ภาควิชาครุศาสตร์วิศวกรรม

คณะครุศาสตร์อุตสาหกรรม

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ใบรับรองปริญญานิพนธ์

ชื่อหัวข้อ เครื่องพีแอลซี

Programmable Logic Controller

ชื่อนักศึกษา

1. นายกฤษฎา	จรัสวานิชย์ผล	รหัสประจำตัว	44035434
2. นายจตุพล	พลายโถ	รหัสประจำตัว	44035437
3. นายเฉลิมชาติ	เวียงคำ	รหัสประจำตัว	44035438
4. นายชูเกียรติ	เนื้อไม้	รหัสประจำตัว	44035440

หลักสูตร ครุศาสตร์อุตสาหกรรมบัณฑิต **สาขาวิชา** เทคโนโลยีการวัดคุมทางอุตสาหกรรม

อาจารย์ที่ปรึกษา ดร.สุรสิทธิ์ ราตรี

อาจารย์ที่ปรึกษาร่วม อาจารย์สุรพงษ์ สิริพงศ์ดี

คณะกรรมการสอบปริญญานิพนธ์		ลายมือชื่อ
1. ดร.สุรสิทธิ์	ราตรี	.....
2. อาจารย์สุรพงษ์	สิริพงศ์ดี	.....
3. อาจารย์สุระชัย	พิมพ์สาลี	.....
4. อาจารย์อมรชัย	ชัยชนะ	.....
5. อาจารย์อำพล	ทองระอา	.....

วัน/เดือน/ปีที่สอบ วันศุกร์ที่ 15 พฤศจิกายน พ.ศ. 2545 เวลา 13.00 น.

สถานที่สอบ ห้อง ค.311 คณะครุศาสตร์อุตสาหกรรม สจล.

ภาควิชารับรองแล้ว

ลงนาม.....

(ผศ.วิสุทธิ์ อธิพรธรรม)

หัวหน้าภาควิชาครุศาสตร์วิศวกรรม

วันที่ 20 เดือน ธ.ค. พ.ศ. 2545

<BT45031B2>
เครื่องพีแอลซี

เอกสารนี้ใช้เพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของลิขสิทธิ์ทุกครั้งที่มีการนำไปใช้

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

ปริญญานิพนธ์

เครื่องพีแอลซี

PROGRAMMABLE LOGIC CONTROLLER

T048319

นายกฤษฎา จรัสวานิชย์ผล
นายจตุพล พลายโถ
นายเฉลิมชาติ เวียงคำ
นายชูเกียรติ เนื้อไม้

เลขหมู่.....
เลขทะเบียน 48319
วัน, เดือน, ปี 10 ต.ค. 2546

.b.....
.i.....

ปริญญานิพนธ์ฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรครุศาสตร์อุตสาหกรรมบัณฑิต
สาขาวิชาเทคโนโลยีการวัดคุมทางอุตสาหกรรม
ภาควิชาครุศาสตร์วิศวกรรม คณะครุศาสตร์อุตสาหกรรม
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2545

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญานิพนธ์

เรื่อง เครื่องพีแอลซี

Programmable Logic Controller

จุดประสงค์

1) เพื่อศึกษาระบบควบคุมการทำงานของเครื่อง PLC
2) เพื่อออกแบบวงจรควบคุมการทำงานของเครื่อง PLC
3) เพื่อการสร้างเครื่อง PLC
4) เพื่อนำเครื่อง PLC ไปใช้งานจริง

ประโยชน์ที่คาดว่าจะได้รับ

1) มีความรู้ในเรื่องของ PLC
2) ได้ต้นแบบวงจรของเครื่อง PLC
3) ได้เครื่องต้นแบบ PLC
4) ได้ผลการทดลองและการทำงานของเครื่อง PLC
5) นำเครื่อง PLC ไปประยุกต์ใช้งานได้จริง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ชื่อหัวข้อ	เครื่องพีแอลซี
นักศึกษา	นายกฤษฎา จรัสวานิช์ผล
	นายจตุพล พลายโถ
	นายเฉลิมชาติ เวียงคำ
	นายชูเกียรติ เนื้อไม้
อาจารย์ที่ปรึกษา	ดร. สุรสิทธิ์ ราตรี
อาจารย์ที่ปรึกษาร่วม	อาจารย์สุรพงษ์ สิริพงศ์ดี
หลักสูตร	ครุศาสตร์อุตสาหกรรมบัณฑิต
สาขาวิชา	เทคโนโลยีการวัดคุมทางอุตสาหกรรม
ปีการศึกษา	2545

บทคัดย่อ

ปัจจุบันเครื่องควบคุมแบบ "ซีเควนซ์" ได้มีใช้กันอย่างแพร่หลายในโรงงานอุตสาหกรรมทั้งในขนาดกลางและขนาดใหญ่ ระบบการควบคุมแบบเก่า โดยลักษณะการควบคุมในปัจจุบันจะเน้นเรื่องการใช้งานได้ง่าย แต่มีประสิทธิภาพในการทำงานสูงทำงานที่ซับซ้อนได้และสามารถโปรแกรมการทำงานได้

โครงงาน นี้เป็นการสร้างเครื่องพีแอลซี ซึ่งมีการทำงานเหมือน PLC ที่ใช้ในโรงงานอุตสาหกรรมและใช้ในการศึกษาขั้นเบื้องต้น เครื่องควบคุมที่ออกแบบใช้ไมโครคอนโทรลเลอร์ MCS – 51 เป็นตัวประมวลผล มีหน่วยป้อนโปรแกรม สามารถต่อออกไปใช้งานกับอุปกรณ์ภายนอกได้ เปลี่ยนแปลงลักษณะและเงื่อนไขการควบคุมได้ง่าย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Thesis Title	Programmable Logic Controller
Students	Mr. Kitsada Jarunvanispon
	Mr. Jatupon Playtho
	Mr. Chslermchart Wiangkham
	Mr. Chookeart Nuamai
Advisor	Dr. Surasit Ratree
Co-Advisor	Mr. Surapong Siripongdee
Education Level	Bachelor of Science in Industrial Education
Program in	Industrial Instrument Technology
Acadamic Year	2002

ABSTRACT

Recently programmable sequence control are many used in factory in factory both medium and large side substitude 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 MCS-51 to be central processing unite and there are input unite and can be connect to extermal device and simple to change the condition of control.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

ปริญญานิพธ์ฉบับนี้สำเร็จลุล่วงไปได้ด้วยดี เนื่องมาจากความร่วมมือของสมาชิกภายในกลุ่มทุกท่าน ขอขอบคุณท่าน ดร. สุรสิทธิ์ ราตรี อาจารย์ สุรพงษ์ สิริพงศ์ดี และคณาจารย์ภาควิชาครุศาสตร์วิศวกรรมทุกท่านที่ให้คำแนะนำ ให้คำปรึกษา แนวความคิด แนวทางแก้ไขปัญหาความรู้ทางด้านเทคนิคต่างๆ ตลอดจนให้ความอนุเคราะห์เครื่องมือ และ อุปกรณ์ต่างๆ ในการจัดทำปริญญานิพนธ์ ขอขอบคุณห้องสมุดคณะครุศาสตร์อุตสาหกรรม ห้องสมุดคณะวิศวกรรมศาสตร์ที่ช่วยอำนวยความสะดวกและเอื้อเฟื้อสถานที่ในการค้นคว้าข้อมูล ขอขอบคุณพี่ๆและเพื่อนๆทุกคนที่ให้คำปรึกษา กำลังใจ ตลอดจนช่วยแก้ปัญหาต่างๆ สุดท้ายที่ควรระลึกถึงอย่างยิ่ง บิดา และมารดาที่เป็นผู้ให้ความสนับสนุนด้านการศึกษา และเป็นผู้ให้กำลังใจด้วยดีตลอดมา ตั้งแต่อดีตจนถึงปัจจุบัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

เรื่อง	หน้า
บทคัดย่อภาษาไทย	I
บทคัดย่อภาษาอังกฤษ	II
กิตติกรรมประกาศ	III
สารบัญ	IV
สารบัญตาราง	VII
สารบัญรูป	VIII
บทที่ 1 บทนำ	1
1.1 ความเป็นมา และความสำคัญของปริญญานิพนธ์	1
1.2 ขีดความสามารถของโครงงาน	1
1.3 เนื้อหาโดยสังเขป	2
บทที่ 2 ทฤษฎีและการทำงาน	3
2.1 กล่าวนำ	3
2.2 การทำงานของพีแอลซี	3
2.3 การทำงานของวงจรแลดเดอร์	7
2.3.1 หลักการทำงานของ PLC	7
2.3.2 การปรับปรุงหรือการแก้ไขวงจร	8
2.4 โครงสร้างพีแอลซี	9
2.5 อุปกรณ์เสมือนภายในพีแอลซี	10
2.6 สัญลักษณ์ในวงจรแลดเดอร์	10
2.7 ภาษาของพีแอลซี	11
2.7.1 อุปกรณ์และเบอร์อุปกรณ์	11
2.7.2 ชุดคำสั่ง	11
2.7.3 ตัวอย่างการใช้คำสั่ง	13
2.7.4 คำสั่งและการโปรแกรม	14
2.8 หน่วยประมวลผลพีแอลซี	15
2.9 หน่วยความจำพีแอลซี	15
2.9.1 PLC แบ่งหน่วยความจำออกเป็น 2 ส่วน	15

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

เรื่อง	หน้า
2.9.2 หน่วยความจำที่ใช้กับพีแอลซี	16
2.9.3 หน่วยความจำชนิดต่างๆ	16
2.10 หน่วยจ่ายกำลังงานพีแอลซี	17
2.11 คุณลักษณะของหน่วยอินพุตและเอาต์พุตพีแอลซี	17
2.11.1 คุณลักษณะทางไฟฟ้า	18
2.11.2 คุณสมบัติของสัญญาณอินพุต	18
2.11.3 คุณลักษณะเอาต์พุต	19
2.12 หน่วยป้อนโปรแกรมให้พีแอลซี	19
2.13 ข้อเปรียบเทียบระหว่าง Z-80 และ 8051	19
2.13.1 การพัฒนาไมโครคอนโทรลเลอร์	21
2.14 ผังการทำงาน	23
บทที่ 3 การออกแบบ การสร้าง และการทำงาน	26
3.1 ส่วนของฮาร์ดแวร์ (HARD WERE)	26
3.1.1 การถอดรหัสตำแหน่งของหน่วยความจำ	26
3.1.2 การต่อ MCS-51 เข้ากับหน่วยความจำแบบ EPROM และ RAM	27
3.1.3 การเชื่อมต่อ 8255 กับ MCS-51	27
3.1.4 CONTROLLER วงจรรวม	29
3.1.5 ส่วนของแหล่งจ่ายกำลังงาน	30
3.1.6 ส่วนของอินพุต	31
3.1.7 วงจรเอาต์พุต	32
3.1.8 วงจร KEY BOARD	33
3.1.9 วงจร DISPLAY LCD	34
3.2 ส่วนของซอฟต์แวร์ (SOFT WARE)	34
3.2.1 ลักษณะโครงสร้าง	36
3.2.2 MAIN PROGRAM	39
3.2.3 โปรแกรมย่อย SET PROGRAM	40
3.2.4 โปรแกรมย่อย RUN PROGRAM	41

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

เรื่อง	หน้า
บทที่ 4 การทดลองและผลการทดลอง	42
4.1 การทดลองการทำงานของโปรแกรมสแกนคีย์และการแสดงผลรวมทั้งทดลองปุ่ม FUNCTION พิเศษต่างๆ	42
4.1.1 การทดลองการใช้คีย์ที่ใช้ในการเขียนโปรแกรมและแสดงผล	42
4.1.2 การกดปุ่ม FUNCTION	43
4.2 การทดลองการทำงานของโปรแกรมย่อย	45
4.2.1 การทดลองโปรแกรมย่อย LD	45
4.2.2 การทดลองโปรแกรมย่อย LD-NOT	47
4.2.3 การทดลองโปรแกรมย่อย AND และ AND-NOT	48
4.2.4 การทดลองโปรแกรมย่อย OR และ OR-NOT	49
4.2.5 การทดลองโปรแกรมย่อย ANB, ORB	49
4.2.6 การทดลองโปรแกรมย่อย SET, RES	50
4.2.7 การทดลองโปรแกรมย่อย TIMER	51
4.2.8 การทดลองโปรแกรมย่อย COUNTER	52
บทที่ 5 บทสรุป ปัญหา แนวทางแก้ไขและพัฒนา	53
5.1 บทสรุป	53
5.2 ปัญหาและอุปสรรค	53
5.3 แนวทางการพัฒนา	54
ภาคผนวก ก เครื่องต้นแบบ	55
ภาคผนวก ข วงจรและแผ่นวงจรพิมพ์	58
ภาคผนวก ค ผังการทำงานและโปรแกรม	66
ภาคผนวก ง รายละเอียดและคุณสมบัติของอุปกรณ์	107
บรรณานุกรม	146
ประวัติ	147

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

ตาราง	หน้า
ตารางที่ 2.1 สัญลักษณ์ในวงแลดเดอร์	10
ตารางที่ 2.2 ข้อเปรียบเทียบระหว่าง Z-80 และ 8051	20
ตารางที่ 2.3 ขนาด 4 บิตไมโครคอนโทรลเลอร์	21
ตารางที่ 2.4 ขนาด 8 บิต ไมโครคอนโทรลเลอร์	21
ตารางที่ 2.5 ขนาด 16 บิตไมโครคอนโทรลเลอร์	23
ตารางที่ 3.1 CPU STATUS INDICATORS	34
ตารางที่ 4.1 ทดลองการรับคีย์บอร์ดที่ใช้ในการเขียนโปรแกรม LADDER	42

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป

รูป	หน้า
รูปที่ 2.1 ตัวอย่างการกวนของเหลว	4
รูปที่ 2.2 วงจรการใช้รีเลย์	4
รูปที่ 2.3 วงจรการใช้ PLC (อินพุต โมดูล)	5
รูปที่ 2.4 วงจรการใช้ PLC (เอาต์พุต โมดูล)	6
รูปที่ 2.5 วงจรแลดเดอร์ (PLC LADER LOGIC DIAGRAM)	7
รูปที่ 2.6 วงจรรีเลย์ที่แก้ไขใหม่	8
รูปที่ 2.7 แลดเดอร์ลอจิกไดอะแกรมของ PLC ที่แก้ไขใหม่	8
รูปที่ 2.8 โครงสร้างพีแอลซี	9
รูปที่ 2.9 คำสั่ง LD, LDI และ OUT	14
รูปที่ 2.10 การเขียนโปรแกรมลงในซอฟต์แวร์	14
รูปที่ 2.11 THE COMMONLY USED FLOWCHARTING SYMBOLS	24
รูปที่ 3.1 โครงสร้างพีแอลซีที่สร้างขึ้น	26
รูปที่ 3.2 การต่อ MCS-51 เข้ากับ EPROM	27
รูปที่ 3.3 การเชื่อมต่อ 8255 กับ MCS-51	28
รูปที่ 3.4 บอร์ดหลัก	29
รูปที่ 3.5 แหล่งจ่ายไฟตรง 5 โวลท์	30
รูปที่ 3.6 วงจรอินพุต 8 วงจร	31
รูปที่ 3.7 วงจรเอาต์พุต 8 วงจร	32
รูปที่ 3.8 วงจร KEY BOARD	33
รูปที่ 3.9 วงจร DISPLAY LCD	34
รูปที่ 3.10 ขั้นตอนการทำงานของโปรแกรมภาษาแอสเซมบลีในไมโครคอนโทรลเลอร์	35
รูปที่ 3.11 ลักษณะโครงสร้าง	36
รูปที่ 3.12 ขั้นตอนการใช้งานของพีแอลซี	38
รูปที่ 3.13 ผังการทำงาน MAIN PROGRAM	39
รูปที่ 3.14 โปรแกรมย่อย SET PROGRAM	40
รูปที่ 3.15 โปรแกรมย่อย RUN PROGRAM	41
รูปที่ 4.1 หน้าจอ LCD เมื่อกดปุ่ม MOD	42

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป (ต่อ)

รูป	หน้า
รูปที่ 4.2 หน้าจอเมื่อสิ้นสุดการเขียนโปรแกรม	43
รูปที่ 4.3 การทดลองปุ่ม MOD	43
รูปที่ 4.4 หน้าจอเมื่อกดปุ่ม 	43
รูปที่ 4.5 การกดปุ่ม INS	44
รูปที่ 4.6 การกดปุ่ม DEL	44
รูปที่ 4.7 การกดปุ่ม 	45
รูปที่ 4.8 การกดปุ่ม 	45
รูปที่ 4.9 โปรแกรมทดลองโปรแกรมย่อย	46
รูปที่ 4.10 โปรแกรมทดลองโปรแกรมย่อย LD-NOT	47
รูปที่ 4.11 โปรแกรมทดลองโปรแกรมย่อย AND, AND-NOT	48
รูปที่ 4.12 โปรแกรมทดลองโปรแกรมย่อย OR, OR-NOT	49
รูปที่ 4.13 โปรแกรมทดลองโปรแกรมย่อย ANB, ORB	50
รูปที่ 4.14 โปรแกรมทดลองโปรแกรมย่อย SET, RES	51
รูปที่ 4.15 โปรแกรมทดลองโปรแกรมย่อย TIMER	52
รูปที่ 4.16 โปรแกรมทดลองโปรแกรมย่อย COUNTER	52
รูปที่ ก.1 เครื่องพีแอลซี	56
รูปที่ ก.2 ด้านหลังของ KEY BOARD, LCD, SWITCHINPUT	56
รูปที่ ก.3 บอร์ดหลัก MCS-51	57
รูปที่ ก.4 ภายในเครื่องพีแอลซี	57
รูปที่ ข.1 การต่อ MCS-51 เข้ากับ EPORM และ RAM	59
รูปที่ ข.2 การเชื่อมต่อ 8255 กับ MCS-51	60
รูปที่ ข.3 บอร์ดหลัก	61
รูปที่ ข.4 แหล่งจ่ายไฟตรง 5 โวลท์	62
รูปที่ ข.5 วงจรอินพุต 8 วงจร	62
รูปที่ ข.6 วงจรเอาต์พุต 8 วงจร	63
รูปที่ ข.7 วงจร KEY BOARD	64
รูปที่ ข.8 วงจร DISPLAY LCD	65

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป (ต่อ)

รูป	หน้า
รูปที่ ค.1 ขั้นตอนการทำงานของโปรแกรมภาษาแอสเซมบลีในไมโครคอนโทรลเลอร์	67
รูปที่ ค.2 ลักษณะโครงสร้าง	68
รูปที่ ค.3 ขั้นตอนการทำงานของพีแอลซี	69
รูปที่ ค.4 ผังการทำงาน MAIN PROGRAM	70
รูปที่ ค.5 โปรแกรมย่อย SET PROGRAM	71
รูปที่ ค.6 โปรแกรมย่อย RUN PROGRAM	72
รูปที่ ค.7 โปรแกรมการทำงานของเครื่อง PLC	106

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1
บทนำ

1.1 ความเป็นมา และความสำคัญของปริญญานิพนธ์

พีแอลซี (PLC) ย่อมาจากคำว่า "Programmable Logic Controller" เป็นอุปกรณ์ควบคุมอิเล็กทรอนิกส์ที่มีหน่วยความจำในการเก็บโปรแกรม สำหรับควบคุมการทำงานของอุปกรณ์ต่าง ๆ ที่ต่อกับขั้วเข้า และขั้วออกของมัน

พีแอลซีได้รับการพัฒนา เพื่อใช้ทดแทนระบบควบคุมแบบเก่าที่ใช้รีเลย์ ซึ่งติดตั้งดัดแปลงและแก้ไขลำบาก มาเป็นระบบควบคุมใหม่ ที่ใช้วงจรอิเล็กทรอนิกส์แทนรีเลย์และใช้การเขียนโปรแกรม ทำนองเดียวกับการเขียนโปรแกรมคอมพิวเตอร์ กำหนดเงื่อนไขการควบคุมแทนการเดินสายเชื่อมต่อวงจรไฟฟ้าแบบเก่าเพื่อเพิ่มความสะดวกระยะแรก

ระยะแรกพีแอลซีถูกใช้แทนวงจรรีเลย์ในการควบคุมเครื่องจักรที่มีการควบคุมแบบ เปิดและปิด เท่านั้น หลังจากนั้น พีแอลซี มีขนาดเล็ก นำหนักเบา ติดตั้งบำรุงรักษาง่ายใช้ไฟฟ้าน้อยกว่ารีเลย์ทำให้มีผู้นำ พีแอลซี มาใช้ในโรงงานอุตสาหกรรมต่าง ๆ เพิ่มขึ้น จึงมีแนวความคิดที่จะสร้างเครื่อง พีแอลซี ขึ้นมา ซึ่งภายในจะประกอบด้วยไมโครคอนโทรลเลอร์ ซึ่งเหมาะสมที่จะศึกษาไมโครคอนโทรลเลอร์ แต่ยังไม่ค่อยมีความรู้เรื่องไมโครคอนโทรลเลอร์ เท่าไรนักและจะศึกษาการใช้เครื่อง พีแอลซี พร้อมไปด้วย

1.2 ขีดความสามารถของโครงงาน

โครงงานนี้มีความสามารถดังนี้

1) มีจอแสดงผล (Display) LCD แบบ ขนาด 16 X 2 ตัวอักษร

2) มีแป้นพิมพ์ (Keyboard) ขนาด 5 X 4 เป็นอย่างน้อย

3) มีจำนวนอินพุต 8 อินพุต (5-25 โวลท์ 50 มิลลิแอมป์)

4) มีจำนวนเอาต์พุต 16 เอาต์พุต (5 โวลท์)

5) มีอุปกรณ์หน่วงเวลา (Timer) 2 ตัว

6) มีอุปกรณ์นับจำนวน (Counter) 2 ตัว

7) มีตัวแปลงสัญญาณอนาลอกเป็นดิจิตอล (A/D) 1 ชุด

8) มีคำสั่งพื้นฐาน ตัวอย่างเช่น AND NAND OR NOR NOT

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.3 เนื้อหาโดยสังเขป

เนื้อหาภายในปริญญานิพนธ์ฉบับนี้แบ่งออกเป็นบทต่างๆ เพื่อความสะดวกต่อการศึกษาและทำความเข้าใจ ในแต่ละบทจะประกอบด้วยเนื้อหาที่สำคัญดังนี้

บทที่ 1 บทนำ เป็นเนื้อหาเกี่ยวกับความเป็นมาและความสำคัญของปัญหา ที่ทำให้เกิดโครงงานนี้ขึ้น

บทที่ 2 ทฤษฎีและหลักการ ประกอบด้วยเนื้อหาในทางทฤษฎีที่เกี่ยวข้อง ซึ่งทำให้ผู้อ่านได้มีความรู้ความเข้าใจที่เป็นพื้นฐานเสียก่อน อันจะเป็นประโยชน์ต่อการทำความเข้าใจกับวงจรที่ใช้งานจริงต่อไป

บทที่ 3 การออกแบบการสร้างและการทำงาน โดยกล่าวถึงการสร้าง และการออกแบบฮาร์ดแวร์ และซอฟต์แวร์ รวมทั้งหลักการทำงานในส่วนต่างๆ ซึ่งจะทำให้ผู้อ่านมีความเข้าใจการทำงานได้ตรงตามวัตถุประสงค์หรือไม่

บทที่ 4 การทดลองและผลการทดลอง กล่าวถึงขั้นตอนการทดลองและการทดสอบประสิทธิภาพ ในการทำงานของฮาร์ดแวร์ และซอฟต์แวร์ของโครงงานนี้ เพื่อตรวจสอบว่าโครงงานนี้สามารถทำงานได้ตรงตามวัตถุประสงค์ หรือไม่

บทที่ 5 บทสรุปปัญหาแนวทางแก้ไขและพัฒนา เป็นการสรุปผลการทำงาน และได้เสนอแนะแนวทางในการแก้ไข และแนวทางในการพัฒนา ให้มีประสิทธิภาพ และการใช้งานได้อย่างกว้างขวางมากขึ้น

ในภาคผนวกแสดงรายละเอียดของโปรแกรม และรายการของอุปกรณ์ต่างๆ ที่ใช้จัดทำโครงงานนี้

ภาคผนวก ก เครื่องต้นแบบ

ภาคผนวก ข วงจรและแผ่นวงจรพิมพ์

ภาคผนวก ค ผังการทำงานและโปรแกรม

ภาคผนวก ง รายละเอียด และคุณสมบัติของอุปกรณ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2
ทฤษฎีและหลักการทำงาน

2.1 กล่าวนำ

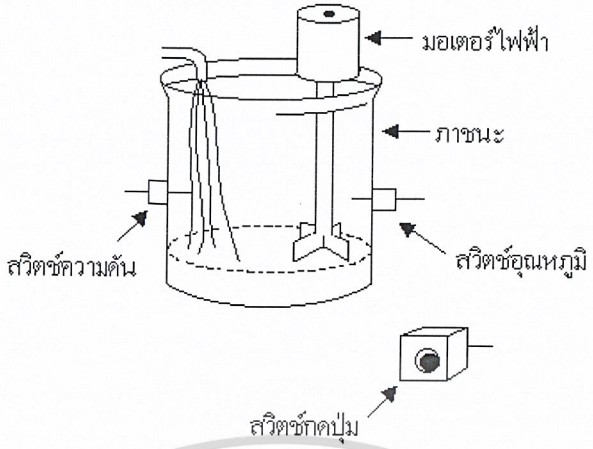
เนื้อหาของปริญญานิพนธ์ในบทนี้เป็นทฤษฎีและหลักการที่นำมาใช้ประกอบการสร้างโครงงาน โดยพีแอลซีเป็นอุปกรณ์ที่สำคัญ ในการควบคุมเครื่องจักรในโรงงานอุตสาหกรรมให้ทำงานอย่างอัตโนมัติ ในระบบ FA (Factor Automation) พีแอลซี จะใช้ในการปรับปรุงประสิทธิภาพการทำงานของเครื่องจักร ลดของเสีย ทำให้เครื่องจักรทำงานได้เองโดยอัตโนมัติเป็นการลดงานของโรงงาน พีแอลซี ที่ใช้ในโรงงานตั้งแต่ พีแอลซี ที่มีขนาดใหญ่หรือเป็นระบบควบคุม จนกระทั่งถึง พีแอลซี ขนาดเล็กซึ่งใช้ตัวเดียวได้

ในบทนี้ จะเน้น พีแอลซี ขนาดเล็กที่ใช้งานตัวเดียว จะแนะนำให้รู้จักการทำงานและวิธีนำมาใช้ในโรงงานอุตสาหกรรม

2.2 การทำงานของพีแอลซี

เครื่องจักรที่ควบคุมด้วย PLC จะมีความสามารถเขียนโปรแกรม การทำงานของเครื่องจักร และมีความยืดหยุ่นในการเขียนโปรแกรม เช่น การเปลี่ยนแปลงแก้ไขเพิ่มเติมก็สามารถกระทำได้ ซึ่งรวมถึงมี ไทเมอร์ (Timer) เคาน์เตอร์ (Counter) หรือคำสั่งพิเศษต่างๆ เช่น MOV Data และอื่นๆ อีกมากมาย เพื่อใช้ควบคุมอุปกรณ์ภายนอก ไม่ว่าจะเป็นมอเตอร์ โซลีนอยด์ (Solenoid) หลอดไฟ เป็นต้น นอกจากนี้ยังมีการติดต่อสื่อสารระหว่าง PLC กับคอมพิวเตอร์ เพื่อแลกเปลี่ยนข้อมูลระหว่างกันหรืออาจจะติดต่อกับจอชนิดสัมผัส (Touch Screen) เพื่ออำนวยความสะดวกต่อสัญญาณอินพุตและสัญญาณเอาต์พุต ยิ่งไปกว่านั้นการติดต่อกับคอมพิวเตอร์เพื่อให้คอมพิวเตอร์เป็นตัวควบคุม PLC อีกทีหนึ่ง ซึ่งจะทำให้ขีดความสามารถในการควบคุมสูงขึ้นอีก และต่อไปนี้จะได้กล่าวถึงการทำงานของ PLC โดยเริ่มจากตัวอย่างการควบคุมมอเตอร์กวนของเหลวในภาชนะโดยมีเงื่อนไขต่างๆ

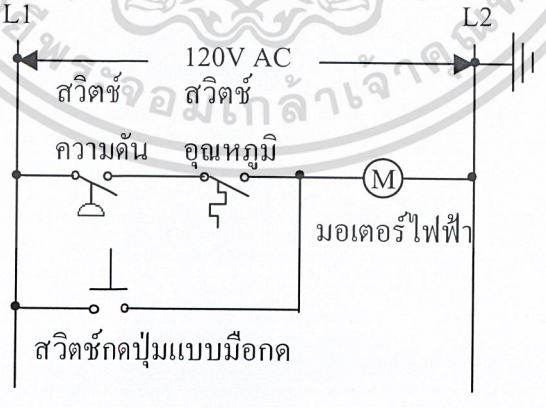
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.1 ตัวอย่างการกวนของเหลว

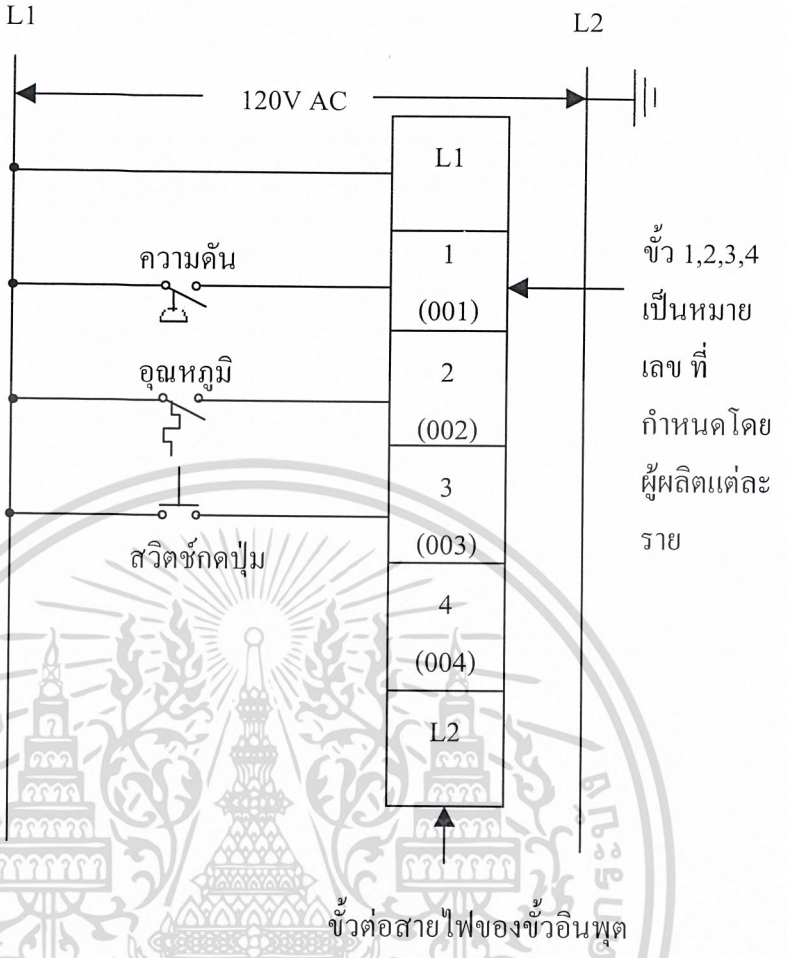
ตามรูปที่ 2.1ใช้มอเตอร์ไฟฟ้าสำหรับกวนของเหลวที่บรรจุอยู่ในภาชนะเมื่อมีเงื่อนไขว่ามีอุณหภูมิและความดันถึงค่าที่กำหนดเท่านั้น หรือสามารถให้มอเตอร์ไฟฟ้าทำงาน ด้วยการกดสวิตช์ก็ได้ โดยแยกสวิตช์กดออกมาต่างหาก

มอเตอร์ไฟฟ้าจะหมุนได้ก็ต่อเมื่อหน้าคอนแทคของเซนเซอร์อุณหภูมิ และเซนเซอร์ความดันต่อร่วมกัน การต่อวงจรทำได้จากการใช้วงจรรีเลย์ (Relay) ตามรูปที่ 2.2 และการใช้ PLC ตามรูปที่ 2.3 และ 2.4 ในกรณีของการใช้วงจรรีเลย์ มอเตอร์ (M) ทำงานได้เมื่อสวิตช์ของอุณหภูมิและความดันต่อกันหรือสวิตช์กดปุ่มแบบมือกดถูกกด



รูปที่ 2.2 วงจรการใช้รีเลย์

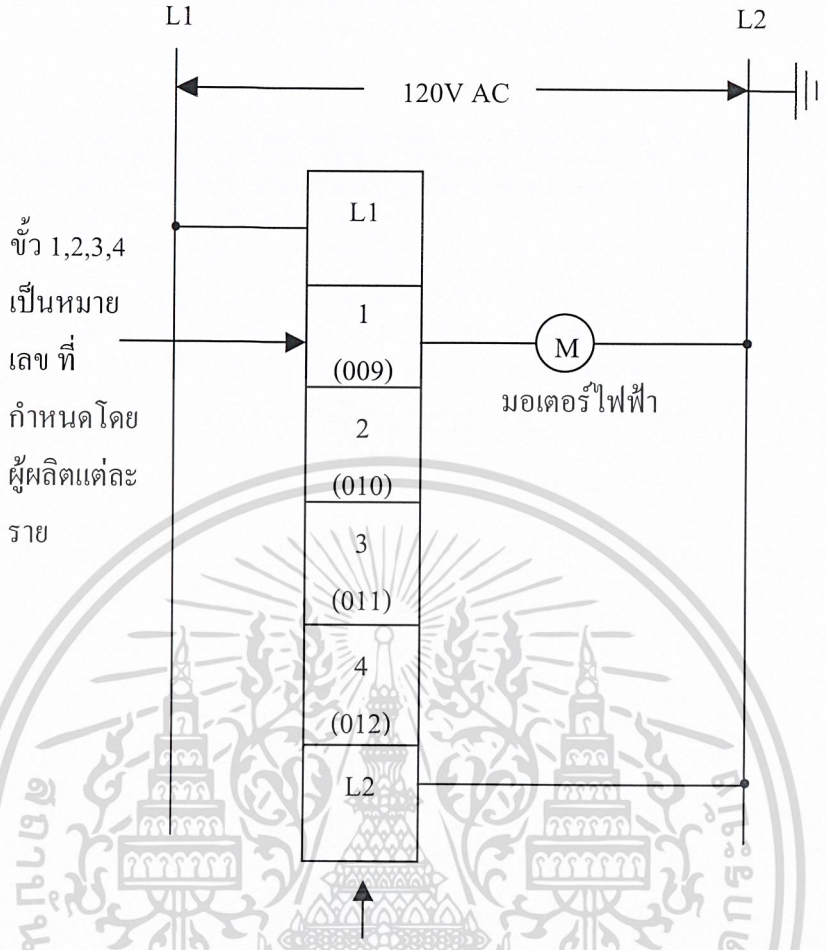
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.3 วงจรการใช้ PLC (อินพุต โมดูล)

แต่ถ้าใช้ PLC ทำงาน ตามรูปที่ 2.3 เป็นการต่ออุปกรณ์ด้านอินพุต ซึ่งมีคอนแทคของความดัน คอนแทคของอุณหภูมิ และคอนแทคของสวิตช์กดปุ่มแบบมือกด ซึ่งต่อเข้าไปยังขั้วสัญญาณเข้า (Input Module) โดย L1 ต่อเข้าที่แหล่งจ่าย ขั้ว 1 (001) ต่อเข้าสวิตช์ความดัน ขั้ว 2 (002) ต่อเข้าสวิตช์อุณหภูมิ ขั้ว 3 (003) ต่อเข้าสวิตช์กดปุ่ม และ L2 ต่อเข้าที่แหล่งจ่าย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

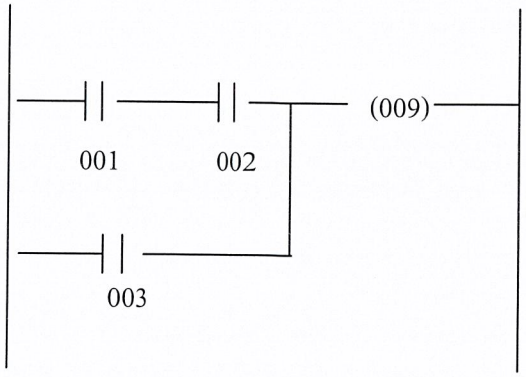


รูปที่ 2.4 วงจรการใช้ PLC (เอาต์พุต โมดูล)

ส่วนรูปที่ 2.4 (มอเตอร์ไฟฟ้า) เข้ากับขั้วต่อสัญญาณออก (Output Module) โดย L1 ต่อแหล่งจ่าย ขั้ว 1 (009) ต่อเข้าที่มอเตอร์ และ L2 เป็นขั้วต่อแหล่งจ่ายไฟของขั้วเอาต์พุต

การเขียน PLC แลดเดอร์ (PLC Ladder Logic Diagram) เพื่อส่งโปรแกรมเข้าไปยังหน่วยความจำของ CPU ดูได้จากรูปที่ 2.5

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



หมายเหตุ : หมายเลข 001, 002, 003 ใช้แทนสวิตช์ความดัน สวิตช์อุณหภูมิ และสวิตช์กดปุ่ม ตามลำดับ

รูปที่ 2.5 วงจรแลดเดอร์ (PLC LADER LOGIC DIAGRAM)

2.3 การทำงานของวงจรแลดเดอร์

เมื่อ PLC อยู่ในสภาวะพร้อมทำงาน (RUN Mode) แล้ว เมื่อมีโปรแกรมถูกป้อนเข้าไปยังหน่วยความจำของ CPU ทำให้ CPU ประมวลผลและได้ผลลัพธ์เป็นสัญญาณเอาต์พุตหน้าคอนแทคตามรูปที่ 2.5 ซึ่งเป็นชนิดปกติเปิด (Normally Open) เพราะฉะนั้น ถ้าคอนแทค 001 และ 002 ต่อกัน ก็ทำให้เกิดเอาต์พุต 009 หรือหน้าคอนแทค 003 ต่อกัน ก็ทำให้เกิดเอาต์พุต 009 ได้เช่นกัน ลักษณะนี้เรียกว่า รัง (Rung) คือ มีสัญญาณอินพุตหนึ่งหรือมากกว่าที่ทำให้เกิดเอาต์พุตหนึ่งหรือมากกว่า

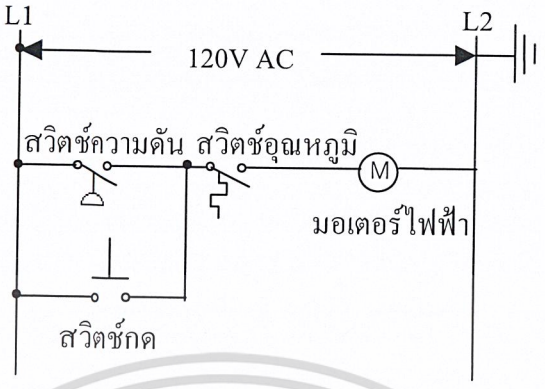
2.3.1 หลักการทำงานของ PLC

อธิบายตามเหตุการณ์ต่างๆ มีดังต่อไปนี้ คือ เมื่อมีสัญญาณอินพุตเข้ามาจะถูกเก็บเป็นความจำไว้ในส่วนของความจำ (หน้าคอนแทคที่ติดต่อกัน เรียกว่า ลอจิก 1 ส่วนคอนแทคที่ไม่ติดต่อกัน เรียกว่า ลอจิก 0 หลังจากนั้นแลดเดอร์ไดอะแกรมก็จะสรุปผลรวมกับคอนแทคภายในว่าให้เป็นคอนแทคเปิด (Open) หรือปิด (Closed) ขึ้นอยู่กับการบันทึกของหน่วยความจำ ถ้าหากต้องการสัญญาณเอาต์พุต ค่าของลอจิกต้องเป็นเลข 1 ซึ่งหมายถึงชุดหน้าคอนแทคของโมดูลอินเตอร์เฟส (Module Interface) ต่อกัน แต่ถ้าไม่มีกระแสไฟฟ้าไหลในวงจร ทำให้ขดลวดความจำของคอยล์ (Coil Memory) มีค่าเป็นลอจิกเลข 0 และโมดูลอินเตอร์เฟส (Module Interface) ต่อกัน

การทำงานของ PLC เมื่อครบ 1 รอบของลำดับดังกล่าวนี้เรียกว่า สแกน (Scan) ส่วน Scan Time คือเวลาที่ต้องการสำหรับ 1 รอบการทำงาน ซึ่งเป็นตัววัดค่าความเร็วการทำงานของ PLC 1 สแกน ใช้เวลาประมาณ 1-10 ms ขึ้นอยู่กับความยาวของโปรแกรมและชนิดของอินพุตและเอาต์พุต

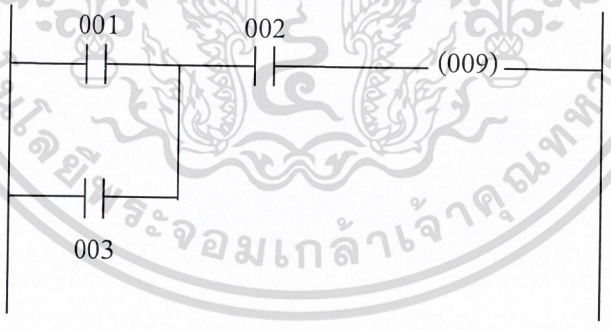
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3.2 การปรับปรุงหรือการแก้ไขวงจร



รูปที่ 2.6 วงจรรีเลย์ที่แก้ไขใหม่

จากที่กล่าวในส่วนของเครื่องผสมของเหลวมาแล้วข้างต้นนั้น ถ้าต้องการปรับปรุงวงจรเสียใหม่ตามรูปที่ 2.6 นี้ คือ เปลี่ยนแปลงในส่วนของสวิตช์กดปุ่ม (Push Button Switch) ที่สามารถทำงานได้ที่ความดันใดๆ ก็ได้แต่ต้องมีอุณหภูมิตามค่าที่กำหนดเท่านั้น ทำได้โดยสวิตช์กดปุ่มกับสวิตช์ความดันต่อกันแบบขนาน (OR)

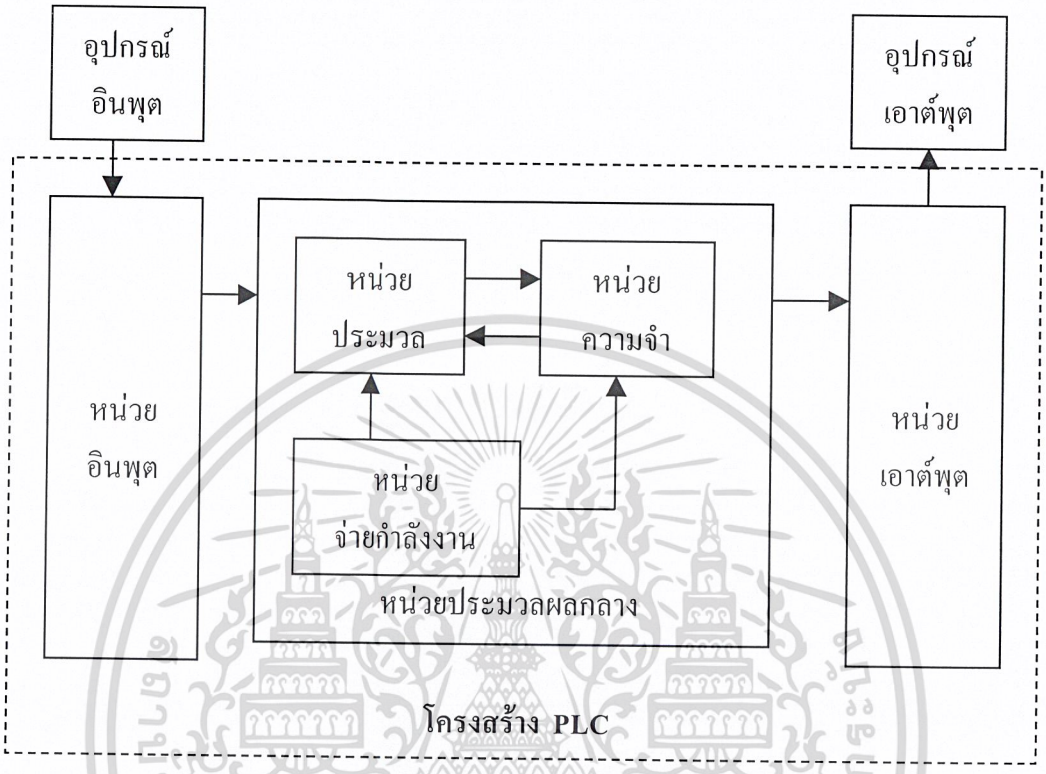


รูปที่ 2.7 แลดเดอร์ลอจิกไดอะแกรมของ PLC ที่แก้ไขใหม่

เมื่อปรับปรุงแก้ไขวงจรของรีเลย์ก็ต้องเดินสายไฟฟ้าใหม่ แต่ถ้าใช้ PLC เป็นตัวควบคุมไม่จำเป็นต้องเดินสายไฟฟ้าที่เข้ากับอุปกรณ์ใหม่ เพียงแต่เขียนแลดเดอร์ลอจิกไดอะแกรมของ PLC ใหม่เท่านั้นตามรูปที่ 2.7

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4 โครงสร้างพีแอลซี



รูปที่ 2.8 แสดงโครงสร้างพีแอลซี

PLC เป็นคอมพิวเตอร์ชนิดที่ถูกออกแบบมาเพื่อใช้ในงานควบคุมระบบต่างๆ ในโรงงานอุตสาหกรรม ซึ่งต้องทนต่ออุณหภูมิ ความชื้น ฝุ่นละออง ระบบไฟฟ้าที่มีการรบกวนไม่สม่ำเสมอ และต้องทนต่อแรงกระแทก และแรงสั่นสะเทือนที่เกิดขึ้นต่างๆได้ โครงสร้างพื้นฐาน ของ PLC ยังคล้ายคลึงกับ ระบบคอมพิวเตอร์ทั่วๆไปคือ ประกอบด้วยหน่วยอินพุต (INPUT-INTERFACE) หน่วยประมวลผลกลาง (PROCESSOR) และ หน่วยเอาต์พุต (OUTPUT INTERFACE) ซึ่งในการติดตามควบคุม การทำงานของระบบนั้น PLC ยังต้องมีการติดต่อกับอุปกรณ์ภายนอกต่างๆ คือ อุปกรณ์อินพุตและอุปกรณ์เอาต์พุต ซึ่งถึงแม้จะไม่ใช่ส่วนประกอบของ PLC แต่ก็ถือว่า เป็นส่วนหนึ่งของระบบ PLC

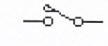
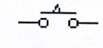
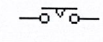
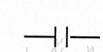
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.5 อุปกรณ์เสมือนภายในพีแอลซี

ภายใน พีแอลซี จะมีอุปกรณ์ที่ใช้ในการควบคุมซีเคว้นซ์ เช่น รีเลย์ ไทเมอร์ เคาน์เตอร์ และอุปกรณ์อื่นๆ อุปกรณ์แต่ละตัวจะมีหน้าสัมผัสทั้งชนิด ปกติปิด (NC : Norrnal Close) และปกติเปิด (NO : Normal Open) และมีหน้าสัมผัสมากมายนับไม่ถ้วน การโปรแกรมพีแอลซีให้ทำงานก็ คือ การสร้างวงจรรีเลย์โดยการนำองค์ประกอบเหล่านี้มาต่อเป็นวงจรนั่นเอง

2.6 สัญลักษณ์ในวงจรแลดเดอร์

ตารางที่ 2.1 สัญลักษณ์ในวงจรแลดเดอร์

อุปกรณ์	ชนิด NO	ชนิด NC
ปุ่มกด (กดติดปล่อยดับ)		
ปุ่มกดแบบล็อก		
รีเลย์ความร้อน(OCR)		
สวิตช์ (ทั่วไป)		
สวิตช์ (มีกลไก)		
รีเลย์		
คอนแทคเตอร์		
ไทเมอร์ (on-delay)		
ไทเมอร์ (off-delay)		
อินพุตเอาต์พุตรีเลย์ ไทเมอร์ เคาน์เตอร์ รีเลย์ช่วย		

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.7 ภาษาของพีแอลซี

ภายในพีแอลซีจะเป็นที่รวมของ รีเลย์ ไทเมอร์ เคาน์เตอร์และอุปกรณ์อื่นๆ อีกหลายๆ ตัว การต่อวงจรของอุปกรณ์เหล่านี้ ทำได้โดยการป้อนโปรแกรมผ่านทางแผงโปรแกรม การต่อวงจรของอุปกรณ์ต่างๆ เหล่านี้ นอกจากการต่อหน้าสัมผัสและคอยล์แล้ว ยังมีเกณฑ์ต่างๆ ขึ้นอยู่กับชนิดของอุปกรณ์ กฎเกณฑ์เหล่านี้ คือ คำสั่งนั้นเอง

คำสั่งประกอบด้วยรหัสคำสั่ง และเบอร์อุปกรณ์ บางคำสั่งอาจไม่ต้องมีเบอร์อุปกรณ์ก็ได้ เราจำเป็นต้องรู้เกี่ยวกับเบอร์ของอุปกรณ์ต่างๆ ที่เราสามารถใช้ได้

ในภาษาของพีแอลซีจะกล่าวถึงคำสั่งบูลลีน และคำสั่งแลดเดอร์ ที่ใช้งานจริงสำหรับ PLC ที่ได้จัดทำขึ้นนี้ พร้อมทั้งขอบเขตของอุปกรณ์ในการงาน และการโปรแกรมลงในโปรแกรมที่ได้สร้างขึ้น

2.7.1 อุปกรณ์และเบอร์อุปกรณ์

1) X : อินพุตออปโต้คอปเปลอร์ (Input Optocoupled)

เป็นอุปกรณ์ออปโต้สำหรับใช้รับสัญญาณจากสวิตซ์ตรวจจับ ซึ่งต่อเข้ามาที่ขั้วเข้าของ PLC เบอร์ของอินพุตออปโต้คอปเปลอร์นี้จะนำหน้าด้วย X เสมอ

2) Y : เอาต์พุตรีเลย์ (Output Relay)

เป็นรีเลย์ขาออกที่ใช้ในการขับโหลดที่ต่อกับขั้วออกของพีแอลซีเอาต์พุตรีเลย์จะมีเบอร์นำหน้าด้วย Y เสมอ

3) M : รีเลย์ช่วย (Auxiliary Relay)

เป็นรีเลย์ที่ใช้ในการสร้างวงจรควบคุมซีเคว้นซ์ ภายในพีแอลซีจะจัดเตรียมรีเลย์ช่วยไว้ มากมายเบอร์ของรีเลย์ช่วยนี้จะนำหน้าด้วย M เสมอ

4) T : ไทเมอร์ (Timer)

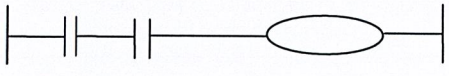
เป็นรีเลย์ที่มีการหน่วงเวลาในการปิดเปิดหน้าสัมผัส ภายในพีแอลซีจะเตรียมไทเมอร์ไว้ให้ใช้หลายตัวเบอร์ของไทเมอร์จะนำหน้าด้วย T เสมอ

5) C : เคาน์เตอร์ (Counter)

เป็นสัญญาณอินพุตและสัญญาณตั้งค่านับใหม่ (Reset) จากสัญญาณอินพุตชนิดต่างๆ ได้ไม่จำกัด

2.7.2 ชุดคำสั่ง

1) ภาษาแลดเดอร์

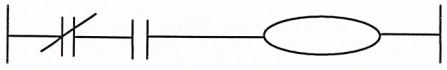


ภาษาบูลลีน LD (Load)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ความหมาย เป็นการนำค่าสภาวะที่กำหนดเข้ามาสู่โปรแกรม

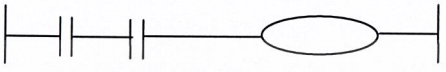
วิธีการใช้งาน ต้องกำหนดหมายเลขหรือตำแหน่งให้กับอุปกรณ์

2) ภาษาแลดเดอร์ 

ภาษาบูลลีน LDI (Load Inverse)

ความหมาย เป็นการนำค่าสภาวะที่กำหนดเข้ามาสู่โปรแกรมในลักษณะตรงข้าม

วิธีการใช้งาน ต้องกำหนดหมายเลขหรือตำแหน่งให้กับอุปกรณ์

3) ภาษาแลดเดอร์ 

ภาษาบูลลีน AND

ความหมาย เป็นการนำค่าสภาวะที่กำหนดเข้ามาทำลอจิก AND กัน

วิธีการใช้งาน ใช้เมื่อเงื่อนไขที่ต้องการเกิดในลักษณะของการอนุกรม ตั้งแต่สองขึ้นไป

4) ภาษาแลดเดอร์ 

ภาษาบูลลีน ANI (AND Inverse)

ความหมาย เป็นการนำค่าสภาวะที่กำหนดเข้ามาทำลอจิก AND Inverse กัน

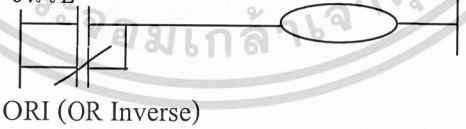
วิธีการใช้งาน ใช้เมื่อเงื่อนไขที่ต้องการเกิดในลักษณะของการอนุกรม ตั้งแต่สองขึ้นไป

5) ภาษาแลดเดอร์ 

ภาษาบูลลีน OR

ความหมาย เป็นการนำค่าสภาวะที่กำหนดเข้ามาทำลอจิก OR กัน

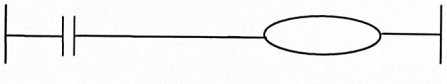
วิธีการใช้งาน ใช้เมื่อเงื่อนไขที่ต้องการเกิดในลักษณะของการอนุกรม ตั้งแต่สองขึ้นไป

6) ภาษาแลดเดอร์ 

ภาษาบูลลีน ORI (OR Inverse)

ความหมาย เป็นการนำค่าสภาวะที่กำหนดเข้ามาทำลอจิก OR Inverse กัน

วิธีการใช้งาน ใช้เมื่อเงื่อนไขที่ต้องการเกิดในลักษณะของการอนุกรม ตั้งแต่สองขึ้นไป

7) ภาษาแลดเดอร์ 

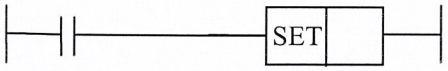
ภาษาบูลลีน OUT

ความหมาย ใช้เพื่อควบคุมสถานะของอุปกรณ์ปลายทาง ให้มีการทำงานตามเงื่อนไขข้างหน้า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

วิธีการใช้งาน ใช้เมื่อต้องการนำ ค่าสภาวะ ออกมาทางอุปกรณ์ปลายทาง ต่าง ๆ

8) ภาษาแลดเดอร์

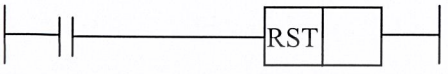


ภาษาบูลลีน SET

ความหมาย ใช้เพื่อควบคุมสถานะของอุปกรณ์ปลายทางให้มีการทำงาน ON ค้าง

วิธีการใช้งาน ใช้เมื่อต้องการนำ ค่าสภาวะ ON ออกมาทางอุปกรณ์ปลาย (ใช้ได้กับรีเลย์ช่วย M และ เอาต์พุตรีเลย์ Y เท่านั้น)

9) ภาษาแลดเดอร์



ภาษาบูลลีน RST (Reset)

ความหมาย ใช้เพื่อควบคุมสถานะของอุปกรณ์ปลายทาง ให้มีการทำงาน OFF

วิธีการใช้งาน ใช้เมื่อต้องการนำ ค่าสภาวะ OFF ออกมาทางอุปกรณ์ปลาย (ใช้ได้กับรีเลย์ช่วย M เคาน์เตอร์ C และ เอาต์พุตรีเลย์ Y เท่านั้น)

10) ภาษาแลดเดอร์



ภาษาบูลลีน END

ความหมาย คำสั่งบอกจุดจบของโปรแกรม

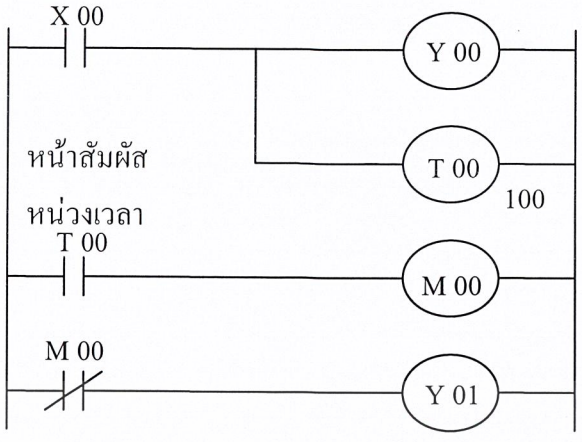
วิธีการใช้งาน พีแอลซี ทั่วไปจะทำงาน 3 จังหวะใหญ่ คือ อ่านอินพุตทำงานตามโปรแกรม แล้วจึงส่งเอาต์พุตออกไป ถ้าเราเขียนคำสั่ง END ลงในโปรแกรมพีแอลซีจะหยุดทำงานตาม โปรแกรม ณ จุดนั้น แล้วจัดส่งเอาต์พุตออกไปโดยไม่ไปสนใจโปรแกรมที่อยู่ลำดับถัดมาเลย

เราสามารถนำคำสั่ง END มาใช้ในการทดสอบโปรแกรมได้ โดยแทรกคำสั่ง END ไว้ตรงท้ายโปรแกรมแต่ละช่วง และลองทดสอบเฉพาะโปรแกรมท่อนนั้น เมื่อทดสอบว่าโปรแกรมทำงานถูกต้องก็ลบคำสั่ง END ออกและไปทดสอบโปรแกรมที่อยู่ ถัดไป ทำเช่นนี้เรื่อยๆจนสามารถทดสอบทั้งโปรแกรม

2.7.3 ตัวอย่างการใช้คำสั่ง

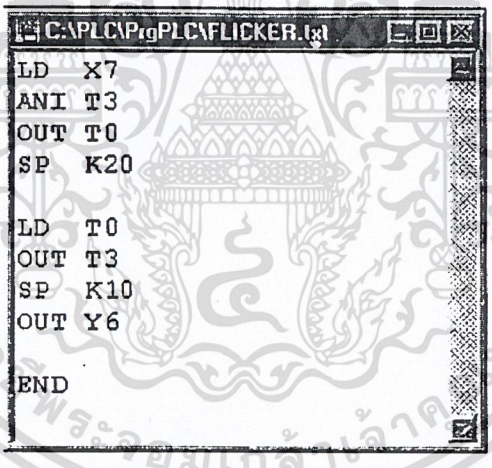
LD	LD ต่อหน้าสัมผัส a กับสายจ่ายไฟ
LDI	LDI Load Inverse
OUT	OUT คำสั่งขับคอยล์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.9 คำสั่ง LD, LDI และ OUT

2.7.4 คำสั่งและการโปรแกรม



รูปที่ 2.10 การเขียนโปรแกรมลงในซอฟต์แวร์

การโปรแกรมภาษาบูลลีนลงในซอฟต์แวร์ (Software) โดยมีรายละเอียดดังต่อไปนี้

1) โปรแกรมเกิดจากการนำคำสั่งต่าง ๆ มาเขียนเรียงกัน การเขียนเรียงลำดับของคำสั่งก็คือหมายเลขขอบรรทัด ซึ่งการเขียนโปรแกรมบูลลีนไม่จำเป็นต้องใส่ค่า

2) คำสั่งจะประกอบด้วย รหัสคำสั่ง (opcode) และเบอร์อุปกรณ์ (operand) บางคำสั่งอาจจะไม่ต้องมีเบอร์อุปกรณ์ก็มี รหัสคำสั่งอาจเรียกย่อว่าคำสั่งก็ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในกรณีของเคาน์เตอร์ และ ไทเมอร์ จะมีคำสั่งเพิ่มอีกบรรทัดเป็นการระบุค่าขอบเขตการทำงานของอุปกรณ์นั้น ๆ โดย opcode คือ SP และ operand ขึ้นต้นด้วย K และตามด้วยค่าที่ต้องการ

3) PLC จะอ่านคำสั่งในโปรแกรมเริ่มจากลำดับ 0 จนถึงลำดับที่เป็นคำสั่ง END อ่านคำสั่งและทำงานตามคำสั่งนั้น เมื่อถึงคำสั่ง END ก็จะเริ่มมาอ่านคำสั่งที่ลำดับ 0 ใหม่ วนเป็นสองรอบเช่นนี้อยู่ตลอดเวลา การทำงานหนึ่งวงรอบเรียกว่า เวลาสแกน (scan time) บางครั้งเรียกว่าเวลาหนึ่ง ไซเกิล (cycle time) เวลานี้ไม่ควรเกิน 10 msec.

2.8 หน่วยประมวลผลพีแอลซี

หน่วยประมวลผลทำหน้าที่ดูแลการทำงานทั้งหมดของพีแอลซีคือนำโปรแกรมไปใช้ (user program) มาปฏิบัติเพื่อควบคุมอุปกรณ์ภายนอกตามเงื่อนไขการควบคุมที่ผู้เขียนโปรแกรมต้องการควบคุมการติดต่อรับส่งข้อมูลระหว่าง CPU กับหน่วยอินพุตกับเอาต์พุต และหน่วยอินพุตกับเอาต์พุตกับอุปกรณ์ภายนอก ติดต่อกับผู้ใช้และอุปกรณ์ร่วม ตรวจสอบสภาพการทำงานของพีแอลซีโดยมีโปรแกรมบริหารระบบ (operating system) เป็นผู้ควบคุม

หน่วยประมวลผลพีแอลซีประกอบขึ้นจากไมโครคอนโทรลเลอร์ ซึ่งนอกจากจะทำหน้าที่ควบคุมวงจรรีเลย์ในการควบคุมแบบ ON/OFF เหมือนวงจรตรรกแล้วยังสามารถคำนวณทางคณิตศาสตร์และติดต่อกับอุปกรณ์ร่วมภายนอก

2.9 หน่วยความจำพีแอลซี

หน่วยความจำทำหน้าที่เก็บโปรแกรมบริหารระบบและข้อมูลต่างๆ ที่พีแอลซีใช้ในการประมวลผล

2.9.1 PLC แบ่งหน่วยความจำออกเป็น 2 ส่วน

1) หน่วยความจำระบบ (System Memory) เก็บโปรแกรมบริหารระบบและข้อมูลของระบบ

2) หน่วยความจำผู้ใช้ (User Memmory) เก็บโปรแกรมผู้ใช้ข้อมูลของหน่วยอินพุตและเอาต์พุต และอุปกรณ์ภายใน

หน่วยความจำของพีแอลซีมีลักษณะการใช้แตกต่างกัน บางส่วนต้องการหน่วยความจำที่มีความเร็วสูง และไม่อนุญาติให้ผู้ใช้เปลี่ยนแปลงข้อมูลภายใน หรือในบางครั้งต้องการหน่วยความจำที่เปลี่ยนแปลงแก้ไขได้แต่ต้องเก็บข้อมูลได้ถึงแม้จะไม่มีกระแสไฟฟ้าเลี้ยง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.9.2 หน่วยความจำที่ใช้กับพีแอลซี

แบ่งเป็น 2 ชนิด คือ

1) โวลาไทล์ (Volatile)

2) นอนโวลาไทล์ (Nonvolatile)

ข้อมูลในหน่วยความจำชนิดโวลาไทล์จะสูญหายหมด ถ้าไม่มีกระแสไฟฟ้าแต่หน่วยความจุชนิดนอนวาไทล์จะเก็บรักษาข้อมูลไว้ โดยปัจจุบันพีแอลซีประกอบด้วย หน่วยความจำทั้ง 2 ชนิด

2.9.3 หน่วยความจำชนิดต่างๆ

1) ROM (Read Only Memory)

เป็นหน่วยความจำที่ไม่อนุญาติให้ผู้ใช้เปลี่ยนแปลงแก้ไขข้อมูลภายใน แต่สามารถเก็บรักษาข้อมูลไว้ได้แม้ว่าจะไม่มีกระแสไฟฟ้า เหมาะสำหรับเก็บโปรแกรมระบบหรือโปรแกรมผู้ใช้ที่เสร็จสมบูรณ์และไม่ต้องการแก้ไขอีก

2) RAM (Random Access Memory)

เป็นหน่วยความจำ ที่ผู้ใช้สามาถเปลี่ยนแปลงแก้ไขข้อมูลได้ เหมาะสำหรับเก็บโปรแกรมผู้ใช้ที่อยู่ในช่วงกำลังพัฒนา หรือต้องการเปลี่ยนแปลงบ่อยครั้ง โดยมีหน่วยความจำสำรองรักษาข้อมูลไว้ RAM เป็นหน่วยความจำที่มีความเร็วสูงเมื่อเปรียบเทียบกับหน่วยความจำชนิดอื่น

3) PROM (Programmable Read Only Memory)

เป็น ROM ชนิดหนึ่งที่ผู้ใช้สามารถป้อนข้อมูลได้โดยใช้อุปกรณ์พิเศษแต่ลบหรือแก้ไขข้อมูลภายในไม่ได้พีแอลซีจะไม่นิยมใช้ PROM เพราะใช้งานไม่สะดวกและมีความเร็วต่ำ

4) EPROM (Erasable Programmable Read Only Memory)

เป็น PROM ชนิดหนึ่ง ซึ่งสามารถลบข้อมูลภายในได้โดยการฉายแสงอัลตราไวโอเลตหรือ UV (Ultra-Violet)ประมาณ 20 นาที แล้วโปรแกรมข้อมูลใหม่ EPROM เหมาะสำหรับเก็บโปรแกรมผู้ใช้ที่เสร็จสมบูรณ์แล้ว

5) EEPROM (Electrically Erasable Programmable Read Only Memory)

มีลักษณะคล้ายกับ EAROM แต่การลบข้อมูลเก่าไม่จำเป็นต้องลบทั้งหมด ผู้ใช้สามารถแก้ไขข้อมูลเฉพาะตำแหน่งที่ต้องการได้ ข้อเสียของ EEPROM คือการแก้ไขข้อมูลทำได้ช้า อายุการใช้งานค่อนข้างสั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

6) วงแหวนแม่เหล็ก (Magnatic Core)

เป็นหน่วยความจำแบบเก่าที่ใช้กับพีแอลซีรุ่นแรกๆ โดยใช้กระแสไฟฟ้าเปลี่ยนทิศทางของสนามแม่เหล็กและทิศทางการหมุนของวงแหวนแม่เหล็ก ปัจจุบันหน่วยความจำแบบนี้ไม่นิยมใช้เพราะราคาแพง ความเร็วต่ำ และมีขนาดใหญ่

7) NOVRAM (Non-Volatile Random Access Memory)

ถูกผลิตขึ้นโดยนำ RAM ที่มีความเร็วสูงและเปลี่ยนแปลงข้อมูลได้ง่าย ทำงานร่วมกับ EEPROM ซึ่งเก็บข้อมูลได้โดยไม่ต้องใช้กระแสไฟฟ้า ข้อมูลแต่ละบิตของ NOVRAM ประกอบด้วยหน่วยความจำ 2 ส่วน คือ RAM และ EEPROM การใช้งานตามปกติ RAM จะเก็บข้อมูลที่ถูกเปลี่ยนแปลงและส่งไปเก็บใน EEPROM เมื่อแก้ไขข้อมูลเสร็จเรียบร้อย ซึ่งข้อมูลจะไม่เสียหายถึงแม้จะไม่มีกระแสไฟฟ้า ข้อมูลจะถูกส่งกลับให้ RAM เมื่อต้องการเรียกใช้ข้อมูลจะถูกส่งกลับให้ RAM เมื่อต้องการเรียกใช้ข้อมูลเดิมในการแก้ไขโปรแกรมครั้งต่อไป

2.10 หน่วยจ่ายกำลังงานพีแอลซี

หน่วยจ่ายกำลังงานมีหน้าที่จ่ายกระแสไฟฟ้า ให้กับส่วนต่างๆ ของ พีแอลซี คือ หน่วยประมวลผล หน่วยความจำ และหน่วยอินพุตและเอาต์พุต โดยรักษาแรงดันไฟฟ้าให้คงที่และแจ้งให้หน่วยประมวลผลทราบเมื่อทำงานผิดปกติ

แหล่งจ่ายไฟทำหน้าที่จ่ายกำลังไฟฟ้าให้ CPU และส่วนอื่นๆ ใน PLC ส่วนมากแล้วจะมีให้เลือก 3 แบบ คือ 24 V DC 120 V AC และ 240 V AC

ขนาดของแหล่งจ่ายไฟจะมีอยู่ด้วยกัน 3 แบบ ดังที่กล่าวแล้ว ซึ่งต้องเลือกให้เหมาะสม ทั้งนี้ขึ้นอยู่กับชนิดและจำนวนของอินพุตและเอาต์พุต อัตราการกินกระแสไฟฟ้าของ CPU และ I/O Unit

2.11 คุณลักษณะของหน่วยอินพุตและเอาต์พุตพีแอลซี

ส่วนของอินพุตและเอาต์พุต (I/O Unit) จะต่อร่วมกับชุดควบคุมเพื่อรับสภาวะและสัญญาณต่างๆ เช่น หน่วยอินพุตรับสัญญาณหรือสภาวะแล้วส่งไปยัง CPU เพื่อประมวลผล CPU ประมวลผลแล้วจะส่งให้ส่วนของเอาต์พุต เพื่อให้อุปกรณ์ทำงานตามที่โปรแกรมเอาไว้

การใช้พีแอลซีเราควรทราบถึงวิธีการกำหนดคุณลักษณะของพีแอลซีและหน่วยอินพุตกับเอาต์พุตของผู้ผลิต ทั้งคุณลักษณะทางไฟฟ้าและการติดตั้งและรายละเอียดต่อไปนี้ก่อนการใช้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

48319

2.11.1 คุณลักษณะทางไฟฟ้า

มีลักษณะที่สำคัญดังต่อไปนี้คือ

1) แรงดันอินพุต (input voltage rating)

ระดับสัญญาณที่หน่วยอินพุตใช้แยกความแตกต่างระหว่างสภาวะ ON และ OFF ของอุปกรณ์ภายนอก สัญญาณอินพุตสามารถเปลี่ยนแปลงได้ประมาณ 10 ถึง 15 เปอร์เซ็นต์

2) กระแสอินพุต (input current rating)

จำนวนกระแสที่หน่วยอินพุตต้องการจากอุปกรณ์อินพุตเมื่อมีสภาวะ ON และ OFF

3) ระดับเทลโฮลด์ (input threshold voltage)

ช่วงการเปลี่ยนระดับแรงดันอินพุตที่หน่วยอินพุต สามารถแยกความแตกต่างระหว่างสภาวะ ON กับสภาวะ OFF

4) ช่วงเวลาอินพุต (input delay)

ช่วงเวลาที่หน่วยอินพุตรับสภาวะของอุปกรณ์อินพุต

5) แรงดันเอาต์พุต (output voltage rating)

ระดับสัญญาณไฟฟ้าของหน่วยเอาต์พุตเมื่อมีสภาวะ ON และ OFF แรงดันเอาต์พุตนี้อาจเปลี่ยนแปลงประมาณ 10 เปอร์เซ็นต์

6) กระแสเอาต์พุต (output current rating)

จำนวนกระแสไฟฟ้าสูงสุดที่หน่วยเอาต์พุตสามารถจ่ายให้กับอุปกรณ์ภายนอกหรืออนุญาตให้กระแสไฟฟ้าไหลผ่านเมื่อมีสภาวะ ON

7) กำลังเอาต์พุต (output power rating)

ขนาดกำลังไฟฟ้าสูงสุดที่หน่วยเอาต์พุตจ่ายให้อุปกรณ์ภายนอก

8) อัตตราการใช้กระแส (current requirement)

จำนวนกระแสไฟฟ้าที่หน่วยอินพุตและเอาต์พุตต้องการจากหน่วยจ่ายกำลังงาน

นอกจากนี้ยังต้องคำนึงถึงอัตราการทนกระแสสูงสุด กระแสรั่วซึม ช่วงเวลาเอาต์พุต ON ช่วงเวลาเอาต์พุต OFF ความละเอียด ความเป็นฉนวนไฟฟ้า ซึ่งในโครงงานนี้จะไม่คำนึงถึงค่าเหล่านี้มากนัก

2.11.2 คุณสมบัติของสัญญาณอินพุต

ทำให้สัญญาณเข้า ได้ระดับที่เหมาะสมกับ PLC ส่วนในการส่งสัญญาณระหว่างอินพุตกับ CPU จะติดต่อกันด้วยลำแสง ซึ่งอาศัยอุปกรณ์ประเภทโฟโตทรานซิสเตอร์ เพื่อต้องการแยกสัญญาณ (Isolate) ทางไฟฟ้าให้ออกจากกัน เป็นการป้องกันไม่ให้ CPU เสียหายเมื่ออินพุตเกิดลัดวงจร และส่วนหน้าสัมผัสจะต้องไม่สั่นสะเทือน (Contact Chattering)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.11.3 คุณลักษณะเอาต์พุต

ในส่วนของเอาต์พุตจะทำหน้าที่รับค่าสภาวะที่ได้จากการประมวลผลของ CPU แล้วนำค่าเหล่านี้ไปควบคุมอุปกรณ์ทำงาน เช่น รีเลย์ โซลีนอยด์ หรือหลอดไฟ เป็นต้น นอกจากนั้นแล้วยังทำหน้าที่แยกสัญญาณของหน่วยประมวลผลกลาง (CPU) ออกจากอุปกรณ์เอาต์พุต โดยปกติเอาต์พุตนี้จะมีความสามารถขับโหลดด้วยกระแสไฟฟ้าประมาณ 1-2 แอมแปร์ แต่ถ้าโหลดต้องการกระแสไฟฟ้ามากกว่านี้ จะต้องต่อเข้ากับอุปกรณ์ขับอื่นเพื่อขยายให้รับกระแสไฟฟ้ามากขึ้น เช่น รีเลย์ หรือคอนแทคเตอร์ เป็นต้น

2.12 หน่วยป้อนโปรแกรมให้พีแอลซี

หน่วยป้อนโปรแกรม ทำหน้าที่ป้อนโปรแกรมลงในหน่วยความจำ ซึ่งโปรแกรมบริหารระบบจะนำไปประมวลผลเพื่อควบคุมอุปกรณ์ภายนอกอีกทีหนึ่ง ผู้ผลิตพีแอลซีพยายามให้ความสะดวกแก่ผู้ใช้มากขึ้น โดยการหาหน่วยป้อนโปรแกรมที่ใช้ง่ายและมีความสามารถพิเศษต่างๆเพิ่มขึ้น เช่น การตรวจสอบแก้ไขโปรแกรมในหน่วยความจำขณะที่พีแอลซีกำลังปฏิบัติงาน การติดต่อกับระบบโครงข่ายและอุปกรณ์ร่วมอื่นๆ หน่วยป้อนโปรแกรมของพีแอลซีมี 4 ประเภท คือ

1) เครื่องป้อนโปรแกรมแบบจอภาพ (CRT Programmer)
2) เครื่องป้อนโปรแกรมขนาดเล็ก (Mini-Programmer)
3) เครื่องบันทึกโปรแกรม (Program Loader)
4) เครื่องบันทึกข้อมูลลงหน่วยความจำ (Memory Burner)
5) คอมพิวเตอร์

2.13 ข้อเปรียบเทียบระหว่าง Z-80 และ 8051

Z-80 เป็นไมโครโปรเซสเซอร์ขนาด 8 บิต ซึ่งผลิตโดยบริษัทไซลอก ส่วน 8051 เป็นซิงเกิลชิพไมโครคอนโทรลเลอร์ ที่ผลิตโดยบริษัทอินเทลมีข้อแตกต่างดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2.2 ข้อเปรียบเทียบระหว่าง Z-80 และ 8051

โครงสร้าง	Z80	8051
Pin Configurations		
Total pins	40	40
Address pins	16 (fixed)	16
Data pins	8 (fixed)	8
Interrupt pins	2 (fixed)	2
I/O pins	0	32
Architecture		
8-bit Registers	20	34
16-bit Registers	4	2
Stack size	64K	128 bytes
Internal ROM	0	4K bytes
Internal RAM	0	128K bytes
External Memory	64K	128K bytes
Flags	6	4
Timers	0	2
Parallel port	0	4
Serial port	0	1
Instruction Sets		
(types/Variations)	4/14	2/6
External moves	2/4	0
Block moves	4/4	12/12
Bit mainpulate	0	3/3
Jump on bit	3/15	2/2
Stack	203	49
Single byte	490	62
Multi-byte		

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.13.1 การพัฒนาไมโครคอนโทรลเลอร์

1) 4 บิตไมโครคอนโทรลเลอร์

มีผลิตจำหน่ายอยู่หลายบริษัทด้วยกันดังนี้คือ

ตารางที่ 2.3 ขนาด 4 บิตไมโครคอนโทรลเลอร์

Manufacturer : Model	Pins : I/O	Counters	RAM (bytes)	ROM (bytes)	Other Features
Hitachi : HMMS40	28:10	-	32	512	10-bit ROM
National : COP420	28:23	1	64	1K	Serial bit I/O
OKI : MSM6411	16:11	-	32	1K	
TI : TMS 1000	28:23	-	64	1K	LED display
Toshiba : TLCS47	42:35	2	128	2K	Serial bit I/O

2) 8 บิต ไมโครคอนโทรลเลอร์

มีผลิตด้วยกันหลายบริษัทซึ่งมีรายละเอียดดังนี้ คือ

ตารางที่ 2.4 ขนาด 8 บิตไมโครคอนโทรลเลอร์

Manufacturer : Model	Pins : I/O	Counters	RAM (bytes)	ROM (bytes)	Other Features
Intel : 8048	40:27	1	64	1K	External Memory to 8K
Intel : 8051	40:32	2	128	4K	External Memory to 128K;serial port
National : COP820	28:24	1	64	1K	Serial bit I/O
Motorola : 6805	28:20		64	1K	
Motorola : 68HC11	52:40	2	256	8K	Serial Ports;

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2.4 (ต่อ) ขนาด 8 บิตไมโครคอนโทรลเลอร์

Manufacturer : Model	Pins : I/O	Counters	RAM (bytes)	ROM (bytes)	Other Features
					A/D;watch dog timer (WDT)
Rockwell : 6500/1	40:32	1	64	2K	
Signetics : 87C552	68:48	3	256	8K	Serial port; A/D;WDT
TI : TMS7500	40:32	1	128	2K	Extermal memory to 64K
TI : TMS370C050	68:55	2	256	4K	External memory to 112K; A/D;serial port;WDT
Zilog : Z8	40:32	2	128	2K	External memory to 124K; Serial port
Zilog : Z8820	44:40	2	270	8K	External memory to 128K; serial port

3) 16 บิต ไมโครคอนโทรลเลอร์

บริษัทที่ผลิตจำหน่ายในตลาดมีหลายบริษัทและมีรายละเอียดดังนี้ คือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

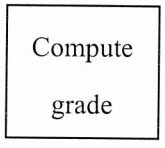
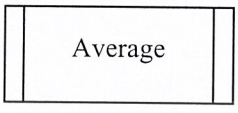
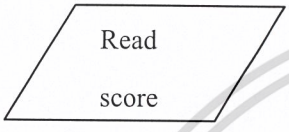
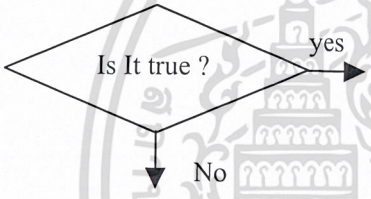
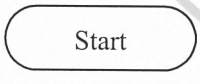
ตารางที่ 2.5 ขนาด 16 บิต ไมโครคอนโทรลเลอร์

Manufacturer : Model	Pins : I/O Counters	RAM (bytes)	ROM (bytes)	Other Features
Hitachi : H8/532	84:65 5	1K	32K	External memory to 1 mega byte;serial port;A/D;pulse width modulation
Intel ; 8096	68:40 2	232	8K	External memory to64K;Serial port;A/D;WDT; Pulse Width modulation
National : HPC16164	64:52 4	512	16K	External memory to64K;Serial port;A/D;WDT; PulseWidth Modulation

2.14 ผังการทำงาน

ถูกนำมาใช้ในการเขียนการทำงานของโปรแกรม โดยใช้บล็อคสัญญลักษณ์เป็นตัวกำหนดความหมายซึ่งมีความจำเป็นในการออกแบบโปรแกรมดังแสดงในรูปที่ 2.11

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Symbol	Name	Function
	Process	Used to indicate any type of Arithmetic or procedural operation
	Predefined process	Used to invoke a subroutine Wich is normally used as a Predefined process in assembly language
	Input/output	Used for any input or output operation
	Decision	Used to ask a question in a Program with either two or Three outcomes
	Connector	Allows the flowchart to be Program with euther two or Three outcomes
	Terminal	Indicates the starting or ending Point of the program or Predefined process

รูปที่ 2.11 THE COMMONLY USED FLOWCHARTING SYMBOLS

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

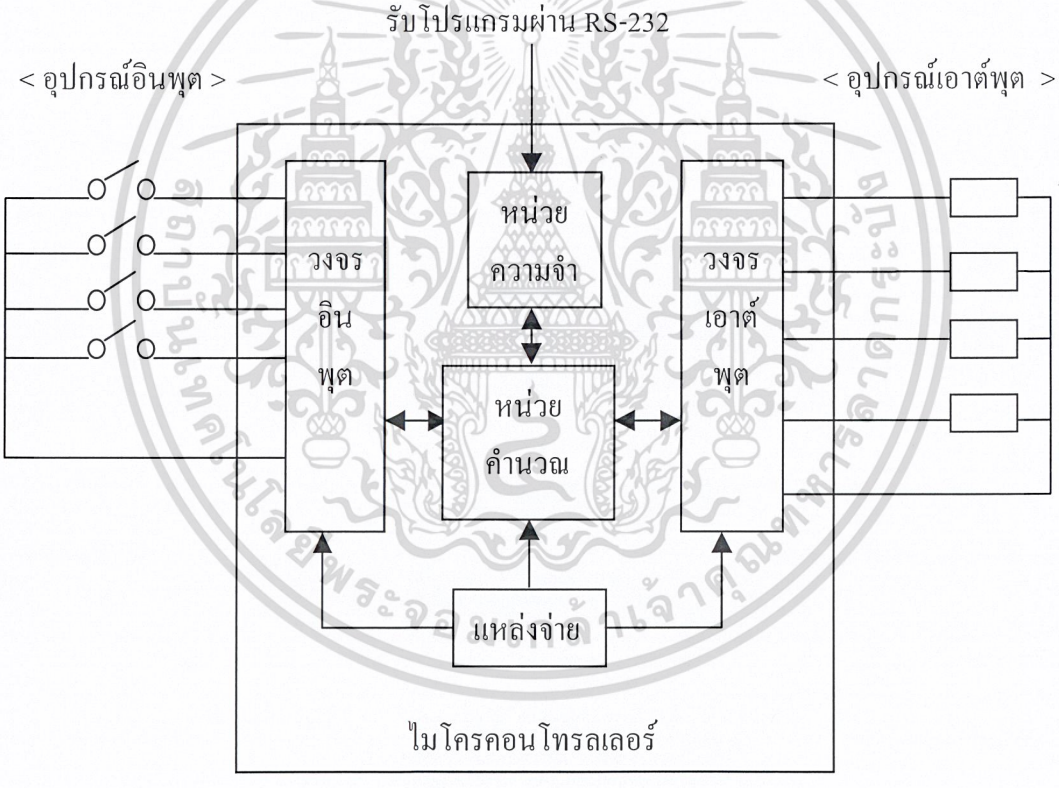
Process Symbol	แสดง Process การทำงานของโปรแกรมหรืออาจจะเป็นคณิตศาสตร์ แต่ไม่ควรเขียนเป็นภาษาเอสเซมบลีในบล็อคนี้
Predefined proess	บล็อคนี้เป็นบล็อคที่สำคัญมากในการเขียนโปรแกรม เพราะมันเป็นผลการทำงานของ Subroutine ที่จำเป็นสำหรับ Main Program
Input/Output Symbol	บล็อคนี้ใช้สำหรับอ่านอินพุตเข้ามา หรือการส่งข้อมูลออกไป เช่น การแสดงผล
Decision Synbol	ใช้สำหรับตัดสินใจ โดยบล็อคนี้จะมีอินพุตเข้า 1 ทางและออก 2 ทาง โดยมีความหมายถึง Yes หรือ No หรือไม่ก็ใช้คำว่า 0 หรือ 1 บางครั้งอาจจะใช้คำว่า Ture หรือ Fault ก็ได้
The Connector	ใช้เชื่อมโยงโฟลว์ชาร์ตย่อย ๆ เข้าด้วยกันในบล็อคนี้อาจจะเป็นอักษร (Character) หรือเลขใด ๆ (Number) ก็ได้
Terminal Symbol	บล็อคนี้ใช้แสดงจุดเริ่มต้นของโปรแกรม (START) จุดสิ้นสุดของโปรแกรม (END) และการกระโดดกลับ (Return)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3
การออกแบบ การสร้าง และการทำงาน

3.1 ส่วนของฮาร์ดแวร์ (HARDWARE)

ภายในพีแอลซีที่สร้างขึ้นมา จะมีโครงสร้างดังต่อไปนี้ ซึ่งฮาร์ดแวร์ของเครื่อง PLC ได้ใช้ไมโครคอนโทรลเลอร์ เป็นหัวใจหลักของโครงงานนี้ โดยภายในไมโครคอนโทรลเลอร์จะประกอบไปด้วย วงจรอินพุต วงจรเอาต์พุต หน่วยความจำ หน่วยคำนวณ และมีแหล่งจ่ายไฟไว้คอยจ่ายกระแสไฟให้กับทุกส่วน



รูปที่ 3.1 โครงสร้างพีแอลซีที่สร้างขึ้น

3.1.1 การถอดรหัสตำแหน่งของหน่วยความจำ

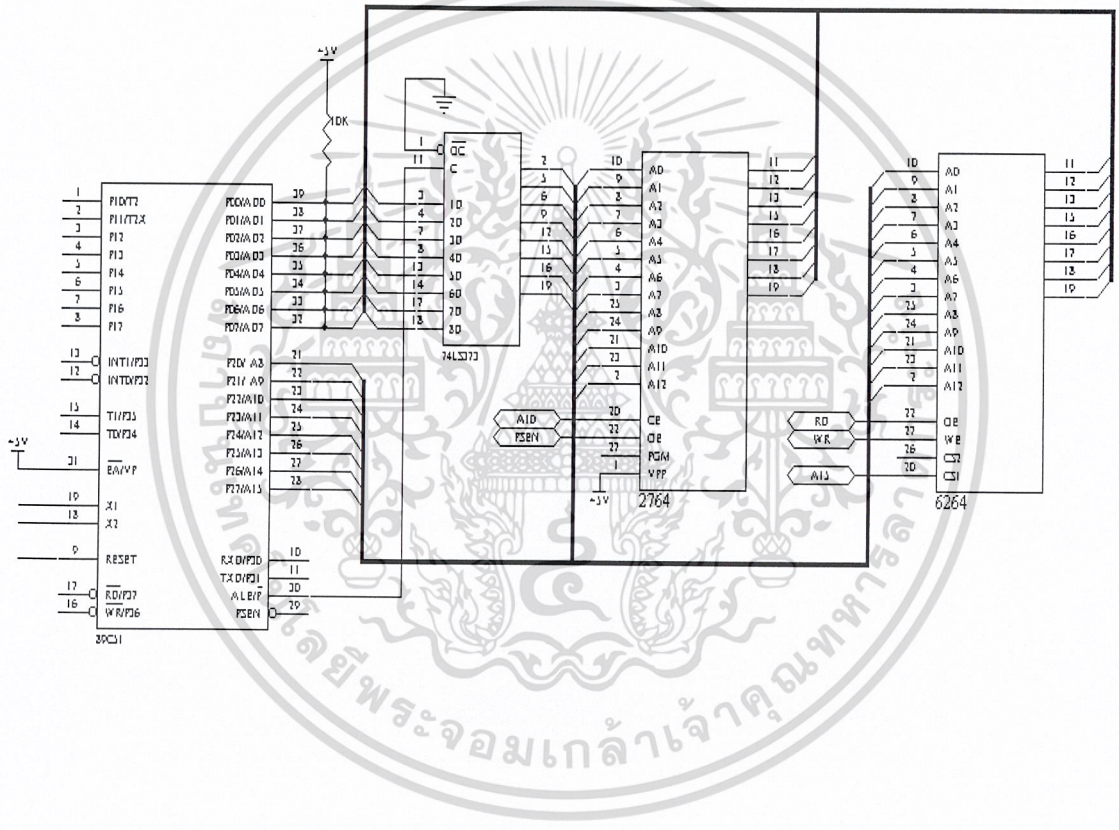
การถอดรหัสตำแหน่งของหน่วยความจำมีความจำเป็นอย่างยิ่งเพราะว่าหน่วยความจำที่นำมาใช้มีขนาดเล็กกว่า 64 Kbyte ดังนั้นจึงต้องนำมาเรียงกันแล้วใช้วงจรถอดรหัสตำแหน่งของหน่วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ความจำแยกเป็นแต่ละช่วง ถ้าใช้หน่วยความจำขนาด 4 Kbyte จำนวน 8 ตัว จะได้เท่ากับ 32 Kbyte ถ้าให้ตัวแรกเพิ่มที่แอดเดรส 8000H ตัวที่ 2 ก็จะเริ่มที่แอดเดรส 9000H

3.1.2 การต่อ MCS-51 เข้ากับหน่วยความจำแบบ EPROM และ RAM

การต่อใช้งานนี้เราอาจจะไม่ใช้ Decoder มาถอดรหัสตำแหน่งของหน่วยความจำได้ ถ้าเราติดตั้งเพียงอย่างละ 1 ตัวเท่านั้น เพราะแอดเดรสของหน่วยความจำและข้อมูล ใช้แอดเดรสซ้อนกันอยู่ สัญญาณที่ใช้อ่านข้อมูลจากหน่วยความจำอีพรอม ก็คือ PSEN และสัญญาณที่ใช้อ่านข้อมูลจากหน่วยความจำแบบแรมก็คือ RD การเขียนข้อมูลลงแรม จะใช้ WR รายละเอียดดังรูปที่ 3.2

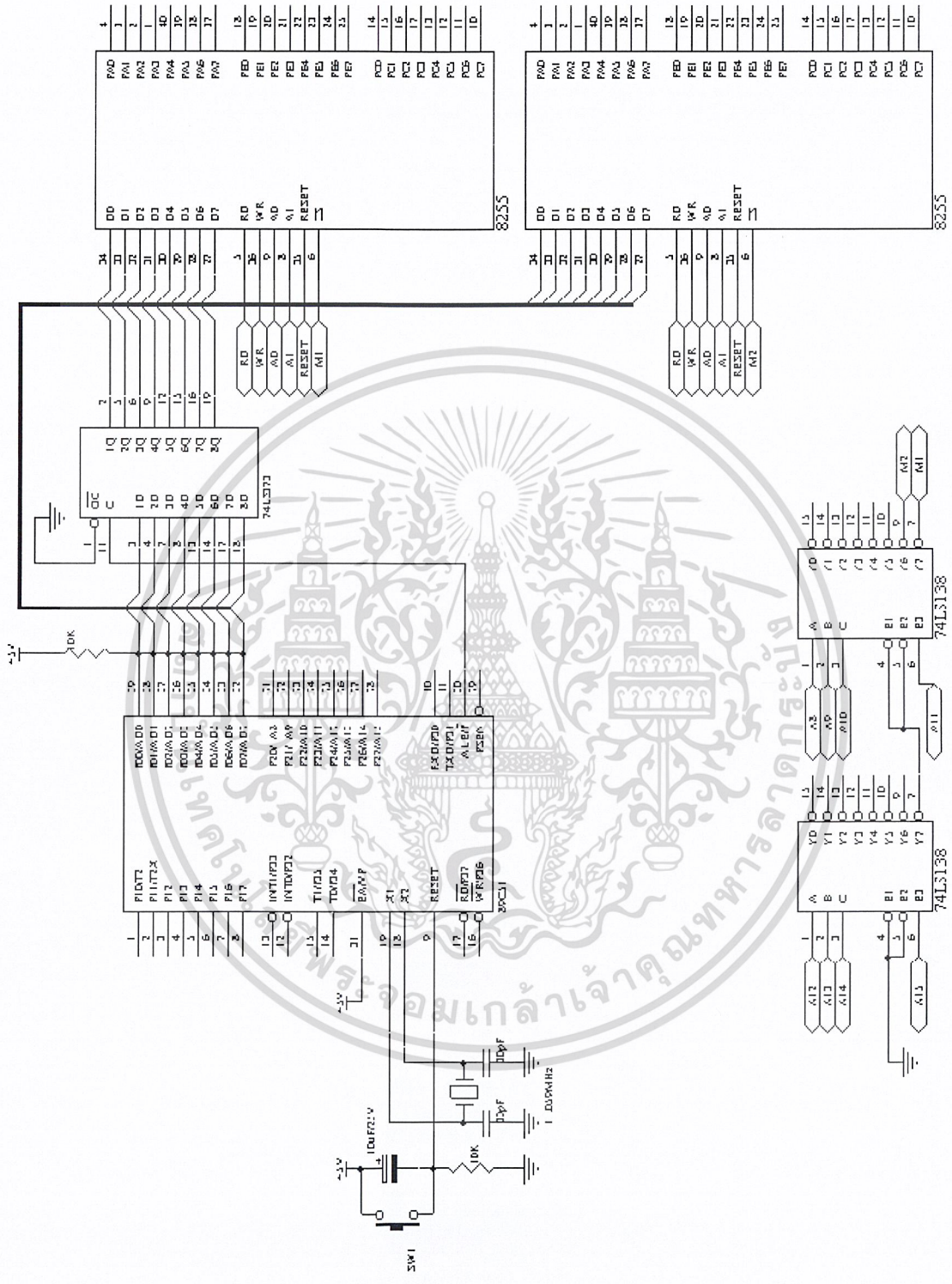


รูปที่ 3.2 การต่อ MCS-51 เข้ากับ EPROM และ RAM

3.1.3 การเชื่อมต่อ 8255 กับ MCS-51

เป็นชิพขนาด 40 ขามีอยู่ 3 พอร์ท คือ A, B, C เป็นพอร์ท 8 บิตที่สามารถโปรแกรมให้เป็นอินพุต หรือเอาต์พุตก็ได้ โดยที่พอร์ท C ยังแบ่งเป็น 4 บิตบนและ 4 บิตล่าง

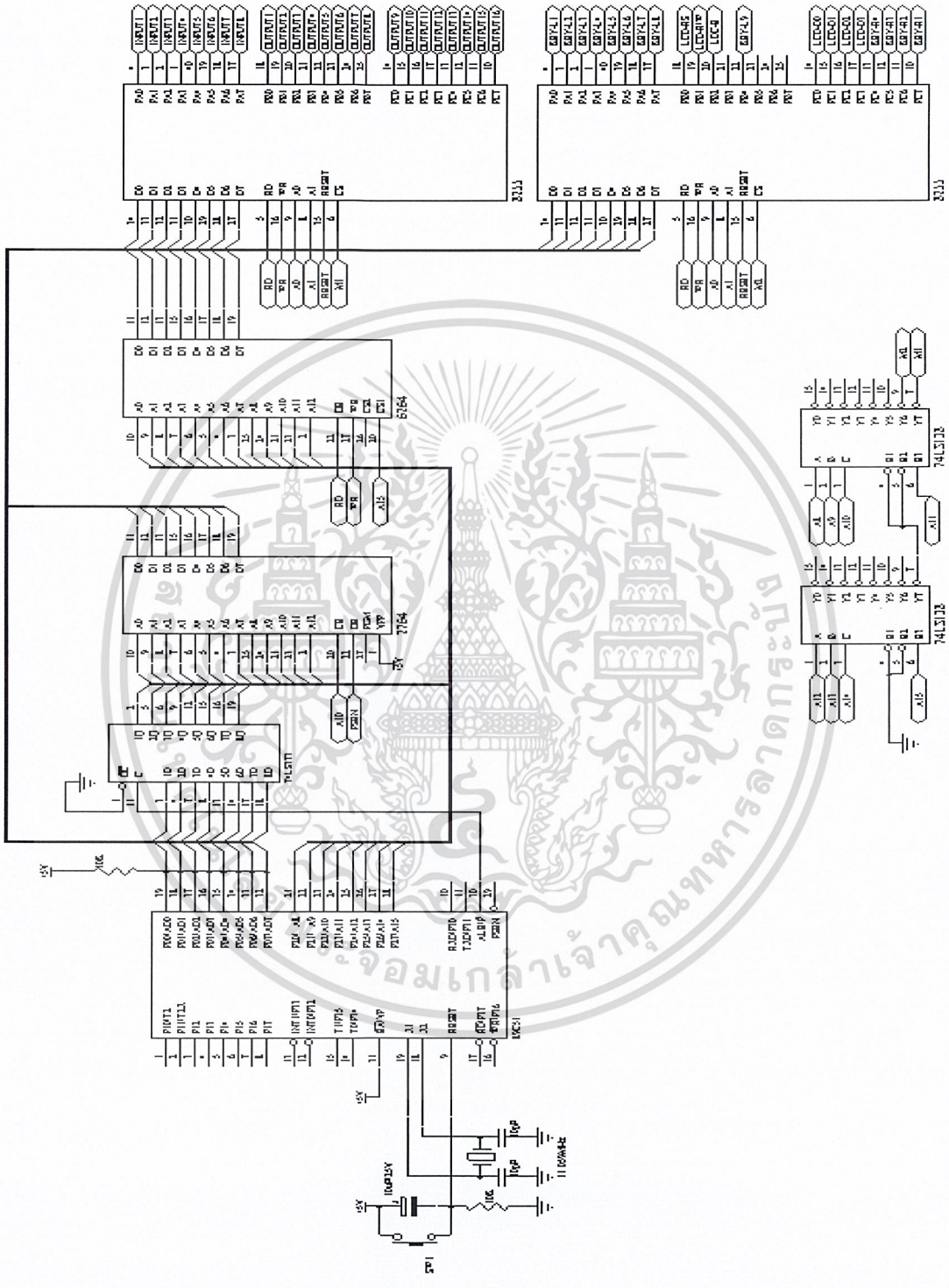
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.3 การเชื่อมต่อ 8255 กับ MCS-51

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1.4 CONTROLLER วงจรรวม



รูปที่ 3.4 บอร์ดหลัก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บอร์ดหลักในรูปที่3.4เป็นส่วนของไมโครคอนโทรลเลอร์ซึ่งบอร์ดนี้ได้ใช้ไมโครคอนโทรลเลอร์ ตระกูล MCS-51 เบอร์ 89C51RD2

1) แนวทางการพัฒนา

1.1) ใช้เครื่องโปรแกรม CPU แล้ว SET Jumper EA ไปที่ Internal

1.2) ใช้ EPROM Emulator แล้ว SET Jumper EA ไปที่ External

1.3) ใช้ CPU เบอร์ 89C51RD2 ในการพัฒนา โดยมีขั้นตอนการใช้ SW_ISP_LOAD ดังนี้

การเข้าสู่โหมด DOWNLOAD (ใช้โปรแกรม WinISP ของ Phillips)

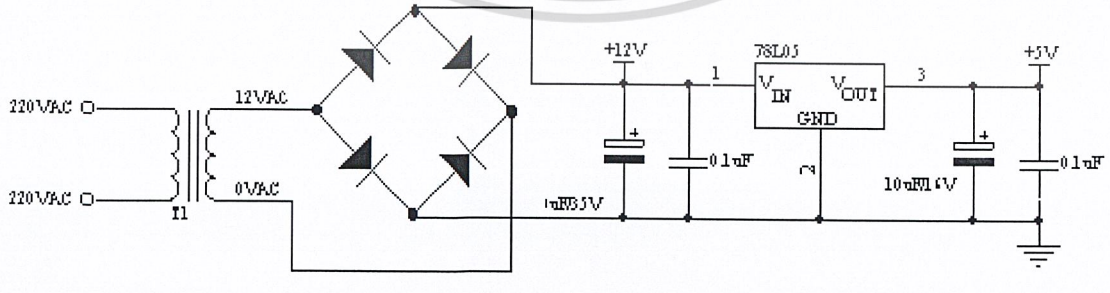
การเข้าสู่โหมด D/L ก็คือการให้ CPU เข้าไปทำงานที่ตำแหน่ง BOOT ROM โดยภายใน CPU จะมี BOOT ROM ขนาด 1KB ซึ่งถูกโปรแกรมมาจากโรงงาน

โดยจะอยู่ที่ตำแหน่ง FC00H-FFFFH ซึ่งจะเป็นโปรแกรมที่ใช้โหลดข้อมูลลงในหน่วยความจำ Flash ขนาด 64K และภายใน CPU จะมี vecot ซึ่งถูกกำหนดมาจากโรงงานให้มีค่า FCH (ห้ามแก้ไขค่า VECTOR)

2) การเข้าสู่โหมดการ RUN

หลังจากที่โหลด โปรแกรมเสร็จเรียบร้อยแล้ว จะต้องกด SWITCH RESET เพื่อให้ CPU ไปทำงานที่ตำแหน่ง 0000H แต่ CPU เบอร์ 89C51RD2 จะใช้ข้อมูลใน STATUS เป็นตัวกำหนดให้ CPU กระโดดไปทำงานที่ตำแหน่งนั้นๆ ดังนั้นถ้าต้องการให้ CPU ไปทำงานที่ตำแหน่ง 0000H หลังจากกด SWITCH RESET จะต้องกำหนดข้อมูลใน STATUS ให้เป็น 00H แต่ถ้าต้องการให้ CPU ไปทำงานที่ตำแหน่งอื่นที่ไม่ใช่ตำแหน่ง 0000H ก็ให้กำหนดให้ STATUS อยู่ที่ตำแหน่งนั้นๆ เช่น ถ้าต้องการให้ CPU ไปทำงานที่ตำแหน่ง 2100H หลังจากกด SWITCH RESET จะต้องกำหนดข้อมูลใน STATUS ให้เป็น 21H

3.1.5 ส่วนของแหล่งจ่ายกำลังงาน



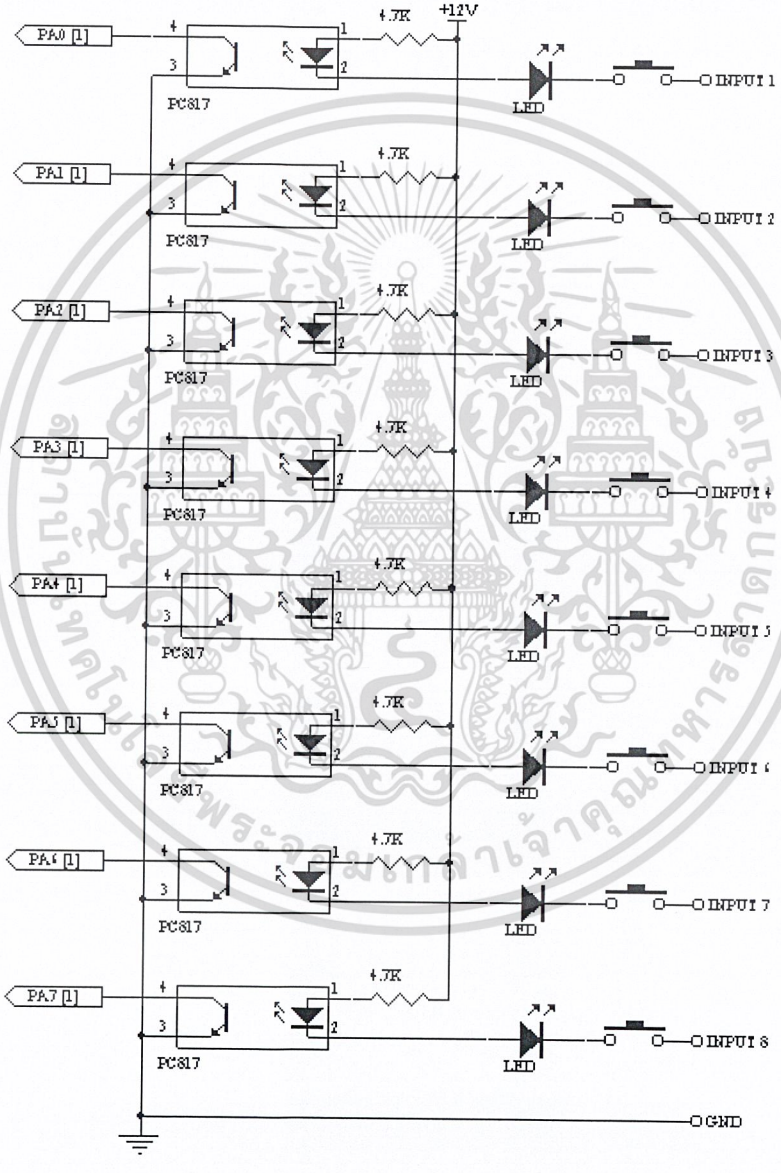
รูปที่ 3.5 แหล่งจ่ายไฟตรง 5 โวลท์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 3.5 เป็นวงจรของแหล่งจ่ายไฟตรง 5 โวลท์ทำหน้าที่ จ่ายกระแสไฟฟ้าให้กับส่วนต่างๆ ของ PLC คือ หน่วยประมวลผลกลาง หน่วยความจำ และ หน่วยอินพุตและเอาต์พุต โดยรักษาแรงดันไฟฟ้าให้คงที่ และแจ้งให้หน่วยประมวลผลทราบเมื่อระบบไฟฟ้ามีอาการผิดปกติ

3.1.6 ส่วนของอินพุต

ส่วนของอินพุตคือวงจรออปโตคอปเปลอร์ 8 วงจร



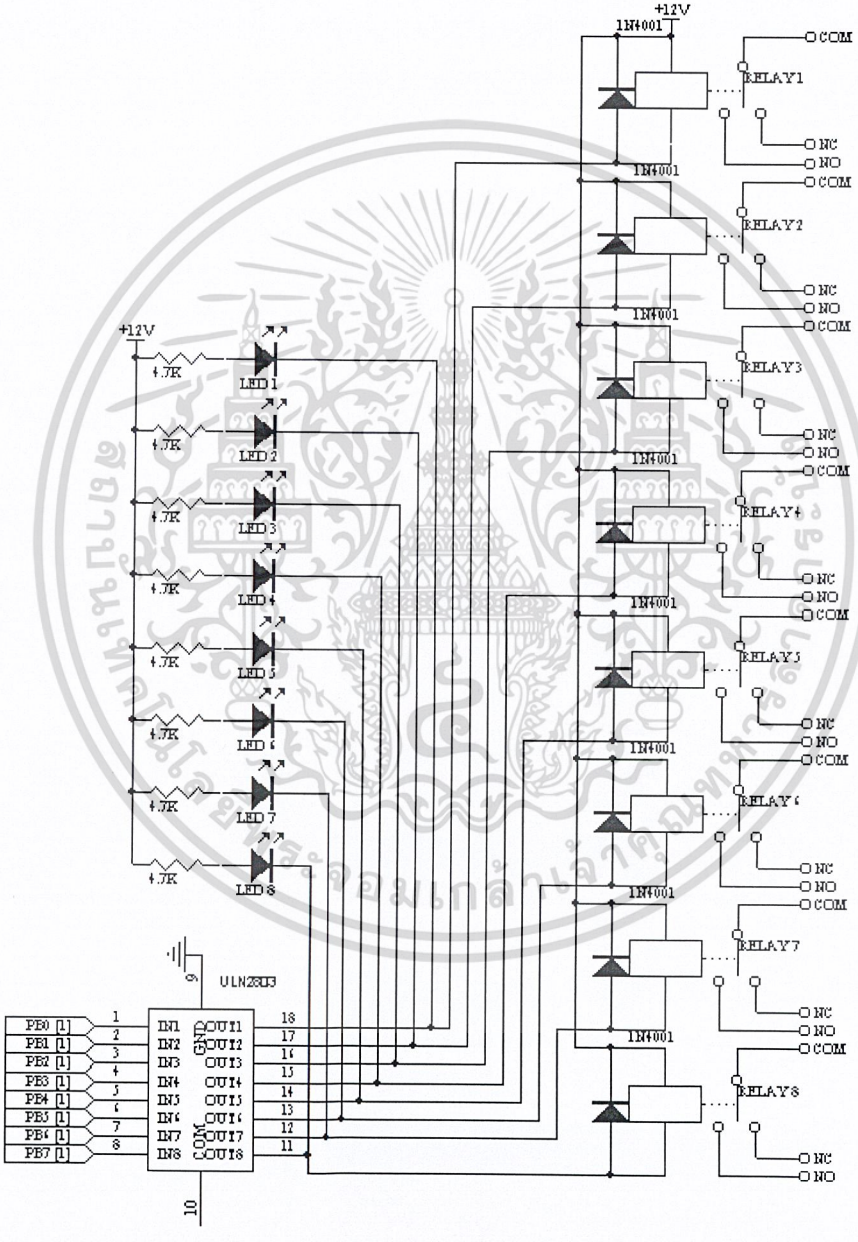
รูปที่ 3.6 วงจรอินพุต 8 วงจร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 3.6 เป็นวงจรอินพุต 8 วงจร ซึ่งจะต่อร่วมกับชุดควบคุมเพื่อรับสภาวะและสัญญาณต่างๆ เช่น หน่วยอินพุตรับสัญญาณหรือสภาวะแล้วส่งไปยัง CPU เพื่อประมวลผล เมื่อ CPU ประมวลผลแล้วจะส่งให้ส่วนเอาต์พุต เพื่อให้อุปกรณ์ทำงานตามที่โปรแกรมไว้

3.1.7 วงจรเอาต์พุต

ส่วนของหน่วยเอาต์พุต คือวงจรรีเลย์ 16 วงจร ในรูปแสดงให้ดู 8 วงจร



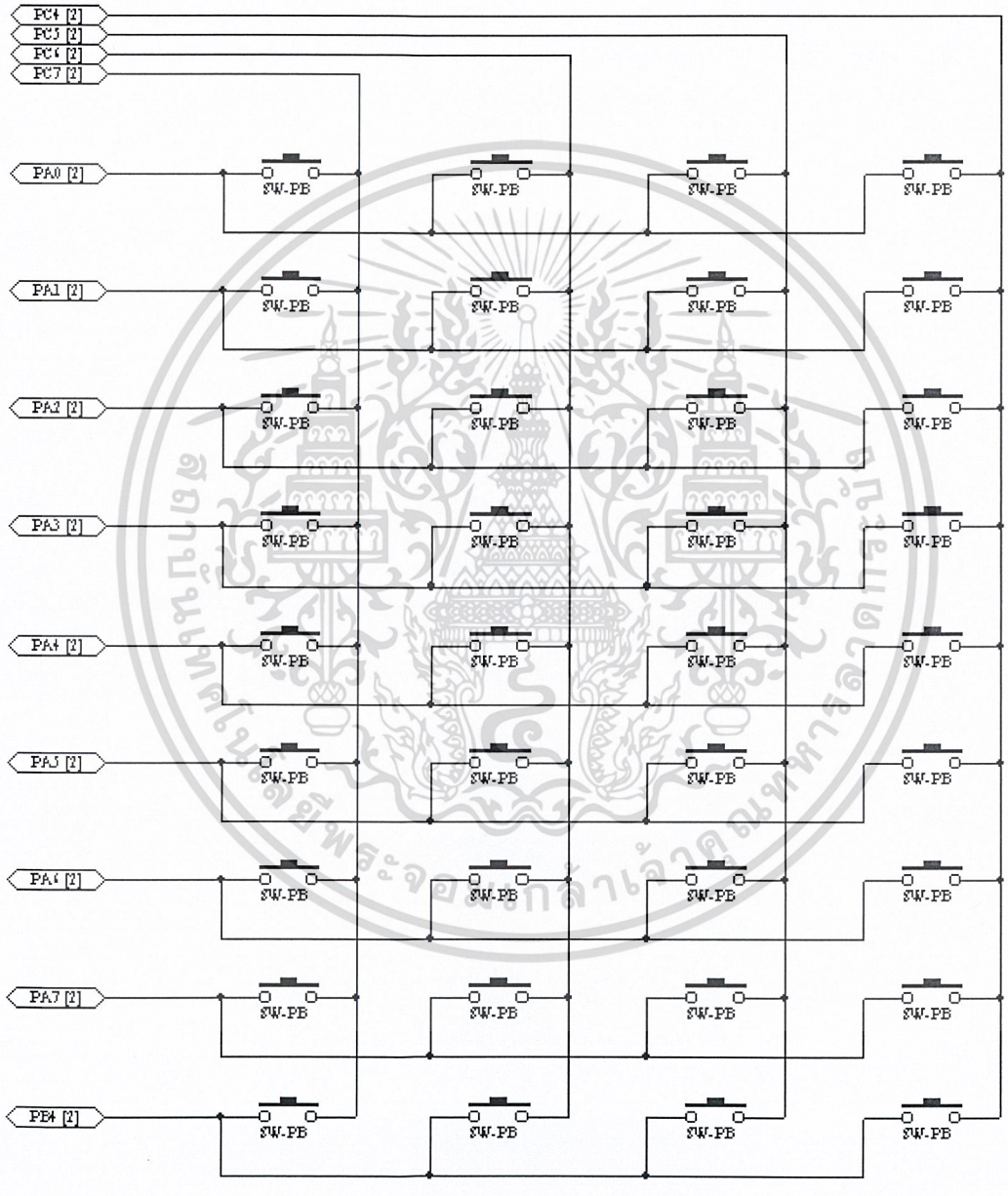
รูปที่ 3.7 วงจรเอาต์พุต 8 วงจร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 3.7 เป็นวงจรเอาต์พุต 8 วงจร จะทำหน้าที่รับค่าสภาวะที่ได้จากการประมวลผลของ PLC แล้วนำค่าเหล่านี้ไปควบคุมอุปกรณ์ทำงาน เช่น รีเลย์ โซลีนอยด์ หรือหลอดไฟ เป็นต้น

3.1.8 วงจร KEY BOARD

ส่วนของ KEY BOARD 36 ปุ่ม

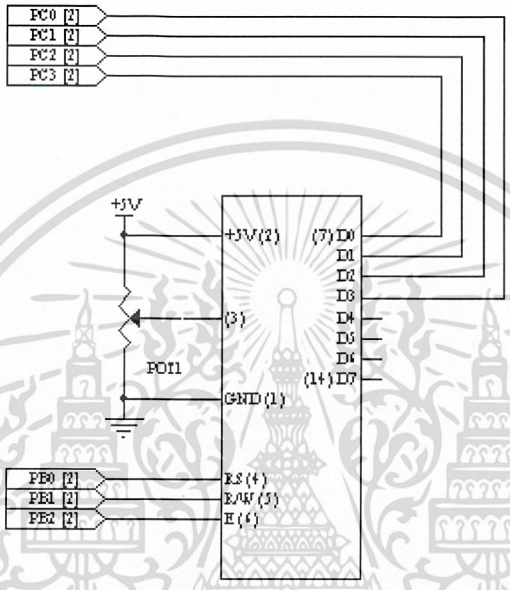


รูปที่ 3.8 วงจร KEY BOARD

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 3.8 เป็นวงจร KEY BOARD ซึ่งใช้เป็นที่ป้อนโปรแกรม จะควบคุมโปรแกรมของผู้ใช้ลงในหน่วยความจำของ PLC นอกจากนั้นแล้วยังทำหน้าที่ติดต่อระหว่างผู้ใช้กับ PLC เพื่อให้ผู้ใช้สามารถตรวจการปฏิบัติงานของ PLC และผลการควบคุมเครื่องจักรและกระบวนการตามโปรแกรมควบคุมที่ผู้ใช้เขียนขึ้นได้อีกด้วย

3.1.9 วงจร DISPLAY LCD



รูปที่ 3.9 วงจร DISPLAY LCD

จากรูปที่ 3.9 เป็นวงจร DISPLAY LCD ทำหน้าที่ใช้แสดงผลอินพุตและเอาต์พุตด้วย LCD

3.2 ส่วนของซอฟต์แวร์ (SOFWARE)

โปรแกรมภาษาแอสเซมบลีที่ป้อนให้กับไมโครคอนโทรลเลอร์เพื่อใช้ในการเลือกว่าขณะนี้จะใช้ในโหมดของการป้อนโปรแกรมหรือโหมดของการรันโปรแกรม

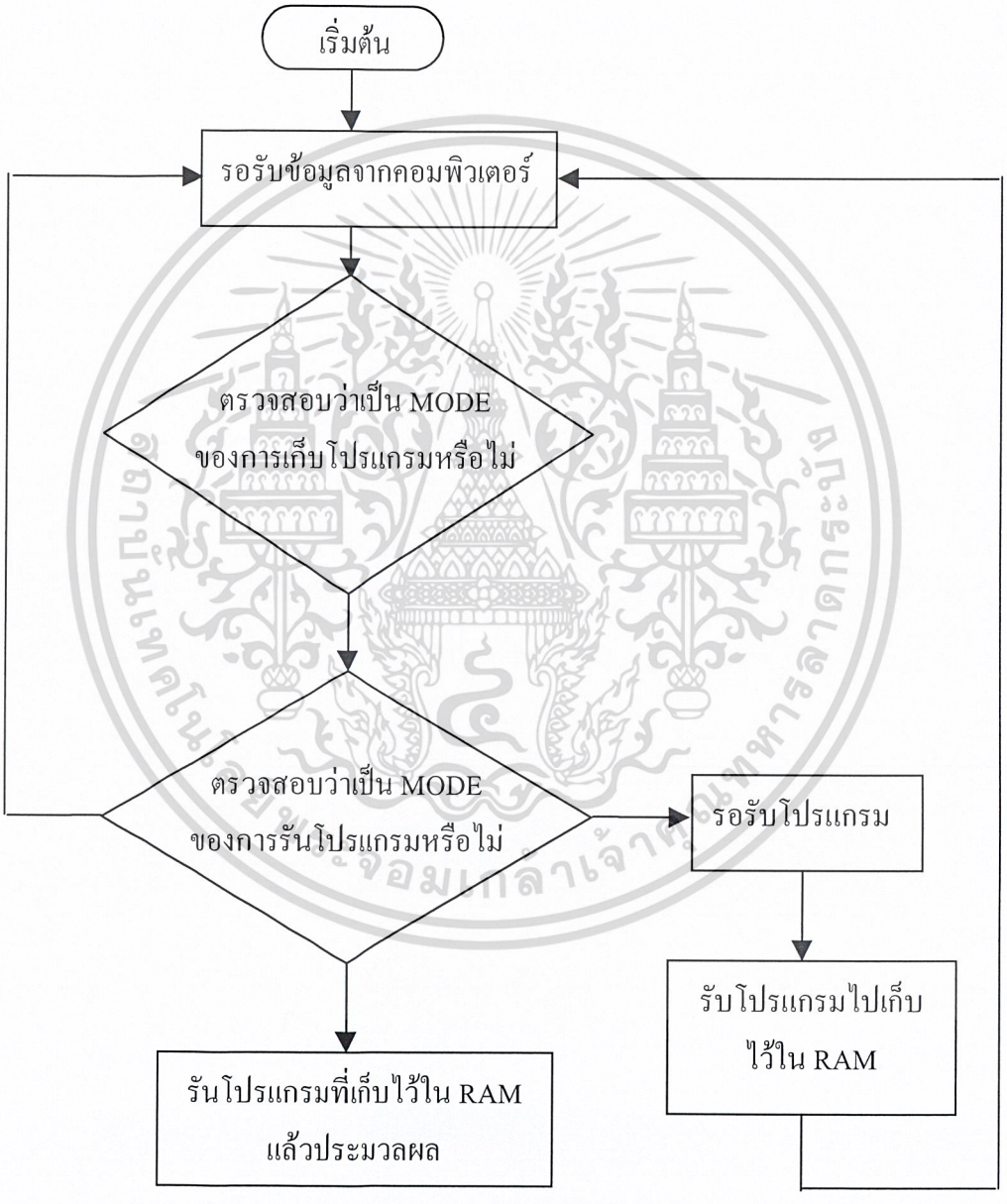
ตารางที่ 3.1 CPU STATUS INDICATORS

CPU STATUS INDICATORS		
RUN	ON	CPU อยู่ในโหมดของการรันโปรแกรม
	OFF	CPU อยู่ในโหมดของการป้อนโปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โหมดของการป้อนโปรแกรมใช้เมื่อต้องการเปลี่ยนแปลงโปรแกรมภาษาแลดเดอร์และในขณะที่ป้อนโปรแกรมหน่วยประมวลผลจะไม่ทำการรันโปรแกรมใดๆ ทั้งสิ้น

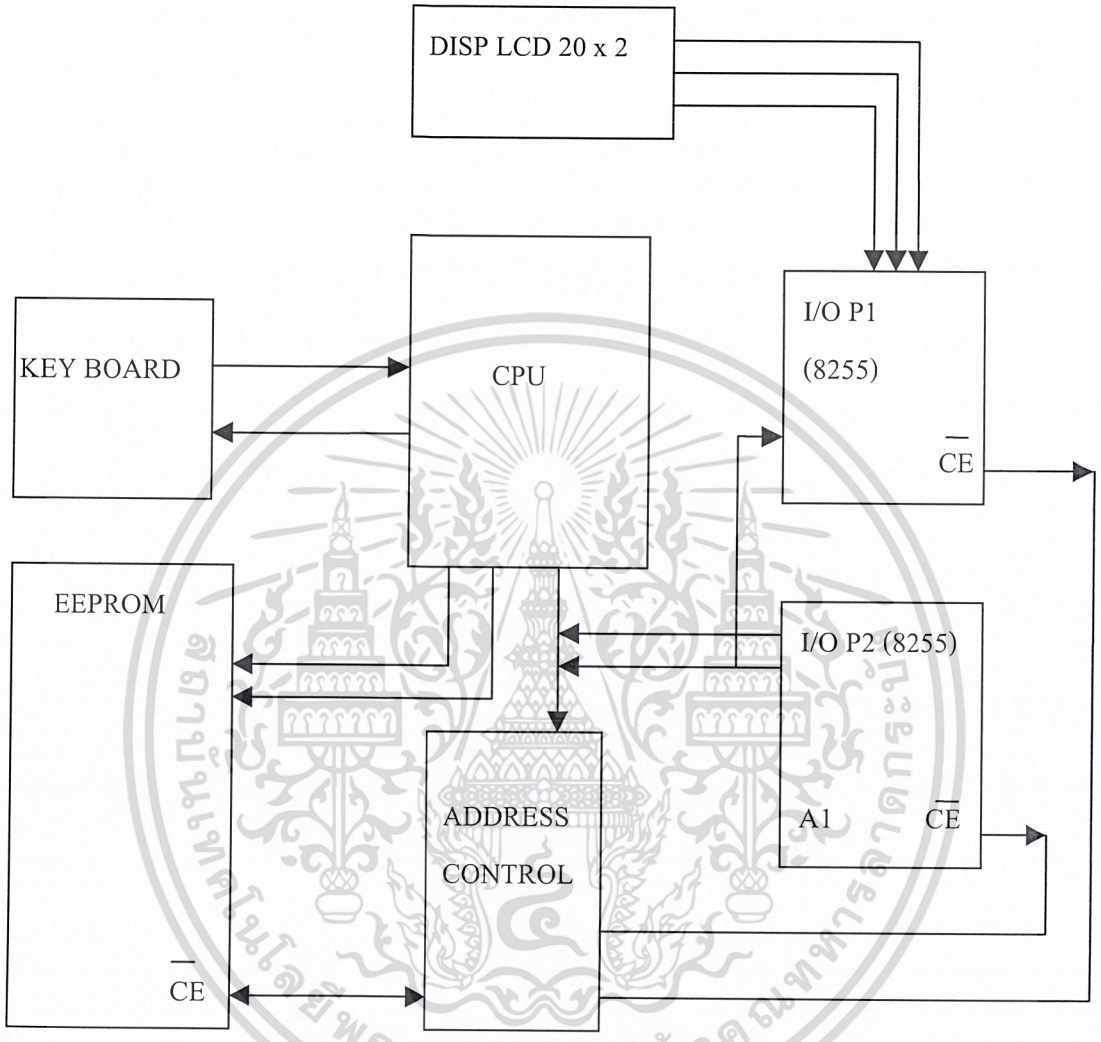
โหมดของการรันโปรแกรมใช้เมื่อต้องการรันโปรแกรมภาษาแลดเดอร์ที่ป้อนเข้ามาล่าสุด และในขณะนั้นหน่วยประมวลผลจะไม่รับโปรแกรมใดๆทั้งสิ้น โดยจะมีขั้นตอนการทำงานดังรูปที่ 3.10



รูปที่ 3.10 ขั้นตอนการทำงานของโปรแกรมภาษาแอสเซมบลีในไมโครคอนโทรลเลอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.1 ลักษณะโครงสร้าง



รูปที่ 3.11 ลักษณะโครงสร้าง

1) ขั้นตอนแรกหลังจาก RESET CPU

1.1) ติดต่อ EEPROM X ตำแหน่ง ADDRESS 0000-008F เรากำหนดให้เป็นตำแหน่งที่ใช้ในการบันทึกข้อมูลและตำแหน่ง PROGRAM ย่อยต่างๆ ไว้เช่น

1.2) จำนวนโปรแกรมที่บันทึกไว้แล้วรวมถึง ADDRESS ของจุดเริ่มต้นของโปรแกรมนั้นและที่สำคัญตำแหน่งปิดท้ายของโปรแกรมเราจะทำการใช้ RET ไว้ใน ADDRESS ท้ายสุด

1.3) จำนวนที่เหลือของหน่วยความจำ

1.4) ค่าคำสั่งควบคุมต่าง ๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หลักการก็ คือ เมื่อ CPU ได้รับ KEY คำสั่งให้ RUN PROGRAM เราจะกำหนดใช้ CPU CALL ไป ณ. ตำแหน่ง ADDRESS RESS ของโปรแกรมนั้นและเมื่อเรียบร้อยแล้วจะเจอคำสั่ง RET ปิดท้าย ดังนั้น CPU จะกระโดดกลับมา ณ. ตำแหน่งเดิมทันที เพื่อรับคำสั่งต่อไป อาจจะเป็นคำสั่งโปรแกรมใหม่ หรือโปรแกรมเดิมก็ได้

รอทำงาน คือ เรากำหนดตำแหน่งของโปรแกรมนั้นไว้ก่อน

เช่น SUB _ AND

SUB _ OR

SUB _ NOT

และกำหนดค่า REGISTER ใช้งานไว้หลายๆ ตัวเช่น

DIN _ AND = DATA INPUT _ AND GATE

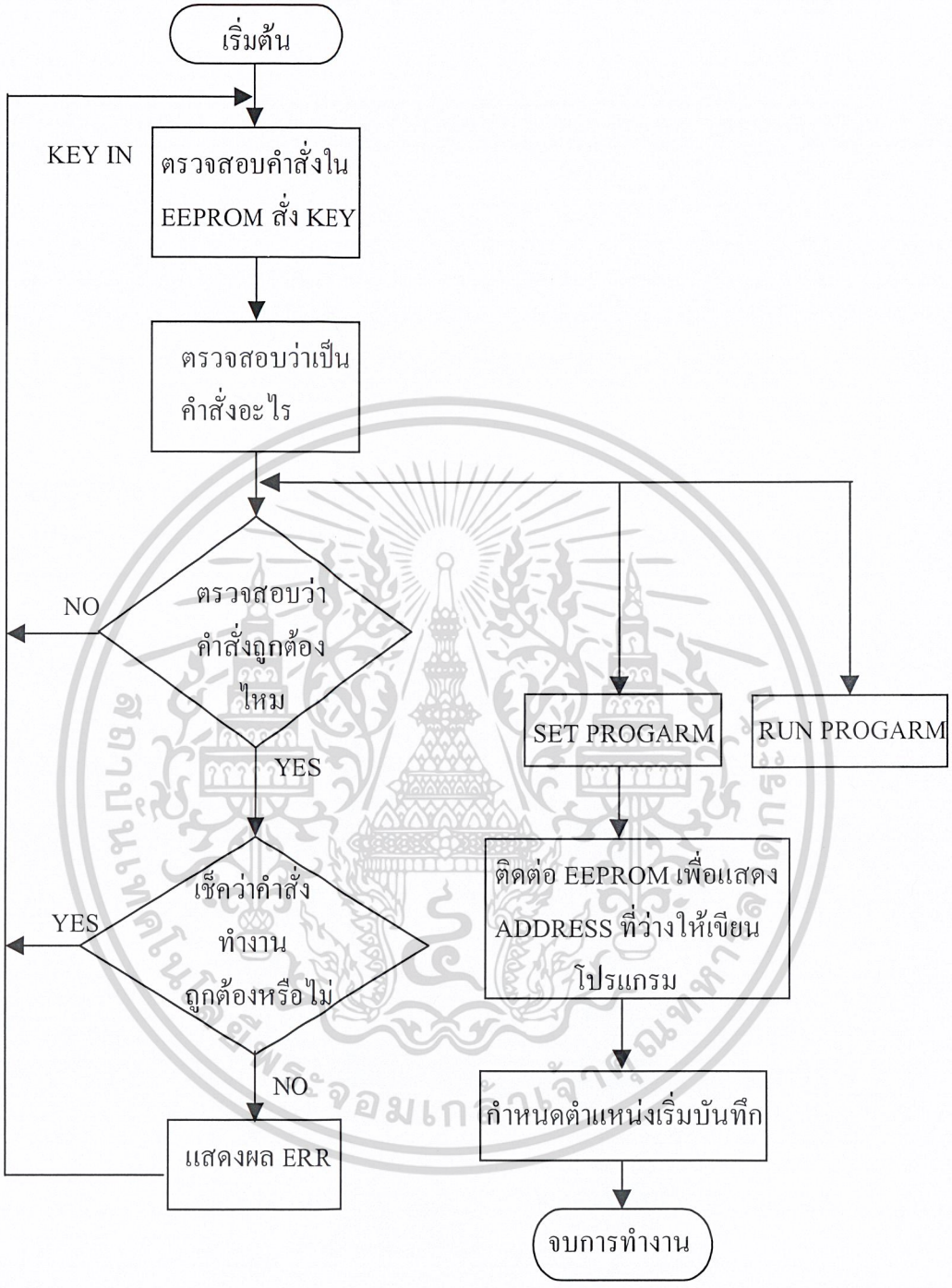
DOUT _ AND = DATA OUTPUT _ AND GATE

โดยกำหนดไว้ที่ตำแหน่ง ADD 20H – 7F เพื่อจะได้แก้ไขข้อมูลเป็นระดับ BIT ได้ การกำหนดแบบนี้เพื่อหลีกเลี่ยงการใช้งาน REG ที่สำคัญ เช่น

REG ACC เพราะเราต้องใช้ในการ +, -

DPR เพราะต้องใช้ในการกำหนด ADDRESS

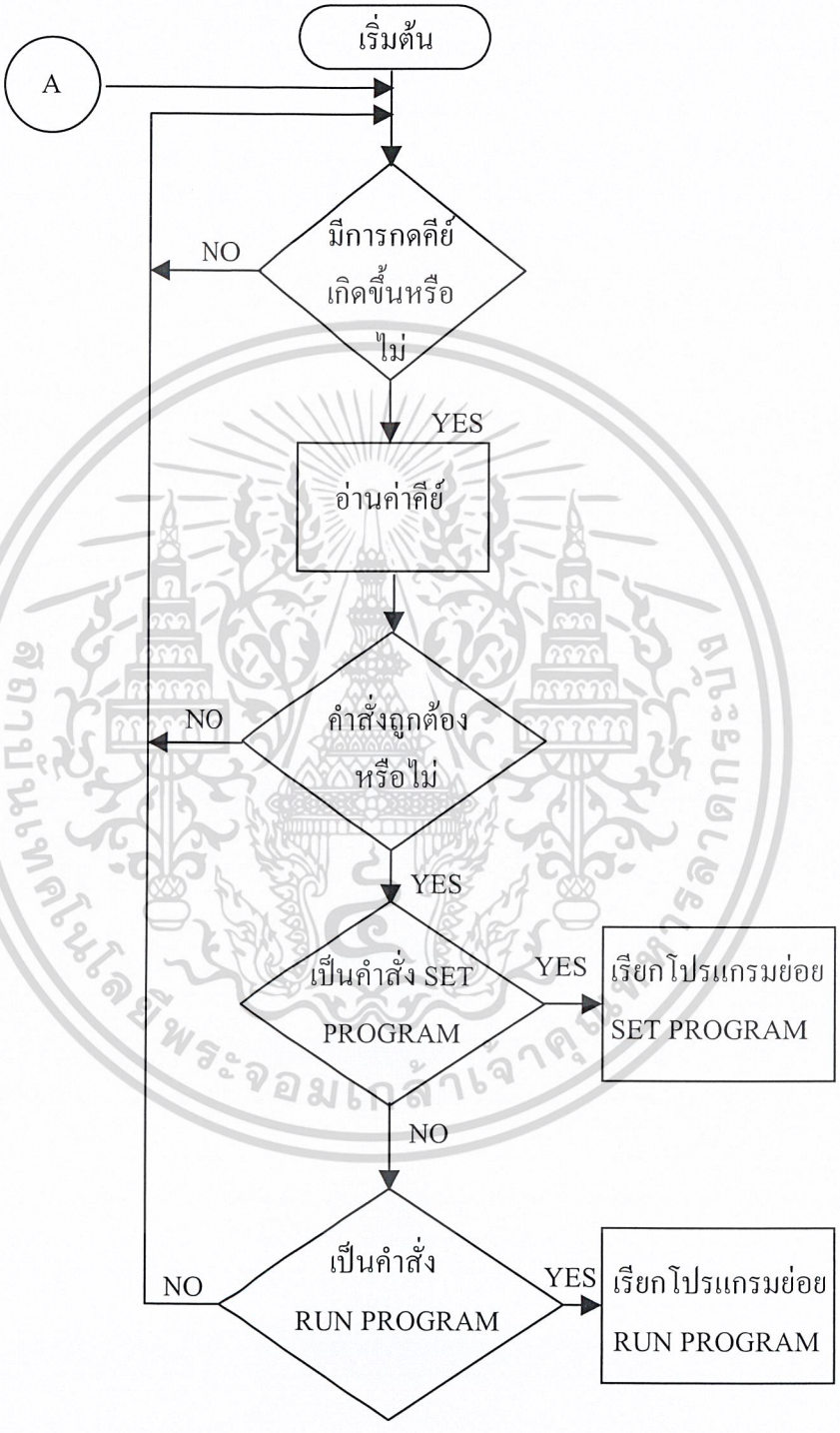
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.12 ขั้นตอนการใช้งานของพีแอลซี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

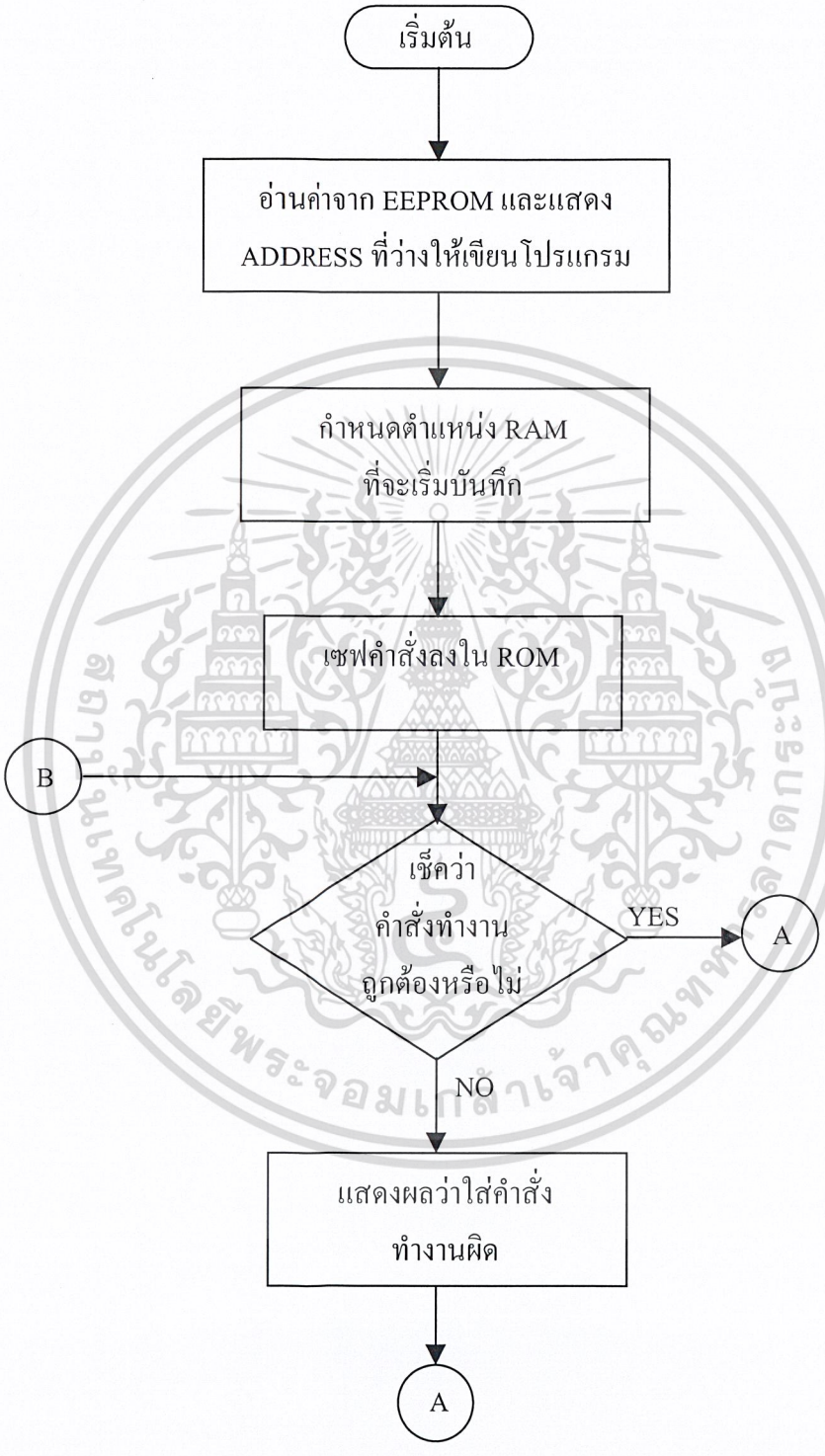
3.2.2 MAIN PROGRAM



รูปที่ 3.13 ผังการทำงาน MAIN PROGRAM

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

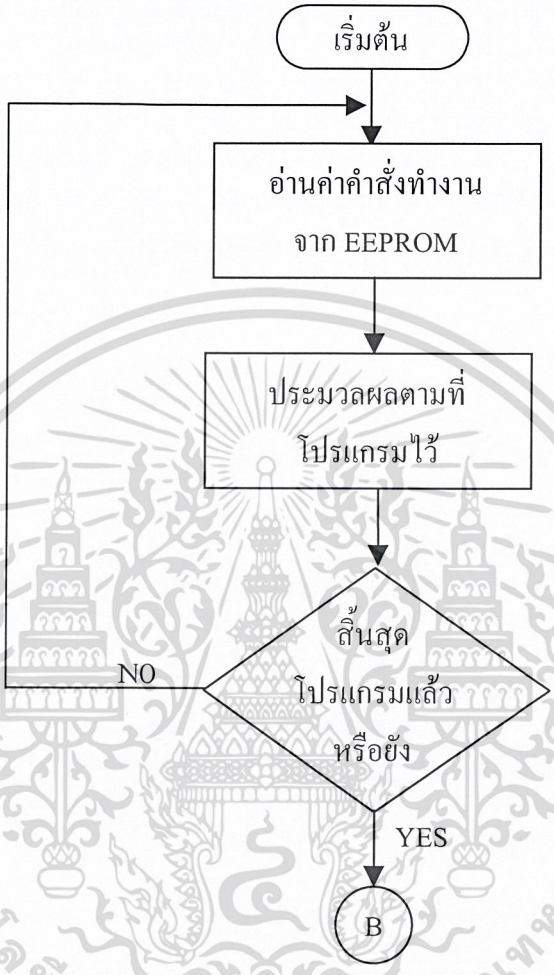
3.2.3 โปรแกรมย่อย SET PROGRAM



รูปที่ 3.14 โปรแกรมย่อย SET PROGRAM

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.4 โปรแกรมย่อย RUN PROGRAM



รูปที่ 3.15 โปรแกรมย่อย RUN PROGRAM

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

การทดลองและผลการทดลอง

4.1 การทดลองการทำงานของโปรแกรมสแกนคีย์และการแสดงผลรวมทั้งทดลองปุ่ม FUNCTION พิเศษต่างๆ

4.1.1 การทดลองการใช้คีย์ที่ใช้ในการเขียนโปรแกรมและการแสดงผล

เมื่อกดปุ่ม MOD จะเข้าสู่การป้อนโปรแกรม LADDER โดยจะแสดงหมายเลขบรรทัดเริ่มต้น ดังรูป



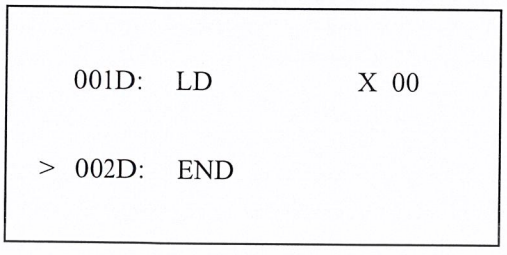
รูปที่ 4.1 หน้าจอ LCD เมื่อกดปุ่ม MOD

ตารางที่ 4.1 ทดลองการรับคีย์บอร์ดที่ใช้ในการเขียนโปรแกรม LADDER

คำสั่ง	อุปกรณ์
LD	X 00
OUT	Y 00
TIM	T 00
CNT	C 00
OR	00
AND	00
SET	00
RES	00
END	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

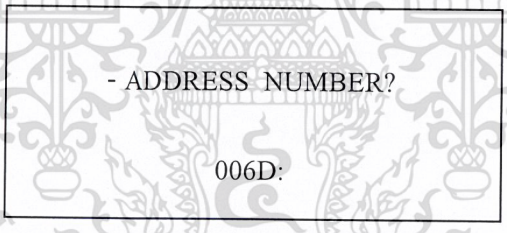
ที่หน้าจอ LCD จะแสดงผลตรงตามรูปแบบโปรแกรมที่ป้อนให้ โดยที่คำสั่ง END จะมีหมายเลขบรรทัด 002D:



รูปที่ 4.2 หน้าจอเมื่อสิ้นสุดการเขียนโปรแกรม

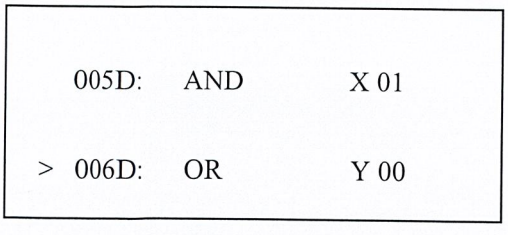
4.1.2 การกดปุ่ม FUNCTION

การกดปุ่ม MOD ต่อจากการทดลองที่แล้ว เมื่อกดปุ่ม MOD 1 MOD จะเป็นการเข้าสู่การป้อนหมายเลขบรรทัดที่ต้องการกระโดดไป โดยป้อนหมายเลขบรรทัดเป็น 006D: เมื่อป้อนแล้วที่หน้าจอจะแสดงผลดังรูป



รูปที่ 4.3 การทดลองปุ่ม MOD

และเมื่อกดปุ่ม  จะทำให้หน้าจอ LCD แสดงผลดังรูป

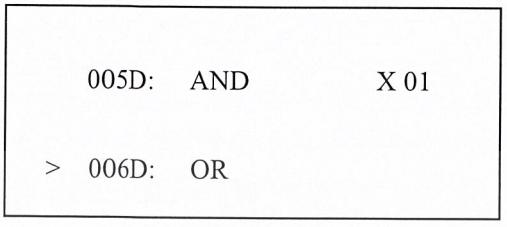


รูปที่ 4.4 หน้าจอเมื่อกดปุ่ม 

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

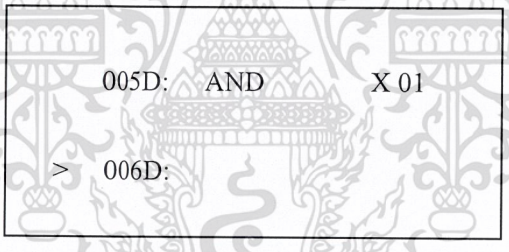
1) การทดลองปุ่ม FUNTION INS, DEL

จากหัวข้อที่แล้วเมื่อกดปุ่ม INS จะทำให้โปรแกรมเลื่อนไป 1 บรรทัด และบรรทัดที่ 006D: จะว่างไว้แสดงผลดังรูป



รูปที่ 4.5 การกดปุ่ม INS

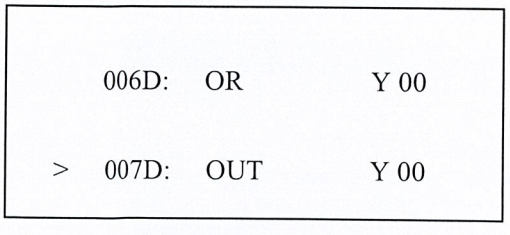
และเมื่อกดปุ่ม DEL ทำให้โปรแกรมที่ถูกเลื่อนไปถอยกลับมา 1 บรรทัดดังเดิม ดังแสดงในรูป



รูปที่ 4.6 การกดปุ่ม DEL

2) การทดลองปุ่ม FUNCTION 

จากหัวข้อที่แล้วเมื่อเรากดปุ่ม  1 ครั้ง จะได้ผลดังรูป



รูปที่ 4.7 การกดปุ่ม 

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

และเมื่อกดปุ่ม ⬆ 1 ครั้ง จะได้ผลดังรูป

```
> 005D:   AND      X 01

  006D:   OR       Y 01
```

รูปที่ 4.8 การกดปุ่ม ⬆

ทดลองการกดปุ่ม ➡ เคอร์เซอร์จะเลื่อนไปทางขวาครั้งละ 1 ตำแหน่ง

ทดลองการกดปุ่ม ⬅ เคอร์เซอร์จะเลื่อนกลับมาทางซ้ายครั้งละ 1 ตำแหน่ง

จากผลที่ได้แสดงว่า ปุ่มคำสั่งและปุ่มตัวเลขที่ใช้ในการเขียนโปรแกรมแลดเดอร์ทำงานได้ถูกต้อง และการแสดงผลตรงตามโปรแกรม การทดลองปุ่ม FUNCTION ทุกปุ่มทำงานตรงตามหน้าที่ได้ถูกต้อง

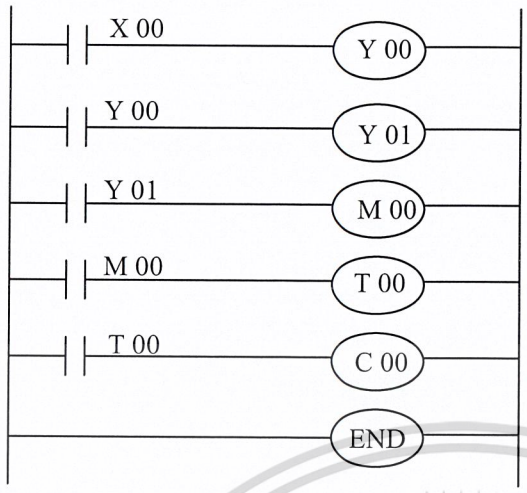
4.2 การทดลองการทำงานของโปรแกรมย่อย

ในหัวข้อนี้จะเป็นการกล่าวถึงการทดลองการทำงานของโปรแกรมย่อยทั้งหมดว่าผลการทำงานเป็นไปตามที่ได้ออกแบบเอาไว้หรือไม่โดยการทดลองเขียนเป็นโปรแกรมแลดเดอร์สั้นๆ เพื่อตรวจดูว่าผลการทำงานถูกต้องหรือไม่ซึ่งสามารถแบ่งออกเป็นการทดลองในส่วนของแต่ละโปรแกรมดังนี้

4.2.1 การทดลองโปรแกรมย่อย LD

โปรแกรมย่อย LD สามารถกระทำการนำค่าสภาวะของอินพุต เอาต์พุต AUX TIMER COUNTER เข้ามาเก็บไว้ในหน่วยความจำ เพื่อนำค่าสภาวะไปใช้ต่อไป การทดลองจึงจะทำการ LD ในทุกส่วนและนำค่าสภาวะส่งออกที่ส่วนเอาต์พุตซึ่งมี LED สภาวะ 1 LED จะสว่าง สภาวะ 0 LED ดับ โดยเขียนโปรแกรมทดลองได้ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



คำสั่ง	อุปกรณ์
LD	X 00
OUT	Y 00
LD	Y 00
OUT	Y 01
LD	Y 01
OUT	M 00
LD	M 00
OUT	T 00
LD	T 00
OUT	C 00
END	

รูปที่ 4.9 โปรแกรมทดลองโปรแกรมย่อย

1) คำอธิบาย

1.1) คำสั่ง LD เป็นการอ่านค่าสภาวะของหน่วยต่างๆ โดยแยกตามค่าอุปกรณ์

1.2) คำสั่ง OUT เป็นการส่งผลลัพธ์ของโปรแกรมแลดเดอร์ไปควบคุมส่วนอุปกรณ์

1.3) อุปกรณ์หมายเลข X 00 เป็นหน่วยอินพุต

อุปกรณ์หมายเลข Y 00 เป็นหน่วยเอาต์พุต

อุปกรณ์หมายเลข M 00 เป็นหน่วย AUX RELAY

อุปกรณ์หมายเลข T 00 เป็นหน่วย TIMER

อุปกรณ์หมายเลข C 00 เป็นหน่วย CNT

การทดลองโปรแกรมย่อย LD นี้จะเลือกส่วนอุปกรณ์ทุกชนิดทดลองเพียงอย่างละ 1 หน่วย

2) ผลการทดลอง

เมื่อรันโปรแกรม LED ทุกดวงจะดับ เพราะอินพุตเบอร์ X 00 ไม่ได้รับ INPUT และ OUTPUT หมายเลข Y 00 อยู่ในสภาวะ OFF, AUX RELAY เบอร์ M 00 ก็จะมีสภาวะเป็น OFF เนื่องจากการเคลียร์ค่า AUX RELAY ทุกตัวให้ OFF ตอนแรกของโปรแกรมหลัก TIMER หมายเลข T 00 จะมีสภาวะ OFF เพราะยังไม่ได้ถูกเรียกใช้งาน COUNTER หมายเลข C 00 ก็เป็นเช่นเดียวกับ TIMER

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อทดลองให้อินพุตที่อินพุตหมายเลข X 00 เป็นลอจิก 1จะเป็นผลทำให้ LED ของเอาต์พุตหมายเลข Y 00 ติด และจากการ LD ค่าสภาวะของเอาต์พุตหมายเลข Y 00 มา OUTPUT ออกที่เอาต์พุต หมายเลข Y 01 ก็จะทำให้ LED ของเอาต์พุตหมายเลข Y 01 ติดด้วย

แสดงถึงว่าโปรแกรมย่อย LD สามารถทำการนำค่าสภาวะของอุปกรณ์มาใช้ในโปรแกรมได้อย่างถูกต้อง

4.2.2 การทดลองโปรแกรมย่อย LD-NOT

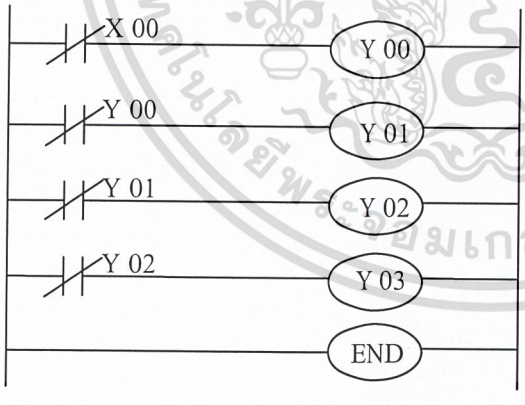
โปรแกรมย่อย LD-NOT เป็นการอ้างถึงสภาวะของอุปกรณ์ในสภาวะตรงข้ามกับสภาวะจริง (สภาวะแบบตรงกันข้าม) โดยเขียนโปรแกรมทดลองดังนี้

1) ผลการทดลอง

เมื่อรันโปรแกรม LED ประจำตำแหน่งเอาต์พุตหมายเลข Y 00, Y 02, Y 03, Y 04 จะติดสว่างเพราะสภาวะจริงของอุปกรณ์ทุกตัวที่ทำการLD-OUT เป็น 0 แต่ถูกกลับสภาวะเป็นตรงกันข้ามจึงได้สภาวะเป็น 1 แต่เอาต์พุตหมายเลข Y 01 จะดับ

เมื่อให้อินพุตลอจิก 1 แต่อุปกรณ์อินพุตหมายเลข X 00 โปรแกรมย่อย LD จะกลับสภาวะให้เป็น 0 ทำให้ส่งสภาวะ 0 ให้เอาต์พุตหมายเลข Y 00 LED ก็จะดับ และเมื่อ LD-NOT อุปกรณ์หมายเลข Y 00 ส่งให้ OUTPUT หมายเลข Y 01 LED หมายเลข Y 01 ก็จะสว่าง

แสดงว่าโปรแกรมย่อยLD-NOT ทำการกลับสภาวะจริงให้เป็นตรงกันข้ามได้อย่างถูกต้อง



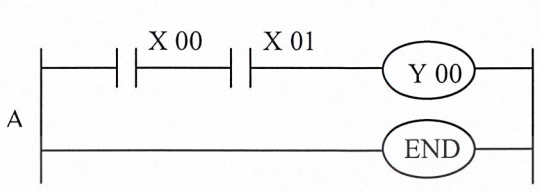
คำสั่ง	อุปกรณ์
LD- NOT	X 00
OUT	Y 00
LD-NOT	Y 00
OUT	Y 01
LD-NOT	Y 01
OUT	Y 02
END	

รูปที่ 4.10 โปรแกรมทดลองโปรแกรมย่อย LD-NOT

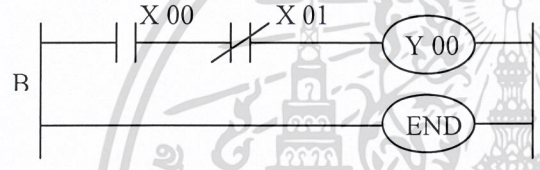
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2.3 การทดลองโปรแกรมย่อย AND และ AND-NOT

โปรแกรมย่อยANDและAND-NOTเป็นการนำเอาค่าสภาวะของอุปกรณ์2 ตัวมาต่ออนุกรมกัน โดยเขียนโปรแกรมทดลองดังนี้



คำสั่ง	อุปกรณ์
LD	X 00
AND	X 01
OUT	Y 00
END	



คำสั่ง	อุปกรณ์
LD	X 00
AND-NOT	X 01
OUT	Y 00
END	

รูปที่ 4.11 โปรแกรมทดลองโปรแกรมย่อย AND, AND-NOT

1) ผลการทดลอง

โปรแกรม A เป็นการทดลองโปรแกมย่อย AND เมื่อรันโปรแกรมหากไม่ได้ให้อินพุตแก่อุปกรณ์อินพุตหมายเลข X 00 และ X 01 หรือให้อินพุตลอจิก 1 แก่อุปกรณ์ตัวใดตัวหนึ่งเท่านั้น อุปกรณ์เอาต์พุต หมายเลข Y 00 ก็ยังคง OFF

เมื่อให้อินพุตแก่อุปกรณ์หมายเลข X 00 และ X 01 เป็น 1 ทั้งคู่อุปกรณ์ เอาต์พุต หมายเลข Y 00 จึงจะ ON ทำให้ LED ประจำเอาต์พุตติดสว่าง

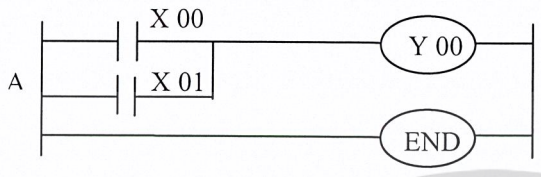
โปรแกรม B ทดลองโปรแกรมย่อย AND-NOT เมื่อรันโปรแกรมแล้วให้ลอจิก 1 แก่อุปกรณ์อินพุต หมายเลข X 00 ทำให้ LED ประจำเอาต์พุต Y 00 ติดสว่าง นอกจากการให้อุปกรณ์อินพุตเป็นแบบนี้แล้ว กรณีอื่นจะให้อุปกรณ์เอาต์พุตหมายเลข Y 00 เป็น OFF ทั้งหมด

แสดงถึงว่าการทำงานของการ AND และ AND-NOT ทำงานได้ถูกต้อง

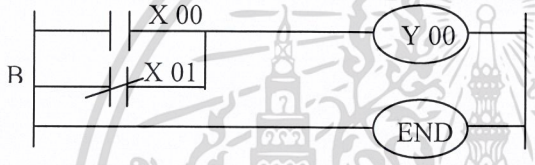
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2.4 การทดลองโปรแกรมย่อย OR และ OR-NOT

โปรแกรมย่อย OR และ OR-NOT เป็นการนำค่าสภาวะของอุปกรณ์ 2 ตัวมาต่อขนานกัน โดยเขียนโปรแกรมทดลองได้ดังนี้



คำสั่ง	อุปกรณ์
LD	X 00
OR	X 01
OUT	Y 00
END	



คำสั่ง	อุปกรณ์
LD	X 00
OR-NOT	X 01
OUT	Y 00
END	

รูปที่ 4.12 โปรมแกรมทดลองโปรแกรมย่อย OR, OR-NOT

1) ผลการทดลอง

โปรแกรม A เป็นการทดลองโปรแกรมย่อย OR เมื่อรันโปรแกรมหากให้สภาวะ 0 แก่อินพุต หมายเลข X 00 และ X 01 จะทำให้ LED แสดงสภาวะเอาต์พุตหมายเลข Y 00 ดับ นอกจากนี้ในกรณีป้อนอินพุตแบบอื่น จะทำให้ LED ติดสว่าง

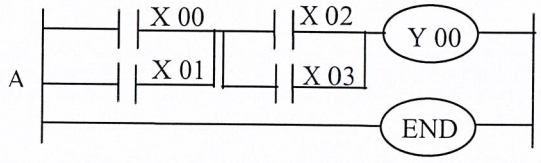
โปรแกรม B เป็นการทดลองโปรแกรมย่อย OR-NOT เมื่อรันโปรแกรมหากให้สภาวะ 0 แก่อินพุต หมายเลข X 00 และให้ลอจิก 1 แก่อุปกรณ์อินพุตหมายเลข X 01 LED แสดงสภาวะเอาต์พุต หมายเลข Y 00 จะดับ นอกจากนี้ในกรณีป้อนอินพุตแบบอื่นจะทำให้ LED ติดสว่าง

แสดงว่าการทำงานของโปรแกรมย่อย OR และ OR-NOT ทำงานได้ถูกต้อง

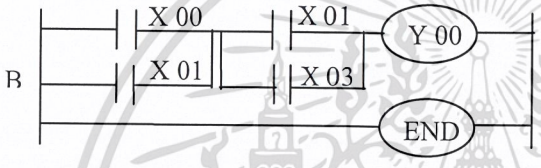
4.2.5 การทดลองโปรแกรมย่อย ANB, ORB

โปรแกรมย่อย ANB เป็นการนำเอากลุ่ม (BLOCK) ของอุปกรณ์ 2 กลุ่มมาต่ออนุกรมกัน โปรแกรมย่อย ORB เป็นการนำเอากลุ่ม (BLOCK) ของอุปกรณ์ 2 กลุ่มมาต่อขนานกัน โดยเขียนโปรแกรมทดลองดังนี้

เอกสารนี้สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



คำสั่ง	อุปกรณ์
LD	X 00
OR	X 01
LD	X 02
OR	X 03
ANB	
OUT	Y 00
END	



คำสั่ง	อุปกรณ์
LD	X 00
AND	X 01
LD	X 02
AND	X 03

รูปที่ 4.13 โปรแกรมทดลองโปรแกรม ANB, ORB

1) ผลการทดลอง

โปรแกรม A เป็นการทดลองโปรแกรมย่อย ANB หากให้ลอจิก 1 แก่อินพุต หมายเลข X 00 และ X 01 ตัวใดตัวหนึ่งหรือทั้ง 2 ตัว และให้ลอจิก 1 แก่อุปกรณ์อินพุต หมายเลข X 02 และ X 03 ตัวใดตัวหนึ่งหรือทั้ง 2 ตัว จะทำให้ LED แสดงสภาวะของเอาต์พุตหมายเลข Y 00 ติดสว่าง หากให้ลอจิกแก่อินพุตในแบบอื่นใดนอกจากนี้ LED จะไม่ติด

โปรแกรม B เป็นการทดลองโปรแกรมย่อย ORB หากให้ลอจิก 1 แก่อินพุต หมายเลข X 00 และ X 01 ทั้งคู่ หือให้แก่ทั้ง 4 ตัวจะทำให้ LED แสดงสภาวะของเอาต์พุตหมายเลข Y 00 ติดสว่าง หากให้ลอจิกแก่อินพุตในแบบอื่นใดนอกจากนี้ LED จะไม่ติด

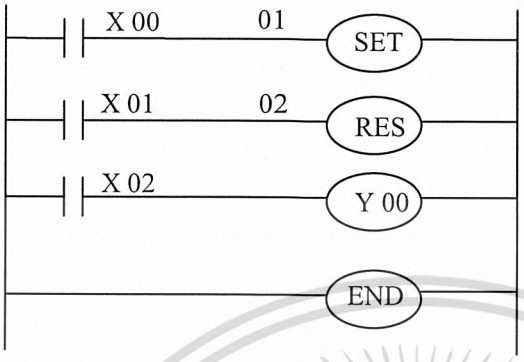
แสดงว่าการทำงานของโปรแกรมย่อย ANB, ORB ทำงานได้ถูกต้อง

4.2.6 การทดลองโปรแกรมย่อย SET, RES

โปรแกรมย่อย SET เป็นการนำให้หน่วยอินพุตหรือหน่วยเอาต์พุตมีสภาวะการทำงานเป็นจริงตลอดเวลาหรือ ON ตลอดเวลา โดยไม่ขึ้นกับสภาวะจริง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรมย่อย RES เป็นการนำให้อินพุตหรือหน่วยเอาต์พุตมีสภาวะการทำงานเป็นเท็จหรือ OFF ตลอดเวลา โดยไม่ขึ้นกับสภาวะจริง เขียนโปรแกรมทดลองได้ดังนี้



คำสั่ง	อุปกรณ์
LD	X 00
SET	01
LD	X 01
RES	02
LD	X 02
OUT	Y 00
END	

รูปที่ 4.14 โปรแกรมทดลองโปรแกรมย่อย SET, RES

1) ผลการทดลอง

เมื่อรันโปรแกรมและให้ลอจิก 1 แก่อินพุตหมายเลข X 00 จะทำให้ LED แสดงสภาวะของอุปกรณ์เอาต์พุตหมายเลข Y 00 ติดสว่าง ในขณะที่ยังให้ลอจิก 1 แก่อินพุต หมายเลข X 00 อยู่ ให้ลอจิก 1 แก่อินพุตหมายเลข X 01 ด้วยมีผลทำให้ LED แสดงสภาวะของเอาต์พุตหมายเลข Y 00 ติด

แสดงว่าการทำงานของโปรแกรมย่อย SET, RES ทำงานได้ถูกต้อง

4.2.7 การทดลองโปรแกรมย่อย TIMER

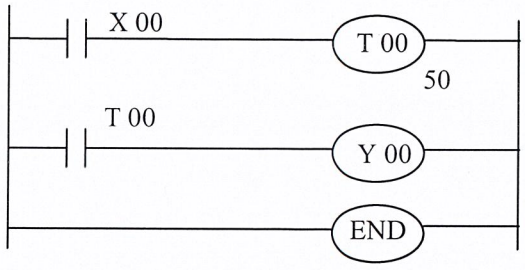
โปรแกรมย่อยนี้จะทำการตั้งเวลาตามที่กำหนดโดยตั้งได้สูงสุด 99.9 วินาทีต่อ TIMER 1 ตัว เขียนโปรแกรมทดลองได้ดังนี้

1) ผลการทดลอง

เมื่อให้ลอจิก 1 แก่อุปกรณ์อินพุตหมายเลข X 00 และเริ่มจับเวลาพอเวลาผ่านไป 5 วินาที จะทำให้ LED แสดงสภาวะของเอาต์พุตหมายเลข Y 00 ติดสว่าง

แสดงว่าโปรแกรมย่อย TIMER ทำการลดค่าลงตามเวลาที่ตั้งไว้ได้ถูกต้อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

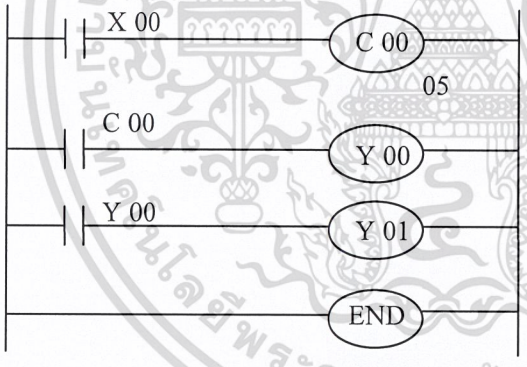


คำสั่ง	อุปกรณ์
LD	X 00
TIME	T 00
	50
LD	T 00
OUT	Y 00
END	

รูปที่ 4.15 โปรแกรมทดลองโปรแกรมย่อย TIME

4.2.8 การทดลองโปรแกรมย่อย COUNTER

โปรแกรมย่อยนี้จะทำการตั้งค่า COUNTER โดยตั้งได้สูงสุด 999 ครั้งต่อ COUNTER 1 ตัว เขียนโปรแกรมทดลองได้ดังนี้



คำสั่ง	อุปกรณ์
LD	X 00
CNT	C 00
	05
LD	C 00
OUT	Y 00
LD	Y 00
OUT	Y 01
END	

รูปที่ 4.16 โปรแกรมทดลองโปรแกรมย่อย COUNTER

1) ผลการทดลอง

ให้ลอจิก 1 แก่อุปกรณ์อินพุตหมายเลข X 00 ตลอดเวลา และให้ลอจิก 1 กับลอจิก 0 สลับกันแก่อุปกรณ์อินพุตหมายเลข X 00 ทั้งหมด อย่างละ 5 ครั้ง จะทำให้ LED แสดงสภาวะของเอาต์พุตหมายเลข Y 00 ติดสว่าง

แสดงว่า COUNTER ได้ทำการนับตามเวลาที่ตั้งไว้ถูกต้อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5
บทสรุป ปัญหา แนวทางแก้ไข และพัฒนา

5.1 บทสรุป

จากการทำโครงงานพีแอลซีในส่วนของซอฟท์แวร์ต้องทำการเขียนโปรแกรมเดลไฟล์ (DELPHI) เพื่อแปลงภาษาบูลีน เป็นไฟล์ในรูปแบบของ Intel-Hex File ซึ่งมีขั้นตอนในการแปลงภาษามากมาย และต้องใช้วิธีการตัดคำออกเป็นส่วน ๆ ซึ่งทางผู้จัดทำต้องใช้เวลาในการศึกษาโปรแกรมภาษาต่างๆ และพบปัญหาในการคอมไพล์ (Compile), ความไม่สอดคล้องกันระหว่างภาษาต่างๆ, ปัญหาในการแปลงภาษาแอสแซมบลี้เป็นออพโค้ด (Opcode) โดยใช้ภาษาเดลไฟล์

ในส่วนของฮาร์ดแวร์ การทดสอบวงจรไมโครคอนโทรลเลอร์ ซึ่งต้องแยกส่วนในการทดสอบ เช่น

- การเก็บข้อมูลลงการใช้รอมและแรม
- การส่งข้อมูลแบบอนุกรมผ่านพอร์ท RS-232
- ทดสอบการทำงานของพอร์ต

ส่วนวงจรอินพุตและวงจรเอาต์พุตจะมีปัญหาตรงที่ เมื่อเรายังไม่ได้กดอินพุต แต่ทำการรันโปรแกรมแล้ว ปรากฏว่าเอาต์พุตออกแสดงที่ LCD เลย ปัญหาส่วนใหญ่จะเกิดขึ้นในส่วนโปรแกรมเป็นส่วนใหญ่

จุดประสงค์ของการทำปริญญานิพนธ์นี้คือ สามารถนำพีแอลซีไปใช้ในการควบคุมสวิตช์ของเครื่องจักรต่างๆ ได้จริง แล้วยังจะทำให้ประหยัดเวลาในการปิดและเปิดสวิตช์ ประหยัดพลังงานไฟฟ้า และยังช่วยควบคุมกระบวนการผลิตถูกต้องตามที่ได้โปรแกรมไว้ แต่ พีแอลซี นี้ ยังมีข้อเสียอยู่บ้าง เช่น ขนาดของบอร์ดของส่วนประมวลผล ส่วนอินพุตและเอาต์พุตมีขนาดใหญ่เกินไป ซึ่งไม่เหมาะสมกับการใช้งานจริง ซึ่งถ้าทำการออกแบบลายวงจรใหม่ โดยใช้อุปกรณ์เหมือนเดิมทุกอย่างก็จะทำให้พีแอลซีมีขนาดเล็กลง และสามารถนำไปใช้งานได้จริง โดยนำส่วน NO ของรีเลย์ไปต่อกับสวิตช์ของเครื่องจักรต่างๆ

5.2 ปัญหาและอุปสรรค

1. เนื่องจากเป็นเครื่องที่สร้างขึ้นมาใหม่ จึงหาต้นแบบมาศึกษายากทำให้การทำโครงงานล่าช้า และ อาศัยความเข้าใจในการสร้างขึ้นมา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. ข้อมูลเกี่ยวกับเครื่อง PLC ที่ใช้งานจริงเพื่อนำมาเป็นต้นแบบหาได้น้อย เพราะยังไม่มีผู้ทำการเขียนเรื่อง PLC มากนัก รายละเอียดบางส่วนจึงได้จากการศึกษาเครื่อง PLC บ้างจากประสบการณ์ที่เคยใช้บ้าง

3. การทำกล่องต้องทำเป็นกล่องโลหะ เนื่องจากรูปแบบกล่อง ที่ต้องการไม่มีขายในท้องตลาดและร้านที่รับทำหายาก เนื่องจำนวนน้อยร้านส่วนใหญ่จะรับทำจำนวนมากๆ จึงทำให้กล่องเสร็จช้า

5.3 แนวทางการพัฒนา

1. ในขณะนี้การรันโปรแกรมจะเริ่มรันที่บรรทัดแรกเท่านั้น ถ้าหากจะให้ทำงานโปรแกรมอื่นจะต้องเขียนทับของเก่าทำให้จะกลับมาทำงานโปรแกรมเก่าอีกไม่ได้ควรทำให้เลือกรันที่บรรทัดใดก็ได้จะทำให้เขียนโปรแกรมใช้งานได้หลาย ๆ โปรแกรม โดยเขียนขึ้นครั้งเดียว

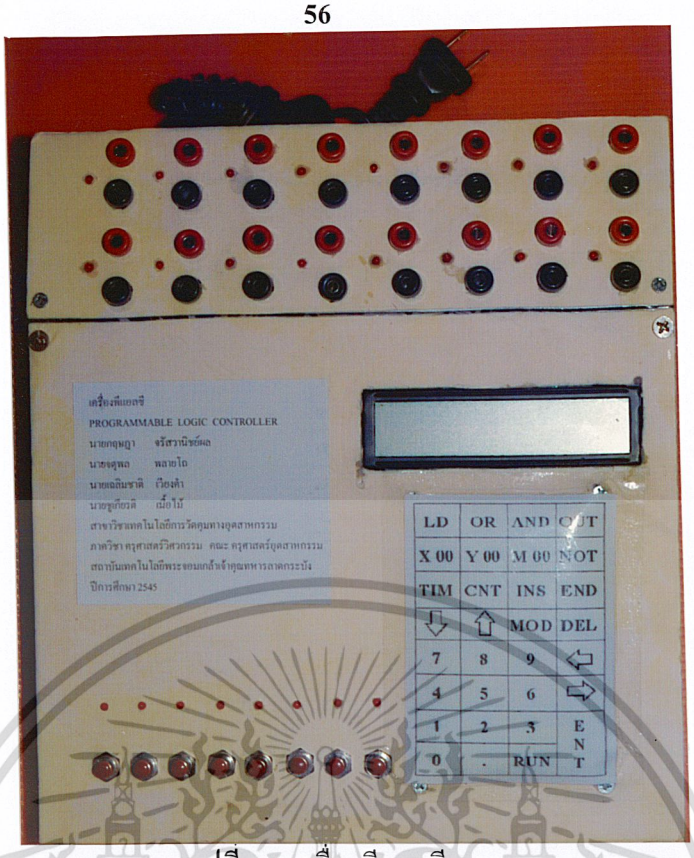
2. เครื่อง PLC ที่ใช้งานจริงจะพัฒนาไปอย่างรวดเร็ว มีฟังก์ชันพิเศษเพิ่มขึ้นมากมายการพัฒนาจึงควรให้มีฟังก์พิเศษมากขึ้น

3. ควรสร้างและพัฒนาขนาดให้เล็กลง เพื่อจะได้สะดวกต่อการเรียนการนำไปศึกษา

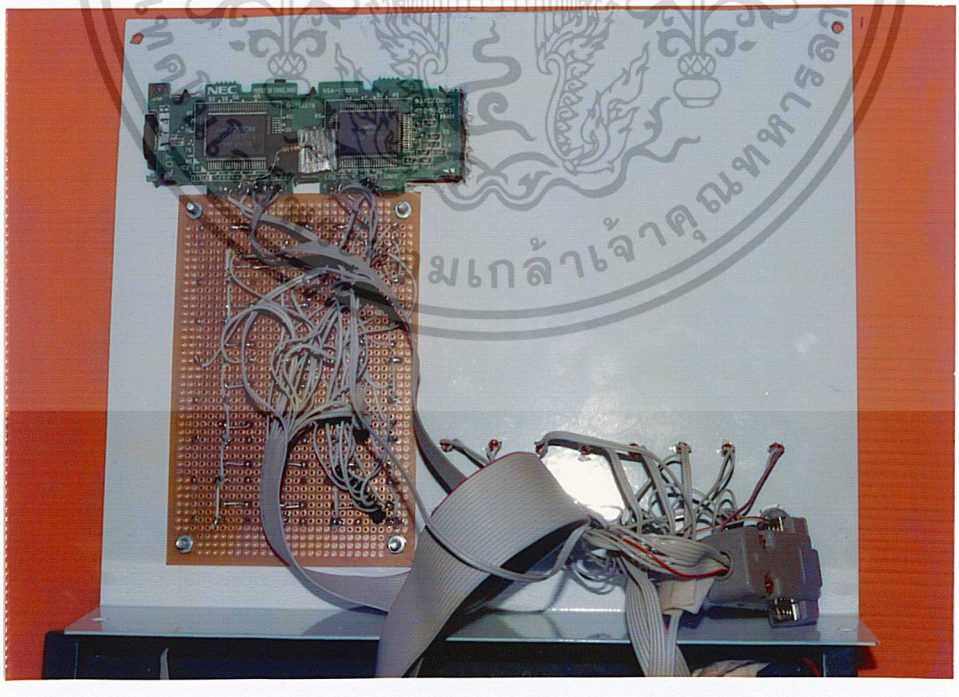
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ก
เครื่องต้นแบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

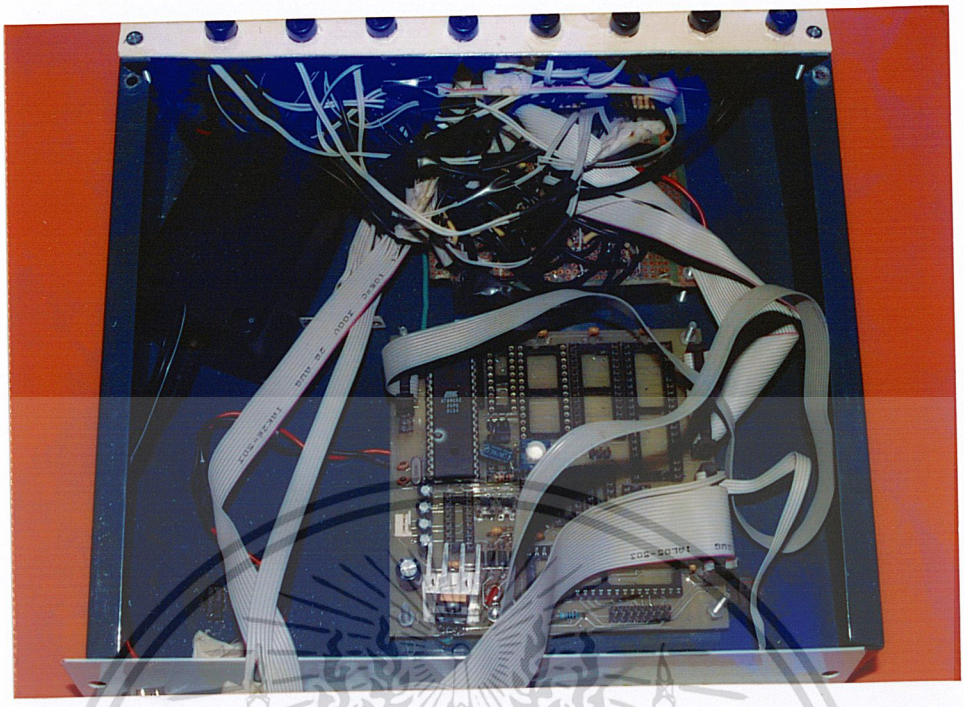


รูปที่ ก.1 เครื่องพีแอลซี

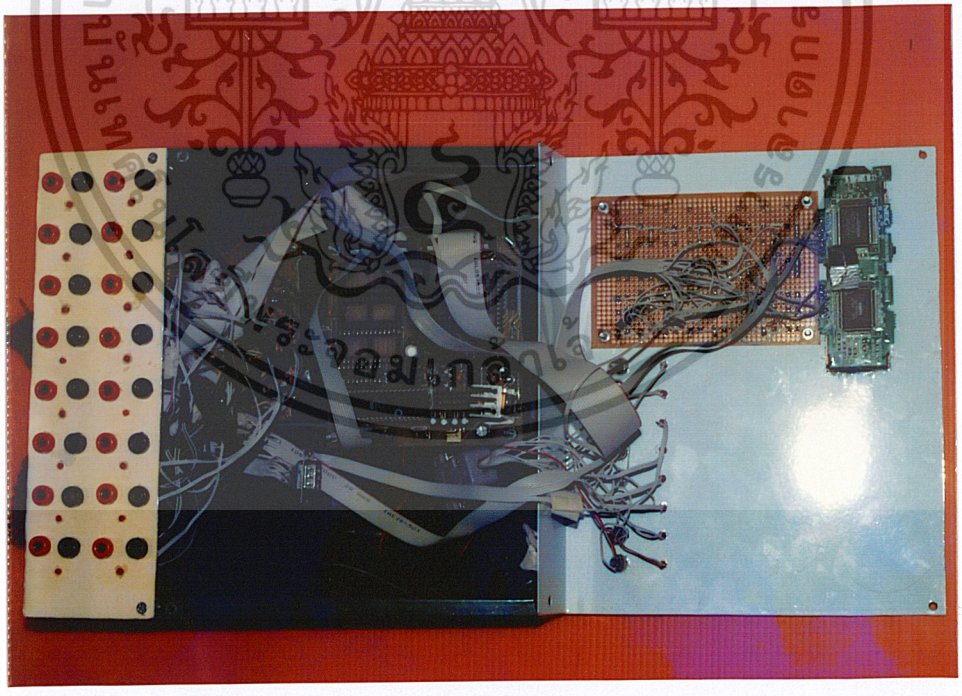


รูปที่ ก.2 ด้านหลังของ KEY BOARD, LCD, SWITCHINPUT

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ ก.3 บอร์ดหลัก MCS-51



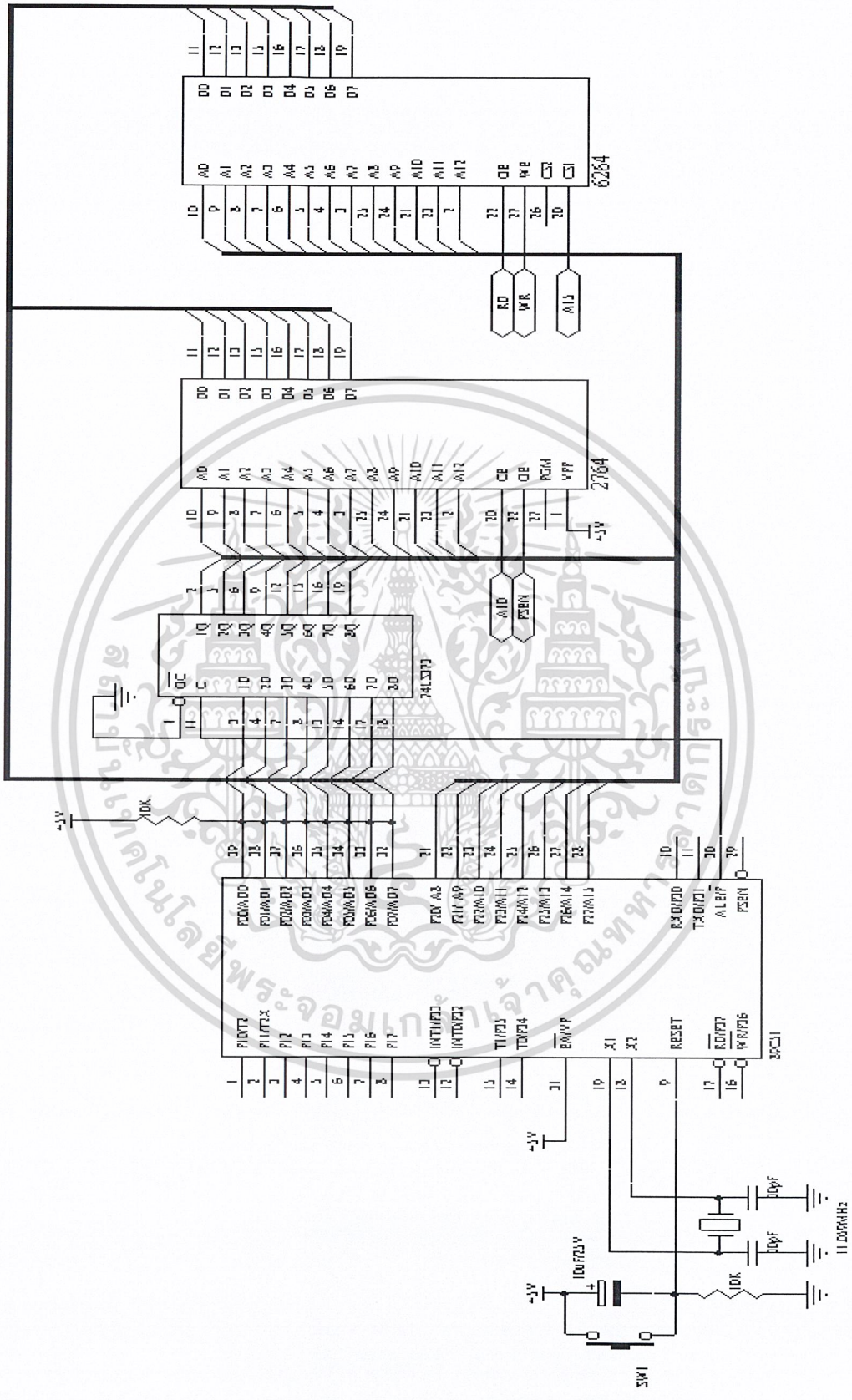
รูปที่ ก.4 ภายในเครื่องพีแอลซี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ข

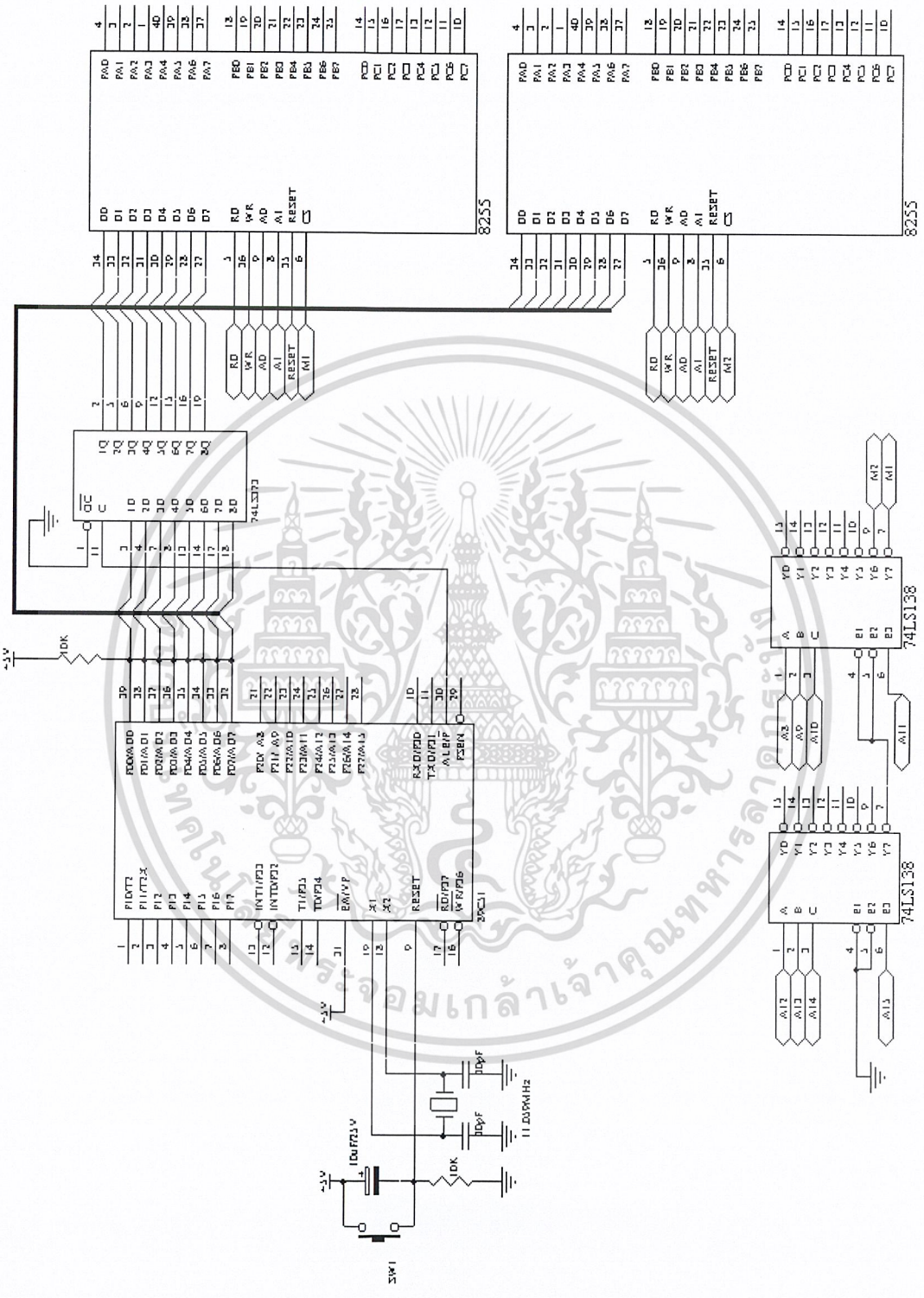
วงจรและแผ่นวงจรพิมพ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



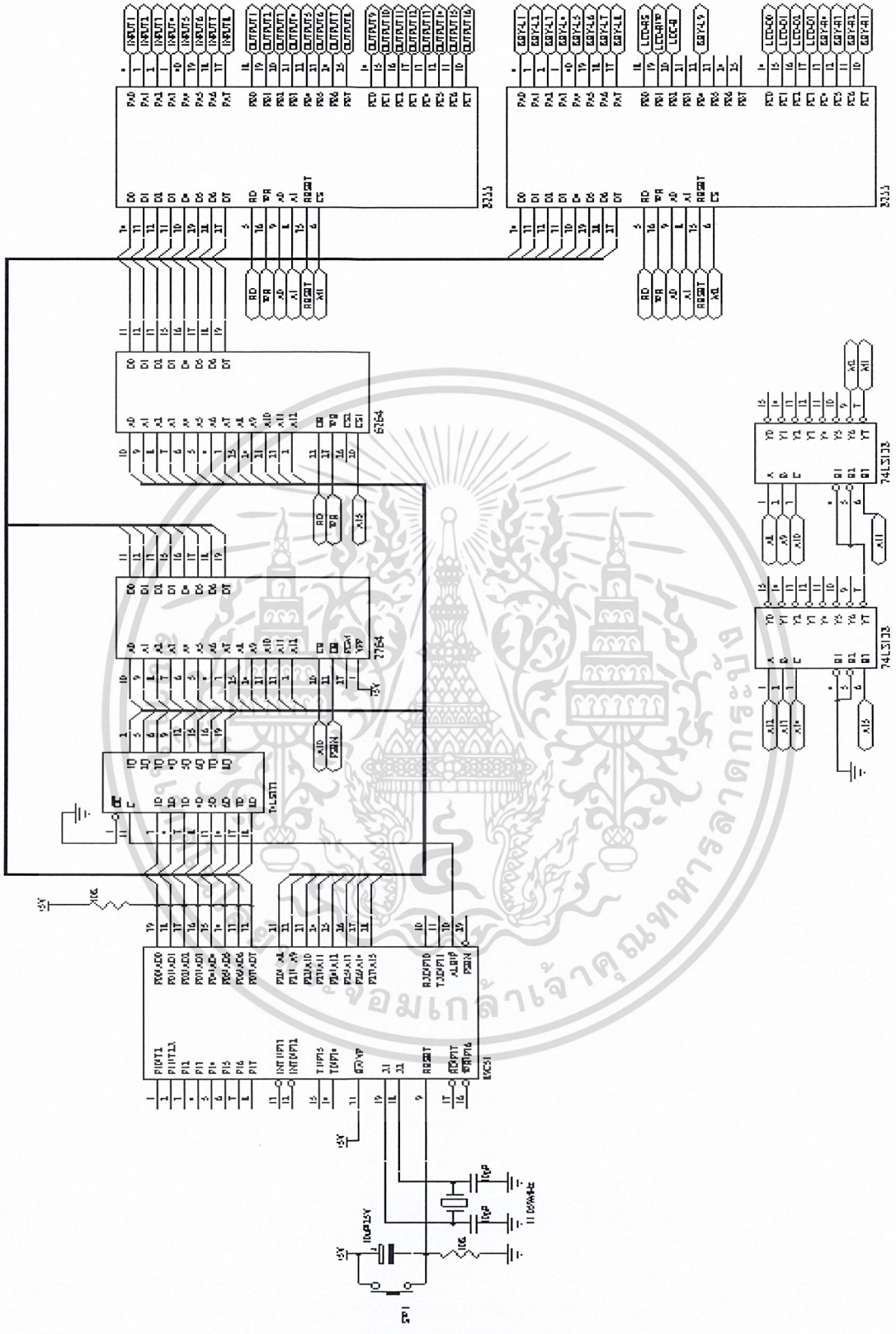
รูปที่ ข.1 การต่อ MCS-51 เข้ากับ EPROM และ RAM

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



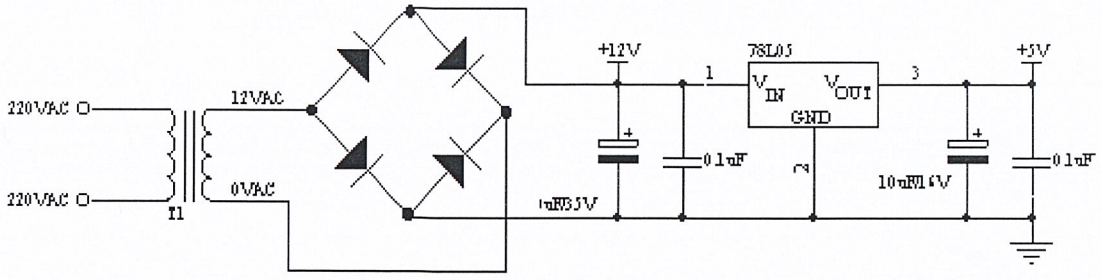
รูปที่ ข.2 การเชื่อมต่อ 8255 กับ MCS-51

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

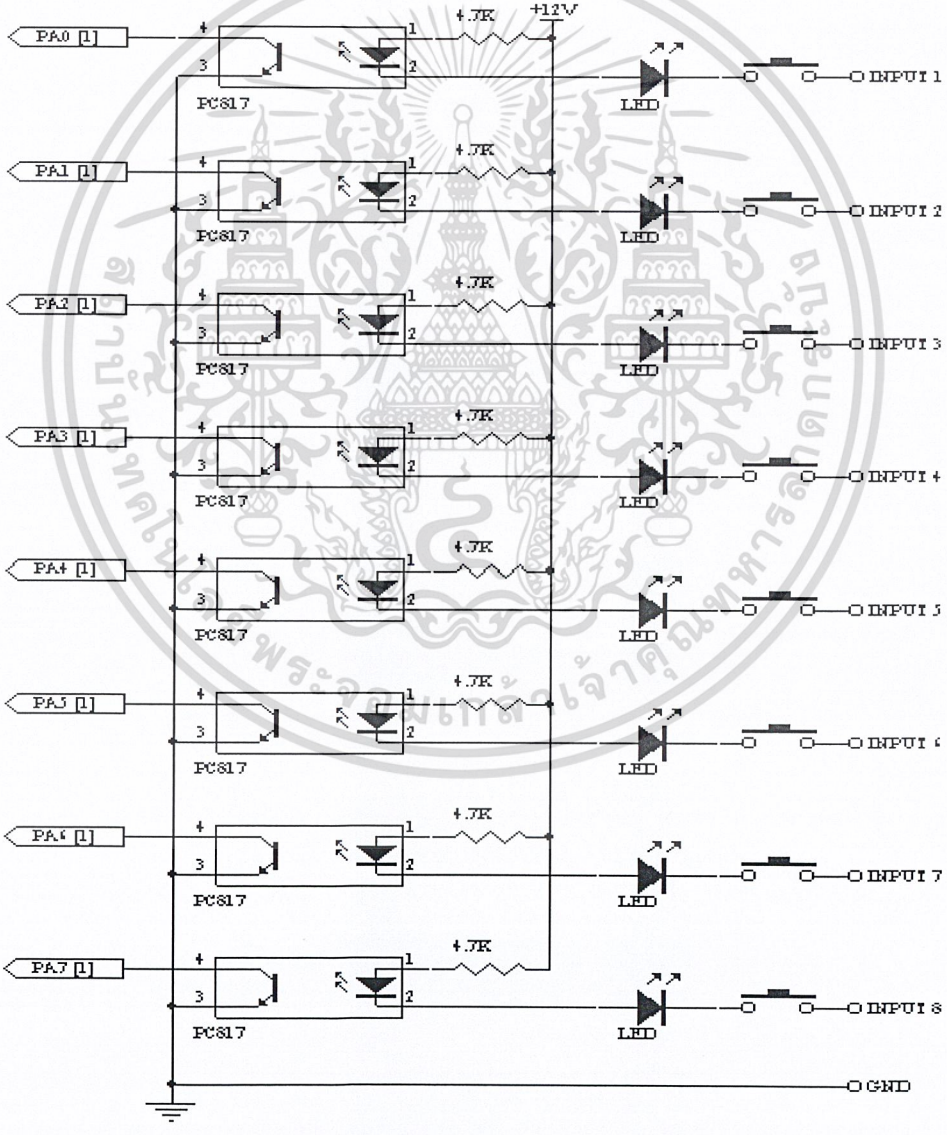


รูปที่ ข.3 บอร์ดหลัก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

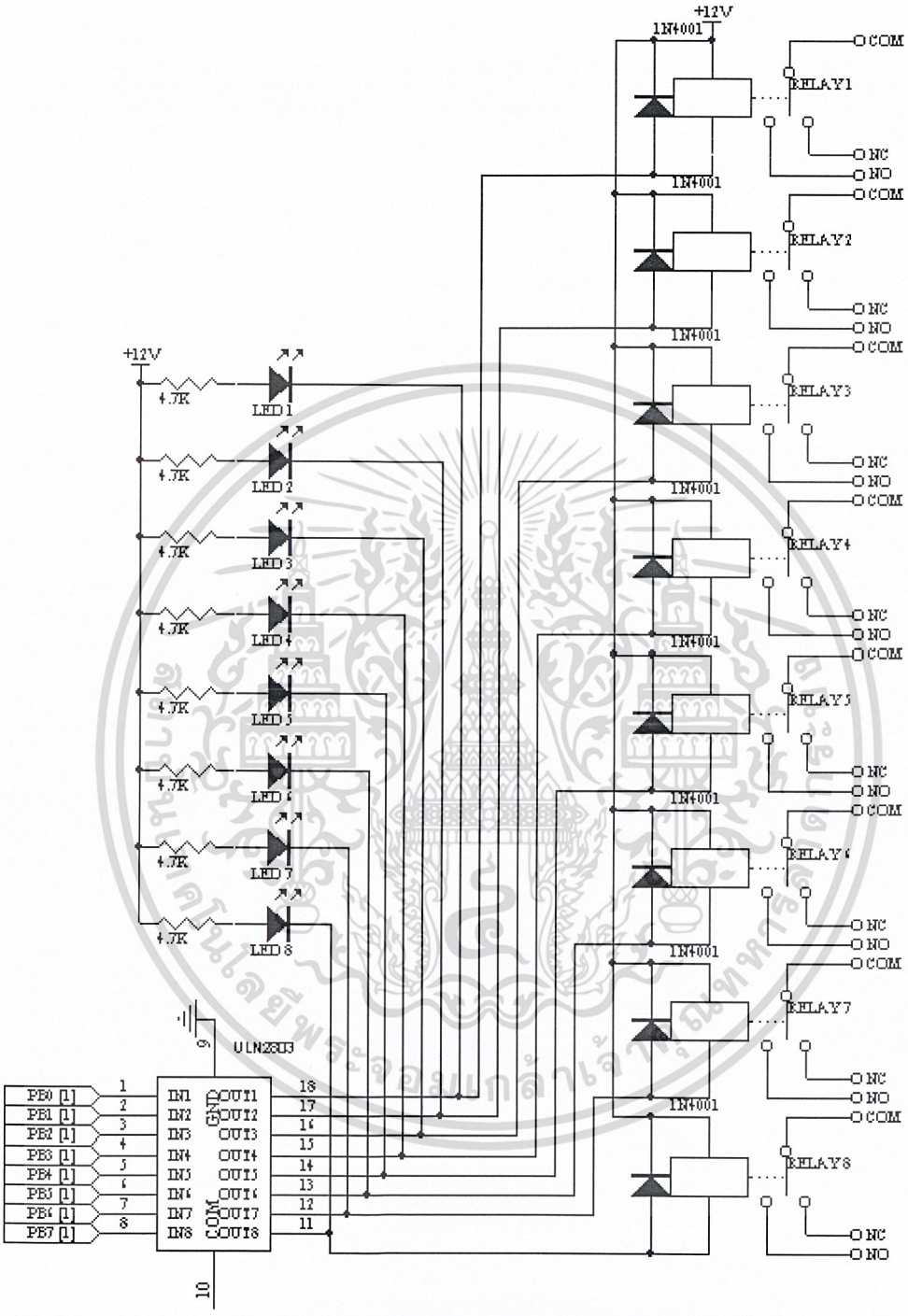


รูปที่ ข.4 แหล่งจ่ายไฟตรง 5 โวลท์



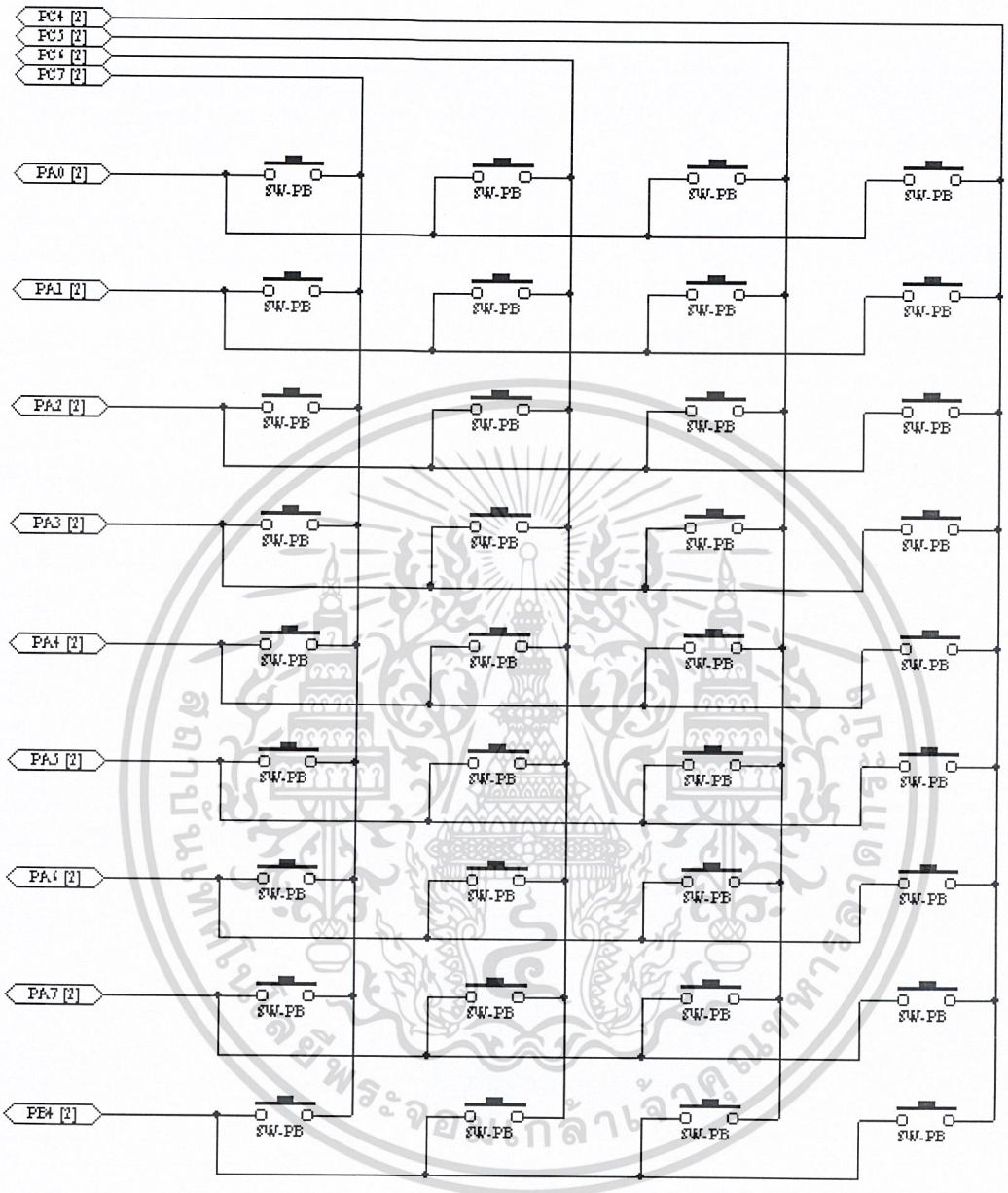
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ ข.5 วงจรอินพุต 8 วงจร



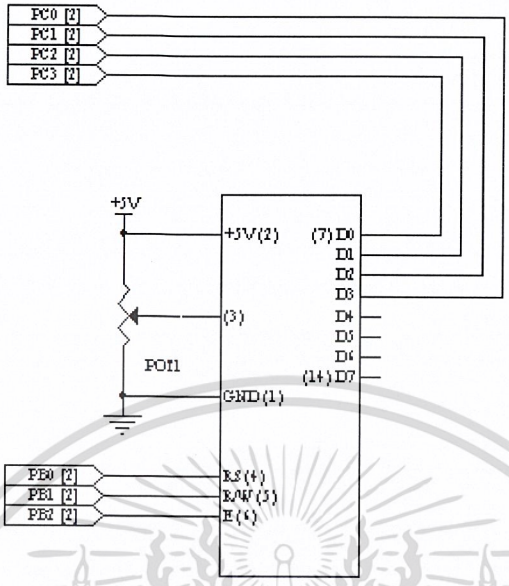
รูปที่ ข.6 วงจรเอาต์พุต 8 วงจร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ ข.7 วงจร KEY BOARD

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



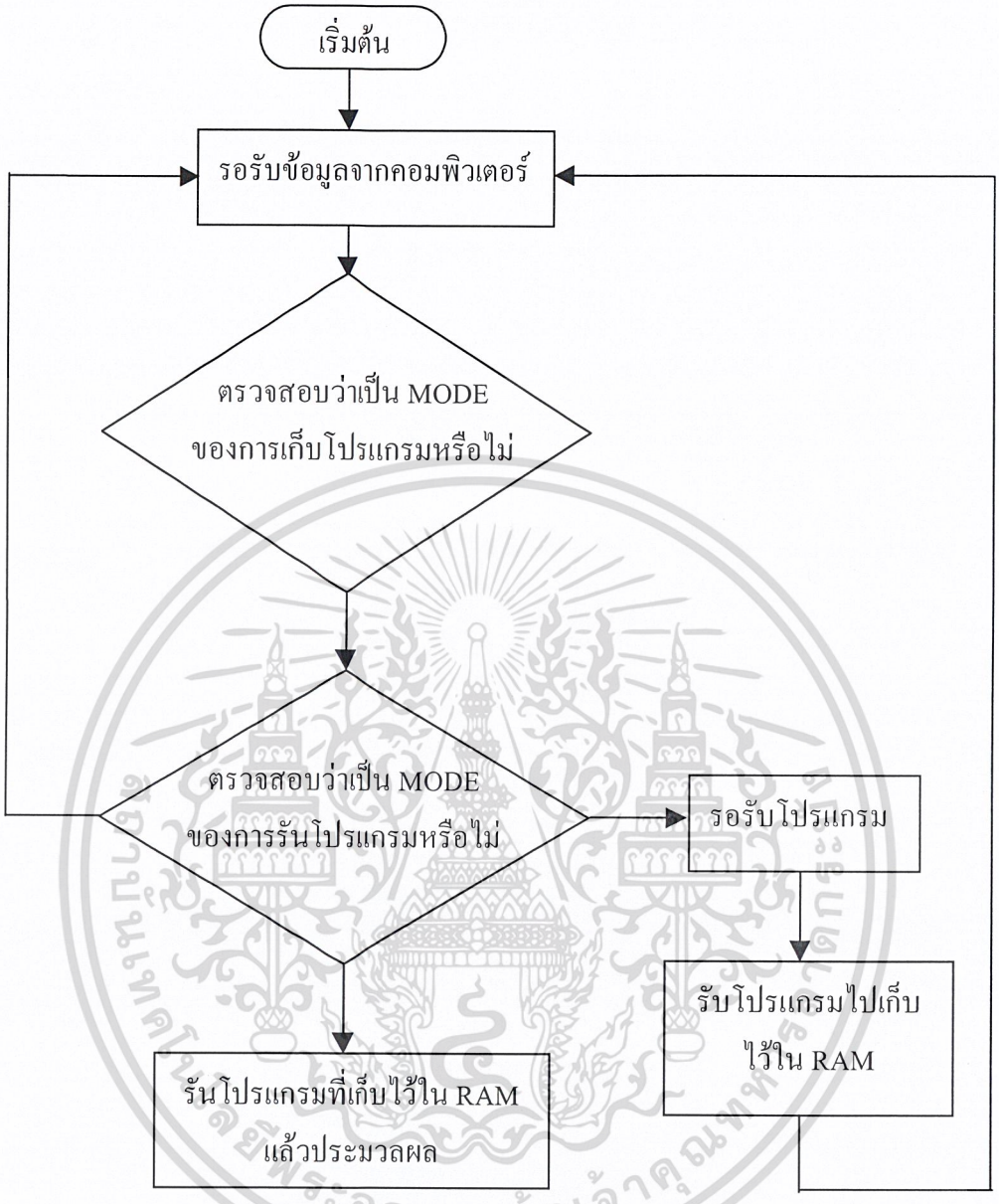
รูปที่ ข.8 วงจร DISPLAY LCD

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ค

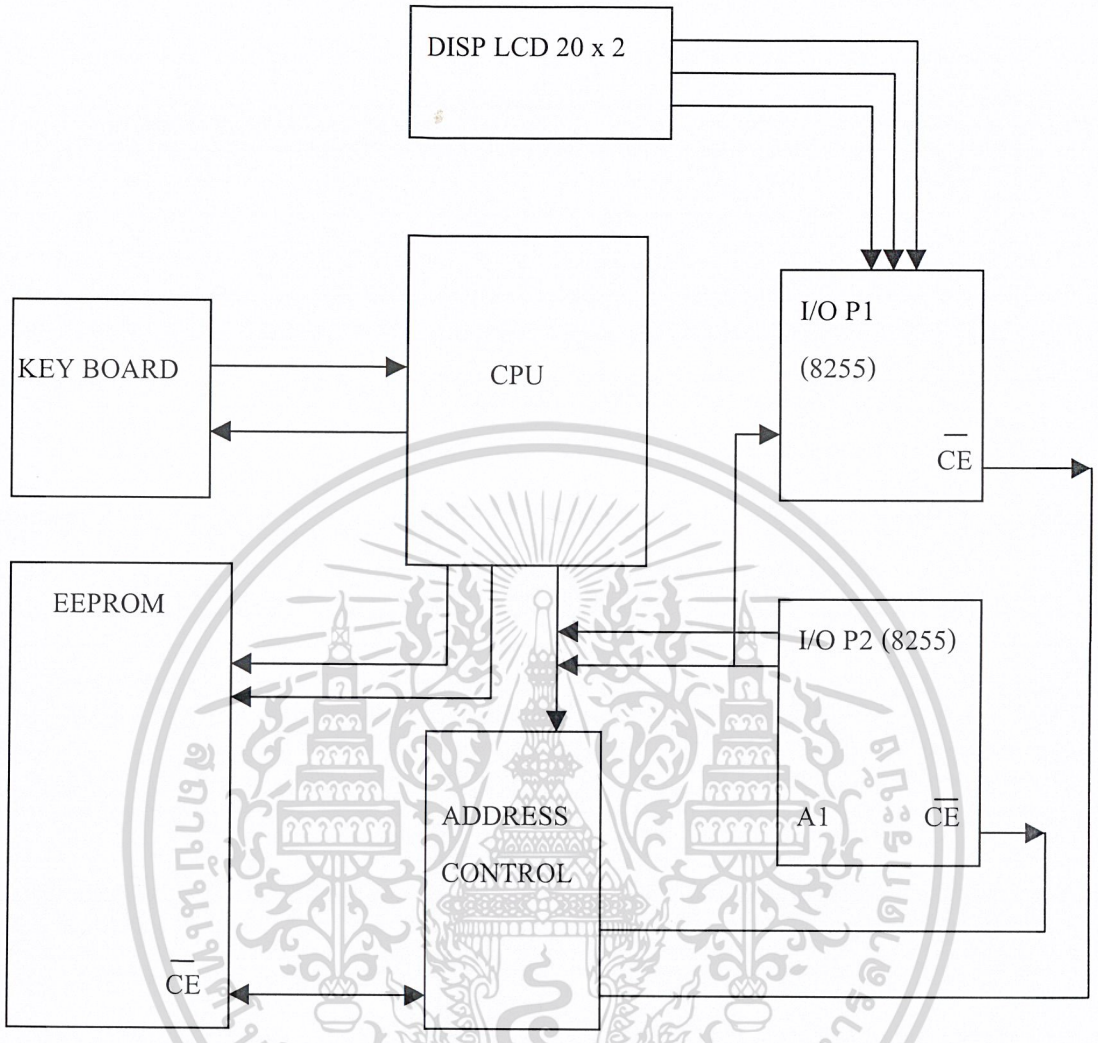
ผังการทำงานและโปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



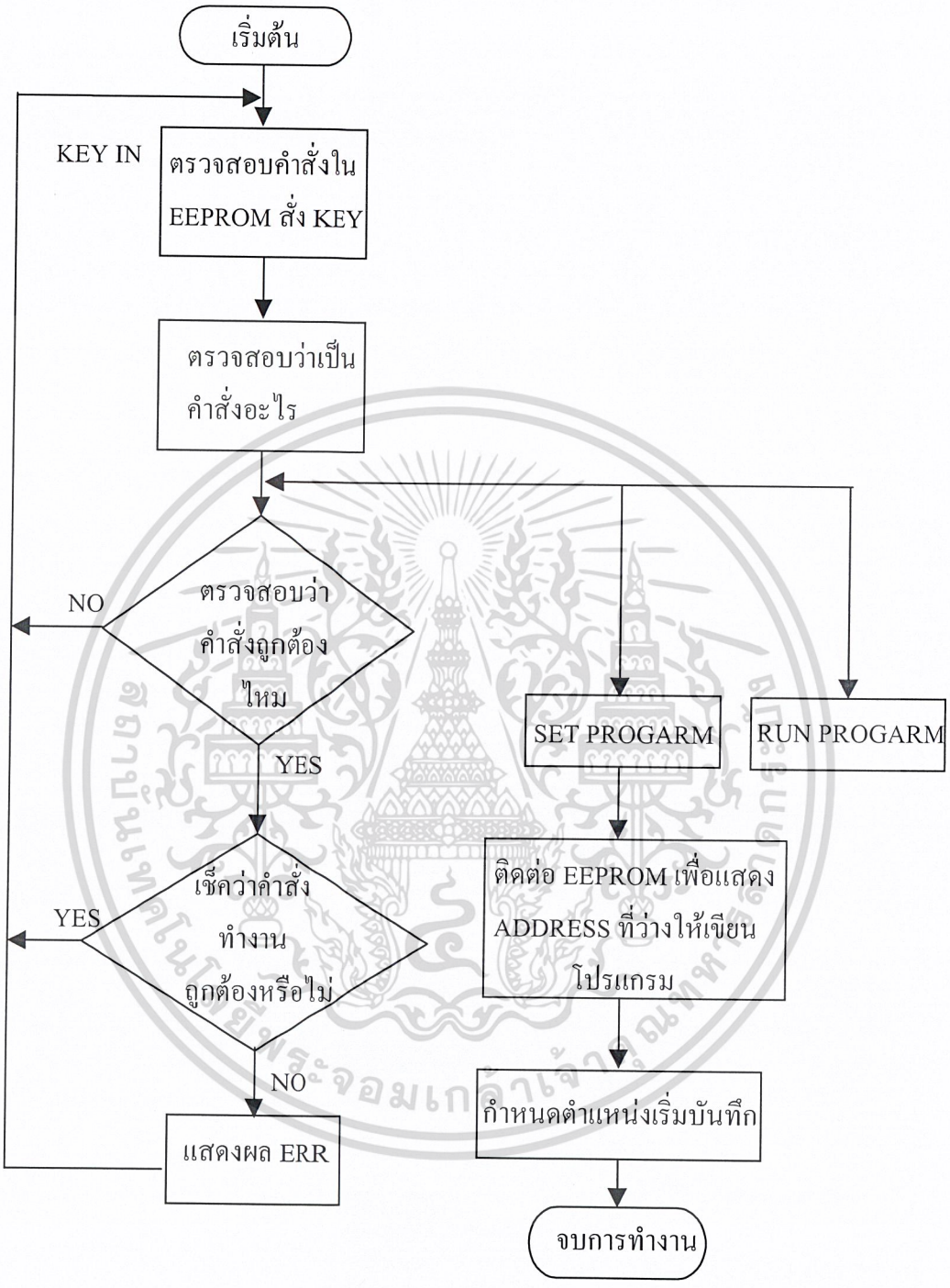
รูปที่ ค.1 ขั้นตอนการทำงานของโปรแกรมภาษาแอสเซมบลีในไมโครคอนโทรลเลอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



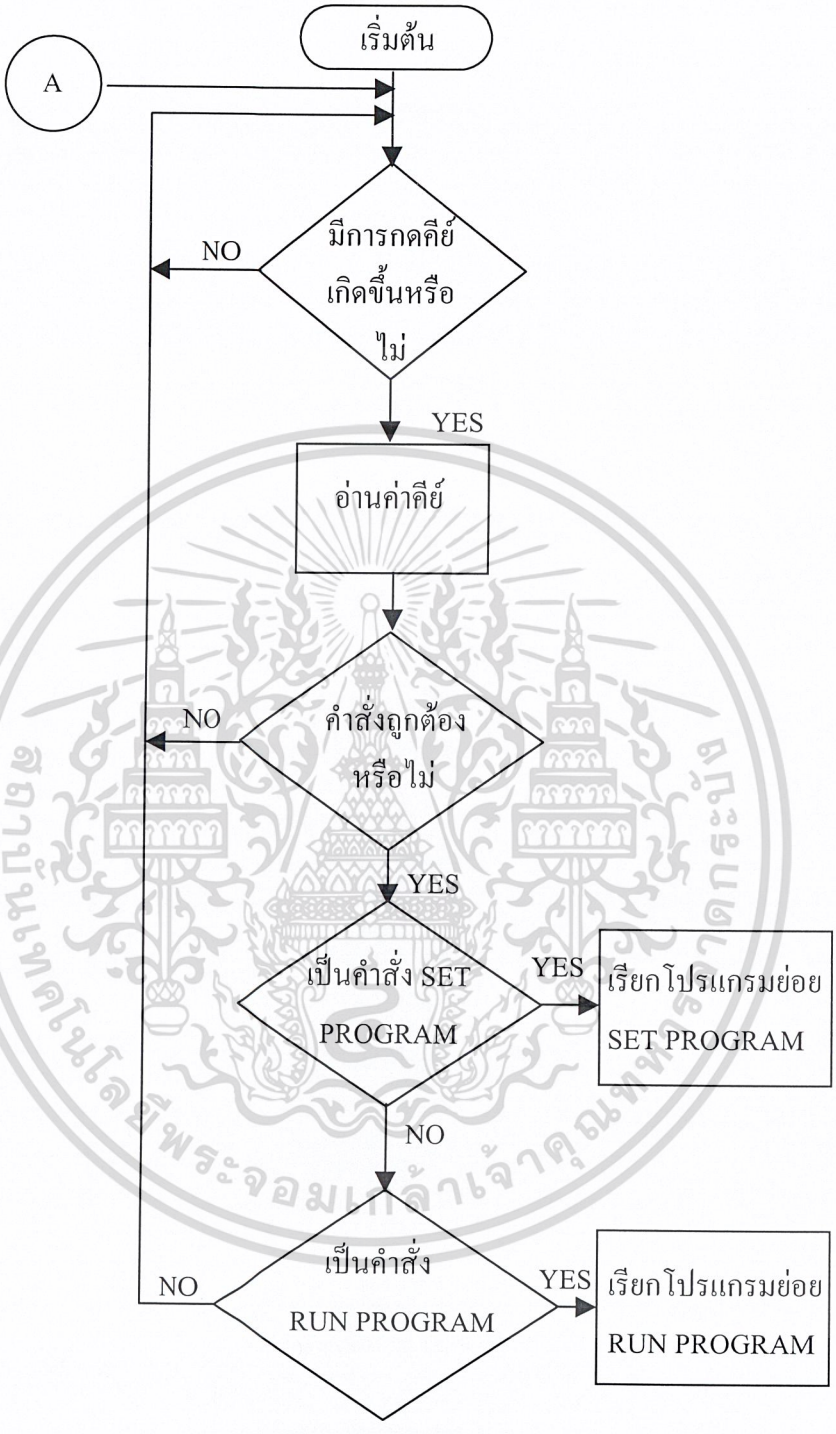
รูปที่ ค.2 ลักษณะโครงสร้าง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



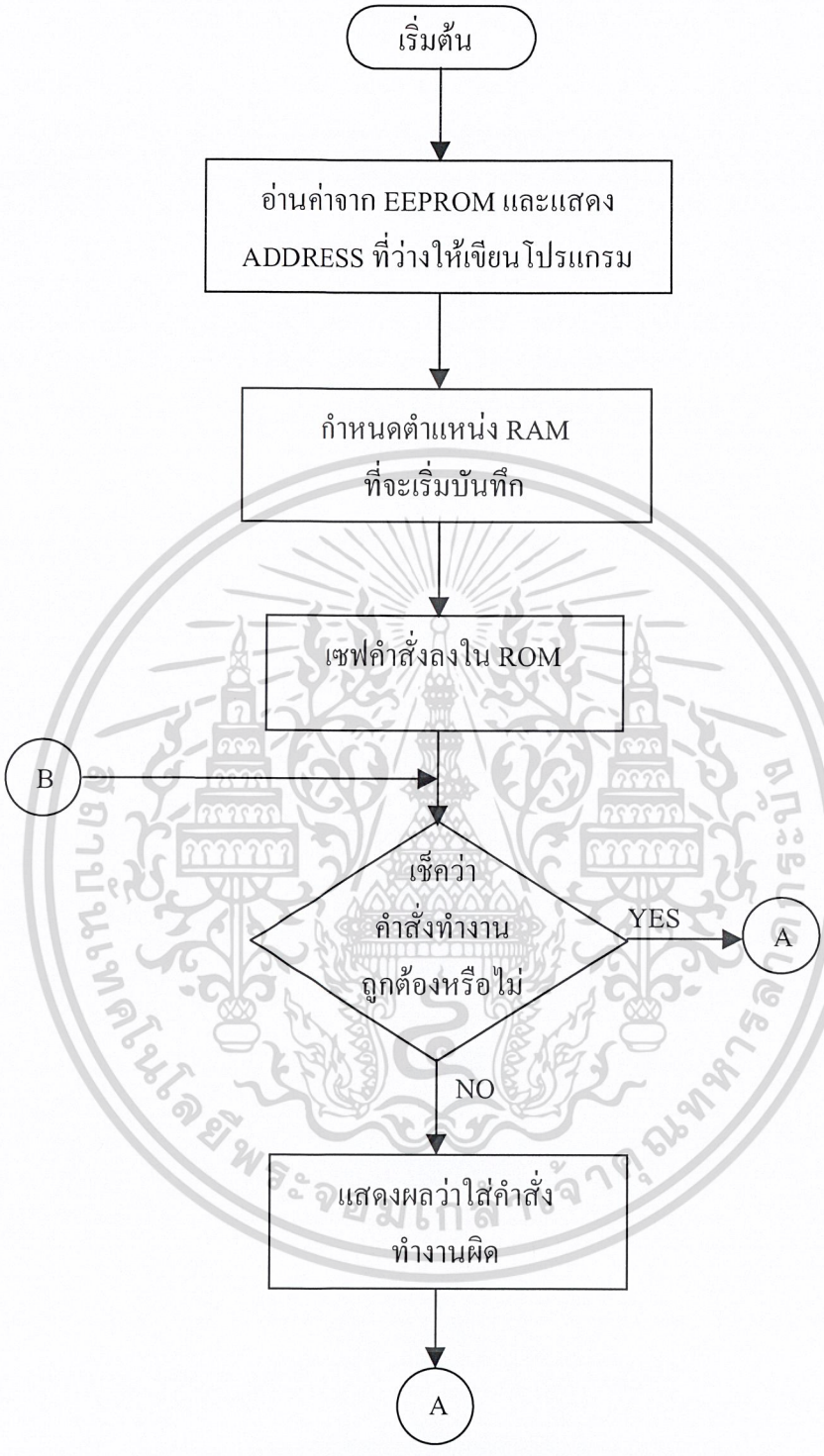
รูปที่ ค.3 ขั้นตอนการใช้งานของพีแอลซี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



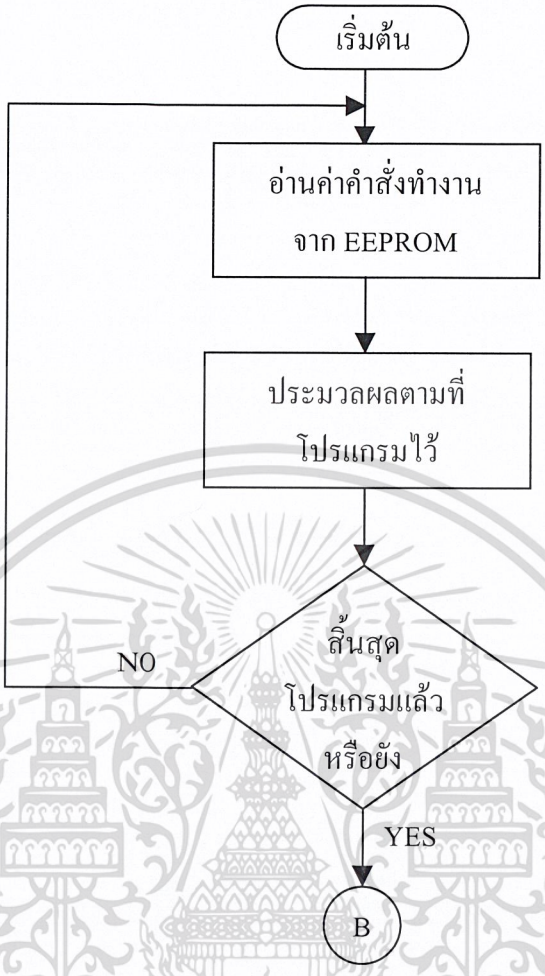
รูปที่ ค.4 ผังการทำงาน MAIN PROGRAM

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ ค.5 โปรแกรมย่อย SET PROGRAM

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ ค.6 โปรแกรมย่อย RUN PROGRAM

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

DMode	Bit	00H
Prog1	Bit	01H
Prog2	Bit	02H
Prog3	Bit	03H
Prog4	Bit	04H
KeySt1	Bit	05H
KeySt2	Bit	06H
KeySt3	Bit	07H
RX00	Bit	10H
RX01	Bit	11H
RX02	Bit	12H
RX03	Bit	13H
RX04	Bit	14H
RX05	Bit	15H
RX06	Bit	16H
RX07	Bit	17H
RM00	Bit	18H
RM01	Bit	19H
RM02	Bit	1AH
RM03	Bit	1BH
RM04	Bit	1CH
RM05	Bit	1DH
RM06	Bit	1EH
RM07	Bit	1FH
RY00	Bit	20H
RY01	Bit	21H
RY02	Bit	22H
RY03	Bit	23H
RY04	Bit	24H
RY05	Bit	25H
RY06	Bit	26H
RY07	Bit	27H
RY10	Bit	28H
RY11	Bit	29H

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

RY12 Bit 2AH

RY13 Bit 2BH

RY14 Bit 2CH

RY15 Bit 2DH

RY16 Bit 2EH

RY17 Bit 2FH

KB_LD Bit 30H

KB_X0 Bit 31H

KB_TM Bit 32H

KB_Dw Bit 33H

KB_7 Bit 34H

KB_4 Bit 35H

KB_1 Bit 36H

KB_0 Bit 37H

KB_OR Bit 38H

KB_Y0 Bit 39H

KB_CT Bit 3AH

KB_Up Bit 3BH

KB_8 Bit 3CH

KB_5 Bit 3DH

KB_2 Bit 3EH

KB_Dot Bit 3FH

KB_AND Bit 40H

KB_M0 Bit 41H

KB_INS Bit 42H

KB_Edit Bit 43H

KB_9 Bit 44H

KB_6 Bit 45H

KB_3 Bit 46H

KB_Stop Bit 47H

KB_Out Bit 48H

KB_Not Bit 49H

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
KB_END          Bit   4AH
KB_Save         Bit   4BH
KB_INC          Bit   4CH
KB_DEC          Bit   4DH
KB_EN           Bit   4EH
KB_Ret          Bit   4FH

Status1         EQU   20H
Status2         EQU   21H
RamX0       EQU   22H
RamM0           EQU   23H
RamY0           EQU   24H
RamY1       EQU   25H
SDKey0          EQU   26H
SDKey1          EQU   27H
SDKey2          EQU   28H
SDKey3      EQU   29H
SDLcd           EQU   2AH
SFunc           EQU   2BH
SData           EQU   2CH

SaveAdd         EQU   30H
SRamAdd         EQU   31H
SAddPro1        EQU   32H
SAddPro2        EQU   33H

SAddPro3        EQU   34H
SAddPro4        EQU   35H
Ram_Hi          EQU   36H
Ram_Lo          EQU   37H
TCount          EQU   38H


        ORG   0000H
        LJMP  RStartPro
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        ORG    000BH
        LJMP   Ser_Disp     ;
        RetI

        ORG    0100H

RStartPro:  MOV    P0,#00H
        MOV    P3,#00H
        CLR    P2.5
        CLR    P2.5
        MOV    P3,#00H
        SETB   P2.5
        SETB   P2.5

        MOV    P2,#10111111H
        MOV    P3,#0FFH
        MOV    P1,#0FFH
        LCALL  Delay
        LCALL  Delay
        LCALL  Ins_LCD
        LCALL  Delay
        MOV    DPTR,#Tb_RStart0
        LCALL  Send_String
        LCALL  Delay_L

        MOV    TMOD,#22H
        MOV    TH0,#06H
        MOV    TL0,#06H
        MOV    IE,#82H
        SETB   TR0
        LCALL  Delay
        LJMP   Test_Key
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
MainPro:
                MOV    DPTR,#Tb_RStart1
                LCALL  Send_String

                LCALL  Delay_L
MKPro00:    JB     KB_Edit,MKPro01
                ;JB    Prog1,MainEdit1
                ;JB    Prog2,MainEdit2
                ;JB    Prog3,MainEdit3
                ;JB    Prog4,MainEdit4
                LJMP   MEditPro
MKPro01:    JB     KB_Stop,MKPro00
                JNB    Prog1,MainRunP1
                ;JNB   Prog2,MainRunP2
                ;JNB   Prog3,MainRunP3
                ;JNB   Prog4,MainRunP4
                MOV    DPTR,#Tb_NoPro
                LCALL  Send_String
                LCALL  Delay_L
                LCALL  Delay_L
                LJMP   MainPro

MainRunP1:  MOV    DPTR,#Tb_ProEr
                LCALL  Send_String
                LCALL  Delay_L
                LCALL  Delay_L
                LJMP   MainPro

MEditPro:
MEditP00:   JB     KB_EN,MKPro03
MKPro03:    JB     KB_Ret,MKPro00
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

KB_9

KB_6

KB_3

KB_8

KB_5

KB_2

KB_7

KB_4

KB_1

KB_0

KB_Dot

KB_LD

KB_AND

KB_OR

KB_TM

KB_CT

KB_X0

KB_Y0

KB_M0

KB_Out

KB_Not

KB_END

KB_INC

KB_DEC

KB_Save

KB_Dw

KB_Up

KB_INS

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        KB_Edit
        KB_Stop

        KB_EN
        KB_Ret


                MOV    SaveAdd,#81H
StartLoad:      LCALL  DispShow
KeyUp:          SETB   P1.6
                JB     P1.6,KeyDw
                JNB    P1.6,$
                NOP
                JNB    P1.6,$
                JNB    P1.6,$
                NOP
                JNB    P1.6,$
                DeC    SaveAdd
                LJMP   StartLoad

KeyDw:          SETB   P1.7
                JB     P1.7,KeyLd
                JNB    P1.7,$
                NOP
                JNB    P1.7,$
                JNB    P1.7,$
                NOP
                JNB    P1.7,$
                INC    SaveAdd

                LJMP   StartLoad

KeyLd:          SETB   P3.0
                JB     P3.0,KeyAd
                JNB    P3.0,$
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                NOP
                JNB    P3.0,$
                JNB    P3.0,$
                NOP
                JNB    P3.0,$
                MOV    R1,SaveAdd
                MOV    A,@R1
                clr    ACC.6
                CLR    ACC.7
                MOV    @R1,A
                LJMP   StartLoad

KeyAd:          SETB   P3.1
                JB     P3.1,KeyOr
                JNB    P3.1,$
                NOP
                JNB    P3.1,$
                JNB    P3.1,$
                NOP
                JNB    P3.1,$
                MOV    R1,SaveAdd
                MOV    A,@R1
                clr    ACC.6
                setb   ACC.7
                MOV    @R1,A
                LJMP   StartLoad

KeyOr:          SETB   P3.2
                JB     P3.2,KeyUt
                JNB    P3.2,$
                NOP
                JNB    P3.2,$
                JNB    P3.2,$
                NOP
                JNB    P3.2,$
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                MOV    R1,SaveAdd
                MOV    A,@R1
                SETB   ACC.6
                CLR    ACC.7
                MOV    @R1,A
                LJMP   StartLoad


KeyUt:          SETB   P3.3
                JB     P3.3,KeyX0
                JNB    P3.3,$
                NOP
                JNB    P3.3,$
                JNB    P3.3,$
                NOP
                JNB    P3.3,$
                MOV    R1,SaveAdd
                MOV    A,@R1
                SETB   ACC.6
                SETB   ACC.7
                MOV    @R1,A
                LJMP   StartLoad

KeyX0:          SETB   P3.4
                JB     P3.4,KeyY0
                JNB    P3.4,$
                NOP
                JNB    P3.4,$
                JNB    P3.4,$
                NOP
                JNB    P3.4,$
                MOV    R1,SaveAdd
                MOV    A,@R1
                CLR    ACC.5
                CLR    ACC.4
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                MOV    @R1,A
                LJMP   StartLoad

KeyY0:          SETB   P3.5
                JB     P3.5,KeyRu
                JNB    P3.5,$
                NOP
                JNB    P3.5,$
                JNB    P3.5,$
                NOP
                JNB    P3.5,$
                MOV    R1,SaveAdd
                MOV    A,@R1
                SETB   ACC.5
                CLR    ACC.4
                MOV    @R1,A
                LJMP   StartLoad

KeyRu:          SETB   P3.6
                JB     P3.6,KeyNt

                JNB    P3.6,$
                NOP
                JNB    P3.6,$
                JNB    P3.6,$
                NOP
                JNB    P3.6,$
                MOV    R1,SaveAdd
                MOV    A,@R1
                MOV    C,Acc.3
                INC    A
                MOV    Acc.3,C
                MOV    @R1,A
                LJMP   StartLoad
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
KeyNt:      SETB   P3.7
            JB     P3.7,KeyEnd
            JNB    P3.7,$
            NOP
            JNB    P3.7,$
            JNB    P3.7,$
            NOP
            JNB    P3.7,$
            MOV    R1,SaveAdd
            MOV    A,@R1
            CPL    Acc.3
            MOV    @R1,A
            LJMP   StartLoad
KeyEnd:     MOV    P2,#0FFH
Nex0:       JB     P2.0,Nex1
            MOV    P0,#00H
            LJMP   StartLoad
Nex1:       JB     P2.1,Nex2
            SETB   P0.1
            LJMP   StartLoad
Nex2:       JB     P2.2,Nex3
            SETB   P0.2
            LJMP   StartLoad
Nex3:       JB     P2.3,Nex4
            SETB   P0.3
            LJMP   StartLoad
Nex4:       JB     P2.4,Nex5
            SETB   P0.4
            LJMP   StartLoad
Nex5:       JB     P2.5,Nex6
            SETB   P0.5
            LJMP   StartLoad
Nex6:       JB     P2.6,Nex7
            SETB   P0.6
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                LJMP    StartLoad
Nex7:           JB      P2.7,Nex8
                SETB    P0.7
Nex8:           LJMP    StartLoad


        ; Function Install Lcd
        ;MSC Set4Bits=>   Bl Rw E Rs D7 D6 D5 D4
        ;                 Rd Gn E Rs D7 D6 D5 D4
        ; LCD Modune  :> Bl Rw CE Rs D7 D6 D5 D4
        ; Now Project !!  XX Rd XX Rs D7 D6 D5 D4 >> CE:P1.5

Ins_LCD:        MOV     P1,#0E3H
                LCALL   Wr_CLK
                LCALL   Delay
                LCALL   Delay
                LCALL   Delay
                LCALL   Delay
                MOV     P1,#0E3H
                LCALL   Wr_CLK
                LCALL   Delay
                LCALL   Delay
                MOV     P1,#0E3H
                LCALL   Wr_CLK
                LCALL   Delay
                LCALL   Delay
                MOV     P1,#0E2H
                LCALL   Wr_CLK
                LCALL   Delay
                LCALL   Delay

                MOV     A,#28H
                LCALL   Wr_Cont
                LCALL   Delay
                MOV     A,#06H
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                LCALL   Wr_Cont
                LCALL   Delay
                MOV     A,#0FH
                LCALL   Wr_Cont
                LCALL   Delay
                MOV     A,#01H
                LCALL   Wr_Cont
                LCALL   Delay
                MOV     A,#02H
                LCALL   Wr_Cont
                LCALL   Delay

            Ret

Wr_Cont:    MOV     SDLcd,#0EEH
            LJMP    Wr_LCD

Wr_Data:    MOV     SDLcd,#0FFH
Wr_LCD:     MOV     R0,#SDLcd
            XCHD    A,@R0
            SWAP    A
            MOV     P1,A
            LCALL   Wr_CLK
            MOV     P1,SDLcd
            LCALL   Wr_CLK
          ; MOV     P1,#0FFH
            MOV     R0,#0FFH
            DJNZ    R0,$
            DJNZ    R0,$
          Ret

Wr_CLK:     SETB    P1.5
            MOV     R0,#0FH
            DJNZ    R0,$
            CLR     P1.5
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                MOV    R0,#0FH
                DJNZ   R0,$
            Ret


Send_String:  LCALL  String_Up
              LCALL  String_Dw
          Ret

String_Up:    MOV    A,#80H
              LCALL  Wr_Cont
              MOV    R7,#14H
Send_St1:     CLR    A
              MOVC   A,@A+DPTR
              LCALL  Wr_Data
              INC    DPTR
              DJNZ   R7,Send_St1
          Ret

String_Dw:    MOV    A,#0C0H
              LCALL  Wr_Cont
              MOV    R7,#14H
Send_St2:     CLR    A
              MOVC   A,@A+DPTR

              LCALL  Wr_Data
              INC    DPTR
              DJNZ   R7,Send_St2
          Ret

    ;**************************************
    ; Serv Timer0 User Reg=> A,R1,R4,R5,R6,R7
    ;

Ser_Disp:     JB     DMode,Mode_RKey
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Mode_RPro:   CLR    P2.7
             NOP
             MOV    P0,RamY0
             MOV    P3,RamY1
             CLR    P2.5
             MOV    P3,RamY1
             SETB   P2.5
             MOV    P3,#0FFH
             CLR    P2.4
             CLR    P2.4
             MOV    RamX0,P3
             MOV    RamX0,P3
             SETB   P2.4
Mode_RKey:   CPL    P2.7
             MOV    P3,#0FFH
             CLR    P2.3
             MOV    SDKey3,P3
             MOV    SDKey3,P3
             SETB   P2.3
             CLR    P2.2
             MOV    SDKey2,P3
             MOV    SDKey2,P3
             SETB   P2.2
             CLR    P2.1
             MOV    SDKey1,P3
             MOV    SDKey1,P3
             SETB   P2.1
             CLR    P2.0
             MOV    SDKey0,P3
             MOV    SDKey0,P3
             SETB   P2.0
          Reti

RunShow:
AShowR:      MOV    A,#0C0H      ;Set DataBits 7 Address 00H
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                LCALL   Wr_Cont
                MOV     R7,#14H

                MOV     DPTR,#Tb_Help
Send_AS1:       MOV     A,TCount
                MOVC    A,@A+DPTR
                LCALL   Wr_Data
                INC     DPTR
                DJNZ    R7,Send_AS1
                INC     TCount
AShowE:         Ret

        ;-------- Function Install Device ----------

Tb_Demo0:       DB      "  024D:- LD  X01-+ "
                DB      " r>025D:- AND X02-+-"
Tb_Func0:       DB      "LD "
Tb_Func1:       DB      "LDI"
Tb_Func2:       DB      "AND"
Tb_Func3:       DB      "ANI"
Tb_Func4:       DB      "OR "
Tb_Func5:       DB      "ORI"
Tb_Func6:       DB      "OUT"
Tb_Free:        DB      "= END == "

        ;               12345678901234567890
Send_SFunc:     MOV     A,#00H
                MOVC    A,@A+DPTR
                LCALL   Wr_Data
                INC     DPTR

                MOV     A,#00H
                MOVC    A,@A+DPTR
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                LCALL   Wr_Data
                INC     DPTR

                MOV     A,#00H
                MOVC    A,@A+DPTR
                LCALL   Wr_Data
                INC     DPTR
        Ret

DispShow:       MOV     SRamAdd,SaveAdd     ; SAdd <-- Save Address
                DEC     SRamAdd
                MOV     A,SRamAdd           ; SRamAdd <-- Save Ram Address
                MOV     A,#80H
                LCALL   Wr_Cont
                MOV     R1,A

                LCALL   TData
                MOV     R1,A
                LCALL   TFunc
                MOV     A,#' '
                LCALL   Wr_Data
                MOV     A,#' '
                LCALL   Wr_Data
                LCALL   DispFmt

                MOV     SRamAdd,SaveAdd     ; SRamAdd <-- Save Address
                MOV     A,#0C0H
                LCALL   Wr_Cont
                MOV     R1,A
                LCALL   TData
                MOV     R1,A
                LCALL   TFunc
                MOV     A,#' '
                LCALL   Wr_Data
                MOV     A,#'>'
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                LCALL   Wr_Data
                LCALL   DispFmt
            Ret

DispFmt:    MOV     A,#' '
                LCALL   Wr_Data
                MOV     A,SRamAdd       ; SRamAdd <-- Save Address
                ANL     A,#7FH
                LCALL   FData
                MOV     A,#'D'
                LCALL   Wr_Data
                MOV     A,#':'
                LCALL   Wr_Data
                MOV     A,#'-'
                LCALL   Wr_Data
                MOV     A,#' '
                LCALL   Wr_Data
                MOV     A,SFunc
SFunc0:     CJNE    A,#00H,SFunc1
                MOV     DPTR,#Tb_Func0
                LCALL   Send_SFunc
                LJMP    NexFunc
SFunc1:     CJNE    A,#01H,SFunc2
                MOV     DPTR,#Tb_Func1
                LCALL   Send_SFunc
                LJMP    NexFunc
SFunc2:     CJNE    A,#02H,SFunc3
                MOV     DPTR,#Tb_Func2

                LCALL   Send_SFunc
                LJMP    NexFunc
SFunc3:     CJNE    A,#03H,SFunc4
                MOV     DPTR,#Tb_Func3
                LCALL   Send_SFunc
                LJMP    NexFunc
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
SFunc4:     CJNE   A,#04H,SFunc5
            MOV    DPTR,#Tb_Func4
            LCALL  Send_SFunc
            LJMP   NexFunc
SFunc5:     CJNE   A,#05H,SFunc6
            MOV    DPTR,#Tb_Func5
            LCALL  Send_SFunc
            LJMP   NexFunc
SFunc6:     CJNE   A,#06H,SFunc7
            MOV    DPTR,#Tb_Func6
            LCALL  Send_SFunc
            LJMP   NexFunc
SFunc7:     MOV    DPTR,#Tb_Free
            LCALL  Send_SFunc
            LCALL  Send_SFunc
            LCALL  Send_SFunc
            LJMP   NFuncE

NexFunc:    MOV    A,SData
            ANL    A,#00110000B
SData0:     CJNE   A,#00000000B,SData1
            MOV    A,#' '
            LCALL  Wr_Data
            MOV    A,#'X'
            LCALL  Wr_Data
            MOV    A,#'0'
            LCALL  Wr_Data
            MOV    A,SData
            ANL    A,#0FH
            add    a,#30H
            LCALL  Wr_Data
            MOV    A,#' '
            LCALL  Wr_Data
            LJMP   NFuncE
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
SData1:     CJNE   A,#00010000B,SData2
            MOV    A,#' '
            LCALL  Wr_Data
            MOV    A,#'M'
            LCALL  Wr_Data
            MOV    A,#'0'

            LCALL  Wr_Data
            MOV    A,SData
            ANL    A,#0FH
            add    a,#30H
            LCALL  Wr_Data
            MOV    A,#' '
            LCALL  Wr_Data
            LJMP   NFuncE

SData2:     ; CJNE   A,#00100000B,SData3
            MOV    A,#' '
            LCALL  Wr_Data
            MOV    A,#'Y'
            LCALL  Wr_Data
            MOV    A,#'0'
            LCALL  Wr_Data
            MOV    A,SData
            ANL    A,#0FH
            add    a,#30H
            LCALL  Wr_Data
            MOV    A,#' '
            LCALL  Wr_Data
            LJMP   NFuncE

SData3:     MOV    A,#' '
            LCALL  Wr_Data
            MOV    A,#'Y'
            LCALL  Wr_Data
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                MOV    A,#'1'
                LCALL  Wr_Data
                MOV    A,SData
                ANL    A,#0FH
                add    a,#30H
                LCALL  Wr_Data
                MOV    A,#' '
                LCALL  Wr_Data
NFuncE:         MOV    A,#'*'
                LCALL  Wr_Data
                MOV    A,#'-'
                LCALL  Wr_Data
                LCALL  Wr_Data
              Ret

TData:          MOV    R1,SRamAdd
                MOV    A,@R1
                ANL    A,#00110111B
                MOV    SData,A

              Ret

TFunc:          MOV    R1,SRamAdd
                MOV    A,@R1
                ANL    A,#11001000B
TFunc0:         CJNE   A,#00000000B,TFunc1
                MOV    SFunc,#00H ; <----- LD
                Ret
TFunc1:         CJNE   A,#00001000B,TFunc2
                MOV    SFunc,#01H ; <----- LDI
                Ret
TFunc2:         CJNE   A,#01000000B,TFunc3
                MOV    SFunc,#02H ; <----- AND
                Ret
TFunc3:         CJNE   A,#01001000B,TFunc4
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                MOV    SFunc,#03H  ; <----- ANI
                Ret
TFunc4:         CJNE   A,#10000000B,TFunc5
                MOV    SFunc,#04H  ; <----- OR
                Ret
TFunc5:         CJNE   A,#10001000B,TFunc6
                MOV    SFunc,#05H  ; <----- ORI
                Ret
TFunc6:         CJNE   A,#11000000B,TFunc7
                MOV    SFunc,#06H  ; <----- OUT
                Ret
TFunc7:         MOV    SFunc,#07H ; <----- Free
                Ret

FData:          MOV    RAm_Hi,#00H
                MOV    RAm_Lo,A
                LCALL  Htod
                MOV    A,Ram_Hi
                ANL    A,#0FH
                ADD    A,#30H
                LCALL  Wr_Data
                MOV    A,Ram_Lo
                SWAP   A
                ANL    A,#0FH
                ADD    A,#30H
                LCALL  Wr_Data
                MOV    A,Ram_Lo
                ANL    A,#0FH
                ADD    A,#30H
                LCALL  Wr_Data
            Ret

;   XXXX -- X -- i -------  XXX ----------- i
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
;   IIII  NOT  000 001 010 011 100 101 110 111
;   IIII   I   I   I   I   I   I   I   I
;   0000 -> LD  X00 X01 X02 X03 X04 X05 X06 X07
;   0001 -> LD  M00 M01 M02 M03 M04 M05 M06 M07
;   0010 -> LD  Y00 Y01 Y02 Y03 Y04 Y05 Y06 Y07
;   0011 -> LD  Y10 Y11 Y12 Y13 Y14 Y15 Y16 Y17
;
;   0100 -> AND X00 X01 X02 X03 X04 X05 X06 X07
;   0101 -> AND M00 M01 M02 M03 M04 M05 M06 M07
;   0110 -> AND Y00 Y01 Y02 Y03 Y04 Y05 Y06 Y07
;   0111 -> AND Y10 Y11 Y12 Y13 Y14 Y15 Y16 Y17
;
;   1000 -> OR  X00 X01 X02 X03 X04 X05 X06 X07
;   1001 -> OR  M00 M01 M02 M03 M04 M05 M06 M07
;   1010 -> OR  Y00 Y01 Y02 Y03 Y04 Y05 Y06 Y07
;   1011 -> OR  Y10 Y11 Y12 Y13 Y14 Y15 Y16 Y17
;
;   1100 ->
;   1101 -> OUT M00 M01 M02 M03 M04 M05 M06 M07
;   1110 -> OUT Y00 Y01 Y02 Y03 Y04 Y05 Y06 Y07
;   1111 -> OUT Y10 Y11 Y12 Y13 Y14 Y15 Y16 Y17


Tb_Demo:      DB    "  028D:- OUT T01 * "
              DB    " > 029D:- SP K052 *-"

Tb_RStart0:   DB    " Programmable Logic "
              DB    " Controller ( PLC ) "

Tb_Help:      DB    " Basic Programmable Logic Controller Simulator V"
              DB    "erstion 1.01, Running System On Microcontroler I"
              DB    "ntRam128K, ExtRam256K, Display LCDModule 20 x 2,"
              DB    " Keyboard MATRIX 5 x 8, Input 8 Ch, Output Relay"
              DB    "Driver 16 Ch, Timer 8 Ch, Counter 8 Ch, SubProMA"
              DB    "X 128 Step.                "
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Delay_L:    LCALL  Delay
            LCALL  Delay
            LCALL  Delay
            LCALL  Delay
          Ret

Delay:      MOV    R6,#00H
Delay_W:    MOV    R5,#00H
            DJNZ   R5,$

            DJNZ   R6,Delay_W
          Ret

;*************************************************************
;*************************************************************
;*  Sub Program Htod: Format Data form Hex to Decimal        *
;*       Input  = DPTR     (Ram_Hi,Ram_Lo)       *
;*       Output = R1,R2,R3  (Ram_Hi,Ram_Lo)      *
;*       Register at Work R0 R1 R2 R3 R4 R5 DPTR A  *
;*************************************************************

Htod:       CLR    A
            CLR    C
            MOV    R1,A
            MOV    R2,A
            MOV    R3,A
            MOV    R4,#16          ;shift 16 bit
            MOV    DPL,Ram_Lo
            MOV    DPH,Ram_Hi
Htod1:      MOV    A,DPL
            RLC    A
            MOV    DPL,A
            MOV    A,DPH
            RLC    A
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                MOV    DPH,A
                MOV    R5,#3          ;add decimal
                MOV    R0,#3          ;index to r3
Htod2:          MOV    A,@R0
                ADDC   A,ACC
                DA     A
                MOV    @R0,A
                DEC    R0
                DJNZ   R5,Htod2
                DJNZ   R4,Htod1
                MOV    Ram_Lo,R3
                MOV    Ram_Hi,R2
                RET

Test_Key:       MOV    A,#80H
                LCALL  Wr_Cont
                MOV    C,KB_LD
                LCALL  Send_K
                MOV    C,KB_X0

                LCALL  Send_K
                MOV    C,KB_TM
                LCALL  Send_K
                MOV    C,KB_Dw
                LCALL  Send_K
                MOV    C,KB_7
                LCALL  Send_K
                MOV    C,KB_4
                LCALL  Send_K
                MOV    C,KB_1
                LCALL  Send_K
                MOV    C,KB_0
                LCALL  Send_K
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
MOV    C,KB_OR
LCALL  Send_K
MOV    C,KB_Y0
LCALL  Send_K
MOV    C,KB_CT
LCALL  Send_K
MOV    C,KB_Up
LCALL  Send_K
MOV    C,KB_8
LCALL  Send_K
MOV    C,KB_5
LCALL  Send_K
MOV    C,KB_2
LCALL  Send_K
MOV    C,KB_Dot
LCALL  Send_K

MOV    A,#0C0H
LCALL  Wr_Cont
MOV    C,KB_AND
LCALL  Send_K
MOV    C,KB_M0
LCALL  Send_K
MOV    C,KB_INS
LCALL  Send_K
MOV    C,KB_Edit
LCALL  Send_K
MOV    C,KB_9
LCALL  Send_K
MOV    C,KB_6
LCALL  Send_K
MOV    C,KB_3
LCALL  Send_K

MOV    C,KB_Stop
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            LCALL   Send_K

            MOV     C,KB_Out
            LCALL   Send_K
            MOV     C,KB_Not
            LCALL   Send_K
            MOV     C,KB_END
            LCALL   Send_K
            MOV     C,KB_Save
            LCALL   Send_K
            MOV     C,KB_INC
            LCALL   Send_K
            MOV     C,KB_DEC
            LCALL   Send_K
            MOV     C,KB_EN
            LCALL   Send_K
            MOV     C,KB_Ret
            LCALL   Send_K
            LCALL   Delay_L
            LJMP    Test_Key
Send_K:     JC      Wr_DHi
            MOV     A,#30H
            LCALL   Wr_Data
        RET
Wr_DHi:     MOV     A,#31H
            LCALL   Wr_Data
        RET


END
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
PortA1          EQU   0FEFCH
PortB1          EQU   0FEFDH
PortC1          EQU   0FEFEH
InsStaus1       EQU   0FEFFH
PortA2          EQU   0FFFCH
PortB2          EQU   0FFFDH
PortC2          EQU   0FFFEH
InsStaus2       EQU   0FFFFH

DLcd.0          Bit   00H
DLcd.1          Bit   01H

DLcd.2          Bit   02H
DLcd.3          Bit   03H
DLcd.4          Bit   04H
DLcd.5          Bit   05H
DLcd.6          Bit   06H
DLcd.7          Bit   07H

DataLCD         EQU   20H
DShow           EQU   21H


        ORG   0000H
        LCALL  Delay_L
        LCALL  Ins_P8255
        LCALL  Ins_LCD
        LCALL  Delay_L

        MOV    DPTR,#Tb_RStart0
        LCALL  Send_String
        LCALL  Delay_L
        LCALL  Delay_L
        LCALL  Delay_L
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            MOV    DShow,#00H
Main:       LCALL  Show_PString
            LCALL  Delay
            LCALL  Delay
            LJMP   Main


Ins_P8255:  MOV    DPTR,#InsStaus1 ; PortA:  Input
            MOV    A,#90H          ; PortB:  Output
            LCALL  SEnd8255        ; PortCh: Output
            LCALL  SEnd8255        ; PortCh: Output
            NOP                    ; PortCl: Output
            NOP

            MOV    DPTR,#InsStaus2 ; PortA:  Output  LCD
            MOV    A,#8AH          ; PortB:  Input   KeyIn
            LCALL  SEnd8255        ; PortCh: Input   KeyIn
            LCALL  SEnd8255        ; PortCh: Input   KeyIn
            NOP                    ; PortCl: Output  KeyCon
            NOP
        Ret

SEnd8255:   MOVX   @DPTR,A

            MOVX   @DPTR,A
            MOVX   @DPTR,A
            MOVX   @DPTR,A
            MOVX   @DPTR,A
        REt

REad8255:   MOVX   A,@DPTR
            MOVX   A,@DPTR
            MOVX   A,@DPTR
            MOVX   A,@DPTR
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            MOVX    A,@DPTR
        REt


; Function Install Lcd
;MSC Set4Bits=>   Bl Rw E Rs D7 D6 D5 D4
;                 Rd Gn E Rs D7 D6 D5 D4

Ins_LCD:    MOV     P1,#01100011B
            Nop
            MOV     P1,#01000011B
            LCALL   Delay
            LCALL   Delay
            LCALL   Delay
            MOV     P1,#01100011B
            Nop
            MOV     P1,#01000011B
            LCALL   Delay
            MOV     P1,#01100011B
            Nop
            MOV     P1,#01000011B
            LCALL   Delay
            MOV     P1,#01100010B
            Nop
            MOV     P1,#01000010B
            LCALL   Delay

            MOV     A,#28H
            ACALL   Wr_Cont
            MOV     A,#06H
            ACALL   Wr_Cont
            MOV     A,#0FH
            ACALL   Wr_Cont
            MOV     A,#01H
            ACALL   Wr_Cont
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        MOV    A,#02H
        ACALL  Wr_Cont

    Ret

Wr_Cont:    SETB   DLcd.7
        CLR    DLcd.4
        LCALL  Wr_LCD
        Nop
    Ret

Wr_Data:    SETB   DLcd.7
        SETB   DLcd.4
        LCALL  Wr_LCD
        Nop
    Ret

Wr_LCD:     MOV    B,A
        SWAP   A
        ANL    A,#0FH       ; CLR ACC Bits Hi
        ADD    A,DataLCD
        MOV    P1,A
        SETB   P1.5
        Nop
        CLR    P1.5
        MOV    A,B
        ANL    A,#0FH       ; CLR ACC Bits Hi
        ADD    A,DataLCD
        MOV    P1,A
        SETB   P1.5
        Nop
        CLR    P1.5
        MOV    R0,#00H
        DJNZ   R0,$
Wr_LE:      Ret
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Send_String:   MOV    A,#80H
               LCALL  Wr_Cont
               MOV    R7,#14H
Send_St1:      CLR    A
               MOVC   A,@A+DPTR
               LCALL  Wr_Data
               INC    DPTR
               DJNZ   R7,Send_St1

               MOV    A,#0C0H
               LCALL  Wr_Cont
               MOV    R7,#14H
Send_St2:      CLR    A
               MOVC   A,@A+DPTR

               LCALL  Wr_Data
               INC    DPTR
               DJNZ   R7,Send_St2
               Ret

Show_PString:  MOV    A,#80H      ;Set DataBits 7 Address 00H
               LCALL  Wr_Cont
               MOV    R7,#14H
               MOV    DPTR,#Tb_Help
Send_PS1:      MOV    A,DShow
               MOVC   A,@A+DPTR
               LCALL  Wr_Data
               INC    DPTR
               DJNZ   R7,Send_PS1

               MOV    A,#0C0H
               LCALL  Wr_Cont
               MOV    R7,#14H
Send_PS2:      MOV    A,DShow
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            MOVC   A,@A+DPTR
            LCALL  Wr_Data
            INC    DPTR
            DJNZ   R7,Send_PS2
            INC    DShow
        Ret

Tb_RStart0:   DB     " Programmable Logic "
            DB     " Controller ( PLC ) "

Tb_About:     DB     "Micro Sign Circuit Co, Ltd. Tel: (662)434-6277-8"
            DB     " it takes .3 sec to hear your own echo? 2. x= V*"
            DB     "t = 343 m/s * .15sec =  51m 2. Whose sound is fa"
            DB     "ster in air, girls or boys? 3. neither- same air"
            DB     " (medium) so same speed 3. Find the period of a "
            DB     " Test. Data End                    "
            ;              xx<--End

Tb_Help:      DB     "Basic PLC Simulator Version 1.0    --> Microcont"
            DB     "roler MCS-51 IntRam 128K ExtRam 256K  --> Displa"
            DB     "y LCD Module 20 x 2  --> Keyboard MATRIX 5 x 8  "
            DB     "--> Input 8 Ch.  --> Output Relay Driver 16 Ch. "
            DB     "--> Timer 2 Ch.  --> Counter 2 Ch.  --> A/D Conv"
            DB     "erter x Ch.                        "

Delay_L:      ACALL  Delay
            ACALL  Delay

            ACALL  Delay
            ACALL  Delay
        Ret

Delay:        MOV    R0,#00H
Delay_W:      MOV    R1,#00H
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        DJNZ    R1,$
        DJNZ    R0,Delay_W
    Ret

END
```

รูปที่ ค.7 โปรแกรมการทำงานเครื่อง PLC

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ง

รายละเอียด และคุณสมบัติของอุปกรณ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Features

- Compatible with MCS-51™ Products
- 4 Kbytes of In-System Reprogrammable Flash Memory
  - Endurance: 1,000 Write/Erase Cycles
- Fully Static Operation: 0 Hz to 24 MHz
- Three-Level Program Memory Lock
- 128 x 8-Bit Internal RAM
- 32 Programmable I/O Lines
- Two 16-Bit Timer/Counters
- Six Interrupt Sources
- Programmable Serial Channel
- Low Power Idle and Power Down Modes

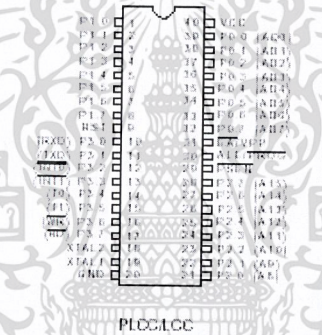
Description

The AT89C51 is a low-power, high-performance CMOS 8-bit microcomputer with 4 Kbytes of Flash Programmable and Erasable Read Only Memory (PEROM). The device is manufactured using Atmel's high density nonvolatile memory technology and is compatible with the industry standard MCS-51™ instruction set and pinout. The on-chip Flash allows the program memory to be reprogrammed in-system or by a conventional nonvolatile memory programmer. By combining a versatile 8-bit CPU with Flash on a monolithic chip, the Atmel AT89C51 is a powerful microcomputer which provides a highly flexible and cost effective solution to many embedded control applications.

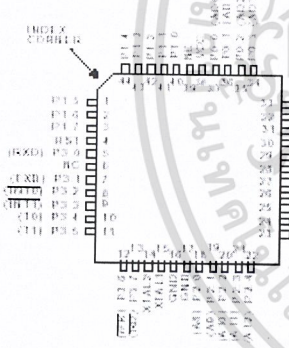
(continued)

Pin Configurations

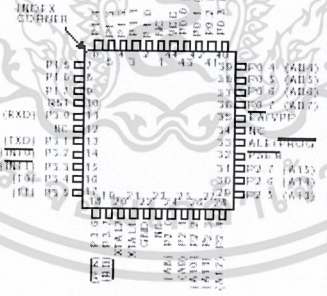
PDIP/Cerdip



PQFP/TQFP



PLCC/LCC



8-Bit Microcontroller with 4 Kbytes Flash

AT89C51

0265E

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Description (Continued)

The AT89C51 provides the following standard features: 4 Kbytes of Flash, 128 bytes of RAM, 32 I/O lines, two 16-bit timer/counters, a five vector two-level interrupt architecture, a full duplex serial port, on-chip oscillator and clock circuitry. In addition, the AT89C51 is designed with static logic for operation down to zero frequency and supports two software selectable power saving modes. The Idle Mode stops the CPU while allowing the RAM, timer/counters, serial port and interrupt system to continue functioning. The Power Down Mode saves the RAM contents but freezes the oscillator disabling all other chip functions until the next hardware reset.

Pin Description

V_{CC}

Supply voltage.

GND

Ground.

Port 0

Port 0 is an 8-bit open drain bidirectional I/O port. As an output port each pin can sink eight TTL inputs. When 1s are written to port 0 pins, the pins can be used as high-impedance inputs.

Port 0 may also be configured to be the multiplexed low-order address/data bus during accesses to external program and data memory. In this mode P0 has internal pullups.

Port 0 also receives the code bytes during Flash programming, and outputs the code bytes during program verification. External pullups are required during program verification.

Port 1

Port 1 is an 8-bit bidirectional I/O port with internal pullups. The Port 1 output buffers can sink/source four TTL inputs. When 1s are written to Port 1 pins they are pulled high by the internal pullups and can be used as inputs. As inputs, Port 1 pins that are externally being pulled low will source current (I_{IL}) because of the internal pullups.

Port 1 also receives the low-order address bytes during Flash programming and program verification.

Port 2

Port 2 is an 8-bit bidirectional I/O port with internal pullups. The Port 2 output buffers can sink/source four TTL inputs. When 1s are written to Port 2 pins they are pulled high by the internal pullups and can be used as inputs. As inputs, Port 2 pins that are externally being pulled low will source current (I_{IL}) because of the internal pullups.

Port 2 emits the high-order address byte during fetches from external program memory and during accesses to external data memory that use 16-bit addresses (MOVX @ DPTR). In this application it uses strong internal pullups when emitting 1s. During accesses to external data memory that use 8-bit addresses (MOVX @ RI), Port 2 emits the contents of the P2 Special Function Register.

Port 2 also receives the high-order address bits and some control signals during Flash programming and verification.

Port 3

Port 3 is an 8-bit bidirectional I/O port with internal pullups. The Port 3 output buffers can sink/source four TTL inputs. When 1s are written to Port 3 pins they are pulled high by the internal pullups and can be used as inputs. As inputs, Port 3 pins that are externally being pulled low will source current (I_{IL}) because of the pullups.

Port 3 also serves the functions of various special features of the AT89C51 as listed below:

Port Pin	Alternate Functions
P3.0	RXD (serial input port)
P3.1	TXD (serial output port)
P3.2	$\overline{INT0}$ (external interrupt 0)
P3.3	$\overline{INT1}$ (external interrupt 1)
P3.4	T0 (timer 0 external input)
P3.5	T1 (timer 1 external input)
P3.6	$\overline{WR}$ (external data memory write strobe)
P3.7	$\overline{RD}$ (external data memory read strobe)

Port 3 also receives some control signals for Flash programming and programming verification.

RST

Reset input. A high on this pin for two machine cycles while the oscillator is running resets the device.

ALE/$\overline{PROG}$

Address Latch Enable output pulse for latching the low byte of the address during accesses to external memory. This pin is also the program pulse input ($\overline{PROG}$) during Flash programming.

In normal operation ALE is emitted at a constant rate of 1/6 the oscillator frequency, and may be used for external timing or clocking purposes. Note, however, that one ALE pulse is skipped during each access to external Data Memory.

If desired, ALE operation can be disabled by setting bit 0 of SFR location 8EH. With the bit set, ALE is active only during a MOVX or MOVC instruction. Otherwise, the pin is weakly pulled high. Setting the ALE-disable bit has no effect if the microcontroller is in external execution mode.

$\overline{PSEN}$

Program Store Enable is the read strobe to external program memory.

(continued)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Pin Description (Continued)

When the AT89C51 is executing code from external program memory, PSEN is activated twice each machine cycle, except that two PSEN activations are skipped during each access to external data memory.

EA/VPP

External Access Enable. EA must be strapped to GND in order to enable the device to fetch code from external program memory locations starting at 0000H up to FFFFH. Note, however, that if lock bit 1 is programmed, EA will be internally latched on reset.

EA should be strapped to VCC for internal program executions.

This pin also receives the 12-volt programming enable voltage (VPP) during Flash programming, for parts that require 12-volt VPP.

XTAL1

Input to the inverting oscillator amplifier and input to the internal clock operating circuit.

XTAL2

Output from the inverting oscillator amplifier.

Oscillator Characteristics

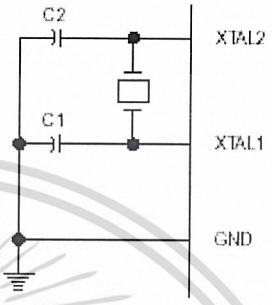
XTAL1 and XTAL2 are the input and output, respectively, of an inverting amplifier which can be configured for use as an on-chip oscillator, as shown in Figure 1. Either a quartz crystal or ceramic resonator may be used. To drive the device from an external clock source, XTAL2 should be left unconnected while XTAL1 is driven as shown in Figure 2. There are no requirements on the duty cycle of the external clock signal, since the input to the internal clocking circuitry is through a divide-by-two flip-flop, but minimum and maximum voltage high and low time specifications must be observed.

Idle Mode

In idle mode, the CPU puts itself to sleep while all the on-chip peripherals remain active. The mode is invoked by software. The content of the on-chip RAM and all the special functions registers remain unchanged during this mode. The idle mode can be terminated by any enabled interrupt or by a hardware reset.

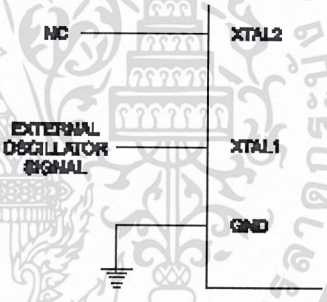
It should be noted that when idle is terminated by a hardware reset, the device normally resumes program execution, from where it left off, up to two machine cycles before the internal reset algorithm takes control. On-chip hard-

Figure 1. Oscillator Connections



Notes: C1, C2 = 30 pF ± 10 pF for Crystals
= 40 pF ± 10 pF for Ceramic Resonators

Figure 2. External Clock Drive Configuration



Status of External Pins During Idle and Power Down

Mode	Program Memory	ALE	PSEN	PORT0	PORT1	PORT2	PORT3
Idle	Internal	1	1	Data	Data	Data	Data
Idle	External	1	1	Float	Data	Address	Data
Power Down	Internal	0	0	Data	Data	Data	Data
Power Down	External	0	0	Float	Data	Data	Data

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ware inhibits access to internal RAM in this event, but access to the port pins is not inhibited. To eliminate the possibility of an unexpected write to a port pin when Idle is terminated by reset, the instruction following the one that invokes Idle should not be one that writes to a port pin or to external memory.

Power Down Mode

In the power down mode the oscillator is stopped, and the instruction that invokes power down is the last instruction executed. The on-chip RAM and Special Function Registers retain their values until the power down mode is terminated. The only exit from power down is a hardware reset. Reset redefines the SFRs but does not change the on-chip RAM. The reset should not be activated before V_{CC} is restored to its normal operating level and must be held active long enough to allow the oscillator to restart and stabilize.

Program Memory Lock Bits

On the chip are three lock bits which can be left unprogrammed (U) or can be programmed (P) to obtain the additional features listed in the table below:

When lock bit 1 is programmed, the logic level at the $\overline{EA}$ pin is sampled and latched during reset. If the device is powered up without a reset, the latch initializes to a random value, and holds that value until reset is activated. It is necessary that the latched value of EA be in agreement with the current logic level at that pin in order for the device to function properly.

Lock Bit Protection Modes

Program Lock Bits				
	LB1	LB2	LB3	Protection Type
1	U	U	U	No program lock features.
2	P	U	U	MOVC instructions executed from external program memory are disabled from fetching code bytes from internal memory, $\overline{EA}$ is sampled and latched on reset, and further programming of the Flash is disabled.
3	P	P	U	Same as mode 2, also verify is disabled.
4	P	P	P	Same as mode 3, also external execution is disabled.

Programming the Flash

The AT89C51 is normally shipped with the on-chip Flash memory array in the erased state (that is, contents = FFH) and ready to be programmed. The programming interface accepts either a high-voltage (12-volt) or a low-voltage (V_{CC}) program enable signal. The low voltage programming mode provides a convenient way to program the AT89C51 inside the user's system, while the high-voltage programming mode is compatible with conventional third party Flash or EPROM programmers.

The AT89C51 is shipped with either the high-voltage or low-voltage programming mode enabled. The respective top-side marking and device signature codes are listed in the following table.

	Vpp = 12 V	Vpp = 5 V
Top-Side Mark	AT89C51 xxxx yyww	AT89C51 xxxx-5 yyww
Signature	(030H)=1EH (031H)=51H (032H)=FFH	(030H)=1EH (031H)=51H (032H)=05H

The AT89C51 code memory array is programmed byte-by-byte in either programming mode. *To program any non-blank byte in the on-chip Flash Memory, the entire memory must be erased using the Chip Erase Mode.*

Programming Algorithm: Before programming the AT89C51, the address, data and control signals should be set up according to the Flash programming mode table and Figures 3 and 4. To program the AT89C51, take the following steps.

1. Input the desired memory location on the address lines.
2. Input the appropriate data byte on the data lines.
3. Activate the correct combination of control signals.
4. Raise $\overline{EA}$/Vpp to 12 V for the high-voltage programming mode.
5. Pulse ALE/$\overline{PROG}$ once to program a byte in the Flash array or the lock bits. The byte-write cycle is self-timed and typically takes no more than 1.5 ms. Repeat steps 1 through 5, changing the address and data for the entire array or until the end of the object file is reached.

$\overline{Data}$ Polling: The AT89C51 features $\overline{Data}$ Polling to indicate the end of a write cycle. During a write cycle, an at-

(continued)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Programming the Flash (Continued)

tempted read of the last byte written will result in the complement of the written datum on PO.7. Once the write cycle has been completed, true data are valid on all outputs, and the next cycle may begin. Data Polling may begin any time after a write cycle has been initiated.

Ready/Busy: The progress of byte programming can also be monitored by the RDY/BSY output signal. P3.4 is pulled low after ALE goes high during programming to indicate BUSY. P3.4 is pulled high again when programming is done to indicate READY.

Program Verify: If lock bits LB1 and LB2 have not been programmed, the programmed code data can be read back via the address and data lines for verification. The lock bits cannot be verified directly. Verification of the lock bits is achieved by observing that their features are enabled.

Chip Erase: The entire Flash array is erased electrically by using the proper combination of control signals and by holding ALE/PROG low for 10 ms. The code array is written with all "1's. The chip erase operation must be executed before the code memory can be re-programmed.

Reading the Signature Bytes: The signature bytes are read by the same procedure as a normal verification of locations 030H,

031H, and 032H, except that P3.6 and P3.7 must be pulled to a logic low. The values returned are as follows.

(030H) = 1EH indicates manufactured by Atmel
(031H) = 51H indicates 89C51
(032H) = FFH indicates 12 V programming
(032H) = 05H indicates 5 V programming

Programming Interface

Every code byte in the Flash array can be written and the entire array can be erased by using the appropriate combination of control signals. The write operation cycle is self-timed and once initiated, will automatically time itself to completion.

All major programming vendors offer worldwide support for the Atmel microcontroller series. Please contact your local programming vendor for the appropriate software revision.

Flash Programming Modes

Mode		RST	PSEN	ALE/PROG	EA/VPP	P2.6	P2.7	P3.6	P3.7
Write Code Data		H	L	⌐‾⌐	H/12V(1)	L	H	H	H
Read Code Data		H	L	H	H	L	L	H	H
Write Lock	Bit - 1	H	L	⌐‾⌐	H/12V	H	H	H	H
	Bit - 2	H	L	⌐‾⌐(2)	H/12V	H	H	L	L
	Bit - 3	H	L	⌐‾⌐	H/12V	H	L	H	L
Chip Erase		H	L	⌐‾⌐	H/12V	H	L	L	L
Read Signature Byte		H	L	H	H	L	L	L	L

Notes: 1. The signature byte at location 032H designates whether V_{PP} = 12 V or V_{PP} = 5 V should be used to enable programming.

2. Chip Erase requires a 10 ms PROG pulse.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Figure 3. Programming the Flash

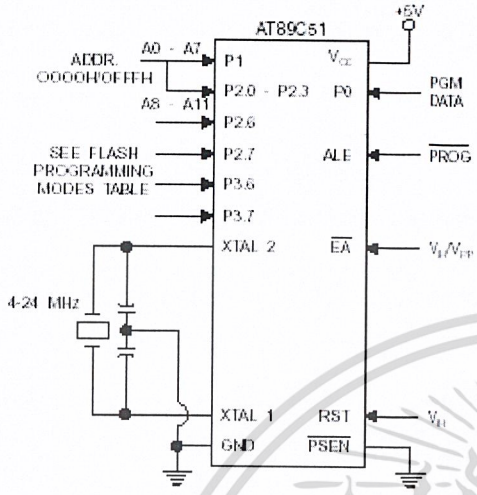
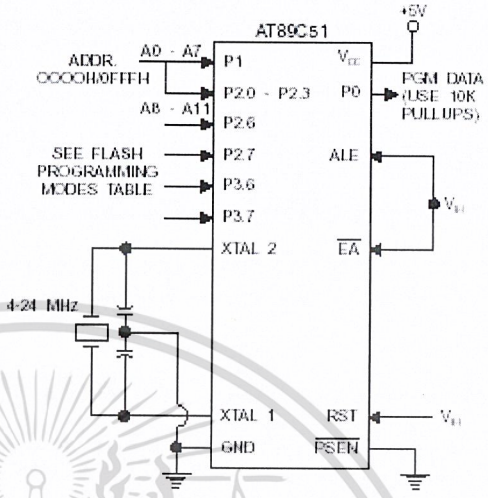


Figure 4. Verifying the Flash



Flash Programming and Verification Characteristics

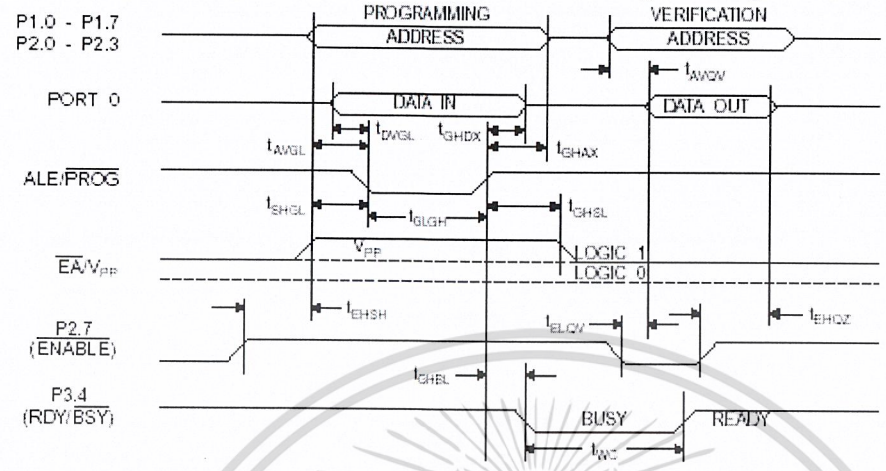
T_A = 21°C to 27°C, V_{CC} = 5.0 ± 10%

Symbol	Parameter	Min	Max	Units
V_{PP}[1]	Programming Enable Voltage	11.5	12.5	V
I_{PP}[1]	Programming Enable Current		1.0	mA
$1/t_{CLCL}$	Oscillator Frequency	4	24	MHz
t_{AVGL}	Address Setup to $\overline{PROG}$ Low	48t_{CLCL}		
t_{GHAX}	Address Hold After $\overline{PROG}$	48t_{CLCL}		
t_{DVGL}	Data Setup to $\overline{PROG}$ Low	48t_{CLCL}		
t_{GHDX}	Data Hold After $\overline{PROG}$	48t_{CLCL}		
t_{EHSH}	P2.7 ($\overline{ENABLE}$) High to V_{PP}	48t_{CLCL}		
t_{SHGL}	V_{PP} Setup to $\overline{PROG}$ Low	10		µs
t_{GHSL}[1]	V_{PP} Hold After $\overline{PROG}$	10		µs
t_{GLGH}	$\overline{PROG}$ Width	1	110	µs
t_{AVQV}	Address to Data Valid		48t_{CLCL}	
t_{ELQV}	$\overline{ENABLE}$ Low to Data Valid		48t_{CLCL}	
t_{EHQV}	Data Float After $\overline{ENABLE}$	0	48t_{CLCL}	
t_{GHBL}	$\overline{PROG}$ High to $\overline{BUSY}$ Low		1.0	µs
t_{WC}	Byte Write Cycle Time		2.0	ms

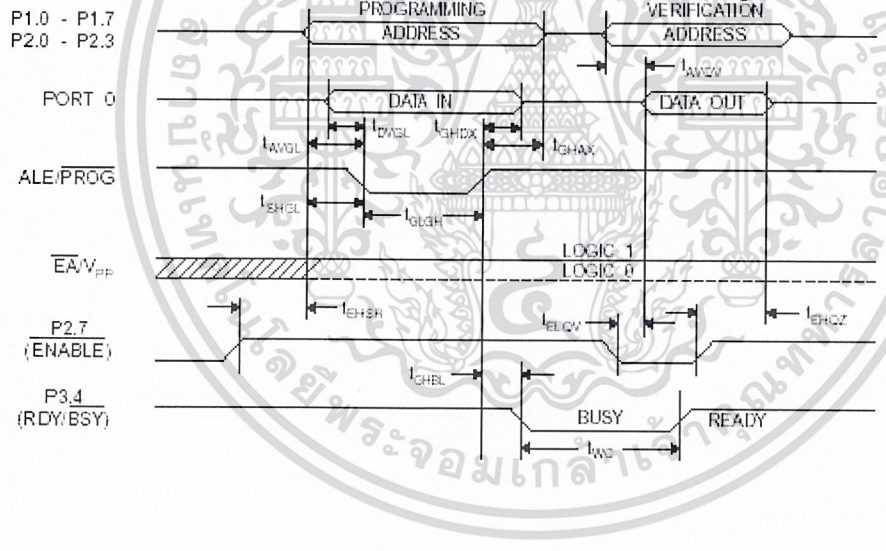
Note: 1. Only used in 12-volt programming mode.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Flash Programming and Verification Waveforms - High Voltage Mode



Flash Programming and Verification Waveforms - Low Voltage Mode



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Absolute Maximum Ratings*

Operating Temperature	-55°C to +125°C
Storage Temperature	-65°C to +150°C
Voltage on Any Pin with Respect to Ground	-1.0 V to +7.0 V
Maximum Operating Voltage	6.6 V
DC Output Current	15.0 mA

*NOTICE: Stresses beyond those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other conditions beyond those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

D.C. Characteristics

T_A = -40°C to 85°C, V_{CC} = 5.0 V ± 20% (unless otherwise noted)

Symbol	Parameter	Condition	Min	Max	Units
V_{IL}	Input Low Voltage	(Except $\overline{EA}$)	-0.5	0.2 V_{CC}-0.1	V
V_{IL1}	Input Low Voltage ($\overline{EA}$)		-0.5	0.2 V_{CC}-0.3	V
V_{IH}	Input High Voltage	(Except XTAL1, RST)	0.2 V_{CC}+0.9	V_{CC}+0.5	V
V_{IH1}	Input High Voltage	(XTAL1, RST)	0.7 V_{CC}	V_{CC}+0.5	V
V_{OL}	Output Low Voltage[1] (Ports 1,2,3)	I_{OL} = 1.6 mA		0.45	V
V_{OL1}	Output Low Voltage[1] (Port 0, ALE, $\overline{PSEN}$)	I_{OL} = 3.2 mA		0.45	V
V_{OH}	Output High Voltage (Ports 1,2,3, ALE, $\overline{PSEN}$)	I_{OH} = -60 µA, V_{CC} = 5 V ± 10%	2.4		V
		I_{OH} = -25µA	0.75 V_{CC}		V
		I_{OH} = -10 µA	0.9 V_{CC}		V
V_{OH1}	Output High Voltage (Port 0 in External Bus Mode)	I_{OH} = -800 µA, V_{CC} = 5 V ± 10%	2.4		V
		I_{OH} = -300 µA	0.75 V_{CC}		V
		I_{OH} = -80 µA	0.9 V_{CC}		V
I_{IL}	Logical 0 Input Current (Ports 1,2,3)	V_{IN} = 0.45 V		-50	µA
I_{TL}	Logical 1 to 0 Transition Current (Ports 1,2,3)	V_{IN} = 2 V		-650	µA
I_{LI}	Input Leakage Current (Port 0, $\overline{EA}$)	0.45 < V_{IN} < V_{CC}		±10	µA
RRST	Reset Pulldown Resistor		50	300	KΩ
C_{IO}	Pin Capacitance	Test Freq. = 1 MHz, T_A = 25°C		10	pF
I_{CC}	Power Supply Current	Active Mode, 12 MHz		20	mA
		Idle Mode, 12 MHz		5	mA
	Power Down Mode[2]	V_{CC} = 6 V		100	µA
		V_{CC} = 3 V		40	µA

Notes: 1. Under steady state (non-transient) conditions, I_{OL} must be externally limited as follows:
Maximum I_{OL} per port pin 10 mA
Maximum I_{OL} per 8-bit port:
Port 0:26 mA
Ports 1,2, 3:15 mA
Maximum total IOL for all output pins:71 mA
If IOL exceeds the test condition, VOL may exceed the related specification. Pins are not guaranteed to sink current greater than the listed test conditions.
2. Minimum VCC for Power Down is 2 V.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

A.C. Characteristics

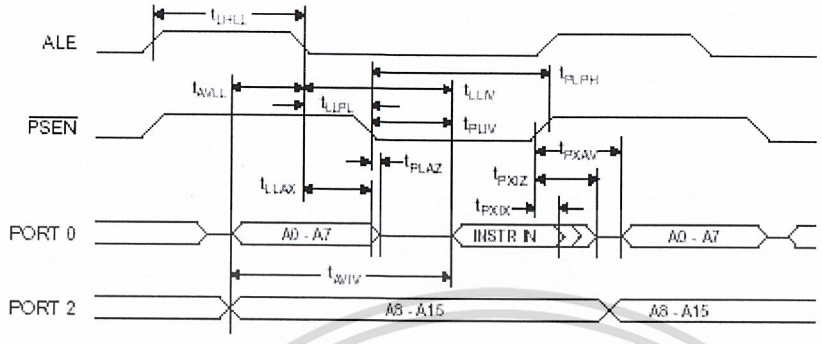
(Under Operating Conditions: Load Capacitance for Port 0, ALE/$\overline{PROG}$, and $\overline{PSEN}$ = 100 pF; Load Capacitance for all other outputs = 80 pF)

External Program and Data Memory Characteristics

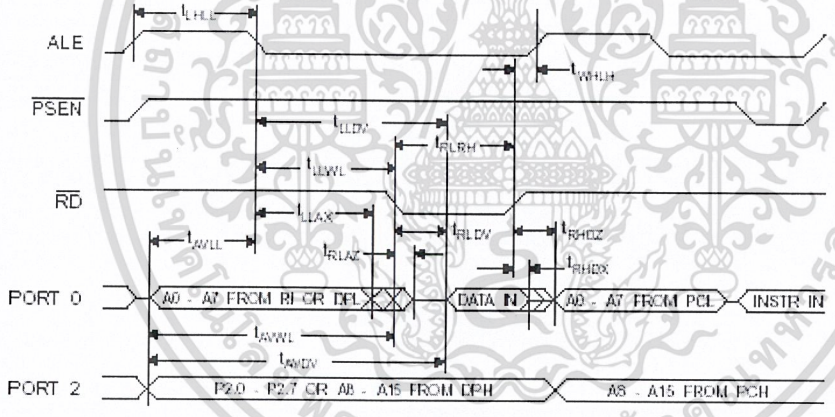
Symbol	Parameter	12 MHz Oscillator Min	12 MHz Oscillator Max	16 to 24 MHz Oscillator Min	16 to 24 MHz Oscillator Max	Units
$1/t_{CLCL}$	Oscillator Frequency			0	24	MHz
t_{LHLL}	ALE Pulse Width	127		$2t_{CLCL}-40$		ns
t_{AVLL}	Address Valid to ALE Low	28		$tCLCL-13$		ns
t_{LLAX}	Address Hold After ALE Low	48		$tCLCL-20$		ns
t_{LLIV}	ALE Low to Valid Instruction In		233		$4tCLCL-65$	ns
t_{LLPL}	ALE Low to $\overline{PSEN}$ Low	43		$t_{CLCL}-13$		ns
t_{PLPH}	$\overline{PSEN}$ Pulse Width	205		$3t_{CLCL}-20$		ns
t_{PLIV}	$\overline{PSEN}$ Low to Valid Instruction In		145		$3t_{CLCL}-45$	ns
t_{PXIX}	Input Instruction Hold After $\overline{PSEN}$	0		0		ns
t_{PXIZ}	Input Instruction Float After $\overline{PSEN}$		59		$t_{CLCL}-10$	ns
t_{PXAV}	$\overline{PSEN}$ to Address Valid	75		$t_{CLCL}-8$		ns
t_{AVIV}	Address to Valid Instruction In		312		$5t_{CLCL}-55$	ns
t_{PLAZ}	$\overline{PSEN}$ Low to Address Float		10		10	ns
t_{RLRH}	$\overline{RD}$ Pulse Width	400		$6t_{CLCL}-100$		ns
t_{WLWH}	$\overline{WR}$ Pulse Width	400		$6t_{CLCL}-100$		ns
t_{RLDV}	$\overline{RD}$ Low to Valid Data In		252		$5t_{CLCL}-90$	ns
t_{RHDX}	Data Hold After $\overline{RD}$	0		0		ns
t_{RHDZ}	Data Float After $\overline{RD}$		97		$2t_{CLCL}-28$	ns
t_{LLDV}	ALE Low to Valid Data In		517		$8tCLCL-150$	ns
t_{AVDV}	Address to Valid Data In		585		$9tCLCL-165$	ns
t_{LLWL}	ALE Low to $\overline{RD}$ or $\overline{WR}$ Low	200	300	$3t_{CLCL}-50$	$3t_{CLCL}+50$	ns
t_{AVWL}	Address to $\overline{RD}$ or $\overline{WR}$ Low	203		$4t_{CLCL}-75$		ns
t_{QVWX}	Data Valid to $\overline{WR}$ Transition	23		$t_{CLCL}-20$		ns
t_{QVWH}	Data Valid to $\overline{WR}$ High	433		$7t_{CLCL}-120$		ns
t_{WHQX}	Data Hold After $\overline{WR}$	33		$t_{CLCL}-20$		ns
t_{RLAZ}	$\overline{RD}$ Low to Address Float		0		0	ns
t_{WHLH}	$\overline{RD}$ or $\overline{WR}$ High to ALE High	43	123	$t_{CLCL}-20$	$t_{CLCL}+25$	ns

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

External Program Memory Read Cycle

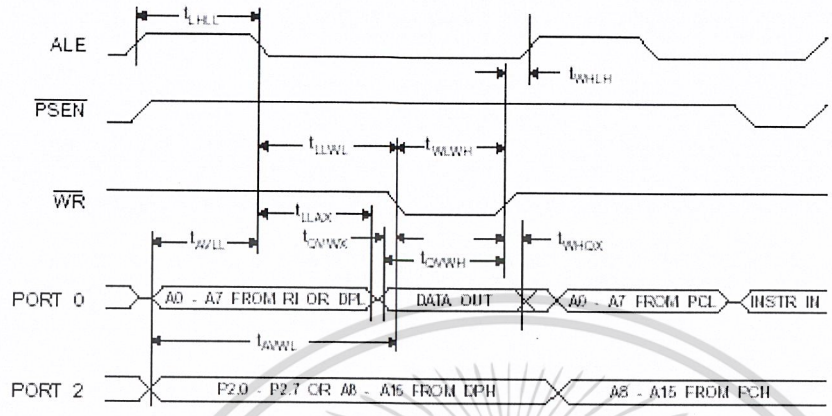


External Data Memory Read Cycle

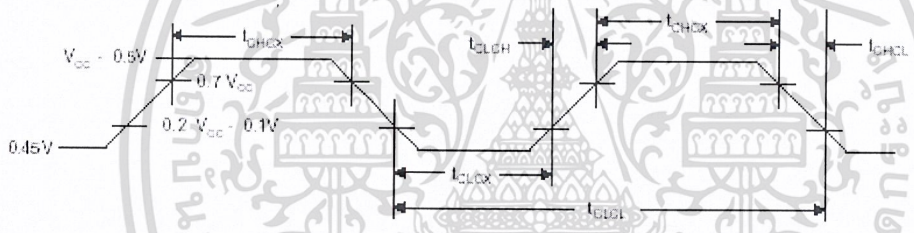


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

External Data Memory Cycle



External Clock Drive Waveforms



External Clock Drive

Symbol	Parameter	Min	Max	Units
1/tCLCL	Oscillator Frequency	0	24	MHz
tCLCL	Clock Period	41.6		ns
tCHCX	High Time	15		ns
tCLCX	Low Time	15		ns
tCLCH	Rise Time		20	ns
tCHCL	Fall Time		20	ns

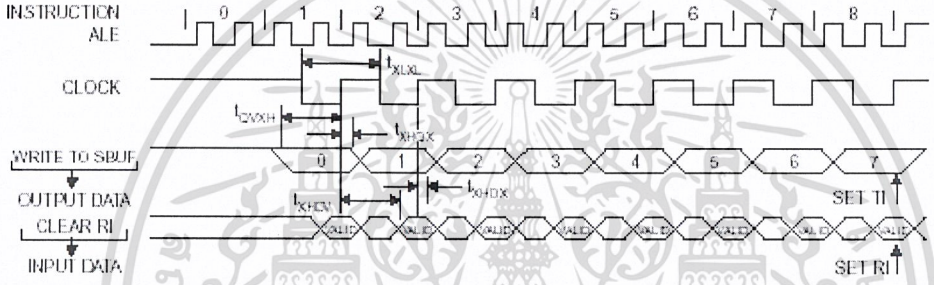
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Serial Port Timing: Shift Register Mode Test Conditions

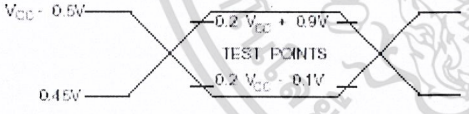
(V_{CC} = 5.0 V ± 20%; Load Capacitance = 80 pF)

Symbol	Parameter	12 MHz Osc		Variable Oscillator		Units
		Min	Max	Min	Max	
t_{XLXL}	Serial Port Clock Cycle Time	1.0		$12t_{CLCL}$		µs
t_{QVXH}	Output Data Setup to Clock Rising Edge	700		$10t_{CLCL}$-133		ns
t_{XHQX}	Output Data Hold After Clock Rising Edge	50		$2t_{CLCL}$-33		ns
t_{XHDX}	Input Data Hold After Clock Rising Edge	0		0		ns
t_{XHDV}	Clock Rising Edge to Input Data Valid		700		$10t_{CLCL}$-133	ns

Shift Register Mode Timing Waveforms

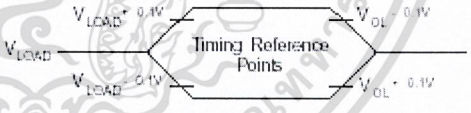


AC Testing Input/Output Waveforms[1]



Note: 1. AC Inputs during testing are driven at V_{CC} - 0.5 V for a logic 1 and 0.45 V for a logic 0. Timing measurements are made at V_{IH} min. for a logic 1 and V_{IL} max. for a logic 0.

Float Waveforms[1]



Note: 1. For timing purposes, a port pin is no longer floating when a 100 mV change from load voltage occurs. A port pin begins to float when a 100 mV change from the loaded V_{OH}/V_{OL} level occurs.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Ordering Information

Speed (MHz)	Power Supply	Ordering Code	Package	Operation Range
12	5 V ± 20%	AT89C51-12AC AT89C51-12JC AT89C51-12PC AT89C51-12QC	44A 44J 40P6 44Q	Commercial (0°C to 70°C)
		AT89C51-12AI AT89C51-12JI AT89C51-12PI AT89C51-12QI	44A 44J 40P6 44Q	Industrial (-40°C to 85°C)
		AT89C51-12AA AT89C51-12JA AT89C51-12PA AT89C51-12QA	44A 44J 40P6 44Q	Automotive (-40°C to 125°C)
	5 V ± 10%	AT89C51-12DM AT89C51-12LM	40D6 44L	Military (-55°C to 125°C)
		AT89C51-12DM/883 AT89C51-12LM/883	40D6 44L	Military/883C Class B, Fully Compliant (-55°C to 125°C)
16	5 V ± 20%	AT89C51-16AC AT89C51-16JC AT89C51-16PC AT89C51-16QC	44A 44J 40P6 44Q	Commercial (0°C to 70°C)
		AT89C51-16AI AT89C51-16JI AT89C51-16PI AT89C51-16QI	44A 44J 40P6 44Q	Industrial (-40°C to 85°C)
		AT89C51-16AA AT89C51-16JA AT89C51-16PA AT89C51-16QA	44A 44J 40P6 44Q	Automotive (-40°C to 125°C)
20	5 V ± 20%	AT89C51-20AC AT89C51-20JC AT89C51-20PC AT89C51-20QC	44A 44J 40P6 44Q	Commercial (0°C to 70°C)
		AT89C51-20AI AT89C51-20JI AT89C51-20PI AT89C51-20QI	44A 44J 40P6 44Q	Industrial (-40°C to 85°C)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Ordering Information

Speed (MHz)	Power Supply	Ordering Code	Package	Operation Range
24	5 V ± 20%	AT89C51-24AC AT89C51-24JC AT89C51-24PC AT89C51-24QC	44A 44J 44P6 44Q	Commercial (0°C to 70°C)
		AT89C51-24AI AT89C51-24JI AT89C51-24PI AT89C51-24QI	44A 44J 44P6 44Q	Industrial (-40°C to 85°C)

Package Type	
44A	44 Lead, Thin Plastic Gull Wing Quad Flatpack (TQFP)
40D6	40 Lead, 0.600" Wide, Non-Windowed, Ceramic Dual Inline Package (Cerdip)
44J	44 Lead, Plastic J-Leaded Chip Carrier (PLCC)
44L	44 Pad, Non-Windowed, Ceramic Leadless Chip Carrier (LCC)
40P6	40 Lead, 0.600" Wide, Plastic Dual Inline Package (PDIP)
44Q	44 Lead, Plastic Gull Wing Quad Flatpack (PQFP)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

intel

8255A-5
PROGRAMMABLE PERIPHERAL INTERFACE

- MCS-85 Compatible 8255A-5
- 24 Programmable I/O Pins
- Completely TTL Compatible
- Fully Compatible with Intel Microprocessor Families
- Improved Timing Characteristics
- Direct Bit Set/Reset Capability Easing Control Application Interface
- Reduces System Package Count
- Improved DC Driving Capability
- Available in EXPRESS
  - —Standard Temperature Range
  - —Extended Temperature Range
- 40 Pin DIP Package

(See Intel Packaging Order Number: 231369-001, Package Type P)

The Intel 8255A is a general purpose programmable I/O device designed for use with Intel microprocessors. It has 24 I/O pins which may be individually programmed in 2 groups of 12 and used in 3 major modes of operation. In the first mode (MODE 0), each group of 12 I/O pins may be programmed in sets of 4 to be input or output. In MODE 1, the second mode, each group may be programmed to have 8 lines of input or output. Of the remaining 4 pins, 3 are used for handshaking and interrupt control signals. The third mode of operation (MODE 2) is a bidirectional bus mode which uses 8 lines for a bidirectional bus, and 5 lines, borrowing one from the other group, for handshaking.

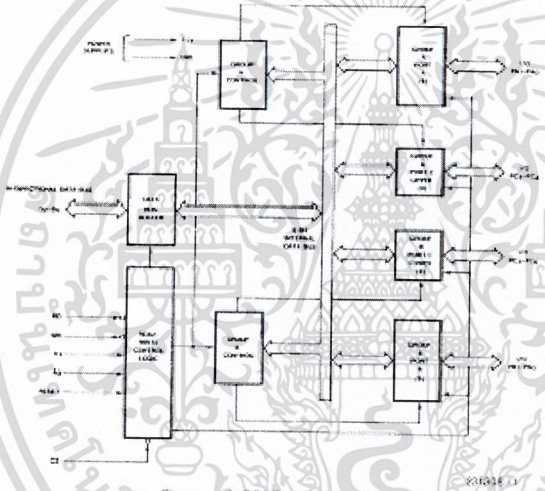


Figure 1. 8255A Block Diagram

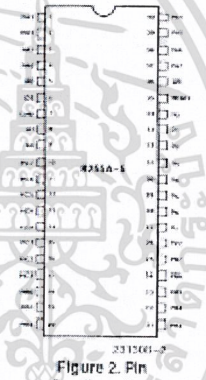


Figure 2. Pin Configuration

Intel Corporation assumes no responsibility for the use of any circuitry other than circuitry embodied in an Intel product. No other circuit patent licenses are implied. Information contained herein supersedes previously published specifications on these devices from Intel.
© INTEL CORPORATION, 1980

September 1993
Order Number: 231309-004

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

8255A FUNCTIONAL DESCRIPTION

General

The 8255A is a programmable peripheral interface (PPI) device designed for use in Intel microcomputer systems. Its function is that of a general purpose I/O component to interface peripheral equipment to the microcomputer system bus. The functional configuration of the 8255A is programmed by the system software so that normally no external logic is necessary to interface peripheral devices or structures.

Data Bus Buffer

This 3-state bidirectional 8-bit buffer is used to interface the 8255A to the system data bus. Data is transmitted or received by the buffer upon execution of input or output instructions by the CPU. Control words and status information are also transferred through the data bus buffer.

Read/Write and Control Logic

The function of this block is to manage all of the internal and external transfers of both Data and Control or Status words. It accepts inputs from the CPU Address and Control busses and in turn, issues commands to both of the Control Groups.

($\overline{CS}$)

Chip Select. A "low" on this input pin enables the communication between the 8255A and the CPU.

($\overline{RD}$)

Read. A "low" on this input pin enables the 8255A to send the data or status information to the CPU on the data bus. In essence, it allows the CPU to "read from" the 8255A.

($\overline{WR}$)

Write. A "low" on this input pin enables the CPU to write data or control words into the 8255A.

(A_0 and A_1)

Port Select 0 and Port Select 1. These input signals, in conjunction with the RD and WR inputs, control the selection of one of the three ports or the control word registers. They are normally connected to the least significant bits of the address bus (A_0 and A_1).

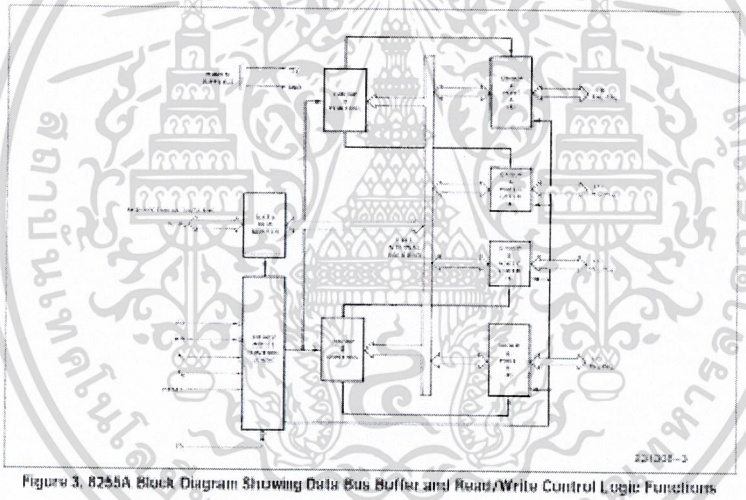


Figure 3. 8255A Block Diagram Showing Data Bus Buffer and Read/Write Control Logic Functions

© INTEL CORPORATION, 1980

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

8255A BASIC OPERATION

A_1	A_0	$\overline{RD}$	$\overline{WR}$	$\overline{CS}$	Input Operation (READ)
0	0	0	1	0	Port A → Data Bus
0	1	0	1	0	Port B → Data Bus
1	0	0	1	0	Port C → Data Bus
					Output Operation (WRITE)
0	0	1	0	0	Data Bus → Port A
0	1	1	0	0	Data Bus → Port B
1	0	1	0	0	Data Bus → Port C
1	1	1	0	0	Data Bus → Control
					Disable Function
X	X	X	X	1	Data Bus → 3-State
1	1	0	1	0	Illegal Condition
X	X	1	1	0	Data Bus → 3-State

(RESET)

Reset. A "high" on this input clears the control register and all ports (A, B, C) are set to the input mode.

Group A and Group B Controls

The functional configuration of each port is programmed by the systems software. In essence, the CPU "outputs" a control word to the 8255A. The control word contains information such as "mode", "bit set", "bit reset", etc. that initializes the functional configuration of the 8255A.

Each of the Control blocks (Group A and Group B) accepts "commands" from the Read/Write Control Logic, receives "control words" from the internal data bus and issues the proper commands to its associated ports.

Control Group A—Port A and Port C upper (C7 - C4)

Control Group B—Port B and Port C lower (C3 - C0)

The Control Word Register can **Only** be written into. No Read operation of the Control Word Register is allowed.

Ports A, B, and C

The 8255A contains three 8-bit ports (A, B, and C). All can be configured in a wide variety of functional characteristics by the system software but each has its own special features or "personality" to further enhance the power and flexibility of the 8255A.

Port A. One 8-bit data output latch/buffer and one 8-bit data input latch.

Port B. One 8-bit data input/output latch/buffer and one 8-bit data input buffer.

Port C. One 8-bit data output latch/buffer and one 8-bit data input buffer (no latch for input). This port can be divided into two 4-bit ports under the mode control. Each 4-bit port contains a 4-bit latch and it can be used for the control signal outputs and status signal inputs in conjunction with ports A and B.

© INTEL CORPORATION, 1980

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

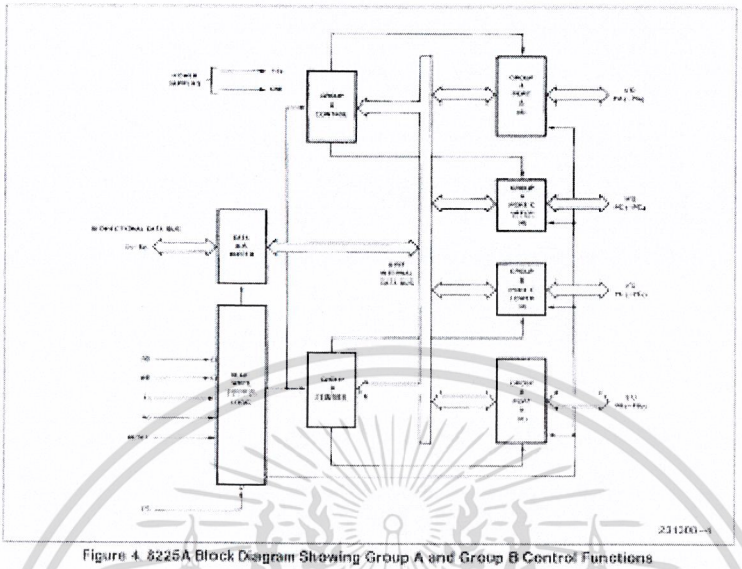
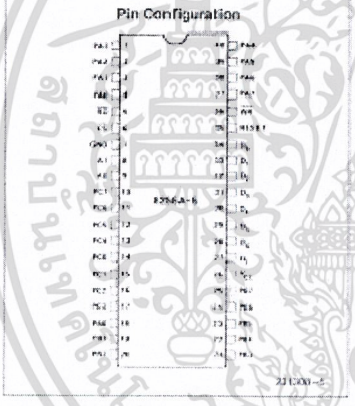


Figure 4. 8225A Block Diagram Showing Group A and Group B Control Functions

Pin Configuration



Pin Names	
D_7–D_0	Data Bus (Bi-Directional)
RESET	Reset Input
CS	Chip Select
RD	Read Input
WR	Write Input
A0, A1	Port Address
PA7–PA0	Port A (BIT)
PB7–PB0	Port B (BIT)
PC7–PC0	Port C (BIT)
V_{CC}	+5 Volts
GND	0 Volts

8255A OPERATIONAL DESCRIPTION

Mode Selection

There are three basic modes of operation that can be selected by the system software:

© INTEL CORPORATION, 1980

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Mode 0—Basic Input/Output

Mode 1—Strobed Input/Output

Mode 2—Bi-Directional Bus

When the reset input goes "high" all ports will be set to the input mode (i.e., all 24 lines will be in the high impedance state). After the reset is removed the 8255A can remain in the input mode with no additional initialization required. During the execution of the system program any of the other modes may be selected using a single output instruction. This allows a single 8255A to service a variety of peripheral devices with a simple software maintenance routine.

The modes for Port A and Port B can be separately defined, while Port C is divided into two portions as required by the Port A and Port B definitions. All of the output registers, including the status flip-flops, will be reset whenever the mode is changed. Modes may be combined so that their functional definition can be "tailored" to almost any I/O structure. For instance; Group B can be programmed in Mode 0 to monitor simple switch closings or display computational results, Group A could be programmed in Mode 1 to monitor a keyboard or tape reader on an interrupt-driven basis.

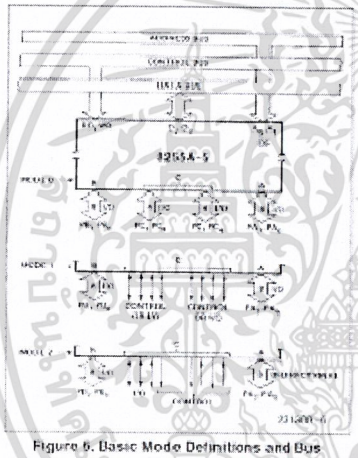


Figure 6. Basic Mode Definitions and Bus Interface

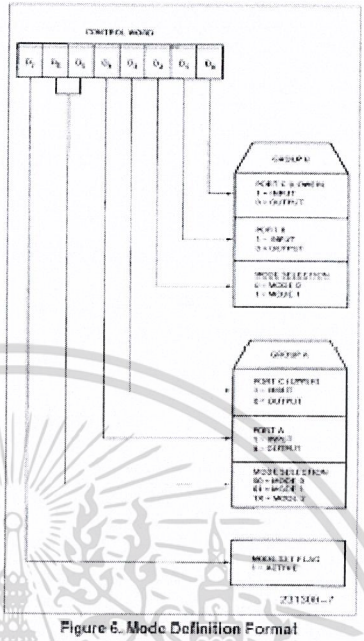


Figure 6. Mode Definition Format

The mode definitions and possible mode combinations may seem confusing at first but after a cursory review of the complete device operation a simple, logical I/O approach will surface. The design of the 8255A has taken into account things such as efficient PC board layout, control signal definition vs PC layout and complete functional flexibility to support almost any peripheral device with no external logic. Such design represents the maximum use of the available pins.

Single Bit Set/Reset Feature

Any of the eight bits of Port C can be Set or Reset using a single OUTput instruction. This feature reduces software requirements in Control-based applications.

© INTEL CORPORATION, 1980

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

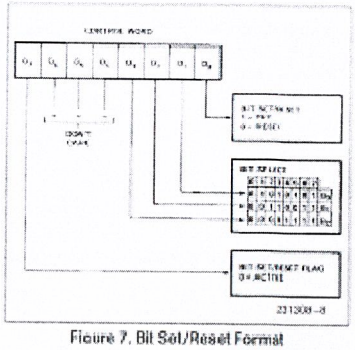


Figure 7. Bit Set/Reset Format

When Port C is being used as status/control for Port A or B, these bits can be set or reset by using the Bit Set/Reset operation just as if they were data output ports.

Interrupt Control Functions

When the 8255A is programmed to operate in mode 1 or mode 2, control signals are provided that can be used as interrupt request inputs to the CPU. The interrupt request signals, generated from port C, can be inhibited or enabled by setting or resetting the associated INTE flip-flop, using the bit set/reset function of port C.

This function allows the Programmer to disallow or allow a specific I/O device to interrupt the CPU without affecting any other device in the interrupt structure.

INTE flip-flop definition:

(BIT-SET)—INTE is set—Interrupt enable

(BIT-RESET)—INTE is RESET—Interrupt disable

NOTE:
All Mask flip-flops are automatically reset during mode selection and device Reset.

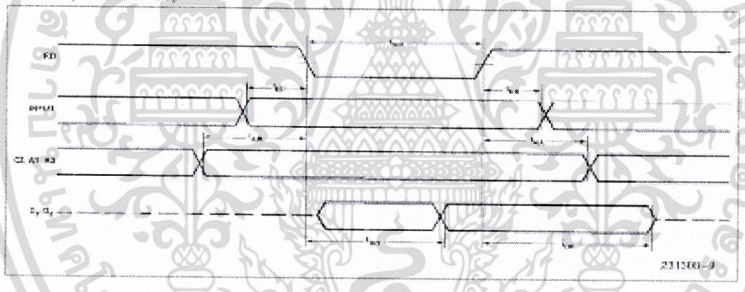
Operating Modes

MODE 0 (Basic Input/Output). This functional configuration provides simple input and output operations for each of the three ports. No "handshaking" is required, data is simply written to or read from a specified port.

Mode 0 Basic Functional Definitions:

- Two 8-bit ports and two 4-bit ports.
- Any port can be input or output.
- Outputs are latched.
- Inputs are not latched.
- 16 different Input/Output configurations are possible in this Mode.

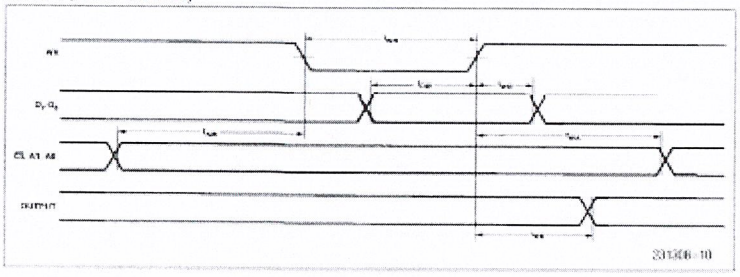
MODE 0 (BASIC INPUT)



© INTEL CORPORATION, 1980

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

MODE 0 (BASIC OUTPUT)



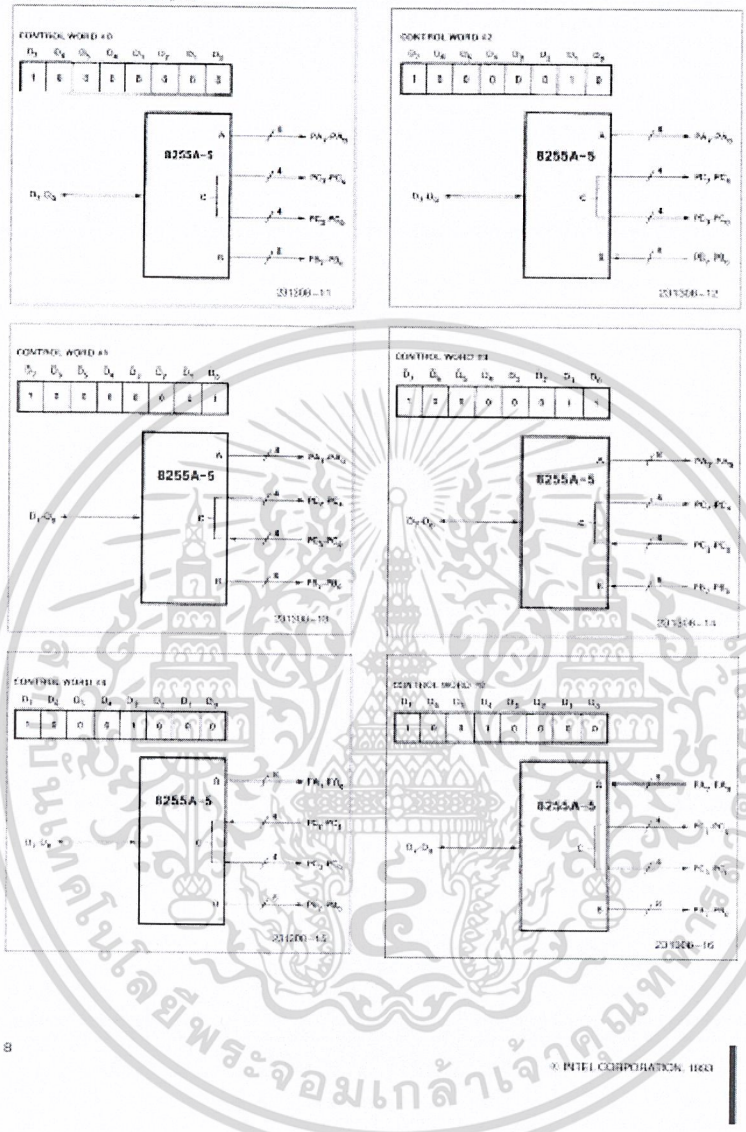
MODE 0 PORT DEFINITION

A		B		Group A			Group B	
D_4	D_3	D_1	D_0	Port A	Port C (Upper)	#	Port B	Port C (Lower)
0	0	0	0	OUTPUT	OUTPUT	0	OUTPUT	OUTPUT
0	0	0	1	OUTPUT	OUTPUT	1	OUTPUT	INPUT
0	0	1	0	OUTPUT	OUTPUT	2	INPUT	OUTPUT
0	0	1	1	OUTPUT	OUTPUT	3	INPUT	INPUT
0	1	0	0	OUTPUT	INPUT	4	OUTPUT	OUTPUT
0	1	0	1	OUTPUT	INPUT	5	OUTPUT	INPUT
0	1	1	0	OUTPUT	INPUT	6	INPUT	OUTPUT
0	1	1	1	OUTPUT	INPUT	7	INPUT	INPUT
1	0	0	0	INPUT	OUTPUT	8	OUTPUT	OUTPUT
1	0	0	1	INPUT	OUTPUT	9	OUTPUT	INPUT
1	0	1	0	INPUT	OUTPUT	10	INPUT	OUTPUT
1	0	1	1	INPUT	OUTPUT	11	INPUT	INPUT
1	1	0	0	INPUT	INPUT	12	OUTPUT	OUTPUT
1	1	0	1	INPUT	INPUT	13	OUTPUT	INPUT
1	1	1	0	INPUT	INPUT	14	INPUT	OUTPUT
1	1	1	1	INPUT	INPUT	15	INPUT	INPUT

© INTEL CORPORATION, 1980

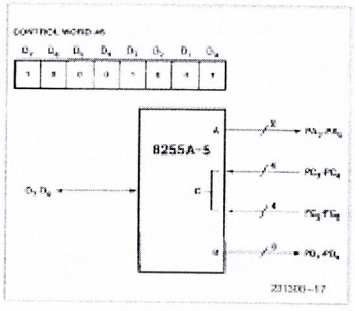
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

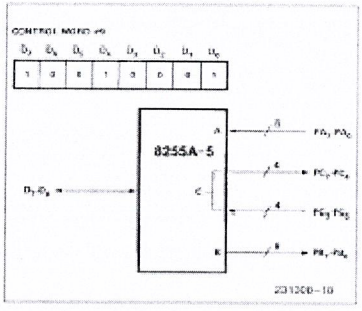
MODE CONFIGURATIONS

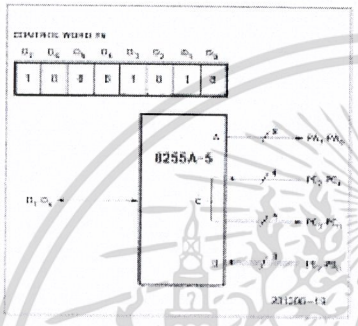


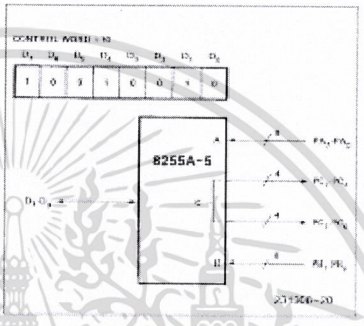
© INTEL CORPORATION, 1980

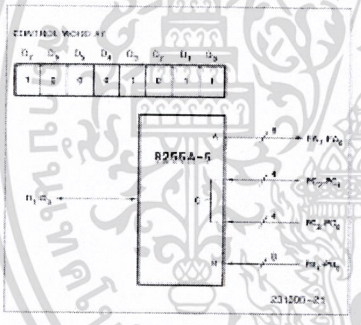
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

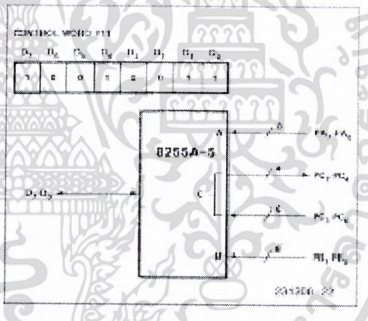






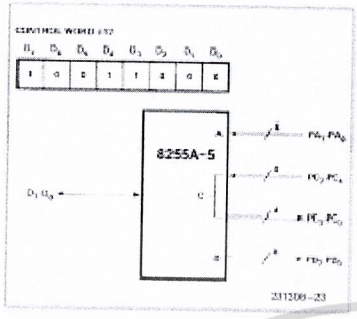


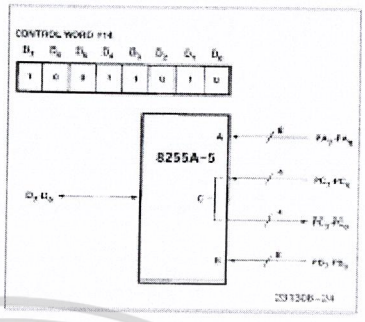


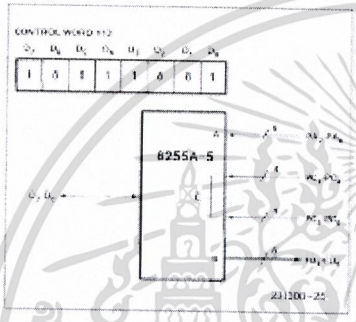


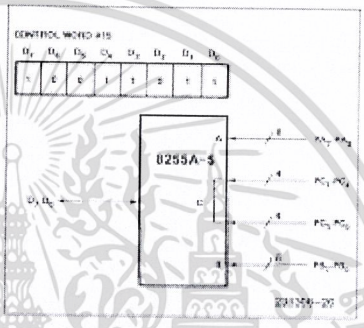
© INTEL CORPORATION, 1980

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้









Operating Modes

MODE 1 (Strobed Input/Output). This functional configuration provides a means for transferring I/O data to or from a specified port in conjunction with strobes or "handshaking" signals. In mode 1, port A and port B use the lines on port C to generate or accept these "handshaking" signals.

Mode 1 Basic Functional Definitions:

- Two Groups (Group A and Group B)
- Each group contains one 8-bit data port and one 4-bit control/data port.
- The 8-bit data port can be either input or output. Both inputs and outputs are latched.
- The 4-bit port is used for control and status of the 8-bit data port.

Input Control Signal Definition

STB (Strobe Input). A "low" on this input loads data into the input latch.

IBF (Input Buffer Full F/F)

A "high" on this output indicates that the data has been loaded into the input latch; in essence, an acknowledgement. IBF is set by STB input being low and is reset by the rising edge of the RD input.

INTR (Interrupt Request)

A "high" on this output can be used to interrupt the CPU when an input device is requesting service. INTR is set by the STB is a "one", IBF is a "one" and INTE is a "one". It is reset by the falling edge of RD. This procedure allows an input device to request service from the CPU by simply strobing its data into the port.

© INTEL CORPORATION, 1980

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

INTE A

Controlled by bit set/reset of PC_4.

INTE B

Controlled by bit set/reset of PC_2.

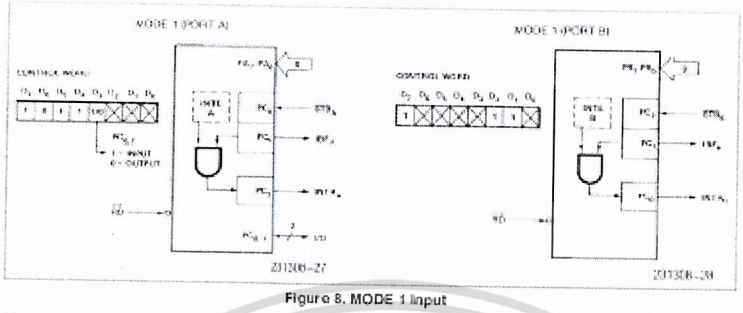


Figure 8. MODE 1 Input

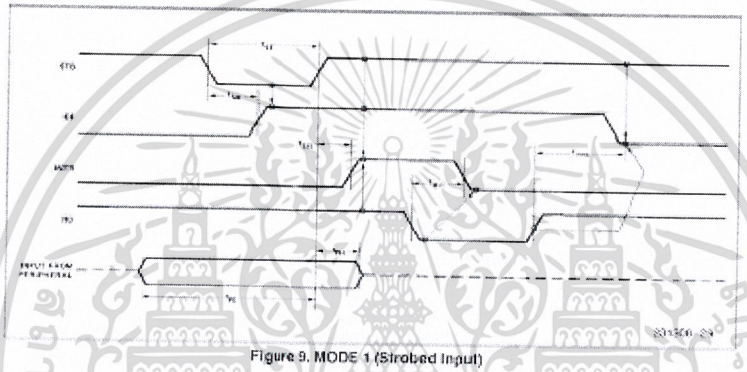


Figure 9. MODE 1 (Strobed Input)

© INTEL CORPORATION, 1980

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Output Control Signal Definition

$\overline{OBF}$ (Output Buffer Full F/F). The $\overline{OBF}$ output will go "low" to indicate that the CPU has written data out to the specified port. The $\overline{OBF}$ F/F will be set by the rising edge of the $\overline{WR}$ input and reset by $\overline{ACK}$ input being low.

$\overline{ACK}$ (Acknowledge Input). A "low" on this input informs the 8255A that the data from port A or port B has been accepted. In essence, a response from the peripheral device indicating that it has received the data output by the CPU.

INTR (Interrupt Request). A "high" on this output can be used to interrupt the CPU when an output device has accepted data transmitted by the CPU. INTR is set when $\overline{ACK}$ is a "one", $\overline{OBF}$ is a "one", and INTE is a "one". It is reset by the falling edge of $\overline{WR}$.

INTE A

Controlled by bit set/reset of PC_6.

INTE B

Controlled by bit set/reset of PC_2.

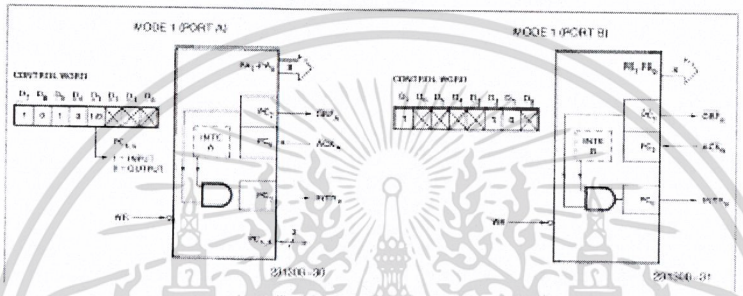


Figure 10. MODE 1 Output

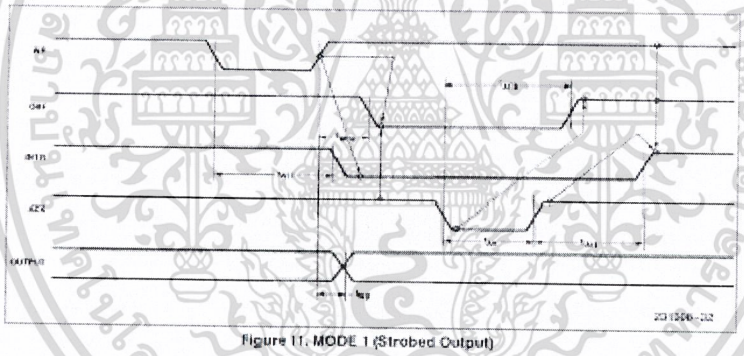


Figure 11. MODE 1 (Strobed Output)

© INTEL CORPORATION, 1980

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

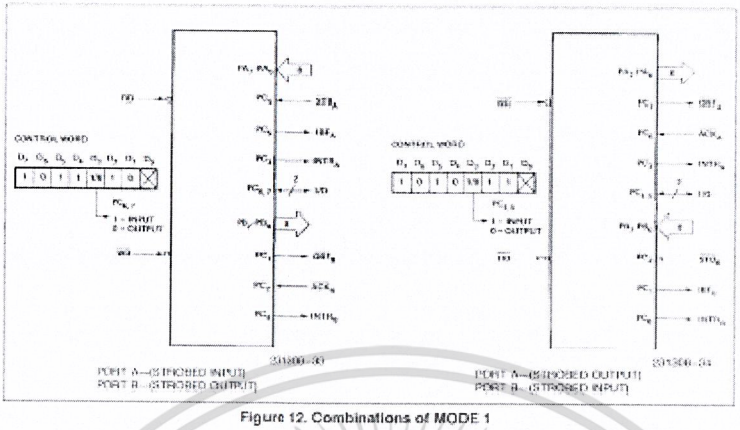


Figure 12. Combinations of MODE 1

Combinations of MODE 1

Port A and Port B can be individually defined as input or output in MODE 1 to support a wide variety of strobed I/O applications.

Operating Modes

MODE 2 (Strobed Bidirectional Bus I/O). This functional configuration provides a means for communicating with a peripheral device or structure on a single 8-bit bus for both transmitting and receiving data (bidirectional bus I/O). "Handshaking" signals are provided to maintain proper bus flow discipline in a similar manner to MODE 1. Interrupt generation and enable/disable functions are also available.

MODE 2 Basic Functional Definitions:

- Used in Group A only.
- One 8-bit, bi-directional bus Port (Port A) and a 5-bit control Port (Port C).
- Both inputs and outputs are latched.
- The 5-bit control port (Port C) is used for control and status for the 8-bit, bi-directional bus port (Port A).

Bidirectional Bus I/O Control Signal Definition

INTR (Interrupt Request). A high on this output can be used to interrupt the CPU for both input or output operations.

Output Operations

OBF (Output Buffer Full). The OBF output will go "low" to indicate that the CPU has written data out to port A.

ACK (Acknowledge). A "low" on this input enables the tri-state output buffer of port A to send out the data. Otherwise, the output buffer will be in the high impedance state.

INTE 1 (The INTE Flip-Flop Associated with OBF). Controlled by bit set/reset of PC_6.

Input Operations

STB (Strobe Input). A "low" on this input loads data into the input latch.

© INTEL CORPORATION, 1980

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

IBF (Input Buffer Full F/F). A "high" on this output indicates that data has been loaded into the input latch.

INTE 2 (The INTE Flip-Flop Associated with IBF). Controlled by bit set/reset of PC_2.

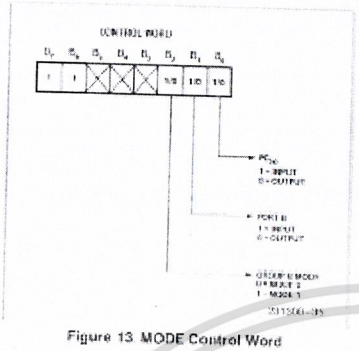


Figure 13. MODE Control Word

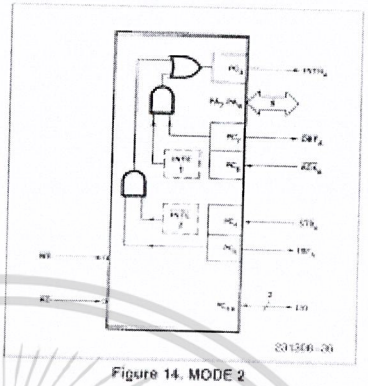
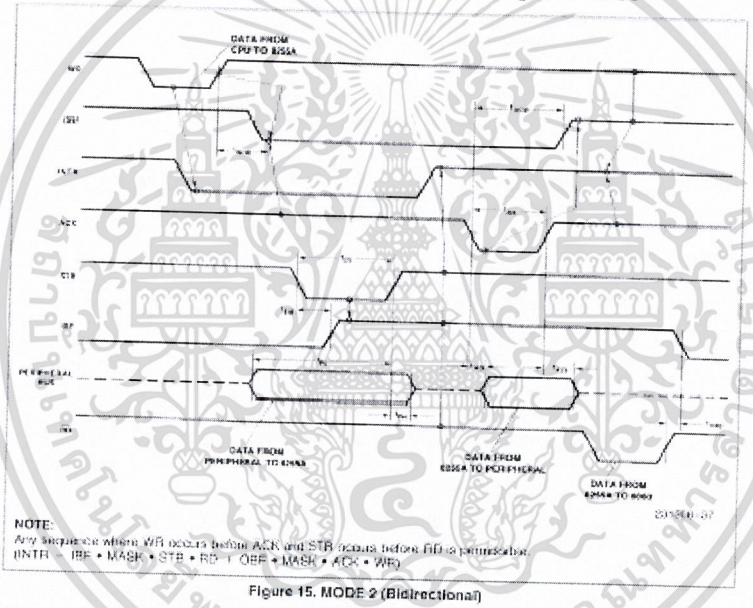


Figure 14. MODE 2



NOTE:
Any sequence where WR occurs before ACK and STB occurs before RD is permissible.
(INTR = IBF • MASK • STB • RD + OBF • MASK • ACK • WR)

Figure 15. MODE 2 (Bidirectional)

© INTEL CORPORATION, 1980

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

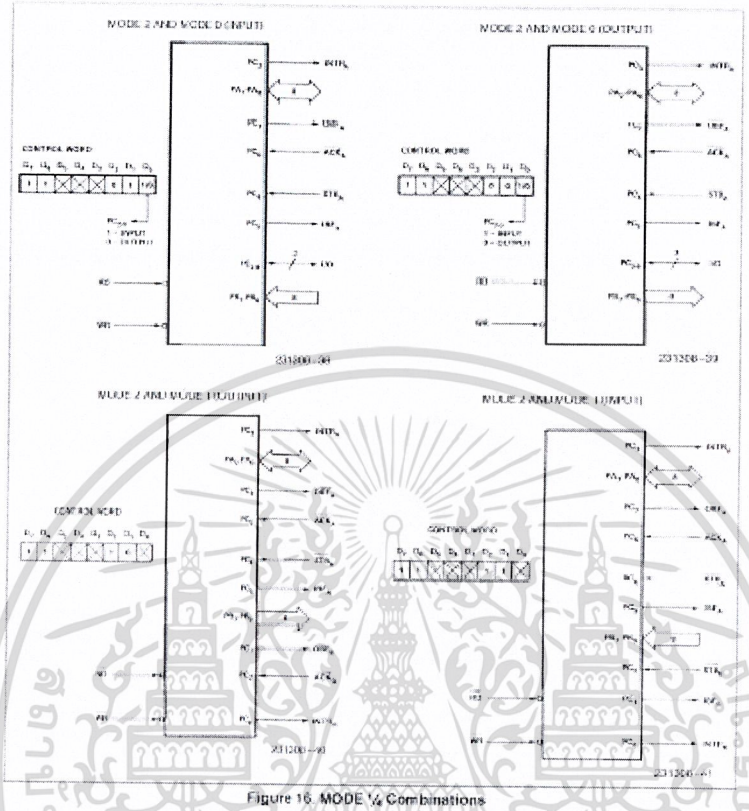


Figure 16. MODE 2 Combinations

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

intel

Mode Definition Summary

	MODE 0		MODE 1		MODE 2	
	IN	OUT	IN	OUT	GROUP A ONLY	
PA_0	IN	OUT	IN	OUT	⟷	
PA_1	IN	OUT	IN	OUT	⟷	
PA_2	IN	OUT	IN	OUT	⟷	
PA_3	IN	OUT	IN	OUT	⟷	
PA_4	IN	OUT	IN	OUT	⟷	
PA_5	IN	OUT	IN	OUT	⟷	
PA_6	IN	OUT	IN	OUT	⟷	
PA_7	IN	OUT	IN	OUT	⟷	
PB_0	IN	OUT	IN	OUT	—	MODE 0 OR MODE 1 ONLY
PB_1	IN	OUT	IN	OUT	—	
PB_2	IN	OUT	IN	OUT	—	
PB_3	IN	OUT	IN	OUT	—	
PB_4	IN	OUT	IN	OUT	—	
PB_5	IN	OUT	IN	OUT	—	
PB_6	IN	OUT	IN	OUT	—	
PB_7	IN	OUT	IN	OUT	—	
PC_0	IN	OUT	$INTR_B$	$INTR_B$	I/O	
PC_1	IN	OUT	IBF_B	OBF_B	I/O	
PC_2	IN	OUT	STB_B	ACK_B	I/O	
PC_3	IN	OUT	$INTR_A$	$INTR_A$	$INTR_A$	
PC_4	IN	OUT	STB_A	I/O	STB_A	
PC_5	IN	OUT	IBF_A	I/O	IBF_A	
PC_6	IN	OUT	I/O	ACK_A	ACK_A	
PC_7	IN	OUT	I/O	OBF_A	OBF_A	

Special Mode Combination Considerations

There are several combinations of modes when not all of the bits in Port C are used for control or status. The remaining bits can be used as follows:

If Programmed as Inputs—

All input lines can be accessed during a normal Port C read.

If Programmed as Outputs—

Bits in C upper (PC_7–PC_4) must be individually accessed using the bit set/reset function.

Bits in C lower (PC_3–PC_0) can be accessed using the bit set/reset function or accessed as a group by writing into Port C.

Source Current Capability on Port B and Port C

Any set of eight output buffers, selected randomly from Ports B and C can source 1 mA at 1.5 volts. This feature allows the 8255 to directly drive Darlington type drivers and high-voltage displays that require such source current.

Reading Port C Status

In Mode 0, Port C transfers data to or from the peripheral device. When the 8255 is programmed to function in Modes 1 or 2, Port C generates or accepts "hand-shaking" signals with the peripheral device. Reading the contents of Port C allows the programmer to test or verify the "status" of each peripheral device and change the program flow accordingly.

There is no special instruction to read the status information from Port C. A normal read operation of Port C is executed to perform this function.

© INTEL CORPORATION, 1980

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

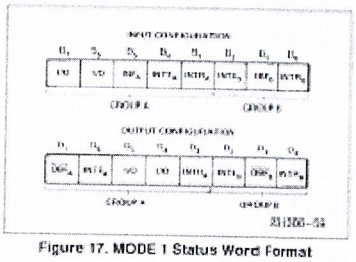


Figure 17. MODE 1 Status Word Format

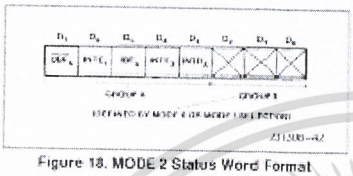


Figure 18. MODE 2 Status Word Format

APPLICATIONS OF THE 8255A

The 8255A is a very powerful tool for interfacing peripheral equipment to the microcomputer system. It represents the optimum use of available pins and is flexible enough to interface almost any I/O device without the need for additional external logic.

Each peripheral device in a microcomputer system usually has a "service routine" associated with it. The routine manages the software interface between the device and the CPU. The functional definition of the 8255A is programmed by the I/O service routine and becomes an extension of the system software. By examining the I/O devices interface characteristics for both data transfer and timing, and matching this information to the examples and tables in the detailed operational description, a control word can easily be developed to initialize the 8255A to exactly "fit" the application. Figures 19 through 25 represent a few examples of typical applications of the 8255A.

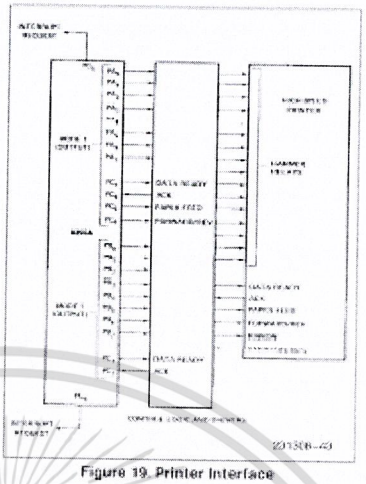


Figure 19. Printer Interface

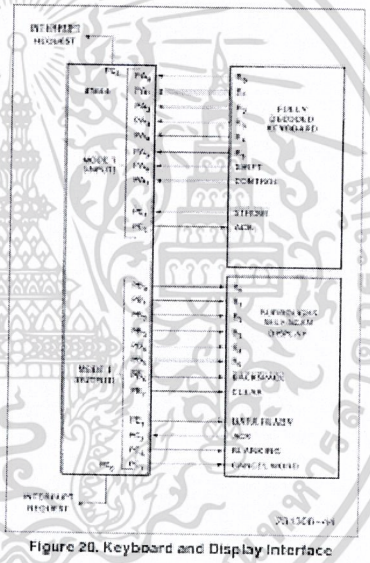


Figure 20. Keyboard and Display Interface

© INTEL CORPORATION, 1980

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

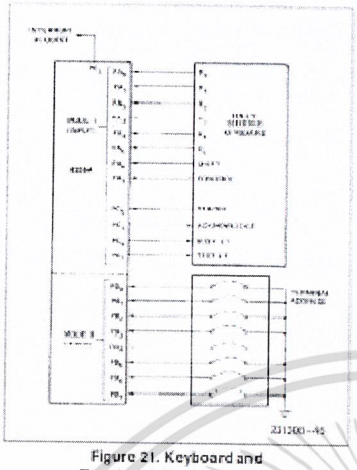


Figure 21. Keyboard and Terminal Address Interface

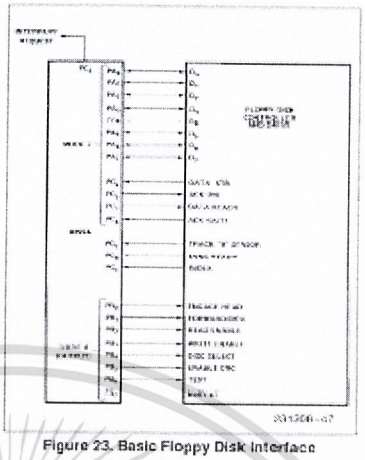


Figure 23. Basic Floppy Disk Interface

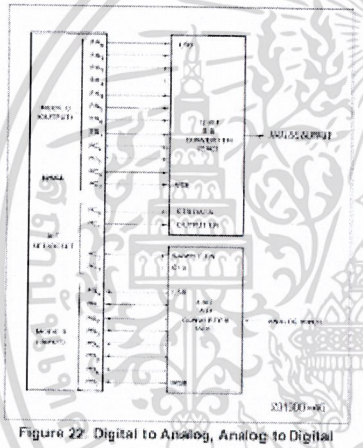


Figure 22. Digital to Analog, Analog to Digital

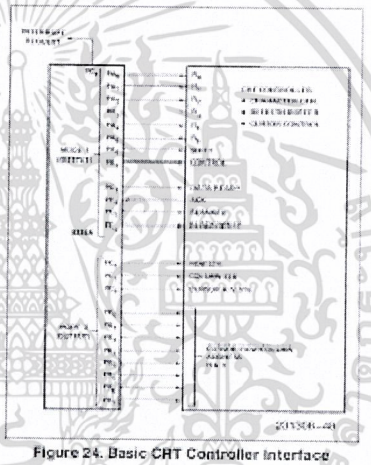


Figure 24. Basic CRT Controller Interface

© INTEL CORPORATION, 1980

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

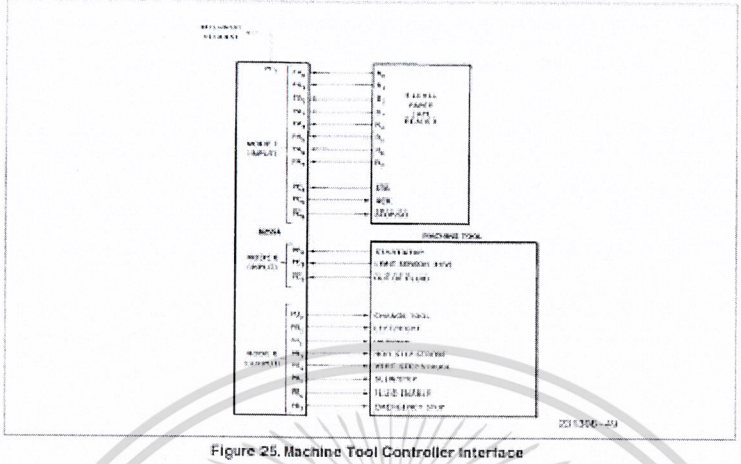


Figure 25. Machine Tool Controller Interface

ABSOLUTE MAXIMUM RATINGS*

Ambient Temperature Under Bias . . . . 0°C to 70°C
Storage Temperature . . . . . −65°C to +150°C
Voltage on Any Pin
with Respect to Ground . . . . . −0.5V to +7V
Power Dissipation . . . . . 1 Watt

NOTICE: This is a production data sheet. The specifications are subject to change without notice.

*WARNING: Stressing the device beyond the "Absolute Maximum Ratings" may cause permanent damage. These are stress ratings only. Operation beyond the "Operating Conditions" is not recommended and extended exposure beyond the "Operating Conditions" may affect device reliability.

D.C. CHARACTERISTICS T_A = 0°C to 70°C, V_{CC} = +5V ± 10%, GND = 0V*

Symbol	Parameter	Min	Max	Unit	Test Conditions
V_{IL}	Input Low Voltage	−0.5	0.8	V	
V_{IH}	Input High Voltage	2.0	V_{CC}	V	
V_{OL} (DB)	Output Low Voltage (Data Bus)		0.45*	V	I_{OL} = 2.5 mA
V_{OL} (PCn)	Output Low Voltage (Peripheral Port)		0.45*	V	I_{OL} = 1.7 mA
V_{OH} (DB)	Output High Voltage (Data Bus)	2.4		V	I_{OH} = −400 μA
V_{OH} (PER)	Output High Voltage (Peripheral Port)	2.4		V	I_{OH} = −200 μA
I_{DAR}(1)	Darlington Drive Current	−1.0	−4.0	mA	R_{EXT} = 750Ω; V_{EXT} = 1.5V
I_{CC}	Power Supply Current		120	mA	
I_{IL}	Input Load Current		±10	μA	V_{IN} = V_{CC} to 0V
I_{OFL}	Output Float Leakage		±10	μA	V_{OUT} = V_{CC} to 0.45V

NOTE:
1. Available on any 8 pins from Port B and C.

* INTEL CORPORATION, 1980

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

intel

CAPACITANCE T_A = 25°C, V_{CC} = GND = 0V

Symbol	Parameter	Min	Typ	Max	Unit	Test Conditions
C_{IN}	Input Capacitance			10	pF	f_c = 1 MHz[4]
$C_{I/O}$	I/O Capacitance			20	pF	Unmeasured pins returned to GND[4]

A.C. CHARACTERISTICS T_A = 0°C to 70°C, V_{CC} = +5V ±10%, GND = 0V*

Bus Parameters

READ

Symbol	Parameter	8255A-5		Unit
		Min	Max	
t_{AR}	Address Stable before READ	0		ns
t_{RA}	Address Stable after READ	0		ns
t_{RR}	READ Pulse Width	300		ns
t_{RD}	Data Valid from READ[1]		200	ns
t_{DF}	Data Float after READ	10	100	ns
t_{RV}	Time between READs and/or WRITEs	850		ns

WRITE

Symbol	Parameter	8255A-5		Unit
		Min	Max	
t_{AW}	Address Stable before WRITE	0		ns
t_{WA}	Address Stable after WRITE	20		ns
t_{WW}	WRITE Pulse Width	300		ns
t_{DW}	Data Valid to WRITE (T.E.)	100		ns
t_{WD}	Data Valid after WRITE	30		ns

OTHER TIMINGS

Symbol	Parameter	8255A-5		Unit	
		Min	Max	Min	Max
t_{WB}	WR = 1 to Output[1]		350	ns	
t_{IR}	Peripheral Data before RD	0		ns	
t_{HR}	Peripheral Data after RD	0		ns	
t_{AK}	ACK Pulse Width	300		ns	
t_{ST}	STB Pulse Width	500		ns	
t_{PS}	Per. Data before T.E. of STB	0		ns	
t_{PH}	Per. Data after T.E. of STB	180		ns	
t_{AD}	ACK = 0 to Output[1]		300	ns	
t_{KD}	ACK = 1 to Output Float	20	250	ns	

© INTEL CORPORATION, 1980

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

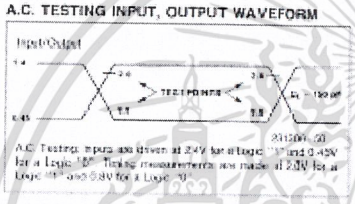
A.C. CHARACTERISTICS (Continued)

OTHER TIMINGS (Continued)

Symbol	Parameter	8255A-5		Unit
		Min	Max	
t_{WOB}	WR = 1 to OBF = 0(1)		650	ns
t_{AOB}	ACK = 0 to OBF = 1(1)		350	ns
t_{SIB}	STB = 0 to IBF = 1(1)		300	ns
t_{RIB}	RD = 1 to IBF = 0(1)		300	ns
t_{RIT}	RD = 0 to INTR = 0(1)		400	ns
t_{SIT}	STB = 1 to INTR = 1(1)		300	ns
t_{AIT}	ACK = 1 to INTR = 1(1)		350	ns
t_{WIT}	WR = 0 to INTR = 0(1, 3)		850	ns

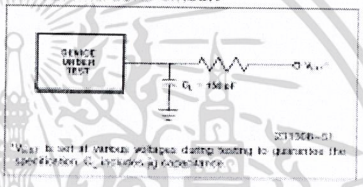
NOTES:
1. Test Conditions: C_L = 150 pF.
2. Period of Reset pulse must be at least 50 μs during or after power on. Subsequent Reset pulse can be 500 ns min.
3. INTR ↑ may occur as early as WR ↓.
4. Sampled, not 100% tested.
*For Extended Temperature EXPRESS, use M8255A electrical parameters.

A.C. TESTING INPUT, OUTPUT WAVEFORM



A.C. Testing: Inputs are driven at 2.4V for a Logic "1" and 0.45V for a Logic "0". Timing measurements are made at 2.0V for a Logic "1" and 0.8V for a Logic "0".

A.C. TESTING LOAD CIRCUIT



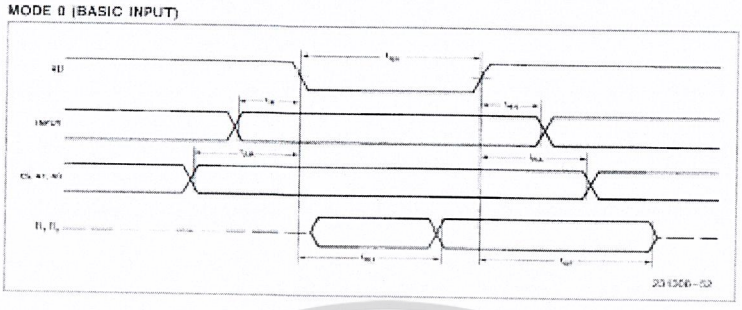
*V_{OH} is set at various voltages during testing to guarantee the specification of I_{OH} including leakage.

© INTEL CORPORATION, 1980

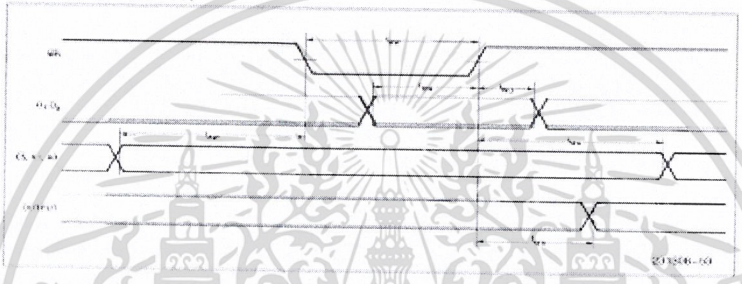
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

WAVEFORMS

MODE 0 (BASIC INPUT)



MODE 0 (BASIC OUTPUT)

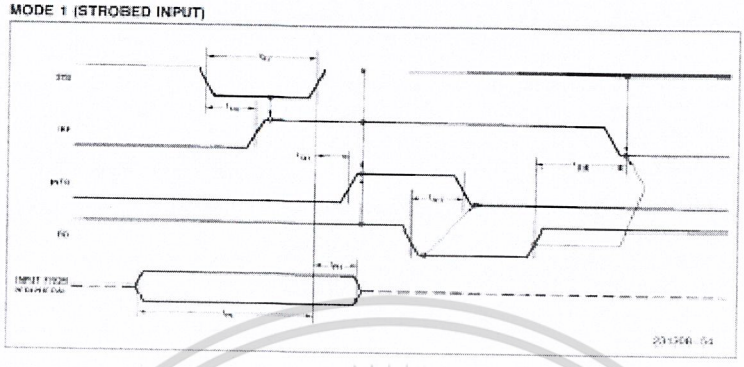


© INTEL CORPORATION, 1980

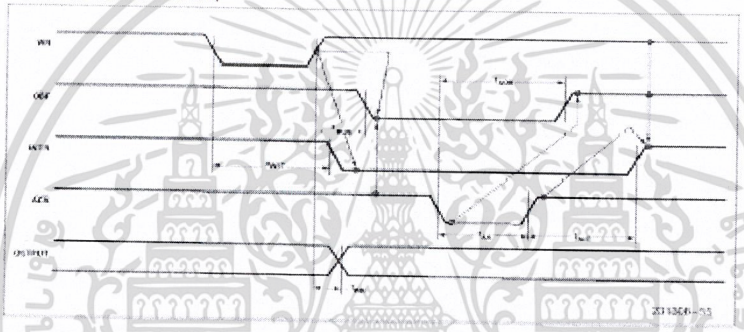
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

WAVEFORMS (Continued)

MODE 1 (STROBED INPUT)



MODE 1 (STROBED OUTPUT)

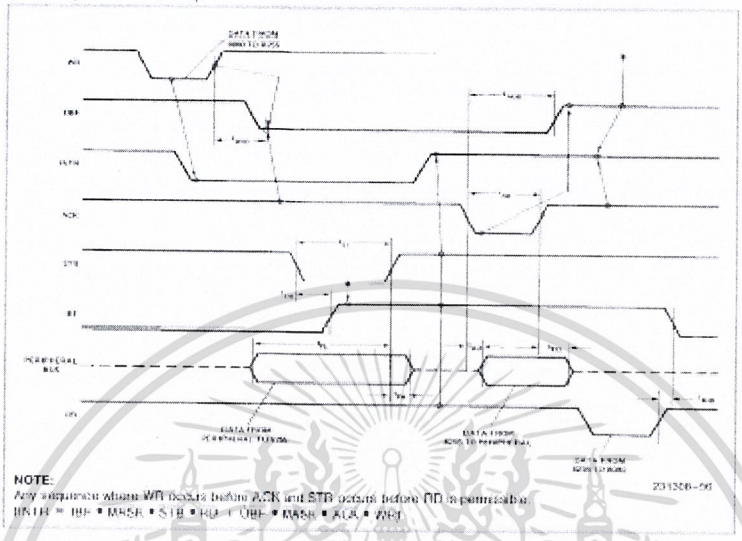


* INTEL CORPORATION, 1980

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

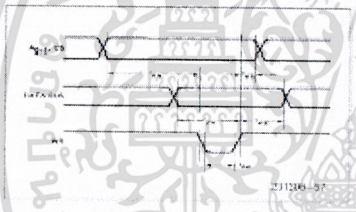
WAVEFORMS (Continued)

MODE 2 (BIDIRECTIONAL)

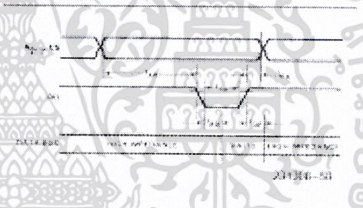


NOTE:
Any sequence where WR occurs before ACK and STB occurs before RD is permissible.
(INTR = IBF • MASK • STB • RD + OBF • MASK • ACK • WR)

WRITE TIMING



READ TIMING



REVISION HISTORY

The following list represents the key differences between REV. 003 and 004 of the 8255A Data Sheet.

1. References to the "8255A" are changed to "8255A-5".
2. Specifications for the 8255A are removed. Only the 8255A-2 remains in production.

© INTEL CORPORATION, 1983

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

กฤษดา วิศวธีรานนท์. **PC ตัวควบคุมซีเควันซ์ หลักการทำงานและการประยุกต์**. พิมพ์ครั้งที่ 5. กรุงเทพฯ :โรงพิมพ์จุฬาลงกรณ์มหาวิทยาลัย. 2539

พรจิต ประทุมสุวรรณ และคณะ. **ทฤษฎีและการใช้งาน PROGRAMMABLE CONTROLLER**. กรุงเทพฯ : โรงพิมพ์เรือนแก้วการพิมพ์. 2536

สมยศ จุณณะปิยะ. **การใช้งานไมโครคอนโทรลเลอร์ ตระกูล MCS-51**. พิมพ์ครั้งที่ 3. กรุงเทพฯ : คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

กนก กุสุมาลย์นุกูล. **คู่มือการเขียนโปรแกรม Delphi4 ฉบับใช้งานได้จริง**. พิมพ์ครั้งที่ 1. กรุงเทพฯ : บริษัท ซัคเซส มีเดีย จำกัด.2542

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ประวัติ



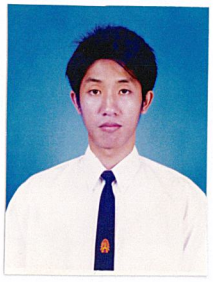
ชื่อผู้ทำปริญญานิพนธ์	นายกฤษฎา จรัสวานิชย์ผล
วันเดือนปีเกิด	7 สิงหาคม พ.ศ. 2524
สถานที่เกิด	จังหวัดนนทบุรี
ภูมิลำเนาเดิม	จังหวัดนนทบุรี
ที่อยู่ปัจจุบัน	211/150 หมู่ 9 ถนนติวานนท์ ตำบลบางกระสอ อำเภอเมือง จังหวัดนนทบุรี 11000
โทรศัพท์	01-3448089

ประวัติการศึกษา

ประถมศึกษา	โรงเรียนติวานนท์ศึกษา
มัธยมศึกษาตอนต้น	โรงเรียนวัดบวรนิเวศ
ประกาศนียบัตรวิชาชีพ (ปวช.)	โรงเรียนช่างกล ขส.ทบ.
ประกาศนียบัตรวิชาชีพชั้นสูง (ปวส.)	เทคโนโลยีสยาม
ปริญญาตรี	สาขาเทคโนโลยีการวัดคุมทางอุตสาหกรรม ภาควิชาครุศาสตร์วิศวกรรม คณะครุศาสตร์อุตสาหกรรม
ผลงานที่ได้รับรางวัล	-
ทุนการศึกษา	-
คติพจน์	ความจริงเป็นสิ่งที่ไม่ตาย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ประวัติ



ชื่อผู้ทำปริญญานิพนธ์	นายจตุพล พลายโถ
วันเดือนปีเกิด	3 มิถุนายน พ.ศ. 2522
สถานที่เกิด	จังหวัดสุพรรณบุรี
ภูมิลำเนาเดิม	จังหวัดสุพรรณบุรี
ที่อยู่ปัจจุบัน	357 หมู่ 2 ตำบลรั้วใหญ่ อำเภอเมือง จังหวัดสุพรรณบุรี 72000
โทรศัพท์	01- 3250025

ประวัติการศึกษา

ประถมศึกษา	โรงเรียนสุพรรณภูมิ
มัธยมศึกษาตอนต้น	โรงเรียนกัณฑสูติศึกษาลัย
ประกาศนียบัตรวิชาชีพ (ปวช.)	วิทยาลัยเทคนิคสุพรรณบุรี
ประกาศนียบัตรวิชาชีพชั้นสูง (ปวส.)	โรงเรียนช่างฝีมือทหาร
ปริญญาตรี	สาขาเทคโนโลยีการวัดคุมทางอุตสาหกรรม ภาควิชาครุศาสตร์วิศวกรรม คณะครุศาสตร์อุตสาหกรรม
ผลงานที่ได้รับรางวัล	-
ทุนการศึกษา	-
คติพจน์	ความแน่นอนคือความไม่แน่นอน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ประวัติ



ชื่อผู้ทำปริญญานิพนธ์	นายเฉลิมชาติ เวียงคำ
วันเดือนปีเกิด	12 สิงหาคม พ.ศ. 2523
สถานที่เกิด	จังหวัดอุตรดิตถ์
ภูมิลำเนาเดิม	จังหวัดอุตรดิตถ์
ที่อยู่ปัจจุบัน	64 หมู่ 1 ตำบลบ้านเกาะ อำเภอเมือง จังหวัดอุตรดิตถ์ 53000
โทรศัพท์	01-8399062

ประวัติการศึกษา

ประถมศึกษา	โรงเรียนเทศบาลท่าอิฐ
มัธยมศึกษาตอนต้น	โรงเรียนอุตรดิตถ์
ประกาศนียบัตรวิชาชีพ (ปวช.)	วิทยาลัยเทคนิคอุตรดิตถ์
ประกาศนียบัตรวิชาชีพชั้นสูง (ปวส.)	สถาบันเทคโนโลยีราชมงคลวิทยาเขตตาก
ปริญญาตรี	สาขาเทคโนโลยีการวัดคุมทางอุตสาหกรรม ภาควิชาครุศาสตร์วิศวกรรม คณะครุศาสตร์อุตสาหกรรม
ผลงานที่ได้รับรางวัล	-
ทุนการศึกษา	-
คติพจน์	เพียงความอดทนเท่านั้นที่จะถึงฝั่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ประวัติ



ชื่อผู้ทำปริญญานิพนธ์	นายชูเกียรติ เนื้อไม้
วันเดือนปีเกิด	23 ธันวาคม พ.ศ. 2522
สถานที่เกิด	จังหวัดกำแพงเพชร
ภูมิลำเนาเดิม	จังหวัดกำแพงเพชร
ที่อยู่ปัจจุบัน	5/1 หมู่ 3 ถนนกำแพงเพชร – สุโขทัย ตำบลพรานกระต่าย อำเภอพรานกระต่าย จังหวัดกำแพงเพชร 62110
โทรศัพท์	055 – 761309

ประวัติการศึกษา

ประถมศึกษา	โรงเรียนอนุบาลกำแพงเพชร
มัธยมศึกษาตอนต้น	โรงเรียนกำแพงเพชรพิทยาคม
ประกาศนียบัตรวิชาชีพ (ปวช.)	สถาบันเทคโนโลยีราชมงคลวิทยาเขตตาก
ประกาศนียบัตรวิชาชีพชั้นสูง (ปวส.)	สถาบันเทคโนโลยีราชมงคลวิทยาเขตตาก
ปริญญาตรี	สาขาเทคโนโลยีการวัดคุมทางอุตสาหกรรม ภาควิชาครุศาสตร์วิศวกรรม คณะครุศาสตร์อุตสาหกรรม
ผลงานที่ได้รับรางวัล	-
ทุนการศึกษา	-
คติพจน์	ทุกปัญหาล้วนย่อมมีทางออกเสมอ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้