Smarter. Greener. Together.

Industrial Automation Headquarters

Taiwan: Delta Electronics, Inc.
Taoyuan Technology Center
No.18, Xinglong Rd., Taoyuan District,
Taoyuan City 33068, Taiwan
TEL: +886-3-362-6301 / FAX: +886-3-371-6301

Asia

China: Delta Electronics (Shanghai) Co., Ltd.
No.182 Minyu Rd., Pudong Shanghai, P.R.C.
Post code : 201209
TEL: +86-21-6872-3988 / FAX: +86-21-6872-3996
Customer Service: 400-820-9595

Japan: Delta Electronics (Japan), Inc.
Industrial Automation Sales Department
2-1-14 Shibadaimon, Minato-ku
Tokyo, Japan 105-0012
TEL: +81-3-5733-1155 / FAX: +81-3-5733-1255

Korea: Delta Electronics (Korea), Inc.
1511, 219, Gasan Digital 1-Ro., Geumcheon-gu,
Seoul, 08501 South Korea
TEL: +82-2-515-5305 / FAX: +82-2-515-5302

Singapore: Delta Energy Systems (Singapore) Pte Ltd.
4 Kaki Bukit Avenue 1, #05-04, Singapore 417939
TEL: +65-6747-5155 / FAX: +65-6744-9228

India: Delta Electronics (India) Pvt. Ltd.
Plot No.43, Sector 35, HSIIDC Gurgaon,
PIN 122001, Haryana, India
TEL: +91-124-4874900 / FAX: +91-124-4874945

Thailand: Delta Electronics (Thailand) PCL.
909 Soi 9, Moo 4, Bangpoo Industrial Estate (E.P.Z),
Pattana 1 Rd., T.Phraksa, A.Muang,
Samutprakarn 10280, Thailand
TEL: +66-2709-2800 / FAX: +66-2709-2827

Australia: Delta Electronics (Australia) Pty Ltd.
Unit 20-21/45 Normanby Rd., Notting Hill Vic 3168, Australia
TEL: +61-3-9543-3720

Americas

USA: Delta Electronics (Americas) Ltd.
5101 Davis Drive, Research Triangle Park, NC 27709, U.S.A.
TEL: +1-919-767-3813 / FAX: +1-919-767-3969

Brazil: Delta Electronics Brazil
Rua Itapeva, 26 - 3°, andar Edificio Itapeva,
One - Bela Vista 01332-000 - São Paulo - SP - Brazil
TEL: +55-12-3932-2300 / FAX: +55-12-3932-237

Mexico: Delta Electronics International Mexico S.A. de C.V.
Gustavo Baz No. 309 Edificio E PB 103
Colonia La Loma, CP 54060
Tlalnepantla, Estado de México
TEL: +52-55-3603-9200

EMEA

EMEA Headquarters: Delta Electronics (Netherlands) B.V.
Sales: Sales.IA.EMEA@deltaww.com
Marketing: Marketing.IA.EMEA@deltaww.com
Technical Support: iatechnicalsupport@deltaww.com
Customer Support: Customer-Support@deltaww.com
Service: Service.IA.emea@deltaww.com
TEL: +31(0)40 800 3900

BENELUX: Delta Electronics (Netherlands) B.V.
Automotive Campus 260, 5708 JZ Helmond, The Netherlands
Mail: Sales.IA.Benelux@deltaww.com
TEL: +31(0)40 800 3900

DACH: Delta Electronics (Netherlands) B.V.
Coesterweg 45, D-59494 Soest, Germany
Mail: Sales.IA.DACH@deltaww.com
TEL: +49(0)2921 987 0

France: Delta Electronics (France) S.A.
ZI du bois Challand 2, 15 rue des Pyrénées,
Lisses, 91090 Evry Cedex, France
Mail: Sales.IA.FR@deltaww.com
TEL: +33(0)1 69 77 82 60

Iberia: Delta Electronics Solutions (Spain) S.L.U
Ctra. De Villaverde a Vallecas, 265 1º Dcha Ed.
Hormigueras – P.I. de Vallecas 28031 Madrid
TEL: +34(0)91 223 74 20

Carrer Llacuna 166, 08018 Barcelona, Spain
Mail: Sales.IA.Iberia@deltaww.com

Italy: Delta Electronics (Italy) S.r.l.
Via Meda 2–22060 Novedrate(CO)
Piazza Grazioli 18 00186 Roma Italy
Mail: Sales.IA.Italy@deltaww.com
TEL: +39 039 8900365

Russia: Delta Energy System LLC
Vereyskaya Plaza II, office 112 Vereyskaya str.
17 121357 Moscow Russia
Mail: Sales.IA.RU@deltaww.com
TEL: +7 495 644 3240

Turkey: Delta Greentech Elektronik San. Ltd. Sti. (Turkey)
Şerifali Mah. Hendem Cad. Kule Sok. No:16-A
34775 Ümraniye – İstanbul
Mail: Sales.IA.Turkey@deltaww.com
TEL: + 90 216 499 9910

MEA: Eltek Dubai (Eltek MEA DMCC)
OFFICE 2504, 25th Floor, Saba Tower 1,
Jumeirah Lakes Towers, Dubai, UAE
Mail: Sales.IA.MEA@deltaww.com
TEL: +971(0)4 2690148

*We reserve the right to change the information in this manual without prior notice.

DELTA_IA-CNC_NC Series MLC_AM_EN_20210630

Delta CNC Solution MLC Application Manual

Delta CNC Solution MLC Application Manual

www.deltaww.com

Smarter. Greener. Together.

Preface

Thank you for choosing this product. Before using the product, please read through this manual carefully in order to ensure the correct use of the product. In addition, please place the manual safely for quick reference whenever is needed.

This manual includes:

- List of MLC devices
- MLC basic instructions
- Introduction to MLC application instructions
- List of MLC application instructions
- MLC Special M, D commands and functions
- MLC application examples

Features of NC series controllers

- Built-in 32-bit highspeed dual CPU for multi-task execution and performance improvement
- Friendly HMI Interface
- Auto tuning interface are provided for optimizing the machine's performance efficiency
- CNC Soft software tools to facilitate the development of customized images
- Front USB interface to facilitate data access, data backup, and parameters copy
- Spindle forms for users to choose between communication type and analog voltage type
- Serial I/O modules for flexible I/O contacts configuration

How to use this manual:

This manual can be used as reference while learning NC controllers. It lists instructions, special M and D commands, as well as instructs how to edit MLC with application examples. Before using and setting this product, please read through this manual carefully.

DELTA technical services

Please consult the distributors or DELTA customer service center if any problem occurs.

Safety Precautions

- Please follow the instruction of pin definition when wiring. Ground is a must.
- When the power is being supplied, do not disconnect the controller, change the wiring, or touch the power supply.

Please pay close attention to the following safety precautions during inspecting, installation, operating, maintenance and troubleshooting.

The symbols of “**DANGER**”, “**WARNING**” and “**STOP**” represent:

DANGER	**It indicates the potential hazards. It is possible to cause severe injury or fatal harm if not follow the instructions.**
WARNING	**It indicates the potential hazards. It is possible to cause minor injury or lead to serious damage of the product or even malfunction if not follow the instructions.**
STOP	**It indicates the absolute prohibited activity. It is possible to damage the product or cannot be used due to malfunction if not follow the instructions.**

Installation

DANGER

- Please follow the installation instructions in this manual; otherwise it may cause damage to the equipment.
- It is prohibited to expose the product to the environment containing water, corrosive gas, inflammable gas etc. Otherwise, electric shock or fire may occur.

Wiring

DANGER

- Please connect the ground terminal to class-3 ground system (under 100 Ω). Poor grounding may result in electric shock or fire.

Operation

WARNING

- Correctly plan out the I/O actions with MLC Editor Software, or abnormal results may occur.
- Before operation, please properly adjust the parameter settings of the machine, otherwise it may cause abnormal operation.
- Please ensure the emergency stop can be activated at any time, and avoid operating the machine in unprotected condition.

STOP

- Do not modify wiring while power is being supplied. Otherwise, it may cause personal injury due to electric shock.
- Never use a sharp-pointed object to touch the panel, as doing this might dent the screen and lead to malfunction of the controller.

Maintenance and Inspection

STOP

- While power is being supplied, do not disassemble the controller panel or touch the internal parts, otherwise electric shock may occur.
- Do not touch the ground terminal within 10 minutes after turning off the power, as the residual voltage may cause electric shock.
- Turn OFF the power first before replacing backup battery, and recheck the system settings afterwards.
- Do not block the vent holes during operation, as malfunction may easily occur due to poor ventilation.

Wiring Method

WARNING

- Power supply: In order to avoid danger, use a 24 V_{DC} power supply for the controller and comply with the wire specification when wiring.
- Wiring materials: Use multi-stranded twisted-pair wires or multi-core shielded-pair wires to isolate all cables.
- The maximum cable length for remote I/O signals and DMCNET communication is 20 m and the maximum cable length for other signal cable is 10 m.
- To control the input and output signals, a 24 V_{DC} power is required for the controller I/O and remote I/O.

Wiring of Communication Circuit

WARNING

- DMCNET wiring: The wiring materials should be in compliance with the standard specification.
- Please make sure the wiring between the controller and servo drive is tight and secure, as loose cables may cause abnormal operation.

For the differences among the various versions, please refer to DELTA's website for the latest information (http://www.deltaww.com/industrialautomation/).

(This page is intentionally left blank.)

Table of Contents

1 MLC Devices

2 Basic MLC Instructions

3 MLC Application Instructions Overview

4 MLC Application Instructions Description

5 Instructions and Functions of MLC Special M Relay and Special D Register

6 MLC Application Examples

MLC Devices 1

This chapter describes the functions, quantity, and definitions of the MLC devices.

1.1 List of MLC devices

The NC series MLC contains a variety of devices listed as follows.

1.1.1 Quantity and setting ranges of MLC devices

Type	Device	Item		Quantity		Setting range
Relay (Bit)	X	External input relay		0 - 63 256 - 511	320 points in total	I/O
	Y	External output relay		0 - 63 256 - 511	320 points in total	I/O
	M	Auxiliary relay		0 - 3071	3,072 points in total	I/O
	A	Alarm		0 - 511	512 points in total	I/O
	T	Timer		0 - 255	256 points in total	I/O, Word
	C	Counter	16-bit	0 - 63	80 points in total	I/O, Word
			32-bit	64 - 77		
			32-bit, high-speed	78 - 79		
Register (Word)	T	Timer	16-bit	0 - 255	256 points in total	0 to 65535
	C	Counter	16-bit	0 - 63	80 points in total	0 to 65535
			32-bit	64 - 77		-2147,483,648 to +2147,483,647
			32-bit, high-speed	78 - 79		-2147,483,648 to +2147,483,647
	D	Data register	16-bit	0 - 1535	1,536 points in total	-32,768 to +32,767
	V	Index register	16-bit	0 - 7	8 points in total	-32,768 to +32,767
	Z	Index register	16-bit	0 - 7	8 points in total	-32,768 to +32,767
Indicator	N	Loop indicator		0 - 7	8 points in total	None
	P	Jump indicator		0 - 255	256 points in total	None
	I	Interrupt indicator (IX00 - IX07) (IC00 - IC01) (IR00 - IR31)		0 - 41	42 points in total	None
Constant	K	Decimal constant		N/A	N/A	N/A
Floating-point number	F	Floating-point number		N/A	N/A	N/A

1.1.2 Settings of MLC devices

Device name	Function and range					Total points
Input signal X (Bit)	On board	MPG	Undefined	Machine operation panel B	Remote	320
	X0 - X31	X32 - X39	X40 - X63	X64 - X255	X256 - X511	
Output signal Y (Bit)	On board		Undefined	Machine operation panel B	Remote	320
	Y0 - Y39		Y40 - Y63	Y64 - Y255	Y256 - Y511	
Auxiliary relay M (Bit)	General purpose	Non-volatile	System special M		System special M	3072
	M0 - M511	M512 - M1023	M1024 - M2335		M2816 - M3071	
Alarm A (Bit)	A0 - A511					512
Timer T Bit	T0 - T199 (Unit: 100 ms) T200 - T255 (Unit: 10 ms)					256
Timer T Word	T0 - T255 (16-bit, range: 0 - 65535)					
Counter C Bit	C0 - C79					80
Counter C Word & DWord	16-bit (counting up) C0 - C63		32-bit (counting up / down) C64 - C77 (When M2944 - M2957 are On, the counter counts down)		32-bit, high-speed C78 - C79	
	Counting range 0 - 65535		Counting range -2,147,483,648 to +2,147,483,647		Refer to DHSCS and DHSCR in CH04	
Data register D (Word)	General purpose	Non-volatile	System special D			1,536
	D0 - D511 (-32768 to +32767)	D512 - D1023	D1024 - D1535			
Register V (Word)	V0 - V7 (-32768 to +32767)					8
Register Z (Word)	Z0 - Z7 (-32768 to +32767)					8
Loop indicator N	For master control loop: N0 - N7					8
Jump indicator P	For CJ and CALL: P0 - P255					256
Interrupt indicator I	On-board interrupt		IX00 - IX07			42
	Counter interrupt		IC00 - IC01			
	Remote I/O interrupt		IR00 - IR31			
Constant K	Decimal constant		K-32,768 to K+32,767 (16-bit operation)			N/A
			K-2,147,483,648 to K+2,147,483,647 (32-bit operation)			N/A
Floating-point number F	Floating-point number with 3 decimal places		-99999.999 to +99999.999			N/A

1.2 Values and constants

The NC series MLC performs operations for different control purposes using the data types as follows. See the following descriptions for each data type.

1.2.1 Binary number (BIN)

The MLC performs operations and stores values with binary numbers. The binary numbers and related terms are as follows.

1. Bit: the basic unit of binary values, either 0 or 1.
2. Nibble: composed of four consecutive bits (such as bit0 to bit3), representing the values 0 to 15 in decimal or 0 to F in hexadecimal.
3. Byte: composed of two consecutive nibbles (8 bits, such as bit0 to bit7), representing 00 to FF in hexadecimal.
4. Word: composed of two consecutive bytes (16 bits, such as bit0 to bit15), representing the 4-digit values 0000 to FFFF in hexadecimal.
5. Double Word: composed of two consecutive words (32 bits, such as bit0 to bit31), representing the 8-digit values 00000000 to FFFFFFFF in hexadecimal.

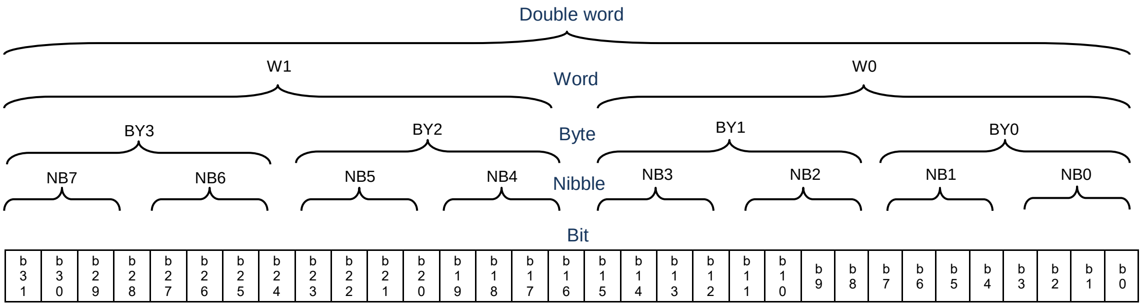


Figure 1.2.1.1 Relationship between nibble, byte, word, and double word in a binary system

1.2.2 Decimal number (DEC)

The MLC performs operations and stores values with binary numbers. On the other hand, it uses decimal numbers for the following items.

1. Device number:
   - Numbers for external input devices: X0 - X39, X64 - X511…
     Numbers for external output devices: Y0 - Y39, Y64 - Y511…
   - Numbers for devices of M, A, T, C, D, V, Z, K, P, I, and N, such as M10 and T30.
2. Constant K:
   - In the MLC, a decimal value is prefixed with a “K”. For example, K100 indicates the decimal value 100.
   - A constant K is used as a setting value for the timer (T) or counter (C). For example, TMR T0 K50.
   - In an application instruction, it is used as an operator. For example, MOV K123 D0.

   Note: a combination of K and a bit device (X, Y, M, or A) represents data in the format of nibble, byte, word, or double word. For example, K2Y10 and K4M100. In this case, K1 - K4 represent a 4-bit, 8-bit, 12-bit, and 16-bit data respectively.
3. Floating-point constant F:
   In the MLC, a floating-point value is prefixed with an “F”. In an application instruction, it is used as an operator. For example, FADD F12.3 F0 D0.

1.3 Numbers and functions of external input / output contacts (X, Y)

1.3.1 Numbers of input / output contacts

In the MLC, the numbers of the inputs and outputs start from X0 and Y0 respectively, including on-board I/O, I/O on the machine operation panel B, and remote I/O.

Device	On-board I/O	I/O on panel B	Expansion I/O (Remote I/O)							
			Station 1	Station 2	Station 3	Station 4	Station 5	Station 6	Station 7	Station 8
Input X	X0 - X39	X64 - X255	X256 - X287	X288 - X319	X320 - X351	X352 - X383	X384 - X415	X416 - X447	X448 - X479	X480 - X511
Output Y	Y0 - Y39	Y64 - Y255	Y256 - Y287	Y288 - Y319	Y320 - Y351	Y352 - Y383	Y384 - Y415	Y416 - Y447	Y448 - Y479	Y480 - Y511

Note: the starting numbers of input / output points on the expansion I/O correspond to the connecting station number. There are 8 stations in total with up to 256 input and output points respectively.

1.3.2 Functions of input / output relays

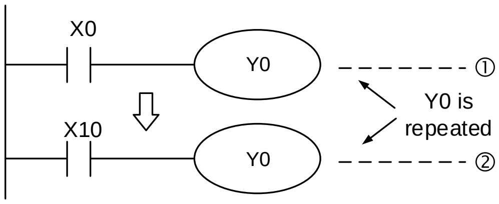
The input / output relays can enable and disable the MLC actions. The following describes the relay functions and state changes.

The functions of the input / output contacts:

1. Input contact X: connects to the input device and reads the input signal to the MLC. Contact A or B of each input contact X can be used for unlimited times in the program. The input contact X becomes On when the input device becomes On, and vice versa.
2. Output contact Y: outputs the On or Off signal to drive the load connecting to the contact. The output contact can be divided into relays and transistors. Contact A or B can be used for unlimited times in the program.

Pay attention to the following when using the output contacts.

It is suggested to use a unique number for the output coil in the program, or the output state is determined by the last circuit which outputs Y according to the MLC program scan principle. See the following figure for illustration.



The final output of Y0 is determined by circuit ②. That is, if X10 is On, Y0 is on; if X10 is Off, Y0 is off.

1.4 Numbers and functions of the auxiliary relay (M)

The auxiliary relay makes MLC programming easily. It starts from M0, including general purpose relays, non-volatile relays, system special relays, and MLC special relays. See the following for details.

Auxiliary relay (M)		
General purpose	M0 - M511, 512 points. Volatile.	3,072 points in total (including reserved auxiliary relays)
Non-volatile	M512 - M1023, 512 points. Non-volatile.	
System special	M1024 - M2335, 1,312 points. Volatile.	
MLC special	M2816 - M3071, 256 points. Volatile.	

1.4.1 Functions of the auxiliary relay

Same as the output relay Y, the auxiliary relay M has output coil and A & B contacts, which can be used for unlimited times in the program. You can use the auxiliary relay M to combine the control loop but cannot directly drive the external load. There are three types of auxiliary relays.

1. General purpose relay: if a power failure occurs during MLC operation, the status of the relay is reset to Off and remains Off when power is resumed.
2. Non-volatile auxiliary relay: if a power failure occurs during MLC operation, the status of the relay is retained and the status remains after power is resumed.
3. Special auxiliary relay: for sending NC / MLC status or signals. MLC special relay can be used for all devices. For example, M2944 is for C64 to count down. Each special relay has its own specific function. Do not use the undefined ones.

Pay attention to the following when using the auxiliary relays.

It is suggested to use a unique number for the auxiliary relay output, or the output status is determined by the last circuit which outputs M according to the MLC program scan principle.

1.5 Numbers and functions of the user-defined alarm relay (A)

You can use the user-defined alarm relay to trigger alarms with specific I/O actions for finding the user-defined errors when programming the MLC. The alarm relay starts from A0.

User-defined alarm relay (A)		
General purpose	A0 - A511, 512 points. Volatile.	512 points in total

1.5.1 Function of user-defined alarm relay

Same as the output relay Y, the user-defined alarm relay A has output coil and A & B contacts, and they can be used for unlimited times in the program. You can use the user-defined alarm relay A to combine the control loop but cannot directly drive the external load. If a power failure occurs during MLC operation, the status of the general purpose user-defined alarm relay is reset to Off and remains Off when power is resumed.

1.6 Numbers and functions of the timer (T)

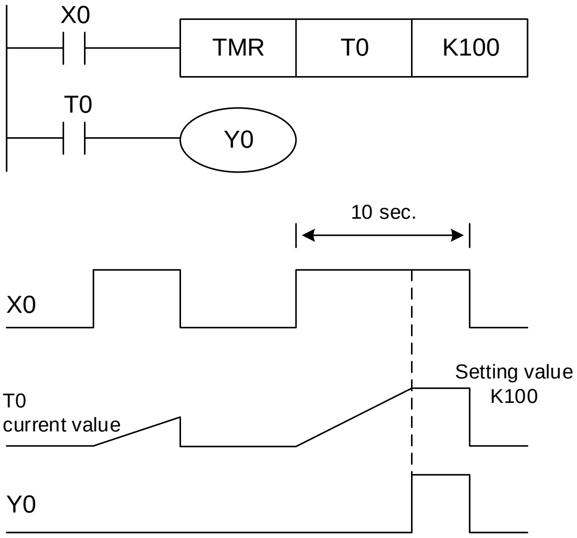
You can use the timer to count with specific I/O actions to have the device execute the programmed action after being triggered for a specific period of time when programming the MLC. The timer starts from T0.

Timer (T)		
General purpose, time-base unit: 100 ms	T0 - T199, 200 points.	256 points in total
General purpose, time-base unit: 10 ms	T200 - T255, 56 points.	

1.6.1 Timer settings

The MLC timer offers 10 ms and 100 ms increments and counts up. When the current time of the timer reaches the setting time, the output coil is On. The setting value a decimal K, or you can specify the data register D as the setting value.

Timer: the general purpose timer starts to count when the TMR instruction is executed. When the TMR instruction is in execution, once the setting time is reached, the output coil is On. See the following diagram.



When X0 is On, the timer T0 counts up by the increment of 100 ms. Once the current value of the timer is equal to the setting value K100, the output coil T0 is On. When X0 is Off or a power failure occurs, the current value of the timer T0 is reset to 0, and the output coil T0 is Off.

Methods for specifying the setting value are as follows.

The actual setting value of the timer = time unit * setting value.

Constant K: directly specify the constant K as the setting value.

Register D: indirectly specify the data in register D as the setting value.

1.7 Numbers and functions of the counter (C)

You can use the counter to count with specific I/O actions to have the elements execute the programmed actions after being triggered for specific times when programming the MLC.

The counter starts from C0.

Counter (C)		
16-bit up counter, general purpose	C0 - C63, 64 points. Volatile.	80 points in total
32-bit up/down counter, general purpose	C64 - C77, 14 points. Use M2944 - M2957 for the counter to count down.	
32-bit high-speed counter	C78, C79	

Item	16-bit counter	32-bit counter
Type	General purpose	General purpose
Count direction	Counting up	Counting up and counting down
Setting value	0 to 65,535	-2,147,483,648 to +2,147,483,647
Type of setting value	Constant K or data register D	Constant K or data register D (assign two registers)
Change of current value	Stop counting once the setting value is reached	Stop counting once the setting value is reached
Output contact	The contact is On and remains On once the setting value is reached	The contact is On and remains On when the counter counts up and reaches the setting value The contact is On and remains On when the counter counts down and reaches the setting value
Reset	The RST instruction resets the current value to 0 and the contact to Off.	
Contact action	The contacts are on after the MLC scan is complete.	

1.7.1 Functions of the counter

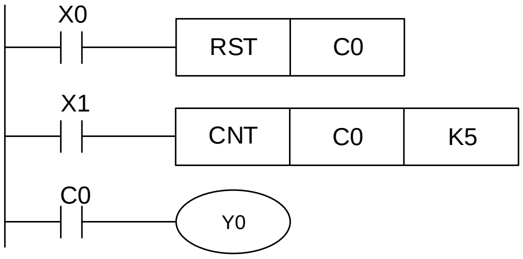
When the input signal of the counter pulse changes from Off to On, the current value of the counter increments by 1. If the input signal remains On, the current value continues to increment by 1. When the current value of the counter reaches the setting value, the output coil is On. The setting value is a decimal K, or you can specify the data register D as the setting value. The following describes the functions of the 16-bit and 32-bit counters.

- 16-bit counter: C0 - C63

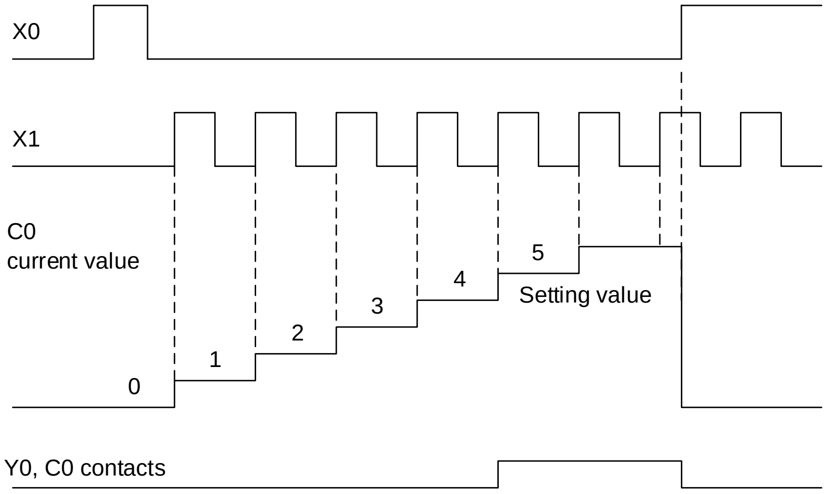
The setting range of the 16-bit counter is K0 - K65,535. (When the setting value is K0 or K1, the output contact is On upon the first count.) You can set the setting value of the counter with the constant K or with the value in the data register D (not including the special data registers D1024 - D1535).

Example:

1. When X0 is On, the RST instruction is executed to reset the current value of C0 to 0 and the output contact to Off.
2. When X1 changes from Off to On, the current value of the counter increments by 1. If X1 remains On, the current value continues to increment by 1.
3. When the current value of the counter C0 reaches the setting value K5, the C0 contact is On. When the current value of C0 is equal to the setting value K5, the current value of C0 remains at K5 and is not triggered by the X1 signal until the current value of C0 is reset to 0 when X0 becomes On.



Counter C0 - timing diagram of contact actions



- 32-bit counter: C64 - C77

The setting range of the 32-bit general purpose counter is K-2,147,483,648 to K+2,147,483,647. To switch the 32-bit counter to count up or count down, the special auxiliary relays M2944 - M2957 are required. For example, when M2944 is Off, C64 counts up; when M2944 is On, C64 counts down. The same is true for the remaining special auxiliary relays. You can use the constant K or the data register D as the setting value, which can be either positive or negative. To use the data register D, two consecutive data registers are required for one setting value. If the current value of the counter is 2,147,483,647, the value becomes -2,147,483,648 when the counter counts up. Similarly, if the current value of the counter is -2,147,483,648, the value becomes 2,147,483,647 when the counter counts down.

- 32-bit high-speed counter: C78 - C79

Different from C0 - C77, the 32-bit high-speed counter only takes effect when used with the specified application instruction. For detailed information, refer to DHSCS (Compare setup) and DHSCR (Compare reset) in Chapter 4.

1.8 Numbers and functions of the registers (D, V, Z)

1.8.1 Data register (D)

Data registers are for storing 16-bit numeric data in the range of -32,768 to +32,767. The highest bit is the positive or negative sign. You can combine two 16-bit registers into one 32-bit register. If you only assign one D for a 32-bit data, the system automatically assigns the register of the number D+1 as the upper 16-bit and the register of the number D as the lower 16-bit (refer to the following example). The highest bit is the positive or negative sign. These registers can store the numeric data in the range of -2,147,483,648 to +2,147,483,647.

For example, if you assign D0 for a 32-bit data, the system automatically assigns D1 for the same 32-bit data, with D0 as the lower 16-bit and D1 as the upper 16-bit.

Data register (D)		
General purpose	D0 - D511, 512 points	1,536 points in total
Non-volatile	D512 - D1023, 512 points	
Special purpose	D1024 - D1535, 512 points	

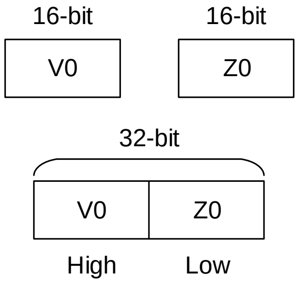
Table 1.8.1.1 Types of data registers

Data registers are divided into the four types as follows.

1. General purpose register: when MLC status is switched from “RUN (in execution)” to “MLC Stop”, the data is retained, but the data is reset to 0 when power is off.
2. Non-volatile register: when the MLC power is off, data in these registers is not cleared. And the data before power off is retained. To clear the contents in the non-volatile registers, use the RST or ZRST instruction.
3. Special purpose register: every special purpose register has their own definition and usage, mainly used for storing system statuses, error messages, and monitored statuses.
4. Index register (V, Z): index registers are 16-bit registers, including 16 points, V0 - V7 and Z0 - Z7. To use index registers as 32-bit registers, you need to use V registers. When V registers are used for 32-bit instructions, Z registers are unavailable. Refer to Section 1.8.2 for details.

1.8.2 Index registers (V, Z)

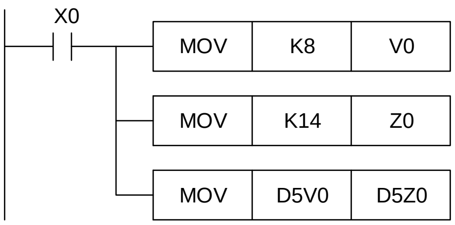
Same as general data registers, V and Z registers are 16-bit data registers which can be freely read and written. As for 32-bit data, you have to assign V registers, and Z registers are included in V registers in this case. Thus, Z registers can no longer be used, or the contents in V registers (32-bit data) would be incorrect. Refer to the following figure and table for detailed information.



Index registers V & Z for 32-bit data	
V0	Z0
V1	Z1
V2	Z2
V3	Z3
V4	Z4
V5	Z5
V6	Z6
V7	Z7

Same as the general operands, the index registers can be used for data movement or comparison, but some instructions do not support index registers. Therefore, V and Z can be used for modifying operands.

Example:



When X0 is On, set V0 to 8 and Z0 to 14, and then D5V0 = D(5+8) = D13 and D5Z0 = D(5+14) = D19. Thus, the content in D13 is moved to D19.

1.9 Indicators (N, P) and interrupt indicator (I)

MLC has indicators N, P, and I, which you can use to have the MLC run the specified programs when programming, reducing the error caused by the scan time of MLC.

Indicator				
N	For master control loop		N0 - N7, 8 points	Control point of master control loop
P	For CJ and CALL		P0 - P255, 256 points	Position indicator of CJ and CALL
I	For interruption	On-board hardware interrupt	IX00 - IX07, 8 points	Position indicator of the interrupt subroutine
		Hardware counting interrupt	IC00 - IC01, 2 points	
		Remote I/O hardware interrupt	IR00 - IR31, 32 points	

1. Indicators N and P

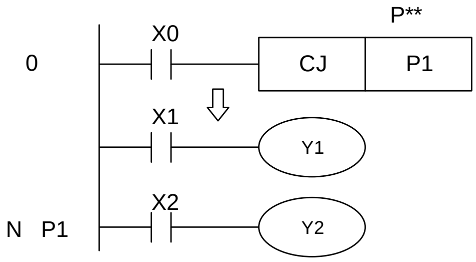
- Indicator N: used with the MC (master control start) and MCR instructions. When MC is executed, the instructions between MC and MCR are executed normally.
- Indicator P: used with the application instructions API 00 CJ, API 01 CALL, and API 02 SRET. For detailed information, refer to the descriptions of CJ, CALL, and SRET instructions in Chapter 4.

Example 1

When X0 is On, the program automatically jumps from address 0 to address N (the specified label P1) and continues to execute, skipping the addresses in the middle.

When X0 is Off, the program starts from address 0 and continues executing the programs in sequence without executing the CJ instruction.

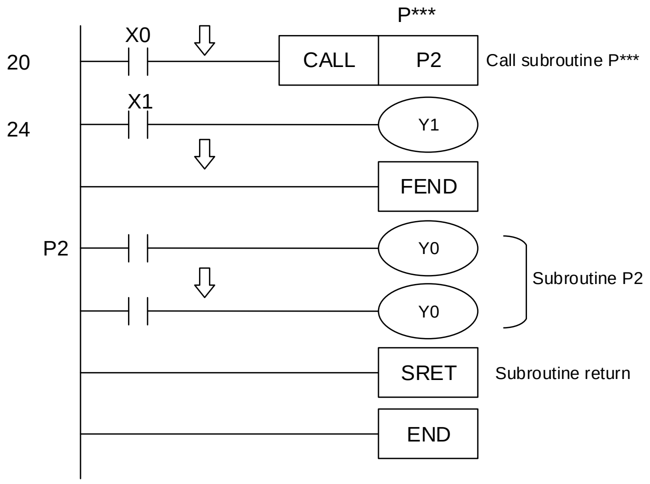
Conditional jump (CJ):



Example 2

When X0 is On, the system executes the CALL instruction and jumps to the subroutine specified by P2. As soon as the SRET instruction is executed, the system jumps back to address 24 and continues executing the programs.

Call subroutine (CALL) and subroutine return (SRET)



2. Interrupt indicator I

The interrupt action has to be combined with the instructions EI (Enable Interrupt), DI (Disable Interrupt), and IRET (Interrupt Return). Refer to Chapter 4 for detailed descriptions.

- External interrupt: because of the special design of the software in the MLC, when the input signals of X0 - X7 are rising-edge triggered or falling-edge triggered, the system finishes the instruction in execution at present and immediately jumps to the interrupt instruction. The system jumps back to the original program and continues to execute when the IRET instruction is executed. The external interrupt signals include: subroutine interrupt indicators, IX00(X0), IX01(X1), IX02(X2), IX03(X3), IX04(X4), IX05(X5), IX06(X6), and IX07(X7), and IR00 - IR31, which correspond to the 32 X inputs (X256 - X287) of the 0th remote I/O card.
- Interrupt when the counter reaches the set value: the compare instruction API 32 DHSCS of the high-speed counter can have the system interrupt the program in execution and jump to the specified interrupt subroutine when the set value is reached. The interrupt indicators are IC00 and IC01.

Basic MLC Instructions 2

This chapter provides detailed descriptions and usages for the basic MLC instructions.

2.1 List of basic instructions

NC series MLC uses many different basic instructions. This section lists all of the basic instructions and their functions, operands, execution speed, and STEP(S).

- General instructions

Instruction code	Function	Operand	Execution speed (μs)	STEP
LD	Load contact A	X, Y, M, A, T, C	-	1 - 2
LDI	Load contact B	X, Y, M, A, T, C	-	1 - 2
AND	Serial connect contact A	X, Y, M, A, T, C	-	1 - 2
ANI	Serial connect contact B	X, Y, M, A, T, C	-	1 - 2
OR	Parallel connect contact A	X, Y, M, A, T, C	-	1 - 2
ORI	Parallel connect contact B	X, Y, M, A, T, C	-	1 - 2
ANB	Serial connect circuit block	-	-	1
ORB	Parallel connect circuit block	-	-	1
MPS	Store in stack	-	-	1
MRD	Read stack (stack pointer remains)	-	-	1
MPP	Read stack	-	-	1

- Output instructions

Instruction code	Function	Operand	Execution speed (μs)	STEP
OUT	Driving coil	Y, M, A	-	1 - 2
SET	Action remains (ON)	Y, M, A	-	1 - 2
RST	Clear contact or register	Y, M, A, T, C, D, V, Z	-	1 - 2

- Timer and counter

Instruction code	Function	Operand	Execution speed (μs)	STEP
TMR	16-bit timer	T-K or T-D	9.6	3
CNT	16-bit counter	C-K or C-D (16-bit)	12.8	3
DCNT	32-bit counter	C-K or C-D (32-bit)	14.3	3

- Main control instructions

Instruction code	Function	Operand	Execution speed (μs)	STEP
MC	Connection of serial contacts	N0 - N7	5.6	1
MCR	Disconnection of serial contacts	N0 - N7	5.7	1

- Contact rising / falling edge detection instructions

Instruction code	Function	Operand	Execution speed (μs)	STEP
LDP	Start rising edge detection	X, Y, M, A, T, C	-	2
LDF	Start falling edge detection	X, Y, M, A, T, C	-	2
ANDP	Serial connection of rising edge detection	X, Y, M, A, T, C	-	2
ANDF	Serial connection of falling edge detection	X, Y, M, A, T, C	-	2

Instruction code	Function	Operand	Execution speed (μs)	STEP
ORP	Parallel connection of rising edge detection	X, Y, M, A, T, C	-	2
ORF	Parallel connection of falling edge detection	X, Y, M, A, T, C	-	2

- Upper and lower differential output instructions

Instruction code	Function	Operand	Execution speed (μs)	STEP
PLS	Upper differential output	X, Y, M, A, T, C	-	3
PLF	Lower differential output	X, Y, M, A, T, C	-	3

- End instruction

Instruction code	Function	Operand	Execution speed (μs)	STEP
END	Program ends	-	-	1

- Other instructions

Instruction code	Function	Operand	Execution speed (μs)	STEP
NOP	No operation	-	-	1
INV	Invert the operation result		-	1
P	Indicator	P0 - P255	-	1
I	Interrupt indicator	IX□□, IC□□, IR□□ (Refer to Chapter 4 for the values of □□.)	-	1

2.2 Description of basic instructions

This section provides detailed information about the function, operand, description, usage, and example of each instruction.

■ LD: Load contact A

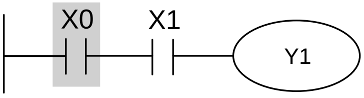
Instruction	Function	Applicable model
LD	Load contact A	NC series

Operand							
X0 - X39 X64 - X511	Y0 - Y39 Y64 - Y511	M0 - M3,071	A0 - A511	T0 - T255	C0 - C77	D0 - D1,535	V, Z
▪	▪	▪	▪	▪	▪	-	-

Description:

The LD instruction applies to contact A at the beginning of the left rail or contact A at the beginning of a contact loop block. Use the LD instruction to save the current content and store the acquired contact status in the accumulator register.

Example:



LD (X0) ladder diagram

■ LDI: Load contact B

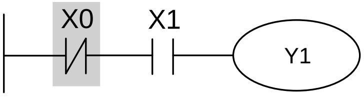
Instruction	Function	Applicable model
LDI	Load contact B	NC series

Operand							
X0 - X39 X64 - X511	Y0 - Y39 Y64 - Y511	M0 - M3,071	A0 - A511	T0 - T255	C0 - C77	D0 - D1,535	V, Z
▪	▪	▪	▪	▪	▪	-	-

Description:

The LDI instruction applies to contact B at the beginning of the left rail or contact B at the beginning of a contact loop block. Use the LDI instruction to save the current content and store the acquired contact status in the accumulator register.

Example:



LDI (X0) ladder diagram

■ AND: Serial connect contact A

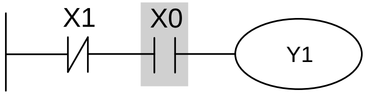
Instruction	Function	Applicable model
AND	Serial connect contact A	NC series

Operand							
X0 - X39 X64 - X511	Y0 - Y39 Y64 - Y511	M0 - M3,071	A0 - A511	T0 - T255	C0 - C77	D0 - D1,535	V, Z
▪	▪	▪	▪	▪	▪	-	-

Description:

The AND instruction connects contact A in series. It reads the current status of the specified contacts and performs the AND operation using the acquired data with the results of the logic operation prior to the contact. It stores the result in the accumulator register.

Example:



AND (X0) ladder diagram

■ ANI: Serial connect contact B

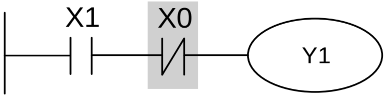
Instruction	Function	Applicable model
ANI	Serial connect contact B	NC series

Operand							
X0 - X39 X64 - X511	Y0 - Y39 Y64 - Y511	M0 - M3,071	A0 - A511	T0 - T255	C0 - C77	D0 - D1,535	V, Z
▪	▪	▪	▪	▪	▪	-	-

Description:

The ANI instruction connects contact B in series. It reads the current status of the specified contacts and performs the AND operation using the acquired data with the results of the logic operation prior to the contact. It stores the result in the accumulator register.

Example:



ANI (X0) ladder diagram

■ OR: Parallel connect contact A

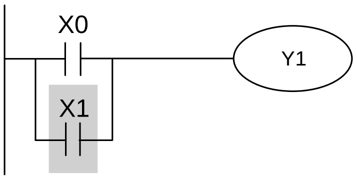
Instruction	Function	Applicable model
OR	Parallel connect contact A	NC series

Operand							
X0 - X39 X64 - X511	Y0 - Y39 Y64 - Y511	M0 - M3,071	A0 - A511	T0 - T255	C0 - C77	D0 - D1,535	V, Z
▪	▪	▪	▪	▪	▪	-	-

Description:

The OR instruction parallel connects contact A. It reads the current status of the specified contacts and performs the OR operation using the acquired data with the results of the logic operation prior to the contact. It stores the result in the accumulator register.

Example:



OR (X1) ladder diagram

■ ORI: Parallel connect contact B

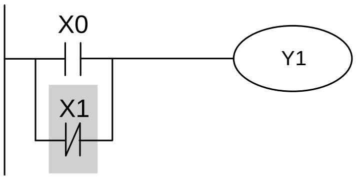
Instruction	Function	Applicable model
ORI	Parallel connect contact B	NC series

Operand							
X0 - X39 X64 - X511	Y0 - Y39 Y64 - Y511	M0 - M3,071	A0 - A511	T0 - T255	C0 - C77	D0 - D1,535	V, Z
▪	▪	▪	▪	▪	▪	-	-

Description:

The ORI instruction parallel connects contact B. It reads the current status of the specified contacts and performs the OR operation using the acquired data with the results of the logic operation prior to the contact. It stores the result in the accumulator register.

Example:



ORI (X1) ladder diagram

ANB: Serial connect circuit block

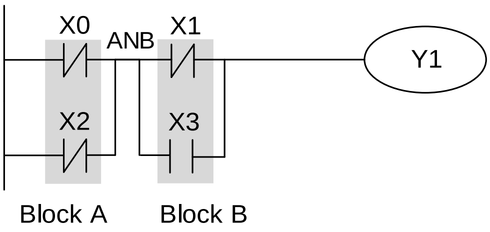
Instruction	Function	Applicable model
ANB	Serial connect circuit block	NC series

Operand							
X0 - X39 X64 - X511	Y0 - Y39 Y64 - Y511	M0 - M3,071	A0 - A511	T0 - T255	C0 - C77	D0 - D1,535	V, Z
-	-	-	-	-	-	-	-

Description:

The ANB instruction performs the AND operation using the results of the previously saved logic operation and the current value in the accumulator register.

Example:



ANB (X0+X2), (X1+X3) ladder diagram

ORB: Parallel connect circuit block

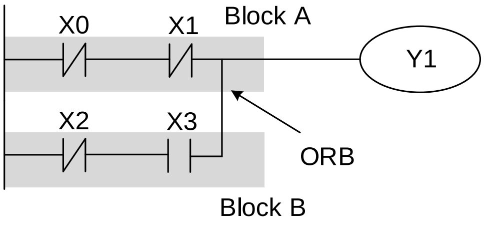
Instruction	Function	Applicable model
ORB	Parallel connect circuit block	NC series

Operand							
X0 - X39 X64 - X511	Y0 - Y39 Y64 - Y511	M0 - M3,071	A0 - A511	T0 - T255	C0 - C77	D0 - D1,535	V, Z
-	-	-	-	-	-	-	-

Description:

The ORB instruction performs the OR operation using the results of the previously saved logic operation and the current value in the accumulator register.

Example:



ORB (X0+X1), (X2+X3) ladder diagram

■ MPS: Store in stack

Instruction	Function	Applicable model
MPS	Store in stack	NC series

Operand							
X0 - X39 X64 - X511	Y0 - Y39 Y64 - Y511	M0 - M3,071	A0 - A511	T0 - T255	C0 - C77	D0 - D1,535	V, Z
-	-	-	-	-	-	-	-

Description:

The MPS instruction stores the current value in the accumulator register to the stack register. (Stack pointer increases by 1.)

■ MRD: Read stack (stack pointer remains)

Instruction	Function	Applicable model
MRD	Read stack (stack index remains)	NC series

Operand							
X0 - X39 X64 - X511	Y0 - Y39 Y64 - Y511	M0 - M3,071	A0 - A511	T0 - T255	C0 - C77	D0 - D1,535	V, Z
-	-	-	-	-	-	-	-

Description:

The MRD instruction reads the content in the stack and stores it in the accumulator register. (Stack pointer remains unchanged.)

■ MPP: Retrieve from stack

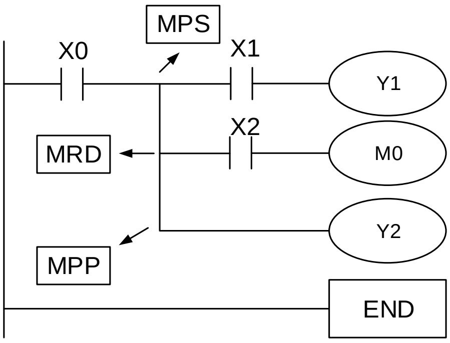
Instruction	Function	Applicable model
MPP	Retrieve from stack	NC series

Operand							
X0 - X39 X64 - X511	Y0 - Y39 Y64 - Y511	M0 - M3,071	A0 - A511	T0 - T255	C0 - C77	D0 - D1,535	V, Z
-	-	-	-	-	-	-	-

Description:

The MPP instruction retrieves the results of the previously saved logic operation in the stack and stores it in the accumulator register. (Stack pointer decreases by 1.)

Example:



MPS, MRD, MPP ladder diagram

■ OUT: Drive coil

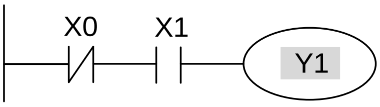
Instruction	Function	Applicable model
OUT	Drive coil	NC series

Operand							
X0 - X39 X64 - X511	Y0 - Y39 Y64 - Y511	M0 - M3,071	A0 - A511	T0 - T255	C0 - C77	D0 - D1,535	V, Z
-	▪	▪	▪	-	-	-	-

Description:

The OUT instruction outputs the results of the logic operation prior to the OUT coil to the specified bit.

Example:



OUT (Y1) ladder diagram

■ SET: Action remains (ON)

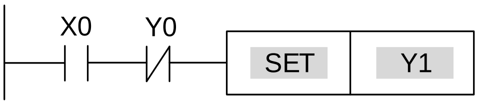
Instruction	Function	Applicable model
SET	Action remains (ON)	NC series

Operand							
X0 - X39 X64 - X511	Y0 - Y39 Y64 - Y511	M0 - M3,071	A0 - A511	T0 - T255	C0 - C77	D0 - D1,535	V, Z
-	▪	▪	▪	-	-	-	-

Description:

When the SET instruction is triggered, the specified bit is set to On and remains On. You can use the RST instruction to set this bit to Off no matter the SET instruction is triggered or not.

Example:



SET (Y1) ladder diagram

■ RST: Clear contact or register

Instruction	Function	Applicable model
RST	Clear contact or register	NC series

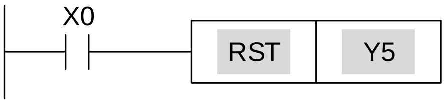
Operand							
X0 - X39 X64 - X511	Y0 - Y39 Y64 - Y511	M0 - M3,071	A0 - A511	T0 - T255	C0 - C77	D0 - D1,535	V, Z
-	▪	▪	▪	▪	▪	▪	▪

Description:

When the RST instruction executes, the actions of the specified devices are as follows.

Device	Action
Y, M, A	Coils and contacts are set to Off.
T, C	The current timing or count value resets to 0 and the coils and contacts are set to Off.
D, V, Z	The content value resets to 0.

Example:



RST (Y5) ladder diagram

■ TMR: 16-bit timer

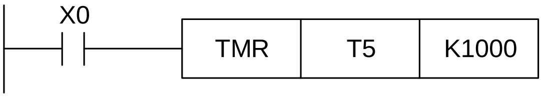
Instruction	Function	Applicable model
TMR	16-bit timer	NC series

Operand	
T-K	T0 - T255, K0 - K65,535
T-D	T0 - T255, D0 - D1,535

Description:

When the TMR instruction executes, the specified timer coil is energized and the timer starts counting. When reaching the set timing, the specified timer is set to On. When the TMR instruction stops executing, the timer value is reset to zero.

Example:



TMR (T5) ladder diagram

■ CNT: 16-bit counter

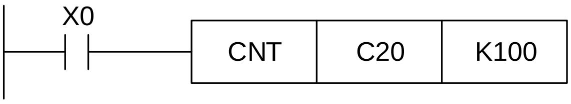
Instruction	Function	Applicable model
CNT	16-bit counter	NC series

Operand	
C-K	C0 - C63, K0 - K65,535
C-D	C0 - C63, D0 - D1,535

Description:

When the CNT instruction changes from Off to On, the specified counter's coil switches from de-energized to energized state and increases the count value by 1. When reaching the set count, the specified counter is set to On. When the counter reaches the set count, the counter's contacts and count value remain the same even when receiving more counting pulse inputs. You can use the RST instruction to restart counting or clear the value.

Example:



CNT (C20) ladder diagram

■ DCNT: 32-bit counter

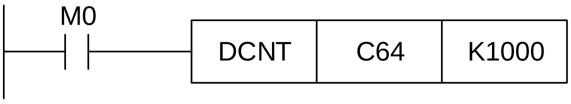
Instruction	Function	Applicable model
DCNT	32-bit counter	NC series

Operand	
C-K	C64 - C77, K-2,147,483,648 to K+2,147,483,647
C-D	C64 - C77, D0 - D1,535

Description:

The DCNT instruction enables the 32-bit counters C64 - C77. When C64 - C77 is used and the DCNT instruction changes from Off to On, the counter's current value increases or decreases by 1, and the increase or decrease of the value is determined by the status of special M (M2944 - M2957).

Example:



DCNT (C64) ladder diagram

■ MC / MCR: Connection / disconnection of serial contacts

Instruction	Function	Applicable model
MC / MCR	Connection / disconnection of serial contacts	NC series

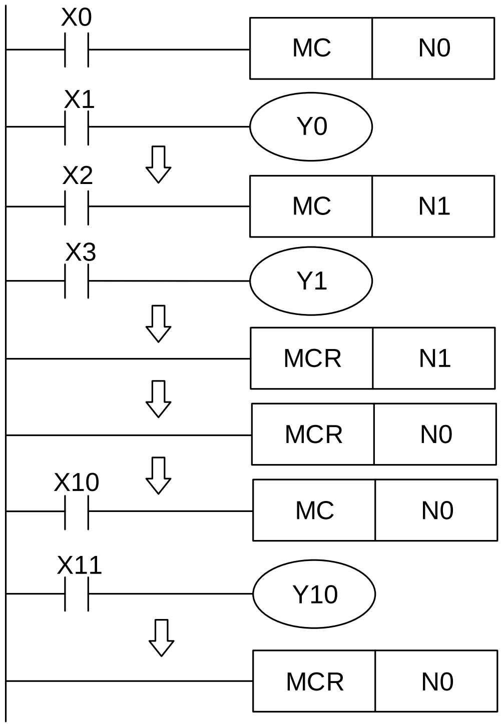
Operand
N0 - N7

Description:

The MC instruction is the master control start instruction. When the MC instruction is executed, the instructions between MC and MCR instructions are executed as usual. When the MC instruction is Off, the actions of the instructions between MC and MCR instructions are as follows.

Instruction type	Action
General timer	The timer value is reset to zero, the coil is de-energized, and the contact remains inactive.
Counter	The coil is de-energized, and the count value and contacts remain at the current state.
Coils driven by the OUT instruction	All coils are de-energized.
Devices driven by the SET and RST instructions	The devices remain at the current state.
Application instructions	All instructions remain inactive. The FOR-NEXT nested loop executes for N times, but any instruction in the FOR-NEXT loop will execute the same actions as the instructions between MC and MCR.

Example:



MC / MCR ladder diagram

■ LDP: Start rising edge detection

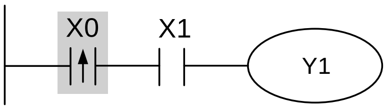
Instruction	Function	Applicable model
LDP	Start rising edge detection	NC series

Operand							
X0 - X39 X64 - X511	Y0 - Y39 Y64 - Y511	M0 - M3,071	A0 - A511	T0 - T255	C0 - C77	D0 - D1,535	V, Z
▪	▪	▪	▪	▪	▪	-	-

Description:

The usage of the LDP instruction is the same as that of LD, but the action is different. Use the LDP instruction to save the current content and store the acquired contact status of the rising edge in the accumulator register.

Example:



LDP (X0) ladder diagram

■ LDF: Start falling edge detection

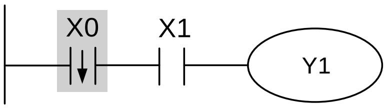
Instruction	Function	Applicable model
LDF	Start falling edge detection	NC series

Operand							
X0 - X39 X64 - X511	Y0 - Y39 Y64 - Y511	M0 - M3,071	A0 - A511	T0 - T255	C0 - C77	D0 - D1,535	V, Z
▪	▪	▪	▪	▪	▪	-	-

Description:

The usage of the LDF instruction is the same as that of LD, but the action is different. Use the LDF instruction to save the current content and store the acquired contact status of the falling edge in the accumulator register.

Example:



LDF (X0) ladder diagram

■ ANDP: Serial connection of rising edge detection

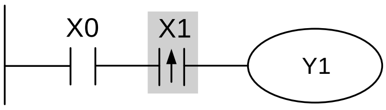
Instruction	Function	Applicable model
ANDP	Serial connection of rising edge detection	NC series

Operand							
X0 - X39 X64 - X511	Y0 - Y39 Y64 - Y511	M0 - M3,071	A0 - A511	T0 - T255	C0 - C77	D0 - D1,535	V, Z
▪	▪	▪	▪	▪	▪	-	-

Description:

The ANDP instruction serial connects the rising edge detection of the contact.

Example:



ANDP (X1) ladder diagram

■ ANDF: Serial connection of falling edge detection

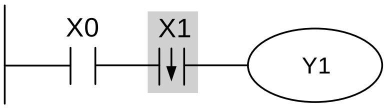
Instruction	Function	Applicable model
ANDF	Serial connection of falling edge detection	NC series

Operand							
X0 - X39 X64 - X511	Y0 - Y39 Y64 - Y511	M0 - M3,071	A0 - A511	T0 - T255	C0 - C77	D0 - D1,535	V, Z
▪	▪	▪	▪	▪	▪	-	-

Description:

The ANDF instruction serial connects the falling edge detection of the contact.

Example:



ANDF (X1) ladder diagram

■ ORP: Parallel connection of rising edge detection

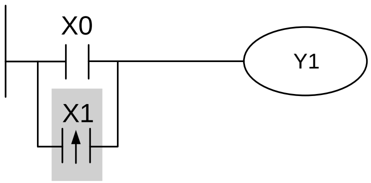
Instruction	Function	Applicable model
ORP	Parallel connection of rising edge detection	NC series

Operand							
X0 - X39 X64 - X511	Y0 - Y39 Y64 - Y511	M0 - M3,071	A0 - A511	T0 - T255	C0 - C77	D0 - D1,535	V, Z
▪	▪	▪	▪	▪	▪	-	-

Description:

The ORP instruction parallel connects the rising edge detection of the contact.

Example:



ORP (X0, X1) ladder diagram

■ ORF: Parallel connection of falling edge detection

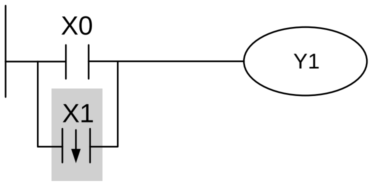
Instruction	Function	Applicable model
ORF	Parallel connection of falling edge detection	NC series

Operand							
X0 - X39 X64 - X511	Y0 - Y39 Y64 - Y511	M0 - M3,071	A0 - A511	T0 - T255	C0 - C77	D0 - D1,535	V, Z
▪	▪	▪	▪	▪	▪	-	-

Description:

The ORF instruction parallel connects the falling edge detection of the contact.

Example:



ORF (X0, X1) ladder diagram

■ PLS: Upper differential output

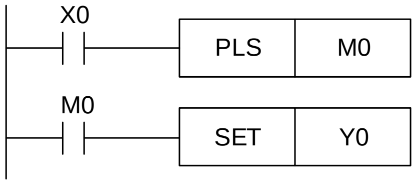
Instruction	Function	Applicable model
PLS	Upper differential output	NC series

Operand							
X0 - X39 X64 - X511	Y0 - Y39 Y64 - Y511	M0 - M3,071	A0 - A511	T0 - T255	C0 - C77	D0 - D1,535	V, Z
▪	▪	▪	▪	▪	▪	-	-

Description:

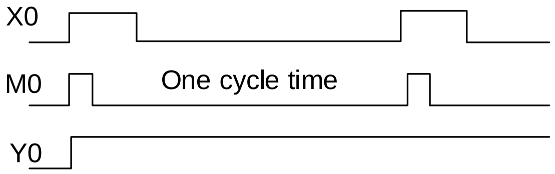
The PLS instruction is for the upper differential output. When X0 switches from Off to On (rising-edge triggered), the PLS instruction executes and M0 sends one pulse with the length of one scanning time cycle.

Example:



PLS (M0) ladder diagram

Timing diagram:



PLS (M0) timing diagram

■ PLF: Lower differential output

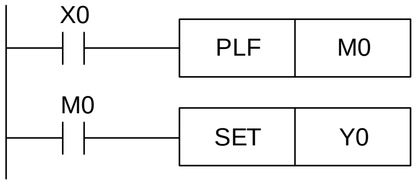
Instruction	Function	Application model
PLF	Lower differential output	NC series

Operand							
X0 - X39 X64 - X511	Y0 - Y39 Y64 - Y511	M0 - M3,071	A0 - A511	T0 - T255	C0 - C77	D0 - D1,535	V, Z
▪	▪	▪	▪	▪	▪	-	-

Description:

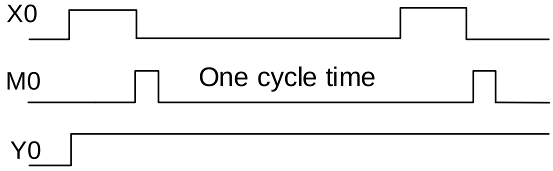
The PLF instruction is for the lower differential output. When X0 switches from On to Off (falling-edge triggered), the PLF instruction executes and M0 sends one pulse with the length of one scanning time cycle.

Example:



PLF (M0) ladder diagram

Timing diagram:



PLF (M0) timing diagram

■ END: Program ends

Instruction	Function	Applicable model
END	Program ends	NC series

Operand							
X0 - X39 X64 - X511	Y0 - Y39 Y64 - Y511	M0 - M3,071	A0 - A511	T0 - T255	C0 - C77	D0 - D1,535	V, Z
-	-	-	-	-	-	-	-

Description:

The END instruction is the ladder or program’s last instruction. PLC scans the program from address 0 to the END instruction, and then returns to address 0 to restart the scan.

NOP: No operation

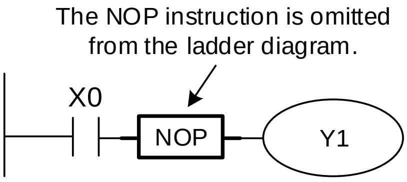
Instruction	Function	Applicable model
NOP	No operation	NC series

Operand							
X0 - X39 X64 - X511	Y0 - Y39 Y64 - Y511	M0 - M3,071	A0 - A511	T0 - T255	C0 - C77	D0 - D1,535	V, Z
-	-	-	-	-	-	-	-

Description:

The NOP instruction performs no operation in the program so the results of the logic operation remain the same after execution. Use this instruction when you want to delete an instruction without changing the program length.

Example:



NOP ladder diagram

INV: Invert the operation result

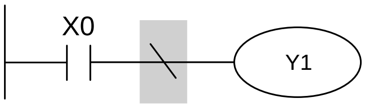
Instruction	Function	Applicable model
INV	Invert the operation result	NC series

Operand							
X0 - X39 X64 - X511	Y0 - Y39 Y64 - Y511	M0 - M3,071	A0 - A511	T0 - T255	C0 - C77	D0 - D1,535	V, Z
-	-	-	-	-	-	-	-

Description:

Inverts the results of the logic operation prior to the INV instruction and stores the results in the accumulator register.

Example:



INV ladder diagram

P: Indicator

Instruction	Function	Applicable model
P	Indicator	NC series

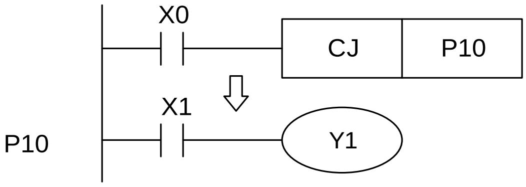
Operand
P0 - P255

Description:

Indicator P is used for the jump instruction CJ and subroutine calling instruction CALL.

Indicator P does not need to start with number 0, but the numbers cannot be used repeatedly, or unexpected errors will occur.

Example:



P ladder diagram

I: Interrupt indicator

Instruction	Function	Applicable model
I	Interrupt indicator	NC series

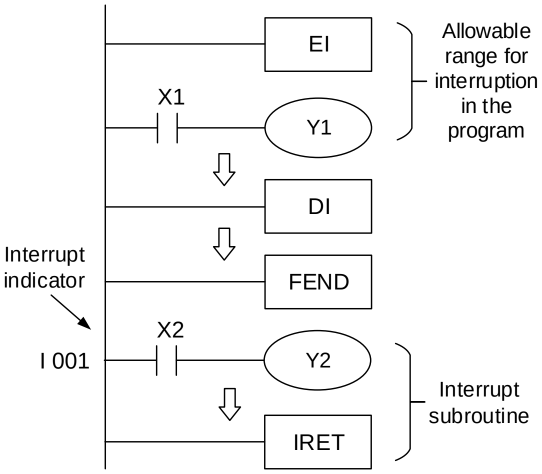
Operand
IX00 - IX07, IC00 - IC01, IR00 - IR31

Description:

The starting point of the interrupt service routine must be indicated with the interrupt indicator (I□□□).

Use the IRET instruction to end the interruption and return to the main program. Use the interrupt indicator with the IRET, EI, and DI application instructions.

Example:



I ladder diagram

MLC Application Instructions Overview

3

This chapter introduces the logic and format of the MLC application instructions.

3.1 List of application instructions

The NC series MLC includes many different application instructions which are listed in the following table. Refer to Chapter 4 for more details about these instructions.

Type	API	Instruction code		Number of operand	Function	STEP	
		16-bit	32-bit			16-bit	32-bit
Loop control	00	CJ	-	1	Conditional jump	2	-
	01	CALL	-	1	Call subroutine	2	-
	02	SRET	-	-	End of subroutine	1	-
	03	IRET	-	-	Return from interruption	1	-
	04	EI	-	-	Enable interruption	1	-
	05	DI	-	-	Disable interruption	1	-
	06	FEND	-	-	End of main program	1	-
	07	FOR	-	1	Start of nested loop	3	-
	08	NEXT	-	-	End of nested loop	1	-
Transmission and comparison	09	MOV	DMOV	2	Move data	4	6
	67	-	FMOV	2	Move data	-	6
	10	CML	DCML	2	Inverting and transfer	4	5
	11	BCD	DBCD	2	Convert BIN data to BCD data	4	4
	12	BIN	DBIN	2	Convert BCD data to BIN data	4	4
Four logical operations	13	ADD	DADD	3	BIN addition	6	8
	14	SUB	DSUB	3	BIN subtraction	6	8
	15	MUL	DMUL	3	BIN multiplication	6	8
	16	DIV	DDIV	3	BIN division	6	8
	17	INC	DINC	1	Plus one (BIN)	3	3
	18	DEC	DDEC	1	Minus one (BIN)	3	3
	19	WAND	DWAND	3	AND operation	6	8
	20	WOR	DWOR	3	OR operation	6	8
	21	WXOR	DWXOR	3	XOR operation	6	8
	22	NEG	DNEG	1	Take the negative number (2's complement)	3	3
Rotation	23	ROR	DROR	2	Rotate right	4	4
	24	ROL	DROL	2	Rotate left	4	4
Data processing	25	ZRST	-	2	Zone reset	4	-
	26	DECO	-	3	Decoder	6	-
	27	ENCO	-	3	Encoder	6	-
	28	BON	DBON	3	Bit state monitoring	6	7
	29	ANS	-	3	Alarm output	5	-
	30	ANR	-	-	Alarm reset	1	-
High-speed processing	31	REF	-	2	I/O refresh	3	-
	32	-	DHSCS	3	Comparison setting (high-speed counter)	-	5
	33	-	DHSCR	3	Comparison reset (high-speed counter)	-	5
Convenient instructions	34	ALT	-	1	On / Off alternation	3	-
	68	WRTL	-	2	Write the servo torque limit	4	-
	69	RDTL	-	2	Read the torque limit flag	4	-

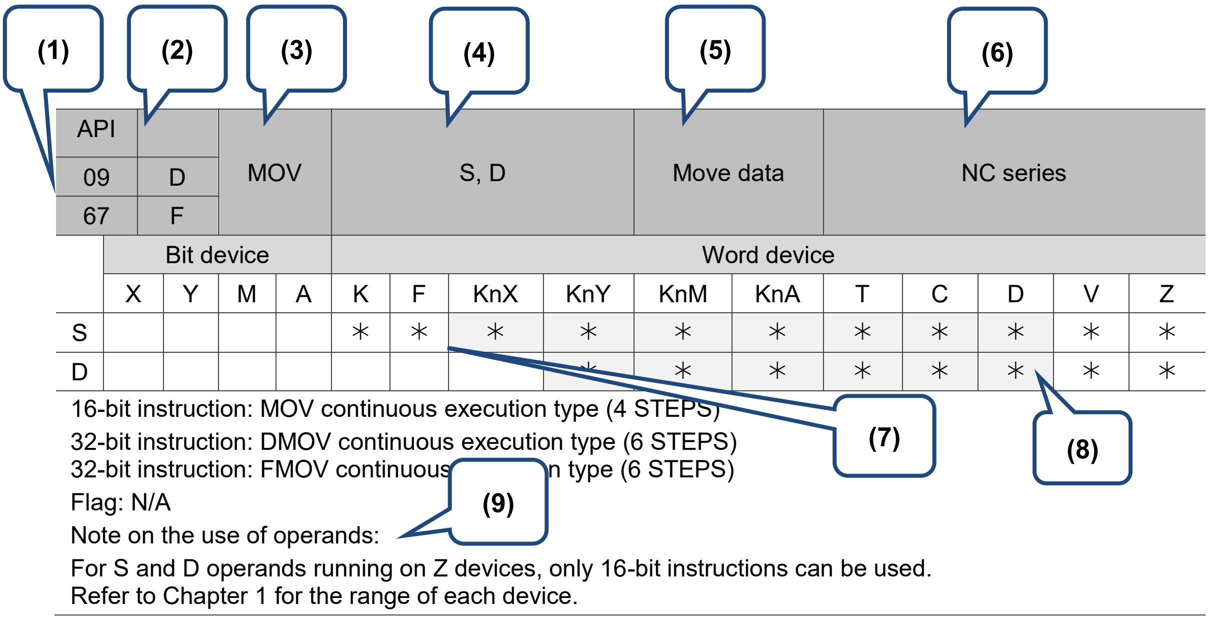
Type	API	Instruction code		Number of operand	Function	STEP	
		16-bit	32-bit			16-bit	32-bit
Basic instructions	35	PLS	-	1	Upper differential output	3	-
	36	TMR	-	2	Timer	3	-
	37	CNT	DCNT	2	Counter	3	3
	38	PLF	-	1	Lower differential output	3	-
Compare contact type	39	LD=	DLD=	2	$S_1 = S_2$	4	6
	40	LD>	DLD>	2	$S_1 > S_2$	4	6
	41	LD<	DLD<	2	$S_1 < S_2$	4	6
	42	LD<>	DLD<>	2	$S_1 \neq S_2$	4	6
	43	LD<=	DLD<=	2	$S_1 \leqq S_2$	4	6
	44	LD>=	DLD>=	2	$S_1 \geqq S_2$	4	6
	45	AND=	DAND=	2	$S_1 = S_2$	4	6
	46	AND>	DAND>	2	$S_1 > S_2$	4	6
	47	AND<	DAND<	2	$S_1 < S_2$	4	6
	48	AND<>	DAND<>	2	$S_1 \neq S_2$	4	6
	49	AND<=	DAND<=	2	$S_1 \leqq S_2$	4	6
	50	AND>=	DAND>=	2	$S_1 \geqq S_2$	4	6
	51	OR=	DOR=	2	$S_1 = S_2$	4	6
	52	OR>	DOR>	2	$S_1 > S_2$	4	6
	53	OR<	DOR<	2	$S_1 < S_2$	4	6
	54	OR<>	DOR<>	2	$S_1 \neq S_2$	4	6
	55	OR<=	DOR<=	2	$S_1 \leqq S_2$	4	6
	56	OR>=	DOR>=	2	$S_1 \geqq S_2$	4	6
	57	VRT	DVRT	3	Logical switch table	70	134
Floating point operation	58	-	FADD	3	Binary floating-point number addition	-	7
	59	-	FSUB	3	Binary floating-point number subtraction	-	7
	60	-	FMUL	3	Binary floating-point number multiplication	-	7
	61	-	FDIV	3	Binary floating-point division	-	7
	62	-	FCMP	3	Compare binary floating-point numbers	-	7
	63	-	FINT	2	Convert binary floating-point number to BIN integer	-	5
	64	-	FDOT	2	Convert BIN integer to binary floating point	-	5
	65	-	FRAD	2	Convert degrees to radians	-	5
	66	-	FDEG	2	Convert radians to degrees	-	5

Note: the instructions listed in the above table are applicable to NC series models.

3.2 Descriptions of the application instructions

Setting the application instructions are required for the various control methods of MLC.
This section introduces the format and composition of application instructions and explains the related terms.

3.2.1 Format of application instructions



API		MOV	S, D	Move data	NC series
09	D				
67	F				

	Bit device				Word device										
	X	Y	M	A	K	F	KnX	KnY	KnM	KnA	T	C	D	V	Z
S					*	*	*	*	*	*	*	*	*	*	*
D								*	*	*	*	*	*	*	*

16-bit instruction: MOV continuous execution type (4 STEPS)
32-bit instruction: DMOV continuous execution type (6 STEPS)
32-bit instruction: FMOV continuous n type (6 STEPS)
Flag: N/A
Note on the use of operands:
For S and D operands running on Z devices, only 16-bit instructions can be used.
Refer to Chapter 1 for the range of each device.

(1) Application instruction API code.

(2) The upper box indicates this application instruction is a 16-bit instruction. If there is a hyphen (-) in the box, it indicates there is no 16-bit instruction for this application instruction.
The lower box indicates this application instruction is a 32-bit instruction. For a 32-bit instruction, the box is left blank or indicated by D / F, for example, API 09 DMOV and API 59 FSUB. If there is a hyphen (-) in the box, it indicates there is no 32-bit instruction for this application instruction.

(3) Name of application instruction.

(4) Operand format of application instruction.

(5) Function description of application instruction.

(6) Applicable models for this application instruction.

(7) Devices marked with an asterisk (*) can be used by the operand.

(8) Devices marked with an asterisk (*) in a gray box can be modified with the index registers V and Z.

(9) Notes when applying operand.

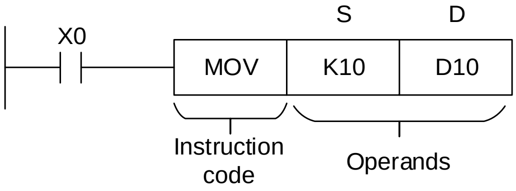
An application instruction is composed of two parts: instruction name and operand.
The instruction name indicates the function and the operand indicates the devices for the operation of this instruction.

The instruction name typically occupies 1 step and each one of its operand occupies 2 (16-bit instruction) or 4 (32-bit instruction) steps.

3.2.2 Input of application instructions

Most application instructions contain more than one operand, but there are some application instructions that have no operand, such as EI and DI.

The application instructions of the NC series MLC are specified by API 00 - API 69.
Each application instruction is represented by a unique instruction code. For example, the instruction code for API 09 is MOV (Move data). Different application instructions have different operands. Take MOV as an example. If you use the MLC Editor to input application instructions, simply input the instruction code “MOV” for API 09.



This instruction moves the value of the operand specified by S to the destination operand specified by D. In this instruction:

Source operand S: if there is more than one source operand, use S_1, S_2, and so on to represent the source operands.

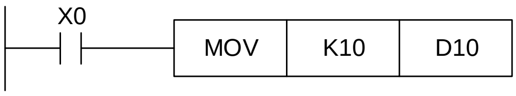
Destination operand D: if there is more than one destination operand, use D_1, D_2, and so on to represent the destination operands.

If the operand can only specify constant K / F or a register, use m, m1, m2, n, n1, and n2 to represent the operand.

3.2.3 Operand length (16-bit or 32-bit instruction)

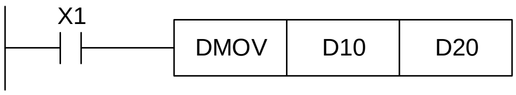
Depending on the content in the operand, the length of an operand can be 16 bits or 32 bits. 16-bit and 32-bit instructions are for processing data of different lengths. The letter “D” is prefixed to a 16-bit instruction to form a 32-bit instruction.

16-bit MOV instruction



When X0 = On, operand K10 is sent to operand D10.

32-bit DMOV instruction

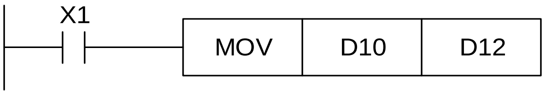


When X1 = On, operand (D11, D10) is sent to operand (D21, D20).

3.2.4 Instruction execution type

MLC instructions are executed continuously.

Example:



When X1 = On, the MOV instruction is executed once in each scan cycle. Thus, it is called a continuous execution instruction.

3.2.5 Specified object of the operand

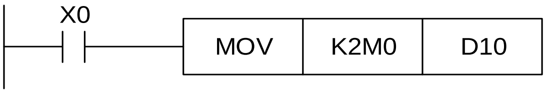
The specified object of the operand has the following features.

1. Bit devices X, Y, M, and A can be combined into word devices. In the application instruction, the data is stored in the form of KnX, KnY, KnM, and KnA for operations.
2. Data register D, timer T, counter C, and index registers V and Z are specified by general operands.
3. A data register is typically 16 bits in length which is also the length of a D register. Specify two consecutive D registers to form a 32-bit register.
4. If the operand of a 32-bit instruction specifies D0, the 32-bit data register formed by (D1, D0) is occupied. D1 is the upper 16 bits and D0 is the lower 16 bits. The same rule applies to the timer T and 16-bit counter.
5. When the 32-bit counters C64 - C77 are used as data registers, they can only be specified by the operand of a 32-bit instruction.

3.2.6 Devices of the operand

The definitions of the devices assigned by the operands are as follows.

1. Devices X, Y, M, and A can only be set as single points of On / Off, which are defined as bit devices.
2. 16-bit or 32-bit devices T, C, and D and registers V and Z are defined as word devices.
3. For word device operations, add Kn in front of the bit devices X, Y, M, and A to define them as word devices. n = 1 indicates 4 bits, so for a 16-bit instruction, n can be 1 - 4 and for a 32-bit instruction, n can be 1 - 8. For example, K2M0 indicates the length of the device is 8 bits, including M0 - M7.



When X0 = On, the values in M0 - M7 are moved to bits 0 - 7 of D10 and bits 8 - 15 are set to 0.

3.2.7 Data processing of the word devices formed by bit devices

The corresponding values of 16-bit and 32-bit instructions are as follows.

16-bit instructions	
The range of value specified by 16-bit instruction: K-32,768 to K+32,767	
Values contained in bit groups K1 - K4:	
K1 (4 bits)	0 - 15
K2 (8 bits)	0 - 255
K3 (12 bits)	0 - 4,095
K4 (16 bits)	-32,768 to +32,767

32-bit instructions	
The range of value specified by 32-bit instruction: K-2,147,483,648 to K+2,147,483,647	
Values contained in bit groups K1 - K8:	
K1 (4 bits)	0 - 15
K2 (8 bits)	0 - 255
K3 (12 bits)	0 - 4,095
K4 (16 bits)	0 - 65,535
K5 (20 bits)	0 - 1,048,575
K6 (24 bits)	0 - 167,772,165
K7 (28 bits)	0 - 268,435,455
K8 (32 bits)	-2,147,483,648 to +2,147,483,647

3.3 Processing numeric values with application instructions

This section describes how devices with numeric values are processed by the MLC application instructions.

Devices X, Y, M, and A with only On / Off functions are called bit devices. Devices T, C, D, V, and Z for storing numeric values are called word devices. With a specific declaration, bit devices can also be used in an operand of the application instruction in the form of a numeric value.
The declaration is to add the number of bits, expressed with Kn, in front of the bit device.
A 16-bit number is expressed by K1 - K4 and a 32-bit number is expressed by K1 - K8.
For example, K2M0 is an 8-bit number composed of M0 - M7. Send K1M0, K2M0, and K3M0 to a 16-bit register and fill the vacant upper bits with 0. Send K1M0, K2M0, K3M0, K4M0, K5M0, K6M0, and K7M0 to a 32-bit register and fill the vacant upper bits with 0. In a 16-bit or 32-bit operation, if the contents of the operand specify bit devices of K1 - K3 or K4 - K7, all vacant upper bits will be filled with 0, so this operation is generally regarded as a positive number operation.

- Specifying continuous numbers: take data register D as an example. The continuous numbers of D are D0, D1, D2, D3, D4, and so on. For bit devices with assigned numbers, the continuous numbers are as follows.

Specifying continuous numbers			
K1X0	K1X4	K1X8	K1X12......
K2Y0	K2Y8	K2Y16	K2Y24......
K3M0	K3M12	K3M24	K3M36......
K4A0	K4A16	K4A32	K4A48......

As shown in the above table, the continuous numbers of X devices for K1 are multiples of 4 and the continuous numbers of X devices for K2 are multiples of 8. Do not skip numbers to avoid confusion. For example, K1X0 and K1X5 are not multiples of 4.

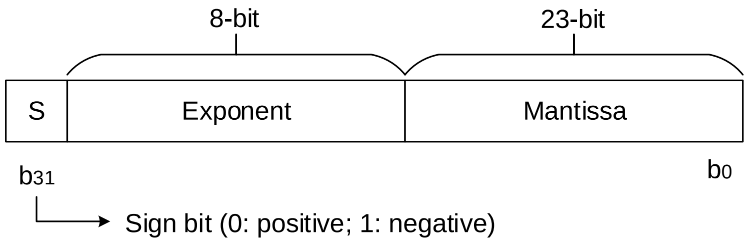
Note: if K4Y0 is used in a 32-bit operation, the upper 16 bits are regarded as 0. For 32-bit data, use K8Y0.

■ The internal numeric operation of the NC series MLC is conducted in the format of BIN integers.
If you use decimal point (floating point) operation instructions, the decimal values (floating point numbers) can be calculated.

Decimal point (floating point) relevant application instructions		
API 58 (FADD)	API 61 (FDIV)	API 64 (FDOT)
API 59 (FSUB)	API 62 (FCMP)	API 65 (FRAD)
API 60 (FMUL)	API 63 (FINT)	API 66 (FDEG)

■ Binary floating point number representation

The NC series MLC represents floating point numbers with 32 bits by adopting the IEEE 754 standard. The format is as follows.



Valid range of values: $(-1)^S$ x 2^{E-B} x 1.M (B = 127)

Range of values expressed by 32-bit floating point: ±2^{-126} to ±2^{+128} or ±1.1755×10^{-38} to ±3.4028×10^{+38}.

Example 1: express 23 as a 32-bit floating point number.

Steps:

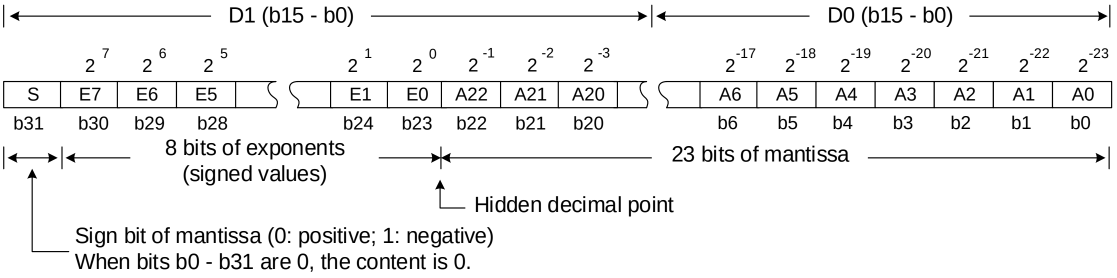
1. Convert 23 into a binary number: 23.0 = 10111.
2. Normalize the binary number: 10111 = 1.0111 × 24. 0111 is the mantissa and 4 is the exponent.
3. Obtain the stored value of the exponent: ∵E-B = 4 → E-127 = 4 ∴E = 131 = 10000011_2.
4. Combine the sign bit, exponent, and mantissa into a floating point number.

0 100**0001**1 011**1000**0000**0000**0000**0000**$_2$ = $41B80000_{16}$

Example 2: express -23.0 as a 32-bit floating point number.

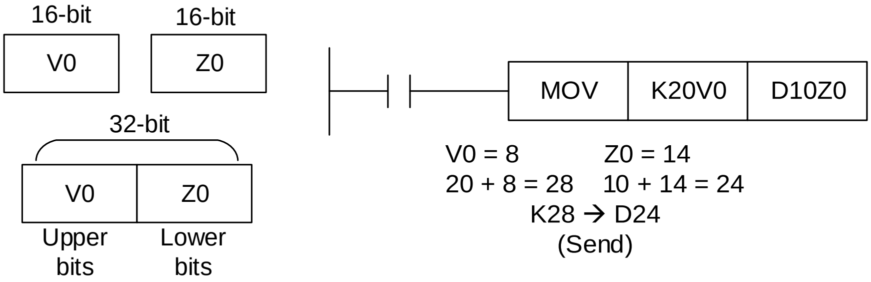
The conversion steps for the floating point format of -23.0 and 23.0 are the same but the sign bit is changed to 1.

MLC uses two consecutive registers to form a 32-bit floating point number. The following example uses the registers (D1, D0) to store a binary floating point number.



3.4 Modify operands with index registers V and Z

The index register is a 16-bit register. There are a total of 16 points of V and Z in the NC series models. Refer to the following for more details.



As shown in the above figure, contents in the operands vary according to the contents in V and Z. That is, the operands are modified by V and Z, which is called indexing. For example, V0 = 8 and K20V0 represent constant K28 (20 + 8). When the condition is met, constant K28 is sent to register D24.

Registers V and Z are 16-bit data registers that can be read and written. When a 32-bit length is required, specify register V. In this case, register V overwrites register Z and Z cannot be used anymore. Otherwise, errors will occur in register V. The (V, Z) combination used for 32-bit index register is (V0, Z0), (V1, Z1), (V2, Z2) …, and (V7, Z7).

Devices in the NC series that can be modified are P, KnX, KnY, KnM, KnA, T, C, and D. Specify V or Z to modify a 16-bit register and specify V to modify a 32-bit register.

3.5 Application instruction index

The following table lists the application instruction indexes in alphabetical order:

Type	API	Instruction code		Function
		16-bit	32-bit	
A	13	ADD	DADD	BIN addition
	29	ANS	-	Alarm output
	30	ANR	-	Alarm reset
	34	ALT	-	On / Off alternation
	45	AND=	DAND=	$S_1 = S_2$
	46	AND>	DAND>	$S_1 > S_2$
	47	AND<	DAND<	$S_1 < S_2$
	48	AND<>	DAND<>	$S_1 \neq S_2$
	49	AND<=	DAND<=	$S_1 \leqq S_2$
	50	AND>=	DAND>=	$S_1 \geqq S_2$
B	11	BCD	DBCD	Convert BIN data to BCD data
	12	BIN	DBIN	Covert BCD data to BIN data
	28	BON	DBON	Bit state monitoring
C	00	CJ	-	Conditional jump
	01	CALL	-	Call subroutine
	10	CML	DCML	Invert and transfer
	37	CNT	DCNT	Counter
D	05	DI	-	Disable interruption
	16	DIV	DDIV	BIN division
	18	DEC	DDEC	Minus one (BIN)
	26	DECO	-	Decoder
E	04	EI	-	Enable interruption
	27	ENCO	-	Encoder
F	06	FEND	-	End of main program
	07	FOR	-	Start of nested loop
	58	-	FADD	Binary floating-point number addition
	59	-	FSUB	Binary floating-point number subtraction
	60	-	FMUL	Binary floating-point number multiplication
	61	-	FDIV	Binary floating-point number division
	62	-	FCMP	Compare binary floating-point numbers
	63	-	FINT	Convert binary floating-point number to BIN integer (remove decimal)
	64	-	FDOT	Convert BIN integer to binary floating-point number
	65	-	FRAD	Convert degrees to radians
	66	-	FDEG	Convert radians to degrees
	67	-	FMOV	Move data
H	32	-	DHSCS	Compare setting (high-speed counter)
	33	-	DHSCR	Comparison reset (high-speed counter)
I	03	IRET	-	Return from interruption
	17	INC	DINC	Plus one (BIN)

Type	API	Instruction code		Function
		16-bit	32-bit	
L	39	LD=	DLD=	$S_1 = S_2$
	40	LD>	DLD>	$S_1 > S_2$
	41	LD<	DLD<	$S_1 < S_2$
	42	LD<>	DLD<>	$S_1 \neq S_2$
	43	LD<=	DLD<=	$S_1 \leqq S_2$
	44	LD>=	DLD>=	$S_1 \geqq S_2$
M	09	MOV	DMOV	Move data
	15	MUL	DMUL	BIN multiplication
N	08	NEXT	-	End of nested loop
	22	NEG	DNEG	Take the negative number (Two's complement)
O	51	OR=	DOR=	$S_1 = S_2$
	52	OR>	DOR>	$S_1 > S_2$
	53	OR<	DOR<	$S_1 < S_2$
	54	OR<>	DOR<>	$S_1 \neq S_2$
	55	OR<=	DOR<=	$S_1 \leqq S_2$
	56	OR>=	DOR>=	$S_1 \geqq S_2$
P	35	PLS	-	Upper differential output
	38	PLF	-	Lower differential output
R	23	ROR	DROR	Rotate right
	24	ROL	DROL	Rotate left
	31	REF	-	I/O refresh
	69	RDTL	-	Flag for reading the torque limit
S	02	SRET	-	End of subroutine
	14	SUB	DSUB	BIN subtraction
T	36	TMR	-	Timer
V	57	VRT	DVRT	Logical switch table
W	19	WAND	DWAND	AND operation
	20	WOR	DWOR	OR operation
	21	WXOR	DWXOR	XOR operation
	68	WRTL	-	Write the servo torque limit
Z	25	ZRST	-	Zone reset

(This page is intentionally left blank.)

MLC Application Instructions Description 4

This chapter provides the detailed function and definition of each API for the MLC.

4.1 Loop control instructions

■ API-00 CJ: Conditional jump

API			CJ		S				Conditional jump		NC series				
00	-														
	Bit device				Word device										
	X	Y	M	A	K	F	KnX	KnY	KnM	KnA	T	C	D	V	Z
S															
D															

16-bit instruction: CJ continuous execution type (2 steps).
32-bit instruction: none.
Flag: none.
Notes on the use of operands: operand S can assign indicator P.
V and Z registers can specify the number of P.
The S operand of the NC series can specify P0 - P255.

Instruction description:

S: the target to jump of the conditional jump instruction.

Use the CJ instruction when only part of the program needs to be executed or for two-way output to save the scanning time. Different CJ instructions can assign the same indicator P; however, CJ and CALL instructions cannot simultaneously assign the same indicator P or an error occurs.

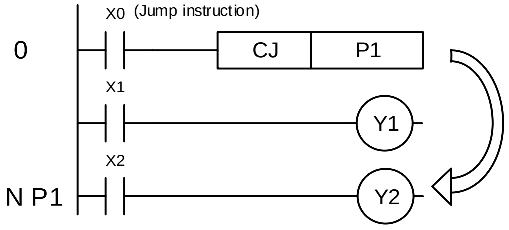
The device actions when the jump instruction is executed:

1. All of the instructions you send are executed.
2. Devices Y, M, and A remain their states before and after the jump instruction is executed.
3. The currently running 10 ms and 100 ms timer continue to time.
4. The currently running C78 and C79 high-speed counters continue to count and the output point operates as usual.
5. The general type instructions are not executed.
6. The instructions in execution, API-53 DHSCS and API-54 DHSCR, are continued.

Program example 1:

When X0 is On, the program automatically jumps from address 0 to N (the assigned label P1) to continue to run. The program between address 0 and address N is not executed.

When X0 is Off, the program continues to run from address 0 and what follows, and the CJ instruction is not executed.



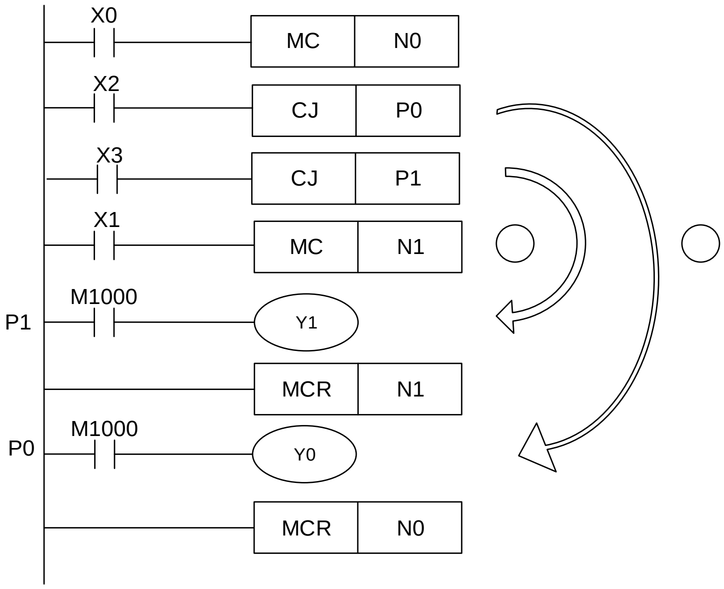
Program example 2:

You can use a CJ instruction between MC and MCR instructions in the following 5 conditions.

1. Out of MC - MCR instructions.
2. From outside of the MC to within the MC instruction, as shown in P1 of the following figure.
3. From the N level of the MC to another MC of the same level.
4. From within the MC to outside of the MCR.
5. Jumping from one set of MC - MCR to another set of MC - MCR.

Execution procedure:

When used between MC and MCR instructions, the CJ instruction can only be applied to outside of the MC - MCR loop or within the same N layer of the MC - MCR loop. Jumping from one MC - MCR to another MC - MCR leads to program errors. In other words, only Items 1 and 3 described above are allowed, whereas others can cause errors.



Example 3:

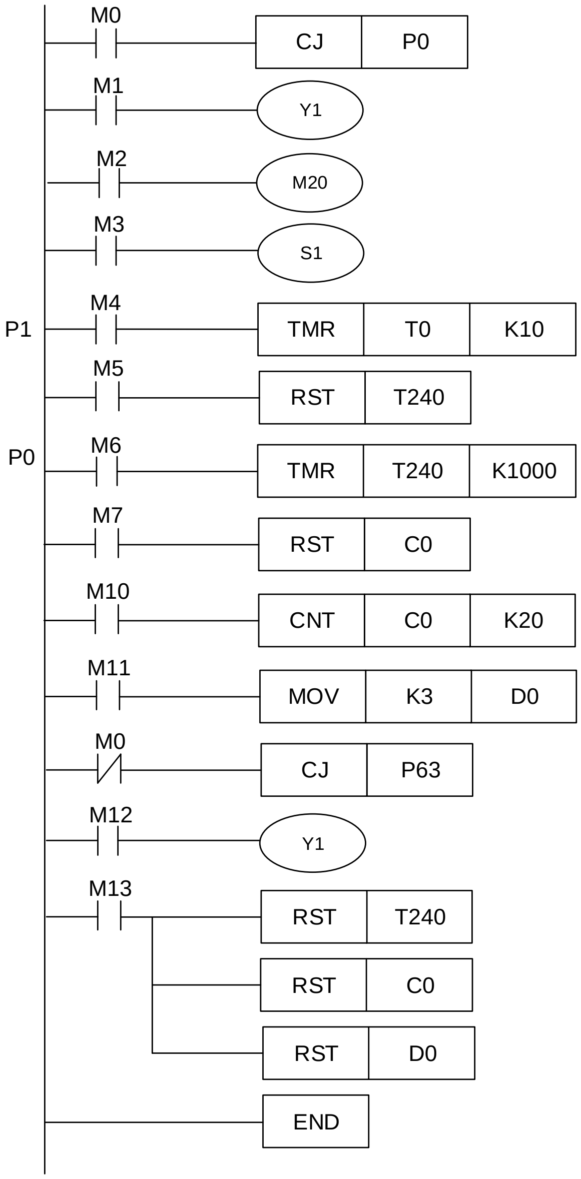
Y1 is a two-way output point. When M0 is Off, Y1 is controlled by M1; when M0 is On, Y1 is controlled by M12.

Stage change of each device in the following ladder diagram:

Device	Contact state before CJ is executed	Contact state during execution of CJ	Output coil state during execution of CJ
Y, M, A	M1, M2, and M3 are Off	M1, M2, and M3 go from Off to On	Y1[*1], M20, and S1 are Off
	M1, M2, and M3 are On	M1, M2, and M3 go from On to Off	Y1[*1], M20, and S1 are On
10 ms and 100 ms timers	M4 is Off	M4 goes form Off to On	The timer T0 does not count.
	M4 is On	M4 goes from On to Off	The timer T0 continues to time and remains. M0 goes from On to Off and the timer times until T0 goes to On.
C0 - C77	M7 and M10 are off	M10 On / Off triggered	The counter C0 does not count.
	M7 is Off; M10 On / Off triggered	M10 On / Off triggered	Counter C0 stops counting and holds. When M0 goes off, C0 resumes counting.
C78 and C79	When the high speed counters (C78 and C79) are activated and a CJ instruction is executed, they continue to count and the output points continues to operate.		

Device	Contact state before CJ is executed	Contact state during execution of CJ	Output coil state during execution of CJ
Application instruction	M11 is Off	M11 goes from Off to On	Application instructions are not executed
	M11 is On	M11 goes from On to Off	The skipped application instructions are not executed, but API-53 DHSCS and API-54 DHSCR continue to be executed.

Note:
*1: Y1 is a two-way output. When M0 is Off, Y1 is controlled by M1; when M0 is On, Y1 is controlled by M12.



■ API-01 CALL: Call subroutine

API		CALL	S				Call subroutine		NC series					
01	-													
	Bit device			Word device										
	X	Y	M	A	K	F	KnX	KnY	KnM	KnA	T	C	D	V	Z

16-bit instruction: CALL continuous execution type (2 steps)
32-bit instruction: none.
Flag: none.
Notes on the use of operands:
Operand S can assign indicator P.
V and Z registers can specify the number of P.
The S operand of the NC series can specify P0 - P255.

Instruction description:

S: the indicator for calling the subroutine.

Place the subroutine specified by S after the FEND instruction. When the number of P is being executed by the CALL instruction, avoid specifying the same number for the CJ instruction. If you use the CALL instruction solely, you can call the subroutine of the same indicator number for unlimited times. If you use the CALL instruction in the subroutine to call another subroutine, the maximum is 5 layers including the the instruction itself (the subroutine in the 6th layer is not executed).

■ API-02 SRET: End of subroutine

API		SRET	-				End of the subroutine		NC series					
02	-													
	Bit device			Word device										
	X	Y	M	A	K	F	KnX	KnY	KnM	KnA	T	C	D	V	Z

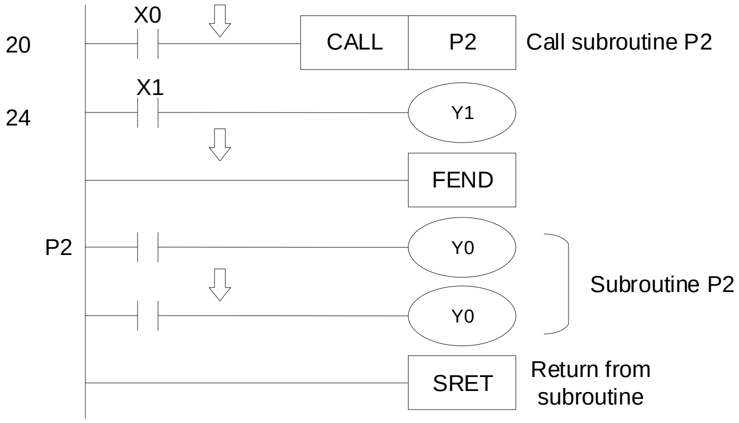
16-bit instruction: SRET continuous execution type (1 step).
32-bit instruction: none.
Flag: none.
Notes on the use of operands: no operand.
Contact for activating the instruction is not required.

Instruction description:

The SRET instruction indicates the end of the subroutine. After the subroutine is completely executed, the program returns to the main program from SRET and starts from the next instruction following the CALL instruction.

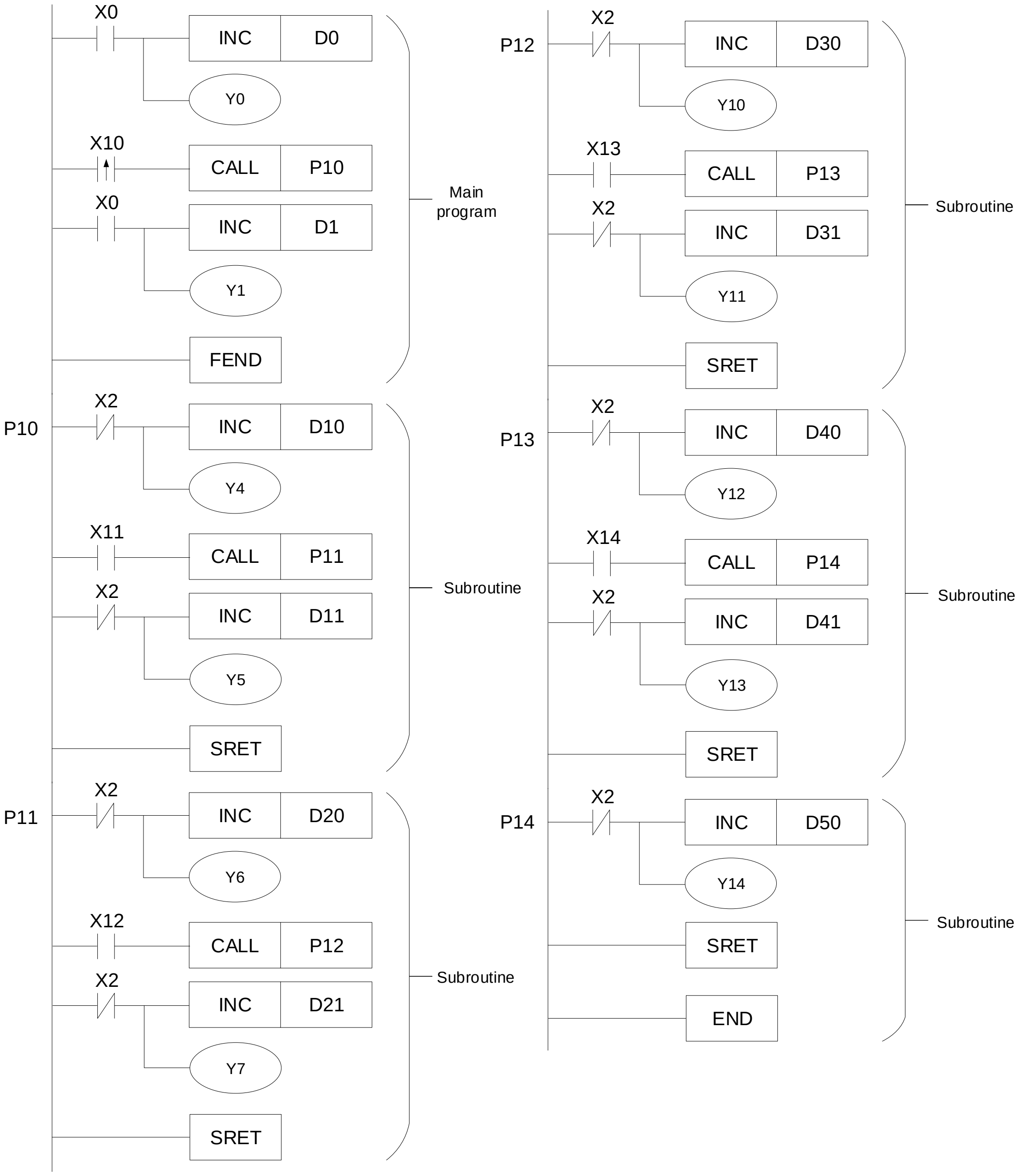
Program example 1:

When X0 is On, execute the CALL instruction and jump to P2 to execute the specified subroutine. When SRET is executed, return to address 24 and then continues to execute what follows.



Program example 2:

1. When X10 is rising-edge triggered from Off to On to execute the CALL P10 instruction, the program jumps to subroutine assigned by P10.
2. When X11 is On, CALL P11 is executed and the program jumps to the subroutine assigned by P11.
3. When X12 is On, CALL P12 is executed and program jumps to the subroutine assigned by P12.
4. When X13 is On, CALL P13 is executed and program jumps to the subroutine assigned by P13.
5. When X14 is On, CALL P14 is executed and program jumps to the subroutine assigned by P14. When SRET is executed, return to the previous subroutine P※ and continue the execution.
6. After SRET is executed in subroutine P10, return to the main program.



API-03 IRET: Return from interruption

API			IRET		-				Return from interrupt		NC series				
03	-														
	Bit device				Word device										
	X	Y	M	A	K	F	KnX	KnY	KnM	KnA	T	C	D	V	Z

16-bit instruction: IRET continuous execution type (1 step).
32-bit instruction: none.
Flag: none.
Notes on the use of operands: no operand.
Contact for activating the instruction is not required.

Instruction description:

IRET interrupts the current program and inserts a subroutine. After the interruption is completed, return to the main program from IRET and continue executing the next instruction where the main program was interrupted.

API-04 EI: Enable interruption

API			EI		-				Enable interruption		NC series				
04	-														
	Bit device				Word device										
	X	Y	M	A	K	F	KnX	KnY	KnM	KnA	T	C	D	V	Z

16-bit instruction: EI continuous execution type (1 step).
32-bit instruction: none.
Flag: M2880 - M2889 and M2896 - M2927. Refer to the supplementary note of API-05 DI.
Notes on the use of operands: no operand.
Contact for activating the instruction is not required.
Pulse width of the interruption signal must be over 200 us.
For ranges of the I number for each model, refer to API-05 DI supplementary notes.

API-05 DI: Disable interruption

API			DI		-				Disable interruption		NC series				
05	-														
	Bit device				Word device										
	X	Y	M	A	K	F	KnX	KnY	KnM	KnA	T	C	D	V	Z

16-bit instruction: DI continuous execution type (1 step).
32-bit instruction: none.
Flag: none.
Notes on the use of operands: no operand.
Contact for activating the instruction is not required.

Instruction description:

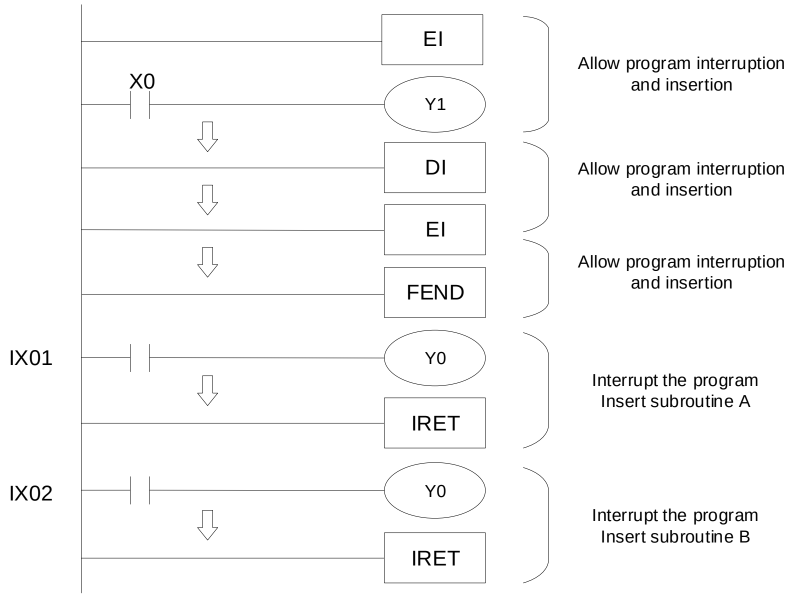
1. Using an EI instruction in the program means the subroutine interruption is allowed, such as external interruption and high-speed counter interruption.
2. A subroutine interruption is allowed in the program between EI and DI instructions. The DI instruction is not required when the program has no “disable interruption” section.
3. For the NC series models, if the interruption disabling special relays M2880 - M2889 and M2896 - M2927 are not activated, the corresponding interruption requests are not executed even in the section allowing interruptions.
4. The indicator (I) used for interruptions must be placed after the FEND instruction.
5. While an program interruption is running, other interruptions are not allowed.
6. When there is more than one interruption, the priority goes to the one which is being executed. When there are several interruptions, the priority goes to the interruption with a

smaller indicator number.

7. The interruption request between DI and EI instructions cannot be immediately executed. This interruption is reserved and will be executed in a section allowing interruption.
8. When you use an interruption indicator, do not repeatedly use the high-speed counter driven with the same X input contact.
9. To immediately change the I/O action during an interruption, place a REF instruction in the program to update the I/O state.

Program example 1:

During the operation of MLC, when the program scans the section from instructions EI to DI and X1 = On or X2 = On, it interrupts the program and insert subroutines A or B. After the IRET instruction is executed, the execution returns to the main program and continues.


EI
X0
Y1
Allow program interruption and insertion
DI
Allow program interruption and insertion
EI
Allow program interruption and insertion
FEND
IX01
Y0
IRET
Interrupt the program Insert subroutine A
IX02
Y0
IRET
Interrupt the program Insert subroutine B

Supplementary note:

Indicator I for interruption in NC series models:

1. OnBoard interruption: 8 points; (IX00, X0), (IX01, X1), (IX02, X2), (IX03, X3), (IX04, X4), (IX05, X5), (IX06, X6) and (IX07, X7).
2. High speed counter interruption: 2 points; IC00 and IC01. (Work with the API-32 DHSCS instruction to generate interruption signals.)
3. Remote I/O interruption: 32 points; IR00 - IR31. IR00 - IR31 correspond to the Remote input points X256 - X287 respectively. The IR (IR00 to IR31) interruptions respectively correspond to the 32 X input points of the Remote I/O Card 0.
4. In the MLC system, the interruptions are executed in FIFO (first in, first out) order.

The flag of “prohibiting interruption indicator insertion” in the NC series models:

Special M	Description
M2880	When M2880 is set to On, IX00 input point interruption is enabled. Input point: controller main board I/O: X0.
M2881	When M2881 is set to On, IX01 input point interruption is enabled. Input point: controller main board I/O: X1.
M2882	When M2882 is set to On, IX02 input point interruption is enabled. Input point: controller main board I/O: X2.
M2883	When M2883 is set to On, IX03 input point interruption is enabled. Input point: controller main board I/O: X3.
M2884	When M2884 is set to On, IX04 input point interruption is enabled. Input point: controller main board I/O: X4.
M2885	When M2885 is set to On, IX05 input point interruption is enabled. Input point: controller main board I/O: X5.
M2886	When M2886 is set to On, IX06 input point interruption is enabled. Input point: controller main board I/O: X6.
M2887	When M2887 is set to On, IX07 input point interruption is enabled. Input point: controller main board I/O: X7.
M2888	When M2888 is set to On, IC00 input point interruption is enabled. Input point: hardware high speed input counter 0.
M2889	When M2889 is set to On, IC01 input point interruption is enabled. Input point: hardware high speed input counter 1.
M2896	When M2896 is set to On, IR00 input interruption is enabled. Input point: X0 of Remote I/O card 0.
M2897	When M2897 is set to On, IR01 input interruption is enabled. Input point: X1 of Remote I/O card 0.
M2898	When M2898 is set to On, IR02 input interruption is enabled. Input point: X2 of Remote I/O card 0.
M2899	When M2899 is set to On, IR03 input interruption is enabled. Input point: X3 of Remote I/O card 0.
M2900	When M2900 is set to On, IR04 input interruption is enabled. Input point: X4 of Remote I/O card 0.
M2901	When M2901 is set to On, IR05 input interruption is enabled. Input point: X5 of Remote I/O card 0.
M2902	When M2902 is set to On, IR06 input interruption is enabled. Input point: X6 of Remote I/O card 0.
M2903	When M2903 is set to On, IR07 input interruption is enabled. Input point: X7 of Remote I/O card 0.
M2904	When M2904 is set to On, IR08 input interruption is enabled. Input point: X8 of Remote I/O card 0.
M2905	When M2905 is set to On, IR09 input interruption is enabled. Input point: X9 of Remote I/O card 0.
M2906	When M2906 is set to On, IR10 input interruption is enabled. Input point: X10 of Remote I/O card 0.
M2907	When M2907 is set to On, IR11 input interruption is enabled. Input point: X11 of Remote I/O card 0.
M2908	When M2908 is set to On, IR12 input interruption is enabled. Input point: X12 of Remote I/O card 0.
M2909	When M2909 is set to On, IR13 input interruption is enabled. Input point: X13 of Remote I/O card 0.
M2910	When M2910 is set to On, IR14 input interruption is enabled. Input point: X14 of Remote I/O card 0.

Special M	Description
M2911	When M2911 is set to On, IR15 input interruption is enabled. Input point: X15 of Remote I/O card 0.
M2912	When M2912 is set to On, IR16 input interruption is enabled. Input point: X16 of Remote I/O card 0.
M2913	When M2913 is set to On, IR17 input interruption is enabled. Input point: X17 of Remote I/O card 0.
M2914	When M2914 is set to On, IR18 input interruption is enabled. Input point: X18 of Remote I/O card 0.
M2915	When M2915 is set to On, IR19 input interruption is enabled. Input point: X19 of Remote I/O card 0.
M2916	When M2916 is set to On, IR20 input interruption is enabled. Input point: X20 of Remote I/O card 0.
M2917	When M2917 is set to On, IR21 input interruption is enabled. Input point: X21 of Remote I/O card 0.
M2918	When M2918 is set to On, IR22 input interruption is enabled. Input point: X22 of Remote I/O card 0.
M2919	When M2919 is set to On, IR23 input interruption is enabled. Input point: X23 of Remote I/O card 0.
M2920	When M2920 is set to On, IR24 input interruption is enabled. Input point: X24 of Remote I/O card 0.
M2921	When M2921 is set to On, IR25 input interruption is enabled. Input point: X25 of Remote I/O card 0.
M2922	When M2922 is set to On, IR26 input interruption is enabled. Input point: X26 of Remote I/O card 0.
M2923	When M2923 is set to On, IR27 input interruption is enabled. Input point: X27 of Remote I/O card 0.
M2924	When M2924 is set to On, IR28 input interruption is enabled. Input point: X28 of Remote I/O card 0.
M2925	When M2925 is set to On, IR29 input interruption is enabled. Input point: X29 of Remote I/O card 0.
M2926	When M2926 is set to On, IR30 input interruption is enabled. Input point: X30 of Remote I/O card 0.
M2927	When M2927 is set to On, IR31 input interruption is enabled. Input point: X31 of Remote I/O card 0.

■ API-06 FEND: End of main program

API			FEND		-				End of main program		NC series				
06	-														
	Bit device				Word device										
	X	Y	M	A	K	F	KnX	KnY	KnM	KnA	T	C	D	V	Z

16-bit instruction: FEND continuous execution type (1 step)
32-bit instruction: none.
Flag: none.
Notes on the use of operands: no operand.
Contact for activating the instruction is not required.

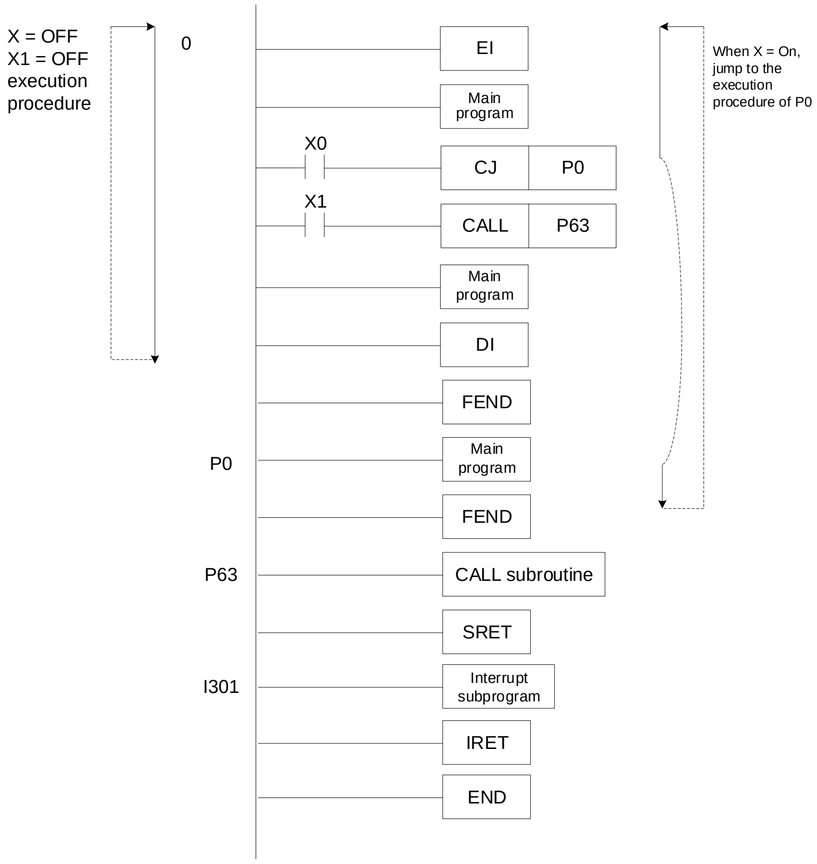
Instruction description:

This instruction indicates the end of the main program. When the MLC executes this instruction, FEND functions the same as the END instruction. Note that the CALL instruction must be put following the FEND instruction, and then the SRET instruction must be put at the end of the subroutine. To interrupt a program, put the interrupt instruction after FEND and then put IRET at the service program being executed. When using multiple FEND instructions, place the subroutine and the interruption service program between the last FEND and END instructions.

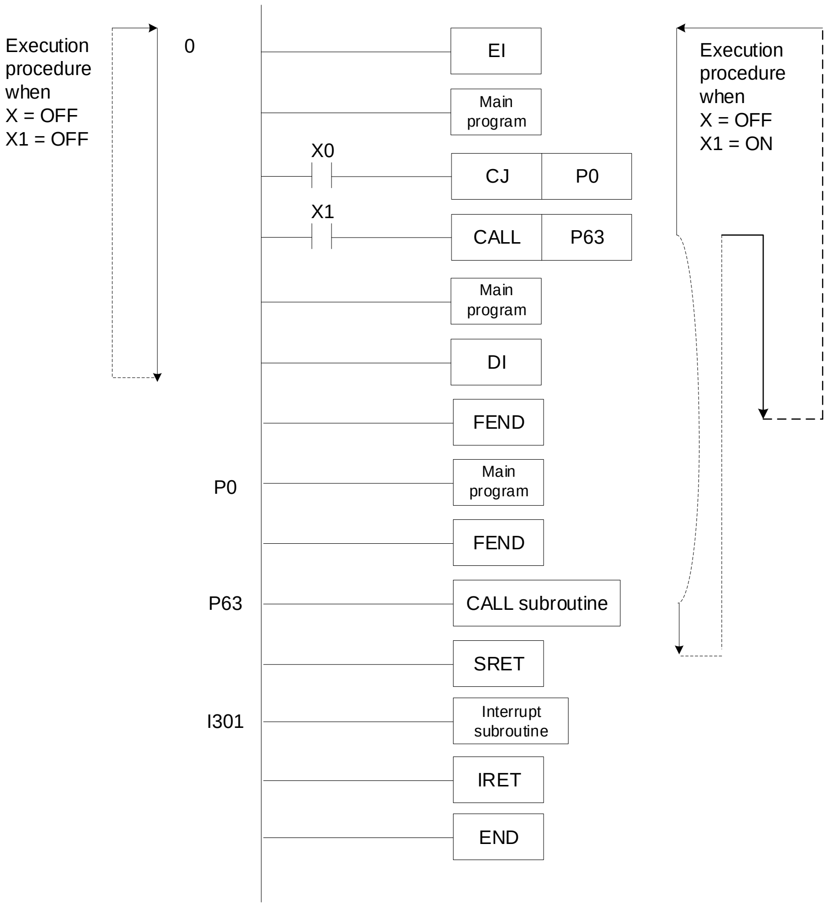
A program error occurs in the following conditions:

1. After the CALL instruction is executed, FEND is executed before SRET.
2. After the FOR instruction is executed, FEND is executed before NEXT.

Execution procedure of CJ instruction:



Execution procedure of CALL instruction:



API-07 FOR: Start of nested loop

API			FOR		S				Start of nested loop		NC series				
07	-														
	Bit device				Word device										
	X	Y	M	A	K	F	KnX	KnY	KnM	KnA	T	C	D	V	Z

16-bit instruction: FOR continuous execution type (3 steps).
32-bit instruction: none.
Flag: none.
Notes on the use of operands: contact for activating the instruction is not required.
Refer to the specification of each model for the valid range of each device.

Instruction description:

S: the number of time the loop is executed.

API-08 NEXT: End of nested loop

API			NEXT		-				End of nested loop		NC series				
08	-														
	Bit device				Word device										
	X	Y	M	A	K	F	KnX	KnY	KnM	KnA	T	C	D	V	Z

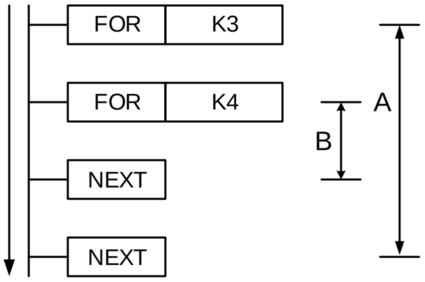
16-bit instruction: NEXT continuous execution type (1 step).
32-bit instruction: none.
Flag: none.
Notes on the use of operands: no operand.
Contact for activating the instruction is not required.

Instruction description:

1. The FOR...NEXT loop is executed for the number of times (N) assigned by FOR and the loop is exited. Next, the execution continues following the NEXT instruction.
2. The range of the repeating times: N = K1 - K32,767. N is regarded as K1 when N ≦ 1.
3. To skip the FOR...NEXT loop, use the CJ instruction to exit the loop.
4. A program error occurs in the following conditions:
   a. The NEXT instruction is placed before FOR.
   b. A FOR instruction has no corresponding NEXT instruction.
   c. A NEXT instruction is placed after FEND or END instructions.
   d. The number of FOR instructions is different from the number of NEXT instructions.
5. The FOR...NEXT loops can nest for up to 5 layers. Be aware that the more the layers, the more time required for MLC scanning.

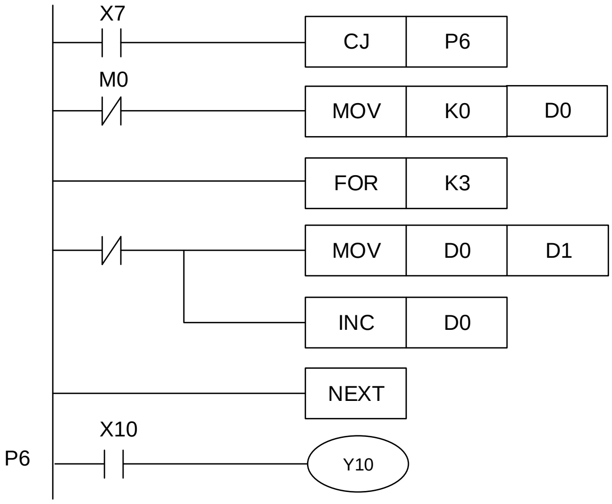
Program example 1:

The program A repeats three times, and the execution goes to the program after the NEXT instruction. Each time the program A is executed, the program B is executed four 4 times. Therefore, the program B is executed 12 times (3 x 4 = 12) in total.



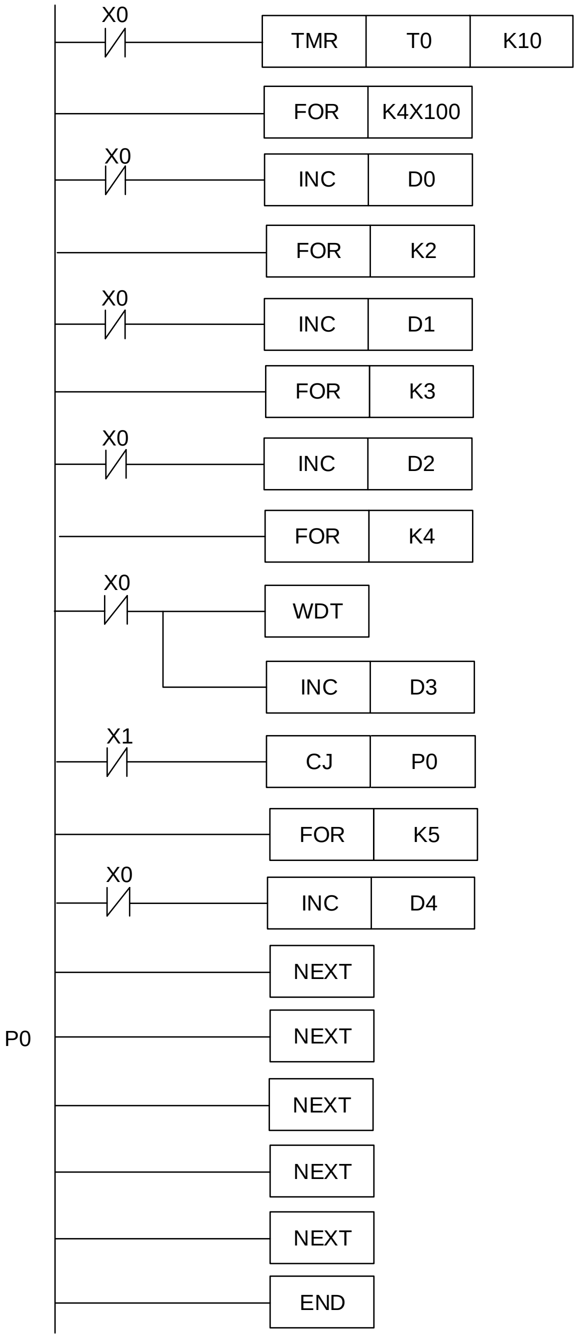
Program example 2:

When X7 is Off, the MLC executes the program between FOR and NEXT instructions. When X7 is On, the CJ instruction is executed and the execution jumps to P6, skipping the program between FOR and NEXT.



Program example 3:

To skip the FOR...NEXT loop, use a CJ instruction. When X1 is On, you can use the CJ instruction to skip the most inner FOR...NEXT layer and have the execution jumped to P0.



4.2 Transmission and comparison instructions

■ API-09 MOV: Move data

API		MOV	S, D	Move data	NC series
09	D				
67	F				

	Bit device				Word device										
	X	Y	M	A	K	F	KnX	KnY	KnM	KnA	T	C	D	V	Z
S					*	*	*	*	*	*	*	*	*	*	*
D								*	*	*	*	*	*	*	*

16-bit instruction: MOV continuous execution type (4 steps)
32-bit instruction: DMOV continuous execution type (6 steps)
32-bit instruction: FMOV continuous execution type (6 steps)
Flag: none.
Notes on the use of operands: if operands S and D are used in register Z, only 16-bit instruction is applicable.
Refer to Chapter 1 for the range of each device.

Instruction description:

S: data source; D: destination for the data to be moved.

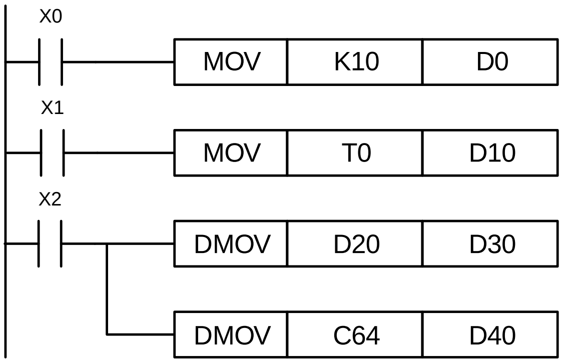
When the MOV instruction is executed, the data contained in S is directly moved to D. If MOV is not executed, the contents in D remains unchanged.
To move the 32-bit operation result (such as the application instruction FMUL) and 32-bit current value of high speed counter, DMOV is required. To move the floating-point number device, use the FMOV instruction.

Program example:

To move the 16-bit data, use the MOV instruction.

a. When X0 is Off, the data in D10 remains unchanged. If X0 is On, the value of K10 is moved to register D0.
b. When X1 is Off, the data in D10 remains unchanged. If X1 is On, the current value of T0 is moved to register D10.

To move the 32-bit data, use a DMOV instruction.
When X2 is Off, the data in (D31, D30) and (D41, D40) remain unchanged. If X2 is On, the current values in (D21, D20) are moved to (D31, D30) registers. Meanwhile, the current value of C64 is moved to data registers (D41, D40).



■ API-10 CML: Invert and transfer

API		CML	S, D	Invert and transfer	NC series
10	D				

	Bit device				Word device										
	X	Y	M	A	K	F	KnX	KnY	KnM	KnA	T	C	D	V	Z
S					*								*		
D													*		

16-bit instruction: CML continuous execution type (4 steps).
32-bit instruction: DCML continuous execution type (5 steps).
Flag: none.
Notes on the use of operands: if operands S and D are used in register Z, only 16-bit instruction is applicable.
Refer to Chapter 1 for the range of each device.

Instruction description:

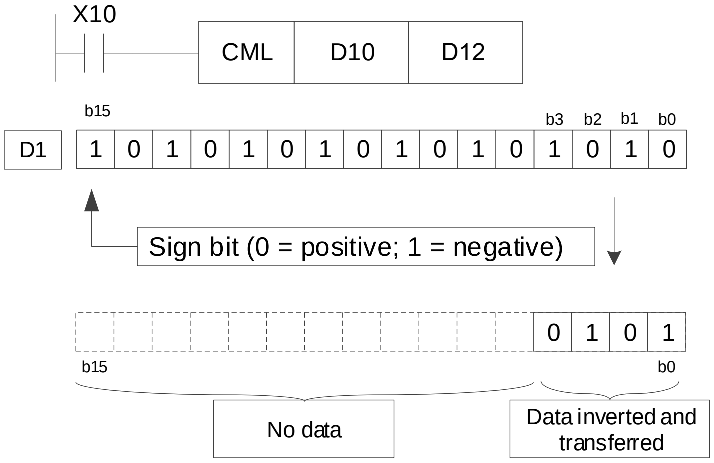
S: data source. D: destination device for the data to transfer to.

Invert the phase of all contents in S (0→1, 1→0) and send the result to device D. If the content is a K constant, this K constant is automatically converted to a BIN value.

Program example 1:

You can use this instruction when inverted phase output is required.

When X10 is On, contents of b0 - b3 are inverted and sent to D12.



■ API-11 BCD: Convert BIN data to BCD data

API		BCD	S, D	Convert BIN data to BCD data	NC series
11	D				

	Bit device				Word device										
	X	Y	M	A	K	F	KnX	KnY	KnM	KnA	T	C	D	V	Z
S													*		
D													*		

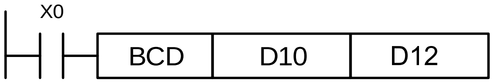
16-bit instruction: BCD continuous execution type (4 steps).
32-bit instruction: DBCD continuous execution type (4 steps).
Flag: M2828 (computing error).
Notes on the use of operands: if operands S and D are used in register Z, only 16-bit instruction is applicable.
Refer to Chapter 1 for the range of each device.

Instruction description:

Converts the content of data source S (BIN value) to BCD, and saves the result in D. When the result of BCD conversion exceeds K0 - K9,999, and M2828 is On. When the result of DBCD conversion exceeds K0 - K99,999,999, M2828 is On. MLC arithmetic operations and the execution of INC and DEC instructions are performed in BIN format. Thus, use the BCD instruction to convert BIN data to BCD data if displaying data in decimal format is required.

Program example:

When X0 is On, BIN data in D10 is converted to BCD data, and the units digit of the result is saved in D12.



■ API-12 BIN: Convert BCD data to BIN data

API		BIN	S, D	Covert BCD data to BIN data	NC series
12	D				

	Bit device				Word device										
	X	Y	M	A	K	F	KnX	KnY	KnM	KnA	T	C	D	V	Z
S													∗		
D													∗		

16-bit instruction: BIN continuous execution type (4 steps).
32-bit instruction: DBIN continuous execution type (4 steps).
Flag: none.
Notes on the use of operands: if operands S and D are used in register Z, only 16-bit instruction is applicable.
Refer to Chapter 1 for the range of each device.

Instruction description:

S: data source; D: conversion result.

Converts the contents of data source S (BCD: 0 - 9,999) to BIN data and saves the result in D.

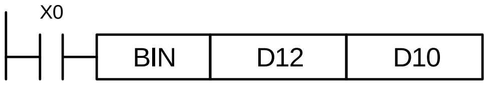
Valid value range of contents in S:

BCD: 0 - 9,999; DBCD: 0 - 99,999,999.

This instruction is not required for constant K and H as they are automatically converted into BIN format.

Program example:

When X0 is On, the BCD data in D12 is converted to BIN data, and the result is saved in D10.



4.3 Arithmetic and logic operation instructions

■ API-13 ADD: BIN addition

API		ADD	S_1, S_2, D	BIN addition	NC series
13	D				

	Bit device				Word device										
	X	Y	M	A	K	F	KnX	KnY	KnM	KnA	T	C	D	V	Z
S_1					∗						∗	∗	∗		
S_2					∗						∗	∗	∗		
D											∗	∗	∗		

16-bit instruction: ADD continuous execution type (6 steps).
32-bit instruction: DADD continuous execution type (8 steps).
Flag: M2824 (zero flag), M2825 (borrow flag), and M2826 (carry flag). (Please refer to the supplementary notes.)
Notes on the use of operands: if operands S_1, S_2 and D are used in register Z, only 16-bit instruction is applicable.
Refer to Chapter 1 for the range of each device.

Instruction description:

S_1: summand; S_2: addend; D: sum.

Adds data sources S_1 and S_2 in BIN format and saves the result in D. The highest bit of each data is the sign bit, which can be used for algebraic addition operations. 0 represents a positive sign and 1 represents a negative sign. For example: 3 + (-9) = -6.

Flag changes in BIN addition:

16-bit BIN addition:

1. If the addition result is 0, the zero flag M2824 is On.
2. If the addition result is less than -32,768, the borrow flag M2825 is On.
3. If the addition result is greater than than 32,767, the carry flag M2826 is On.

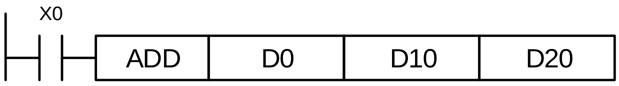
32-bit BIN addition:

1. If the addition result is 0, the zero flag M2824 is On.
2. If the addition result is less than -2,147,483,648, the borrow flag M2825 is On.
3. If the addition result is greater than than 2,147,483,647, the carry flag M2826 is On.

Program example 1:

16-bit BIN addition:

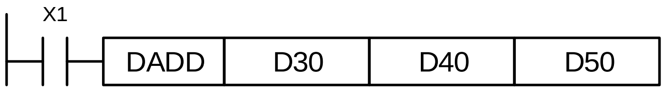
When X0 is On, add the summand D0 and addend D10 and save the result in D20.



Program example 2:

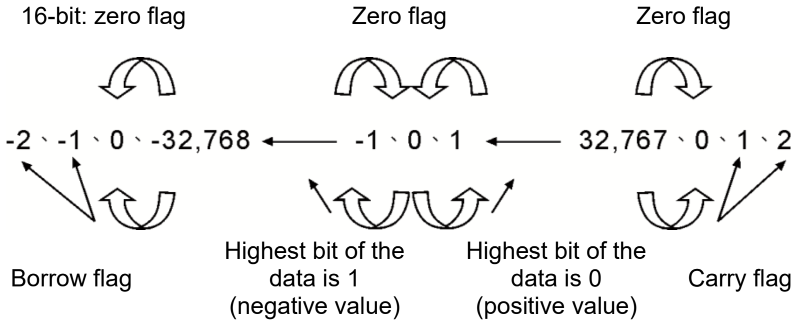
32-bit BIN addition:

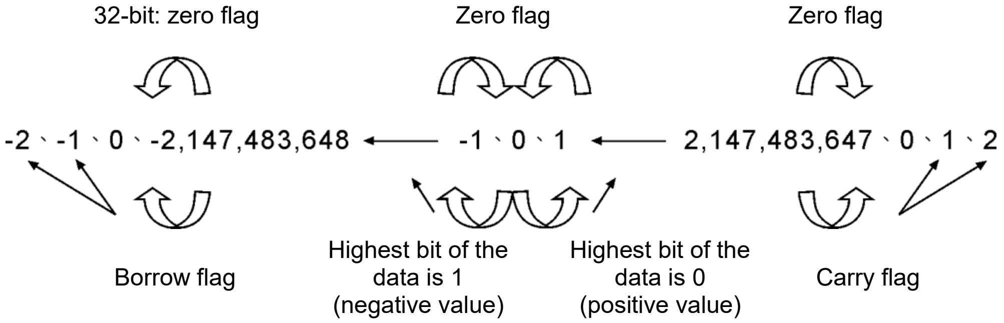
When X1 is On, add the content in (D31, D30)(summand) and the content in (D41, D40) (addend), and save the sum in (D51, D50). (D30, D40 and D50 are the lower 16-bit data, whereas D31, D41 and D51 are the higher 16-bit data.)



Supplementary note:

1. Flag actions and the positive/negative sign of the values:


16-bit: zero flag
Zero flag
Zero flag
-2、-1、0、-32,768
-1、0、1
32,767、0、1、2
Borrow flag
Highest bit of the data is 1 (negative value)
Highest bit of the data is 0 (positive value)
Carry flag



API-14 SUB: BIN subtraction

API			SUB		S_1, S_2, D				BIN subtraction		NC series				
14	D														
	Bit device				Word device										
	X	Y	M	A	K	F	KnX	KnY	KnM	KnA	T	C	D	V	Z
S1					＊						＊	＊	＊		
S2					＊						＊	＊	＊		
D											＊	＊	＊		

16-bit instruction: SUB continuous execution type (6 steps).
32-bit instruction: DSUB continuous execution type (8 steps).
Flag: M2824 (zero flag), M2825 (borrow flag), and M2826 (carry flag). Refer to API-13 ADD supplementary notes.
Notes on the use of operands: if operands S_1, S_2 and D are used in register Z, only 16-bit instruction is applicable.
Refer to Chapter 1 for the range of each device.

Instruction description:

S_1: minuend S_2: subtrahend D: difference

Subtracts the data sources S_1 and S_2 in BIN format and saves the result in D. The highest bit of each data is the sign bit, which can be used for algebraic addition operations. 0 represents a positive sign and 1 represents a negative sign.

Flag changes in BIN subtraction:

16-bit BIN subtraction:

1. If the subtraction result = 0, the zero flag M2824 is On.
2. If the subtraction result < -32,768, the borrow flag M2825 is On.
3. If the subtraction result > 32,767, the carry flag M2826 is On.

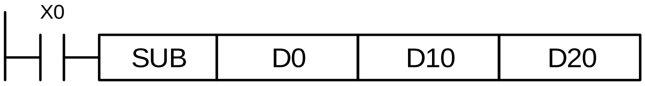
32-bit BIN subtraction:

1. If the subtraction result = 0, the zero flag M2824 is On.
2. If the subtraction result < -2,147,483,648, the borrow flag M2825 is On.
3. If the subtraction result > 2,147,483,647, the carry flag M2826 is On.

Refer to API-13 ADD supplementary notes for the flags and the positive/negative sign of the values.

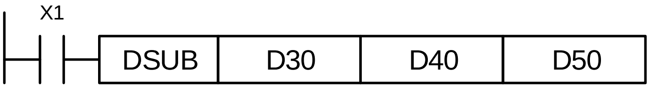
Program example 1:

16-bit BIN subtraction: when X0 is On, subtract the content of D10 from D0, and save the difference in D20.



Program example 2:

32-bit BIN addition: when X1 is On, subtract the content of (D41, D40) from (D31, D30), and save the difference in (D51, D50). (D30, D40 and D50 are the lower 16-bit data, whereas D31, D41 and D51 are the higher 16-bit data.)



■ API-15 MUL: BIN multiplication

API			MUL		S_1, S_2, D				BIN multiplication		NC series				
15	D														
	Bit device				Word device										
	X	Y	M	A	K	F	KnX	KnY	KnM	KnA	T	C	D	V	Z
S1					∗						∗	∗	∗		
S2					∗						∗	∗	∗		
D											∗	∗	∗		

16-bit instruction: MUL continuous execution type (6 steps).
32-bit instruction: DMUL continuous execution type (8 steps).
Flag: none.
Notes on the use of operands: if operands S_1, S_2 and D are used in register Z, only 16-bit instruction is applicable.
In 16-bit instruction, operand D takes consecutive 2 devices.
In 32-bit instruction, operand D takes consecutive 4 devices.
Refer to Chapter 1 for the range of each device.

Instruction description:

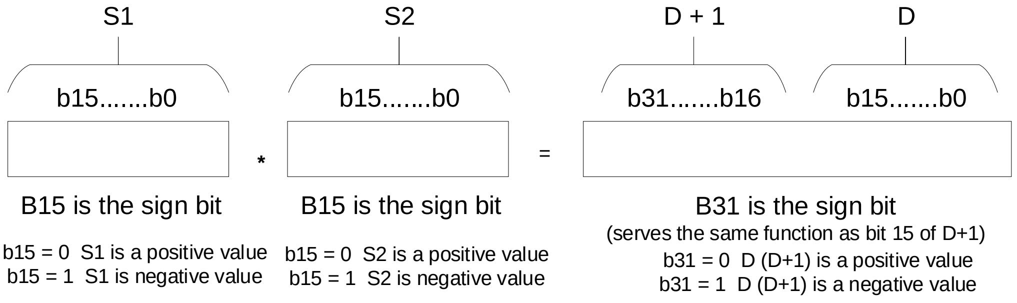
S_1: multiplicand; S_2: multiplier; D: product

Multiplies values in data source S_1 and S_2 in signed binary format and saves the product in D.

For 16-bit and 32-bit operations, pay attention to the positive/negative signs of S_1, S_2 and D.

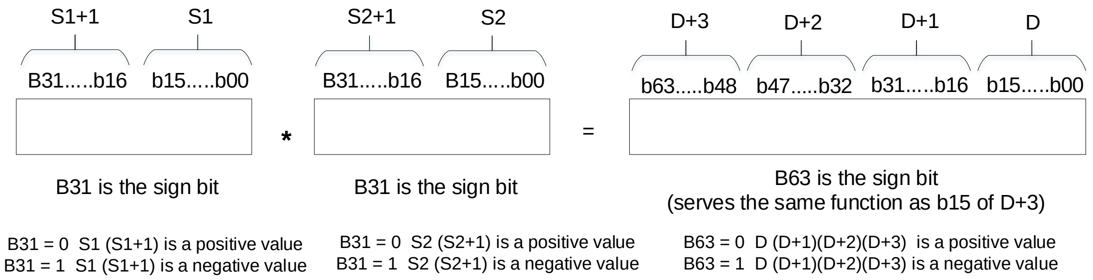
16-bit BIN multiplication:

If D serves as a bit device, you can assign K1 - K4 as 16-bit, occupying consecutive 2 sets of 16-bit devices.


S1
S2
D + 1
D
b15.......b0
b15.......b0
b31.......b16
b15.......b0
*
=
B15 is the sign bit
B15 is the sign bit
B31 is the sign bit
(serves the same function as bit 15 of D+1)
b15 = 0 S1 is a positive value
b15 = 1 S1 is negative value
b15 = 0 S2 is a positive value
b15 = 1 S2 is negative value
b31 = 0 D (D+1) is a positive value
b31 = 1 D (D+1) is a negative value

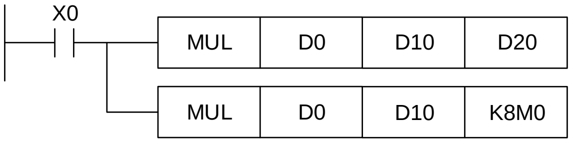
32-bit BIN multiplication:

If D serves as a bit device, you can assign K1 - K8 as 32-bit, and D stores the lower 32-bit data only.


S1+1
S1
B31.....b16
b15.....b00
*
S2+1
S2
B31.....b16
B15.....b00
=
D+3
D+2
D+1
D
b63.....b48
b47.....b32
b31.....b16
b15.....b00
B31 is the sign bit
B31 is the sign bit
B63 is the sign bit
(serves the same function as b15 of D+3)
B31 = 0 S1 (S1+1) is a positive value
B31 = 1 S1 (S1+1) is a negative value
B31 = 0 S2 (S2+1) is a positive value
B31 = 1 S2 (S2+1) is a negative value
B63 = 0 D (D+1)(D+2)(D+3) is a positive value
B63 = 1 D (D+1)(D+2)(D+3) is a negative value

Program example:

When X0 is On, the 16-bit D0 is multiplied by the 16-bit D10 to obtain a 32-bit product. The higher 16 bits are saved in D21 and the lower 16 bits are saved in D20. On / Off state of the most left bit indicates the positive / negative sign of the value.



■ API-16 DIV: BIN division

API			DIV		S_1, S_2, D				BIN division		NC series				
16	D														
	Bit device				Word device										
	X	Y	M	A	K	F	KnX	KnY	KnM	KnA	T	C	D	V	Z
S1					*						*	*	*		
S2					*						*	*	*		
D											*	*	*		

16-bit instruction: DIV continuous execution type (6 steps).
32-bit instruction: DDIV continuous execution type (8 steps).
Flag: M2828 computing error.
Notes on the use of operands: if operands S_1, S_2 and D are used in register Z, only 16-bit instruction is applicable.
In 16-bit instruction, operand D takes consecutive 2 devices.
In 32-bit instruction, operand D takes consecutive 4 devices.
Refer to Chapter 1 for the range of each device.

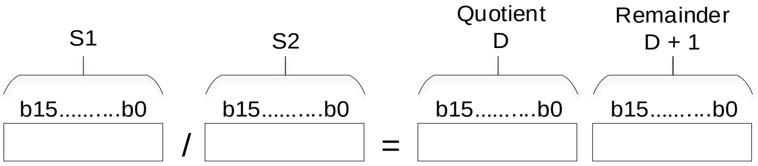
Instruction description:

S_1: dividend; S_2: divisor; D: quotient and remainder.

Divides data source S_1 by S_2 in signed binary format and saves the quotient and remainder in D. For 16-bit and 32-bit operations, pay attention to the positive / negative signs of S_1, S_2 and D. If the divisor is 0, this instruction is not executed. When M2828 is On, D1467 records the error code 0002 (Hex).

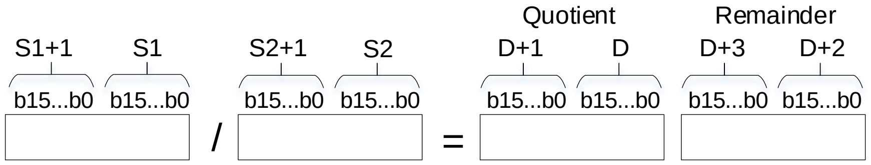
16-bit BIN division:

If D serves as a bit device, you can assign K1 - K4 as 16-bit, occupying consecutive 2 sets of 16-bit data and bringing forth the quotient and remainder.



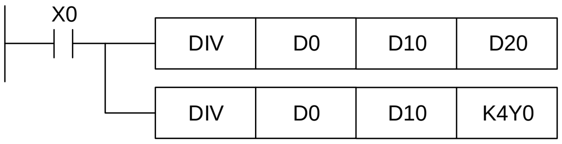
32-bit BIN division:

If D serves as a bit device, you can assign K1 - K8 as 32-bit, bringing forth the quotient with no remainder.



Program example:

When X0 is On, D0 is divided by D10, and the quotient is saved in D20 and the remainder is saved in D21. ON/OFF of the most left bit indicates the positive/negative sign of the result.



■ **API-17 INC: Plus one (BIN)**

API			INC		D				Plus one (BIN)		NC series				
17	D														
	Bit device				Word device										
	X	Y	M	A	K	F	KnX	KnY	KnM	KnA	T	C	D	V	Z
D											*	*	*		

16-bit instruction: INC continuous execution type (3 steps).
32-bit instruction: DINC continuous execution type (3 steps).
Flag: none.
Notes on the use of operands: if operand D is used in register V, only the 16-bit instruction is applicable.□

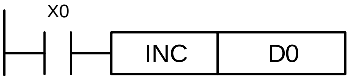
Instruction description:

D: destination device

When the INC instruction is executed, the value in the specified device D increases by 1 in each scanning cycle of the program. In 16-bit operation, 32,767 plus 1 is -32,768. In 32-bit operation, 2,147,483,647 plus 1 is -2,147,483,648. The operation result of this instruction does not affect flags M2824 - M2826.

Program example:

When X0 goes from Off to On, the value of D0 automatically adds 1.



■ API-18 DEC: Minus one (BIN)

API			DEC		D				Minus one (BIN)		NC series				
18	D														
	Bit device				Word device										
	X	Y	M	A	K	F	KnX	KnY	KnM	KnA	T	C	D	V	Z
D											∗	∗	∗		

16-bit instruction: DEC continuous execution type (3 steps).
32-bit instruction: DDEC continuous execution type (3 steps).
Flag: none.
Notes on the use of operands: if operand D is used in register V, only the 16-bit instruction is applicable.

Instruction description:

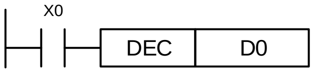
D: destination device

When this instruction is executed, the value in the specified device D decreases by 1 in each scanning cycle of the program.

In 16-bit operation, -32,768 minus 1 is 32,767. In 32-bit operation, -2,147,483,648 minus 1 is 2,147,483,647. The operation result of this instruction does not affect flags M2824 - M2826.

Program example:

When X0 goes from Off to On, the value of D0 automatically decreases by 1.



■ API-19 WAND: AND operation

3			WAND		S_1, S_2, D				AND operation		NC series				
19	D														
	Bit device				Word device										
	X	Y	M	A	K	F	KnX	KnY	KnM	KnA	T	C	D	V	Z
S1					∗								∗		
S2					∗								∗		
D													∗		

16-bit instruction: WAND continuous execution type (6 steps).
32-bit instruction: DWAND continuous execution type (8 steps).
Flag: none.
Notes on the use of operands: if operands S_1, S_2 and D are used in register Z, only the16-bit instruction is applicable.

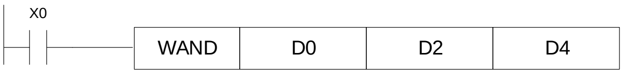
Instruction description:

S_1: source data device 1; S_2: source data device 2; D: operation result

Performs the AND operation for data source S_1 and S_2, and saves the result in D. In the logic of the AND operation, the operation result is 0 as long as S_1 or S_2 is 0.

Program example 1:

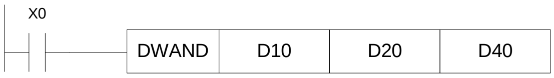
When X0 is On, perform the WAND operation for the 16-bit registers D0 and D2, and save the result in D4.



	b15															b0		
D0	0	0	1	1	1	1	0	1	0	1	0	0	0	1	0	0	S1	Before execution
									WAND									
D2	0	1	0	1	0	1	1	1	1	0	1	1	0	1	0	0	S2	
									⇩									
D4	0	0	0	1	0	1	0	1	0	0	0	0	0	1	0	0	D	After execution

Program example 2:

When X1 is On, perform the DWAND operation for the 32-bit registers (D11, D10) and (D21, D20), and save the result in (D41, D40).



				b31																b15															b0
Before execution	S1	D11	D10	1	1	1	1	1	1	1	1	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	1	1	1	1
																			DWAND																
	S2	D21	D20	0	0	0	1	0	0	1	0	0	0	1	1	0	1	0	0	0	0	0	1	0	0	1	0	0	0	1	1	0	1	0	0
																			⇩																
After execution	D	D41	D40	0	0	0	1	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	1	0	0	0	0	0	0	1	0	0

■ API-20 WOR: OR operation

API			WOR		S_1, S_2, D				OR operation			NC series			
20	D														
	Bit device				Word device										
	X	Y	M	A	K	F	KnX	KnY	KnM	KnA	T	C	D	V	Z
S1					∗								∗		
S2					∗								∗		
D													∗		

16-bit instruction: WOR continuous execution type (6 steps).
32-bit instruction: DWOR continuous execution type (8 steps).
Flag: none.
Notes on the use of operands: if operands S_1, S_2 and D are used in register Z, only 16-bit instruction is applicable.

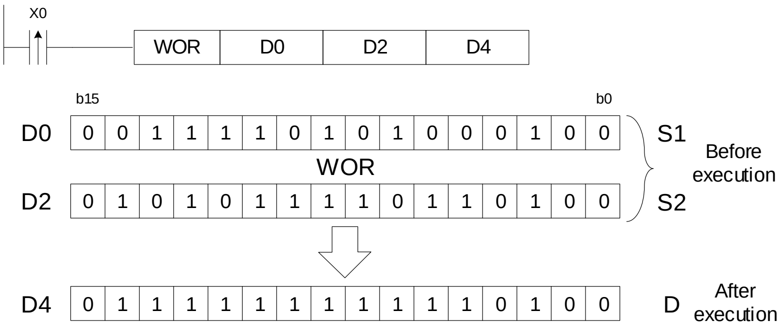
Instruction description:

S_1: source data device 1; S_2: source data device 2; D: operation result

Performs the OR operation on data source S_1 and S_2, and saves the result in D. In the logic of the OR operation, the operation result is 1 as long as S_1 or S_2 is 1.

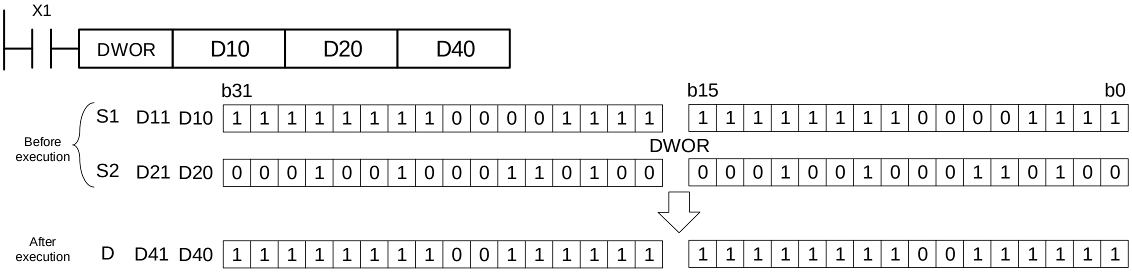
Program example 1:

When X0 is On, perform the WOR operation for the 16-bit registers D0 and D2, and save the result in D4.


X0
WOR
D0
D2
D4
b15
b0
D0 0 0 1 1 1 1 0 1 0 1 0 0 0 1 0 0
WOR
D2 0 1 0 1 0 1 1 1 1 0 1 1 0 1 0 0
S1
S2
Before execution
D4 0 1 1 1 1 1 1 1 1 1 1 1 0 1 0 0
D
After execution

Program example 2:

When X1 is On, perform the DWOR operation for the 32-bit registers (D11, D10) and (D21, D20), and save in (D41, D40).


X1
DWOR
D10
D20
D40
b31
b15
b0
Before execution
S1 D11 D10 1 1 1 1 1 1 1 1 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 1 1 1 1
DWOR
S2 D21 D20 0 0 0 1 0 0 1 0 0 0 1 1 0 1 0 0 0 0 0 1 0 0 1 0 0 0 1 1 0 1 0 0
After execution
D D41 D40 1 1 1 1 1 1 1 1 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 1 1 1 1 1 1

■ API-21 WXOR: XOR operation

API		WXOR	S₁, S₂, D				XOR operation		NC series					
21	D													
	Bit device				Word device									
	X	Y	M	A	K	F	KnX	KnY	KnM	KnA	T	C	D	V	Z
S_1					＊								＊		
S_2					＊								＊		
D													＊		

16-bit instruction: WXOR continuous execution type (6 steps).
32-bit instruction: DWXOR continuous execution type (8 steps).
Flag: none.
Notes on the use of operands: if operands S_1, S_2 and D are used in register Z, only 16-bit instruction is applicable.

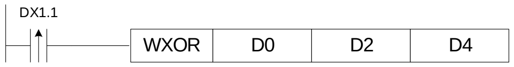
Instruction description:

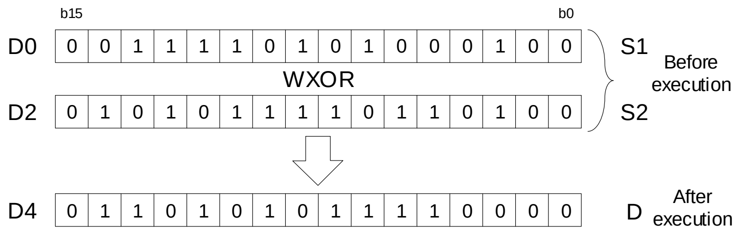
S_1: source data device 1; S_2: source data device 2; D: operation result

Performs the XOR operation for data source S_1 and S_2, and saves the result in D. In the logic of XOR operation logic, if S_1 = S_2, the result in D is 0. If S1 ≠ S2, the result in D is 1.

Program example 1:

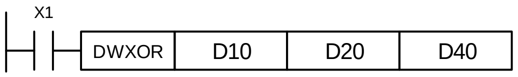
When X0 is ON, perform the WXOR operation for the 16-bit registers D0 and D2, and save the result in D4.

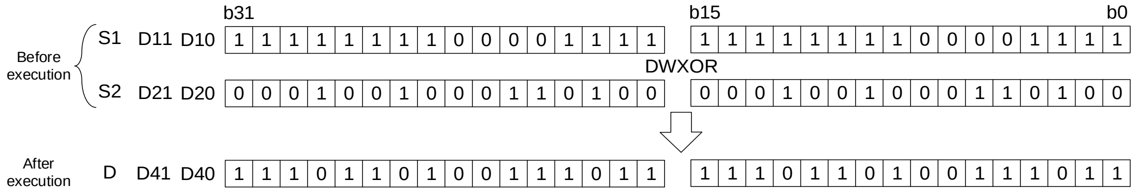



b15
b0
D0 0 0 1 1 1 1 0 1 0 1 0 0 0 1 0 0 S1
WXOR
D2 0 1 0 1 0 1 1 1 1 0 1 1 0 1 0 0 S2
Before execution
D4 0 1 1 0 1 0 1 0 1 1 1 1 0 0 0 0 D
After execution

Program example 2:

When X1 is On, perform the DWXOR operation for the 32-bit registers (D11, D10) and (D21, D20), and save the result in (D41, D40).




b31
b15
b0
Before execution
S1 D11 D10 1 1 1 1 1 1 1 1 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 1 1 1 1
DWXOR
S2 D21 D20 0 0 0 1 0 0 1 0 0 0 1 1 0 1 0 0 0 0 0 1 0 0 1 0 0 0 1 1 0 1 0 0
After execution
D D41 D40 1 1 1 0 1 1 0 1 0 0 1 1 1 0 1 1 1 1 1 0 1 1 0 1 0 0 1 1 1 0 1 1

■ API-22 NEG: Take the negative number (Two’s complement)

<table>
<tr><td>API</td><td colspan="2"></td><td colspan="2" rowspan="2">NEG</td><td colspan="4" rowspan="2">D</td><td colspan="2" rowspan="2">Take the negative number (Two’s complement)</td><td colspan="5" rowspan="2">NC series</td></tr>
<tr><td>22</td><td colspan="2">D</td></tr>
<tr><td></td><td colspan="4">Bit device</td><td colspan="11">Word device</td></tr>
<tr><td></td><td>X</td><td>Y</td><td>M</td><td>A</td><td>K</td><td>F</td><td>KnX</td><td>KnY</td><td>KnM</td><td>KnA</td><td>T</td><td>C</td><td>D</td><td>V</td><td>Z</td></tr>
<tr><td>D</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td>*</td><td></td><td></td></tr>
<tr><td colspan="16">16-bit instruction: NEG continuous execution type (3 steps).
32-bit instruction: DNEG continuous execution type (3 steps).
Flag: none.
Notes on the use of operands: if operand D is used in register Z, only 16-bit instruction is applicable.</td></tr>
</table>

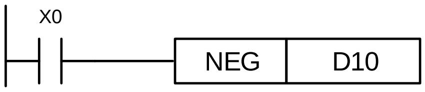
Instruction description:

D: the device requiring two’s complement

Converts a negative BIN value to an absolute value.

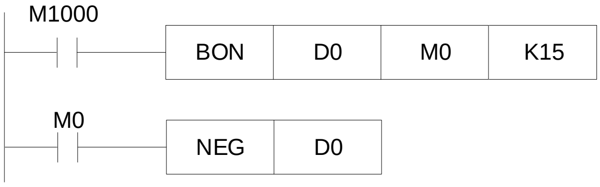
Program example 1:

When X0 goes from Off to On, all bits in D10 are inverted (0→1, 1→0), its value adds 1, and this result is saved in the original register D10.



Program example 2:

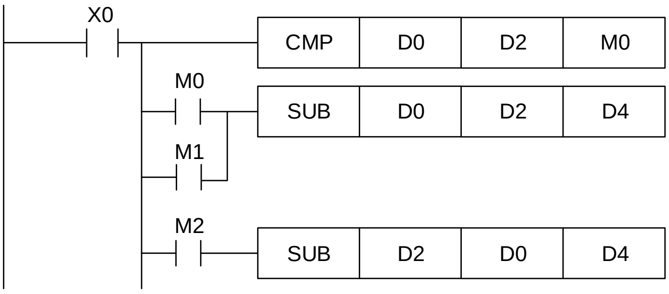
Get the negative absolute value: when the 15th bit of D0 is 1, M0 is On, meaning D0 is a negative value. When M0 is On, use the NEG instruction to get two’s compliment of D0 and then get its absolute value.



Example 3:

Get the absolute value of the difference from subtraction operation:

1. When X0 is On and D0 > D2, then M0 is On.
2. When X0 is On and D0 = D2, then M1 is On.
3. When X0 is On and D0 < D2, then M2 is On.
4. In this case, the value in D4 remains positive.



Supplementary notes on the negative value and its absolute value:

The highest (most left) bit in the register is a sign bit, 0 represents a positive value while 1 represents a negative value.

You can use the NEG instruction (API22) to convert a negative value to its absolute value.

(D0) = 2

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

(D0) = 1

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

(D0) = 0

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

(D0) = -1	1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	($\overline{D0}$) + 1 = 1	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1
(D0) = -2	1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0	($\overline{D0}$) + 1 = 2	0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0
(D0) = -3	1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1	($\overline{D0}$) + 1 = 3	0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1
(D0) = -4	1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0	($\overline{D0}$) + 1 = 4	0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0
(D0) = -5	1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1	($\overline{D0}$) + 1 = 5	0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 1
	⋮		⋮
(D0) = -32,765	1 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1	($\overline{D0}$) + 1 = 32,765	0 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1
(D0) = -32,766	1 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0	($\overline{D0}$) + 1 = 32,766	0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0
(D0) = -32,767	1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1	($\overline{D0}$) + 1 = 32,767	0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
(D0) = -32,768	1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	($\overline{D0}$) + 1 = -32,768	1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Max. absolute value is 32,767

4.4 Rotate and shift instructions

■ API-23 ROR: Rotate right

<table>
<tr><td>API</td><td colspan="2"></td><td colspan="2" rowspan="2">ROR</td><td colspan="4" rowspan="2">D, n</td><td colspan="2" rowspan="2">Rotate right</td><td colspan="5" rowspan="2">NC series</td></tr>
<tr><td>23</td><td colspan="2">D</td></tr>
<tr><td rowspan="2"></td><td colspan="4">Bit device</td><td colspan="11">Word device</td></tr>
<tr><td>X</td><td>Y</td><td>M</td><td>A</td><td>K</td><td>F</td><td>KnX</td><td>KnY</td><td>KnM</td><td>KnA</td><td>T</td><td>C</td><td>D</td><td>V</td><td>Z</td></tr>
<tr><td>D</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td>*</td><td></td><td></td></tr>
<tr><td>n</td><td></td><td></td><td></td><td></td><td>*</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
</table>

16-bit instruction: ROR continuous execution type (4 steps).
32-bit instruction: DROR continuous execution type (4 steps).
Flag: M2826 (carry flag)
Notes on the use of operands: if operand D is used in register Z, only the 16-bit instruction is applicable.

If operand D is assigned to KnY, KnM, and KnS, only K4 (16-bit) and K8 (32-bit) are valid.
Range of n: K1 - K16 (16-bit), K1 - K32 (32-bit).

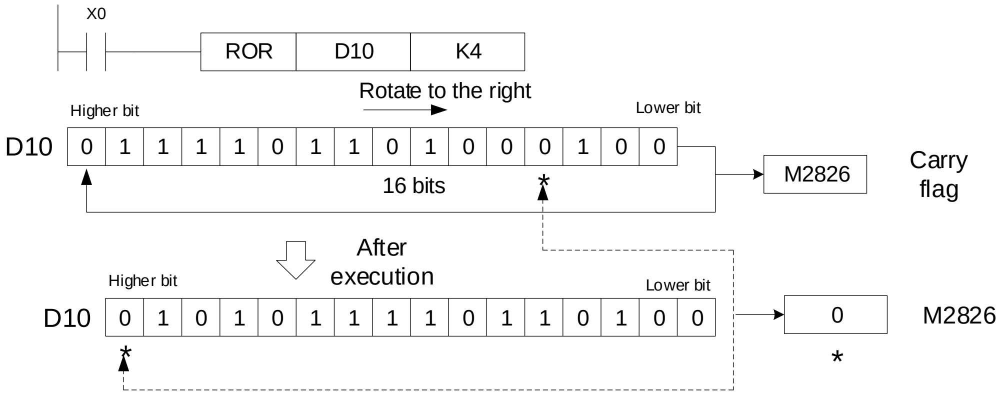
Instruction description:

D: the device to be rotated; n: the number of bits to be rotated for 1 rotation

Rotates the device content assigned by D to the right for n bits.

Program example:

When X0 goes from Off to On, the 16 bits in D10 rotates to the right in the unit of 4 bits. As shown in the following figure, the bit marked with * is sent to the carry flag M2826.



■ API-24 ROL: Rotate left

API		ROL	D, n	Rotate left	NC series
24	D				

	Bit device				Word device										
	X	Y	M	A	K	F	KnX	KnY	KnM	KnA	T	C	D	V	Z
D													∗		
n					∗										

16-bit instruction: ROL continuous execution type (4 steps).
32-bit instruction: DROL continuous execution type (4 steps).
Flag: M2826 (carry flag)
Notes on the use of operands: if operand D is used in register Z, only 16-bit instruction is applicable.
If D is assigned to KnY, KnM, and KnS, only K4 (16-bit) and K8 (32-bit) are valid.
Range of n: K1 - K16 (16-bit), K1 - K32 (32-bit).

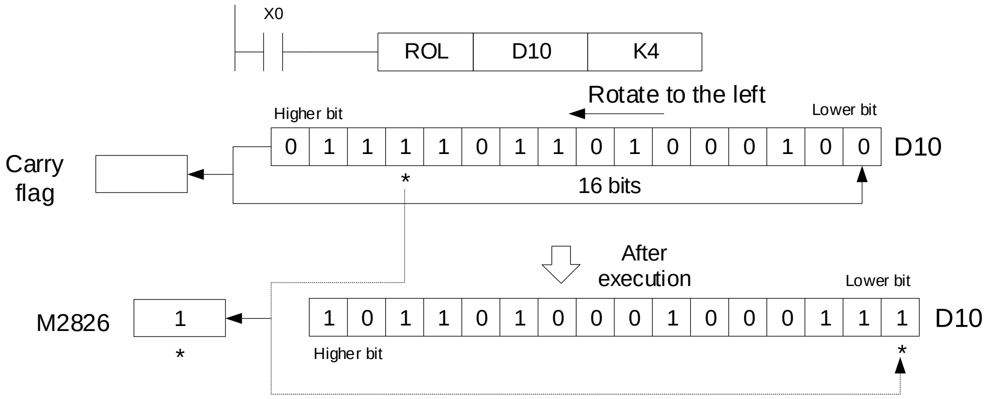
Instruction description:

D: the device to be rotated; n: the number of bits to be rotated for 1 rotation

Rotates the device content assigned by D to the left for n bits.

Program example:

When X0 goes from Off to On, the 16 bits in D10 rotates to the left in the unit of 4 bits. As shown in the following figure, the bit marked with * is sent to the carry flag M2826.



4.5 Data processing instructions

■ API-25 ZRST: Zone reset

API			ZRST		D_1, D_2				Zone reset		NC series				
25	-														
	Bit device				Word device										
	X	Y	M	A	K	F	KnX	KnY	KnM	KnA	T	C	D	V	Z
D_1		*	*	*							*	*	*		
D_2		*	*	*							*	*	*		

16-bit instruction: ZRST continuous execution type (4 steps).
32-bit instruction: none.
Flag: none.
Notes on the use of operands:
No. of D1 must be ≦ No. of D2,
Assign the same type of device for operands D_1 and D_2.

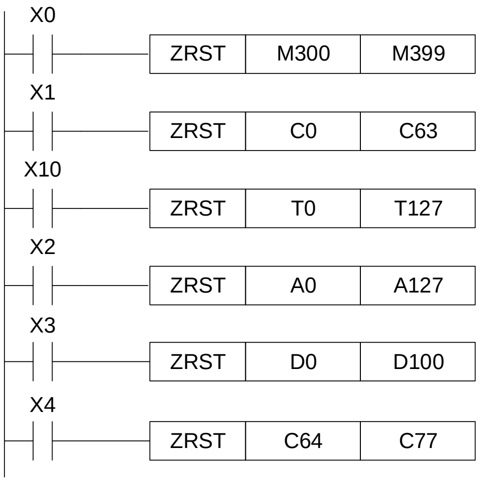
Instruction description:

D_1: the device for starting the zone reset; D_2: the device for ending the zone reset

In NC series models, the 16-bit and 32-bit counters cannot use the ZRST instruction at the same time. When the number of D_1 is larger than the number of D_2, only the device assigned by D_2 is reset.

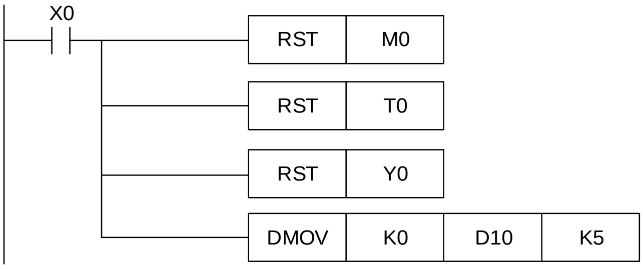
Program example:

1. When X0 is ON, the auxiliary relays M300 - M399 are reset to Off.
2. When X1 is On, the 16-bit counters C0 - C63 are all reset (Write the value 0; contacts and coils are reset to Off).
3. When X10 is On, the timers T0 - T127 are all reset. (Write the value 0; contacts and coils are reset to Off).
4. When X2 is On, the alarm flags A0 - A127 are all reset to Off.
5. When X3 is On, the data registers D0 - D100 are all reset to 0.
6. When X4 is On, the 32-bit counters C64 - C77 are all reset. (Write the value 0; contacts and coils are reset to Off).



Supplementary note:

You can use the RST instruction independently on the devices, such as bit devices Y, M, and A, as well as the word devices T, C, and D. Likewise, you can use the instruction DMOV (API-09) to send K0 to word devices T, C, and D or bit registers KnY, KnM, and KnA for the reset, as shown in the following figure.



■ API-26 DECO: Decoder

API			DECO		S, D, n				Decoder		NC series				
26	-														
	Bit device				Word device										
	X	Y	M	A	K	F	KnX	KnY	KnM	KnA	T	C	D	V	Z
S	*	*	*	*	*								*		
D		*	*	*									*		
n					*										

16-bit instruction: DECO continuous execution type (6 steps).
32-bit instruction: none.
Flag: none.
Notes on the use of operands: when D is a bit device, the range of operand n is 1 - 8.
When D is a word device, the range of operand n is 1 - 4.

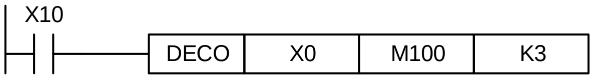
Instruction description:

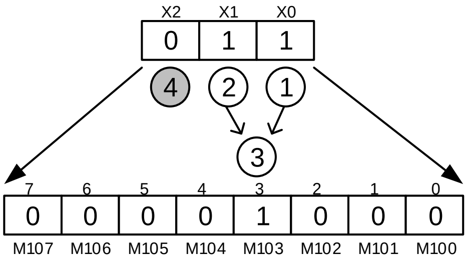
S: source device for decoding; D: device for saving the decoded result; n: length of the decoded bits

Decodes the lower n bits of S and saves the results which length is 2^n bits in D.

Program example 1:

1. When D is a bit device, n = 1 to 8. If n = 0 or n > 8, an error occurs.
2. When n = 8, this instruction can decode up to 2^8 points (= 256 points). (Be aware of the devices' storage range after decoding. Do not use the device that has been occupied.)





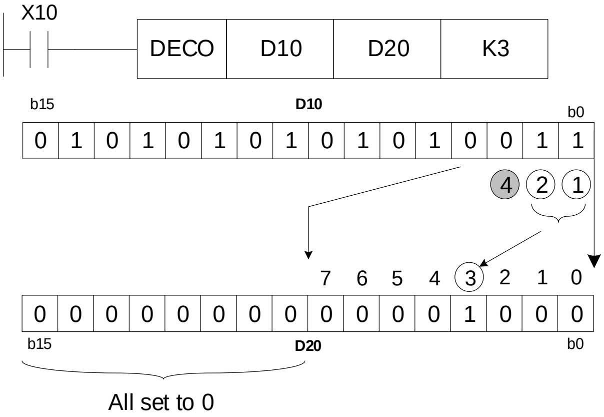
a. When X10 goes from Off to On, the DECO instruction decodes values in X0 - X2 to M100 - M107.

b. If the data source is 1 + 2 = 3, M103 is set to 1, which is the third bit starting from M100.

c. After the DECO instruction is complete and X10 goes to Off, the content that has been decoded remains its state.

Program example 2:

1. When D is a word device, n = 1 to 4. If n = 0 or n > 4, an error occurs.
2. When n = 4, this instruction can decode up to 2^4 points (= 16 points).



a. When X10 goes from Off to On, the DECO instruction decodes values in (b2 - b0) of D10 and save the result to (b7 - b0) of D20. The bits (b15 - b8) in D20 that have not been used are all set to 0.

b. The lower 3 bits of D10 are decoded and saved in the lower 8 bits of D20. The higher 8 bits are all set to 0.

c. When the DECO instruction is complete and X10 goes to Off, the bit that has been decoded operates as usual.

■ API-27 ENCO: Encoder

API		ENCO	S, D, n	Encoder	NC series
27	-				

	Bit device				Word device										
	X	Y	M	A	K	F	KnX	KnY	KnM	KnA	T	C	D	V	Z
S	*	*	*	*									*		
D													*		
n					*										

16-bit instruction: ENCO continuous execution type (6 steps).
32-bit instruction: none.
Flag: none.
Notes on the use of operands: when D is a bit device, the range of operand n is 1 - 8.
When D is a word device, the range of operand n is 1 - 4.

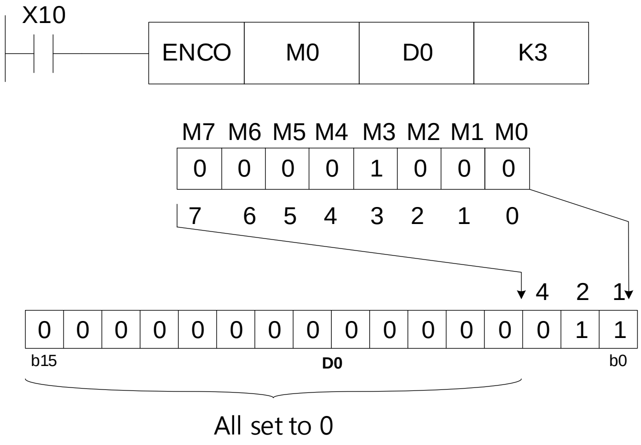
Instruction description:

S: source device for encoding; D: device for saving the encoded value; n: length of the encoded bits

Encodes the lower 2^n bits of S and saves the result is in D. If multiple bits in device S are 1, the lower bits are not processed.

Program example 1:

1. When S is a bit device, n = 1 to 8. If n = 0 or n > 8, an error occurs.
2. When n = 8, this instruction can encode up to 2^8 points (= 256 points).

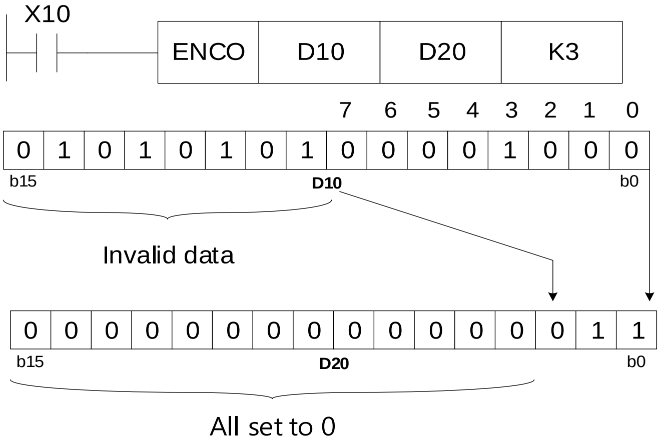


a. When X0 goes from Off to On, the ENCO instruction encodes 2^3 bits data in (M0 - M7) and saves the result in the lower 3 bits (b2 - b0) of D0. The unused bits (b15 - b3) in D0 are all set to 0.

b. When the ENCO instruction is complete and X0 goes to Off, the data in D remains unchanged.

Program example 2:

1. When S is a word device, n = 1 to 4. If n = 0 or n > 4, an error occurs.
2. When n = 4, this instruction can encode up to 2^4 points (= 16 points).



a. When X0 goes from Off to On, the ENCO instruction encodes 2^3 bits data in (b0 - b7) and saves the result in the lower 3 bits (b2 - b0) of D20. The unused bits (b15 - b3) in D20 are all set to 0. (b8 - b15 in D10 are invalid data.)

b. When the ENCO instruction is complete and X0 goes to Off, the data in D stays the same.

API-28 BON: Bit state monitoring

API		BON	S, D, n	Bit state monitoring	NC series
28	D				

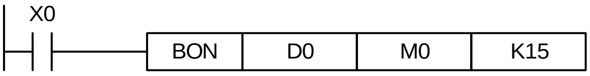
	Bit device				Word device										
	X	Y	M	A	K	F	KnX	KnY	KnM	KnA	T	C	D	V	Z
S					*						*	*	*		
D		*	*	*											
n					*										

16-bit instruction: BON continuous execution type (6 steps).
32-bit instruction: DBON continuous execution type (7 steps).
Flag: none.
Notes on the use of operands: if operand S is used in register Z, only the 16-bit instruction is applicable.
n = 0 to 15 (16-bit instruction)
n = 0 to 31 (32-bit instruction)

Instruction description:

S: source device; D: device for saving the result; n: monitoring bit (starting from 0)

Program example:



	b15															b0	
D0	0	0	0	0	1	0	0	0	1	0	1	0	0	0	1	1	⇨ M0 = Off
D0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	1	1	⇨ M0 = On

1. When X0 is On, if the 15th bit of D0 is 1, M0 is On. If it is 0, M0 goes Off.
2. When X0 goes Off, M0 remains its previous state.

API-29 ANS: Alarm output

API		ANS	S, M, D	Alarm output	NC series
29	-				

	Bit device				Word device										
	X	Y	M	A	K	F	KnX	KnY	KnM	KnA	T	C	D	V	Z
S											*				
M					*										
D				*											

16-bit instruction: ANS continuous execution type (5 steps).
32-bit instruction: none.
Flag: none.
Notes on the use of operands:
Range of S: T0 - T255 for NC series models.
Range of m: K1 to K32,767 in the unit of 100 ms or 10 ms, which is determined by T(n). n = 0 to 255.
Range of A: A0 to A511 for NC series models.
T0 to T199 (unit: 100 ms), T200 to T255 (unit: 10 ms).

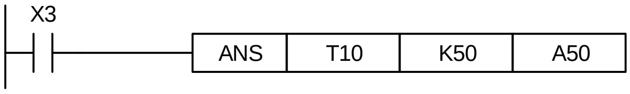
Instruction description:

S: alarm detection timer; m: time setting; D: alarm device

The ANS instruction is used for triggering the alarm.

Program example:

When X3 is On for over 5 seconds, the alarm flag A50 goes ON. A50 remains On even if X3 goes Off. However, T10 resets to Off, and its current value is 0.



■ API-30 ANR: Alarm reset

<table>
<tr><td>API</td><td colspan="2"></td><td colspan="2" rowspan="2">ANR</td><td colspan="4" rowspan="2">-</td><td colspan="2" rowspan="2">Alarm reset</td><td colspan="5" rowspan="2">NC series</td></tr>
<tr><td>30</td><td colspan="2">-</td></tr>
<tr><td rowspan="2"></td><td colspan="4">Bit device</td><td colspan="11">Word device</td></tr>
<tr><td>X</td><td>Y</td><td>M</td><td>A</td><td>K</td><td>F</td><td>KnX</td><td>KnY</td><td>KnM</td><td>KnA</td><td>T</td><td>C</td><td>D</td><td>V</td><td>Z</td></tr>
</table>

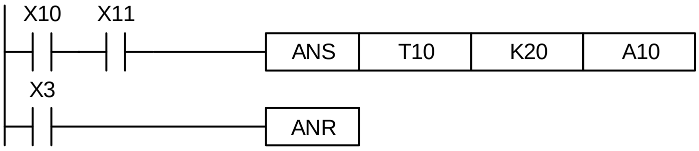
16-bit instruction: ANR continuous execution type (1 step).
32-bit instruction: none.
Flag: none.
Notes on the use of operands: no operand.

Instruction description:

The ANR instruction is used for alarm reset. When multiple alarms are On (triggered), the alarm with a smaller number is reset.

Program example:

1. When X10 and X11 are On simultaneously for over 2 seconds, the alarm flag A10 goes On. A10 remains On even if X10 and X11 go Off. However, T10 resets to Off, and its current value is 0.
2. When X10 and X11 are On simultaneously for less than 2 seconds, the current value of T10 resets to 0.
3. When X3 goes from Off to On, the current alarm is reset. You can use can use alarm flags A0 - A511 for the NC series models .
4. When X3 goes from Off to On, the alarm with the second smallest number is cleared.



4.6 High-speed processing instructions

■ API-31 REF: I/O refresh

API		REF	D, n	I/O refresh	NC series
31	-				

	Bit device				Word device										
	X	Y	M	A	K	F	KnX	KnY	KnM	KnA	T	C	D	V	Z
D	∗	∗													
n					∗										

16-bit instruction: REF continuous execution type (3 steps).
32-bit instruction: none.
Flag: none.
Notes on the use of operands: you must assign the points such as X0, X10, Y0 and Y10 whose number ends with 0 for the operand D. (Refer to the supplementary notes.)
Range of n: n = 8 to 256, and n has to be a multiple of 8.

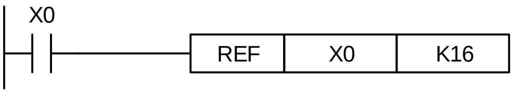
Instruction description:

D: the device for starting performing I/O refresh; n: the number of devices to perform I/O refresh

1. The state of all MLC input/output contacts are updated after the END instruction is scanned. When the program starts to scan, the MLC reads the state from the external input point and saves this state to the input point memory. After the END instruction is executed, the MLC sends the content of the output point memory to the output device. Therefore, you can use this instruction when the latest input/output data are required for the operation.
2. You must assign the points such as X0, X10, Y0 and Y10 whose number ends with 0 for the operand D. The range of n is 8 to 256 and has to be a multiple of 8; otherwise, the number is regarded as an error. The application range varies with the models. Refer to the following supplementary notes.

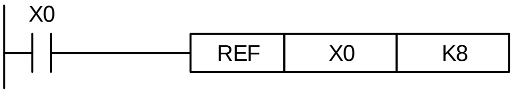
Program example 1:

When X0 is On, the MLC immediately reads the state of the input points X0 - X15 and updates the input signals without any delay.



Program example 2:

When X0 is On, the 8 output signals from Y0 - Y7 are immediately sent to the output points and updated without waiting for the END instruction.



Supplementary note:

The NC series models can process the input/output contacts of main board I/O and RIO, i.e. n = K8 or n= K16.

■ API-32 DHSCS: Comparison setting (high-speed counter)

API	-	DHSCS	S_1, S_2, D	Comparison setting	NC series
32	D				

	Bit device				Word device										
	X	Y	M	A	K	F	KnX	KnY	KnM	KnA	T	C	D	V	Z
S_1					∗								∗		
S_2												∗			
D		∗	∗	∗											

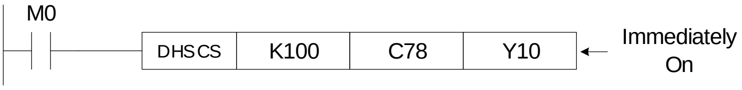
16-bit instruction: none.
32-bit instruction: DHSCS continuous execution type (5 steps).
Flag: M2888 - M2889, for disabling high-speed counter interruption. (Refer to Program example 3.)
Notes on the use of operands: you have to assign high-speed counters C78 and C79 for operand S_2. (Please refer to the supplementary notes.)
The operand D can assign IC00 and IC01, and D can be modified by indirect registers V and Z.
Refer to Chapter 1 for the range of each device.
Only the 32-bit instruction DHSCS is valid.
This instruction sets a comparison value for the high-speed counter and sets the device to On when the counter reaches the set comparison value.
The high-speed counter counts with the I/O signals. When the counter reaches the set value, it returns an interruption signal. If the interruption is enabled, the counter outputs with High Level and can output the signal to the M device or On Board Y device. If the counter outputs to the Y device, the output result differs based on the polarity setting (normally open / normally closed contact type) for the Y device.

Instruction description:

S_1: comparison value; S_2: high speed counter number; D: comparison result

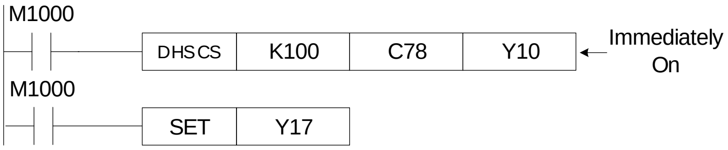
1. The high-speed counter is triggered by an external high-speed input signal. When the count of the high speed counter assigned by S_2 increase by 1 or decreases by 1, DHSCS instruction immediately compares the values. When the counter's current value equals the comparison value assigned by S_1, the device assigned by D goes to On and remains On even when S_1 does not equal S_2 afterwards.
2. If the devices assigned by D are Y0 - Y23 (only On board Y) and the comparison value equals the current value of the high-speed counter, Y0 - Y23 (only On board Y) are immediately set to On, and this changes the signals of the external mechanical parts. The rest of the Y devices are still influenced by the scan cycle. On the other hand, devices M and A act immediately without being influenced by the scan cycle.

Program example 1:



When the MLC executes the RUN instruction and M0 is On, the DHSCS instruction is executed. If the current value of C78 changes from 99 to 100 or from 101 to 100, Y10 goes to On and this signal is instantly output to the external Y10 output point and remains On all the time.

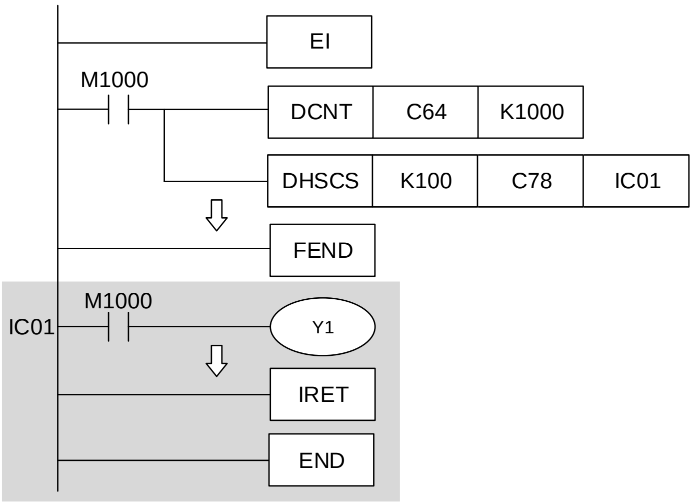
Program example 2:



Differences between the Y output of DHSCS instruction and general Y output:

1. When the current value of C79 changes from 99 to 100 or from 101 to 100, Y10 of the DHSCS instruction outputs immediately to the external output contacts by inserting an interruption, which is irrelevant to the MLC scan cycle. However, there is a 10 ms delay (caused by the relay) or 10 us delay (caused by the resistor) in the output.
2. When the current value of C79 changes from 99 to 100, the contact C79 immediately goes to On. When SET Y17 is executed, the output of Y17 is influenced by the scan cycle, so it goes to On after the END instruction is executed.

Example 3:



High-speed counter interruption:

1. You can assign IC00 and IC01 as the interruption timing settings for the operand D of DHSCS. In other words, when the counter reaches its set value, the interruption for the service program is executed.
2. On the NC series models, there are limits of using high-speed counter for interruption. When the DHSCS instruction assigns an I for interruption, the high-speed counter cannot be used in other DHSCS and DHSCR instructions. Incorrect use of high-speed counter will cause program errors.
3. On the NC series models, when the active high-speed counter reaches its set value, it triggers the interruption. C78 serves as the first counter, and IC00 or IC01 is assigned as the indicator of interruption.
4. When the current value of C78 goes from 99 to 100 and from 101 to 100 (counting down by referring to the MLC parameter #312), the program jumps to the interruption indicator IC01 to interrupt the service subprogram.

On the NC series models, M2888 - M2889 correspond to the high-speed counters IC00 - IC01 respectively. When M2888 is Off, the interruption of IC00 is disabled.

Interruption indicator	Interruption prohibited flag
IC00	M2888
IC01	M2889

■ API-33 DHSCR: Comparison reset (high-speed counter)

API	-	DHSCR	S_1, S_2, D	Comparison reset	NC series
33	D				

	Bit device				Word device										
	X	Y	M	A	K	F	KnX	KnY	KnM	KnA	T	C	D	V	Z
S_1					*								*		
S_2												*			
D		*	*	*											

16-bit instruction: none.
32-bit instruction: DHSCR continuous execution type (5 steps).
Flag: none.
Notes on the use of operands: you have to assign high-speed counters C78 and C79 for operand S_2. (Refer to the supplementary notes for API-32 DHSCS.)
Operand D can assign the same counter number assigned by S_2 (only applicable to C78 and C79).
Only the 32-bit instruction DHSCR is valid.
This instruction sets a comparison value for the high-speed counter and sets the device to Off when the counter reaches the set comparison value.
The high-speed counter counts with the I/O signals. When the counter reaches the set value, it returns an interruption signal. If the interruption is enabled, the counter outputs with Low Level and can output the signal to the M device or On Board Y device. If the counter outputs to the Y device, the output result differs based on the polarity setting (normally open / normally closed contact type) for the Y device.

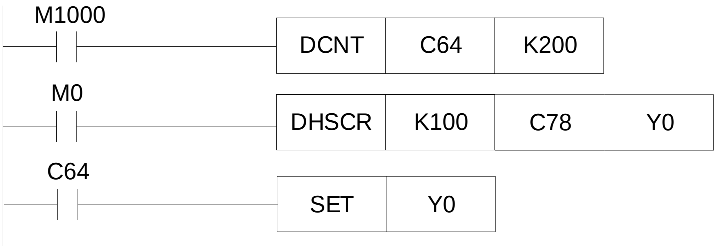
Instruction description:

S_1: comparison value; S_2: high-speed counter number; D: comparison result

1. The high-speed counter is triggered by an external high-speed input signal. When the count of the high-speed counter assigned by S_2 increases by 1 or decreases by 1, the DHSCR instruction immediately performs the comparison. When the counter's current value equals the comparison value assigned by S_1, the device assigned by D goes Off and remains Off even if the comparison results become unequal afterwards.
2. If the devices assigned by D are Y0 - Y23 (only On board Y) and the comparison value equals the current value of the high-speed counter, Y0 - Y23 (only On board Y) are immediately set to Off, and this changes the signals of the external mechanical parts. On the other hand, devices M and S act immediately without being influenced by the scan cycle.

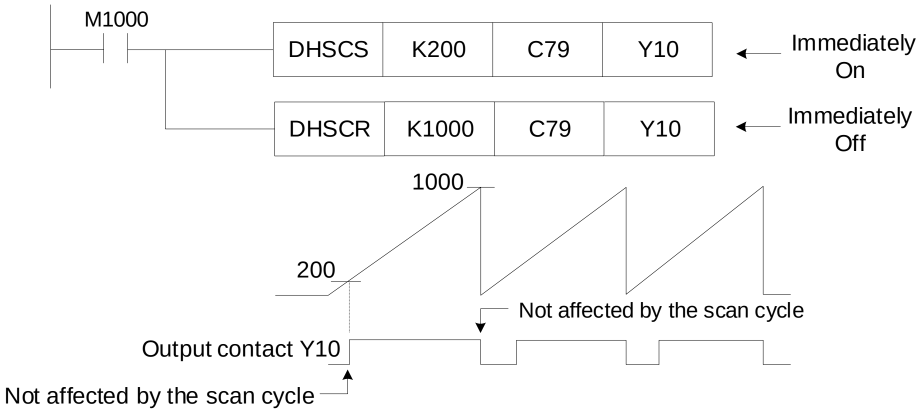
Program example 1:

1. When M0 is On and the current value of high-speed counter C78 goes from 99 to 100 or from 101 to 100, Y0 is cleared and goes to Off.
2. When the current value of counter C64 goes from 199 to 200, the C64 contact goes On and has Y0 switched to On; however, the program scan time is delayed.
3. Y0 is the component to be immediately reset when the counter reaches the set count. You can also assign Y0 with the high-speed counter of the same number. (Refer to Program example 2.)



Program example 2:

When you set Y10 as the high-speed counter with the same number and the current value of high-speed counter C79 goes from 999 to 1000 or from 1001 to 1000, the C79 contact is cleared and set to Off.



4.7 Shortcut instructions

■ API-34 ALT: On / Off alternation

API		ALT	D	On / Off alternation	NC series
34	-				

	Bit device				Word device										
	X	Y	M	A	K	F	KnX	KnY	KnM	KnA	T	C	D	V	Z
D		*	*	*											

16-bit instruction: ALT continuous execution type (3 steps).
32-bit instruction: none.
Flag: none.
Notes on the use of operands: refer to Chapter 1 for the range of each device.

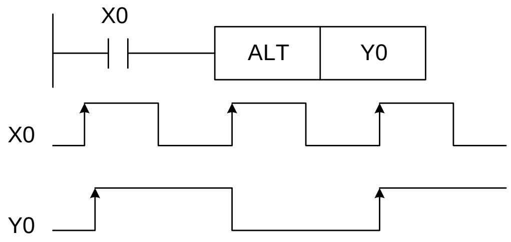
Instruction description:

D: destination device

This instruction is usually used as an execution type instruction (ALT).

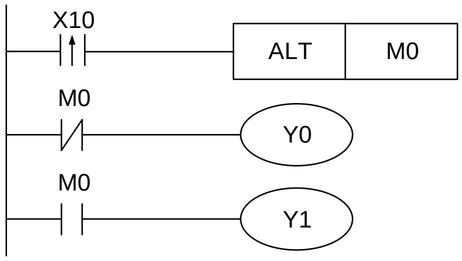
Program example 1:

When X0 changes from Off to On for the first time, Y0 goes to On. When X0 goes On for the second time, Y0 turns Off.



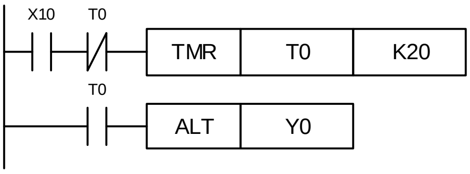
Program example 2:

Use a single switch to enable and disable the control. At the beginning, M0 is Off thus Y0 is On and Y1 is Off. When X10 switches between On and Off, M0 goes On, thus Y1 is On and Y0 is Off. For the second time that X10 switches between On and Off, M0 goes Off, thus Y0 is ON and Y1 is Off.



Example 3:

The ALT instruction can be used to enable Y0 flashing. When X10 is On, T0 generates one pulse every 2 seconds and Y0 switches between On and Off every time in accordance with pulses from T0.



4.8 Contact type comparison instructions

■ API-39 to 44 LD※: Compare the contact type

API		LD※	S_1, S_2	Compare the contact type	NC series
39 - 44	D				

	Bit device				Word device										
	X	Y	M	A	K	F	KnX	KnY	KnM	KnA	T	C	D	V	Z
S_1					*						*	*	*		
S_2					*						*	*	*		

16-bit instruction: LD※ continuous execution type (4 steps).

16-bit instruction: DLD※ continuous execution type (6 steps).

Flag: none.

Notes on the use of operands: refer to Chapter 1 for the range of each device.

※: =, >, <, <>, ≦, ≧

Instruction description:

S_1: data source device 1; S_2: data source device 2

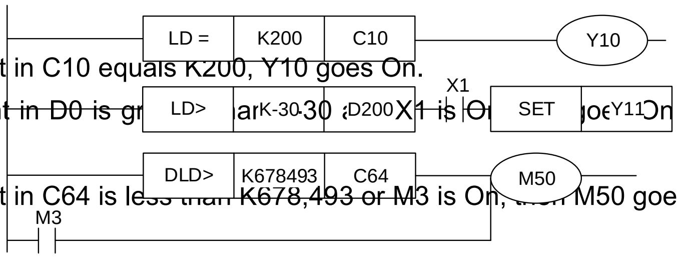
1. Take API 39 (LD=) instruction as an example: if S_1 equals S_2, the continuity of the instruction is enabled; if S_1 does not equal S_2, the continuity of the instruction is disabled. The LD※ instruction may connect to a bus bar directly, as shown in the following table.

API No.	16-bit instruction	32-bit instruction	Continuity condition	No-continuity condition
39	LD =	DLD =	$S_1 = S_2$	$S_1 \neq S_2$
40	LD >	DLD >	$S_1 > S_2$	$S_1 \leqq S_2$
41	LD <	DLD <	$S_1 < S_2$	$S_1 \geqq S_2$
42	LD < >	DLD < >	$S_1 \neq S_2$	$S_1 = S_2$
43	LD < =	DLD < =	$S_1 \leqq S_2$	$S_1 > S_2$
44	LD > =	DLD > =	$S_1 \geqq S_2$	$S_1 < S_2$

2. Use the 32-bit instruction (DLD※) to compare the 32-bit counters (C64 - C77). If you use the 16-bit instruction (AND※) for comparison, the MLC treats it as a program error and the ERROR indicator on the panel flashes.

Program example:

1. When the content in C10 equals K200, Y10 goes On.
2. When the content in D0 is greater than K-30 and X1 is On, Y11 goes On and remains its state.
3. When the content in C64 is less than K678,493 or M3 is On, then M50 goes to On.



■ API-45 to 50 AND※: Compare the contact type

API		AND※	S_1, S_2	Compare the contact type	NC series
45 - 50	D				

	Bit device				Word device										
	X	Y	M	A	K	F	KnX	KnY	KnM	KnA	T	C	D	V	Z
S_1					∗						∗	∗	∗		
S_2					∗						∗	∗	∗		

16-bit instruction: AND※ continuous execution type (4 steps).

32-bit instruction: DAND※ continuous execution type (6 steps).

Flag: none.

Notes on the use of operands: refer to Chapter 1 for the range of each device.

※: =, >, <, <>, ≦, ≧

Instruction description:

S_1: data source device 1; S_2: data source device 2

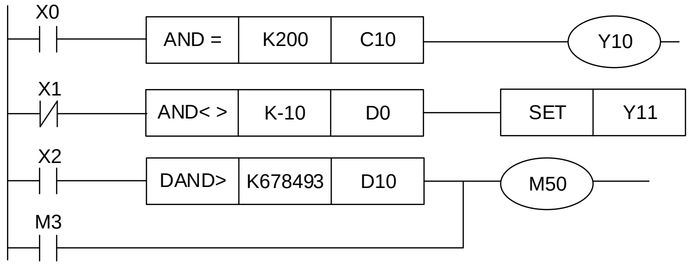
1. Compares the contents in S_1 and S_2. Take the API-45 (AND=) instruction as an example: when S_1 equals S_2, the continuity of the instruction is enabled; when S_1 does not equal S_2, the continuity of the instruction is disabled.
2. AND※ is a comparison instruction that connects to contacts in parallel, as shown in the following table.

API No.	16-bit instruction	32-bit instruction	Continuity condition	No-continuity condition
45	AND =	DAND =	$S_1 = S_2$	$S_1 \neq S_2$
46	AND >	DAND >	$S_1 > S_2$	S_1 ≦ S_2
47	AND <	DAND <	$S_1 < S_2$	S_1 ≧ S_2
48	AND < >	DAND < >	$S_1 \neq S_2$	$S_1 = S_2$
49	AND < =	DAND < =	S_1 ≦ S_2	$S_1 > S_2$
50	AND > =	DAND > =	S_1 ≧ S_2	$S_1 < S_2$

3. Use the 32-bit instruction (DAND※) to compare 32-bit counters (C64 - C77). If you use the 16-bit instruction (AND※), the MLC treats it as a program error and the ERROR indicator on the panel flashes.

Program example:

1. When X0 is On and the value of C10 equals K200, Y10 goes to On.
2. When X1 is Off and the content of D0 does not equal K-10, Y11 goes to On and remains its state.
3. When X2 is On and the content of register D10 (D11) is less than K678,493 or M3 is On, M50 goes to On.



■ API-51 to 56 OR※: Compare the contact type

API		OR※	S_1, S_2	Compare the contact type	NC series
51 - 56	D				

	Bit device				Word device										
	X	Y	M	A	K	F	KnX	KnY	KnM	KnA	T	C	D	V	Z
S_1					∗						∗	∗	∗		
S_2					∗						∗	∗	∗		

16-bit instruction: OR※ continuous execution type (4 steps).

32-bit instruction: DOR※ continuous execution type (6 steps).

Flag: none.

Notes on the use of operands: refer to Chapter 1 for the range of each device.

※: =, >, <, <>, ≦, ≧

Instruction description:

S_1: data source device 1; S_2: data source device 2

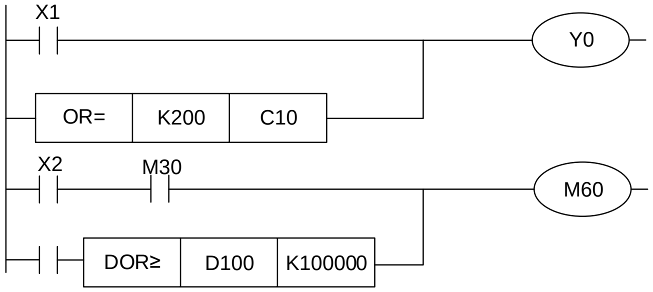
1. Compares contents in S_1 and S_2. Take the API-51(OR=) instruction as an example. When the S_1 equals S_2, the continuity of the instruction is enabled; when S_1 does not equal S_2, the instruction is disabled.
2. OR※ is a comparison instruction that connects to contacts in parallel, as shown in the following table.

API No.	16-Bit instruction	32-bit instruction	Continuity condition	No-continuity condition
51	OR =	DOR =	$S_1 = S_2$	$S_1 \neq S_2$
52	OR >	DOR >	$S_1 > S_2$	$S_1 \leqq S_2$
53	OR <	DOR <	$S_1 < S_2$	$S_1 \geqq S_2$
54	OR < >	DOR < >	$S_1 \neq S_2$	$S_1 = S_2$
55	OR < =	DOR < =	$S_1 \leqq S_2$	$S_1 > S_2$
56	OR > =	DOR > =	$S_1 \geqq S_2$	$S_1 < S_2$

3. Use the 32-bit instruction (DOR※) to compare 32-bit counters (C64 - C77). If you use the 16-bit instruction (OR※) is used, the MLC treats it as a program error and the ERROR indicator on the panel flashes.

Program example:

1. When X1 is On and the value in C10 equals K200, Y0 goes to On.
2. When both X2 and M30 are On, or the content in the 32-bit register D100 (D101) is greater or equal to K100, 000, then M60 goes On.



■ API-57 VRT: Logical switch table

API			VRT		S, n, D				Logical switch table		NC series				
57	D														
	Bit device				Word device										
	X	Y	M	A	K	F	KnX	KnY	KnM	KnA	T	C	D	V	Z
S	∗	∗	∗								∗	∗			
n					∗										
D													∗		

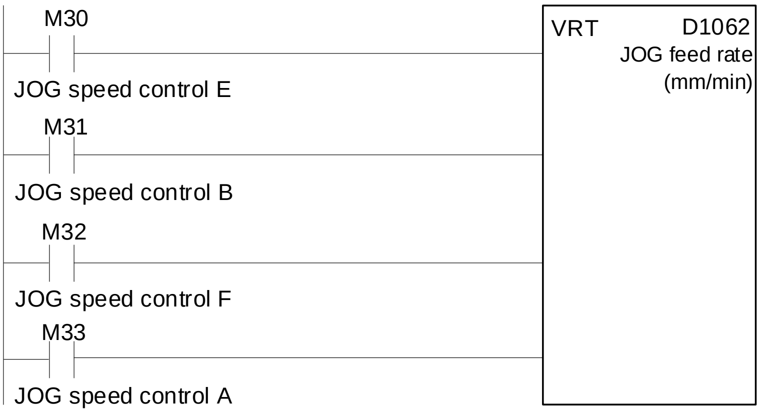
16-bit instruction: VRT continuous execution type (70 steps).
32-bit instruction: DVRT continuous execution type (134 steps).
Flag: none.
Notes on the use of operands: none.

Instruction description:

S: source device to be switched; n: number of source devices; D: switch result

1. Uses the initial source device assigned by S and the number of devices assigned by n for the switch. When the source device performs switching, the corresponding values in the logic (variable) table is moved to the specified register D.
2. Note that registers D, X, Y, M, T or C can be assigned as the source device. When the contact switches on the source device, you can switch the source to the default value.

Program example 1:



	+0	+1	+2	+3	+4
‣ 0	0	20	32	50	79
5	126	200	320	500	790
10	1260	2000	3200	5000	7900
15	12600				

When M30 is On, M31 is On, M32 is Off, and M33 is Off, M30 - M33 is 3 in binary format, and its corresponding value in the variable table is 50. Thus, 50 is saved in D1062.

4.9 Floating-point number operation instructions

■ API-58 FADD: Binary floating-point number addition

API	-	FADD	S_1, S_2, D	Binary floating-point number addition	NC series
58					

	Bit device				Word device										
	X	Y	M	A	K	F	KnX	KnY	KnM	KnA	T	C	D	V	Z
S_1						＊							＊		
S_2						＊							＊		
D													＊		

16-bit instruction: none.
32-bit instruction: FADD continuous execution type (7 steps).
Flag: M2824 (zero flag)
Notes on the use of operands: refer to Chapter 1 for the range of each device.
Only 32-bit instruction FADD is valid.

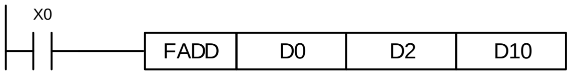
Instruction description:

S_1: summand; S_2: addend; D: sum

1. Adds the contents of S_1 and S_2, and saves the result in the register assigned by D. The addition is executed in binary floating-point number format.
2. If the operand of S_1 or S_2 is assigned with constant K or F, this instruction converts the constant to binary floating-point number for addition.
3. S_1 and S_2 can assign the register with the same number. In such case, if a continuous execution type instruction is executed and while the condition contact stays On, the addition operation is performed on the register each scan cycle.
4. If the operation result is 0, the zero flag M2824 goes to On.

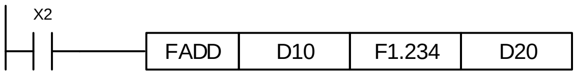
Program example 1:

When X0 is On, add the binary floating-point number of (D1, D0) and the binary floating-point number of (D3, D2), and save the sum in (D11, D10).



Program example 2:

When X2 is On, add the binary floating-point number of (D11, D10) and F1.234 (automatically converted to binary floating-point number), and save the sum in (D21, D20).



■ API-59 FSUB: Binary floating-point number subtraction

API	-		FSUB		S$_1$, S$_2$, D				Binary floating-point number subtraction		NC series				
59															
	Bit device				Word device										
	X	Y	M	A	K	F	KnX	KnY	KnM	KnA	T	C	D	V	Z
S$_1$						∗							∗		
S$_2$						∗							∗		
D													∗		

16-bit instruction: none.
32-bit instruction: FSUB continuous execution type (7 steps).
Flag: M2824 (zero flag).
Notes on the use of operands: refer to Chapter 1 for the range of each device.
Only the 32-bit instruction FSUB is valid.

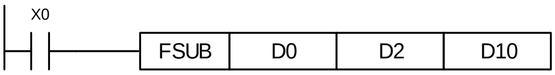
Instruction description:

S$_1$: minuend; S$_2$: subtrahend; D: difference

1. Subtracts S$_2$ from S$_1$, and saves the result in the register assigned by D. The subtraction is executed in binary floating-point number format.
2. If S$_1$ or S$_2$ is assigned with constant K or F, this instruction converts the constant to binary floating-point number for subtraction.
3. S$_1$ and S$_2$ can assign the register with the same ID number. In such case, if an instruction of continuous execution type is executed and while the condition contact stays On, the subtraction operation is performed on the register each scan cycle.
4. If the operation result is 0, then the zero flag M2824 goes On.

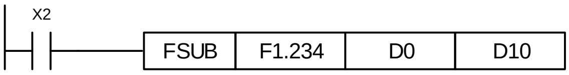
Program example 1:

When X0 is On, the subtract the binary floating-point number of (D3﹐D2) from the floating-point number of (D1, D0) and save the difference in (D11, D10).



Program example 2:

When X2 is On, subtract the binary floating-point number of (D1, D0) from F1.234 (automatically converted to binary floating-point number), and save the remainder in (D11, D10).



■ API-60 FMUL: Binary floating-point number multiplication

API	-	FMUL	S_1, S_2, D	Binary floating-point number multiplication	NC series
60					

	Bit device				Word device										
	X	Y	M	A	K	F	KnX	KnY	KnM	KnA	T	C	D	V	Z
S_1						*							*		
S_2						*							*		
D													*		

16-bit instruction: none.
32-bit instruction: FMUL continuous execution type (7 steps).
Flag: M2824 (zero flag).
Notes on the use of operands: refer to Chapter 1 for the range of each device.
Only the 32-bit instruction FMUL is valid.

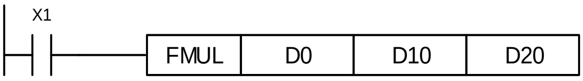
Instruction description:

S_1: multiplicand; S_2: multiplier; D: product

1. Multiplies S_1 and S_2, and saves the result in the register assigned by D. The multiplication is executed in binary floating-point number format.
2. If S_1 or S_2 is assigned with constant K or F, this instruction converts the constant to binary floating-point number for multiplication.
3. S_1 and S_2 can assign the register with the same ID number. In such case, if an instruction of continuous execution type is executed and while the condition contact stays On, the multiplication operation is performed on the register each scan cycle.
4. If the operation result is 0, the zero flag M2824 goes to On.

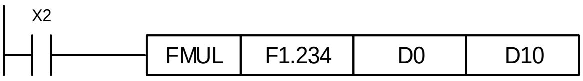
Program example 1:

When X0 is On, multiply the binary floating-point number of (D1, D0) and the binary floating-point number of (D11, D10), and save the result in the register specified by (D21, D20).



Program example 2:

When X2 is On, multiply F1.234 (automatically converted to binary floating- point number) and the binary floating-point number of (D1, D0), and save the result in (D11, D10).



■ API-61 FDIV: Binary floating-point number division

API	-	FDIV	S_1, S_2, D	Binary floating-point number division	NC series
61					

	Bit device				Word device										
	X	Y	M	A	K	F	KnX	KnY	KnM	KnA	T	C	D	V	Z
S_1						∗							∗		
S_2						∗							∗		
D													∗		

16-bit instruction: none.
32-bit instruction: FDIV continuous execution type (7 steps).
Flag: M2824 (zero flag).
Notes on the use of operands: refer to Chapter 1 for the range of each device.
Only the 32-bit instruction FDIV is valid.

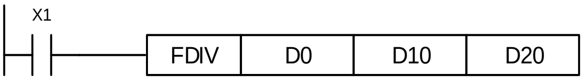
Instruction description:

S_1: dividend; S_2: divisor; D: quotient and remainder

1. Divides S_1 by S_2, and saves the result in register assigned by D. The division is executed in binary floating-point number format.
2. If S_1 or S_2 is assigned with constant K or F, this instruction converts the constant to binary floating-point number for division.
3. If the content of S_2 is 0, this division is regarded as “computing error” and the instruction is not executed. M1067 and M1068 go On along with the error code H’0E19 recorded in D1067.
4. If the operation result is 0, then the zero flag M2824 goes On.

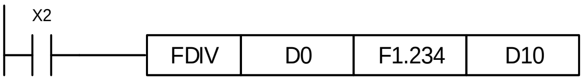
Program example 1:

When X0 is On, divide the binary floating-point number of (D1, D0) by (D11, D10), and save the remainder in the register specified by (D21, D20).



Program example 2:

When X2 is On, divide the binary floating-point number of (D1, D0) by F1.234 (automatically converted to binary floating-point number), and save the result in (D11, D10).



API-62 FCMP: Compare binary floating-point numbers

API	-	FCMP	S_1, S_2, D	Compare binary floating-point numbers	NC series
62					

	Bit device				Word device										
	X	Y	M	A	K	F	KnX	KnY	KnM	KnA	T	C	D	V	Z
S_1						*							*		
S_2						*							*		
D		*	*	*											

16-bit instruction: none.
32-bit instruction: FCMP continuous execution type (7 steps).
Flag: none.
Notes on the use of operands: refer to Chapter 1 for the range of each device.
Only the 32-bit instruction FCMP is valid.
Operand D occupies consecutive 3 points.

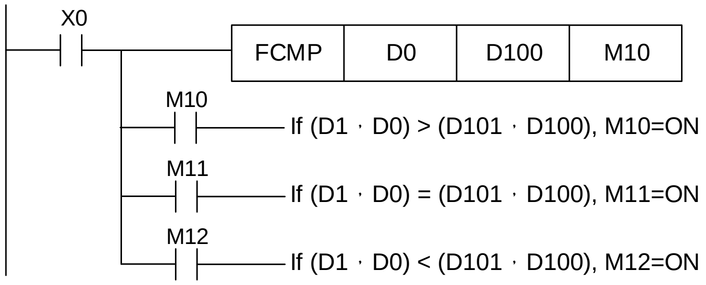
Instruction description:

S_1: binary floating-point number to be compared (1); S_2: binary floating-point number to be compared (2) ; D: comparison result, occupying consecutive 3 points.

1. Compares S_1 and S_2, and saves the comparison result (>, =, <) in the register assigned by D.
2. If S_1 or S_2 is assigned with constant K or F, the instruction converts the constant to binary floating-point number for comparison.

Program example:

1. If the assigned device is M10, then M10 - M12 are automatically used.
2. When X0 is On, the FCMP instruction is executed and one of M10 - M12 goes to On. When X0 is Off, the FCMP instruction is not executed and M10 - M12 remain the state before X0 went Off.
3. To get the result from ≧, ≦, and ≠ statements, arrange M10 - M12 in series or parallel.
4. To clear the result, use the RST and ZRST instructions.


X0
FCMP
D0
D100
M10
M10
If (D1 · D0) > (D101 · D100), M10=ON
M11
If (D1 · D0) = (D101 · D100), M11=ON
M12
If (D1 · D0) < (D101 · D100), M12=ON

■ API-63 FINT: Convert binary floating-point number to BIN integer

API	-	FINT	S, D	Convert binary floating-point number to BIN integer	NC series
63					

	Bit device				Word device										
	X	Y	M	A	K	F	KnX	KnY	KnM	KnA	T	C	D	V	Z
S						*							*		
D													*		

16-bit instruction: none.
32-bit instruction: FINT continuous execution type (5 steps).
Flag: M2824 (zero flag).
Notes on the use of operands: refer to Chapter 1 for the range of each device.
Only the 32-bit instruction FINT is valid.
Operand D takes consecutive 2 devices.

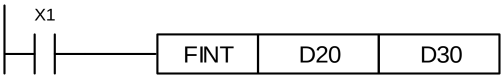
Instruction description:

S: source device to be converted; D: conversion result

Converts the register content assigned by S from the format of binary floating-point number to BIN integer, and saves the result in the register assigned by D. The floating-point number of the BIN integer is discarded. The FINT instruction is the opposite operation of API-64 FDOT instruction. If the conversion results in 0, the zero flag M2824 goes On.

Program example:

When X1 is On, convert the binary floating-point numbers of (D21, D20) to BIN integers, and save the result in (D31, D30) with the floating-point numbers discarded.



■ API-64 FDOT: Convert BIN integer to binary floating-point number

API	-	FDOT	S, D	Convert BIN integer to binary floating-point number	NC series
64					

	Bit device				Word device										
	X	Y	M	A	K	F	KnX	KnY	KnM	KnA	T	C	D	V	Z
S						*							*		
D													*		

16-bit instruction: none.
32-bit instruction: FDOT continuous execution type (5 steps).
Flag: M2824 zero flag.
Notes on the use of operands: refer to Chapter 1 for the range of each device.
Only the 32-bit instruction FDOT is valid.
Operand D occupies consecutive 2 points.

Instruction description:

S: source device for conversion; D: device for storing the conversion result

1. When M1081 is Off, convert the BIN integer to binary floating-point number. Meanwhile, the S (source device) of the 16-bit instruction FDOT occupies 1 register, and the device D stored with the conversion result occupies 2 registers.
   a. If the conversion result is 0, the zero flag M2824 is On.
2. When M1081 is On, convert the binary floating-point number to BIN integer (decimal places are discarded). Meanwhile, the source device S for the 16-bit instruction FLT instruction occupies 2 registers, and the device D stored with the conversation result

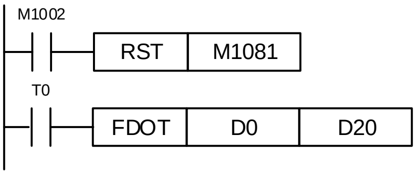
occupies 1 register. The operations are the same with the INT instruction.

a. If S is 0, the M1020 zero flag is On.

b. After the conversion, D saves the data of 16 bits.

Program example 1:

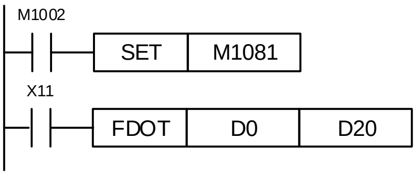
1. When M1081 is Off, convert the BIN integer to binary floating-point number.
2. When X11 is On, convert D1 & D0 (with BIN integers) to D21 & D20 (with binary floating-point numbers).
3. If the 32-bit register D0(D1) = K100,000, then X11 goes On and the 32-bit value of the converted floating-point number is H4735000. This result is saved in the 32-bit register D20(D21).



Program example 2:

1. When M1081 is On, convert the binary floating-point number to BIN integer (decimal places are discarded).
2. When X11 is On, convert D1 & D0 (binary floating-point number) to D21 & D20 (with BIN integers).

   If D0 (D1) = H47C35000, the value of the converted floating-point number is 100,000. This result is saved in 32-bit register D20 (D21).



- **API-65 FRAD: Convert degrees to radians**

<table>
<tr><td>API</td><td colspan="2">-</td><td colspan="2" rowspan="2">FRAD</td><td colspan="4" rowspan="2">S, D</td><td colspan="2" rowspan="2">Convert degrees to radians</td><td colspan="5" rowspan="2">NC series</td></tr>
<tr><td>65</td><td colspan="2"></td></tr>
<tr><td rowspan="2"></td><td colspan="4">Bit device</td><td colspan="11">Word device</td></tr>
<tr><td>X</td><td>Y</td><td>M</td><td>A</td><td>K</td><td>F</td><td>KnX</td><td>KnY</td><td>KnM</td><td>KnA</td><td>T</td><td>C</td><td>D</td><td>V</td><td>Z</td></tr>
<tr><td>S</td><td></td><td></td><td></td><td></td><td></td><td>*</td><td></td><td></td><td></td><td></td><td></td><td></td><td>*</td><td></td><td></td></tr>
<tr><td>D</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td>*</td><td></td><td></td></tr>
<tr><td colspan="16">16-bit instruction: none.
32-bit instruction: FRAD continuous execution type (5 steps).
Flag: M2824 (zero flag).
Notes on the use of operands: refer to Chapter 1 for the range of each device.
Only 32-bit FRAD instruction is valid.</td></tr>
</table>

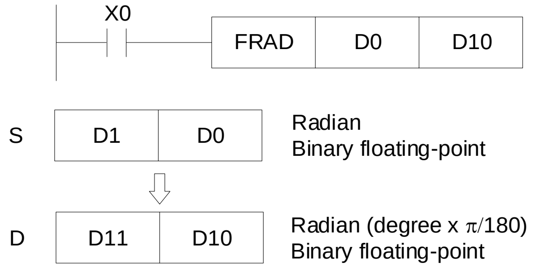
Instruction description:

S: data source (degree); D: conversion result (radian).

1. Use the formula “radian = degree x (π/180)” to convert degrees to radians.
2. If the conversion result is 0, the zero flag M2824 is On.

Program example:

When X0 is On, convert the degrees in binary floating-point number of (D1, D0) to radians, and save the result in (D11, D10) in the data type of binary floating-point number.



■ API-66 FDEG: Convert radians to degrees

API	-	FDEG	S, D	Convert radians to degrees	NC series
66					

	Bit device				Word device										
	X	Y	M	A	K	F	KnX	KnY	KnM	KnA	T	C	D	V	Z
S						∗							∗		
D													∗		

16-bit instruction: none.
32-bit instruction: FDEG continuous execution type (5 steps).
Flag: M2824 (zero flag).
Notes on the use of operands: refer to Chapter 1 for the range of each device.
Only 32-bit FDEG instruction is valid.

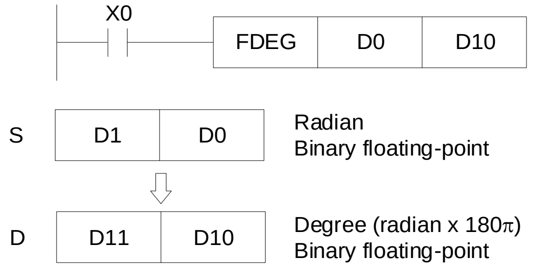
Instruction description:

S: data source (radian); D: conversion result (degree).

1. Use the formula “degree = radian x (180/π)” to convert radians to degrees.
2. If the conversion result is 0, the zero flag M2824 is On.

Program example:

When X0 is On, convert the radians of (D1, D0) in the binary floating-point number format to degrees, and save the result in (D11, D10) in the data type of binary floating-point number.



4.10 NC application instructions

■ API-68 WRTL: Write the servo torque limit

API			WRTL		S, D				Write the servo torque limit		NC series				
68	-														
	Bit device				Word device										
	X	Y	M	A	K	F	KnX	KnY	KnM	KnA	T	C	D	V	Z
S					*								*		
D					*								*		

16-bit instruction: WRTL continuous execution type (4 steps).
32-bit instruction: none.
Flag: none.
Notes on the use of operands: refer to Chapter 1 for the range of each device.

Instruction description:

S: the servo station number to be written; D: the torque limit to be written.

■ API-69 RDTL: Flag for reading the torque limit

API			RDTL		S, D				Flag for reading the torque limit		NC series				
68	-														
	Bit device				Word device										
	X	Y	M	A	K	F	KnX	KnY	KnM	KnA	T	C	D	V	Z
S					*								*		
D		*	*	*									*		

16-bit instruction: RDTL continuous execution type (4 steps).
32-bit instruction: none.
Flag: none.
Notes on the use of operands: refer to Chapter 1 for the range of each device.

Instruction description:

S: the servo station number to be read; D: the destination device for the output.

When the torque value of the station to be read reaches the limit, the output destination Y, M, and A are set to On.

Instructions and Functions of MLC Special M Relay and Special D Register

5

This chapter describes the definitions and functions of all the special M relays and special D registers in the NC system.

5.1 Definition of MLC special M relay and special D register

The MLC (Motion Logic Control) and the NC are two independent systems. The MLC system performs button triggering control, MLC axis movements, and other logic controls, while the NC system manages system and servo axis related functions. The MLC special M relays and D registers serve as the I/O interface between these two systems for data exchange and signal transmission. The output mentioned in this chapter refers to the signals sent to the NC system from the MLC special M relays and D registers. The input refers to the signals sent to the MLC special M relays and D registers from the NC system. The M letter prefixed codes are in bit format referring to signal 0 (OFF) or 1 (ON). The D prefixed codes are in word format referring to numerical values such as 1000. The MLC special M and D codes are all expressed in the form of M- and D- suffixed with four digits.

Data exchanges between the two systems are categorized into four groups.
1: MLC bit output from MLC to NC (special M, bit output)
2: MLC bit input from NC to MLC (special M, bit input)
3: MLC word output from MLC to NC (special D, word output)
4: MLC word input from NC to MLC (special D, word input)

5.2 List of special M and special D

The following tables list the definitions and functions of all special M relays and special D registers in the NC system.

5.2.1 List of special M relays

Function name	Special M	Description	Device type
HMI output point 1	M1024	The status of this special M relay is paired with #1801.	R/W
HMI output point 2	M1025	The status of this special M relay is paired with #1802.	R/W
HMI output point 3	M1026	The status of this special M relay is paired with #1803.	R/W
HMI output point 4	M1027	The status of this special M relay is paired with #1804.	R/W
HMI output point 5	M1028	The status of this special M relay is paired with #1805.	R/W
HMI output point 6	M1029	The status of this special M relay is paired with #1806.	R/W
HMI output point 7	M1030	The status of this special M relay is paired with #1807.	R/W
HMI output point 8	M1031	The status of this special M relay is paired with #1808.	R/W
HMI output point 9	M1032	The status of this special M relay is paired with #1809.	R/W
HMI output point 10	M1033	The status of this special M relay is paired with #1810.	R/W
HMI output point 11	M1034	The status of this special M relay is paired with #1811.	R/W
HMI output point 12	M1035	The status of this special M relay is paired with #1812.	R/W
HMI output point 13	M1036	The status of this special M relay is paired with #1813.	R/W
HMI output point 14	M1037	The status of this special M relay is paired with #1814.	R/W
HMI output point 15	M1038	The status of this special M relay is paired with #1815.	R/W
HMI output point 16	M1039	The status of this special M relay is paired with #1816.	R/W
HMI output point 17	M1040	The status of this special M relay is paired with #1817.	R/W
HMI output point 18	M1041	The status of this special M relay is paired with #1818.	R/W
HMI output point 19	M1042	The status of this special M relay is paired with #1819.	R/W
HMI output point 20	M1043	The status of this special M relay is paired with #1820.	R/W
HMI output point 21	M1044	The status of this special M relay is paired with #1821.	R/W
HMI output point 22	M1045	The status of this special M relay is paired with #1822.	R/W
HMI output point 23	M1046	The status of this special M relay is paired with #1823.	R/W
HMI output point 24	M1047	The status of this special M relay is paired with #1824.	R/W
HMI output point 25	M1048	The status of this special M relay is paired with #1825.	R/W
HMI output point 26	M1049	The status of this special M relay is paired with #1826.	R/W
HMI output point 27	M1050	The status of this special M relay is paired with #1827.	R/W
HMI output point 28	M1051	The status of this special M relay is paired with #1828.	R/W
HMI output point 29	M1052	The status of this special M relay is paired with #1829.	R/W
HMI output point 30	M1053	The status of this special M relay is paired with #1830.	R/W
HMI output point 31	M1054	The status of this special M relay is paired with #1831.	R/W
HMI output point 32	M1055	The status of this special M relay is paired with #1832.	R/W

<table>
<tr><th>Function name</th><th>Special M</th><th>Description</th><th>Device type</th></tr>
<tr><td>System mode selection:
0: auto execution
1: program edit
2: manual input
3: MPG feeding
4: jog feeding
5: rapid feeding
6: homing</td><td>M1056
M1057
M1058
M1059</td><td>You can use M1056 - M1059 to switch between the NC system modes.
<table>
<tr><th colspan="4">Binary</th><th rowspan="2">Decimal</th><th rowspan="2">NC system mode</th></tr>
<tr><th>M1059 (Bit 3)</th><th>M1058 (Bit 2)</th><th>M1057 (Bit 1)</th><th>M1056 (Bit 0)</th></tr>
<tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>AUTO</td></tr>
<tr><td>0</td><td>0</td><td>0</td><td>1</td><td>1</td><td>EDIT</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>0</td><td>2</td><td>MDI</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>1</td><td>3</td><td>MPG</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>0</td><td>4</td><td>JOG</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>1</td><td>5</td><td>RAPID</td></tr>
<tr><td>0</td><td>1</td><td>1</td><td>0</td><td>6</td><td>HOME</td></tr>
</table></td><td>R/W</td></tr>
<tr><td>Single block execution</td><td>M1060</td><td>In AUTO mode, program stops after one block is executed.</td><td>R/W</td></tr>
<tr><td>Cycle Start</td><td>M1061</td><td>Informs the NC system to execute Cycle Start.</td><td>R/W</td></tr>
<tr><td>NC pause</td><td>M1062</td><td>The NC controller pauses right after M1062 is triggered.</td><td>R/W</td></tr>
<tr><td>System stop</td><td>M1063</td><td>The system stops operating.</td><td>R/W</td></tr>
<tr><td>NC system reset</td><td>M1064</td><td>After M1076 or the reset signal of machine operation panel A is triggered, the system will reset, and M1064 is triggered.</td><td>R</td></tr>
<tr><td>Dummy execution</td><td>M1065</td><td>After M1065 is triggered, the moving speed F of G01 in AUTO mode refers to the feedrate specified in the D1062 register.</td><td>R/W</td></tr>
<tr><td>Optional stop
(M01 program stop)</td><td>M1066</td><td>Optional stop key. When the program executes M01, the controller immediately stops.</td><td>R/W</td></tr>
<tr><td>Single block skip (‘/’)</td><td>M1067</td><td>The program skips the block containing the symbol '/' when this function is enabled.</td><td>R/W</td></tr>
<tr><td>Lock all axes movements</td><td>M1068</td><td>Locks the X, Y, and Z axes movements of the machine.</td><td>R/W</td></tr>
<tr><td>Lock Z axis movement</td><td>M1069</td><td>Locks the Z axis movement of the machine.</td><td>R/W</td></tr>
<tr><td>Ignore axis limit</td><td>M1070</td><td>The limit signal of each axis is ignored when this function is enabled.</td><td>R/W</td></tr>
<tr><td>Lock M, S, and T codes</td><td>M1071</td><td>Locks M, S, and T codes. The program skips M, S, and T codes in the execution.</td><td>R/W</td></tr>
<tr><td>DMCNET connection successful</td><td>M1072</td><td>The MLC sends this signal after the NC system confirms that the DMCNET connection is successful. Note that this signal signifies the connection is successful instead of the Servo On status.</td><td>R</td></tr>
<tr><td>Macro call initial preparation</td><td>M1074</td><td>Macro call initialization. (Only works with the correct macro ID in AUTO mode)</td><td>R/W</td></tr>
<tr><td>Macro call activation</td><td>M1075</td><td>Activates macro call. (Only works with the correct macro ID in AUTO mode)</td><td>R/W</td></tr>
<tr><td>NC system reset</td><td>M1076</td><td>When M1076 is triggered, the NC system resets (MLC > NC).</td><td>R/W</td></tr>
<tr><td>Main program stops at M99</td><td>M1077</td><td>When Pr308 [Bit 9] is set to 1 and M1077 is set to on, the NC system stops machining when it reads M99.</td><td>R/W</td></tr>
<tr><td>M96 program interruption</td><td>M1078</td><td>After the NC executes M96, if M1078 is triggered, the NC system interrupts the main program and jumps to the subprogram to execute it.</td><td>R/W</td></tr>
<tr><td>MLC emergency stop</td><td>M1079</td><td>When M1079 is set to on, the NC system triggers the emergency stop.</td><td>R/W</td></tr>
<tr><td>MPG simulation</td><td>M1080</td><td>During program execution, you can use the MPG to control the speed of movement trajectories.</td><td>R/W</td></tr>
</table>

Function name	Special M	Description	Device type
Disable inhibit zone protection	M1085	When M1085 is triggered, the inhibit zone protection is disabled.	R/W
Trigger for synchronous control	M1088	To use the synchronous control function, you have to set M1088 to on so the NC system can enable the function.	R/W
X slave axis follows the master axis	M1089	Sets the X axis as the slave axis for synchronous control.	R/W
Y slave axis follows the master axis	M1090	Sets the Y axis as the slave axis for synchronous control.	R/W
Z slave axis follows the master axis	M1091	Sets the Z axis as the slave axis for synchronous control.	R/W
A slave axis follows the master axis	M1092	Sets the A axis as the slave axis for synchronous control.	R/W
B slave axis follows the master axis	M1093	Sets the B axis as the slave axis for synchronous control.	R/W
C slave axis follows the master axis	M1094	Sets the C axis as the slave axis for synchronous control.	R/W
U slave axis follows the master axis	M1095	Sets the U axis as the slave axis for synchronous control.	R/W
V slave axis follows the master axis	M1096	Sets the V axis as the slave axis for synchronous control.	R/W
W slave axis follows the master axis	M1097	Sets the W axis as the slave axis for synchronous control.	R/W
Trigger for command transfer	M1098	To use the function of command transfer, you have to set M1098 to on so the NC system can enable the function.	R/W
X axis receives command from the master axis	M1099	Specifies the X axis as the axis to receive the transfer command.	R/W
Y axis receives command from the master axis	M1100	Specifies the Y axis as the axis to receive the transfer command.	R/W
Z axis receives command from the master axis	M1101	Specifies the Z axis as the axis to receive the transfer command.	R/W
A axis receives command from the master axis	M1102	Specifies the A axis as the axis to receive the transfer command.	R/W
B axis receives command from the master axis	M1103	Specifies the B axis as the axis to receive the transfer command.	R/W
C axis receives command from the master axis	M1104	Specifies the C axis as the axis to receive the transfer command.	R/W
U axis receives command from the master axis	M1105	Specifies the U axis as the axis to receive the transfer command.	R/W
V axis receives command from the master axis	M1106	Specifies the V axis as the axis to receive the transfer command.	R/W
W axis receives command from the master axis	M1107	Specifies the W axis as the axis to receive the transfer command.	R/W
Panel MPG pulse +	M1118	This is the forward pulse signal when you use the keys on machine operation panel B as the MPG function. Refer to the description of D1040 for the enabling method.	R/W

Function name	Special M	Description	Device type
Panel MPG pulse -	M1119	This is the reverse pulse signal when you use the keys on machine operation panel B as the MPG function. Refer to the description of D1040 for the enabling method.	R/W
1st spindle forward rotation	M1120	Sets the 1st spindle to rotate in forward direction.	R/W
1st spindle reverse rotation	M1121	Sets the 1st spindle to rotate in reverse direction.	R/W
1st spindle gear ratio selection	M1122 M1123	The 1st spindle gear ratio is selected by the combination of the two bits M1122 and M1123, which corresponds to 0 - 3 in decimal, representing the four sets of gear ratio (Pr422 - Pr429). For example, to use the settings of “Gear ratio numerator 4 (Pr428) and “Gear ratio denominator 4 (Pr429)”, the setting value is 3 (0011 in binary), and the corresponding two bits in MLC are M1122 and M1123. The bit settings are as follows. M1122 = ON, M1123 = ON	R/W
1st spindle positioning control	M1124	Positions the 1st spindle.	R/W
1st spindle retraction after tapping	M1125	Retracts the 1st spindle after tapping.	R/W
Lathe spindle C / S axis switching	M1126	Trigger M1126 to switch between C and S axes for the lathe spindle. Note: this function is only available for lathe.	R/W
1st spindle analog voltage proportional gain	M1127	Select the parameter which the 1st spindle analog voltage proportional gain refers to with M1127. When M1127 is on, refer to Pr413. When M1127 is off, refer to Pr419.	R/W
2nd spindle forward rotation	M1136	Sets the 2nd spindle to rotate in forward direction.	R/W
2nd spindle reverse rotation	M1137	Sets the 2nd spindle to rotate in reverse direction.	R/W
2nd spindle gear ratio selection	M1138 M1139	The 2nd spindle gear ratio is selected by the combination of the two bits M1138 and M1139, which corresponds to 0 - 3 in decimal, representing the four sets of gear ratio (Pr462 - Pr469). For example, to use the settings of “2nd spindle gear ratio numerator 4 (Pr468) and “2nd spindle gear ratio denominator 4 (Pr469)”, the setting value is 3 (0011 in binary), and the corresponding two bits in MLC are M1138 and M1139. The bit settings are as follows. M1138 = ON, M1139 = ON	R/W
2nd spindle positioning control	M1140	Positions the 2nd spindle.	R/W
2nd spindle retraction after tapping	M1141	Retracts the 2nd spindle after tapping.	R/W
2nd spindle analog voltage proportional gain	M1143	Select the parameter which the 2nd spindle analog voltage proportional gain refers to with M1143. When M1143 is on, refer to Pr453. When M1143 is off, refer to Pr459.	R/W
Notification of MST code execution complete	M1152	When M1152 is triggered, this signal informs the NC system that the M, S or T code execution is complete.	R/W
Tool magazine 1 moves forward	M1168	Moves tool magazine 1 forward. When this special M relay is triggered, the tool pot deviation (D1372) minus 1, and the standby tool pot number (D1373) adds 1.	R/W

Function name	Special M	Description	Device type
Tool magazine 1 moves backward	M1169	Moves tool magazine 1 backward. When this special M relay is triggered, the tool pot deviation (D1372) adds 1, and the standby tool pot number (D1373) minus 1.	R/W
Tool 1 exchange	M1170	Exchanges tool data in tool magazine 1.	R/W
Tool magazine 1 reset	M1171	When this special M relay is triggered, the tool data in tool magazine 1 is reset.	R/W
Tool magazine 2 moves forward	M1172	Moves tool magazine 2 forward. When this special M is triggered, the tool pot deviation (D1376) minus 1, and the standby tool pot number (D1377) adds 1.	R/W
Tool magazine 2 moves backward	M1173	Moves tool magazine 2 backward. When this special M relay is triggered, the tool pot deviation (D1376) adds 1, and the standby tool pot (D1377) minus 1.	R/W
Tool 2 exchange	M1174	Exchanges tool data in tool magazine 2.	R/W
Tool magazine 2 reset	M1175	When this special M relay is triggered, the tool data in tool magazine 2 is reset.	R/W
Activate X axis (MLC axis)	M1184	Trigger for MLC X axis activation.	R/W
Activate Y axis (MLC axis)	M1185	Trigger for MLC Y axis activation.	R/W
Activate Z axis (MLC axis)	M1186	Trigger MLC Z axis activation.	R/W
Activate A axis (MLC axis)	M1187	Trigger for MLC A axis activation.	R/W
Activate B axis (MLC axis)	M1188	Trigger for MLC B axis activation.	R/W
Activate C axis (MLC axis)	M1189	Trigger for MLC C axis activation.	R/W
Activate U axis (MLC axis)	M1190	Trigger for MLC U axis activation.	R/W
Activate V axis (MLC axis)	M1191	Trigger for MLC V axis activation.	R/W
Activate W axis (MLC axis)	M1192	Trigger for MLC W axis activation.	R/W
Activate the spindle (MLC axis)	M1193	Trigger MLC spindle activation.	R/W
MLC axis incremental motion command	M1194	When this special M relay is triggered, the system regards the special D positioning commands specified by each MLC axis as incremental commands.	R/W
NC / MLC axis switching (X axis)	M1200	M1200 = on, MLC axis; M1200 = off, NC axis.	R/W
NC / MLC axis switching (Y axis)	M1201	M1201 = on, MLC axis; M1201 = off, NC axis.	R/W
NC / MLC axis switching (Z axis)	M1202	M1202 = on, MLC axis; M1202 = off, NC axis.	R/W
NC / MLC axis switching (A axis)	M1203	M1203 = on, MLC axis; M1203 = off, NC axis.	R/W
NC / MLC axis switching (B axis)	M1204	M1204 = on, MLC axis; M1204 = off, NC axis.	R/W
NC / MLC axis switching (C axis)	M1205	M1205 = on, MLC axis; M1205 = off, NC axis.	R/W
NC / MLC axis switching (U axis)	M1206	M1206 = on, MLC axis; M1206 = off, NC axis.	R/W
NC / MLC axis switching (V axis)	M1207	M1207 = on, MLC axis; M1207 = off, NC axis.	R/W
NC / MLC axis switching (W axis)	M1208	M1208 = on, MLC axis; M1208 = off, NC axis.	R/W

Function name	Special M	Description	Device type
X axis forward jog control	M1216	Trigger for X axis forward jog operation.	R/W
Y axis forward jog control	M1217	Trigger for Y axis forward jog operation.	R/W
Z axis forward jog control	M1218	Trigger for Z axis forward jog operation.	R/W
A axis forward jog control	M1219	Trigger for A axis forward jog operation.	R/W
B axis forward jog control	M1220	Trigger for B axis forward jog operation.	R/W
C axis forward jog control	M1221	Trigger for C axis forward jog operation.	R/W
U axis forward jog control	M1222	Trigger for U axis forward jog operation.	R/W
V axis forward jog control	M1223	Trigger for V axis forward jog operation.	R/W
W axis forward jog control	M1224	Trigger for W axis forward jog operation.	R/W
X axis reverse jog control	M1226	Trigger for X axis reverse jog operation.	R/W
Y axis reverse jog control	M1227	Trigger for Y axis reverse jog operation.	R/W
Z axis reverse jog control	M1228	Trigger for Z axis reverse jog operation.	R/W
A axis reverse jog control	M1229	Trigger for A axis reverse jog operation.	R/W
B axis reverse jog control	M1230	Trigger for B axis reverse jog operation.	R/W
C axis reverse jog control	M1231	Trigger for C axis reverse jog operation.	R/W
U axis reverse jog control	M1232	Trigger for U axis reverse jog operation.	R/W
V axis reverse jog control	M1233	Trigger for V axis reverse jog operation.	R/W
W axis reverse jog control	M1234	Trigger for W axis reverse jog operation.	R/W
X axis homing control	M1236	Trigger for X axis homing.	R/W
Y axis homing control	M1237	Trigger for Y axis homing.	R/W
Z axis homing control	M1238	Trigger for Z axis homing.	R/W
A axis homing control	M1239	Trigger for A axis homing.	R/W
B axis homing control	M1240	Trigger for B axis homing.	R/W
C axis homing control	M1241	Trigger for C axis homing.	R/W
U axis homing control	M1242	Trigger for U axis homing.	R/W
V axis homing control	M1243	Trigger for V axis homing.	R/W
W axis homing control	M1244	Trigger for W axis homing.	R/W
Cancel X axis 1st software limit	M1248	Trigger for canceling X axis 1st software limit.	R/W
Cancel Y axis 1st software limit	M1249	Trigger for canceling Y axis 1st software limit.	R/W
Cancel Z axis 1st software limit	M1250	Trigger for canceling Z axis 1st software limit.	R/W
Cancel A axis 1st software limit	M1251	Trigger for canceling A axis 1st software limit.	R/W

Function name	Special M	Description	Device type
Cancel B axis 1st software limit	M1252	Trigger for canceling B axis 1st software limit.	R/W
Cancel C axis 1st software limit	M1253	Trigger for canceling C axis 1st software limit.	R/W
Cancel U axis 1st software limit	M1254	Trigger for canceling U axis 1st software limit.	R/W
Cancel V axis 1st software limit	M1255	Trigger for canceling V axis 1st software limit.	R/W
Cancel W axis 1st software limit	M1256	Trigger for canceling W axis 1st software limit.	R/W
Lock X axis	M1257	Trigger for locking X axis.	R/W
Lock Y axis	M1258	Trigger for locking Y axis.	R/W
Lock Z axis	M1259	Trigger for locking Z axis.	R/W
Lock A axis	M1260	Trigger for locking A axis.	R/W
Lock B axis	M1261	Trigger for locking B axis.	R/W
Lock C axis	M1262	Trigger for locking C axis.	R/W
Lock U axis	M1263	Trigger for locking U axis.	R/W
Lock V axis	M1264	Trigger for locking V axis.	R/W
Lock W axis	M1265	Trigger for locking W axis.	R/W
X axis Servo Off	M1266	Trigger this special M relay to set the X axis to Servo Off status.	R/W
Y axis Servo Off	M1267	Trigger this special M relay to set the Y axis to Servo Off status.	R/W
Z axis Servo Off	M1268	Trigger this special M relay to set the Z axis to Servo Off status.	R/W
A axis Servo Off	M1269	Trigger this special M relay to set the A axis to Servo Off status.	R/W
B axis Servo Off	M1270	Trigger this special M relay to set the B axis to Servo Off status.	R/W
C axis Servo Off	M1271	Trigger this special M relay to set the C axis to Servo Off status.	R/W
U axis Servo Off	M1272	Trigger this special M relay to set the U axis to Servo Off status.	R/W
V axis Servo Off	M1273	Trigger this special M relay to set the V axis to Servo Off status.	R/W
W axis Servo Off	M1274	Trigger this special M relay to set the W axis to Servo Off status.	R/W
Switch MLC X axis command to incremental	M1280	When this special M relay is triggered, the MLC X axis positioning command (D1064) is incremental.	R/W
Switch MLC Y axis command to incremental	M1281	When this special M relay is triggered, the MLC Y axis positioning command (D1066) is incremental.	R/W
Switch MLC Z axis command to incremental	M1282	When the special M relay is triggered, the MLC Z axis positioning command (D1068) is incremental.	R/W
Switch MLC A axis command to incremental	M1283	When the special M relay is triggered, the MLC A axis positioning command (D1070) is incremental.	R/W
Switch MLC B axis command to incremental	M1284	When the special M relay is triggered, the MLC B axis positioning command (D1072) is incremental.	R/W
Switch MLC C axis command to incremental	M1285	When the special M relay is triggered, the MLC C axis positioning command (D1074) is incremental.	R/W

Function name	Special M	Description	Device type
Switch MLC U axis command to incremental	M1286	When the special M relay is triggered, the MLC U axis positioning command (D1076) is incremental.	R/W
Switch MLC V axis command to incremental	M1287	When the special M relay is triggered, the MLC V axis positioning command (D1078) is incremental.	R/W
Switch MLC W axis command to incremental	M1288	When the special M relay is triggered, the MLC W axis positioning command (D1080) is incremental.	R/W
MLC X axis control mode	M1289	When the special M relay is on, X axis is in speed mode; when the special M relay is off, X axis is in position mode.	R/W
MLC Y axis control mode	M1290	When the special M relay is on, Y axis is in speed mode; when the special M relay is off, Y axis is in position mode.	R/W
MLC Z axis control mode	M1291	When the special M relay is on, Z axis is in speed mode; when the special M relay is off, Z axis is in position mode.	R/W
MLC A axis control mode	M1292	When the special M relay is on, A axis is in speed mode; when the special M relay is off, A axis is in position mode.	R/W
MLC B axis control mode	M1293	When the special M relay is on, B axis is in speed mode; when the special M relay is off, B axis is in position mode.	R/W
MLC C axis control mode	M1294	When the special M relay is on, C axis is in speed mode; when the special M relay is off, C axis is in position mode.	R/W
MLC U axis control mode	M1295	When the special M relay is on, U axis is in speed mode; when the special M relay is off, U axis is in position mode.	R/W
MLC V axis control mode	M1296	When the special M relay is on, V axis is in speed mode; when the special M relay is off, V axis is in position mode.	R/W
MLC W axis control mode	M1297	When the special M relay is on, W axis is in speed mode; when the special M relay is off, W axis is in position mode.	R/W
MLC X axis high-speed input point triggering	M1298	When you set this special M relay to on during MLC axis mode and trigger the corresponding high-speed input point of X axis, the MLC X axis stops immediately.	R/W
MLC Y axis high-speed input point triggering	M1299	When you set this special M relay to on during MLC axis mode and trigger the corresponding high-speed input point of Y axis, the MLC Y axis stops immediately.	R/W
MLC Z axis high-speed input point triggering	M1300	When you set this special M relay to on during MLC axis mode and trigger the corresponding high-speed input point of Z axis, the MLC Z axis stops immediately.	R/W
MLC A axis high-speed input point triggering	M1301	When you set this special M relay to on during MLC axis mode and trigger the corresponding high-speed input point of A axis, the MLC A axis stops immediately.	R/W
MLC B axis high-speed input point triggering	M1302	When you set this special M relay to on during MLC axis mode and trigger the corresponding high-speed input point of B axis, the MLC B axis stops immediately.	R/W
MLC C axis high-speed input point triggering	M1303	When you set this special M relay to on during MLC axis mode and trigger the corresponding high-speed input point of C axis, the MLC C axis stops immediately.	R/W
MLC U axis high-speed input point triggering	M1304	When you set this special M relay to on during MLC axis mode and trigger the corresponding high-speed input point of U axis, the MLC U axis stops immediately.	R/W

Function name	Special M	Description	Device type
MLC V axis high-speed input point triggering	M1305	When you set this special M relay to on during MLC axis mode and trigger the corresponding high-speed input point of V axis, the MLC V axis stops immediately.	R/W
MLC W axis high-speed input point triggering	M1306	When you set this special M relay to on during MLC axis mode and trigger the corresponding high-speed input point of W axis, the MLC W axis stops immediately.	R/W
Switch the spindle speed source	M1307	When this special M relay is on, the 1st and 2nd spindle speed refer to D1148 and D1152 respectively. When this special M relay is off, the spindle speed refers to the S code in the program.	R/W
Permit X axis movement	M1312	When Pr501 (Axis movement protection) is set to 1, you have to set this special M relay to on during machining for the X axis to move.	R/W
Permit Y axis movement	M1313	When Pr501 (Axis movement protection) is set to 1, you have to set this special M relay to on during machining for the Y axis to move.	R/W
Permit Z axis movement	M1314	When Pr501 (Axis movement protection) is set to 1, you have to set this special M relay to on during machining for the Z axis to move.	R/W
Permit A axis movement	M1315	When Pr501 (Axis movement protection) is set to 1, you have to set this special M relay to on during machining for the A axis to move.	R/W
Permit B axis movement	M1316	When Pr501 (Axis movement protection) is set to 1, you have to set this special M relay to on during machining for the B axis to move.	R/W
Permit C axis movement	M1317	When Pr501 (Axis movement protection) is set to 1, you have to set this special M relay to on during machining for the C axis to move.	R/W
Permit U axis movement	M1318	When Pr501 (Axis movement protection) is set to 1, you have to set this special M relay to on during machining for the U axis to move.	R/W
Permit V axis movement	M1319	When Pr501 (Axis movement protection) is set to 1, you have to set this special M relay to on during machining for the V axis to move.	R/W
Permit W axis movement	M1320	When Pr501 (Axis movement protection) is set to 1, you have to set this special M relay to on during machining for the W axis to move.	R/W
Lock machine X axis movement in positive direction	M1344	When Pr485 [Bit 5] is set to 1 (Lock axis moving direction in AUTO mode), in AUTO and MDI modes, setting the special M relay to on disables X axis from moving in the positive direction. In JOG, MPG, and RAPID modes, setting this special M relay to on disables X axis from moving in the positive direction whether Pr485 is set or not. In HOME mode, setting this special M relay to on disables X axis from executing homing.	R/W
Lock machine Y axis movement in positive direction	M1345	When Pr485 [Bit 5] is set to 1 (Lock axis moving direction in AUTO mode), in AUTO and MDI modes, setting the special M relay to on disables Y axis from moving in the positive direction. In JOG, MPG, and RAPID modes, setting this special M relay to on disables Y axis from moving in the positive direction whether Pr485 is set or not. In HOME mode, setting this special M relay to on disables Y axis from executing homing.	R/W

Function name	Special M	Description	Device type
Lock machine Z axis movement in positive direction	M1346	When Pr485 [Bit 5] is set to 1 (Lock axis moving direction in AUTO mode), in AUTO and MDI modes, setting the special M relay to on disables Z axis from moving in the positive direction. In JOG, MPG, and RAPID modes, setting this special M relay to on disables Z axis from moving in the positive direction whether Pr485 is set or not. In HOME mode, setting this special M relay to on disables Z axis from executing homing.	R/W
Lock machine A axis movement in positive direction	M1347	When Pr485 [Bit 5] is set to 1 (Lock axis moving direction in AUTO mode), in AUTO and MDI modes, setting the special M relay to on disables A axis from moving in the positive direction. In JOG, MPG, and RAPID modes, setting this special M relay to on disables A axis from moving in the positive direction whether Pr485 is set or not. In HOME mode, setting this special M relay to on disables A axis from executing homing.	R/W
Lock machine B axis movement in positive direction	M1348	When Pr485 [Bit 5] is set to 1 (Lock axis moving direction in AUTO mode), in AUTO and MDI modes, setting the special M relay to on disables B axis from moving in the positive direction. In JOG, MPG, and RAPID modes, setting this special M relay to on disables B axis from moving in the positive direction whether Pr485 is set or not. In HOME mode, setting this special M relay to on disables B axis from executing homing.	R/W
Lock machine C axis movement in positive direction	M1349	When Pr485 [Bit 5] is set to 1 (Lock axis moving direction in AUTO mode), in AUTO and MDI modes, setting the special M relay to on disables C axis from moving in the positive direction. In JOG, MPG, and RAPID modes, setting this special M relay to on disables C axis from moving in the positive direction whether Pr485 is set or not. In HOME mode, setting this special M relay to on disables C axis from executing homing.	R/W
Lock machine U axis movement in positive direction	M1350	When Pr485 [Bit 5] is set to 1 (Lock axis moving direction in AUTO mode), in AUTO and MDI modes, setting the special M relay to on disables U axis from moving in the positive direction. In JOG, MPG, and RAPID modes, setting this special M relay to on disables U axis from moving in the positive direction whether Pr485 is set or not. In HOME mode, setting this special M relay to on disables U axis from executing homing.	R/W
Lock machine V axis movement in positive direction	M1351	When Pr485 [Bit 5] is set to 1 (Lock axis moving direction in AUTO mode), in AUTO and MDI modes, setting the special M relay to on disables V axis from moving in the positive direction. In JOG, MPG, and RAPID modes, setting this special M relay to on disables V axis from moving in the positive direction whether Pr485 is set or not. In HOME mode, setting this special M relay to on disables V axis from executing homing.	R/W
Lock machine W axis movement in positive direction	M1352	When Pr485 [Bit 5] is set to 1 (Lock axis moving direction in AUTO mode), in AUTO and MDI modes, setting the special M relay to on disables W axis from moving in the positive direction. In JOG, MPG, and RAPID modes, setting this special M relay to on disables W axis from moving in the positive direction whether Pr485 is set or not. In HOME mode, setting this special M relay to on disables W axis from executing homing.	R/W

Function name	Special M	Description	Device type
Lock machine X axis movement in negative direction	M1353	When Pr485 [Bit 5] is set to 1 (Lock axis moving direction in AUTO mode), in AUTO and MDI modes, setting the special M relay to on disables X axis from moving in the negative direction. In JOG, MPG, and RAPID modes, setting this special M relay to on disables X axis from moving in the negative direction whether Pr485 is set or not. In HOME mode, setting this special M relay to on disables X axis from executing homing.	R/W
Lock machine Y axis movement in negative direction	M1354	When Pr485 [Bit 5] is set to 1 (Lock axis moving direction in AUTO mode), in AUTO and MDI modes, setting the special M relay to on disables Y axis from moving in the negative direction. In JOG, MPG, and RAPID modes, setting this special M relay to on disables Y axis from moving in the negative direction whether Pr485 is set or not. In HOME mode, setting this special M relay to on disables Y axis from executing homing.	R/W
Lock machine Z axis movement in negative direction	M1355	When Pr485 [Bit 5] is set to 1 (Lock axis moving direction in AUTO mode), in AUTO and MDI modes, setting the special M relay to on disables Z axis from moving in the negative direction. In JOG, MPG, and RAPID modes, setting this special M relay to on disables Z axis from moving in the negative direction whether Pr485 is set or not. In HOME mode, setting this special M relay to on disables Z axis from executing homing.	R/W
Lock machine A axis movement in negative direction	M1356	When Pr485 [Bit 5] is set to 1 (Lock axis moving direction in AUTO mode), in AUTO and MDI modes, setting the special M relay to on disables A axis from moving in the negative direction. In JOG, MPG, and RAPID modes, setting this special M relay to on disables A axis from moving in the negative direction whether Pr485 is set or not. In HOME mode, setting this special M relay to on disables A axis from executing homing.	R/W
Lock machine B axis movement in negative direction	M1357	When Pr485 [Bit 5] is set to 1 (Lock axis moving direction in AUTO mode), in AUTO and MDI modes, setting the special M relay to on disables B axis from moving in the negative direction. In JOG, MPG, and RAPID modes, setting this special M relay to on disables B axis from moving in the negative direction whether Pr485 is set or not. In HOME mode, setting this special M relay to on disables B axis from executing homing.	R/W
Lock machine C axis movement in negative direction	M1358	When Pr485 [Bit 5] is set to 1 (Lock axis moving direction in AUTO mode), in AUTO and MDI modes, setting the special M relay to on disables C axis from moving in the negative direction. In JOG, MPG, and RAPID modes, setting this special M relay to on disables C axis from moving in the negative direction whether Pr485 is set or not. In HOME mode, setting this special M relay to on disables C axis from executing homing.	R/W
Lock machine U axis movement in negative direction	M1359	When Pr485 [Bit 5] is set to 1 (Lock axis moving direction in AUTO mode), in AUTO and MDI modes, setting the special M relay to on disables U axis from moving in the negative direction. In JOG, MPG, and RAPID modes, setting this special M relay to on disables U axis from moving in the negative direction whether Pr485 is set or not. In HOME mode, setting this special M relay to on disables U axis from executing homing.	R/W

Function name	Special M	Description	Device type
Lock machine V axis movement in negative direction	M1360	When Pr485 [Bit 5] is set to 1 (Lock axis moving direction in AUTO mode), in AUTO and MDI modes, setting the special M relay to on disables V axis from moving in the negative direction. In JOG, MPG, and RAPID modes, setting this special M relay to on disables V axis from moving in the negative direction whether Pr485 is set or not. In HOME mode, setting this special M relay to on disables V axis from executing homing.	R/W
Lock machine W axis movement in negative direction	M1361	When Pr485 [Bit 5] is set to 1 (Lock axis moving direction in AUTO mode), in AUTO and MDI modes, setting the special M relay to on disables W axis from moving in the negative direction. In JOG, MPG, and RAPID modes, setting this special M relay to on disables W axis from moving in the negative direction whether Pr485 is set or not. In HOME mode, setting this special M relay to on disables W axis from executing homing.	R/W
HMI input point 1	M2080	The status of this special M relay can be switched by the NC system variable #1864.	R
HMI input point 2	M2081	The status of this special M relay can be switched by the NC system variable #1865.	R
HMI input point 3	M2082	The status of this special M relay can be switched by the NC system variable #1866.	R
HMI input point 4	M2083	The status of this special M can be switched by the NC system variable #1867.	R
HMI input point 5	M2084	The status of this special M can be switched by the NC system variable #1868.	R
HMI input point 6	M2085	The status of this special M can be switched by the NC system variable #1869.	R
HMI input point 7	M2086	The status of this special M can be switched by the NC system variable #1870.	R
HMI input point 8	M2087	The status of this special M can be switched by the NC system variable #1871.	R
HMI input point 9	M2088	The status of this special M can be switched by the NC system variable #1872.	R
HMI input point 10	M2089	The status of this special M can be switched by the NC system variable #1873.	R
HMI input point 11	M2090	The status of this special M can be switched by the NC system variable #1874.	R
HMI input point 12	M2091	The status of this special M can be switched by the NC system variable #1875.	R
HMI input point 13	M2092	The status of this special M can be switched by the NC system variable #1876.	R
HMI input point 14	M2093	The status of this special M can be switched by the NC system variable #1877.	R
HMI input point 15	M2094	The status of this special M can be switched by the NC system variable #1878.	R
HMI input point 16	M2095	The status of this special M can be switched by the NC system variable #1879.	R
HMI input point 17	M2096	The status of this special M can be switched by the NC system variable #1880.	R
HMI input point 18	M2097	The status of this special M can be switched by the NC system variable #1881.	R
HMI input point 19	M2098	The status of this special M can be switched by the NC system variable #1882.	R
HMI input point 20	M2099	The status of this special M can be switched by the NC system variable #1883.	R

Function name	Special M	Description	Device type
HMI input point 21	M2100	The status of this special M can be switched by the NC system variable #1884.	R
HMI input point 22	M2101	The status of this special M can be switched by the NC system variable #1885.	R
HMI input point 23	M2102	The status of this special M can be switched by the NC system variable #1886.	R
HMI input point 24	M2103	The status of this special M can be switched by the NC system variable #1887.	R
HMI input point 25	M2104	The status of this special M can be switched by the NC system variable #1888.	R
HMI input point 26	M2105	The status of this special M can be switched by the NC system variable #1889.	R
HMI input point 27	M2106	The status of this special M can be switched by the NC system variable #1890.	R
HMI input point 28	M2107	The status of this special M can be switched by the NC system variable #1891.	R
HMI input point 29	M2108	The status of this special M can be switched by the NC system variable #1892.	R
HMI input point 30	M2109	The status of this special M can be switched by the NC system variable #1893.	R
HMI input point 31	M2110	The status of this special M can be switched by the NC system variable #1894.	R
HMI input point 32	M2111	The status of this special M can be switched by the NC system variable #1895.	R
Machine started and NC system ready	M2112	When the NC system is ready, it sets this special M relay to on.	R
NC system macro alarm occurs	M2113	When the macro alarm (MR_) occurs in the NC system, the NC sets this special M relay to on.	R
NC system emergency stop	M2114	After you press the EMERGENCY STOP key to stop the system immediately, the system sets this special M relay to on.	R
Servo enabled	M2115	When the servo is ready, the NC sets this special M relay to on.	R
X axis at the origin	M2119	When the machine coordinate of X axis is 0, the NC sets this special M relay to on.	R
Y axis at the origin	M2120	When the machine coordinate of Y axis is 0, the NC sets this special M relay to on.	R
Z axis at the origin	M2121	When the machine coordinate of Z axis is 0, the NC sets this special M relay to on.	R
A axis at the origin	M2122	When the machine coordinate of A axis is 0, the NC sets this special M relay to on.	R
B axis at the origin	M2123	When the machine coordinate of B axis is 0, the NC sets this special M relay to on.	R
C axis at the origin	M2124	When the machine coordinate of C axis is 0, the NC sets this special M relay to on.	R
U axis at the origin	M2125	When the machine coordinate of U axis is 0, the NC sets this special M relay to on.	R
V axis at the origin	M2126	When the machine coordinate of V axis is 0, the NC sets this special M relay to on.	R
W axis at the origin	M2127	When the machine coordinate of W axis is 0, the NC sets this special M relay to on.	R
MLC X axis high-speed input point triggering feedback	M2128	When the MLC X axis high-speed input point triggering special M relay (M1298) is set to on and the DI of X axis is triggered, the NC sets this special M relay to on.	R

Function name	Special M	Description	Device type
MLC Y axis high-speed input point triggering feedback	M2129	When the MLC Y axis high-speed input point triggering special M relay (M1299) is set to on and the DI of Y axis is triggered, the NC sets this special M relay to on.	R
MLC Z axis high-speed input point triggering feedback	M2130	When the MLC Z axis high-speed input point triggering special M relay (M1300) is set to on and the DI of Z axis is triggered, the NC sets this special M relay to on.	R
MLC A axis high-speed input point triggering feedback	M2131	When the MLC A axis high-speed input point triggering special M relay (M1301) is set to on and the DI of A axis is triggered, the NC sets this special M relay to on.	R
MLC B axis high-speed input point triggering feedback	M2132	When the MLC B axis high-speed input point triggering special M relay (M1302) is set to on and the DI of B axis is triggered, the NC sets this special M relay to on.	R
MLC C axis high-speed input point triggering feedback	M2133	When the MLC C axis high-speed input point triggering special M relay (M1303) is set to on and the DI of C axis is triggered, the NC sets this special M relay to on.	R
MLC U axis high-speed input point triggering feedback	M2134	When the MLC U axis high-speed input point triggering special M relay (M1304) is set to on and the DI of U axis is triggered, the NC sets this special M relay to on.	R
MLC V axis high-speed input point triggering feedback	M2135	When the MLC V axis high-speed input point triggering special M relay (M1305) is set to on and the DI of V axis is triggered, the NC sets this special M relay to on.	R
MLC W axis high-speed input point triggering feedback	M2136	When the MLC W axis high-speed input point triggering special M relay (M1306) is set to on and the DI of W axis is triggered, the NC sets this special M relay to on.	R
HSI 1	M2142	When the G31P1 (high-speed input contact 1) Skip signal is input to the controller, the NC sets this special M relay to on.	R
HSI 2	M2143	When the G31P2 (high-speed input contact 2) Skip signal is input to the controller, the NC sets this special M relay to on.	R
Port 1 positive hardware limit	M2144	When the positive hardware limit of Port 1 is triggered, the NC sets this special M relay to on.	R
Port 1 negative hardware limit	M2145	When the negative hardware limit of Port 1 is triggered, the NC sets this special M relay to on.	R
Port 1 home signal	M2146	When the home signal of Port 1 is triggered, the NC sets this special M relay to on.	R
Port 2 positive hardware limit	M2148	When the positive hardware limit of Port 2 is triggered, the NC sets this special M relay to on.	R
Port 2 negative hardware limit	M2149	When the negative hardware limit of Port 2 is triggered, the NC sets this special M relay to on.	R
Port 2 home signal	M2150	When the home signal of Port 2 is triggered, the NC sets this special M relay to on.	R
Port 3 positive hardware limit	M2152	When the positive hardware limit of Port 3 is triggered, the NC sets this special M relay to on.	R
Port 3 negative hardware limit	M2153	When the negative hardware limit of Port 3 is triggered, the NC sets this special M relay to on.	R
Port 3 home signal	M2154	When the home signal of Port 3 is triggered, the NC sets this special M relay to on.	R
Port 4 positive hardware limit	M2156	When the positive hardware limit of Port 4 is triggered, the NC sets this special M relay to on.	R
Port 4 negative hardware limit	M2157	When the negative hardware limit of Port 4 is triggered, the NC sets this special M relay to on.	R
Port 4 home signal	M2158	When the home signal of Port 4 is triggered, the NC sets this special M relay to on.	R
Port 5 positive hardware limit	M2160	When the positive hardware limit of Port 5 is triggered, the NC sets this special M relay to on.	R

Function name	Special M	Description	Device type
Port 5 negative hardware limit	M2161	When the negative hardware limit of Port 5 is triggered, the NC sets this special M relay to on.	R
Port 5 home signal	M2162	When the home signal of Port 5 is triggered, the NC sets this special M relay to on.	R
Port 6 positive hardware limit	M2164	When the positive hardware limit of Port 6 is triggered, the NC sets this special M relay to on.	R
Port 6 negative hardware limit	M2165	When the negative hardware limit of Port 6 is triggered, the NC sets this special M relay to on.	R
Port 6 home signal	M2166	When the home signal of Port 6 is triggered, the NC sets this special M relay to on.	R
Port 7 positive hardware limit	M2168	When the positive hardware limit of Port 7 is triggered, the NC sets this special M relay to on.	R
Port 7 negative hardware limit	M2169	When the negative hardware limit of Port 7 is triggered, the NC sets this special M relay to on.	R
Port 7 home signal	M2170	When the home signal of Port 7 is triggered, the NC sets this special M relay to on.	R
Port 8 positive hardware limit	M2172	When the positive hardware limit of Port 8 is triggered, the NC sets this special M relay to on.	R
Port 8 negative hardware limit	M2173	When the negative hardware limit of Port 8 is triggered, the NC sets this special M relay to on.	R
Port 8 home signal	M2174	When the home signal of Port 8 is triggered, the NC sets this special M relay to on.	R
Port 9 positive hardware limit	M2176	When the positive hardware limit of Port 9 is triggered, the NC sets this special M relay to on.	R
Port 9 negative hardware limit	M2177	When the negative hardware limit of Port 9 is triggered, the NC sets this special M relay to on.	R
Port 9 home signal	M2178	When the home signal of Port 9 is triggered, the NC sets this special M relay to on.	R
M code execution	M2208	When the M code is executed in the program, the NC sets M2208 to on. Until M1152 (M, S, and T codes execution complete) is triggered, M2208 set to off. The NC does not trigger M2208 when the program executes M00, M01, M02, M30, M98, M99, or an M code that is used for macro call.	R
S code execution	M2209	When the S code is executed in the program, the NC sets M2209 to on. Until M1152 (M, S, and T codes execution complete) is triggered, M2209 is set to off. The NC does not trigger M2209 when an S code is used for macro call.	R
T code execution	M2210	When the T code (Standby tool number) is executed in the program, the system sets M2210 to on. Until M1152 (M, S, and T codes execution complete) is triggered, M2210 is set to off. The NC does not trigger M2210 when a T code is used for macro call. M2210 is related to the station ID in the tool magazine. The NC system triggers M2210 only when the T code value is set within the specified range of tool number for the tool magazine parameter.	R
Tool magazine 1 reset complete	M2212	After you use M1171 to reset the tool magazine, the NC sets this special M relay to on. (M1171 is effective only when in AUTO or MDI mode)	R
Tool magazine 2 reset complete	M2213	After you use M1175 to reset the tool magazine, the NC sets this special M relay to on. (M1175 is effective only when in AUTO or MDI mode)	R
M96 (program interruption) in execution	M2216	When M96 (program interruption) is in execution, the NC sets this special M relay to on.	R

Function name	Special M	Description	Device type
G code ready	M2223	After loading the G codes, the NC sets this special M relay to on.	R
Macro call initial preparation complete	M2224	Indicates the macro call initial preparation is complete.	R
Macro call in execution	M2225	The special M is on when macro call is in execution.	R
Macro call error	M2226	Indicates a macro call error.	R
Synchronous function in execution	M2227	When the synchronous function is in execution, the NC sets this special M relay to on.	R
Transfer function in execution	M2228	When the transfer function is in execution, the NC sets this special M relay to on.	R
NC system reset complete	M2229	After resetting, the NC sets this special M relay to on.	R
MPG in forward operation	M2232	This special M relay is on when the MPG is in forward operation. This special M relay is off when the MPG is in reverse operation or stationary.	R
MPG in reverse operation	M2233	This special M relay is on when the MPG is in reverse operation. This special M relay is off when the MPG is in forward operation or stationary.	R
System stops at M99	M2238	When reading M99, the NC sets this special M relay to on.	R
Lathe C / S axis switching	M2239	When the lathe system is switched from Spindle mode to C axis mode, the NC sets this special M relay to on.	R
Channel alarm message	M2240	When an error occurs in the NC channel, the NC sets this special M relay to on.	R
Auto execution (AUTO)	M2241	When the NC is in AUTO mode, it sets this special M relay to on.	R
Program edit (EDIT)	M2242	When the NC is in EDIT mode, it sets this special M relay to on.	R
Manual input (MDI)	M2243	When the NC is in MDI mode, it sets this special M relay to on.	R
MPG feeding (MPG)	M2244	When the NC is in MPG mode, it sets this special M relay to on.	R
Jog (JOG)	M2245	When the NC is in JOG mode, it sets this special M relay to on.	R
Rapid feeding (RAPID)	M2246	When the NC is in RAPID mode, it sets this special M relay to on.	R
Homing (HOME)	M2247	When the NC is in HOME mode, it sets this special M relay to on.	R
Single block execution	M2249	When the program pauses after executing one single block, the NC sets this special M relay to on.	R
Program in execution	M2250	When the NC is executing the program, it sets this special M relay to on.	R
Pause	M2251	When the NC pauses, it sets this special M relay to on.	R
M00 program stop	M2252	When the NC reads M00, it sets this special M relay to on.	R
M01 program stop (optional)	M2253	When the NC reads M01, it sets this special M relay to on.	R
M02 end of program	M2254	When the NC reads M02, it sets this special M relay to on.	R
M30 end of program and returns	M2255	When the NC reads M30, it sets this special M relay to on.	R
1st spindle reaches the target speed	M2256	When the 1st spindle speed reaches the target value, the NC sets this special M relay to on.	R

Function name	Special M	Description	Device type
1st spindle reaches zero speed	M2257	When the 1st spindle speed reaches zero speed, the NC sets this special M relay to on.	R
1st spindle positioning complete	M2258	When the 1st spindle reaches the target position, the NC sets this special M relay to on.	R
1st spindle is in the rigid tapping mode	M2259	When the 1st spindle performs tapping, the NC sets this special M relay to on.	R
1st spindle rigid tapping interruption	M2260	When the 1st spindle tapping is interrupted, the NC sets this special M relay to on.	R
2nd spindle reaches the target speed	M2261	When the 2nd spindle speed reaches the target value, the NC sets this special M relay to on.	R
2nd spindle reaches zero speed	M2262	When the 2nd spindle speed reaches zero speed, the NC sets this special M relay to on.	R
2nd spindle positioning complete	M2263	When the 2nd spindle reaches the target position, the NC sets this special M relay to on.	R
2nd spindle is in the rigid tapping mode	M2264	When the 2nd spindle performs tapping, the NC sets this special M relay to on.	R
2nd spindle rigid tapping interruption	M2265	When the 2nd spindle tapping is interrupted, the NC sets this special M relay to on.	R
Program ends	M2271	When the machining program ends, the NC sets this special M relay to on.	R
X axis homing complete	M2272	When the X axis homing is complete, the NC sets this special M relay to on.	R
Y axis homing complete	M2273	When the Y axis homing is complete, the NC sets this special M relay to on.	R
Z axis homing complete	M2274	When the Z axis homing is complete, the NC sets this special M relay to on.	R
A axis homing complete	M2275	When the A axis homing is complete, the NC sets this special M relay to on.	R
B axis homing complete	M2276	When the B axis homing is complete, the NC sets this special M relay to on.	R
C axis homing complete	M2277	When the C axis homing is complete, the NC sets this special M relay to on.	R
U axis homing complete	M2278	When the U axis homing is complete, the NC sets this special M relay to on.	R
V axis homing complete	M2279	When the V axis homing is complete, the NC sets this special M relay to on.	R
W axis homing complete	M2280	When the W axis homing is complete, the NC sets this special M relay to on.	R
X axis positioned at the 2nd reference point	M2286	When the X axis reaches the 2nd reference point, the NC sets this special M relay to on.	R
Y axis positioned at the 2nd reference point	M2287	When the Y axis reaches the 2nd reference point, the NC sets this special M relay to on.	R
Z axis positioned at the 2nd reference point	M2288	When the Z axis reaches the 2nd reference point, the NC sets this special M relay to on.	R
A axis positioned at the 2nd reference point	M2289	When the A axis reaches the 2nd reference point, the NC sets this special M relay to on.	R
B axis positioned at the 2nd reference point	M2290	When the B axis reaches the 2nd reference point, the NC sets this special M relay to on.	R
C axis positioned at the 2nd reference point	M2291	When the C axis reaches the 2nd reference point, the NC sets this special M relay to on.	R

Function name	Special M	Description	Device type
U axis positioned at the 2nd reference point	M2292	When the U axis reaches the 2nd reference point, the NC sets this special M relay to on.	R
V axis positioned at the 2nd reference point	M2293	When the V axis reaches the 2nd reference point, the NC sets this special M relay to on.	R
W axis positioned at the 2nd reference point	M2294	When the W axis reaches the 2nd reference point, the NC sets this special M relay to on.	R
X axis positioned at the 3rd reference point	M2295	When the X axis reaches the 3rd reference point, the NC sets this special M relay to on.	R
Y axis positioned at the 3rd reference point	M2296	When the Y axis reaches the 3rd reference point, the NC sets this special M relay to on.	R
Z axis positioned at the 3rd reference point	M2297	When the Z axis reaches the 3rd reference point, the NC sets this special M relay to on.	R
A axis positioned at the 3rd reference point	M2298	When the A axis reaches the 3rd reference point, the NC sets this special M relay to on.	R
B axis positioned at the 3rd reference point	M2299	When the B axis reaches the 3rd reference point, the NC sets this special M relay to on.	R
C axis positioned at the 3rd reference point	M2300	When the C axis reaches the 3rd reference point, the NC sets this special M relay to on.	R
U axis positioned at the 3rd reference point	M2301	When the U axis reaches the 3rd reference point, the NC sets this special M relay to on.	R
V axis positioned at the 3rd reference point	M2302	When the V axis reaches the 3rd reference point, the NC sets this special M relay to on.	R
W axis positioned at the 3rd reference point	M2303	When the W axis reaches the 3rd reference point, the NC sets this special M relay to on.	R
MLC X axis positioning complete	M2304	When the X axis is in MLC axis control mode and X axis reaches the positioning point, the NC sets this special M relay to on. If the MLC is in speed mode, M2304 indicates the speed is reached.	R
MLC Y axis positioning complete	M2305	When the Y axis is in MLC axis control mode and Y axis reaches the positioning point, the NC sets this special M relay to on. If the MLC is in speed mode, M2305 indicates the speed is reached.	R
MLC Z axis positioning complete	M2306	When the Z axis is in MLC axis control mode and Z axis reaches the positioning point, the NC sets this special M relay to on. If the MLC is in speed mode, M2306 indicates the speed is reached.	R
MLC A axis positioning complete	M2307	When the A axis is in MLC axis control mode and A axis reaches the positioning point, the NC sets this special M relay to on. If the MLC is in speed mode, M2307 indicates the speed is reached.	R
MLC B axis positioning complete	M2308	When the B axis is in MLC axis control mode and B axis reaches the positioning point, the NC sets this special M relay to on. If the MLC is in speed mode, M2308 indicates the speed is reached.	R
MLC C axis positioning complete	M2309	When the C axis is in MLC axis control mode and C axis reaches the positioning point, the NC sets this special M relay to on. If the MLC is in speed mode, M2309 indicates the speed is reached.	R
MLC U axis positioning complete	M2310	When the U axis is in MLC axis control mode and U axis reaches the positioning point, the NC sets this special M relay to on. If the MLC is in speed mode, M2310 indicates the speed is reached.	R

Function name	Special M	Description	Device type
MLC V axis positioning complete	M2311	When the V axis is in MLC axis control mode and V axis reaches the positioning point, the NC sets this special M relay to on. If the MLC is in speed mode, M2311 indicates the speed is reached.	R
MLC W axis positioning complete	M2312	When the W axis is in MLC axis control mode and W axis reaches the positioning point, the NC sets this special M relay to on. If the MLC is in speed mode, M2312 indicates the speed is reached.	R
X axis is moving	M2320	When X axis is moving (in any modes), the NC sets this special M relay to on.	R
Y axis is moving	M2321	When Y axis is moving (in any modes), the NC sets this special M relay to on.	R
Z axis is moving	M2322	When Z axis is moving (in any modes), the NC sets this special M relay to on.	R
A axis is moving	M2323	When A axis is moving (in any modes), the NC sets this special M relay to on.	R
B axis is moving	M2324	When B axis is moving (in any modes), the NC sets this special M relay to on.	R
C axis is moving	M2325	When C axis is moving (in any modes), the NC sets this special M relay to on.	R
U axis is moving	M2326	When U axis is moving (in any modes), the NC sets this special M relay to on.	R
V axis is moving	M2327	When V axis is moving (in any modes), the NC sets this special M relay to on.	R
W axis is moving	M2328	When W axis is moving (in any modes), the NC sets this special M relay to on.	R
X axis is moving in forward direction	M2336	This special M relay is on when X axis is moving in forward direction.	R
Y axis is moving in forward direction	M2337	This special M relay is on when Y axis is moving in forward direction.	R
Z axis is moving in forward direction	M2338	This special M relay is on when Z axis is moving in forward direction.	R
A axis is moving in forward direction	M2339	This special M relay is on when A axis is moving in forward direction.	R
B axis is moving in forward direction	M2340	This special M relay is on when B axis is moving in forward direction.	R
C axis is moving in forward direction	M2341	This special M relay is on when C axis is moving in forward direction.	R
U axis is moving in forward direction	M2342	This special M relay is on when U axis is moving in forward direction.	R
V axis is moving in forward direction	M2343	This special M relay is on when V axis is moving in forward direction.	R
W axis is moving in forward direction	M2344	This special M relay is on when W axis is moving in forward direction.	R
X axis is moving in reverse direction	M2345	This special M relay is on when X axis is moving in reverse direction.	R
Y axis is moving in reverse direction	M2346	This special M relay is on when Y axis is moving in reverse direction.	R
Z axis is moving in reverse direction	M2347	This special M relay is on when Z axis is moving in reverse direction.	R
A axis is moving in reverse direction	M2348	This special M relay is on when A axis is moving in reverse direction.	R
B axis is moving in reverse direction	M2349	This special M relay is on when B axis is moving in reverse direction.	R
C axis is moving in reverse direction	M2350	This special M relay is on when C axis is moving in reverse direction.	R

Function name	Special M	Description	Device type
U axis is moving in reverse direction	M2351	This special M relay is on when U axis is moving in reverse direction.	R
V axis is moving in reverse direction	M2352	This special M relay is on when V axis is moving in reverse direction.	R
W axis is moving in reverse direction	M2353	This special M relay is on when W axis is moving in reverse direction.	R
NC / MLC axis switching complete (X axis)	M2354	When the NC sets M1200 to on to switch X axis from NC to MLC, the NC sets this special M relay to on.	R
NC / MLC axis switching complete (Y axis)	M2355	When the NC sets M1201 to on to switch Y axis from NC to MLC, the NC sets this special M relay to on.	R
NC / MLC axis switching complete (Z axis)	M2356	When the NC sets M1202 to on to switch Z axis from NC to MLC, the NC sets this special M relay to on.	R
NC / MLC axis switching complete (A axis)	M2357	When the NC sets M1203 to on to switch A axis from NC to MLC, the NC sets this special M relay to on.	R
NC / MLC axis switching complete (B axis)	M2358	When the NC sets M1204 to on to switch B axis from NC to MLC, the NC sets this special M relay to on.	R
NC / MLC axis switching complete (C axis)	M2359	When the NC sets M1205 to on to switch C axis from NC to MLC, the NC sets this special M relay to on.	R
NC / MLC axis switching complete (U axis)	M2360	When the NC sets M1206 to on to switch U axis from NC to MLC, the NC sets this special M relay to on.	R
NC / MLC axis switching complete (V axis)	M2361	When the NC sets M1207 to on to switch V axis from NC to MLC, the NC sets this special M relay to on.	R
NC / MLC axis switching complete (W axis)	M2362	When the NC sets M1208 to on to switch W axis from NC to MLC, the NC sets this special M relay to on.	R
X axis positioned at the 4th reference point	M2368	When the X axis reaches the 4th reference point, the NC sets this special M relay to on.	R
Y axis positioned at the 4th reference point	M2369	When the Y axis reaches the 4th reference point, the NC sets this special M relay to on.	R
Z axis positioned at the 4th reference point	M2370	When the Z axis reaches the 4th reference point, the NC sets this special M relay to on.	R
A axis positioned at the 4th reference point	M2371	When the A axis reaches the 4th reference point, the NC sets this special M relay to on.	R
B axis positioned at the 4th reference point	M2372	When the B axis reaches the 4th reference point, the NC sets this special M relay to on.	R
C axis positioned at the 4th reference point	M2373	When the C axis reaches the 4th reference point, the NC sets this special M relay to on.	R
U axis positioned at the 4th reference point	M2374	When the U axis reaches the 4th reference point, the NC sets this special M relay to on.	R
V axis positioned at the 4th reference point	M2375	When the V axis reaches the 4th reference point, the NC sets this special M relay to on.	R
W axis positioned at the 4th reference point	M2376	When the W axis reaches the 4th reference point, the NC sets this special M relay to on.	R
X axis homing complete	M2377	When the X axis origin coordinate is established, the “Homing complete” symbol (⊕) is displayed in the POS screen, and the NC sets this special M relay to on.	R
Y axis homing complete	M2378	When the Y axis origin coordinate is established, the “Homing complete” symbol is displayed in the POS screen, and the NC sets this special M relay to on.	R

Function name	Special M	Description	Device type
Z axis homing complete	M2379	When the Z axis origin coordinate is established, the “Homing complete” symbol is displayed in the POS screen, and the NC sets this special M relay to on.	R
A axis homing complete	M2380	When the A axis origin coordinate is established, the “Homing complete” symbol is displayed in the POS screen, and the NC sets this special M relay to on.	R
B axis homing complete	M2381	When the B axis origin coordinate is established, the “Homing complete” symbol is displayed in the POS screen, and the NC sets this special M relay to on.	R
C axis homing complete	M2382	When the C axis origin coordinate is established, the “Homing complete” symbol is displayed in the POS screen, and the NC sets this special M relay to on.	R
U axis homing complete	M2383	When the U axis origin coordinate is established, the “Homing complete” symbol is displayed in the POS screen, and the NC sets this special M relay to on.	R
V axis homing complete	M2384	When the V axis origin coordinate is established, the “Homing complete” symbol is displayed in the POS screen, and the NC sets this special M relay to on.	R
W axis homing complete	M2385	When the W axis origin coordinate is established, the “Homing complete” symbol is displayed in the POS screen, and the NC sets this special M relay to on.	R
X axis Servo On / Off	M2386	When X axis is in Servo On status, the NC sets this special M relay to on.	R
Y axis Servo On / Off	M2387	When Y axis is in Servo On status, the NC sets this special M relay to on.	R
Z axis Servo On / Off	M2388	When Z axis is in Servo On status, the NC sets this special M relay to on.	R
A axis Servo On / Off	M2389	When A axis is in Servo On status, the NC sets this special M relay to on.	R
B axis Servo On / Off	M2390	When B axis is in Servo On status, the NC sets this special M relay to on.	R
C axis Servo On / Off	M2391	When C axis is in Servo On status, the NC sets this special M relay to on.	R
U axis Servo On / Off	M2392	When U axis is in Servo On status, the NC sets this special M relay to on.	R
V axis Servo On / Off	M2393	When V axis is in Servo On status, the NC sets this special M relay to on.	R
W axis Servo On / Off	M2394	When W axis is in Servo On status, the NC sets this special M relay to on.	R
Movement in X axis	M2400	When Pr501 (Axis movement protection) is set to 1, if the X axis moves during machining, the NC sets the corresponding special M relay to on.	R
Movement in Y axis	M2401	When Pr501 (Axis movement protection) is set to 1, if the Y axis moves during machining, the NC sets the corresponding special M relay to on.	R
Movement in Z axis	M2402	When Pr501 (Axis movement protection) is set to 1, if the Z axis moves during machining, the NC sets the corresponding special M relay to on.	R
Movement in A axis	M2403	When Pr501 (Axis movement protection) is set to 1, if the A axis moves during machining, the NC sets the corresponding special M relay to on.	R
Movement in B axis	M2404	When Pr501 (Axis movement protection) is set to 1, if the B axis moves during machining, the NC sets the corresponding special M relay to on.	R
Movement in C axis	M2405	When Pr501 (Axis movement protection) is set to 1, if the C axis moves during machining, the NC sets the corresponding special M relay to on.	R

Function name	Special M	Description	Device type
Movement in U axis	M2406	When Pr501 (Axis movement protection) is set to 1, if the U axis moves during machining, the NC sets the corresponding special M relay to on.	R
Movement in V axis	M2407	When Pr501 (Axis movement protection) is set to 1, if the V axis moves during machining, the NC sets the corresponding special M relay to on.	R
Movement in W axis	M2408	When Pr501 (Axis movement protection) is set to 1, if the W axis moves during machining, the NC sets the corresponding special M relay to on.	R
MLC instruction operation result is 0	M2824	If the operation result is 0 when the MLC executes the ADD, DADD, FADD, SUB, DSUB, FSUB, FMUL, FDIV, FINT, FDOT, FRAD, and FDEG instructions, the NC sets this special M relay to on.	R
MLC instruction borrow operation	M2825	If the operation result is smaller than -32,768 when the MLC executes the ADD and SUB instructions, the NC sets this special M relay to on to indicate a borrow. If the operation result is smaller than -2,147,483,648 when the MLC executes the DADD and DSUB instructions, the NC sets this special M relay to on to indicate a borrow.	R
MLC instruction carry operation	M2826	If the operation result is greater than 32,767 when the MLC executes the ADD and SUB instructions, the NC sets this special M relay to on to indicate a carry. If the operation result is greater than 2,147,483,647 when the MLC executes the DADD and DSUB instructions, the NC sets this special M relay to on to indicate a carry.	R
MLC instruction operation error	M2828	If the dividend is 0 when the MLC executes the DIV, DDIV, and FDIV instructions, the NC sets this special M relay to on. If the conversion range exceeds K0 - K9,999 when the MLC executes the BCD instruction, the NC sets this special M relay to on. If the conversion range exceeds K0 - K99,999,999 when the MLC executes the DBCD instruction, the NC sets this special M relay to on.	R
DMCNET connection status - Station 1	M2864	DMCNET connection status: success = 1; failure = 0	R
DMCNET connection status - Station 2	M2865	DMCNET connection status: success = 1; failure = 0	R
DMCNET connection status - Station 3	M2866	DMCNET connection status: success = 1; failure = 0	R
DMCNET connection status - Station 4	M2867	DMCNET connection status: success = 1; failure = 0	R
DMCNET connection status - Station 5	M2868	DMCNET connection status: success = 1; failure = 0	R
DMCNET connection status - Station 6	M2869	DMCNET connection status: success = 1; failure = 0	R
DMCNET connection status - Station 7	M2870	DMCNET connection status: success = 1; failure = 0	R
DMCNET connection status - Station 8	M2871	DMCNET connection status: success = 1; failure = 0	R
DMCNET connection status - Station 9	M2872	DMCNET connection status: success = 1; failure = 0	R
DMCNET connection status - Station 10	M2873	DMCNET connection status: success = 1; failure = 0	R
DMCNET connection status - Station 11	M2874	DMCNET connection status: success = 1; failure = 0	R
DMCNET connection status - Station 12	M2875	DMCNET connection status: success = 1; failure = 0	R

Function name	Special M	Description	Device type
IX00 interrupt input	M2880	When this special M relay is set on, the IX00 interrupt input function is enabled. Input point: X0 of the controller mainboard I/O.	R/W
IX01 interrupt input	M2881	When this special M relay is set on, the IX01 interrupt input function is enabled. Input point: X1 of the controller mainboard I/O.	R/W
IX02 interrupt input	M2882	When this special M relay is set on, the IX02 interrupt input function is enabled. Input point: X2 of the controller mainboard I/O.	R/W
IX03 interrupt input	M2883	When this special M relay is set on, the IX03 interrupt input function is enabled. Input point: X3 of the controller mainboard I/O.	R/W
IX04 interrupt input	M2884	When this special M relay is set on, the IX04 interrupt input function is enabled. Input point: X4 of the controller mainboard I/O.	R/W
IX05 interrupt input	M2885	When this special M relay is set on, the IX05 interrupt input function is enabled. Input point: X5 of the controller mainboard I/O.	R/W
IX06 interrupt input	M2886	When this special M relay is set on, the IX06 interrupt input function is enabled. Input point: X6 of the controller mainboard I/O.	R/W
IX07 interrupt input	M2887	When this special M relay is set on, the IX07 interrupt input function is enabled. Input point: X7 of the controller mainboard I/O.	R/W
IC00 interrupt input	M2888	When this special M relay is set to on, the IC00 interrupt input function is enabled. Input point: hardware high-speed input counting point 0.	R/W
IC01 interrupt input	M2889	When this special M relay is set to on, the IC01 interrupt input function is enabled. Input point: hardware high-speed input counting point 1.	R/W
IR00 interrupt input	M2896	When this special M relay is set to on, the IR00 interrupt input function is enabled. Input point: X0 of the 0th RIO card.	R/W
IR01 interrupt input	M2897	When this special M relay is set to on, the IR01 interrupt input function is enabled. Input point: X1 of the 0th RIO card.	R/W
IR02 interrupt input	M2898	When this special M relay is set to on, the IR02 interrupt input function is enabled. Input point: X2 of the 0th RIO card.	R/W
IR03 interrupt input	M2899	When this special M relay is set to on, the IR03 interrupt input function is enabled. Input point: X3 of the 0th RIO card.	R/W
IR04 interrupt input	M2900	When this special M relay is set to on, the IR04 interrupt input function is enabled. Input point: X4 of the 0th RIO card.	R/W
IR05 interrupt input	M2901	When this special M relay is set to on, the IR05 interrupt input function is enabled. Input point: X5 of the 0th RIO card.	R/W
IR06 interrupt input	M2902	When this special M relay is set to on, the IR06 interrupt input function is enabled. Input point: X6 of the 0th RIO card.	R/W
IR07 interrupt input	M2903	When this special M relay is set to on, the IR07 interrupt input function is enabled. Input point: X7 of the 0th RIO card.	R/W

Function name	Special M	Description	Device type
IR08 interrupt input	M2904	When this special M relay is set to on, the IR08 interrupt input function is enabled. Input point: X8 of the 0th RIO card.	R/W
IR09 interrupt input	M2905	When this special M relay is set to on, the IR09 interrupt input function is enabled. Input point: X9 of the 0th RIO card.	R/W
IR10 interrupt input	M2906	When this special M relay is set to on, the IR10 interrupt input function is enabled. Input point: X10 of the 0th RIO card.	R/W
IR11 interrupt input	M2907	When this special M relay is set to on, the IR11 interrupt input function is enabled. Input point: X11 of the 0th RIO card.	R/W
IR12 interrupt input	M2908	When this special M relay is set to on, the IR12 interrupt input function is enabled. Input point: X12 of the 0th RIO card.	R/W
IR13 interrupt input	M2909	When this special M relay is set to on, the IR13 interrupt input function is enabled. Input point: X13 of the 0th RIO card.	R/W
IR14 interrupt input	M2910	When this special M relay is set to on, the IR14 interrupt input function is enabled. Input point: X14 of the 0th RIO card.	R/W
IR15 interrupt input	M2911	When this special M relay is set to on, the IR15 interrupt input function is enabled. Input point: X15 of the 0th RIO card.	R/W
IR16 interrupt input	M2912	When this special M relay is set to on, the IR16 interrupt input function is enabled. Input point: X16 of the 0th RIO card.	R/W
IR17 interrupt input	M2913	When this special M relay is set to on, the IR17 interrupt input function is enabled. Input point: X17 of the 0th RIO card.	R/W
IR18 interrupt input	M2914	When this special M relay is set to on, the IR18 interrupt input function is enabled. Input point: X18 of the 0th RIO card.	R/W
IR19 interrupt input	M2915	When this special M relay is set to on, the IR19 interrupt input function is enabled. Input point: X19 of the 0th RIO card.	R/W
IR20 interrupt input	M2916	When this special M relay is set to on, the IR20 interrupt input function is enabled. Input point: X20 of the 0th RIO card.	R/W
IR21 interrupt input	M2917	When this special M relay is set to on, the IR21 interrupt input function is enabled. Input point: X21 of the 0th RIO card.	R/W
IR22 interrupt input	M2918	When this special M relay is set to on, the IR22 interrupt input function is enabled. Input point: X22 of the 0th RIO card.	R/W
IR23 interrupt input	M2919	When this special M relay is set to on, the IR23 interrupt input function is enabled. Input point: X23 of the 0th RIO card.	R/W
IR24 interrupt input	M2920	When this special M relay is set to on, the IR24 interrupt input function is enabled. Input point: X24 of the 0th RIO card.	R/W
IR25 interrupt input	M2921	When this special M relay is set to on, the IR25 interrupt input function is enabled. Input point: X25 of the 0th RIO card.	R/W

Function name	Special M	Description	Device type
IR26 interrupt input	M2922	When this special M relay is set to on, the IR26 interrupt input function is enabled. Input point: X26 of the 0th RIO card.	R/W
IR27 interrupt input	M2923	When this special M relay is set to on, the IR27 interrupt input function is enabled. Input point: X27 of the 0th RIO card.	R/W
IR28 interrupt input	M2924	When this special M relay is set to on, the IR28 interrupt input function is enabled. Input point: X28 of the 0th RIO card.	R/W
IR29 interrupt input	M2925	When this special M relay is set to on, the IR29 interrupt input function is enabled. Input point: X29 of the 0th RIO card.	R/W
IR30 interrupt input	M2926	When this special M relay is set to on, the IR30 interrupt input function is enabled. Input point: X30 of the 0th RIO card.	R/W
IR31 interrupt input	M2927	When this special M relay is set to on, the IR31 interrupt input function is enabled. Input point: X31 of the 0th RIO card.	R/W
Lock user permission	M2934	User permission is locked when this special M relay is on. Method: after Pr10015 (Account permission activation method) is set to 1, when M2934 is set to on, the user permission is locked.	R/W
Restrict program editing	M2935	Restricts program editing in the controller.	R/W
DCNT counter C64 counts down	M2944	When this special M relay is set to on, the counter decrements by 1 every time it is triggered.	R/W
DCNT counter C65 counts down	M2945	When this special M relay is set to on, the counter decrements by 1 every time it is triggered.	R/W
DCNT counter C66 counts down	M2946	When this special M relay is set to on, the counter decrements by 1 every time it is triggered.	R/W
DCNT counter C67 counts down	M2947	When this special M relay is set to on, the counter decrements by 1 every time it is triggered.	R/W
DCNT counter C68 counts down	M2948	When this special M relay is set to on, the counter decrements by 1 every time it is triggered.	R/W
DCNT counter C69 counts down	M2949	When this special M relay is set to on, the counter decrements by 1 every time it is triggered.	R/W
DCNT counter C70 counts down	M2950	When this special M relay is set to on, the counter decrements by 1 every time it is triggered.	R/W
DCNT counter C71 counts down	M2951	When this special M relay is set to on, the counter decrements by 1 every time it is triggered.	R/W
DCNT counter C72 counts down	M2952	When this special M relay is set to on, the counter decrements by 1 every time it is triggered.	R/W
DCNT counter C73 counts down	M2953	When this special M relay is set to on, the counter decrements by 1 every time it is triggered.	R/W
DCNT counter C74 counts down	M2954	When this special M relay is set to on, the counter decrements by 1 every time it is triggered.	R/W
DCNT counter C75 counts down	M2955	When this special M relay is set to on, the counter decrements by 1 every time it is triggered.	R/W
DCNT counter C76 counts down	M2956	When this special M relay is set to on, the counter decrements by 1 every time it is triggered.	R/W
DCNT counter C77 counts down	M2957	When this special M relay is set to on, the counter decrements by 1 every time it is triggered.	R/W
File queue auto processing	M2980	In AUTO mode, the system automatically executes the next program in the FILE QUEUE list when this special M relay is triggered.	R/W

Function name	Special M	Description	Device type
G00 teaching triggered	M2992	When G00 teaching is used, the NC sets this special M relay to on.	R
G01 teaching triggered	M2993	When G01 teaching is used, the NC sets this special M relay to on.	R
G00 teaching record complete	M2994	When G00 teaching is used and the motion path is recorded, the special M relay is on.	R
G01 teaching record complete	M2995	When G01 teaching is used and the motion path is recorded, the special M relay is on.	R

5.2.2 List of special D registers

Function name	Special D	Description	Device type
Number of the processing complete pieces (32-bit)	D1018	Set the value using D1022 in the Process screen or with the MLC input. Set Pr10015 [Bit 6] to decide the display source. D1019 is the high word of this 32-bit register. When Pr10015 [Bit 6] is 0, the display source is D1022 and D1023. When Pr10015 [Bit 6] is 1, the display source is D1018 (D1019) and D1020 (D1021).	R/W
Number of the processing target (32-bit)	D1020	Set the value using D1023 in the Process screen or with the MLC input. D1021 is the high word of this 32-bit register.	R/W
Number of the processing complete pieces	D1022	Set the value using D1022 in the Process screen or with the MLC input.	R/W
Number of the processing target	D1023	Set the value using D1023 in the Process screen or with the MLC input.	R/W
HMI output point 1	D1024	This value is paired with #1833.	R/W
HMI output point 2	D1025	This value is paired with #1834.	R/W
HMI output point 3	D1026	This value is paired with #1835.	R/W
HMI output point 4	D1027	This value is paired with #1836.	R/W
HMI output point 5	D1028	This value is paired with #1837.	R/W
HMI output point 6	D1029	This value is paired with #1838.	R/W
HMI output point 7	D1030	This value is paired with #1839.	R/W
HMI output point 8	D1031	This value is paired with #1840.	R/W
HMI output point 9	D1032	This value is paired with #1841.	R/W
HMI output point 10	D1033	This value is paired with #1842.	R/W
HMI output point 11	D1034	This value is paired with #1843.	R/W
HMI output point 12	D1035	This value is paired with #1844.	R/W
HMI output point 13	D1036	This value is paired with #1845.	R/W
HMI output point 14	D1037	This value is paired with #1846.	R/W
HMI output point 15	D1038	This value is paired with #1847.	R/W
HMI output point 16	D1039	This value is paired with #1848.	R/W
MPG operation mode number	D1040	Sets the MPG operation mode. Set D1040 to 0 to use the external MPG. Set D1040 to 10 to use the keys on the machine operation panel B as the MPG function, with the pulse control trigger flags of M1118 and M1119.	R/W
MPG operation channel selection	D1041	Designates the MPG operation channel. The default value is 0.	R/W
MPG pulse magnification	D1042	Sets the MPG pulse magnifications ×1, ×10, and ×100. The movement is the minimum unit 0.001 mm multiply by the magnification. For example, 1 × 0.001 = 0.001 mm.	R/W
Select the axis with MPG	D1043	Selects the axis to move with the MPG. Set 0 for X axis, 1 for Y axis, 2 for Z axis, 3 for A axis, and 4 for B axis.	R/W
DMCNET ready for HMI	D1048	When the NC is ready, D1048 indicates the number of connecting DMCNET axes by representing the binary data in decimal. For example, D1048 represents the binary data 1111 as 15 in decimal to indicate there are 4 axes connected.	R

Function name	Special D	Description	Device type
Cutting feed rate adjustment	D1056	Sets the percentage of the cutting feed rate (F) in NC program. If you set F to 1000 and the current value of D1056 is 50, it means the actual command speed is 500 mm/min (= 1000 x 50%).	R/W
Rapid movement speed adjustment	D1058	Sets the percentage of G00's speed (rapid movement). For example, if the rapid movement speed is 6000 and D1058 is set to 50, it means the actual speed of G00 is 3000 mm/min (= 6000 x 50%).	R/W
1st spindle speed adjustment	D1060	Sets the percentage of the S value specified for the 1st spindle in the program. For example, if S1000 is given in the program and D1060 is set to 30, it means the actual spindle speed is 300 rpm (= 1000 x 30%).	R/W
Speed setting for jog feeding and dry run	D1062	Sets the movement speed F for dry run in JOG or AUTO mode. If you set this special D register to 50, it indicates F50 (mm/min) with the range of 0 - 65535 mm/min. This is a 32-bit special D register and D1063 is the high word.	R/W
MLC X axis positioning command	D1064	Specifies the X axis moving amount in the MLC axis mode. Unit: mm, inch. This is a 32-bit special D register.	R/W
MLC Y axis positioning command	D1066	Specifies the Y axis moving amount in the MLC axis mode. Unit: mm, inch. This is a 32-bit special D register.	R/W
MLC Z axis positioning command	D1068	Specifies the Z axis moving amount in the MLC axis mode. Unit: mm, inch. This is a 32-bit special D register.	R/W
MLC A axis positioning command	D1070	Specifies the A axis moving amount in the MLC axis mode. Unit: mm, inch. This is a 32-bit special D register.	R/W
MLC B axis positioning command	D1072	Specifies the B axis moving amount in the MLC axis mode. Unit: mm, inch. This is a 32-bit special D register.	R/W
MLC C axis positioning command	D1074	Specifies the C axis moving amount in the MLC axis mode. Unit: mm, inch. This is a 32-bit special D register.	R/W
MLC U axis positioning command	D1076	Specifies the U axis moving amount in the MLC axis mode. Unit: mm, inch. This is a 32-bit special D register.	R/W
MLC V axis positioning command	D1078	Specifies the V axis moving amount in the MLC axis mode. Unit: mm, inch. This is a 32-bit special D register.	R/W
MLC W axis positioning command	D1080	Specifies the W axis moving amount in the MLC axis mode. Unit: mm, inch. This is a 32-bit special D register.	R/W
MLC X axis positioning speed	D1082	Specifies the X axis moving speed in the MLC axis mode. Unit: mm/min, inch/min. This is a 32-bit special D register.	R/W
MLC Y axis positioning speed	D1084	Specifies the Y axis moving speed in the MLC axis mode. Unit: mm/min, inch/min. This is a 32-bit special D register.	R/W
MLC Z axis positioning speed	D1086	Specifies the Z axis moving speed in the MLC axis mode. Unit: mm/min, inch/min. This is a 32-bit special D register.	R/W
MLC A axis positioning speed	D1088	Specifies the A axis moving speed in the MLC axis mode. Unit: rpm. This is a 32-bit special D register.	R/W

Function name	Special D	Description	Device type
MLC B axis positioning speed	D1090	Specifies the B axis moving speed in the MLC axis mode. Unit: rpm. This is a 32-bit special D register.	R/W
MLC C axis positioning speed	D1092	Specifies the C axis moving speed in the MLC axis mode. Unit: rpm. This is a 32-bit special D register.	R/W
MLC U axis positioning speed	D1094	Specifies the U axis moving speed in the MLC axis mode. Unit: mm/min, inch/min. This is a 32-bit special D register.	R/W
MLC V axis positioning speed	D1096	Specifies the V axis moving speed in the MLC axis mode. Unit: mm/min, inch/min. This is a 32-bit special D register.	R/W
MLC W axis positioning speed	D1098	Specifies the W axis moving speed in the MLC axis mode. Unit: mm/min, inch/min. This is a 32-bit special D register.	R/W
MLC spindle positioning speed	D1100	Specifies the spindle moving speed in the MLC axis mode. Unit: rpm. This is a 32-bit special D register.	R/W
Main file name	D1102	When Pr12017 (Record system data in special D registers) is set to 1, this special D register records the file names of O0000 - O9999 which are recently used.	R
Total machining time	D1103	When Pr12017 (Record system data in special D registers) is set to 1, this special D register records the total machining time in seconds together with D1104.	R
Single machining time	D1105	When Pr12017 (Record system data in special D registers) is set to 1, this special D register records the single machining time in seconds together with D1106.	R
System time: year, month	D1107	When Pr12017 (Record system data in special D registers) is set to 1, this special D register records the year and month of the system time.	R
System time: date, hour	D1108	When Pr12017 (Record system data in special D registers) is set to 1, this special D register records the date and hour of the system time.	R
System time: minute, second	D1109	When Pr12017 (Record system data in special D registers) is set to 1, this special D register records the minute and second of the system time.	R
Macro call file name	D1111	Specifies the macro call file name O9xxx. For example, if D1111 writes K9100, the NC system calls the macro named O9100.	R/W
2nd spindle speed adjustment	D1112	Sets the percentage of the S value specified for the 2nd spindle in the program. For example, if S1000 is given in the program and D1112 is set to 30, it means the actual spindle speed is 300 rpm (= 1000 x 30%).	R/W
Spindle analog voltage output port 2	D1114	When the spindle is in DMCNET mode, you can use this special D register to output the analog voltage. Setting range: -1000 to 1000; unit: 0.01V. This function is only available on the B series products.	R/W
Lathe tool number selection	D1115	When you set the lathe parameter Pr308 [Bit 13] to 1, this special D register displays the lathe tool number in the range of 0 - 65535.	R/W
Spindle analog voltage output port 1	D1125	When the spindle is in DMCNET mode, you can use this special D register to output the analog voltage. Setting range: -1000 to 1000; unit: 0.01V.	R/W

Function name	Special D	Description	Device type
Equi-slope filter compensation for X axis	D1126	X axis filter compensation value in the unit of mm.	R/W
Equi-slope filter compensation for Y axis	D1128	Y axis filter compensation value in the unit of mm.	R/W
Equi-slope filter compensation for Z axis	D1130	Z axis filter compensation value in the unit of mm.	R/W
Equi-slope filter compensation for A axis	D1132	A axis filter compensation value in the unit of mm or deg.	R/W
Equi-slope filter compensation for B axis	D1134	B axis filter compensation value in the unit of mm or deg.	R/W
Equi-slope filter compensation for C axis	D1136	C axis filter compensation value in the unit of mm or deg.	R/W
Equi-slope filter compensation for U axis	D1138	U axis filter compensation value in the unit of mm or deg.	R/W
Equi-slope filter compensation for V axis	D1140	V axis filter compensation value in the unit of mm or deg.	R/W
Equi-slope filter compensation for W axis	D1142	W axis filter compensation value in the unit of mm or deg.	R/W
Linear axis compensation speed	D1144	The linear axis compensation speed in the unit of mm/sec.	R/W
Rotation axis compensation speed	D1146	The rotation axis compensation speed in the unit of deg/sec.	R/W
1st spindle speed (written with special D register)	D1148	Write the 1st spindle speed with this special D register. (Must be used with M1307)	R/W
Arc preview speed	D1150	Adjusts the preview speed of the arc block when the PREVIEW function is used. This is a 32-bit special D register.	R/W
2nd spindle speed (written with special D register)	D1152	Write the 2nd spindle speed with this special D register. (Must be used with M1307)	R/W
Spindle tool number (written with special D register)	D1172	Write the spindle tool number with this special D register. (The setting is available only when Pr308 [Bit 13] = 1)	R/W
Standby tool number (written with special D register)	D1173	Write the standby tool number with this special D register. (The setting is available only when Pr308 [Bit 13] = 1)	R/W
Command tool number (written with special D register)	D1174	Write the command tool number with this special D register. (The setting is available only when Pr308 [Bit 13] = 1)	R/W
HMI input point 1	D1336	The value of this special D register changes with the system variable #1896.	R
HMI input point 2	D1337	The value of this special D register changes with the system variable #1897.	R
HMI input point 3	D1338	The value of this special D register changes with the system variable #1898.	R
HMI input point 4	D1339	The value of this special D register changes with the system variable #1899.	R
HMI input point 5	D1340	The value of this special D register changes with the system variable #1900.	R
HMI input point 6	D1341	The value of this special D register changes with the system variable #1901.	R
HMI input point 7	D1342	The value of this special D register changes with the system variable #1902.	R

Function name	Special D	Description	Device type
HMI input point 8	D1343	The value of this special D register changes with the system variable #1903.	R
HMI input point 9	D1344	The value of this special D register changes with the system variable #1904.	R
HMI input point 10	D1345	The value of this special D register changes with the system variable #1905.	R
HMI input point 11	D1346	The value of this special D register changes with the system variable #1906.	R
HMI input point 12	D1347	The value of this special D register changes with the system variable #1907.	R
HMI input point 13	D1348	The value of this special D register changes with the system variable #1908.	R
HMI input point 14	D1349	The value of this special D register changes with the system variable #1909.	R
HMI input point 15	D1350	The value of this special D register changes with the system variable #1910.	R
HMI input point 16	D1351	The value of this special D register changes with the system variable #1911.	R
1st spindle command speed (32-bit)	D1364	When the 1st spindle S code is executed in a program, the value of the S code is stored in both registers D1369 and D1364. When the value of the 1st spindle command speed exceed 16 bits, you cannot access the 1st spindle command speed with D1369. In this case, you can access the actual spindle command speed with D1364. Unit: rpm. D1365 is the high word of this 32-bit register.	R
M code data	D1368	When an M code is executed in a program, the value of the M code is stored in register D1368. When the M code is used for macro call, the value in this special D register does not change. The M codes mentioned here do not include M00, M01, M02, M30, M98, and M99.	R
1st spindle S code data	D1369	When the 1st spindle S code is executed in a program, the value of the S code is stored in register D1369. When the S code is used for macro call, the value in this special D register does not change. Unit: rpm.	R
T code data (command)	D1370	When a T code is executed in a program, the value of the T code is stored in register D1370. When the T code is used for macro call, the value in this special D register does not change. D1370 is related to the station ID in the tool magazine. The NC system triggers D1370 only when the T code value is set within the specified range of tool number for the tool magazine parameter.	R
Tool magazine 1 T code data (standby)	D1371	Records the latest T code data read in tool magazine 1.	R
Tool magazine 1 tool pot deviation	D1372	The deviation between the positions specified in D1370 and D1371 in tool magazine 1. When the tool magazine rotates forward (M1168) or backward (M1169) during tool exchange, the tool magazine needs to rotate according to the value in D1372 for compensating the offset.	R
Tool magazine 1 tool pot number (standby)	D1373	The standby tool pot number in tool magazine 1 is stored in this special D register.	R
Tool magazine 1 spindle tool number (in use)	D1374	The spindle tool number currently used in tool magazine 1.	R
Tool magazine 2 T code data (standby)	D1375	Records the latest T code data read in tool magazine 2.	R

Function name	Special D	Description	Device type
Tool magazine 2 tool pot deviation	D1376	The deviation between the positions specified in D1370 and D1375 in tool magazine 2. When the tool magazine rotates forward (M1172) or backward (M1173) during tool exchange, the tool magazine needs to rotate according to the value in D1376 for compensating the offset.	R
Tool magazine 2 tool pot number (standby)	D1377	The standby tool pot number in tool magazine 2 is stored in this special D register.	R
Tool magazine 2 spindle tool number (in use)	D1378	The spindle tool number currently used in tool magazine 2.	R
Feedrate	D1379	Accesses the feedrate during cutting.	R
1st spindle actual speed (32-bit)	D1380	Accesses the 1st spindle actual speed. D1381 is the high word of this 32-bit register. You can use Pr399 [Bit 12] to change the value displaying source. Pr399 [Bit 12] = 0: the source is the S code command in the program. Pr399 [Bit 12] = 1: the source is the spindle current speed.	R
Current G code (G01, G02, or G03)	D1383	When you use G01, G02, or G03, this special D register displays the corresponding number of the G code. (G01 = 1, G02 = 2, G03 = 3)	R
X axis machine coordinate	D1384	X axis machine coordinate. This is a 32-bit special D register.	R
Y axis machine coordinate	D1386	Y axis machine coordinate. This is a 32-bit special D register.	R
Z axis machine coordinate	D1388	Z axis machine coordinate. This is a 32-bit special D register.	R
A axis machine coordinate	D1390	A axis machine coordinate. This is a 32-bit special D register.	R
B axis machine coordinate	D1392	B axis machine coordinate. This is a 32-bit special D register.	R
C axis machine coordinate	D1394	C axis machine coordinate. This is a 32-bit special D register.	R
U axis machine coordinate	D1396	U axis machine coordinate. This is a 32-bit special D register.	R
V axis machine coordinate	D1398	V axis machine coordinate. This is a 32-bit special D register.	R
W axis machine coordinate	D1400	W axis machine coordinate. This is a 32-bit special D register.	R
X axis absolute coordinate	D1402	X axis absolute coordinate. The 32-bit floating-point number comprises D1402 and D1403. D1402 is the low word and D1403 is the high word.	R
Y axis absolute coordinate	D1404	Y axis absolute coordinate. The 32-bit floating-point number comprises D1404 and D1405. D1404 is the low word and D1405 is the high word.	R
Z axis absolute coordinate	D1406	Z axis absolute coordinate. The 32-bit floating-point number comprises D1406 and D1407. D1406 is the low word and D1407 is the high word.	R
A axis absolute coordinate	D1408	A axis absolute coordinate. The 32-bit floating-point number comprises D1408 and D1409. D1408 is the low word and D1409 is the high word.	R
B axis absolute coordinate	D1410	B axis absolute coordinate. The 32-bit floating-point number comprises D1410 and D1411. D1410 is the low word and D1411 is the high word.	R

Function name	Special D	Description	Device type
C axis absolute coordinate	D1412	C axis absolute coordinate. The 32-bit floating-point number comprises D1412 and D1413. D1412 is the low word and D1413 is the high word.	R
U axis absolute coordinate	D1414	U axis absolute coordinate. The 32-bit floating-point number comprises D1414 and D1415. D1414 is the low word and D1415 is the high word.	R
V axis absolute coordinate	D1416	V axis absolute coordinate. The 32-bit floating-point number comprises D1416 and D1417. D1416 is the low word and D1417 is the high word.	R
W axis absolute coordinate	D1418	W axis absolute coordinate. The 32-bit floating-point number comprises D1418 and D1419. D1418 is the low word and D1419 is the high word.	R
X axis DMCNET current monitoring	D1420	X axis DMCNET current monitoring. The data is a 16-bit integer. (The data is not updated in HOME mode)	R
Y axis DMCNET current monitoring	D1421	Y axis DMCNET current monitoring. The data is a 16-bit integer. (The data is not updated in HOME mode)	R
Z axis DMCNET current monitoring	D1422	Z axis DMCNET current monitoring. The data is a 16-bit integer. (The data is not updated in HOME mode)	R
A axis DMCNET current monitoring	D1423	A axis DMCNET current monitoring. The data is a 16-bit integer. (The data is not updated in HOME mode)	R
B axis DMCNET current monitoring	D1424	B axis DMCNET current monitoring. The data is a 16-bit integer. (The data is not updated in HOME mode)	R
C axis DMCNET current monitoring	D1425	C axis DMCNET current monitoring. The data is a 16-bit integer. (The data is not updated in HOME mode)	R
U axis DMCNET current monitoring	D1426	U axis DMCNET current monitoring. The data is a 16-bit integer. (The data is not updated in HOME mode)	R
V axis DMCNET current monitoring	D1427	V axis DMCNET current monitoring. The data is a 16-bit integer. (The data is not updated in HOME mode)	R
W axis DMCNET current monitoring	D1428	W axis DMCNET current monitoring. The data is a 16-bit integer. (The data is not updated in HOME mode)	R
SP1 DMCNET current monitoring	D1429	SP1 DMCNET current monitoring. The data is a 16-bit integer. (The data is not updated in spindle positioning mode)	R
SP2 DMCNET current monitoring	D1430	SP2 DMCNET current monitoring. The data is a 16-bit integer. (The data is not updated in spindle positioning mode)	R
Workpiece coordinate system	D1450	Displays the current workpiece coordinate system. Format: GXXPXX. For example, G55 = 5500, G54P01 = 5401.	R
2nd spindle actual speed (32-bit)	D1458	Accesses the 2nd spindle actual speed. D1359 is the high word of this 32-bit register. You can use Pr439 [Bit 12] to change the value displaying source. Pr439 [Bit 12] = 0: the source is the S code command in the program. Pr439 [Bit 12] = 1: the source is the spindle current speed.	R

Function name	Special D	Description	Device type
2nd spindle command speed (32-bit)	D1460	When the 2nd spindle S code is executed in a program, the value is stored in both registers D1462 and D1460. When the value of the 2nd spindle command speed exceed 16 bits, you cannot access the 2nd spindle command speed with D1462. In this case, you can access the actual spindle command speed with D1460. Unit: rpm. D1021 is the high word of this 32-bit register.	R
2nd spindle S code data	D1462	When the 2nd spindle S code is executed in a program, the value of the S code is stored in register D1462. When the S code is used for macro call, the value in this special D register does not change. Unit: rpm.	R
Use ADC, TAD, or DAC	D1464 - D1495	Function of these special D registers changes according to the RIO setting in the NC system. When the RIO is set for ADC, the four IN ports of the ADC stations (4 - 7) correspond to the following four sets of special D codes. D1464 - D1467, D1472 - D1475, D1480 - D1483, D1488 - D1491. When the RIO is set for TAD, the four IN ports of the TAD stations (4 - 7) correspond to the following four sets of special D codes. D1464 - D1471, D1472 - D1479, D1480 - D1487, D1488 - D1491. When the RIO is set for DAC, the four IN ports of the DAC stations (4 - 7) correspond to the following four sets of special D codes. D1464 - D1467, D1472 - D1475, D1480 - D1483, D1488 - D1491.	R/W
User-defined hardware signal	D1500	Setting each bit of this special D register to define the positive and negative limits and the home signals of the NC system.	W
User-defined hardware signal	D1501	Setting each bit of this special D register to define the positive and negative limits and the home signals of the NC system.	W
User-defined hardware signal	D1502	Setting each bit of this special D register to define the positive and negative limits and the home signals of the NC system.	W
User-defined hardware signal	D1503	Setting each bit of this special D register to define the positive and negative limits and the home signals of the NC system.	W
X axis residual coordinate	D1506	X axis residual coordinate. The 32-bit floating-point number comprises D1506 and D1507. D1506 is the low word and D1507 is the high word.	R
Y axis residual coordinate	D1508	Y axis residual coordinate. The 32-bit floating-point number comprises D1508 and D1509. D1508 is the low word and D1509 is the high word.	R
Z axis residual coordinate	D1510	Z axis residual coordinate. The 32-bit floating-point number comprises D1510 and D1511. D1510 is the low word and D1511 is the high word.	R
A axis residual coordinate	D1512	A axis residual coordinate. The 32-bit floating-point number comprises D1512 and D1513. D1512 is the low word and D1513 is the high word.	R
B axis residual coordinate	D1514	B axis residual coordinate. The 32-bit floating-point number comprises D1514 and D1515. D1514 is the low word and D1515 is the high word.	R
C axis residual coordinate	D1516	C axis residual coordinate. The 32-bit floating-point number comprises D1516 and D1517. D1516 is the low word and D1517 is the high word.	R

Function name	Special D	Description	Device type
U axis residual coordinate	D1518	U axis residual coordinate. The 32-bit floating-point number comprises D1518 and D1519. D1518 is the low word and D1519 is the high word.	R
V axis residual coordinate	D1520	V axis residual coordinate. The 32-bit floating-point number comprises D1520 and D1521. D1520 is the low word and D1521 is the high word.	R
W axis residual coordinate	D1522	W axis residual coordinate. The 32-bit floating-point number comprises D1522 and D1523. D1522 is the low word and D1523 is the high word.	R
Servo magazine 1 command tool number	D1524	Command tool number of the 1st servo magazine.	R/W
Servo magazine 1 command character	D1525	Command character of the 1st servo magazine.	R/W
Servo magazine 1 tool number feedback	D1526	Displays the tool number of the 1st servo magazine.	R/W
Servo magazine 1 status feedback	D1527	Displays the current status of the 1st servo magazine.	R/W
Servo magazine 2 command tool number	D1528	Command tool number of the 2nd servo magazine.	R/W
Servo magazine 2 command character	D1529	Command character of the 2nd servo magazine.	R/W
Servo magazine 2 tool number feedback	D1530	Displays the tool number of the 2nd servo magazine.	R/W
Servo magazine 2 status feedback	D1531	Displays the current status of the 2nd servo magazine.	R/W
Servo magazine 3 command tool number	D1532	Command tool number of the 3rd servo magazine.	R/W
Servo magazine 3 command character	D1533	Command character of the 3rd servo magazine.	R/W
Servo magazine 3 tool number feedback	D1534	Displays the tool number of the 3rd servo magazine.	R/W
Servo magazine 3 status feedback	D1535	Displays the current status of the 3rd servo magazine.	R/W

5.3 Special M and special D functions

5.3.1 Mode switching

The controller provides 7 system modes which can be switched by setting the MLC special M relay to on or off. The special M relays related to system modes are as follows.

Function name	Special M	Function name	Special M
System mode selection: 0: auto execution (AUTO) 1: program edit (EDIT) 2: manual input (MDI) 3: MPG feeding (MPG) 4: jog feeding (JOG) 5: rapid feeding (RAPID) 6: homing (HOME)	M1056 M1057 M1058 M1059	MPG feeding (MPG)	M2244
Auto execution (AUTO)	M2241	Jog (JOG)	M2245
Program edit (EDIT)	M2242	Rapid feeding (RAPID)	M2246
Manual input (MDI)	M2243	Homing (HOME)	M2247

5.3.2 Processing

The controller provides various special M relays and special D registers related to the machining statuses. You can determine the current machining status of the controller with these special M and special D, and program the ladder diagram for function protection or executing specific actions.

Function name	Special M	Function name	Special M
Single block execution	M1060	Single block execution	M2249
Cycle Start	M1061	Program in execution	M2250
NC pause	M1062	Pause	M2251
Dummy execution	M1065	M00 program stop	M2252
Optional stop (M01 program stop)	M1066	M01 program stop (optional)	M2253
Single block skip (‘/’)	M1067	M02 end of program	M2254
Main program stops at M99	M1077	M30 end of program and returns	M2255
M96 program interruption	M1078	Spindle is in the rigid tapping mode	M2259
MPG simulation	M1080	Rigid tapping interruption	M2260
NC system macro alarm occurs	M2113	Program ends	M2271
M96 (program interruption) in execution	M2216	File queue auto processing	M2980
System stops at M99	M2238	-	-

Function name	Special D	Function name	Special D
Number of the processing complete pieces (32-bit)	D1018	Rapid movement speed adjustment	D1058
Number of the processing target (32-bit)	D1020	Equi-slope filter compensation for X axis	D1126
Number of the processing complete pieces	D1022	Equi-slope filter compensation for Y axis	D1128
Number of the processing target	D1023	Equi-slope filter compensation for Z axis	D1130
Cutting feedrate adjustment	D1056	Equi-slope filter compensation for A axis	D1132
Equi-slope filter compensation for B axis	D1134	Rotation axis compensation speed	D1146
Equi-slope filter compensation for C axis	D1136	Arc preview speed	D1150
Equi-slope filter compensation for U axis	D1138	Feedrate	D1379
Equi-slope filter compensation for V axis	D1140	Current G code (G01, G02, or G03)	D1383
Equi-slope filter compensation for W axis	D1142	Workpiece coordinate system	D1450
Linear axis compensation speed	D1144	-	-

5.3.3 Axis status

The controller provides the corresponding special M relays and special D registers for each axis. With these special M relays and special D registers, you can lock the axis, cancel the 1st software limit, or access the status, position, or current data of the axis.

Function name	Special M	Function name	Special M
Lock all axes movements	M1068	Port 8 positive hardware limit	M2172
Lock Z axis movement	M1069	Port 8 negative hardware limit	M2173
Ignore axis limit	M1070	Port 8 home signal	M2174
Cancel X axis 1st software limit	M1248	Port 9 positive hardware limit	M2176
Cancel Y axis 1st software limit	M1249	Port 9 negative hardware limit	M2177
Cancel Z axis 1st software limit	M1250	Port 9 home signal	M2178
Cancel A axis 1st software limit	M1251	X axis positioned at the 2nd reference point	M2286
Cancel B axis 1st software limit	M1252	Y axis positioned at the 2nd reference point	M2287
Cancel C axis 1st software limit	M1253	Z axis positioned at the 2nd reference point	M2288
Cancel U axis 1st software limit	M1254	A axis positioned at the 2nd reference point	M2289
Cancel V axis 1st software limit	M1255	B axis positioned at the 2nd reference point	M2290
Cancel W axis 1st software limit	M1256	C axis positioned at the 2nd reference point	M2291
Lock X axis	M1257	U axis positioned at the 2nd reference point	M2292
Lock Y axis	M1258	V axis positioned at the 2nd reference point	M2293
Lock Z axis	M1259	W axis positioned at the 2nd reference point	M2294
Lock A axis	M1260	X axis positioned at the 3rd reference point	M2295
Lock B axis	M1261	Y axis positioned at the 3rd reference point	M2296
Lock C axis	M1262	Z axis positioned at the 3rd reference point	M2297
Lock U axis	M1263	A axis positioned at the 3rd reference point	M2298
Lock V axis	M1264	B axis positioned at the 3rd reference point	M2299
Lock W axis	M1265	C axis positioned at the 3rd reference point	M2300
X axis Servo Off	M1266	U axis positioned at the 3rd reference point	M2301
Y axis Servo Off	M1267	V axis positioned at the 3rd reference point	M2302
Z axis Servo Off	M1268	W axis positioned at the 3rd reference point	M2303
A axis Servo Off	M1269	X axis is moving	M2320
B axis Servo Off	M1270	Y axis is moving	M2321
C axis Servo Off	M1271	Z axis is moving	M2322

Function name	Special M	Function name	Special M
U axis Servo Off	M1272	A axis is moving	M2323
V axis Servo Off	M1273	B axis is moving	M2324
W axis Servo Off	M1274	C axis is moving	M2325
Permit X axis movement	M1312	U axis is moving	M2326
Permit Y axis movement	M1313	V axis is moving	M2327
Permit Z axis movement	M1314	W axis is moving	M2328
Permit A axis movement	M1315	X axis is moving in forward direction	M2336
Permit B axis movement	M1316	Y axis is moving in forward direction	M2337
Permit C axis movement	M1317	Z axis is moving in forward direction	M2338
Permit U axis movement	M1318	A axis is moving in forward direction	M2339
Permit V axis movement	M1319	B axis is moving in forward direction	M2340
Permit W axis movement	M1320	C axis is moving in forward direction	M2341
Lock X axis movement in positive direction	M1344	U axis is moving in forward direction	M2342
Lock Y axis movement in positive direction	M1345	V axis is moving in forward direction	M2343
Lock Z axis movement in positive direction	M1346	W axis is moving in forward direction	M2344
Lock A axis movement in positive direction	M1347	X axis is moving in reverse direction	M2345
Lock B axis movement in positive direction	M1348	Y axis is moving in reverse direction	M2346
Lock C axis movement in positive direction	M1349	Z axis is moving in reverse direction	M2347
Lock U axis movement in positive direction	M1350	A axis is moving in reverse direction	M2348
Lock V axis movement in positive direction	M1351	B axis is moving in reverse direction	M2349
Lock W axis movement in positive direction	M1352	C axis is moving in reverse direction	M2350
Lock X axis movement in negative direction	M1353	U axis is moving in reverse direction	M2351
Lock Y axis movement in negative direction	M1354	V axis is moving in reverse direction	M2352
Lock Z axis movement in negative direction	M1355	W axis is moving in reverse direction	M2353
Lock A axis movement in negative direction	M1356	X axis positioned at the 4th reference point	M2368
Lock B axis movement in negative direction	M1357	Y axis positioned at the 4th reference point	M2369
Lock C axis movement in negative direction	M1358	Z axis positioned at the 4th reference point	M2370
Lock U axis movement in negative direction	M1359	A axis positioned at the 4th reference point	M2371
Lock V axis movement in negative direction	M1360	B axis positioned at the 4th reference point	M2372
Lock W axis movement in negative direction	M1361	C axis positioned at the 4th reference point	M2373

Function name	Special M	Function name	Special M
X axis at the origin	M2119	U axis positioned at the 4th reference point	M2374
Y axis at the origin	M2120	V axis positioned at the 4th reference point	M2375
Z axis at the origin	M2121	W axis positioned at the 4th reference point	M2376
A axis at the origin	M2122	X axis homing complete	M2377
B axis at the origin	M2123	Y axis homing complete	M2378
C axis at the origin	M2124	Z axis homing complete	M2379
U axis at the origin	M2125	A axis homing complete	M2380
V axis at the origin	M2126	B axis homing complete	M2381
W axis at the origin	M2127	C axis homing complete	M2382
Port 1 positive hardware limit	M2144	U axis homing complete	M2383
Port 1 negative hardware limit	M2145	V axis homing complete	M2384
Port 1 home signal	M2146	W axis homing complete	M2385
Port 2 positive hardware limit	M2148	X axis Servo On / Off	M2386
Port 2 negative hardware limit	M2149	Y axis Servo On / Off	M2387
Port 2 home signal	M2150	Z axis Servo On / Off	M2388
Port 3 positive hardware limit	M2152	A axis Servo On / Off	M2389
Port 3 negative hardware limit	M2153	B axis Servo On / Off	M2390
Port 3 home signal	M2154	C axis Servo On / Off	M2391
Port 4 positive hardware limit	M2156	U axis Servo On / Off	M2392
Port 4 negative hardware limit	M2157	V axis Servo On / Off	M2393
Port 4 home signal	M2158	W axis Servo On / Off	M2394
Port 5 positive hardware limit	M2160	Movement in X axis	M2400
Port 5 negative hardware limit	M2161	Movement in Y axis	M2401
Port 5 home signal	M2162	Movement in Z axis	M2402
Port 6 positive hardware limit	M2164	Movement in A axis	M2403
Port 6 negative hardware limit	M2165	Movement in B axis	M2404
Port 6 home signal	M2166	Movement in C axis	M2405
Port 7 positive hardware limit	M2168	Movement in U axis	M2406
Port 7 negative hardware limit	M2169	Movement in V axis	M2407
Port 7 home signal	M2170	Movement in W axis	M2408

Function name	Special D code	Function name	Special D code
X axis machine coordinate	D1384	Y axis DMCNET current monitoring	D1421
Y axis machine coordinate	D1386	Z axis DMCNET current monitoring	D1422
Z axis machine coordinate	D1388	A axis DMCNET current monitoring	D1423
A axis machine coordinate	D1390	B axis DMCNET current monitoring	D1424
B axis machine coordinate	D1392	C axis DMCNET current monitoring	D1425
C axis machine coordinate	D1394	U axis DMCNET current monitoring	D1426
U axis machine coordinate	D1396	V axis DMCNET current monitoring	D1427
V axis machine coordinate	D1398	W axis DMCNET current monitoring	D1428
W axis machine coordinate	D1400	SP1 DMCNET current monitoring	D1429
X axis absolute coordinate	D1402	SP2 DMCNET current monitoring	D1430
Y axis absolute coordinate	D1404	X axis residual coordinate	D1506
Z axis absolute coordinate	D1406	Y axis residual coordinate	D1508
A axis absolute coordinate	D1408	Z axis residual coordinate	D1510
B axis absolute coordinate	D1410	A axis residual coordinate	D1512
C axis absolute coordinate	D1412	B axis residual coordinate	D1514
U axis absolute coordinate	D1414	C axis residual coordinate	D1516
V axis absolute coordinate	D1416	U axis residual coordinate	D1518
W axis absolute coordinate	D1418	V axis residual coordinate	D1520
X axis DMCNET current monitoring	D1420	W axis residual coordinate	D1522

5.3.4 Homing

The controller homing process is triggered by special M relays. You can determine the homing sequence according to the machine design.

Function name	Special M	Function name	Special M
X axis homing control	M1236	X axis homing complete	M2272
Y axis homing control	M1237	Y axis homing complete	M2273
Z axis homing control	M1238	Z axis homing complete	M2274
A axis homing control	M1239	A axis homing complete	M2275
B axis homing control	M1240	B axis homing complete	M2276
C axis homing control	M1241	C axis homing complete	M2277
U axis homing control	M1242	U axis homing complete	M2278
V axis homing control	M1243	V axis homing complete	M2279
W axis homing control	M1244	W axis homing complete	M2280

5.3.5 Jog operation

Jog operation and jogging speed of each axis are determined by special M relays and special D registers. You can change the speed and direction (forward or reverse) with the ladder diagram at all times.

Note: set the system mode to JOG mode for these special M relays and special D registers to take effect.

Function name	Special M	Function name	Special M
X axis forward jog control	M1216	X axis reverse jog control	M1226
Y axis forward jog control	M1217	Y axis reverse jog control	M1227
Z axis forward jog control	M1218	Z axis reverse jog control	M1228
A axis forward jog control	M1219	A axis reverse jog control	M1229
B axis forward jog control	M1220	B axis reverse jog control	M1230
C axis forward jog control	M1221	C axis reverse jog control	M1231
U axis forward jog control	M1222	U axis reverse jog control	M1232
V axis forward jog control	M1223	V axis reverse jog control	M1233
W axis forward jog control	M1224	W axis reverse jog control	M1234

Function name	Special D	Function name	Special D
Speed setting for jog feeding and dry run	D1062	-	-

5.3.6 MPG

MPG axis switching and pulse magnification are set with special D registers. You can also send pulse signals with special M relays.

Function name	Special M	Function name	Special M
Panel MPG pulse +	M1118	MPG in forward operation	M2232
Panel MPG pulse -	M1119	MPG in reverse operation	M2233

Function name	Special D	Function name	Special D
MPG operation mode number	D1040	MPG pulse magnification	D1042
MPG operation channel selection	D1041	Select the axis with MPG	D1043

5.3.7 G31

When you use the G31 Skip command, the system sends the special M signals for you to check if the command signal is actually triggered.

Function name	Special M	Function name	Special M
HSI 1	M2142	HSI 2	M2143

5.3.8 One-key macro call

Pressing one key on the controller can call a specific program for machining.

Note: set the system mode to AUTO mode for these functions to take effect.

Function name	Special M	Function name	Special M
Macro call initial preparation	M1074	Macro call in execution	M2225
Macro call activation	M1075	Macro call error	M2226
Macro call initial preparation complete	M2224	-	-

Function name	Special D	Function name	Special D
Macro call file name	D1111	-	-

5.3.9 MLC axis

In addition to the jog operation in JOG mode or the programmed movement in AUTO mode, you can move the axis to a specific position or set the axis to constantly move at a specific speed with MLC.

Function name	Special M	Function name	Special M
Activate X axis (MLC axis)	M1184	Switch MLC X axis command to incremental	M1280
Activate Y axis (MLC axis)	M1185	Switch MLC Y axis command to incremental	M1281
Activate Z axis (MLC axis)	M1186	Switch MLC Z axis command to incremental	M1282
Activate A axis (MLC axis)	M1187	Switch MLC A axis command to incremental	M1283
Activate B axis (MLC axis)	M1188	Switch MLC B axis command to incremental	M1284
Activate C axis (MLC axis)	M1189	Switch MLC C axis command to incremental	M1285
Activate U axis (MLC axis)	M1190	Switch MLC U axis command to incremental	M1286
Activate V axis (MLC axis)	M1191	Switch MLC V axis command to incremental	M1287
Activate W axis (MLC axis)	M1192	Switch MLC W axis command to incremental	M1288
Activate the spindle (MLC axis)	M1193	MLC X axis control mode	M1289
MLC axis incremental motion command	M1194	MLC Y axis control mode	M1290
NC / MLC axis switching (X axis)	M1200	MLC Z axis control mode	M1291
NC / MLC axis switching (Y axis)	M1201	MLC A axis control mode	M1292
NC / MLC axis switching (Z axis)	M1202	MLC B axis control mode	M1293
NC / MLC axis switching (A axis)	M1203	MLC C axis control mode	M1294
NC / MLC axis switching (B axis)	M1204	MLC U axis control mode	M1295
NC / MLC axis switching (C axis)	M1205	MLC V axis control mode	M1296
NC / MLC axis switching (U axis)	M1206	MLC W axis control mode	M1297
NC / MLC axis switching (V axis)	M1207	MLC X axis high-speed input point triggering	M1298
NC / MLC axis switching (W axis)	M1208	MLC Y axis high-speed input point triggering	M1299

Function name	Special M	Function name	Special M
MLC Z axis high-speed input point triggering	M1300	MLC Y axis positioning complete	M2305
MLC A axis high-speed input point triggering	M1301	MLC Z axis positioning complete	M2306
MLC B axis high-speed input point triggering	M1302	MLC A axis positioning complete	M2307
MLC C axis high-speed input point triggering	M1303	MLC B axis positioning complete	M2308
MLC U axis high-speed input point triggering	M1304	MLC C axis positioning complete	M2309
MLC V axis high-speed input point triggering	M1305	MLC U axis positioning complete	M2310
MLC W axis high-speed input point triggering	M1306	MLC V axis positioning complete	M2311
MLC X axis high-speed input point triggering feedback	M2128	MLC W axis positioning complete	M2312
MLC Y axis high-speed input point triggering feedback	M2129	NC / MLC axis switching complete (X axis)	M2354
MLC Z axis high-speed input point triggering feedback	M2130	NC / MLC axis switching complete (Y axis)	M2355
MLC A axis high-speed input point triggering feedback	M2131	NC / MLC axis switching complete (Z axis)	M2356
MLC B axis high-speed input point triggering feedback	M2132	NC / MLC axis switching complete (A axis)	M2357
MLC C axis high-speed input point triggering feedback	M2133	NC / MLC axis switching complete (B axis)	M2358
MLC U axis high-speed input point triggering feedback	M2134	NC / MLC axis switching complete (C axis)	M2359
MLC V axis high-speed input point triggering feedback	M2135	NC / MLC axis switching complete (U axis)	M2360
MLC W axis high-speed input point triggering feedback	M2136	NC / MLC axis switching complete (V axis)	M2361
MLC X axis positioning complete	M2304	NC / MLC axis switching complete (W axis)	M2362

Function name	Special D	Function name	Special D
MLC X axis positioning command	D1064	MLC X axis positioning speed	D1082
MLC Y axis positioning command	D1066	MLC Y axis positioning speed	D1084
MLC Z axis positioning command	D1068	MLC Z axis positioning speed	D1086
MLC A axis positioning command	D1070	MLC A axis positioning speed	D1088
MLC B axis positioning command	D1072	MLC B axis positioning speed	D1090
MLC C axis positioning command	D1074	MLC C axis positioning speed	D1092
MLC U axis positioning command	D1076	MLC U axis positioning speed	D1094
MLC V axis positioning command	D1078	MLC V axis positioning speed	D1096
MLC W axis positioning command	D1080	MLC W axis positioning speed	D1098
-	-	MLC spindle positioning speed	D1100

5.3.10 Teaching mode

The NC provides the teaching mode for easier programming in JOG mode. You can also perform programming with these MLC special M relays.

Function name	Special M	Function name	Special M
G00 teaching triggered	M2992	G00 teaching record complete	M2994
G01 teaching triggered	M2993	G01 teaching record complete	M2995

5.3.11 M, S, T codes

When the M, S, and T codes are executed in the program, the NC system sends the corresponding special M signals to the MLC. For example, when M03 is executed in the program, M2208 is set to on, and the data is written to the corresponding special D register. The following table lists the corresponding special M relays and special D registers of the M, S, T codes.

Function name	Special M	Function name	Special M
Lock M, S, and T codes	M1071	S code execution	M2209
Notification of MST code execution complete	M1152	T code execution	M2210
M code execution	M2208	-	-

Function name	Special D	Function name	Special D
M code data	D1368	1st spindle S code data	D1369
T code data (command)	D1370	2nd spindle S code data	D1462

5.3.12 Synchronous control

The controller provides the synchronous axis control functions, which are triggered with these special M relays.

Function name	Special M	Function name	Special M
Trigger for synchronous control	M1088	C slave axis follows the master axis	M1094
X slave axis follows the master axis	M1089	U slave axis follows the master axis	M1095
Y slave axis follows the master axis	M1090	V slave axis follows the master axis	M1096
Z slave axis follows the master axis	M1091	W slave axis follows the master axis	M1097
A slave axis follows the master axis	M1092	Synchronous function in execution	M2227
B slave axis follows the master axis	M1093	-	-

5.3.13 Command transfer

The controller provides the transfer functions for the axis commands, which are triggered with these special M relays.

Function name	Special M	Function name	Special M
Trigger for command transfer	M1098	C axis receives command from the master axis	M1104
X axis receives command from the master axis	M1099	U axis receives command from the master axis	M1105
Y axis receives command from the master axis	M1100	V axis receives command from the master axis	M1106
Z axis receives command from the master axis	M1101	W axis receives command from the master axis	M1107
A axis receives command from the master axis	M1102	Transfer function in execution	M2228
B axis receives command from the master axis	M1103	-	-

5.3.14 Spindle

You can operate the spindle with the following special M relays and adjust the speed and magnification with the following special D registers.

Function name	Special M	Function name	Special M
1st spindle forward rotation	M1120	Switch the spindle speed source	M1307
1st spindle reverse rotation	M1121	Lathe C / S axis switching	M2239
1st spindle gear ratio selection	M1122 M1123	1st spindle reaches the target speed	M2256
1st spindle positioning control	M1124	1st spindle reaches zero speed	M2257
1st spindle retraction after tapping	M1125	1st spindle positioning complete	M2258
Lathe spindle C / S axis switching	M1126	1st spindle is in the rigid tapping mode	M2259
1st spindle analog voltage proportional gain	M1127	1st spindle rigid tapping interruption	M2260
2nd spindle forward rotation	M1136	2nd spindle reaches the target speed	M2261
2nd spindle reverse rotation	M1137	2nd spindle reaches zero speed	M2262
2nd spindle gear ratio selection	M1138 M1139	2nd spindle positioning complete	M2263
2nd spindle positioning control	M1140	2nd spindle is in the rigid tapping mode	M2264
2nd spindle retraction after tapping	M1141	2nd spindle rigid tapping interruption	M2265
2nd spindle analog voltage proportional gain	M1143	-	-

Function name	Special D	Function name	Special D
1st spindle speed adjustment	D1060	2nd spindle speed (written with special D register)	D1152
2nd spindle speed adjustment	D1112	1st spindle command speed (32-bit)	D1364
Spindle analog voltage output port 2	D1114	1st spindle actual speed (32-bit)	D1380
Spindle analog voltage output port 1	D1125	2nd spindle actual speed (32-bit)	D1458
1st spindle speed (written with special D register)	D1148	2nd spindle command speed (32-bit)	D1460

5.3.15 Tool magazine

The controller provides the tool magazine management functions. Aside from using macros for tool change, you need to exchange tool data with the MLC to ensure the data in the controller is correct.

Function name	Special M	Function name	Special M
Tool magazine 1 moves forward	M1168	Tool magazine 2 moves backward	M1173
Tool magazine 1 moves backward	M1169	Tool 2 exchange	M1174
Tool 1 exchange	M1170	Tool magazine 2 reset	M1175
Tool magazine 1 reset	M1171	Tool magazine 1 reset complete	M2212
Tool magazine 2 moves forward	M1172	Tool magazine 2 reset complete	M2213

Function name	Special D	Function name	Special D
Lathe tool number selection	D1115	Servo magazine 1 command tool number	D1524
Spindle tool number (written with special D register)	D1172	Servo magazine 1 command character	D1525
Standby tool number (written with special D register)	D1173	Servo magazine 1 tool number feedback	D1526
Command tool number (written with special D register)	D1174	Servo magazine 1 status feedback	D1527
Tool magazine 1 T code data (standby)	D1371	Servo magazine 2 command tool number	D1528
Tool magazine 1 tool pot deviation	D1372	Servo magazine 2 command character	D1529
Tool magazine 1 tool pot number (standby)	D1373	Servo magazine 2 tool number feedback	D1530
Tool magazine 1 spindle tool number (in use)	D1374	Servo magazine 2 status feedback	D1531
Tool magazine 2 T code data (standby)	D1375	Servo magazine 3 command tool number	D1532
Tool magazine 2 tool pot deviation	D1376	Servo magazine 3 command character	D1533
Tool magazine 2 tool pot number (standby)	D1377	Servo magazine 3 tool number feedback	D1534
Tool magazine 2 spindle tool number (in use)	D1378	Servo magazine 3 status feedback	D1535

5.3.16 NC system action

The following are the special M relays and special D registers which are commonly used to indicate NC system information and actions.

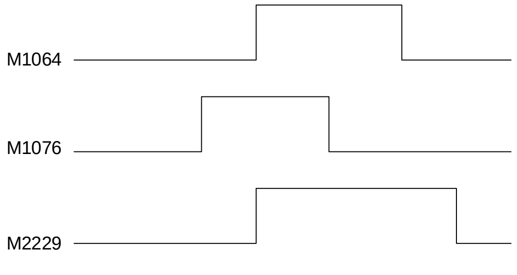
Function name	Special M	Function name	Special M
System stop	M1063	NC system emergency stop	M2114
NC system reset	M1064	G code ready	M2223
NC system reset	M1076	NC system reset complete	M2229
MLC emergency stop	M1079	Channel alarm message	M2240
Disable inhibit zone protection	M1085	Lock user permission	M2934
Machine started and NC system ready	M2112	Restrict program editing	M2935

Function name	Special D	Function name	Special D
DMCNET ready for HMI	D1048	System time: minute, second	D1109
Main file name	D1102	Use ADC, TAD, or DAC	D1464 - D1495
Total machining time	D1103, D1104	User-defined hardware signal	D1500
Single machining time	D1105, D1106	User-defined hardware signal	D1501
System time: year, month	D1107	User-defined hardware signal	D1502
System time: date, hour	D1108	User-defined hardware signal	D1503

Note:

Time sequence of the M1064, M1076, and M2229 actions.

When you trigger M1076 to reset the system, M1076 is set to on to inform the system of resetting. When the system starts resetting, M1076 is set to off and M1064 becomes off as well. After resetting, the system sets M2229 to on. And then, then system sets M2229 to off after 1 second.



5.3.17 DMCNET connection

The controller provides the function of monitoring the servo connection status for you to program the related alarms and actions.

Function name	Special M	Function name	Special M
DMCNET connection successful	M1072	DMCNET connection status - Station 6	M2869
Servo enabled	M2115	DMCNET connection status - Station 7	M2870
DMCNET connection status - Station 1	M2864	DMCNET connection status - Station 8	M2871
DMCNET connection status - Station 2	M2865	DMCNET connection status - Station 9	M2872
DMCNET connection status - Station 3	M2866	DMCNET connection status - Station 10	M2873
DMCNET connection status - Station 4	M2867	DMCNET connection status - Station 11	M2874
DMCNET connection status - Station 5	M2868	DMCNET connection status - Station 12	M2875

5.3.18 MLC interrupt program

MLC provides the interrupt inputs for you to program the corresponding actions based on the requirements of time sequence.

Function name	Special M	Function name	Special M
IX00 interrupt input	M2880	IR11 interrupt input	M2907
IX01 interrupt input	M2881	IR12 interrupt input	M2908
IX02 interrupt input	M2882	IR13 interrupt input	M2909
IX03 interrupt input	M2883	IR14 interrupt input	M2910
IX04 interrupt input	M2884	IR15 interrupt input	M2911
IX05 interrupt input	M2885	IR16 interrupt input	M2912
IX06 interrupt input	M2886	IR17 interrupt input	M2913
IX07 interrupt input	M2887	IR18 interrupt input	M2914
IC00 interrupt input	M2888	IR19 interrupt input	M2915
IC01 interrupt input	M2889	IR20 interrupt input	M2916
IR11 interrupt input	M2907	IR21 interrupt input	M2917
IR00 interrupt input	M2896	IR22 interrupt input	M2918
IR01 interrupt input	M2897	IR23 interrupt input	M2919
IR02 interrupt input	M2898	IR24 interrupt input	M2920
IR03 interrupt input	M2899	IR25 interrupt input	M2921
IR04 interrupt input	M2900	IR26 interrupt input	M2922
IR05 interrupt input	M2901	IR27 interrupt input	M2923
IR06 interrupt input	M2902	IR28 interrupt input	M2924
IR07 interrupt input	M2903	IR29 interrupt input	M2925
IR08 interrupt input	M2904	IR30 interrupt input	M2926

Function name	Special M	Function name	Special M
IR09 interrupt input	M2905	IR31 interrupt input	M2927
IR10 interrupt input	M2906	-	-

5.3.19 MLC instruction

The following table lists the special M relays for switching the MLC application instructions.

Function name	Special M	Function name	Special M
MLC instruction operation result is 0	M2824	DCNT counter C69 counts down	M2949
MLC instruction borrow operation	M2825	DCNT counter C70 counts down	M2950
MLC instruction carry operation	M2826	DCNT counter C71 counts down	M2951
MLC instruction operation error	M2828	DCNT counter C72 counts down	M2952
DCNT counter C64 counts down	M2944	DCNT counter C73 counts down	M2953
DCNT counter C65 counts down	M2945	DCNT counter C74 counts down	M2954
DCNT counter C66 counts down	M2946	DCNT counter C75 counts down	M2955
DCNT counter C67 counts down	M2947	DCNT counter C76 counts down	M2956
DCNT counter C68 counts down	M2948	DCNT counter C77 counts down	M2957

5.3.20 HMI special M relay output

You can read the values of variables #1801 - #1832 in the NC program to read the status of the HMI output points in the MLC system. Variables #1801 - #1832 are paired with MLC HMI output points M1024 - M1055 respectively. For example, #1801 is paired with M1024. If M1024 is on, the value of the variable #1801 in the NC program is 1. On the other hand, if M1024 is off, the value of #1801 is 0.

Refer to the following table for the bits for the HMI output points in the MLC system and their corresponding variables in the NC system (MLC > NC).

Function name	Special M	Variable	Function name	Special M	Variable
HMI output point 1	M1024	#1801	HMI output point 17	M1040	#1817
HMI output point 2	M1025	#1802	HMI output point 18	M1041	#1818
HMI output point 3	M1026	#1803	HMI output point 19	M1042	#1819
HMI output point 4	M1027	#1804	HMI output point 20	M1043	#1820
HMI output point 5	M1028	#1805	HMI output point 21	M1044	#1821
HMI output point 6	M1029	#1806	HMI output point 22	M1045	#1822
HMI output point 7	M1030	#1807	HMI output point 23	M1046	#1823
HMI output point 8	M1031	#1808	HMI output point 24	M1047	#1824
HMI output point 9	M1032	#1809	HMI output point 25	M1048	#1825
HMI output point 10	M1033	#1810	HMI output point 26	M1049	#1826
HMI output point 11	M1034	#1811	HMI output point 27	M1050	#1827
HMI output point 12	M1035	#1812	HMI output point 28	M1051	#1828
HMI output point 13	M1036	#1813	HMI output point 29	M1052	#1829
HMI output point 14	M1037	#1814	HMI output point 30	M1053	#1830
HMI output point 15	M1038	#1815	HMI output point 31	M1054	#1831
HMI output point 16	M1039	#1816	HMI output point 32	M1055	#1832

5.3.21 HMI special M relay input

You can write the values to variables #1864 - #1895 in the NC program to change the signal status of the HMI input points in the MLC system. Variables #1864 - #1895 are paired with the MLC HMI input points M2080 - M2111 respectively. For example, #1864 is paired with M2080. If you set #1864 to 1 in the NC program, M2080 is on in the MLC program. On the other hand, if you set #1864 to 0 in the NC program, M2080 is off in the MLC program.

Refer to the following table for the bits for HMI input points in the MLC system and their corresponding variables in the NC system (NC > MLC).

Function name	Special M	Variable	Function name	Special M	Variable
HMI input point 1	M2080	#1864	HMI input point 17	M2096	#1880
HMI input point 2	M2081	#1865	HMI input point 18	M2097	#1881
HMI input point 3	M2082	#1866	HMI input point 19	M2098	#1882
HMI input point 4	M2083	#1867	HMI input point 20	M2099	#1883
HMI input point 5	M2084	#1868	HMI input point 21	M2100	#1884
HMI input point 6	M2085	#1869	HMI input point 22	M2101	#1885
HMI input point 7	M2086	#1870	HMI input point 23	M2102	#1886
HMI input point 8	M2087	#1871	HMI input point 24	M2103	#1887
HMI input point 9	M2088	#1872	HMI input point 25	M2104	#1888
HMI input point 10	M2089	#1873	HMI input point 26	M2105	#1889
HMI input point 11	M2090	#1874	HMI input point 27	M2106	#1890
HMI input point 12	M2091	#1875	HMI input point 28	M2107	#1891
HMI input point 13	M2092	#1876	HMI input point 29	M2108	#1892
HMI input point 14	M2093	#1877	HMI input point 30	M2109	#1893
HMI input point 15	M2094	#1878	HMI input point 31	M2110	#1894
HMI input point 16	M2095	#1879	HMI input point 32	M2111	#1895

5.3.22 HMI special D register output

You can write the values to variables #1833 - #1848 in the NC program to change the values of the HMI output points in the MLC system. Variables #1833 - #1848 are paired with the MLC HMI output points D1024 - D1039 respectively. For example, #1833 is paired with D1024. If the output value of D1024 is 100, the value of #1833 is 100. In other words, the value of #1833 changes with the value of D1024.

Refer to the following table for the registers for HMI output points in the MLC system and their corresponding variables in the NC system (MLC > NC).

Function name	Special D	Variable	Function name	Special D	Variable
HMI output point 1	D1024	#1833	HMI output point 9	D1032	#1841
HMI output point 2	D1025	#1834	HMI output point 10	D1033	#1842
HMI output point 3	D1026	#1835	HMI output point 11	D1034	#1843
HMI output point 4	D1027	#1836	HMI output point 12	D1035	#1844
HMI output point 5	D1028	#1837	HMI output point 13	D1036	#1845
HMI output point 6	D1029	#1838	HMI output point 14	D1037	#1846
HMI output point 7	D1030	#1839	HMI output point 15	D1038	#1847
HMI output point 8	D1031	#1840	HMI output point 16	D1039	#1848

5.3.23 HMI special D register input

You can write the values to variables #1896 - #1911 in the NC program to change the values of the HMI input points in the MLC system. Variables #1896 - #1911 are paired with the MLC HMI input points D1336 - D1351 respectively. For example, #1896 is paired with D1336. If the value of #1896 is 101 in the NC program, the value of D1336 in MLC is 101. In other words, the value of D1336 in the MLC program changes with the value of #1896 in the NC system.

Refer to the following table for the registers for HMI input points in the MLC system and their corresponding variables in the NC system (NC > MLC).

Function name	Special D	Variable	Function name	Special D	Variable
HMI input point 1	D1336	#1896	HMI input point 9	D1344	#1904
HMI input point 2	D1337	#1897	HMI input point 10	D1345	#1905
HMI input point 3	D1338	#1898	HMI input point 11	D1346	#1906
HMI input point 4	D1339	#1899	HMI input point 12	D1347	#1907
HMI input point 5	D1340	#1900	HMI input point 13	D1348	#1908
HMI input point 6	D1341	#1901	HMI input point 14	D1349	#1909
HMI input point 7	D1342	#1902	HMI input point 15	D1350	#1910
HMI input point 8	D1343	#1903	HMI input point 16	D1351	#1911

(This page is intentionally left blank.)

MLC Application Examples 6

This chapter provides frequently used MLC applications, which includes examples such as analog spindle gear switch, spindle retraction during tapping, and one-key macro call.

6.1 Mode switch

The NC series provides 7 types of operating mode. You can change between system modes by setting M1056 - M1059 to On / Off with the MLC.

■ MLC special M relays

Mode switch and status devices:

NC system mode switch and corresponding M status							
	0: Auto AUTO	1: Edit EDIT	2: Manual MDI	3: MPG MPG	4: JOG JOG	5: Rapid RAPID	6: Origin HOME
Bit 0	M1056	M1056	M1056	M1056	M1056	M1056	M1056
Bit 1	M1057	M1057	M1057	M1057	M1057	M1057	M1057
Bit 2	M1058	M1058	M1058	M1058	M1058	M1058	M1058
Bit 3	M1059	M1059	M1059	M1059	M1059	M1059	M1059
Mode Feedback	M2241	M2242	M2243	M2245	M2245	M2246	M2247

Note: the special M relay in gray indicates it is Off; the special M relay in black indicates it is On.

[Mode switch]: M1056 - M1059

When you need to change the mode, set **[Mode switch]** special M to On / Off in bit format (as shown in the above table).

✓ When composing ladder diagrams on the MLC, apart from directly setting the corresponding special M to On, you can use MOV K0 - 6 K1M1056 to input the corresponding decimal value, and the system automatically converts the value to binary format.

[Mode feedback]: M2241 - M2247

When the system finishes the mode change, it sends the corresponding **[Mode feedback]** special M.

✓ When the states of **[Mode switch]** special M relays are changed, the NC system sends the corresponding **[Mode feedback]** special M within 4 ms.

■ MLC example

In general applications, press the key or rotate the knob on the mahine operation panel B to trigger the signal for switching the mode. The following examples describe how to write the MLC programs for key-triggering and knob-triggering.

Trigger by key:

The mode switch signal is triggered only when you press the key. Therefore, at each trigger, the MLC writes the corresponding decimal value to K1M1056 with the MOV instruction.

K1 represents a set of binary digits of four special M relays starting from M1056. M1056 - M1059 in binary format are switched to On / Off to have the system change between modes and the **[Mode feedback]** special M relay is used to output signals.

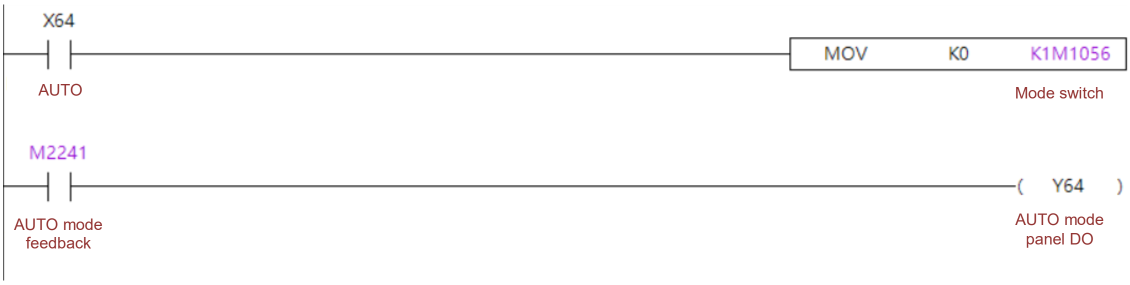
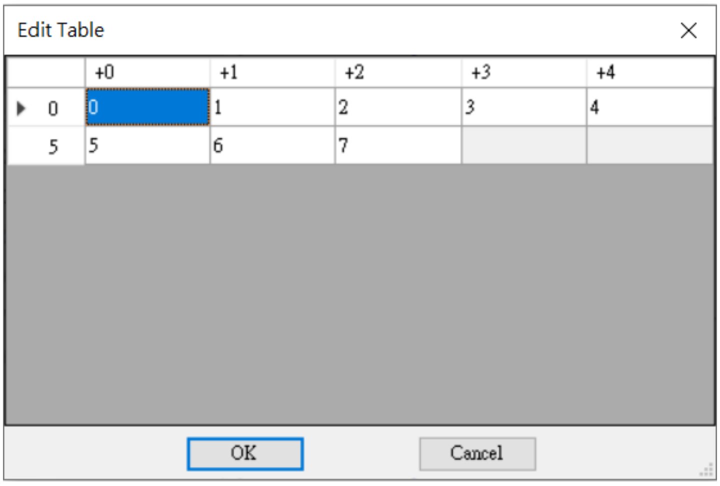


Figure 6.1.1 Mode switching by key

Trigger by knob:

The trigger signals are input to the MLC continuously. Therefore, when you use the VRT instruction to create the table (as shown in Figure 6.1.2) and input the corresponding values to the table, the VRT instruction outputs the corresponding values to the register. Next, the MOV instruction moves the values of the register to K1M1056 to switch the system mode and the system uses the **[Mode feedback]** special M to output signals.



	+0	+1	+2	+3	+4
0	0	1	2	3	4
5	5	6	7		

Figure 6.1.2 VRT table

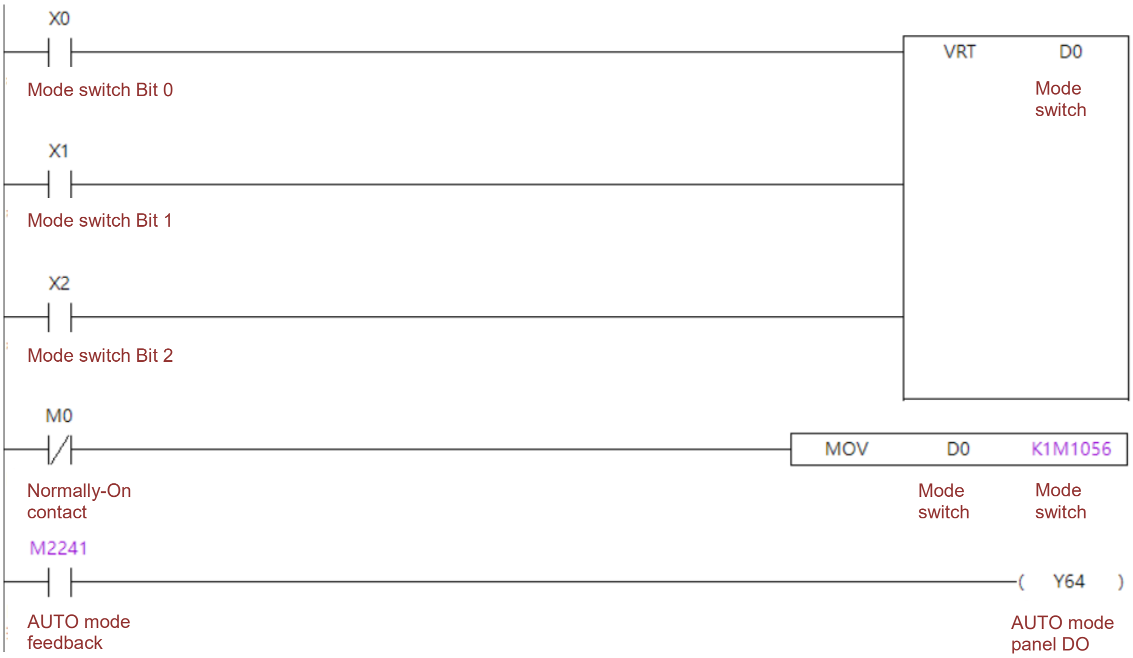


Figure 6.1.3 Mode switching by knob

6.2 Machining, single block function, and feed rate override

On the controller, you can use the special M relays to start and stop the machining. In addition, you can switch to single block execution mode and change the machining speed override setting.

■ **MLC special M relays and special D register**

Single block execution	M1060	Single block pause	M2249
Cycle start	M1061	Program in execution	M2250
Program stop (Feed hold)	M1062	Program pause	M2251
Optional stop	M1066	M01 optional stop	M2253
Single block skip	M1062	M00 Program stop	M2252
MPG simulation	M1080	M02 End of program	M2254
Cutting feed rate override	D1056	M30 End of program and return	M2255

[Single block execution] and [Single block pause]: M1060 and M2249

When the system is in AUTO or MDI mode, you can set **[Single block execution]** to On before the program starts or during program execution, and the controller executes one block at a time.

✓ When **[Single block execution]** is set to On before the program starts, the system executes the blocks in sequence and sets **[Single block pause]** to On each time it finishes executing a block.

✓ When **[Single block execution]** is set to On during program execution, the system finishes executing the current block and stops, and then sets **[Single block pause]** to On.

✓ When the system stops because of **[Single block execution]**, you can set **[Cycle start]** to On to execute the next block.

✓ When you set **[Single block execution]** to Off when the program is stopped, the system executes the following program in sequence after you set **[Cycle start]** to On. Next, the system sets **[Single block pause]** to Off.

[Cycle start] and [Program in execution]: M1061 and M2250

When the system loads the program in AUTO or MDI mode and you set **[Cycle start]** to On, the controller executes the current main file to perform machining, and then sets **[Program in execution]** to On.

✓ When **[Cycle start]** is set to On, the **[Cycle start]** special M relay has to stay On for at least one PLC scanning cycle.

✓ After **[Program in execution]** is set to On, the system will set it to Off in the following conditions.

a. When the system is executing the program, it sets **[Single block execution]** to On and finishes executing a single block.

b. When the system is executing the program and M00, M02, or M30 is executed.

c. When the system is executing the program and the Reset signal is triggered.

d. When the system is executing the program and there is no program to execute next.

e. When the system is executing the program and you change the system mode.

[Program stop (Feed hold)] and [Program pause]: M1062 and M2251

When the system is executing the program in AUTO or MDI mode and you set **[Program stop]** to On, the controller immediately stops the current program execution, keeps the axis at the current coordinates, and sets **[Program pause]** to On.

- ✓ When **[Program stop (Feed hold)]** is set to On, the **[Program stop (Feed hold)]** special M has to stay On for at least one PLC scanning cycle.
- ✓ When **[Program pause]** is set to On, the system will set it to Off in the following conditions.
  a. When **[Cycle start]** is set to On again.
  b. When the Reset signal is triggered.

[Optional stop] and [M01 optional stop]: M1066 and M2253

When the system is executing the program in AUTO or MDI mode and you set **[Optional stop]** to On and M01 is executed, the controller immediately stops the current program execution and sets **[M01 optional stop]** to On.

- ✓ When **[Optional stop]** is set to On and the system executes M01 and then stops the program execution, you can set **[Cycle start]** to On to carry on executing the following program and set **[M01 optional stop]** to Off.
- ✓ When you set **[Optional stop]** to Off when the program is stopped, the system executes the following program after you set **[Cycle start]** to On. Next, the system sets **[M01 optional stop]** to Off.
- ✓ When **[M01 optional stop]** is set to On, the system will set it to Off in the following conditions.
  a. When **[Cycle start]** is set to On again.
  b. When the Reset signal is triggered.
  c. When you change the system to HOME mode.

[Single block skip]: M1062

When the system is executing the program in AUTO or MDI mode and you set **[Single block skip]** to On and there is a “/” symbol in the block, the controller skips this block and directly executes the next block.

[MPG simulation]: M1080

When the system is executing the program in AUTO or MDI mode and you set **[MPG simulation]** to On, all the motion blocks in the program are controlled by the MPG pulse input.

- ✓ When **[MPG simulation]** is On, apart from the motion blocks, all programs are executed in a regular way.
- ✓ When the motion block is executed during program execution, if the system sets **[MPG simulation]** to On, the system immediately stops the axis and the axis stays at the current coordinates.
- ✓ When **[MPG simulation]** is On and the system has stopped the axis movement, if you set **[MPG simulation]** to Off, the system immediately executes the following motion blocks.

[M00 Program stop], [M02 End of program], and [M30 End of program and return]: M2253, M2252, and M2254

When the system is executing the program in AUTO or MDI mode and it executes M00, M02, or M30, the system sets the corresponding special M relays to On.

✓ When **the above special M relays** are set to On, they will be set to Off in the following conditions.

a. When the Reset signal is triggered.

b. When you set **[Cycle start]** to On again.

c. When you change the system to HOME mode.

[Cutting feed rate override]: D1056

When the system is executing the program in AUTO or MDI mode, you can refer to the F feed rate set in the program and set **[Cutting feed rate override]** to adjust the feed rate with the override setting.

✓ The allowable input range for **[Cutting feed rate override]** is 0 - 65535.

✓ When the feed rate exceeds the value of Pr318 (Maximum cutting feed rate), the actual feed rate is limited within the value of Pr318.

✓ When the value of **[Cutting feed rate override]** is changed, the feed rate is also changed.

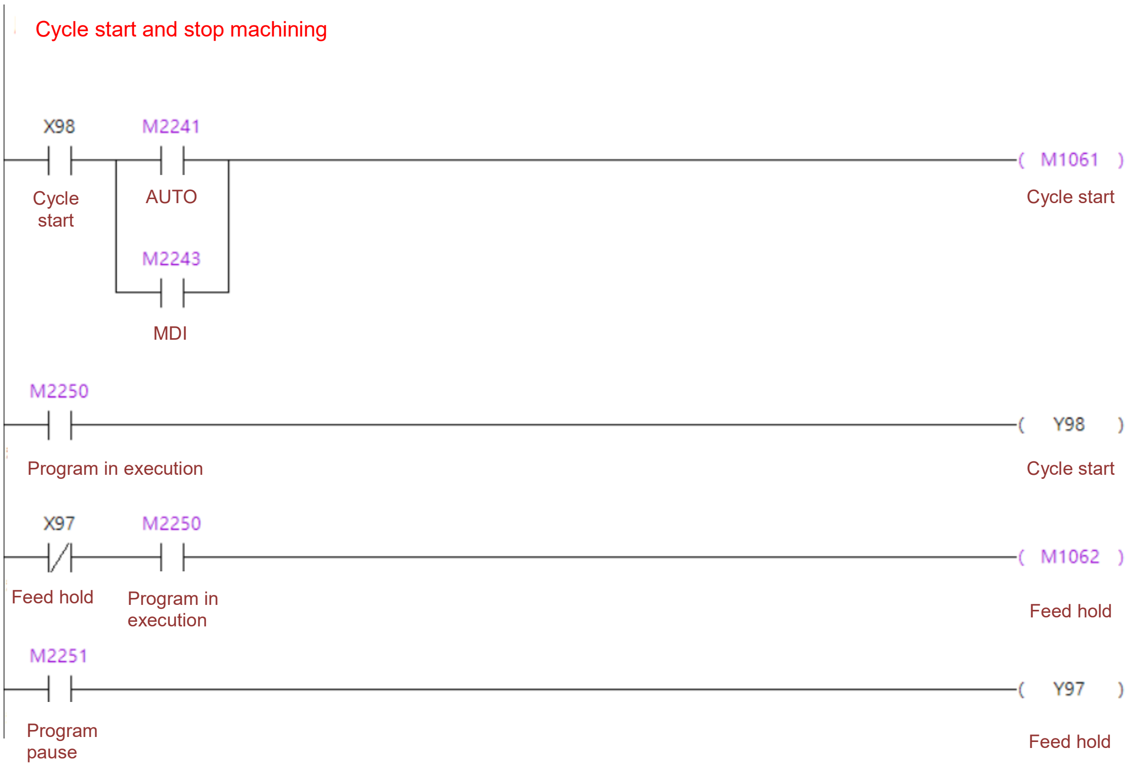
■ **Relevant parameter:**

Cutting speed setting:

When the system is executing the program in AUTO or MDI mode, the feed rate will be the F feed rate specified in the program times the ratio of **[Cutting feed rate override]**, but the actual feed rate is limited within the range of Pr318 (Maximum cutting feed rate).

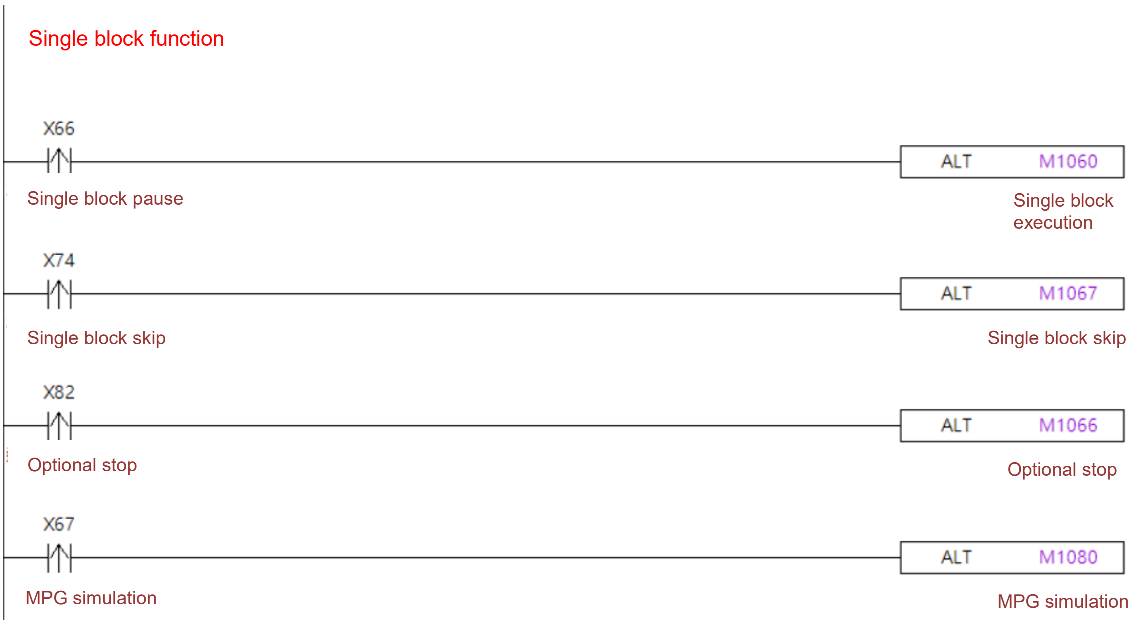
- **MLC example**

The machining related ladder diagram illustrated with the MLC example is as follows.

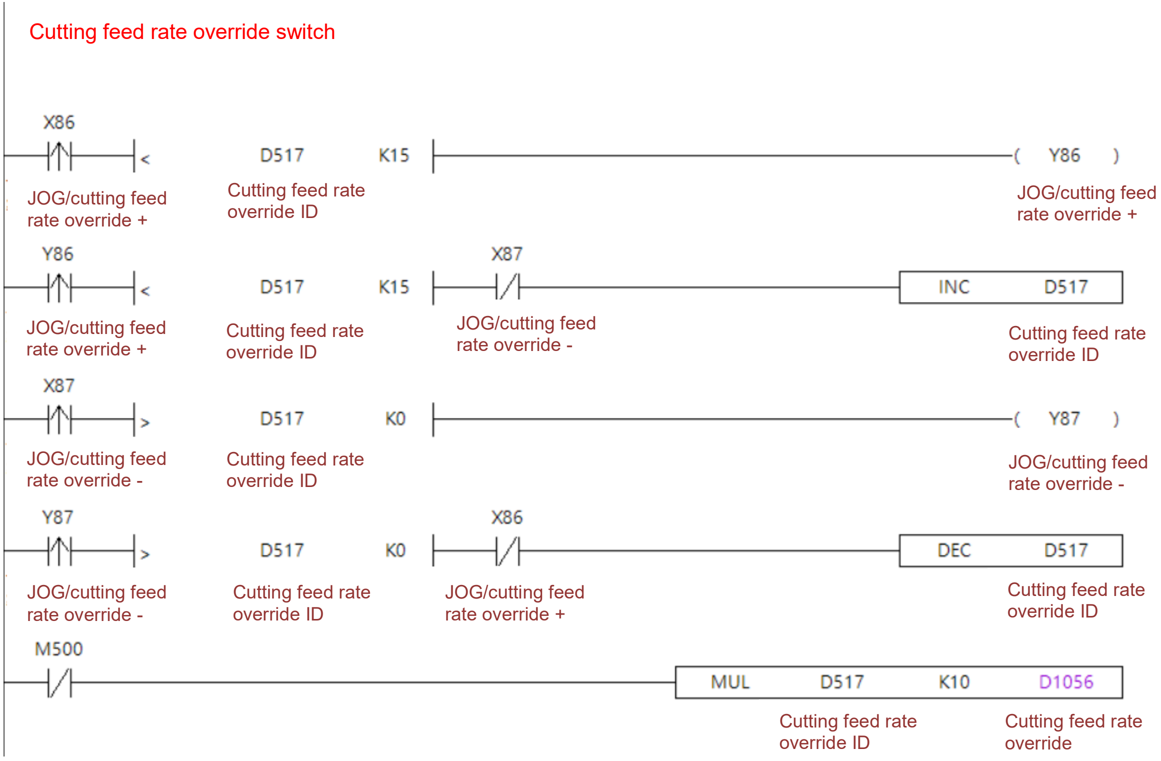


Program execution procedure

1. Use the key signal to activate the machining operation and use the normally open (A) contact to limit the mode to AUTO or MDI.
2. The system sets M2250 (Program in execution) to On, and then sets the cycle start DO to On.
3. The system can trigger M1062 (Program stop) only when the program is being executed.
4. When the program pauses, the program stop DO is set to On.



1. Use the rising edge of the key signal to trigger ALT and to switch the special M for enabling/disabling the single block pause function.
2. Use the rising edge of the key signal to trigger ALT and to switch the special M for enabling/disabling the single block skip function.
3. Use the rising edge of the key signal to trigger ALT and to switch the special M for enabling/disabling the optional stop function.
4. Use the rising edge of the key signal to trigger ALT and to switch the special M for enabling/disabling the MPG simulation.


Cutting feed rate override switch
X86
< D517 K15
Y86
JOG/cutting feed rate override +
Cutting feed rate override ID
JOG/cutting feed rate override +
Y86
< D517 K15
X87
INC D517
JOG/cutting feed rate override +
Cutting feed rate override ID
JOG/cutting feed rate override -
Cutting feed rate override ID
X87
> D517 K0
Y87
JOG/cutting feed rate override -
Cutting feed rate override ID
JOG/cutting feed rate override -
Y87
> D517 K0
X86
DEC D517
JOG/cutting feed rate override -
Cutting feed rate override ID
JOG/cutting feed rate override +
Cutting feed rate override ID
M500
MUL D517 K10 D1056
Cutting feed rate override ID
Cutting feed rate override

1. Use the key signal to set the cutting feed rate override with the maximum as 15 and the minimum as 0, and write this feed rate override ratio to the register.
2. Multiply the feed rate override ratio by 10 and save the result to the special D register.

6.3 MPG usage

On the controller, the axis and feed rate for MPG operation is switched with special M relays and special D registers.

- **MLC special M relays and special D registers**

Panel MPG pulse+	M1118	MPG operating mode ID	D1040
Panel MPG pulse-	M1119	MPG operating channel settings	D1041
MPG forward operation	M2232	MPG pulse magnification status	D1042
MPG reverse operation	M2233	MPG axis selection status	D1043

[Panel MPG pulse+] and [Panel MPG pulse-]: M1118 and M1119

When the system is in MPG mode and **[MPG operating mode ID]** is 10, you can use the key to set **[Panel MPG pulse + / -]** to On. Each time the special M is set to On, the controller outputs a MPG signal and refers to **[MPG pulse magnification status]** and **[MPG axis selection status]** to move the axis by the distance of one MPG pulse signal.

[MPG forward operation] and [MPG reverse operation]: M2232 and M2233

When you input pulses to the system using the physical MPG, the system sets **[MPG forward operation / reverse operation]** to On based on the pulses it receives. If there is no pulse input, this special M relay is set to Off.

[MPG operating mode ID]: D1040

To use the physical MPG to input pulses, set the mode ID to 0. To use the panel MPG to input pulses, set the mode ID to 10.

[MPG operating channel settings]: D1041

This special M relay is used when there are multiple channels in the system. The default is currently 0. (The multi-channel function is currently not available.)

[MPG pulse magnification status]: D1042

When you rotate the MPG by one scale, the system receives one pulse in the unit of 0.001. And the axis moves according to **[MPG pulse magnification status]**, which magnification ratio is x1, x10, or x100.

[MPG axis selection status]: D1043

You can use the MPG to switch the axis to be moved. 0 = X axis; 1 = Y axis; 2 = Z axis..., and so on to W axis.

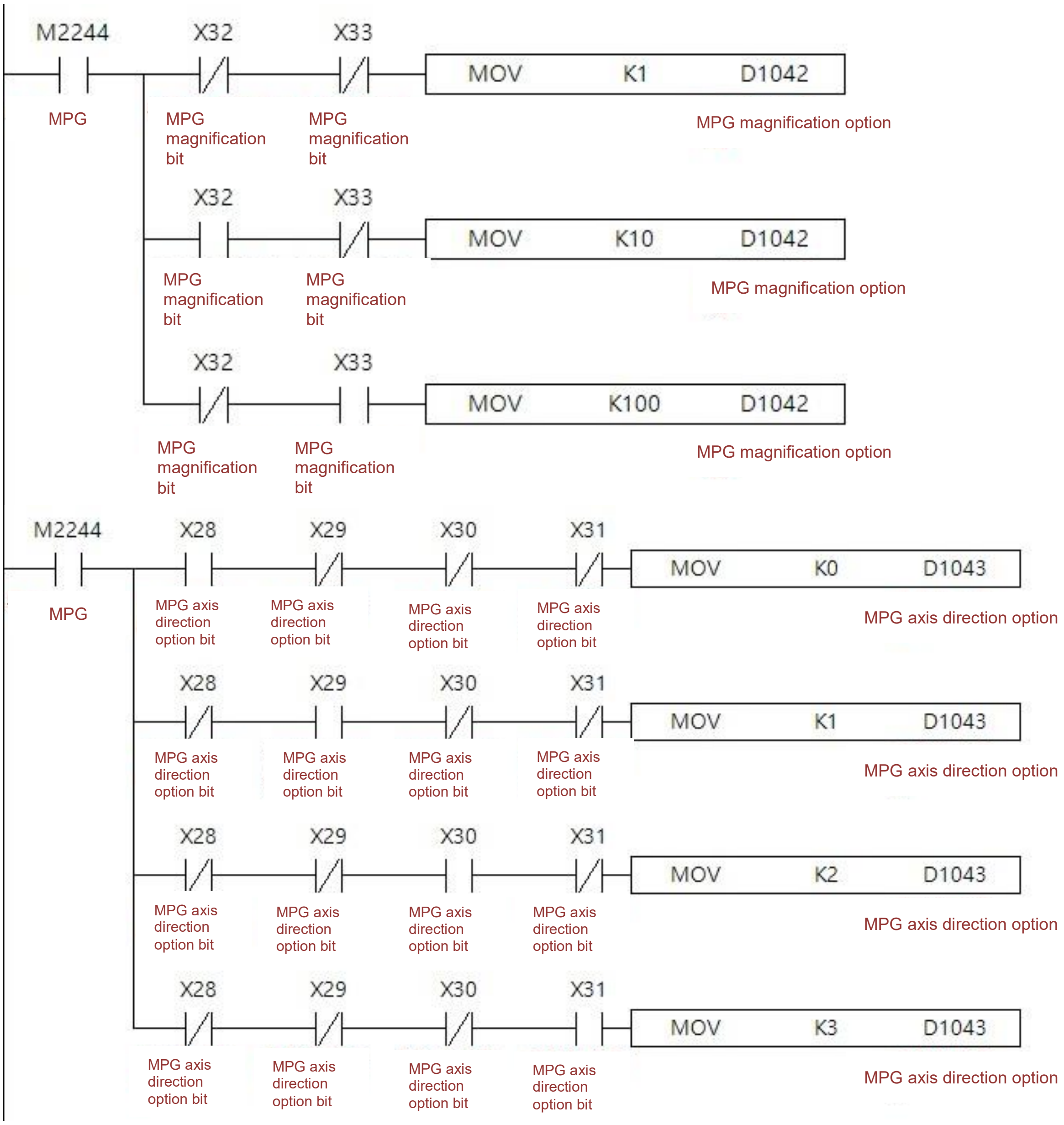
- **Relevant parameters:**

Pr47: you can use Pr47 (MPG gain) to adjust the stability of the MPG pulses. If you always need to switch more than one scale to have the controller follow the order, try increasing the value of Pr47.

Pr48: when you rotate the MPG by one scale and the controller does not react immediately every time, or sometimes the controller receives the MPG pulses but the MPG actually remains still, you can set Pr48 (MPG filter) to prevent noise interference.

■ MLC example

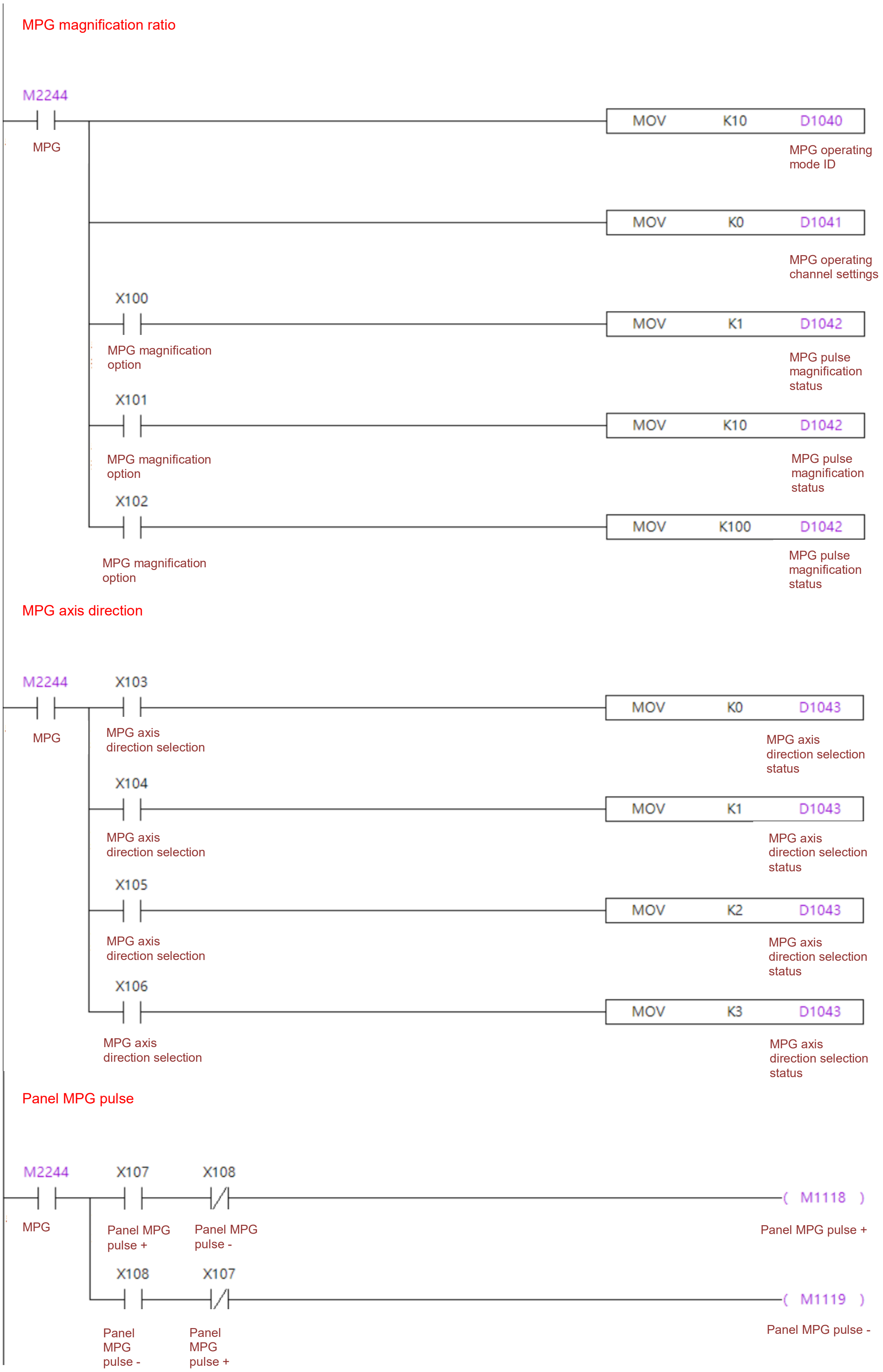
The example for 4-axis MPG is as follows.



Program execution procedure

1. In MPG mode, use the MPG signal to choose the pulse magnification ratio to use.
2. Use the MPG signal to select the axis to move.
3. Operate the MPG to move the axis.

The X-axis panel MPG pulse is used for an example as follows.



1. When the system is in MPG mode, switch the system from MPG mode to panel MPG pulse mode.
2. When the system is in MPG mode, set the MPG channel by its default.
3. When the system is in MPG mode, select the magnification ratio to use.
4. When the system is in MPG mode, select the axis to use.
5. When the system is in MPG mode, set **[Panel MPG pulse]** for the moving direction to On.

6.4 JOG

On the controller, the speed and movement for jog operation is determined by the MLC. The jog application is as follows.

- **MLC special M relays and special D register**

Axis	Jog in positive direction	Jog in negative direction	Lock machine axis movement	Lock axis movement	Axis in motion	Moving in positive direction	Moving in negative direction	Jog feed rate override
X axis	M1216	M1226	M1068	M1257	M2320	M2336	M2345	D1062
Y axis	M1217	M1227		M1258	M2321	M2337	M2346	
Z axis	M1218	M1228	M1068 M1069	M1259	M2322	M2338	M2347	
A axis	M1219	M1229	M1068	M1260	M2323	M2339	M2348	
B axis	M1220	M1230		M1261	M2324	M2340	M2349	
C axis	M1221	M1231		M1262	M2325	M2341	M2350	
U axis	M1222	M1232		M1263	M2326	M2342	M2351	
V axis	M1223	M1233		M1264	M2327	M2343	M2352	
W axis	M1224	M1234		M1265	M2328	M2344	M2353	

[Jog in positive direction]: M1216 - M1224

When the system is in JOG mode, you can set **[Jog in positive direction]** to On to have the axis move in the positive direction or set it to Off to stop the axis movement.

[Jog in negative direction]: M1226 - M1234

When the system is in JOG mode, you can set **[Jog in negative direction]** to On to have the axis move in the negative direction or set it to Off to stop the axis movement.

[Lock machine axis movement]: M1068 and M1069

When **M1068 [Lock machine axis movement]** is On, **ALL** axis commands are not sent to the servo; the controller takes them as absolute coordinate commands. When M1068 is Off, **ALL** axis commands are sent to the servo and the motor operates accordingly.

- ✓ Except the Z axis which has its own function of locking the machine axis movement, **all the rest axes are controlled with M1068.**
- ✓ When **[Lock machine axis movement]** is set to On, if it is set to Off after the axis moves, the absolute coordinates do not change to the position where this special M was On. After you change the mode or cycle the power, the system updates the absolute coordinates based on the machine coordinates and coordinate system.

[Lock axis movement]: M1257 - M1265

When **[Lock axis movement]** is set to On, **ALL** axes remain still. When it is set to Off, **ALL** axes can move normally.

[Axis in motion]: M2320 - M2328

When the axis is moving, regardless of its direction, the system sets **[Axis in motion]** to On. When the axis stops moving, the system sets **[Axis in motion]** to Off.

[Moving in positive direction]: M2336 - M2344

When the axis is moving in the positive direction, the system sets **[Moving in positive direction]** to On. When the positive movement stops, the system sets **[Moving in positive direction]** to Off.

[Moving in negative direction]: M2345 - M2353

When the axis moves in the negative direction, the system sets **[Moving in negative direction]** to On. When the negative movement stops, the system sets **[Moving in negative direction]** to Off.

[Jog feed rate override]: D1062

When the axis is moving in JOG mode and you set Pr305 (Jog speed mode) to 0, the system refers to the value of **[Jog feed rate override]** as the moving speed. The linear axis moves according to the F setting value (unit: mm/min). The rotation axis refers to the value of **[Jog feed rate override]** and refers to Pr634 (Rotation axis unit selection) to determine whether the unit is rpm or deg/min.

✓ When Pr305 (Jog speed mode) is set to 1, the value of **[Jog feedrate override]** is in the unit of % and the maximum is 100%.

■ **Relevant parameter settings**

Speed parameter settings:

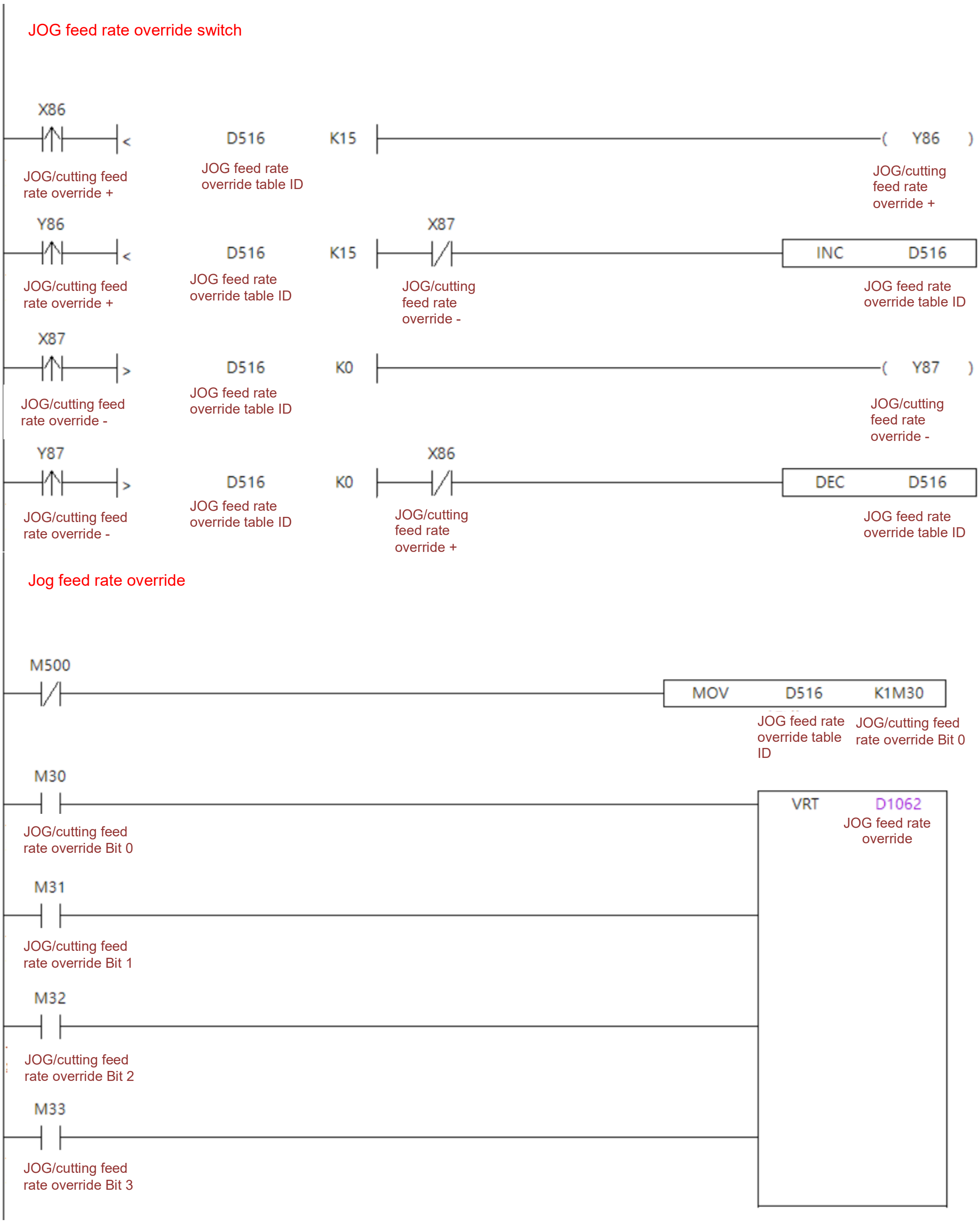
The jog speed is determined by the following parameters: Pr621 (Manual rapid traverse and maximum speed), Pr622 (Acceleration/deceleration time), and Pr623 (S-curve time).

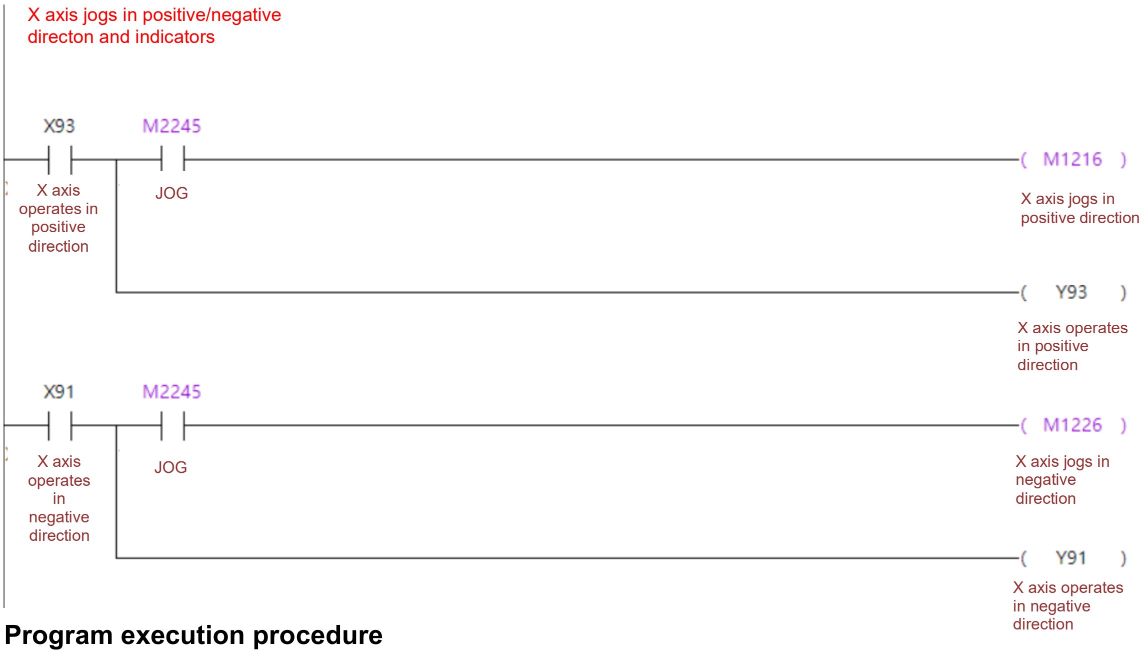
Jog speed mode setting:

Apart from using **[Jog feed rate override]** to set the Jog speed, you can use Pr305 (Jog speed mode) to set the speed in the unit of percentage. When Pr305 is set to 1, you can set Pr640 (100% Jog speed) and use it as the setting basis, and the system refers to **[Jog feed rate override]** to determine the jog speed.

■ MLC example

The X-axis jog operation is used as an example as follows. You can change the value of **[Jog feed rate override]** when the axis is moving, so it is not necessary to complete the setting of **[Jog feed rate override]** before operating the axis.


JOG feed rate override switch
X86
JOG/cutting feed rate override +
<
D516
JOG feed rate override table ID
K15
Y86
JOG/cutting feed rate override +
Y86
JOG/cutting feed rate override +
<
D516
JOG feed rate override table ID
K15
X87
JOG/cutting feed rate override -
INC D516
JOG feed rate override table ID
X87
JOG/cutting feed rate override -
>
D516
JOG feed rate override table ID
K0
Y87
JOG/cutting feed rate override -
Y87
JOG/cutting feed rate override -
>
D516
JOG feed rate override table ID
K0
X86
JOG/cutting feed rate override +
DEC D516
JOG feed rate override table ID
Jog feed rate override
M500
MOV D516 K1M30
JOG feed rate override table ID
JOG/cutting feed rate override Bit 0
M30
JOG/cutting feed rate override Bit 0
VRT D1062
JOG feed rate override
M31
JOG/cutting feed rate override Bit 1
M32
JOG/cutting feed rate override Bit 2
M33
JOG/cutting feed rate override Bit 3


X axis jogs in positive/negative directon and indicators
X93
M2245
M1216
X axis operates in positive direction
JOG
X axis jogs in positive direction
Y93
X axis operates in positive direction
X91
M2245
M1226
X axis operates in negative direction
JOG
X axis jogs in negative direction
Y91
X axis operates in negative direction

Program execution procedure

1. Use the key signal to increase or decrease the value of **[Jog feed rate override]** with the maximum as 15 and the minimum as 0, and then write the value to the register.
2. Use the MOV instruction to move the override value of the register to the digit of K1M30.
3. Use the digit of K1M30 to input the value of the VRT table to D1062.
4. Perform X-axis jog operation using the X-axis positive / negative JOG keys.

Note: adjust the DI for each axis and special M relays according to the application.

6.5 Rapid traverse

There are two types of rapid traverse for the controller, manual rapid traverse and rapid traverse commands such as G00 during program execution. Both types use the same speed and override ratio. The rapid traverse application is as follows.

- **MLC special D register**

Rapid traverse speed adjustment rate	D1058

[Rapid traverse speed adjustment rate]: D1058

When the system is in RAPID mode or executes the rapid traverse command such as G00 during program execution, the system refers to Pr316 (G00 feed rate), uses Pr621 (G00/manual maximum speed) as the 100% moving speed, and adjusts the speed based on **D1058 [Rapid traverse speed adjustment rate]**.

✓ The allowable input range for **D1058** is -32768 to +32767.

✓ When the rapid traverse speed exceeds the setting of Pr316 (G00 feed rate) or Pr621 (G00/manual maximum speed), the rapid traverse speed is still limited by Pr316 and Pr621.

✓ When the value of **D1058** changes, the rapid traverse speed changes as well.

✓ When the value of **D1058** is set to 0%, the system refers to Pr315 (G00 feed rate at 0%).

- **Relevant parameters:**

AUTO mode rapid traverse setting:

In AUTO mode, the G00 rapid traverse is influenced by Pr315 (G00 feed rate at 0%), Pr316 (G00 feed rate), Pr317 (G00 ACC/DEC time constant), Pr511 (G00 S-curve time constant), Pr621 (G00/manual maximum speed limit), Pr622 (G00/manual ACC/DEC time), and Pr623 (G00/manual S-curve time).

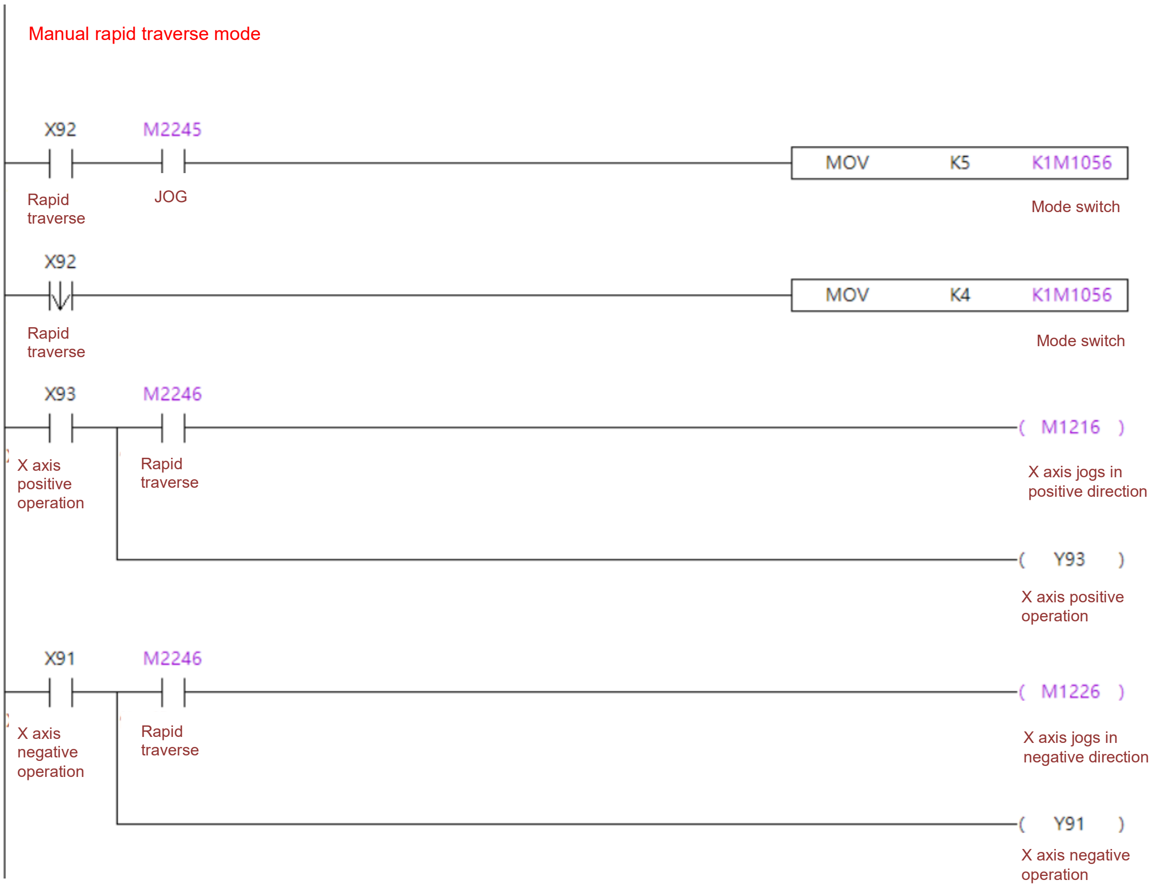
✓ Pr315 and Pr316 set the maximum speed limits and ACC/DEC speed settings of all axes for program execution. Use Pr621, Pr622, and Pr623 to individually set each axis, and the system refers to these settings to calculate the execution speed when multiple axes moves simultaneously.

Manual rapid mode setting:

In Manual rapid mode, the moving speed is influenced by Pr621 (G00/manual maximum speed limit), Pr622 (G00/manual ACC/DEC time), and Pr623 (G00/manual S-curve time).

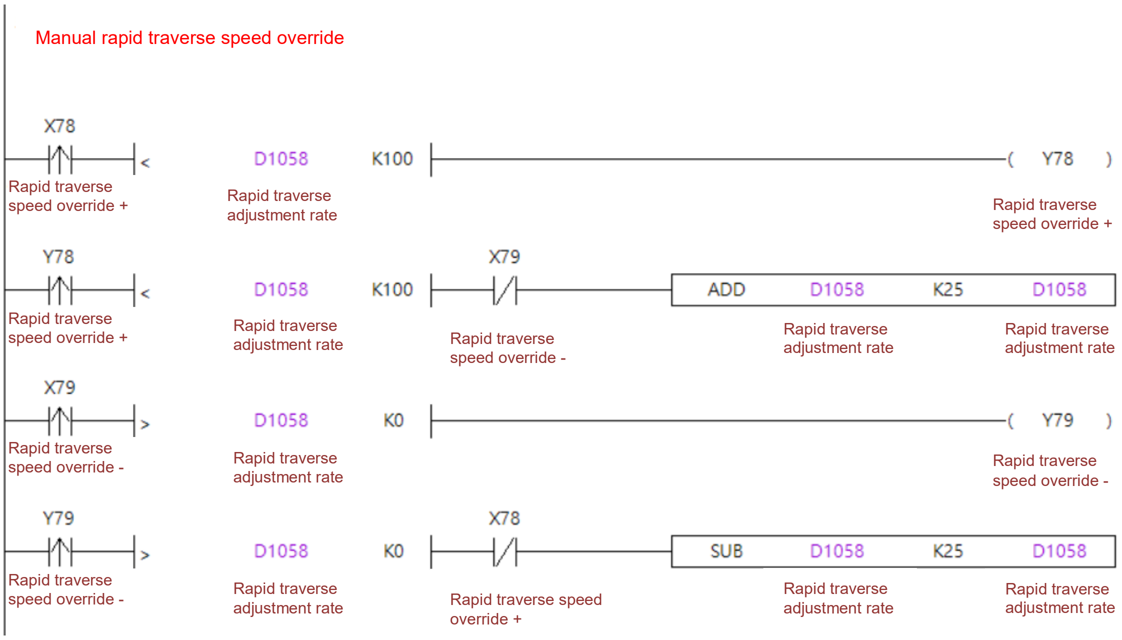
■ MLC example

The X axis is used in the following example to explain the switching of Manual rapid mode and RAPID mode speed override.


Manual rapid traverse mode
X92
M2245
Rapid traverse
JOG
MOV K5 K1M1056
Mode switch
X92
Rapid traverse
MOV K4 K1M1056
Mode switch
X93
M2246
X axis positive operation
Rapid traverse
M1216
X axis jogs in positive direction
Y93
X axis positive operation
X91
M2246
X axis negative operation
Rapid traverse
M1226
X axis jogs in negative direction
Y91
X axis negative operation

Program execution procedure

1. Set the system to JOG mode.
2. Hold the RAPID mode button to switch to RAPID mode.
   - ✓ To general users, the Manual rapid mode is not a frequently used mode. It is usually used when the system is in JOG mode but you shortly need to move the axis with the speed higher than the maximum jog speed. Therefore, this mode takes effect under multiple conditions; when you release the RAPID mode button, the system returns to JOG mode.
3. When the system is in RAPID mode and you press the X axis direction key, the system moves the axis based on the setting of Pr621 (G00/manual maximum speed limit).


Manual rapid traverse speed override
X78
Rapid traverse speed override +
<
D1058
Rapid traverse adjustment rate
K100
Y78
Rapid traverse speed override +
Y78
Rapid traverse speed override +
<
D1058
Rapid traverse adjustment rate
K100
X79
Rapid traverse speed override -
ADD
D1058
Rapid traverse adjustment rate
K25
D1058
Rapid traverse adjustment rate
X79
Rapid traverse speed override -
>
D1058
Rapid traverse adjustment rate
K0
Y79
Rapid traverse speed override -
Y79
Rapid traverse speed override -
>
D1058
Rapid traverse adjustment rate
K0
X78
Rapid traverse speed override +
SUB
D1058
Rapid traverse adjustment rate
K25
D1058
Rapid traverse adjustment rate

Program execution procedure

Use the key signal to increase / decrease the override rate with the maximum as 100 and the minimum as 0. Each trigger increases / decreases the rate by 25 and writes the override rate to **D1058 [Rapid traverse speed adjustment rate]**.

6.6 Homing

To perform the homing procedure in the system, you need to trigger the procedure in HOME mode as well as trigger the special M relay to notify the system about the axis for homing. More detailed introduction to the homing function is as follows.

- **MLC special M relays**

Axis	Homing control	Homing complete
X axis	M1236	M2272
Y axis	M1237	M2273
Z axis	M1238	M2274
A axis	M1239	M2275
B axis	M1240	M2276
C axis	M1241	M2277
U axis	M1242	M2278
V axis	M1243	M2279
W axis	M1244	M2280

[Homing control]: M1236 - M1244

When the system is in HOME mode, you can set **[Homing control]** to On to have the corresponding axis return to the origin.

✓ These special M relays are valid only in HOME mode.

[Homing complete]: M2272 - 2280

When **[Homing control]** is set to On and the system completes the homing procedure, the system sets **[Homing complete]** to On.

✓ When **[Homing complete]** is set to On and the axis is in the following conditions, **[Homing complete]** will be set to Off.

a. The axis is moving in JOG or MPG mode.

b. Use AUTO or MDI mode to perform machining.

c. When a non-absolute type motor is used, Pr616 (Origin search mode) is not set to 5, and the controller is power cycled.

d. When the absolute motor is used, Pr616 (Origin search mode) is set to 5, and the motor loses its origin.

■ **Relevant parameters:**

Origin search mode setting:

When homing, the system refers to Pr616 (Origin search mode) and Pr617 (Origin search setting) to determine the behavior of searching for the hard stop and motor's Z phase.

Coordinate setting for homing:

When the system performs homing and finds the hard stop and motor's Z phase by referring to the settings of Pr616 and Pr617, the system can refer to Pr606 (Machine origin coordinate) and regards the Z phase plus the offset distance as the machine origin.

Setting of ignoring the Z phase distance:

When the motor leaves the hard stop and Z phase, if the grid distance is very close to a screw pitch distance and you cannot adjust the distance between the hard stop and Z phase, set Pr617 [Bit 5] (Origin search - ignore Z phase distance) to 1, and input the value (grid volume / screw pitch*100%) to Pr645 (Ignore Z phase distance). When the motor leaves the hard stop and the grid value is within the range of Pr645, the system looks for the next Z phase in the same direction and regards it as the origin.

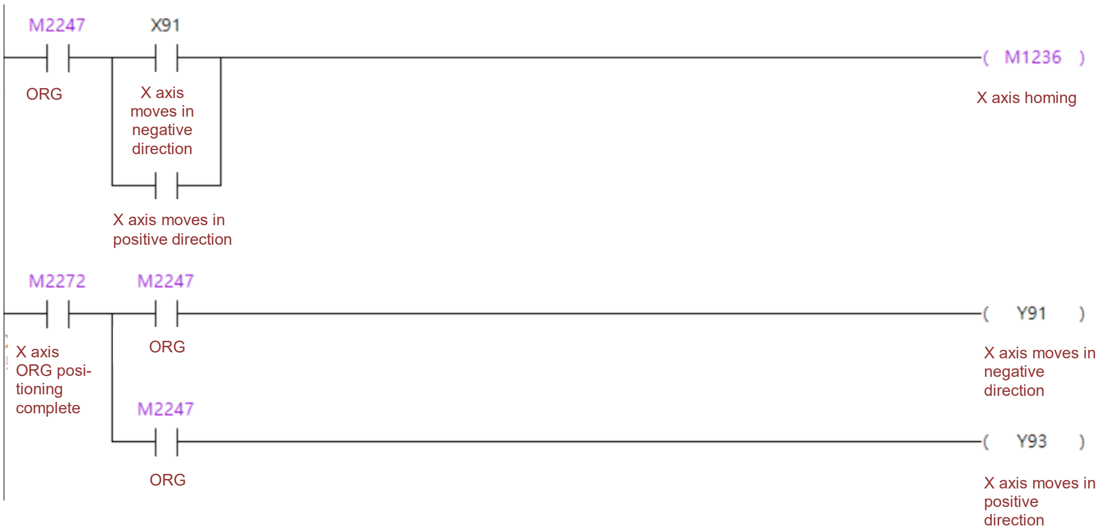
✓ When you can adjust the hard stop position, adjust the hard stop first instead of setting this parameter.

Homing speed parameter setting:

When homing, the homing speed is influenced by the following parameters: Pr618 (1st homing speed), Pr619 (2nd homing speed), Pr622 (ACC/DEC time), and Pr623 (S-curve time).

■ **MLC example**

To perform the homing procedure, the system has to be in HOME mode; in addition, you need to use the keyto set the special M to On. The X axis is used in the following example for explanation.


M2247
X91
(M1236)
ORG
X axis moves in negative direction
X axis homing
X axis moves in positive direction
M2272
M2247
(Y91)
X axis ORG posi-tioning complete
ORG
X axis moves in negative direction
M2247
(Y93)
ORG
X axis moves in positive direction

Program execution procedure

1. Switch the system to HOME mode.
2. Use the key to set the special M relay for X axis homing to On.
3. Output the homing complete signal to the button indicator.

Note: adjust the DIs and special M relays for each axis according to the applications.

6.7 M, S, and T codes execution

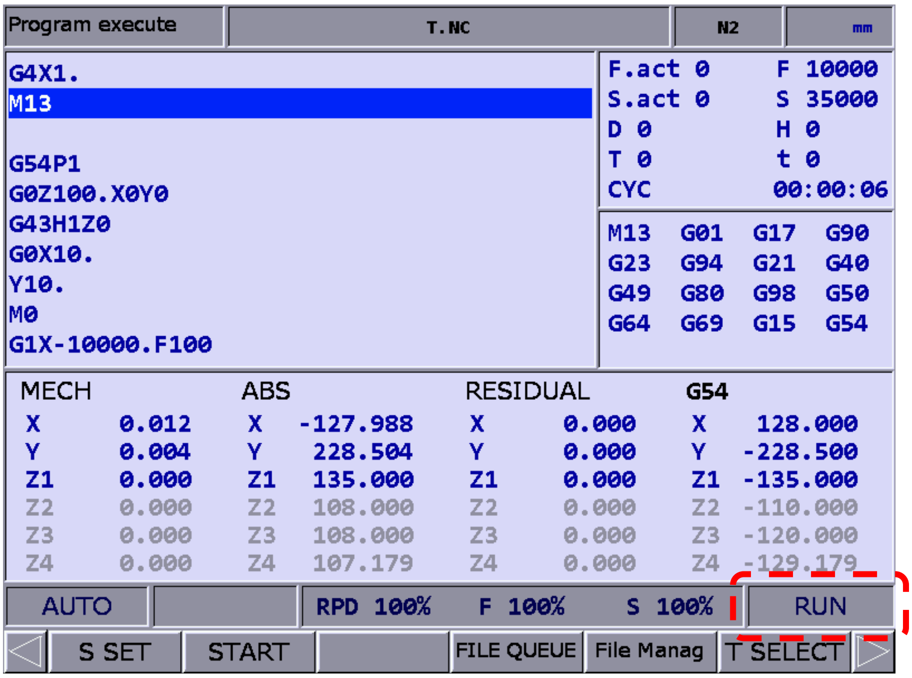
Most of the time, each M, S, and T code on the controller is relevant to the machine action; therefore, when the controller executes M, S, or T code, it triggers the M, S, or T code execution. This special M relay must be confirmed and cleared by the MLC. The functions of M, S, and T codes are as follows.

■ MLC special M relays and special D registers

Lock M, S, and T codes	M1071
M, S, and T codes execution complete	M1152
M code execution	M2208
S code execution	M2209
T code execution	M2210
M code data	D1368
S code data	D1369
T code data (instruction)	D1370

[Lock M, S, and T codes]: M1071

When the system is executing the program and M, S, or T code is executed, the system stops executing the program that follows and the state shows “RUN” (as shown in Figure 6.7.1). When **[Lock M, S, and T codes]** is On, the controller skips the M, S, and T codes, and then continues to execute the program which follows. In addition, the execution flags for M, S, and T codes will not be set to On.



```
Program execute            T.NC                 N2      mm

G4X1.                                 F.act 0      F 10000
M13                                   S.act 0      S 35000
                                      D 0          H 0
G54P1                                 T 0          t 0
G0Z100.X0Y0                           CYC          00:00:06
G43H1Z0                               M13  G01  G17  G90
G0X10.                                G23  G94  G21  G40
Y10.                                  G49  G80  G98  G50
M0                                    G64  G69  G15  G54
G1X-10000.F100

MECH              ABS                RESIDUAL          G54
X     0.012       X   -127.988       X     0.000       X    128.000
Y     0.004       Y    228.504       Y     0.000       Y   -228.500
Z1    0.000       Z1   135.000       Z1    0.000       Z1  -135.000
Z2    0.000       Z2   108.000       Z2    0.000       Z2  -110.000
Z3    0.000       Z3   108.000       Z3    0.000       Z3  -120.000
Z4    0.000       Z4   107.179       Z4    0.000       Z4  -129.179

AUTO         RPD 100%    F 100%    S 100%    RUN
```

Figure 6.7.1 M, S, and T codes in execution

[M, S, and T codes execution complete]: M1152

When the system is executing the program and the M, S, or T code is executed, the system stops executing the program which follows, sets the state to “RUN” (as shown in Figure 6.7.1), and sets the [M, S, and T codes execution] flags to On. After the MLC program planned by the users is complete, you need to set **M1152 [M, S, and T codes execution complete]** to On and then the controller can set the [M, S, and T codes execution] flags to Off to continue executing the program which follows.

[M code execution]: M2208

When the system is executing the program and an M code is executed, the system stops executing the program which follows, sets the state to “RUN” (as shown in Figure 6.7.1), and sets **M2208 [M code execution]** to On. After the MLC program planned by the users is complete, you need to set **M1152 [M, S, and T codes execution complete]** to On and then the controller can set **M2208 [M code execution]** to Off to continue executing the program which follows.

[S code execution]: M2209

When the system is executing the program and an S code is executed, the system stops executing the program which follows, sets the state to “RUN” (as shown in Figure 6.7.1), and sets **M2209 [S code execution]** to On. After the MLC program planned by the users is complete, you need to set **M1152 [M, S, and T codes execution complete]** to On and then the controller can set **M2209 [S code execution]** to Off to continue executing the program which follows.

[T code execution]: M2210

When the system is executing the program and a T code executed, the system stops executing the program which follows, sets the state to “RUN” (as shown in Figure 6.7.1), and sets **M2210 [T code execution]** to On. After the MLC program planned by the users is complete, you need to set **M1152 [M, S, and T codes execution complete]** to On and then the controller can set **M2210 [T code execution]** to Off to continue executing the program which follows.

[M code data]: D1368

When the system is executing the program and an M code is executed, the system stops executing the program which follows, sets the state to “RUN” (as shown in Figure 6.7.1), and writes the M code data which is executed to **D1368 [M code data]**.

- ✓ For example, when the system executes M13, **[M code data]** displays 13.

[S code data]: D1369

When the system is executing the program and an S code is executed, the system stops executing the program which follows, sets the state to “RUN” (as shown in Figure 6.7.1), and writes the S code data which is executed to **D1369 [S code data]**.

- ✓ For example, when the system executes S4000, **[S code data]** displays 4000.
- ✓ When there are multiple T codes in the block to be read, only the last T code is written.

[T code data (instruction)]: D1370

When the system is executing the program and a T code is executed, the system stops executing the program which follows, sets the state to "RUN" (as shown in Figure 6.7.1), and writes the T code data which is executed to **D1370 [T code data]**.

✓ For example, when the system executes T5, **D1370 [T code data]** displays 5.

✓ The T code in the program has to be within the range set in Pr338 and Pr341, so the T code can be displayed.

✓ When there is a T code executed in the program, if the tool magazine 1 and tool magazine 2 are not set to 1 in Pr337, the system sends an alarm and does not write the T code data to **D1370 [T code data]**.

■ **Relevant parameters:**

M code macro call program setting:

Pr13 - 22 (M code macro call O_____).

When the M code is set for an M code macro call and executed by the controller, the controller neither sets **M2208 [M code execution]** to On nor writes the M code data to **D1368 [M code data]**. Instead, the controller refers to the parameter setting and executes the corresponding macro.

✓ When the program in execution is already a macro called by the M code, if this M code is executed, the corresponding macro will not be executed. In addition, the controller sets **M2208 [M code execution]** to On and writes the M code data to **D1368 [M code data]**.

✓ When the parameter is set to 0, M code macro call is disabled.

T code macro call O9000 setting:

Set Pr23 to enable the T code macro call O9000.

When the T code macro call program is set to 1, if the controller executes this T code, it neither sets **M2210 [T code execution]** to On nor writes the T code data to **D1370 [T code data]**; instead, it calls O9000 for execution.

✓ When the program in execution is already the O9000 program, the controller sets **M2210 [T code execution]** to On and writes the executed T code value to **D1370 [T code data].**

✓ When Pr23 is set to 0, T code macro call O9000 is disabled.

Execution of macro program before T code:

- ✓ Set Pr305 for executing macro program before T code.
- ✓ Set Pr506 for executing macro program before T code.
- ✓ Set Pr507 for the line number for executing macro program before T code.

When Pr305 Execute macro program before T code is set to 1, the controller refers to the settings of Pr506 and Pr507 to call the macro set in Pr506 in the set line before T code. Calling the macro will not set **M2210 [T code execution]** to On and the T code data is not written to **D1370 [T code data]**.

- ✓ In the macro program which is executed before T code, if you re-execute the T code thus O9000 is called, this neither sets **M2210 [T code execution]** to On nor writes the executed T code value to **D1370 [T code data]**.
- ✓ When executing the O9000 called by the macro program executed before T code, re-executing the T code does not call O9000. Meanwhile, the system sets **M2210 [T code execution]** to On and writes the executed T code value to **D1370 [T code data]**.
- ✓ The function of executing macro program before T code is only supported by the firmware of woodworking machine.
- ✓ Set Pr23 (T-code macro call - O9000) to 1 to have the function take effect.

Synchronous execution for M / S / T code and G00

When Pr307 is set to 0, the M, S, and T codes cannot be executed with G00 in the same line of program. When Pr307 is set to 1, the M, S, and T codes can be executed with the line which contains G00. When G00 is complete but the M, S, or T code is not complete, the system stays in the state of "RUN" and waits for the M, S, and T codes to complete.

Setting of Halt M code:

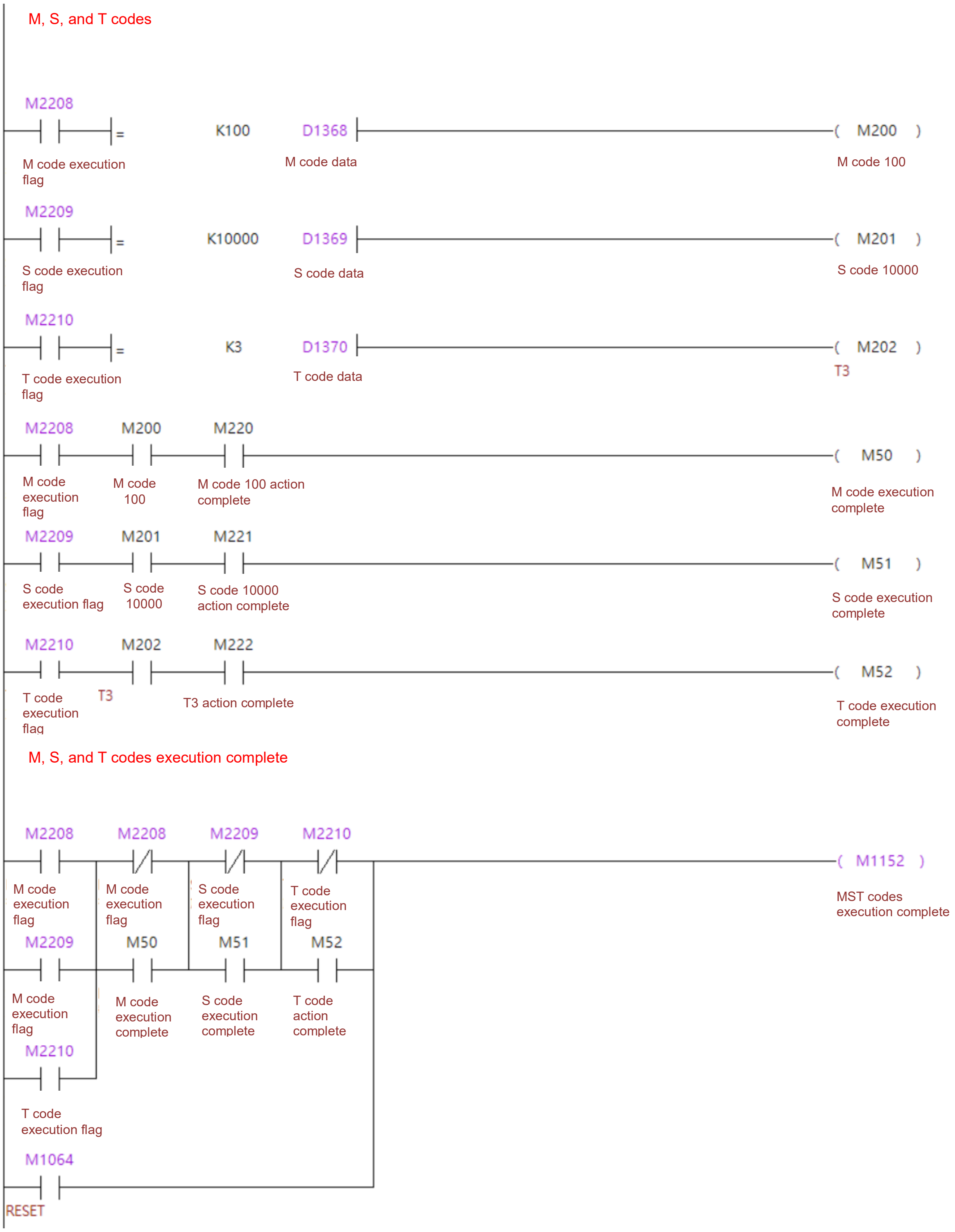
Pr350 - 359 Halt M code.

When the controller is executing the program, the system automatically plans the path by looking ahead of the following program which has not been executed. When the M code is set to halt the look-ahead of M code, once the system reads the M code, it does not look ahead of the program. After the system completes executing the M code, it continues to look ahead of the program.

- ✓ When the controller is executing the program and reads the Halt M code:
  - a. The system continues to execute the program which follows and sets the state as "RUN" (as shown in Figure 6.7.1).
  - b. The system sets **[M code execution]** to On.
  - c. The system writes the executed M code value to **[M code data]**.
  - d. You can plan the MLC actions for the M code normally.
  - e. **[M, S, and T codes execution complete]** has to be set to On for executing this M code.

MLC example

The following example illustrates the execution procedure of M, S, and T codes.



Program execution procedure

M code execution procedure

1. When the system is executing the program and executes M100, it sets **[M code execution]** to On.
2. At the time **[M code execution]** is set to On, the system writes 100 to **[M code data]**.
3. When the conditions are met, M200 in the MLC is set to On, so you can use M200 to trigger the required MLC actions.
4. Once you finish the planned MLC actions, set M220 of the MLC to On.
5. When you set M220 to On, M50 is set to On because the conditions are met.
6. When M50 is On, the **[M, S, and T codes execution complete]** flag is set to On because the conditions are met.
7. When the system confirms that the **[M, S, and T code execution complete]** flag is On, it regards that the M code execution is complete and sets **[M code execution]** to Off.
8. M code actions complete.

S code execution procedure

1. When the system is executing the program and executes S10000, it sets **[S code execution]** to On.
2. At the time **[S code execution]** is set to On, the system writes 10000 to **[S code data]**.
3. When the conditions are met, M201 in the MLC is set to On, so you can use M201 to trigger the required MLC action.
4. Once you finish the planned MLC actions, set M221 of the MLC to On.
5. When M221 is set to On, M51 is set to On because the conditions are met.
6. When M51 is set to On, the **[M, S, and T codes execution complete]** flag is set to On because the conditions are met.
7. When the system confirms that the **[M, S, and T codes execution complete]** flag is On, it regards that the S code execution is complete and sets **[S code execution]** to Off.
8. S code actions complete.

T code execution procedure

1. When the system is executing the program and executes T3, it sets **[T code execution]** to On.
2. At the time **[T code execution]** is set to On, the system writes 3 to **[T code data]**.
3. When the conditions are met, M202 in the MLC is set to On, so you can use M202 to trigger the required MLC actions.
4. Once you finish the planned MLC actions, set M222 of the MLC to On.
5. When M222 is set to On, M52 is set to On because the conditions are met.
6. When M52 is set to On, the **[M, S, and T codes execution complete]** flag is set to On because the conditions are met.
7. When the system confirms that the **[M, S, and T codes execution complete]** flag is On, it regards that the T code execution is complete and sets **[T code execution]** to Off.
8. T code actions complete.

Important: this section is illustrated only based on T code. For more details about the tool magazine and T code functions, refer to Section 6.14.

6.8 M96 subroutine interruption

If you need to use an external signal to have the machine perform specific actions from time to time, such as vacuuming and spraying oil on workpieces, use M96 subroutine interruption. The function description is as follows.

■ MLC special M relays

M96 subroutine interruption	M1078
M96 subroutine interruption in operation	M2216

[M96 subroutine interruption]: M1078

When the controller executes M96 during program execution, it uses the external signal to set **M1078 [M96 subroutine interruption]** to On, so the system immediately stops the current program execution and calls the macro.

[M96 subroutine interruption in operation]: M2216

When the controller executes M96 during program execution, sets M1078 [M96 subroutine interruption] to On, and executes the macro call, the system will set **M2216 [M96 subroutine interruption in operation]** to On. Once the execution for M99 is complete in the macro and the system returns to the main program, the system sets **M2216** to Off.

■ Relevant parameter:

Execution mode setting:

When the system finishes the macro call and returns to the main program, it refers to Pr51 [Bit 10] (Return mode after triggering M96) to determine if it is to carry on executing the next block or the block which was interrupted.

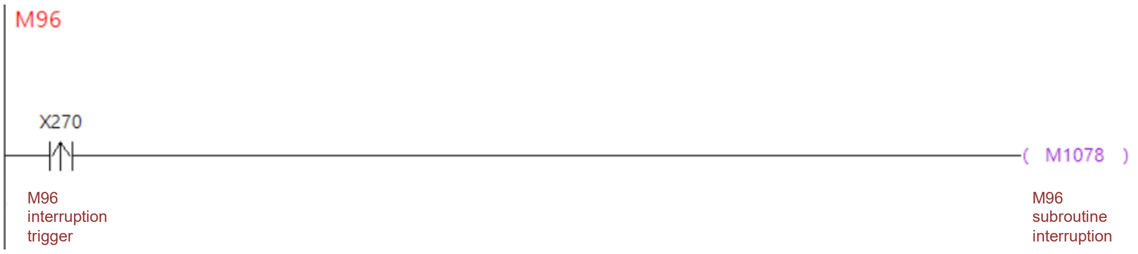
✓ When execution for the macro call program is complete, you can use M99P__ to return to the specified line.

■ MLC example

The following describes the M96 subprogram interruption function using the main program, M96 macro call, and MLC ladder diagram as example.

Main program	M96 macro call
G90G54G0X100.Y100. G0Z-20. M96P9090 G1X50.F1500 G1Y50. G1X0. G1Y0. G1X100. G1Y100. M97 M30	G0Z50. M50 (worktable vacuuming) G0Z-15. G91G1Z-5.F300 G90 M99

MLC:



Program execution procedure

1. When the system is executing the program and M96P9090 is executed, the system is ready to call the macro for execution anytime.
2. When the main program runs to G1Y50, X270 is triggered. When the system detects that **[M96 subroutine interruption]** is On, it automatically calls O9090 for execution.
3. When the system executes M99 of O9090, it determines the execution mode based on Pr51 [Bit 10] (Return mode after triggering M96)
   - ✓ If you set to return to the original block (the interrupted line), the system executes G1Y50.
   - ✓ If you set to go to the next line, the system executes G1X0.
4. Upon returning to the main program, the system continues to execute the following blocks until it reaches M97. Then, the M96 function is disabled and no longer controlled by **[M96 subroutine interruption]**.

Important:

1. When the program in which you execute M96 (subroutine interruption) is nested within another program layer and you return to that program layer, the system automatically disables the M96 function.

   Example:

Execution layer	M98 status (On / Off)
Layer 1	Off
Layer 2	Off
Layer 2; execute M96	On
Layer 2; trigger M96 subroutine interruption	On
Layer 3; M96 macro; execute to M99	On
Layer 2	On
Layer 2; execute to M99.	On
Layer 1	Off (automatically disabled)

2. When the macro called by M96 subroutine interruption function is being executed, do not execute M96; otherwise, the system activates the alarm B656 (M96 is sent when subroutine interruption is in operation).
3. When executing G16, G24, G41, G42, and G51, do not execute M96; otherwise, the system activates the alarm B655 (Specific function remains enabled during M96 execution).

6.9 M99 halt function for main program

When the main file is executed on the controller and M99 is executed, the main file will be repeatedly executed. After you enable this function, M99 is treated as a general M code and the system checks the MLC conditions to confirm if the external actions are complete.

- **MLC special M relays**

M99 halt function for main program	M1077
M99 halt	M2238

[M99 halt function for main program]: M1077

When M99 halt function for the main program is enabled, the system sets **M1077** to On. When M99 is executed during main program execution, the system sets the **[M, S, and T codes execution complete]** flag to On. Then, the system stops the execution for the main program and returns to the first line of the program (same as executing M30) and sets the controller status to “Ready”.

[M99 halt]: M2238

When M99 halt function for the main program is enabled and M99 is executed during main program execution, the system sets **M2238 [M99 halt]** to ON, and then sets it to Off until the **[M, S, and T codes execution complete]** flag goes On.

✓ If M99 is executed during main program execution and you trigger the Reset signal when **M2238** is On but the **[M, S, and T codes execution complete]** flag remains Off, then **M2238** will remain On. **M2238** is set to Off until the next time M99 is executed and the **[M, S, and T codes execution complete]** flag goes On.

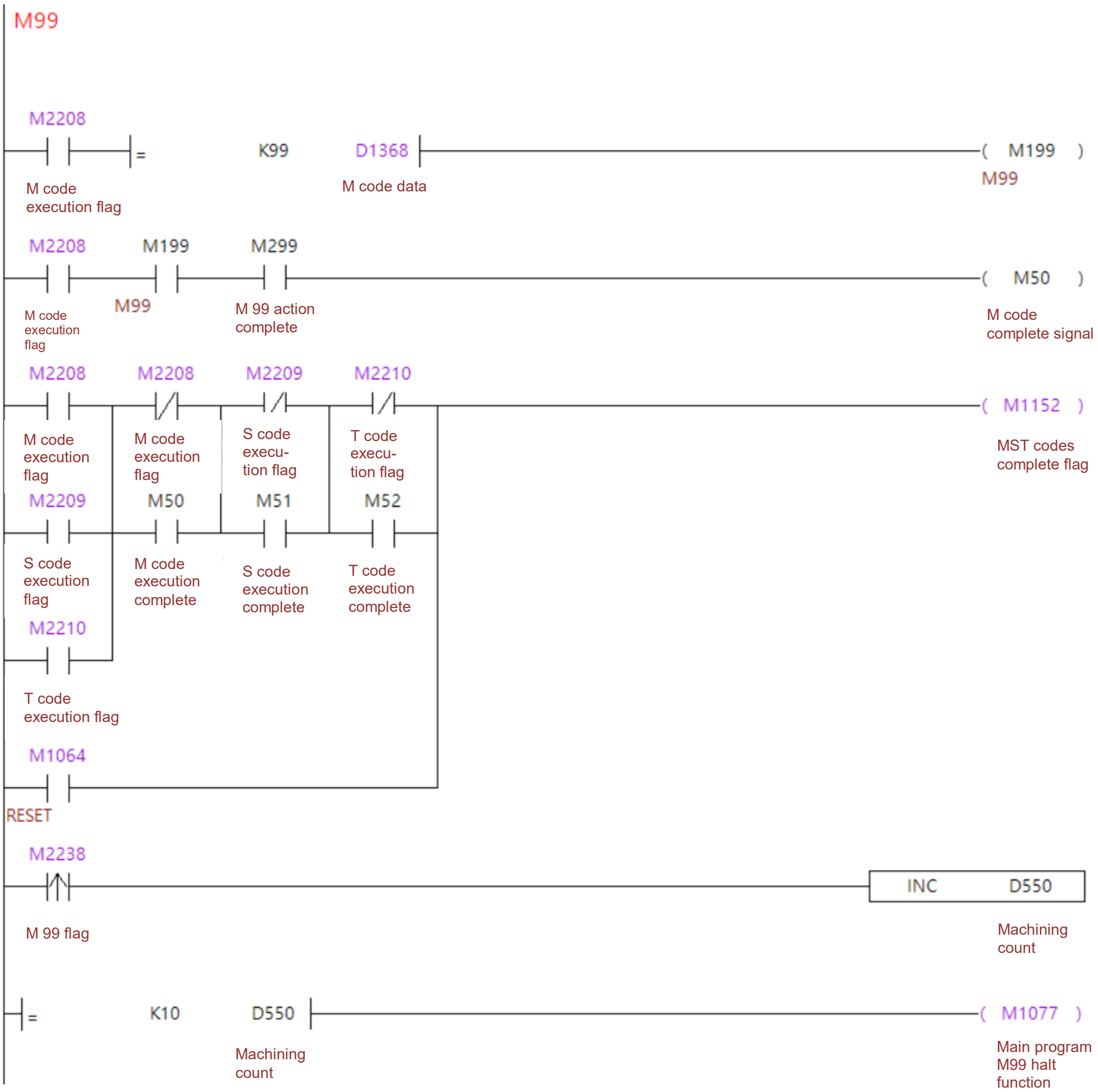
- **Relevant parameter:**

Enable M99 halt function for main program:

When Pr308 [Bit 9] (Enable M99 halt function) is set to 1, this function becomes effective when the controller executes the main file and M99 is executed.

■ MLC example

The following example uses M99 as a general M code for explanation.



Program execution procedure

1. When M99 is executed during main program execution, the system sets **M2238 [M99 halt]** to On.
2. When **M2238 [M99 halt]** is On, the count value of D550 increases by 1 and the M code execution flag is set to On.
3. When you set M299 to On, which indicates M99 action complete, the **[M, S, and T codes execution complete]** flag is set to On because the condition is met.
4. When the **[M, S, and T codes execution complete]** flag goes On, the system sets **M2238 [M99 halt]** to Off.
5. The above 1 - 4 steps are repeated 10 times.
6. When the count value of D550 reaches 10, **M1077 [M99 halt function for main program]** is On because the condition is met.
7. When M99 is executed during main program execution, the system sets **M2238 [M99 halt]** to On.

8. When **M2238 [M99 halt]** is On, the count value of D550 increases by 1 and the M code execution flag is set to On.
9. When you set M299 to On, which indicates M99 action complete, the **[M, S, and T codes execution complete]** flag is set to On because the condition is met.
10. Once the **[M, S, and T codes execution]** flag is set to On, the system sets **[M99 halt]** to Off. Because **[M99 halt function for main program]** is set to On, the system stops machining and returns to the first line of the program. Then, the controller sets its status to “Ready”.
11. M99 halt function for main program is complete.

6.10 1st software limit cancellation / hardware limit cancellation

When the controller is moving the axis, there are software and hardware limit settings to protect the axis from exceeding the allowable range. Two sets of software and hardware limit settings are available. When use, you need to cancel the 1st software limit before using the 2nd software limit. When the axis reaches the hardware limit, the system immediately stops executing all the axis motion commands. To resume the movement, you have to cancel the hardware limit. The following example describes how to cancel the 1st software limit and the hardware limit.

- **MLC special M relays**

Axis	1st software limit cancellation	Hardware limit cancellation
X	M1248	M1070
Y	M1249	
Z	M1250	
A	M1251	
B	M1252	
C	M1253	
U	M1254	
V	M1255	
W	M1256	

[1st software limit cancellation]: M1248 - M1256

When **[1st software limit cancellation]** is set to On, the 1st software limit of the specific axis is canceled.

[Hardware limit cancellation]: M1070

When **[Hardware limit cancellation]** is set to On, the hardware limits of all axes are canceled.

- **Relevant parameters:**

Software limit setting:

Pr602 1st positive software limit; Pr603 1st negative software limit; Pr604 2nd positive software limit; Pr605 2nd negative software limit.

The system refers to the parameter settings for software limits and the corresponding alarm occurs when the axis reaches the limit position.

✓ If you are using an incremental motor, perform homing procedure once the controller is power cycled to have the software limit setting take effect.

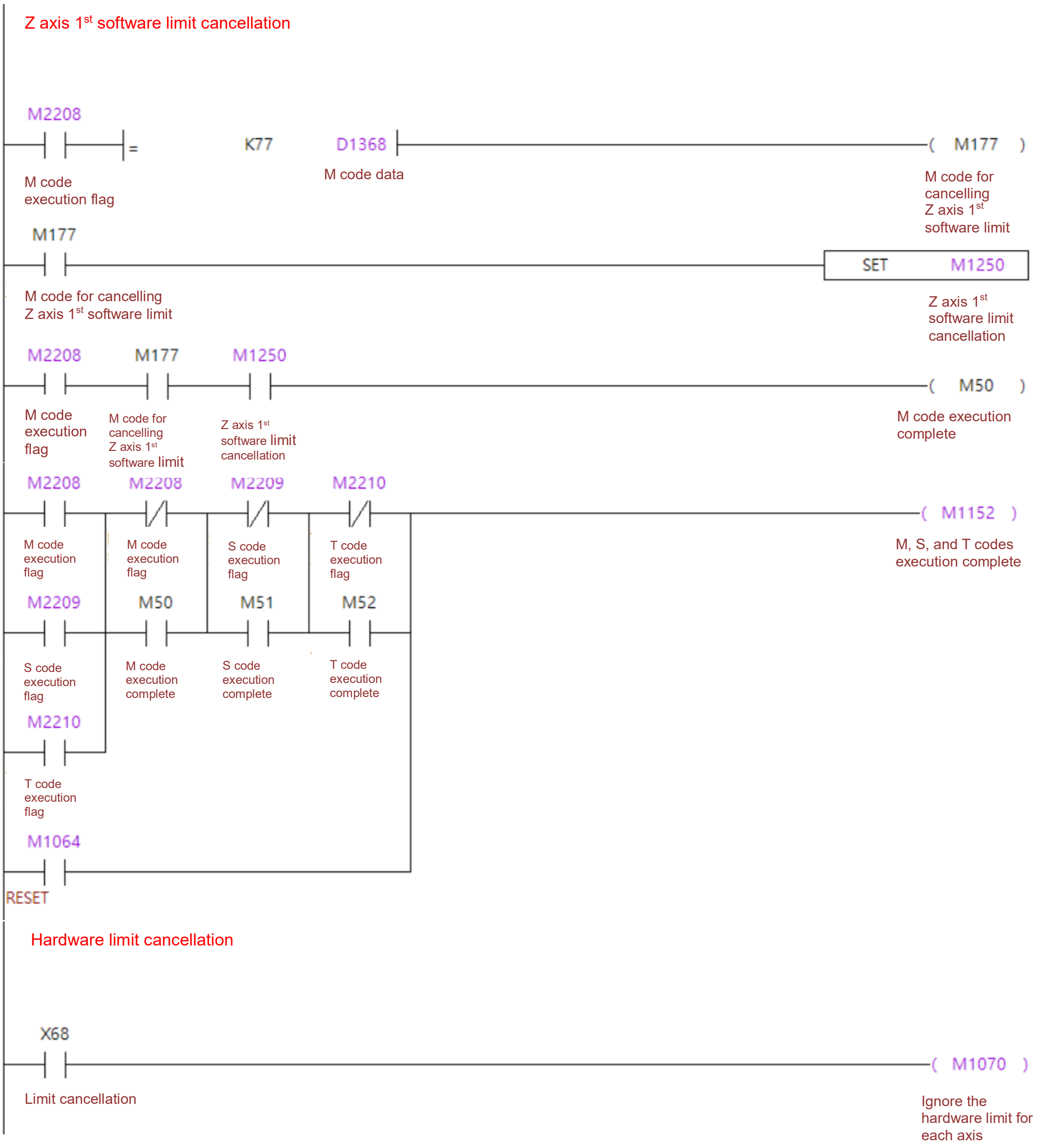
Hardware limit setting:

You can refer to the hardware limit settings for each axis to set Pr628 (Positive limit) and Pr628 (Negative limit) and have the system correctly identify the hardware limit status of each axis.

■ MLC example

The following example uses the M code to disable the 1st software limit of the Z axis and provides the descriptions for manually triggering the signal for hardware limit cancellation.

Disable the 1st software limit of Z axis	
Axis	Z axis
Environment settings	The 1st software positive limit is set to 70. The 2nd software positive limit is set to 140.
Program execution	G90G54X100. G0Z-50. M77 G0Z134. M30


Z axis 1st software limit cancellation
M2208
=
K77
D1368
M177
M code execution flag
M code data
M code for cancelling Z axis 1st software limit
M177
SET M1250
M code for cancelling Z axis 1st software limit
Z axis 1st software limit cancellation
M2208
M177
M1250
M50
M code execution flag
M code for cancelling Z axis 1st software limit
Z axis 1st software limit cancellation
M code execution complete
M2208
M2208
M2209
M2210
M1152
M code execution flag
M code execution flag
S code execution flag
T code execution flag
M, S, and T codes execution complete
M2209
M50
M51
M52
S code execution flag
M code execution complete
S code execution complete
T code execution complete
M2210
T code execution flag
M1064
RESET
Hardware limit cancellation
X68
M1070
Limit cancellation
Ignore the hardware limit for each axis

Program execution procedure

Software limit

1. When the system is executing the main program and executes M77, M177 is set to On.
2. When M177 is On, **[Z axis 1st software limit cancellation]** is also set to On.
3. The system does not send the alarm for reaching the 1st software limit when it continues to execute the main program and G0Z134 is executed.

Hardware limit

The hardware limit alarm occurs when the mechanical part reaches the hardware limit in AUTO or MDI mode operation.

You can press the key for canceling the limit. Once the conditions are met, **[Hardware limit cancellation]** goes to On, and the controller temporarily ignores the hardware limit signal. You can move the mechanical part to a safe position with JOG or MPG operation.

- ✓ When the hardware limit is canceled, pay attention to the axis moving direction to prevent damage to the mechanical parts.

6.11 Spindle control (forward / reverse operation; stop; positioning; speed override)

On the controller, the spindle is controlled by multiple special M relays and special D registers. The description for forward / reverse operation, positioning, and speed override of the spindle is as follows.

- **MLC special M relays and special D registers**

1st spindle forward operation	M1120
1st spindle reverse operation	M1121
1st spindle positioning control	M1124
1st spindle speed command source	M1307
1st spindle reaches target speed	M2256
1st spindle reaches zero speed	M2257
1st spindle positioning complete	M2258
1st spindle speed adjustment ratio	D1060
1st spindle speed (written with special D register)	D1148
1st spindle command speed (32-bit)	D1364
1st spindle actual speed (32-bit)	D1380

[1st spindle forward operation] and [1st spindle reverse roperation]: M1120 and M1121

When **[1st spindle forward operation]** or **[1st spindle reverse operation]** is On, the spindle starts rotating forward or reversely referring to the S code data.

[1st spindle positioning control]: M1124

When **[1st spindle positioning control]** is On, the spindle uses the Z pulse as the zero degree and positions based on the the setting of Pr405 (Spindle positioning offset). When the spindle completes positioning, it stops rotating.

[1st spindle speed command source]: M1307

When **[1st spindle speed command source]** is On, the spindle speed command refers to [1st spindle speed (written with special D register)]. When **M1307 [1st spindle speed command source]** is set to Off, the spindle speed command refers to the S code setting in the program.

[1st spindle reaches target speed]: M2256

When the spindle rotates forward or reversely and reaches the set target speed, the system sets **[1st spindle reaches target speed]** to On.

- ✓ When the rotation speed changes, **[1st spindle reaches target speed]** is set to Off. Until the spindle reaches the target speed, **[1st spindle reaches target speed]** is set to On again.
- ✓ When the speed command is 0 and the spindle stops, **[1st spindle reaches target speed]** is set to On.
- ✓ The duration when **[1st spindle target speed reached]** is On is influenced by Pr406 (Spindle target speed deviation).

[1st spindle reaches zero speed]: M2257

When the spindle speed reaches zero or the spindle stops, the system sets **M2257 [1st spindle reaches zero speed]** to On.

- ✓ When the spindle starts rotating forward or reversely, **M2257** is set to Off.
- ✓ When the rotation command becomes a non-zero value, **M2257** is set to Off.

[1st spindle positioning complete]: M2258

When **M1124 [1st spindle positioning control]** is On and the spindle completes the positioning based on Pr405 (Spindle positioning offset), the system sets **M2258 [1st spindle positioning complete]** to On.

- ✓ When the spindle starts rotating, **M2258** is set to Off.

[1st spindle speed adjustment ratio]: D1060

When the spindle starts rotating forward or reversely, you can refer to the S speed set in the program and **D1060** to adjust the spindle speed with the speed override.

- ✓ The applicable input rage for **D1060 [1st spindle speed adjustment]** is 0 - 65535.
- ✓ When the spindle command speed exceeds the maximum spindle speed set in Pr409, the spindle speed is limited by Pr409 (Spindle maximum speed).
- ✓ When the value of **D1060 [1st spindle speed adjustment ratio]** is changed, the spindle speed changes immediately.

[1st spindle speed (written with special D register)]: D1148

When M1307 [1st spindle speed command source] is set to On, the spindle speed command refers to **D1148 [1st spindle speed (written with special D register)]**.

- ✓ The applicable input rage for **D1148 [1st spindle speed (written with special D reigster)]** is 0 - 65535.
- ✓ When the spindle command speed exceeds the maximum spindle speed set in Pr409, the spindle speed is limited by Pr409 (Spindle maximum speed).
- ✓ When the value of **D1148 [1st spindle speed (written with special D register)]** is changed, the spindle speed changes immediately.

[1st spindle command speed (32-bit)]: D1364

When the controller executes the S code in the program, it writes the value of this S code to **[S code data]** as well as **D1364 [1st spindle command speed (32-bit)]**. If the S code exceeds the range of 16 bits (65535), you can use **D1364** to get the correct data.

- ✓ D1365 is the high word for **[1st spindle command speed (32-bit)]**.

[1st spindle actual speed (32-bit)]: D1380

When the spindle rotates forward or reversely, **D1380 [1st spindle actual speed (32-bit)]** displays the spindle's actual speed.

- ✓ D1381 is the high word for **[1st spindle actual speed (32-bit)]**.
- ✓ The value of **[1st spindle actual speed (32-bit)]** will refer to the setting of Pr399 [Bit 12] (Spindle speed D1380 display mode).
  a. When Pr399 [Bit 12] = 0, the source is the S code in the program.
  b. When Pr399 [Bit 12] = 1, the source is the actual speed of the spindle.

■ **Relevant parameters:**

Spindle application function settings:

On the controller, the spindle application function switches to On or Off by referring to Pr399 (Spindle application setting). The relevant settings are as follows.

- ✓ Spindle function switch; 0: Off; 1: On.
- ✓ Analog closed loop control; 0: Off; 1: On.
- ✓ Spindle output mode; 0: communication mode; 1: reserved; 2: analog mode.
- ✓ Speed control mode; 0: reserved; 1: rpm.
- ✓ Spindle encoder magnification; 0: 1000 times; 1: 4 times
- ✓ Source of the analog spindle speed; 0: refer to the command; 1: refer to the encoder.
- ✓ Analog spindle feedback encoder setting; 0: use the spindle encoder; 1: use the motor encoder.
- ✓ Spindle speed reference; 0: the program; 1: the default spindle speed of Pr398
- ✓ Spindle max. speed command check; 0: Off; 1: On.
- ✓ Spindle speed D1380 display mode; 0: displays S code; 1: displays the current speed.
- ✓ Spindle voltage output mode; 0: ±10V; 1: 0 to 10V.
- ✓ Multi-stage spindle feedback encoder switch; 0: Off; 1: On.

Speed parameter settings:

Spindle speed control is set based on the following parameters.

- ✓ Pr398 (Spindle speed default setting): the S code default value after the controller is powered on.
- ✓ Pr406 (Deviation from the spindle target speed): when the deviation between the actual spindle speed and the command speed is within the range set in Pr406, the system sets **[Spindle reaches the target speed]** to On.

  Example: when the command speed is 1000 and the deviation is set to 100, if the actual speed is 900 to 1100 rpm, and the system sets **[Spindle reaches the target speed]** to On.
- ✓ Pr408 (Deviation from spindle zero speed): when the deviation between the actual spindle speed and the zero speed is within the range set in Pr408, the system sets **[Spindle reaches the zero speed]** to On.

  Example: when the zero speed deviation is set to 100 and the actual speed is less than 100 rpm, the system sets **[Spindle reaches the zero speed]** to On.
- ✓ Pr409 (Spindle max. speed): when the spindle is a communication spindle, this parameter limits the spindle's maximum speed. When the spindle is controlled by analog voltage, this

parameter is the reference for the spindle speed voltage ratio calculation.

- ✓ Pr411 (Spindle acceleration/deceleration time constant): sets the acceleration and deceleration time of the spindle.
- ✓ Pr412 (Spindle S-curve time constant): sets the S-curve time constant when the spindle accelerates and decelerates.

Positioning settings:

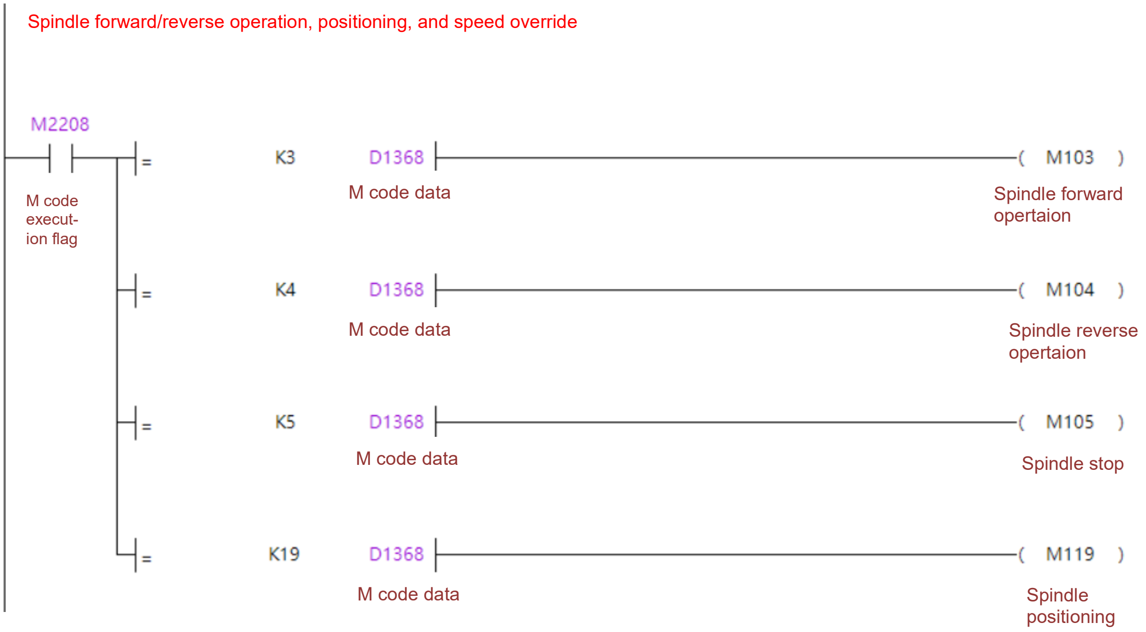
Parameters for spindle positioning function are as follows.

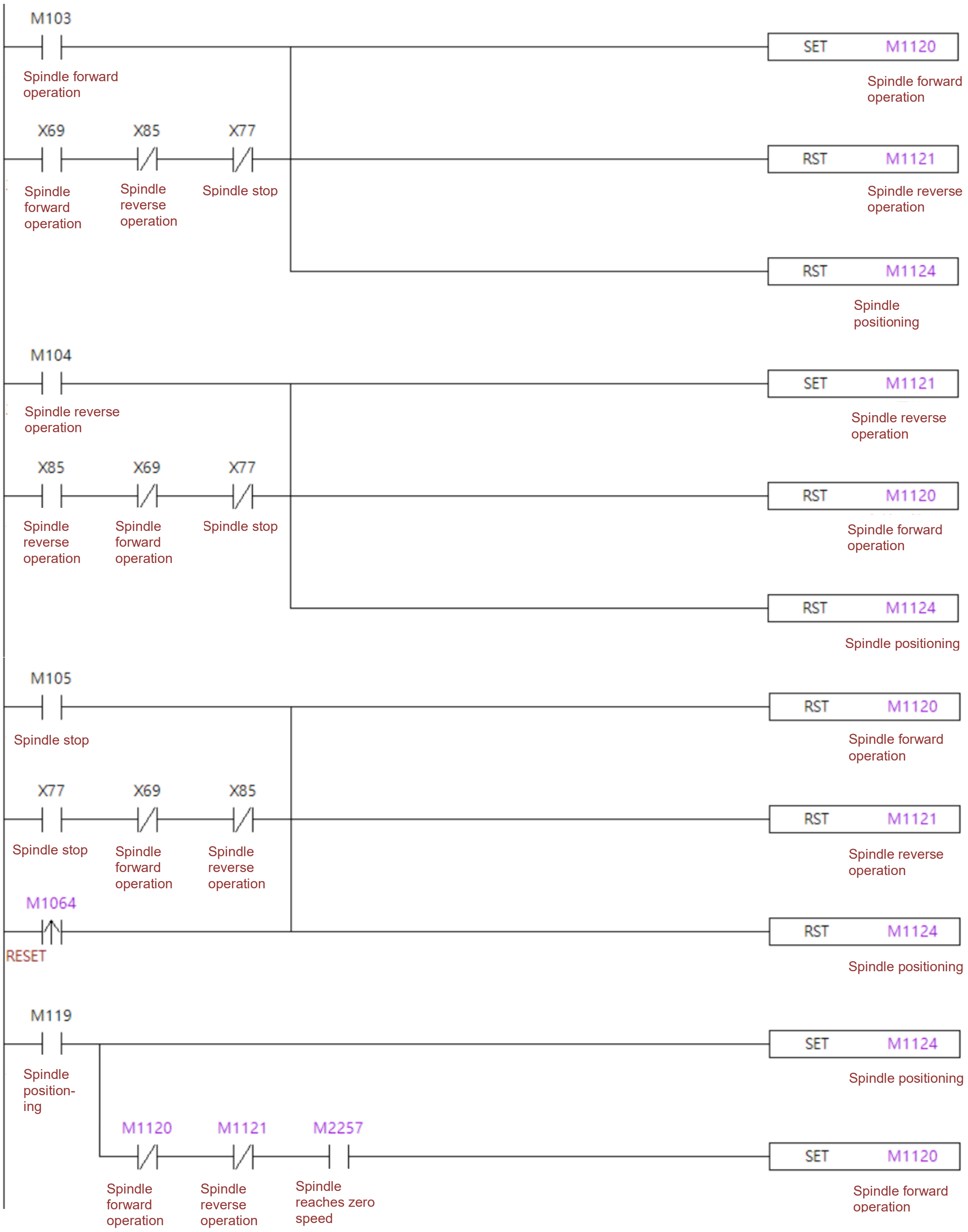
- ✓ Pr404 (Spindle high-speed positioning speed): when the spindle rotates and the controller receives the spindle positioning command, the spindle positions at the speed set in Pr404.
- ✓ Pr405 (Spindle positioning offset (Z phase)): when the controller receives the spindle positioning command, the spindle positions at the position, which is the Z phase plus the distance set in Pr405.
- ✓ Pr406 (Spindle positioning deviation): when the spindle is positioning and the deviation between the actual position and the position set in Pr405 is within the value set in Pr406, the system sets **[Spindle positioning complete]** to On.

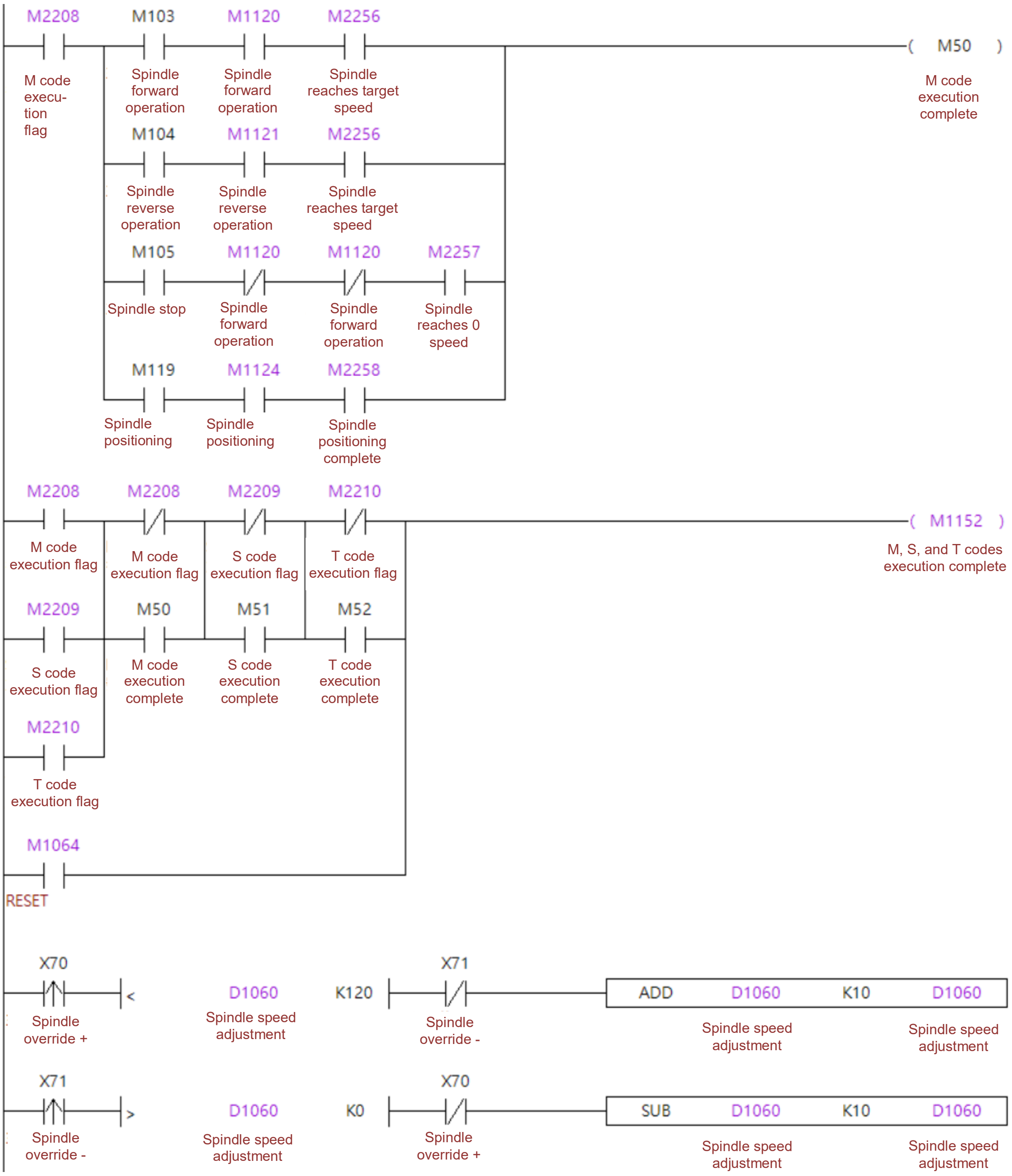
  Example: when the deviation is set to 500 and the spindle offset is 1000, the spindle angle is within 5 - 15 degrees, and the system sets **[Spindle positioning complete]** to On.
- ✓ Pr420 (Spindle low-speed positioning): when the spindle is rotating and the controller receives the spindle positioning command, the spindle positions at the speed set in Pr420.

■ MLC example

The following illustrates the usage of spindle forward / reverse operation, stop, positioning, and speed override.




M103
Spindle forward operation
SET M1120
Spindle forward operation
X69
X85
X77
Spindle forward operation
Spindle reverse operation
Spindle stop
RST M1121
Spindle reverse operation
RST M1124
Spindle positioning
M104
Spindle reverse operation
SET M1121
Spindle reverse operation
X85
X69
X77
Spindle reverse operation
Spindle forward operation
Spindle stop
RST M1120
Spindle forward operation
RST M1124
Spindle positioning
M105
Spindle stop
RST M1120
Spindle forward operation
X77
X69
X85
Spindle stop
Spindle forward operation
Spindle reverse operation
RST M1121
Spindle reverse operation
M1064
RESET
RST M1124
Spindle positioning
M119
Spindle position-ing
SET M1124
Spindle positioning
M1120
M1121
M2257
Spindle forward operation
Spindle reverse operation
Spindle reaches zero speed
SET M1120
Spindle forward operation



Program execution procedure

Forward / reverse operation and stop

1. If you press the key of spindle forward operation, reverse operation, or stop operation, the corresponding **[Spindle forward operation]** or **[Spindle reverse operation]** special M relay is set On or Off to have the spindle rotate forward, reversely, or stopped.
2. When the program runs to M3, M4, or M5, it uses the M code procedure to set **[Spindle forward operation]** or **[Spindle reverse operation]** to On or Off to have the spindle rotate forward, reversely, or stopped.
3. Confirm the spindle status with **[Spindle reaches target speed]** and **[Spindle reaches zero speed]** and end the M code procedure.

Spindle positioning

1. When M19 is executed in the program, the M code procedure sets **[Spindle positioning control]** to On.
2. Confirm the spindle positioning is complete with **[Positioning complete]** and end the M code procedure.

Spindle speed adjustment ratio

Use the key signal to increase or decrease the spindle speed adjustment ratio. The maximum is 120 and the minimum is 0. Each trigger increases or decreases the ratio by 10 and writes the ratio to **[Spindle speed adjustment ratio]**.

6.12 Spindle gear ratio switch

In the controller, there are four sets of spindle gear ratio parameters and need to be switched with the MLC. The description of the spindle gear ratio switch is as follows.

■ MLC special M relays

Spindle gear ratio switching and state devices:

	0: use Pr422 and Pr423 as the gear ratio	1: use Pr424 and Pr425 as the gear ratio	2: use Pr426 and Pr427 as the gear ratio	3: use Pr428 and Pr429 as the gear ratio
Bit 0	M1122	M1122	M1122	M1122
Bit 1	M1123	M1123	M1123	M1123

Note: the special M relay in gray indicates it is Off; the special M relay in black indicates it is On.

[Spindle gear ratio selection Bit 0] and [Spindle gear ratio selection Bit 1]: M1122 and M1123

When you need to switch the spindle gear ratio, set **[Spindle gear ratio selection Bit 0] and [Spindle gear ratio selection Bit 1]** to On / Off in bit format (as shown in the above table).

- ✓ When writing / editing the ladder diagram in the MLC, if you intend to use MOV K0-3 K1M1122, be aware of that M1124 and M1125 will also be influenced.

■ Relevant parameters:

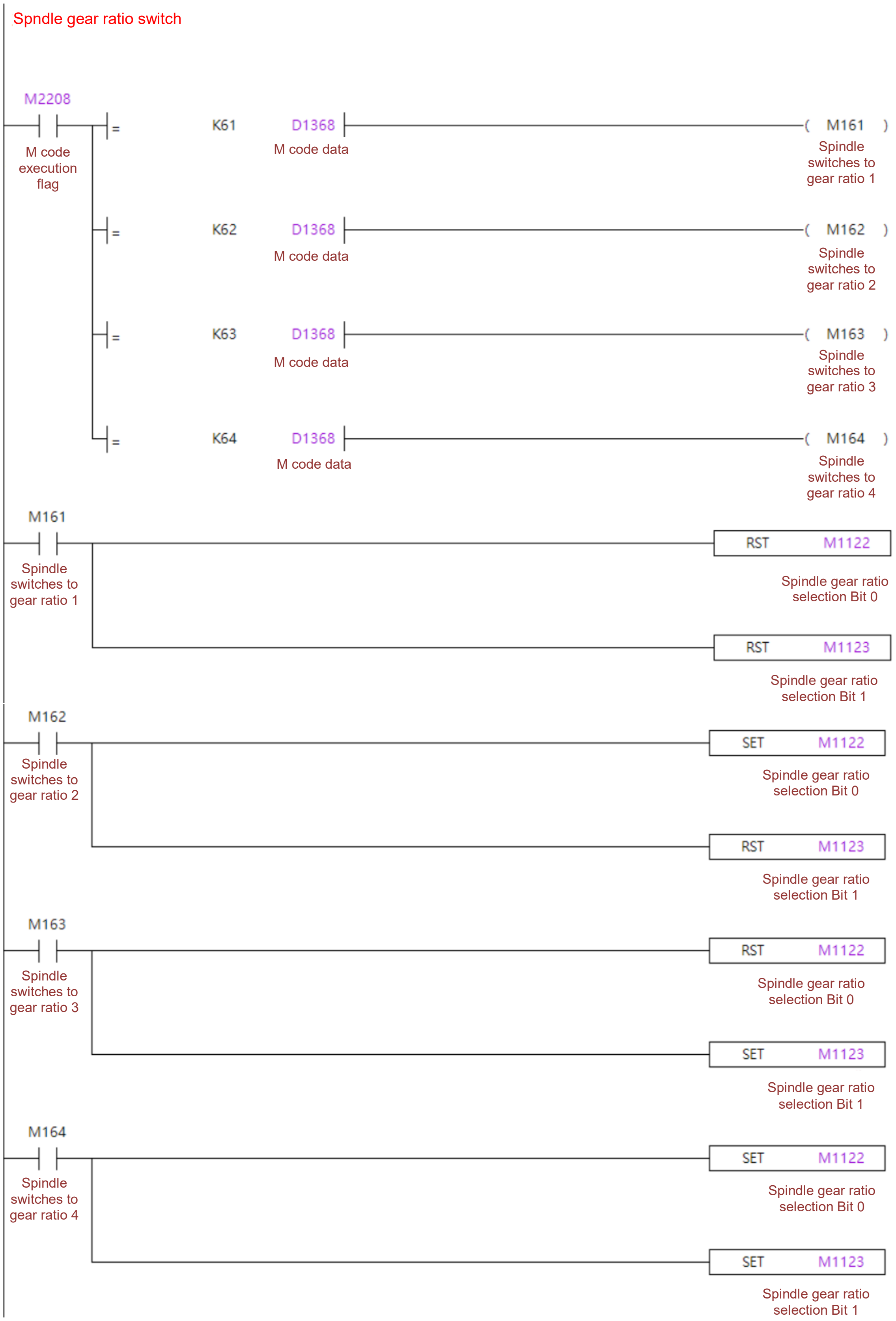
Spindle gear ratio setting:

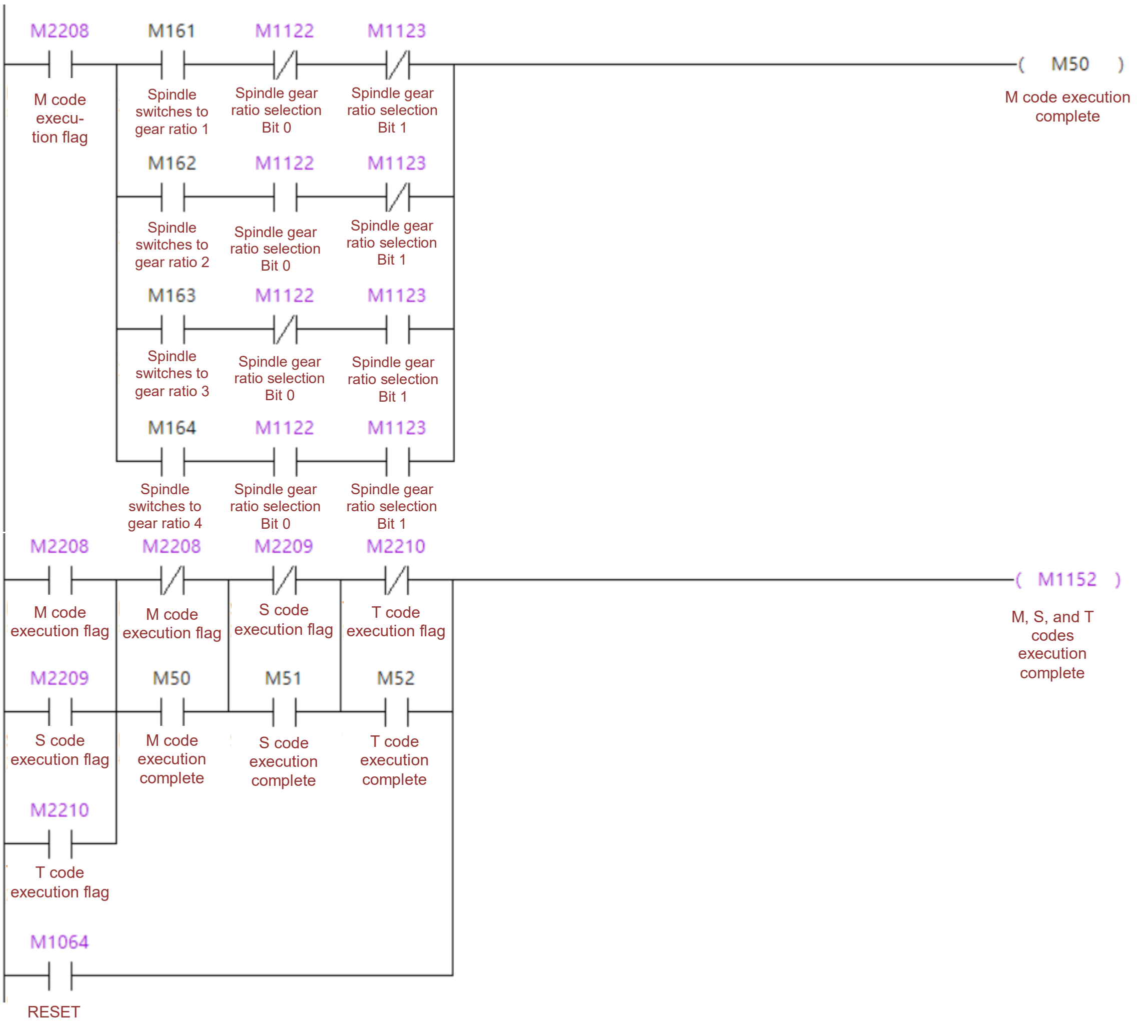
When you switch the gear ratio with **[Spindle gear ratio selection Bit 0 and [Spindle gear ratio selection Bit 1]**, the system refers to Pr422 - Pr429 to switch between gear ratios. The system provides four sets of gear ratios for switching.

- ✓ Pr422: gear ratio numerator 1, pairing with Pr423 as the same set of gear ratio.
  Pr423: gear ratio denominator 1, pairing with Pr422 as the same set of gear ratio.
- ✓ Pr424: gear ratio numerator 2, pairing with Pr425 as the same set of gear ratio.
  Pr425: gear ratio denominator 2, pairing with Pr424 as the same set of gear ratio.
- ✓ Pr426: gear ratio numerator 3, pairing with Pr427 as the same set of gear ratio.
  Pr427: gear ratio denominator 3, pairing with Pr426 as the same set of gear ratio.
- ✓ Pr428: gear ratio numerator 4, pairing with Pr429 as the same set of gear ratio.
  Pr429: gear ratio denominator 4, pairing with Pr428 as the same set of gear ratio.

■ MLC example

When switching between gear ratios during program execution, you need to use the M code to complete the switching, instead of simply setting **[Spindle gear ratio selection Bit 0]** and **[Spindle gear ratio selection Bit 1]** to On. The example of using the M code to switch the gear ratio is as follows.


Spndle gear ratio switch
M2208
M code execution flag
=
K61
D1368
M code data
M161
Spindle switches to gear ratio 1
=
K62
D1368
M code data
M162
Spindle switches to gear ratio 2
=
K63
D1368
M code data
M163
Spindle switches to gear ratio 3
=
K64
D1368
M code data
M164
Spindle switches to gear ratio 4
M161
Spindle switches to gear ratio 1
RST M1122
Spindle gear ratio selection Bit 0
RST M1123
Spindle gear ratio selection Bit 1
M162
Spindle switches to gear ratio 2
SET M1122
Spindle gear ratio selection Bit 0
RST M1123
Spindle gear ratio selection Bit 1
M163
Spindle switches to gear ratio 3
RST M1122
Spindle gear ratio selection Bit 0
SET M1123
Spindle gear ratio selection Bit 1
M164
Spindle switches to gear ratio 4
SET M1122
Spindle gear ratio selection Bit 0
SET M1123
Spindle gear ratio selection Bit 1


M2208
M code execu-tion flag
M161
Spindle switches to gear ratio 1
M1122
Spindle gear ratio selection Bit 0
M1123
Spindle gear ratio selection Bit 1
M50
M code execution complete
M162
Spindle switches to gear ratio 2
M163
Spindle switches to gear ratio 3
M164
Spindle switches to gear ratio 4
M2208
M code execution flag
M2209
S code execution flag
M2210
T code execution flag
M1152
M, S, and T codes execution complete
M51
S code execution complete
M52
T code execution complete
M1064
RESET

Program execution procedure:

1. When M61 is executed in the program, the system uses the M code procedure to set **[Spindle gear ratio selection Bit 0]** and **[Spindle gear ratio selection Bit 1]** to Off.
2. Use M161 to switch the spindle to gear ratio 1 and use **[Spindle gear ratio selection Bit 0]** and **[Spindle gear ratio selection Bit 1]** to confirm the switching is complete, and then complete the M code procedure.
3. When M62 is executed during program execution, the system uses the M code procedure to set **[Spindle gear ratio selection Bit 0]** to On and **[Spindle gear ratio selection Bit 1]** to Off.
4. Use M162 to switch the spindle gear ratio to 2 and use **[Spindle gear ratio selection Bit 0]** and **[Spindle gear ratio selection Bit 1]** to confirm the gear switching is complete, and then complete the M code procedure.
5. When M63 is executed during program execution, the system uses the M code procedure to set **[Spindle gear ratio selection Bit 0]** to Off and **[Spindle gear ratio selection Bit 1]** to On.
6. Use M163 to switch the spindle gear ratio to 3 and use **[Spindle gear ratio selection Bit 0]** and **[Spindle gear ratio selection Bit 1]** to confirm the gear switching is complete, and then complete the M code procedure.
7. When M64 is executed during program execution, the system uses the M code procedure to set **[Spindle gear ratio selection Bit 0]** and **[Spindle gear ratio selection Bit 1]** to On.

8. Use M164 to switch the spindle gear ratio to 42 and use **[Spindle gear ratio selection Bit 0]** and **[Spindle gear ratio selection Bit 1]** to confirm the gear switching is complete, and then complete the M code procedure.

Important:

If you use the M code to switch the gear ratio with **[Spindle gear ratio selection Bit 0]** and **[Spindle gear ratio selection Bit 1],** it only switches the spindle speed command. If there is a physical mechanical part for gear switch, you need to compose the corresponding MLC and the output DO, so the external mechanical part can correctly change the gear ratio.

6.13 Lathe spindle / C axis mode switch

On the lathe system which integrates both lathe and milling functions, applications of switching between the spindle and C axis are frequently used. Descriptions for the spindle / C axis switch of the lathe system is as follows.

- **MLC special M relays**

Lathe system spindle / C axis mode switch function	M1126
Lathe system spindle / C axis mode switched	M2239

[Lathe system spindle / C axis mode switch function]: M1126

When **M1126 [Lathe system spindle / C axis mode switch function]** is On, the system switches the spindle to C axis and performs positioning.

- ✓ When the system is in AUTO or MDI mode, you need to use the M code procedure to have the switching take effect. If it is in JOG or MPG mode, you can dynamically complete the switching.
- ✓ When **M1126 [Lathe system spindle / C axis mode switch function]** is On, the system automatically sends M29 to perform the M code procedure and the MLC adds the required M code procedure corresponding to M29.
- ✓ When **M1126 [Lathe system spindle / C axis mode switch function]** is On and the positioning is complete, the system sends M05 to perform the M code procedure and the MLC adds the required M code procedure corresponding to M05.

[Lathe system spindle / C axis mode switched]: M2239

After the system switches from spindle to C axis mode, the system sets **M2239 [Lathe system spindle / C axis mode switched]** to On.

- **Relevant parameters:**

M code switching setting:

To change between the spindle and C axis modes in the lathe system, you must use the M code for axis mode switch in AUTO or MDI mode. The system changes the axis mode based on the following parameter settings.

- ✓ Pr358 (M code for changing the lathe Spindle mode to C axis mode): sets the M code for changing from Spindle to C axis mode. This M code automatically stops the function of lookahead.
- ✓ Pr359 (M code for changing the lathe system from C axis to Spindle mode): sets the M code for changing from C axis to Spindle mode. This M code automatically stops the function of lookahead.

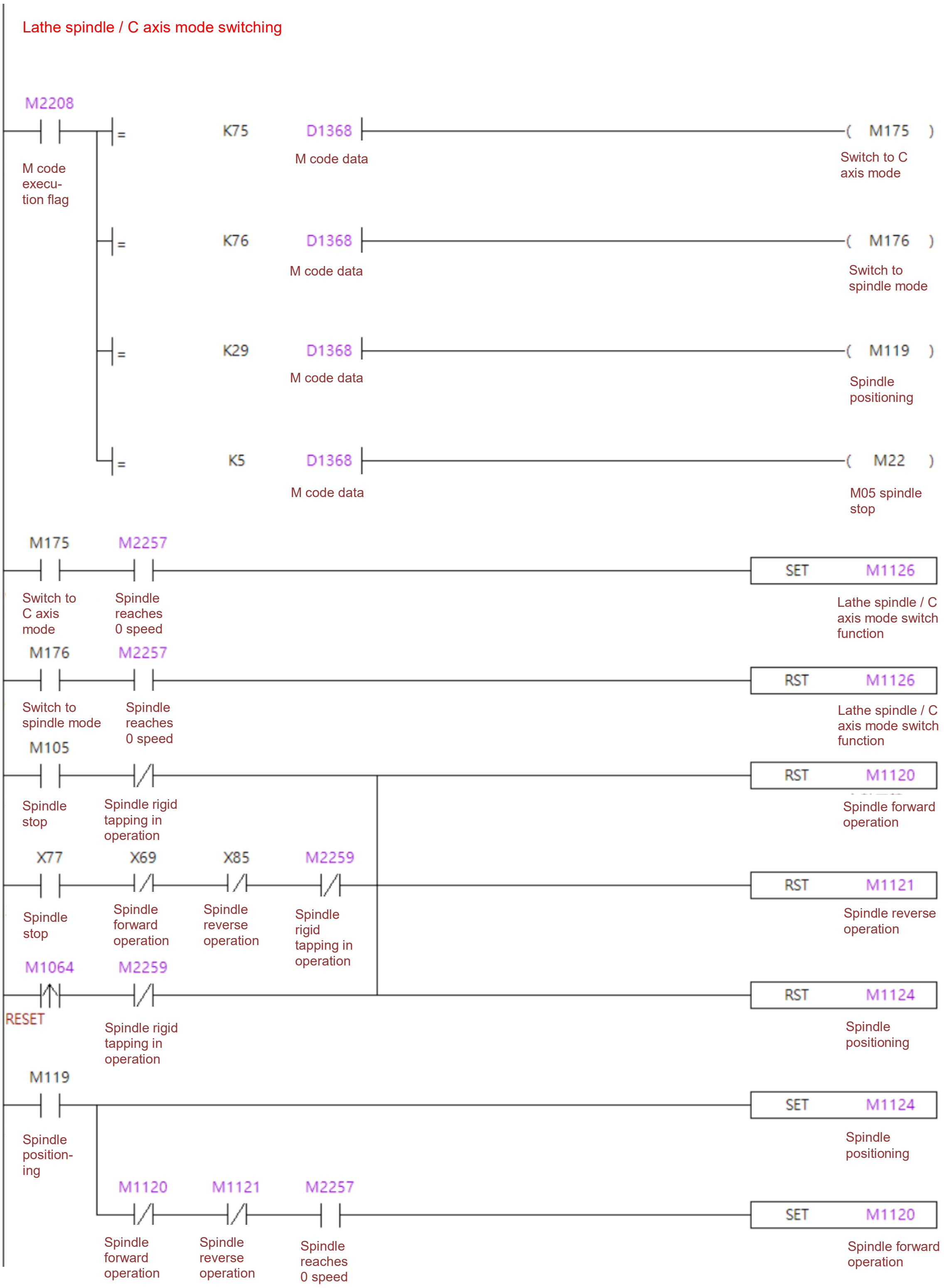
Lathe system C axis mode setting:

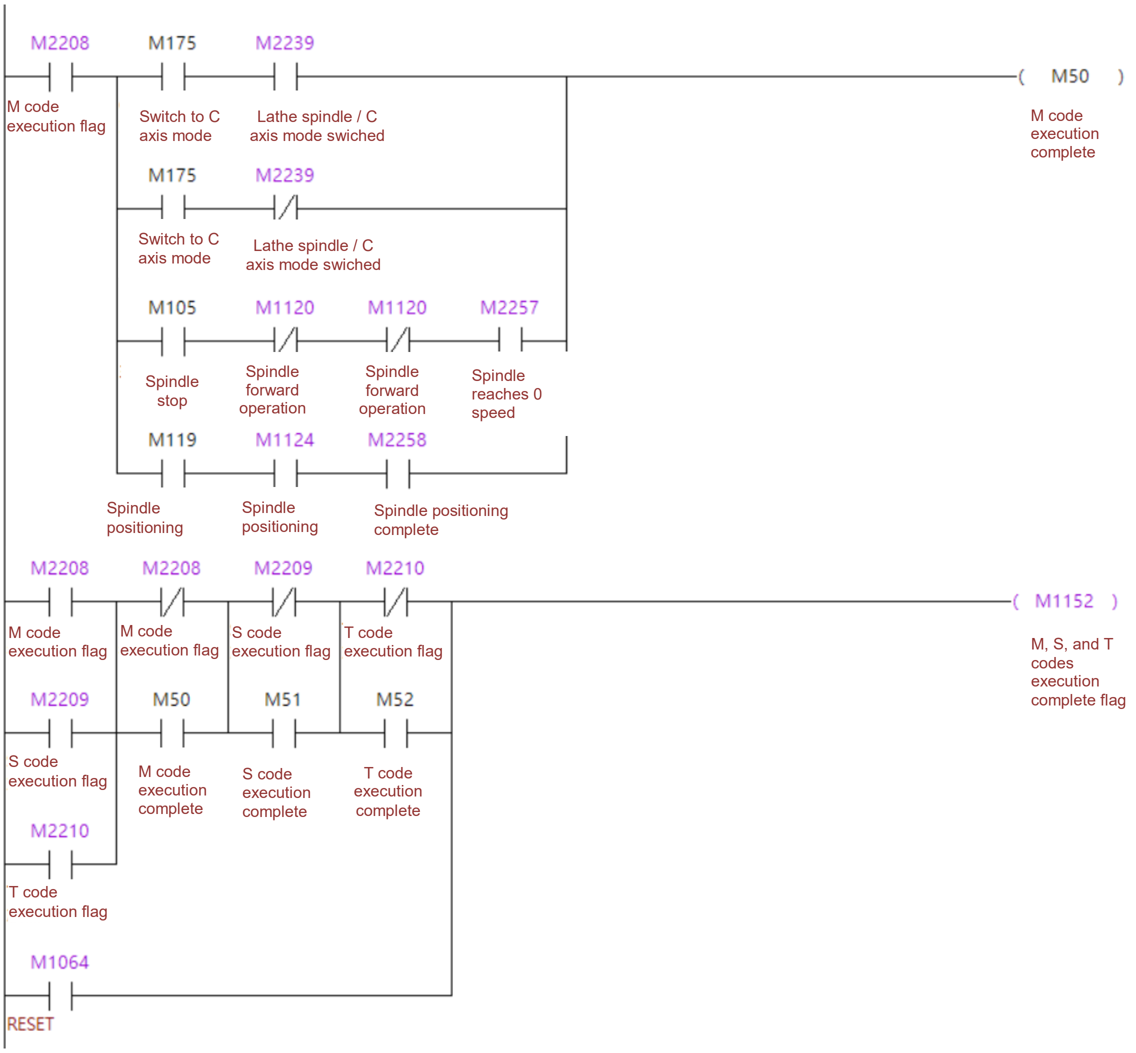
The system refers to the lathe system C axis mode setting to determine whether the spindle / C axis mode switching is activated.

- ✓ Pr308 [Bit 15] (Lathe system C axis mode): when this bit is set to 0, the spindle / C axis mode switching is enabled. When this bit is set to 1, you need to set the C axis and the spindle with the channel settings and the system cannot switch between the two modes.

■ MLC example

The following uses M175 and M176 as an example, in which Pr358 is set to 75 and Pr359 is set to 76.





Program execution procedure

1. When the program runs to M75, M code procedure sets **M1126 [Lathe system spindle / C axis mode switch function]** to On, and the system sends the M29 signal.
2. When M119 is set to On, the system automatically performs positioning. Once the positioning is complete, the system sends the M05 signal.
3. When M22 is set to On, the spindle immediately stops operating and performs positioning, and then the system switches to C axis mode.
4. When the system sets **M2239 [Lathe system spindle / C axis mode switched]** to On, it checks the status of **M2239** and then ends the M code procedure for M75.
5. When M76 is executed during program execution, the system uses the M code procedure to set **M1126 [Lathe system spindle / C axis mode switch function]** to Off.
6. When the system sets **[Lathe system spindle / C axis mode switched]** to Off, it checks the status of **[Lathe system spindle / C axis mode switched]** and ends the M code procedure.

Important:

Ensure the spindle is stopped before you switch between the spindle and C axis modes.

6.14 Tapping interruption and auto retraction

Tapping includes a series of operations. If it is inevitable to press the RESET or EMERGENCY STOP key, or even cut off the power during the operations, the system triggers the signal of tapping interruption and automatically retracts the spindle and Z axis with the MLC.

■ MLC special M relay

Tapping interruption and spindle retraction related special M relay	
Spindle retraction after tapping	M1125 (SP1), M1141 (SP2)
Spindle is in the rigid tapping mode	M2259 (SP1), M2264 (SP2)
Rigid tapping interruption	M2260 (SP1), M2265 (SP2)

Other related special M relay	
Spindle forward operation	M1120 (SP1), M1136 (SP2)
Spindle reverse operation	M1121 (SP1), M1137 (SP2)
Spindle positioning control	M1124 (SP1), M1140 (SP2)
Spindle positioning complete	M2258 (SP1), M2263 (SP2)
Z axis forward jog control	M1218

[Spindle retraction after tapping]: M1125, M1141

If the system is interrupted when performing tapping, set **[Spindle retraction after tapping]** to on to have the system retract the tap with the auto retraction function.

✓ Do not set **[Spindle retraction after tapping]** to on when the system is performing tapping.

[Spindle is in the rigid tapping mode]: M2259, M2264

The system sets **[Spindle is in the rigid tapping mode]** to on when performing tapping.

✓ **[Spindle is in the rigid tapping mode]** is set to off when RESET or EMERGENCY STOP is triggered.

[Rigid tapping interruption]: M2260, M2265

The system sets **[Rigid tapping interruption]** to on when the tapping is interrupted.

✓ Avoid the following operations when **[Rigid tapping interruption]** is set to on. Otherwise, the system disables the tapping interruption and sets **[Rigid tapping interruption]** to off.

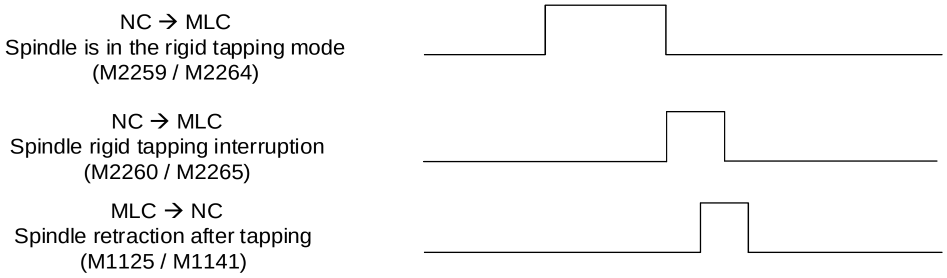
a. Set [Spindle positioning control] and [Spindle forward operation] to off.
b. Set [Cycle Start] to on again.
c. Move any axes.
d. Cycle power to the system when **[Rigid tapping interruption]** is on.

✓ When Pr307 [Bit 8 - 9] (EMG stop mode) is 0 and EMERGENCY STOP is triggered, the servo becomes off and **[Rigid tapping interruption]** remains off. When Pr307 [Bit 8 - 9] is 1 and EMERGENCY STOP is triggered, **[Rigid tapping interruption]** becomes off when the servo is off.

✓ When **[Rigid tapping interruption]** is set to on, set [Spindle retraction after tapping] to on and complete the retraction, and then **[Rigid tapping interruption]** is set to off.

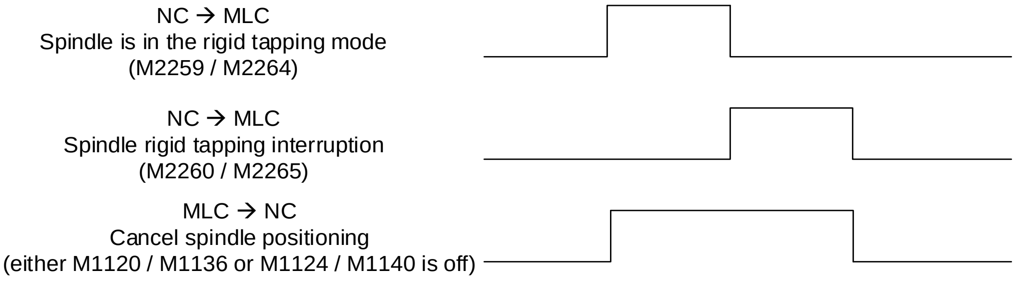
Time sequence description:

A. Timing diagram of M2259 / M2264, M2260 / M2265, and M1125 / M1141


NC → MLC
Spindle is in the rigid tapping mode
(M2259 / M2264)
NC → MLC
Spindle rigid tapping interruption
(M2260 / M2265)
MLC → NC
Spindle retraction after tapping
(M1125 / M1141)

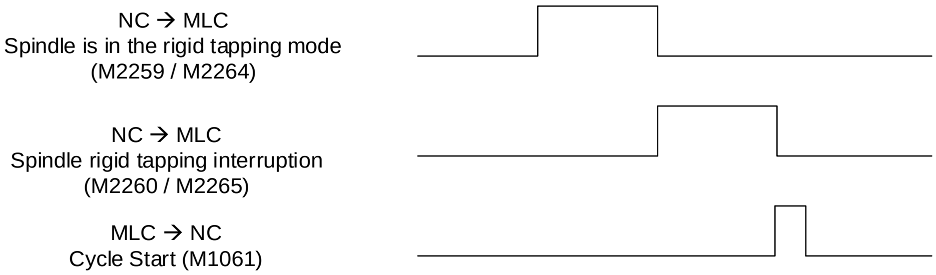
1. When the system starts performing tapping, it sets **[Spindle is in the rigid tapping mode]** to on.
2. If any interrupt occurs during tapping when [Spindle forward / reverse operation] and [Spindle positioning control] are on, the system sets **[Spindle is in the rigid tapping mode]** to off and **[Spindle rigid tapping interruption]** to on at the same time.
3. When **[Spindle rigid tapping interruption]** is on, set **[Spindle retraction after tapping]** to on, and the system automatically retracts the spindle and Z axis.
4. After the retraction is complete, **[Spindle rigid tapping interruption]** is set to off.
5. When **[Spindle rigid tapping interruption]** is set to off, you can set **[Spindle retraction after tapping]** to off.

B. Timing diagram of M2259 / M2264, M2260 / M2265, and M1120 / M1136 or M1124 / M1140


NC → MLC
Spindle is in the rigid tapping mode
(M2259 / M2264)
NC → MLC
Spindle rigid tapping interruption
(M2260 / M2265)
MLC → NC
Cancel spindle positioning
(either M1120 / M1136 or M1124 / M1140 is off)

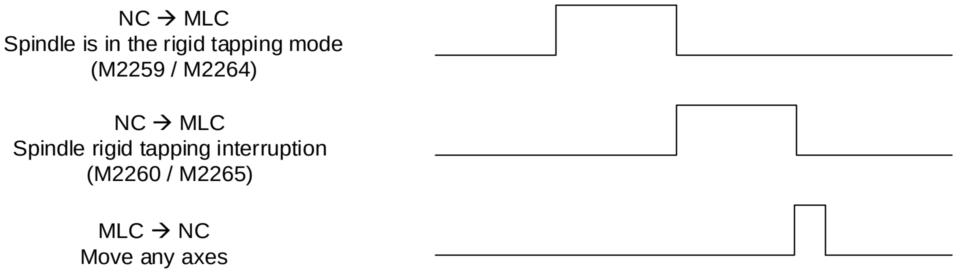
1. When the system starts performing tapping, it sets **[Spindle is in the rigid tapping mode]** to on.
2. If any interrupt occurs during tapping when [Spindle forward / reverse operation] and [Spindle positioning control] are on, the system sets **[Spindle is in the rigid tapping mode]** to off and **[Spindle rigid tapping interruption]** to on at the same time.
3. When **[Spindle rigid tapping interruption]** is on, if [Spindle forward operation] or [Spindle positioning control] is set to off, the system sets **[Spindle rigid tapping interruption]** to off at the same time.

C. Timing diagram of M2259 / M2264, M2260 / M2265, and M1061



1. When the system starts performing tapping, it sets **[Spindle is in the rigid tapping mode]** to on.
2. If any interrupt occurs during tapping when [Spindle forward / reverse operation] and [Spindle positioning control] are on, the system sets **[Spindle is in the rigid tapping mode]** to off and **[Spindle rigid tapping interruption]** to on at the same time.
3. When M1061 [Cycle Start] is on, the system sets **[Spindle is in the rigid tapping mode]** to off at the same time.

D. Timing diagram of M2259 / M2264, M2260 / M2265, and M1216 - M1219 / M1226 - M1229


NC → MLC
Spindle is in the rigid tapping mode
(M2259 / M2264)
NC → MLC
Spindle rigid tapping interruption
(M2260 / M2265)
MLC → NC
Move any axes

1. When the system starts performing tapping, it sets **[Spindle is in the rigid tapping mode]** to on.
2. If any interrupt occurs during tapping when [Spindle forward / reverse operation] and [Spindle positioning control] are on, the system sets **[Spindle is in the rigid tapping mode]** to off and **[Spindle rigid tapping interruption]** to on at the same time.
3. When M1216 - M1219 [Forward jog operation] or M1226 - M1229 [Reverse jog operation] is set to on for any axes, the system sets **[Spindle rigid tapping interruption]** to off at the same time.

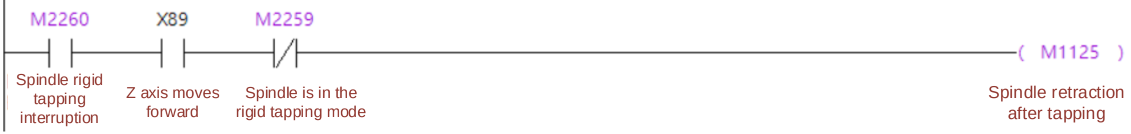
■ Relevant parameter

Pr307 [Bit 8 & Bit 9] (EMG stop mode):

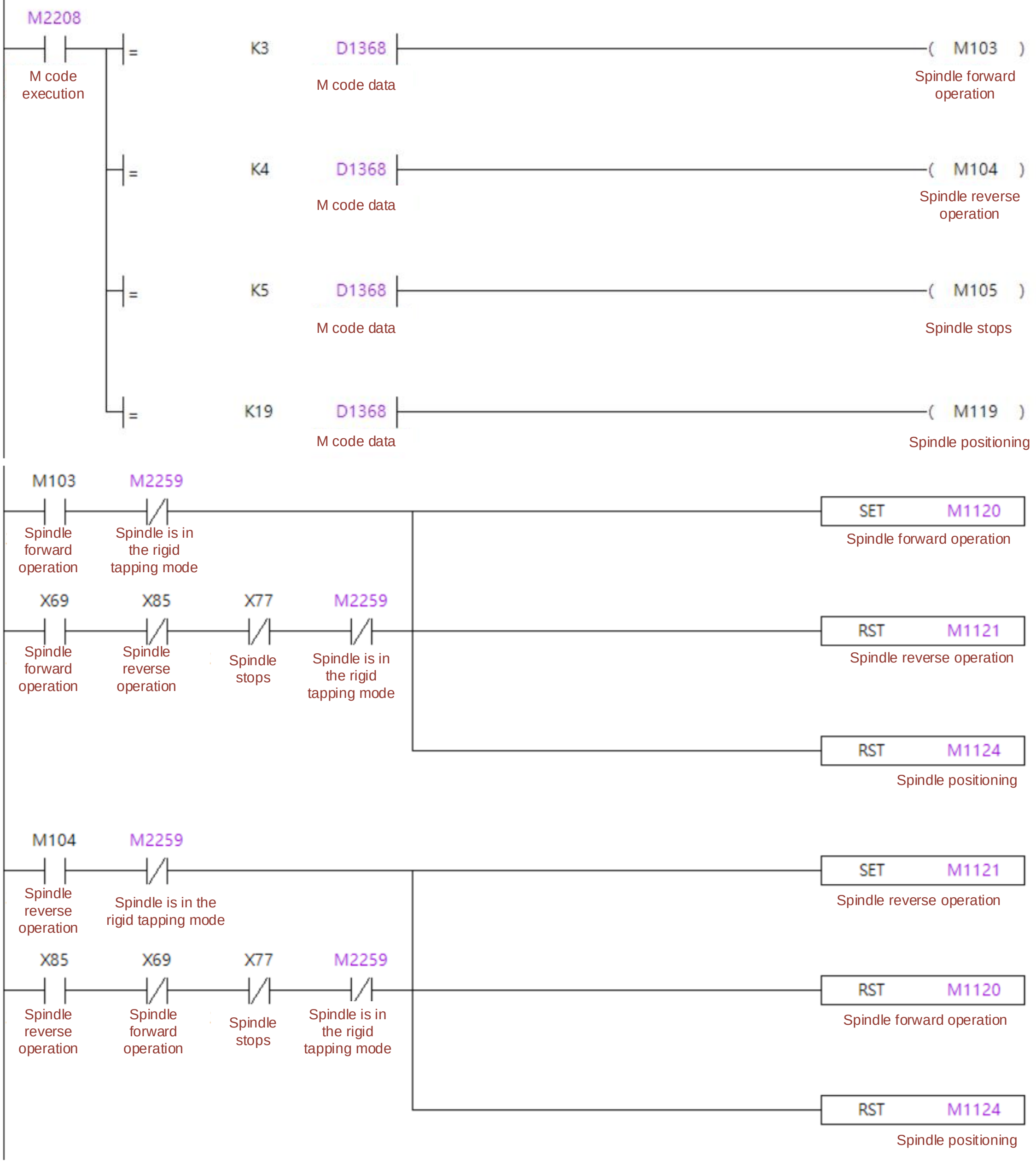
The servo switches to Servo Off and **[Spindle rigid tapping interruption]** is set to off once EMERGENCY STOP is triggered when Pr307 [Bit 8 & 9] is set to 0 or 1. To use the retraction function after EMERGENCY STOP is triggered, you have to set Pr307 [Bit 8 & 9] to 2 for the system to remain Servo On after the system is stopped.

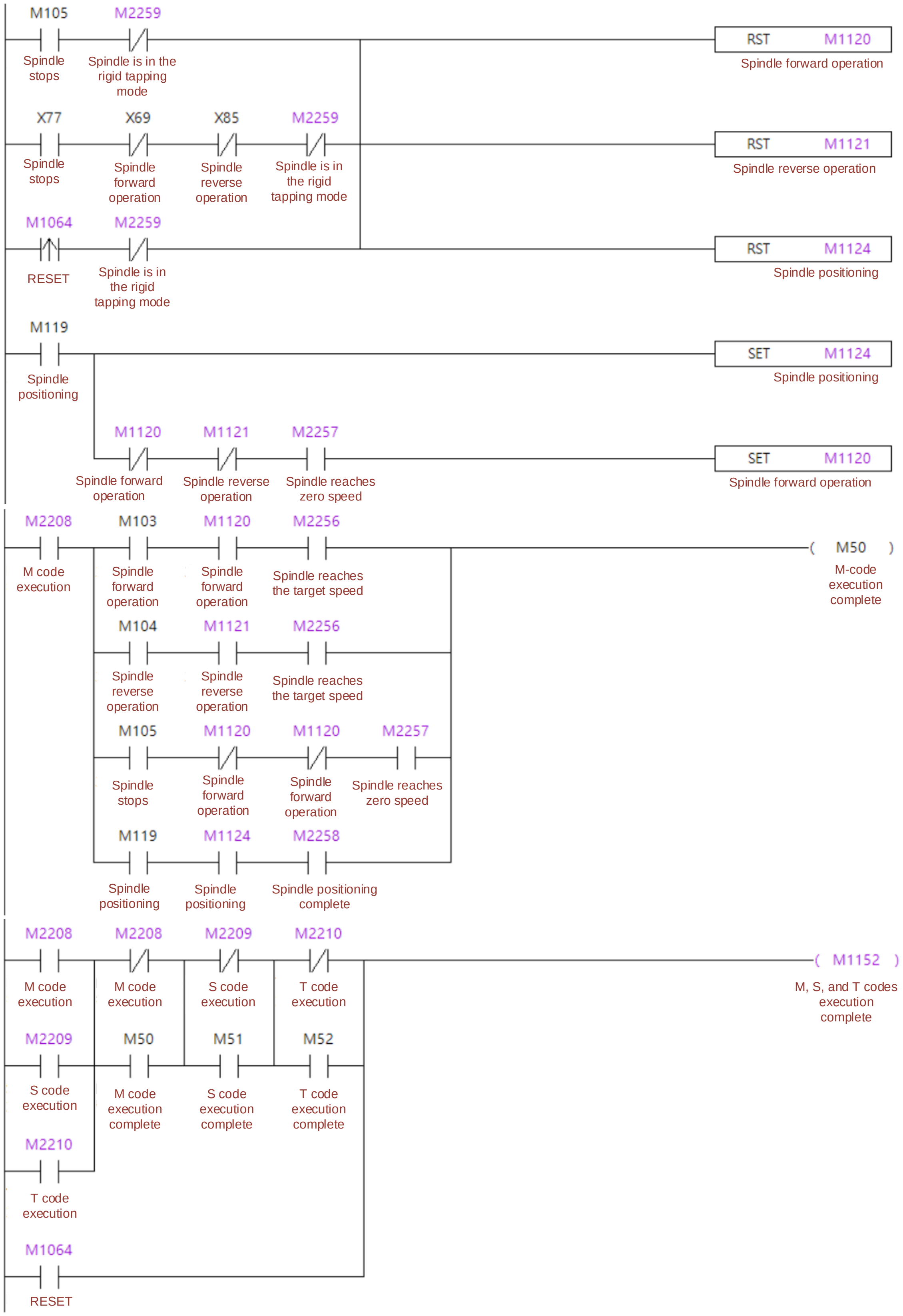
■ MLC example

Spindle tapping and retraction


M2260
X89
M2259
M1125
Spindle rigid tapping interruption
Z axis moves forward
Spindle is in the rigid tapping mode
Spindle retraction after tapping

Spindle forward / reverse operation, positioning, speed override


M2208
M code execution
=
K3
D1368
M code data
M103
Spindle forward operation
=
K4
D1368
M code data
M104
Spindle reverse operation
=
K5
D1368
M code data
M105
Spindle stops
=
K19
D1368
M code data
M119
Spindle positioning
M103
M2259
Spindle forward operation
Spindle is in the rigid tapping mode
SET M1120
Spindle forward operation
X69
X85
X77
M2259
Spindle forward operation
Spindle reverse operation
Spindle stops
Spindle is in the rigid tapping mode
RST M1121
Spindle reverse operation
RST M1124
Spindle positioning
M104
M2259
Spindle reverse operation
Spindle is in the rigid tapping mode
SET M1121
Spindle reverse operation
X85
X69
X77
M2259
Spindle reverse operation
Spindle forward operation
Spindle stops
Spindle is in the rigid tapping mode
RST M1120
Spindle forward operation
RST M1124
Spindle positioning


M105
Spindle stops
M2259
Spindle is in the rigid tapping mode
RST M1120
Spindle forward operation
X77
Spindle stops
X69
Spindle forward operation
X85
Spindle reverse operation
M2259
Spindle is in the rigid tapping mode
RST M1121
Spindle reverse operation
M1064
RESET
M2259
Spindle is in the rigid tapping mode
RST M1124
Spindle positioning
M119
Spindle positioning
SET M1124
Spindle positioning
M1120
Spindle forward operation
M1121
Spindle reverse operation
M2257
Spindle reaches zero speed
SET M1120
Spindle forward operation
M2208
M code execution
M103
Spindle forward operation
M1120
Spindle forward operation
M2256
Spindle reaches the target speed
M50
M-code execution complete
M104
Spindle reverse operation
M1121
Spindle reverse operation
M2256
Spindle reaches the target speed
M105
Spindle stops
M1120
Spindle forward operation
M1120
Spindle forward operation
M2257
Spindle reaches zero speed
M119
Spindle positioning
M1124
Spindle positioning
M2258
Spindle positioning complete
M2208
M code execution
M2208
M code execution
M2209
S code execution
M2210
T code execution
M1152
M, S, and T codes execution complete
M2209
S code execution
M50
M code execution complete
M51
S code execution complete
M52
T code execution complete
M2210
T code execution
M1064
RESET

Program execution procedure:

1. To perform tapping, spindle positioning is required. (Refer to Section 6.11 for more details)
2. **[Spindle is in the rigid tapping mode]** is on when the system is performing tapping. In the meantime, avoid setting [Spindle forward operation] and [Spindle positioning control] to off through RESET.
3. After RESET is triggered, the system stops executing the program, and the spindle stops rotating and starts positioning.
4. When **[Spindle rigid tapping interruption]** is set to on, the system automatically retracts the spindle when you set **[Spindle retraction after tapping]** to on by pressing the key for moving the Z axis forward.
5. When the retraction is complete, the system sets **[Spindle rigid tapping interruption]** to off.

Important:

1. Avoid the following operations during tapping.
   a. Switch the system mode.
   b. Set **[Spindle positioning control]** to off.
   c. Set **[Spindle retraction after tapping]** to on.
2. To disable the tapping interruption, set both [Spindle forward operation] and [Spindle positioning control] to off.
3. To re-execute the program or move any axes, set both [Spindle forward operation] and [Spindle positioning control] to off.
4. When the tapping is interrupted and [Spindle rigid tapping interruption] is set to on, the spindle and Z axis stop at the position where the interrupt occurs, and [Spindle positioning complete] remains on.
5. When the tapping is interrupted and [Spindle rigid tapping interruption] is set to on, the MPG operation is not applicable.
6. If the retraction after tapping is required when EMERGENCY STOP is triggered, set Pr307 [Bit 8 & 9] (EMG stop mode) to 2. In this case, the system remains Servo On when stopped with the emergency stop.

6.15 JOG retraction when power off

The auto retraction mentioned in Section 6.14 is the function of retracting the spindle and Z axis automatically when power is on. This section describes the retraction in JOG mode when power is off.

■ MLC special M relay

JOG retraction when power off	
Spindle retraction after tapping	M1125 (SP1), M1141 (SP2)
Z axis forward jog control	M1218

[Spindle retraction after tapping]: M1125, M1141

If the system is interrupted when performing tapping, set **[Spindle retraction after tapping]** to on to have the system retract the tap with the auto retraction function.

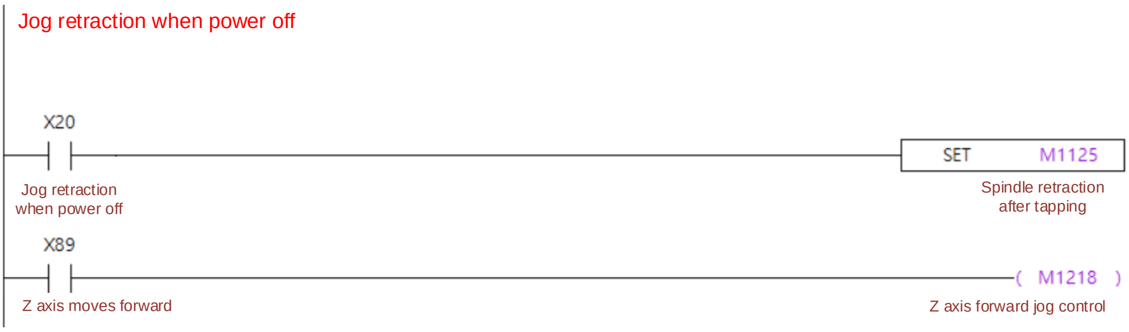
✓ Do not set **[Spindle retraction after tapping]** to on when the system is performing tapping.

[Z axis forward jog control]: M1218

When the system is in JOG mode, setting **[Z axis forward jog control]** to on moves Z axis in the forward direction. When it is set to off, Z axis stops.

■ MLC example

The following illustrates the Z axis retraction with jog operation when power is off.


Jog retraction when power off
X20
SET M1125
Jog retraction when power off
Spindle retraction after tapping
X89
(M1218)
Z axis moves forward
Z axis forward jog control

Program execution procedure:

1. When you turn off the power during tapping, the spindle and Z axis stop at the position where power is off.
2. When power is on, set [Spindle retraction after tapping] to on by triggering X20.
3. When [Z axis forward jog control] is triggered in JOG mode, the system retracts the spindle and Z axis simultaneously.

Important:

1. The system automatically records the thread pitch in the variable #10500 during tapping, so do not write to #10500 during tapping to avoid overwriting.
2. To trigger **[Spindle retraction after tapping]** in JOG mode, you have to set **[Spindle forward operation]** to on at the same time.
3. When the function of jog retraction is enabled, Z axis can move only in the forward direction. The retraction speed can be adjusted with the JOG speed while the maximum speed is limited to the corresponding F value when the rotation speed is 300.

6.16 One-key macro call

The one-key macro call function enables you to have the system call the specific macro with the MLC by triggering the signals. The MLC determines the conditions and then switches the macros for execution.

■ MLC special M relay and special D register

Macro call initial preparation	M1074	Macro call initial preparation complete	M2224
Macro call activation	M1075	Macro call in execution	M2225
Macro call error	M2226	Macro call file name	D1111

[Macro call initial preparation]: M1074

When **[Macro call initial preparation]** is set to on, the system prepares for the macro call by accessing the name to be called from [Macro call file name].

- ✓ This special M relay takes effect only in AUTO mode when it is set to on.
- ✓ The initialization takes effect only when there is corresponding macro file in the O_MACRO folder, CF card, or internal memory. Otherwise, the system generates an alarm to remind you of failing to open the file.

[Macro call initial preparation complete]: M2224

When [Macro call initial preparation] is set to on, the system prepares for the macro call and sets **[Macro call initial preparation complete]** to on when the initialization is complete.

- ✓ When **[Macro call initial preparation complete]** is set to on, the system executes the macro that is called instead of the current main file even if M1061 [Cycle Start] is triggered.
- ✓ When **[Macro call initial preparation complete]** is set to on, it is set to off once RESET is triggered.

[Macro call activation]: M1075

When [Macro call initial preparation complete] is set to on, setting **[Macro call activation]** to on triggers the system to execute the macro specified in [Macro call file name].

- ✓ This special M relay takes effect only in AUTO mode when it is set to on.
- ✓ This special M relay takes effect only when [Macro call initial preparation complete] is set to on.

[Macro call in execution]: M2225

When [Macro call initial preparation complete] is set to on, set [Macro call activation] to on and execute the macro, and the system sets **[Macro call in execution]** to on.

[Macro call error]: M2226

When you set [Macro call initial preparation] to on in non-AUTO mode or the value in [Macro call file name] exceeds 10000, the system sets **[Macro call error]** to on.

[Macro call file name]: D1111

When [Macro call initial preparation] is set to on, the system sets the macro to be executed referring to the value set in **[Macro call file name]**.

- ✓ Setting range: 1 - 9999.
- ✓ The macro O0001 is called when D1111 is set to 1 and the macro O1234 is called when D1111 is set to 1234. The same is true for others.
- ✓ When the file name ranges from 1 - 8999, the systems determines the source of the macro based on the parameter setting.
  - a. When Pr10017 [Subprogram call file source] is 0, the system calls the macro in the main file directory before the one-key macro call function is used. If there is no corresponding macro file in the directory, the system generates an alarm to remind you of failing to open the file.
  - b. When Pr10017 [Subprogram call file source] is 1, the system calls the macro in the USB root directory. If there is no corresponding macro file in the USB root directory, the system calls the macro in the main file directory before the one-key macro call function is used. If there is no corresponding macro file in the main file directory, the system generates an alarm to remind you of failing to open the file.
- ✓ When the file name ranges from 9000 - 9999, the system calls the macro referring to the setting of Pr10017 [Macro call file source].
  - a. When Pr10017 [Macro call file source] is 0, the system calls the macro in the O_MACRO folder in the CF card.
  - b. When Pr10017 [Macro call file source] is 1, the system calls the macro in the O_MACRO folder in the internal memory.
- ✓ When you set D1111 to 10000 or above, executing [Macro call initial preparation] triggers the system to set [Macro call error] to on, and the system cannot execute a macro.

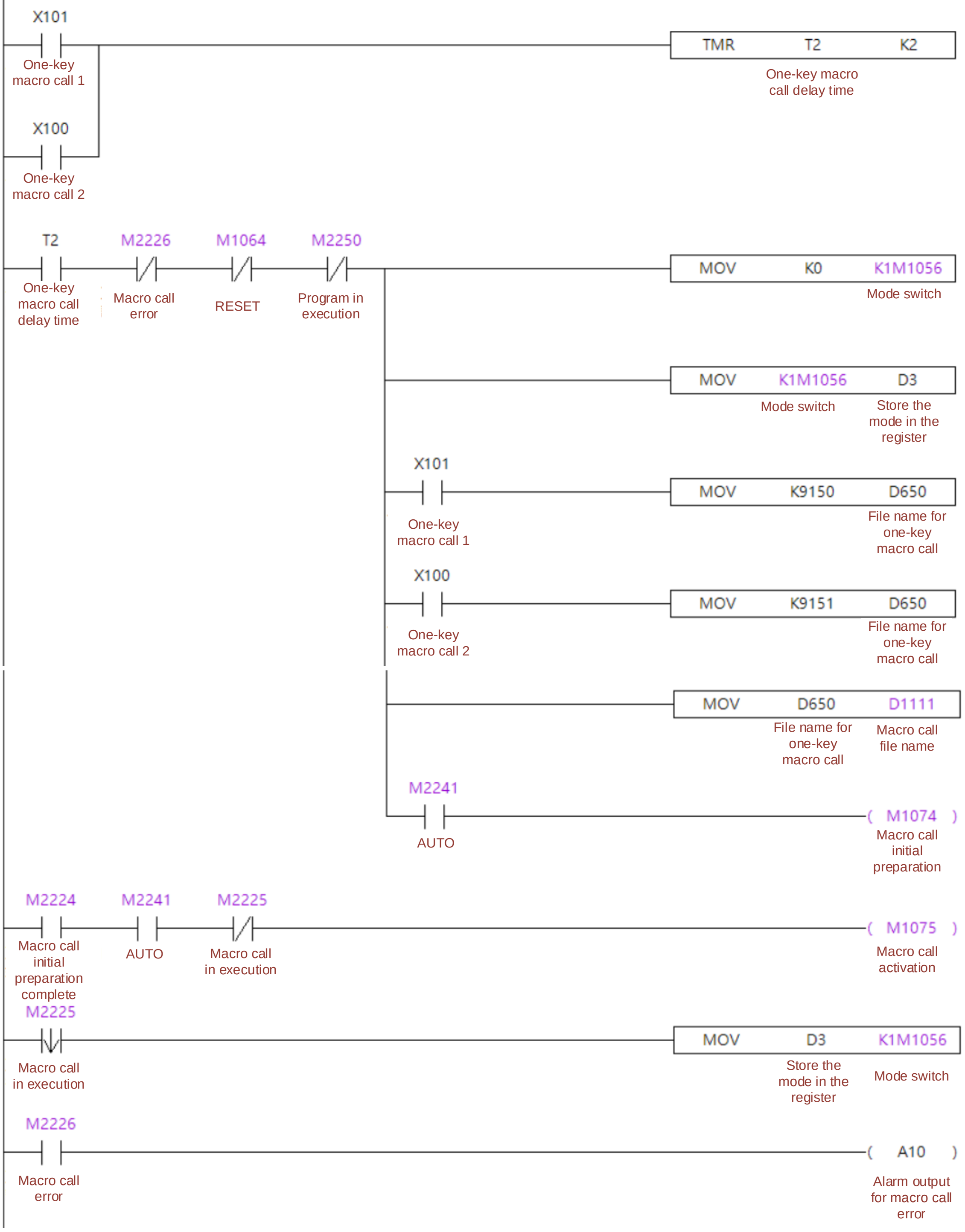
■ **Relevant parameters**

Setting macro call file source:

1. When the macro file name to be called ranges from 9000 - 9999, setting Pr10017 [Macro call file source] can determine the file source. When the setting value is 0, the system calls the macro file in the CF card. When the setting value is 1, the system calls the macro file in the internal memory.
2. When the macro file name to be called ranges from 1 - 8999, setting Pr10017 [Subprogram call file source] can determine the file source. When the setting value is 0, the system calls the subprogram in the main file directory. When the setting value is 1, the system calls the subprogram in the USB root directory. If there is no corresponding subprogram in the USB root directory, the system searches for the subprogram in the main file directory.

MLC example

One-key macro call



Program execution procedure:

1. Call two different macros with the X100 and X101 signals.
2. When X100 is on and then the signal is filtered for 0.2 sec, the system switches to AUTO mode, records the operation mode before switched, assigns a value to D650, assigns a value to D1111 through D650, and sets **[Macro call initial preparation]** to on in AUTO mode.
3. When the system sets **[Macro call initial preparation complete]** to on, if the system is in AUTO mode and does not execute the macro call, **[Macro call activation]** is set to on.
4. After the last macro is executed and the system sets **[Macro call in execution]** to off, the system switches back to the operation mode recorded in Step 2.

Important:

It is recommended not to use the one-key macro call function when the program is executed in AUTO or MDI mode since this function initializes some machining statuses, causing inconsistent macro execution. To call macros with external triggers when the program is executed in AUTO or MDI mode, refer to Section 6.8 M96 program interruption.

6.17 Tool magazine control with I/O

The controller changes tools with an external tool exchanger and transmits data or status with I/O. The following describes how to control the tool magazine with I/O.

■ **MLC special M relay and special D register**

Tool magazine control with I/O			
Tool magazine 1		Tool magazine 2	
Tool magazine 1 moves forward	M1168	Tool magazine 2 moves forward	M1172
Tool magazine 1 moves backward	M1169	Tool magazine 2 moves backward	M1173
Tool 1 exchange	M1170	Tool 2 exchange	M1174
Tool magazine 1 reset	M1171	Tool magazine 2 reset	M1175
Tool magazine 1 reset complete	M2212	Tool magazine 2 reset complete	M2213
Tool magazine 1 T code data (standby)	D1371	Tool magazine 2 T code data (standby)	D1375
Tool magazine 1 tool pot deviation	D1372	Tool magazine 2 tool pot deviation	D1376
Tool magazine 1 tool pot number (standby)	D1373	Tool magazine 2 tool pot number (standby)	D1377
Tool magazine 1 spindle tool number (in use)	D1374	Tool magazine 2 spindle tool number (in use)	D1378

[Tool magazine moves forward]: M1168, M1172

When **[Tool magazine moves forward]** is set to on, the standby tool pot number and standby tool number increase by 1. When both the standby tool pot number and standby tool number are the maximum numbers, set [Tool magazine moves forward] to on again, and the standby tool pot number and standby tool number become 1.

✓ The operation is the same for both tool magazine 1 and tool magazine 2.

[Tool magazine moves backward]: M1169, M1173

When **[Tool magazine moves backward]** is set to on, the standby tool pot number and standby tool number decrease by 1. When both the standby tool pot number and standby tool number are 1, set [Tool magazine moves backward] to on again, and the standby tool pot number and standby tool number become the maximum numbers.

✓ The operation is the same for both tool magazine 1 and tool magazine 2.

[Tool exchange]: M1170, M1174

When **[Tool exchange]** is set to on, the system exchanges the spindle tool number with the standby tool number.

✓ The operation is the same for both tool magazine 1 and tool magazine 2.

[Tool magazine reset]: M1171, M1175

To reset the tool magazine, in addition to using the tool setting function in the OFS screen, you can set **[Tool magazine reset]** to on for the system to reset the tool numbers and arrange the tools in ascending order based on the tool pot sequence.

- ✓ After the tool magazine is reset, the spindle tool number refers to Pr336 [Bit 10] (Spindle tool No. after magazine reset). When Pr336 [Bit 10] is 0, the spindle tool number is 0. When Pr336 [Bit 10] is 1, the spindle tool number refers to the settings of Pr338, Pr340, Pr341, and Pr343, and is the maximum tool number increases by 1 (where tool numbers are arranged in ascending order).

  Take tool magazine 1 for example. When Pr338 is 16 and Pr340 is set to 3, the spindle tool number after magazine reset is 19, the tool number in tool pot 1 is 3, and the tool number in tool pot 2 is 4.
- ✓ The system sets the standby tool pot after magazine reset to the default standby tool pot by referring to the settings of Pr339 and Pr342.
- ✓ After magazine reset, the system refers to the settings of Pr340 and Pr343 as the command tool numbers, and the standby tool numbers are the command tool numbers increase by 1. The tool pots are arranged in ascending order starting from tool pot 1 according to the tool numbers set in the tool magazine.
- ✓ The operation is the same for both tool magazine 1 and tool magazine 2.

[Tool magazine reset complete]: M2212, M2213

When you set [Tool magazine reset] to on in AUTO or MDI mode, **[Tool magazine reset complete]** is set to on after the system completes resetting the tool magazine.

- ✓ These special M relays are available only in AUTO or MDI mode.
- ✓ These special M relays are set to off when the RESET button is pressed after magazine reset.
- ✓ The operation is the same for both tool magazine 1 and tool magazine 2.

[T code data (standby)]: D1371, D1375

When reading the T code in the program during execution in AUTO or MDI mode, the system not only executes the T code procedure but writes the T code data to the **[T coda data (standby)]** register of the corresponding tool magazine according to the parameter setting.

- ✓ The system writes the T code of the corresponding tool magazine to the corresponding **[T coda data (standby)]** register referring to the settings of Pr338 and Pr341.
- ✓ The operation is the same for both tool magazine 1 and tool magazine 2.

- **Relevant parameters**

Tool magazine on / off:

- ✓ Pr337 (Tool magazine switch) sets the tool magazine to on or off. When Pr337 [Bit 0] (Tool magazine 1) is set to 1, tool magazine 1 is on. When Pr337 [Bit 0] is set to 0, tool magazine 1 is off. The same is true for Pr337 [Bit 1] (Tool magazine 2).
- ✓ If Pr337 [Bit 0 & 1] are not set to 1 when the T code is executed in the program, the system generates an alarm.

Total tool number:

- ✓ The total tool number of the tool magazine refers to the settings in Pr338 (Tool magazine 1 total tool No.) and Pr341 (Tool magazine 2 total tool No.).

Tool pot number after magazine reset:

- ✓ After magazine reset, the system sets the default standby tool pot number by referring to the settings of Pr339 (Tool magazine 1 standby tool No. after reset) and Pr342 (Tool magazine 2 standby tool No. after reset).

Start tool number:

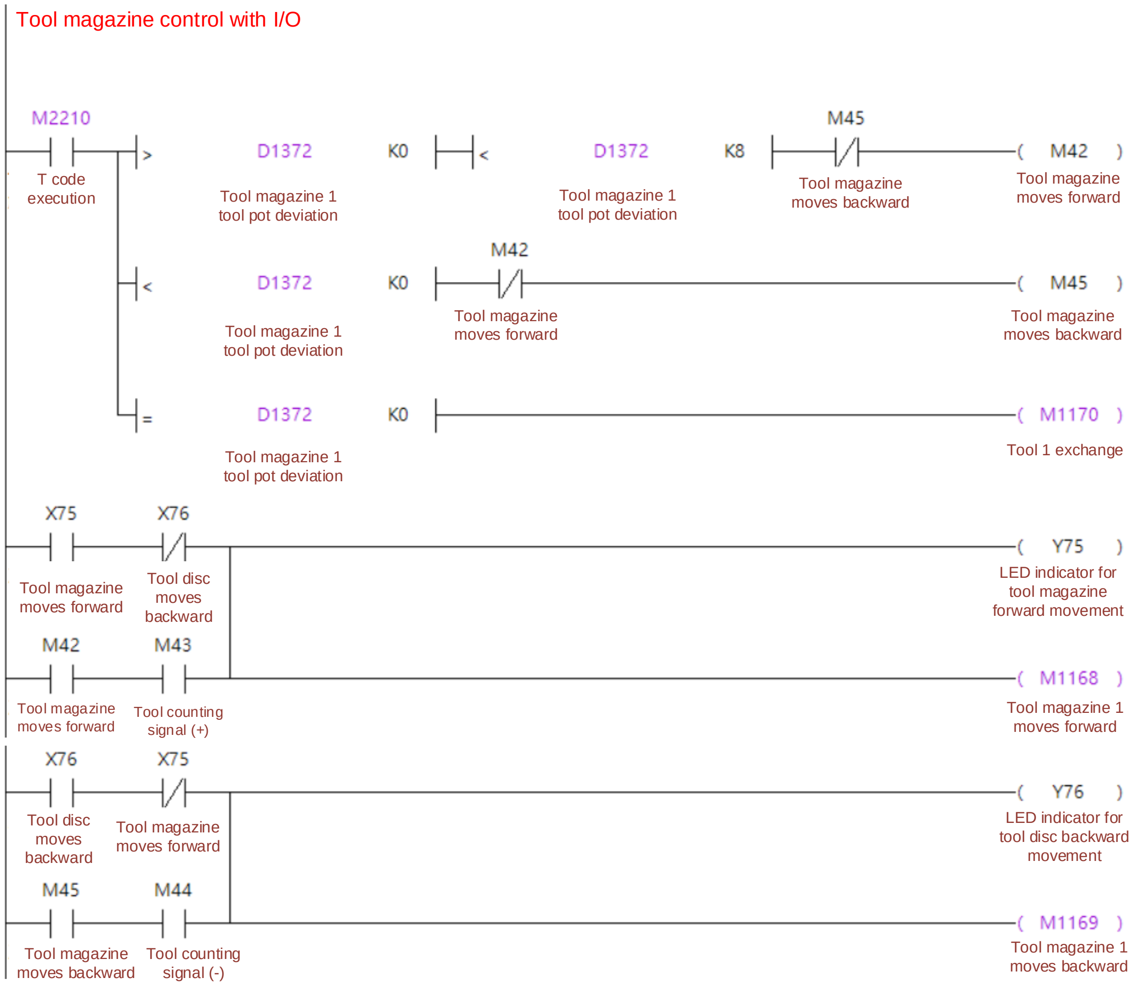
- ✓ After magazine reset, the system sets the command tool numbers by referring to the settings of Pr340 (Tool magazine 1 start tool No.) and Pr343 (Tool magazine 2 start tool No.). The standby tool numbers are the command tool numbers increase by 1. The tool pots are arranged in ascending order starting from tool pot 1 according to the tool numbers set in the magazine.

Spindle tool number after magazine reset:

- ✓ After you reset the tool magazine, the spindle tool number is displayed referring to the setting of Pr336 [Bit 10] (Spindle tool No. after magazine reset). When Pr336 [Bit 10] is 0, the spindle tool number after magazine reset is 0. When Pr336 [Bit 10] is 1, the spindle tool number refers to the settings of total tool number and start tool number, and is the maximum tool number increases by 1 (where tool numbers are arranged in ascending order).

■ MLC example

The following example illustrates the program execution with tool magazine 1 set to hold 16 tools.


Tool magazine control with I/O
M2210
T code execution
>
D1372
Tool magazine 1 tool pot deviation
K0
<
D1372
Tool magazine 1 tool pot deviation
K8
M45
Tool magazine moves backward
M42
Tool magazine moves forward
<
D1372
Tool magazine 1 tool pot deviation
K0
M42
Tool magazine moves forward
M45
Tool magazine moves backward
=
D1372
Tool magazine 1 tool pot deviation
K0
M1170
Tool 1 exchange
X75
Tool magazine moves forward
X76
Tool disc moves backward
Y75
LED indicator for tool magazine forward movement
M42
Tool magazine moves forward
M43
Tool counting signal (+)
M1168
Tool magazine 1 moves forward
X76
Tool disc moves backward
X75
Tool magazine moves forward
Y76
LED indicator for tool disc backward movement
M45
Tool magazine moves backward
M44
Tool counting signal (-)
M1169
Tool magazine 1 moves backward

Program execution procedure:

1. When reading the T code in the program, the system sets [T code execution] to on and automatically writes the difference between the current standby tool pot and command tool pot to **[Tool magazine 1 tool pot deviation]**.
2. Since the deviation is displayed in both positive and negative values, when there are only 16 tools in the tool magazine, the maximum deviation is ±8.
3. The system determines in which direction and how many positions the tool magazine has to move by performing logic statement. Then, it sets [Tool magazine 1 moves forward] or [Tool magazine 1 moves backward] to on by referring to the corresponding signals such as tool magazine moves forward and external tool counting.
4. When the value of **[Tool magazine 1 tool pot deviation]** is 0, the tool data is exchanged. Besides, the tools of an external mechanical part can be changed with the MLC.

6.18 MLC axis control

You can dynamically switch a specific axis to the NC axis mode or MLC axis mode with the following special M relays. In MLC axis mode, you can perform position control, speed control, and applications requiring rotations similar to the spindle rotation or positioning control.

■ MLC special M relay and special D register

MLC axis control and status:

	Activate the axis	Switch between absolute / incremental command	Switch between MLC position / speed mode	Positioning command	Speed command	Position / speed reached	The axis is moving
X axis	M1184	M1280	M1289	D1064	D1082	M2304	M2320
Y axis	M1185	M1281	M1290	D1066	D1084	M2305	M2321
Z axis	M1186	M1282	M1291	D1068	D1086	M2306	M2322
A axis	M1187	M1283	M1292	D1070	D1088	M2307	M2323
B axis	M1188	M1284	M1293	D1072	D1090	M2308	M2324
C axis	M1189	M1285	M1294	D1074	D1092	M2309	M2325
U axis	M1190	M1286	M1295	D1076	D1094	M2310	M2326
V axis	M1191	M1287	M1296	D1078	D1096	M2311	M2327
W axis	M1192	M1288	M1297	D1080	D1098	M2312	M2328
All	-	M1194	-	-	-	-	-

NC / MLC mode switch and status display:

	Switch between NC and MLC axis mode	NC / MLC axis status
X axis	M1200	M2354
Y axis	M1201	M2355
Z axis	M1202	M2356
A axis	M1203	M2357
B axis	M1204	M2358
C axis	M1205	M2359
U axis	M1206	M2360
V axis	M1207	M2361
W axis	M1208	M2362

[Activate the axis]: M1184 - M1192

When an axis is in MLC axis mode, setting the special M relay [Activate the axis] to on enables the axis to execute the motion command. Setting [Activate the axis] to off stops the axis.

- ✓ Prior to triggering the special M relay [Activate the axis] for axis activation, you have to set [Positioning command] and [Speed command] in advance for at least one MLC scan time.
- ✓ The updated value of the positioning command special D takes effect when the rising edge of [Activate the axis] is triggered the next time.
- ✓ The updated value of the speed command special D takes effect once the MLC scans the change. There is no need to re-trigger [Activate the axis].

[Switch between absolute / incremental command]: M1280 - M1288, M1194

In MLC position mode, when the special M relay [Switch between absolute / incremental command] is off, the axis receives an absolute command and moves to the machine coordinate specified by [Positioning command]. When [Switch between absolute / incremental command] is on, the axis receives an incremental command and moves the distance specified by [Positioning command].

- ✓ Note that when M1194 [MLC axis incremental motion command] is set to on, all MLC axes are in incremental mode. When M1194 is set to off, each axis refers to the settings of [Switch between absolute / incremental command] respectively.

[Switch between MLC position / speed mode]: M1289 - M1297

When the special M relay [Switch between MLC position / speed mode] is off, the MLC is in position mode and moves the axis to the specified position by referring to [Positioning command] and [Speed command]. When [Switch between MLC position / speed mode] is on, the MLC is in speed mode and rotates the axis at a constant speed by referring to [Speed command].

[Positioning command]: D1064, D1066, D1068, D1070, D1072, D1074, D1076, D1078, D1080

In MLC position mode, the mode can be further divided into absolute and incremental modes. In absolute mode, [Positioning command] specifies the target machine coordinate. In incremental mode, [Positioning command] specifies the moving amount with incremental values.

- ✓ Note that this special D register is a floating-point number which takes up two consecutive register addresses.
- ✓ The axis does not refer to [Positioning command] in MLC speed mode.

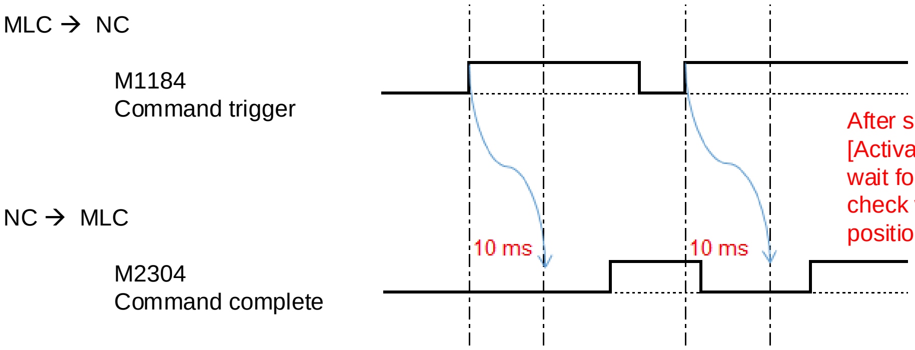
[Speed command]: D1082, D1084, D1086, D1088, D1090, D1092, D1094, D1096, D1098

When in MLC axis mode, X, Y, and Z axes are linear axes with the speed unit of mm/min. A, B, C, U, V, and W axes can be set to linear axes or rotation axes by referring to Pr634 [Bit 1 - 3] (Rotation axis feed mode). The speed unit of the rotation axis refers to the setting of Pr634 [Bit 7] (Rotation axis unit), which can be either rpm or deg/min.

- ✓ When in MLC position mode, the axis refers to the absolute value of the speed command for the speed and moves to the position specified by the positioning command.
- ✓ When in MLC speed mode, the axis refers to the speed command special D register for its rotation direction and speed.
- ✓ Note that this special D register is a floating-point number which takes up two consecutive register addresses.
- ✓ When the value of the speed command special D register is updated during machining, the axis changes its moving speed once the MLC scans the change.

[Position / speed reached]: M2304 - M2312

- ✓ When the MLC axis is in position mode, [Position / speed reached] indicates whether the position is reached. If the position is reached, this special M relay is on. If the position is not reached, it is off.
- ✓ When the MLC axis is in speed mode, [Position / speed reached] indicates whether the speed is reached. If the speed is reached, this special M relay is on. If the speed is not reached, it is off.
- ✓ Pay special attention when programming the MLC ladder diagram by referring to the following timing diagram. After [Activate the axis] is set to on, you need to wait for 10 ms and then check for [Position / speed reached] to avoid misoperation in case the status feedback is not updated timely.



After setting the special M [Activate the axis] to on, wait for 10 ms and then check whether the position is reached.

[The axis is moving]: M2320 - M2328

Refer to the status of the special M relay [The axis is moving] to determine whether the specific axis is moving.

[Switch between NC and MLC axis mode]: M1200 - M1208

These special M relays dynamically switch the system between NC and MLC axis modes. When [Switch between NC and MLC axis mode] is on, the system switches to MLC axis mode. When [Switch between NC and MLC axis mode] is off, the system switches to NC axis mode.

In the CONFIG (Channel setting) screen, you have to set the axis as an NC axis.

- ✓ To switch the axis to NC or MLC axis mode, you have to use the halt M code in AUTO or MDI mode.
- ✓ You have to switch the axis to NC or MLC axis mode when the axis is not in motion, or alarm B637 occurs.

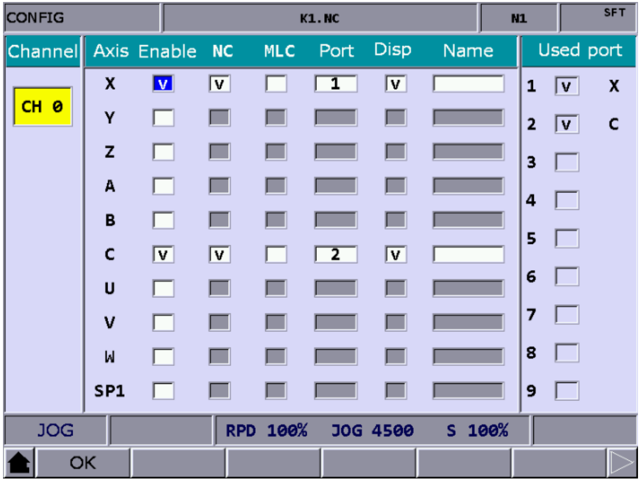
[NC / MLC axis status]: M2354 - M2328

When the system is in the NC axis mode, [NC / MLC axis status] is off. When the system is in the MLC axis mode, [NC / MLC axis status] is on.

■ Relevant parameters

Channel setting:

When an axis is set as an NC axis, it can be dynamically switched between NC and MLC axes. When the axis is set as an MLC axis, it cannot be dynamically switched.



Axis	Enable	NC	MLC	Port	Disp	Name
X	☑	☑	□	1	☑	
Y	□					
Z	□					
A	□					
B	□					
C	☑	☑	□	2	☑	
U	□					
V	□					
W	□					
SP1	□					

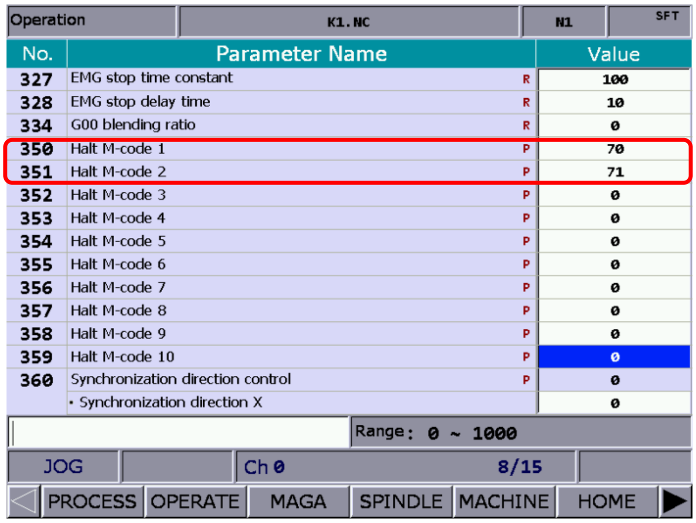
Used port		
1	☑	X
2	☑	C
3	□	
4	□	
5	□	
6	□	
7	□	
8	□	
9	□	

Speed parameters:

The MLC axis moving speed is affected by these parameters: Pr621 G00/manual maximum speed, Pr622 G00/manual ACC/DEC time, and Pr623 G00/manual S-curve time.

Operating parameters:

You have to use the halt M code parameters to switch the axis between NC and MLC axes with MLC program in AUTO mode.

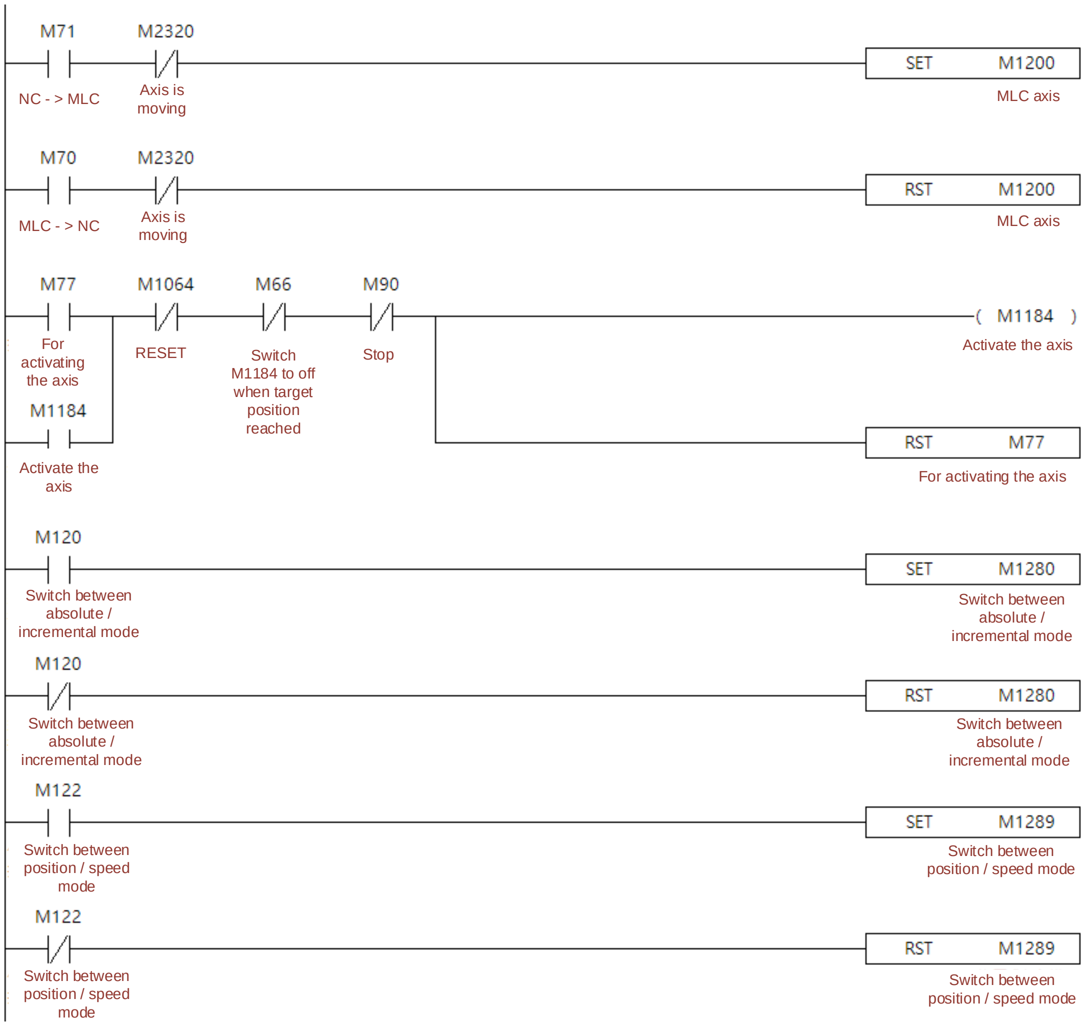


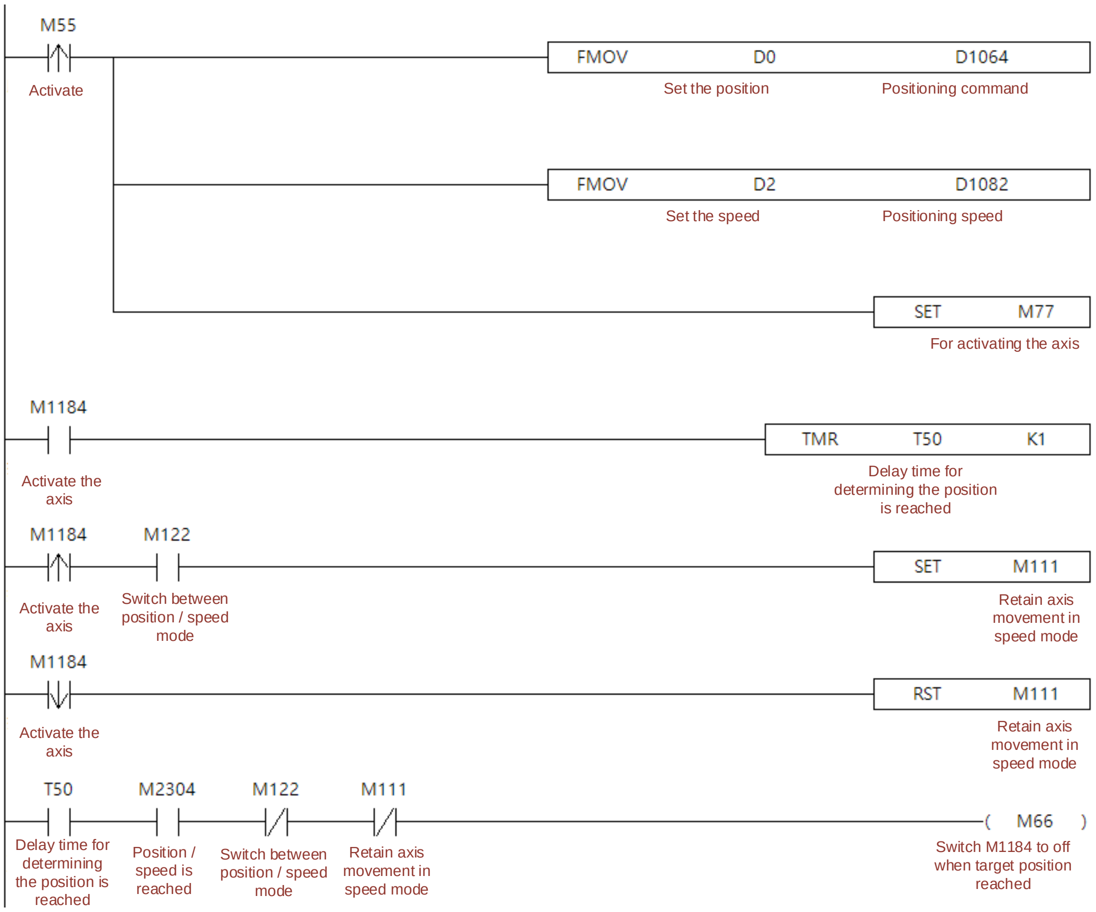
No.	Parameter Name		Value
327	EMG stop time constant	R	100
328	EMG stop delay time	R	10
334	G00 blending ratio	R	0
350	Halt M-code 1	P	70
351	Halt M-code 2	P	71
352	Halt M-code 3	P	0
353	Halt M-code 4	P	0
354	Halt M-code 5	P	0
355	Halt M-code 6	P	0
356	Halt M-code 7	P	0
357	Halt M-code 8	P	0
358	Halt M-code 9	P	0
359	Halt M-code 10	P	0
360	Synchronization direction control	P	0
	• Synchronization direction X		0

Range: 0 ~ 1000

- **MLC example**

The following takes X axis as an example. Switching the axis between NC and MLC axis modes is only available when the axis is not in motion.


M71
M2320
SET M1200
NC - > MLC
Axis is moving
MLC axis
M70
M2320
RST M1200
MLC - > NC
Axis is moving
MLC axis
M77
M1064
M66
M90
(M1184)
For activating the axis
RESET
Switch M1184 to off when target position reached
Stop
Activate the axis
M1184
Activate the axis
RST M77
For activating the axis
M120
SET M1280
Switch between absolute / incremental mode
Switch between absolute / incremental mode
M120
RST M1280
Switch between absolute / incremental mode
Switch between absolute / incremental mode
M122
SET M1289
Switch between position / speed mode
Switch between position / speed mode
M122
RST M1289
Switch between position / speed mode
Switch between position / speed mode


M55
Activate
FMOV D0 D1064
Set the position
Positioning command
FMOV D2 D1082
Set the speed
Positioning speed
SET M77
For activating the axis
M1184
Activate the axis
TMR T50 K1
Delay time for determining the position is reached
M1184
M122
Activate the axis
Switch between position / speed mode
SET M111
Retain axis movement in speed mode
M1184
Activate the axis
RST M111
Retain axis movement in speed mode
T50
M2304
M122
M111
M66
Delay time for determining the position is reached
Position / speed is reached
Switch between position / speed mode
Retain axis movement in speed mode
Switch M1184 to off when target position reached

Program execution procedure:

Position mode:

1. M71 Switches from NC axis to MLC axis. (This action is not required if the axis is set as an MLC axis in the channel setting screen.)
2. M120 Sets the system to absolute or incremental mode.
3. Set M122 to off and the system is in position mode.
4. Set D0 for the position value.
5. Set D2 for the speed value.
6. Trigger M55 to write the position and speed values and activate the axis.

Speed mode:

1. M71 Switches from NC axis to MLC axis. (This action is not required if the axis is set as an MLC axis in the channel setting screen.)
2. Set M122 to on and the system is in speed mode.
3. Set D2 for the speed value.
4. Trigger M55 to write the speed value and activate the axis.

Important:

1. The homing speed for the rotation axis refers to the settings of Pr618 and Pr619 (unit: rpm).
2. Pay attention to the execution timing of the program. Execute M1152 after M1200 - M1208 to ensure correct operation.
3. When a program is executed but the system does not perform any actions, check for the special M relays [NC / MLC axis status] (M2354 - M2362).
4. When a program is executed but the system does not perform any actions, check if the values in the positioning command special D registers (D1064 - D1080) and speed command special D registers (D1082 - D1098) are floating-point numbers.
5. The corresponding halt M code parameter is required in AUTO mode when the axis is switched from MLC to NC axis mode.
6. You need to re-trigger [Activate the axis] to have the updated value take effect except the speed command special D which takes immediate effect after being modified.

6.19 Synchronous control and command transfer

The system provides the functions of synchronous axis control and transferring command to another axis, which are enabled or disabled with the MLC. The following describes the two functions.

- **MLC special M relay**

Synchronous control and command transfer						
Axis	Trigger for synchronous control	Slave axis follows the master axis	Synchronous function in execution	Trigger for command transfer	Receive command from the master axis	Transfer function in execution
X axis	M1088	M1089	M2227	M1098	M1099	M2228
Y axis		M1090			M1100	
Z axis		M1091			M1101	
A axis		M1092			M1102	
B axis		M1093			M1103	
C axis		M1094			M1104	
U axis		M1095			M1105	
V axis		M1096			M1106	
W axis		M1097			M1107	

[Trigger for synchronous control]: M1088

To enable the synchronous control function, set **[Trigger for synchronous control]** to on and then set [Slave axis follows the master axis] to on for the corresponding axis, so the system can activate the synchronous control function.

[Slave axis follows the master axis]: M1089 - M1097

When [Trigger for synchronous control] is on, set **[Slave axis follows the master axis]** to on for the corresponding axis at the same time, so the system can activate the synchronous control function for the axis.

[Synchronous function in execution]: M2227

After activating the synchronous control function, the system sets **[Synchronous function in execution]** to on.

- ✓ To enable or disable the synchronous control function when the program is executed in AUTO or MDI mode, the system sets **[Synchronous function in execution]** to on or off in the following conditions.
  - a. When [M, S, and T codes execution complete] is set to on and the M code procedure is complete.
  - b. When [Trigger for synchronous control] and [Slave axis follows the master axis] are set to on or off through RESET.
- ✓ When [Trigger for synchronous control] and [Slave axis follows the master axis] are set to on or off in JOG or MPG mode, **[Synchronous function in execution]** is set to on or off immediately.
- ✓ When [Trigger for synchronous control] and [Slave axis follows the master axis] are set to on or off in HOME mode, **[Synchronous function in execution]** is set to on or off as soon as [Homing control] is set to on for the corresponding axis.

[Trigger for command transfer]: M1098

To enable the transfer function, set **[Trigger for command transfer]** to on and then set [Receive command from the master axis] to on for the corresponding axis, so the system can activate the transfer function.

[Receive command from the master axis]: M1099 - M1107

When [Trigger for command transfer] is on, set **[Receive command from the master axis]** to on for the corresponding axis at the same time, so the system can activate the transfer function for the axis.

[Transfer function in execution]: M2228

After enabling the transfer function, the system sets **[Transfer function in execution]** to on.

- ✓ To enable or disable the transfer function when the program is executed in AUTO or MDI mode, the system sets **[Transfer function in execution]** to on or off in the following conditions.
  a. When [M, S, and T codes execution complete] is set to on and the M code procedure is complete.
  b. When [Trigger for command transfer] and [Receive command from the master axis] are set to on or off through RESET.
- ✓ When [Trigger for command transfer] and [Receive command from the master axis] are set to on or off in JOG or MPG mode, **[Transfer function in execution]** is set to on or off immediately.
- ✓ If the system is switched to HOME mode when **[Transfer function in execution]** is on, **[Transfer function in execution]** is set to off.

■ **Relevant parameters**

Parameters for synchronization:

- ✓ Pr360 (Synchronization direction control): sets whether the motor moving directions are the same or different for the synchronous axes. When Pr360 is 0, the slave axis moves in the same direction as the master axis synchronously. When Pr360 is set to 1, the slave axis moves in the different direction from the master axis synchronously.
- ✓ Pr361 - 369 (Synchronous control): sets the master axis to be followed. 1 indicates X axis, 2 indicates Y axis, 3 indicates Z axis, and the same is true for others. If you set Pr364 to 3, A axis follows Z axis synchronously.
- ✓ Pr617 [Bit 4] (Homing option for sync. Motion): sets whether the slave axes and the master axis return to the origin synchronously or each slave axis returns to the origin individually during synchronous control.
- ✓ Pr642 (Synchronous tolerance): sets the tolerance between the master and slaves during synchronous control. When the error exceeds the setting value of this parameter, the system generates alarm B645 (Excessive synchronous following error).
- ✓ Pr10009 (Sync coordinate setting): sets whether to display the coordinate of the synchronous axis (Bit 0) or the workpiece coordinate of the synchronous axis (Bit 2). When Pr10009 [Bit 0] or [Bit 2] is 0, the coordinate is not displayed. When Pr10009 [Bit 0] or [Bit 2] is 1, the coordinate is displayed.

Parameters for command transfer:

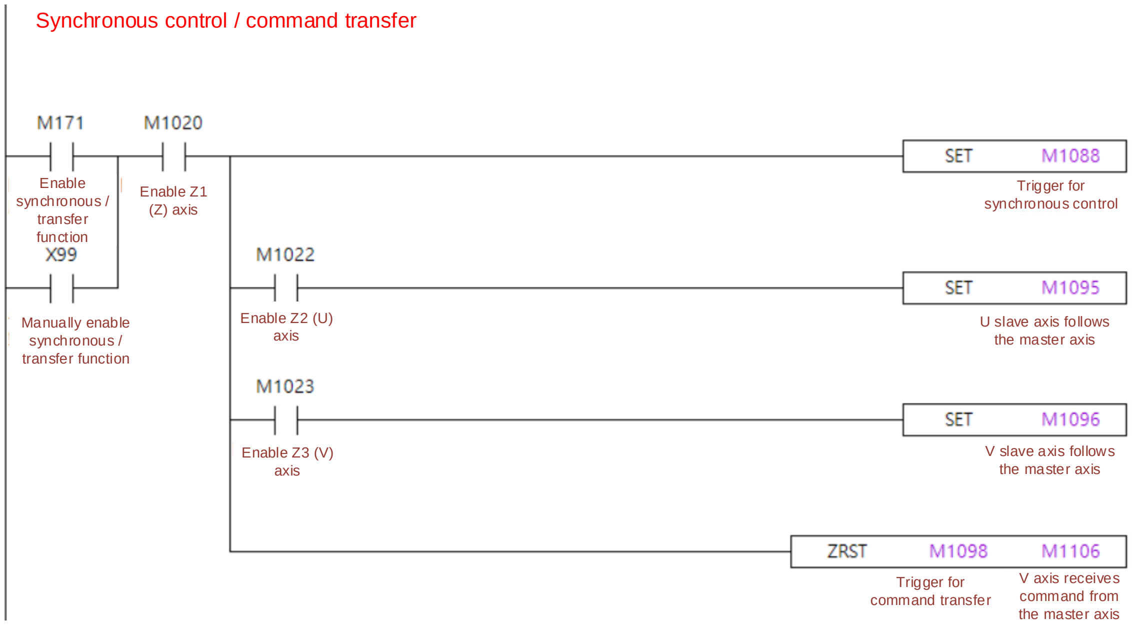
Pr371 - Pr379 (Transfer control): specifies the master axis to transfer the command. When the parameter is set to 1, X axis is specified as the master axis, 2 means Y axis, 3 means Z axis, and the same is true for others. If you set Pr374 to 3, Z axis is the master axis which transfers the command to A axis.

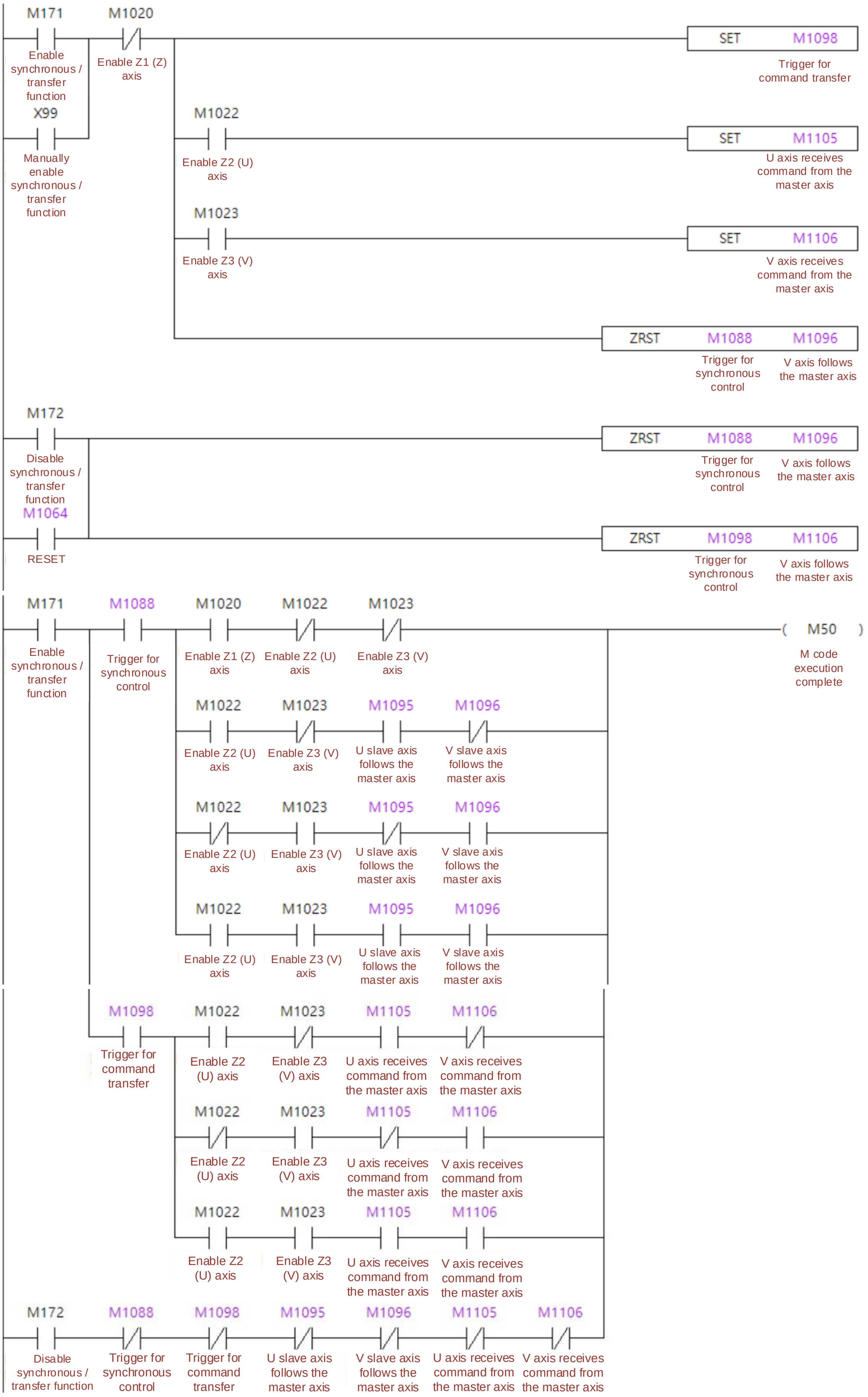
Halt M code:

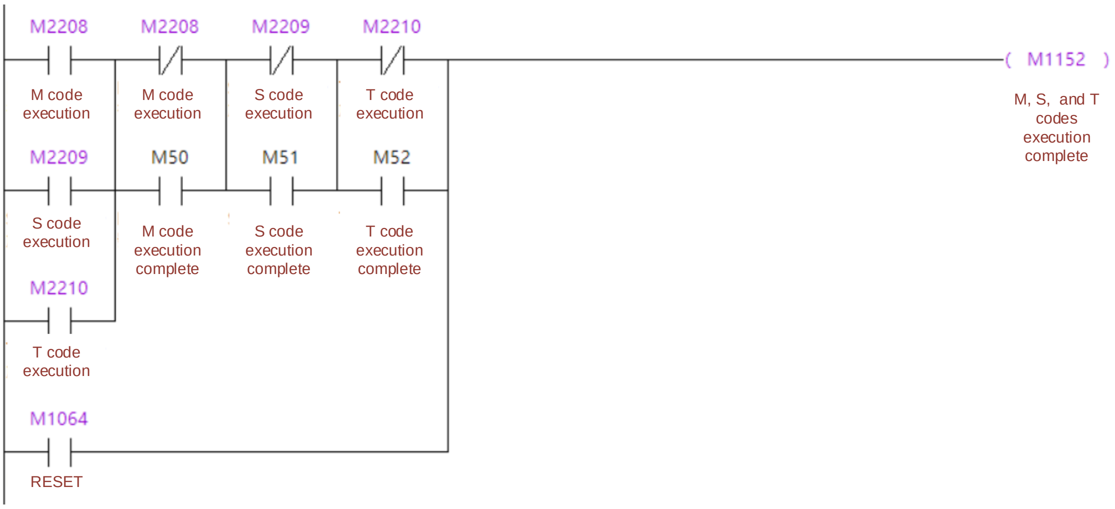
Enabling or disabling the function of synchronous control or command transfer in AUTO or MDI mode is done with M code. The required M code has to be set in the halt M code. Refer to Section 6.7 for detailed information.

■ MLC example

Z, U, and V axes are illustrated in the following example. M1020 enables Z axis, M1022 enables U axis, and M1023 enables V axis. Set the corresponding special M relay to on to activate the axis. In addition, the following illustrates how to enable and disable the synchronous control and command transfer functions with the MLC.


Synchronous control / command transfer
M171
Enable synchronous / transfer function
X99
Manually enable synchronous / transfer function
M1020
Enable Z1 (Z) axis
SET M1088
Trigger for synchronous control
M1022
Enable Z2 (U) axis
SET M1095
U slave axis follows the master axis
M1023
Enable Z3 (V) axis
SET M1096
V slave axis follows the master axis
ZRST M1098 M1106
Trigger for command transfer
V axis receives command from the master axis


M171
Enable synchronous / transfer function
X99
Manually enable synchronous / transfer function
M1020
Enable Z1 (Z) axis
SET M1098
Trigger for command transfer
M1022
Enable Z2 (U) axis
SET M1105
U axis receives command from the master axis
M1023
Enable Z3 (V) axis
SET M1106
V axis receives command from the master axis
ZRST M1088 M1096
Trigger for synchronous control
V axis follows the master axis
M172
Disable synchronous / transfer function
M1064
RESET
ZRST M1088 M1096
Trigger for synchronous control
V axis follows the master axis
ZRST M1098 M1106
Trigger for synchronous control
V axis follows the master axis
M171
Enable synchronous / transfer function
M1088
Trigger for synchronous control
M1020
Enable Z1 (Z) axis
M1022
Enable Z2 (U) axis
M1023
Enable Z3 (V) axis
M50
M code execution complete
M1022
Enable Z2 (U) axis
M1023
Enable Z3 (V) axis
M1095
U slave axis follows the master axis
M1096
V slave axis follows the master axis
M1022
Enable Z2 (U) axis
M1023
Enable Z3 (V) axis
M1095
U slave axis follows the master axis
M1096
V slave axis follows the master axis
M1022
Enable Z2 (U) axis
M1023
Enable Z3 (V) axis
M1095
U slave axis follows the master axis
M1096
V slave axis follows the master axis
M1098
Trigger for command transfer
M1022
Enable Z2 (U) axis
M1023
Enable Z3 (V) axis
M1105
U axis receives command from the master axis
M1106
V axis receives command from the master axis
M1022
Enable Z2 (U) axis
M1023
Enable Z3 (V) axis
M1105
U axis receives command from the master axis
M1106
V axis receives command from the master axis
M1022
Enable Z2 (U) axis
M1023
Enable Z3 (V) axis
M1105
U axis receives command from the master axis
M1106
V axis receives command from the master axis
M172
Disable synchronous / transfer function
M1088
Trigger for synchronous control
M1098
Trigger for command transfer
M1095
U slave axis follows the master axis
M1096
V slave axis follows the master axis
M1105
U axis receives command from the master axis
M1106
V axis receives command from the master axis



Program execution procedure:

1. When X99 is on in JOG or MPG mode, the system determines to execute the synchronous control or command transfer function depending on whether Z axis is enabled.
2. When the synchronous control function is enabled, ensure to disable the command transfer function to avoid misoperation, and vice versa.
3. The synchronous control or command transfer function can be disabled through RESET.
4. When the program reads M171 in AUTO or MDI mode, the system determines to execute the synchronous control or command transfer function depending on whether Z axis is enabled.
5. When the synchronous control function is enabled, ensure to disable the command transfer function to avoid misoperation, and vice versa.
6. After the function is enabled, the M code procedure is complete.
7. When the program reads M172, the synchronous control or command transfer function is disabled.

Important:

1. When the synchronous control or command transfer function is executed in AUTO or MDI mode, **[Synchronous function in execution]** or **[Transfer function in execution]** is set to on only when the function is enabled and the M code procedure is complete. Accordingly, you cannot determine whether the M, S, and T codes execution is complete by referring to the status of [Synchronous function in execution] or [Transfer function in execution].
2. You need to set the halt M code function for the M codes used for function enabling and disabling in AUTO and MDI modes.
3. The system checks whether to enable the synchronous control or command transfer function at different time points in different modes, which are described as follows.
   a. AUTO or MDI mode: when the M code procedure is complete or after RESET is triggered.
   b. JOG or MPG mode: at all times.
   c. HOME mode:

      Synchronous control: when [Trigger for synchronous control] and [Slave axis follows the master axis] are set to on or off, **[Synchronous function in execution]** is set to on or off as soon as [Homing control] is set to on for the corresponding axis.

      Transfer: when the system is switched to HOME mode with **[Command transfer in**

execution] set to on, **[Command transfer in execution]** becomes off. When **[Command transfer in execution]** is off, the system neither checks the statuses of **[Trigger for command transfer]** and **[Receive command from the master axis]** nor enables or disables the command transfer function. However, the system operates according to the mode you switch to.

d. EDIT: the system does not check for the enabling of the function and operates according to the mode you switch to.

4. One axis can only be the master axis or slave axis at the same time.
5. Multiple slave axes can follow the same master axis for synchronous control or command transfer at the same time.
6. When the command transfer function is enabled, if the program reads the movement command for the slave axis, the slave axis does not move and the movement command is skipped.
7. The command transfer function supports the cutting cycle command for Z axis.
8. When Pr616 (Origin search mode) is 0 - 5 and the synchronous control function is enabled, if Pr617 [Bit 4] (Homing option for sync. Motion) is 0, the slave axes and the master axis perform the homing procedure synchronously.
9. When the synchronous control function is enabled, the program cannot execute the command for the slave axis and displays alarm B015 (Sync cmd error).
10. The settings of the machine parameters and homing modes for the synchronous axes should be consistent.
11. When A, B, C, U, V, and W axes are the slave axes and X, Y, and Z axes are the master axes, the setting of Pr634 [Bit 1 - 3] (Rotation axis feed mode) for A, B, C, U, V, and W axes has to be 5. When A, B, C, U, V, and W axes are the master axes, the setting of Pr634 [Bit 1 - 3] for these axes has to be consistent.
12. When the program executes G28, G29, G30, G52, or G53 with the command transfer function enabled, the system automatically disables the command transfer function and performs the corresponding action of the G code, and then enables the command transfer function after the G code is complete. If RESET is triggered during G code execution, the system stops and operates according to the current mode.

6.20 Synchronous gantry control

You can use the synchronous gantry control function with M code in AUTO and MDI modes, directly enable or disable the function in JOG and MPG modes, or execute the function at startup. The following describes the synchronous gantry control function.

■ MLC special M relay

Axis	Trigger for synchronous control	Slave axis follows the master axis	Synchronous function in execution
X axis	M1088	M1089	M2227
Y axis		M1090	
Z axis		M1091	
A axis		M1092	
B axis		M1093	
C axis		M1094	
U axis		M1095	
V axis		M1096	
W axis		M1097	

[Trigger for synchronous control]: M1088

To enable the synchronous control function, set **[Trigger for synchronous control]** to on and then set [Slave axis follows the master axis] to on for the corresponding axis, so the system can activate the synchronous control function.

[Slave axis follows the master axis]: M1089 - M1097

When [Trigger for synchronous control] is on, set **[Slave axis follows the master axis]** to on for the corresponding axis, so the system can activate the synchronous control function for the axis.

[Synchronous function in execution]: M2227

After enabling the synchronous control function, the system sets **[Synchronous function in execution]** to on.

- ✓ When the program is executed in AUTO or MDI mode, to enable or disable the synchronous control function, **[Synchronous function in execution]** is set to on or off in the following conditions.
    a. When [M, S, and T codes execution complete] is set to on and the M code procedure is complete.
    b. When [Trigger for synchronous control] and [Slave axis follows the master axis] are set to on or off through RESET.
- ✓ When [Trigger for synchronous control] and [Slave axis follows the master axis] are set to on or off in JOG or MPG mode, **[Synchronous function in execution]** is set to on or off immediately.
- ✓ When [Trigger for synchronous control] and [Slave axis follows the master axis] are set to on or off in HOME mode, **[Synchronous function in execution]** is set to on or off as soon as [Homing control] is set to on for the corresponding axis.

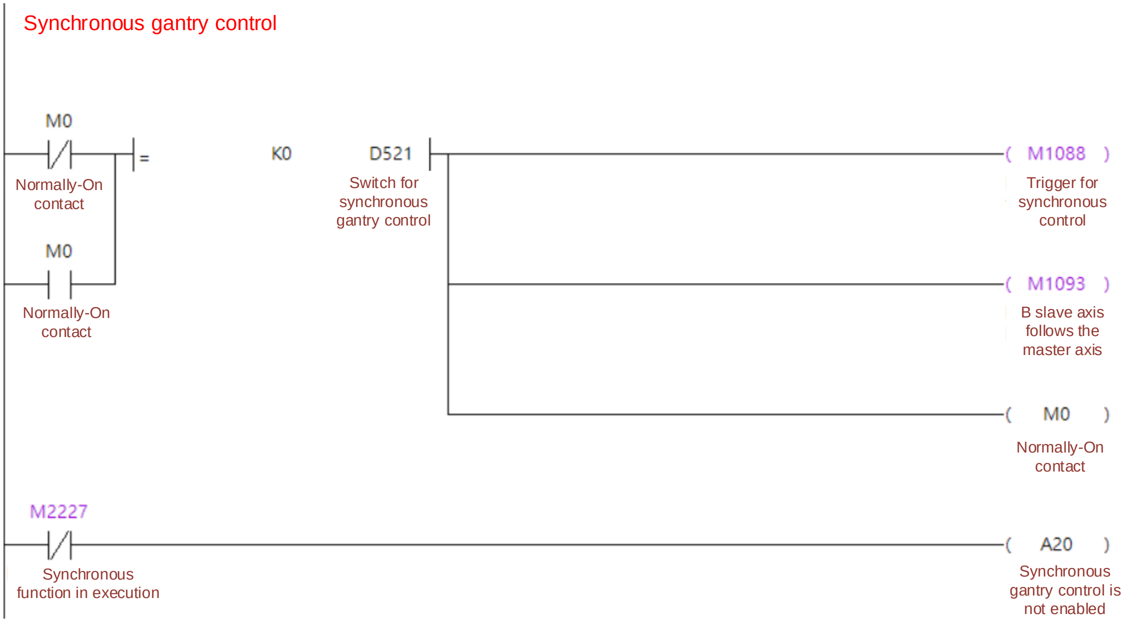
■ **Relevant parameters**

Parameters for synchronization:

- ✓ Pr360 (Synchronization direction control): sets whether the motor moving directions are the same or different for the synchronous axes. When Pr360 is 0, the slave axis moves in the same direction as the master axis synchronously. When Pr360 is set to 1, the slave axis moves in the different direction from the master axis synchronously.
- ✓ Pr361 - Pr369 (Synchronous control): sets the master axis to be followed. When the parameter is set to 1, X axis is the master axis. 2 indicates Y axis, 3 indicates Z axis, and the same is true for others. If you set Pr364 to 3, the slave A axis follows Z axis synchronously.
- ✓ Pr617 [Bit 4] (Homing option for sync. Motion): sets whether the slave axes and the master axis return to the origin synchronously or each slave axis returns to the origin individually during synchronous control.
- ✓ Pr642 (Synchronous tolerance): sets the tolerance between the master and slaves during synchronous control. When the error exceeds the setting value of this parameter, the system generates alarm B645 (Excessive synchronous following error).
- ✓ Pr10009 (Sync coordinate setting): sets whether to display the coordinate of the synchronous axis (Bit 0) or the workpiece coordinate of the synchronous axis (Bit 2). When Pr10009 [Bit 0] or [Bit 2] is set to 0, the coordinate is not displayed. When set to 1, the coordinate is displayed.

■ MLC example

The following example illustrates the synchronous gantry control between Y axis and B axis.


Synchronous gantry control
M0
Normally-On contact
M0
Normally-On contact
=
K0
D521
Switch for synchronous gantry control
M1088
Trigger for synchronous control
M1093
B slave axis follows the master axis
M0
Normally-On contact
M2227
Synchronous function in execution
A20
Synchronous gantry control is not enabled

Program execution procedure:

1. As soon as the system is powered on, if the switch for synchronous gantry control is 0, M0 is constantly on, and the system sets [Trigger for synchronous control] and [Slave axis follows the master axis] to on.
2. If the synchronous control function is not correctly enabled, the system generates an alarm with the MLC to remind the user.

Important:

1. To use the synchronous gantry control function, avoid enabling or disabling the synchronous control function in AUTO or MDI mode which might damage the machine.
2. According to the enabling rules, set the system to JOG or MPG mode when it is powered on, so the synchronous control function can be correctly enabled.
3. Multiple slave axes can follow the same master axis for synchronous control or command transfer at the same time.
4. When Pr616 (Origin search mode) is 0 - 5 and the synchronous control function is enabled, if Pr617 [Bit 4] (Homing option for sync. Motion) is 0, the slave axes and the master axis perform the homing procedure synchronously.
5. When the synchronous control function is enabled, the program cannot execute the command for the slave axis and displays alarm B015 (Sync cmd error).
6. The settings of the machine parameters and homing modes for the synchronous axes should be consistent.
7. When A, B, C, U, V, and W axes are the slave axes and X, Y, and Z axes are the master axes, the setting of Pr634 [Bit 1 - 3] (Rotation axis feed mode) for A, B, C, U, V, and W axes has to be 5. When A, B, C, U, V, and W axes are the master axes, the setting of Pr634 [Bit 1 - 3] for these axes has to be consistent.

6.21 3-spindle tapping

With the functions of NC / MLC switch, synchronous control, and command transfer, the system can perform the 3-spindle tapping operation. Because the settings of the servo parameters have to match for executing the tapping operation, the fixed combinations of axes are Z and A, U and B, and V and C. Currently, the 3-spindle tapping function is available only on the firmware for multi-spindle system.

■ MLC special M relay

For synchronous control and command transfer:

Axis	Trigger for synchronous control	Slave axis follows the master axis	Synchronous function in execution	Trigger for command transfer	Receive command from the master axis	Transfer function in execution
A axis	M1088	M1092	M2227	M1098	M1102	M2228
B axis		M1093			M1103	
C axis		M1094			M1104	
U axis		M1095			M1105	
V axis		M1096			M1106	
W axis		M1097			M1107	

For MLC axes:

	Activatie the axis	Switch between MLC position / speed mode	Speed command	Position / speed reached	The axis is moving
A axis	M1187	M1292	D1088	M2307	M2323
B axis	M1188	M1293	D1090	M2308	M2324
C axis	M1189	M1294	D1092	M2309	M2325

For switching between NC / MLC axis mode:

	Switch between NC and MLC axis mode	NC / MLC axis status
A axis	M1203	M2357
B axis	M1204	M2358
C axis	M1205	M2359

Refer to Sections 6.18 and 6.19 for detailed descriptions for the special M relays and special D registers mentioned above.

- **Relevant parameters**

Speed parameters:

The MLC axis moving speed is affected by these parameters: Pr621 G00/manual maximum speed, Pr622 G00/manual ACC/DEC time, and Pr623 G00/manual S-curve time.

Parameters for synchronization:

- ✓ Pr360 (Synchronization direction control) sets the synchronous directions of the motors to be the same or different.
  Pr360 = 0: the synchronous direction for the slave axes and the master axis is the same.
  Pr360 = 1: the synchronous direction for the slave axes and the master axis is different.
- ✓ Pr361 - Pr369 (Synchronous control): sets the master axis to be followed. When the parameter is set to 1, X axis is the master axis. 2 indicates Y axis, 3 indicates Z axis, and the same is true for others. If you set Pr364 to 3, the slave A axis follows Z axis synchronously.
- ✓ Pr617 [Bit 4] (Homing option for sync. Motion): sets whether the slave axes and the master axis return to the origin synchronously or each slave axis returns to the origin individually during synchronous control.
- ✓ Pr642 (Synchronous tolerance): sets the tolerance between the master and slaves during synchronous control. When the error exceeds the setting value of this parameter, the system generates alarm B645 (Excessive synchronous following error).
- ✓ Pr10009 (Sync coordinate setting): sets whether to display the coordinate of the synchronous axis (Bit 0) or the workpiece coordinate of the synchronous axis (Bit 2). When Pr10009 [Bit 0] or [Bit 2] is 0, the coordinate is not displayed. When Pr10009 [Bit 0] or [Bit 2] is 1, the coordinate is displayed.

Parameters for command transfer:

Pr371 - Pr379 (Transfer control): specifies the master axis to transfer the command. When the parameter is set to 1, X axis is specified as the master axis, 2 means Y axis, and 3 means Z axis. If you set Pr374 to 3, Z axis is the master axis which transfers the command to A axis.

Halt M code:

Enabling or disabling the synchronous control or command transfer function in AUTO or MDI mode is done with M code. The required M code has to be set in the halt M code. Refer to Section 6.7 for detailed information.

Servo parameters:

The matching of servo parameters for the tapping operation is usually done by the controller. As for the 3-spindle tapping, you need to manually set the servo parameters. Set P2-00 for the spindles A, B, and C with half of the setting value of P2-00 for the linear axes Z, U, and V. Then set P2-01 to 50 and P2-27 to 1 for Z, U, and V axes.

■ MLC example

To perform 3-spindle tapping operation, in addition to pairing Z, U, and V axes with A, B, and C axes, A, B, and C axes are set as MLC spindles, which should be switched back to the NC axis mode before tapping so the system can match the parameter settings for the axes. The following describes the MLC instructions and precautions in different applications.

Environment settings:

Z axis is enabled by M1020, U axis is enabled by M1022, and V axis is enabled by M1023.

Set Pr365 (Synchronous control B) to 4.

Set Pr366 (Synchronous control C) to 4.

Set Pr367 (Synchronous control U) to 3.

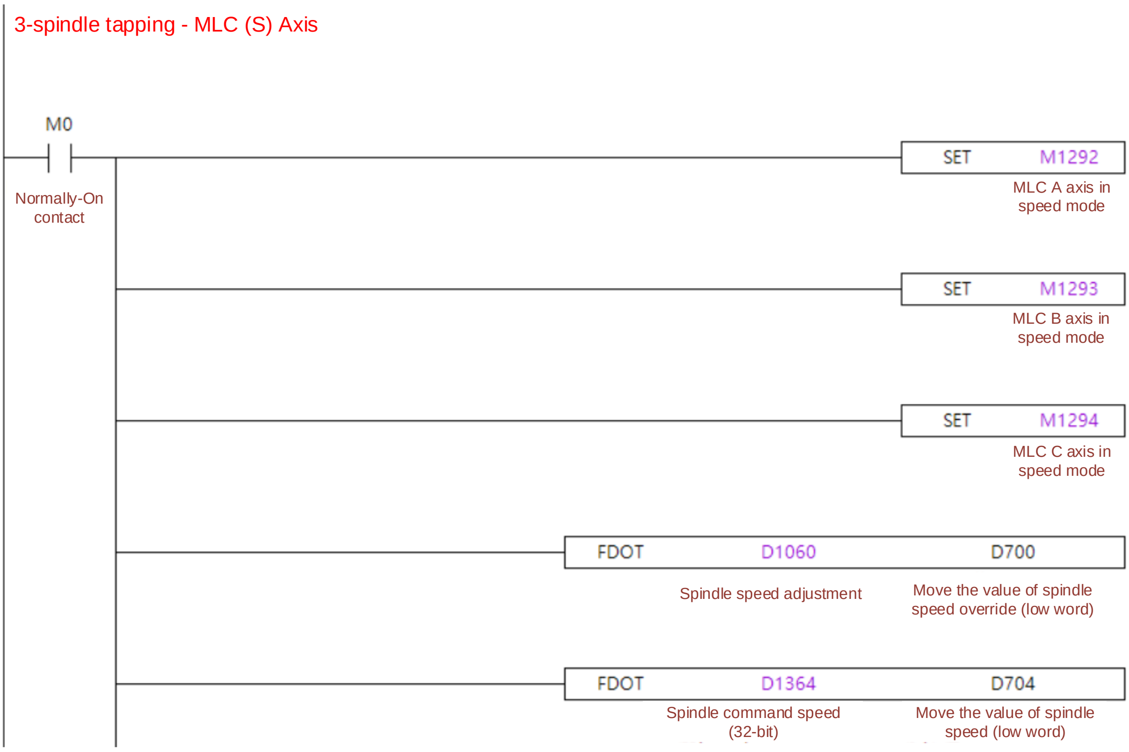
Set Pr368 (Synchronous control V) to 3.

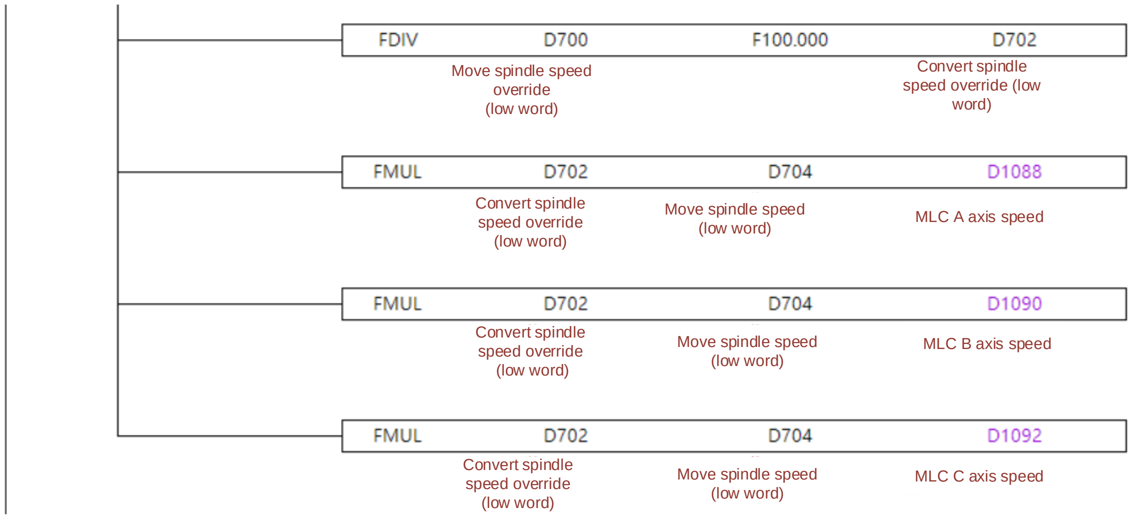
Set Pr375 (Transfer control B) to 4.

Set Pr376 (Transfer control C) to 4.

Set Pr377 (Transfer control U) to 3.

Set Pr378 (Transfer control V) to 3.

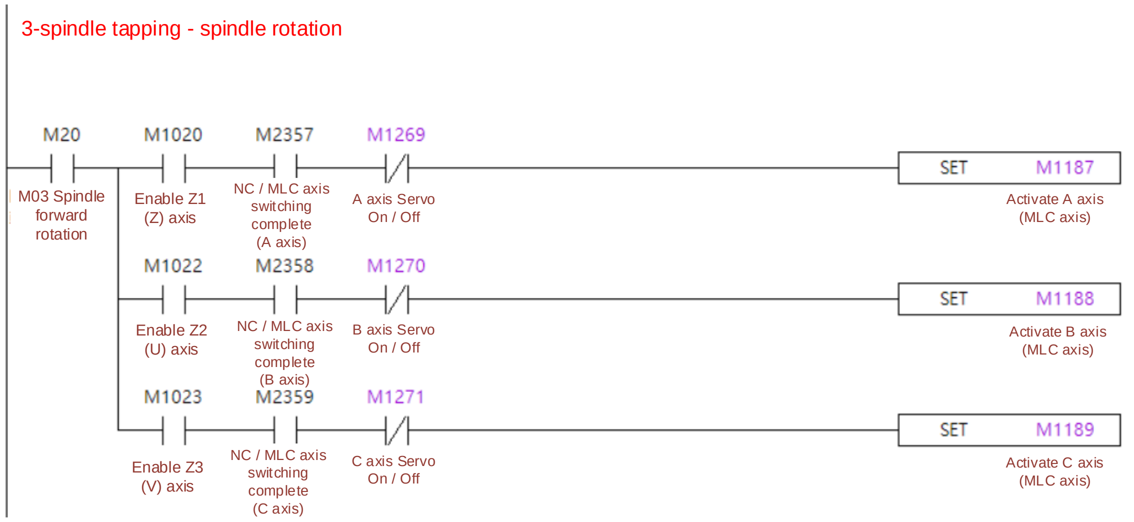



FDIV
D700
F100.000
D702
Move spindle speed override (low word)
Convert spindle speed override (low word)
FMUL
D702
D704
D1088
Convert spindle speed override (low word)
Move spindle speed (low word)
MLC A axis speed
FMUL
D702
D704
D1090
MLC B axis speed
FMUL
D702
D704
D1092
MLC C axis speed

Program execution procedure:

MLC (S) axis:

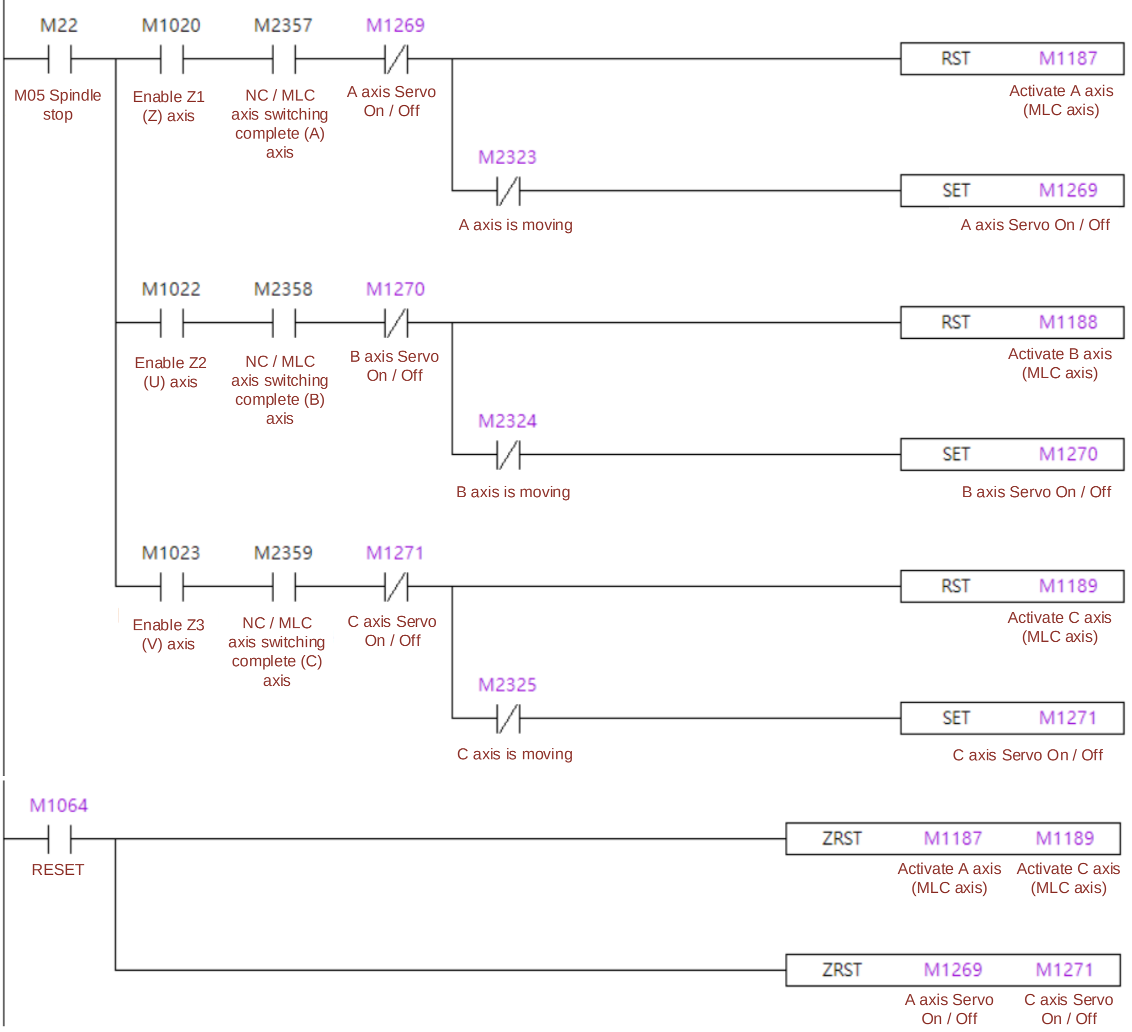
1. As soon as powered on, the system sets the MLC A, B, and C axes to be in speed mode.
2. Constantly convert the spindle speed override and rotation speed values into floating-point format with the API FDOT. As soon as the override and speed values are converted, the values are immediately written to D700 and D704.
3. Divide the override value by 100 and write it to D702 for speed control with speed override.
4. Multiply the rotation speed in floating-point format in D704 by the calculated override vlaue in D702, and set the result as the speed for the MLC A, B, and C axes.


3-spindle tapping - spindle rotation
M20
M03 Spindle forward rotation
M1020
Enable Z1 (Z) axis
M2357
NC / MLC axis switching complete (A axis)
M1269
A axis Servo On / Off
SET
M1187
Activate A axis (MLC axis)
M1022
Enable Z2 (U) axis
M2358
NC / MLC axis switching complete (B axis)
M1270
B axis Servo On / Off
SET
M1188
Activate B axis (MLC axis)
M1023
Enable Z3 (V) axis
M2359
NC / MLC axis switching complete (C axis)
M1271
C axis Servo On / Off
SET
M1189
Activate C axis (MLC axis)

Spindle rotation:

1. When the system executes the program in AUTO mode, it sets M20 to on in the M code procedure as soon as reading M3.
2. When M20 is on, the system determines whether the enabled axis (Z, U, or V) and the corresponding rotation axis are switched to MLC axis mode and are Servo On.
3. Set the enabled axis to rotate at the MLC axis speed.

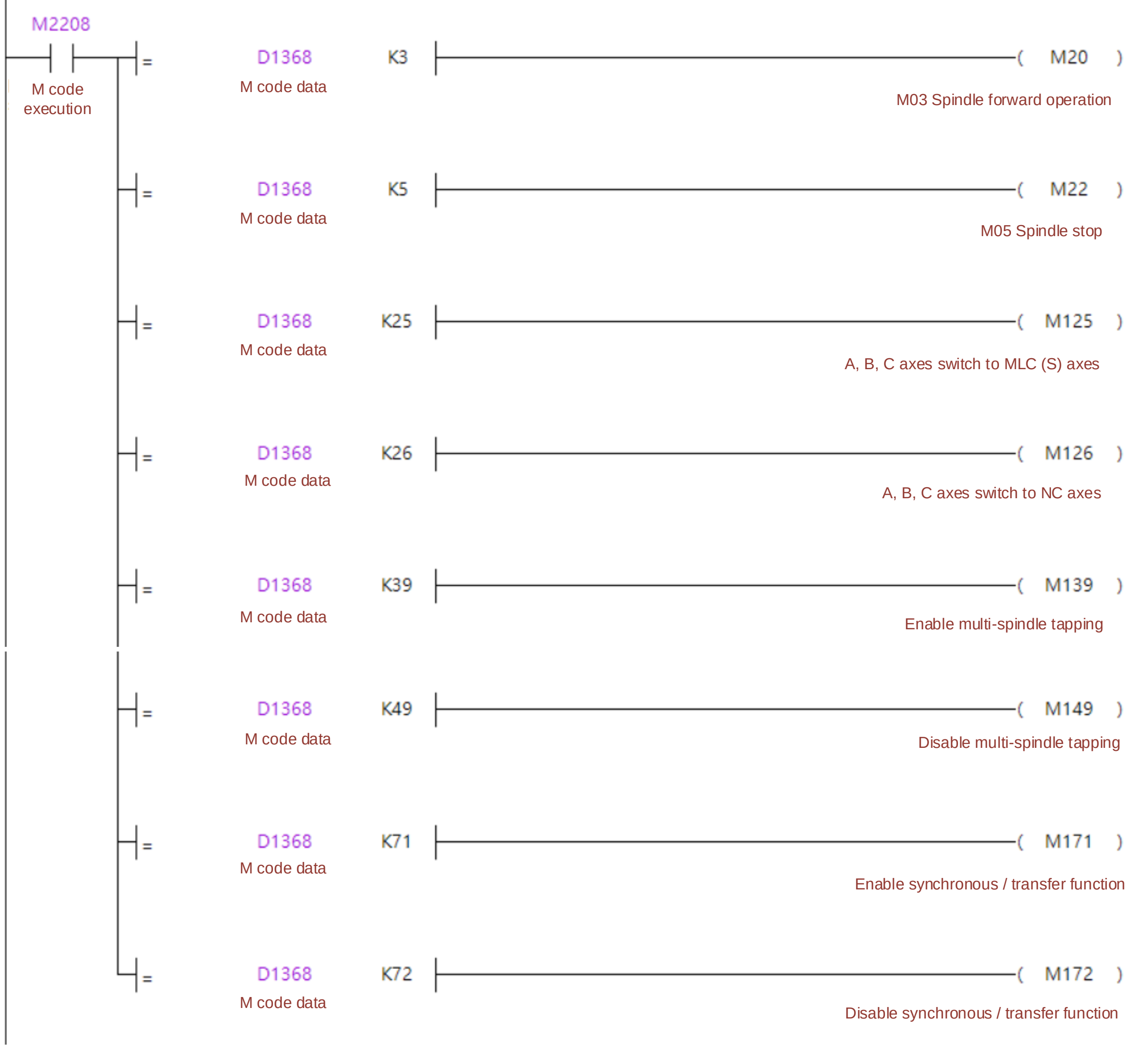
3-spindle tapping - spindle stops



Spindle stops:

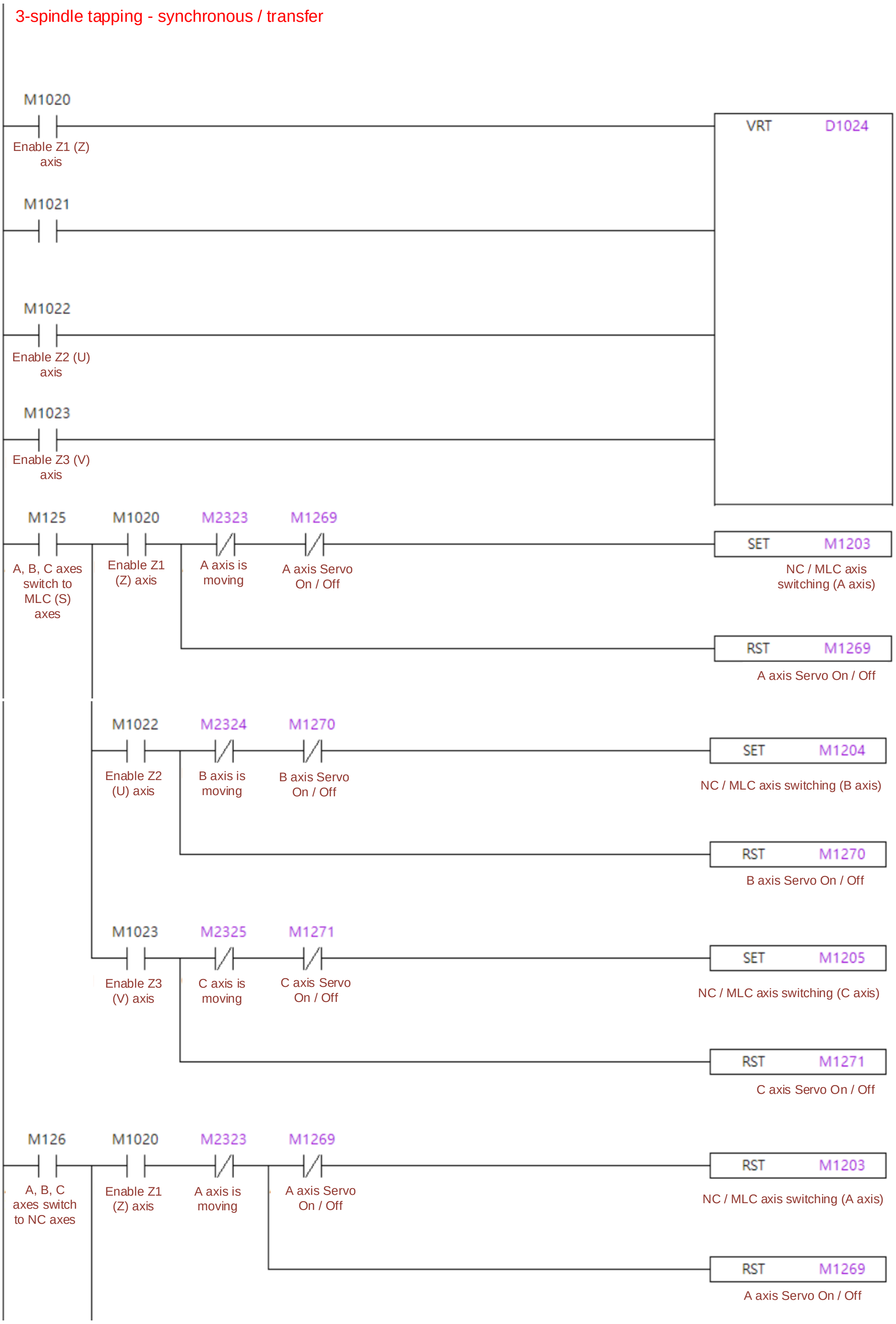
1. When the system executes the program in AUTO mode, it sets M22 to on in the M code procedure as soon as reading M5.
2. When M22 is on, the system determines whether the enabled axis (Z, U, or V) and the corresponding rotation axis are switched to MLC axis mode and are Servo On.
3. Stop the enabled axis.
4. After the axis stops rotating, set the [Axis Servo On / Off] special M relay to on to set the axis to Servo Off status, so the servo can prevent overflow by clearing the number of rotations.

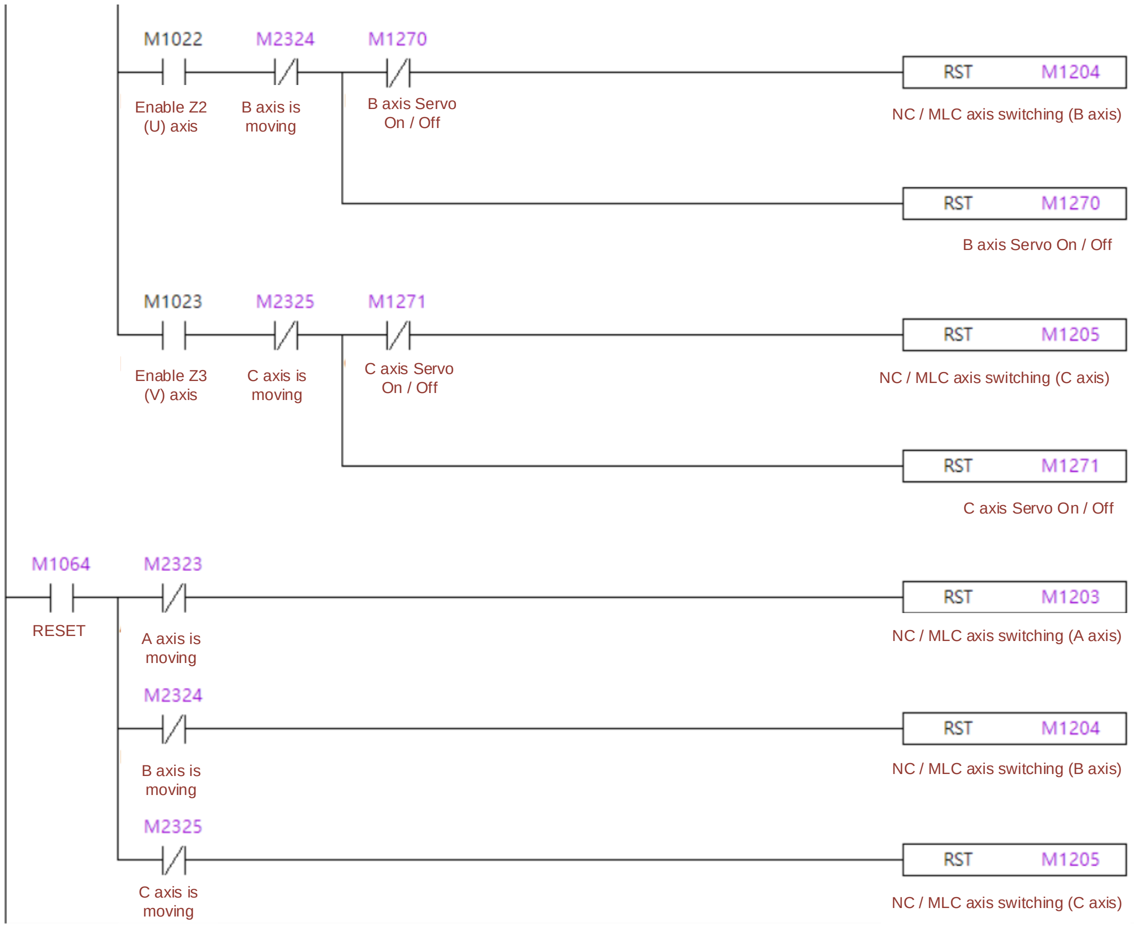
3-spindle tapping – M code procedure


M2208
M code execution
D1368
M code data
K3
M20
M03 Spindle forward operation
D1368
M code data
K5
M22
M05 Spindle stop
D1368
M code data
K25
M125
A, B, C axes switch to MLC (S) axes
D1368
M code data
K26
M126
A, B, C axes switch to NC axes
D1368
M code data
K39
M139
Enable multi-spindle tapping
D1368
M code data
K49
M149
Disable multi-spindle tapping
D1368
M code data
K71
M171
Enable synchronous / transfer function
D1368
M code data
K72
M172
Disable synchronous / transfer function

M code procedure:

When the system executes the program in AUTO mode, it enters the M code procedure as soon as reading the M code.


3-spindle tapping - synchronous / transfer
M1020
Enable Z1 (Z) axis
M1021
M1022
Enable Z2 (U) axis
M1023
Enable Z3 (V) axis
VRT D1024
M125
A, B, C axes switch to MLC (S) axes
M1020
Enable Z1 (Z) axis
M2323
A axis is moving
M1269
A axis Servo On / Off
SET M1203
NC / MLC axis switching (A axis)
RST M1269
A axis Servo On / Off
M1022
Enable Z2 (U) axis
M2324
B axis is moving
M1270
B axis Servo On / Off
SET M1204
NC / MLC axis switching (B axis)
RST M1270
B axis Servo On / Off
M1023
Enable Z3 (V) axis
M2325
C axis is moving
M1271
C axis Servo On / Off
SET M1205
NC / MLC axis switching (C axis)
RST M1271
C axis Servo On / Off
M126
A, B, C axes switch to NC axes
M1020
Enable Z1 (Z) axis
M2323
A axis is moving
M1269
A axis Servo On / Off
RST M1203
NC / MLC axis switching (A axis)
RST M1269
A axis Servo On / Off



Z axis switch:

When you select the Z, U, or V axis, the corresponding special M relay is set to on, and the corresponding value is written to D1024 through VRT.

Procedure for switching from NC to MLC axis mode:

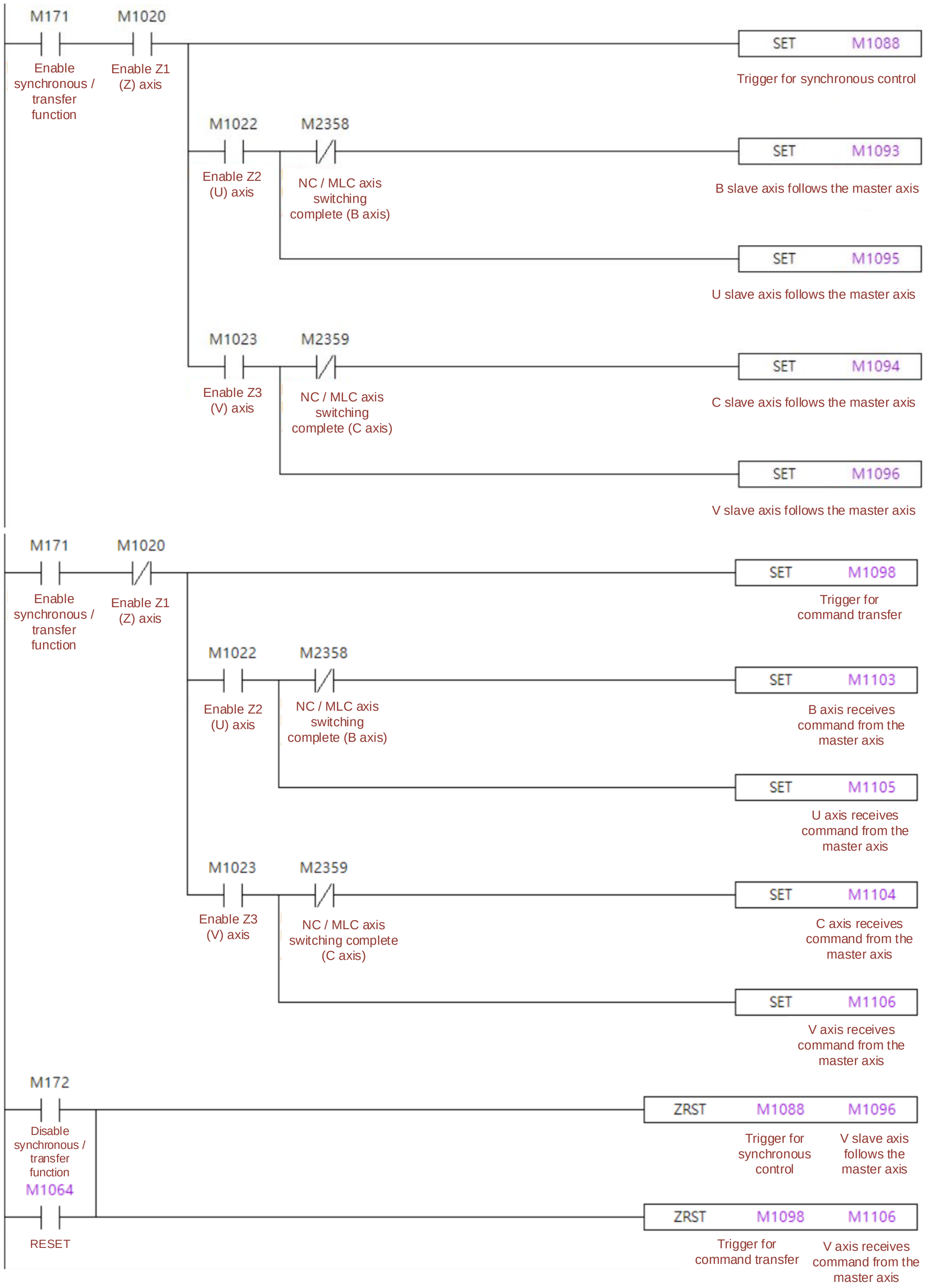
1. When the system executes the program in AUTO mode, it enters the M code procedure and sets M125 to on as soon as reading M25.
2. After M125 is set to on, determine which axis is enabled and then switch the corresponding rotation axis to the Servo On status.
3. When the rotation axis is in the Servo On status, ensure the axis is stopped and then switch it to the MLC axis mode.

Procedure for switching from MLC to NC axis mode:

1. When the system executes the program in AUTO mode, it enters the M code procedure and sets M126 to on as soon as reading M26.
2. After M126 is set to on, determine which axis is enabled and then switch the corresponding rotation axis to the Servo On status.
3. When the rotation axis is in the Servo On status, ensure the axis is stopped and then switch it to the NC axis mode.

Switch from MLC to NC axis mode when reset:

When RESET is triggered, the axis is switched from MLC to NC axis mode.

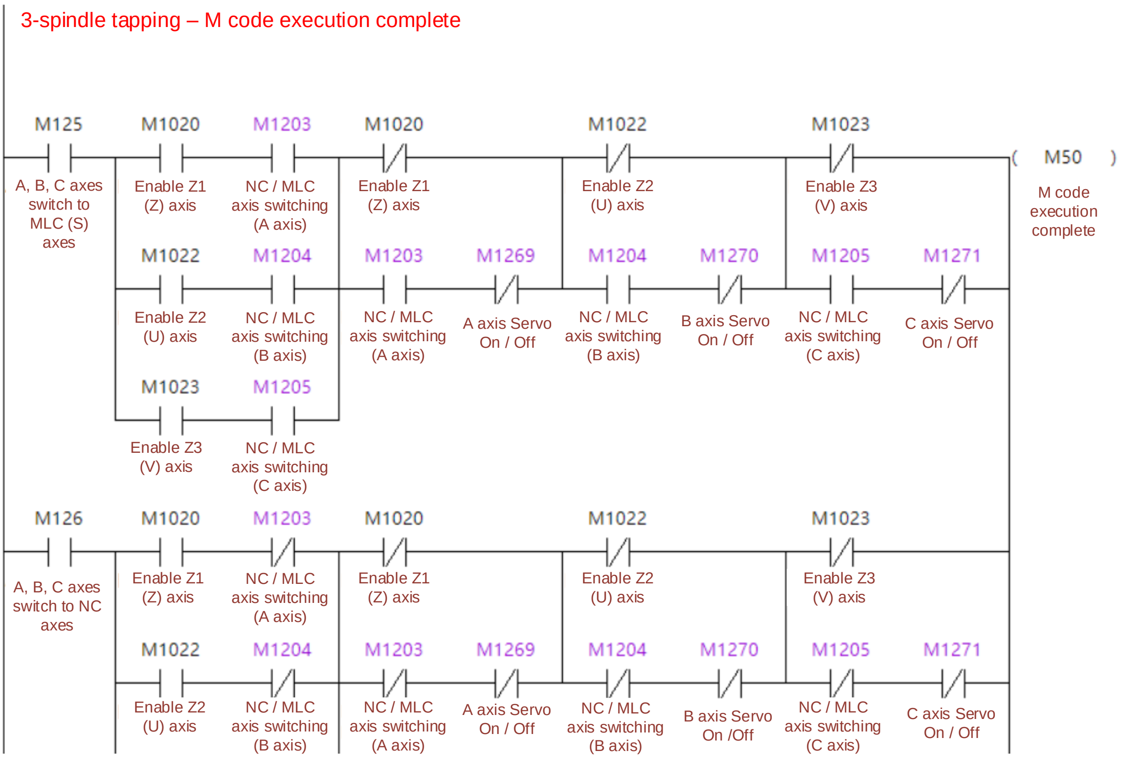

M171
M1020
Enable synchronous / transfer function
Enable Z1 (Z) axis
SET M1088
Trigger for synchronous control
M1022
M2358
Enable Z2 (U) axis
NC / MLC axis switching complete (B axis)
SET M1093
B slave axis follows the master axis
SET M1095
U slave axis follows the master axis
M1023
M2359
Enable Z3 (V) axis
NC / MLC axis switching complete (C axis)
SET M1094
C slave axis follows the master axis
SET M1096
V slave axis follows the master axis
M171
M1020
Enable synchronous / transfer function
Enable Z1 (Z) axis
SET M1098
Trigger for command transfer
M1022
M2358
Enable Z2 (U) axis
NC / MLC axis switching complete (B axis)
SET M1103
B axis receives command from the master axis
SET M1105
U axis receives command from the master axis
M1023
M2359
Enable Z3 (V) axis
NC / MLC axis switching complete (C axis)
SET M1104
C axis receives command from the master axis
SET M1106
V axis receives command from the master axis
M172
Disable synchronous / transfer function
ZRST M1088 M1096
Trigger for synchronous control
V slave axis follows the master axis
M1064
RESET
ZRST M1098 M1106
Trigger for command transfer
V axis receives command from the master axis

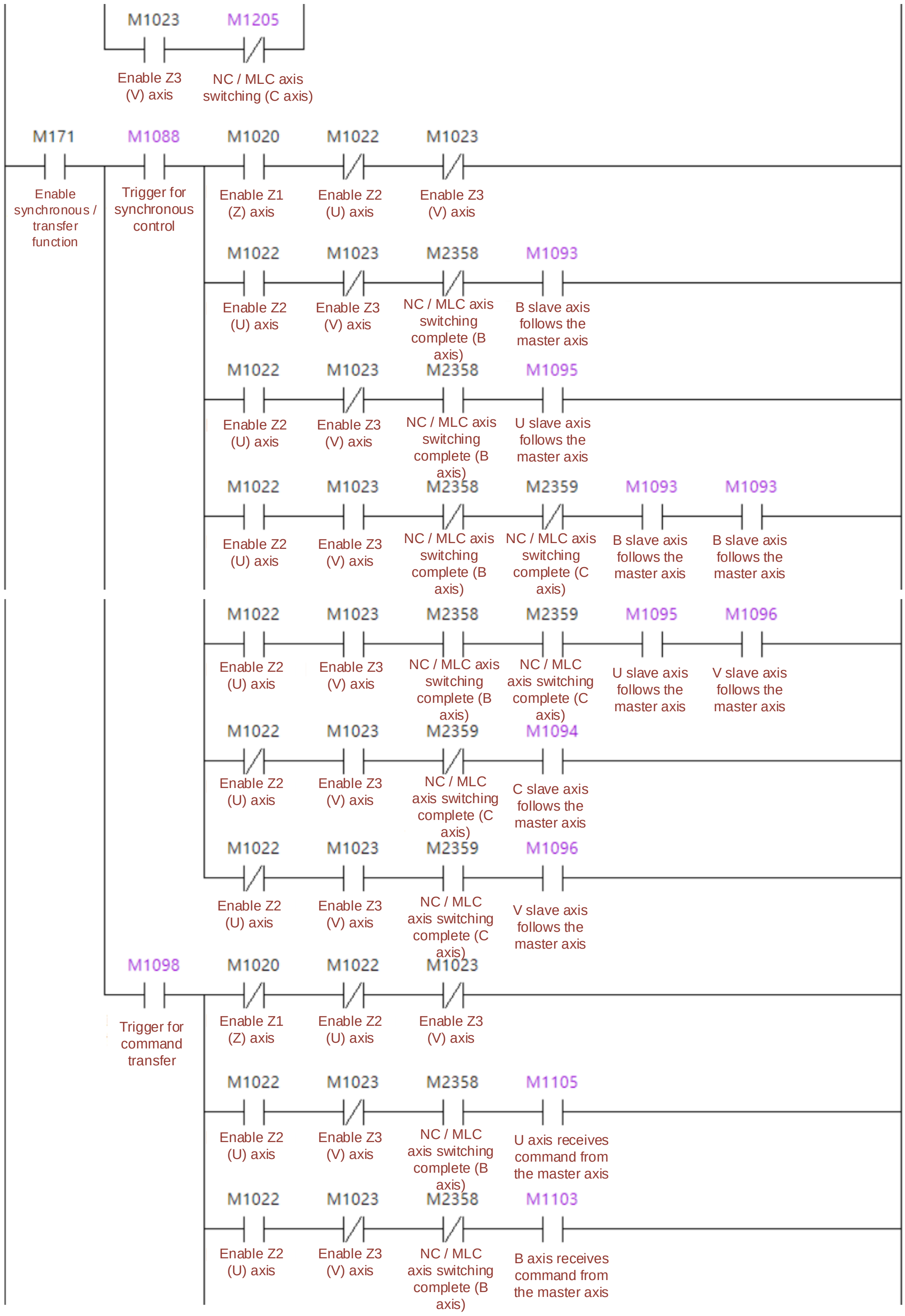
Synchronous / transfer (non-tapping) procedure:

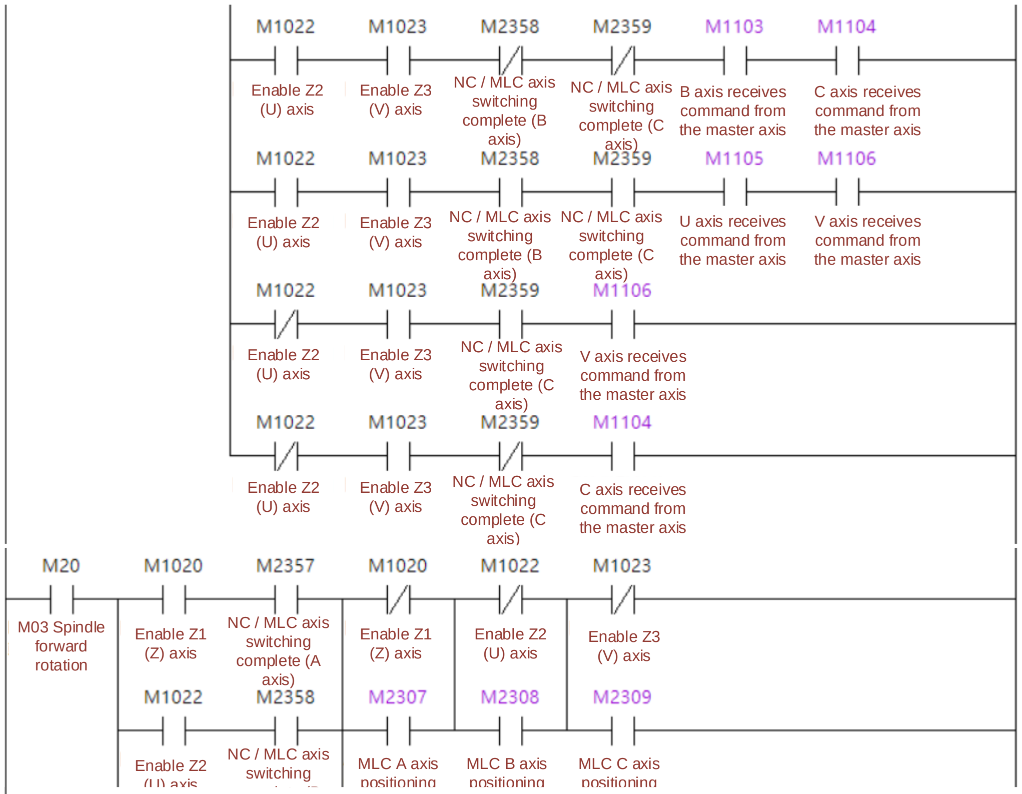
1. When the system executes the program in AUTO mode, it enters the M code procedure and sets M171 to on as soon as reading M71.
2. After M171 is on, the system determines if Z axis is enabled to decide whether to enter the synchronous or transfer procedure.
3. When entering the synchronous or transfer procedure, the system determines which axis is enabled, and if the rotation axis is selected and set as an MLC axis. If the rotation axis is not set as an MLC axis, it means the system is going to perform the tapping operation. Thus, the synchronous control or transfer function has to be enabled.
4. After finishing the judgment mentioned in Step 3, the system enables the synchronous control or transfer function for the corresponding axis.

Note: the rotation axis is set as the MLC spindle during non-tapping operation. Accordingly, you can enable the synchronous or transfer function only for U and V axes.

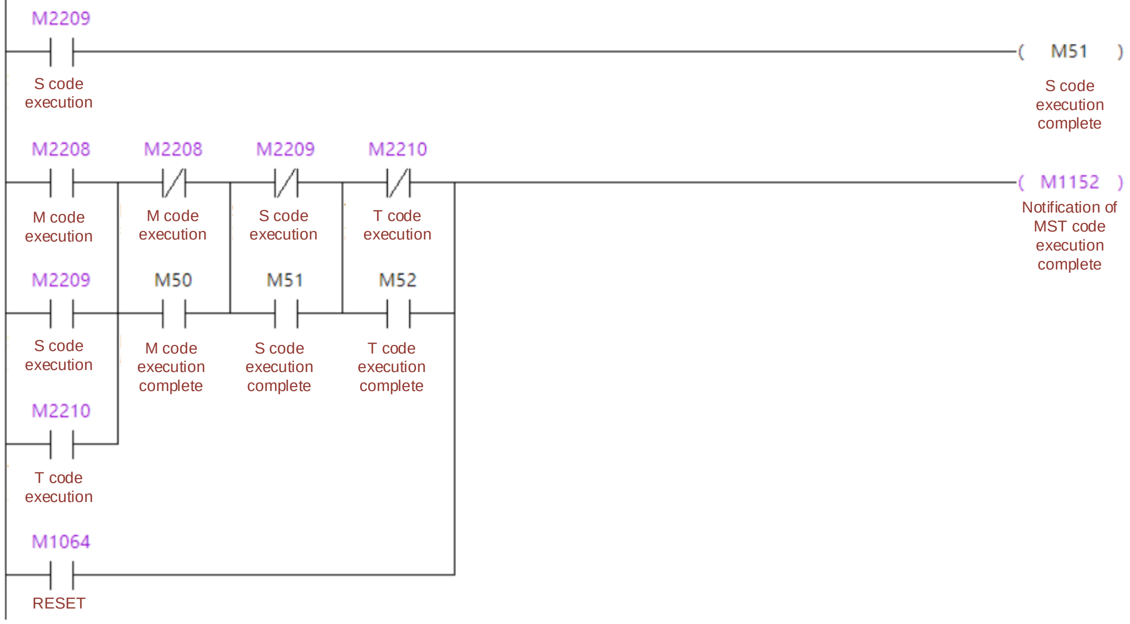
3-spindle tapping – M code execution complete




M1023
M1205
Enable Z3 (V) axis
NC / MLC axis switching (C axis)
M171
M1088
M1020
M1022
M1023
Enable synchronous / transfer function
Trigger for synchronous control
Enable Z1 (Z) axis
Enable Z2 (U) axis
Enable Z3 (V) axis
M1022
M1023
M2358
M1093
Enable Z2 (U) axis
Enable Z3 (V) axis
NC / MLC axis switching complete (B axis)
B slave axis follows the master axis
M1022
M1023
M2358
M1095
Enable Z2 (U) axis
Enable Z3 (V) axis
NC / MLC axis switching complete (B axis)
U slave axis follows the master axis
M1022
M1023
M2358
M2359
M1093
M1093
Enable Z2 (U) axis
Enable Z3 (V) axis
NC / MLC axis switching complete (B axis)
NC / MLC axis switching complete (C axis)
B slave axis follows the master axis
B slave axis follows the master axis
M1022
M1023
M2358
M2359
M1095
M1096
Enable Z2 (U) axis
Enable Z3 (V) axis
NC / MLC axis switching complete (B axis)
NC / MLC axis switching complete (C axis)
U slave axis follows the master axis
V slave axis follows the master axis
M1022
M1023
M2359
M1094
Enable Z2 (U) axis
Enable Z3 (V) axis
NC / MLC axis switching complete (C axis)
C slave axis follows the master axis
M1022
M1023
M2359
M1096
Enable Z2 (U) axis
Enable Z3 (V) axis
NC / MLC axis switching complete (C axis)
V slave axis follows the master axis
M1098
M1020
M1022
M1023
Trigger for command transfer
Enable Z1 (Z) axis
Enable Z2 (U) axis
Enable Z3 (V) axis
M1022
M1023
M2358
M1105
Enable Z2 (U) axis
Enable Z3 (V) axis
NC / MLC axis switching complete (B axis)
U axis receives command from the master axis
M1022
M1023
M2358
M1103
Enable Z2 (U) axis
Enable Z3 (V) axis
NC / MLC axis switching complete (B axis)
B axis receives command from the master axis


M1022
M1023
M2358
M2359
M1103
M1104
Enable Z2 (U) axis
Enable Z3 (V) axis
NC / MLC axis switching complete (B axis)
NC / MLC axis switching complete (C axis)
B axis receives command from the master axis
C axis receives command from the master axis
M1022
M1023
M2358
M2359
M1105
M1106
Enable Z2 (U) axis
Enable Z3 (V) axis
NC / MLC axis switching complete (B axis)
NC / MLC axis switching complete (C axis)
U axis receives command from the master axis
V axis receives command from the master axis
M1022
M1023
M2359
M1106
Enable Z2 (U) axis
Enable Z3 (V) axis
NC / MLC axis switching complete (C axis)
V axis receives command from the master axis
M1022
M1023
M2359
M1104
Enable Z2 (U) axis
Enable Z3 (V) axis
NC / MLC axis switching complete (C axis)
C axis receives command from the master axis
M20
M1020
M2357
M1020
M1022
M1023
M03 Spindle forward rotation
Enable Z1 (Z) axis
NC / MLC axis switching complete (A axis)
Enable Z1 (Z) axis
Enable Z2 (U) axis
Enable Z3 (V) axis
M1022
M2358
M2307
M2308
M2309
Enable Z2 (U) axis
NC / MLC axis switching
MLC A axis positioning
MLC B axis positioning
MLC C axis positioning

3-spindle tapping - S code and MST code execution complete



Important:

1. Manually set the servo parameters to match them for the tapping operation.
2. To set the rotation axes as MLC spindles, use M3 or M4 to call the macro, switch the axes to the Servo On status in sequence in the macro, and then switch the axes to MLC axis mode.
3. When the rotation axes rotate as MLC spindles, to stop the spindles, use M5 to call the macro and stop the MLC axes in sequence in the macro, and then switch the axes to the Servo Off status, so the servo prevents overflow by clearing the number of rotations.
4. Before performing the tapping operation, use M code for macro call, switch the axes to the

Servo On status in sequence in the macro, switch the axes from MLC to NC axis mode, position the rotation axes to the angle for tapping, and then enable the synchronous control or transfer function.

5. According to the example, during non-tapping operation, switch the rotation axis to MLC axis mode before executing the synchronous control or transfer operation, so the MLC can correctly determine the condition.
6. When the synchronous control function is enabled, the program cannot execute the command for the slave axis and displays alarm B015 (Sync cmd error).
7. The settings of the machine parameters and homing modes for the synchronous axes should be consistent.
8. When A, B, C, U, V, and W axes are the slave axes and X, Y, and Z axes are the master axes, the setting of Pr634 [Bit 1 - 3] (Rotation axis feed mode) for A, B, C, U, V, and W axes has to be 5. When A, B, C, U, V, and W axes are the master axes, the setting of Pr634 [Bit 1 - 3] for these axes has to be consistent.

6.22 User-defined hardware signal

When using the RIO function for hardware limit and origin signals, the system assigns the corresponding input points. By editing the MLC program with the following special D registers, you can freely set the input points for hardware limit and origin signals of the axes.

- **MLC special D register**

User-defined hardware signal	D1500
User-defined hardware signal	D1501
User-defined hardware signal	D1502
User-defined hardware signal	D1503

[User-defined hardware signal]: D1500 - D1503

When Pr2006 [Bit 14] (Origin / limit planning) is set to 1, the value is assigned to **[User-defined hardware signal]** in bits, allowing you to define the positive and negative limit signals and origin signals with the MLC.

Bit definition:

D1500		D1501	
Bit	Function	Bit	Function
Bit 0	Port 1 hardware left limit signal	Bit 0	Port 5 hardware left limit signal
Bit 1	Port 1 hardware right limit signal	Bit 1	Port 5 hardware right limit signal
Bit 2	Port 1 origin signal	Bit 2	Port 5 origin signal
Bit 3	Reserved	Bit 3	Reserved
Bit 4	Port 2 hardware left limit signal	Bit 4	Port 6 hardware left limit signal
Bit 5	Port 2 hardware right limit signal	Bit 5	Port 6 hardware right limit signal
Bit 6	Port 2 origin signal	Bit 6	Port 6 origin signal
Bit 7	Reserved	Bit 7	Reserved
Bit 8	Port 3 hardware left limit signal	Bit 8	Port 7 hardware left limit signal
Bit 9	Port 3 hardware right limit signal	Bit 9	Port 7 hardware right limit signal
Bit 10	Port 3 origin signal	Bit 10	Port 7 origin signal
Bit 11	Reserved	Bit 11	Reserved
Bit 12	Port 4 hardware left limit signal	Bit 12	Port 8 hardware left limit signal
Bit 13	Port 4 hardware right limit signal	Bit 13	Port 8 hardware right limit signal
Bit 14	Port 4 origin signal	Bit 14	Port 8 origin signal
Bit 15	Reserved	Bit 15	Reserved

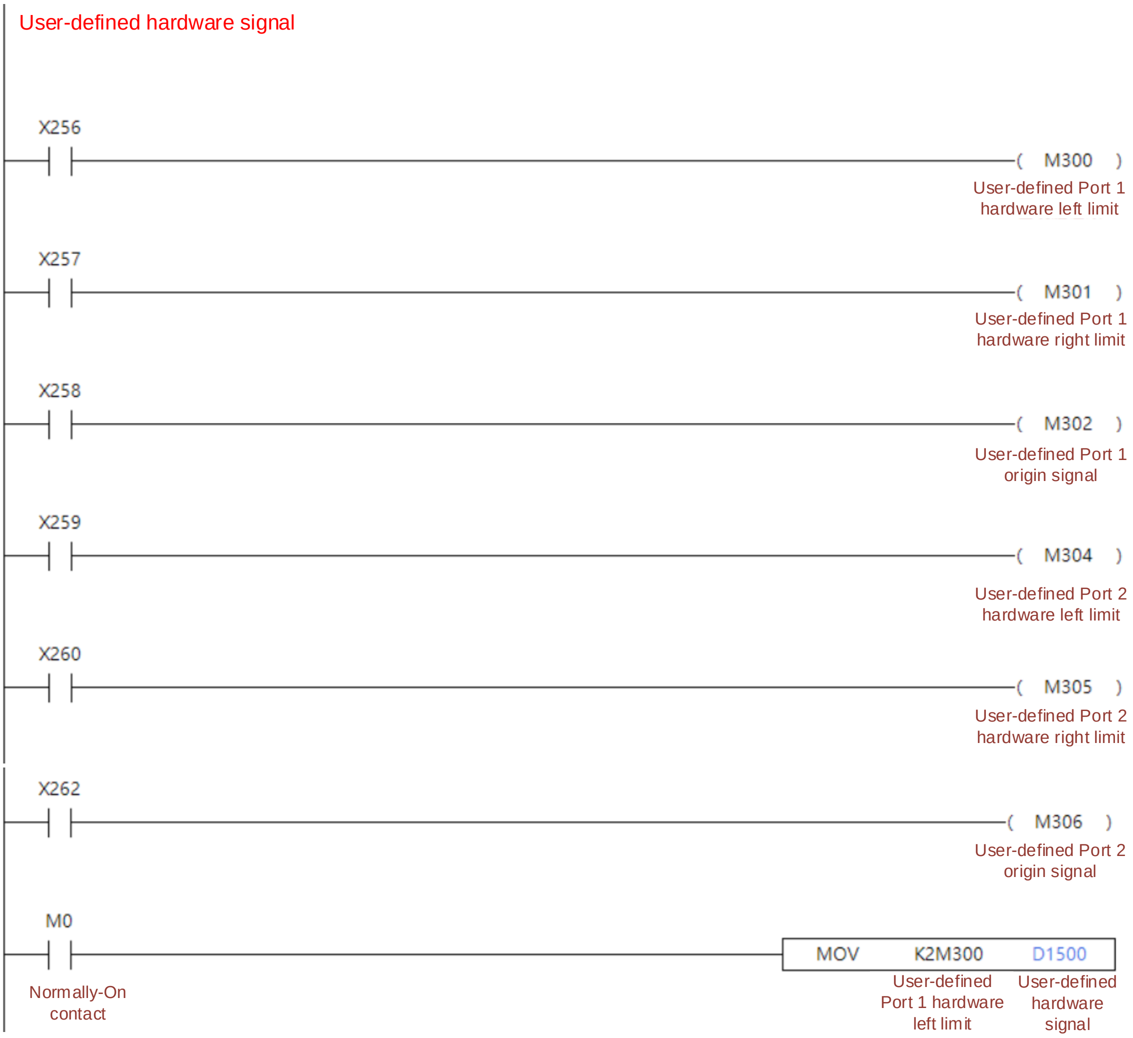
D1502		D1503	
Bit	Function	Bit	Function
Bit 0	Port 9 hardware left limit signal	Bit 0	Port 13 hardware left limit signal
Bit 1	Port 9 hardware right limit signal	Bit 1	Port 13 hardware right limit signal
Bit 2	Port 9 origin signal	Bit 2	Port 13 origin signal
Bit 3	Reserved	Bit 3	Reserved
Bit 4	Port 10 hardware left limit signal	Bit 4	Port 14 hardware left limit signal
Bit 5	Port 10 hardware right limit signal	Bit 5	Port 14 hardware right limit signal
Bit 6	Port 10 origin signal	Bit 6	Port 14 origin signal
Bit 7	Reserved	Bit 7	Reserved
Bit 8	Port 11 hardware left limit signal	Bit 8	Port 15 hardware left limit signal
Bit 9	Port 11 hardware right limit signal	Bit 9	Port 15 hardware right limit signal
Bit 10	Port 11 origin signal	Bit 10	Port 15 origin signal
Bit 11	Reserved	Bit 11	Reserved
Bit 12	Port 12 hardware left limit signal	Bit 12	Port 16 hardware left limit signal
Bit 13	Port 12 hardware right limit signal	Bit 13	Port 16 hardware right limit signal
Bit 14	Port 12 origin signal	Bit 14	Port 16 origin signal
Bit 15	Reserved	Bit 15	Reserved

- **Relevant parameters**

Origin / limit planning:

When Pr2006 [Bit 4] (Origin / limit planning) is set to 0, the system sets the RIO input points as the origin and limit signals of the specific port in sequence according to the original setting of origin and limit in the RIO. When Pr2006 [Bit 4] is set to 1, the RIO points are only for receiving the input signals. You have to write the corresponding signals to [User-defined hardware signal] with the MLC so the system performs the corresponding actions.

■ MLC example


User-defined hardware signal
X256
M300
User-defined Port 1 hardware left limit
X257
M301
User-defined Port 1 hardware right limit
X258
M302
User-defined Port 1 origin signal
X259
M304
User-defined Port 2 hardware left limit
X260
M305
User-defined Port 2 hardware right limit
X262
M306
User-defined Port 2 origin signal
M0
Normally-On contact
MOV
K2M300
D1500
User-defined Port 1 hardware left limit
User-defined hardware signal

Program execution procedure:

1. You can plan actions of the RIO input points with the user-defined hardware signals. In the example, the origin of Port 2 is X262 rather than X261; when X260 is triggered, M305 becomes on.
2. When M305 is on, the data is automatically written to D1500 in bits. Meanwhile, the system generates an alarm indicating the hardware limit is reached. Since there are only two axes in the example, the instruction K2M300 is used. If there are three axes, the instruction K3M300 is used instead.

Important:

1. Bit 3, Bit 7, Bit 11, and Bit 15 of the [User-defined hardware signal] are reserved.
2. When the instruction MOV K__M__ D1500 is used, K1M_ indicates there are four M devices, K2M_ indicates there are eight M devices, and the same is true for others. These M devices are written to the reserved bits as well. Pay special attention to this when programming the MLC.
3. When the user-defined hardware signals are in use, the hardware limits do not refer to the settings for positive limit sensor, negative limit sensor, and home sensor in Pr628.

6.23 Dynamic compensation for axis

When an axis requires compensation, the compensation is set with the offset coordinate system or compensation parameters in advance, which cannot be promptly adjusted during operation. For applications requiring dynamic compensation, the system provides the option of dynamic compensation function. Refer to the following for descriptions.

■ **MLC special D register**

Function name	Device number	Unit
Equi-slope filter compensation for X axis	D1126	mm
Equi-slope filter compensation for Y axis	D1128	mm
Equi-slope filter compensation for Z axis	D1130	mm
Equi-slope filter compensation for A axis	D1132	mm or deg
Equi-slope filter compensation for B axis	D1134	mm or deg
Equi-slope filter compensation for C axis	D1136	mm or deg
Equi-slope filter compensation for U axis	D1138	mm or deg
Equi-slope filter compensation for V axis	D1140	mm or deg
Equi-slope filter compensation for W axis	D1142	mm or deg
Linear axis compensation speed	D1144	mm/s
Rotation axis compensation speed	D1146	deg/s

[Equi-slope filter compensation for axis]: D1126, D1128, D1130, D1132, D1134, D1136, D1138, D1140, D1142

Sets the maximum compensation amount for the axis during dynamic compensation. When you set 5 for the linear axis, the maximum compensation amount is 5 mm. That is, the original command value increases by 5 mm. When you set 10 for the rotation axis, the maximum compensation amount is 10 degrees. That is, the original rotation degree increases by 10 degrees.

- ✓ When performing dynamic compensation for the axis, the system compensates for the specified axis by equi-slope referring to the setting of [Axis compensation speed] until reaching the set amount in **[Equi-slope filter compensation for axis]**.
- ✓ The maximum compensation amount for the linear axis is ±10 mm.
- ✓ The maximum compensation amount for the rotation axis is ±360 degrees.
- ✓ When the input value exceeds the maximum compensation amount, the system compensates only for the maximum compensation amount.
- ✓ The **[Equi-slope filter compensation for axis]** special D registers are 32-bit, so the input value has to be in the floating-point format.

[Axis compensation speed]: D1144, D1146

Sets the displacement to be compensated per time unit for axis compensation. When you set the axis compensation speed to 5 for the linear axis, the system compensates 5 mm per second, until reaching the setting value of [Equi-slope filter compensation for axis]. When you set the axis compensation speed to 3 for the rotation axis, the system compensates 3 degrees per second, until reaching the setting of [Equi-slope filter compensation for axis].

- ✓ The maximum compensation speed for the linear axis is 10 mm/s.
- ✓ The maximum compensation speed for the rotation axis is 360 degree/s.
- ✓ When the input value exceeds the maximum compensation speed, the system compensates at the maximum compensation speed.
- ✓ The **[Axis compensation speed]** special D registers are 32-bit, so the input value has to be in the floating-point format.

■ Relevant parameters

Dynamic compensation settings:

When Pr308 [Bit 7] (Dynamic compensation for axis movement) is set to 0, the compensation function is disabled.

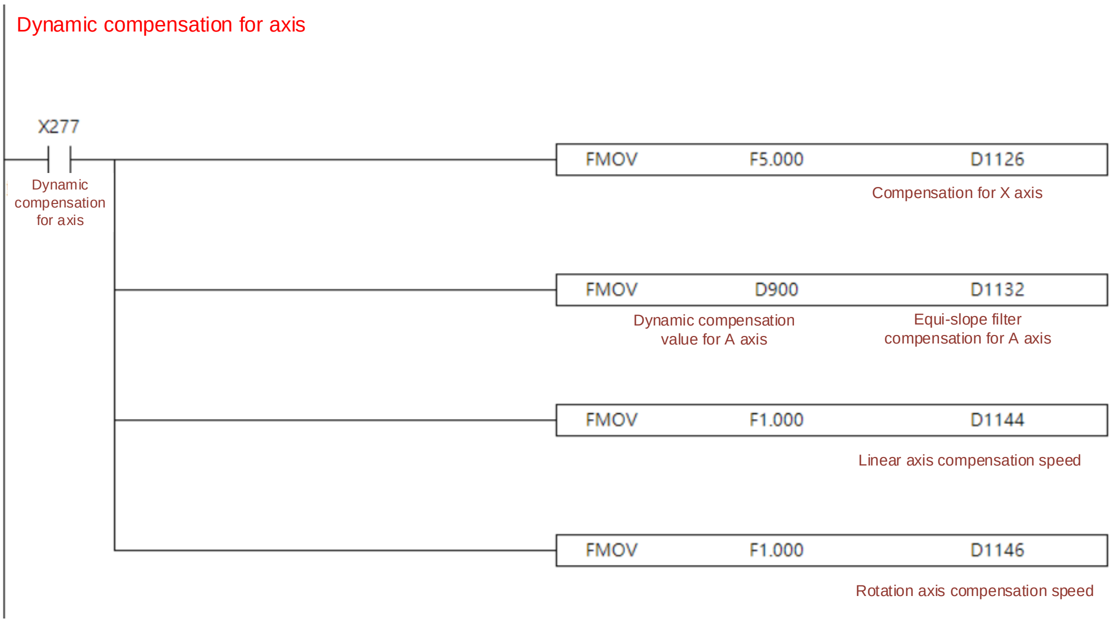
When Pr308 [Bit 7] is set to 1, the compensation function is enabled.

Linear and rotation axes settings:

The system refers to Pr634 [Bit 1 - 3] (Rotation axis feed mode) to determine whether A, B, C, U, V, and W axes are linear or rotation axes when performing compensation.

■ MLC example

Refer to X and A axes in the following MLC example.


Dynamic compensation for axis
X277
Dynamic compensation for axis
FMOV F5.000 D1126
Compensation for X axis
FMOV D900 D1132
Dynamic compensation value for A axis
Equi-slope filter compensation for A axis
FMOV F1.000 D1144
Linear axis compensation speed
FMOV F1.000 D1146
Rotation axis compensation speed

Program execution procedure:

1. When X277 is on, write 5 to the dynamic compensation for X axis, write the value in D900 to the dynamic compensation for A axis, and then write the compensation speed for the axes respectively.
2. After the value is written, the system immediately performs the compensation.

Important:

The dynamic compensation for the axes is available in all system modes. As soon as the compensation value is changed, the system performs the compensation.

Revision History

Release Date	Version	Revised sections	Revision contents
June, 2021	V4.0		
		1.1.1	Modified the device range
		1.1.2	Modified the table and the range of the devices
		1.1.2	Modified the counter range and added the information about 32-bit high-speed counter
		1.3.1	Corrected the number of I/O of the main board.
		1.6.1	Corrected the diagram; X1 is changed to T0
		1.8.1	Corrected the table contents and re-categorize the contents based on the definitions
		1.8.1	Corrected the description of index registers
		1.9	Corrected the number of devices for interruption
		1.9	Corrected the Remote I/O input range of external interruption
		1.9	Corrected the assigned API number for interruption after counter reached
		2.1	Corrected the step information of the output instruction
		2.2	Corrected the K range of the 16-bit timer T-K
		2.2	Corrected the K range of the 16-bit counter C-K
		2.2	Corrected the operands of the 32-bit counter
		2.2	Corrected the operand range for the interruption instruction
		3.1	Added the FMOV instruction
		3.1	Added the WRTL instruction
		3.1	Added the RDTL instruction
		3.1	Corrected the step information of PLF
		3.2.1	Added the table including both 16-bit and 32-bit (F) instructions
		3.2.2	Modified the API numbering
		3.5	Added the FMOV instruction
		3.5	Added the WRTL instruction
		3.5	Added the RDTL instruction
		4.1	Corrected the flag signal of interruption enabling instruction

Release Date	Version	Revised sections	Revision contents
		4.2	Added the FMOV instruction descriptions
		4.2	Corrected the instruction example diagram of CML (invert and transit) instruction
		4.2	Corrected the flag signal and example diagram of the instruction for converting BIN to BCD
		4.3	Corrected the flag signal and step information of BIN division
		4.3	Corrected the step information of WXOR and XOR
		4.4	Corrected the example diagram for instruction of rotation to the right
		4.4	Corrected the example diagram and step information for instruction of rotation to the left
		4.4	Corrected the source device for the encoder
		4.5	Corrected the instruction name of DECO
		4.5	Corrected the name of BON
		4.5	Correct the step information of ANS (Alarm trigger)
		4.6	Corrected the flag signal of DHSCS comparison setting (high-speed counter)
		4.9	Modified the title
		4.9	Corrected the flag signal for binary floating-point number addition
		4.9	Corrected the flag signal for binary floating-point number subtraction
		4.9	Corrected the flag signal for binary floating-point number multiplication
		4.9	Corrected the flag signal for binary floating-point number division
		4.9	Corrected the flag signal for converting integers to binary floating point number
		4.9	Corrected the flag signal for converting degrees to radians
		4.9	Corrected the flag signal for converting radians to degrees
		4.10	Added the WRTL instruction descriptions
		4.10	Added the RDTL instruction descriptions

Release Date	Version	Revised sections	Revision contents
		5.2	Added the special M for M99 halt function for the main program
		5.2	Added the special M for M96 interruption
		5.2	Added the special M for MLC emergency stop trigger
		5.2	Added the special M for cancelling restricted zone protection
		5.2	Added the special M relays of “slave axis follows the master axis” for UVW axes
		5.2	Added the special M relays of UVW axes receive commands from the master axis
		5.2	Added the special M relay for lathe system spindle / C axis switch
		5.2	Added the special M relay for 1st spindle analog voltage ratio gain
		5.2	Added the special M relays for 2nd spindle forward / reverse operation
		5.2	Added the special M relays for 2nd spindle gear ratio selection
		5.2	Added the special M relays for 2nd spindle positioning control
		5.2	Added the special M relays for 2nd spindle return after tapping
		5.2	Added the special M relays for 2nd spindle analog voltage ratio gain
		5.2	Added the special M relays for NC/MLC switch for X – W axes
		5.2	Added the special M relays for increment command switch for MLC X – W axes
		5.2	Added the special M relays for control modes of MLC X – W axes
		5.2	Added the special M relay for high-speed input point trigger of MLC X – W axes
		5.2	Added the special M relay for spindle speed command source switch
		5.2	Added the special M relays for movement permission of X – W axes

Release Date	Version	Revised sections	Revision contents
		5.2	Added the special M relays for X – W axes machine positive lock
		5.2	Added the special M relays for X – W axes machine negative lock
		5.2	Corrected the name and description of M2113
		5.2	Added the special M relays for high-speed input feedback of MLC X – W axes
		5.2	Added the special M for M96 (program interruption) in execution
		5.2	Added the special M for G code preparation complete
		5.2	Added the special M for MPG forward and reverse operations
		5.2	Added the special M for M99 halt function
		5.2	Added the special M for lathe spindle / C axis mode switch
		5.2	Added the special M for 2nd spindle target speed reached
		5.2	Added the special M for 2nd spindle zero speed reached
		5.2	Added the special M relay for 2nd spindle positioning complete
		5.2	Added the special M relay for 2nd spindle rigid tapping in execution
		5.2	Added the special M relay for 2nd spindle rigid tapping interrupted
		5.2	Removed the special M relay for spindle positioned at origin
		5.2	Added the special M relay for X – W axes positioned at 3rd reference point
		5.2	Added the special M relays for X – W axes forward and reverse operations
		5.2	Added the special M relays for NC/MLC switch complete of X – W axes
		5.2	Added the special M relays for X – W axes positioned at the 4th reference point
		5.2	Added the special M relays for homing status of X

Release Date	Version	Revised sections	Revision contents
			– W axes
		5.2	Added the special M relays for Servo On/Off status of X – W axes
		5.2	Added the special M relays for indicating movement of X – W axes
		5.2	Added the special M relay for indicating that the MLC instruction operation result is 0
		5.2	Added the special M relay for MLC instruction operation borrow flag
		5.2	Added the special M for MLC instruction carry operation
		5.2	Added the special M for MLC instruction operation error
		5.2	Added the special M relays for DMCNET connection status (stations 1 – 12)
		5.2	Added the special M relays for the count down switch of DCNT C64 – C77 counters
		5.2	Added the special M relays of auto machining for file queue
		5.2	Added the special D register (32-bit) for the count of the machined workpieces
		5.2	Added the special D register (32-bit) for target machining count
		5.2	Added the special D register for “DMCNET Ready for HMI”
		5.2	Added the special D register for main file name
		5.2	Added the special D register for total machining time
		5.2	Added the special D register for single machining time
		5.2	Added the special D register for system time - year and month
		5.2	Added the special D register for system time -day and hour
		5.2	Added the special D register for system time - minute and second
		5.2	Added the special D register for 2nd spindle speed

Release Date	Version	Revised sections	Revision contents
			adjustment
		5.2	Added the special D register for spindle analog voltage output Port 2
		5.2	Added the special D register for lathe tool number selection
		5.2	Added the special D register for spindle analog voltage output Port 1
		5.2	Added the special D register of equal slope compensation filter for X – W axes
		5.2	Added the special D register for linear axis compensation speed
		5.2	Added the special D register for rotation axis compensation speed
		5.2	Added the special D register for 1st spindle speed (written with special D)
		5.2	Added the special D register for arc preview speed
		5.2	Added the special D register for 2nd spindle speed (written with special D)
		5.2	Added the special D register for spindle tool number (written with special D)
		5.2	Added the special D register for standby tool number (written with special D)
		5.2	Added the special D register for command tool number (written with special D)
		5.2	Added the special D register (32-bit) for 1st spindle command speed (32-bit))
		5.2	Corrected the name of the 1st spindle S code data special D
		5.2	Corrected the name of the 1st spindle actual speed special D (32-bit)
		5.2	Added the special D registers for A –W axes machine coordinates
		5.2	Added the special D registers for X –W axes absolute coordinates
		5.2	Added the special D registers for X – W axes DMCNET current monitoring

Release Date	Version	Revised sections	Revision contents
		5.2	Added the special D registers for SP1 and SP2 DMCNET current monitoring
		5.2	Added the special D registers for workpiece coordinates
		5.2	Added the special D register for actual speed (32-bit) of Spindle 2
		5.2	Added the special D register for command speed (32-bit) of Spindle 2
		5.2	Added the special D register for S code data of Spindle 2
		5.2	Added the special D registers for ADC, TAD, and DAC
		5.2	Added the special D registers for user-defined hardware signals
		5.2	Added the special D registers for residual coordinate of X – W axes
		5.2	Added the special D registers for servo tool magazines 1 – 3 command tool
		5.2	Added the special D registers (word) for servo tool magazines 1 - 3
		5.2	Added the special D registers for servo tool magazines 1 - 3 feedback tool number
		5.3	Re-categorize the special M relays and special registers by their functions
		6	Restructured the chapter and added application examples
July, 2017	V3.0		
		5.2	Cancels the function of M1073 G31 MLC input
		5.2	Added the activation condition description for M1075 macro call activation
		5.2	Modified the function names of M1118 and M1119
		5.2	Corrected the special D descriptions in M1172 and M1173
		5.2	Added the special M relays for MLC UVW axes motion trigger
		5.2	Added the description of M1194 MLC axis incremental motion command

Release Date	Version	Revised sections	Revision contents
		5.2	Added the special M relays for forward jog operation of UVW axes
		5.2	Added the special M relays for reverse jog operation of UVW axes
		5.2	Added the special M relays for homing trigger of UVW axes
		5.2	Added the special M relays for cancelling 1st software limits of UVW axes
		5.2	Added the special M relays for UVW axes lock
		5.2	Added the special M relays for UVW axes Servo Off
		5.2	Corrected the description of M2142 and M2143
		5.2	Corrected the description of M2150
		5.2	Added the descriptions for M2168 – M2178 (Port 7 – Port 9 axis signals)
		5.2	Added the special M relays for UVW axes homing complete
		5.2	Added the special M relays for UVW axes positioned at 2nd reference point
		5.2	Added the special M relays for MLC UVW axes positioning complete
		5.2	Added the special M relays for UVW axes in motion
		5.2	Corrected the corresponding functions of M2896 – M2927
		5.2	Corrected descriptions of M2992 - M2995 teach function
		5.2	Update the special M summary table
		5.3	Added the MLC axis positioning special D for U, V, and W axes
		5.3	Added the MLC axis positioning speed special D for U, V, and W axes
		5.3	Modified the description of D1372
		5.3	Modified the description of D1376
		5.3	Modified the description of D1379
		5.3	Update the special D summary table
		6.7	Added the spindle forward operation example

Release Date	Version	Revised sections	Revision contents
		6.8	Added the spindle forward/reverse operation example
		6.9	Added the example of cycle start and feedhold
		6.10	Added the MLC initial M example
		6.11	Added the mode switch examples
		6.12	Added the example of MPG magnification switch
		6.13	Added the example of triggering the Jog operation
		6.14	Added the example of using MPG for axis selection
		6.15	Added the example of homing of each system axis
		6.16	Added the example of cancelling the limits
		6.17	Added the MPG simulation example
		6.18	Added the example of M, S, and T codes execution complete
		6.19	Added the examples of single block execution and optional stop
		6.20	Added the examples of Jog feed rate and cutting feed rate override
		6.21	Added the rapid traverse example
		6.22	Added the spindle speed override example
		6.23	Added the MLC tool magazine function example
		6.24	Added the rapid traverse example
September, 2016	V2.0	1.2.2	Add section introduction and new notes.
		1.3.2	Add section introduction and notes for using output contacts.
		1.4	Add section introduction.
		1.4.1	Add new notes for using auxiliary relays.
		1.5	Add section introduction.
		1.6	Add section introduction.
		1.7	Add section introduction.
		1.7.1	Delete the example of 32-bit counter.
		1.8.1	Add the example of assigning D0 as the 32-bit register. The data register types are changed to 4 types.

Release Date	Version	Revised sections	Revision contents
		1.8.2	Add new table of the 32-bit indirect index registers.
		1.9	Add section introduction and amend range of indicators for external interruption.
		2.1	Add section introduction. Add new column “Execution speed (us)” to the tables in this section. Delete notes.
		2.2	Add section introduction.
		3.1	Delete the column of Model and Page in the table.
		3.2	Add section introduction.
		3.3	Add section introduction. Add descriptions to the instruction of assigning continuous numbers.
		3.4	Add section introduction. Add introduction of how to use registers.
		3.5	Delete the column of Model and Page in the table.
		4	Change the table format.
		4.1	The table contents of API-00 have been changed. The table in example 3 has been added with new column of “C78, C79”.
		4.1	Amend the table in the supplementary notes of API-05, which explains the flags that disable the insertion of interruption indicators in NC series models
		5	New chapter.
		6	New chapter.
July, 2013	V1.0	-	-

For more information about NC Series MLC Application Manual, please refer to:

(1) NC Series Command Guidelines

(2) Delta CNC Solution NC Series User Manual for Operation and Maintenance