MITSUBISHI ELECTRIC
Changes for the Better

PLC
การใช้งานการตั้งโปรแกรม

หลักสูตรนี้มีไว้สำหรับผู้เข้าร่วมที่ผ่านหลักสูตรพื้นฐาน MELSEC-Q ซีรีส์ และพร้อมสำหรับการเรียนรู้ขั้นตอนถัดไปของการตั้งโปรแกรม

Copyright ©2014 Mitsubishi Electric Corporation. All Rights Reserved.

บทนำ วัตถุประสงค์ของหลักสูตร

หลักสูตรนี้มีไว้สำหรับผู้ใช้ที่ผ่านหลักสูตรพื้นฐานมาแล้ว หรือผู้ที่มีความรู้เพียงพอสำหรับการศึกษารายละเอียดเพิ่มเติมในด้านฟังก์ชัน และการใช้งานPLC แบบ MELSEC-Q ซีรีส์

ในการศึกษาหลักสูตรนี้ คุณจะได้เรียนรู้เกี่ยวกับการใช้งานอุปกรณ์ต่างๆ ของPLC แบบ Q ซีรีส์ การกำหนดค่าและการวินิจฉัยระบบ CPU รวมถึงการใช้ฟังก์ชันพื้นฐานของPLC แบบ Q ซีรีส์

บทนำ โครงสร้างของหลักสูตร

เนื้อหาของหลักสูตรนี้มีดังนี้
เราขอแนะนำให้คุณเริ่มต้นจากบทที่ 1

บทที่ 1 การตั้งค่าและการปรับเปลี่ยนอุปกรณ์

ศึกษาวิธีการตั้งค่าและเปลี่ยนแปลงการตั้งค่าอุปกรณ์ และฟังก์ชันแลตช์

บทที่ 2 วิธีใช้อุปกรณ์ที่มีฟังก์ชันหลากหลาย

ศึกษาวิธีใช้Retentive Timer, Index Register, Special Relay และSpecial Register

บทที่ 3 หน่วยความจำของโมดูล CPU และไฟล์รีจิสเตอร์

ศึกษาเกี่ยวกับประเภทของหน่วยความจำที่สามารถใช้กับโมดูล CPU และวิธีใช้การลงทะเบียนไฟล์

บทที่ 4 โปรแกรมที่มีจำนวนจริง

ศึกษาเกี่ยวกับการดำเนินการกับจำนวนจริงและการทำงานโดยใช้จำนวนจริง

บทที่ 5 แนวคิดของหมายเลข I/O และวิธีใช้ฟังก์ชันการกำหนด I/O

ศึกษาเกี่ยวกับแนวคิดของหมายเลข I/O และวิธีใช้ฟังก์ชันการกำหนด I/O

แบบทดสอบประเมินผล

เกณฑ์การผ่าน: 60% ขึ้นไป

บทนำ วิธีการใช้งานเครื่องมือการเรียนรู้อิเล็กทรอนิกส์นี้

ไปที่หน้าถัดไป		ไปที่หน้าถัดไป
กลับไปยังหน้าที่แล้ว		กลับไปยังหน้าที่แล้ว
เลื่อนไปยังหน้าที่ต้องการ		ระบบจะแสดง "สารบัญ" ช่วยให้คุณสามารถไปยังหน้าต่างๆ ได้
ออกจากการเรียนรู้		ออกจากการเรียนรู้ ออกจากการเรียนรูระบบจะปิดหน้าต่าง เช่น หน้าต่าง "เนื้อหา" และการเรียนรู้

บทนำ ข้อควรระวังในการใช้งาน

ข้อควรระวังด้านความปลอดภัย

เมื่อคุณเรียนรู้โดยการใช้งานผลิตภัณฑ์จริง โปรดอ่านข้อควรระวังต่างๆ ในคู่มือการใช้งานอย่างละเอียดให้เข้าใจ

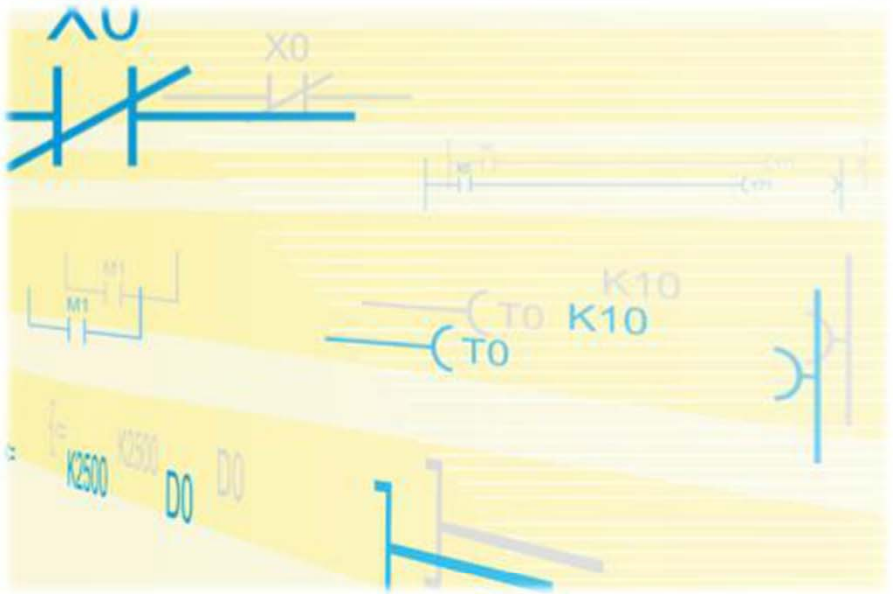
ข้อควรระวังในหลักสูตรนี้

- หน้าจอที่แสดงของเวอร์ชั่นที่คุณใช้อาจจะแตกต่างจากในหลักสูตรนี้
  หลักสูตรนี้จะใช้ซอฟต์แวร์เวอร์ชันต่อไปนี้:
  - GX Works2 เวอร์ชัน 1.91V

บทที่ 1 การตั้งค่าและการปรับเปลี่ยนอุปกรณ์

บทนี้จะอธิบายถึงวิธีเปลี่ยนแปลงการตั้งค่าอุปกรณ์

ส่วนที่ 1.1: การระบุอุปกรณ์
ส่วนที่ 1.2: วิธีปรับแต่งจำนวนจุดอุปกรณ์
ส่วนที่ 1.3: การบันทึกสถานะอุปกรณ์ขณะปิดหรือรีเซ็ตเครื่อง
ส่วนที่ 1.4: สรุป



1.1 การระบุอุปกรณ์

1.1.1 ข้อมูลจำเพาะบิทของเวิร์ดดีไวส์

โดยปกติเวิร์ดดีไวส์จะถูกกำหนดโดยใช้ข้อมูลเวิร์ด แต่จะสามารถกำหนดโดยใช้ข้อมูลบิท (เช่น จำนวนเต็ม ฯลฯ) ได้ด้วย สามารถใช้งานข้อมูลบิทในเวิร์ดดีไวส์ต่างๆ เช่น ดาต้ารีจิสเตอร์ (D) และไฟล์รีจิสเตอร์ (R) ได

ตัวอย่าง: ลงทะเบียนข้อมูล (D)

0	0	1	0	0	1	1	0	1	1	0	0	0	1	0	1

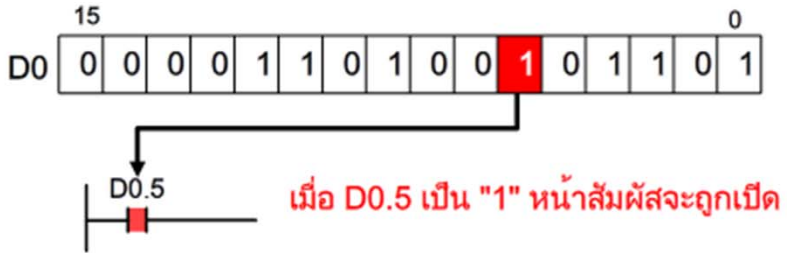
รูปแบบข้อมูลจำเพาะข้อมูลบิท

D□.□

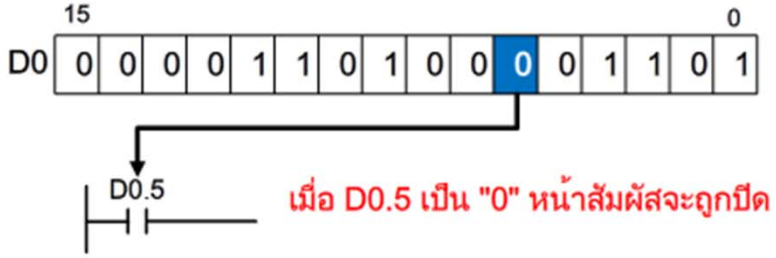
- สัญลักษณ์เวิร์ดดีไวส์ (D, W หรือ R)
- หมายเลขอุปกรณ์
- บิท (0 ถึง F)

ตัวอย่างโปรแกรม

ตัวอย่าง 1 เมื่อบิท 5 ของดาต้ารีจิสเตอร์ "D0" คือ 1



เมื่อบิท 5 ของดาต้ารีจิสเตอร์ "D0" คือ 0



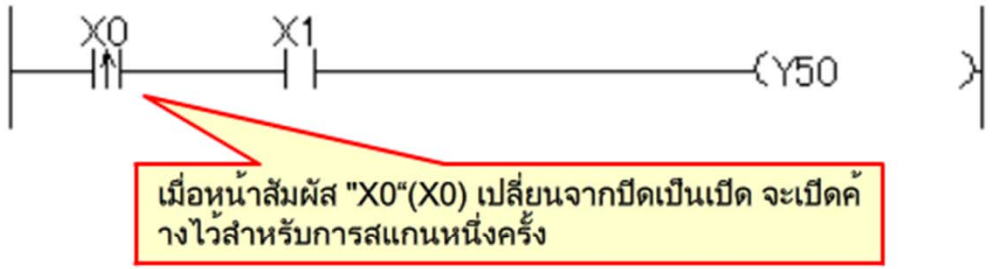
ตัวอย่าง 2

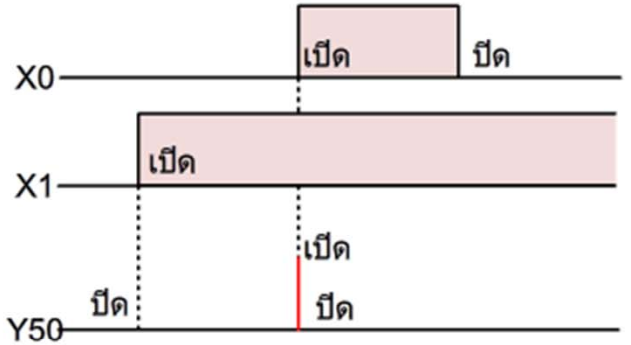
เมื่อ D10.2 คือ "0" ค่าจะถูกเปลี่ยนกลับเป็น "1 (เปิด)"

1.1.2 ข้อมูลจำเพาะขอบขึ้น (Leading Edge) หรือขอบลง (Trailing Edge) สำหรับหน้าสัมผัส

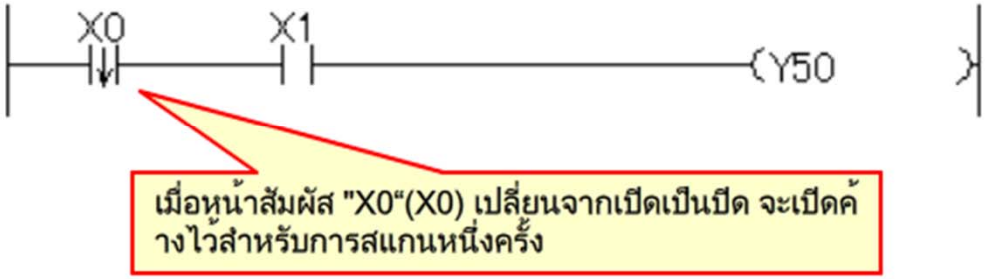
สำหรับการทำงานเปิด/ปิดของหน้าสัมผัส จะสามารถตั้งค่าสัญญาณเป็นเปิดได้เฉพาะการสแกนหนึ่งครั้งที่ขอบขึ้นหรือขอบลงของหน้าสัมผัส ซึ่งจะเป็นประโยชน์สำหรับการตั้งโปรแกรมเงื่อนไขอินพุทสัญญาณขอบขึ้นหรือขอบลง

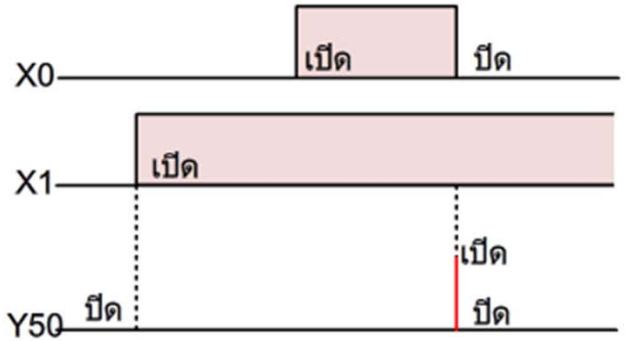
ตัวอย่างโปรแกรมสำหรับหน้าสัมผัสขอบขึ้น





ตัวอย่างโปรแกรมสำหรับหน้าสัมผัสขอบลง





1.2 วิธีปรับแต่งจำนวนจุดอุปกรณ์

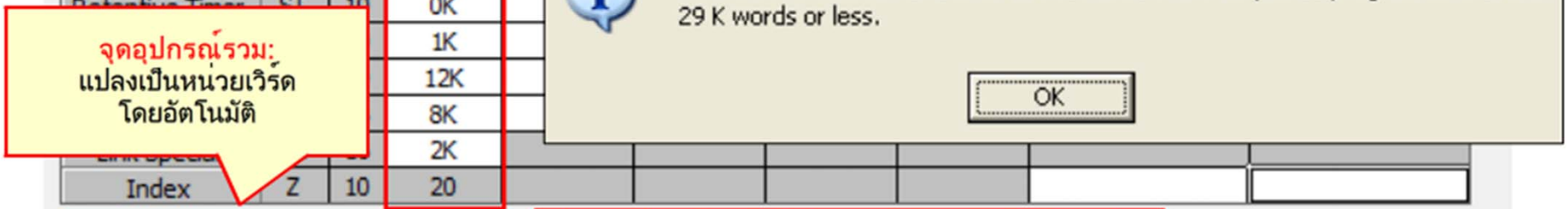
CPU โมดูลที่แตกต่างกันจะมีจำนวนจุดอุปกรณ์ที่แตกต่างกัน โดยที่จำนวนอุปกรณ์จะได้รับการจัดสรรไว้ในเบื้องต้น ตามความจุของโมดูล CPU ที่ใช้
เมื่อจำนวนของจุดที่จัดสรรไปยังอุปกรณ์ที่ใช้งานบ่อยไม่เพียงพอ ให้ลดจุดที่จัดสรรไปยังอุปกรณ์อื่นลง และนำจุดเหล่านั้นไปใช้กับอุปกรณ์ที่ใช้งานบ่อยๆ ไปที่ แท็บ Device (อุปกรณ์) ของหน้าต่าง PLC Parameter (PLC พารามิเตอร์) เพื่อเปลี่ยนแปลงการตั้งค่า

ตัวอย่างของหน้าจอการตั้งค่าอุปกรณ์

	Sym.	Dig.	Device Points	Local Device Start	Local Device End
Input Relay	X	16	8K		
Output Relay	Y	16	8K		
Internal Relay	M	10	8K		
Latch Relay	L	10	8K		
Link Relay	B	16	8K		
Annunciator	F	10	2K		
Link Special	SB	16	2K		
Edge Relay	V	10	2K		
Step Relay	S	10	8K		
Timer	T	10	2K		
[illegible]	ST	10	0K		
[illegible]			1K		
[illegible]			12K		
[illegible]			8K		
[illegible]			2K		
Index	Z	10	20		

Device Points (จุดอุปกรณ์):
- ตามค่าเริ่มต้น จะมีการตั้งค่าเบื้องต้นมาแล้ว
- ค่าในเซลล์สีขาวนั้นสามารถเปลี่ยนแปลงได้
- ตั้งค่าจุดอุปกรณ์เป็นหน่วย 16 จุด
- 1K จุด หมายถึงจุดข้อมูลจริง 1024 จุด

หากความจุของจุดอุปกรณ์ที่ตั้งค่าเกินความจุของโมดูล CPU จะมีข้อความแจ้งให้ปรับเปลี่ยนการตั้งค่าปรากฏขึ้น



จุดอุปกรณ์รวม: แปลงเป็นหน่วยเวิร์ดโดยอัตโนมัติ

Device Total	28.8	K Words
Word Device	25.0	K Words
Bit Device	44.0	K Bits

The total number of device points is up to 29 K words.
Latch(1) : Able to clear the value by using a [illegible] clear.
[illegible] be executed by r[illegible]
When using the local devices, please do the file setting at PLC file setting parameter.

จำนวนจุดอุปกรณ์สูงสุด = ความจุของโมดูล CPU
ตัวอย่างเช่น ความจุของโมดูล CPU Q06UDEHCPU คือ 29K เวิร์ด

1.3 การบันทึกสถานะอุปกรณ์ขณะปิดหรือรีเซ็ตเครื่อง

ฟังก์ชันแลตช์

เมื่อใช้ฟังก์ชันแลตช์ โมดูล CPU จะเก็บค่าอุปกรณ์ไว้ ในขณะที่หยุดการทำงาน
ตัวอย่างเช่น หากเกิดเหตุพลังงานไฟฟ้าขัดของชั่วคราวนานเกินขีดจำกัดที่กำหนดไว้โมดูล CPU จะเก็บข้อมูลไว้ในขณะที่หยุดทำงาน และนำมาใช้ขณะเริ่มทำงานการควบคุมเชิงลำดับใหม่

หากไม่ใช้งานฟังก์ชันแลตช์ ค่าอุปกรณ์จะถูกรีเซ็ตกลับเป็นค่าเริ่มต้น (อุปกรณ์บิทจะถูกปิด และเวิร์ดดีไวส์จะเปลี่ยนเป็น "0") ในเหตุการณ์ต่อไปนี้:

(1) ปิดเครื่อง
(2) รีเซ็ตโดยใช้สวิตช์ "RUN/STOP/RESET" (ทำงาน/หยุด/รีเซ็ต)
(3) เกิดเหตุพลังงานไฟฟ้าขัดของชั่วคราวนานเกินขีดจำกัดที่กำหนดไว้ในโมดูลแหล่งจ่ายไฟ

การตั้งค่าช่วงแลตช์

เลือกแท็บ Device (อุปกรณ์) ในหน้าต่าง PLC Parameter (PLC พารามิเตอร์) ของ GX Works2 เพื่อตั้งค่าช่วงแลตช์ ด้านล่างคือตัวอย่างการตั้งค่าไปยังแลตช์รีเลย์ L0 ถึง L1024 และดาต้ารีจิสเตอร์ D0 ถึง D128

				A	B	C	D	
	Sym.	Dig.	Device Points	Latch (1) Start	Latch (1) End	Latch (2) Start	Latch (2) End	Loc
Input Relay	X	16	8K					
Output Relay	Y	16	8K					
Internal Relay	M	10	8K					
Latch Relay	L	10	8K			0	1024	
Link Relay	B	16	8K					
Annunciator	F	10	2K					
Link Special	SB	16	2K					
Edge Relay	V	10	2K					
Step Relay	S	10	8K					
Timer	T	10	2K					
Retentive Timer	ST	10	0K					
Counter	C	10	1K					
Data Register	D	10	12K			0	128	
Link Register	W	16	8K					
Link Special	SW	16	2K					
Index	Z	10	20					

A	Latch (1) Start (แลตช์ (1) เริ่ม)	ระบุหมายเลขเริ่มต้นของช่วงแลตช์ที่จะตั้งค่า
C	Latch (2) Start (แลตช์ (2) เริ่ม)	
B	Latch (1) End (แลตช์ (1) จบ)	ระบุหมายเลขจบของช่วงแลตช์ที่จะตั้งค่า
D	Latch (2) End (แลตช์ (2) จบ)	

* ดูความแตกต่างระหว่างแลตช์ (1) และ (2) ได้ที่หน้าถัดไป

1.3 การบันทึกสถานะอุปกรณ์ขณะปิดหรือรีเซ็ตเครื่อง

วิธีล้างข้อมูลแลตช์

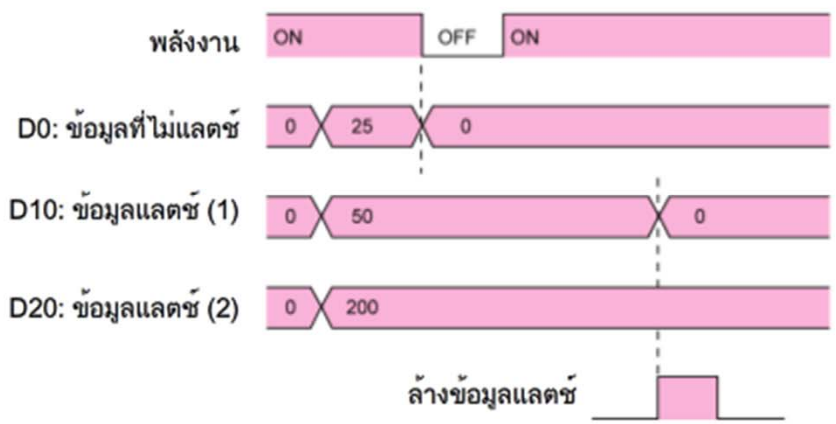
วิธีล้างค่าสำหรับแลตช์ (1) และ (2) จะแตกต่างกัน

แลตช์ (1): ล้างข้อมูลที่แลตช์จากหน้าต่าง Remote Operation (รีโมตโอเปอร์เรชั่น) ของ GX Works2 ใช้แลตช์ 1 เมื่อจำเป็นต้องล้างข้อมูลที่แลตช์ ณ สถานที่ติดตั้ง

แลตช์ (2): ล้างข้อมูลที่แลตช์โดยใช้คำแนะนำเฉพาะของโปรแกรม ใช้แลตช์ 2 เมื่อไม่จำเป็นต้องล้างข้อมูลที่แลตช์ ณ สถานที่ติดตั้ง

	Sym.	Dig.	Device Points	Latch (1) Start	Latch (1) End	Latch (2) Start	Latch (2) End	Loca
Input Relay	X	16	8K					
Output Relay	Y	16	8K					
Internal Relay	M	10	8K					
Latch Relay	L	10	8K			0	1024	
Link Relay	B	16	8K					
Annunciator	F	10	2K					
Link Special	SB	16	2K					
Edge Relay	V	10	2K					
Step Relay	S	10	8K					
Timer	T	10	2K					
Retentive Timer	ST	10	0K					
Counter	C	10	1K					
Data Register	D	10	12K	0	128			
Link Register	W	16	8K					
Link Special	SW	16	2K					
Index	Z	10	20					

แผนผังกำหนดเวลา



วิธีล้างข้อมูลที่แลตช์ผ่านรีโมตโอเปอร์เรชั่น

เลือก Online (ออนไลน์) บนแถบเมนูของ GX Works2 จากนั้นเลือก Remote Operation (รีโมตโอเปอร์เรชั่น)

1.4 สรุป

ในบทนี้ คุณได้เรียนรู้เกี่ยวกับ:

- การระบุอุปกรณ์
- วิธีปรับแต่งจำนวนจุดอุปกรณ์
- การบันทึกสถานะอุปกรณ์ขณะปิดหรือรีเซ็ตเครื่อง

จุด

การเปลี่ยนจำนวนจุดอุปกรณ์	• CPU โมดูลที่แตกต่างกันจะมีจำนวนจุดอุปกรณ์ที่แตกต่างกัน และจำนวนอุปกรณ์จะถูกจัดสรรเบื้องต้นตามความจุโมดูล CPU • เมื่อจำนวนของจุดที่จัดสรรไปยังอุปกรณ์ที่ใช้งานบ่อยไม่เพียงพอ ให้ลดจุดที่จัดสรรไปยังอุปกรณ์อื่นลง และนำจุดเหล่านั้นไปใช้กับอุปกรณ์ที่ใช้งานบ่อยๆ
ฟังก์ชันแลตช์	ฟังก์ชันแลตช์ของโมดูล CPU จะเก็บค่าอุปกรณ์ขณะปิดเครื่องหรือรีเซ็ต และจะใช้ข้อมูลที่เก็บไว้เมื่อเริ่มทำงานต่อ ค่าที่เก็บไว้จะถูกล้างออกด้วยคำสั่งล้างแลตช์

บทที่ 2 วิธีใช้อุปกรณ์ที่มีฟังก์ชันหลากหลาย

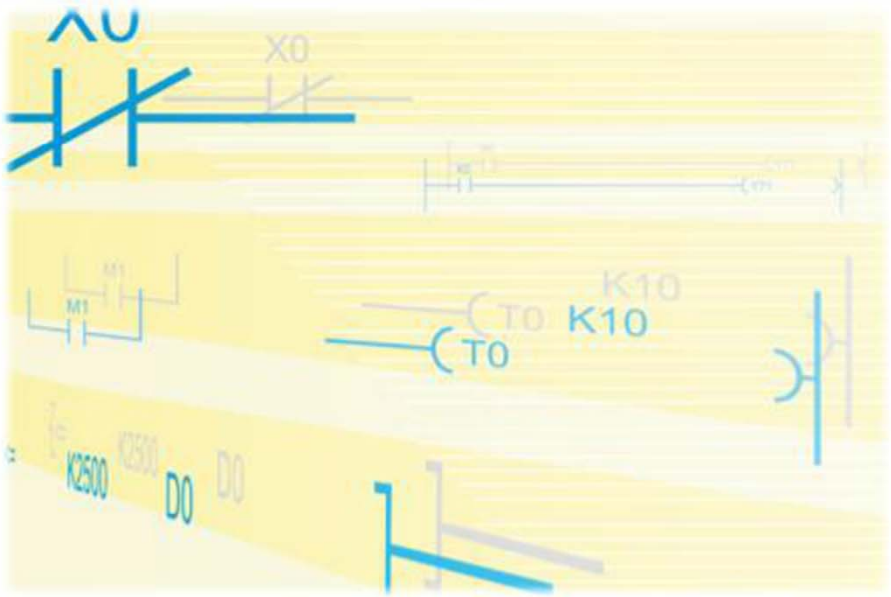
บทนี้จะอธิบายเกี่ยวกับอุปกรณ์ที่มีฟังก์ชันต่างๆ ฝังไว้
อุปกรณ์จำพวกRetentive Timerหรือ Index Registerจะมีฟังก์ชันของตัวเอง ซึ่งจะแตกต่างจากอุปกรณ์จำพวกดาต้ารีจิสเตอร์ ซึ่งสามารถจัดเก็บค่าได้เพียงอย่างเดียว

ส่วนที่ 2.1: วิธีใช้Retentive Timer

ส่วนที่ 2.2: วิธีใช้Index Register

ส่วนที่ 2.3: วิธีใช้Special RelayและSpecial Register

ส่วนที่ 2.4: สรุป



2.1 วิธีใช้Retentive Timer

2.1.1 ความแตกต่างระหว่างTimerและRetentive Timer

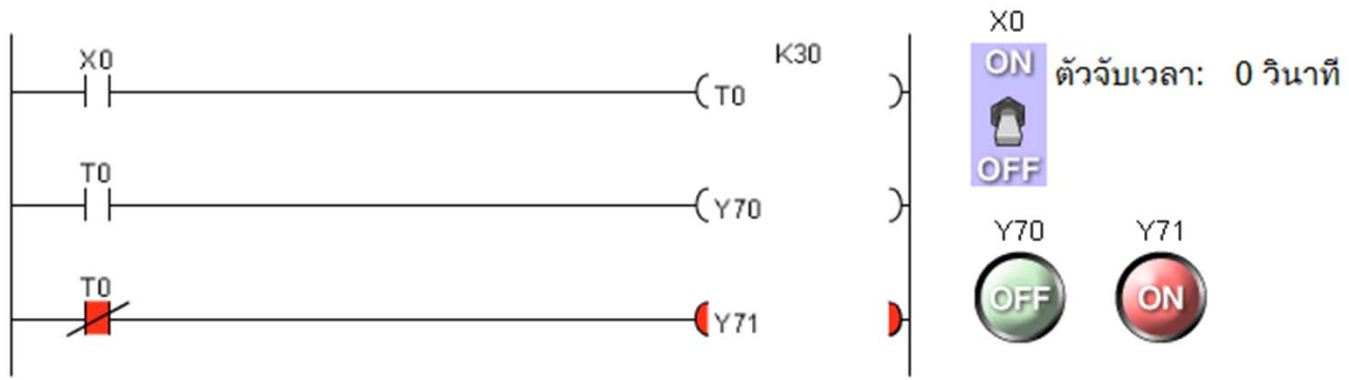
ทั้งTimerและRetentive Timer จะถูกนำไปใช้ในโปรแกรมซีเควนท์ในการทำงานที่เกี่ยวกับการจับเวลา
* คำอธิบายของTimerจะอยู่ในหลักสูตรพื้นฐาน MELSEC-Q ซีรีส์

(a) Timer

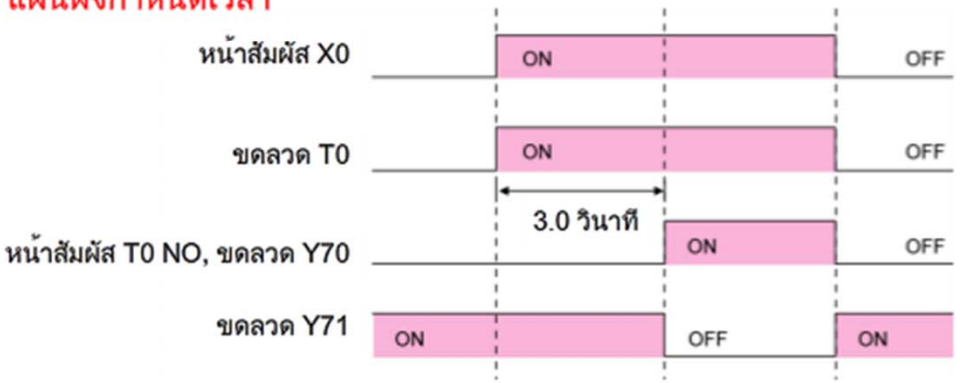
Timerจะเปิดหน้าสัมผัสในกำหนดเวลาหนึ่ง หลังจากทำการเปิดขดลวด เมื่อปิดขดลวด ค่าของTimerจะถูกรีเซ็ตเป็น "0" สัญลักษณ์อุปกรณ์ของTimerคือ "T"

โปรแกรมแลดเดอร์และการทำงาน

เปิด/ปิดสวิตช์เพื่อดูการทำงานของTimer
3 วินาที หลังจากที่ X0 เปลี่ยนเป็นเปิด Y70 จะเปลี่ยนเป็นเปิดตาม และ Y71 จะเปลี่ยนเป็นสถานะปิด



แผนผังกำหนดเวลา

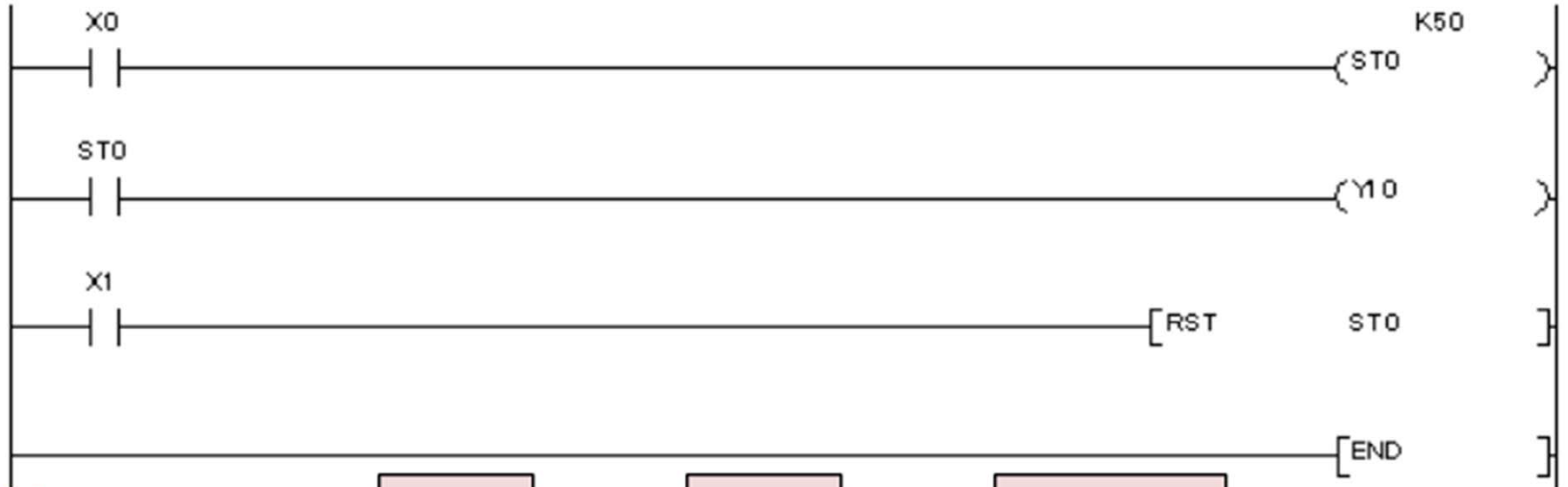


2.1.1 ความแตกต่างระหว่างTimerและRetentive Timer

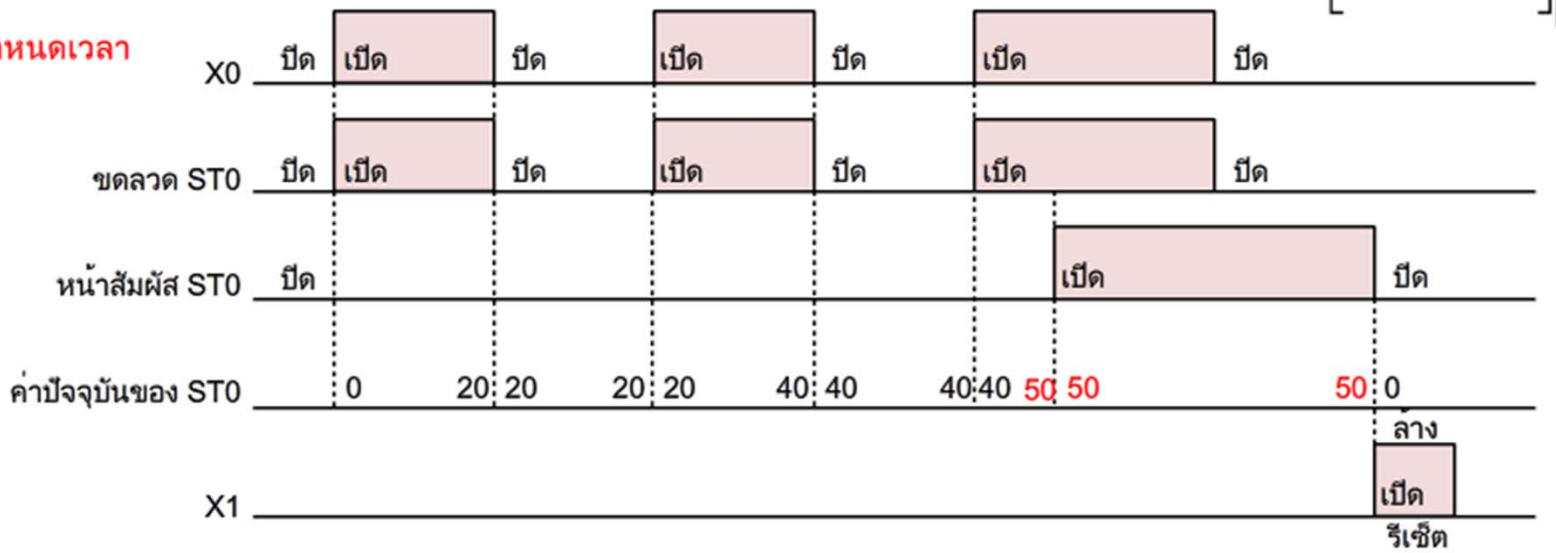
(b) Retentive Timer

Retentive Timerจะมีประโยชน์สำหรับการวัดRunning Timesรวม
Retentive Timerจะเปิดหน้าสัมผัส (ปิดเป็นเปิด) เมื่อถึงช่วงเวลาหนึ่งหลังการเปิดขดลวด เมื่อขดลวดถูกปิด ค่าของTimer จะไม่ถูกรีเซ็ต และจะถูกเก็บไว้ เมื่อขดลวดเปิดอีกครั้ง Timerจะเริ่มทำการจับเวลาอีกครั้งต่อจากค่าที่เก็บไว้ สัญลักษณ์อุปกรณ์ของRetentive Timerคือ "ST"

โปรแกรมแลดเดอร์



แผนผังกำหนดเวลา



2.1.2 การทำงานของRetentive Timer

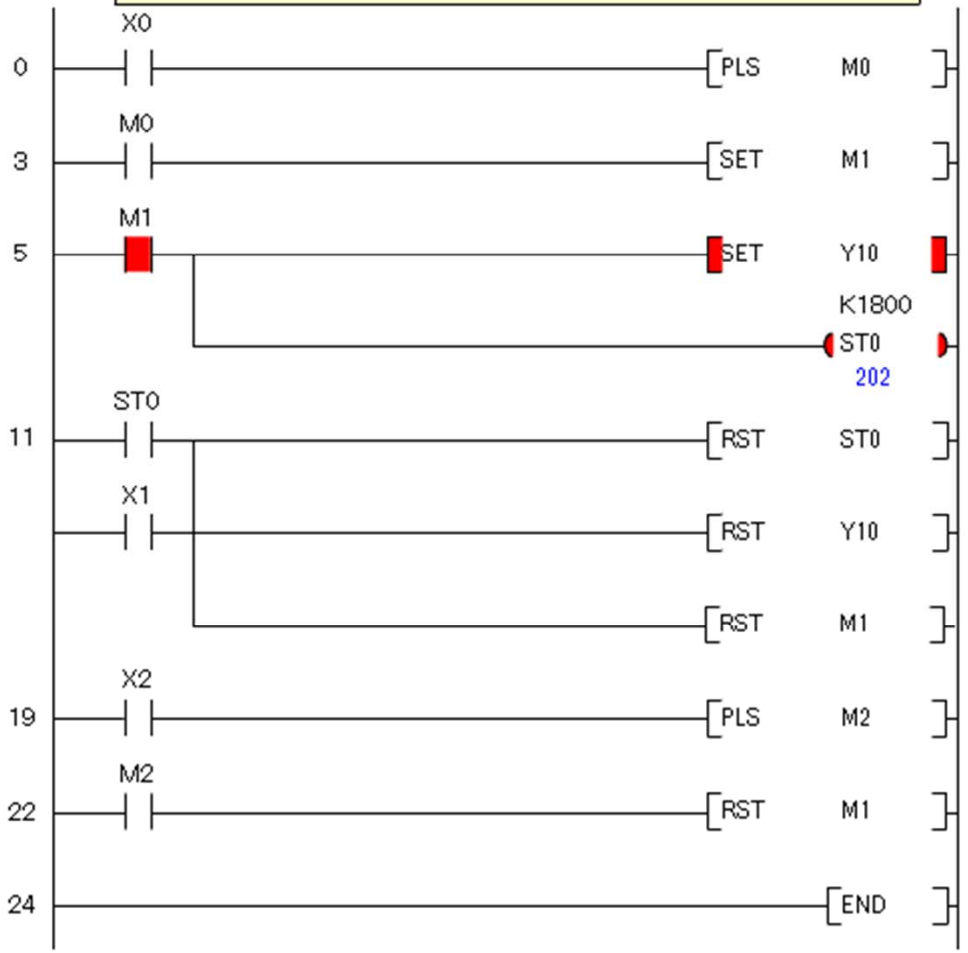
มาดูกันว่าRetentive Timerทำงานบนเครื่องจักรที่ใช้อินพุทสวิตช์อย่างไร (X0 ถึง X2)
* Retentive Timer (ST0) จะถูกตั้งค่าในหน่วย 100 ms

ตัวจับเวลา ST0 ถูกตั้งค่าเป็น K1800 = 180,000 ms (3 นาที)/100 ms

Running time (จับเวลาโดยTimer) 20 วินาที

อินพุท สวิตช์

- X0: เปิด เริ่มทำงานหรือเริ่มทำงานใหม่จากสถานะหยุดชั่วคราว — OFF
- X1: เปิด สิ้นสุดการทำงานและรีเซ็ตเวลาที่จับไว้ — OFF
- X2: เปิด หยุดการทำงานชั่วคราว และเก็บค่าเวลาที่จับไว้ — OFF



Y10: สัญญาณเริ่มต้น

ST0: Retentive Timer

2.1.3 การเตรียมการเพื่อใช้Retentive Timer

จำนวนของจุดที่Retentive Timerใช้ในเบื้องต้นจะเป็น "0" ในการใช้Retentive Timer จะต้องมีการจัดสรรบางจุด
เปิดหน้าต่าง PLC Parameter (PLC พารามิเตอร์) ของ GX Works2 เลือกแท็บ Device (อุปกรณ์) และตั้งค่าจำนวนจุดอุ
ปกรณ์ที่Retentive Timerใช้
ด้านล่างคือตัวอย่างการตั้งค่าเพื่อใช้ ST0 ถึง ST63 (64 จุด) สำหรับRetentive Timer

	Sym.	Dig.	Device Points	Latch (1) Start	Latch (1) End	Latch (2) Start	Latch (2) End	Local Device Start	Local Device End
Input Relay	X	16	8K						
Output Relay	Y	16	8K						
Internal Relay	M	10	8K						
Latch Relay	L	10	8K						
Link Relay	B	16	8K						
Annunciator	F	10	2K						
Link Special	SB	16	2K						
Edge Relay	V	10	2K						
Step Relay	S	10	8K						
Timer	T	10	2K						
Retentive Timer	ST	10	64						
Counter	C	10	1K						
Data Register	D	10	12K						
Link Register	W	16	8K						
Link Special	SW	16	2K						
Index	Z	10	20						

Device Total	28.9	K Words
Word Device	25.1	K Words
Bit Device	44.2	K Bits

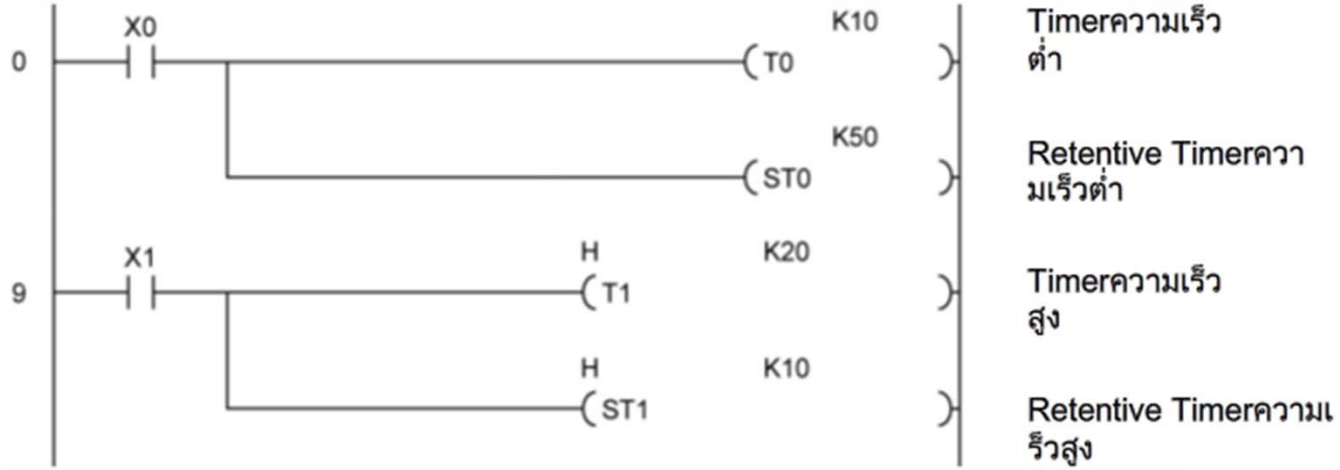
The total number of device points is up to 29 K words.
Latch(1) : Able to clear the value by using a latch clear.
Latch(2) : Unable to clear the value by using a latch clear. Clearing will be executed by remote operation or program.
Scan time is extended by the latch range setting (including L).
If the latch is necessary, please set the required minimum latch range.
When using the local devices, please do the file setting at PLC file setting parameter.

2.1.4 ความแตกต่างระหว่างTimerความเร็วต่ำ และTimerความเร็วสูง

	หน่วย	ตัวอย่างโปรแกรม	การทำงาน
Timerความเร็วต่ำ	100 ms (ค่าเริ่มต้น)	K50 {T0 }	Timerความเร็วต่ำ T0 จะนับทุกๆ 5 วินาที
Timerความเร็วสูง	10 ms (ค่าเริ่มต้น)	H K50 {T1 }	Timerความเร็วสูง T1 จะนับ 0.5 วินาที
Retentive Timerความเร็วต่ำ	100 ms (ค่าเริ่มต้น)	K50 {ST0 }	Retentive Timerความเร็วต่ำ ST0 จะนับ 5 วินาที
Retentive Timerความเร็วสูง	10 ms (ค่าเริ่มต้น)	H K50 {ST1 }	Retentive Timerความเร็วสูง ST1 จะนับ 0.5 วินาที

หน่วยเบื้องต้นในการจับเวลาคือ 100 ms สำหรับTimerความเร็วต่ำ และ 10 ms สำหรับTimerความเร็วสูง ดูวิธีการเปลี่ยนหน่วยได้ที่หน้าถัดไป

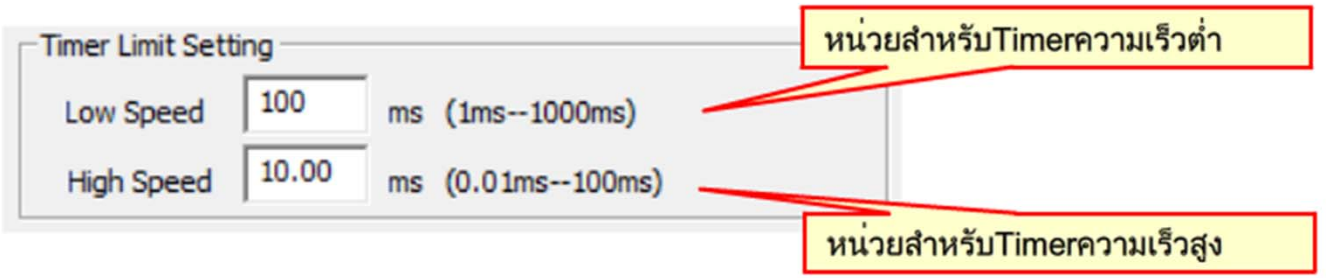
ด้านล่างคือตัวอย่างโปรแกรมแลดเดอร์ซึ่งมีTimer



2.1.4 ความแตกต่างระหว่างTimerความเร็วต่ำและTimerความเร็วสูง

วิธีเปลี่ยนหน่วยTimer

เปลี่ยน Timer Limit Setting (การตั้งค่าขีดจำกัดTimer)บนแท็บ PLC System (ระบบ PLC) ของ หน้าต่าง PLC Parameter (พารามิเตอร์ PLC)
ด้านล่างคือตัวอย่างการตั้งค่าบนหน้าจอ PLC System (ระบบ PLC)



2.2 วิธีใช้Index Register

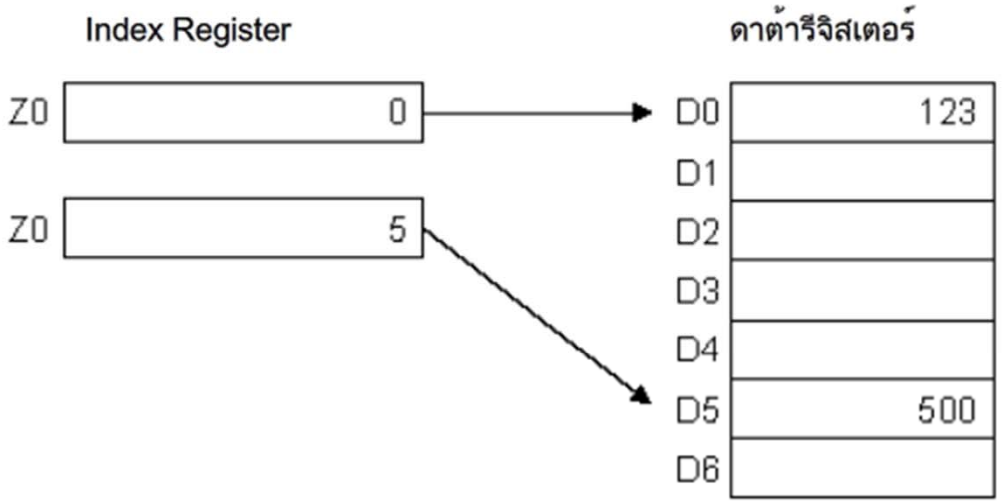
Index Register "Z" เมื่อทำงานร่วมกับอุปกรณ์อื่น จะทำหน้าที่กำหนด (ทำดัชนี) หมายเลขอุปกรณ์ของอุปกรณ์ที่จะควบคุม
Index Registerจะเป็นประโยชน์ในการลดความซับซ้อนโปรแกรม เนื่องจากจะสามารถบรรยายถึงอุปกรณ์หลายรายการแบบเป็นชุดได้

•เมื่อใช้งานIndex Register จะมีการเขียนไว้ถัดจากสัญลักษณ์อุปกรณ์ และหมายเลขอุปกรณ์จะปรากฏขึ้นด้านล่าง เพื่อระบุอุปกรณ์เป้าหมายการควบคุมจริง
อุปกรณ์เป้าหมายการควบคุมจริง = สัญลักษณ์อุปกรณ์ (หมายเลขอุปกรณ์ + Index Register)
•สามารถใช้ได้ 16 จุด ตั้งแต่ Z0 ถึง Z15 ในIndex Register

ตัวอย่างIndex Register

เมื่ออุปกรณ์ถูกระบุเป็น "D0Z0" จะหมายความถึง D (0 + Z0) ดังนั้น หมายเลขอุปกรณ์จะเท่ากับ "0 + (ค่าของ Z0)"

ตัวอย่าง: เมื่อ Z0 = 0 หมายเลขอุปกรณ์คือ D0
เมื่อ Z0 = 5 หมายเลขอุปกรณ์คือ D5



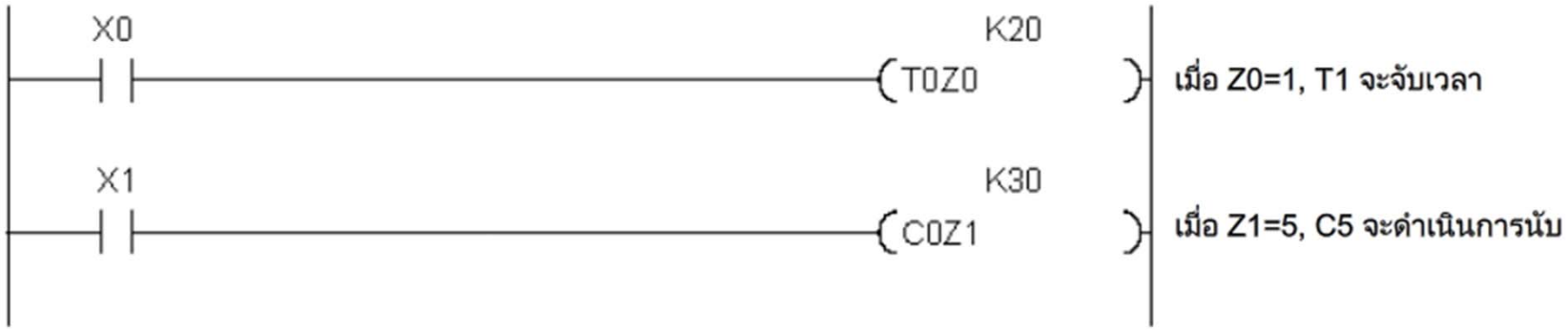
2.2 วิธีใช้Index Register

อุปกรณ์ที่สามารถทำดัชนีได้โดยIndex Register

สามารถทำดัชนีอุปกรณ์ต่อไปนี้ได้โดยใช้Index Register:

อุปกรณ์บิท	X, Y, M, L, S, B, F
เวิร์ดดีไวส์	T, C, D, R, W
ค่าคงที่	K, H
ตัวชี้	P

หมายเหตุ: สำหรับหน้าสัมผัสและขดลวดที่ใช้ในTimerและCounter จะสามารถใช้ได้เฉพาะIndex Register Z0 และ Z1



2.2 วิธีใช้ Index Register

การลดความซับซ้อนของโปรแกรมโดยใช้Index Register

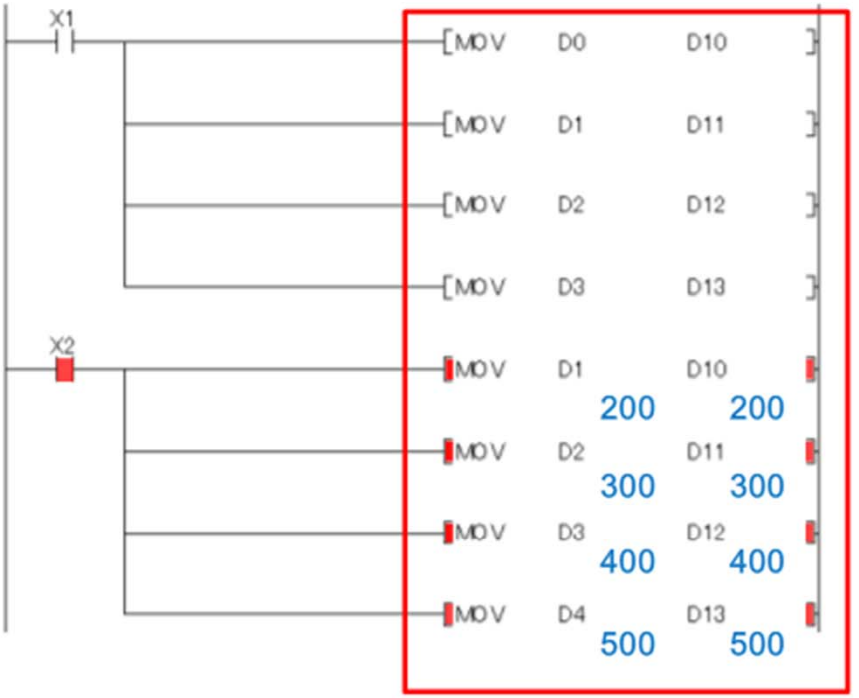
Index Registerด้านล่าง จะโอนค่าใน "D0 ถึง D4" ไปยัง "D10 ถึง D13" เมื่อ X1 หรือ X2 ถูกเปิด โปรแกรม (1) และ (2) จะให้ผลลัพธ์เดียวกัน

ในโปรแกรม (1) ข้อมูลจะถูกโอนโดยตรง

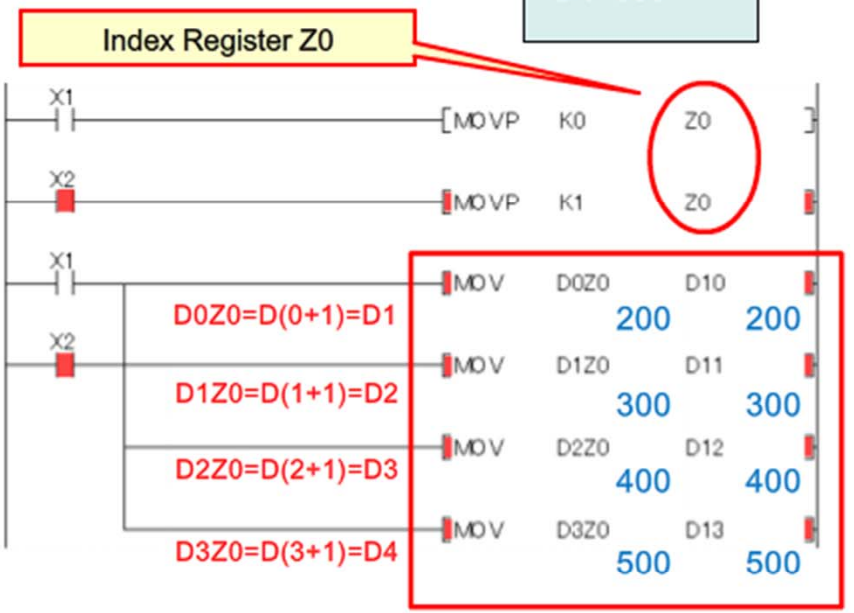
ในโปรแกรม (2) ข้อมูลจะถูกโอนผ่านIndex Register

ค่าเริ่มต้น
D0=100
D1=200
D2=300
D3=400
D4=500

(1) ตัวอย่างแบบไม่มี Index Register



(2) ตัวอย่างแบบใช้ Index Register



การลดความซับซ้อนของโปรแกรม

2.2.1 การทำงานIndex Register

คลิกอินพุทสวิตช์ X0 ถึง X5 เพื่อดูวิธีการทำงานของIndex Register Z0
*K0 ถึง K400 จะถูกจัดเก็บไว้ในดาต้ารีจิสเตอร์ D0 ถึง D4

Data register

D0	0
D1	100
D2	200
D3	300
D4	400

Index register Z0 = 0

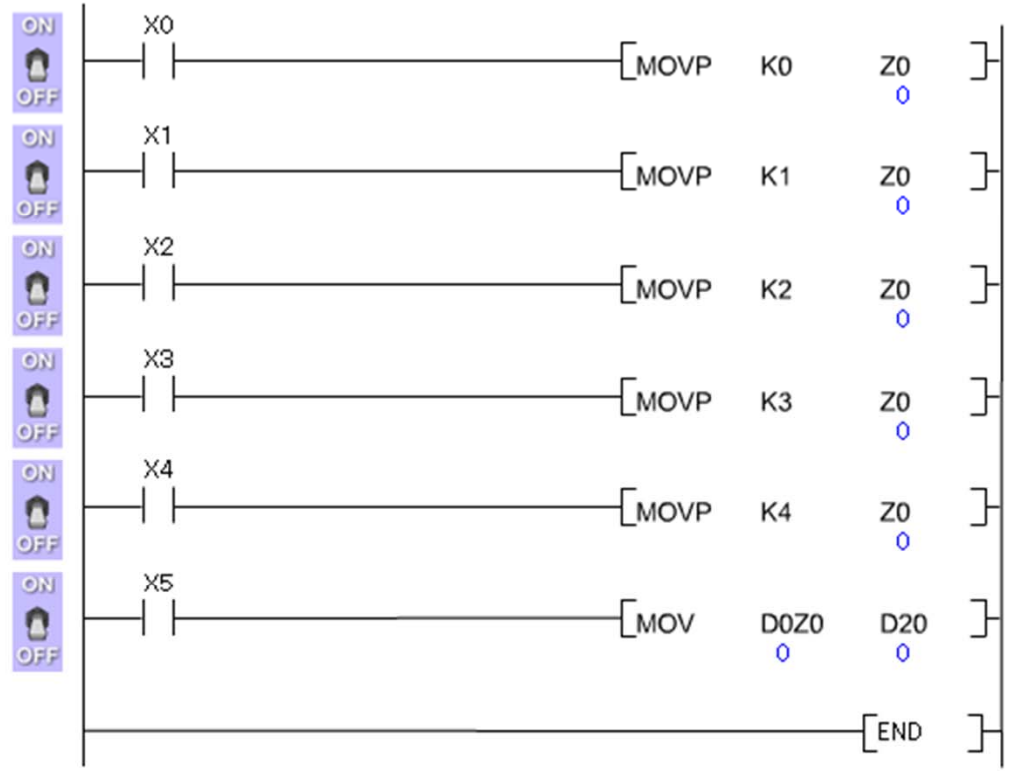
Data register D20 = 0

รีเซ็ตการ simulation

รีเซ็ต

อินพุทสวิตช์

- X0:เปิด — "0" จะถูกโอนไปยัง Index Register Z0
- X1:เปิด — "1" จะถูกโอนไปยัง Index Register Z0
- X2:เปิด — "2" จะถูกโอนไปยัง Index Register Z0
- X3:เปิด — "3" จะถูกโอนไปยัง Index Register Z0
- X4:เปิด — "4" จะถูกโอนไปยัง Index Register Z0
- X5:เปิด — ค่าของดาต้ารีจิสเตอร์ที่กำหนดโดย Z0 จากในการลงทะเบียน D0 และ D4 จะถูกโอนไปยัง "D20"



2.3 วิธีใช้Special RelayและSpecial Register

Special RelayและSpecial Registerที่ใช้ในโมดูล CPU มีฟังก์ชันและการทำงานที่กำหนดไว้ล่วงหน้า รีเลย์ภายในที่ใช้สำหรับข้อมูลบิท (เปิด/ปิด) จะมีชื่อว่า "Special Relay (SM)" และInternal Registerที่ใช้สำหรับข้อมูลเวิร์ด จะถูกเรียกว่า "Special Register (SD)"

ในโปรแกรม จะถูกใช้เป็นเงื่อนไขการตัดสินสำหรับการทำงาน โดยจะถูกใช้เป็นเงื่อนไขการตรวจสอบ ซึ่งสามารถกำหนดบนการตรวจสอบอุปกรณ์ของ GX Works2 ได
Special RelayและSpecial Registerจะถูกจัดหมวดหมู่ตามประเภทด้านล่าง

ข้อมูลการวินิจฉัย: SM0 ถึง 199, SD0 ถึง 199
จัดเก็บผลการวินิจฉัยของโมดูล CPU
ข้อมูลerrorในการวินิจฉัยและรหัสerrorต่างๆ

ข้อมูลระบบ: SM200 ถึง 399, SD200 ถึง 399
จัดเก็บข้อมูลระบบของโมดูล CPU
ข้อมูลโมดูล CPU, ข้อมูลนาฬิกา, ฯลฯ

นาฬิกา/Counterระบบ: SM400 ถึง 499, SD400 ถึง 499
สัญญาณนาฬิกาและค่าcounterที่ใช้เป็นองค์ประกอบการกำหนดเวลาพื้นฐาน
สัญญาณนาฬิกาต่างๆ

ข้อมูลการสแกน: SM500 ถึง 599, SD500 ถึง 599
จัดเก็บข้อมูลการดำเนินการสแกนของโปรแกรม
ข้อมูลเวลาการสแกนต่างๆ

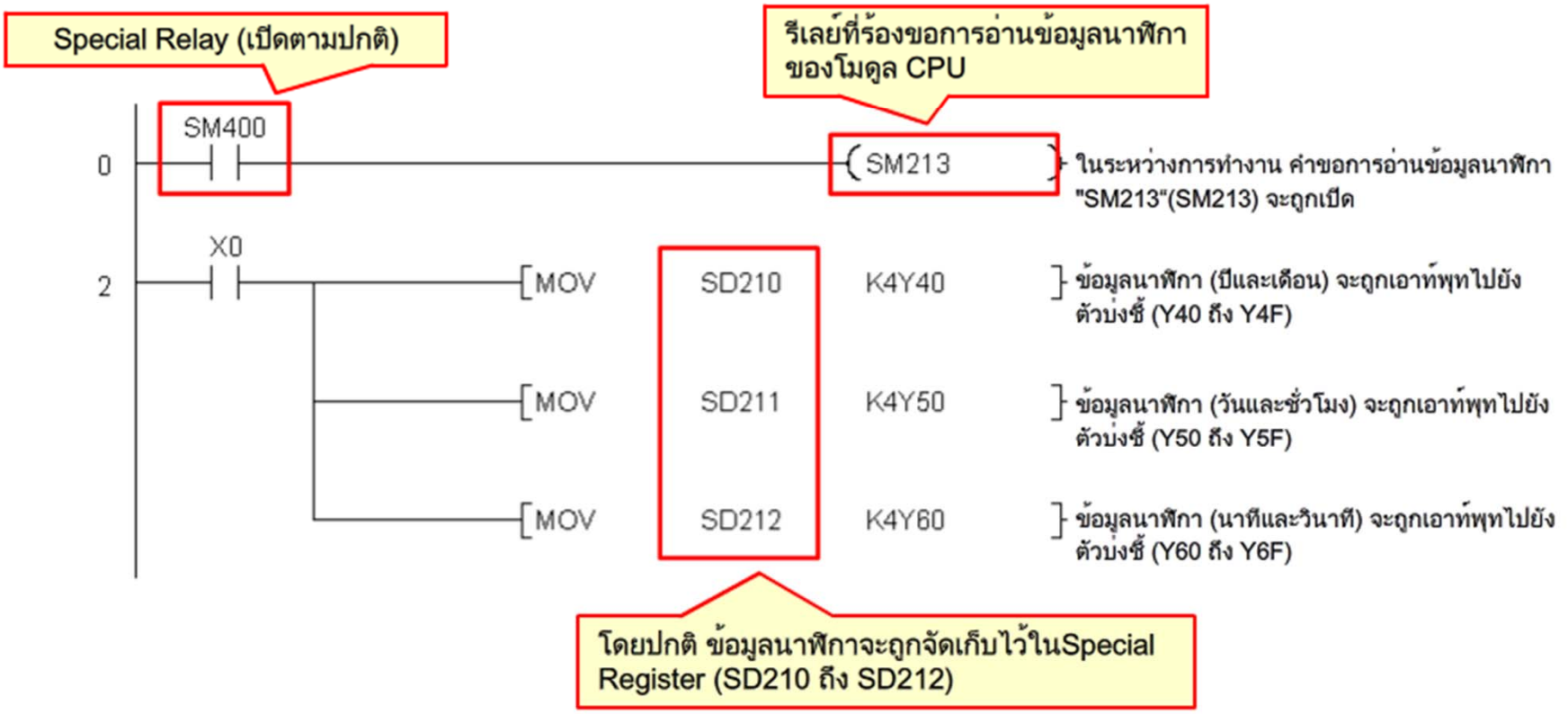
ข้อมูลการ์ดหน่วยความจำ: SM600 ถึง 699, SD600 ถึง 699
จัดเก็บข้อมูลการ์ด เช่น การใช้การ์ดหน่วยความจำ และไฟล์รีจิสเตอร์
เปิดใช้งาน/ปิดใช้งานการ์ดหน่วยความจำ

ข้อมูลคำสั่ง: SM700 ถึง 799, SD700 ถึง 799
จัดเก็บสถานะการดำเนินงานและข้อมูลควบคุมเกี่ยวกับคำสั่งพิเศษ
แฟล็กการดำเนินงานคำสั่ง

ข้อมูลการแก้จุดบกพร่อง: SM800 ถึง 899, SD800 ถึง 899
จัดเก็บข้อมูลเกี่ยวกับการแก้จุดบกพร่อง
การตรวจสอบสถานะการติดตามย้อนกลับ

2.3.1 โปรแกรมตัวอย่างที่ใช้รีเลย์/Special Register

โปรแกรมตัวอย่างในการร้องขอข้อมูลนาฬิกาของโมดูล CPU



2.4 สรุป

ในบทนี้ คุณได้เรียนรู้เกี่ยวกับ:

- วิธีใช้Retentive Timer
- วิธีใช้Index Register
- วิธีใชSpecial RelayและSpecial Register

จุด

การใช้Retentive Timer	•ในการใช้Retentive Timer จะต้องจัดสรรบางจุดลงในหน้าต่าง PLC Parameter (PLC พารามิเตอร์) •เวลาที่จับ (ค่าปัจจุบัน) และสถานะหน้าสัมผัส (เปิด/ปิด) ของRetentive Timerจะไม่ถูกล้าง แม้ว่าจะมีการเปลี่ยนแปลงจนไม่เป็นไปตามเงื่อนไขการอินพุทหลังจากหมดเวลา •โปรแกรมจะต้องการแลดเดอร์ในการรีเซ็ตตัวจับเวลาแบบเก็บค่า (จะมีการใช้คำสั่ง RST)
การใช้Index Register	• Index Register "Z" จะถูกระบุไว้ถัดจากอุปกรณ์ที่ใช้ในโปรแกรม ตัวอย่างเช่น "D0Z5" • สามารถใชงานได้ 16 จุด จาก Z0 ถึง Z15 สำหรับIndex Register
ฟังก์ชันของ Special RelayและSpecial Register	Special RelayและSpecial Registerจะถูกใช้เพื่อระบุเงื่อนไขภายในของโมดูล CPU รวมถึงข้อมูลการวินิจฉัยและข้อมูลระบบ

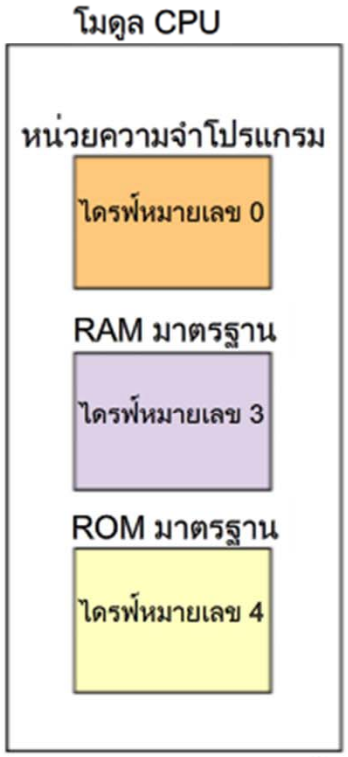
บทที่ 3 หน่วยความจำของโมดูล CPU และไฟล์รีจิสเตอร์

บทนี้จะอธิบายเกี่ยวกับหน่วยความจำที่โมดูล CPU สามารถใช้ได้ และวิธีใช้ไฟล์รีจิสเตอร์

ส่วนที่ 3.1: หน่วยความจำของโมดูล CPU
ส่วนที่ 3.2: วิธีใช้ไฟล์รีจิสเตอร์
ส่วนที่ 3.3: สรุป

3.1 หน่วยความจำของโมดูล CPU

CPU โมดูลสามารถใช้หน่วยความจำได้สองประเภท ประเภทที่อยู่ภายในโมดูล CPU และการ์ดหน่วยความจำที่สามารถเสียบลงในสล็อตของโมดูล CPU ได้ เพื่อให้มั่นใจในการเข้าถึงโมดูล CPU ของ GX Works2 จะต้องทำการระบุหมายเลขไดรฟ์ซึ่งจะกำหนดประเภทหน่วยความจำเป้าหมายอย่างถูกต้อง



ประเภทหน่วยความจำ		ประเภทข้อมูลที่จัดเก็บไว้	เงื่อนไขข้อมูลขณะปิดเครื่อง	การฟอร์แมตหน่วยความจำเพื่อการใช้งานครั้งแรก
โมดูล CPU	หน่วยความจำโปรแกรม	•พารามิเตอร์ •โปรแกรม	จัดเก็บโดยใช้แบตเตอรี่ของโมดูล CPU	ต้องการ (ใช้ GX Works2)
	RAM มาตรฐาน	•ลงทะเบียนไฟล์ •อุปกรณ์โลคอล		
	ROM มาตรฐาน	•พารามิเตอร์ •โปรแกรม	จัดเก็บโดยไม่ต้องใช้แบตเตอรี่	ไม่ต้องการ
การ์ดหน่วยความจำ	RAM	•พารามิเตอร์ •โปรแกรม •ลงทะเบียนไฟล์ •อุปกรณ์โลคอล	จัดเก็บโดยใช้แบตเตอรี่ของการ์ดหน่วยความจำ	ต้องการ (ใช้ GX Works2)
	ROM	•พารามิเตอร์ •โปรแกรม •ลงทะเบียนไฟล์	จัดเก็บโดยไม่ต้องใช้แบตเตอรี่	ไม่ต้องใช้แฟลชการ์ด ต้องการ ATA การ์ด (ใช้ GX Works2)

- โปรแกรมที่จัดเก็บไว้ใน ROM มาตรฐานหรือการ์ดหน่วยความจำ จะถูกบูท (โหลด) ไปยังหน่วยความจำโปรแกรมของโมดูล CPU และเรียกใช้งานเมื่อโมดูล CPU เริ่มทำงาน
- เมื่อไฟล์รีจิสเตอร์ถูกบันทึกไว้ใน RAM มาตรฐาน ความเร็วในการเข้าถึงไฟล์รีจิสเตอร์ จะเร็วเท่าๆ กับการเข้าถึงดาต้ารีจิสเตอร์ (D)
- ขณะใช้งาน RAM มาตรฐาน การปิดเครื่องโดยไม่มีแบตเตอรี่สำรอง จะเป็นการล้างข้อมูลที่จัดเก็บไว้ใน RAM
- โดยทั่วไป จะมีการใช้ RAM อ่าน/เขียนความเร็วสูง เพื่อเริ่มต้นการทำงานของระบบ และใช้ ROM ในการทำงานของระบบอย่างต่อเนื่อง

3.2 วิธีใช้ไฟล์รีจิสเตอร์

ภาพรวมของไฟล์รีจิสเตอร์

- ไฟล์รีจิสเตอร์คือเวิร์ดดีไวส์ที่ใช้ในการขยายดาต้ารีจิสเตอร์ (D)
- เมื่อเปรียบเทียบกับดาต้ารีจิสเตอร์ ไฟล์รีจิสเตอร์จะสามารถรองรับข้อมูลได้ในปริมาณมาก
- ไฟล์รีจิสเตอร์จะถูกจัดเก็บไว้ใน RAM มาตรฐานของโมดูล CPU หรือในการ์ดหน่วยความจำ (RAM)
- ข้อมูลที่จัดเก็บไว้ในไฟล์รีจิสเตอร์ จะไม่ถูกลางออก แม้ในขณะที่ปิดเครื่อง หรือในขณะที่รีเซ็ตโมดูล CPU
- สัญลักษณ์อุปกรณ์คือ "ZR"

การทำงานของโปรแกรมแลดเดอร์

ปิดสวิตช์เครื่องและอินพุทสวิตช์เปิด/ปิด เพื่อจำลองการทำงานของไฟล์รีจิสเตอร์

สลับสวิตช์แหล่งจ่ายจากเปิดเป็นปิด จากนั้นกลับไปที่เปิดอีกครั้ง เพื่อตรวจสอบว่าข้อมูลในไฟล์รีจิสเตอร์ ZR0 และ ZR1 นั้นยังคงอยู่

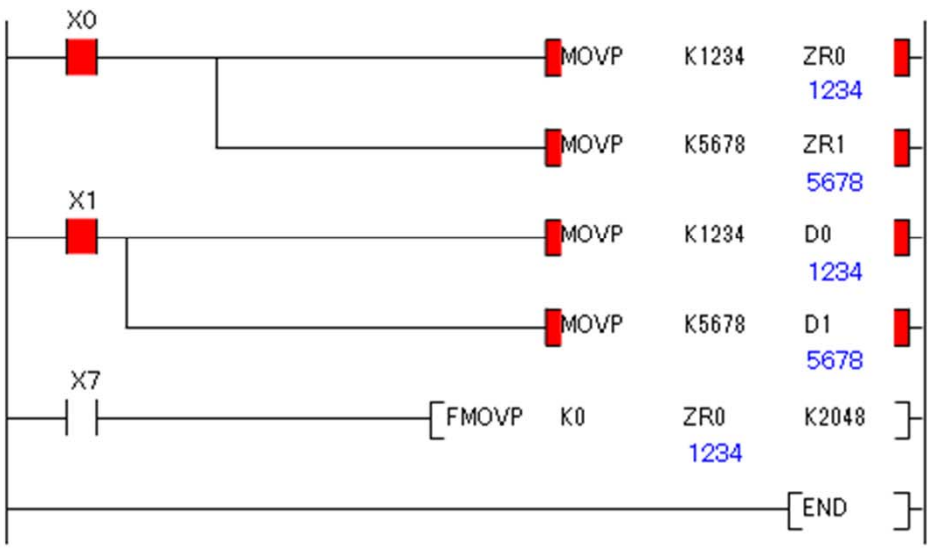
รีเซ็ตการ simulation

รีเซ็ต

File register		Data register	
ZR0	1234	D0	1234
ZR1	5678	D1	5678

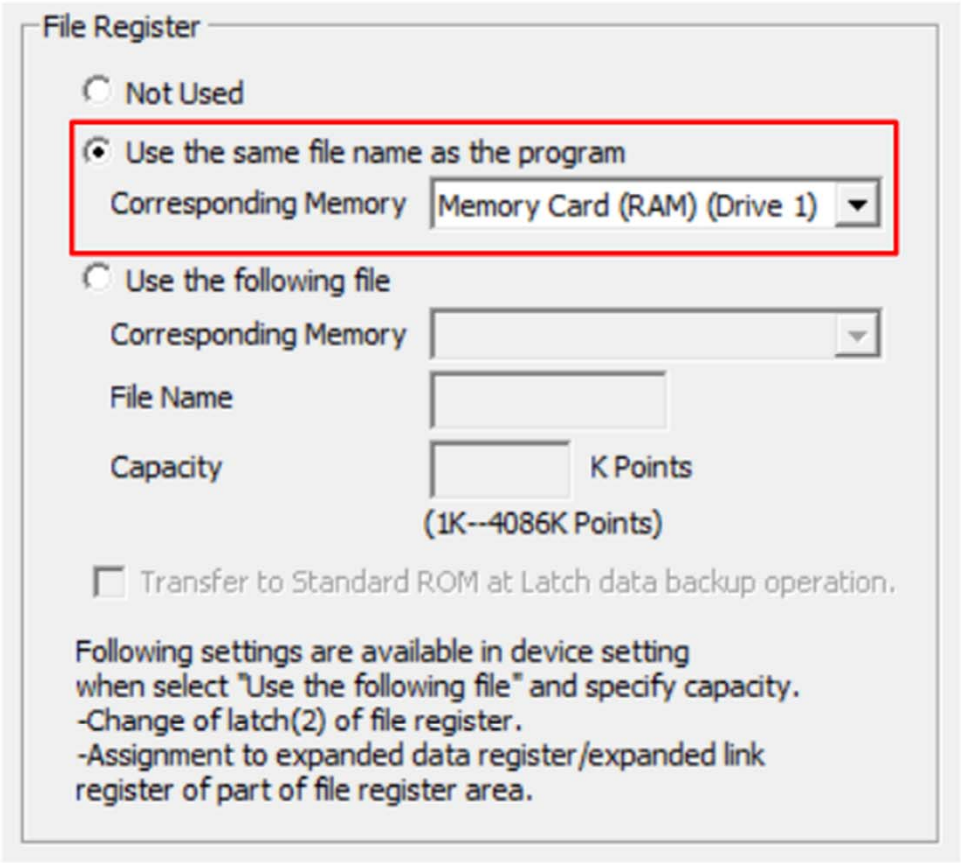
อินพุทสวิตช์

- X0: เปิด — ข้อมูลจะถูกเขียนลงในไฟล์รีจิสเตอร์ ZR0 และ ZR1
- X1: เปิด — ข้อมูลจะถูกเขียนลงในดาต้ารีจิสเตอร์ D0 และ D1
- X7: เปิด — ข้อมูลในไฟล์รีจิสเตอร์ ZR0 และ ZR1 จะถูกตั้งค่ากลับไปที่ "0"



3.2 วิธีใช้ไฟล์รีจิสเตอร์

ส่วนนี้จะอธิบายเกี่ยวกับการตั้งค่าที่กำหนดไฟล์รีจิสเตอร์โลคอลเป็นที่จัดเก็บปลายทาง ในหน้าต่าง PLC Parameter (PLC พารามิเตอร์) ให้เลือกแท็บ PLC File (PLC ไฟล์) จากนั้น เลือก "Use the same file name as the program" (ใช้ชื่อไฟล์เดียวกับโปรแกรม) สำหรับ File Register (ไฟล์รีจิสเตอร์) และกำหนดหน่วยความจำที่จัดเก็บปลายทาง
โปรดทราบว่าจะต้องใช้การ์ดหน่วยความจำสำหรับการตั้งค่านี้ (RAM มาตรฐาน สามารถจัดเก็บได้เพียงไฟล์รีจิสเตอร์เดียว)



สำหรับ "Corresponding Memory" (หน่วยความจำที่ตรงกัน) ให้เลือก "Memory card (RAM)" (การ์ดหน่วยความจำ (RAM))
จะต้องทำการตั้งค่าไฟล์รีจิสเตอร์สำหรับแต่ละโปรแกรม การตั้งค่าจะถูกเขียนลงบนโมดูล CPU ที่ PLC Write

3.3 สรุป

ในบทนี้ คุณได้เรียนรู้เกี่ยวกับ:

- หน่วยความจำของโมดูล CPU
- วิธีใช้ไฟล์รีจิสเตอร์

จุด

การใช้ไฟล์รีจิสเตอร์	ในการใช้ไฟล์รีจิสเตอร์ จะต้องเลือก RAM มาตรฐานของโมดูล CPU หรือการ์ดหน่วยความจ ำเป็นที่เก็บข้อมูลปลายทาง สำหรับการตั้งค่า ให้ไปที่แท็บ PLC File (ไฟล์ PLC) ของหน้าต่า ง PLC Parameter (PLC พารามิเตอร์) ไฟล์รีจิสเตอร์จะเก็บข้อมูลไว้ แม้ในขณะที่ปิดเครื่อง

บทที่ 4 โปรแกรมที่มีจำนวนจริง

บทนี้จะอธิบายวิธีการที่โปรแกรมดำเนินการกับจำนวนจริง และคำสั่งการทำงาน

ส่วนที่ 4.1: การใช้งานและสัญลักษณ์ของจำนวนจริง
ส่วนที่ 4.2: คำสั่งการทำงานของจำนวนจริง
ส่วนที่ 4.3: คำสั่งการแปลงระหว่างจำนวนเต็มและจำนวนจริง
ส่วนที่ 4.4: สรุป

4.1 การใช้งานและสัญลักษณ์ของจำนวนจริง

การใช้งานของจำนวนจริง

- "จำนวนจริง" คือค่าตัวเลขรวมจุดทศนิยม
- โดยปกติ โปรแกรมซีเควนท์จะถูกกำหนดค่าโดยใช้จำนวนเต็ม อย่างไรก็ตาม จะต้องมีการใช้จำนวนจริงพร้อมจุดทศนิยมในโปรแกรมที่การทำงานทางคณิตศาสตร์ขั้นสูง เช่น ฟังก์ชันตรีโกณมิติและเอ็กซ์โพเนนเชียล
- ข้อมูลตัวเลขของจำนวนจริงจะถูกอ้างอิงถึงเป็น "ข้อมูลโฟลทติงพอยน์"

ข้อควรระวัง

- จำนวนจริงหนึ่งค่า จะใช้สองเวิร์ดดีไวส์ต่อเนื่องกันเสมอ (ใช้พื้นที่หน่วยความจำ 32 บิท) โดยไม่คำนึงถึงขนาดของตัวเลข
- ในโปรแกรมซีเควนท์ จะสามารถใช้ คำสั่งการทำงานเฉพาะ (บวก ลบ คูณ หาร ฟังก์ชันพิเศษ ฯลฯ) ซึ่งดำเนินการกับจำนวนจริงได้ สามารถใช้คำสั่งการแปลง เช่น ระหว่างจำนวนเต็มและจำนวนจริงได้ด้วย

สัญลักษณ์จำนวนจริง

"E" จะถูกใช้แทนจำนวนจริง

(1) การแสดงค่าคงที่ด้วยจำนวนจริง

ในการเขียนค่าคงที่ ให้เริ่มต้นด้วย "E"

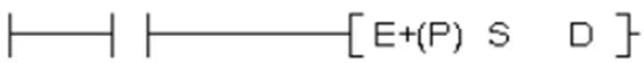
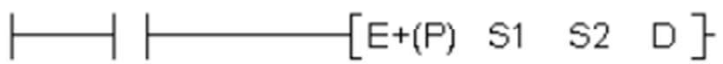
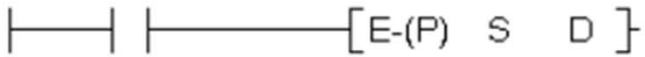
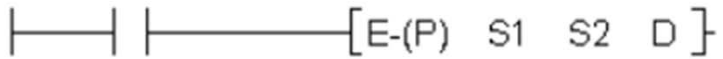
การแสดงแบบปกติ	เขียนค่าตัวเลขตามที่เป็น (ตัวอย่าง) 10.2345 เขียนเป็น "E10.2345".
การแสดงแบบเอ็กซ์โพเนนเชียล	เขียนค่าตัวเลขตามที่เป็น "(ค่าตัวเลข)x10^{n}" (ตัวอย่าง) 1234.0 เขียนเป็น "E1.234+3"

(2) คำสั่งที่มีจำนวนจริง

เพิ่ม "E" ที่ด้านหน้าของคำสั่ง
เช่น คำสั่งโอน จะเป็น "EMOV" และคำสั่งบวกหรือลบจะเป็น "E+" หรือ "E−"

4.2 คำสั่งการทำงานของจำนวนจริง

4.2.1 คำสั่งบวกและลบ

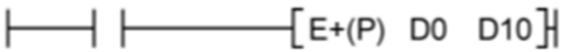
รหัสคำสั่ง	ตัวอย่างแลดเดอร์	
E+ (บวก)	 จะเป็นการดำเนินการแบบจำนวนจริง "D + S = D"	 จะเป็นการดำเนินแบบจำนวนจริง "S1 + S2 = D"
E- (ลบ)	 จะเป็นการดำเนินการแบบจำนวนจริง "D - S = D"	 จะเป็นการดำเนินแบบจำนวนจริง "S1 - S2 = D"

S (ต้นทาง): ข้อมูลก่อนการทำงาน (ค่าคงที่ หมายเลขอุปกรณ์)
D (ปลายทาง): ปลายทางของข้อมูลหลังการทำงาน (หมายเลขอุปกรณ์)
P: คำสั่งที่จะดำเนินการบนขอบขึ้น
S1 และ S2: สองรายการข้อมูลที่จะทำงาน

หมายเหตุ:
ในการทำงานแบบจำนวนจริง S1, S2 และ D ในแลดเดอร์จะต้องเป็นจำนวนจริงทั้งหมด
จำนวนเต็มและจำนวนจริงจะต้องไม่ปนกันในการทำงาน

4.2.1 คำสั่งบวกและลบ

ตัวอย่างโปรแกรมที่มีคำสั่งเพิ่มเติม

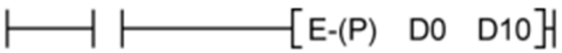


จำนวนจริงแบบโฟลทติงพอยน์ (32 บิท)			จำนวนจริงแบบโฟลทติงพอยน์ (32 บิท)			จำนวนจริงแบบโฟลทติงพอยน์ (32 บิท)	
D11	D10	+	D1	D0	=	D11	D10
2.54			10.55			13.09	

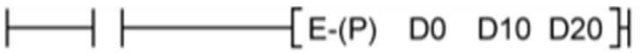
จำนวนจริงแบบโฟลทติงพอยน์ (32 บิท)			จำนวนจริงแบบโฟลทติงพอยน์ (32 บิท)			จำนวนจริงแบบโฟลทติงพอยน์ (32 บิท)	
D1	D0	+	D11	D10	=	D21	D20
1000.000			3.140			1003.140	

4.2.1 คำสั่งบวกและลบ

ตัวอย่างโปรแกรมที่มีคำสั่งลบ

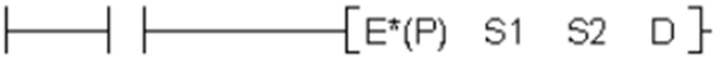


จำนวนจริงแบบโฟลทติงพอยน์ (32 บิท)			จำนวนจริงแบบโฟลทติงพอยน์ (32 บิท)			จำนวนจริงแบบโฟลทติงพอยน์ (32 บิท)	
D11	D10	–	D1	D0	=	D11	D10
1000.000			320.560			679.440	



จำนวนจริงแบบโฟลทติงพอยน์ (32 บิท)			จำนวนจริงแบบโฟลทติงพอยน์ (32 บิท)			จำนวนจริงแบบโฟลทติงพอยน์ (32 บิท)	
D1	D0	–	D11	D10	=	D21	D20
2.540			10.550			-8.010	

4.2.2 คำสั่งคูณและหาร

รหัสคำสั่ง	ตัวอย่างแลดเดอร์
E* (การคูณ)	 จะเป็นการดำเนินแบบจำนวนจริง "S1 * S2 = D"
E/ (การหาร)	 จะเป็นการดำเนินแบบจำนวนจริง "S1 / S2 = D"

S1, S2 (ต้นทาง): สองรายการข้อมูลที่จะทำงาน
D (ปลายทาง): ปลายทางของข้อมูลหลังการทำงาน (หมายเลขอุปกรณ์)
P: คำสั่งที่จะดำเนินการบนขอบขึ้น

หมายเหตุ:
ในการทำงานแบบจำนวนจริง S1, S2 และ D ในแลดเดอร์จะต้องเป็นจำนวนจริงทั้งหมด
จำนวนเต็มและจำนวนจริงจะต้องไม่ปนกันในการทำงาน

4.2.2 คำสั่งคูณและหาร

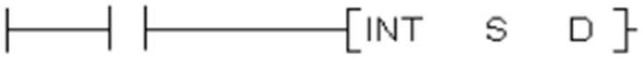
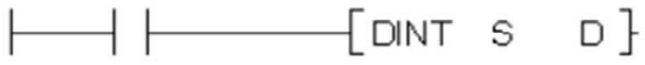
ตัวอย่างโปรแกรมที่มีคำสั่งคูณ

จำนวนจริงแบบโฟลทติงพอยน์ (32 บิท)			จำนวนจริงแบบโฟลทติงพอยน์ (32 บิท)			จำนวนจริงแบบโฟลทติงพอยน์ (32 บิท)	
D1	D0	×	D11	D10	=	D21	D20
1000.000			25.590			25590.000	

ตัวอย่างโปรแกรมที่มีคำสั่งหาร

จำนวนจริงแบบโฟลทติงพอยน์ (32 บิท)			จำนวนจริงแบบโฟลทติงพอยน์ (32 บิท)			จำนวนจริงแบบโฟลทติงพอยน์ (32 บิท)	
D1	D0	÷	D11	D10	=	D21	D20
1000.000			25.590			39.078	

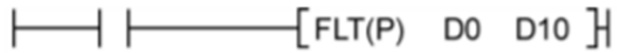
4.3 คำสั่งการแปลงระหว่างจำนวนเต็มและจำนวนจริง

รหัสคำสั่ง	ตัวอย่างแลดเดอร์	
FLT (การแปลงจำนวนเต็มเป็นจำนวนจริง)	จำนวนเต็ม (16 บิท) จะถูกแปลงเป็นจำนวนจริง (32 บิท) 	จำนวนเต็ม (32 บิท) จะถูกแปลงเป็นจำนวนจริง (32 บิท)
INT (การแปลงจำนวนเต็มเป็นจำนวนจริง)	จำนวนจริง (32 บิท) จะถูกแปลงเป็นจำนวนเต็ม (16 บิท) 	จำนวนจริง (32 บิท) จะถูกแปลงเป็นจำนวนเต็ม (32 บิท) 

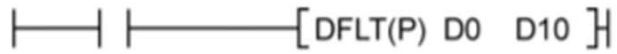
S (ต้นทาง): ข้อมูลก่อนการทำงาน (ค่าคงที่ หมายเลขอุปกรณ์)
D (ปลายทาง): ปลายทางของขอมูลหลังการทำงาน (หมายเลขอุปกรณ์)

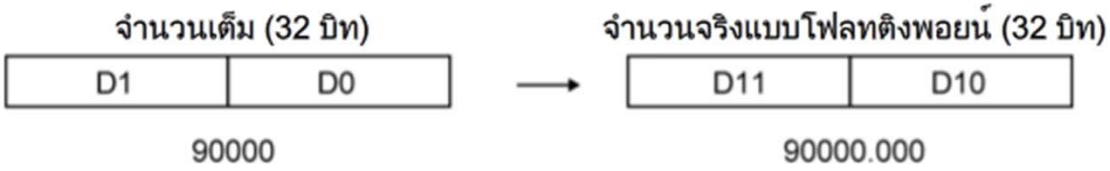
4.3 คำสั่งการแปลงระหว่างจำนวนเต็มและจำนวนจริง

ตัวอย่างโปรแกรมที่มีคำสั่งแปลงจำนวนเต็ม (16 บิท) / จำนวนจริง (32 บิท)



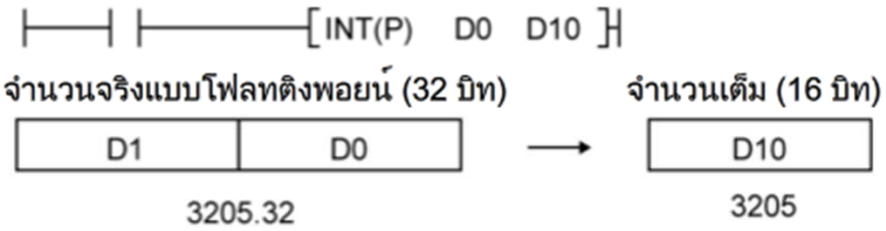
ตัวอย่างโปรแกรมที่มีคำสั่งแปลงจำนวนเต็ม (32 บิท) / จำนวนจริง (32 บิท)



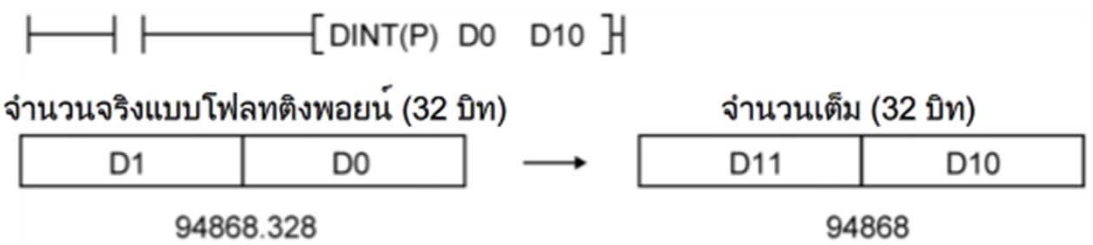


4.3 คำสั่งการแปลงระหว่างจำนวนเต็มและจำนวนจริง

ตัวอย่างโปรแกรมที่มีคำสั่งการแปลงจำนวนจริง (32 บิท) / จำนวนเต็ม (16 บิท)



ตัวอย่างโปรแกรมที่มีคำสั่งการแปลงจำนวนจริง (32 บิท) / จำนวนเต็ม (32 บิท)



4.4 สรุป

ในบทนี้ คุณได้เรียนรู้เกี่ยวกับ:

- การใช้งานและสัญลักษณ์ของจำนวนจริง
- คำสั่งการทำงานของจำนวนจริง
- คำสั่งการแปลงระหว่างจำนวนเต็มและจำนวนจริง

จุด

การทำงานของจำนวนจริง	•ข้อมูลจำนวนจริงจะใช้หน่วยความจำ 2 เวิร์ด (32 บิท) •เพิ่ม E ด้านหน้าของคำสั่งการทำงานของจำนวนจริง เช่น E* (การคูณ) •จะไม่สามารถประมวลผลจำนวนเต็มและจำนวนจริงพร้อมกันได้ จะต้องทำการแปลงจำนวนเต็มเป็นจำนวนจริง ก่อนที่จะประมวลผลการทำงานทางคณิตศาสตร์

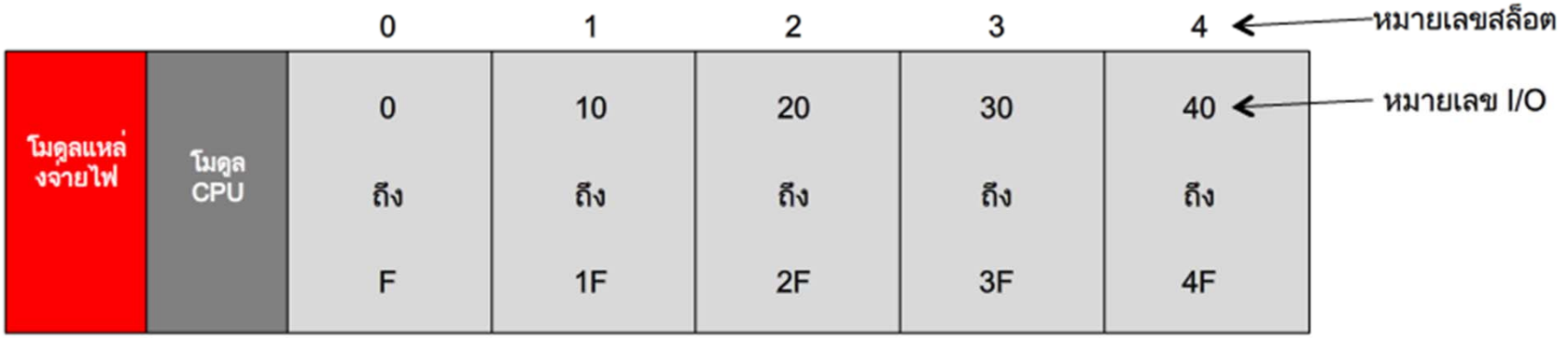
บทที่ 5 แนวคิดของหมายเลข I/O และวิธีใช้ฟังก์ชันการกำหนด I/O

บทนี้จะอธิบายเกี่ยวกับแนวคิดของหมายเลข I/O และวิธีใช้ฟังก์ชันการกำหนด I/O

ส่วนที่ 5.1: แนวคิดของหมายเลข I/O
ส่วนที่ 5.2: หมายเลข I/O สำหรับextension base
ส่วนที่ 5.3: การตรวจสอบการมอบหมายหมายเลข I/O บนการตรวจสอบระบบ
ส่วนที่ 5.4: วิธีใช้ฟังก์ชันการกำหนด I/O
ส่วนที่ 5.5: สรุป

5.1 แนวคิดของหมายเลข I/O

หมายเลข I/O จะถูกกำหนดไปยังโมดูล I/O บนbase ตามที่แสดงด้านล่าง
(I/O โมดูลจะมีอยู่สามประเภท: ประเภท 16, 32 และ 64 จุด ตัวอย่างที่แสดงด้านล่างจะใช้ I/O โมดูลประเภท 16 จุด)



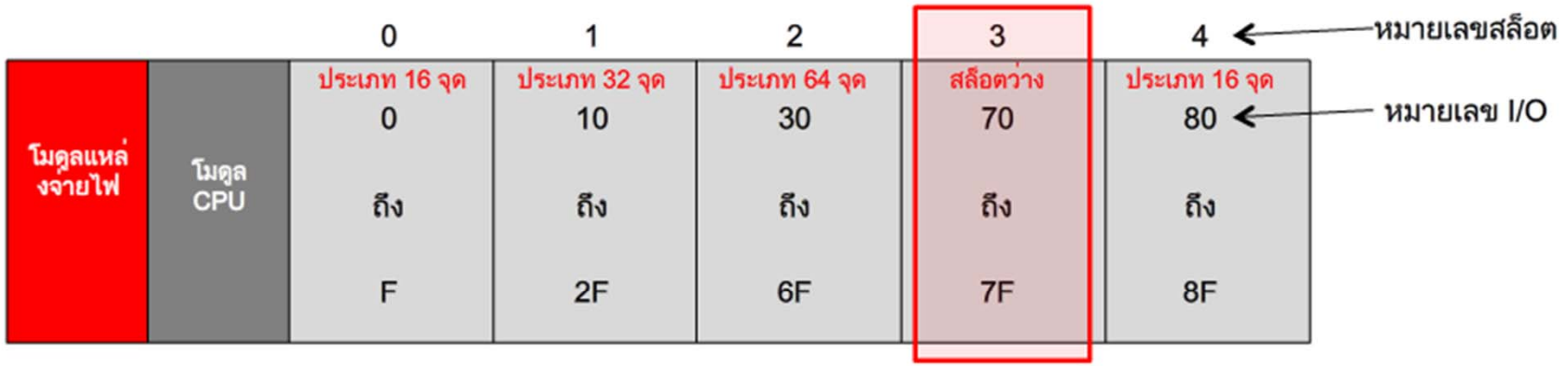
(ตัวอย่าง) base Q35B ที่มีห้า I/O สล็อต

หมายเลข I/O (ค่าฐานสิบหก 0 ถึง F) จะถูกมอบหมายให้กับแต่ละสล็อต (โมดูล) ตามลำดับ โดยเริ่มจากสล็อตที่ใกล้กับโมดูล CPU ที่สุด แ
ตละสล็อต (โมดูล) จะถูกมอบหมายตามค่าเริ่มต้นด้วย 16 หมายเลข I/O

5.1 แนวคิดของหมายเลข I/O

เมื่อใช้งาน I/O โมดูล 16, 32 และ 64 จุดร่วมกัน หมายเลข I/O จะถูกกำหนดดังนี้:

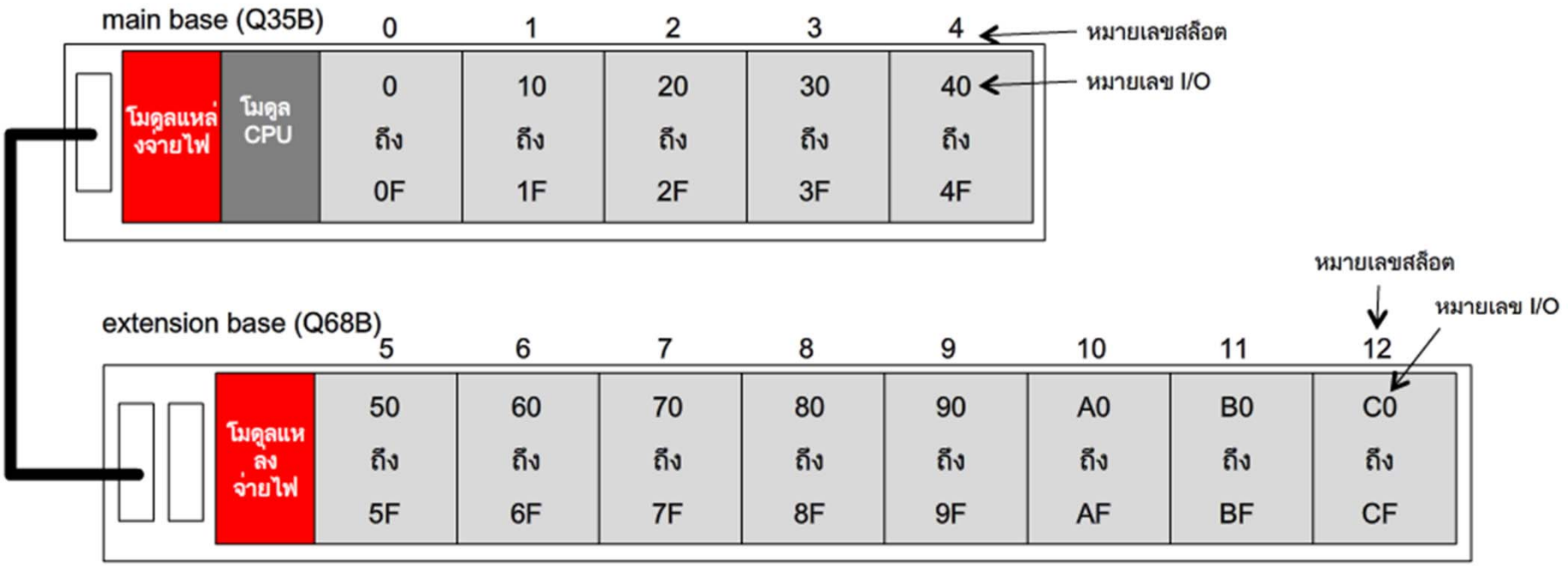
หากมีสล็อตว่างตรงกลางbase หมายเลข I/O จะถูกกำหนดลงในสล็อตด้วย (ในการตั้งค่าเบื้องต้น)



หมายเหตุ: 16 หมายเลข I/O (ค่าฐานสิบหก) จะถูกมอบหมายให้กับสล็อตว่างตามค่าเริ่มต้น อย่างไรก็ตาม การตั้งค่านี้สามารถเปลี่ยนแปลงได้ และสามารถตั้งคาหมายเลข I/O ในชวง 0 ถึง 64 ไดในยูนิต 16 จุด

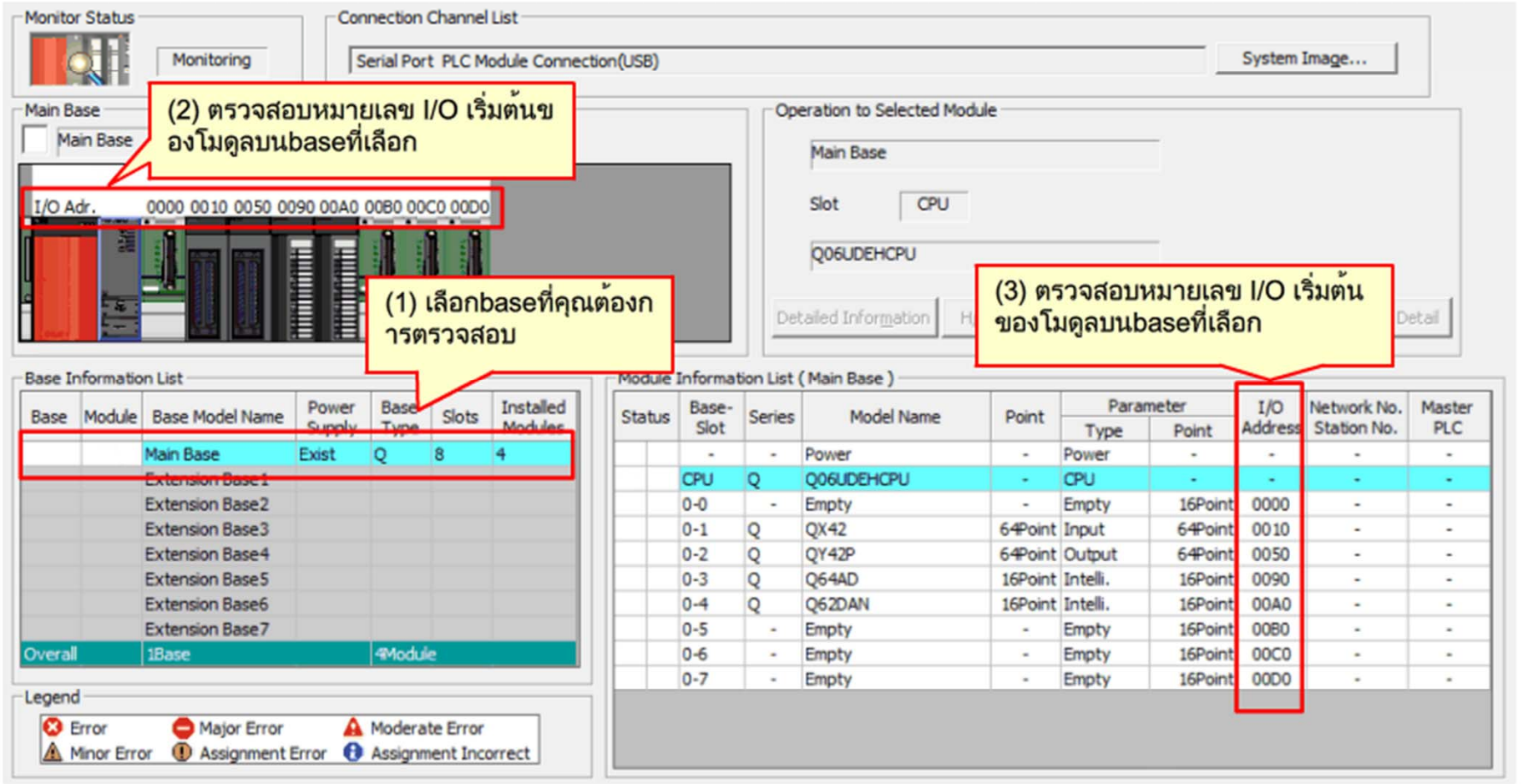
5.2 หมายเลข I/O สำหรับextension base

หมายเลข I/O ของแต่ละโมดูล ซึ่งสอดคล้องกับรีเลย์อินพุท (X)/เอาท์พุท (Y) ของโมดูล CPU จะถูกมอบหมายโดยอัตโนมัติ ด้วยการตรวจหาโมดูลบนbase
หมายเลข I/O ของโมดูลบนextension baseจะถูกมอบหมายโดยอัตโนมัติเช่นกัน ถัดจากหมายเลข I/O สุดท้ายของmain base
ภาพต่อไปนี้จะแสดงวิธีการมอบหมายหมายเลข I/O โดยใช้โมดูล 16 จุด



5.3 การตรวจสอบการมอบหมายหมายเลข I/O บนการตรวจสอบระบบ

ในการตรวจสอบการมอบหมายหมายเลข I/O ให้ไปที่เมนูของ GX Works2 เลือก Diagnostics (การวินิจฉัย) จากนั้นเลือก System Monitor (การตรวจสอบระบบ)



5.4 วิธีใช้ฟังก์ชันการกำหนด I/O

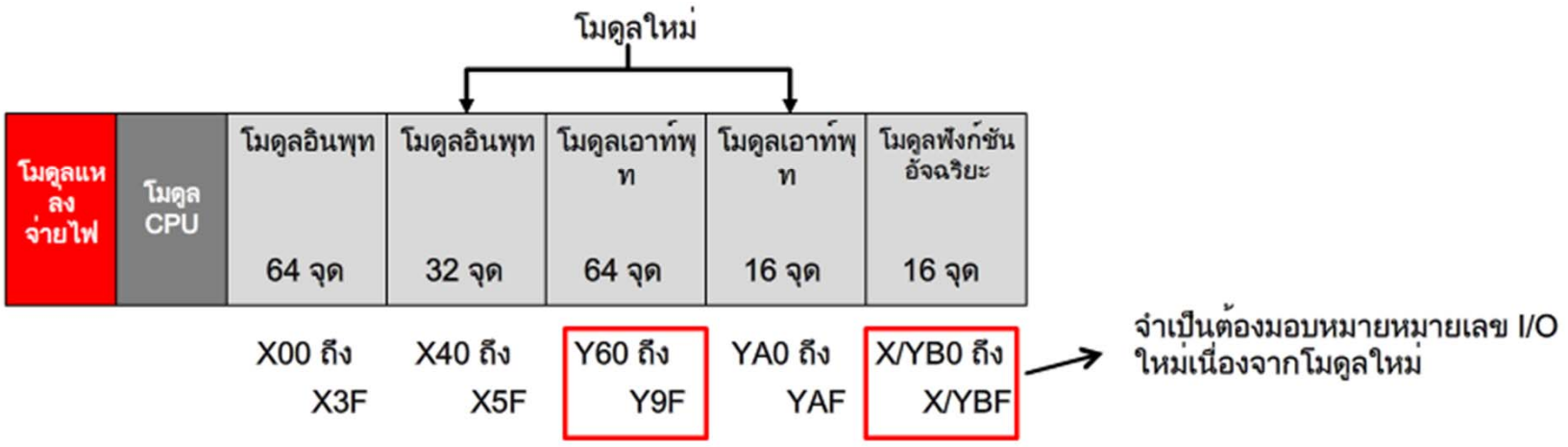
ฟังก์ชันการกำหนด I/O จะมอบหมายหมายเลข I/O แบบตายตัวไปยังสล็อตbaseแทนโมดูลที่ติดตั้งไว้ ซึ่งหมายความว่าจะไม่จำเป็นต้องมอบหมายหมายเลข I/O สำหรับโมดูลที่มีอยู่ซ้ำอีก แม้ว่าการตั้งค่าระบบจะเปลี่ยนแปลงไป (เช่น เมื่อมีการเพิ่มโมดูลใหม่)

(1) ไม่มีฟังก์ชันการกำหนด I/O

การตั้งค่าระบบโดยไม่มีโมดูลใหม่

โมดูลแหล่งจ่ายไฟ	โมดูล CPU	โมดูลอินพุท	โมดูลเอาท์พุท	โมดูลฟังก์ชันอัจฉริยะ	
		64 จุด	64 จุด	16 จุด	
		X00 ถึง X3F	Y40 ถึง Y7F	X/Y80 ถึง X/Y8F	

การตั้งค่าระบบโดยมีโมดูลใหม่ (เพิ่มอินพุทโมดูล 32 จุด และเอาท์พุทโมดูล 16 จุด)

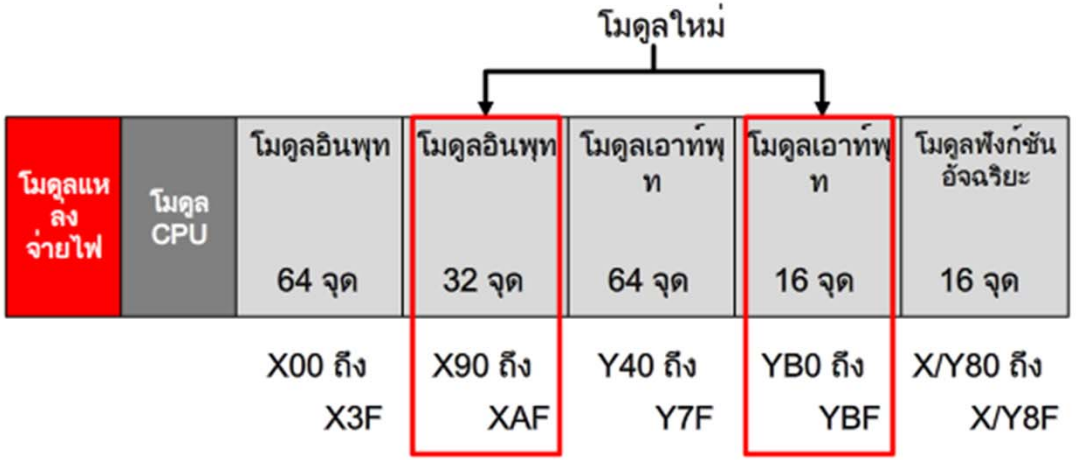


5.4 วิธีใช้ฟังก์ชันการกำหนด I/O

(2) มีฟังก์ชันการกำหนด I/O

การตั้งค่าระบบโดยไม่มีโมดูลใหม่

การตั้งค่าระบบโดยมีโมดูลใหม่ (เพิ่มอินพุทโมดูล 32 จุด และเอาท์พุทโมดูล 16 จุด)



เนื่องจากหมายเลข I/O ของโมดูลที่มีอยู่จะยังคงไม่เปลี่ยนแปลง จะต้องแก้ไขเฉพาะโปรแกรมที่เพิ่มโมดูลเท่านั้น

5.4 วิธีใช้ฟังก์ชันการกำหนด I/O

สามารถกำหนดการตั้งค่าการกำหนด I/O ได้จาก GX Works2 เปิดหน้าต่าง PLC Parameter (PLC พารามิเตอร์) จากนั้นเลือกแท็บ I/O Assignment (การกำหนด I/O)
หมายเลข I/O ใดๆ อาจถูกมอบหมายให้กับแต่ละสล็อตได้ทั้งนั้น ไม่ว่าลำดับทางกายภาพของสล็อตจะเป็นอย่างไร

ตั้งค่าหมายเลข I/O เริ่มต้นของโมดูล

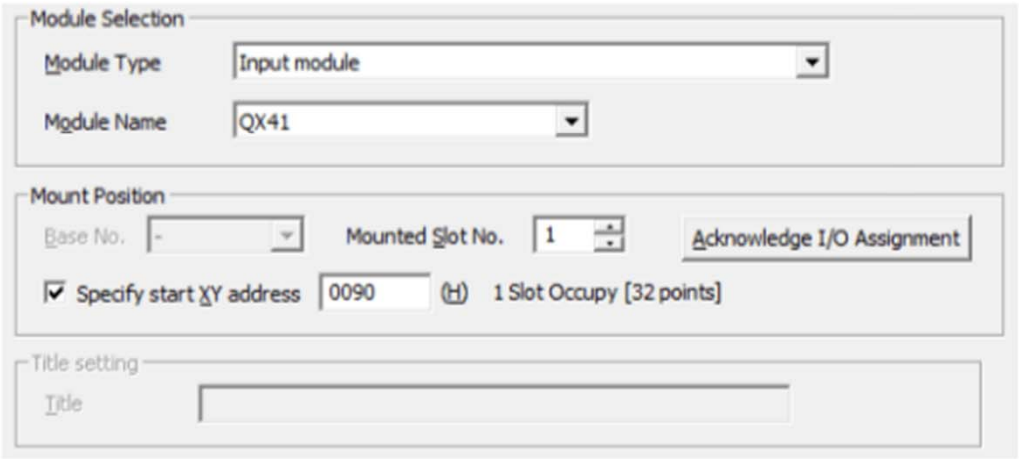
I/O Assignment(*1)

No.	Slot	Type	Model Name	Points	Start XY
0	PLC	PLC			
1	0(*-0)	Input	QX42	64Points	0000
2	1(*-1)	Input	QX41	32Points	0090
3	2(*-2)	Output	QY42	64Points	0040
4	3(*-3)	Output	QY50	16Points	00B0
5	4(*-4)	Intelligent	Q62DA	16Points	0080
6	5(*-5)				
7	6(*-6)				

Switch Setting | Detailed Setting | Select PLC type | New Module

Assigning the I/O address is not necessary as the CPU does it automatically.
Leaving this setting blank will not cause an error to occur.

หมายเลข I/O ไม่มีหมายเลขต่อเนื่อง อาจสามารถข้ามบางหมายเลขได้
หากมีการคาดว่าระบบจะขยายขนาดในอนาคต อาจสามารถสำรองบางหมายเลขเอาไว้ได้



คลิก New Module (โมดูลใหม่) เพื่อเปิดหน้าต่างนี้
จากตรงนี้ คุณสามารถเลือกและลงทะเบียนประเภทโมดูลและชื่อโมดูลได้ โดยใช้รายการดรอปดาวน์

5.4.1 การตั้งค่าสล็อตbase

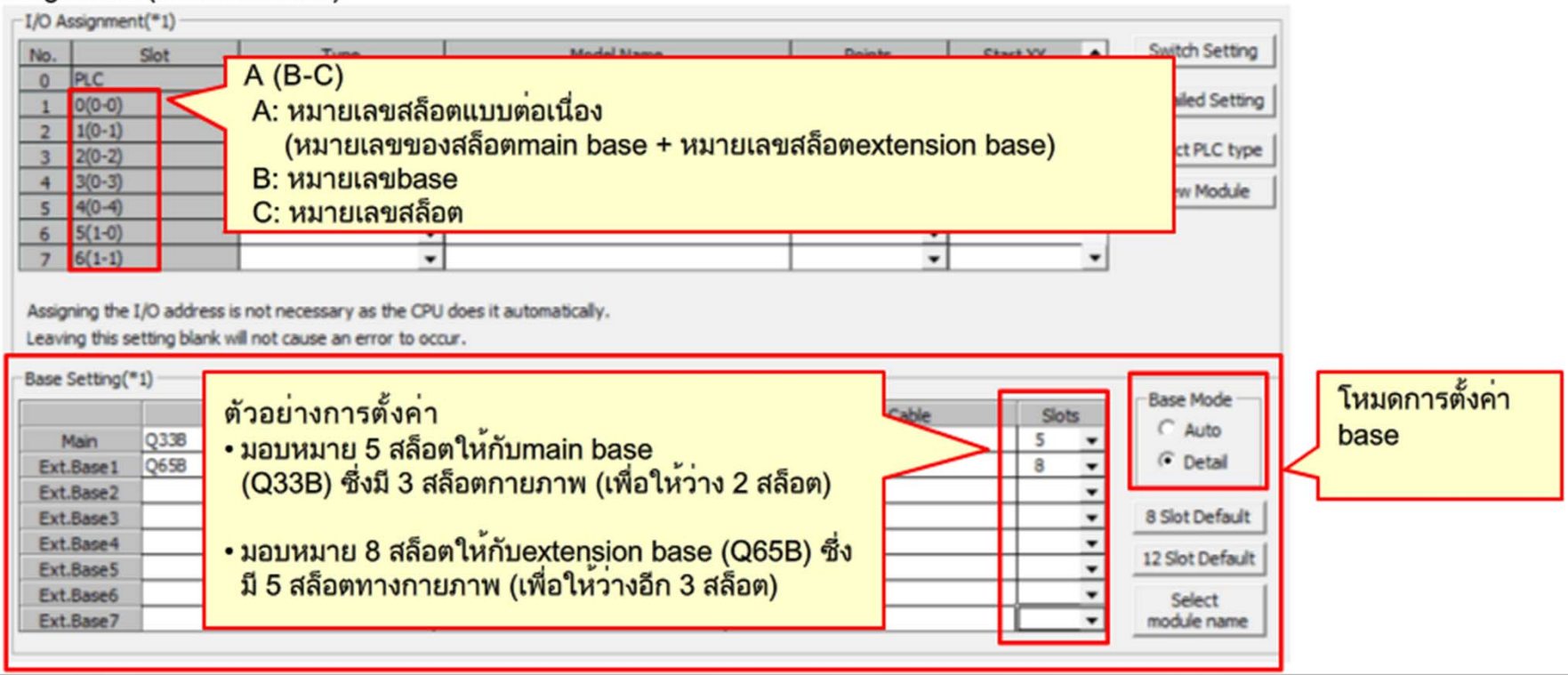
แต่ละสล็อตของbaseจะมีหมายเลขที่เรียกว่าหมายเลขสล็อต ซึ่งสามารถกำหนดได้จากภายในการตั้งค่าการกำหนด I/O หมายเลขสล็อตจะถูกมอบหมายโดยอัตโนมัติ (ในเกือบทุกกรณี) โดยสามารถตั้งค่าแบบแมนนวลได้ด้วยโหมดแบบละเอียด โหมดแบบละเอียดนี้จะมีประโยชน์ในการสำรองหมายเลขสล็อตบางส่วนไว้ เพื่อการขยายระบบในอนาคต

โหมดอัตโนมัติ (ค่าเริ่มต้น)

หมายเลขสล็อตจะถูกตั้งค่าโดยอัตโนมัติตามจำนวนสล็อตทางกายภาพของbase (main หรือ extension)
เมื่อเชื่อมต่อextension baseเข้ากับmain base หมายเลขสล็อตของextension base จะถูกกำหนดไว้ถัดจากหมายเลขสล็อตสุดท้ายของmain base
(ตัวอย่าง) เมื่อmain baseมีห้าสล็อต (หมายเลขสล็อต 0 ถึง 4) สล็อตที่เชื่อมต่อกับextension base จะถูกกำหนดหมายเลขโดยเริ่มต้นด้วย 5

โหมดแบบละเอียด

ตั้งค่าจำนวนสล็อตสำหรับแต่ละbase สามารถตั้งค่าหมายเลขใดๆ ก็ได้ เมื่อใช้โหมดแบบละเอียด จะต้องใช้การตั้งค่านี้สำหรับbaseทั้งหมดที่อยู่ระหว่างการใช้งาน ในการดำเนินการตั้งค่า ให้เปิดหน้าต่าง PLC Parameter (PLC พารามิเตอร์) และเลือกแท็บ I/O Assignment (การกำหนด I/O)



5.5 สรุป

ในบทนี้ คุณได้เรียนรู้เกี่ยวกับ:

- แนวคิดของหมายเลข I/O
- หมายเลข I/O สำหรับextension base
- การตรวจสอบการมอบหมายหมายเลข I/O บนการตรวจสอบระบบ
- วิธีใช้ฟังก์ชันการกำหนด I/O

จุด

แนวคิดของหมายเลข I/O และวิธีใช้ฟังก์ชันการกำหนด I/O	• หมายเลข I/O ของแต่ละอินพุท/เอาท์พุทโมดูล จะถูกกำหนดตามลำดับในหน่วย 16 จุด (0 ถึง F) จากสล็อตที่อยู่ใกล้กับโมดูล CPU ที่สุด • หากมีสล็อตว่างตรงกลางbase หมายเลข I/O จะถูกกำหนดลงในสล็อตว่างด้วย • หมายเลข I/O ของโมดูลบนextension baseจะถูกกำหนดโดยอัตโนมัติเช่นกัน ถัดจากหมายเลข I/O สุดท้ายของmain base • สำหรับฟังก์ชันการกำหนด I/O จะสามารถกำหนดหมายเลข I/O ได้ โดยไม่สนใจลำดับทางกายภาพของสล็อตบนbase

ทดสอบ แบบทดสอบประเมินผล

หลังจากที่คุณผ่านบทเรียนทั้งหมดของหลักสูตร PLC การใช้งานการตั้งโปรแกรม คุณก็จะพร้อมสำหรับการทำแบบทดสอบประเมินผล หากคุณยังไม่มั่นใจเกี่ยวกับหัวข้อต่างๆ ที่จะทดสอบ โปรดทบทวนหัวข้อเหล่านั้น
คำถามในแบบทดสอบประเมินผลนี้มีทั้งหมด 6 ข้อ (29 รายการ)
คุณสามารถทำแบบทดสอบประเมินผลได้หลายครั้งตามต้องการ

วิธีการตอบคำถามในแบบทดสอบ

หลังจากเลือกคำตอบแล้ว ให้คลิกปุ่ม ตอบ คำตอบของคุณจะหายไปถ้าคุณดำเนินการต่อโดยไม่คลิกปุ่ม ตอบ (จะถือว่าคุณยังไม่ได้ตอบคำถามนั้น)

ผลคะแนน

จำนวนคำตอบที่ถูกต้อง จำนวนคำถาม เปอร์เซ็นต์คำตอบที่ถูกต้อง และผลลัพธ์ที่แสดงว่าผ่าน/ไม่ผ่าน จะปรากฏบนหน้าผลคะแนน

คำตอบที่ถูกต้อง: 2

จำนวนคำถามทั้งหมด: 9

เปอร์เซ็นต์: 22%

คุณต้องตอบคำถามถูกต้องเกินกว่า 60% จึงจะผ่านการทดสอบ

ดำเนินการต่อ | ทบทวน | ลองใหม่

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

ทดสอบ แบบทดสอบประเมินผล 1

ประโยคต่อไปนี้เป็นคำอธิบายของRetentive Timer เลือกคำที่เมื่อเติมลงในช่องว่าง จะทำให้ประโยคมีความหมายสมบูรณ์

เมื่อเป็นไปตามเงื่อนไขของ --Select-- ขดลวดจะ --Select-- และRetentive Timerจะเริ่มทำการจับเวลา

ค่าของRetentive Timerจะถูกเก็บไว้ แม้ว่าเงื่อนไขจะเปลี่ยนเป็น --Select-- เงื่อนไขอินพุทในระหว่างการวัดค่า

เมื่อขดลวด --Select-- อีกครั้ง Timerจะเริ่มทำการจับเวลาอีกครั้งต่อจากค่าที่เก็บไว้

เมื่อวัดได้ถึงตามค่าที่ตั้งไว้ การหมดเวลาจะทำงานและ --Select-- จะเปิด

แม้ว่าขดลวดจะถูกปิดหลังจากหมดเวลา แต่ค่าที่วัดได้จะไม่ถูกล้าง และหน้าสัมผัสจะยังคงเปิดอยู่

จะมีการใช้คำสั่ง --Select-- เพื่อล้างค่าที่วัด และปิดหน้าสัมผัส

คำตอบ กลับ

ทดสอบ แบบทดสอบประเมินผล 2

จงทำให้โปรแกรมซีเควนท์ซึ่งใช้ทำงานต่อไปนี้ร่วมกับRetentive Timerนี้สมบูรณ์:

รายละเอียดการทำงาน:

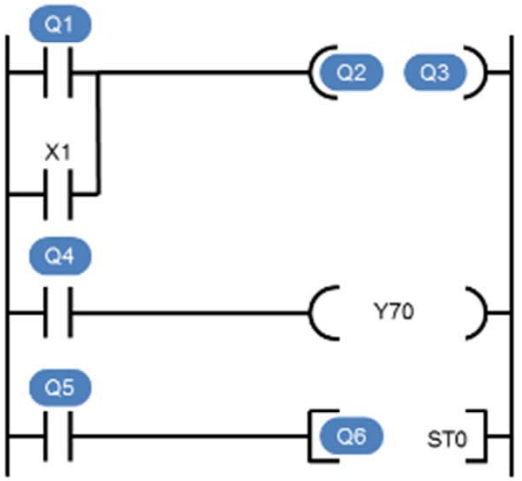
1) Retentive Timer (ST0) จะจับเวลาที่สัญญาณอินพุท X0 หรือ X1 เปิดอยู่

2) เมื่อระยะเวลาการเปิดของ X0 หรือ X1 ถึง 30 วินาที ขดลวด (Y70) จะเปิด ON เพื่อเปิดไฟหมดเวลา

3) เมื่อ X2 เปิด หน้าสัมผัสของRetentive Timer (ST0) จะถูกปิด และค่าที่วัดได้ (ค่าปัจจุบัน) จะถูกรีเซ็ต

Q1 --Select-- Q2 --Select-- Q3 --Select-- Q4 --Select--

Q5 --Select-- Q6 --Select--



คำตอบ กลับ

ทดสอบ แบบทดสอบประเมินผล 3

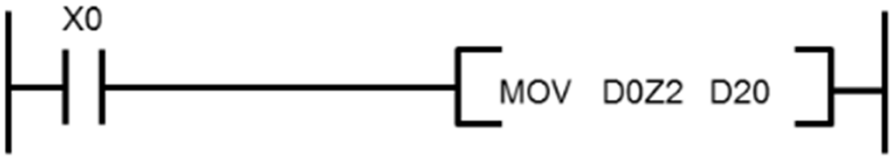
ด้านล่างคือโปรแกรมที่ใช้Index Register “Z2” เลือกค่าที่จะจัดเก็บในData Register (D20) เมื่อ X0 ถูกเปิด ภายใต้แต่ละเงื่อนไข:

1) เมื่อค่าที่จัดเก็บไว้ใน Z2 คือ 0, --Select-- จะถูกจัดเก็บไว้ใน D20

2) เมื่อค่าที่จัดเก็บไว้ใน Z2 คือ 1, --Select-- จะถูกจัดเก็บไว้ใน D20

3) เมื่อค่าที่จัดเก็บไว้ใน Z2 คือ 2, --Select-- จะถูกจัดเก็บไว้ใน D20

4) เมื่อค่าที่จัดเก็บไว้ใน Z2 คือ 3, --Select-- จะถูกจัดเก็บไว้ใน D20



ค่าทีจัดเก็บไว้ใน Data Register

D0	100
D1	200
D2	400
D3	500

คำตอบ กลับ

ทดสอบ แบบทดสอบประเมินผล 4

ประโยคต่อไปนี้จะอธิบายถึงFile Registerใน QCPU เลือกคำที่เมื่อเติมลงในช่องว่าง จะทำให้ประโยคมีความหมายสมบูรณ์

1) File Registerคือเวิร์ดดีไวส์ที่ใช้ในการขยายส่วนขยายData Register (D) และจะถูกนำเสนอโดยสัญลักษณ์อุปกรณ์

--Select--

2) ส่วนนี้จะไม่เหมือนกับData Register คือ ข้อมูลที่จัดเก็บไว้ในFile Registerจะไม่ใช่ --Select-- แม้ในขณะที่ปิดเครื่องหรือรีเซ็ต CPU โมดูล

3) โดยปกติแล้ว File Registerจะถูกจัดเก็บไว้ในการ์ดหน่วยความจำ (RAM) หรือใน --Select-- ภายใน CPU โมดูล

4) หากต้องการใช้File Register คุณจะต้องทำการตั้งค่าที่จำเป็นภายในแท็บ --Select-- ของหน้าต่าง PLC พารามิเตอร์

คำตอบ กลับ

ทดสอบ แบบทดสอบประเมินผล 5

ในบรรดาค่าตัวเลขที่ใช้ในตัวควบคุมแบบตั้งโปรแกรมได้ ค่าตัวเลขที่ปราศจากค่าฐานสิบ จะถูกอ้างอิงถึงเป็นจำนวนเต็ม และค่าที่มีจุดทศนิยม จะถูกอ้างอิงถึงเป็นจำนวนจริง
เลือกค่าที่เมื่อเติมลงในช่องว่างแล้ว จะทำให้ข้อความอธิบายจำนวนจริงต่อไปนี้มีความหมายสมบูรณ์

1) จำนวนจริงหนึ่งค่าจะใช้ --Select-- เวิร์ดดีไวส์ และต้องการพื้นที่หน่วยความจำ --Select-- บิท

2) ค่าตัวเลขที่ใช้จัดการกับจำนวนจริง จะถูกอ้างอิงถึงเป็น --Select-- เช่น ค่าตัวเลข 2.035 จะถูกกล่าวถึงเป็น --Select-- ในโปรแกรมซีเควนท์

3) คำสั่งที่ใช้จัดการจำนวนจริง จะมี --Select-- นำหน้า

4) คำสั่งเลขคณิตที่จัดการจำนวนจริง --Select-- จะมีทั้งจำนวนเต็มและจำนวนจริงในเวลาเดียวกัน เพื่อใช้ในการทำงาน

คำตอบ กลับ

ทดสอบ แบบทดสอบประเมินผล 6

ทำให้โปรแกรมซีเควนท์ต่อไปนี้สมบูรณ์โดยใช้จำนวนจริง

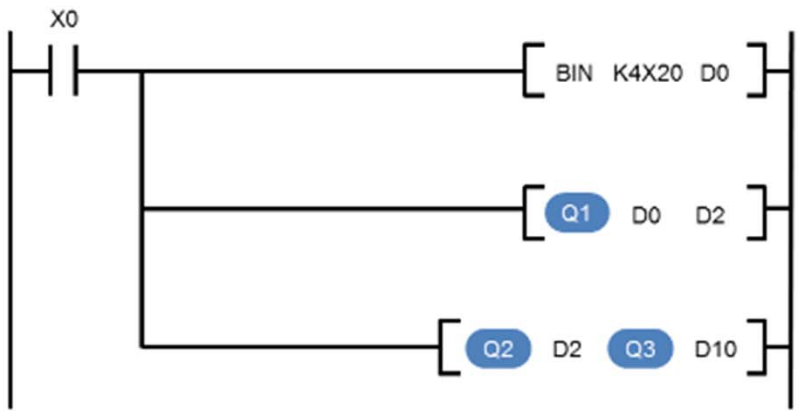
รายละเอียดโปรแกรม

1) เมื่อ X0 ถูกเปิด ข้อมูลการทำงานใน X20 ถึง X2F (ข้อมูล BCD) จะถูกอ่านและจัดเก็บไว้ใน D0

2) ค่าใน D0 จะถูกแปลงเป็นจำนวนจริง และถูกจัดเก็บไว้ใน D2

3) ค่าใน D2 จะคูณด้วย 3.14 และผลิตภัณฑ์จะถูกจัดเก็บไว้ใน D10

Q1 --Select-- Q2 --Select-- Q3 --Select--



คำตอบ กลับ

ทดสอบ คะแนนทดสอบ

คุณทำบททดสอบประเมินผลเสร็จแล้ว ผลลัพธ์ของคุณแบ่งส่วนได้ดังนี้
ในการสิ้นสุดแบบทดสอบประเมินผล ให้ไปต่อยังหน้าถัดไป

คำตอบที่ถูกต้อง : 6

จำนวนคำถามทั้งหมด : 6

เปอร์เซ็นต์ : 100%

ดำเนินการต่อ ทบทวน

ขอแสดงความยินดี คุณผ่านการทดสอบแล้ว

คุณผ่านหลักสูตร PLC การใช้งานการตั้งโปรแกรม

ขอขอบคุณสำหรับการเรียนรู้หลักสูตรนี้

เราหวังว่าคุณจะเพลิดเพลินกับบทเรียน และข้อมูลที่คุณได้รับจากหลักสูตรนี้
จะเป็นประโยชน์ในอนาคต

คุณสามารถทบทวนหลักสูตรได้หลายครั้งตามต้องการ

ทบทวน ปิด