

## บทที่ 7

### ออกแบบการควบคุมหุ่นยนต์อุตสาหกรรมด้วย PLC

#### หัวข้อเรื่อง

1. แนะนำบอร์ด PLC รุ่น CFX1N-10MR 6IN/4 OUT
2. การเริ่มต้นใช้งานโปรแกรม GX Developer
3. การเชื่อมต่อและเขียนโปรแกรมลง PLC
4. คำสั่งพื้นฐานในการออกแบบโปรแกรม
5. การเขียนโปรแกรมควบคุมหุ่นยนต์อุตสาหกรรมด้วยบอร์ด PLC

#### สาระสำคัญ

แนะนำพื้นฐานในส่วนการใช้งาน PLC จะประกอบด้วย 3 ส่วนหลัก

1. PLC
2. Link cable หรือ สายลิงค์ PLC หรือ สาย Download Program
3. Computer พร้อมกับตัวโปรแกรมในการเขียน LADDER DIAGRAM

ส่วนประกอบของอุปกรณ์ ที่จะมา Test พวกสายไฟ , ปุ่มกด , LAMP ต่างๆ และ ก็ Power Supply และปรกติในการใช้งานกับ PLC เราจะเขียนโปรแกรมบนคอมพิวเตอร์ และก็ดาวน์โหลดผ่านสายลิงค์ ลงไปยังตัว PLC และก็การใช้งานก็คือปล่อยให้ PLC ทำงานแบบ Standalon

#### การเชื่อมต่อ PLC

ในส่วนของ FX1S จะเป็น Family ของ PLC ยี่ห้อ Mitsubishi จะเป็นตระกูลของ FXseries จะมีตั้งแต่ FX1S ,FX1N,FX2N,FX2NC,FX3U ก็จะมี Feature สเปกที่ต่างกันออกไป FX1S จะเป็นรุ่นที่มี Specification ที่ต่ำที่สุดในบรรดา FX series

## สมรรถนะประจำหน่วยการเรียนรู้

แสดงความรู้การใช้บอร์ด PLC รุ่น CFX1N-10MR 6IN/4 OUT ควบคุมหุ่นยนต์อุตสาหกรรม

## จุดประสงค์การเรียนรู้

จุดประสงค์ทั่วไป

1. เพื่อให้มีความรู้ความเข้าใจคุณสมบัติของบอร์ด PLC รุ่น CFX1N-10MR 6IN/4 OUT
2. เพื่อให้มีความรู้ความเข้าใจการติดตั้งและใช้งานโปรแกรม GX Developer
3. เพื่อให้สามารถนำความรู้ไปประยุกต์ออกแบบหุ่นยนต์อุตสาหกรรม
4. เพื่อให้ตระหนักถึงความสำคัญของใช้โปรแกรม PLC ควบคุมหุ่นยนต์อุตสาหกรรม

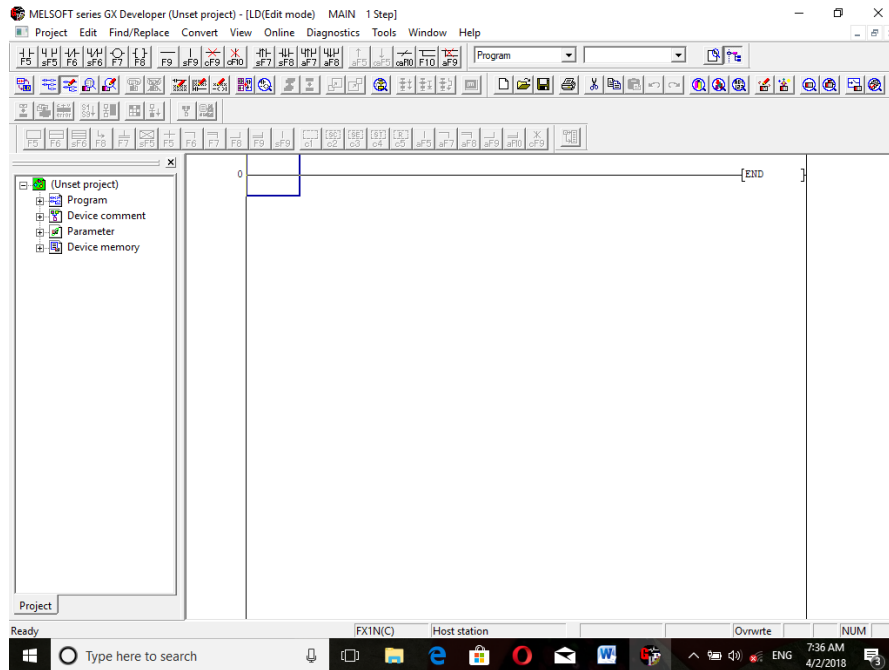
## 2. การเริ่มต้นใช้งานโปรแกรม GX Developer

การเริ่มต้นใช้งาน GX Developer โดยเริ่มต้นเข้าสู่โปรแกรม GX Developer โดยคลิกปุ่มstart

>Program>MELSOFT Application>DX Developer



รูปที่ 7.1 การเข้าสู่โปรแกรม GX Developer



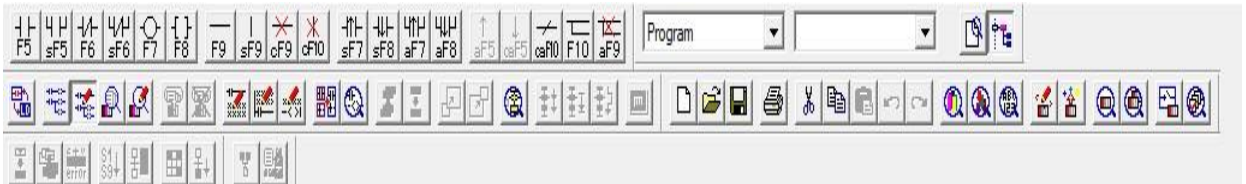
รูปที่ 7.2 องค์ประกอบในหน้าจอโปรแกรม GX Developer

หน้าจอโปรแกรม GX Developer จะประกอบด้วยส่วนที่สำคัญ 4 ส่วนคือ

1. Project data list
2. Toolbar
3. Status bar
4. Program Mode Area

**Project data list** เป็นส่วนที่ทำหน้าที่ในการแสดงรายละเอียดของโปรแกรม โดยแยกรายละเอียดออกเป็นส่วนของโปรแกรมที่ออกแบบ (Program), ส่วนที่แสดงรายละเอียดของอุปกรณ์(Device Comment) และส่วนที่แสดงรายละเอียดของอุปกรณ์ใน PLC รุ่นที่เลือกใช้ (PLC Parameter)

**Toolbar** เป็นส่วนที่นำเสนอปุ่มแสดงคำสั่งที่ใช้ในการทำงานออกแบบโปรแกรม PLC ในรูปแบบต่างๆ ดังเช่นจากรูปที่ 7.2 Toolbar Program เป็น Toolbar ที่มีคำสั่งที่ใช้ในการจัดการโปรแกรม Write Mode, Read Mode, Monitor หรือ Toolbar Ladder symbols ที่มีคำสั่งที่ใช้ในการออกแบบโปรแกรมแลตเตอร์ เช่น หน้าสัมผัสแบบปิด NC หน้าสัมผัสแบบเปิด NC เป็นต้น

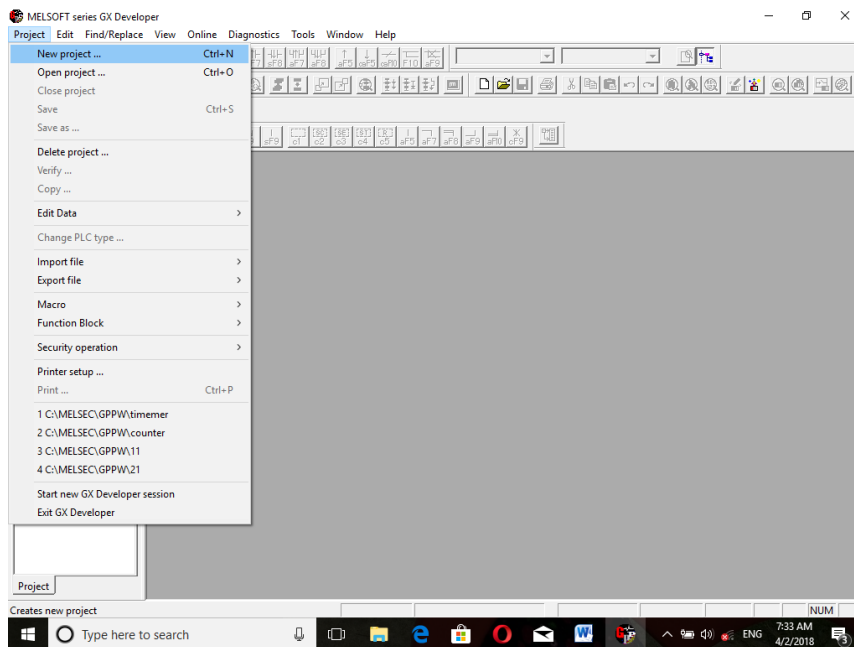


รูปที่ 7.3 เครื่องมือต่างๆที่ใช้ในโปรแกรมใน Toolbar

Program Mode Area เป็นพื้นที่ที่ใช้ในการออกแบบโปรแกรมตามลักษณะที่ผู้ต้องการในการออกแบบโปรแกรม ซึ่งสามารถเลือกรูปแบบการออกแบบ เช่น การออกแบบโปรแกรม Ladder, Instruction List เป็นต้น

การสร้างโปรแกรมใหม่เมื่อเปิดโปรแกรม GX Developer ผู้ใช้จะต้องเริ่มสร้างโปรแกรมใหม่ เพื่อเป็นการกำหนดให้โปรแกรมทราบว่า ผู้ใช้ต้องการออกแบบโปรแกรมสำหรับนำไปใช้กับ PLC รุ่นใด โปรแกรมจะนำค่าพารามิเตอร์และอุปกรณ์มาให้ผู้ใช้สามารถใช้ในการออกแบบโปรแกรม

การสร้างโปรแกรมใหม่ทำได้โดยการคลิกเมนู Project ใน Menu Bar และคลิกเลือก New Project ดังรูปที่ 7.4



รูปที่ 7.4 การเลือกคำสั่งสร้างโปรแกรม (New Project)

การกำหนดเลือกรุ่น PLC ทำได้โดยหลังจากเลือกคำสั่ง New Project แล้ว โปรแกรมจะแสดงหน้าจอ New Project ขึ้นมาให้ผู้ใช้เลือกชนิดและรุ่นของ PLC โดยรายละเอียดของหน้าจอ New Project ประกอบด้วย

1. PLC Series สำหรับให้ผู้ใช้เลือกรุ่นของ PLC Mitsubishi ที่จะใช้ในการเขียนโปรแกรม ซึ่งสามารถเลือกได้ ดังนี้ QCPU(Q-Mode),QCPU(A-Mode),QnACPU,ACPU,MOTION(SCPU), FXCPU,CNC(M6/M7)

2. PLC type สำหรับเลือกชนิดของ PLC แต่ละรุ่น เช่นใน FXCPU สามารถเลือกรุ่นได้ดังนี้ FX0(S),FX0N,FX1,FX1S,FX1N(C),FXU/FX2C,FX2N(C),FX3U(C)

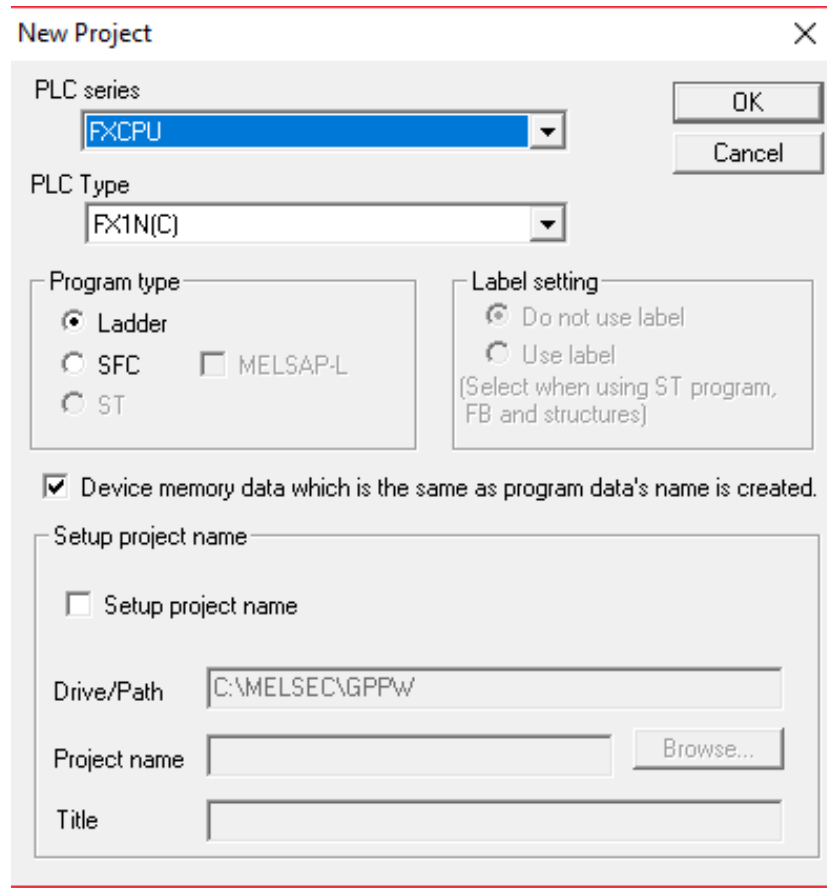
3. Program Type สำหรับการเลือกใช้ภาษาในการเขียนโปรแกรม PLC ซึ่งสามารถเลือกเขียนได้ 2 ภาษา คือ Ladder,SFC

4. Setup Project name สำหรับใช้ในการตั้งชื่อโปรเจค โดยคลิกเครื่องหมายถูกหน้า Setup Project name

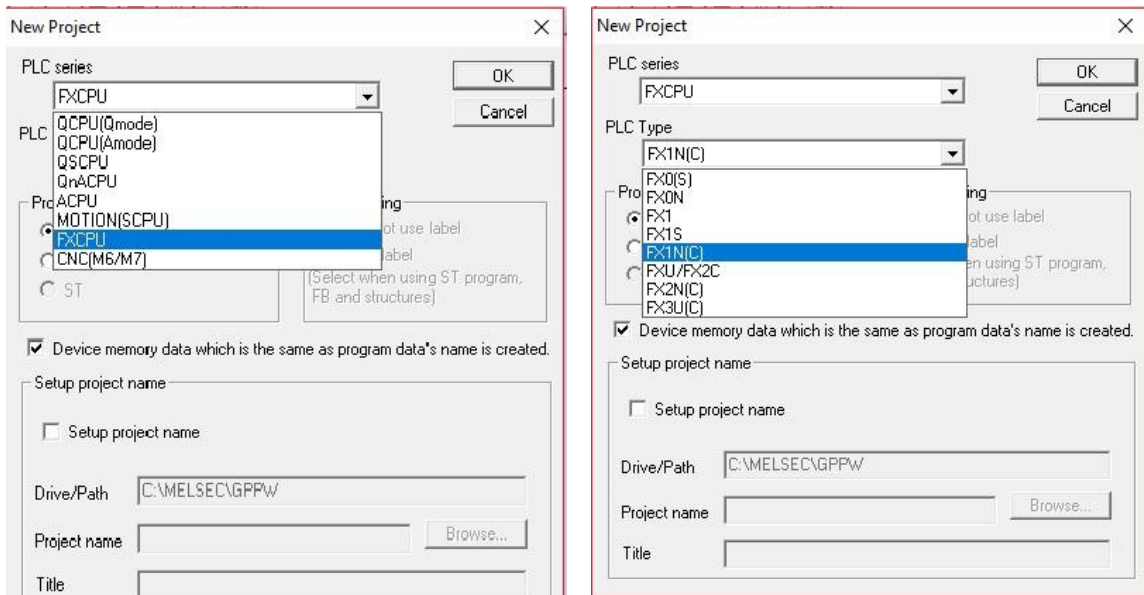
5. Drive/Path สำหรับเลือก Drive และโฟลเดอร์ที่ต้องการจัดเก็บโปรแกรมที่ได้จากการออกแบบลงใน Hard Disk

6. Project name สำหรับตั้งชื่อโปรเจคตามต้องการ

7. Title สำหรับใส่ชื่อหัวข้อแต่ละโปรเจค



รูปที่ 7.5 หน้าจอ New Project



รูปที่ 7.6 การเลือกชนิดและรุ่นของ PLC ในหน้าจอ New Project

### 3. การเชื่อมต่อและเขียนโปรแกรมลง PLC

มีขั้นตอนในการทำดังนี้

#### 3.1 การเตรียมความพร้อมก่อนการใช้งานคอมพิวเตอร์กับ PLC

การเตรียมความพร้อมก่อนการใช้งานเริ่มจาก

3.1.1 ตรวจสอบแหล่งจ่ายไฟ (Power Supply) ของ PLC ต้องมีการจ่ายไฟให้กับขั้ว I/O

3.1.2 ตรวจสอบวงจร Input และ Output และการต่อสายว่าถูกต้องตามที่ออกแบบไว้

หรือไม่

3.1.3 เขียนโปรแกรมลงใน PLC สั่งงานให้ PLC เข้าสู่ Mode Run สังเกตว่าเกิด Error

ขึ้นหรือไม่

3.1.4 ตรวจสอบการทำงานของโปรแกรมด้วยการสั่งงาน Monitor ของโปรแกรม GX Developer เพื่อตรวจสอบการทำงานว่าถูกต้องตามที่ออกแบบไว้หรือไม่

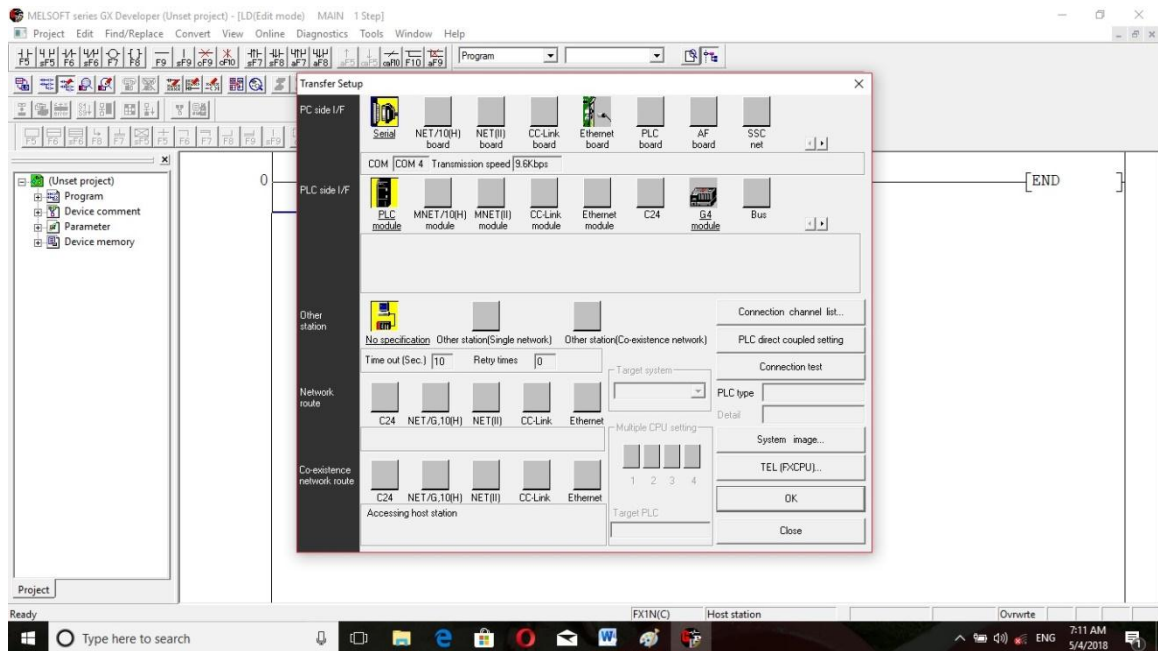
#### 3.2 การเชื่อมต่อระหว่างคอมพิวเตอร์และ PLC

ในการเชื่อมต่อการทำงานระหว่างคอมพิวเตอร์และ PLC มีขั้นตอนการปฏิบัติดังนี้

3.2.1 จ่ายกระแสไฟจาก Power Supply ให้กับ PLC

3.2.2 ต่อสายเชื่อมต่อระหว่างคอมพิวเตอร์และ PLC เพื่อใช้ในการส่งถ่ายข้อมูลที่ออกแบบ (Transfer) โดยจะมีทั้ง แบบ Serial และแบบ USB Port

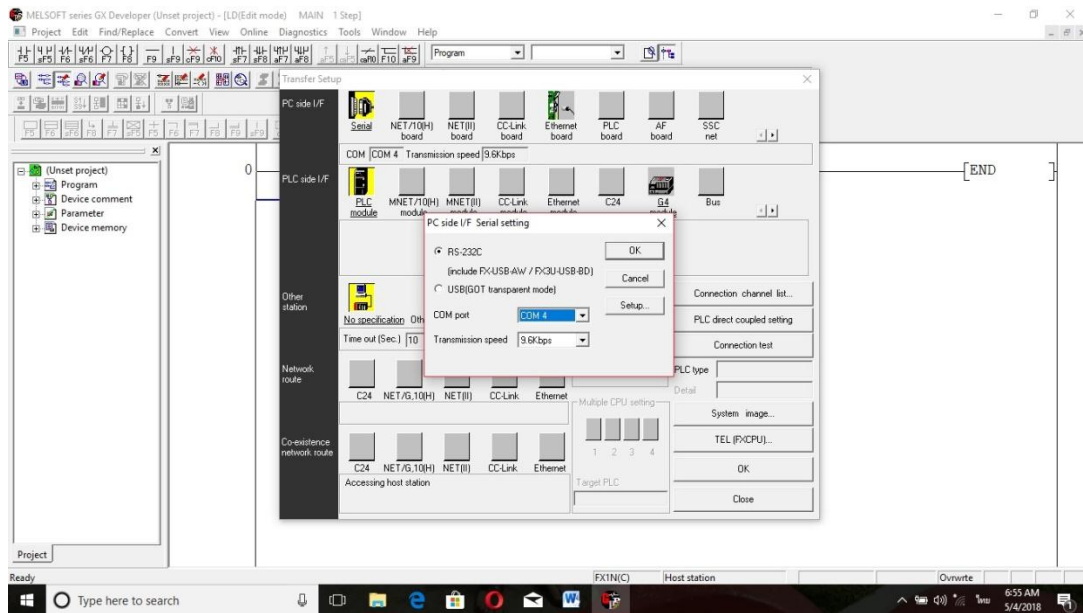
3.2.3 ปรับสวิตช์ PLC ให้อยู่ในตำแหน่ง PLC Stop Mode



รูปที่ 7.7 การเข้าสู่คำสั่ง Transfer Setup เพื่อตรวจสอบการเชื่อมต่อ

3.2.4 เลือกช่องทางการติดต่อสื่อสารระหว่างคอมพิวเตอร์และ PLC ตามที่ได้เชื่อมต่อไว้ โดยคลิกเมนู Online>Transfer ...หน้าต่าง Transfer Setup จะแสดงขึ้นมา คลิกเลือก Serial USB ในส่วน PC/side I/F หน้าต่าง PC side I/F Serial setting จะแสดงขึ้นมา ให้คลิกเลือกชนิดการเชื่อมต่อ RS-232C หรือ USB หากเป็นชนิด RS-232C ต้องกำหนด COM Port และความเร็วการรับ – ส่งข้อมูล ของคอมพิวเตอร์ (Transmission Speed) และ PLC module ในส่วน PLC side I/F





รูปที่ 7.8 การเลือกการเชื่อมต่อระหว่าง RS-232 Port และ USB Port

3.2.5 ตรวจสอบการเชื่อมต่อระหว่างคอมพิวเตอร์และ PLC โดยคลิกปุ่ม Connection Test ในหน้าต่าง Transfer Setup เพื่อทดสอบการเชื่อมต่อหากสามารถเชื่อมต่อได้สำเร็จจะแสดงผลการเชื่อมต่อว่าติดต่อกับ PLC รุ่นใด

รูปที่ แสดงการเชื่อมต่อกับ PLC FX1N(C)

### 3.3 การทดสอบการทำงานของอินพุต

เป็นการทดสอบการทำงานของอุปกรณ์อินพุตที่ต่อสายกับ PLC ว่ามีการเชื่อมต่อเรียบร้อยหรือไม่ และการเชื่อมต่อถูกต้องตรงตามหมายเลขของอุปกรณ์ที่กำหนดไว้ในการออกแบบโปรแกรมหรือไม่

รูปที่ การเข้าสู่คำสั่ง Batch Monitor ในการทดสอบการทำงานของอินพุต

ขั้นตอนในการทดสอบการทำงานของอินพุต ให้คลิกเมนู Online > Monitor > Device batch หรือคลิกปุ่ม Batch monitor ใน standard Toolbar หน้าต่าง Device batch monitor จะแสดงขึ้นมา ให้ใส่เบอร์อุปกรณ์ที่ช่อง Device เช่น X1 จากนั้นคลิกปุ่ม Start Monitor

ในการตรวจสอบการทำงานของอุปกรณ์อินพุตชนิด ON-OFF ทำได้โดยการส่งสัญญาณไฟฟ้า ON-OFF ที่อุปกรณ์อินพุตไปยัง PLC โดยสังเกตการทำงานในหน้าต่าง Device batch monitor หากมีการเปลี่ยนค่าระหว่าง 0 และ 1 ตามสัญญาณที่ให้ 0=OFF, 1=ON แสดงว่าอินพุตตัวนั้นๆ สามารถทำงานได้ถูกต้อง โดยเมื่อสั่งงานให้สวิตช์ X0-X6 ON ผลที่ได้จาก Device batch monitor จะแสดงดังรูป

รูปที่ ผลที่ได้จาก Device batch monitor

### 3.4 การทดสอบการทำงานของเอาต์พุต

เป็นการทดสอบการทำงานของอุปกรณ์เอาต์พุตที่ต่อสายกับ PLC ว่ามีการเชื่อมต่อเรียบร้อยหรือไม่และการเชื่อมต่อถูกต้องตามหมายเลขของอุปกรณ์ที่กำหนดไว้ในการออกแบบโปรแกรมหรือไม่ โดยในขั้นตอนการทดสอบ PLC จะต้องอยู่ใน Mode PLC Stop เสมอ

รูปที่ การเข้าสู่คำสั่ง Device test ในการทดสอบการทำงานของเอาต์พุต

ขั้นตอนในการทดสอบการทำงานของเอาต์พุต ทำได้โดยคลิกเมนู Online >Device test หรือคลิกปุ่ม Device Test ใน Standard Toolbar หน้าต่าง Device test จะแสดงขึ้นมาใส่เบอร์อุปกรณ์ที่ช่อง Device เช่น Y1 จากนั้นคลิกปุ่ม FORCE ON, FORCE OFF หรือ Toggle Force โดยเมื่อคลิกปุ่ม FORCE ON เอาต์พุตหมายเลขนั้นต้องทำงาน และเมื่อคลิกปุ่ม FORCE OFF เอาต์พุตจะหยุดทำงาน

รูปที่ หน้าต่าง Device test ขณะทดสอบการทำงานของเอาต์พุต

### 3.5 การทดสอบการทำงานของโปรแกรม

การทดสอบการทำงานของโปรแกรมเป็นขั้นตอนตรวจสอบการทำงาน และตรวจความถูกต้องของโปรแกรมก่อนการเขียนข้อมูลลงใน PLC ซึ่งในขั้นตอนนี้เป็นการทดสอบการทำงานที่กระทำบน Program GX Developer

ขั้นตอนการทดสอบการทำงานของโปรแกรม ทำได้โดยให้คลิกเมนู Tool>Start ladder test หรือคลิกปุ่ม Start or End Ladder logic test ใน Program Toolbar โปรแกรม GX Developer จะเริ่มทดสอบการทำงานของ Ladder Logic ด้วยวิธีการ Write to PLC ในแบบการจำลองการทำงานของ PLC ในคอมพิวเตอร์ (Simulation) ซึ่งจะยังไม่มีการส่งข้อมูลเข้าสู่ PLC

### รูปที่ การเข้าสู่คำสั่ง Start or End Ladder logic test

#### รูปที่ การเข้าสู่การทดสอบแลตเตอร์ลอจิก

เมื่อโปรแกรมเข้าสู่การทดสอบการทำงานของโปรแกรมแล้ว ผู้ใช้สามารถทดสอบการทำงานโดยการทดสอบให้สัญญาณสั่งงานอุปกรณ์ในลักษณะเดียวกับการทำงานจริงที่ออกแบบไว้ โดยการคลิกขวาแล้วเลือกคำสั่ง Device test... หน้าต่าง Device Test จะแสดงขึ้นมา ผู้ใช้สามารถป้อนสัญญาณอินพุตเพื่อสั่งงานได้โดยใส่หมายเลขอุปกรณ์ในช่อง Device ที่ต้องการให้สัญญาณและคลิกปุ่ม FORCE ON, FORCE OFF หรือ Toggle Force เมื่อป้อนสัญญาณสั่งงานตามลักษณะโปรแกรมที่ออกแบบแล้วตรวจสอบความถูกต้อง โดยสังเกตที่สถานะการทำงานของอุปกรณ์ต่างๆ ในโปรแกรมว่าทำงานถูกต้องตามที่ออกแบบไว้หรือไม่ เมื่อตรวจสอบความเรียบร้อยสามารถกลับไปสู่ขั้นตอนต่อไปโดยออกคำสั่ง Start ladder logic test โดยคลิกปุ่ม Start or End ladder logic test ใน Program Toolbar อีกครั้งหนึ่ง

รูปที่ การเข้าสู่คำสั่งทดสอบอุปกรณ์

รูปที่ การทดสอบอุปกรณ์

### 3.6 การเขียนข้อมูลลงใน PLC

การเขียนข้อมูลลงใน PLC เป็นขั้นตอนการส่งข้อมูลโปรแกรมที่ได้ออกแบบไว้ในโปรแกรม GX Developer ส่งไปยัง PLC

รูปที่ การเข้าสู่คำสั่ง Write to PLC

ขั้นตอนการส่งข้อมูลลงใน PLC (Write to PLC) ทำได้โดยการคลิกเมนู Online >Write to PLC หรือคลิกปุ่ม Write to PLC ใน Standard Toolbar จากนั้นคลิกเลือกข้อมูลที่จะส่งถ่ายไปยัง PLC ในหน้าต่าง Write to PLC ซึ่งจะมีให้เลือกคือ Program, Device Comment และ Parameter ซึ่งทุกครั้งจะต้องเลือก Program และ Parameter เนื่องจากเป็นข้อมูลสำคัญในการสั่งงาน PLC เมื่อเลือกรายการเรียบร้อยแล้วคลิกปุ่ม Execute เพื่อยืนยันการส่งข้อมูล

สำหรับ PLC รุ่น Q CPU (Q Mode) ต้อง Format PLC Memory ก่อนเสมอจึงจะส่งถ่ายข้อมูล

รูปที่ การเลือกข้อมูลเพื่อเขียนโปรแกรมสู่ PLC (Write to PLC)

### 3.7 การอ่านข้อมูลจาก PLC

การอ่านข้อมูลจาก PLC เป็นขั้นตอนการดึงข้อมูลโปรแกรมที่อยู่ใน PLC ออกมาเพื่ออ่านหรือปรับปรุงแก้ไขโปรแกรม GX Developer

รูปที่ การเข้าสู่คำสั่ง Read from PLC..

ขั้นตอนการอ่านข้อมูลจาก PLC (Read from PLC) ทำได้โดยการคลิกเมนู Online >Read from PLC... หรือคลิกปุ่ม Read From PLC ใน Standard Toolbar จากนั้นเลือกขข้อมูลที่จะดึงข้อมูลจาก PLC ในหน้าต่าง Read from PLC ซึ่งจะมีให้เลือกคือ Program, Device memory และ Parameter เมื่อเลือกเรียบร้อยแล้วคลิกปุ่ม Execute เพื่อยืนยันการดึงข้อมูล

รูปที่ การเลือกข้อมูลเพื่อดึงโปรแกรมจาก PLC (Read from PLC...)

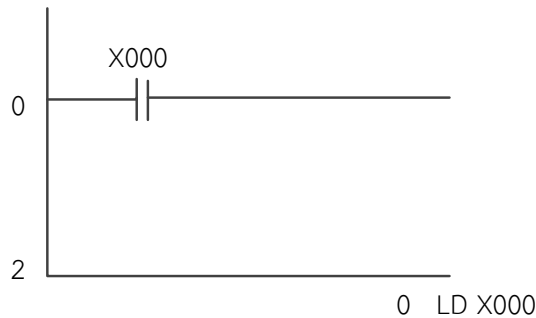
**3.คำสั่งพื้นฐานในการออกแบบโปรแกรม**

คำสั่งพื้นฐานในวงจรแลดเดอร์ (Ladder) ที่ควรทราบมีดังนี้

3.1 คำสั่ง Load, Load Inverse

คำสั่ง Load(LD) และ Load Inverse(LDI) เป็นคำสั่งที่เริ่มต้นในแต่ละ Line โดยจะเชื่อมต้นในแต่ละ Line โดยจะเชื่อมต่อกับเส้นแนวตั้งทางด้านซ้าย ใช้กับหน้าสัมผัสประเภทอินพุต (X) เอาต์พุต(Y) อุปกรณ์รีเลย์ช่วย (M,S) อุปกรณ์ตั้งเวลา(T) และอุปกรณ์นับจำนวน(C) เป็นตัวสั่งงานเพื่อขับ Load ในบรรทัดนั้นๆ

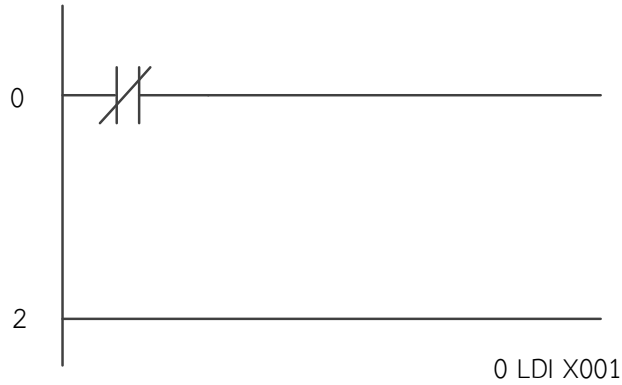
คำสั่ง Load จะใช้กับหน้าสัมผัสแบบปกติเปิด (Normally Open : NO) เสมอ สัญลักษณ์ที่ใช้ในภาษา Instruction List คือ LD และมีโปรแกรม Steps เท่ากับ 1



(ภาษา Instruction List Steps ที่ 0 Load อินพุต X000)

รูปที่ 7.1 คำสั่ง Load หน้าสัมผัสปกติเปิดอินพุต X000

คำสั่ง Load Inverse จะใช้กับหน้าสัมผัสแบบปกติปิด (Normally Closed:NC) เสมอ สัญลักษณ์ที่ใช้ในภาษา Instruction List คือ LDI และมีโปรแกรม Steps เท่ากับ 1

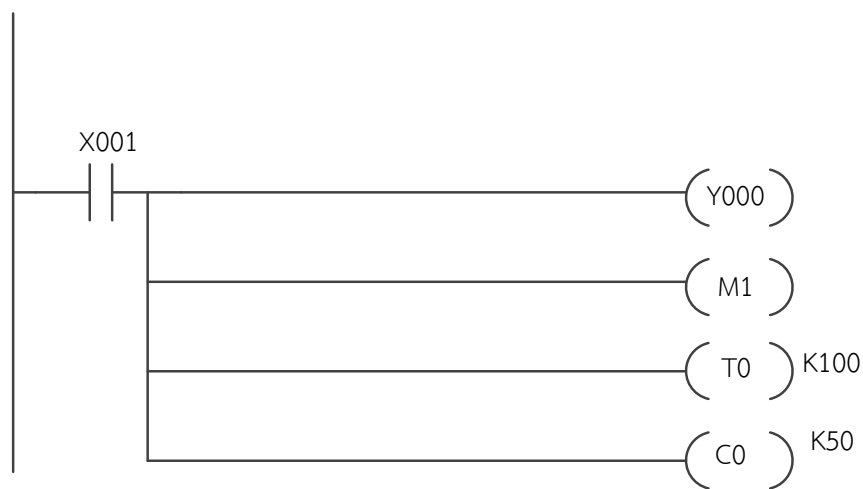


(ภาษา Instruction List Steps ที่ 0 Load Invers อินพุต X001)

รูปที่ 7.2 คำสั่ง Load Inverse หน้าสัมผัสปกติปิดอินพุต X001

### 3.2 คำสั่ง OUT

คำสั่ง out เป็นคำสั่งที่สิ้นสุดในแต่ละ Line โดยจะเชื่อมต่อกับเส้นแนวตั้งทางด้านขวา ใช้กับ คอยล์ประเภทเอาต์พุต (Y) อุปกรณ์รีเลย์ช่วย (M,S) อุปกรณ์ตั้งเวลา (T) และอุปกรณ์นับจำนวน (C) โดยคำสั่ง Out สามารถเชื่อมต่อในแบบขนานได้หลายตัว และทำงานพร้อมกันเมื่อคำสั่ง Load สั่งงานจ่ายกระแสไฟมาขั้วสัญลักษณ์ที่ใช้ในภาษา Instruction List คือ OUT



รูปที่ 7.3 คำสั่ง OUT ประเภทต่างๆ



คำสั่ง OUT ประเภทอุปกรณ์ตั้งเวลา (Timer :T) เป็นคำสั่งที่ทำหน้าที่ตั้งเวลาเพื่อหน่วงเวลาของช่วงการทำงานในขั้นตอนที่มีการตั้งเงื่อนไขการทำงานตามเวลาที่กำหนด โดยคำสั่ง Out ประเภทตั้งเวลา (Timer) นี้จะต้องกำหนดค่าของข้อมูลค่าตัวหน่วงเวลา โดยกำหนดเป็นค่าคงที่ด้วยตัวอักษรและตามด้วยค่าเวลา เช่น K100 ซึ่งค่าเวลา 100 จะขึ้นอยู่กับชนิดของอุปกรณ์ตั้งเวลา เช่น ใน PLC รุ่น FX3U มีอุปกรณ์ตั้งเวลา 511 ตัว โดย T000-T191 เป็น Timer ชนิด 100 msec

ดังนั้น T000 K100 ค่าหน่วงเวลาจะเท่ากับ 100x100msec เท่ากับ 10 วินาที เมื่อคอยส์ของ Timer ถูกนับครบ 10 วินาที หน้าสัมผัสของ Timer จะทำงาน (NO) โดยการตั้งค่าคงที่ K มีค่าเป็นแบบ 16bit คือ 1-32,767 และมีโปรแกรม Steps เท่ากับ 3 ดังตารางที่ 7.1

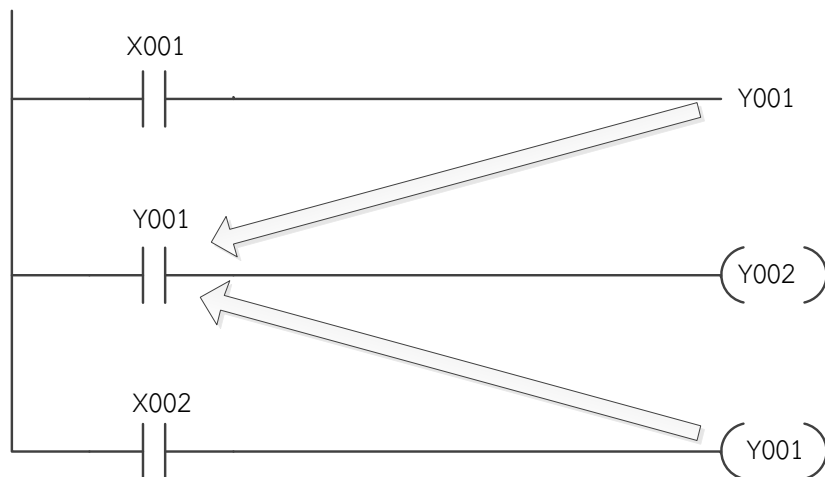
คำสั่ง Out ประเภทอุปกรณ์นับจำนวน (Counter:C) เป็นคำสั่งที่ใช้ในนับจำนวนเพื่อตั้งค่าการทำงานให้มีการทำงานครบตามจำนวนที่ตั้งไว้ตามขั้นตอนการทำงานที่กำหนด โดยคำสั่ง Out ประเภทอุปกรณ์นับจำนวน (Counter) นี้จะต้องกำหนดค่าของข้อมูลการนับจำนวน โดยกำหนดเป็นค่าคงที่ด้วยตัวอักษรละตามด้วยค่าจำนวน

อุปกรณ์หน่วงค่าเวลา (Timers) ชนิด 16 บิต	ค่าความละเอียดของอุปกรณ์หน่วงค่าเวลา PLC Mitsubishi FX1N,FX2N		
	100msec	10msec	1msec
อุปกรณ์หน่วงค่าเวลาทั่วไป	T0-T199	T200-T245	T246-T249
อุปกรณ์หน่วงค่าเวลาแบบจำค่า	T250-T255	T200-T245	T246-T249
อุปกรณ์นับจำนวน(Counters)	อุปกรณ์นับจำนวน PLC Mitsubishi FX1N,FX2N		
	ชนิด 16 บิต (-32,768 ถึง +32,767)	ชนิด 32 บิต แบบนับขึ้นและนับลง (-2,147,438,648 ถึง +2,147,438,647)	
อุปกรณ์หน่วงค่าเวลาทั่วไป	C0-C99	C200-C219	

อุปกรณ์ช่วงเวลาแบบจำค่า	C100-C199	C200-C234
-------------------------	-----------	-----------

การใช้คอยล์ในคำสั่ง Out ในการใช้คอยล์โดยปกติไม่ควรใช้งานซ้ำ เนื่องจากอาจทำให้โปรแกรมเกิดความผิดพลาดขึ้นจากการที่มีการสั่งงานจากหน้าสัมผัส หรืออินพุตที่แตกต่างกันในแต่ละบรรทัด โดยที่โปรแกรมจะทำงานจากซ้ายไปขวา และจากบนลงล่าง ดังนั้น หากมีการทำงานของคอยล์ซ้ำกันจะทำให้บรรทัดที่มีใช้หน้าสัมผัสของคอยล์เป็นตัวสั่งงานเกิดความสับสนในเรื่องของขั้นตอนและการทำงานขึ้น

จากรูปที่ 7.4 มีการใช้คอยล์ Output Y001 ซ้ำกัน 2 ตัว โดยในบรรทัดแรกกำหนดให้หน้าสัมผัสอินพุต X001 เป็นตัวสั่งให้คอยล์ Y001 ON เมื่อหน้าสัมผัสอินพุต X001 ON ส่งผลให้คอยล์ Y001 ON หน้าสัมผัส Y001 ในบรรทัดที่สองจะทำงานและสั่งให้ Output Y002 ทำงานด้วย และจากหลักการของโปรแกรมจะทำงานจากบรรทัดบนลงล่าง ดังนั้น ในบรรทัดสุดท้ายหากหน้าสัมผัสอินพุต X002 OFF ส่งผลให้ Output Y001 ไม่ทำงานโปรแกรมจะประมวลผลใหม่รายงานผลว่าหน้าสัมผัส Y001 OFF แต่ Y002 ON อยู่ หรืออาจ OFF ได้ตามลักษณะโปรแกรม ทำให้เกิดความสับสนในการตรวจสอบการทำงานของอุปกรณ์ในโปรแกรมได้



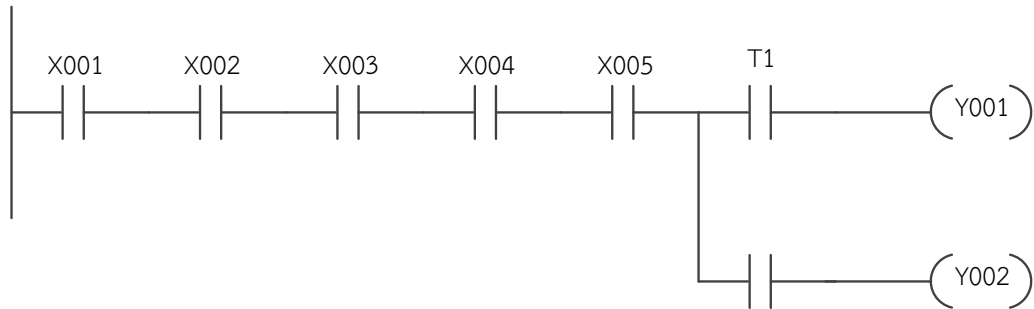
รูปที่ 7.4 การใช้คอยล์ Y001 ซ้ำ

### 7.3 คำสั่ง And ,And Inverse

คำสั่ง And(AND) และ And Invers (ANI) เป็นคำสั่งที่ใช้ในการต่อหน้าสัมผัสแบบอนุกรม โดยต่อจากหน้าสัมผัสในคำสั่ง Load หรือ Load Inverse ซึ่งสามารถใช้หน้าสัมผัสประเภทอินพุต (X)

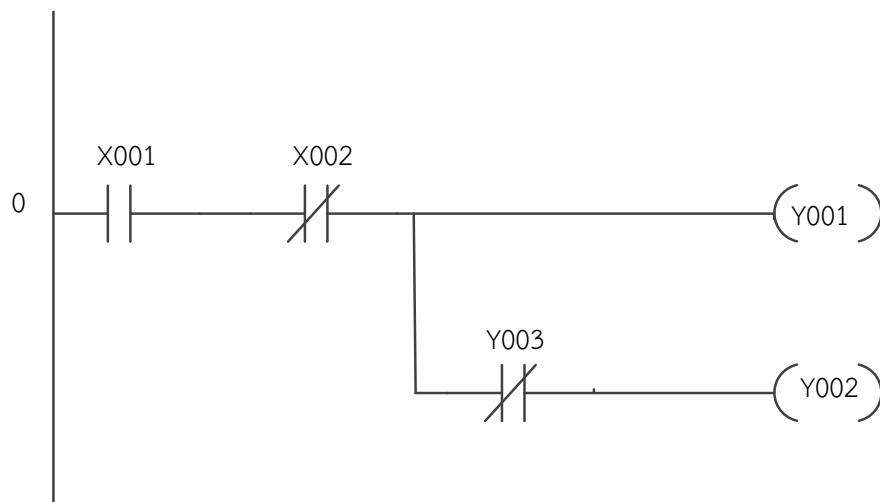
เอาต์พุต(Y) อุปกรณ์รีเลย์ช่วย (M,S) อุปกรณ์ตั้งเวลา (T) และอุปกรณ์นับจำนวน (C) มาต่ออนุกรม และสามารถต่อหน้าสัมผัสได้อย่างไม่จำกัด แต่ต้องไม่เกิน 10 หน้าสัมผัสในหนึ่งบรรทัดเพื่อใช้เป็นเงื่อนไข “และ” เพิ่มเติมจากคำสั่ง Load หรือ Load Inverse ในการสั่งงานเพื่อขับ Load ในบรรทัดนั้นๆ

คำสั่ง And จะใช้กับหน้าสัมผัสแบบปกติเปิด (Normally Open : NO) เสมอ สัญลักษณ์ที่ใช้ในภาษา Instruction List คือ AND และมีโปรแกรม Steps เท่ากับ 1



รูปที่ 7.5 การใช้งานคำสั่ง AND หลายๆหน้าสัมผัส

จากรูปที่ 7.5 โปรแกรมจะมีขั้นตอนการทำงานโดยหากคอยส์ Y001 จะ ON ได้ก็ต่อเมื่อมีการสั่งงานให้หน้าสัมผัสอินพุต X001,X002,X003,X004,X005 และหน้าสัมผัสอุปกรณ์ตั้งเวลา T001 ทำงานพร้อมกันและคอยส์ Y002 จะ ON และหน้าสัมผัสเอาต์พุต Y003 เท่ากับ 1



รูปที่ 7.6 การใช้งานคำสั่ง ANI หลายๆหน้าสัมผัส

จากรูปที่ 7.6 โปรแกรมจะมีขั้นตอนการทำงาน โดยตำแหน่งปกติหากหน้าสัมผัสอินพุต X001 ทำงาน จะทำให้คอยส์ Y001 และ Y002 ON แต่หากคอยส์ Y003 ON หน้าสัมผัสเอาต์พุต Y003 เปิดจะทำให้ คอยส์ Y002 OFF และหากหน้าสัมผัสอินพุต X002 ทำงานหน้าสัมผัสเปิด คอยส์ Y001 และ Y002 ก็ จะ OFF ทันที เนื่องจากไม่มีกระแสไฟฟ้าส่งถึงคอยส์เหล่านั้น

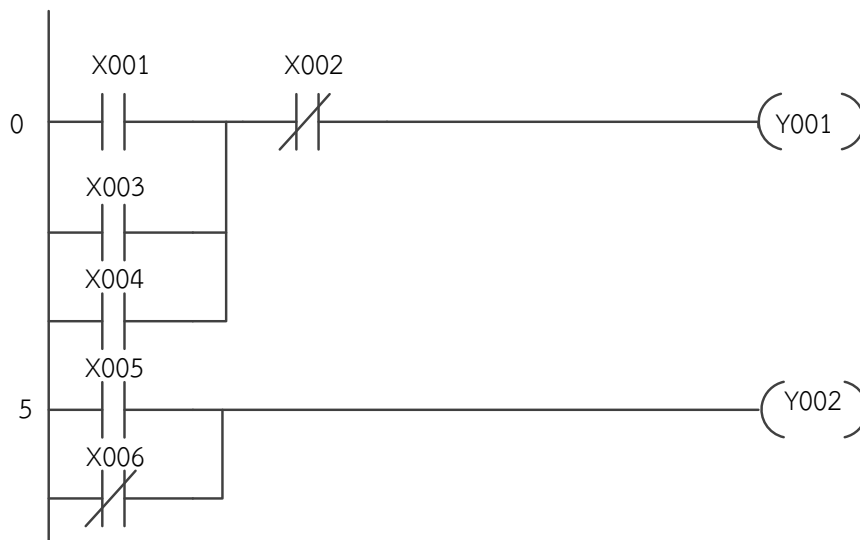
#### 7.4 คำสั่ง Or, Or Inverse

คำสั่ง Or(Or) และ Or Inverse(ORI) เป็นคำสั่งที่ใช้ในการต่อหน้าสัมผัสแบบขนาน โดยต่อ ขนานกับหน้าสัมผัสในคำสั่ง Load หรือ Load Inverse ต่อขนานกับวงจรที่มีการใช้คำสั่ง And หรือ And Inverse ซึ่งสามารถใช้หน้าสัมผัสประเภทอินพุต(X)เอาต์พุต (Y) อุปกรณ์รีเลย์ช่วย(M,S) อุปกรณ์ตั้ง เวลา (T) และอุปกรณ์นับจำนวน(C) มาต่อขนาน โดยทางด้านซ้ายมือคำสั่ง OrหรือOr Inverse จะอยู่ กับเส้นตั้งด้านซ้ายมือเสมอ

การต่อดำเนินการด้วยคำสั่ง OR หรือ Or Inverse นี้สามารถต่อหน้าสัมผัสได้อย่างไม่จำกัดจำนวน แต่ ต้องไม่เกิน 24 แฉวเพื่อใช้เป็นเงื่อนไข “หรือ” เพิ่มเติมจากคำสั่ง Load หรือ Load Inverse ในการ สั่งงานเพื่อขับ Load ในบรรทัดนั้นๆ

คำสั่ง Or Inverse ใช้กับหน้าสัมผัสแบบปกติเปิด (Normally Open:NO) เสมอ สัญลักษณ์ที่ใช้ ในภาษา Instruction List คือ Or และมีโปรแกรม Steps เท่ากับ 1

คำสั่ง Or Inverse ใช้กับหน้าสัมผัสแบบปกติปิด(Nomally Closed:NC ) เสมอสัญลักษณ์ที่ใช้ ในภาษา Instruction List คือ ORI และมีโปรแกรม Steps เท่ากับ 1



รูปที่ 7.7 การต่อใช้งานคำสั่ง OR and ORI หลายๆหน้าสัมผัส

จากรูปที่ 7.7 โปรแกรมจะมีขั้นตอนการทำงานโดยคอยส์ Y001 จะ ON ได้ต่อเมื่อน้ำสัมผัสอินพุต X001หรือ X003 หรือ X004 เป็นการต่อแบบขนาน ซึ่งแต่ละหน้าที่สัมผัสเป็นแบบปกติเปิดจึงเป็นชนิดคำสั่ง Or Inverse ผลของการต่อแบบนี้ช่วยให้การเริ่มต้นของวงจรหรือเครื่องจักรที่ออกแบบวงจรแบบคำสั่ง Or Inverse สามารถเลือกสั่งงานที่น้ำสัมผัสใดก็ได้ โดยอาจมีการออกแบบให้เป็นสวิตช์ปุ่มกดที่มีอยู่ในที่ต่างกันได้ แต่เมื่อใดที่น้ำสัมผัส X002 ทำงานน้ำสัมผัสเปิด คอยส์ Y001 จะหยุดทำงานทันที

ในบรรทัดถัดมาเป็นการต่อแบบปกติปิด ซึ่งส่วนใหญ่จะออกแบบน้ำสัมผัสอินพุต X006 เป็นประเภทเซนเซอร์หรือลิมิตสวิตช์เป็นต้น

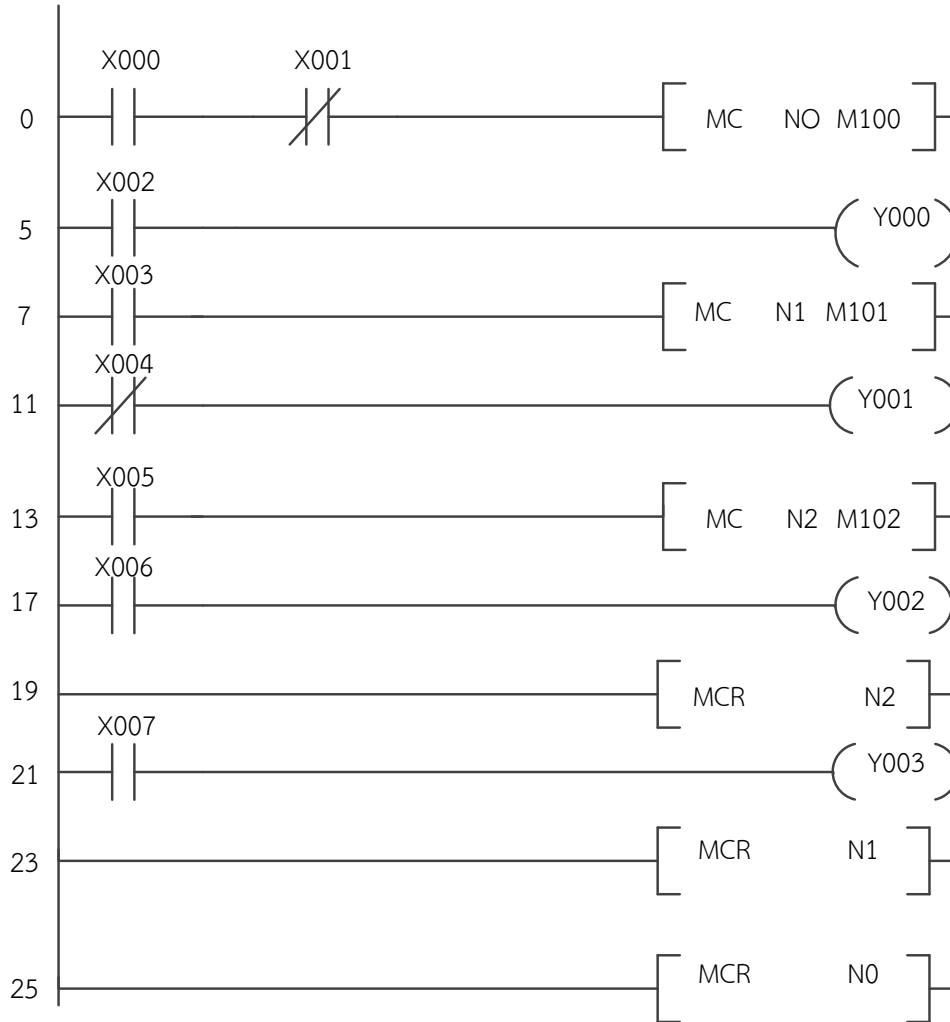
## 7.5 คำสั่ง Master Control, Reset

คำสั่ง Master Control and Reset (MC,MCR) เป็นคำสั่งที่ใช้ควบคุมการทำงานของชุดวงจรให้เป็นไปตามลำดับขั้นการทำงาน ซึ่งในแต่ละชุดลำดับจะถูกควบคุมด้วยคำสั่ง MC busline ซึ่งทำหน้าที่เปรียบเสมือนเป็นการสร้างเส้นทางไฟขึ้นใหม่ในวงจร

คำสั่ง MC จะมีน้ำสัมผัสที่ทำหน้าที่ในการสั่งงานให้ชุดควบคุมต่อไฟเลี้ยงลูปทำงาน N ซึ่งลูปทำงาน N จะมีหมายเลขกำกับแต่ละลูป เช่น (MC No M100) จะเป็นการสั่งงานให้ลูป NO ทำงาน รีเลย์ M100 จะต่อสัญญาณไฟไปรอับชุดการทำงานที่อยู่ในลูปนั้นๆ และจะสิ้นสุดหรือยกเลิกการทำงานเมื่อน้ำสัมผัสที่สั่งงานคำสั่ง MC เป็น Off

คำสั่ง MCR รีเซตลูป ทำให้อุปกรณ์หรือโหลดในลูปนั้นๆหยุดทำงาน ยกเว้นอุปกรณ์ประเภท Retentive Time (ชนิดเวลาสะสม)

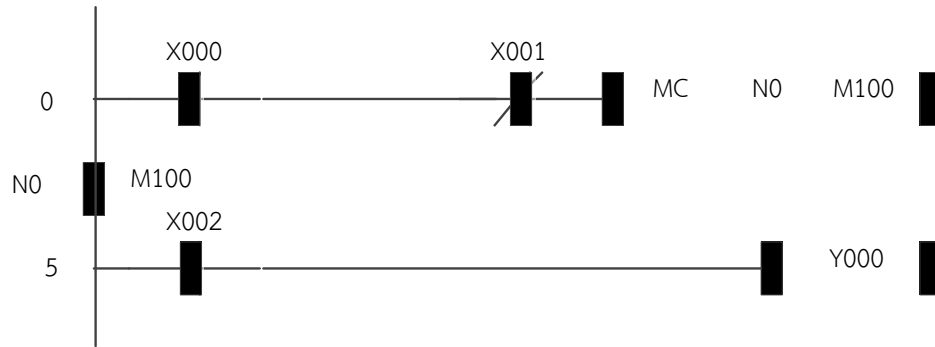
คำสั่ง MC จะสามารถซ้อนลูปภายในได้ 8 ชั้น กำหนดโดย N0-N7(ลูปนับเบอร์) โดยคำสั่ง MC มีโปรแกรม Steps เท่ากับ 3 ใช้สั่งงานน้ำสัมผัสประเภทเอาต์พุต(Y) อุปกรณ์รีเลย์ช่วย (M) และคำสั่ง MCR มีโปรแกรม Steps เท่ากับ 2 ใช้ในการรีเซตลูป



รูปที่ 7.8 ตัวอย่างการทำงานคำสั่ง MC,MCR

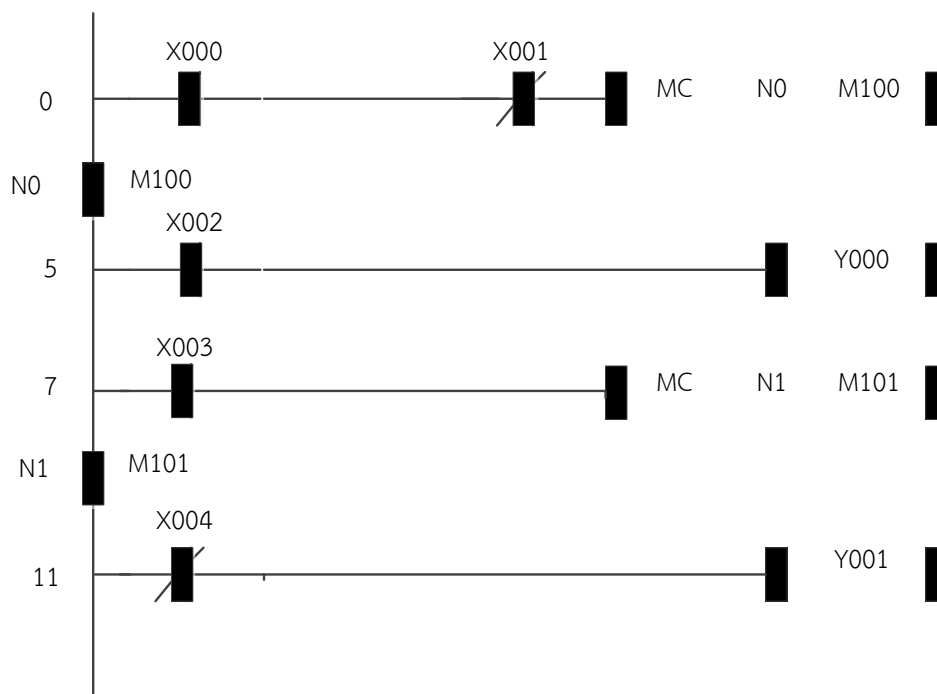
จากรูปสามารถอธิบายการทำงานได้ดังนี้

1. เริ่มต้นการทำงานจากหน้าสัมผัส X000 ON คำสั่ง MC จะสั่งให้ลูป NO ทำงานรีเลย์ M100 จะถูกสั่งงานทำให้มีกระแสไฟส่งไปที่หน้าสัมผัส X002 และเมื่อหน้าสัมผัส X002 ON จะทำให้คอยล์ Output Y000 ทำงานดังรูปที่ 7.9 แต่หากหน้าสัมผัส X000 off เมื่อใดจะทำให้คำสั่ง MC Reset คอยล์ Output Y000 หยุดทำงาน

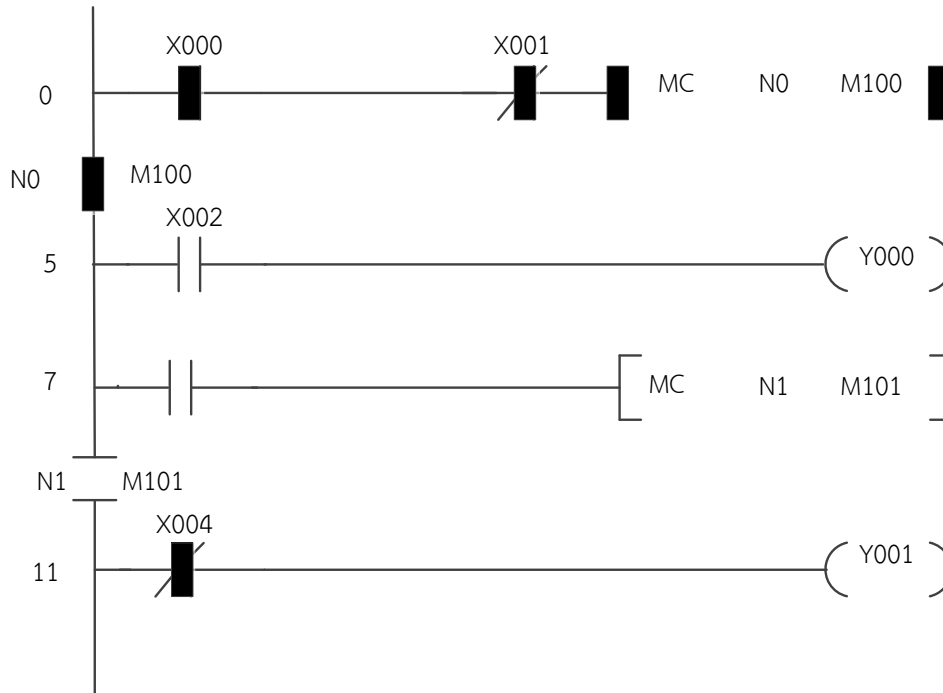


รูปที่ 7.9 แสดงการสั่งงานคอยล์ Y000

2. เมื่อหน้าสัมผัส X000 ON นอกจากจะส่งกระแสไฟเพื่อขับโหลด Y000 แล้ว ยังส่งกระแสไฟมาที่หน้าสัมผัส X003 ด้วย และเมื่อหน้าสัมผัส X003 ON จะสั่งให้ลูป N1ทำงานรีเลย์ M101 จะถูกสั่งงานทำให้มีกระแสไฟส่งไปที่หน้าสัมผัส X004 ซึ่งเป็นหน้าสัมผัสแบบปกติปิด ทำให้คอยล์ Output Y001 ทำงานดังรูปที่ 7.10 แต่หาก หน้าสัมผัส X003 หรือ X000 off เมื่อใดจะทำให้คำสั่ง MC Reset คอยล์ Output Y001 หยุดทำงาน แต่จะไม่ขึ้นอยู่กับหน้าสัมผัส X002 ดังรูปที่ 7.11



รูปที่ 7.10 แสดงการสั่งงานคอยล์ Y001



รูปที่ 7.11 แสดงการยกเลิกสั่งงานคอยส์ Y001

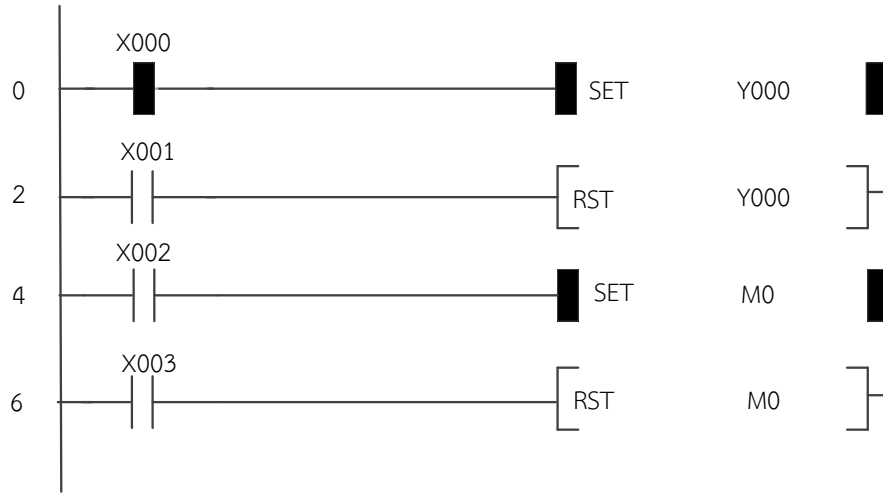
## 7.6 คำสั่ง Set,Reset

คำสั่ง Set and Reset (SET,RST) เป็นคำสั่งที่ใช้ควบคุมการทำงานของอุปกรณ์ โดยคำสั่ง Set จะทำงานเมื่อมีกระแสไฟจ่ายไปที่คำสั่ง Set ซึ่งอาจจะมาจากหน้าสัมผัสปกติเปิดหรือปกติปิด โดยคำสั่ง Set จะส่งคำสั่งสัญญาณ 1 เพื่อสั่งงานให้อุปกรณ์ที่ถูกควบคุม ON และจะค้างตำแหน่ง ON ไปตลอดแม้ว่าคำสั่ง ON ไปตลอดแม้ว่าคำสั่ง Set จะไม่มีกระแสไฟมาเลี้ยงแล้วก็ตาม อุปกรณ์ที่ถูกควบคุมจะ Off ได้ต่อเมื่อมีคำสั่ง Reset ส่งคำสั่งสัญญาณ 0 มาสั่งให้อุปกรณ์นั้นๆ OFF และอุปกรณ์ก็จะ OFF จนกระทั่งมีคำสั่ง Set ส่งสัญญาณ 1 มากระทำอีกครั้ง

คำสั่ง Set และ Reset สามารถกับอุปกรณ์เบอร์เดียวกันได้หลายครั้ง และคำสั่ง Reset สามารถใช้ในการรีเซ็ตข้อมูลที่บรรจุในอุปกรณ์ต่างๆ เช่น อุปกรณ์เก็บข้อมูลเชิงตัวเลข (Data Registers :D) เป็นต้น โดยคำสั่ง Set สามารถใช้สั่งงานหน้าสัมผัสประเภทเอาต์พุต (Y) อุปกรณ์รีเลย์ช่วย (M,S) มีโปรแกรม Steps สำหรับหน้าสัมผัสประเภทเอาต์พุต (Y) อุปกรณ์รีเลย์ช่วย(S) และรีเลย์ช่วยพิเศษ (Special Relay:M)เท่ากับ 2 ส่วนคำสั่ง Reset ใช้สั่งงานหน้าสัมผัสประเภทเอาต์พุต (Y) อุปกรณ์รีเลย์ช่วย(M,S) อุปกรณ์เก็บข้อมูลเชิงตัวเลข (D)อุปกรณ์นับจำนวน (C) มีโปรแกรม Steps



สำหรับหน้าสัมผัสประเภทเอาต์พุต (Y) อุปกรณ์รีเลย์ช่วย (M)เท่ากับ 1 อุปกรณ์เก็บข้อมูลเชิงตัวเลข (D) เท่ากับ 3



รูปที่ 7.12 ตัวอย่างการทำงานคำสั่ง SET,RESET

### 7.7 คำสั่ง Timer ,Counter(Reset)

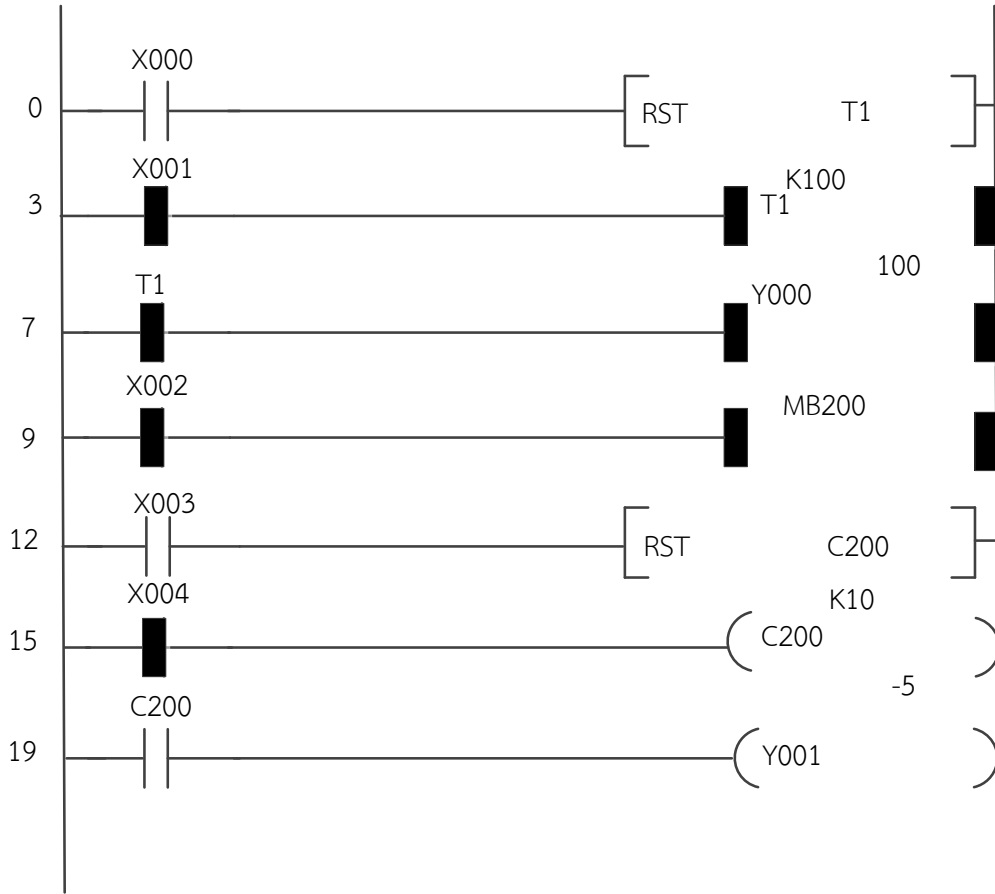
คำสั่ง Reset Timer,Reset Counter (RST) เป็นคำสั่งที่ใช้ควบคุมการทำงานของอุปกรณ์ประเภทอุปกรณ์ตั้งเวลา (T) และอุปกรณ์นับจำนวน (C) โดยเมื่อหน้าสัมผัสที่ควบคุมการทำงานของอุปกรณ์ตั้งเวลา (T) หรืออุปกรณ์นับจำนวน (C) ทำงานทำให้อุปกรณ์นับเวลาหรือจำนวนจนครบตามที่ตั้งไว้ หน้าสัมผัสของอุปกรณ์ตั้งเวลา (T) หรืออุปกรณ์นับจำนวน (C) จะสั่งงานให้คอยส์เอาต์พุตทำงาน และเมื่อ Reset อุปกรณ์อุปกรณ์ตั้งเวลา(T) หรืออุปกรณ์นับจำนวน (C) จะหยุดสั่งงานทำให้คอยส์เอาต์พุตทำงานด้วย โดยคำสั่ง RST มีโปรแกรม Steps เท่ากับ 3 ดังตัวอย่างในรูปที่ 7.13

สามารถอธิบายการทำงานได้ดังนี้

1. เริ่มต้นการทำงานจากหน้าสัมผัส X000 ON คำสั่ง RST จะสั่งรีเซ็ต T1ทำให้อุปกรณ์และหน้าสัมผัส T1 หยุดทำงาน หากต้องการให้อุปกรณ์นับเวลา T1 เริ่มต้นทำงานต้องยกเลิกการรีเซ็ตโดยคำสั่งหน้าสัมผัส X000 OFF และสั่งให้หน้าสัมผัส X001 ON อุปกรณ์นับเวลา T1 จะเริ่มนับเวลาตามค่าคงที่ K100 มีค่าเท่ากับ 100 msec หรือ 10 วินาที เมื่อครบตามเวลาที่ตั้งไว้หน้าสัมผัส T1

จะสั่งงานให้คอยส์เอาต์พุต Y000 ทำงาน และหยุดทำงานต่อเมื่อหน้าสัมผัส X000 ON คำสั่ง RST ทำงานอีกครั้ง

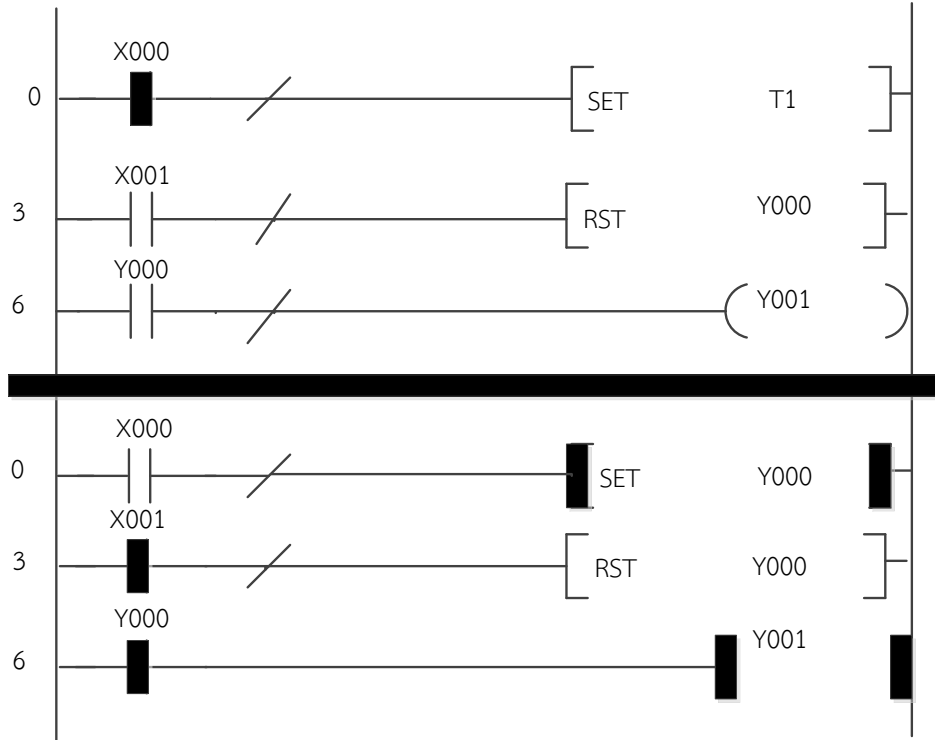
2. การรีเซ็ตต่ออุปกรณ์นับจำนวน โดยสั่งงานให้หน้าสัมผัสอินพุต X003 ON คำสั่ง RST จะสั่งรีเซ็ต C200 ทำให้อุปกรณ์และหน้าสัมผัส C200 หยุดทำงาน หากต้องการให้อุปกรณ์นับจำนวน C200 เริ่มต้นทำงานต้องยกเลิกการรีเซ็ต โดยคำสั่งให้หน้าสัมผัส X003 OFF และสั่งให้หน้าสัมผัส X004 ON อุปกรณ์นับเวลา C200 จะเริ่มนับเวลาตามค่าคงที่ K10 มีค่าเท่ากับ 10 ครั้ง โดยนับจำนวนครั้งการ ON-OFF ของหน้าสัมผัสอินพุต X004 ซึ่งอุปกรณ์นับจำนวน C200 เป็นชนิด 32 bit (ชนิดนับขึ้น , นับลง) โดยใช้รีเลย์ช่วยพิเศษ M8200 เมื่อ M8200 ทำงาน อุปกรณ์นับจำนวนที่ทำงานต่อจาก M8200 จะเป็นแบบนับลง ดังนั้น จากภาพหน้าสัมผัสอินพุต X002 ทำงานให้อุปกรณ์นับจำนวน C200 นับจาก 0 ไปหา -10 ตามค่าคงที่ K10 เมื่อครบตามจำนวนที่ตั้งไว้ หน้าสัมผัส C200 จะสั่งงานให้คอยส์เอาต์พุต Y001 ทำงานและหยุดทำงานต่อเมื่อหน้าสัมผัส X003 ON คำสั่ง RST ทำงานอีกครั้ง



รูปที่ 7.14 ตัวอย่างการทำงานคำสั่ง Reset Timer&Counter

## 7.8 คำสั่ง Inverse

คำสั่ง Inverse(INV) เป็นคำสั่งที่ใช้เปลี่ยนสถานการณ์ทำงานของแลตเตอร์ (Ladder) เป็นในทางตรงกันข้ามประโยชน์เพื่อใช้ในการปรับปรุงแก้ไขโปรแกรมหรือทดสอบการทำงานในลักษณะตรงกันข้ามโยคำสั่ง INV มีโปรแกรม Steps เท่ากับ 1



รูปที่ 7.15 ตัวอย่างการทำงานของคำสั่ง Inverse

จากรูปที่ 7.15 สามารถอธิบายการทำงานได้ดังนี้

การทำงานในวงจรปกติที่ไม่มีคำสั่ง Inverse เมื่อหน้าสัมผัส X000 ON คำสั่ง Set จะสั่งให้เอาต์พุต Y000 ทำงานค้างตำแหน่งจนกว่าหน้าสัมผัสอินพุต X000 ON คำสั่ง Reset จะส่งสัญญาณ 0 เพื่อตัดการทำงานคอยส์เอาต์พุต Y000

การทำงานในวงจรที่ต่อคำสั่ง Inverse ก่อนหน้าคำสั่ง Set และ Reset ทำให้การทำงานเป็นในลักษณะตรงกันข้ามดังรูปที่ 7.15 (ล่าง) การที่จะสั่งให้คอยส์เอาต์พุต Y000 ทำงานต้องสั่งงานให้หน้าสัมผัสอินพุต X001 ON และหากต้องการ Reset การทำงาน ต้องสั่งงานที่หน้าสัมผัสอินพุต X000 ON ซึ่งจะเห็นว่าเป็นการสลับการทำงานของคำสั่ง Set และ Reset

### 7.9 คำสั่ง End

คำสั่ง End (END) เป็นคำสั่งหยุดการทำงานของโปรแกรม ณ บรรทัดนั้นๆ ในการใช้งานคำสั่ง End ปกติ จะใส่ไว้ในบรรทัดสุดท้ายของโปรแกรม แต่ในขั้นตอนการทดสอบการทำงานของโปรแกรมที่มีขั้นตอนการทำงานที่ยาวมากๆ สามารถแยกขั้นตอนออกเป็นย่อยๆ โดยคำสั่ง End ได้ ณ จุดที่ต้องการ

แยกทดสอบ เมื่อทดสอบเรียบร้อยแล้วจึงค่อยลบคำสั่ง End ทั้ง มิฉะนั้นโปรแกรมจะทำงานสิ้นสุดที่  
ขั้นตอนที่มีคำสั่ง End

ในคำสั่งพื้นฐานทุกๆคำสั่งที่ผ่านมา ในตัวอย่างบรรทัดสุดท้ายจะต้องเพิ่มคำสั่ง End ทั้งหมด  
มิฉะนั้นโปรแกรมอาจเกิดการผิดพลาดขึ้นได้ โดยคำสั่ง END มีโปรแกรม Steps เท่ากับ 1



รูปที่ 7.16 ตัวอย่างการทำงานของคำสั่ง End

