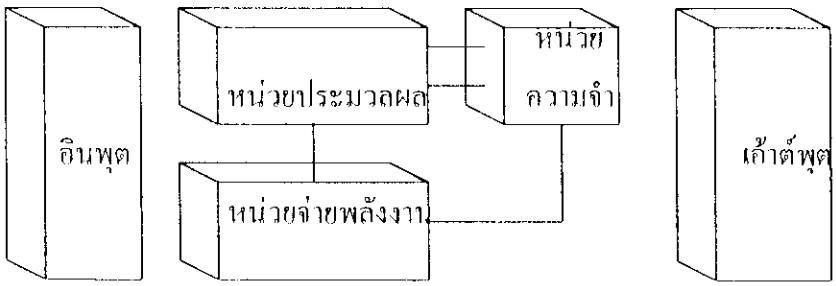
บทที่ 4

การเขียนโปรแกรมควบคุมการทำงานด้วย PLC

4.1 หน่วยอินพุต/เอ้าต์พุต

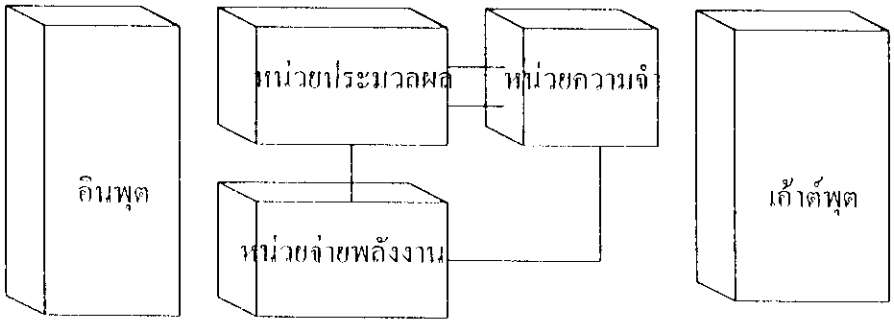
หน่วยอินพุต/เอ้าต์พุตทำหน้าที่ติดต่อระหว่าง PLC กับอุปกรณ์ภายนอก หน่วยอินพุตจะรับสภาวะและค่าที่วัดได้จากอุปกรณ์ภายนอก เช่น การ ON/OFF ของสวิตช์ ตำแหน่งเครื่องจักร ระดับของเหลว อุณหภูมิ ความดัน ระดับแรงดันและกระแสไฟฟ้าส่งต่อให้ PLC CPU จะใช้ค่าหรือสภาวะจากหน่วยอินพุต/เอ้าต์พุตเป็นข้อมูลในการประมวล ผลตามโปรแกรมผู้ใช้ และส่งผลที่ได้ไปหน่วยเอ้าต์พุตเพื่อควบคุมอุปกรณ์ภายนอกเช่น รีเลย์ มอเตอร์ไฟฟ้า ปั๊มและวาล์วดังรูปต่อไปนี้ จะแสดงหน่วยอินพุต/เอ้าต์พุตของ PLC



รูปที่ 4.1 หน่วยอินพุต/เอ้าต์พุต

4.2 หน่วยประมวลผลกลาง

หน่วยประมวลผลกลางหรือ CPU ประกอบไปด้วยหน่วยประมวลผล หน่วยความจำ และหน่วยจ่ายพลังงาน หน่วยประมวลผลทำหน้าที่ดูแลการทำงานทั้งหมดของ CPU คือนำโปรแกรมผู้ใช้ (User Program) มาปฏิบัติเพื่อควบคุมอุปกรณ์ภายนอกตามเงื่อนไขการควบคุมที่ผู้เขียนโปรแกรมต้องการ ควบคุมการติดต่อรับส่งระหว่าง CPU กับหน่วยอินพุต/เอ้าต์พุต และหน่วยอินพุต/เอ้าต์พุตกับอุปกรณ์ภายนอกติดต่อกับผู้ใช้และอุปกรณ์ร่วม ตรวจสอบสภาพการทำงาน ของ PLC โดยมีโปรแกรมบริหารระบบ(Supervisory Program หรือ Operating system) เป็นผู้ควบคุมอีกทีหนึ่ง รูปแสดงโครงสร้างของ CPU ดังต่อไปนี้

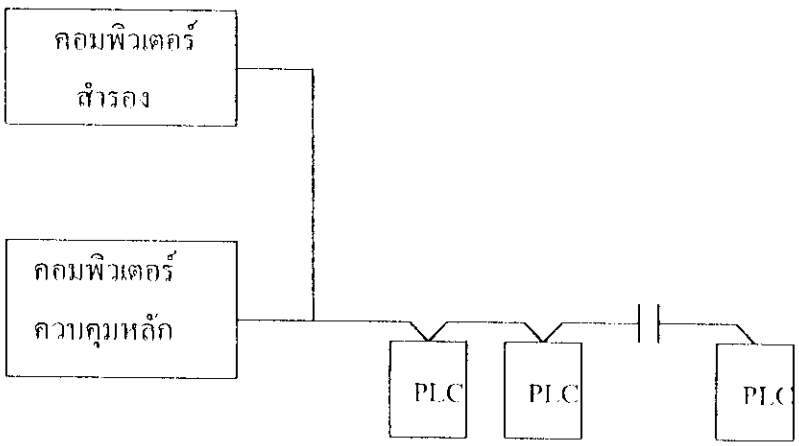


รูปที่ 4.2 โครงสร้างของ CPU

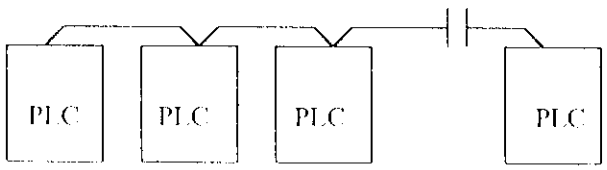
4.3 ระบบโครงข่าย PLC

ปัจจุบันระบบควบคุมขนาดใหญ่ประกอบไปด้วยคอมพิวเตอร์ควบคุมหลัก PLC เครื่องควบคุมหุ่นยนต์อุตสาหะกรรม เครื่องควบคุมเชิงตัวเลข และอุปกรณ์อื่นๆ ที่มีการทำงานร่วมกัน จึงต้องการระบบโครงข่ายที่ใช้ในการสื่อสารข้อมูลระหว่าง PLC ด้วยกัน และระหว่าง PLC กับอุปกรณ์อื่น ๆ ในระบบควบคุมที่มีประสิทธิภาพและความเร็วสูง สามารถทำงานเป็นปกติภายในสภาพแวดล้อมของโรงงานอุตสาหะกรรมต่างๆ เพื่อทำหน้าที่สั่งงานและกระจายข้อมูลที่แสดงสภาพควบคุมและปฏิบัติงานของตนไปยังอุปกรณ์อื่นในระบบควบคุม ระบบโครงข่ายนี้ใช้ในการสื่อสารข้อมูล ของ PLC หรือเส้นทางการสื่อสารความเร็วสูง มีลักษณะที่แตกต่างไปจากระบบโครงข่ายของคอมพิวเตอร์ทั่วไป ถึงแม้สามารถดัดแปลงระบบโครงข่ายของคอมพิวเตอร์เพื่อสื่อสารข้อมูลร่วมกับ PLC ได้ก็ตาม

การสื่อสาร PLC มี 2 ระบบ คือ ระบบนายกับบ่าว (Master slave systems)ใช้คอมพิวเตอร์ควบคุมหลักในการสื่อสารข้อมูลทั้งหมดในระบบโครงข่าย และระบบนายกับนาย(Peer to peer systems)จะไม่มีคอมพิวเตอร์ควบคุมหลักทำหน้าที่ควบคุมการสื่อสารข้อมูลในระบบโครงข่าย



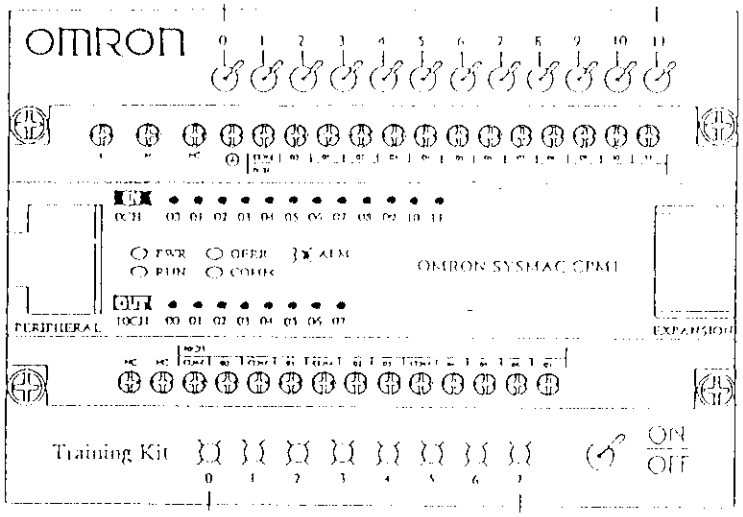
รูปที่ 4.3 การสื่อสารข้อมูลระบบนายกับบ่าว(Master slave system)



รูปที่ 4.4 การสื่อสารข้อมูลระบบนายกับนาย(Peer To peer system)

4.4 ส่วนประกอบของCPM1 กับชุดร่วมฝึกในการ Training

1. CPM1 Training Kit

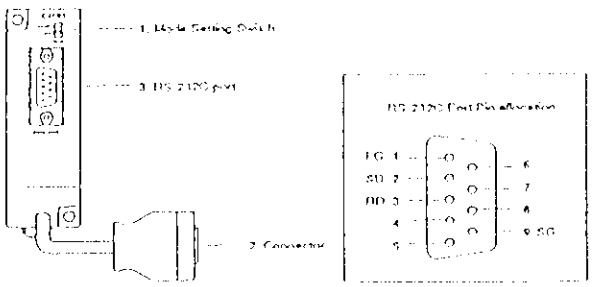


รูปที่ 4.5 CPM1 Training Kit

1.1ในชุด Training Kit CPM1ใช้ต่อกับไฟ AC220V

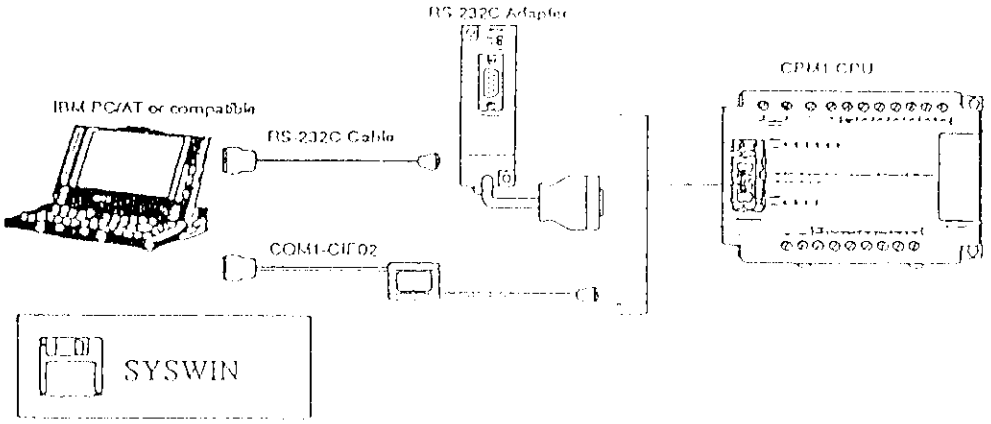
- มี Togle Switch ซึ่งใช้แทนอินพุต จำนวน 12 จุดอินพุต (Input Point)
- มี Lamp Indicators ซึ่งใช้แทน Output จำนวน 8 Output Point

1. RS232C Adapter



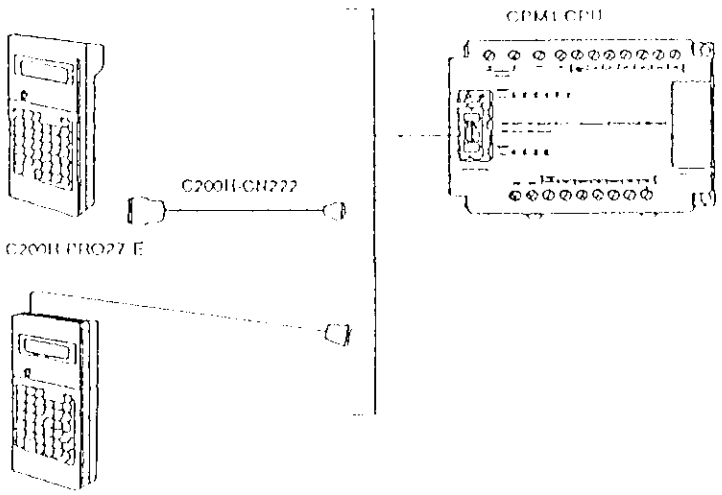
รูปที่ 4.6 RS232C Adapter

เป็นอุปกรณ์ที่ใช้ในการต่อระหว่าง Computer กับ PLC หรือก็คือ ตัวแปลงสัญญาณจาก RS232C เป็นแบบ Peripheralแต่ต้อง set DIPswitch ที่ตัว RS232C adapterไปที่ host link ถ้าใช้ กับ computer



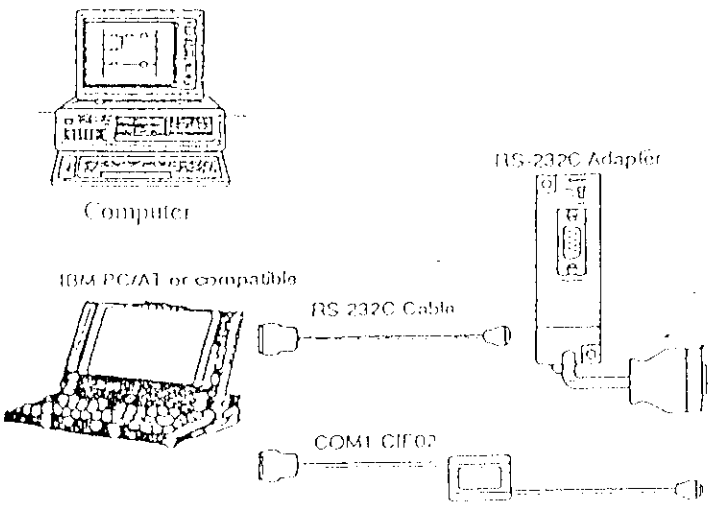
รูปที่ 4.7 รูปแบบของการต่อใช้งานกับ SYSWIN

แต่ถามีการใช้ Programming console ก็จะสามารถต่อได้ที่ Port peripheralของ CPM1 ได้เลยไม่ต้องใช้ RS232C adapter



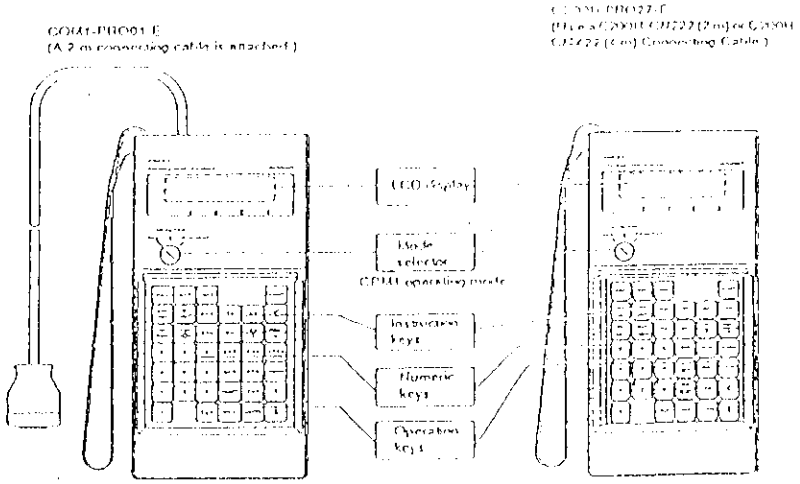
รูปที่ 4.8 รูปแสดงการใช้ Programming console

3.computer เป็นอุปกรณ์ที่ใช้สำหรับติดต่อกับ PLC ด้วยการเขียนโปรแกรมสั่งงานจากcomputer ไม่ว่าจะเป็นโปรแกรม SYSWIN หรือ SSS โดยผ่าน RS232C link adapter



รูปที่ 4.9รูปแสดงการต่ออุปกรณ์ที่ใช้เขียนโปรแกรมจาก computer

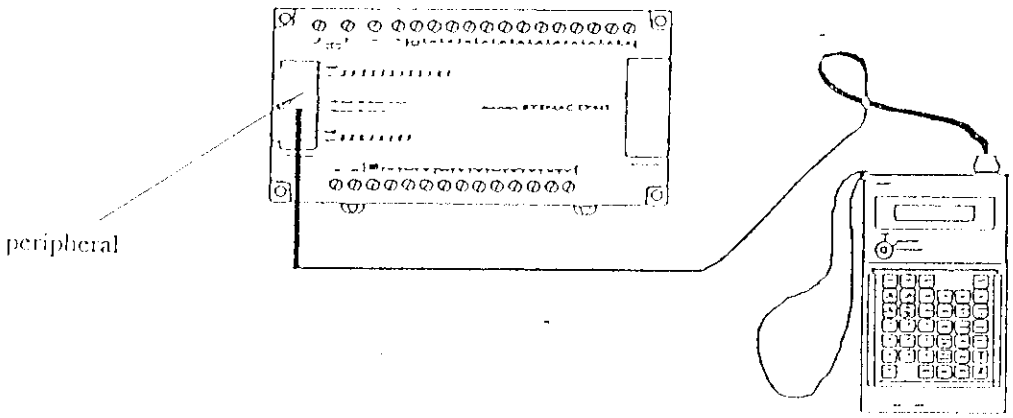
4.Programming console ใช้สำหรับการติดต่อกับ PLC ในรูปของการใช้คำสั่งแบบ mnemonic โดยสามารถต่อโดยตรงทาง peripheral port ที่ตัว CPU เลย



รูปที่ 4.10 รูปแสดงการใช้ Programming console ติดต่อกับ PLC

4.5ขั้นตอนการเขียนโปรแกรมในรูปของ MNEMONIC ด้วย Programming console

ลักษณะของการต่อกับ CPM1 Training Kit กับ Programming console



รูปที่ 4.11 รูปแสดงลักษณะของการต่อกับ CPM1 Training Kit กับ Programming console

การเขียนโปรแกรม

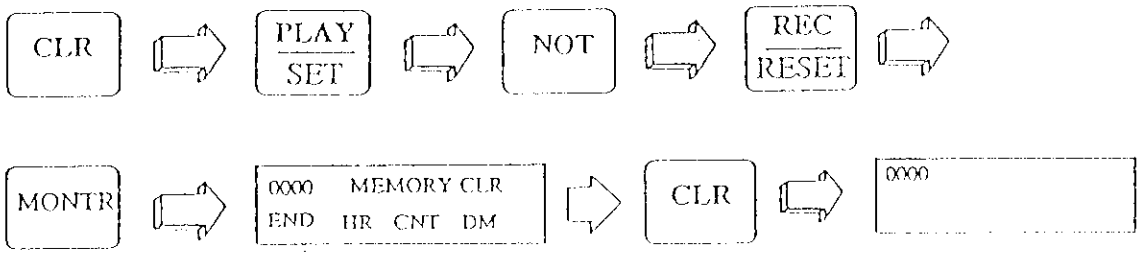
1.ต่อสายของ Programming console โดยตรงเข้าที่ Peripheral Port ของ CPM1 Training Kit

2. จะมี Password ขึ้นที่ Display ของ Programming console จะต้อง key ที่ Programming console เพื่อข้าม password ไปให้ Display เป็น 00000(ต้องอยู่ใน mode ของ Program เท่านั้น)

<PROGRAM>
PASSWORD!

CLR → MONTR → CLR → <PROGRAM>

3.ถ้าต้องการ Clear โปรแกรมใน CPM1 Training Kit ให้ลบออกไป เพื่อจะเขียนโปรแกรมใหม่ตามตัวอย่างที่ให้มา ทำดังนี้



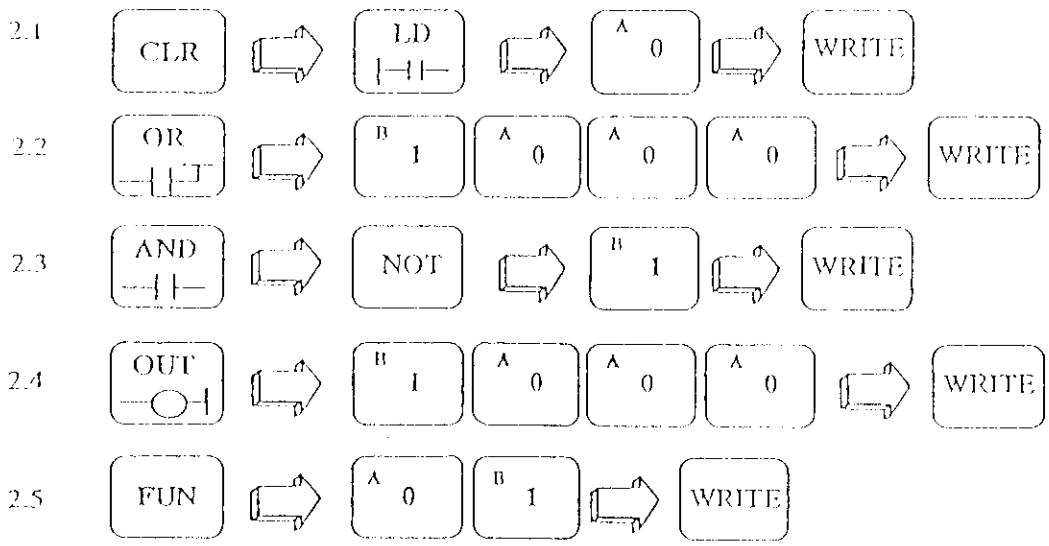
The display should now look like this

4. จากนั้นกด CLR ไปจนกว่า Display ที่ programming console เป็น 00000

การเขียนโปรแกรมตัวอย่าง

1.) เลื่อน Key switch mode ไปที่ Programming console

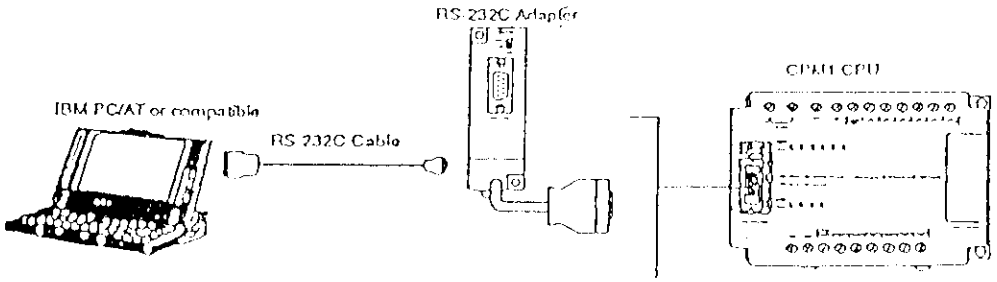
2.) กด Key ดังนี้



5.จากนั้นเลื่อน key switch mode ไปที่ RUN MODE โปรแกรมที่เขียนไว้จะเริ่มทำงานทันที

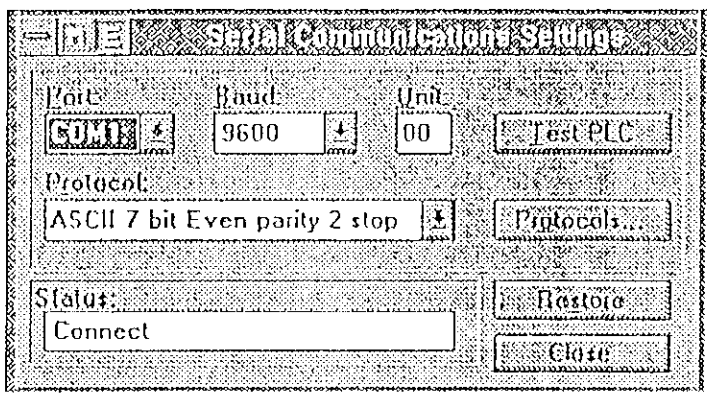
4.6 การ Transfer โปรแกรมที่เราเขียนไว้ไปยัง PLC

1.การต่อ PLC กับ computer ด้วยสาย RS232C ตามรูป



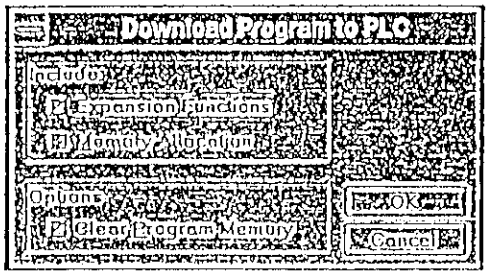
รูปที่ 4.12 แสดงการต่อ PLC กับ computer ด้วยสาย RS232C

2.เข้าไปที่ Project Menu แล้วเลือกไปที่ Serial Communication Setting และ Set ค่าต่างๆ ตามรูป

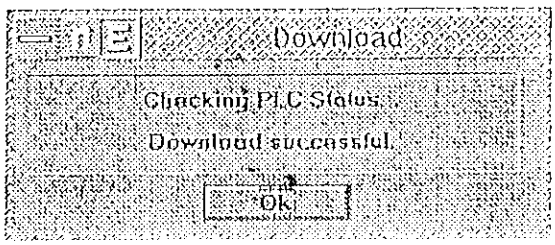


จากนั้นเลือกที่ Test PLC กด click ที่ status ที่ display จะแสดง connect เพื่อแสดงว่าการ communicate ทำได้จากนั้น click ที่ close (indicator Comm ที่ตัว CPU จะกระพริบ)

3.เข้าที่ Online menu และเลือกไปยัง download to PLC จะปรากฏดังรูป เพื่อถามว่าต้องการ clear โปรแกรมใน 0memory หรือไม่แล้วก็ click OK



4.จากนั้นที่หน้าจอจะขึ้นว่า Successful ให้ click OK ตามรูป



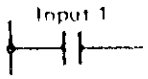
4.7คำสั่งพื้นฐานสำหรับ PLC ทั่วไป

PLC(Programmable Logic controller) หรือ PC(Programmable Controller) เป็นเครื่องที่ใช้สำหรับควบคุมการทำงานของเครื่องจักร โดยการป้อนโปรแกรม ซึ่งการป้อนโปรแกรมนั้น จำเป็นต้องทราบคำสั่งต่าง โดยคำสั่งต่อไปนี้ เป็นคำสั่งพื้นฐานสำหรับ PLC ทั่วไป ทั้งนี้อาจจะแตกต่างกันบ้าง แต่ละยี่ห้อแต่ก็มีหลักการเหมือนๆ กัน

คำสั่ง Load

LOD

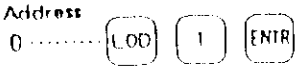
- ตัวอย่างวงจรรีเลย์



- รายการโปรแกรม

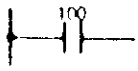
Address	Instruction Word	Data
0	LOD	1
1		

- การป้อนรหัส



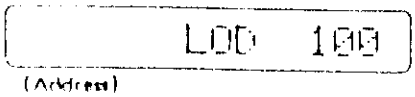
คำสั่งLoad เป็นคำสั่งที่ใช้สำหรับเริ่มต้นของโปรแกรมหรือการเริ่มของแต่ละ Rung (รัง) ตัวอย่างสำหรับ Input, Output, Internal and special relays

- ตัวอย่างวงจรรีเลย์



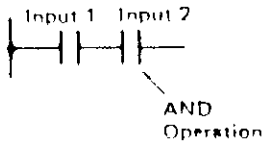
- การป้อนรหัส

- จอแสดงผล



คำสั่ง AND

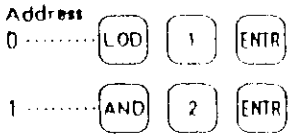
- ตัวอย่างวงจรรีเลย์



- รายการโปรแกรม

Address	Instruction Word	Data
0	LOD	1
1	ANO	2

- การป้อนรหัส



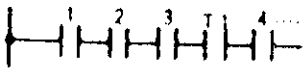
คำสั่งLoad เป็นคำสั่งที่ใช้สำหรับต่อคอนแทกที่ต่อกันเป็นแบบอนุกรม(Series circuit) หลักการทำงานของวงจรเมื่อคำสั่ง AND หรือ AND truth table คือ

Input 1	Input 2	Operation Result
OFF	OFF	OFF
ON	OFF	OFF
OFF	ON	OFF
ON	ON	ON

จากตารางดังกล่าวจะได้ผลลัพท์เป็นสภาวะ ON เมื่อ Input1 และ Input2 มีสภาวะเป็นON ทั้งคู่ แต่ถ้า Input ตัวหนึ่งมีสภาวะเป็น OFF จะได้ผลลัพท์เป็น OFF ด้วย

สามารถใช้คำสั่ง AND ต่อเนื่องภายใน 1 รัง(Rung) เช่น

- ตัวอย่างวงจรรีเลย์

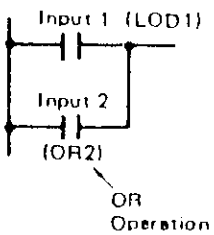


- รายการโปรแกรม

Address	Instruction Word	Data
0	LOD	1
1	AND	2
2	AND	3
3	AND T	1
4	AND	4

คำสั่ง OR

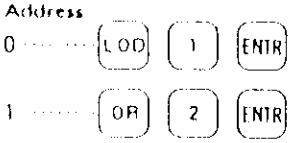
- ตัวอย่างวงจรรีเลย์



- รายการโปรแกรม

Address	Instruction Word	Data
0	LOD	1
1	OR	2

- การป้อนรหัส



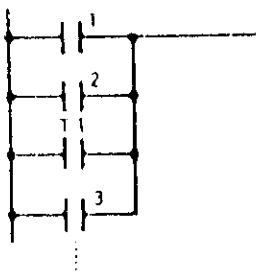
เป็นคำสั่งที่ใช้สำหรับโปรแกรมของคอนแทกที่ต่อกันแบบขนาน(Parallel Contact circuit)หลักการทำงานของวงจร เมื่อใช้คำสั่ง OR หรือ OR truh table

Input 1	Input 2	Operation Result
OFF	OFF	OFF
ON	OFF	ON
OFF	ON	ON
ON	ON	ON

ผลลัพท์ที่มีสภาวะ OFF ได้มีในกรณีเดียวเท่านั้นคือ สภาวะของ Input1 และ Input2 เป็น OFF นอกเหนือจากนั้นผลลัพท์เป็น ON ทั้งหมดเป็นสภาวะของ Input ตัวใดตัวหนึ่งเป็น ON

สามารถใช้คำสั่ง OR ได้อย่างต่อเนื่อง ภายใน 1 รังเช่น

- ตัวอย่างวงจรรีเลย์

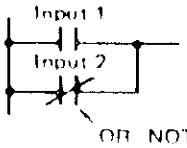


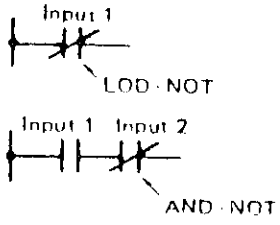
- รายการโปรแกรม

Address	Instruction Word	Data
0	LOD	1
1	OR	2
2	OR T	1
3	OR	3

คำสั่ง NOT

- ตัวอย่างวงจรรีเลย์





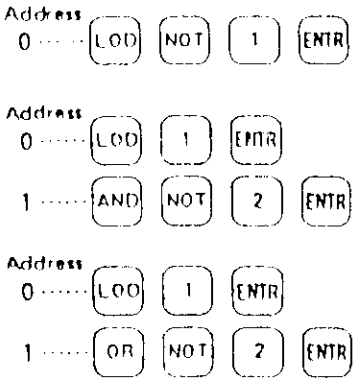
- รายการโปรแกรม

Address	Instruction Word	Data
0	LOD NOT	1

Address	Instruction Word	Data
0	LOD	1
1	AND NOT	2

Address	Instruction Word	Data
0	LOD	1
1	OR NOT	2

- การป้อนรหัส

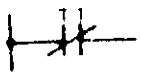


คำสั่ง NOT เป็นคำสั่งที่มีผลตรงข้ามกับคำสั่งเดิม ดังตัวอย่างในตาราง หรือทำหน้าที่เป็นคำสั่งช่วย ของคำสั่ง LOD, AND หรือ OR

Input	Output
OFF	ON
ON	OFF

ตัวอย่างการใช้ NOT input of timer status

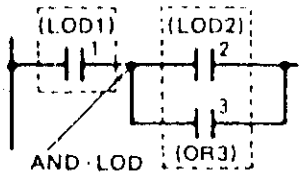
- ตัวอย่างวงจรรีเลย์



- การป้อนรหัส

คำสั่ง AND Load

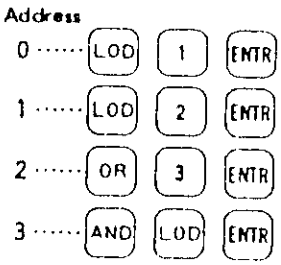
- ตัวอย่างวงจรรีเลย์



- รายการโปรแกรม

Address	Instruction Word	Data
0	LOD	1
1	LOD	2
2	OR	3
3	AND LOD	

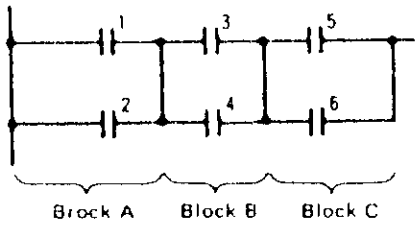
- การป้อนรหัส



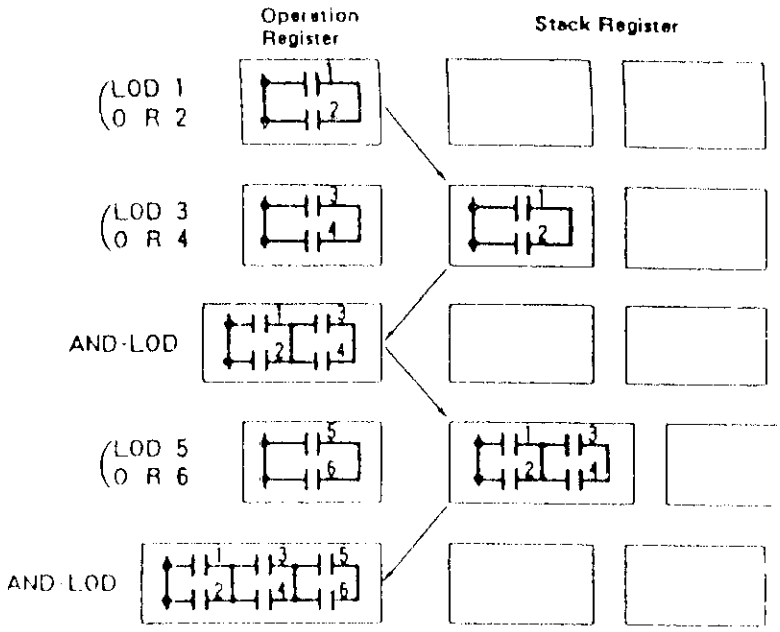
คำสั่ง AND Load เป็นคำสั่งที่เชื่อมวงจรที่เริ่มต้นด้วยคำสั่ LOD ให้ต่อกันเป็นอนุกรม (Series) ขั้นตอนการทำงานในการใช้คำสั่ง AND LOD ใน register และ strack register status

เมื่อใช้คำสั่ง LOD1 แล้วตามด้วย LOD2 OR3 จะทำให้คำสั่ง LOD1 ถูก shift down ไว้ใน strack register เมื่อใช้คำสั่ง AND LOD จะทำให้คำสั่ง LOD1 ถูก shift up กับคำสั่ง LOD2 ,OR3 ทันทีโดยต่อกันแบบอนุกรม

ตัวอย่างวงจรที่สามารถใช้คำสั่ง AND LOD ได้สองแนวทางคือ

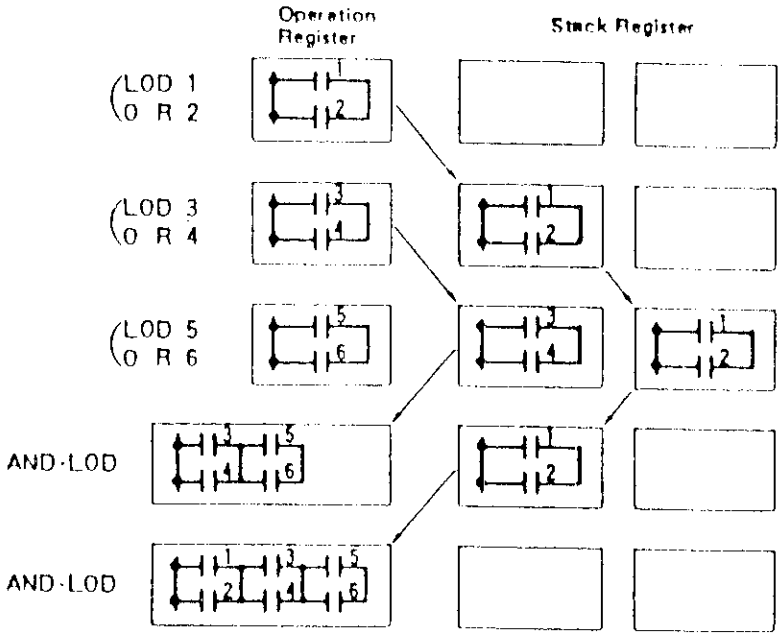


แนวทางที่ 1 ใช้คำสั่ง LOD1,OR2 แล้วตามด้วยคำสั่ง LOD3,OR4จะทำให้คำสั่ง LOD1,OR2



Instruction Word	Data
LOD	1
OR	2
LOD	3
OR	4
AND LOD	
LOD	5
OR	6
AND LOD	

ถูก shift down เก็บไว้ใน strack register แต่เมื่อใช้คำสั่ง AND LOD จะทำให้คำสั่ง LOD1,OR2 ถูก shift down กับคำสั่ง LOD3,OR4 แบบอนุกรม(AND) ทันที เมื่อคำสั่ง LOD5,LOD6 จะทำให้กลุ่มของ AND LOD ในครั้งแรกถูก shift down เก็บไว้ใน strack register ต่อเมื่อใช้คำสั่ง AND LOD อีกครั้งหนึ่ง ทำให้กลุ่มของ AND LOD ครั้งแรกถูก shift up กับ LOD5,OR6 ทันที แนวทางที่2 ใช้คำสั่ง LOD1,LOD2ตามด้วยคำสั่ง LOD3,OR4 จะทำให้คำสั่ง LOD1,OR2 ถูก



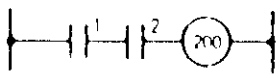
Instruction Word	Data
LOD	1
OR	2
LOD	3
OR	4
LOD	5
OR	6
AND LOD	
AND LOD	

Shift down เก็บไว้ที่ strack register 1 เมื่อใช้คำสั่ง LOD5,OR6 จะทำให้ คำสั่ง LOD1,OR2 ถูกShift down ใน strack register สองส่วน LOD3,OR4 ถูก shift down ใน strack register 1 เมื่อใช้คำสั่ง AND LOD จะทำให้ คำสั่ง LOD3 ,OR4 ถูก shift up กับคำสั่ง LOD5,OR6 และเมื่อใช้คำสั่ง AND LOD ครั้งที่ 2 จะทำให้คำสั่ง LOD1,OR2ถูก shift up กับ ชุดคำสั่งของ AND LOD ครั้งแรกทันที

หมายเหตุคำสั่ง AND LOD = จำนวนของคำสั่ง LOD ลบด้วย 1

คำสั่ง OUTPUT

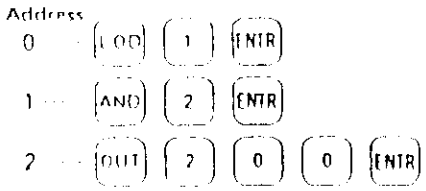
- ตัวอย่างวงจรรีเลย์



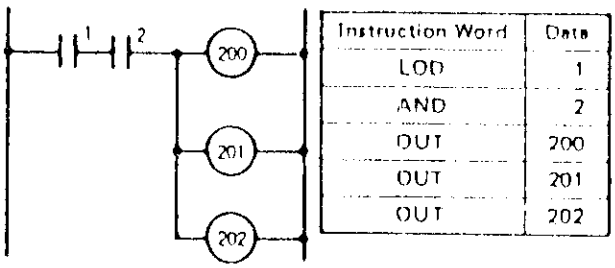
- รายการโปรแกรม

Address	Instruction Word	Data
0	LOD	1
1	AND	2
2	OUT	200

- การป้อนรหัส

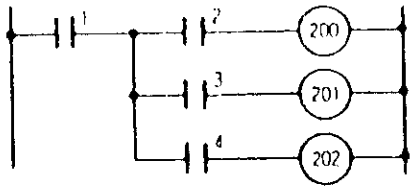


คำสั่งOUT เป็นคำสั่งที่ใช้สำหรับใช้เกกิดผลลัพท์และแสดงถึงการจบของวงจรในแต่ละรัง สามารถใช้คำสั่ง OUT ได้หลายๆ ตัวใน 1 รัง เช่น

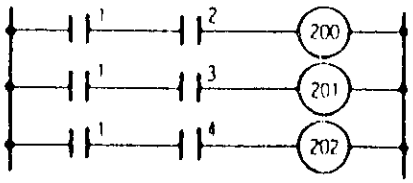


Instruction Word	Data
LOD	1
AND	2
OUT	200
OUT	201
OUT	202

วงจรไม่สามารถโปรแกรมได้เช่น

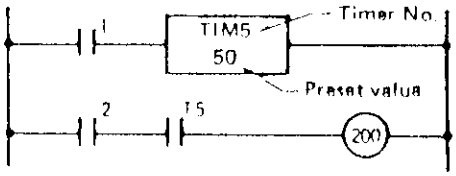


ควรเปลี่ยนเป็นวงจรดังนี้



คำสั่ง TIM

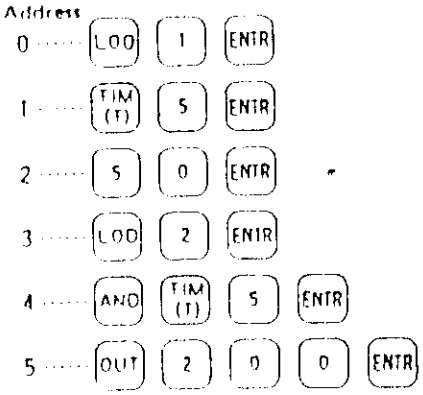
- ตัวอย่างวงจรรีเลย์



- รายการโปรแกรม

Address	Instruction Word	Data
0	LOD	1
1	TIM	5
2		50
3	LOD	2
4	AND T	5
5	OUT	200

- การป้อนรหัส



การที่จะให้ timer ทำงานได้ถูกต้องนั้นจะต้องมีค่าที่ใส่เข้าไปสองค่าด้วยกันคือ ค่าหมายเลขของไทเมอร์ และค่าที่ให้ไทเมอร์จับเวลาทำงาน ซึ่งมีค่าตั้งแต่ 0-9999 สามารถใช้คำสั่ง OUT หลังคำสั่ง TIM ได้ทันทีคุณสมบัติอื่นๆ ของTIM

Address	Instruction Word	Data
0	LOD	1
1	TIM	5
2		50
3	OUT	200

- เมื่ออินพุตของไทเมอร์มีสภาวะเป็น on จะทำให้สัญญาณนาฬิกาเริ่มจับค่าเวลาทันที
- เมื่อไทเมอร์ทำงานถึงค่าเวลาที่ตั้งไว้(Present value time)ทำให้เอาต์พุตของไทเมอร์มีสภาวะเป็น ON
- เมื่ออินพุตของไทเมอร์มีสภาวะเป็น OFF ทำให้ Present value ถูกตั้งค่าทันที
- หลังจากเวลาได้ผ่านไปถึงค่านับ จะทำให้ค่าที่นับนั้นๆ ยังคงเป็น 0 จนกระทั่ง timer input มีสภาวะ OFF
- ไม่สามรถใช้ไทเมอร์ตัวเดียวกันในโปรแกรมเดียวกันมากกว่า 1 ครั้ง
- ถ้าเปลี่ยนแปลง present value ขณะที่ไทเมอร์กำลังทำงาน จะทำให้ไทเมอร์ยังคงค่าเดิมอยู่ (ไม่เปลี่ยนแปลงตามค่าใหม่ แต่อย่างไรก็ตามถ้า present value เปลี่ยนค่าเป็น 0 ทำให้ไทเมอร์หยุดการทำงานทันที