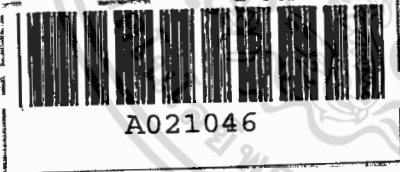


ปริญญานิพนธ์

ชุดสาธิต PLC

PROGRAMMABLE LOGIC CONTROLLER DEMOSTATE

นายเนรมิตร สิตตะวิบูล
นายพรชัย ศิลปกิจโกศล
นายอดิศักดิ์ โพรพิลากร
นายอัฐพร เพชรสุวรรณ



เลขหมู่.....
เลขทะเบียน.....1278.....
วัน เดือน ปี.....-3 พ.ศ.2538.....

021046

ปริญญานิพนธ์ฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรครุศาสตรบัณฑิต

สาขาวิชาอิเล็กทรอนิกส์และคอมพิวเตอร์
ภาควิชา ครุศาสตร์วิศวกรรม
คณะ ครุศาสตร์อุตสาหกรรม

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2537

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาควิชา วิศวกรรมศาสตร์วิศวกรรม
คณะ วิศวกรรมศาสตร์อุตสาหกรรม

วิทยาลัยเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ใบรับรองปริญญาานิพนธ์

ชื่อหัวข้อปริญญาานิพนธ์ ชุคสาธิต PLC

PROGRAMMABLE LOGIC CONTROLLER DEMOSTATE

ชื่อผู้ศึกษา

1. นายเนรมิตร สิตตะวิบูล
2. นายพรชัย ศิลปกิจโกศล
3. นายอดิศักดิ์ โพธิ์พลากร
4. นายอัฐพร เพชรสุวรรณ

อาจารย์ผู้ควบคุมปริญญาานิพนธ์

1. ผศ.ประเชิญ ไทรแจ่มจันทร์
2. อาจารย์กิติพงษ์ มะโน
3. อาจารย์วรวิทย์ สมหา

คณะกรรมการสอบปริญญาานิพนธ์	ลายมือ
อาจารย์วิสุทธิ์ อธิพรธรรม	
อาจารย์พีระวุฒิ สุวรรณจันทร์	
อาจารย์สมชาย หมั่นสายญาติ	
อาจารย์โกศล ตราชู	
อาจารย์ปิยะ จิตธรรมมาภิรมย์	

วันเดือนปี ที่สอบ วันที่ 24 เดือนธันวาคม พ.ศ.2537 เวลา 10.00 น.ถึง 11.30 น.

สถานที่สอบ ห้อง ค.301 คณะวิศวกรรมศาสตร์อุตสาหกรรม



ลงนาม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษา (ผศ.ดร.ธีรพล มาพิพัฒน์ อ.อมรยา) ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของลิขสิทธิ์ทุกครั้งที่มีการนำไปใช้

ปริญญานิพนธ์

เรื่อง ชุดสาธิต PLC

PROGRAMMABLE LOGIC CONTROLLER DEMOSTATE

ผู้จัดทำ

นายเนรมิตร สิตตะวินุต

นายพรชัย ศิลปกิจโกศล

นายอดิศักดิ์ โพธิ์พลากร

นายอัฐพร เพชรสุวรรณ

อาจารย์ที่ปรึกษา

ลงนาม

(ผศ.ประเชิญ ไทรแจ่มจันทร์)

ลงนาม

(อาจารย์กิติพงศ์ มะโน)

ลงนาม

(อาจารย์วรวิทย์ สมหา)

หัวหน้าภาควิชา

ลงนาม

(ผศ.ดร.ธีรพล เทพหัสดิน ณ อยุธยา)

ปริญญานิพนธ์

เรื่อง ชุดสาธิต PLC

PROGRAMMABLE LOGIC CONTROLLER DEMOSTATE

จุดประสงค์

1. เพื่อศึกษาการใช้งาน ไมโครคอนโทรลเลอร์
2. เพื่อศึกษาการใช้งาน PLC (PROGRAMMABLE LOGIC CONTROLLER)
3. เพื่อศึกษาการนำเอา ไมโครคอนโทรลเลอร์ มาออกแบบเป็น PLC
4. เพื่อศึกษาออกแบบ และ การเขียนโปรแกรมให้ ไมโครคอนโทรลเลอร์ สามารถติดต่อกับอุปกรณ์ อินพุต และ เอาท์พุต ได้
5. เพื่อสร้างชุดสาธิต PLC โดยใช้ ไมโครคอนโทรลเลอร์

ประโยชน์ที่คาดว่าจะได้รับ

1. ได้ความรู้เกี่ยวกับการนำเอา ไมโครคอนโทรลเลอร์ มาออกแบบ PLC
2. ได้ความรู้และเข้าใจการเขียน แลคเคอร์ ความคุมการทำงานของ PLC
3. ได้ชุดสาธิต PLC มาไว้ศึกษา และ ใช้งาน

ชุดสาธิต PLC

นายเนรมิตร สิตตะวิบูล
นายพรชัย ศิลปกิจโกศล
นายอดิศักดิ์ โพธิ์พลากร
นายอัฐพร เพชรสุวรรณ

อาจารย์ที่ปรึกษา
ศส.ประเชิญ ไทรแจ่มจันทร์
อาจารย์กิตติพงศ์ มะโน
อาจารย์วรวิทย์ สมหา
ปีการศึกษา 2537

บทคัดย่อ

ปัจจุบันเครื่องควบคุมแบบ “ซีเควนซ์” ได้มีใช้กันอย่างแพร่หลายในโรงงานอุตสาหกรรม ทั้งในขนาดกลาง และ ขนาดใหญ่แทน ระบบการควบคุมแบบเก่า โดยลักษณะการควบคุมในปัจจุบันจะเน้นเรื่องการใช้งานให้ง่าย แต่มีประสิทธิภาพในการทำงานสูงทำงานที่ซับซ้อนได้ และสามารถโปรแกรมการทำงานได้

โครงการ นี้เป็นการสร้างขึ้นเพื่อเป็นชุดสาธิตการทำงานของ PLC ซึ่งชุดสาธิตจะมีการทำงานเหมือนกับ PLC ที่ใช้ในโรงงานอุตสาหกรรม และ ใช้ในการศึกษาขั้นเบื้องต้น เครื่องควบคุมที่ออกแบบใช้ไมโครโปรเซสเซอร์ ขนาด 8 บิต เบอร์ Z80180 เป็นหน่วยประมวลผลกลาง และมีหน่วยป้อนโปรแกรม สามารถต่อออกไปใช้งาน กับอุปกรณ์ภายนอกได้ เปลี่ยนแปลงลักษณะ และ เงื่อนไขการควบคุมได้ง่าย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

PROGRAMMABLE LOGIC CONTROLLER DEMOSTATE

MR.NERAMIT	SITTAWIBUL
MR.PORNCHAI	SILLAPAKUKOSOL
MR.ADISAK	PHOTIPALAKORN
MR.ATTAPORN	PHETSUWAN

ADVISOR

ASIS.PROF.PRACHERN	TRAICHAMCHAN
MR.KITIPONG	MANO
MR.WORAWTT	SOMHA

1994

ABSTRACT

Recently programmable sequence control are many used in factory both medium and large side substitute old control system.

By now the feature of programmable logic control are emphasis on simple for used but high performance, working difficult and can programmable.

This thesis was create a set of demonstation of programmable logic controller (PLC). which set of demonstation similar work with PLC that used in factory and begin educate. PLC was design by used 8 bit microprocessor Z80180 to be central processing unite and there are input unite and can be connect to external device and simple to change the condition of control.

กิตติกรรมประกาศ

ปริญญานิพนธ์ ฉบับนี้สามารถสำเร็จลุล่วงไปได้ด้วยดี ก็เป็นเพราะความกรุณาของท่าน อาจารย์กิติพงศ์ มะโน ,อาจารย์ วรวิทย์ สมหา ,อาจารย์ สันติ คันทรระกุล ,อาจารย์โกศล ทรายู และ อาจารย์ภาควิชาครุศาสตร์วิศวกรรมทุกท่านที่กรุณาให้คำปรึกษา และ ช่วยเหลือทั้งทางด้านเทคนิค ตลอดจนเครื่องมือ และ อุปกรณ์ต่าง ๆทางผู้จัดทำขอขอบพระคุณเป็นอย่างสูง

นอกจากนี้ ยังมีเพื่อน ๆที่ให้คำปรึกษา ช่วยเหลือ และให้กำลังใจ ตลอดจนช่วยกันแก้ปัญหาต่าง ๆจนทำให้ปริญญานิพนธ์สำเร็จลงได้ ผู้จัดทำขอขอบคุณในน้ำใจของเพื่อน ๆทุกคน



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

เรื่อง	หน้า
บทที่ 1 บทนำ	1
บทที่ 2 ทฤษฎีการทำงานของ PLC	3
2.1 การทำงานของระบบ PLC	3
2.2 องค์ประกอบใน PLC	5
2.2.1 หน่วยประมวลผลกลาง	5
2.2.2 หน่วยอินพุต และ เอาท์พุท	7
2.2.3 หน่วยป้อนข้อมูล	11
2.3 คำสั่งที่ใช้เขียนโปรแกรม PLC	11
2.3.1 ภาษาแลดเดอร์	12
2.3.2 ภาษาบูลีน	13
2.3.3 ภาษาบล็อก	13
2.3.4 คำสั่งข้อความภาษาอังกฤษ	14
บทที่ 3 การออกแบบ PLC	15
3.1 คุณสมบัติของเครื่อง PLC	15
3.2 ส่วนต่าง ๆ ของการออกแบบ	15
3.2.1 หน่วยประมวลผล	15
3.2.2 หน่วยอินพุต และ เอาท์พุท	24
3.2.3 หน่วยป้อนข้อมูล และ แสดงผล	31
3.3 ตัวอย่างการเขียนโปรแกรม PLC	32
3.3.1 คำสั่ง LD,LD-NOT,OUT	32
3.3.2 คำสั่ง AND	33
3.3.3 คำสั่ง OR,OR-NOT	34
3.3.4 คำสั่ง ANB,ORB	35
3.3.5 คำสั่ง SET,RES	36
3.3.6 คำสั่ง TIM,K	37

เอกสารนี้ 3.3.7 คำสั่ง CNT ไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เรื่อง	หน้า
3.4 เทคนิคการเขียนโปรแกรม	39
3.4.1 การยวบวงจรให้ง่าย(1)	39
3.4.2 การยวบวงจรให้ง่าย(2)	40
3.4.3 เทคนิคการเขียนภาษามูลดิน	40
3.4.4 เทคนิคการเขียนโปรแกรม	41
3.5 วงจรที่ควรหลีกเลี่ยงในการเขียนโปรแกรม PLC	41
3.5.1 วงจรที่เป็นสะพานเชื่อม	41
3.5.2 วงจรเอาต์พุตไม่อยู่ทางขวามือสุด	42
3.5.3 วงจรที่อุปกรณ์เอาต์พุตใช้มากกว่า 1 แห่งในโปรแกรม	42
3.5.4 การควบคุมอุปกรณ์เอาต์พุตที่เป็นเงื่อนไขมากกว่า 1 เงื่อนไขใน 1 รังค์	43
บทที่ 4 การทดลองและผลการทดลอง	44
4.1 การทดลองการทำงานของโปรแกรมสแกนคีย์และการแสดงผล	44
4.1.1 การทดลองการใช้คีย์ที่ใช้ในการเขียนโปรแกรมและการแสดงผล	44
4.1.2 การกดปุ่ม FUNCTION	45
4.2 การทดลองการทำงานของโปรแกรมย่อย	48
4.2.1 การทดลองโปรแกรมย่อย LD	48
4.2.2 การทดลองโปรแกรมย่อย LD-NOT	49
4.2.3 การทดลองโปรแกรมย่อย AND,AND-NOT	50
4.2.4 การทดลองโปรแกรมย่อย OR,OR-NOT	51
4.2.5 การทดลองโปรแกรมย่อย ANB,ORB	52
4.2.6 การทดลองโปรแกรมย่อย SET,RES	53
4.2.7 การทดลองโปรแกรมย่อย TIMER	53
4.2.8 การทดลองโปรแกรมย่อย COUNTER	54
บทที่ 5 บทวิจารณ์สรุปและแนวทางการพัฒนา	55
5.1 บทสรุป	55
5.2 ปัญหาและอุปสรรค	55
5.3 แนวทางการพัฒนา	55

เรื่อง	หน้า
ภาคผนวก ก. ไฟล์ชาร์ต	56
ภาคผนวก ข. วงจรที่ใช้ใน PLC	72
ภาคผนวก ค. โปรแกรมของ PLC	79
ภาคผนวก ง. ใบงาน	186
บรรณานุกรม	



สารบัญตาราง

ตาราง	หน้า
ตารางที่ 2.1 การแบ่งหน่วยความจำของ PLC	6
ตารางที่ 2.2 สัญญาที่ใช้ใน PLC	8
ตารางที่ 3.1 การทำงานของ ST	19
ตารางที่ 3.2 อุปกรณ์และหมายเลขอุปกรณ์	29
ตารางที่ 3.3 คำสั่งภาษาบูลีนและภาษาแลดเดอร์	30



รูปที่ 3.15 การขুবวงจรให้ง่าย (1)	39
รูปที่ 3.16 การขুবวงจรให้ง่าย (2)	40
รูปที่ 3.17 เทคนิคการเขียนภาษานูลีน	40
รูปที่ 3.18 เทคนิคการเขียนโปรแกรม	41
รูปที่ 3.19 วงจรที่เป็นสะพานเชื่อม	42
รูปที่ 3.20 การแปลงวงจรที่เป็นสะพานเชื่อม	42
รูปที่ 3.21 วงจรที่เอาท์พุทไม่อยู่ทางขวามือ	42
รูปที่ 3.22 การแปลงวงจรที่เอาท์พุทไม่อยู่ทางขวามือ	42
รูปที่ 3.23 วงจรที่เอาท์พุทใช้มากกว่า 1 ริงก์ในโปรแกรม	43
รูปที่ 3.24 การแปลงวงจรที่เอาท์พุทใช้มากกว่า 1 ริงก์ในโปรแกรม	43
รูปที่ 3.25 วงจรควบคุมเอาท์พุทที่เป็นเงื่อนไขมากกว่า 1 เงื่อนไข	43
รูปที่ 3.26 การแปลงวงจรควบคุมเอาท์พุทที่เป็นเงื่อนไขมากกว่า 1 เงื่อนไข	43
รูปที่ 4.1 หน้าจอ LCD เมื่อเริ่มจ่ายไฟ	44
รูปที่ 4.2 หน้าจอ LCD เมื่อกดปุ่ม ADD	44
รูปที่ 4.3 หน้าจอ LCD เมื่อสิ้นสุดการเขียนโปรแกรม	45
รูปที่ 4.4 แสดงการกดปุ่ม ADD	46
รูปที่ 4.5 แสดงการกดปุ่ม \downarrow	46
รูปที่ 4.6 แสดงการกดปุ่ม INS	46
รูปที่ 4.7 แสดงการกดปุ่ม DEL	47
รูปที่ 4.8 แสดงการกดปุ่ม \downarrow 1 ครั้ง	47
รูปที่ 4.9 แสดงการกดปุ่ม \uparrow 1 ครั้ง	47
รูปที่ 4.10 โปรแกรมทดลองโปรแกรมย่อย LD	48
รูปที่ 4.11 โปรแกรมทดลองโปรแกรมย่อย LD-NOT	50
รูปที่ 4.12 โปรแกรมทดลองโปรแกรมย่อย AND,AND-NOT	50
รูปที่ 4.13 โปรแกรมทดลองโปรแกรมย่อย OR,OR-NOT	51
รูปที่ 4.14 โปรแกรมทดลองโปรแกรมย่อย ANB,ORB	52
รูปที่ 4.15 โปรแกรมทดลองโปรแกรมย่อย AET,RES	53
รูปที่ 4.16 โปรแกรมทดลองโปรแกรมย่อย TIMER	54
รูปที่ 4.17 โปรแกรมทดลองโปรแกรมย่อย COUNTER	54

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1

บทนำ

ในปัจจุบันโรงงานอุตสาหกรรมในประเทศไทยได้เริ่มใช้ PLC (PROGRAMMABLE LOGIC CONTROLLER) หรือ PC (PROGRAMMABLE CONTROLLER) แทนวงจร รีเลย์ ที่ใช้ในการควบคุมแบบอันดับเพิ่มขึ้น ซึ่งระบบเก่าที่ใช้รีเลย์นั้นติดตั้งคัดแปลงแก้ไขลำบากมาเป็นระบบการควบคุมแบบใหม่ที่ใช้วงจร อิเล็กทรอนิกส์แทนรีเลย์ และ ใช้การเขียนโปรแกรมทำนองเดียวกับการเขียนโปรแกรมคอมพิวเตอร์ กำหนดเงื่อนไข การควบคุมแทน การเดินสายไฟเชื่อมต่อดวงจรไฟฟ้าแบบเก่าเพื่อความสะดวกระบบรีเลย์เหมาะสมกับการควบคุมขนาดเล็กและไม่มี การขยายในอนาคต เนื่อง จากวงจรรีเลย์ จะมีราคาต่ำกว่า PC แต่สำหรับงานควบคุมขนาดใหญ่ ที่ซับซ้อนควรใช้PCเพราะมีประสิทธิภาพสูงความก้าวหน้าทางเทคโนโลยีทางด้าน ไมโคร-โปรเซสเซอร์ ทำให้โฉมหน้า PC เปลี่ยนแปลงไป ทั้งทางด้านฮาร์ดแวร์ และ ซอฟต์แวร์ ทางฮาร์ดแวร์เช่นทำให้ PC มีความเร็วสูงขึ้นโดยใช้บิตสไลซ์ไมโครโปรเซสเซอร์ (BIT SLICE MICROPROCESSOR) และมี PC ขนาดเล็กราคาถูกหรือ PLC ที่ใช้งานแทนวงจรรีเลย์ ในการควบคุมแบบ ON/OFF โดยเฉพาะได้ถูกสร้างขึ้นทางด้านซอฟต์แวร์ เช่น คำสั่งบล็อก (BLOCK INSTRUCTION) จะถูกใช้ร่วมกับ คำสั่งแลดเดอร์ (LADDER DIAGRAM INSTRUCTION) เพื่อความสะดวกในการเขียนโปรแกรม และ การเคลื่อน ย้ายแปลงค่าข้อมูลของ PLC ทำพร้อมกันทั้งหมดในเวลาเดียวกันทำให้การทำงานของ PLC เร็วขึ้น

ทั้งทางด้าน ฮาร์ดแวร์ และ ซอฟต์แวร์ของ PLC จะแบ่งเป็น ส่วนประกอบย่อย ๆ เรียกว่า โมดูล (MODULE) ทำงานร่วมกัน แต่ละโมดูลมีหน้าที่ของตนเอง แต่สามารถสับเปลี่ยน โมดูลที่มีหน้าที่เดียวกันแทนกันได้ เพื่อให้เหมาะสมกับ ลักษณะงานที่ต้องการทำให้การเปลี่ยนแปลงแก้ไขหรือขยายขอบเขตการใช้งาน PLC ทำได้ง่ายทั้งในแง่ของฮาร์ดแวร์ และ ซอฟต์แวร์ เช่นการเปลี่ยนแปลงขนาด และ ชนิดของ อินพุท/ เอาท์พุท และ หน่วยความจำ

PLC เป็นคอมพิวเตอร์เฉพาะงานประเภทหนึ่งซึ่งลักษณะ โครงสร้างของ PLC เหมือนกับคอมพิวเตอร์ทั่ว ๆ ไปแต่ PLC ถูกออกแบบขึ้น เพื่อใช้ในการควบคุมโดยเฉพาะความแตกต่างระหว่าง PLC กับคอมพิวเตอร์ ทั่วๆ ไปคือ

1. PLCถูกออกแบบและสร้างขึ้นให้ทนต่อสภาพแวดล้อมในโรงงานอุตสาหกรรมโดยเฉพาะ เช่น อุณหภูมิที่สูง และ ต่ำมาก ๆ ความชื้นสูง ระบบไฟฟ้าที่มีการรบกวนไม่สม่ำเสมอการสั่นสะเทือน และ การกระแทกอย่างรุนแรงบ่อยครั้ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้ภายในเพื่อการศึกษาค้นคว้าเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. การโปรแกรม และ การใช้งาน PLC ทำได้ง่ายไม่ยุ่งยาก เหมือนคอมพิวเตอร์ทั่วไป PLC มีระบบ การตรวจสอบตัวเอง ตั้งแต่ช่วงติดตั้งจนถึงช่วงใช้งาน ทำให้การบำรุงรักษาทำได้ง่าย

3. PLC ถูกพัฒนาให้มีความตัดสินใจสูงขึ้นเรื่อย ๆ ทำให้การใช้งานสะดวกขึ้นขณะที่วิธีใช้คอมพิวเตอร์ยุ่งยาก และ ซับซ้อนขึ้นเรื่อย ๆ เพื่อให้ผู้ใช้ สามารถปฏิบัติตามโปรแกรมของผู้ใช้ได้พร้อมกันหลายโปรแกรม

PLC ทำให้การควบคุมมีความคล่องตัวสูงขึ้นระบบควบคุมที่ใช้ PLC ทำงานด้วยโปรแกรม ภายในหน่วยความจำซึ่งแตกต่างจากระบบรีเลย์ที่ใช้การเดินสายทำให้ระบบควบคุมแบบ PLC เปลี่ยนแปลงแก้ไขลักษณะเงื่อนไขการควบคุมได้มีความคล่องตัวในการควบคุมสูงเพียงป้อนโปรแกรมใหม่ ให้กับหน่วยความจำ ซึ่งต่างจากระบบรีเลย์ ที่จะต้องเดินสายใหม่ทั้งหมดเมื่อต้องการที่จะเปลี่ยนแปลงลักษณะการควบคุม PLC มีระดับการตัดสินใจสูงนอกจากจะควบคุมอุปกรณ์ภายนอกให้ทำงาน ตามต้องการแล้ว ยังสามารถติดต่อกับอุปกรณ์อื่น ๆ เช่นสามารถติดต่อตอบโต้ และแสดงสภาพการทำงานให้ผู้ใช้ทราบทางจอภาพ และ รับเงื่อนไขการควบคุมจากคอมพิวเตอร์หลักได้

การติดตั้ง PLC ทำได้ง่าย เพื่อลดค่าใช้จ่ายในการติดตั้ง PLC จะใช้เนื้อที่ในการติดตั้งเพียงครึ่งหนึ่งของระบบรีเลย์การใช้ PLC ทดแทนระบบรีเลย์สามารถติดตั้ง PLC เข้ากับแผงควบคุมเดิม และ เดินสายเชื่อม ระหว่างหน่วยอินพุต/เอาต์พุตของ PLC กับจุดต่อภายในแผงควบคุมได้ โดยง่าย และ ป้อนโปรแกรมการควบคุม ให้หน่วยความจำ

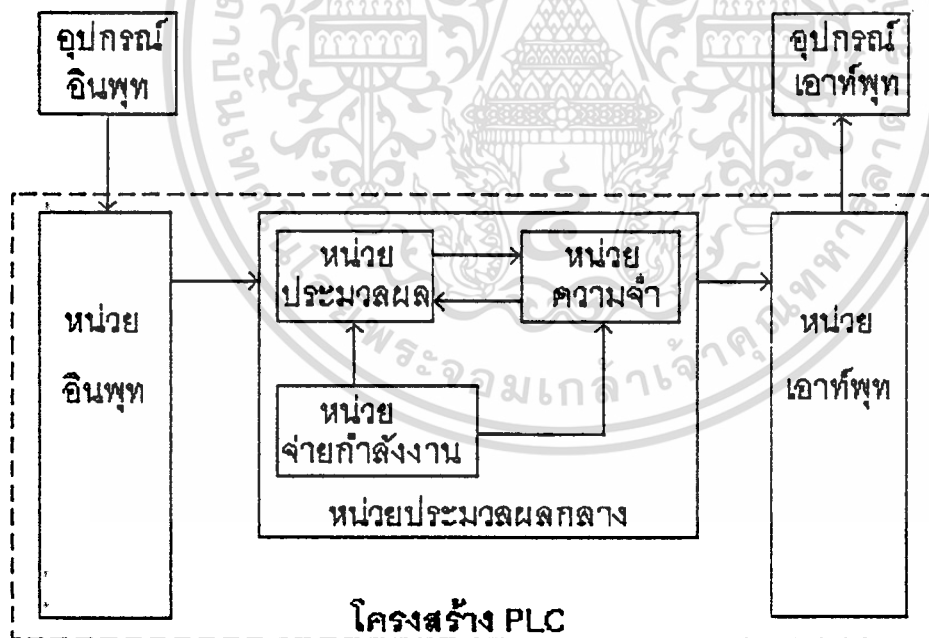
การบำรุงรักษา PLC ประกอบขึ้นด้วยวงจรอิเล็กทรอนิกส์ที่มีลักษณะเป็น โมดูล มีการตรวจสอบสภาพการทำงานของตัวเองและค้นหาจุดบกพร่องหรือจุดเสียได้ง่ายการ ซ่อมแซมเพียงแต่สับเปลี่ยนโมดูลที่เสียออกขณะเดียวกัน PLC สามารถ ตรวจสอบสถานะ ON หรือ OFF ของอุปกรณ์ภายนอกทุกขั้นตอนการทำงาน ของ PLC ทำให้การค้นหาสิ่งผิดปกติในระบบควบคุมง่าย PLC เป็นอุปกรณ์ควบคุม ที่สามารถใช้ในงานควบคุมได้ทุกประเภท และ สามารถจะติดต่อกับ PLC หรือระบบควบคุมอื่น ๆ พร้อมทั้งจัดทำรายงานแผนการผลิต มีระบบตรวจสอบความบกพร่องของตนเองทำให้ PLC ถูกใช้งานอย่าง กว้างขวาง ในโรงงานอุตสาหกรรมทุกสาขา เพื่อเพิ่มให้มีประสิทธิภาพการผลิตให้ดียิ่งขึ้น

ในอนาคต PLC จะมีประสิทธิภาพสูงขึ้น เนื่องจากการพัฒนาทั้ง ทางด้านฮาร์ดแวร์และซอฟต์แวร์ PLC ต่างระบบสามารถติดต่อ และ ใช้งานร่วมกันโดยใช้ระบบสื่อสารที่มีความเร็วสูง CPU และ หน่วยความจำจะมีความเร็วสูงขึ้น การโปรแกรมจะใช้ภาษาระดับสูงที่คล้ายภาษามนุษย์มากขึ้น

บทที่ 2

ทฤษฎีการทำงานของ PLC

PLC เป็นคอมพิวเตอร์ชนิดหนึ่ง ที่ถูกออกแบบมา เพื่อใช้ในงานควบคุมระบบต่าง ๆ ในโรงงาน อุตสาหกรรม ซึ่งต้องทนต่อ อุณหภูมิความชื้นฝุ่นละอองระบบไฟฟ้าที่มีการรบกวนไม่สม่ำเสมอ และ ต้องทนต่อแรงกระแทก และ แรงสั่นสะเทือนที่เกิดขึ้นต่าง ๆ ได้ โครงสร้าง พื้นฐาน ของ PLC ยัง คล้ายคลึงกับ ระบบคอมพิวเตอร์ทั่ว ๆ ไปคือ ประกอบด้วยหน่วยอินพุต (INPUT-INTERFACE) หน่วยประมวลผลกลาง (PROCESSOR) และ หน่วยเอาต์พุต (OUTPUT INTERFACE) ซึ่งในการ ติดตามควบคุม การทำงานของระบบนั้น PLC ยังต้องมีการติดต่อกับอุปกรณ์ภายนอกต่าง ๆ คือ อุปกรณ์ อินพุต และ อุปกรณ์เอาต์พุต ซึ่งถึงแม้จะไม่ใช้ส่วนประกอบของ PLC แต่ก็ถือว่าเป็นส่วนหนึ่งของ ระบบ PLC ดังแสดงในรูป



รูปที่ 2.1 แสดงระบบ PLC

2.1 การทำงานของระบบ PLC จะประกอบด้วยขั้นตอนต่างๆ ดังนี้

1. อ่านสถานะของอุปกรณ์อินพุตซึ่งอาจจะเป็นสวิตช์ต่างๆหรืออุปกรณ์ตรวจจับสัญญาณ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า (SENSOR) ที่วัดค่าสัญญาณอะนาล็อกเข้ามาทางหน่วยอินพุต

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. หน่วยอินพุทจะทำการแปลงสถานะของอุปกรณ์อินพุทให้อยู่ในรูปแบบที่หน่วยประมวลผลกลางสามารถเข้าใจได้แล้วส่งไปเก็บไว้ในหน่วยความจำของหน่วยประมวลผลกลาง

3. หน่วยประมวลผลนำโปรแกรมของผู้ใช้มาปฏิบัติตามทีละ 1 คำสั่งโดยเริ่มจาก คำสั่งแรกจนสิ้นสุดโปรแกรมในหน่วยความจำโดยหน่วยประมวลผล จะใช้สถานะต่าง ๆ ของอุปกรณ์อินพุทที่เก็บไว้ในหน่วยความจำมาทำการประมวลผลตามเงื่อนไขที่กำหนดไว้ในแต่ละคำสั่งถ้าผลการปฏิบัติตามคำสั่งใดทำให้สถานะของอุปกรณ์เอาต์พุท มีการเปลี่ยนแปลง ผลดังกล่าวจะถูกบันทึกไว้ในหน่วยความจำจากนั้นหน่วยประมวลผลจะนำคำสั่งต่อไปของผู้ใช้ขึ้นมาปฏิบัติตาม

4. หน่วยเอาต์พุทนำสถานะของเอาต์พุทจุดต่างๆในหน่วยความจำมาแปลงให้อยู่ในรูปแบบสัญญาณที่สามารถควบคุมอุปกรณ์เอาต์พุทได้ เพิ่มส่งต่อไปยังอุปกรณ์เอาต์พุท

5. อุปกรณ์เอาต์พุทซึ่งอาจเป็นหลอดไฟฟ้า กระดิ่งไฟฟ้า มอเตอร์ไฟฟ้า หรือวาล์วควบคุม จะทำการเปิดเปิดตามสัญญาณที่ส่งมาควบคุมโดยหน่วยเอาต์พุท

การทำงานของ 1 รอบของ PLC จากขั้นตอนที่ 1 ถึงขั้นตอนที่ 5 เราเรียกว่า การสแกน และ ช่วงเวลาที่ PLC ใช้ในการสแกน 1 รอบว่า ช่วงเวลาสแกน (SCAN TIME) ซึ่ง จะใช้เวลาประมาณ 1 ถึง 100 มิลลิวินาทีขึ้นอยู่กับ ขนาดความยาวของโปรแกรมผู้ใช้ประเภท และ จำนวนของอุปกรณ์อินพุท และ อุปกรณ์เอาต์พุทรวมทั้งคุณลักษณะของ PLC เอง

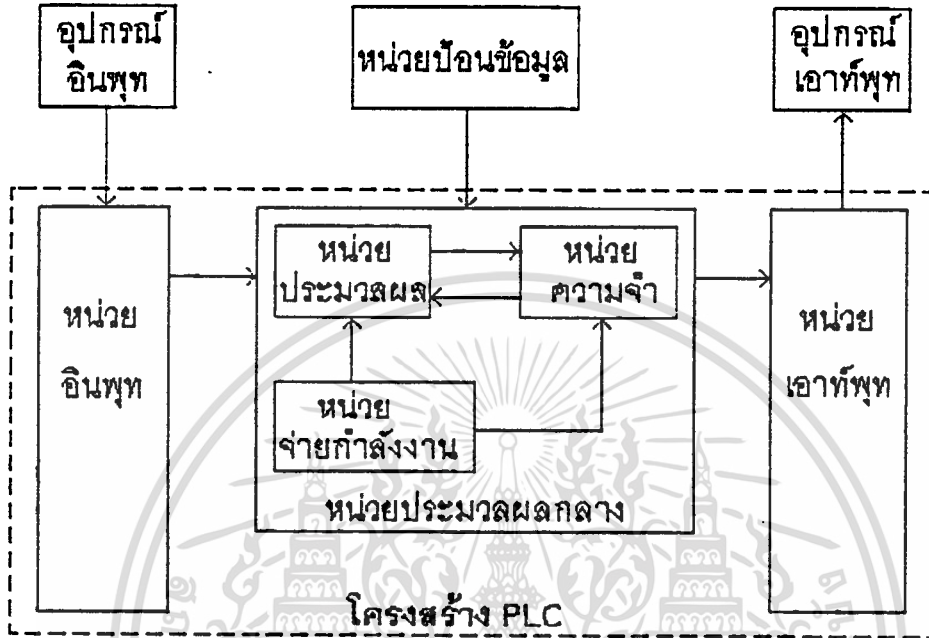


รูปที่ 2.2 การสแกนของ PLC

ช่วงเวลาสแกนของ PLC จะกำหนดขีดความสามารถของ PLC ในการตรวจจับการเปลี่ยนแปลงของอุปกรณ์ภายนอก และ การควบคุมเครื่องจักร เช่น ที่มีช่วงเวลาสแกน 10 มิลลิวินาที ย่อมไม่สามารถ รับสถานะแท้จริงของอุปกรณ์ ที่มีการเปลี่ยนแปลงทุก 8 มิลลิวินาที ได้ ถ้าใช้ PLC ดังกล่าว การควบคุมจะผิดพลาดหมด

อุปกรณ์อีกชนิดหนึ่งในระบบ PLC คือหน่วยป้อนข้อมูล (PROGRAMMING DEVICE) ซึ่งใช้สำหรับป้อนโปรแกรมผู้ใช้ เข้าไปยังหน่วยความจำของ PLC โดยโปรแกรมผู้ใช้จะเป็นตัว

กำหนดเงื่อนไขที่หน่วยประมวลผลนำไปใช้ในการตัดสินใจว่าจะส่งสัญญาณไปควบคุมอุปกรณ์เอาต์พุตในลักษณะใดเมื่อสถานะอุปกรณ์อินพุตเปลี่ยนแปลง



รูปที่ 2.3 หน่วยบ่อนข้อมูล

2.2 องค์ประกอบในระบบ PLC

2.2.1. หน่วยประมวลผลกลาง

มีหน้าที่ควบคุมการทำงานทั้งหมดของเครื่อง PLC ประกอบด้วยหน่วยประมวลผล หน่วยความจำ และหน่วยจ่ายกำลังงาน โดยในแต่ละส่วนมีหน้าที่ดังนี้

หน่วยประมวลผล มีหน้าที่นำโปรแกรมผู้ใช้ (USER PROGRAM) มาปฏิบัติเพื่อควบคุมอุปกรณ์ภายนอกตามเงื่อนไขการควบคุมตามที่ผู้เขียนโปรแกรมต้องการควบคุมการติดต่อ รับ-ส่งข้อมูลระหว่างหน่วยประมวลผลกลาง กับหน่วยอินพุต/เอาต์พุต และ หน่วยอินพุต/เอาต์พุต กับอุปกรณ์ภายนอก ติดต่อกับผู้ใช้ และ อุปกรณ์ร่วมตรวจสอบสภาพ การทำงานของ PLC โดยมีโปรแกรมบริหารระบบ (OPERATING SYSTEM) เป็นผู้ควบคุมอีกที

โดยทั่วไป หน่วยประมวลผล จะเป็นไมโครโปรเซสเซอร์ ซึ่งนอกจากจะทำหน้าที่แทนวงจรรีเลย์ในการควบคุมแบบ ON/OFF เหมือนวงจรตรรกแล้วยังสามารถคำนวณทางคณิตศาสตร์ และ ติดต่อกับอุปกรณ์ภายนอกได้ซึ่งในที่นี้เราใช้ Z80180 เป็นไมโครโปรเซสเซอร์ โดยใช้ Z80180 คอลโทรลเลอร์ ซึ่งเป็น CPU ตระกูล Z80 เป็น CPU หลักประจำบอร์ดที่ใช้ประโยชน์ด้านการคำนวณ ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หน่วยความจำ มีหน้าที่เก็บโปรแกรม และ ข้อมูลต่าง ๆ ที่ PLC ใช้ในการประมวลผล PLC แบ่งหน่วยความจำออกเป็น 2 ส่วนใหญ่ ๆ คือหน่วยความจำระบบ และ หน่วยความจำผู้ใช้

โปรแกรมบริหารระบบ (OPERATING SYSTEM)	SYSTEM MEMORY
ข้อมูลจากการทำงานโปรแกรมระบบ (SCRATCH PAD)	
โปรแกรมผู้ใช้ (USER PROGRAM)	USER MEMORY
ข้อมูลจากการทำงานโปรแกรมผู้ใช้ (SCRATCH PAD)	
ตารางข้อมูล (DATA TABLE)	

ตารางที่ 2.1 การแบ่งหน่วยความจำของ PLC

หน่วยความจำระบบ (SYSTEM MEMORY) จะเป็นส่วนที่ถูกหนดมาจาก โรงงานผู้ผลิต ซึ่ง ไม่นอนุญาตให้ผู้ใช้เข้าไปเปลี่ยนแปลงแก้ไขหน่วยความจำในส่วนนี้จะเก็บโปรแกรมในส่วนของ การตรวจสอบสภาพการทำงาน ของ PLC การติดต่อกับ หน่วยอินพุต และ เอาท์พุต และ โปรแกรมที่ทำการแปลคำสั่งของโปรแกรมผู้ใช้ให้อยู่ในรูปแบบที่หน่วยประมวลผลเข้าใจ นอกจากนี้ยังมี ส่วนของหน่วยความจำที่ใช้ในการเก็บสถานะหรือ ผลการทำงาน ชั่วคราวที่เกิดจากการ ปฏิบัติการโปรแกรมบริหารระบบ

หน่วยความจำผู้ใช้ (USER MEMORY หรือ APPLICATION MEMORY)เป็นส่วนที่ใช้เก็บโปรแกรมผู้ใช้งานซึ่งเป็นตัวกำหนดเงื่อนไข การควบคุมในงานระบบ ใดระบบหนึ่ง รวมทั้ง ผลของการปฏิบัติการชั่วคราวของ โปรแกรมผู้ใช้งาน นอกจากนี้หน่วยความจำในส่วนนี้ ยังใช้เป็น ที่เก็บสถานะต่าง ๆ ของอุปกรณ์ภายนอก และ อุปกรณ์ภายในหรือที่เรียกว่า ตารางข้อมูลซึ่งเราสามารถแบ่งออกเป็น 4 ส่วนด้วยกันคือ

1. ตารางอินพุต (INPUT TABLE)
2. ตารางเอาท์พุต (OUTPUT TABLE)
3. ตารางอุปกรณ์ภายใน (INTERNAL DISCRETE TABLE)
4. ตารางรีจิสเตอร์ (REGISTER TABLE)

ตารางอินพุท มีหน้าที่ เก็บข้อมูลที่แสดงสถานะของอุปกรณ์อินพุทภายนอกโดยจะเก็บค่า "1" แทนสถานะการ ON และ เก็บค่า "0" แทนสถานะการ OFF ของอุปกรณ์ อินพุท ซึ่งข้อมูลของตารางอินพุทจะเปลี่ยนสถานะตาม อุปกรณ์ภายนอก ที่เชื่อมต่อขณะเริ่มต้นการสแกน ทุกครั้ง

ตารางเอาต์พุท มีหน้าที่ เก็บข้อมูลที่แสดงสถานะของอุปกรณ์เอาต์พุทภายนอก โดยจะเก็บค่า "1" แทนสถานะ ON และ จะเก็บค่า "0" แทนสถานะการ OFF ของอุปกรณ์เอาต์พุท โดย อุปกรณ์เอาต์พุทภายนอกที่เชื่อมต่อกับ อุปกรณ์เอาต์พุท จะเปลี่ยนสถานะตามตารางเอาต์พุทก่อนที่จะสิ้นสุดการสแกนทุกครั้ง

ตารางอุปกรณ์ภายใน มีหน้าที่ เก็บสถานะของอุปกรณ์ภายในที่ไม่มีจุดเชื่อมต่อกับอุปกรณ์ภายนอก เช่นสถานะของอุปกรณ์หน่วงเวลา อุปกรณ์นับจำนวนหรือ รีเลย์ภายในซึ่งใช้ในการเก็บสถานะการควบคุมชั่วคราว เช่นเดียวกับรีเลย์ควบคุมในวงจรรีเลย์

ตารางรีจิสเตอร์ มีหน้าที่ เก็บข้อมูลที่เป็นตัวเลข หรือ ข้อมูลอะนาลอก ซึ่งไม่สามารถใช้ตารางอินพุท/เอาต์พุทหรือตารางอุปกรณ์ภายในเก็บรักษาได้เพราะมีค่าเป็นตัวเลขไม่ใช่สถานะ ON/OFF ตารางรีจิสเตอร์ของ PLC ประกอบด้วยอินพุทรีจิสเตอร์เอาต์พุทรีจิสเตอร์และรีจิสเตอร์ภายใน

อินพุทรีจิสเตอร์จะเก็บข้อมูลจากอุปกรณ์อินพุทซึ่งจะเปลี่ยนค่าเมื่อเริ่มการสแกนทุกครั้ง เอาต์พุทรีจิสเตอร์ จะเก็บข้อมูลที่ควบคุม อุปกรณ์เอาต์พุท และ จะส่งค่าเมื่อสิ้นสุด การสแกนทุกครั้ง ส่วนรีจิสเตอร์ภายใน จะเก็บข้อมูลชั่วคราวที่ได้จากการประมวลผล และ การคำนวณทางคณิตศาสตร์ค่าคงที่ของการหน่วงเวลา และ จำนวนนับรวมทั้งค่าของเวลาปัจจุบัน และ จำนวนนับปัจจุบันของอุปกรณ์ภายใน

หน่วยจ่ายกำลังงาน ทำหน้าที่ จ่ายกระแสไฟฟ้าให้กับส่วนต่าง ๆ ของ PLC คือหน่วยประมวลผลกลาง หน่วยความจำ และ หน่วยอินพุท/เอาต์พุท โดยรักษาแรงดันไฟฟ้าให้คงที่ และ แจกจ่ายให้หน่วยประมวลผลทราบเมื่อระบบไฟฟ้ามีอาการผิดปกติ

2.2.2 หน่วยอินพุท/เอาต์พุท

ทำหน้าที่ติดต่อระหว่างPLCกับอุปกรณ์ภายนอกซึ่งมีทั้งหน่วยอินพุท/เอาต์พุท แบบตรรก (DIGITAL INPUT/OUTPUT) ซึ่งใช้ในการ ควบคุมแบบ ON/OFF และ หน่วยอินพุท/เอาต์พุท แบบตัวเลข (ANALOG INPUT/OUTPUT) ซึ่งได้ตรวจับระดับสัญญาณ ต่าง ๆ หรือตำแหน่งของเครื่องจักร ทำให้ขอบเขตของการ ใช้ PLC ในงาน ควบคุมกว้างขึ้น

สัญญาณ ไฟฟ้าที่อุปกรณ์อินพุท/เอาต์พุทแบบตรรกใช้มีหลายค่าทั้งแบบแรงดันไฟฟ้า กระแสตรง และ กระแสสลับซึ่งเราพอสรุป สัญญาณมาตรฐานที่หน่วยอินพุท เอาต์พุทใช้ในการติดต่อกับ PLC ได้ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อุปกรณ์อินพุท	อุปกรณ์เอาต์พุท
24 VAC/DC	12-48 VAC/DC
48 VAC/DC	120 VAC/DC
120 VAC/DC	220 VAC/DC
220 VAC/DC	หน้าสัมผัสรีเลย์
ระดับ TTL	ระดับ TTL

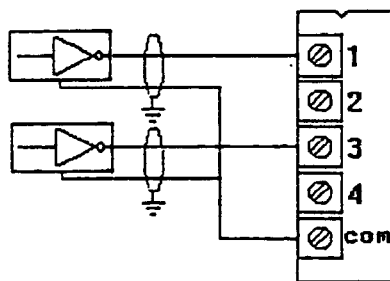
ตารางที่ 2.2 สัญญาณที่ใช้ใน PLC

หน่วยอินพุทแบบตรรกะ จะทำหน้าที่ รับค่าทางไฟฟ้า ของอุปกรณ์ภายนอก และ เปลี่ยนสถานะทางตรรกะ เพื่อเก็บไว้ในอินพุทของหน่วยความจำ โดยจะมีค่าเป็น "1" เมื่ออุปกรณ์อินพุทอยู่ในสถานะ ON หรือปิดวงจรไฟฟ้า และจะมีค่าเป็น "0" เมื่ออุปกรณ์ อินพุทอยู่ในสถานะ OFF หรือ เปิดวงจรไฟฟ้า

หน่วยเอาต์พุท จะรับสถานะตรรกะจากตารางเอาต์พุทในหน่วยความจำ และ เปลี่ยนเป็น ค่าทางไฟฟ้า เพื่อควบคุมอุปกรณ์ภายนอกอีกทีหนึ่ง ค่า "1" หมายถึงการ ON หรือ การต่อวงจรไฟฟ้า และ ค่าที่เป็น "0" หมายถึงการ OFF หรือตัดวงจรไฟฟ้า ซึ่งการจะเลือกใช้หน่วยอุปกรณ์อินพุท/เอาต์พุทชนิดที่ถูกต้อง และ เหมาะสมจำเป็นต้องเข้าใจคุณลักษณะ และ การทำงานของหน่วยอินพุท/เอาต์พุทแต่ละชนิดด้วย

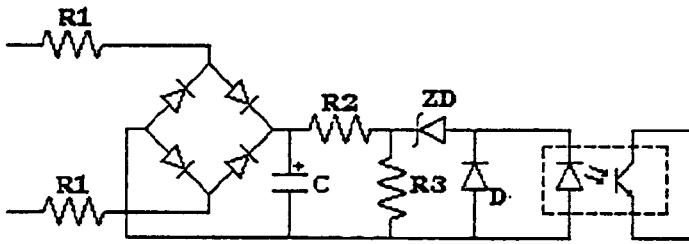
การเชื่อมต่อหน่วยอินพุทแบบ TTL

ทำหน้าที่ในการรับสัญญาณอินพุทจากอุปกรณ์อิเล็กทรอนิกส์ชนิด TTL (TRANSISTOR TRANSISTER LOGIC) หรือ อุปกรณ์ที่ใช้สัญญาณไฟฟ้า 5V DC ซึ่งมีวิธีการเชื่อมต่อดังรูป



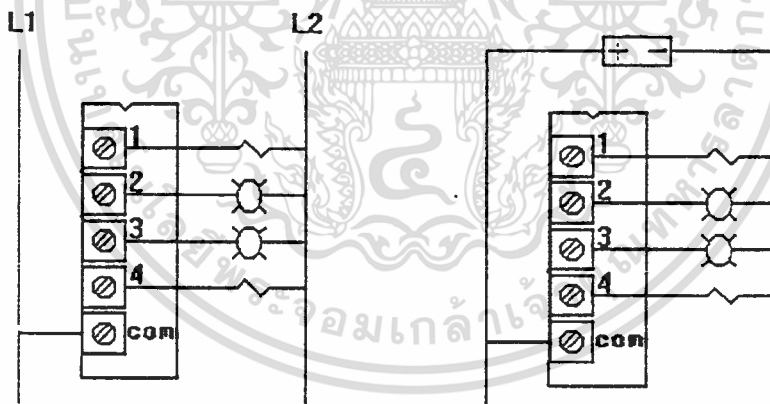
เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ 2.4 การเชื่อมต่อหน่วยอินพุทแบบ TTL ให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หน่วยอินพุตแบบ AC/DC



รูปที่ 2.5 วงจรหน่วยอินพุตแบบ AC/DC

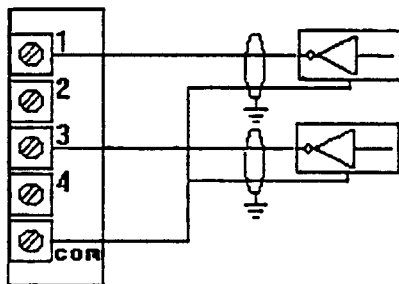
หน่วยอินพุตแบบ AC/DC ประกอบด้วย วงจรเรกติไฟร์ (RECTIFIER) วงจรกรองความถี่ (FILTER) วงจรตรวจจับ (DETECTOR) และวงจรตรรกะที่ทำหน้าที่ส่งข้อมูลให้กับหน่วยประมวลผลระหว่าง วงจรตรรกะกับวงจรส่วนอื่น จะใช้การเชื่อมต่อแบบออปติค (OPTICAL COUPLING) เพื่อป้องกันหน่วยประมวลผลกลาง จากการรบกวนของอุปกรณ์ภายนอก



รูปที่ 2.6 การเชื่อมต่อหน่วยอินพุตแบบ AC/DC

หน่วยเอาต์พุตแบบ TTL

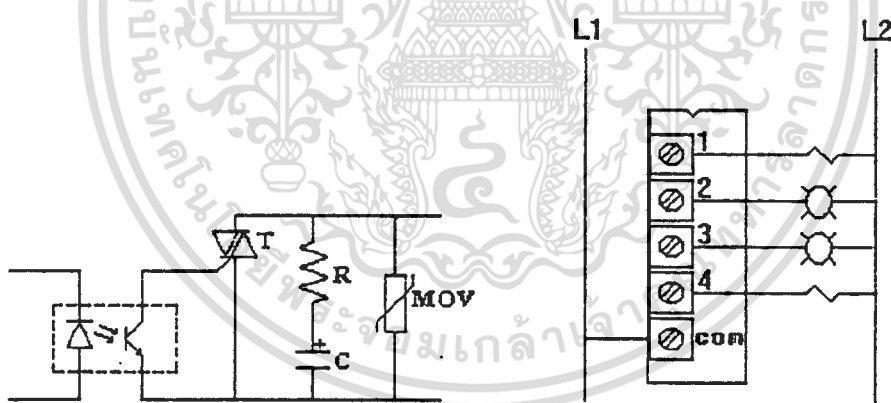
ทำหน้าที่ ควบคุมการทำงานของอุปกรณ์อิเล็กทรอนิกส์ที่เป็นแบบ TTL หรืออุปกรณ์ที่ใช้สัญญาณไฟ 5V DC ซึ่งมีวิธีการเชื่อมต่อดังรูป



รูปที่ 2.7 การเชื่อมต่อหน่วยเอาต์พุตแบบ TTL

หน่วยเอาต์พุตแบบ AC

หน่วยเอาต์พุตแบบ AC จะเปลี่ยนสถานะควบคุมจากหน่วยประมวลผลกลางให้เป็นระดับแรงดันไฟฟ้ากระแสสลับ เพื่อใช้ในการ ควบคุมการทำงานของอุปกรณ์เอาต์พุตโดยประกอบด้วย วงจรตรรกซึ่งติดต่อกับ หน่วยประมวลผลกลางวงจรเชื่อมต่อ แบบออปติค สวิตซ์อิเล็คทรอนิกส์ และ วงจรรองความถี่



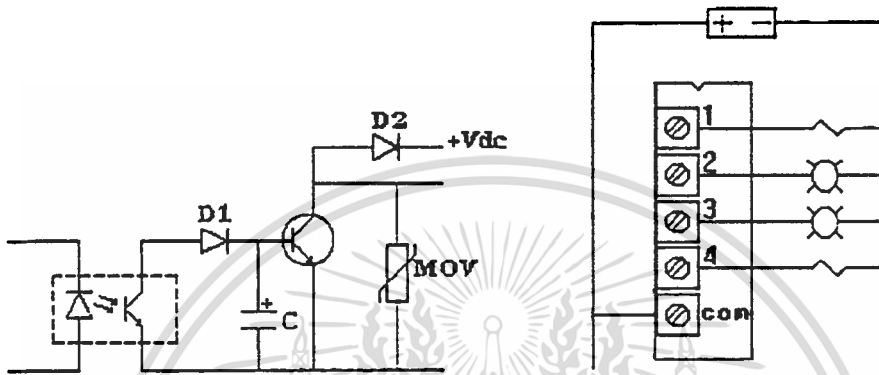
รูปที่ 2.8 วงจรหน่วยเอาต์พุตแบบ AC/DC

วงจรเชื่อมต่อแบบออปติคจะทำหน้าที่แยกระบบไฟของชุด PLC ออกจากอุปกรณ์ภายนอก (ELECTRICAL ISOLATE) เพื่อป้องกันไม่ให้ระบบไฟ ของอุปกรณ์ภายนอก เข้ามาทำความเสียหายให้กับ หน่วยประมวลผลกลาง สวิตซ์อิเล็คทรอนิกส์ ในหน่วยเอาต์พุตนี้มักใช้ ไทริแอด (TRIAC) หรือ SCR ซึ่งเปิด และ ปิดวงจรไฟฟ้าได้โดยไม่เกิดประกายไฟเหมือนหน้าสัมผัสรีเลย์ จึงไม่รบกวนการทำงานของอุปกรณ์ภายนอก และ ยืดอายุการใช้งานของ PL

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หน่วยเอาต์พุตแบบ DC

ทำหน้าที่ในการควบคุมอุปกรณ์เอาต์พุตที่ทำงานด้วยสัญญาณไฟฟ้ากระแสตรงลักษณะการทำงานของวงจรคล้ายกับหน่วยเอาต์พุตแบบ AC แต่ใช้ทรานซิสเตอร์เป็นตัว เปิด-ปิด วงจรไฟฟ้า แทนไทรแอด และ SCR



รูปที่ 2.9 วงจรหน่วยเอาต์พุตแบบ DC และ การเชื่อมต่อหน่วยเอาต์พุตแบบ DC

2.2.3. หน่วยป้อนข้อมูล

ทำหน้าที่ ใช้ในการป้อนตรวจสอบ และ แก้ไขโปรแกรมในหน่วยความจำ ซึ่งโปรแกรมบริหารระบบ จะนำไปประมวลผล เพื่อควบคุมอุปกรณ์ภายนอก อีกที่หนึ่งหน่วยป้อนข้อมูลของ PLC แบ่งออกเป็น 3 ประเภทคือ

เครื่องป้อนโปรแกรมแบบจอภาพ (CRT PROGRAMMER) ประกอบด้วย จอภาพเป็นพิมพ์ และ วงจรอิเล็กทรอนิกส์ที่ทำหน้าที่ติดต่อกับ PLC มีลักษณะคล้ายคอมพิวเตอร์ทั่วไปผู้ใช้สามารถป้อนโปรแกรมโดยใช้ ภาษาแลดเดอร์ หรือ สัญลักษณ์รีเลย์โดยตรง

เครื่องป้อนโปรแกรมขนาดเล็ก (MINI PROGRAMMER) มีลักษณะคล้ายเครื่องคำนวณเลข ประกอบด้วยเป็นพิมพ์ และ ตัวแสดงผลชนิด LCD หรือ LED แสดงผลได้ครั้งละ 1-2บรรทัด เป็นพิมพ์ ประกอบด้วย เป็นพิมพ์ตัวเลข คำสั่ง และ ฟังก์ชันพิเศษต่าง ๆ

คอมพิวเตอร์บริษัทผู้ผลิต PLC บางแห่งได้จัดทำโปรแกรมสำเร็จรูป เพื่อใช้คอมพิวเตอร์ส่วนบุคคล ทำหน้าที่ ป้อนโปรแกรม ติดต่อกับ PLC และ จัดทำรายงานต่าง ๆ

2.3 คำสั่งที่ใช้เขียนโปรแกรม PLC

คำสั่งที่ใช้เขียนโปรแกรม PLC มี 4 ภาษาคือ ภาษาแลดเดอร์ (LADDER DIAGRAM) ภาษานูลีน ภาษาบล็อก และ คำสั่งข้อความภาษาอังกฤษ ซึ่งแต่ละภาษาก็มีวิธีใช้แตกต่างกัน ภาษาแลดเดอร์ และ ภาษานูลีนเป็นภาษาพื้นฐาน ที่ใช้กับ PLC ขนาดเล็ก แทนอุปกรณ์รีเลย์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อุปกรณ์หน่วงเวลา และ อุปกรณ์นับจำนวน ในการควบคุมแบบ ON/OFF ภาษาบล็อก และ คำสั่งข้อความภาษาอังกฤษ เป็นภาษาระดับสูง มักใช้กับการควบคุม ที่ซับซ้อนหรือมีการคำนวณทางคณิตศาสตร์ ที่เกี่ยวข้อง เช่น การควบคุมแบบอะนาลอกและการควบคุมตำแหน่ง

2.3.1 ภาษาแลดเดอร์

ประกอบด้วยสัญลักษณ์หน้าสัมผัส(CONTACT) และ ขดลวด(COIL) มีลักษณะคล้ายวงจรรีเลย์การเขียนภาษาโปรแกรมภาษาแลดเดอร์จากวงจรรีเลย์จึงทำได้ง่ายโปรแกรม ภาษาแลดเดอร์ที่ประกอบด้วย หน้าสัมผัสต่าง ๆ ที่ทำงานร่วมกันเพื่อส่งสถานะการควบคุมไปยังอุปกรณ์เอาต์พุต 1 จุดเราเรียกว่า รังก์ (RUNG) และ บางครั้งในโปรแกรมภาษา แลดเดอร์ 1 รังก์ อาจมีอุปกรณ์เอาต์พุตมากกว่า 1 จุดแต่อุปกรณ์เอาต์พุต เหล่านี้ ต้องได้รับการควบคุมจากจุดเดียวกันเสมอ การเขียนโปรแกรมแลดเดอร์จึงต้องระบุตำแหน่งหรือหมายเลขของอุปกรณ์ที่ต้องการให้ถูกต้อง และ ตรงกันทุกครั้งโดยหมายเลขของหน้าสัมผัส และ ขดลวด นี้ จะสัมพันธ์ กับตำแหน่งหน่วยความจำในส่วนตารางข้อมูล

สัญลักษณ์หน้าสัมผัส หมายถึง การรับสถานะ ของหน่วยอินพุต/เอาต์พุต และ อุปกรณ์ภายใน เพื่อปฏิบัติลอจิกตามเงื่อนไขควบคุม หน้าสัมผัสที่ใช้ในวงจรรีเลย์ หรือ วงจรแลดเดอร์ มีด้วยกัน 2 ชนิดคือ

หน้าสัมผัสปกติเปิด (NORMALLY-OPEN CONTACT) 

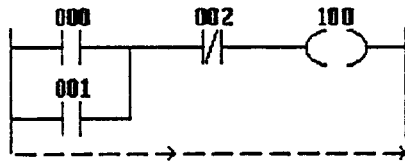
ถ้าหน้าสัมผัสแบบปกติเปิดมีสถานะเป็นจริงหรือ ON หมายถึงการปิดหรือต่อวงจรไฟฟ้า ถ้าหน้าสัมผัสแบบปกติเปิดมีสถานะเป็นเท็จหรือ OFF หมายถึงการเปิดหรือตัดวงจรไฟฟ้า

หน้าสัมผัสปกติปิด (NORMALLY-CLOSED CONTACT) 

ถ้าหน้าแบบสัมผัสปกติปิดมีสถานะเป็นจริงหรือ ON หมายถึงการเปิดหรือตัดวงจรไฟฟ้า ถ้าหน้าสัมผัสแบบปกติปิดมีสถานะเป็นเท็จหรือ OFF หมายถึงการปิดหรือต่อวงจรไฟฟ้า

สัญลักษณ์ขดลวด (COIL) 

หมายถึง การส่งคำสั่งหรือผลการควบคุมไปยังหน่วยเอาต์พุต หรืออุปกรณ์ภายใน รูปแบบการจัดเรียงหน้าสัมผัสทำให้เกิดสถานะการควบคุมที่เรียกว่าสถานะรังก์อุปกรณ์เอาต์พุต จะทำงาน (ON) เมื่อรังก์เป็นจริง หรือ มีเส้นทางผ่านหน้าสัมผัส ปิดจากทางด้านซ้ายของรังก์ มาที่ ขดลวดเอาต์พุต และ สิ้นสุดที่ปลายด้านขวาของรังก์ เกิดขึ้นอย่างน้อย 1 เส้นทาง



รูปที่ 2.10 โปรแกรมภาษาแลคเคอร์ 1 รังค์

2.3.2. ภาษาบูลีน

เป็นภาษาพื้นฐานของ PLC เช่นเดียวกับภาษาแลคเคอร์คำสั่งในภาษาบูลีนมีลักษณะคล้ายกับพีชคณิตบูลีนในปัจจุบันคำสั่งภาษาบูลีนประกอบด้วยกลุ่มคำสั่งเช่นเดียวกับภาษาแลคเคอร์คือวงจรรีเลย์ และ ปฏิบัติลอจิกการหน่วงเวลา และนับจำนวนการคำนวณทางคณิตศาสตร์,การจัดการข้อมูล และ คำสั่งควบคุม โปรแกรม

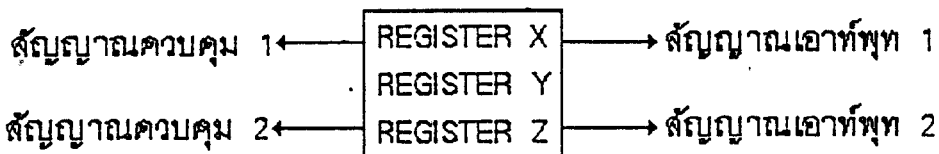
LD	000
OR	001
AND-NOT	002
OUT	020

โปรแกรมภาษาบูลีนที่ทำหน้าที่เหมือนวงจรแลคเคอร์ข้างบน

2.3.3 ภาษาบล็อก

เป็นการเขียนโปรแกรมคำสั่งของ PLC โดยใช้สัญลักษณ์ต่าง ๆ คล้ายกับภาษา แลคเคอร์ แต่จัดไว้ในบล็อกรูปสี่เหลี่ยม ภาษาบล็อกนี้ จะใช้กับคำสั่ง หรือการควบคุมที่ค่อนข้างจะซับซ้อนหรือมีข้อมูลที่เป็นตัวเลขเกี่ยวข้องด้วย เช่นการควบคุมที่มีการคำนวณทางคณิตศาสตร์ และ การควบคุมตำแหน่งเครื่องจักร มักจะใช้ร่วมกับภาษาแลคเคอร์ ภาษาบล็อกแบ่งออกเป็น 4 กลุ่มคือ

1. คำสั่งหน่วงเวลาและนับจำนวน
2. คำสั่งการคำนวณทางคณิตศาสตร์
3. คำสั่งการจัดการข้อมูล
4. คำสั่งการเคลื่อนย้ายข้อมูล



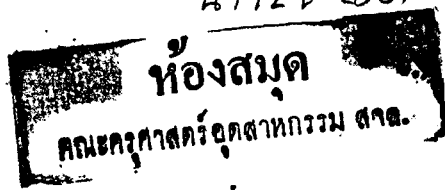
รูปที่ 2.11 ลักษณะคำสั่งบล็อก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปแสดงลักษณะ ของคำสั่งบล็อกซึ่งประกอบไป ด้วยคำสั่งหรือสัญญาควบคุมใช้รับ สัญญาจาก หน่วยอินพุต/เอาต์พุต และ อุปกรณ์ภายในสัญญาเอาต์พุตของคำสั่ง บล็อกจะทำหน้าที่ควบคุม และ แสดงผลการปฏิบัติของคำสั่งบล็อก

2.3.4 คำสั่งข้อความภาษาอังกฤษ คำสั่งข้อความภาษาอังกฤษ ที่ใช้กับ PLC จะถูกคัดแปลงจาก ภาษาระดับสูงของคอมพิวเตอร์ เช่น ภาษา BASIC และ PASCAL ทำให้การเขียนโปรแกรมมีความคล่อง ตัวสูง และ มีประสิทธิภาพสูงขึ้นโปรแกรมที่เขียนขึ้นเข้าใจง่าย และ เหมาะสำหรับ แก้ไขในภายหลัง





บทที่ 3

การออกแบบ PLC

3.1 คุณสมบัติของเครื่อง PLC

คุณสมบัติโดยทั่วไป

1. CPU (CENTRAL PROCESSOR UNIT) ใช้ CPU ของ ZILOG เบอร์ Z80180 ทำงานที่ความถี่ 6.144 MHZ (ใช้ความถี่นาฬิกา 12.288 MHZ)
2. ROM (READ ONLY MEMORY) ใช้ EPROM เบอร์ 27256 ความจุทั้งหมด 32 K BYTE อยู่ที่ตำแหน่ง 00000H-07FFFH
3. RAM (RANDOM ACCESS MEMORY) ใช้หน่วยความจำ STATIC ขนาด 32K BYTE เบอร์ 62256 อยู่ที่ตำแหน่ง 10000H-17FFFH
4. หน่วย อินพุต/เอาต์พุต (I/O) มี 16 INPUT, 8 OUTPUT พร้อม LED แสดงสถานะการทำงาน และ จุดต่อแบบสกรู (SCREW TERMINALS)
5. อุปกรณ์ภายใน (INTERNAL DEVICE) มีอุปกรณ์หน่วงเวลา (TIMER) 16 ตัว และ อุปกรณ์ นับจำนวน (COUNTER) 8 ตัว และ รีเลย์ช่วย (AUXILIARY RELAY) 240 ตัว
6. หน่วยแสดงผล (DISPLAY) เป็น LCD ขนาด 2x16 ตัวอักษร
7. แป้นพิมพ์ (KEYBOARD) ขนาด 32 ปุ่ม คือ ปุ่มสำหรับป้อนคำสั่งข้อมูลที่เป็นตัวเลข ปุ่มสำหรับการแก้ไข โปรแกรมและปุ่มสำหรับสั่งงาน
8. ความจุโปรแกรม ป้อนโปรแกรมได้สูงสุด 1000 บรรทัด
9. แหล่งจ่ายไฟ (POWER SUPPLY) 9Vdc, 12Vdc
10. แหล่งจ่ายไฟ OUTPUT ภายนอก 5Vdc, 12Vdc, 24Vdc

3.2 ส่วนต่าง ๆ ของการออกแบบ

3.2.1 หน่วยประมวลผล

หน่วยประมวลผลของ PLC ชุดนี้ใช้บอร์ดควบคุม CP-JR180 ซึ่งใช้ไมโครโปรเซสเซอร์ เบอร์ Z80180 เป็น CPU

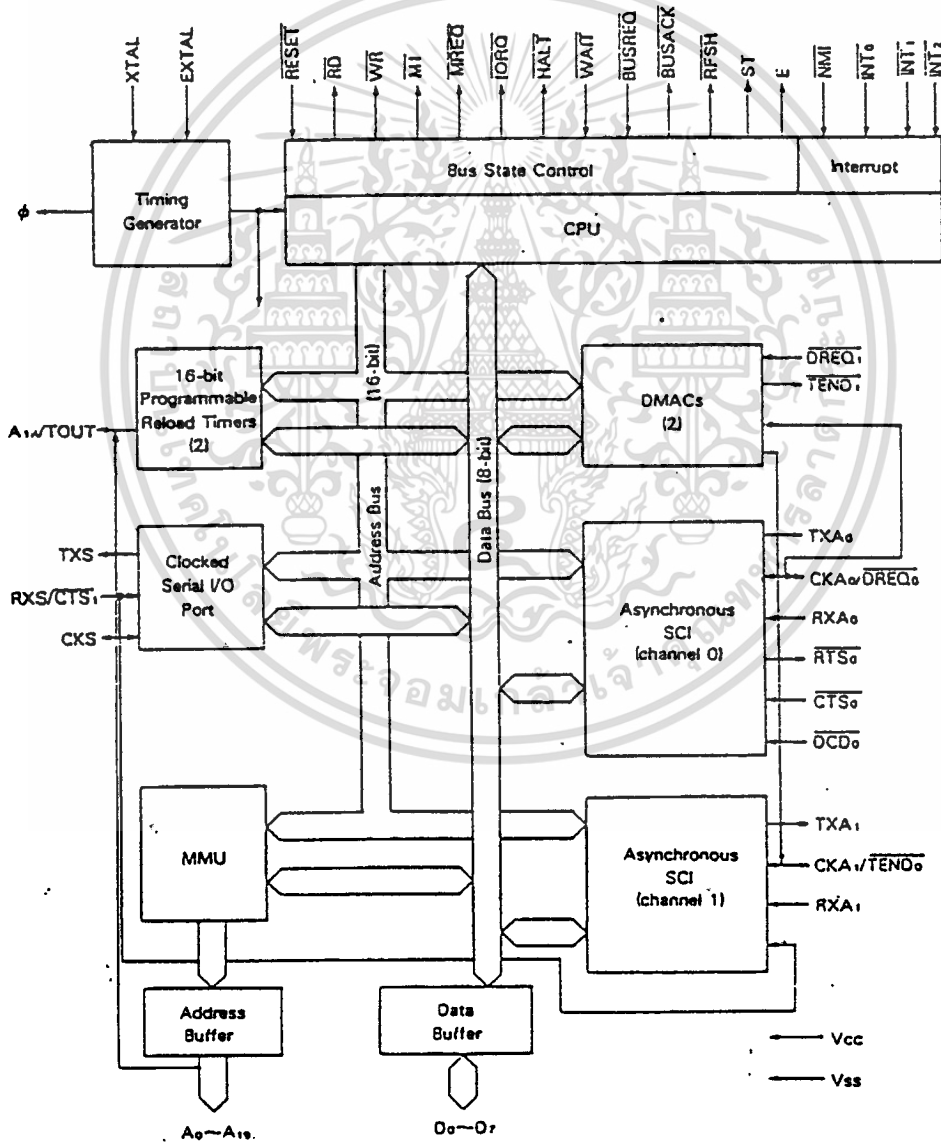
- ใช้ CPU เบอร์ Z80180 แบบ 68 PIN PLCC TYPE RUN ด้วยความเร็ว 6.144MHZ เป็น CPU ประจำบอร์ด
- อ้างอิงหน่วยความจำ RAM และ ROM ได้ถึง 128K BYTE ON BOARD
- มี PORT 82C55 ให้ใช้งาน 1 ตัว โดยเป็นแบบ CMOS TYPE และ อยู่ในรูปตัวถัง PIN 40 ขนาดเล็ก ทำให้ลดขนาดของ PCB ได้มาก
- มี PORT ต่อเข้ากับส่วนแสดงผลจอภาพแบบ LCD ได้ทั้ง LCD แบบ CHARACTER หรือแบบ GRAPHI TYPE
- มีวงจรส่วน REAL TIME CLOCK เป็นฐานเวลาจริงให้กับระบบโดยใช้ เบอร์ 6242
- ใช้ไอซี เฉพาะงาน DS1232 โดยเป็นวงจร POWER RESET และ เป็นวงจร WATCH DOG เพิ่ม ความมั่นใจในการใช้งานมากขึ้น
- ใช้กับแรงดัน 9-7 โวลต์ โดยมีไอซี REGULATOR 7805 ในตัว
- มีคอนเน็คเตอร์ มาตรฐานสามารถต่อเข้ากับบอร์ด ต่างๆในท้องตลาดได้เป็นอย่างดี
- มี ข้อมูล BOARD JR-180 2SERIAL PORT ต่อใช้งานใช้ max 232

CPU Z80180 ของบริษัท ZILOG โดยเป็น SUPPER SET ของ CPU Z80 ใช้คำสั่งของ Z80 ทั้งหมด และ ยังเพิ่มอีก 12 ชุด คำสั่งใช้งานเช่นคำสั่งคูณ (MLT) Z80180 ความเร็วขนาด 6 MHZ แต่ก็สามารถใช้กับความถี่ 6.144MHZ ได้ด้วยทำTST g (คำสั่ง TEST BIT ใน REGISTER) เป็นต้น ในบอร์ด JR-180 นี้เราเลือกใช้ให้ทำงาน 1 คำสั่งใช้เวลาเพียง 0.48 uSEC เท่านั้น ปัญหาที่ตามมาก็คือ จำเป็นต้องใช้ EPROM หรือ RAM ที่มี ACCESS TIME ต่ำ ๆ ได้เท่านั้น หรือซึ่งในหัวข้อนี้ทาง ZILOG ได้ออกแบบให้แก้ไขจุดนี้ได้ โดยสามารถ SET WAIT STATE ภายในตัว CPU ในการ READ WRITE MEMORY หรือในการ READ,WRITE I/O DEVICE ได้ด้วยซึ่งเป็นข้อดี

Z80180 เป็น CPU ที่มีความสามารถสูงที่ได้รวม CHIP สำคัญอื่น ๆไว้ใน CPU CHIP เดียวจึงทำให้มีลักษณะคล้ายกับ CPU ที่ใช้ในงานควบคุมในจำพวก "SINGLE CHIP" แต่เนื่องจาก SINGLE CHIP มีข้อดีคือเป็นระบบเล็กราคาถูก แต่มีข้อเสียคือการโปรแกรมควบคุมค่อนข้างยากในตอนเริ่มต้น และกับระบบงานขนาดใหญ่ แต่ Z80180 ทางด้านโปรแกรมจะสะดวกอย่างมากเพราะคำสั่งที่ใช้มี มากกว่า และ ตรงไปตรงมาทั้งคู่มือภาษาไทย และตัวอย่างการใช้งานอย่างมากมาย เพราะ CPU Z80180 นี้เป็น SUPPER COMPAT Z80 คือคำสั่งทั้งหมดยังเป็น Z80 และ ได้เพิ่มชุดคำสั่ง ขึ้นมา เพื่อเพิ่มความสะดวกในการใช้งานขึ้นอีก เมื่อมองดูระบบ MICRO CONTROLLER "SINGLE CHIP" แล้ว Z80180 จะดีกว่าตรงที่ไม่มี ROM, RAM, PORT แต่ถ้าเป็นงานในระดับอุตสาหกรรมแล้วระบบของ Z80180 กับ CHIP MICRO CONTROLLER

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จะไม่ต่างกันเลยเพราะ ความต้องการเนื้อที่ในการเก็บข้อมูลมาก และ PORT มากตามจึงทำให้ ต้องต่อเพิ่มภายนอกขึ้นจึงทำให้ Z80180 ในระดับงานควบคุมอุตสาหกรรมค่อนข้างตัวมากกว่าเพราะ ภายใน Z80180 ประกอบด้วย CMOS , OSCILATOR ในตัว RUN ที่ 10MHZ MMU CHIP อ่าง MEMORY ได้ 1 MBYTE , DMA 2 CHANEL , PORT สื่อสาร UART CHANAL , CLOCK SERIAL I/O , 16 BIT TIMER COUNTER และเกี่ยวกับ PORT สื่อสาร สามารถทำ MULTI PROCESSOR COMMUNICATION



รูปภาพที่ 3.1 โครงสร้างภายในของ Z80180

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นอนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขบวนการใช้งาน

A0-A19 ADDRESS BUS ระหว่าง RESET จะเป็น HIGH IMPEDANCE

BUSAK CKNOWLEDGE เป็น ขา OUTPUT ACTIVE LOWทำงานก็ต่อเมื่อ Z80180 ตอบสนองต่อการขอ BUS ของBUSRQ และ จะทำให้ BUS ข้อมูล BUS ADDRESS สัญญาณ CONTROL บางเส้นเป็น HIGH IMPEDANCE

BUSRQ BUS REQBUS เป็นขา INPUT ACTIVELOW ซึ่งจะมีความสำคัญสูงกว่า NMI โดยมีการตรวจสอบ สัญญาณนี้ทุก ๆ การสิ้นสุดของMACHINE CYCLE

CXA0,CXA1 ASYNCHRONOUS CLOCK 0 และ 1 เป็นขาสัญญาณ CLOCK แบบ 2 ทิศทาง คือการใช้เป็นขา INPUT หรือ OUTPUT ก็ได้

CKS SERIAL CLOCKเป็นขา CLOCK 2ทิศทางของ CSI/O

CLOCK เป็นขาOUTPUT โดยจะเป็นครึ่งหนึ่งของ XTAL หรือ CLOCKOUT เช่น XTAL 12 MHZ Z80180 จะRUNที่ 6 MHZ

CTS0,CTS1 CLEAR TO SEND 0 และ 1 เป็นขา INPUT ACTIVE LOW ในการควบคุม MODEM

D0-D7 DATA BUS เป็นแบบ 2 ทิศทาง

DCD0 DATA CARRIER DETECT 0 เป็นขา INPUT ACTIVE LOW ใช้ควบคุมในการติดต่อกับ MODEM ของ ASCII CHANEL 0

DREQ0,DREQ1 DMA REQUEST 0 และ1 เป็นขา INPUT ACTIVE LOW ใช้ในการขอ DMA และ ขานี้ จะโปรแกรมได้ให้ตรวจสอบสัญญาณที่ ขอบ หรือ ระดับได้

E ENABLE CLOCK เป็นขา OUTPUT ACTIVE HIGH ซึ่งใช้บังคับการทำงานกับอุปกรณ์ภายนอก ระหว่างการทำงานเกี่ยวกับ BUS และ ใช้เชื่อมต่อกับอุปกรณ์ในตระกูล 68XX และ 80XX

HALT เป็นขา OUTPUT ACTIVE LOW จะทำงานเมื่อทำคำสั่ง HALT หรือ SLP

INT0 MASKABLE INTERRUPT 0 เป็นขา INPUT ACTIVE LOW สัญญาณขานี้จะถูกตรวจทุกๆสิ้นสุดคำสั่ง

INT1,INT2 เช่นเดียวกับ INT0 แต่มีระดับความสำคัญรองลงมาตามลำดับ

IORQ เป็นขา OUTPUT เพื่อบอกว่าจะติดต่อกับ I/O หรือขา IOE ใน 64180

M1 MACHINE CYCLE 1 เป็นขา OUTPUT ACTIVE LOW จะทำงานเมื่อ FETCH OP-CODE หรือเป็นขา LIR ของ 64180

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

NMI NON MASKABLE INTERRUPT เป็นขา INPUT ACTIVE LOW ขานี้จะตอบรับการ INTERRUPT เสมอโดย ไม่สามารถหยุดด้วย SOFTWARE RD เป็นขาที่ใช้ในการอ่านข้อมูลจาก MEMORY หรือ I/O RFSH เป็นขาที่ให้ ADDRESS LOW (A0-A7) ไป REFRESH DYNAMIC RAM หรือ ขา RFE ของ 64180

RTS0 REQUEST TO SEND เป็นขา OUTPUT ACTIVE LOW ขานี้ใช้ควบคุมสัญญาณ MODEM ของ ASCII CANEL 0

RXA0,RXA1 RECEIVE DATA0 และ 1 เป็นขารับสัญญาณจาก SERIAL PORT ของ ASCII

RXS CLOCK SERIAL RECEIVE DATA เป็นขารับสัญญาณจาก SERIAL ของ CSIO

ST STATUS เป็นขา OUTPUT ACTIVE HIGH ให้แสดงสถานะการทำงานของ CPU โดยร่วมกับ M1 และ HALT ดังตาราง

ST	HALT	M1	OPERATION
0	1	0	CPU OPERATION (1st OP-CODE FETCH)
1	1	0	CPU OPERATION (2nd OP-CODE AND 3rd OP-CODE FETCH)
1	1	1	CPU OPERATION (MC EXCEPT FOR OP-CODE FEYCH)
0	X	1	DMA OPERATION
0	0	0	HALT MODE
1	0	1	SLEEP MODE (INCLUDING) SYSTEM STOP MODE)

NOTE X: DON'T CARE

MC: MACHINE CYCLE

ตารางที่ 3.1 การทำงานของขา ST

TEND0,TEND1TRANSFER END 0 และ 1 เป็นขา OUTPUT ACTIVE LOW ใช้แสดงถึงว่าทำ DMA ถิ่นสุดลงแล้ว

TOUT TIMER OUT ใช้กำเนิดพัลส์จาก PRT CHANEL 1

TXA0,TXA1 TRANSMIT DATA 0 และ 1 เป็นขาส่งข้อมูล SERIAL ของ ASIC

TXS CLOCK SERIAL TRANSMIT DATA เป็นขาส่งข้อมูล SERIAL ของ CSIO

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษานี้เท่านั้น เมื่ออนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

WAIT ขา INPUT ACTIVE LOW จะถูกตรวจที่ขอบขาของ CLOCK ลูกที่ 2 ของทุก ๆ MACHINE เพื่อเป็นการรอให้อุปกรณ์ภายนอกทำงานให้ทันกับการทำงานของ CPU

WR ใช้สำหรับการส่งข้อมูลไปยัง I/O หรือ MEMORY

XTAL เป็นขาที่ใช้ต่อกับ XTAL

ขาที่ MULTIPLEX

A18/TOUT ระหว่าง RESET จะเป็น A18 แต่ถ้ามีการเลือก SET BIT TOC1 หรือ TOC0 ใน TIMER CONTROL REGISTOR (TCR) จะทำหน้าที่ เป็นTOUT

TCKA0/DREQ0 ระหว่าง RESET ขานี้จะเป็น CKA0 แต่ถ้า DM1 หรือ SM1 ใน DMA MODE REGISTOR (DMODE) ถูก SET เป็น 1 จะเป็นขา DREQ0

CKA1/TEND0 ระหว่าง RESET จะเป็นขา CKA1 แต่ถ้า BIT CKA1D ใน ASCII ถูก SET จะเป็นขา TEND0

RXS/CTS1 ระหว่าง RESET ขา นี้จะเป็น RXS ถ้า BIT CTS1E ถูก SET จะเป็นขา CTS1 ส่วนของ MEMORY

CP-JR180 สามารถต่อใส่หน่วยความจำได้สูงสุด 128K BYTE ON BOARD โดยใช้ไอซี DECODE แบ่งหน่วยความจำเป็นช่วงๆ ได้ 8 ช่วง ช่วงละ 32K BYTE ใช้ A15 ,A16 ,A17 มาเข้ายัง IC U6 เบอร์ 74ALS138 และ ใช้สัญญาณควบคุม การติดต่อหน่วยความจำ MREQ และ A19 มาควบคุมขา G2A G2B เพื่อให้เกิดสัญญาณ CS เฉพาะ การติดต่อกับหน่วยความจำเท่านั้น

SOCKET U2 สามารถใส่ EPROM ขนาด 64K BYTE (27512) หรือ 23K (27256) ได้โดยใช้ JUMPER J1 เป็นตัวเลือกเบอร์ EPROM และเราใช้ DIODE เบอร์ 1N4148 2 ตัวต่อในลักษณะ AND GATE ให้ DECODE ได้ 2 ช่วง ADDRESS

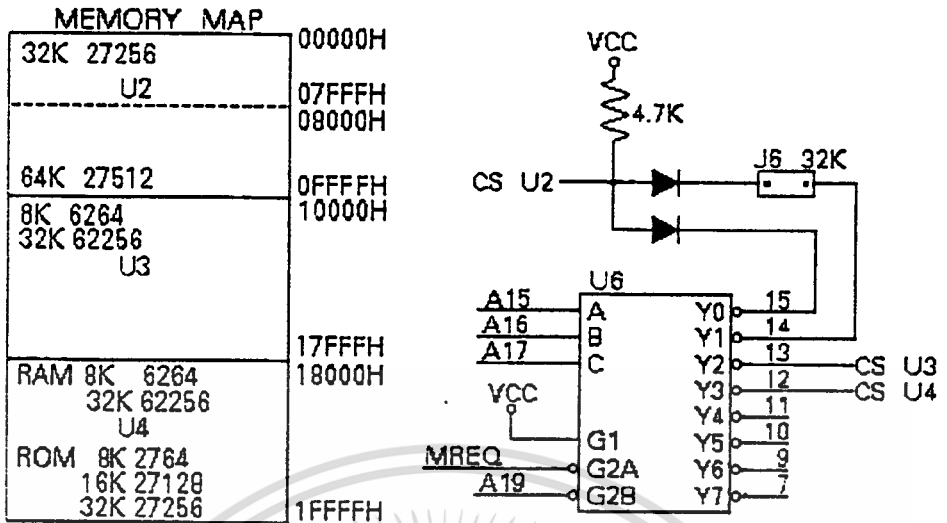
U2 นี้หน่วยความจำเริ่มจาก 0000H ถึง FF

SOCKET U3 สามารถใส่ RAM ขนาด 32K BYTE (62256) หรือ 8K BYTE (6264) ได้โดยใช้ JUMPER J4 เป็นตัวเลือกเบอร์ RAM ในวงจรส่วนนี้เราสามารถใส่ BATTERY ขนาด 3 VOLT เพื่อ BACKUP ข้อมูลใน RAM ได้ด้วยโดยใช้ MOSFET เบอร์ BS170 เป็นส่วนกันสัญญาณรบกวนจากการเปิดปิดระบบไฟไม่ให้เกิดเข้ารบกวน ขา CS ของ RAM เพื่อผลการ BACKUP ที่ดี

U3 นี้หน่วยความจำเริ่มจาก 10000H-17FFFFH

SOCKET เราสามารถใส่ RAM หรือ ROM ได้โดยใช้ JUMPER SW เป็นตัว เลือกเบอร์ ไอซี ที่เราจะใส่ และ ใช้ JUMPER J3 เป็นตัวประนในกรณีจะใช้ RAM และ ก็ต้องการ BACKUP ข้อมูลด้วย

U4 นี้หน่วยความจำเริ่มจาก 18000H ถึง 1FFFFH เท่านั้น ไม่นอนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.2 MEMORY MAP CP-JR 180

ส่วนของ PORT

ในส่วนนี้เราใช้ ไอซี ดีโค้ด U7 74ALS138 เป็นตัว ดีโค้ดแบ่งช่วง แอดเดรส พอร์ต โดยใช้สัญญาณควบคุม I/O DEVICE (IORQ) และ M1 มาควบคุมเพื่อให้เกิดสัญญาณ CS ขึ้นเฉพาะในการติดต่อกับส่วน I/O เท่านั้น

8255 พอร์ต เราใช้ไอซี พอร์ต ใช้งานประจำบอร์ดโดยเป็น พอร์ต ขนาด 8 BIT 3 พอร์ต ใช้งานโดยมี ตำแหน่ง แอดเดรสดังนี้

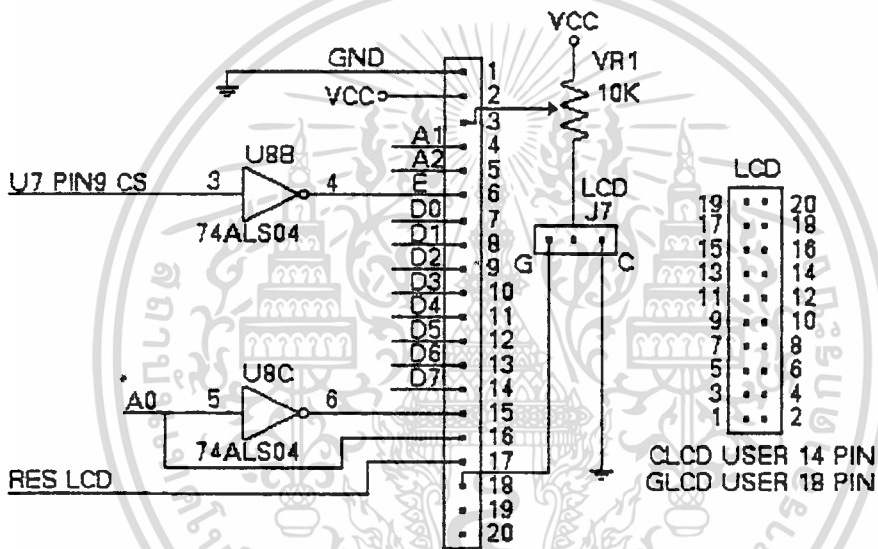
- พอร์ต A = 80H
- พอร์ต B = 81H
- พอร์ต C = 82H
- พอร์ตควบคุม = 83H

RTC พอร์ต ใช้ไอซี REAL TIME CLOCK / CALENDAR ชนิด CMOS เบอร์ 6264 โดยต่อขา ADDRESS BUS และ DATA BUS ขนาด 4 บิต เข้าโดยตรงกับ CPU REALTIME CLOCK ตัวนี้จะสามารถบอกได้ทั้ง วัน, เดือน, ปี, วันในรอบสัปดาห์ ชั่วโมง, นาที, วินาที และปีในส่วนของปีอธิกสุรทินได้ด้วย มีตำแหน่ง แอดเดรส จาก A0H ถึง BFH

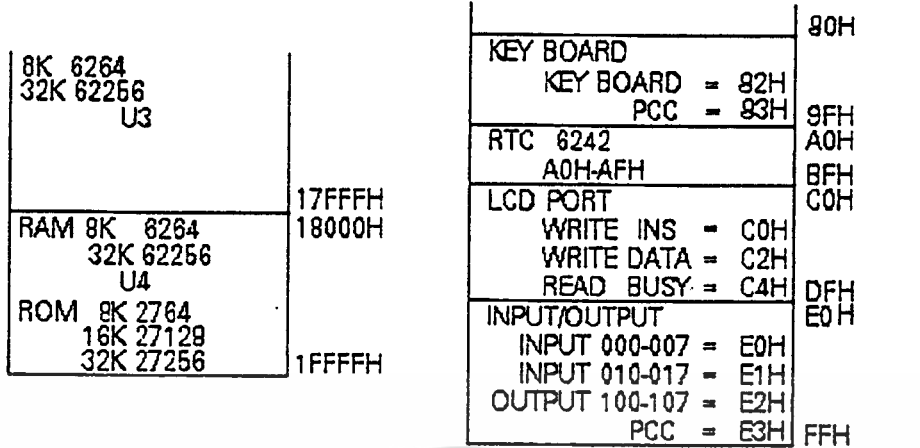
LCD พอร์ต สามารถต่อเข้ากับโมดูล LCD ได้ทั้งแบบตัวอักษร และ แบบกราฟฟิก โดยตรงไม่ต้องต่อผ่านพอร์ตอีกโดยออกแบบให้ต่อเข้ากับ บัสของ Z80180 ได้ โดยตรง เราใช้สัญญาณ ดีโค้ดจาก U7 (Y6) ผ่าน INVERTOR U8B เพื่อ กลับสถานะของสัญญาณสร้างเป็น

สัญญาณ E CLOCK ของ LCD ใช้ A1 และ A2 เป็น สัญญาณ RS และ R/W และใช้ U8C เป็น ตัวกำหนด เพจ (PAGE) การทำงานของ LCD ในแบบกราฟฟิก จากสัญญาณแอดเดรส A0

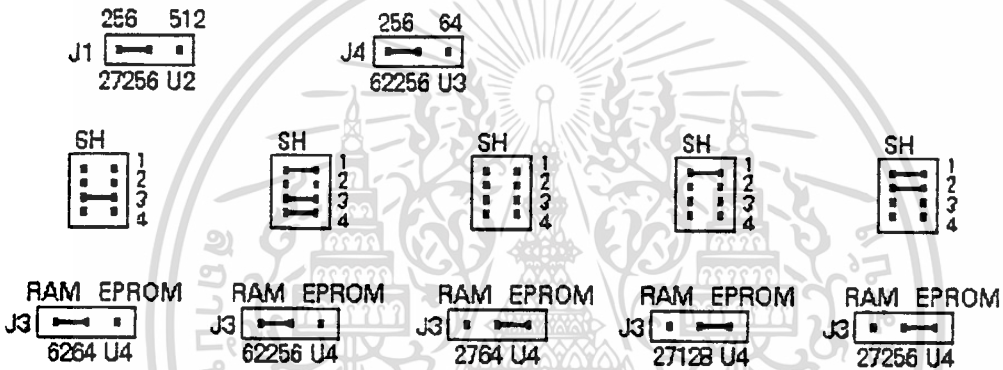
LCD ADDRESS PORT CHANNEL PORT	ADDRESS PORT
WRITE DATA INSTRUCTION	C0H
WRITE DATA TO CG OR DD RAM	C2H
READ BUSY FLAG AND ADDRESS	C4H



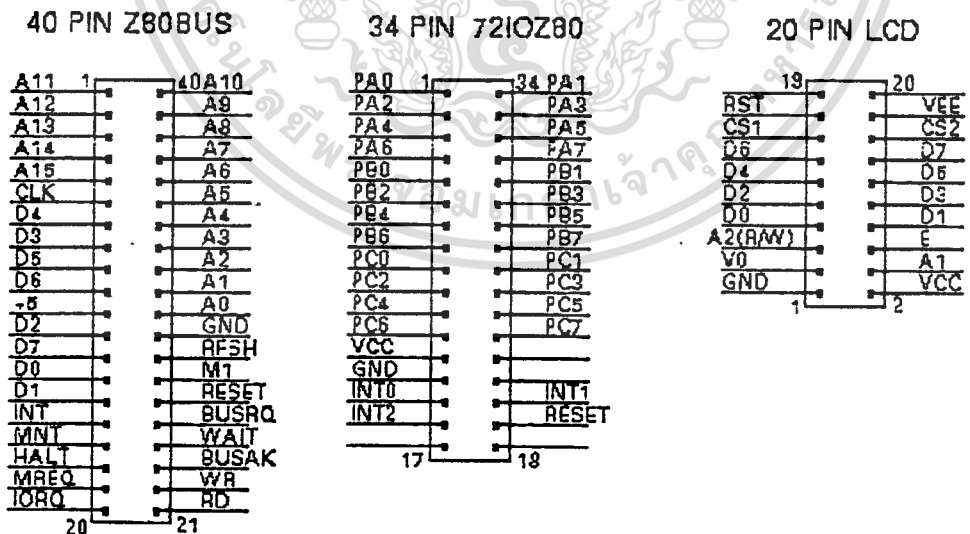
รูปที่ 3.3 การต่อ PORT CP-JR 180



(ก)0



(ข)



รูปที่ 3.4

(ก). ตำแหน่งแอดเดรสอุปกรณ์ต่าง ๆ

(ข). SET JUMPER

(ค). BUS ติดต่ออุปกรณ์อินพุต/เอาต์พุต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาดูเท่านั้น เมื่อผู้ผู้ใดให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.2 หน่วยอินพุท/เอาต์พุท

หน่วยอินพุท/เอาต์พุท จะทำหน้าที่ติดต่อระหว่าง PLC กับอุปกรณ์ภายนอก โดยหน่วยอินพุท ทำหน้าที่รับสถานะ และ วัดค่าจากอุปกรณ์ภายนอกเช่นการ ON/OFF ของสวิตช์ตำแหน่ง เครื่องจักร ระดับของเหลว อุณหภูมิ ความดัน ระดับแรงดัน และ กระแสไฟฟ้าจะส่งต่อไปให้ PLC และ PLC จะใช้ค่า หรือ สถานะจากหน่วยอินพุท/เอาต์พุท เป็นข้อมูลในการประมวลผลตามผู้ใช้ และส่งผลที่ได้ไปที่หน่วยเอาต์พุทเพื่อควบคุมอุปกรณ์ภายนอก เช่น รีเลย์ มอเตอร์ไฟฟ้า ป้อนวาล์ว

สัญญาณไฟฟ้าที่อุปกรณ์ อินพุท/เอาต์พุท แบบตรรกะใช้มีหลายค่าทั้งไฟฟ้ากระแสตรง และ ไฟฟ้ากระแสสลับ แต่ใน PLC จะให้สัญญาณไฟฟ้าระดับ TTL เท่านั้น

PLC มี 16 อินพุท, 8 เอาต์พุท โดยหน่วยอินพุทอยู่ที่หมายเลข 000-007 และ หมายเลข 010-017 หน่วยเอาต์พุทอยู่ที่ หมายเลข 100-107 โดยมีหลอดไฟ (LED) จะแสดงสถานะของหน่วยอินพุท/เอาต์พุทแต่ละตัวซึ่ง LED จะสว่างเมื่อหน่วยอินพุท เอาต์พุทอยู่ในสถานะทำงานหรือ ON และ LED จะดับเมื่อหน่วยอินพุท/เอาต์พุทอยู่ใน สถานะไม่ทำงาน หรือ OFF

การอ้างถึงอุปกรณ์ หน่วยอินพุท/เอาต์พุท

หน่วยอินพุทอ้างถึง ได้ด้วยคำสั่ง LD, LD-NOT, AND, AND-NOT, OR, OR-NOT, SET, RES

หน่วยเอาต์พุทอ้างถึง ได้ด้วยคำสั่ง OUT, LD, LD-NOT, AND, AND-NOT, OR, OR-NOT, SET, RES

I/O FORCING

อุปกรณ์หน่วย อินพุท/เอาต์พุท จะมีข้อแตกต่างจากอุปกรณ์อื่น ๆ คือ เราสามารถบังคับให้อุปกรณ์ I/O อยู่ในสถานะการ ON หรือ OFF ตลอดเวลาได้ ซึ่งเราเรียกว่าการ FORCING ซึ่ง I/O ที่ได้รับการ FORCING นี้จะคงสถานะ ON หรือ OFF ไว้ตลอดเวลาโดยจะไม่ขึ้นอยู่กับผลลัพธ์ของโปรแกรมควบคุม(ในกรณีอุปกรณ์เอาต์พุท) หรือสถานะแท้จริงของ อุปกรณ์ภายนอก (ในกรณีของอุปกรณ์อินพุท) ซึ่งวิธีจะทำให้การหาข้อผิดพลาดของโปรแกรม (DEBUGING) ทำได้ง่ายขึ้น

วิธีการ FORCING I/O

FORCE ให้ ON ด้วยคำสั่ง SET

FORCE ให้ OFF ด้วยคำสั่ง RES

ข้อควรระวังในการ FORCING I/O

ในกรณีที่เรทำการ FORCING อุปกรณ์เอาต์พุทให้ ON นั้นเมื่อเรากดปุ่ม SYSTEM RESET จะทำให้การ FORCING ถูกยกเลิกเพราะฉะนั้นเราควรแน่ใจว่า การยกเลิก FORCING นี้จะไม่ทำให้เกิด อันตรรกะกับงานที่ทำการควบคุมอยู่

การ FORCING นี้จะใช้ได้กับอุปกรณ์หน่วย อินพุท/เอาต์พุทเท่านั้นเราจะใช้กับ รีเลย์ช่วย (AUXILIARY RELAY), อุปกรณ์หน่วงเวลา (TIMER) หรืออุปกรณ์นับจำนวน (COUNTER) ไม่ได้

รีเลย์ช่วย (AUXILIARY RELAY)
ทำหน้าที่เหมือน กับรีเลย์ควบคุม (CONTROL RELAY) ในวงจรไฟฟ้า คือใช้เป็นตัวเก็บผลการประมวลผลชั่วคราว เพื่อช่วยให้การออกแบบโปรแกรมควบคุมทำได้ง่ายขึ้น

การอ้างถึงส่วนต่าง ๆ ของรีเลย์ช่วย

หน้าสัมผัสปกติเปิด (NOMALLY OPENED) อ้างโดยใช้คำสั่ง LD, AND, OR

หน้าสัมผัสปกติปิด (NOMALLY CLOSED) อ้างโดยใช้คำสั่ง LD-NOT, AND-NOT, OR-NOT

ขดลวด (COIL) อ้างโดยใช้คำสั่ง OUT

สำหรับหน้าสัมผัส (CONTACTS) ทั้งแบบปกติเปิด และ แบบปกติปิดนั้น เราสามารถเรียกใช้กี่ครั้งก็ได้ในวงจรรีเลย์ซึ่งใน PLC มีรีเลย์ช่วยอยู่ 240 ตัวอยู่ที่หมายเลข 200-207, 210-217, ..., 490-497

อุปกรณ์หน่วงเวลา (TIMER)

อุปกรณ์หน่วงเวลาทำหน้าที่หน่วงเวลาในวงจรรีเลย์เพื่อ ปิด หรือ เปิด วงจรไฟฟ้า และ เมื่อเวลาผ่านไประยะเวลาหนึ่งอุปกรณ์หน่วงเวลาที่ใช้ใน PLC จะเป็นแบบชนิดเก็บค่า (RELENTIVE TIMER) ซึ่งสามารถหยุดการหน่วงเวลาชั่วคราวได้โดยอุปกรณ์หน่วงเวลา จะทำการนับเวลาทุก ๆ 0.1 วินาที และ เราสามารถกำหนดได้จาก 0-999 หรือนานที่สุด 99.9 วินาที

ใน PLC มีอุปกรณ์หน่วงเวลาที่ใช้ได้ 16 ตัวจากหมายเลข 500-507 และหมายเลข 510-517
TIMER ใน PLC ประกอบด้วยส่วนต่าง ๆ ดังนี้

1. สัญญาณควบคุม 2 สัญญาณคือ RESET/ENABLE INPUT และ START/STOP INPUT
2. เอาต์พุทที่แสดงสถานะของ TIMER คือ TIMED-OUT OUTPUT
3. ส่วนที่เก็บค่าเวลาปัจจุบันของ TIMER และ ค่าเริ่มต้น (SETPOINT) ที่กำหนดไว้



ขาสัญญาณต่างๆจะมีผลต่อการทำงานของ TIMER ดังนี้

1. ขา RESET/ENABLE INPUT

- ช่วง ON จะเป็นการ ENABLE TIMER ให้อยู่ในช่วง TIMING STATE ขา TIMED-OUT จะขึ้นอยู่กับค่าเวลาปัจจุบันของ TIMER

- ช่วง OFF จะเป็นการ RESET TIMER ให้อยู่ในช่วง RESET STATE ขา TIMED-OUT จะอยู่ในสถานะ OFF

- ช่วงการเปลี่ยนจากสถานะจาก OFF เป็น ON ทำให้ค่าปัจจุบันของ TIMER มีค่าเท่ากับค่าเวลาที่ตั้งไว้ (SETPOINT) ขา TIMED-OUT จะ OFF และ TIMER จะเข้าสู่ TIMING STATE ในกรณีที่เราตั้งค่า SETPOINT ให้เท่ากับศูนย์ ขา TIMED-OUT จะ ON และ TIMER จะเข้าสู่ TIMED-OUT STATE ทันที

2. ขา START/STOP INPUT ทำหน้าที่ควบคุมการลดค่าเวลาลงของ TIMER ในช่วง TIMING STATE

- ช่วง ON TIMER จะลดค่าลงทุก ๆ 0.1วินาที

- ช่วง OFF TIMER จะหยุดการลดค่าลงชั่วคราว

- การเปลี่ยนแปลงสถานะต่าง ๆ ของขา START/STOP INPUT จะไม่มีผลต่อขา TIMED-OUT OUTPUT

3. ขา TIMED-OUT OUTPUT จะ ON เมื่อขา RESET/ENABLE INPUT อยู่ในสถานะ ON และ ค่าปัจจุบันของ TIMER มีค่า=0 (TIMED-OUT STATE) เท่านั้น โดยไม่ขึ้นกับสถานะของขา START/ STOP INPUT

เมื่อเริ่มจ่ายไฟให้กับ ระบบ TIMER จะเข้าสู่สถานะ RESET STATE และ ขา TIMED-OUT OUTPUT จะ OFF ค่า SETPOINT และ ค่าเวลาปัจจุบันของ TIMER จะเป็น 0 ทั้งคู่

การอ้างขาต่าง ๆ ของ TIMER

ขา RESET/ENABLE INPUT อ้างได้จากคำสั่ง TIM XXX , K NNN

เมื่อ NNN = ค่า SETPOINT ของ TIMER (0-999)

XXX = หมายเลขของ TIMER (500-507 และ 510-517)

ขา START/STOP INPUT อ้างได้จากคำสั่ง OUT XXX

ขา TIMED-OUT OUTPUT อ้างได้จากคำสั่ง LD, AND, OR-NOT...XXX

ข้อควรระวังในการใช้ TIMER

เนื่องจาก เมื่อ TIMER อยู่ในสถานะ ENABLE นั้น TIMER จะลดค่าลงเรื่อย ๆ โดยไม่ขึ้นกับวงจรควบคุม (RELAY LADDER LOGIC) ดังนั้นจึงอาจทำให้เกิดปัญหาขึ้นในระบบควบคุม

ได้ โดยปัญหาจะเกิดขึ้นในกรณีที่เรานำ TIMED-OUT CONTACTS ไปควบคุมหลาย ๆ จุดใน
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรม เช่น ถ้าเรานำเอา TIMED-OUT CONTACT ไปควบคุมจุด 2 จุดในโปรแกรมขณะที่โปรแกรมสแกนมาที่จุดแรก TIMER ยังคงอยู่ในสถานะ TIMING STATE แต่กว่าที่โปรแกรมสแกน มาถึงจุดที่ 2 นั้น TIMER ได้เปลี่ยนสถานะจาก TIMING STATE มาเป็น TIMED-OUT STATE แล้ว จึงทำให้การทำงานของ โปรแกรมผิดไป

วิธีป้องกันปัญหานี้จะทำได้โดยการใช้TIMED-OUT CONTACTเพียงจุดเดียวในโปรแกรม โดยนำไปควบคุม AUXILIARY RELAY แล้วจึงนำ CONTACT ของ AUXILIARY RELAY นี้ ไปควบคุม จุดต่าง ๆในโปรแกรมแทน TIMED-OUT CONTACT ของ TIMER

ข้อควรระวังอีกอย่างหนึ่ง ในการกำหนดค่าเวลาให้ TIMER คือ หน่วยที่ใส่จะต้องมีค่าเป็น 0.1 วินาที และ มีค่าอยู่ระหว่าง 0-999 เท่านั้น เช่น ถ้าต้องการหน่วงเวลา 5 วินาที จะต้องตั้งค่า SETPOINT=50

ถ้ากำหนดค่าSETPOINTให้ TIMER มากกว่า 1 แห่งในโปรแกรมจะถือเอาค่า SETPOINT ในรั้งล่าสุดท้ายเป็นหลัก

อุปกรณ์นับจำนวน (COUNTER)

อุปกรณ์นับจำนวน (COUNTER) จะทำหน้าที่นับจำนวนในวงจรรีเลย์ เพื่อปิดหรือเปิด วงจรไฟฟ้าเมื่อทำการนับถึงค่าที่กำหนดอุปกรณ์นับจำนวนจะทำการตรวจนับสัญญาณพัลส์จาก หน่วย อินพุต/เอาต์พุต และ อุปกรณ์ภายใน โดยอุปกรณ์นับจำนวน ที่ใช้ใน PLC จะเป็นแบบ DOWN- COUNTER ที่สามารถตั้งค่าการนับได้ ตั้งแต่ 0-999 จำนวน และมีให้ใช้ได้ 8 ตัว จาก หมายเลข 600-607

อุปกรณ์นับจำนวนที่ใช้ใน PLC มีส่วนประกอบดังนี้

- 1.สัญญาณควบคุม RESET/ENABLE INPUT
- 2.ขาตรวจนับสัญญาณ COUNT INPUT
- 3.เอาต์พุตที่แสดงสถานะของอุปกรณ์นับจำนวน COUNTED-OUT OUTPUT
- 4.หน่วยความจำที่เก็บ ค่าจำนวนนับปัจจุบันของ COUNTER และ ค่าเริ่มต้น (SETPOINT) ที่กำหนดไว้



รูปที่ 3.6 COUNTER ใน PLC

ขาสัญญาต่าง ๆ จะมีผลต่อการทำงานของ COUNTER ดังนี้

1. ขา RESET/ENABLE INPUT ทำหน้าที่กำหนดการทำงานของ COUNTER

- ช่วง ON จะเป็นการ ENABLE COUNTER ให้อยู่ใน COUNTING STATE ขา COUNTED-OUT OUTPUT จะขึ้นอยู่กับค่าปัจจุบันของ COUNTER

- ช่วง OFF จะเป็นการ RESET COUNTER ให้อยู่ใน RESET STATE ขา COUNTED-OUT OUTPUT จะอยู่ในสถานะ OFF

- ช่วงเปลี่ยนจาก OFF เป็น ON ทำให้ค่าปัจจุบันมีค่าเท่ากับค่าที่ตั้งไว้ ขา COUNTED-OUT OUTPUT จะ OFF และ COUNTER เข้าสู่ COUNTING STATE กรณีที่ตั้งค่า SETPOINT ให้เท่ากับ ศูนย์ ขา COUNTED-OUT OUTPUT จะ ON และ COUNTER จะเข้าสู่ COUNTED-OUT STATE

2. ขา COUNT INPUT ในขณะที่ COUNTER อยู่ใน COUNTING STATE COUNTER จะทำการตรวจนับ การเปลี่ยนแปลงของขา COUNT INPUT นี้โดย COUNTER จะทำการลดค่าปัจจุบันลง 1 เมื่อขา COUNT INPUT มีการเปลี่ยนแปลงสถานะจาก OFF เป็น ON

3. ขา COUNTED-OUT OUTPUT จะ ON เมื่อขา RESET/ENABLE INPUT อยู่ในสถานะ ON และ ค่าปัจจุบันของ COUNTER มีค่าเท่ากับ 0 เท่านั้น (COUNTED-OUT STATE)

เมื่อเริ่มจ่ายไฟให้กับระบบ COUNTER จะเข้าสู่สถานะ RESET STATE ขา COUNTED-OUT OUTPUT จะ OFF ค่า SETPOINT และ ค่าปัจจุบันของ COUNTER จะเป็น 0 ทั้งคู่

การอ้างขาค้าง ๆ ของ COUNTER

ขา RESET/ENABLE INPUT อ้างได้ด้วยคำสั่ง CNT XXX , K NNN

เมื่อ NNN = ค่า SETPOINT ของ COUNTER (0-999)

XXX = หมายเลขของ COUNTER (600-607)

ขา COUNT INPUT อ้างได้ด้วยคำสั่ง OUT XXX

ขา COUNTED-OUT OUTPUT อ้างได้จากคำสั่ง LD,LD-NOT,AND XXX

ถ้าการกำหนดค่า SETPOINT ให้กับ COUNTER มากกว่า 1 แห่งในโปรแกรม จะถือเอาค่า SETPOINT ในรังค์สุดท้ายเป็นหลัก

อุปกรณ์	หมายเลขอุปกรณ์	จำนวน
หน่วยอินพุต (OPTO INPUT)	000-007, 010-017	16
หน่วยเอาต์พุต (RELAY OUTPUT)	100-107	8
รีเลย์ช่วย (AUXILIARY RELAY)	200-207, 210-217, 220-227, 230-237 240-247, 250-257, 260-267, 270-277 280-287, 290-297, 300-307, 310-317 320-327, 330-337, 340-347, 350-357 360-367, 370-377, 380-387, 390-397 400-407, 410-417, 420-427, 430-437 440-447, 450-457, 460-467, 470-477 480-487, 490-497	240
อุปกรณ์หน่วงเวลา (TIMER)	500- 507, 510- 517	16
อุปกรณ์นับจำนวน (COUNTER)	600- 607	8

ตารางที่ 3.2 อุปกรณ์ และ หมายเลขอุปกรณ์ใน PLC

คำอธิบายภาษาแลดเดอร์

คำสั่ง	ความหมาย
LD	การอ้างถึงหน้าสัมผัสแบบปกติเปิด (NOMALLY OPENED)
LD-NOT	การอ้างถึงหน้าสัมผัสแบบปกติปิด (NOMALLY CLOSED)
AND	การนำหน้าสัมผัสแบบปกติเปิดมาต่อแบบอนุกรม
AND-NOT	การนำหน้าสัมผัสแบบปกติปิดมาต่อแบบอนุกรม
OR	การนำหน้าสัมผัสแบบปกติเปิดมาต่อแบบขนาน
OR-NOT	การนำหน้าสัมผัสแบบปกติปิดมาต่อแบบขนาน
ANB	การนำเอากลุ่มของหน้าสัมผัสจำนวน 2 กลุ่มมาต่ออนุกรมกัน
ORB	การนำเอากลุ่มของหน้าสัมผัสจำนวน 2 กลุ่มมาต่อขนานกัน
OUT	การส่งสถานะของรังโก้ไปควบคุมอุปกรณ์เอาต์พุต
SET	คำสั่งบังคับให้อุปกรณ์อินพุต เอาต์พุต ON ตลอดเวลา
RES	คำสั่งบังคับให้อุปกรณ์อินพุต เอาต์พุต OFF ตลอดเวลา
TIM	สั่งให้อุปกรณ์หน่วงเวลาเริ่มต้นหน่วงเวลา
CNT	สั่งให้อุปกรณ์นับจำนวนเริ่มทำการนับจำนวน
K	กำหนดค่าคงที่ให้กับอุปกรณ์หน่วงเวลาหรืออุปกรณ์นับจำนวน
END	จบโปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คำสั่งที่ใช้ใน PLC

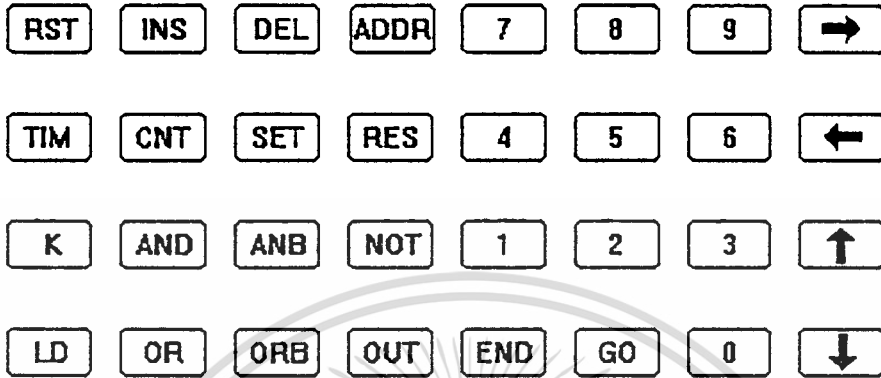
PLC ใช้ภาษาบูลีนซึ่งมีลักษณะ คล้ายสัญลักษณ์ของพีชคณิตบูลีน ในตารางข้างล่าง แสดง คำสั่งภาษาบูลีน และ ภาษาแลคเตอร์

การกดปุ่ม	ภาษาบูลีน	ภาษาแลคเตอร์	หน้าจอ PLC
	LD		LD
	LD-NOT		LD-NOT
	AND		AND
	AND-NOT		AND-NOT
	OR		OR
	OR-NOT		OR-NOT
	ANB		ANB
	ORB		ORB
	OUT		OUT
	SET		SET
	RES		RES
	TIM		TIM
	CNT		CNT
	K		K
	END		END

ตารางที่ 3.3 คำสั่งภาษาบูลีน และ ภาษาแลคเตอร์



3.2.3 หน่วยป้อนข้อมูล และ แสดงผล

การทำงานของ PLC ทำได้โดยการสั่งงานผ่านทางคีย์บอร์ด ซึ่งมีอยู่ทั้งหมด 32 คีย์ ดังรูป



รูปที่ 3.7 คีย์บอร์ดของ PLC

ซึ่งแบ่งออกเป็น 4 กลุ่มดังนี้

1. คีย์ตัวเลข (NUMERIC KEYS) ใช้สำหรับป้อนข้อมูล ที่เป็นตัวเลข เช่น เลขที่บรรทัดของคำสั่ง หรือข้อมูลของคำสั่งนั้น ๆ
2. คีย์คำสั่ง (COMMAND KEYS) ใช้สำหรับป้อนคำสั่งในโปรแกรม มีอยู่ด้วยกัน 13 คีย์ คือ OUT, SET, RES, TIM, CNT, K, AND, ANB, LD, OR, ORB, NOT, และ END
3. คีย์แก้ไข (EDITOR KEYS) มีด้วยกัน 6 คีย์ คือ ADDR, DATA, INS, DEL,  และ  เพื่ออำนวยความสะดวกในการป้อนโปรแกรม
4. คีย์ควบคุม (CONTROL KEYS) ใช้สำหรับควบคุมการทำงานของเครื่อง มีด้วยกัน 3 คีย์ คือ RST, GO, MON

การใช้งานคีย์บอร์ดของ PLC

1. คีย์ RST ใช้สำหรับรีเซ็ต (RESET) ระบบ ในขณะที่เครื่องทำงานผิดพลาดหรือในกรณีที่เราต้องการให้เครื่องหยุดทำงาน ในขณะใดขณะหนึ่ง
2. คีย์ ADDR ใช้สำหรับบอกเครื่องว่า เราจะทำการป้อนโปรแกรมที่บรรทัดที่เท่าใด หรือว่าเราต้องการที่จะอ่านบรรทัดที่เท่าใด
3. คีย์ INS ใช้สำหรับการแทรกบรรทัดในโปรแกรม โดยจะเลื่อนคำสั่งในบรรทัดปัจจุบันไปยังบรรทัดถัดไป และ คำสั่งในบรรทัดถัดไปก็จะถูกเลื่อนลงไปเรื่อยๆ ถ้าหากว่าโปรแกรมมีจำนวนมากกว่าที่เครื่องจะเก็บไว้ได้คำสั่งในบรรทัดสุดท้ายจะถูกลบหายไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คำอธิบาย

1. คำสั่ง LD, LD-NOT เป็นการอ่านสถานะของหน่วยอินพุตแบบปกติและแบบตรงกันข้าม
2. คำสั่ง OUT เป็นการส่งผลลัพธ์ของโปรแกรมแลคเคอร์ ไปควบคุมหน่วยเอาต์พุต
3. ใน PLC จะมีหน่วยอินพุตอยู่ 16 หน่วยอยู่ที่หมายเลข 000-007 และ 010-017 และมีหน่วยเอาต์พุตอยู่ 8 หน่วยอยู่ที่หมายเลข 100-107

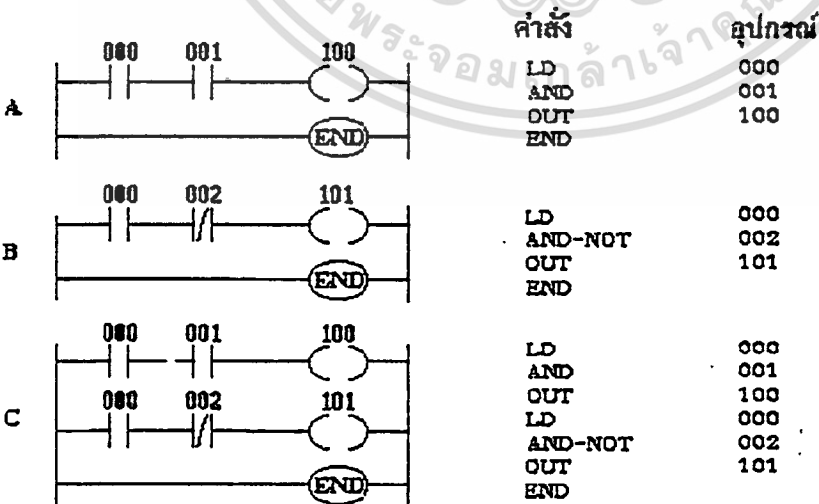
ผลลัพธ์

A. ถ้าหน่วยอินพุตหมายเลข 000 ทำงานหรือ ON (LED ประจำหน่วยอินพุต หมายเลข 000 สว่าง) ทำให้หน่วยเอาต์พุตหมายเลข 100 ทำงานหรือ NO ด้วย (LED ประจำหน่วยเอาต์พุต หมายเลข 100 สว่าง) แต่ถ้าหน่วยอินพุตหมายเลข 000 ไม่ทำงานหรือ OFF (LED ประจำหน่วยอินพุต หมายเลข 000 ดับ) จะทำให้หน่วยเอาต์พุต หมายเลข 100 ไม่ทำงานหรือ OFF ด้วย (LED ประจำหน่วยเอาต์พุตหมายเลข 100 ดับ)

B. ถ้าหน่วยอินพุตหมายเลข 001 ทำงานหรือ ON (LED ประจำหน่วยอินพุตหมายเลข 001 สว่าง) ทำให้หน่วยเอาต์พุตหมายเลข 100 ไม่ทำงานหรือ OFF (LED ประจำหน่วยเอาต์พุต หมายเลข 100 ดับ) แต่ถ้าหน่วยอินพุตหมายเลข 001 ไม่ทำงานหรือ OFF (LED ประจำหน่วยอินพุต หมายเลข 001 ดับ) จะมีผลทำให้หน่วยเอาต์พุตหมายเลข 101 ทำงาน หรือ ON ขึ้นมา (LED ประจำหน่วยเอาต์พุตหมายเลข 101 สว่าง)

C. ถ้าหน่วยอินพุต 000 ON จะทำให้หน่วยเอาต์พุต 100 ON ถ้าหน่วยอินพุต 001 ON จะทำให้หน่วยเอาต์พุต 101 OFF


3.3.2 คำสั่ง AND,




รูปที่ 3.9 คำสั่ง AND, AND-NOT

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. คีย์ DEL ใช้สำหรับลบคำสั่ง ในบรรทัดปัจจุบัน และ เครื่องจะทำการเลื่อนคำสั่งในบรรทัดถัดไปขึ้นมาแทน

5. คีย์  ใช้สำหรับเลื่อนขึ้นไปบรรทัดก่อนหน้า 1 บรรทัด

6. คีย์  ใช้สำหรับเลื่อนลงไปบรรทัดถัดไป 1 บรรทัด

7. คีย์คำสั่ง (COMMAND KEYS) ใช้สำหรับป้อนคำสั่งต่าง ๆ ในขณะที่ทำการป้อนโปรแกรมให้กับเครื่อง ซึ่งแบ่งออกเป็น 3 กลุ่มด้วยกันคือ

7.1 คีย์ คำสั่งที่ทำหน้าที่ แทนคำสั่งต่าง ๆ ในภาษาบูลีน จะประกอบด้วยคีย์ OUT, SET, RES, K, AND, ANB, LD, OR, ORB, และ END

7.2 คีย์ ที่ใช้ร่วมกับคีย์คำสั่งอื่น ๆ เพื่อให้เกิดคำสั่งใหม่ในภาษาบูลีน ซึ่งมีเพียง 1 คีย์ คือ NOT เช่นเมื่อกดร่วมกับคีย์ LD จะเปลี่ยนคำสั่ง LD เป็น LD-NOT แต่ถ้ากดคีย์ NOT อีกครั้ง เครื่องจะเปลี่ยนคำสั่ง LD-NOT ให้กลับมาเป็นคำสั่ง LD อีกครั้ง

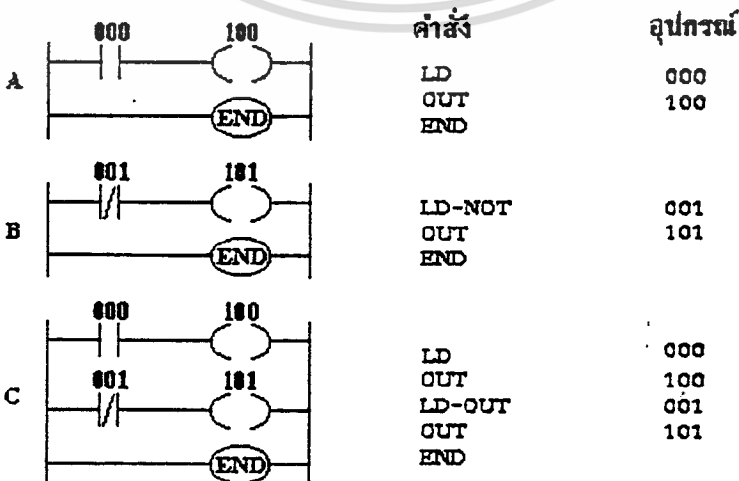
7.3 คีย์ที่ใช้ระบุคำสั่ง และ ชนิดอุปกรณ์ ซึ่งประกอบด้วยคีย์ TIM และ CNT โดยถ้าเรากดคีย์นี้ ในโหมดการรับคำสั่ง จะเป็นการป้อนคำสั่ง TIM และ CNT

8. คีย์ตัวเลข (NUMERIC KEY) ใช้สำหรับป้อนลำดับที่บรรทัดของโปรแกรม หรือข้อมูลที่เป็นตัวเลขทั้งหลายของโปรแกรม

9. คีย์ GO ใช้สำหรับสั่งให้เครื่องทำงานตามโปรแกรมบูลีนที่เราเขียนไว้

3.3 ตัวอย่างการเขียนโปรแกรม PLC

3.3.1 คำสั่ง LD,LD-NOT,OUT



คำอธิบาย

คำสั่ง AND และ AND-NOT เป็นการนำเอาสถานะของหน่วย อินพุต/เอาต์พุต หรือ อุปกรณ์ภายในมากกว่า 1 ตัวมาอนุกรมกัน

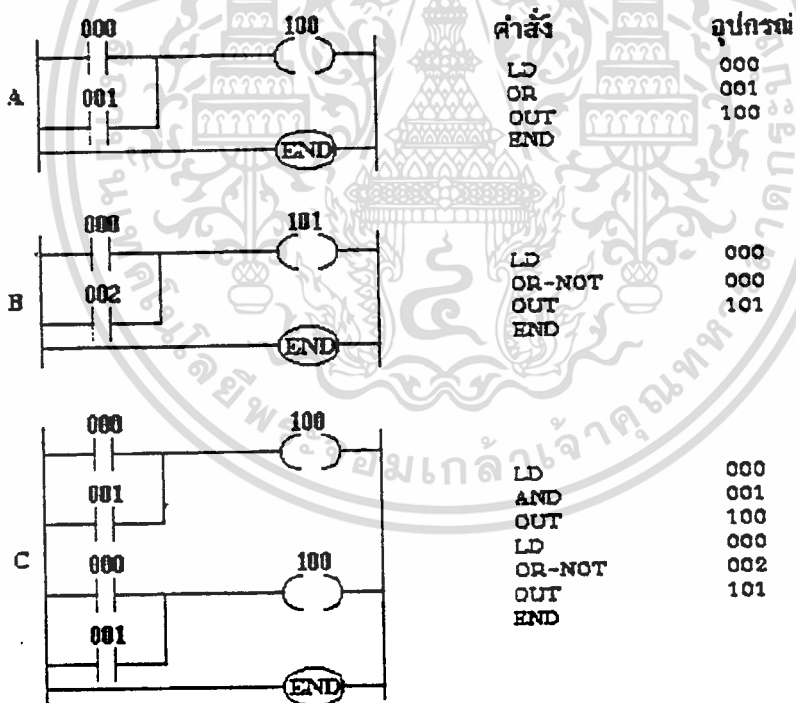
ผลลัพธ์

A. เมื่อหน่วยอินพุต 000 และ 001 ON ทั้งคู่ หน่วยเอาต์พุต 100 จะทำงาน หรือ ON ด้วย แต่ ถ้าหน่วยอินพุต 000 และ 001 มีตัวใดตัวหนึ่งไม่ทำงาน (OFF) หรือ ไม่ทำงานทั้งคู่ จะเป็นผลทำให้ หน่วยเอาต์พุต 100 ไม่ทำงาน หรือ (OFF) ทันที

B. หน่วยเอาต์พุตหมายเลข 101 จะทำงานหรือ ON กรณีเดียวคือเมื่อหน่วยอินพุต 000 ON และ หน่วยอินพุต 002 OFF เท่านั้น

C. เป็นการนำเอาผลลัพธ์ A และ B รวมกัน

3.3.3 คำสั่ง OR, OR-NOT



รูปที่ 3.10 คำสั่ง OR, OR-NOT

คำอธิบาย

คำสั่ง OR และ OR-NOT เป็นการนำเอาสถานะ ของหน่วยอินพุต/เอาต์พุต หรือ อุปกรณ์

เอกสารภายในมากกว่า 1 ตัวมาต่อขนานกันใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

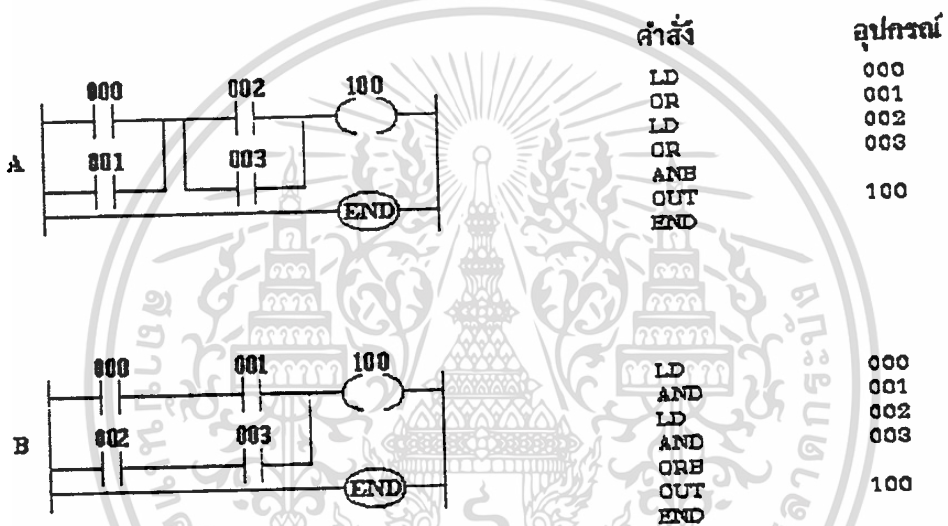
ผลลัพธ์

A. เมื่อหน่วยอินพุต 000 และ 001 ตัวใดตัวหนึ่ง ON หน่วยเอาต์พุต 100 จะทำงาน หรือ ON ทันที

B. ถ้าหน่วยอินพุตหมายเลข 000 ทำงาน (ON) หรือ หน่วยอินพุต 002 ไม่ทำงาน (OFF) จะทำให้หน่วยเอาต์พุต 101 ทำงาน (ON)

C. เป็นการนำเอาผลลัพธ์ A และ B มารวมกัน

3.3.4 คำสั่ง AND, ORB



รูปที่ 3.11 คำสั่ง AND, ORB

คำอธิบาย

1. คำสั่ง ANB เป็นการนำกลุ่ม (BLOCK) ของอุปกรณ์ 2 กลุ่มมาต่ออนุกรมกัน ซึ่งใน PLC สามารถนำกลุ่มของอุปกรณ์ มาต่ออนุกรมกันได้สูงสุดจำนวน 10 กลุ่ม ต่อโปรแกรม 1 รังค์
2. คำสั่ง ORB เป็นการนำกลุ่ม (BLOCK) ของอุปกรณ์ 2 กลุ่มมาต่อขนานกัน ซึ่งใน PLC เราสามารถนำกลุ่มของอุปกรณ์ มาต่อขนานกันได้สูงสุดจำนวน 10 กลุ่ม ต่อโปรแกรม 1 รังค์
3. เราสามารถใช้คำสั่ง LD, LD-NOT, AND, AND-NOT, OR, OR-NOT มาใช้ร่วมกับ คำสั่ง ANB หรือ ORB ได้

ผลลัพธ์

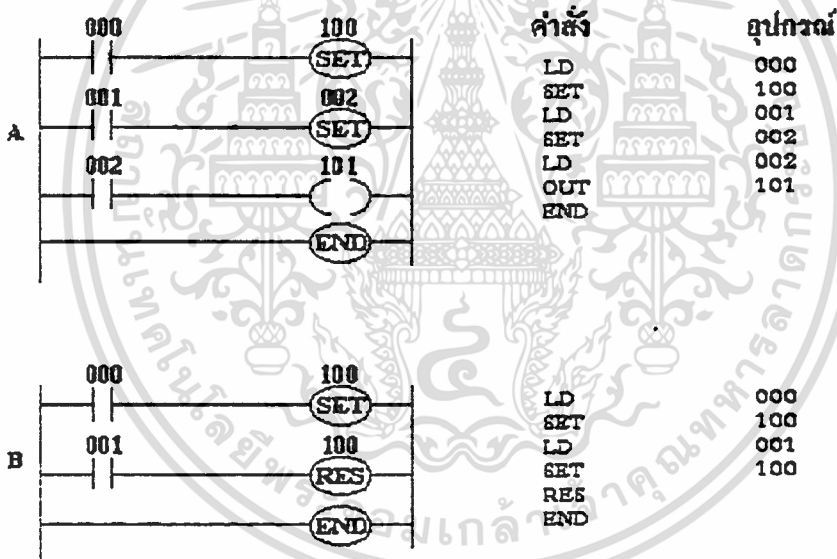
A. ถ้า (หน่วยอินพุต 000 หรือ 001 ON อย่างน้อย 1 ตัว) และ (หน่วยอินพุต 002 หรือ 003 ON อย่างน้อย 1 ตัว) จะทำให้หน่วยเอาต์พุต 100 ON ไม่เช่นนั้นหน่วยเอาต์พุต 100 จะ OFF ตามสมการพีชคณิตข้างล่าง

$$100 = (000 \text{ OR } 001) \text{ AND } (002 \text{ OR } 003)$$

B. ถ้า (หน่วยอินพุต 000 และ 001 ON ทั้งคู่) หรือ (หน่วยอินพุต 002 และ 003 ON ทั้งคู่) จะทำให้หน่วยเอาต์พุต 100 ทำงาน (ON) ไม่เช่นนั้นหน่วยเอาต์พุต 100 จะ OFF ตามสมการข้างล่าง

$$100 = (000 \text{ AND } 001) \text{ OR } (002 \text{ AND } 003)$$

3.3.5 คำสั่ง SET และ RES



รูปที่ 3.12 คำสั่ง SET, RES

คำอธิบาย

1. คำสั่ง SET เป็นการทำให้ หน่วยอินพุต หรือ หน่วยเอาต์พุต มีสถานะเป็นจริงหรือ ON ตลอดเวลา โดยจะไม่ขึ้นกับสถานะจริงๆ ของหน่วยอินพุต หรือ ผลลัพธ์ของวงกบที่ควบคุม หน่วยเอาต์พุตนั้นเลย

2. คำสั่ง RES เป็นการทำให้หน่วยอินพุต หรือ หน่วยเอาต์พุต มีสถานะการทำงาน เป็นเท็จหรือ OFF ตลอดเวลา โดยไม่ขึ้นกับสถานะจริงๆ ของหน่วยอินพุตหรือ ผลลัพธ์ของวงกบที่

3. คำสั่ง SET และ RES จะใช้ได้เฉพาะกับ หน่วยอินพุต/เอาต์พุตเท่านั้นโดยไม่สามารถนำไปใช้กับ อุปกรณ์รีเลย์ช่วย (Auxiliary Relay) อุปกรณ์หน่วงเวลา (Timer) หรืออุปกรณ์นับจำนวน (Counter)

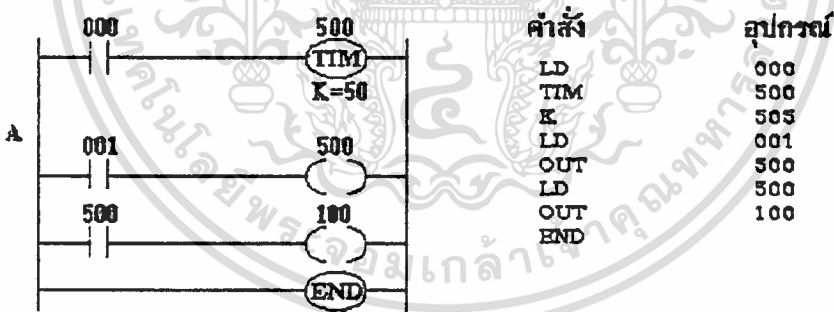
4. ในกรณีที่มียังคำสั่ง SET และ RES อยู่ในโปรแกรมเดียวกัน จะถือเอาคำสั่งรังค์สุดท้ายเป็นหลัก

ผลลัพท์

A. เมื่อหน่วยอินพุต 000 ON จะทำให้หน่วยเอาต์พุต 100 ON จากนั้นหน่วย เอาต์พุต 002 จะ ON ค้างตลอดเวลา โดยไม่ขึ้นกับหน่วยอินพุต 000 อีกในช่วงแรกหน่วยเอาต์พุต 101 จะ ON/OFF ตามหน่วยอินพุต 002 แต่เมื่อหน่วยอินพุต 001 ON ทำให้หน่วยอินพุต 002 ถูกมองเหมือนกับ ON ตลอดเวลา เป็นผลให้หน่วยเอาต์พุต 100 ON ตลอด โดยไม่ขึ้นกับสถานะจริง ๆ ของหน่วยอินพุต 002 อีก (LED ประจำหน่วยอินพุต 002 จะยังคงสว่าง/ดับตามสถานะจริง ๆ ของสวิทช์ภายนอก)

B. หน่วยอินพุต 000 ON จะทำให้หน่วยเอาต์พุต 100 ON แต่เมื่อหน่วยอินพุต 001 ON จะทำให้หน่วยเอาต์พุต 100 OFF

3.3.6 คำสั่ง TIM และ K



รูปที่ 3.13 คำสั่ง TIM, K

คำอธิบาย

1. อุปกรณ์หน่วงเวลา (TIMER) จะเริ่มต้นหน่วงเวลาเมื่อหน่วยอินพุต 000 ON จะยกเลิกการหน่วงเวลาเมื่อหน่วยอินพุต 000 เป็น OFF และ จะเริ่มต้นหน่วงเวลาใหม่ เมื่อหน่วยอินพุต 000 กลับมา ON อีกครั้ง

2. หน่วยอินพุต 001 เป็นตัวควบคุมว่าจะให้อุปกรณ์หน่วงเวลาทำการลดเวลาลงเรื่อยๆ

หรือหยุดการลดเวลาลงชั่วคราวโดยถ้าหน่วยอินพุต 001 ON จะทำให้เวลาที่กำหนดไว้ลดลงแต่

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ถ้าหน่วยอินพุต 001 เป็น OFF อุปกรณ์หน่วยเวลา จะหยุดการลดค่าเวลาลงชั่วคราว และจะเริ่มลดค่าเวลาต่อเมื่อหน่วยอินพุต 001 กลับมา ON อีกครั้ง

3. คำสั่ง K ใช้ในการกำหนดค่าเวลาเริ่มต้น ให้กับอุปกรณ์หน่วยเวลาโดยจะต้องอยู่ต่อจากคำสั่ง TIM เสมอ และ ค่าเวลาที่กำหนด จะต้องมีหน่วยเป็น 0.1 วินาที ดังนั้นจากตัวอย่าง K = 050 หมายถึง ทำการหน่วยเวลา 5 วินาที

4. มี TIMER ให้ใช้ได้ 16 ตัวอยู่ที่หมายเลข 500-507 และ 510-517

5. การป้อนค่า 500 ทำได้โดยการกดปุ่ม

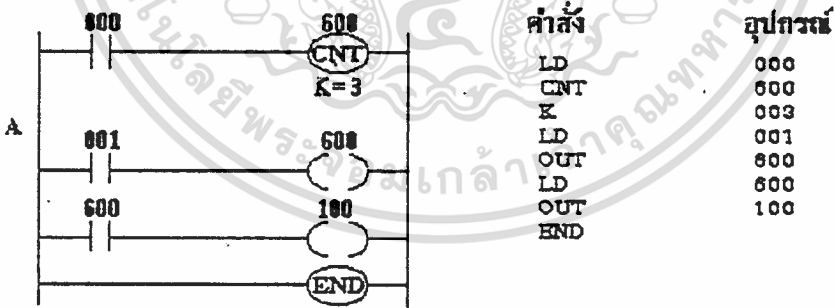
TIM 5 0 0

ผลลัพธ์

1. เมื่อหน่วยอินพุต 000 ON โปรแกรมจะอ่านว่า K = 50 ไปใส่ให้กับ อุปกรณ์หน่วยเวลา หมายเลข 500 แต่อุปกรณ์หน่วยเวลา จะยังคงไม่ทำการหน่วยเวลานั้นที่ ต่อเมื่อหน่วยอินพุต 001 ON อุปกรณ์หน่วยเวลาจะเริ่มหน่วยเวลา และ เมื่อครบ 5 วินาทีหน่วยเอาต์พุต 100 จะ ON ขึ้นมา

2. ในขณะที่อุปกรณ์หน่วยเวลาทำการหน่วยเวลานั้น ถ้าเรา OFF หน่วยอินพุต 001 อุปกรณ์หน่วยเวลาจะหยุดการหน่วยเวลาไว้ชั่วคราว และ จะเริ่มหน่วยเวลาต่อ เมื่อหน่วยอินพุต 001 กลับมา ON อีกครั้ง

3.3.7 คำสั่ง CNT



รูปที่ 3.14 คำสั่ง CNT

คำอธิบาย

1. อุปกรณ์นับจำนวน (COUNTER) จะทำการนับจำนวน การเปลี่ยนแปลงสถานะของหน่วยอินพุตตามจำนวนที่ตั้งไว้

2. หน่วยอินพุต 000 เป็นการควบคุมการทำงานของ อุปกรณ์นับจำนวน โดย อุปกรณ์นับจำนวนจะนับต่อไปเรื่อย ๆ เมื่อหน่วยอินพุต 000 อยู่ในสถานะ ON และเมื่อหน่วยอินพุต 000

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เปลี่ยน เป็น OFF อุปกรณ์นับจำนวน จะยกเลิกการนับและจะเริ่มต้นนับใหม่ เมื่อหน่วยอินพุต 000 เปลี่ยน กลับมา ON อีกครั้ง

3. คำสั่ง K เป็นคำสั่งที่ใช้ตั้งค่าจำนวนนับของอุปกรณ์นับจำนวน

4. อุปกรณ์นับจำนวน จะทำการนับการเปลี่ยนแปลง ของหน่วยอินพุต 001 โดยจะทำการนับทุกครั้งที่หน่วยอินพุต 001 มีการเปลี่ยนแปลงสถานะจาก OFF เป็น ON

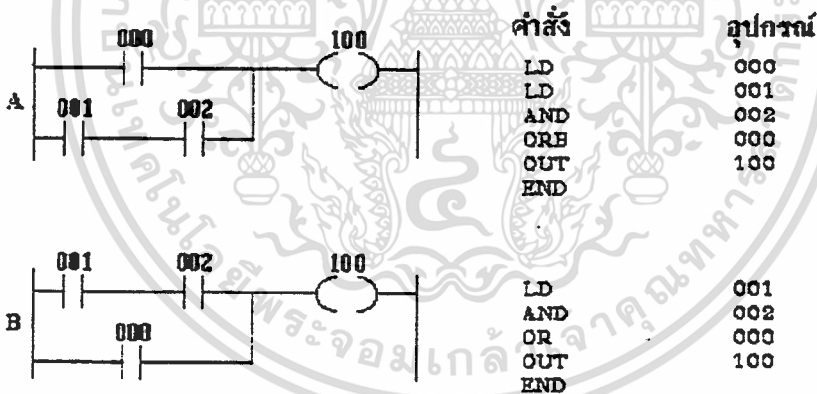
5. มีอุปกรณ์นับจำนวนให้ใช้ได้ 8 ตัว อยู่ที่หมายเลข 600-607

ผลลัพธ์

เมื่อหน่วยอินพุต 000 ON อุปกรณ์นับจำนวน จะเริ่มทำงาน โดยทำการเปลี่ยนสถานะจาก OFF เป็น ON ของหน่วยอินพุต 001 เมื่อครบ 3 ครั้ง อุปกรณ์นับจำนวนจะ ON และ ทำให้หน่วยเอาต์พุต 100 ON

3.4 เทคนิคการเขียนโปรแกรม

3.4.1 การยวบวงจรให้ง่าย (1)

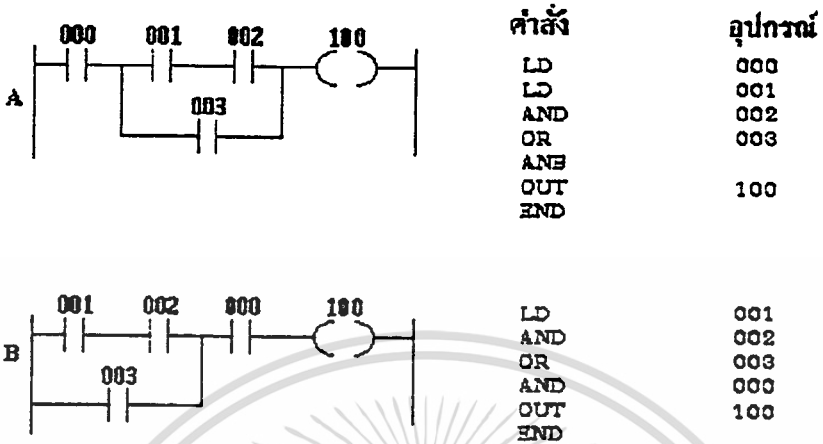


รูปที่ 3.15 การยวบวงจรให้ง่าย (1)

คำอธิบาย

วงจร A และ วงจร B มีการทำงานเหมือนกัน แต่วงจร B ดูง่ายกว่า และสามารถลดคำสั่งลงได้ 1 คำสั่ง

3.4.2 การยวบวงจรให้ง่าย (2)

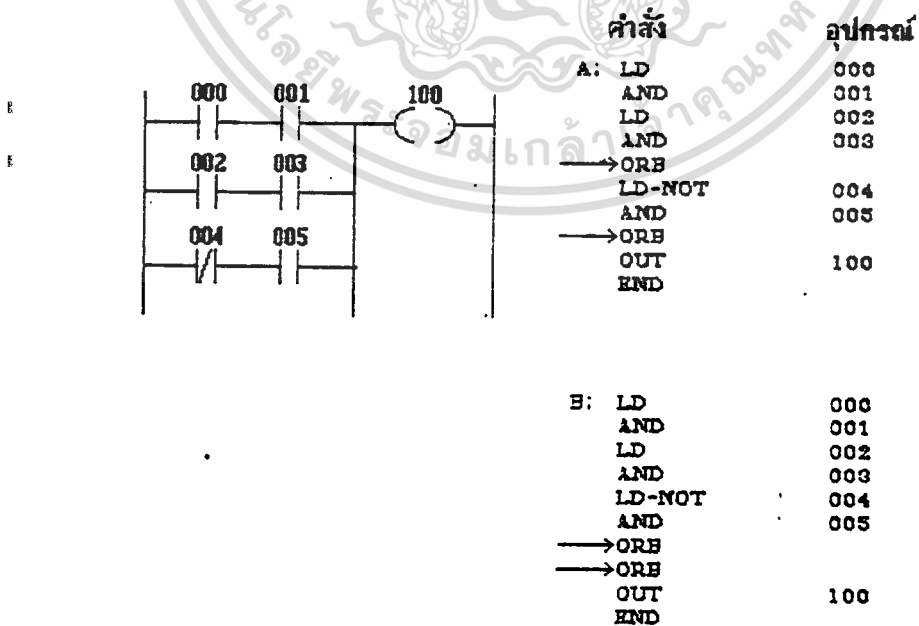


รูปที่ 3.16 การยวบวงจรให้ง่าย (2)

คำอธิบาย

1. วงจร A และวงจร B มีการทำงานเหมือนกัน แต่วงจร B ง่ายกว่า
2. วงจร B สามารถลดได้ 1 คำสั่ง
3. วงจร A ต้องใช้คำสั่ง ANB แต่วงจร B ใช้คำสั่ง AND ธรรมดา

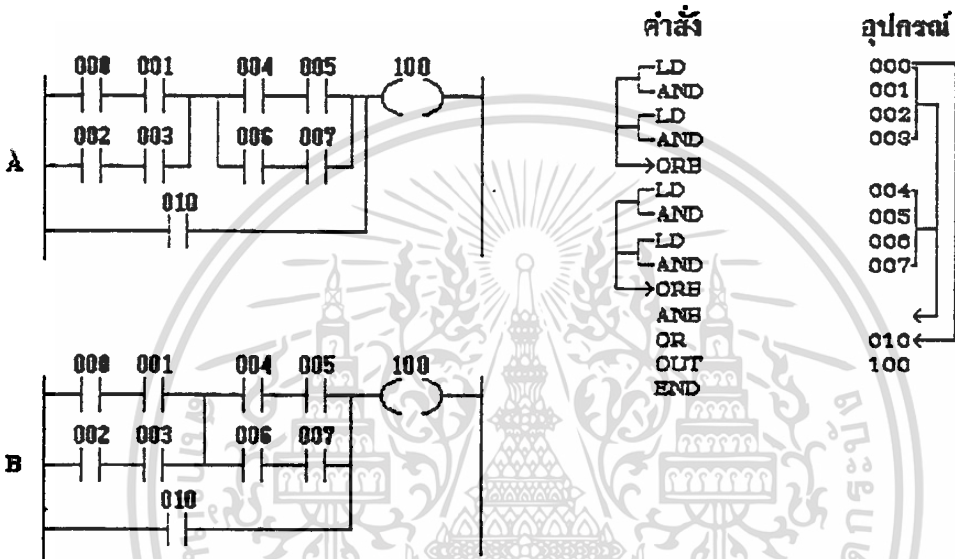
3.4.3 เทคนิคการเขียนภาษาบูลีน



คำอธิบาย

1. ถ้าริงก์มีมากกว่า 2 ริงก์มาขนานกันควรยุบกลุ่มของริงก์ให้เหลืออย่างน้อยที่สุดด้วยคำสั่ง ORB
2. การเขียนโปรแกรมในข้อ A เป็นการเขียนโปรแกรมที่ดีข้อ B เป็นการเขียนโปรแกรมที่ไม่ดี

3.4.4 เทคนิคการเขียนโปรแกรม



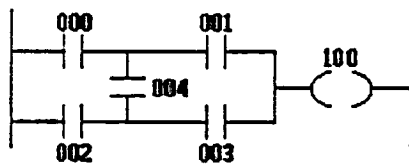
รูปที่ 3.18 เทคนิคการเขียนโปรแกรม

คำอธิบาย

1. วงจร B เป็นการเขียนวงจรที่ดีกว่าวงจร A

3.5 วงจรที่ควรหลีกเลี่ยงในการเขียนโปรแกรม PLC

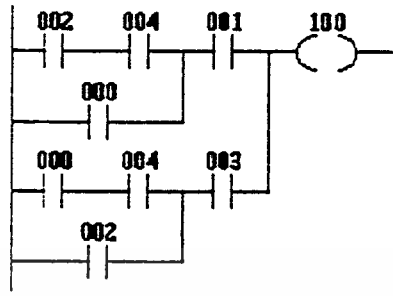
3.5.1 วงจรที่เป็นสะพานเชื่อม



รูปที่ 3.19 วงจรที่เป็นสะพานเชื่อม

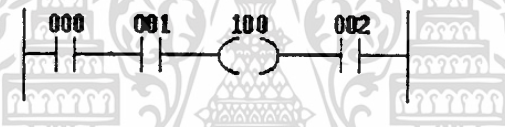
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ซึ่งเราไม่สามารถแปลงเป็นโปรแกรมภาษาบูลีนได้เราต้องแปลงวงจรนี้ใหม่ดังรูปข้างล่าง



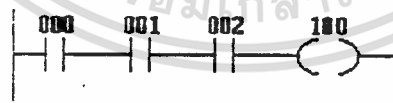
รูปที่ 3.20 การแปลงวงจรที่เป็นสะพานเชื่อม

3.5.2..วงจรที่เอาท์พุทไม่อยู่ทางขวามือสุด



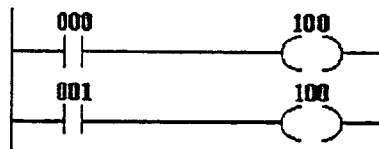
รูปที่ 3.21 วงจรที่เอาท์พุทไม่อยู่ทางขวามือสุด

วงจรแลคเคอร์รี่นี้ สามารถที่จะทำงานได้ แต่ไม่สามารถแปลงเป็นโปรแกรมภาษาบูลีนได้ เราต้องแปลงวงจรนี้ใหม่ดังรูปข้างล่าง



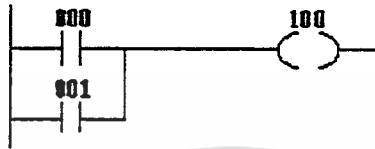
รูปที่ 3.22 การแปลงวงจรที่เอาท์พุทไม่อยู่ทางขวามือ

3.5.3 วงจรที่อุปกรณ์เอาท์พุทใช้มากกว่า 1 แห่งในโปรแกรม



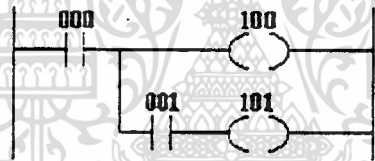
รูปที่ 3.23 วงจรที่อุปกรณ์เอาท์พุทใช้มากกว่า 1 แห่งในโปรแกรม

วงจรข้างบนนี้อุปกรณ์เอาต์พุต 100 จะขึ้นอยู่กับอุปกรณ์อินพุต 001 เพียงอย่างเดียวเพราะ PLC จะทำการสแกนอุปกรณ์อินพุต/เอาต์พุตต่อเมื่อทำงานตามโปรแกรม ครบรอบแล้วเท่านั้น นั่นคือสถานะของ 001 จะเขียนทับสถานะของ 000 เพื่อหลีกเลี่ยง ปัญหาดังกล่าว เราต้องเขียนโปรแกรมแลคเคอร์ใหม่ดังนี้

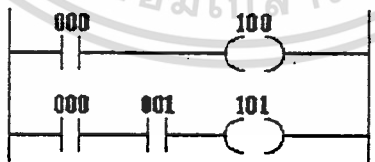


รูปที่ 3.24 การแปลงวงจรอุปกรณ์เอาต์พุตใช้มากกว่า 1 แห่งในโปรแกรม

3.5.4 การควบคุมอุปกรณ์เอาต์พุต ที่เป็นเงื่อนไขมากกว่า 1 เงื่อนไขใน 1 ริงก์



รูปที่ 3.25 วงจรควบคุมเอาต์พุตที่เป็นเงื่อนไขมากกว่า 1 เงื่อนไข
วงจรข้างบนควรแก้ไขใหม่ดังรูปข้างล่าง



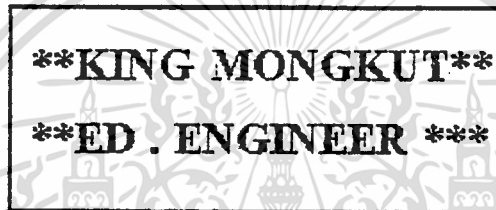
รูปที่ 3.26 การแปลงวงจรควบคุมเอาต์พุตที่เป็นเงื่อนไขมากกว่า 1 เงื่อนไข

บทที่ 4

การทดลองและผลการทดลอง

4.1 การทดลองการทำงานของโปรแกรมสแกนคีย์และการแสดงผลรวมทั้งทดลองปุ่ม FUNCTION พิเศษต่างๆ

4.1.1 การทดลองการใช้คีย์ที่ใช้ในการเขียนโปรแกรมและการแสดงผล
เมื่อเริ่มจ่ายไฟให้เครื่อง PLC จอภาพ LCD จะแสดงคิงรูป



รูปที่ 4.1 หน้าจอ LCD เมื่อเริ่มจ่ายไฟ

เมื่อกดปุ่ม ADDR จะเข้าสู่การป้อนโปรแกรม LADDER โดยจะแสดงหมายเลขบรรทัด
เริ่มต้นคิงรูป



รูปที่ 4.2 หน้าจอ LCD เมื่อกดปุ่ม ADD

ทดลองการรับคีย์บอร์ดที่ใช้ในการเขียนโปรแกรม LADDER โดยเขียนโปรแกรมดังนี้

คำสั่ง	อุปกรณ์
LD	000
OUT	100

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คำสั่ง	อุปกรณ์
TIM	500
CNT	600
K	234
OR	000
AND	007
ORB	
ANB	
SET	000
RES	001
END	

ที่หน้าจอ LCD จะแสดงผลตรงตามรูปแบบโปรแกรมที่ป้อนให้ โดยที่คำสั่ง END จะมีหมายเลขบรรทัด 011



011 RES 001
012 END

รูปที่ 4.3 หน้าจอเมื่อสิ้นสุดการเขียนโปรแกรม

4.1.2 การกดปุ่ม FUNCTION


การกดปุ่ม ADDR

ต่อจากการทดลองที่แล้ว เมื่อกดปุ่ม ADDR จะเป็นการเข้าสู่การป้อนหมายเลขบรรทัดที่ต้องการกระโดดไป โดยป้อนหมายเลขบรรทัดเป็น 006 เมื่อป้อนแล้วที่หน้าจอจะแสดงผลดังรูป

-ADDRESS NUMBER?

006

รูปที่ 4.4 การทดลองปุ่ม ADDR

และเมื่อกดปุ่ม  จะทำให้หน้าจอ LCD แสดงผลดังรูป

005 K # 234
006 OR 000

รูปที่ 4.5 หน้าจอเมื่อกดปุ่ม 

การทดลองปุ่ม FUNCTION INS,DEL

จากหัวข้อที่แล้วเมื่อกดปุ่ม INS จะทำให้โปรแกรมเลื่อนไป 1 บรรทัด และบรรทัดที่ 006 จะว่างไว้แสดงผลดังรูป

005 K # 234
006

รูปที่ 4.6 แสดงการกดปุ่ม INS

และเมื่อกดปุ่ม DEL ทำให้โปรแกรมที่ถูกเลื่อนไปถอยกลับมา 1 บรรทัดดั้งเดิมดังแสดงในรูป

005	K	#	234
006	OR		000

รูปที่ 4.7 แสดงการกดปุ่ม DEL

การทดลองปุ่ม FUNCTION , , ,  จากหัวข้อที่แล้วเมื่อเรากดปุ่ม  1 ครั้ง
จะได้ผลดังรูป

006	OR	000
007	AND	007


รูปที่ 4.8 แสดงการกดปุ่ม 

และเมื่อกดปุ่ม  1 ครั้งจะได้ผลดังรูป

005	K	#	234
006	OR		000

รูปที่ 4.9 แสดงการกดปุ่ม 

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ทดลองการกดปุ่ม  เคอร์เซอร์จะเลื่อนไปทางขวาครั้งละ 1 ตำแหน่ง

ทดลองการกดปุ่ม  เคอร์เซอร์จะเลื่อนกลับมาทางซ้ายครั้งละ 1 ตำแหน่ง

สรุปผลการทดลอง

จากผลที่ได้แสดงว่าปุ่มคำสั่งและปุ่มตัวเลขที่ใช้ในการเขียนโปรแกรมแลคเตอร์ทำงานได้ถูกต้อง และ การแสดงผลตรงตามโปรแกรม การทดลองปุ่ม FUNCTION ทุกปุ่มทำงานตรงตามหน้าที่ได้ถูกต้อง

4.2 การทดลองการทำงานของโปรแกรมย่อย

ในหัวข้อนี้จะเป็นการกล่าวถึงการทดลองการทำงานของโปรแกรมย่อยทั้งหมดว่าผลการทำงานเป็นไปตามที่ได้ออกแบบเอาไว้หรือไม่โดยการทดลองเขียนเป็นโปรแกรมแลคเตอร์สั้น ๆ เพื่อตรวจสอบว่าผลการทำงานถูกต้องหรือไม่ซึ่งสามารถแบ่งออกเป็นการทดลองในส่วนของแต่ละโปรแกรมดังนี้

4.2.1 การทดลองโปรแกรมย่อย LD

โปรแกรมย่อย LD สามารถกระทำการนำค่าสถานะของอินพุต เอ้าท์พุท AUX TIMER COUNTER เข้ามาเก็บไว้ในหน่วยความจำ เพื่อนำค่าสถานะไปใช้ต่อไป การทดลองจึงจะทำการ LD ในทุกส่วนและนำค่าสถานะส่งออกที่ส่วนเอ้าท์พุทซึ่งมี LED แสดง สถานะ 1 LED จะสว่าง สถานะ 0 LED คับ โดยเขียนโปรแกรมทดลองได้ดังนี้

	คำสั่ง	อุปกรณ์
	LD	000
	OUT	100
	LD	100
	OUT	101
	LD	200
	OUT	102
	LD	500
	OUT	103
	LD	600
	OUT	104
	LD	END
	OUT	104
	END	

รูปที่ 4.10 โปรแกรมทดลองโปรแกรมย่อย LD

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คำอธิบาย

1. คำสั่ง LD เป็นการอ่านค่าสถานะของหน่วยต่าง ๆ โดยแยกตามค่าอุปกรณ์
2. คำสั่ง OUT เป็นการส่งผลลัพธ์ของโปรแกรมแลคเคอร์ไปควบคุมส่วนอุปกรณ์
3. อุปกรณ์หมายเลข 000 เป็นหน่วยอินพุท
 อุปกรณ์หมายเลข 100 เป็นหน่วยเอาต์พุท
 อุปกรณ์หมายเลข 200 เป็นหน่วย AUX RELAY
 อุปกรณ์หมายเลข 500 เป็นหน่วย TIMER
 อุปกรณ์หมายเลข 600 เป็นหน่วย CNT

การทดลองโปรแกรมย่อย LD นี้จะเลือกส่วนอุปกรณ์ทุกชนิดทดลองเพียงอย่างละ 1 หน่วย ผลการทดลอง

เมื่อรันโปรแกรม LED ทุกดวงจะดับ เพราะอินพุทเบอร์ 000 ไม่ได้รับ INPUT และ OUTPUT หมายเลข 100 อยู่ในสถานะ OFF, AUX RELAY เบอร์ 200 ก็จะมีสถานะเป็น OFF เนื่องจากการเคลียร์ค่า AUX RELAY ทุกตัวให้ OFF ตอนแรกของโปรแกรมหลัก TIMER หมายเลข 500 จะมีสถานะ OFF เพราะยังไม่ได้ถูกเรียกใช้งาน COUNTER หมายเลข 600 ก็เป็นเช่นเดียวกับ TIMER

เมื่อทดลองให้อินพุทที่อินพุทหมายเลข 000 เป็นลอจิก 1 จะเป็นผลทำให้ LED ของเอาต์พุท หมายเลข 100 ติด และ จากการ LD ค่าสถานะของเอาต์พุทหมายเลข 100 มา OUTPUT ออกที่เอาต์พุท หมายเลข 101 ก็จะทำให้ LED ของเอาต์พุทหมายเลข 101 ติดด้วย

แสดงถึงว่าโปรแกรมย่อย LD สามารถทำการนำค่าสถานะ ของอุปกรณ์มาใช้ในโปรแกรม ได้อย่างถูกต้อง

4.2.2 การทดลองโปรแกรมย่อย LD-NOT

โปรแกรมย่อย LD-NOT เป็นการอ้างถึงสถานะของอุปกรณ์ในสถานะตรงข้ามกับสถานะจริง (สถานะแบบตรงกันข้าม) โดยเขียนโปรแกรมทดลองดังนี้

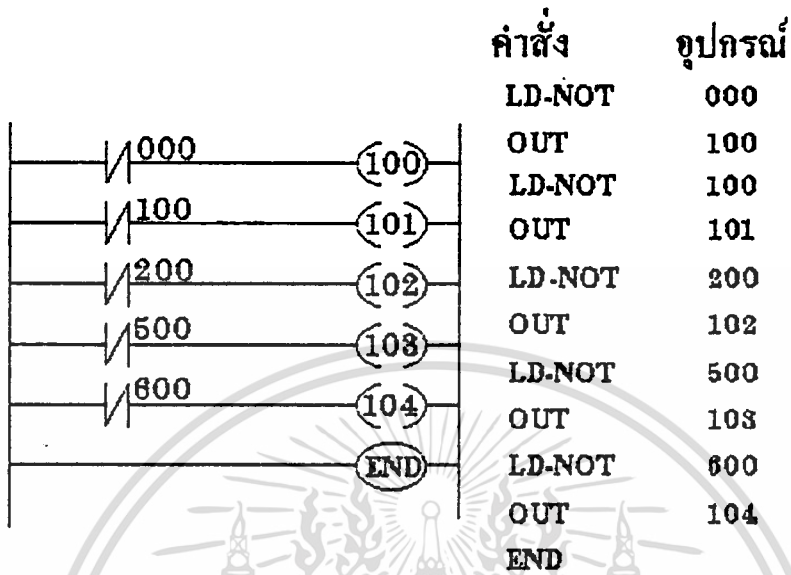
ผลการทดลอง

เมื่อรันโปรแกรม LED ประจำตำแหน่งเอาต์พุทหมายเลข 100,102,103,104 จะติดสว่าง เพราะสถานะจริงของอุปกรณ์ทุกตัวที่ทำการ LD-OUT เป็น 0 แต่ถูกกลับสถานะเป็นตรงกันข้าม จึงได้สถานะเป็น 1 แต่เอาต์พุทหมายเลข 101 จะดับ

เมื่อให้อินพุทลอจิก 1 แต่อุปกรณ์อินพุทหมายเลข 000 โปรแกรมย่อย LD จะกลับสถานะ ให้เป็น 0 ทำให้ส่งภาวะ 0 ให้เอาต์พุทหมายเลข 100 LED ก็จะดับ และเมื่อ LD-NOT อุปกรณ์ หมายเลข 100 ส่งให้ OUTPUT หมายเลข 101 LED หมายเลข 101 จึงจะสว่าง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

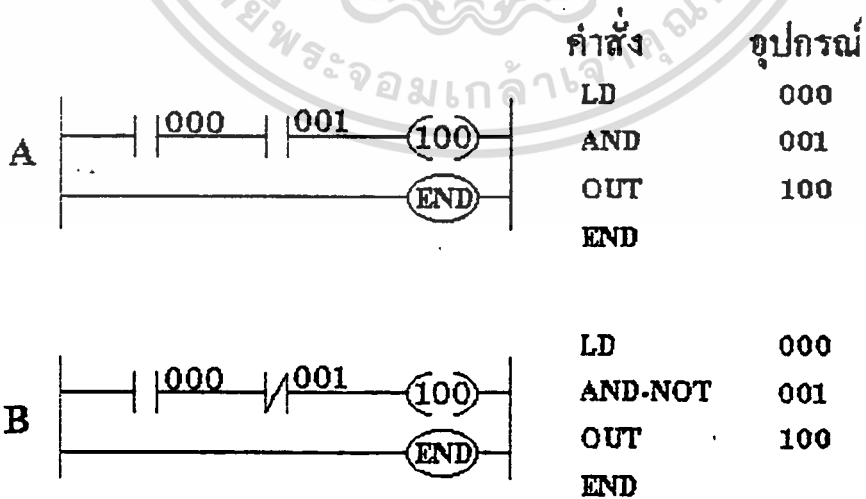
แสดงว่าโปรแกรมย่อย LD-NOT ทำการกลับสภาวะจริงให้เป็นตรงกันข้ามได้อย่างถูกต้อง



รูปที่ 4.11 โปรแกรมทดลองโปรแกรมย่อย LD-NOT

4.2.3 การทดลองโปรแกรมย่อย AND และ AND-NOT

โปรแกรมย่อย AND และ AND-NOT เป็นการนำเอาค่าสภาวะของอุปกรณ์ 2 ตัว มาต่ออนุกรมกัน โดยเขียนโปรแกรมทดลองดังนี้



รูปที่ 4.12 โปรแกรมทดลองโปรแกรมย่อย AND,AND-NOT

ผลการทดลอง

โปรแกรม A เป็นการทดลองโปรแกรมย่อย AND เมื่อรันโปรแกรมหากไม่ให้อินพุตแก่ อุปกรณ์ อินพุตหมายเลข 000 และ 001 หรือให้อินพุตลอจิก 1 แก่ อุปกรณ์ตัวใดตัวหนึ่งเท่านั้น อุปกรณ์เอาต์พุต หมายเลขก็ยังคง OFF

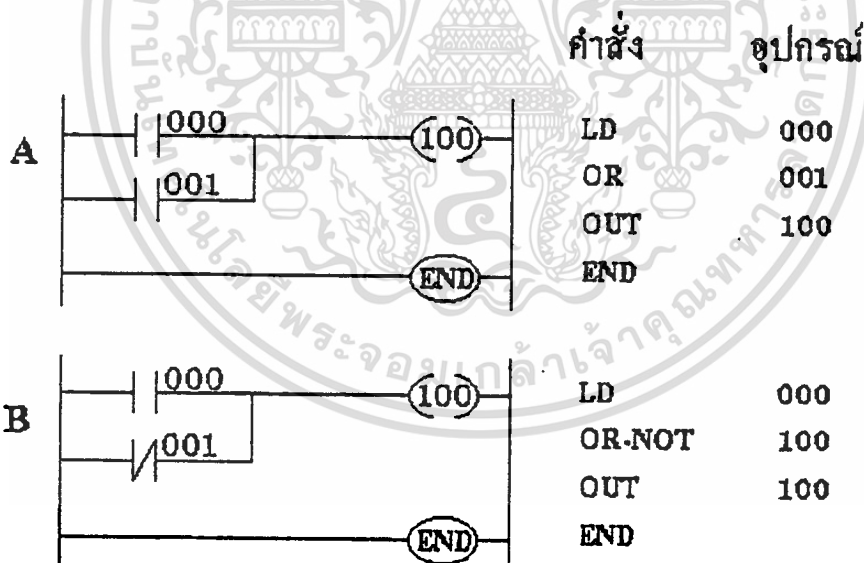
เมื่อให้อินพุตแก่ อุปกรณ์หมายเลข 000 และ 001 เป็น 1 ทั้งคู่ อุปกรณ์ เอาต์พุต หมายเลข 100 จึงจะ ON ทำให้ LED ประจำเอาต์พุตติดสว่าง

โปรแกรม B ทดลองโปรแกรมย่อย AND-NOT เมื่อรันโปรแกรมแล้วให้ลอจิก 1 แก่ อุปกรณ์อินพุต หมายเลข 000 ทำให้ LED ประจำเอาต์พุต 100 ติดสว่าง นอกจากการให้อุปกรณ์ อินพุตเป็นแบบนี้แล้ว กรณีอื่นจะให้ อุปกรณ์เอาต์พุตหมายเลข 100 เป็น OFF ทั้งหมด

แสดงถึงว่าการทำงานของ การ AND และ AND-NOT ทำงานได้ถูกต้อง

4.2.4 การทดลองโปรแกรมย่อย OR และ OR-NOT

โปรแกรมย่อย OR และ OR-NOT เป็นการนำค่าสถานะของอุปกรณ์ 2 ตัวมาต่อขนานกัน โดยเขียนโปรแกรมทดลองได้ดังนี้



รูปที่ 4.13 โปรแกรมทดลองโปรแกรมย่อย OR,OR-NOT

ผลการทดลอง

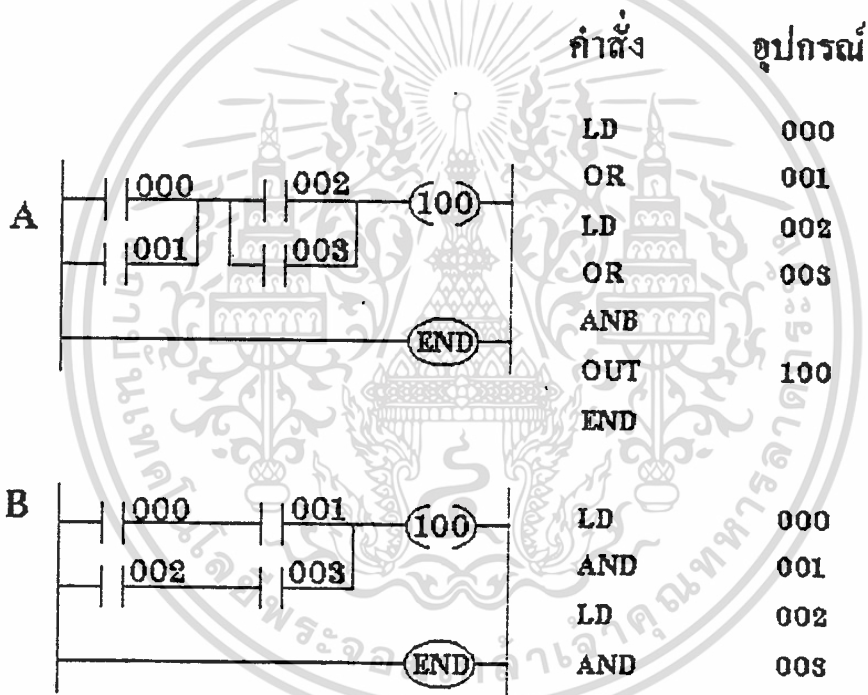
โปรแกรม A เป็น การทดลองโปรแกรมย่อย OR เมื่อรันโปรแกรม หากให้สถานะ 0 แก่ อินพุต หมายเลข 000 และ 001 จะทำให้ LED แสดงสถานะเอาต์พุตหมายเลข 100 ดับ นอกจากนี้ ในกรณีป้อนอินพุตแบบอื่น จะทำให้ LED ติดสว่าง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรม B เป็นการทดลองโปรแกรมย่อย OR-NOT เมื่อรันโปรแกรมหากให้สถานะ 0 แก่อินพุต หมายเลข 000 และให้ลอจิก 1 แก่อุปกรณ์อินพุตหมายเลข 001 LED แสดงสถานะเอาต์พุต หมายเลข 100 จะดับ นอกจากนี้ในกรณีป้อนอินพุต แบบอื่นจะทำให้ LED ติดสว่าง แสดงว่าการทำงานของโปรแกรมย่อย OR และ OR-NOT ทำงานได้ถูกต้อง

4.2.5 การทดลองโปรแกรมย่อย ANB , ORB

โปรแกรมย่อย ANB เป็นการนำเอากลุ่ม (BLOCK) ของอุปกรณ์ 2 กลุ่มมาต่ออนุกรมกัน โปรแกรมย่อย ORB เป็นการนำกลุ่ม (BLOCK) ของอุปกรณ์ 2 กลุ่มมาต่อขนานกัน โดยเขียนโปรแกรมทดลอง ดังนี้



รูปที่ 4.14 โปรแกรมทดลองโปรแกรม ANB,ORB

ผลการทดลอง

โปรแกรม A เป็นการทดลองโปรแกรมย่อย ANB หากให้ลอจิก 1 แก่อินพุต หมายเลข 000 และ 001 ตัวใดตัวหนึ่งหรือ ทั้ง 2 ตัว และให้ลอจิก 1 แก่อุปกรณ์อินพุต หมายเลข 002 และ 003 ตัวใดตัวหนึ่งหรือทั้ง 2 ตัว จะทำให้ LED แสดงสถานะของเอาต์พุตหมายเลข 100 ติดสว่าง หากให้ลอจิกแก่อินพุตในแบบอื่นใดนอกจากนี้ LED จะไม่ติด

โปรแกรม B เป็นการทดลองโปรแกรมย่อย ORB หากให้ลอจิก 1 แก่อินพุต หมายเลข 000 และ 001 ทั้งคู่ หรือ ให้แก่อินพุตหมายเลข 002 และ 003 ทั้งคู่ หรือ ให้แก่ทั้ง 4 ตัวจะทำให้

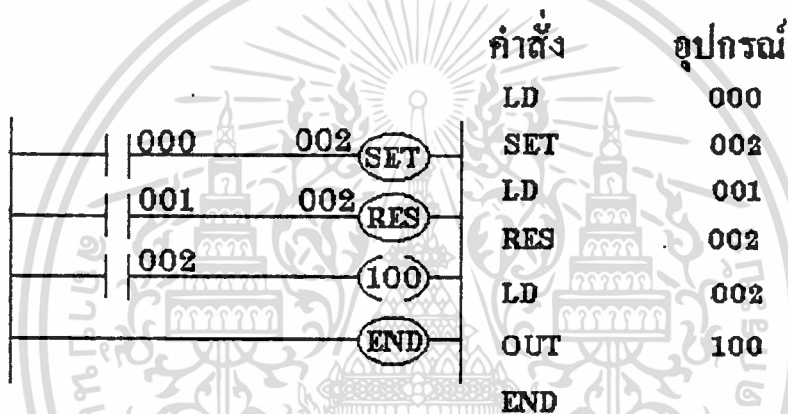
LED แสดงสถานะของเอาต์พุตหมายเลข 100 ติดสว่าง หากให้ลอจิกแก่อินพุตในแบบอื่นใดนอกจากนี้ LED จะไม่ติด

แสดงว่าการทำงานของโปรแกรมย่อย ANB ,ORB ทำงานได้ถูกต้อง

4.2.6 การทดลองโปรแกรมย่อย SET, RES

โปรแกรมย่อย SET เป็นการทำให้หน่วยอินพุตหรือหน่วยเอาต์พุตมีสถานะการทำงานเป็นจริงตลอดเวลาหรือ ON ตลอดเวลา โดยไม่ขึ้นกับสถานะจริง

โปรแกรมย่อย RES เป็นการทำให้อินพุตหรือหน่วยเอาต์พุตมีสถานะการทำงานเป็นเท็จหรือ OFF ตลอดเวลา โดยไม่ขึ้นกับสถานะจริง เขียนโปรแกรมทดลองได้ดังนี้



รูปที่ 4.15 โปรแกรมทดลองโปรแกรมย่อย SET,RES

ผลการทดลอง

เมื่อรันโปรแกรมและให้ลอจิก 1 แก่อินพุตหมายเลข 000 จะทำให้ LED แสดงสถานะของอุปกรณ์เอาต์พุตหมายเลข 100 ติดสว่าง ในขณะที่ยังให้ลอจิก 1 แก่อินพุต หมายเลข 000 อยู่ ให้ลอจิก 1 แก่อินพุตหมายเลข 001 ด้วยมีผลทำให้ LED แสดงสถานะของเอาต์พุตหมายเลข 100 ติด แสดงว่าการทำงานของโปรแกรมย่อย SET ,RES ทำงานได้ถูกต้อง

4.2.7 การทดลองโปรแกรมย่อยสำหรับ TIMER

โปรแกรมย่อยนี้จะทำการตั้งเวลาตามที่กำหนดโดยตั้ง ได้สูงสุด 99.9 วินาทีต่อ TIMER 1ตัว เขียนโปรแกรมทดลองได้ดังนี้

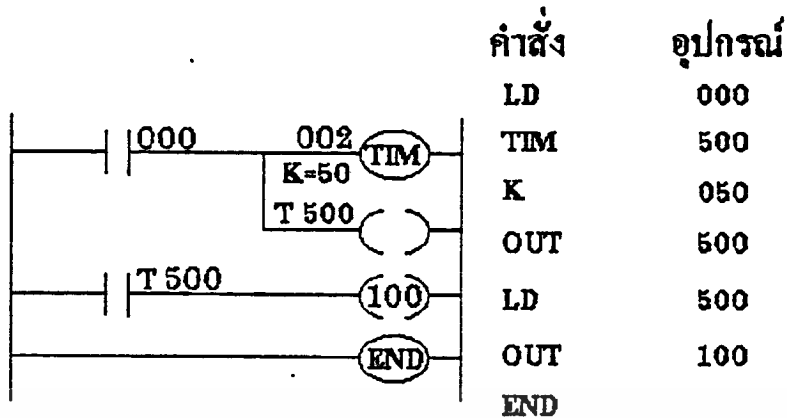
ผลการทดลอง

เมื่อให้ลอจิก 1 แก่อุปกรณ์อินพุตหมายเลข 000 และเริ่มจับเวลาพอเวลาผ่านไป 5 วินาทีจะทำให้ LED แสดงสถานะของเอาต์พุตหมายเลข 100 ติดสว่าง

แสดงว่าโปรแกรมย่อย TIMER ทำการลดค่าลงตามเวลาที่ตั้งไว้ได้ถูกต้อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่อนำไปใช้ประโยชน์ด้านการค้า

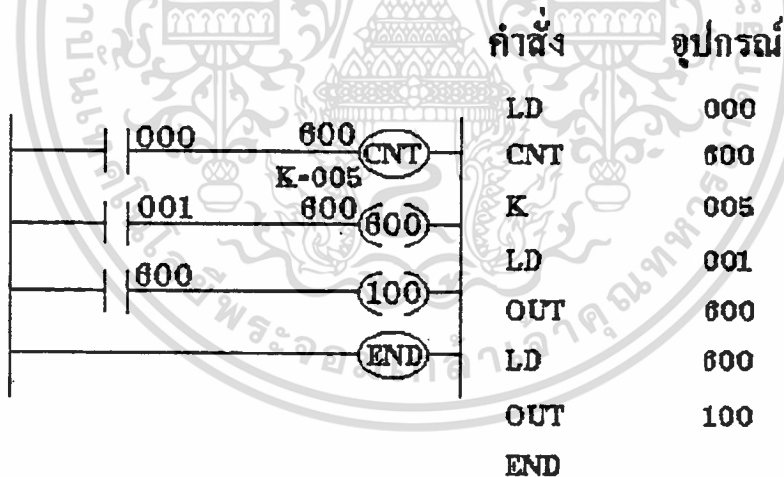
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.16 โปรแกรมทดลองโปรแกรมย่อย TIME

4.2.8 การทดลองโปรแกรมย่อยสำหรับ COUNTER

โปรแกรมย่อยนี้จะทำการตั้งค่า COUNTER โดยตั้งได้สูงสุด 999 ครั้งต่อ COUNTER 1 ตัว เขียนโปรแกรมทดลองได้ดังนี้



รูปที่ 4.17 โปรแกรมทดลองโปรแกรมย่อย COUNTER

ผลการทดลอง

ให้ลอจิก 1 แก่อุปกรณ์อินพุทหมายเลข 000 ตลอดเวลา และ ให้ลอจิก 1 กับลอจิก 0 สลับกัน แก่อุปกรณ์อินพุทหมายเลข 001 ทั้งหมด อย่างละ 5 ครั้ง จะทำให้ LED แสดงสถานะของเอาต์พุทหมายเลข 100 ติดสว่าง

แสดงว่า COUNTER ได้ทำการนับตามเวลาที่ตั้งไว้ได้ถูกต้อง

บทที่ 5

บทวิจารณ์สรุปและแนวทางการพัฒนา

5.1 บทสรุป

เครื่อง PLC เครื่องนี้สามารถปฏิบัติงานตามฟังก์ชันพื้นฐานเหมือนเครื่อง PLC ที่ใช้งานจริงทั่วไปจึงเหมาะที่จะนำไปเป็นชุดทดลอง หรือชุดฝึกเกี่ยวกับการศึกษาการใช้งาน PLC ศึกษาการเขียนโปรแกรม PLC เบื้องต้น และ คุณผลการทำงานของ โปรแกรมที่เขียนขึ้นจากภาคอินพุต และ เอาท์พุตของเครื่องได้ และ สามารถนำ OUTPUT ไปควบคุมการทำงานของอุปกรณ์ต่าง ๆ ได้เนื่องจากเอาท์พุตเป็น RELAY มีหน้าสัมผัสเครื่อง PLC เครื่องนี้สามารถปฏิบัติงานตามฟังก์ชันพื้นฐานเหมือนเครื่อง PLC ที่ใช้งานจริงทั่วไป จึงเหมาะที่จะนำไปเป็น ชุดทดลองเพื่อให้ใช้งานทั้งปกติปิด และ ปกติซึ่งสามารถใช้งานควบคุมอุปกรณ์ต่าง ๆ ได้มากพอสมควร และ คุณภาพที่ได้ใกล้เคียงกับ PLC ที่ขายกันอยู่ในท้องตลาดที่มีราคาสูงในปัจจุบัน

5.2 ปัญหา และ อุปสรรค

1. เนื่องจากเป็นเครื่องที่สร้างขึ้นใหม่ จึงขาดแบบมาศึกษาทำให้การทำโครงการล่าช้า และ อาศัยความเข้าใจในการสร้างขึ้น

2 ข้อมูลเกี่ยวกับเครื่อง PLC ที่ใช้งานจริงเพื่อนำมาเป็นต้นแบบหาได้น้อย เพราะยังไม่มีผู้ทำการเขียนเรื่อง PLC มากนัก รายละเอียดบางส่วนจึงได้จากการศึกษาเครื่อง PLC บ้างจากประสบการณ์ที่เคยใช้บ้าง

3 การออกแบบลายวงจรของ ภาคอินพุต/เอาท์พุต จะเป็นแผ่นใหญ่เนื่องจาก ต้องใช้ BUFFER กันทุก INPUT และ OUTPUT ทำให้ออกแบบยาก และ ราคาที่ทำค่อนข้างสูง

4.การทำกล่องต้องสั่งทำเป็นกล่องโลหะเนื่องจากรูปแบบกล่องที่ต้องการไม่มีขายในท้องตลาดและร้านที่รับทำหาได้ยากเนื่องจากทำจำนวนน้อยร้านส่วนใหญ่จะรับทำจำนวนมาก ๆ จึงทำให้กล่องเสร็จช้า

5.3 แนวทางการพัฒนา

1.ในขณะที่การรันโปรแกรมจะเริ่มรันที่บรรทัดแรกเท่านั้น ถ้าหากจะให้ทำงานโปรแกรมอื่นจะต้องเขียนทับของเก่าทำให้จะกลับมาทำงานโปรแกรมเก่าอีกไม่ได้ควรทำให้เลือกรันที่บรรทัด ใดก็ได้จะทำให้เขียนโปรแกรมใช้งานได้หลายๆ โปรแกรม โดยเขียนขึ้นครั้งเดียว

2.เครื่อง PLC ที่ใช้งานจริงจะพัฒนาไปอย่างรวดเร็ว มีฟังก์ชันพิเศษเพิ่มขึ้นมากมาย การพัฒนาจึงควรให้มีฟังก์ชันพิเศษมากขึ้น

เอกสารนี้ 3. ควรสร้างและพัฒนาขนาดให้เล็กลง เพื่อจะได้สะดวกต่อการเรียนการนำไปศึกษา
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แผนผังโปรแกรมมอเนเตอร์

1. โปรแกรมย่อย LD

ทำการนำค่าสถานะเข้ามาเก็บไว้ โดยจะนำค่าสถานะจากส่วนต่าง ๆ ของเครื่อง PLC โดยแบ่งด้วยค่า OPERANT 3หลักที่ตามหลังคำสั่ง โดยแบ่งออกเป็นส่วนต่าง ๆ ดังนี้

000-007,010-017 เป็นค่าตำแหน่งของINPUT

100-107 เป็นค่าตำแหน่งของOUTPUT

200-207,210-217,220-227,230-237,240-247,250-257

260-267,270-277,280-287,290-297,300-307,310-317

320-327,330-337,340-347,350-357,360-367,370-377

380-387,390-397,400-407,410-417,420-427,430-437

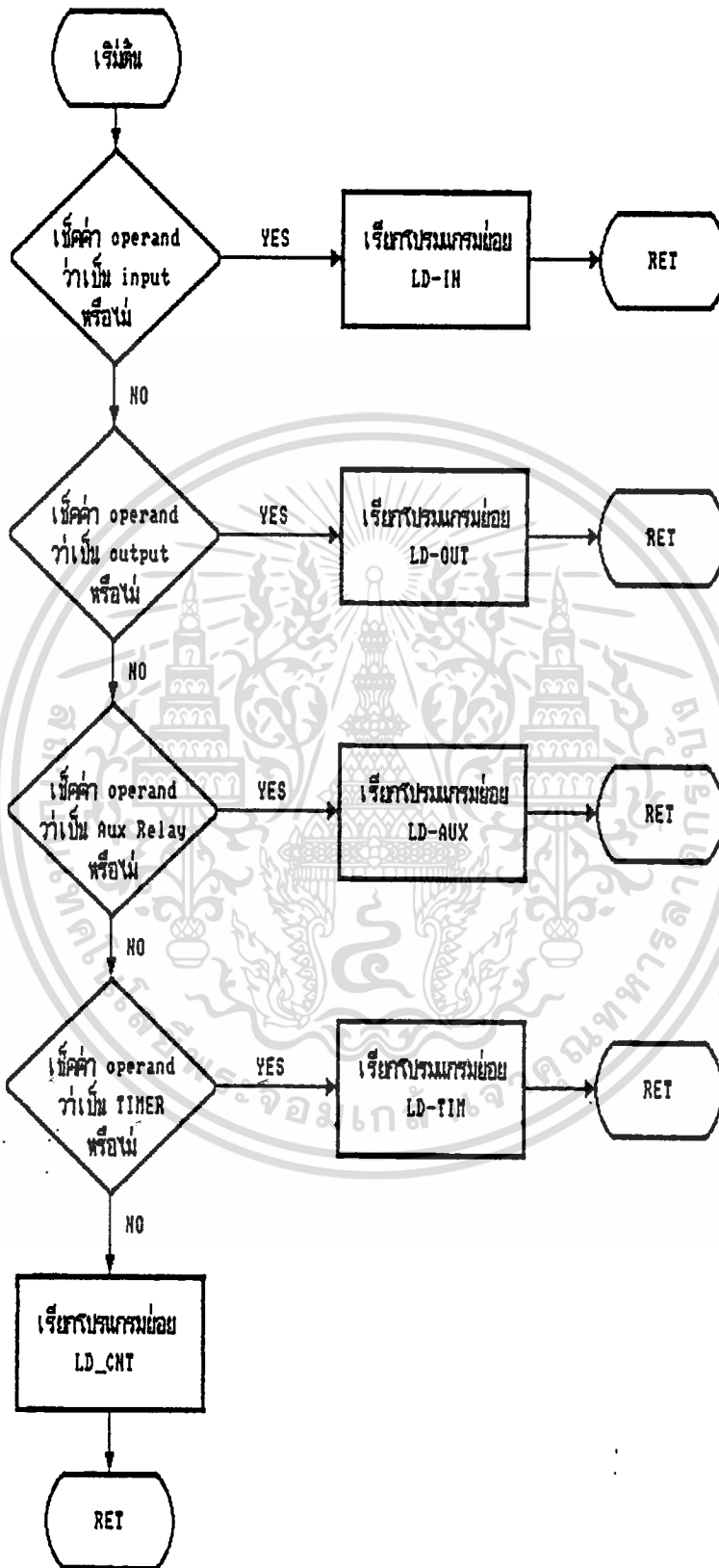
440-447,450-457,460-467,470-477,480-487,490-497 เป็นค่าตำแหน่งของAUX RELAY

500-507,510-517 เป็นค่าตำแหน่งของ TIMER

600-607 เป็นค่าตำแหน่งของ COUNTER

โดยมีโฟลว์ชาร์ต (FLOW CHART) ดังแสดงในรูป

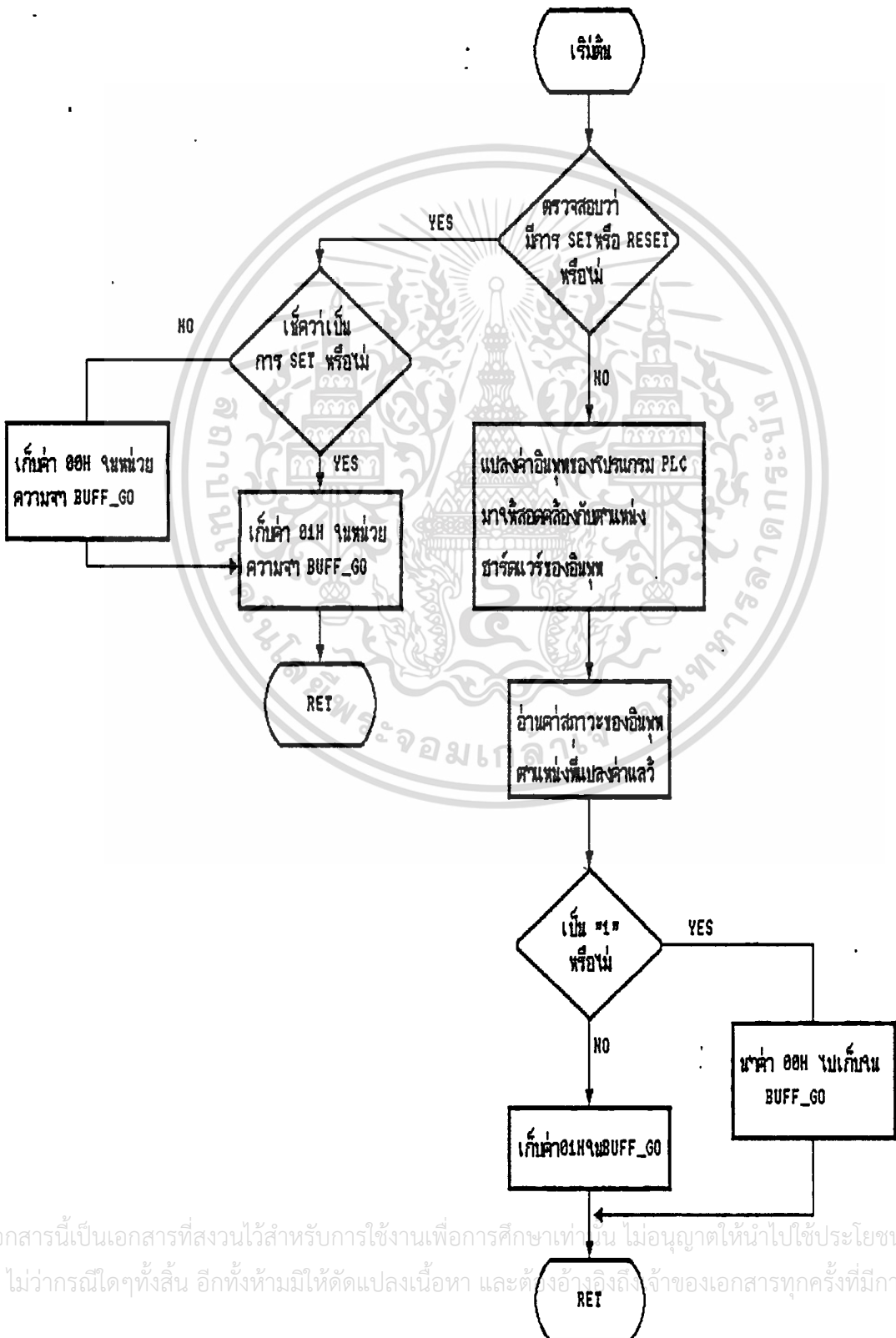




เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

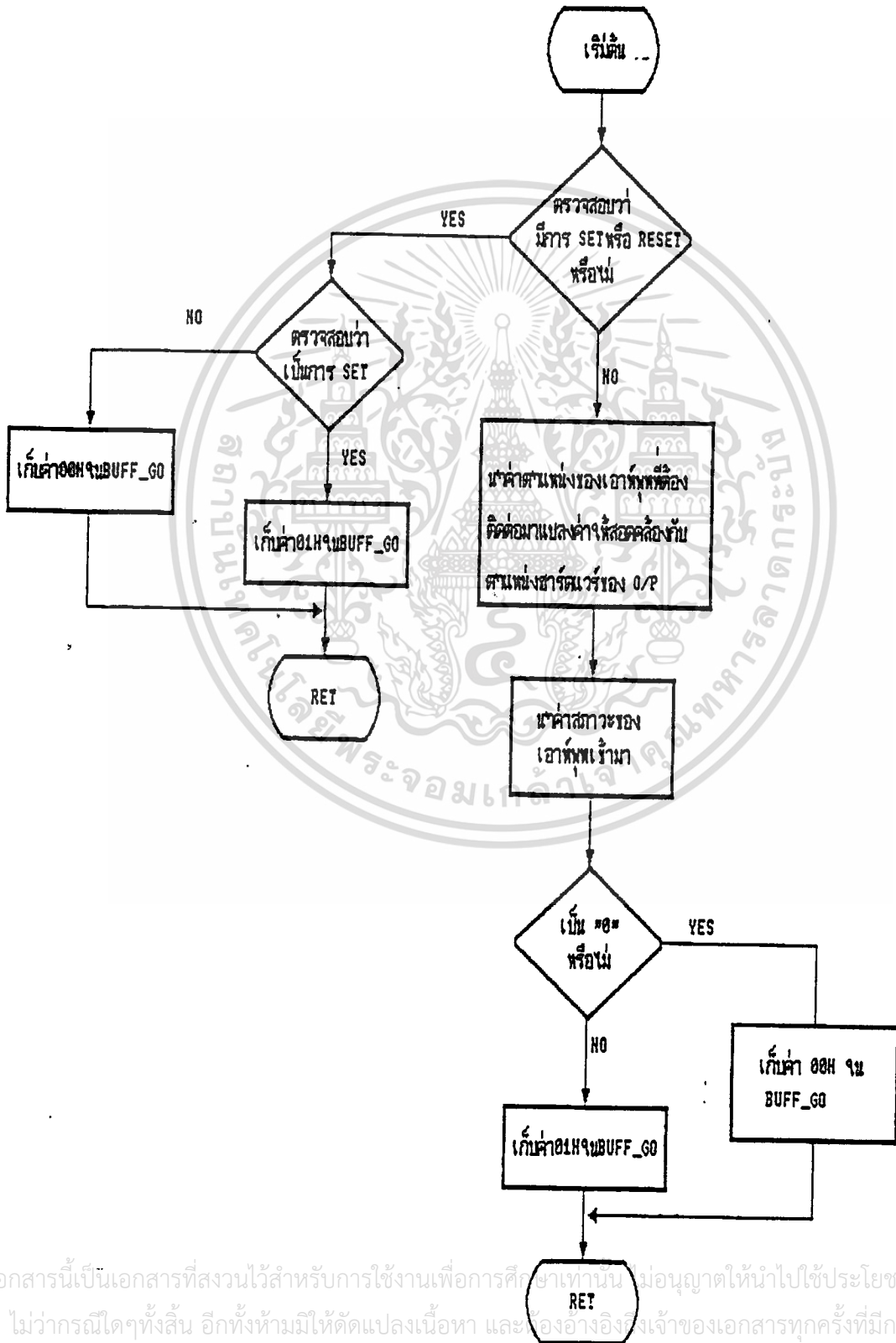
1.1 โปรแกรมย่อย LD-IN

โปรแกรมย่อยนี้ใช้สำหรับนำค่าสถานะของอินพุตภายนอกเข้ามาเพื่อทำการใด ๆ ตามโปรแกรมควบคุม ค่าสถานะอินพุตที่เข้ามานี้เป็นค่าในขณะที่โปรแกรมย่อยกำลังทำการประมวลผลอยู่ ค่าสถานะอินพุตจะถูกเก็บไว้ในหน่วยความจำที่ตำแหน่งชื่อ BUFF_GO



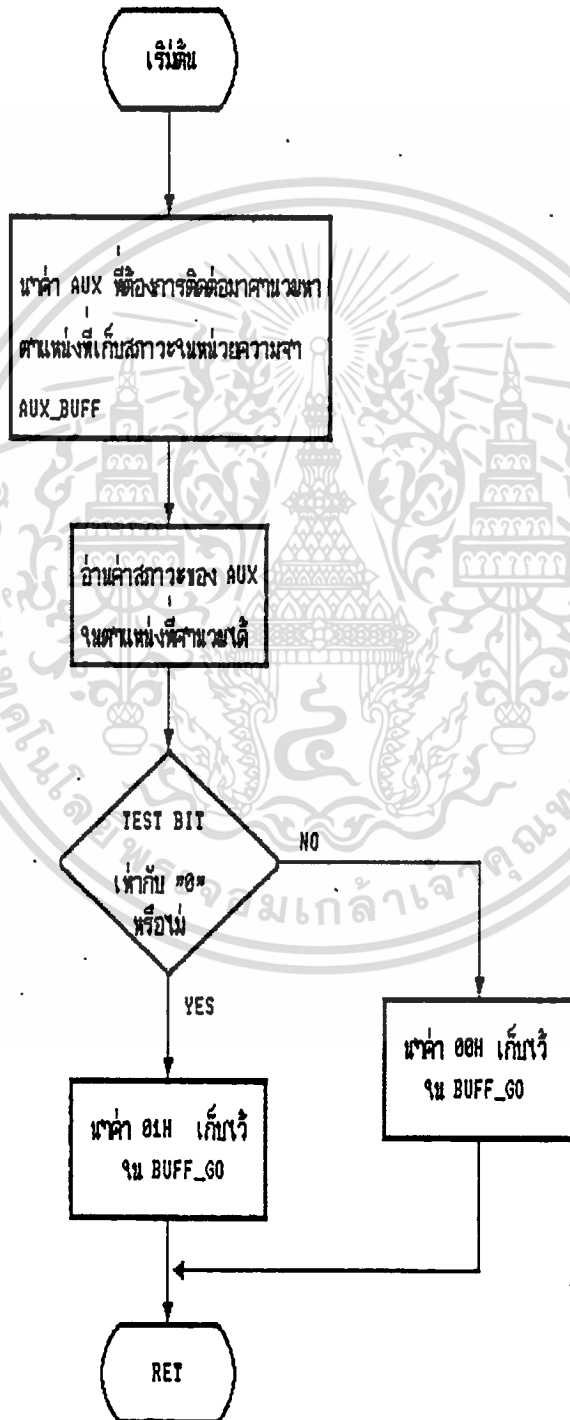
1.2 โปรแกรม LD-OUT

โปรแกรมนี้อาศัยในการนำค่าสถานะของเอาต์พุต จากเอาต์พุตของเครื่อง PLC ตามตำแหน่ง ที่กำหนดด้วยค่า OPERAND แต่ละ ตำแหน่งมีค่าสถานะได้ 2 ค่า คือ 00H และ 01H เท่านั้น ค่าสถานะทั้ง 2 นี้จะบอกถึงสถานะ 0 หรือ 1 หรือเรียกอีกอย่างหนึ่งคือ สถานะ OFF หรือ ON นั่นเอง คังโฟล์ทวารท์ต่อไปนี้



1.3 โปรแกรมย่อย LD-AUX

โปรแกรมย่อยนี้มีลักษณะการทำงานเช่นเดียวกับ LD-OUT ซึ่งค่าจะถูกเก็บในหน่วยความจำชื่อ AUX-BUFF ขนาด 1 บิต ต่อ 1 AUX และทั้งหมดมี AUX 240 ตัว จึงใช้หน่วยความจำ AUX-BUFF 30 ไบท์

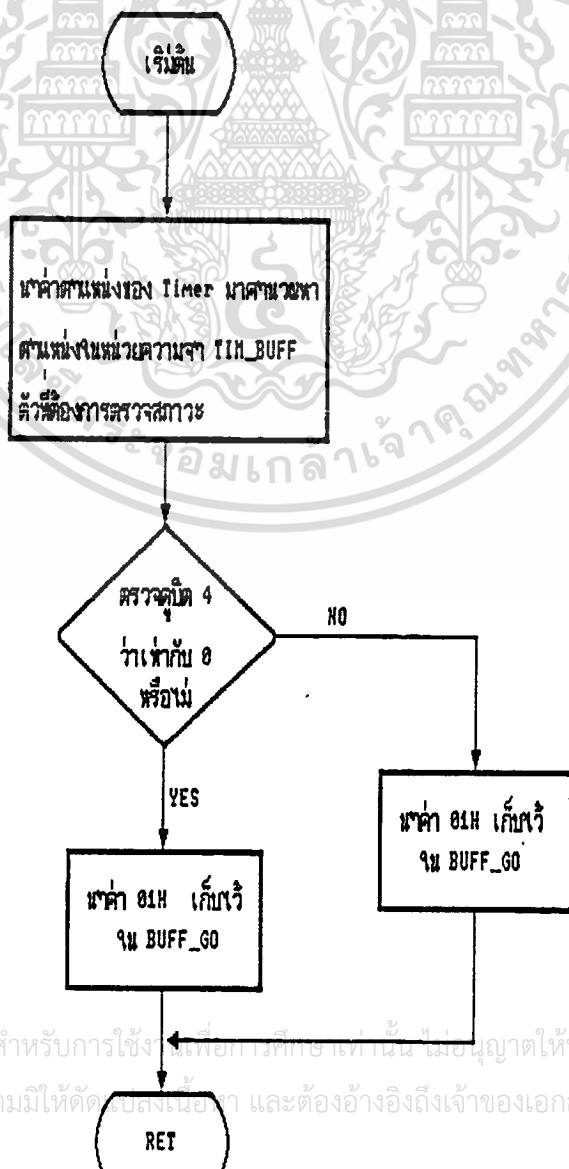
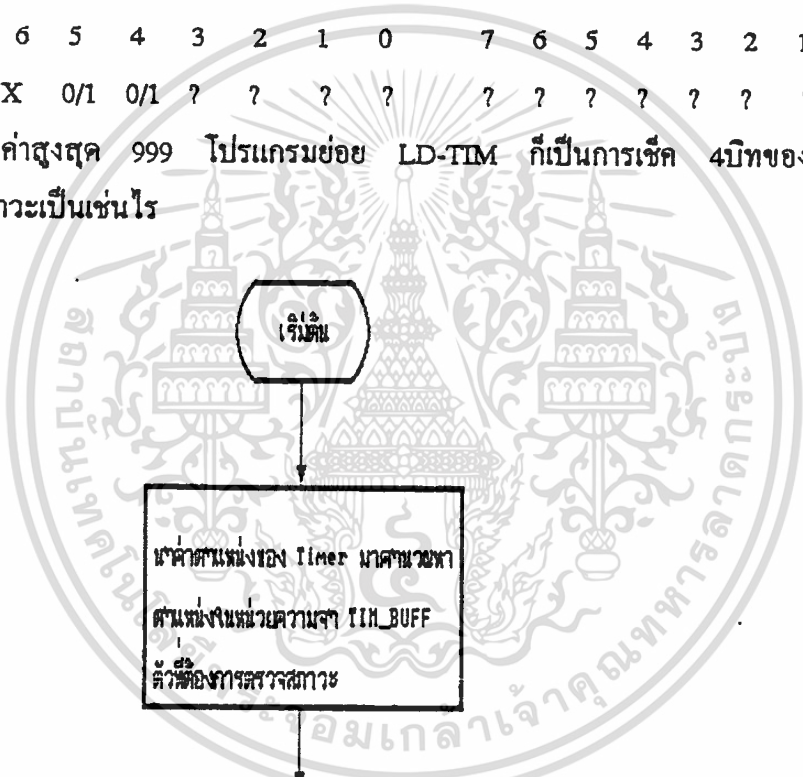


1.4 โปรแกรมย่อย LD-TIM

โปรแกรมย่อยนี้มีไว้สำหรับนำค่าสถานะของตัวตั้งเวลาเข้ามา เพื่อทำการประมวลผลตามโปรแกรมควบคุม ค่าสถานะของตัวตั้งเวลา มี 2 ค่าขึ้นอยู่กับการทำงานของตัวตั้งเวลาว่าถึงเวลาที่ตั้งไว้หรือยัง เมื่อตัวตั้งเวลาทำงาน และครบตามเวลาที่กำหนดไว้ค่าสถานะดังกล่าวก็จะเป็น 01H หรือสถานะ ON นั่นเอง แต่ถ้ายังไม่ครบตามเวลาที่ตั้งไว้ค่าสถานะดังกล่าวก็จะเป็น 00H หรือสถานะ OFF ในการทำงานของโปรแกรมย่อยนี้จะนำค่าสถานะดังกล่าวจากหน่วยความจำ ซึ่งตัวตั้งเวลา 1 ตัวจะใช้เนื้อที่ ในหน่วยความจำขนาด 2 ไบต์ โดยมีรูปแบบดังนี้

	ไบต์สูง								ไบต์ต่ำ							
บิต	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
	X	X	0/1	0/1	?	?	?	?	?	?	?	?	?	?	?	?

โดยค่าที่ตั้งไว้มีค่าสูงสุด 999 โปรแกรมย่อย LD-TIM ก็เป็นการเช็ค 4บิตของไบร์สูงของ TIMERว่ามีสถานะเป็นเช่นไร



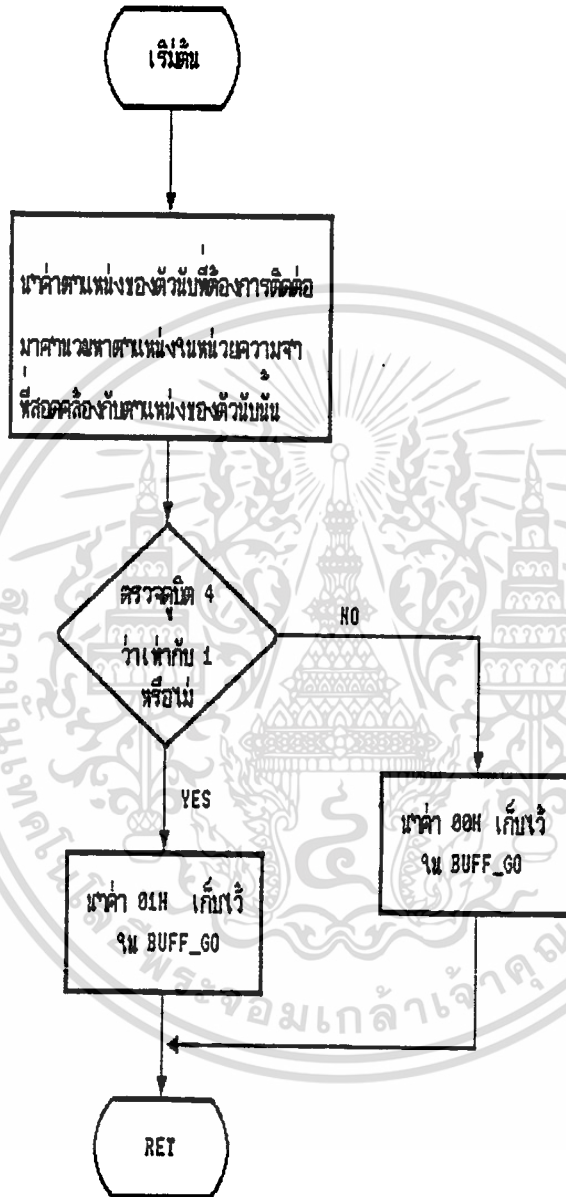
1.5 โปรแกรมย่อย LD-CNT

โปรแกรมย่อยนี้มีไว้สำหรับนำค่าสถานะของตัวนับเข้ามาเพื่อทำการประมวลผลตามโปรแกรมควบคุม สถานะของตัวนับมีอยู่ 2 ค่าเช่นกัน ขึ้นกับการนับอินพุตได้ครบตามจำนวนนับที่กำหนดไว้หรือไม่ เมื่อการนับของอินพุตได้ครบตามจำนวนที่ตั้งไว้ ค่าของสถานะจะเป็น 01H หรือ ON แต่ถ้ายังไม่ครบตามจำนวนที่กำหนด ค่าสถานะดังกล่าวก็จะเป็น 00H หรือสถานะ OFF นั่นเอง

ในการทำงานของโปรแกรมย่อยนี้ จะนำค่าสถานะของตัวนับจากหน่วยความจำซึ่งตัวนับ 1 ตัว จะใช้เนื้อที่ในหน่วยความจำขนาด 2 ไบต์ โดยมีรูปแบบดังนี้

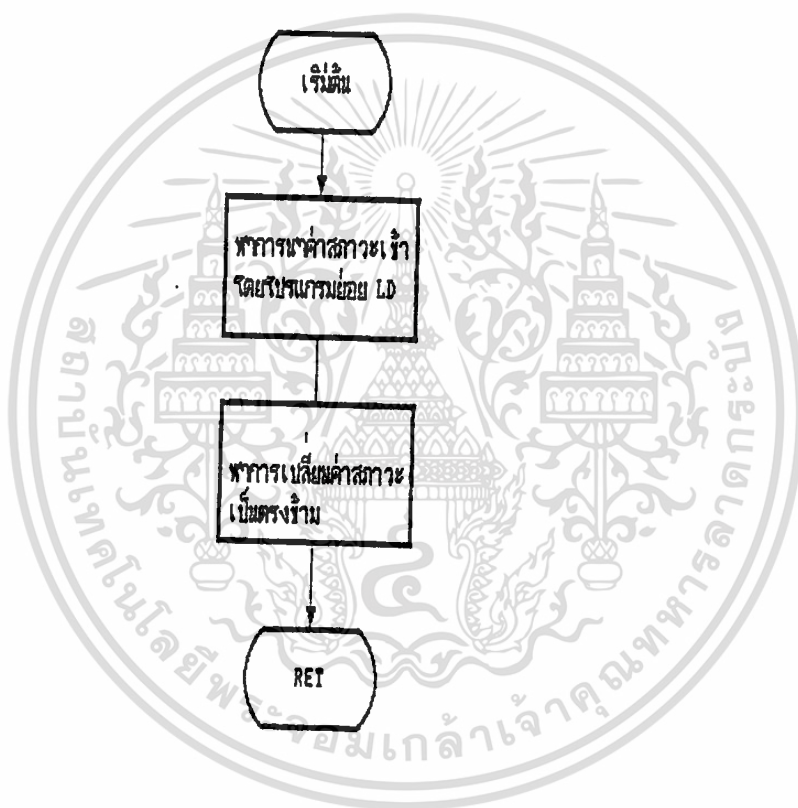
ไบต์สูง								ไบต์ต่ำ								
บิต	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
ค่าเก่า	X	ON/OFF	0/1	หลักร้อย				หลักสิบ				หลักหน่วย				

บิต 4 ของไบต์สูงเป็นบิตสถานะ ซึ่งมีคุณสมบัติเช่นเดียวกับตัวตั้งเวลาคือเมื่อตัวนับทำการนับครบตามจำนวนที่กำหนด บิต 4 ก็จะมีสถานะเป็น 1 แต่ถ้ายังไม่ครบตามจำนวนนับ ก็จะมีสถานะลอจิกเป็น 0 ซึ่งการทำงานของโปรแกรมย่อยนี้จะเป็นการตรวจสอบบิต 4 นั่นเอง เมื่อตรวจสอบพบว่า บิต 4 เป็น 1 ก็จะนำค่าสถานะ 01H เก็บไว้ใน BUFF-GO แต่ถ้าบิต 4 เป็น 0 ก็จะนำค่าสถานะ 00H เก็บไว้ใน BUFF-GO



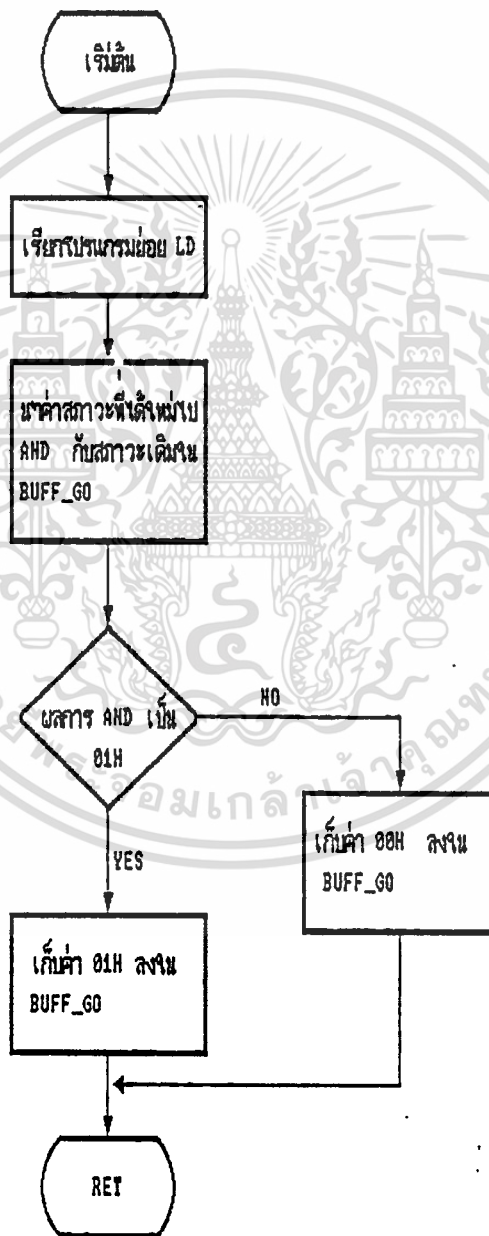
2. โปรแกรมย่อย LD-NOT

โปรแกรมย่อยนี้ใช้สำหรับเก็บค่าสถานะของอินพุตปกติไม่ทำงาน (MOVMAJLY OPEN) เมื่อรับสถานะเข้ามาแล้ว ก็จะทำการกลับสถานะให้อยู่ในสถานะตรงกันข้าม คือถ้าไม่ทำงาน (ลอจิก 0) ก็จะกลับเป็นสถานะทำงาน (ลอจิก 1) เพื่อให้สอดคล้องกับจุดมุ่งหมายของการใช้งานของคำสั่งนี้วิธีการของโปรแกรมย่อยนี้คือ จะนำค่าสถานะโดยโปรแกรมย่อย LD แล้วทำการแปลงค่าจาก 00H เป็น 01H จาก 01H เป็น 00H ดังแสดงโฟลว์ชาร์ท



3. โปรแกรมย่อย AND

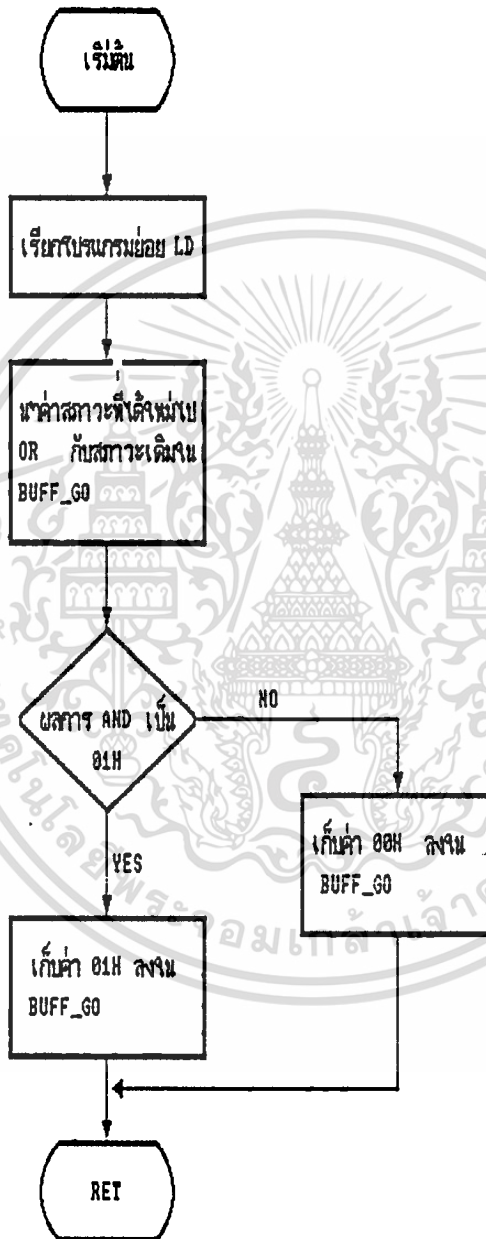
โปรแกรมย่อยนี้ใช้ในการนำค่าสถานะของค่าของส่วนใด ๆ ของเครื่อง PLC เข้ามา ซึ่งจะทราบว่านำค่าจากส่วนใดด้วยค่า OPERAND 3 หลักที่ตามหลังคำสั่ง AND และนำมากระทำการ AND กับค่าสถานะก่อนหน้าใน BUFF-GO ค่าสถานะที่ได้จากการทำการ AND ก็จะเก็บไว้ในหน่วยความจำ BUFF-GO โดยลบสถานะเดิมใน BUFF-GO ที่ใช้ไปแล้ว โดยนำสถานะที่ได้จากการ AND เขียนทับลงไป



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. โปรแกรมย่อย OR

เช่นเดียวกับโปรแกรมย่อย AND แต่นำค่าสถานะที่ได้มาใหม่มากระทำการ OR กับสถานะเดิมใน BUFF_GO แทน



5. โปรแกรมย่อย SET

โปรแกรมย่อยนี้จะทำการ SET BIT ประจำตำแหน่งของส่วน INPUT กับ OUTPUT ให้เป็น 1 ซึ่งส่วน

อินพุต 000-007 จะอ้างอิงกับหน่วยความจำชื่อ IN 0 SET ขนาด 1 ไบท์

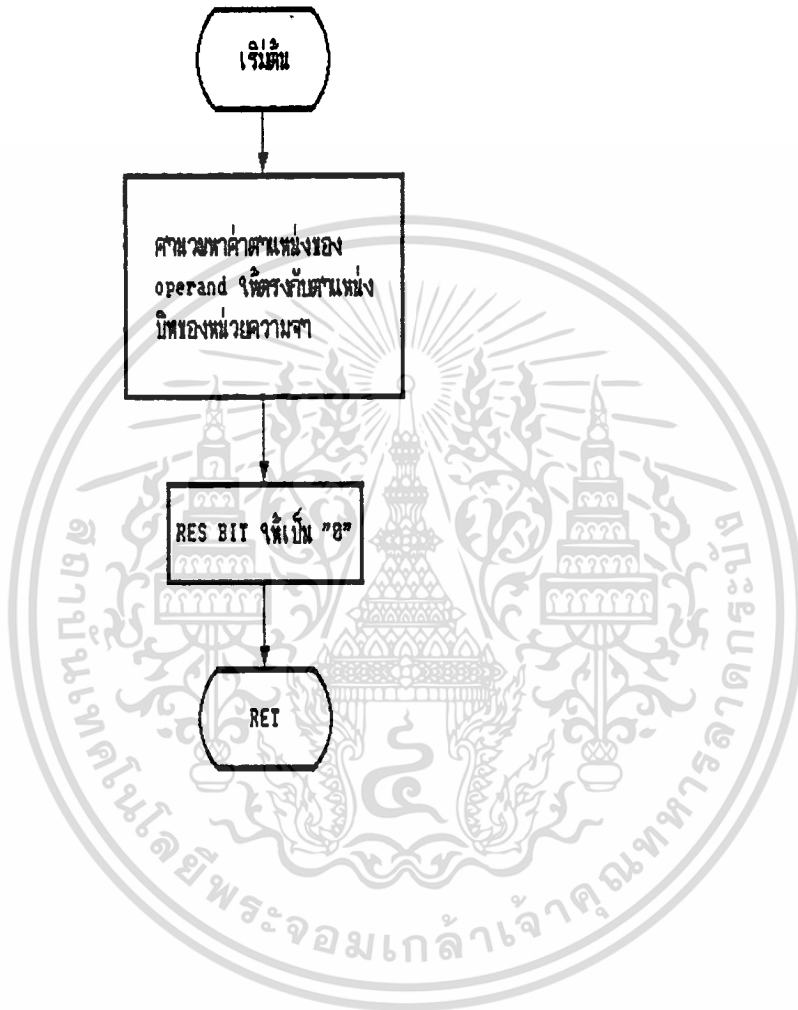
อินพุต 010-017 จะอ้างอิงกับหน่วยความจำชื่อ IN 1 SET ขนาด 1 ไบท์

อินพุต 100-107 จะอ้างอิงกับหน่วยความจำชื่อ OUT SET ขนาด 1 ไบท์
ซึ่งใช้อ้างอิงส่วน อินพุต หรือเอาต์พุต 1 ตัวต่อ 1 บิท



6. โปรแกรมย่อย RES

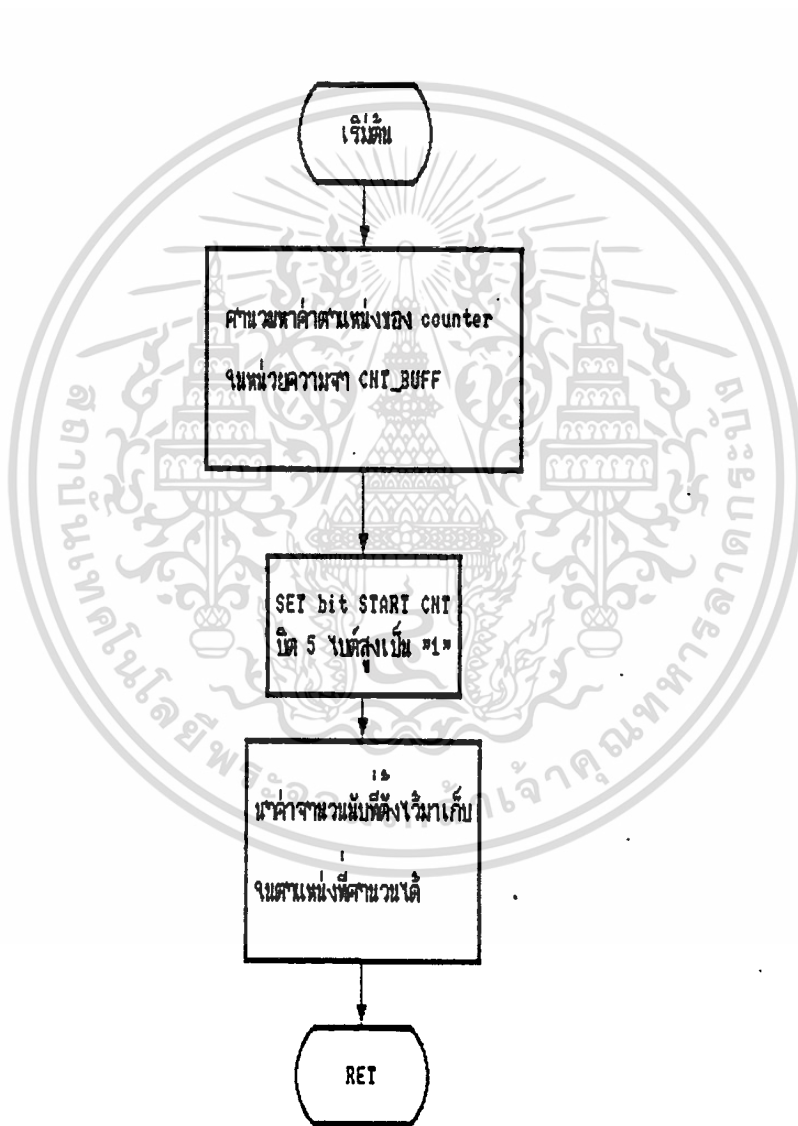
โปรแกรมย่อยนี้จะใช้เหมือนกับโปรแกรมย่อย SET แต่จะกระทำการ RESET BIT ประจำตำแหน่งของส่วน INPUT กับ OUTPUT ให้เป็น 0



7. โปรแกรมย่อย CNT

ตัวนับที่ออกแบบไว้ใน PLC ตัวนับแต่ละตัวที่ออกแบบไว้จะ ใช้เนื้อที่ในหน่วยความจำขนาด 2 ไบท์

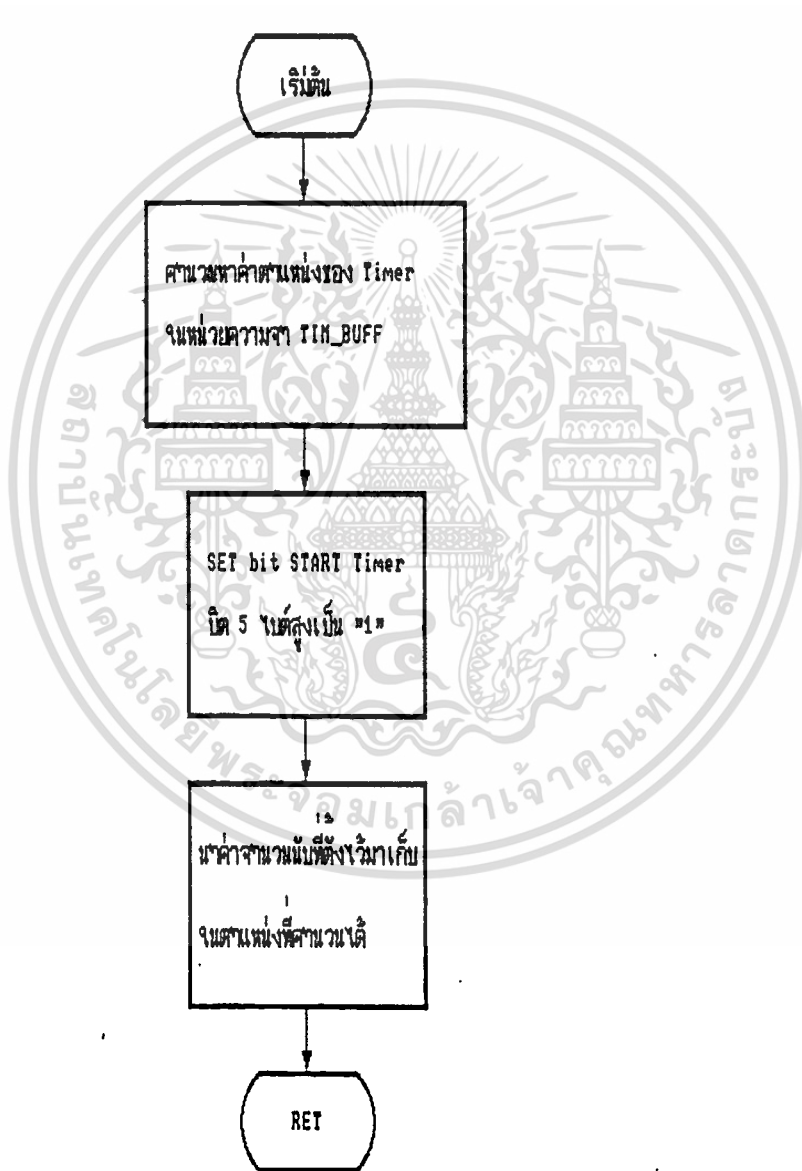
ส่วนโปรแกรมย่อย CNT นี้จะทำการ ON หรือ OFF COUNTER ตามค่าสถานะของอุปกรณ์ที่ถูก LD เข้ามาก่อนการเรียกใช้ COUNTER และยังทำการเก็บจำนวนนับจากค่า K ลงในพื้นที่ของ COUNTER



8. โปรแกรมย่อย TIMER

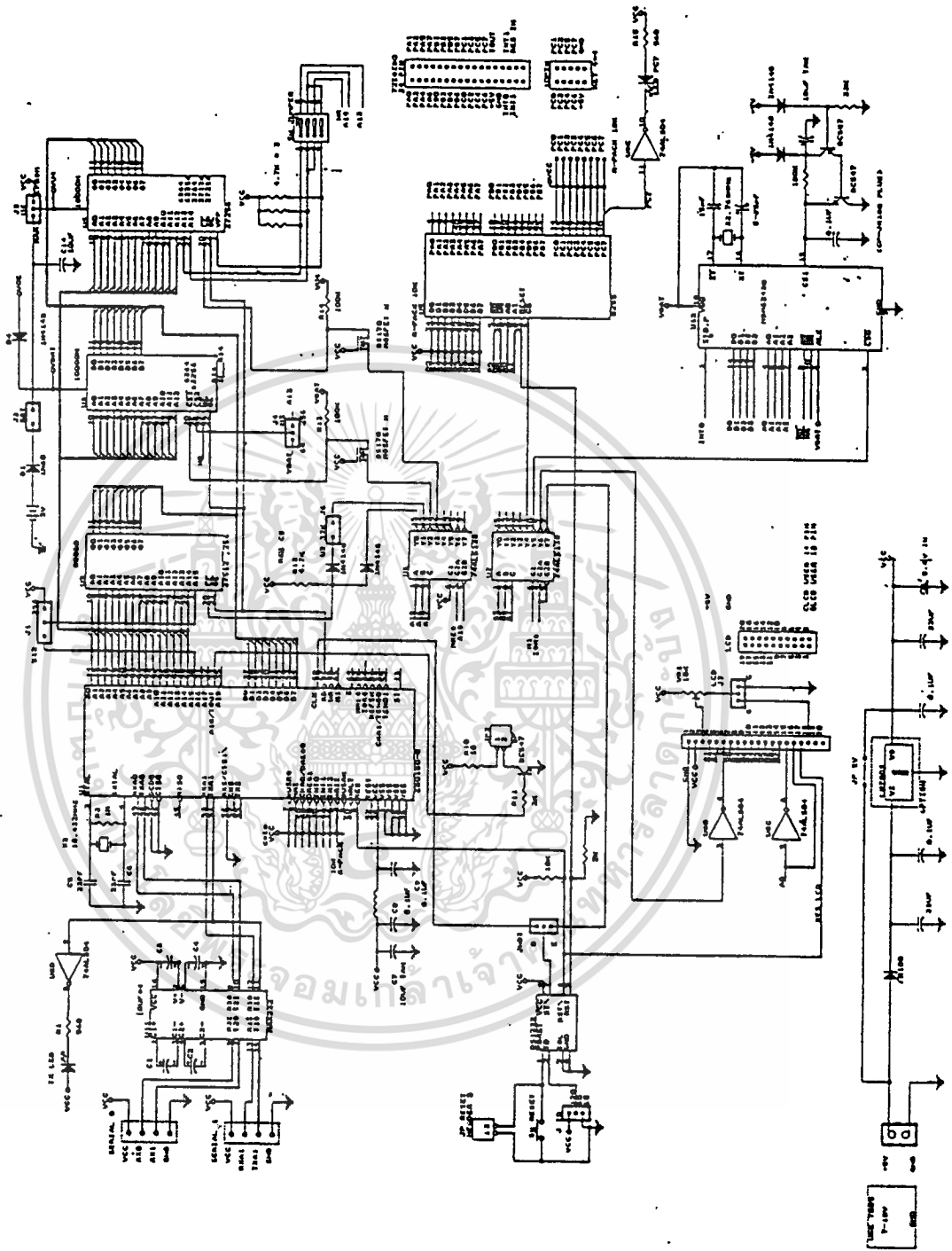
TIMER ที่ออกแบบไว้ใน PLC ใช้วิธีการทางซอฟต์แวร์ และ ฮาร์ดแวร์ร่วมกันโดยที่ TIMER แต่ละตัวจะใช้เนื้อที่ขนาด 2 ไบท์

ส่วนของโปรแกรมย่อย TIMER นี้จะทำการ ON หรือ OFF TIMER ตามค่าสถานะของ อุปกรณ์ที่ถูก LD เข้ามาก่อนการเรียกใช้ TIMER และก็ทำการเก็บค่าเวลาที่ตั้งไว้ จากค่า K ลงในพื้นที่ของ TIMER



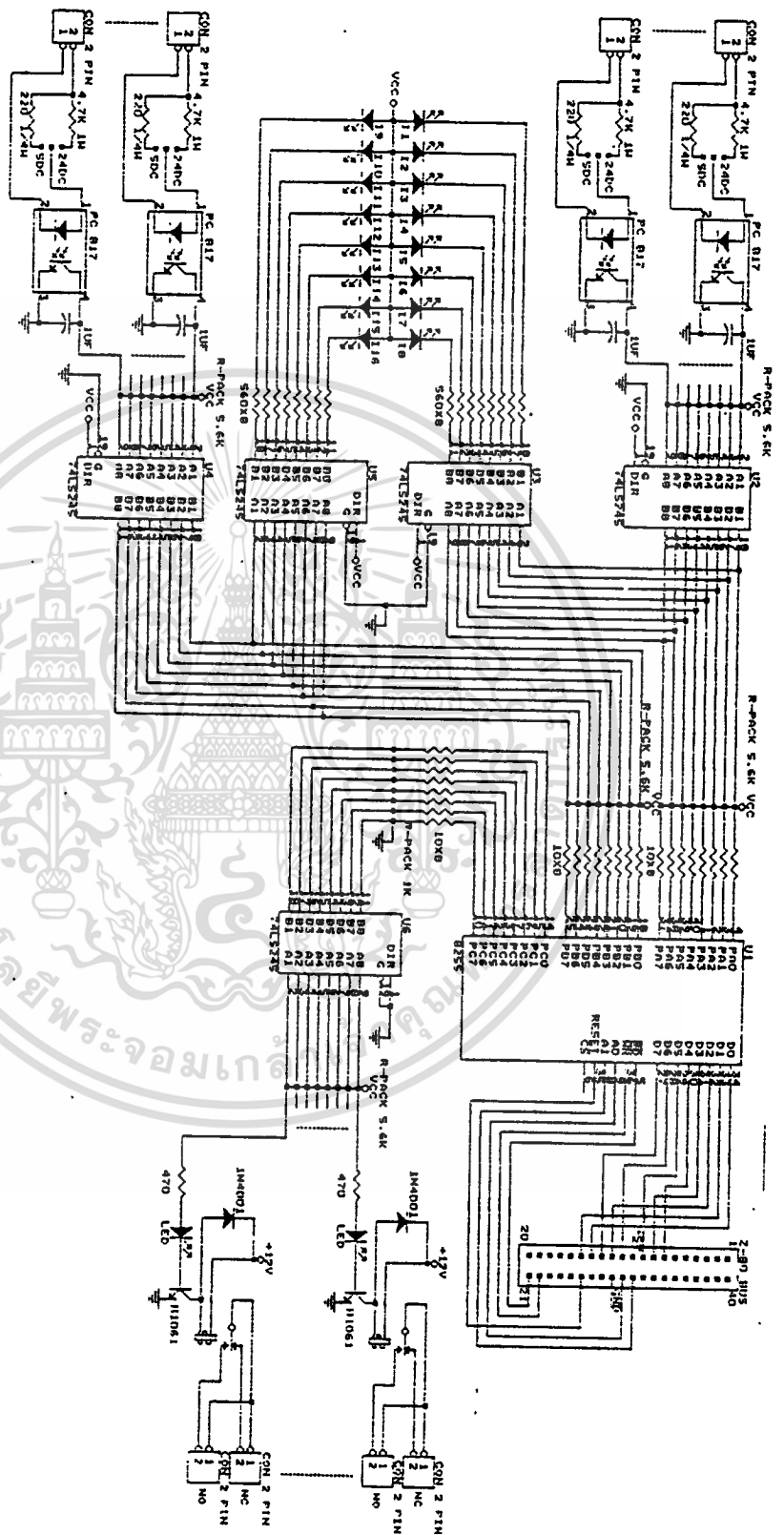


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

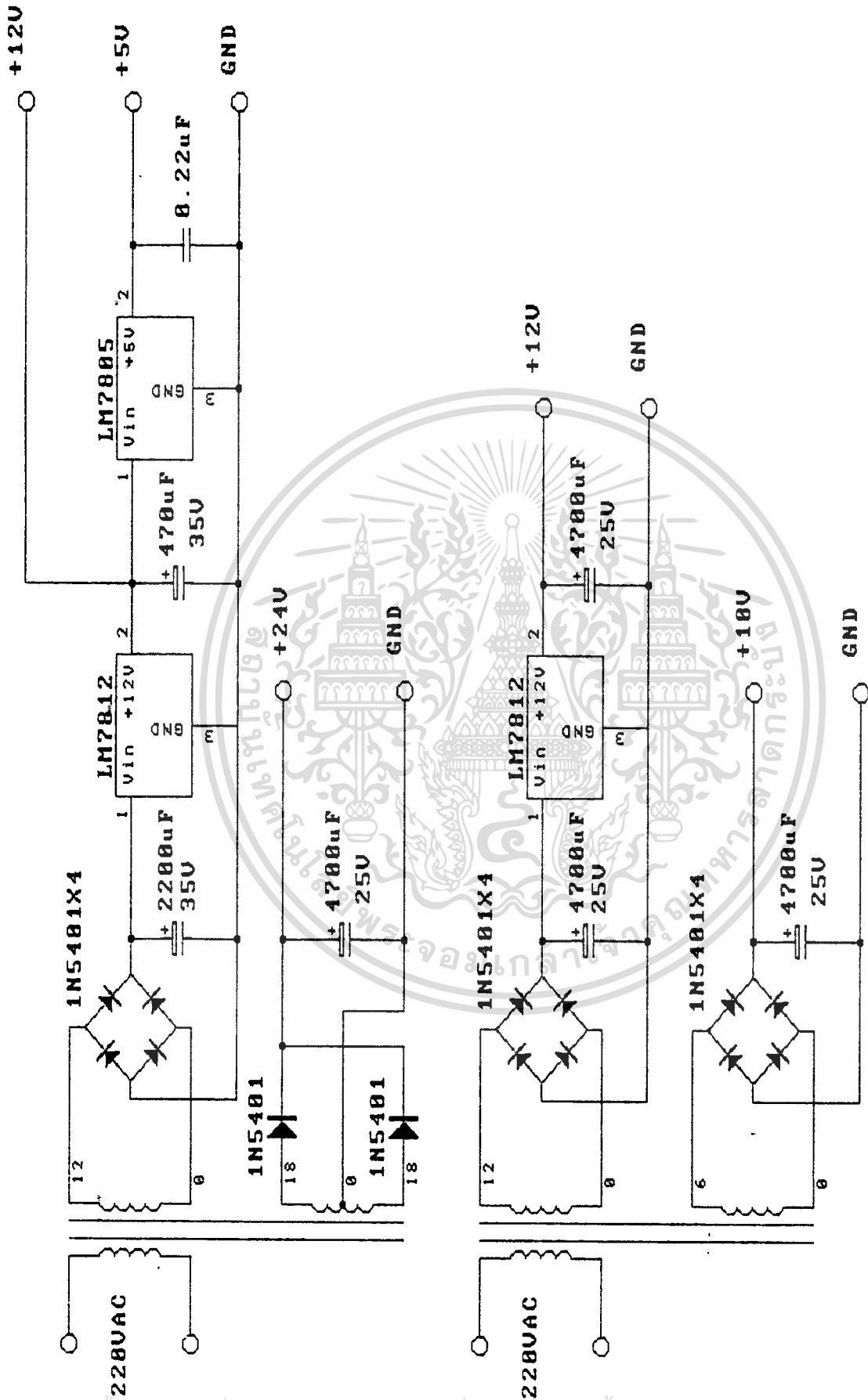


วงจรสมบูรณของบอร์ด CP-JR180

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

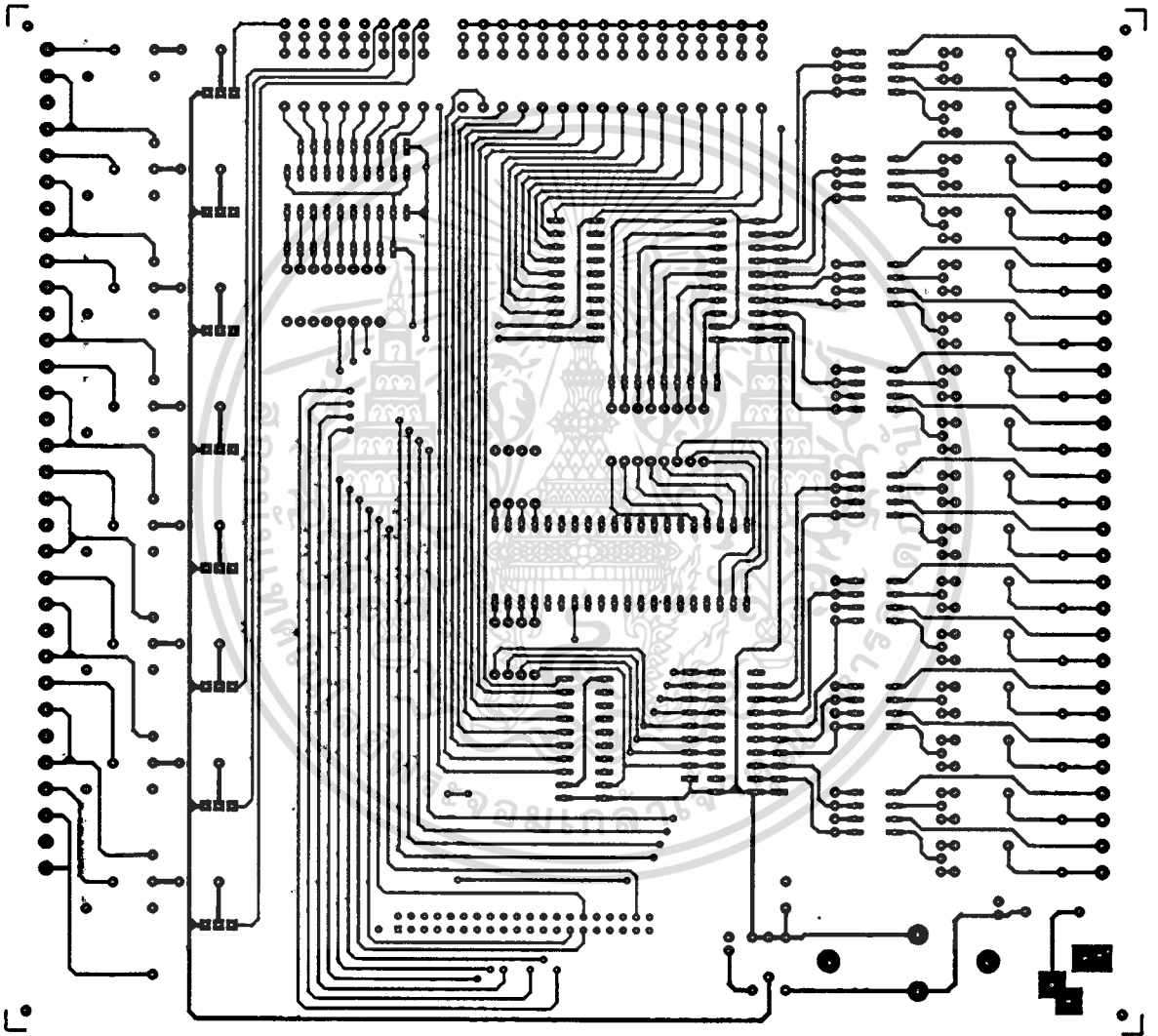


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

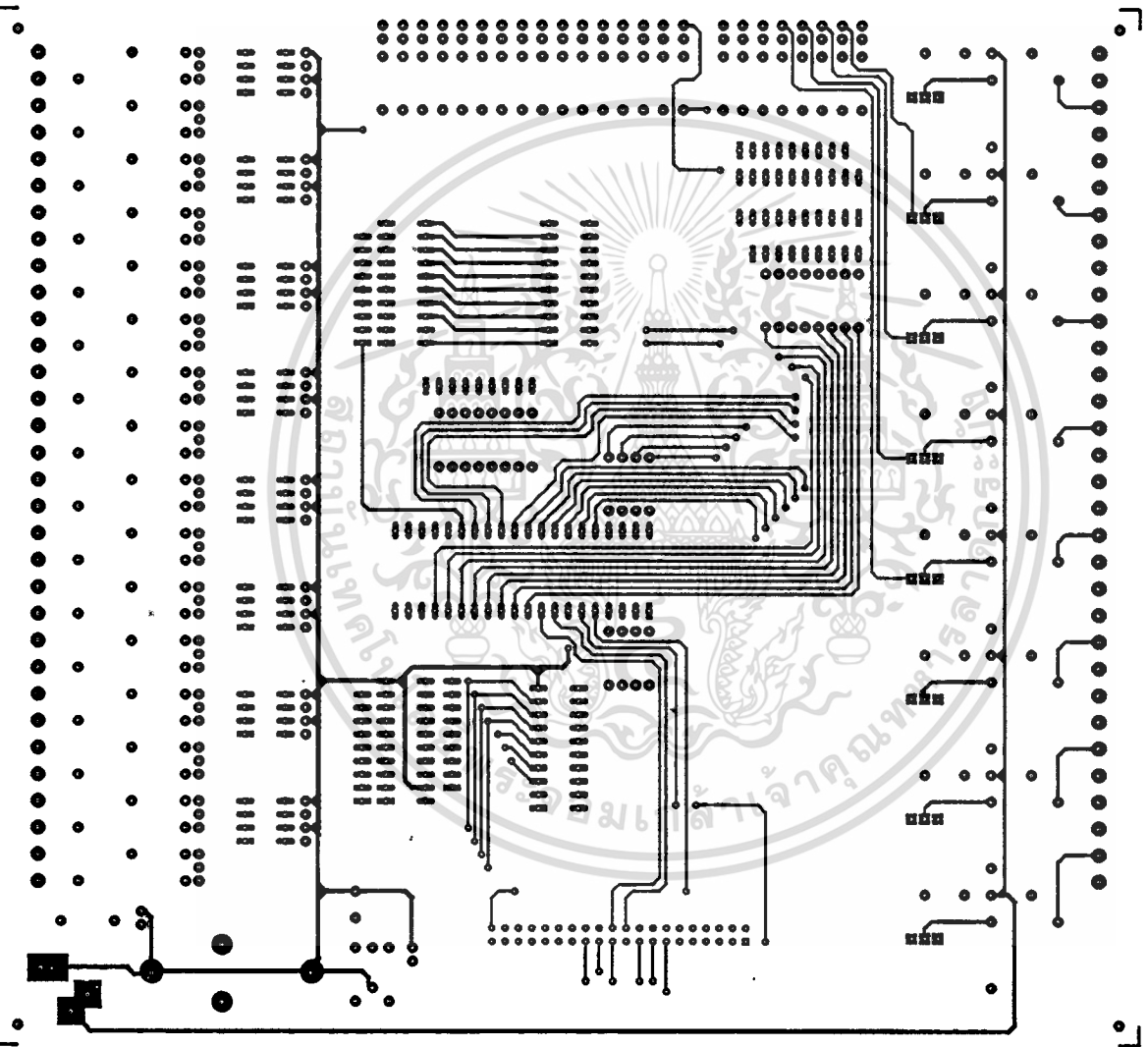


POWER SUPPLY

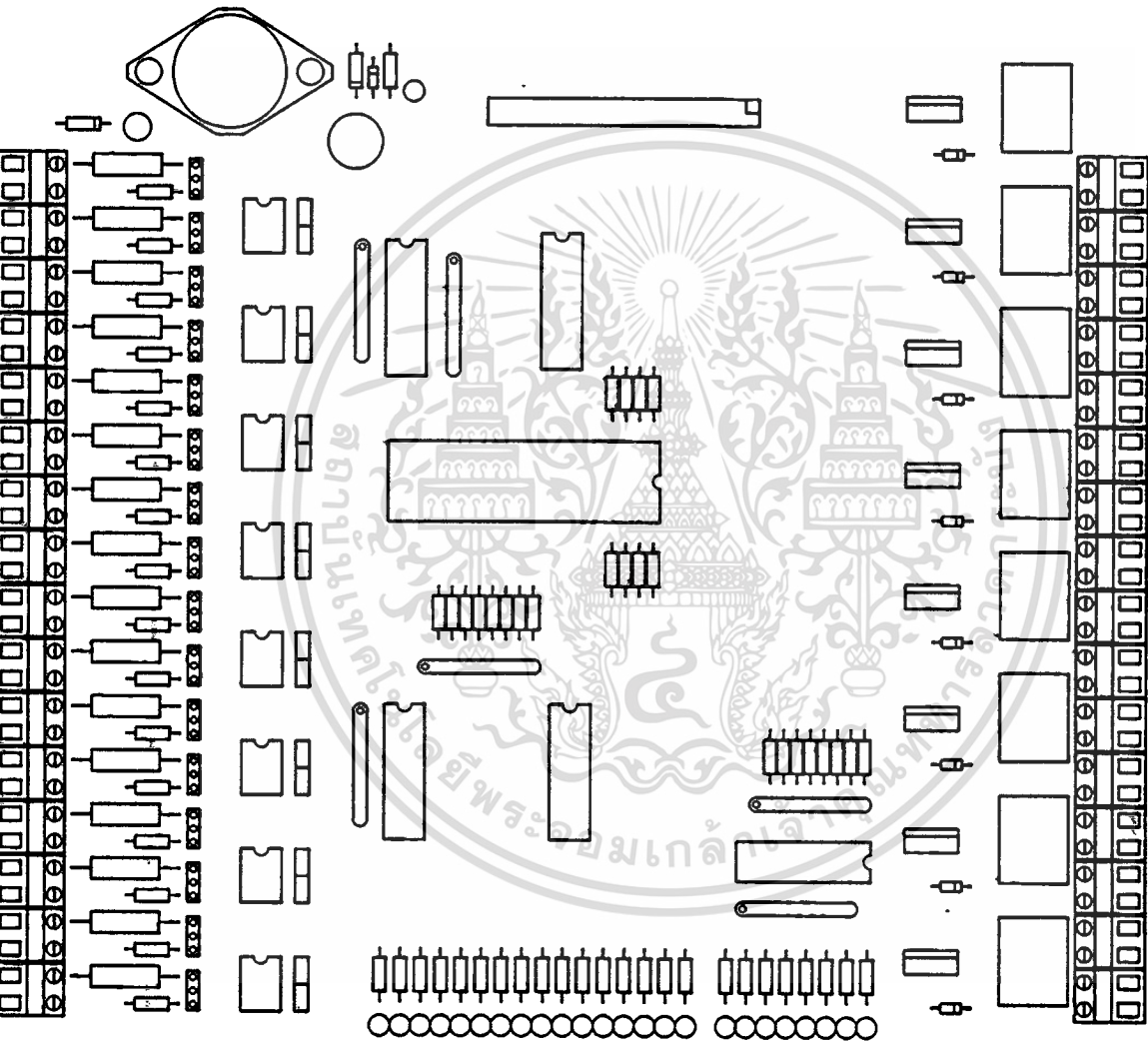
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาคผนวก ค.โปรแกรมใน PLC

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

000001 0000 ;*****
000002 0000 ;*          MONITOR PROGRAM          *
000003 0000 ;*    PROGRAMMABLE LOGIC CONTROLLER    *
000004 0000 ;*          DEMONSTATE                *
000005 0000 ;*****
000006 0000
000007 0000 ;***** PORT KEY *****
000008 0082 .EQU PORTC_KEY,82H
000009 0083 .EQU PORTCT1,83H
000010 0000
000011 0000 ;***** I/O LCD PORT *****
000012 00C0 .EQU PDATA,0C0H
000013 00C2 .EQU PSING,0C2H
000014 00C4 .EQU PREAD,0C4H
000015 00C6 .EQU PREAD_DATA,0C6H
000016 0000
000017 0000 ;***** EQUATE SYSTEM VALUE *****
000018 F000 .EQU KEYIN,0F000H ;KEYIN: .DS 1
000019 F001 .EQU KEY_DATA,0F001H ;KEY_DATA: .DS 1
000020 F002 .EQU DATA_ADDR,0F002H ;DATA_ADDR: .DW 1
000021 F004 .EQU LINE,0F004H ;LINE: .DW 1
000022 F006 .EQU BUFF_DATA_ADDR,0F006H ;BUFF_DATA_ADDR: .DW 1
000023 F008 .EQU BUFF_LINE,0F008H ;BUFF_LINE: .DW 1
000024 F00A .EQU ADDR_NUM,0F00AH ;ADDR_NUM: .DW 1
000025 F00C .EQU POINT,0F00CH ;POINT .DW 1
000026 F00F .EQU BUFF_CURSOR,0F00FH ;ADDR CUR SOR .DS 1
000027 F010 .EQU FIRST_START,0F010H ;CHECK FIRST .DS 1

```

```

000028 0000
000029 0000          ;***** FUNC_GO *****
000030 F011          .EQU BUFFER1,0F011H
000031 F012          .EQU BUFFER2,0F012H
000032 F013          .EQU CNT_P_BUFF,0F013H          ;BUFFER SAVE POINTER ADDR CO
000033 F015          .EQU AUX_P_BUFF,0F015H          ;BUFFER SAVE POINTER ADDR AU
000034 F017          .EQU TIM_P_BUFF,0F017H
000035 F020          .EQU DATA_POINTER,0F020H          ;POINTER FOR GO .DW 1
000036 F022          .EQU POINT_BUFF,0F022H
000037 F024          .EQU BUFF_GO,0F024H          ;BUF FOR 1 INTERPRETER
000038 F030          .EQU AUXPOINT,0F030H
000039 F050          .EQU TIMPOINT,0F050H
000040 F070          .EQU CNTPOINT,0F070H
000041 0000
000042 F080          .EQU IN_0,0F080H
000043 F081          .EQU IN_1,0F081H
000044 F082          .EQU OUTPUT,0F082H
000045 F083          .EQU INOSET,0F083H
000046 F084          .EQU IN1SET,0F084H
000047 F085          .EQU OUTSET,0F085H
000048 0000
000049 00E0          .EQU PORTA,0E0H
000050 00E1          .EQU PORTB,0E1H
000051 00E2          .EQU PORTC,0E2H
000052 00E3          .EQU PORT_CONTROL,0E3H
000053 0000
000054 000C          .EQU TMDROL,0CH

```

```

000055 000D      .EQU  TMDROH,0DH
000056 000E      .EQU  RLDROL,0EH
000057 000F      .EQU  RLDROH,0FH
000058 0010      .EQU  TCR,10H
000059 0033      .EQU  IL,33H

000060 0000
000061 0000      ;*****
000062 0000      .ORG  0000H
000063 0000 210080  POWER:  LD    HL,8000H
000064 0003 00      POWER1: NOP
000065 0004 2B      DEC    HL
000066 0005 7C      LD    A,H
000067 0006 B5      OR    L
000068 0007 20FA     JR    NZ,POWER1
000069 0009 3100FF     LD    SP,OFF00H ;LD STACK POINTER
000070 000C CD6E00     TLCD:  CALL  INITLCD
000071 000F 211800     LD    HL,TAB1
000072 0012 CD3800     CALL  WRP1
000073 0015 C39700     JP    RESTART
000074 0018 2A2A4B494E47204D TAB1:  .DB  "***KING HONGKUT**"
000075 0020 4F4E474B55542A2A
000076 0028 2A2A45442E454E47 TAB2:  .DB  "***ED.ENGINEER***"
000077 0030 494E4545522A2A2A
000078 0038 3E00      WRP1:  LD    A,00H
000079 003A CD4900      CALL  GOTO
000080 003D CD5A00      CALL  WRLINE
000081 0040 3E40      LD    A,40H

```

000080	0042	CD4900	CALL	GOTO
000081	0045	CD5A00	CALL	WRLINE
000082	0048	C9	RET	
000083	0049	CBFF	GOTO:	SET 7,A
000084	004B	D3C0	OUT	(PDATA),A
000085	004D	CD5100	CALL	READ
000086	0050	C9	RET	
000087	0051	F5	READ:	PUSH AF
000088	0052	DBC4	READ1:	IN A,(PREAD)
000089	0054	CB7F	BIT	7,A
000090	0056	20FA	JR	NZ,READ1
000091	0058	F1	POP	AF
000092	0059	C9	RET	
000093	005A	0610	WRLINE:	LD B,10H
000094	005C	7E	TEST11:	LD A,(HL)
000095	005D	57	LD	D,A
000096	005E	CD8700	CALL	WRBYTE
000097	0061	23	INC	HL
000098	0062	10F8	DJNZ	TEST11
000099	0064	C9	RET	
000100	0065	7E	WRLINE1:	LD A,(HL)
000101	0066	57	LD	D,A
000102	0067	CD8700	CALL	WRBYTE
000103	006A	23	INC	HL
000104	006B	10F8	DJNZ	WRLINE1
000105	006D	C9	RET	
000106	006E			

```

000107 006E          ;*****
000108 006E          ;*   INITIAL LCD   *
000109 006E          ;*****
000110 006E 3E38      INITLCD:   LD    A,00111000B ;function set 38H
000111 0070          ;DL=1 8 bit,N=1 1/16 duty,F=0 5x7
000112 0070 D3C0      OUT    (PDATA),A
000113 0072 CD8E00    CALL   DELAY1
000114 0075 CD8E00    CALL   DELAY1      ;DELAY > 4.1 ms
000115 0078 3E0F      LD    A,00001111B ;display on/off control
000116 007A          ;D=1 off,C=1 cursor on,B=1 blink
000117 007A D3C0      OUT    (PDATA),A
000118 007C CD8E00    CALL   DELAY1
000119 007F 3E01      CLSLCD: LD    A,00000001B ;clear all display
000120 0081 D3C0      OUT    (PDATA),A
000121 0083 CD5100    CALL   READ
000122 0086 C9        RET
000123 0087
000124 0087          ;***** WRITE BYTE SUB *****
000125 0087 7A        WEBYTE:  LD    A,D
000126 0088 D3C2      OUT    (PSING),A
000127 008A CD5100    CALL   READ
000128 008D C9        RET
000129 008E
000130 008E          ;***** READY SUB *****
000131 008E C5        DELAY1:  PUSH  BC
000132 008F F5        PUSH  AF

```

```

000134 0092 10FE          DELAY12:  DJNZ  DELAY12
000135 0094 F1           POP    AF
000136 0095 C1           POP    BC
000137 0096 C9           RET
000138 0097
000139 0097                ;**** CHECK KEY RST BEFORE*****
000140 0097 3E91          RESTART:  LD    A,91H                ;SET PORTC UPPER IS O/P
000141 0099 D383                OUT   (PORTCT1),A                ;PORTC LOWER IS I/P
000142 009B 3E30          SCARST:  LD    A,30H                ;check key addr for start
000143 009D D382                OUT   (PORTC_KEY),A
000144 009F DB82                IN    A,(PORTC_KEY)
000145 00A1 E60F                AND   0FH
000146 00A3 FE0E                CP    0EH
000147 00A5 20F4                JR    NZ,SCARST
000148 00A7 CD6B02          CALL   DELAY
000149 00AA CD6B02          CALL   DELAY
000150 00AD CD7F00          SCAN_RST: CALL  CLSLCD
000151 00B0 3E00                LD    A,00H
000152 00B2 CD4900          CALL   GOTO
000153 00B5 210000          LD    HL,TAB_SHOW
000154 00B8 0610                LD    B,10H
000155 00BA CD6500          CALL   WRLINE1
000156 00BD 3E10                LD    A,10H
000157 00BF CD4900          CALL   GOTO
000158 00C2 1800                JR    START
000159 00C4

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

000160 00C4                ;***** MAIN *****
000161 00C4                ;***** START PROGRAM *****
000162 00C4 CD7902          START:    CALL  STARTUP
000163 00C7 DD21FC9F        LD      IX,09FFCH
000164 00CB DD2202F0        LD      (DATA_ADDR),IX
000165 00CF CDAD01          CALL  DIS_PAGE
000166 00D2 3E00           LD      A,00H
000167 00D4 CD4900          CALL  GOTO
000168 00D7 3E20           LD      A,20H
000169 00D9 0610           LD      B,10H
000170 00DB CDA302          START1:  CALL  WRITE
000171 00DE 10FB           DJNZ   START1
000172 00E0 3E44           LD      A,44H
000173 00E2 CD4900          CALL  GOTO
000174 00E5 FD2104F0        LD      IY,LINE
000175 00E9 FD7E00          LD      A,(IY+00H)
000176 00EC D601           SUB    01H
000177 00EE 27            DAA
000178 00EF FD7700          LD      (IY+00H),A
000179 00F2 FD7E01          LD      A,(IY+01H)
000180 00F5 DE00           SBC    A,00H
000181 00F7 27            DAA
000182 00F8 FD7701          LD      (IY+01H),A      ;LINE = 001
000183 00FB DD2100A0        LD      IX,0A000H
000184 00FF DD2202F0        LD      (DATA_ADDR),IX
000185 0103 CD5100          ST:     CALL  READ
000186 0106 DBC4           IN      A,(PREAD)

```

```

000187 0108 CBBF      RES    7,A           ;BIT 7 TO 0
000188 010A 57        LD     D,A           ;REG "D" SAVE ADDR DD RAM
000189 010B E60F      AND    0FH
000190 010D FE00      CP     00H
000191 010F 2005      JR     NZ,STCON0
000192 0111 CD3901     CALL  ST_LINE
000193 0114 18ED      JR     ST
000194 0116 CD3903     STCON0: CALL  KEY
000195 0119 3A01F0     LD     A,(KEY_DATA)
000196 011C CB6F      BIT    5,A           ;CP BIT 5 OF A (20H-28H)
000197 011E 2808      JR     Z,STCON1
000198 0120 CD3004     CALL  ST_FUNC
000199 0123 CD6B02     CALL  DELAY
000200 0126 18DB      JR     ST
000201 0128 7A        STCON1: LD     A,D
000202 0129 E60F      AND    0FH
000203 012B FE04      CP     04H           ;ADDRESS CURSOR OF COMMAN
000204 012D 2005      JR     NZ,STCON2     ;CALL ST_COM
000205 012F CDF201     CALL  ST_COM
000206 0132 18CF      JR     ST
000207 0134 CD4C02     STCON2: CALL  ST_OPER
000208 0137 18CA      JR     ST
000209 0139
000210 0139           ;***** CHECK LINE (ST_LINE). *****
000211 0139 DBC4      ST_LINE: IN    A,(PREAD)
000212 013B CBBF      RES    7,A
000213 013D 57        LD     D,A           ;SAVE ADDR DD RAM IN "D"

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้เผยแพร่ไปใช้ประโยชน์ทางการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

000214 013E 7A          L_CON:  LD  A,D          ;LD ADDRESS DD RAM TO "A"
000215 013F FE50          CP  50H
000216 0141 2005          JR  NZ,L_CONO
000217 0143 CD6C01        CALL END_PAGE          ;IF END OF LINE 2
000218 0146 180A          JR  L_ED
000219 0148 FE10          L_CONO: CP  10H
000220 014A C0            RET  NZ
000221 014B 3EC0          LD  A,0C0H          ;SET CURSOR TO 40H (OF D
000222 014D D3C0          L_END:  OUT  (PDATA),A
000223 014F CD5100        CALL READ
000224 0152 FD2104F0      L_ED:  LD  IY,LINE          ;DISPLAY LINE NUMBER+1 0
000225 0156 FD7E00          LD  A,(IY+00H)
000226 0159 C601          ADD  A,01H
000227 015B 27            DAA
000228 015C FD7700          LD  (IY+00H),A
000229 015F FD7E01          LD  A,(IY+01H)
000230 0162 CE00          ADC  A,00H
000231 0164 27            DAA
000232 0165 FD7701          LD  (IY+01H),A
000233 0168 CDA902        CALL DISLINE
000234 016B C9            L_RET:  RET
000235 016C E5            END_PAGE:  PUSH HL
000236 016D DDE5          PUSH IX
000237 016F 2A02F0        LD  HL,(DATA_ADDR)  ;SAVE (DATA_ADDR)
000238 0172 2206F0        LD  (BUFF_DATA_ADDR),HL
000239 0175 DD2A04F0      LD  IX,(LINE)        ;SAVE (LINE)
000240 0179 DD2208F0      LD  (BUFF_LINE),IX

```

```

000241 017D AF          SUB_DATA_ADDR: XOR  A          ;CLEAR CARRY FLAG
000242 017E 7D          LD    A,L          ;L = 2 BYTE LOW OF DATA_A
000243 017F D604        SUB    04H          ;DEC DATA_ADDR 04H (DATA_
000244 0181 6F          LD    L,A          ;SAVE TO L
000245 0182 3802        JR    C,SUB_CARRY
000246 0184 1804        JR    DIS_ENDPAGE
000247 0186 7C          SUB_CARRY: LD    A,H
000248 0187 D601        SUB    01H
000249 0189 67          LD    H,A          ;HL = DATA_ADDR - 4
000250 018A 2202F0       DIS_ENDPAGE: LD    (DATA_ADDR),HL
000251 018D DD2A02F0     LD    IX,(DATA_ADDR) ;DATA_ADDR - 4
000252 0191 CDAD01       CALL  DIS_PAGE
000253 0194 3E40        LD    A,40H        ;CURSOR TO FIRST LINE 2 FOR CALL
000254 0196 CD4900       CALL  GOTO          ;FOR DISP LINE NUMBER DOUBLE WIT
000255 0199 DD2A06F0     LD    IX,(BUFF_DATA_ADDR) ;DATA_ADDR PRESEN
000256 019D DD2202F0     LD    (DATA_ADDR),IX
000257 01A1 DD2A08F0     LD    IX,(BUFF_LINE)
000258 01A5 DD2204F0     LD    (LINE),IX
000259 01A9 DDE1        POP  IX
000260 01AB E1          POP  HL
000261 01AC C9          RET
000262 01AD
000263 01AD          ;*****DISPLAY 1 PAGE *****
000264 01AD DDE5        DIS_PAGE:  PUSH  IX
000265 01AF CD6E00       CALL  INITLCD      ;CURSOR TO HOME
000266 01B2 1E02        LD    E,02H
000267 01B4 DD2A02F0     LD    IX,(DATA_ADDR)

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

000268 01B8 CDA902          PAGE_2LINE: CALL DISLINE
000269 01BB CDDE02          CALL DISCOM
000270 01BE DD23            INC IX
000271 01C0 DD2202F0        LD (DATA_ADDR),IX
000272 01C4 0603            LD B,03H
000273 01C6 CDD102          OPER3_DIGIT: CALL DISOPER
000274 01C9 DD23            INC IX
000275 01CB DD2202F0        LD (DATA_ADDR),IX
000276 01CF 10F5            DJNZ OPER3_DIGIT
000277 01D1 3E40            LD A,40H
000278 01D3 CD4900          CALL GOTO
000279 01D6 FD2104F0        LD IY,LINE
000280 01DA FD7E00          LD A,(IY+00H)
000281 01DD C601            ADD A,01H ;ADD LINE -->LINE+1
000282 01DF 27              DAA
000283 01E0 FD7700          LD (IY+00H),A
000284 01E3 FD7E01          LD A,(IY+01H)
000285 01E6 CE00            ADC A,00H
000286 01E8 27              DAA
000287 01E9 FD7701          LD (IY+01H),A
000288 01EC 1D              DEC E
000289 01ED 20C9            JR NZ,PAGE_2LINE
000290 01EF DDE1            POP IX
000291 01F1 C9              RET
000292 01F2
000293 01F2                ;***** IF CURSOR IS ON COMMAND POSITION *****
000294 01F2 DDE5            ST_COM: PUSH IX

```

```

000295 01F4 3A01F0      LD    A, (KEY_DATA)
000296 01F7 CB77        BIT   6,A
000297 01F9 282F        JR    Z,NO_COM
000298 01FB DD2A02F0     LD    IX, (DATA_ADDR)
000299 01FF DD7700      LD    (IX+00H),A
000300 0202 CDDE02      CALL DISCOM
000301 0205 CD6B02      CALL DELAY
000302 0208 3A01F0     LD    A, (KEY_DATA)
000303 020B FE50        CP    50H           ;CHECK COMMAND NO OPER "A"
000304 020D 2005        JR    NZ,C_CONO
000305 020F CD2D02     CALL COM_NO_OPER
000306 0212 1810        JR    C_CON2
000307 0214 FE51        C_CONO: CP    51H           ;CHECK KEY "ORB"
000308 0216 2005        JR    NZ,C_CON1
000309 0218 CD2D02     CALL COM_NO_OPER
000310 021B 1807        JR    C_CON2
000311 021D FE52        C_CON1: CP    52H           ;CHECK KEY "END"
000312 021F 2003        JR    NZ,C_CON2
000313 0221 CD2D02     CALL COM_NO_OPER
000314 0224 DD23        C_CON2: INC    IX
000315 0226 DD2202F0    LD    (DATA_ADDR), IX
000316 022A DDE1        NO_COM: POP    IX
000317 022C C9          RET
000318 022D DD23        COM_NO_OPER: INC  IX
000319 022F 3E20        LD    A,20H           ;GET SPACE TO OPER
000320 0231 DD7700      LD    (IX+00H),A
000321 0234 DD23        INC    IX

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

000322 0236 3E20	LD	A, 20H
000323 0238 DD7700	LD	(IX+00H), A
000324 023B DD23	INC	IX
000325 023D 3E20	LD	A, 20H
000326 023F DD7700	LD	(IX+00H), A
000327 0242 CDD102	CALL	DISOPER
000328 0245 CDD102	CALL	DISOPER
000329 0248 CDD102	CALL	DISOPER
000330 024B C9	RET	
000331 024C		
000332 024C	;***** CURSOR IS ON OPERAND POSITION *****	
000333 024C DDE5	ST_OPER:	PUSH IX
000334 024E 3A01F0	LD	A, (KEY_DATA)
000335 0251 CB77	BIT	6, A
000336 0253 2013	JR	NZ, NO_NUM
000337 0255 DD2A02F0	LD	IX, (DATA_ADDR)
000338 0259 DD7700	LD	(IX+00H), A
000339 025C CDD102	CALL	DISOPER
000340 025F CD6B02	CALL	DELAY
000341 0262 DD23	INC	IX
000342 0264 DD2202F0	LD	(DATA_ADDR), IX
000343 0268 DDE1	NO_NUM:	POP IX
000344 026A C9	RET	
000345 026B		
000346 026B	;***** DELAY FOR KEY *****	
000347 026B C5	DELAY:	PUSH BC
000348 026C 0625	LD	B, 25H

```

000349 026E 0EFF          DELAY2:  LD    C,0FFH
000350 0270 00          DELAY3:  NOP
000351 0271 00          NOP
000352 0272 0D          DEC    C
000353 0273 20FB          JR     NZ,DELAY3
000354 0275 10F7          DJNZ  DELAY2
000355 0277 C1          POP   BC
000356 0278 C9          RET
000357 0279
000358 0279          ;***** SET FOR START *****
000359 0279 AF          STARTUP: XOR   A
000360 027A 3A10F0        LD    A,(FIRST_START)
000361 027D FE55          CP    55H
000362 027F 2815          JR     Z,NOFIRST ;SET DAT FOR CHECK FIRST
000363 0281 3E55          LD    A,55H
000364 0283 3210F0        LD    (FIRST_START),A
000365 0286 2100A0        LD    HL,0A000H
000366 0289 3E20          LD    A,20H ;20H=SPACE
000367 028B 0610          LD    B,10H
000368 028D 0EFF          STLOOP: LD    C,0FFH
000369 028F 77          STLOOP1: LD    (HL),A ;FILL SPACE AT DATA
000370 0290 0D          DEC    C
000371 0291 23          INC    HL
000372 0292 20FB          JR     NZ,STLOOP1
000373 0294 10F7          DJNZ  STLOOP
000374 0296 2100A0        NOFIRST: LD    HL,0A000H ;SET DATA_ADDR AT A000H
000375 0299 2202F0        LD    (DATA_ADDR),HL

```

```

000376 029C 210000          LD    HL,0000H
000377 029F 2204F0          LD    (LINE),HL
000378 02A2 C9              RET
000379 02A3
000380 02A3                  ;***** WRITE ASCII TO LCD *****
000381 02A3 D3C2          WRITE:  OUT  (PSING),A
000382 02A5 CD5100        CALL  READ
000383 02A8 C9              RET
000384 02A9
000385 02A9                  ;*****
000386 02A9                  ; DISPLAY LINE NUMBER
000387 02A9                  ;*****
000388 02A9 C5          DISLINE:  PUSH  BC
000389 02AA ED4B04F0      LD    BC,(LINE)
000390 02AE 78              LD    A,B
000391 02AF E60F          AND   0FH
000392 02B1 C630          ADD   A,30H          ;30H=ASCII OF "0"
000393 02B3 CDA302        CALL  WRITE          ;DISPLAY FIRST COL OF LIN
000394 02B6 79              LD    A,C
000395 02B7 07              RLCA
000396 02B8 07              RLCA
000397 02B9 07              RLCA
000398 02BA 07              RLCA          ;SWAP COL 2 TO 4BIT LOW
000399 02BB E60F          AND   0FH
000400 02BD C630          ADD   A,30H
000401 02BF CDA302        CALL  WRITE          ;DISPLAY COL 2 OF LINE NU
000402 02C2 79              LD    A,C

```

```

000403 02C3 E60F      AND   OFH
000404 02C5 C630      ADD   A,30H
000405 02C7 CDA302     CALL  WRITE
000406 02CA 3E20      LD    A,20H           ;DISPLAY SPACE 1 COLOUME
000407 02CC CDA302     CALL  WRITE
000408 02CF C1        POP   BC
000409 02D0 C9        RET
000410 02D1
000411 02D1           ;***** DISPLAY OPERAND *****
000412 02D1           ; INPUT = NUMBER OPERAND IN REG IX
000413 02D1           ;*****
000414 02D1 DD7E00     DISOPER: LD    A,(IX+00H)
000415 02D4 FE20      CP    20H           ;CP FOR SPACE
000416 02D6 2802      JR    Z,OPERSP
000417 02D8 C630      OPERCON: ADD   A,30H           ;ADD BY ASCII CHARACTOR "
000418 02DA CDA302     OPERSP: CALL  WRITE
000419 02DD C9        RET
000420 02DE
000421 02DE           ;***** DISPLAY COMMAND *****
000422 02DE           ; INPUT = COMMAND NUMBER IN REG IX
000423 02DE           ;*****
000424 02DE DD7E00     DISCOM: LD    A,(IX+00)
000425 02E1 FE20      CP    20H           ;IF (DATA_ADDR) = 20H IS
000426 02E3 2010      JR    NZ,DISCOM1
000427 02E5 21450A     LD    HL,TAB_OPERAND
000428 02E8 017000     LD    BC,70H           ;0EH*8 GO TAB "
000429 02EB 09        ADD   HL,BC

```

```

000430 02EC CD3303          CALL WRT_KEY
000431 02EF 3E20           LD   A,20H
000432 02F1 CDA302          CALL WRITE
000433 02F4 C9             RET
000434 02F5 FE50           DISCOM1: CP   50H
000435 02F7 DA1803          JP   C,OPERAND
000436 02FA C3FD02          JP   NO_OPERAND
000437 02FD DD7E00          NO_OPERAND: LD  A,(IX+00) ;LD DATA FROM DATA_ADDR
000438 0300 E60F           AND  0FH
000439 0302 4F             LD  C,A
000440 0303 212D0A          LD  HL,TAB_NO_OPERAND
000441 0306 CB11           RL  C
000442 0308 CB11           RL  C
000443 030A CB11           RL  C ;MUL C BY #8
000444 030C 0600           LD  B,00H
000445 030E 09            ADD  HL,BC
000446 030F CD3303          CALL WRT_KEY
000447 0312 3E20           LD  A,20H ;ASCII SPACE
000448 0314 CDA302          CALL WRITE
000449 0317 C9             RET
000450 0318 DD7E00          OPERAND: LD  A,(IX+00)
000451 031B E60F           AND  0FH
000452 031D 4F             LD  C,A
000453 031E 21450A          LD  HL,TAB_OPERAND
000454 0321 0600           LD  B,00H
000455 0323 CB11           RL  C
000456 0325 CB11           RL  C

```

```

000457 0327 CB11      RL    C
000458 0329 09       ADD   HL,BC
000459 032A CD3303    CALL  WRT_KEY
000460 032D 3E20     LD    A,20H           ;ASCII SPACE
000461 032F CDA302    CALL  WRITE
000462 0332 C9       RET
000463 0333 0608     WRT_KEY: LD    B,08H
000464 0335 CD5C00    CALL  TEST11
000465 0338 C9       RET
000466 0339
000467 0339          ;*****
000468 0339          ;***** SCAN KEY *****
000469 0339          ;*****
000470 0339 0608     KEY:   LD    B,08H           ; loop colume counter
000471 033B 1E00     LD    E,00H           ; start count
000472 033D 7B       SCAN1: LD    A,E
000473 033E D382     OUT   (PORTC_KEY),A
000474 0340 DB82     IN    A,(PORTC_KEY)
000475 0342 E60F     AND   OFH
000476 0344 FE0F     CP    OFH
000477 0346 2008     JR    NZ,SCAN2        ; jump if key press
000478 0348 7B       LD    A,E
000479 0349 CE10     ADC   A,010H
000480 034B 5F       LD    E,A
000481 034C 10EF     DJNZ  SCAN1
000482 034E 18E9     JR    KEY
000483 0350 B3       SCAN2: OR    E,SCAN1

```

เอกสารนี้เป็นลิขสิทธิ์ของสถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ไม่อนุญาติให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

000484 0351 3200F0          LD   (KEYIN),A
000485 0354 CD6B02          CALL DELAY
000486 0357 CD6B02          CALL DELAY
000487 035A                ;DEBOUND:  PUSH AF
000488 035A                ;DEBOUND1: IN   A,(PORTC_KEY)
000489 035A                ;          AND   OFH
000490 035A                ;          CP    OFH
000491 035A                ;          JR    NZ,DEBOUND1
000492 035A                ;          POP   AF
000493 035A
000494 035A                ;*****
000495 035A                ;          SUB  KEYCODE
000496 035A                ;*****
000497 035A 3A00F0          KEYCODE:  LD   A,(KEYIN)
000498 035D E6F0          AND   OFOH
000499 035F 07          RLCA
000500 0360 07          RLCA
000501 0361 07          RLCA
000502 0362 07          RLCA
000503 0363 47          LD   B,A
000504 0364 3A00F0          LD   A,(KEYIN)
000505 0367 E60F          AND   OFH
000506 0369 FE07          CP    07H
000507 036B CA7D03          JP    Z,KLIN1
000508 036E FE0B          CP    0BH
000509 0370 CA8403          JP    Z,KLIN2
000510 0373 FE0D          CP    0DH

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

000511 0375 CA8B03          JP    Z, KLIN3
000512 0378 FE0E           CP    0EH
000513 037A CA9203          JP    Z, KLIN4
000514 037D 210DOA          KLIN1: LD    HL, LINE1
000515 0380 CD9903          CALL  FIND_KEY
000516 0383 C9             RET
000517 0384 21150A          KLIN2: LD    HL, LINE2
000518 0387 CD9903          CALL  FIND_KEY
000519 038A C9             RET
000520 038B 211DOA          KLIN3: LD    HL, LINE3
000521 038E CD9903          CALL  FIND_KEY
000522 0391 C9             RET
000523 0392 21250A          KLIN4: LD    HL, LINE4
000524 0395 CD9903          CALL  FIND_KEY
000525 0398 C9             RET
000526 0399 48             FIND_KEY: LD    C, B
000527 039A 0600           LD    B, 00H
000528 039C 09             ADD   HL, BC
000529 039D 7E             LD    A, (HL)
000530 039E 3201F0          LD    (KEY_DATA), A
000531 03A1 CD6B02          CALL  DELAY
000532 03A4 C9             RET
000533 03A5
000534 03A5                ;***** FUNC_NOT *****
000535 03A5 DDE5          FUNC_NOT: PUSH  IX
000536 03A7 DD2A02F0        LD    IX, (DATA_ADDR)
000537 03AB AF            XOR   A                ;CLR CARRY FLAG

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

000538	03AC	DD2B	DEC	IX	;REDUCE DATA_ADDR (- 1)
000539	03AE	DD7E00	LD	A, (IX+00H)	;GET OLD DATA
000540	03B1	FE40	CP_LD:	CP	40H ;CHECK COMMAND "LD"
000541	03B3	2007	JR	NZ, CP_LD_NOT	
000542	03B5	3E49	LD	A, 49H	;CHANGE TO "LD-NOT"
000543	03B7	DD7700	LD	(IX+00H), A	
000544	03BA	1863	JR	FUN_NOT_CON	
000545	03BC	FE49	CP_LD_NOT:	CP	49H ;CHECK COMMAND "LD-NOT"
000546	03BE	2007	JR	NZ, CP_AND	
000547	03C0	3E40	LD	A, 40H	;CHANGE TO "LD"
000548	03C2	DD7700	LD	(IX+00H), A	
000549	03C5	1858	JR	FUN_NOT_CON	
000550	03C7	FE41	CP_AND:	CP	41H ;CHECK COMMAND "AND"
000551	03C9	2007	JR	NZ, CP_AND_NOT	
000552	03CB	3E4A	LD	A, 4AH	
000553	03CD	DD7700	LD	(IX+00H), A	
000554	03D0	184D	JR	FUN_NOT_CON	
000555	03D2	FE4A	CP_AND_NOT:	CP	4AH
000556	03D4	2007	JR	NZ, CP_OR	
000557	03D6	3E41	LD	A, 41H	
000558	03D8	DD7700	LD	(IX+00H), A	
000559	03DB	1842	JR	FUN_NOT_CON	
000560	03DD	FE42	CP_OR:	CP	42H
000561	03DF	2007	JR	NZ, CP_OR_NOT	
000562	03E1	3E4B	LD	A, 4BH	
000563	03E3	DD7700	LD	(IX+00H), A	
000564	03E6	1837	JR	FUN_NOT_CON	

000565	03E8	FE4B	CP_OR_NOT:	CP	4BH
000566	03EA	2007		JR	NZ, CP_TIM
000567	03EC	3E42		LD	A, 42H
000568	03EE	DD7700		LD	(IX+00H), A
000569	03F1	18B2		JR	FUNC_NOT_CON
000570	03F3	FE43	CP_TIM:	CP	43H
000571	03F5	2007		JR	NZ, CP_TIM_NOT
000572	03F7	3E4C		LD	A, 4CH
000573	03F9	DD7700		LD	(IX+00H), A
000574	03FC	1821		JR	FUN_NOT_CON
000575	03FE	FE4C	CP_TIM_NOT:	CP	4CH
000576	0400	2007		JR	NZ, CP_CNT
000577	0402	3E43		LD	A, 43H
000578	0404	DD7700		LD	(IX+00H), A
000579	0407	1816		JR	FUN_NOT_CON
000580	0409	FE44	CP_CNT:	CP	44H
000581	040B	2007		JR	NZ, CP_CNT_NOT
000582	040D	3E4D		LD	A, 4DH
000583	040F	DD7700		LD	(IX+00H), A
000584	0412	180B		JR	FUN_NOT_CON
000585	0414	FE4D	CP_CNT_NOT:	CP	4DH
000586	0416	200F		JR	NZ, FUN_NO_NOT
000587	0418	3E44		LD	A, 44H
000588	041A	DD7700		LD	(IX+00H), A
000589	041D	1800		JR	FUN_NOT_CON
000590	041F	3E44	FUN_NOT_CON:	LD	A, 44H

เอกสารนี้เป็นทรัพย์สินของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี ไม่ควรเผยแพร่โดยไม่ได้รับอนุญาตให้ นำไปใช้ในเชิงพาณิชย์

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

000592 0424 CDDE02          CALL DISCOM
000593 0427 DD23           FUN_NO_NOT: INC IX
000594 0429 DD2202F0       LD (DATA_ADDR), IX
000595 042D DDE1          POP IX
000596 042F C9           RET
000597 0430
000598 0430           ;*****
000599 0430
000600 0430 FE29          ST_FUNC: CP 29H
000601 0432 CCA503       CALL Z, FUNC_NOT
000602 0435 FE23          CP 23H
000603 0437 CC5E04       CALL Z, FUNC_ADDR
000604 043A FE22          CP 22H
000605 043C CCF705       CALL Z, FUNC_DEL
000606 043F FE25          CP 25H
000607 0441 CC5D08       CALL Z, FUNC_DOWN
000608 0444 FE26          CP 26H
000609 0446 CC6107       CALL Z, FUNC_UP
000610 0449 FE21          CP 21H
000611 044B CCF608       CALL Z, FUNC_INS
000612 044E FE24          CP 24H
000613 0450 CCCD0A       CALL Z, FUNC_GO
000614 0453 FE28          CP 28H
000615 0455 CC9D09       CALL Z, FUNC_RIGHT
000616 0458 FE27          CP 27H
000617 045A CCD509       CALL Z, FUNC_LEFT
000618 045D C9           RET

```

```

000619 045E
000620 045E ;*****FUNCTION ADDRESS *****
000621 045E 00 FUNC_ADDR: NOP
000622 045F CD7F00 FA_NEW: CALL CLSLCD
000623 0462 CD5100 CALL READ ;READ BUSY FLAG
000624 0465 FD21BDOA LD IY,TAB_ADDR
000625 0469 0610 LD B,10H
000626 046B FD7E00 FA_WRITE: LD A,(IY+00H)
000627 046E CDA302 CALL WRITE ;DISPLAY "-NUMBER ADDRESS
000628 0471 FD23 INC IY
000629 0473 10F6 DJNZ FA_WRITE
000630 0475 3EC6 LD A,0C6H
000631 0477 D3C0 OUT (PDATA),A ;GET CURSOR TO POSITION 0
000632 0479 CD5100 CALL READ
000633 047C CD6B02 CALL DELAY
000634 047F CD3903 FA_L2: CALL KEY
000635 0482 3A01F0 LD A,(KEY_DATA)
000636 0485 FE23 CP 23H ;CHECK KEY ADDR AGAIN
000637 0487 28D6 JR Z,FA_NEW
000638 0489 FE25 FA_CON0: CP 25H ;CHECK KEY DOWN FOR "GO"
000639 048B 283A JR Z,FA_GO
000640 048D 3A01F0 FA_CON1: LD A,(KEY_DATA)
000641 0490 E6F0 AND 0F0H
000642 0492 FE00 CP 00H ;CHECK
000643 0494 20E9 JR NZ,FA_L2 ;NOT NUMBER KEY GO RECIVE
000644 0496 DBC4 IN A,(PREAD) ;CHECK ADDRESS-DD RAM OF
000645 0498 CBBF RES 7,A ;FOR CHECK COLUMN OF NUMB

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ทางการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

000646	049A	FE49	CP	49H		;MORE 3 COLUMN?
000647	049C	201C	JR	NZ,FA_CON2		
000648	049E	3EC6	LD	A,0C6H		;MORE 3 COLUMN REAL
000649	04A0	D3C0	OUT	(PDATA),A		
000650	04A2	CD5100	CALL	READ		
000651	04A5	0603	LD	B,03H		
000652	04A7	3E20	FA_CLS:	LD	A,20H	;FILL SPACE TO NUMBER 3 C
000653	04A9	CDA302	CALL	WRITE		
000654	04AC	10F9	DJNZ	FA_CLS		
000655	04AE	3EC6	LD	A,0C6H		;GET CURSOR TO FIRST NUM
000656	04B0	D3C0	OUT	(PDATA),A		
000657	04B2	CD5100	CALL	READ		
000658	04B5	CD6B02	CALL	DELAY		
000659	04B8	18C5	JR	FA_L2		
000660	04BA	3A01F0	FA_CON2:	LD	A,(KEY_DATA)	
000661	04BD	C630	ADD	A,30H		
000662	04BF	CDA302	CALL	WRITE		
000663	04C2	CD6B02	CALL	DELAY		
000664	04C5	18B8	JR	FA_L2		
000665	04C7	0EC6	FA_GO:	LD	C,0C6H	;SAVE FIRST CURSOR POSIT
000666	04C9	0603	LD	B,03H		;COUNTER FOR LOOP
000667	04CB	FD210AF0	LD	IY,ADDR_NUM		
000668	04CF	FD360000	LD	(IY+00H),00H		
000669	04D3	FD360100	LD	(IY+01H),00H		
000670	04D7	79	FA_GOO:	LD	A,C	;USE CURSOR POSITION
000671	04D8	D3C0	OUT	(PDATA),A		;GOTO CURSOR POSITION
000672	04DA	CD5100	CALL	READ		

```

000673 04DD DBC6      FA_G01:   IN    A,(PREAD_DATA) ;READ DATA AT CURSOR POIN
000674 04DF CB7F      BIT    7,A           ;CHECK BIT 7 IF "1" JUMP
000675 04E1 20F4      JR     NZ,FA_G00
000676 04E3 FE20      CP     20H          ;CP SPACE " "
000677 04E5 2820      JR     Z,FA_G03
000678 04E7 07        FA_G01.1: RLCA           ;SWAP "A"
000679 04E8 07        RLCA
000680 04E9 07        RLCA
000681 04EA 07        RLCA
000682 04EB 57        LD     D,A          ;BUFFER
000683 04EC 1E04      LD     E,04H        ;COUNTER FOR LOOP
000684 04EE AF         XOR    A            ;CLEAR CARRY FLAG
000685 04EF DD210AF0   LD     IX,ADDR_NUM
000686 04F3 7A        FA_G02:   LD     A,D
000687 04F4 17        RLA              ;SHIFT LEFT PASS C FLAG
000688 04F5 57        LD     D,A
000689 04F6 DD7E00   LD     A,(IX+00H)  ;LD     A,(ADDR_NUM)
000690 04F9 17        RLA
000691 04FA DD7700   LD     (IX+00H),A
000692 04FD DD7E01   LD     A,(IX+01H)  ;LD     A,(ADDR_NUM+01)
000693 0500 17        RLA
000694 0501 DD7701   LD     (IX+01H),A
000695 0504 1D        DEC    E
000696 0505 20EC      JR     NZ,FA_G02
000697 0507 0C        FA_G03:   INC    C           ;INC CURSOR=CURSOR + 1
000698 0508 10CD      DJNZ  FA_G00
000699 050A DD2104F0   LD     IX,LINE

```

000700	050E	DD360001	LD	(IX+00H),01H	
000701	0512	DD360100	LD	(IX+01H),00H	;SET LINE 001H
000702	0516	FD2102F0	LD	IY,DATA_ADDR	
000703	051A	FD360000	LD	(IY+00H),00H	
000704	051E	FD3601A0	LD	(IY+01H),0A0H	;SET DATA_ADDR A000H
000705	0522	DD2104F0	FA_G04:	LD	IX,LINE
000706	0526	DD7E01	LD	A,(IX+01H)	;LD A,LINE+1
000707	0529	FD210AF0	LD	IY,ADDR_NUM	
000708	052D	FD4E01	LD	C,(IY+01H)	;LD C,ADDR_NUM+1
000709	0530	B9	CP	C	
000710	0531	200B	JR	NZ,FA_G05	
000711	0533	DD7E00	LD	A,(IX+00H)	;LD A,LINE
000712	0536	FD4E00	LD	C,(IY+00H)	;LD C,ADDR_NUM
000713	0539	B9	CP	C	
000714	053A	2002	JR	NZ,FA_G05	
000715	053C	182C	JR	FA_G06	;IF SAME ADDR_NUM+LINE
000716	053E	DD2102F0	FA_G05:	LD	IX,DATA_ADDR
000717	0542	DD7E00	LD	A,(IX+00H)	;ADD DATA_ADDR + 4
000718	0545	C604	ADD	A,04H	
000719	0547	DD7700	LD	(IX+00H),A	
000720	054A	DD7E01	LD	A,(IX+01H)	
000721	054D	CE00	ADC	A,00H	
000722	054F	DD7701	LD	(IX+01H),A	
000723	0552	FD2104F0	LD	IY,LINE	
000724	0556	FD7E00	LD	A,(IY+00H)	;INC LINE
000725	0559	C601	ADD	A,01H	
000726	055B	27	DAA		

```

000727 055C FD7700          LD    (IY+00H),A
000728 055F FD7E01          LD    A,(IY+01H)
000729 0562 CE00           ADC    A,00H
000730 0564 27             DAA
000731 0565 FD7701          LD    (IY+01H),A
000732 0568 18B8          JR    FA_GO4
000733 056A                ;*****
000734 056A                ; DEC LINE PRESENT ,SUB DATA_ADDE BY 04H
000735 056A                ; FOR DIS_PAGE
000736 056A                ; AND GET DATA_ADDR TO LINE
000737 056A                ;*****
000738 056A FD2104F0        FA_GO6: LD    IY,LINE
000739 056E FD7E01          LD    A,(IY+01H)
000740 0571 FE00           CP    00H
000741 0573 200A          JR    NZ,FA_GO7
000742 0575 FD7E00          LD    A,(IY+00H)
000743 0578 FE00           CP    00H
000744 057A 2003          JR    NZ,FA_GO7
000745 057C C35F04          JP    FA_NEW
000746 057F FD2A04F0        FA_GO7: LD    IY,(LINE)
000747 0583 FD2208F0          LD    (BUFF_LINE),IY ;SAVE LINE TO BUFFER
000748 0587 FD2104F0          LD    IY,LINE ;
000749 058B FD7E00          LD    A,(IY+00H) ;
000750 058E D601           SUB    01H ;
000751 0590 27             DAA ;
000752 0591 FD7700          LD    (IY+00H),A ;
000753 0594 FD7E01          LD    A,(IY+01H) ;

```

เอกสารนี้เป็นเอกสารที่จัดทำขึ้นไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

000754 0597 DE00          SBC  A,00H          ;
000755 0599 27           DAA                  ;
000756 059A FD7701       LD   (IY+01H),A     ;DEC LINE = LINE - 1
000757 059D DD2A02F0     LD   IX,(DATA_ADDR)
000758 05A1 DD2206F0     LD   (BUFF_DATA_ADDR),IX ;SAVE DATA_ADDR T
000759 05A5 DD2102F0     LD   IX,DATA_ADDR  ;
000760 05A9 DD7E00       LD   A,(IX+00H)    ;
000761 05AC D604         SUB  04H            ;
000762 05AE DD7700       LD   (IX+00H),A    ;
000763 05B1 DD7E01       LD   A,(IX+01H)    ;
000764 05B4 DE00          SBC  A,00H          ;
000765 05B6 DD7701       LD   (IX+01H),A    ;DATA_ADDR = DATA_ADDR -
000766 05B9                                     ;IX = DATA_ADDR
000767 05B9 FD7E00       FA_G07: LD  A,(IY+00H)

```

**** Warning **** attempt to redefine symbol

```

000768 05BC FE00          CP   00H
000769 05BE 201A         JR   NZ,FA_G09
000770 05C0 FD7E01       LD   A,(IY+01H)
000771 05C3 FE00          CP   00H
000772 05C5 2013         JR   NZ,FA_G09     ;IF LINE 000 DEL LINE 1
000773 05C7 CDAD01       CALL DIS_PAGE
000774 05CA 3E00          LD   A,00H
000775 05CC CD4900       CALL GOTO          ;CURSOR TO LINE 1
000776 05CF 0610         LD   B,10H
000777 05D1 3E20          LD   A,20H
000778 05D3 CDA302       FA_G08: CALL WRITE

```

```

000779 05D6 10FB          DJNZ  FA_G08
000780 05D8 1803          JR    FA_G010
000781 05DA CDAD01        FA_G09:  CALL  DIS_PAGE      ;AFTER DIS_PAGE
000782 05DD DD2A06F0      FA_G010: LD    IX,(BUFF_DATA_ADDR)
000783 05E1 DD2202F0      LD    (DATA_ADDR),IX ;GET IX TO DATA_ADDR
000784 05E5 3EC4          LD    A,0C4H          ;GET CURSOR TO COMMAND LI
000785 05E7 D3C0          OUT   (PDATA),A
000786 05E9 CD5100        CALL  READ
000787 05EC FD2A08F0      LD    IY,(BUFF_LINE)
000788 05F0 FD2204F0      LD    (LINE),IY      ;GET IY TO LINE
000789 05F4 C30301        FA_END:  JP    ST
000790 05F7
000791 05F7                ;***** FUNCTION DELEAT LINE *****
000792 05F7                ; SAVE ADDR DD RAM IN "C"
000793 05F7
000794 05F7 DDES          FUNC_DEL:  PUSH  IX
000795 05F9 FDES          PUSH  IY
000796 05FB AF            XOR    A
000797 05FC DD2A04F0      LD    IX,(LINE)
000798 0600 DD2208F0      LD    (BUFF_LINE),IX ;SAVE LINE
000799 0604 DD2A02F0      LD    IX,(DATA_ADDR)
000800 0608 DD2206F0      LD    (BUFF_DATA_ADDR),IX ;SAVE DATA_ADDR
000801 060C DD2102F0      LD    IX,DATA_ADDR   ;( IX = DATA_ADDR
000802 0610 FD210CF0      LD    IY,POINT       ;( IY = POINT )
000803 0614 DBC4          IN    A,(PREAD)      ;GET CURSOR POSIT
000804 0616 47            LD    B,A
000805 0617 CD5100        CALL  READ

```

เอกสารนี้เป็นเอกสารสงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

000806 061A 78          LD    A,B
000807 061B CBBF       RES    7,A          ;CLS BIT 7 (BUSY
000808 061D 320FF0     LD    (BUFF_CURSOR),A  ;SAVE ADDRESS CUR
000809 0620 E60F       AND    0FH          ;WANT COLUMN ONLY
000810 0622 FE04       CP     04H          ;IF COMMAND COLUM
000811 0624 2014       JR     NZ,FD_CON1   ;NOT COMMAND
000812 0626 AF         XOR    A
000813 0627 DD7E00     LD    A,(IX+00H)      ;
000814 062A C604       ADD    A,04H          ;
000815 062C FD7700     LD    (IY+00H),A      ;DATA_ADDR+4 IN P
000816 062F DD7E01     LD    A,(IX+01H)      ;
000817 0632 CE00       ADC    A,00H          ;
000818 0634 FD7701     LD    (IY+01H),A      ;FILL IN POINT
000819 0637 C37406     JP     FD_CUR
000820 063A FE0D       FD_CON1: CP    0DH          ;IF OPER 1 = +3
000821 063C 2014       JR     NZ,FD_CON2
000822 063E AF         XOR    A
000823 063F DD7E00     LD    A,(IX+00H)      ;ADD DATA_ADDR + 3
000824 0642 C603       ADD    A,03H
000825 0644 FD7700     LD    (IY+00H),A
000826 0647 DD7E01     LD    A,(IX+01H)
000827 064A CE00       ADC    A,00H
000828 064C FD7701     LD    (IY+01H),A      ;FILL IN POINT
000829 064F C37406     JP     FD_CUR
000830 0652 FE0E       FD_CON2: CP    0EH          ;IF OPER 2 = +2
000831 0654 2014       JR     NZ,FD_CON3
000832 0656 AF         XOR    A

```

000833	0657	DD7E00	LD	A, (IX+00H)	
000834	065A	C602	ADD	A, 02H	
000835	065C	FD7700	LD	(IY+00H), A	
000836	065F	DD7E01	LD	A, (IX+01H)	
000837	0662	CE00	ADC	A, 00H	
000838	0664	FD7701	LD	(IY+01H), A	
000839	0667	C37406	JP	FD_CUR	
000840	066A	DD2A02F0	FD_CON3:	LD	IX, (DATA_ADDR) ;OPER 3 = +1
000841	066E	DD23	INC	IX	
000842	0670	DD220CF0	LD	(POINT), IX	
000843	0674	DD210CF0	FD_CUR:	LD	IX, POINT ;IX = POINT
000844	0678	0604	FD_LOOP:	LD	B, 04H ;COUNTER
000845	067A	2A0CF0	FD_GET:	LD	HL, (POINT)
000846	067D	7E	LD	A, (HL)	
000847	067E	F5	PUSH	AF	
000848	067F	AF	XOR	A	
000849	0680	DD7E00	LD	A, (IX+00H)	
000850	0683	D604	SUB	04H	
000851	0685	DD7700	LD	(IX+00H), A	
000852	0688	DD7E01	LD	A, (IX+01H)	
000853	068B	DE00	SBC	A, 00H	
000854	068D	DD7701	LD	(IX+01H), A	;POINT - 4
000855	0690	F1	POP	AF	
000856	0691	2A0CF0	LD	HL, (POINT)	
000857	0694	77	LD	(HL), A	
000858	0695	DD7E00	LD	A, (IX+00H)	
000859	0698	C605	ADD	A, 05H	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

000860 069A DD7700      LD    (IX+00H),A
000861 069D DD7E01      LD    A,(IX+01H)
000862 06A0 CE00        ADC  A,00H
000863 06A2 DD7701      LD    (IX+01H),A      ;POINT + 5
000864 06A5 10D3        DJNZ  FD_GET
000865 06A7 FEC0        CP    0C0H            ;CP FOR TO ADDRESS C000H
000866 06A9 20CD        JR    NZ,FD_LOOP     ;MOVE LINE TO LAST LINE 8
000867 06AB
000868 06AB      ;*****
000869 06AB      ;SET LINE = LINE - 1 ; SET DATA_ADDR = DATA_ADDR - 4 FOR DI
000870 06AB
000871 06AB AF      FD_CON4:  KOR  A
000872 06AC FD2104F0  LD    IY,LINE
000873 06B0 FD7E00      LD    A,(IY+00H)
000874 06B3 D601      SUB  01H
000875 06B5 27        DAA
000876 06B6 FD7700      LD    (IY+00H),A
000877 06B9 FD7E01      LD    A,(IY+01H)
000878 06BC DE00        SBC  A,00H
000879 06BE 27        DAA
000880 06BF FD7701      LD    (IY+01H),A      ;LINE = LINE
000881 06C2 DD2A06F0     LD    IX,(BUFF_DATA_ADDR)
000882 06C6 DD2B        DEC  IX
000883 06C8 DD2B        DEC  IX
000884 06CA DD2B        DEC  IX
000885 06CC DD2B        DEC  IX
000886 06CE DD2202F0     LD    (DATA_ADDR),IX ;DATA_ADDR = DATA_ADDR -

```

```

000887 06D2 3A0FF0      LD    A,(BUFF_CURSOR) ;GET ADDR CURSOR FROM BUF
000888 06D5 E60F        AND   0FH
000889 06D7 DD2102F0     LD    IX,DATA_ADDR    ; IX = DATA_ADDR
000890 06DB FEOF          FD_COLO: CP    0FH      ;IF IS COLUMN OPERAND 3
000891 06DD 2013        JR    NZ,FD_COL1      ;
000892 06DF AF          XOR   A
000893 06E0 DD7E00     LD    A,(IX+00H)      ;DATA_ADDR - 3
000894 06E3 D603        SUB   03H
000895 06E5 DD7700     LD    (IX+00H),A
000896 06E8 DD7E01     LD    A,(IX+01H)
000897 06EB DE00        SBC   A,00H
000898 06ED DD7701     LD    (IX+01H),A
000899 06F0 182C        JR    FD_CHK
000900 06F2 FE0E          FD_COL1: CP    0EH      ;IF IS COLUMN OPERAND 2
000901 06F4 2013        JR    NZ,FD_COL2      ;
000902 06F6 AF          XOR   A
000903 06F7 DD7E00     LD    A,(IX+00H)      ; DATA_ADDR - 2
000904 06FA D602        SUB   02H
000905 06FC DD7700     LD    (IX+00H),A
000906 06FF DD7E01     LD    A,(IX+01H)
000907 0702 DE00        SBC   A,00H
000908 0704 DD7701     LD    (IX+01H),A
000909 0707 1815        JR    FD_CHK
000910 0709 FE0D          FD_COL2: CP    0DH      ;IF IS COLUMN OPERAND 1
000911 070B 2011        JR    NZ,FD_CHK      ;IF NZ IS COMMAND COLUMN
000912 070D AF          XOR   A

```

เอกสารนี้เป็นทรัพย์สินของสำนักงานเพื่อใช้ในการปฏิบัติงานเท่านั้น ไม่ควรเผยแพร่โดยไม่ได้รับอนุญาต

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

000914 0711 D601          SUB  01H
000915 0713 DD7700       LD   (IX+00H),A
000916 0716 DD7E01       LD   A,(IX+01H)
000917 0719 DE00         SBC  A,00H
000918 071B DD7701       LD   (IX+01H),A
000919 071E
000920 071E                ;IF CURSOR IS AT COMMAND POSITION NOT DEC DATA_ADDR
000921 071E
000922 071E FD2108F0      FD_CHK: LD  IY,BUFF_LINE ;CHECK IF LINE 001
000923 0722 FD7E00       LD   A,(IY+00H)
000924 0725 FE01         CP   01H
000925 0727 201A         JR   NZ,FD_END
000926 0729 FD7E01       LD   A,(IY+01H)
000927 072C FE00         CP   00H
000928 072E 2013         JR   NZ,FD_END
000929 0730 CDAD01       CALL DIS_PAGE
000930 0733 3E00         LD   A,00H
000931 0735 CD4900       CALL GOTO ;SET CURSOR TO LINE 1
000932 0738 0610        LD   B,10H
000933 073A 3E20         LD   A,20H
000934 073C CDA302      FD_CHK1: CALL WRITE ;DEL LINE 1
000935 073F 10FB        DJNZ FD_CHK1
000936 0741 1803        JR   FD_END1
000937 0743 CDAD01      FD_END: CALL DIS_PAGE
000938 0746 DD2A08F0    FD_END1: LD  IX,(BUFF_LINE)
000939 074A DD2204F0    LD   (LINE),IX
000940 074E FD2A06F0    LD   IY,(BUFF_DATA_ADDR)

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่ออนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

000941 0752 FD2202F0      LD   (DATA_ADDR), IY
000942 0756 3A0FF0        LD   A, (BUFF_CURSOR)      ;GET ADDR CURSO
000943 0759 CD4900        CALL GOTO
000944 075C FDE1          POP  IY
000945 075E DDE1          POP  IX
000946 0760 C9            RET
000947 0761
000948 0761                ;***** FUNC UP *****
000949 0761                ;DISPLAY LINE OLD 1 LINE
000950 0761                ;*****
000951 0761 DDE5          FUNC_UP:  PUSH IX
000952 0763 FDE5          PUSH IY
000953 0765 FD2A02F0      LD   IY, (DATA_ADDR)      ;SAVE DATA_ADDR I
000954 0769 FD2206F0      LD   (BUFF_DATA_ADDR), IY
000955 076D DD2A04F0      LD   IX, (LINE)
000956 0771 DD2208F0      LD   (BUFF_LINE), IX     ;SAVE LINE TO BUF
000957 0775 AF            XOR   A
000958 0776 DD2104F0      LD   IX, LINE             ;IF LINE 001 NO U
000959 077A DD7E01        LD   A, (IX+01H)
000960 077D E60F          AND  0FH
000961 077F FE00          CP   00H
000962 0781 200A          JR   NZ, FU_START
000963 0783 DD7E00        LD   A, (IX+00H)
000964 0786 FE01          CP   01H
000965 0788 2003          JR   NZ, FU_START
000966 078A C35808        JP   FU_END
000967 078D DBC4          FU_START: IN  A, (PREAD)

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

000968 078F 47          LD    B,A
000969 0790 CD5100      CALL  READ
000970 0793 78          LD    A,B
000971 0794 320FF0      LD    (BUFF_CURSOR),A      ;SAVE CURSOR TO B
000972 0797 AF          XOR   A
000973 0798 DD2104F0    LD    IX,LINE
000974 079C DD7E00      LD    A,(IX+00H)
000975 079F D602        SUB   02H
000976 07A1 27          DAA
000977 07A2 DD7700      LD    (IX+00H),A
000978 07A5 DD7E01      LD    A,(IX+01H)
000979 07A8 DE00        SBC  A,00H
000980 07AA 27          DAA
000981 07AB DD7701      LD    (IX+01H),A      ;LINE - 2
000982 07AE DD2A02F0    LD    IX,(DATA_ADDR)
000983 07B2 0608        LD    B,08H
000984 07B4 DD2B        UP_LOOP: DEC  IX
000985 07B6 10FC        DJNZ  UP_LOOP      ;DATA_ADDR - 8 FOR DIS_PA
000986 07B8 DD2202F0    LD    (DATA_ADDR),IX
000987 07BC AF          XOR   A
000988 07BD 3A0FF0      FU_CON1: LD    A,(BUFF_CURSOR)
000989 07C0 E60F        AND  0FH
000990 07C2 FE0D        CP   0DH      ;IF CURSOR AT OPER 1
000991 07C4 200C        JR   NZ,FU_CON2  ;DATA_ADDR - 1
000992 07C6 DD2A02F0    LD    IX,(DATA_ADDR)
000993 07CA DD2B        DEC  IX
000994 07CC DD2202F0    LD    (DATA_ADDR),IX

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

000995	07D0	1824		JR	FU_DISP	
000996	07D2	FE0E	FU_CON2:	CP	0EH	;IF CURSOR AT OPER 2
000997	07D4	200E		JR	NZ,FU_CON3	;DATA_ADDR - 2
000998	07D6	DD2A02F0		LD	IX,(DATA_ADDR)	
000999	07DA	DD2B		DEC	IX	
001000	07DC	DD2B		DEC	IX	
001001	07DE	DD2202F0		LD	(DATA_ADDR),IX	
001002	07E2	1812		JR	FU_DISP	
001003	07E4	FE0F	FU_CON3:	CP	0FH	;IF CURSOR AT OPER 3
001004	07E6	200E		JR	NZ,FU_DISP	;DATA_ADDR - 3
001005	07E8	DD2A02F0		LD	IX,(DATA_ADDR)	
001006	07EC	DD2B		DEC	IX	
001007	07EE	DD2B		DEC	IX	
001008	07F0	DD2B		DEC	IX	
001009	07F2	DD2202F0		LD	(DATA_ADDR),IX	
001010	07F6	FD2104F0	FU_DISP:	LD	IY,LINE	
001011	07FA	FD7E00		LD	A,(IY+00H)	
001012	07FD	FE00		CP	00H	;IF LINE 002 : PRESS KEY
001013	07FF	201A		JR	NZ,FU_DISP3	
001014	0801	FD7E01		LD	A,(IY+01H)	
001015	0804	FE00		CP	00H	
001016	0806	2013		JR	NZ,FU_DISP3	
001017	0808	CDAD01		CALL	DIS_PAGE	
001018	080B	3E00		LD	A,00H	
001019	080D	CD4900		CALL	GOTO	
001020	0810	0610		LD	B,10H	
001021	0812	3E20		LD	A,20H	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

001022 0814 CDA302      FU_DISP2:  CALL  WRITE
001023 0817 10FB              DJNZ  FU_DISP2
001024 0819 1803              JR    FU_DISP4
001025 081B CDAD01      FU_DISP3:  CALL  DIS_PAGE
001026 081E 3A0FF0      FU_DISP4:  LD    A,(BUFF_CURSOR)
001027 0821 CD4900              CALL  GOTO          ;GET-CURSOR TO COMMAND LI
001028 0824 AF              XOR   A
001029 0825 FD2106F0      LD    IY,BUFF_DATA_ADDR
001030 0829 DD2102F0      LD    IX,DATA_ADDR
001031 082D FD7E00      LD    A,(IY+00H)
001032 0830 D604              SUB   04H
001033 0832 DD7700      LD    (IX+00H),A
001034 0835 FD7E01      LD    A,(IY+01H)
001035 0838 DE00              SBC  A,00H
001036 083A DD7701      LD    (IX+01H),A      ;DATA_ADDR = DATA_ADDR OL
001037 083D AF              XOR   A
001038 083E FD2108F0      LD    IY,BUFF_LINE
001039 0842 DD2104F0      LD    IX,LINE
001040 0846 FD7E00      LD    A,(IY+00H)
001041 0849 D601              SUB   01H
001042 084B 27              DAA
001043 084C DD7700      LD    (IX+00H),A
001044 084F FD7E01      LD    A,(IY+01H)
001045 0852 DE00              SBC  A,00H
001046 0854 27              DAA
001047 0855 DD7701      LD    (IX+01H),A      ;LINE = LINE_OLD - 1
001048 0858 FDE1      FU_END:  POP  IY

```

```

001049 085A DDE1          POP   IX
001050 085C C9           RET
001051 085D
001052 085D          ;*****
001053 085D          ; FUNCTION DOWN DISPLAY LINE NEXT
001054 085D          ;*****
001055 085D DDE5          FUNC_DOWN: PUSH  IX
001056 085F FDE5          PUSH  IY
001057 0861 FD2A04F0       LD    IY,(LINE)
001058 0865 FD2208F0       LD    (BUFF_LINE),IY      ;SAVE LINE OLD
001059 0869 DD2A02F0       LD    IX,(DATA_ADDR)
001060 086D DD2206F0       LD    (BUFF_DATA_ADDR),IX ;SAVE DATA_ADDR OL
001061 0871 DBC4          IN    A,(PREAD)
001062 0873 47           LD    B,A
001063 0874 CD5100       CALL  READ
001064 0877 78           LD    A,B
001065 0878 320FF0       LD    (BUFF_CURSOR),A    ;SAVE CURSOR TO B
001066 087B 3A0FF0       FDWCON1: LD    A,(BUFF_CURSOR)
001067 087E E60F          AND   0FH
001068 0880 FE0D          CP    0DH                ;IF CURSOR ON OPE
001069 0882 200C          JR    NZ,FDWCON2        ;DATA_ADDR - 1
001070 0884 DD2A02F0       LD    IX,(DATA_ADDR)    ;FOR DIS_PAGE
001071 0888 DD2B          DEC   IX
001072 088A DD2202F0       LD    (DATA_ADDR),IX
001073 088E 1824          JR    FD_DISP
001074 0890 FE0E          FDWCON2: CP    0EH        ;IF CURSOR ON OPE
001075 0892 200E          JR    NZ,FDWCON3        ;DATA_ADDR - 2

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

001076	0894	DD2A02F0	LD	IX, (DATA_ADDR)	
001077	0898	DD2B	DEC	IX	
001078	089A	DD2B	DEC	IX	
001079	089C	DD2202F0	LD	(DATA_ADDR), IX	
001080	08A0	1812	JR	FD_DISP	
001081	08A2	FE0F	FDWCON3:	CP	OFH
001082	08A4	200E	JR	NZ, FD_DISP	
001083	08A6	DD2A02F0	LD	IX, (DATA_ADDR)	
001084	08AA	DD2B	DEC	IX	
001085	08AC	DD2B	DEC	IX	
001086	08AE	DD2B	DEC	IX	
001087	08B0	DD2202F0	LD	(DATA_ADDR), IX	
001088	08B4	CDAD01	FD_DISP:	CALL	DIS_PAGE
001089	08B7	3A0FF0	LD	A, (BUFF_CURSOR)	
001090	08BA	CD4900	CALL	GOTO	
001091	08BD	AF	XOR	A	
001092	08BE	FD2108F0	LD	IY, BUFF_LINE	
001093	08C2	DD2104F0	LD	IX, LINE	
001094	08C6	FD7E00	LD	A, (IY+00H)	
001095	08C9	C601	ADD	A, 01H	
001096	08CB	27	DAA		
001097	08CC	DD7700	LD	(IX+00H), A	
001098	08CF	FD7E01	LD	A, (IY+01H)	
001099	08D2	CE00	ADC	A, 00H	
001100	08D4	27	DAA		
001101	08D5	DD7701	LD	(IX+01H), A	;LINE = LINE OLD + 1
001102	08D8	AF	XOR	A	

```

001103 08D9 FD2106F0      LD    IY,BUFF_DATA_ADDR
001104 08DD DD2102F0      LD    IX,DATA_ADDR
001105 08E1 FD7E00        LD    A,(IY+00H)
001106 08E4 C604          ADD  A,04H
001107 08E6 DD7700        LD    (IX+00H),A
001108 08E9 FD7E01        LD    A,(IY+01H)
001109 08EC CE00          ADC  A,00H
001110 08EE DD7701        LD    (IX+01H),A      ;DATA_ADDR = DATA_ADDR OL
001111 08F1 FDE1          POP  IY
001112 08F3 DDE1          POP  IX
001113 08F5 C9           RET
001114 08F6
001115 08F6      ;*****
001116 08F6      ; FUNCTION INSERT
001117 08F6      ;*****
001118 08F6 DDE5      FUNC_INS:  PUSH  IX
001119 08F8 FDE5      PUSH  IY
001120 08FA FD2A04F0    LD    IY,(LINE)
001121 08FE FD2208F0    LD    (BUFF_LINE),IY ;SAVE LINE TO BUFFER
001122 0902 DD210CF0    LD    IX,POINT
001123 0906 DD3600FB    LD    (IX+00H),0FBH
001124 090A DD3601BF    LD    (IX+01H),0BFH  ;POINT = BFFBH
001125 090E AF          FI:      XOR  A
001126 090F FD2102F0    LD    IY,DATA_ADDR
001127 0913 FD4600      LD    B,(IY+00H)
001128 0916 DD7E00      LD    A,(IX+00H)
001129 0919 90          SUB  B

```

เอกสารนี้เป็นทรัพย์สินของสำนักงานเพื่อการใช้งานเพื่อวัตถุประสงค์อื่นนั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

001130	091A	3802	JR	C,FI_1	
001131	091C	1809	JR	FI_NOCA	
001132	091E	DD7E01	FI_1:	LD	A, (IX+01H)
001133	0921	FD4601		LD	B, (IY+01H)
001134	0924	98	SBC	A,B	
001135	0925	381B	JR	C,FI_CARY	
001136	0927	0604	FI_NOCA:	LD	B,04H
001137	0929	2AOCFO		LD	HL, (POINT)
001138	092C	7E	FI_GET:	LD	A, (HL)
001139	092D	23		INC	HL
001140	092E	23		INC	HL
001141	092F	23		INC	HL
001142	0930	23		INC	HL
001143	0931	77		LD	(HL),A ;POINT + 4
001144	0932	AF		KOR	A
001145	0933	7D		LD	A,L
001146	0934	D605		SUB	05H
001147	0936	6F		LD	L,A
001148	0937	7C		LD	A,H
001149	0938	DE00		SBC	A,00H
001150	093A	67		LD	H,A
001151	093B	22OCFO		LD	(POINT),HL ;POINT - 5
001152	093E	10EC		DJNZ	FI_GET
001153	0940	18CC		JR	FI
001154	0942	2AOCFO	FI_CARY:	LD	HL, (POINT) ;POINT AT OPER 3 OF LINE
001155	0945	23		INC	HL
001156	0946	3E20		LD	A,20H ;GET SPACE IN "A"

```

001157 0948 0604          LD    B,04H
001158 094A 77           FI_SP: LD    (HL),A
001159 094B 23           INC   HL
001160 094C 10FC        DJNZ  FI_SP
001161 094E AF           FI_DISPLAY: XOR  A
001162 094F DD2104F0     LD    IX,LINE
001163 0953 DD7E00      LD    A,(IX+00H)
001164 0956 D601        SUB   01H
001165 0958 27           DAA
001166 0959 DD7700      LD    (IX+00H),A
001167 095C DD7E01      LD    A,(IX+01H)
001168 095F DE00        SBC  A,00H
001169 0961 27           DAA
001170 0962 DD7701      LD    (IX+01H),A ;LINE - 1 FOR DIS_PAGE
001171 0965 AF           XOR  A
001172 0966 DD210CF0     LD    IX,POINT
001173 096A FD2102F0     LD    IY,DATA_ADDR
001174 096E DD7E00      LD    A,(IX+00H)
001175 0971 D603        SUB   03H
001176 0973 FD7700      LD    (IY+00H),A
001177 0976 DD7E01      LD    A,(IX+01H)
001178 0979 DE00        SBC  A,00H ;POINT - 3 TO DATA_ADDR
001179 097B FD7701      LD    (IY+01H),A ;FOR DIS_PAGE
001180 097E CDAD01      CALL DIS_PAGE
001181 0981 DD2A0CF0     LD    IX,(POINT)
001182 0985 DD23        INC  IX
001183 0987 DD2202F0     LD    (DATA_ADDR),IX ;DATA_ADDR TO COMMAND LIN

```

```

001184 098B 3E44          LD    A,44H
001185 098D CD4900        CALL  GOTO          ;CURSOR OT COMMAND LINE 2
001186 0990 FD2A08F0      LD    IY,(BUFF_LINE)
001187 0994 FD2204F0      LD    (LINE),IY
001188 0998 FDE1          POP   IY
001189 099A DDE1          POP   IX
001190 099C C9            RET
001191 099D
001192 099D              ;*****
001193 099D              ; FUNCTION RIGHT
001194 099D              ;*****
001195 099D DDE5          FUNC_RIGHT: PUSH  IX
001196 099F FDE5          PUSH  IY
001197 09A1 DBC4          IN    A,(PREAD)    ;GET CURSOR POSITION
001198 09A3 CBBF          RES   7,A
001199 09A5 FE4F          CP    4FH          ;IF CURSOR AT OPER3
001200 09A7 2002          JR    NZ,F_RIG1    ;NOT MOVE
001201 09A9 1825          JR    F_RIG_END
001202 09AB FE4E          F_RIG1:  CP    4EH          ;IF CURSOR AT OPER2
001203 09AD 2007          JR    NZ,F_RIG2    ;GET CURSOR --> RIGHT
001204 09AF 3E4F          LD    A,4FH
001205 09B1 CD4900        CALL  GOTO
001206 09B4 1810          JR    F_RIG4
001207 09B6 FE4D          F_RIG2:  CP    4DH          ;IF CURSOR AT OPER1
001208 09B8 2007          JR    NZ,F_RIG3    ;GET CURSOR --> RIGHT
001209 09BA 3E4E          LD    A,4EH
001210 09BC CD4900        CALL  GOTO

```

```

001211 09BF 1805                JR    F_RIG4
001212 09C1 3E4D                F_RIG3: LD    A,4DH                ;IF CURSOR AT COMMAND
001213 09C3 CD4900              CALL  GOTO
001214 09C6 DD2A02F0            F_RIG4: LD    IX,(DATA_ADDR)
001215 09CA DD23                INC   IX
001216 09CC DD2202F0            LD    (DATA_ADDR),IX
001217 09D0 FDE1                F_RIG_END: POP  IY
001218 09D2 DDE1                POP  IX
001219 09D4 C9                  RET
001220 09D5
001221 09D5                    ;*****
001222 09D5                    ; FUNCTION LEFT
001223 09D5                    ;*****
001224 09D5 DDE5                FUNC_LEFT: PUSH IX
001225 09D7 FDE5                PUSH  IY
001226 09D9 DBC4                IN    A,(PREAD)                ;GET CURSOR POSITION
001227 09DB CBBF                RES   7,A
001228 09DD FE44                CP    44H                      ;IF CURSOR AT COMMAND
001229 09DF 2002                JR    NZ,F_LEF1                ;NOT MOVE
001230 09E1 1825                JR    F_LEF_END
001231 09E3 FE4D                F_LEF1: CP    4DH                ;IF CURSOR AT OPER1
001232 09E5 2007                JR    NZ,F_LEF2                ;GET CURSOR <-- LEFT
001233 09E7 3E44                LD    A,44H
001234 09E9 CD4900              CALL  GOTO
001235 09EC 1810                JR    F_LEF4
001236 09EE FE4E                F_LEF2: CP    4EH                ;IF CURSOR AT OPER2
001237 09F0 2007                JR    NZ,F_LEF3                ;GET CURSOR <-- LEFT

```

```

001238 09F2 3E4D          LD    A,4DH
001239 09F4 CD4900        CALL  GOTO
001240 09F7 1805          JR    F_LEF4
001241 09F9 3E4E          F_LEF3: LD    A,4EH          ;IF CURSOR AT OPER3
001242 09FB CD4900        CALL  GOTO
001243 09FE DD2A02F0      F_LEF4: LD    IX,(DATA_ADDR)
001244 0A02 DD2B          DEC   IX
001245 0A04 DD2202F0      LD    (DATA_ADDR),IX
001246 0A08 FDE1          F_LEF_END: POP  IY
001247 0A0A DDE1          POP  IX
001248 0A0C C9           RET
001249 0A0D
001250 0A0D          ;*****
001251 0A0D          ;      TABLE ANYTHING
001252 0A0D          ;*****
001253 0A0D 4042514652240025 LINE1:  .DB   40H,42H,51H,46H,52H,24H,00H,25H
001254 0A15 4541502901020326 LINE2:  .DB   45H,41H,50H,29H,01H,02H,03H,26H
001255 0A1D 4344474804050627 LINE3:  .DB   43H,44H,47H,48H,04H,05H,06H,27H
001256 0A25 2021222307080928 LINE4:  .DB   20H,21H,22H,23H,07H,08H,09H,28H
001257 0A2D
001258 0A2D 414E422020202020 TAB_NO_OPERAND: .DB   "ANB   "
001259 0A35 4F52422020202020          .DB   "ORB   "
001260 0A3D 454E442020202020          .DB   "END   "
001261 0A45
001262 0A45 4C44202020202020 TAB_OPERAND:  .DB   "LD    "
001263 0A4D 414E442020202020          .DB   "AND   "
001264 0A55 4F52202020202020          .DB   "OR    "

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

001265 0A5D 54494D2020202020 .DB "TIM "
001266 0A65 434E542020202020 .DB "CNT "
001267 0A6D 4B20202023202020 .DB "K # "
001268 0A75 4F55542020202020 .DB "OUT "
001269 0A7D 5345542020202020 .DB "SET "
001270 0A85 5245534554202020 .DB "RESET "
001271 0A8D 4C442D4E4F542020 .DB "LD-NOT "
001272 0A95 414E442D4E4F5420 .DB "AND-NOT "
001273 0A9D 4F522D4E4F542020 .DB "OR-NOT "
001274 0AA5 54494D2D4E4F5420 .DB "TIM-NOT "
001275 0AAD 434E542D4E4F5420 .DB "CNT-NOT "
001276 0AB5 2020202020202020 .DB " "
001277 0ABD
001278 0ABD 2D41444452455353 TAB_ADDR: .DB "-ADDRESS NUMBER?"
001279 0ACD
001280 0ACD ;*****
001281 0ACD ;*****
001282 0ACD ;*****
001283 0ACD 216314 FUNC_GO: LD HL,TAB_RUN
001284 0AD0 CD3800 CALL WRP1 ;DISPLAY RUN
001285 0AD3 0650 LD B,50H
001286 0AD5 AF XOR A
001287 0AD6 2130F0 LD HL,AUXPOINT
001288 0AD9 77 GO1: LD (HL),A
001289 0ADA 23 INC HL
001290 0ADB 10FC DJNZ GO1 ;CLEAR AUX TIM CNT

```

```

001291 0ADD 0606          LD    B,06H
001292 0ADF 2180F0       LD    HL,IN_0
001293 0AE2 77           G02:  LD    (HL),A
001294 0AE3 23           INC   HL
001295 0AE4 10FC        DJNZ  G02           ;CLEAR IN SET OUT
001296 0AE6 2100A0      LD    HL,0A000H
001297 0AE9 2220F0      LD    (DATA_POINTER),HL
001298 0AEC 3E92        LD    A,92H           ;CONTROL PORT A,B
001299 0AEE D3E3        OUT  (PORT_CONTROL),A ;           C
001300 0AF0 3E00        LD    A,00H           ;set int vector
001301 0AF2 ED3933      OUTO (IL),A
001302 0AF5 3E30        LD    A,30H
001303 0AF7 ED47        LD    I,A
001304 0AF9 3E00        LD    A,00H           ;set time int prt
001305 0AFB ED390C      OUTO (TMDROL),A
001306 0AFE ED390E      OUTO (RLDROL),A
001307 0B01 3E78        LD    A,78H
001308 0B03 ED390D      OUTO (TMDROH),A
001309 0B06 ED390F      OUTO (RLDROH),A
001310 0B09 3E11        LD    A,11H
001311 0B0B ED3910      OUTO (TCR),A
001312 0B0E FB         EI
001313 0B0F
001314 0B0F           ;*****
001315 0B0F
001316 0B0F 2124F0      INTERPRETER: LD    HL,0F024H

```

เอกสารนี้เป็นเอกสารของบริษัทไมโครซอฟท์ จำกัด ใช้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่ควรนำออกจำหน่ายโดยไม่ได้รับอนุญาตจากไมโครซอฟท์ จำกัด การนำออกจำหน่ายโดยไม่ได้รับอนุญาตจากไมโครซอฟท์ จำกัด อาจทำให้ข้อมูลของบริษัทเสียหายได้

```

001318 0B15
001319 0B15 AF          INTERP1:  XOR  A
001320 0B16 2A20F0      LD    HL, (DATA_POINTER)
001321 0B19 7E          LD    A, (HL)          ;GET COMMAND
001322 0B1A 23          INC   HL                ;INC DATA TO POINT OPER 1
001323 0B1B FE40        CP    40H              ;LD
001324 0B1D 2006        JR    NZ, INTER1
001325 0B1F CDC90B      CALL LD_FUNC
001326 0B22 C3B50B      JP    INTER_ED
001327 0B25 FE49        INTER1: CP    49H              ;LDNOT
001328 0B27 2006        JR    NZ, INTER2
001329 0B29 CD3DOE      CALL LDNOT_FUNC
001330 0B2C C3B50B      JP    INTER_ED
001331 0B2F FE41        INTER2: CP    41H              ;AND
001332 0B31 2006        JR    NZ, INTER3
001333 0B33 CD580E      CALL AND_FUNC
001334 0B36 C3B50B      JP    INTER_ED
001335 0B39 FE4A        INTER3: CP    4AH              ;ANDNOT
001336 0B3B 2006        JR    NZ, INTER4
001337 0B3D CD690E      CALL ANDNOT_FUNC
001338 0B40 C3B50B      JP    INTER_ED
001339 0B43 FE42        INTER4: CP    42H              ;OR
001340 0B45 2006        JR    NZ, INTER5
001341 0B47 CD7A0E      CALL OR_FUNC
001342 0B4A C3B50B      JP    INTER_ED
001343 0B4D FE4B        INTER5: CP    4BH              ;ORNOT
001344 0B4F 2006        JR    NZ, INTER6

```

```

001345 0B51 CD8B0E          CALL  ORNOT_FUNC
001346 0B54 C3B50B          JP    INTER_ED
001347 0B57 FE50            INTER6: CP    50H          ;ANB
001348 0B59 2006            JR    NZ, INTER7
001349 0B5B CD9C0E          CALL  ANB
001350 0B5E C3B50B          JP    INTER_ED
001351 0B61
001352 0B61 FE51            INTER7: CP    51H          ;ORB
001353 0B63 2006            JR    NZ, INTER8
001354 0B65 CDAA0E          CALL  ORB
001355 0B68 C3B50B          JP    INTER_ED
001356 0B6B
001357 0B6B FE47            INTER8: CP    47H          ;SET
001358 0B6D 2006            JR    NZ, INTER9
001359 0B6F CD8D12          CALL  SET_FUNC
001360 0B72 C3B50B          JP    INTER_ED
001361 0B75
001362 0B75 FE48            INTER9: CP    48H          ;RES
001363 0B77 2005            JR    NZ, INTER10
001364 0B79 CD7813          CALL  RES_FUNC
001365 0B7C 1837            JR    INTER_ED
001366 0B7E
001367 0B7E FE43            INTER10: CP   43H          ;TIM
001368 0B80 200F            JR    NZ, INTER11
001369 0B82 CD290F          CALL  TIM
001370 0B85 2A20F0          LD    HL, (DATA_POINTER)
001371 0B88 23              INC   HL

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่าการตีพิมพ์สิ่งอื่น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

001372 0B89 23          INC  HL
001373 0B8A 23          INC  HL
001374 0B8B 23          INC  HL
001375 0B8C 2220F0      LD   (DATA_POINTER),HL
001376 0B8F 1824        JR   INTER_ED
001377 0B91
001378 0B91 FE44        INTER11: CP  44H          ;CNT
001379 0B93 200F        JR   NZ, INTER12
001380 0B95 CDB80E      CALL CNT
001381 0B98 2A20F0      LD   HL, (DATA_POINTER)
001382 0B9B 23          INC  HL          ;IF COMMAND 'CNT' CROSS 4
001383 0B9C 23          INC  HL          ;BECAUSE COMMAND #K USE I
001384 0B9D 23          INC  HL
001385 0B9E 23          INC  HL
001386 0B9F 2220F0      LD   (DATA_POINTER),HL
001387 0BA2 1811        JR   INTER_ED
001388 0BA4
001389 0BA4            ;**** #K NOT CARE BEFORE ADD DATA_POINTER + 4 BEHIDE (CNT,
001390 0BA4
001391 0BA4 FE46        INTER12: CP  46H          ;OUT
001392 0BA6 2005        JR   NZ, INTER_CHK
001393 0BA8 CDCA0F      CALL OUT_FUNC
001394 0BAB 1808        JR   INTER_ED
001395 0BAD
001396 0BAD FE52        INTER_CHK: CP  52H          ;END
001397 0BAF CCC20B      CALL Z, INTER_END
001398 0BB2            ; CP  20H          ;SPACE

```

```

001399 OBB2                ; CALL Z,OPER_ERR12
001400 OBB2 C30F0B         JP    INTERPRETER
001401 OBB5
001402 OBB5 2A20F0         INTER_ED: LD    HL,(DATA_POINTER)
001403 OBB8 23             INC    HL
001404 OBB9 23             INC    HL
001405 OBBA 23             INC    HL
001406 OBBB 23             INC    HL
001407 OBBC 2220F0         LD    (DATA_POINTER),HL
001408 OBBF C3150B         JP    INTERP1
001409 OBC2
001410 OBC2 2100A0         INTER_END: LD    HL,0A000H
001411 OBC5 2220F0         LD    (DATA_POINTER),HL
001412 OBC8 C9             RET
001413 OBC9
001414 OBC9                ;*****
001415 OBC9                ;***** SUB ROUTINE LD *****
001416 OBC9                ; HL (START) => POINTER IN OPER '1'
001417 OBC9                ; "C" SAVE OPER 3 ; "D" SAVE DATA FROM PORT
001418 OBC9                ;*****
001419 OBC9 7E             LD_FUNC: LD    A,(HL)        ;GET OPER 1
001420 OBCA FE00           CP    00H                ;IF INPUT FROM PORT
001421 OBCC 2006           JR    NZ,LD_CHK1
001422 OBCE CD3C0C         CALL LD_PORT
001423 OBD1 C33C0E         JP    LD_END
001424 OBD4
001425 OBD4 FE01           LD_CHK1: CP    01H                ;IF OUTPUT FROM PORT

```

เอกสารนี้เป็นเอกสารสงวนไว้สำหรับกรณีศึกษาเพื่อการศึกษาเท่านั้น ไม่นิยมนำไปเผยแพร่โดยไม่ได้รับอนุญาต

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

001426 0BD6 2006      JR    NZ,LD_CHK2
001427 0BD8 CD170D    CALL  LD_OUT
001428 0BDB C33COE    JP    LD_END
001429 0BDE
001430 0BDE FE02      LD_CHK2: CP    02H      ;IF AUX NO. 200-297
001431 0BE0 2012      JR    NZ,LD_CHK3
001432 0BE2 23        INC   HL
001433 0BE3 7E        LD    A,(HL)      ;GET OPER 2
001434 0BE4 E5        PUSH  HL          ;SAVE DATA_POINTER
001435 0BE5 2130F0    LD    HL,AUXPOINT
001436 0BE8 85        ADD   A,L         ;ADD OPER 2 FOR POINT BYT
001437 0BE9 6F        LD    L,A
001438 0BEA 2215F0    LD    (AUX_P_BUFF),HL ;SAVE POINT ADDR BYTE OF
001439 0BED E1        POP   HL
001440 0BEE CD3COD    CALL  LD_AUX
001441 0BF1 C33COE    JP    LD_END
001442 0BF4
001443 0BF4 FE03      LD_CHK3: CP    03H      ;IF AUX NO. 300-397
001444 0BF6 2014      JR    NZ,LD_CHK4
001445 0BF8 23        INC   HL
001446 0BF9 7E        LD    A,(HL)      ;GET OPER 2
001447 0BFA C60A      ADD   A,0AH       ;OPER 2 + 0AH
001448 0BFC E5        PUSH  HL          ;SAVE DATA_POINTER
001449 0BFD 2130F0    LD    HL,AUXPOINT
001450 0C00 85        ADD   A,L         ;ADD (OPER2 + 0AH),L
001451 0C01 6F        LD    L,A         ;IF OPER 1=3

```

001453	0C05	E1	POP	HL	
001454	0C06	CD3COD	CALL	LD_AUX	
001455	0C09	C33COE	JP	LD_END	
001456	0C0C				
001457	0C0C	FE04	LD_CHK4:	CP	04H ;IF AUX NO. 400-497
001458	0C0E	2014	JR	NZ,LD_CHK5	
001459	0C10	23	INC	HL	
001460	0C11	7E	LD	A,(HL)	;GET OPER 2
001461	0C12	C614	ADD	A,14H	;IF OPER 1 = 4 (400-497)
001462	0C14	E5	PUSH	HL	;^ AUX POINT BEGIN AUXPOI
001463	0C15	2130F0	LD	HL,AUXPOINT	
001464	0C18	85	ADD	A,L	
001465	0C19	6F	LD	L,A	
001466	0C1A	2215F0	LD	(AUX_P_BUFF),HL	;SAVE POINT ADDR AUX NO_?
001467	0C1D	E1	POP	HL	
001468	0C1E	CD3COD	CALL	LD_AUX	
001469	0C21	C33COE	JP	LD_END	
001470	0C24				
001471	0C24	FE05	LD_CHK5:	CP	05H ;IF TIMER
001472	0C26	2006	JR	NZ,LD_CHK6	
001473	0C28	CDDCOD	CALL	LD_TIMER	
001474	0C2B	C33COE	JP	LD_END	
001475	0C2E				
001476	0C2E	FE06	LD_CHK6:	CP	06H ;IF LD COUNTER
001477	0C30	2006	JR	NZ,LD_CHK7	
001478	0C32	CDA20D	CALL	LD_CNT	
001479	0C35	C33COE	JP	LD_END	

```

001480 0C38 CD0000      LD_CHK7:   CALL  OPER_ERR1
001481 0C3B C9           RET
001482 0C3C           ;*****
001483 0C3C 23          LD_PORT:   INC   HL
001484 0C3D 7E          LD   A,(HL)      ;GET OPER 2
001485 0C3E FE00       CP   00H
001486 0C40 201C       JR   NZ,LD_P_B   ;IF INPUT PORT A
001487 0C42 23          LD_P_A:    INC   HL
001488 0C43 7E          LD   A,(HL)      ;GET OPER 3
001489 0C44 4F          LD   C,A         ;SAVE A IN C
001490 0C45 D608       SUB  08H         ;CHECK IF OVER 7
001491 0C47 3803       JR   C,LD_P_A1
001492 0C49 CD0000     CALL  OPER_ERR2
001493 0C4C DRE0          LD_P_A1:   IN   A,(PORTA)
001494 0C4E 2F          CPL
001495 0C4F 3211F0      LD   (BUFFER1),A ;SAVE DATA PORT IN BUFFER
001496 0C52 3A80F0     LD   A,(IN_0)
001497 0C55 3212F0     LD   (BUFFER2),A
001498 0C58 3A83F0     LD   A,(IN0SET)
001499 0C5B 47          LD   B,A         ;SAVE BIT 1/0 OF I/P 0 IN
001500 0C5C 181C       JR   CHK_P_B
001501 0C5E
001502 0C5E 23          LD_P_B:    INC   HL
001503 0C5F 7E          LD   A,(HL)      ;GET OPER 3
001504 0C60 4F          LD   C,A         ;SAVE A IN C
001505 0C61 D608       SUB  08H         ;CHECK IF OVER 7
001506 0C63 3803       JR   C,LD_P_B1

```

```

001507 0C65 CD0000          CALL  OPER_ERR2
001508 0C68 DBE1           LD_P_B1:  IN    A, (PORTB)
001509 0C6A 2F             CPL
001510 0C6B 3211F0         LD    (BUFFER1),A      ;SAVE DATA PORT IN BUFFER
001511 0C6E 3A81F0         LD    A, (IN_1)
001512 0C71 3212F0         LD    (BUFFER2),A
001513 0C74 3A84F0         LD    A, (IN1SET)
001514 0C77 47             LD    B,A              ;SAVE BIT 1/0 OF I/P 1 IN
001515 0C78 1800          JR    CHK_P_B
001516 0C7A
001517 0C7A          ;*****
001518 0C7A          ; B= INOSET, IN1SET ANY BIT IS STATUS OF INPUT
001519 0C7A          ; D= DATA FROM IN PORT REALLY
001520 0C7A          ; E= IN_0, IN_1 ANY BIT IF '0' NO SET/RES
001521 0C7A          ; '1' SET/RES
001522 0C7A          ;*****
001523 0C7A 3A11F0         CHK_P_B:  LD    A, (BUFFER1)
001524 0C7D 57             LD    D,A              ;SAVE DATA PORT IN 'D'
001525 0C7E 3A12F0         LD    A, (BUFFER2)
001526 0C81 5F             LD    E,A              ;SAVE BIT SET/RES OF PORT
001527 0C82 AF             XOR   A
001528 0C83 79             LD    A,C              ;GET OPER 3 IN A
001529 0C84
001530 0C84 FE00          CHK_P_B0: CP    00H      ;CHECK PORT BIT 0
001531 0C86 200E          JR    NZ, CHK_P_B1
001532 0C88 CB43          BIT   0,E
001533 0C8A 2805          JR    Z, C_P_B0

```

เอกสารนี้เป็นทรัพย์สินส่วนไว้สำหรับการใช้งานเพื่อการศึกษาและไม่อนุญาตให้มีการนำออกโดยไม่ได้รับอนุญาต

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

001534 0C8C CB40          BIT    0,B          ;TEST BIT 0 OF (INOSSET/IN
001535 0C8E C3060D        JP     LD_P_END
001536 0C91 CB42          C_P_B0:  BIT    0,D          ;TEST BIT 0 OF DATA FROM
001537 0C93 C3060D        JP     LD_P_END
001538 0C96
001539 0C96 FE01          CHK_P_B1: CP     01H          ;CHECK PORT BIT 1
001540 0C98 200E          JR     NZ,CHK_P_B2
001541 0C9A CB4B          BIT    1,E
001542 0C9C 2805          JR     Z,C_P_B1          ;IF NO SET 'LD PORT NORMA
001543 0C9E CB48          BIT    1,B          ;TEST BIT 1 OF (INOSSET/IN
001544 0CA0 C3060D        JP     LD_P_END
001545 0CA3 CB4A          C_P_B1:  BIT    1,D          ;TEST BIT 1 OF DATA IN PO
001546 0CA5 C3060D        JP     LD_P_END
001547 0CA8
001548 0CAB FE02          CHK_P_B2: CP     02H          ;CHECK PORT BIT 2
001549 0CAA 200E          JR     NZ,CHK_P_B3
001550 0CAC CB53          BIT    2,E
001551 0CAE 2805          JR     Z,C_P_B2          ;IF NO SET 'LD PORT NORMA
001552 0CB0 CB50          BIT    2,B          ;TEST BIT 2 OF (INOSSET/IN
001553 0CB2 C3060D        JP     LD_P_END
001554 0CB5 CB52          C_P_B2:  BIT    2,D          ;TEST BIT 2 OF DATA IN PO
001555 0CB7 C3060D        JP     LD_P_END
001556 0CBA
001557 0CBA FE03          CHK_P_B3: CP     03H          ;CHECK PORT BIT 3
001558 0CBC 200E          JR     NZ,CHK_P_B4
001559 0CBE CB5B          BIT    3,E

```

001561	OCC2 CB58		BIT	3,B	
001562	OCC4 C3060D		JP	LD_P_END	
001563	OCC7 CB5A	C_P_B3:	BIT	3,D	
001564	OCC9 C3060D		JP	LD_P_END	
001565	OCCC				
001566	OCCC FE04	CHK_P_B4:	CP	04H	;CHECK PORT BIT 4
001567	OCCE 200C		JR	NZ,CHK_P_B5	
001568	OCDO CB63		BIT	4,E	
001569	OCD2 2804		JR	Z,C_P_B4	
001570	OCD4 CB60		BIT	4,B	
001571	OCD6 182E		JR	LD_P_END	
001572	OCD8 CB62	C_P_B4:	BIT	4,D	
001573	OCDA 182A		JR	LD_P_END	
001574	OCDC				
001575	OCDC FE05	CHK_P_B5:	CP	05H	;CHECK PORT BIT 5
001576	OCDE 200C		JR	NZ,CHK_P_B6	
001577	OCE0 CB6B		BIT	5,E	
001578	OCE2 2804		JR	Z,C_P_B5	
001579	OCE4 CB68		BIT	5,B	
001580	OCE6 181E		JR	LD_P_END	
001581	OCE8 CB6A	C_P_B5:	BIT	5,D	
001582	OCEA 181A		JR	LD_P_END	
001583	OCEC				
001584	OCEC FE06	CHK_P_B6:	CP	06H	;CHECK PORT BIT 6
001585	OCEE 200C		JR	NZ,CHK_P_B7	
001586	OCFO CB73		BIT	6,E	
001587	OCF2 2804		JR	Z,C_P_B6	


```

001642 0D41 D608          SUB  08H          ;CHECK IF OPER 3 OVER '7'
001643 0D43 3803          JR   C,LD_AUX1
001644 0D45 CD0000        CALL OPER_ERR2
001645 0D48
001646 0D48 2A15F0        LD_AUX1: LD   HL, (AUX_P_BUFF)
001647 0D4B 7E            LD   A, (HL)      ;GET DATA AUX 1 BYTE:1 CO
001648 0D4C 3212F0        LD   (BUFFER2),A ;SAVE DATA 8 AUX OF COLUM
001649 0D4F AF            XOR  A
001650 0D50 3A11F0        LD   A, (BUFFER1)
001651 0D53 2112F0        LD   HL, BUFFER2
001652 0D56 4E            LD   C, (HL)
001653 0D57
001654 0D57 FE00          AUX_B0: CP   00H
001655 0D59 2004          JR   NZ, AUX_B1
001656 0D5B CB41          BIT  0, C
001657 0D5D 1832          JR   AUX_END
001658 0D5F FE01          AUX_B1: CP   01H
001659 0D61 2004          JR   NZ, AUX_B2
001660 0D63 CB49          BIT  1, C
001661 0D65 182A          JR   AUX_END
001662 0D67 FE02          AUX_B2: CP   02H
001663 0D69 2004          JR   NZ, AUX_B3
001664 0D6B CB51          BIT  2, C
001665 0D6D 1822          JR   AUX_END
001666 0D6F FE03          AUX_B3: CP   03H
001667 0D71 2004          JR   NZ, AUX_B4
001668 0D73 CB59          BIT  3, C

```

```

001669 0D75 181A          JR    AUX_END
001670 0D77 FE04          AUX_B4: CP    04H
001671 0D79 2004          JR    NZ,AUX_B5
001672 0D7B CB61          BIT   4,C
001673 0D7D 1812          JR    AUX_END
001674 0D7F FE05          AUX_B5: CP    05H
001675 0D81 2004          JR    NZ,AUX_B6
001676 0D83 CB69          BIT   5,C
001677 0D85 180A          JR    AUX_END
001678 0D87 FE06          AUX_B6: CP    06H
001679 0D89 2004          JR    NZ,AUX_B7
001680 0D8B CB71          BIT   6,C
001681 0D8D 1802          JR    AUX_END
001682 0D8F CB79          AUX_B7: BIT   7,C
001683 0D91
001684 0D91 2804          AUX_END: JR    Z,LDAUX00
001685 0D93 3E01          LD    A,01H
001686 0D95 1802          JR    AUX_END1
001687 0D97 3E00          LDAUX00: LD    A,00H
001688 0D99 2A22F0        AUX_END1: LD    HL,(POINT_BUFF)
001689 0D9C 77           LD    (HL),A          ;SAVE DATA IN BUFFER FOR
001690 0D9D 23           INC   HL
001691 0D9E 2222F0        LD    (POINT_BUFF),HL
001692 0DA1 C9           RET
001693 0DA2           ;*****
001694 0DA2 23           LD_CNT:  INC   HL
001695 0DA3 7E           LD    A,(HL)          ;GET OPER 2

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้มีการเผยแพร่หรือใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

001696 ODA4 FE00	CP	00H	
001697 ODA6 2803	JR	Z,LD_CNT1	
001698 ODA8 CD0000	CALL	OPER_ERR6	
001699 ODAB AF	LD_CNT1:	XOR	A
001700 ODAC 23	INC	HL	
001701 ODAD 7E	LD	A, (HL)	;GET OPER 3
001702 ODAE 3211F0	LD	(BUFFER1),A	
001703 ODB1 D608	SUB	08H	
001704 ODB3 3803	JR	C,LD_CNT2	
001705 ODB5 CD0000	CALL	OPER_ERR2	
001706 ODB8 3A11F0	LD_CNT2:	LD	A, (BUFFER1)
001707 ODBB 87	ADD	A,A	;A * 2
001708 ODBC 2170F0	LD	HL,CNTPOINT	
001709 ODBF 85	ADD	A,L	
001710 ODC0 6F	LD	L,A	;POINT NUMBER COUNTER LSM
001711 ODC1 23	INC	HL	;POINT MSB
001712 ODC2 7E	LD	A, (HL)	;GET MSB OF COUNTER
001713 ODC3 CB6F	BIT	5,A	;CHECK BIT SET COUNTER ON
001714 ODC5 2002	JR	NZ,CHK_FINISH	
001715 ODC7 1804	JR	LDO_CNT	
001716 ODC9 CB67	CHK_FINISH:	BIT	4,A
001717 ODCB 2004	JR	NZ,LD1_CNT	
001719 ODCD 3E00	LDO_CNT:	LD	A,00H
001720 ODCF 1802	JR	LDCNT_END	
001721 ODD1 3E01	LD1_CNT:	LD	A,01H
001722 ODD3 2A22F0	LDCNT_END:	LD	HL, (POINT_BUFF)
001723 ODD6 77	LD	(HL),A	

```

001724 ODD7 23          INC  HL
001725 ODD8 2222F0     LD   (POINT_BUFF),HL
001726 ODDB C9        RET
001727 ODDC           ;*****
001728 ODDC AF        LD_TIMER: XOR  A
001729 ODDD 23        INC  HL
001730 ODDE 7E        LD   A, (HL)
001731 ODDF FE00      CP   00H
001732 ODE1 2807     JR   Z,LD_T_0_7
001733 ODE3 FE01     CP   01H
001734 ODE5 281D     JR   Z,LD_T_8_15
001735 ODE7 CD0000   CALL OPER_ERR6
001736 ODEA          LD_T_0_7: INC  HL
001737 ODEA 23        LD   A, (HL)
001738 ODEB 7E        LD   (BUFFER1),A
001739 ODEC 3211F0   SUB  08H
001741 ODF1 3803     JR   C,LD_TIM1
001742 ODF3 CD0000   CALL OPER_ERR2
001743 ODF6 3A11F0   LD_TIMER: LD   A, (BUFFER1)
001744 ODF9 87        ADD  A,A
001745 ODFA 2150F0   LD   HL,TIMPOINT
001746 ODFD 85        ADD  A,L
001747 ODFE 6F        LD   L,A
001748 ODFF 2217F0   LD   (TIM_P_BUFF),HL
001749 OE02 181C     JR   LD_T_CHK
001750 OE04

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

001751 0E04 23	LD_T_8_15:	INC	HL
001752 0E05 7E		LD	A, (HL)
001753 0E06 3211F0		LD	(BUFFER1),A
001754 0E09 D608		SUB	08H
001755 0E0B 3803		JR	C,LD_TIM2
001756 0E0D CD0000		CALL	OPER_ERR2
001757 0E10 3A11F0	LD_TIM2:	LD	A, (BUFFER1)
001758 0E13 87		ADD	A,A
001759 0E14 2150F0		LD	HL,TIMPOINT
001760 0E17 2E60		LD	L,60H
001761 0E19 85		ADD	A,L
001762 0E1A 6F		LD	L,A
001763 0E1B 2217F0		LD	(TIM_P_BUFF),HL
001764 0E1E 1800		JR	LD_T_CHK
001765 0E20			
001766 0E20 2A17F0	LD_T_CHK:	LD	HL, (TIM_P_BUFF)
001767 0E23 23		INC	HL
001768 0E24 7E		LD	A, (HL)
001769 0E25 CB6F		BIT	5,A
001770 0E27 2808		JR	Z,LDO_TIM
001771 0E29 CB67		BIT	4,A
001772 0E2B 2804		JR	Z,LDO_TIM
001773 0E2D			
001774 0E2D 3E01	LD1_TIM:	LD	A,01H
001775 0E2F 1802		JR	LD_T_END
001776 0E31 3E00	LDO_TIM:	LD	A,00H
001777 0E33 2A22F0	LD_T_END:	LD	HL, (POINT_BUFF)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

001778 0E36 77          LD   (HL),A
001779 0E37 23          INC  HL
001780 0E38 2222F0      LD   (POINT_BUFF),HL
001781 0E3B C9          RET
001782 0E3C C9          LD_END: RET
001783 0E3D
001784 0E3D          ;*****
001785 0E3D          ;***** SUB ROUTINE LDNOT *****
001786 0E3D          ;*****
001787 0E3D CDC90B      LDNOT_FUNC: CALL LD_FUNC
001788 0E40 AF          XOR  A
001789 0E41 2A22F0      LD   HL,(POINT_BUFF)
001790 0E44 2B          DEC  HL
001791 0E45 7E          LD   A,(HL) ;GET BUFF_GO
001792 0E46 FE00        CP   00H ;IF = 00H
001793 0E48 2804        JR   Z,CPL_TO_1
001794 0E4A CB87        RES  0,A ;IF 1 RES TO 0
001795 0E4C 1804        JR   LDNOT_1
001796 0E4E CBC7        CPL_TO_1: SET  0,A ;SET TO 1
001797 0E50 1800        JR   LDNOT_1
001798 0E52
001799 0E52 77          LDNOT_1: LD   (HL),A
001800 0E53 23          INC  HL
001801 0E54 2222F0      LD   (POINT_BUFF),HL
001802 0E57 C9          RET
001803 0E58

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

001804 0E58 ;*****
001805 0E58 ;***** SUB ROUTINE AND *****
001806 0E58 ; C = SAVE BUFF OLD
001807 0E58 ;*****
001808 0E58 CDC90B AND_FUNC: CALL LD_FUNC
001809 0E5B 2A22F0 LD HL,(POINT_BUFF)
001810 0E5E 2B DEC HL
001811 0E5F 7E LD A,(HL) ;GET RESAULT OF LD OPER 0
001812 0E60 2B DEC HL
001813 0E61 4E LD C,(HL) ;GET RESAULT OLD
001814 0E62 A1 AND C
001815 0E63 77 LD (HL),A ;GET RESAULT AND COMPLETE
001816 0E64 23 INC HL
001817 0E65 2222F0 LD (POINT_BUFF),HL ;POINT_BUFF_GO NEXT(SPACE
001818 0E68 C9 RET
001819 0E69
001820 0E69 ;*****
001821 0E69 ;***** SUB ROUTINE ANDNOT *****
001822 0E69 ;*****
001823 0E69 CD3D0E ANDNOT_FUNC: CALL LDNOT_FUNC
001824 0E6C 2A22F0 LD HL,(POINT_BUFF)
001825 0E6F 2B DEC HL
001826 0E70 7E LD A,(HL) ;GET RESAULT OF LD OPER 0
001827 0E71 2B DEC HL
001828 0E72 4E LD C,(HL) ;GET RESAULT OLD
001829 0E73 A1 AND C
001830 0E74 77 LD (HL),A ;GET RESAULT ANDNOT COMPL

```

```

001831 0E75 23          INC  HL
001832 0E76 2222F0     LD   (POINT_BUFF),HL ;POINT BUFF_GO NEXT(SPACE
001833 0E79 C9          RET
001834 0E7A
001835 0E7A          ;*****
001836 0E7A          ;***** SUB ROUTINE OR *****
001837 0E7A          ;*****
001838 0E7A CDC90B     OR_FUNC:  CALL LD_FUNC
001839 0E7D 2A22F0     LD   HL,(POINT_BUFF)
001840 0E80 2B          DEC  HL
001841 0E81 7E          LD   A,(HL)          ;GET RESAULT OF LD OPER 0
001842 0E82 2B          DEC  HL
001843 0E83 4E          LD   C,(HL)          ;GET RESAULT OLD
001844 0E84 B1          OR   C
001845 0E85 77          LD   (HL),A          ;GET RESAULT OR COMPLETE
001846 0E86 23          INC  HL
001847 0E87 2222F0     LD   (POINT_BUFF),HL ;POINT BUFF_GO NEXT(SPACE
001848 0E8A C9          RET
001849 0E8B
001850 0E8B          ;*****
001851 0E8B          ;***** SUB ROUTINE ORNOT *****
001852 0E8B          ;*****
001853 0E8B CD3D0E     ORNOT_FUNC: CALL LDNOT_FUNC
001854 0E8E 2A22F0     LD   HL,(POINT_BUFF)
001855 0E91 2B          DEC  HL
001856 0E92 7E          LD   A,(HL)          ;GET RESAULT OF LD OPER 0
001857 0E93 2B          DEC  HL

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

001858 0E94 4E          LD    C,(HL)          ;GET RESAULT OLD
001859 0E95 B1          OR    C
001860 0E96 77          LD    (HL),A          ;GET RESAULT ORNOT COMPLE
001861 0E97 23          INC   HL
001862 0E98 2222F0      LD    (POINT_BUFF),HL ;POINT BUFF_GO NEXT(SPACE
001863 0E9B C9          RET
001864 0E9C
001865 0E9C          ;*****
001866 0E9C          ;***** SUB ROUTINE ANB *****
001867 0E9C          ;*****
001868 0E9C 2A22F0      ANB:   LD    HL,(POINT_BUFF)
001869 0E9F 2B          DEC   HL
001870 0EA0 7E          LD    A,(HL)          ;GET RESAULT OLD
001871 0EA1 2B          DEC   HL
001872 0EA2 4E          LD    C,(HL)          ;GET RESAULT OLD OLD
001873 0EA3 A1          AND   C
001874 0EA4 77          LD    (HL),A          ;GET RESAULT AND COMPLETE
001875 0EA5 23          INC   HL
001876 0EA6 2222F0      LD    (POINT_BUFF),HL
001877 0EA9 C9          RET
001878 0EAA
001879 0EAA          ;*****
001880 0EAA          ;***** SUB ROUTINE ORB *****
001881 0EAA          ;*****
001882 0EAA 2A22F0      ORB:   LD    HL,(POINT_BUFF)
001883 0EAD 2B          DEC   HL
001884 0EAE 7E          LD    A,(HL)          ;GET RESAULT OLD

```

```

001885 0EAF 2B          DEC  HL
001886 0EB0 4E          LD   C,(HL)          ;GET RESAULT OLD OLD
001887 0EB1 B1          OR   C
001888 0EB2 77          LD   (HL),A          ;GET RESAULT OR COMPLETE
001889 0EB3 23          INC  HL
001890 0EB4 2222F0      LD   (POINT_BUFF),HL
001891 0EB7 C9          RET
001892 0EB8
001893 0EB8          ;*****
001894 0EB8          ;***** SUB ROUTINE COUNTER *****
001895 0EB8          ; HL => POINTER DATA OPER 1
001896 0EB8          ; (CNT_P_BUFF) = POINT ADDRESS BUFFER OF COUNTER(1 CNT = 2
001897 0EB8          ;*****
001898 0EB8 AF          CNT:   XOR   A
001899 0EB9 7E          LD   A,(HL)          ;GET OPER 1
001900 0EBA FE06          CP   06H
001901 0EBC 2803          JR   Z,CNT1
001902 0EBE CD0000      CALL OPER_ERR9
001903 0EC1 AF          CNT1: XOR   A
001904 0EC2 23          INC  HL
001905 0EC3 7E          LD   A,(HL)          ;GET OPER 2
001906 0EC4 FE00          CP   00H
001907 0EC6 2803          JR   Z,CNT2
001908 0EC8 CD0000      CALL OPER_ERR6
001909 0ECB AF          CNT2: XOR   A
001910 0ECC 23          INC  HL
001911 0ECD 7E          LD   A,(HL)          ;GET OPER 3

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

001912	OECE 3211F0		LD	(BUFFER1),A	
001913	OED1 D608		SUB	08H	
001914	OED3 3803		JR	C,CNT3	
001915	OED5 CD0000		CALL	OPER_ERR2	
001916	OED8 3A11F0	CNT3:	LD	A,(BUFFER1)	
001917	OEDB 87		ADD	A,A	;A * 2
001918	OEDC 2170F0		LD	HL,CNTPOINT	
001919	OEDF 85		ADD	A,L	
001920	OEE0 6F		LD	L,A	;POINT NUMBER OF COUNTER
001921	OEE1 2213F0		LD	(CNT_P_BUFF),HL	;SAVE POINT CNT LSB
001922	OEE4 2A22F0		LD	HL,(POINT_BUFF)	
001923	OEE7 2B		DEC	HL	
001924	OEE8 7E		LD	A,(HL)	
001925	OEE9 2222F0		LD	(POINT_BUFF),HL	
001926	OEEC FE00		CP	00H	
001927	OEEE 2009		JR	NZ,CNT4	
001928	OEF0 2A13F0		LD	HL,(CNT_P_BUFF)	
001929	OEF3 23		INC	HL	
001930	OEF4 7E		LD	A,(HL)	
001931	OEF5 CBAF		RES	5,A	
001932	OEF7 77		LD	(HL),A	
001933	OEF8 C9		RET		
001934	OEF9				
001935	OEF9 2A13F0	CNT4:	LD	HL,(CNT_P_BUFF)	
001936	OEFc 23		INC	HL	
001937	OEFD 7E		LD	A,(HL)	
001938	OEFE CB6F		BIT	5,A	

```

001939 0F00 2801      JR    Z,GET_K      ;IF NOT SET
001940 0F02 C9        RET
001941 0F03
001942 0F03          ;***** GET DATA #K *****
001943 0F03 AF        GET_K:   XOR    A
001944 0F04 2A20F0    LD     HL,(DATA_POINTER)
001945 0F07 23        INC    HL
001946 0F08 23        INC    HL
001947 0F09 23        INC    HL
001948 0FOA 23        INC    HL
001949 0FOB 7E        LD     A,(HL)      ;GET COMMAND #K
001950 0F0C FE45      CP     45H        ;45H KEYCODE OF #K
001951 0FOE 2800      JR     Z,CNT_K1
001952 0F10
001953 0F10 DD2A13F0  CNT_K1: LD     IX,(CNT_P_BUFF)
001954 0F14 23        INC    HL
001955 0F15 7E        LD     A,(HL)      ;GET OPER 1.OF #K
001956 0F16 F620      OR     20H        ;BITC 7 6 5 4 13
001957 0F18          ; [ OLD - SET KOB !DA
001958 0F18 DD7701    LD     (IX+1H),A   ;GET TO CNTPOINT MSB
001959 0F1B 23        INC    HL
001960 0F1C 7E        LD     A,(HL)      ;GET OPER 2
001961 0F1D 07        RLCA
001962 0F1E 07        RLCA
001963 0F1F 07        RLCA
001964 0F20 07        RLCA
001965 0F21 47        LD     B,A        ;SAVE OPER 2(SWAP) IN B

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

001966 0F22 23          INC  HL
001967 0F23 7E          LD   A,(HL)          ;GET OPER 3
001968 0F24 B0          OR   B                ;BIT [ 7 6 5 4 : 3 2 1 0
001969 0F25              ;      OPER 2      OPER 3
001970 0F25 DD7700      LD   (IX+00H),A      ;GET TO CNT POINT LSB
001971 0F28 C9          RET
001972 0F29
001973 0F29          ;*****
001974 0F29          ;***** SUB ROUTINE TIMER
001975 0F29          ;*****
001976 0F29 AF          TIM:  XOR  A
001977 0F2A 7E          LD   A,(HL)
001978 0F2B FE05          CP   05H
001979 0F2D 2803          JR   Z,TIM1
001980 0F2F CD0000          CALL OPER_ERR9
001981 0F32 AF          TIM1: XOR  A
001982 0F33 23          INC  HL
001983 0F34 7E          LD   A,(HL)
001984 0F35 FE00          CP   00H
001985 0F37 2807          JR   Z,TIM_0_7
001986 0F39 FE01          CP   01H
001987 0F3B 281D          JR   Z,TIM_8_15
001988 0F3D CD0000          CALL OPER_ERR6
001989 0F40 23          TIM_0_7: INC  HL
001990 0F41 7E          LD   A,(HL)
001991 0F42 3211F0        LD   (BUFFER1),A
001992 0F45 D608          SUB  08H

```

001993 0F47 3803		JR	C,TIM2
001994 0F49 CD0000		CALL	OPER_ERR2
001995 0F4C 3A11F0	TIM2:	LD	A,(BUFFER1)
001996 0F4F 87		ADD	A,A
001997 0F50 2150F0		LD	HL,TIMPOINT
001998 0F53 85		ADD	A,L
001999 0F54 6F		LD	L,A
002000 0F55 2217F0		LD	(TIM_P_BUFF),HL
002001 0F58 181C		JR	TIM4
002002 0F5A			
002003 0F5A 23	TIM_8_15:	INC	HL
002004 0F5B 7E		LD	A,(HL)
002005 0F5C 3211F0		LD	(BUFFER1),A
002006 0F5F D608		SUB	08H
002007 0F61 3803		JR	C,TIM3
002008 0F63 CD0000		CALL	OPER_ERR2
002009 0F66 3A11F0	TIM3:	LD	A,(BUFFER1)
002010 0F69 87		ADD	A,A
002011 0F6A 2150F0		LD	HL,TIMPOINT
002012 0F6D 2E60		LD	L,60H
002013 0F6F 85		ADD	A,L
002014 0F70 6F		LD	L,A
002015 0F71 2217F0		LD	(TIM_P_BUFF),HL
002016 0F74 1800		JR	TIM4
002017 0F76			
002018 0F76 2A22F0	TIM4:	LD	HL,(POINT_BUFF) ;CHECK DATA OF LD BEFORE
002019 0F79 2B		DEC	HL ;FOR START/STOP TIMER

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

002020 0F7A 2222F0	LD	(POINT_BUFF),HL	
002021 0F7D 7E	LD	A,(HL)	
002022 0F7E FE00	CP	00H	
002023 0F80 2009	JR	NZ,TIM5	;IF NON ZERO MADE TIMER
002024 0F82 2A17F0	LD	HL,(TIM_P_BUFF)	;IF ZERO RES TIMER
002025 0F85 23	INC	HL	
002026 0F86 7E	LD	A,(HL)	;GET MSB OF TIMER
002027 0F87 CBB7	RES	6,A	;RES BIT 'E' OF TIMER 16
002028 0F89 77	LD	(HL),A	
002029 0F8A C9	RET		
002030 0F8B			
002031 0F8B			;BIT DATA FOR TIMER 16 BIT # BIT F = BIT CHK LD #K DATA 0
002032 0F8B			; BIT E = BIT CHK LD #K DATA N
002033 0F8B			; BIT D = BIT SET RES TIMER
002034 0F8B			; BIT C = BIT TEST TIMER FINIS
002035 0F8B			; BIT B-0 = COUNTER OF TIMER
002036 0F8B			
002037 0F8B AF	TIMS:	XOR A	
002038 0F8C 2A17F0	LD	HL,(TIM_P_BUFF)	
002039 0F8F 23	INC	HL	
002040 0F90 7E	LD	A,(HL)	
002041 0F91 3211F0	LD	(BUFFER1),A	
002042 0F94 07	RLCA		;ROTATE LEFT
002043 0F95 E680	AND	80H	;FOR MOVE BIT 6 UPPER TO
002044 0F97 47	LD	B,A	;SAVE BIT F IN 'B'
002045 0F98 3A11F0	LD	A,(BUFFER1)	
002046 0F9B B0	OR	B	

```

002047 0F9C CB7F          BIT    7,A           ;TEST GET #K PASS
002048 0F9E 2801          JR     Z,GET_K_TIM  ;IF '0' GET #K
002049 0FA0 C9             RET                 ;IF '1' GET PASS
002050 0FA1
002051 0FA1                ;***** GET #K TIMER *****
002052 0FA1 AF            GET_K_TIM: XOR    A
002053 0FA2 2A20F0         LD     HL,(DATA_POINTER)
002054 0FA5 23             INC    HL
002055 0FA6 23             INC    HL
002056 0FA7 23             INC    HL
002057 0FA8 23             INC    HL
002058 0FA9 7E            LD     A,(HL)
002059 0FAA FE45          CP     45H
002060 0FAC 2803          JR     Z,TIM_K1
002061 0FAE CD0000        CALL  OPER_ERR11
002062 0FB1
002063 0FB1 DD2A17F0      TIM_K1: LD     IX,(TIM_P_BUFF)
002064 0FB5 23             INC    HL
002065 0FB6 7E            LD     A,(HL)
002066 0FB7 F640          OR     40H           ;SET BIT 'E' FOR TELL "LD
002067 0FB9 DD7701          LD     (IX+01H),A
002068 0FBC 23             INC    HL
002069 0FBD 7E            LD     A,(HL)
002070 0FBE 07             RLCA
002071 0FBF 07             RLCA
002072 0FC0 07             RLCA
002073 0FC1 07             RLCA

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

002074 OFC2 47          LD    B,A
002075 OFC3 23          INC   HL
002076 OFC4 7E          LD    A,(HL)
002077 OFC5 B0          OR    B
002078 OFC6 DD7700      LD    (IX+00H),A      ;GET DATA COUNTER OF TIMER
002079 OFC9 C9          RET
002080 OFCA
002081 OFCA          ;*****
002082 OFCA          ;***** SUB ROUTINE OUT *****
002083 OFCA          ;HL (START) => POINTER DATA OPER 1
002084 OFCA          ; [ REGISTER USE FOR OUT PORT]
002085 OFCA          ; C = OPER 3      ; D = SAVE DATA FOR OUT
002086 OFCA          ;B = BUFF FOR AND(BUFF IS 0) DATA WITH OUTPUT BEFORE "OUT"
002087 OFCA          ;          OR (BUFF IS 1)
002088 OFCA          ;*****
002089 OFCA AF          OUT_FUNC:  XOR   A
002090 OFCB E5          PUSH  HL          ;SAVE DATA POINTER
002091 OFCC          ;'HL' POINT OPER 1
002092 OFCC 7D          LD    A,L
002093 OFCD D605        SUB    05H
002094 OFCF 6F          LD    L,A
002095 OFD0 7C          LD    A,H
002096 OFD1 DE00        SBC   A,00H
002097 OFD3 67          LD    H,A          ;HL - 5 FOR CP COMMAND =
002098 OFD4 7E          LD    A,(HL)      ;GET DATA COMMAND
002099 OFD5 FE46        CP    46H          ;46H = COMMAND 'OUT'
002100 OFD7 2002        JR    NZ,OUT_F00

```

```

002101 OFD9 1804          JR   OUT_F01
002102 OFDB FE45          OUT_F00: CP   45H          ;45H = COMMAND ' #K'
002103 OFDD 2007          JR   NZ,OUT_F0
002104 OFDF 2A22F0        OUT_F01: LD   HL,(POINT_BUFF) ;IF 'OUT' TO 'OUT' USE
002105 OFE2 23            INC  HL          ;DATA FOR OUT OLD
002106 OFE3 2222F0        LD   (POINT_BUFF),HL
002107 OFE6
002108 OFE6 E1            OUT_F0: POP  HL          ;GET DATA_POINTER FROM SA
002109 OFE7 7E            LD   A,(HL)       ;GET OPER 1
002110 OFE8
002111 OFE8 FE01          OUT_F1: CP   01H          ;CHECK OUT PORT
002112 OFEA 2006          JR   NZ,OUT_F2
002113 OFEC CD5210        CALL OUT_PORT
002114 OFEF C38512        JP   OUT_END
002115 OFF2
002116 OFF2 FE02          OUT_F2: CP   02H          ;IF AUX 200-297
002117 OFF4 2012          JR   NZ,OUT_F3
002118 OFF6 23            INC  HL
002119 OFF7 7E            LD   A,(HL)       ;GET OPER 2
002120 OFF8 E5            PUSH HL
002121 OFF9 2130F0        LD   HL,AUXPOINT
002122 OFFC 85            ADD  A,L
002123 OFFD 6F            LD   L,A
002124 OFFE 2215F0        LD   (AUX_P_BUFF),HL ;SAVE POINT ADDE OF AUX 1
002125 1001 E1            POP  HL
002126 1002 CDF310        CALL OUT_AUX
002127 1005 C38512        JP   OUT_END

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

002129	1008	FE03	OUT_F3:	CP	03H	;IF AUX 300-397
002130	100A	2014		JR	NZ,OUT_F4	
002131	100C	23		INC	HL	
002132	100D	7E		LD	A,(HL)	;GET OPER 2
002133	100E	C60A		ADD	A,0AH	
002134	1010	E5		PUSH	HL	
002135	1011	2130F0		LD	HL,AUXPOINT	
002136	1014	85		ADD	A,L	
002137	1015	6F		LD	L,A	
002138	1016	2215F0		LD	(AUX_P_BUFF),HL	;SAVE POINT ADDR AUX 1 BY
002139	1019	E1		POP	HL	
002140	101A	CDF310		CALL	OUT_AUX	
002141	101D	C38512		JP	OUT_END	
002142	1020					
002143	1020	FE04	OUT_F4:	CP	04H	;IF AUX 400-497
002144	1022	2014		JR	NZ,OUT_F5	
002145	1024	23		INC	HL	
002146	1025	7E		LD	A,(HL)	;GET OPER 2
002147	1026	C614		ADD	A,14H	
002148	1028	E5		PUSH	HL	
002149	1029	2130F0		LD	HL,AUXPOINT	
002150	102C	85		ADD	A,L	
002151	102D	6F		LD	L,A	
002152	102E	2215F0		LD	(AUX_P_BUFF),HL	
002153	1031	E1		POP	HL	
002154	1032	CDF310		CALL	OUT_AUX	
002155	1035	C38512		JP	OUT_END	

```

002156 1038
002157 1038 FE05          OUT_F5:   CP    05H          ;IF OUT TIMER
002158 103A 2006          JR     NZ,OUT_F6
002159 103C CD1A12        CALL   OUT_TIM
002160 103F C38512        JP     OUT_END
002161 1042
002162 1042 FE06          OUT_F6:   CP    06H          ;IF OUT COUNTER
002163 1044 2006          JR     NZ,OUT_F7
002164 1046 CD9C11        CALL   OUT_CNT
002165 1049 C38512        JP     OUT_END
002166 104C
002167 104C CD0000        OUT_F7:   CALL   OPER_ERR1
002168 104F C38512        JP     OUT_END
002169 1052
002170 1052                ;*****
002171 1052 23          OUT_PORT: INC   HL          ;INC DATA POINTER
002172 1053 7E          LD     A,(HL)          ;GET OPER 2
002173 1054 FE00          CP     00H
002174 1056 C40000        CALL   NZ,OPER_ERR4
002175 1059 23          OUT_P_1: INC   HL
002176 105A 7E          LD     A,(HL)          ;GET OPER 3
002177 105B 4F          LD     C,A             ;SAVE OPER 3 IN 'C'
002178 105C D608          SUB    08H
002179 105E 3803          JR     C,OUT_P_2
002180 1060 CD0000        CALL   OPER_ERR2
002181 1063 2A22F0        OUT_P_2: LD     HL,(POINT_BUFF)
002182 1066 2B          DEC   HL

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่าการณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

002183	1067	7E	LD	A, (HL)	
002184	1068	57	LD	D,A	;SAVE DATA FOR OUT
002185	1069	FE01	CP	01H	
002186	106B	2843	JR	Z,OUT_1	
002187	106D				
002188	106D	79	OUT_0: LD	A,C	;GET OPER 3 IN A
002189	106E	FE00	CP	00H	
002190	1070	2004	JR	NZ,CHKO_B1	
002191	1072	06FE	LD	B,0FEH	;DATA FOR AND OUTPUT FOR
002192	1074	1834	JR	OUT_OEND	
002193	1076	FE01	CHKO_B1: CP	01H	
002194	1078	2004	JR	NZ,CHKO_B2	
002195	107A	06FD	LD	B,0FDH	;DATA FOR AND OUTPUT FOR
002196	107C	182C	JR	OUT_OEND	
002197	107E	FE02	CHKO_B2: CP	02H	
002198	1080	2004	JR	NZ,CHKO_B3	
002199	1082	06FB	LD	B,0FBH	;DATA FOR AND OUTPUT FOR
002200	1084	1824	JR	OUT_OEND	
002201	1086	FE03	CHKO_B3: CP	03H	
002202	1088	2004	JR	NZ,CHKO_B4	
002203	108A	06F7	LD	B,0F7H	;DATA FOR AND OUTPUT FOR
002204	108C	181C	JR	OUT_OEND	
002205	108E	FE04	CHKO_B4: CP	04H	
002206	1090	2004	JR	NZ,CHKO_B5	
002207	1092	06EF	LD	B,0EFH	;DATA FOR AND OUTPUT FOR
002208	1094	1814	JR	OUT_OEND	
002209	1096	FE05	CHKO_B5: CP	05H	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

002210 1098 2004      JR    NZ,CHK0_B6
002211 109A 06DF      LD    B,0DFH          ;DATA FOR AND OUTPUT FOR
002212 109C 180C      JR    OUT_OEND
002213 109E FE06      CHK0_B6: CP    06H
002214 10A0 2004      JR    NZ,CHK0_B7
002215 10A2 06BF      LD    B,0BFH          ;DATA FOR AND OUTPUT FOR
002216 10A4 1804      JR    OUT_OEND
002217 10A6 067F      CHK0_B7: LD    B,7FH          ;DATA FOR AND OUTPUT FOR
002218 10A8 1800      JR    OUT_OEND
002219 10AA
002220 10AA DBE2      OUT_OEND: IN    A,(PORTC)    ;GET DATA FROM PORT C
002221 10AC A0        AND   B                ;DATA 'B' FOR SET OFF BIT
002222 10AD D3E2      OUT   (PORTC),A
002223 10AF C9        RET
002224 10B0
002225 10B0 79        OUT_1: LD    A,C          ;GET OPEE 3
002226 10B1 FE00      CP    00H
002227 10B3 2004      JR    NZ,CHK1_B1
002228 10B5 0601      LD    B,01H          ;DATA FOR OR OUTPUT FOR B
002229 10B7 1834      JR    OUT1_END
002230 10B9 FE01      CHK1_B1: CP    01H
002231 10BB 2004      JR    NZ,CHK1_B2
002232 10BD 0602      LD    B,02H          ;DATA FOR OR OUTPUT FOR B
002233 10BF 182C      JR    OUT1_END
002234 10C1 FE02      CHK1_B2: CP    02H
002235 10C3 2004      JR    NZ,CHK1_B3
002236 10C5 0604      LD    B,04H          ;DATA FOR OR OUTPUT FOR B

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

002237 10C7 1824          JR   OUT1_END
002238 10C9 FE03          CHK1_B3: CP   03H
002239 10CB 2004          JR   NZ,CHK1_B4
002240 10CD 0608          LD   B,08H          ;DATA FOR OR OUTPUT FOR B
002241 10CF 181C          JR   OUT1_END
002242 10D1 FE04          CHK1_B4: CP   04H
002243 10D3 2004          JR   NZ,CHK1_B5
002244 10D5 0610          LD   B,10H          ;DATA FOR OR OUTPUT FOR B
002245 10D7 1814          JR   OUT1_END.END
002246 10D9 FE05          CHK1_B5: CP   05H
002247 10DB 2004          JR   NZ,CHK1_B6
002248 10DD 0620          LD   B,20H          ;DATA FOR OR OUTPUT FOR B
002249 10DF 180C          JR   OUT1_END
002250 10E1 FE06          CHK1_B6: CP   06H
002251 10E3 2004          JR   NZ,CHK1_B7
002252 10E5 0640          LD   B,40H          ;DATA FOR OR OUTPUT FOR B
002253 10E7 1804          JR   OUT1_END
002254 10E9 0680          CHK1_B7: LD   B,80H          ;DATA FOR OR OUTPUT FOR B
002255 10EB 1800          JR   OUT1_END
002256 10ED
002257 10ED DBE2          OUT1_END: IN   A,(PORTC)
002258 10EF B0           OR   B
002259 10F0 D3E2          OUT. (PORTC),A
002260 10F2 C9           RET
002261 10F3
002262 10F3          ;*****
002263 10F3

```

```

002264 10F3 23          OUT_AUX:  INC  HL
002265 10F4 7E          LD    A, (HL)          ;GET OPER 3
002266 10F5 3211F0      LD    (BUFFER1),A      ;SAVE OPER 3 IN BUFFER1
002267 10F8 D608        SUB   08H
002268 10FA 3803        JR    C,OUT_AUX1
002269 10FC CD0000      CALL OPER_ERR2
002270 10FF
002271 10FF 2A15F0      OUT_AUX1: LD    HL, (AUX_P_BUFF)
002272 1102 7E          LD    A, (HL)
002273 1103 3212F0      LD    (BUFFER2),A      ;SAVE DATA 1 BYTE OF AUX C
002274 1106 2A22F0      LD    HL, (POINT_BUFF)
002275 1109 2B          DEC   HL
002276 110A 7E          LD    A, (HL)
002277 110B FE00        CP    00H
002278 110D 2044        JR    NZ,OUT_AU_1
002279 110F
002280 110F AF          OUT_AU_0: XOR   A
002281 1110 3A11F0      LD    A, (BUFFER1)      ;get oper 3
002282 1113 2112F0      LD    HL, BUFFER2
002283 1116 4E          LD    C, (HL)
002284 1117 FE00        CP    00H
002285 1119 2004        JR    NZ,AUX_CH1
002286 111B CB81        RES   0, C
002287 111D 1878        JR    OUT_A_END
002288 111F FE01        AUX_CH1: CP    01H
002289 1121 2004        JR    NZ,AUX_CH2
002290 1123 CB89        RES   1, C

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

002291	1125	1870		JR	OUT_A_END	
002292	1127	FE02	AUX_CH2:	CP	02H	
002293	1129	2004		JR	NZ,AUX_CH3	
002294	112B	CB91		RES	2,C	
002295	112D	1868		JR	OUT_A_END	
002296	112F	FE03	AUX_CH3:	CP	03H	
002297	1131	2004		JR	NZ,AUX_CH4	
002298	1133	CB99		RES	3,C	
002299	1135	1860		JR	OUT_A_END	
002300	1137	FE04	AUX_CH4:	CP	04H	
002301	1139	2004		JR	NZ,AUX_CH5	
002302	113B	CBA1		RES	4,C	
002303	113D	1858		JR	OUT_A_END	
002304	113F	FE05	AUX_CH5:	CP	05H	
002305	1141	2004		JR	NZ,AUX_CH6	
002306	1143	CBA9		RES	5,C	
002307	1145	1850		JR	OUT_A_END	
002308	1147	FE06	AUX_CH6:	CP	06H	
002309	1149	2004		JR	NZ,AUX_CH7	
002310	114B	CBB1		RES	6,C	
002311	114D	1848		JR	OUT_A_END	
002312	114F	CBB9	AUX_CH7:	RES	7,C	
002313	1151	1844		JR	OUT_A_END	
002314	1153					
002315	1153	AF	OUT_AU_1:	XOR	A	
002316	1154	3A11F0		LD	A,(BUFFER1)	;get oper 3
002317	1157	2112F0		LD	HL,BUFFER2	

002318 115A 4E		LD	C, (HL)
002319 115B FE00		CP	00H
002320 115D 2004		JR	NZ, A1_CHK1
002321 115F CBC1		SET	0, C
002322 1161 1834		JR	OUT_A_END
002323 1163 FE01	A1_CHK1:	CP	01H
002324 1165 2004		JR	NZ, A1_CHK2
002325 1167 CBC9		SET	1, C
002326 1169 182C		JR	OUT_A_END
002327 116B FE02	A1_CHK2:	CP	02H
002328 116D 2004		JR	NZ, A1_CHK3
002329 116F CBD1		SET	2, C
002330 1171 1824		JR	OUT_A_END
002331 1173 FE03	A1_CHK3:	CP	03H
002332 1175 2004		JR	NZ, A1_CHK4
002333 1177 CBD9		SET	3, C
002334 1179 181C		JR	OUT_A_END
002335 117B FE04	A1_CHK4:	CP	04H
002336 117D 2004		JR	NZ, A1_CHK5
002337 117F CBE1		SET	4, C
002338 1181 1814		JR	OUT_A_END
002339 1183 FE05	A1_CHK5:	CP	05H
002340 1185 2004		JR	NZ, A1_CHK6
002341 1187 CBE9		SET	5, C
002342 1189 180C		JR	OUT_A_END
002343 118B FE06	A1_CHK6:	CP	06H
002344 118D 2004		JR	NZ, A1_CHK7

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

002345 118F CBF1          SET    6,C
002346 1191 1804          JR     OUT_A_END
002347 1193 CBF9          A1_CHK7: SET    7,C
002348 1195 1800          JR     OUT_A_END
002349 1197
002350 1197 2A15F0        OUT_A_END: LD     HL,(AUX_P_BUFF)
002351 119A 71            LD     (HL),C
002352 119B C9            RET
002353 119C
002354 119C                ;*****
002355 119C AF            OUT_CNT: XOR    A
002356 119D 23            INC    HL
002357 119E 7E            LD     A,(HL) ;GET OPER 2
002358 119F FE00         CP     00H
002359 11A1 2803         JR     Z,OUT_C1
002360 11A3 CD0000        CALL  OPER_ERR6
002361 11A6 23            OUT_C1: INC    HL
002362 11A7 7E            LD     A,(HL) ;GET OPER 3
002363 11A8 3211F0        LD     (BUFFER1),A
002364 11AB D608          SUB    08H
002365 11AD 3803          JR     C,OUT_C2
002366 11AF CD0000        CALL  OPER_ERR2
002367 11B2 3A11F0        OUT_C2: LD     A,(BUFFER1)
002368 11B5 87            ADD    A,A ;A * 2
002369 11B6 2170F0        LD     HL,CNTPOINT
002370 11B9 85            ADD    A,L
002371 11BA 6F            LD     L,A

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

002372 11BB 2213F0      LD      (CNT_P_BUFF),HL ;SAVE POINT ADDR BUFFER 0
002373 11BE DD2A13F0    LD      IX,(CNT_P_BUFF)
002374 11C2 DD7E01      LD      A,(IX+01H)      ;GET DATA BIT [ 7 6
002375 11C5                      ;                OLD      S
002376 11C5 CB6F        BIT     5,A            ;TEST BIT SET START COUNT
002377 11C7 2002        JR      NZ,OUT_C3
002378 11C9 1804        JR      OUT_C_END
002379 11CB
002380 11CB CB67        OUT_C3: BIT     4,A            ;TEST BIT KOB OF COUNTER
002381 11CD 2801        JR      Z,CHK_O_CNT      ;IF NOT KOB GO COUNTER
002382 11CF
002383 11CF C9          OUT_C_END: RET
002384 11D0
002385 11D0 CB7F        CHK_O_CNT: BIT    7,A            ;BIT '7' IS OLD DATA FROM
002386 11D2 2033        JR      NZ,OLD_1        ;IF OLD '0' NEW '0' NO C
002387 11D4                      ;                NEW '1' COUN
002388 11D4 2A22F0      OLD_0:  LD      HL,(POINT_BUFF)
002389 11D7 2B          DEC     HL
002390 11D8 7E          LD      A,(HL)
002391 11D9 FE00        CP      00H
002392 11DB 28F2        JR      Z,OUT_C_END
002393 11DD DD7E00      LD      A,(IX+00H)      ;IF OLD '0' NEW '1'
002394 11E0 D601        SUB     01H
002395 11E2 27          DAA
002396 11E3 DD7700      LD      (IX+00H),A
002397 11E6 DD7E01      LD      A,(IX+01H)
002398 11E9 DE00        SBC     A,00H

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

002399 11EB CBBF          SET  7,A          ;SET BIT OLD IS '1' FOR 0
002400 11ED DD7701        LD   (IX+01H),A
002401 11F0 E60F          AND  0FH
002402 11F2 FE00          CP   00H          ;CP COLUMN HUNDREAD
002403 11F4 20D9          JR   NZ,OUT_C_END
002404 11F6 DD7E00        LD   A,(IX+00H)   ;GET DATA FROM LSB OF CNT
002405 11F9 FE00          CP   00H
002406 11FB 20D2          JR   NZ,OUT_C_END
002407 11FD DD7E01        LD   A,(IX+01H)
002408 1200 CBE7          SET  4,A          ;SET BIT KOB
002409 1202 DD7701        LD   (IX+01H),A
002410 1205 18C8          JR   OUT_C_END
002411 1207
002412 1207 2A22F0        OLD_1: LD   HL,(POINT_BUFF)
002413 120A 2B            DEC  HL
002414 120B 7E            LD   A,(HL)
002415 120C FE00          CP   00H          ;IF OLD '1' NEW '0' RES 0
002416 120E
                                ;          NEW '1' NO OP
002417 120E 20BF          JR   NZ,OUT_C_END
002418 1210 DD7E01        LD   A,(IX+01H)
002419 1213 CBBF          RES  7,A          ;RES BIT OLD DATA IS '0'
002420 1215 DD7701        LD   (IX+01H),A
002421 1218 18B5          JR   OUT_C_END
002422 121A
002423 121A                ;*****
002424 121A AF            OUT_TIM: XOR  A
002425 121B 23            INC  HL            ;GET OPER 2

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

002426	121C	7E	LD	A, (HL)	
002427	121D	FE00	CP	00H	
002428	121F	2807	JR	Z, O_TIM_0_7	
002429	1221	FE01	CP	01H	
002430	1223	281E	JR	Z, O_TIM_8_15	
002431	1225	CD0000	CALL	OPER_ERR6	
002432	1228	AF	O_TIM_0_7:	XOR	A
002433	1229	23	INC	HL	;GET OPER 3
002434	122A	7E	LD	A, (HL)	
002435	122B	3211F0	LD	(BUFFER1), A	
002436	122E	D608	SUB	08H	
002437	1230	3803	JR	C, O_TIM2	
002438	1232	CD0000	CALL	OPER_ERR2	
002439	1235	3A11F0	O_TIM2:	LD	A, (BUFFER1)
002440	1238	87	ADD	A, A	
002441	1239	2150F0	LD	HL, TIMPOINT	
002442	123C	85	ADD	A, L	
002443	123D	6F	LD	L, A	
002444	123E	2217F0	LD	(TIM_P_BUFF), HL	
002445	1241	181D	JR	O_TIM4	
002446	1243				
002447	1243	AF	O_TIM_8_15:	XOR	A
002448	1244	23	INC	HL	;GET OPER 3
002449	1245	7E	LD	A, (HL)	
002450	1246	3211F0	LD	(BUFFER1), A	
002451	1249	D608	SUB	08H	
002452	124B	3803	JR	C, O_TIM3	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

002453 124D CD0000          CALL  OPER_ERR2
002454 1250 3A11F0          O_TIM3: LD   A,(BUFFER1)
002455 1253 87              ADD   A,A
002456 1254 2150F0          LD    HL,TIMPOINT
002457 1257 2E60            LD    L,60H
002458 1259 85              ADD   A,L
002459 125A 6F              LD    L,A
002460 125B 2217F0          LD    (TIM_P_BUFF),HL
002461 125E 1800            JR    O_TIM4
002462 1260
002463 1260 2A22F0          O_TIM4: LD   HL,(POINT_BUFF)
002464 1263 2B              DEC   HL
002465 1264 7E              LD   A,(HL)
002466 1265 FE00            CP   00H
002467 1267 200A            JR   NZ,O_TIM5 ;IF "1" START TIMER(SET B
002468 1269 2A17F0          LD   HL,(TIM_P_BUFF) ;IF "0" STOP TIMER (RESET
002469 126C 23              INC   HL ;HL POINT MSB OF TIMER
002470 126D 7E              LD   A,(HL)
002471 126E CBAF            RES   5,A ;STOP TIMER
002472 1270 77              LD   (HL),A
002473 1271 1811            JR   O_T_END
002474 1273
002475 1273 2A17F0          O_TIM5: LD   HL,(TIM_P_BUFF)
002476 1276 23              INC   HL
002477 1277 7E              LD   A,(HL)
002478 1278 CB77            BIT   6,A ;TEST BIT GET #K
002479 127A 2005            JR   NZ,O_TIM6 ;IF BIT 7 ='0' NOT GET

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

002480 127C CBAF          RES  5,A          ;NOT ACTIVE TIMER
002481 127E 77           LD   (HL),A
002482 127F 1803        JR   O_T_END
002483 1281 CBFF          O_TIMER: SET  5,A          ;ACTIVE TIMER
002484 1283 77           LD   (HL),A
002485 1284 C9           O_T_END: RET
002486 1285
002487 1285              ;*****
002488 1285 2A22F0        OUT_END: LD   HL,(POINT_BUFF)
002489 1288 2B           DEC  HL
002490 1289 2222F0        LD   (POINT_BUFF),HL ;BY NOT HAS COMMAND 'LD'
002491 128C C9           RET              ;USE DATA OUT FROM 'OUT_B
002492 128D
002493 128D              ;*****
002494 128D              ;***** SUBROUTINE SET *****
002495 128D              ;*****
002496 128D 7E           SET_FUNC: LD  A,(HL)      ;GET OPER 1
002497 128E FE00         CP   00H
002498 1290 2039        JR   NZ,SET_OUT
002499 1292 23           SET_IN:  INC  HL
002500 1293 7E           LD   A,(HL)          ;GET OPER 2 INPUT
002501 1294 FE00         CP   00H
002502 1296 201A        JR   NZ,SET_IN1
002503 1298 23           SET_IN0: INC  HL
002504 1299 7E           LD   A,(HL)          ;GET OPER 3 INPUT
002505 129A 3211F0       LD   (BUFFER1),A     ;SAVE OPER 3
002506 129D 2180F0       LD   HL,IN_0

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

002507 12A0 46	LD B, (HL)	;GET SET/RES BIT OF IN_0
002508 12A1 CDE512	CALL S_R_BIT	
002509 12A4 2180F0	LD HL, IN_0	
002510 12A7 70	LD (HL), B	
002511 12A8 3A83F0	LD A, (IN0SET)	
002512 12AB B1	OR C	
002513 12AC 3283F0	LD (IN0SET), A	
002514 12AF C37013	JP SET_END	
002515 12B2		
002516 12B2 23	SET_IN1: INC HL	
002517 12B3 7E	LD A, (HL)	;GET OPER 3 INPUT1
002518 12B4 3211F0	LD (BUFFER1), A	
002519 12B7 2181F0	LD HL, IN_1	
002520 12BA 46	LD B, (HL)	
002521 12BB E5	PUSH HL	
002522 12BC CDE512	CALL S_R_BIT	
002523 12BF E1	POP HL	
002524 12C0 70	LD (HL), B	
002525 12C1 3A84F0	LD A, (IN1SET)	
002526 12C4 B1	OR C	
002527 12C5 3284F0	LD (IN1SET), A	
002528 12C8 C37013	JP SET_END	
002529 12CB		
002530 12CB 23	SET_OUT: INC HL	
002531 12CC 23	INC HL	
002532 12CD 7E	LD A, (HL)	;GET OPER 3 OF OUTPUT
002533 12CE 3211F0	LD (BUFFER1), A	

002534 12D1 2182F0		LD	HL,OUTPUT	
002535 12D4 46		LD	B,(HL)	
002536 12D5 E5		PUSH	HL	
002537 12D6 CDE512		CALL	S_R_BIT	
002538 12D9 E1		POP	HL	
002539 12DA 70		LD	(HL),B	
002540 12DB 3A85F0		LD	A,(OUTSET)	
002541 12DE B1		OR	C	
002542 12DF 3285F0		LD	(OUTSET),A	
002543 12E2 C37013		JP	SET_END	
002544 12E5				
002545 12E5 2A22F0	S_R_BIT:	LD	HL,(POINT_BUFF)	
002546 12E8 2B		DEC	HL	
002547 12E9 7E		LD	A,(HL)	
002548 12EA FE00		CP	00H	
002549 12EC 2039		JR	NZ,SET_BIT	
002550 12EE 0E00		LD	C,00H	; 'C' = 00H FOR 'OR' NO CH
002551 12F0 3A11F0	R_BIT:	LD	A,(BUFFER1)	;GET OPER 3
002552 12F3 FE00		CP	00H	;IF BIT 0
002553 12F5 2003		JR	NZ,RS_B_1	
002554 12F7 CB80		RES	0,B	
002555 12F9 C9		RET		
002556 12FA FE01	RS_B_1:	CP	01H	;IF BIT 1
002557 12FC 2003		JR	NZ,RS_B_2	
002558 12FE CB88		RES	1,B	
002559 1300 C9		RET		
002560 1301 FE02	RS_B_2:	CP	02H	;IF BIT 2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

002561	1303	2003		JR	NZ,RS_B_3	
002562	1305	CB90		RES	2,B	
002563	1307	C9		RET		
002564	1308	FE03	RS_B_3:	CP	03H	;IF BIT 3
002565	130A	2003		JR	NZ,RS_B_4	
002566	130C	CB98		RES	3,B	
002567	130E	C9		RET		
002568	130F	FE04	RS_B_4:	CP	04H	;IF BIT 4
002569	1311	2003		JR	NZ,RS_B_5	
002570	1313	CBA0		RES	4,B	
002571	1315	C9		RET		
002572	1316	FE05	RS_B_5:	CP	05H	;IF BIT 5
002573	1318	2003		JR	NZ,RS_B_6	
002574	131A	CBA8		RES	5,B	
002575	131C	C9		RET		
002576	131D	FE06	RS_B_6:	CP	06H	;IF BIT 6
002577	131F	2003		JR	NZ,RS_B_7	
002578	1321	CBB0		RES	6,B	
002579	1323	C9		RET		
002580	1324	CBB8	RS_B_7:	RES	7,B	;BIT 7
002581	1326	C9		RET		
002582	1327					
002583	1327	3A11F0	SET_BIT:	LD	A, (BUFFER1)	;GET OPER 3
002584	132A	0E00		LD	C,00H	;FOR SET BIT TO 'OR'
002585	132C	FE00		CP	00H	;IF BIT 0
002586	132E	2005		JR	NZ,S_B_1	
002587	1330	CB00		SET	0,B	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

002588	1332	CBC1	SET	0,C
002589	1334	C9	RET	
002590	1335	FE01	S_B_1: CP	01H
002591	1337	2005	JR	NZ,S_B_2
002592	1339	CBC8	SET	1,B
002593	133B	CBC9	SET	1,C
002594	133D	C9	RET	
002595	133E	FE02	S_B_2: CP	02H
002596	1340	2005	JR	NZ,S_B_3
002597	1342	CBD0	SET	2,B
002598	1344	CBD1	SET	2,C
002599	1346	C9	RET	
002600	1347	FE03	S_B_3: CP	03H
002601	1349	2005	JR	NZ,S_B_4
002602	134B	CBD8	SET	3,B
002603	134D	CBD9	SET	3,C
002604	134F	C9	RET	
002605	1350	FE04	S_B_4: CP	04H
002606	1352	2005	JR	NZ,S_B_5
002607	1354	CBE0	SET	4,B
002608	1356	CBE1	SET	4,C
002609	1358	C9	RET	
002610	1359	FE05	S_B_5: CP	05H
002611	135B	2005	JR	NZ,S_B_6
002612	135D	CBE8	SET	5,B
002613	135F	CBE9	SET	5,C
002614	1361	C9	RET	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

002615 1362 FE06          S_B_6:    CP    06H
002616 1364 2005          JR    NZ,S_B_7
002617 1366 CBF0          SET   6,B
002618 1368 CBF1          SET   6,C
002619 136A C9            RET
002620 136B CBF8          S_B_7:    SET   7,B
002621 136D CBF9          SET   7,C
002622 136F C9            RET
002623 1370
002624 1370 2A22F0        SET_END:  LD    HL,(POINT_BUFF)
002625 1373 2B            DEC   HL
002626 1374 2222F0        LD    (POINT_BUFF),HL
002627 1377 C9            RET
002628 1378
002629 1378                ;*****
002630 1378                ;***** SUBROUTINE RESET *****
002631 1378                ;*****
002632 1378 7E            RES_FUNC: LD   A,(HL)          ;GET OPER 1
002633 1379 FE00          CP    00H
002634 137B 2039          JR    NZ,RES_OUT
002635 137D 23            RES_IN:  INC   HL
002636 137E 7E            LD    A,(HL)          ;GET OPER 2 INPUT
002637 137F FE00          CP    00H
002638 1381 201A          JR    NZ,RES_IN1
002639 1383 23            RES_IN0: INC   HL
002640 1384 7E            LD    A,(HL)          ;GET OPER 3 INPUT0
002641 1385 3211F0        LD    (BUFFER1),A     ;SAVE OPER 3

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

002642 1388 2180F0      LD    HL, IN_0
002643 138B 46          LD    B, (HL)          ;GET SET/RES BIT OF IN_0
002644 138C CDD013      CALL RS_R_BIT
002645 138F 2180F0      LD    HL, IN_0
002646 1392 70          LD    (HL), B
002647 1393 3A83F0      LD    A, (IN0SET)
002648 1396 A1          AND   C
002649 1397 3283F0      LD    (IN0SET), A
002650 139A C35B14      JP    RES_END
002651 139D
002652 139D 23          RES_IN1: INC   HL
002653 139E 7E          LD    A, (HL)          ;GET OPER 3 INPUT1
002654 139F 3211F0      LD    (BUFFER1), A
002655 13A2 2181F0      LD    HL, IN_1
002656 13A5 46          LD    B, (HL)
002657 13A6 E5          PUSH HL
002658 13A7 CDD013      CALL RS_R_BIT
002659 13AA E1          POP   HL
002660 13AB 70          LD    (HL), B
002661 13AC 3A84F0      LD    A, (IN1SET)
002662 13AF A1          AND   C
002663 13B0 3284F0      LD    (IN1SET), A
002664 13B3 C35B14      JP    RES_END
002665 13B6
002666 13B6 23          RES_OUT: INC   HL
002667 13B7 23          INC   HL
002668 13B8 7E          LD    A, (HL)          ;GET OPER 3 OF OUTPUT

```

002669 13B9 3211F0	LD	(BUFFER1),A	
002670 13BC 2182F0	LD	HL,OUTPUT	
002671 13BF 46	LD	B,(HL)	
002672 13C0 E5	PUSH	HL	
002673 13C1 CDD013	CALL	RS_R_BIT	
002674 13C4 E1	POP	HL	
002675 13C5 70	LD	(HL),B	
002676 13C6 3A85F0	LD	A,(OUTSET)	
002677 13C9 A1	AND	C	
002678 13CA 3285F0	LD	(OUTSET),A	
002679 13CD C35B14	JP	RES_END	
002680 13D0			
002681 13D0 2A22F0	RS_R_BIT: LD	HL,(POINT_BUFF)	
002682 13D3 2B	DEC	HL	
002683 13D4 7E	LD	A,(HL)	
002684 13D5 FE00	CP	00H	
002685 13D7 2039	JR	NZ,RES_BIT	
002686 13D9 0KFF	LD	C,OFFH	; 'C' = FFH FOR 'AND' NO C
002687 13DB 3A11F0	R_BIT: LD	A,(BUFFER1)	;GET OPER 3
002688 13DE FE00	CP	00H	;IF BIT 0
002689 13E0 2003	JR	NZ,R_B_1	
002690 13E2 CB80	RES	0,B	
002691 13E4 C9	RET		
002692 13E5 FE01	R_B_1: CP	01H	;IF BIT 1
002693 13E7 2003	JR	NZ,R_B_2	
002694 13E9 CB88	RES	1,B	
002695 13EB C9	RET		

002696 13EC FE02	R_B_2:	CP	02H	;IF BIT 2
002697 13EE 2003		JR	NZ,R_B_3	
002698 13F0 CB90		RES	2,B	
002699 13F2 C9		RET		
002700 13F3 FE03	R_B_3:	CP	03H	;IF BIT 3
002701 13F5 2003		JR	NZ,R_B_4	
002702 13F7 CB98		RES	3,B	
002703 13F9 C9		RET		
002704 13FA FE04	R_B_4:	CP	04H	;IF BIT 4
002705 13FC 2003		JR	NZ,R_B_5	
002706 13FE CBA0		RES	4,B	
002707 1400 C9		RET		
002708 1401 FE05	R_B_5:	CP	05H	;IF BIT 5
002709 1403 2003		JR	NZ,R_B_6	
002710 1405 CBA8		RES	5,B	
002711 1407 C9		RET		
002712 1408 FE06	R_B_6:	CP	06H	;IF BIT 6
002713 140A 2003		JR	NZ,R_B_7	
002714 140C CB80		RES	6,B	
002715 140E C9		RET		
002716 140F CBB8	R_B_7:	RES	7,B	;BIT 7
002717 1411 C9		RET		
002718 1412 3A11F0	RES_BIT:	LD	A,(BUFFER1)	;GET OPER 3
002719 1415 0KFF		LD	C,OFFH	;FOR SET BIT TO 'AND'
002720 1417 FE00		CP	00H	;IF BIT 0
002721 1419 2005		JR	NZ,RES_B_1	
002722 141B CBC0		SET	0,B	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

002723 141D CB81	RES	0,C
002724 141F C9	RET	
002725 1420 FE01	RES_B_1: CP	01H
002726 1422 2005	JR	NZ,RES_B_2
002727 1424 CBC8	SET	1,B
002728 1426 CB89	RES	1,C
002729 1428 C9	RET	
002730 1429 FE02	RES_B_2: CP	02H
002731 142B 2005	JR	NZ,RES_B_3
002732 142D CBD0	SET	2,B
002733 142F CB91	RES	2,C
002734 1431 C9	RET	
002735 1432 FE03	RES_B_3: CP	03H
002736 1434 2005	JR	NZ,RES_B_4
002737 1436 CBD8	SET	3,B
002738 1438 CB99	RES	3,C
002739 143A C9	RET	
002740 143B FE04	RES_B_4: CP	04H
002741 143D 2005	JR	NZ,RES_B_5
002742 143F CBE0	SET	4,B
002743 1441 CBA1	RES	4,C
002744 1443 C9	RET	
002745 1444 FE05	RES_B_5: CP	05H
002746 1446 2005	JR	NZ,RES_B_6
002747 1448 CBE8	SET	5,B
002748 144A CBA9	RES	5,C
002749 144C C9	RET	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรรมใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

002750 144D FE06          RES_B_6:  CP    06H
002751 144F 2005          JR      NZ,RES_B_7
002752 1451 CBFO          SET     6,B
002753 1453 CBB1          RES     6,C
002754 1455 C9            RET
002755 1456 CBF8          RES_B_7:  SET     7,B
002756 1458 CBB9          RES     7,C
002757 145A C9            RET
002758 145B
002759 145B 2A22FO          RES_END:  LD     HL,(POINT_BUFF)
002760 145E 2B              DEC     HL
002761 145F 2222FO          LD     (POINT_BUFF),HL
002762 1462 C9            RET
002763 1463
002764 1463                ;*****
002765 1463 2A2A2A2A2052554E TAB_RUN: .DB    "**** RUNING ****"
146B 494E47202A2A2A2A
002766 1473 2A2A2A2A2A2A2A .DB    "*****"
147B 2A2A2A2A2A2A2A2A
002767 1483
002768 1483                ;***** INTERRUPT VECTOR *****
002769 3000                .ORG   3000H
002770 3000 1230          INT1:   .DRW  EMPTY
002771 3002 1230          INT2:   .DRW  EMPTY
002772 3004 0020          PRT0:   .DRW  2000H
002773 3006 1230          PRT1:   .DRW  EMPTY
002774 3008 1230          DMA0:   .DRW  EMPTY

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

002775 300A 1230      DMA1:      .DRW  EMPTY
002776 300C 1230      CSIO:      .DRW  EMPTY
002777 300E 1230      ASCIO:     .DRW  EMPTY
002778 3010 1230      ASCI1:     .DRW  EMPTY
002779 3012 ED76      EMPTY:    SLP
002780 3014
002781 3014           ;***** INTERRUPT SERVICE ROUTINE *****
002782 2000           .ORG 2000H
002783 2000 F3      INT_TIMER: DI
002784 2001 08      EX      AF,AF'
002785 2002 D9      EXX
002786 2003 ED3810   INT_0:    INO  A,(TCR)
002787 2006 CB77      BIT  6,A
002788 2008 28F9      JR   Z,INT_0
002789 200A ED380C   INT_0:    INO  A,(TMDROL)
002790 200D AF      XOR  A
002791 200E 0610      LD   B,10H      ;COUNTER FOR TIMER 16 CHA
002792 2010 2150F0   LD   HL,TIMPOINT
002793 2013 23      INT_1:    INC  HL
002794 2014 7E      LD   A,(HL)
002795 2015 CB6F      BIT  5,A      ;TEST BIT SET
002796 2017 2820      JR   Z,INT_2
002797 2019 CB67      BIT  4,A      ;TEST BIT KOB
002798 201B 201C      JR   NZ,INT_2
002799 201D AF      COUNT_TIN: KOR  A
002800 201E 2B      DEC  HL
002801 201F 7E      LD   A,(HL)

```

002802 2020 D601	SUB 01H
002803 2022 27	DAA
002804 2023 77	LD (HL),A
002805 2024 4F	LD C,A
002806 2025 23	INC HL
002807 2026 7E	LD A,(HL)
002808 2027 DE00	SBC A,00H
002809 2029 77	LD (HL),A
002810 202A E60F	AND OFH
002811 202C FE00	CP 00H
002812 202E 2009	JR NZ,INT_2
002813 2030 79	LD A,C
002814 2031 FE00	CP 00H
002815 2033 2004	JR NZ,INT_2
002816 2035 7E	LD A,(HL)
002817 2036 CBE7	SET 4,A
002818 2038 77	LD (HL),A
002819 2039 23	INT_2: INC HL
002820 203A 10D7	DJNZ INT_1
002821 203C D9	EKK
002822 203D 08	EX AF,AF'
002823 203E FB	EI
002824 203F ED4D	RETI
002825 2041	
002826 2041	.END

A1_CHK1 =1163 CHK_P_B =0C7A FA_G00 =04D7 GO2 =0AE2 LD_CNT2 =0DB8 OUT_F01 =0FDF
 A1_CHK2 =116B CHK_P_B0=0C84 FA_G01 =04DD GOTO =0049 LD_END =0E3C OUT_F1 =0FE8
 A1_CHK3 =1173 CHK_P_B1=0C96 FA_G01.1=04E7 IL =0033 LD_FUNC =0BC9 OUT_F2 =0FF2
 A1_CHK4 =117B CHK_P_B2=0CA8 FA_G010 =05DD INOSET =F083 LD_OUT =0D17 OUT_F3 =1008
 A1_CHK5 =1183 CHK_P_B3=0CBA FA_G02 =04F3 IN1SET =F084 LD_OUT1 =0D20 OUT_F4 =1020
 A1_CHK6 =118B CHK_P_B4=0CCC FA_G03 =0507 INITLCD =006E LD_OUT2 =0D2A OUT_F5 =1038
 A1_CHK7 =1193 CHK_P_B5=0CDC FA_G04 =0522 INT1 =3000 LD_PORT =0C3C OUT_F6 =1042
 ADDR_NUM=F00A CHK_P_B6=0C8C FA_G05 =053E INT2 =3002 LD_P_A =0C42 OUT_F7 =104C
 ANB =0E9C CHK_P_B7=0CFC FA_G06 =056A INTER1 =0B25 LD_P_A1 =0C4C OUT_FUNC=0FCA
 ANDNOT_F=0E69 CLSLCD =007F FA_G07 =057F INTER10 =0B7E LD_P_B =0C5E OUT_PORT=1052
 AND_FUNC=0E58 CNT =0EB8 FA_G08 =05D3 INTER11 =0B91 LD_P_B1 =0C68 OUT_P_1 =1059
 ASCIO =300E CNT1 =0EC1 FA_G09 =05DA INTER12 =0BA4 LD_P_END=0D06 OUT_P_2 =1063
 ASCI1 =3010 CNT2 =0ECB FA_L2 =047F INTER2 =0B2F LD_TIM1 =0DF6 OUT_TIM =121A
 AUXPOINT=F030 CNT3 =0ED8 FA_NEW =045F INTER3 =0B39 LD_TIM2 =0E10 PAGE_2LI=01B8
 AUX_B0 =0D57 CNT4 =0EF9 FA_WRITE=046B INTER4 =0B43 LD_TIMER=0DDC PDATA =00C0
 AUX_B1 =0D5F CNTPOINT=F070 FDWCON1 =087B INTER5 =0B4D LD_T_0_7=0DEA POINT =F00C
 AUX_B2 =0D67 CNT_K1 =0F10 FDWCON2 =0890 INTER6 =0B57 LD_T_8_1=0E04 POINT_BU=F022
 AUX_B3 =0D6F CNT_P_BU=F013 FDWCON3 =08A2 INTER7 =0B61 LD_T_CHK=0E20 PORTA =00E0
 AUX_B4 =0D77 COM_NO_0=022D FD_CHK =071E INTER8 =0B6B LD_T_END=0E33 PORTB =00E1
 AUX_B5 =0D7F COUNT_TI=201D FD_CHK1 =073C INTER9 =0B75 LINE =F004 PORTC =00E2
 AUX_B6 =0D87 CPL_TO_1=0E4E FD_COLO =06DB INTERP1 =0B15 LINE1 =0A0D PORTCT1 =0083
 AUX_B7 =0D8F CP_AND =03C7 FD_COL1 =06F2 INTERPRE=0B0F LINE2 =0A15 PORTC_KE=0082
 AUX_CH1 =111F CP_AND_N=03D2 FD_COL2 =0709 INTER_CH=0BAD LINE3 =0A1D PORT_CON=00E3
 AUX_CH2 =1127 CP_CNT =0409 FD_CON1 =063A INTER_ED=0BB5 LINE4 =0A25 POWER =0000
 AUX_CH3 =112F CP_CNT_N=0414 FD_CON2 =0652 INTER_EN=0BC2 NOFIRST =0296 POWER1 =0003
 AUX_CH4 =1137 CP_LD =03B1 FD_CON3 =066A INT_0 =2003 NO_COM =022A PREAD =00C4
 AUX_CH5 =113F CP_LD_NO=03BC FD_CON4 =06AB INT_1 =2013 NO_NUM =0268 PREAD_DA=00C6

AUX_CH6 =1147 CP_OR =03DD FD_CUR =0674 INT_2 =2039 NO_OPERA=02FD PRTO =3004
 AUX_CH7 =114F CP_OR_NO=03E8 FD_DISP =08B4 INT_TIME=2000 O_TIM2 =1235 PRT1 =3006
 AUX_END =0D91 CP_TIM =03F3 FD_END =0743 IN_0 =F080 O_TIM3 =1250 PSING =00C2
 AUX_END1=0D99 CP_TIM_N=03FE FD_END1 =0746 IN_1 =F081 O_TIM4 =1260 R_BIT =12F0
 AUX_P_BU=F015 CSIO =300C FD_GET =067A KEY =0339 O_TIM5 =1273 R_B_1 =13E5
 BUFFER1 =F011 DATA_ADD=F002 FD_LOOP =0678 KEYCODE =035A O_TIM6 =1281 R_B_2 =13EC
 BUFFER2 =F012 DATA_POI=F020 FI =090E KEYIN =F000 O_TIM_0_=1228 R_B_3 =13F3
 BUFF_CUR=F00F DELAY =026B FIND_KEY=0399 KEY_DATA=F001 O_TIM_8_=1243 R_B_4 =13FA
 BUFF_DAT=F006 DELAY1 =008E FIRST_ST=F010 KLIN1 =037D O_T_END =1284 R_B_5 =1401
 BUFF_GO =F024 DELAY12 =0092 FI_1 =091E KLIN2 =0384 OLD_0 =11D4 R_B_6 =1408
 BUFF_LIN=F008 DELAY2 =026E FI_CARY =0942 KLIN3 =038B OLD_1 =1207 R_B_7 =140F
 C_CONO =0214 DELAY3 =0270 FI_DISPL=094E KLIN4 =0392 OPER3_DI=0106 READ =0051
 C_CON1 =021D DISCOM =02DE FI_GET =092C L_CON =013E OPERAND =0318 RRAD1 =0052
 C_CON2 =0224 DISCOM1 =02F5 FI_NOCA =0927 L_CONO =0148 OPERCON =02D8 RESTART =0097
 C_P_B0 =0C91 DISLINE =02A9 FI_SP =094A LED =0152 OPERSP =02DA RES_BIT =1412
 C_P_B1 =0CA3 DISOPER =02D1 FUNC_ADD=045E L_END =014D OPER_ERR=0000 RES_B_1 =1420
 C_P_B2 =0CB5 DIS_ENDP=018A FUNC_DEL=05F7 L_RET =016B ORB =0EAA RES_B_2 =1429
 C_P_B3 =0CC7 DIS_PAGE=01AD FUNC_DOW=085D LDO_CNT =0DCD ORNOT_FU=0E8B RES_B_3 =1432
 C_P_B4 =0CD8 DMA0 =3008 FUNC_GO =0ACD LDO_TIM =0E31 OR_FUNC =0E7A RES_B_4 =143B
 C_P_B5 =0CE8 DMA1 =300A FUNC_INS=08F6 LD1_CNT =0DD1 OUT1_END=10ED RES_B_5 =1444
 C_P_B6 =0CF8 EMPTY =3012 FUNC_LEF=09D5 LD1_TIM =0E2D OUTPUT =F082 RES_B_6 =144D
 C_P_B7 =0D04 END_PAGE=016C FUNC_NOT=03A5 LDAUX00 =0D97 OUTSET =F085 RES_B_7 =1456
 CHKO_B1 =1076 F_LEF1 =09E3 FUNC_RIG=099D LDCNT_EN=0DD3 OUT_0 =106D RES_END =145B
 CHKO_B2 =107E F_LEF2 =09EE FUNC_UP =0761 LDNOT_1 =0E52 OUT_0END=10AA RES_FUNC=1378
 CHKO_B3 =1086 F_LEF3 =09F9 FUN_NOT_=041F LDNOT_FU=0E3D OUT_1 =10B0 RES_IN =137D
 CHKO_B4 =108E F_LEF4 =09FE FUN_NO_N=0427 LD_0 =0DOC OUT_AUX =10F3 RES_INO =1383
 CHKO_B5 =1096 F_LEF_EN=0A08 FU_CON1 =07BD LD_1 =0DOE OUT_AUX1=10FF RES_IN1 =139D

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

CHK0_B6 =109E F_RIG1 =09AB FU_CON2 =07D2 LD_AUX =0D3C OUT_AU_0=110F RES_OUT =13B6
 CHK0_B7 =10A6 F_RIG2 =09B6 FU_CON3 =07E4 LD_AUX1 =0D48 OUT_AU_1=1153 RLDR0H =000F
 CHK1_B1 =10B9 F_RIG3 =09C1 FU_DISP =07F6 LD_CHK1 =0BD4 OUT_A_KN=1197 RLDR0L =000E
 CHK1_B2 =10C1 F_RIG4 =09C6 FU_DISP2=0814 LD_CHK2 =0BDE OUT_C1 =11A6 RS_B_1 =12FA
 CHK1_B3 =10C9 F_RIG_KN=09D0 FU_DISP3=081B LD_CHK3 =0BF4 OUT_C2 =11B2 RS_B_2 =1301
 CHK1_B4 =10D1 FA_CLS =04A7 FU_DISP4=081E LD_CHK4 =0C0C OUT_C3 =11CB RS_B_3 =1308
 CHK1_B5 =10D9 FA_CONO =0489 FU_END =0858 LD_CHK5 =0C24 OUT_CNT =119C RS_B_4 =130F
 CHK1_B6 =10E1 FA_CON1 =048D FU_START=078D LD_CHK6 =0C2E OUT_C_EN=11CF RS_B_5 =1316
 CHK1_B7 =10E9 FA_CON2 =04BA GET_K =0F03 LD_CHK7 =0C38 OUT_END =1285 RS_B_6 =131D
 CHK_FINI=0DC9 FA_END =05F4 GET_K_TI=0FA1 LD_CNT =0DA2 OUT_F0 =0FE6 RS_B_7 =1324
 CHK_O_CN=11D0 FA_GO =04C7 GO1 =0AD9 LD_CNT1 =0DAB OUT_F0O =0FDB RS_R_BIT=13D0





เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทดลองที่ 1 การควบคุมการปิด-เปิดมอเตอร์

1. ลักษณะงาน

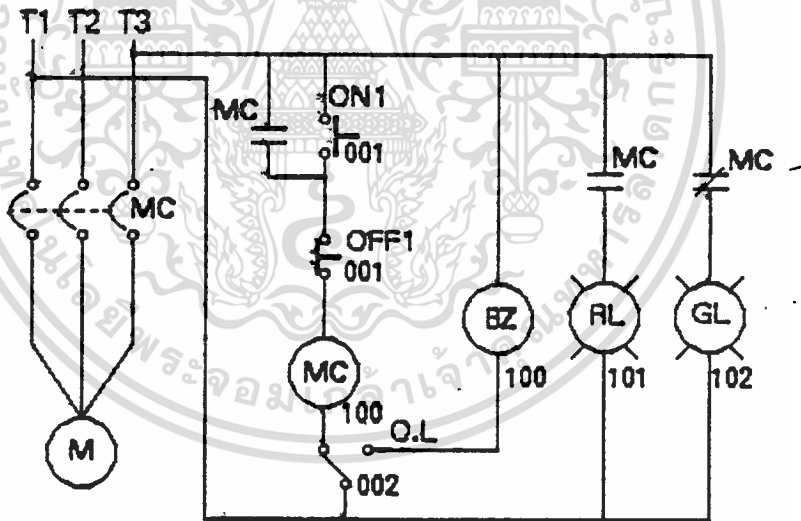
ทำการควบคุม มอเตอร์ โดยมีปุ่ม เปิด-ปิด (ON/OFF) มอเตอร์ (MC) แยกจากกัน และมี การแสดงสภาวะการทำงานต่าง ๆ ของมอเตอร์ ดังนี้

(1) เมื่อกดปุ่ม ON มอเตอร์ (MC) จะทำงาน หลอดไฟสีแดง (RL) ติด

(2) เมื่อกดปุ่ม OFF มอเตอร์ (MC) จะหยุดทำงานหลอดไฟสีแดง RL จะดับ และ หลอดไฟ สีเขียว (GL) จะสว่างแทน

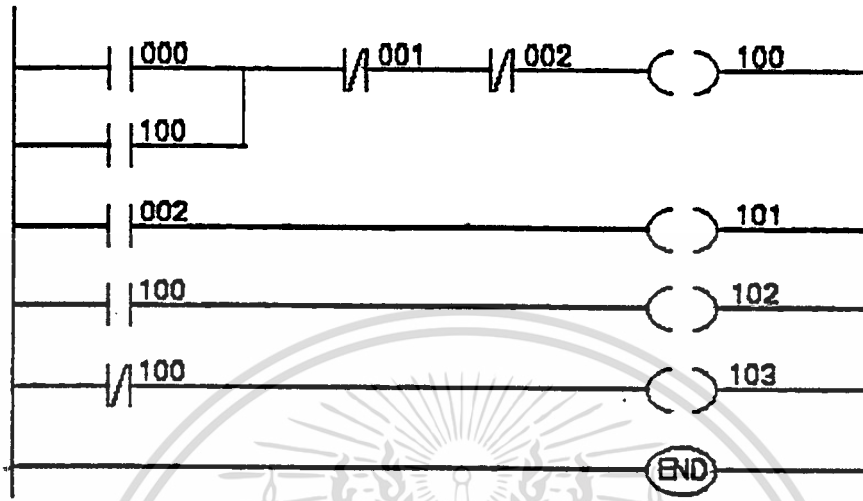
(3) ในกรณีที่มอเตอร์ (MC) ทำงาน เกินขีดจำกัด (OVERLOAD) สวิตช์ จะเปลี่ยนตำแหน่ง มาที่ O.L. ทำให้มอเตอร์ (MC) หยุดทำงานทันที และ ลำโพง (BZ) จะส่งเสียงเตือน

2. วงจรรีเลย์



วงจรรีเลย์การควบคุมการปิดเปิดมอเตอร์

3. วงจรแลตเคอร์



วงจรแลตเคอร์ควบคุมการเปิด-ปิดมอเตอร์

4. โปรแกรมบูตลีน

ลำดับที่	คำสั่ง	อุปกรณ์
000	LD	800
001	OR	100
002	AND-NOT	001
003	AND-NOT	002
004	OUT	100
005	LD	002
006	OUT	101
007	LD	020
008	OUT	102
009	LD-OUT	100
010	OUT	103
011	END	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5. คำอธิบาย

- (1). คำสั่ง ลำดับที่ 0-5 ใช้ในการควบคุม การทำงานของมอเตอร์ และ ป้องกัน การจ่าย กระแส มากเกินไปของมอเตอร์ (OVERLOAD)
- (2). คำสั่งลำดับที่ 6 ใช้ส่งสัญญาณเตือน เมื่อเกิดการ (OVERLOAD)
- (3). คำสั่งลำดับที่ 8-10 ใช้บอกสถานะการทำงานของมอเตอร์

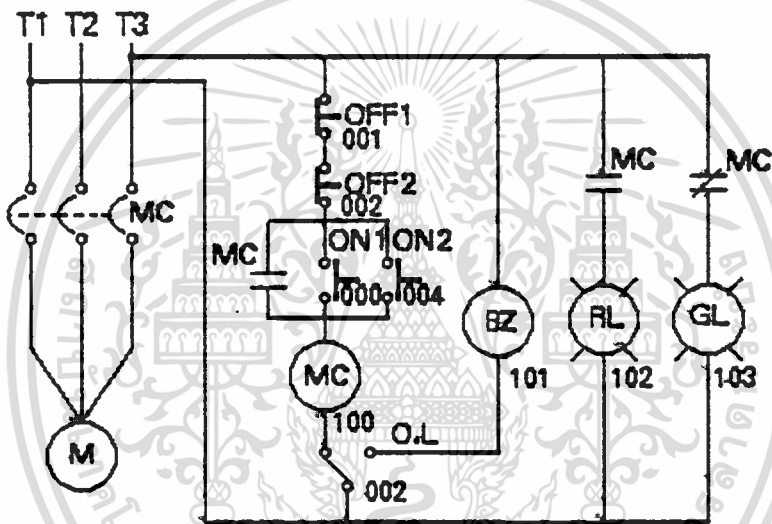


การทดลองที่ 2 การควบคุมมอเตอร์โดยใช้สวิตช์ 2 ชุด

1. ลักษณะงาน

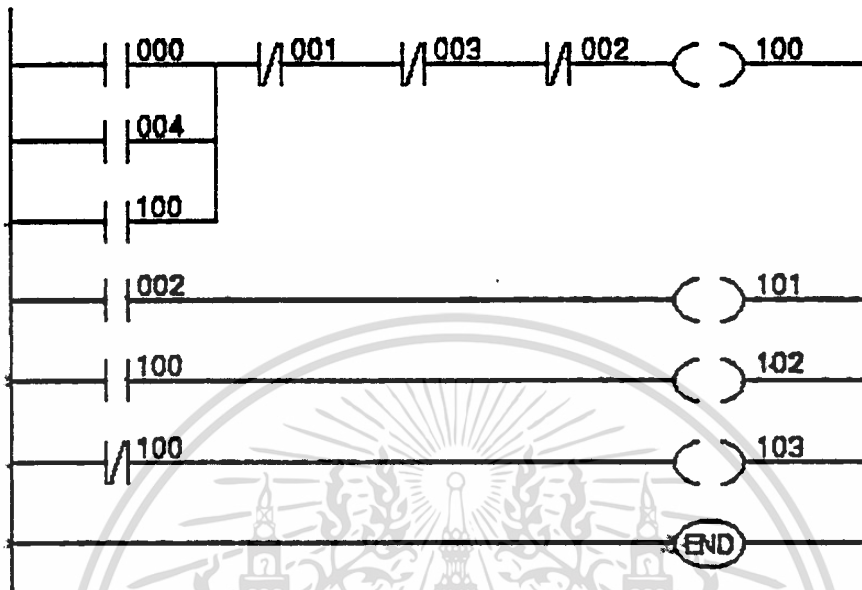
- (1). มีปุ่มควบคุมการเปิด-ปิดมอเตอร์ 2 ชุด ที่อิสระจากกัน
- (2). เหมือนตัวอย่างในการทดลองที่ 1 แต่เพิ่มปุ่มควบคุมเป็น 2 ชุด

2. วงจรรีเลย์



วงจรรีเลย์ควบคุมมอเตอร์โดยใช้สวิตช์ 2 ชุด

3. วงจรแลตเตอร์



วงจรรแลตเตอร์ควบคุมมอเตอร์โดยใช้สวิตช์ 2 ชุด

4. โปรแกรมบดลิน

ลำดับที่	คำสั่ง	อุปกรณ์
000	LD	000
001	OR	004
002	OR	100
003	AND-NOT	001
004	AND-NOT	003
005	AND-NOT	002
006	OUT	100
007	LD	002
008	OUT	101
009	LD	100
010	OUT	102
011	LD-NOT	100

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

012

OUT

103

013

END

5. คำอธิบาย

(1). คำสั่งที่ 0-7 ใช้ควบคุมการ ปิด-เปิด มอเตอร์ ผ่านปุ่มเปิด 000 กับ 004 และ หยุด มอเตอร์ ด้วยปุ่ม 001 กับ 003

(2). คำสั่งที่ 8 ขึ้นไปใช้บอก สภาวะการทำงานของมอเตอร์



การทดลองที่ 3 การควบคุมมอเตอร์ 3 เฟส 2 ทิศทาง (1)

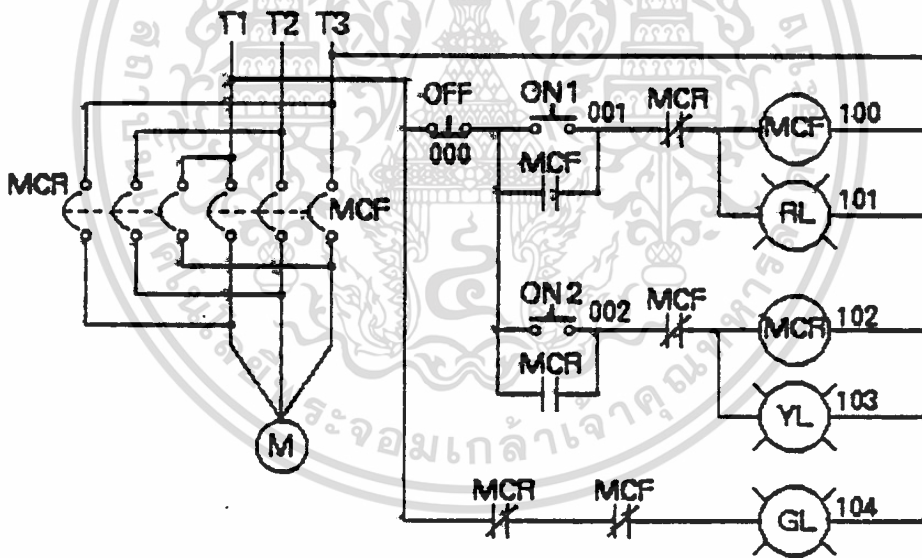
1. ลักษณะงาน

(1). ใช้รีเลย์ 2 ตัว เพื่อควบคุมทิศทางการหมุนของมอเตอร์ โดย (MCF) หมุนในทิศทางตามเข็มนาฬิกา และ (MCR) หมุนในทิศทางทวนเข็มนาฬิกา

(2). มีไฟบอกสถานะการทำงานของมอเตอร์ โดย

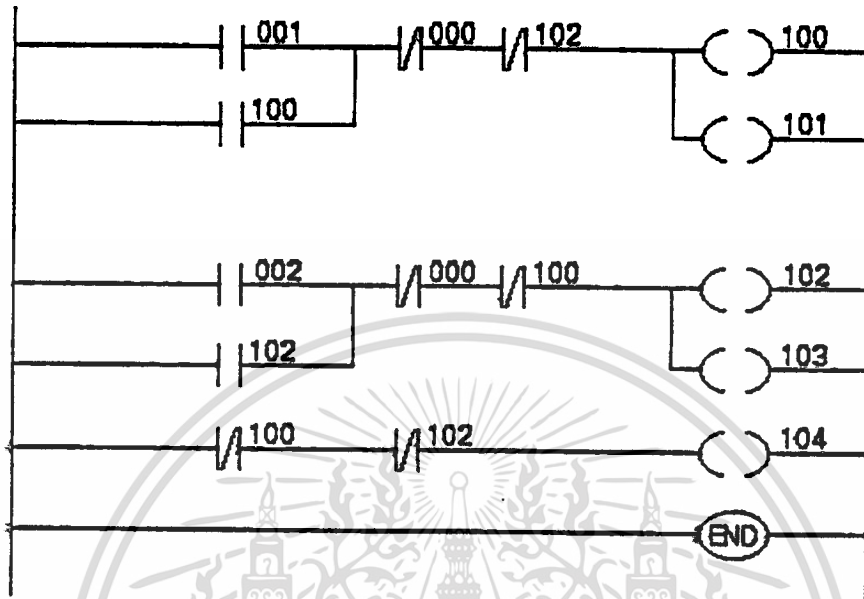
- เมื่อหมุนตามเข็มนาฬิกา ให้หลอดไฟสีเขียว (YL) สว่าง
- เมื่อหมุนทวนเข็มนาฬิกา ให้หลอดไฟสีแดง (RL) สว่าง
- ขณะที่มอเตอร์หยุดหมุน ให้หลอดไฟสีเขียว (GL) สว่าง

2. วงจรรีเลย์



วงจรรีเลย์ควบคุมมอเตอร์ 3 เฟส 2 ทิศทาง (1)

3. วงจรแลตเตอร์



วงจรแลตเตอร์ควบคุมมอเตอร์ 3 เฟส 2 ทิศทาง (1)

4. โปรแกรมมอดิน

ลำดับที่	คำสั่ง	อุปกรณ์
000	LD	001
001	OR	100
002	AND-NOT	000
003	AND-NOT	102
004	OUT	100
005	OUT	101
006	LD	002
007	OR	102
008	AND-NOT	000
009	AND-OUT	100
010	OUT	102
011	OUT	103

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

012	LD-NOT	100
013	AND-NOT	102
014	OUT	104
015	END	

5. คำอธิบาย

(1). ในการเปลี่ยนทิศทาง การหมุนของมอเตอร์ 3 เฟส ทำได้โดยการสลับสายไฟที่ต่อเข้าตัวมอเตอร์ ซึ่งในการออกแบบเราใช้รีเลย์ MCF และ MCR เป็นตัวกำหนดทิศทางการหมุนของมอเตอร์

(2). เราใช้คอนแทคของ MCF และ MCR มาควบคุมซึ่งกัน และกันเพื่อป้องกันการลัดวงจร (SHOT CIRCUIT) เมื่อรีเลย์ MCF และ MCR ทำงานพร้อมกัน

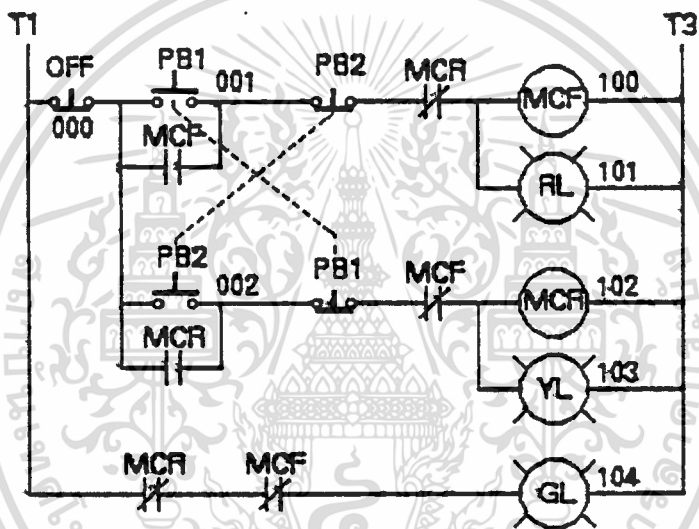
(3). ในการทดลองนี้ เราต้องกดปุ่ม OFF เพื่อหยุดการทำงานของมอเตอร์ ก่อนที่จะเปลี่ยนทิศทางการหมุนของมอเตอร์

การทดลองที่ 4 การควบคุมมอเตอร์ 3 เฟส 2 ทิศทาง (2)

1. ลักษณะงาน

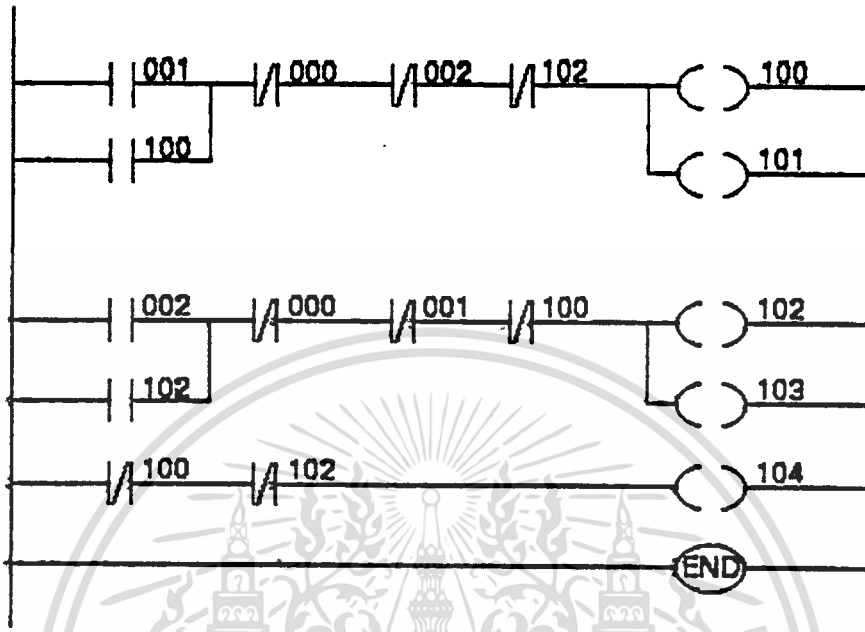
(1). เหมือนการทดลองที่ 3 แต่ต้องการให้สามารถ จะกดปุ่มเปลี่ยนทิศทางการหมุน ของมอเตอร์ได้ทันที โดยไม่ต้องกด OFF เพื่อหยุดมอเตอร์ก่อน

2. วงจรรีเลย์



วงจรรีเลย์ควบคุมมอเตอร์ 3 เฟส 2 ทิศทาง (2)

3. วงจรแลตเตอร์



วงจรถัดเตอร์ควมมมอเตอร์ 3 เฟส 2 ทิศทาง (2)

4. คำสั่งขุดลีน

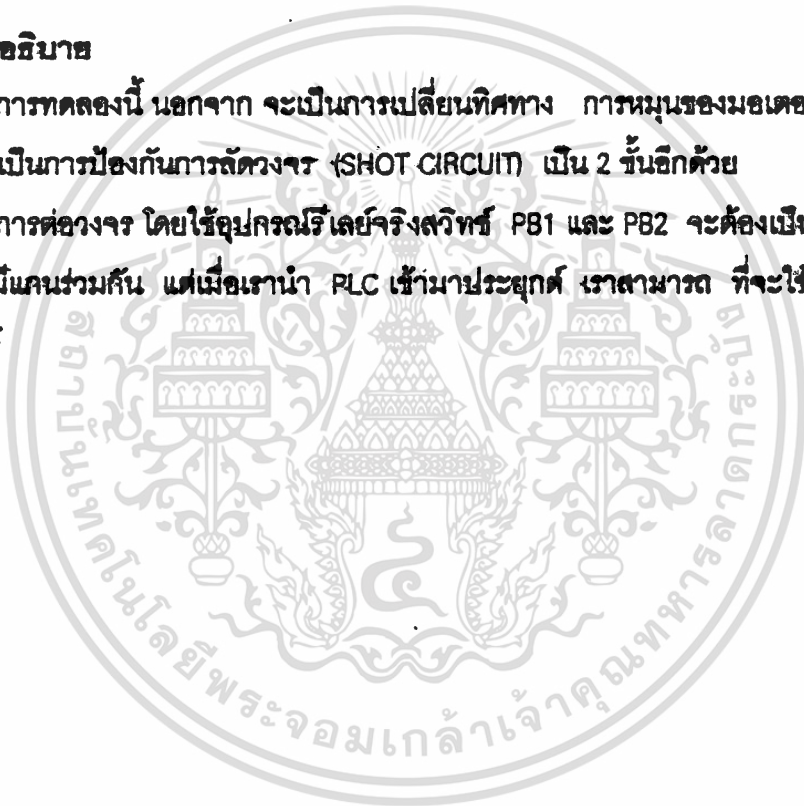
ลำดับที่	คำสั่ง	อุปกรณ์
000	LD	001
001	OR	100
002	AND-NOT	000
003	AND-NOT	002
004	AND-NOT	102
005	OUT	100
006	OUT	101
007	LD	002
008	OR	102
009	AND-NOT	000
010	AND-NOT	001
011	AND-NOT	100

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

012	OUT	102
013	OUT	103
014	LD-NOT	100
015	AND-NOT	102
016	OUT	104
017	END	

5. คำอธิบาย

- (1). ในการทดลองนี้ นอกจาก จะเป็นการเปลี่ยนทิศทาง การหมุนของมอเตอร์แบบทันทีทันใดแล้ว ยังเป็นการป้องกันการลัดวงจร (SHOT CIRCUIT) เป็น 2 ชั้นอีกด้วย
- (2). ในการต่อวงจร โดยใช้อุปกรณ์รีเลย์จริงสวิตช์ P81 และ P82 จะต้องเป็นสวิตช์แบบ 2 คอนแทคที่มีแกนร่วมกัน แต่เมื่อเรานำ PLC เข้ามาประยุกต์ เราสามารถ ที่จะใช้สวิตช์แบบ 1 คอนแทคได้

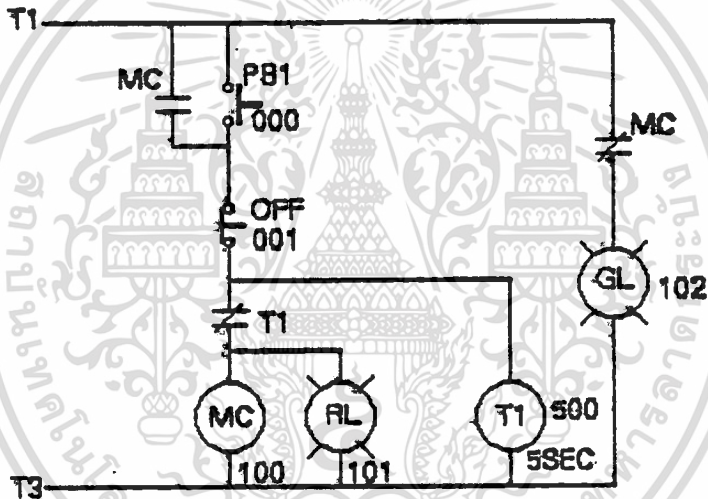


การทดลองที่ 5 การหมุนมอเตอร์ชั่วคราว

1. ลักษณะงาน

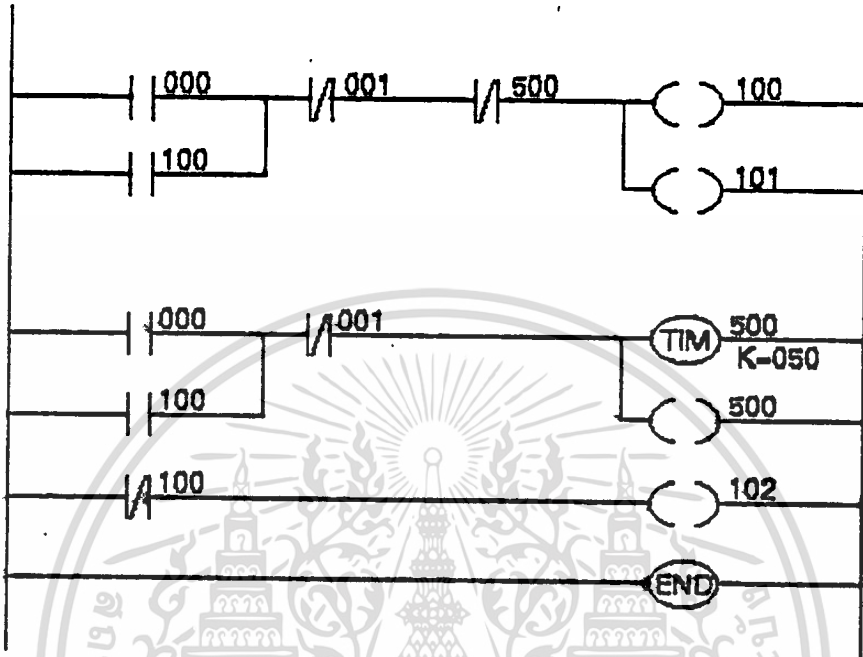
- (1). เมื่อกดปุ่ม PB1 มอเตอร์จะทำงานในช่วงเวลาที่กำหนด แล้วหยุดเองโดยอัตโนมัติ
- (2). มีไฟแสดงการทำงานของมอเตอร์ โดยไฟสีแดง (RL) จะสว่าง เมื่อมอเตอร์ทำงาน และไฟสีเขียว (GL) จะสว่างเมื่อมอเตอร์ไม่ได้ทำงาน

2. วงจรรีเลย์



วงจรรีเลย์การหมุนของมอเตอร์ชั่วคราว

3. วงจรแลตเตอร์



วงจรถัดเตอร์การหมุนของมอเตอร์ช้าลง

4. โปรแกรมบดติน

ลำดับที่	คำสั่ง	อุปกรณ์
000	LD	000
001	OR	100
002	AND-NOT	001
003	AND-NOT	500
004	OUT	100
005	OUT	101
006	LD	000
007	OR	100
008	AND-NOT	001
009	TIM	500
010	K	050

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

011	OUT	500
012	LD-NOT	100
013	OUT	102
014	END	

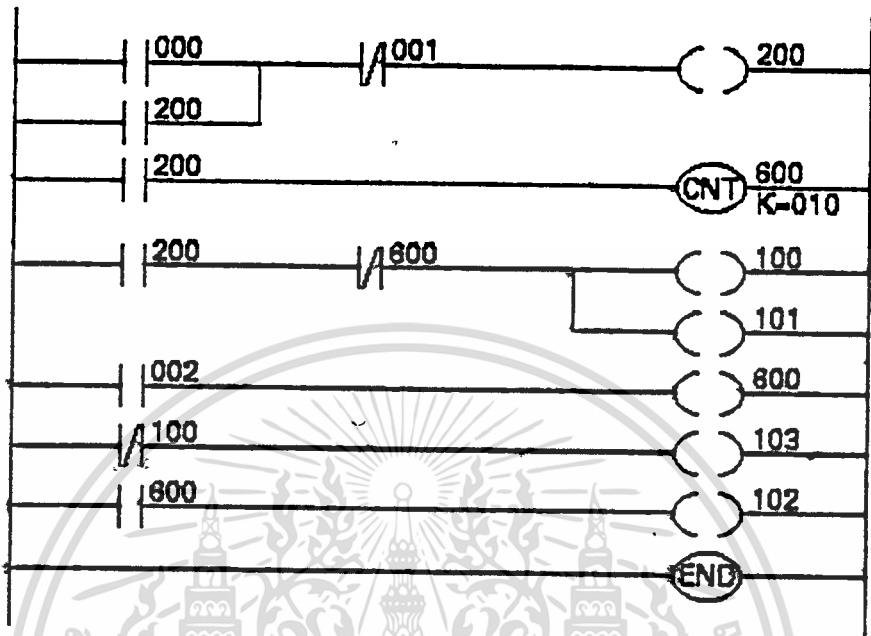
5. คำอธิบาย

(1). คำสั่ง ลำดับที่ 9-10 เป็นการตั้งค่าคงที่ให้กับอุปกรณ์หน่วงเวลา (TIMER) เนื่องจาก TIMER ที่ใช้มีความละเอียด 0.1 วินาที ดังนั้นเมื่อเราต้องการให้ TIMER หน่วงเวลา 5 วินาที เราต้องตั้งค่าคงที่เท่ากับ 50

(2). คำสั่ง ลำดับที่ 11 เป็นคำสั่ง (ENABLE) ให้ TIMER ทำการหน่วงเวลาตามปกติ



3. วงจรแลตเตอร์



วงจรถ่ายแลตเตอร์เครื่องตรวจนับสิ่งของ

4. โปรแกรมมอดูลีน

ลำดับที่	คำสั่ง	อุปกรณ์
000	LD	000
001	OR	200
002	AND-NOT	001
003	OUT	200
004	LD	200
005	CNT	600
006	K	010
007	LD	200
008	AND-NOT	600
009	OUT	100
010	OUT	101
011	LD	002

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

012	OUT	600
013	LD-NOT	100
014	OUT	103
015	LD	600
016	OUT	102
017	END	

5. คำอธิบาย

(1). ในการทดลอง นี้เราใช้รีเลย์ภายใน (AUXILIARY RELAY) เข้ามาช่วยในบรรทัดที่ 0-3 เพื่อให้การเขียนวงจรทำได้ง่ายขึ้น ซึ่งรีเลย์ภายในนี้จะไม่มีการต่อเข้ากับอุปกรณ์ภายนอกจริง ๆ

(2). บรรทัดที่ 4-6 เป็นการเริ่มสั่งให้อุปกรณ์นับจำนวน (COUNTER) ทำงานพร้อมตั้งค่าเริ่มต้น (SET POINT) ให้กับ COUNTER

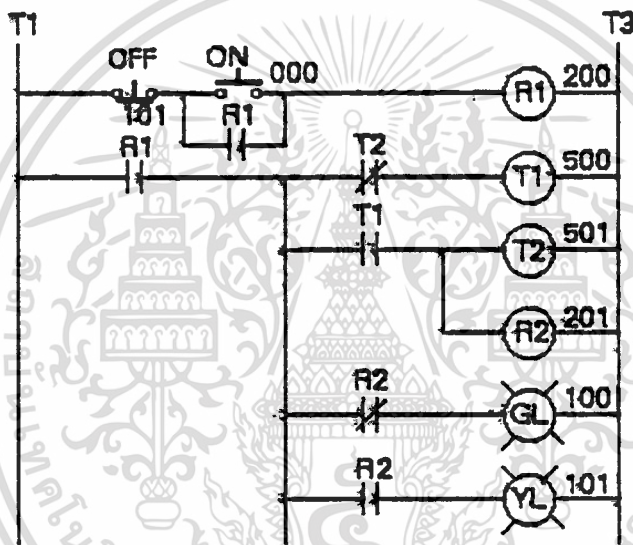
(3). บรรทัดที่ 11-12 จะเป็นการส่งสัญญาณที่ได้จากอุปกรณ์ตรวจจับ (SENSOR) ให้กับ COUNTER เพื่อทำงานนับจำนวน

การทดลองที่ 7 การเปิด-ปิดเป็นจังหวะ

1. ลักษณะงาน

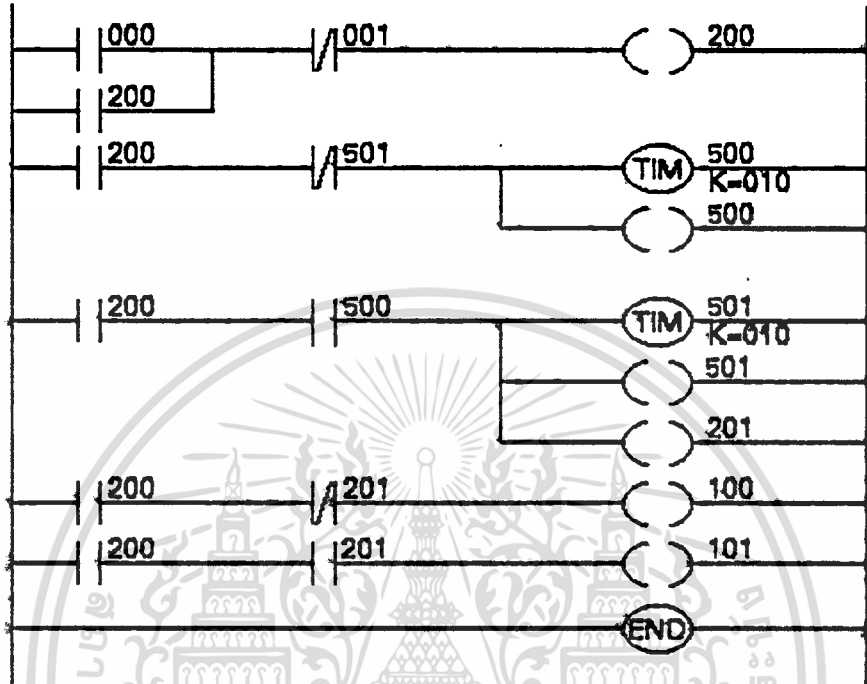
- (1). เมื่อกดปุ่ม ON ให้หลอด GL ไฟสว่าง 1 วินาที หลอดไฟ YL สว่าง 1 วินาที สลับกันไปมา
- (2). เมื่อกดปุ่ม OFF หลอดไฟดับทั้งคู่

2. วงจรรีเลย์



วงจรรีเลย์การเปิด-ปิดเป็นจังหวะ

3. วงจรแลดเดอร์



วงจรแลดเดอร์การปิด-เปิดเป็นจังหวะ

4. ไปจนกรมชุดสั้น

ลำดับที่	คำสั่ง	อุปกรณ์
000	LD	000
001	OR	200
002	AND-NOT	001
003	OUT	200
004	LD	200
005	AND-NOT	501
006	TIM	500
007	K	010
008	OUT	500
009	LD	200

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

010	AND	500
011	TIM	501
012	K	010
013	OUT	501
014	OUT	201
015	LD	200
016	AND-NOT	201
017	OUT	100
018	LD	200
019	AND	201
020	OUT	101
021	END	

5. คำอธิบาย

(1). เมื่อเริ่มทำงาน T1 และ T2 จะ OFF ทั้งคู่ ดังนั้นกระแสไฟจะจ่ายให้กับ T1 เมื่อเวลาผ่านไป 1 วินาที T1 จะทำงานทำให้ มีกระแสไฟจ่ายเข้า T2 1 วินาทีต่อมา T2 ทำงานเป็นการตัดกระแสไฟฟ้าที่เข้า T1 ทำให้ T1 OFF ทำให้ T2 OFF ด้วยการทำงานจะวนเวียนอย่างนี้ไปเรื่อย ๆ

(2). การนำ AUXILIARY RELAY หมายเลข 201 เข้ามาช่วยเพราะ TIMER เปลี่ยนแปลงได้ทุกครั้ง เมื่อเวลานำผลของ TIMER มาควบคุมมากกว่า 1 จุดในโปรแกรมจึงควรนำค่าสถานะของ TIMER ในขณะใดขณะหนึ่งมาเป็นค่า ในการควบคุม โดยเก็บค่าไว้ใน AUXILIARY RELAY

การทดลองที่ 8 การเพิ่มรีเซ็ตของอุปกรณ์นับจำนวน

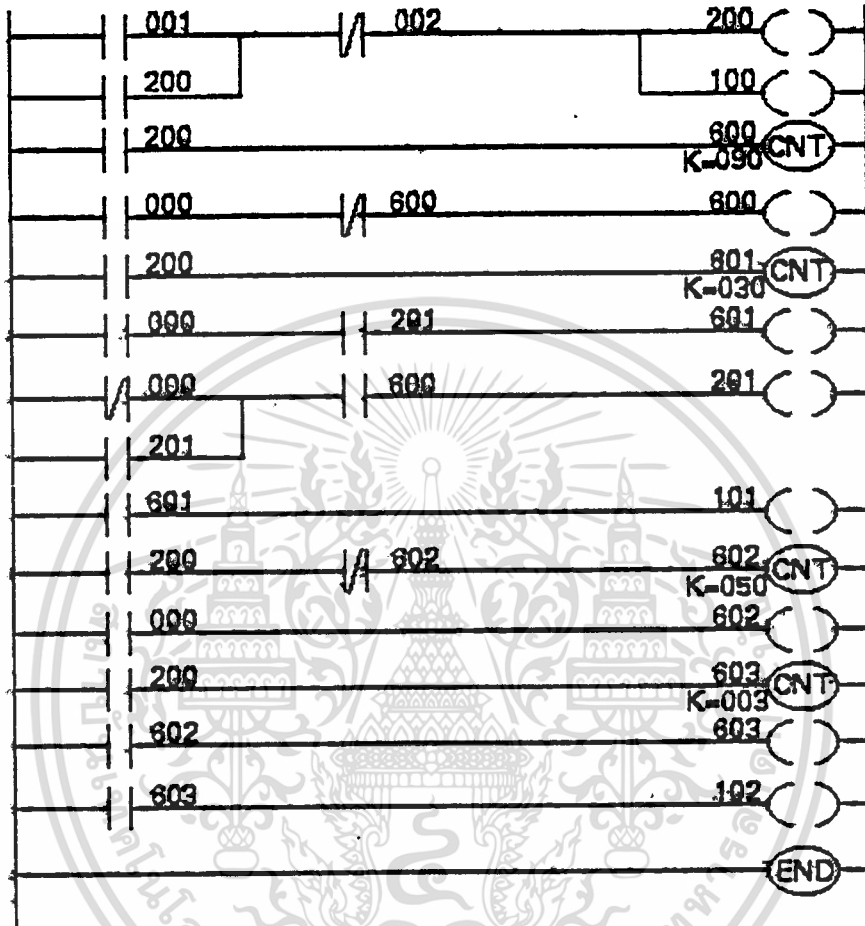
1. ลักษณะงาน

- (1). เครื่องนับจำนวนจะส่งเสียงเตือน เมื่อนับได้ 120 และ 150 ครั้ง
- (2). จะเริ่มนับได้ต่อเมื่อกดปุ่ม ON
- (3). เมื่อกดปุ่ม OFF เครื่องจะหยุดการนับ

2. การกำหนดอุปกรณ์อินพุต-เอาต์พุต

อุปกรณ์	หน้าที่
000	เหตุการณ์ที่ทำकारนับ
001	ปุ่มเริ่มการทำงาน (ON)
002	ปุ่มหยุดการทำงาน (OFF)
100	ไฟแสดงสถานะเครื่องจักร
101	สัญญาณเมื่อนับได้ 120 ครั้ง
102	สัญญาณเมื่อนับได้ 150 ครั้ง

3. วงจรแลตเตอร์



วงจรแลตเตอร์การเพิ่มขีดจำกัดของอุปกรณ์นับจำนวน

4. โปรแกรมบลูติน

ลำดับที่	คำสั่ง	อุปกรณ์
000	LD	001
001	OR	200
002	AND-NOT	002
003	OUT	200
004	OUT	100
005	LD	200
006	CNT	600
007	K	090

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

008	LD	000
009	AND-NOT	600
010	OUT	600
011	LD	200
012	CNT	601
013	K	030
014	LD	000
015	AND	201
016	OUT	601
017	LD-NOT	000
018	OR	201
019	AND	600
020	OUT	201
021	LD	601
022	OUT	101
023	LD	200
024	AND-NOT	602
025	CNT	602
026	K	050
027	LD	000
028	OUT	602
029	LD	200
030	CNT	603
031	K	003
032	LD	602
033	OUT	603
034	LD	603
035	OUT	102
036	END	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5. คำอธิบาย

(1). COUNTER 600 และ 601 เป็นการเพิ่มจำนวนนับในวิธีของการบวก โดยในช่วงแรก 600 จะทำการนับก่อน เมื่อถึงค่าสูงสุดที่กำหนดไว้ 600 จะส่งส่วนที่เหลือให้ 601 นับต่อ นั่นคือจำนวนนับรวมจะมีค่าเท่ากับ $600+601$

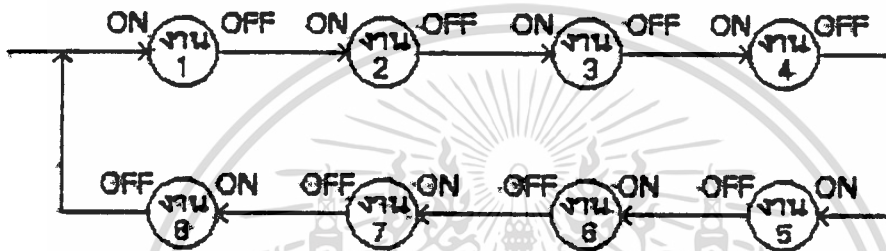
(2). COUNTER 602 และ 603 จะเป็นการเพิ่มจำนวนนับ ในวิธีของการคูณ โดยที่ 602 จะเป็นตัวนับเหตุการณ์จริง เมื่อนับถึงค่าที่กำหนด 602 จะส่งสัญญาณไปให้ 603 แล้วทำการเริ่มนับใหม่ โดย 602 จะส่งสัญญาณให้กับ 603 ทุก ๆ ครั้งที่นับถึงค่าที่กำหนด ส่วน 603 จะทำการนับว่า 602 ส่งสัญญาณมาให้กี่ครั้ง นั่นคือจำนวนนับรวมจะมีค่าเท่ากับ 602×603



การทดลองที่ 9 การควบคุมงานที่ทำงานตามลำดับ

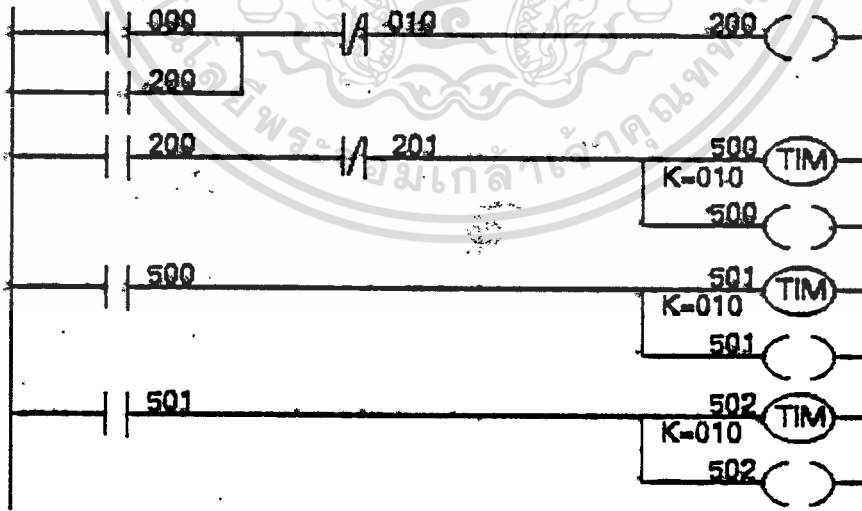
1. ลักษณะงาน

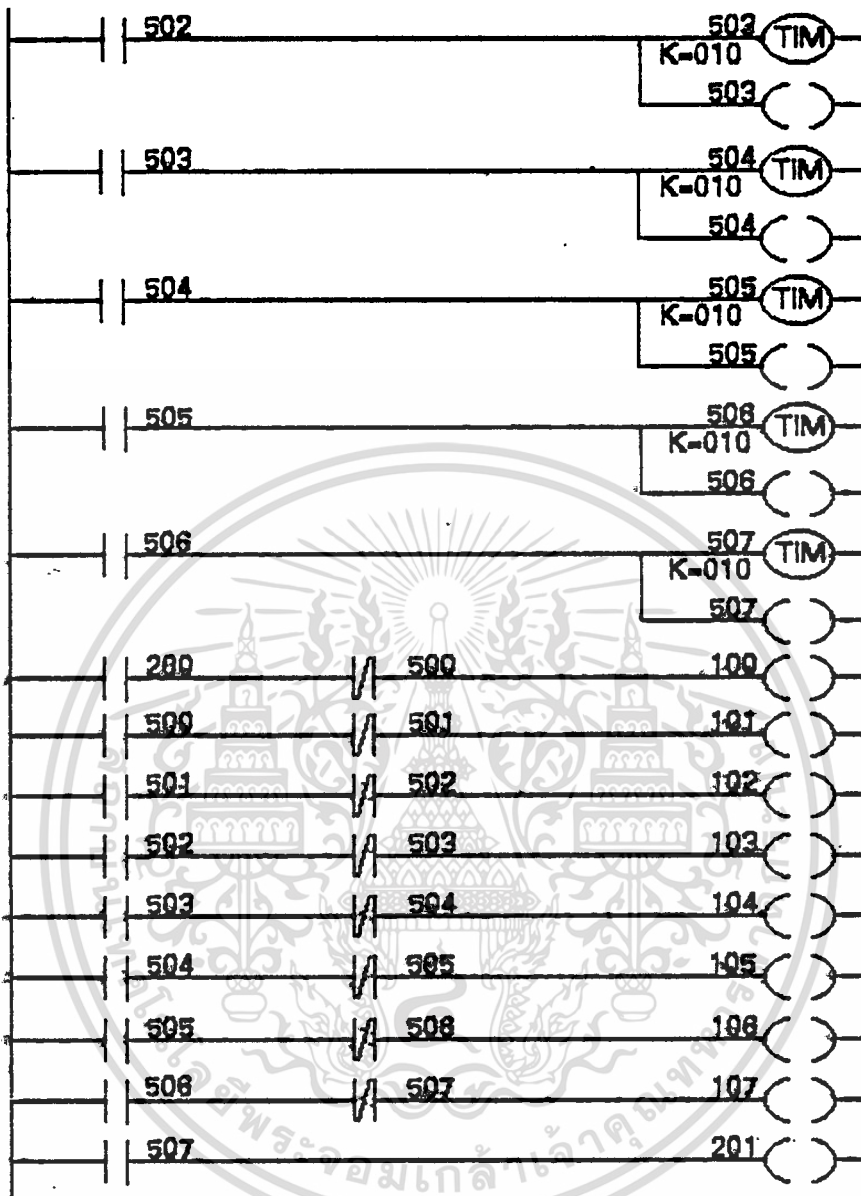
- (1). เมื่อจ่ายไฟ เครื่องจะยังไม่ทำงาน เมื่อกดปุ่ม ON (000) เครื่องจะเริ่มทำงานที่ 1,2,3,...8 แล้วกลับมาทำงานที่ 1 ใหม่
- (2).. ต้องกดปุ่ม OFF (000) ก่อน ถึงจะเริ่มกดปุ่ม ON ใหม่ได้



วงจรการควบคุมการทำงานตามลำดับ

2. วงจรลadder





วงจรแลตเตอร์การควบคุมงานที่ทำงานตามลำดับ

3. โปรแกรมบุดิติน

ลำดับที่	คำสั่ง	อุปกรณ์
000	LD	000
001	OR	200
002	AND-NOT	001
003	OUT	200

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

004	LD	200
005	AND-NOT	201
006	TIM	500
007	K	010
008	OUT	500
009	LD	500
010	TIM	501
011	K	010
012	OUT	501
013	LD	501
014	TIM	502
015	K	010
016	OUT	502
017	LD	502
018	TIM	503
019	K	010
020	OUT	503
021	LD	503
022	TIM	504
023	K	010
024	OUT	504
025	LD	504
026	TIM	505
027	K	010
028	OUT	505
029	LD	505
030	TIM	506
031	K	010
032	OUT	506

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

033	LD	506
034	TIM	507
035	K	010
036	OUT	507
037	LD	200
038	AND-NOT	500
039	OUT	100
040	LD	500
041	AND-NOT	501
042	OUT	101
043	LD	501
044	AND-NOT	502
045	OUT	102
046	LD	502
047	AND-NOT	503
048	OUT	103
049	LD	503
050	AND-NOT	504
051	OUT	104
052	LD	504
053	AND-NOT	505
054	OUT	105
055	LD	505
056	AND-NOT	506
057	OUT	106
058	LD	506
059	AND-NOT	507
060	OUT	107
061	LD	507

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

062

OUT

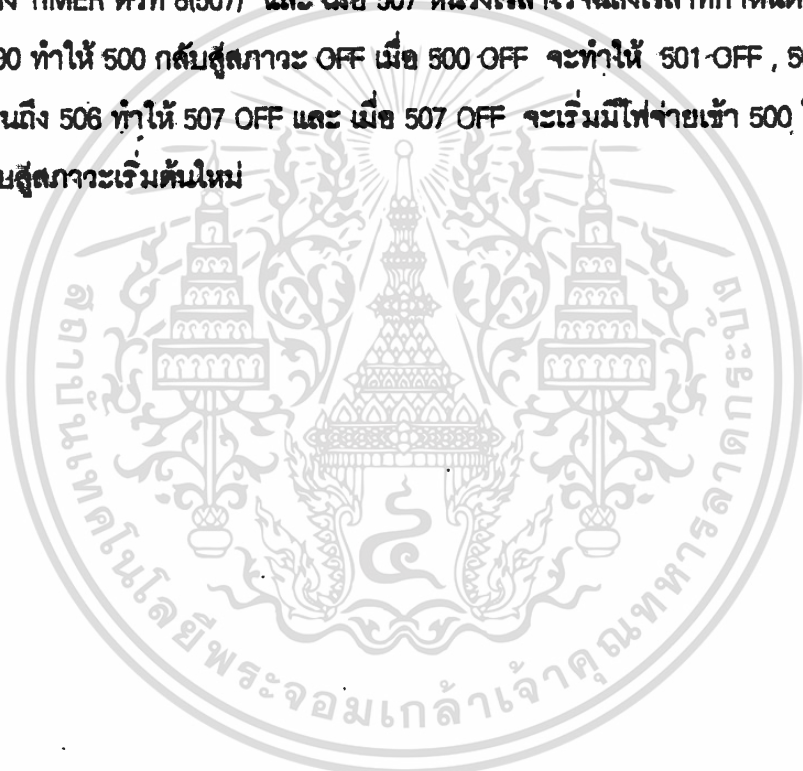
201

063

END

4. คำอธิบาย

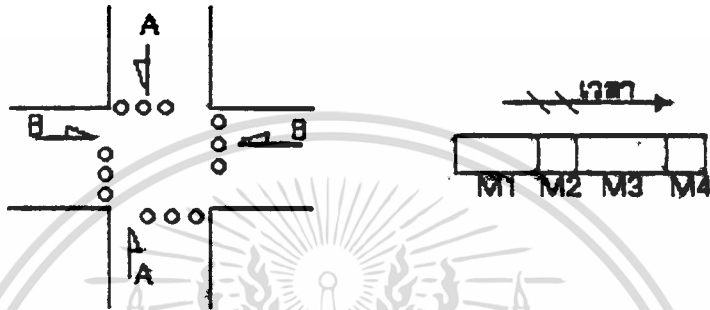
(1). ในการทดลองนี้ เราให้อุปกรณ์หน่วงเวลา (TIMER) เป็นตัวกระตุ้น ให้มีการเปลี่ยนแปลงสถานะการทำงาน เริ่มแรกเมื่อกดปุ่ม ON(000) จะเป็นการกระตุ้นให้ TIMER ตัวที่ 1(500) ทำงาน และ เมื่อถึงเวลาที่กำหนด 500 จะไปกระตุ้นให้ TIMER ตัวที่ 2 (501) ทำงานตาม ซึ่งจะเป็นเช่นนี้ไปเรื่อยๆ จนถึง TIMER ตัวที่ 8(507) และ เมื่อ 507 หน่วงเวลาไว้จนถึงเวลาที่กำหนด ก็จะตัดไฟที่จ่ายให้กับ 500 ทำให้ 500 กลับสู่สถานะ OFF เมื่อ 500 OFF จะทำให้ 501-OFF, 501 OFF ทำให้ 502 OFF จนถึง 506 ทำให้ 507 OFF และ เมื่อ 507 OFF จะเริ่มมีไฟจ่ายเข้า 500 ใหม่ ทำให้การทำงานกลับสู่สถานะเริ่มต้นใหม่



การทดลองที่ 10 การควบคุมไฟจราจร

1. ลักษณะงาน

(1). ทำการควบคุมสัญญาณไฟจราจรในแยกต่าง



ลักษณะการควบคุมไฟจราจร

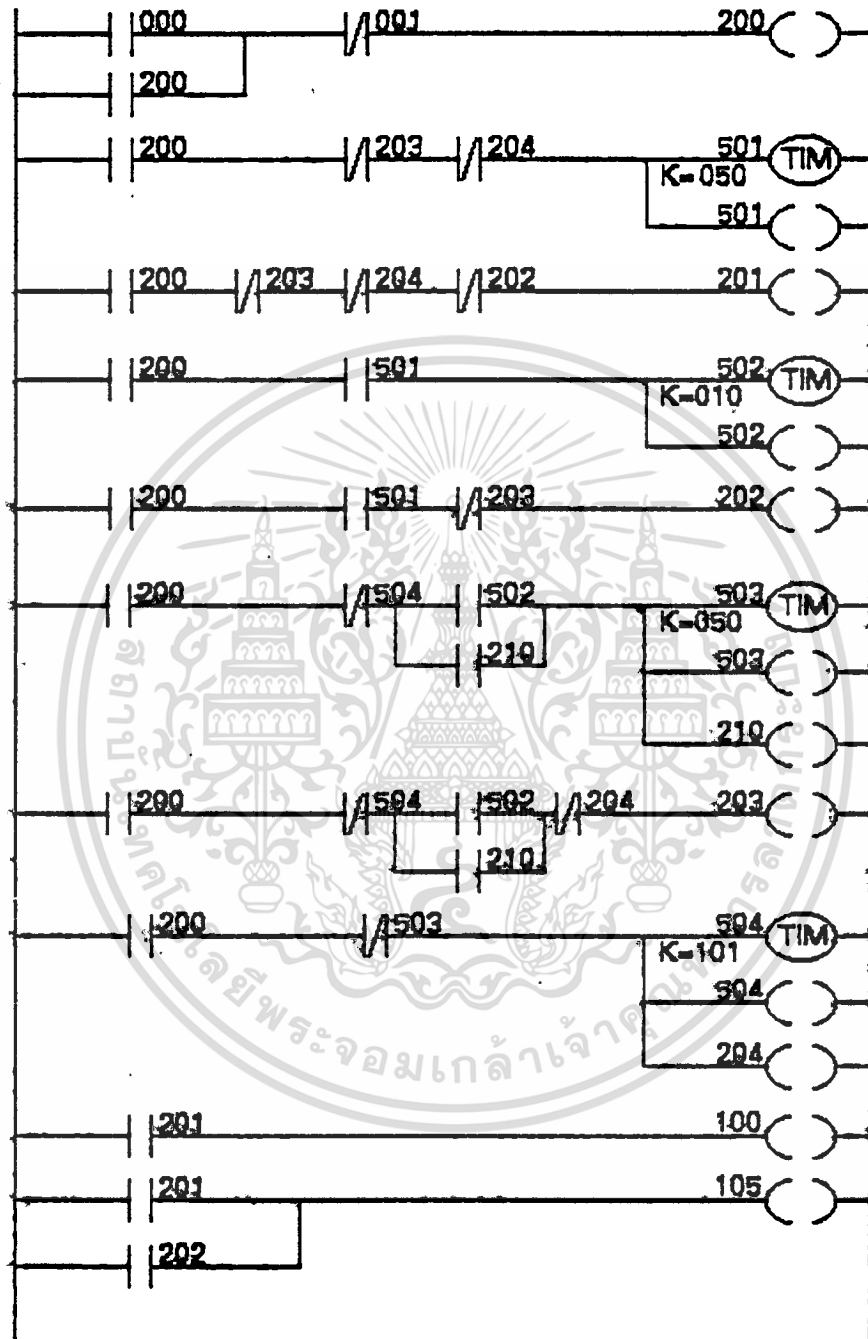
(2). ปลั๊กยรตทีละ 2 ช่องทาง

2. การกำหนดอุปกรณ์อินพุต-เอาท์พุต

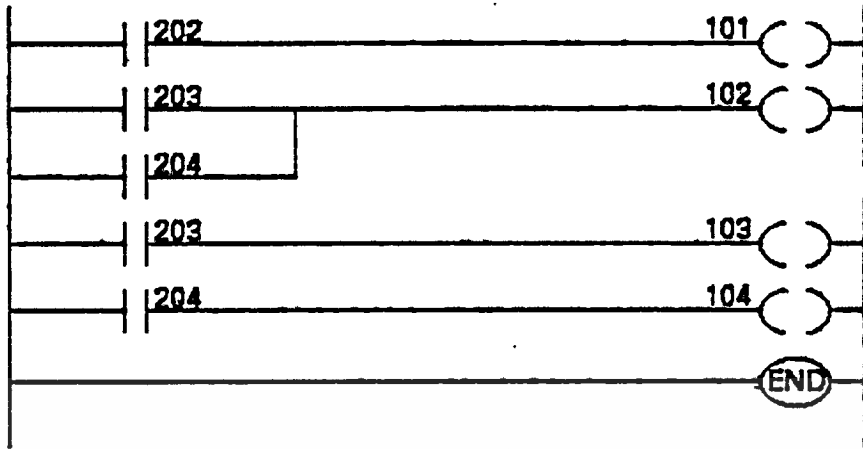
อุปกรณ์	หน้าที่
000	ปุ่มเริ่มทำงาน ON
001	ปุ่มหยุดทำงาน OFF
100	ไฟเขียวของแยก A
101	ไฟเหลืองของแยก A
102	ไฟแดงของแยก A
103	ไฟเขียวของแยก B
104	ไฟเหลืองของแยก B
105	ไฟแดงของแยก B

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. วงจรแลตเตอร์



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



วงจรแลตเตอร์ควบคุมไฟจราจร

3. โปรแกรมบดลิน

ลำดับที่

คำสั่ง

อุปกรณ์

000	LD	600
001	OR	200
002	AND-NOT	001
003	OUT	200
004	LD	200
005	AND-NOT	203
006	AND-NOT	204
007	TIM	501
008	K	050
009	OUT	501
010	LD	200
011	AND-NOT	203
012	AND-NOT	204
013	AND-NOT	202
014	OUT	502
015	LD	201
016	AND	200

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

017	TIM	502
018	K	010
019	OUT	502
020	LD	200
021	AND	501
022	AND-NOT	203
023	OUT	202
024	LD	200
025	AND-NOT	504
026	LD	502
027	OR	210
028	ANB	
029	TIM	503
030	K	050
031	OUT	503
032	OUT	210
033	LD	200
034	AND-NOT	504
035	LD	502
036	OR	210
037	ANB	
038	AND-NOT	204
039	OUT	203
040	LD	200
041	AND-NOT	503
042	TIM	504
043	K	010
044	OUT	504
045	OUT	204

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

046	LD	201
047	OUT	100
048	LD	201
049	OR	202
050	OUT	105
051	LD	202
052	OUT	101
053	LD	203
054	OR	204
055	OUT	102
056	LD	203
057	OUT	103
058	LD	204
059	OUT	104
060	END	

4. คำอธิบาย

(1). ช่วงเวลาที่ M1, M2 ในตารางเวลาของข้อที่ 1 คือ ช่วงเวลาไฟเขียว และ ไฟเหลืองของแยก A หรือ รวมกันเป็นช่วงเวลาไฟแดงของแยก B

(2). เราใช้ TIMER 501, 502, 503 และ 504 เป็นตัวตั้งเวลาช่วงเปลี่ยนสัญญาณไฟจากเขียว A-เหลือง A - เขียว B - เหลือง B ตามลำดับซึ่งเราสามารถกำหนดระยะเวลาของสัญญาณไฟได้จากค่า K ของ TIMER แต่ละตัว.

บรรณานุกรม

1. กฤษคำ วิศวกรรมันท์ “PC ตัวควบคุมซีเควีนซ์ หลักการทำงานและการประยุกต์”
กรุงเทพมหานคร จุฬาลงกรณ์มหาวิทยาลัย, 2534
2. พรจิต ประทุมสุวรรณ และคณะ “ทฤษฎีและการใช้งาน PROGRAMMABLE
CONTROLLER” กรุงเทพมหานคร โรงพิมพ์เรือนแก้วการพิมพ์, 2536
3. “MICROPROCESSOR DATA BOOK TECHNICAL MANUAL Z80180 Z180 MPU”
ETT CO., LTD.
4. FRANK D. PETRUZZELLA “PROGRAMMABLE LOGIC CONTROLLERS”
McGraw-Hill, 1989

