

สารบัญ

Introductions	1
Basic PLC Operation	3
PLC Programing	6
Basic PLC Command Functions	16
Laboratory Excercies	24

Basic Programmable Logic Controller

Introductions

เอกสารชุดนี้จะเน้นให้นักศึกษาได้เข้าใจการทำงานของชุดควบคุมประเภทหนึ่งที่ใช้กันอย่างแพร่หลายในอุตสาหกรรม ชุดควบคุมนี้ก็คือ Programmable Logic Controllers หรือเรียกย่อๆ ว่า 'PLC'

ความหมายของ PLC ก่อนข้างจะทำงานตามชื่อที่ใช้เรียกอุปกรณ์นี้ กล่าวคืออุปกรณ์นี้จะเป็นชุดควบคุมที่ใช้คอมพิวเตอร์เป็นกลไกสำคัญในการควบคุมการทำงาน โดยชุดควบคุมนี้จะพิจารณาว่า input มีลักษณะเช่นไร จากเครื่องจะวิเคราะห์ตามตรรกะ (logic) ที่ผู้ใช้ได้กำหนดโปรแกรมไว้ และจากนั้นจะสั่งการให้มี output เพื่อไปควบคุมหรือ control อุปกรณ์ต่างๆ ตามที่เราต้องการ กล่าวโดยสรุปคือ PLC จะทำงานควบคุมอุปกรณ์ต่างๆ ตามตรรกะของสถานะนี้ โดยผู้ใช้สามารถที่จะกำหนดและปรับเปลี่ยนโปรแกรมที่สร้างขึ้นได้

PLC เริ่มพัฒนาครั้งแรกในช่วงปี ค.ศ. 1968 เพื่อตอบสนองความต้องการของอุตสาหกรรมยานยนต์ในประเทศสหรัฐอเมริกา และเริ่มติดตั้งเพื่อใช้งานจริงในภาคอุตสาหกรรมการผลิตในปี ค.ศ. 1969 ซึ่งในช่วงแรกนี้ logic ที่ใช้ยังไม่ซับซ้อนมากนักและจะเน้นไปที่ on/off controlled

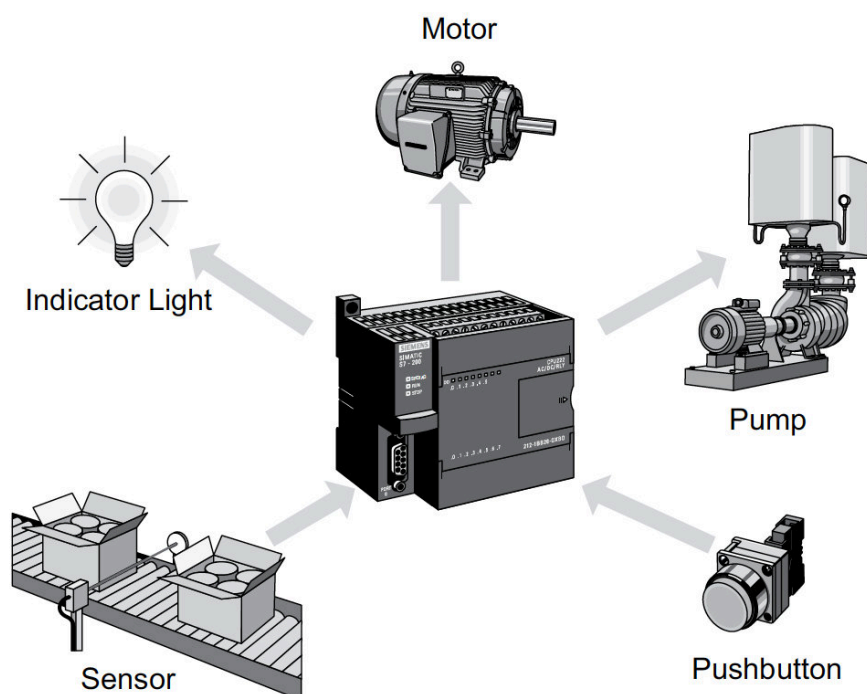
ในช่วงเวลาต่อมา PLC ก็ได้มีการพัฒนาขึ้น สามารถที่จะใช้ควบคุมอุปกรณ์ได้หลากหลายมากขึ้น สามารถที่จะส่งและรับสัญญาณที่มีค่า voltage ที่แตกต่างกันได้ ซึ่งทำให้ PLC ก้าวเข้าสู่ช่วง analog input/output ได้ ส่วนในช่วงยุคปี 1980's ได้มีการกำหนดมาตรฐานสำหรับ PLC ที่รู้จักกันในชื่อ manufacturing automation protocol (MAP) มีการลดขนาดของ PLC ให้เล็กลง นอกเหนือจากนั้นยังได้เริ่มมีการใช้ software ช่วยในการเขียน program บน PC ในลักษณะของสัญลักษณ์ หรือ symbolic programming ทำให้สามารถที่จะทำการโปรแกรมได้โดยไม่ต้องใช้อุปกรณ์พ่วงต่อที่เขียน code ลงไปบน PLC

ในยุคปี ค.ศ. 1990's เราได้พบเห็นการนำเสนอ protocol และ function ใหม่ๆ ที่ใช้งานกับ PLC ซึ่งการพัฒนาเหล่านี้เกิดจากการแข่งขันของบริษัทผู้ผลิต ที่พยายามที่จะให้อุปกรณ์ของตนเองใช้งานได้ตรงตามความต้องการของลูกค้าให้มากที่สุด แต่เพื่อให้การพัฒนา PLC ยังคงอยู่ในมาตรฐานที่ยอมรับกันได้ทั่วโลก จึงได้มีการกำหนด มาตรฐานขึ้นโดยล่าสุดได้มีการกำหนดมาตรฐาน "IEC 1131-3" ซึ่งพยายามที่จะรวมภาษาการเขียนโปรแกรม PLC ให้เป็นมาตรฐาน ซึ่งในปัจจุบันนี้ ภาษามาตรฐานของ PLC ได้แก่ function block diagrams, instruction lists, C และ structured text ซึ่งการที่ผู้ใช้จะเลือกใช้วิธีการเขียนอย่างไหนนั้นก็ขึ้นอยู่กับความเหมาะสมและความถนัดของผู้ใช้เป็นคนสำคัญ

จุดประสงค์หลักของเอกสารฉบับนี้เป็นการทำขึ้นมาเพื่อประกอบการปฏิบัติการ ซึ่งมีจุดมุ่งหมายหลักคือ

- * ให้นักศึกษาเข้าใจองค์ประกอบและหลักการทำงานของ PLC
- * ให้นักศึกษามีความเข้าใจการเขียนชุดคำสั่งที่จะใช้งานร่วมกับ PLC
- * ให้นักศึกษามีความสามารถในการเชื่อมต่อสายสัญญาณทั้ง input และ output จาก PLC ได้
- * เนื่องจากมีผู้ผลิตอยู่หลายรายที่ได้ผลิต PLC เพื่อใช้สำหรับอุตสาหกรรมการผลิตอัตโนมัติ ซึ่งแต่ละแบบก็จะมี ความแตกต่างกันบ้างในรายละเอียด ดังนั้นในเอกสารนี้ เราจะทำการบอกถึงความแตกต่างระหว่างบริษัทผู้ผลิต ชุดคำสั่งที่สำคัญและข้อมูลพื้นฐานของ PLC แต่ละผู้ผลิต ซึ่งผู้ผลิตที่เราจะกล่าวถึงในที่นี้ จะประกอบด้วย OMRON, SIEMENS, MITSUBISHI และ KEYENCE

สิ่งที่ทำให้ PLC เป็นที่นิยมอย่างกว้างขวางในแวดวงอุตสาหกรรมอัตโนมัติก็คือ อุปกรณ์นี้มีความทนทาน ทนต่อสภาพอุณหภูมิที่เปลี่ยนแปลงไปได้ดี มีชุด input/output บรรจुरวมเข้าอยู่กับชุดควบคุม และการเขียนชุดหรือภาษาที่ใช้ นั้นง่ายต่อความเข้าใจ รูปที่ 1 แสดงถึงอุปกรณ์ต่างๆที่เชื่อมต่อกับ PLC



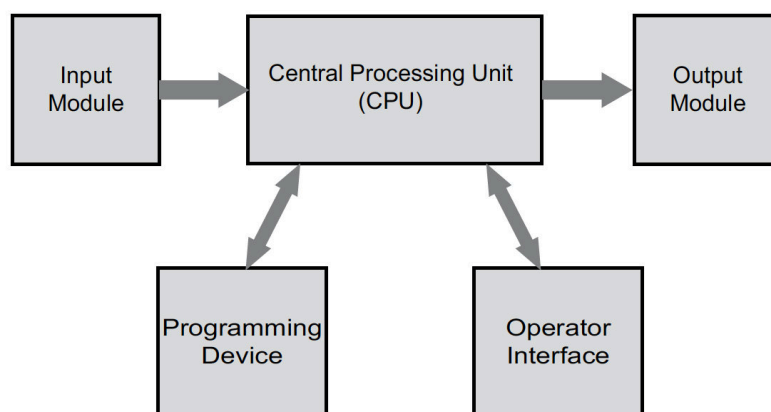
รูปที่ 1 อุปกรณ์ที่เชื่อมต่อกับ PLC เพื่อใช้ในการควบคุม

Basic PLC Operation

ในการทำงานของ PLC จะมีส่วนประกอบอยู่หลายส่วนที่ทำงานสอดประสานกัน สำหรับส่วนประกอบหลักของ PLC จะประกอบด้วยส่วนหรือ module ต่างๆ ดังต่อไปนี้

- Input modules
- Central Processing Unit (CPU),
- Output Modules
- Programming device

ซึ่งการเชื่อมโยงการทำงานของอุปกรณ์ต่างๆ แสดงในรูปที่ 2 และนอกเหนือจากส่วนประกอบหลักนี้แล้ว PLC ยังมีส่วนประกอบที่ทำหน้าที่เชื่อมโยงหรือประสานการทำงานของเครื่องร่วมกับผู้ใช้ หรือ Operator Interface ร่วมอยู่ด้วย สำหรับรายละเอียดและหน้าที่หลักของส่วนประกอบต่างๆ ที่กล่าวมาแล้วจะมีดังต่อไปนี้



รูปที่ 2 ส่วนประกอบของ PLC

Input Module

ลักษณะและประเภทของ input module นี้จะแตกต่างกันออกไปตามชนิดของข้อมูลที่จะส่งให้กับ PLC ที่เราพบเห็นมากที่สุดจะเป็น DC voltage digital input หรือ discrete input คือจะเป็นการจ่ายไฟที่ voltage ค่าหนึ่ง อาจเป็น 5 หรือ 24 volt หรือไม่มีการจ่ายไฟให้ ซึ่งก็หมายความว่าเราได้ logical input เป็น ON หรือ OFF จาก input module นี้ อย่างไรก็ตามในปัจจุบันเราเริ่มพบเห็น PLC ที่รองรับ analog input มากขึ้น นั่นคือสามารถที่จะตรวจจับปริมาณไฟที่เข้ามาที่ช่องสัญญาณว่ามีปริมาณเท่าใดแทนที่จะตรวจจับเพียงแต่ว่ามีหรือไม่มีไฟมาที่ช่องสัญญาณ ซึ่งจะทำให้สามารถตรวจจับอุณหภูมิ ความดัน หรือปริมาณอื่นๆ ได้ ซึ่งทำให้ระบบการควบคุมจะสมบูรณ์และซับซ้อนมากขึ้นได้

CPU

หน่วยประมวลผลกลางนี้ มีหน้าที่ประมวลผลที่ได้รับ โดยจะเริ่มจากการที่รับเอาข้อมูลจาก input module ทั้งหมดเข้ามาพิจารณา จากนั้นก็จะมีการประมวลผลตามตรรกะที่กำหนดไว้โดยโปรแกรม จากนั้นหน่วยประมวลผลกลางนี้จะทำหน้าที่ตัดสินใจ ทำการสั่งการ output module ให้เปิดหรือปิด ส่งสัญญาณ output ต่างๆ

สำหรับในหน่วยประมวลผลกลางนี้จะทำหน้าที่เหมือนคอมพิวเตอร์ทั่วไป คือจะมีหน่วยย่อยๆ ประกอบอยู่ด้วยเช่นหน่วยความจำทั้ง ROM และ RAM นอกเหนือจากนั้นยังมีหน้าสัมผัสแบบต่างๆ เช่นชุด control relay, spacial relay, contacts และอื่นๆ อีกมาก แต่เราจะไม่ขอลำถึงในรายละเอียดในขั้นนี้

Output Modules

หน้าที่หลักของส่วน output modul นี้ จะทำหน้าที่เปลี่ยนสัญญาณที่ได้รับในการตัดสินใจของ CPU ให้กลายเป็นสัญญาณควบคุมที่จะส่งออกไปทางช่องสัญญาณของ output module นี้ สำหรับช่องสัญญาณนี้อาจจะเป็นช่องสัญญาณแบบ digital คือมีเฉพาะสัญญาณ ON หรือ OFF เท่านั้น หรือจะเป็นสัญญาณแบบ analog ที่สามารถปรับค่าต่างๆ ให้เข้าหรือออกจากอุปกรณ์นี้ได้ตามความต้องการ

Programming Device

อุปกรณ์เขียนและแก้ไขโปรแกรม หรือ Programming Device นี้เป็นอุปกรณ์ที่มีหน้าที่ในการเขียนชุดคำสั่งที่ผู้ใช้งานต้องการเข้าสู่เครื่อง PLC หรือทำหน้าที่ตั้งโปรแกรมที่บรรจุอยู่ใน CPU ของ PLC ออกมาสู่เครื่องนี้ นอกเหนือจากนี้ในบางรุ่นยังสามารถที่จะทำหน้าที่ตรวจสอบสถานะการทำงานของ PLC ว่าสามารถที่จะทำงานได้ตามที่ผู้เขียนโปรแกรมต้องการหรือไม่

PLC Processing

เมื่อมีส่วนประกอบหลายส่วนใน PLC มันจึงมีการกำหนดกระบวนการการทำงานของ PLC เอาไว้ ขั้นตอนการทำงานของ PLC นั้นจะมีอยู่ 2 mode คือ

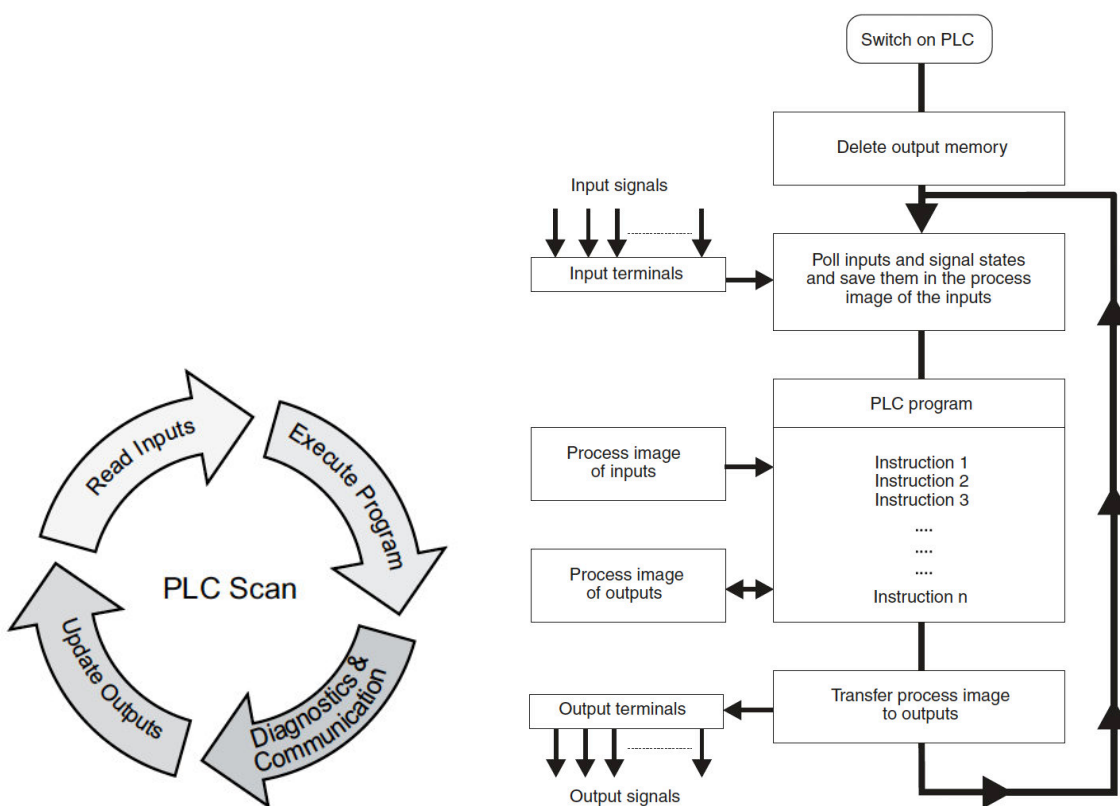
- * Edit mode หมายถึงโหมดที่ผู้ใช้จะทำการเขียนโปรแกรมหรือแก้ไขโปรแกรม ซึ่งในระหว่างที่อยู่ในโหมดนี้เครื่องจะไม่มี การรับข้อมูลจาก input มาทำการวิเคราะห์ หรือส่งข้อมูลใดๆ ออกไปทางช่อง output
- * Run Mode ในโหมดที่เครื่องจะทำงานตามที่เราได้โปรแกรมไว้ โดยระหว่างที่อยู่ในโหมดนี้ เราไม่สามารถที่จะแก้ไขโปรแกรมได้ และสำหรับในบางผู้ผลิตจะมี monitor mode เพิ่มขึ้นมา

โหมดนี้จะเป็นส่วนพิเศษของ run mode โดยเราจะเพิ่มการแสดงผล สถานะต่างๆ ของ PLC ในขณะนั้นให้เรารับไปพร้อมกับการทำงานของเครื่องไปด้วย

เมื่ออยู่ใน run mode การทำงานของ PLC เริ่มจากเปิดเครื่องจะมีขั้นตอนดังนี้

1. ลบคำสั่งเดิมที่ output module
2. ดึงข้อมูลจาก input module เข้ามาและประมวลสัญญาณทั้งหมดส่งให้ CPU
3. CPU รับข้อมูล Input แล้วเปรียบเทียบกับตรรกะของโปรแกรม แล้ววิเคราะห์ว่าจะต้องดำเนินการเช่นไรบ้าง จากนั้นก็จะส่งสัญญาณตามข้อกำหนดที่วิเคราะห์ให้กับโปรแกรม ส่งไปให้กับ output module
4. Output module รับข้อมูลจาก CPU จากนั้นจะแปลงสัญญาณคำสั่งไปเป็นสัญญาณควบคุมการทำงานของชุด output ตามที่กำหนด
5. เมื่อครบรอบการทำงานนี้จะถือว่า PLC ทำงานครบหนึ่งรอบหรือหนึ่ง scan จากนั้นก็จะเริ่มรอบทำงานต่อไป โดยจะกลับเข้าไปอ่านสถานะของ input มามีสถานะเป็นอย่างไร

ลักษณะการทำงานของ PLC นี้ แสดงในรูปที่ 3



รูปที่ 3 PLC Scan

Addressing

ช่องสัญญาณ input หรือ output แต่ละช่องจะต้องมีชื่อหรือหมายเลขกำกับเพื่อที่จะได้ทราบถึงความแตกต่างรวมทั้งอุปกรณ์ภายในที่ประเภท relay ต่างๆ ที่อยู่ภายใน CPU ล้วนแต่จะต้องมีหมายเลขหรือชื่อกำกับด้วยกันทั้งสิ้น ซึ่งเราจะเรียกการกำหนดตัวเลขหรือตัวอักษรนั้นว่า addressing ซึ่งการกำหนดนี้จะมีลักษณะของการกำหนด เป็นหมายเลขหรือตัวอักษรที่แสดงถึงช่องสัญญาณ หรือแสดง channel จากนั้นจะตามด้วยหมายเลข bit หรือหมายเลขอุปกรณ์ที่อยู่ในช่วงสัญญาณนั้น ยกตัวอย่างเช่นเราอาจกำหนดหมายเลขอุปกรณ์แบบ input007 ก็หมายถึงว่าเป็นช่องสัญญาณในด้าน input และเป็นอุปกรณ์หมายเลข 007 ในช่องสัญญาณนั้น

เนื่องจากมีผู้ผลิต PLC หลายบริษัทที่ใช้อยู่ในอุตสาหกรรมในประเทศของเรา และผู้ผลิตแต่ละรายจะมีการกำหนดรายละเอียดของ addressing นี้แตกต่างกันออกไป นอกเหนือจากนี้จำนวน address ของผู้ผลิตรายเดียวกันแต่รุ่น PLC ที่แตกต่างกันออกไปก็อาจจะแตกต่างกัน หรือมีจำนวนไม่เท่ากัน สำหรับ addressing ของผู้ผลิตทั้งสี่บริษัทในเบื้องต้นจะมีการกำหนด จะเป็นไปตามตารางที่ 1 นี้ ซึ่งรายละเอียดของแต่ละผู้ผลิต จะต้องพิจารณาจากคู่มือของเครื่องอีกครั้งหนึ่ง

INSTRUCTION	OMRON	MITSHUBISHI	KEYENCE	SIEMENS
Input Address	000.00, 000.01,... 007.15	X000, X001,...	0000, 0001,...,00415	I0.0, I0.1, ...,I1.6,...
Output Adress	010.00, 010.01,... 017.15	Y000, Y002,..	0500, 0501,..., 0915	Q0.0, Q0.2, ..., Q1.6,...
Timer	TIM000, TIM001,...	T000, T001,...	TMR000&T000,...	
Counter	CNT000, CNT001,...	C000, C001, ...	C000, C001,...	
Internal Relay	00800, 00801,...09915	M000, M001, M003,...		

ตารางที่ 1 เปรียบเทียบการกำหนด addressing ของแต่ละผู้ผลิต

ข้อมูลที่อยู่ในตารางที่ 1 นั้นถือเป็นข้อมูลพื้นฐานเท่านั้น ในรายละเอียดของแต่ละผู้ผลิตก็จะมี ความแตกต่างกันไป และในชุดคำสั่งอื่นๆ ก็แตกต่างกันออกไปอีก ซึ่งเราจะได้กล่าวถึงในขั้นตอนการสั่งชุดคำสั่งต่อไป

PLC Programing

การจะโปรแกรม PLC นั้นในยุคแรกอาจมีวิธีการไม่มากนัก แต่ในยุคหลังนี้มีวิธีการให้เลือกเขียนได้หลายวิธีตามที่เราได้กล่าวไว้ก่อนหน้านี้แล้ว แต่ในเอกสารที่เราจะกล่าวถึงต่อไปนี้จะให้ความสำคัญหลักเพียงสองประเภทคือการเขียน Lader Diagram และการเขียนโปรแกรม code ซึ่งสำหรับผู้ ที่คุ้นเคยกับกาเขียนวงจรไฟฟ้าจะเรียนรู้การเขียน lader มากกว่าและสำหรับผู้ผลิตแต่ละรายแม้จะมีการกำหนด addressing ที่แตกต่างกัน แต่ ladder diagram จะมีความคล้ายคลึงกันมาก สำหรับแบบที่ 2 จะเป็นการเขียน induction code ซึ่งวิธีการนี้เป็นการใช้ตัวหนังสือทั้งหมดด้วยวิธีการนี้ผู้เขียนจะ

สามารถเขียนโปรแกรมด้วยใช้ programming console ได้ แก่ไขได้ง่าย สะดวกและรวดเร็ว ข้อเสียคือ ในโปรแกรมที่ซับซ้อนต้องอาศัยผู้ที่มีความเชี่ยวชาญในการแก้ และแต่ละผู้ผลิตจะมีการกำหนดวิธีการเขียน instruction code ที่แตกต่างกันออกไปโดยขึ้นเชิง ลักษณะการเขียนโปรแกรมทั้งสองแบบเราจะกล่าวถึงต่อไป

Symbol for Ladder Diagram

การเขียน ladder diagram เป็นการเขียนแผนผังกำหนดการทำงานของ PLC ซึ่งถ้าพิจารณาแล้วก็เป็นการพัฒนาจากการเขียนแผนภาพวงจรทางไฟฟ้านั่นเอง เพียงแต่การเขียนจะมีการจัดระเบียบมากกว่า ทำให้มีรูปแบบเป็นขั้นบันได เราจึงเรียกมันว่า ladder diagram สัญลักษณ์ที่เราจะใช้กับ ladder diagram จะประกอบด้วย contact, coil และ box เป็นหลัก ซึ่งรายละเอียดของแต่ละตัวเป็นดังนี้คือ

Contact

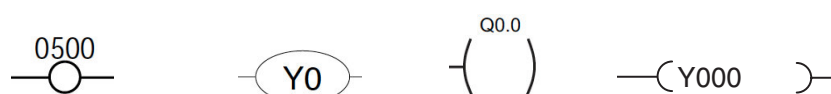
จะแสดงเป็นหน้าสัมผัส เหมือนกับการเขียนวงจรไฟฟ้าทั่วไป ซึ่งจะประกอบด้วยหน้าสัมผัส 2 แบบคือ Normally Closed และ Normally Open ซึ่งจะมีสัญลักษณ์ตามรูปที่ 4 ซึ่ง Relay แต่ละตัวที่เราเรียกใช้เราอาจจะเลือก ที่เป็นแบบ NO หรือ NC หรือจะเรียกใช้ทั้งสองแบบในวงจรเดียวกันก็ได้ สำหรับในการเขียนโปรแกรมนั้น หน้าสัมผัสจะต้องมีหมายเลข address ของหน้าสัมผัสนั้นประกอบอยู่ด้วย ซึ่งจากรูปที่ 4 จะเป็นการกำหนดว่าเป็นหน้าสัมผัสของ input relay แบบปกติเปิด X0 และหน้าสัมผัส input relay แบบปกติปิด X1 ซึ่งเป็นการกำหนดที่ใช้กับเครื่องของบริษัท Mishubishi



รูปที่ 4 Contact ใน PLC ตามข้อกำหนดของ Mitshubishi

Coil

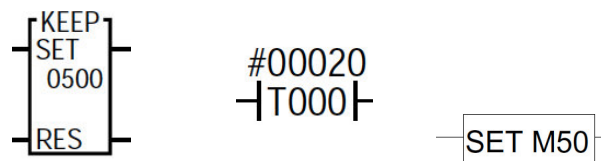
เป็นลักษณะของ relay ที่อาจจะเป็น output หรือ relay พิเศษ อื่นๆ ก็ได้ ซึ่ง coil แต่ละอัน ก็จะมี contact ที่เป็นของ coil นั้นด้วยและ contact ก็จะสามารถเป็นได้ทั้งแบบ NO และ NC สำหรับสัญลักษณ์ของ coil ที่ใช้กันอยู่มีสามแบบ ตามที่แสดงในรูปที่ 5 ซึ่งก็จะเหมือนกับหน้าสัมผัส คือจะต้องมีหมายเลขหรือตัวอักษรกำกับ address ของ coil และ relay นั้นด้วย



รูปที่ 5 สัญลักษณ์ของ coil ที่นิยมใช้สามแบบตามบริษัทผู้ผลิต

Box

กล่องหรือ box นี้จะเป็นสัญลักษณ์แทน relay ประเภทพิเศษแบบต่างๆ ที่เราจะต้องมีการกำหนดค่าเพิ่มเติมให้กับ relay เหล่านั้น ซึ่งอาจจะเป็น timer relay ซึ่งต้องมีการกำหนดค่าเวลา หรือ counter relay ที่ต้องมีการกำหนดค่าจำนวนเหตุการณ์ที่จะนับลงไปด้วย สำหรับ box นี้จะมีลักษณะแตกต่างกันออกไป ตามแต่ลักษณะของผู้ผลิต สำหรับกรณีทั่วไปแล้วจะแสดงไว้ในรูปที่ 6 ซึ่งใน Box นี้จะมีทั้งคำสั่งหรือ instruction เช่นบอกว่าเป็น timer และมี Operant ซึ่งจะบอกตำแหน่งของ Relay ระยะเวลาที่ให้หน่วงเวลา เป็นต้น



รูปที่ 6 สัญลักษณ์ของ Box ในรูปแบบต่างๆ

Internal and Spatial Relays

นอกเหนือจากหน้าสัมผัสที่เป็น input relay และ output coil แล้ว ใน PLC ยังมี relay และ coil อื่นอีก โดยอุปกรณ์เหล่านี้จะเป็นอุปกรณ์ที่อยู่ภายใน CPU ไม่สามารถที่จะต่อออกสู่ภายนอกได้จริง อุปกรณ์เหล่านี้ก็เช่น Timer, Counter และ Internal Relay อื่นๆ ซึ่งการกำหนด address ของอุปกรณ์เหล่านี้จะขึ้นอยู่กับผู้ผลิตแต่ละราย เช่นของบริษัท OMRON เราจะให้ relay หมายเลข 008.00, 008.01,... หรือสำหรับของบริษัท Mitshubishi จะใช้ M000, M001,... แทน internal relay เหล่านี้ สำหรับจำนวนและหมายเลขของ internal relay ในแต่ละรุ่นนั้นขอให้ผู้ผู้ใช้งานตรวจสอบกับคู่มือการใช้งานก่อนที่จะทำการเขียนโปรแกรม

PLC Programming Instruction

ตามที่กล่าวมาแล้ว การเขียนโปรแกรมของ PLC มีสองแบบหลักๆ คือ Ladder Diagram และ instruction code ในหัวข้อนี้เราจะยกตัวอย่างการเขียนโปรแกรมควบคุมการทำงานของ PLC โดยในเบื้องต้นนี้เราจะยกตัวอย่างพื้นฐานแบบง่ายๆ ก่อนและทำการเปรียบเทียบในการเขียนโปรแกรมทั้งสองแบบเพื่อการเปรียบเทียบเพื่อให้นักศึกษาเข้าใจหลักการเบื้องต้น สำหรับกระบวนการเขียนโปรแกรมจะเป็นดังนี้

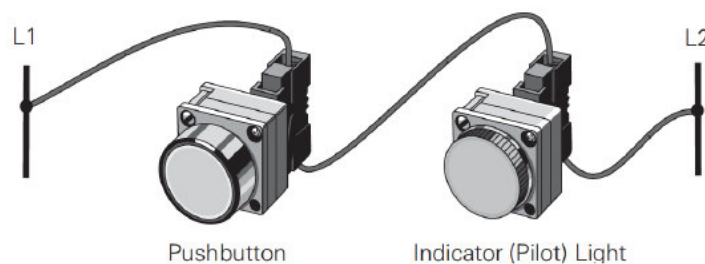
1. ขั้นแรกผู้ผู้กำหนดความต้องการในการทำงานของโปรแกรมโดยเริ่มจากการเขียน ladder diagram ขึ้นมาก่อน โดยมีการกำหนดหมายเลขของ relay และการเชื่อมต่อสายสัญญาณควบคุมให้ครบถ้วน โดยทั่วไปการเขียน diagram นี้จะเหมือนกันไม่ว่าเราจะใช้ PLC ของผู้ผลิตรายใดก็ตาม จะมีข้อยกเว้นในกรณีการเขียน Box ของแต่ผู้ผลิตอาจแตกต่างกัน

2. ขั้นที่สองผู้ใช้จะนำ ladder diagram นั้นมาเขียนเป็น instruction code ซึ่งลักษณะการเขียน instruction ซึ่งโดยภาพรวมจะมีข้อกำหนดที่คล้ายกันในแต่ละผู้ผลิต แต่อาจมีความแตกต่างกันบ้างในการกำหนดเรียกชื่อและลักษณะของคำสั่งเฉพาะ โดย instruction code นี้จะส่งเข้าสู่ PLC ทาง console unit และในการกลับกันเราสามารถที่จะ dowload program ที่อยู่กับ PLC มาที่ console unit เพื่อการตรวจสอบได้

อย่างไรก็ตามในปัจจุบันนี้การเขียนโปรแกรมนั้นบริษัทผู้ผลิตมักมีโปรแกรมคอมพิวเตอร์เพื่อให้ผู้ใช้เขียน ladder diagram ลงไปในโปรแกรม จากนั้นจะสามารถ upload ส่วนของ ladder diagram นั้นเข้าสู่ CPU ของ PLC โดยโปรแกรมจะทำหน้าที่เปลี่ยน ladder diagram เป็นภาษาที่ CPU เข้าใจได้โดยไม่จำเป็นต้องแปลง ladder diagram เป็น instruction code ผ่านทาง console unit เพื่อส่งผ่านข้อมูลให้กับ PLC ซึ่งเป็นขั้นตอนที่ใช้กันในสมัยก่อน

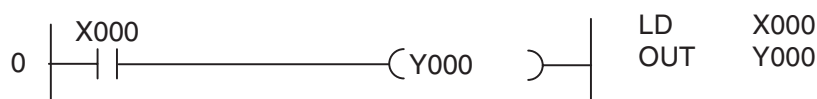
LOAD and OUT Insruction

ในการเริ่มการเขียนการควบคุมแต่ละสายของชุดควบคุมเราจะเรียกว่า line หรือ circuit หรือ Network ตามแต่ละบริษัท ถ้าเราต้องการเขียนโปรแกรมเพื่อควบคุมการปิดเปิดหลอดไฟหนึ่งดวงโดยใช้สวิทช์หนึ่งอัน ตามรูปที่ 7 (a) หรือที่เราเรียกว่าการต่อสายจริงหรือ Hard Wire



รูปที่ 7 (a) การต่อสายไฟฟ้าแบบตรง (hard wire)

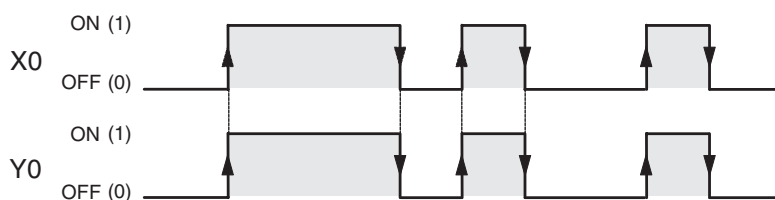
จากรูปข้างบนนี้เราสามารถที่จะเขียน logic ในการควบคุม PLC ดังที่แสดงในรูปที่ 7(b) ซึ่งเป็นการกำหนด addressing ตามบริษัท Mitshubishi



รูปที่ 7 การควบคุมปิดเปิดหลอดไฟ และการเขียน Ladder Diagram

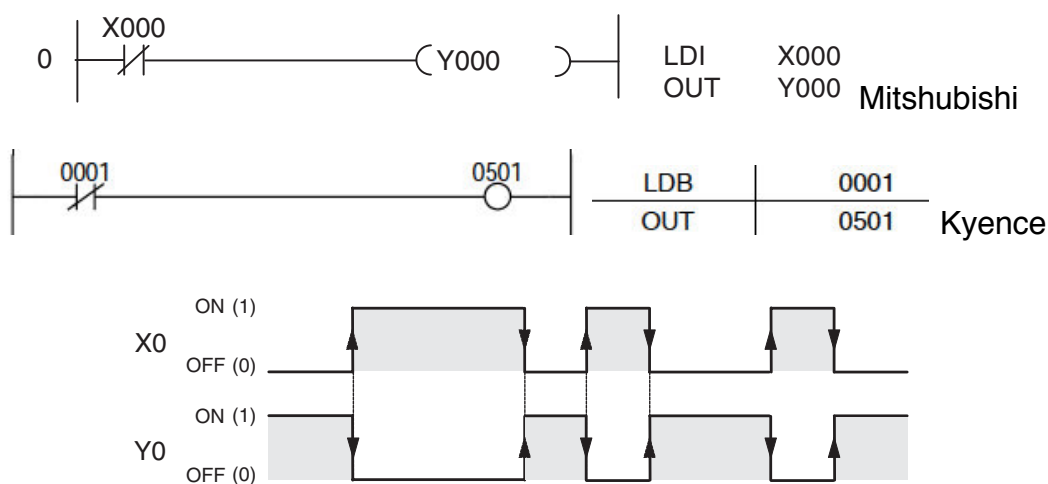
จากในรูปที่ 7(b) เราสามารถที่จะนำมาเขียนเป็น instruction code ได้เป็นสองบรรทัด โดยในบรรทัดแรกจะเป็นคำสั่ง LOAD วึ่งเป็นคำสั่งที่เราใช้กำหนดให้เป็นการเริ่ม line หรือ circuit ใหม่ จากนั้น address X000 จะหมายถึง input relay หมายเลข 000 ลำดับต่อไป เราสั่งคำสั่ง out ซึ่งจะเป็นการสิ้นสุด ของ circuit นั้น และสั่งให้สิ้นสุดที่ output relay หมายเลข 000

ดังนั้นเมื่อเราทำการให้โปรแกรมทำงาน ถ้าเรา ON หรือ OFF input relay X000 ก็จะทำให้ output relay Y000 เกิดการ ON หรือ OFF ไปด้วย ดังที่แสดงในรูปที่ 8 ซึ่งเรานิยมเรียกแผนภาพในรูปที่ 8 นี้ว่า timing diagram หรือ logical diagram หรือแผนภาพการตอบสนอง ตามแต่ละผู้ผลิต



รูปที่ 8 การตอบสนอง ตามโปรแกรมในรูปที่ 7

ในการ load relay นั้นเราสามารถที่จะเรียกใช้ได้ทั้งแบบ NO หรือ NC ซึ่งถ้าจากโจทย์ที่กล่าวถึงมาก่อนหน้านี้ ถ้าเราจะเรียกหน้าสัมผัส NC เราจะใช้คำสั่ง load และกำหนดว่าเป็นหน้าสัมผัสตรงกันข้ามกับ load ปกติ ซึ่งอาจจะใช้คำสั่ง LOAD NOT (OMRON), LOAD INVERSE หรือ LDI (MITSHUBISHI), LOAD BAR หรือ LDB (Kyence) การเรียกหน้าสัมผัสที่เป็นแบบปกติปิดมาใช้นั้นต่อละบริษัทจะเรียกแตกต่างกันออกไป



รูปที่ 9 การใช้คำสั่ง Load NC

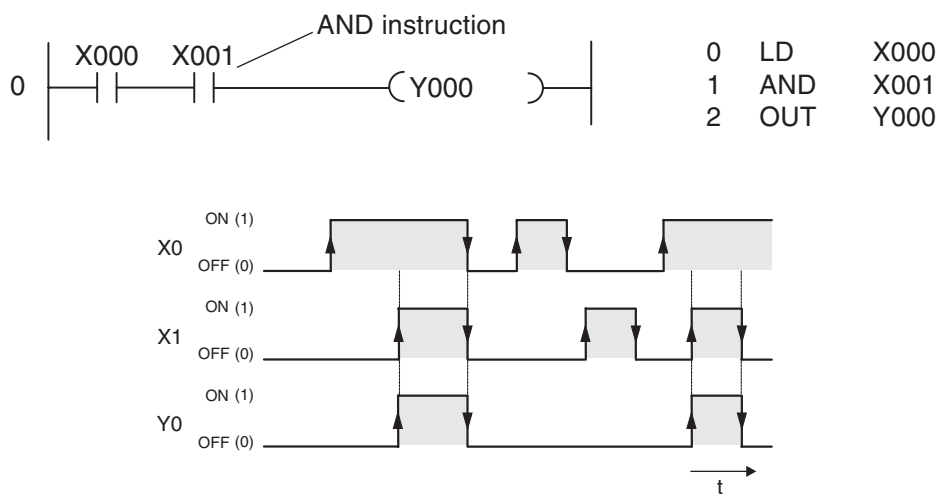
เราสามารถที่จะกล่าวโดยสรุปในการเรียกหน้าสัมผัสแบบปกติปิดมาเชื่อมต่อของผู้ผลิตรายต่างๆ เป็นดังนี้

- MITSHUBISHI จะใช้คำสั่งเป็น Inverse ซึ่งมักจะใช้ตัวย่อ I เช่นจากการ load หน้าสัมผัสปกติเปิดจะใช้ LD แต่ถ้าเรียกหน้าสัมผัสปกติปิดจะใช้ LDI
- KEYENCE จะใช้คำสั่งเป็น BAR ซึ่งมักจะใช้ตัวย่อ B เช่นจากการ load หน้าสัมผัสปกติเปิดจะใช้ LD แต่ถ้าเรียกหน้าสัมผัสปกติปิดจะใช้ LDB

- OMRON จะใช้คำสั่งเป็น NOT เช่นจากการ load หน้าสัมผัสปกติเปิดจะใช้ LD แต่ถ้าเรียก หน้าสัมผัสปกติปิดจะใช้ LD NOT

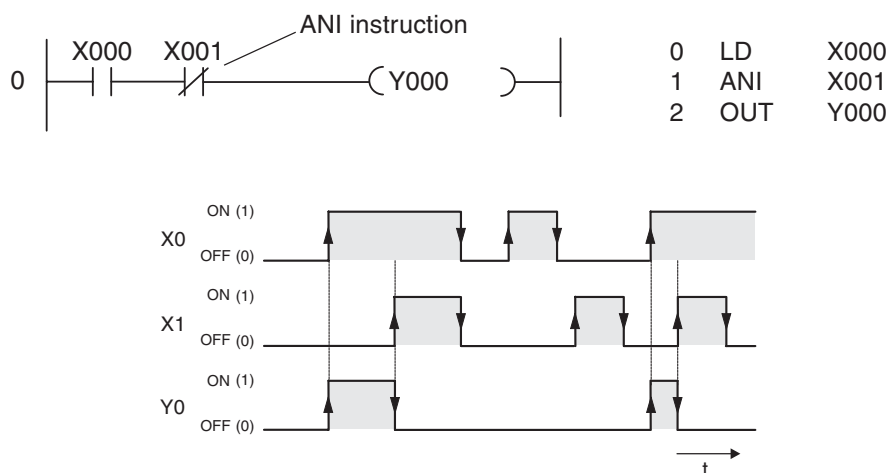
AND & AND NOT Instruction

เมื่อเราเชื่อมต่อหน้าสัมผัสตั้งแต่สองชุดขึ้นไป การที่เรานำหน้าสัมผัสมาต่อกันแบบอนุกรม ชุดคำสั่งที่เราใช้จะเป็นคำสั่ง AND หรือ AND NOT ซึ่งบางผู้ผลิตจะเรียกหน้าสัมผัสปกติปิดนี้เป็น AND BAR หรือ ANB หรือ AND INVERSE หรือ ANI เป็นต้น สำหรับ AND instruction แสดงในรูปที่ 10 แสดง ladder diagram และ line instruction code รวมถึงการตอบสนองของโปรแกรม จะเห็นว่า output Y0 จะ on ก็ต่อเมื่อหน้าสัมผัสทั้งสองต้อง on ด้วย



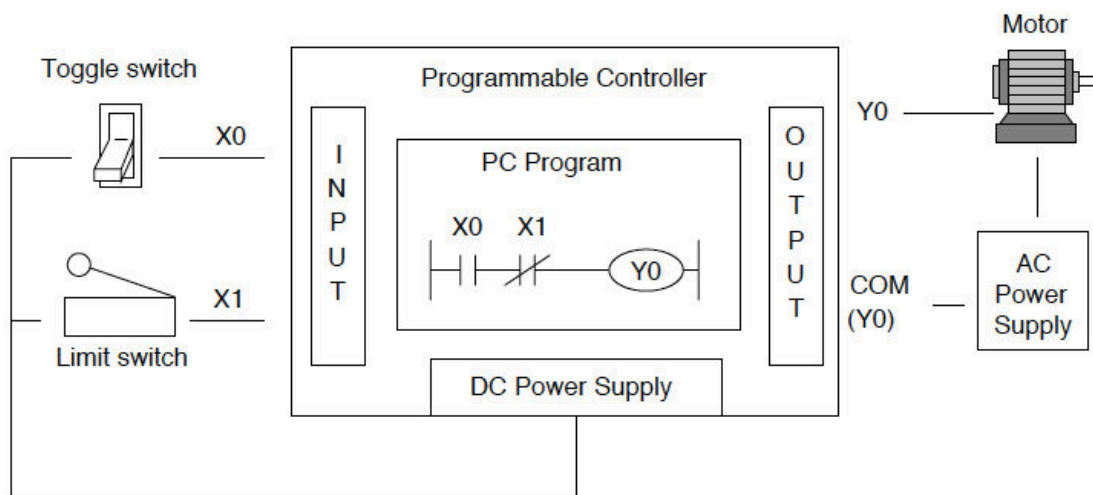
รูปที่ 10 การเชื่อมต่อหน้าสัมผัสแบบ AND

สำหรับตัวอย่างของ AND NOT จะเหมือนการเชื่อมต่อหน้าสัมผัส NO แบบอนุกรมกับหน้าสัมผัสแบบ NC ซึ่งได้แสดงในรูปที่ 11 รวมถึงการตอบสนองเมื่อมีการทำงานของเครื่อง



รูปที่ 11 การเชื่อมต่อและการตอบสนองของการเชื่อมต่อ NO และ NC Contact

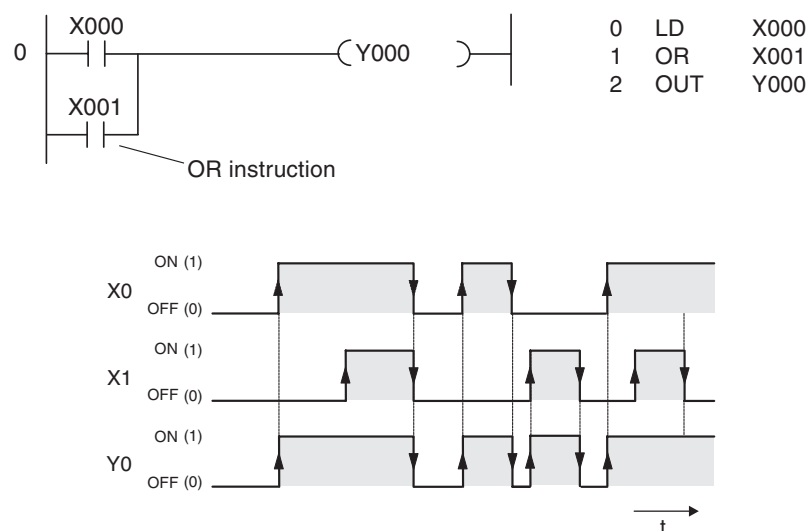
ตัวอย่างการนำมาใช้งานก็เช่นถ้าเราจะเปิดมอเตอร์เพื่อควบคุมการทำงานของอุปกรณ์หนึ่ง โดยใช้ toggle switch แต่มีข้อกำหนดว่าอุปกรณ์นั้นจะต้องไม่อยู่ในตำแหน่งที่เรากำหนด เช่นมอเตอร์ต้องไม่หมุนถ้าแขนของหุ่นยนต์ไปอยู่ที่ตำแหน่งหมุนไปด้านขวาสุดเป็นต้น ซึ่งการตรวจจับจะทำด้วย limit switch การเชื่อมต่อของอุปกรณ์ทั้ง input และ output แสดงในรูปที่ 12



รูปที่ 12 การเชื่อมต่ออุปกรณ์เข้ากับ PLC NO

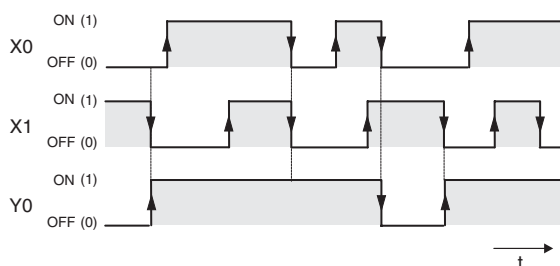
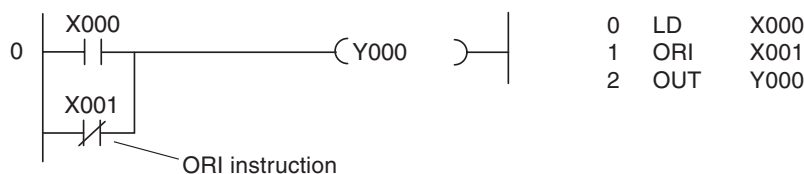
OR Instruction

สำหรับการเชื่อมต่อหน้าสัมผัส เพื่อให้ได้ตรรกะเป็น หรือ (OR) สามารถกระทำได้โดยการเชื่อมต่อหน้าสัมผัสเป็นแบบขนาน ซึ่งสามารถเขียนเป็น lader diagram ได้ตามรูปที่ 13 พร้อมทั้งแสดง line comamn ตามรูปแบบของผู้ผลิตบริษัท Mitshubishi และได้ผลการตอบสนองของระบบไว้ด้วย



รูปที่ 13 การเชื่อมต่อ OR Instruction และการตอบสนอง

เช่นเดียวกับการเชื่อมต่อแบบ AND เราสามารถที่จะเลือกหน้าสัมผัสแบบปกติมาเชื่อมต่อแล OR หรือที่เราเรียกว่า OR NOT instruction ซึ่งแสดงในรูปที่ 13 จะเห็นว่าสำหรับ Mitshubishi นั้นเราจะใช้คำสั่งใน line comman เป็น OR Inverse หรือ ORI

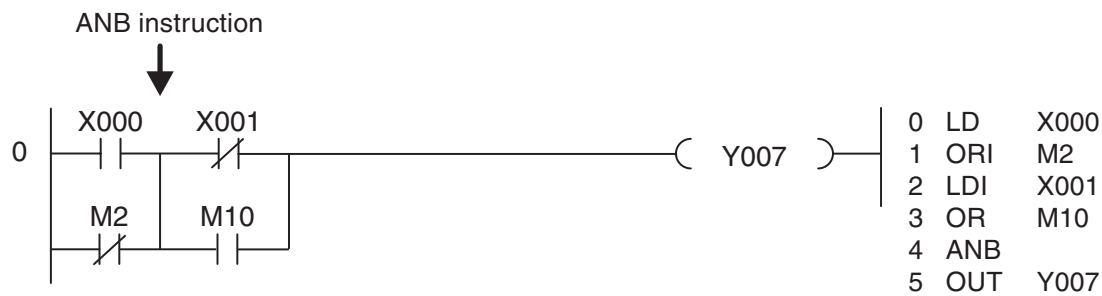


รูปที่ 14 การเชื่อมต่อแบบ OR NOT Instruction และการตอบสนอง

ในหลายกรณีเราจะต้องเชื่อมต่อหน้าสัมผัสที่ยุกยากขึ้นกว่าที่ผ่านมา คือจะต้องมีทั้ง AND และ OR Instruction อยู่รวมกัน ซึ่งกรณีนี้เราจะเขียนหน้าสัมผัสที่ละเส้นหรือเรียกว่าแต่ละ BLOCK จากนั้นจึงนำแต่ละ BLOCK ซึ่งการเชื่อมต่อนี้ก็จะมีทั้ง AND BLOCK และ OR BLOCK

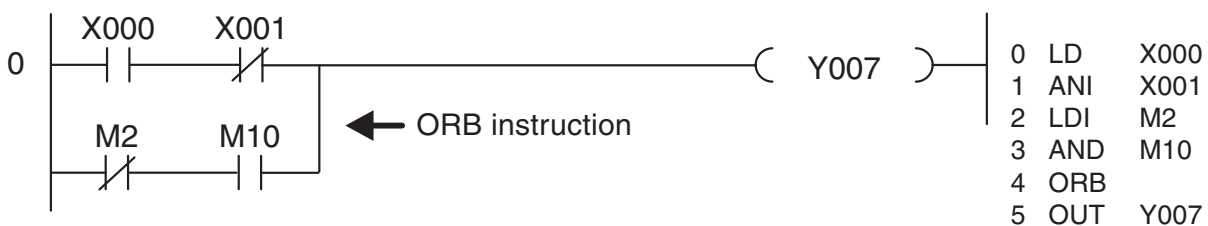
AND BLOCK instruction

พิจารณาจากรูปที่ 15 เราต้องการว่ามี Block อยู่ 2 Block โดยที่ block แรก จะมีหน้าสัมผัสแบบปกติเปิด X000 ต่อขนานอยู่กับหน้าสัมผัสปกติปิดของหน้าสัมผัส M2 และมาเชื่อมต่อแบบ AND กับอีก block หนึ่ง ซึ่งเป็น block ที่หน้าสัมผัสแบบปกติปิด X001 เชื่อมต่ออย่างขนานกับหน้าสัมผัสปกติเปิดของ relay M10 ซึ่งจะเห็นว่าใน line instruction เราจะต้องกำหนดคำสั่งที่ละ block ก่อน โดยจากรูปที่ 14 บรรทัดที่ 0 และ 1 จะนำหน้าสัมผัสสองตัวมาต่อขนานกันเป็น block แรก จากนั้นจะทำการสร้าง block ที่สองในบรรทัดที่ 2 และ 3 ซึ่งตอนแรกจะเหมือนกับการสร้าง line ที่สองใหม่ขึ้นมาอีก 1 line จากนั้น บรรทัดที่ 4 จะเป็นการสั่งให้นำ block แรกและ block ที่สองมาเชื่อมต่อแบบอนุกรมกัน หรือสั่ง AND BLOCK หรือ ANB ตามรูปที่ 15



รูปที่ 15 AND BLOCK Instruction

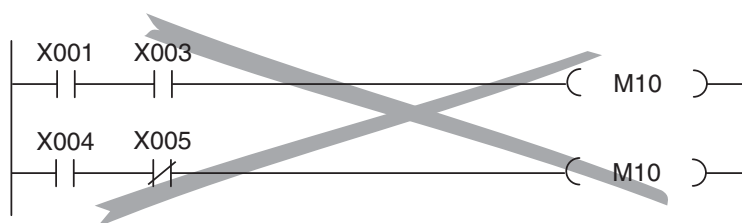
ในการทำงานเดียวกัน หากเรามีการสร้าง block ตั้งแต่สอง block ขึ้นไป แล้วจะนำมาต่อขนานกัน อันดับแรกให้เราสร้าง block ขึ้นมาก่อนที่ละ block โดยแต่ละ block จะขึ้นด้วยคำสั่ง load แล้วจึงนำแต่ละ block มาต่อขนานกันโดยคำสั่ง OR BLOCK ดังที่แสดงในรูปที่ 16 จะเห็นว่าหลักจากที่เราทำการ load block ในบรรทัดที่ 0 และบรรทัดที่ 2 แล้ว จะทำการเชื่อมต่อ block ทั้งสองเข้าด้วยกันที่บรรทัด 4



รูปที่ 16 OR BLOCK Instruction

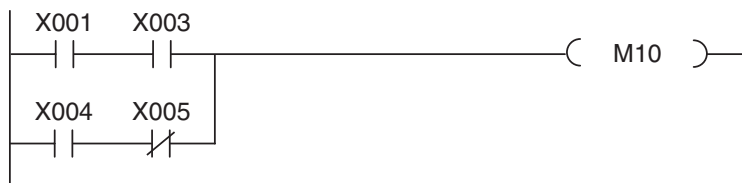
Double Output

ในการเขียน diagram นั้นเมื่อเรามีการเขียนวงจรมากกว่า 1 สาย เพื่อจะควบคุม output หลายๆ อย่าง สิ่งที่เราพบเห็นว่าเป็นความผิดพลาดขึ้นในอันดับแรกเลย คือเราพบว่ามีคำสั่ง OUT ให้กับ relay หรือ coil ตัวเดียวกันมากกว่าหนึ่งครั้ง ซึ่งเมื่อเกิดเหตุการณ์อย่างนี้ขึ้นโปรแกรมจะเกิดการผิดพลาดเพราะตรรกะที่กำหนดให้มีโอกาสที่จะเป็นจริงและเท็จพร้อมกันได้ ตัวอย่างเช่นหากเราพิจารณารูปที่ 17 จะเห็นว่ามีคำสั่ง out ให้กับ relay M10 อยู่สองครั้ง ซึ่งทำให้ CPU ไม่สามารถที่จะตัดสินใจได้ว่าถ้าตรรกะของทั้งสอง line ขัดแย้งกันควรจะกำหนด output เป็นเช่นไร



รูปที่ 17 การเขียน ladder diagram ที่ผิดพลาดโดยมี Double Output

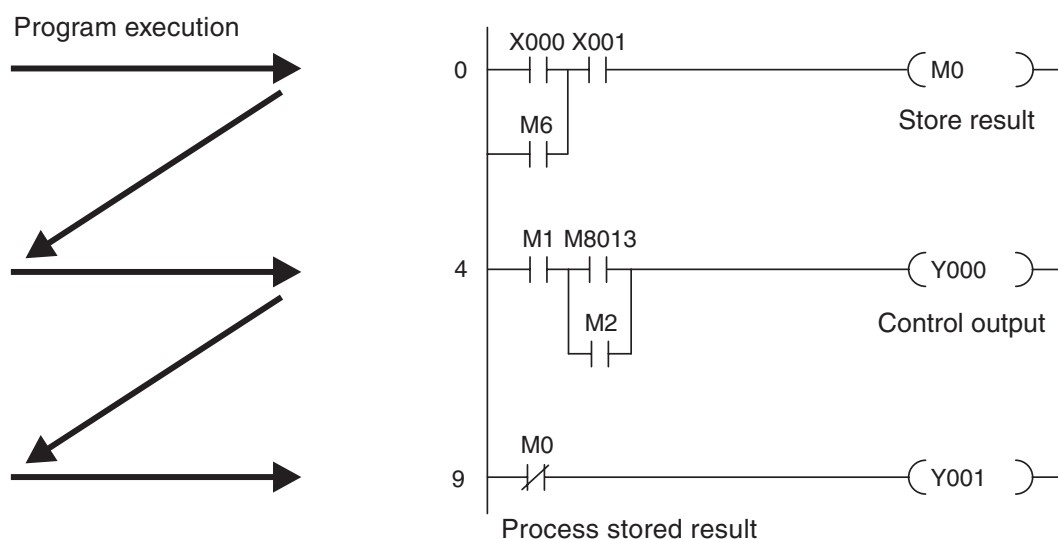
จากรูปที่ 17 หากเราต้องการแก้ไขให้ถูกต้อง โดยมีจุดประสงค์ว่าไม่ว่า line 1 หรือ line 2 เป็นจริงจะให้ relay M10 นั้น ON ขึ้นมา เราจะต้องเขียนในลักษณะของ OR Block ตามรูปที่ 18



รูปที่ 18 การเขียน Lader Diagram ที่ถูกต้องของรูปที่ 17

Program Executions

ในการเขียน Lader Diagram เพื่อให้ CPU ของ PLC ทำการทำงานเพื่อตัดสินใจนั้น ในการทำงาน 1 รอบ หรือเราจะเรียกว่า 1 scan cycle ซึ่งในการทำงานหนึ่งรอบนี้โปรแกรมจะเริ่มทำงานใน line ที่ 1 ก่อน โดยจะเริ่มวิเคราะห์ตรรกะจากซ้ายมือไปขวามือ หลังจากที่ได้วิเคราะห์ line ที่ 1 ก็将继续ทำการวิเคราะห์ line ที่ 2 และ line ต่อๆ ไปจากซ้ายไปขวาเรื่อยๆ ต่อไป ดังแสดงไว้ในรูปที่ 19



รูปที่ 19 การวิเคราะห์โปรแกรมของ PLC ในแต่ละรอบการทำงาน

อย่างไรก็ตามในการวิเคราะห์แต่ละ line นั้น PLC ยังไม่ได้มีการ update output เพียงแต่เป็นการสร้างรูปแบบของ output ไว้สำหรับในแต่ละ line โดยนำมาสร้างเป็น input image ไว้ก่อน เมื่อครบ 1 รอบการทำงานแล้ว CPU จะทำการส่งข้อมูลของ output ที่วิเคราะห์ได้ทั้งหมด ไปให้ output module ทำการ update พร้อมกันทั้งหมดในครั้งเดียว ดังนั้นไม่ว่าโปรแกรมจะมีขนาดความยาวเท่าใดและซับซ้อนเท่าใดก็ตามเราจะเห็นว่ามี การแสดง output ออกมาพร้อมกัน ทำให้เราอาจกล่าวได้ว่า PLC จะพิจารณา program จากซ้ายมาขวาพร้อมกันในทุกๆ line ซึ่งอาจไม่ถูกต้องมากนัก

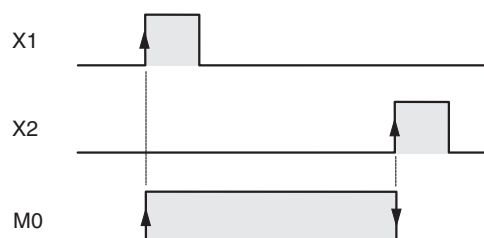
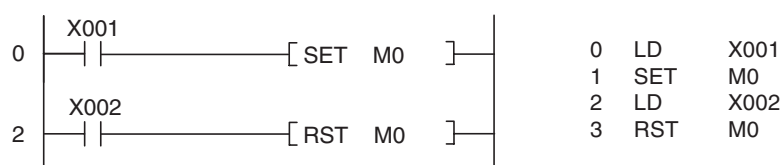
Basic PLC Command Functions

หัวข้อต่อไปนี้จะกล่าวถึงคำสั่งพื้นฐานที่ใช้ในการควบคุม PLC ในเบื้องต้น ซึ่งคำสั่งพื้นฐานในความจริงแล้วสำหรับ PLC นั้นก็มีอยู่เป็นจำนวนมากเช่นกัน เอกสารฉบับนี้ได้เลือกเฉพาะคำสั่งที่พบบ่อยในการใช้ PLC ควบคุมระบบอัตโนมัติพื้นฐาน ซึ่งคำสั่งที่เราจะกล่าวถึงในที่นี้มีดังนี้

1. SET and RESET (or KEEP) Function
2. TIMER and COUNTER
3. DIFF UP and DIFF DOWN
4. MASTER CONTROL

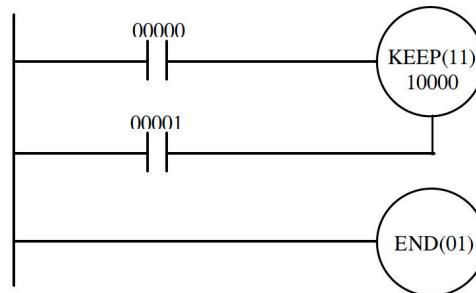
SET and RESET Function (KEEP Function)

คำสั่ง SET จะเป็นคำสั่งที่ให้ relay หรือ coil ที่กำหนดเปลี่ยนสภาพเป็น ON และจะคงสภาพนี้ต่อไปเรื่อยๆ ไม่ว่าคำสั่งจะมี input มาที่ relay นี้หรือไม่ และ relay นี้จะเปลี่ยนสภาพไปก็ต่อเมื่อมีการใช้คำสั่ง reset ไปที่ relay นี้อีกครั้งหนึ่ง ดังเช่นที่แสดงในรูป เมื่อมี input X001 เข้ามาจะมีสัญญาณไปที่คำสั่ง SET ให้ทำการ set relay M0 ทำให้ relay นี้มีสภาพเป็น ON และแม้ว่า input X001 จะเปลี่ยนไปเป็น OFF และไม่มีสัญญาณไปที่ M0 แต่ M0 ก็ยังคงจะ ON ต่อไป ตามที่แสดงในรูปที่ 20 และ relay นี้จะเปลี่ยนเป็น OFF ก็ต่อเมื่อมีสัญญาณมาจาก input X2 เพื่อให้ส่งสัญญาณไป reset relay M0 อีกครั้งหนึ่ง ในกรณีพิเศษที่มีสัญญาณ set และ reset ไปที่ relay ตัวเดียวกันพร้อมกัน PLC จะตัดสินใจใช้คำสั่งที่อยู่ตำแหน่งที่ต่ำกว่าใน ladder diagram ในรูปที่ 20 การเขียนโปรแกรมให้อุปกรณ์ทำงานต่อไป แม้ว่าสัญญาณ input จะขาดหายไปแล้วก็ตาม และจะทำต่อไปจนกว่าจะมีสัญญาณอื่นมากำหนดให้หยุดทำงาน เราเรียกการทำงานลักษณะนี้ว่า Latching



รูปที่ 20(a) การใช้คำสั่ง SET และ RESET รวมถึงการตอบสนอง

สำหรับเครื่องของ OMRON นั้นคำสั่งนี้จะใช้คำสั่ง KEEP ซึ่งถ้าใช้คำสั่งจะเป็น function 11 โดย KEEP นี้จะเป็น Box Function คือมี input เข้าสองตำแหน่ง ตำแหน่งบนจะเป็นคำสั่ง set และตำแหน่งล่างจะเป็น reset ซึ่งแสดงให้เห็นในรูปที่ 20 (b) จะเห็นว่า output 10000 จะ ON เมื่อมีสัญญาณ 00000 มาและติดต่อกันอย่างต่อเนื่องแม้สัญญาณจะหายไป และจะ OFF เมื่อมีสัญญาณมาที่ 00001



รูปที่ 20(b) การใช้คำสั่ง KEEP ของ OMRON

TIMER and COUNTER

สำหรับ Timer และ Counter นั้นส่วนใหญ่จะเป็น relay coil ชุดเดียวกันคือจะใช้ address ในย่านเดียวกัน เพียงแต่ต้องกำหนดให้ PLC ทราบว่าเราจะใช้ coil นั้นเป็น Timer หรือ Counter รายละเอียดของ relay coil ทั้งสองเป็นดังนี้

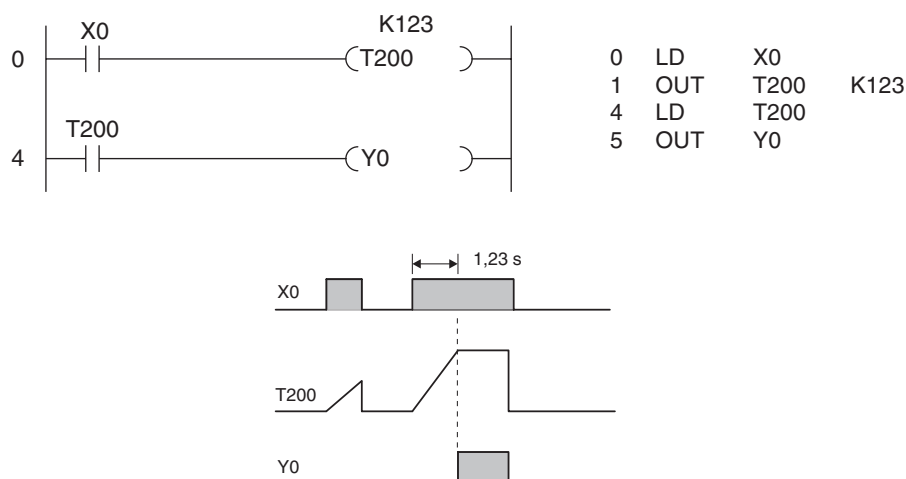
TIMER

Timer หรือตัวหน่วงเวลาจะเป็น coil เมื่อมีสัญญาณเข้ามาหรือสัญญาณขาดไป จะไม่ได้ตอบสนองอย่างทันทีทันใด แต่จะมีการหน่วงเวลาไว้ สำหรับ timer ที่ใช้กันอยู่จะมีดังนี้

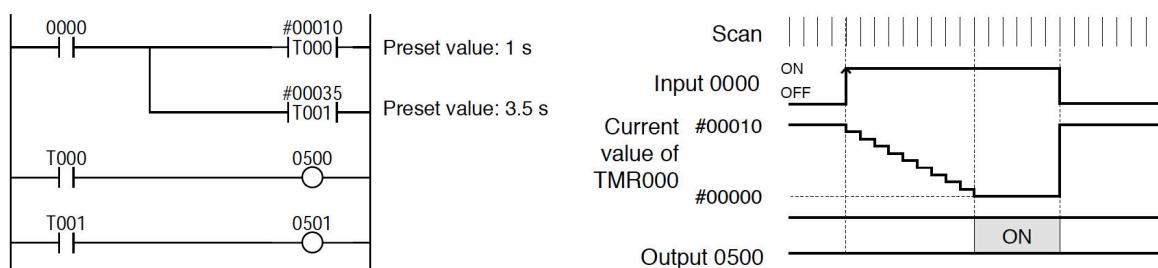
- ON Delay Timer เป็น relay ที่เมื่อได้รับสัญญาณเข้ามาแล้ว จะไม่ทำงานทันทีแต่จะหน่วงเวลาไว้ก่อนเมื่อครบกำหนดเวลา หน้าสัมผัสของ coil นั้นจึงจะเริ่มทำงานและทำงานต่อเนื่องไปเรื่อยๆ จนกว่าสัญญาณเลี้ยง coil จะขาดไปหรือสำหรับบริษัทผู้ผลิตบางบริษัท การที่จะยกเลิกหน้าสัมผัสของ coil นั้นจะต้องคำสั่ง reset ไปที่ coil นั้นอีกครั้งหนึ่ง ส่วนในกรณีที่มีสัญญาณมาที่ timer coil แต่ระยะเวลายังไม่ได้ครบตามที่กำหนดไว้แล้วแต่สัญญาณขาดไปก่อนหน้าสัมผัสก็จะไม่ทำงาน เมื่อมีสัญญาณเข้ามาใหม่การจับเวลาจะเริ่มนับใหม่อีกครั้งหนึ่ง
- Tentative ON Delay Timer เป็น timer คล้าย on delay timer แต่จะแตกต่างกันที่ถ้าสัญญาณขาดหายไปก่อนที่จะจับเวลาได้ครบตามที่กำหนด จะมีการจำไว้ว่าสัญญาณขาดไปเมื่อการจับเวลานับไปถึงเวลาใดแล้ว และเมื่อมีสัญญาณกลับมาใหม่จะเป็นการนับเวลาต่อเนื่องไปจากเวลาเดิมจนครบเวลาที่กำหนดหน้าสัมผัสก็จะทำงาน ทั้งนี้การนับต่อจากเวลาเดิมนี้อาจยกเว้นกรณีที่ มี สัญญาณ Reset เข้ามาก่อนในระหว่างนั้น

- OFF Delay Timer เป็น relay ที่ทำงานตรงกันข้ามกับแบบ ON Delay คือทันทีที่ได้รับสัญญาณเข้ามาแล้ว หน้าสัมผัสของ coil นั้นจะทำงานทันทีแต่เมื่อสัญญาณมาที่ coil นั้นขาดหายไป หน้าสัมผัสจะไม่ขาดออกจากกันทันทีแต่จะหน่วงเวลาไว้ก่อนเมื่อครบกำหนดเวลาหน้าสัมผัสของ coil นั้นจึงจะแยกออกจากกัน และถ้าในระหว่างที่นับเวลาจะแยกหน้าสัมผัสปรากฏว่ามีสัญญาณเข้ามา การนับจะหยุดลงและจะเริ่มนับใหม่อีกครั้งหนึ่งเมื่อหน้าสัญญาณขาดหายไป ยกเว้นในกรณี Tentative OFF Delay Timer จะมีการนับเวลาอย่างต่อเนื่องตามที่ได้อธิบายมาแล้ว

ส่วนการกำหนดค่าตัวเลขการหน่วงเวลานั้นแล้วแต่รุ่นหรือผู้ผลิต แต่โดยส่วนใหญ่แล้ว timer ที่มีความเร็วปกติจะมีจังหวะช่วงการวัด 1 ครั้ง เท่ากับ 0.1 วินาที นั่นคือถ้าเราใส่เลขหน่วงเวลาไว้ เป็นค่า K100 สำหรับ Mitsubishi หรือ #100 สำหรับ OMRON เสมือนกับเรากำหนดให้มีการหน่วงเวลา $0.1 \times 100 = 10$ วินาที นั่นเอง ตัวอย่างของการเขียนคำสั่งของ ON Delay Timer ของ Mitsubishi จะอยู่ในรูปที่ 21 และสำหรับในรูปที่ 22 การเขียนโปรแกรมของบริษัท Keyence

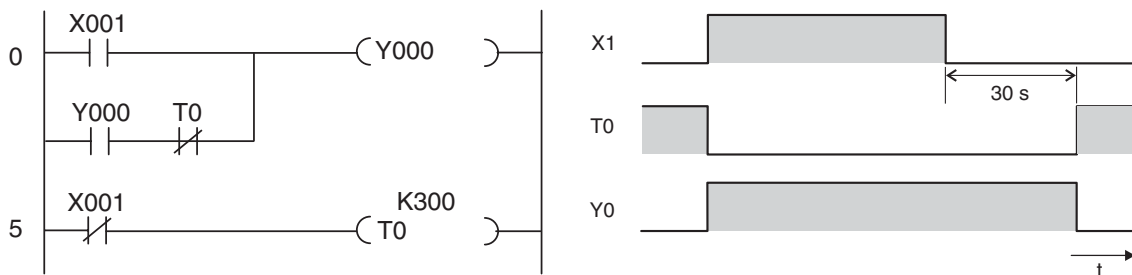


รูปที่ 21 การเขียนคำสั่งและการตอบสนองของการใช้ ON Relay Timer ของ Mishubishi



รูปที่ 22 การเขียนคำสั่งและการตอบสนองของการใช้ ON Relay Timer ของ Keyence

ในเครื่อง PLC บางรุ่น ผู้ผลิตจะมี Timer ให้เฉพาะแบบ ON Delay Timer เพียงอย่างเดียว แต่เราก็สามารถที่จะประยุกต์การใช้งานแบบ off delay ได้ ยกตัวอย่างเช่นถ้าพิจารณาจากรูปที่ 23 เราต้องการให้พัลลัมที่สั่งการโดย output relay Y000 ทำงานต่อไปอีก 30 วินาที แต่ Timer ที่เราใช้สำหรับ PLC รุ่นนี้มีเฉพาะ ON Delay Timer เราจึงต้องปรับเปลี่ยนมาใช้ให้เป็น OFF Delay Timer



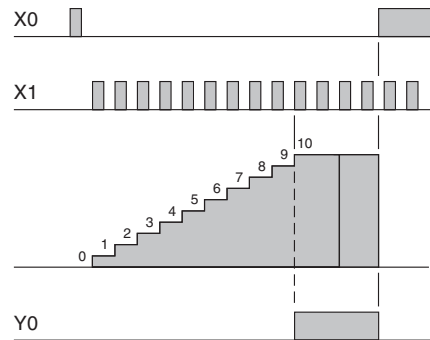
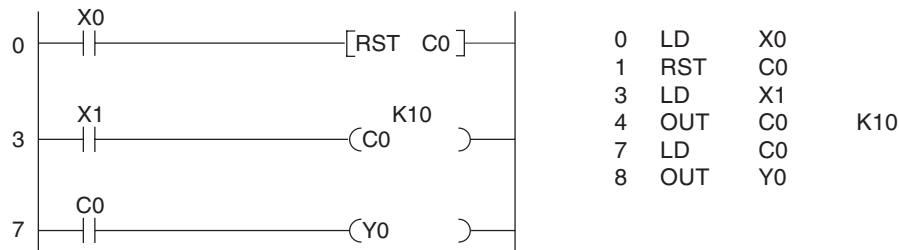
รูปที่ 23 การเขียนโปรแกรม Switch off Delay Time สำหรับ ON Delay Timer

จากรูปที่ 23 จะเห็นว่าเมื่อเราเปิด input X001 พัลลัมจะทำงานตาม Y000 ทันที และเมื่อไฟสัญญาณ X001 ขาดไป นั่นคือให้เครื่องหยุดทำงาน แต่ไฟสัญญาณยังคงส่งไปให้ Y000 อยู่ เนื่องจากการ latching ของสัญญาณในสายล่าง แต่เนื่องจากใน line 2 เราได้ให้ timer T0 เริ่มจับเวลาด้วยค่า 300 วินาทีที่ relay X001 ถูกยกออกเนื่องจากเราตั้งหน้าสัมผัสปกติปิดมาใช้ ดังนั้นสัญญาณจะคงเลี้ยง coil Y000 ไปอีก 30 วินาที หลังจากนั้น timer coil T0 จะทำงาน ทำให้หน้าสัมผัส T0 ที่เป็นปกติปิดแยกออกจากกัน coil Y000 ก็จะขาดและพัลลัมก็จะดับ หลังจากที่มีการสั่งตัดสัญญาณไปแล้ว 30 วินาที

COUNTER

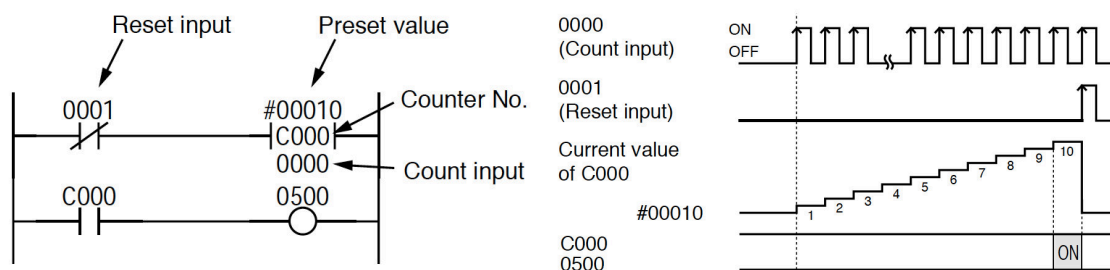
Counter coil เป็น coil ที่จะทำการนับ pluse ของสัญญาณ input ที่ส่งไปที่ coil นั้น และเมื่อจำนวนครั้งครบตามค่าที่กำหนดไว้ relay ของ coil นั้นก็จะเปลี่ยนสภาพไปเป็น ON และจะ ON ไปจนกว่ามีคำสั่งไป reset coil นั้น

ในรูปที่ 24 เป็นการเขียนดปรแกรมของบริษัท Mitsubushi โดยเมื่อมีสัญญาณ pule คือ ON/OFF ไปที่ Counter C0 หนึ่งครั้ง counter ก็จะนับ 1 และนับเพิ่มขึ้นไปเรื่อยๆ จนกระทั่งครบตามค่าที่กำหนดให้ ในที่นี้คือ 10 หน้า contact ของ coil C0 ก็จะทำงานหรือ ON ทำให้มี output ออกไปที่ Y0 และถึงจะมีสัญญาณไปที่ C0 อีกเท่าใดก็ตาม contact C0 ก็ยังคงรักษาสภาพ ON ไว้ จนกว่า จะมีสัญญาณมาจาก X0 เพื่อสั่ง reset counter C0 ก็จะทำให้หน้าสัมผัสเปลี่ยนกลับไปอยู่ในสภาพ OFF ทันที ส่วนในกรณีที่มิสัญญาณมา reset มาที่ counter ก่อนที่จะนับครบแล้วเปลี่ยนเป็น ON มันจะทำให้ counter เริ่มกลับไปนับ 1 ใหม่หลังจากมี pulse มาหลังจากคำสั่ง reset



รูปที่ 24(a) คำสั่ง counter ของ Mitshubishi

ในกรณีของคำสั่ง counter ในทำนองเดียวกันนี้ของบริษัท Keyence จะใช้สัญลักษณ์และคำสั่งตามที่แสดงในรูปที่ 24(b)



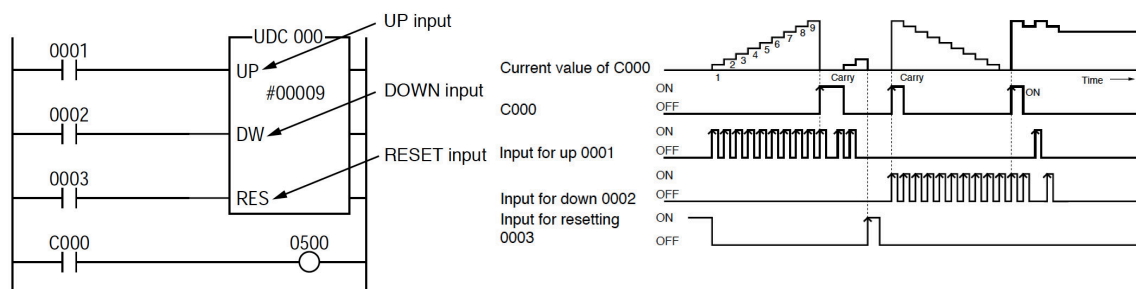
รูปที่ 24(a) คำสั่ง counter ของ Keyence

ข้อควรระวังในการใช้ counter และ timer นั้นก็คือส่วนใหญ่แล้ว relay ทั้งสองจะใช้ address เดียวกัน จึงควรหลีกเลี่ยงที่จะใช้หมายเลขกำกับตัวเดียวกัน เช่น Mitshubishi เราจะต้องไม่กำหนด Timer เป็น T001 แล้วเรียกใช้ Conter C001 อีก เพราะจะเป็น relay address เดียวกันและโปรแกรมจะ error ได้ และก็จะเป็นในทำนองเดียวกันสำหรับผู้ผลิตรายอื่น

UP/DOWN Counter

ใน PLC บางรุ่นและบางผู้ผลิต จะมี counter ที่สามารถนับขึ้นคือบวกเพิ่มขึ้นทีละ 1 เมื่อมีสัญญาณมาที่ช่อง UP ของ counter และจะมีการลบค่าออกทีละ 1 เมื่อมีสัญญาณมาที่ช่อง DOWN ของ counter และเมื่อครบจำนวน contact ของ counter ก็จะมี ON โดย conter แบบนี้จะเรียกว่า

Count UP/Down Counter อย่างไรก็ตามการทำงานของ counter นี้ค่อนข้างจะซับซ้อนกว่า counter ปกติ ซึ่งถ้าพิจารณาร่วมกับรูปที่ 25 เราจะเข้าใจได้มากขึ้น



รูปที่ 25 การทำงานของ Count Up/Down Counter

การที่หน้าสัมผัสของ Up/Down Counter จะ on ได้จะมีสภาวะดังนี้ หากเราเริ่มจำนวนนับเริ่มต้นใน counter เป็น 0 และตั้งค่าจำนวนครั้งไว้ที่ 9

เมื่อมีสัญญาณเปลี่ยนจาก OFF มาเป็น ON หนึ่งครั้ง (diff up) counter จะนับ 1 และนับเพิ่มไปอีกแต่ละสัญญาณที่มาที่ UP input เมื่อครบค่าที่กำหนดไว้ในที่นี้คือ 9 แล้ว relay พร้อมที่จะเปลี่ยนสภาพ จากนั้นเมื่อมีสัญญาณมาอีกหนึ่งครั้งเป็นครั้งที่ 10 relay ก็จะเปลี่ยนเป็น ON (คือไม่ได้ ON ครั้งที่ 9 เหมือน counter relay ธรรมดา) และถึงแม้จะไม่มีคำสั่ง reset หากเราให้สัญญาณที่ Up Input เข้าไปอีกครั้งหนึ่งที่จังหวะ diff down ของสัญญาณ relay ของ counter ก็จะเปลี่ยนเป็น OFF และก็จะมัลักษณะคล้ายกันเมื่อมีสัญญาณมาที่ Down Input สรุปการ ON ของ relay updown Counter ที่ตั้งค่าไว้ที่ #N ได้ดังนี้

- ❖ ON เมื่อมี UP Input เพิ่มขึ้นจากค่าจาก N ไปเป็น N+1
- ❖ ON เมื่อมี Down Input ลดค่าจาก 0 ไปเป็น -1
- ❖ ON เมื่อมี UP Input เพิ่มค่าจาก -1 เป็น 0
- ❖ ON เมื่อมี Down Input ลดค่าจาก N+1 ไปเป็น N

นอกเหนือจากนั้น relay จะ OFF เสมอ รายละเอียดจะดูได้จากรูปที่ 25

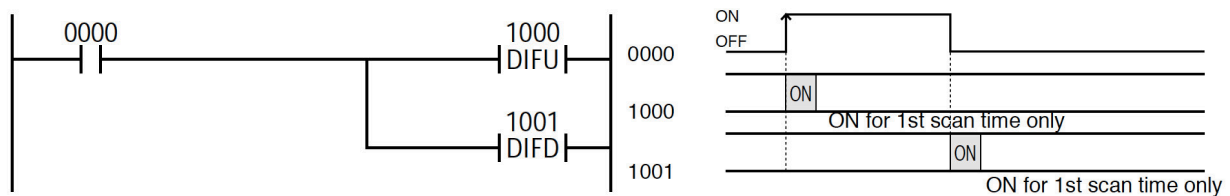
Diff UP and Diff Down Function

ฟังก์ชันนี้จะให้สัญญาณ pluse สั้นๆ ออกมาในหนึ่งรอบการทำงานของเครื่อง หรือ 1 scan cycle โดยพิจารณาว่ามีการเปลี่ยนแปลงสัญญาณจาก ON เป็น OFF หรือ OFF เป็น ON หรือไม่ สัญญาณที่เกิดขึ้นนั้นจะรวดเร็วมากเพราะแต่ละรอบในการทำงานของ PLC ในปัจจุบันนี้ถือว่าเร็วมาก โดยส่วนมากถ้าเราพิจารณาว่าฟังก์ชันนี้ทำงานหรือไม่โดยการต่อเข้ากับหลอดไฟ เราอาจไม่เห็นหลอด

ไฟกระพริบ เพราะจังหวะให้ไฟกับหลอดไฟเร็วมากจนหลอดไฟยังไม่สามารถเปล่งแสงได้ ไฟก็จะดับไปแล้ว

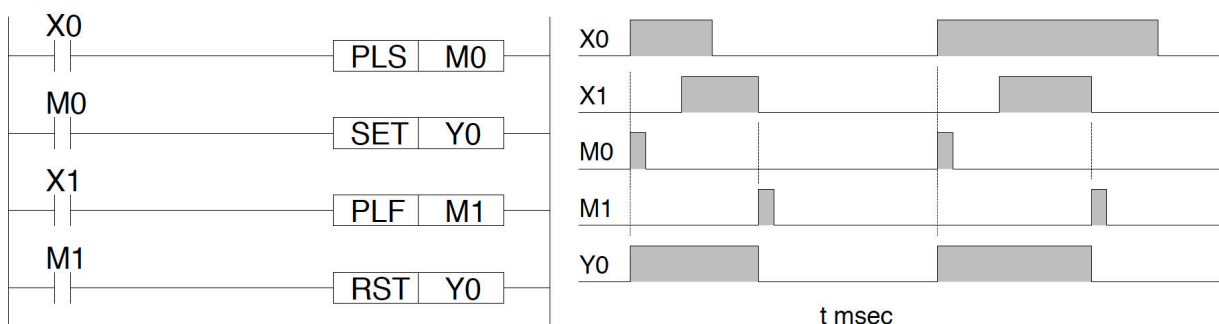
- Diff Up Function นั้นจะให้ค่าสัญญาณออกมา 1 pluse ระยะเวลาเท่ากับ time scan 1 รอบ ของ PLC ถ้าหากว่าในรอบการทำงานนั้น input ที่ coil มีการเปลี่ยนค่าจาก OFF มาเป็น ON
- Diff Down Function นั้นจะให้ค่าสัญญาณออกมา 1 pluse ระยะเวลาเท่ากับ time scan 1 รอบ ของ PLC ถ้าหากว่าในรอบการทำงานนั้น input ที่ coil มีการเปลี่ยนค่าจาก ON มาเป็น OFF

ถ้าพิจารณาจากรูปที่ 26 จะเห็นว่าถ้าในรอบการทำงานเราสลับไฟจาก input 0000 จาก OFF ไปเป็น ON เราจะได้ไฟที่ output 1000 มา 1 pluse เพราะมีการเรียกสัญญาณ diff up (DIFU) ส่วน เราจะได้ pluse ออกไปที่ 1001 ก็เมื่อเราปิด input 0000 จาก ON ไปเป็น OFF



รูปที่ 26 การทำงานของ Diff Up และ Diff Down ของ Keyence

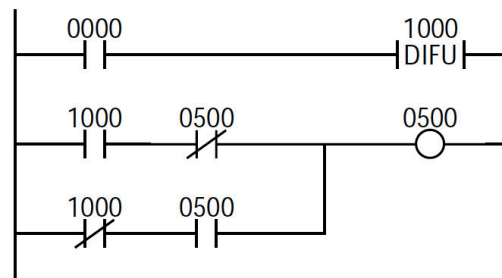
สำหรับเครื่องของ Mitshubishi คำสั่งนี้จะใช้เป็น Leading and Trailing Pulse และจะย่อด้วย PLS และ PLF ตามลำดับ ซึ่งลักษณะการทำงานนั้น PLS จะทำงานเหมือนกับ Diif Up และ PLF จะทำงานเหมือนกับ Diff Down พิจารณาจากตัวอย่างแผนภาพและการตอบสนองในรูปที่ 27



รูปที่ 27 การทำงานของ PLS และ PLF ของ Mitshubishi

สำหรับการนำฟังก์ชันนี้ไปใช้งานนั้น เราจะพบเห็นได้ทั่วไปสำหรับระบบควบคุมอัตโนมัติ โดยเฉพาะที่ใช้ limit switch หรือ push button ในการตรวจจับ สำหรับตัวอย่างที่ยกให้เห็นในรูปที่ 28 นี้ เป็นโปรแกรมของ Keyency โดยสมมุติว่า input 0000 จะเป็น push button อันหนึ่ง และ output 0500 นั้นจะเป็น หลอดไฟดวงหนึ่ง วงจรนี้เรียกว่า Alternating circuit เราขอทิ้งรายละเอียดไว้ให้คุณ

ได้ลองเขียนและวิเคราะห์การทำงานเองในห้องปฏิบัติการ

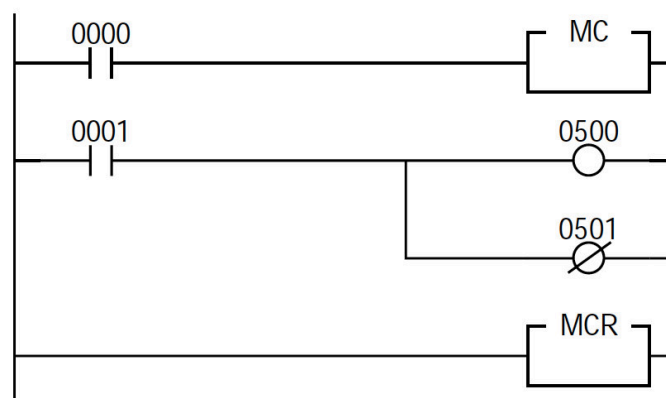


รูปที่ 28 Alternating Circuit

Master Control (or Interlock, IL)

Master Control หรือ interlock ขึ้นกับผู้ผลิต (Keyence หรือ Omron ตามลำดับ) เป็นชุดคำสั่งที่ทำให้ command line ที่อยู่ระหว่างคำสั่ง Master Control, MC (หรือ interlock, IL) กับคำสั่ง Master Control Clear, MCR (หรือ Interlock Clear, IC) นั้นไม่ทำงาน ถ้าหากว่าไม่มีสัญญาณเข้าไปที่ MC แต่ถ้ามีสัญญาณเข้าไป command line ที่อยู่ระหว่างนั้นจะทำงานตามปกติ

ถ้าเราพิจารณารูปที่ 29 เราจะเห็นว่าถ้าเราเปิด input 0000 ไว้ ไม่ว่าเราจะเปิดหรือปิด input 0001 ก็ตาม output 500 และ 501 จะเป็น OFF เสมอ แต่ถ้าเรา ON input 0000 และ ON input 0001 เราจะพบว่า output 500 จะ ON และ 501 จะ OFF และจะตรงกันถ้าเรา OFF input 0001 เพราะการที่เรา ON input 0000 จะทำให้ command line ที่อยู่ระหว่าง MC และ MCR นั้นได้รับการพิจารณาจาก CPU ของ PLC ซึ่งตรงข้ามกับกรณี input 0000 OFF จะทำให้ command line ที่อยู่ระหว่าง MC และ MCR นั้นไม่ได้รับการพิจารณาจาก CPU ของ PLC

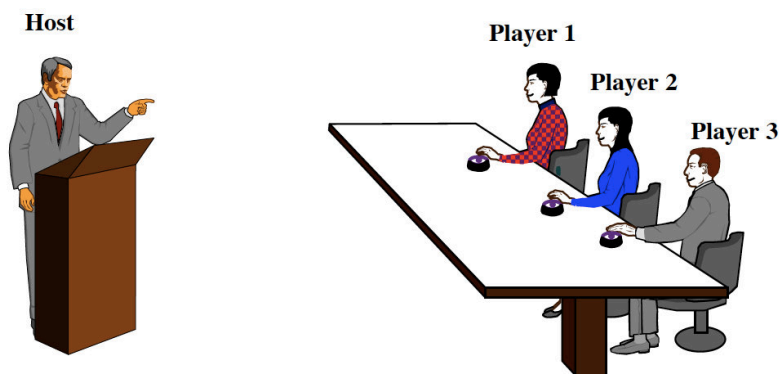


รูปที่ 29 การใช้คำสั่ง MC และ MCR

Laboratory Excercies

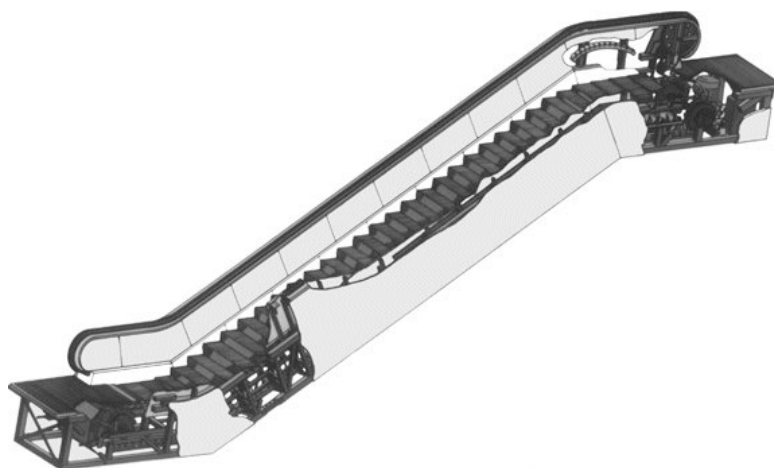
สำหรับปฏิบัติการนี้ เราจะให้นักศึกษาได้ฝึกการใช้โปรแกรมในการเขียน ladder diagram เรียนรู้วิธีการ upload และ download โปรแกรมระหว่าง computer และ PLC เนื่องจากเราจะใช้ PLC ที่แตกต่างกันและให้นักศึกษาหมุนเวียนกันใช้ผู้ผลิต เราจึงจะแจกรายละเอียดของแบบฝึกหัดไว้ต่างหากที่ห้องปฏิบัติการทดลอง สำหรับการทดลองจะมีเป็นข้อๆ ดังนี้

1. จงเขียน Lader Diagram เพื่อควบคุมการทำงานของอุปกรณ์ที่ควบคุมการทำงานด้วยการเชื่อมต่อสวิตช์ input แบบ AND 2 อันและมี output 1 อัน เขียน ladder diagram บน computer แล้ว upload ไปสู่เครื่อง PLC
2. จงเขียน Lader Diagram เพื่อควบคุมการทำงานของอุปกรณ์ที่ควบคุมการทำงานด้วยการเชื่อมต่อสวิตช์ input แบบ OR 2 อันและมี output 1 อัน เขียน ladder diagram บน computer แล้ว upload ไปสู่เครื่อง PLC
3. จงเขียน lader diagram เพื่อควบคุมการเปิดปิดไฟบันได คือผู้ใช้ไม่ว่าจะสลับสวิชไฟไปในทิศทางใด ก็สามารถที่จะเปิดหรือปิดไฟตามที่เรต้องการ พร้อมทั้งทดสอบการทำงานที่ควบคุมด้วย PLC
4. จงเขียน lader diagram เพื่อควบคุมการเปิดปิดไฟ ถ้ามี push button เป็น input อยู่ 2 เมื่อกดปุ่มแรก หลอดไฟที่ output จะติด และติดต่อไป แม้ว่าเราจะปล่อยสวิตช์ไปแล้วก็ตาม ไฟจะดับก็ต่อเมื่อเรากด push botton อันที่สอง
5. จงเขียน lader diagram เพื่อควบคุมการเปิดปิดไฟของผู้เล่นเกมสโว์ โดยมีพิธีกร 1 คน และผู้เข้าแข่งขันอีก 3 คน เมื่อกรรมการเปิดไฟของตน ผู้เข้าแข่งขันจึงจะสามารถแย่งการกดไฟได้ การกดก่อนไฟกรรมการติดจะไม่มีผล และเมื่อผู้เข้าแข่งขันผู้ใดกดไฟได้ก่อน ไฟตนเองจะติด และผู้เข้าแข่งขันที่เหลืออีกสองคนจะไม่สามารถกดไฟให้ติดได้ จนกระทั่งกรรมการปิดไฟตนเองวงจรจึงจะกลับเข้าสู่สภาวะเริ่มต้นอีกครั้งหนึ่ง



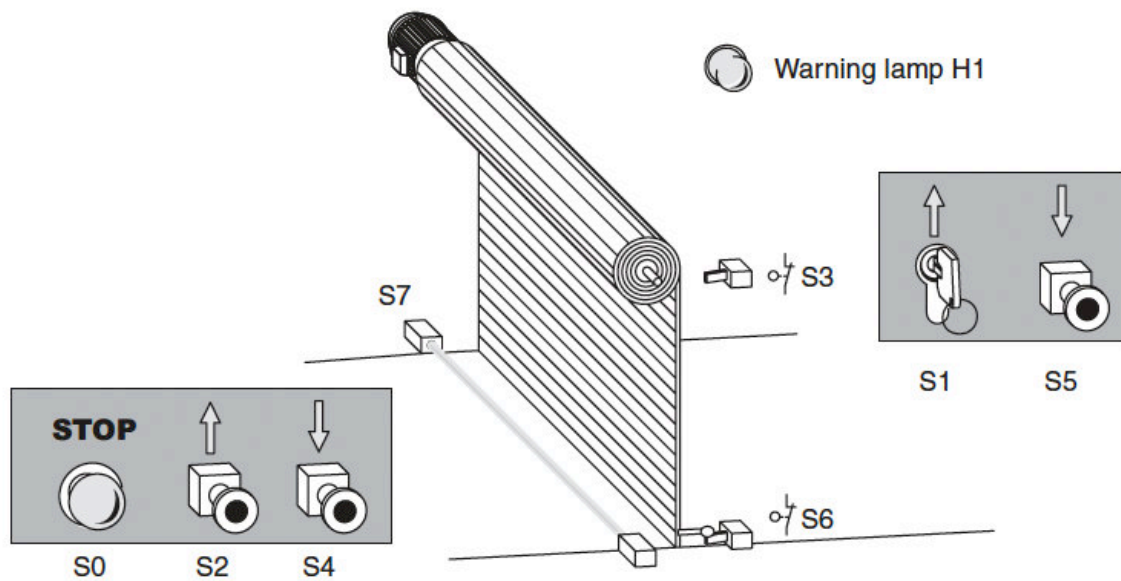
รูปสำหรับโจทย์ข้อ 5

6. จงเขียน lader diagram เพื่อควบคุมการเปิดปิดไฟเริ่มจากไฟดับอยู่และเมื่อเราเปิด master switch ไฟดวงหนึ่งจะกระพริบติดดับสลับกันทุก 1 วินาทีไปเรื่อยๆ และหลอดไฟนี้จะหยุดกระพริบเมื่อเราสั่ง off master switch
7. จงเขียน lader diagram เพื่อควบคุมการเปิดปิดไฟ 3 ดวงเริ่มจากทุกดวงดับหมด เมื่อเปิด สวิทช์หลักจะเริ่มต้นการทำงานคือ ไฟดวงที่หนึ่งติดหนึ่งวินาที ขณะที่ดวงอื่นดับ จากนั้น เปลี่ยนเป็นดวงที่สองติดหนึ่งวินาที ขณะที่ดวงอื่นดับ แล้วเปลี่ยนเป็นดวงที่สามติดหนึ่งวินาที ขณะที่ดวงอื่นดับ ต่อด้วยดวงที่สองติดหนึ่งวินาที ขณะที่ดวงอื่นดับ จากนั้นจะย้อนกลับมา เป็นดวงที่หนึ่งติดดวงเดียวหนึ่งวินาที แล้ววนไปเรื่อยๆ จนกว่าจะสลับสวิทช์หลัก
8. ออกแบบโปรแกรมควบคุมการทำงานแบบประหยัดพลังงานของบันไดเลื่อน ที่เซิงบันไดจะมี เซนเซอร์ตรวจจับว่ามีคนเดินผ่านหรือไม่ ถ้าไม่มีคนเดินผ่านเป็นระยะเวลา 30 วินาที PLC จะสั่งให้มอเตอร์บันไดเลื่อนหยุดหมุนและจะหมุนก็ต่อเมื่อมีคนมาที่เซิงบันไดอีก



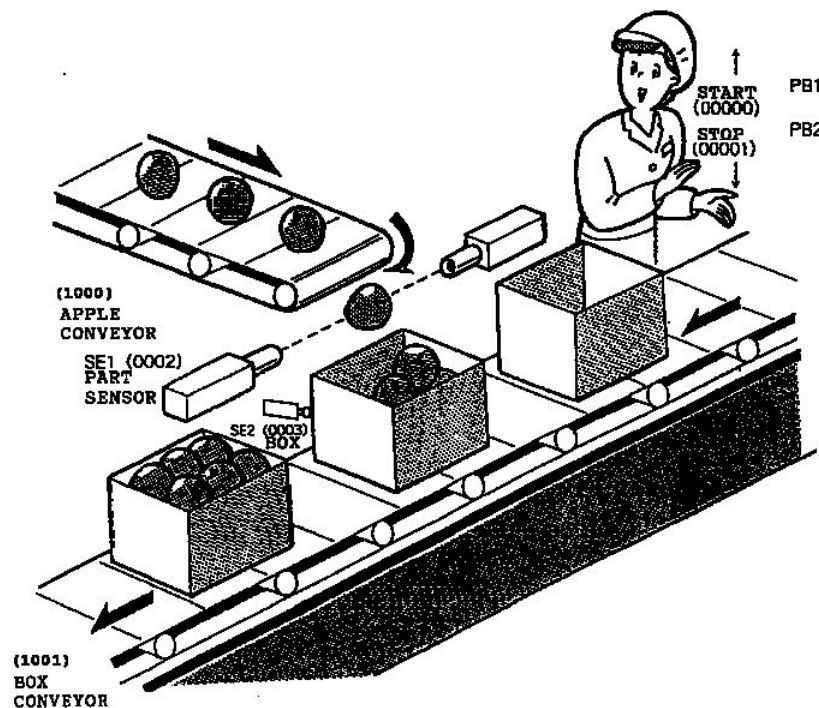
รูปสำหรับโจทย์ข้อ 8

9. จงเขียน lader diagram เพื่อควบคุมการเปิดปิดไฟ 3 หลอด โดยเริ่มต้นทั้งสามหลอดจะดับ ทั้งหมด ถ้ากด push button A หนึ่งครั้งหลอดไฟจะติด 1 หลอด เมื่อกดครั้งที่สองหลอดไฟ จะติดเพิ่มขึ้นอีกหนึ่งหลอด เมื่อกดครั้งที่สามหลอดไฟจะติดเพิ่มขึ้นอีกหนึ่งหลอด แต่เมื่อกด ครั้งที่สี่หลอดไฟทุกดวงจะดับหมด และหากเราเริ่มกดใหม่ วงจรก็จะเริ่มวนต่อไป
10. จงเขียน lader diagram เพื่อควบคุมการเปิดปิดประตูโรงรถ โดยทางเข้าจะมีกุญแจ S1 เพื่อให้ประตูเปิดส่วน push button S5 ด้านนอกจะสั่งให้ประตูปิดลง สำหรับด้านในจะมี push button 2 อันสำหรับ S2 จะให้ประตูเลื่อนขึ้นและ S4 จะทำให้ประตูเลื่อนลง นอกจากนั้นที่ ประตูจะมี Limit Switch S3 ตรวจสอบว่าประตูขึ้นสุด และ S4 เพื่อตรวจสอบว่าประตูเลื่อน ปิดสุด นอกเหนือจากนั้นเพื่อความปลอดภัยป้องกันไม่ให้ประตูปิดระหว่างที่มีรถจอดขวาง อยู่จะมี switch S7 เป็น safty switch อยู่ และระหว่างประตูเลื่อนเปิดหรือปิดไฟ H1 จะติด และลองทดสอบการทำงานบน PLC ด้วยไฟแสดงผลด้วย



รูปสำหรับโจทย์ข้อ 10

11. เมื่อกดปุ่ม PB1 (Start Push Button) กล่องที่อยู่จะเคลื่อนที่โดย conveyor ของกล่อง และ เมื่อกกล่องเข้าประจำที่เรียบร้อย conveyor กล่องจะหยุด และ conveyor ของ apple จะเริ่มทำงาน เมื่อ apple ตกลงบนกล่องครบ 10 ลูก conveyor ของ apple จะหยุดทำงาน และ conveyor ของกล่องจะเริ่มเคลื่อนที่อีกครั้ง เป็น cycle ต่อไป และจะหยุดเมื่อกด PB2 (STOP Push Button) จงเขียนโปรแกรม PLC เพื่อควบคุมระบบดังกล่าว



รูปสำหรับโจทย์ข้อ 11