หอสมุดกลาง สถาบันวิทยบริการ จุฬาลงกรณ์มหาวิทยาลัย

บทที่ 4

เครื่องควบคุมแบบลำดับที่โปรแกรมได้โดย FPGA

4.1 ลักษณะของเครื่องควบคุมชนิดโปรแกรมได้

เครื่องควบคุมชนิดโปรแกรมได้ที่มีอยู่ในปัจจุบันแบ่งออกได้เป็น 3 ขนาดใหญ่ โดยพิจารณาจากจำนวนอินพุต/เอาต์พุต และขนาดหน่วยความจำที่สามารถเขียนคำสั่งได้

ขนาด PLC	จำนวนอินพุต/เอาต์พุตสูงสุด	ขนาดหน่วยความจำ (ไบต์)
ขนาดเล็ก	40/40	2 K
ขนาดกลาง	128/128	8 K
ขนาดใหญ่	> 128 /> 128	> 8 K

ตารางที่ 4.1 แสดงการแบ่งขนาดของ PLC

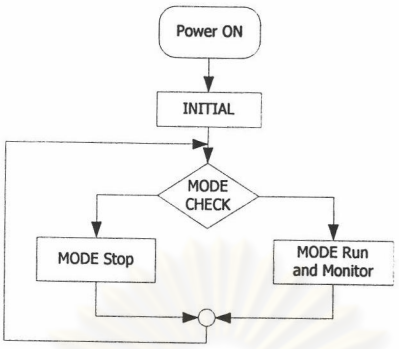
4.2 แนวคิดในการออกแบบ PLC

ออกแบบ PLC ขนาดเล็กที่คำนึงถึงความสามารถและความสะดวกในการใช้งานหรือการบำรุงรักษา สามารถติดต่อกับสัญญาณอินพุตและเอาต์พุตมาตรฐานต่างๆ ได้ ภาษาที่โปรแกรมควรจะง่าย และสามารถแก้ไขเปลี่ยนแปลงโปรแกรมที่หน้างานได้

4.3 แนวความคิดในการออกแบบระบบ

1. คำสั่งในการโปรแกรม ต้องประกอบไปด้วยคำสั่งพื้นฐานของ PLC และคำสั่งเพิ่มเติมเพื่อให้เพียงพอในการใช้งานในอุตสาหกรรม
2. การป้อนโปรแกรมและแสดงผล ระบบที่ออกแบบควรจะมีความสะดวกในการป้อนโปรแกรม และต้องสามารถเปลี่ยนแปลงแก้ไขโปรแกรมที่หน้างานได้
3. จำนวนคำสั่งที่สามารถโปรแกรมได้ จำนวนคำสั่งนี้ควรจะสัมพันธ์กับจำนวนอินพุตและเอาต์พุตสูงสุดของระบบด้วย และต้องมากพอสำหรับโปรแกรมที่ใช้งานได้จริง

4.4 แนวความคิดในการออกแบบซอฟต์แวร์



รูปที่ 4.1 ลักษณะการทำงานของโปรแกรมควบคุม

โปรแกรมควบคุมการทำงานของเครื่อง PLC ควรออกแบบให้มีประสิทธิภาพที่สุด โดยแบ่งการทำงานได้เป็น 2 โหมด [1] คือ

1. **โหมดหยุด** (Stop mode)

ทำหน้าที่รอการป้อนโปรแกรมขั้นบันไดจากผู้ใช้ แล้วเขียนลงในหน่วยความจำโปรแกรม (EEPROM) เพื่อสามารถเก็บรักษาโปรแกรมขั้นบันไดขณะที่ไม่มีไฟเลี้ยง จากนั้นจึงค่อยอ่านโปรแกรมจากหน่วยความจำโปรแกรม (EEPROM) มาไว้ในหน่วยความจำภายในส่วนโปรแกรม เพื่อรอพร้อมเข้าสู่โหมดทำงานต่อไป

2. **โหมดทำงาน** (Run and Monitor mode)

ในโหมดนี้ PLC ทำงานตามคำสั่งของโปรแกรมขั้นบันไดที่เก็บในหน่วยความจำภายในส่วนโปรแกรม และส่งค่าสถานะของรีเลย์ภายในมายังคอมพิวเตอร์ส่วนบุคคลเพื่อแสดงผล

4.5 ตัวอย่าง PLC ขนาดเล็ก

Item	OMRON PLC [15] CPM2A-20	FUJI [16] NB0-P14	MITSUBISHI [17] MELSEC FX1S-14MR
จำนวน Input	12	8	12
จำนวน Output	8	6	8
Output Type	Relay	Relay	Relay
Execution Time	Basic Ins 0.64 μS/Step Special Ins 7.8 μS/Step	Seq Ins 0.7 to 10.3 μS/Step Data Ins 4.9 to 56 μS/Step	Basic Ins 0.55 to 0.7 μS/Step Applied Ins 1.65 to 100 μS/Step
Memory Size Type	RAM, FLASH	320 Steps CMOS-RAM, EEPROM	RAM, EEPROM
Timer/Counter	256/256	32/32	95/16
Timer Range	0.001 – 65535 วินาที	0.01 – 655 วินาที	0.001 – 3276.7วินาที
Counter Range	1 - 65535	1 - 65536	1 - 32767
Data Registers		32	128
Internal Coil	256	256	384
Program Capacity	4,096 ขั้น	320 ขั้น	2000 ขั้น
Programming	Ladder	Ladder, Mnemonic	Ladder with Func B
จำนวนคำสั่ง	Basic 14 Special Ins 105	Seq Ins 23 Data Ins 21	Basic Ins 20 Applied Ins 35

ตารางที่ 4.2 ตัวอย่าง PLC ขนาดเล็กที่ใช้ในอุตสาหกรรม

I20651843

4.6 ข้อกำหนด (Specifications) ของ PLC ในวิทยานิพนธ์นี้

จากตารางที่ 4.2 แสดงถึงข้อกำหนด (Specifications) ของ PLC ขนาดเล็กที่ใช้กันอยู่จริงในอุตสาหกรรม จึงได้ทำข้อกำหนด PLC ของวิทยานิพนธ์นี้ไว้ตามตารางที่ 4.3 เพื่อให้มีความสามารถในการใช้งานได้จริง แต่ทั้งนี้ต้องคำนึงถึงทรัพยากรของ FPGA ชิปที่เลือกใช้ด้วย ซึ่งได้เลือกใช้ FPGA ของบริษัท ALTERA EPF10K70RC240-4

ระบบการควบคุม	PLC คอนโทรลเลอร์ชิป โดย FPGA CHIP
จำนวนอินพุต	16 จุด
จำนวนเอาต์พุต	8 จุด
ตัวป้อนโปรแกรม	ป้อนโปรแกรมทางเครื่องคอมพิวเตอร์ส่วนบุคคล ผ่านทางพอร์ตอนุกรม RS232
ระบบการโปรแกรม	โปรแกรมภาษาขั้นบันได (Ladder Diagram)
ความยาวของคำสั่ง	2 Byte / Step
คำสั่งพื้นฐาน	มีคำสั่ง LD, LDI, AND, ANDI, OR, ORI, OUT, SET, RST, CNT, TIM, ANDB, ORB, END, MC, MCC, CMP, JMP, JME, ADD, SUB, MOV, NOP
ความเร็วเฉลี่ยคำสั่ง	1 μS/Step
Output Type	Relay
Memory Type	1k Bytes RAM, EEPROM
Program Capacity	384 ขั้น
Timer/Counter	60/60
Timer Range	0.001 – 163830 วินาที
Counter Range	1 – 163830
Data Registers	128 × 16 บิต
Input Unit Voltage	24 Volt DC

ตารางที่ 4.3 สรุปรวมลักษณะ PLC คอนโทรลเลอร์ชิป

4.7 ระบบเลขฐาน

เพื่อให้เกิดความง่ายและคุ้นเคยกับผู้ใช้จึงเลือกใช้ระบบเลข**ฐานสิบ**กับ PLC นี้ ซึ่งค่า Operand จะนับต่อเนื่องกันเริ่มจาก 0 ถึงจำนวนรีเลย์ภายในชนิดนั้นสูงสุดลบหนึ่งเสมอ เช่น จำนวน Input สูงสุดคือ 16 ก็จะแทน Input ด้วย I0 ถึง I15

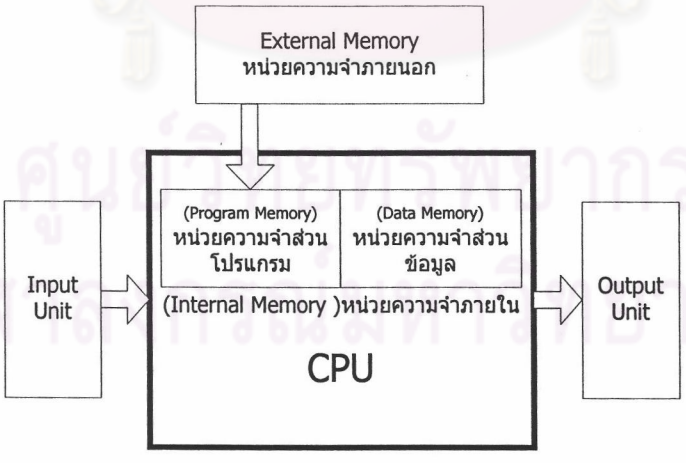
ตัวอย่าง

LD	I10
AND	A19
OR	T30
OUT	O7

4.8 การอ่านเขียนอินพุต/เอาต์พุต

จากหัวข้อ 2.8 เพื่อไม่ให้เกิดปัญหาเมื่อโปรแกรมขั้นบันไดยาว สถานะของอินพุตตอนต้นโปรแกรม และตอนท้ายโปรแกรมไม่ตรงกันทำให้เกิดความผิดพลาดในการควบคุมจึงเลือกใช้วิธีเขียนอินพุต/เอาต์พุตแบบ Mass input/output copy

4.9 โครงสร้างหน่วยความจำของ PLC



รูปที่ 4.2 โครงสร้างหน่วยความจำของ PLC

หน่วยความจำของ PLC แบ่งได้ออกเป็น 2 ส่วน คือ

1. **หน่วยความจำภายใน** (Internal Memory) ใช้ SRAM ที่อยู่ที่อยู่ภายใน FPGA ชิป ขณะที่ PLC ทำงานการควบคุมจะติดต่อกับหน่วยความจำส่วนนี้ ซึ่งมีทั้งข้อมูลโปรแกรมขั้นบันได และข้อมูลสถานะของรีเลย์ภายใน แบ่งออกเป็น 2 ส่วนดังนี้
   a. หน่วยความจำภายในส่วนโปรแกรม (Program Memory)
   b. หน่วยความจำภายในส่วนข้อมูล (Data Memory)
2. **หน่วยความจำภายนอก** (External Memory) ใช้ EEPROM สำหรับหน่วยความจำส่วนนี้ ทำหน้าที่เก็บโปรแกรมขั้นบันไดไว้ขณะที่ PLC ไม่มีไฟเลี้ยง

4.10 โครงสร้างการจัดเก็บข้อมูลในหน่วยความจำภายใน

Internal Mem addr	0 - 15
0 - 383	Ladder Program

	Internal Mem addr	Data Mem addr	0 - 15
180H	384-7	0 - 3	Input Relay (I0 - I63)
	388-9	4 - 5	Output Relay (O0 - O31)
	390 - 399	06 - 15	Auxiliary Relay (A0 - A159)
	400-1	16 - 17	SPARE | GR | EQ | LE SPARE
	402 - 461	18 - 77	Timer/Counter Data (T/C 0 - T/C 59)
	462 - 511	78 - 127	Data Register (D0 - D49)

รูปที่ 4.3 โครงสร้างหน่วยความจำภายใน (Internal Memory)

หน่วยความจำภายในที่กำหนดไว้ใน PLC นี้มีขนาด 512 word หรือขนาด 1 Kbyte โดยแบ่งออกเป็น 2 ส่วนคือ

1. **หน่วยความจำภายในส่วนโปรแกรม** (Program Memory) ส่วนนี้จะเก็บโปรแกรมขั้นบันได เมื่อ PLC ทำงานจะเรียกโปรแกรมขั้นบันไดจากหน่วยความจำในส่วนนี้ทีละคำสั่ง (Fetch) มาทำการตีความ (Decode) ปฏิบัติตามคำสั่ง (Execute) ทุกครั้งที่มีการเปิดเครื่องหรือเมื่อมีการเขียนโปรแกรมใหม่จากผู้ใช้ โปรแกรมขั้นบันไดจะถูกอ่านจากหน่วยความจำโปรแกรมภายนอกมาไว้หน่วยความจำภายในส่วนโปรแกรมนี้

2. **หน่วยความจำภายในส่วนข้อมูล** (Data Memory) ส่วนนี้จะเก็บสถานะรีเลย์ในการควบคุมต่างๆ ไว้ เมื่อ PLC ทำงานตามคำสั่งโปรแกรมขั้นบันไดจะติดต่ออ่าน/เขียนสถานะรีเลย์กับหน่วยความจำในส่วนนี้ แบ่งตามชนิดต่างๆ [1] ได้ดังนี้

Input Relay

หมายถึง หน่วยความจำที่เก็บสถานะของอินพุต ในการทำงานทุกๆ 1 รอบการทำงาน (Scan Time) PLC จะอ่านสถานะของอินพุตแล้วนำมาเก็บลงไว้ใน Input Relay จำนวนอินพุตจะต้องมีมากพอ ใช้สัญลักษณ์ "I"

Output Relay

หมายถึง หน่วยความจำที่เก็บสถานะของเอาต์พุต ในการทำงานทุกๆ 1 รอบการทำงาน (Scan Time) จากการทำงานตามคำสั่งโปรแกรมบันไดได้ผลลัพธ์แล้วนำมาเก็บใน Output Relay จำนวนเอาต์พุตต้องมีมากเพียงพอ มักกำหนดไว้ที่ครึ่งหนึ่งของจำนวนอินพุต ใช้สัญลักษณ์ "O"

Auxiliary Relay

หมายถึง รีเลย์ช่วยในการทำงานเหมือนกับ Input / Output Relay ต่างกันเพียงไม่ได้ติดต่อกับอินพุตและเอาต์พุตโดยตรง โดยมากแล้วจะกำหนดให้มีจำนวนมากกว่าเป็นสองเท่าของจำนวน Input/Output Relay รวมกัน ใช้สัญลักษณ์ "A"

Timer/Counter Data

หมายถึง หน่วยความจำที่ใช้เก็บข้อมูลของคำสั่ง TIM หรือคำสั่ง CNT การเก็บข้อมูลจะมีขนาด 16 บิต แสดงในรูปที่ 4.4 โดย 14 บิตแรกใช้เก็บข้อมูลการนับ บิตที่ 15 เป็น LS (Last Status) สำหรับเก็บสถานะเก่าของสัญญาณ และบิตที่ 16 เป็น O (Output Status) สำหรับเก็บสถานะ On/Off ของ Timer/Counter นั้นๆ เนื่องจากทั้ง Timer และ Counter นั้นใช้พื้นที่หน่วยความจำเดียวกัน ดังนั้นเมื่อตัวเลขกำหนดตำแหน่งในหน่วยความจำใดถูกใช้ไปแล้วไม่ว่าจะเป็นของ Timer หรือ Counter จะไม่สามารถใช้ได้อีก สำหรับเครื่อง PLC นี้มี Timer/counter 60 ตำแหน่ง ใช้สัญญาลักษณ์ "T" หรือ "C"

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
O	LS	Count Number													

O : Output status **LS** : Last Pulse Status Bit 13 - 0 Count Number 14bits

รูปที่ 4.4 แสดงรูปแบบข้อมูล Timer/Counter Data

Data Register

เป็นหน่วยความจำสำหรับเก็บข้อมูลที่ใช้กับคำสั่งเกี่ยวกับการจัดการข้อมูล หน่วยความจำนี้แต่ละตัวกำหนดไว้ให้มีขนาด 14 บิต บิตที่ 15 และ 16 ไม่ใช้ เพื่อให้คำสั่งเกี่ยวกับการจัดการข้อมูล (Data Handling) สามารถใช้รวมได้กับทั้ง Timer/Counter Data ที่มีขนาด 14 บิตเช่นกัน รูปแบบของ Data register แสดงในรูปที่ 4.5 ใช้สัญญาลักษณ์ "D"

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
X	X	Data													

X : Not used Bit 13 - 0 Data 14bits

รูปที่ 4.5 แสดงรูปแบบข้อมูล Data Register

4.11 โครงสร้างการเก็บโปรแกรมภาษาขั้นบันไดในหน่วยความจำภายนอก

Ladder Program Statement List

	command	address	machine code
0	LD	I00	0000000000000000
1	AND	A10	0010000001101010
2	OUT	O07	0110000000100111
n	END		1111111100100000
512			

EEPROM 16x512

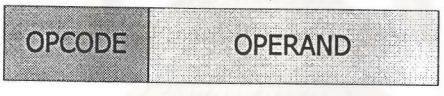
7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	1	1	0	1	0	1	0
0	0	1	0	0	0	0	0
0	0	1	0	0	1	1	1
0	1	1	0	0	0	0	0
0	0	0	0	0	0	0	0
1	1	1	1	0	0	1	0

รูปที่ 4.6 การเก็บข้อมูลโปรแกรมขั้นบันไดในหน่วยความจำภายนอก

เนื่องจาก PLC ต้องสามารถเก็บโปรแกรมขั้นบันไดไว้ได้ในขณะที่ไม่ได้เปิดเครื่อง ดังนั้น PLC จึงจำเป็นต้องมีหน่วยความจำโปรแกรมภายนอก ซึ่งใช้ EEPROM ในการเก็บโปรแกรม การทำงานจะเริ่มเมื่อผู้ใช้ต้องการเขียนโปรแกรมขั้นบันไดใหม่ลงใน PLC ต้องทำในขณะที่ PLC อยู่ในโหมดหยุด (Stop Mode) CPU ของ PLC จะรับค่าโปรแกรมมาทีละ 8 บิต แล้วทำการเขียนไปยังหน่วยความจำโปรแกรมภายนอกจนจบข้อมูลโปรแกรม จากนั้นทำการอ่านจากหน่วยความจำโปรแกรมภายนอก แล้วนำไปเขียนใส่หน่วยความจำภายในส่วนโปรแกรม เพื่อเตรียมพร้อมสำหรับการทำงานตามโปรแกรมต่อไป

การที่ไม่กำหนดให้ CPU ของ PLC ขณะอยู่ในโหมดทำงานติดต่อกับหน่วยความจำโปรแกรมภายนอกโดยตรงก็เนื่องจากการอ่าน/เขียนค่าในหน่วยความจำนี้ใช้เวลานานมากกว่าการอ่านข้อมูลจากหน่วยความจำภายในส่วนโปรแกรม สำหรับโครงสร้างการเก็บข้อมูลในหน่วยความจำโปรแกรมภายนอกนั้น จากรูปที่ 4.6 จะเก็บข้อมูลภาษาเครื่อง Machine Code แต่ละบรรทัดที่มีขนาด 16 บิต โดยจะเก็บ 8 บิตล่างก่อน แล้วค่อยตามด้วย 8 บิตบน

4.12 Instruction Machine [18,19]



รูปที่ 4.7 รูปแบบ Instruction Machine

คำสั่งในโปรแกรมขั้นบันไดของ PLC ประกอบด้วย 2 ส่วนคือ OPCODE และ OPERAND ในส่วนของ OPCODE มีขนาด 4 บิตจะบอกว่าเป็นคำสั่งอะไรและ OPERAND คือข้อมูลในหน่วยความจำภายในส่วนข้อมูลที่คำสั่งนั้นๆ ติดต่อด้วย อาจมีขนาด 12 หรือ 8 บิต ขึ้นอยู่กับว่าติดต่อกับข้อมูลส่วนใด ถ้าต้องติดต่อกับ Input/Output หรือ Auxiliary รีเลย์ ต้องใช้ 12 บิตในการอ้างเพื่อเข้าถึงตำแหน่งนั้นๆ ถ้าเป็นข้อมูล Timer/Counter ที่เป็นแบบ Word 16 บิตใช้เพียง 8 บิตในการอ้างเพื่อเข้าถึงตำแหน่งนั้นและถ้าเป็นคำสั่งเช่น ANDB, ORB หรือ END คำสั่งเหล่านี้ไม่ต้องติดต่อกับหน่วยความจำเลย จึงไม่ต้องมีการอ้างเพื่อเข้าถึงข้อมูล ดังนั้นหนึ่งคำสั่งจึงกำหนดไว้ 16 บิต และเพื่อเพิ่มคำสั่งให้มากขึ้น จึงกำหนด Instruction Machine ไว้ 3 แบบ ตามการเข้าถึงข้อมูลคือ

แบบที่ 1 Normal Instruction Machine คำสั่งที่ติดต่อกับข้อมูลแบบบิต

Normal Instructions Machine

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Op-code				Address								bitno			

Instruction Machine แบบนี้ใช้กับคำสั่งที่ต้องการเข้าถึงข้อมูลแบบบิต โดยกำหนดให้ 4 บิตแรก (บิตที่ 15 ถึง 12) เป็น Op-code ตัวบอกว่าคำสั่งนี้เป็นคำสั่งอะไร 8 บิตถัดมา (บิตที่ 11 ถึง 4) เป็น Address ตัวบอกว่าติดต่อกับหน่วยความจำภายในส่วนข้อมูลตำแหน่งใด และ 4 บิตสุดท้าย (บิตที่ 3 ถึง 0) เป็น bitno ตัวบอกว่าเป็นข้อมูลในบิตที่เท่าไร

แบบที่ 2 Expand 1 Instruction Machine

Expand 1 Instructions Machine

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	1	1	1	Op-code expand				Address							

Instruction Machine แบบนี้ใช้กับคำสั่งที่ต้องการเข้าถึงข้อมูลแบบ Word โดยกำหนดให้ 4 บิตแรก (บิตที่ 15 ถึง 12) เป็น "1111" เพื่อเป็นส่วนขยายจาก Normal Instruction Machine 4 บิตถัดมา (บิตที่ 11 ถึง 8) เป็น Op-code ตัวบอกว่าคำสั่งนี้เป็นคำสั่งอะไร 8 บิตสุดท้าย (บิตที่ 7 ถึง 0) เป็น Address ตัวบอกว่าติดต่อกับหน่วยความจำภายในส่วนข้อมูลตำแหน่งใด

แบบที่ 3 Expand 2 Instruction Machine

Expand 2 Instructions Machine

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	1	1	1	1	1	1	1	Op-code expand							

Instruction Machine แบบนี้ใช้กับคำสั่งที่ไม่ติดต่อกับข้อมูลในหน่วยความจำภายในส่วนข้อมูลใดเลย โดยกำหนดให้ 8 บิตแรก (บิตที่ 15 ถึง 8) เป็น "11111111" เพื่อเป็นส่วนขยายต่อมาจาก Expand 2 Instruction Machine 4 บิตถัดมา (บิตที่ 7 ถึง 4) เป็น Op-code ตัวบอกว่าคำสั่งนี้เป็นคำสั่งอะไร ส่วน 4 บิตสุดท้ายไม่ได้ใช้

4.13 คำสั่งเบื้องต้นที่ใช้ใน PLC [20]

Normal PLC command

NO	Opcode	Mnemonic	Symbol
0	0000	LD	
1	0001	LDI	
2	0010	AND	
3	0011	ANDI	
4	0100	OR	
5	0101	ORI	
6	0110	OUT	
7	0111	SET	
8	1000	RST	
9	1001	TIM	
A	1010	CNT	
B	1011		
C	1100		
D	1101		
E	1110		
F	1111	EXPAND	

Expand 1 PLC command

NO	Opcode	Mnemonic	Symbol
0	0000	CMP	CMP
1	0001	ADD	ADD
2	0010	SUB	SUB
3	0011	MOV	MOV
4	0100		
5	0101		
6	0110		
7	0111		
8	1000		
9	1001		
A	1010		
B	1011		
C	1100		
D	1101		
E	1110		
F	1111	EXPAND	

Expand 2 PLC command

NO	Opcode	Mnemonic	Symbol
0	0000	ANDB	
1	0001	ORB	
2	0010	END	END
3	0011	MC	MC
4	0100	MCC	MCC
5	0101	JMP	JMP
6	0110	JME	JME
7	0111	NOP	NOP
8	1000		
9	1001		
A	1010		
B	1011		
C	1100		
D	1101		
E	1110		
F	1111		

ตารางที่ 4.4 คำสั่งและสัญลักษณ์พื้นฐานที่ใช้ใน PLC

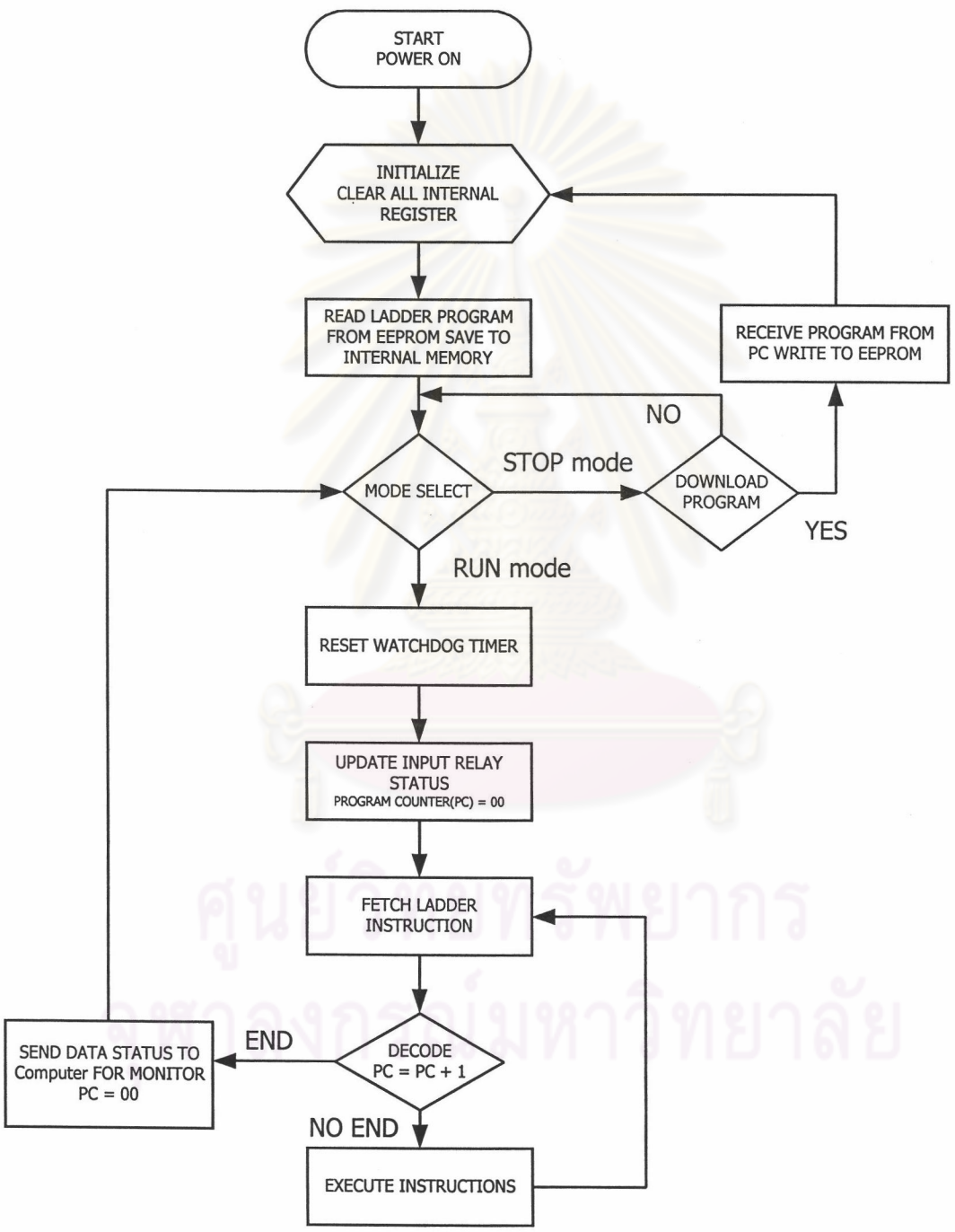
กำหนดคำสั่งของ PLC นี้ไว้ทั้งหมด 23 คำสั่ง โดยมีคำสั่งที่ดำเนินการทางลอจิก เช่น คำสั่ง LD, LDI, AND, ANDI, OR, ORI, OUT คำสั่ง Timer ที่เป็น Delay Time On และ Counter คำสั่งเกี่ยวกับการนับ คำสั่งด้านการจัดการข้อมูล Data Handling มีคำสั่ง CMP, ADD, SUB, MOV คำสั่งที่ใช้ในการควบคุมในโปรแกรมขั้นบันได เช่น ANDB, ORB, END, MC, JMP, NOP โดยทุกคำสั่งจะถูกจัดแบ่งไว้ตามการเข้าถึงข้อมูล โดยแต่ละคำสั่งจะมีรูปแบบของ Instructions Machine และ Opcode ดังแสดงในตารางที่ 4.4

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

4.14 โฟล์วชาร์ตแสดงการทำงานของ PLC

โฟล์วชาร์ตการทำงานของ PLC แบ่งได้เป็น 4 ส่วนดังนี้

1. โฟล์วชาร์ตการทำงานหลักของ PLC

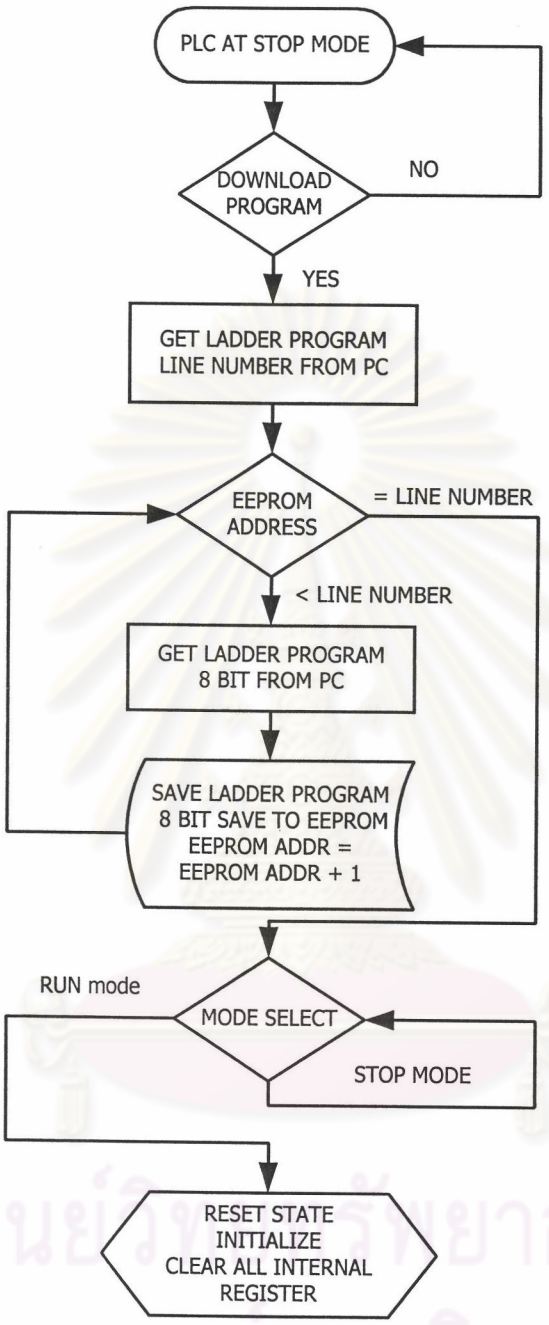


รูปที่ 4.8 โฟล์วชาร์ตการทำงานหลักของ PLC

การทำงาน

1. เริ่มตั้งแต่การเปิดไฟเข้า PLC (Power On) PLC จะเริ่มทำการกำหนดค่าเริ่มต้นให้ Register ต่างๆ (Initialize)
2. อ่านค่าจากหน่วยความจำโปรแกรมภายนอกที่เก็บโปรแกรมขั้นบันไดไว้ เพื่อมาบันทึกไว้ที่หน่วยความจำภายในส่วนโปรแกรม
3. จากนั้นมายังสถานะ Mode Select จะตรวจดูว่าผู้ใช้งานเลือกโหมดการทำงานเป็นโหมดหยุดหรือโหมดทำงาน
4. ถ้าผู้ใช้เลือกโหมดหยุดจะหยุดทำงาน หรือผู้ใช้สามารถโหลดโปรแกรมขั้นบันไดใหม่ได้ หรือผู้ใช้สามารถกลับมายังโหมดทำงานโดยปรับที่สวิตซ์เลือกโหมด
5. ถ้าผู้ใช้จะโหลดโปรแกรมใหม่ก็จะรับข้อมูลโปรแกรมขั้นบันไดจากคอมพิวเตอร์ที่แต่ละ 8 บิต จนครบแล้วกลับไปยังสถานะเริ่มต้นข้อ 1 ใหม่ รายละเอียดส่วนนี้แสดงในโฟล์วชาร์ตรูปที่ 4.9 ต่อไป
6. ถ้าผู้ใช้เลือกโหมดทำงาน PLC ก็จะทำการรีเซต Watchdog เพื่อแสดงว่าสถานะของ PLC ยังทำงานปกติอยู่
7. อ่านสถานะของ Input Relay เก็บไว้ในหน่วยความจำภายในส่วนข้อมูลพร้อมกำหนดให้ Register Program Counter (PC) เป็น 0 บอกให้รู้ว่าให้เริ่มทำงานโปรแกรมขั้นบันไดแรก และเป็นจุดเริ่มต้นของ Scan Time ด้วย
8. PLC จะอ่านโปรแกรมขั้นบันไดจากหน่วยความจำภายในส่วนโปรแกรมทีละคำสั่ง 16 บิต (Fetch)
9. เมื่อ PLC อ่านข้อมูลคำสั่งมาแล้วจะทำการตีความ (Decode) เพื่อให้รู้ว่าเป็นคำสั่งอะไร พร้อมกันนั้นก็เพิ่มค่า Program counter อีกหนึ่ง เตรียมพร้อมสำหรับการอ่านข้อมูลครั้งต่อไป
10. เมื่อ PLC รู้ว่าคำสั่งนั้นเป็นคำสั่งอะไรก็จะดำเนินการตามนั้น แล้วกลับไปทำงานตามข้อที่ 8 เพื่อทำงานคำสั่งอื่นต่อไป
11. แต่ถ้าข้อ 9 ตีความได้เป็นคำสั่ง END แสดงว่า PLC ได้ทำงานตามคำสั่งขั้นบันไดจนมาถึงคำสั่งสุดท้ายแล้ว ค่า Program Counter จะถูกกำหนดให้เป็น 00 อีกครั้งเพื่อให้ PLC กลับไปทำคำสั่งแรกใหม่ ก่อนที่ PLC จะเริ่มทำงานตามคำสั่งแรกใหม่จะส่งค่าข้อมูลหน่วยความจำภายในส่วนข้อมูลกลับมาแสดงผลที่คอมพิวเตอร์

2. โฟล์วชาร์ตการโหลดโปรแกรมลง PLC



รูปที่ 4.9 โฟล์วชาร์ตการโหลดโปรแกรมลง PLC

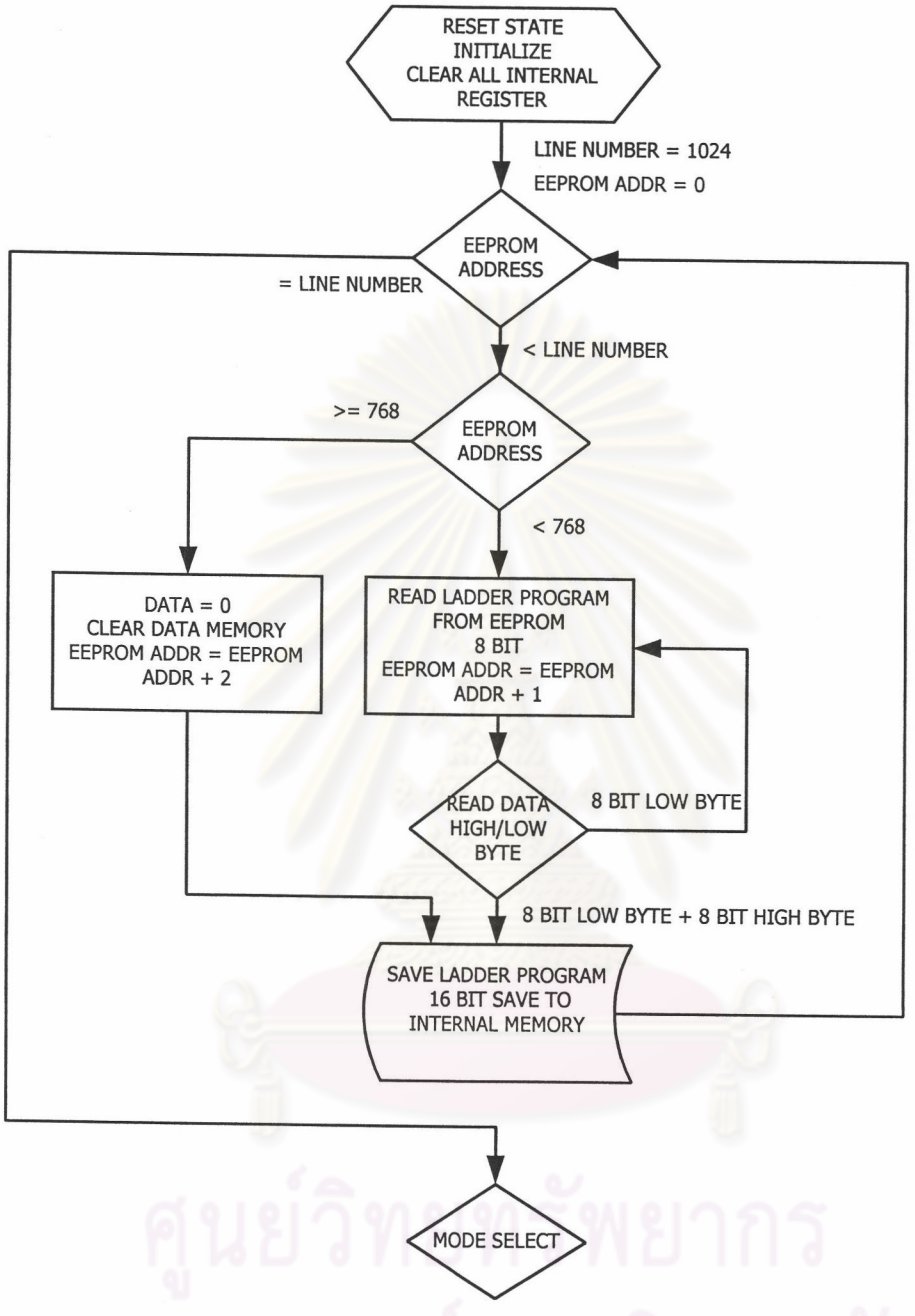
ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

การทำงาน

1. เริ่มต้นที่เมื่อ PLC อยู่ในโหมดหยุด
2. ผู้ใช้เลือกโหลดโปรแกรมหรือไม่ ถ้าไม่ก็จะอยู่ที่โหมดหยุด (Stop Mode)
3. เมื่อเลือกโหลดคอมพิวเตอร์จะส่งจำนวนบรรทัดของข้อมูลที่จะส่งมาก่อน
4. จากนั้นรอรับข้อมูลที่ละ 8 บิตจากคอมพิวเตอร์
5. บันทึกข้อมูลที่รับมาบันทึกลงหน่วยความจำโปรแกรมภายนอกทีละ 8 บิต แล้วเพิ่มค่าตัวอ้างตำแหน่งหน่วยความจำนี้อีกหนึ่ง เพื่อรอการบันทึกข้อมูลครั้งต่อไป
6. เมื่อข้อมูลโปรแกรมขั้นบันไดที่รับมาเท่ากับจำนวนบรรทัดของข้อมูลแล้ว ก็จะรอผู้ใช้เลือกโหมดการทำงานของ PLC ถ้าเลือกที่โหมดทำงาน PLC ก็จะทำการ Reset เตรียมเริ่มทำงาน แต่ถ้ายังไม่เลือกอยู่ที่โหมดหยุดตามเดิม PLC จะไม่ทำงานและโหลดโปรแกรมใหม่ไม่ได้เช่นกัน

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

3. โฟล์วชาร์ตการอ่านข้อมูลจาก EEPROM บันทึกลงหน่วยความจำข้อมูลภายใน



รูปที่ 4.10 โฟล์วชาร์ตการอ่านข้อมูลจาก EEPROM บันทึกลงหน่วยความจำข้อมูลภายใน

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

การทำงาน

1. เริ่มเมื่อมีการ Reset PLC ไม่ว่าจากการกดปุ่มรีเซต หรือจากการโหลดโปรแกรมเสร็จแล้วเปลี่ยนมาเป็นโหมดทำงาน จะกำหนดให้ตัวชี้ตำแหน่งของ EEPROM เป็น 0 (Addr EEPROM) และจำนวนบรรทัดที่จะบันทึก (LINE NUMBER) เท่ากับ 1024 เท่ากับหน่วยความจำภายในทั้งหมดในหน่วย Byte
2. ตรวจสอบว่า Addr EEPROM มีค่าเท่ากับ Line Number หรือยัง เป็นการหาจุดสิ้นสุดการบันทึก
3. ถ้า Addr EEPROM น้อยกว่า Line Number ต้องทำการตรวจสอบต่อไปว่า Addr EEPROM น้อยกว่าหรือมากกว่า 768 ซึ่งก็คือจำนวน Byte ของหน่วยความจำภายในส่วนโปรแกรม
4. ถ้า Addr EEPROM น้อยกว่า 768 แสดงว่าต้องอ่านโปรแกรมขั้นบันไดจาก EEPROM มาทีละ 8 บิต โดยอ่าน 8 บิตล่างก่อน ตามด้วย 8 บิตบน ประกอบกันให้ครบ 16 บิต การอ่านแต่ละครั้ง Addr EEPROM ถูกบวกเพิ่มทีละหนึ่งด้วย
5. เมื่อครบ 16 บิตก็ทำการบันทึกลงในหน่วยความจำข้อมูลภายในส่วนโปรแกรม หน่วยความจำภายในทั้งหมดอ่าน/เขียนข้อมูลครั้งละ 16 บิต
6. ถ้า Addr EEPROM มากกว่า 768 แสดงว่าจะมาถึงตำแหน่งของหน่วยความจำข้อมูลภายในส่วนข้อมูลแล้ว เนื่องจากต้องทำการ Clear ค่าหน่วย ความจำส่วนนี้ทุกครั้งเมื่อทำการรีเซต PLC จึงกำหนดข้อมูลให้เท่ากับ 0 เพื่อทำการบันทึก
7. ถ้า Addr EEPROM เท่ากับ 1024 แล้วแสดงว่าขบวนการนี้สิ้นสุดแล้ว PLC ก็จะไปยังส่วน Mode Select เพื่อรอให้ผู้ใช้เลือกโหมดการทำงานของ PLC ต่อไป

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

4. โฟล์วชาร์ตแสดงการส่งสถานะข้อมูลของ PLC ไปยังคอมพิวเตอร์

99	100	Data Address	Data Low Byte	Data High Byte	ZERO Correction

รูปที่ 4.11 รูปแบบการส่งข้อมูลหน่วยความจำภายในส่วนข้อมูลไปคอมพิวเตอร์ส่วนบุคคล

การส่งสถานะของหน่วยความจำภายในส่วนข้อมูลซึ่งมีทั้งหมด 128 ตำแหน่ง ได้กำหนดรูปแบบดังแสดงในรูปที่ 4.11 เริ่มต้นด้วยการที่ PLC ส่ง 99 และ 100 ไปก่อนเพื่อบอกว่ามีการเริ่มส่งข้อมูลแล้ว จากนั้น 8 บิตถัดไปก็คือ ตำแหน่งของหน่วยความจำนั้น 8 บิตถัดไปอีกคือข้อมูล 8 บิตล่าง ตามด้วยอีก 8 บิตบน และสุดท้ายตามด้วยการส่งข้อมูลอีก 8 บิตเพื่อแก้ไขข้อมูลที่ผิดพลาด

รูปที่ 4.12 โฟล์ชาร์ตแสดงการส่งสถานะข้อมูลของ PLC ไปยังคอมพิวเตอร์

การทำงาน

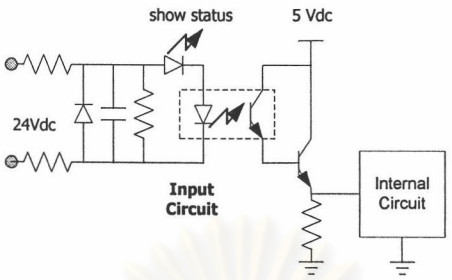
1. เมื่อ PLC ทำงานจนมาถึงคำสั่งสุดท้ายของโปรแกรมขั้นบันได จะทำการส่งสถานะของหน่วยความจำภายในส่วนข้อมูลไปยังเครื่องคอมพิวเตอร์ส่วนบุคคลเพื่อแสดงผล เริ่มต้นด้วยการตรวจสอบว่าสถานะการส่ง (Transmitter) พร้อมหรือยัง ถ้าไม่พร้อมเนื่องจากยังส่งข้อมูลเก่าไม่เสร็จก็ข้ามส่วนนี้ไปเริ่มทำตามโปรแกรมขั้นบันไดใหม่
2. ถ้าสถานะการส่งพร้อมก็เริ่มจากการอ่านข้อมูลหน่วยความจำภายในส่วนข้อมูลจำนวน 16 บิต ตามตำแหน่งตัวชี้ (Data Addr)
3. เนื่องจากต้องส่งข้อมูลเป็นลำดับ เริ่มต้นด้วยการส่ง 99 "01100011" เมื่อ count คือ 0 เพื่อบอกการเริ่มต้น หลังจากส่งเสร็จแล้วกำหนดให้ count เป็น 1
4. การส่งข้อมูลครั้งต่อไปเมื่อสถานะการส่งพร้อม คือ 100 "01100100" เมื่อ count คือ 1 หลังการส่งเสร็จแล้วกำหนดให้ count เป็น 2
5. การส่งข้อมูลครั้งต่อไปเมื่อสถานะการส่งพร้อมคือตำแหน่งของหน่วยความจำภายในส่วนข้อมูลส่งเสร็จแล้วกำหนดให้ count เป็น 3
6. การส่งข้อมูลครั้งต่อไปเมื่อสถานะการส่งพร้อมคือข้อมูล 8 บิตล่าง ส่งเสร็จแล้วกำหนดให้ count เป็น 4
7. การส่งข้อมูลครั้งต่อไปเมื่อสถานะการส่งพร้อมคือข้อมูล 8 บิตบน ส่งเสร็จแล้วกำหนดให้ count เป็น 5
8. การส่งข้อมูลครั้งต่อไปเมื่อสถานะการส่งพร้อมคือการแก้ไขข้อมูลที่ผิด 8 บิต ส่งเสร็จแล้วกำหนดให้ count เป็น 0 จากนั้นเพิ่มตัวอ้างตำแหน่งข้อมูลอีกหนึ่ง (Data addr = Data addr + 1) เพื่อเตรียมการส่งข้อมูลในตำแหน่งถัดไป

4.15 วงจรอินพุต/เอาต์พุต [19]

ภาคอินพุต (Input Module) ทำหน้าที่รับสัญญาณจากสวิทช์และเซนเซอร์ (Sensor) ต่างๆ ในระบบแล้วแปลงให้เป็นระดับสัญญาณไฟที่เหมาะสมที่ ซีพียูสามารถรับได้ สัญญาณไฟที่ป้อนเข้ามายังอินพุตอาจเป็นสัญญาณไฟ AC หรือ DC ก็ได้

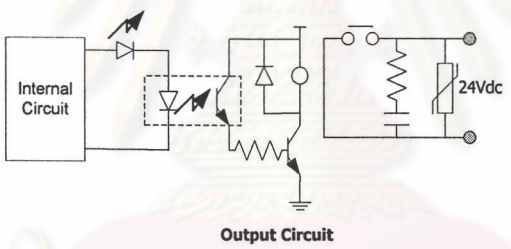
ภาคเอาต์พุต (Output Module) เป็นตัวรับสัญญาณจากซีพียูแล้วทำหน้าที่ขยายสัญญาณออกให้มีขนาดใหญ่พอที่จะขับอุปกรณ์ภายนอก เช่น Contactor และ Solenoid Valve เป็นต้น นอกจากนี้ยังทำหน้าที่แยกสัญญาณภายในและสัญญาณภายนอกออกจากกันเพื่อป้องกันระดับสัญญาณไฟสูงจากภายนอกเข้ามาทำความเสียหายแก่ซีพียู โดยปกติจะมีความสามารถในการขับโหลดด้วยกระแสประมาณ 2-5 Amp.

PLC ในวิทยานิพนธ์นี้ใช้เฉพาะอินพุตและเอาต์พุตกับสัญญาณดีซีมาตรฐาน 24 โวลท์ ซึ่งใช้กันมากในอุตสาหกรรม โดยทุกๆ ช่องสัญญาณจะมีหลอด LED แสดงสถานะ



รูปที่ 4.13 วงจรอินพุต

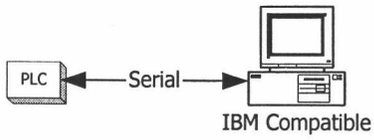
วงจรอินพุต ต้องลดสัญญาณรบกวนให้น้อยที่สุด โดยใช้คาปาซิเตอร์เป็นตัวกรองสัญญาณรบกวนออก ป้องกันระบบจากระดับแรงดันเกินจากภายนอก โดยใช้วิธีแยกระบบออกจากกันติดต่อกันด้วยแสง (Optical Isolation)



รูปที่ 4.14 วงจรเอาต์พุต

วงจรเอาต์พุต ต้องสามารถป้องกันอัตราการเปลี่ยนแปลงของแรงดัน (dv/dt) ซึ่งเกิดจากโหลดภายนอกได้ ป้องกันโดยต่อไดโอดคร่อมขดลวดของรีเลย์ และต้องแยกระบบออกจากเอาต์พุตเพื่อป้องกันระดับแรงดันไฟเกินจากภายนอกโดยใช้ Opto-Isolator

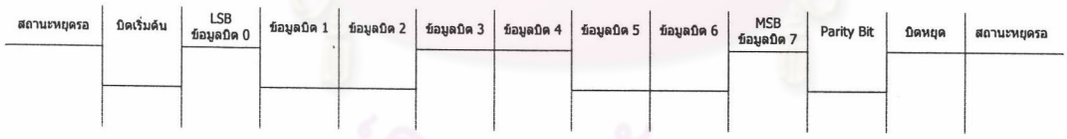
4.16 วงจรการสื่อสารกับคอมพิวเตอร์ส่วนบุคคล [21]



รูปที่ 4.15 การสื่อสารระหว่างคอมพิวเตอร์กับ PLC

เนื่องจาก PLC ที่พัฒนาขึ้นนี้ถูกออกแบบให้มีการโปรแกรมหรือการแสดงผลผ่านคอมพิวเตอร์ส่วนบุคคล ดังนั้นจึงต้องมีส่วนที่ติดต่อกันระหว่าง PLC และคอมพิวเตอร์ส่วนบุคคล ซึ่งใช้การสื่อสารข้อมูลแบบอนุกรม (Serial Interface)

การสื่อสารข้อมูลอนุกรมใช้แบบอะซิงโครนัสคือการรับและส่งข้อมูลไปในสายโดยไม่จำเป็นต้องมีสัญญาณนาฬิการ่วมด้วยเหมือนกับการรับส่งข้อมูลแบบซิงโครนัส แต่จะใช้การกำหนดค่าสัญญาณนาฬิกาทั้งภาครับและภาคส่งให้มีค่าเท่ากัน สัญญาณนาฬิกาที่ใช้เรียกอัตราการถ่ายทอดข้อมูล หรือโบด์เรต (baudrate) มีหน่วยเป็นบิตต่อวินาที (bit per second : bps) เลือกใช้ 9600 bps



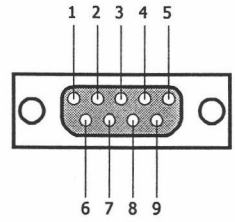
รูปที่ 4.16 รูปแบบการสื่อสารข้อมูลแบบอะซิงโครนัส

รูปแบบของข้อมูลที่ใช้ในการรับส่งแบบอะซิงโครนัสประกอบด้วย 4 ส่วนคือ

1. บิตเริ่มต้น (Start Bit) มีค่าเป็นศูนย์ขนาดหนึ่งบิต
2. บิตข้อมูลแบบอนุกรมจะมีขนาด 8 บิต โดยเริ่มที่บิตล่างก่อน
3. บิตตรวจสอบพาริตี้ (Parity Bit) จะมีขนาด 1 บิตหรือไม่มี
4. บิตปิดท้าย (Stop Bit) เลือกให้มีขนาด 2 บิต

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

ส่วนการเปลี่ยนระดับแรงดันของทีทีแอล จะถูกเปลี่ยนให้เป็นตามมาตรฐาน RS-232 โดยใช้ชิป MAX232 ระดับลอจิก "0" มีระดับแรงดัน +3 ถึง +12V ในขณะที่ลอจิก "1" มีระดับแรงดัน –3V จนถึง –12V เพื่อให้สามารถต่อเข้ากับพอร์ต COM1 ของคอมพิวเตอร์ได้

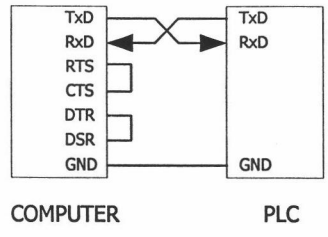


รูปที่ 4.17 คอนเน็กเตอร์อนุกรม 9 ขาหรือแบบ DB-9

คอนเน็กเตอร์ DB-9	คอนเน็กเตอร์ DB-25	ชื่อของสายสัญญาณ	ชนิดของสายสัญญาณ
1	8	Data Carrier Detect : DCD	อินพุต
2	3	Received Data : RxD	อินพุต
3	2	Transmitted Data : TxD	เอาต์พุต
4	20	Data Terminal Ready : DTR	เอาต์พุต
5	7	Signal Ground	-
6	6	Data Set Ready : DSR	อินพุต
7	4	Request to Send : RTS	เอาต์พุต
8	5	Clear to Send : CTS	อินพุต
9	22	Ring Indicator : RI	อินพุต

ตารางที่ 4.5 การจัดขาคอนเน็กเตอร์อนุกรมมาตรฐาน RS-232 แบบ DB-9 และ DB-25

มาตรฐานการเชื่อมต่อแบบ RS-232 จะใช้คอนเน็กเตอร์แบบ DB-25 ตัวผู้หรือ DB-9 ตัวผู้ ซึ่งคอนเน็กเตอร์แบบ DB-25 จะมีขาต่อใช้งานเพียง 9 เส้นเช่นเดียวกับคอนเน็กเตอร์แบบ DB-9 เนื่องจากขาอื่นๆ ที่เคยใช้งานในอดีต ปัจจุบันมีการใช้งานไม่มากนักจึงถูกยกเลิกไป แสดงรูปร่างและตำแหน่งขาดังรูป



รูปที่ 4.18 การเชื่อมสายระหว่างคอมพิวเตอร์กับ PLC

สำหรับการเชื่อมต่อคอมพิวเตอร์กับอุปกรณ์ภายนอกลูกศรในรูปที่ 4.15 แสดงถึงทิศทางของข้อมูลเป็นการเชื่อมต่อแบบ Null Modem ในลักษณะที่ใช้สายสัญญาณเพียง 3 เส้น โดยเส้นหนึ่งสำหรับส่งข้อมูลอีกเส้นสำหรับรับข้อมูล และเส้นสุดท้ายเป็นกราวนด์สำหรับรายละเอียดหน้าที่การทำงานในแต่ละขาของพอร์ตอนุกรม RS-232 มีดังนี้

- Receive Data : RD **หรือ** RxD ขานี้ใช้เพื่อรับสัญญาณอนุกรมเข้ามายังคอมพิวเตอร์โดยนำข้อมูลที่อ่านได้เก็บไว้ในรีจิสเตอร์บัฟเฟอร์
- Transmitted Data : TD **หรือ** TxD ขานี้ใช้เพื่อส่งข้อมูลออกจากคอมพิวเตอร์ โดยนำข้อมูลที่เก็บอยู่ในบัฟเฟอร์สำหรับส่งข้อมูลส่งออกไป
- Data Terminal Ready : DTR เป็นขาสัญญาณที่ส่งออกจากคอมพิวเตอร์เพื่อให้อุปกรณ์ปลายทางรับรู้ว่าต้องการติดต่อด้วย โดยขา DTR นี้จะต้องเชื่อมต่อกับขา DSR ของอุปกรณ์ปลายทาง และขา DTR ของอุปกรณ์ปลายทางจะต้องเชื่อมต่อกับขา DSR ของคอมพิวเตอร์ ถ้าใช้การเชื่อมต่อเป็นแบบ Null Modem ซึ่งใช้สายในการเชื่อมต่อเพียง 3 เส้น จะต้องต่อขา DTR และ DSR ของตัวมันเองเข้าด้วยกันและต้องต่อกับขา DCD ด้วยในกรณีที่โปรแกรมสื่อสารที่ใช้มีการตรวจจับสัญญาณพาห์
- Signal Ground : GND ขากราวนด์ของระบบ
- Data Set Ready : DSR ขานี้จะใช้คู่กับขา DTR เพื่อตรวจสอบการเชื่อมต่อกันระหว่างคอมพิวเตอร์กับอุปกรณ์ปลายทาง ซึ่งขา DSR นี้จะเป็นขาสำหรับรับข้อมูลจากภายนอกซึ่งถูกส่งมาจากขา DTR
- Request to Send : RTS เป็นขาสำหรับส่งสัญญาณร้องขอให้ทางอุปกรณ์ปลายทางส่งข้อมูลกลับมายังคอมพิวเตอร์ โดยขาที่รับสัญญาณ RTS ก็คือขา CTS ในกรณีที่ใช้การเชื่อมต่อแบบ Null Modem 3 สาย จะต้องเชื่อมต่อขา RTS และ CTS ของตัวมันเองเข้าด้วยกัน เพื่อจะให้การรับและส่งข้อมูลสามารถเกิดขึ้นได้ตลอดเวลา
- Clear to Send : CTS ขานี้จะคอยรับสัญญาณจากขา RTS เมื่อรับสัญญาณได้ข้อมูลที่ขานี้จึงถูกใช้เพื่อตรวจสอบอุปกรณ์ต่อพ่วงว่าพร้อมที่จะรับข้อมูลหรือไม่

วิทยานิพนธ์นี้เลือกการสื่อสารอนุกรมแบบอะซิงโครนัส โบด์เรต 9600 bps ข้อมูลแบบอนุกรม 8 บิต บิตปิดท้าย 2 บิต ไม่มีบิตตรวจสอบพาริตี้ ใช้คอนเน็กเตอร์แบบ DB-9 ระดับแรงดันทีทีแอลของ FPGA ถูกเปลี่ยนเป็นสัญญาณมาตรฐาน RS-232 โดยใช้ชิป MAX232 การเชื่อมสายระหว่างคอมพิวเตอร์กับ PLC ดังรูปที่ 4.15