# ความรู้พื้นฐานเกี่ยวกับ

### โปรแกรมเมเบิลคอนโทรลเลอร์

and Communit



#### โปรแกรมเมเบิลคอนโทรลเลอร์

ในอดีตการควบคุมเครื่องจักรกลและระบบควบคุมแบบอัตโนมัติมักใช้อุปกรณ์ควบคุมแบบ หน้าสัมผัสเชิงกลมีรีเลย์และแมกเนติกส์คอน แทคเตอร์ (Magnetic Contactor) เป็นตัวควบคุม หลักของการทำงาน ซึ่งมีข้อเสียอยู่หลายประการอาทิระบบควบคุมมีขนาดใหญ่ทำให้ สิ้นเปลืองเนื้อที่ การ ติดตั้งใช้เวลานาน การเปลี่ยนแปลงแก้ไขโปรแกรมทำได้ยาก ไม่เหมาะกับระบบที่มีการควบคุมแบบซับซ้อน



เมื่อการแข่งขันการผลิตภาคอุตสาหกรรมในตลาดโลกมีสูงขึ้น ความต้องการของเจ้าของโรงงาน อุตสาหกรรมที่จะสามารถผลิตได้ครั้ง ละจำนวนมาก ๆ มีต้นทุนการผลิต ต่ำ การจ้างแรงงานและงบประมาณ ด้านการบำรุงรักษาต่ำแต่ทำงานได้อย่างรวดเร็ว มีความความ ปลอดภัยในการใช้งานเครื่องจักร สามารถ ควบคุมผลกระทบต่อสิ่งแวดล้อม ควบคุมกระบวนการที่มีความยืดหยุ่นและทำงานในกระบวนการที่ ซับซ้อน จากความต้องการดังกล่าวระบบควบคุมอัตโนมัติสมัยใหม่จึงเข้ามามีความสำคัญต่อการพัฒนา อุตสาหกรรม โปรแกรมเมเบิลลอจิกคอนโทรลเลอร์ (Programmable Logic Controller : PLC) จึงเป็นอุปกรณ์ที่ ถูกคิดค้นขึ้นเพื่อควบคุมการ ทำงานของเครื่องจักรแบบอัตโนมัติและถูกพัฒนาอย่างต่อเนื่องให้ตอบสนอง ความต้องการของโรงงานอุตสาหกรรมสมัยใหม่ โปรแกรมเมเบิล ลอจิกคอนโทรลเลอร์จึงเป็นหัวใจหลักที่ใช้ ควบคุมการทำงานของเครื่องจักรหรือระบบต่าง ๆ อาทิอุตสาหกรรมเหล็กผลิตอาหารรถยนต์แขน กลใน งานอุตสาหกรรม ฯลฯ

โปรแกรมเมเบิลคอนโทรลเลอร์ (Programmable Controller : PC) จึงหมายถึงเครื่องควบคุม อัตโนมัติด้วยระบบคอมพิวเตอร์ สำหรับควบคุมเครื่องจักร หรืออุตสาหกรรมที่ต้องการระบบอัตโนมัติและ ความแม่นยำสูงสามารถโปรแกรมให้เป็นไปตามที่ต้องการได้โดยมีหน่ วยความจำในการเก็บโปรแกรมและ หน่วยอินพุต (Input) /เอาต์พุต (Output) สำหรับต่ออุปกรณ์เพื่อใช้งาน ข้อดีของการนำโปรแกรมเมเบิลคอนโทรลเลอร์มาใช้ในงานอุตสาหกรรม

- ใช้การเขียนโปรแกรมแทนการเดินสายไฟฟ้าต่อวงจรช่วยลดจำนวนสายไฟฟ้า
- 2. ตัวโปรแกรมสามารถเปลี่ยนแปลงและแก้ไขโปรแกรมได้ง่าย สะดวกและรวดเร็ว
- 3. มีฟังก์ชั่นทางคณิตศาสตร์และคำสั่งในการแปลงข้อมูลหลายแบบ
- 4. ตัวตั้งเวลา (Timer) และตัวนับ (Counter) จะเป็นคำสั่งซึ่งอยู่ในรูปของซอฟต์แวร์ (Software) ทำให้กำหนดค่าต่างๆได้ง่าย และ สามารถเปลี่ยนแปลงค่าได้ตลอดเวลา
- การเพิ่มและขยายระบบสามารถทำได้ง่ายและสะดวก
- 6. ใช้ควบคุมกระบวนการผลิตได้ทั้งแบบแอนะลอก (Analog) และแบบดิจิทัล (Digital)
- 7. มีความทนทานต่อสภาพแวดล้อมของโรงงานอุตสาหกรรม
- 8. การติดตั้ง การดูแลรักษา การซ่อมบำรุงทำได้ง่ายใช้พื้นที่ติดตั้งอุปกรณ์น้อย
- 9. ขณะ PLC ทำงานสามารถตรวจสอบสภาวะการทำงานและค่าสถานะต่างๆของอุปกรณ์ได้

ประเภทของโปรแกรมเมเบิลคอนโทรลเลอร์

การนำ PLC ไปใช้งานจะพิจารณาจากจำนวนอินพุตและเอาต์พุตของงานที่นำไปควบคุมเป็นหลัก ความหลากหลายของรุ่น PLC ที่บริษัทผู้ผลิตจำหน่ายออกมาจึงผลิตเพื่อตอบสนองความต้องการใช้งานที่ ต่างกัน ทั้งนี้สามารถแบ่งประเภทของโปรแกรมเมเบิลคอนโทรล เลอร์ได้ดังนี้

การแบ่งตามขนาดของหน่วยความจำ (Memory Unit) และของจำนวน อินพุตและเอาต์พุต (Input/Output channels) 1. โปรแกรมเมเบิลคอนโทรลเลอร์ขนาดเล็ก (Small or Micro size) มีจำนวนอินพุต/ เอาต์พุตไม่เกิน 128 จุดหน่วยความจำประมาณ 1. ไป แกรมเมเบิลคอนโทรลเลอร์ขนาดเล็ก (Small or Micro size) มีจำนวนอินพุต/ เอาต์พุตไม่เกิน 128 จุดหน่วยความจำประมาณ

4 Kbyte (2,000 Statements)



2. โปรแกรมเมเบิลคอนโทรลเลอร์ขนาดกลาง (Medium size) มีจำนวนอินพุต/เอาต์พุต ไม่เกิน 1,024 จุด หน่วยความจำประมาณ 16
 Kbyte (8,000 Statements)



3. โปรแกรมเมเบิลคอนโทรลเลอร์ขนาดใหญ่ (Large size) มีจำนวนอินพุต/เอาต์พุตไม่เกิน 2,048 จุด หน่วยความจำประมาณ 64 Kbyte (32,000 Statements)



4. โปรแกรมเมเบิลคอนโทรลเลอร์ขนาดใหญ่มาก (Very large size) มีจำนวนอินพุต/ เอาต์พุตประมาณ 8,192 จุด หน่วยความจำ ประมาณ 256 Kbyte (128,000Statements) เป็น PLC แบบ แร็ก (Rack) สามารถต่อเพิ่มอินพุต/เอาต์พุตตามจำนวนที่ต้องการการขยาย ระบบ

การแบ่งตามลักษณะโครงสร้างภายนอก

 โปรแกรมเมเบิลคอนโทรลเลอร์แบบคอมแพ็กต์ (Compact) หรือบล็อก (Block) มีหน่วย ประมวลผล หน่วยความจำ หน่วยอินพุต หน่วยเอาต์พุตและหน่วยแหล่งจ่ายพลังงานอยู่บนโครงสร้าง เดียวกัน เหมาะกับงานใช้จำนวนอินพุตและเอาต์พุตที่แน่นอนและไม่ต้องการ ขยายระบบ





2. โปรแกรมเมเบิลคอนโทรลเลอร์แบบมอดูล (Module) หรือแร็ก (Rack) ส่วนประกอบแต่ ละส่วนเป็นมอดูลแยกออกจากกัน เช่น มอดูลอินพุตและเอาต์พุต (Module input-output) มอดูลแอนะลอก (Module analog) โดยติดตั้งมอดูลอยู่บนแร็ก หากมีมอดูลเสียหาย สามารถถอดไปซ่อมและระบบยังคง ทำงานต่อไปได้จำนวนอินพุตและเอาต์พุตและมอดูลฮาร์ดแวร์ (Module hardware) มีความยืดหยุ่นนิ ยมใช้ โรงงานอุตสาหกรรมขนาดกลางและใหญ่



#### โครงสร้างของโปรแกรมเมเบิลคอนโทรลเลอร์

โครงสร้างของโปรแกรมเมเบิลคอนโทรลเลอร์มีส่วนประกอบหลักที่สำคัญ 4 หน่วยคือ

- 1. หน่วยประมวลผล (Central Processing Unit,CPU)
- 2. หน่วยความจำ (Memory Unit)
- 3. หน่วยแหล่งจ่ายพลังงาน (Power Supply Unit)
- 4. หน่วยอินพุตและเอาต์พุต (Inputs/Outputs Unit)



หลักการทำงานของหน่วยต่างๆ ภายในโปรแกรมเมเบิลคอนโทรลเลอร์

#### หน่วยประมวลผล (Central Processing Unit, CPU)

หน่วยประมวลผลเปรียบเสมือนส่วนมันสมอง ทำหน้าที่ควบคุมการทำงานและจัดการระบบ ทั้งหมดโดยจะรับสัญญาณและปริมาณต่าง ๆ จากหน่วยอินพุตเพื่อใช้เป็นข้อมูลในการประมวลผลจากนั้น จะประมวลผลข้อมูลแล้วส่งสัญญาณและปริมาณต่างๆไปยังหน่วยเอาต์พุตเพื่อ ควบคุมอุปกรณ์ทำงานอ

หน่วยประมวลผลจะทำงานวนเป็นรอบ ในหนึ่งรอบการทำงานเรียกว่า 1 สแกนไทม์ (Scantime) ซึ่งความเร็วของหน่วยประมวลผล ขึ้นอยู่กับขนาดของหน่วยความจำ จำนวนข้อมูลที่ป้อนและ ความเร็วของซีพียู (CPU)



หน่วยความจำ(Memory Unit)

หน่วยความจำเป็นส่วนที่ทำหน้าที่เก็บโปรแกรมและข้อมูลที่ใช้ในการทำงาน ขนาดของ หน่วยความจำเป็นสิ่งที่กำหนดความสามารถ ของ PLC โดยแบ่งหน่วยความจำของ PLC แบ่งออกได้ 2 แบบ ดังนี้คือ

 หน่วยความจำชั่วคราว (Random Access Memory : RAM) เป็นหน่วยความจำที่ใช้เก็บ โปรแกรมที่ผู้ใช้สร้างขึ้นแล้วป้อนให้กับ PLC หน่วยความจำแบบนี้ต้องมีไฟเลี้ยงเพื่อป้องกันข้อมูลไม่ให้สูญ หาย แรม (RAM) จะใช้เก็บโปรแกรมและข้อมูลที่ทำงานจากการสั่ง RUN/STOP ของ PLC ดังนั้นใน PLC จะมีแบตเตอรี่สำรอง (Battery Backup) ไว้สำหรับกรณีที่ไฟเลี้ยงหลักไม่จ่ายให้กับ PLC เพื่อป้องกัน ข้อมูล ไม่ให้สูญหาย

 หน่วยความจำถาวร (Read Only Memory : ROM) เป็นหน่วยความจำที่ใช้สำหรับ โปรแกรมและข้อมูลในรอม (ROM) ไม่จำ เป็นต้องใช้ไฟเลี้ยง เป็นชุดสำรองโปรแกรมและข้อมูลเพื่อป้องกัน กรณีที่โปรแกรมและข้อมูลในรอมสูญหายผู้ใช้งานสามารถถ่ายโอนโปรแกรม และข้อมูลเข้าในรอมได้ใหม่ รอมแบ่งออกได้เป็น 2 ชนิด ดังนี้  EPROM (Erasable Programmable Read Only Memory) หน่วยความจำชนิดนี้ จะต้องใช้เครื่องมือพิเศษในการเขียน โปรแกรม การลบโปรแกรมทำได้โดยใช้แสงอัลตราไวโอเลต มีข้อดี ตรงที่โปรแกรมจะไม่สูญหายแม้ไฟดับ จึงเหมาะกับการใช้งานที่ไม่ต้องการ เปลี่ยนโปรแกรม

 EEPROM (Electrically Erasable Programmable Read Only Memory) เ ป็น หน่วยความจำที่ถูกพัฒนาจากข้อเสียของ แรม (RAM) และ EPROM ให้มีขีดความสามารถที่ดีขึ้น สามารถ เขียนและอ่านข้อมูลโดยใช้ไฟฟ้า ไม่ต้องใช้เครื่องมือพิเศษในการเขียน โปรแกรมและลบโปรแกรม ไม่ต้องมี แบตเตอรี่สำรองไฟเมื่อไฟดับ เป็นหน่วยความจำที่รวมคุณสมบัติที่ดีของหน่วยความจำแรม (RAM) และ EPROM เอาไว้ด้วยกัน ซึ่งในปัจจุบันมีการพัฒนาหน่วยความจำ EEPROM ทำให้สามารถเก็บข้อมูลได้ เหมือนฮาร์ดดิสก์คือ สามารถเขียนและ ลบข้อมูลได้ตามความต้องการ หน่วยความจำชนิดนี้มีขนาดเล็ก น้ำหนักเบา พกพาสะดวก เช่น คอมแพ็กต์แฟลช (Compact Flash) ซึ่ง PLC รุ่นใหม่จะมีช่องสำหรับเสียบ หน่วยความจำประเภทเอสดีการ์ด (SD Card) แฟลชเม็มโมรี (Flash Memory) สำหรับบันทึกข้อมูลไว้ให้ สามารถถอดและเคลื่อนย้ายได้โดยได้รับความนิยมเพิ่มขึ้นสูงมากเพราะสามารถเก็บข้อมูลขนาดใหญ่ได้ หน่วยแหล่งจ่ายพลังงาน (Power Supply Unit) หน่วยแหล่งจ่ายพลังงานทำหน้าที่จ่ายพลังงานและรักษาระดับแรงดันให้กับอุปกรณ์และ วงจรภายในโปรแกรมเมเบิลคอนโทรลเลอร์

และมีจุดจ่ายพลังงานสำหรับการต่อวงจรภาคอินพุตและ เอาต์พุต

หน่วยอินพุตและเอาต์พุต (Inputs/Outputs Unit)

หน่วยอินพุต (Input Unit) ทำหน้าที่รับสัญญาณจากอุปกรณ์ทางด้านอินพุตทั้งที่เป็นสวิตช์ (Switch) หรือเซนเซอร์ (Sensor) แบบ ต่าง ๆ โดยอุปกรณ์ทางด้านอินพุตจะถูกแปลงสัญญาณเพื่อส่งเข้าไป ที่ CPU หน่วยอินพุตของโปรแกรมเมเบิลคอนโทรลเลอร์แบ่งเป็น 2 ประเภทใหญ่คือหน่วยอินพุตแบบดิจิทัล และแบบแอนะลอก

หน่วยอินพุตแบบดิจิทัล (Digital Input) หมายถึงอินพุตที่สามารถรับสัญญาณที่มีลักษณะ การทำงานเพียง 2 สภาวะ คือ สภาวะ ทำงาน (On) และสภาวะหยุดทำงาน (Off) จากอุปกรณ์อินพุตแบบ ต่าง ๆ เช่น สวิตช์แบบต่าง ๆ เซนเซอร์แสง (Photoelectric Sensor) เซนเซอร์ตรวจจับระยะสั้น (Proximity Sensor) เป็นต้น มีโครงสร้างลักษณะการต่อสายของวงจรอินพุตของซีพียู (CPU)



ภาษาที่ใช้ในการเขียนโปรแกรมของโปรแกรมเมเบิลคอนโทรลเลอร์

โปรแกรมเมเบิลคอนโทรลเลอร์จะมีภาษาที่ใช้ในการเขียนโปรแกรมที่เป็นมาตรฐานทั่วไป ที่สามารถ ศึกษาและเรียนรู้ได้ง่าย ปัจจุบันจะ มีภาษาที่ใช้ในการเขียนโปรแกรมเมเบิลคอนโทรลเลอร์ดังนี้คือ

ภาษาแลดเดอร์ไดอะแกรม (Ladder Diagram : LD)

ภาษา Ladder Diagram หรือ Ladder Logic Editor : LAD เป็นแบบที่นิยมใช้ในงาน ควบคุมด้วย PLC การเขียนมีลักษณะคล้ายวง จรควบคุม (Control Circuit) ในงานควบคุมมอเตอร์คือ การ นำหน้าสัมผัสหรืออุปกรณ์ต่าง ๆ มาต่อเป็นวงจรผู้ใช้เข้าใจการทำงานได้ง่าย ทางเดินของ Ladder แต่ละ แถวเรียกว่า เน็ตเวิร์ก (Network) ภาษา Function Block Diagram มีลักษณะคล้ายชุดคำสั่งของวงจรลอจิกเกต ไม่มี สัญลักษณ์ของคอนแทคและคอยล์ (Coil) รูปแบบ บล็อกของคำสั่ง FBD จะมีความใกล้เคียงกัน แต่ชุดคำสั่ง ในแต่ละบล็อกแตกต่างกัน การเขียนโปรแกรม FBD จะเป็นลักษณะการนำบล็อกมา เรียงต่อกัน

Network 1	Network Title		
	10.0 OR Q0.0 II.1 O	AND Q0.0	
Network 2	Q0.1		

ภาษาสเตทเมนต์ลิสต์ (Statement List : STL)

ภาษา Statement List เป็นภาษาดิจิตอลที่สามารถรับรู้โดยอาศัย CODE คำสั่ง เช่น LD, AN, O, = เป็นต้น ผู้ใช้ต้องมีความชำนาญในการ เขียนโปรแกรมเพราะเป็นภาษาเฉพาะของเครื่อง ผู้ใช้ต้อง เข้าใจสัญลักษณ์ของภาษาที่ใช้ในแต่ละคำสั่งจึงจะสามารถเขียนโปรแกรมได้

Network LD O AN =	1 10 00 10 00	Network Title . 0 . 1 . 0	
Network LD =	.2 IO QO	. 2 . 1	

ภาษาที่ใช้ในการเขียนโปรแกรมของโปรแกรมเมเบิลคอนโทรลเลอร์

อุปกรณ์ต่อร่วมกับโปรแกรมเมเบิลคอนโทรลเลอร์ทำหน้าที่ช่วยเพิ่มความสะดวกและความรวดเร็วใน การใช้งาน ใช้ในการตั้งค่าเพื่อป้ อนข้อมูลการควบคุมเครื่องจักรมีซอฟแวร์ที่สามารถเขียนโปรแกรมผ่าน คอมพิวเตอร์รองรับเทคโนโลยีต่างๆ มีฟังก์ชั่นการใช้ที่หลากหลาย อุปกรณ์ต่อร่วมกับมีหลายชนิด เช่น

Human Machine Interface (HMI)

Human Machine Interface (HMI) เป็นอุปกรณ์ที่ใช้ในการติดต่อสื่อสารระหว่างผู้ใช้กับ เครื่องจักร ใช้ในการควบคุมและแสดงผล แสดงสถานะการทำงาน สภาวะของการควบคุม ฯลฯ มีฟังก์ชั่น การใช้งานเพื่อความสะดวกให้กับผู้ใช้งานตามเทคโนโลยีของบริษัทที่นำมา ออกแบบและพัฒนาขึ้น อาทิ Text Display Touch Screen เป็นต้น



Programming Console หรือ Hand Held Programmer

Programming Console หรือ Hand Held Programmer เป็นอุปกรณ์ที่ใช้สำหรับการเขียน โปรแกรมแบบมือถือมีขนาดเล็ก สะดวก ในการพกพาและใช้งานซึ่งในอดีตนิยมใช้ในการเขียนโปรแกรม แต่ ปัจจุบันนิยมใช้คอมพิวเตอร์ในการเขียนโปรแกรมมากกวาเน' องจาก คอมพ<sup>ี</sup>่ วเตอริ ์มีราคาถูกลงและซอฟต์แวร์ ที่ใช้เขียนโปรแกรมได้พัฒนาให้ใช้งานได้ง่ายขึ้น



คอมพิวเตอร์และซอฟแวร์ (Computer and Software)

คอมพิวเตอร์ใช้สำหรับเขียนโปรแกรมแก้ไขโปรแกรมและแสดงสภาวะการควบคุมโดยใช้ ร่วมกับซอฟต์แวร์ของโปรแกรมเมเบิล คอนโทรลเลอร์ยี่ห้อนั้น ๆ เช่น PLC SIEMENS ใช้โปรแกรม STEP 7 Micro/WIN หรือ PLC ORMRON ใช้โปรแกรม CX-Programmer ทั้งนี้ แต่ละบริษัทจะพัฒนาซอฟต์แวร์ให้ สามารถรองรับฟังก์ชั่นที่เพิ่มขึ้นตามการพัฒนาของโปรแกรมเมเบิลคอนโทรลเลอร์





## ความรู้พื้นฐานทางดิจิทัล



ประเภทของข้อมูลพื้นฐานภายในโปรแกรมเมเบิลคอนโทรลเลอร์

ข้อมูลภายในของ PLC จะมีคำจำกัดความในการใช้งานที่เรียกกันคือ บิต (Bit) ไบต์ (Byte) เวิร์ด (Word) มีหลักการเรียกและ เปรียบเทียบข้อมูลแบบต่างๆ ดังนี้คือ





ในการใช้งาน PLC มีการใช้งานระบบเลขฐาน (Number System) อยู่ด้วยกัน 3 ระบบ คือ ระบบ เลขฐานสอง (Binary System) เลขฐานสิบ (Decimal System) และเลขฐานสิบหก (Hexadecimal System) ระบบเลขฐานสอง (Binary System : Bin) ใช้จำนวนนับตัวเลข 0 และ 1 ระบบเลขฐานสิบ (Decimal System : Dec) ใช้จำนวนนับตัวเลข 0 – 9 หรือที่เรียก อีกชื่อหนึ่งว่า Binary Code Decimal : BCD CODE

#### ระบบเลขฐานสิบหก (Hexadecimal System : Hex)

ใช้จำนวนนับตัวเลข 0 – 15 และใช้อักษรภาษาอังกฤษ A – F แทนตัวเลขสองหลัก 10 – 15 ทั้งนี้สามารถกำหนดความสัมพันธ์ของตัว เลขฐานสองเลขฐานสิบและเลขฐานสิบหกได้

เลขฐานสิบ	เลขฐานสอง	เลขฐานสิบหก
(Decimal System)	(Binary System)	(Hexadecimal System)
0	0000	0
1	0001	1
2	0010	2
3	0011	3
4	0100	4
5	0101	5
6	0110	6
7	0111	7
8	1000	8
9	1001	9
10	1010	A
11	1011	В
12	1100	С
13	1101	D
14	1110	E
15	1111	F



การแปลงเลขฐานสองเป็นเลขฐานสิบ

ในการแปลงเล<sup>ื</sup>่อฐานสองเป็นเล<sup>ื</sup>่อฐานสิบนั้น สามารถพิจารณาจากน้ำหนัก (Weight) ของเลข ฐานในแต่ละบิต

	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00	
	2 <sup>15</sup>	2 <sup>14</sup>	2 <sup>13</sup>	2 <sup>12</sup>	2 <sup>11</sup>	2 <sup>10</sup>	2 <sup>9</sup>	2 <sup>8</sup>	27	2 <sup>6</sup>	2 <sup>5</sup>	2 <sup>4</sup>	2 <sup>3</sup>	2 <sup>2</sup>	2 <sup>1</sup>	2 <sup>0</sup>	ک <sup>Bit</sup>
	32768	16384	8192	4096	2048	1024	512	256	128	64	32	16	8	4	2	1	Weight
																	Data
/													<b>I</b>			•	1

al and Communit



ตัวอย่างที่ 2.1

จงแปลงข้อมูลเลขฐานสองขนาด 16 บิต ซึ่งมีค่าข้อมูลคือ 0000 0011 0110 110"1" \_"2" ให้เป็นเลขฐานสิบ จะมีค่าเท่ากับเท่าใด วิธีทำแบบที่ 1

- 1. ให้นำข้อมูลเลขฐานสองขนาด 16 บิตค่าข้อมูลคือ 0000 0011 0110 110"1" \_"2" ใส่ลงไปใน ช่อง Data โดยเรียงตามบิต
- 2. พิจารณาน้ำหนักข้อมูลในแต่ละบิตโดยพิจารณาเฉพาะข้อมูลในบิตที่เป็น 1 เท่านั้น แล้ว นำค่าน้ำหนักของข้อมูลที่บิตเป็น 1 มาบวก กันจะได้ผลลัพธ์เป็นเลขฐานสิบ



วิธีทำแบบที่ 2

เป็นการนำค่าข้อมูลในแต่ละบิตคูณกับน้ำหนักในแต่ละบิต นำค่าที่ได้มา บวกกัน 0000 0011 0110 110"1" \_"2" = (0×"2" ^"15" )+(0×"2" ^14)+(0×"2" ^13)+(0×"2" ^12)+(0×"2" ^11)+(0×"2" ^"10" )+(1×"2" ^9)+(1×"2" ^8)

+(0×"2" ^7)+(1×"2" ^"6" )+(1×"2" ^5)+(0×"2" ^4)+(1× "2" ^3)+(1× "2" ^"2" )+(0× "2" ^"2" )+(1× "2" ^"0" )

= 512 + 256 + 64 + 32 + 8 + 4 + 1

= 87"7" \_10

ตอบ เลขฐานสองค่าข้อมูลคือ 0000 0011 0110 110"1" \_"2" แปลงเป็นเลขฐานสิบ มีค่าเท่ากับ 87"7" \_"10" การแปลงเลขฐานสองเป็นเลขฐานสิบ

ในการแปลงเลขฐานสิบเป็นเลขฐานสองนั้น สามารถทำได้โดยวิธีการหารสั้น โดยการนำค่า เลขฐานสิบที่ต้องการแปลงหารด้วยสองแล้ วเขียนผลหารและเศษที่เหลือจากการหารในแต่ละครั้งไว้ทำต่อ จนกระทั่งผลของการหารเป็น 0 สุดท้าย เขียนเศษที่ได้จากการหาร โดยเขียน จากด้านล่างขึ้นด้านบน

```
ตัวอย่างที่ 2.2
```

จงแปลงข้อมูลเลขฐานสิบ 5"5" \_"10" ให้เป็นเลขฐานสอง วิธีทำ

ให้นำข้อมูลเลขฐานสิบ มาตั้งหารด้วย 2 ตลอดจนกว่าจะเหลือค่าน้อยกว่าตัวหาร

 2. นำเศษที่ได้จากการหารเป็นคำตอบโดย เขียนข้อมูลเริ่มจากด้านล่างขึ้นด้านบน จะได้ 11011"1" \_"2" ตอบ 5"5" \_"10" แปลงให้เป็นเลขฐานสองมีค่า = 11011"1" \_"2"

55÷2       27       1         27÷2       13       1         13÷2       6       1	1 1 9 1/1 1 9	ผลหาร	เศษ
27÷2     13     1       13÷2     6     1	55 <b>÷</b> 2	27	1
13÷2 6 (1)	27 <b>÷</b> 2	13	1
	13 <b>÷</b> 2	6	1
6÷2 3 0	6 <b>÷</b> 2	3	0
3÷2 1 1	3 <b>÷</b> 2	1	1

ตัวอย่างที่ 2.3

```
จงแปลงข้อมูลเลขฐานสอง 101101100111000"1" _"2" ให้เป็นเลขฐานสิบหก
วิธีทำ
```

```
ให้นำข้อมูลเลขฐานสองขนาด 16 บิตมาแบ่งออกเป็นกลุ่ม จะได้ 4 กลุ่ม กลุ่มละ 4 บิต ข้อมูลเลขฐานสอง 101101100111000"1"
"2" = 1011 0110 0111 0001
```

พิจารณาค่าฎ้าหนัก (Weight) ของเลขฐานสองแต่ละกลุ่มแล้วบวกค่าฎ้าหนักภายใน กลุ่ม จะได้ผลลัพธ์เป็นเลขฐานสิบหกในแต่ละ หลัก (Digit)



ตอบ ข้อมูลเลขฐานสอง 1011 0110 0111 000"1" \_"2" เป็นเลขฐานสิบหกเท่ากับ B67"1" \_16

#### การแปลงเลขฐานสิบหกเป็นเลขฐานสอง

สี่หลักด้วยการเปรียบเทียบค่าน้ำหนัก

จะใช้หลักการคล้ายกับการแปลงเลขฐานสองเป็นเลขฐานสิบหก โดยให้พิจารณากลับจากค่า ของเลขฐานสิบหกในแต่ละหลักแปลงออก ออกเป็นเลขฐานสอง(โดยเลขฐานสิบหก 1 หลัก (Digit) จะมีค่า เท่ากับเลขฐานสองขนาด 4 บิต) ตัวอย่างที่ 2.4 จงแปลงข้อมูลเลขฐานสิบหก 5FB"4" \_"16" ให้เป็นเลขฐานสอง

1. ให้นำค่าของเลขฐานสิบหกที่ได้จากโจทย์ใส่ลงไปในแต่ละหลัก (Digit)

									-								
		Digit	3 = <mark>5</mark>			Digit	t 2 = F			Digit	1 = B			Digit	0 = 4		
	8	4	2	1	8	4	2	1	8	4	2	1	8	4	2	1	Weight
																	Data
1																	

2. พิจารณาค่าของเลขฐานสิบหกในแต่ละหลักว่ามีค่าเท่ากับเท่าไร จากนั้นพิจารณากลับ จากเลขฐานสิบหกในแต่ละหลัก ให้เป็นเลขฐานสอง

Digit 3 = 5Digit 2 = F(16)Digit 1= B (11) Digit 0 = 48 Weight 2 0 0 0 0 4+1 = 5F = 16B = 111111 1011

ตอบ เลขฐานสิบหก 5FB"4" \_"16" แปลงเป็นเลขฐานสองได้เท่ากับ 0101 1111 1011 010"0" \_"2"

#### การแปลงเลขฐานสิบหกเป็นเลขฐานสิบ

ในการแปลงเลขฐานสิบหกเป็นเลขฐานสิบ ให้ใช้หลักการกระจายค่าออกมาเป็นผลรวมของ เลขฐานสิบ

ตัวอย่างที่ 2.5 จงแปลงข้อมูลเลขฐานสิบหก 4CA"9" <u>"16" ให้เป็นเลขฐานสิบ</u>

วิธีทำ 4CA"9" \_"16" = (4×1"6" ^"3" )+(C×1"6" ^2)+(A×1"6" ^1)+(9×1"6" ^0)

= (4×1"6" ^"3" )+(12×1"6" ^2)+(10×1"6" ^1)+(9×1"6" ^0)

= 16384 + 3072 + 160 + 9

ตอบ 4CA"9" \_"16" = 1962"5" \_"10"



PLC ประกอบขึ้นจากอุปกรณ์และวงจรทางดิจิทัล (Digital) มีส่วนประกอบพื้นฐานคือลอจิกเกต (Logic Gate) และทำงานด้วย หลักการระบบเลขฐานสองมีเลข 0 กับเลข 1

หลักการพื้นฐานของ AND Gate

ถ้ากำหนดให้ A และ B แทนตัวแปรอินพุต (Input) เมื่อตัวแปร A มากระทำการ AND กับตัว แปร B ได้ผลลัพธ์ออกเป็น Y สามารถเขียนสมการลอจิกทางด้านเอาต์พุต (Output) ได้ดังนี้

#### $= A \cdot E$

จากสมการลอจิก เครื่องหมาย (•) คือ การคูณแบบ AND สามารถ<mark>เขียนตารางความจริง (tru</mark>th table) และสัญลักษณ์ได้



#### หลักการพื้นฐานของ OR Gate ถ้ากำหนดให้ A และ B แทนตัวแปรอินพุตทั้งสอง เมื่อตัวแปร A มากระทำการ OR กับตัว แปร B ได้ผลลัพธ์เป็น Y ทำให้สามารถเขียน สมการลอจิกทางเอาต์พุตได้ดังนี้

Y = A + B

จากสมการลอจิก เครื่องหมาย (+) ไม่ใช่การบวกเลขแบบธรรมดาแต่จะเป็นการบวกแบบ OR ซึ่งสามารถเขียนตารางความจริง (truth table) และสัญลักษณ์ได้



#### หลักการพื้นฐานของ NOT Gate

ตัวกระทำ NOT ใช้กับตัวแปรอินพุตเดียว แตกต่างจากตัวกระทำ OR และ AND เช่น ถ้าให้ A แทนตัวแปรที่ป้อนอินพุตของตัว กระทำ NOT และได้ผลลัพธ์เป็น Y เขียนสมการลอจิกทางเอาต์พุตได้ดังนี้

สัญลักษณ์ขีด (bar) บนตัว A แทนการกระทำ NOT สมการ Y = NOT A หรือเท่ากับส่วน กลับของ A หรือเท่ากับ A bar สามารถเขียน ตารางความจริงและสัญลักษณ์ได้



#### หลักการพื้นฐานของNOR Gate

NOR Gate ที่มี 2 อินพุต การกระทำของ NOR Gate จะมีค่าเท่ากับการนำ OR Gate มาต่อ ร่วมกับ NOT Gate สามารถเขียนสมการ ลอจิกเอาต์พุตของ NOR Gate ได้ดังนี้

Y = ("A + B")

จากสมการลอจิกจะเห็นว่า NOR Gate มีการกระทำแรกเป็น OR ของอินพุตและการกระทำ ที่สองเป็นการกระทำ NOT บนผลบวกแบบ OR สามารถเขียนตารางความจริงและสัญลักษณ์ได้


หลักการพื้นฐานของ NAND Gate

NAND Gate ที่มี 2 อินพุต การกระทำของ NAND Gate จะมีค่าเท่ากับการนำ AND Gate มาต่อร่วมกับ NOT Gate ดังนั้น สมการลอจิกสำหรับเอาต์พุตของ NAND Gate จึงเขียนได้ดังนี้

#### $Y = \overline{A \cdot E}$

จากสมการลอจิกจะเห็นว่า NAND Gate มีการกระทำแรกเป็นการกระทำ AND ของอินพุต และการกระทำที่สองเป็นการกระทำ NOT บนผล คูณแบบ AND สามารถเขียนตารางความจริงและ สัญลักษณ์ได้

				A 6727-5 30	
ľ	А	В	$Y = \overline{A \cdot B}$		
	0	0	1		
	0	1	0	N 51 N	
	1	0	1	N. U. 0	
	1	1	0	Community	
ľ	ตารางความจริง (truth table)				สัญลักษณ์ของ NAND Gate
	0 0 1 1 ตา	0 1 0 1 รางความ	1 0 1 0 จริง (truth table)	N. Ū. U.	A B B สัญลักษณ์ของ NANE

หลักการพื้นฐานของExclusive OR Gate

เอาต์พุตของ Exclusive OR Gate จะมีค่าเป็น Logical 1 ก็ต่อเมื่ออินพุต มี Logical ต่างกัน และ เอาต์พุตจะมีค่าเป็น

Logical 0 ก็ต่อเมื่ออินพุตมี Logical เหมือนกัน ดังนั้นสมการลอจิกสำหรับ เอาต์พุตของ Exclusive OR Gate จึงเขียนได้ดังนี้

Y = A 🕀 B

สามารถเขียนตารางความจริงและสัญลักษณ์ของ Exclusive OR Gate ได้





# คุณสมบัติของ PLC SIEMENS รุ่น SIMATIC S7-200



โครงสร้างภายนอกของ PLC SIEMENS รุ่น SIMATIC S7-200

PLC SIEMENS รุ่น SIMATIC S7-200 โครงสร้างภายนอกถูกออกแบบมาให้มีส่วนประกอบทั้งหมด อยู่บนโครงสร้างเดียวกันดังนี้

- 1. หลอดแสดงโหมดการทำงานใช้สำหรับแสดงตำแหน่งโหมดการทำงาน PLC
- 2. ขั้วต่อทางด้านเอาต์พุตใช้สำหรับต่ออุปกรณ์เอาต์พุตเข้า PLC
- 3. จุดจ่ายไฟเข้า CPU ใช้สำหรับจ่ายไฟเข้า CPU เข้า PLC
- 4. สวิตช์เลือกโหมด RUN, TERM, STOP ใช้สำหรับเปลี่ยนโหมดการทำงานของ PLC
- 5. พอร์ต (Port) ใช้สำหรับขยายมอดูลอินพุ<mark>ต(Input) และเอาต์พุต (Output) ให้มีจ</mark>ำนวนที่มากขึ้น

6. สัญญาณแอนะลอกจำลองแบบปรับค่าได้ (0 –255) ใช้สำหรับเป็นสัญญาณแอนะลอกจำลอง แบบปรับค่าได้ (Analog adjustment) เพื่อนำค่ามาทดสอบโปรแกรมสัญญาณแอนะลอก

- 7. แหล่งจ่ายไฟ 24 DCV/180 mA เป็นแหล่งจ่ายไฟให้กับสวิตช์หรือเซนเซอร์ที่นำมาต่อเข้า PLC
- 8. ขั้วต่อทางด้านอินพุตใช้สำหรับต่ออุปกรณ์อินพุตเข้า PLC
- 9. จุดต่อสายเคเบิ้ลใช้สำหรับต่อสายเคเบิ้ล PC/PPI ระหว่างคอมพิวเตอร์กับ PLC



คุณลักษณะของ PLC SIEMENS รุ่น SIMATIC S7-200

ในงานควบคุมแบบอัตโนมัติมีขนาดของงานและรูปแบบของการควบคุมที่แตกต่างกันออกไป ผู้ใช้ สามารถเลือกใช้ PLC ให้มีความ เหมาะสมกับงานที่ต้องการควบคุม สามารถดูข้อมูลทางเทคนิคเฉพาะ แต่ละรุ่นเพื่อพิจารณานำไปใช้งาน รวมถึงจำนวนอินพุตและเอาต์พุต ของ PLC รุ่น SIMATIC S7–200

Specific technical data on the CPUs:	pecific technical data on the CPUs: 🖌 💦 👌				
Feature	CPU 2211	CPU 222'	CPU 2241	CPU 224XP <sup>1</sup> CPU 224XPsi <sup>2</sup>	CPU 2261
		Î. L			ad L
Integrated dig. inputs/outputs	6 DI/4 DO	8 DI/6 DO	14 DI/10 DO	14 DI/10 DO	24 DV16 DO
Digital inputs/outputs/max. number of channels with expansion modules	-	48/46/94	114/110/224	114/110/224	128/128/256
Analog inputs/outputs/max. number of channels with expansion modules	- :	16/8/16	32/28/44	2 Al/1 AO integrated 32/28/44	32/28/44
Program memory	4 KByte	4 KByte	8/12 KByte	12/16 KByte	16/24 KByte
Data memory	2 KByte	2 KByte	8 KByte	10 KByte	10 KByte
Storage of dyn. data via high-performance capacitor	typ. 50 h	typ. 50 h	typ. 100 h	typ. 100 h	typ. 100 h
High-speed counters	4 x 30 kHz, of which 2 x 20 kHz A/B counter usable	4 x 30 kHz, of which 2 x 20 kHz A/B counter usable	6 x 30 kHz, of which 4 x 20 kHz A/B counter usable	4 x 30 kHz, 2 x 200 kHz of which 3 x 20 kHz + 1 x 100 kHz A/B counter usable	6 x 30 kHz, of which 4 x 20 kHz A/B counter usable
Communications interfaces RS 485	1	1	1	2	2
Supported protocols:				both interfaces	both interfaces
- PPI master/slave	yes	yes	yes	yes	yes
- MPI slave	yes	yes	yes	yes	yes
- Freeport (freely config. ASCII protocol)	yes	yes	yes	yes	yes
Optional communications possibilities	not expandable	yes, PROFIBUS DP Slave and/or AS-Interface Master/Ethernet/ Internet/Modem	yes, PROFIBUS DP Slave and/or AS-Interface Master/Ethernet/ Internet/Modem	yes, PROFIBUS DP Slave and/or AS-Interface Master/Ethernet/ Internet/Modem	yes, PROFIBUS DP Slave and/or AS-Interface Master/Ethernet/ Internet/Modem
Built-in 8-bit analog potentiometer	1	1	2	2	2
(for commissioning, value change)					
Real-time clock	optional	optional	yes	yes	yes
Integrated 24-V-DC sensor supply volt.	max. 180 mA	max. 180 mA	max. 280 mA	max. 280 mA	max. 400 mA
Removable terminal strip	-	-	yes	yes	yes
Dimensions (W x H x D in mm)	90 x 80 x 62	90 x 80 x 62	120.5 x 80 x 62	140 x 80 x 62	196 x 80 x 62

เป็นตารางแสดงข้อมูลทางเทคนิคเฉพาะของ PLC SIEMENS รุ่น SIMATIC S7- 200 ทั้ง 4 CPU โดยจะบอกข้อมู ลทางเทคนิคตั้งแต่ CPU 221 จนถึง CPU ใหญ่สุดในรุ่น SIMATICS7-200 คือ CPU 226 ซึ่งข้อมูลทางเทคนิคจะบ่งบ อกถึงคุณลักษณะที่สำคัญในแต่ละรุ่น โดยในที่นี้จะขออธิบายใน CPU 222

Feature	CPU 222	ความหมาย
Integrated dig.	0.01/0.00	มีจำนวนดิจิทัลอินพุต 8 จุดและดิจิทัล
Inputs/Outputs	8 DI/6 DO	เอาต์พุต 6 จุด
Digital Inputs/Outputs/		สามารถขยายเพิ่มเป็นดิจิทัลอินพุต 48 จุด
max.number of channels	48/46/94	และดิจิทัลเอาต์พุต 46 จุด จำนวนสูงสุดใน
with expansion modules		การขยายของมอดูลไม่เกิน 224 จุด
Analog		สามารถขยายเพิ่มเป็นแอนะลอกอินพุต
Inputs/Outputs/max.	16/8/16	16 จุดและแอนะลอกเอาต์พุต 8 จุด จำนวน
number of channels with		สูงสุดในการขยายของมอดูลไม่เกิน 16 จุด
expansion modules		
Program memory	4 kbyte	ความจุหน่วยความโปรแกรม = 4 kbyte
Data memory	2 kbyte	ความจุหน่วยความจำข้อมูล = 2 kbyte
High – speed counters	4×30 kHZ, of Which 2×20	เคาน์เตอร์ความเร็วสูง 4 ช่อง ที่ความถี่ 30
	KHZ A/B counter usable	kHZ และ 2 ช่อง ที่ความถี่ 20 kHZ ใช้
		เคาน์เตอร์แบบ A/B
Communication interfaces	4	การเชื่อมต่อสื่อสารอินเตอร์เฟสด้วยพอร์ต
RS485	1	(Port) RS 485 มี 1 พอร์ต
Supported protocols		
- PPI Master/Slave	yes	
- MPI Slave	yes	3564311 เป็รเติดอิต
- Freeport	yes	
Optional communications	yes , Profibus DP Slave	รองรับการเชื่อมต่อสื่อสารแบบ
possibilities	AS-interface	Profibus DP Slave AS-interface
	Master/Ethernet/Internet/	Master/Ethernet/internet
	Modem	Modem

Feature	CPU 222	ความหมาย
Built-in 8 bit analog	1	มีแอนะลอกแบบปรับค่าได้ ขนาด 8 บิต
potentiometer	I	
Real – time clock	optional	ไม่มีนาฬิกา Real – time
Removable terminal strip	-	ไม่มีแถบขั้วต่อแบบถอดได้
Dimensions (W×H×D	90×80×62	โครงสร้างภายนอกขนาดความกว้าง สูง ลึก
mm.)		กว้าง 90 mm สูง 80 mm และลึก 62 mm





อุปกรณ์ต่อขยาย (Expansion Modules) หมายถึง อุปกรณ์ที่นำมาต่อร่วมกับ PLC เพื่อเพิ่มจำนวน อินพุตหรือเอาต์พุตของ PLC ให้

มีจำนวนมากขึ้น มีให้เลือกเพิ่มได้ทั้งแบบดิจิทัล (Digital) และแบบ แอนะลอก (Analog) หรืออินพุตเอาต์พุตแบบพิเศษเฉพาะงาน





อุปกรณ์สั่งงานและจอแสดงผล (Text Display and Touch Screen) เป็นอุปกรณ์ที่ทำหน้าที่สื่อสาร ข้อมูลกับ PLC โดยสื่อสารระหว่

างผู้ปฏิบัติงานกับเครื่องจักร สามารถแสดงผลตั้งค่าข้อมูล สั่งงาน กำหนดค่า ปรับตั้งค่า รวมถึงการตรวจสอบข้อมูลของเครื่องจักร มี Port

สื่อสารได้หลายรูปแบบ เช่น Series Port และ Ethernet Port เป็นต้น



#### สาย PC/PPI cable

สาย PC/PPI cable ใช้ในการเชื่อมต่อเพื่อสื่อสารข้อมูลระหว่างคอมพิวเตอร์กับ PLC ของ Siemens รุ่น S7–200 ลักษณะของหัวต่ อสายเคเบิ้ล PC/PPI จะมีให้เลือกใช้ 2 แบบ คือ แบบหัวต่อ USB และแบบ RS232 (COM Port) ดังแสดงในรูปที่ 3.6 ในการเลือกใช้สาย เคเบิ้ล หากเป็นสายเคเบิ้ลหัวต่อสายแบบ USB ผู้ใช้ไม่ต้องปรับตั้งอัตราความเร็วในการรับ-ส่งข้อมูล (Baud Rate) แต่หากเป็นสายเคเบิ้ลหัวต่ อสาย แบบ RS232 จะต้องปรับตั้ง Dipswitch (Dual in-line package switch) เพื่อกำหนดอัตราความเร็วในการ รับ-ส่งข้อมูลให้มีค่า มาตรฐานเท่ากับ 9.6 kbps







ตำแหน่งพื้นที่ใช้งานของ PLC SIEMENS รุ่น SIMATIC S7-200

ในการเขียนโปรแกรมสั่งงาน PLC ถ้าไม่กำหนดตำแหน่ง Address ของหน่วยความจำให้ตรงกับ ตำแหน่งอินพุตและเอาต์พุตของ PLC จะส่งผลให้ไม่สามารถทำงานได้ดังนั้นจึงควรทราบถึงหน่วยความจำ ขนาดของข้อมูล ตำแหน่ง (Address) ในการเข้าถึงข้อมูลของ S7–200 มีดังนี้

การเข้าถึงข้อมูลของ S7-200

S7-200 เก็บข้อมูลไว้ในตำแหน่งหน่วยความจำต่าง ๆ ที่มีที่อยู่เฉพาะจะต้องระบุตำแหน่ง (Address) หน่วยความจำที่ต้องการเข้ าถึงให้ชัดเจน ช่วยให้โปรแกรมสามารถเข้าถึงข้อมูลโดยตรง

Representation	Byte (B)	Word (W)	Double Word (D)
Unsigned Integer	0 to 255	0 to 65,535	0 to 4,294,967,295
	0 to FF	0 to FFFF	0 to FFFF FFFF
Signed Integer	-128 to +127	-32,768 to +32,767	-2,147,483,648 to +2,147,483,647
	80 to 7F	8000 to 7FFF	8000 0000 to 7FFF FFFF
Real IEEE 32-bit	Not applicable	Not applicable	+1.175495E-38 to +3.402823E+38
Floating Point			- 1.175495E-38 to -3.402823E+38

การเข้าถึง Bit ในพื้นที่หน่วยความจำต้องระบุพื้นที่หน่ วยความจำและตำแหน่ง Byte และ Bit ตัวอย่างดังรูปที่ 3.7 เรียกอีกอย่างหนึ่งว่า "ไบต์.บิต" (Byte.Bit) เช่น I0.4 เมื่อ I คือ อินพุต, 0 คือ Byte ที่ 0 (ตามด้วยจุด "." เพื่อแยกตำ แหน่งของบิต) และ 4 คือ Bit ที่ 4





การเข้าถึงข้อมูลในพื้นที่หน่วยความจำ (V, I, Q, M, S, L และ SM) ใช้รูปแบบ byte Address ในการ เข้าถึงข้อมูลไบต์เวิร์ ดหรือ ดับเบิ้ลเวิร์ด ในหน่วยความจำต้องระบุตำแหน่งในลักษณะ คล้ายกับการระบุ ตำแหน่งของบิตรวมถึงการระบุพื้นที่การกำหนด ขนาดข้อมูลและตำแหน่งไบต์เริ่มต้นของไบต์เวิร์ดหรือ ดับเบิ้ลเวิร์ด ดังแสดงในรูปที่ 3.8 และข้อมูลในพื้นที่หน่วยความจำอื่น ๆ (เช่น T, C, HC และ Accumulators) สามารถเข้าถึงได้โดยใช้รูปแบบ ตำแหน่งซึ่งรวมถึงตัวระบุพื้นที่และหมายเลขอุปกรณ์ ขอบเขตและคุณลักษณะของหน่วยความจำ PLC SIEMENS รุ่น SIMATIC S7–200 การติดต่อกับ CPU ต้องทราบขอบเขตของหน่วยความจำ หากกำหนดค่าเกินขอบเขตที่ กำหนดจะส่งผลทำให้โปรแกรมเกิดผิดพลาด ซึ่งขอบเขตหน่วยความจำมีดังตาราง

นอกจากนี้ใน PLCSIEMENS รุ่น SIMATIC S7–200 จะมีหน่ วยความจำพิเศษไว้ให้ใช้งาน เพื่อเป็น เครื่องมือที่ช่วยให้เขียน โปรแกรมได้ง่ายขึ้นและสั้นลง ซึ่งมีหน่วยความจำพิเศษจำนวนมาก ตัวอย่าง หน่วยความจำพิเศษที่นำไปใช้งานบ่อย ๆ มีดังตาราง

หน่วยความจำพิเศษ	ตำแหน่ง	ความหมาย
Always_On	SM0.0	Always ON
First_Scan_On	SM0.1	ON for the first scan cycle only
Olaph, 60p	0140.4	Clock pulse that is ON for 30 s
CIOCK_6US	SM0.4	OFF for 30 s, for a duty cycle time of 1 min
Olecik 1a	0N0 5	Clock pulse that is ON for 0.5 s
CIOCK_IS	SIMI0.5	OFF for 0.5 s, for a duty cycle time of 1 s.
Marta Ostitati	0140 7	Indicates the current position of the mode switch: 0 =
Mode_Switch SM0.		TERM, 1 = RUN
Pot0_Value	SMB28	Set to value corresponding to analog adjustment 0
Pot1_Value	SMB29	Set to value corresponding to analog adjustment 1

Description	CPU221	CPU222	CPU224	CPU226
User program size	2 K	words	4 Kw	rords
User data size	1 Kwords		2.5 Kwords	1 Kwords
Process-image input register		10.0 to	115.7	
Process-image output register		Q0.0 to	Q15.7	
Analog input (read only)		AIW0 to AIW30		•
Analog output (write only)			AQW0 to AQW30	
Variable memory (V)		VB0 to	VB2047	
Local memory (L)		LB0 to	LB63	
Bit memory (M)		M0.0 to	M31.7	
Special Memory (SM)	SM0.0 to	SM0.0 to	SM0.0 to	SM0.0 to
	SM179.7	SM299.7	SM179.7	SM299.7
Read only Memory		SM0.0 to	SM29.7	
Timer		256 (TO	to T255)	
Timer Retentive on-delay 1 mS		Т0,	T64	
Timer Retentive on-delay 10 ms		T1 to T4, and	d T65 to T68	
Timer Retentive on-delay 100 ms	T5 to T31, and T69 to T95			
Timer On-off delay 1 mS		T32,	T96	
Timer On-off delay 10 ms	T33 to T36, and T97 to T100			
Timer On-off delay 100 ms	T37 to T63, and T101 to T255			
Counters		C0 -	C255	
High-speed counters	HC0,HC3,H	IC4 and HC5	HC0,HC3,H	C4 and HC5
Sequential control relay (S)		S0.0 to	S31.7	
Accumulator registers	AC0 to AC3			
Jumps/Labels	0 to 255			
Call/Subroutine	0 to 63			
Interrupt routines	0 to 127			
Positive/negative transitions	256			
PID loops		0 t	o 7	
Ports		Port 0		Port 0



## โปรแกรม STEP 7-Micro/WIN



ข้อแนะนำก่อนการติดตั้งโปรแกรม STEP 7-Micro/WIN V4.0 SP9

ก่อนการติดตั้งควรเตรียมอุปกรณ์สำหรับการติดตั้งดังนี้

- เครื่องคอมพิวเตอร์พื้นที่ Hard Disk ไม่น้อยกว่า 50 MB
- ระบบปฏิบัติการ Windows XP หรือ Windows 7 (32 bit หรือ 64 bit)
- จอ VGA หรือจอที่มีความละเอียด 1024 × 768 ขึ้นไป
- โปรแกรม STEP 7-Micro/WIN V4.0 SP9
- สายเคเบิ้ล PC/PPI (USB or RS232 Type)
- PLC SIEMENS รุ่น SIMATIC S7–200

ก่อนการติดตั้งโปรแกรม STEP 7–Micro/WIN V4.0 SP9 จะต้องติดตั้งโปรแกรม STEP 7 V4.0 SP7 ก่อน แล้วจึงจะสามารถติดตั้งโปรแกรม V4.0 SP9 ได้หากข้ามขั้นตอนไปติดตั้งโปรแกรม V4.0 SP9 โดยไม่ ติดตั้งโปรแกรม V4.0 SP7 ระบบจะแสดงข้อความดัง



ให้ติดตั้งโปรแกรม STEP 7-Micro/WIN V4.0 SP7 โดยทำตามขั้นตอนการ ติดตั้งไปเรื่อย ๆจนครบ ระบบจะให้ Restart เครื่อง รอจนระบบ Restart เครื่อง เสร็จ จากนั้นจึงเริ่มขั้นตอนการติดตั้งโปรแกรม STEP 7-Micro/WIN V4.0 SP9 การติดตั้งโปรแกรม STEP 7-Micro/WIN V4.0 SP9

ขั้นตอนการติดตั้ง โปรแกรม STEP 7-Micro/WIN V4.0 SP9 มีขั้นตอนดังนี้

- 1. ใส่แผ่นซีดี (CD) โปรแกรม STEP 7-Micro/WIN V4.0 SP9 ลงใน Drive CD ROM
- 2. เปิด Windows Explorer คลิก (Click) เลือก Drive CD-ROM จากนั้น ดับเบิ้ลคลิก (Double Click) ที่ไอคอน Setup



3. จะปรากฏหน้าต่างเลือกภาษาที่ใช้ในโปรแกรมติดตั้งขึ้นมา ให้คลิกเลือก English (United States) จากนั้น คลิกปุ่ม NEXT ระบบ จะเริ่มดำเนินการขั้นตอนในการติดตั้ง

STEP 7-Micro/WIN - JackatShield Wizard Choose Setup 1 Select the language installation from the choices below.	STEP 7-Micro/WIN - InstallShield Wizard
English (United States) Frierch (Standard) German Itelian Spanish	Cancel

4. จะมีข้อความให้ Uninstall โปรแกรม STEP 7–Micro/WIN V4.0 SP7 ใน Add/Remove Programs ให้คลิกปุ่ม OK



7. ระบบจะเริ่มขั้นตอนถอนการติดตั้ง เมื่อเสร็จสิ้นระบบจะสั่ง Restart เครื่องให้ Restart ตาม คำแนะนำ จากนั้นจึงเริ่มขั้นตอนในการติดตั้ง โปรแกรม STEP 7–Micro/WIN V4.0 SP9 ใหม่อีกครั้ง โดยให้ ดับเบิ้ลคลิกที่ไฟล์ Setup ในแผ่นซีดีโปรแกรม STEP 7–Micro/WIN V4.0 SP9



8. จะปรากฏหน้าต่างเลือกภาษาที่ใช้ในโปรแกรมติดตั้งขึ้นมา ให้คลิกเลือก English (United States) จากนั้นคลิกปุ่ม NEXT

Choose Setup Language	lation from the choices below	
Select the language for the initia		
English (United States)		
French (Standard)	_	
Italian Spanish		C
		~
		1.91
	¥	
InstallShield		

#### 9. จะปรากฏหน้าต่างเริ่มต้นขั้นตอนการติดตั้งให้คลิก Next แล้วจะปรากฏหน้าต่างส่วน License Agreement ให้คลิกปุ่ม Yes



10. เลือกตำแหน่งโฟลเดอร์ปลายทาง (Choose Destination Folder) ที่จะติดตั้งโปรแกรมหาก ต้องการเปลี่ยนให้คลิก Browse เพื่อเลือก โฟลเดอร์ใหม่ หรือถ้ายอมรับโฟลเดอร์ปลายทางการติดตั้งตามที่ โปรแกรมกำหนดให้คลิกเลือก Next แล้วระบบจะเริ่มดำเนินการติดตั้ง

STEP 7-Micro/WIN - InstallShield Wizard		STEP 7-Micro/WIN - InstallShield Wizard	×
Choose Destination Location Select toker where Setup will install files.		Setup Status	
Setup will install STEP 7-Micro/WIN in the following folder.		STEP 7-Micro/WIN Setup is performing the requested operations.	
To install to this folder, click Next. To install to a different folder, click Browse and select another folder.	n d i v.		
	2	C.\\STEP 7-MicroWIN V4.0\WhatsNew\A\index_files\slide0248.htm	
	al and a mini		
Destination Folder	and Commun		
C:\Program Files\Siemens\STEP 7-MicroW/IN V4.0 Browse			
Instal/Shield		InstallShield	
< Back Next> Cancel		0	Cancel

11. การติดตั้งจะดำเนินต่อไปให้รอจนกระทั่งปรากฏหน้าต่าง InstallShield Wizard Complete แสดงว่าขั้นตอนในการติดตั้งสมบูรณ์ จากนั้นคลิกFinish



12. เมื่อเปิดเครื่องคอมพิวเตอร์และเข้าสู่หน้าต่างปกติของระบบ Windows ให้สังเกตไอคอน โปรแกรม STEP 7–Micro/WIN V4.0 SP9 บน หน้าต่าง Desktop ให้ดับเบิ้ลคลิกเมาส์ทางซ้ายที่ไอคอน โปรแกรม STEP 7 รอสักครู่หน้าต่างแรกของโปรแกรมจะปรากฏขึ้น



ส่วนประกอบของโปรแกรม STEP 7-Micro/WIN V4.0 SP9

ส่วนประกอบของโปรแกรม STEP 7-Micro/WIN V4.0 SP9 มีความหมายดังนี้



1. Menu bar เป็นแถบเมนูซึ่งแสดงคำสั่งของโปรแกรม ได้แก่ กลุ่มคำสั่ง File, Edit, View, PLC, Debug, Tool. Windows, Help 2. Toolbars เป็นแถบเครื่องมือซึ่งแสดงรูปของไอคอนคำสั่งที่ เรียกใช้งานบ่อย 3. Navigation bar เป็นแถบน้ำทางซึ่งแสดงกลุ่มของการ ควบคุมได้แก่ กลุ่ม View ปุ่มควบคุมสำหรับ Program Block, Symbol Table, Status Chart Data block, Cross Reference, Communication และ Set PG/PC Interface กลุ่ม Tool ปุ่มควบคุมสำหรับคำสั่ง Wizard

4. Output Windows เป็นหน้าต่างเอาต์พุตที่แสดงข้อมูลเมื่อติดต่อสื่อสารกับ PLC เช่น การ ตรวจสอบโปรแกรม (Compile) โปรแกรมจะ แสดงข้อความรายละเอียดในการตรวจสอบ

5. Instruction Tree เป็นส่วนแสดงกลุ่มคำสั่งที่ใช้ในการเขียนโปรแกรม สามารถคลิกเมาส์ที่ เครื่องหมาย + หน้าโฟลเดอร์เพื่อเปิดชุดคำสั่ง โดยสามารถลากหรือดับเบิ้ลคลิกคำสั่งที่ต้องการ เพื่อมาวาง ในส่วนพื้นที่เขียนโปรแกรม Program Editor

6. Status bar เป็นแถบสถานะแสดงการทำงานของโปรแกรม STEP 7-Micro/WIN

7. Program Editor หรือ Main Program เป็นส่วนพื้นที่ในการเขียนโปรแกรม



#### แถบเครื่องมือมาตรฐาน (Standard Toolbar)

#### แถบเครื่องมือมาตรฐานมีสัญลักษณ์ชื่อและความหมายตามตารางดังนี้

8	จัญลักษณ์และชื่อคำสั่ง	ความหมาย
r	New Project	สร้างงานใหม่
<b>*</b>	Open Project	เปิดงานที่สร้างไว้แล้ว
	Save Project	บันทึกงานที่กำลังทำอยู่
<b>a</b>	Print	สั่งพิมพ์ ออกทางเครื่องพิมพ์
Cà.	Print Preview	ดูงานก่อนสั่งพิมพ์
Ж	Cut	ตัดข้อมูลส่วนที่เลือกไว้ ออกจากงานและสามารถนำข้อมูลใน ส่วนที่ถูกตัด นำไปวางไว้ในตำแหน่งที่ต้องการ
	Сору	คัดลอกข้อมูลส่วนที่เลือกไว้จากงานและวางไว้ในตำแหน่งที่ ต้องการ
<b>R</b>	Paste	นำข้อมูลส่วนที่เลือกไว้จากคำสั่ง Cut หรือ Copy วางไว้ตำแหน่งที่ต้องการ
ŝ	Undo	ยกเลิกคำสั่งที่เพิ่งกระทำก่อนหน้า หรือที่เพิ่งกระทำล่าสุด
V	Compile	ตรวจสอบข้อผิดพลาดในหน้าที่กำลังเขียนอยู่ในปัจจุบัน
	Compile All	ตรวจสอบข้อผิดพลาดทั้งหมด ประกอบด้วยProgram Block , Data Block และ System Block ในการออกแบบโปรแกรมงาน ปัจจุบัน

	8	งัญลักษณ์และชื่อคำสั่ง	ความหมาย
			ดึงข้อมูลหรือโปรแกรมจาก PLC ขึ้นมายัง
í	-	Upload	โปรแกรม STEP 7–Micro/WIN ในงานใหม่
	<u> </u>	Download	น้ำงานที่ออกแบบในโปรแกรม Download ลงใน PLC
	8.1	Cort According	เรียงลำดับชื่อ หรือแอดเดรสจากอักษร A – Z ในหน้า Symbol
/	24	Son Ascending	Table และ Status Chart
2	8.*	Sort Descending	เรียงลำดับชื่อ หรือแอดเดรสจากอักษร Z – A ในหน้า Symbol
ł	21 Sort Descending		Table และ Status Chart
1	12	Option	เปิดหน้าต่างข้อกำหนด (Option)เพื่อเข้าไป ค่าต่าง ๆ

#### แถบเครื่องมือตรวจสอบแก้ไข (Debug Toolbar)

#### แถบเครื่องมือตรวจสอบแก้ไขมีสัญลักษณ์ชื่อและความหมายตามตารางดังนี้

สัญร์	จักษณ์และชื่อคำสั่ง	คำสั่ง : ความหมาย	สัญลัก
<b>R</b>	Pause Trend View	หยุดแสดงค่ากราฟ แผนภูมิสถานะ ค่าภายในแอดเดรสของ PLC ที่ต้องการให้ค้างสภาวะปัจจุบัน	
60^	Single Read	ดูค่าภายในแอดเดรสของ PLC ที่ต้องการบนหน้า Status Chart ที่ค่าปัจจุบันแค่ครั้งเดียวเท่านั้น	第
5	Write All	เขียนค่าต่าง ๆ ลงไปใน Address ที่ต้องการของ PLC	
Ĥ	Force	บังคับสถานะการทำงาน Input (อินพุต), Output (เอาต์พุต) ให้ On–Off โดยไม่ขึ้นกับโปรแกรม	
f	Unforce	ยกเลิกการบังคับสถานะการทำงานอิน Input (อินพุต), Output (เอาต์พุต) เฉพาะส่วนที่เลือกเท่านั้น	
ť	Unforce All	ยกเลิกการบังคับสถานะกานทำงาน Input (อินพุต), Output (เอาต์พุต) ทั้งหมด	ū. u.
6	Read All Forced	ใช้ในการอ่าน Contact หรือ Coil (คอยล์) ที่ถูกบังคับสถานะ ทั้งหมดโดยแสดงที่หน้า Status Chart	mmunity Educ
	Trend View	แสดงค่ากราฟ แผนภูมิสถานะค่าภายในแอดเดรสต่าง ๆ ของ PLC	

สั	ญลักษณ์และชื่อคำสั่ง	คำสั่ง : ความหมาย
►	RUN	เริ่มต้นขั้นตอนสั่งการทำงานของ PLC
	STOP	หยุดขั้นตอนสั่งการทำงานของ PLC
<b>6</b>	Program Status	ดูสภาวะการทำงานของโปรแกรมภายใน PLC
器	Pause Program Status	หยุดดูสภาวะการทำงานของโปรแกรมภายใน PLC ชั่วคราว
<b>F</b>	Chart Status	ดูค่าภายในแอดเดรสของ PLC ที่ต้องการบนหน้า Status Chart แบบ Real time

#### แถบเครื่องมือทั่วไป (Common Toolbar)

#### แถบเครื่องมือทั่วไปมีสัญลักษณ์ชื่อและความหมายตามดังนี้

	สัญลักษณ์และชื่อคำสั่ง	ความหมาย
HHO	Delete Network	ลบ Network ออกจากพื้นที่การออกแบบโปรแกรม
	Toggle POU Comments	เปิด / ปิด การแสดงข้อความอธิบายโปรแกรม POU (Program Organizational Unit)
IN IN	Toggle Network Comments	เปิด / ปิด การแสดงข้อความอธิบายบนNetwork
部	Toggle Symbol Information Table	เปิด / ปิด หน้าต่างรายละเอียดข้อความของ Symbol , Address และ Comment
<b>*</b>	Toggle Bookmark	เปิด / ปิด การเลือกที่คั่น Network ที่ต้องการ ดูสถานะการทำงานต่าง ๆ
≫	Next Bookmark	การเคลื่อนย้ายที่คั่น Network ถัดไปบนโปรแกรม
*	Previous Bookmark	การเคลื่อนย้ายที่คั่น Network ที่ผ่านมาบนโปรแกรม
1	Remove All Bookmarks	ยกเลิกการใช้งานที่คั่น Network ทั้งหมด
CH+	Apply All Symbol in Project	ปรับปรุงการแก้ไขรายชื่อของ Symbol ในตารางแสดง รายละเอียดของข้อความของ Symbol , Address และ Comment
sym	Create Table Undefined Symbols	ตรวจเซ็คการกำหนดรายชื่อของ Symbol ถ้ามี ข้อผิดพลาด (Error) จะสร้างหน้า Symbol Table เพิ่มขึ้นมาทันที

	สัญลักษณ์และชื่อคำสั่ง	ความหมาย
ню	Insert Network	แทรกNetwork เพิ่มเข้าไปในพื้นที่การออกแบบ โปรแกรม
78	2	

### แถบเครื่องมือคำสั่ง(Instruction Toolbar) แถบเครื่องมือคำสั่งมีสัญลักษณ์ชื่อและความหมายตามดังนี้

สัญลัก	เษณ์และชื่อคำสั่ง	ความหมาย
7	Line Down	การเชื่อมต่อเส้นลงด้านล่าง
←	Line Left	การเชื่อมต่อเส้นไปทางซ้าย
<b>→</b>	Line Right	การเชื่อมต่อเส้นไปทางขวา
⊣⊦	Contact	ชุดคำสั่งที่อยู่ในรูปแบบ Contact เช่น Normally Open, Compare, Positive Transition
-0	Coil	ชุดคำสั่งที่อยู่ในรูปแบบ Coil เช่น Output , Jump – Label, Set, Reset
1	Box	ชุดคำสั่งที่อยู่ในรูปแบบ Box เช่น Timer, Counter, Move

tial and Community

แถบดำเนินการ (Navigation Bar)

แถบดำเนินการเป็นแถบที่ใช้เลือกคำสั่งต่าง ๆ ของโปรแกรม เช่น คำสั่ง Program Block, Communication เป็นต้น มีสัญลักษณ์ชื่อ และความหมายตามตารางดังนี้

สัญลั	ักษณ์และชื่อคำสั่ง	ความหมาย
	Program Block	เปิดหน้าต่างที่ใช้สร้างโปรแกรมหรือจัดการในส่วนของโปรแกรม
题	Symbol Table	ใช้สำหรับกำหนดชื่อคำสั่งหรือสัญลักษณ์และรายละเอียดต่าง ๆ
	Status Chart	ใช้แสดงค่าสถานะของแอดเดรสต่าง ๆ ภายใน PLC เช่น Timer, Counter, V Memory
	Data Block	ใช้ในกำหนดข้อมูลให้กับ PLC
	System Block	เป็นการกำหนดแก้ไขค่าระบบต่าง ๆ ของ PLC
-	Cross Reference	ใช้ในการดูรายละเอียดต่าง ๆ ทั้งหมดของแอดเดรส
	Communications	ใช้ในการติดต่อสื่อสารระหว่างคอมพิวเตอร์กับ PLC
	Set PG/PC Interface	การกำหนดรูปแบบการเชื่อมต่อที่ใช้ในการติดต่อข้อมูล

#### 4.3.6 ส่วนประกอบต่าง ๆ ของ Instruction Tree ในโปรแกรม STEP 7-Micro/WIN



ส่วนของข้อมูลและรายละเอียดของ Project เช่น Program Block, Symbol Table, Tool เป็นต้น



ส่วนของชุดคำสั่งของ Step 7 Micro/win เช่น กลุ่มคำสั่ง Bit Logic, Counters, Timers เป็นต้น การติดต่อสื่อสารระหว่างคอมพิวเตอร์กับโปรแกรมเมเบิลคอนโทรลเลอร์

การติดต่อสื่อสาร (Communication) สามารถทำได้ดังนี้

 ในส่วนของ Navigation bar จะมีไอคอน Communication ที่ใช้สำหรับติดต่อสื่อสารข้อมูล ระหว่างคอมพิวเตอร์กับ PLC ได้โดย เลื่อนเมาส์ไปชี้ที่ Communication คลิกเมาส์ซ้าย 1 ครั้ง



and Commun

 จะปรากฏหน้าต่าง Communications ขึ้นมา ให้เลื่อนเมาส์ไปชี้ที่ข้อความ Double Click to Refresh จากนั้นดับเบิ้ลคลิกเมาส์ซ้าย 1 ครั้ง

Local Remote: PLC Type:	0	Address: 0	
Vpdate PLC type in proje	ect		
Network Parameters			2
Interface:	PC/PPI cable(USB)		
Protocol:	PP1		
Mode:	11-bit		
Highest Station (HSA):	31 rs		
Transmission Rate	2220		
Baud Rate:	9.6 kbps		

 หากติดต่อสื่อสารข้อมูล (Communication) ระหว่าง PLC กับคอมพิวเตอร์สำเร็จ จะปรากฏรูป PLC พร้อมข้อความบอกขนาด CPU ของ PLC ขึ้นมา

.ocal:	0	Address: 0
Remote:	2 💌	CPU 224 REL 02.00
PLC Type:	CPU 224 REL 02.00	Address: 2, 9.6 kbps
Update PLC type	in project	

 หากการติดต่อสื่อสารข้อมูล (Communication) ระหว่าง PLC กับคอมพิวเตอร์ไม่สำเร็จ จะแสดง ข้อความว่า Error: Communications time-out ดังรูปที่ 4.24 แสดงถึงการเชื่อมต่อมีข้อผิดพลาดให้ ตรวจสอบการเชื่อมต่อสายเคเบิ้ล ความเร็ว Baud Rate และการตั้งค่า Port (พอร์ต) ที่ Set PG/PC Interface ทำได้โดยการเลื่อนเมาส์ไปที่ Set PG/PC Interface คลิกเมาส์ซ้าย 1 ครั้ง

Comn	nunications			×		
L.	ddress 	0	T 🖶 PC/PPI Cable	(PPI)		
	Remote:		Address: 0	-Click to Refresh		
1 8	PLC Type:	CPU 224 REL 02.00	Error: C Check	ommunications time-out, the port number, network	2	
	✓ Update PLC type in proj	ect	attache	c baub rate, and the d cable.	5	
-N	etwork Parameters					
	nterface:	PC/PPI cable(USB)				
	Protocol:	PPI				
	lode:	10-bit				
	lighest Station (HSA):	31				
	Supports multiple maste	ers				
C D	ansmission Rate					
34	Baud Rate:	9.6 kbps				
2	Search all baud rates					
	Set PG/PC Interface			K Cancel		
2 L	ข้อ	ia ta -	SIL 1994 N	וע	8	
ชาย 1 ค	เรงดงรูเ	Jท 4.25 จ <b>ะ</b>	ะปรากฏเ	<i>ง</i> นาตาง	Propertie	es– PC/PI
	ข		<b>6</b> 41			
erface		×	Proper	ties - PC/PPI cable(PPI)	_	×
LLDP / DCP PNIO Adapter				Lord Consulting 1		
nt of the Application:			PPI	Local Connection		-1
<ul> <li>V -&gt; PC/PPI cable(PPI)</li> </ul>	*	20				1
or Micro/WIN)			0	connection to:	USB	
arameter Assignment Used:				-		
able(PPI)	Properties		1	Modem connection		

5. เลื่อนเมาส์ไปที่ปุ่ม Properties คลิกเมาส์ซ้าย 1 ครั้งดังรูปที่ 4.25 จะปรากฏหน้าต่าง Properties- PC/PPI Cable (PPI) ขึ้นมาคลิกที่ Tap หัวข้อ Local Connection



6. ตรวจสอบการเชื่อมต่อพอร์ตที่ Connection to ว่ากำหนดพอร์ตถูกต้องหรือไม่ ถ้าสายเคเบิ้ลเป็น หัวต่อแบบ USB ให้ตั้งเป็น USB ถ้าเป็ นหัวต่อแบบ Serial ให้ตั้งเป็น COM Port เช่น Com1 หรือ Com2 ให้ ตรงกับหมายเลขของพอร์ตที่ต่ออยู่

S'	Connection to: USB	
2		
1		
16	5	

 หาก Communication แล้วปรากฏข้อความ Error ให้ตรวจ Error เบื้องต้น ดังนี้คือ (1) เชื่อมต่อสายPC/PPI cable ระหว่าง คอมพิวเตอร์กับ PLC แล้วหรือไม่
 (2) จ่ายไฟเข้าเครื่อง PLC แล้วหรือไม่
(3) ตั้งค่า Dipswitch กำหนดค่า Baud Rate ถูกต้องหรือไม่ (หัวต่อ RS232)

(4) ได้กำหนดพอร์ตหัวต่อสายเคเบิ้ลถูกต้องแล้วหรือไม่

8. หลังทำการตรวจเช็ค Error เบื้องต้นให้ Communication อีกครั้ง

tdress	PC/PPI Cable(PPI)
.ocal: 0	Address: 0
Remote: 2	Double-Click
VLC Type:	to Refresh
Undate DI C type in project	

การเปลี่ยนค่าพารามิเตอร์ของโปรแกรมเมเบิลคอนโทรลเลอร์

การเปลี่ยนค่าพารามิเตอร์ของ PLC สามารถทำได้ดังนี้

1. คลิกเปิดที่เมนู View เลื่อนลงมาที่ Component เลือก System Block หรือคลิกที่ไอคอน System Block บน Navigation Bar



 จะปรากฏหน้าต่าง System Block คลิกที่ Communication Port ค่า PLC แอดเดรสจะตั้งไว้ที่ 2 และคลิกค่า Baud Rate ของ PLC จะ ตั้งไว้ที่ 9.6 Kbps คลิกปุ่ม OK ถ้าต้องการเปลี่ยนค่าให้ใช้เมาส์คลิกเพื่อเปลี่ยน จากนั้นคลิกปุ่ม OK

communicate to a giver	Communication Ports
Communication Ports     Retentive Ranges     Retentive Ranges     Password     Output Tables     Jinput Filters     Pulse Catch Bits     Buckground Time     EM Configurations     Configure LED     Increase Memory	Ports         2           Port 0         Port 1           PLC Address:         2           Highest Address:         31           Baud Rate:         9.6 kbps           Retry Count:         3           Gap Update Factor:         10
Click for liels and Summert	Configuration parameters must be downloaded before they take effect.

3. คลิกที่ไอคอนคำสั่งดาวน์โหลด (Download) เพื่อ load ค่าพารามิเตอร์เข้า PLC

View PLC Debug Tools Window	vs Help
🎒 💪   X 🖻 🖷   🗤   🗹 🔯	▲ 🛓   8↓ 8↑
🚟 🏭 🔺 🛠 🕻 🧏 🗯 🖉	← → È ∓
Project1	
🔤 🔤 🖓 What's New	
🛱 CPU 224 REL 02.00	
🕀 🕀 💼 Program Block	
📄 🖻 🖻 Symbol Table	
📄 🕀 💼 Status Chart	
📄 📄 💼 Data Block	
📄 🕀 📲 System Block	

## ข้อกำหนดในการเขียนโปรแกรม

 จำนวนคอนแทคของ Input, Output, Auxiliary Relay (M), TIM, CNT สามารถนำมาใช้ได้ตาม ความต้องการของผู้เขียนโปรแกรม แต่การเขียนโปรแกรมที่ดีต้องพยายามเขียนโปรแกรมให้สั้น หรือน้อย ที่สุด เพื่อจะทำให้ความเร็วในการประมวลผลการทำงาน (Scantime) มีค่าน้อยลง

 การเขียนโปรแกรมเพื่อต่อคอยล์ให้กับ Bus (บัส) ทางซ้ายโดยตรงไม่สามารถทำได้ในกรณีที่ ต้องการให้มีลักษณะการทำงานเหมือนกับต่ อกับบัสทางซ้ายโดยตรง ให้นำรีเลย์พิเศษ SM0.0 (Bit Always On) มีสภาวะเป็น "On" เมื่อ Run โปรแกรมมาใช้งาน





3. ไม่สามารถเขียนโปรแกรมโดยที่มีคอนแทคอยู่ทางขวาของเอาต์พุตคอยล์ได้



- จำนวนคอนแทคที่ใช้ในการต่ออนุกรม หรือขนานสามารถใช้ได้ไม่จำกัดจำนวน ขึ้นอยู่กับความ ต้องการใช้ของผู้เขียน
   เอาต์พุตทุกเอาต์พุตมีคอนแทคช่วยสามารถใช้คอนแทคช่วยได้ไม่จำกัดจำนวน
- 6. เอาต์พุตคอยล์สามารถเขียนโปรแกรมให้ขนานกันได้เพื่อรับสัญญาณจากอินพุตตำแหน่ง เดียวกัน



7. ไม่สามารถเขียนโปรแกรมให้มีการทำงานของเอาต์พุตคอยล์ 2 จุด ที่ทำงานแยกจากกัน ใน Network เดียวกันได้





ให้ทดลองเขียนโปรแกรมตาม Ladder Diagram



ขั้นตอนการเขียนโปรแกรม

1. เปิดเครื่องคอมพิวเตอร์เมื่อเข้าสู่หน้าต่างปกติของระบบ Windows ดับเบิ้ลคลิกที่ไอคอน ของโปรแกรม STEP 7-Micro/WIN V4.0 SP9 เพื่อเข้าสู่โปรแกรม

2. ตรวจสอบการตั้งค่า Baud rate เท่ากับ 9.6 Kbps (กรณีที่สายแบบหัวต่อ RS232) เชื่อมต่อสายเคเบิ้ลระหว่าง PLC กับ คอมพิวเตอร์ตรวจสอบ Mode Selector Switch ให้ตั้งอยู่ตำแหน่ง TERM

3. ทำการติดต่อสื่อสาร(Communication) ระหว่างคอมพิวเตอร์กับ PLC

4. เขียนโปรแกรมตาม Ladder โดยเลื่อนเมาส์ไปชี้ที่คำสั่ง Bit Logic ใน Instruction Tree จากนั้นดับเบิ้ลคลิกเมาส์ซ้าย



6. การเลือกใช้คำสั่งที่ต้องการใช้งานให้เลื่อนเมาส์ไปชี้ที่คำสั่งนั้น ๆ จากนั้นดับเบิ้ลคลิก เมาส์ซ้าย

![](_page_79_Figure_1.jpeg)

8. เลื่อนเมาส์เพื่อเลือกคำสั่งต่อไป เขียนโปรแกรมต่อตาม Ladder

![](_page_80_Figure_1.jpeg)

9. เขียนโปรแกรมตาม Ladder ตัวอย่างจ<mark>น</mark>ครบ

10. การเรียกใช้คำสั่งเพื่อเข้าสู่หน้าจอหลัก (Main Program) มีวิธีลัด โดยไม่ต้องดับเบิ้ลคลิกที่คำสั่ง ทุกครั้ง แต่ใช้วิธีคลิกเพื่อเรียกคำสั่ง โดยตรง คือเลื่อนเมาส์ไปชี้ที่ Bit Logic คลิกเมาส์ขวา 1 ครั้งจะปรากฏ หน้าต่างการตั้งค่า ให้เลื่อนตัวเลือกลงมาที่คำสั่ง Single–Click to

Open an Item

![](_page_80_Figure_5.jpeg)

 การใช้คำสั่ง Single-Click to Open an Item หากต้องการเลือกใช้คำสั่งใด ๆ เพียงคลิกเมาส์ ซ้ายที่คำสั่งหน้าสัมผัสที่ต้องการ 1 ครั้ง คำสั่งนั้นจะเข้ามาใน Main Program ทันที (ไม่ต้องใช้วิธีดับเบิ้ล คลิกเมาส์ซ้าย ทำให้สามารถเขียนโปรแกรมได้รวดเร็วยิ่งขึ้น)

![](_page_81_Figure_1.jpeg)

12. เมื่อเขียนโปรแกรม Ladder ครบตามที่กำหนดให้พิมพ์ตัวอักษรเพื่อกำหนดตำแหน่งอินพุต เอาต์พุต โดยเลื่อนเมาส์ไปชี้ที่คำสั่งที่ต้องกา รพิมพ์กำหนดตำแหน่ง คลิกเมาส์ซ้าย 1 ครั้ง จะปรากฏส่วน คลิกเมาส์ขวา คลิกเมาส์ซ้าย 1 ครั้ง เพื่อเลือกใช้งาน 81 พื้นที่ให้พิมพ์พิมพ์ตำ แหน่งให้ตรงตามโปรแกรมแล้วกด Enter เคอร์เซอร์จะเลื่อนไปคำสั่งถัดไป ให้พิมพ์ ตำแหน่งอินพุตเอาต์พุตจนครบตามโปรแกรมตัวอย่าง

![](_page_81_Figure_3.jpeg)

#### 13. จะได้โปรแกรมตรงตามที่กำหนด

![](_page_82_Figure_1.jpeg)

14. ขั้นตอนต่อไป คือการตรวจสอบโปรแกรม (Compile) ก่อนจะดาวน์โหลดโปรแกรมเข้าเครื่อง PLC และทดสอบโปรแกรม (RUN) ตาม ลำดับต่อไป ในที่นี้ให้ผู้ใช้เลือกวิธีการตรวจสอบโปรแกรมทั้งหมด (Compile All) เพื่อตรวจสอบโปรแกรมและค่าต่าง ๆ ที่มีการกำหนดเรียก ใช้โดยเลื่อนเมาส์ไปชี้ที่คำสั่ง Compile All คลิกเมาส์ซ้าย 1 ครั้ง

Debug	Tools W	indows	Help		
Þ C	n   🕅	🖻 🔺	<b>_</b>	₿↓	81
% %	🎋   🏗 🤋	<u>~</u> ]]¬	, t	+	→

15. จะปรากฏข้อความจากการ Compile ขึ้นมาในส่วนของ Output Windows เช่น Total Errors = 0 แสดงว่าในการตรวจสอบทั้งหมด ไม่มีข้อผิดพลาด แต่ถ้ามีข้อผิดพลาด (Error) จะมีข้อมูลแสดงให้ทราบ ทันทีเช่น Total Errors = 1 หรือ 2 พร้อมแสดงรายละเอียดว่า

ผิดพลาดในจุดใด แถวใด ส่วนใด

![](_page_83_Figure_2.jpeg)

16. หาก Compile ผ่าน ไม่มี Error (Total Error = 0) ขั้นตอนต่อไปคือการ Download ข้อมูล ไปยังหน่วยความจำของ PLC (หากมีการแก้ไขหรือเปลี่ยนแปลงโปรแกรมจะต้อง Compile Program และดาวน์โหลดใหม่ทุกครั้ง)
 17. การดาวน์โหลดให้เลื่อนเมาส์ไปซี้ที่คำสั่งดาวน์โหลดบน Tool Bars คลิกเมาส์

Debug	Tools Windows Help
Þ.	\$\$   🗹 🗗   📥 🛨 👫
**	🏂   🏗 🗶 📗 ⊐ 🛨 ← →

	PPI Con	nection	
		Remote Address: 2	CEL 224 REL 02 00
		Click Download to begin.	
	Opt	sions ¥	Download Cancel
. ขั้นตอนการดาวน์โหลดจ	จะเริ่มขึ้นรอจนกระทั่งก <sup>า</sup>	ารดาวน์โหลดเส'	ร็จสมบูรณ์
		Download	
		PPI Connection Use the Options button to select blocks to downlo	
			ed.
		Remote Address: 2	ed. CPU 224 REL 02.00
		Remote Address: 2	ed. CPU 224 REL 02.00
		Remote Address: 2 Downloading System Block 100; Options 1 Options Doptions	ed. CPU 224 REL 02.00 Carcel
		Penote Address 2 Downloading System Block 1001 Options 2 Options Phogram Block Phogram Block Phogram Block Phogram Block Phogram Block Phogram Block Phogram Block Phogram Block Phogram Block	ed. CFU 224 REL 02.00 Carcel To - PLC To - PLC To - PLC To - PLC
		Renote Address 2 Downloading System Block 100: Options 2 Program Block. Program Block	ed. CPU 24 REL 02.00 Cancel To: PLC To: PLC

18. จะปรากฏหน้าต่างดาวน์โหลดขึ้นมา เลื่อนเมาส์ไปชี้ที่คำสั่งดาวน์โหลดคลิกเมาส์ซ้าย 1

20. หากดาวน์โหลดเสร็จสมบูรณ์จะปรากฏข้อความในส่วน Output Windows ข้อความว่า Download to PLC Download was Successful (Download ไปที่ PLC ดาวน์โหลดเสร็จสิ้นสมบูรณ์)

Set PG/PC	●-② Timets ●-∰ Libraries ⊕- Call Subroutines	
Downloading to PLC Download was Succe Ready	essful	

21. ขั้นตอนต่อไปคือการทดสอบการทำงานของโปรแกรม (Run Program) ให้เลื่อนเมาส์ไปซี้ที่คำสั่ง Run บนแถบ Tool bars คลิกเมาส์ซ้ าย 1 ครั้ง

Windo	ws H	elp			
<b>V</b>	<b>≜</b> 2	<b>-</b>   £↓ £	†   🖪		85
🕈 🖭	- ↓	± ← →		0 1	

22. จะปรากฏหน้าต่าง RUN ขึ้นมา พร้อมคำถาม Place the PLC in RUN mode (ต้องการเปลี่ยน โหมดเป็นโหมด Run หรือไม่) ถ้าต้อง การเปลี่ยนให้ตอบ Yes ถ้าไม่ต้องการเปลี่ยนให้ตอบ No (ในที่นี้ให้ ตอบ Yes)

RU	N	<b>×</b>	
C	Place the PLI	C in RUN mode?	
	Yes	No	

23. สังเกตหลอดไฟแสดงโหมดการใช้งานที่ PLC จะเลื่อนไปติดที่โหมด RUN ขั้นตอนต่อไป เลื่อน เมาส์ไปชี้ที่คำสั่ง Program Status บน Tool Bars จากนั้นคลิกเมาส์ซ้าย 1 ครั้ง เพื่อแสดงสถานะการ ทำงานของโปรแกรมควบคู่ไปด้วย

![](_page_86_Figure_1.jpeg)

24. โปรแกรมจะปรากฏเป็นเส้นสีน้ำเงินเพื่อแสดงสถานการณ์ทำงาน ทำให้รู้สถานะของอินพุตและ เอาต์พุต และมองโปรแกรมได้ชัดเจน ยิ่งขึ้น ขณะทดสอบโปรแกรม

![](_page_86_Figure_3.jpeg)

25. การเปลี่ยนโหมด PLC กลับไปที่โหมด STOP กรณีที่โปรแกรมที่เขียนเกิดข้อผิดพลาดต้องการ แก้ไขโปรแกรมหรือต้องการหยุดการทำงาน ของเครื่องทำได้โดยการยกเลิกการทำงาน Program Status ก่อน คือการเลื่อนเมาส์ไปชี้ที่คำสั่ง Program Status คลิกเมาส์ซ้าย 1 ครั้ง เส้

นสีของโปรแกรมจะกลับเข้าสู่ รูปปกติ

![](_page_86_Figure_6.jpeg)

26. ขั้นตอนต่อไปคือ การเปลี่ยนโหมด PLC กลับไปเป็นโหมด STOP ทำได้โดยการเลื่อนเมาส์ไปชี้ที่ คำสั่ง STOP

![](_page_87_Figure_1.jpeg)

27. คลิกเมาส์ซ้าย 1 ครั้ง จะปรากฏหน้าต่าง STOP พร้อมกับคำถามว่าต้องการเปลี่ยนเป็นโหมด STOP หรือไม่ ถ้าต้องการเปลี่ยนให้ตอบ Yes ถ้าไม่ต้องการเปลี่ยนให้ตอบ No (ในที่นี้ให้ตอบ Yes)

No.	STOP	
	Place the PLC in STOP mode?	
	Yes No	

28. เมื่อเปลี่ยนโหมดกลับไปที่โหมด STOP แล้ว สามารถแก้ไขเปลี่ยนแปลงโปรแกรมได้ตาม ต้องการและเมื่อแก้ไขโปรแกรมเสร็จสมบูรณ์ให้ เริ่มตามขั้นตอนเดิมคือ การ Compile All, Download, RUN และ Program Status ตามลำดับต่อไป การต่ออุปกรณ์อินพุตของเครื่องโปรแกรมเมเบิลคอนโทรลเลอร์

ในการต่ออุปกรณ์อินพุตของเครื่องโปรแกรมเมเบิลคอนโทรลเลอร์รุ่น S7–200 จะมีลักษณะการต่อ สายของวงจรอินพุตไฟตรงดังนี้คือ 1. การต่อวงจรภาคอินพุตไฟตรงแบบ Sink หลักการต่อวงจรอินพุตแบบ Sink Type ทำได้โดยการ ต่อขั้ว 0 V จากเครื่อง PLC หรือ จากแหล่งจ่ายภายนอกต่อเข้าที่ขั้ว 1M (Common Input) ของ PLC และต่อ ขั้วไฟบวก 24 V จากเครื่อง PLC หรือจากแหล่งจ่ายภายนอก เข้ากับขั้วด้านหนึ่งของอุปกรณ์และอีกขั้วของ อุปกรณ์ต่อเข้ากับขั้วอินพุตของ PLC

การต่อวงจรอินพุตไฟตรงแบบ Source หลักการต่อวงจรอินพุตแบบ Source Type ทำได้โดยการ ต่อขั้วไฟบวก 24 V จากเครื่อง
 PLC หรือจากแหล่งจ่ายภายนอกต่อเข้าที่ขั้ว 1M (Common Input) ของ PLC และต่อขั้ว 0 V จากเครื่อง PLC หรือจากแหล่งจ่ายภายนอก
 เข้ากับขั้วด้านหนึ่งของอุปกรณ์และอีกขั้ว ของอุปกรณ์อินพุตต่อเข้ากับขั้วอินพุตของ PLC

![](_page_88_Picture_3.jpeg)

![](_page_89_Picture_0.jpeg)

# การเขียนโปรแกรมบน

# STEP 7-Micro/WIN

![](_page_91_Picture_0.jpeg)

![](_page_92_Picture_0.jpeg)

การกำหนดชื่อในตารางสัญลักษณ์ (Symbol Table) เป็นการกำหนดชื่อหรือข้อความลงไปใน โปรแกรมที่เขียน เพื่อให้สะดวกในการ จดจำหน้าที่การทำงานของอินพุตเอาต์พุตและตำแหน่งต่าง ๆ ใน โปรแกรมซึ่งเมื่อเขียนโปรแกรมเสร็จ หากต้องการกำหนดชื่อหรือข้อความ สามารถกำหนดได้ในตาราง สัญลักษณ์ (Symbol Table) โดย

1. คลิก Symbol Table ใน Navigation Bar หรือสามารถดับเบิ้ลคลิกที่ Symbol Table ใน Instruction Tree แล้วดับเบิ้ลคลิกที่

User1

![](_page_92_Picture_4.jpeg)

2. การกำหนดรายละเอียดต่างๆในการกำหนดค่า ประกอบด้วยส่วนต่างๆ ดังนี้ (1) Symbol ให้กำหนดชื่อหรือหน้าที่ของอินพุตเอาต์ พุตตำแหน่งที่ต้องการ เช่น Start, Stop, มอเตอร์ 1 (Motor1), มอเตอร์ 2 (Motor2) เป็นต้น

(1) Symbol ให้กำหนดชื่อหรือหน้าที่ของอินพุตเอาต์พุตตำแหน่งที่ต้องการ เช่น Start, Stop, มอเตอร์ 1 (Motor1), มอเตอร์ 2 (Motor2) เป็นต้น

(2) Address ให้กำหนดตำแหน่ง Address อินพุต เอาต์พุตของ PLC เช่น 10.0, Q0.0, T0, C10, VB0, VW10, VD100 เป็นต้น

(3) Comment ให้กำหนดรายละเอียดคำอธิบายของอุปกรณ์เช่น มอเตอร์ (Motor), ปั๊ม (Pump), กระบอกสูบ (Cylinder) เป็นต้น

留 🔺 🌤 🌤 🎘 🛣 📗	]7	· → た	→  ++ -(> -[]			
Project1			· 2 · · · I · ·	. 3 4		· · 6
CPU 224 PEL 02.00		09	Symbol	Address	Com	ment
Drog zz4 HEL 02.00	1		Start	10.0	Green Switch	
Frogram block	2		Stops	10.1	Red Switch	- 1
Symbol Table	3		Motor1	Q0.0	Motor1	
POLL Sumbole	4					
Status Chart	5					

3. เมื่อกำหนดรายละเอียดใน Symbol Table เสร็จเรียบร้อยคลิกที่คำสั่ง Compile All และกลับไป ยังหน้าต่าง Program Block โดยเลื่อนเมาส์ไปซี้ที่คำสั่ง Program Block ใน Navigation Bar หรือดับเบิ้ล คลิกที่คำสั่ง Program Block ใน Instruction Tree จากนั้น ดับเบิ้ลคลิกที่ Main Program จะได้ลักษณะของ โปรแกรมที่ได้กำหนดชื่อ ตำแหน่งและรายละเอียดต่าง ๆ ขึ้นมาใหม่ เพื่อให้ผู้ใช้สามารถ จดจำได้ง่ายขึ้น

![](_page_93_Figure_3.jpeg)

การดูสภาวะการทำงานใน Status Chart

การดูสภาวะการทำงานใน Status Chart เป็นการดูและตรวจสอบค่าพารามิเตอร์ต่าง ๆ ในขณะที่ โปรแกรมกำลังทำงาน เช่น การดู สภาวะการทำงานของเอาต์พุตค่าของเวลา (Timer) ค่าจำนวนนับของ ตัวนับ (Counter) เป็นต้น สามารถทำได้ดังนี้

1. คลิกที่ Status Chart ใน Navigation Bar หรือสามารถดับเบิ้ลคลิกที่ Status Chart ใน Instruction Tree จากนั้นดับเบิ้ลคลิกที่

User1

![](_page_94_Figure_4.jpeg)

2. จะปรากฏหน้าต่าง Status Chart ขึ้นมา

+	± ← → [++ ↔			
	· · · · · 2 · · · I	3	4 5 .	6 7 .
	Address	Format	Current Value	New Value
1		Signed		
2		Signed		

3. รายละเอียดต่างๆในการกำหนดค่า ประกอบด้วยส่วนต่างๆ ดังนี้

3.1 ตำแหน่ง (Address) ให้กำหนดตำแหน่งต่าง ๆ ที่ต้องการดูสถานะเช่น 10.0, Q0.0, T0, C10 เป็นต้น

3.2 Format เป็นการกำหนดรูปแบบของข้อมูล เช่น Bit (บิต), Signed, Unsigned

3.3 Current Value เป็นการแสดงค่าข้อมูลหรือสถานะการทำงานที่กำหนดแบบ Real Time เช่น การ On-Off ของ Contact ค่าเวลาของ Timer หรือค่า V Memory ของ PLC

3.4 New Value เป็นการกำหนดค่าข้อมูล หรือสถานะการทำงานของตำแหน่งที่กำหนด เช่น การ สั่งการ On-Off ของคอนแทค ค่า Set Point ให้กับค่าการนับของ Counter และการป้อนค่าใหม่ให้กับ V Memory

4. ทดลองกำหนดรายละเอียด เพื่อดูสภาวะการทำงาน ดังแสดงในรูป

	Address	Format	Current Value	New Value
1	Start:10.0	Bit		
2	Stops:10.1	Bit		
3	Motor1:Q0.0	Bit		
4		Signed		

5. เมื่อกำหนดรายละเอียดเสร็จ สามารถดูขั้นตอนการทำงานของ Chart Status ได้โดยคลิกที่เมนู Debug ใน Menu Bar หรือคลิกที่คำสั่ง

Start Chart Status ในแถบเครื่องมือ

![](_page_96_Picture_2.jpeg)

6. จะปรากฏสถานะของข้อมูลส่วน Current Value ขึ้นมา

-	Address Form		Current Value	New Value	
1	Start:10.0	Bit	2#0		
2	Stops:10.1	Bit	2#0		
3	Motor1:Q0.0	Bit	2#0		
4		Signed			

7. หากต้องการยกเลิก Chart Status คลิกที่คำสั่ง Debug ใน Menu Bar หรือคลิกที่คำสั่ง Stop Chart Status ในแถบเครื่องมืออีกครั้ง

PLC	Deb	ug Tools Windows Help	_			
A Pro		First Scan Multiple Scans				
	<ul> <li>Image: A start of the start of</li></ul>	Start Program Status Use Execution Status Pause Program Status	· · · · · · · · · · · · · · · · · · ·			
			322	Bit	Currer	
•		Start Chart Status		Bit	-	
	_	Pause Trend Chart		Signed		

8. การกลับไปยังหน้าต่างของโปรแกรมหลักคลิกที่ Program Block ใน Navigation Bar หรือดับเบิ้ล คลิกที่ Program Block ใน

Instruction Tree แล้วดับเบิ้ลคลิกที่ MAIN (OB1)

![](_page_97_Picture_4.jpeg)

![](_page_98_Picture_0.jpeg)

การ Upload Project คือ การนำโปรแกรมที่มีอยู่ใน PLC ขึ้นมายังคอมพิวเตอร์เพื่อใช้งานหรือ กระทำการใด ๆ สามารถทำได้ตามขั้นตอนใน การ Upload Project ดังนี้

1. คลิกที่เมนู File เลื่อนลงมา คลิกที่ (1) คำสั่ง Upload (อัปโหลด) หรือสามารถคลิกที่ (2) คำสั่ง Upload ที่ปรากฏในแถบเครื่องมือบน หน้าจอโปรแกรม

1	New	Ctrl+N	1 -
	Open Close	Ctrl+0	 
	Save Save As	Ctrl+S	
P	Set Password		
¢	Import Export		
	Upload	Ctrl+U	Net
	Download	Ctrl+D	

### 2. จะปรากฏหน้าต่าง Upload ขึ้นมา คลิก Upload

![](_page_99_Picture_1.jpeg)

3. รอจนกระทั่งการ Upload สมบูรณ์เมื่อ Upload เสร็จสมบูรณ์จะปรากฏโปรแกรมที่อยู่ใน PLC ขึ้นมาที่หน้าจอคอมพิวเตอร์

CPU 224 REL 02:00
₽_1 _
Cancel
1

![](_page_100_Picture_0.jpeg)

ก่อนการแทรกโปรแกรม ให้สังเกตจะมีคำว่า INS (Insert หมายถึง การเขียนโปรแกรม แบบเขียนแทรก) และเมื่อเรากดปุ่ม Insert บนคีย์บอร์ดของคอมพิวเตอร์ตัวอักษรจะเปลี่ยนจาก INS เป็น คำว่า OVR (Over หมายถึง การเขียนโปรแกรมแบบเขียนทับ)

![](_page_100_Figure_2.jpeg)

การแทรกโปรแกรมสามารถทำได้ดังนี้

1.ในโปรแกรมแสดงข้อความ INS ผู้ใช้สามารถแทรกโปรแกรมที่ต้องการ เพิ่มเติมลงไปได้โดยเลื่อนเมาส์ไปคลิกยังตำแหน่งที่ต้องการแทรก

![](_page_100_Figure_5.jpeg)

### 2. ดับเบิ้ลคลิกที่คำสั่งที่ต้องการแทรกเข้ามาในโปรแกรม จะปรากฏคำสั่งนั้น เข้ามา ในโปรแกรมทันที

![](_page_101_Figure_1.jpeg)

 หากรูปแบบการเขียนโปรแกรมอยู่ในโหมด OVR ซึ่งหมายถึงการเขียน โปรแกรมแบบเขียนทับของเดิม เมื่อดับเบิ้ลคลิกที่คำสั่งที่ต้องการ แทรก คำสั่งนั้นจะเขียนทับคำสั่งเก่าทันที

Network 3			
—≯			
Network 4	(	۲	
		Network1 Row 2, Col 2 OVR	)

![](_page_101_Picture_4.jpeg)

4. คลิกที่คำสั่งใหม่ คำสั่งใหม่จะเขียนทับคำสั่งเก่าทันที

![](_page_102_Figure_1.jpeg)

การแทรก Network (Insert Network) สามารถทำได้ตามขั้นตอนดังนี้

1. หากต้องการแทรก Network ใหม่ลงไปใน Network 2 ให้เลื่อนเมาส์ไปชี้ที่ Network 2 จากนั้นคลิกขวาเลื่อนลงมาที่กลุ่มคำสั่ง Insert แล้วเลื่อนไปยังกลุ่มย่อย Network(s) คลิกเมาส์ซ้าย หรือกด F3

![](_page_102_Figure_4.jpeg)

เลื่อนเมาส์ไปชี้ที่ Network 2 จากนั้นคลิก ขวาเลื่อนลงมา ที่กลุ่มคำสั่ง Insert แล้ว เลื่อนไปยังกลุ่มย่อย Network(s) คลิก เมาส์ซ้ายหรือ กด F3 2. โปรแกรมใน Network ที่ 2 จะถูกเลื่อนลงมายัง Network 3 และให้ผู้ใช้สามารถออกแบบ โปรแกรมที่ต้องการแทรกลงไปได้

![](_page_103_Figure_1.jpeg)

![](_page_104_Picture_0.jpeg)

การลบ (Delete) โปรแกรมที่เขียนผิด มีขั้นตอนในการลบตำแหน่งที่เขียนผิดดังนี้

การลบ Column (Delete Column)

ให้เลื่อนเมาส์ไปชี้ที่ตำแหน่งต้องการลบคลิกเมาส์ซ้ายให้กรอบสี่เหลี่ยมล้อมตำแหน่งที่เขียน ผิด จากนั้นให้กดปุ่ม Delete ที่คีย์บอร์ด คำสั่งที่ต้องการลบจะถูกลบออกไปทันทีและสามารถที่จะแก้ไขให้ โปรแกรมถูกต้องต่อไปได้

![](_page_104_Figure_4.jpeg)

การลบ Network (Delete Network)

ให้เลื่อนเมาส์ไปชี้ที่ตำแหน่งต้องการลบคลิกเมาส์ซ้ายให้กรอบสี่เหลี่ยมล้อมตำแหน่งที่เขียน ผิด จากนั้นให้กดปุ่ม Delete ที่คีย์บอร์ด คำสั่งที่ต้องการลบจะถูกลบออกไปทันทีและสามารถที่จะแก้ไขให้ โปรแกรมถูกต้องต่อไปได้

![](_page_104_Figure_7.jpeg)

![](_page_105_Picture_0.jpeg)

การคัดลอกและวาง (Copy And Paste) กรณีโปรแกรมที่ต้องการเขียนในแต่ละ Network มีลักษณะ คล้ายคลึงกัน การคัดลอก โปรแกรมเป็นวิธีการที่ช่วยให้การเขียนโปรแกรมมีความรวดเร็วมากยิ่งขึ้น ซึ่งมี ขั้นตอนดังนี้

1. เลื่อนเมาส์ไปที่ Network ที่ต้องการคัดลอก แล้วดับเบิ้ลคลิกเมาส์ซ้ายให้ขึ้นกรอบสีน้ำเงินที่ Network ที่ต้องการคัดลอก จากนั้น

คลิกเมาส์ขวา 1 ครั้ง เลื่อนแถบสีน้ำเงินลงมาที่คำว่า Copy หรือกดคีย์ ลัด Ctrl + C

![](_page_105_Picture_4.jpeg)

![](_page_106_Figure_0.jpeg)

4. โปรแกรมที่คัดลอกจะถูกวางลงใน Network ที่ต้องการ ผู้ใช้สามารถแก้ไขตำแหน่งอินพุตเอาต์พุต ให้ถูกต้องตามโปรแกรมต่อไป

![](_page_107_Figure_1.jpeg)

ในการใช้งาน PLC ผู้ใช้อาจทำการ Download (ดาวน์โหลด) โปรแกรมลงใน PLC ติดต่อกันหลาย ครั้ง จนทำให้ PLC เกิดทำงาน ผิดพลาดได้ผู้ใช้สามารถป้องกันปัญหาที่อาจเกิดขึ้นนี้โดยทำการ Clear โปรแกรมที่อยู่ใน PLC เดิมก่อน จากนั้นถึงจะทำการดาวน์โหลดซึ่งมี ขั้นตอนในการ Clear ดังนี้
1. คลิกที่ PLC ใน Menu Bar และเลื่อนลงมาที่คำสั่ง Clear แล้วคลิกเมาส์ซ้าย



2. จะปรากฏหน้าต่างคำสั่ง Clear คลิกที่ปุ่มคำสั่ง Clear โปรแกรมที่อยู่ใน PLC จะถูกเคลียร์ทั้งหมด พร้อมทั้งโหมดการทำงานของ PLC จะ

กลับไปอยู่ที่โหมด STOP



การสั่งงานผ่าน Program Editor (Force)

การสั่งงานผ่าน Program Editor (Force) เป็นคำสั่งบังคับการทำงานของตำแหน่งที่ต้องการ ให้มี สถานะเป็น On- Off โดยการใช้งานคำสั่ง

(Force) PLC ต้องอยู่ในสภาวะ Run Mode เท่านั้น มีขั้นตอนใน การสั่งงานดังนี้

1. เมื่อเขียนโปรแกรมเสร็จเรียบร้อย ทำการ Compile, Download, Run Mode และสั่ง Status Program

2. การสั่งงานผ่าน Program Editor (Force) เลื่อนเมาส์ไปซี้คำสั่งที่ต้องการสั่งงาน คลิกเมาส์ขวา เลื่อนลงมาที่คำสั่ง Force หรือเลื่อนเมาส์

ไปชี้ที่คำสั่ง Force บนแถบเครื่องมือ

2↓ 2† 1	b 🖪 🖉 ₩ 🖾 🖾 🖾 🕼 🖓 📕 🗸 📙
· ← →   + ⊦	-0 1
	2 · · · I · · · 3 · · · I · · · 4 · · · I · · · 5 ·
	ymbol VarType DataType
Network 1	Network Title
10.0=0FF	Options
Q0.0=OF	Write
	Force
	Unforce
	Pause Program Status 🔫

3. จะปรากฏหน้าต่างคำสั่ง Force ขึ้นมา พร้อมบอกตำแหน่งที่จะ Force รวมถึงสถานะของ ตำแหน่งที่จะสั่งงานให้ On หรือ Off หากตำ

แหน่งและสถานะถูกต้องให้คลิก Force

orce			×
Ad	ldress: 10.0		
١	Value: 🚺		
		 <b>*</b>	

4. สังเกตตำแหน่งที่ถูกสั่งงาน (Force) จะมีสภาว<mark>ะเป็น On และมีสัญลักษณ์รูปกุญแจบอกส</mark>ภาวะ การถูกควบคุมของตำแหน่งนั้น ๆ



5. การยกเลิกการสั่งงาน (Unforce) เลื่อนเมาส์ไปชี้ที่ตำแหน่งที่ถูกควบคุม คลิกเมาส์ขวา เลื่อนลง มาที่คำสั่ง Unforce หรือเลื่อนเมาส์ไปชี้ที่

ตำแหน่งที่ถูกควบคุม แล้วเลื่อนเมาส์ไปคลิกที่คำสั่ง Unforce บน แถบเครื่องมือ

2		
×/	Symbol Vari ype Data i ype	
	Network 1 Network Title	
	Options	1.000
	Write	
	Q0.0=0N Force	
	Unforce 4	0
	Pause Program Status	1 2

5. หากต้องการยกเลิกการสั่งงานทั้งหมดทุกจุดที่เคยสั่งงานไว้เลื่อนเมาส์ไปคลิกที่คำสั่ง Unforce All บนแถบเครื่องมือ



้การต่ออุปกรณ์เอาต์พุตของเครื่องโปรแกรมเมเบิลคอนโทรลเลอร์

ในการต่ออุปกรณ์เอาต์พุตของเครื่องโปรแกรมเมเบิลคอนโทรลเลอร์รุ่น S7–200 จะมีลักษณะการต่อ สายของวงจรเอาต์พุต ตามวัสดุ อุปกรณ์ที่นำมาทำเป็นเอาต์พุตดังนี้คือ

 การต่อวงจรเอาต์พุตชนิดรีเลย์หลักการต่อวงจรเอาต์พุตชนิดรีเลย์กรณีอุปกรณ์เอาต์พุตใช้กับ ไฟตรงทำได้โดยการต่อขั้วไฟบวก 24
 V จากเครื่อง PLC หรือจากแหล่งจ่ายภายนอกต่อเข้าที่ขั้วเอาต์พุต1L (Common Output) ของ PLC และต่อขั้ว 0 V จากเครื่อง PLC หรือ จากแหล่งจ่ายภายนอกเข้ากับขั้วด้าน หนึ่งของอุปกรณ์และอีกขั้วของอุปกรณ์ที่ต่อเข้ากับขั้วเอาต์พุตของ PLC ที่ใช้งาน หากในกรณีที่อุปกรณ์ เอาต์พุตใช้กับไฟสลับ ทำได้โดยการนำขั้ว L จากแหล่งจ่ายภายนอกต่อเข้าที่ขั้วเอาต์พุต 1L (Common Output) ของ PLC และต่อขั้ว N จากแหล่งจ่ายภายนอกเข้ากับขั้วด้านหนึ่งของอุปกรณ์และอีกขั้วของ อุปกรณ์ต่อเข้ากับขั้วเอาต์พุตของ PLC ที่ใช้งาน

 การต่อสายของวงจรภาคเอาต์พุตชนิดทรานซิสเตอร์ 24 DCV การต่อวงจร ทำได้โดยการต่อขั้วไฟ บวก 24 V จากเครื่อง PLC หรือ จากแหล่งจ่ายภายนอกต่อเข้าที่ขั้วเอาต์พุต 1L+ ของ PLC และต่อขั้ว 0 V จากเครื่อง PLC หรือจากแหล่งจ่ายภายนอกเข้าที่ขั้วเอาต์พุต 1M ของ PLC และต่อเข้ากับขั้วด้านหนึ่งของ อุปกรณ์และอีกขั้วของอุปกรณ์เอาต์พุตต่อเข้ากับขั้วเอาต์พุตของ PLC ที่ใช้งาน



# กลุ่มคำสั่ง Bit Logic





คำสั่ง Normally Open (NO) หรือคำสั่งหน้าสัมผัสปกติเปิด โดยหน้าสัมผัสจะเปลี่ยนสภาวะจาก ปกติเปิดเป็นสภาวะปิด เมื่อมีการ เปลี่ยนสัญญาณอินพุตจาก 0 เป็น 1



คำสั่ง Normally Closed

# คำสั่ง Output (เอาต์พุต)

คำสั่งเอาต์พุต (Output) เป็นคำสั่งการแสดงผลของเอาต์พุต

การใช้คำสั่งหน้าสัมผัสปกติเปิด (Normally Open : NO) คำสั่งหน้าสัมผัสปกติปิด (Normally Closed : NC) และคำสั่งเอาต์พุต

bit

ตัวอย่าง การทำงานของโปรแกรม เป็นการใช้คำสั่งหน้าสัมผัสปกติเปิดและคำสั่งหน้าสัมผัส ปกติปิดควบคุมการทำงานของเอาต์พุตโดยมีการทำงานของ โปรแกรม คือเมื่ออินพุต I0.0 ทำงาน จะทำให้คำสั่งเอาต์พุต Q0.0 และ Q0.1 ทำงานและเมื่ออินพุต I0.1 ทำงานจะทำให้ คำสั่งเอาต์พุต Q0.0 และ Q0.1 หยุดทำงาน



## คำสั่ง Set (S) และคำสั่ง Reset (R)

คำสั่ง Set เป็นคำสั่งให้ตำแหน่งบิต (bit) ที่ถูกระบุทำงาน (Lock) ดังรูปที่ 6.5 (ก) และคำสั่ง Reset เป็นคำสั่งให้ตำแหน่งบิตที่ถูกระบุ หยุดทำงาน (Unlock) ดังรูปที่ 6.5 (ข) ตามรายละเอียดจำนวนของจุด (N) โดยเริ่มต้นตามรายละเอียดของตำแหน่งบิตผู้ใช้สามารถที่จะสั่ง Set และ Reset ตามรายละเอียดจำนวน ของจุดได้คือค่า 1 – 255 ถ้าใช้คำสั่ง Reset กับ Timer หรือเคาน์เตอร์ (Counter) จะมีผลทำให้ค่ าเวลาของ Timer และค่าการนับของเคาน์เตอร์ถูกเคลียร์ไปด้วย

bit

 ท
 ท
 ท

 คำสั่ง Set (S)
 คำสั่ง Reset (R)

 ตัวอย่าง การใช้คำสั่ง Set (S) และการใช้คำสั่ง Reset (R)

 การทำงานของโปรแกรมเป็นการใช้คำสั่ง Set และ Reset เพื่อควบคุมการทำงานของเอาต์พุต

 การทำงานคือเมื่ออินพุต 10.0 ทำงาน จะทำให้คำสั่ง Set ทำงาน ส่งผลให้เอาต์พุตตั้งแต่ตำแหน่ง

 Q0.0 ทำงานและทำงานไปจนถึงตำแหน่งของ N ในที่นี้คือ 4 จุดหรือ 4 บิต ฉะนั้นจะเห็นเอาต์พุต

 ทำงานตั้งแต่ตำแหน่ง Q0.0 จนถึง Q0.3 และเมื่ออินพุต 10.1 ทำงาน จะทำให้คำสั่ง Reset ทำงาน

 ส่งผลให้เอาต์พุตตั้งแต่ตำแหน่งของเอาต์พุต Q0.0 –Q0.3 หยุดทำงาน

bit



#### วิธีการเขียนโปรแกรม

1. เปิดโปรแกรม STEP 7-Micro/WIN แล้ว Communication เพื่อติดต่อสื่อสารข้อมูลระหว่าง คอมพิวเตอร์กับ PLC จากนั้น ดับเบิ้ลคลิกที่

Bit Logic ใน Instructions จะปรากฏคำสั่งหน้าสัมผัสที่ใช้ใน การเขียนโปรแกรม

2. เขียนโปรแกรมตามที่กำหนดจนถึงคำสั่ง Set (S) ดับเบิ้ลคลิกที่คำสั่ง Set





3. กำหนดรายละเอียดในการใช้งานคำสั่ง Set โดยด้านบนใส่เอาต์พุตบิตเริ่มต้น คือ Q0.0 และ ด้านล่างใส่จำนวนจุด (N) ที่ต้องการให้ทำงาน

คือ 4 เขียนโปรแกรมต่อไปจนครบตามที่กำหนด แล้วทดลองการทำงานของโปรแกรมตามขั้นตอน



้คำสั่ง Positive Transition (P) และ คำสั่ง Negative Transition (N)

คำสั่ง Positive Transition เป็นคำสั่งที่มีผลต่อตำแหน่งที่ต่อร่วมกับคำสั่ง P ด้วยระยะเวลาเพียง 1 Scantime เมื่อได้รับสัญญาณสภาวะ On หรือช่วงขอบขาขึ้นจากคอนแทคที่ควบคุมคำสั่ง P



คำสั่ง Positive Transition เป็นคำสั่งที่มีผลต่อตำแหน่งที่ต่อร่วมกับคำสั่ง P ด้วยระยะเวลาเพียง 1 Scantime เมื่อได้รับสัญญาณสภาวะ On หรือช่วงขอบขาขึ้นจากคอนแทคที่ควบคุมคำสั่ง P



ตัวอย่าง การใช้คำสั่ง Positive Transition และคำสั่ง Negative Transition การทำงานของโปรแกรม เมื่ออินพุต I0.0 ทำงานหรือมีสภาวะ On (ช่วงขอบ ขาขึ้น) จะทำให้คำสั่ง Positive Transition ทำงาน ส่งผลให้เอาต์พุต Q0.0 ทำงานเป็นระยะเวลา 1 Scantime และเมื่ออินพุต I0.0 หยุดทำงานหรือมี สภาวะ Off (ช่วงขอบขาลง) จะทำให้เอาต์พุต Q0.1 ทำงานเป็นระยะเวลา 1 Scantime



#### วิธีการเขียนโปรแกรม

 เปิดโปรแกรม STEP 7-Micro/WIN ทำการ Communication เพื่อติดต่อสื่อสารข้อมูลระหว่าง คอมพิวเตอร์กับ PLC จากนั้นดับเบิ้ลคลิก ที่ Bit Logic ใน Instructions จะปรากฏคำสั่งหน้าสัมผัสที่ใช้ใน การเขียนโปรแกรม 2. เขียนโปรแกรมตามโปรแกรมตัวอย่างจนถึงคำสั่ง Positive Transition ดับเบิ้ลคลิกที่คำสั่ง Positive Transition



เขียนโปรแกรมจนครบตามโปรแกรมตัวอย่าง จากนั้นทดลองการทำงานของ โปรแกรมตามขั้นตอน



ตามที่กล่าวมาแล้วคำสั่ง Positive และ Negative Transition มีผลเพียง Scantime เดียว โดยคำสั่ง Positive Transition จะมีผลช่ วงขอบขาขึ้นและ Negative Transition จะมีผลช่วงขอบขาลงของอินพุตที่ ควบคุมคำสั่ง Positive และ Negative Transition



# คำสั่ง Jump to Label

การใช้คำสั่ง Jump to Label เป็นคำสั่งที่ขึ้นต้นด้วยคำสั่ง JMP และต้องลงท้ายด้วยคำสั่ง LBL เสมอ การระบุจำนวนสามารถระบุจุด ได้สูงสุดถึง 256 จุด (0-255 จุด) การเขียนโปรแกรมคำสั่ง Jump และ Label จะต้องอยู่ในหน้าเดียวกัน โดยมีเงื่อนไขคือเมื่อ คำสั่ง Jump ทำงาน โปรแกรมซึ่งอยู่ระหว่างคำสั่ง Jump และ Label จะคงค้างสถานะ(เปรียบเหมือน CPU ของ PLC ไม่ประมวลผลโปรแกรมที่อยู่ระหว่ างคำสั่ง Jump กับ Label) ถ้าเอาต์พุตทำงานก่อนคำสั่ง Jump จะทำงานเอาต์พุตนั้น ๆ จะคงทำงานค้างตลอด เปลี่ยนแปลงการทำงานไม่ ได้จนกว่าคำสั่ง Jump จะหยุดทำงาน หากเมื่อคำสั่ง Jump หยุดทำงานทุก เงื่อนไขจะกลับสู่สภาวะปกติ





การทำงานของโปรแกรม

เมื่อคำสั่ง Jump ยังไม่ทำงาน (I0.0 มีสภาวะ Off) โปรแกรม ทุก Network จะยังทำงานตามปกติแต่เมื่อคำสั่ง Jump ทำงาน (I0.0 มีสภาวะ On และคงสภาวะ On ไว้) จะทำให้ โปรแกรมซึ่งอยู่ระหว่างคำสั่ง Jump และ Label (โปรแกรมใน Network ที่ 2) จะค้างสภาวะการทำงานเดิมก่อนหน้า (CPU ไม่ประมวลผลการทำงาน) และเมื่อคำสั่ง Jump หยุดทำงาน (I0.0 มีสภาวะ Off) โปรแกรมทุก Network จะกลับมาทำงาน ตามปกติ



วิธีออกแบบโปรแกรม

1. ดับเบิ้ลคลิกที่กลุ่มคำสั่ง Bit Logic ใน Instructions จะปรากฏคำสั่งหน้าสัมผัสที่ใช้เขียน โปรแกรม

2. เขียนโปรแกรมตามโปรแกรมตัวอย่างที่ 6.4 จนถึงคำสั่ง Jump ดับเบิ้ลคลิกที่กลุ่มคำสั่งProgram Control จากนั้นคลิกเลือกคำสั่ง Jump



4. เขียนโปรแกรมต่อไปจนถึงคำสั่ง Label



5. เลือกคำสั่ง Label กำหนดค่าด้านบน ???? คำสั่ง Label เป็นค่า 0 แล้วเขียนโปรแกรมต่อจนครบ ตามโปรแกรมตัวอย่างที่กำหนด



6. ทดลองการทำงานของโปรแกรมตามขั้นตอน ดังรูปที่ 6.22 การทำงานของโปรแกรมจะทำงานเป็น ปกติคือ On–Off 10.1 Output Q0.0 ทำงาน On–Off 10.2 Output Q0.0 หยุดทำงาน กรณีที่ 10.0 ยังไม่ On (คำสั่ง Jump ยังไม่ทำงาน)



7. เมื่อ On Input 10.0 ดังรูปที่ 6.23 เมื่อ On10.0 (คำสั่ง Jump ทำงาน) จะทำให้โปรแกรมซึ่งอยู่ ระหว่างคำสั่ง JMP และ LBL ค้างสภาวะ การทำงานเดิมก่อนหน้าโดยจะไม่ถูกประมวลผลการทำงานจาก CPU และเมื่อคำสั่ง Jump หยุดทำงานอีกครั้ง (10.0 Off) การทำงานของ โปรแกรมทุก Network จะกลับ ทำงานตามปกติ





# กลุ่มคำสั่ง Timers





หลักการทำงานของคำสั่ง On-Delay Timer (TON) คือเมื่อมีสัญญาณ On เข้าที่ขา IN คำสั่ง TON จะเริ่มนับเวลาเมื่อค่าเวลาที่นับ มีค่าเท่ากับค่าเวลาที่กำหนดไว้ใน P Reset Time (PT) คอนแทคช่วยของ TON จะทำงานและเมื่อหยุดจ่ายสัญญาณเข้าที่ขา IN ค่าเวลาของ TON ที่กำลังนับจะถูก Reset เวลากลับ เป็น 0 และคอนแทคช่วยของ TON จะกลับสู่สภาวะเดิม



คุณลักษณะของคำสั่ง On-Delay Timer (TON)

ชนิด	ค่าความละเอียด	ค่าสูงสุดในการตั้งเวลา	หมายเลขตัวตั้งเวลา
(Timer Type)	(Resolution)	(Maximum Value)	(Timer Number)
TON (non-retentive)	1 ms	32.767 s	T32, T96
	10 ms	327.67 s	T33-T36 , T97-T100
	100 ms	3,276.7 s	T37-T63 , T101-T255

การทำงานของโปรแกรม

เมื่อ On อินพุต I0.0 ส่งผลให้เอาต์พุต Q0.0 ทำงาน (On) คอนแทคช่วยของเอาต์พุต Q0.0 เปลี่ยน สภาวะจากปกติเปิดเป็นสภาวะปิ ดมีสัญญาณเข้าที่ ขา IN ของ TON (T32) และ TON จะเริ่มนับเวลาและ เมื่อค่าเวลาของ TON มีค่าเท่ากับ 3 วินาทีคอน แทคช่วยของ TON จะทำงานเปลี่ยนสภาวะจากปกติ เปิดเป็นสภาวะปิดส่งผลให้เอาต์พุต Q0.1 ทำงาน

และเมื่ออินพุต I0.1 มีสภาวะ On ส่งผลให้ เอาต์พุต Q0.0 หยุดทำงาน (Off) ไม่มีสัญญาณเข้าที่ ขา IN ของ TON (T32) TON จะ หยุดทำงาน ค่าเวลา ของ TON จะถูก Reset กลับเป็น 0 คอนแทคช่วยของ TON จะกลับสู่สภาวะเดิมเป็นปกติเปิดใน Network ที่ 3 ส่งผล ให้เอาต์พุต Q0.1 หยุดทำงาน



วิธีเขียนโปรแกรม

- 1. เปิดโปรแกรม STEP 7-Micro/WIN ทำการ Communication เพื่อติดต่อสื่อสารระหว่างคอมพิวเตอร์ กับ PLC
- 2. เขียนโปรแกรมตาม Ladder Diagram จนถึงคำสั่ง Timer ดับเบิ้ลคลิกเมาส์ซ้ายที่ กลุ่มคำสั่ง Timer จะปรากฏคำสั่ง Timer ทั้ง 5 ชนิด



3. เลื่อนเมาส์ไปที่คำสั่ง TON คลิกเมาส์ซ้าย คำสั่ง TON จะปรากฏเข้ามาใน Main Program



4. กำหนดค่าหมายเลขตัวตั้งเวลาของ On-Delay Timer (TON) และค่าเวลาของ TON (PT) โดย กำหนดค่าตามตัวอย่างและเขียน โปรแกรมต่อไปจนครบเมื่อเขียนโปรแกรมครบให้ตรวจสอบความถูกต้อง ของโปรแกรมสังเกตผลการตรวจสอบในช่อง Output Windows ว่ามี Error หรือไม่ถ้ามีให้กลับไปแก้ไข โปรแกรมให้ถูกต้องถ้าไม่มีให้ทำตามขั้นตอนต่อไปคือการดาวน์โหลด (Download) จากนั้นทดลองการ ทำงานของโปรแกรม (RUN)

5. หากต้องการดูสภาวะการทำงานของโปรแกรม ทำได้โดยการเลื่อนเมาส์ไปที่คำสั่ง Program Status บน Tool Bars คลิกเมาส์ซ้าย 1 ครั้ง โปรแกรมจะแสดง สภาวะการทำงานเป็นเส้นสีน้ำเงิน ทำให้มองเห็นการ ทำงานของโปรแกรมชัดเจนยิ่งขึ้น



6. ทดลองการทำงานโปรแกรมโดยโยกสวิตช์I0.0 (On-Off) ส่งผลให้เอาต์พุต
 Q0.0 ทำงาน TON จะเริ่มนับ เวลา เมื่อครบตามค่าที่ตั้งไว้คอนแทคช่วยของ
 TON จะ เปลี่ยนสภาวะทำให้เอาต์พุต Q0.1 ทำงาน(ค่าเวลาของ TON จะยัง
 นับต่อไปเรื่อย ๆ จนกว่าจะมีการตัดสัญญาณ ออกจาก TON)

7. เมื่อโยกสวิต ซ์ I0.1 (On-Off) ส่งผลให้ เอาต์พุต Q0.0 หยุดทำงานเป็นการ ตัดสัญญาณออก จาก TON ทำให้ค่าเวลาของ TON จะ Reset กลับไป เป็น
"0" คอนแทคช่วยของ TON กลับสู่สภาวะเดิม คือปกติเปิดทำให้เอาต์พุต
Q0.1 ห ยุดทำงาน



8. กรณีโปรแกรมที่ทำการทดลองถูกต้องหรือ อาจผิดพลาด และต้องการแก้ไข

โปรแกรม สามารถทำ ได้ดังนี้คือยกเลิกการทำงานของ Program Status ก่อน

โดยเลื่อนเมาส์ไปชี้ที่คำสั่ง Program Status คลิกเมาส์ซ้าย 1 ครั้ง แล้วหน้าจอ

จะกลับ สู่ปกติ

 ขั้นตอนต่อไปคือ การเปลี่ยนโหมดกลับไปเป็น โหมด STOP โดยการเลื่อน เมาส์ไปชี้ที่คำสั่ง STOP คลิก เมาส์ซ้าย 1





10. จะปรากฏหน้าต่าง STOP พร้อมกับคำถาม ว่าต้องการเปลี่ยนเป็นโหมด STOP หรือไม่ ถ้าต้องการให้ ตอบ Yes ถ้าไม่ต้องการให้ตอบ No

STOP	×
Place the PLC in	sTOP mode?

11. เมื่อเปลี่ยนเป็นโหมด STOP แล้ว สามารถแก้ไขหรือเปลี่ยนแปลงวงจรได้เมื่อแก้ไขเสร็จสมบูรณ์ ให้ทำตามขั้นตอน คือ Compile All,

Download, RUN และ Program Status ตามลำดับ

and Commun

## คำสั่ง Retentive On-Delay Timer (TONR)

หลักการทำงานของคำสั่ง Retentive On-Delay Timer (TONR) จะมีลักษณะการทำงานคล้ายกับ TON จะต่างกันเพียงบางจุดคือ เมื่อมีสัญญาณสภาวะ On เข้าที่ขา IN คำสั่ง TONR จะเริ่มนับเวลา ขณะที่ TONR กำลังนับเวลา หากหยุดจ่ายสัญญาณเข้าที่ขา IN ค่าเวลา ของ TONR จะหยุดและค้างเวลาตำแหน่ง ที่หยุดเอาไว้เมื่อจ่ายสัญญาณสภาวะ On เข้าที่ขา IN ของคำสั่ง TONR อีกครั้ง เวลาในการนับของ TONR จะเริ่มนับต่อจากตำแหน่งที่หยุดและเมื่อค่าเวลาของ TONR ที่นับมีค่าเท่ากับค่าเวลาที่กำหนดไว้ใน P Reset Time (PT) คอนแทคช่ วยของ TONR จะทำงาน เมื่อหยุดสัญญาณเข้าที่ขา IN คำสั่ง TONR ค่าเวลา ในการนับของ TONR จะหยุดและค้างเวลาอีก จะไม่สามารถ Reset ค่าเวลาของ TONR ให้เป็น "0" ได้ใน การ Reset ค่าเวลาของ TONR จะต้องใช้คำสั่ง Reset จากภายนอก เมื่อ TONR ถูก Reset ค่าเวลาจะ Reset กลับไปเป็น 0 คอนแทคช่วยจะกลับสู่สภาวะเดิม



คุณลักษณะของคำสั่ง Retentive On-Delay Timer (TONR)

	ชนิด	ค่าความละเอียด	ค่าสูงสุดในการตั้งเวลา	หมายเลขตัวตั้งเวลา
I	Timer Type	Resolution	Maximum Value	Timer Number
9	TONP	1 ms	32.767 s	T0, T64
	(retentive)	10 ms	327.67 s	T1 – T4, T65 – T68
		100 ms	3276.7 s	T5 – T31, T69 –T95

การทำงานของโปรแกรม

เมื่อ On อินพุต 10.0 ทำให้ Q0.0 ทำงาน คอนแทคช่วย Q0.0 เปลี่ยนสภาวะจาก ปกติเปิดเป็นปิดมีสัญญาณเข้าที่ขา IN คำสั่ง TONR (T1) จะเริ่มนับเวลา ขณะที่ TONR นับเวลา หากอินพุต I0.1 มีสภาวะ On เอาต์พุต Q0.0 จะหยุดทำงานไม่มีสัญญาณเข้าที่ ขา IN คำสั่ง TONR (T1) ค่าเวลาการนับของ TONR จะหยุดและค้างเวลา ตำแหน่งที่ หยุดไว้และเมื่ออินพุต I0.0 มีสภาวะ On อีกครั้ง TONR จะเริ่มนับจากตำแหน่งที่หยุด และเมื่อค่าเวลา TONR มีค่าเท่ากับ 2.5 วินาทีคำสั่ง TONR จะทำงานทำให้คอนแทคช่ ้วยของ TONR เปลี่ยน สภาวะจากเปิดเป็นปิด ทำให้เอาต์พุต Q0.1 ทำงาน และเมื่ออินพุต I0.1 มีสภาวะ On เอาต์พุต Q0.0 จะหยุด ทำงานไม่มีสัญญาณ เข้าที่ขา IN คำสั่ง TONR(T1) TONR จะหยุด และค้างเวลาตำแหน่งที่หยุดเอาไว้อีกครั้ง ไม่สามารถ Reset ค่าเวลา ได้ (การ Reset ค่าเวลาให้กลับเป็น 0 ต้องเขียนคำสั่ง Reset โปรแกรมจากภายนอกส่งเข้าไปที่ TONR) และเมื่อ TONR Reset ส่งผลให้เอาต์ พุต Q0.1 หยุดทำงานจากรูป I0.2 เป็นอินพุตที่ Reset คำสั่ง TONR (T1)



วิธีออกแบบโปรแกรม

 เปิดโปรแกรม STEP 7-Micro/WIN ทำการ Communication เพื่อติดต่อสื่อสารระหว่าง คอมพิวเตอร์ กับ PLC เริ่มต้นการเขียน โปรแกรมตามตัวอย่าง

 เขียนโปรแกรมตามขั้นตอนจนถึงคำสั่ง TONR ดับเบิ้ลคลิกเมาส์ซ้ายที่กลุ่มคำสั่ง Timers จะ ปรากฏคำสั่ง Timer ทั้ง 5 ชนิด เลื่อนเมาส์ ไปชี้ที่คำสั่ง TONR คลิกเมาส์ซ้ายเพื่อเรียกใช้งานคำสั่ง TONR และคำสั่ง TONR จะปรากฏเข้ามาใน Main Program



3. กำหนดค่าหมายเลขตัวตั้งเวลาของ TONR และค่าเวลาของ TONR (PT) ตามโปรแกรมตัวอย่าง และเขียนโปรแกรมจนถึงคำสั่ง Reset

เข้าต้องการ Reset

4. ทดลองการทำงานโดยการโยกสวิตช์10.0 (On-Off) สังเกตการทำงานของ TONR ค่าเวลาจะเริ่ม นับเพิ่มขึ้นเรื่อย ๆ



5. โยกสวิตซ์ I0.1 (On–Off) จะสังเกตเห็นว่า เอาต์พุต Q0.0 จะหยุดทำงาน (Off) ไม่มีสัญญาณเข้า ที่ขา IN ของคำสั่ง TONR (T1) ค่ าเวลาในการนับของ Timer จะหยุดและค้างเวลาตำแหน่งที่หยุดเอาไว้



6. เมื่อโยกสวิตช์ 10.0 ขึ้นอีกครั้ง TONR จะเริ่ม นับค่าเวลาต่อจากเวลาที่หยุด

ไว้และจะนับเวลาจน ครบตามค่าเวลาที่ตั้งไว้คอนแทคช่วยของ TONR

จะ เปลี่ยนสภาวะส่งผลให้เอาต์พุตทำงาน

7. โยกสวิตช์ I0.1(On-Off) เอาต์พุต Q0.0 จะ หยุดทำงาน(Off) สังเกตค่ าเวลาของTONR จะไม่ Reset แม้จะตัดสัญญาณออกจากคอยล์ Timer

หากต้องการ Reset ค่าเวลาของ TONR จะต้องเขียนโปรแกรม Reset เพิ่มขึ้นเข้าที่ TONR ใน ที่นี้คือการโยกสวิตช์ I0.2 ขึ้น โปรแกรมจะกลับสู่ สภาวะเริ่มต้นใหม่อีกครั้ง








หลักการทำงานของคำสั่ง Off-Delay Timer (TOF) คือเมื่อมีสัญญาณสภาวะ On ให้อินพุตเข้าที่ขา IN ของ TOF คอนแทคช่วยของ TOF จะทำงานทันทีและเมื่อหยุดจ่ายสัญญาณที่ขา IN จะทำให้ TOF เริ่ม นับเวลา และเมื่อค่าเวลา TOF มีค่าเท่ากับค่าเวลาที่กำหนดไว้ใน P Reset Time (PT) คอนแทคช่วยของ TOF จะหยุดทำงาน



คุณลักษณะของคำสั่ง Off-Delay Timer (TOF)

ชนิด	ค่าความละเอียด	ค่าสูงสุดในการตั้งเวลา	หมายเลขตัวตั้งเวลา
Timer Type	Resolution	Maximum Value	Timer Number
TOF (non-retentive)	1 ms	32.767 s	T32, T96
	10 ms	327.67 s	T33-T36, T97-T100
	100 ms	3276.7 s	T37-T63, T101-T255

การทำงานของโปรแกรม

เมื่อมีสภาวะ On ที่อินพุต I0.0 ส่งผลให้เอาต์พุต Q0.0 ทำงาน (On) คอนแทคช่วยของเอาต์พุต Q0.0 เปลี่ยนสภาวะ จากปกติเปิดเป็ นสภาวะปิดใน Network 2 ทำให้มีสัญญาณ เข้าที่ขา IN ของ TOF (T200) คอนแทคช่วยของ TOF ใน Network 3 จะทำงานทันทีส่งผลให้ เอาต์พุต Q0.1 ทำงาน (On)

เมื่ออินพุต I0.1 มีสภาวะ On ส่งผลให้เอาต์พุต Q0.0 หยุดทำงาน(Off) ไม่มีสัญญาณเข้าที่ขา IN ของ TOF (T200) TOF จะเริ่มนับ เวลาและเมื่อค่าเวลาของ TOF มีค่าเท่ากับ 4 วินาทีคอนแทคช่วยของ TOF จะกลับสู่สภาวะเดิมใน Network 3 เอาต์พุต Q0.1 หยุดทำงาน



#### วิธีออกแบบโปรแกรม

 เปิดโปรแกรม STEP 7-Micro/WIN ทำการ Communication เพื่อติดต่อสื่อสารระหว่างคอมพิวเตอร์ กับ PLC แล้วเริ่มต้นการเขียน โปรแกรมตามตัวอย่าง

- 2. เขียนโปรแกรมตามขั้นตอน จนถึงคำสั่ง Timer ดับเบิ้ลคลิกเมาส์ซ้ายที่กลุ่มคำสั่ง Timers จะ ปรากฏคำสั่ง Timer ทั้ง 5 ชนิด
- 3. เลื่อนเมาส์ไปที่คำสั่ง TOF คลิกเมาส์ซ้ายเพื่อเร<mark>ียกใช้งานคำสั่ง TOF คำสั่ง TOF จะปรากฏเข้ามา</mark> ใน Main Program



4. กำหนดค่าหมายเลขตัวตั้งเวลาของ TOF และค่าเวลาของ TOF (PT) ตามโปรแกรมตัวอย่างและ เขียนโปรแกรมต่อไปจนครบ เมื่อเขียน โปรแกรมครบให้ตรวจสอบความถูกต้องของโปรแกรมสังเกตผลการ ตรวจสอบในช่อง Output Windows ว่ามี Error หรือไม่ถ้ามีให้กลับไป แก้ไขโปรแกรมให้ถูกต้อง ถ้าไม่มีให้ ทำตามขั้นตอนต่อไปคือการดาวน์โหลดแล้วทดลองการทำงานของโปรแกรม (RUN) หากต้องการดูสภาวะ การทางานของวงจรสามารถทำได้โดยเลื่อนเมาส์ไปขี้ที่คำสั่ง Program Status บน Tool Bars

5. ทดลองการทำงานโดยโยกสวิตช์10.0 (On–Off) เอาต์พุต Q0.0 จะทำงานคอนแทคช่วย ของ TOF ใน Network 3 เปลี่ยนสภาวะจากปกติ เปิดเป็นสภาวะปิดทันทีส่งผลให้เอาต์พุต Q0.1 ทำงาน แต่ TOF ยังไม่เริ่มนับเวลาจนกว่าอินพุต10.1 มีสภาวะ On ตัดสัญญาณออกจากเอาต์

พุต TOF



6. เมื่อโยกสวิตซ์I0.1 (On-Off) เอาต์พุต Q0.0 หยุดทำงาน TOF จะเริ่มนับเวลา



7. ค่าเวลาของ TOF จะนับไปเรื่อย ๆจนค่าที่ นับมีค่าเท่ากับค่าที่ตั้งไว้จะสั่งให้คอนแทคช่วยของ TOF กลับสู่สภาวะปกติ โปรแกรมจะกลับสู่ สภาวะ เริ่มต้นใหม่



ตัวอย่าง การออกแบบ Ladder Diagram โดยใช้คำสั่ง Timers

1. ศึกษาขั้นตอนการทำงาน

(1) เมื่อกดสวิตช์ Start ให้เอาต์พุต 1 มีสภาวะ On แล้วให้หน่วงเวลา 1 วินาที

(2) จากนั้นให้เอาต์พุต 2 มีสภาวะ Onให้เอาต์พุต 1 มีสภาวะ Off

(3) เอาต์พุต 2 มีสภาวะ On ให้หน่วงเวลาไว้ 1 วินาที แล้วให้เอาต์พุต 3 มีสภาวะ On ให้เอาต์พุต 2 มีสภาวะ Off

(4) เมื่อกดสวิตช์ Stop ให้เอาต์พุตทุกตัวมีสภาวะ Off ทั้งหมด

2. กำหนดตำแหน่งอินพุต/เอาต์พุต/ตัวตั้งเวลา (Input/Output/Timer)

อินพุต(Input)	เอาต์พูต (Output)	ตัวตั้งเวลา(Timer)
สวิตช์ Start : 10.0	เอาต์พุต 1 : Q0.0	ตัวตั้งเวลาตัวที่ 1 (TON) : T101
สวิตช์ Stop : 10.1	เอาต์พุต 2 : Q0.1	ตัวตั้งเวลาตัวที่ 2 (TON) : T102
	เอาต์พุต 3 :Q0.2	

#### 3. เขียน Ladder Diagram



ตัวอย่าง การออกแบบ Ladder Diagram โดยใช้คำสั่ง TIMERS และ SET, RESET 1. ศึกษาขั้นตอนการทำงาน

(1) เมื่อกดสวิตช์ Start ส่งผลให้เอาต์พุต1 – เอาต์พุต3 มีสภาวะ On–Off ทุก ๆ 0.3 วินาทีไปเรื่อย ๆจนกว่าจะกดสวิตช์ Stop

- (2) เมื่อกดสวิตช์ Stop ให้เอาต์พุต 1 และ เอาต์พุต 3 มีสภาวะ Off
- 2. กำหนดตำแหน่งอินพุต/เอาต์พุต/ตัวตั้งเวลา (Input/Output/Timer)
- 3. เขียน Ladder Diagram





# กลุ่มคำสั่ง Counters



## คำสั่ง Count Up (CTU)

คำสั่ง Count Up (CTU) หรือคำสั่งนับขึ้น ประกอบด้วย 2 Input คือ Count Up Input (CU) และ Reset Input (R) การใช้งาน Count Up จะต้องระบุ Counter ranges ว่าเป็นหมายเลขเท่าไร (ค่า Counter ranges : CXXX สามารถระบุได้ตั้งแต่หมายเลข C0–C255) และค่าในการนับ (Preset Values) เป็นจำนวน เท่าไร (ค่า Preset Value : PV สามารถตั้งค่าการนับได้ตั้งแต่ 0–32,767 ครั้ง)



ตัวอย่าง การใช้งานคำสั่ง Count Up (CTU) การทำงานของโปรแกรม เมื่อโยกสวิตช์ I0.0 (On-Off) มีสัญญาณเข้าที่ขา CU สังเกตค่าการนับของ CTU จะเริ่มนับที่ 0 และนับขึ้นเรื่อย ๆ ตามจังหวะการโยกสวิตช์ I0.0 เมื่อ โยกสวิตช์ I0.0 (On-Off) จำนวน 4 ครั้ง คำสั่ง CTU จะทำงาน ส่งผลให้ คอนแทคช่วยของ CTU เปลี่ยนสภาวะจากปกติเปิ ดเป็นปิดทำให้เอาต์พุต Q0.0 ทำงาน และเมื่อโยกสวิตช์ I0.1 (On-Off) มีสัญญาณเข้าที่ขา Reset คำสั่ง CTU จะถูก Reset ทำให้ CTU หยุดทำงาน ค่าการนับกลับไปค่า เริ่มต้นที่ 0 คอนแทคช่วย ของ CTU ใน Network 2 จะกลับสู่สภาวะเดิม กลับเป็นสภาวะเปิด เอาต์พุต Q0.0 จะหยุดทำงาน



วิธีออกแบบโปรแกรม

 เปิดโปรแกรม STEP 7-Micro/WIN ทำการ Communication เพื่อติดต่อสื่อสารระหว่าง คอมพิวเตอร์ กับ PLC เริ่มต้นการเขียน โปรแกรม

2. ให้เขียนโปรแกรมตามตัวอย่าง จนถึงกลุ่มคำสั่ง Counters ดับเบิ้ลคลิกที่คำสั่ง Counters จะ ปรากฏคำสั่ง Count Up (CTU) ขึ้นมา



3. ให้เลื่อนเมาส์ชี้ที่คำสั่ง CTU คลิกเมาส์ซ้าย 1 ครั้งจะปรากฏคำสั่ง CTU ใน Main Program



#### 4. จากนั้นเขียนโปรแกรมตามจนครบ



5. หากโปรแกรมที่ทดลองผิดพลาด สามารถกลับไปแก้ไขได้เมื่อแก้ไขเสร็จสมบูรณ์ให้ทำตาม ขั้นตอนเดิม คือ Compile All, Download, RUN และ Program Status ตามลำดับ 6. ทดลองการทำงาน โดยการโยกสวิตช์ 10.0 สังเกตค่าการนับของเคาน์เตอร์จะเริ่มนับที่ 0 และเมื่อโยกสวิตช์ 10.0 เป็นจำนวน 4ครั้ง ค่าการ นับจะเพิ่มขึ้นเรื่อย ๆ ตามจังหวะในการโยกสวิตช์ 10.0 เมื่อครบ 4 ครั้ง คำสั่ง CTU ทำงาน ส่งผลให้เอาต์พุต Q0.0 ทำงานด้วย



7. โยกสวิตช์ I0.1 เพื่อReset การทำงานของ CTU ค่าการนับของ CTU จะถูก Reset กลับไปเป็น 0 เอาต์พุต Q0.0 จะหยุดการทำงาน



### คำสั่ง Count Down (CTD)

คำสั่ง Count Down หรือคำสั่งแบบนับลง ประกอบด้วย 2 Input คือ Count Down Input (CD) และ Load Input (LD) (การ ทำงานของ Load Input มีการทำงานเหมือนกับ Reset Input แตกต่างกันเฉพาะที่ ชื่อสำหรับเรียกขา Input เท่านั้น) การเรียกใช้งานคำสั่ง Count Down ต้องระบุ Counter ranges ว่าเป็น หมายเลขใด (โดยค่า Counter ranges : Cxxx ระบุได้ตั้งแต่หมายเลข C0–C255) และ ค่าในการนับ (Preset Values) เป็นจำนวนเท่าไร (ค่า Preset Value : PV จะตั้งค่าการนับได้ตั้งแต่ 0–32,767 ครั้ง) และที่ สำคัญคือหาก Count Down หมายเลขใดถูกใช้ไปแล้ว ไม่ว่าจะเป็น Counter แบบใดก็ตาม จะเรียกใช้ หมายเลขนั้นซ้ำอีกไม่ได้



ตัวอย่าง การใช้งานคำสั่ง Count Down (CTD)

เริ่มต้นการทำงานของโปรแกรมค่าการนับของ CTD จะ เริ่มที่ 5 (กรณีที่ค่าการนับไม่เริ่มที่ 5 ให้โยกสวิตช์I0.1 (OnOff) ก่อนหนึ่งครั้ง) เมื่อโยกสวิตช์ I0.0 (On-Off) มีสัญญาณ เข้าที่ขา CD ค่าการนับของ CTD จะนับลดลงเรื่อย ๆ ตาม จังหวะการโยกสวิตช์ I0.0 เมื่อโยก สวิตช์I0.0 (On-Off) เป็น จำนวน 5 ครั้ง (ตามค่าที่ตั้งไว้ที่ PV) ค่าการนับจะลดลงจนถึง 0 คำสั่ง CTD จะทำงาน ส่งผลให้คอนแทคช่วยของ CTD ใน Network 2 เปลี่ยนสภาวะจากปกติเปิดเป็นปิด ทำให้เอาต์พุต Q0.0 ทำงาน และเมื่อโยกสวิตช์ I0.1 (On-Off) มีสัญญาณเข้าที่ขา LD คำสั่ง CTD จะถูก Reset ทำให้ CTD หยุดทำงาน ค่าการ นับจะกลับไปค่าเริ่มต้นที่ 5 คอนแทคช่วยของ CTD ใน Network 2 จะกลับสู่สภาวะเดิมกลับเป็นสภาวะเปิด เอาต์พุต Q0.0 จะหยุดทำงาน



วิธีออกแบบโปรแกรม

 เปิดโปรแกรม STEP 7-Micro/WIN ทำการ Communication เพื่อติดต่อสื่อสารระหว่าง คอมพิวเตอร์ กับ PLC เริ่มต้นเขียนโปรแกรม ตามตัวอย่าง

ให้เขียนโปรแกรมตามตัวอย่าง จนถึงกลุ่ม คำสั่ง Counters ดับเบิ้ลคลิกที่กลุ่มคำสั่ง Counters จะปรากฏคำ สั่ง Count Down (CTD) ขึ้นมา



 3. ให้เลื่อนเมาส์ชี้ที่คำสั่ง CTD คลิกเมาส์ซ้าย 1 ครั้งจะปรากฏคำสั่ง CTD ใน Main Program
4. เครื่องหมาย ???? ด้านบน ให้ระบุ Counter ranges (จากตัวอย่างเป็น Counter C19) เครื่องหมาย???? ในส่วน PV ให้ระบุค่าจำนวนนับ ของเคาน์เตอร์ (จากตัวอย่างตั้งค่าการนับ 5 ครั้ง)



5. จากนั้นเขียนโปรแกรมตามตัวอย่างจนครบ

 หากโปรแกรมที่ทดลองผิดพลาด สามารถกลับไปแก้ไขได้เมื่อแก้ไขเสร็จสมบูรณ์ให้ทำตาม ขั้นตอนเดิม คือ Compile All, Download, RUN และ Program Status ตามลำดับ  7. ทดลองการทำงาน โดยโยกสวิตช์ 10.0 สังเกตค่าการนับของเคาน์เตอร์จะเริ่มนับที่ 5 (หากค่าการนับไม่เริ่มต้นที่ 5 ให้โยกสวิตช์ 10.1 เพื่อ Reset ค่าการนับของ CTD ก่อน)



 8. 8. โยกสวิตช์ I0.0 จำนวน 5 ครั้ง ค่าการนับจะลดลงเรื่อย ๆ ตามการโยกสวิตช์ I0.0 เมื่อครบ 5 ครั้ง ค่าการนับจะลดลงจนถึง 0 คำสั่ง CTD ทำงาน ส่งผลให้เอาต์พุต Q0.0 ทำงาน



9. โยกสวิตช์ I0.1 เพื่อ Reset การทำงานของ CTD ค่าการนับของ CTD จะถูก Reset กลับไปเป็น 5



### คำสั่ง Count Down (CTD)

คำสั่ง Count Up/Down หรือคำสั่งตัวนับแบบนับขึ้นและนับลงประกอบด้วย 3 Input คือ Count Up Input (CU), Count Down Input (CD) และ Reset Input (R) การเรียกใช้งานคำสั่ง Count Up/Down จะต้องระบุ Counter ranges ว่าเป็นหมายเลขใด (ค่า Counter ranges : Cxxxสามารถระบุหมายเลขได้ ตั้งแต่หมายเลข C0–C255) และค่าการนับ (Preset Values) เป็นจำนวนเท่าไร (ค่า Preset Value : PV สามารถตั้งค่าการนับได้ตั้งแต่ 0–32,767 ครั้ง) และหาก Count Up/Down หมายเลขใดถูกใช้ไปแล้ว ไม่ว่า จะเป็น Counter แบบใดก็ตาม จะเรียกใช้หมายเลขนั้นซ้ำอีกไม่ได้



ตัวอย่าง การใช้งานคำสั่ง Counter Up/Down (CTUD)

การทำงานของโปรแกรม เมื่อโยกสวิตซ์ I0.0 (On-Off) มีสัญญาณเข้าที่ขา CU ค่าการนับของ CTUD จะเริ่มนับที่ 0 และค่าการนับ จะ เพิ่มขึ้นเรื่อย ๆ ตามจังหวะการโยกสวิตซ์ I0.0 เมื่อค่า การนับมีค่า ≥ 6 คำสั่ง CTUD จะทำงานส่งผลให้คอน แทคช่วยของ CTUDใน Network 2 เปลี่ยนสภาวะจาก ปกติเปิดเป็นสภาวะปิด ทำให้เอาต์พุต Q0.0 ทำงาน และเมื่อโยกสวิตซ์ I0.1 (On-Off) มีสัญญาณเข้าที่ขา CD ค่าการนับจะลดลงเรื่อย ๆ ตามจังหวะการโยก สวิตซ์ I0.1 เมื่อค่าการนับมีค่าน้อยกว่า 6 คำสั่ง CTUD จะหยุดทำงานส่งผลให้คอนแทคช่ วยของ CTUD ใน Network 2 กลับสภาวะเดิม จากสภาวะปิดเป็นสภาวะ เปิด ทำให้เอาต์พุต Q0.0 หยุดทำงาน และเมื่อโยกสวิตช์ I0.2 (On-Off) มีสัญญาณข้า ที่ขา Reset (R)คำสั่ง CTUD จะ ถูก Reset ทำให้CTUD หยุดทำงานค่าการนับกลับไปเริ่มต้นที่ 0 คอนแทคช่วย CTUD ใน Network 2 จะกลับสู่สภาวะเดิมกลับเป็น ปกติเปิด เอาต์พุต Q0.0 จะหยุดทำงาน



วิธีออกแบบโปรแกรม

 เปิดโปรแกรม STEP 7-Micro/WIN ทำการ Communication เพื่อติดต่อสื่อสารระหว่าง คอมพิวเตอร์ กับ PLC เริ่มต้นเขียนโปรแกรม ตามตัวอย่าง

2. ให้เขียนโปรแกรมตามตัวอย่าง จนถึงกลุ่มคำสั่ง Counters ดับเบิ้ลคลิกที่กลุ่มคำสั่ง Countersจะ ปรากฏคำสั่ง Count Up/Down (CTUD) ขึ้นมา



3. ให้เลื่อนเมาส์ชี้ที่คำสั่ง CTUD คลิกเมาส์ซ้าย 1 ครั้งจะปรากฏคำสั่ง CTUD ใน Main Program

4. เครื่องหมาย ???? ด้านบน ให้ระบุ Counter ranges (จากตัวอย่างเป็น Counter C38) เครื่องหมาย ???? ในส่วน PV ให้ระบุค่าจำนวน

นับของ CTUD (จากตัวอย่างค่าการนับ = 6 ครั้ง)



5. จากนั้นเขียนโปรแกรมจนครบ

6. หากโปรแกรมที่ทดลองผิดพลาด สามารถกลับไปแก้ไขได้เมื่อแก้ไขเสร็จสมบูรณ์ให้ทำตาม ขั้นตอนเดิม คือ Compile All, Download,

RUN และ Program Status ตามลำดับ

7. ทดลองการทำงาน โดยการโยกสวิตซ์ 10.0 สังเกตค่าการนับของ Counter จะเริ่มนับที่ 0 (หากค่าไม่เริ่มต้นที่ 0 ให้โยกสวิตซ์ 10.2 เพื่อ Reset คำสั่ง CTUD ก่อน)



 8. 8. โยกสวิตช์ I0.0 (On-Off) ค่าการนับของ CTUD จะเริ่มนับที่ 0 และค่าการนับจะเพิ่มขึ้นเรื่อย ๆ ตามจังหวะในการโยกสวิตช์ I0.0 เมื่อค่ าการนับมีค่า ≥ 6 CTUD จะทำงาน ส่งผลให้เอาต์พุต Q0.0 ทำงาน



9. โยกสวิตช์ I0.1 (On-Off) ค่าการนับของ CTUD จะลดลงเรื่อย ๆ เมื่อค่าการนับมีค่าน้อยกว่า 6 CTUD จะหยุดทำงาน ส่งผลให้เอาต์พุต Q0.0 หยุดทำงาน



10. โยกสวิตช์ I0.2 เพื่อ Reset การทำงานของ CTUD ค่าการนับของ CTUD จะ Reset กลับมามีค่า 0





# การควบคุมมอเตอร์โดยใช้

# โปรแกรมเมเบิลคอนโทรลเลอร์





ประเภทของการควบคุมมอเตอร์สามารถแบ่งวิธีของการควบคุมมอเตอร์ได้ 3 วิธีคือ การควบคุมด้วยมือ (Manual control)

การควบคุมด้วยมือคือการใช้ ผู้ปฏิบัติงานทำหน้าที่ควบคุมอุปกรณ์เครื่องกลให้ ทำงาน โดยใช้วิธีการจ่ายแรงดันไฟฟ้าให้กับ มอเตอร์ ไฟฟ้าโดยตรงหรือผ่านอุปกรณ์ควบคุมการ ทำงานด้วยมือแบบต่าง ๆ เช่นการเสียบปลั๊กหรือ ใช้สวิตช์สตาร์ทมอเตอร์ (Starter Switch) เป็นต้น วิธีการควบคุมด้วยมือนี้มักจะใช้กับมอเตอร์ไฟฟ้า ที่มีขนาดเล็กประเภทเครื่องใช้ไฟฟ้าภายในบ้าน ทั่วไปเพื่อการเริ่มเริ่มต้นทำงานหรือ หยุดเครื่องเป็น ส่วนใหญ่



9.1.2 การควบคุมกึ่งอัตโนมัติ (Semi Automatic control) การควบคุมกึ่งอัตโนมัติเป็นการนำอุปกรณ์แมกเนติกส์คอนแทคเตอร์และสวิตช์ปุ่ มกด เข้า มาช่วยควบคุมการทำงานของมอเตอร์ไฟฟ้า สามารถออกแบบวงจรควบคุม เพื่อให้เริ่มต้นการทำงาน (Start) และหยุดทำงาน (Stop) ได้อีกทั้งยังสามารถจัดวางตู้ ควบคุมให้ห่างจากเครื่องจักร เพื่อเพิ่มความ ปลอดภัยให้กับผู้ควบคุมมากยิ่งขึ้น

การควบคุมอัตโนมัติ (Automatic control) การควบคุมอัตโนมัติมีลักษณะคล้ายกับการควบคุมแบบกึ่งอัตโนมัติแตกต่างกัน หลังจาก กดปุ่มเริ่มต้นการทำงานระบบจะทำงานเองตลอดเวลาจึงต้องติดตั้งอุปกรณ์ เสริมเข้าไปในวงจรควบคุมการ ทำงานเพื่อให้ระบบสามารถทำงานได้อย่างต่อเนื่องและ ถูกต้องตามเงื่อนไขการทำงาน เช่น การติดตั้งสวิตช์ ลูกลอย (Float Switch) ทำหน้าที่ ตรวจวัดระดับฎ้าในถัง สั่งให้มอเตอร์ปั้มทำงานเมื่อฎ้าหมดถังและสั่งให้ มอเตอร์หยุด เมื่อฎ้าเต็มถัง หรือการติดตั้งสวิตช์ความดัน (Pressure Switch) ทำหน้าที่ตรวจวัด ปริมาณ ความดันลม เพื่อสั่งให้ปั้มลมทำงานเป็นต้น







สำหรับการควบคุมมอเตอร์ ไฟฟ้า สามารถแบ่งส่วนประกอบ ที่สำคัญในการควบคุมมอเตอร์ ไฟฟ้าออกเป็น 2 ส่วน ด้วยกันคือ วงจรกำลัง (Power Circuit) และ วงจรควบคุม (Control Circuit)



จากรูปแสดงการควบคุมมอเตอร์ไฟฟ้าโดยใช้แมก เนติกส์คอน แทคเตอร์เป็นอุปกรณ์ควบคุมหลักในการเริ่มทำงาน และหยุดทำงาน ของมอเตอร์มีลักษณะการทำงานคือเมื่อกดสวิตช์ S2 มอเตอร์จะเริ่ม ต้นทำงานและเมื่อกดสวิตช์ S1 มอเตอร์จะหยุด ทำงาน

ในปัจจุบันการควบคุมเครื่องจักรที่มีมอเตอร์ไฟฟ้าเป็นอุปกรณ์ต้นกำลังในโรงงานอุตสาหกรรมนิยม นำเครื่องควบคุมแบบอัตโนมัติ หลากหลายรูปแบบมาใช้งาน เพื่อควบคุมช่วยทำให้ระบบมีความรวดเร็ว แม่นยำและตรวจสอบแก้ไขระบบได้ง่ายขึ้น อีกทั้งยังทำให้ระบบมี ความน่าเชื่อถือมากขึ้น และ PLC เป็น หนึ่งในเครื่องควบคุมอัตโนมัติที่นิยมใช้แต่เราไม่สามารถนำเอาต์พุตของ PLC ไปควบคุมมอเตอร์ โดยตรง เนื่องจากกระแสของมอเตอร์ค่อนข้างสูงซึ่งอาจทำให้เอาต์พุตของ PLC เสียหายได้จึงต้องต่ออุปกรณ์ขับกั้น กลางระหว่างเอาต์พุตของ PLC กับมอเตอร์และอุปกรณ์ที่นิยมใช้คือแมกเนติกส์คอนแทคเตอร์ การควบคุมมอเตอร์โดยใช้โปรแกรมเมเบิลคอนโทรลเลอร์

การควบคุมมอเตอร์โดยใช้โปรแกรมเมเบิลคอนโทรลเลอร์หมายถึงการนำสัญญาณเอาต์พุตของ PLC ไปต่อเข้ากับอุปกรณ์ขับตรงส่วน

้ควบคุมคอยล์ (Coil) ของแมกเนติกส์คอนแทคเตอร์และเปลี่ยนวงจร ควบคุม (Control Circuit) มาเป็น Ladder Diagram เพื่อป้อนเข้าไป

ใน PLC โดยที่วงจรกำลัง (Power Circuit) ยังคงต่อใช้งานตามปกติ

ขั้นตอนในการเปลี่ยนวงจรควบคุมมาเป็น Ladder Diagram ปฏิบัติดังนี้

1. กำหนดตำแหน่งอินพุต/เอาต์พุต (Input/Output)

อินพุต(Input)	เอาต์พูต (Output)
Overload Relay (F3) : 10.0	มอเตอร์ (K1) : Q0.0
Switch Stop (S1): I0.1	
Switch Start (S2) : 10.2	

2. แปลงวงจรจากวงจรควบคุมมาเป็น Ladder Diagram



แปลงวงจรใหม่ ให้ง่ายในการทำความเข้าใจการทำงานของโปรแกรมโดยการนำส่วนที่ หน้าสัมผัสขนานกันมาวางไว้ด้านหน้า



- 4. ป้อนโปรแกรมตาม Ladder Diagram โหลดลงใน PLC
- 5. ต่อสายควบคุมทางด้านอินพุต/เอาต์พุต (Input/Output) ตามรูป

การต่ออุปกรณ์ทางด้านอินพุต/เอาต์พุต (Input/Output)



ตัวอย่างการควบคุมมอเตอร์โดยใช้โปรแกรมเมเบิลคอนโทรลเลอร์

การควบคุมมอเตอร์ไฟฟ้า 3 เฟส ด้วยวงจรกลับทางหมุนหลังจากหยุด มอเตอร์ (Reversing After stop)



 ศึกษาเงื่อนไขการทำงาน : เมื่อกดสวิตซ์ S3 จะทำให้คอนแทคเตอร์ K1 ทำงาน ส่งผล ให้มอเตอร์ทำงานกำหนดให้ทิศทางการหมุนตามเข็ม นาฬิกา กดสวิตซ์ S1 มอเตอร์หยุดการทำงาน และเมื่อ กดสวิตซ์ S2 จะทำให้คอนแทคเตอร์ K2 ทำงาน ส่งผลให้มอเตอร์ทำงานกำหนดให้ทิศ ทางการหมุนทวน เข็มนาฬิกา หากกดสวิตซ์ S1 มอเตอร์หยุดการทำงาน 2. กำหนดตำแหน่งอินพุต/เอาต์พุต (Input/Output)

อินพุต (Input)	เอาต์พูต (Output)
Switch Forward(S3): I0.1	มอเตอร์หมุนตามเข็ม (K1) : Q0.0
Switch Reverse(S2) : I0.2	มอเตอร์หมุนทวนเข็ม (K2) : Q0.1
Switch Stop (S1) : I0.3	

3. เขียน Ladder Diagram


4. ต่ออุปกรณ์ทางด้านอินพุต/เอาต์พุต (Input/Output)



การควบคุมมอเตอร์ไฟฟ้า 3 เฟส ด้วยวงจรกลับทางหมุนแบบ Jogging



 ศึกษาเงื่อนไขการทำงาน : เมื่อกดสวิตช์ S2 จะทำให้คอนแทคเตอร์ K1 ทำงาน ส่งผล ให้มอเตอร์ทำงานกำหนดให้ทิศทางการหมุนตามเข็ม นาฬิกา และเมื่อกดสวิตช์ S2 จะทำให้คอนแทคเตอร์ K2 ทำงาน ส่งผลให้มอเตอร์ทำงานกำหนดให้ทิศทางการหมุนทวนเข็มนาฬิกา

#### 2. กำหนดตำแหน่งอินพุต/เอาต์พุต (Input/Output)

อินพุต (Input)	เอาต์พุต (Output)
Switch Forward (S2) I0.0	มอเตอร์หมุนตามเข็ม (K1) : Q0.0
Switch Reverse (S3) I0.1	มอเตอร์หมุนทวนเข็ม (K2) : Q0.1

3. เขียน Ladder Diagram



4. ต่ออุปกรณ์ทางด้านอินพุต/เอาต์พุต (Input/Output)



การควบคุมมอเตอร์ไฟฟ้า 3 เฟส ด้วยวงจรกลับทางหมุนแบบ Interlocking



 ศึกษาเงื่อนไขการทำงาน : เมื่อกดสวิตช์ S2 จะทำให้คอนแทคเตอร์ K1 ทำงาน ส่งผล ให้มอเตอร์ทำงานกำหนดให้ทิศทางการหมุนตามเข็ม นาฬิกา กดสวิตช์ S1 มอเตอร์หยุดการทำงาน และเมื่อ กดสวิตช์S3จะทำให้คอนแทคเตอร์ K2 ทำงาน ส่งผลให้มอเตอร์ทำงานกำหนดให้ทิศ ทางการหมุนทวนเข็ม นาฬิกา หากกดสวิตช์ S1 มอเตอร์หยุดการทำงาน

#### 2. กำหนดตำแหน่งอินพุต/เอาต์พุต (Input/Output)

อินพุต (Input)	เอาต์พุต (Output)
Switch Forward (S2) : 10.0	มอเตอร์หมุนทวนเข็ม (K2) : Q0.1
Switch Reverse (S3) : I0.1	
Switch Stop (S1) : I0.2	มอเตอร์หมุนตามเข็ม (K1) : Q0.0

#### 3. เขียน Ladder Diagram



4. ต่ออุปกรณ์ทางด้านอินพุต/เอาต์พุต (Input/Output)





## การควบคุมระบบนิวเมติกส์ โดยใช้โปรแกรมเมเบิ้ลคอนโทรลเลอร์



#### ส่วนประกอบพื้นฐานของระบบนิวเมติกส์

ระบบนิวเมติกส์จะมีส่วนประกอบพื้นฐานที่สำคัญ คือ ส่วนต้นกำลัง ส่วนควบคุมคุณภาพลมอัดส่วน อุปกรณ์ควบคุมการทำงานส่วนอุป กรณ์ทำงานและส่วนอุปกรณ์ในระบบท่อทาง โดยส่วนต่าง ๆ ในระบบ นิวเมติกส์จะมีหน้าที่ในการทำงานแตกต่างกันไป ตามรายละเอียด ดังนี้คือ

ส่วนต้นกำลัง หมายถึง ส่วนที่ทำหน้าที่สร้างแรงดันลมอัดที่มีคุณภาพเพื่อนำมาใช้งานในระบบ นิวเมติกส์ประกอบไปด้วยส่วนต่างๆ ดังนี้
(1) อุปกรณ์ขับ (Driving unit) ทำหน้าที่ขับเครื่องอัดลม ได้แก่เครื่องยนต์หรือมอเตอร์ไฟฟ้า
(2) เครื่องอัดลม (Air compressor) ทำหน้าที่อัดอากาศที่ความดันบรรยากาศให้มีความดันสูง กว่าบรรยากาศปกติ



(3) เครื่องระบายความร้อน (After cooler) ทำหน้าที่หล่อเย็นอากาศอัดให้เย็นตัวลง

(4) ตัวกรองลมหลัก (Main line air filter)ทำหน้าที่กรองลมก่อนที่จะนำไปใช้งาน

(5) ถังเก็บลม (Air receiver) เป็นอุปกรณ์ใช้เก็บลมที่ได้จากเครื่องอัดลมและจ่ายลมที่ความดัน คงที่สมู่าเสมอให้แก่ระบบนิวเมติกส์ถัง เก็บลมจะต้องมีลิ้นระบายความดัน (Pressure relief valve) เพื่อระบายความดันที่เกินสู่บรรยากาศป้องกันอันตรายที่เกิดขึ้นเมื่อความดัน สูงกว่าปกติส่วนสวิตช์ควบคุม ความดัน (Pressure Switch) ใช้ควบคุมการเปิด-ปิดการทำงานของมอเตอร์ไฟฟ้าที่ขับเครื่องอัดลมเมื่อ ความ ดันของลมสูงถึงค่าที่ตั้งไว้

(6) เครื่องกำจัดความชื้น (Separator) หรือเครื่องทำลมแห้งอุปกรณ์นี้จะช่วยแยกเอาความชื้น และละอองน้ำมันที่แฝงมากับอากาศอัด ก่อนที่อากาศอัดจะถูกนำไปใช้งาน

 ส่วนควบคุมคุณภาพลมอัด (F.R.L Unit หรือ FRL Combination/ Preparation unit) หมายถึง ส่วนที่ทำหน้าที่ควบคุมคุณภาพลมอัด ทำให้ลมปราศจากฝุ่นละออง คราบน้ำมันและน้ำมันก่อนที่จะ นำไปใช้ในระบบนิวเมติกส์ประกอบด้วย ตัวกรองลม (Air filter) วาล์วปรับ ความดันพร้อมเกจ (Air Regulator) อุปกรณ์ผสมน้ำมันหล่อลื่น (AirLubricator)

(1) อักษร F หรือ AF มาจากคำเต็มว่า Air Filter หรือตัวกรองลม

(2) อักษร R หรือ AR มาจากคำเต็มว่า Air Regulator หรือวาล์วปรับความดันตัวปรับแรงดันลม เครื่องปรับแรงดันลม อุปกรณ์ ควบคุมความดันลม (3) อักษร L หรือ AL มาจากคำเต็มว่า Air Lubricator หรือ ตัวผสมน้ำมันหล่อลื่น เครื่องผสม น้ำมันหล่อลื่น อุปกรณ์ผสมน้ำมัน หล่อลื่น



 ส่วนอุปกรณ์ควบคุมการทำงาน หมายถึง ส่วนที่ทำหน้าที่ควบคุม การเริ่มทำงานและหยุดทำงานของวงจร ควบคุมทิศทางการไหลของลม ควบคุมอัตราการไหลของลมและ ควบคุมความดันลม นั่นคือวาล์ว (Valve) ควบคุมชนิดต่างๆในระบบนิวเมติกส์



4. ส่วนอุปกรณ์ทำงานหมายถึงส่วนที่ทำหน้าที่เปลี่ยนกำลังงานของไหลให้เป็นกำลังกล เช่น กระบอกสูบลมชนิดต่างๆ

กระบอกสูบ (Pneumatic Air Cylinder or Actuator) คือ อุปกรณ์ลมที่ใช้ลมทำให้ก้าน กระบอกลม เคลื่อนที่ไปในแนวเส้นตรง หรือ

หมุน 90, 180, 270 หรือ 360 องศา เป็นอุปกรณ์ที่เปลี่ยน พลังงานในรูปแบบความดันลมให้เป็นพลังงานกล



กระบอกสูบมาตรฐาน(Standard Cylinder)



กระบอกสูบแบบหยิบ-จับชิ้นงาน (Air Gripper Cylinder)



กระบอกสูบแบบกลมขนาดเล็ก(Mini Cylinder)



กระบอกสูบคอมแพคแบบสไลด์ด้านข้าง (Compact Slide Cylinder Roller Bearing)



กระบอกสูบจับยึดจับชิ้นงานแบบโรตารี่เคลื่อนที่ หมุน 90 ° (Rotary Clamp Cylinder (Push and Clockwise))



กระบอกสูบแบบสองแกน(Twin-Rod Cylinder)



กระบอกสูบไม่มีแกน (Rod less Magnetic Cylinder)



กระบอกสูบแบบคอมแพ็ค(Compact Cylinder)

5. อุปกรณ์ในระบบท่อทาง หมายถึง ส่วนที่เป็นท่อทางไหลของลมในระบบนิวเมติกส์ระบบท่อนี้ รวมถึงท่อลมและข้อต่อลมชนิดต่างๆ (1) ท่อลม (Tube) ใช้เชื่อมโยงระหว่างปั้มลมไปต่อกับวาล์วไปต่อกับกระบอกลม ท่อลมที่ดีควรมี คุณภาพดีมีขนาดสม่ำเสมอตลอด ความยาวทั้งเส้น มีขนาดพอดีกับข้อต่อลม (Air Fitting) ทุกชนิด โดยท่อ ลมที่ทำจากไนล่อน จะมีความเหนียวและคงทนต่อการสึกหรอของ ผิวได้ดีเป็นพิเศษ (Abrasion resistant) จึงได้รับความนิยมใช้งานในระบบนิวเมติกส์โดยทั่วไปและแพร่หลายมานาน แต่สำหรับการติดตั้ง ระบบ นิวเมติกส์ที่มีความซับซ้อนและมีพื้นที่จำกัดที่ต้องม้วนท่อลม (Tight Bend Radius) เช่น ในตู้ควบคุมไฟฟ้า นิยมใช้ท่อลมที่ทำจากโพ ลี่ยูรีเทน ซึ่งมีน้ำหนักเบา มีความยืดหยุ่นทนต่อแรงจากการสั่นสะเทือน และการ กัดกร่อนของสารเคมีได้ดีมีหลายสีให้เลือกใช้งานสำหรับการ ติดตั้งที่ต้องจำแนกชนิดของวงจร



ท่อลมไนลอน



ท่อลมโพลี่ยูรีเทน



ท่อลมโพลี่ยูรีเทนแบบขดสปริง

(2) ข้อต่อลม (Air Fitting) หรือฟิตติ้ง (Fitting) คือ ข้อต่อสำหรับเสียบสายลมที่ใช้กับระบบ นิวเมติกส์มีหน้าที่เชื่อมต่อระหว่างอุป กรณ์นิวเมติกส์กับสายลม หรือเชื่อมต่อระหว่างสายลมกับสายลม หรือหลายท่อสายลมบนฟิตติ้งลมก็ได้ข้อต่อลมมีหลายแบบ เช่น แบ่งชนิด ตามวัสดุและแบ่งชนิดตาม ลักษณะการใช้งาน เช่น ข้อต่อพลาสติก ข้อต่อสวมเร็ว (Quick Coupler) และข้อต่อโลหะ





ข้อต่อลมแบบต่างๆ (Air Fitting)

ข้อต่อสวมเร็ว (Quick Coupler)



6. สัญลักษณ์ระบบนิวเมติกส์ไฟฟ้า มีที่ใช้อยู่หลายระบบด้วยกัน เช่น ASA : American Standard Association), ISO : International Standard Organization, JIS: Japanese Industrial Standards, JIC : Joint Industry Conference, DIN : Deutsche Industrie Norm แต่ละระบบจะมีความแตกต่างกันไม่มากนัก ตำแหน่งการทำงานของวาล์วและสัญลักษณ์ที่ใช้แทนตำแหน่งการทำงานของวาล์วจะ แทนด้วยรูปสี่เหลี่ยม ดังต่อไปนี้

ตารางวิธีการกำหนดสัญลักษณ์ของวา<mark>ล์</mark>ว

สัญลักษณ์	ความหมาย
1	วาล์วควบคุม 1 ตำแหน่ง
1 0	วาล์วควบคุม 2 ตำแหน่งเป็นตำแหน่งปกติ 1 ตำแหน่งและทำงาน 1 ตำแหน่ง
1 2	วาล์วควบคุม 2 ตำแหน่งเป็นตำแหน่งการทำงาน 2 ตำแหน่ง
1 0 2	วาล์วควบคุม 3 ตำแหน่งมีตำแหน่งกลางเป็นตำแหน่งพัก(ปกติ) และมี 2 ตำแหน่งที่ทำงาน

#### ตารางสัญลักษณ์ของรูวาล์ว

หน้าที่	ตัวอักษรย่อ	ตัวเลข	ตัวอักษร
รูต่อลมอัดเข้ <mark>า</mark> วาล์ว	Sup	1	Ρ
รูต่ออัดไปใช้งาน	Out	2,4	A,B
รูระบายลมทิ้ง	Ex	3,5	R
รูต่อลมเข้าวาล์วเพื่อบังคับการทำงานของวาล์ว	Signal IN	12,14	X,Y,Z

ตัวอย่างการเขียนสัญลักษณ์ของรูวาล์วด้วยตัวเลข Double Solenoid Valve - 5/3



#### ตัวอย่างการเขียนสัญลักษณ์ของรูวาล์วด้วยตัวอักษร Double Solenoid Valve - 5/3



ตารางสัญลักษณ์ของวาล์วควบคุมทิศทาง

ลำดับ	สัญลักษณ์	ความหมาย
1.		วาล์วควบคุมทิศทางแบบ 2/2 ปกติปัด (2/2 Valve Normally Closed)
		วาล์วควบคุมทิศทางแบบ 2/2 ปกติเปิด (2/2 Valve Normally Open)
2.		วาล์วควบคุมทิศทางแบบ 3/2 ปกติปัด (3/2 Valve Normally Closed)
		วาล์วควบคุมทิศทางแบบ 3/2 ปกติเปิด (3/2 Valve Normally Open)
3.		วาล์วควบคุมทิศทางแบบ 3/3 ตำแหน่งกลางปิด (3/3 Valve Mid-Position Closed)
4.		วาล์วควบคุมทิศทางแบบ 4/2 (4/2 Valve)
5.		วาล์วควบคุมทิศทางแบบ 4/3 ตำแหน่งกลางปิด (4/3 Valve Mid-Position Closed)

#### ตารางสัญลักษณ์ของวาล์วควบคุมทิศทาง

#### ตารางสัญลักษณ์ของอุปกรณ์ส่งลมอัด

	ลำดับ	สัญลักษณ์	ความหมาย
	6.		วาล์วควบคุมทิศทางแบบ 4/3 ตำแหน่งกลางเร่งระบายลม (4/3 Valve Mid-Position Exhausted)
	7.		วาล์วควบคุมทิศทางแบบ 5/3 (5/3 Valve)
1	8.		วาล์วควบคุมทิศทางแบบ 5/3 ตำแหน่งกลางปิด (5/3 Valve Mid-Position Closed)
10	9.		วาล์วควบคุมทิศทางแบบ 5/3 ตำแหน่งกลางเร่งระบายลม (5/3 Valve Mid-Position Exhausted)

ลำดับ	สัญลักษณ์	ความหมาย
1.	o—	ท่อทางที่มาจากแหล่งจ่ายลม
2.		ท่อลมที่มีลมเข้าปกติ
3.		ท่อลมที่เป็นสัญญาณสั่ง
4.		ท่อระบายลมทิ้ง
5.		เส้นกรอบอุปกรณ์ที่มีเส้นอุปกรณ์หลายตัวอยู่ในชุด เดียวกัน
6.	$\checkmark$	ท่อลมชนิดยึดหยุ่นตัวได้
7.		จุดต่อลมสามทาง จุดต่อลมสี่ทาง

#### ตารางสัญลักษณ์ของวาล์วควบคุมทิศทาง

	ลำดับ	สัญลักษณ์	ความหมาย
	8.		ท่อลมข้ามกัน(ไม่ใช่จุดต่อ)
1	9.	•X	ท่อลมที่ถูกอุด
	10.	•X~_•	ข้อต่อลม
	11.	• <del>-</del> +•	ข้อต่อลมชนิดต่ออย่างรวดเร็ว
Sec. Sol	12.	•->+(>-•	ข้อต่อลมซนิดต่ออย่างรวดเร็ว มีวาล์วกันกลับ

#### ตารางสัญลักษณ์และความหมายของกลไกบังคับวาล์ว

1. การเลื่อนสิ้นโดยใช้ไฟฟ้า (Electrical Actuator)					
			ใช้ขดลวดแม่เหล็กไฟฟ้า 1 ชุด ทำให้วาล์วเลื่อน		
			ใช้ขดลวดแม่เหล็กไฟฟ้า 2 ชุด ทำให้วาล์วเลื่อน		
			ใช้ขดลวดแม่เหล็กไฟฟ้ามากกว่า 1 ชุด ทำงาน		
			ทิศทางตรงกันข้ามเพื่อให้วาล์วเลื่อน		
	2. การเลื่อนลิ้นโดยใช้มือหรือเท้า (Manual Actuator)				
	۴_	General	ใช้กล้ามเนื้อในการเลื่อน (บุ๋มกด) แบบทั่วไป		
	Ē	Push button	ใช้มือกด (ปุ่มกด)		
	₽	Lever	ใช้คันโยกหรือมือบิด		
	거	Padle	ใช้เท้าเหยียบ		
	<u>گ</u> ے	Detent	ใช้คันโยกหรือมือบิดแบบล็อกตำแหน่ง		



#### ตารางสัญลักษณ์และความหมายของกลไกบังคับวาล์ว

3. การเลื่อนลิ้นโดยใช้ระบบไก (Mechanical Actuator)			
	Plunger	ใช้กลไกภายนอกเป็นตัวกด ทำงานสองทาง	
	Spring	ใช้สบริงดันวาล์วให้กลับตำแหน่งปกติ	
¢⊂	Roller lever	ใช้ล้อลูกกลิ้งเป็นตัวกด ทำงานได้สองทาง	
₩.	Roller lever with idle return (Roller Trip)	ใช้ล้อลูกกลิ้งเป็นตัวกด ทำงานได้ทางเดียว	
_4. การเลื่อนลิ้นโดยใ	ชัลม (Pneumatic Actuator)		
	Direct by Pressure (Pilot)	ใช้ลมในการเลื่อนวาล์วโดยตรง	
	Direct by Pressure Relief	ระบายลมให้วาล์วเลื่อนโดยตรง	
	- Differential Pressure	เลื่อนลิ้นด้วยความดันที่แตกต่าง	
	Indirect by Pressure (Pilot)	ใช้ลมไปเลื่อนวาล์วทางอ้อม	
	Indirect by Pressure Relief	ระบายจากวาล์วทางอ้อม	
5. การเลื่อนลิ้นแบบ	ผสม (combined Actuation)		
	Push botton and Pressure	ใช้ปุ่มกดและใช้ลมช่วย	
₽ E	Push botton or Pressure	ใช้ปุ่มกดหรือใช้ลมช่วย	
	Roller lever and Pressure	ใช้กลไกลูกกลิ้งและลมช่วยเลื่อนวาล์ว	
	Solenoid and Pressure	ใช้ขดลวดแม่เหล็กไฟฟ้าและลมช่วยเลื่อนวาล์ว	
<u>w</u> <u>]</u> w	Spring Centred	ใช้สบริงดันวาล์วให้อยู่ในตำแหน่งกลาง	

การควบคุมระบบนิวเมติกส์โดยใช้โปรแกรมเมเบิลคอนโทรลเลอร์

ในการควบคุมระบบนิวเมติกส์ไฟฟ้า สามารถแบ่งส่วนประกอบที่สำคัญของระบบนิวเมติกส์ไฟฟ้า ออกได้เป็นสองส่วนด้วยกัน คือวงจร นิวเมติกส์หรือวงจรกำลัง (Pneumatic Circuit) และวงจรไฟฟ้าหรือ วงจรควบคุม (Electric Circuit)



ในการนำ PLC ไปควบคุมระบบนิวเมติกส์คือ การนำสัญญาณจากเอาต์พุตของ PLC ไปต่อเข้ากับ ส่วนควบคุมการทำงานในระบบนิวเม ติกส์นั่นคอวาล์วควบคุมการทำงาน ซึ่งทำงานโดยใช้คอยล์ไฟฟ้าหรือที่ เรียกกันว่า โซลินอยล์วาล์ว เพื่อให้สามารถรับสัญญาณไฟฟ้าจากเอาต์ พุตของ PLC ได้และเปลี่ยน วงจรไฟฟ้าหรือวงจรควบคุม (Electric Circuit) เป็นโปรแกรมควบคุม (Ladder Diagram) เพื่อป้อนเข้าไป ใน PLC โดยที่วงจรนิวเมติกส์หรือวงจรกำลัง (Pneumatic Circuit) ยังคงเหมือนเดิม



ตัวอย่างขั้นตอนในการปฏิบัติการนำ PLC ไปควบคุมระบบนิวเมติกส 1. กำหนดตำแหน่งอินพุต/เอาต์พุต (Input/Output)

อินพุต (Input)	เอาต์พุต (Output)
Switch Start : 10.0	Solenoid valve (Y1) :Q0.0

2. แปลงวงจรจากวงจรไฟฟ้า (Electric Circuit) มาเป็นแลดเดอร์ไดอะแกรม (Ladder)



- 3. ป้อนโปรแกรมตาม Ladder Diagram โหลดลงใน PLC
- 4. ต่อสายควบคุมทางด้านอินพุต/เอาต์พุต (Input/Output)



์ตัวอย่างการควบคุมระบบนิวเมติกส์โดยใช้โปรแกรมเมเบิลคอนโทรลเลอร์

ตัวอย่างการควบคุมกระบอกสูบสองทางใช้โซลินอยล์วาล์วแบบ 5/2 Single acting solenoid valve 1. เงื่อนไขการทำงาน

เมื่อกด Switch Start ส่งผลให้ โซลินอยล์วาล์ว Y1 ทำงาน และค้างสภาวะการทำงานไว้ตลอดเวลา เมื่อ โซลินอยล์วาล์ว Y1 ทำงาน ส่งผลให้กระบอกสูบเคลื่อนที่ออกและเมื่อกด Switch Stop จะทำให้โซลินอยล์วาล์ว Y1 หยุดทำงาน เมื่อโซลินอยล์วาล์ว Y1 หยุดทำงาน ส่งผลให้กระบอกสูบเคลื่อนที่กลับตำแหน่งเดิม 2. กำหนดตำแหน่งอินพุต/เอาต์พุต (Input/Output)



อินพุต (Input)	เอาต์พุต (Output)
Switch Start : I0.0	Solenoid valve (Y1) :Q0.0
Switch Stop : I0.1	

#### 3. เขียน Ladder Diagram



4. 4. ต่ออุปกรณ์ทางด้านอินพุต/เอาต์พุต (Input/Output)



ตัวอย่างการควบคุมกระบอกสูบสองทางโดยใช้โซลินอยล์วาล์วแบบ 5/2 Double acting solenoid valve 1. เงื่อนไขการทำงาน

เมื่อกด Switch Startจะทำให้โซ ลินอยล์วาล์ว Y1 ทำงาน Y2 หยุดทำงาน ส่งผลให้กระบอกสูบเคลื่อนที่ ออก ตัวตั้งเวลาจะตั้งเวลา 2 วินาทีโซลินอยล์วาล์ว Y2 จะทำงาน โซลิ นอยล์วาล์ว Y1 หยุดทำงาน ส่งผลให้กระบอกสูบเคลื่อนที่กลับ ตัวตั้งเวลา จะตั้งเวลาอีก 2 วินาทีให้โซลินอยล์วาล์ว Y1 ทำงาน โซลินอยล์วาล์ว Y2 หยุดทำงานอีกครั้ง และวนการทำงานอย่างนี้ ไปเรื่อย ๆ จนกว่าจะกด Switch Stop จะทำให้โซลินอยล์วาล์ว Y1 และ Y2 หยุดทำงาน

2. กำหนดตำแหน่งอินพุต/เอาต์พุต (Input/Output)



อินพุต (Input)	เอาต์พุต (Output)
Switch Start : I0.0	Solenoid valve (Y1) :Q0.0
Switch Stop: I0.1	Solenoid valve (Y2) : Q0.1

#### 3. เขียน Ladder Diagram



ตัวอย่างการควบคุมกระบอกสูบสองทางโดยใช้โซลินอยล์วาล์วแบบ 5/2 Single acting solenoid valve จำนวน 2 ตัว



1. เงื่อนไขการทำงาน

กด Switch Start ส่งผลให้โซลินอยล์วาล์ว Y1 ทำงานและค้าง สภาวะการทำงานไว้เมื่อโซลินอยล์วาล์ว Y1 ทำงาน กระบอกสูบ A จะ เคลื่อนที่ออก Timer (TON) จะตั้ง เวลา 2 วินาทีเมื่อเวลาครบส่งผลให้โซลินอยล์วาล์ว Y2 ทำงาน กระบอกสูบ B จะเคลื่อนที่ออก เมื่อกด Switch Stop ส่งผลให้โซลินอยล์วาล์ว Y1 และ Y2 หยุดทำงาน กระบอกสูบ A และ B จะเคลื่อนที่กลับ ตำแหน่งเดิม

#### 2. กำหนดตำแหน่งอินพุต/เอาต์พุต/ตัวตั้งเวลา (Input/Output/Timer)

อินพุต (Input)	เอาต์พูต (Output)	ตัวตั้งเวลา(Timer)
Switch Start : 10.0	Solenoid valve (Y1) : Q0.0	ตัวตั้งเวลาตัวที่ 1 (TON) : T101
Switch Stop : I0.1	Solenoid valve (Y2) : Q0.1	

3. 3. เขียน Ladder Diagram



4. ต่ออุปกรณ์ทางด้านอินพุต/เอาต์พุต (Input/Output)





### ตัวอย่างการออกแบบโปรแกรม PLC

# และการประยุกต์ใช้งาน


การออกแบบโปรแกรมจาก Timing diagram

Timing Diagram คือ ผังแสดงการทำงานของสัญญาณต่าง ๆ เทียบกับคาบเวลา โดยเขียนด้วยเส้น หรือสัญลักษณ์แสดงความหมาย

ต่าง ๆ แทนการอธิบายด้วยข้อความตัวอักษร ประโยชน์ที่ได้จากผังแสดง ขั้นตอนการทำงานดังกล่าวคือ

 สามารถมองเห็นสภาวะการทำงานของอินพตุ /เอาต์พุต ทุกจุดในช่วงเวลาต่างๆของลำดับขั้นการ ควบคุม โดยการมองภาพรวม ๆเพียง ครั้งเดียว

2. แสดงให้เห็นการทำงานแบบต่อเนื่องได้ง่าย ๆ ถึงสภาวะการทำงานของเอาต์พุตตัวหนึ่งไปยัง เอาต์พุตอีกตัวหนึ่ง

## ความหมายของเส้นและสัญลักษณ์ที่ใช้ในการเขียน Timing diagram



sustrial and Community



สัญลักษณ์แสดงการทำงานของสัญญาณต่าง ๆ ตำแหน่ง 1 แสดงถึง สภาวะไม่ทำงาน ตำแหน่ง 2 แสดงระยะที่กำลังการเปลี่ยนสภาวะ จากไม่ทำงาน (Off) เป็นสภาวะทำงาน (On) ตำแหน่ง 3 แสดงระยะที่กำลังการเปลี่ยนสภาวะ จากทำงาน(On) เป็น สภาวะไม่ทำงาน (Off) ตำแหน่ง 4 แสดงการกลับสู่สภาวะเดิมคือไม่ทำงาน



หากนำสัญลักษณ์แสดงการทำงานของสัญญาณต่าง ๆ มาใช้แทนตำแหน่งของอินพุต (I) และเอาต์พุต (Q) จะมีลักษณะการทำงานดังนี้

Input (I) เปรียบเทียบเหมือนการทำงานของสวิตซ์ที่มีลักษณะการ ทำงานคือ กด – ปล่อย ตำแหน่ง 1 สภาวะเริ่มต้น ยังไม่มีการกดสวิตซ์ใด ๆ ตำแหน่ง 2 กดสวิตซ์ ตำแหน่ง 3 ปล่อยสวิตซ์ ตำแหน่ง 4 การกลับสู่สภาวะเดิม สภาวะไม่ทำงาน



เอาต์พุต (Q) เปรียบเทียบเหมือนการทำงานของหลอดไฟที่มีลักษณะ การทำงานคือ ติด – ดับ ตำแหน่ง 1 สภาวะไม่ทำงาน ตำแหน่ง 2 หลอดไฟติด ตำแหน่ง 3 หลอดไฟดับ ตำแหน่ง 4 การกลับสู่สภาวะเดิม คือสภาวะไม่ทำงาน



เมื่อนำสัญลักษณ์แสดงการทำงานของสัญญาณต่าง ๆ มาเขียนให้มีความเกี่ยวเนื่องกัน จาก Timing diagram อธิบายขั้นตอนการทำงานของ อินพุตและเอาต์พุต ณ จุดเวลาหรือตำแหน่งนั้น ๆ ได้ ดังนี้

ตำแหน่ง 1 ไม่มีการทำงานใด ๆเนื่องจาก อินพุต 10.0 ยังไม่ทำงาน ทำให้เอาต์พุต Q0.0 ไม่ทำงาน ตำแหน่ง 2 อินพุต 10.0 ทำงาน ส่งผลให้เอาต์พุตQ0.0 ทำงาน ตำแหน่ง 3 อินพุต 10.0 หยุดทำงาน ส่งผลให้เอาต์พุต Q0.0 หยุดทำงาน ตำแหน่ง 4 กลับสู่สภาวะเดิม



จาก Timing diagram อธิบายขั้นตอนการทำงานของอินพุตและเอาต์พุต ได้ดังนี้ ตำแหน่ง 1 ไม่มีการทำงานใด ๆเนื่องจาก อินพุต 10.0 ไม่ทำงาน ทำให้เอาต์พุต Q0.0 ไม่ทำงาน ตำแหน่ง 2 อินพุต 10.0 ทำงาน ส่งผลให้เอาต์พุต Q0.0 ทำงาน

ตำแหน่ง 3 อินพุต 10.0 หยุดทำงาน แต่เอาต์พุต Q0.0 ไม่หยุดการทำงานยังคงทำงานต่อ

จาก Timing diagram อธิบายขั้นตอนการทำงานของอินพุตและเอาต์พุต ได้ดังนี้

ตำแหน่ง 1 อินพุต I0.0 ทำงาน ส่งผลให้เอาต์พุต Q0.0 ทำงาน ตำแหน่ง 2 อินพุต I0.0 หยุดทำงานแต่เอาต์พุต Q0.0 ยังคงทำงาน

ตำแหน่ง 3 อินพุต 10.1 ทำงาน ส่งผลให้เอาต์พุต Q0.0 หยุดการทำงาน

ตำแหน่ง 4 อินพุต I0.1 หยุดทำงาน กลับสู่สภาวะเดิม





ตัวอย่างการออกแบบ Ladder Diagram จาก Timing Diagram

ตัวอย่างเรื่องโปรแกรม Timing diagram แบบเรียงลำดับ การทำงานของโปรแกรม

เมื่อกด Switch Start : I0.0 ให้ Output 1 : Q0.0 ทำงานหลังจากนั้น 1.5 วินาทีให้ Output 2 : Q0.1 ทำงาน กด Switch Stop : I0.1 ให้ Output 1 : Q0.0 และ Output 2 : Q0.1 หยุดทำงาน

Timing diagram

กำหนดตำแหน่งอินพุต/เอาต์พุต/ตัวตั้งเวลา (Output/Input/Timer)

อินพุต (Input)	เอาต์พุต (Output)	ตัวตั้งเวลา (Timer)
Switch Start : 10.0	Output 1 : Q0.0	ตัวตั้งเวลาตัวที่ 1 (TON) : T101
Switch Stop : I0.1	Output 2 : Q0.1	



Ladder Diagram



ตัวอย่างที่ 11.2 เรื่องโปรแกรม Timing diagram แบบสลับการทำงาน การทำงานของโปรแกรม

เมื่อกด Switch Start : I0.0 ให้ Output 1 : Q0.0 ทำงานหลังจากนั้น 1 วินาทีให้ Output 2 : Q0.1 ทำงาน กด Switch Stop : I0.1 ให้ Output 1 : Q0.0 และ Output 2 : Q0.1 หยุดทำงาน



Timing diagram

กำหนดตำแหน่งอินพุต/เอาต์พุต/ตัวตั้งเวลา (Output/Input/Timer)

อินพุต (Input)	เอาต์พุต (Output)	ตัวตั้งเวลา (Timer)
Switch Start: I0.0	Output 1: Q0.0	ตัวตั้งเวลาตัวที่ 1 (TON): T105
Switch Stop: 10.1	Output 2 :Q0.1	ตัวตั้งเวลาตัวที่ 2 (TON): T106
	Output 3 : Q0.2	

## Ladder Diagram



การออกแบบโปรแกรมจากขั้นตอนการทำงาน

การออกแบบโปรแกรมจากขั้นตอนการทำงานแบบต่างๆเป็นการนำตัวอย่างของงานในลักษณะการ ควบคุมแบบกึ่งอัตโนมัติและ

อัตโนมัติที่มีใช้ตามอาคาร สถานที่และในโรงงานอุตสาหกรรม เพื่อฝึกให้ผู้เรียน ได้ออกแบบโปรแกรม สามารถประยุกต์ใช้งาน PLC ได้

ขั้นตอนในการออกแบบโปรแกรม

- 1. ให้ศึกษาขั้นตอนการทำงานของโปรแกรม
- 2. การกำหนดอินพุต / เอาต์พุต
- 3. ออกแบบ Ladder Diagram ตามขั้นตอนการทำงาน

ตัวอย่างการออกแบบ Ladder Diagram จากขั้นตอนการทำงาน

ตัวอย่างเรื่องการออกแบบโปรแกรมควบคุมมอเตอร์สายพานลำเลียง



1. การทำงานของโปรแกรม

(1) เมื่อ On สวิตช์ I0.0 ให้เอาต์พุต Q0.0 (มอเตอร์สายพานลำเลียง 1) มีสภาวะ On

(2) เมื่อเอาต์พุต Q0.0 (มอเตอร์สายพานลำเลียง 1) มีสภาวะ On จะทำให้วัตถุหรือ สิ่งของ เคลื่อนที่ไปตามสายพานหากวัตถุหรือ สิ่งของ ที่เคลื่อนที่ไปตามสายพานเคลื่อนที่ไม่ถึงตำแหน่งของ อินพุต I0.2 (เซนเซอร์ (Sensor) 1 ตรวจเช็คตำแหน่ง) มีสภาวะ Off เป็นระยะ เวลาเกิน 5 วินาทีให้เอาต์พุต Q0.0 (มอเตอร์สายพานลำเลียง1) มีสภาวะ Off (3) เมื่อเอาต์พุต Q0.0 (มอเตอร์สายพานลำเลียง1) มีสภาวะ On จะทำให้วัตถุหรือ สิ่งของ เคลื่อนที่ไปตามสายพาน จนถึงตำแหน่ง ของอินพุต I0.2 (เซนเซอร์ 1 ตรวจเช็คตำแหน่ง) เมื่ออินพุต I0.2 มีสภาวะ On (ตรวจเช็คพบวัตถุหรือสิ่งของ) จะสั่งงานให้เอาต์พุต Q0.1 (มอเตอร์สายพานลำเลียง 2) มีสภาวะ On

(4) เมื่อเอาต์พุต Q0.1 (มอเตอร์สายพานลำเลียง 2) มีสภาวะ On จะทำให้วัตถุหรือ สิ่งของเคลื่อนที่ไปตามสายพานที่ 2 จนถึงอินพุต I0.3 (เซนเซอร์ 2 ตรวจเช็คตำแหน่ง) เมื่ออินพุต I0.3 มี สภาวะ On (ตรวจเช็คพบวัตถุหรือสิ่งของ) จะสั่งงานให้เอาต์พุต Q0.0 และ Q0.1 (มอเตอร์สายพานลำเลียง 1 และมอเตอร์สายพานลำเลียง 2) มีสภาวะ Off

(5) เมื่อ On สวิตช์อินพุต 10.1 ให้หยุดการทำงานทั้งหมด

## 2. กำหนดตำแหน่ง อินพุต/เอาต์พุต (Input/Output)

อินพุต (Input)	เอาต์พูต (Output)
Switch Start : 10.0	มอเตอร์สายพานลำเลียง 1 : Q0.0
Switch Stop : I0.1	มอเตอร์สายพานลำเลียง 2 : Q0.1
Sensor 1ตรวจเช็คตำแหน่ง: I0.2	
Sensor 2ตรวจเช็คตำแหน่ง : 10.3	

3. เขียน Ladder Diagram



ตัวอย่างเรื่องการออกแบบโปรแกรมควบคุมลานจอดรถอัตโนมัติ



1. การทำงานของโปรแกรม

(1) เมื่อ On สวิตช์อินพุต 10.0 จะเริ่มการทำงานของลานจอดรถอัตโนมัติเมื่อ Sensor 1 ทำงาน จะสั่งงานให้ Motor1 ทำงานเปิดประตูไม้กั้นทางเข้า
ระตูไม้กั้นทางเข้า และหลังจาก Sensor1 หยุดทำงาน 5 วินาที จะสั่งงานให้ Motor1 หยุดทำงานปิดประตูไม้กั้นทางเข้า
(2) เมื่อ Sensor 2 ทำงาน จะสั่งงานให้ Motor2 ทำงานเปิดประตูไม้กั้นทางออก และ หลังจาก Sensor2 หยุดทำงาน 5 วินาทีจะ สั่งงานให้ Motor2 หยุดทำงานปิดประตูไม้กั้นทางออก และ หลังจาก Sensor2 หยุดทำงาน 5 วินาทีจะ สั่งงานให้ Motor2 หยุดทำงานปิดประตูไม้กั้นทางออก และ หลังจาก Sensor2 หยุดทำงาน 5 วินาทีจะ

(3) ช่วงจังหวะ Sensor1 หยุดทำงาน จะส่งผลให้ Counter1 นับขึ้น

(4) ช่วงจังหวะ Sensor2 หยุดทำงาน จะส่งผลให้ Counter1 นับลง

(5) เมื่อจำนวนรถในลานจอดรถครบ 40 คัน จะทำให้ป้ายไฟแสดงรถเต็มทำงาน สั่งห้าม ไม่ให้เปิดประตูไม้กั้นทางเข้า

(6) เมื่อ On สวิตช์อินพุต 10.1 ให้ระบบหยุดการทำงานทั้งหมด

2. กำหนดตำแหน่ง อินพุต (Input) / เอาต์พุต (Output)

อินพด (Input)	เอาต์พต (Output)
9 1 1 /	9 (5 1 7
Switch Start : 10.0	ป้ายไฟแสดงรถเต็ม: Q0.0
Switch Stop : I0.1	Motor1 ไม้กั้นทางเข้า: Q0.1
Sensor 1 : 10.2	Motor2 ไม้กั้นทางเข้า: Q0.2
Sensor 2 : 10.3	

and Commun

## 3. เขียน Ladder Diagram



