ตัวควบคุมระบบเซอร์โว

พื้นฐานโมดูลโมชั่น MELSEC iQ-R ซีรีส์ (การควบคุมตำแหน่ง RD78G(H))

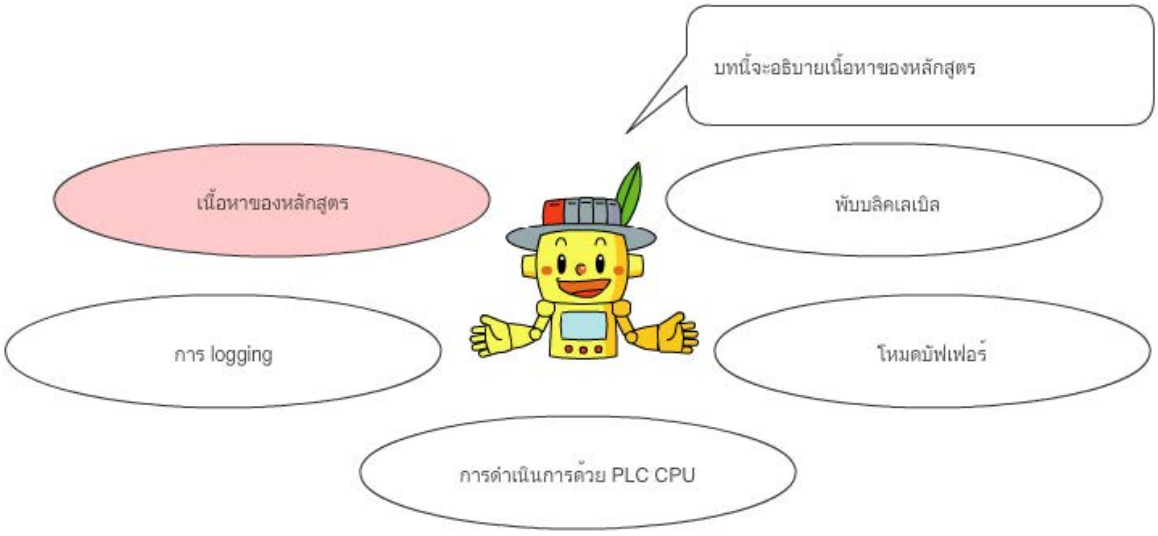
หลักสูตรอบรมนี้เป็นหลักสูตรสำหรับผู้ที่จะสร้างระบบควบคุมการเคลื่อนไหวโดยใช้โมดูลโมชั่น MELSEC iQ-R ซีรีส์ เป็นครั้งแรก

คลิกที่ปุ่มถัดไปที่ด้านขวาบนของหน้าจอเพื่อไปยังหน้าถัดไป

L(CTS)00765THA

บทนำ วัตถุประสงค์ของหลักสูตร

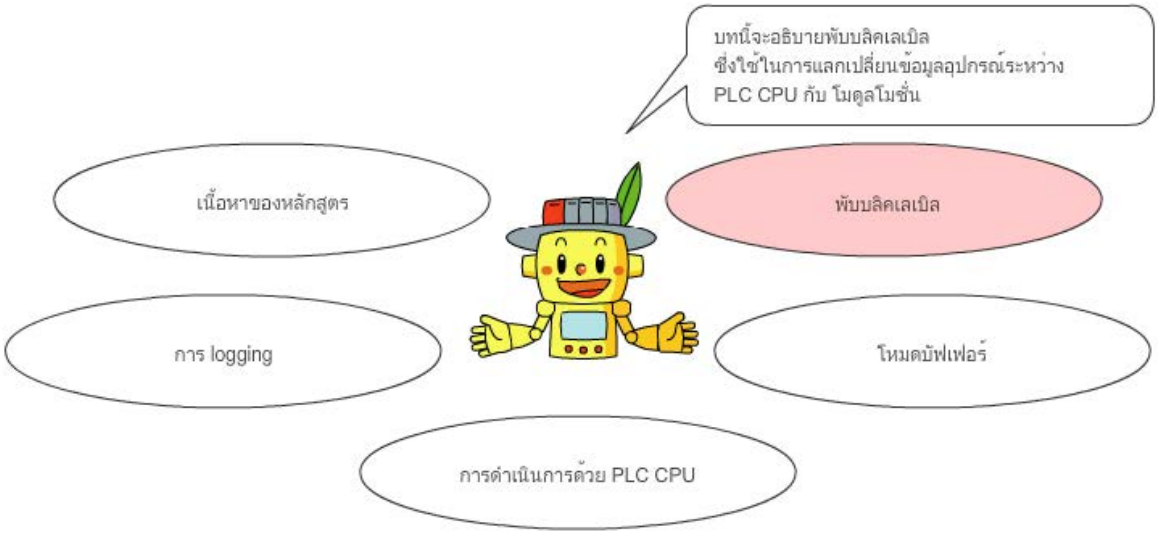
วัตถุประสงค์ของหลักสูตรนี้คือเพื่อพัฒนาความรู้และความเข้าใจเกี่ยวกับการควบคุมตำแหน่งของระบบควบคุมการเคลื่อนไหวโดยใช้โมดูลโมชั่น MELSEC iQ-R ซีรีส์



หลักสูตรนี้เป็นหลักสูตรต่อเนื่องจาก พื้นฐานโมดูลโมชั่น MELSEC iQ-R ซีรีส์ (RD78G(H) เริ่มต้น)
คุณต้องจบหลักสูตรเริ่มต้นก่อนที่จะเรียนหลักสูตรนี้

บทนำ วัตถุประสงค์ของหลักสูตร

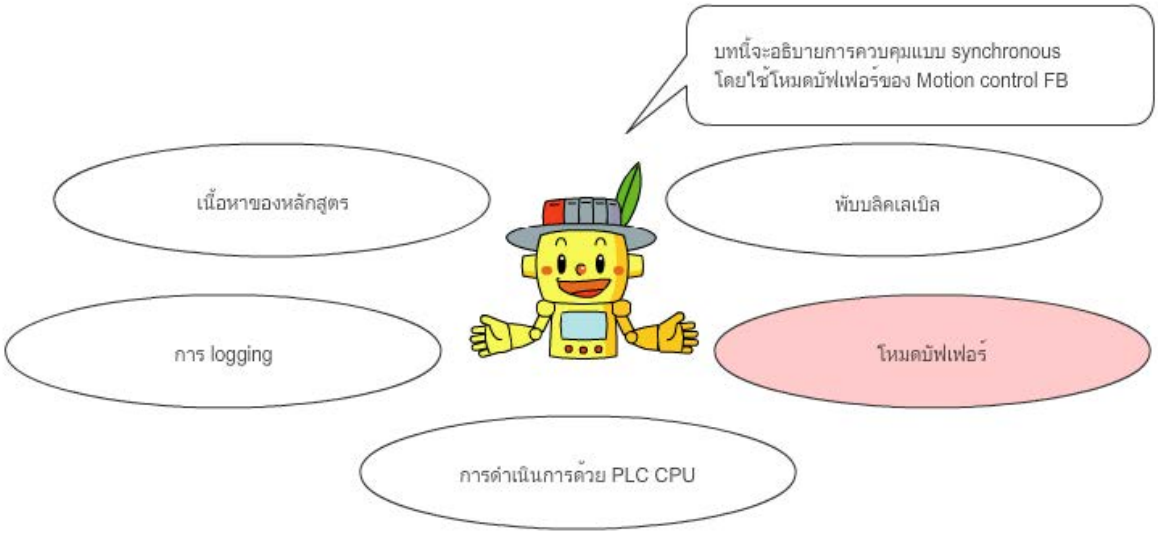
วัตถุประสงค์ของหลักสูตรนี้คือเพื่อพัฒนาความรู้และความเข้าใจเกี่ยวกับการควบคุมตำแหน่งของระบบควบคุมการเคลื่อนไหวโดยใช้โมดูลโมชั่น MELSEC iQ-R ซีรีส์



หลักสูตรนี้เป็นหลักสูตรต่อเนื่องจาก พื้นฐานโมดูลโมชั่น MELSEC iQ-R ซีรีส์ (RD78G(H) เริ่มต้น)
คุณต้องจบหลักสูตรเริ่มต้นก่อนที่จะเรียนหลักสูตรนี้

บทนำ วัตถุประสงค์ของหลักสูตร

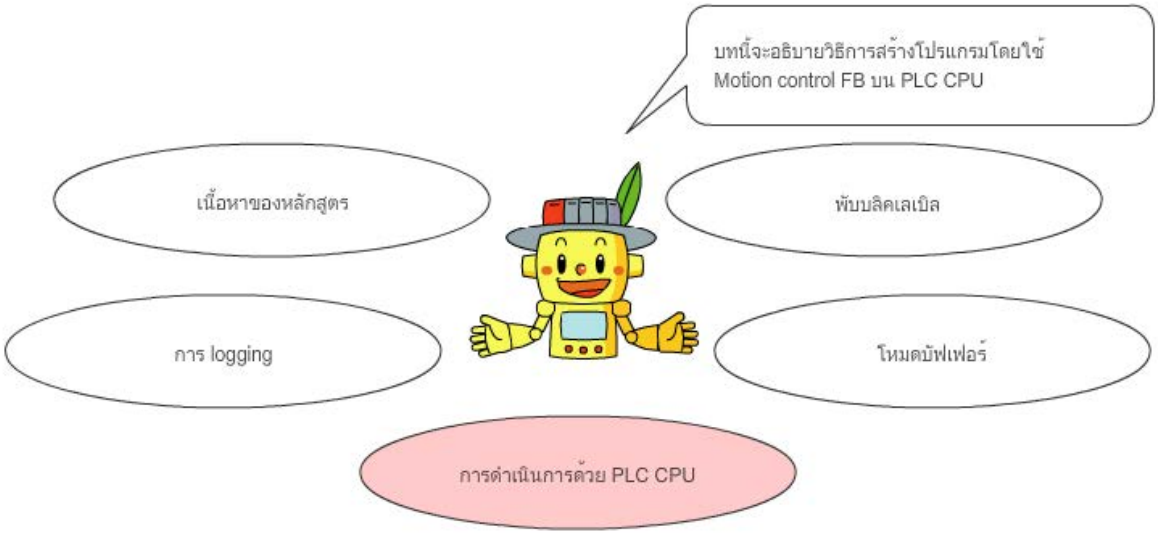
วัตถุประสงค์ของหลักสูตรนี้คือเพื่อพัฒนาความรู้และความเข้าใจเกี่ยวกับการควบคุมตำแหน่งของระบบควบคุมการเคลื่อนไหวโดยใช้โมดูลโมชั่น MELSEC iQ-R ซีรีส์



หลักสูตรนี้เป็นหลักสูตรต่อเนื่องจาก พื้นฐานโมดูลโมชั่น MELSEC iQ-R ซีรีส์ (RD78G(H) เริ่มต้น)
คุณต้องจบหลักสูตรเริ่มต้นก่อนที่จะเรียนหลักสูตรนี้

บทนำ วัตถุประสงค์ของหลักสูตร

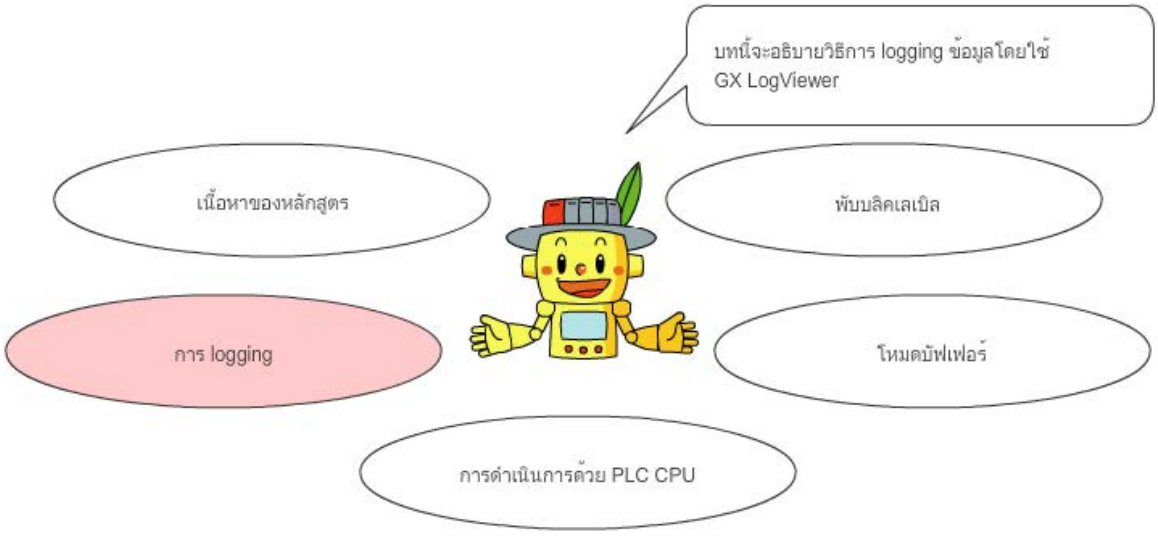
วัตถุประสงค์ของหลักสูตรนี้คือเพื่อพัฒนาความรู้และความเข้าใจเกี่ยวกับการควบคุมตำแหน่งของระบบควบคุมการเคลื่อนไหวโดยใช้โมดูลโมชั่น MELSEC iQ-R ซีรีส์



หลักสูตรนี้เป็นหลักสูตรต่อเนื่องจาก พื้นฐานโมดูลโมชั่น MELSEC iQ-R ซีรีส์ (RD78G(H) เริ่มต้น)
คุณต้องจบหลักสูตรเริ่มต้นก่อนที่จะเรียนหลักสูตรนี้

บทนำ วัตถุประสงค์ของหลักสูตร

วัตถุประสงค์ของหลักสูตรนี้คือเพื่อพัฒนาความรู้และความเข้าใจเกี่ยวกับการควบคุมตำแหน่งของระบบควบคุมการเคลื่อนไหวโดยใช้โมดูลโมชั่น MELSEC iQ-R ซีรีส์



หลักสูตรนี้เป็นหลักสูตรต่อเนื่องจาก พื้นฐานโมดูลโมชั่น MELSEC iQ-R ซีรีส์ (RD78G(H) เริ่มต้น)
คุณต้องจบหลักสูตรเริ่มต้นก่อนที่จะเรียนหลักสูตรนี้

บทนำ โครงสร้างหลักสูตร

เนื้อหาของหลักสูตรนี้มีดังนี้
เราขอแนะนำให้คุณเริ่มต้นจากบทที่ 1

บทที่ 1 - เนื้อหาของหลักสูตร

บทนี้จะอธิบายเนื้อหาของหลักสูตร

บทที่ 2 - พับบลิคเลเบิล

บทนี้จะอธิบาย พับบลิคเลเบิล ซึ่งใช้ในการแลกเปลี่ยนข้อมูลของอุปกรณ์ระหว่าง PLC CPU กับ โมดูลโมชั่น

บทที่ 3 - โหมดบัฟเฟอร์

บทนี้จะอธิบายการควบคุมแบบ synchronous โดยใช้โหมดบัฟเฟอร์ของ Motion control FB

บทที่ 4 - การดำเนินการด้วย PLC CPU

บทนี้จะอธิบายวิธีการสร้างโปรแกรมโดยใช้ Motion control FB บน PLC CPU

บทที่ 5 - การ logging

บทนี้จะอธิบายวิธีการ loggingข้อมูลโดยใช้ GX LogViewer

แบบทดสอบประเมินผล

แบบทดสอบมีทั้งหมด 4 ส่วน (7 ข้อ) เกณฑ์การผ่านหลักสูตร: 60% ขึ้นไป

ไปที่หน้าถัดไป	>	ไปที่หน้าถัดไป
กลับไปยังหน้าที่แล้ว	<	กลับไปยังหน้าที่แล้ว
เลื่อนไปยังหน้าที่ต้องการ	TOC	ระบบจะแสดง "สารบัญ" ช่วยให้คุณสามารถไปยังหน้าต่างๆ ได้
ออกจากการเรียนรู้	X	ออกจากการเรียนรู้ ออกจากการเรียนรู้ระบบจะปิดหน้าต่าง เช่น หน้าจอ "เนื้อหา" และการเรียนรู้

บทนำ ข้อควรระวังในการใช้งาน

■ข้อควรระวังด้านความปลอดภัย

เมื่อใช้ผลิตภัณฑ์จริงเพื่อการเรียนรู้ โปรดอ่าน "ข้อควรระวังด้านความปลอดภัย" ในคู่มือของผลิตภัณฑ์ที่จะใช้อย่างละเอียด และให้ความสำคัญกับความปลอดภัยและการใช้งานที่เหมาะสม

■ข้อควรระวังในหลักสูตรนี้

ภาพหน้าจอที่แสดงในหลักสูตรอาจแตกต่างจากซอฟต์แวร์จริงของคุณขึ้นอยู่กับเวอร์ชันที่ใช้ หลักสูตรนี้ใช้เวอร์ชันของซอฟต์แวร์ต่อไปนี้ สำหรับเวอร์ชันล่าสุดของซอฟต์แวร์ โปรดตรวจสอบที่เว็บไซต์ของ Mitsubishi Electric FA

MELSOFT GX Works3	Ver.1.066U	Motion Control Setting function Ver.1.012N
GX LogViewer	Ver.1.106K	
MELSOFT MR Configurator2	Ver.1.110Q หรือใหม่กว่า	

เวอร์ชันเฟิร์มแวร์ของPLC CPU ต้องเท่ากับ 44 หรือใหม่กว่า (46 หรือใหม่กว่าสำหรับ RD78GH)
เวอร์ชันเฟิร์มแวร์ของโมดูลโมชั่นต้องเท่ากับ 10 หรือใหม่กว่า
สำหรับวิธีอัปเดตเวอร์ชันเฟิร์มแวร์ โปรดดูที่เว็บไซต์ MITSUBISHI ELECTRIC FA หรือคู่มือการกำหนดค่าโมดูล

ไอคอน แสดงถึงคู่มืออ้างอิง
เนื้อหาของคู่มือการใช้งานซึ่งอธิบายอยู่ในหลักสูตรนี้เป็นเนื้อหาสำหรับเวอร์ชันดังต่อไปนี้
หากเวอร์ชันแตกต่างกัน ตำแหน่งของคำอธิบายและเนื้อหาอาจแตกต่างออกไปเล็กน้อย

ชื่อคู่มือ	หมายเลขคู่มือ	เวอร์ชัน
MELSEC iQ-R Motion Module User's Manual (Startup)	IB-0300406	E
MELSEC iQ-R Motion Module User's Manual (Application)	IB-0300411	E
MELSEC iQ-R Motion Module User's Manual (Network)	IB-0300426	E
MELSEC iQ-R Programming Manual (Motion Module Instructions, Standard Functions/Function Blocks)	IB-0300431	E
MELSEC iQ-R Programming Manual (Motion Control Function Blocks)	IB-0300533	C
MELSEC iQ-R Structured Text (ST) Programming Guide Book	SH-081483	F
MELSEC iQ-R Programming Manual (CPU Module Instructions, Standard Functions/Function Blocks)	SH-081266	Z
MELSEC iQ-R CPU Module User's Manual (Application)	SH-081264	AK

บทที่ 1 เนื้อหาของหลักสูตร

1.1 ภาพรวมของหลักสูตร

ต่อไปนี้คือภาพรวมของหลักสูตร

บทที่ 1 เนื้อหาของหลักสูตร

บทนี้จะอธิบายเนื้อหาของหลักสูตร

↓

บทที่ 2 พับบลิคเลเบิล

บทนี้จะอธิบายพับบลิคเลเบิล ซึ่งใช้ในการแลกเปลี่ยนข้อมูลอุปกรณ์ระหว่าง PLC CPU กับ โมดูลโมชั่น

↓

บทที่ 3 โหมดบัฟเฟอร์

บทนี้จะอธิบายการควบคุมแบบ synchronous โดยใช้โหมดบัฟเฟอร์ของ Motion control FB

↓

บทที่ 4 การดำเนินการด้วย PLC CPU

บทนี้จะอธิบายวิธีการสร้างโปรแกรมโดยใช้ Motion control FB บน PLC CPU

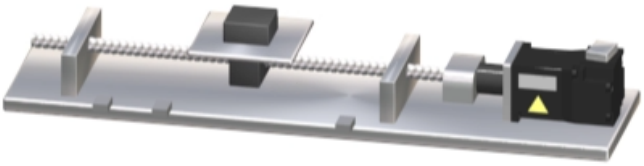
↓

บทที่ 5 - การ logging

บทนี้จะอธิบายวิธีการ logging ข้อมูลโดยใช้ GX LogViewer

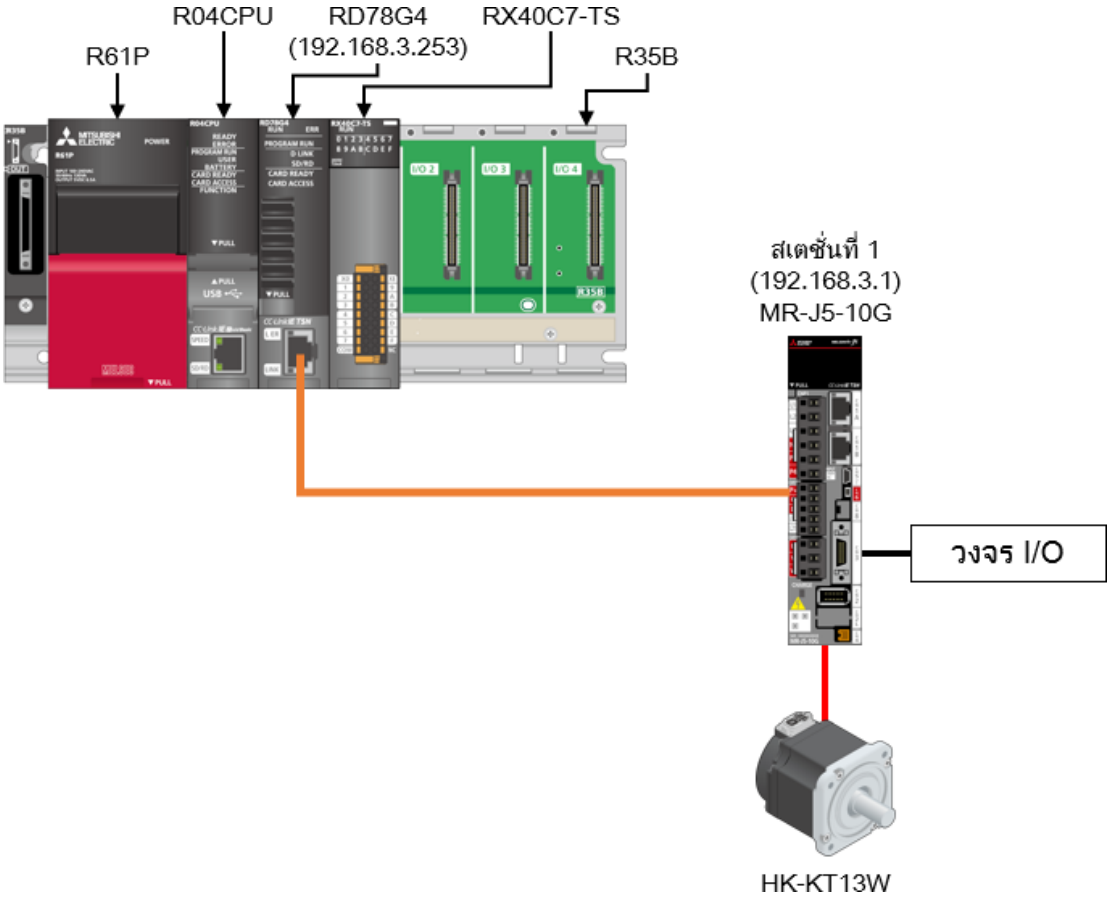
1.2 การกำหนดค่าเครื่องเป้าหมาย

หลักสูตรนี้ใช้กลไกบอลสกรูแกนเดียวแบบเดียวกับที่ใช้ในหลักสูตรเริ่มต้น



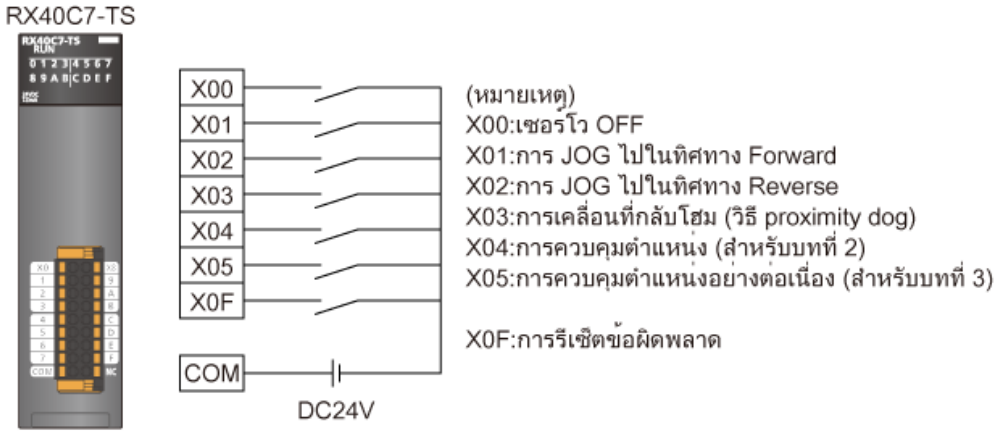
1.3 การกำหนดค่าระบบเป้าหมาย

การกำหนดค่าของระบบเป้าหมายมีดังนี้
ถอดโมดูลอินพุตแบบระยะไกลออกจากระบบที่ใช้ในหลักสูตรเริ่มต้น และเพิ่มโมดูลอินพุต RX40C7-TS ลงในช่อง 1 ของยูนิตฐานของตัวควบคุมแบบตั้งโปรแกรมได้
หมายเลขสถานีของเซอร์โวแอมพลิฟายเออร์ MR-J5-10G เปลี่ยนเป็น 1 และที่อยู่ IP เปลี่ยนเป็น 192.168.3.1



1.4 การเดินสายวงจรภายนอก

การเดินสายสำหรับตัวควบคุมแบบตั้งโปรแกรมได้และเซอร์โวแอมพลิฟายเออร์ และวิธีการเชื่อมต่อของเซอร์โวมอเตอร์เหมือนกับที่อธิบายไว้ในหลักสูตรเริ่มต้น
ต่อไปนี้คือการเดินสายวงจรภายนอกของโมดูลอินพุต



(หมายเหตุ) เนื่องจากหมายเลข I/ORX40C7-TS เท่ากับ 0020H X20 ถึง X25 และ X2F ถูกนำไปใช้ในโปรแกรม

บทที่ 2 พับบลิคเลเบิล

เมื่อโมดูลโมชั่นถูกควบคุมโดยโมดูลอินพุตของตัวควบคุมแบบตั้งโปรแกรมได้ เช่นเดียวกับในระบบที่ใช้ในหลักสูตรนี้ที่อธิบายไว้ในบทที่ 1 CPU PLC และโมดูลโมชั่นจะต้องแลกเปลี่ยนข้อมูลอุปกรณ์

มีสองวิธีดังต่อไปนี้

1. ใช้พับบลิคเลเบิล
2. ใช้หน่วยความจำบัฟเฟอร์ของโมดูลโมชั่น

บทนี้จะอธิบายวิธีการแลกเปลี่ยนข้อมูลโดยใช้ พับบลิค เลเบิล

ดาวน์โหลดโปรแกรมตัวอย่างเพื่อใช้ในบทนี้และบทที่ 3 โดยคลิกที่ลิงค์ด้านล่าง

RD78GBasic2_sample1.zip (1.34MB)

[จุดสำคัญ]
เมื่อใช้หน่วยความจำบัฟเฟอร์ ให้คัดลอกข้อมูลที่จะแลกเปลี่ยนไปยังพื้นที่ผู้ใช้ (U□\G11478000 to G11997999)

(ตัวอย่างโปรแกรม)

<PLC CPU>

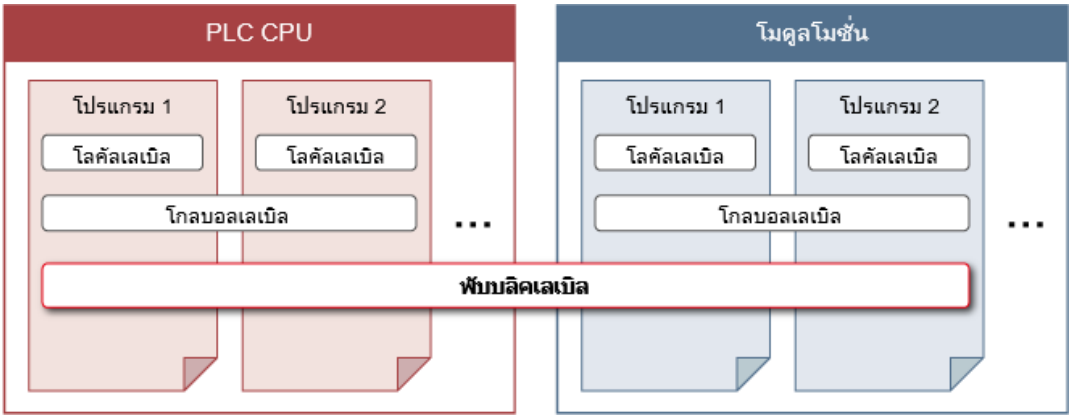
การเริ่มต้นการกำหนดตำแหน่ง

<โมดูลโมชั่น>

```
MC_MoveAboslute_1(
    Execute:= G11478000.0 ,
      :
      :
);
```

2.1 พับบลิคเลเบิล คืออะไร

พับบลิคเลเบิล คือป้ายกำกับที่ใช้ร่วมกันซึ่งสามารถใช้ได้ทั้งในโมดูลโมชั่นและ PLC CPU
ต่อไปนี้คือพื้นที่ที่เกี่ยวข้องของ โลคัลเลเบิล โกลบอลเลเบิล, และ พับบลิคเลเบิล



2.2 การตั้งค่า พับบลิคเลเบิล

(1) วิธีการรีจิสเตอร์ พับบลิคเลเบิล

รีจิสเตอร์ พับบลิคเลเบิลl จากโกลบอลเลเบิลของโมดูลโมชั่น

ตรวจสอบให้แน่ใจว่าคอลัมน์ "Public Label" ปรากฏในเครื่องมือแก้ไขโกลบอลเลเบิลของหน้าจอ Motion Module Setting Function

ตั้งค่าป้ายกำกับที่จะลงทะเบียนเป็น พับบลิคเลเบิล เป็น "Enabled"

การดำเนินการดังกล่าวจะเป็นการเปิดใช้งานคอลัมน์ "Motion Control Attribute"

เลือกว่าจะอ่านหรือเขียนแต่ละ เลเบิล จาก/ไปยัง PLC CPU

	Label Name	Data Type	Class	Initial	Constant	Japanese	English(Display Target)	Chinese	Remark	Public Label	Motion Control Attribute
1	G_bSVONCMD	Bit	VAR_GLOBAL				Servo ON			Enabled	WRITE (=> Motion)
2	G_leJogVelocity	FLOAT [Double Precision]	VAR_GLOBAL				JOG Velocity			Enabled	WRITE (=> Motion)
3	G_bJogFwd	Bit	VAR_GLOBAL				JOG Forward			Enabled	WRITE (=> Motion)
4	G_bJogBwd	Bit	VAR_GLOBAL				JOG Backward			Enabled	WRITE (=> Motion)
5	G_bJogBusy	Bit	VAR_GLOBAL				JOG Busy			Enabled	READ (Motion =>)
6	G_lePosition0	FLOAT [Double Precision]	VAR_GLOBAL				Position0 Address			Disabled	-
7	G_bHomingCMD	Bit	VAR_GLOBAL				Homing Command			Enabled	WRITE (=> Motion)
8	G_bHomingDone	Bit	VAR_GLOBAL				Homing Done			Enabled	READ (Motion =>)
9	G_bHomingReq	Bit	VAR_GLOBAL				Homing Request			Enabled	READ (Motion =>)
10	G_bPosCMD	Bit	VAR_GLOBAL				Positioning Command			Enabled	WRITE (=> Motion)
11	G_bPosDone	Bit	VAR_GLOBAL				Positioning Done			Enabled	READ (Motion =>)
12	G_bPosReq	Bit	VAR_GLOBAL				Positioning Start Request			Enabled	READ (Motion =>)
13	G_bErrorReset	Bit	VAR_GLOBAL				Error Reset			Enabled	WRITE (=> Motion)
14	G_bContPosCMD	Bit	VAR_GLOBAL				Continuous Positioning Command			Enabled	WRITE (=> Motion)
15	G_bContPosReq	Bit	VAR_GLOBAL				Continuous Positioning Start Request			Enabled	WRITE (=> Motion)
16	G_bContPosDone	Bit	VAR_GLOBAL				Continuous Positioning Done			Enabled	READ (Motion =>)
17											

[จุดสำคัญ]
หากไม่เห็นคอลัมน์ของ พับบลิคเลเบิล ให้เลื่อนตารางไปทางขวา

(2) ประเภทข้อมูลที่สามารถลงทะเบียนเป็น พับบลิคเลเบิล ได้
ตารางต่อไปนี้แสดงประเภทข้อมูลที่สามารถรีจิสเตอร์เป็นพับบลิคเลเบิลได้

ประเภทตัวแปร	ประเภท	การเลือกอาร์เรย์	การตั้งค่าพับบลิคเลเบิล	หมายเหตุ
โกลบอลเลเบิล	Simple type	ไม่	○	ไม่สามารถตั้งค่าได้สำหรับป้ายกำกับและคลาสต่อไปนี้ ■เลเบิล • String type label • Timer type label • Counter type label • Long counter type label • Retentive timer label • Long retentive time type label • Long timer type label ■คลาส • คลาส VAR_GLOBAL_CONSTANT
		ใช่	△(หมายเหตุ 1,2)	
	Structured data type	ไม่	△(หมายเหตุ 3)	
		ใช่	△(หมายเหตุ 1,2,4,5)	
	FB (รวมถึง Motion control FB)	ไม่	×	
		ใช่	×	
Program	-	-	×	
Program block local label	-	-	×	
Structured data type	-	-	△(หมายเหตุ 3,5)	
Motion control FB structured data	-	-	△(หมายเหตุ 6,7)	

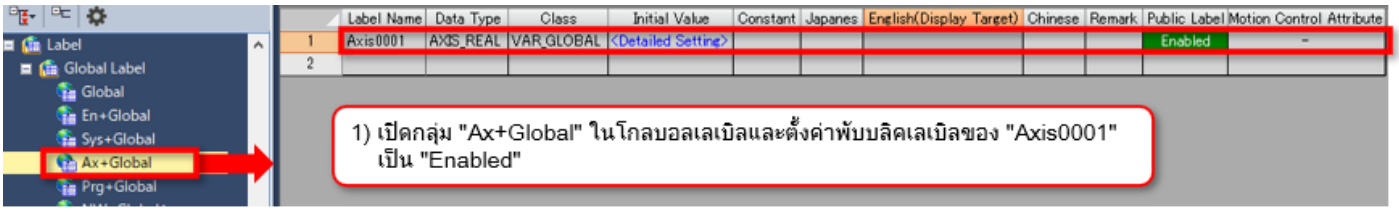
(หมายเหตุ)
1. การตั้งค่าพับบลิคเลเบิล ไม่สามารถกำหนดแต่ละองค์ประกอบของอาร์เรย์ได้
2. เมื่อใช้อาร์เรย์ประเภทบิต พับบลิคเลเบิล จะไม่สามารถตั้งค่าเป็น "Enabled" ได้ (ใน structured data เฉพาะสมาชิกที่เกี่ยวข้องเท่านั้นที่ไม่สามารถตั้งค่าเป็น "Enabled" ได้)
3. เมื่อใช้ประเภทสตริงเป็นสมาชิกของประเภท structured data สมาชิกจะไม่สามารถตั้งค่าเป็น "Enabled" ได้
4. Structured data ที่มีชั้นสูงสุดสี่ชั้นสามารถเปิดเผยต่อสาธารณะได้
5. เมื่อใช้อาร์เรย์ของ structured data เป็นสมาชิกของ structured data สมาชิกจะไม่สามารถตั้งค่าเป็น "Enabled" ได้
6. สามารถใช้ในโปรแกรมของ PLCopen Motion control FB โดยโมดูล CPU ได้
7. เมื่อใช้ประเภทสตริงใน Motion control FB structured data จะไม่สามารถตั้งค่าประเภท Motion control FB structured data ได้

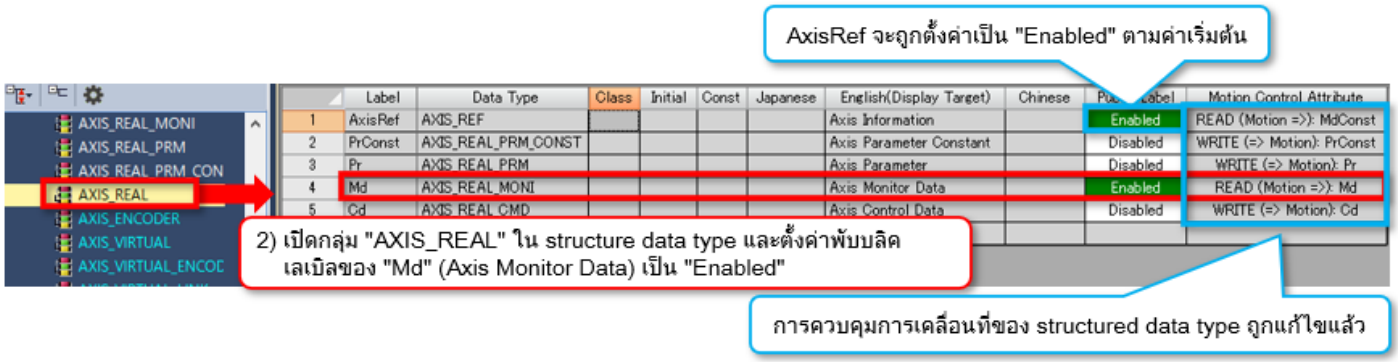
(3) วิธีการรีจิสเตอร์ structured data เป็น โลคัลเลเบิล

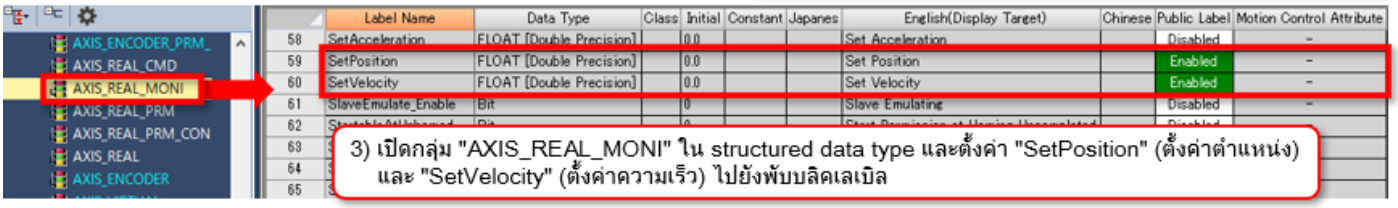
หากต้องการตั้งค่าข้อมูลของ structured data ที่เตรียมไว้ในระบบ เช่น ข้อมูลการตรวจสอบแกนให้เป็นพับบลิคเลเบิล ให้รีจิสเตอร์ตามชั้นของ structured data ดังที่แสดงด้านล่าง

หลักสูตรนี้จะอธิบายวิธีการรีจิสเตอร์การตั้งค่าตำแหน่ง (SetPosition) และการตั้งค่าความเร็ว (SetVelocity) ข้อมูลการมอนิเตอร์ (Md) ของแกนที่กำลังหมุน (Axis_Real) เป็นป้ายกำกับสาธารณะ

[วิธีการตั้งค่า AxisName.Md.SetPosition (ตำแหน่งปัจจุบันของคำสั่ง) และ AxisName.Md.SetVelocity (ความเร็วปัจจุบันของคำสั่ง) ให้เป็นโลคัล เลเบิล]



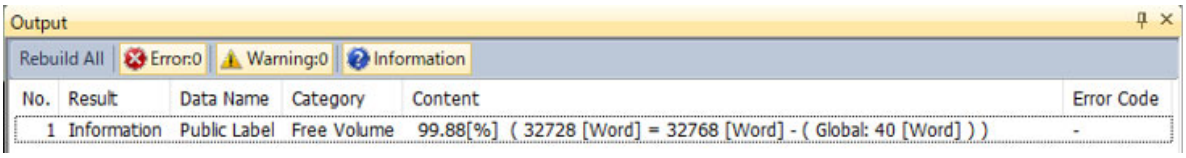




(4) Reflecting the public labels

เลือก [Convert] → [Rebuild All] ในเมนู

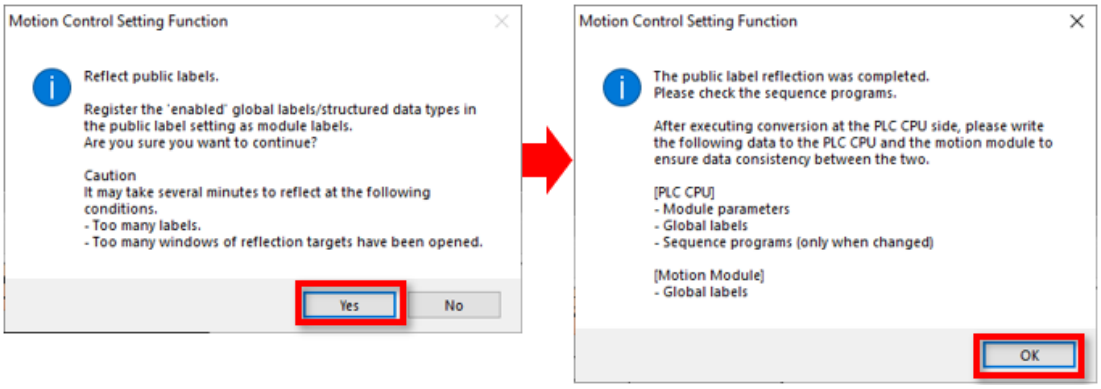
ความจุของพื้นที่ว่างของ พับบลิคเลเบิล จะถูกแสดงเป็นข้อมูลในหน้าต่างเอาท์พุต



เมื่อกระบวนการ rebuild all ทั้งหมดเสร็จสมบูรณ์แล้ว เลือกเมนู [Convert] → [Reflect Public Labels]

คลิก [Yes] ในหน้าต่างป๊อปอัปต่อไปนี้

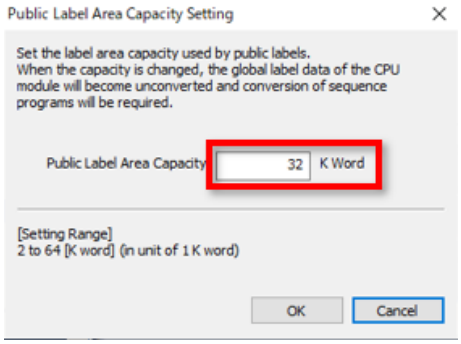
เมื่อมีข้อความระบุว่าพับบลิคเลเบิล ถูก reflected สำเร็จแล้ว คลิกปุ่ม [OK]



(หมายเหตุ) ต่าความจุเริ่มต้นของหน่วยความจำที่สามารถใช้เพื่อริจิสเตอร์ พับบลิคเลเบิล คือ 32K words

สามารถเพิ่มความจุได้สูงสุด 64K words

หากต้องการเปลี่ยนค่าความจุ ตั้งค่าขนาดหน่วยความจำจากเมนู [Convert] → [Public Label Capacity Setting]



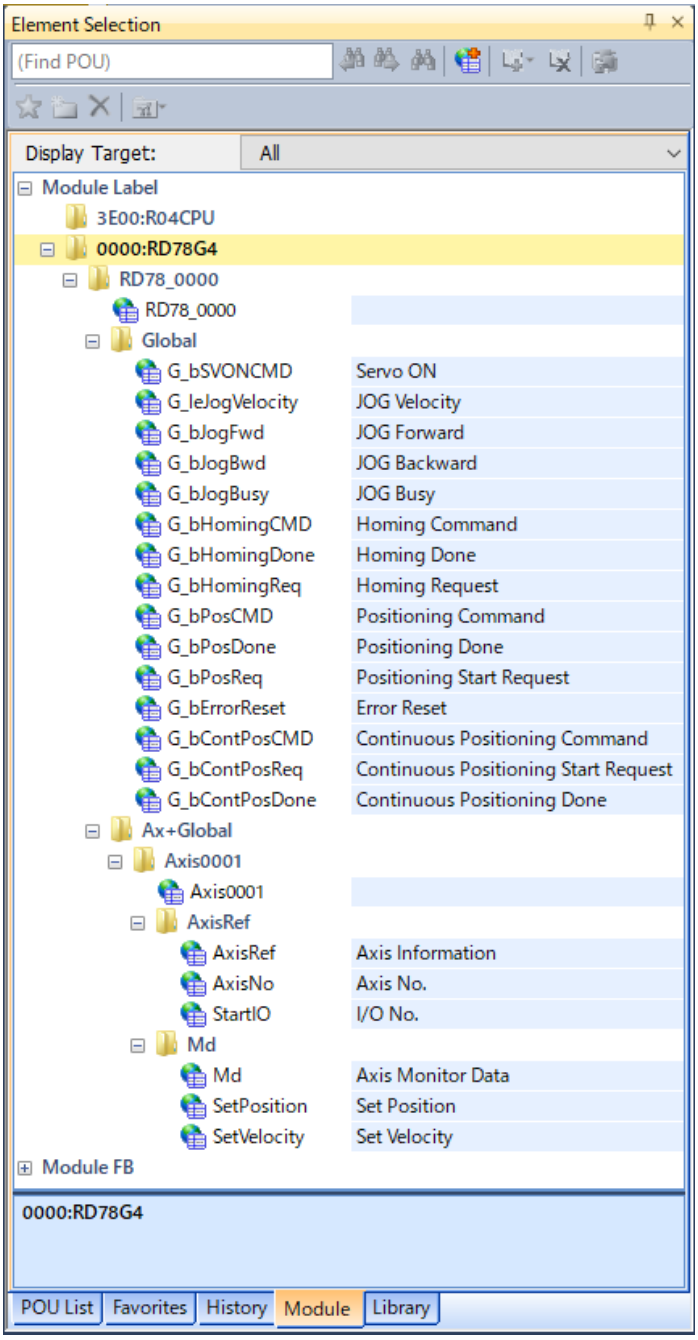
(5) การตรวจสอบเลเบิลจาก PLC CPU

พับบลิค เลเบิล ที่สะท้อนจะถูกลงรีจิสเตอร์กับโมดูล เลเบิลที่ PLC CPU

เลือกโมดูลเลเบิล จากหน้าต่าง Element Selection ของ GX Works3 และตรวจสอบว่ามีการรีจิสเตอร์ พับบลิคเลเบิลแล้ว [0000:RD78G4] ใน [Module Label]

หลังจากเปลี่ยนการตั้งค่าพับบลิคเลเบิลแล้ว ให้ดำเนินการ "Reflect Public Labels" อีกครั้งเสมอ

เมื่อใช้พับบลิค เลเบิล ใน PLC CPU ให้ rebuild all ใหม่

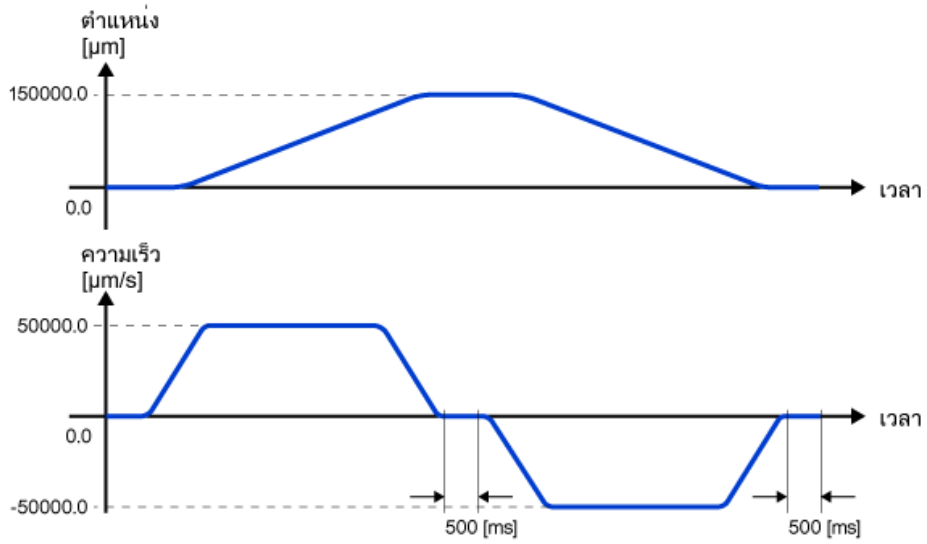


2.3 ตัวอย่างโปรแกรม

(1) ตัวอย่างโปรแกรมการควบคุม
สัญญาณอินพุตของโปรแกรมตัวอย่างที่ใช้ในบทนี้ถูกกำหนดไว้ดังนี้

อินพุต	การดำเนินการ
X20	Servo off (หมายเหตุ)
X21	การดำเนินการ JOG การหมุนเดินหน้า
X22	การดำเนินการ JOG แบบหมุนถอยหลัง
X23	การเคลื่อนที่กลับโฮม
X24	การควบคุมตำแหน่ง
X25	การควบคุมตำแหน่งอย่างต่อเนื่อง (บทที่ 3)

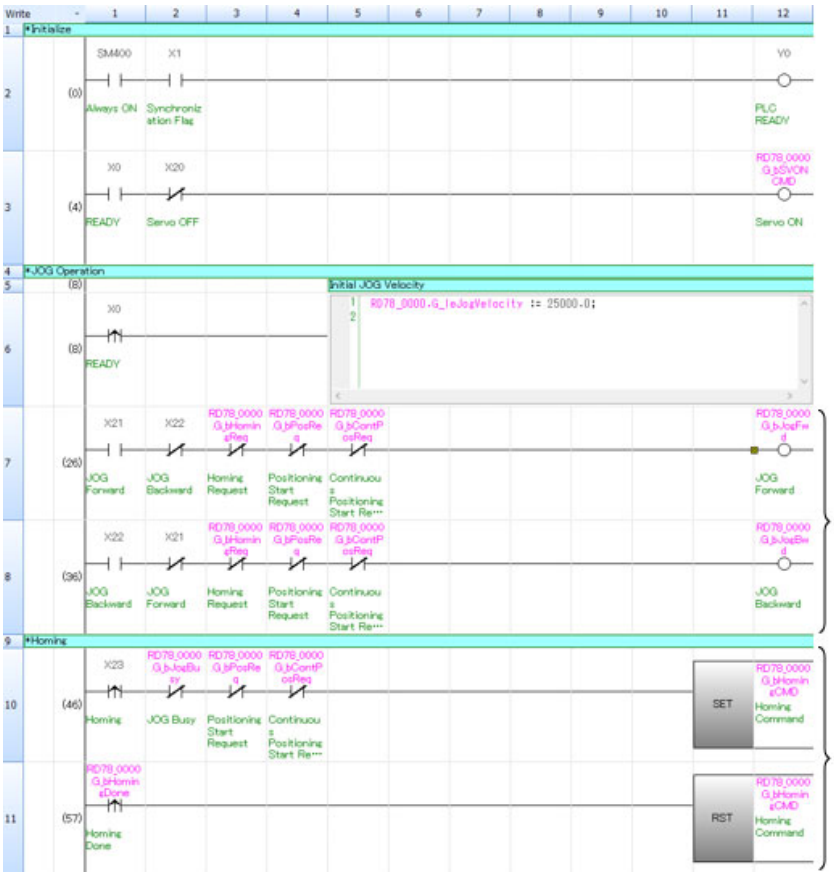
ต่อไปนี้จะแสดงรูปแบบการดำเนินการของ X24: การควบคุมตำแหน่ง



(หมายเหตุ) โปรแกรมตัวอย่างนี้รันเซอร์โว ON โดยอัตโนมัติเมื่อ PLC CPU ถูกตั้งค่าเป็น RUN เมื่อเปิดเครื่องโดยมีสัญญาณสตาร์ท ON เซอร์โวมอเตอร์อาจทำงาน

(2) โปรแกรมของ PLC CPU

1) MAIN (แลดเดอร์ สแกนโปรแกรม)



ON Y0 ก่อน

เปิด X0 เซอร์โวจะถูกดำเนินการ ON
เปิด X20 เพื่อดำเนินการเซอร์โว OFF

ตั้งค่าเริ่มต้นของความเร็ว JOG
โปรแกรมนี้ใช้ inline ST
เนื่องจากการควบคุมการเคลื่อนไหวของโกลบอลเลเบิล "G_JogVelocity" ที่เก็บความเร็ว JOG ถูกตั้งค่าเป็น "WRITE (→Motion)" จึงต้องตั้งค่าตัวเลขใน PLC CPU

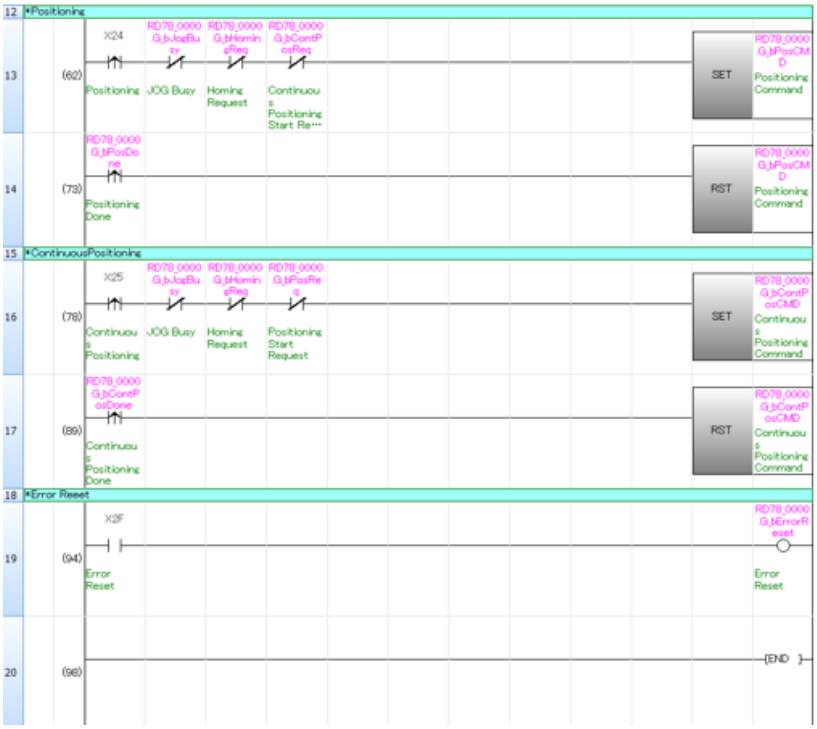
เปิดสัญญาณเริ่มต้นของการดำเนินการ JOG
การดำเนินการดังกล่าวจะป้องกันไม่ให้หมุนไปในทิศทาง Forward และ Reverse พร้อมกัน
อินเทอร์ล็อคถูกตั้งค่าเพื่อป้องกันไม่ให้เริ่มการดำเนินการ JOG ในขณะที่โปรแกรมอื่นกำลังดำเนินการอยู่

การเคลื่อนที่กลับโฮม (X23) จะยังคงอยู่ในพับบลิคเลเบิล G_bHomingCMD และถูกส่งไปยังโมดูลโมชั่นเป็นเงื่อนไขการเริ่มต้นของการเคลื่อนที่กลับโฮม
อินเตอร์ล็อคถูกตั้งค่าเพื่อป้องกันการเคลื่อนที่กลับโฮมในขณะที่โปรแกรมอื่นกำลังทำงานอยู่
เมื่อได้รับแจ้งว่าโมดูลโมชั่นเปิดสัญญาณการเคลื่อนที่กลับโฮมเสร็จแล้ว G_bHomingCMD จะถูกรีเซ็ตที่ขอบที่ขอบขาขึ้นของสัญญาณนั้น

(2) โปรแกรมของ PLC CPU

1) ส่วนที่ต่อเนื่องของ MAIN (แลดเดอร์ สแกนโปรแกรม)

(ต่อจากหน้าที่แล้ว)



ขอบขาขึ้นของการเริ่มต้นการควบคุมตำแหน่ง (X24) จะยังคงอยู่ใน G_bPosCMD และถูกส่งไปยังโมดูลโมชั่นเป็นเงื่อนไขการเริ่มต้นของการควบคุมตำแหน่ง
อินเตอร์ล็อคได้รับการตั้งค่าเพื่อป้องกันไม่ให้เริ่มการควบคุมตำแหน่งในขณะที่โปรแกรมอื่นกำลังทำงานอยู่
เมื่อได้โมดูลโมชั่นเปิดสัญญาณการเคลื่อนที่กลับโฮมเสร็จแล้ว G_bPosCMD จะถูกรีเซ็ตที่ขอบที่ขอบขาขึ้นของสัญญาณนั้น

การเริ่มต้นตำแหน่งแบบต่อเนื่องที่อธิบายไว้ในบทที่ 3 สำหรับรายละเอียด โปรดดูข้อ 3.4

ข้อผิดพลาดจะถูกรีเซ็ตเมื่อเปิด X2F

(2) โปรแกรมของ PLC CPU

2) MONITOR (ST สแกนโปรแกรม)

SetPosition (ตั้งค่าตำแหน่ง) และ SetVelocity (ตั้งค่าความเร็ว) ของการมอนิเตอร์แกนที่ตั้งเป็นพับบลิคเลเบิล จะถูกจัดเก็บไว้ใน D0 และ D2

เนื่องจาก SetPosition และ SetVelocity เป็นประเภท double precision real number ทั้งสองจะถูกแปลงเป็น signed double word พื่อให้สามารถจัดการได้ง่ายโดย PLC CPU (หมายเหตุ)

แม้ว่าไม่ได้ใช้อุปกรณ์เหล่านี้ในหลักสูตร อุปกรณ์เหล่านี้จะถูกใช้เพื่อแสดงข้อมูลในโปรแกรมลำดับอื่นและ GOT และเพื่อวัตถุประสงค์อื่น

```
1  D0:D := LREAL_TO_DINT(RD78_0000.Axis0001.Md.SetPosition);
2  D2:D := LREAL_TO_DINT(RD78_0000.Axis0001.Md.SetVelocity);
3
```

ระบุ signed double word type เป็น "D0:D"

(หมายเหตุ) เมื่อแปลงประเภทdouble precision real number เป็นประเภท signed double word แล้ว ถ้าค่าที่จะแปลงอยู่นอกช่วงตั้งแต่ -2147483648 ถึง 2147483647 จะเกิดข้อผิดพลาดในการคำนวณ

(3) โปรแกรมของโมดูลโมชั่น

1) ServoON_JOG (ประเภทการดำเนินการปกติ)

```
1  //-----Servo ON-----
2  MC_Power_1(
3      Axis     := Axis0001.AxisRef,
4      Enable   := TRUE ,
5      ServoON  := G_bSVONCMD,
6      Busy     => bPowerBusy
7  );
8
9  //-----JOG Operation-----
10 //Initial Value Setting
11 IF (bPowerBusy) THEN
12     leJogAcceleration := 50000.0;
13     leJogDeceleration := 50000.0;
14     leJogJerk         := 0.0;
15 END_IF;
16
17 //JOG
18 MCv_Jog_1(
19     Axis         := Axis0001.AxisRef ,
20     JogForward   := G_bJogFwd ,
21     JogBackward  := G_bJogBwd ,
22     Velocity     := G_leJogVelocity ,
23     Acceleration:= leJogAcceleration ,
24     Deceleration:= leJogDeceleration ,
25     Jerk         := leJogJerk ,
26     Busy         => G_bJogBusy
27 );
28
```

รับสัญญาณเซอร์โว ON (G_bSVONCD) จาก PLC CPU และดำเนินการเซอร์โว ON

จัดเก็บค่าของการเร่งความเร็ว การลดความเร็ว และการกระตุกของการ JOG ไว้ที่เลเบิลเมื่อสัญญาณ Busy ของ MC_Power_1 ON

รับสัญญาณเริ่มต้น JOG และความเร็ว JOG จาก PLC CPU

ส่งกลับเอาต์พุต Busy ไปยัง PLC CPU

(หมายเหตุ) ในโปรแกรมตัวอย่างนี้ สัญญาณ I/O ของ FB ที่ไม่ได้ใช้หรือไม่ได้เปลี่ยนไปจากค่าเริ่มต้นจะถูกละเว้น

(3) โปรแกรมของโมดูลโมชั่น

2) การเคลื่อนที่กลับโฮม (ประเภทการดำเนินการปกติ)

```
1  //-----Homing Operation-----
2  //Initial Value Setting, Operation Start Request
3  IF G_bHomingCMD THEN
4      G_TePosition0 := 0.0 ;
5      G_bHomingReq  := TRUE ;
6  ELSE
7      G_bHomingReq  := FALSE ;
8  END_IF;
9
10 //Homing
11 MC_Home_1(
12     Axis            := Axis0001.AxisRef ,
13     Execute         := G_bHomingReq,
14     Position        := G_TePosition0 ,
15     Done            => bHomingDone ,
16     Busy            => G_bHomingBusy ,
17     CommandAborted  => bHomingAborted ,
18     Error           => bHomingError
19 );
20
21 //Done Signal => PLC CPU
22 G_bHomingDone := bHomingDone OR bHomingAborted OR bHomingError;
23
```

รับสัญญาณคำสั่งการเคลื่อนที่กลับโฮม (G_bHomingCMD) จาก PLC CPU
จัดเก็บตำแหน่งโฮมไว้ที่เลเบิลและเปิดคำสั่งการเคลื่อนที่กลับโฮม (G_bHomingReq)

ปิด G_bHomingReq เมื่อปิด G_bHomingCMD

ส่งคืนสัญญาณการดำเนินการแล้วเสร็จไปยัง PLC CPU หลังจากที่สัญญาณการเคลื่อนที่กลับโฮมเสร็จแล้ว (เอาต์พุต Done ON), การดำเนินการถูกขัดจังหวะ (เอาต์พุต CommandAborted ON), หรือมีข้อผิดพลาดเกิดขึ้น (เอาต์พุต Error ON)

(3) โปรแกรมของโมดูลโมชั่น

3) การวางตำแหน่ง (ประเภทการดำเนินการปกติ)

```
1  //-----Positioning Operation-----
2  //Initial Value Setting, Operation Start Request
3  IF G_bPosCMD THEN
4      lePosition1        := 150000.0;
5      lePosVelocity      := 50000.0;
6      lePosAcceleration  := 100000.0;
7      lePosDeceleration  := 100000.0;
8      lePosJerk          := 200000.0;
9      G_bPosReq := TRUE;
10 ELSE
11     G_bPosReq := FALSE;
12 END_IF;
13
14 //Positioning1
15 MC_MoveAbsolute_1(
16     Axis            := Axis0001.AxisRef ,
17     Execute         := G_bPosReq ,
18     Position        := lePosition1 ,
19     Velocity        := lePosVelocity ,
20     Acceleration    := lePosAcceleration ,
21     Deceleration    := lePosDeceleration ,
22     Jerk            := lePosJerk ,
23     Direction       := MC_DIRECTION_mcShortestWay ,
24     BufferMode      := MC_BUFFER_MODE_mcAborting ,
25     Done            => bMoveAbs1Done ,
26     CommandAborted  => bMoveAbs1Aborted ,
27     Error           => bMoveAbs1Error
28 );
29 //Dwell
30 TON_1(IN:= bMoveAbs1Done ,PT:= T#500ms ,Q=> bDwell1_out );
31 //Positioning2
32 MC_MoveAbsolute_2(
33     Axis            := Axis0001.AxisRef ,
34     Execute         := bDwell1_out ,
35     Position        := G_lePosition0 ,
36     Velocity        := lePosVelocity ,
37     Acceleration    := lePosAcceleration ,
38     Deceleration    := lePosDeceleration ,
39     Jerk            := lePosJerk ,
40     Direction       := MC_DIRECTION_mcShortestWay ,
41     BufferMode      := MC_BUFFER_MODE_mcAborting ,
42     Done            => bMoveAbs2Done ,
43     CommandAborted  => bMoveAbs2Aborted ,
44     Error           => bMoveAbs2Error
45 );
46 //Dwell
47 TON_2(IN:= bMoveAbs2Done ,PT:= T#500ms ,Q=> bDwell2_out );
48
49 //Error Signal, Aborted Signal
50 bPosError   := bMoveAbs1Error OR bMoveAbs2Error;
51 bPosAborted := bMoveAbs1Aborted OR bMoveAbs2Aborted;
52
53 //Done Signal => PLC CPU
54 G_bPosDone := bDwell2_out OR bPosError OR bPosAborted;
55
```

รับสัญญาณเริ่มต้นการควบคุมตำแหน่ง (G_bPosCMD) จาก PLC CPU ข้อมูลที่จำเป็นสำหรับการควบคุมตำแหน่งจะถูกจัดเก็บไว้ในเลเบิล และคำขอเริ่มต้นการควบคุมตำแหน่ง (G_bPositioningReq) จะถูกเปิดใช้งาน

ปิด G_bPositioningReq หลังจากที่ G_bPosCMD ถูกปิดโดย CPU PLC

MC_MoveAbsolute จะดำเนินการวางตำแหน่ง

เมื่อทำงานจบเอาต์พุตของ MC_MoveAbsolute_1 จะเปิด เพื่อจะเริ่มหน่วงเวลา

หลังจากหยุดอยู่กับที่ (dwell time) เสร็จสิ้น MC_MoveAbsolute จะดำเนินการตำแหน่งอีกครั้ง

เริ่มการหน่วงเวลาเมื่อเอาต์พุต Done ของ MC_MoveAbsolute_2 เปิดอยู่

ส่งคืนสัญญาณการดำเนินการแล้วเสร็จไปยัง PLC CPU หลังจากหมดเวลาหยุดอยู่กับที่ (dwell time) หรือเอาต์พุต Error หรือเอาต์พุต CommandAborted ของ MC_MoveAbsolute เปิดอยู่

(3) โปรแกรมของโมดูลโมชั่น

4) รีเซ็ตข้อผิดพลาด (ประเภทการดำเนินการปกติ)

```
1   //Axis Error Reset
2   MC_Reset_1(
3       Axis    := Axis0001.AxisRef ,
4       Execute := G_bErrorReset
5   );
6
7   //System Error Reset
8   MCv_MotionErrorReset_1(
9       Execute := G_bErrorReset
10  );
11
```

รับสัญญาณรีเซ็ตข้อผิดพลาด (G_bErrorReset) จาก PLC CPU และดำเนินการรีเซ็ตข้อผิดพลาดของแกนและการรีเซ็ตข้อผิดพลาดของระบบ

2.4 การเขียนโปรแกรม

เขียนโปรแกรมและพารามิเตอร์ไปยัง PLC CPU และโมดูลโมชั่น

1) หลังจากที่โปรแกรมทั้งหมดใน PLC CPU ถูกr ebuilt แล้ว เลือก [Online] → [Write to PLC] ในแถบเครื่องมือของ GX Works3 เพื่อเขียนข้อมูลทั้งหมดไปยัง PLC CPU

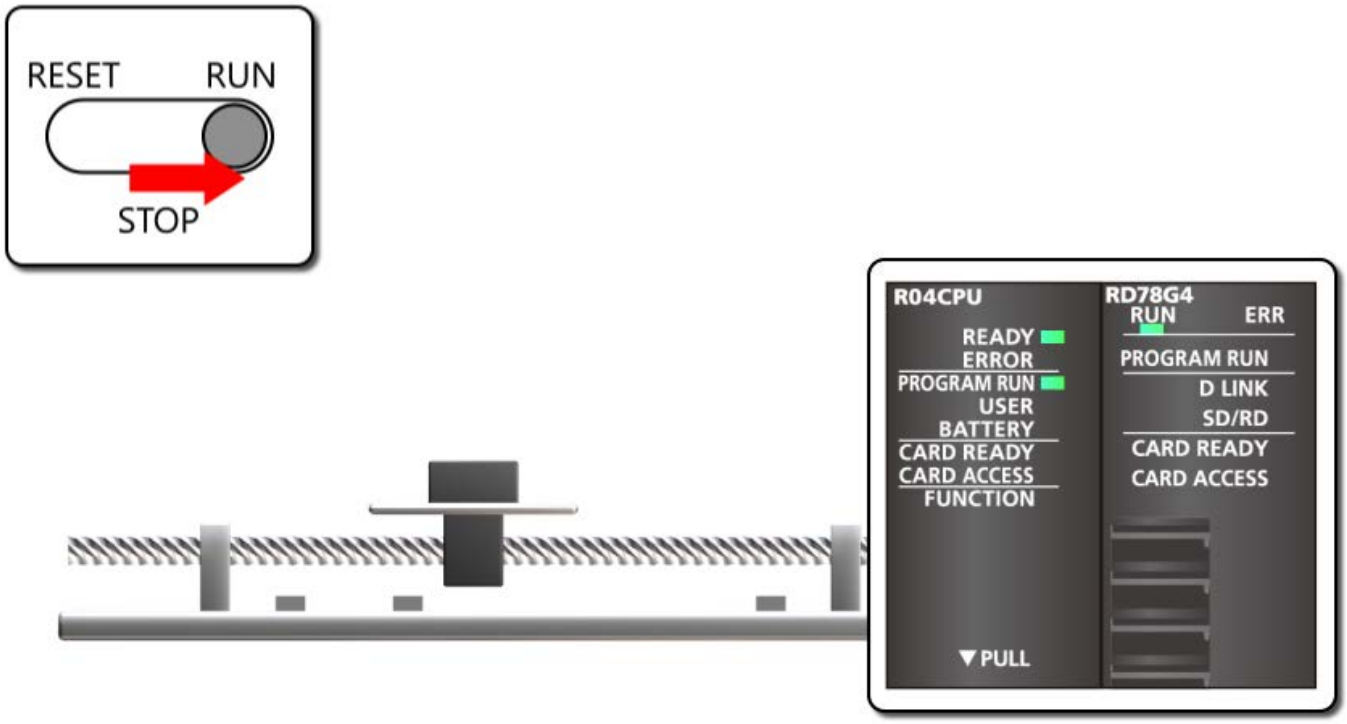
2) เมื่อเขียนพารามิเตอร์ไปยัง PLC CPU การสื่อสารกับโมดูลโมชั่นจะถูกเปิดใช้งาน
เลือก [Online] → [Write to Module] ในแถบเครื่องมือของ Motion Control Setting Function เพื่อเขียนข้อมูลทั้งหมดไปยังโมดูลโมชั่น

3) รีเซ็ต PLC CPU เพื่อสิ้นสุดการดำเนินการเขียน

คลิกปุ่มเล่นที่ด้านซ้ายล่างของหน้าต่าง

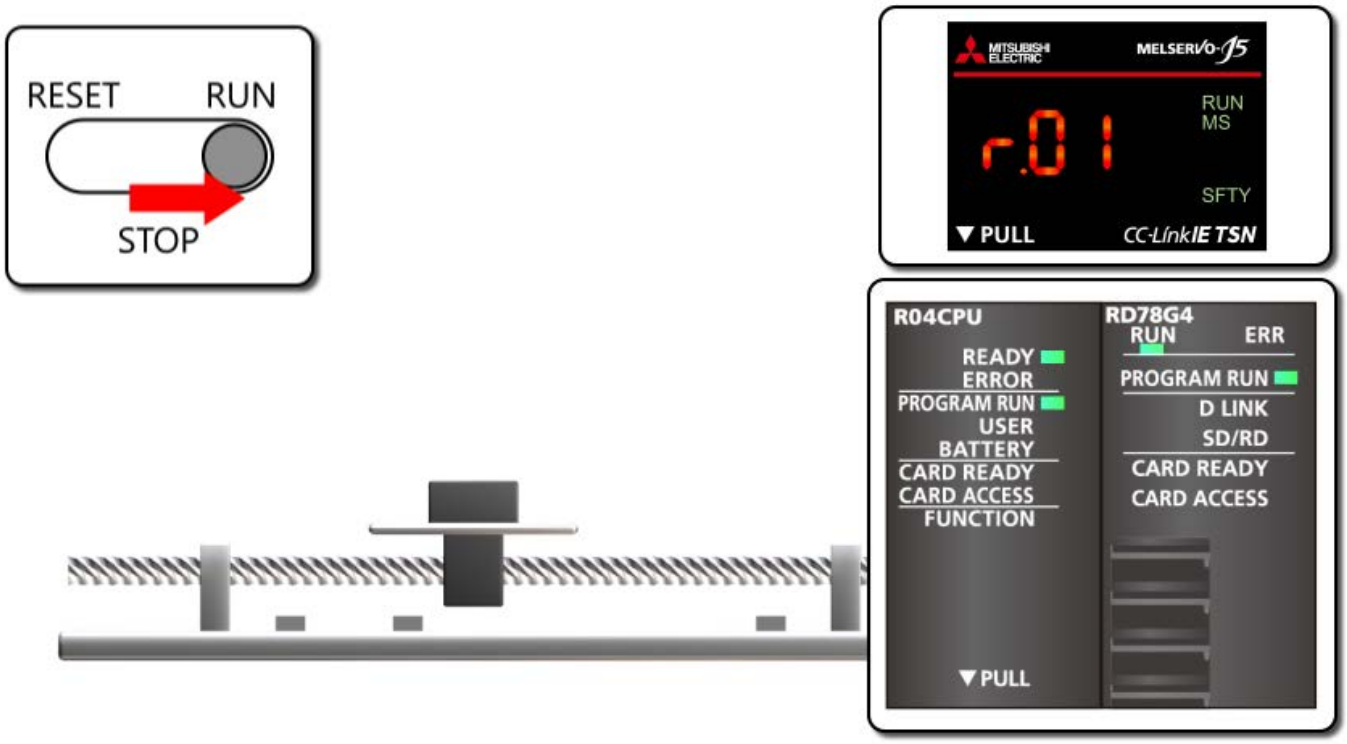
ตรวจสอบการดำเนินการของโปรแกรมตัวอย่าง
ก่อนเริ่มการดำเนินการ ตรวจสอบให้แน่ใจว่าได้เขียนโปรแกรมของ
PLC CPU และโมดูลโมชั่นแล้ว

2.5 การตรวจสอบการดำเนินการ



ตั้งค่าสวิตช์ RUN/STOP/RESET ของ PLC CPU เป็น RUN
ไฟ READY และไฟ PROGRAM RUN ของ PLC CPU จะติด
ไฟ RUN ของโมดูลโมชั่นจะติด

2.5 การตรวจสอบการดำเนินการ

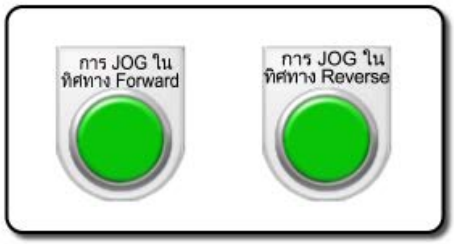


รอจนกว่าไฟ PROGRAM RUN ของโมดูลโมชั่นจะติด
"r.01" จะแสดงบนเซอร์โวแอมพลิฟายเออร์ (จุดจะติดค้าง)
เซอร์โวมอเตอร์จะเข้าสู่สถานะเซอร์โว ON

2.5 การตรวจสอบการดำเนินการ

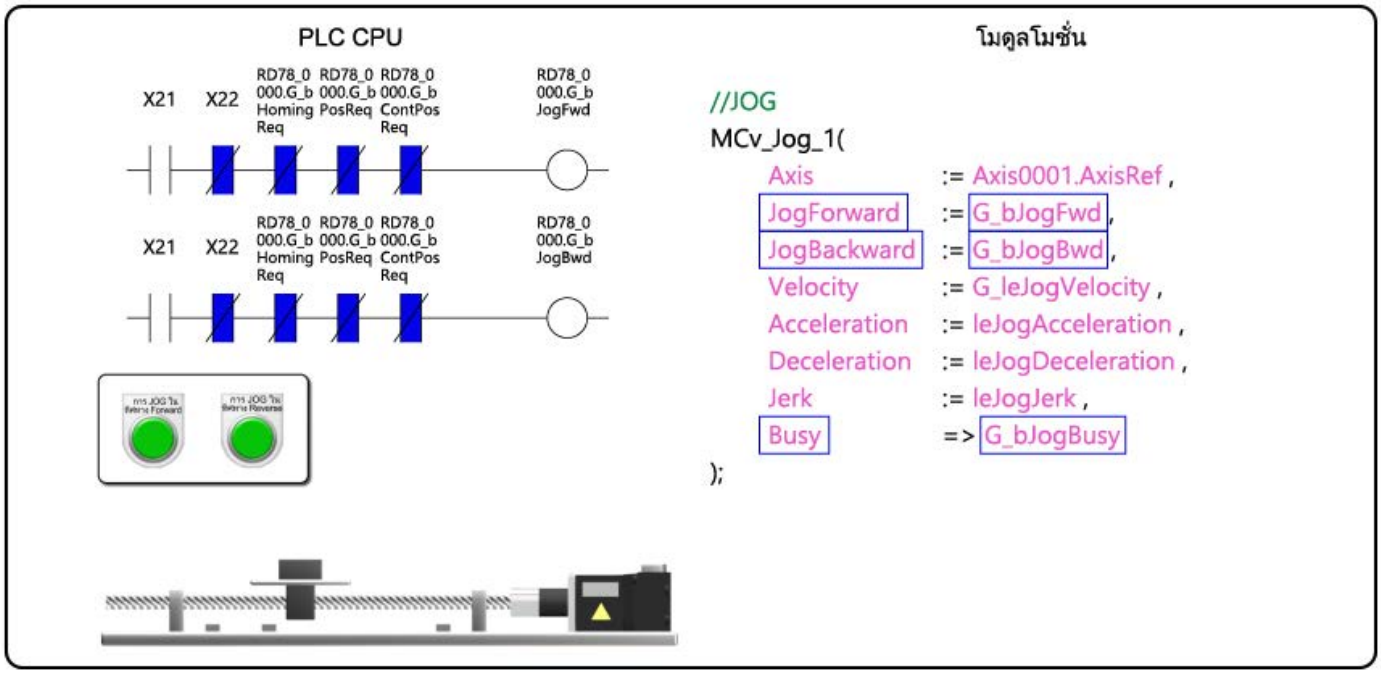
เปิด X20 เพื่อดำเนินการเซอร์โว OFF
"r.01" จะแสดงบนเซอร์โวแอมพลิฟายเออร์ (จุดจะกระพริบ)
ปิด X20 เพื่อดำเนินการเซอร์โว ON อีกครั้ง

2.5 การตรวจสอบการดำเนินการ



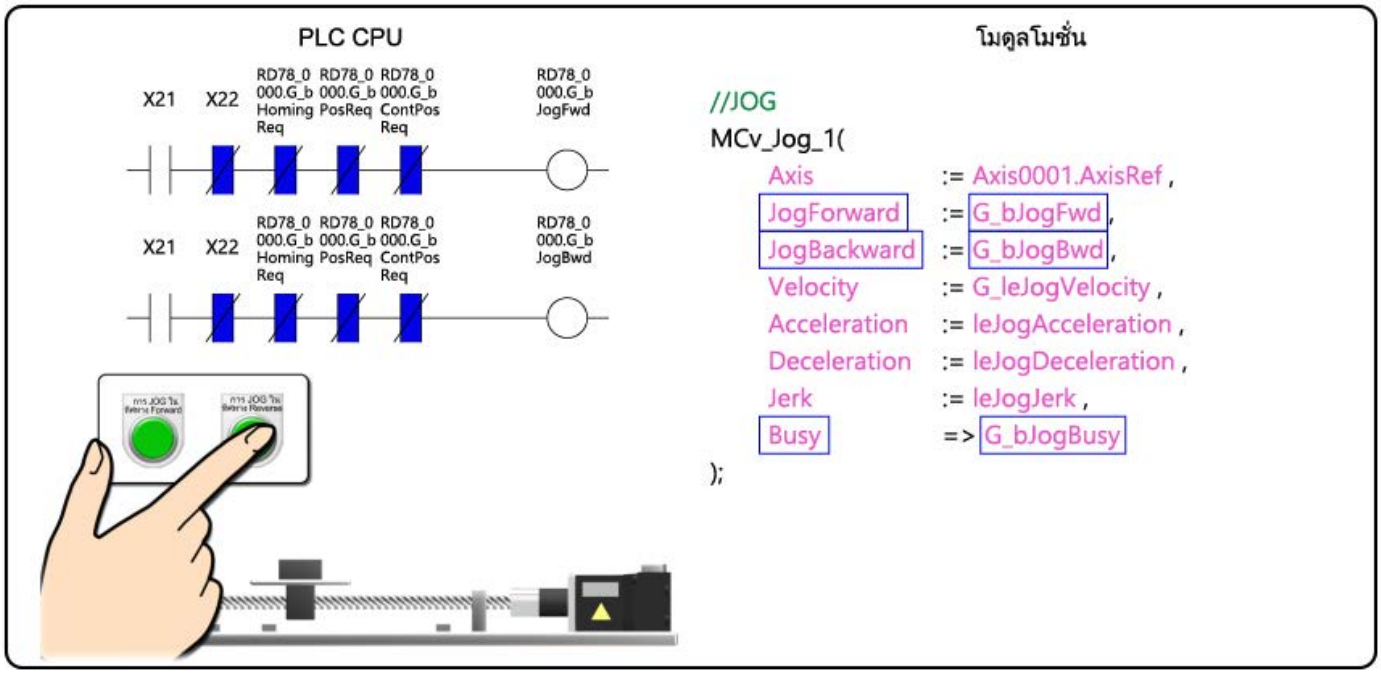
เปิดการหมุน JOG ไปทิศทาง Forward (X21)
address จะมีตัวเลขที่เพิ่มขึ้น และปิดเพื่อหยุด
เปิดการหมุน JOG ไปทิศทาง Reverse (X22)
address จะมีตัวเลขที่ลดลง และปิดเพื่อหยุด

2.5 การตรวจสอบการดำเนินการ



ตรวจสอบโดยการมอนิเตอร์โปรแกรม
เมื่อ X21 เปิดอยู่ "RD78_0000.G_bJogFwd" จะเปิด และ
"G_bJogFwd" ที่ด้านโมดูลโมชั่นจะเปิด
เมื่ออินพุต JogForward ของ MCv_Jog_1 เปิด การ JOG ในทิศทาง Forward จะเริ่มต้น

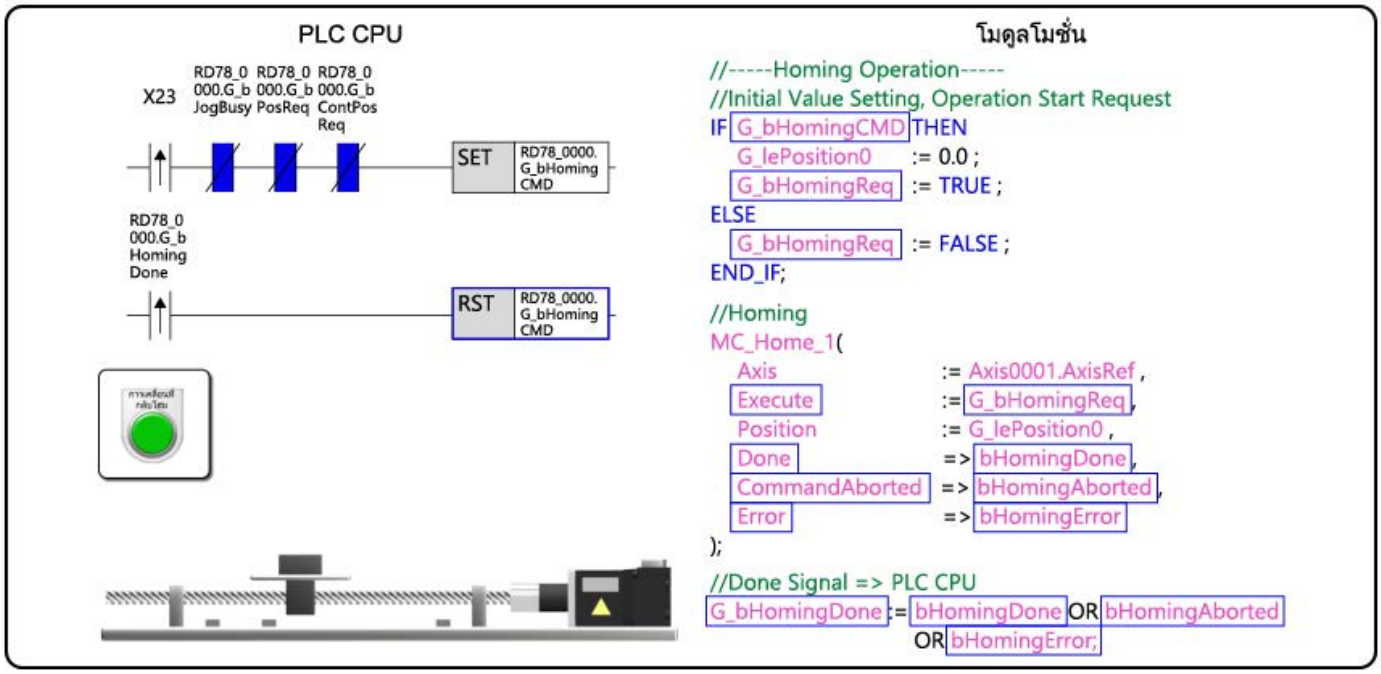
2.5 การตรวจสอบการดำเนินการ



เมื่อ X22 เปิดอยู่ "RD78_0000.G_bJogBwd" จะเปิด และ "G_bJogBwd" ที่ด้านโมดูลโมชั่นจะเปิด
เมื่ออินพุต JogBackward ของ MCv_Jog_1 เปิด การ JOG ในทิศทาง Reverse จะเริ่มต้น

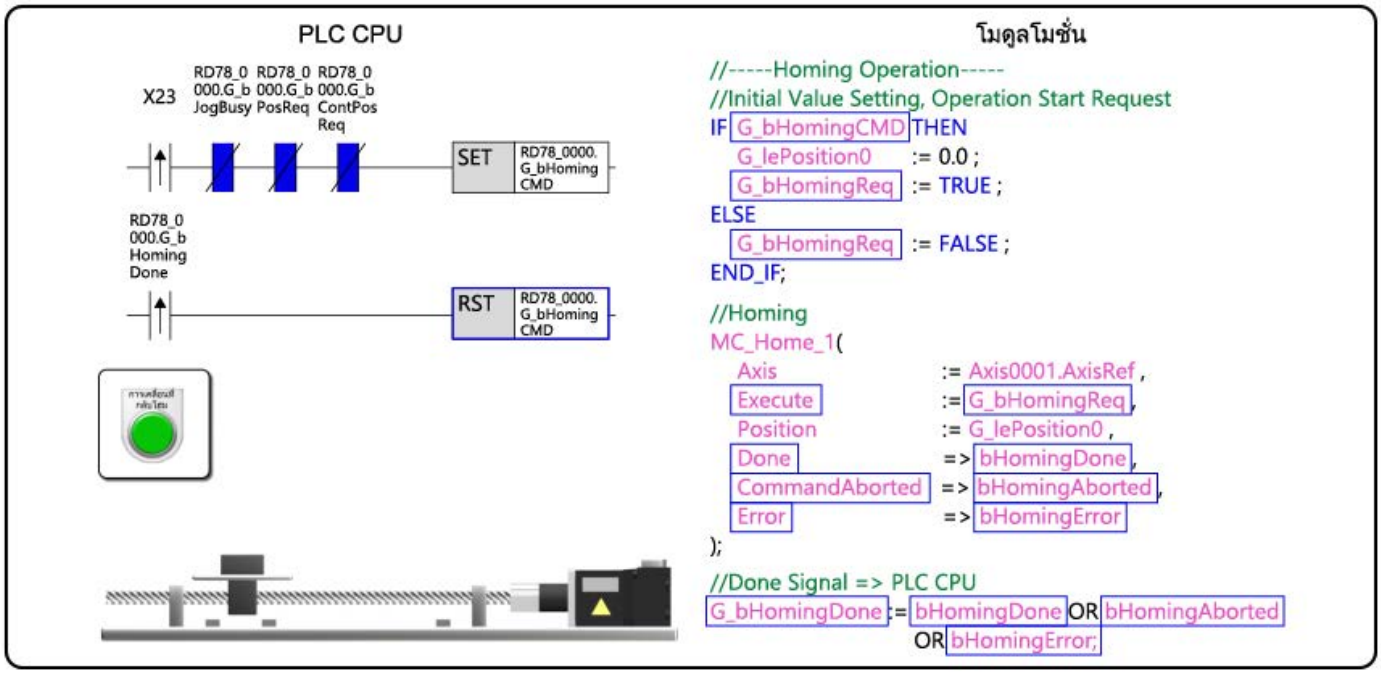
เปิดคำสั่งการเคลื่อนที่กลับโฮม (X23) เพื่อเริ่มทำโฮม
ดำเนินการเคลื่อนที่กลับโฮมด้วยวิธี proximity dog (ลบ 33 ออกจาก Pr.PT45)
แกนจะหยุดเลย dog ไปเล็กน้อย และจะตั้งค่าจุดนั้นเป็นตำแหน่งโฮม

2.5 การตรวจสอบการดำเนินการ



ตรวจสอบโดยการมอนิเตอร์โปรแกรม
เมื่อเปิด X23 "RD78_0000.G_bHomingCMD" จะถูกตั้งค่า
"G_bHomingCMD" ที่ด้านโมดูลโมชั่นจะเปิด และ
"G_bHomingReq" ซึ่งเป็นคำสั่งการดำเนินการของ MC_Home_1 จะเปิด

2.5 การตรวจสอบการดำเนินการ

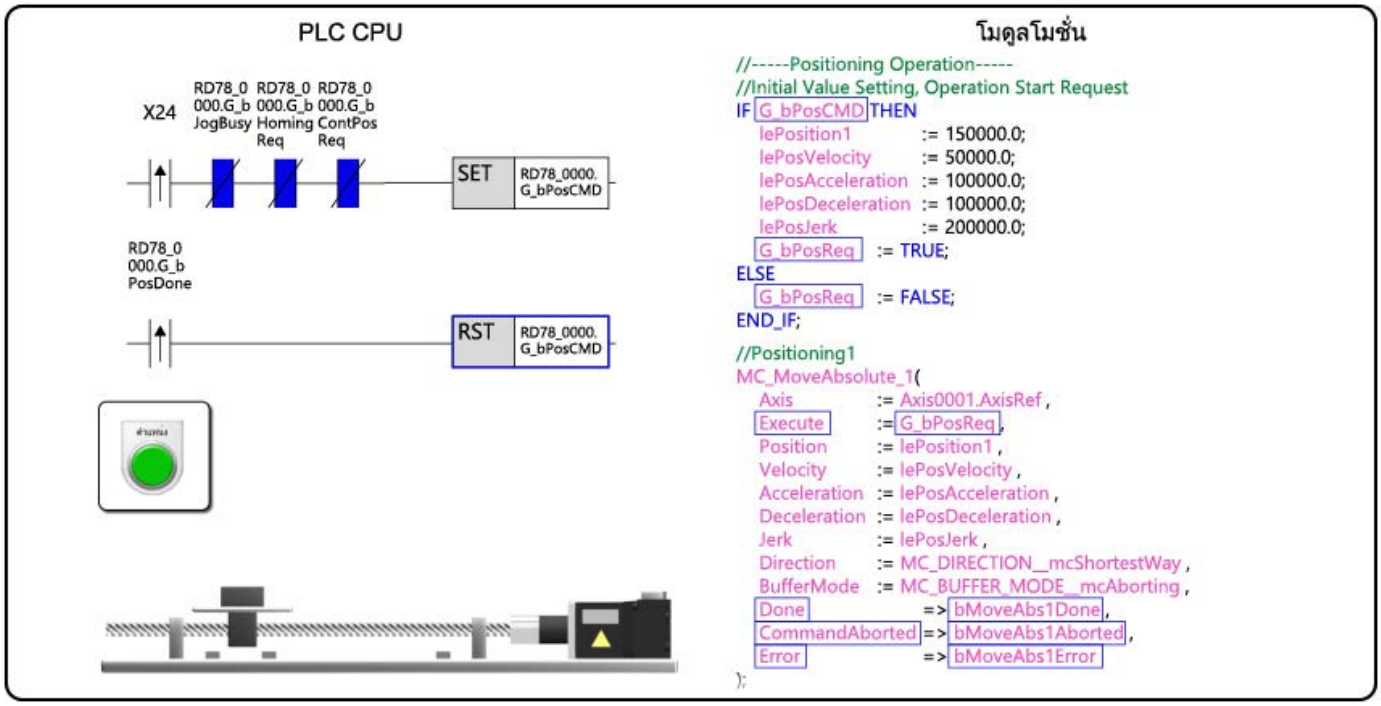


เมื่อการเคลื่อนที่กลับโฮมเสร็จแล้ว เอาต์พุต Done และ "G_bHomingDone" จะเปิด "G_bHomingCMD" บนด้าน CPU ของ PLC จะถูกรีเซ็ตเป็นสถานะเริ่มต้น

2.5 การตรวจสอบการดำเนินการ

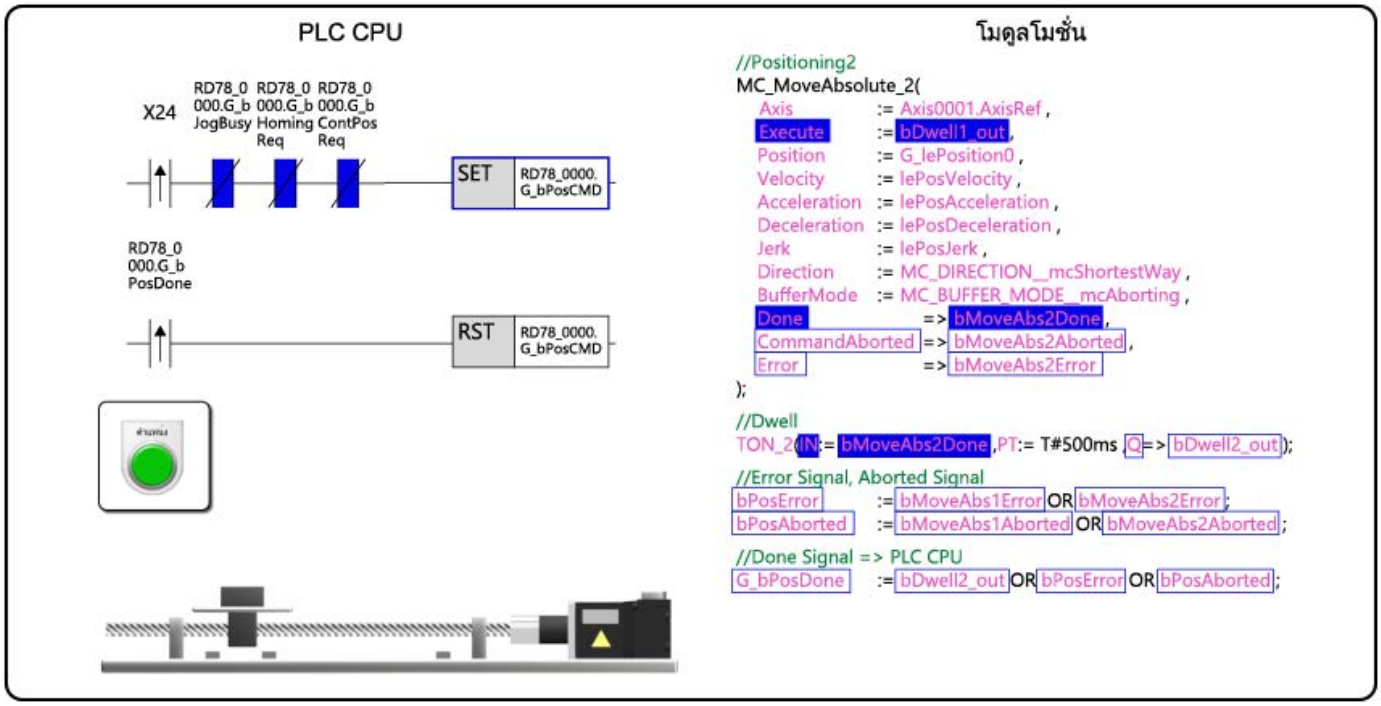
การเปิดการเริ่มตำแหน่ง (X24) จะเริ่มการเคลื่อนที่ขึ้นลง
แกนจะเคลื่อนที่ไปข้างหน้า 150 มม. และหยุดเป็นเวลา
0.5 วินาที และเคลื่อนที่กลับไป 150 มม. และหยุดเป็นเวลา 0.5 วินาที

2.5 การตรวจสอบการดำเนินการ



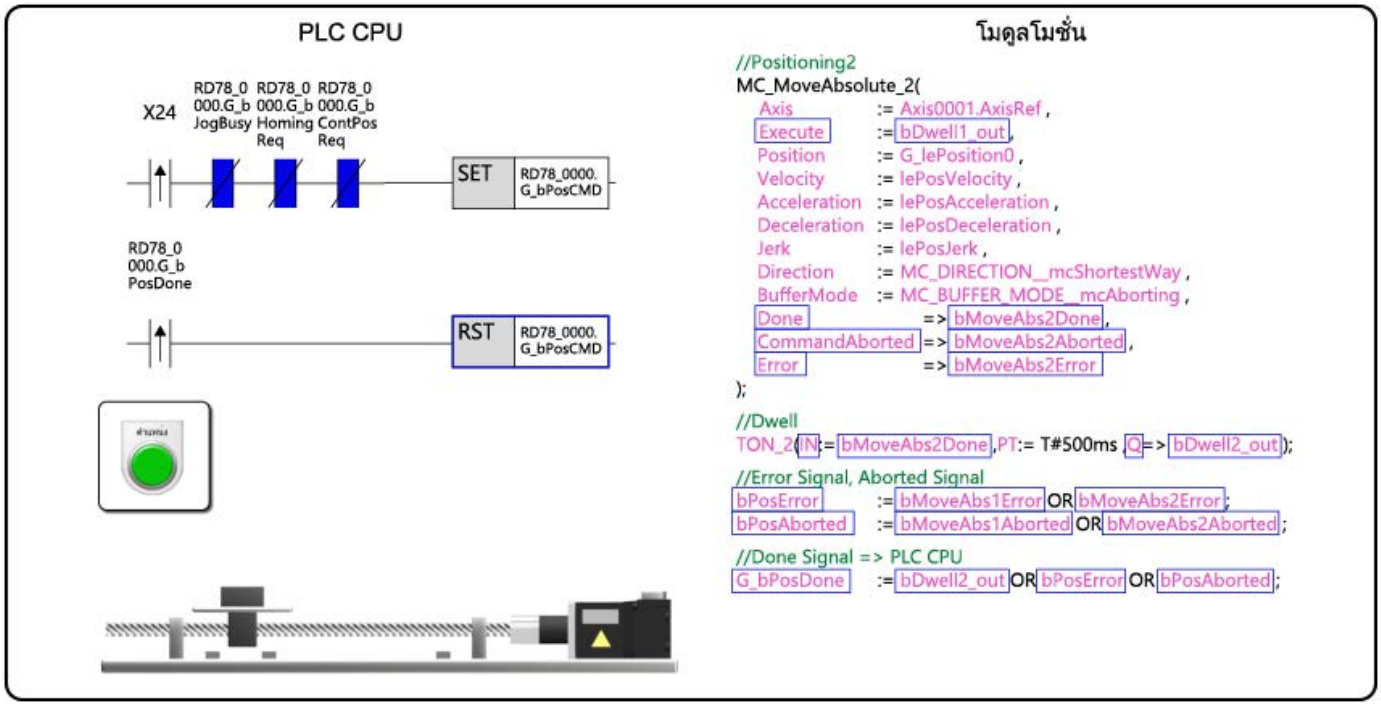
ตรวจสอบโดยการมอนิเตอร์โปรแกรม
เมื่อเปิด X24 "RD78_0000.G_bPosCMD" จะถูกตั้งค่า
"G_bPosCMD" ที่ด้านโมดูลโมชั่นจะเปิด และ "G_bPosReq"
ซึ่งเป็นคำสั่งการดำเนินการของ MC_MoveAbsolute_1 จะเปิด

2.5 การตรวจสอบการดำเนินการ



หลังจากการเคลื่อนไหวขึ้นลงเสร็จแล้วและหมดเวลาการอยู่กับที่
"G_bPosDone" จะเปิด
"G_bPosCMD" บนด้าน CPU ของ PLC จะถูกรีเซ็ตเป็นสถานะเริ่มต้น

2.5 การตรวจสอบการดำเนินการ



เป็นอันเสร็จสิ้นการตรวจสอบการดำเนินการ
ไปที่หน้าถัดไป

2.6 บทสรุปของบทนี้

ในบทนี้คุณได้เรียนรู้เกี่ยวกับ:

- พับบลิคเลเบิล คืออะไร
- การตั้งค่า พับบลิคเลเบิล
- ตัวอย่างโปรแกรม
- การเขียนโปรแกรม
- การตรวจสอบการดำเนินการ

ประเด็นสำคัญ

พับบลิคเลเบิล คืออะไร	• พับบลิคเลเบิล คือเลเบิลที่ใช้ร่วมกันซึ่งสามารถใช้ได้ทั้งในโมดูลโมชั่นและ PLC CPU
การตั้งค่า พับบลิคเลเบิล	• รีจิสเตอร์ พับบลิคเลเบิลจากโกลบอล เลเบิล ของโมดูลโมชั่น • เลือกว่าจะอ่านหรือเขียนแต่ละเลเบิล from/to PLC CPU • หากต้องการตั้งค่าข้อมูลของ structured data ที่เตรียมไว้ในระบบให้เป็นพับบลิคเลเบิล ให้รีจิสเตอร์ พับบลิคเลเบิล ตามชั้นของ structured data • หลังจากตั้งค่าพับบลิคเลเบิล ในโมดูลโมชั่นแล้ว ให้ rebuild all และ reflect พับบลิคเลเบิล • พับบลิคเลเบิล จะถูกรีจิสเตอร์กับโมดูลเลเบิลที่ PLC CPU
ตัวอย่างโปรแกรม	• บทนี้จะอธิบายตัวอย่างโปรแกรมต่อไปนี้: โปรแกรมแลดเดอร์ของ PLC CPU ที่ใช้พับบลิคเลเบิล เพื่อแลกเปลี่ยนสัญญาณเริ่มต้นการกำหนดตำแหน่งและสัญญาณการกำหนดตำแหน่งเสร็จสมบูรณ์
การเขียนโปรแกรม	• เขียนข้อมูลไปยัง PLC CPU ก่อน จากนั้นจึงเขียนข้อมูลไปยังโมดูลโมชั่น
การตรวจสอบการดำเนินการ	• คุณได้ตรวจสอบการดำเนินการของโปรแกรมตัวอย่างในวิดีโอแล้ว

บทที่ 3 โหมดบัฟเฟอร์

โหมดบัฟเฟอร์ดำเนินการอย่างต่อเนื่องโดยเริ่มการดำเนินการ FB หลายครั้ง ของ Motion control FB
สามารถตั้งค่ากับอินพุต BufferMode ของ Motion control FB ได้
สามารถเริ่ม FB ได้สูงสุดสองครั้งพร้อมกันสำหรับแต่ละแกนและกลุ่มแกน

(ตัวอย่าง) MC_MoveAbsolute

```
MC_MoveAbsolute_1(
    Axis              := Axis0001.AxisRef ,
    Execute           := G_bPositioningReq ,
    ContinuousUpdate:= FALSE ,
    Position          := lePosition1 ,
    Velocity          := lePosVelocity ,
    Acceleration      := lePosAcceleration ,
    Deceleration      := lePosDeceleration ,
    Jerk              := lePosJerk ,
    Direction         := MC_DIRECTION__mcShortestWay ,
    BufferMode        := MC_BUFFER_MODE__mcAborting ,
    Options           := 0 ,//mcAccDec
    Done              => bMoveAbs1Done ,
    CommandAborted    => bMoveAbs1Aborted ,
    Error             => bMoveAbs1Error
);
```

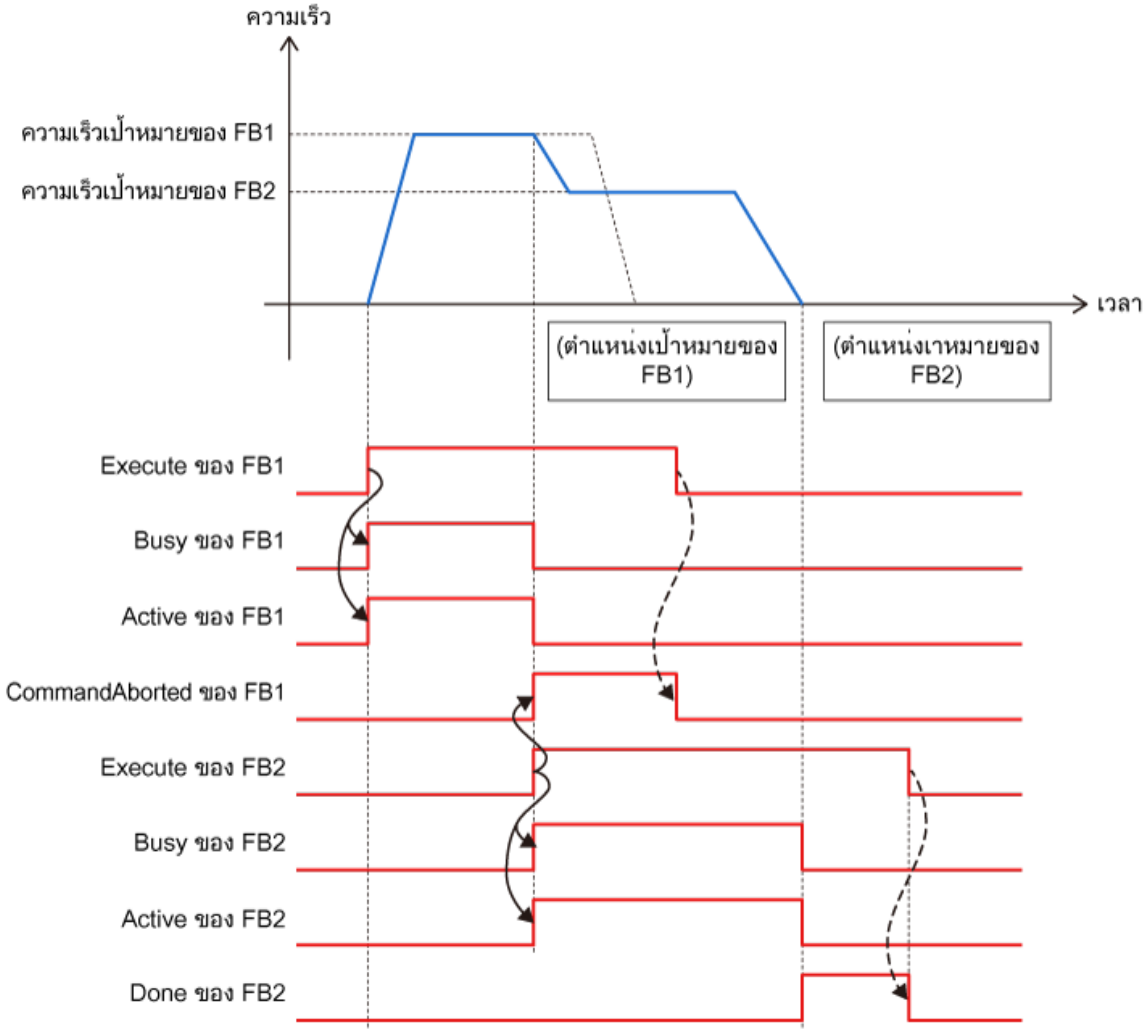
0 หรือ MC_BUFFER_MODE _ mcAborting ··· FB ที่กำลังถูกดำเนินการอยู่จะถูกขัดจังหวะ และ FB ถัดไปจะถูกดำเนินการทันที
1 หรือ MC_BUFFER_MODE__mcBuffered ··· เมื่อการดำเนินการของ FB ที่กำลังถูกดำเนินการอยู่เสร็จแล้ว FB ถัดไปจะถูกดำเนินการ
2 หรือ MC_BUFFER_MODE__mcBlendingLow ··· ถ้าค่าความเร็วเป้าหมายต่ำกว่าค่าที่กำลังทำงานอยู่ที่ FB จะถูกตั้งค่าเป็นค่าความเร็วสำรองไว้ในบัฟเฟอร์เพื่อไว้ใช้ในการเปลี่ยนความเร็ว
3 หรือ MC_BUFFER_MODE__mcBlendingPrevious ··· ค่าความเร็วเป้าหมายที่กำลังทำงานอยู่ที่ FB จะถูกเปลี่ยนความเร็ว
4 หรือ MC_BUFFER_MODE__mcBlendingNext ··· ค่าความเร็วเป้าหมายของ FB จะถูกตั้งค่าไว้ในบัฟเฟอร์เพื่อไว้ใช้ในการเปลี่ยนความเร็ว
5 หรือ MC_BUFFER_MODE__mcBlendingHigh ··· ถ้าค่าความเร็วเป้าหมายสูงกว่าค่าที่กำลังทำงานอยู่ที่ FB จะถูกตั้งค่าเป็นค่าความเร็วสำรองไว้ในบัฟเฟอร์เพื่อไว้ใช้ในการเปลี่ยนความเร็ว

[จุดสำคัญ]
สำหรับอินพุต Direction และ BufferMode ระบุหมายเลข หรือตัวแจงนับ ENUM เริ่มต้นด้วย MC_BUFFER_MODE และ MC_DIRECTION
สำหรับรายละเอียดของตัวแจงนับ ENUM ตัว ดูคู่มือต่อไปนี้

MELSEC iQ-R Programming Manual (Motion Control Function Blocks)
2 VARIABLES AND MOTION CONTROL FB
2.2 List of Enumerators

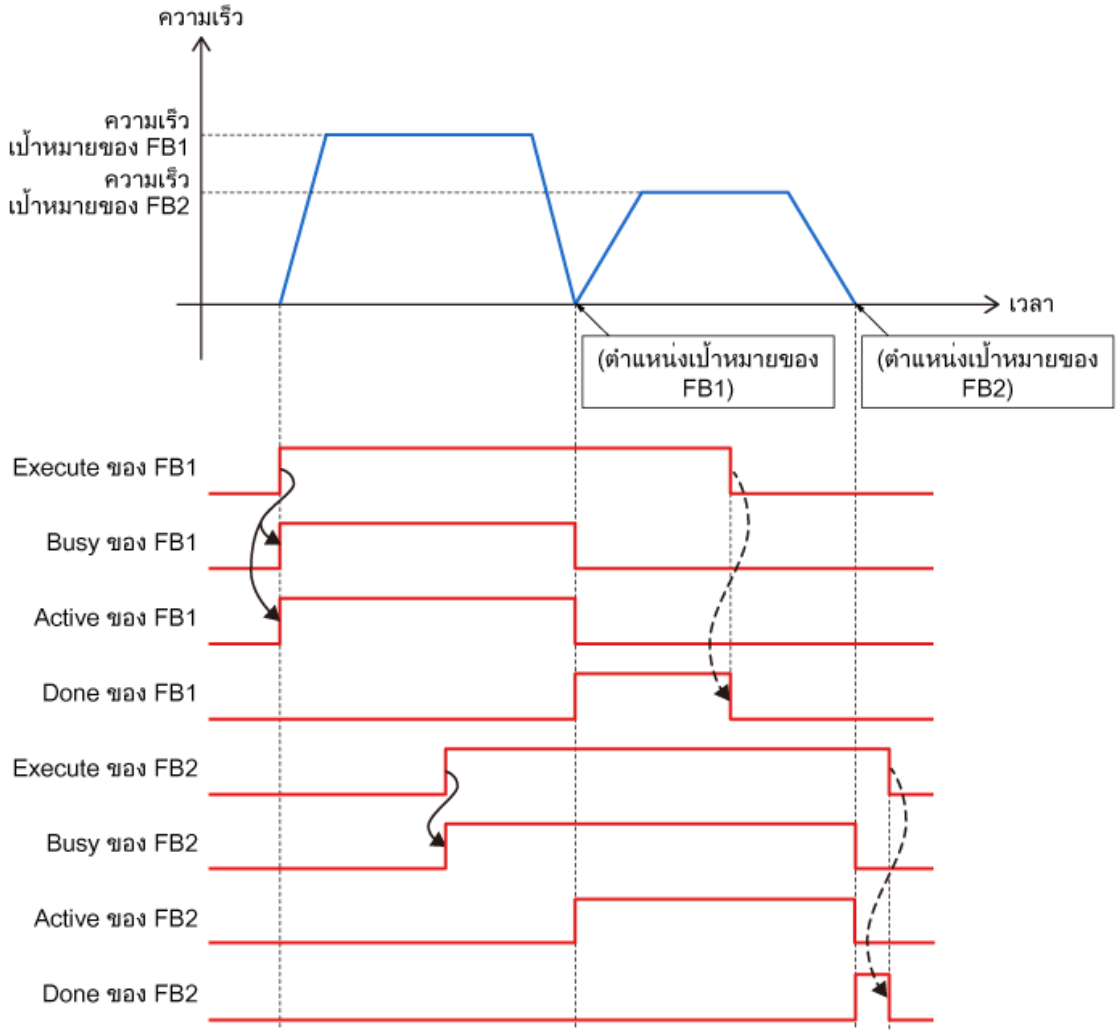
3.1 การยกเลิก

แผนภาพต่อไปนี้แสดงการดำเนินการเมื่อ BufferMode ถูกตั้งค่าเป็น 0:
mcAborting FB ที่กำลังถูกดำเนินการอยู่จะถูกขัดจังหวะ และ FB ถัดไปจะถูกดำเนินการทันที



3.2 ทำบัปเฟอร์แล้ว

แผนภาพต่อไปนี้แสดงการดำเนินการเมื่อ BufferMode ถูกตั้งค่าเป็น 1:mcBuffered
เมื่อการดำเนินการของ FB ที่กำลังถูกดำเนินการอยู่เสร็จแล้ว FB ถัดไปจะถูกดำเนินการ

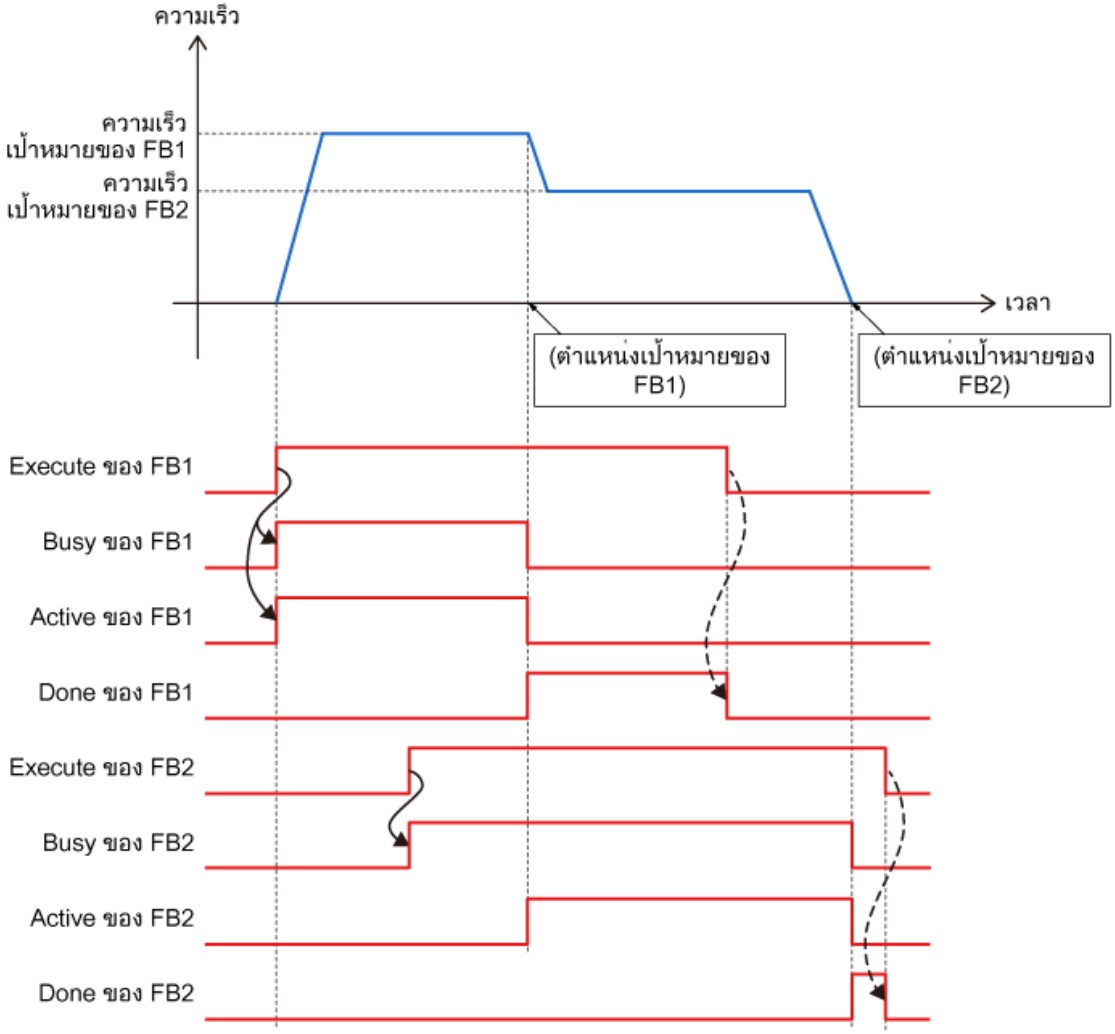


3.3 การผสม

เมื่อ BufferMode ถูกตั้งค่าเป็น mcBlending*** FB ถัดไปจะถูกดำเนินการอย่างต่อเนื่องหลังจากถึงตำแหน่งเป้าหมายของ FB ที่กำลังดำเนินการแล้ว
ในคำอธิบายต่อไปนี้ FB ที่จะถูกดำเนินการก่อน คือ FB1 และ FB ที่จะถูกบัฟเฟอร์คือ FB2

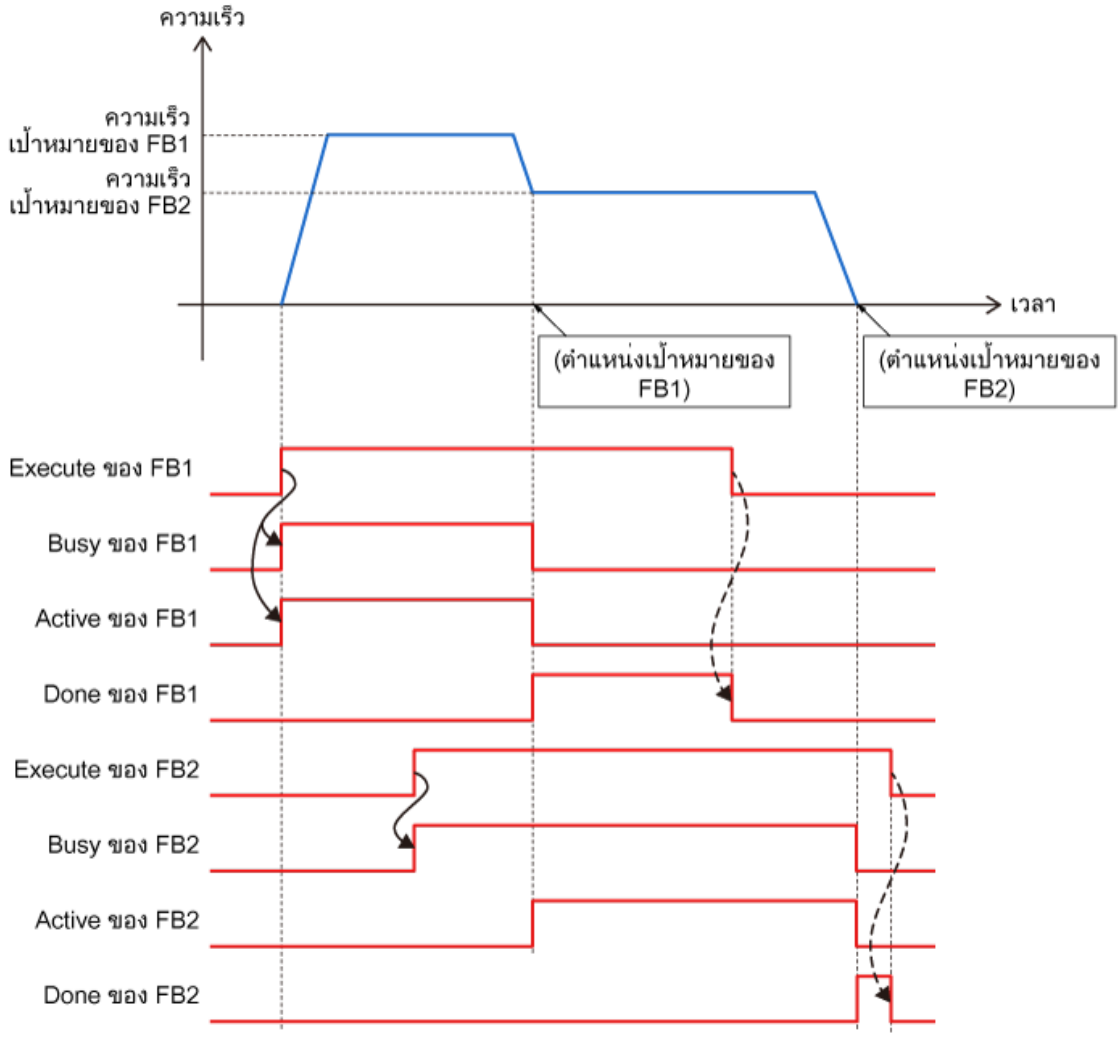
(1) BlendingPrevious

แผนภาพต่อไปนี้แสดงการดำเนินการเมื่อ BufferMode ถูกตั้งค่าเป็น 3: mcBlendingPrevious
การดำเนินการจะถูกดำเนินการที่ความเร็วเป้าหมายของ FB1 จนถึงตำแหน่งเป้าหมายของ FB1
เมื่อสลับการดำเนินการไปที่ FB2 ความเร็วจะเปลี่ยนเป็นความเร็วเป้าหมายของ FB2 และย้ายไปยังตำแหน่งเป้าหมายของ FB2



(2) BlendingNext

แผนภาพต่อไปนี้แสดงการดำเนินการเมื่อ BufferMode ถูกตั้งค่าเป็น 4: mcBlendingNext
ความเร็วจะถูกเปลี่ยนเป็นความเร็วเป้าหมายของ FB2 เมื่อดำเนินการถึงตำแหน่งเป้าหมายของ FB1



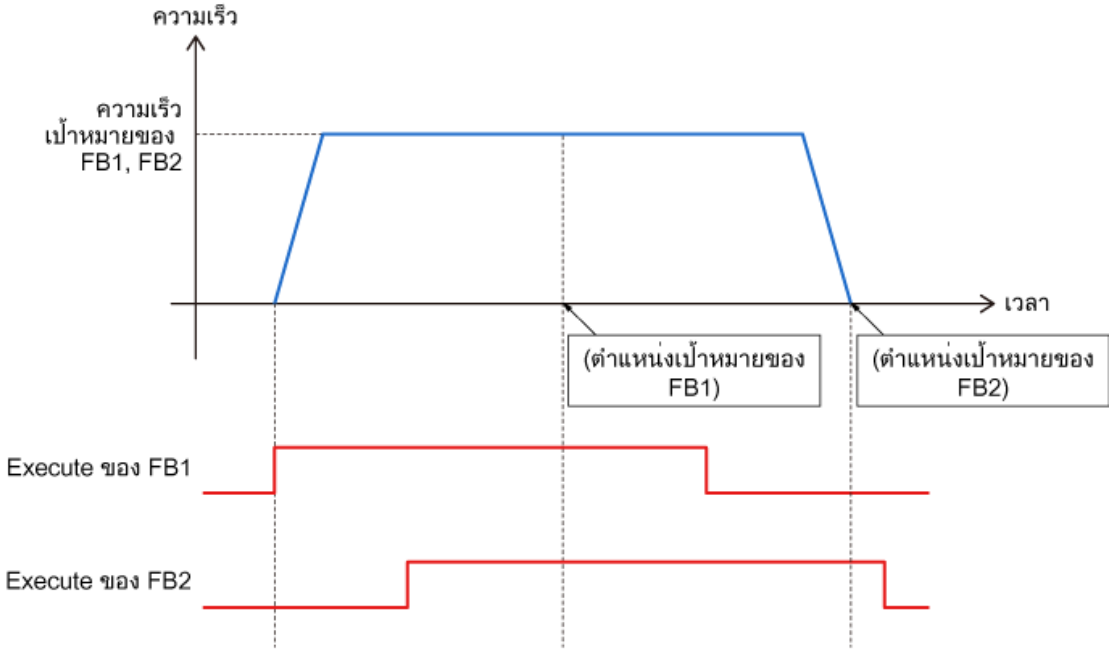
(3) BlendingLow, BlendingHigh

การดำเนินการเมื่อ BufferMode ถูกตั้งค่าเป็น 2: mcBlendingLow หรือ 5: mcBlendingHigh จะแตกต่างกันขึ้นอยู่กับว่าความเร็วเป้าหมายของ FB1 และ FB2 ความเร็วใดมากกว่า

การตั้งค่า	ความเร็วเป้าหมายของ FB1 > ความเร็วเป้าหมายของ FB2	ความเร็วเป้าหมายของ FB1 < ความเร็วเป้าหมายของ MFB2
2: mcBlendingLow	การดำเนินการเดียวกันกับ BlendingPrevious	การดำเนินการเดียวกันกับ BlendingNext
5: mcBlendingHigh	การดำเนินการเดียวกันกับ BlendingNext	การดำเนินการเดียวกันกับ BlendingPrevious

[จุดสำคัญ]

แผนภาพต่อไปนี้แสดงรูปคลื่นความเร็วสำหรับBlendingPrevious BlendingNext BlendingHigh และ BlendingLow เมื่อความเร็วเป้าหมายของ FB1 และ FB2 เท่ากัน



3.4 ตัวอย่างโปรแกรม

(1) ตัวอย่างการดำเนินการของโปรแกรม
บทนี้ใช้โปรแกรมตัวอย่างที่ใช้ในบทที่ 2
ตรวจสอบความแตกต่างในการดำเนินการของโหมดบัฟเฟอร์ในโปรแกรมที่ขึ้นต้นด้วย X25

รายการ	FB1 (MC_MoveAbsolute)	FB2 (MC_MoveAbsolute)
หมายเลขตำแหน่ง	75000.0 [μm]	150000.0 [μm]
ความเร็ว	50000.0 [μm/s]	25000.0 [μm/s]
การเร่งความเร็ว การชะลอความเร็ว	100000.0 [μm/s^2]	50000.0 [μm/s^2]
กระตุก	200000.0 [μm/s^3]	100000.0 [μm/s^3]

(2) โปรแกรมของ PLC CPU
MAIN (แลดเดอร์ สแกนโปรแกรม)

ขอบขาขึ้นของการเริ่มต้นการควบคุมตำแหน่ง (X25) จะยังคงอยู่ใน G_bContPosCMD และถูกส่งไปยังโมดูลโมชั่นเป็นเงื่อนไขการเริ่มต้นของการควบคุมการวางตำแหน่งที่ต่อเนื่อง
อินเตอร์ล็อคได้รับการตั้งค่าเพื่อป้องกันไม่ให้เริ่มการควบคุมตำแหน่งในขณะที่โปรแกรมอื่นกำลังทำงานอยู่ เมื่อได้รับแจ้งว่าโมดูลโมชั่นเปิดสัญญาณการเคลื่อนที่ต่อเนื่องกลับโฮม G_bContPosCMD จะถูกรีเซ็ตที่ขอบขาขึ้นขึ้นของสัญญาณนั้น

(3) โปรแกรมโมดูลโมชั่น
การกำหนดตำแหน่งต่อเนื่อง (ประเภทการดำเนินการปกติ)

```
1  //-----Continuous Positioning Operation-----
2  //Initial Value Setting, Operation Start Request
3  IF G_bContPosCMD THEN
4      lePosition1          := 75000.0;
5      lePosVelocity1       := 50000.0;
6      lePosAcceleration1   := 100000.0;
7      lePosDeceleration1   := 100000.0;
8      lePosJerk1           := 200000.0;
9      lePosition2          := 150000.0;
10     lePosVelocity2       := 25000.0;
11     lePosAcceleration2   := 50000.0;
12     lePosDeceleration2   := 50000.0;
13     lePosJerk2           := 100000.0;
14     G_bContPosReq := TRUE;
15 ELSE
16     G_bContPosReq := FALSE;
17 END_IF;
18
   //Positioning1
FB1 MC_MoveAbsolute_1(
21     Axis         := Axis0001.AxisRef ,
22     Execute      := G_bContPosReq ,
23     Position     := lePosition1 ,
24     Velocity     := lePosVelocity1 ,
25     Acceleration:= lePosAcceleration1 ,
26     Deceleration:= lePosDeceleration1 ,
27     Jerk         := lePosJerk1 ,
28     Direction    := MC_DIRECTION_mcShortestWay ,
29     Done            => bMoveAbs1Done ,
30     Active          => bMoveAbs1Active ,
31     CommandAborted  => bMoveAbs1Aborted ,
32     Error           => bMoveAbs1Error
33 );
   //Positioning2
FB2 MC_MoveAbsolute_2(
36     Axis         := Axis0001.AxisRef ,
37     Execute      := bMoveAbs1Active ,
38     Position     := lePosition2 ,
39     Velocity     := lePosVelocity2 ,
40     Acceleration:= lePosAcceleration2 ,
41     Deceleration:= lePosDeceleration2 ,
42     Jerk         := lePosJerk2 ,
43     Direction    := MC_DIRECTION_mcShortestWay ,
44     BufferMode   := MC_BUFFER_MODE_mcBuffered,
45     Done            => bMoveAbs2Done ,
46     CommandAborted  => bMoveAbs2Aborted,
47     Error           => bMoveAbs2Error
48 );
49 //Dwell
50 SET(bMoveAbs2Done,bDwell_In);
51 TON_1(IN:= bDwell_In ,PT:= T#100ms ,Q=> bDwell_out );
52
53 //Error Signal, Aborted Signal
54 bError   := bMoveAbs1Error OR bMoveAbs2Error;
55 bAborted := bMoveAbs1Aborted OR bMoveAbs2Aborted;
56
57 //Done Signal => PLC CPU
58 G_bContPosDone := bDwell_out OR bError OR bAborted;
59 //Reset Dwell_In
60 RST(G_bContPosDone,bDwell_In);
61
```

รับสัญญาณเริ่มต้นการควบคุมตำแหน่งต่อเนื่อง (G_bContPosCMD) จาก PLC CPU
ข้อมูลที่จำเป็นสำหรับตำแหน่งจะถูกจัดเก็บไว้ในเลเบิล และคำขอเริ่มต้นการควบคุมตำแหน่งต่อเนื่อง (G_bContPositioningReq) จะถูกเปิดใช้งาน

ปิด G_bContPositioningReq หลังจากปิด G_bContPosCMD

FB1 (MC_MoveAbsolute_1) จะดำเนินการวางตำแหน่ง

FB2 (MC_MoveAbsolute_2) จะเริ่มต้นทำงานด้วยเอาต์พุต Active ของ FB1(MC_MoveAbsolute_1) จากนั้น FB2 จะถูกบัฟเฟอร์ในขณะที่ FB1 กำลังทำงาน

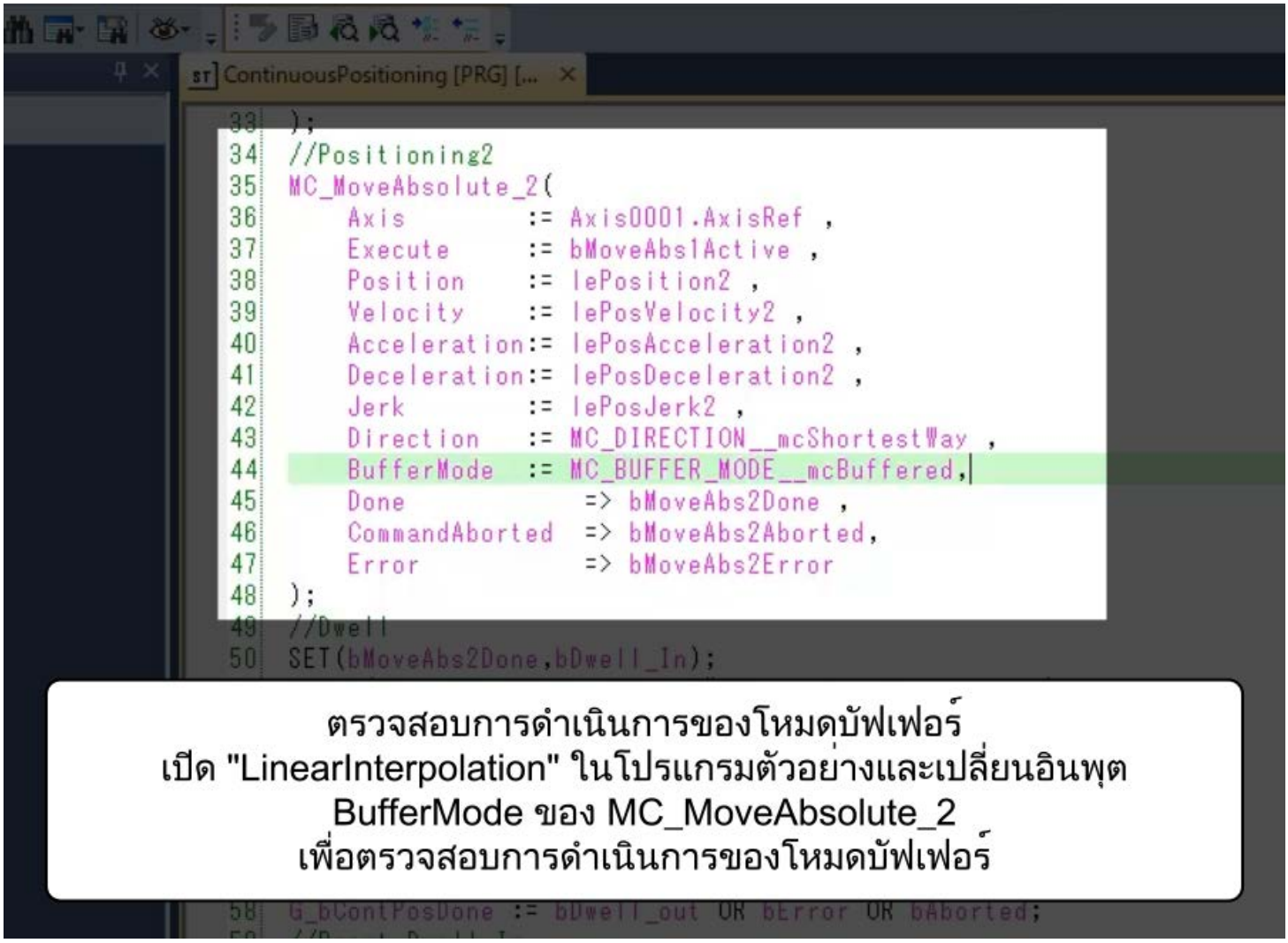
เปลี่ยนอินพุต BufferMode ของ FB2 (MC_MoveAbsolute_2) และตรวจสอบการดำเนินการของโหมดบัฟเฟอร์

การหยุดอยู่กับที่ (dwell time) จะเริ่มต้นโดยเอาต์พุต Done ของ FB2 (MC_MoveAbsolute_2)

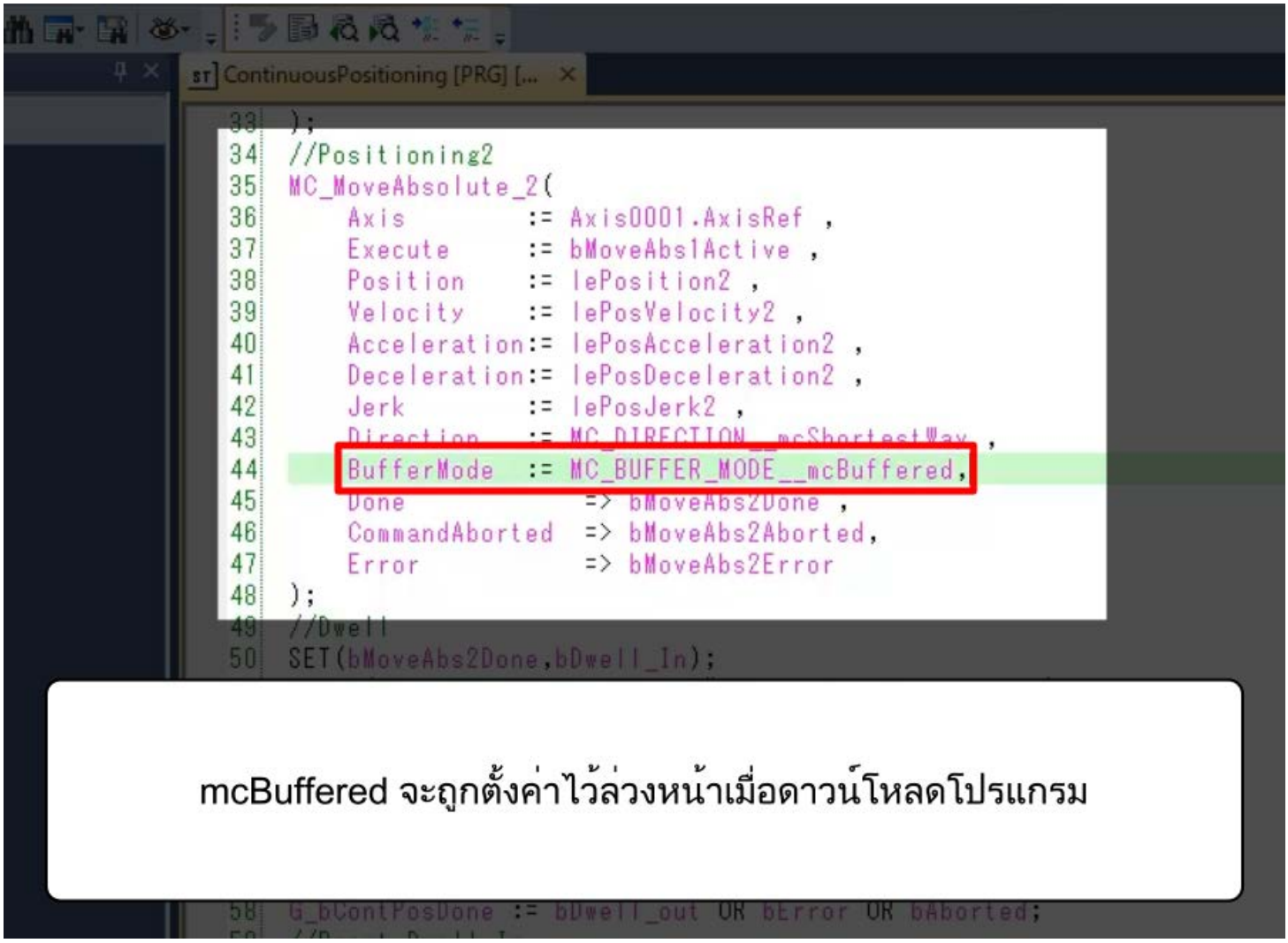
ส่งคืนสัญญาณการดำเนินการแล้วเสร็จไปยัง PLC CPU หลังจากหมดเวลาหยุดอยู่กับที่ (dwell time) หรือเอาต์พุต Error หรือเอาต์พุต CommandAborted ของ MC_MoveAbsolute เปิดอยู่
ในเวลาเดียวกัน อินพุทของการหน่วงเวลาจะถูกรีเซ็ต

คลิกปุ่มเล่นที่ด้านซ้ายล่างของหน้าต่าง

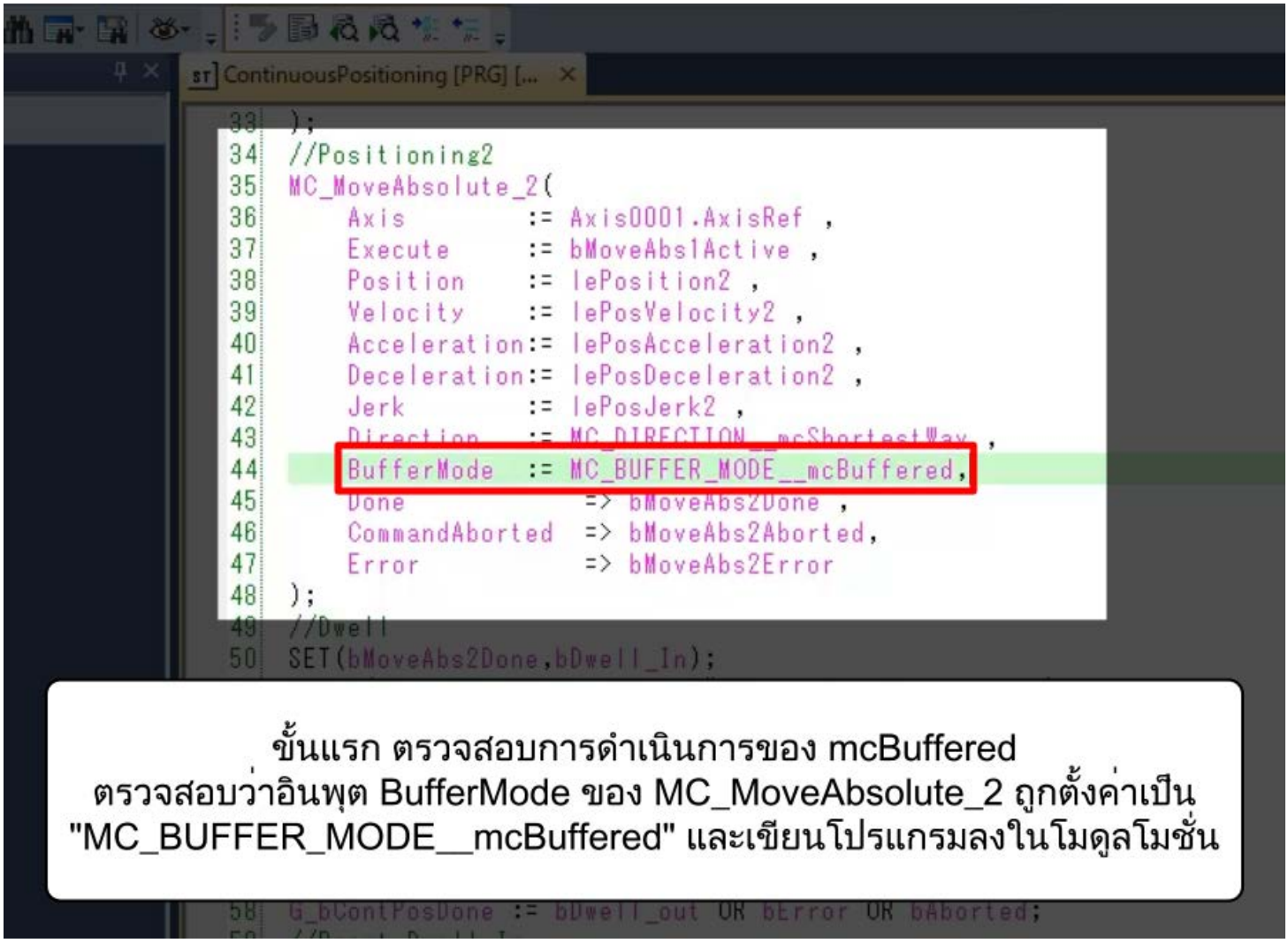
3.5 การตรวจสอบการดำเนินการ



3.5 การตรวจสอบการดำเนินการ

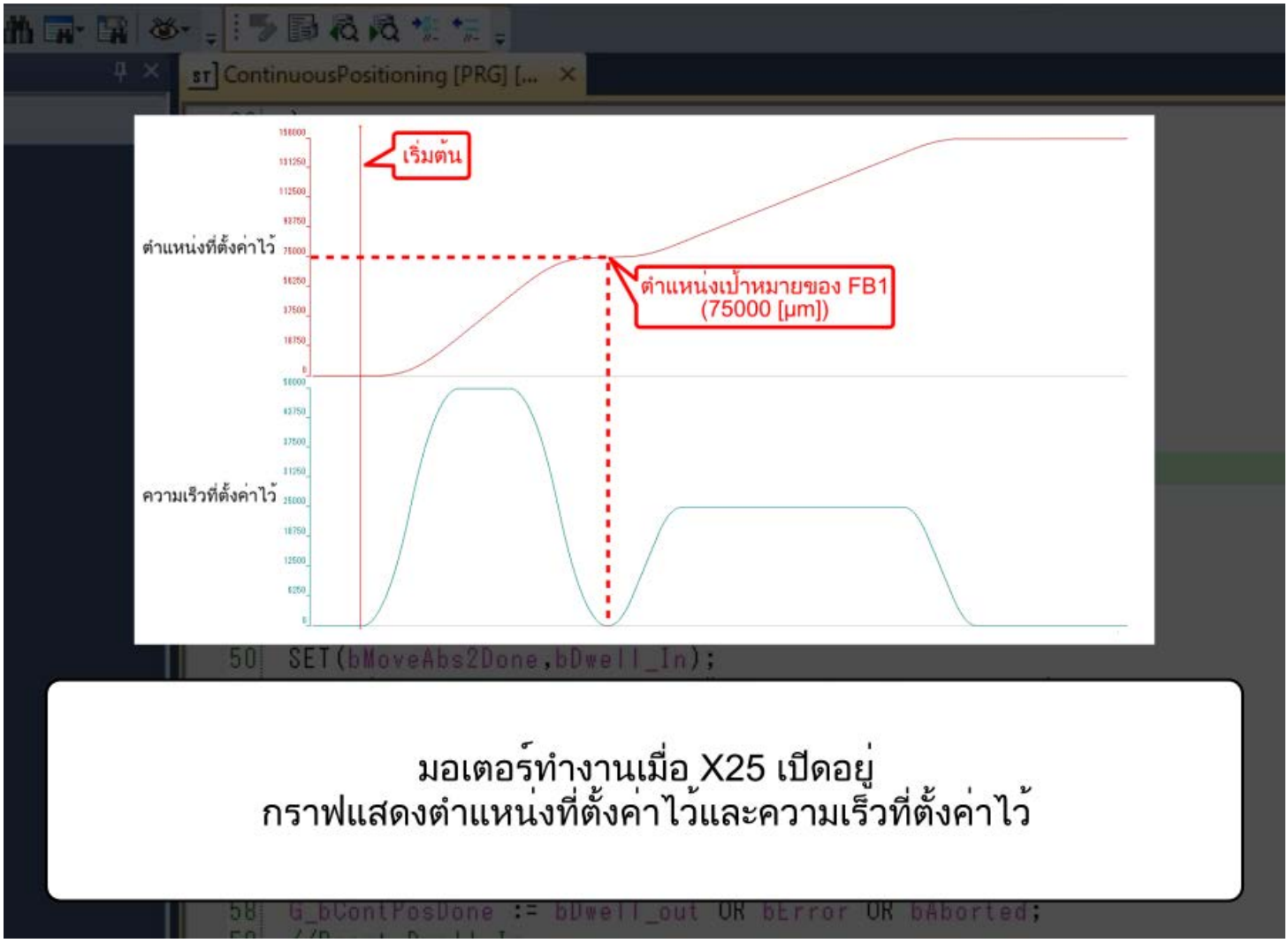


3.5 การตรวจสอบการดำเนินการ



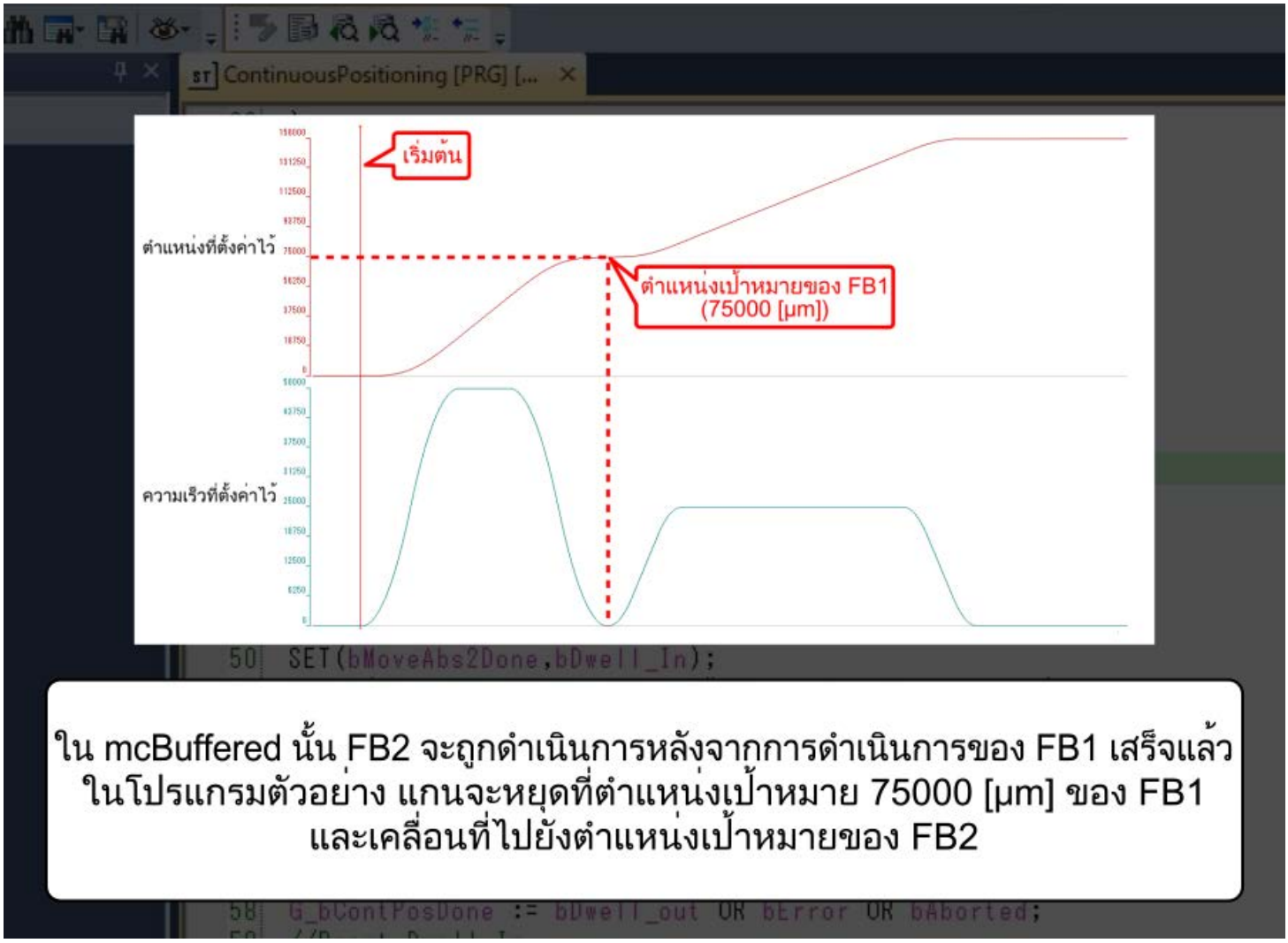
ขั้นแรก ตรวจสอบการดำเนินการของ mcBuffered
ตรวจสอบว่าอินพุต BufferMode ของ MC_MoveAbsolute_2 ถูกตั้งค่าเป็น "MC_BUFFER_MODE__mcBuffered" และเขียนโปรแกรมลงในโมดูลโมชั่น

3.5 การตรวจสอบการดำเนินการ



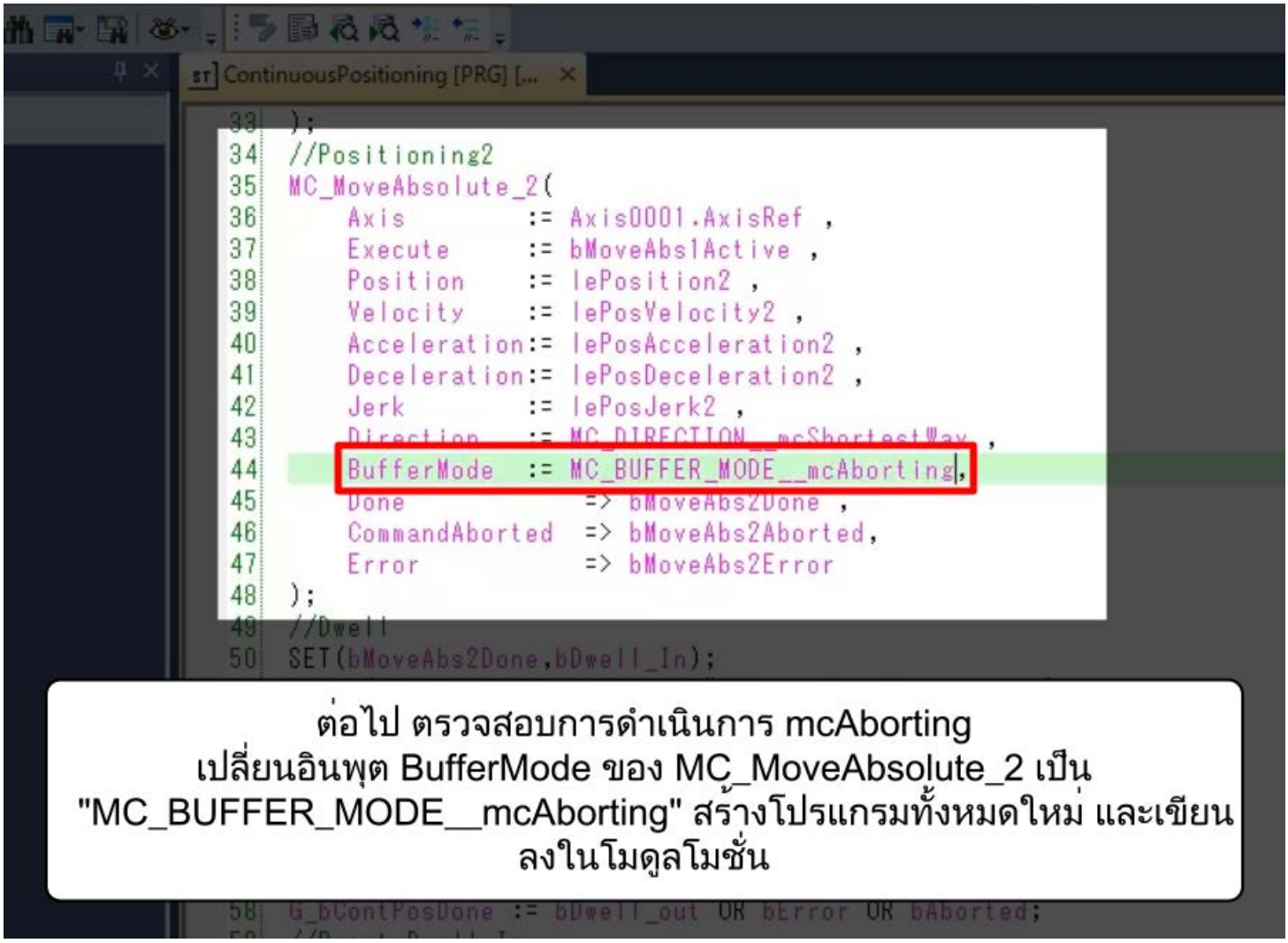
มอเตอร์ทำงานเมื่อ X25 เปิดอยู่
กราฟแสดงตำแหน่งที่ตั้งค่าไว้และความเร็วที่ตั้งค่าไว้

3.5 การตรวจสอบการดำเนินการ



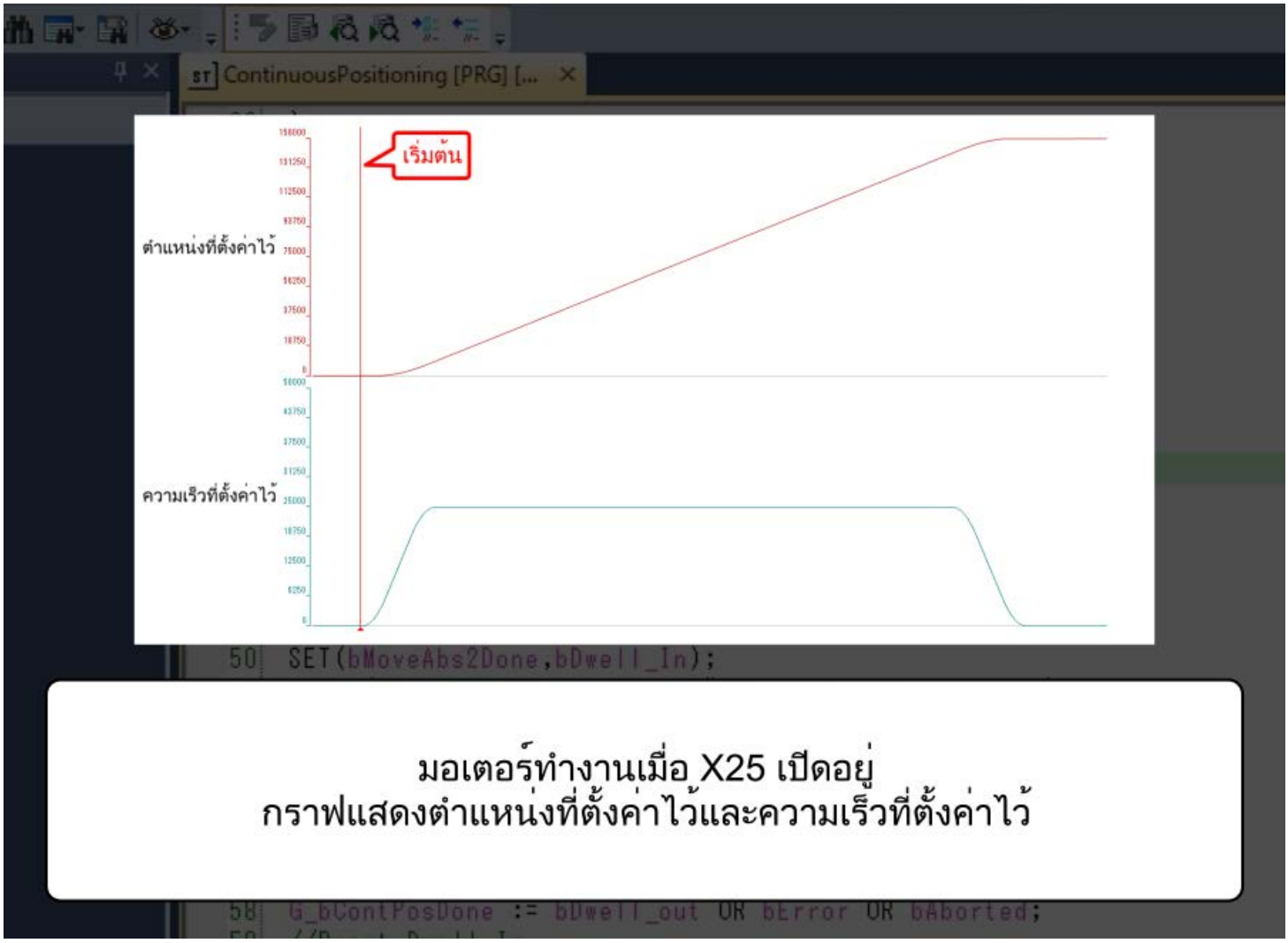
ใน mcBuffered นั้น FB2 จะถูกดำเนินการหลังจากการดำเนินการของ FB1 เสร็จแล้ว ในโปรแกรมตัวอย่าง แกนจะหยุดที่ตำแหน่งเป้าหมาย 75000 [µm] ของ FB1 และเคลื่อนที่ไปยังตำแหน่งเป้าหมายของ FB2

3.5 การตรวจสอบการดำเนินการ



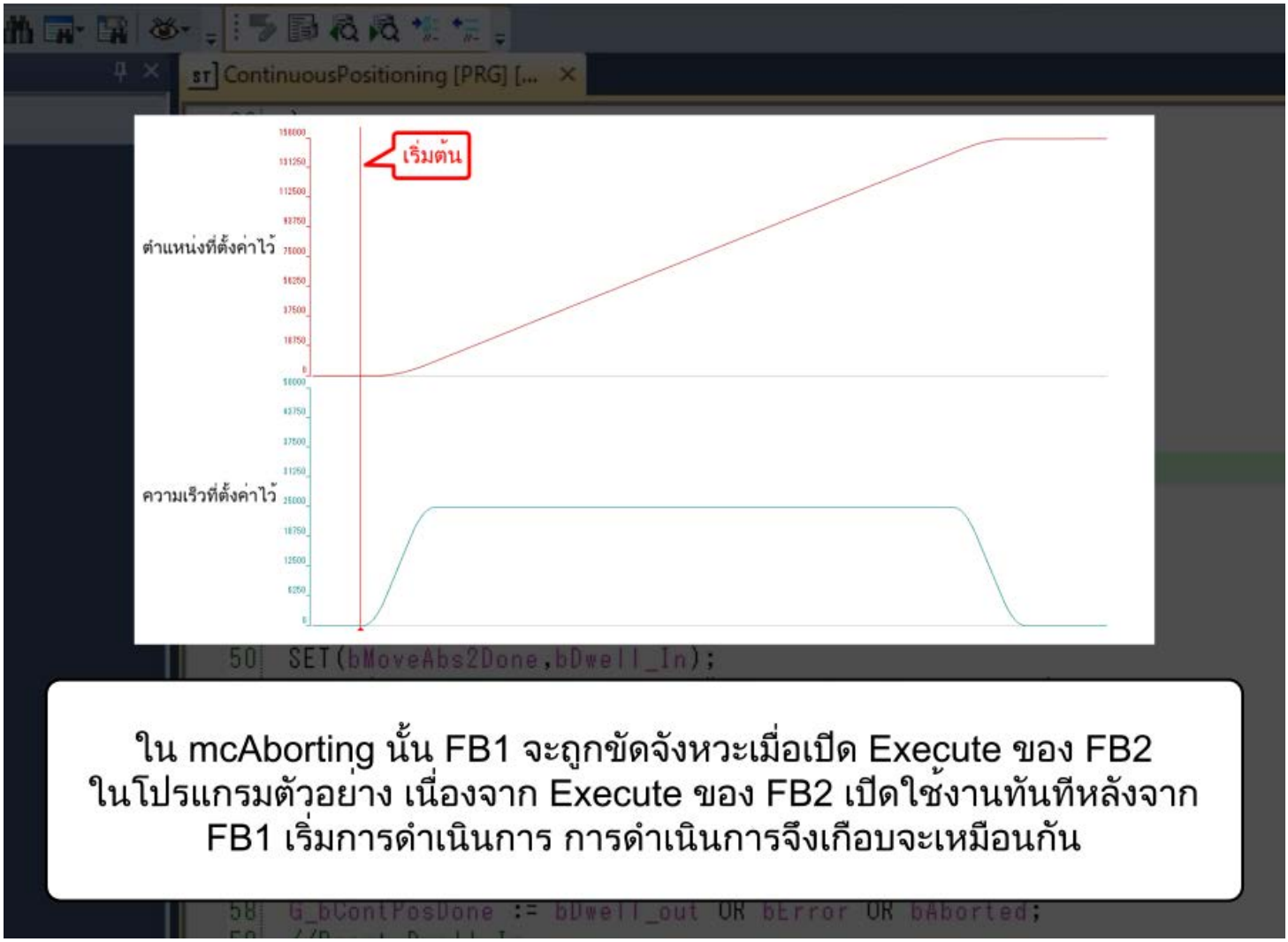
ต่อไป ตรวจสอบการดำเนินการ mcAborting
เปลี่ยนอินพุต BufferMode ของ MC_MoveAbsolute_2 เป็น "MC_BUFFER_MODE__mcAborting" สร้างโปรแกรมทั้งหมดใหม่ และเขียนลงในโมดูลโมชั่น

3.5 การตรวจสอบการดำเนินการ



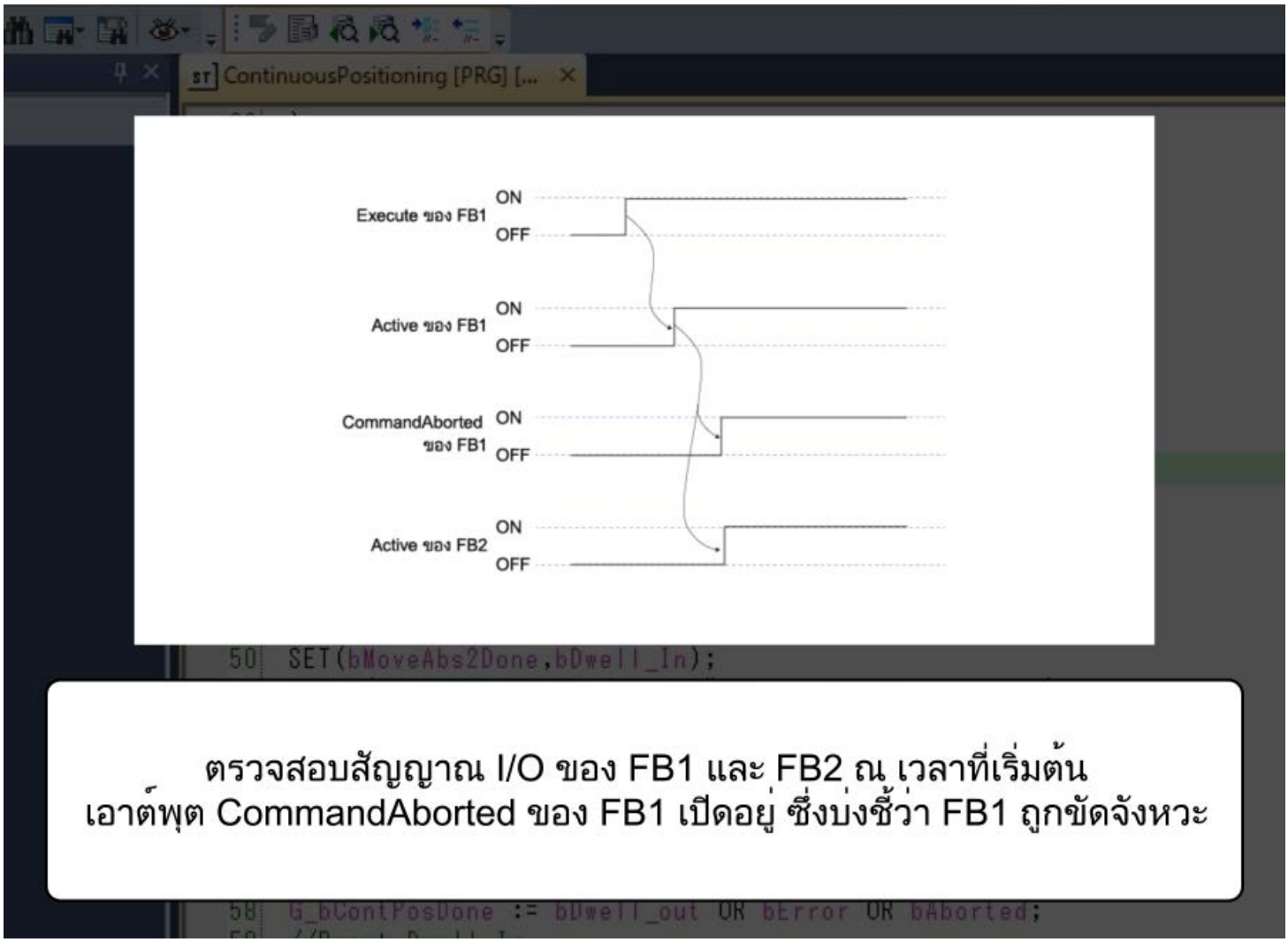
มอเตอร์ทำงานเมื่อ X25 เปิดอยู่
กราฟแสดงตำแหน่งที่ตั้งค่าไว้และความเร็วที่ตั้งค่าไว้

3.5 การตรวจสอบการดำเนินการ



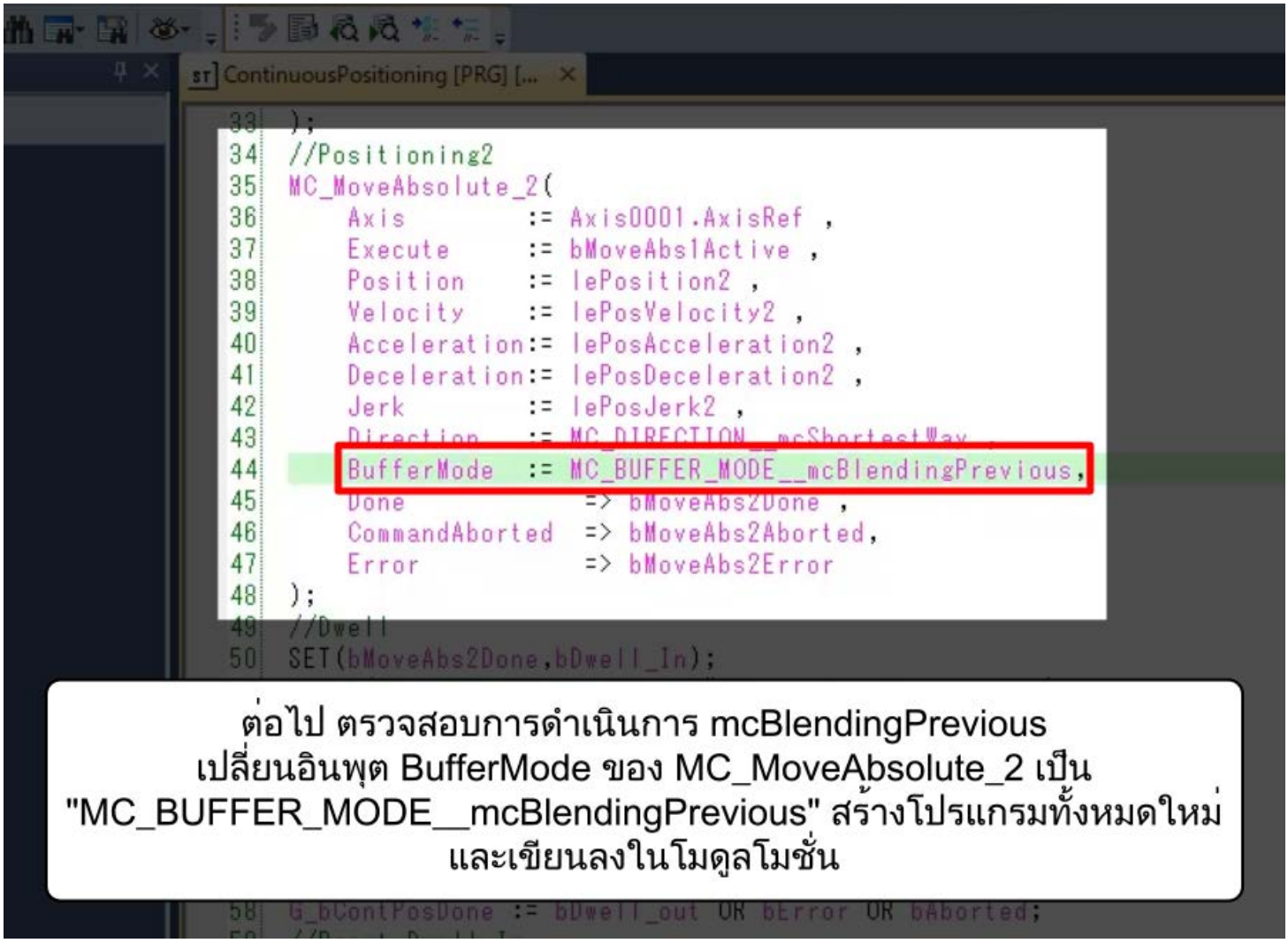
ใน mcAborting นั้น FB1 จะถูกขัดจังหวะเมื่อเปิด Execute ของ FB2 ในโปรแกรมตัวอย่าง เนื่องจาก Execute ของ FB2 เปิดใช้งานทันทีหลังจาก FB1 เริ่มการดำเนินการ การดำเนินการจึงเกือบจะเหมือนกัน

3.5 การตรวจสอบการดำเนินการ



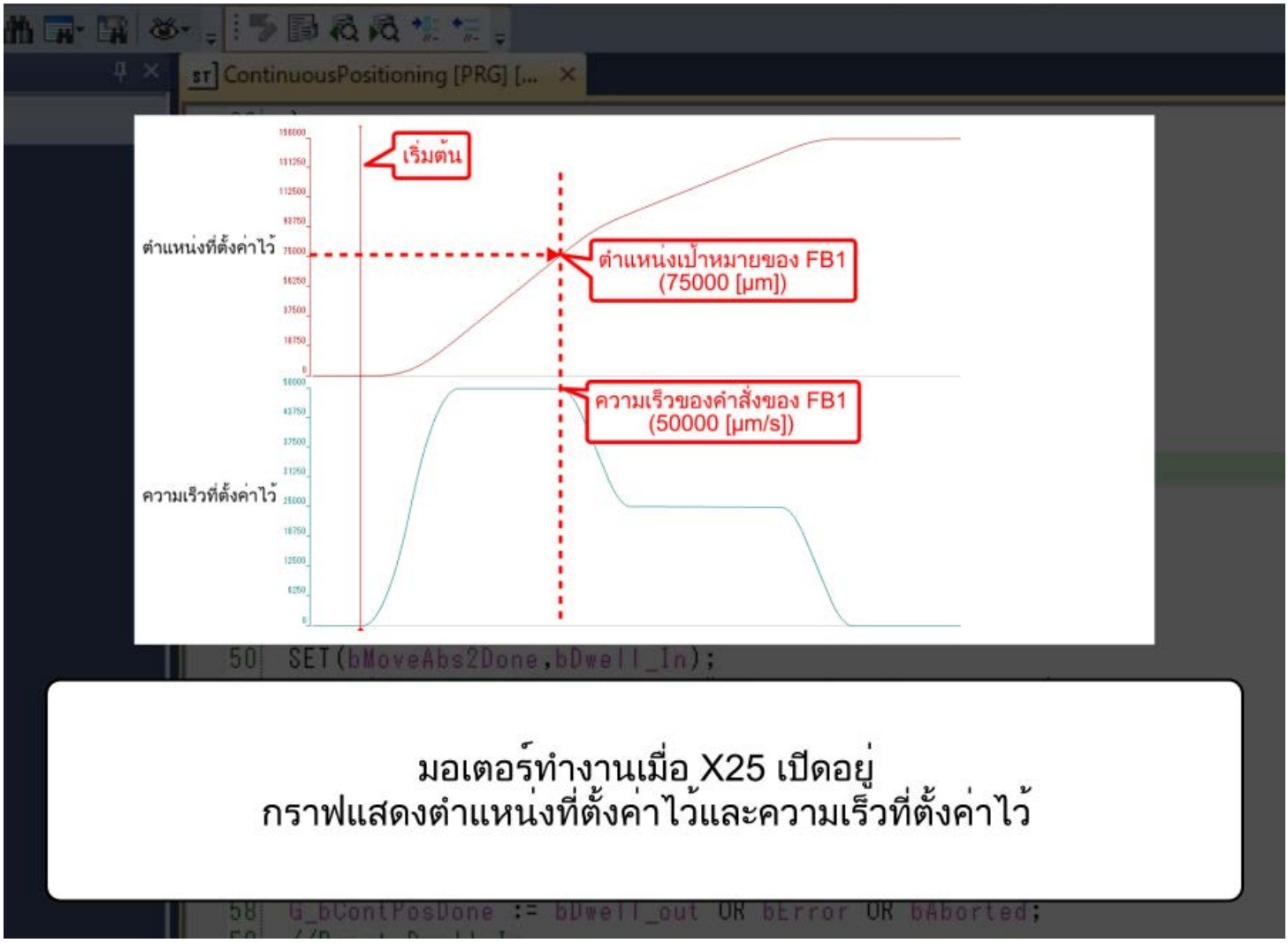
ตรวจสอบสัญญาณ I/O ของ FB1 และ FB2 ณ เวลาที่เริ่มต้น
เอาต์พุต CommandAborted ของ FB1 เปิดอยู่ ซึ่งบ่งชี้ว่า FB1 ถูกขัดจังหวะ

3.5 การตรวจสอบการดำเนินการ



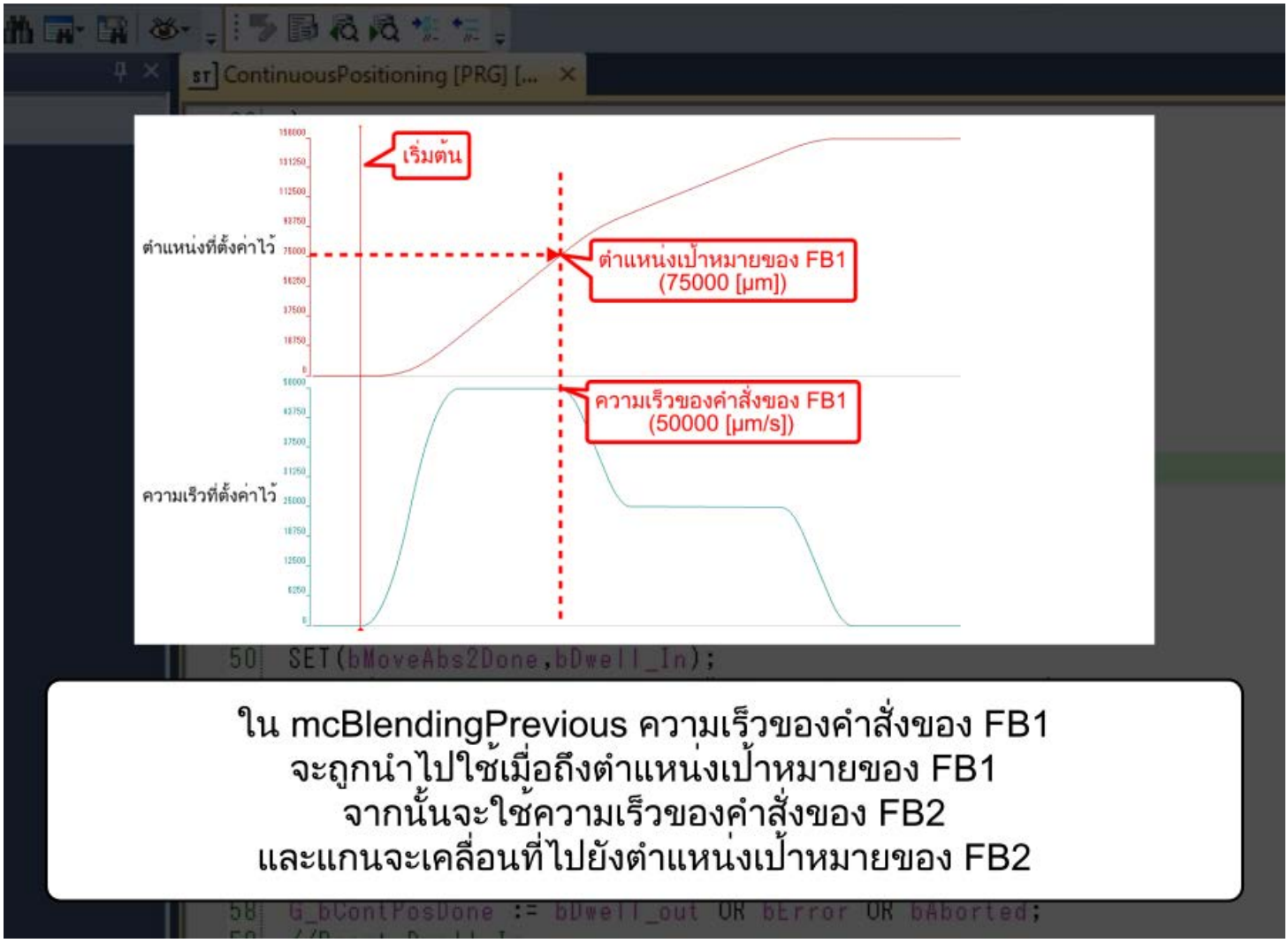
ต่อไป ตรวจสอบการดำเนินการ mcBlendingPrevious
เปลี่ยนอินพุต BufferMode ของ MC_MoveAbsolute_2 เป็น
"MC_BUFFER_MODE__mcBlendingPrevious" สร้างโปรแกรมทั้งหมดใหม่
และเขียนลงในโมดูลโมชั่น

3.5 การตรวจสอบการดำเนินการ



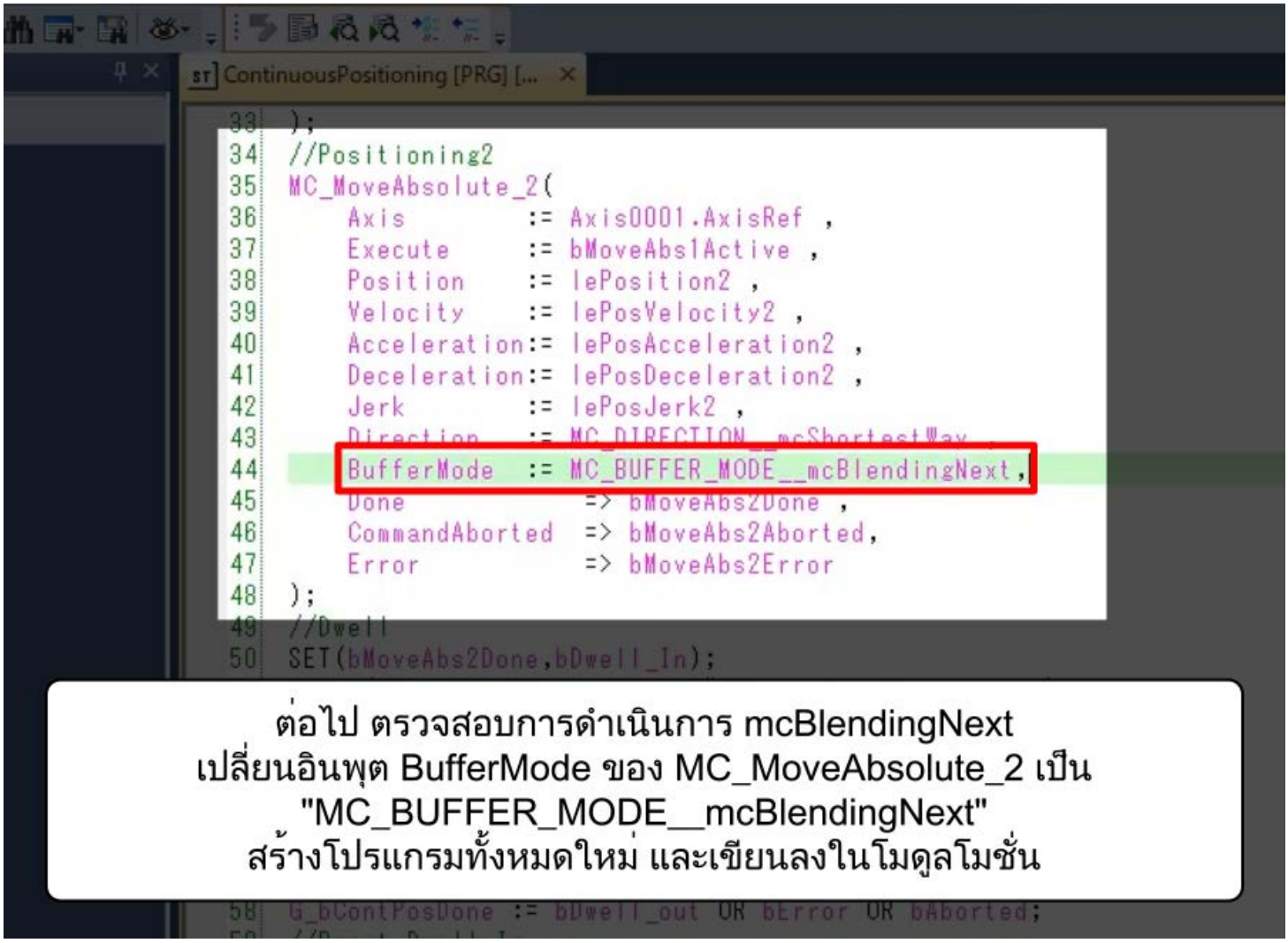
มอเตอร์ทำงานเมื่อ X25 เปิดอยู่
กราฟแสดงตำแหน่งที่ตั้งค่าไว้และความเร็วที่ตั้งค่าไว้

3.5 การตรวจสอบการดำเนินการ



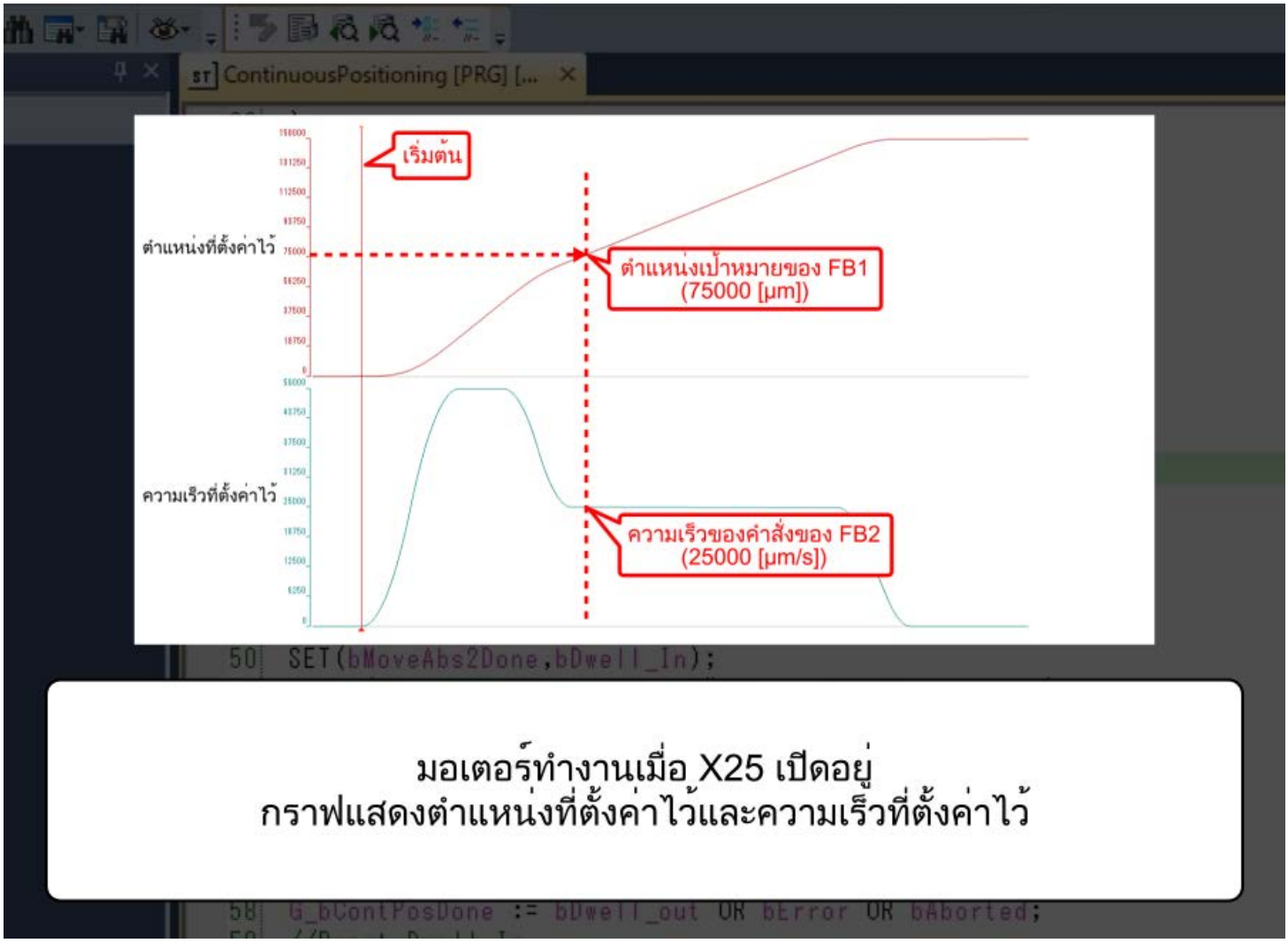
ใน mcBlendingPrevious ความเร็วของคำสั่งของ FB1
จะถูกนำไปใช้เมื่อถึงตำแหน่งเป้าหมายของ FB1
จากนั้นจะใช้ความเร็วของคำสั่งของ FB2
และแกนจะเคลื่อนที่ไปยังตำแหน่งเป้าหมายของ FB2

3.5 การตรวจสอบการดำเนินการ



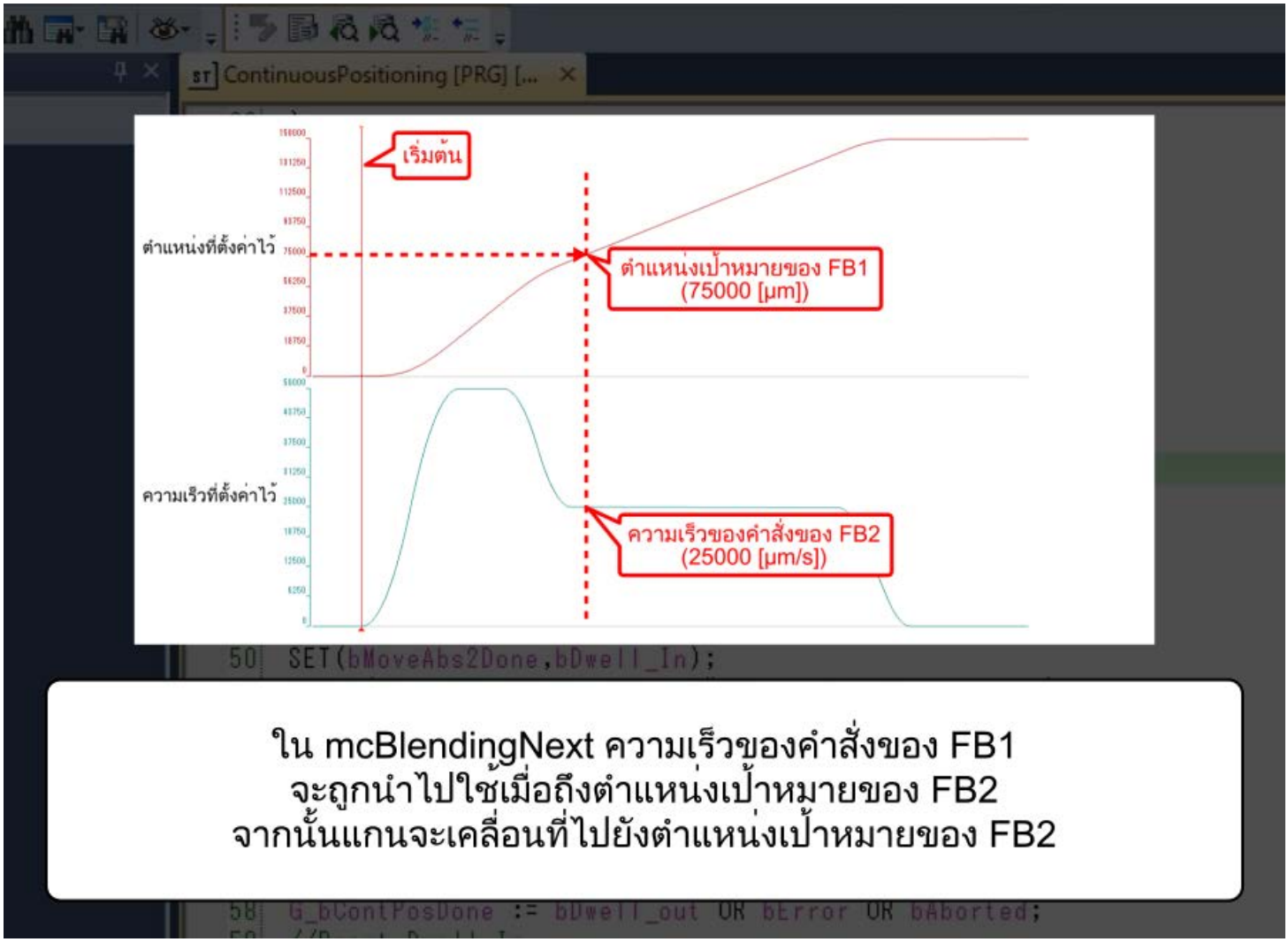
ต่อไป ตรวจสอบการดำเนินการ mcBlendingNext
เปลี่ยนอินพุต BufferMode ของ MC_MoveAbsolute_2 เป็น
"MC_BUFFER_MODE__mcBlendingNext"
สร้างโปรแกรมทั้งหมดใหม่ และเขียนลงในโมดูลโมชั่น

3.5 การตรวจสอบการดำเนินการ



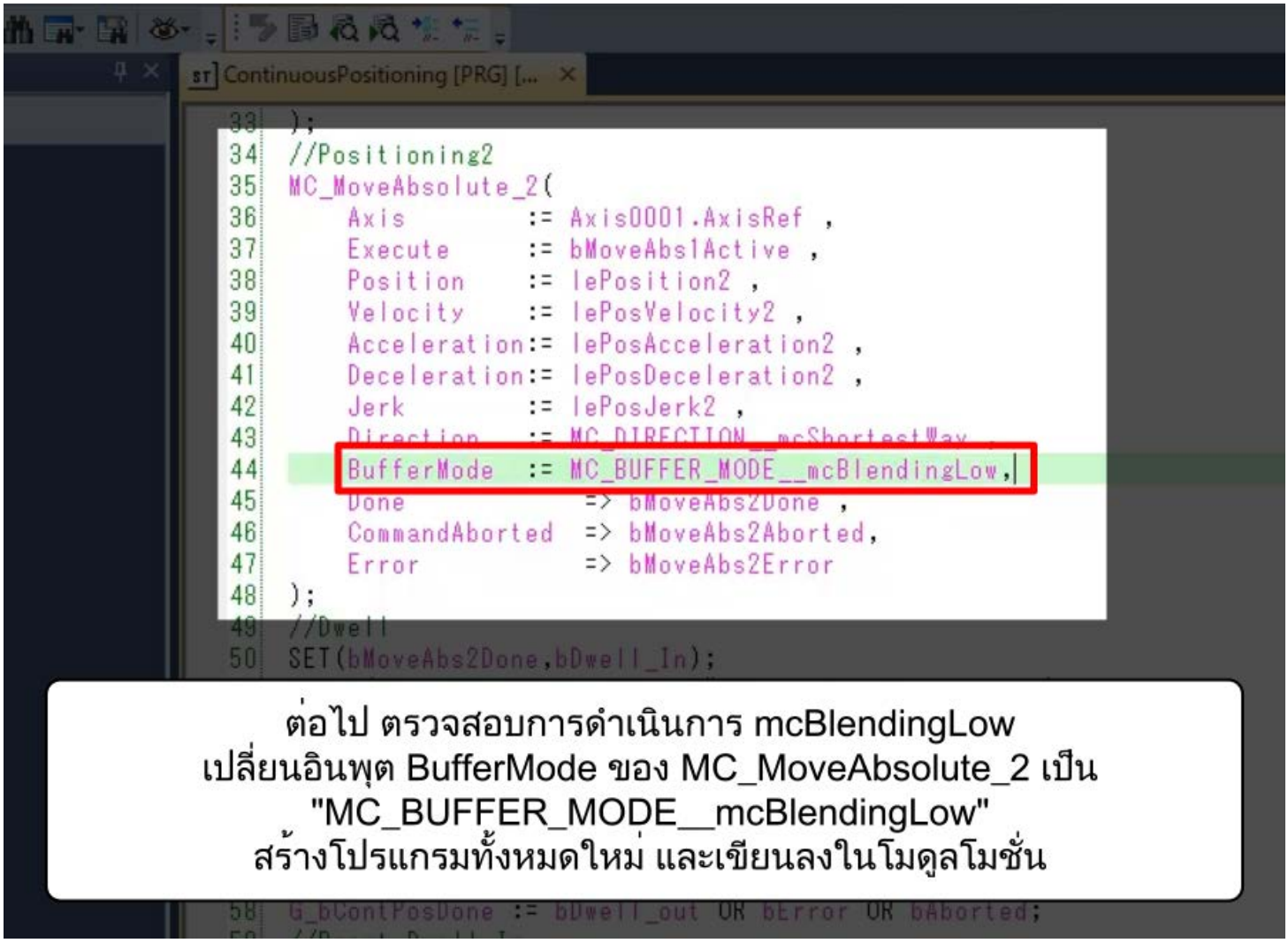
มอเตอร์ทำงานเมื่อ X25 เปิดอยู่
กราฟแสดงตำแหน่งที่ตั้งค่าไว้และความเร็วที่ตั้งค่าไว้

3.5 การตรวจสอบการดำเนินการ



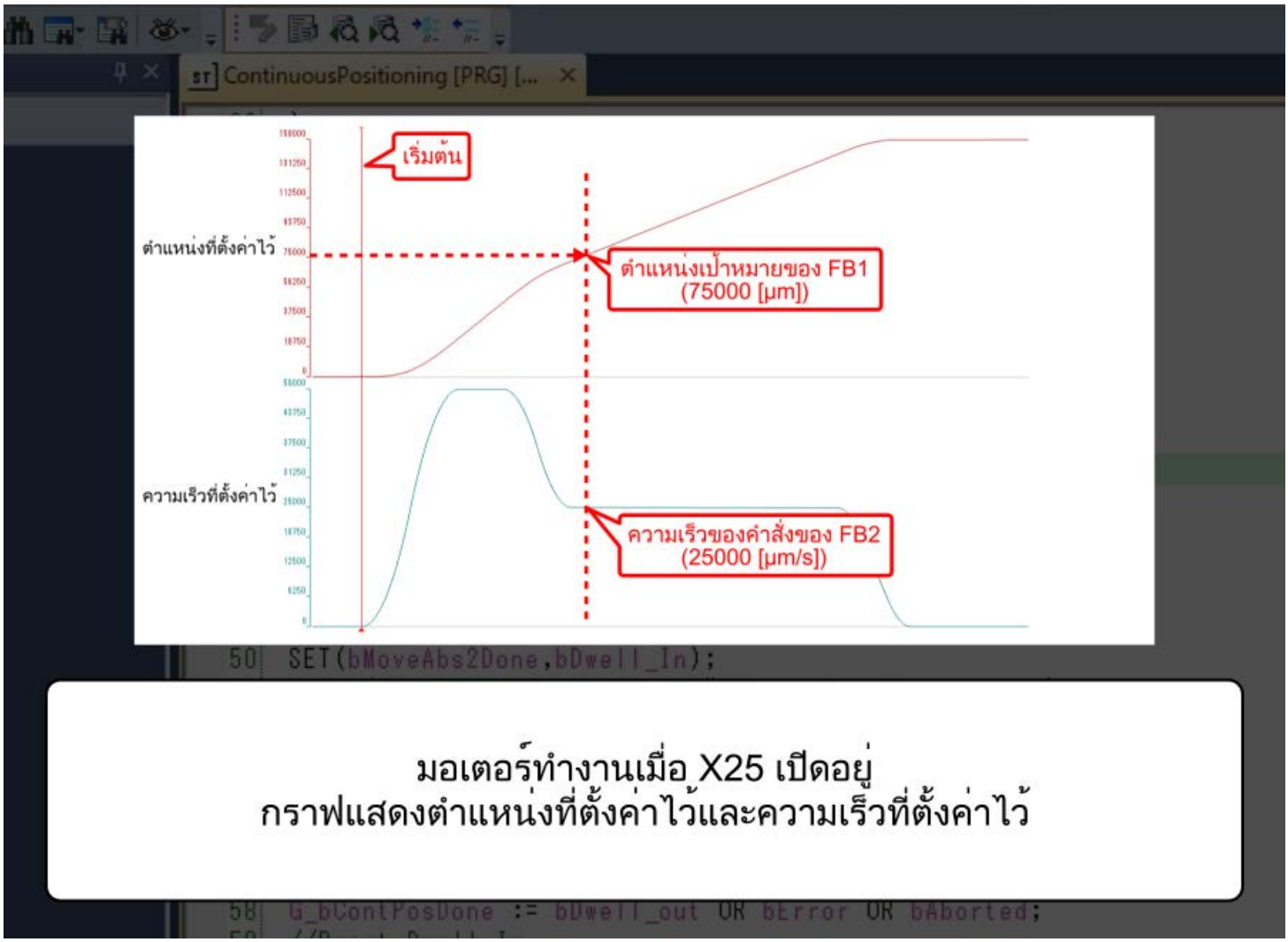
ใน mcBlendingNext ความเร็วของคำสั่งของ FB1
จะถูกนำไปใช้เมื่อถึงตำแหน่งเป้าหมายของ FB2
จากนั้นแกนจะเคลื่อนที่ไปยังตำแหน่งเป้าหมายของ FB2

3.5 การตรวจสอบการดำเนินการ



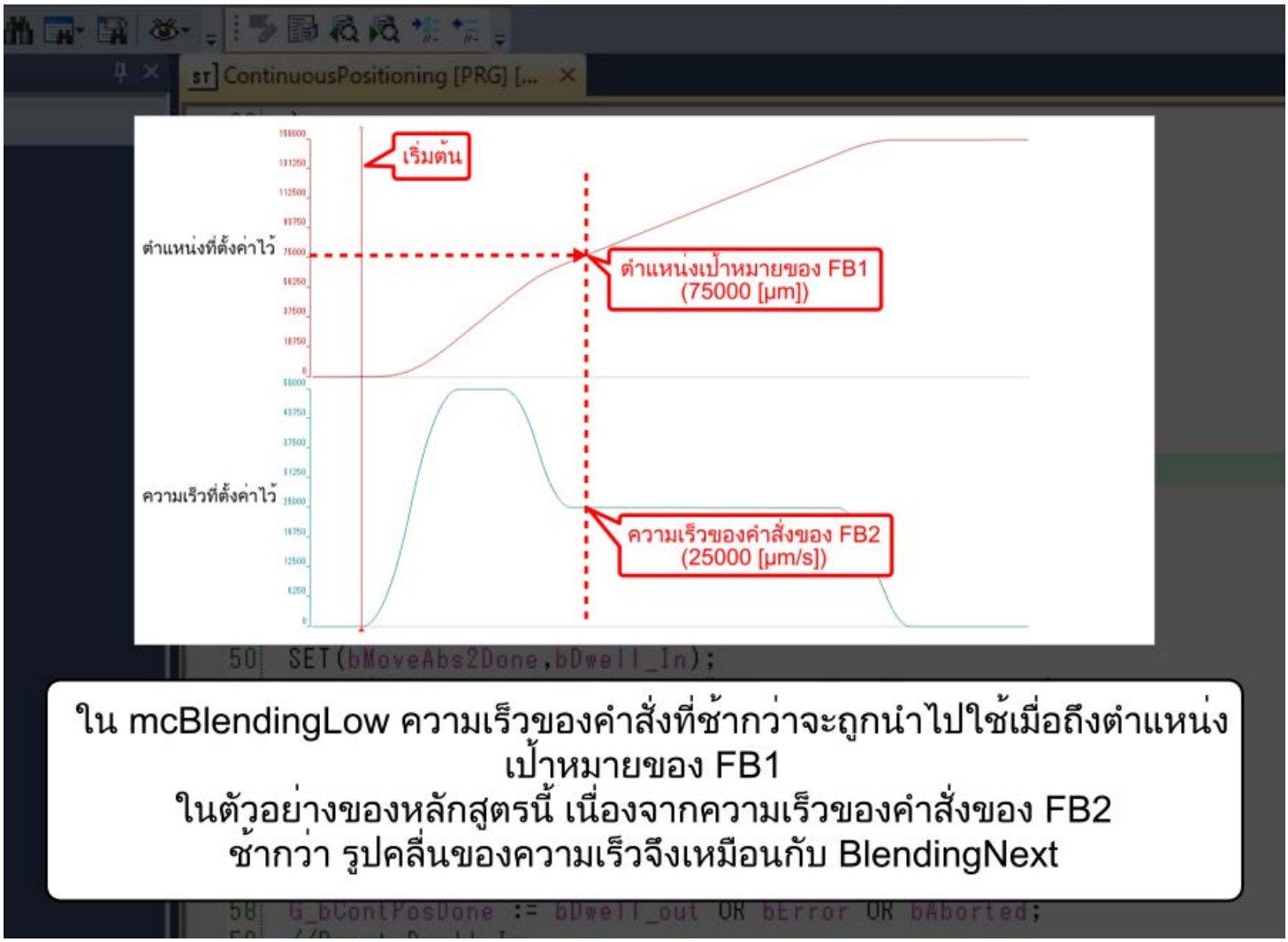
ต่อไป ตรวจสอบการดำเนินการ mcBlendingLow
เปลี่ยนอินพุต BufferMode ของ MC_MoveAbsolute_2 เป็น
"MC_BUFFER_MODE__mcBlendingLow"
สร้างโปรแกรมทั้งหมดใหม่ และเขียนลงในโมดูลโมชั่น

3.5 การตรวจสอบการดำเนินการ



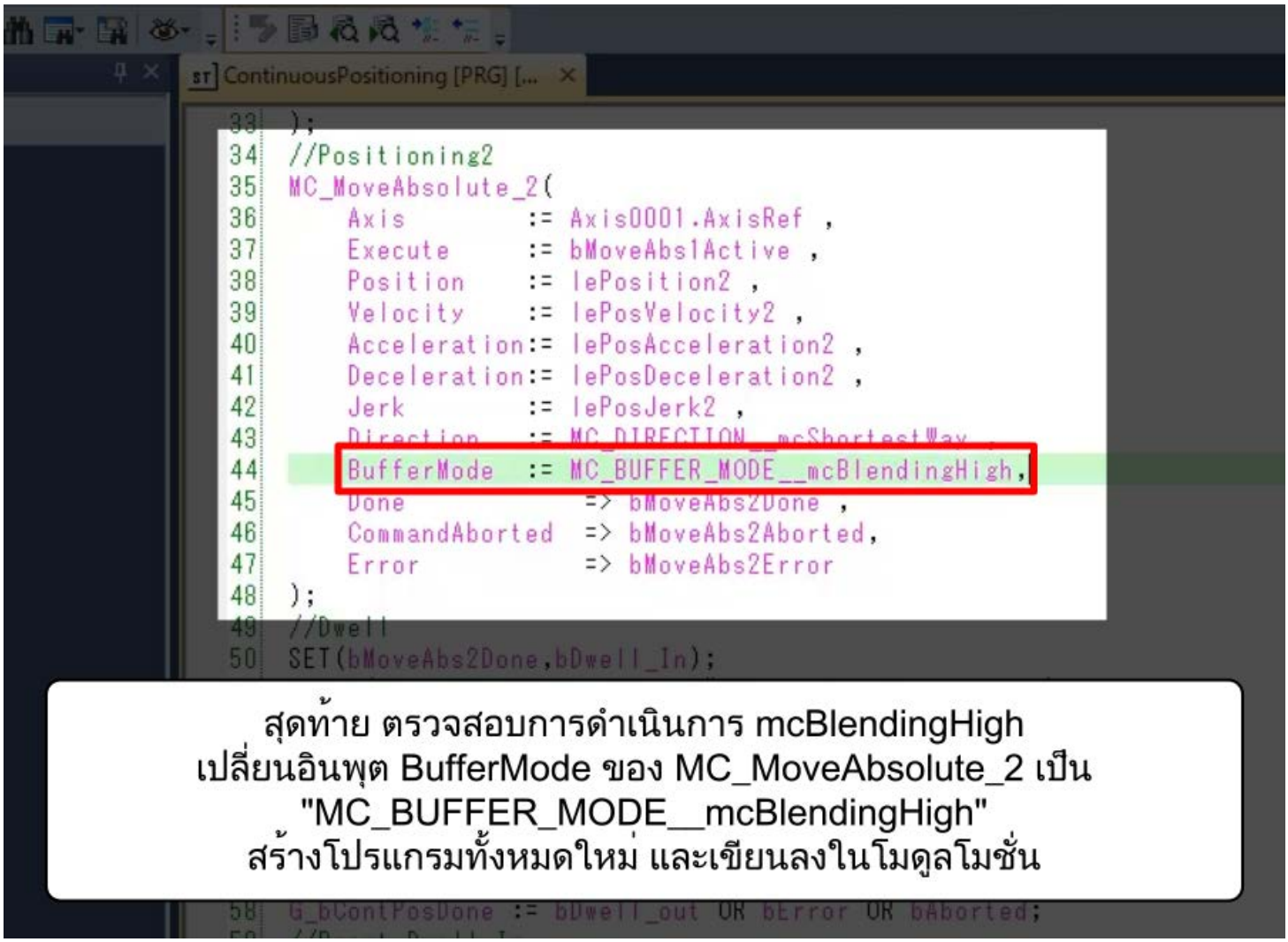
มอเตอร์ทำงานเมื่อ X25 เปิดอยู่
กราฟแสดงตำแหน่งที่ตั้งค่าไว้และความเร็วที่ตั้งค่าไว้

3.5 การตรวจสอบการดำเนินการ

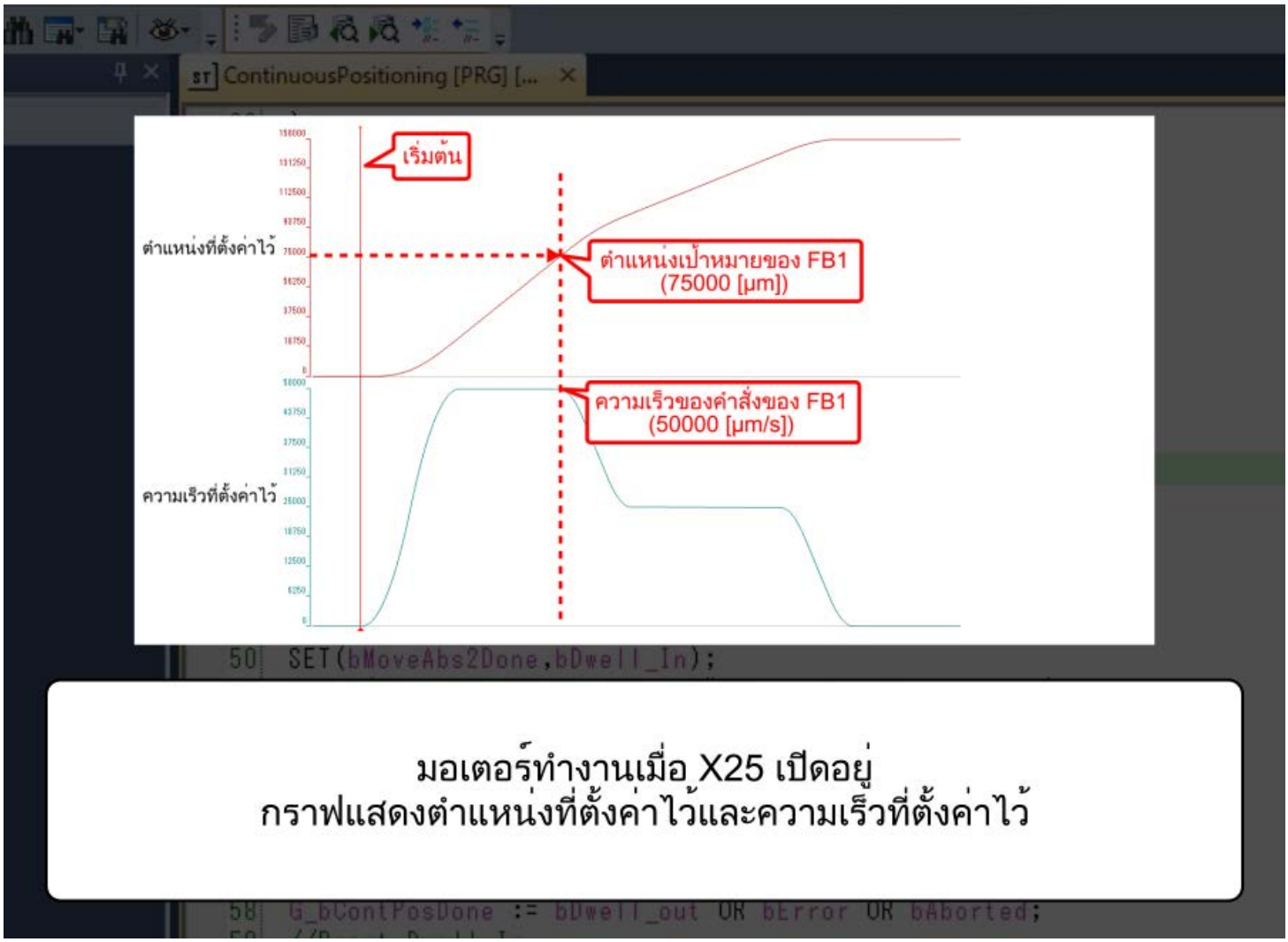


ใน mcBlendingLow ความเร็วของคำสั่งที่ช้ากว่าจะถูกนำไปใช้เมื่อถึงตำแหน่งเป้าหมายของ FB1
ในตัวอย่างของหลักสูตรนี้ เนื่องจากความเร็วของคำสั่งของ FB2 ช้ากว่า รูปคลื่นของความเร็วจึงเหมือนกับ BlendingNext

3.5 การตรวจสอบการดำเนินการ

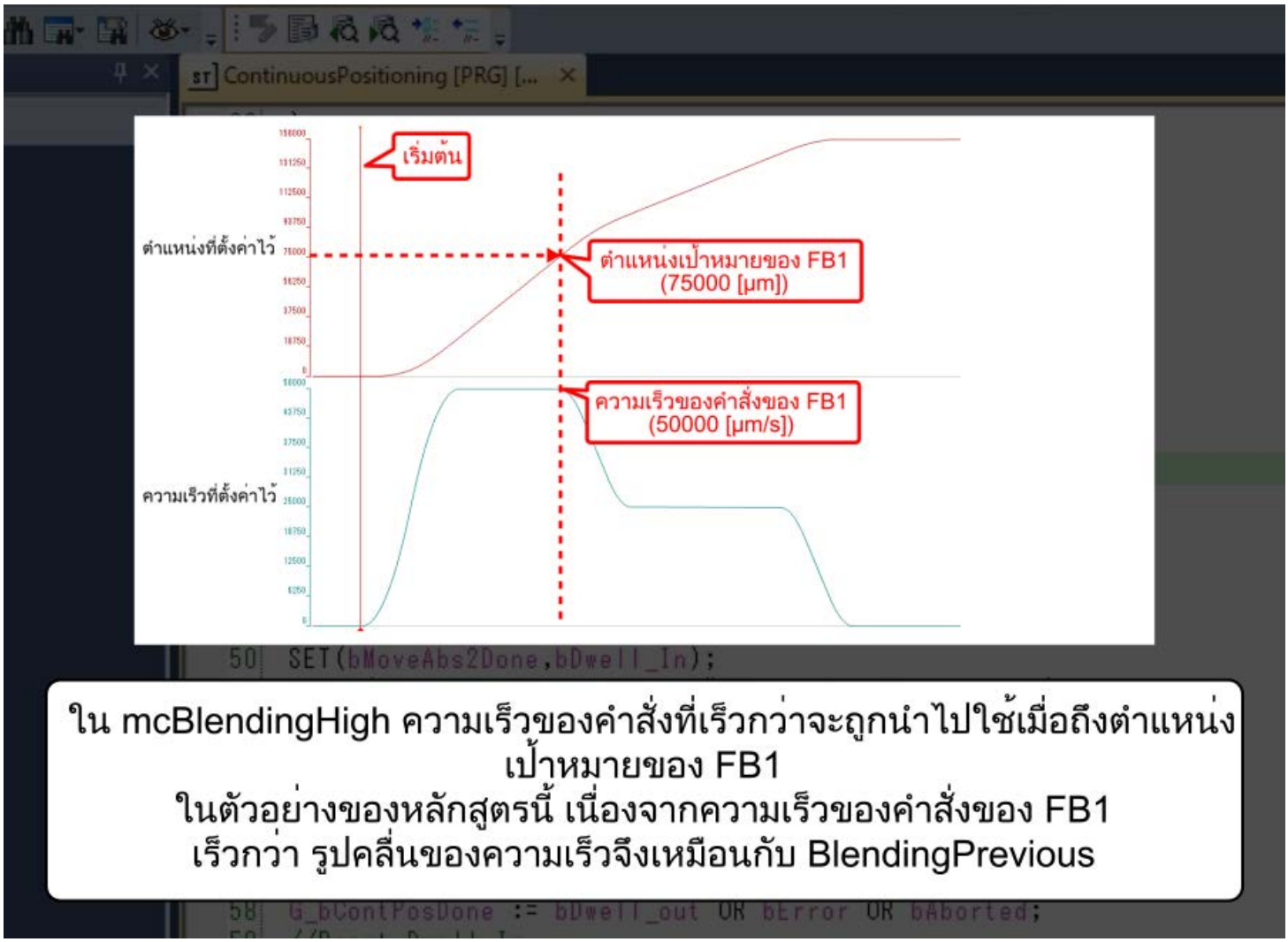


3.5 การตรวจสอบการดำเนินการ



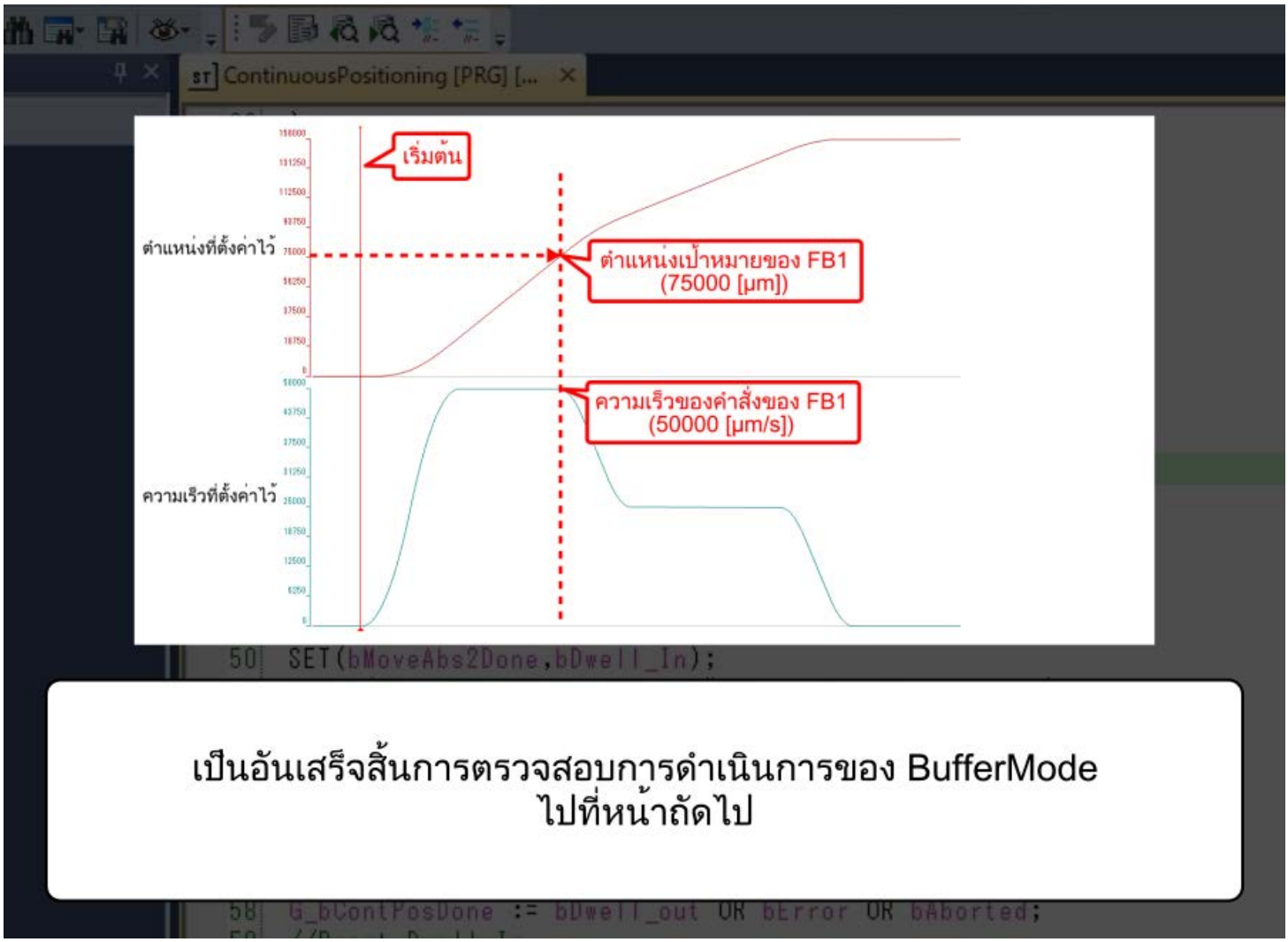
มอเตอร์ทำงานเมื่อ X25 เปิดอยู่
กราฟแสดงตำแหน่งที่ตั้งค่าไว้และความเร็วที่ตั้งค่าไว้

3.5 การตรวจสอบการดำเนินการ



ใน mcBlendingHigh ความเร็วของคำสั่งที่เร็วกว่าจะถูกนำไปใช้เมื่อถึงตำแหน่งเป้าหมายของ FB1
ในตัวอย่างของหลักสูตรนี้ เนื่องจากความเร็วของคำสั่งของ FB1 เร็วกว่า รูปคลื่นของความเร็วจึงเหมือนกับ BlendingPrevious

3.5 การตรวจสอบการดำเนินการ



เป็นอันเสร็จสิ้นการตรวจสอบการดำเนินการของ BufferMode
ไปที่หน้าถัดไป

3.6 บทสรุปของบทนี้

ในบทนี้คุณได้เรียนรู้เกี่ยวกับ:

- การยกเลิก
- ทำบัปเฟอร์แล้ว
- การผสม
- ตัวอย่างโปรแกรม
- การตรวจสอบการดำเนินการ

ประเด็นสำคัญ

การยกเลิก	• เมื่อ FB ประเภทการดำเนินการกำลังทำงานอยู่ และ FB ประเภทการดำเนินการถัดไปถูกดำเนินการ การยกเลิกจะขัดจังหวะ FB ที่กำลังถูกดำเนินการอยู่และจะดำเนินการ FB ถัดไป
ทำบัปเฟอร์แล้ว	• เมื่อ FB ประเภทการดำเนินการกำลังทำงานอยู่ และ FB ประเภทการดำเนินการถัดไปถูกดำเนินการ การบัฟเฟอร์จะรอจนกว่า FB ที่กำลังถูกดำเนินการอยู่จะเสร็จและจะดำเนินการ FB ถัดไป
การผสม	• เมื่อ FB ประเภทการดำเนินการกำลังทำงานอยู่ และ FB ประเภทการดำเนินการถัดไปถูกดำเนินการ การผสมจะดำเนินการ FB ถัดไปโดยไม่หยุดการดำเนินการของ FB ที่กำลังถูกดำเนินการอยู่ • ในการผสม มีวิธีการสลับความเร็วสี่วิธี: BlendingLow BlendingHigh BlendingPrevious และ BlendingNext
ตัวอย่างโปรแกรม	• เลือกโหมดบัฟเฟอร์ด้วยอินพุต BufferMode ของการดำเนินการ FB
การตรวจสอบการดำเนินการ	• คุณได้ตรวจสอบความแตกต่างในการดำเนินการของแต่ละโหมดบัฟเฟอร์ในวิดีโอแล้ว

บทที่ 4 การดำเนินการด้วย PLC CPU

ดาวน์โหลดโปรแกรมตัวอย่างเพื่อใช้ในบทนี้จากลิงค์ด้านล่าง
เนื้อหาของโปรแกรมเหมือนกับโปรแกรมตัวอย่างที่อธิบายไว้ในบทที่ 2 และบทที่ 3
เฉพาะวิธีการเขียนโปรแกรมเท่านั้นที่แตกต่างกัน

RD78GBasic2_sample2.zip (1.39 MB)

4.1 การรีจิสเตอร์ไลบารี่ข้อมูลโมดูลโมชั่น FB

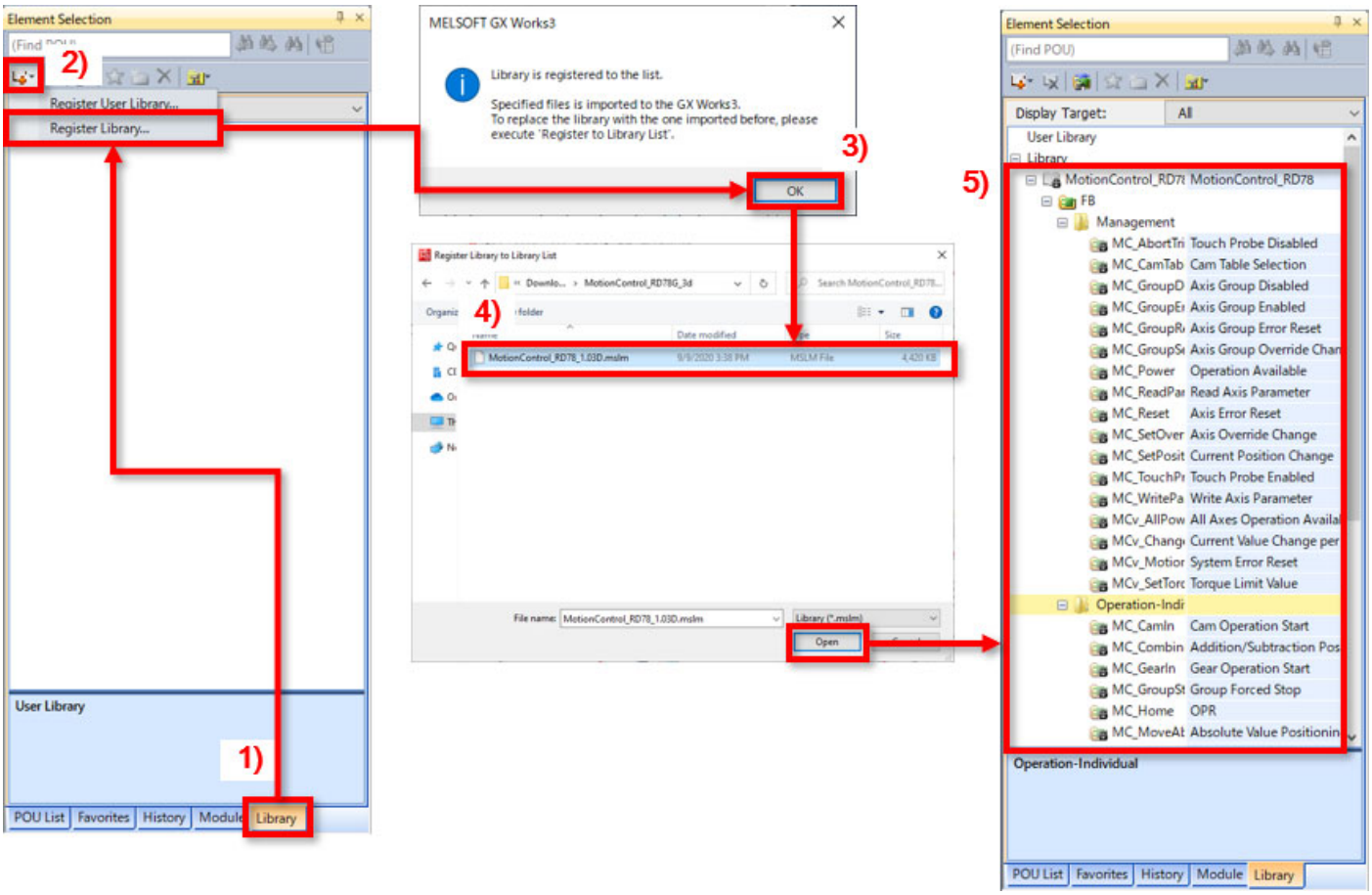
(1) ดาวน์โหลดคลังข้อมูล FB
Motion control FB สามารถใช้ในโปรแกรมของ PLC CPU โดยรีจิสเตอร์ไลบารี่ข้อมูล FB สำหรับโมดูลโมชั่นไปที่ GX Works3
ดาวน์โหลดไลบารี่ข้อมูล FB จากลิงค์ด้านล่าง และแตกไฟล์ ZIP ไปยังปลายทางที่ต้องการ

MotionControl_RD78G_3d.zip(4.29 MB)

(หมายเหตุ) สามารถดาวน์โหลดไลบารี่ข้อมูล FB เวอร์ชันล่าสุดได้จากเว็บไซต์ส่วนกลางของ MITSUBISHI ELECTRIC FA

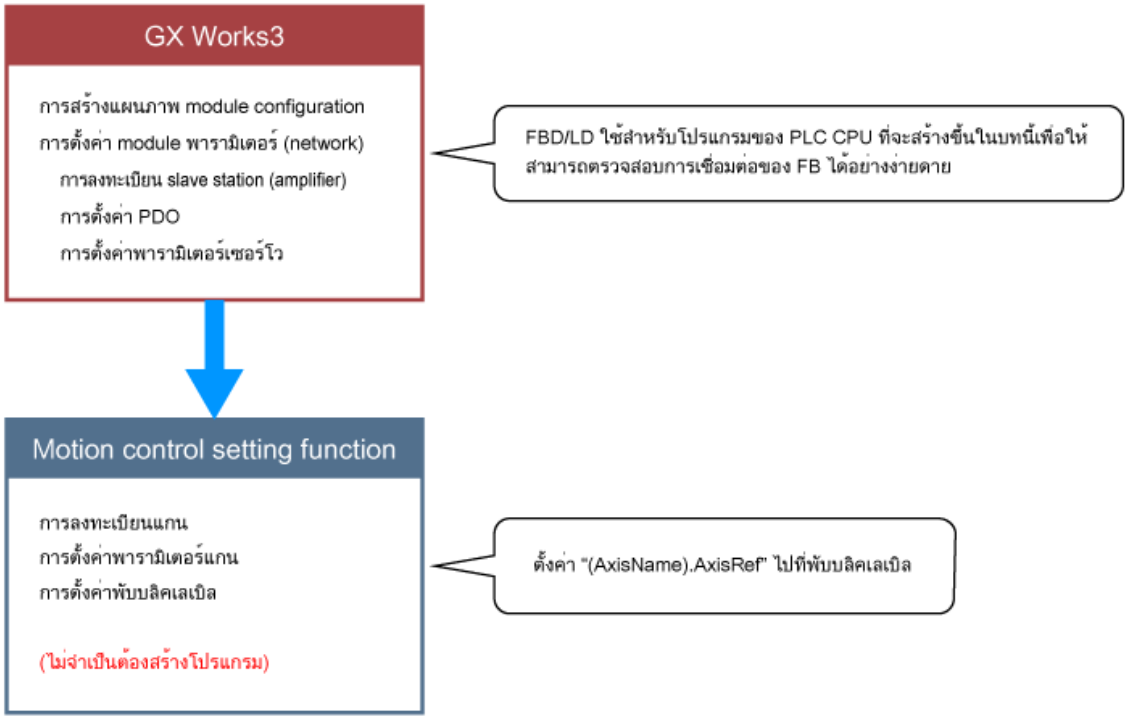
(2) การรีจิสเตอร์ไลบารี่ข้อมูล FB

1) เปิดโปรเจ็คใน GX Works3 และเปิดแถบคลังข้อมูลในหน้าต่าง Element Selection
2) คลิกปุ่ม [Register to Library List] ที่ส่วนด้านบน และเลือก [Register Library]
3) เมื่อมีข้อความ "Library is registered to the list" ให้คลิก [OK]
4) เลือกไลบารี่ข้อมูล FB "MotionControl_RD78_****.mslm" และคลิก [Open]
   (**** ระบุเวอร์ชั่น)
5) Motion control FB ถูกรีจิสเตอร์ไปยังไลบารี่ข้อมูลในหน้าต่างการเลือกองค์ประกอบ

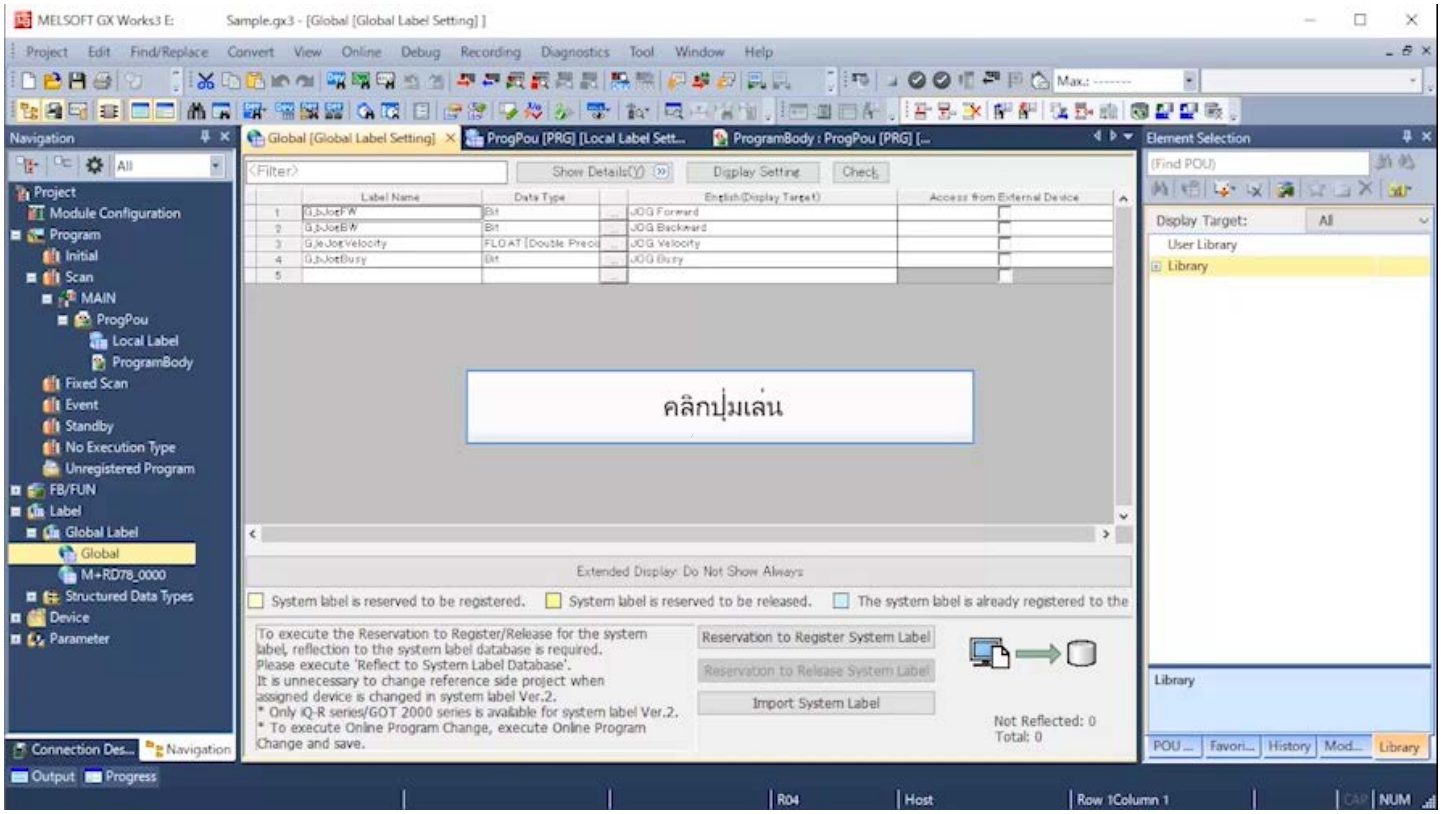


4.2 การสร้างโปรเจ็ค

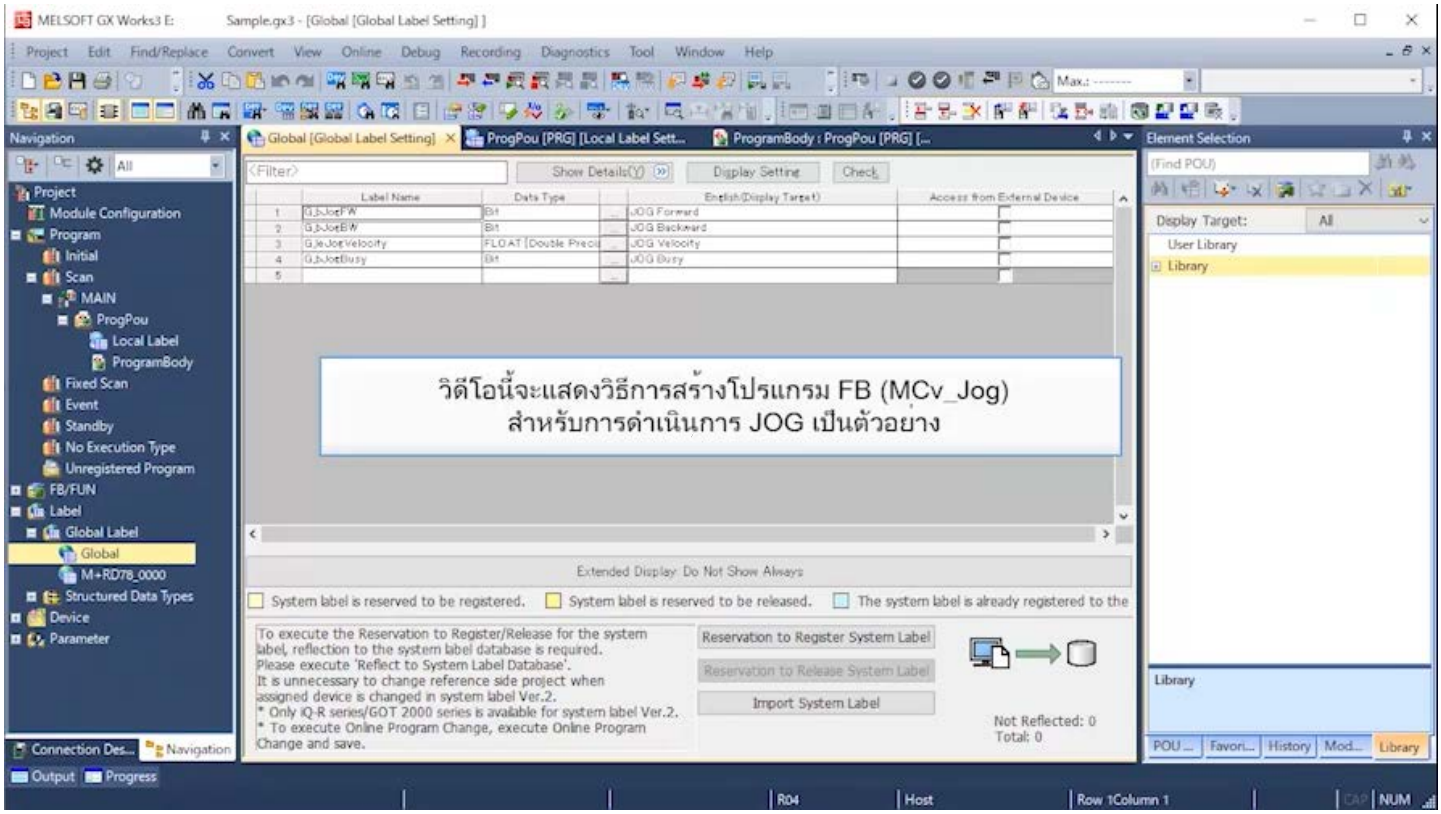
ขั้นตอนสำหรับการสร้างโปรเจ็คเหมือนกับที่อธิบายไว้ในส่วนที่แล้ว



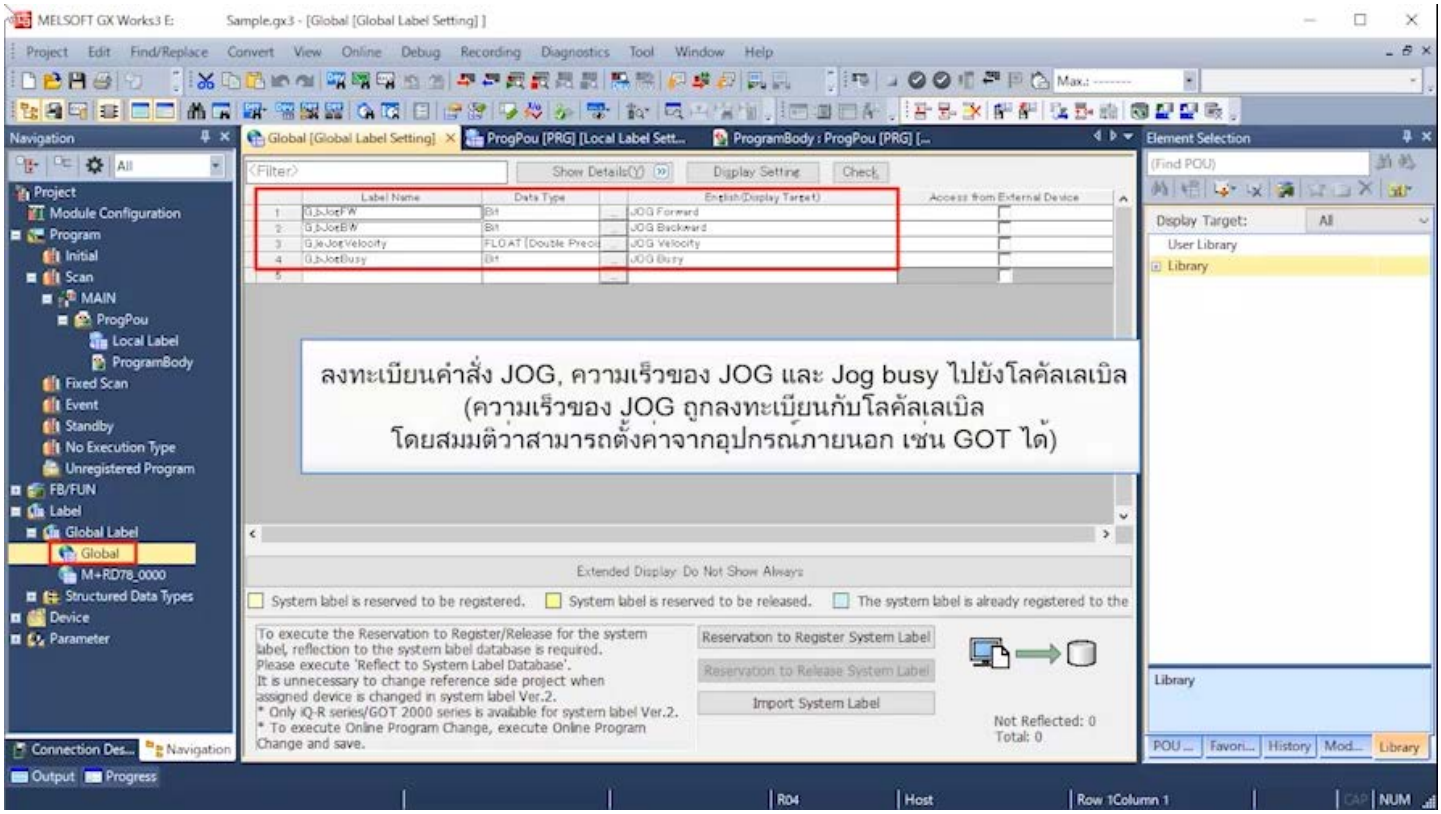
4.3 วิธีใช้ Motion Control FB



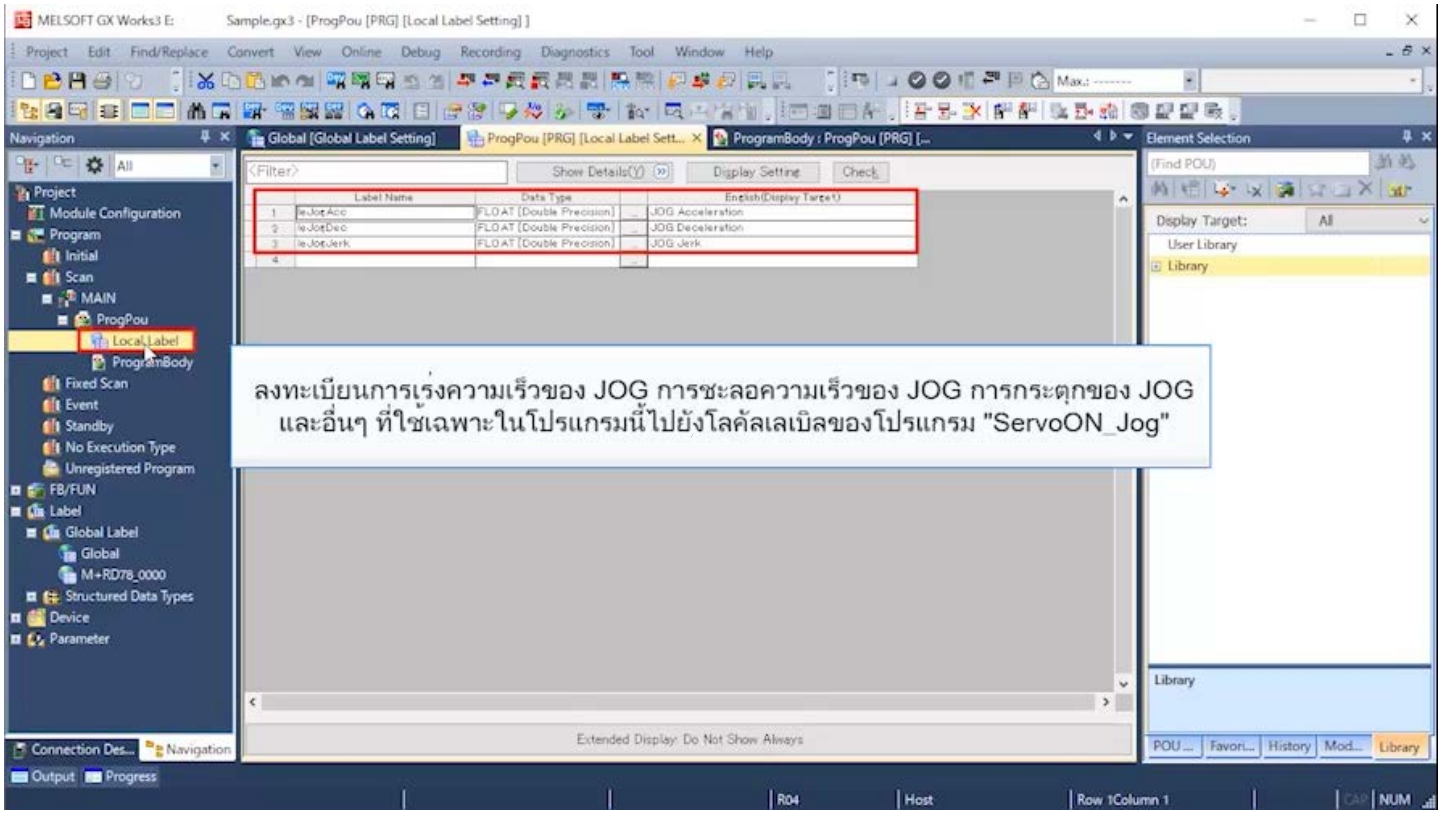
4.3 วิธีใช้ Motion Control FB



4.3 วิธีใช้ Motion Control FB

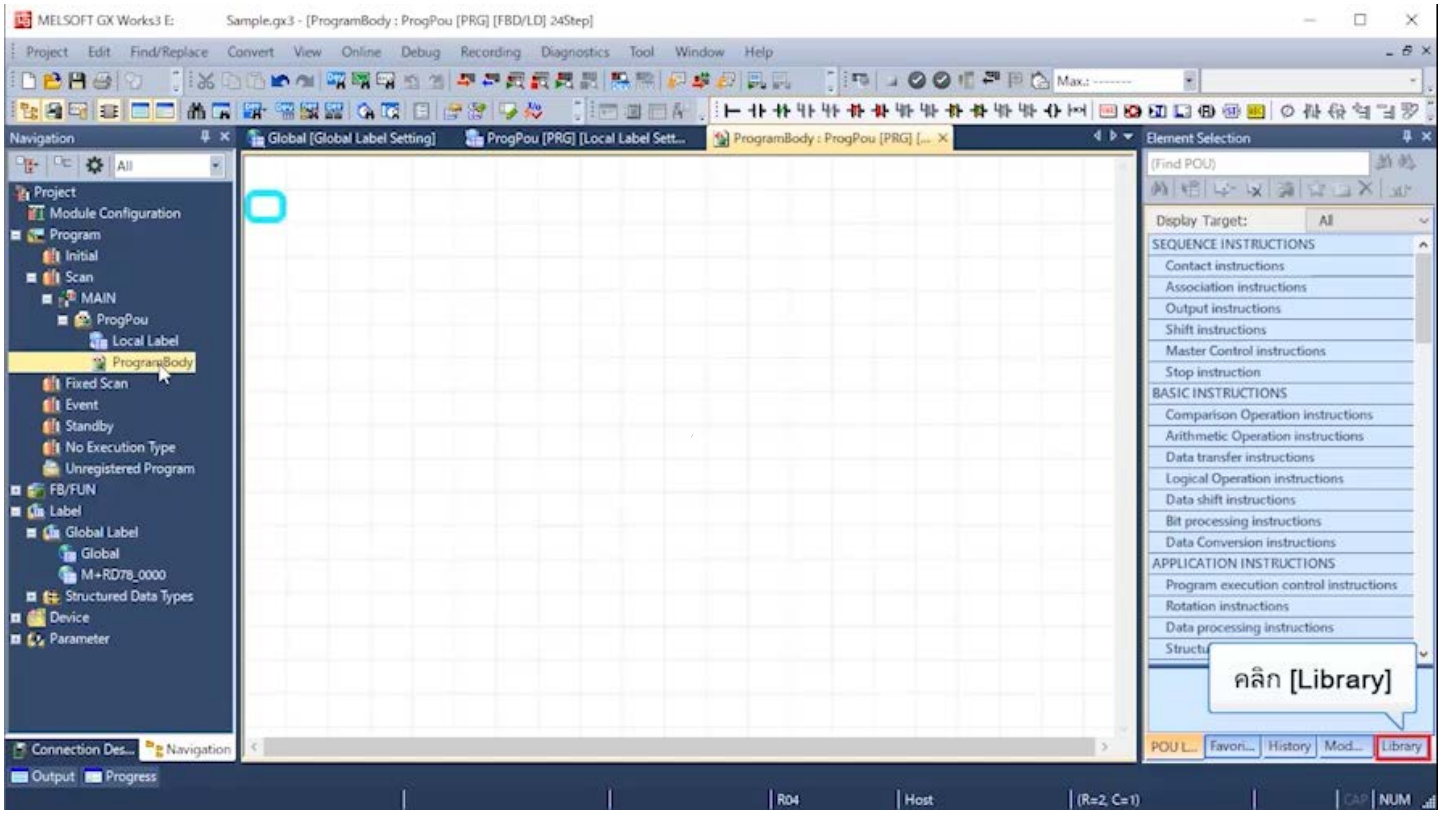


4.3 วิธีใช้ Motion Control FB

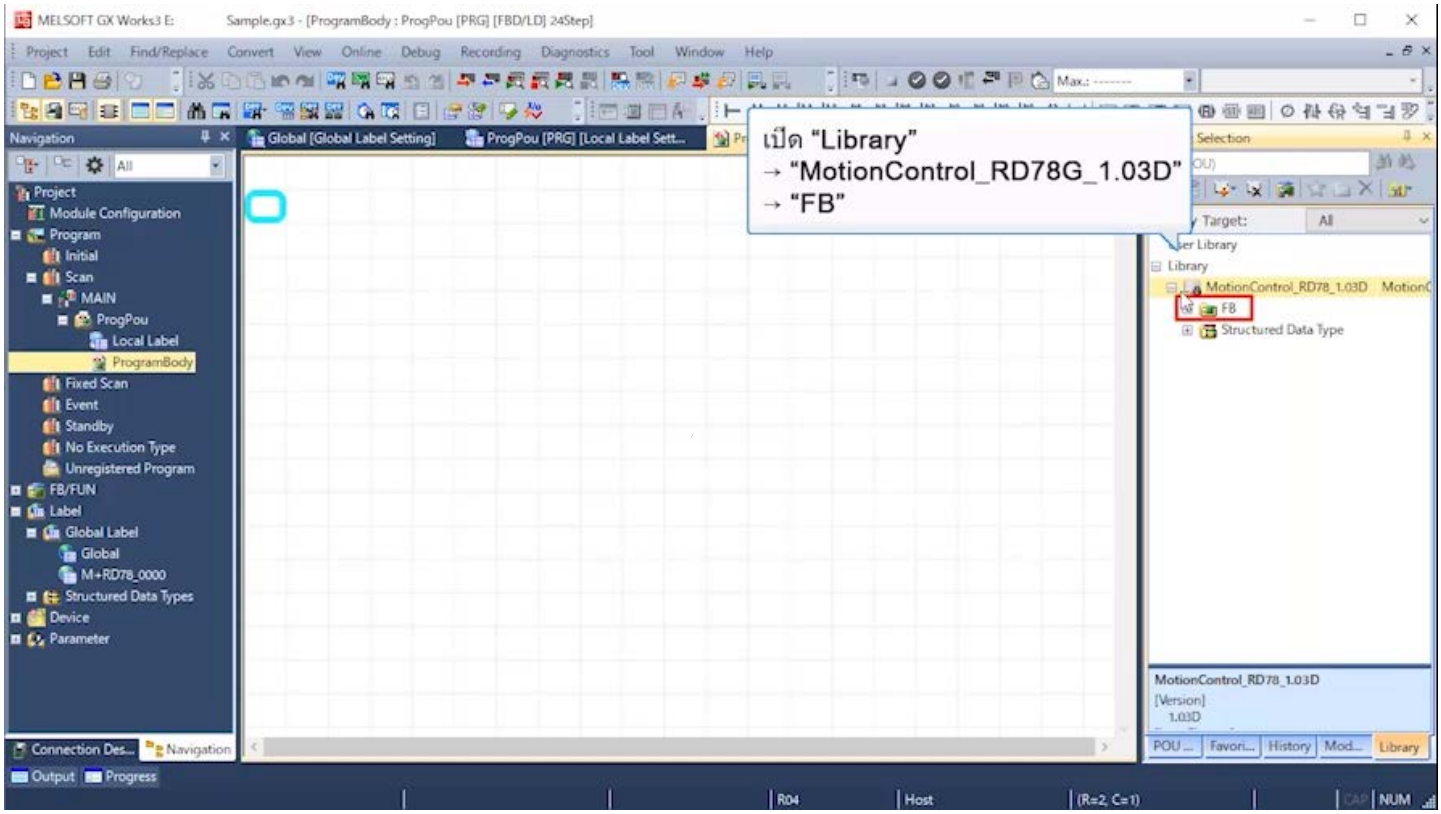


ลงทะเบียนการเร่งความเร็วของ JOG การชะลอความเร็วของ JOG การกระตุกของ JOG และอื่นๆ ที่ใช้เฉพาะในโปรแกรมนี้ไปยังโลคัลเลเบิลของโปรแกรม "ServoON_Jog"

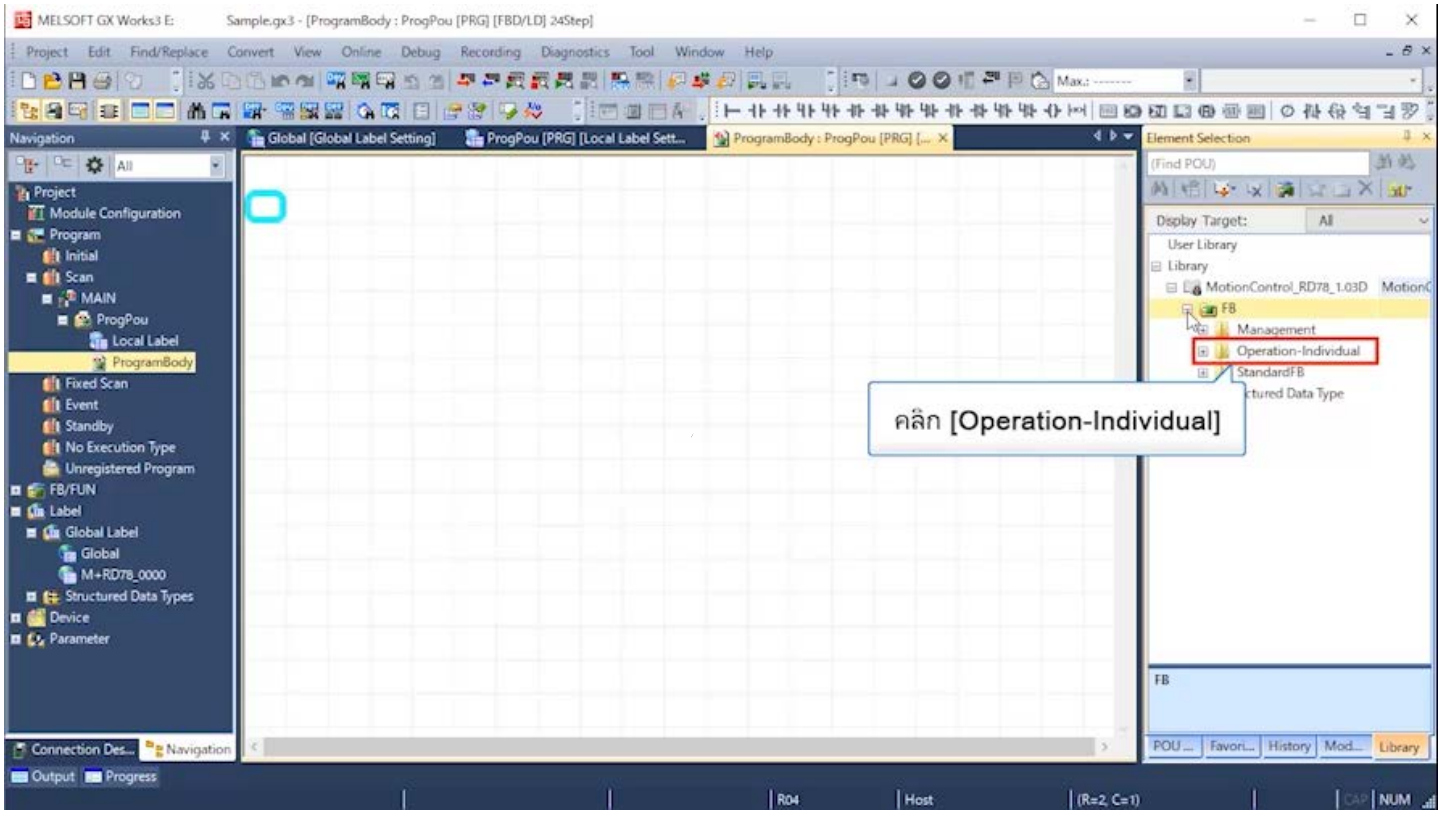
4.3 วิธีใช้ Motion Control FB



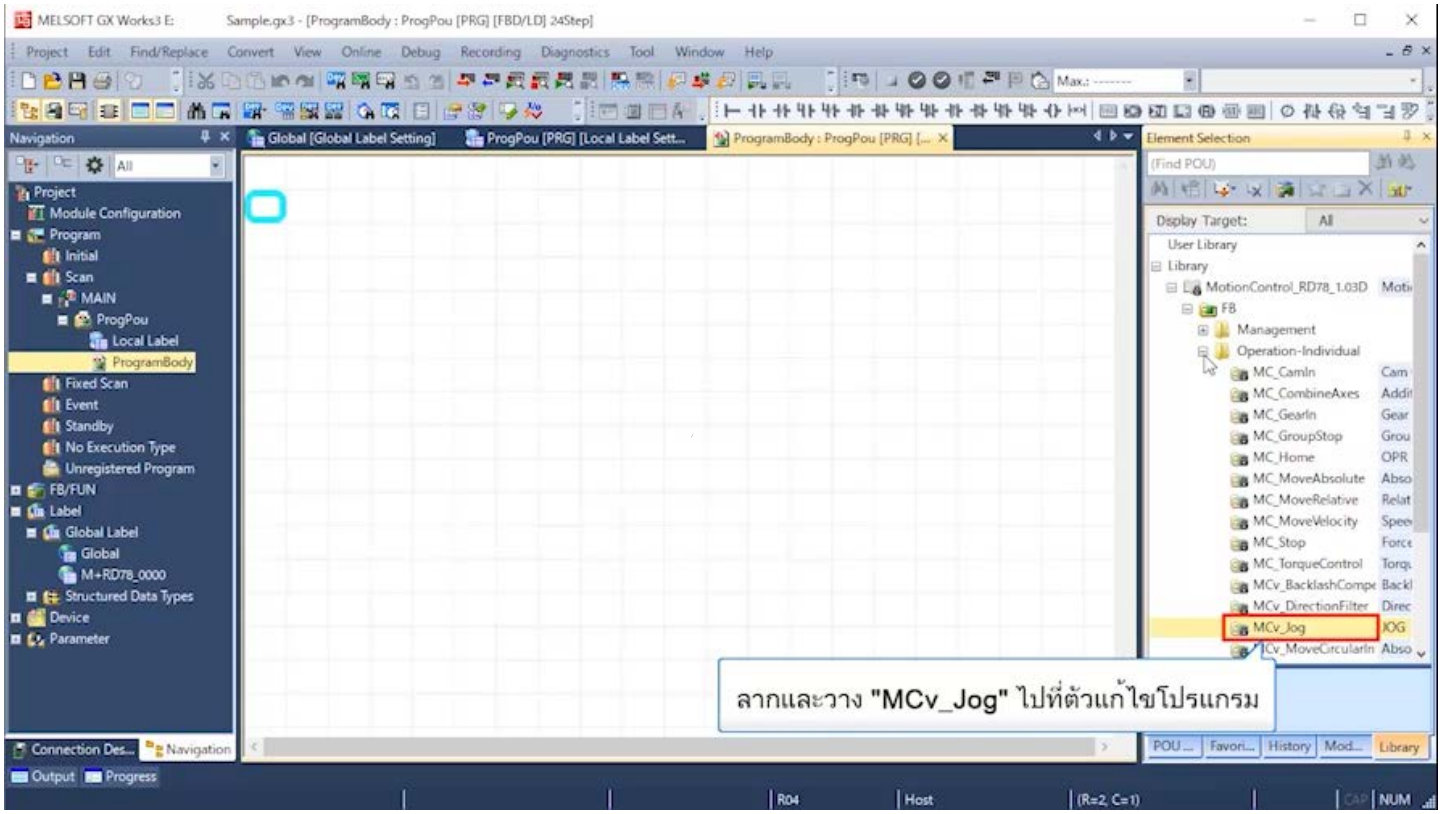
4.3 วิธีใช้ Motion Control FB



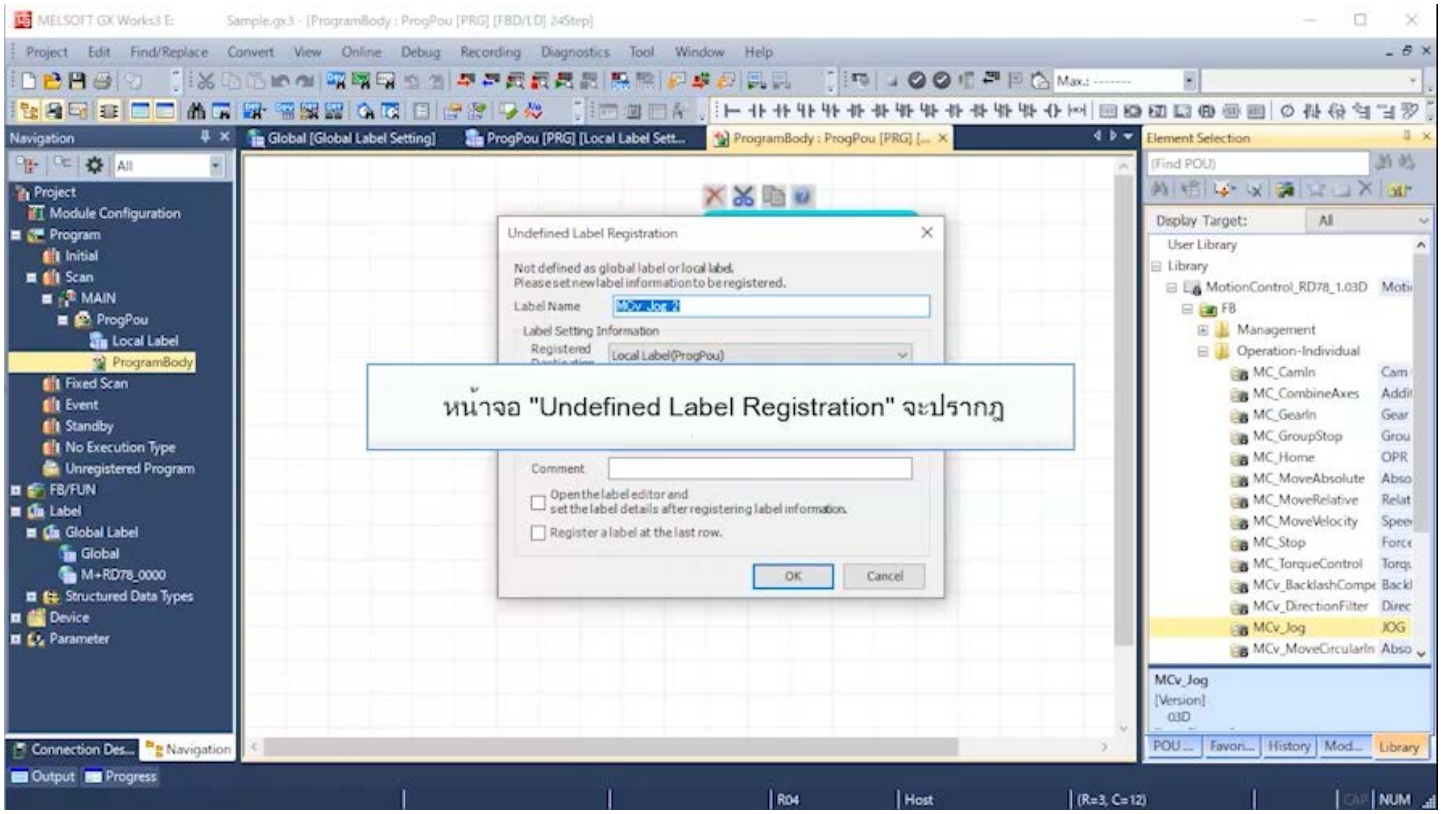
4.3 วิธีใช้ Motion Control FB



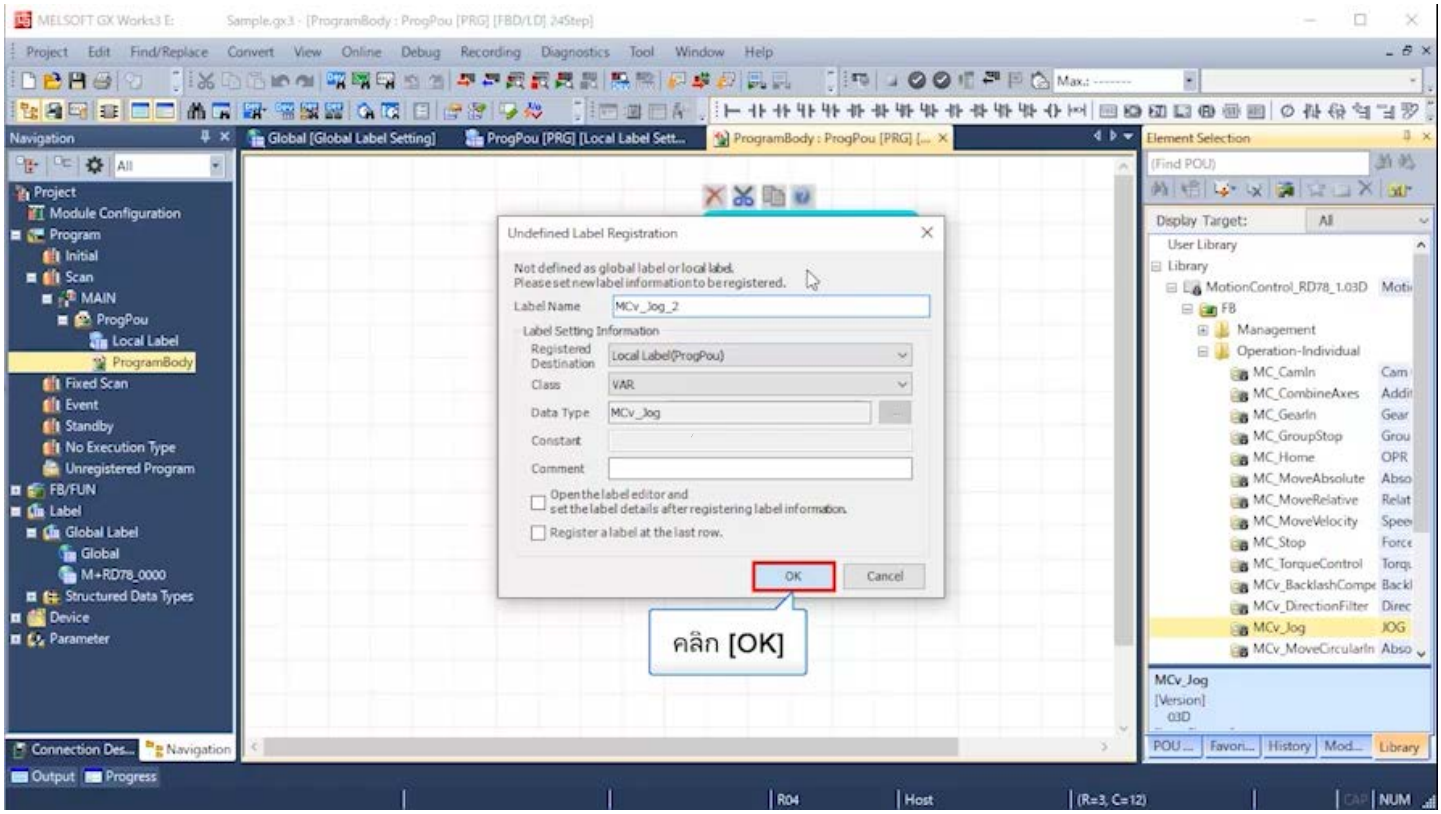
4.3 วิธีใช้ Motion Control FB



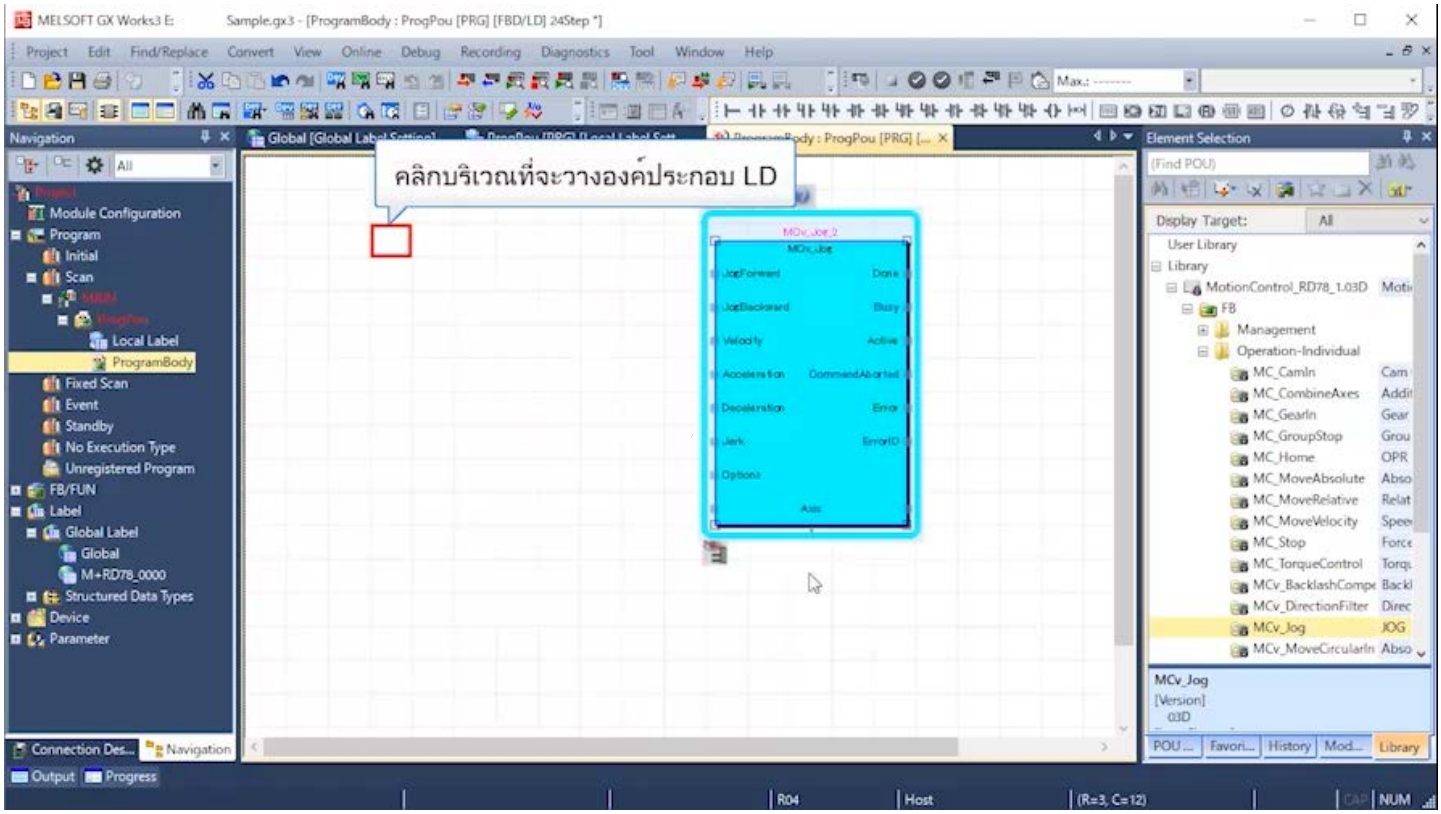
4.3 วิธีใช้ Motion Control FB



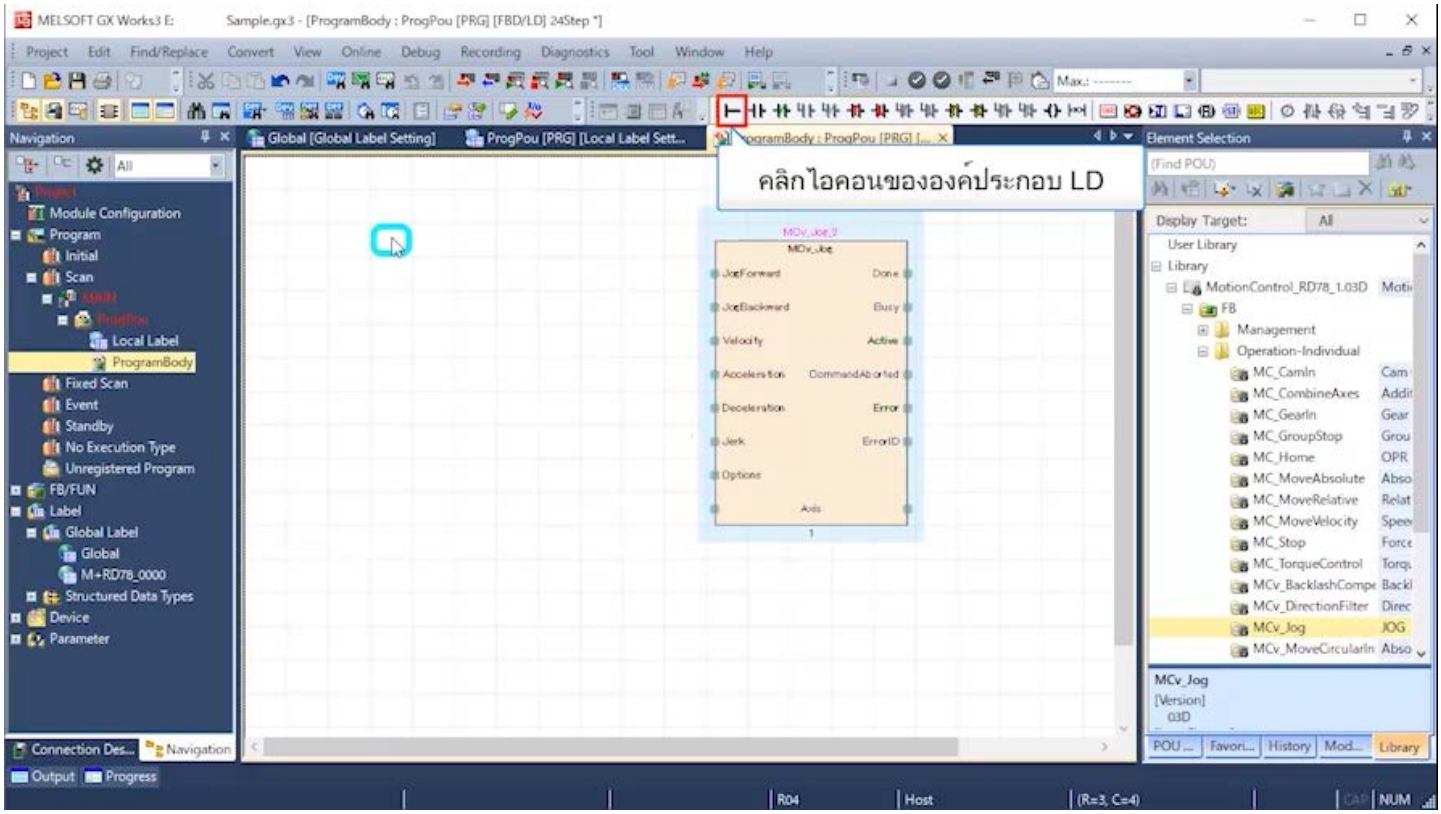
4.3 วิธีใช้ Motion Control FB



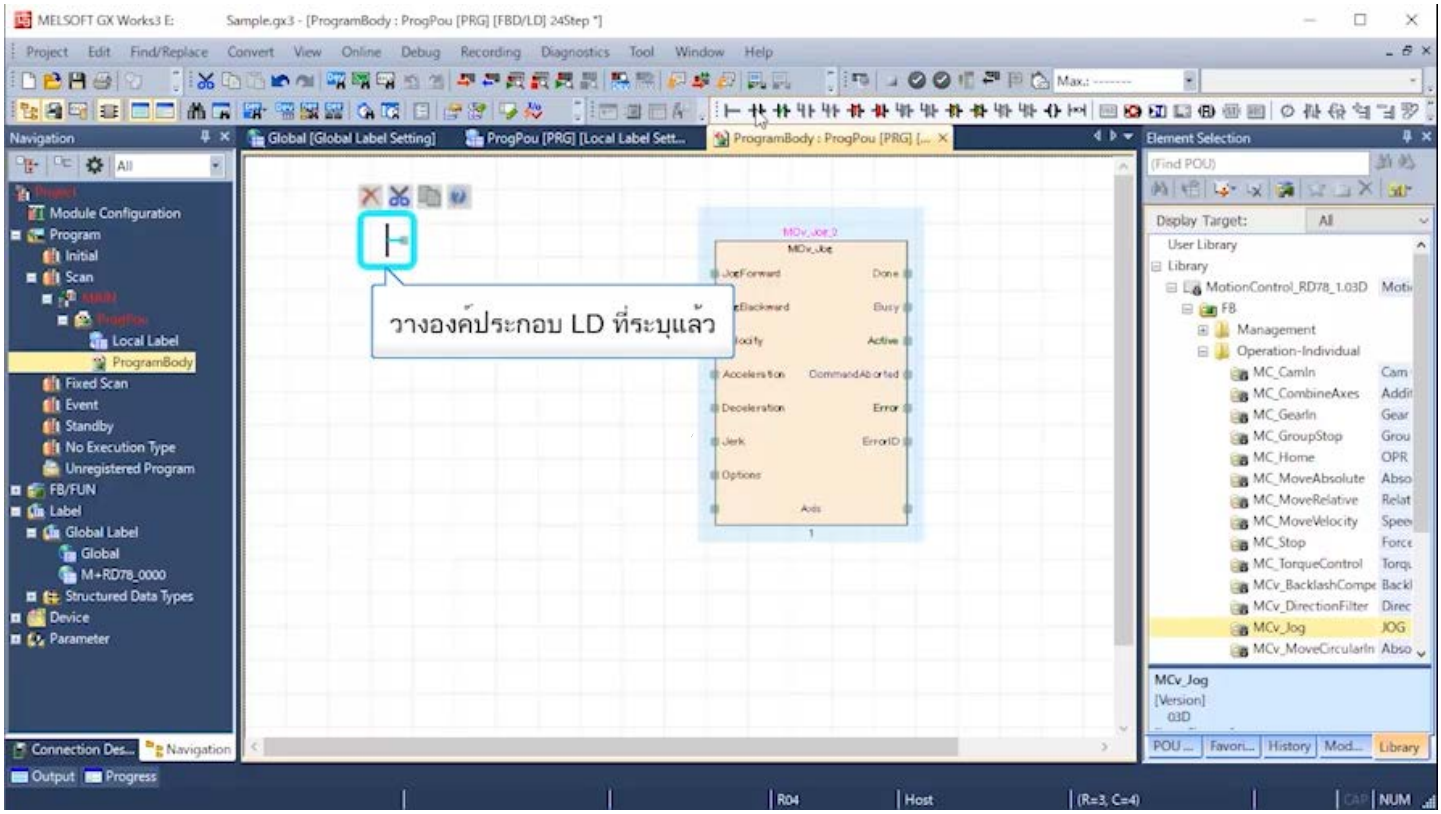
4.3 วิธีใช้ Motion Control FB



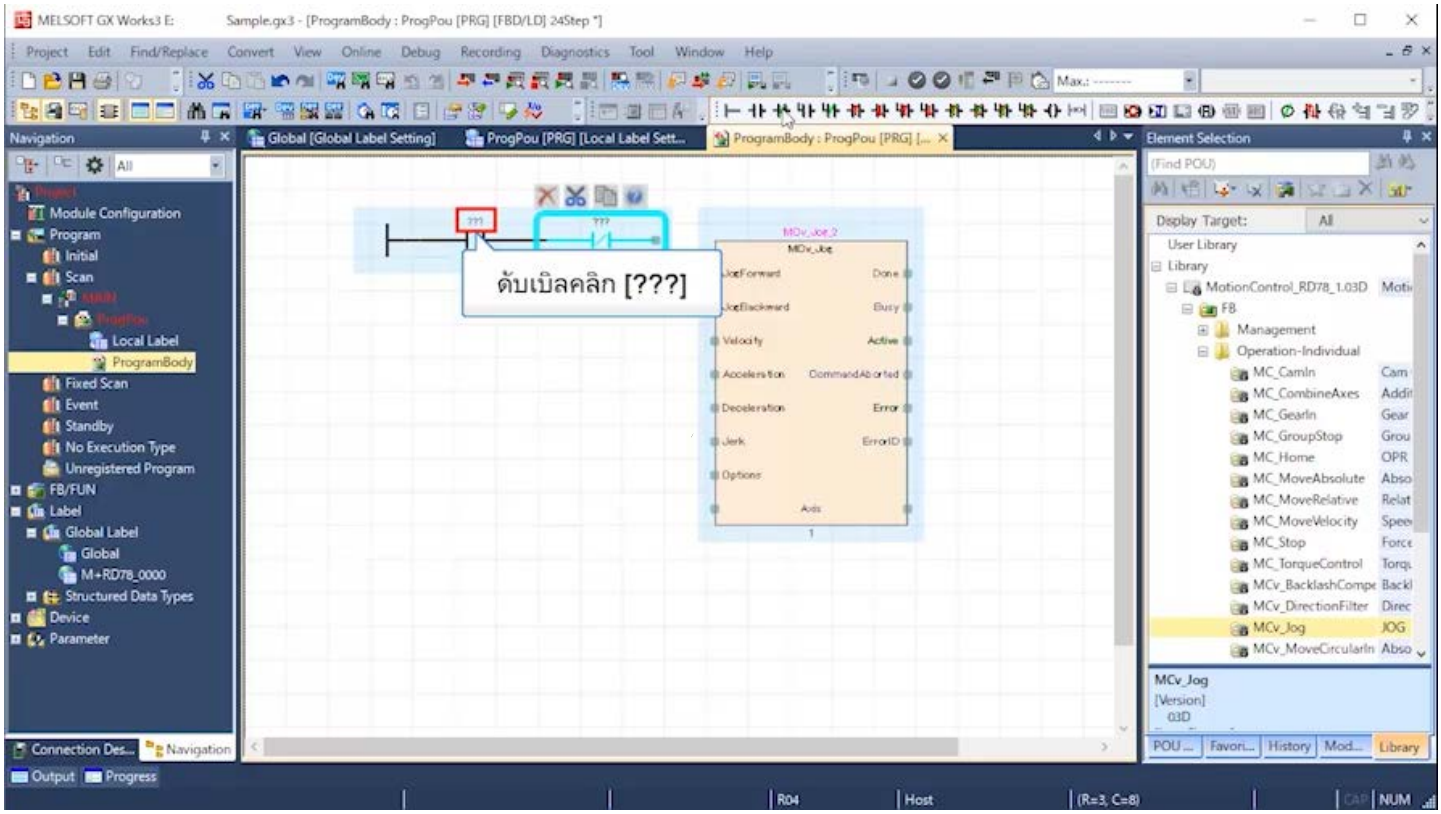
4.3 วิธีใช้ Motion Control FB



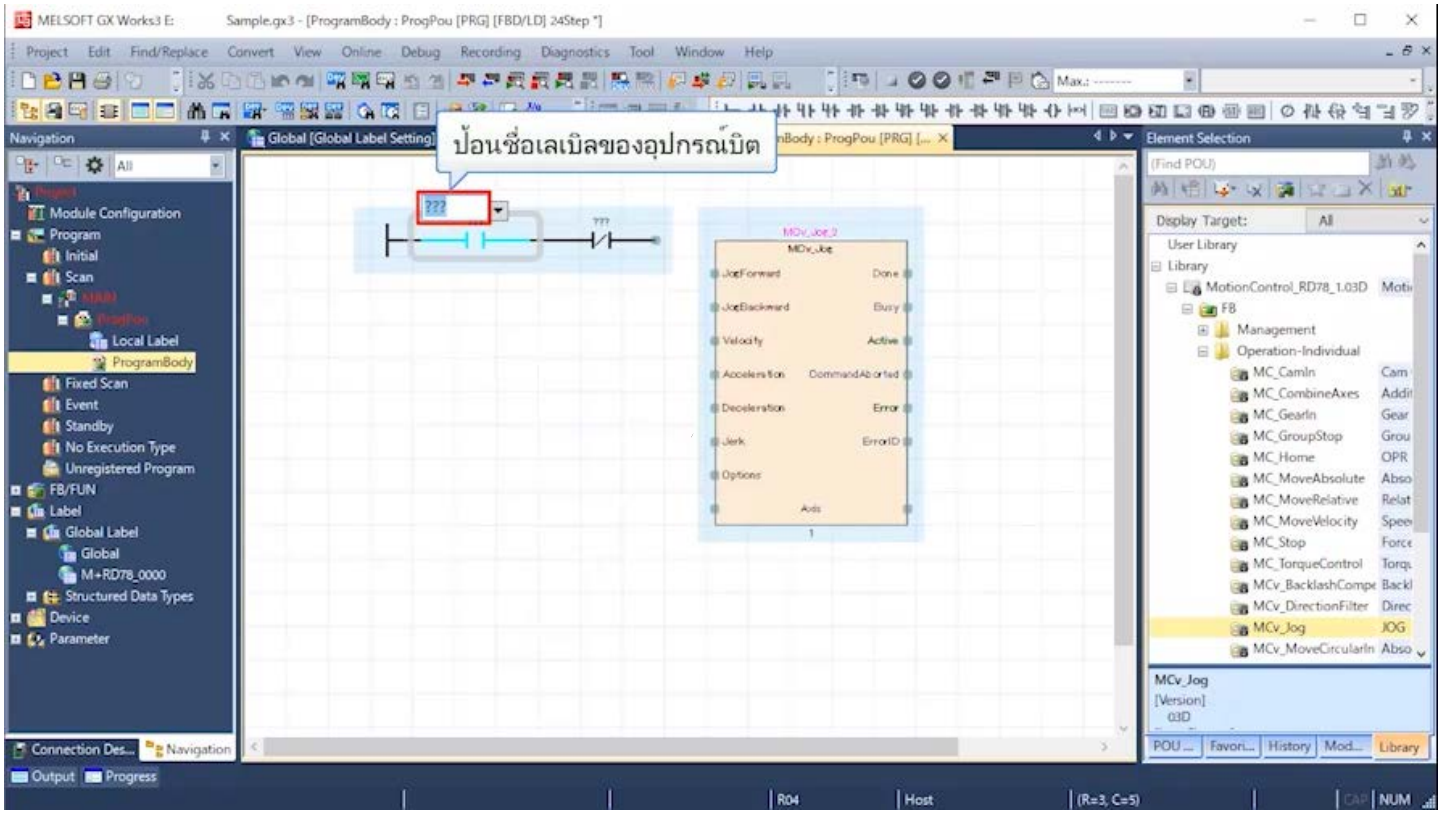
4.3 วิธีใช้ Motion Control FB



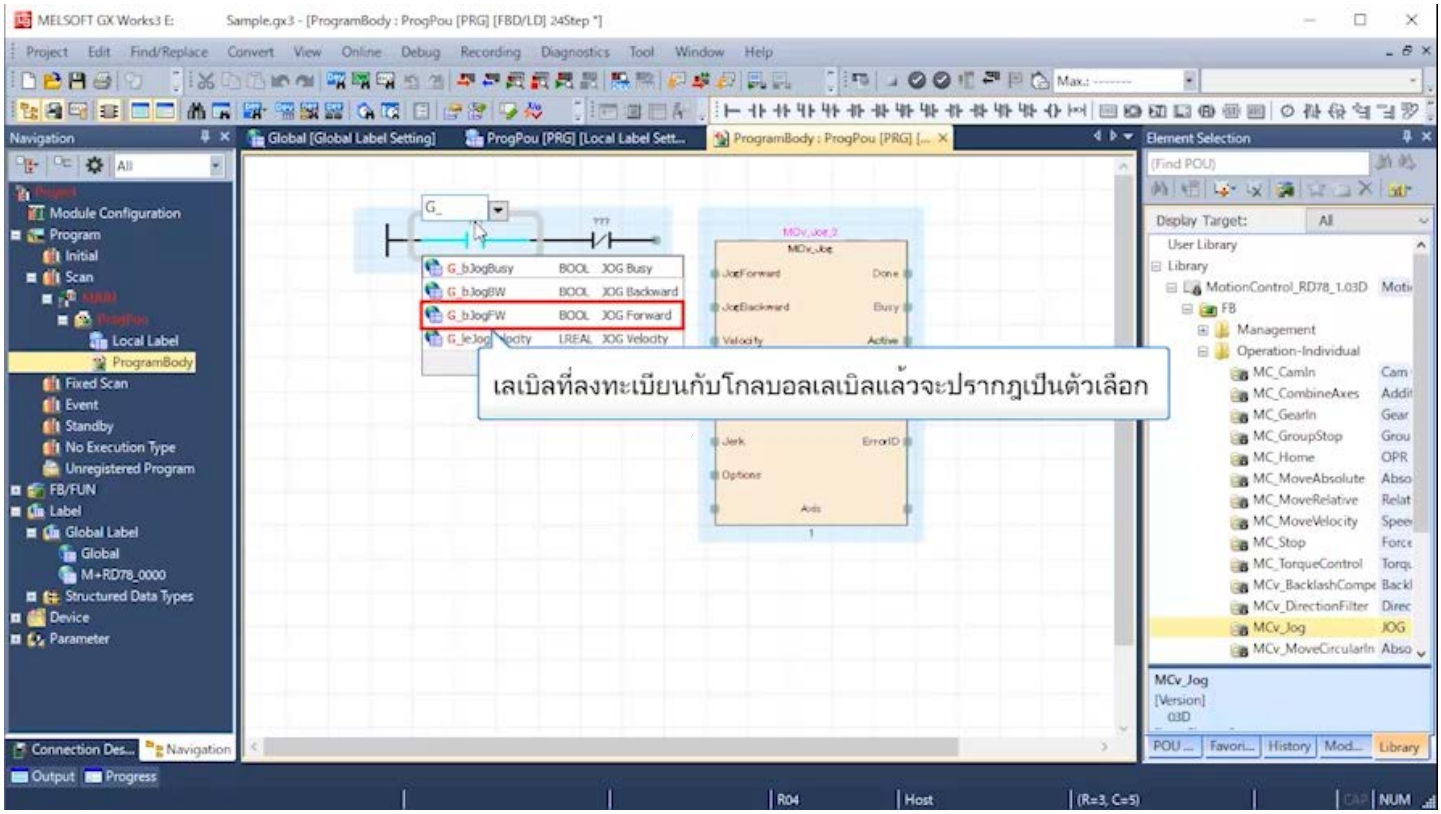
4.3 วิธีใช้ Motion Control FB



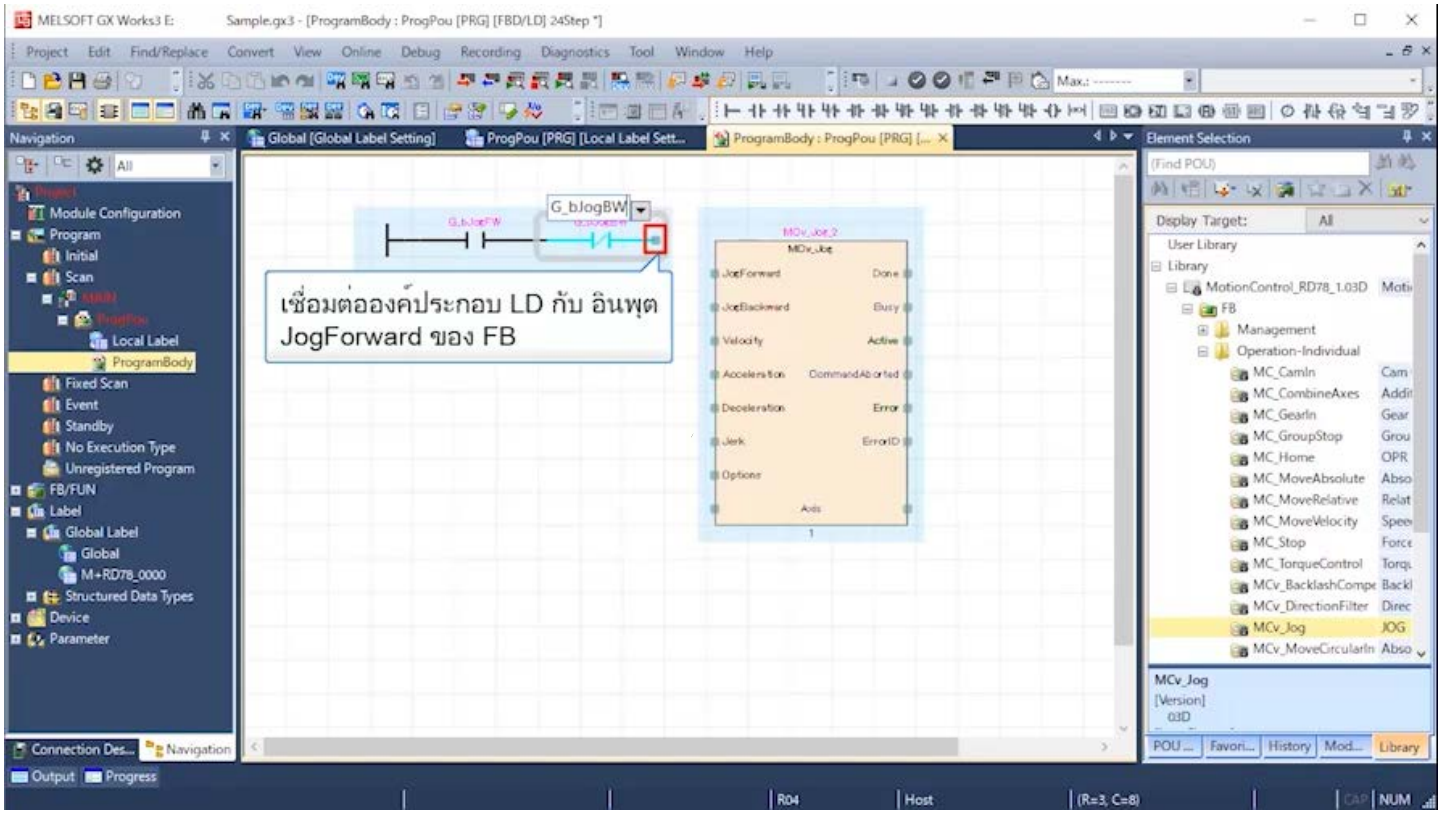
4.3 วิธีใช้ Motion Control FB



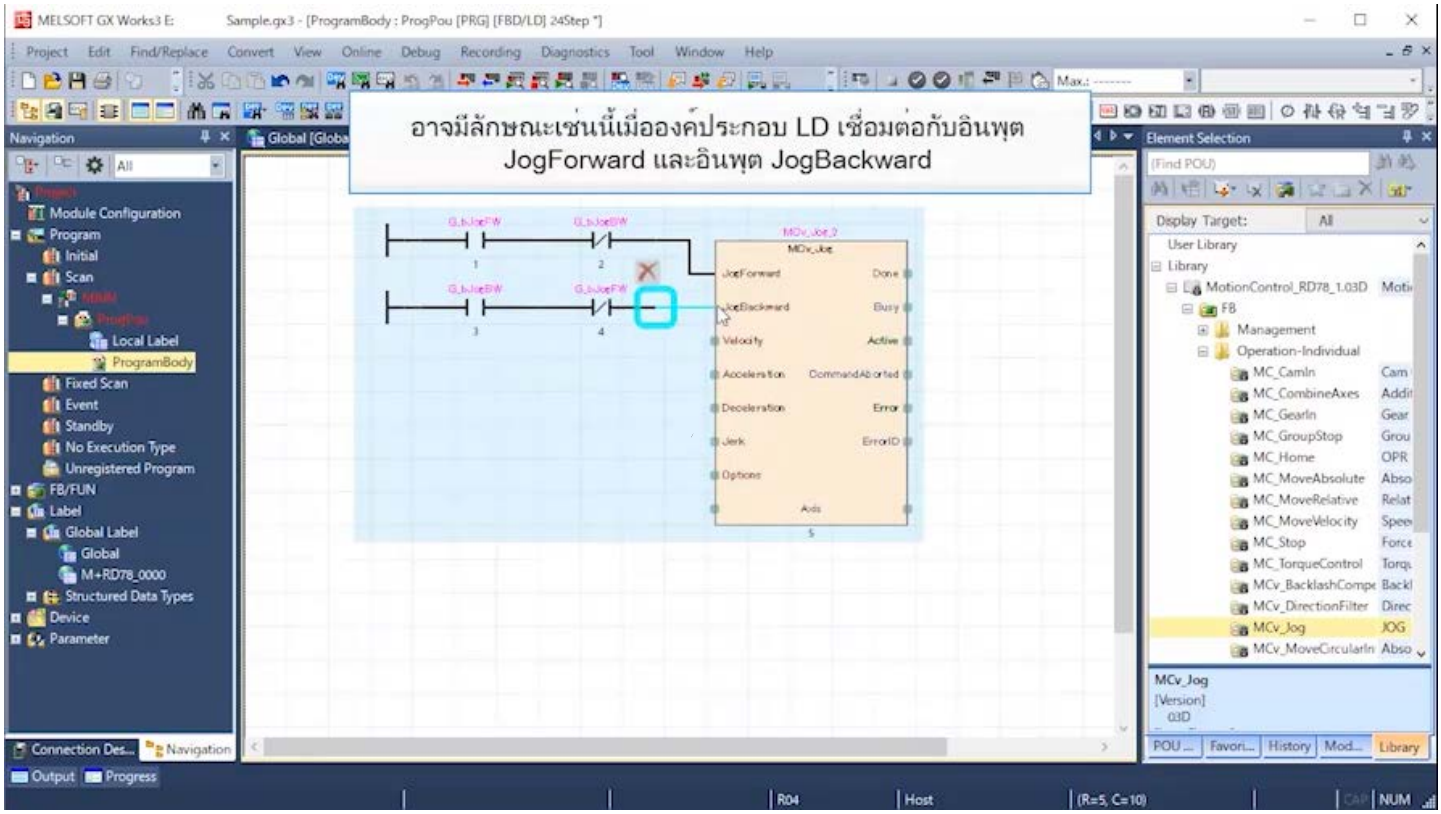
4.3 วิธีใช้ Motion Control FB



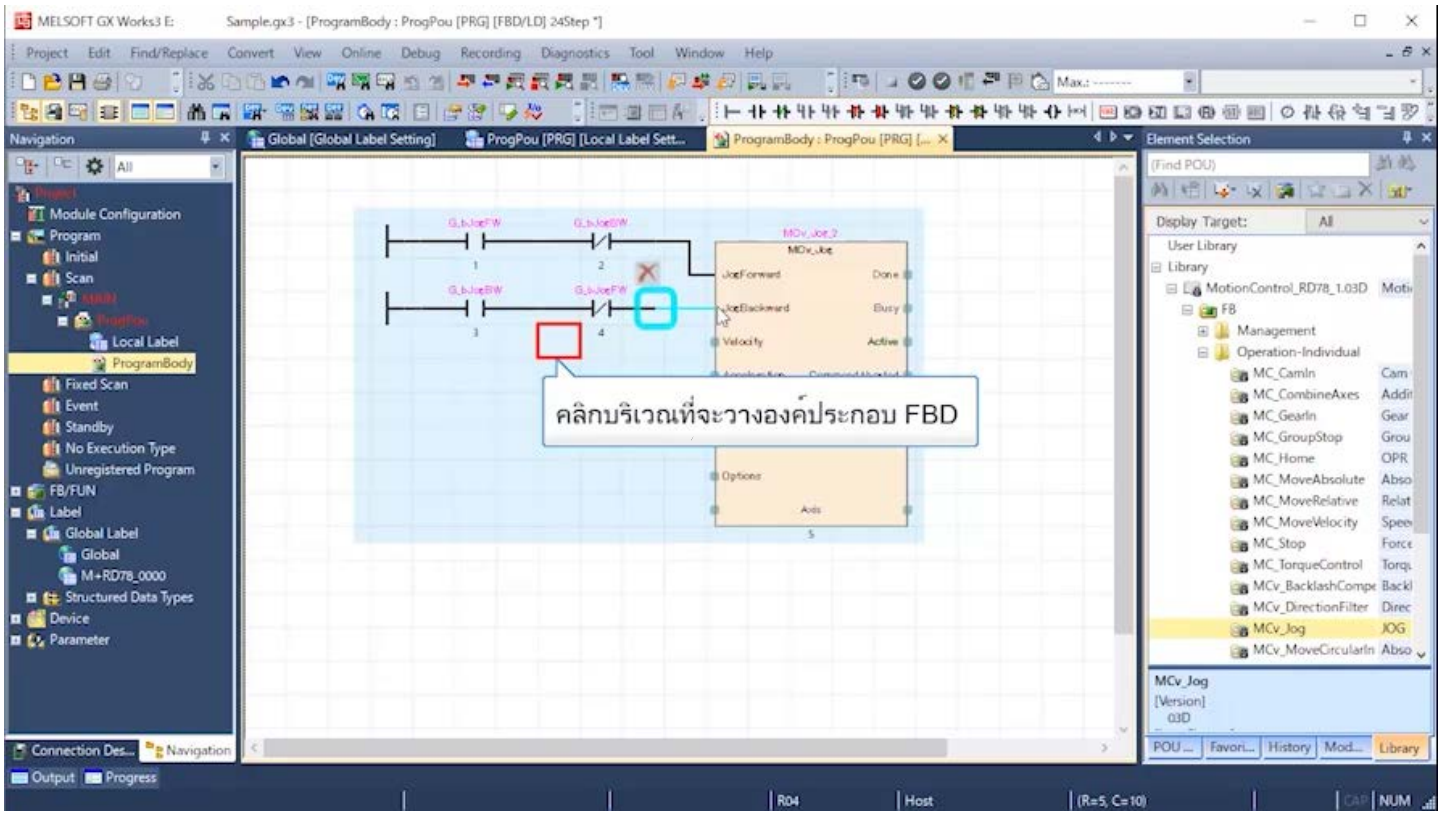
4.3 วิธีใช้ Motion Control FB



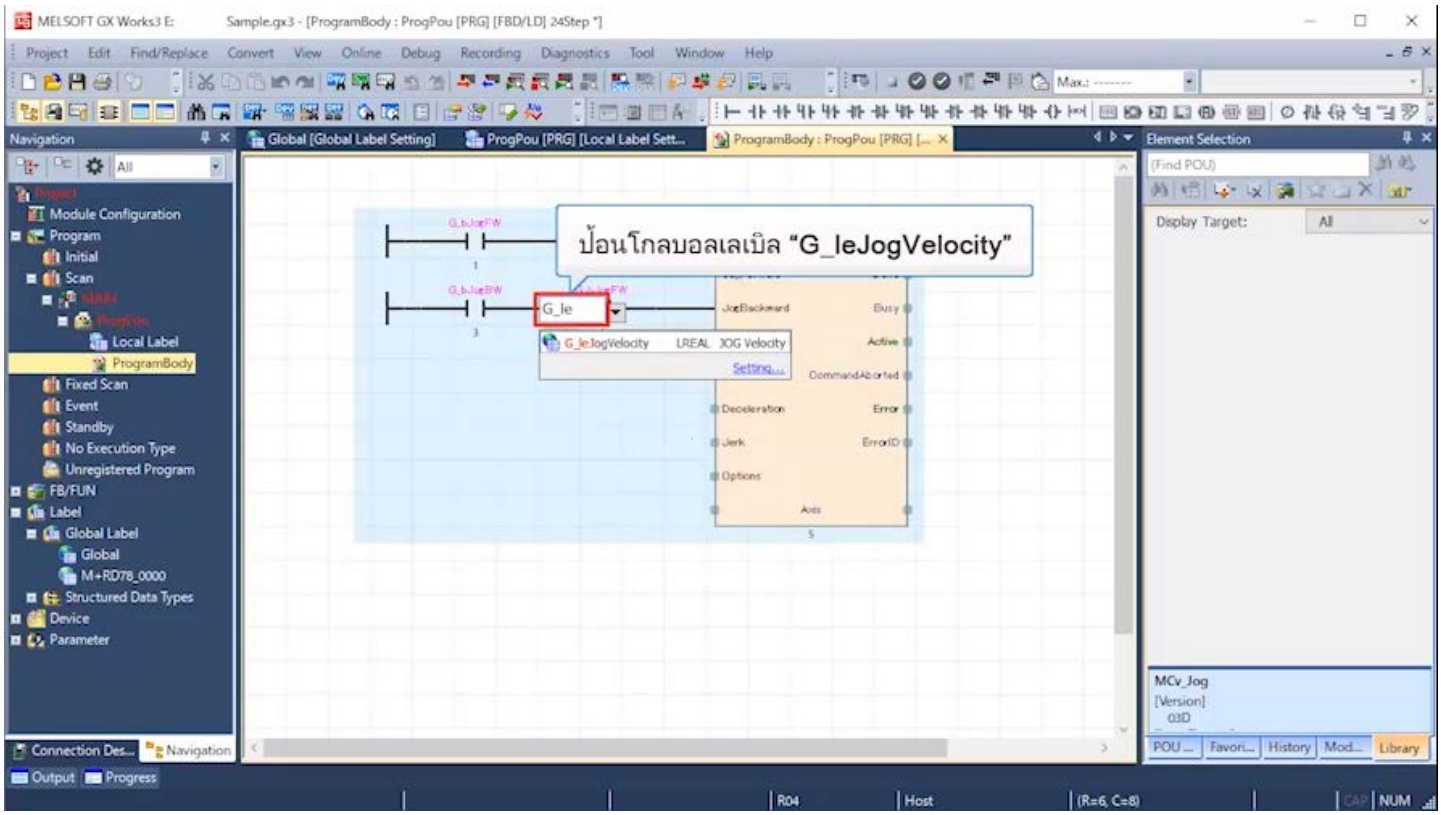
4.3 วิธีใช้ Motion Control FB



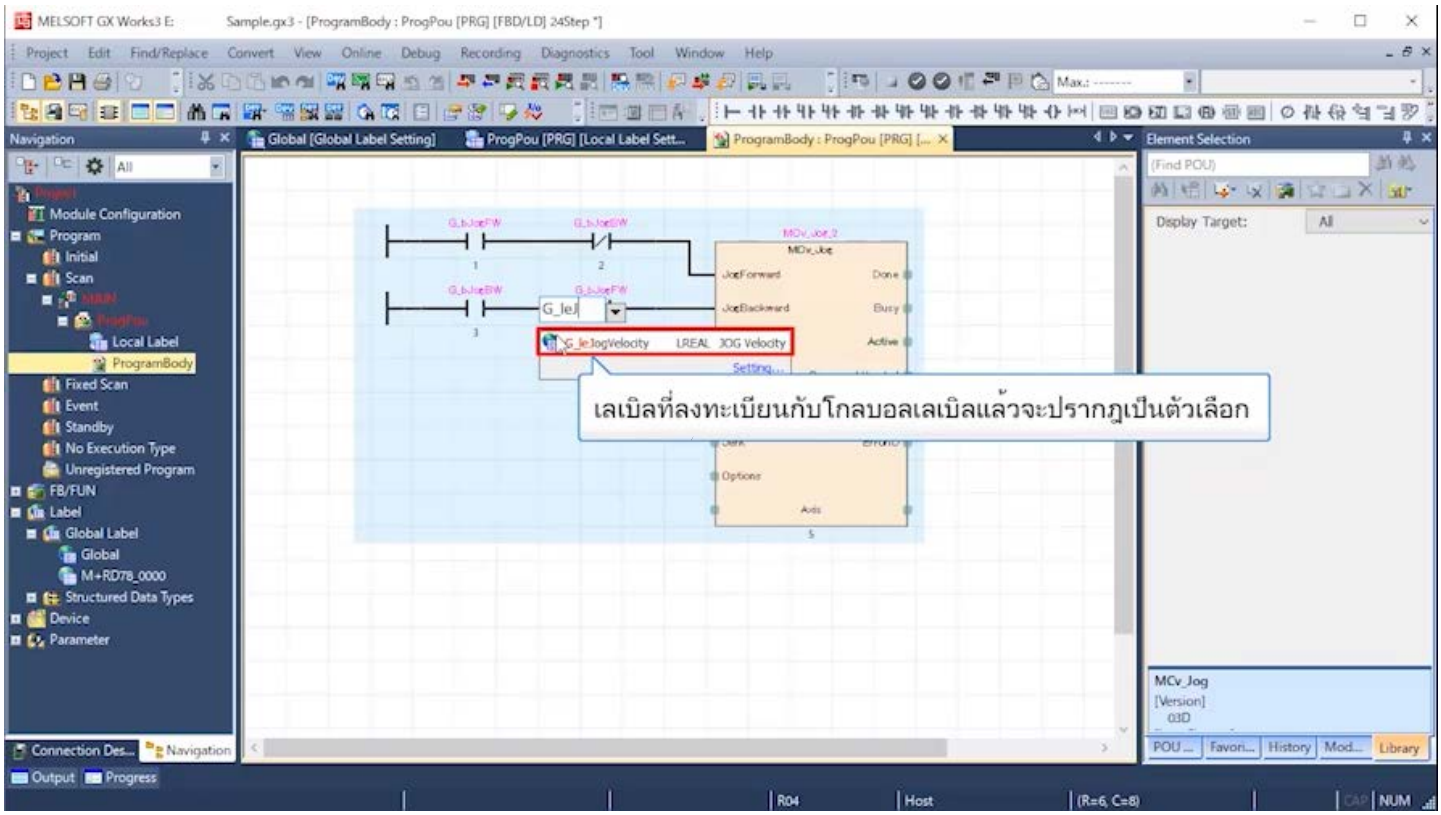
4.3 วิธีใช้ Motion Control FB



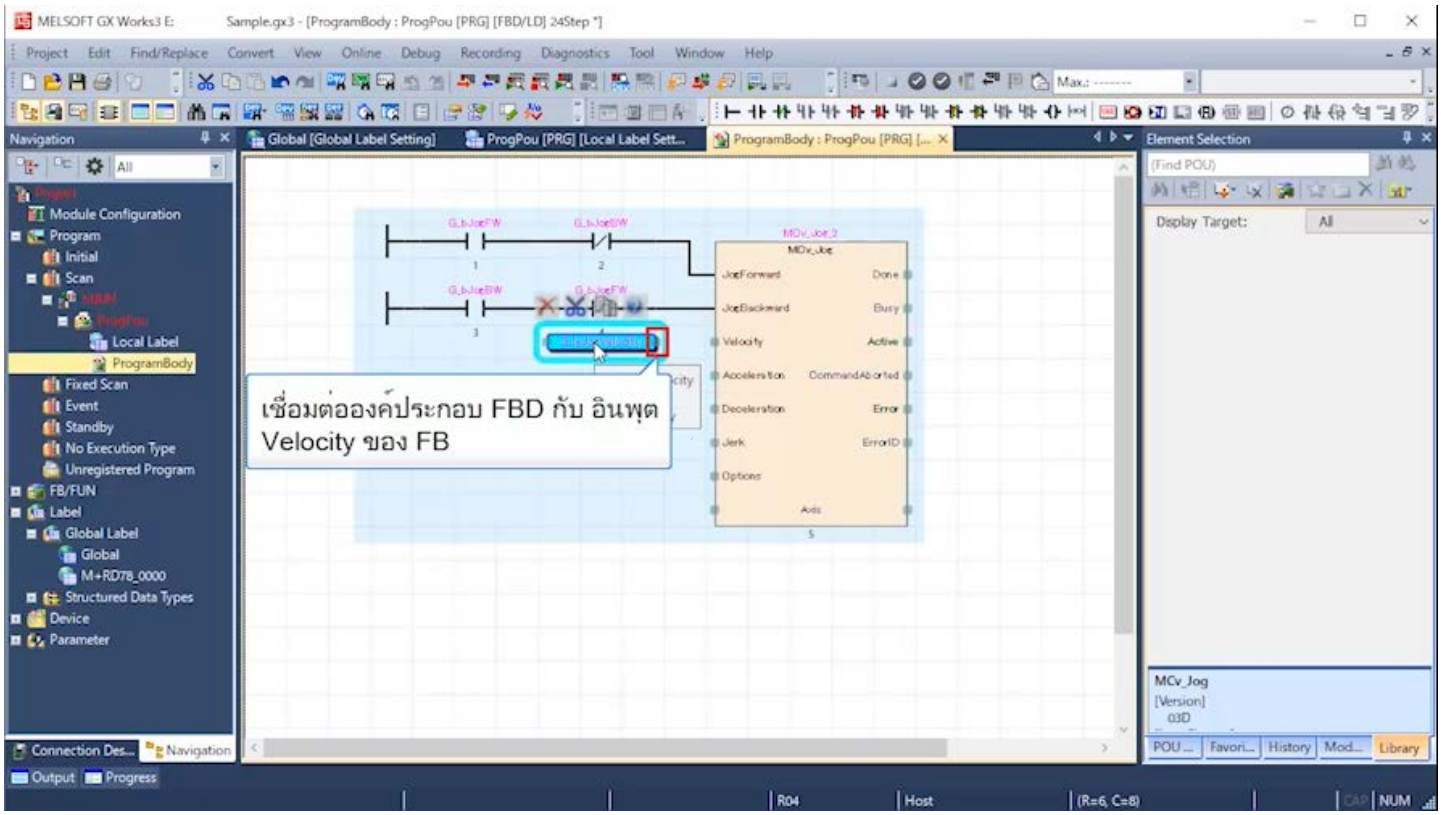
4.3 วิธีใช้ Motion Control FB



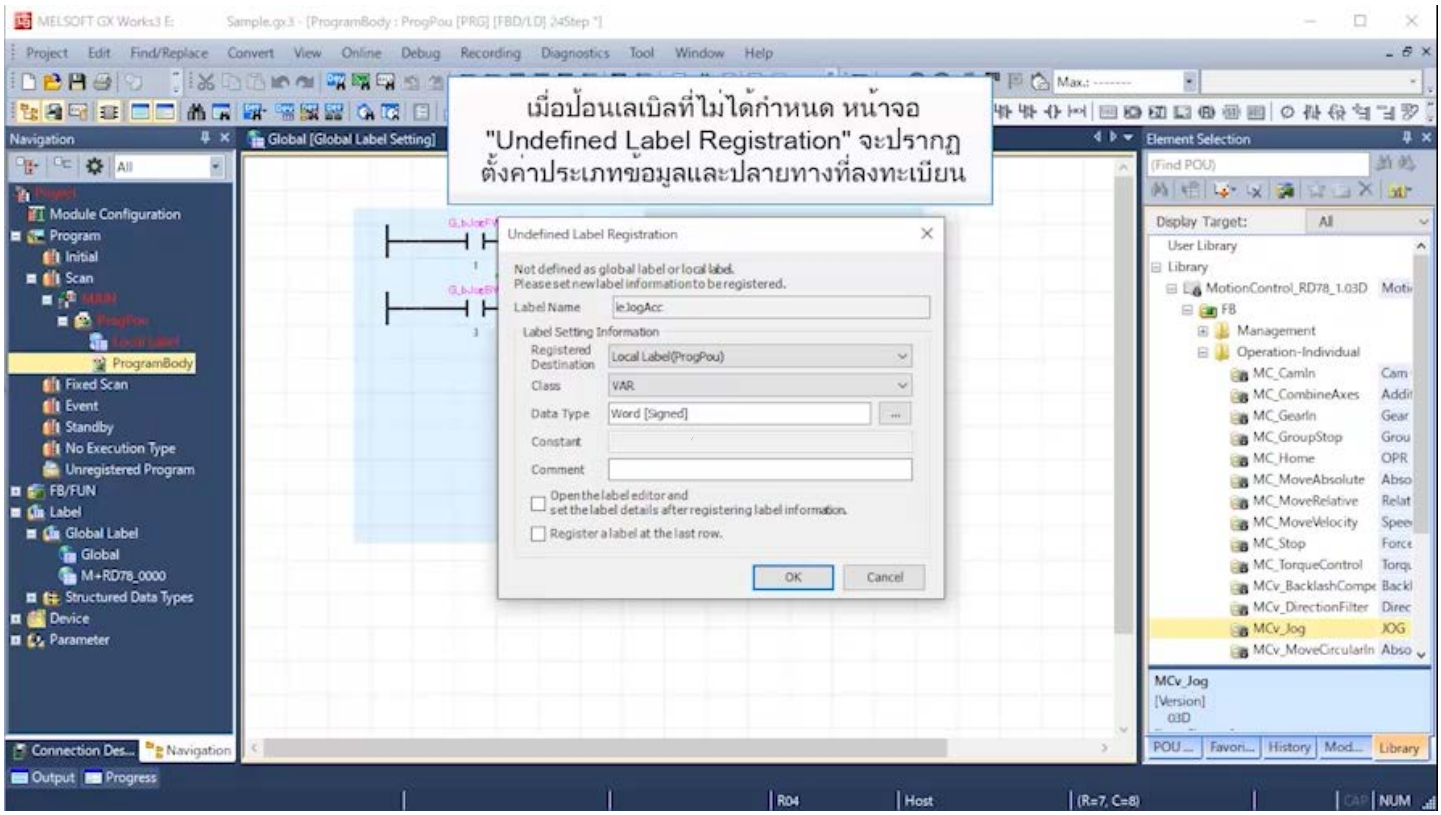
4.3 วิธีใช้ Motion Control FB



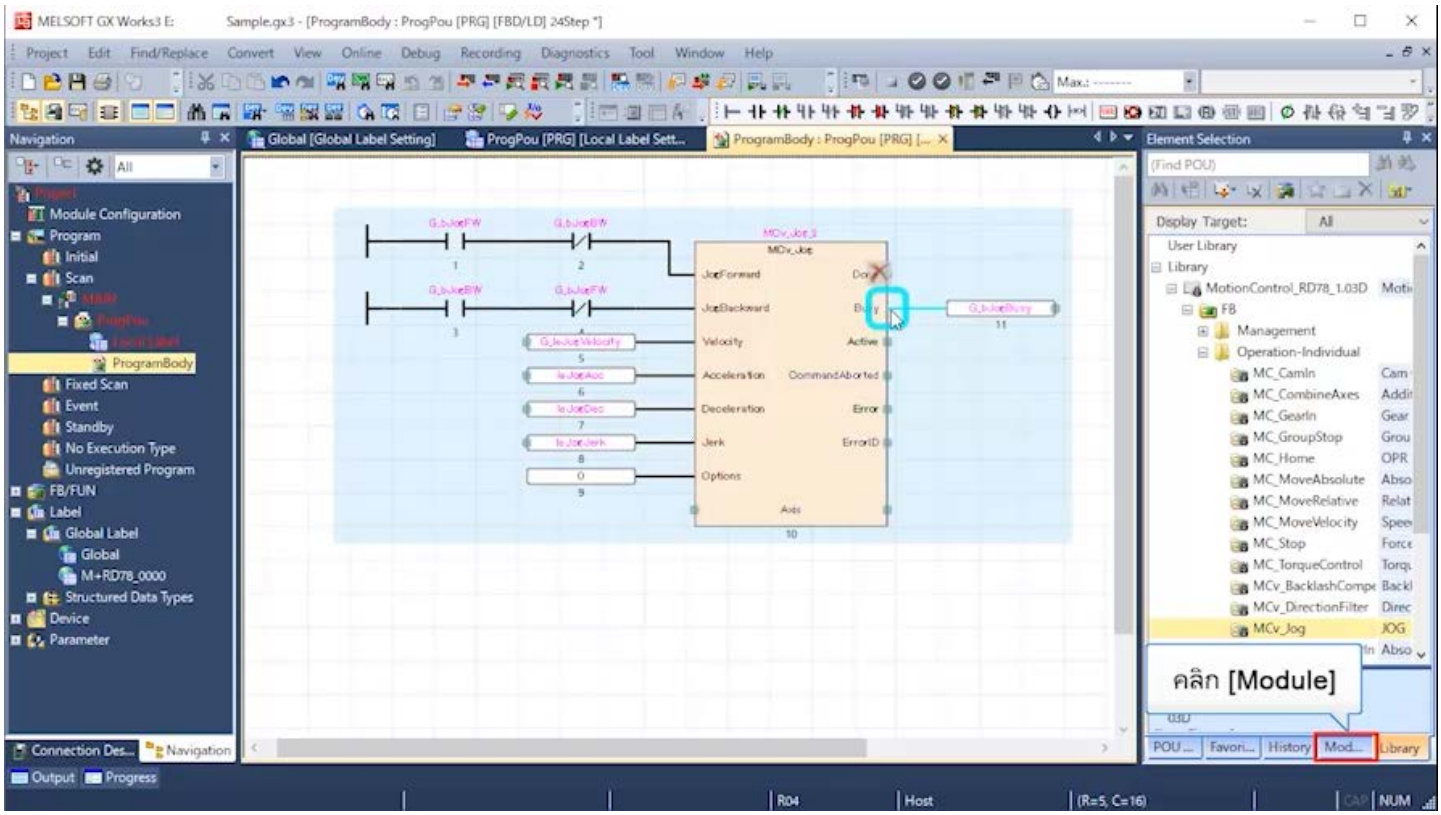
4.3 วิธีใช้ Motion Control FB



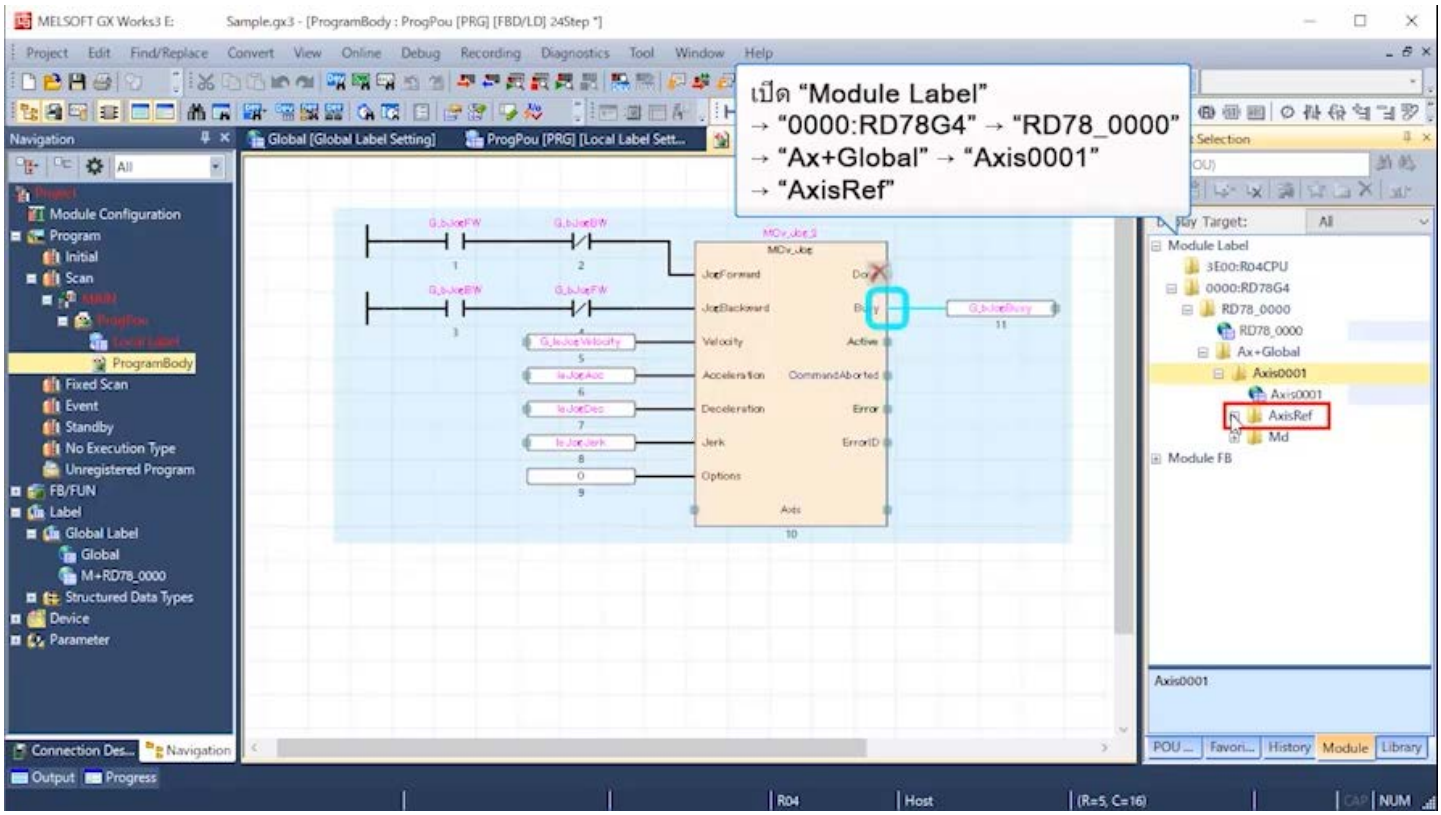
4.3 วิธีใช้ Motion Control FB



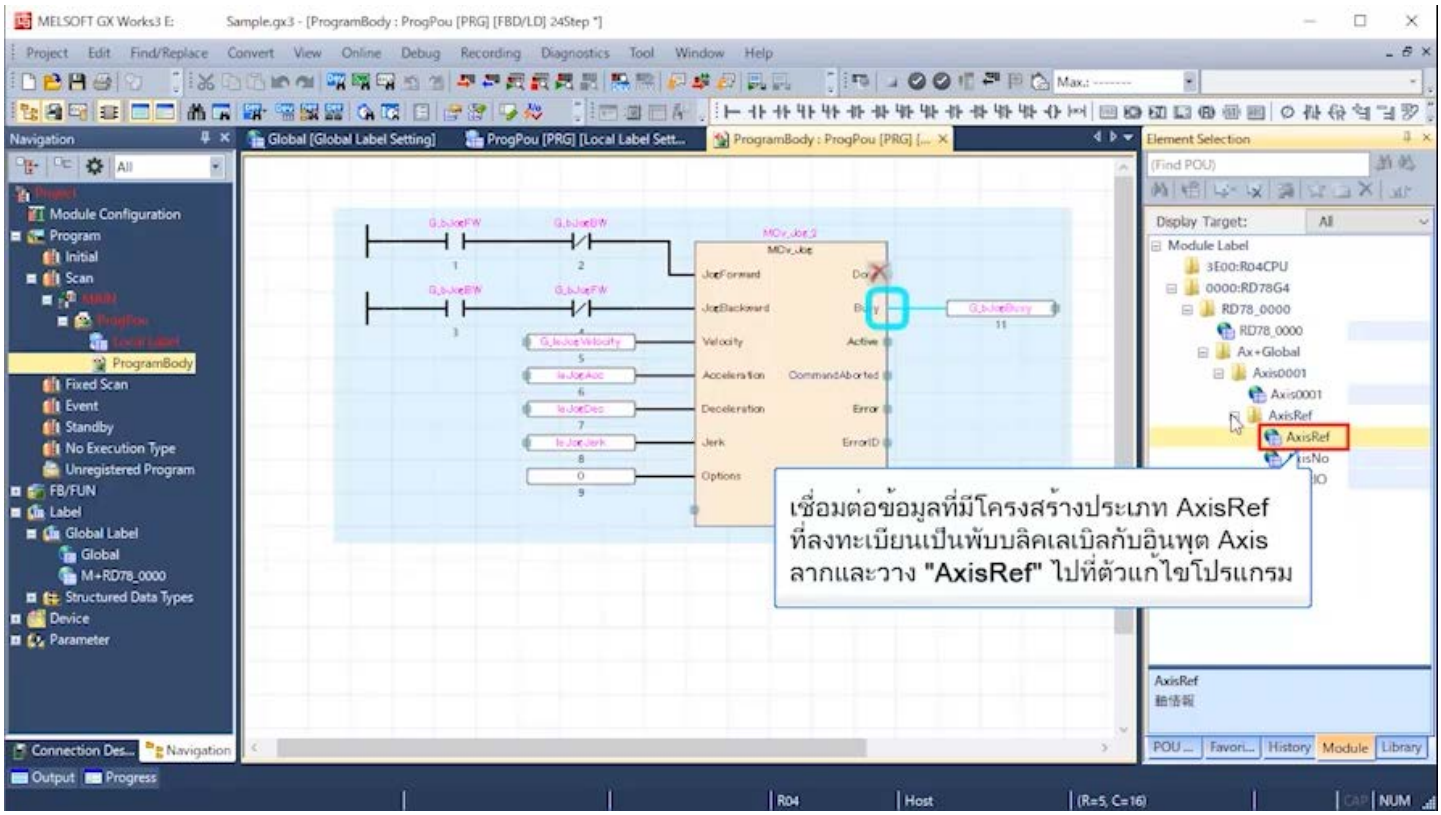
4.3 วิธีใช้ Motion Control FB



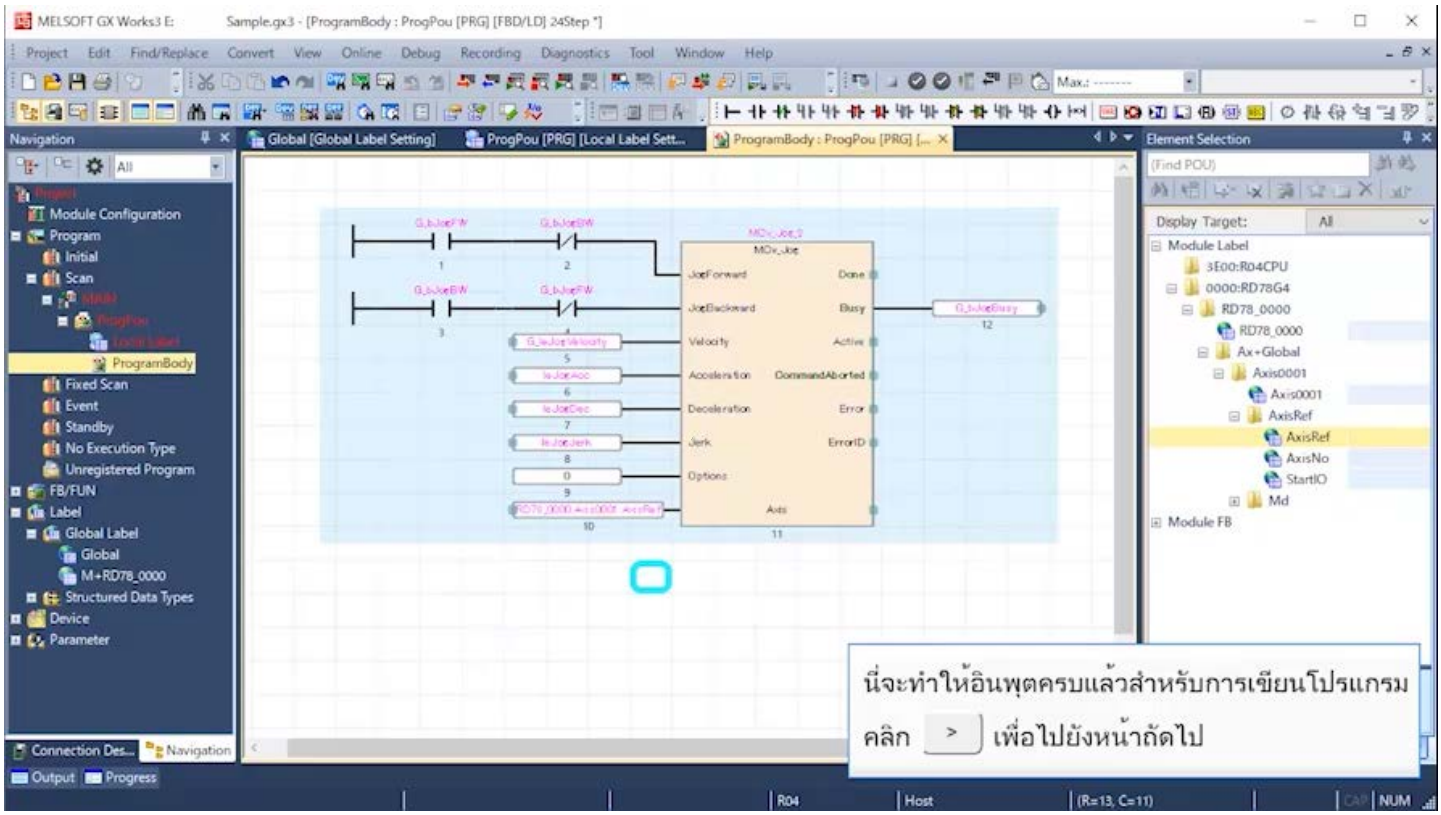
4.3 วิธีใช้ Motion Control FB



4.3 วิธีใช้ Motion Control FB



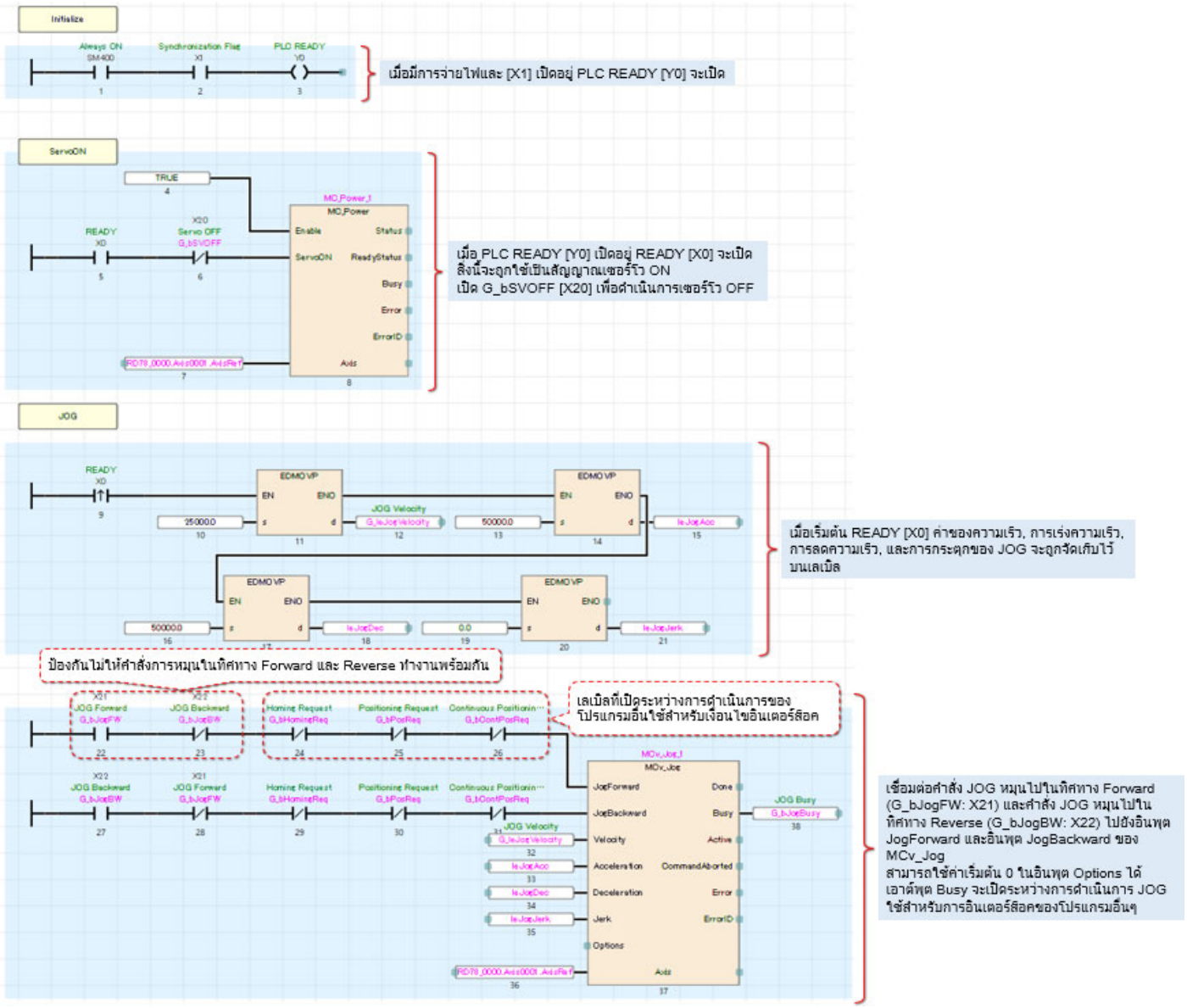
4.3 วิธีใช้ Motion Control FB



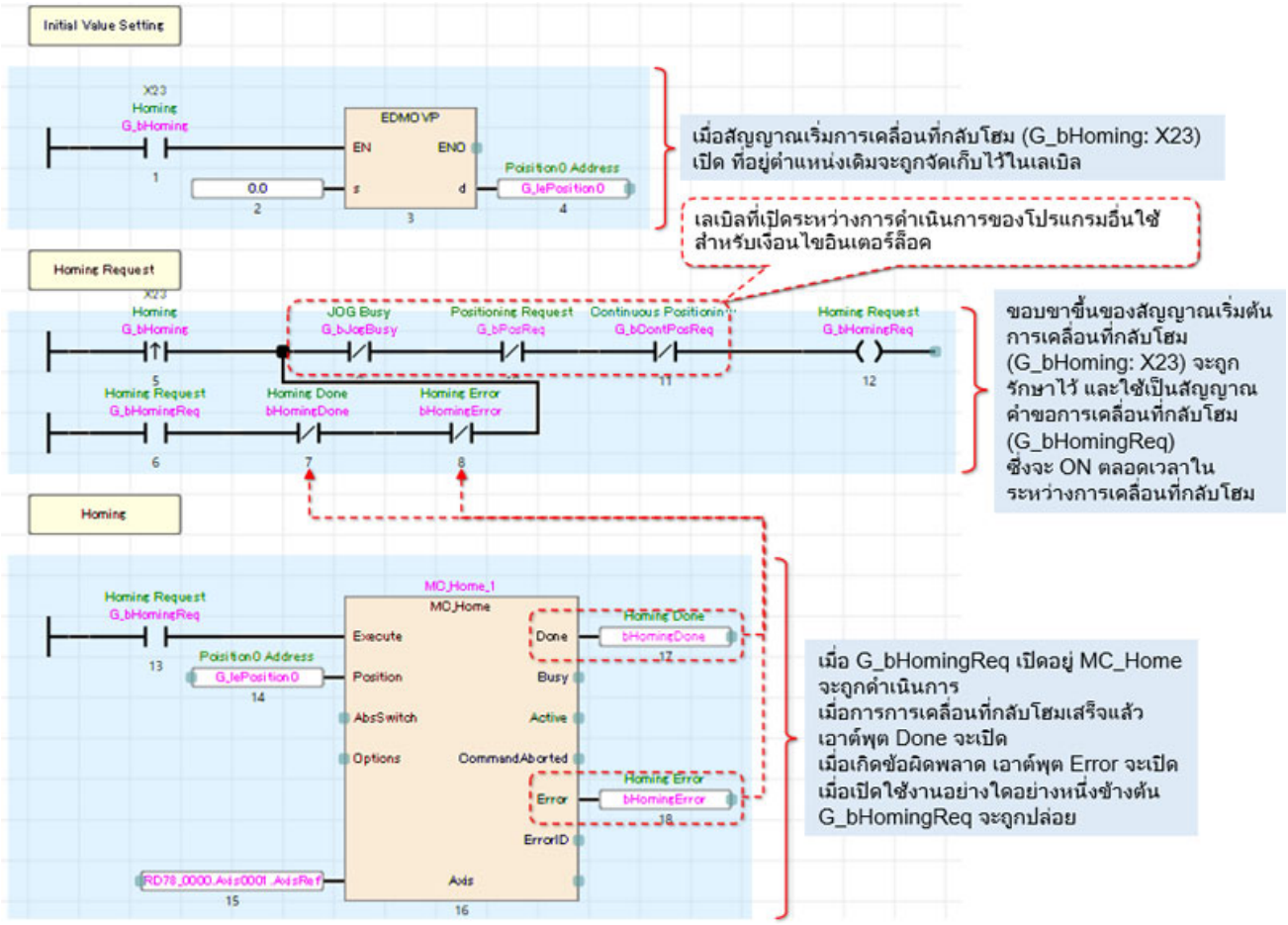
4.4 คำอธิบายโปรแกรมตัวอย่าง

(1) ServoON_Jog

โปรแกรมนี้จะดำเนินการ PLC ready ON, servo ON, and JOG operation

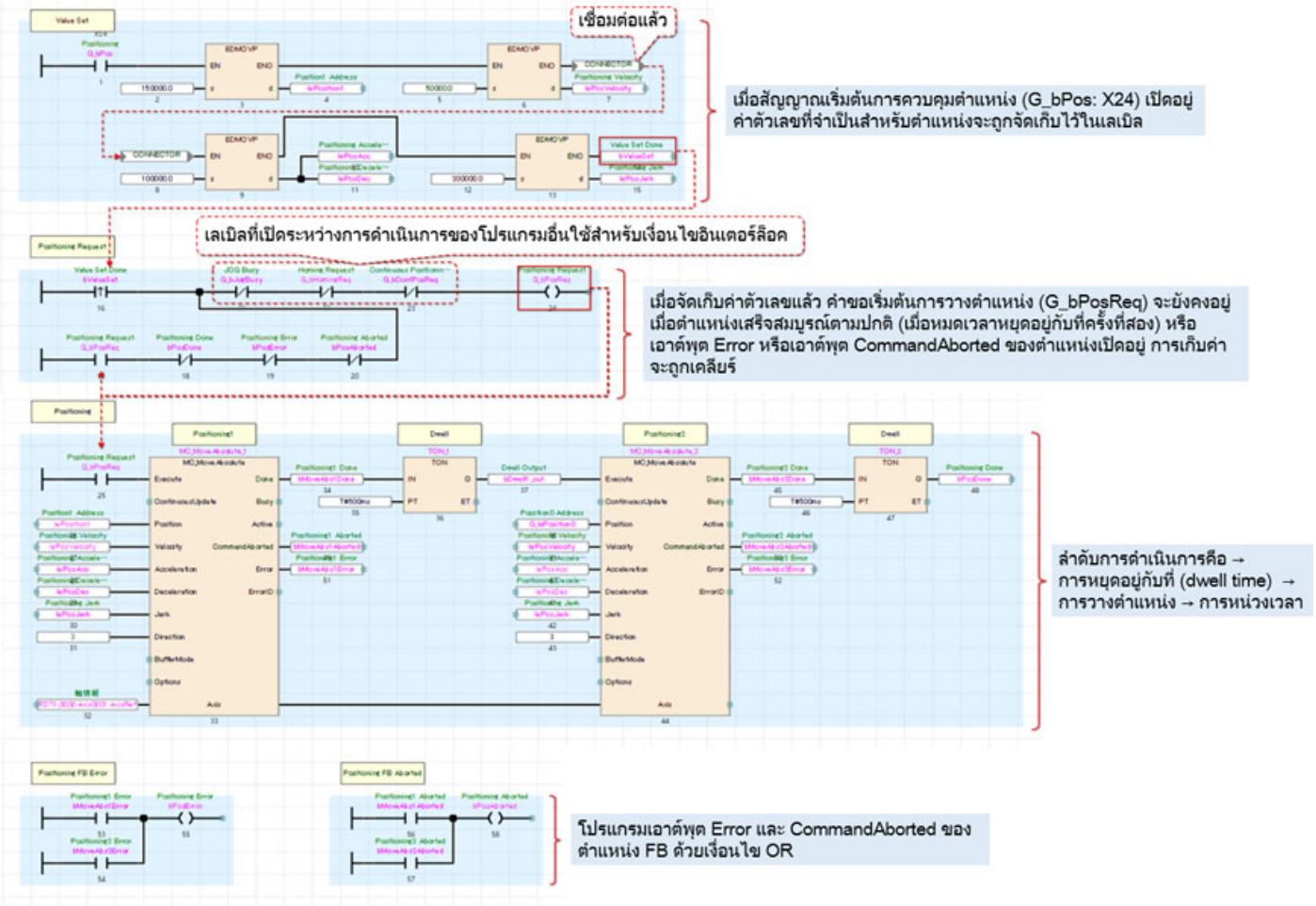


(2) การกลับคืนสู่จุดโฮม
โปรแกรมนี้จะทำการกลับตำแหน่งโฮม



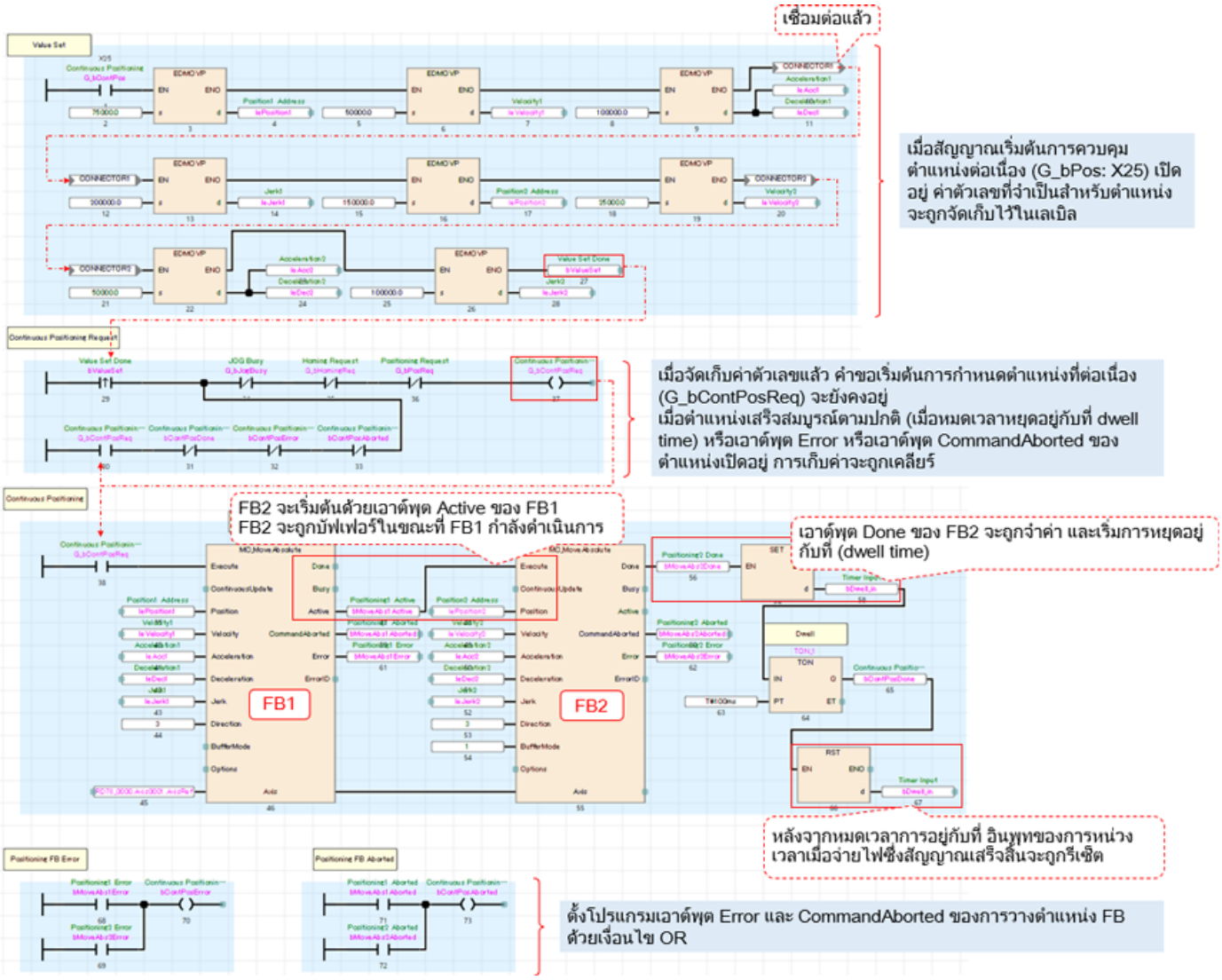
(3) การวางตำแหน่ง

โปรแกรมนี้จะดำเนินการกำหนดตำแหน่งแกนเดียว



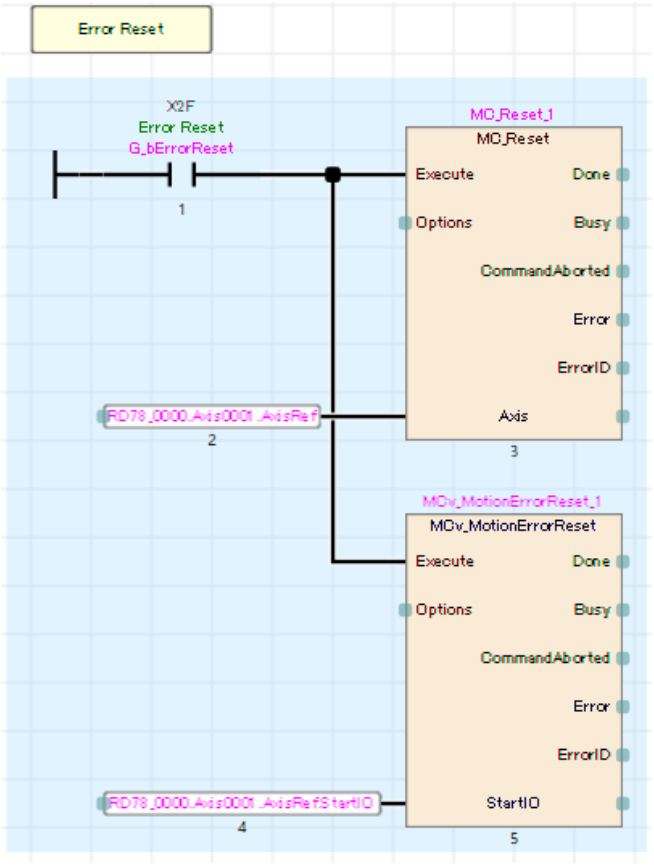
(4) ContinuousPositioning

โปรแกรมนี้จะดำเนินการวางตำแหน่งอย่างต่อเนื่องโดยใช้โหมดบัฟเฟอร์



(5) ErrorReset

โปรแกรมนี้จะทำการรีเซ็ตข้อผิดพลาด



เมื่อสัญญาณรีเซ็ตข้อผิดพลาด (G_bErrorReset: X2F) เปิดอยู่ การรีเซ็ตข้อผิดพลาดของแกน (MC_Reset) และการรีเซ็ตข้อผิดพลาดของระบบ (MCv_MotionErrorReset) จะถูกดำเนินการ

(6) การมอนิเตอร์

SetPosition (ตั้งค่าตำแหน่ง) และ SetVelocity (ตั้งค่าความเร็ว) ของการมอนิเตอร์แกนที่ตั้งเป็นพับบลิคเลเบิล จะถูกจัดเก็บไว้ใน D0 และ D2 เนื่องจาก SetPosition และ SetVelocity เป็นประเภท double precision real number ทั้งสองจะถูกแปลงเป็น signed double word พื่อให้สามารถจัดการได้ง่ายโดย PLC CPU (หมายเหตุ)

อุปกรณ์คำเหล่านี้ไม่ได้ใช้ในหัวข้อ

อุปกรณ์เหล่านี้ใช้เพื่อแสดงบนโปรแกรมลำดับอื่นๆ และ GOT และเพื่อวัตถุประสงค์อื่น โปรแกรมนี้อธิบายด้วย ST

```
1  G_dSetPosition := LREAL_TO_DINT(RD78_0000.Axis0001.Md.SetPosition);
2  G_dSetVelocity := LREAL_TO_DINT(RD78_0000.Axis0001.Md.SetVelocity);
3
```

G_dSetPosition → D0
G_dSetVelocity → D2

(หมายเหตุ) เมื่อแปลงประเภท double precision real number เป็นประเภท signed double word แล้ว ถ้าค่าที่จะแปลงอยู่นอกช่วงตั้งแต่ -2147483648 ถึง 2147483647 จะเกิดข้อผิดพลาดในการคำนวณ

4.5 การเขียนโปรแกรม

เขียนโปรแกรมและพารามิเตอร์ไปยัง PLC CPU และโมดูลโมชั่น
โปรแกรมถูกเขียนไปยังโมดูล CPU เท่านั้น จำเป็นต้องเขียนพารามิเตอร์แกนและพับบลิค เลเบิลไปที่โมดูลโมชั่น

1) หลังจากที่โปรแกรมทั้งหมดใน PLC CPU ถูกrebuiltแล้ว เลือก [Online] → [Write to PLC] ในแถบเครื่องมือของ GX Works3 เพื่อเขียนข้อมูลทั้งหมดไปยัง PLC CPU

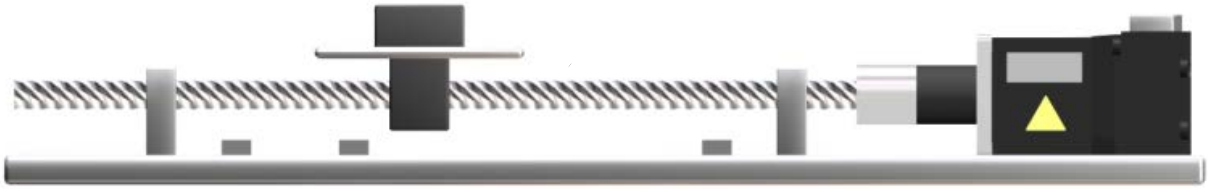
2) เมื่อเขียนพารามิเตอร์ไปยัง PLC CPU การสื่อสารกับโมดูลโมชั่นจะถูกเปิดใช้งาน เลือก [Online] → [Write to Module] ในแถบเครื่องมือของ Motion Control Setting Function เพื่อเขียนข้อมูลทั้งหมดไปยังโมดูลโมชั่น

3) รีเซ็ต PLC CPU เพื่อสิ้นสุดการดำเนินการเขียน

4.6 ตัวอย่างการตรวจสอบการดำเนินการของโปรแกรม

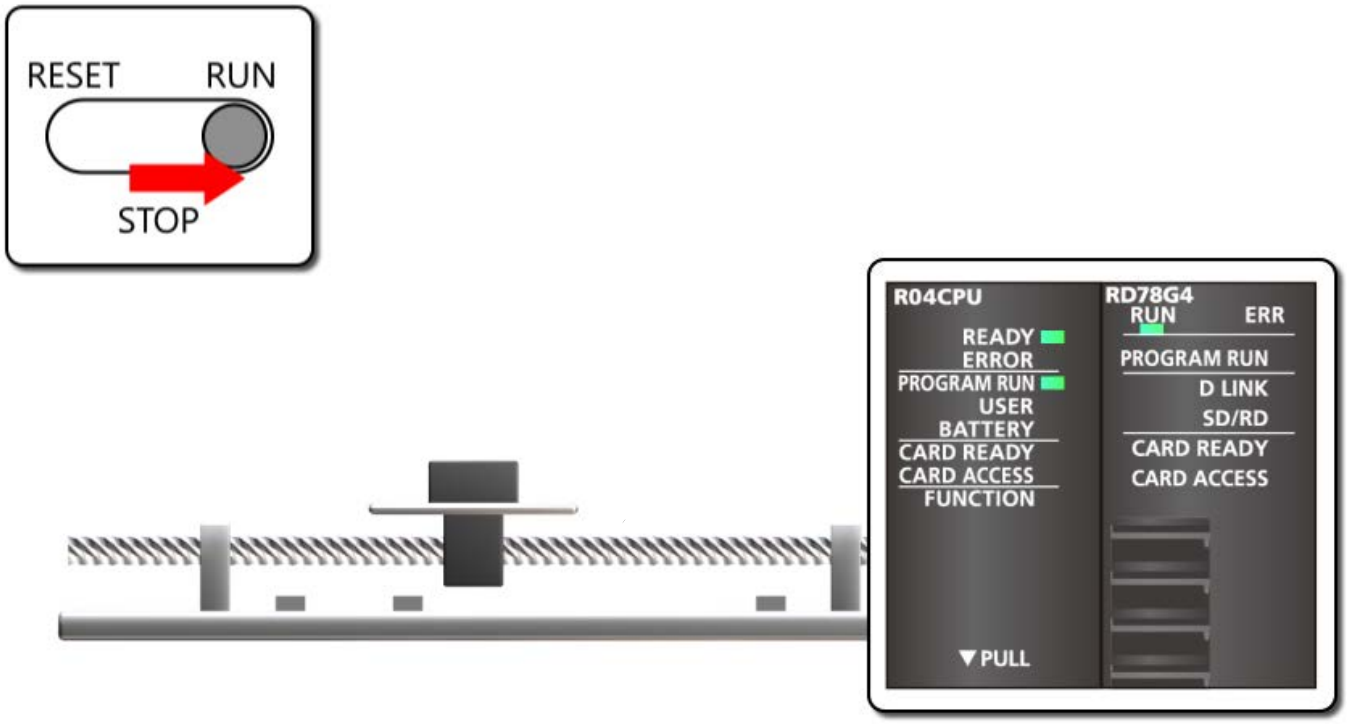
คลิกปุ่มเล่นที่ด้านซ้ายล่างของหน้าต่าง

4.6 ตัวอย่างการตรวจสอบการดำเนินการของโปรแกรม



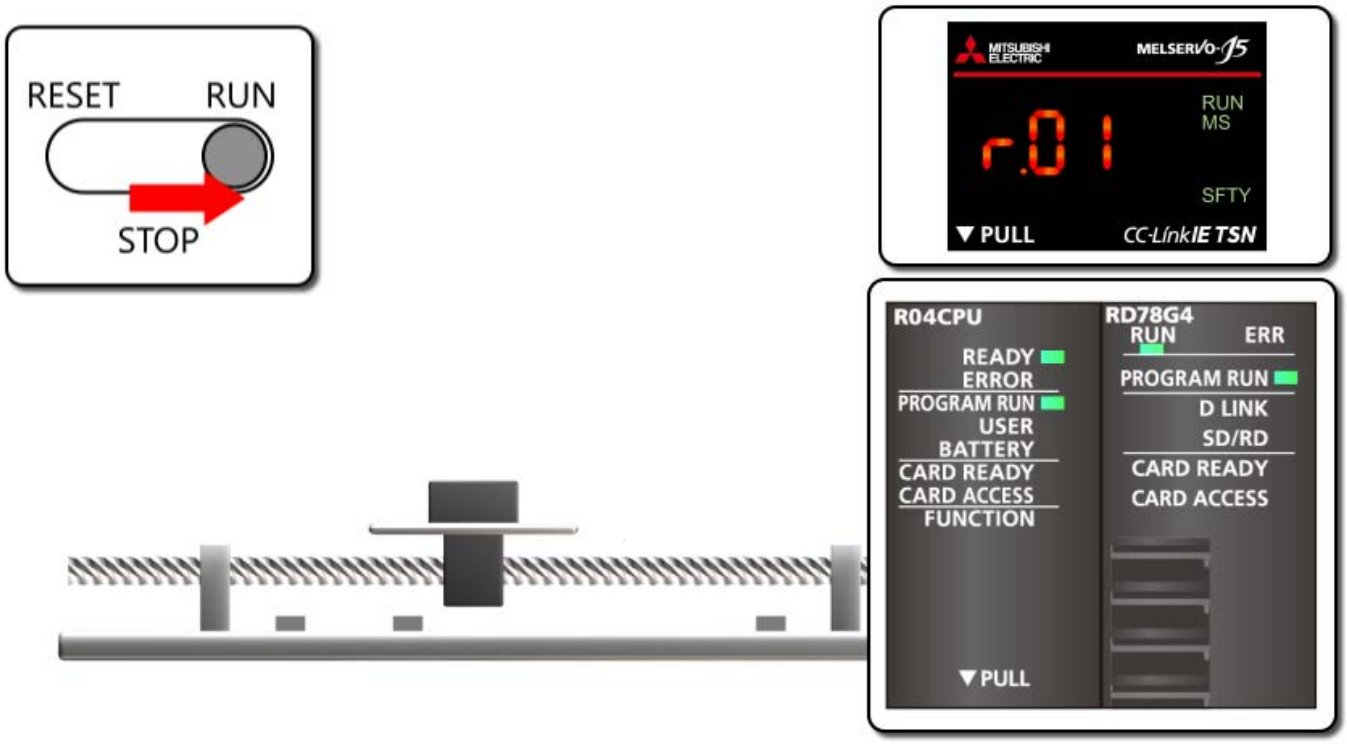
ตรวจสอบการดำเนินการของโปรแกรมตัวอย่าง
ก่อนเริ่มการดำเนินการ ตรวจสอบให้แน่ใจว่าโปรแกรม
และพารามิเตอร์ถูกเขียนไปยัง PLC CPU และโมดูลโมชั่น

4.6 ตัวอย่างการตรวจสอบการดำเนินการของโปรแกรม



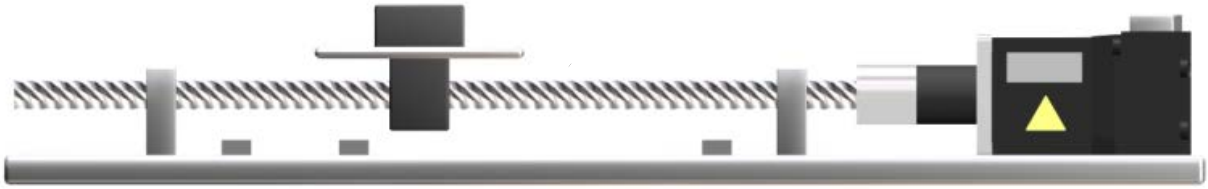
ตั้งค่าสวิตช์ RUN/STOP/RESET ของ PLC CPU เป็น RUN
ไฟ READY และไฟ PROGRAM RUN ของ PLC CPU จะติด
ไฟ RUN ของโมดูลโมชั่นจะติด

4.6 ตัวอย่างการตรวจสอบการดำเนินการของโปรแกรม



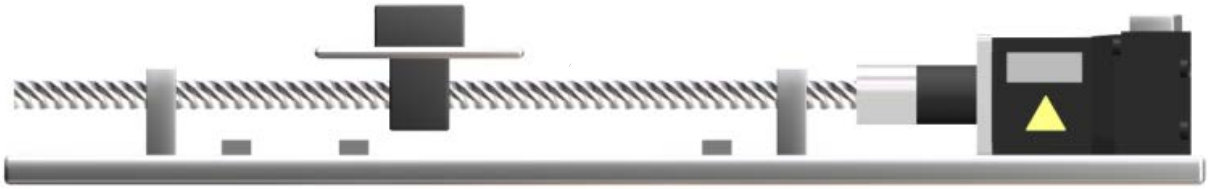
รอจนกว่าไฟ PROGRAM RUN ของโมดูลโมชั่นจะติด
"r.01" จะแสดงบนเซอร์โวแอมพลิฟายเออร์ (จุดจะติดค้าง)
เซอร์โวมอเตอร์จะเข้าสู่สถานะเซอร์โว ON

4.6 ตัวอย่างการตรวจสอบการดำเนินการของโปรแกรม



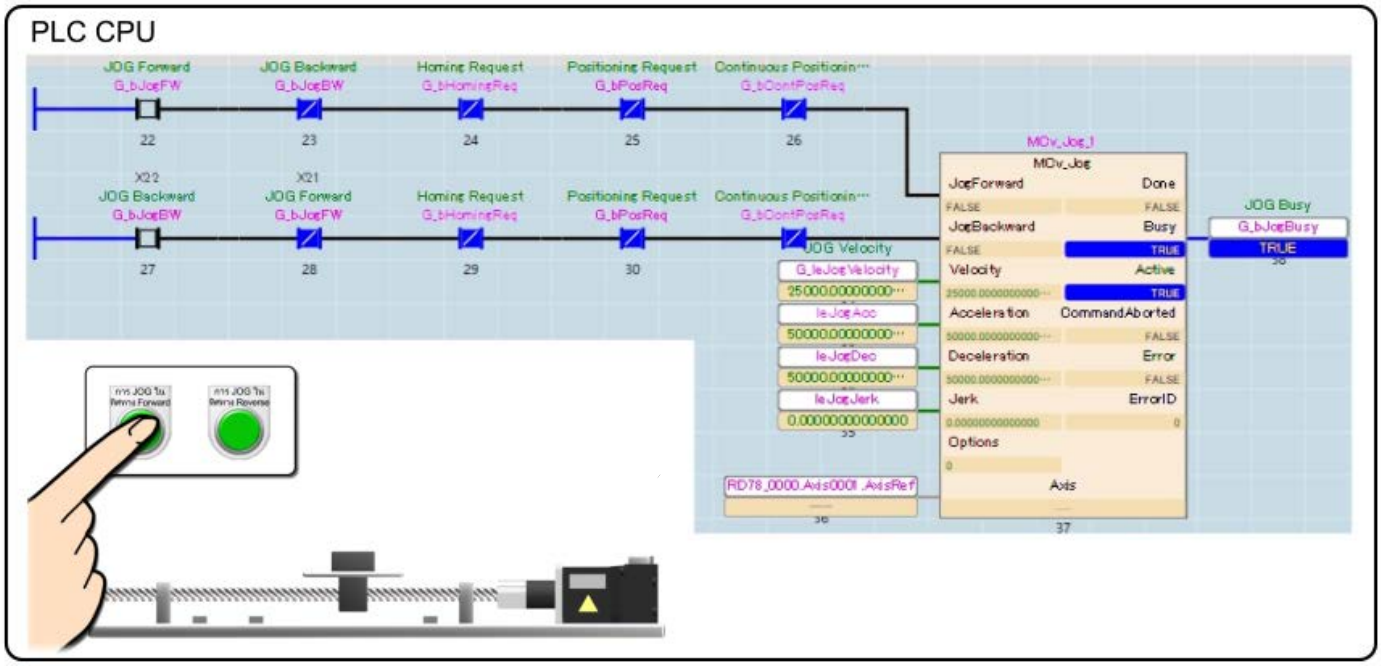
เปิด X20 เพื่อดำเนินการเซอร์โว OFF
"r.01" จะแสดงบนเซอร์โวแอมพลิฟายเออร์ (จุดจะกระพริบ)
ปิด X20 เพื่อดำเนินการเซอร์โว ON อีกครั้ง

4.6 ตัวอย่างการตรวจสอบการดำเนินการของโปรแกรม



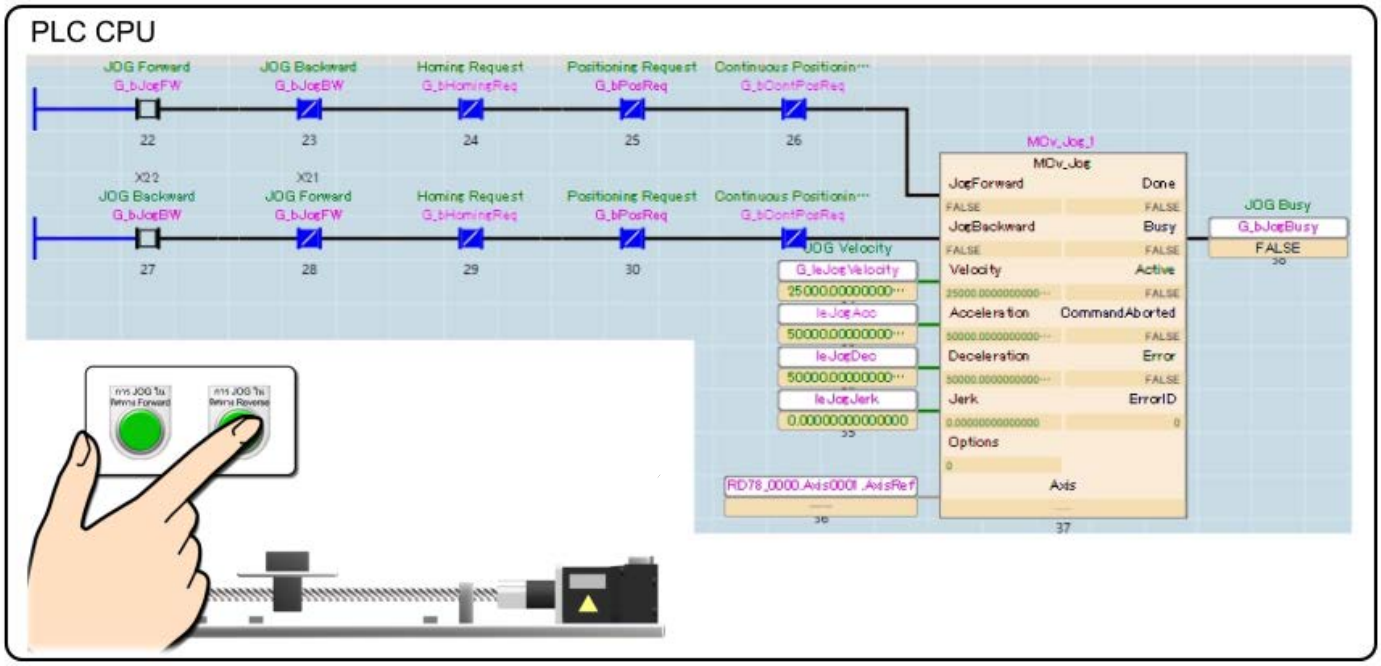
เปิดการหมุน JOG ไปทิศทาง Forward (X21)
address จะมีตัวเลขที่เพิ่มขึ้น และปิดเพื่อหยุด
เปิดการหมุน JOG ไปทิศทาง Reverse (X22)
address จะมีตัวเลขที่ลดลง และปิดเพื่อหยุด

4.6 ตัวอย่างการตรวจสอบการดำเนินการของโปรแกรม

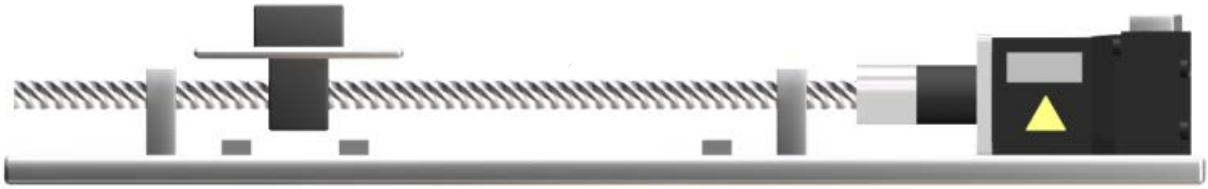


ตรวจสอบโดยการมอนิเตอร์โปรแกรม
เมื่อ X21 เปิดอยู่ อินพุต JogForward ของ MCv_Jog_1 จะเปิด
ดำเนินการหมุนตามปกติของ JOG
เอาต์พุต Busy และ "G_bJogBusy" จะเปิดระหว่างการดำเนินการ

4.6 ตัวอย่างการตรวจสอบการดำเนินการของโปรแกรม

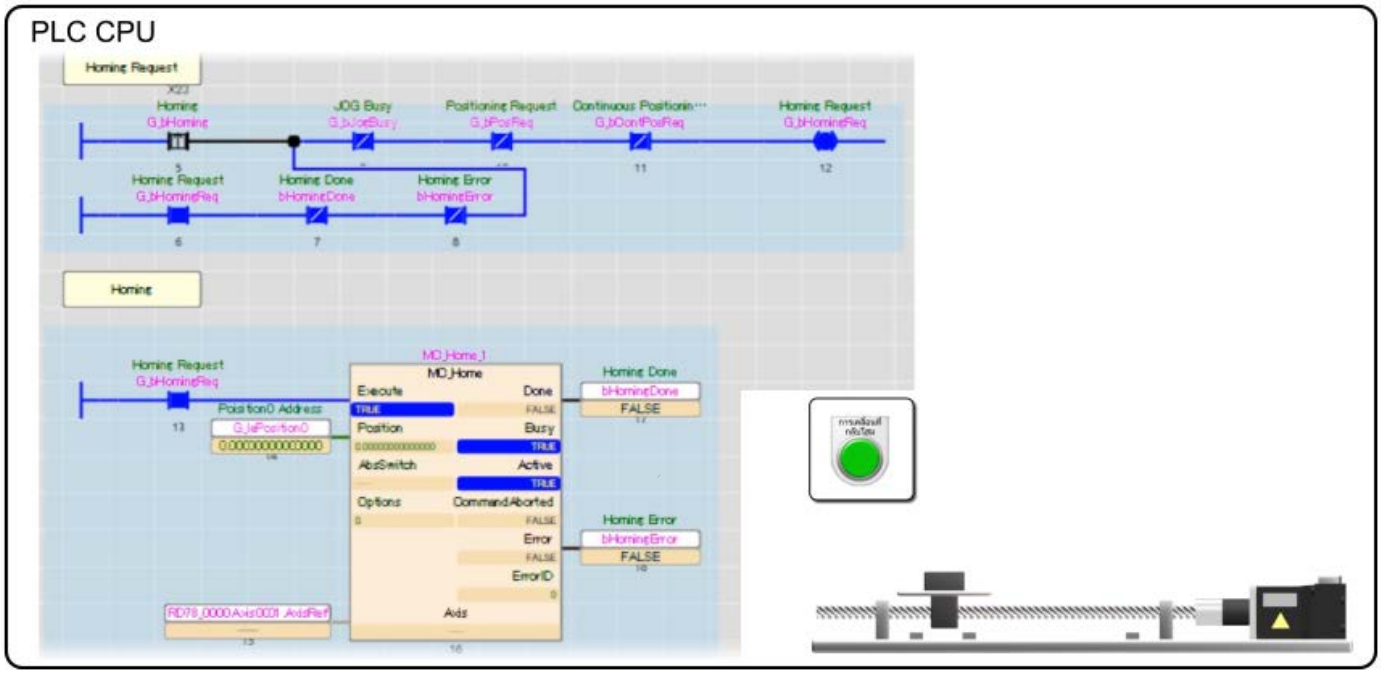


เมื่อ X22 เปิดอยู่ อินพุต JogBackward ของ MCv_Jog_1 จะเปิด
การหมุนในทิศทาง Reverse
เอาต์พุต Busy และ "G_bJogBusy" จะเปิดระหว่างการดำเนินการ



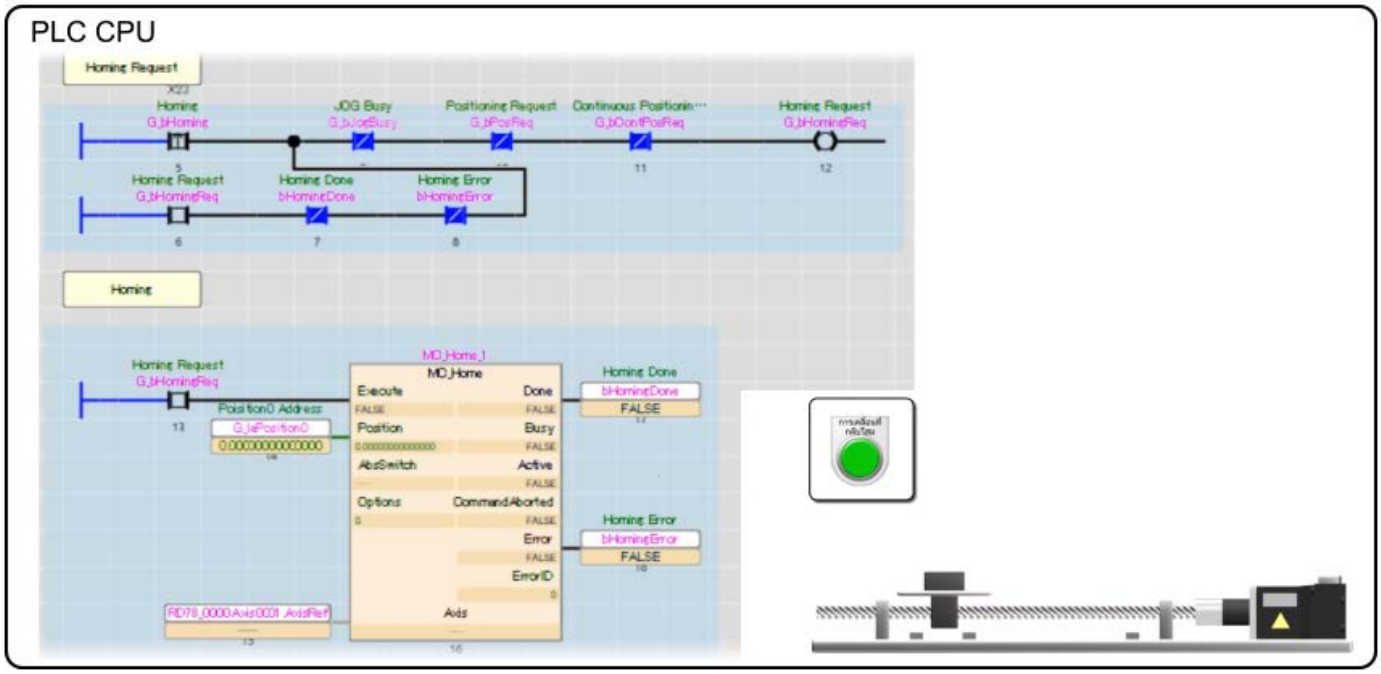
เปิดการเคลื่อนที่กลับโฮม (X23) เพื่อเริ่มการทำโฮม
ดำเนินการเคลื่อนที่กลับโฮมด้วยวิธี proximity dog (ลบ 33 ออกจาก Pr.PT45)
แกนจะหยุดเลย dog ไปเล็กน้อย และจะตั้งค่าจุดนั้นเป็นตำแหน่งเดิม

4.6 ตัวอย่างการตรวจสอบการดำเนินการของโปรแกรม

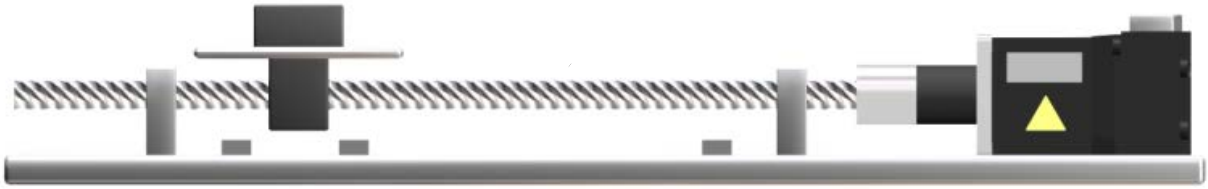


ตรวจสอบโดยการมอนิเตอร์โปรแกรม
เมื่อ X23 เปิด ตำแหน่งโฮมจะถูกจัดเก็บไว้ในเลเบิล
"G_bHomingReq" ซึ่งเป็นคำสั่งการดำเนินการของ MC_Home_1
จะเปิดและถูกคงค่าไว้

4.6 ตัวอย่างการตรวจสอบการดำเนินการของโปรแกรม

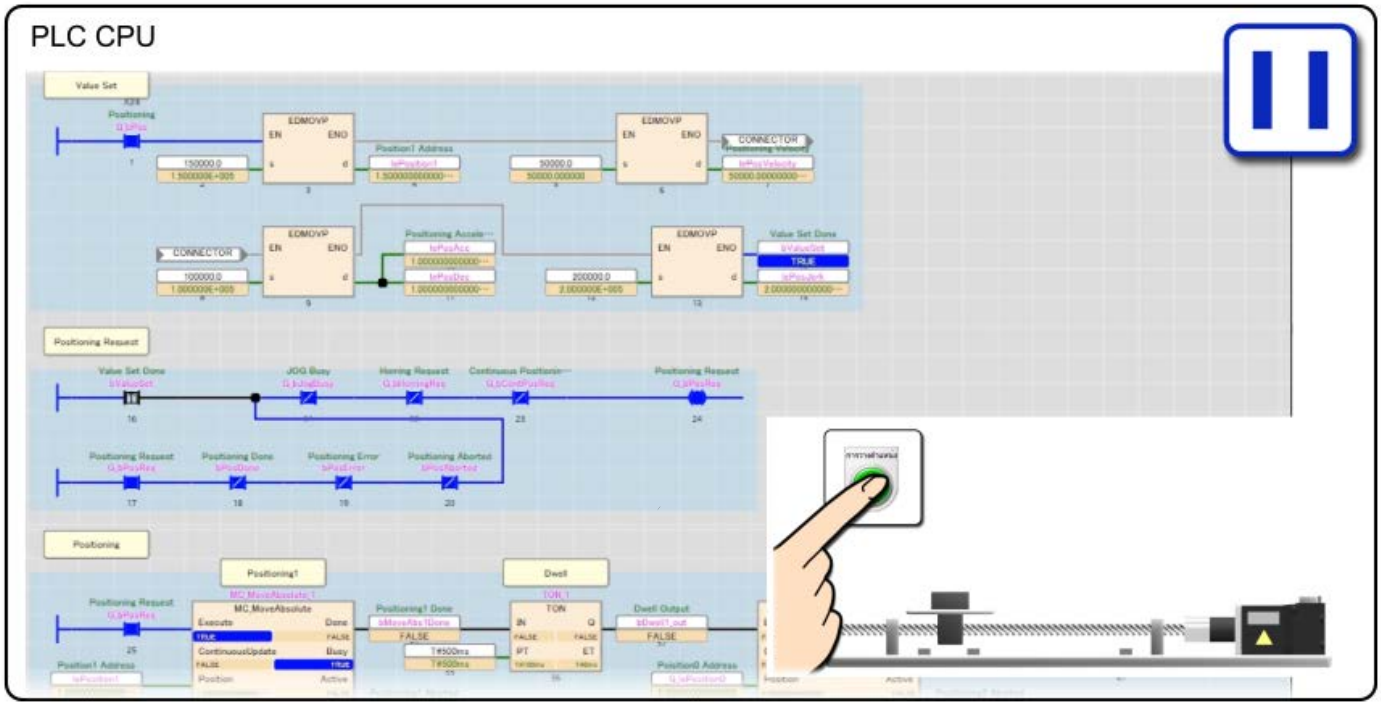


การเคลื่อนที่กลับโฮมจะเริ่มต้น
เมื่อการเคลื่อนที่กลับโฮมเสร็จแล้ว เอาต์พุต Done และ "bHomingDone"
จะเปิด และการคงค่า "G_bHomingReq" จะถูกยกเลิก



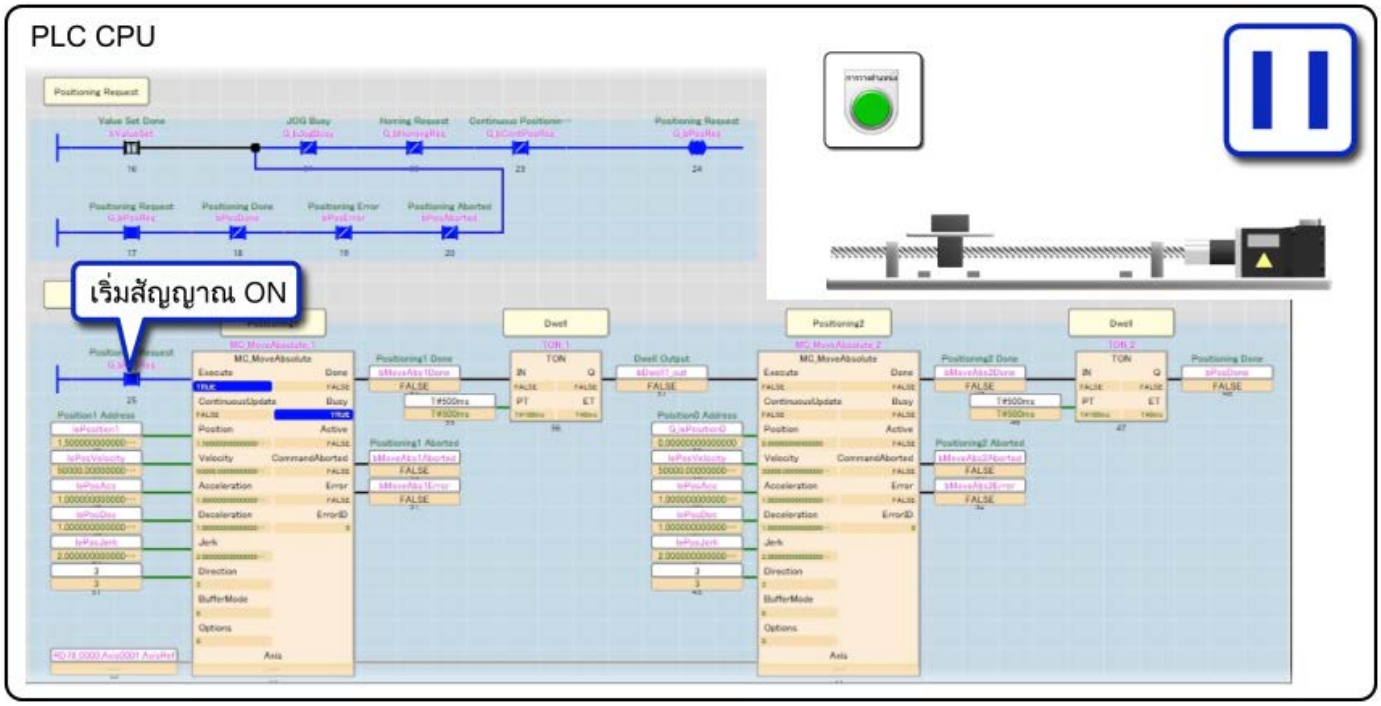
การเปิดการเริ่มตำแหน่ง (X24) จะเริ่มการเคลื่อนที่ขึ้นลง
แกนจะเคลื่อนที่ไปข้างหน้า 150 มม. และหยุดเป็นเวลา 0.5 วินาที
และเคลื่อนที่กลับไป 150 มม. และหยุดเป็นเวลา 0.5 วินาที

4.6 ตัวอย่างการตรวจสอบการดำเนินการของโปรแกรม



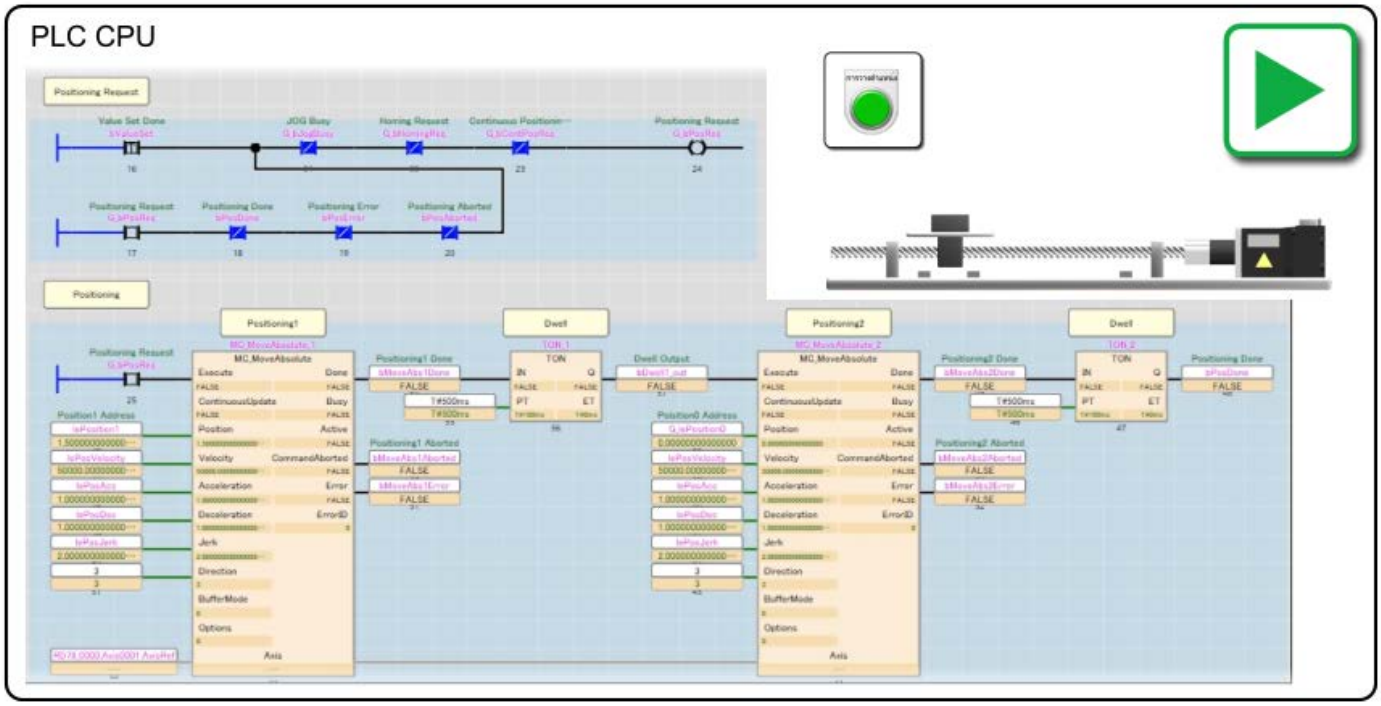
ตรวจสอบโดยการมอนิเตอร์โปรแกรม
เมื่อ X24 เปิด ข้อมูลสำหรับตำแหน่งจะถูกจัดเก็บไว้ในแต่ละเลเบิล และ "bValueSet" จะเปิด
"G_bPosReq" ซึ่งเป็นคำสั่งการดำเนินการของ MC_MoveAbsolute_1 จะเปิดและถูกคงค่าไว้ที่ขอบที่ขอบขาขึ้นของ "bValueSet"

4.6 ตัวอย่างการตรวจสอบการดำเนินการของโปรแกรม



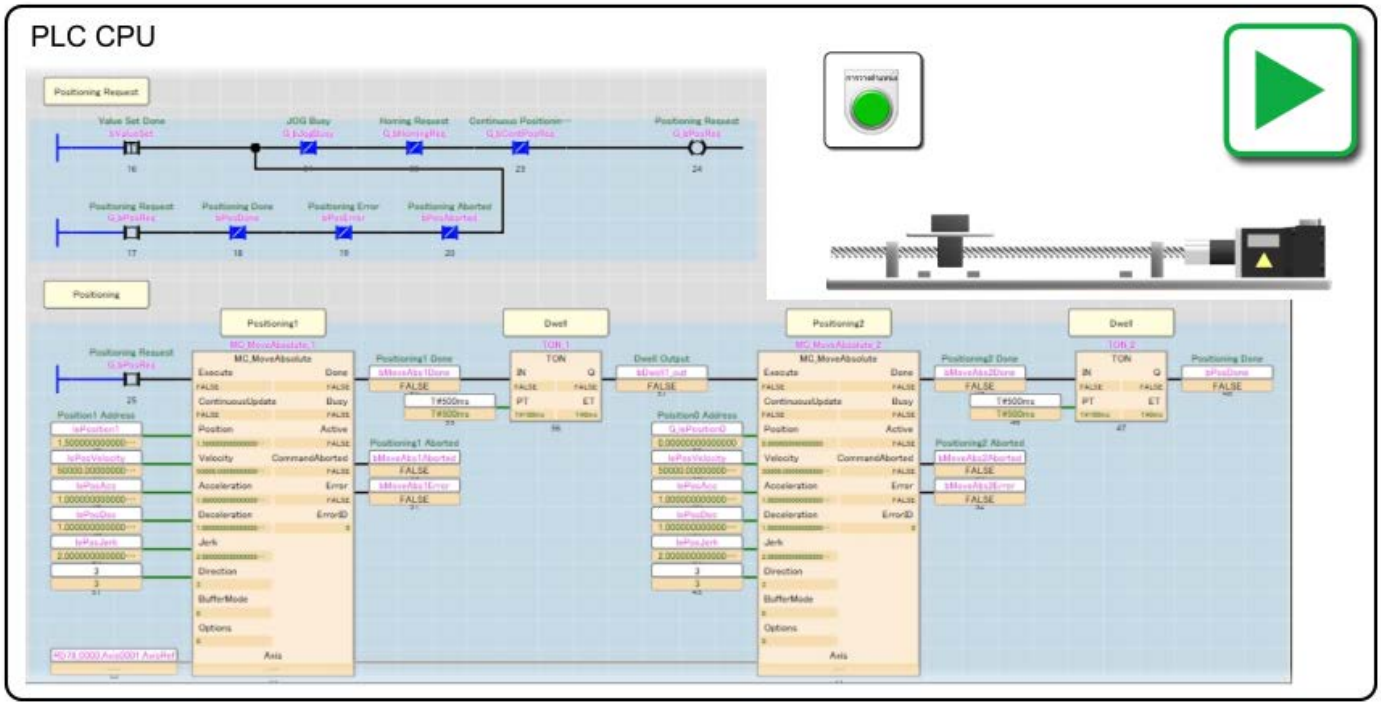
เมื่อ "G_bPosReq" เปิด MC_MoveAbsolute_1
จะเริ่มทำงานและเซอร์โวมอเตอร์จะเริ่มทำงาน

4.6 ตัวอย่างการตรวจสอบการดำเนินการของโปรแกรม

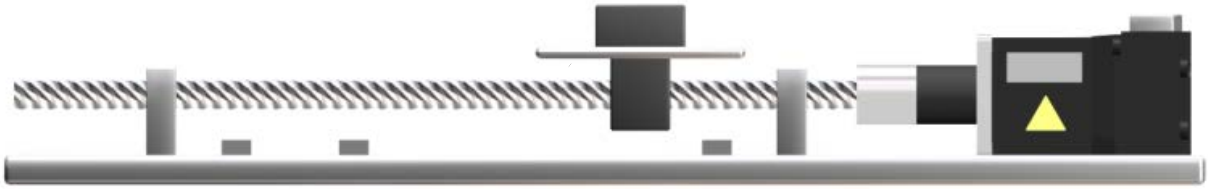


เมื่อการวางตำแหน่งโดย MC_MoveAbsolute_1 เสร็จแล้ว TON_1
ซึ่งเป็นการอยู่กับที่ จะดำเนินการ
เมื่อผ่านไป 500 ms MC_MoveAbsolute_2
จะถูกดำเนินการ และเซอร์โวมอเตอร์จะเริ่มทำงาน

4.6 ตัวอย่างการตรวจสอบการดำเนินการของโปรแกรม

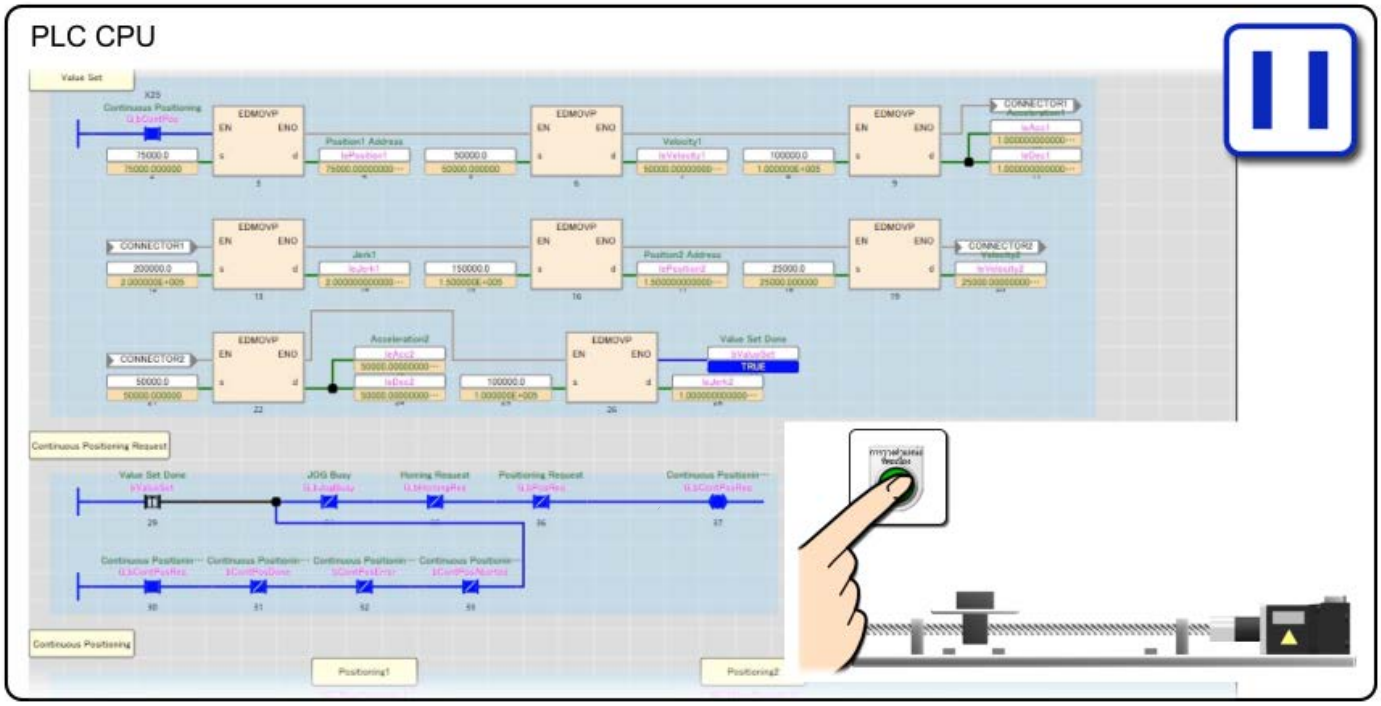


เมื่อการวางตำแหน่งโดย MC_MoveAbsolute_2 เสร็จแล้ว TON_2
ซึ่งเป็นการอยู่กับที่ จะดำเนินการ
เมื่อผ่านไป 500 ms การรักษา "G_bPosReq" จะถูกล้าง
และรีเซ็ตเป็นสถานะเริ่มต้น



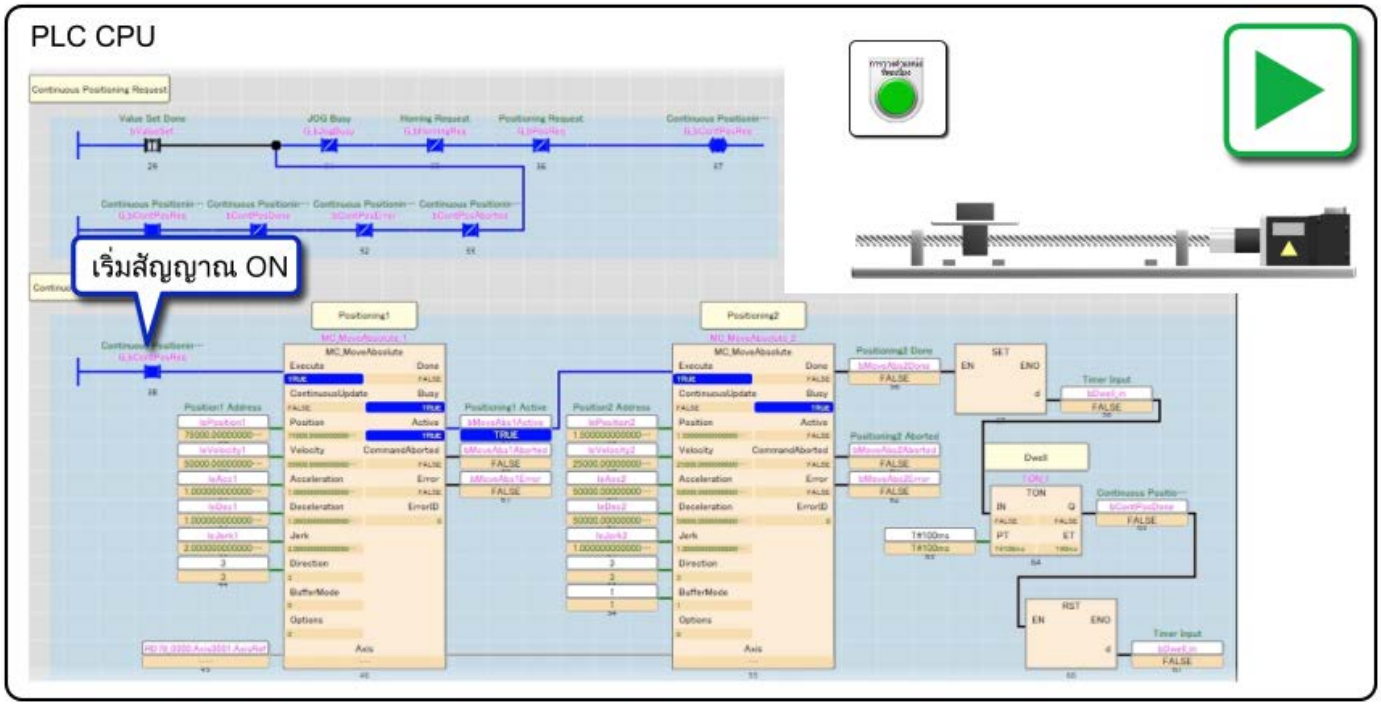
เปิดการเริ่มต้นตำแหน่งต่อเนื่อง (X25)
เพื่อเริ่มการดำเนินการของโหมดบัฟเฟอร์ (mc_Buffered)

4.6 ตัวอย่างการตรวจสอบการดำเนินการของโปรแกรม



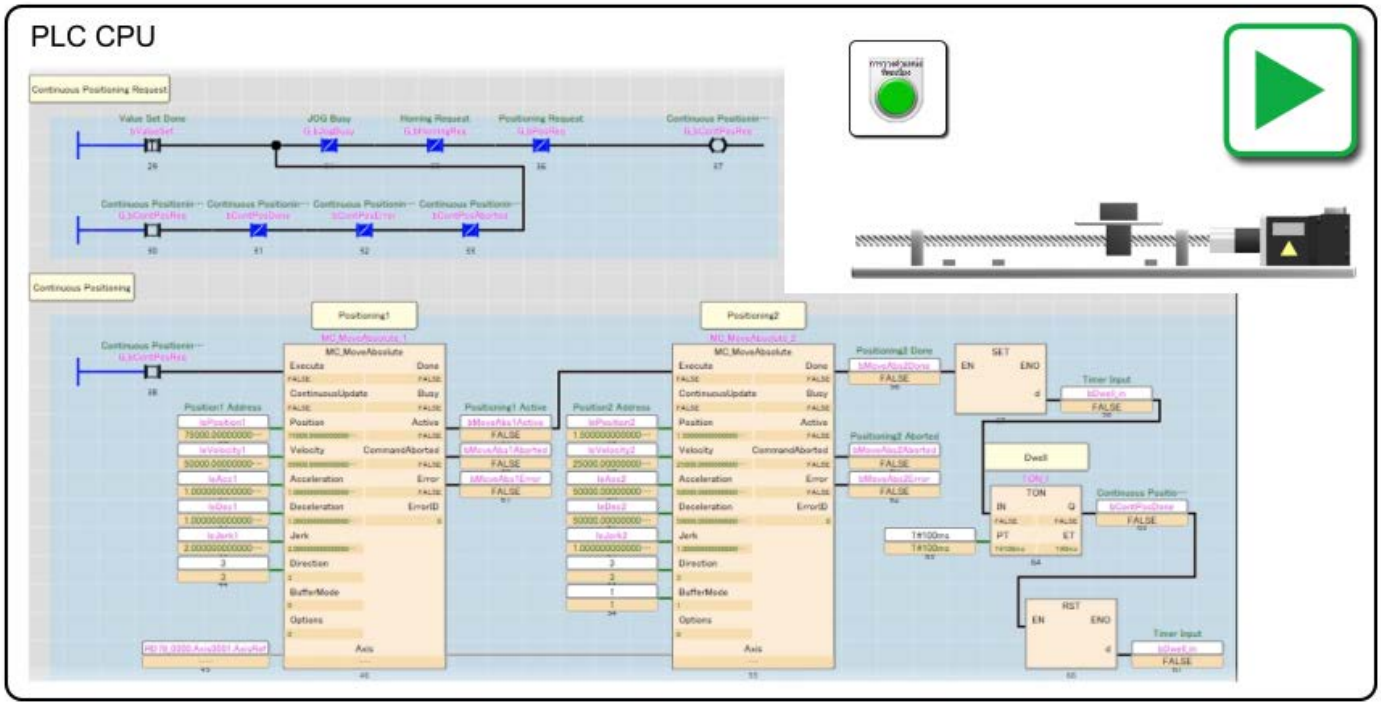
ตรวจสอบโดยการมอนิเตอร์โปรแกรม
เมื่อ X25 เปิด ข้อมูลสำหรับตำแหน่งจะถูกจัดเก็บไว้ในแต่ละเลเบิล และ "bValueSet" จะเปิด
"G_bContPosReq" ซึ่งเป็นคำสั่งการดำเนินการของ MC_MoAbsolute_1 จะเปิดและถูกคงค่าไว้ที่ขอบที่ขอบขาขึ้นของ "bValueSet"

4.6 ตัวอย่างการตรวจสอบการดำเนินการของโปรแกรม



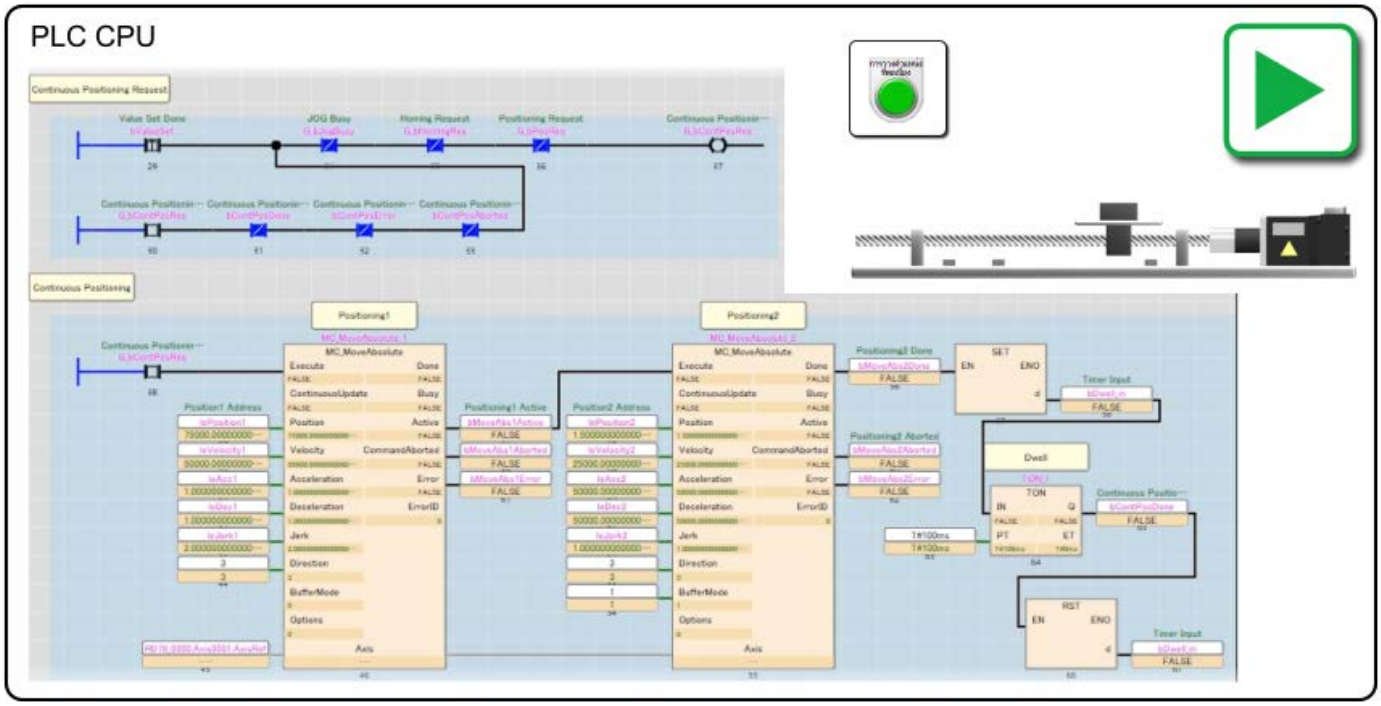
เมื่อ "G_bContPosReq" เปิด MC_MoveAbsolute_1
จะเริ่มทำงานและเซอร์โวมอเตอร์จะเริ่มทำงาน
ในขณะนี้ เนื่องจากเอาต์พุต Active คือคำสั่งการดำเนินการของ
MC_MoveAbosolute_2 ดังนั้น MC_MoveAbsolute_2 จึงถูกบัฟเฟอร์

4.6 ตัวอย่างการตรวจสอบการดำเนินการของโปรแกรม



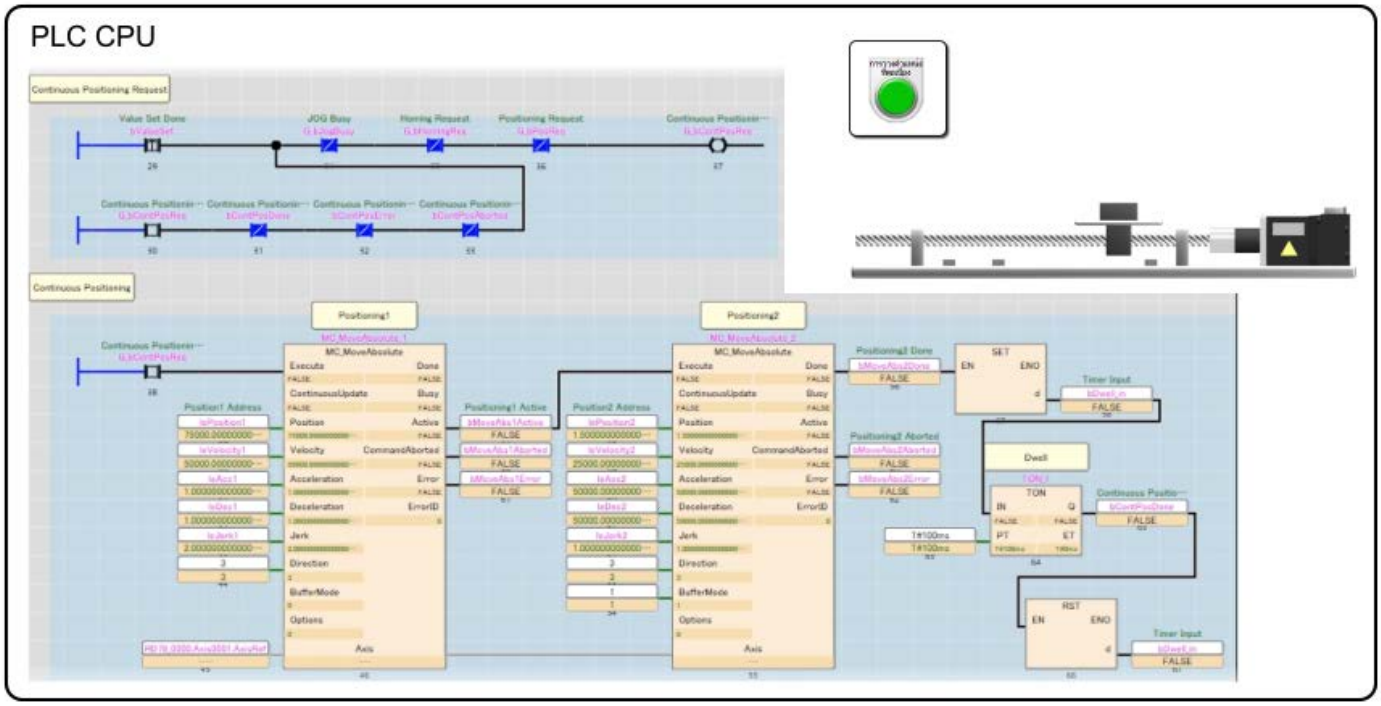
เมื่อการดำเนินการของ MC_MoveAbosolute_1 เสร็จแล้ว
MC_MoveAbsolute_2 ที่ถูกบัฟเฟอร์จะถูกดำเนินการ
เมื่อการดำเนินการของ MC_MoveAbosolute_2 เสร็จแล้ว TON_1
ซึ่งเป็นการอยู่กับที่ จะดำเนินการ

4.6 ตัวอย่างการตรวจสอบการดำเนินการของโปรแกรม



เมื่อผ่านไป 100 ms การคงค่า "G_bContPosReq" จะถูกเคลียร์
และรีเซ็ตเป็นสถานะเริ่มต้น

4.6 ตัวอย่างการตรวจสอบการดำเนินการของโปรแกรม



เป็นอันเสร็จสิ้นการตรวจสอบการดำเนินการ
ไปที่หน้าถัดไป

4.7 บทสรุปของบทนี้

ในบทนี้คุณได้เรียนรู้เกี่ยวกับ:

- การรีจิสเตอร์ ไลบารี่ข้อมูลโมดูลโมชั่น FB
- การสร้างโปรเจ็ค
- วิธีใช้ Motion Control FB
- คำอธิบายโปรแกรมตัวอย่าง
- ตัวอย่างการตรวจสอบการดำเนินการของโปรแกรม

ประเด็นสำคัญ

การรีจิสเตอร์ ไลบารี่ข้อมูลโมดูลโมชั่น FB	• ไลบารี่ข้อมูล FB ต้องถูกลงทะเบียนไปยัง GX Works3 เพื่อใช้ Motion control FB ใน PLC CPU
การสร้างโปรเจ็ค	• กำหนดค่าพารามิเตอร์แกนและการตั้งค่าอื่นๆ เช่นเดียวกับเมื่อตั้งโปรแกรมไปยังโมดูลโมชั่น
วิธีใช้ Motion Control FB	• คุณสามารถวาง Motion control FB ลงในตัวแก้ไขโปรแกรมได้โดยการลากและวางจากแท็บไลบารี่ข้อมูลของหน้าต่างการเลือกองค์ประกอบของ GX Works3 • เชื่อมต่อหน้าสัมผัสและเลเบิลเข้ากับอินพุต/เอาท์พุตของ FB
คำอธิบายโปรแกรมตัวอย่าง	• คุณได้สร้างโปรแกรมที่คล้ายกับโปรแกรมตัวอย่างในบทที่ 2 และบทที่ 3 โดยใช้ PLC CPU เท่านั้น
ตัวอย่างการตรวจสอบการดำเนินการของโปรแกรม	• คุณได้ตรวจสอบการดำเนินการของโปรแกรมตัวอย่างในวิดีโอแล้ว

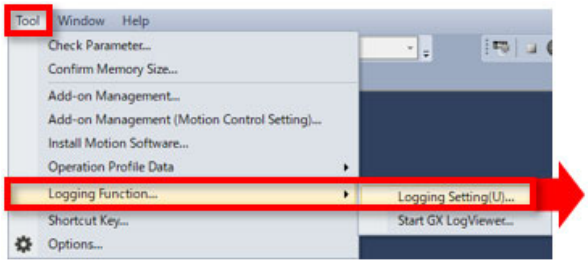
บทที่ 5 การ Logging

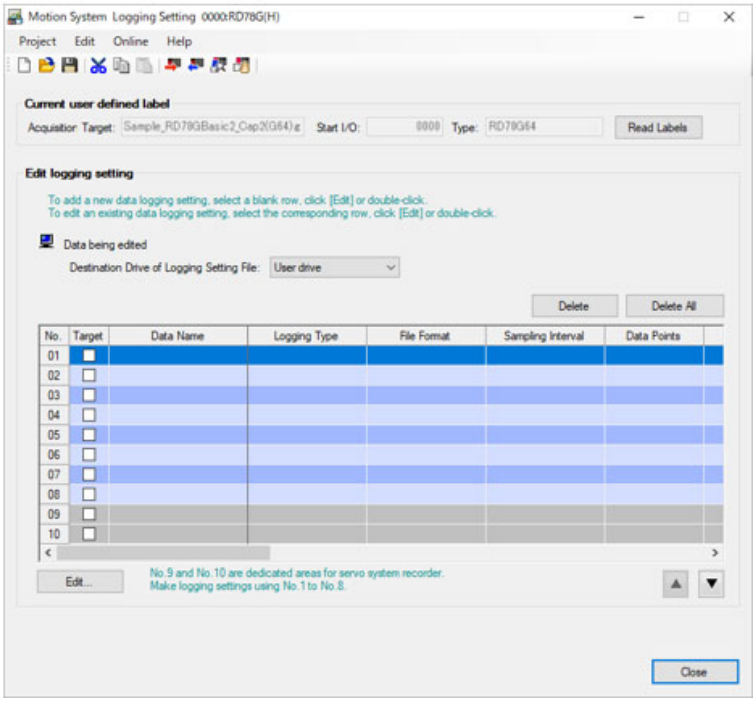
บทนี้จะอธิบายวิธีการบันทึกข้อมูลของโมดูลโมชั่นและแสดงในกราฟ
ในหลักสูตรนี้ โปรแกรมเริ่มต้นการกำหนดตำแหน่งของโปรแกรมตัวอย่างในบทที่ 2 และบทที่ 3 จะถูกบันทึกเป็นตัวอย่าง

(หมายเหตุ) ไม่สามารถบันทึกโปรแกรมในบทที่ 4 ด้วยขั้นตอนที่อธิบายไว้ในบทนี้ได้
ต้องใช้ "CPU Module Logging Configuration Tool"

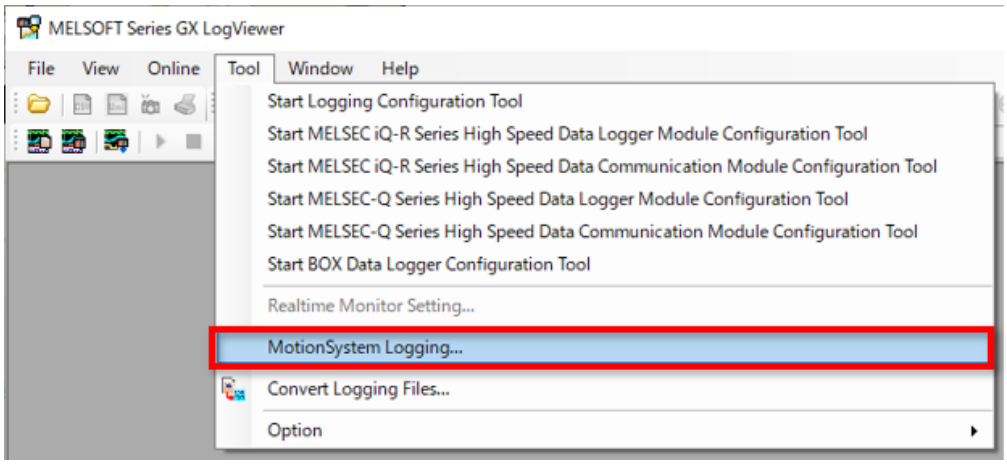
5.1 การเริ่มต้นเครื่องมือกำหนดค่าการบันทึก

เลือก [Tool] → [Logging Function] → [Logging Setting] จากแถบเครื่องมือของหน้าจอฟังก์ชันการตั้งค่าการควบคุมโมชั่น
เครื่องมือการตั้งค่าการบันทึกระบบการเคลื่อนไหวเริ่มทำงาน



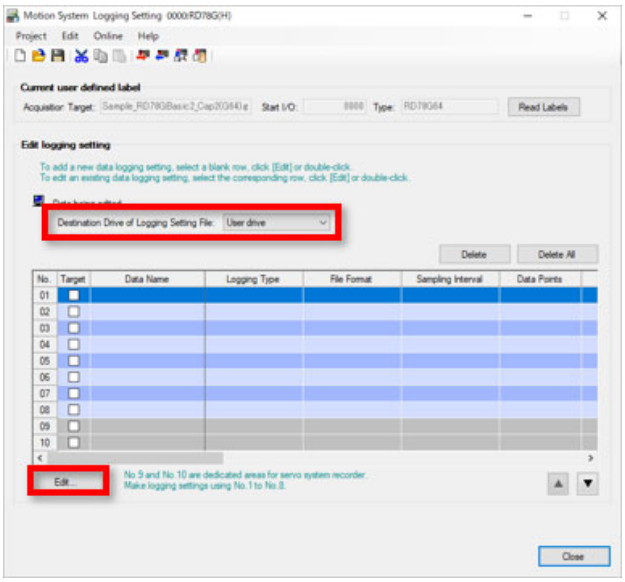


[จุดสำคัญ]
สามารถเริ่มต้นเครื่องมือการตั้งค่าการบันทึกระบบการเคลื่อนไหวได้จาก [Tool] → [MotionSystem Logging] ใน GX LogViewer

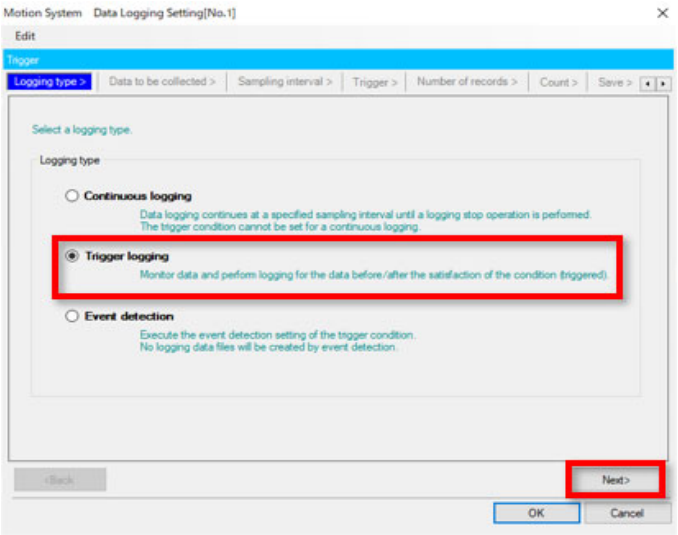


5.2 การตั้งค่าข้อมูลที่จะบันทึก

(1) ในฟิลด์แก้ไขการตั้งค่าการบันทึกของเครื่องมือตั้งค่าการบันทึกระบบการเคลื่อนไหว ให้ตั้งค่าปลายทางการบันทึกของข้อมูลการบันทึก หลังจากนั้น คลิกปุ่ม [Edit]
หน้าจอการตั้งค่าการบันทึกข้อมูลจะปรากฏขึ้น



(2) เลือก [Logging type] จากการ logging อย่างต่อเนื่อง ทริกเกอร์การ logging และการตรวจจับเหตุการณ์
ในหลักสูตรนี้จะอธิบายรายละเอียดของทริกเกอร์การ logging
เลือกทริกเกอร์การ logging แล้วคลิกปุ่ม [Next]

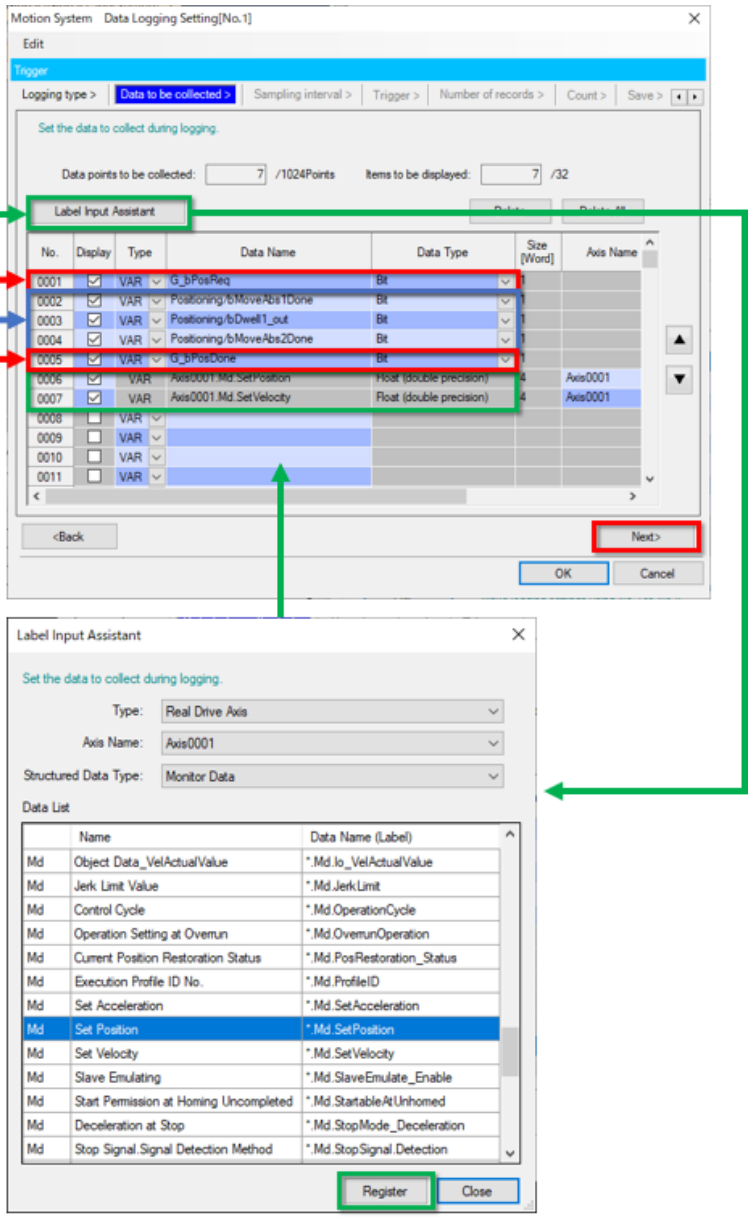


(3) เลเบิลของข้อมูลที่จะบันทึกถูกรีจิสเตอร์ใน [Data to be collected]

1) โกลบอลเลเบิล
ป้อนชื่อโกลบอลเลเบิลในฟิลด์ชื่อข้อมูล
ในฟิลด์ประเภทข้อมูล เลือกประเภทข้อมูลของเลเบิล

2) โลคัลเลเบิล
ป้อนชื่อข้อมูลในรูปแบบ "ชื่อโปรแกรม/ชื่อโลคัลเลเบิล"
ในฟิลด์ประเภทข้อมูล เลือกประเภทข้อมูลของเลเบิล

3) Structured data type
คลิกปุ่ม [Label Input Assistant] และเลือกสมาชิกของประเภท Structured data type จากรายการ เลือกจากรายการแล้วคลิกปุ่ม [Register] เพื่อสะท้อนถึงข้อมูลที่จะรวบรวม



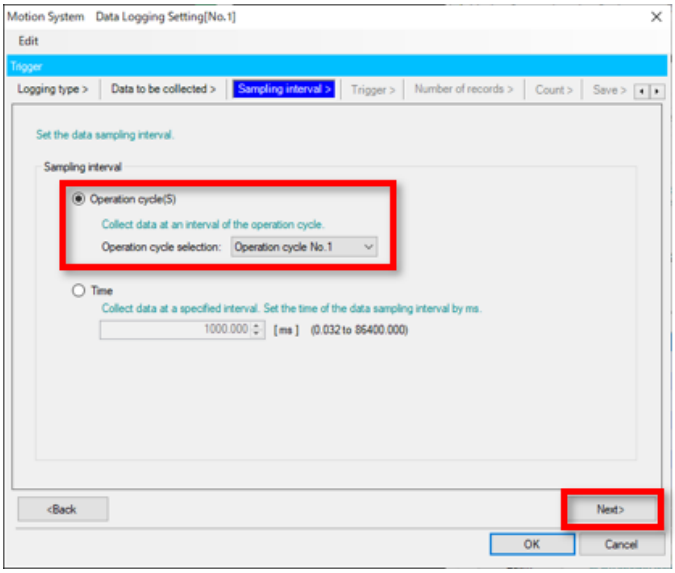
ในหลักสูตรนี้ ข้อมูลต่อไปนี้จะถูกบันทึกไว้เป็นตัวอย่าง

ชื่อข้อมูล
G_bPosReq
Positioning/bMoveAbs1Done
Positioning/bDwell1_out
Positioning/bMoveAbs2Done
G_bPosDone
Axis0001.Md.SetPosition
Axis0001.Md.SetVelocity

คลิกปุ่ม [Next] เมื่อลงทะเบียนเสร็จแล้ว

(4) ตั้งค่าช่วงเวลาการสุ่มตัวอย่างใน [Sampling Interval]
ในหลักสูตรนี้ ใช้รอบการดำเนินการที่ 1 สำหรับการสุ่มตัวอย่าง

หลังจากเลือกช่วงเวลาการสุ่มตัวอย่างแล้ว คลิกปุ่ม [Next]



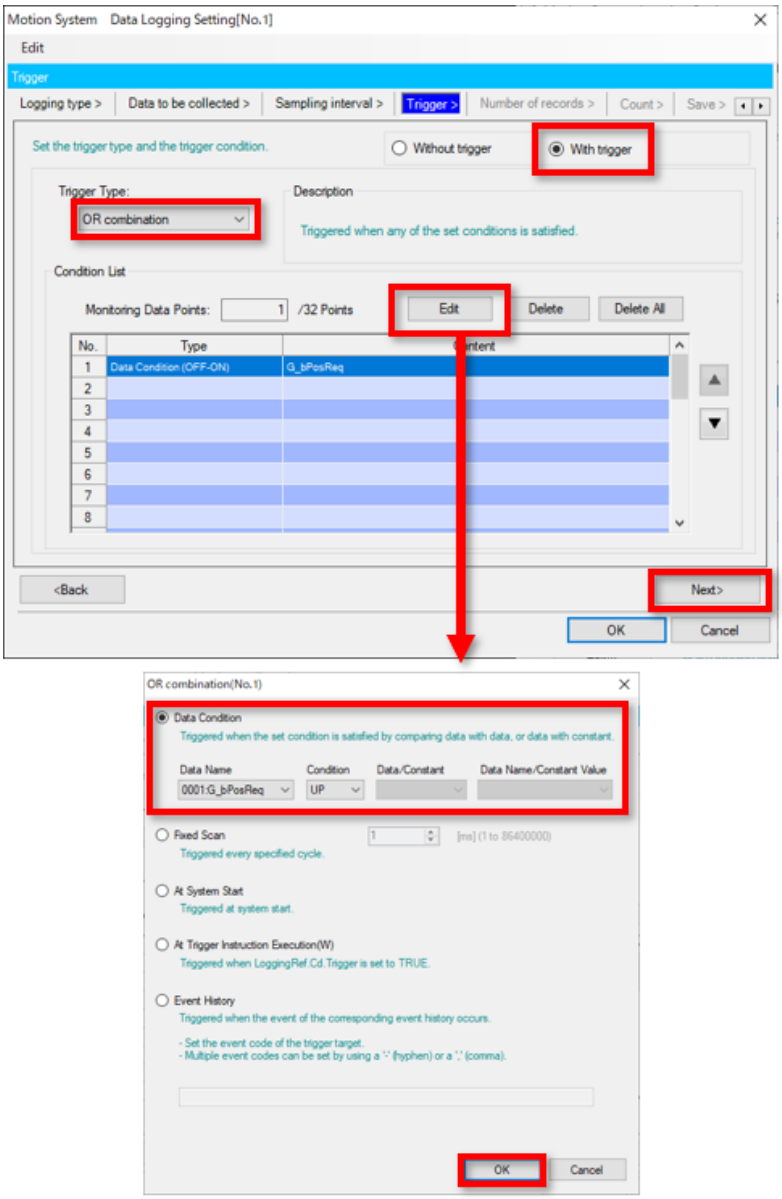
(5) ตั้งค่าเงื่อนไขในการเริ่มต้นการบันทึกใน [Trigger] ในหลักสูตรนี้ การเริ่มต้นบิตที่เป็นสัญญาณเริ่มต้นการกำหนดตำแหน่งจะถูกใช้เป็นทริกเกอร์

1) เลือก [With trigger]

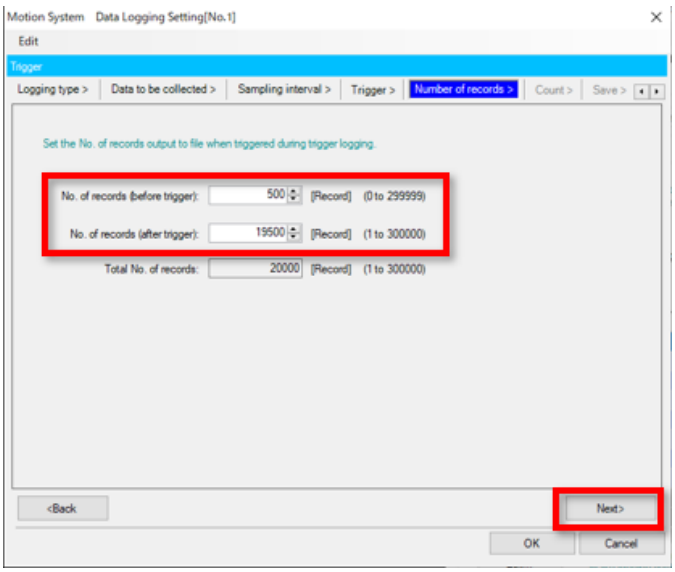
2) เลือก "OR combination" สำหรับประเภทของทริกเกอร์

3) เลือกลำดับที่ 1 ของรายการเงื่อนไข แล้วคลิกปุ่ม แก้ไข หน้าต่างย่อยจะปรากฏขึ้น

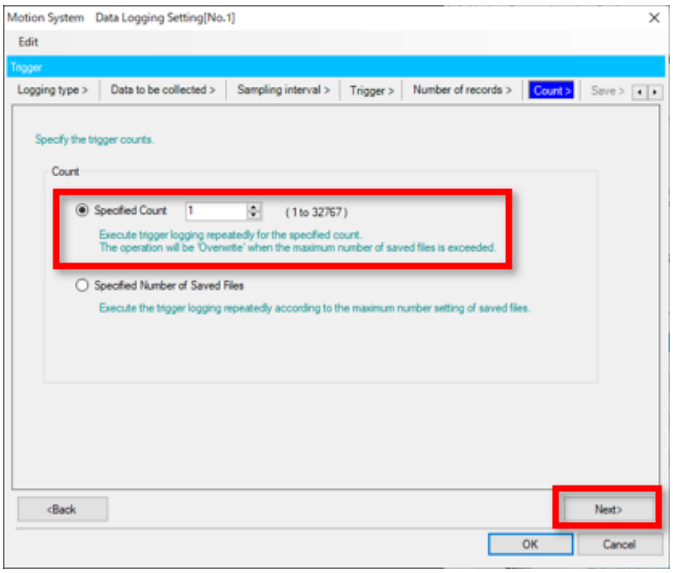
4) เลือก "Data Condition" แล้วเลือก "0001:G_bPosReq" สำหรับชื่อข้อมูล เลือก "UP" สำหรับเงื่อนไข เมื่อเลือกเสร็จแล้ว คลิกปุ่ม [OK]



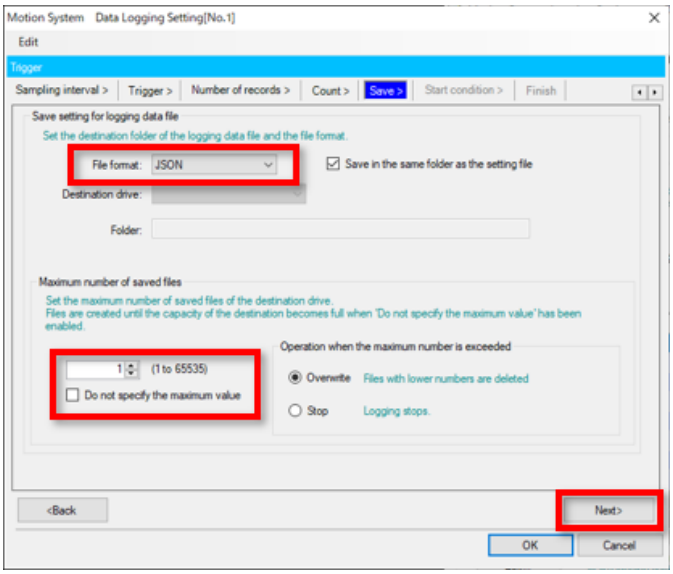
(6) ตั้งค่าจำนวนจุดสุ่มตัวอย่างใน [Number of records]
ในหลักสูตรนี้ จำนวนบันทึก (ก่อนทริกเกอร์) ถูกตั้งค่าเป็น "500" และจำนวนบันทึก (หลักทริกเกอร์) เป็น "19500"
เมื่อตั้งค่าเสร็จแล้ว คลิกปุ่ม [Next]



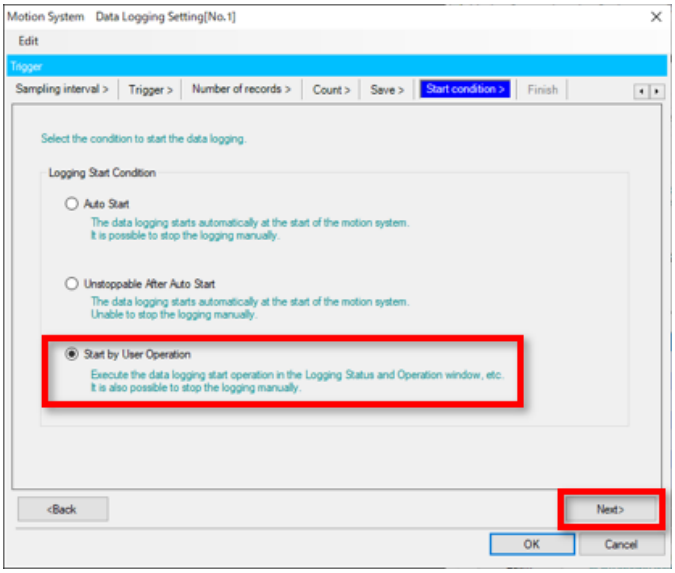
(7) จำนวนการนับบันทึกถูกตั้งค่าไว้ใน [Count] ในหลักสูตรนี้ จำนวนการนับจะถูกตั้งค่าเป็น 1
เมื่อตั้งค่าเสร็จแล้ว คลิกปุ่ม [Next]



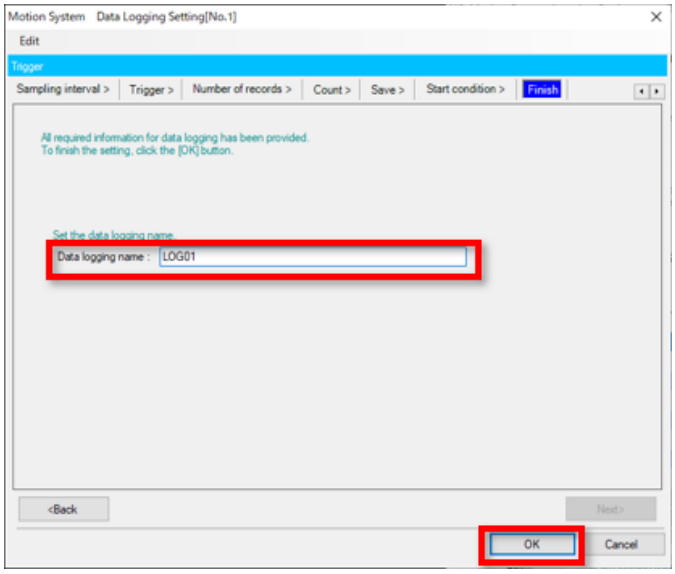
(8) มีการตั้งค่ารูปแบบไฟล์และจำนวนไฟล์ที่บันทึกไว้ของข้อมูลการบันทึกใน [Save] ในหลักสูตรนี้ ค่าเริ่มต้น (รูปแบบ: JSON จำนวนไฟล์ที่บันทึกไว้: 1) ถูกตั้งค่า เมื่อตั้งค่าเสร็จแล้ว คลิกปุ่ม [Next]



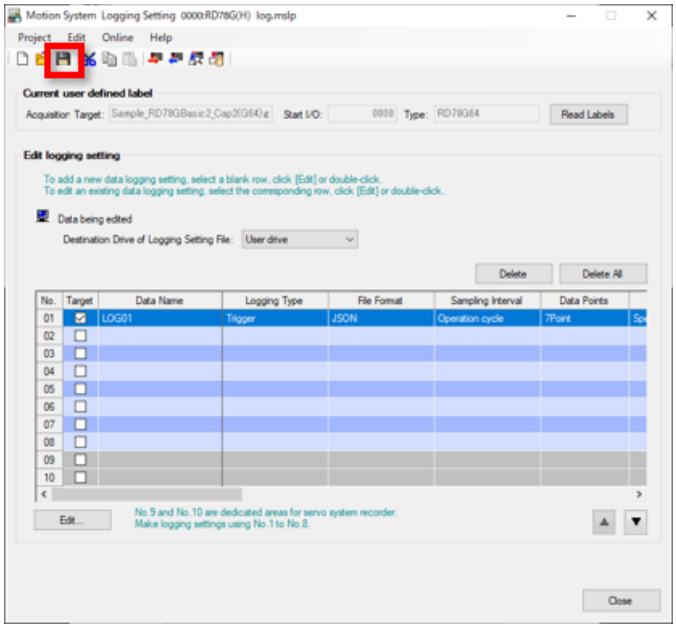
(9) ตั้งค่าเงื่อนไขในการเริ่มต้นการบันทึกใน [Start condition] ในหลักสูตรนี้ "Start by User Operation" ถูกตั้งค่า เมื่อตั้งค่าเสร็จแล้ว คลิกปุ่ม [Next]



(10) ชื่อการบันทึกข้อมูลถูกตั้งค่าไว้ใน [Finish]
ในหลักสูตรนี้ ค่าเริ่มต้น (LOG01) ถูกตั้งค่า
เมื่อตั้งค่าเสร็จแล้ว คลิกปุ่ม [Next]

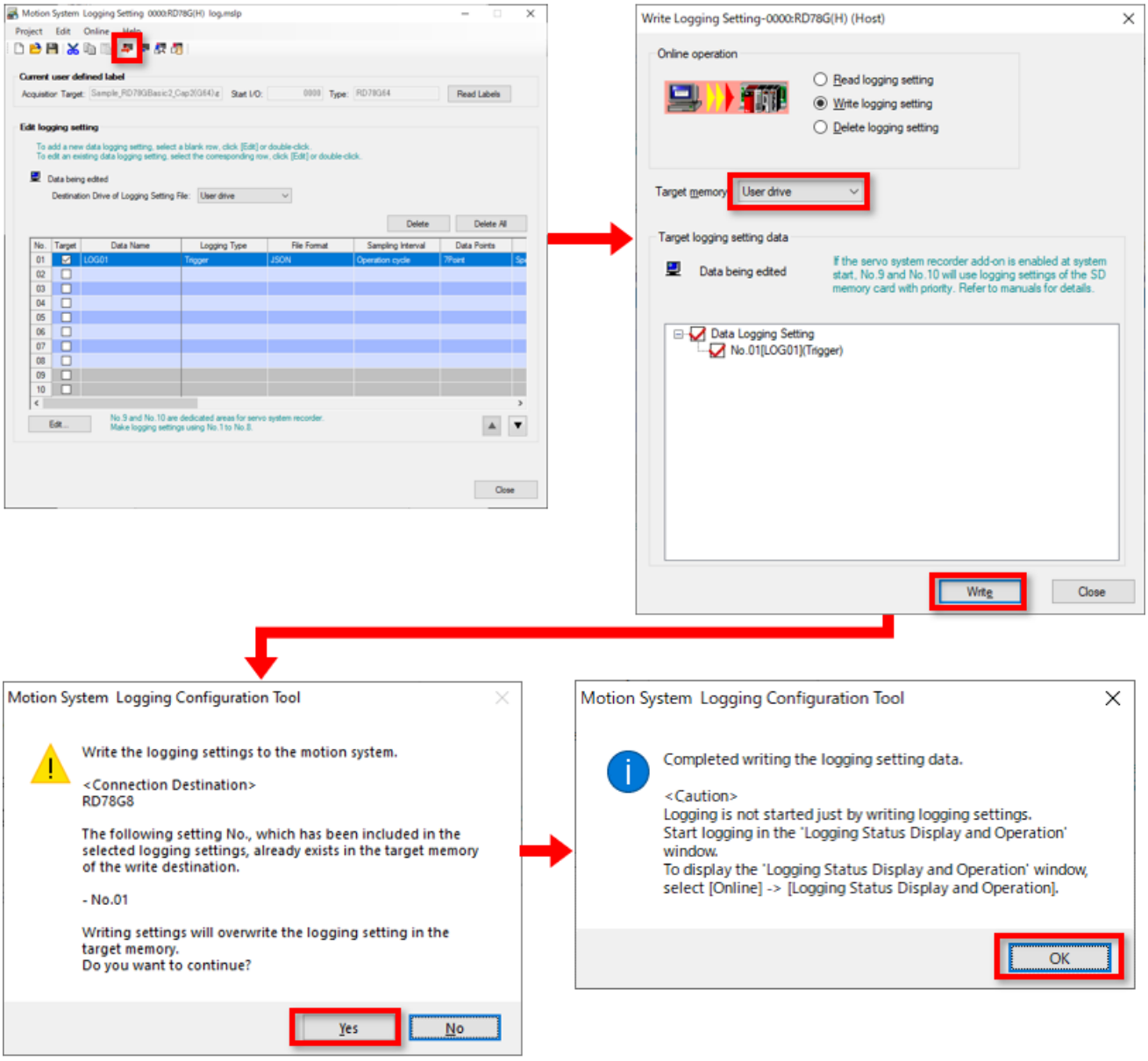


(11) กลับไปที่เครื่องมือการตั้งค่าการบันทึกระบบการเคลื่อนไหว
สามารถบันทึกการตั้งค่าที่กำหนดค่าไว้ได้
คลิกไอคอนบันทึกและบันทึกไปยังปลายทางที่คุณเลือก



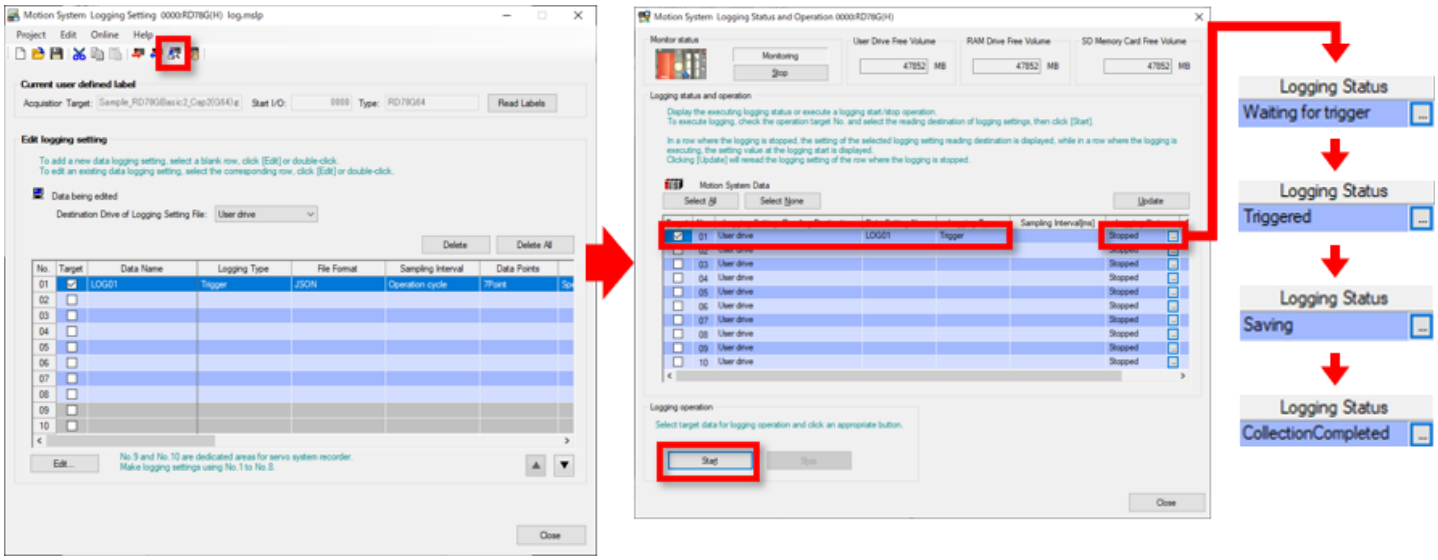
5.3 การเขียนการตั้งค่าการบันทึก

ข้อมูลการตั้งค่าของการบันทึกจะถูกเขียน
คลิกไอคอนการตั้งค่าการบันทึกการเขียน เลือกหน่วยความจำเป้าหมาย แล้วคลิกปุ่ม [Write] button
หน้าต่างยืนยันจะปรากฏขึ้น
คลิกปุ่ม [Yes] แล้วดำเนินการต่อ เมื่อเขียนเสร็จแล้ว คลิกปุ่ม [OK] และปิดหน้าจอ



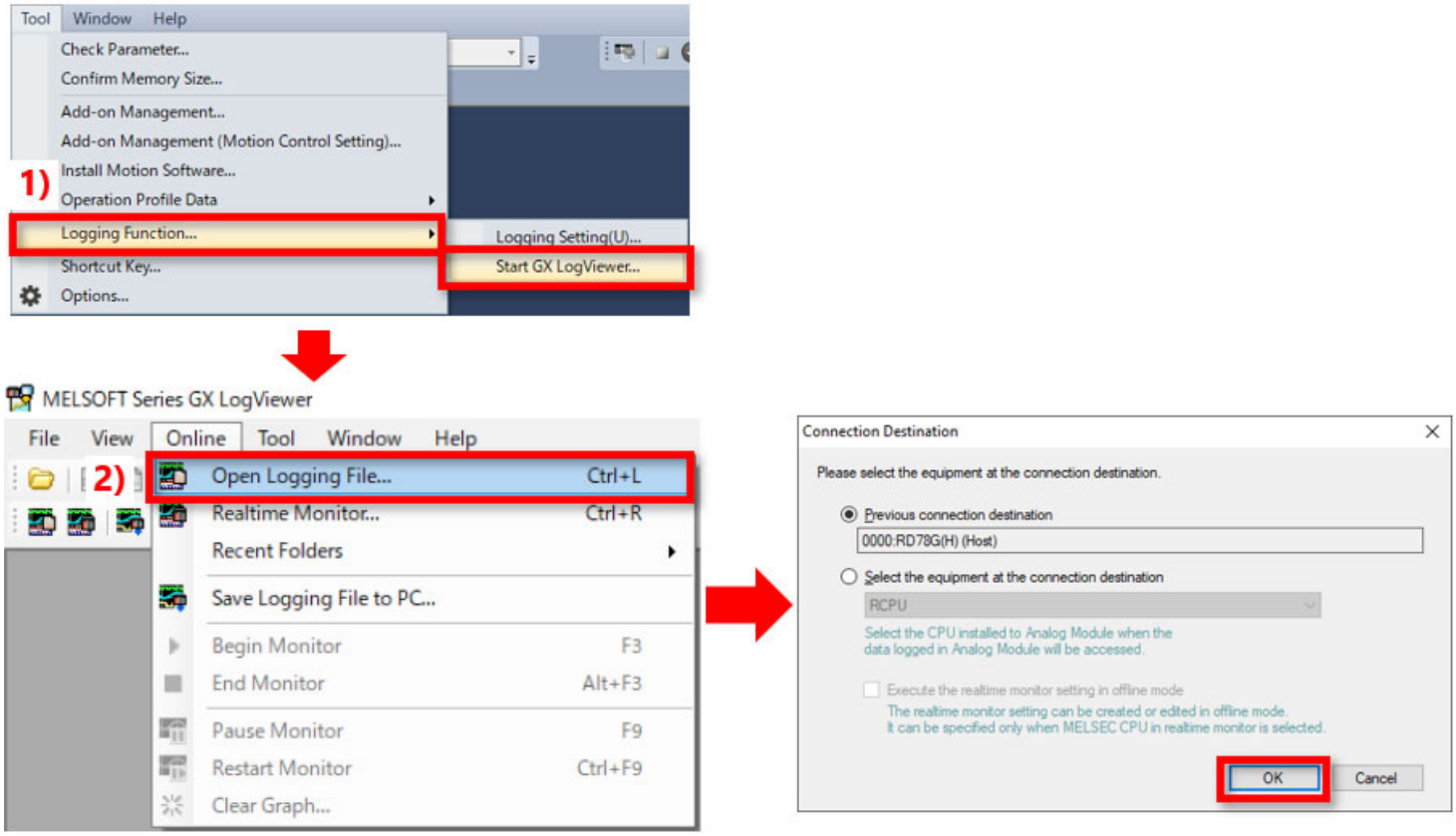
5.4 การเริ่มการบันทึก

เมื่อตั้งค่า "Start by User Operation" ใน 5.2 (9) คลิกไอคอน [Logging Status and Operation] เพื่อแสดงจอ [Logging Status and Operation] และเพื่อเริ่มการบันทึก
เมื่อเลือกชื่อการตั้งค่าข้อมูลการบันทึกที่จะดำเนินการแล้ว คลิกปุ่ม [Start] LoggingStatus switches เป็น “Waiting for trigger"
เมื่อโปรแกรมถูกดำเนินการในสถานะนี้และเป็นไปตามเงื่อนไขทริกเกอร์ (เมื่อ X24 ถูกเปิดเป็น ON ในตัวอย่างนี้ของหลักสูตรนี้) สถานะจะเปลี่ยนเป็น “Triggered"
เมื่อการบันทึกเสร็จแล้ว สถานะจะเปลี่ยนเป็น “CollectionCompleted" จาก “Saving"



5.5 การอ่านข้อมูลที่บันทึก

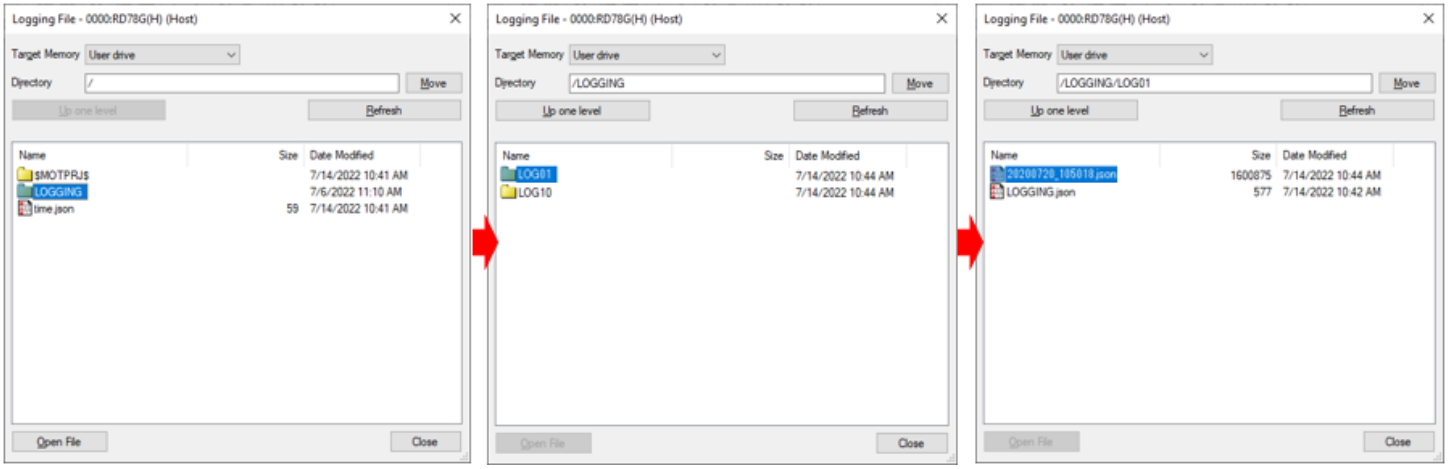
GX LogViewer ถูกใช้เพื่ออ่านข้อมูลที่บันทึก
เลือก [Tool] → [Logging Function] → [Start GX LogViewer] จากแถบเครื่องมือของหน้าจอฟังก์ชันการตั้งค่าการควบคุมโมชั่น
เมื่อเริ่ม GX LogViewer แล้ว เลือก [Online] → [Open Logging File]
เลือก "0000:RD78G(H) (Host)" ในหน้าจอปลายทางการเชื่อมต่อ (หมายเหตุ)



(หมายเหตุ) หากเริ่ม GX LogViewer แล้ว และได้ตั้งค่าการสื่อสารกับโมดูลโมชั่นแล้ว หน้าจอนี้จะไม่ปรากฏ

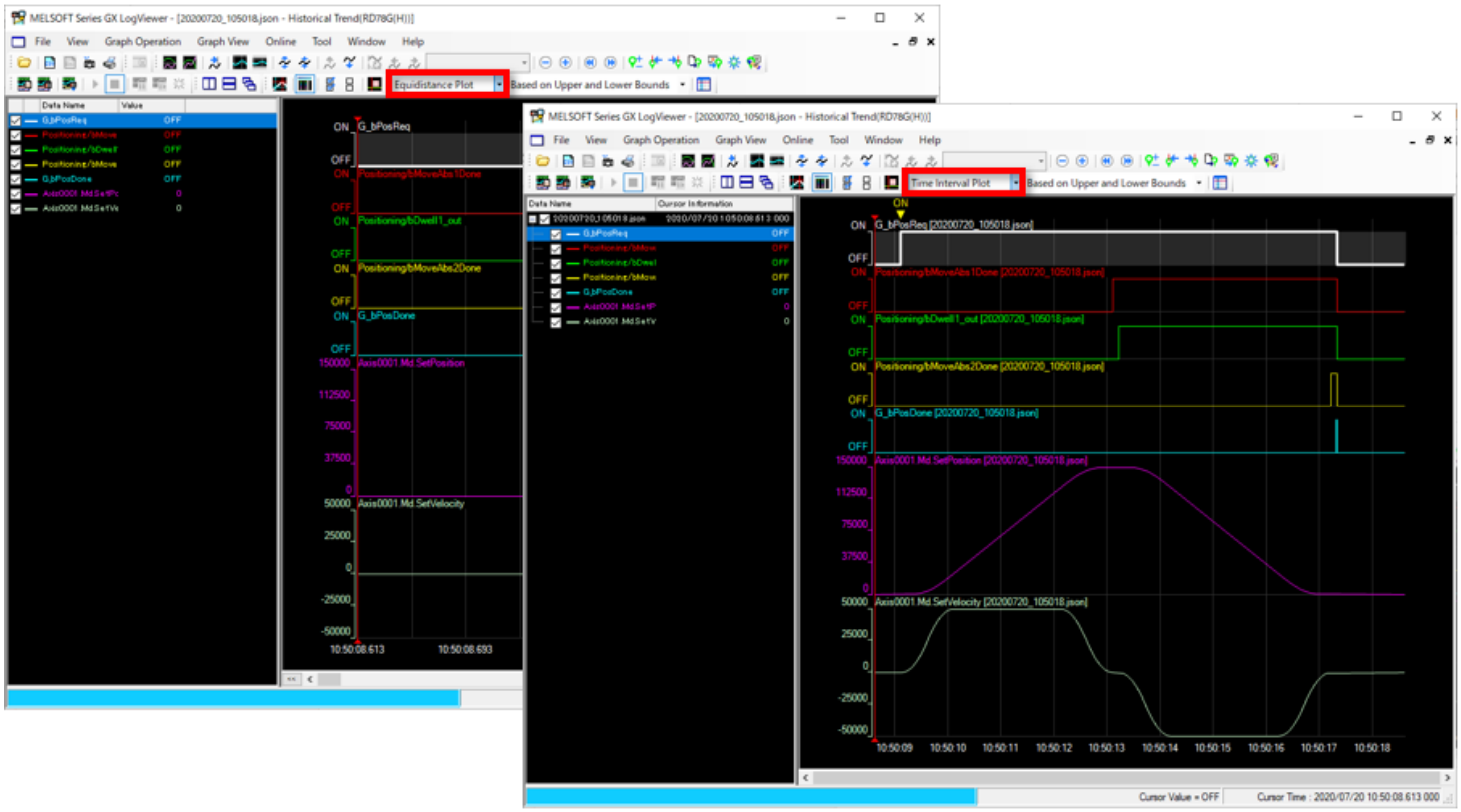
5.5 การอ่านข้อมูลที่บันทึก

เลือกไฟล์บันทึกที่จะอ่าน
ในตัวอย่างของบทนี้ ไดร์ฟของผู้ใช้ใน "LOGGING" → "LOG1" → "(Logged date and time).json" จะถูกเลือก
เลือกชื่อไฟล์และคลิกปุ่ม [Open File]



5.5 การอ่านข้อมูลที่บันทึก

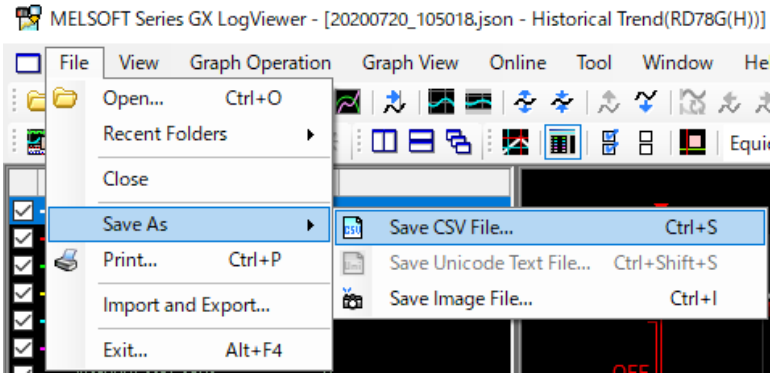
ข้อมูลรูปคลื่นที่ถูกบันทึกใน GX LogViewer จะปรากฏ
เมื่อรูปแบบการทำแผนผังเปลี่ยนจาก"Equidistance Plot" เป็น "Time Interval Plot" แล้ว จะสามารถแสดงรูปคลื่นที่บันทึกไว้ทั้งหมดได้



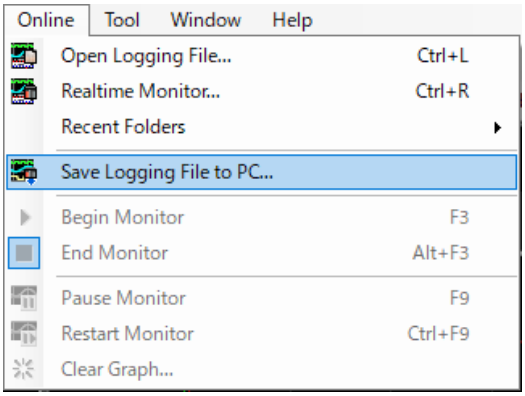
5.6 การบันทึกข้อมูลที่บันทึก

สามารถบันทึกข้อมูลรูปคลื่นที่บันทึกไว้เป็นไฟล์ csv หรือไฟล์ json ได้
(เมื่อบันทึกในรูปแบบ CSV จะสามารถบันทึกเป็นไฟล์ CSV ได้)

1) เมื่อบันทึกเป็นไฟล์ CSV
   เลือก [File] → [Save As] → [Save CSV File] จากแถบเครื่องมือของ GX LogViewer



2) เมื่อบันทึกเป็นไฟล์ json
   เลือก [Online] → [Save Logging File to PC] จากแถบเครื่องมือของ GX LogViewer



5.7 บทสรุปของบทนี้

ในบทนี้คุณได้เรียนรู้เกี่ยวกับ:

- การเริ่มต้นเครื่องมือกำหนดค่าการบันทึก
- การตั้งค่าข้อมูลที่จะบันทึก
- การเขียนการตั้งค่าการบันทึก
- การเริ่มการบันทึก
- การอ่านข้อมูลที่บันทึก
- การบันทึกข้อมูลที่บันทึก

ประเด็นสำคัญ

การเริ่มต้นเครื่องมือกำหนดค่าการบันทึก	• เริ่มเครื่องมือการตั้งค่าการบันทึกระบบการเคลื่อนไหวจากฟังก์ชันการตั้งค่าการควบคุมการเคลื่อนไหว
การตั้งค่าข้อมูลที่จะบันทึก	• ตั้งค่าข้อมูลที่จะบันทึก เงื่อนไขทริกเกอร์ และอื่นๆ โดยปฏิบัติตามขั้นตอนที่แสดงในเครื่องมือการตั้งค่าการบันทึกระบบการเคลื่อนไหว
การเขียนการตั้งค่าการบันทึก	• เขียนข้อมูลการตั้งค่าการบันทึกลงในโมดูลโมชั่นก่อนบันทึก
การเริ่มการบันทึก	• เมื่อเงื่อนไขการเริ่มต้นการบันทึกถูกตั้งค่าเป็น "Start by User Operation" คลิกปุ่มสตาร์ทในหน้าจอ "Logging Status and Operation" เพื่อเริ่มการบันทึก
การอ่านข้อมูลที่บันทึก	• GX LogViewer ถูกใช้เพื่ออ่านข้อมูลที่บันทึก
การบันทึกข้อมูลที่บันทึก	• สามารถบันทึกข้อมูลรูปคลื่นที่บันทึกไว้เป็นไฟล์ csv หรือไฟล์ json ได้

ทดสอบ แบบทดสอบประเมินผล

ขณะนี้คุณจบหลักสูตร**พื้นฐานโมดูลโมชั่น MELSEC iQ-R ซีรีส์ (การควบคุมตำแหน่ง RD78G(H))** และพร้อมที่จะทำการทดสอบสุดท้ายแล้ว หากคุณยังไม่เข้าใจหัวข้อใดๆ ที่ในหลักสูตรนี้ โปรดทบทวนหัวข้อเหล่านั้นก่อนทำการทดสอบ

คำถามในแบบทดสอบประเมินผลนี้มีทั้งหมด 4 ข้อ (7 รายการ)

คุณสามารถทำแบบทดสอบประเมินผลได้หลายครั้งตามต้องการ

ผลคะแนน

จำนวนคำตอบที่ถูกต้อง จำนวนคำถาม เปอร์เซ็นต์คำตอบที่ถูกต้อง และผลลัพธ์ที่แสดงว่าผ่าน/ไม่ผ่านจะปรากฏบนหน้าผลคะแนน

ทดสอบ

แบบทดสอบประเมินผล 1

เลือกคำอธิบายที่ถูกต้องของพับบลิคเลเบิล (คุณสามารถเลือกตอบได้หลายข้อ)

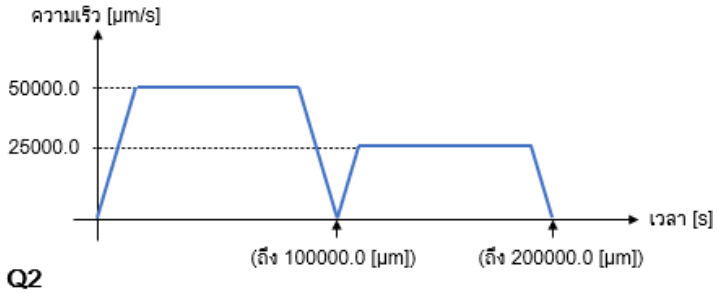
- [ ] ป้ายกำกับสาธารณะคือเลเบิลที่ใช้ร่วมกันซึ่งสามารถใช้ได้ทั้งในโมดูลโมชั่นและ PLC CPU
- [ ] พับบลิคเลเบิลจะถูกรีจิสเตอร์กับโกลบอลเลเบิลที่ PLC CPU
- [ ] เมื่อตั้งค่าโกลบอลเลเบิลเป็นพับบลิคเลเบิล เลือกว่าจะอ่านหรือเขียนเลเบิล from/to PLC CPU

ทดสอบ แบบทดสอบประเมินผล 2

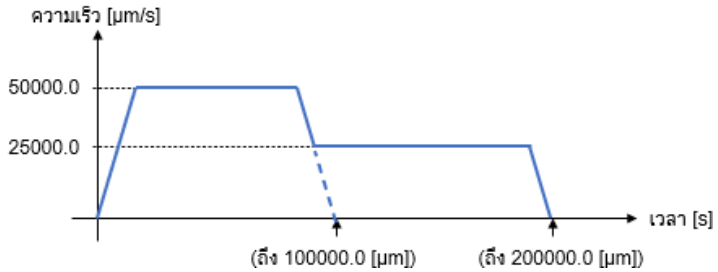
FB1 จะถูกดำเนินการก่อน จากนั้น FB2 จะถูกดำเนินการ
เมื่อตำแหน่งเป้าหมายและความเร็วเป้าหมายของ FB1 และ FB2 เป็นไปตามที่แสดงในตารางต่อไปนี้ เลือกโหมดบัฟเฟอร์ที่จะดำเนินการต่อไป

	ตำแหน่งเป้าหมาย	ความเร็วเป้าหมาย
FB1	100000.0 [μm]	50000.0 [μm/s]
FB2	200000.0 [μm]	25000.0 [μm/s]

Q1



Q2



Q1 เลือกคำตอบที่เหมาะสม

Q2 เลือกคำตอบที่เหมาะสม

Q1:
- 1：mcAborting
- 2：mcBuffered
- 3：mcBlendingNext
- 4：mcBlendingPrevious

Q2:
- 1：mcBlendingNext และ mcBlendingHigh
- 2：mcBlendingPrevious และ mcBlendingHigh
- 3：mcBlendingNext และ mcBlendingLow
- 4：mcBlendingPrevious และ mcBlendingLow

ทดสอบ

แบบทดสอบประเมินผล 3

เลือกประโยคที่ถูกต้องจากรายการต่อไปนี้สำหรับการเขียนโปรแกรมด้วย PLC CPU (คุณสามารถเลือกตอบได้หลายข้อ)

- [ ] ไลบารี่ข้อมูล FB ต้องถูกรีจิสเตอร์ไปยัง GX Works3 เพื่อใช้ Motion control FB สำหรับโมดูลโมชั่นใน PLC CPU
- [ ] วาง motion control FB ในตัวแก้ไขโปรแกรมจากแผนผังโปรเจ็คของ GX Works3
- [ ] ไม่มีพารามิเตอร์ที่จะต้องตั้งค่าสำหรับโมดูลโมชั่น

ทดสอบ แบบทดสอบประเมินผล 4

เลือกคำตอบที่เหมาะสมเพื่อเติมในช่องว่าง

- เริ่มต้น (Q1) เพื่อตั้งค่าข้อมูลที่จะบันทึก
- เขียนข้อมูลที่บันทึกไปที่ (Q2) เพื่อดำเนินการบันทึก
- ใช้ (Q3) เพื่ออ่านข้อมูลที่บันทึกและตรวจสอบรูปคลื่น

Q1 เลือกคำตอบที่เหมาะสม

Q2 เลือกคำตอบที่เหมาะสม

Q3 เลือกคำตอบที่เหมาะสม

Q1:
- 1 : CPU module logging configuration tool
- 2 : เครื่องมือการตั้งค่าการบันทึกระบบการเคลื่อนไหว

Q2:
- 1 : โมดูล CPU
- 2 : โมดูลโมชั่น
- 3 : เซอร์โวแอมพลิฟายเออร์

Q3:
- 1 : MR Configurator2
- 2 : GX LogViewer

ทดสอบ

แบบทดสอบประเมินผล 1

เลือกคำอธิบายที่ถูกต้องของพับบลิคเลเบิล (คุณสามารถเลือกตอบได้หลายข้อ)

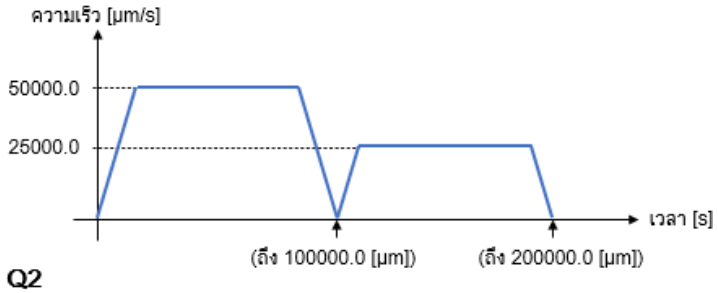
- [x] ป้ายกำกับสาธารณะคือเลเบิลที่ใช้ร่วมกันซึ่งสามารถใช้ได้ทั้งในโมดูลโมชั่นและ PLC CPU
- [ ] พับบลิคเลเบิลจะถูกรีจิสเตอร์กับโกลบอลเลเบิลที่ PLC CPU
- [x] เมื่อตั้งค่าโกลบอลเลเบิลเป็นพับบลิคเลเบิล เลือกว่าจะอ่านหรือเขียนเลเบิล from/to PLC CPU

ทดสอบ แบบทดสอบประเมินผล 2

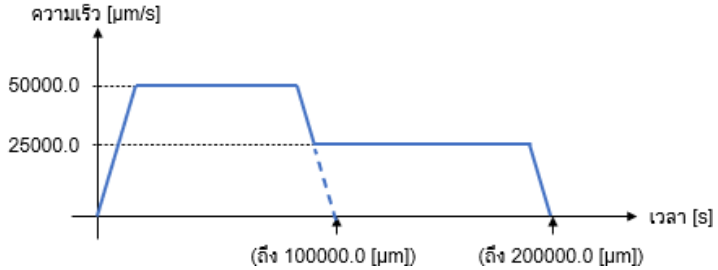
FB1 จะถูกดำเนินการก่อน จากนั้น FB2 จะถูกดำเนินการ
เมื่อตำแหน่งเป้าหมายและความเร็วเป้าหมายของ FB1 และ FB2 เป็นไปตามที่แสดงในตารางต่อไปนี้ เลือกโหมดบัฟเฟอร์ที่จะดำเนินการต่อไป

	ตำแหน่งเป้าหมาย	ความเร็วเป้าหมาย
FB1	100000.0 [μm]	50000.0 [μm/s]
FB2	200000.0 [μm]	25000.0 [μm/s]

Q1



Q2



Q1 2: mcBuffered

Q2 3: mcBlendingNext และ mcBlendingLow

ทดสอบ

แบบทดสอบประเมินผล 3

เลือกประโยคที่ถูกต้องจากรายการต่อไปนี้สำหรับการเขียนโปรแกรมด้วย PLC CPU (คุณสามารถเลือกตอบได้หลายข้อ)

- [x] ไลบารี่ข้อมูล FB ต้องถูกรีจิสเตอร์ไปยัง GX Works3 เพื่อใช้ Motion control FB สำหรับโมดูลโมชั่นใน PLC CPU
- [ ] วาง motion control FB ในตัวแก้ไขโปรแกรมจากแผนผังโปรเจ็คของ GX Works3
- [ ] ไม่มีพารามิเตอร์ที่จะต้องตั้งค่าสำหรับโมดูลโมชั่น

ทดสอบ แบบทดสอบประเมินผล 4

เลือกคำตอบที่เหมาะสมเพื่อเติมในช่องว่าง

- เริ่มต้น (Q1) เพื่อตั้งค่าข้อมูลที่จะบันทึก
- เขียนข้อมูลที่บันทึกไปที่ (Q2) เพื่อดำเนินการบันทึก
- ใช้ (Q3) เพื่ออ่านข้อมูลที่บันทึกและตรวจสอบรูปคลื่น

Q1	2: เครื่องมือการตั้งค่าการบันทึกระบบการเคลื่อนไหว
Q2	2: โมดูลโมชั่น
Q3	2: GX LogViewer

ทดสอบ คะแนนการทดสอบ

คุณทำแบบทดสอบประเมินผลเสร็จสิ้นแล้ว ผลลัพธ์ของคุณมีดังต่อไปนี้
ในการสิ้นสุดแบบทดสอบประเมินผล ให้ไปยังหน้าถัดไป

		1	2	3	4	5	6	7	8	9	10
	แบบทดสอบประเมินผล 1	✔									
	แบบทดสอบประเมินผล 2	✔	✔								
	แบบทดสอบประเมินผล 3	✔									
	แบบทดสอบประเมินผล 4	✔	✔	✔							

จำนวนคำถามทั้งหมด: 7

คำตอบที่ถูกต้อง: 7

เปอร์เซ็นต์: 100 %

ล้าง

คุณจบหลักสูตรพื้นฐานโมดูลโมชั่น MELSEC iQ-R ซีรีส์ (การควบคุมตำแหน่ง RD78G(H)) แล้ว

ขอขอบคุณสำหรับการเรียนรู้หลักสูตรนี้

เราหวังว่าคุณจะเพลิดเพลินกับบทเรียน และข้อมูลที่คุณได้รับจากหลักสูตรนี้จะเป็นประโยชน์ในอนาคต

คุณสามารถทบทวนหลักสูตรได้หลายครั้งตามต้องการ

ทบทวน

ปิด