแบบ Multiprocessor ได้ ในโหมด1,2,3 ถ้า SM2 ถูก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เซ็ตเป็น 1 จะทำให้ R1 ไม่สามารถทำงาน ในโหมด0 SM2 ควรเป็น0

REM เป็นตัวทำให้สามารถรับส่งแบบ Serial ได้ ถ้าเซ็ตตัวมัน จะสามารถรับส่งได้ ถ้าเคลียร์ตัวมันจะไม่สามารถส่งข้อมูลได้

TB8 เป็นบิทที่ 9 ของข้อมูลที่จะส่งไปในโหมด 2 หรือ 3

RB8 เป็นบิทที่ 9 ของข้อมูลที่จะรับมาในโหมด 2 หรือ 3 แต่ในโหมด 1 จะเป็นค่าของ SLop บิท และในโหมด 0 จะไม่ถูกใช้

T1 เป็นตัวตรวจว่ามีการรับ INT จากการส่งข้อมูล โดยมันจะถูกเซ็ตเมื่อการส่งเสร็จสมบูรณ์ และจะถูกเคลียร์ทางซอฟแวร์

R1 เป็นตัวตรวจว่ามีการรับ INT จากการรับข้อมูล โดยมันจะถูกเซ็ตเมื่อการรับเสร็จสมบูรณ์ และจะถูกเคลียร์ทางซอฟแวร์

TCON

TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0

MBS LBS

TF1 เป็นแฟลท overflow ของ Timer/Counter การเซ็ตและเคลียร์ถูกทำทางฮาดร์แวร์

TR1 เป็นตัวควบคุมตัว Timer1 ให้ทำงานหรือหยุดทำงาน การเซ็ตและการเคลียร์ถูกทำทางซอฟแวร์

TF0 เป็นแฟลท overflow ของ Timer/Counter0

TR0 เป็นตัวควบคุมตัว Timer0

IE1 เป็นตัวตรวจว่ามีการรับสัญญาณ INT1 หรือไม่

IT1 เป็นบิทที่กำหนดการรับสัญญาณ INT1 ว่าจะรับช่วงขาลงหรือช่วงสถานะเป็น0 ของสัญญาณ INT1

IE0 เป็นตัวตรวจว่ามีการรับสัญญาณ INT0 หรือไม่

IT0 เป็นบิทที่กำหนดการรับสัญญาณ INT0 ว่าจะรับช่วงขาลงหรือช่วงสถานะเป็น 0 ของสัญญาณ INT0

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

PSW

CY	AC	F0	RS1	RS0	OV	X	P

MSB LSB

CY เป็นแคร์รี่แฟลก (Carry Flag)

AC เป็นอ็อกซิลารี่แฟลก (Auxillary Flag) สำหรับหารทำงานแบบบีซีดี (BCD)

F0 เป็นแฟลกศูนย์ (Flag0)

RS1 เป็นบิทที่ใช้เลือกรีจิสเตอร์แบงค์ (Register Bank) สามารถตั้งและลบด้วยซอฟแวร์

RS0 แสดงการทำงานของรีจิสเตอร์แบงค์

OV แสดงโอเวอร์โฟว์แฟลก (Overflow Flag)

X สำรอง

P แสดงพาริตี้แฟลก (Parity Flag) ตั้งและลบโดยทางฮาร์ดแวร์

หมายเหตุ รูปแบบการตั้งการทำงานของรีจิสเตอร์แบงค์ โดยการตั้ง (RS0,RS1) ดังต่อไปนี้

(0,0) เลือกการทำงานช่วง (00h-07h) (0,1) เลือกการทำงานช่วง (08h-0Fh)

(1,0) เลือกการทำงานช่วง (10h-17h) (1,1) เลือกการทำงานช่วง (18h-1Fh)

SERIAL INTERFACE

- เป็น Full Duplex สามารถรับส่งข้อมูลได้พร้อมกัน
- มี Recrive Buffer คือ 8051 สามารถรับ byte ที่ 2 ได้ก่อนที่ byte แรก จะถูอ่านไปจาก Recrive Register
- Serial Port Recrive และTransmit Register จะ Accecsed ที่ Special Function Register SBUF
- Serial port ทำงานเป็น 4 โหมด
- Mode 0 : baud rate เป็น 1/12 ของ OSC. Frequency.

ข้อมูลอนุกรมจะเข้า/ออกทาง RXD และ TXD ในลักษณะ Shift clock Data จะเป็น 8 บิท

- Mode 1 : จะทำการรับส่งขนาด 10 บิท เป็น Start bit (bit0) Data bit 8 bit (LSB เริ่มก่อน) และ Stop bit ขณะ Recrive Stop bit จะไปยัง RB8 ใน SFR คือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

SCON baud rate เปลี่ยนแปลงได้

- Mode 2: รับ-ส่ง ขนาด 11 bit เป็น Start bit(0),8 bit Data,bita เป็น Programable bit และเป็น Stop bit(1) ในขณะ Transmitt bit9 (TB8 in SCON) สามารถจะกำหนดค่าเป็น 0 หรือ 1 ได้เช่น ใช้เป็น Parity bit ซึ่งกำหนดใน PSW และในขณะ Recrive bit 9 bit นี้จะเข้ามายัง RB8 ของ SCON(ใน SFR) โดยไม่สนใจ Stop bit band rate สามารถโปรแกรมเป็น 1/32 หรือ 1/64 ของ OSC.
- Mode 3: รับ-ส่ง ขนาด 11 บิท เหมือน Mode 2 ทุกประการ แต่ band rate สามารถปรับได้ในทั้ง 4 โหมด การส่งเริ่มโดยคำสั่งใดๆก็ตามที่ใช้ SBUF ซึ่งจะเหมือน Register ปลายทาง การตั้งโหมด 0 โดยให้ RI=0 และ REN=1 ส่วนโหมดอื่น ให้ REN=1

Multiprocessor communication

Mode2 และ 3 สามารถใช้ในงานนี้ บิท 9 จะถูกรับเข้าไปยัง RB8 และตามมาด้วย Stop bit,Serial port จะInterrupt เมื่อ RB8 เป็น 1 ลักษณะเช่นนี้จะไป Enable โดย Set bit SM2 ใน SCON ลักษณะเช่นนี้จึงสามารถนำไปใช้ในงานแบบ Micprocessor ได้ โดยเมื่อ Master processor ต้องการส่ง block ข้อมูลไปยัง slave processor โดยอันดับแรกจะส่ง Address byte ซึ่งจะเจาะจงว่าตัว slave เป็นตัวไหน โดย address bute จะแตกต่างจาก data byte ตรงที่ bit ที่ 9 โดยถ้าเป็น 1 เป็น address byte และเป็น 0 เป็น data byte โดย clock จาก SM2 จะไม่มีผลในโหมด 0 ในโหมด 1 สามารถใช้ SM2 check ความสมบูรณ์ของ stop bit

Serial port control register

Register control อยู่ใน SFR ที่ SCON

(MSB) (LSB)

SM0	SM1	SM2	REN	TB8	RB8	T1	R1

SM0 และ SM1 จะเลือก Serial port mode

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

SM0	SM1	MODE	DISCRIPTION	BAND RATE	
0	0	0	shift register	fosc/12	
0	1	1	8- bit uart	Variable	
1	0	2	9- bit uart	fosc/64	fosc/32
1	1	3	- bit UART Variable	Variable	

SM2 : Enable multiprocessor ใน Mode 2 และ 3 ถ้า SM2=1 แล้ว R1 จะไม่ Active ถ้ารับ bit ที่9 (RB8) เป็น 0 ในโหมด 1 ถ้า SM2=1 แล้ว R1 จะไม่ Active ถ้า stop bit ไม่ได้รับเข้ามา ในโหมด 0 SM2 ควรเป็น 0

REN : Enable การรับ set โดย software ถ้าเป็น 1 จะ Enable เป็น0 จะ Disable

TB8 : จะส่ง bit9 เป็นอะไร set ที่นี้ โดย software ในโหมด2 และ 3

RB8:ในโหมด 2 และ 3 bit9 จะรับไว้ที่นี้ ในโหมด 1 ถ้า RB8=0 แสดงว่าได้รับ stop bit ใน Mode0 จะไม่ใช้

R1 : จะรับ Interropt flag set โดย hardware เมื่อรับข้อมูลครบ 8 บิท ในโหมด 0 R1 ก็จะถูก set เมื่อให้สามารถแยกข้อมูลออกจากกันได้ หรือในโหมดอื่น เมื่อรับ stop bit แล้วก็จะเซ็ตเป็น 1 เช่นกัน ซึ่งต้องเคลียร์โดย software

Band Rate

ในโหมด 0 band rate จะคงที่ = OSC.frcq./12 band rate ในโหมด 2 ขึ้นอยู่กับค่า bit SMOD ใน SFR PCON ถ้า SMOD = 0 band rate เป็น 1/64 ของ OSC.freq. ถ้า SMOD=1 band rate เป็น 1/32 ของ OSC.freq.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เพราะฉนั้น mode 2 band rate = 2SMOD/64 * (OSC.freq.)

ใน 8051 band rate ใน mode 1 และ 3 หาได้จาก Timer 1overflow rate

ใช้ Timer1 เป็นตัวจ่าย band rate

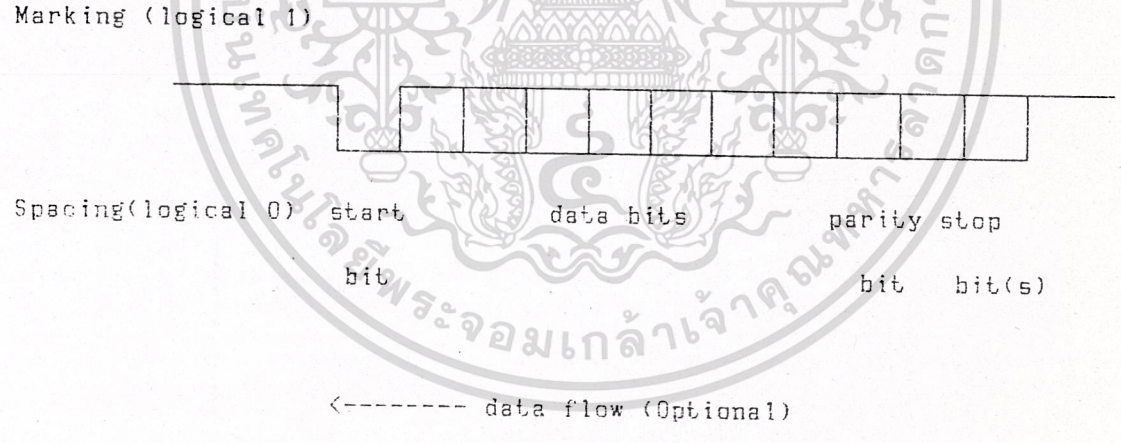
band rate mode 1,3 = 2SMOD/32 * (Timer1 overflow rate)

ในการใช้ Timer 1 เป็นตัว Generate band rate ควรจะ Disable Interrupt ของ Timer 1 และตั้ง configutation ของ Timer 1 เป็น Timer แบบ Auto reload (TMOD =0010B) ในกรณีนี้ band rate จะมีสูตรเป็น

band rate mode 1,3 = (2SMOD/32) * (OSC.freq./<12*<256-<TH1>>>)

การทำงานของพอร์ตอนุกรมอาร์เอส 232 ซี

การติดต่อเป็นแบบอนุกรมอะซิงโครนัส (Asynchronous) มีรูปแบบของสัญญาณดังรูป



รูปแสดงสัญญาณแบบอะซิงโครนัส

สำหรับอุปกรณ์ติดต่อแบบอนุกรมอะซิงโครนัส ใช้ยูเออาร์ที หรือ UART (Universal Asynchronous Receiver Transmitter) ในส่วนของไมโครคอมพิวเตอร์(IBM PC) ใช้ไอซี 8250 อะซิงโครนัสคอมมูนิเคชั่น(Asynchronous Communications Element) ก่อนการใช้งาน UART ต้องทำการกำหนดบอดเรท จำนวนบิทของข้อมูล รูปแบบพาริตี้(Parity) และจำนวนของบิทหยุด (stop bit) เพื่อกำหนดรูปแบบการรับและส่งข้อมูลเมื่อมีการส่งข้อมูล ต้องทำการกำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หนด UART ถ้าทรานสมิตเตอร์โฮลดิ้งรีจิสเตอร์ (Transmitter Holding Register) ว่าง สามารถส่งข้อมูลได้จนสิ้นสุดการส่งซึ่งเอาต์พุตไลน์ (output line) มีลอจิก 1 ข้อมูลของทรานสมิตเตอร์โฮลดิ้งรีจิสเตอร์จะถูกส่งผ่านทรานสมิตเตอร์ชิฟรีจิสเตอร์ (Transmitter Shift Register) โดนเพิ่มส่วนของบิทเริ่มต้น บิทหยุด และพาริตี้บิทจึงส่งไปยังสายส่งต่อไปเมื่อทำการรับข้อมูล จะเก็บในรีซีฟเวอร์ชิฟรีจิสเตอร์ (Receiever Shift Register) และส่งข้อมูลไปที่รีซีฟเวอร์ดาต้ารีจิสเตอร์ (Recever Data Register) เมื่อ UART แสดงสถานะพร้อมไมโครคอมพิวเตอร์จะรับข้อมูลไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

การทดลองและผลการทดลอง

การทำงานของโครงงานคร่าว ๆ

เพื่อความสะดวกในการทำความเข้าใจ เราจะแบ่งการทำงานออกเป็นคร่าว ๆ ดังนี้

1. เราจะสั่งงานให้บอร์ดควบคุมทำงานได้โดยสั่งงานผ่านเครื่อง PC คือเราจะเขียนคำสั่งเป็น แลดเดอร์ไดอะแกรม (เราเขียนได้ในรูปของภาพกราฟฟิก)

2. เมื่อเราเขียนคำสั่งเสร็จเรียบร้อยแล้วเราจะทำการแปลงคำสั่งที่เป็นรูปภาพกราฟฟิกของแลดเดอร์ไดอะแกรมบนจอของ PC ในการแปลงคำสั่ง 1 ครั้งเราจะได้ File 2 File คือ File ที่เป็นคำสั่งภาษาแอสแซมบลี้(Assembly)ของ CPU ตระกูล MCS-51 เป็น File นามสกุล .C51 File ที่ 2 เป็น File เลขฐาน 16 ที่มีรูปแบบ Intel Format เป็น File นามสกุล .HEX

3. เมื่อแปลงคำสั่งเรียบร้อยแล้วก็ส่งคำสั่งที่ได้ไปยังบอร์ดควบคุม โดยโปรแกรมจะส่ง File

4. หน่วยอินพุตจะรับสภาวะที่ด้านอินพุตของหน่วย และส่งไปยังบอร์ดควบคุม ให้บอร์ดควบคุมประมวลผลตามโปรแกรม กล่าวคือ รับเงื่อนไขจากหน่วยอินพุตเข้าไปประมวลผลตามโปรแกรม

5. หน่วยเอาต์พุตจะคงสภาวะ ON หรือ OFF ไปขับโหลดตามคำสั่งที่ส่งมาจากบอร์ดควบคุม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาษาที่ใช้ในโครงงานนี้เป็นภาษาแลดเดอร์มีความหมายของแต่ละคำสั่งดังต่อไปนี้

วงจรรีเลย์และปฏิบัติตรรก ทำหน้าที่แทนวงจรรีเลย์ในการควบคุมแบบ ON/OFF คำสั่งกลุ่มนี้ประกอบด้วย

สัญลักษณ์ ---] [---

คำสั่ง หน้าสัมผัสแบบปกติเปิด (normally-open contact)

หน้าสัมผัสแบบปกติเปิดทำหน้าที่รับสภาวะของหน่วยอินพุต/เอาต์พุตและอุปกรณ์ภายในถ้าหน้าสัมผัสแบบปกติเปิดมีสภาวะ "1" หรือ ON หมายถึงการปิดวงจรไฟฟ้า ถ้าหน้าสัมผัสแบบปกติเปิดมีสภาวะ "0" หรือ OFF หมายถึงการเปิดวงจรไฟฟ้าหรือตัดวงจรไฟฟ้า

สัญลักษณ์ ---]/[---

คำสั่ง หน้าสัมผัสแบบปกติปิด (normally-close contact)

หน้าสัมผัสแบบปกติปิดทำหน้าที่รับสภาวะตรงข้ามของหน่วยอินพุต/เอาต์พุตและอุปกรณ์ภายในถ้าหน้าสัมผัสแบบปกติปิดมีสภาวะ "1" หรือ ON หมายถึงการเปิดหรือตัดวงจรไฟฟ้า ถ้าหน้าสัมผัสแบบปกติปิดมีสภาวะ "0" หรือ OFF หมายถึงการปิดวงจรไฟฟ้า

สัญลักษณ์ ---( )---

คำสั่ง ขดลวดแบบปกติติด (energize coil)

ขดลวดแบบปกติติดทำหน้าที่ส่งสภาวะการควบคุมไปยังหน่วยเอาต์พุตและอุปกรณ์ภายใน ถ้ารังก์มีสภาวะ "1" หรือ ON ขดลวดแบบปกติดจะทำงาน ถ้ารังก์มีสภาวะ "0" หรือ OFF ขดลวดแบบปกติติดจะหยุดทำงาน เมื่อขดลวดแบบปกติติดมีสภาวะ "1" หน้าสัมผัสแบบปกติเปิดที่ตำแหน่งเดียวกันจะมีสภาวะ "1" เมื่อขดลวดแบบปกติติดมีมีสภาวะ "0" หน้าสัมผัสแบบปกติเปิดจะมีสภาวะ "0" สำหรับหน้าสัมผัสแบบปกติปิดจะมีสภาวะตรงข้ามกับหน้าสัมผัสแบบปกติเปิด

สัญลักษณ์ ---(/)---

คำสั่ง ขดลวดแบบปกติดับ(de-energize coil)

ขดลวดแบบปกติดับทำหน้าที่ส่งสภาวะการควบคุมไปยังหน่วยเอาต์พุตและอุปกรณ์ภายในถ้ารั้งก์มีสภาวะ "1" หรือ ON ขดลวดแบบปกติดับจะหยุดทำงาน ถ้ารั้งก์มีสภาวะ "0" หรือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

OFF ขดลวดแบบปกติดับจะทำงาน เมื่อขดลวดแบบแบบปกติดับมีสภาวะเป็น "1" หน้าสัมผัสแบบปกติเปิดที่ตำแหน่งเดียวกันจะมีสภาวะเป็น "0" เมื่อขดลวดแบบแบบปกติดับมีสภาวะเป็น "0" หน้าสัมผัสแบบปกติเปิดที่ตำแหน่งเดียวกันจะมีสภาวะเป็น "1" สำหรับหน้าสัมผัสแบบปกติปิดจะมีสภาวะตรงกันข้ามกับหน้าสัมผัสแบบปกติเปิด

สัญลักษณ์ ---(L)---

คำสั่ง ขดลวดรักษาสภาพติด(Latch Coil)

ขดลวดรักษาสภาพติดทำหน้าที่รักษาสภาวะ "1" ของอุปกรณ์หน่วยเอาต์พุตหรืออุปกรณ์ภายในถ้ารังก็มีสภาวะ "1" ขดลวดรักษาสภาพติดจะเริ่มทำงาน และจะรักษาสภาวะ "1" ไว้จนมีคำสั่งยกเลิก

สัญลักษณ์ ----(U)---

คำสั่ง ยกเลิกขดลวดรักษาสภาพติด(Unlath Coil)

คำสั่งยกเลิกขดลวดรักษาสภาพติดใช้เพื่อยกเลิกการใช้ขดลวดรักษาสภาพติด ทำให้อุปกรณ์หน่วยเอาต์พุตและอุปกรณ์ภายในมีสภาวะเช่นเดียวกับสภาพปกติของรังก์

ตัวอย่างการใช้ขดลวดรักษาสภาพติด

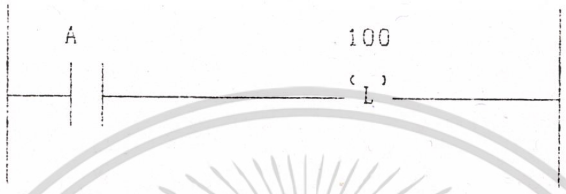
จากรูปเมื่อหน้าสัมผัส A ปิดวงจรไฟฟ้า ขดลวดหมายเลข 100 จะมีสภาวะ ON และจะคงสภาพนี้ไว้จนกระทั่งหน้าสัม B ปิดวงจรไฟฟ้าเพื่อยกเลิกการทำงานของขดลวดรักษาสภาพติด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สัญลักษณ์ ---]↑[---

คำสั่ง หน้าสัมผัสตรวจการเปลี่ยนสภาวะ OFF-ON (OFF-ON Transition Contact)

หน้าสัมผัสตรวจการเปลี่ยนสภาวะ OFF-ON ทำหน้าที่ตรวจสอบการเปลี่ยนสภาวะ OFF สภาวะON ของรังก์หน้าสัมผัสจะมีสภาวะ "1" หรือ ON นานเกิน 1 ช่วงการสแกนของ PC เมื่อตรวจสอบพบการเปลี่ยนสภาวะของรังก์



ตัวอย่างการใช้หน้าสัมผัสตรวจการเปลี่ยนสภาวะ OFF-ON

จากรูปเมื่อหน้าสัมผัส A มีการเปลี่ยนจากสภาวะ OFF สู่สภาวะ ON ขดลวดหมายเลข 100 จะมีสภาวะ ON และจะคงสภาพไว้จนมีคำสั่งยกเลิก

สัญลักษณ์ ---]↓[---

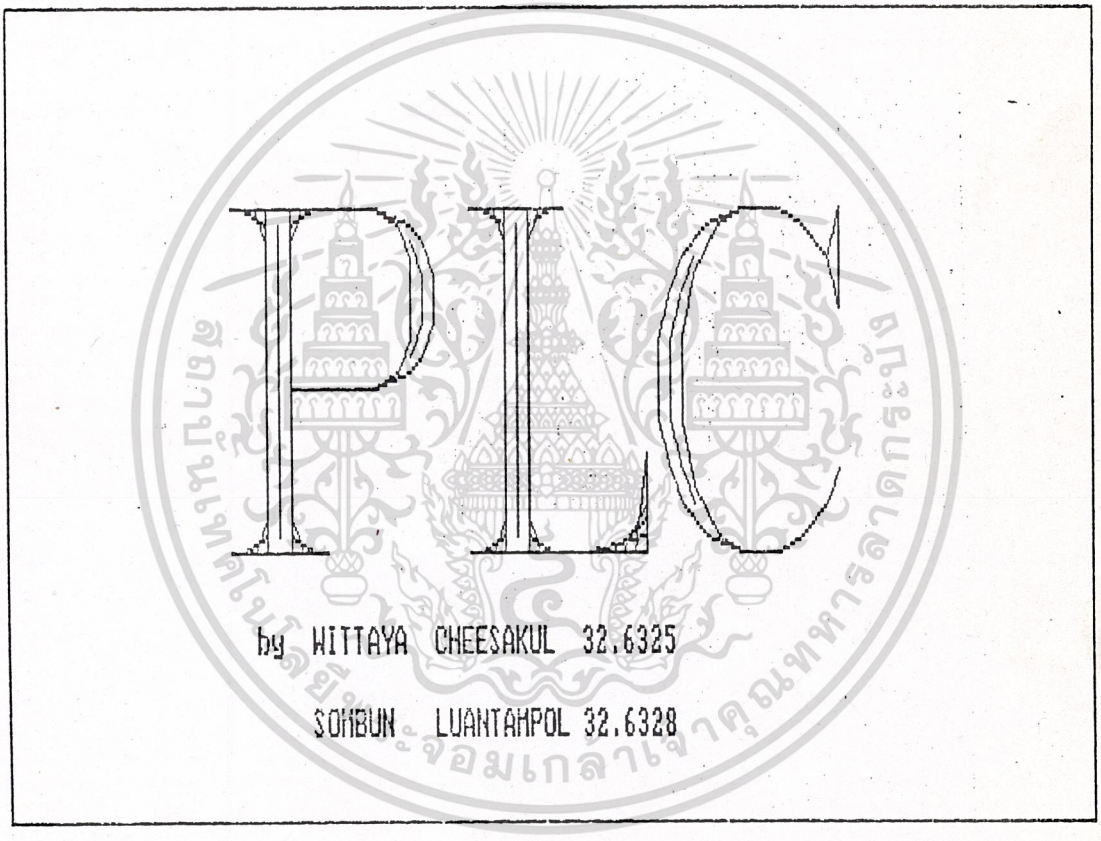
คำสั่งหน้าสัมผัสตรวจการเปลี่ยนสภาวะ ON-OFF(ON-OFF Transition Contact)

หน้าสัมผัสตรวจการเปลี่ยนสภาวะ ON-OFF ทำหน้าที่ตรวจสอบการเปลี่ยนสภาวะ ON สภาวะ OFFของรังก์หน้าสัมผัสจะมีสภาวะ "1" หรือ ON นานเกิน 1 ช่วงการสแกนของ PC เมื่อตรวจสอบพบการเปลี่ยนสภาวะของรังก์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

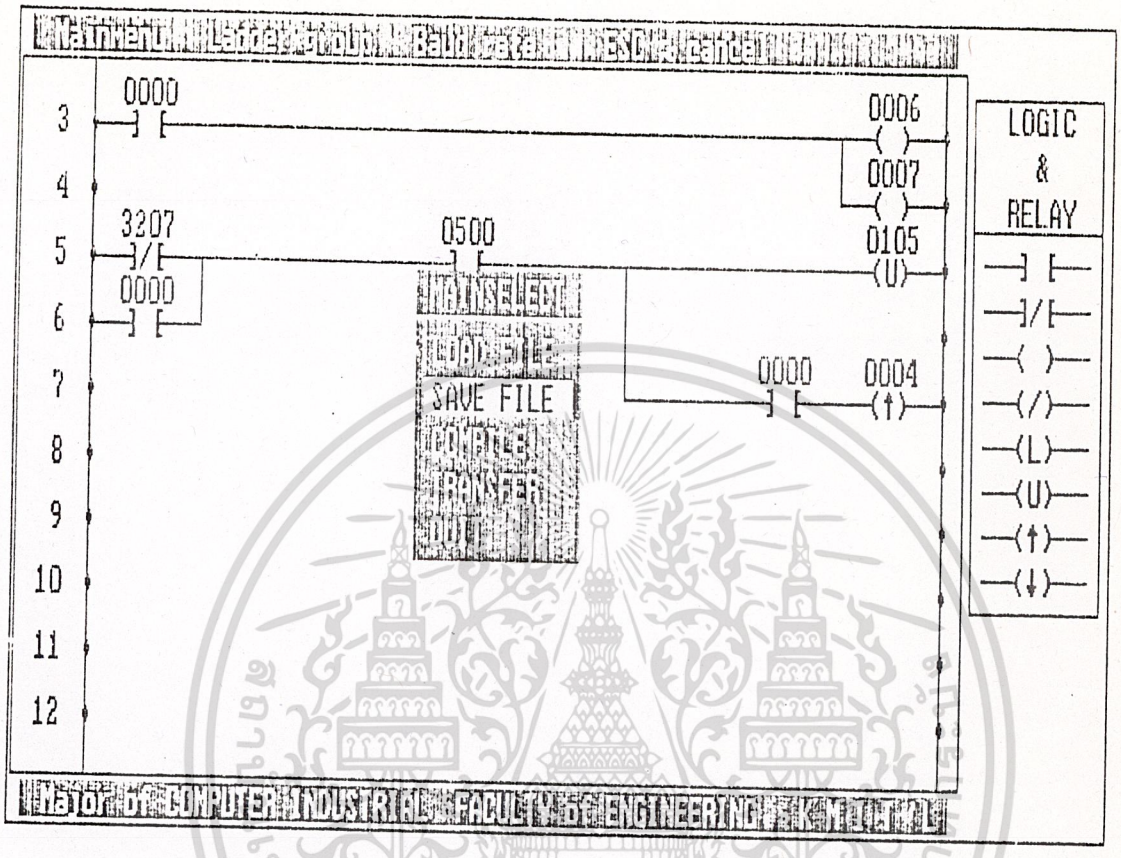
การใช้โปรมแกรม

ต่อไปนี้จะเป็นการแสดงวิธีการใช้งานในส่วนของโปรแกรมโดยการใช้ Menu



รูปที่ 1
แสดงข้อความเมื่อโหลดโปรแกรมครั้งแรก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2

กดอักษร "m" จะเป็นการเลือกการทำงาน

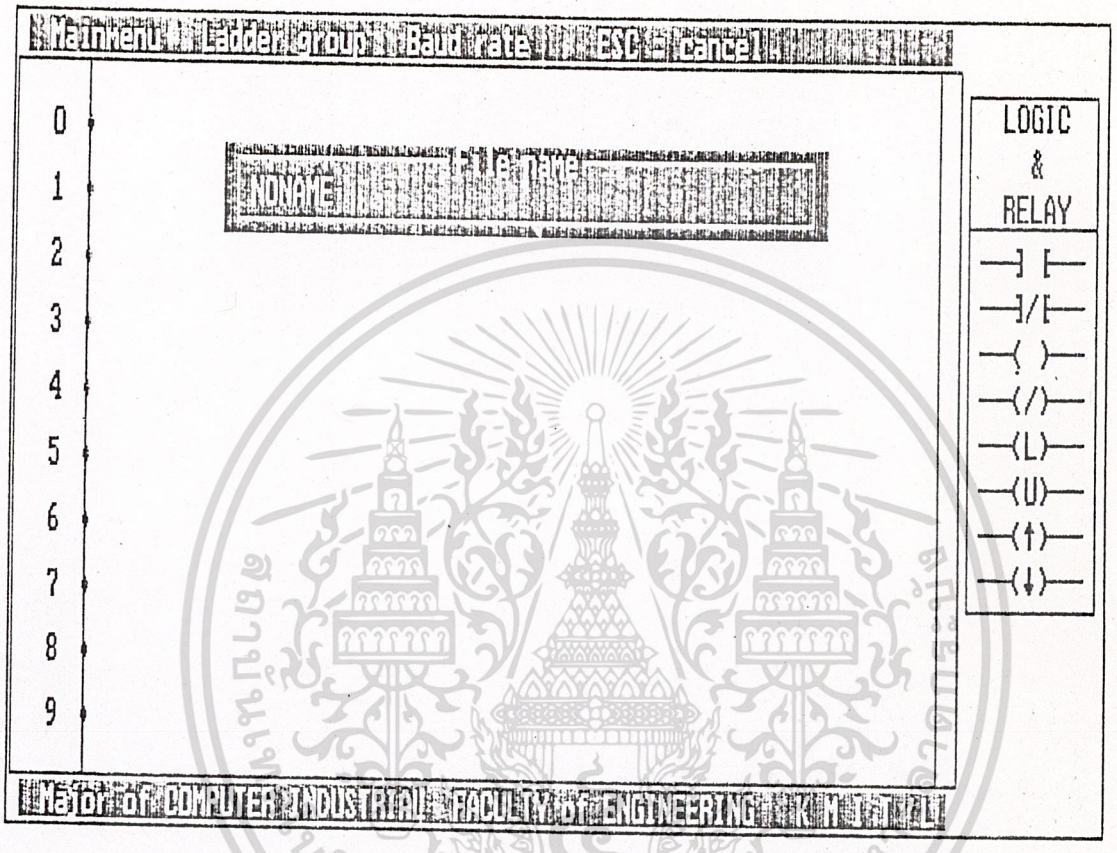
LOAD FILE โหลด ladder file (*.lad) จาก disk มายัง Editor

SAVE FILE เก็บ Ladder Program จาก Editor ลง Disk

COMPILE จะเปลี่ยน Ladder Program ไปเป็น File *.C51 เป็น Code ภาษา Assembly ของ MCS-51 และเป็น *.HEX ซึ่งเป็น Intel Hex Format Code

TRANSFER จะส่ง Intel Hex Format Code ไปยังเครื่อง PLC โดยผ่านทาง Serial Port RS 232

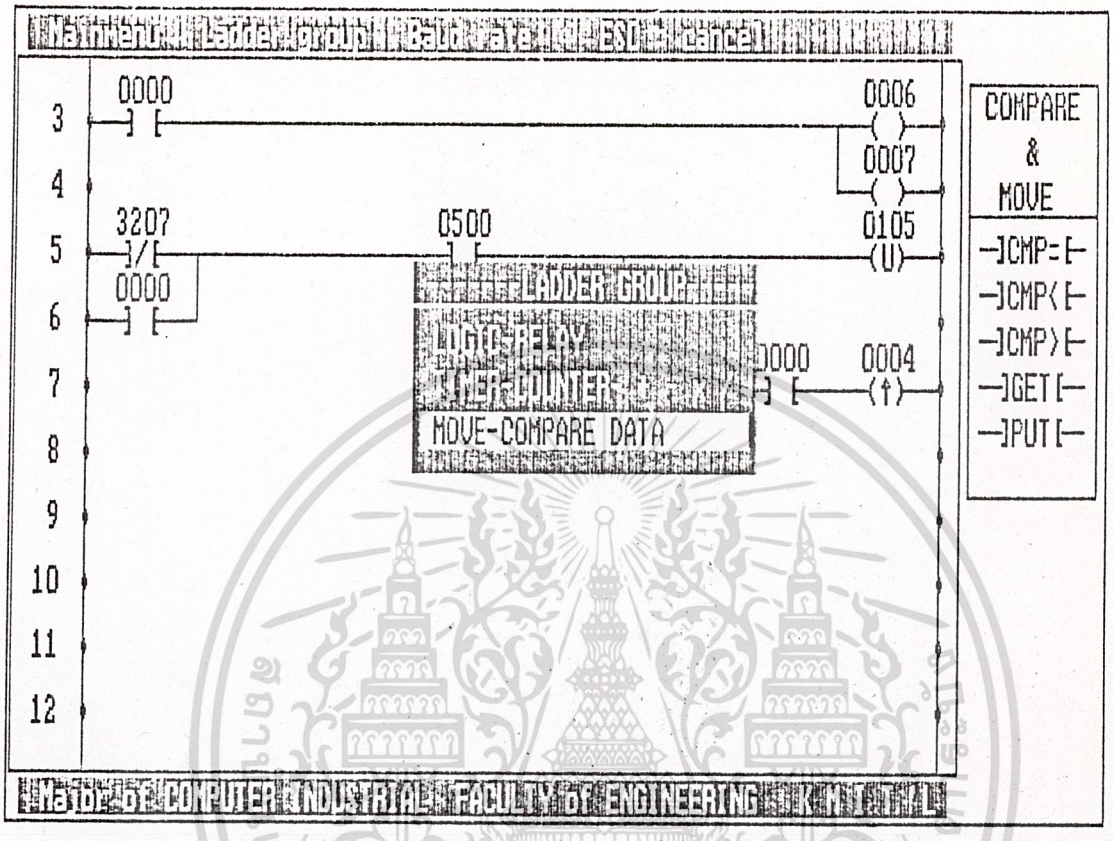
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3

เมื่อเลือกจาก Mainmenu คือ Load File,Save File,Transfer จะให้ใส่ชื่อ File โดยจะปรากฏชื่อ File เดิมที่เคยใส่ไว้ ถ้าเราจะไม่เปลี่ยนแปลงให้กด Key ENTER

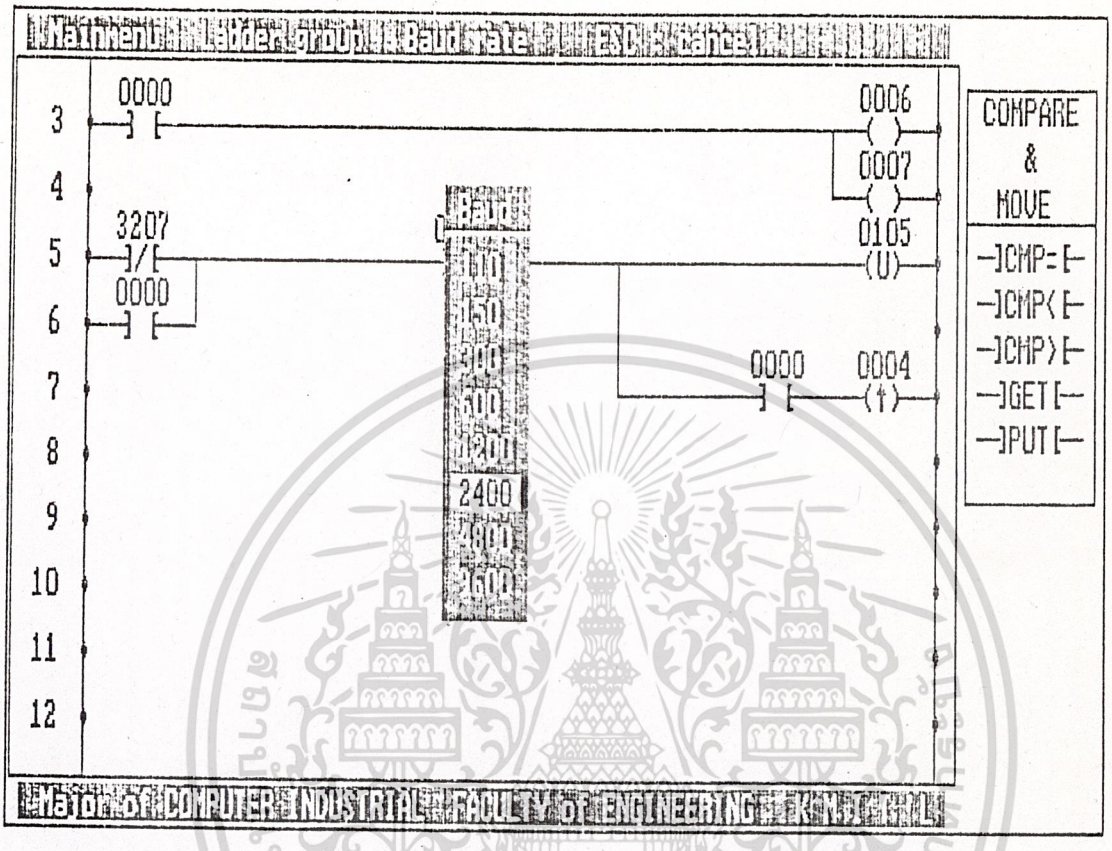
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4

จาก Editor ถ้ากด "L" จะเป็นการเลือกชุดคำสั่ง LADDER เมื่อเลือกแล้วชุดคำสั่งที่ต้องการจะปรากฏทางด้านขวามือ

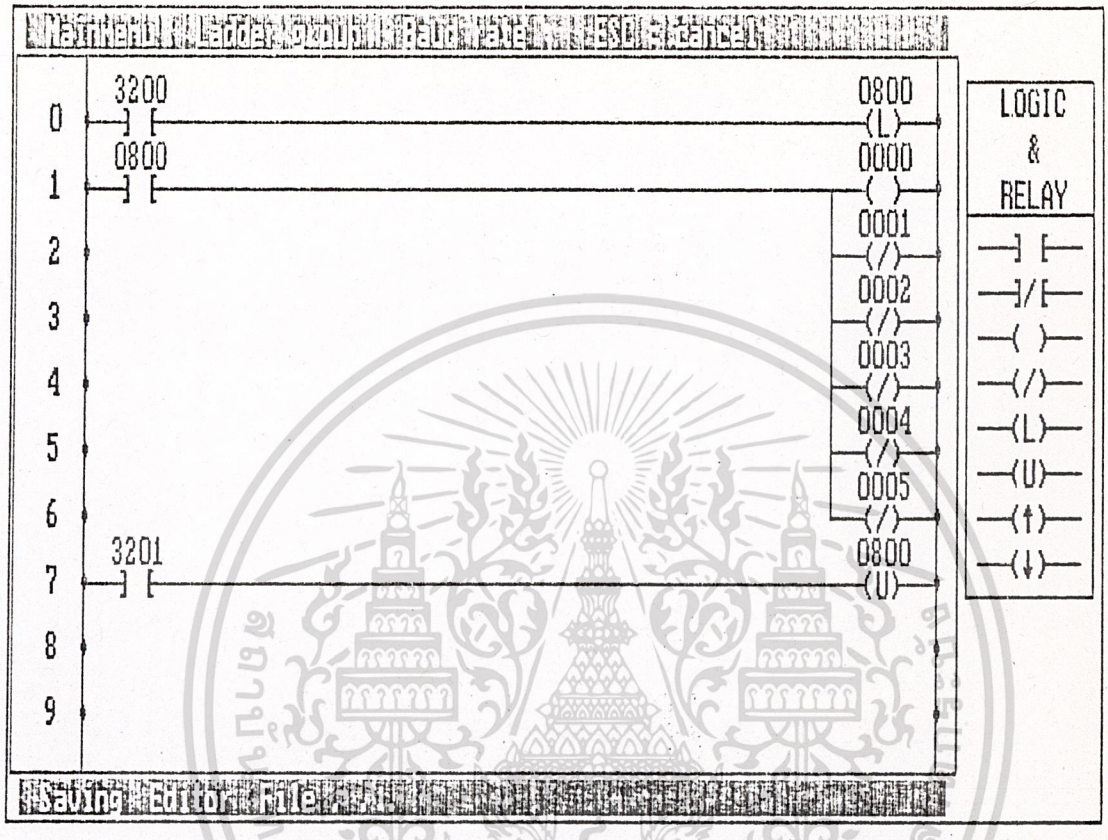
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5

เมื่อกด "B" จะเป็นการกำหนดความเร็วในการส่งข้อมูลจาก Microcomputer ไปยัง PLC โดยผ่านทาง RS 232

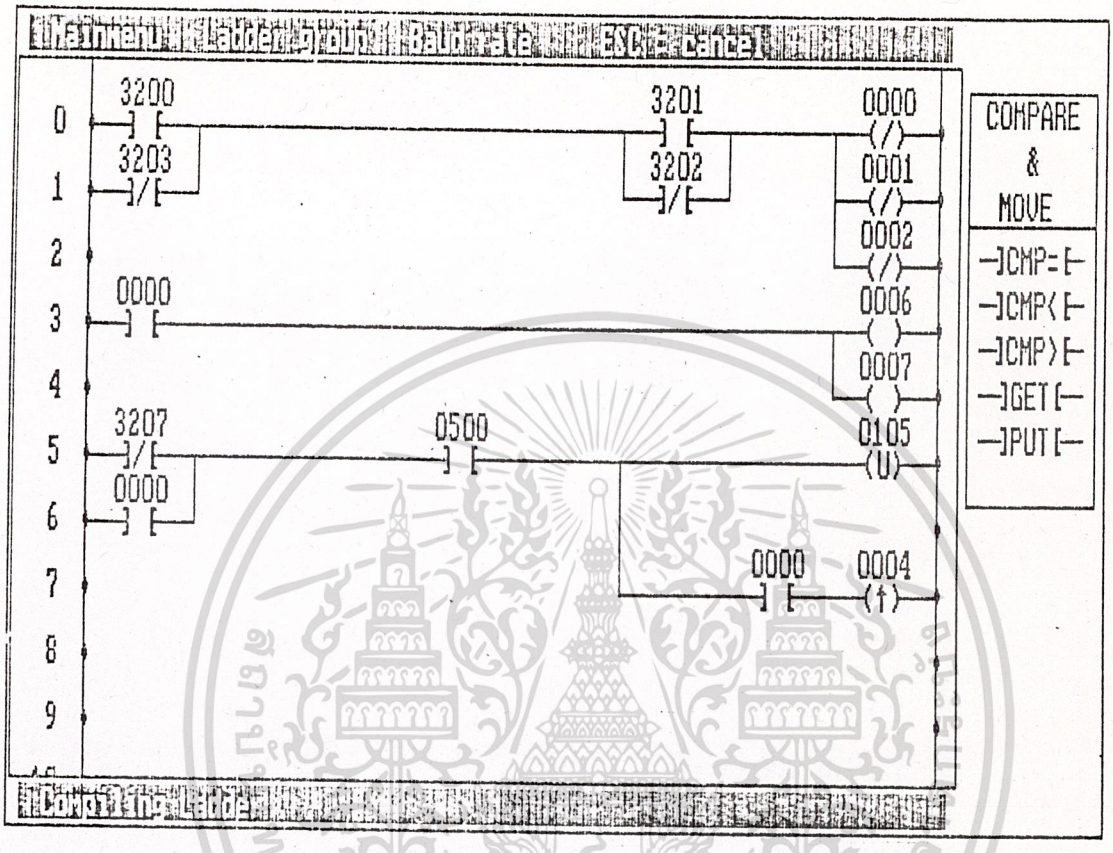
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 6

เมื่อเลือก Save File จาก Main Menu จะปรากฏข้อความ "Saving Editor File..." ข้างล่างของ Editor

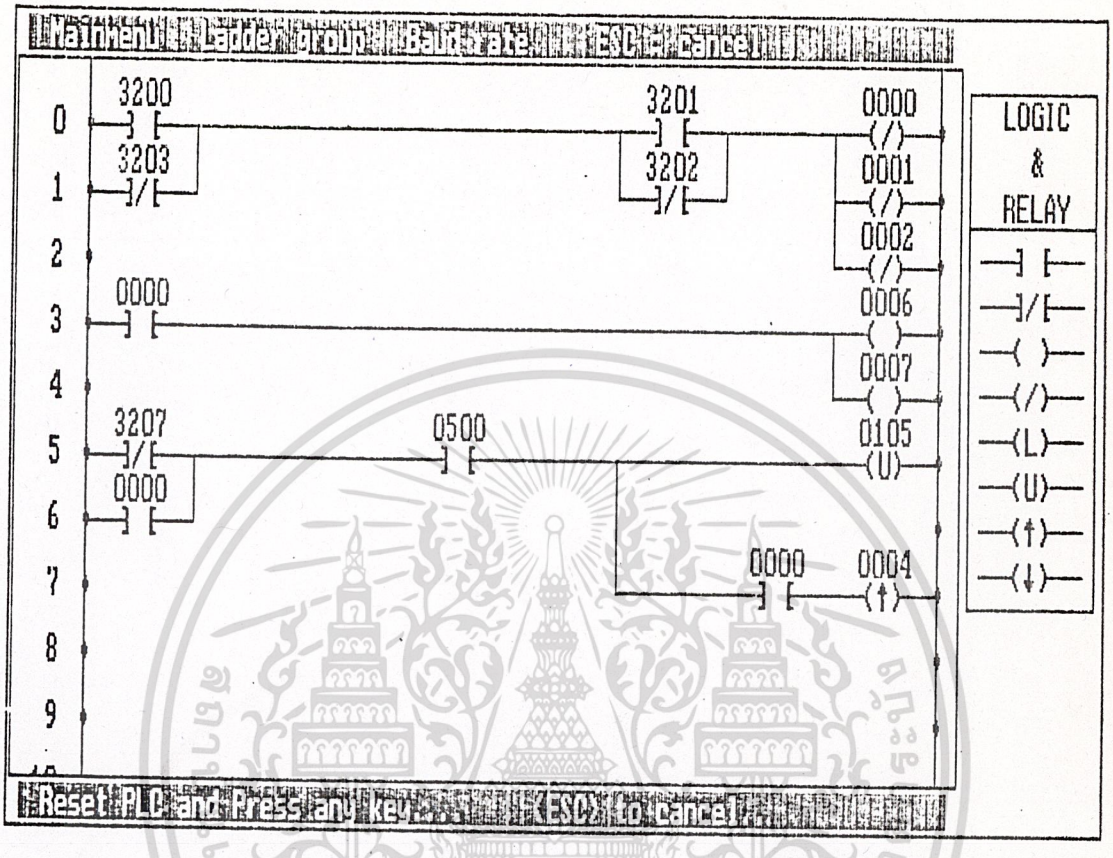
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 7

เมื่อเลือก Compiling Ladder จาก Main Menu จะปรากฏข้อความ Compiling Ladder..." ข้างล่างของ Editor

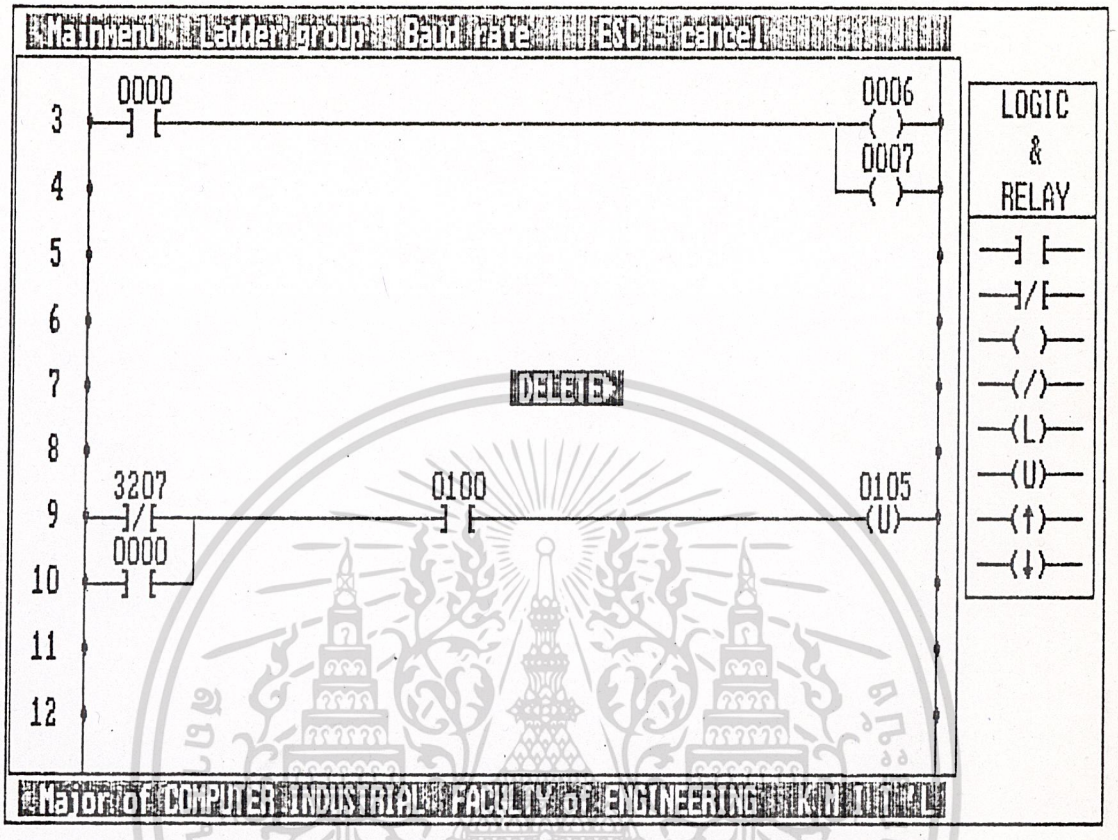
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 8

เมื่อเลือก Transfer Ladder จาก Main Menu จะปรากฏข้อความ Reset PLC and Press any key... <ESC> to Cancel

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 9

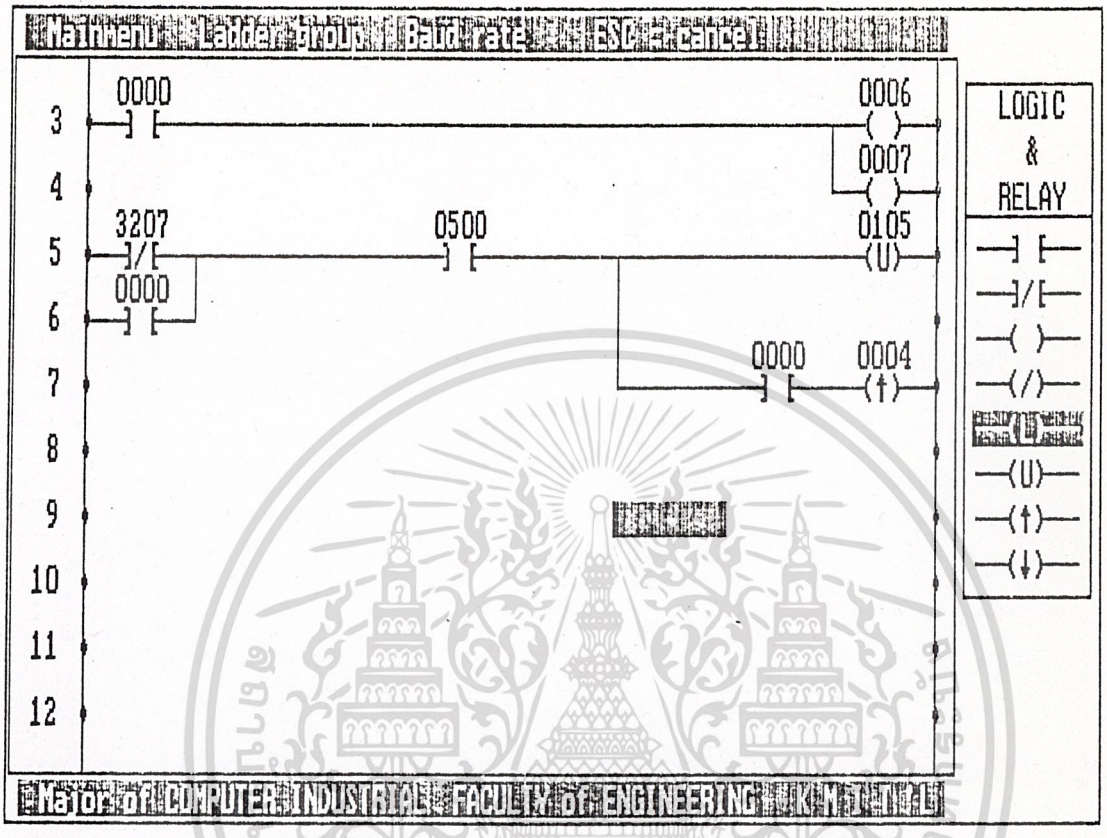
กด Key "Del" จะปรากฏ Cursor "DELETE" เมื่อ Cursor นี้ไปทับที่ใด และกด ENTER จะลบ Ladder ที่ตำแหน่งนั้น

กด Key "-" จะปรากฏเส้นแนวนอน เมื่อกด ENTER จะเป็นการลากเส้นทางแนวนอน

กด Key "+" จะปรากฏเส้นแนวตั้ง เมื่อกด ENTER จะเป็นการลากเส้นทางแนวตั้ง

ถ้ากด Cursor เป็น "Del" เมื่อกด "+" จะลบเส้นทางแนวตั้ง

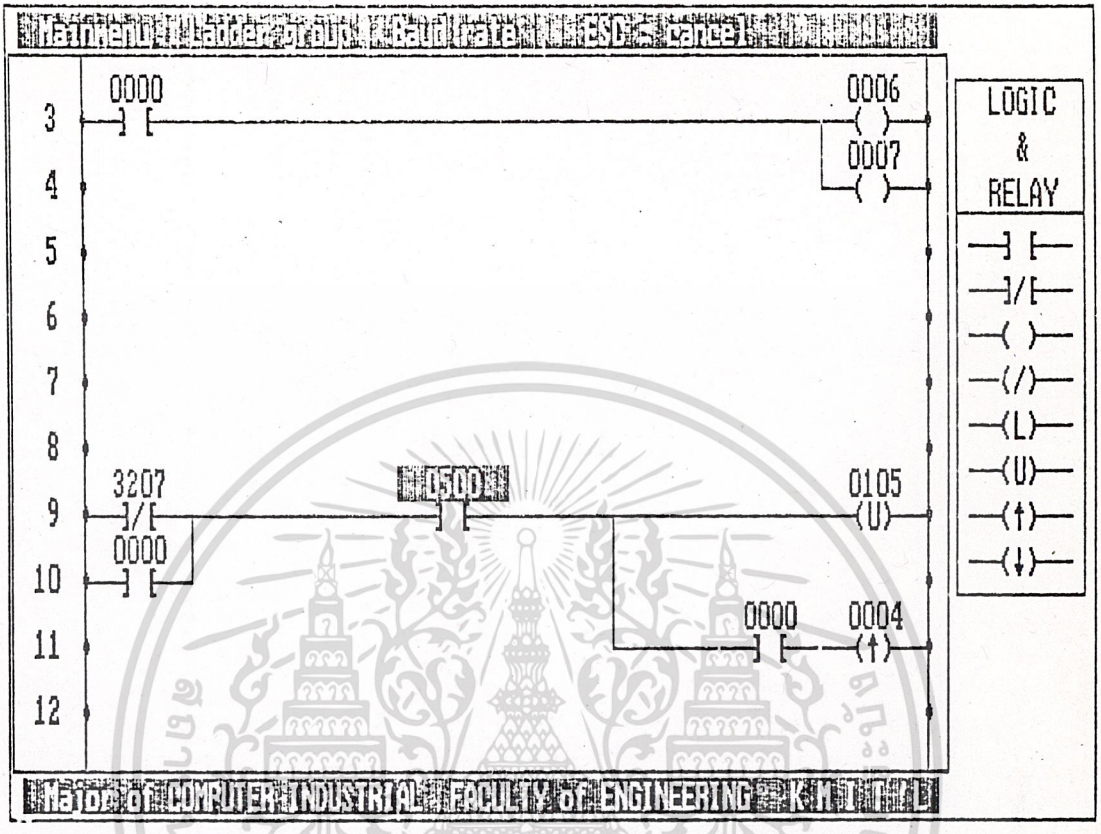
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 10

เมื่อกด Key "Ins" จะเป็นการเลือก สัญญลักษณ์ Ladder ให้มาปรากฏที่ Cursor

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 11

เมื่อกด Key "Home" จะให้ใส่ค่า Channel และตำแหน่ง Input/Output ของ Ladder โดย

กด " " จะเป็นการเลือก Channel โดยเลือกได้จากตำแหน่งที่ 00-39 โดย

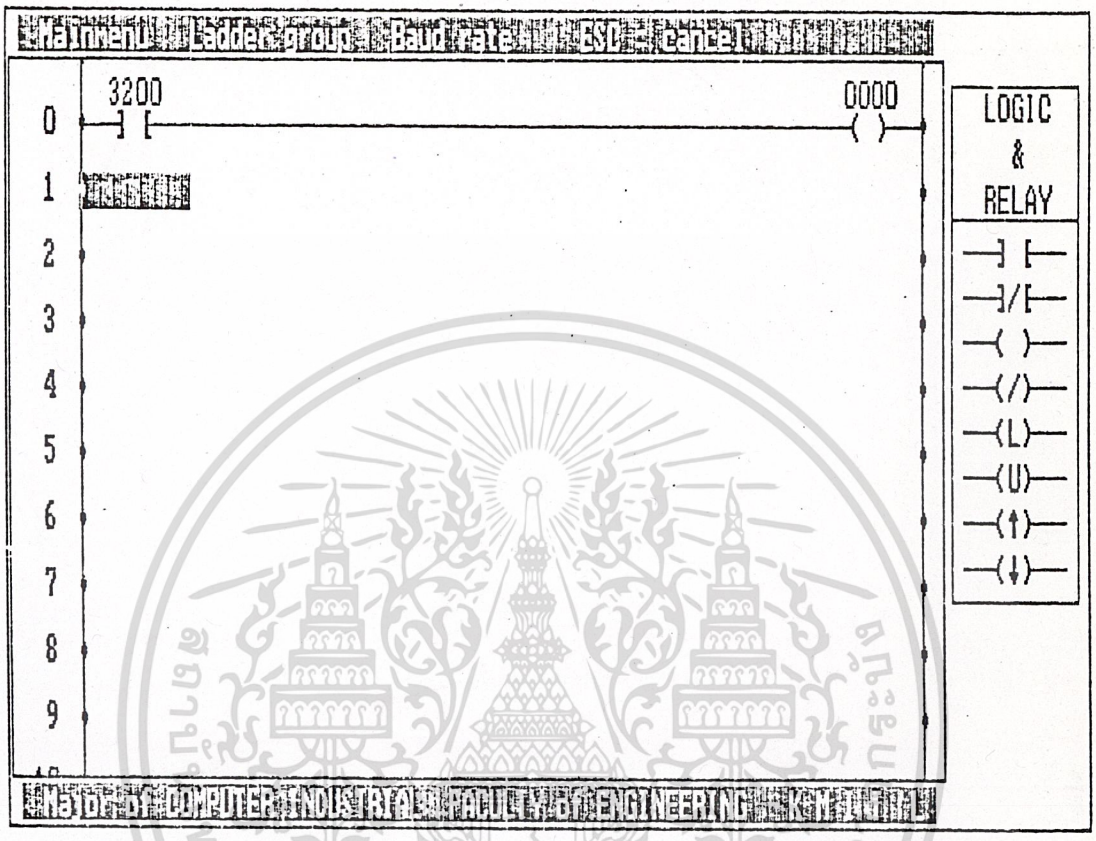
1.Channel 00-07 จะเป็น Output แต่ PLC ที่มีอยู่จะใช้แต่เพียง Channel 00 เท่านั้น นอกนั้นต้องต่อเพิ่มเติมทาง I/O Port

2.Channel 08-31 จะเป็น Internal Relay

3.Channel 32-39 จะเป็น Input แต่ที่ PLC มีอยู่จะใช้เพียง Channel 32 เท่านั้น นอกนั้นต้องต่อเพิ่มเติมทาง I/O Port

กด " " จะเลือกช่องภายใน Channel โดย 1 Channel เลือกได้ 8 ช่อง

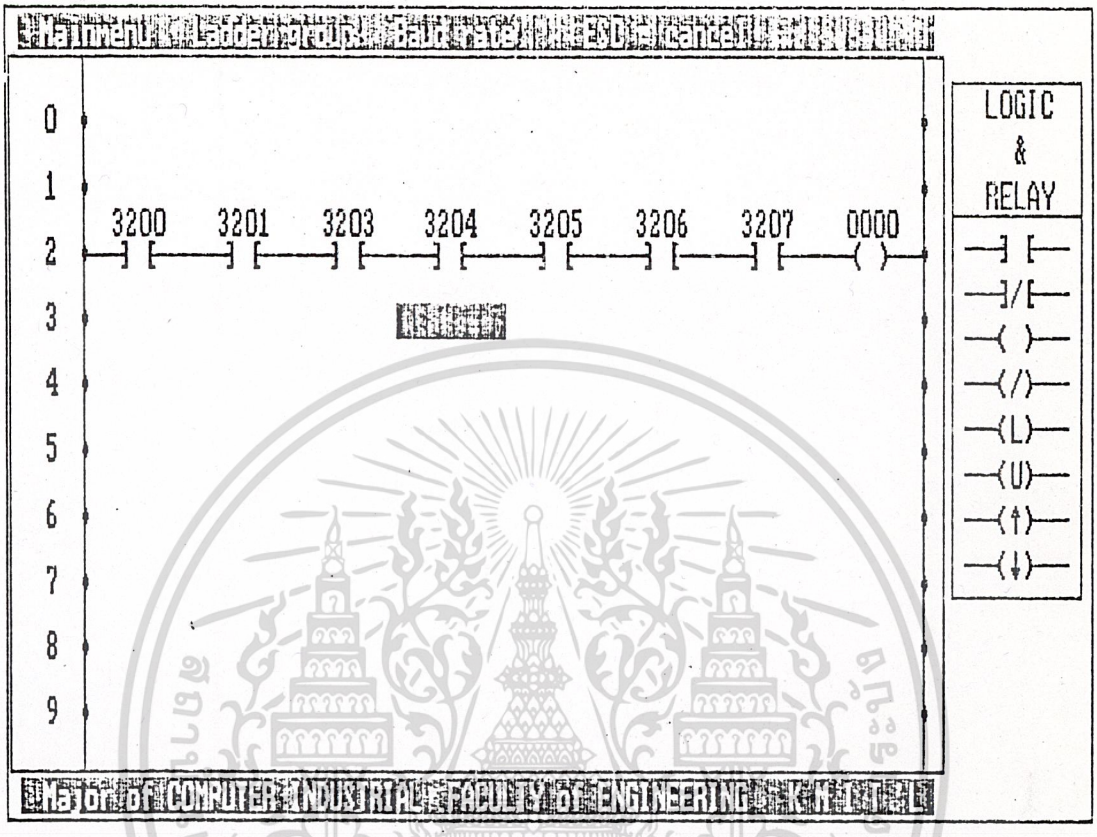
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 12

แสดงการทดลองใช้คำสั่ง อ่าน Input 3200 ไปออกยัง Output 0000

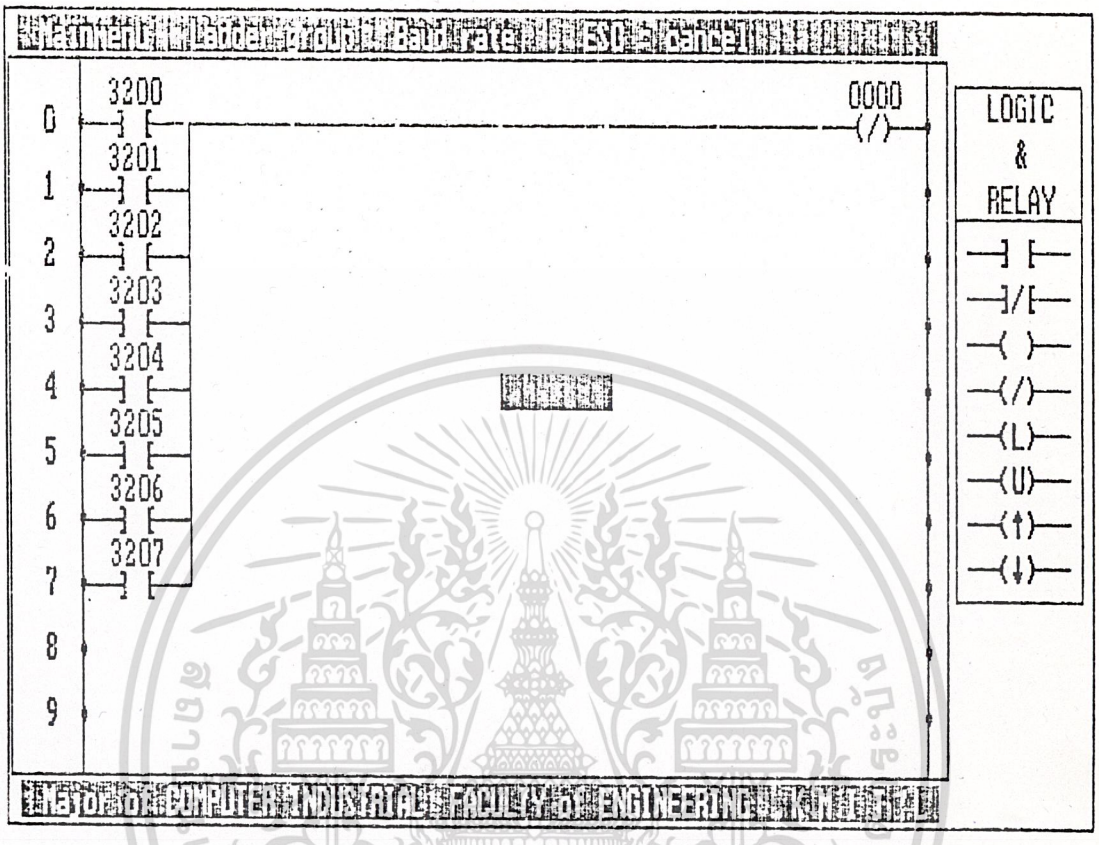
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 13

แสดงการใช้คำสั่ง AND โดยเอา Input 7 จุด AND กัน นำผลลัพธ์ออกทาง Output 0000

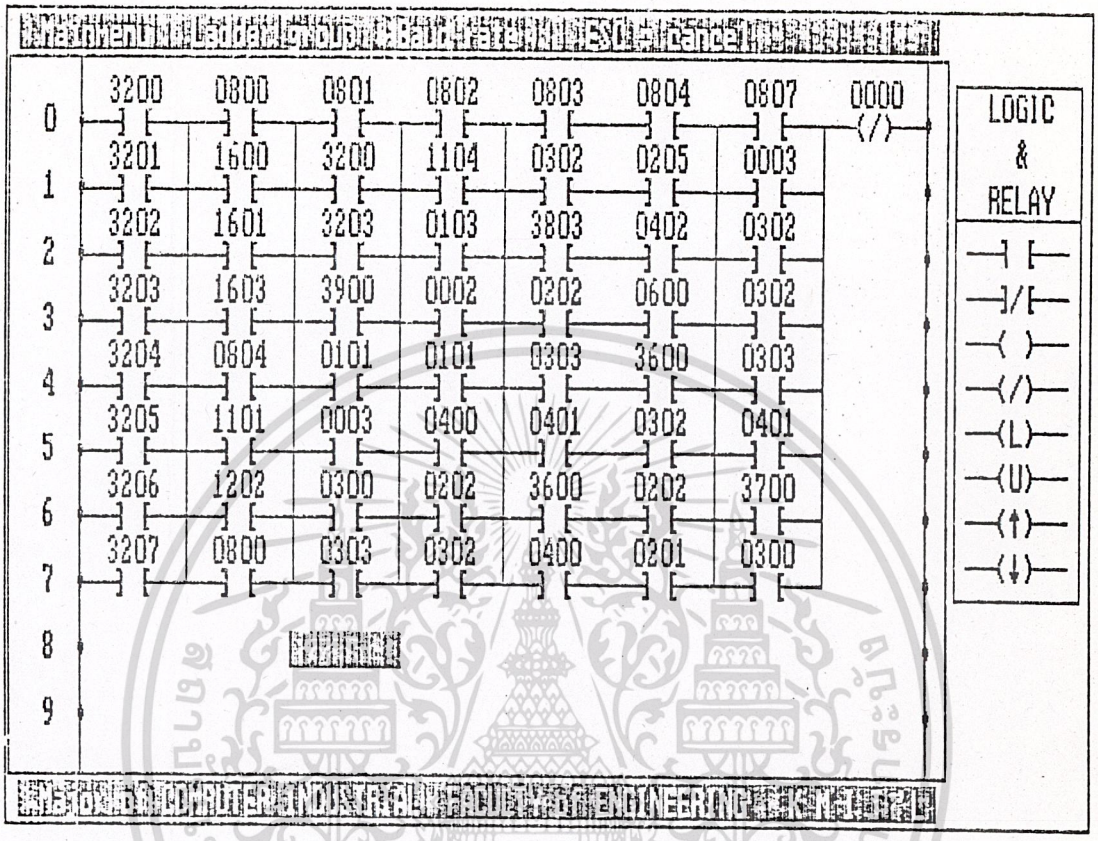
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 14

แสดงการใช้คำสั่ง OR โดยเอา Input 8 จุด OR กัน นำผลลัพธ์ออกทาง Output 0000

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 15

แสดง Program Ladder ที่ซับซ้อน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

บทวิจารณ์และสรุป

5.1 บทวิจารณ์

ข้อดีของการใช้ PLC แบบเขียนโปรแกรมด้วยภาษาแลดเดอร์คือ การเขียนโปรแกรมในการควบคุมขบวนการต่างๆ เราสามารถทำได้ง่ายและสะดวก

โครงงานนี้ใช้หน่วยประมวลผลเป็นCPU ในตระกูล MSC - 51 ซึ่งออกแบบมาสำหรับการควบคุมโดยเฉพาะ ที่สามารถประกอบวงจรทางด้าน Hardware อีกเพียงเล็กน้อยก็สามารถทำงานได้เนื่องจาก CPU ตัวเดียวมีทั้งอินพุต/เอาต์พุตพอร์ต พอร์ต การสื่อสารข้อมูลแบบอนุกรม Timer และ Counter ในตัวเดียวกัน

5.2 สรุป

เราสามารถติดต่อระหว่างไมโครคอนโทรลเลอร์ในตระกูล MCS - 51 เข้ากับไมโครคอมพิวเตอร์โดยผ่านทางพอร์ตอนุกรม RS 232 C ของไมโครคอมพิวเตอร์ เพื่อนำข้อมูลที่เป็นคำสั่งไปควบคุมขบวนการต่างๆ ได้อย่างมีประสิทธิภาพตามแลดเดอร์ที่เขียนขึ้น

การพัฒนาประสิทธิภาพของซอฟแวร์ ในโครงงานนี้เราสามารถพัฒนาให้มีการใช้งานของ Timer, Counter, รวมไปถึงการ บวก ลบ คูณ และ หาร จะทำให้โครงงานนี้มีประสิทธิภาพและความสามารถเท่าเทียม PLC ที่ผลิตจำหน่ายในท้องตลาด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หนังสืออ้างอิง

- PROGRAMMABLE CONTROLLER เทคนิคและการใช้งานเบื้องต้น, สุพรรณ กุลพาณิชย์ กรุงเทพ พ.ศ.2533.

- หลักการทำงานและเทคนิคการประยุกต์ใช้งาน PC/PLC, สุเธียร เกียรติสุนทร กรุงเทพ พ.ศ.2531.

- MCS - 51 MICROCONTROLLER DATA BOOK, Intel Corporation.

- TURBO C V2.0 REFERENCE GUIDE, Borland International.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก

- INTEL HEX FILE

- โปรแกรมภาษา Assambly ของ MCS-51 ที่ได้จากการ Compile Ladder Diagram

- รูปแบบของ INTEL HEX FILE ที่ได้จากการ Compile Ladder Diagram

- Circuit of Control Board

- SOURCE PROGRAM

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

INTEL HEX FILE

INTEL HEX FILE เป็นไฟล์รหัส ASC II ซึ่งสามารถใช้คำสั่ง TYPE ของ DOS หรือ TEXT EDITOR ทั่ว ๆไปเช่น SIDEKICK,WORDSTAR เรียกดูข้อมูลบนมอนิเตอร์ หรือพิมพ์ข้อมูลที่ PRINTER ได้และเป็นไฟล์ที่สามารถส่งผ่านพอร์ตสื่อสารอนุกรมได้ และที่สำคัญคือเป็นไฟล์ที่โปรแกรม แอสแซมเบลอร์ทั่วไป ใช้เป็นเอาต์พุตไฟล์เช่น โปรแกรม SIRM,SXA51,CROSS16, CROSS32,TURBO48 เป็นต้น

รูปแบบของ INTEL HEX FILE ประกอบด้วย เรคคอร์ด 2 ชนิด คือ เรคคอร์ดข้อมูล (DATA RECORD)และเรคคอร์ดสิ้นสุดไฟล์ (END-OF-FILE RECORD)โดยรูปแบบของเรคคอร์ดจะเริ่มต้นด้วยรหัสนำ 9 ตัวอักษร ตามด้วยข้อมูล(ถ้ามี)และรหัสปิดท้าย 2 ตัวอักษร โดยมีรูปแบบดังนี้

:BCAAAATTHH...HHCC

: ตัวอักษรเริ่มต้น

BC -จำนวนไบท์ข้อมูลในเรคคอร์ด มีค่าเป็นเลขฐาน 16 (HEX) (BC=00 ถ้าเป็น เรคคอร์ดสิ้นสุดไฟล์)

AAAA -เป็นแอดเดรสของข้อมูลไบท์แรกในเรคคอร์ด

TT -แสดงชนิดของเรคคอร์ด

(TT=00 ถ้าเป็นเรคคอร์ดข้อมูล)

(TT=01 ถ้าเป็นเรคคอร์ดสิ้นสุดไฟล์)

HH -ข้อมูล 1 ไบท์

CC -ค่าของ CHECKSUM ซึ่งเป็นค่า TWO'S COMPLEMENT ของผลบวกของจำนวนไบท์ ข้อมูล(BC), แอดเดรส(AAAA)และชนิดของเรคคอร์ด(TT)

ตัวอย่าง INTEL HEX FILE FORMAT

:0300000002004EAD

:10000B00C0D053D01BE5622401D4F562B4601D75DD

:10001B006200E5612401D4F561B46010756100E5FF

:10002B00602401D4F560B42403756000C2917F0095

:10003B00DFFED291D200D0D0327B0179000000D906

:10004B00FCDBF875812F75890675BB20D2B9D2A98D

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น และห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
:10005B00D2AF75BCC47560137561237560D28C399
:10006B0012009SD20030000FDE5601200D2743A12F
:10007B000E9E5611200D2742A1200E9E5621200605
:10008B000D2740D1200E9C2008OEB438920759852A
:10009B00438B8075BDE8D282E21200C8FEC394412E
:1000AB00EE4022409504FC4C0E01200C8FEC394F2
:1000BB0041EE40022409540FFED0ED0D223098FD51
:1000CB00C298B599547622120OF3E41200E9ED126D
:02011B00FA22C6
:00000001F
```

จากตัวอย่างนี้จะเห็นว่าใน 1 ไบท์ จะประกอบด้วยหลายเรคคอร์ด และเรคคอร์ดสุดท้าย เป็น เรคคอร์ดสิ้นสุดไฟล์ ซึ่งจะไม่มีไบท์ข้อมูล(HH)

สองตัวแรกเรคคอร์ดแรก

: 03 0000 00 02 00 4E AD

: BC AAAA TT HH HH HH CC

ค่า CC(CHECKSUM)หาได้ดังนี้

03+00+00+00+02+00+4E=53H

ONE'S COMPLEMENT =0FFH-53H

=0ACH

TWO'S COMPLEMENT =0ACH+1

=0ADH

ค่า CC = 0ADH

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรมภาษา Assambly ของ MCS-51 ที่ได้จากการ Compile Ladder Diagram รูปที่ 13

```
;PLC version 1.0 (Test version) for CP-32 board
;By WITTAYA CHEESAKUL  KMITL
;   SOMBUN  LUANTAMPOL KMITL

ORG  8000H
U8255MODE   EQU 91H
CONTROLCH1  EQU 0E0E3H
MOV  A,#U8255MODE
MOV  DPTR,#CONTROLCH1
MOVX @DPTR,A
START:
MOV  DPTR,#0E0E0H
MOVX A,@DPTR
CPL  A
MOV  20H,A
MOV  C,00H
MOV  08H,C
MOV  DPTR,#0E0E0H
MOVX A,@DPTR
CPL  A
MOV  20H,A
MOV C,08H
ANL C,01H
MOV  09H,C
MOV  DPTR,#0E0E0H
MOVX A,@DPTR
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
CPL  A
MOV  20H,A
MOV C,09H
ANL C,03H
MOV  0AH,C
MOV  DPTR,#0E0E0H
MOVX A,@DPTR
CPL  A
MOV  20H,A
MOV C,0AH
ANL C,04H
MOV  0BH,C
MOV  DPTR,#0E0E0H
MOVX A,@DPTR
CPL  A
MOV  20H,A
MOV C,0BH
ANL C,05H
MOV  0CH,C
MOV  DPTR,#0E0E0H
MOVX A,@DPTR
CPL  A
MOV  20H,A
MOV C,0CH
ANL C,06H
MOV  0DH,C
MOV  DPTR,#0E0E0H
MOVX A,@DPTR
CPL  A
MOV  20H,A
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
MOV C,0DH
ANL C,07H
MOV  0EH,C
MOV  DPTR,#201H
MOVX A,@DPTR
MOV  20H,A
MOV  C,0EH
MOV  00H,C
MOV  A,20H
MOVX @DPTR,A
MOV  DPTR,#0E0E1H
MOVX @DPTR,A
LJMP START
END
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

INTEL HEX FILE ที่ได้จากการ Compile Ladder Diagram รูปที่ 13

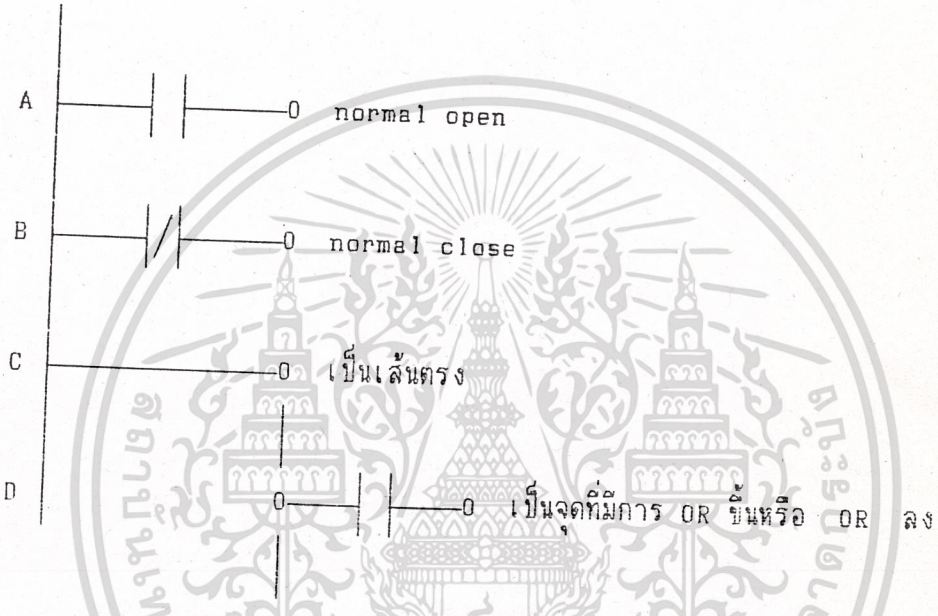
```
:10800000749190E0E3F090E0E0E0F4F520A20092BB
:10801000B90E0E0E0F4F520A208B201920990E0E7
:10802000E0E0F4F520A2098203920A90E0E0E0F497
:10803000F520A20A8204920B90E0E0E0F4F520A281
:10804000BB205920C90E0E0E0F4F520A20C820691
:1080500092OD90E0E0E0F4F520A20DB207920E90E0
:108060000201E0F520A20E9200E520F090E0E1F0A0
:038070000280068 5
:00000001FF
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การวิเคราะห์ LADDER DIAGRAM ของ COMPILER

การวิเคราะห์แบ่งออกได้ 3 ส่วนคือ

1 เมื่อ Load Ladder หลักแรก



A. MOV DPTR,# PORT
MOVX A,@DPTR
MOV 20H,A
MOV C.bit
MOV TEMPORARY,C

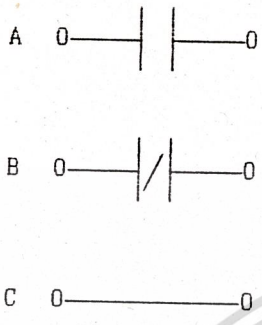
B. MOV DPTR,#PORT
MOVX A,@DPTR
MOV 20H,A
MOV C.bit
CPL C
MOV TEMPORARY,C

C. SETB bit

D. CLR bit

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.เมื่อ Load Ladder ตั้งแต่ 2 หลักขึ้นไป



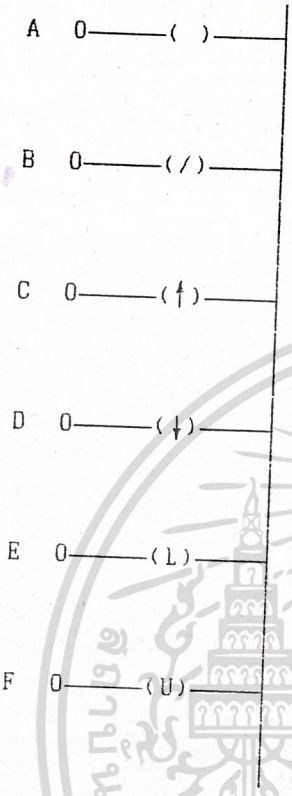
A. MOV DPTR,#PORT
MOV A,@DPTR
MOV 20H,A
MOV C,bit
ANL C,TEMPORARY-1
MOV TEMPORARY-1,C

B. MOV DPTR,#PORT
MOV A,@DPTR
MOV 20H,A
MOV C,bit
ANL C,/TEMPORARY-1
MOV TEMPORARY-1,C

C. MOV C,TEMPORARY-1
MOV TEMPORARY-1,C

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. เมื่อนำสถานะออก OUTPUT



A,B Load TEMPORARY-1 มาออก OUTPUT ถ้าเป็น OUTPUT NOT จะทำการ COMPLEMENT ก่อน

C. จะใช้หลักการคือ

เดิม	ปัจจุบัน	OUTPUT
0	0	0
0	1	1
1	0	0
1	1	0

new \ old	0	0
	1	0

Y = old . new

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

D. จะใช้วิธีคล้ายกับ C. จะได้ ฟังชั่นคือ

Y = old . new

E. เมื่อ OUTPUT จาก A,Bได้รับการ ACTIVE จะคงสภาวะนั้นไว้ตลอด(input set ของ KEEP)

F. เมื่อ OUTPUT จาก A,Bได้รับการ ACTIVE จะทำให้ output ถูก reset เป็น 0 (input reset ของ KEEP)

การ OR แต่ละหลัก

A. MOV C,0EH
ORL C,16H
MOV 0EH,C
MOV 16H,C

B. MOV C,A
ORL C,B
ORL C,C
MOV A,C
MOV B,C
MOV C,C

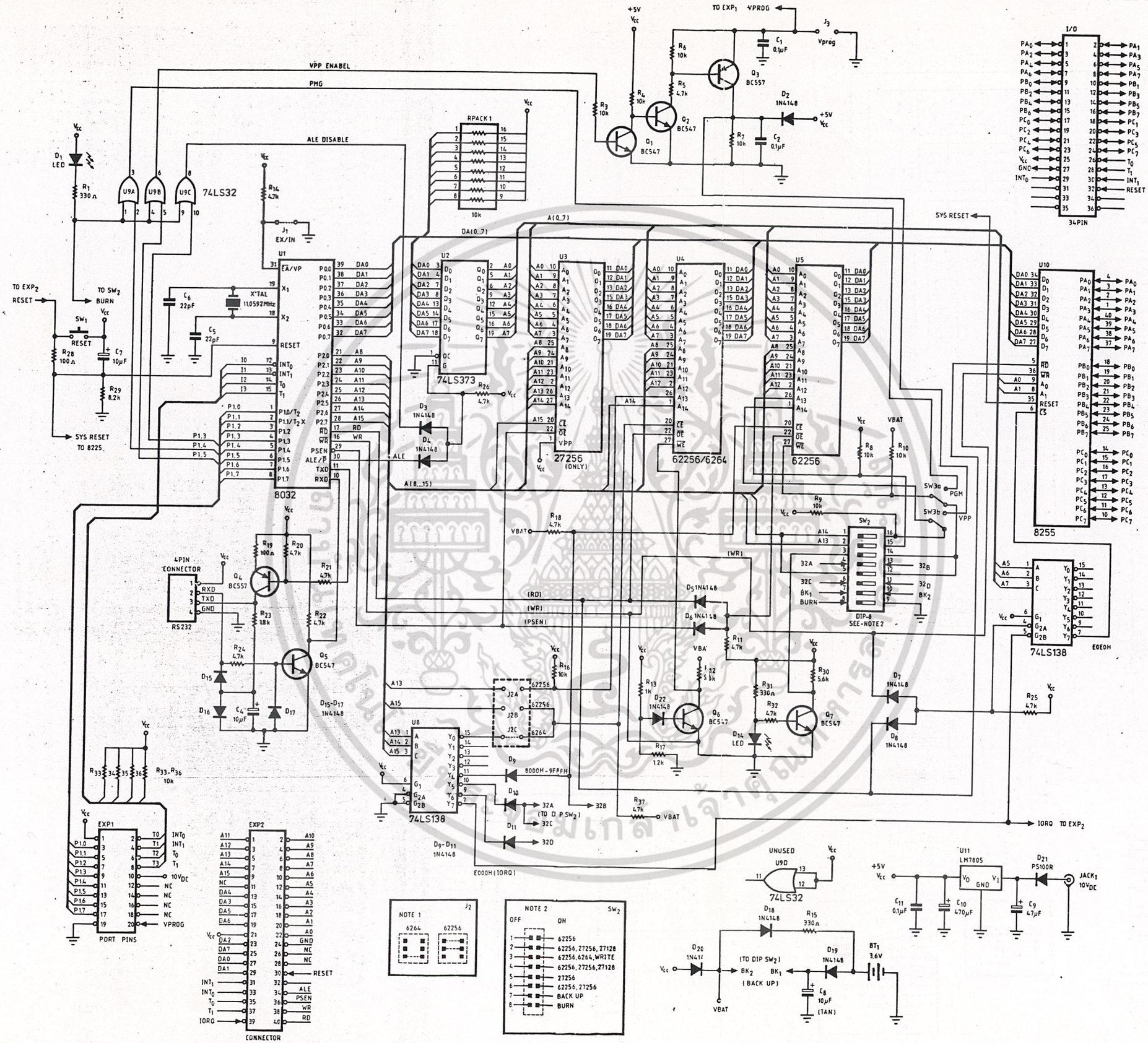
C. MOV C,B
ORL C,C

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
MOV B,C

MOV C,C
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
#include <stdio.h>
#include <graphics.h>
#include <alloc.h>
#include <stdarg.h>
#include <ctype.h>
#include <dos.h>
#include <conio.h>
#include <process.h>
#include <stdlib.h>
#include <math.h>
#include <bios.h>
#include <dir.h>
/*-----key definetion--------*/
#define  UP_ARROW     0x48
#define  DOWN_ARROW   0x50
#define  ESC          0x01
#define  RETURN       0x1c
#define  LEFT_ARROW   0x4b
#define  RIGHT_ARROW  0x4d
#define  PgUp         0x49
#define  PgDn         0x51
#define  HOME         0x47
#define  END          0x4f
#define  F1           0x3b
#define  INS          0x52
#define  LINE         0x4a
#define  DEL          0x53
#define  PLUS         0x4e
#define  E            0x12
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
#define  M              0x32
#define  I              0x17
#define  Q              0x10
#define  _i             0x1769
#define  _I             0x1749
/*#define  _y            0x
#define  _Y             0x*/
#define  CTRL_Y         0x1519
#define  CTRL_I         0x1709
#define  Exit           3
/*----argument definetion-----*/
#define  ON             1
#define  OFF            0
#define  YES            1
#define  NO             0
#define  BIGWINDOW      0
#define  MIDWINDOW      1
#define  SMALLWINDOW    2
#define  MAXROW         9
#define  MAXCOL         7
#define  MAXROWS        200
#define  DOWN           0
#define  UP             1
#define  UNDER          187
#define  HORLINE        'a'
#define  VERLINE        'b'
#define  BLACKGROUND    BLUE
#define  ENDROWS        NULL
#define  ENDPROGRAM     0xff
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
/*-----------------------------------------------*/
#define  NORMAL_OPEN          10
#define  NORMAL_CLOSE         11
#define  ENERGIZE_COIL        12
#define  DE_ENERGIZE_COIL     13
#define  LATCH                14
#define  UN_LATCH             15
#define  OFF_ON_TRANSITION    16
#define  ON_OFF_TRANSITION    17
/*-----------------------------------------------*/
#define  OR                   0
#define  AND                  1
#define  ORG                  "8000H"
#define  CONTROLCH1           "0E0E3H"
typedef  struct table{
     char  data[8];
     char col[7];
     char value[8];
     char value1[8];
     char under;
}TABLE;
#define DATA_READY  0x100
#define Xg      120
char bauds[8]={0x00,0x20,0x40,0x60,0x80,0xa0,0xe0};
char SETTING;
char baud,check;
int  portCOM=0;
/*int  com[4]={0,1,2,3};*/
   char *channaltext[40]={
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        "00","01","02","03","04","05","06","07",
        "08","09","10","11","12","13","14","15",
        "16","17","18","19","20","21","22","23",
        "24","25","26","27","28","29","30","31",
        "32","33","34","35","36","37","38","39"
        };
    char *bittext[8] = {
        "00","01","02","03","04","05","06","07"
    };

   int channal[40]  = {
      0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,
      24,25,26,27,28,29,30,31,32,33,34,35,36,37,38,39
   };
    char bit[8] = {
        0,1,2,3,4,5,6,7
    };
    char page=0;
   /*-------------------------------------------------------------------*/
    char *portname[40] ={
/*"#OUTPUTCH1","#OUTPUTCH2","#OUTPUTCH3","#OUTPUTCH4",
"#OUTPUTCH5","#OUTPUTCH6","#OUTPUTCH7","#OUTPUTHC8",*/
"#201H","#202H","#203H","#204H",
"#205H","#206H","#207H","#208H",
      /*"#INTERNALCH1","#INTERNALCH2","#INTERNALCH3","#INTERNALCH4",
"#INTERNALCH5","#INTERNALCH6","#INTERNALCH7","#INTERNALCH8",
"#INTERNALCH9","#INTERNALCH10","#INTERNALCH11","#INTERNALCH12",
"#INTERNALCH13","#INTERNALCH14","#INTERNALCH15","#INTERNALCH16"
"#INTERNALCH17","#INTERNALCH18","#INTERNALCH19","#INTERNALCH20",
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
"#INTERNALCH21","#INTERNALCH22","#INTERNALCH23","#INTERNALCH24"*/
"#209H","#20AH","#20BH","#20CH",
"#20DH","#20EH","#20FH","#210H",
"#211H","#212H","#213H","#214H",
"#215H","#216H","#217H","#218H",
"#219H","#21AH","#21BH","#21CH",
"#21DH","#21EH","#21FH","#220H",

/*"#INPUTCH1","#INPUTCH2","#INPUTCH3","#INPUTCH4",
"#INPUTCH5","#INPUTCH6","#INPUTCH7","#INPUTCH8",*/

"#0E0E0H","#0E000H","#0E020H","#0E040H",
"#0E060H","#0E080H","#0E0A0H","#0E0C0H"
      };
 char *latch[32]=   {
                /*"#OUTPUTCH1","#OUTPUTCH2","#OUTPUTCH3","#OUTPUTCH4",
"#OUTPUTCH5","#OUTPUTCH6","#OUTPUTCH7","#OUTPUTHC8",*/
    "#221H","#222H","#223H","#224H",
    "#225H","#226H","#227H","#228H",
/*"#INTERNALCH1","#INTERNALCH2","#INTERNALCH3","#INTERNALCH4",
"#INTERNALCH5","#INTERNALCH6","#INTERNALCH7","#INTERNALCH8",
"#INTERNALCH9","#INTERNALCH10","#INTERNALCH11","#INTERNALCH12",
"#INTERNALCH13","#INTERNALCH14","#INTERNALCH15","#INTERNALCH16"
"#INTERNALCH17","#INTERNALCH18","#INTERNALCH19","#INTERNALCH20",
"#INTERNALCH21","#INTERNALCH22","#INTERNALCH23","#INTERNALCH24"*/
    "#229H","#22AH","#22BH","#22CH",
    "#22DH","#22EH","#22FH","#230H",
    "#231H","#232H","#233H","#234H",
    "#235H","#236H","#237H","#238H",
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
      "#239H","#23AH","#23BH","#23CH",
      "#23DH","#23EH","#23FH","#240H"
      };
    char *ON_OFF[32]=   {
                  /*"#OUTPUTCH1","#OUTPUTCH2","#OUTPUTCH3","#OUTPUTCH4",
  "#OUTPUTCH5","#OUTPUTCH6","#OUTPUTCH7","#OUTPUTHC8",*/
      "#241H","#242H","#243H","#244H",
      "#245H","#246H","#247H","#248H",
  /*"#INTERNALCH1","#INTERNALCH2","#INTERNALCH3","#INTERNALCH4",
  "#INTERNALCH5","#INTERNALCH6","#INTERNALCH7","#INTERNALCH8",
  "#INTERNALCH9","#INTERNALCH10","#INTERNALCH11","#INTERNALCH12",
  "#INTERNALCH13","#INTERNALCH14","#INTERNALCH15","#INTERNALCH16"
  "#INTERNALCH17","#INTERNALCH18","#INTERNALCH19","#INTERNALCH20",
  "#INTERNALCH21","#INTERNALCH22","#INTERNALCH23","#INTERNALCH24"*/
      "#249H","#24AH","#24BH","#24CH",
      "#24DH","#24EH","#24FH","#250H",
      "#251H","#252H","#253H","#254H",
      "#255H","#256H","#257H","#258H",
      "#259H","#25AH","#25BH","#25CH",
      "#25DH","#25EH","#25FH","#260H"
      };
    char *OFF_ON[32]=   {
                  /*"#OUTPUTCH1","#OUTPUTCH2","#OUTPUTCH3","#OUTPUTCH4",
  "#OUTPUTCH5","#OUTPUTCH6","#OUTPUTCH7","#OUTPUTHC8",*/
      "#261H","#262H","#263H","#264H",
      "#265H","#266H","#267H","#268H",
  /*"#INTERNALCH1","#INTERNALCH2","#INTERNALCH3","#INTERNALCH4",
  "#INTERNALCH5","#INTERNALCH6","#INTERNALCH7","#INTERNALCH8",
  "#INTERNALCH9","#INTERNALCH10","#INTERNALCH11","#INTERNALCH12",
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
"#INTERNALCH13","#INTERNALCH14","#INTERNALCH15","#INTERNALCH16"
"#INTERNALCH17","#INTERNALCH18","#INTERNALCH19","#INTERNALCH20",
"#INTERNALCH21","#INTERNALCH22","#INTERNALCH23","#INTERNALCH24"*/
    "#269H","#26AH","#26BH","#26CH",
    "#26DH","#26EH","#26FH","#270H",
    "#271H","#272H","#273H","#274H",
    "#275H","#276H","#277H","#278H",
    "#279H","#27AH","#27BH","#27CH",
    "#27DH","#27EH","#27FH","#280H"
    };
 char *portoutput[8]={
     /*"#OUTPUT1","#OUTPUT2","#OUTPUT3","#OUTPUT4",
"#OUTPUT5","#OUTPUT6","#OUTPUT7","#OUTPUT8"*/
"#0E0E1H","#0E001H","#0E021H","#0E041H",
"#0E061H","#0E081H","#0E0A1H","#0E0C1H"
};
 char *bitbuffer[8]={
"00H","01H","02H","03H","04H","05H","06H","07H"
};
 char *temporaly[15][8]={
{"08H","09H","0AH","0BH","0CH","0DH","0EH","0FH"},
{"10H","11H","12H","13H","14H","15H","16H","17H"},
{"18H","19H","1AH","1BH","1CH","1DH","1EH","1FH"},
{"20H","21H","22H","23H","24H","25H","26H","27H"},
{"28H","29H","2AH","2BH","2CH","2DH","2EH","2FH"},
{"30H","31H","32H","3JH","34H","35H","36H","37H"},
{"38H","39H","3AH","3BH","3CH","3DH","3EH","3FH"},
{"40H","41H","42H","43H","44H","45H","46H","47H"},
{"48H","49H","4AH","4BH","4CH","4DH","4EH","4FH"},
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
{"50H","51H","52H","53H","54H","55H","56H","57H"},
{"58H","59H","5AH","5BH","5CH","5DH","5EH","5FH"},
{"60H","61H","62H","63H","64H","65H","66H","67H"},
{"68H","69H","6AH","6BH","6CH","6DH","6EH","6FH"},
{"70H","71H","72H","73H","74H","75H","76H","77H"},
{"78H","79H","7AH","7BH","7CH","7DH","7EH","7FH"}
 };
void *ladder;
/*-------------------------------------------------------------------*/
void graph640x200(void)
 {
  int g_driver,g_mode;
  int cbsize;
  detectgraph(&g_driver,&g_mode);
  switch(g_driver){
       case CGA : g_mode = CGAHI;
break;
       case EGA : g_mode = EGALO;
break;
       case MCGA : g_mode = MCGAMED;
break;
       case EGA64 : g_mode = EGA64LO;
break;
       case VGA : g_mode = VGALO;

break;
      default   :printf("program no support CRT");exit(1);
      }
 cbsize=setgraphbufsize(1000);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
   setgraphbufsize(cbsize);
   initgraph(&g_driver,&g_mode,"");
   }
/*-------------------------------------------------------------*/
void graphmonochom(void)
 {
  int g_driver,g_mode;
  detectgraph(&g_driver,&g_mode);
  switch(g_driver){
      case HERCMONO    : g_mode = HERCMONOHI;
break;
      case EGAMONO     : g_mode = EGAMONOHI;
break;
      default    : printf("program no support CRT");exit(1);
  break;
      }
  initgraph(&g_driver,&g_mode,"");
  }
 /*-------------------------------------------------------------*/
int gprintf(int xloc, int yloc, char *fmt, ... )
{
    va_list argptr;
    char str[140];
    int cnt;
    va_start(argptr,fmt);cnt = vsprintf(str,fmt,argptr);
    outtextxy (xloc,yloc,str);yloc += textheight("H") + 2 ;
    va_end (argptr);return(cnt);
}
/*-------------------------------------------------------------*/
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
int gscanf(char status,int x,int y,char *name,char *title)
{
  char i=0;
  int k;
  char key;
  unsigned size;
  void  *oldimage,*cursor;
   /*-----save old image ---------*/
   size=imagesize(x,y,x+352,y+21);oldimage=malloc(size);
   getimage(x,y,x+352,y+21,oldimage);putimage(x,y,oldimage,XOR_PUT);
   /*-----creat black cursor------*/
   bar(x,y,x+352,y+21);size=imagesize(x+1,y+1,x+8,y+10);
   cursor=malloc(size);getimage(x+1,y+1,x+8,y+10,cursor);
   /*-------------------------------*/
   setcolor(BLACK);outtextxy(x+136,y,title);
   moveto(x+130,y+3);lineto(x+8,y+3);lineto(x+8,y+18);lineto(x+344,y+18);
   lineto(x+344,y+3);lineto(x+210,y+3);
   if(status==OFF){
     while(name[i]!='.')
 gprintf(x+(i-1)*8+16,y+8,"%c",name[++i-1]);
   }
   do{
      k=bioskey(0);
      if((char)k==0)continue;
      else key=(char)k;
      if((key>=48)&&(key<=122)||(key==0x08)||(key==0x0d)){
      switch(key){
case 0x08 : if(i!=0)--i;
    putimage(x+(i*8)+16,y+8,cursor,COPY_PUT);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        break;
    default   :
        if ((i<=39)&&(key!=0x1b))
        *(name+(i++))=toupper(key);
        break;
         }
         if((i!=0)&&(key!=0x0d)) gprintf(x+(i-1)*8+16,y+8,"%c",name[i-1]);
         }
        }while((key!=0x0d)&&(key!=0x1b));
        if(key!=0x1b){
    name[i-1]='.';name[i]='L';name[i+1]='A';
    name[i+2]='D';name[i+3]='\0';
        }
        putimage(x,y,oldimage,COPY_PUT);
        free(oldimage);free(cursor);
        setcolor(BROWN);
        if((i==1)&&(key==0x0d))return(NO); /* if  no name */
        if((key==0x0d)&&(i>1))return(YES);
        if(key==0x1b)return(0x1b);   /* if push ESC */
}
/*-------------------------------------------------------------*/
title()
   {
   int g_driver,g_mode;
   clearviewport();
   settextstyle(TRIPLEX_FONT,HORIZ_DIR,USER_CHAR_SIZE);
   setusercharsize(7,1,8,2);
   setcolor(YELLOW);
   outtextxy(130,20,"PLC");
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    settextstyle(DEFAULT_FONT,HORIZ_DIR,USER_CHAR_SIZE);
    outtextxy(145,150,"Ver. 1.0   by  32.6325   KMIT'L   ");
    }

/*-----------------------------------------------------------------*/
char instruction_group(x,y)
int x;
int y;
   {
   char *menu[]   = {
      "LOGIC-RELAY","TIMER-COUNTER- + - x /","MOVE-COMPARE DATA"
   };
   char i;
   int key;
   char select;
   void *bars,*oldimage;
   unsigned size;
   select = 0;
   /*-----save old image ---------*/
   size = imagesize(x,y,x+200,y+50);oldimage = malloc(size);
   getimage(x,y,x+200,y+50,oldimage);putimage(x,y,oldimage,XOR_PUT);
   /*------------------------------*/
   setcolor(BLACK);bar(x,y,x+200,y+50);line(x,y+10,x+200,y+10);
   outtextxy(x+8,y+2,"-------LADDER GROUP------");
   /*-----create box bar menu------*/
   size = imagesize(x+5,y+12,x+195,y+20);
   bars = malloc(size);
   getimage(x+5,y+12,x+195,y+20,bars);
   /*------------------------------*/
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    for (i=0;i<=2;i++)
        outtextxy(x+12,y+(i*11)+15,menu[i]);
    do{
        putimage(x+5,y+11*select+14,bars,XOR_PUT);
        setcolor(WHITE);
        outtextxy(x+12,y+11*select+15,menu[select]);
        key = bioskey(0);
        key>>=8;
        putimage(x+5,y+11*select+14,bars,XOR_PUT);
        setcolor(BLACK);
        outtextxy(x+12,y+11*select+15,menu[select]);
        switch(key){
     case UP_ARROW    : select=(select>0)?--select:2;
                        break;
     case DOWN_ARROW  : select=(select<2)?++select:0;
        break;
     case ESC         : select=3;
        }
    }while((key!=RETURN)&&(key!=ESC));
    putimage(x,y,oldimage,COPY_PUT);
    free(oldimage);free(bars);return(select);
  }
/*-------------------------------------------------------------------*/
void horline(int x, int y)
{ line(x+1,y+3,x+62,y+3);}
/*-------------------------------------------------------------------*/
void verline(int x, int y)
{ line(x+64,y+4,x+64,y-12);}
/*-------------------------------------------------------------------*/
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
void normal_open_contact(int x,int y)
  {
     moveto(x+1,y+3);lineto(x+23,y+3);outtextxy(x+20,y,"] [");
     moveto(x+39,y+3);lineto(x+62,y+3);
 }
/*-----------------------------------------------------------------*/
void normal_close_contact(x,y)
int x,y;
  {
     moveto(x+1,y+3);lineto(x+23,y+3);outtextxy(x+20,y,"]/[");
     moveto(x+39,y+3);lineto(x+62,y+3);
 }
/*-----------------------------------------------------------------*/
void energize_coil(x,y)
int x,y;
  {
     moveto(x+1,y+3);lineto(x+20,y+3);
     outtextxy(x+20,y,"( )");moveto(x+40,y+3);lineto(x+62,y+3);
  }
/*-----------------------------------------------------------------*/
void de_energize_coil(x,y)
int x,y;
  {
     moveto(x+1,y+3);lineto(x+20,y+3);outtextxy(x+20,y,"(/)");
     moveto(x+40,y+3);lineto(x+62,y+3);
  }
/*-----------------------------------------------------------------*/
void latch_coil(x,y)
int x,y;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    {
        moveto(x+1,y+3);lineto(x+20,y+3);outtextxy(x+20,y,"(L)");
        moveto(x+40,y+3);lineto(x+62,y+3);
    }
/*-------------------------------------------------------------*/
void unlatch_coil(x,y)
int x,y;
  {
      moveto(x+1,y+3);lineto(x+20,y+3);outtextxy(x+20,y,"(U)");
      moveto(x+40,y+3);lineto(x+62,y+3);
  }
/*-------------------------------------------------------------*/
void OFF_ON_transition_contact(x,y)
int x,y;
  {
     gprintf(x+27,y,"%c",24);moveto(x+1,y+3);
     lineto(x+20,y+3);outtextxy(x+20,y,"( )");
     moveto(x+40,y+3);lineto(x+62,y+3);
 }
/*-------------------------------------------------------------*/
void ON_OFF_transition_contact(x,y)
int x,y;
  {
     gprintf(x+27,y,"%c",25);moveto(x+1,y+3);
     lineto(x+20,y+3);outtextxy(x+20,y,"( )");
     moveto(x+40,y+3);lineto(x+62,y+3);
 }
/*-------------------------------------------------------------*/
void time(x,y)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
int x,y;
  {
     line(x+1,y+3,x+13,y+3);outtextxy(x+12,y,"[    ]");
     line(x+50,y+3,x+62,y+3);outtextxy(x+29,y,"T");
  }
/*-------------------------------------------------------------*/
void add(x,y)
int x,y;
  {
     line(x+1,y+3,x+13,y+3);outtextxy(x+12,y,"[    ]");
     outtextxy(x+29,y,"+");line(x+50,y+3,x+62,y+3);
  }

/*-------------------------------------------------------------*/
void sub(x,y)
ini x,y;
  {
     line(x+1,y+3,x+13,y+3);outtextxy(x+12,y,"[    ]");
     line(x+50,y+3,x+62,y+3);outtextxy(x+29,y,"-");
  }
/*-------------------------------------------------------------*/
void mul(x,y)
int x,y;
  {
     line(x+1,y+3,x+13,y+3);outtextxy(x+12,y,"[    ]");
     line(x+50,y+3,x+62,y+3);outtextxy(x+29,y-1,"x");
  }
/*-------------------------------------------------------------*/
void divs(x,y)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
int x,y;
  {
     line(x+1,y+3,x+13,y+3);outtextxy(x+12,y,"[    ]");
     line(x+50,y+3,x+62,y+3);circle(x+31,y+1,2);
     circle(x+31,y+5,2);line(x+25,y+3,x+37,y+3);
  }
/*-------------------------------------------------------------*/
void timeh(x,y)
int x,y;
  {
 line(x+1,y+3,x+13,y+3);outtextxy(x+12,y,"[    ]");
 line(x+50,y+3,x+62,y+3);outtextxy(x+24,y,"TH");
  }
/*-------------------------------------------------------------*/
void ctu(x,y)
int x,y;
  {
 line(x+1,y+3,x+13,y+3);outtextxy(x+12,y,"[CTU]");line(x+50,y+3,x+62,y+3);
  }
/*-------------------------------------------------------------*/
void ctd(x,y)
int x,y;
  {
 line(x+1,y+3,x+13,y+3);outtextxy(x+12,y,"[CTD]");line(x+50,y+3,x+62,y+3);
  }
/*-------------------------------------------------------------*/
void ctr(x,y)
int x,y;
  {
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
 line(x+1,y+3,x+13,y+3);outtextxy(x+12,y,"[CTR]");line(x+50,y+3,x+62,y+3);
  }
/*-------------------------------------------------------------*/
void get_word(x,y)
int x,y;
  {
 line(x+1,y+3,x+14,y+3);outtextxy(x+12,y,"]GET[");line(x+49,y+3,x+62,y+3);
  }
/*-------------------------------------------------------------*/
void put_word(x,y)
int x,y;
  {
 line(x+1,y+3,x+14,y+3);outtextxy(x+12,y,"]PUT[");line(x+49,y+3,x+62,y+3);
  }
/*-------------------------------------------------------------*/
void com_equal(x,y)
int x,y;
  {
line(x+1,y+3,x+11,y+3);outtextxy(x+9,y,"]CMP=[");line(x+53,y+3,x+62,y+3);
  }
/*-------------------------------------------------------------*/
void com_lessthan(x,y)
int x,y;
  {
 line(x+1,y+3,x+11,y+3);outtextxy(x+9,y,"]CMP<[");line(x+53,y+3,x+62 y+3);
  }
/*-------------------------------------------------------------*/
void com_geater(x,y)
int x,y;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    {
   line(x+1,y+3,x+11,y+3);outtextxy(x+9,y,"]CMP>[");line(x+53,y+3,x+62,y+3);
    }
  /*-------------------------------------------------------------*/
  void textlogic_relay(int x,int y)
  {
     setcolor(LIGHTRED);
     rectangle(x,y,x+75,y+125);
     line(x,y+32,x+75,y+32);
     outtextxy(x+20,y+2,"LOGIC");
     outtextxy(x+20,y+13,"  &  ");
     outtextxy(x+20,y+24,"RELAY");
     setcolor(LIGHTGRAY);
     normal_open_contact(x+6,y+37);
     normal_close_contact(x+6,y+48);
     energize_coil(x+6,y+59);
     de_energize_coil(x+6,y+70);
     latch_coil(x+6,y+81);
     unlatch_coil(x+6,y+92);
     OFF_ON_transition_contact(x+6,y+103);
     ON_OFF_transition_contact(x+6,y+114);
  }
  /*-------------------------------------------------------------*/
  void time_count_math(int x,int y)
  {
     setcolor(LIGHTRED);
     rectangle(x,y,x+75,y+140);
     line(x,y+32,x+75,y+32);
     outtextxy(x+4,y+2, "  TIMER");
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        outtextxy(x+4,y+13," COUNTER");
        outtextxy(x+4,y+24," + - x /");
        setcolor(LIGHTGRAY);
        time(x+6,y+37);
        timeh(x+6,y+48);
        ctu(x+6,y+59);
        ctd(x+6,y+70);
        ctr(x+6,y+81);
        add(x+6,y+92);
        sub(x+6,y+103);
        mul(x+6,y+114);
        divs(x+6,y+125);
    }
    /*-------------------------------------------------------------------*/
    void compare_move(int x,int y)
    {
        setcolor(LIGHTRED);
        rectangle(x,y,x+75,y+100);
        line(x,y+32,x+75,y+32);
        outtextxy(x+10,y+2, "COMPARE");
        outtextxy(x+10,y+13,"   & ");
        outtextxy(x+10,y+24," MOVE");
        setcolor(LIGHTGRAY);
        com_equal(x+6,y+37);
        com_lessthan(x+6,y+48);
        com_geater(x+6,y+59);
        get_word(x+6,y+70);
        put_word(x+6,y+81);
    }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
/*-------------------------------------------------------------*/
char logic_relay(int x, int y,char status,char gr)
/* if status equal OFF function is not select menu but show message only */
{
   char select,maxselect;
   static char initial;
   int key;
   unsigned size;
   select = 0;
   initial = initial + 0;
   if (status == OFF){
      /*--------creat box menu*-----------*/
      setcolor(BROWN);
      bar(x+5,y+36,x+70,y+44);
      if (initial != 1){
         size = imagesize(x+5,y+36,x+70,y+44);
  ladder = malloc(size);
         initial = 1;
      }
      getimage(x+5,y+36,x+70,y+44,ladder);
      putimage(x+5,y+36,ladder,XOR_PUT);
      /*-----------------------------*/
      switch(gr){
case 0:textlogic_relay(x,y);break;
case 1:time_count_math(x,y);break;
case 2:compare_move(x,y);break;
      }
   }
   if (status == ON){
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        switch(gr){
  case 0:maxselect=7;break;
  case 1:maxselect=8;break;
  case 2:maxselect=4;break;
        }
     do{

        key = bioskey(0);
        key>>=8;
        putimage(x+5,y+11*select+36,ladder,XOR_PUT);
        switch(key){
     case UP_ARROW    : select = (select > 0) ? --select : maxselect;
                                 break;
     case DOWN_ARROW  : select = (select < maxselect) ? ++select :0;
                                 break;
        }
    } while (key != RETURN);
    putimage(x+5,y+11*select+36,ladder,XOR_PUT);
    return(select);
    }
  }
/*-------------------------------------------------------------*/
void activeviewport(char select)
{
    int driver,mode;
    int sX=0;
    int sY=0;
     detectgraph(&driver,&mode);
    if(driver==HERCMONO){
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
 sX=40;
 sY=74;
    }
  switch(select){
  case  BIGWINDOW :  {
     setviewport(4+sX,13+sY,550+sX,186+sY,0);
     setviewport(555+sX,10+sY,637+sX,185+sY,0);
     setviewport(2+sX,2+sY,638+sX,198+sY,1);break;
                  }
  case  MIDWINDOW :  {
     setviewport(555,10+sY,637+sX,185+sY,0);
     setviewport(2+sX,2+sY,638+sX,198+sY,0);
     setviewport(4+sX,13+sY,550+sX,186+sY,1);break;
                  }
  case  SMALLWINDOW :  {
     setviewport(4+sX,13+sY,550+sX,186+sY,0);
     setviewport(2+sX,2+sY,638+sX,198+sY,0);
     setviewport(555+sX,10+sY,637+sX,184+sY,1);
    }
 }
}
/*-------------------------------------------------------------*/
void displaymenu(int y,char *str_f,char *str_b)
{
  int g_driver,g_mode;
 char *top[]={"Mainmenu    Ladder Group    Baud rate      ESC = cancel",
      "M           L              B              ESC"
     };
 char *down=
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    "Major of COMPUTER INDUSTRIAL  FACULTY of ENGINEERING   K M I T 'L  ";
    detectgraph(&g_driver,&g_mode);
    activeviewport(BIGWINDOW);
    if ((g_driver == EGA) || (g_driver == VGA))
       setcolor(MAGENTA);
    bar(7,y,545,y+8); setcolor(BLACK);

    if(*str_f==NULL){
       if(y==UP)outtextxy(20,y+1,top[0]);else outtextxy(20,y+1,down);
    }
    else outtextxy(20,y+1,str_f);
    if ((g_driver == EGA) || (g_driver == VGA)) setcolor(RED);
    if(*str_f==NULL){
        if(y==UP)outtextxy(20,y+1,top[1]);
        else outtextxy(20,y+1,down);
    }
   else if(*str_b!=NULL) outtextxy(20,y+1,str_b);
   activeviewport(MIDWINDOW);
}
/*-------------------------------------------------------------------*/
void boxmid(void)
{
 activeviewport(BIGWINDOW); setcolor(GREEN);rectangle(1,11,551,185);
}
/*-------------------------------------------------------------------*/
void start(void)
{
 displaymenu(UP,NULL,NULL);displaymenu(UNDER,NULL,NULL);
 boxmid(); activeviewport(SMALLWINDOW);logic_relay(3,9,OFF,0);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
 * select ladder on right hand in editor work *
 *********************************************/
char instruction(char *GROUP)
{
   void *inst;
   unsigned size;
   char ins;
   activeviewport(SMALLWINDOW);
   ins = logic_relay(3,9,ON,group(OFF));
   *GROUP = group(OFF);
   return(ins);
}
/*-------------------------------------------------------------*/
void cleareditor(char updown)
{
   int driver,mode;
   int sX=0;
   int sY=0;
    detectgraph(&driver,&mode);
   if(driver==HERCMONO){sX=40;sY=74;}
     setviewport(4+sX,13+sY 550+sX,186+sY,0);
     setviewport(2+sX,2+sY,638+sX,198+sY,0);
     setviewport(555+sX,10+sY,637+sX,184+sY,0);
     if (updown == UP){                    /* when push up */
 setviewport(4+sX,14+sY,41+sX,33+sY,1);
         clearviewport();                  /* del number of line */
 setviewport(4+sX,14+sY,41+sX,33+sY,0);
 setviewport(48+sX,14+sY,539+sX,33+sY,1);
         clearviewport();
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    setviewport(48+sX,14+sY,539+sX,33+sY,0);
       }
       else{                                  /* when push key down */
    setviewport(48+sX,161+sY,538+sX,186+sY,1);
            clearviewport();
    setviewport(48+sX,161+sY,538+sX,186+sY,0);
    setviewport(4+sX,161+sY,41+sX,186+sY,1);
            clearviewport();
    setviewport(4+sX,161+sY,41+sX,186+sY,0);
       }
       activeviewport(MIDWINDOW);
}
 /*-------------------------------------------------------------------------*/
void oldlineUPDOWN(char status,TABLE *position,int rows,char y)
{
 char i = 0;
 char select,o;
 if (status==ON){
     if((y==0)&&(rows!=0))--rows;else ++rows;
 }
 for(i=0;i<=7;i++){
     if ((position+rows)->data[i] != -1){
 setcolor(GREEN);
 if ((position+rows)->data[i]==HORLINE) horline(62*i+41,16*y+13);
 if(((position+rows)->data[i]<=17)&&((position+rows)->data[i]>9)){
     switch((position+rows)->data[i]){
case  10 : normal_open_contact(62*i+41,16*y+13);break;
case  11 : normal_close_contact(62*i+41,16*y+13);break;
case  12 : energize_coil(62*i+41,16*y+13);break;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
case  13 : de_energize_coil(62*i+41,16*y+13);break;
case  14 : latch_coil(62*i+41,16*y+13);break;
case  15 : unlatch_coil(62*i+41,16*y+13);break;
case  16 : OFF_ON_transition_contact(62*i+41,16*y+13);break;
case  17 : ON_OFF_transition_contact(62*i+41,16*y+13);break;
      } /* end switch */
 }/* end if */
if(((position+rows)->data[i]<=28)&&((position+rows)->data[i]>19)){
     switch((position+rows)->data[i]){
case  20 : time(62*i+41,16*y+13);break;
case  21 : timeh(62*i+41,16*y+13);break;
case  22 : ctu(62*i+41,16*y+13);break;
case  23 : ctd(62*i+41,16*y+13);break;
case  24 : ctr(62*i+41,16*y+13);break;
case  25 : add(62*i+41,16*y+13);break;
case  26 : sub(62*i+41,16*y+13);break;
case  27 : mul(62*i+41,16*y+13);break;
case  28 : divs(62*i+41,16*y+13);break;
      } /* end switch */
}/* end if*/
if(((position+rows)->data[i]<=34)&&((position+rows)->data[i]>29)){
     switch((position+rows)->data[i]){
case  30 : com_equal(62*i+41,16*y+13);break;
case  31 : com_lessthan(62*i+41,16*y+13);break;
case  32 : com_geater(62*i+41,16*y+13);break;
case  33 : get_word(62*i+41,16*y+13);break;
case  34 : put_word(62*i+41,16*y+13);break;
      } /* end switch */
}/* end if*/
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        } /* end check had data */
  /*--------------- if no data then end here ------------*/
      if ((position+rows)->value[i] != -1){
     /*************************
      * print value to screen *
      *************************/
     setcolor(RED);
     select=0;o=0;
     while(select<=39)
        if((position+rows)->value[i]==channal[select++])break;
     while(o<=7)
        if((position+rows)->value1[i]==bit[o++])break;
    gprintf(62*i+58,16*y+5,"%s%s",channaltext[--select],bittext[--o]);

    }/* end if print value */
     if (((position+rows)->data[i] >= 20) && ((position+rows)->data[i]<=23))
       gprintf(62*i+58,16*(y+1)+5,"#%03d",(unsigned char)((position+rows)->unde1
     setcolor(RED);
     if(i<=6){
         if((position+rows)->col[i]==VERLINE)
      verline(62*i+40,16*y+12);
 if((position+rows+1)->col[i]==VERLINE)
     verline(62*i+40,16*(y+1)+12);
    }/* end if  */
  }  /* end for */
} /* end function */
/*-------------------------------------------------------------*/
void scroll(char updown,int rows,TABLE *position)
{
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    unsigned  size;
    void      *oldimage;
    int i,j;
     int g_driver,g_mode;int sX=0;int sY=0;
     detectgraph(&g_driver,&g_mode);
     if(g_driver==HERCMONO){
         sX=40;sY=74;
     }
     activeviewport(MIDWINDOW);
     if(g_driver!=CGA){
   setviewport(4+sX,14+sY,550+sX,186+sY,1);
   clearviewport();
   setviewport(4+sX,14+sY,550+sX,186+sY,0);
   activeviewport(MIDWINDOW);
   line(41,1,41,173);
   line(537,1,537,173);
     }
    switch(updown){
case DOWN :  if(g_driver!=CGA){
      setcolor(GREEN);
      j=0;
for (i=rows-9;i<=rows+1;i++){
   gprintf(38,j*16+12,"%c",07);
    gprintf(534,j*16+12,"%c",07);
   gprintf(4,j*16+12,"%3d",i);
   ++j;
}
for(i=0;i<=9;i++)
    oldlineUPDOWN(OFF,position,rows-9+i,i);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        }
       if(g_driver==CGA){
          size = imagesize(1,17,550,170);
          oldimage = malloc(size);
          getimage(1,17,550,170,oldimage);
          putimage(1,1,oldimage,COPY_PUT);
          cleareditor(DOWN);
          free(oldimage);
          oldlineUPDOWN(ON,position,rows,9);
       } break;

 case  UP   : j=0;
      if(g_driver!=CGA){
   setcolor(GREEN);
   for (i=rows;i<=rows+9;i++){
       gprintf(38,j*16+12,"%c",07);
       gprintf(534,j*16+12,"%c",07);
       gprintf(4,j*16+12,"%3d",i);
       ++j;
    }
 for(i=0;i<=9;i++)
       oldlineUPDOWN(OFF,position,rows+i,i);
     }
     if(g_driver==CGA){
size = imagesize(1,4,550,157);
oldimage = malloc(size);
getimage(1,4,550,157,oldimage);
cleareditor(UP);
putimage(1,20,oldimage,COPY_PUT);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
free(oldimage);
oldlineUPDOWN(ON,position,rows,0);
    }break;
      }
}
/*-----------------------------------------------------------------*/
void values(int *row,int *col,TABLE *position, int *box, int *rows)
{
   int key;

   char s = 0;char o = 0;char si;
   setcolor(BLACK);
do{
   putimage(62*(*col)+40,16*(*row)+4,box,COPY_PUT);
   gprintf(62*(*col)+58,16*(*row)+5,"%s%s",channaltext[s],bittext[o]);
   key = bioskey(0);
   key>>=8;
   switch(key){
       case LEFT_ARROW  :   s = (s > 0) ? --s : 39;break;
       case RIGHT_ARROW :   s = (s < 39) ? ++s : 0;break;
       case UP_ARROW    :   o = (o > 0) ? --o : 7;break;
       case DOWN_ARROW  :   o = (o < 7) ? ++o : 0;break;
       case PLUS        :   if(s<=7)si=8;
   if((s<=15)&&(s>=8))si=16;
   if((s<=23)&&(s>=16))si=24;
   if((s<=31)&&(s>=24))si=32;
   if((s<=39)&&(s>=32))si=0;
   s=si;break;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        case  LINE          :   if(s<=7)si=32;
      if((s<=15)&&(s>=8))si=0;
      if((s<=23)&&(s>=16))si=8;
      if((s<=31)&&(s>=24))si=16;
      if((s<=39)&&(s>=32))si=24;
      s=si;break;
     }
    }while((key!=RETURN)&&(key!=DEL));
    setcolor(GREEN);
    putimage(62*(*col)+40,16*(*row)+4,box,COPY_PUT);
    putimage(62*(*col)+40,16*(*row)+4,box,XOR_PUT);
    if (key!=DEL){
        setcolor(RED);
        sprintf(62*(*col)+58,16*(*row)+5,"%s%s",channaltext[s],bittext[o]);
        (position+(*rows))->value[*col] =  channal[s];
        (position+(*rows))->value1[*col] = bit[o];
    }
    if(key==DEL){
        (position+(*rows))->value[*col] =  -1;
        (position+(*rows))->value1[*col] =  -1;
    }
    setcolor(GREEN);
    if (((position+(*rows))->col[*col-1] != -1)&&(*col!=0))
        verline(62*(*col-1)+40,16*(*row)+12);
    if (((position+(*rows))->col[*col] != -1)&&(*col != MAXCOL))
        verline(62*(*col)+40,16*(*row)+12);
    if (*col == 0)
        verline(-23,16*(*row)+12);
    if (*col == MAXCOL)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        verline(62*MAXCOL+39,16*(*row)+12);
  }
  /*-------------------------------------------------------------*/
  void downform(int start)
    {
     int g_driver,i;int sX=0;int sY=0;
     detectgraph(&g_driver,&i);if(g_driver==HERCMONO){sX=40;sY=74;}
     setviewport(4+sX,14+sY,550+sX,186+sY,1);
     clearviewport();
     setviewport(4+sX,14+sY,550+sX,186+sY,0);
     activeviewport(MIDWINDOW);
     setcolor(GREEN);
     line(41,1,41,173);
     line(537,1,537,173);
    for (i=0;i<=9;i++){
  gprintf(38,i*16+12,"%c",07);
  gprintf(534,i*16+12,"%c",07);
  gprintf(4,i*16+12,"%3d",i+start);
    }
  }
  /*-------------------------------------------------------------*/
  char *load(char status,TABLE *position,unsigned int *datarows,char *re_esc)
  {
  static char name[45];
  char   *bufname;
  FILE *fname,*fopen();
  int j;long n=5L;int i,greturn;unsigned int data;char newfile;
  disable();
  *re_esc=NO;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
/* part  1-------------------------------------------------------------*/
if(status==ON){ /* if ON then enter file name else return file name */
    greturn=gscanf(ON,Xg,20,name,"File name");
    if(greturn!=0x1b){

      /************************
       * Clear old data memory *
       ************************/
       for (i=0;i<=(MAXROWS-1);i++)
    for (j=0;j<=7;j++){
(position+i)->data[j]=-1;
(position+i)->value[j]=-1;
(position+i)->value1[j]=-1;
    }
       for (i=0;i<= (MAXROWS-1);i++){
    (position+i)->under=-1;
    for (j=0;j<=6;j++)
(position+i)->col[j]=-1;
       }
       /*********************************
*          Open file              *
*********************************/
if(greturn==NO){  /* when no enter file name form user */
       fname=fopen("NONAME.LAD","rb");
       newfile=(fname==NULL)?YES:NO;
       bufname="NONAME.LAD";
       i=0;
       while((name[i++]=*bufname++)!='\0')
    ;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
}
        /*-------------------------*/
if(greturn==YES){  /* when user enter file name */
        fname=fopen(name,"rb");
        newfile=(fname==NULL)?YES:NO;
 }
 /************** end of OPEN FILE **************/
        downform(0);
  /*-------------------------------------------------------------------*
        when had data file on disk
    -------------------------------------------------------------------*/
 /*********************************
  * Load number of data form file  *
  *********************************/
  if(newfile==NO){ /* Had file on disk */
  fseek(fname,0L,0);
  fread((char *)&data,sizeof(unsigned int),1,fname);
      }
  if((newfile==NO)&&(data<=MAXROWS)){
  /*********************************
   *      Load data form file      *
   *********************************/
  fseek(fname,n,0);
                          i=0;
  while(i<=data){
fread((char *)(position+i)->data,sizeof(char),8,fname);
  ++i;
  }
  i=0;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        disable();
        fn=fopen(name,"wb+");
        fseek(fn,0L,0);
        fwrite((char *)&data,sizeof(unsigned int),1,fn);fseek(fn,n,0);i=0;
        while(i<=data){
    fwrite((char *)(position+i)->data,sizeof(char),8,fn);++i;
        } i=0;
        while(i<=data){
          fwrite((char *)(position+i)->col,sizeof(char),7,fn);++i;
        } i=0;
        while(i<=data){
          fwrite((char *)(position+i)->value,sizeof(char),8,fn);++i;
        }  i=0;
        while(i<=data){
          fwrite((char *)(position+i)->value1,sizeof(char),8,fn);++i;
        }
        while(i<=data){
          fwrite((char *)(position+i)->under,sizeof(char),1,fn);++i;
        }
          fclose(fn);
        enable();
        displaymenu(UNDER,NULL,NULL);
      }/* end of if gsreturn */
    }  /* end of function */

/*-------------------------------------------------------------*/
void headerMCS51(FILE *mcs)
{
 fprintf(mcs,";PLC version 1.0 (Test version) for CP-32 board\n");
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    fprintf(mcs,";By WITTAYA CHEESAKUL  KMITL\n\n");
    fprintf(mcs,"ORG  %s\n",ORG);
    /* control word 8255 */
    fprintf(mcs,"U8255MODE   EQU 91H\n");
    /* control port CH1 */
    fprintf(mcs,"CONTROLCH1  EQU %s\n",CONTROLCH1);
    fprintf(mcs,"MOV  A,#U8255MODE\n");
    fprintf(mcs,"MOV  DPTR,#CONTROLCH1\n");
    fprintf(mcs,"MOVX @DPTR,A\n");
    fprintf(mcs,"START:\n");
  }
  /*-------------------------------------------------------------*/
  void end_of_mcs51(FILE *mcs)
  {
   fprintf(mcs,"LJMP START\n");

   fprintf(mcs,"END\n");
  }
  /*-------------------------------------------------------------*/
  int check_ladder_have(int *rows,TABLE *position)
  {
    char i;int  first_rows_rung=*rows;char has;
    do{
        i=0;
        while((i<=7)&&((has=(position+first_rows_rung)->data[i])==-1))
    ++i;++first_rows_rung;
    }while((first_rows_rung<=MAXROWS)&&(has==-1));
    return(--first_rows_rung);
  }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
/*-------------------------------------------------------------------*/
int check_or(int *first,TABLE *position)
{
  char i;int  last_rows_rung=*first+1;
  do{
       i=0;
       while((i<=6)&&(((position+last_rows_rung)->col[i])==-1))
  ++i;++last_rows_rung;
  }while((last_rows_rung<MAXROWS)&&(i!=7)); /*if i==7 no or in rows*/
  return(--last_rows_rung);
}
/*-------------------------------------------------------------------*/
void contact(or_and,col,step,temp,relay,port,bit,mcs,position,first,last)
char or_and;  /* check AND or OR operator */
int  col;     /* current collump  of bit addressable */
int  *step;
int  temp;    /* current temporaly of rung */
char relay;   /* type of ladder */
char port ;   /* channal */
char bit;     /* bit of channal */
FILE *mcs;    /* file pointer of mcs-51 assembly */
TABLE *position;
int  first;
int  last;
{

 int start,i;
 if(or_and==AND){
   switch(relay){
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
case OFF_ON_TRANSITION:
case ON_OFF_TRANSITION:
if(relay==OFF_ON_TRANSITION)
   fprintf(mcs,"MOV  DPTR,%s\n",OFF_ON[port]);
else
   fprintf(mcs,"MOV DPTR,%s\n",ON_OFF[port]);
fprintf(mcs,"MOVX A,@DPTR\n");
fprintf(mcs,"MOV  20H,A\n");
fprintf(mcs,"MOV C,%s\n",bitbuffer[bit]);
fprintf(mcs,"MOV 0Fh,C\n");
fprintf(mcs,"MOV C,%s\n",temporaly[temp][col-1]);
fprintf(mcs,"MOV %s,C\n",bitbuffer[bit]);
fprintf(mcs,"MOV A,20H\n");
fprintf(mcs,"MOVX @DPTR,A\n");
/*---------------------------*/
fprintf(mcs,"MOV DPTR,%s\n",portname[port]);
fprintf(mcs,"MOVX A,@DPTR\n");
fprintf(mcs,"MOV 20H,A\n");
if(relay==OFF_ON_TRANSITION)
    fprintf(mcs,"ANL C,/0FH\n");
 else{ fprintf(mcs,"CPL C\n");
       fprintf(mcs,"ANL C,0FH\n");
 }
fprintf(mcs,"MOV %s,C\n",bitbuffer[bit]);
fprintf(mcs,"MOV A,20H\n");
fprintf(mcs,"MOVX @DPTR,A\n");
                        if(port<=7){
    fprintf(mcs,"MOV  DPTR,%s\n",portoutput[port]);
    fprintf(mcs,"MOVX @DPTR,A\n");
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
  } break;
  case NORMAL_OPEN  :
  case NORMAL_CLOSE :
  fprintf(mcs,"MOV  DPTR,%s\n",portname[port]);
  fprintf(mcs,"MOVX A,@DPTR\n");
  fprintf(mcs,"CPL  A\n");
  fprintf(mcs,"MOV  20H,A\n");
  if(col==0)
     fprintf(mcs,"MOV  C,%s\n",bitbuffer[bit]);
  else{
      fprintf(mcs,"MOV C,%s\n",temporaly[temp][col-1]);
      if(relay==NORMAL_OPEN)
         fprintf(mcs,"ANL C,%s\n",bitbuffer[bit]);
      else fprintf(mcs,"ANL C,/%s\n",bitbuffer[bit]);
  }
  if((relay==NORMAL_CLOSE)&&(col==0))
     fprintf(mcs,"CPL C\n");
  fprintf(mcs,"MOV  %s,C\n",temporaly[temp][col]);
  break;
  case ENERGIZE_COIL:
  case DE_ENERGIZE_COIL:
  fprintf(mcs,"MOV  DPTR,%s\n",portname[port]);
  fprintf(mcs,"MOVX A,@DPTR\n");
  fprintf(mcs,"MOV  20H,A\n");
  fprintf(mcs,"MOV  C,%s\n",temporaly[temp][col-1]);
                        if(relay==DE_ENERGIZE_COIL)
      fprintf(mcs,"CPL  C\n");
  fprintf(mcs,"MOV  %s,C\n",bitbuffer[bit]);
  fprintf(mcs,"MOV  A,20H\n");
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    fprintf(mcs,"MOVX @DPTR,A\n");
    if(port<=7){
         fprintf(mcs,"MOV  DPTR,%s\n",portoutput[port]);
         fprintf(mcs,"MOVX @DPTR,A\n");
     } break;
    case LATCH      :
    case UN_LATCH   :
    fprintf(mcs,"MOV DPTR,%s\n",latch[port]);
    fprintf(mcs,"MOVX A,@DPTR\n");
    fprintf(mcs,"MOV 20H,A\n");
    fprintf(mcs,"MOV C,%s\n",temporaly[temp][col-1]);
    if(relay==LATCH)
       fprintf(mcs,"ORL C,%s\n",bitbuffer[bit]);
    else{
       fprintf(mcs,"CPL C\n");/* complement rung*/
       fprintf(mcs,"ANL C,%s\n",bitbuffer[bit]);
    }
    fprintf(mcs,"MOV %s,C\n",bitbuffer[bit]);
    fprintf(mcs,"MOV A,20H\n");
    fprintf(mcs,"MOVX @DPTR,A\n");
    fprintf(mcs,"MOV  R0,A\n");
    fprintf(mcs,"MOV DPTR,%s\n",portname[port]);
    fprintf(mcs,"MOVX A,@DPTR\n");
    fprintf(mcs,"MOV 20H,A\n");
    fprintf(mcs,"CLR %s\n",bitbuffer[bit]);
    fprintf(mcs,"MOV A,20H\n");
    fprintf(mcs,"ORL A,R0\n");
    fprintf(mcs,"MOVX @DPTR,A\n");
    if(port<=7){
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        fprintf(mcs,"MOV  DPTR,%s\n",portoutput[port]);
        fprintf(mcs,"MOVX @DPTR,A\n");
     } break;
    case HORLINE    :if(col==0)
       fprintf(mcs,"SETB %s\n",temporaly[temp][col]);
     else{
       fprintf(mcs,"MOV C,%s\n",temporaly[temp][col-1]);
       fprintf(mcs,"MOV %s,C\n",temporaly[temp][col]);
     } break;
    case -1/*no*/   :if((col<=6)&&((position+*step)->col[col]!=-1))
     fprintf(mcs,"CLR %s\n",temporaly[temp][col]);break;
        }/* end switch */
     }/*end if or_and*/
     /*-------------------------------------------------------------------------*/
    char compile_to_hexformat(TABLE *position)
    {
     int  rows=0;
     int  first_rows_rung,last_rows_rung;
     int  step;
     int  col;
     int  relay;
     FILE *f,*fopen();
     int temp,i;
     char port,bit;
     char name[50];
     char *bufname,*load();
     unsigned int r=0;
     bufname=load(OFF,position,&r,&bit);
     i=4;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
      getch();
       displaymenu(UNDER,NULL,NULL);
      return(-1);
     }
    system(name);
    if(check==NO){
      name[0]='d';name[1]='e';name[2]='l';
      name[3]=' ';
      system(name);
    }
    displaymenu(UNDER,NULL,NULL);
    for(i=0;i<=1;i++){
   sound(300);delay(300);nosound();delay(300);
        } nosound();
   }
   /*-------------------------------------------------------------------*/
   char setcom1(void)
    {
    int out,key;
    char *reset="Reset PLC and Press any key....     <ESC> to cancel..";
    char *wait= "Wait . . .     transfer code";
  bioscom(0,SETTING,portCOM);
  while((out=bioscom(3,0,portCOM))&DATA_READY)bioscom(2,0,portCOM);
  do{
      bioscom(1,'a',portCOM);out=bioscom(2,0,portCOM);
      if (out!='a'){
  sound(1000);
  delay(500);
  nosound();displaymenu(UNDER,reset,NULL);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
  key=bioskey(0);
  if((key>>=8)==ESC)break;
  bioscom(1,0x20,portCOM);bioscom(2,0,portCOM);
  while((out=bioscom(2,0,portCOM) & 0x7f)!=0)
       ;
      }
    }while(out != 'a');
    if(key==ESC)return(OFF);
    displaymenu(UNDER,wait,NULL);
    bioscom(1,0x0d,portCOM);
    while((out=bioscom(2,0,portCOM) & 0x7f)!=0)
      ;
    return(ON);
}

/*-------------------------------------------------------------------*/
void download(void)
{
  char load[7] = {
       'D','L','O','A','D',0x0d,0x0d
       };
 int i;
/* int out;*/
for(i=0;i<=6;i++){
   bioscom(1,load[i],portCOM);
}
    while((bioscom(2,0,portCOM) & 0x7f)!=0)
      ;
     }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
/*-------------------------------------------------------------*/
void runCP32(void)
    {
char run[11] = {
        'a','r','u','n',' ','8','0','0','0','h',0x0d
       };
int i;
/*int register out;*/

/*********************************
 * send ARUN 8000h to CP32      *
 *********************************/
for(i=0;i<=10;i++)
   bioscom(1,run[i],portCOM);
     }

/*-------------------------------------------------------------*/
void getfile(char *name)
{
  FILE *fname;int ch;
/*  int register out;*/
  if((fname=fopen(name,"rt"))!=NULL){
     while((ch=getc(fname))!=EOF){
bioscom(1,ch,portCOM);    /*send data to rs232 */
     }
     while((bioscom(2,0,portCOM) & 0x7f)!=0)
;
     fclose(fname);
  }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
}
/*-------------------------------------------------------------*/
char transfer(TABLE *position)
  {
 char set;
  char *bufname,name[45],gsreturn,i;unsigned int r;
    bufname=load(OFF,position,&r,&i);
    gsreturn=gscanf(OFF,Xg,20,bufname,"File name");i=0;
    while((name[i++]=*bufname++)!='.')
 ;
    name[i]='H';name[i+1]='E';name[i+2]='X';name[i+3]='\0';
    if(gsreturn){
       set=setcom1();
       if(set){
  download();getfile(name);runCP32();
       }
    }
    displaymenu(UNDER,NULL,NULL);
    if(!gsreturn||!set)return(-1);
    for(i=0;i<=1;i++){
sound(600);delay(500);nosound();delay(500);
     }
    nosound();
  }
/*-------------------------------------------------------------*/
void setbaudrate(int x,int y)
 {
   char *rate[]={
      "110","150","300","600","1200","2400","4800","9600"
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    };
    char i;int key;char select;
    void *bars,*oldimage;unsigned size;
    select = 0;
    activeviewport(MIDWINDOW);
       /*-----save old image ---------*/
size=imagesize(x,y,x+50,y+105);oldimage=malloc(size);
getimage(x,y,x+50,y+105,oldimage);putimage(x,y,oldimage,XOR_PUT);
/*-----------------------------*/
setcolor(BLACK);bar(x,y,x+50,y+105);
line(x,y+11,x+50,y+11);
outtextxy(x+8,y+2,"Baud");
         /*-----create box bar menu------*/
size=imagesize(x+5,y+12,x+45,y+20);bars=malloc(size);
getimage(x+5,y+12,x+45,y+20,bars);
/*-----------------------------*/
for(i=0;i<=7;i++)outtextxy(x+10,y+(i*11)+16,rate[i]);
   do{
       putimage(x+5,y+11*select+15,bars,XOR_PUT);
       setcolor(WHITE);
       outtextxy(x+10,y+11*select+16,rate[select]);
       key=bioskey(0);key>>=8;
       putimage(x+5,y+11*select+15,bars,XOR_PUT);
       setcolor(BLACK);
       outtextxy(x+10,y+11*select+16,rate[select]);
       switch(key){
   case UP_ARROW     : select=(select>0)?--select:7;break;
   case DOWN_ARROW   : select=(select<7)?++select:0;break;
   case ESC          : select=ESC;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        }
    }while((key!=RETURN)&&(key!=ESC));
    putimage(x,y,oldimage,COPY_PUT);free(oldimage);free(bars);
    if(key==RETURN){
        baud=bauds[select];
        SETTING=(baud|0x03|0x00|0x00);
    }
}
/*-------------------------------------------------------------------*/
char under(int *row,int *col,TABLE *position, int *box, int *rows)
{
   int key;
   unsigned char s = 1;
   setcolor(BLACK);
do{
   putimage(62*(*col)+40,16*(*row+1)+4,box,COPY_PUT);
   gprintf(62*(*col)+58,16*(*row+1).5,"#%03d",s);
   key = bioskey(0);
   key>>=8;
   switch(key){
       case DOWN_ARROW  :   s = (s > 1) ? --s : 255;break;
       case UP_ARROW    :   s = (s < 255) ? ++s : 1;break;
       case PLUS        :   s=(s<246)?s+=10:1;break;
       case LINE        :   s=(s>10)?s-=10:255;break;
  }
}while((key!=DEL)&&(key!=RETURN));
setcolor(RED);
putimage(62*(*col)+40,16*(*row+1)+4,box,COPY_PUT);
putimage(62*(*col)+40,16*(*row+1)+4,box,XOR_PUT);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    if (key!=DEL){
        gprintf(62*(*col)+58,16*(*row+1)+5,"#%03d",s);
        (position+(*rows))->under =  s;
    }
    if(key==DEL){
        (position+(*rows))->under =  -1;
    }
    setcolor(GREEN);
    if (((position+(*rows+1))->col[*col-1] != -1)&&(*col!=0))
        verline(62*(*col-1)+40,16*(*row+1)+12);
    if (((position+(*rows+1))->col[*col] != -1)&&(*col != MAXCOL))
        verline(62*(*col)+40,16*(*row+1)+12);
    if (*col == 0)
        verline(-23,16*(*row+1)+12);
    if (*col == MAXCOL)
        verline(62*MAXCOL+39,16*(*row+1)+12);
}
/*-------------------------------------------------------------------*/
char mainselect(int x,int y,char init,TABLE *position,
unsigned int *datarows,int *rows,int *row)
 {
   char *menu[]={
      "LOAD FILE","SAVE FILE","COMPILE","TRANSFER","QUIT"
   };
   char i;int key;char select;char re_esc=NO;
   void *bars,*oldimage;unsigned size;
   select = 0;
   activeviewport(MIDWINDOW);
   if(init==NO){
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        /*-----save old image ---------*/
 size=imagesize(x,y,x+96,y+70);oldimage=malloc(size);
 getimage(x,y,x+96,y+70,oldimage);putimage(x,y,oldimage,XOR_PUT);
 /*-----------------------------*/
 setcolor(BLACK);
 bar(x,y,x+96,y+70);
 line(x,y+11,x+96,y+11);
 outtextxy(x+8,y+2,"MAINSELECT");
          /*-----create box bar menu------*/
 size=imagesize(x+5,y+12,x+91,y+20);
 bars=malloc(size);
 getimage(x+5,y+12,x+91,y+20,bars);
 /*-------------------------------*/
 for(i=0;i<=4;i++)outtextxy(x+10,y+(i*11)+16,menu[i]);
    do{
        putimage(x+5,y+11*select+15,bars,XOR_PUT);
        setcolor(WHITE);
        outtextxy(x+10,y+11*select+16,menu[select]);
        key=bioskey(0);key>>=8;
        putimage(x+5,y+11*select+15,bars,XOR_PUT);
        setcolor(BLACK);
        outtextxy(x+10,y+11*select+16,menu[select]);
        switch(key){
    case UP_ARROW    : select=(select>0)?--select:4;break;
    case DOWN_ARROW  : select=(select<4)?++select:0;break;
       }
 }while((key!=RETURN)&&(key!=ESC));
 putimage(x,y,oldimage,COPY_PUT);free(oldimage);free(bars);
 if(key==ESC)return(ESC);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        switch(select){
    case 0 :load(ON,position,datarows,&re_esc);
    if(re_esc==NO)*rows=*row=0;break;
    case 1 :save(position,datarows);break;
    case 2 :compile_to_hexformat(position);break;
    case 3 :transfer(position);break;
   case 4 :return(Q);
         }
    return(select);
  }else{ /* if init ==YES */
       load(ON,position,datarows,&re_esc);*rows=*row=0;return(-1);
     }
}
/*-------------------------------------------------------------------*/
editorwork()
{
 void *box,*inst,*oldimage,*horline,*delete;
 unsigned size;char exit_place,GROUP;
 int  ins,i,j,g_driver,g_mode,key,row,col,rows,k;
 unsigned int datarows=0;void *buffer=0;
 TABLE position[MAXROWS+1];int driver,mode;
 detectgraph(&driver,&mode);
 key=row=col=rows=datarows=0;
 size = imagesize(8,36,73,44);
 inst = malloc(size);
 box  = malloc(size);
 oldimage = malloc(size);
 horline  = malloc(size);
 delete   = malloc(size);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    for (i=0;i<= (MAXROWS-1);++i)
        for (j=0;j<=7;++j){
          position[i].data[j] = -1;position[i].value[j] = -1;
          position[i].value1[j] = -1;
        }
    for (i=0;i<= (MAXROWS-1);++i){
        position[i].under = -1;
        for (j=0;j<=6;++j) position[i].col[j]  = -1;
    }
    activeviewport(MIDWINDOW);setvisualpage(0);
    setcolor(GREEN);
    for(j=0;j<=1;j++){
       setactivepage(j);line(41,1,41,173);line(537,1,537,173);
       for (i=0;i<=9;++i){
          gprintf(38,i*16+12,"%c",07);gprintf(574,i*16+12,"%c",07);
          gprintf(4,i*16+12,"%3d",i);
       }
    }
    /*--------creat box menu-----------*/
     setcolor(BROWN);bar(85,12,150,20);getimage(85,12,150,20,box);
    /*--------creat horline------------*/
     setcolor(BLACK);
     line(87,16,149,16);getimage(85,12,150,20,horline);
    /*--------creat delete cursor------*/
     setcolor(WHITE);line(86,16,149,16);setcolor(BLACK);
     gprintf(91,13,"DELETE%c",16);getimage(85,12,150,20,delete);
     putimage(85,12,delete,XOR_PUT);
    /*-------------------------*/
     exit_place=YES;buffer=box;setvisualpage(1);setactivepage(1);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
          datarows=rows;
        } break;
      case 0x4b00/*LEFT_ARROW*/
       : col=(col>0)?--col:0;break;
      case 0x4d00/*RIGHT_ARROW*/
       : col=(col<MAXCOL)?++col:MAXCOL;break;
      case 0x4900/*PgUp*/
       : col=(col<MAXCOL)?++col:MAXCOL;
      case 0x4800/*UP_ARROW*/
       : if((row==0)&&(rows>0)){
            setactivepage(page);
            if(driver!=CGA)
      scroll(UP,rows-1,position);
            else scroll(UP,rows,position);
            setvisualpage(page);page=(page==1)?0:1;
            row=0;gprintf(4,12,"%3d",rows-1);
                                  }
        row=(row>0)?--row:0;rows=(rows>0)?--rows:0;
                                  break;
    case 0x011b/*ESC*/
     : if(exit_place==NO){
     box=buffer;exit_place=YES;
       }break;
    case 0x326d/*M*/  :
    case 0x324d/*m*/  : if(exit_place==YES)
           if((key=mainselect(230,50,NO,position,&datarows,
     &rows,&row))==Q){
     free(box);free(inst);
     free(oldimage);free(horline);free(delete);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    return(-1);
    }
    break;
   case 0x264c/*L*/ :
   case 0x266c/*l*/ : if(box!=inst)group(ON);break;
   case 0x4700/*HOME*/
    : if (box!=delete){
      values(&row,&col,position,buffer,&rows);
      }break;
  case 0x3042/*B*/ :
  case 0x3062/*b*/ : if(box!=inst)setbaudrate(250,30);break;
  case 0x5200/*INS*/
   : box=buffer;
     putimage(62*col+40,16*row+12,box,XOR_PUT);
     ins=instruction(&GROUP);
     getimage(8,9+11*ins+36,73,9+11*ins+44,inst);
     putimage(8,9+11*ins+36,box,XOR_PUT);
     buffer=box;box=inst;activeviewport(MIDWINDOW);
     putimage(62*col+40,16*row+12,buffer,XOR_PUT);
     exit_place=NO;break;
 case  0x4a2d/*LINE*/
    : /* horizontal line */
     box=buffer;
     putimage(62*col+40,16*row+12,box,XOR_PUT);
     buffer=box;box=horline;
     putimage(62*col+40,16*row+12,buffer,XOR_PUT);
     exit_place=NO;break;
 case 0x5300/*DEL*/
    : /* delete instruction */
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
      box=buffer;
      putimage(62*col+40,16*row+12,box,XOR_PUT);
      buffer=box;box=delete;
      putimage(62*col+40,16*row+12,buffer,XOR_PUT);
      exit_place=NO;break;
   case 0x4e2b/*PLUS*/
      : if(box==delete){
         detectgraph(&g_driver,&g_mode);
         if((g_driver==EGA)||(g_driver==VGA))
    setcolor(BLACKGROUND);
         else setcolor(BLACK);
      }
      if(col!=MAXCOL){
        verline(62*col+40,16*row+12);
        position[rows].col[col]=(box==delete)?-1:VERLINE;
      }
     break;
  case 0x1c0d/*RETURN*/
        : if((exit_place==NO)&&(box!=delete)){
         putimage(62*col+40,16*row+12,box,COPY_PUT);
         putimage(62*col+40,16*row+12,buffer,XOR_PUT);
         if(box==inst)
   position[rows].data[col]=((GROUP+1)*10)+ins;
         if(box==horline)
  position[rows].data[col]=HORLINE;
      }
                     if (box == delete){
     position[rows].data[col] = -1;
     putimage(62*col+40,16*row+12,box,XOR_PUT);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
      putimage(62*col+40,16*row+12,buffer,COPY_PUT);
      putimage(62*col+40,16*row+12,buffer,XOR_PUT);
    }
    setcolor(GREEN);
                    if (col == 0){
line(41,16*row+12,41,16*row+20);
gprintf(38,16*row+12,"%c",07);
    }
                    if (col == MAXCOL){
line(537,16*row+12,537,16*row+20);
gprintf(534,16*row+12,"%c",07);
                    }
                    if (col != 0){
      if (position[rows].col[col-1] != -1)
  verline(62*(col-1)+40,16*row+12);
      if (position[rows+1].col[col-1] != -1)
  verline(62*(col-1)+40,16*(row+1)+12);
                    }
     if ((position[rows].col[col] != -1)&&(col != MAXCOL))
  verline(62*col+40,16*row+12);
     if ((position[rows+1].col[col] != -1)&&(col!=MAXCOL))
  verline(62*(col)+40,16*(row+1)+12);
     break;
      case 0x1709/*CTRL_I*/:   setactivepage(page);
      for(i=MAXROWS-1;i>=rows+1;i--)
position[i]=position[i-1];
      for (j=0;j<=7;++j){
position[rows].data[j] = -1;
position[rows].value[j] = -1;
```

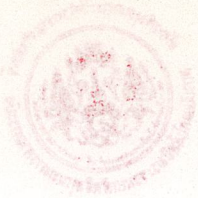
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
      position[rows].value1[j] = -1;
      }
           for (j=0;j<=6;++j) position[rows].col[j]  = -1;
           downform(rows-row);
           for(i=0;i<=9;i++)
       oldlineUPDOWN(OFF,position,rows-row+i,i);
           setvisualpage(page);page=(page==1)?0:1;
           break;
           case 0x1519/*CTRL_Y*/:       setactivepage(page);
           for(i=rows;i<=MAXROWS-1;i++)
      position[i]=position[i+1];
           downform(rows-row); /*parameter=start line */
           for(i=0;i<=9;i++)
       oldlineUPDOWN(OFF,position,rows-row+i,i);
           setvisualpage(page);page=(page==1)?0:1;
           break;
           case 0x4f00/*END*/   :if (box!=delete){
           under(&row,&col,position,buffer,&rows);
         }break;;
        }
      }while(1);
   }
  /*-------------------------------------------------------------*/
      main(int argc,char *argv[])
      {
       int driver,mode;int sX=0;int sY=0;
   check=YES;
       detectgraph(&driver,&mode);
  /*     if(registerbgidriver(CGA_driver)<0)exit(1);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    if(registerbgifont(triplex_font)<0)exit(1);*/
    switch(driver){
  case HERCMONO: graphmonochom();break;
  default      : graph640x200();break;
    }
   if(driver==HERCMONO){sX=40;sY=74;}
   /* draw box screen 2 page */
   setactivepage(0);setvisualpage(0);
   setbkcolor(BLACKGROUND); setcolor(WHITE);
   rectangle(1+sX,1+sY,639+sX,199+sY);setactivepage(1);
   rectangle(1+sX,1+sY,639+sX,199+sY);setactivepage(0);
   setviewport(2+sX,2+sY,638+sX,198+sY,1);
   /* display PLC */
   title();
   /* creat edit form page 1*/
   if(driver!=CGA){
 setactivepage(1);setcolor(YELLOW);
 setbkcolor(BLACKGROUND);start();
   }
   getch();setactivepage(0);activeviewport(MIDWINDOW);clearviewport();
   setvisualpage(1);setactivepage(0);
   start();group(-1);activeviewport(MIDWINDOW);
   /* 1200 baud ,8 bits,1 stop bits,even parity */
   SETTING=(0x80|0x03|0x00|0x18);
   setvisualpage(0);
   editorwork();free(ladder);closegraph();
   }
/*-------------------------------------------------------------*/
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



กิตติกรรมประกาศ

ขอขอบพระคุณ อาจารย์ภากร หุตะสังกาศ อาจารที่ปรึกษาที่ให้คำปรึกษาและอำนวยความสะดวกอย่างดียิ่ง และขอขอบพระคุณ อาจารย์สุพรรณ กุลพาณิชย์ เป็นอย่างสูงที่ได้ประสาทวิชาความรู้ทางด้านการทำงานและการใช้งาน PLC ซึ่งเป็นประโยชน์อย่างยิ่งในการทำปริญญานิพนธ์จนสำเร็จลุล่วงไปด้วยดี

ขอขอบคุณพี่ๆ เพื่อนๆและผู้ที่ไม่สามารถกล่าวถึงได้ในที่นี้ทุกๆคนที่ให้ความช่วยเหลือทางด้านเครื่องมือ ให้คำปรึกษาต่อคณะผู้จัดทำด้วยดีตลอดมา

สุดท้ายนี้ขอขอบพระคุณอย่างสูงต่อ บิดา มารดา ที่ให้การสนับสนุนและอุปการะทุกๆด้านตลอดมาจนสำเร็จการศึกษาจนถึงปัจจุบัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้