MITSUBISHI ELECTRIC
Changes for the Better

PLC

พื้นฐาน GX Works2

หลักสูตรการฝึกอบรมนี้ (หลักสูตรอิเล็กทรอนิกส์) ได้รับการออกแบบมาสำหรับผู้ที่ใช้งานซอฟต์แวร์ GX Works2 เพื่อสร้างโปรแกรมลำดับเป็นครั้งแรก

Copyright ©2014 Mitsubishi Electric Corporation. All Rights Reserved.

บทนำ วัตถุประสงค์ของหลักสูตร

หลักสูตรนี้จะอธิบายเกี่ยวกับความรู้พื้นฐานเกี่ยวกับการใช้งานซอฟต์แวร์ GX Works2 เพื่อการตั้งโปรแกรม การแก้ไขความผิดพลาด และการตรวจสอบการทำงานของตัวควบคุมแบบตั้งโปรแกรมได (PLC) หลักสูตรนี้จัดทำขึ้นสำหรับผู้สร้างโปรแกรมเชิงลำดับสำหรับตัวควบคุม MELSEC-Q ซีรีส์ MELSEC-L ซีรีส์ และ MELSEC-F ซีรีส์

บทนำ โครงสร้างของหลักสูตร

เนื้อหาของหลักสูตรนี้มีดังนี้
เราขอแนะนำให้คุณเริ่มต้นจากบทที่ 1

บทที่ 1 - วิธีการควบคุมระบบ PLC

ภาษาของโปรแกรมและซอฟต์แวร์ที่ใช้สำหรับตั้งโปรแกรมจะได้รับการแนะนำในหลักสูตรนี้

บทที่ 2 - การออกแบบโปรแกรม

คุณจะได้เรียนรู้วิธีการออกแบบโปรแกรมด้วยรายการควบคุมและการกำหนดค่าฮาร์ดแวร์

บทที่ 3 - การตั้งโปรแกรม

คุณจะได้เรียนรู้วิธีการตั้งโปรแกรมโดยใช้ซอฟต์แวร์ GX Works2

บทที่ 4 - การแก้ไขความผิดพลาด

คุณจะได้เรียนรู้วิธีการเขียนโปรแกรมลำดับไปยังโมดูล CPU และการแก้ไขความผิดพลาด

บทที่ 5 - แบบทดสอบประเมินผล

คะแนนที่ผ่านหลักสูตร: 60% ขึ้นไป

บทนำ วิธีการใช้งานเครื่องมือการเรียนรู้อิเล็กทรอนิกส์นี้

ไปที่หน้าถัดไป		ไปที่หน้าถัดไป
กลับไปยังหน้าที่แล้ว		กลับไปยังหน้าที่แล้ว
เลื่อนไปยังหน้าที่ต้องการ		ระบบจะแสดง "สารบัญ" ช่วยให้คุณสามารถไปยังหน้าต่างๆ ได้
ออกจากการเรียนรู้		ออกจากการเรียนรู้ ออกจากการเรียนรู้ระบบจะปิดหน้าต่าง เช่น หน้าจอ "เนื้อหา" และการเรียนรู้

บทนำ

ข้อควรระวังในการใช้งาน

ข้อควรระวังด้านความปลอดภัย

เมื่อคุณเรียนรู้โดยการใช้งานผลิตภัณฑ์จริง โปรดอ่านข้อควรระวังต่างๆ ในคู่มือการใช้งานอย่างละเอียดให้เข้าใจ

ข้อควรระวังในหลักสูตรนี้

- หน้าจอที่แสดงของเวอร์ชันที่คุณใช้อาจจะแตกต่างจากในหลักสูตรนี้

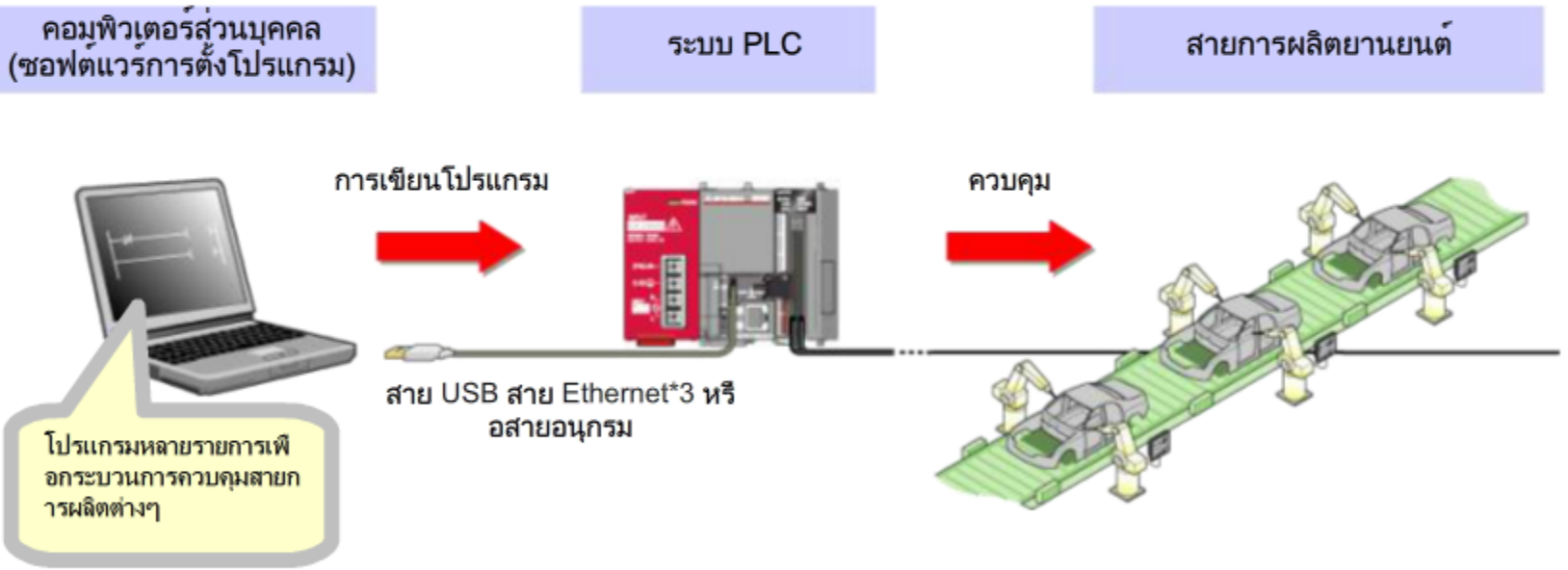
บทที่ 1 วิธีการควบคุมระบบ PLC

หลักสูตรนี้จัดทำขึ้นสำหรับผู้ที่ใช้งานซอฟต์แวร์วิศวกรรม หลักสูตรนี้ครอบคลุมแนวคิดพื้นฐานเกี่ยวกับการจัดการระบบ MELSEC-Q, L และ F ซีรีส์

GX Works 2 (GXW2) ใช้ภาษาโปรแกรมมาตรฐานนานาชาติ รวมถึง Sequential Function Chart (SFC) language, Instruction List (IL)*1, Ladder Logic, Function Block Diagram (FBD)*2 และ Structured Text (ST)

โปรแกรมนี้สร้างขึ้นโดยใช้คอมพิวเตอร์ส่วนบุคคลในการรัน "ซอฟต์แวร์งานวิศวกรรม" GX works2 และมักจะเขียนลง CPU ตัวควบคุมที่สามารถตั้งโปรแกรมได้โดยผ่านทางสาย USB สาย Ethernet*3 หรือสายอนุกรม อาจจะต้องการตั้งโปรแกรมโมดูล CPU ใหม่ตามที่จำเป็นเพื่อประยุกต์ให้เข้ากับการเปลี่ยนแปลงที่ต้องใช้ในการควบคุมที่ตองการ

*1 แผนในอนาคตสำหรับ GX works2
*2 ในปัจจุบันจะเรียกว่า Ladder แบบโครงสร้าง (Structured Ladder) ใน GX works2 ตามแผนการมาตรฐาน IEC
*3 Ethernet เป็นเครื่องหมายการค้าจดทะเบียนของ Xerox Corp



ในหลักสูตรนีจะใช้โลจิก Ladder (หนึ่งในภาษาโปรแกรม PLC ที่ได้รับความนิยมที่สุด) ในโปรแกรมตัวอย่าง แม้ว่าตัวอย่างจะใช้ L ซีรีส์ PLC แต่เนื้อหาของหลักสูตรนีจะใช้ได้กับระบบ Q ซีรีส์ ทั้งหมดด้วย

วิธีการควบคุมพื้นฐานเป็นวิธีการเดียวกันกับ MELSEC-F ซีรีส์ แต่การทำงานและฟังก์ชันบางอย่างแตกต่างกัน

1.1 กระบวนการสร้างระบบ PLC

หลักสูตรอิเล็กทรอนิกส์นี้จะครอบคลุมถึงขั้นตอนการออกแบบซอฟต์แวร์ (แสดงเป็นสีเขียว) ที่จำเป็นสำหรับการดำเนินการระบบการควบคุมแบบตั้งโปรแกรมได

การออกแบบฮาร์ดแวร์

(1) การออกแบบระบบ ..... หลักสูตรพื้นฐาน MELSEC-Q/MELSEC-L

(2) การเลือผลิตภัณฑ์ ..... หลักสูตรพื้นฐาน MELSEC-Q/MELSEC-L

(3) การเตรียมการขั้นสูง ..... หลักสูตรพื้นฐาน MELSEC-Q/MELSEC-L

(4) การติดตั้งและการเดินสายไฟ ... หลักสูตรพื้นฐาน MELSEC-Q/MELSEC-L

(5) การตรวจสอบการเดินสายไฟ ... หลักสูตรพื้นฐาน MELSEC-Q/MELSEC-L

การออกแบบซอฟต์แวร์

(6) การออกแบบโปรแกรม ..... บทที่ 2

(7) การตั้งโปรแกรม ..... บทที่ 3

(8) การแก้ไขความผิดพลาด ..... บทที่ 4

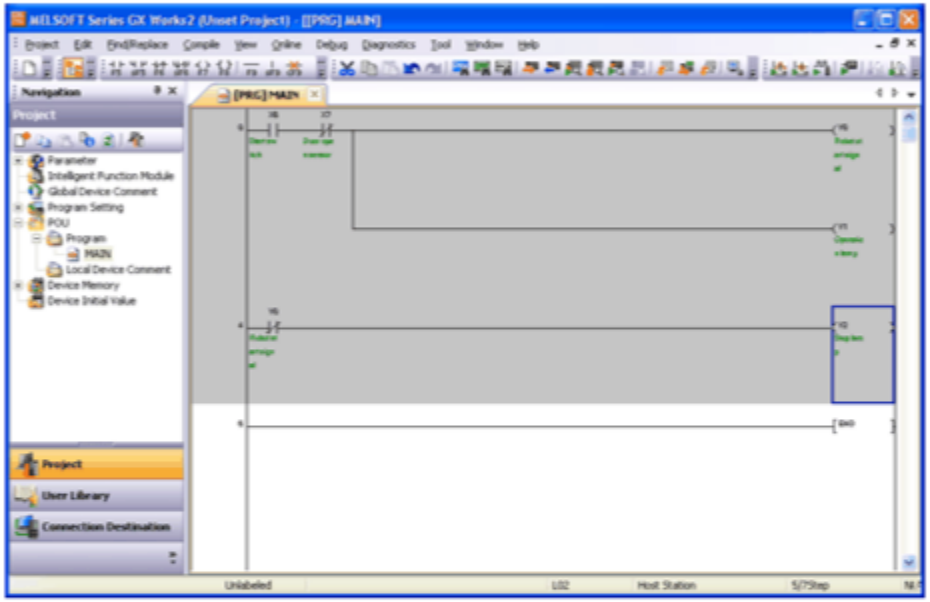
ขอบเขตของ

(9) การดำเนินการ

1.2 สิ่งที่ต้องมีสำหรับการตั้งโปรแกรม

ในหลักสูตรนี้ เราจะมุ่งเน้นที่วิธีการใช้ซอฟต์แวร์วิศวกรรมตัวควบคุมแบบตั้งโปรแกรมได้ GX Works2 เพื่อพัฒนาโปรแกรมระบบตัวอย่าง ฟังชั่นหลักๆ ของ GX Works2 ได้แก่

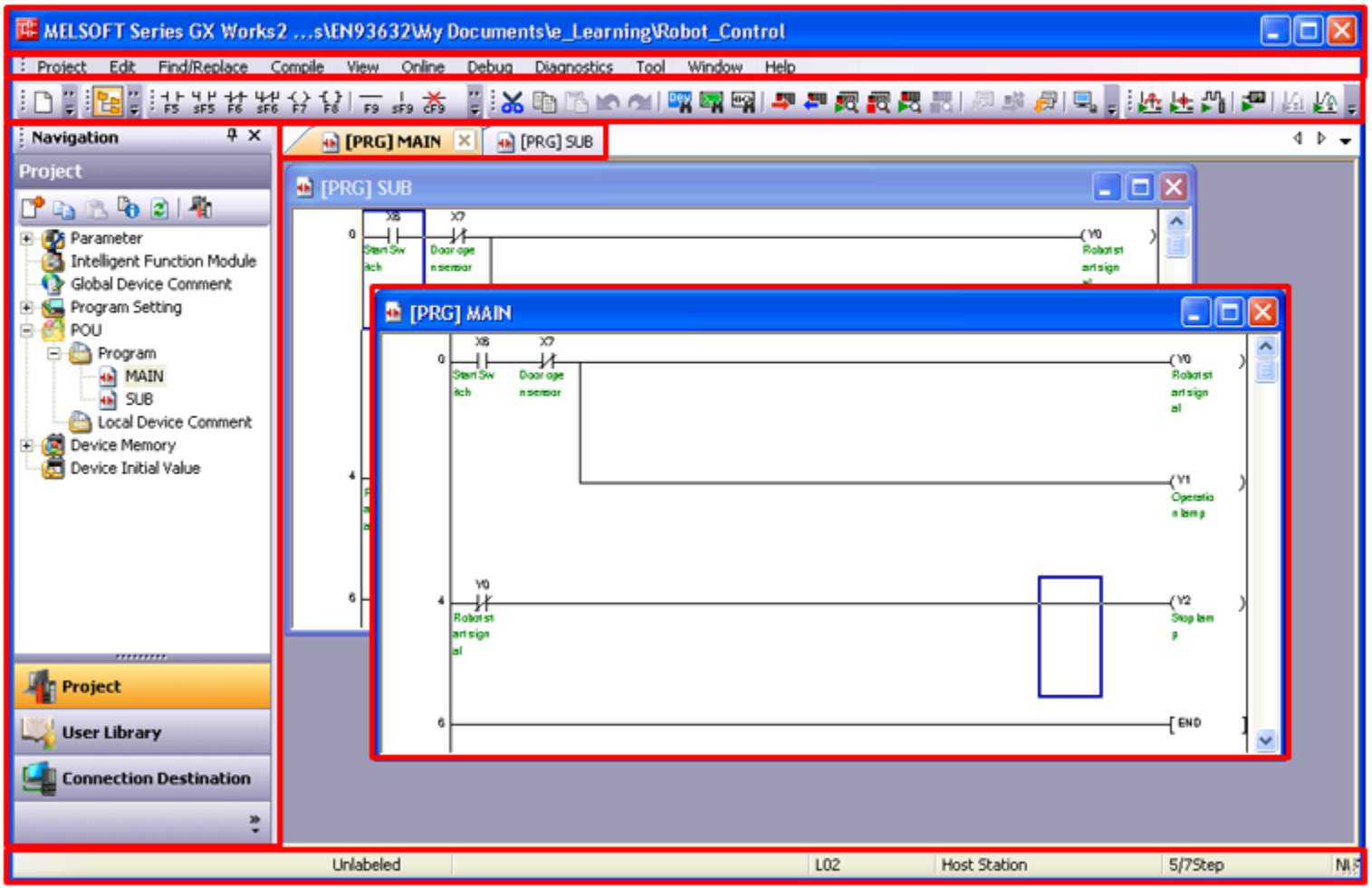
- การจัดการหน่วยความจำและไฟล์
- การพัฒนาโปรแกรมตัวควบคุมแบบตั้งโปรแกรมได้
- การจัดการเอกสารของโปรแกรม (ความคิดเห็น ฯลฯ)
- การอื่นข้อมูลการเขียน (โดยเฉพาะอย่างยิ่งโปรแกรม) จาก/สู่โมดูล CPU
- การตรวจสอบการดำเนินการโปรแกรม
  - การจำลองซอฟต์แวร์ของฮาร์ดแวร์ PLC
  - บังคับ I/O เปิดหรือปิด
  - ตรวจสอบสถานะ I/O และที่อยู่หน่วยความจำ
- ดำเนินการบำรุงรักษา และการแก้ไขปัญหา



1.3 การกำหนดค่าหน้าจอ GX Works2

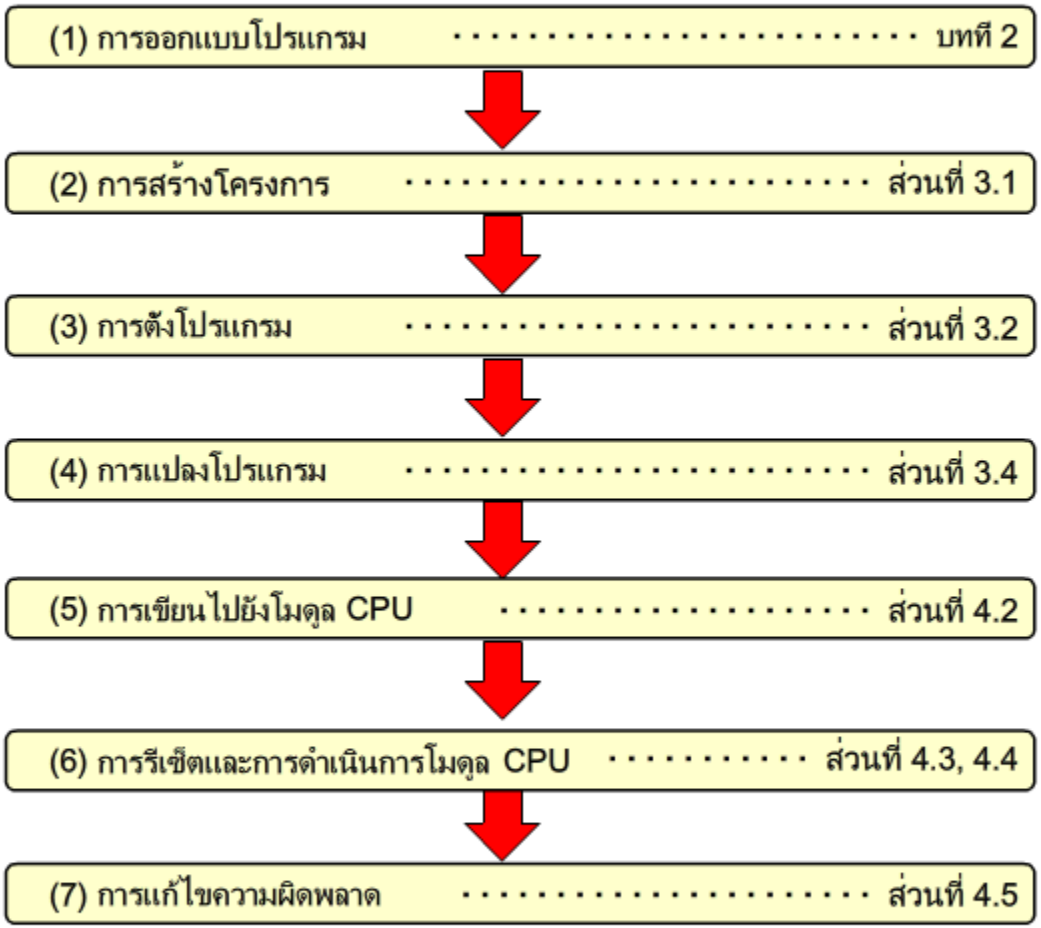
ด้านล่างนี้คือการกำหนดค่าหน้าจอ GX Works2

วางเคอร์เซอร์เมาส์ในกรอบสีแดงเพื่อแสดงฟังก์ชั่นที่เกี่ยวข้อง



1.4 กระบวนการสร้างโปรแกรมลำดับ

สร้างโปรแกรมลำดับด้วยกระบวนการดังต่อไปนี้



บทที่ 2 การสร้างข้อมูลหน้าจอ

ในบทที่ 2 คุณจะได้เรียนรู้เกี่ยวกับวิธีการออกแบบโปรแกรม รวมถึง การกำหนดเนื้อหาของการควบคุมและการแปลงให้เป็นโปรแกรม

การออกแบบโปรแกรม ..... บทที่ 2

การตั้งโปรแกรม ..... บทที่ 3

การแก้ไขความผิดพลาด ..... บทที่ 4

ขั้นตอนการเรียนรู้ในบทที่ 2

2.1 การกำหนดค่าฮาร์ดแวร์ของตัวอย่างระบบที่ใช้สำหรับการเรียนรู้

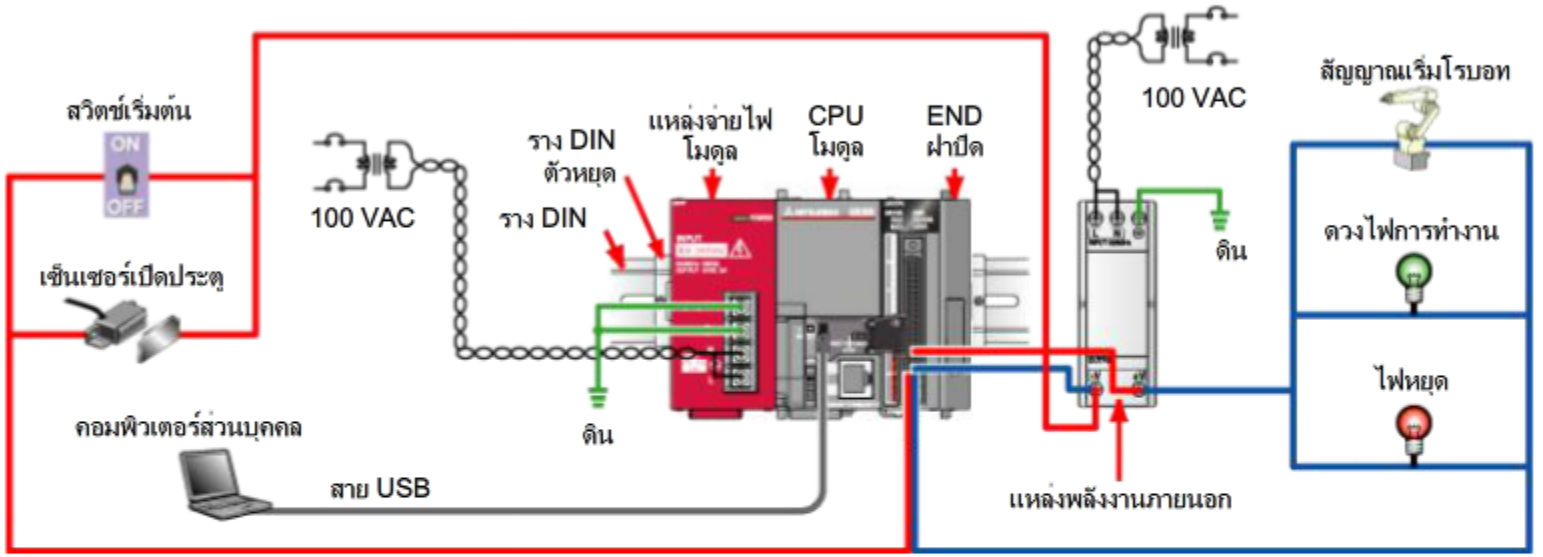
2.2 การกำหนดรายการควบคุม

2.3 การสร้างตารางการตอบสนองของ I/O อุปกรณ์และหมายเลขอุปกรณ์

2.4 การออกแบบโปรแกรม

2.1 การกำหนดค่าฮาร์ดแวร์ของระบบตัวอย่างที่ใช้ในการเรียนรู้

ในหลักสูตรนี้ คุณจะได้สร้างระบบ PLC (ในที่นี้จะเรียกว่า "ระบบตัวอย่าง") ซึ่งจะเริ่มโรบอทตามกระบวนการด้านล่างนี้คือไดอะแกรมของการกำหนดค่าฮาร์ดแวร์ของระบบตัวอย่างและรายการส่วนประกอบฮาร์ดแวร์



รายการ	ส่วนประกอบ	รุ่น	รายละเอียด
ระบบ PLC	โมดูลแหล่งพลังงาน	L61P	จ่ายพลังงานให้แก่โมดูลต่างๆ รวมถึง โมดูล CPU และโมดูล I/O
	โมดูล CPU	L02CPU	ควบคุมระบบ PLC
	ฝาปิด END	L6EC	ติดตั้งที่ด้านขวาของบล็อกระบบ
	สาย USB	MR-J3USBCBL3M	เชื่อมต่อกับคอมพิวเตอร์ส่วนบุคคลที่ติดตั้ง GX Works2 เข้ากับโมดูล CPU
	คอมพิวเตอร์ส่วนบุคคล	–	รันด้วย GX Works2 ที่ติดตั้ง
แหล่งพลังงานภายนอก	–	–	จ่ายพลังงานให้แก่อุปกรณ์ I/O ภายนอก
อุปกรณ์ I/O ภายนอก	สวิตช์	–	ตั้งค่าเป็น ON เพื่อเริ่มการควบคุม
	เซ็นเซอร์	–	ตรวจสอบว่าประตูเปิดหรือปิด
	โรบอท	–	ดำเนินการตามสัญญาณการควบคุม
	ดวงไฟสองดวง	–	ดวงไฟแสดงสถานะการทำงาน

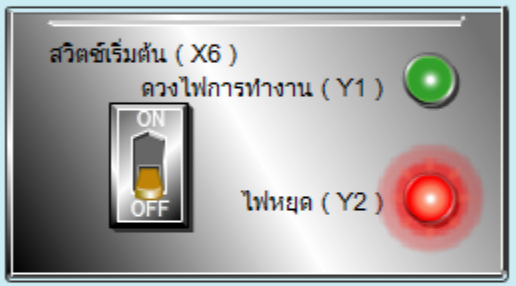
2.2 การกำหนดรายการควบคุม

ขั้นตอนแรกของการออกแบบโปรแกรมคือการระบุอุปกรณ์ที่ต้องการควบคุมและอุปกรณ์ I/O ที่จำเป็นสำหรับการควบคุมที่ต้องการ ในระบบตัวอย่างจะมีการควบคุมการเริ่มและการหยุดของโรบอท หากประตูของรั้วนิรภัยเปิด ระบบจะป้องกันไม่ให้โรบอทเริ่มทำงาน และจะโรบอทจะหยุดทำงานหากประตูเปิดในระหว่างทำงาน
โปรดดูภาพด้านล่างเพื่อทำความเข้าใจวิธีการทำงานของระบบตัวอย่าง

การทำงานของระบบตัวอย่าง

คลิกในวงกลมสีแดง

แผงควบคุมโรบอท



โรบอทในรั้วนิรภัย

เมื่อตั้งค่า สวิตช์เริ่มต้น (X6) เป็น OFF โรบอทเริ่มทำงาน (Y0) จะเปิดและโรบอทจะเริ่มทำงาน ในเวลาเดียวกัน ดวงไฟการทำงาน (Y1) ที่แผงควบคุมจะดับและ ไฟหยุด (Y2) จะติด

เล่นซ้ำ

ก่อนหน้า

2.3 การสร้างตารางตอบสนองของอุปกรณ์ I/O และหมายเลขอุปกรณ์

เป็นความคิดที่ดีที่จะสร้างตารางที่มีอุปกรณ์ I/O และการลงทะเบียนทั้งหมดที่ใช้ใน PLC และข้อมูลที่ตอบสนองสำหรับโปรแกรมต่างๆ ที่สร้างขึ้น การดำเนินการดังกล่าวจะช่วยลดความผิดพลาดที่เกิดขึ้นระหว่างกระบวนการออกแบบและการตั้งโปรแกรม และจะช่วยเพิ่มประสิทธิภาพการตั้งโปรแกรม หากมีตารางการตอบสนองของระบบอยู่แล้ว เช่น ตารางที่สร้างโดยผู้ที่กำหนดค่าฮาร์ดแวร์ โปรดใช้ตารางดังกล่าว

ตารางด้านล่างนี้คือตารางตอบสนองของระบบตัวอย่างที่ใช้ในหลักสูตรนี้

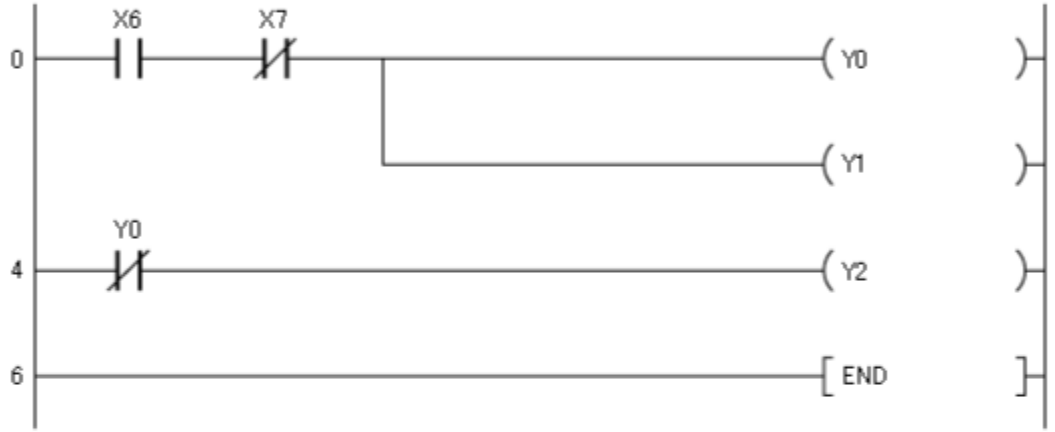
ชื่ออุปกรณ์ I/O	หมายเลขอุปกรณ์	ชนิด I/O	ชนิดอุปกรณ์	รายละเอียด
Start switch (สวิตช์เริ่มต้น)	X6	อินพุท	บิต	สวิตช์นี้จะเริ่มหรือหยุดการทำงานของโรบอท
Door open sensor (เซ็นเซอร์เปิดประตู)	X7	อินพุท	บิต	เซ็นเซอร์นี้จะตรวจสอบว่าประตูของรั้วนิรภัยของโรบอทเปิดหรือไม่ เมื่อประตูเปิด ระบบจะเปิดเซ็นเซอร์ เมื่อประตูปิด ระบบจะปิดเซ็นเซอร์
Robot start signal (สัญญาณเริ่มโรบอท)	Y0	เอาท์พุท	บิต	เมื่อสัญญาณนี้เปิด โรบอทจะเริ่มการทำงาน
Operation lamp (ดวงไฟการทำงาน)	Y1	เอาท์พุท	บิต	ดวงไฟนี้จะติดเมื่อโรบอททำงาน
Stop lamp (ไฟหยุด)	Y2	เอาท์พุท	บิต	ดวงไฟนี้จะติดเมื่อโรบอทหยุด

*หากมีการใช้ข้อมูลคำ ต้องรวมค่าเริ่มต้น ช่วงการตั้งค่า (จำกัดสูงสุดและต่ำสุด) ชนิดข้อมูล (ลงชื่อ จริง ฯลฯ) และความคิดเห็นไว้ในตารางด้วย ข้อมูลนี้จะมีประโยน์สำหรับการออกแบบและการแก้ไขโปรแกรม

2.4 การออกแบบโปรแกรม

การออกแบบโปรแกรมโดยใช้ภาษาโลจิก Ladder ตามรายการควบคุมและตารางการตอบสนอง I/O ต่อไปนี้คือโปรแกรม Ladder และตารางการตอบสนอง I/O ที่ออกแบบมาเพื่อระบบตัวอย่าง

โปรแกรม Ladder

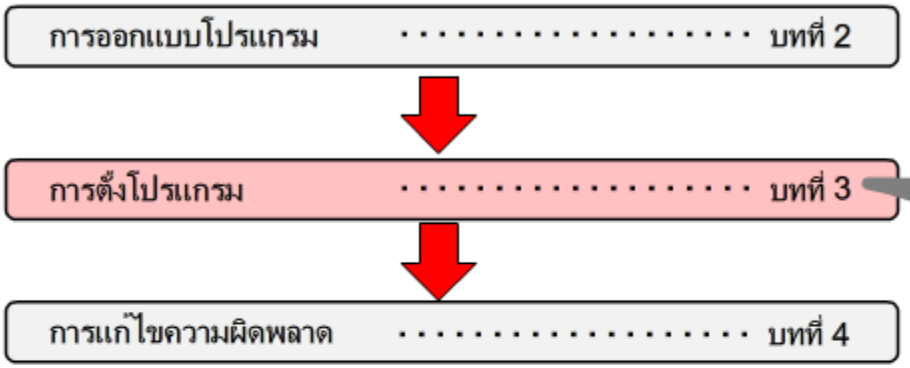


ตารางการตอบสนอง I/O

ชื่ออุปกรณ์ I/O	ชนิด	หมายเลขอุปกรณ์
Start switch (สวิตช์เริ่มต้น)	อินพุท	X6
Door open sensor (เซ็นเซอร์เปิดประตู)	อินพุท	X7
Robot start signal (สัญญาณเริ่มโรบอท)	เอาท์พุท	Y0
Operation lamp (ดวงไฟการทำงาน)	เอาท์พุท	Y1
Stop lamp (ไฟหยุด)	เอาท์พุท	Y2

บทที่ 3 การตั้งโปรแกรม

ในบทที่ 3 คุณจะได้เรียนรู้เกี่ยวกับวิธีการตั้งโปรแกรมโปรแกรมการออกแบบโดยใช้ GX Works2



ขั้นตอนการเรียนรู้ในบทที่ 3

3.1 การสร้างโครงการ
3.2 การสร้างโครงการ
3.3 การทำให้โปรแกรมสามารถเข้าใจได้ง่าย
3.4 การแปลงโปรแกรมให้อยู่ในรูปแบบที่ดำเนินการได้
3.5 การบันทึกโครงการ

3.1 การสร้างโครงการ

ขั้นตอนแรกของการเขียนโปรแกรมคือการสร้างโครงการ
โครงการคือชุดข้อมูล GX Works2 ที่ใช้เพื่อจัดการโปรแกรม
ตารางต่อไปนี้จะแสดงส่วนประกอบที่สำคัญของโครงการ

ชนิดข้อมูล	รายละเอียด
โปรแกรม	รหัสแหล่งข้อมูลและรหัสการเขียนสำหรับการดำเนินการลำดับ CPU
ความคิดเห็น	ชนิดของเอกสารที่แสดงในโปรแกรม โปรดดูรายละเอียดที่ส่วนที่ 3.3 "การทำให้โปรแกรมเข้าใจได้ง่าย"
พารามิเตอร์	มีข้อมูลการตั้งค่าและการกำหนดค่าส่วนใหญ่หรือทั้งหมดสำหรับระบบ
โอนถ่ายการตั้งค่า	ข้อมูลเส้นทางการเชื่อมต่อเป็นสิ่งที่จำเป็นสำหรับการสร้างการสื่อสารระหว่างระบบที่ใช้งาน GX Works2 และโมดูล CPU

โปรแกรม Ladder

GX Works2 จะช่วยให้คุณสามารถเลือกชนิดโครงการได้สองชนิดต่อไปนี้
โปรแกรมตัวอย่างในหลักสูตรนี้จะใช้ชนิด "simple project (โครงการแบบเรียบง่าย)"

ชนิดโครงการ	รายละเอียด
Simple project (โครงการแบบเรียบง่าย)	ชนิดโครงการนี้จะสามารถใช้งานได้กับโครงการ GX Developer โครงการแบบเรียบง่ายจะสามารถแปลงเป็นโครงการแบบโครงสร้างได้ในภายหลัง แต่จะไม่สามารถแปลงกลับมาได้
Structured project (โครงการแบบโครงสร้าง)	โครงการนี้จะสามารถใช้ภาษาการตั้งโปรแกรมเพิ่มเติมที่เรียกว่า Ladder แบบโครงสร้างได้ นอกจากนี้ จะสามารถแบ่งโปรแกรมเป็นส่วนเล็กๆ และจะสามารถจัดทำโมดูลและใช้ซ้ำรหัสที่ใช้งานบ่อยๆ ได้อย่างง่ายดายโดยใช้ไลบรารีผู้ใช้ สามารถจัดทำฉลากในรูปแบบเดียวกันเพื่อใช้งานซ้ำได้อย่างง่ายดาย ซึ่งจะเป็นการเพิ่มประสิทธิภาพการตั้งโปรแกรมและการแก้ไขปัญหาได้ โดยเฉพาะอย่างยิ่งสำหรับโครงการขนาดใหญ่

ฉลาก

ฉลากคือชื่อที่ผู้ใช้สร้างคืนที่จะกลายเป็นนามแฝงสำหรับที่อยู่อุปกรณ์ สามารถใช้ฉลากได้ทั่วโลก ในท้องถิ่น หรือในระบบเมื่อใช้งานร่วมกับ MELSOFT Navigator สามารถสร้างโครงการแบบเรียบง่ายโดยมีหรือไม่มีความสามารถการใช้งานฉลากได้ด้วย สำหรับโครงการตัวอย่างจะไม่มีการใช้งานฉลาก

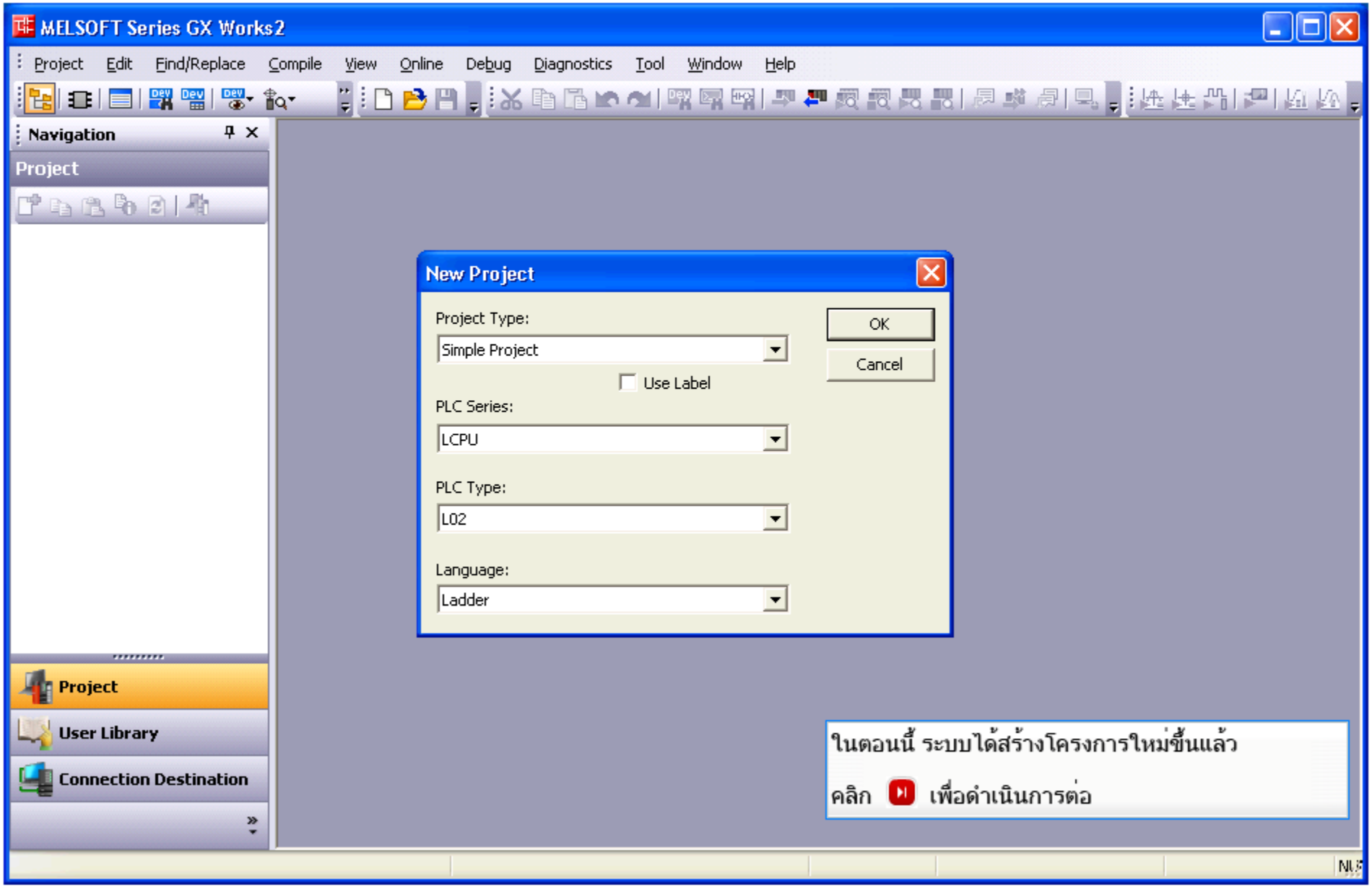
3.1 การสร้างโครงการ

เพื่อเริ่มต้นสร้างโครงการตัวอย่าง ให้ดำเนินการตั้งค่าดังต่อไปนี้
ก่อนการสร้างโครงการ ควรทราบชื่อรุ่นและชุดตัวควบคุมแบบตั้งโปรแกรมได้ รวมถึงชนิดโครงการที่ต้องการใช้งาน

รายการ	รายละเอียด
ชนิดโครงการ	ชนิดโครงการจะเป็นการกำหนดว่าจะมีคุณสมบัติใดบ้างเมื่อเขียนโปรแกรม สำหรับตัวอย่างนี้ โปรดเลือก "simple project"
ใช้ฉลาก	หากต้องใช้ความสามารถในการเขียนโปรแกรมโดยใช้ฉลาก โปรดเลือกรายการนี้ โปรแกรมตัวอย่างจะไม่ใช้ฉลาก ดังนั้น เราจึงไม่ทำเครื่องหมายในกล่องนี้
PLC ซีรีส์	PLC ซีรีส์ จะกำหนดรุ่นที่สามารถเลือกได้ในรายการดร็อปดาวน์ชนิด PLC สำหรับตัวอย่างนี้ โปรดเลือก "LCPU"
ชนิด PLC	ชนิด PLC จะกำหนดว่าจะแปลงโปรแกรมผู้ใช้เป็นรหัสเครื่องอย่างไร เลือกรุ่น PLC ที่จะโปรแกรม ในกรณีนี้คือ "L02"
ภาษาการตั้งโปรแกรม	ภาษาการตั้งโปรแกรมจะกำหนดชนิดของโปรแกรมของโปรแกรมที่สร้างขึ้นโดยอัตโนมัติเป็นครั้งแรก (MAIN) สามารถเพิ่มโปรแกรมเพิ่มเติมที่ใช้ภาษาอื่นๆ ได้ในภายหลัง สำหรับตัวอย่างนี้ โปรดเลือก "Ladder"

โปรดดูหน้าต่อไป ซึ่งจะมีการแสดงกระบวนการสร้างโครงการใหม่

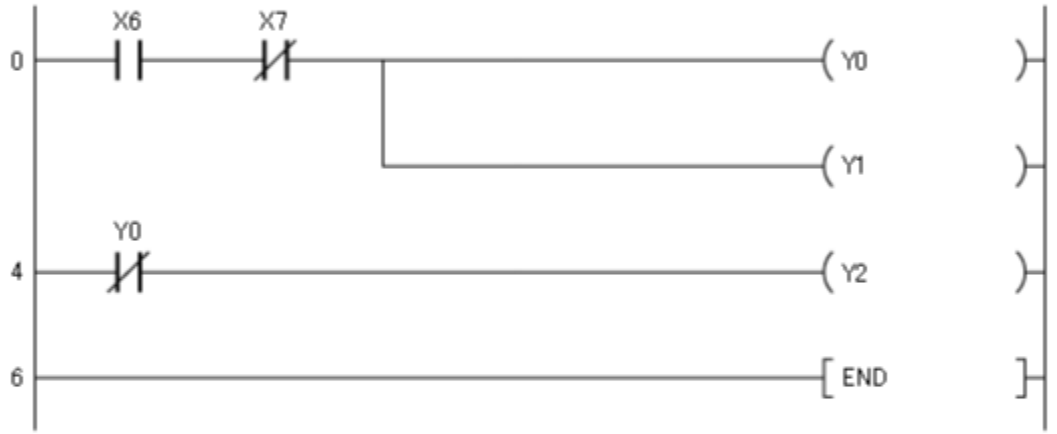
3.1 การสร้างโครงการ



3.2 การสร้างโปรแกรม

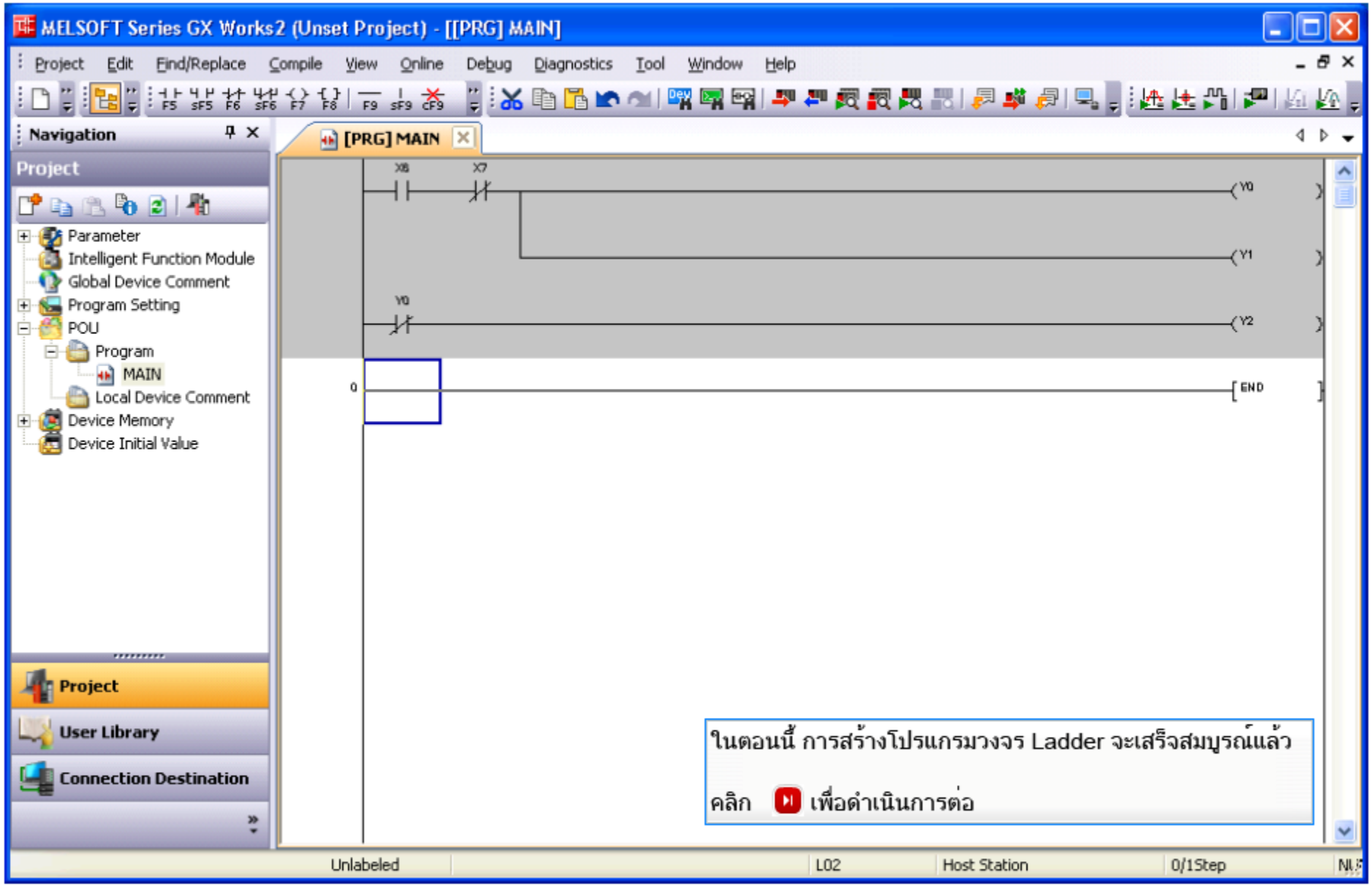
หลักจากสร้างโครงการแล้ว เรามาสร้างโปรแกรมกัน
สร้างโปรแกรมต่อไปนี้และเรียนรู้การทำงานพื้นฐาน (ใส่คำสั่ง เปลี่ยนแปลง ลบ คัดลอกและวาง และใส่/ลบเส้นควบคุม)
ด้านล่างนี้คือโปรแกรมที่ออกแบบสำหรับระบบตัวอย่างในบทที่ 2

โปรแกรมสำหรับระบบตัวอย่าง



ในหน้าต่อไป ให้ลองสร้างโปรแกรมนี้โดยใช้หน้าต่างจำลอง

3.2 การสร้างโปรแกรม



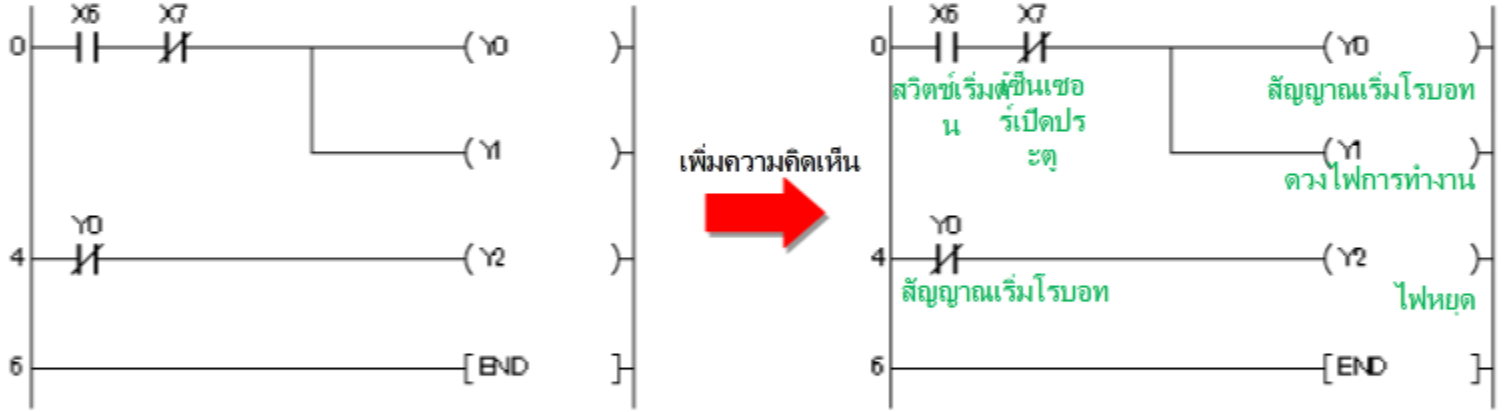
3.3 การทำให้โปรแกรมสามารถเข้าใจได้ง่าย

ในขั้นตอนปัจจุบัน การแสดงภาพของโปรแกรมจะมีเพียงอุปกรณ์ คำสั่ง เส้น และหมายเลขขั้นตอน เมื่อเราดูโปรแกรมที่มีความซับซ้อน อาจจะเป็นการยากที่พิจารณาว่าโปรแกรมกำลังทำอะไรอยู่

- การค้นหาความผิดปกติของการตั้งโปรแกรม เช่น หมายเลขอุปกรณ์หรือคำสั่งไม่ถูกต้อง นั้นกระทำได้ยาก
- โดยรวมแล้ว การประเมินการทำงาน การแก้ไขความผิดพลาด และการขยายโปรแกรมจะกระทำได้ยาก
- หากผู้พัฒนาดั้งเดิมของโปรแกรมไม่ได้เป็นผู้ดูและโปรแกรมอีกต่อไปแล้ว งานการเรียนรู้วิธีการทำงานของโปรแกรมสำหรับบุคคลอื่นๆ อาจจะกระทำได้อยากหรืออาจจะเป็นไปไม่ได้เลย

มาตรการแก้ไข

รวม เอกสาร ไว้ในโปรแกรมเพื่อช่วยให้ทุกๆ คนสามารถเข้าใจวิธีการทำงานของโปรแกรมได้
เพื่อการใช้งานอย่างถูกต้อง ผู้เขียนโปรแกรมทุกคนควรเพิ่มรายละเอียดความเห็นไว้ในโปรแกรมเพื่อช่วยให้ตัวเองหรือบุคคลอื่นสามารถเข้าใจโปรแกรมได้ดีขึ้น

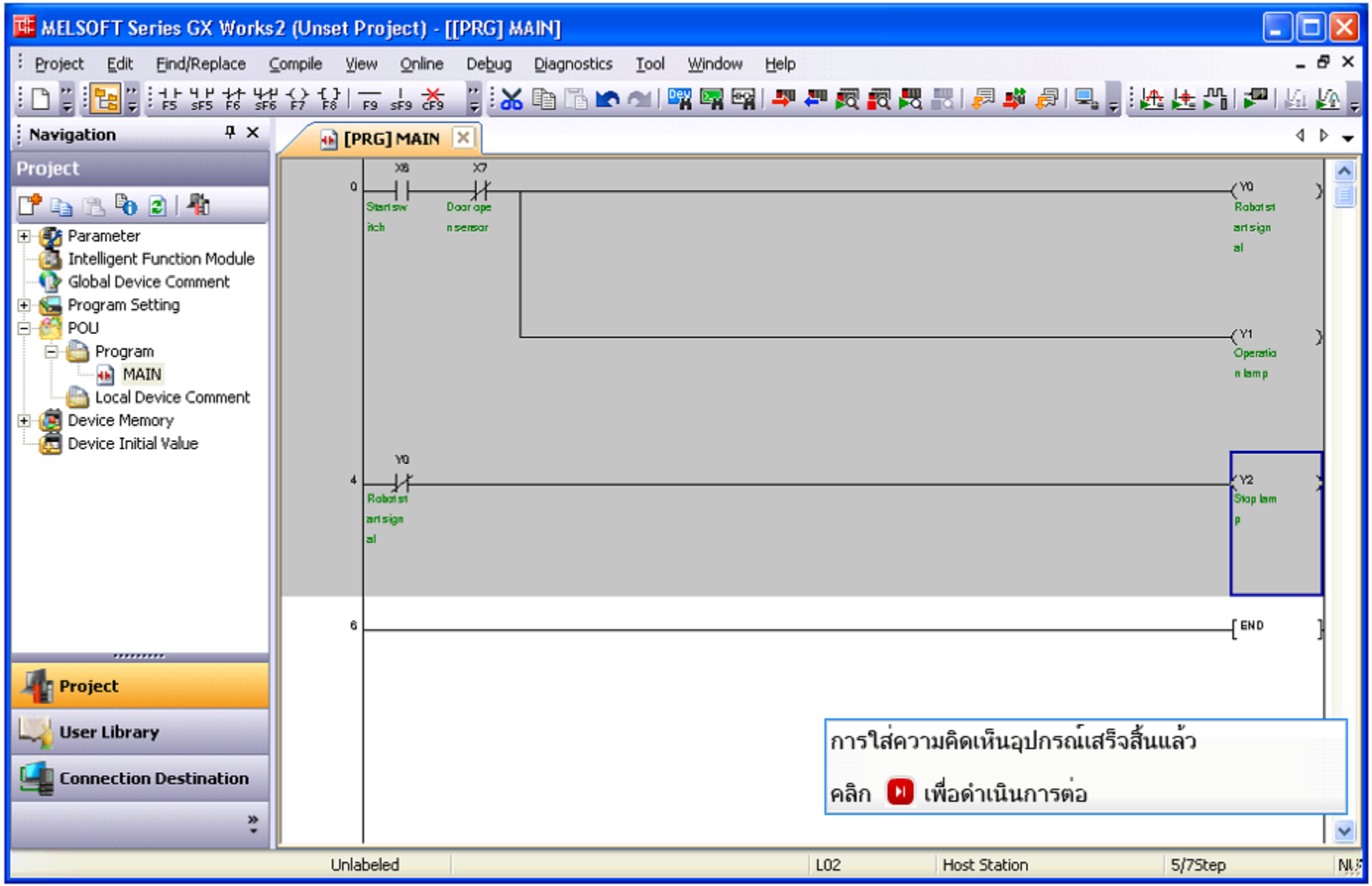


GX Works2 ช่วยให้สามารถใช้ความคิดเห็นได้สามชนิด
สำหรับรายละเอียดอื่นๆ โปรดดูที่คู่มือ GX Works2 โครงการแบบง่าย

ชนิดความคิดเห็น	ขอบเขตความคิดเห็น
Device comment (ความคิดเห็นอุปกรณ์)	ใส่ความคิดเห็น 32 ตัวเพื่อแสดงในอุปกรณ์ที่เลือก (I/O หรือที่อยู่หน่วยความจำอื่นๆ)
Statement (ข้อความ)	ใส่ข้อความไม่เกิน 64 ตัวอักษรเพื่อเพิ่มไว้ที่ด้านบนของบล็อก Ladder ที่เลือก (เหนือหมายเลขขั้นตอน) บล็อก Ladder แต่ละรายการจะสามารถมีข้อความได้หลายข้อความ
Note (หมายเหตุ)	ใส่ข้อความสูงสุด 32 ตัวอักษรเพื่อแสดงเหนือข้อความแอพพลิเคชั่นหรือคอยล์ที่เลือก

หน้าต่อไปจะแสดงขั้นตอนการเพิ่มความคิดเห็นอุปกรณ์สู่โปรแกรมตัวอย่าง

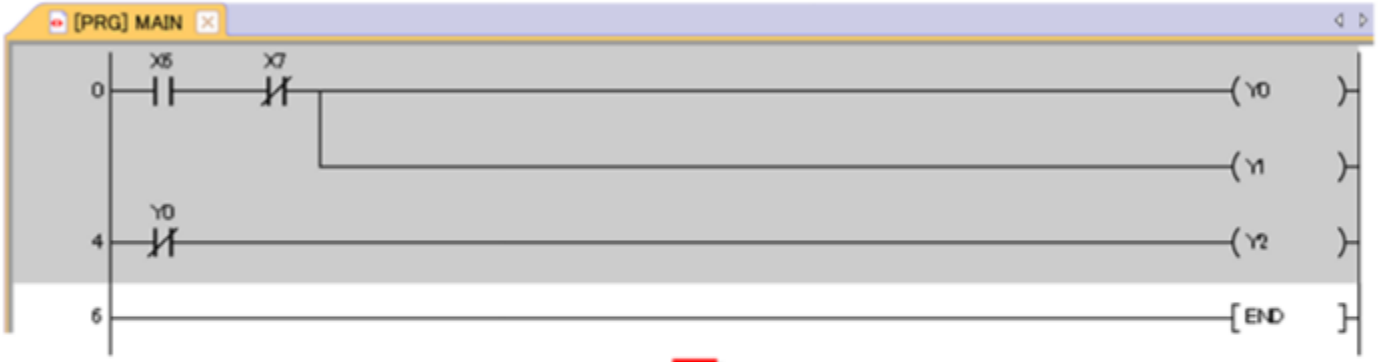
3.3 การทำให้โปรแกรมสามารถเข้าใจได้ง่าย



3.4 การแปลงโปรแกรมให้อยู่ในรูปแบบที่ดำเนินการได้

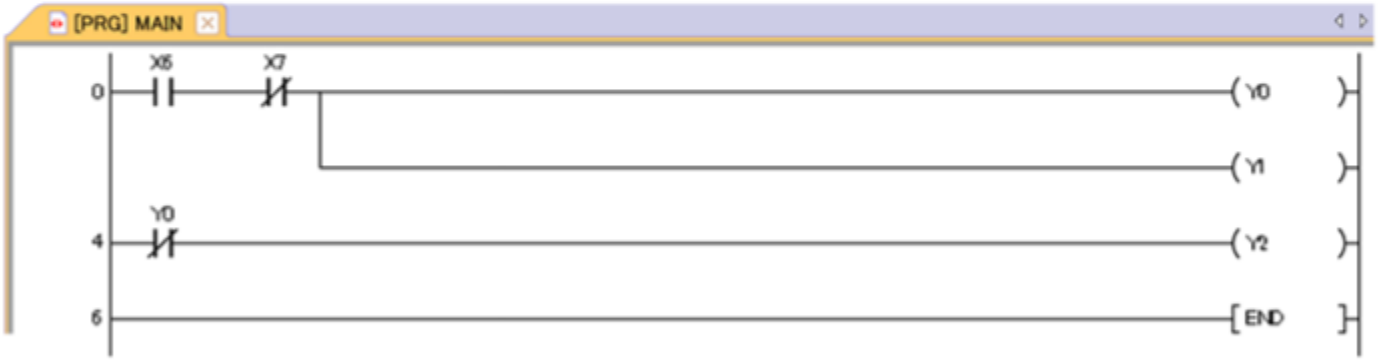
หลักจากที่ตั้งโปรแกรมเสร็จแล้ว คุณจะต้องแปลงโปรแกรมให้อยู่ในรูปแบบที่สามารถดำเนินการในโมดูล CPU ได้ ไม่สามารถดำเนินการหรือบันทึกโปรแกรมที่ยังไม่ได้แปลงได้

สีพื้นหลังของโปรแกรมที่ยังไม่ได้แปลงจะเป็นสีเทาดังที่แสดงไว้ด้านล่าง



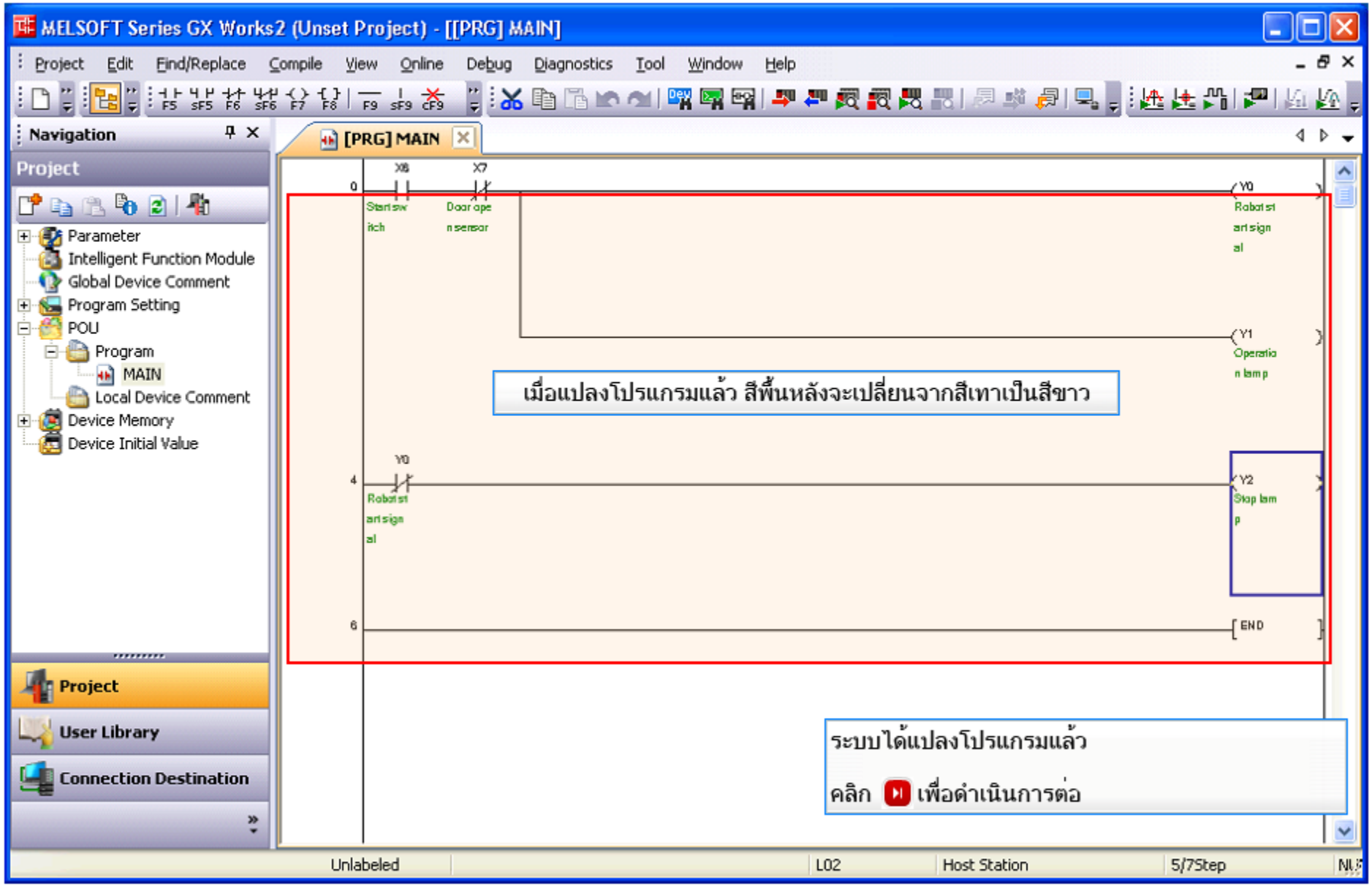
แปลง

หลังจากการแปลง สีพื้นหลังจะเปลี่ยนเป็นสีขาวตามที่แสดงไว้ด้านล่าง



ในหน้าต่อไป ให้ลองแปลงโปรแกรมนี้โดยใช้หน้าต่างจำลอง

3.4 การแปลงโปรแกรมให้อยู่ในรูปแบบที่ดำเนินการได้



3.5 การบันทึกโครงการ

หลังจากที่แปลงโปรแกรมแล้ว ให้บันทึกโครงการที่มีโปรแกรม หากสิ้นสุดการทำงานของ GX Works2 โดยไม่ได้บันทึกโครงการ โปรแกรมที่เกี่ยวข้องจะถูกทำลาย ดังนั้น คุณจึงควรบันทึกโครงการบ่อยๆ
เมื่อบันทึกโครงการใหม่ ให้ระบุชนิดต่อไปนี้ในข้อมูลโครงการ (ไม่จำเป็นต้องดำเนินการดังกล่าวสำหรับการบันทึกทับ)
คุณควรรวมข้อมูลที่จะช่วยให้ผู้อื่นสามารถเข้าใจเนื้อหาการควบคุมโปรแกรม ชื่อระบบ ฯลฯ ได้อย่างง่ายดาย

รายการ	ต้องใช้	รายละเอียด
พาธปลายทางการบันทึก	✔	ระบุโฟลเดอร์ของที่ต้องการให้เป็นที่ตั้งของพื้นที่การทำงาน
รายการโครงการ/พื้นที่การทำงาน		หากมีพื้นที่การทำงานหนึ่งรายการขึ้นไปอยู่แล้วในโฟลเดอร์ที่ระบุไว้ใน "พาธปลายทางการบันทึก" ระบบจะแสดงพื้นที่ปฏิบัติงานที่มีอยู่แล้ว
ชื่อพื้นที่การทำงาน	✔	ระบุชื่อพื้นที่การทำงานที่มีตัวอักษรสูงสุด 128 ตัว
ชื่อโครงการ	✔	ระบุชื่อโครงการที่มีตัวอักษรสูงสุด 128 ตัว
หัวข้อ		ระบุหัวข้อโครงการที่มีตัวอักษรสูงสุด 128 ตัว พารามิเตอร์นี้จะมีประโยชน์เมื่อคุณต้องการกำหนดชื่อที่ยาวกว่าที่สามารถใส่ไว้ในช่อง "ชื่อโครงการ" ได้

พื้นที่การทำงานคือโฟลเดอร์เพื่อใช้จัดกรโครงการหลายๆ โครงการ
ด้านล่างนี้คือตัวอย่างของการใช้งานพื้นที่การทำงาน (ระบบจะจัดการโครงการสำหรับชนิดยานพาหนะแต่ละชนิดในสายการผลิตยานยนต์)

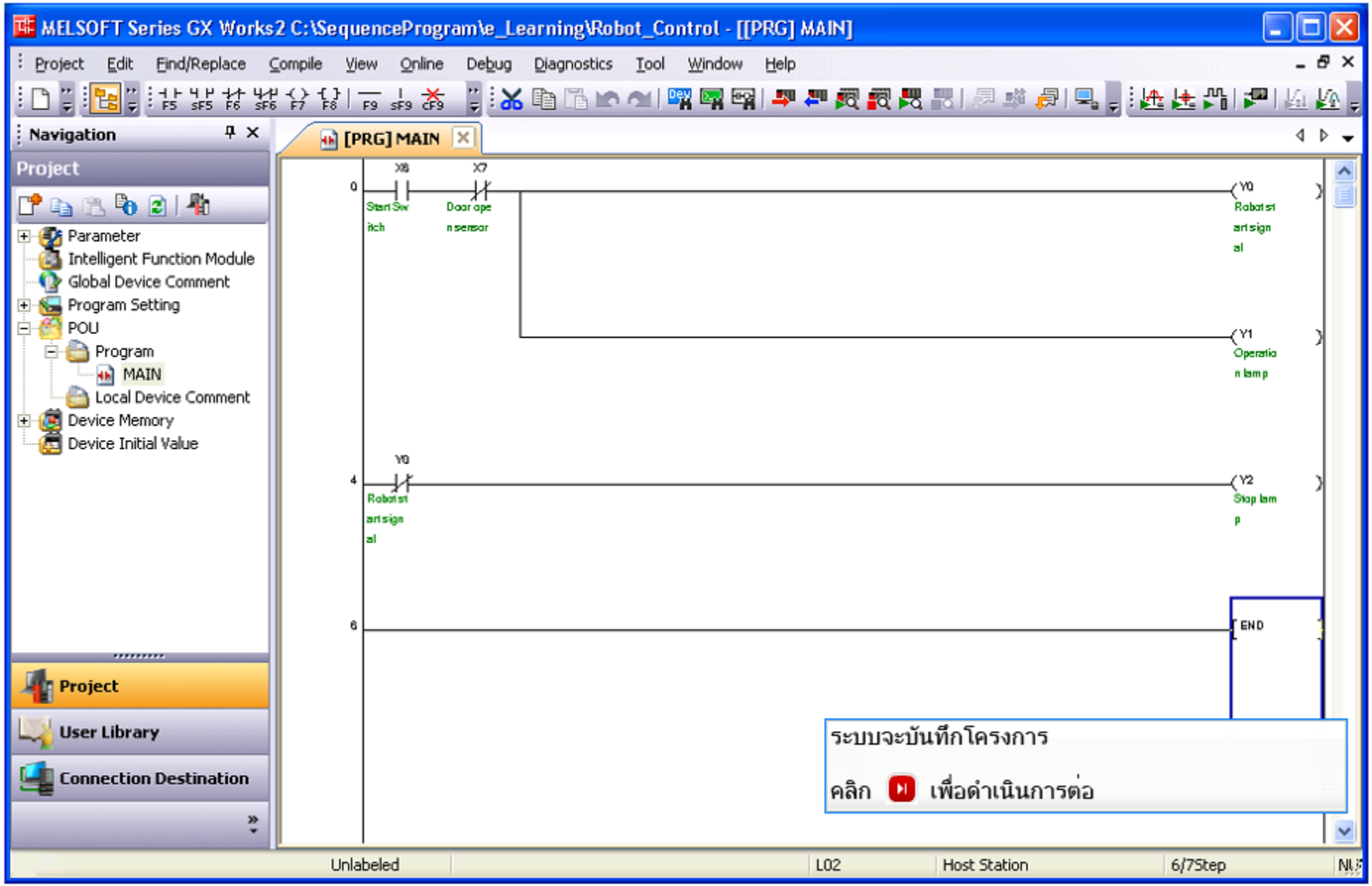
ชื่อพื้นที่การทำงาน	ชื่อโครงการ	หัวข้อ
สายการผลิตยานยนต์	สายการผลิตชนิด A	โปรแกรมการทำงานธรรมดาสำหรับการควบคุมสายการผลิตชนิด A
	สายการผลิตชนิด B	โปรแกรมการทำงานธรรมดาสำหรับการควบคุมสายการผลิตชนิด B
	สายการผลิตชนิด C	โปรแกรมการทำงานธรรมดาสำหรับการควบคุมสายการผลิตชนิด C

หมายเหตุ:

- หากคุณบันทึกโครงการที่มีโปรแกรมที่ยังไม่ได้แปลง ระบบจะทำลายโปรแกรมแต่โปรแกรมที่ยังไม่ได้แปลงเท่านั้น ก่อนบันทึกโครงการ ให้ดำเนินการแปลงโปรแกรมตามที่คุณได้เรียนรู้ในส่วนที่ 3.4
- ระบุพาธปลายทางที่บันทึก ชื่อพื้นที่การทำงาน และชื่อโครงการโดยมีจำนวนตัวอักษรไม่เกิน 150 ตัว

ในหน้าต่อไป ให้ลองบันทึกโครงการโดยใช้หน้าต่างจำลอง

3.5 การบันทึกโครงการ



บทที่ 4 การแก้ไขความผิดพลาด

ในบทที่ 4 คุณจะได้เรียนรู้วิธีการเขียนโปรแกรมลำดับไปยังโมดูล CPU และการแก้ไขความผิดพลาด

การออกแบบโปรแกรม ..... บทที่ 2

↓

การตั้งโปรแกรม ..... บทที่ 3

↓

การแก้ไขความผิดพลาด ..... บทที่ 4

ขั้นตอนการเรียนรู้ในบทที่ 4

4.1 การแก้ไขความผิดพลาด
- 4.1.1 การแก้ไขความผิดพลาดโปรแกรมโดยไม่ใช้โมดูล CPU
- 4.1.2 การเปลี่ยนสถานะของอุปกรณ์ I/O
- 4.1.3 การตรวจสอบสถานะของอุปกรณ์

4.2 การเขียนโปรแกรมไปยังโมดูล CPU
4.3 การเปิดใช้งานโปรแกรมที่เขียน
4.4 การรันโปรแกรม
4.5 การแก้ไขความผิดพลาดโปรแกรม
4.6 การตรวจสอบการทำงานระบบ PLC
4.7 การใช้งานระบบ PLC
4.8 สรุป

4.1 การแก้ไขความผิดพลาดคืออะไร?

เมื่อโปรแกรมหรือส่วนใดส่วนหนึ่งของโปรแกรมถูกเขียนขึ้นมา เป็นสิ่งจำเป็นที่ต้องมีการทดสอบรหัสเพื่อให้แน่ใจว่าโปรแกรมมีการทำงานตามที่คาด

ความผิดพลาดของซอฟต์แวร์ (เมื่อรหัสที่เขียนไม่ทำงานตามที่ต้องการ) เราจะเรียกว่า "bugs (ความผิดพลาด) " และกระบวนการในการค้นหาสาเหตุและการแก้ไขความผิดพลาดจะเรียกว่า "debugging (การแก้ไขความผิดพลาด)"

การทดสอบและการแก้ไขความผิดพลาดเป็นขั้นตอนที่จำเป็นในการสร้างโปรแกรม
โดยเฉพาะอย่างยิ่งในตัวควบคุมแบบตั้งโปรแกรมได้ หากมีความผิดพลาดเกิดขึ้น จะทำให้เกิดการหยุดทำงานของระบบ อุปกรณ์เสียหาย หรืออุบัติเหตุอื่นๆ

ตารางต่อไปนี้จะแสดงฟังก์ชั่นบางประการของ GX Works2 ที่สามารถช่วยในการแก้ไขปัญหาได้

ชื่อฟังก์ชั่น	รายละเอียด
ตัวจำลอง	ฟังก์ชั่นนี้จะใช้เพื่อจำลองการดำเนินการโปรแกรม แม้จะไม่มีโมดูล CPU ก็ตาม ฟังก์ชั่นนี้จะสามารถใช้เพื่อแก้ไขความผิดพลาดในสิ่งแวดล้อมที่ไม่มีโมดูล CPU
การตรวจสอบ	ฟังก์ชั่นนี้ช่วยให้สามารถตรวจสอบสถานะการทำงานและสถานะของอุปกรณ์ต่างๆ ในระหว่างการทำงานของโมดูล CPU ฟังก์ชั่นการตรวจสอบหลายรายการที่มีจะขึ้นอยู่กับการใช้งาน เช่น การตรวจสอบ Ladder การตรวจสอบแต่อุปกรณ์ที่ลงทะเบียน และการตรวจสอบอุปกรณ์ทั้งหมดในชุด
เปลี่ยนค่าปัจจุบัน	ฟังก์ชั่นนี้จะสามารถบังคับการเปลี่ยนแปลงสถานะ (ON ↔ OFF) ของอุปกรณ์ I/O ที่ลงทะเบียนในระหว่างการทำงานของโมดูล CPU ได้ ฟังก์ชั่นนี้มีประโยชน์สำหรับการเปลี่ยนแปลงค่าปัจจุบันของอุปกรณ์คำหรือสถานะของรีเลย์ภายใน
บังคับการลงทะเบียน/การยกเลิกอินพุทเอาท์พุท	ฟังก์ชั่นนี้จะสามารถบังคับการเปลี่ยนแปลงสถานะ (ON ↔ OFF) ของอุปกรณ์ I/O ที่ลงทะเบียนในระหว่างการทำงานของโมดูล CPU ได้ สำหรับการแก้ไขความผิดพลาดหรือการตรวจสอบการดำเนินการด้วยโมดูล CPU เพียงอย่างเดียว จะสามารถใช้ฟังก์ชั่นนี้แทนสวิตช์ได้

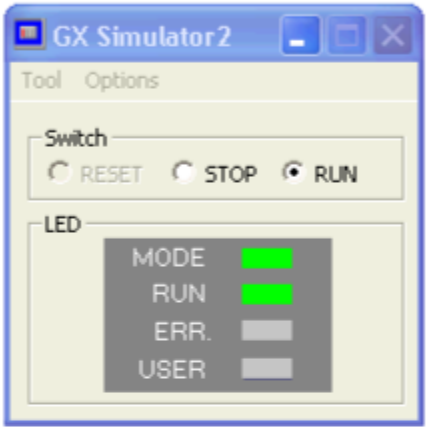
ในส่วนที่เหลือของบทนี้จะอธิบายเกี่ยวกับฟังก์ชั่นเหล่านี้ในรายละเอียดเกี่ยวกับกระบวนการแก้ไขความผิดพลาด

หมายเหตุเกี่ยวกับการแก้ไขความผิดพลาด

อย่าทำการแก้ไขความผิดพลาดในระหว่างที่เชื่อมต่อตัวควบคุมแบบตั้งโปรแกรมได้เข้ากับอุปกรณ์ I/O
ความผิดพลาดในโปรแกรม อุปกรณ์ I/O ที่ถูกบังคับ หรือการเปลี่ยนแปลงค่าคำ อาจส่งผลให้เกิดความเสียหายกับอุปกรณ์ภายนอก หรือร้ายแรงกว่านั้น
หากไม่สามารถเลิกเชื่อมต่อกับระบบ PLC ได้ โปรดใช้ฟังก์ชั่นการจำลอง

4.1.1 การแก้ไขความผิดพลาดของโปรแกรมโดยไม่ใช้โมดูล CPU

หากไม่สามารถใช้โมดูล CPU สำหรับการแก้ไขความผิดพลาดได้ โปรดใช้ ฟังก์ชั่นการจำลอง
โปรแกรมจะสามารถรันบนโมดูล CPU เสมือนจริงโดยซอฟต์แวร์โดยไม่ต้องใช้โมดูล CPU จริงได้



เปิด

ปิด

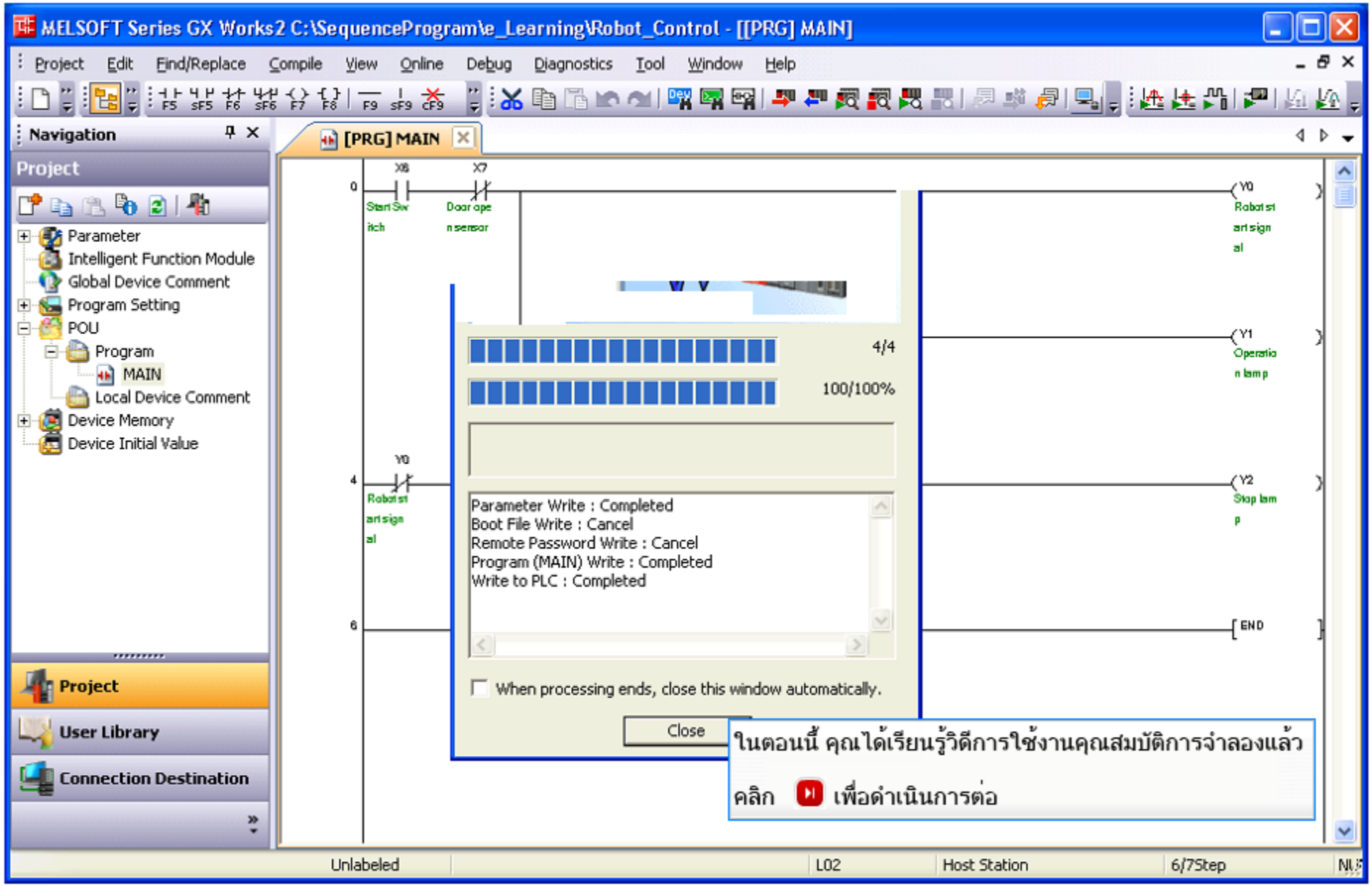
รายการ	สถานะ	รายละเอียด
Switch (สวิตช์)	RUN	รันโมดูล CPU เสมือนจริง
	STOP	หยุดโมดูล CPU เสมือนจริง
	RESET	รีเซ็ตโมดูล CPU เสมือนจริง (ใช้งานได้ในสถานะ STOP เท่านั้น)
LED	MODE	แสดงสถานะโหมดของ CPU เสมือนจริง
	RUN	แสดงสถานะรันของ CPU เสมือนจริง •เปิด: สถานะ RUN •ปิด: สถานะ STOP
	ERR	แสดงสถานะความผิดพลาดของ CPU เสมือนจริง หากมีความผิดพลาดเกิดขึ้น ไฟ LED จะติดหรือกระพริบ
	USER	แสงว่ามีความผิดพลาดของผู้ใช้เกิดขึ้นใน CPU เสมือนจริงหรือไม่ ติดหรือกระพริบเมื่อความผิดพลาดเกิดขึ้น

หมายเหตุการใช้งานฟังก์ชั่นตัวจำลอง

- การแก้ไขความผิดพลาดโดยใช้ฟังก์ชั่นตัวจำลองจะไม่สามารถรับประกันได้ว่าโปรแกรมลำดับจะทำงานได้อย่างถูกต้องหลังจากการแก้ไขความผิดพลาด
- ฟังก์ชั่นตัวจำลองจะดำเนินการอินพุท/เอาท์พุทข้อมูลกับโมดูล I/O โดยใช้หน่วยความจำการจำลองข้อมูล
  ฟังก์ชั่นนี้จะไม่สนับสนุนบางคำสั่ง บางฟังก์ชั่น และบางหน่วยความจำอุปกรณ์ ดังนั้น ผลการทำงานของฟังก์ชั่นตัวจำลองอาจจะแตกต่างจากผลของโมดูล CPU จริง

ในหน้าต่อไป ให้ลองใช้ฟังก์ชั่นตัวจำลองโดยใช้หน้าต่างจำลอง

4.1.1 การแก้ไขความผิดพลาดของโปรแกรมโดยไม่ใช้โมดูล CPU

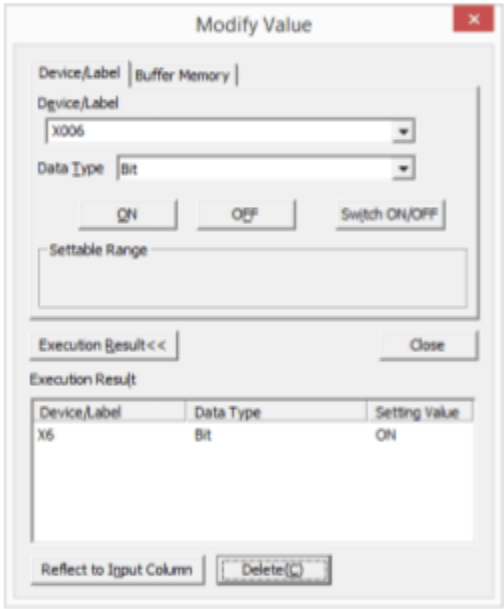


4.1.2 การเปลี่ยนสถานะของอุปกรณ์ I/O

เมื่อทำการแก้ไขความผิดพลาดของโปรแกรมลำดับด้วยโมดูล CPU ที่ไม่ได้เชื่อมต่อกับอุปกรณ์ I/O หรือใช้ฟังก์ชันการจำลอง ให้ใช้ฟังก์ชัน บังคับการลงทะเบียน/การยกเลิกอินพุทเอาท์พุท เพื่อเปลี่ยนสถานะ ON/OFF ของอุปกรณ์ I/O
สามารถบังคับการเปลี่ยนแปลงสถานะของอุปกรณ์ I/O ที่ลงทะเบียนให้เป็น ON หรือ OFF ได้โดยใช้ซอฟต์แวร์

(MELSEC-Q และ MELSEC-L ซีรีส์): จากหน้าจอ “Forced Input Output Registration/Cancellation”
(บังคับการลงทะเบียน/การยกเลิกอินพุทเอาท์พุท)
(MELSEC-F ซีรีส์): จากหน้าจอ "Modify Value" (ปรับค่า)

Forced Input Output Registration/Cancellation
(บังคับการลงทะเบียน/การยกเลิกอินพุทเอาท์พุท)
หน้าจอ (MELSEC-Q and MELSEC-L ซีรีส์)



Modify Value (ปรับค่า) หน้าจอ (MELSEC-F ซีรีส์)

วิธีการเปลี่ยนสถานะของอุปกรณ์อื่นๆ
หากต้องการเปลี่ยนอุปกรณ์ปัจจุบันของอุปกรณ์คำหรือสถานะ ON/OFF ของรีเลย์ภายนอก โปรดใช้ ฟังก์ชันการเปลี่ยนแปลงค่าปัจจุบัน
โปรดดูรายละเอียดได้ที่คู่มือการใช้งาน

4.1.3 การตรวจสอบสถานะของอุปกรณ์

เมื่อเริ่มการจำลอง ระบบจะเริ่มการตรวจสอบโดยอัตโนมัติ หากต้องการเข้าสู่โหมดการตรวจสอบเมื่อเชื่อมต่อกับ CPU ตัวควบคุมแบบตั้งโปรแกรมได้จริง เพียงคลิก ออนไลน์ ตรวจสอบ และเริ่มการตรวจสอบ หรือใช้ทางปุ่มลัดที่แป้นพิมพ์ F3

ในระหว่างโหมดการตรวจสอบ จะสามารถดูค่าและสถานะของอุปกรณ์ทั้งหมดที่ใช้ในโปรแกรมได้ด้วยรหัสโปรแกรม ซึ่งจะช่วยให้ผู้ใช้สามารถดูการเปลี่ยนแปลงของค่าได้โดย รวมถึงผลกระทบของการใช้ฟังก์ชั่น "บังคับการลงทะเบียน/การยกเลิกอินพุทเอาท์พุท" ด้วย

นอกจากนี้ ระบบจะแสดงแถบสถานะการตรวจสอบและข้อมูลพื้นฐานต่างๆ เพื่อประเมินสถานะของ CPU หรือ CPU เสมือนจริงด้วย โปรดดูที่ตารางด้านล่างนี้เพื่อทำความเข้าใจข้อมูลที่อยู่ในแถบ สถานะการตรวจสอบ

เมื่อเชื่อมต่อกับโมดูล CPU

เมื่อใช้ฟังก์ชั่นตัวจำลอง

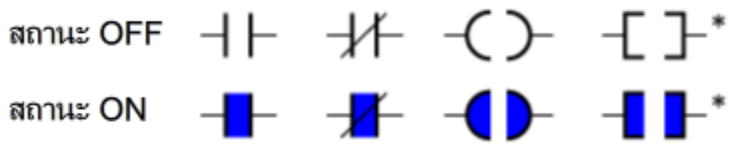
สถานะ	ไอคอน/การแสดง	รายละเอียด
สถานะการเชื่อมต่อ	เมื่อเชื่อมต่อกับโมดูล CPU มื่อใช้ฟังก์ชั่นตัวจำลอง	แสดงสถานะของการเชื่อมต่อกับโมดูล CPU หรือฟังก์ชั่นตัวจำลอง
สถานะ RUN/STOP	RUN STOP	แสดงสถานะรันของ CPU (RUN หรือ STOP)
สถานะ ERR.	ERR. ปิด ERR. on ↔ ERR กระพริบ	แสดงสถานะความผิดพลาดของโมดูล CPU
สถานะ USER	USER ปิด USER เปิด ↔ USER กระพริบ	แสดงสถานะความผิดพลาดผู้ใช้ของโมดูล CPU
เวลาการสแกน	0.000ms	แสดงเวลาการสแกนสูงสุดของโมดูล CPU ที่กำลังตรวจสอบ
สถานะการมี/การไม่มีคำสั่งที่ไม่สนับสนุน	มีคำสั่งที่ไม่สนับสนุน ไม่มีคำสั่งที่ไม่สนับสนุน	แสดงว่ามีคำสั่งที่ไม่สนับสนุนหรือไม่เมื่อดำเนินการฟังก์ชั่นตัวจำลอง การคลิกที่ไอคอนจะเป็นการเปิดหน้าต่างอุปกรณ์/คำสั่งที่ไม่สนับสนุน

4.1.3 การตรวจสอบสถานะของอุปกรณ์

ในระหว่างโหมดการตรวจสอบ ระบบจะแสดงสถานะปัจจุบันของอุปกรณ์ทั้งหมดในโปรแกรม

การแสดงสถานะอุปกรณ์บิต (ON/OFF)

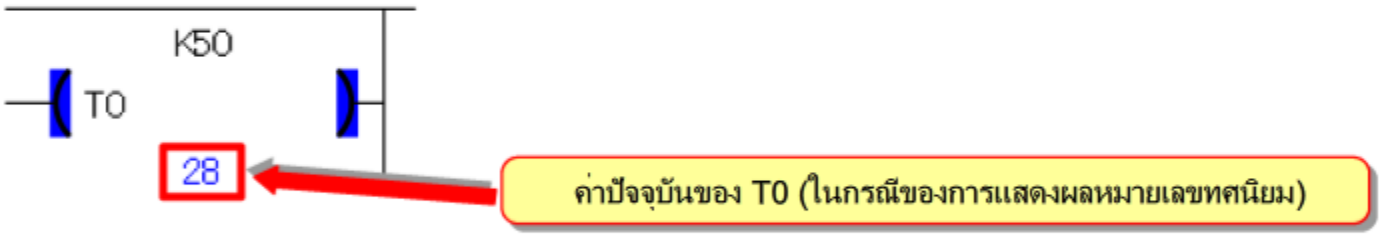
ระบบจะแสดงสถานะ ON/OFF ในระหว่างการตรวจสอบตามที่แสดงไว้ด้านล่างนี้



* การแสดงผลชนิดนี้จะใช้กับ SET, RST, PLS, PLF, SFT, SFTP, MC และคำสั่งการเปรียบเทียบชนิดคอนแท็กเท่านั้น โปรดทราบว่าสำหรับคำสั่ง RST ระบบจะแสดงสถานะ ON/OFF เท่านั้น

การแสดงค่าปัจจุบันของอุปกรณ์คำ (การแสดงหมายเลขทศนิยม/ฐานสิบหก)

ระบบจะแสดงค่าปัจจุบันในการตรวจสอบดังที่แสดงไว้ด้านล่าง



การตรวจสอบเฉพาะอุปกรณ์ที่กำหนด

เมื่อตรวจสอบระบบที่ใหญ่มากๆ หรือมีความซับซ้อน การตรวจสอบแต่อุปกรณ์ที่ต้องการอาจเป็นวิธีการที่เหมาะสม เพื่อการดำเนินการดังกล่าว GX Works2 จะมีหน้าต่างการเฝ้าดูเพื่อให้ผู้ใช้สามารถเพิ่มอุปกรณ์ที่ตนสนใจเพื่อดูสถานะปัจจุบัน และแก้ไขค่าในระหว่างการตรวจสอบ สำหรับรายละเอียด โปรดดูที่คู่มือการใช้งาน GX Works2 คู่มือการใช้งาน (ทั่วไป)

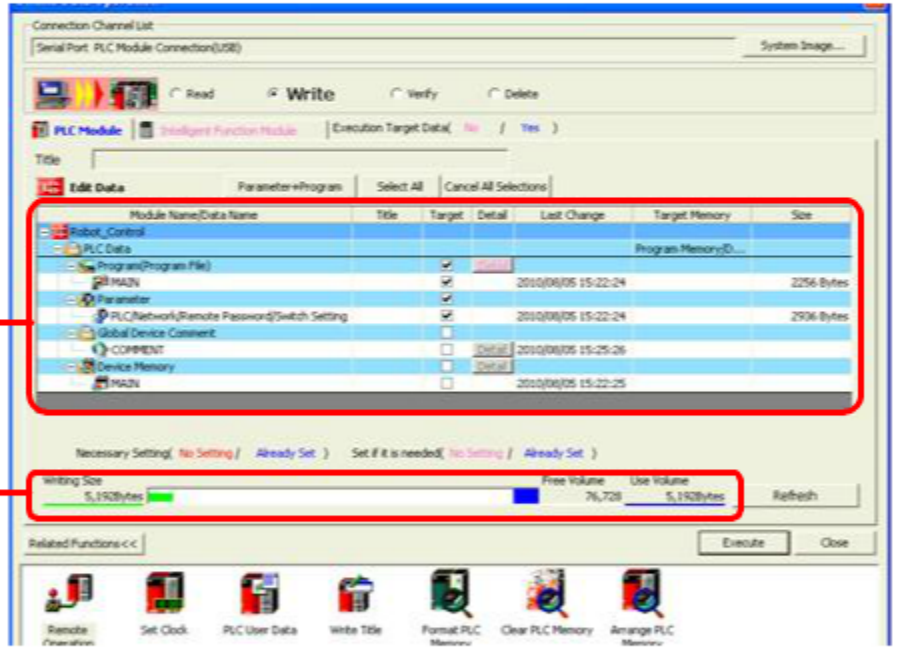
Watch 1

Device/Label	Current Value	Data Type	Class	Device	Comment
X7	-	Bit		X7	Door open sensor
Y0	-	Bit		Y0	Robot start signal
Y1	-	Bit		Y1	Operation lamp
Y0	-	Bit		Y0	Robot start signal
Y2	-	Bit		Y2	Stop lamp
Y0	-	Bit		Y0	Robot start signal

4.2 การเขียนโปรแกรมไปยังโมดูล CPU

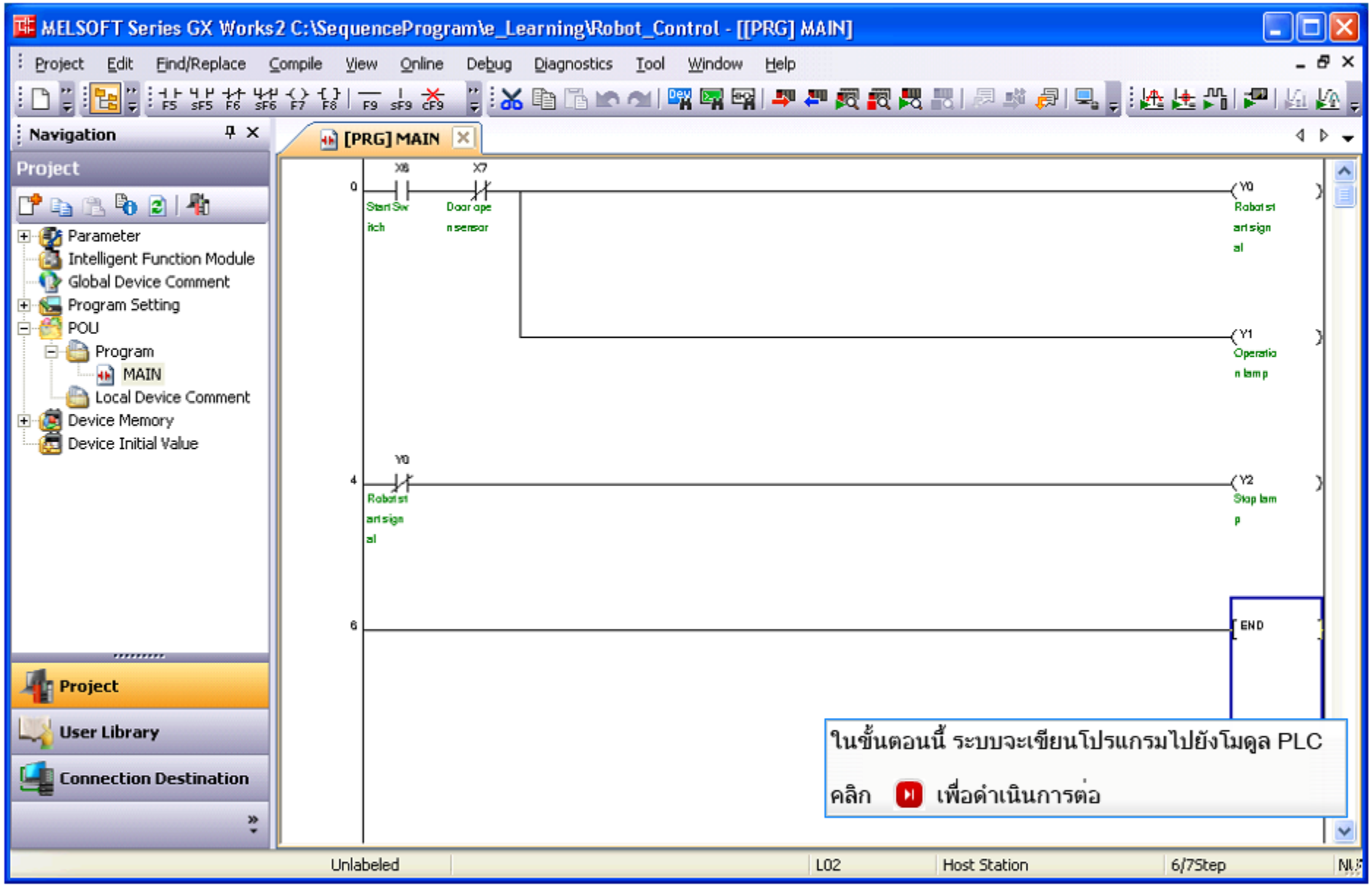
ก่อนทำการแก้ไขความผิดพลาดโมดูล CPU จริง โปรดสั่งให้ CPU อยู่ใน โหมด STOP ตรวจสอบว่ามีการเชื่อมต่อกับ CPU แล้วเขียนโปรแกรมและพารามิเตอร์สู่หน่วยความจำโปรแกรม

ดังที่แสดงในภาพหน้าจอด้านล่าง ฟังก์ชั่นหลักของหน้าต่าง Write to PLC จะช่วยให้ผู้ใช้สามารถเลือกไฟล์ที่ต้องการเขียน เลือกตำแหน่ง และตรวจสอบหน่วยความจำของ CPU ได้ ปุ่มสามปุ่มด้านบนรายการไฟล์จะช่วยให้ผู้ใช้สามารถเลือกไฟล์ที่ต้องการเขียนได้อย่างรวดเร็ว โดยส่วนใหญ่ ปุ่มที่ใช้กับการจำลองคือ “Parameter+Program”



ในหน้าต่อไป ให้ลองเขียนไปยังโมดูล CPU โดยใช้หน้าต่างจำลอง

4.2 การเขียนโปรแกรมไปยังโมดูล CPU



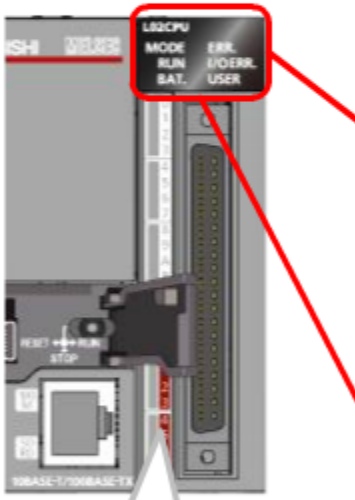
4.3 การเปิดใช้งานโปรแกรมที่เขียน

(MELSEC-F ซีรีส์): ไม่จำเป็นต้องมีการดำเนินการต่อไปนี้

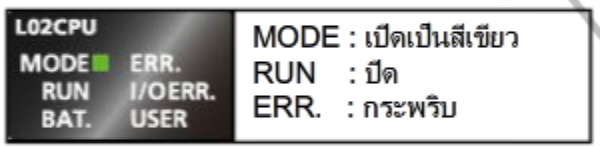
(MELSEC-Q และ MELSEC-L ซีรีส์): จำเป็นต้องมีการดำเนินการต่อไปนี้
หลังจากเขียนโปรแกรมไปยังโมดูล CPU แล้ว ให้ รีเซ็ต โมดูล CPU
ระบบจะไม่สามารถเปิดใช้งานโปรแกรมที่เขียนขึ้นได้หากไม่ได้รีเซ็ตโมดูล CPU

* การดำเนินการนี้จะไม่จำเป็นหากใช้ฟังก์ชันตัวจำลองในการแก้ไขปัญหา

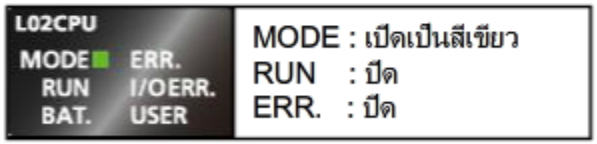
รีเซ็ตโมดูล CPU โดยใช้วิธีการดังต่อไปนี้:



(1) กดสวิตช์ RESET/STOP/RUN ที่แผงด้านหน้าของโมดูล CPU ค้างจนอยู่ในตำแหน่ง RESET (ประมาณ 1 วินาทีขึ้นไป)
[กำลังดำเนินการรีเซ็ต]



(2) ปล่อยสวิตช์หลังจากที่ดวงไฟ LED MODE ติดขึ้นและ ERR. กระพริบ ไฟ LED ทั้งคู่ปิด
[การรีเซ็ตเสร็จสิ้น]



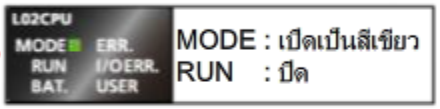
(3) สวิตช์จะเปลี่ยนกลับเป็นตำแหน่ง STOP เมื่อรีเซ็ตเสร็จ

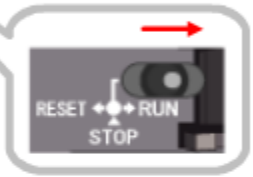
4.4 การรันโปรแกรม

MELSEC-Q และ MELSEC-L ซีรีส์
หลังจากรีเซ็ตเสร็จ ให้รันโปรแกรม
สั่งให้โมดูล CPU เข้าสู่ สถานะ RUN ด้วยวิธีการดังต่อไปนี้เพื่อรันโปรแกรม

* การดำเนินการนี้จะไม่จำเป็นหากใช้ฟังก์ชันตัวจำลองในการแก้ไขปัญหา

(1) ปรับสวิตช์ RESET/STOP/RUN ทีแผงด้านหน้าของโมดูล CPU ไปยังตำแหน่ง RUN
ไฟ LED แสดงในสถานะ STOP





(2) หากดวงไฟ LED RUN ติดเป็นสีเขียว แสดงว่าโปรแกรมรันตามปกติ
ไฟ LED แสดงในสถานะ RUN

MELSEC-F ซีรีส์
หลังจากเขียนโปรแกรมลงในหน่วยหลัก ให้สวิทช์หน่วยหลักไปยังสถานะ RUN ดังต่อไปนี้เพื่อรันโปรแกรม (ไม่จำเป็นต้องรีเซ็ตการทำงาน)

(1) หมุนสวิทช์ RUN/STOP บนแผงหน้าปัดด้านหน้าของหน่วยหลักไปยังตำแหน่ง RUN

ไฟ LED แสดงในสถานะ STOP

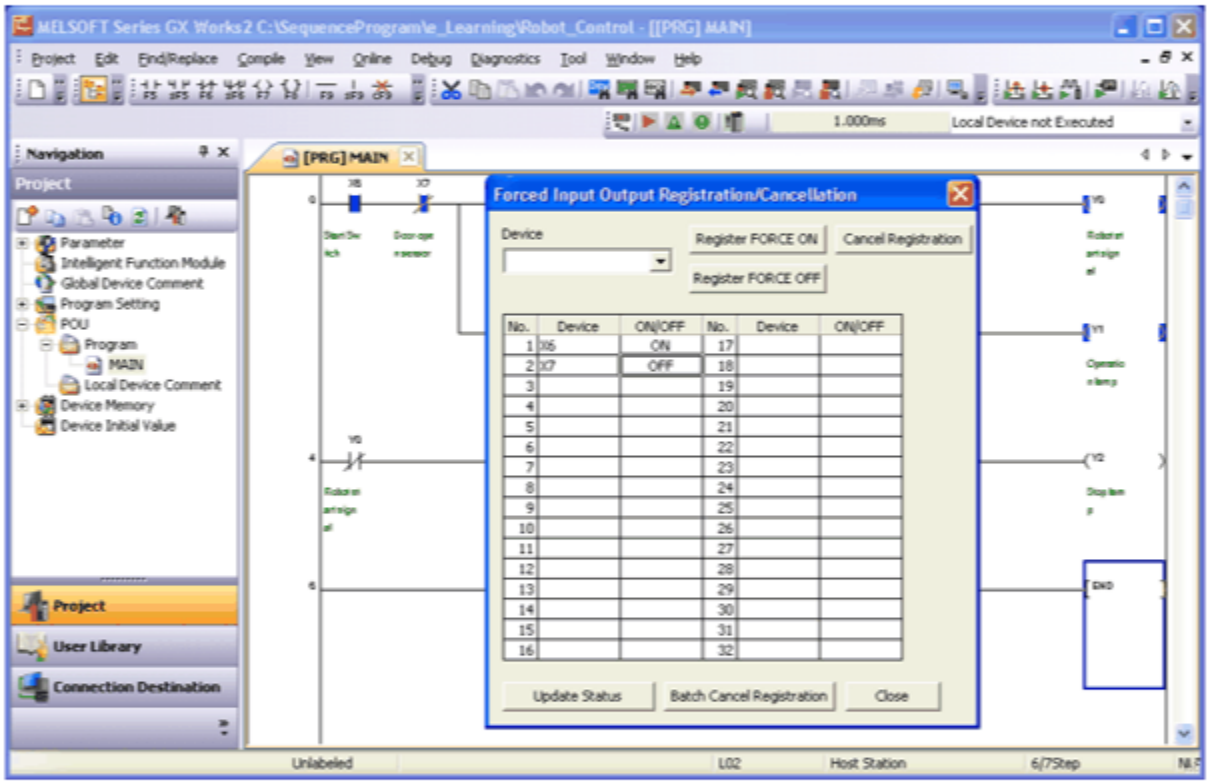
(2) หากไฟ RUN LED ติดสว่าง แสดงว่าโปรแกรมรันเป็นปกติ

ไฟ LED แสดงในสถานะ RUN

4.5 การแก้ไขความผิดพลาดโปรแกรม

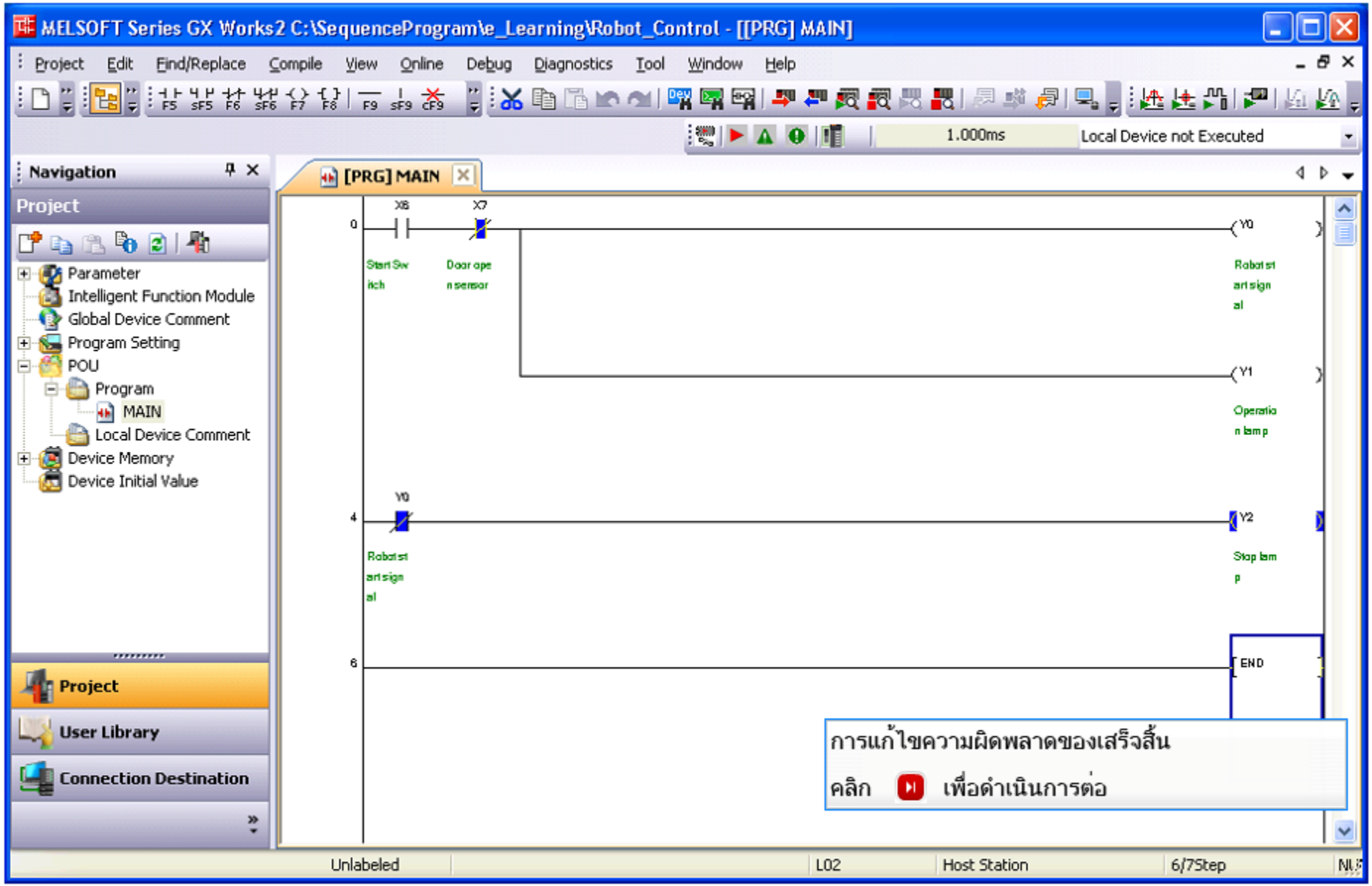
หลังจากที่รันโมดูล CPU แล้ว ใช้ฟังก์ชันบังคับการลงทะเบียน/การยกเลิกอินพุทเอาท์พุทเพื่อเปลี่ยนสถานะของอุปกรณ์แต่ละรายการและตรวจสอบผลลัพธ์ (เอาท์พุท) ของ Ladder

(ตัวอย่างหน้าจอของ MELSEC-Q and MELSEC-L ซีรีส์)



ในหน้าต่อไป ให้ลองแก้ไขความผิดพลาดโดยใช้หน้าต่างจำลอง

4.5 การแก้ไขความผิดพลาดโปรแกรม

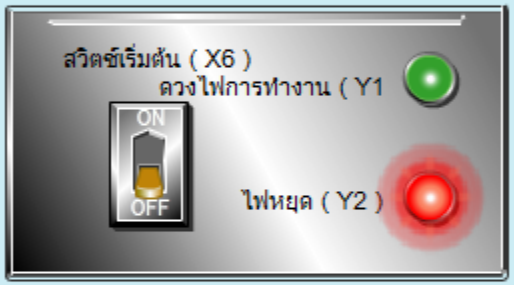


4.6 การตรวจสอบการทำงานระบบ PLC

หลังจากแก้ไขความผิดพลาดของโปรแกรมแล้ว ให้เขียนโปรแกรมไปยังระบบ PLC จริงเพื่อตรวจสอบการทำงานขั้นสุดท้าย
ดำเนินการอุปกรณ์ I/O จริงเพื่อตรวจสอบว่าสามารถทำงานได้ตามที่ต้องการ
แม้เมื่อใช้งานอุปกรณ์ I/O ผู้ใช้จะสามารถตรวจสอบสถานะอุปกรณ์แต่ละตัวได้โดยใช้ฟังก์ชั่นการตรวจสอบของ GX Works2

การทำงานของระบบตัวอย่าง คลิกในวงกลมสีแดง

แผงควบคุมโรบอท

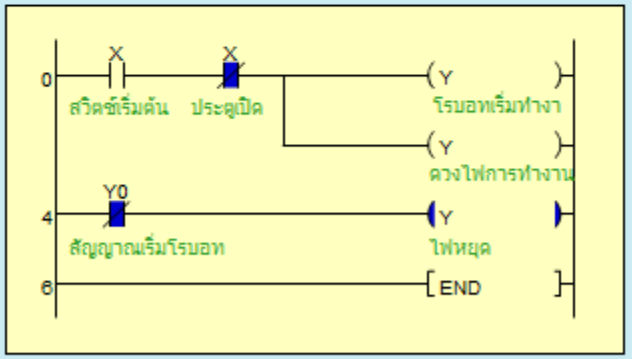


โรบอทในรั้วนิรภัย

เมื่อตั้งค่า สวิตช์เริ่มต้น (X6) เป็น OFF โรบอทเริ่มทำงาน (Y0) จะเปิดและโรบอทจะเริ่มทำงาน ในเวลาเดียวกัน ดวงไฟการทำงาน (Y1) ที่แผงควบคุมจะดับและ ไฟหยุด (Y2) จะติด

เล่นซ้ำ

ก่อนหน้า



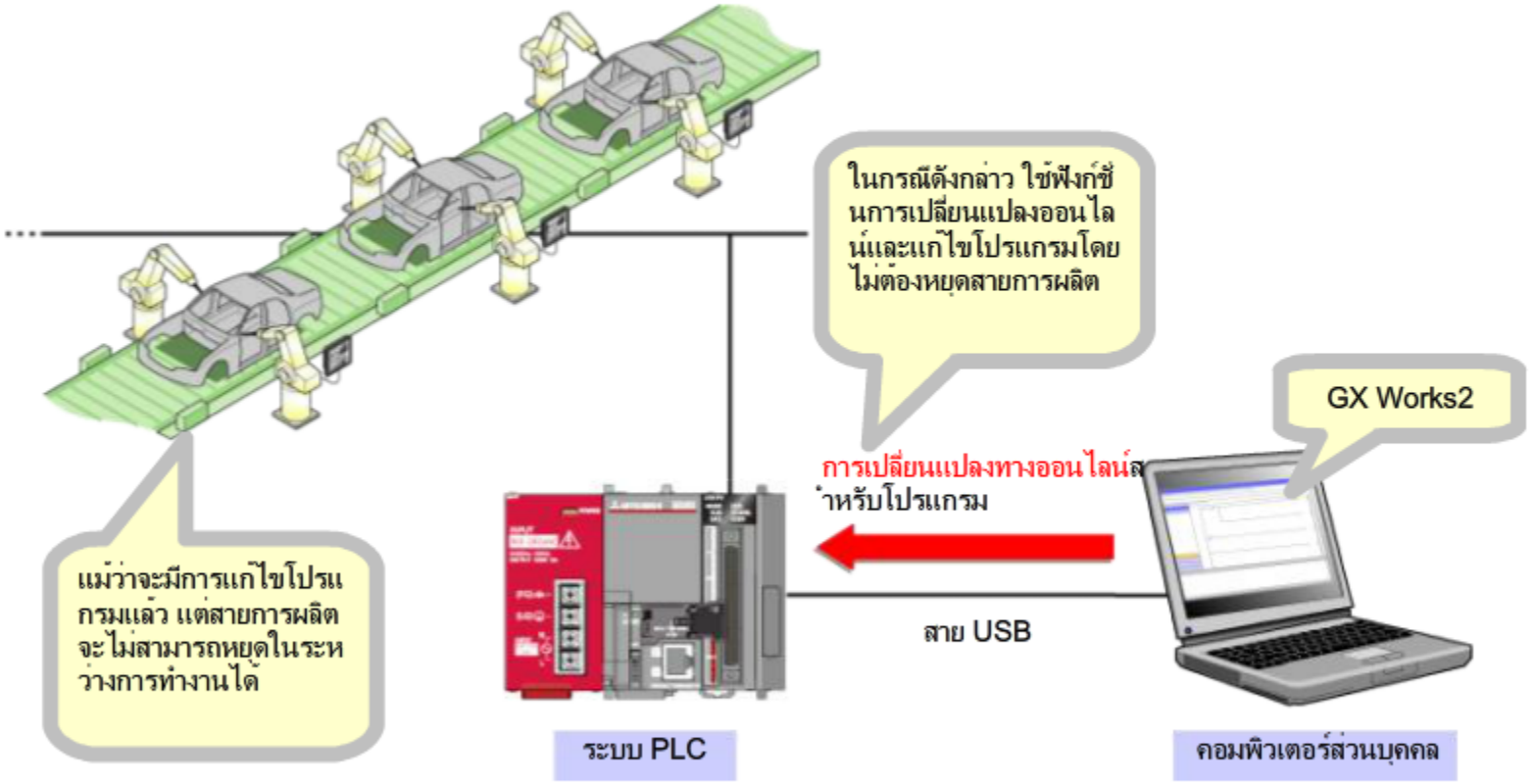
4.7 การใช้งานระบบ PLC

หลังจากการตรวจสอบการทำงาน รันระบบ PLC เพื่อเริ่มใช้งาน

หากต้องการแก้ไขโปรแกรมในระบบที่กำลังทำงาน

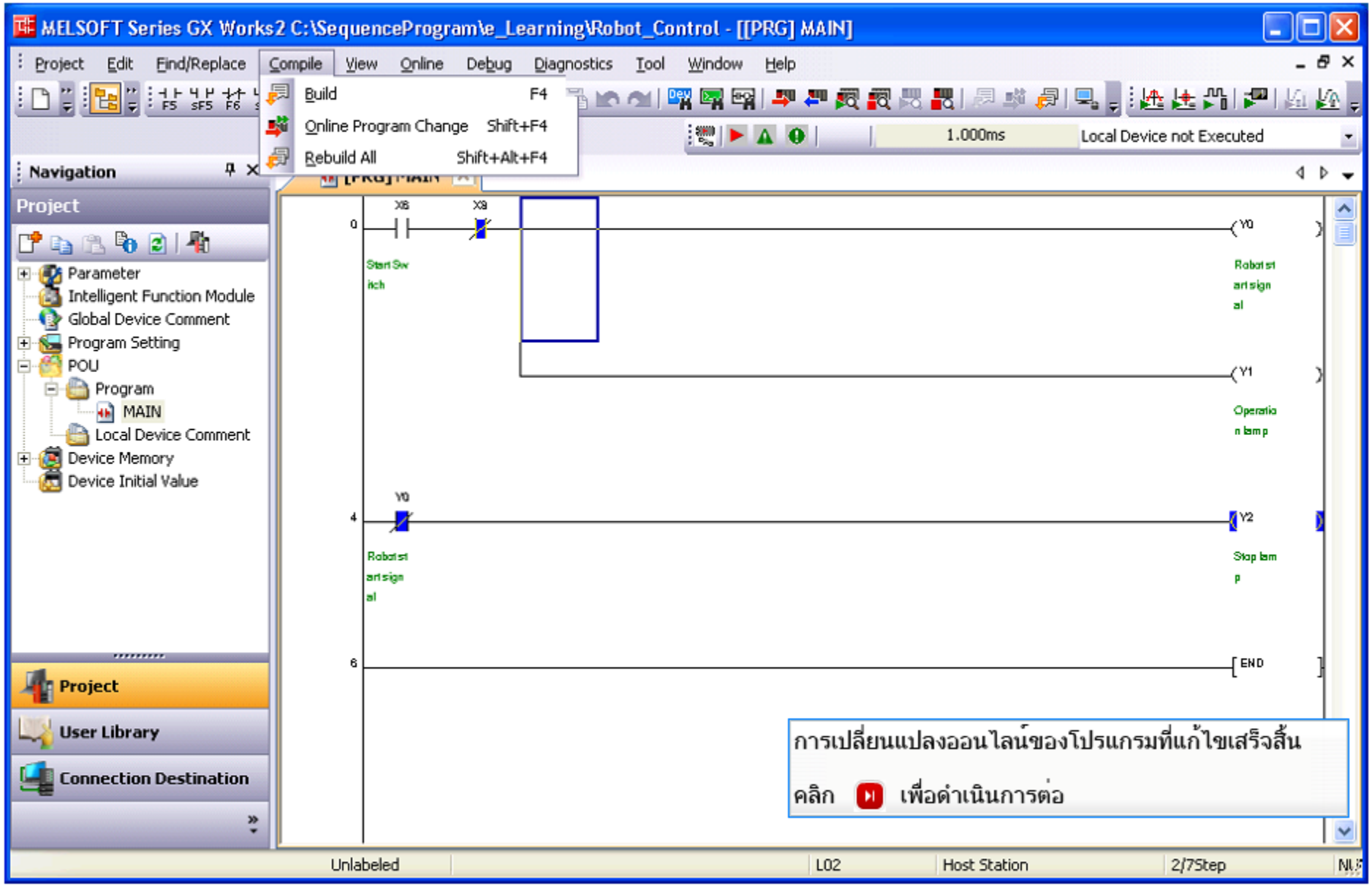
อาจต้องมีการแก้ไขโปรแกรม เช่น การแก้ไขข้อผิดพลาดหรือการขยายระบบหลังจากที่เริ่มใช้งานระบบแล้ว โดยปกติแล้ว จะต้องหยุดระบบ (โมดูล CPU) เพื่อเขียนโปรแกรมแก้ไข แต่ก็ไม่แน่เสมอไป วิธีการแก้ไขปัญหาดังกล่าว GX Works ได้จัดเตรียมฟังก์ชั่นการเปลี่ยนแปลงออนไลน์ ซึ่งจะใช้เพื่อเขียนโปรแกรมโดยไม่ต้องหยุดโมดูล CPU ที่กำลังทำงาน

ตัวอย่าง: สายการผลิตยานยนต์ที่ต้องทำงานตลอด 24 ชั่วโมง



ในหน้าต่อไป ให้ลองใช้ฟังก์ชั่นการเปลี่ยนแปลงออนไลน์โดยใช้หน้าต่างจำลอง

4.7 การใช้งานระบบ PLC



4.8 สรุป

เราได้เรียนรู้เนื้อหาในหลักสูตรพื้นฐานเกี่ยวกับการออกแบบซอฟต์แวร์การควบคุมแบบตั้งโปรแกรมได้ทั้งหมดแล้ว ในหลักสูตรนี้ คุณได้เรียนรู้เกี่ยวกับ:

- ปัจจัยต่างๆ ที่จำเป็นสำหรับการตั้งโปรแกรมระบบ PLC
- แนวทางพื้นฐานสำหรับการออกแบบโปรแกรม รวมถึงการใช้ความคิดเห็น
- วิธีการใช้ GX works2 เพื่อดำเนินงานพื้นฐานของการตั้งโปรแกรม PC
- เทคนิคบางประการที่ใช้สำหรับการแก้ไขความผิดพลาดของโปรแกรม PLC

ทดสอบ แบบทดสอบประเมินผล

ในตอนนี้คุณได้ผ่านหลักสูตรทั้งหมดของ พื้นฐาน PLC GX Works2 แล้ว คุณพร้อมที่จะทำแบบทดสอบท้ายหลักสูตรแล้ว หากคุณยังไม่มั่นใจเกี่ยวกับหัวข้อต่างๆ ที่จะทดสอบ โปรดทบทวนหัวข้อเหล่านั้น
คำถามในแบบทดสอบประเมินผลนี้มีทั้งหมด 5 ข้อ (15 รายการ)
คุณสามารถทำแบบทดสอบประเมินผลได้หลายครั้งตามต้องการ

วิธีการตอบคำถามในแบบทดสอบ

หลังจากเลือกคำตอบแล้ว ให้คลิกปุ่ม Answer คำตอบของคุณจะหายไป ถ้าคุณดำเนินการต่อโดยไม่คลิกปุ่ม Answer (โดยจะถือว่าคุณยังไม่ได้ตอบคำถามนั้น)

ผลคะแนน

จำนวนคำตอบที่ถูกต้อง จำนวนคำถาม เปอร์เซ็นต์คำตอบที่ถูกต้อง และผลลัพธ์ที่แสดงว่าผ่าน/ไม่ผ่านจะปรากฏบนหน้าผลคะแนน

คำตอบที่ถูกต้อง: 2

จำนวนคำถามทั้งหมด: 9

เปอร์เซ็นต์: 22%

คุณต้องตอบคำถามถูกต้องเกินกว่า 60% จึงจะผ่านการทดสอบ

ดำเนินการต่อ | ทบทวน | ลองใหม่

- คลิกปุ่ม ดำเนินการต่อ เพื่อออกจากการทดสอบ
- คลิกปุ่ม ทบทวน เพื่อทบทวนการทดสอบ (ตรวจสอบคำตอบที่ถูกต้อง)
- คลิกปุ่ม ลองใหม่ เพื่อทำการทดสอบใหม่อีกครั้ง

แบบทดสอบ

แบบทดสอบประเมินผล 1

โปรแกรมที่คุณดูแลได้ถูกส่งต่อไปยังผู้ดูแลรายอื่น ผู้ซึ่งประสบกับปัญหาในการทำความเข้าใจต่อรายการควบคุมของโปรแกรม มาตรการป้องกันปัญหานี้ที่เหมาะสมที่สุดคืออะไร?

- [ ] การใช้ฟังก์ชั่นความคิดเห็นของ GX Works2 เพื่อระบุหัวข้อและคำอธิบายที่เหมาะสมสำหรับโปรแกรม
- [ ] อธิบายรายการควบคุมให้แก่บุคคลใหม่ด้วยวาจา
- [ ] หลีกเลี่ยงการส่งมอบโปรแกรมขนาดใหญ่ที่มีความซับซ้อน
- [ ] มอบตารางตอบสนองสำหรับอุปกรณ์ I/O และหมายเลขอุปกรณ์ไปพร้อมกับโปรแกรม

คะแนน ย้อนกลับ

แบบทดสอบ

แบบทดสอบประเมินผล 2

จงทำให้กระบวนการตั้งโปรแกรมที่ถูกต้องสมบูรณ์

ขั้นที่ 1 การออกแบบโปรแกรม

ขั้นที่ 2 (Q1 --Select--)

ขั้นที่ 3 (Q2 --Select--)

ขั้นที่ 4 การแปลงโปรแกรม

ขั้นที่ 5 การบันทึกโครงการ

ขั้นที่ 6 (Q3 --Select--)

ขั้นที่ 7 (Q4 --Select--)

ขั้นที่ 8 การรันโมดูล CPU (RUN)

ขั้นที่ 9 (Q5 --Select--)

ขั้นที่ 10 การตรวจสอบการทำงานระบบ PLC

คะแนน ย้อนกลับ

แบบทดสอบ

แบบทดสอบประเมินผล 3

จงเติมคำในช่องว่างเพื่ออธิบายสิ่งที่ต้องทำหลังจากเขียนโปรแกรมสำเร็จ

เมื่อเขียนโปรแกรมแล้ว จะต้องทดสอบเพื่อให้แน่ใจว่าโปรแกรมทำงานได้ตามต้องการ

A (--Select--) (เมื่อรหัสที่เขียนไม่ทำงานตามที่ต้องการ) จะเรียกว่า

a (--Select--) และกระบวนการหาสาเหตุและแก้ไขปัญหาดังกล่าวจะเรียกว่า

(--Select--).

กระบวนการนี้เป็นขั้นตอนที่สำคัญในการสร้างโปรแกรม

คะแนน ย้อนกลับ

แบบทดสอบ

แบบทดสอบประเมินผล 4

เลือกการใช้งานที่เหมาะสมสำหรับฟังก์ชั่นการทำงาน GX Works2 แต่ละรายการ

ฟังก์ชั่น	การใช้งาน
การจำลอง	--Select--
บังคับการลงทะเบียน/การยกเลิกอินพุทเอาท์พุท	--Select--
เปลี่ยนค่าปัจจุบัน	--Select--
การตรวจสอบ Ladder	--Select--
เฝ้าดู	--Select--

คะแนน ย้อนกลับ

แบบทดสอบ

แบบทดสอบประเมินผล 5

เลือกคำอธิบายที่เหมาะสมสำหรับฟังก์ชั่นการเปลี่ยนแปลงออนไลน์

- ฟังก์ชั่นนี้จะหยุด CPU โดยอัตโนมัติ เขียนโปรแกรมสู่ CPU แล้วรัน CPU โดยอัตโนมัติ
- ฟังก์ชั่นนี้จะเปรียบเทียบโปรแกรมในโมดูลการรัน CPU กับโปรแกรมที่เปิดโดย GX Works2
- ฟังก์ชั่นนี้สามารถเขียนโปรแกรมไปยังโมดูล CPU หลังจากที่หยุดโมดูล CPU ที่กำลังทำงานอย่างปลอดภัย
- ฟังชั่นนี้สามารถเขียนโปรแกรมไปยังโมดูล CPU ที่กำลังทำงานโดยไม่ต้องหยุด

คะแนน ย้อนกลับ

แบบทดสอบ

คะแนนการทดสอบ

คุณทำแบบทดสอบประเมินผลเสร็จเรียบร้อยแล้ว ผลคะแนนของคุณเป็นดังต่อไปนี้
หากต้องการจบแบบทดสอบประเมินผล ให้ไปยังหน้าถัดไป

คำตอบที่ถูกต้อง: 0

จำนวนคำถามทั้งหมด: 5

เปอร์เซ็นต์: 0%

ดำเนินการต่อ | ทบทวน | ลองใหม่

คุณไม่ผ่านการทดสอบ

คุณได้ผ่านหลักสูตร พื้นฐาน PLC GX Works2 แล้ว

ขอขอบคุณสำหรับการเรียนรู้หลักสูตรนี้

เราหวังว่าคุณจะเพลิดเพลินกับบทเรียน และข้อมูลที่คุณได้รับจากหลักสูตรนี้จะเ
ป็นประโยชน์ในอนาคต

คุณสามารถทบทวนหลักสูตรได้หลายครั้งตามต้องการ

ทบทวน ปิด