ครบเครื่องเรื่อง PLC SIEMENS SIMATIC S7-1200

พิศนุรัตน์ เขจร

อาคม พิทักษ์วีระกุล

สารบัญ

บทที่ 5 พื้นฐานการโปรแกรม (Basic Programming)

บทที่ 6 การใช้งาน Timer/Counter เบื้องต้น

บทที่ 7 Data block

บทที่ 8 การใช้งาน Array เบื้องต้น

บทที่ 9 ฟังก์ชันต่างๆของ PLC

บทที่ 10 การใช้งาน Analog Input/Output

บทที่ 11 การใช้งาน High-Speed Counter (HSC)

บทที่ 12 การใช้งาน Pulse output

บทที่ 13 Technology object

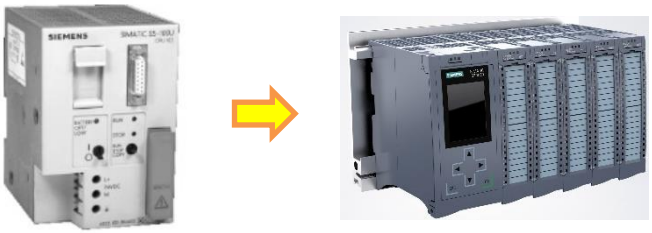
บทที่ 14 การสื่อสารของ S7-1200

บทที่ 1 แนะนำ PLC SIEMENS S7-1200

1.1 แนะนำ PLC siemens S7 รุ่นต่างๆและซอฟต์แวร์ที่ใช้งาน

คำถามที่มักจะพบบ่อยๆสำหรับผู้ที่เริ่มต้นหัดใช้งาน Siemens คือ ต้องใช้งาน software อะไรบ้างเพื่อใช้งานสำหรับ hardware แต่ละรุ่น หรือผู้ใช้งานบางท่านก็ยังเข้าใจผิดว่า TIA Portal เป็น software ประเภท all-in-one คือ ลง TIA Portal ตัวเดียวก็สามารถเลือกใช้งานได้ทั้ง PLC, HMI และ Inverter/servo ซึ่งเป็นความเข้าใจที่ผิด ดังนั้นในบทนี้ เราจะมาสรุปรายการ software ที่ต้องติดตั้งสำหรับการใช้งานอุปกรณ์แต่ละตัว โดยจะเน้นไปที่รุ่นปัจจุบันเท่านั้น

PLC (Programable logic controller) หรือ controller คือชื่อทั่วไปของอุปกรณ์ควบคุมที่ใช้ CPU กรณีค่ายซีเมนต์อุปกรณ์ที่ใช้ในระบบอัตโนมัติ(PLC, Servo, HMI) จะเรียกรวมว่า Simatic (Siemens + Automatic) เช่น Simatic S7-1200, Simatic HMI, Simatic G120 เป็นต้นโดยสำหรับ PLC จะขึ้นต้นด้วย S7 , S5 ซึ่ง S7 คือซีรีส์ในปัจจุบันที่เข้ามาแทน S5 ซึ่ง S5 ไม่ได้ทำการผลิตแล้ว



Simatic S5 Simatic S7 (S7-1500)

รูป 1.1

Simatic S7 รุ่นเก่าๆมี 3 รุ่นหลักคือ Simatic S7-200 (รุ่นเล็ก) Simatic S7-300 (รุ่นกลาง) Simatic S7-400 (รุ่นใหญ่) โดยซอฟต์แวร์ที่ใช้คือ Simatic STEP 7 manger

Simatic S7- 1200 รูป 1.2

ในปัจจุบัน Simatic S7-1200 และ S7-1500 คือรุ่นที่กำลังทำตลาด โดยใช้ซอฟต์แวร์ตัวใหม่คือ STEP 7 โดย STEP 7 เป็นซอฟต์แวร์ที่อยู่ในแพล็ตฟอร์ม TIA Portal

Simatic S7-200

รูป 1.3

S7-200 เป็น PLC รุ่นเก่าที่ไม่ได้ทำตลาดแล้วในปัจจุบัน จะต้องใช้ software ที่ชื่อว่า STEP7-MicroWin4.0 ซึ่งเป็นโปรแกรมฟรี โดย version ล่าสุดคือ STEP7-MicroWin4.0 SP9

Simatic S7-200 SMART

แม้ว่า S7-200 จะหยุดทำตลาดไปแล้วในตลาดทั่วโลก แต่ในจีนนั้นยังคงมีการใช้งานอยู่โดยเป็นรุ่นพิเศษที่ใช้ในจีนเท่านั้นคือ S7-200 SMART

หากท่านพบว่าท่านใช้งาน S7-200 SMART อยู่ ขอให้ทราบว่าท่านต้องช่วยเหลือตนเองเท่านั้น ไม่สามารถขอความช่วยเหลือจากประเทศอื่นได้เลยนอกจากประเทศจีน และ software ที่ใช้จะเป็น software เฉพาะสำหรับ S7-200 SMART ที่ต้องใช้งาน Windows ที่เป็นภาษาจีนด้วยเช่นกัน

รูป 1.4

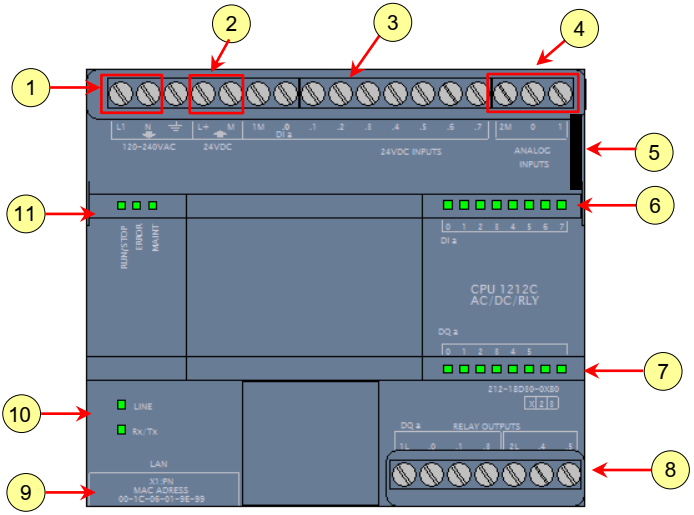
Simatic S7-300/S7-400

รูป 1.5

S7-300 เป็นรุ่นยอดนิยมและน่าจะเป็นรุ่นที่ใช้กันแพร่หลายที่สุด เพราะมีผลิตมานานมาก Software ที่ใช้คือ STEP7 V5.5 หรือ STEP7 V5.6 ที่เป็นตัวล่าสุด

1.2 SIMATIC S7-1200

SIMATIC S7-1200 คือ PLC แบบ compact ซึ่งประกอบด้วย Microprocessor, power supply วงจร input และ output, built-in PROFINET, high-speed motion control I/O และ on-board analog เป็นต้น โดยมีรุ่นของ CPU คือ CPU 1211C, CPU 1212C , CPU 1214C, CPU 1215C และ CPU 1217C



รูป 1.6

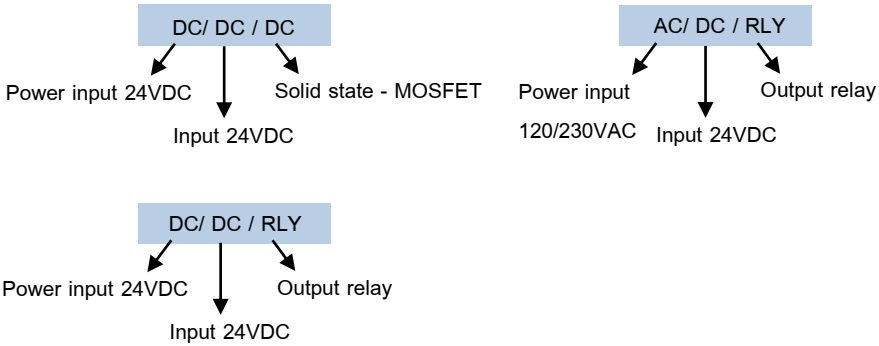
รูปที่ 1.6 เป็นโครงสร้างทางกายภาพของ S7-1200 มีดังนี้

1. Power connector คือ connector สำหรับต่อกับแหล่งจ่ายไฟ
2. 24 VDC Sensor Power Out คือแหล่งจ่ายไฟ 24VDC ที่จ่ายจากตัว PLC ใช้สำหรับจ่ายไฟเลี้ยงวงจรอินพุทหรือเอาต์พุต
3. Input wiring connectors คือ connector สำหรับวายริ่งกับอุปกรณ์อินพุท
4. Analog input connector คือ connector สำหรับวายริ่งกับอุปกรณ์อินพุทแบบอนาล็อก
5. Memory card slot
6. สัญญาณไฟเมื่อมีไฟจ่ายให้บิตอินพุท
7. สัญญาณไฟเมื่อมีบิตเอาต์พุตทำงาน
8. Output wiring connectors คือ connector สำหรับวายริ่งกับอุปกรณ์เอาต์พุต
9. PROFINET connector คือพอร์ตสื่อสารแบบ PROFINET Network (LAN)
10. สัญญาณไฟเมื่อมีการสื่อสารแบบ PROFINET Network (LAN)
11. สัญญาณไฟแสดงสถานะของ CPU เช่น Stop, Run, Error และ Maint

ชนิดไฟเลี้ยง/ชนิด input /ชนิด output

รูป 1.7

รูปที่ 1.7 ที่ตัวถัง PLC จะแสดงรุ่นของ CPU ตัวอย่างในรูปคือ CPU 1212C ส่วนบรรทัดล่างคือ AC/DC/RLY จะบอก ชนิดไฟเลี้ยง/ชนิดแรงดันที่จ่ายให้วงจรอินพุท/ชนิดอุปกรณ์เอาต์พุต โดยมี Code 3 แบบดังนี้



ชนิดไฟเลี้ยง PLC มีสองแบบคือ AC และ DC โดยมีช่วงการจ่ายไฟดังตารางที่ 1.1

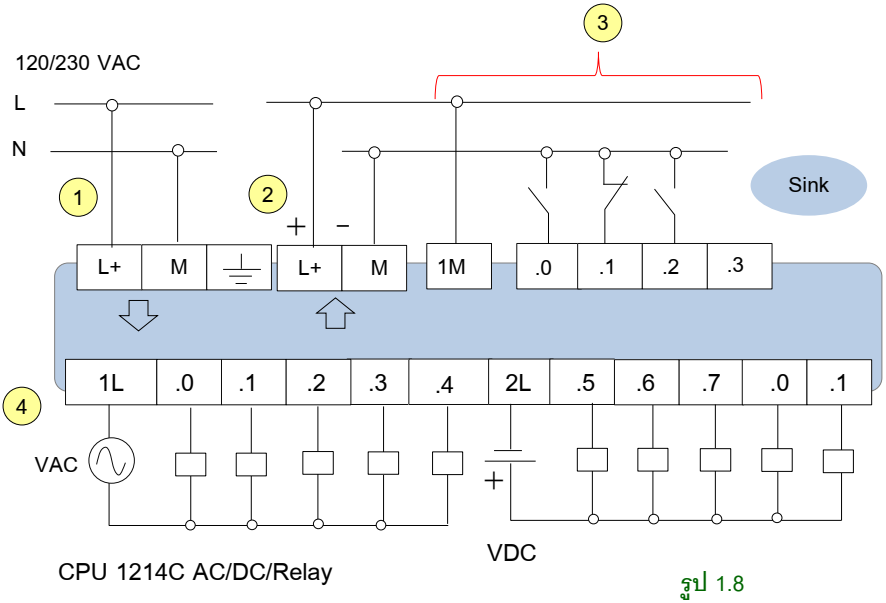
ตาราง 1.1

พิกัดแรงดัน	Tolerance
24 VDC	20.4 VDC to 28.8 VDC
120/230 VAC	85 VAC to 264 VAC, 47 to 63 Hz

ตาราง 1.2 High potential isolation test

High potential isolation test	
24 V/5 V nominal circuits	520 VDC (type test of optical isolation boundaries)
115/230 V circuits to ground	1500 VAC
115/230 V circuits to 115/230 V circuits	1500 VAC
115 V/230V circuits to 24 V/5 V circuits	1500 VAC (3000 VAC / 4242 VDC type test)
Ethernet port to 24V/5V circuits and ground	1500 VAC (type test only)

การวายริ่งอินพุทและเอาต์พุต PLC แบบ AC/DC/Relay



รูป 1.8

① L+ และ M และมีลูกศรชี้เข้าหมายถึงแหล่งจ่ายสำหรับจ่ายให้ PLC กรณีใช้ไฟ AC จะต่อ L+ , M เข้ากับ L และ N

② L+ และ M และมีลูกศรชี้ออกหมายถึงแหล่งจ่ายไฟที่ออกจาก PLC ในรูปใช้ L+ และ M จ่ายไฟให้กับวงจรอินพุทของ PLC

③ วงจรอินพุทของ PLC การวายริ่งอุปกรณ์ภายนอกกับอินพุท PLC จะใช้ไฟเลี้ยงแบบ 24VDC โดยวายริ่งได้ทั้งแบบ sink และ source

1M คือคอมมอนของวงจรอินพุท กรณีวายริ่งแบบ sink ต่อ 1M กับ 24V ส่วนการวายริ่งแบบ source จะต้องจ่ายไฟ 0V เข้า 1M กรณีไม่ได้ใช้แหล่งจ่ายในข้อ 2 สามารถใช้แหล่งจ่ายจากภายนอกก็ได้

④ วงจรเอาต์พุตของ PLC โดยเอาต์พุตเป็นแบบรีเลย์ สามารถใช้ไฟได้ทุกแบบทั้ง AC และ DC โดยมีเอาต์พุตมากสุดสองกลุ่มคือ 1L และ 2L

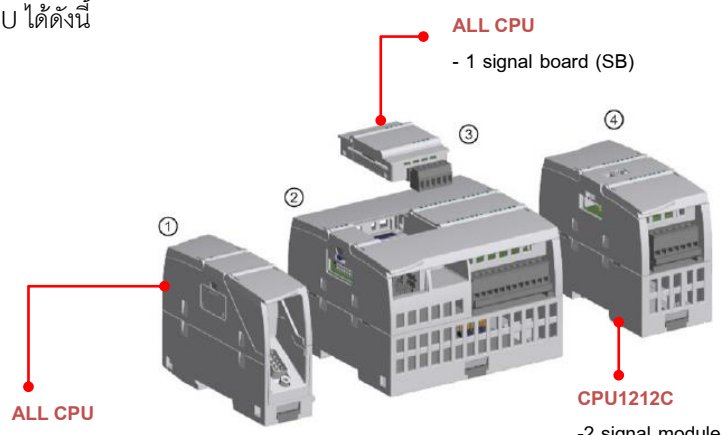
หมายเหตุ

1.CPU ที่เป็น models แบบ relay outputs คุณสามารถติดตั้ง digital signal board (SB) กรณีต้องการใช้ pulse output

2. การใช้งาน input และ output จะใช้เป็นเลขฐาน 8 คือมีแค่ตัวเลข 0 ถึง 7 คือ I0.0 ถึง I0.7, I1.0 ถึง I1.7 ส่วน output คือ Q0.0 ถึง Q0.7, Q1.0 ถึง Q1.7 เป็นต้น เป็นต้น

1.3 S7-1200 modules

S7-1200 modules คือโมดูลเสริมที่ทำงานร่วมกับ CPU S7-1200 ใช้เพื่อเพิ่มฟังก์ชั่นการทำงานต่างๆ , การสื่อสาร เพื่อขยายจำนวน I/O เป็นต้น เราสามารถเพิ่ม modules ให้กับตัว CPU ได้ดังนี้



รูป 1.14

1. **Communication module (CM) และ communication processor (CP)** เป็นการเพิ่มพอร์ตสื่อสารเช่น Profibus หรือ GPRS เป็นต้น ซึ่ง modules เหล่านี้จะติดตั้งที่ด้านข้างซ้ายของ CPU สามารถติดตั้งได้มากสุดจำนวน 3 ตัว

2. **CPU** (CPU 1211C, CPU 1212C, CPU 1214C, CPU 1215C, CPU 1217C)

3. **Signal board (SB)** บอร์ดนี้จะติดตั้งที่ด้านหน้าของตัว CPU โดยติดตั้งได้จำนวน 1 ตัว แบบใดแบบหนึ่ง signal board มี 4 แบบคือ

- 3.1 Digital SB เป็นการเพิ่มบอร์ดที่เพิ่ม I/O จำนวนน้อยๆ
- 3.2 Analog SB เป็นบอร์ดที่รับส่งข้อมูลแบบ analog
- 3.3 Battery Board (BB) เป็นการเพิ่มบอร์ดเพื่อทำ backup ในระยะยาวของ real time clock
- 3.4 Communication board (CB) เป็นบอร์ดแบบสื่อสารเช่น RS485

4. **Signal module (SM)** (digital SM, analog SM, thermocouple SM, RTD SM, technology SM) เป็นการเพิ่ม digital I/O หรือ analog I/O โดย modules เหล่านี้ติดตั้งที่ทางขวาของ CPU โดย CPU1212C ติดตั้งได้มากสุด 3 ตัว ส่วนตัวอื่นๆติดตั้งได้มากสุด 8 ตัว (ยกเว้น CPU 1211C ไม่สามารถใช้ signal module ได้)

สำหรับ Signal board และ Signal module บางครั้งก็เรียกรวมว่า module ก็ได้

จบตัวอย่างบทที่ 1

สนใจซื้ออ่านฉบับเต็มได้ที่ ไลน์ official

@ecy6822d

หรือดูรายละเอียดเพิ่มเติม

www.plcsanook.com

บทที่ 2 ซอฟต์แวร์ STEP 7

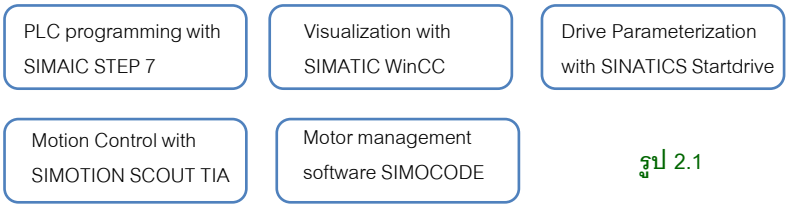
2.1 TIA Portal

หลังจากนี้เป็นต้นไป เราจะถือเป็นรุ่นปัจจุบันทั้งหมด โดยข้อดีของรุ่นใหม่คือ ทุก software จะอ้างอิง platform เดียวกันทั้งหมดคือ TIA Portal

แต่สิ่งที่ยังมีหลายท่านเข้าใจผิดก็คือการเข้าใจว่า TIA Portal เป็น software ประเภท all-in-one คือลง TIA Portal ตัวเดียวแล้วอยากเลือกใช้อะไรก็ได้ ซึ่งเป็นความเข้าใจที่ผิด เพราะจริงๆแล้วไม่มีการติดตั้ง software ที่ชื่อว่า TIA Portal เลย

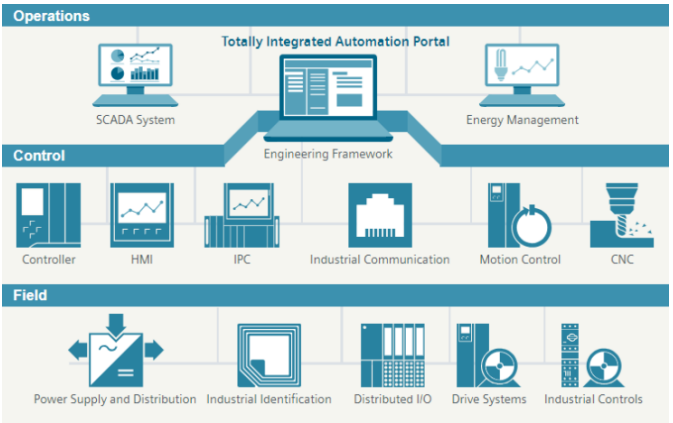
การติดตั้งซอฟต์แวร์สำหรับรุ่นใหม่นั้น เรายังคงจำเป็นต้องลงโปรแกรมแยกกันเหมือนเดิม คือ หากต้องการใช้งาน PLC ให้เราลงโปรแกรม STEP7 (ที่เป็นรุ่นใหม่บน TIA Portal) หรือหากต้องการใช้งาน HMI เราก็ต้องลงโปรแกรม WinCC (ที่เป็นรุ่นใหม่บน TIA Portal) เป็นต้น แต่ไม่ว่าเราจะลงโปรแกรมอะไรก็ตาม ตัว platform TIA Portal จะถูกติดตั้งให้ด้วยอัตโนมัติเสมอ โดยทำหน้าที่เป็น platform ที่ครอบการใช้งาน software ทั้งหมดเอาไว้

Software ทั้งหมดที่ใช้งานบน TIA Portal มีดังนี้



รูป 2.1

ส่วน hardware ทั้งหมดที่สามารถใช้งานบน TIA Portal ก็จะเป็นอุปกรณ์ที่เราได้ใช้งานตามปกติอยู่แล้ว ซึ่งรองรับตั้งแต่ระดับ Field Level จนถึง Operation Level เลยทีเดียว ดังรูป 2.2



รูป 2.2

Version ปัจจุบันของ software บน TIA Portal นั้น ณ ถึงปี 2022 มีถึง V17 แล้ว แต่เนื่องจากเพิ่งออกมาต้นปี 2018 ทำให้ยังไม่ค่อยมีผู้ใช้งานมากนัก ดังนั้น version โดยส่วนใหญ่ที่ใช้กันก็จะเป็น V13, V13 SP1, V13 SP2, V14 และ V14 SP1 เป็นส่วนใหญ่

SIMATIC STEP7

STEP7 เป็น software ที่ใช้เขียน PLC ในรุ่นปัจจุบันคือ S7-1200, S7-1500, S7-300 และ S7-400 โดยแบ่ง software เป็น 2 รุ่นย่อยคือ

STEP7 Basic ใช้เขียน S7-1200 ได้เท่านั้น

STEP7 Professional ใช้เขียน PLC ได้ทุกรุ่น ทั้ง S7-1200, S7-1500, S7-300 และ S7-400

SIMATIC WinCC

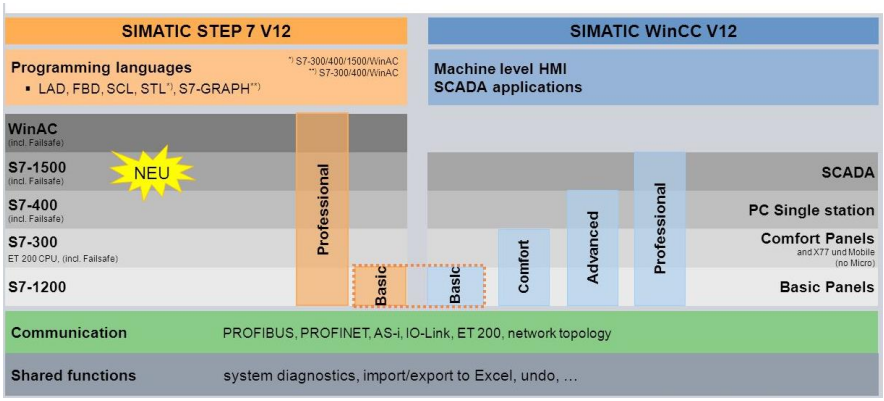
WinCC เป็น software ที่ใช้เขียน HMI/SCADA โดยรุ่นของจอที่เขียนได้ แบ่งตามรุ่นย่อยของ WinCC ดังนี้

WinCC Basic : ใช้เขียน Basic Panel เท่านั้น ซึ่งปกติแล้วตัว WinCC Basic มักไม่ค่อยมีคนซื้อ เพราะว่าหากเราลง STEP7 ไม่ว่าจะเป็น Basic หรือ Professional ก็ตาม จะแถม WinCC Basic มาให้ทันที

WinCC Comfort : ใช้เขียนจอ Comfort Panel (และใช้หลักการรุ่นสูงเขียนรุ่นต่ำได้ คือใช้กับจอ Basic Panel ได้ด้วย)

WinCC Advanced : ใช้เขียนจอ HMI ที่จำลองให้ทำงานบน PC (และใช้เขียน Basic และ Comfort Panel ได้)

WinCC Professional : ถือเป็นตัวสูงสุดทำหน้าที่เป็น SCADA และใช้เขียนจอ HMI ได้ทุกรุ่น

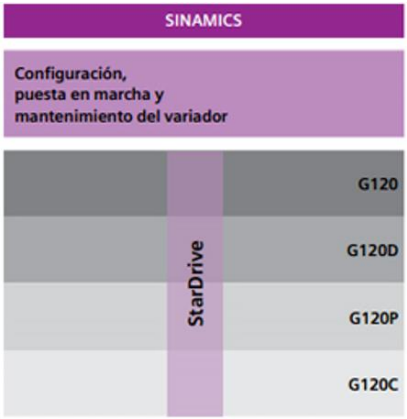


รูป 2.3

รูปที่ 2.3 จะทำให้เราเข้าใจความสัมพันธ์ของทั้ง STEP7 และ WinCC ได้ง่ายขึ้น

StartDrive

StartDrive เป็น software ที่ใช้ตั้งค่าตัว drive/inverter โดยเน้นที่ตระกูล G120 ซึ่งสื่อสารกับ PLC ผ่านทาง Profinet เป็นหลัก



รูป 2.4

หมายเหตุ

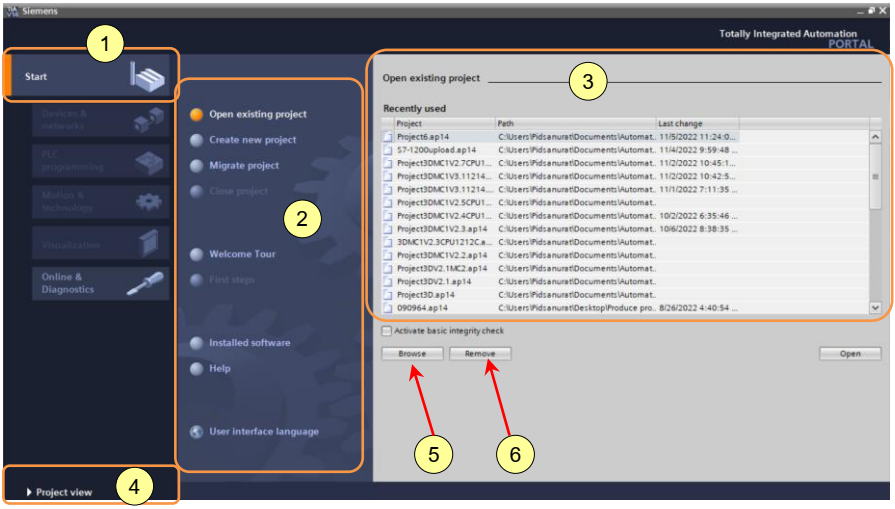
การลง software บน TIA Portal platform นั้น software ทุกตัวที่ลงต้องเป็น version เดียวกัน เช่น

- หากเราลง WinCC Comfort V14 SP1 แล้วต้องการลง StartDrive ก็ต้องลง StartDrive V14 SP1 ด้วย
- หรือหากก่อนหน้าเราใช้งาน STEP7 Basic V13 และ WinCC Comfort V13 ไปแล้ว แล้วเราไปทำการ upgrade STEP7 Basic เป็น V13 SP2 ก็ต้องทำการ upgrade WinCC Comfort ให้เป็น V13 SP2 ด้วย ไม่เช่นนั้นจะไม่สามารถเปิดโปรแกรมใดๆได้เลย

2.2 SIMATIC STEP7

หัวข้อนี้เป็นการแนะนำซอฟต์แวร์ STEP 7 เบื้องต้น เมื่อดับเบิลคลิกไอคอน TIA Portal จะได้หน้าต่าง Portal view ดังรูป 2.6

รูป 2.5



รูป 2.6

Portal view มีส่วนประกอบคือ

① Portals for the different tasks

② Tasks for the selected portal ใช้สำหรับเลือกโหมดต่างๆ เช่นเปิดโปรเจคที่มีอยู่ สร้างโปรเจคใหม่ เป็นต้น

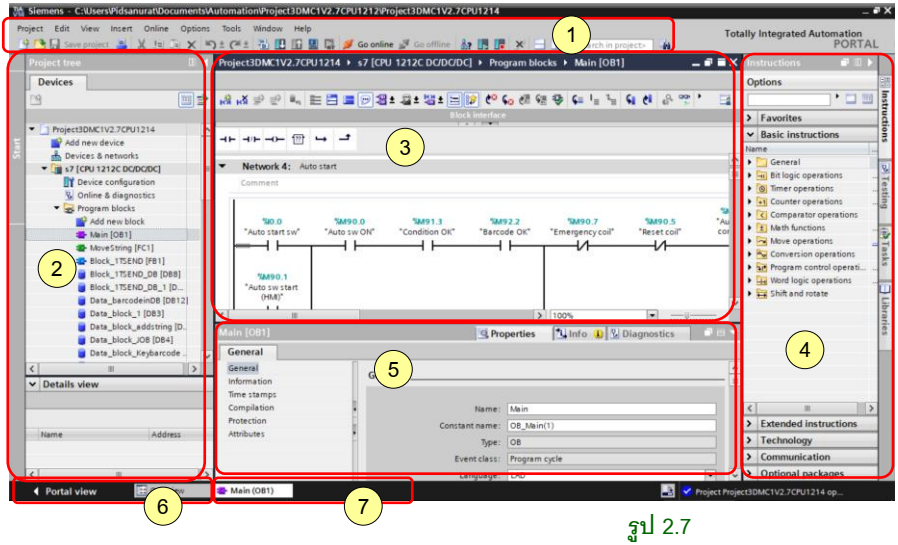
③ Selection panel for the selected action คือ list แสดงรายการชื่อโปรเจคที่เราสร้างไว้ในโฟลเดอร์ เมื่อคลิกที่ชื่อโปรเจคและเลือก Open จะเป็นการเปิด project นั้นขึ้นมา

④Changes to the Project view สำหรับเปลี่ยนไปยัง project view (ปุ่มนี้ใช้สำหรับการเข้าไปยังหน้าจอโปรแกรมที่เราเปิดจากไฟล์เดิม หรือจอโปรแกรมที่เราสร้างโปรเจคขึ้นมาใหม่)

⑤ ปุ่ม (ไอคอน) Browse เมื่อคลิกไอคอนจะเป็นการเปิดโฟลเดอร์ที่เก็บไฟล์โปรแกรมที่เรา save ไว้

⑥ ปุ่ม Remove เป็นปุ่มสำหรับลบรายชื่อโปรแกรมที่แสดงในข้อ 3 เท่านั้น แต่ไม่ได้ลบไฟล์ต้นฉบับที่เก็บในโฟลเดอร์

กรณีเปิด project ขึ้นมา (project ที่มีข้อมูล) จะได้หน้าจอดังรูปที่ 2.7 เรียกว่า Project view



รูป 2.7

Project view มีส่วนประกอบคือ

① Menus และ toolbar

② Project navigator

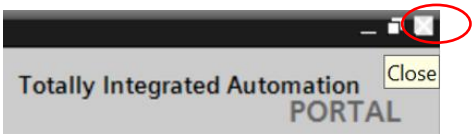
③ Work area ใช้สำหรับเขียนโปรแกรม

④ Task cards ตัวช่วยสำหรับเรียกใช้งานคำสั่งต่างๆ

⑤ Inspector window

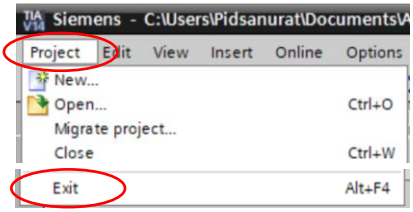
⑥ Changes to the Portal view

⑦ Editor bar เป็นแท็บแสดงหน้าจอต่างๆที่เราเปิดขึ้นมา



รูป 2.8

รูปที่ 2.8 กรณีต้องการปิดโปรแกรม STEP 7 ทำได้โดยคลิกเครื่องหมายกากบาทที่มุมขวาบนของซอฟต์แวร์



รูป 2.9

หรือคลิกที่เมนู Projectแล้วกด Exit

จบตัวอย่างบทที่ 2

สนใจซื้ออ่านฉบับเต็มได้ที่ ไลน์ official

@ecy6822d

หรือดูรายละเอียดเพิ่มเติม

www.plcsanook.com

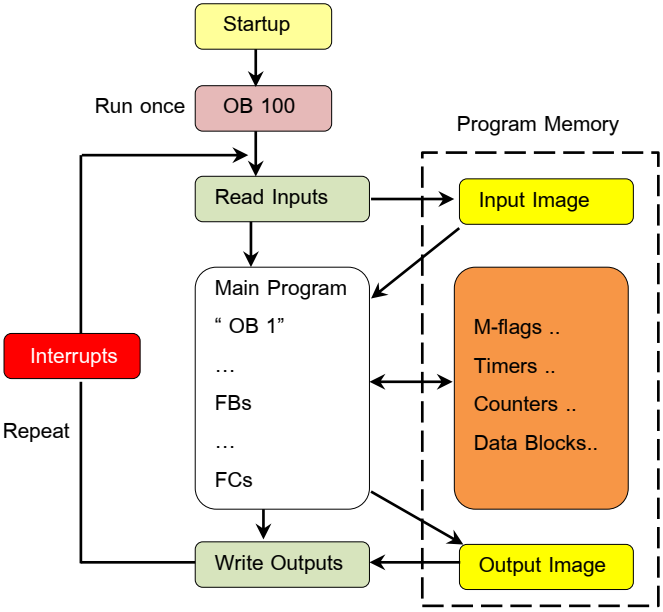
บทที่ 3 PLC Concept

3.1 พื้นฐานการทำงานของ PLC

ในแต่ละ scan cycle ของ PLC นั้นประกอบด้วยการเขียน outputs, อ่าน inputs, ทำงานตามโปรแกรมที่ได้เขียนไว้ และทำในส่วนของ system maintenance หรือ background processing

ภายใต้เงื่อนไขปกติ ทั้ง digital และ analog I/O จะถูก update ข้อมูลพร้อมๆกันในแต่ละ scan cycle โดยอาศัย internal memory ที่เรียกว่า process image ดังนั้น process image จะประกอบด้วยการจับสัญญาณอย่างรวดเร็ว (snapshot) ของ inputs และ outputs จริงๆของ CPU, signal board และ signal module ต่างๆ

- CPU จะทำการอ่าน inputs จริงๆ ก่อนที่จะทำการ execute user program และเก็บค่า input เหล่านี้ไว้ใน process image input area (input image) เพื่อให้มั่นใจได้ว่าค่าต่างๆ เหล่านี้จะยังคงค่าเดิมตลอดการ execution ของ user instructions
- CPU ทำงานตาม logic ที่เขียนไว้ใน user instructions และทำการ update ค่า output ใน process image output area (output image) แทนที่จะเขียนค่าลงไปที่ outputs จริงๆ โดยตรง
- หลังจากที่ทำคำสั่งต่างๆใน user program แล้ว CPU จะทำการเขียนค่า output จาก process image output area ไปยัง outputs จริงๆ



รูป 3.1

Warm restart : จะทำการ initialize all non-retentive system และ user data แต่ไม่ได้รวมถึงการทำ memory reset

Memory reset : คือการทำ clear all work memory, clear retentive & non-retentive memory area, copy load memory to work memory และ set output ให้เป็นสถานะที่ตั้งไว้ที่ “Reaction to CPU STOP” (Properties -> General -> Digital outputs)

Execution of the user program (การดำเนินการของโปรแกรมผู้ใช้)

PLC รองรับ Code block ประเภทต่อไปนี้ ที่ทำให้คุณสร้างโครงสร้างโค๊ดที่มีประสิทธิภาพได้สำหรับโปรแกรมผู้ใช้ของคุณ

OBs : (Organization Blocks) เอาไว้ใช้สำหรับเขียนโปรแกรม OB บางตัวเอาไว้ใช้ในหน้าที่พิเศษตาม event ต่างๆ เช่น interrupt เป็นต้น แต่เราก็สามารถสร้าง OB สำหรับ custom event ได้เช่นกัน

> OB ไม่สามารถเรียกใช้ OB กันเองได้ และไม่สามารถถูกเรียกโดย FC หรือ FB ได้ เฉพาะ event เท่านั้นที่สามารถเรียกใช้ OB ได้ เช่น diagnostic interrupt หรือ time interval
>
> CPU จะเรียกใช้ OB ตาม priority class โดย priority OB ที่สูงกว่าจะเรียกใช้ก่อนตัวที่ priority ต่ำกว่า โดย priority class ที่ต่ำที่สุดคือ 1 (ถูกใช้กับ main program cycle) และสูงสุดคือ class 26 (สำหรับ firmware V4.0 เป็นต้นไป เราสามารถเปลี่ยน priority class ในแต่ละ OB ได้โดยไปตั้งใน attributes ของ OB properties)
>
> เนื่องจาก CPU processing จะทำงานตาม event โดยในแต่ละ event จะไปทำการ trig เพื่อให้ interrupt OB แต่ละตัวทำงาน เราสามารถระบุ interrupt OB จาก event โดยการสร้าง block หรือตอนทำ device configuration หรือด้วย ATTACH / DETACH instruction

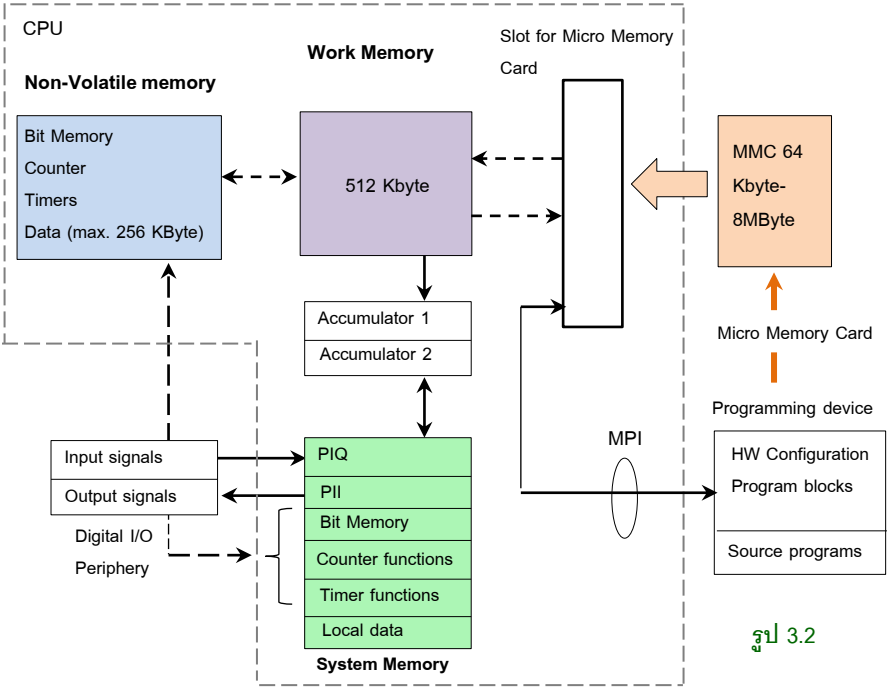
FCs & FBs (Functions & Function Blocks) : ประกอบด้วย program code อยู่ภายในการใช้งาน FB จะไป link กับ Data Block (เรียกว่า Instance DB) ด้วยเพื่อที่จะใช้ Data Block ในการรักษาค่า state ต่างๆเอาไว้ระหว่างการทำงาน

Data Blocks : เป็นตัวเก็บข้อมูล และสามารถถูกใช้โดย program block ได้

Memory Area (User memory)

หน่วยความจำ CPU ของ Siemens S7 หรือเรียกว่า User memory ประกอบด้วย 3 ส่วนคือ

Work Memory : เป็นพื้นที่แบบ volatile storage สำหรับบาง element ของตัว user project ขณะที่มีการทำงานของ user program (Code work memory: FC, FB, OB — Data work memory: Global DB, Instance DB, Technology objects) CPU จะทำการ copy ค่าเหล่านี้จาก Load Memory ไปยัง Work Memory ในขณะที่กำลังทำงาน user program และเนื่องจากเป็น volatile area จึงทำให้ไม่สามารถจดจำค่าได้เมื่อไฟดับ



รูป 3.2

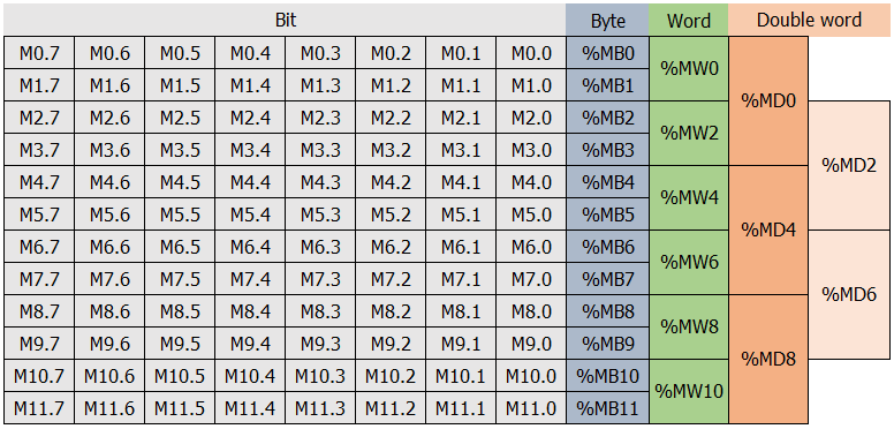
Load Memory: เป็น non-volatile storage ที่ใช้สำหรับ user program, data และ configuration เมื่อมีการ download project ลง CPU (FC,FB, OB, DB, Hardware configuration, Technology objects) มันจะทำการโหลดลง Load Memory เป็นอันดับแรก พื้นที่ Load memory นี้สามารถเป็นได้ทั้ง memory card (ถ้ามี) หรือพื้นที่ใน CPU เองก็ได้ (แต่ memory card จะรองรับพื้นที่มากกว่า) เนื่องจากเป็น non-volatile memory area จึงทำให้ไม่สูญหายแม้ไม่มีการจ่ายไฟให้มัน

Retentive Memory: เป็น non-volatile storage ที่อยู่ในส่วนหนึ่งของ Work Memory โดยจะใช้เพื่อเก็บค่าของ user memory เอาไว้ไม่ให้หายเมื่อไฟดับ (Global DB, Instance DB, Technology objects, bit memories, timers, counters) โดย CPU จะทำการคืนค่าให้ retentive value อีกครั้งเมื่อมีการจ่ายไฟมาใหม่

3.2 การจัดเรียง Address ของ PLC

การกำหนด address ที่เป็นพวก %M นั้น หลายๆท่านมักมีปัญหาในการใช้งานการระบุ address แล้วเกิดพื้นที่ทับซ้อนกันเพราะไม่เข้าใจโครงสร้างและวิธีการระบุ address ที่ถูกต้อง ซึ่งส่งผลให้โปรแกรมทำงานผิดพลาดแล้วจะหาสาเหตุของปัญหายากมาก

ดังนั้นเราจึงต้องเข้าใจโครงสร้าง address ของ PLC Siemens เสียก่อน ซึ่งหลักการนี้สามารถประยุกต์ใช้ได้กับ PLC ทุกรุ่นของ Siemens ตั้งแต่รุ่นแรกไปจนถึงรุ่นล่าสุดเลย โดยอาศัยเพียงรูป 3.3 นี้เท่านั้น



Bit								Byte	Word	Double word	
M0.7	M0.6	M0.5	M0.4	M0.3	M0.2	M0.1	M0.0	%MB0	%MW0	%MD0	
M1.7	M1.6	M1.5	M1.4	M1.3	M1.2	M1.1	M1.0	%MB1			
M2.7	M2.6	M2.5	M2.4	M2.3	M2.2	M2.1	M2.0	%MB2	%MW2		%MD2
M3.7	M3.6	M3.5	M3.4	M3.3	M3.2	M3.1	M3.0	%MB3			
M4.7	M4.6	M4.5	M4.4	M4.3	M4.2	M4.1	M4.0	%MB4	%MW4	%MD4	
M5.7	M5.6	M5.5	M5.4	M5.3	M5.2	M5.1	M5.0	%MB5			
M6.7	M6.6	M6.5	M6.4	M6.3	M6.2	M6.1	M6.0	%MB6	%MW6		%MD6
M7.7	M7.6	M7.5	M7.4	M7.3	M7.2	M7.1	M7.0	%MB7			
M8.7	M8.6	M8.5	M8.4	M8.3	M8.2	M8.1	M8.0	%MB8	%MW8	%MD8	
M9.7	M9.6	M9.5	M9.4	M9.3	M9.2	M9.1	M9.0	%MB9			
M10.7	M10.6	M10.5	M10.4	M10.3	M10.2	M10.1	M10.0	%MB10	%MW10		
M11.7	M11.6	M11.5	M11.4	M11.3	M11.2	M11.1	M11.0	%MB11			

รูป 3.3

รูป 3.3 เป็นการอธิบายการกินพื้นที่ของ address แต่ละตัวได้อย่างครบถ้วนที่สุด ถ้าหากเราเข้าใจการกินพื้นที่นี้แล้ว โอกาสที่จะเกิดพื้นที่ทับซ้อนโดยไม่ตั้งใจนั้น แทบจะไม่มีเลย

ก่อนอื่น สิ่งที่เราต้องเข้าใจพื้นฐานของ PLC Siemens คือ

1. Siemens ใช้พื้นที่ทุกอย่างทั้ง bit, byte, word, double word ร่วมกัน

ผลจากหลักการนี้ ทำให้การใช้งานพื้นที่ Input, Output และ Internal memory แตกต่างจากยี่ห้ออื่นๆ เช่น Mitsubishi ใช้ M0, M1, M2, … แทนการใช้งานแบบ Bit และใช้พื้นที่ D0, D1, D2, … แทนการใช้งานแบบ Word หรือ Double word ดังนั้นจะไม่เกิดการปนกันของการใช้งาน bit กับ word เลย

ในขณะที่ Siemens ใช้พื้นที่ทุกอย่างร่วมกัน ไม่ว่าจะเป็น bit, byte, word หรือ double word ดังนั้นจึงมีโอกาสที่จะเกิดการใช้งานข้อมูลเหล่านี้ทับซ้อนกันได้

2. Siemens ถือกำเนิดมาจากโครงสร้างแบบ Byte

ด้วยหลักการนี้ เราจะเห็นว่า %MB จะเป็น address หลักที่มีการเรียงตั้งแต่ %MB0, %MB1, %MB2, %MB3, … ไปเรื่อยๆ

กรณีที่เราต้องการใช้งานแบบ Bit ก็จะเป็นการใช้งานแต่ละ bit ในแต่ละ byte ดังนั้น

%MB0 จะประกอบไปด้วย %M0.0, %M0.1, %M0.2, ….. , %M0.6 และ %M0.7
%MB1 จะประกอบไปด้วย %M1.0, %M1.1, %M1.2, ….. , %M1.6 และ %M1.7
%MB2 จะประกอบไปด้วย %M2.0, %M2.1, %M2.2, ….. , %M2.6 และ %M2.7
และเป็นแบบนี้ไปเรื่อยๆ จึงจะเห็นว่า ไม่มี %M0.8, %M0.9 เพราะ 1 byte มีแค่ 8 bit เท่านั้น และในการใช้งานแบบบิตจะใช้เป็นเลขฐาน 8 คือมีแค่ 0 ถึง 7 แต่กรณีใช้งานแบบ %MB นั้น จะนับเป็นเลขฐาน 10 อยู่แล้วเช่น %MB8, %MB9 เป็นต้น

กรณีที่เราต้องการใช้งานแบบ Word ก็จะเป็นการเอา 2 byte มารวมกัน เช่น

%MW0 คือการรวมกันของ %MB0(MSB) และ %MB1(LSB)
%MW2 คือการรวมกันของ %MB2(MSB) และ %MB3(LSB)
%MW4 คือการรวมกันของ %MB4(MSB) และ %MB5(LSB)
และเป็นแบบนี้ต่อไปเรื่อยๆ จึงจะเห็นว่า address ของ %MW จะกระโดดไปทีละ 2 เสมอ

กรณีที่เราต้องการใช้งานแบบ Double Word ก็จะเป็นการเอา 4 byte (หรือ 2 word) มารวมกัน เช่น

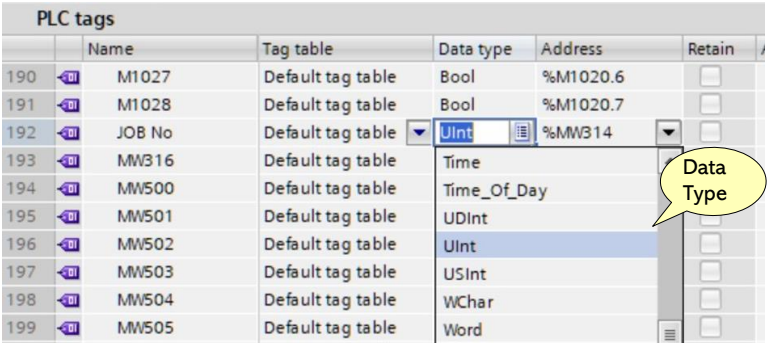
%MD0 คือการรวมกันของ %MB0-%MB3 (หรือ %MW0-%MW2)
%MD4 คือการรวมกันของ %MB4-%MB7 (หรือ %MW4-%MW6)
%MD8 คือการรวมกันของ %MB8-%MB11 (หรือ %MW8-%MW10)
และเป็นแบบนี้ต่อไปเรื่อยๆ จึงจะเห็นว่า address ของ %MD จะกระโดดไปทีละ 4 เสมอ

หมายเหตุ

-ในการอธิบายเนื้อหาในหนังสือ เพื่อความสะดวกอาจจะตัดตัวอักษร % ออก เช่น %MD0 จะเขียนสั้นๆว่า MD0 เป็นต้น
-MSB หมายถึงคำเรียก byte ต่ำ ส่วน LSB คือไบต์สูง

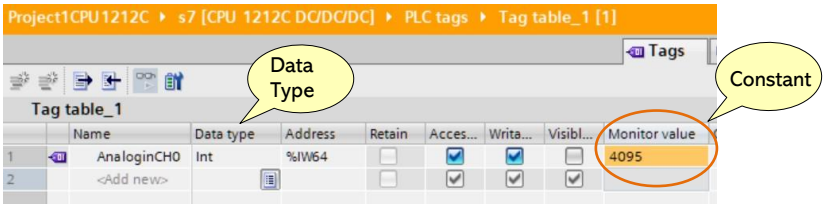
3.4 ประเภทข้อมูล (Data Type)

ประเภทของข้อมูลของตัวแปรมีหลายๆแบบ การตั้งค่า data type สามารถตั้งค่าได้ที่ PLC tag โดยเลือกที่ data type



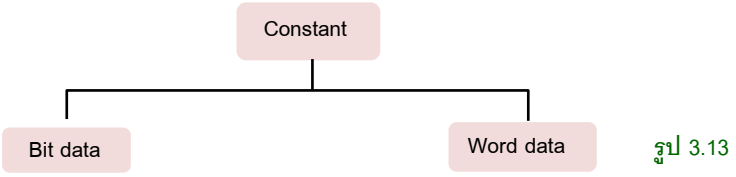
รูป 3.11

รูปที่ 3.12 ไม่ว่าประเภทของข้อมูลของตัวแปรจะใช้เป็นแบบไหน ข้อมูลเหล่านี้ก็คือตัวเลขชนิดหนึ่ง ซึ่งเรียกรวมทั้งหมดว่า ค่าคงที่ (constant)



รูป 3.12

ดังนั้น ค่าคงที่ (constant) คือข้อมูลที่ใช้ในโปรแกรม PLC ถ้าเราไม่สนใจรายละเอียดปลีกย่อยต่างๆแล้ว ค่าคงที่ มีสองแบบใหญ่ๆคือค่าคงที่แบบบิต (bit data) และแบบเวิร์ด (word data) เท่านั้น



รูป 3.13

ค่าคงที่แบบเวิร์ดในที่นี้คือค่าคงที่ที่มีมากกว่า 1 บิตซึ่งอาจหมายถึง byte , word หรือ Double world เป็นต้น

Integer

Integer (จำนวนเต็ม) คือข้อมูลที่เป็นจำนวนเต็มเท่านั้น มีค่าได้ทั้งบวกและลบ ไม่มีจุดทศนิยม และไม่มีอักษร A ถึง F , Integer แบ่งเป็น 3 แบบคือ

1. **Short Integer** , address ที่ใช้ได้เช่น MB, Data block

   USInt (unsigned 8-bit integer), ขนาด 8 บิต ไม่มีเครื่องหมาย +, -

   SInt (signed 8-bit integer) ขนาด 8 บิต มีเครื่องหมาย +, -

2. **Integer** , address ที่ใช้ได้เช่น MW, Data block

   UInt (unsigned 16-bit integer) ขนาด 16 บิต ไม่มีเครื่องหมาย +, -

   Int (signed 16-bit integer) ขนาด 16 บิต ไมีเครื่องหมาย +, -

3. **Double Integer** , address ที่ใช้ได้เช่น MD, Data block

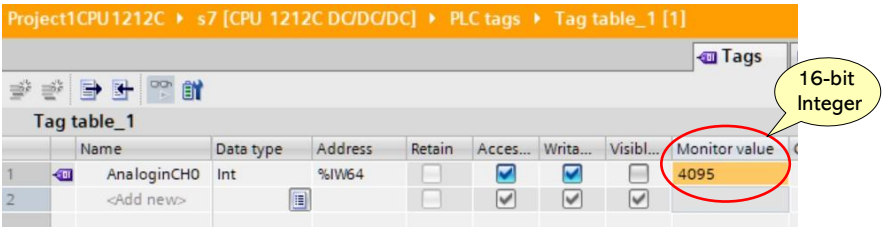
   UDInt (unsigned 32-bit integer) ขนาด 32 บิต ไม่มีเครื่องหมาย +, -

   DInt (signed 32-bit integer) ขนาด 32 บิต มีเครื่องหมาย +, -

ตาราง 3.2 แสดงประเภทข้อมูล , rang ของหมายเลข และตัวอย่างการใช้งาน

Data type	Bit size	Number Range	Constant examples	Address examples
USInt	8	0 to 255	78, 2#01001110	MB0, DB1.DBB4, Tag_name
SInt	8	-128 to 127	+50, 16#50	
UInt	16	0 to 65,535	65295, 0	MW2, DB1.DBW2, Tag_name
Int	16	-32,768 to 32,767	30000, +30000	
UDInt	32	0 to 4,294,967,295	4042322160	MD6, DB1.DBD8, Tag_name
DInt	32	-2,147,483,648 to 2,147,483,647	-2131754992	

จากตาราง ข้อมูลแบบ USInt มีค่าตั้งแต่ 0 ถึง 255, USInt มีค่าตั้งแต่ -128 ถึง 127 เป็นต้น



รูป 3.14

รูปที่ 3.14 แสดงข้อมูลแบบ 16-bit Integer (หรือใช้ตัวย่อ Int) กรณีไม่ต้องการค่าที่เป็นลบต้องตั้งค่าเป็น UInt

จบตัวอย่างบทที่ 3
สนใจซื้ออ่านฉบับเต็มได้ที่ ไลน์ official
@ecy6822d
หรือดูรายละเอียดเพิ่มเติม
www.plcsanook.com

บทที่ 4 Device configuration เบื้องต้น

หัวข้อ Device configuration เบื้องต้นนี้ ประกอบด้วยเนื้อหาตั้งแต่การทำ upload PLC ทั้งแบบการ upload เป็น new station คือดึงทุกอย่างมาหมด และ Upload แบบ unspecified CPU คือดึง hardware configuration ของจริงขึ้นมาแสดงทั้งหมด

รวมถึงการเปลี่ยนรุ่น CPU, การเพิ่ม module ด้วยตัวเอง, การตั้งค่า system memory และ clock memory, การตั้งค่า IP address ให้กับ PLC และการตั้ง protection ให้กับ PLC ในแบบต่างๆเป็นต้น

4.1 การ Upload ข้อมูลจาก CPU

รูป 4.1

การ upload คือการอ่านข้อมูลจาก CPU มายังโปรแกรม STEP 7 โดยมี 2 วิธีที่จะทำการ upload CPU ขึ้นมา คือ

1. **Upload เป็น new station** : เหมาะกับการดึงทุกอย่างของ PLC ขึ้นมา ทั้ง hardware configuration และโปรแกรมที่อยู่ใน PLC ทั้งหมด

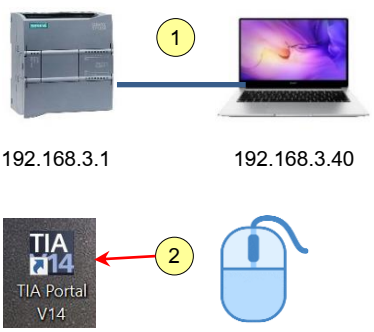
2. **Upload เป็น unspecified CPU เพื่อตรวจสอบ hardware configuration**: เหมาะกับการทำ project ใหม่ขึ้นมาเลย แล้วไม่ต้องการ config ตัว hardware ทุกอย่างเองแต่จะให้โปรแกรมวิ่งไปตรวจสอบ hardware จริงๆแล้วดึง hardware ทุกอย่างขึ้นมาเลย ทำให้มั่นใจได้ว่าเราไม่ config ตัว hardware ผิดไปแน่ๆ

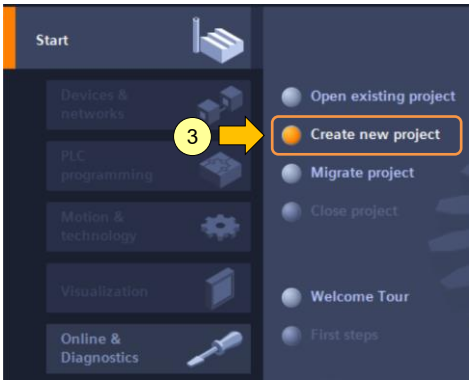
วิธีนี้ดึงมาได้แค่ configuration เท่านั้น โปรแกรมใน CPU ไม่ได้ถูกดึงมาด้วย แต่ข้อดีคือเราไม่ต้องการเลือก CPU และ module ต่างๆด้วยตัวเอง เพราะ software จะดึงมาให้หมด ทำให้ไม่เกิดปัญหาการตั้งค่า module ใน software ผิดกับที่ใช้งานจริง

การ upload วิธีแรกจะได้มาทั้ง hardware configuration และโปรแกรมของ CPU ส่วนวิธีที่สองจะได้มาแค่ hardware configuration ของ PLC ไม่ว่าตัว PLC จะต่อกับ module อะไรไว้บ้างก็ตามจะแสดงมาหมด ทำให้เราไม่ต้องการตั้งค่า hardware เองซึ่งมีโอกาสที่จะใส่ผิดได้ โดยทั้งสองแบบจะต้องทำการสร้าง new project ก่อน

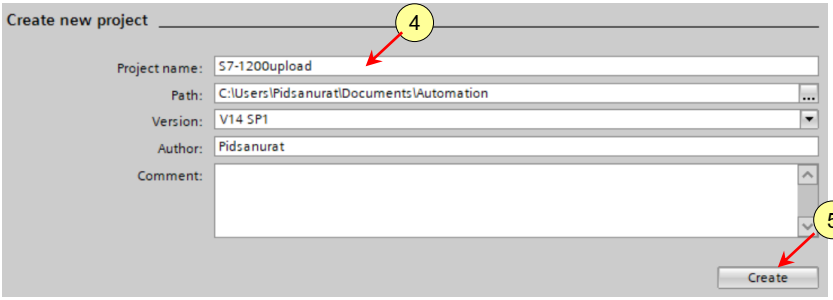
1. การสร้าง New project และ Upload เป็น new station

1. ต่อสาย LAN ระหว่าง PC กับ PLC ⇨ 2. ดับเบิลคลิก TIA Portal ⇨ 3. Create new project



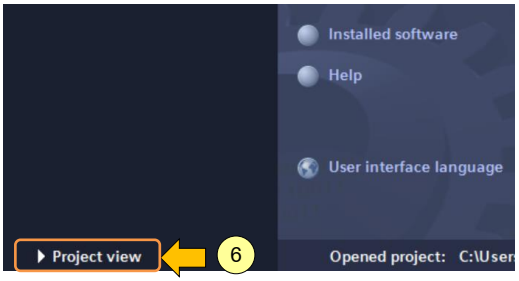


รูป 4.2



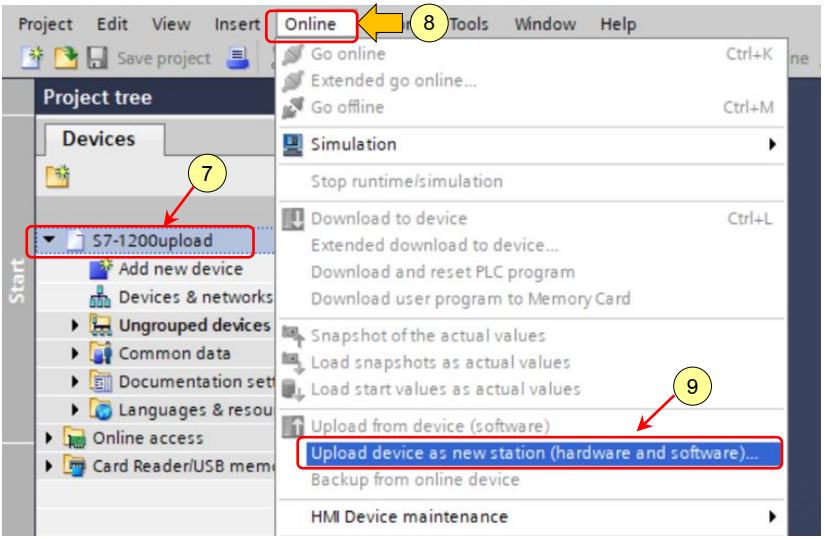
รูป 4.3

4. ตั้งชื่อ Project ตัวอย่างคือ S7-1200upload ⇨ 5. คลิก Create



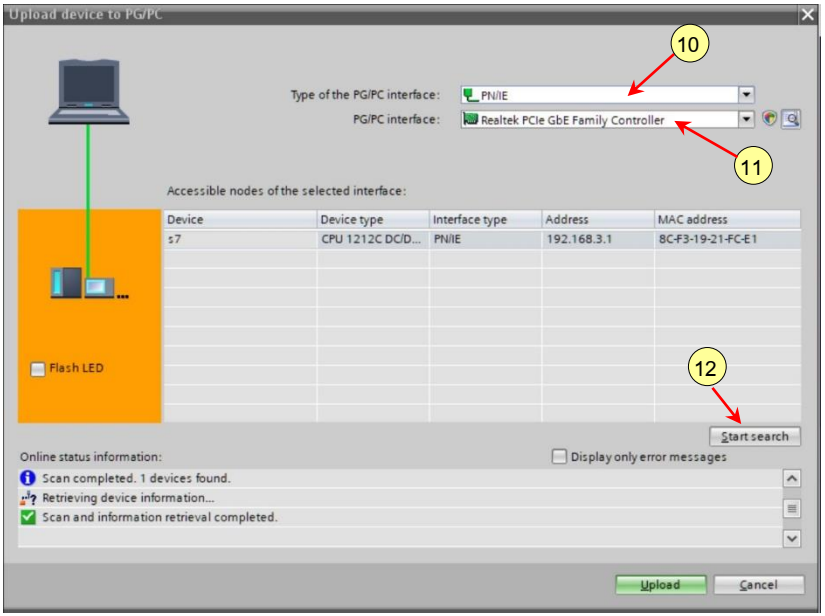
รูป 4.4

6. คลิก Project view เพื่อเข้าสู่โปรแกรมที่เราสร้างขึ้น



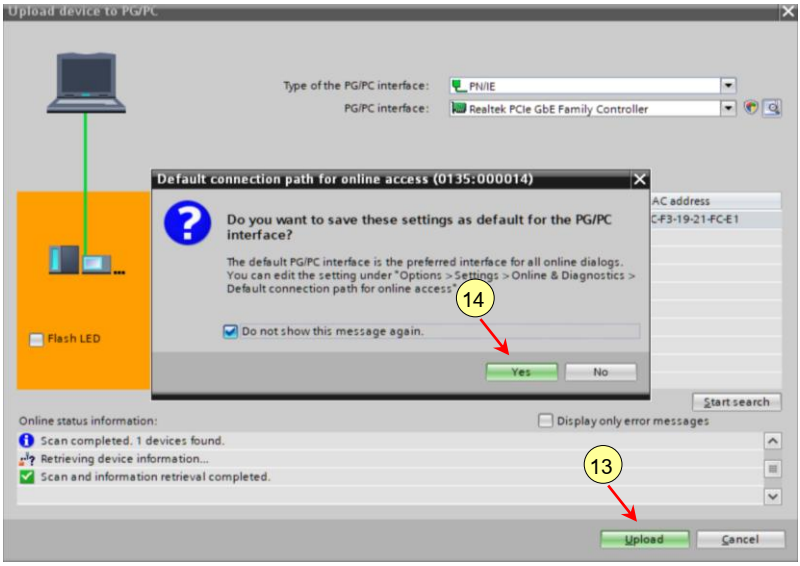
รูป 4.5

7. คลิกชื่อโปรเจค ⇨ 8. คลิกเมนู Online ⇨ 9. คลิก Upload device as new station



รูป 4.6

10. เลือกประเภทของ PG/PC interface ⇨ 11. เลือกอุปกรณ์เชื่อมต่อ (อาจเป็น port LAN แบบ built in หรือใช้ USB to LAN adapter) ⇨ 12. คลิก Start search โปรแกรมจะทำการค้นหา PLC เมื่อเจอ PLC ที่ต้องการแล้วให้เลือกที่ PLC ตัวนั้น

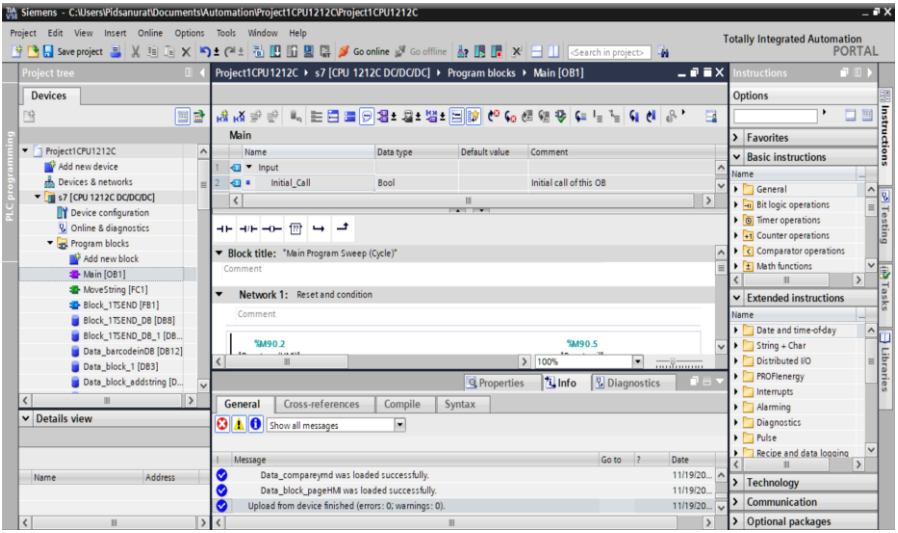


รูป 4.7

13. คลิก Upload

14. คลิก Yes

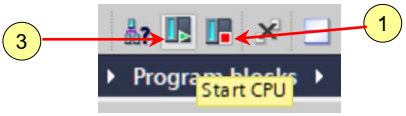
จากนั้นโปรแกรมจะทำการ upload ข้อมูลจาก PLC มาทั้ง hardware configuration และ program blocks ที่ถูกเขียนไว้ใน PLC ทุกอย่างขึ้นมา รูปที่ 4.8 แสดงโปรแกรมที่ upload จาก PLC



รูป 4.8

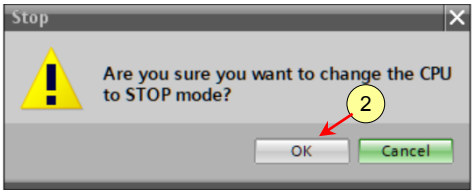
4.6 การ RUN และ STOP CPU

การ RUN CPU (start CPU) คือการทำให้ PLC เริ่มการประมวลผลโปรแกรม ส่วนการ STOP CPU คือการหยุดการประมวลผล CPU เราสามารถควบคุมการ RUN และ STOP CPU ผ่านซอฟต์แวร์ได้โดยคลิกไอคอนดังรูป 4.35



รูป 4.35

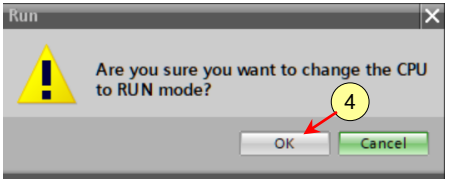
1. กรณีคลิกไอคอน Stop CPU จะมีหน้าต่างแจ้งเตือนก่อน ดังรูปที่ 4.36
2. จากนั้นคลิก OK หลังจากที่ STOP CPUแล้ว สถานะของ PLC จะ STOP ตลอดเวลาแม้ว่าจะทำการปิดและเปิดแหล่งจ่ายไฟใหม่ก็ตาม กรณี PLC STOP หลอดไฟ RUN/STOP จะเป็นสีส้มดังรูป 4.37



รูป 4.36

รูป 4.37

3. กรณีคลิกไอคอน Start CPU จะมีหน้าต่างแจ้งเตือนก่อนเช่นกัน จากนั้น 4. คลิก OK หลังจากที่ Start CPU ไปแล้ว สถานะของ PLC จะ RUN ตลอดเวลาแม้ว่าจะทำการปิดและเปิดแหล่งจ่ายไฟใหม่ก็ตาม ยกเว้น PLC มีการ Error



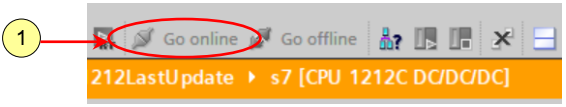
รูป 4.38

รูป 4.39

รูปที่ 4.39 กรณี PLC RUN หลอดไฟ RUN/STOP จะเป็นสีเขียว

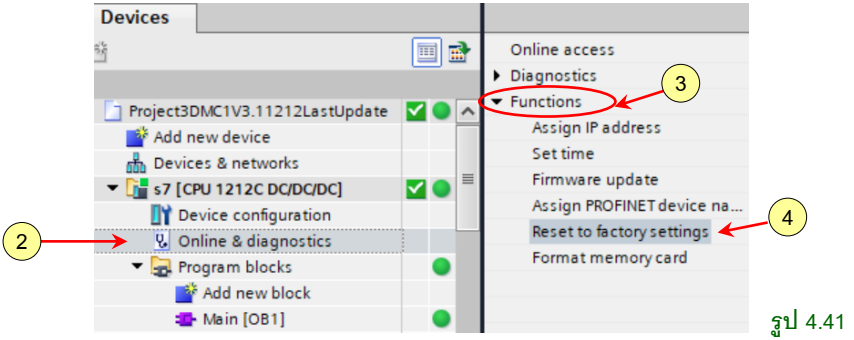
4.7 การทำ Factory reset

กรณีต้องการลบข้อมูลใน PLC ทั้งหมด ให้กลับไปค่าโรงงาน สามารถทำได้ดังนี้



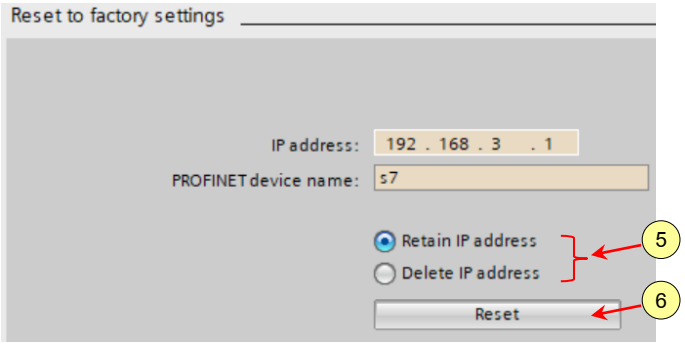
รูป 4.40

1. คลิกไอคอน Go online



รูป 4.41

2. ดับเบิลคลิก Online & diagnostics
3. เลือก Functions
4. เลือก Reset to factory settings

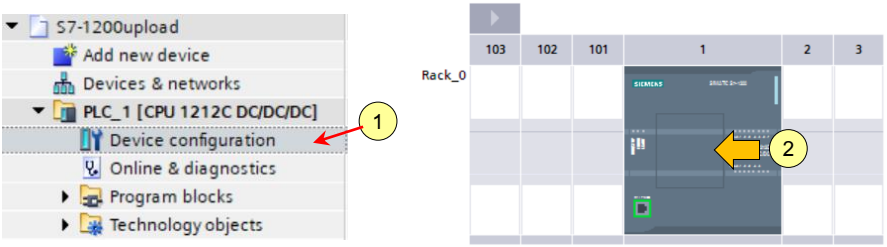


รูป 4.42

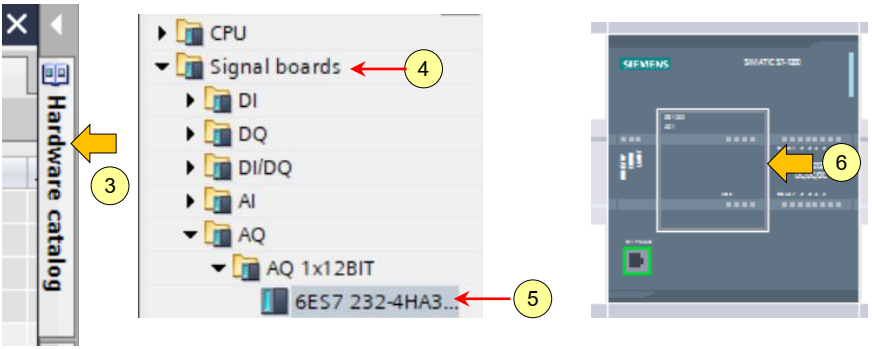
5. เลือกว่าจะคง IP address เดิมไว้หรือไม่ ในรูป 4.42 เลือกคง IP ไว้
6. คลิก Reset จากนั้นข้อมูลใน PLC จะถูกลบทั้งหมด

4.8 การตั้งค่าการเพิ่ม modules และการลบ modules

ตัวอย่างเช่น ต้องการเพิ่ม Analog output (SB) เราสามารถตั้งค่าเพิ่ม modules ให้กับตัว CPU ได้ดังนี้ 1. ดับเบิลคลิก Device configuration



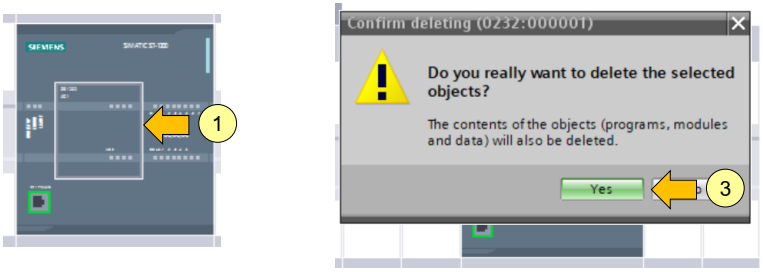
รูป 4.43



รูป 4.44

2. คลิกที่ slot สำหรับติดตั้ง SB ⇨ 3. เลือก hardware catalog ⇨ 4. เลือก Signal board ⇨ 5. ดับเบิลคลิกรุ่นของ AQ ⇨ 6. จะได้ signal board ที่ตัว CPU

สำหรับการลบ modules ทำได้ดังนี้

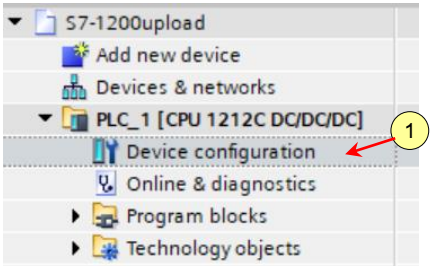


รูป 4.45

1. คลิกที่ module ที่ต้องการลบ ⇨ 2. กดปุ่ม Delete ที่คีย์บอร์ด ⇨ 3. คลิก Yes

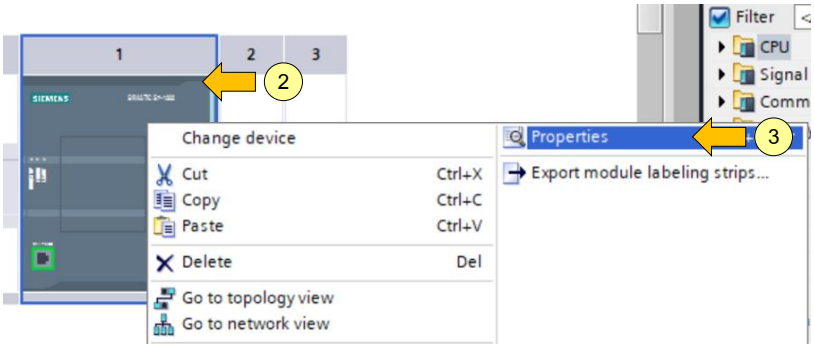
4.9 การกำหนดค่าการดำเนินการของ CPU และ Module

การกำหนดค่าการดำเนินการของ CPU และ Module (Configuring the operation of the CPU and modules) หัวข้อในส่วนนี้จะเน้นไปที่การตั้งค่า Properties ของตัว CPU เป็นหลัก ว่าสามารถตั้งค่าอะไรได้บ้าง โดยสามารถทำได้ดังนี้



1. ดับเบิลคลิกที่ Device Configuration
2. คลิกขวาที่ CPU
3. เลือก Properties

รูป 4.46



รูป 4.47

จากนั้นก็จะเป็นการตั้งค่าตามหัวข้อต่างๆ ดังนี้

1. Configuring the STOP-to-RUN operation of the CPU

เมื่อไรก็ตามที่มีการเปลี่ยน state ของการ operate จาก STOP -> RUN สิ่งที่เกิดขึ้นคือ CPU จะทำการ clear process image inputs, initializes the process image outputs และ process ของ startup Obs

นั่นหมายความว่าการอ่านค่าใดๆจาก process-image inputs ด้วยคำสั่งใน startup OBs จะอ่านค่าเป็นศูนย์ แทนที่จะเป็นค่า input จริงๆจาก physical input ดังนั้นเพื่อที่เราจะสามารถอ่านค่าของ physical input ระหว่าง startup ได้เราต้องทำการใช้งาน immediate read แทน จากนั้น startup OBs และ FC/FB อื่นๆจะทำงานถัดไป หากในโปรแกรมเรามี startup OBs มากกว่า 1 ตัว โปรแกรมจะทำตาม priority คือทำงานตัวที่ใน ตัวเลข OB number น้อยที่สุดก่อน

จบตัวอย่างบทที่ 4

สนใจซื้ออ่านฉบับเต็มได้ที่ ไลน์ official

@ecy6822d

หรือดูรายละเอียดเพิ่มเติม

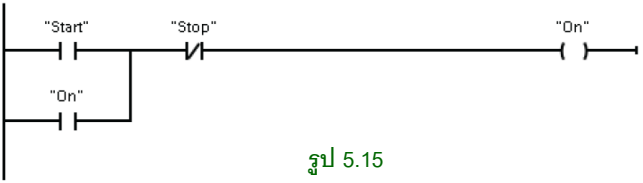
www.plcsanook.com

5.4 Programming Languages

เมื่อเราได้ทำการสร้าง code block ขึ้นมาแล้วเราต้องเลือกภาษาที่ต้องการเขียนด้วย และในโปรแกรมของเราก็สามารถประกอบขึ้นด้วยภาษาหลายๆภาษาประกอบกันก็ได้ STEP7 มีรูปแบบภาษาของการ programming มาตรฐานให้หลักๆดังนี้

Ladder Logic (LAD)

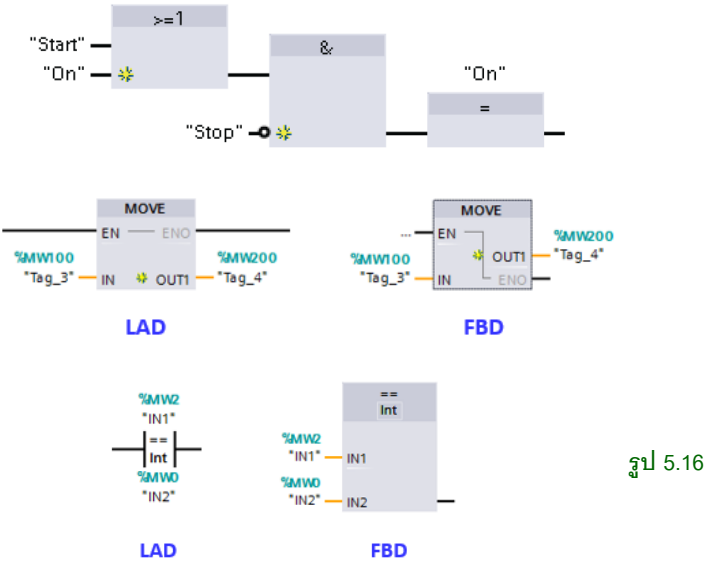
LAD (Ladder Logic หรือ Ladder diagram) เป็นภาษาที่ผู้เริ่มต้นเขียน PLC คุ้นเคยมากที่สุด เพราะเป็นภาษาที่ทำการเลียนแบบมาจาก circuit diagram ที่ง่ายต่อความเข้าใจ เช่น contact NO, NC และ output เป็นต้น



รูป 5.15

Function Block Diagram (FBD)

เป็นการโปรแกรมแบบกราฟิก ซึ่งค่อนข้างคล้ายกับ LAD เพียงแต่ว่าตัวกราฟิกจะถูกแทนที่ด้วยสัญลักษณ์ทางพีชคณิต เช่น AND, OR เป็นต้น และบางคำสั่งก็ใช้กราฟิกเดียวกันทั้ง LAD และ FBD เช่น IEC ไทม์เมอร์ เป็นต้น

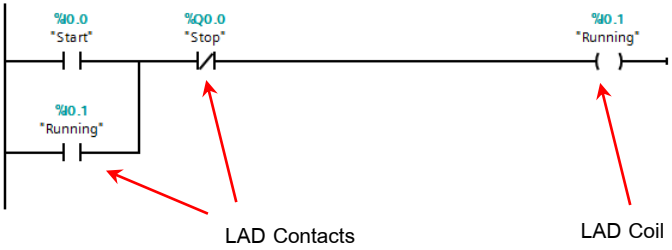


รูป 5.16

5.5 Basic instructions

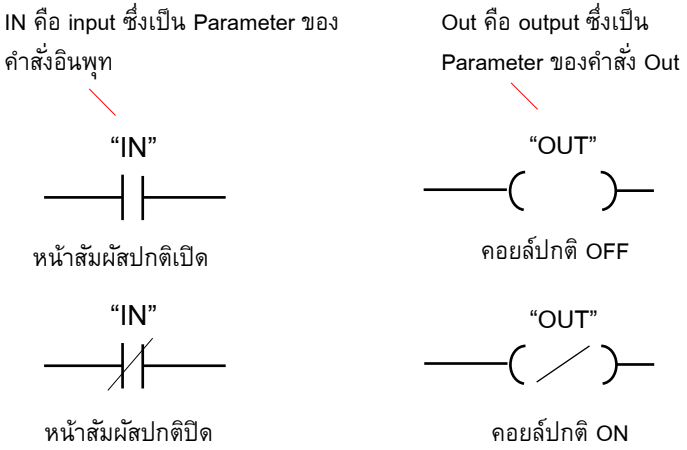
Bit logic contact และ Coil

น่าจะเป็นคำสั่งแรกๆที่ผู้หัดใช้งาน PLC ทุกคนต้องเขียนเป็นอันดับแรกๆ ซึ่งตัวคำสั่งนี้ ไม่ว่า PLC ยี่ห้อไหนๆก็แทบจะไม่มีความแตกต่างกันเลยเพราะค่อนข้างตรงไปตรงมาคือ NO contact, NC contact และ output coil นั่นเอง



รูป 5.26

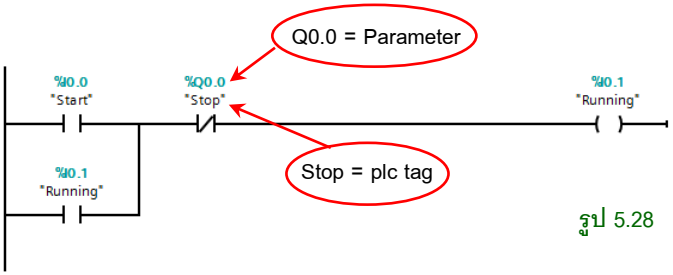
รูปที่ 5.26 หน้าสัมผัสในโปรแกรมแลดเดอร์ ทางซีเมนต์จะเรียกว่า LAD contact ส่วนคอยล์จะเรียกว่า LAD Coil สำหรับประเภทข้อมูลที่ใช้กับ LAD contact คือข้อมูลแบบ Bool ดังนั้นคำสั่งจะเรียกว่าคำสั่งแบบบูลีน (Boolean)



รูป 5.27

รูป 5.27 แสดงสัญลักษณ์ของหน้าสัมผัสและคอยล์ในโปรแกรมแลดเดอร์ การทำงานของคอยล์แบบปกติ OFF คือเมื่อมีไฟจ่ายเข้ามา คอยล์จะทำงาน ส่วนคอยล์ปกติ ON จะทำงานจนกว่าจะมีไฟจ่ายเข้ามา ทำให้ OFF

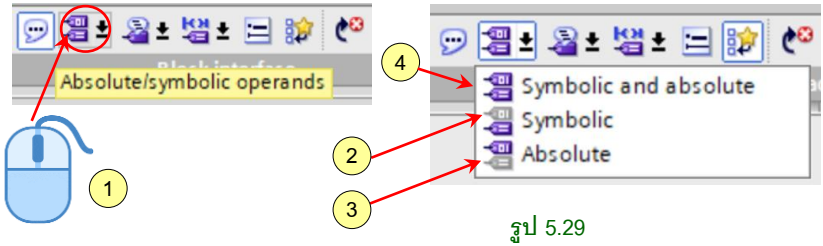
คำว่า Parameter เป็นคำที่ค่อนข้างกว้าง และในงานทางด้านโปรแกรม parameter ก็มีความหมายเดียวกับ variable (ตัวแปร) ได้เช่นกัน



รูป 5.28

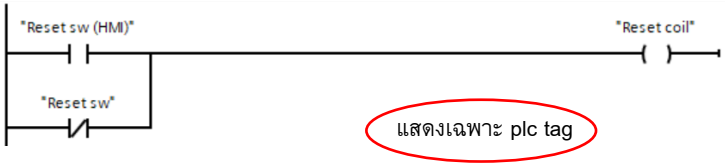
และจะสังเกตว่าที่กราฟิกหน้าสัมผัสและคอยล์ของวงจรจะมีทั้ง parameter (variable) และ plc tag กำกับอยู่ ซึ่ง STEP 7 จะเน้นการใช้ plc tag ในการเชื่อมระหว่างผู้ใช้งาน กับข้อมูลต่างๆสำหรับการเขียนคำสั่ง เนื่องจากง่ายในการค้นหาและสร้างความหมายของโปรแกรม เช่นรูปที่ 5.28 คำว่า stop คือ plc tag ส่วน Q0.0 คือ parameter

เราสามารถเลือกแสดงกราฟิกเฉพาะตัวใดตัวหนึ่งหรือทั้งสองได้ดังนี้

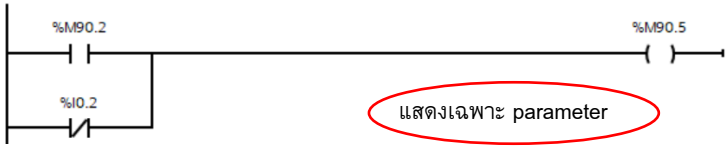


รูป 5.29

1. คลิกที่ไอคอน Absolute/symbolic operands
2. เมื่อคลิกเลือก symbolic โปรแกรมจะแสดงเฉพาะplc tag
3. เมื่อคลิกเลือก Absolute โปรแกรมจะแสดงเฉพาะ parameter
4. เมื่อคลิกเลือก symbolic และ Absolute โปรแกรมจะแสดงทั้ง plc tag และ parameter



รูป 5.30



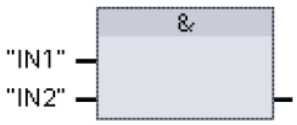
คำสั่งแบบ basic ในบทนี้จะไม่ได้อธิบายละเอียดมากนัก เนื่องจากเป็นคำสั่งที่ใช้งานเหมือนๆกับคำสั่งของ PLC ยี่ห้ออื่นๆเช่น Mitsubishi เพียงแต่กราฟิกและ parameter จะต่างกันเท่านั้น

ตาราง 5.1 ประสิทธิภาพของ CPU 1211C ในการดำเนินการคำสั่ง

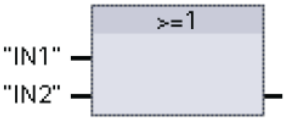
ประเภทคำสั่ง	ความเร็วดำเนินการ (Execution speed)
Boolean	0.08 µs/instruction
Move Word	1.7 µs/instruction
Real math	2.3 µs/instruction

คำสั่งแบบบูลีนจะใช้เวลาในการประมวลผลน้อยที่สุดคืออยู่ที่ 0.08us/คำสั่ง ตาราง 5.1 คือความเร็วประมวลผลของ CPU1211C

กรณีภาษาแบบ FBD หน้าสัมผัสจะไม่ได้ใช้แบบเดี่ยวๆเหมือน LAD ที่มี parameter เดียว สำหรับ FBD จะต้องใช้ 2 พารามิเตอร์ขึ้นไป รูปที่ 5.31 เป็นการใช้งาน AND และ NOT ในแบบ FBD



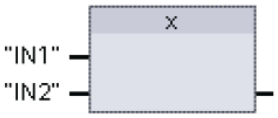
AND boxes



OR boxes

รูป 5.31

การทำงานของ AND boxes คือ IN1 และ IN2 จะต้อง ON ทั้งคู่ (เป็น 1 ทั้งคู่) ซึ่งจะได้ output คือ 1 ON (true) ส่วน OR box IN1หรือ IN2 ตัวใดก็ได้เป็น 1 จะทำให้ output = 1 ON



XOR boxes

Input		Output
A	B	A XOR B
0	0	0
0	1	1
1	0	1
1	1	0

รูป 5.32

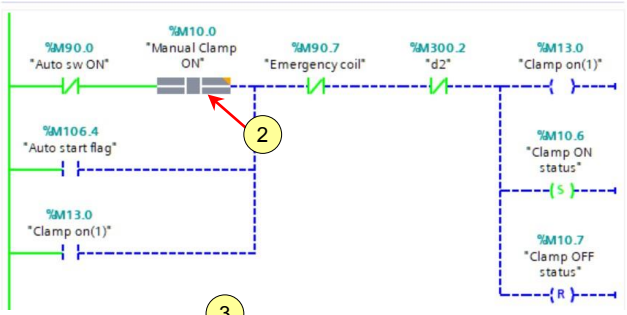
รูปที่ 5.32 การทำงานของ XOR boxes คือ IN1 และ IN2 ต่างกัน ซึ่งจะได้ output คือ 1 ON (true) กรณี IN1 เหมือนกับ IN2 เช่น 0,0 หรือ 1, 1 จะทำให้ output = OFF

การบังคับให้หน้าสัมผัส ON โดยใช้ software

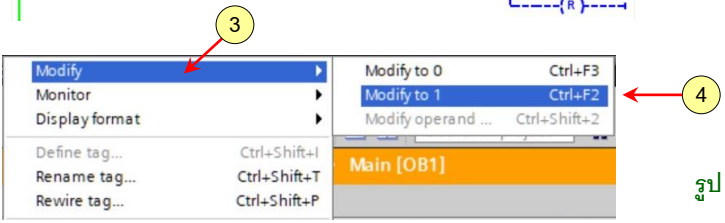
เป็นการทำให้หน้าสัมผัสในโปรแกรมทำงาน ใช้สำหรับทดสอบการทำงานของโปรแกรมบางส่วน โดยสามารถทำได้ในโหมด Online เท่านั้น ซึ่งทำได้ดังนี้

รูป 5.33

1. คลิกไอคอน Monitoring ⇨ 2. คลิกขวาหน้าที่สัมผัส(หรือชื่อ tag) ที่ต้องการ ON ในรูป 5.34 คือ %M10.0

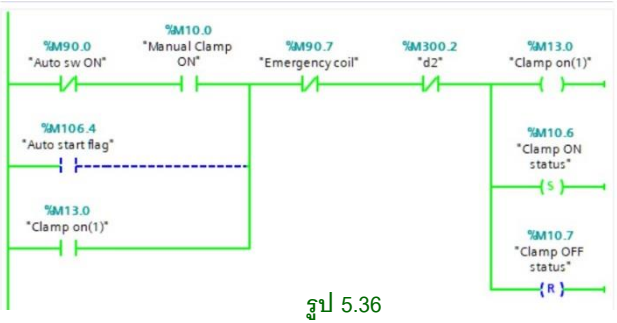


รูป 5.34



รูป 5.35

3. เลือก Modify ⇨ 4. คลิก Modify to 1 หน้าสัมผัส M10.0 จะ ON ดังรูปที่ 5.36 ซึ่งทำให้วงจรชุดนี้ทำงาน

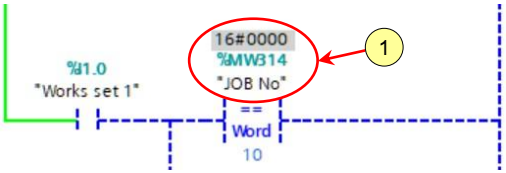


รูป 5.36

กรณีหน้าสัมผัส ON อยู่และต้องการ OFF ทำได้โดยเลือก Modify to 0 (หน้าสัมผัสอินพุท I สามารถบังคับ ON ในโปรแกรมได้เช่นกัน)

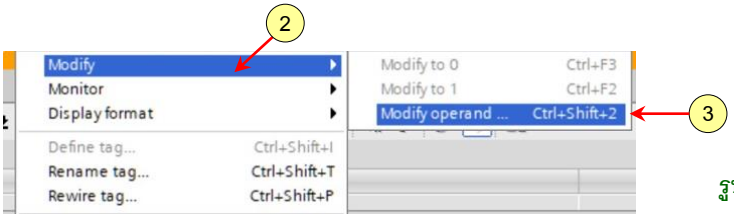
การเปลี่ยนค่าตัวเลขของตัวแปรโดยใช้ software

เป็นการตั้งค่าตัวเลขของตัวแปรต่างๆ ใช้สำหรับทดสอบการทำงานของโปรแกรม โดยสามารถทำได้ในโหมด Online เท่านั้น ซึ่งทำได้ดังนี้



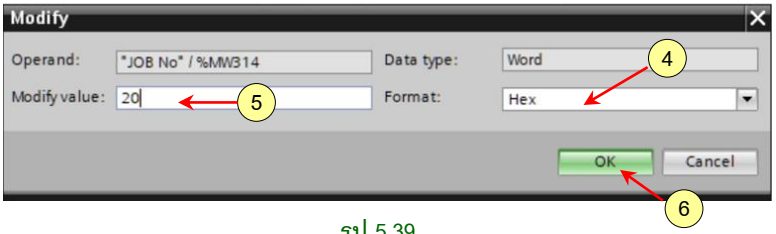
รูป 5.37

1. คลิกขวาที่ชื่อ Tag ที่ต้องการเปลี่ยนค่าในรูปคือ %MW314 ซึ่งมีค่าเท่ากับ 0 โดยเราต้องการเปลี่ยนค่าเป็น 20



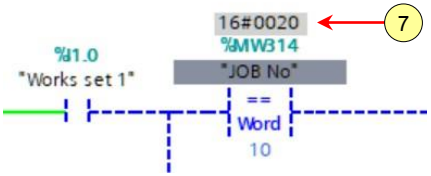
รูป 5.38

2. เลือก Modify ⇨ 3. คลิก Modify operand จะได้รูปที่ 5.39



รูป 5.39

4. เลือกชนิดข้อมูลซึ่งเป็นเลขฐานต่างๆ ⇨ 5. พิมพ์ 20 ⇨ 6. คลิก OK

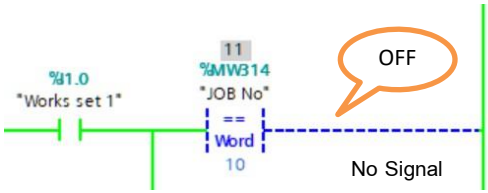


รูป 5.40

7. จากรูปที่ 5.40 จะได้ค่า MW314 เท่ากับ 20

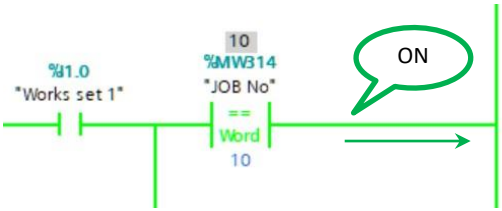
5.6 คำสั่งเปรียบเทียบ (Comparator instruction)

เป็นคำสั่งที่ใช้เปรียบเทียบค่าระหว่างสองตัวแปร เช่น มากกว่า น้อยกว่า เท่ากับ มากกว่าหรือเท่ากับ เป็นต้น เมื่อเงื่อนไขถูกต้อง จะทำให้คำสั่งทำงานคล้ายหน้าสัมผัส ON คือส่ง 1 ไปยังคำสั่งถัดไป



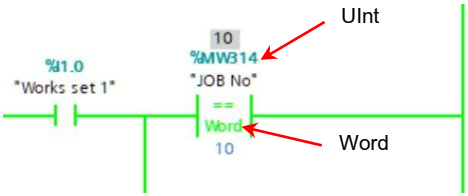
รูป 5.57

รูปที่ 5.57 เป็นคำสั่งที่เปรียบเทียบค่าแบบเท่ากับ(=) ระหว่างสองตัวแปรคือ MW314 และค่าคงที่ 10 ในรูปที่ 5.57 MW314 = 11 ซึ่งค่าไม่เท่ากัน ทำให้คำสั่งไม่จ่ายสัญาณ Output



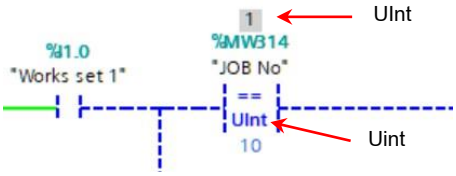
รูป 5.58

รูปที่ 5.59 กรณี MW314 เท่ากับ 10 จะทำให้สัญญาณวิ่งผ่านคำสั่งได้

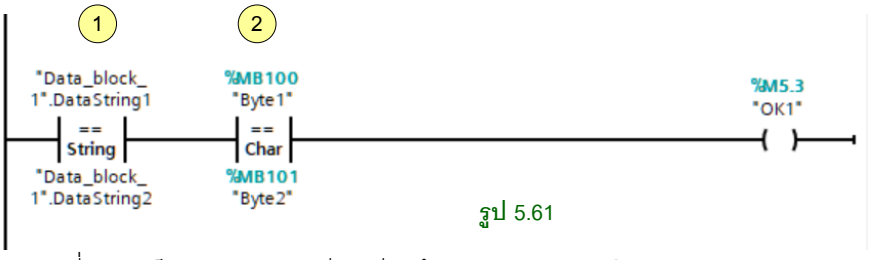


รูป 5.59

รูปที่ 5.59 ประเภทข้อมูลของตัวแปรในคำสั่ง เป็นการเปรียบเทียบข้อมูลแบบ Word แต่ชนิดของ MW314 คือ UInt ซึ่งกรณีนี้ไม่มีผลต่อการทำงานของคำสั่ง เนื่องจากคำสั่งจะมองว่า ถ้าตัวแปรทั้งสองมีค่าเท่ากัน คำสั่งก็จะทำงาน ON ให้สัญญาณผ่านได้ แต่เพื่อความเหมาะสมและไม่ให้สับสน ควรเลือกชนิดข้อมูลให้ตรงกัน ดังรูปที่ 5.60



รูป 5.60



รูป 5.61

วงจรรูปที่ 5.61 เป็นตัวอย่างการเปรียบเทียบข้อมูลแบบตัวอักษร โดย

1. เป็นการใช้คำสั่งเปรียบเทียบข้อมูลแบบ string ระหว่างตัวแปรแบบ data block สองตัวแปร กรณีนี้ตัวอักษรจะต้องเหมือนกัน M5.3 จึงจะ ON

2. คำสั่งเปรียบเทียบข้อมูลแบบ Char ระหว่าง MB100 และ MB101

การเปรียบเทียบข้อมูลแบบ sting และ char จะต้องเป็นการเปรียบเทียบแบบ = เนื่องจากตัวอักษรไม่มีค่ามากกว่าหรือน้อยกว่า

ตาราง 5.2 ประเภทข้อมูลที่ใช้ได้กับคำสั่งเปรียบเทียบ

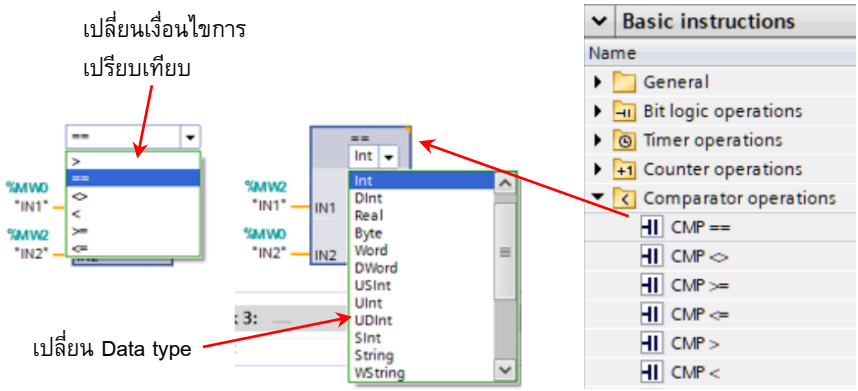
ตัวแปร	ประเภทข้อมูล	รายละเอียด
IN1, IN2	SInt, Int, DInt, USInt, UInt, UDInt, Real, LReal, String, Char, Time, DTL, Constant	ค่าสำหรับเปรียบเทียบ

ตาราง 5.3 สัญลักษณ์และความหมายของการเปรียบเทียบ

สัญลักษณ์	ความหมาย
=	IN1 เท่ากับ IN2
<>	IN1 ไม่เท่ากับ IN2
> =	IN1 มากกว่าหรือเท่ากับ IN2
< =	IN1 น้อยกว่าหรือเท่ากับ IN2
>	IN1 มากกว่า IN2
<	IN1 น้อยกว่า IN2

ตาราง 5.4 ลักษณะคำสั่งของภาษา LAD, FBD และ SCL

LAD	FBD	SCL
"IN1" == Byte "IN2"	== Byte "IN1" — IN1 "IN2" — IN2	**out := in1 = in2;** or **IF in1 = in2 THEN out := 1; ELSE out := 0; END_IF;**



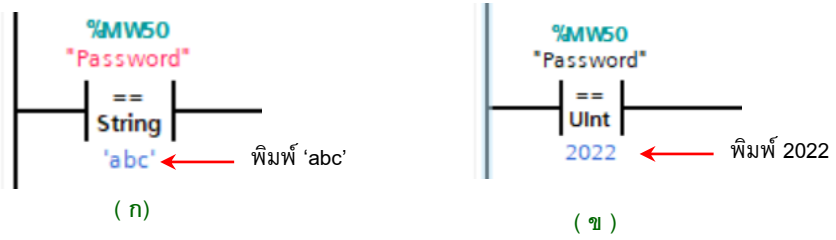
รูป 5.62

รูปที่ 5.62 คำสั่งเปรียบเทียบสามารถดึงได้จาก Basic instruction → Comparator operation เราสามารถดึงอันใดอันหนึ่งออกมาก่อน แล้วถ้าต้องการใช้คำสั่งหลายๆตัวก็สามารถ Copy ไปใช้งาน หากต้องการเปลี่ยนจากเท่ากับเป็นอย่างอื่นเช่น มากกว่าหรือน้อยกว่า ก็ให้ click ที่สัญลักษณ์เพื่อเปลี่ยนได้เลย

หรือเราสามารถเปลี่ยน data type ที่ต้องการได้จากตัวคำสั่งเลย เช่น หากต้องการเปรียบเทียบข้อมูลแบบ Byte ก็ double click ที่คำว่า Int ก็จะมี drop down menu ให้เลือกรูปแบบข้อมูลได้

การระบุข้อมูลโดยใช้ข้อมูลแบบฟิก

ข้อมูลแบบฟิก คือเรากำหนดข้อมูลแบบตายตัว รูปที่ 5.63 เป็นการระบุข้อมูลเป็นแบบฟิก กรณีต้องการระบุข้อมูลในคำสั่งเป็นตัวอักษรทำได้โดยการคลิกที่ช่องใส่ข้อมูลแล้วพิมพ์ตัวอักษรในเครื่องหมาย ‘ ’ ส่วนประเภทข้อมูลต้องเลือกแบบ string



รูป 5.63

คำสั่งรูปที่ 5.63ข กรณีข้อมูลแบบฟิกเป็นตัวเลข สามารถพิมพ์ตัวเลขที่ช่องข้อมูลได้เลย ส่วนประเภทข้อมูลก็ต้องเป็นแบบตัวเลข โดยเป็น type ที่เหมาะสม

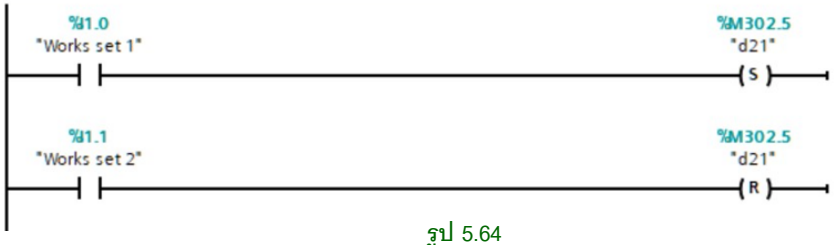
5.7 คำสั่ง Set และ Reset

ตาราง 5.5 ลักษณะคำสั่ง SET และ RESET ของภาษา LAD, FBD ส่วน SCL ไม่มีให้ใช้งาน

LAD	FBD	SCL
"OUT" —(S)—	"OUT" S "IN"	ไม่มีให้
"OUT" —(R)—	"OUT" R "IN"	ไม่มีให้

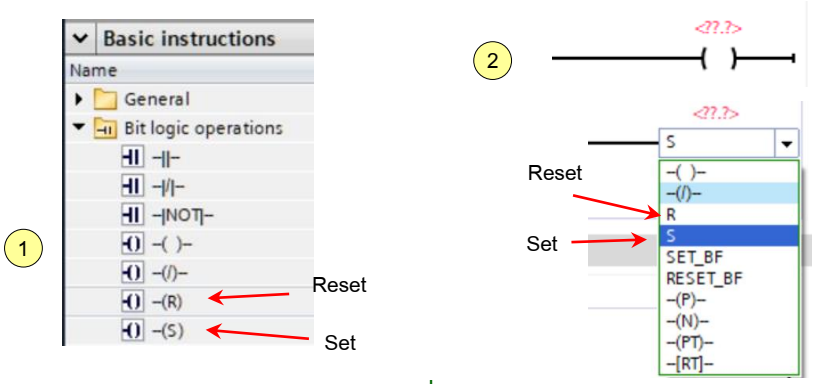
SET (S) เป็นคำสั่งที่ใช้ ON อุปกรณ์แบบบิตให้ทำงาน คำสั่ง SET ทำงานเพียง 1 Cycle (1 scan time) อุปกรณ์ที่คำสั่ง SET สั่งงาน จะทำงานตลอดเวลา

Reset (R) เป็นคำสั่งที่ใช้ OFF อุปกรณ์แบบบิตให้หยุดทำงาน คำสั่ง Reset ทำงานอุปกรณ์นั้นๆจะหยุดทำงาน



รูป 5.64

รูปที่ 5.64 เมื่อ I1.0 ON จะทำให้ M302.5 ทำงาน และเมื่อ I1.1 ON ทำให้ M302.5 หยุดทำงาน การเรียกใช้งานคำสั่ง Set และ reset ทำได้สองแบบคือ 1.ดึงคำสั่งจาก Basic instructions หรือ 2. สร้าง coil จากไอคอนลัด และเลือกประเภทเป็น S หรือ R



รูป 5.65

5.8 คำสั่ง Set Bit Field และ Reset Bit Field

ตาราง 5.6 ลักษณะคำสั่ง SET_BF และ RESET_BF ของภาษา LAD, FBD ส่วน SCL ไม่มีให้ใช้งาน

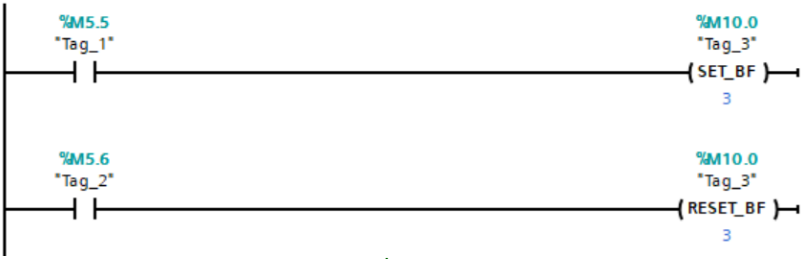
LAD	FBD	SCL
"OUT" —(SET_BF)— "n"	"OUT" SET_BF EN N	ไม่มีให้
"OUT" —(RESET_BF)— "n"	"OUT" RESET_BF EN N	ไม่มีให้

SET_BF เป็นคำสั่งที่ใช้ ON อุปกรณ์แบบบิตให้ทำงาน โดยสามารถให้อุปกรณ์แบบบิต ON ได้ทีละหลายๆตัว โดยอุปกรณ์จะ ON ตลอดเวลา เมื่อคำสั่ง SET ทำงานเพียง 1 Cycle

RESET_BF เป็นคำสั่งที่ใช้ OFF อุปกรณ์แบบบิตให้หยุดทำงาน สามารถรีเซ็ตอุปกรณ์ได้ทีละหลายๆบิต เมื่อคำสั่งทำงาน อุปกรณ์ที่กำหนดไว้จะหยุดทำงาน

ตาราง 5.7 ประเภทข้อมูลที่ใช้ได้กับคำสั่ง SET_BF และ RESET_BF

ตัวแปร	ประเภทข้อมูล
OUT	Bool
n	ค่าคงที่ (UInt)

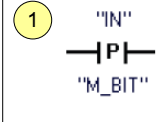
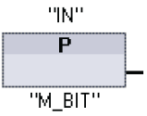
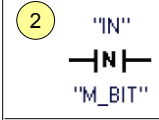
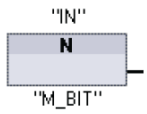
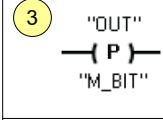
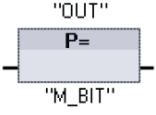
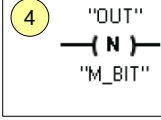
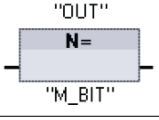


รูป 5.66

รูปที่ 5.66 ตัวอย่างการทำงานของคำสั่งคือ เมื่อ M5.5 ON จะทำให้ M10.0, M10.1 และ M10.2 ทำงาน (กำหนด n = 3, M10.0 คือบิตเริ่มต้น) และเมื่อ M5.6 ON จะทำให้ M10.0, M10.1 และ M10.2 หยุดทำงาน

5.9 คำสั่ง Positive และ Negative edge

ตาราง 5.8 ลักษณะคำสั่ง Positive และ Negative edge

LAD	FBD	SCL
1 		ไม่มีให้
2 		ไม่มีให้
3 		ไม่มีให้
4 		ไม่มีให้

1.Positive edge เป็นคำสั่งแบบหน้าสัมผัส ทำงานเป็นพัลส์แบบขาขึ้น ทำงานตามการ ON ของตัวแปร N โดยหน้าสัมผัสจะ ON เป็นจำนวน 1 scan time

2.Negative edge เป็นคำสั่งแบบหน้าสัมผัส ทำงานเป็นพัลส์แบบขาลง ทำงานตามการ OFF ของตัวแปร N โดยหน้าสัมผัสจะ ON เป็นจำนวน 1 scan time

3.Positive edge เป็นคำสั่งแบบคอยล์ เมื่อมีสัญญาณจ่ายมาให้ coil คอยล์จะทำงานเป็นพัลส์แบบขาขึ้นจำนวน 1 scan time

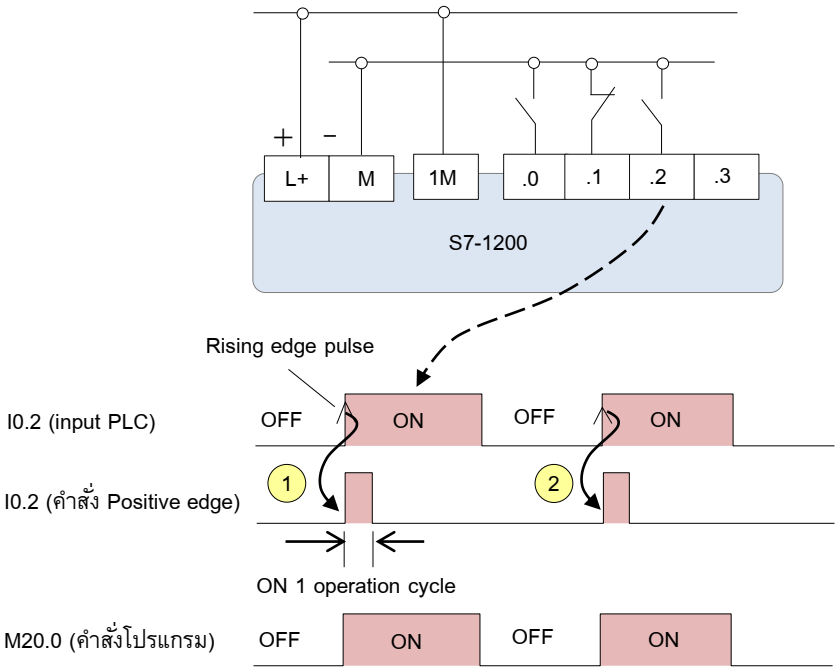
4.Negative edge เป็นคำสั่งแบบคอยล์ เมื่อสัญญาณที่จ่ายมาให้คอยล์มีการ OFF คอยล์จะทำงานเป็นพัลส์แบบขาลงจำนวน 1 scan time

ตาราง 5.9 ประเภทข้อมูลที่ใช้ได้กับคำสั่ง Positive และ Negative edge

ตัวแปร	ประเภทข้อมูล
M_BIT	Bool
IN	Bool
OUT	Bool

ตัวอย่างการทำงานของหน้าสัมผัสแบบ Positive edge

รูป 5.67



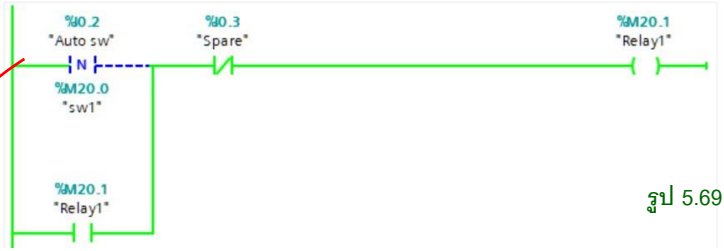
รูป 5.68

รูปที่ 5.68 เป็นการทำงานของคำสั่ง Positive edge แบบหน้าสัมผัส

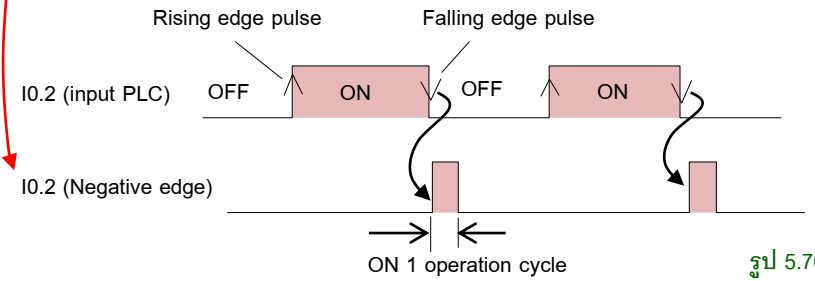
1. เมื่ออินพุท PLC I0.2 ON หน้าสัมผัส I0.2 ในโปรแกรมจะ ON ครั้งเดียวใน 1 รอบการประมวลผลของ PLC และ
2. เมื่อ I0.2 OFF และ ON อีกครั้ง หน้าสัมผัส I0.2 ในโปรแกรมจะ ON ใหม่

สำหรับ M_BIT คืออุปกรณ์แบบบิตที่ทำงานตามการ ON-OFF ของ parameter IN และ OUT ในรูป 5.68 M20.0 จะ ON-OFF ตามการทำงานของอินพุทจริง I0.2

ตัวอย่างการทำงานของหน้าสัมผัส Negative edge



รูป 5.69



รูป 5.70

รูปที่ 5.70 เป็นการทำงานของคำสั่ง Negative edge แบบหน้าสัมผัส เมื่ออินพุท PLC I0.2 ON หน้าสัมผัส I0.2 ในโปรแกรมจะยังไม่ทำงาน และเมื่ออินพุท I0.2 OFF หน้าสัมผัส I0.2 ในโปรแกรมจะ ON โดยทำงานครั้งเดียวใน 1 รอบการประมวลผลของ PLC และทำให้ M20.1 ทำงาน

ตัวอย่างการทำงานของคอยล์แบบ Positive edge

รูป 5.71

รูปที่ 5.71 เป็นการทำงานของคอยล์ Positive edge เมื่ออินพุทรีเลย์ I0.2 ON คอยล์ M20.3 จะ ON ครั้งเดียวใน 1 รอบการประมวลผลของ PLC และเมื่อ I0.2 OFF และ ON อีกครั้ง คอยล์ M20.3 จะ ON ใหม่

จบตัวอย่างบทที่ 5

สนใจซื้ออ่านฉบับเต็มได้ที่ ไลน์ official

@ecy6822d

หรือดูรายละเอียดเพิ่มเติม

www.plcsanook.com