Digitized Automation for a Changing World

# Delta CNC Machine Solution
# NC5 Series - MLC Application Manual

www.deltaww.com

# 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 NC5 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
- 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

- Connect the ground terminals to a Class 3 ground system. Ground resistance should not exceed 100 Ω. Improper grounding may result in communication error, electric shock, or fire.

### Operation

WARNING

- Use the MLCEditor software to correctly configure the I/O functions, or it may cause abnormal operation.
- Properly set the parameters before operating the machine, or it may cause abnormal operation or malfunction.
- Ensure the emergency stop works properly and avoid operating the machine without protection.

STOP

- Do not change the wiring when the power is on, or it may cause electric shock or personnel injury.

## Maintenance and Inspection

- Do not touch the internal part of the controller when the power is on, or it may cause electric shock.
- Do not touch the wiring terminals within 10 minutes after turning off the power, or the residual voltage may cause electric shock.
- Cut off the power before replacing the backup battery. Ensure to check the system settings again after replacing the battery.
- Do not block the ventilation holes when operating the controller, or poor heat dissipation may lead to controller malfunction.

## Wiring Method

- Power: connect a 24 $V_{DC}$ power to the controller and do the wiring according to the specifications to avoid danger.
- Wire selection: use stranded wires and multi-core shielded-pair wires for all signal cables.
- The local I/O and remote I/O of the controller require an external 24 $V_{DC}$ power supply to output and input signals normally.

## Wiring of Communication Circuit

- Ensure the wiring between the controller and the servo drive is firmly connected, or loose connection may result in abnormal operation.

For the differences among the various versions, please refer to DELTA's website for the latest information (https://downloadcenter.deltaww.com/en-US/DownloadCenter).

(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 MLC Special M Relay and Special D Register

# B Special Memory

# 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

<table>
<tr><th>Type</th><th>Device</th><th colspan="2">Item</th><th colspan="2">Quantity</th><th>Setting range</th></tr>
<tr><td rowspan="7">Relay<br>(Bit)</td><td>X</td><td colspan="2">External input relay</td><td>0 ~ 511</td><td>512<br>points in total</td><td>I/O</td></tr>
<tr><td>Y</td><td colspan="2">External output relay</td><td>0 ~ 511</td><td>512<br>points in total</td><td>I/O</td></tr>
<tr><td>M</td><td colspan="2">Auxiliary relay</td><td>0 ~ 65,535</td><td>65,535<br>points in total</td><td>I/O</td></tr>
<tr><td>A</td><td colspan="2">Alarm</td><td>0 ~ 511</td><td>512<br>points in total</td><td>I/O</td></tr>
<tr><td>T</td><td colspan="2">Timer</td><td>0 ~ 255</td><td>256<br>points in total</td><td>I/O、Word</td></tr>
<tr><td rowspan="2">C</td><td rowspan="2">Counter</td><td>16-bit</td><td>0 ~ 63</td><td rowspan="2">80<br>points in total</td><td rowspan="2">I/O、Word</td></tr>
<tr><td>32-bit</td><td>64 ~ 79</td></tr>
<tr><td rowspan="6">Register<br>(Word)</td><td>T</td><td>Timer</td><td>16-bit</td><td>0 ~ 255</td><td>256<br>points in total</td><td>0 ~ 65,535</td></tr>
<tr><td rowspan="2">C</td><td rowspan="2">Counter</td><td>16-bit</td><td>0 ~ 63</td><td rowspan="2">80<br>points in total</td><td>0 ~ 65,535</td></tr>
<tr><td>32-bit</td><td>64 ~ 79</td><td>-2,147,483,648 ~<br>+2,147,483,647</td></tr>
<tr><td>D</td><td>Data<br>register</td><td>16-bit</td><td>0 ~ 65,535</td><td>65,535<br>points in total</td><td>-32,768 ~ +32,767</td></tr>
<tr><td>V</td><td>Index<br>register</td><td>16-bit</td><td>0 ~ 7</td><td>8<br>points in total</td><td>-32,768 ~ +32,767</td></tr>
<tr><td>Z</td><td>Index<br>register</td><td>16-bit</td><td>0 ~ 7</td><td>8<br>points in total</td><td>-32,768 ~ +32,767</td></tr>
<tr><td>Indicator</td><td>P</td><td colspan="2">Jump indicator</td><td>0 ~ 255</td><td>256<br>points in total</td><td>-</td></tr>
<tr><td>Constant</td><td>K</td><td colspan="2">Decimal</td><td>N/A</td><td>N/A</td><td>-32,768 ~ +32,767<br>(16-bit range)<br>-2,147,483,648 ~<br>+2,147,483,647<br>(32-bit range)</td></tr>
<tr><td>Float</td><td>F</td><td colspan="2">Floating-point</td><td>N/A</td><td>N/A</td><td>-99,999.999 ~<br>+99,999.999</td></tr>
</table>

## 1.1.2 Settings of MLC devices

| Device name | Function and range | | | | | Total points |
|---|---|---|---|---|---|---|
| Input signal X (Bit) | On board | MPG | Reserved | Machine operation panel I/O | Remote I/O | 512 |
| | X0 ~ X31 | X32 ~ X39 | X40 ~ X63 | X64 ~ X255 | X256 ~ X511 | |
| Output signal Y (Bit) | On board | Reserved | | Machine operation panel I/O | Remote I/O | 512 |
| | Y0 ~ Y31 | Y32 ~ Y63 | | Y64 ~ Y255 | Y256 ~ Y511 | |
| Auxiliary relay M (Bit) | General purpose | Reserved | MLC → System (NC) | Special M (MLC) | System (NC) → MLC | 65,536 |
| | M0 ~ M14999 | M15000 ~ M19999 | M20000 ~ M28999 | M29000 ~ M29999 | M30000 ~ M38999 | |
| | Special M | Reserved | MLC → System (HMI) | Reserved | System (HMI) → MLC | |
| | M39000 ~ M39999 | M40000 ~ M48999 | M49000 ~ M49999 | M50000 ~ M58999 | M59000 ~ M59999 | |
| | Retentive M | Reserved | N/A | N/A | N/A | |
| | M60000 ~ M61999 | M62000 ~ M65535 | N/A | N/A | N/A | |
| User define Alarm A (Bit) | A0 ~ A511 | | | | | 512 |
| Timer T — Bit | T0 ~ T199 (Unit：100 ms)<br>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)<br>C0 - C63 | | 32-bit (counting up / down)<br>C64 ~ C79 | | | |
| | Counting range<br>0 – 65,535 | | Counting range<br>-2,147,483,648 ~ +2,147,483,647 | | | |
| Data register D (Word) | General purpose | Reserved | MLC → System (NC) | Special D (MLC) | System (NC) → MLC | 65,536 |
| | D0 ~ D9999 | D10000 ~ D19999 | D20000 ~ D28999 | D29000 ~ D29999 | D30000 ~ D38999 | |
| | Special D | Reserved | MLC → System (HMI) | Reserved | System (HMI) → MLC | |
| | D39000 ~ D39999 | D40000 ~ D48999 | D49000 ~ D49999 | D50000 ~ D58999 | D59000 ~ D59999 | |
| | Retentive D | Reserved | N/A | N/A | N/A | |
| | D60000 ~ D61999 | D62000 ~ D65535 | N/A | N/A | N/A | |

| Device name | Function and range | | Total points |
|---|---|---|---|
| Register V (Word) | V0 ~ V7 (-32,768 ~ +32,767) | | 8 |
| Register Z (Word) | Z0 ~ Z7 (-32,768 ~ +32,767) | | 8 |
| Jump indicator P | For CJ and CALL: P0 - P255 | | 256 |
| Constant K | Decimal constant | -32,768 ~ +32,767 (16-bit operation) | N/A |
| | | -2,147,483,648 ~ +2,147,483,647 (32-bit operation) | N/A |
| Floating-point number F | Floating-point with 3 decimal point places | -99,999.999 ~ +99,999.999 | 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.

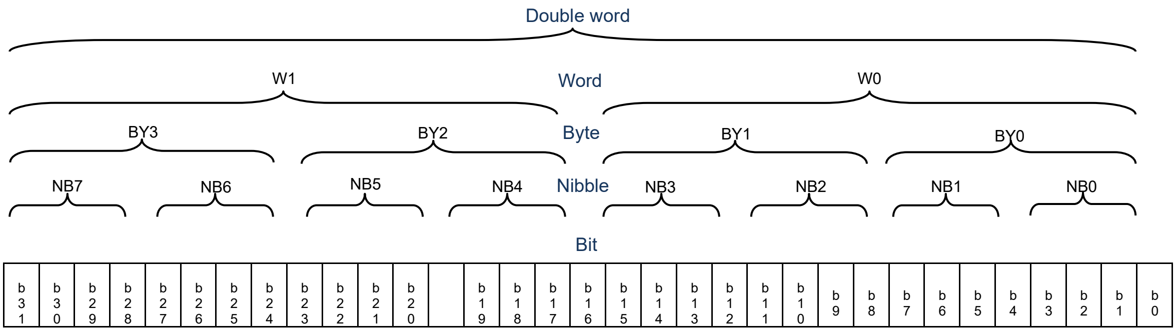



Figure 1.2.1.1 Relationship between nibble, byte, word, and double word in a binary system

1

### 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 ~ X511.
     Numbers for external output devices: Y0 ~ Y511.
   - Numbers for devices of M, A, T, C, D, V, Z, and P, 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 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 Digital input / output relays (X, Y)

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, and remote I/O.

<table>
<tr><th colspan="6">Digital Inputs / Outputs</th></tr>
<tr><th colspan="2">Interface</th><th colspan="2">Digital Inputs X</th><th colspan="2">Digital Outputs Y</th></tr>
<tr><td colspan="2">On-board I/O</td><td>X0 ~ X31</td><td rowspan="12">Total 512</td><td>Y0 ~ Y31</td><td rowspan="12">Total 512</td></tr>
<tr><td colspan="2">MPG</td><td>X32 ~ X39</td><td>-</td></tr>
<tr><td colspan="2">Reserved</td><td>X40 ~ X63</td><td>Y32 ~ Y63</td></tr>
<tr><td colspan="2">I/O on panel</td><td>X64 ~ X255</td><td>Y64 ~ Y255</td></tr>
<tr><td rowspan="8">Remote I/O (EtherCAT)</td><td>Station 1</td><td>X256 ~ X287</td><td>Y256 ~ Y287</td></tr>
<tr><td>Station 2</td><td>X288 ~ X319</td><td>Y288 ~ Y319</td></tr>
<tr><td>Station 3</td><td>X320 ~ X351</td><td>Y320 ~ Y351</td></tr>
<tr><td>Station 4</td><td>X352 ~ X383</td><td>Y352 ~ Y383</td></tr>
<tr><td>Station 5</td><td>X384 ~ X415</td><td>Y384 ~ Y415</td></tr>
<tr><td>Station 6</td><td>X416 ~ X447</td><td>Y416 ~ Y447</td></tr>
<tr><td>Station 7</td><td>X448 ~ X479</td><td>Y448 ~ Y479</td></tr>
<tr><td>Station 8</td><td>X480 ~ X511</td><td>Y480 ~ Y511</td></tr>
</table>

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

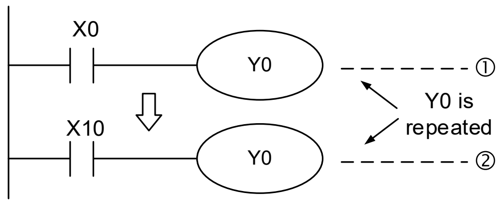
### 1.3.1 Functions of digital input / output relays

The input / output relays can control the logic in the MLC actions. The following describes the relay functions and state changes.

1. Input of X devices: 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 of Y devices: 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 type 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 Auxiliary relay (M)

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

| Auxiliary relay (M) | | | |
|---|---|---|---|
| Definition | Address range | Points | Retentive |
| General purpose | M0 ~ M14999 | 15,000 | - |
| Reserved | M15000 ~ M19999 | 5,000 | - |
| System special M<br>MLC → System (NC) | M20000 ~ M28999 | 9,000 | - |
| MLC special M | M29000 ~ M29999 | 1,000 | - |
| System special M<br>System (NC) → MLC | M30000 ~ M38999 | 9,000 | - |
| System reserved special M | M39000 ~ M39999 | 1,000 | - |
| Reserved | M40000 ~ M48999 | 9,000 | - |
| HMI special M<br>MLC → System (HMI) | M49000 ~ M49999 | 1,000 | - |
| Reserved | M50000 ~ M58999 | 10,000 | - |
| HMI special M<br>System (HMI) → MLC | M59000 ~ M59999 | 1,000 | - |
| Retentive M | M60000 ~ M61999 | 2,000 | ✓ |
| Reserved | M62000 ~ M65535 | 3,536 | - |

## 1.4.1 Functions of the auxiliary relay

Same as the output relay Y, the auxiliary relay M has output coil and A & B type 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:
   Range between M0 to M1499, 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. Retentive auxiliary relay:
   Range between M60000 to M61999, 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:
   Range between M15000 to M59999, for exchanging status or signals of NC, MLC and HMI. Each special relay has its own specific function. Do not use the undefined ones.

## 1.5 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) | | | |
|---|---|---|---|
| Definition | Address range | Points | Retentive |
| General purpose | A0 - A511 | 512 | - |

### 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 type 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 Timer (T)

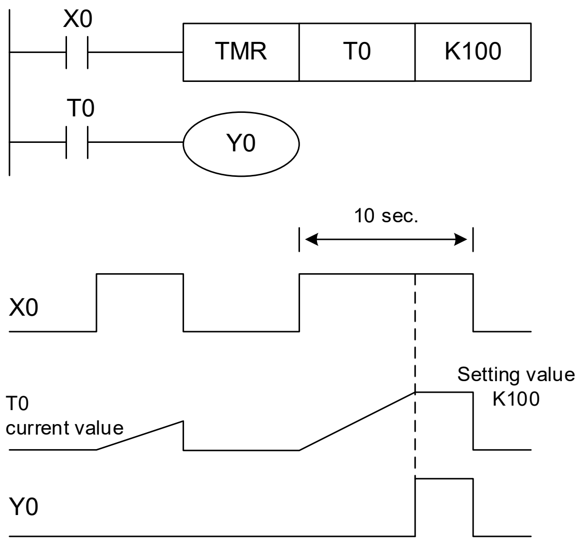
Users can use these timer to count with specific I/O actions to have the device execute the programmed action after being triggered for a specific period of time when programming the MLC.

| Timer (T) | | | | |
|---|---|---|---|---|
| Definition | Unit | Address range | Points | Retentive |
| General purpose | 100 ms | T0 - T199 | 200 | - |
| General purpose | 10 ms | T200 - T255 | 56 | - |

## 1.6.1 Functions of the timer

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 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 x 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

## 1.7 Counter (C)

Users 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.

| Counter (C) | | | | |
|---|---|---|---|---|
| Definition | Unit | Address range | Points | Retentive |
| Up counter | 16-bit | C0 - C63 | 64 | - |
| Up counter | 32-bit | C64 - C77 | 16 | - |

| Item | 16-bit counter | 32-bit counter |
|---|---|---|
| Type | General purpose | |
| Count direction | Counting up | |
| Setting value | 0 to 65,535 | -2,147,483,648 to +2,147,483,647 |
| Type of setting value | Constant K or register D | Constant K or register D<br>(when assign to register D, system will take two registers at once) |
| Value changing | Stop counting once the setting value is reached | |
| Output contact | The contact is ON and remains ON once the setting value is reached | |
| Reset | The RST or ZRST 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 will remain as ON. Counter can use the constant K or the data register D as the setting value

The actual setting value of the counter.

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

Register D: indirectly specify the data in register D as the setting value. When call by 32-bit counter, system will occupy two consecutive registers at once

- 16-bit counter: C0 - C63

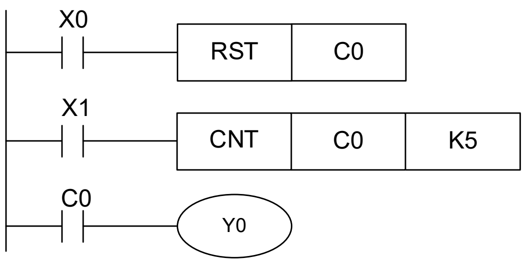
The setting range of the 16-bit counter is K0 - K65,535. When the setting value is K0 or K1, the output C contact is ON upon the first count.

- 32-bit counter: C64 - C79

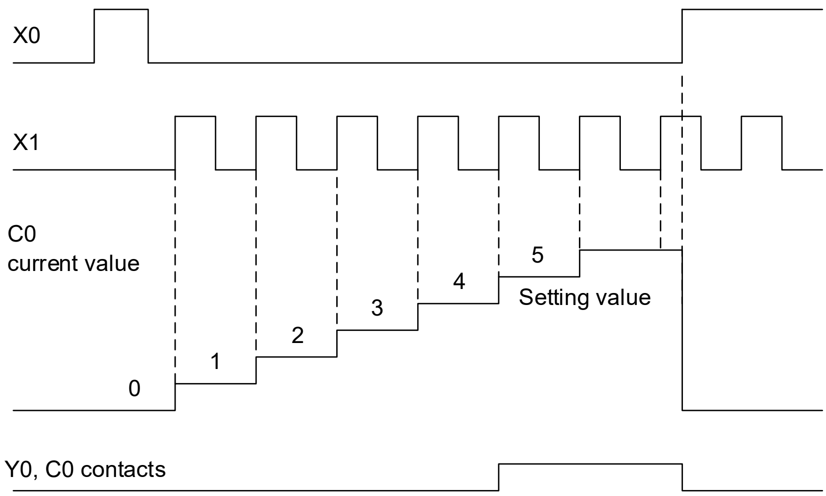
The setting range of the 32-bit counter is K-2,147,483,648 to K+2,147,483,647. When the setting value is K0 or K1, the output C contact is ON upon the first count. To use the data register D, two consecutive data registers are required for one setting value.

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



# 1.8 Registers (D, V, Z)

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. Users 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.

## 1.8.1 Data register (D)

Data registers are divided into the three 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. Retentive 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 retentive 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.

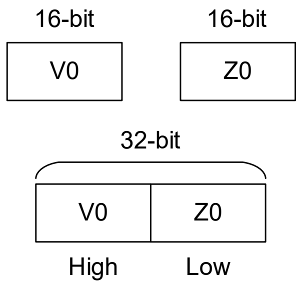
| Data register (D) | | | |
|---|---|---|---|
| Definition | Address range | Points | Retentive |
| General purpose | D0 ~ D9999 | 10,000 | - |
| Reserved | D10000 ~ D19999 | 10,000 | - |
| MLC → System (NC) | D20000 ~ D28999 | 9,000 | - |
| MLC special D | D29000 ~ D29999 | 1,000 | - |
| System (NC) → MLC | D30000 ~ D38999 | 9,000 | - |
| System reserved special D | D39000 ~ D39999 | 1,000 | - |
| Reserved | D40000 ~ D48999 | 9,000 | - |
| MLC → System (HMI) | D49000 ~ D49999 | 1,000 | - |
| Reserved | D50000 ~ D58999 | 10,000 | - |
| System (HMI) → MLC | D59000 ~ D59999 | 1,000 | - |
| Retentive D | D60000 ~ D61999 | 2,000 | ✓ |
| Reserved | D62000 ~ D65535 | 3,536 | - |

## 1.8.2 Index registers (V, Z)

Index registers are defined as below table.

| Index registers (V, Z) | | | | |
|---|---|---|---|---|
| Definition | Unit | Address range | Points | Retentive |
| V register | 16-bit | V0 – V7 | 8 | - |
| Z register | 16-bit | Z0 - Z7 | 8 | - |

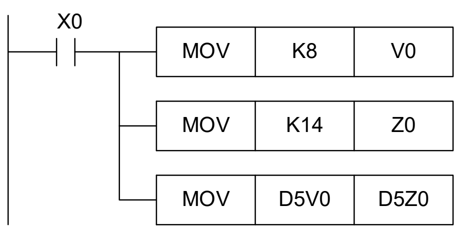
Same as general data registers, V and Z registers are 16-bit data registers which can be read and written. As for 32-bit data, users have to assign the V registers; because of Z registers will be included in the V registers in this case. Thus, Z registers can no longer be used, otherwise the contents 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

## 1.9 Indicators (P)

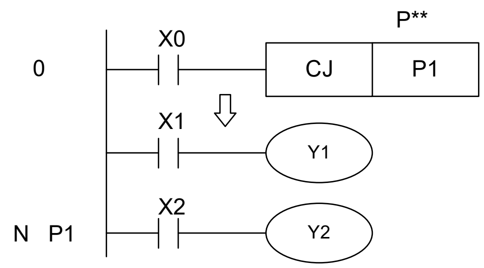
MLC has indicator P, which customer can use to have the MLC run the specified programs when programming, reducing the error caused by the scan time of MLC.

| Indicator registers (P) | | | | |
|---|---|---|---|---|
| Definition | API | Address range | Points | Retentive |
| P register | For CJ and CALL | P0 – P255 | 256 | - |

The indicator P used with the application instructions API-00 CJ and API-01 CALL. For detailed information, refer to the descriptions of CJ and CALL instructions in Chapter 4.

Example 1

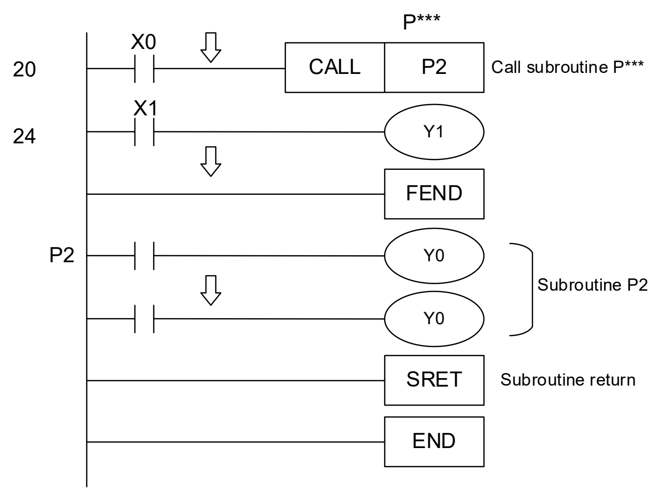
When X0 is ON, the program automatically jumps from address 0 to address N (the specified label P1) and continues to execute, skipping the addresses in the middle.
When X0 is OFF, the program starts from address 0 and continues executing the programs in sequence without executing the CJ instruction.




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.

When X0 is OFF, the program will not execute the CALL and jump to line P2. As regular program process, it will continue with address 24.



# 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 includes 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 | Serials connect contact A | X, Y, M, A, T, C | - | 1 - 2 |
| ANI | Serials 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 | Serials 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 |

■ 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 |
| 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 |

## 2.2 Description of basic instructions

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

### 2.2.1 General instructions

#### LD: Load contact A

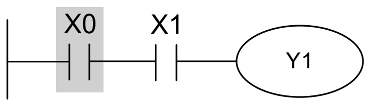
| Instruction | Function | Applicable model |
|---|---|---|
| LD | Load contact A | NC series |

| Operand | | | | | | | |
|---|---|---|---|---|---|---|---|
| X | Y | M | A | T | C | D | 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

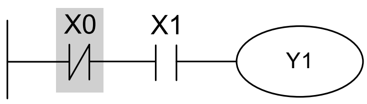
| Instruction | Function | Applicable model |
|---|---|---|
| LDI | Load contact B | NC series |

| Operand | | | | | | | |
|---|---|---|---|---|---|---|---|
| X | Y | M | A | T | C | D | 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

2

## AND: Serial connect contact A

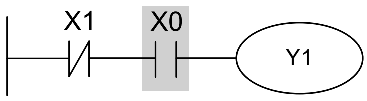
| Instruction | Function | Applicable model |
|---|---|---|
| AND | Serials connect contact A | NC series |

| Operand | | | | | | | |
|---|---|---|---|---|---|---|---|
| X | Y | M | A | T | C | D | 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

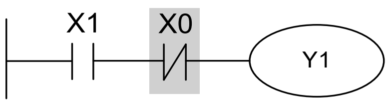
| Instruction | Function | Applicable model |
|---|---|---|
| ANI | Serials connect contact B | NC series |

| Operand | | | | | | | |
|---|---|---|---|---|---|---|---|
| X | Y | M | A | T | C | D | 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

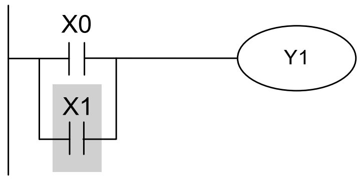
| Instruction | Function | Applicable model |
|---|---|---|
| OR | Parallel connect contact A | NC series |

| Operand | | | | | | | |
|---|---|---|---|---|---|---|---|
| X | Y | M | A | T | C | D | 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

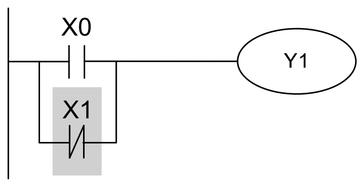
| Instruction | Function | Applicable model |
|---|---|---|
| ORI | Parallel connect contact B | NC series |

| Operand | | | | | | | |
|---|---|---|---|---|---|---|---|
| X | Y | M | A | T | C | D | 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

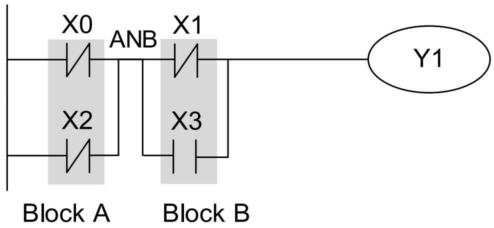
| Instruction | Function | Applicable model |
|---|---|---|
| ANB | Serials connect circuit block | NC series |

| Operand | | | | | | | |
|---|---|---|---|---|---|---|---|
| X | Y | M | A | T | C | D | 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

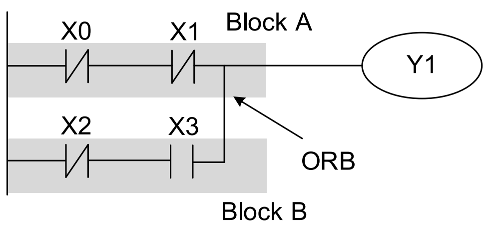
| Instruction | Function | Applicable model |
|---|---|---|
| ORB | Parallel connect circuit block | NC series |

| Operand | | | | | | | |
|---|---|---|---|---|---|---|---|
| X | Y | M | A | T | C | D | 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 | | | | | | | |
|---|---|---|---|---|---|---|---|
| X | Y | M | A | T | C | D | 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 | | | | | | | |
|---|---|---|---|---|---|---|---|
| X | Y | M | A | T | C | D | 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

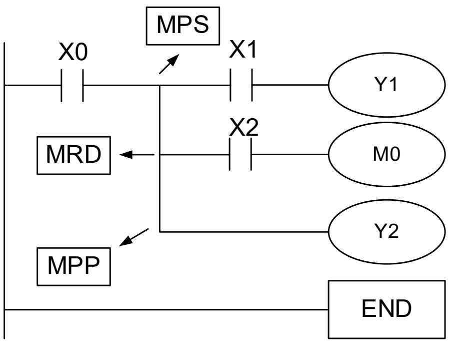
| Instruction | Function | Applicable model |
|---|---|---|
| MPP | Retrieve from stack | NC series |

| Operand | | | | | | | |
|---|---|---|---|---|---|---|---|
| X | Y | M | A | T | C | D | 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

2

## 2.2.2 Output instructions

### OUT: Drive coil

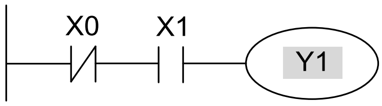
| Instruction | Function | Applicable model |
|---|---|---|
| OUT | Drive coil | NC series |

| Operand | | | | | | | |
|---|---|---|---|---|---|---|---|
| X | Y | M | A | T | C | D | 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)

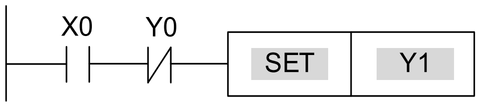
| Instruction | Function | Applicable model |
|---|---|---|
| SET | Action remains (ON) | NC series |

| Operand | | | | | | | |
|---|---|---|---|---|---|---|---|
| X | Y | M | A | T | C | D | 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 |

2

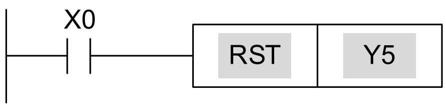
| Operand | | | | | | | |
|---|---|---|---|---|---|---|---|
| X | Y | M | A | T | C | D | V, Z |
| - | ● | ● | ● | ● | ● | ● | ● |

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

### 2.2.3 Timer and counter

#### TMR: 16-bit timer

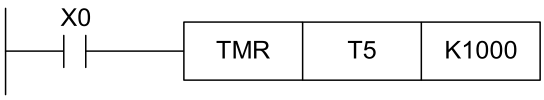
| Instruction | Function | Applicable model |
|---|---|---|
| TMR | 16-bit timer | NC series |

| Operand | |
|---|---|
| T-K | T0 - T255, K0 - K65,535 |
| T-D | T0 - T255, D0 - D65,535 |

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

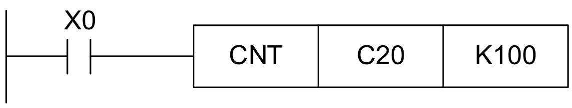
| Instruction | Function | Applicable model |
|---|---|---|
| CNT | 16-bit counter | NC series |

| Operand | |
|---|---|
| C-K | C0 - C63, K0 - K65,535 |
| C-D | C0 - C63, D0 - D65,535 |

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

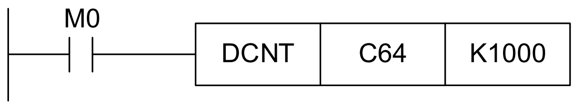
| Instruction | Function | Applicable model |
|---|---|---|
| DCNT | 32-bit counter | NC series |

| Operand | |
|---|---|
| C-K | C64 - C79, K-2,147,483,648 to K+2,147,483,647 |
| C-D | C64 - C79, D0 – D65,535 |

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

2 2

### LDP: Start rising edge detection

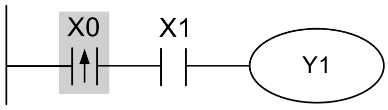
| Instruction | Function | Applicable model |
|---|---|---|
| LDP | Start rising edge detection | NC series |

| Operand | | | | | | | |
|---|---|---|---|---|---|---|---|
| X | Y | M | A | T | C | D | 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

## 2.2.4 Contact rising / falling edge detection instructions

### LDF: Start falling edge detection

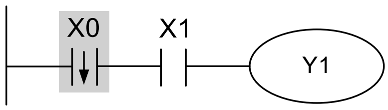
| Instruction | Function | Applicable model |
|---|---|---|
| LDF | Start falling edge detection | NC series |

| Operand | | | | | | | |
|---|---|---|---|---|---|---|---|
| X | Y | M | A | T | C | D | 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

2

## ANDP: Serial connection of rising edge detection

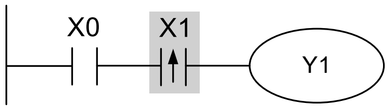
| Instruction | Function | Applicable model |
|---|---|---|
| ANDP | Serial connection of rising edge detection | NC series |

| Operand | | | | | | | |
|---|---|---|---|---|---|---|---|
| X | Y | M | A | T | C | D | 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

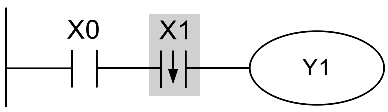
| Instruction | Function | Applicable model |
|---|---|---|
| ANDF | Serial connection of falling edge detection | NC series |

| Operand | | | | | | | |
|---|---|---|---|---|---|---|---|
| X | Y | M | A | T | C | D | V, Z |
| ● | ● | ● | ● | ● | ● | - | - |

Description:

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

Example:




ANDF (X1) ladder diagram

2

## ORP: Parallel connection of rising edge detection

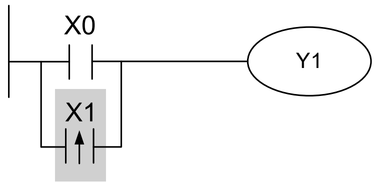
| Instruction | Function | Applicable model |
|---|---|---|
| ORP | Parallel connection of rising edge detection | NC series |

| Operand | | | | | | | |
|---|---|---|---|---|---|---|---|
| X | Y | M | A | T | C | D | 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

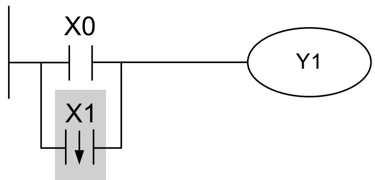
| Instruction | Function | Applicable model |
|---|---|---|
| ORF | Parallel connection of falling edge detection | NC series |

| Operand | | | | | | | |
|---|---|---|---|---|---|---|---|
| X | Y | M | A | T | C | D | V, Z |
| ● | ● | ● | ● | ● | ● | - | - |

Description:

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

Example:




ORF (X0, X1) ladder diagram

## 2.2.5 Upper and lower differential output instructions

### PLS: Upper differential output

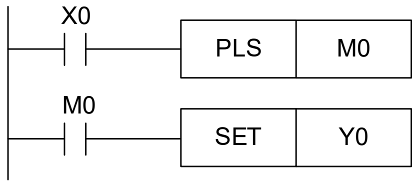
| Instruction | Function | Applicable model |
|---|---|---|
| PLS | Upper differential output | NC series |

| Operand | | | | | | | |
|---|---|---|---|---|---|---|---|
| X | Y | M | A | T | C | D | V, Z |
| ● | ● | ● | ● | ● | ● | - | - |

Description:

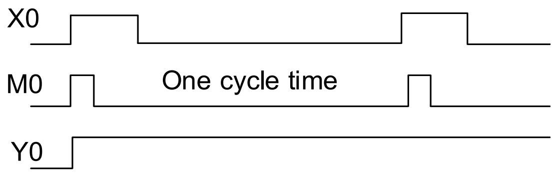
The PLS instruction is for the upper differential output. When X0 switches from Off to ON (rising edge triggered), the PLS instruction executes and M0 sends one pulse with the length of one scanning time cycle.

Example:




PLS (M0) ladder diagram

Timing diagram:




PLS (M0) timing diagram

### PLF: Lower differential output

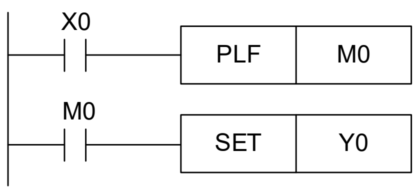
| Instruction | Function | Application model |
|---|---|---|
| PLF | Lower differential output | NC series |

| Operand | | | | | | | |
|---|---|---|---|---|---|---|---|
| X | Y | M | A | T | C | D | V, Z |
| ● | ● | ● | ● | ● | ● | - | - |

Description:

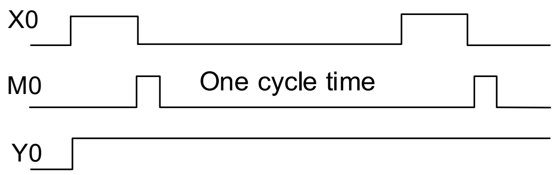
The PLF instruction is for the lower differential output. When X0 switches from On to Off (falling edge triggered), the PLF instruction executes and M0 sends one pulse with the length of one scanning time cycle.

Example:




PLF (M0) ladder diagram

Timing diagram:




PLF (M0) timing diagram

## 2.2.6 End instruction

### END: Program ends

| Instruction | Function | Applicable model |
|---|---|---|
| END | Program ends | NC series |

| Operand | | | | | | | |
|---|---|---|---|---|---|---|---|
| X | Y | M | A | T | C | D | 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.

### 2.2.7 Other instructions

#### NOP: No operation

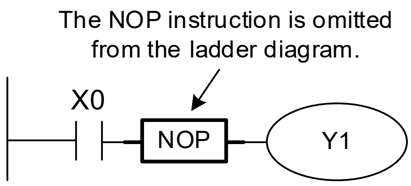
| Instruction | Function | Applicable model |
|---|---|---|
| NOP | No operation | NC series |

| Operand | | | | | | | |
|---|---|---|---|---|---|---|---|
| X | Y | M | A | T | C | D | 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

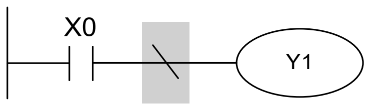
| Instruction | Function | Applicable model |
|---|---|---|
| INV | Invert the operation result | NC series |

| Operand | | | | | | | |
|---|---|---|---|---|---|---|---|
| X | Y | M | A | T | C | D | 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

2

## P: Indicator

| Instruction | Function | Applicable model |
|---|---|---|
| P | Indicator | NC series |

| Operand |
|---|
| P0 - P255 |

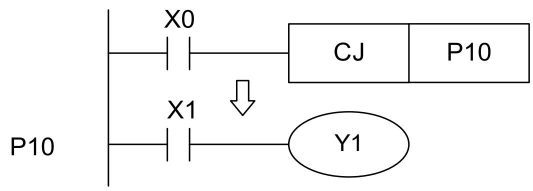
2

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

# MLC Application Instructions Overview

# 3

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

3

## 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 operands | Function | STEP (S) | |
|---|---|---|---|---|---|---|---|
| | | 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 | - |
| | 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 |
| 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 | - |
| 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 | - |

| Type | API | Instruction code | | Number of operands | Function | STEP | |
|---|---|---|---|---|---|---|---|
| | | 16-bit | 32-bit | | | 16-bit | 32-bit |
| 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

## 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 Table format of application instructions

(1) (2) (3) (4) (5) (6)

| 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 | | | | | | | | * | * | * | * | * | * | * | * |

(7) (8) (9)

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: 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.

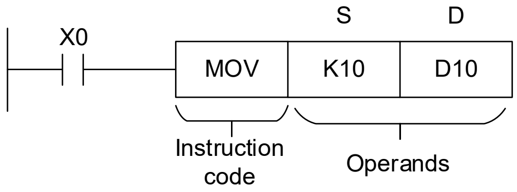
The instruction name indicates the function, and the operand indicates the devices for the operation of this instruction.

The instruction name typically occupies 1 step and each one of its operands 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.

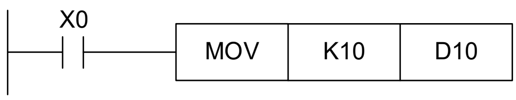
Destination operand D: if there is more than one destination operand, use $D_1$, $D_2$, and so on to represent the destination operands.

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)

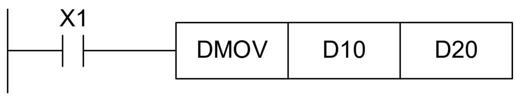
Depending on the content in the operand, the length of an operand can be 16 bits or 32 bits. 16-bit and 32-bit instructions are for processing data of different lengths. The letter “D” is prefixed to a 16-bit instruction to form a 32-bit instruction.

16-bit MOV instruction




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

32-bit DMOV instruction

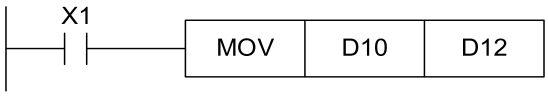



When X1 = On, operand (D11, D10) is sent to operand (D21, D20).

### 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

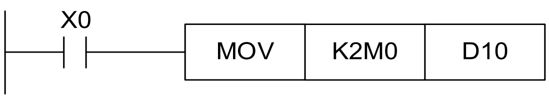
The specified object of the operand has the following features.

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

## 3.3 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.

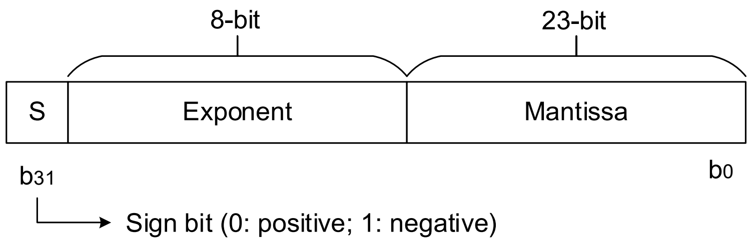
Note: if K4Y0 is used in a 32-bit operation, the upper 16 bits are regarded as 0. For 32-bit data, use K8Y0.

- 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: $(-1)^S \times 2^{E-B} \times 1.M$ (B = 127)
Range of values expressed by 32-bit floating point: $\pm 2^{-126}$ to $\pm 2^{+128}$ or $\pm 1.1755\times 10^{-38}$ to $\pm 3.4028\times 10^{+38}$.

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

Steps:

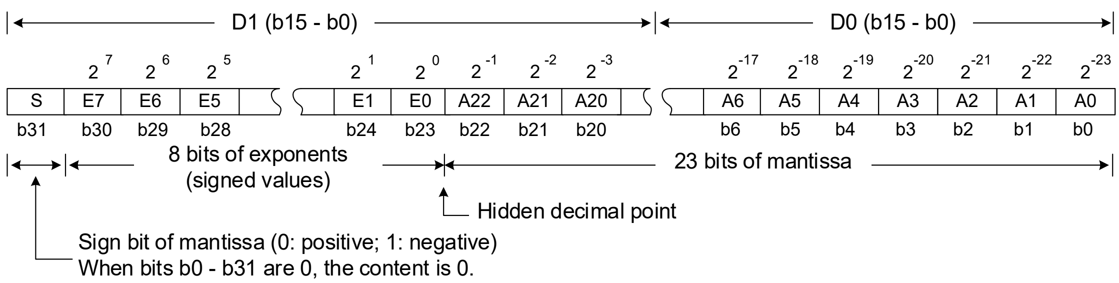
1. Convert 23 into a binary number: 23.0 = 10111.
2. Normalize the binary number: 10111 = 1.0111 × 24. 0111 is the mantissa and 4 is the exponent.
3. Obtain the stored value of the exponent: ∵E-B = 4 → E-127 = 4 ∴E = 131 = $10000011_2$.
4. Combine the sign bit, exponent, and mantissa into a floating-point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

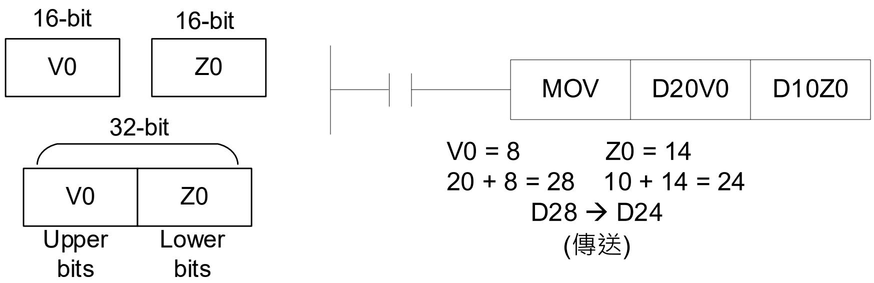
The conversion steps for the floating-point format of -23.0 and 23.0 are the same but the sign bit is changed to 1.

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 V and Z are same as regular register D, which are 16-bit type register. They have total 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, when V0 = 8, the D20V0 will represent register D28 (20 + 8). After the condition is met and the MOV instruction execute, data in the register D28 will sent to register D24.

Devices in the NC series that can apply V and Z registers are P, KnX, KnY, KnM, KnA, T, C, and D. Users who would like to set value to V and Z need to pay attention on the instruction's type whether is 16-bit or 32-bit. According to the V and Z are continues memory, therefore Z register is not allowed to use on the 32-bit instruction and not recognized to set Z register separately. Otherwise, the V register's value will be modified.

| V, Z relation when applying 32-bit instruction | |
|---|---|
| V0 | Z0 |
| V1 | Z1 |
| V2 | Z2 |
| V3 | Z3 |
| V4 | Z4 |
| V5 | Z5 |
| V6 | Z6 |
| V7 | Z7 |

## 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 |
| | 34 | ALT | - | On / Off alternation |
| | 45 | AND= | DAND= | S1 = S2 |
| | 46 | AND> | DAND> | S1 > S2 |
| | 47 | AND< | DAND< | S1 < S2 |
| | 48 | AND<> | DAND<> | S1 ≠ S2 |
| | 49 | AND<= | DAND<= | S1 ≦ S2 |
| | 50 | AND>= | DAND>= | S1 ≧ S2 |
| 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 | 16 | DIV | DDIV | BIN division |
| | 18 | DEC | DDEC | Minus one (BIN) |
| | 26 | DECO | - | Decoder |
| E | 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 |
| I | 17 | INC | DINC | Plus one (BIN) |

| Type | API | Instruction code | | Function |
|---|---|---|---|---|
| | | 16-bit | 32-bit | |
| L | 39 | LD= | DLD= | S1 = S2 |
| | 40 | LD> | DLD> | S1 > S2 |
| | 41 | LD< | DLD< | S1 < S2 |
| | 42 | LD<> | DLD<> | S1 ≠ S2 |
| | 43 | LD<= | DLD<= | S1 ≦ S2 |
| | 44 | LD>= | DLD>= | S1 ≧ S2 |
| 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= | S1 = S2 |
| | 52 | OR> | DOR> | S1 > S2 |
| | 53 | OR< | DOR< | S1 < S2 |
| | 54 | OR<> | DOR<> | S1 ≠ S2 |
| | 55 | OR<= | DOR<= | S1 ≦ S2 |
| | 56 | OR>= | DOR>= | S1 ≧ S2 |
| P | 35 | PLS | - | Upper differential output |
| | 38 | PLF | - | Lower differential output |
| R | 23 | ROR | DROR | Rotate right |
| | 24 | ROL | DROL | Rotate left |
| 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 |
| 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 | | | | | | | | | | | | | | | |

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.

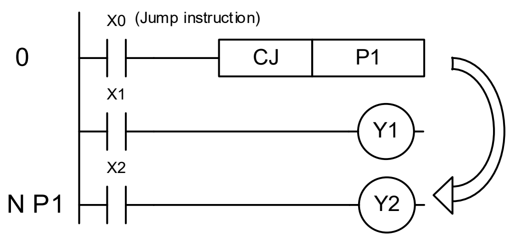
The device actions when the jump instruction is executed:

1. All 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.



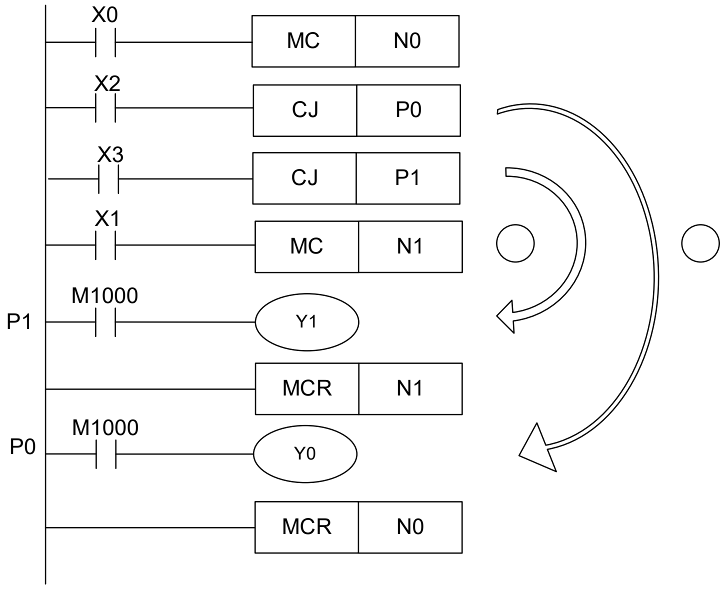
Program example 2:

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.

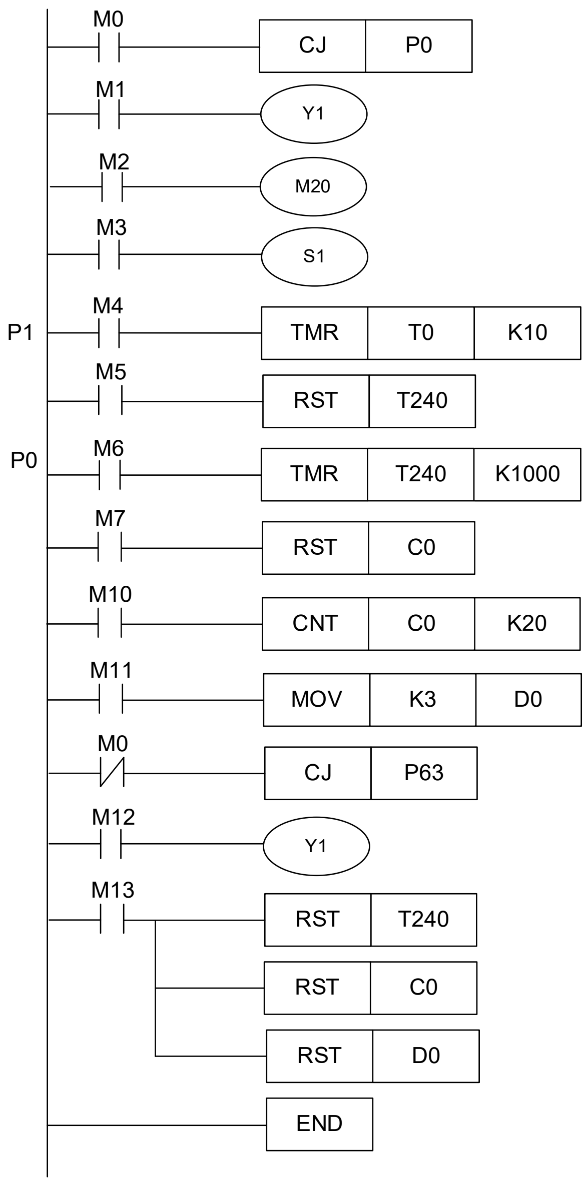
Stage change of each device in the following ladder diagram:

| 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. |

| Device | Contact state before CJ is executed | Contact state during execution of CJ | Output coil state during execution of CJ |
|---|---|---|---|
| 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. | | |
| 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 |
| S | | | | | | | | | | | | | | | |

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 6$^{th}$ 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 |

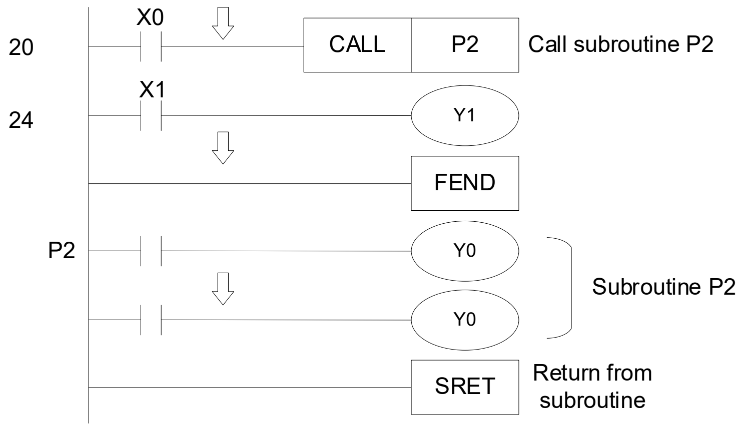
16-bit instruction: SRET continuous execution type (1 step).
32-bit instruction: none.
Flag: none.
Notes on the use of operands: no operand.
Contact for activating the instruction is not required.

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:

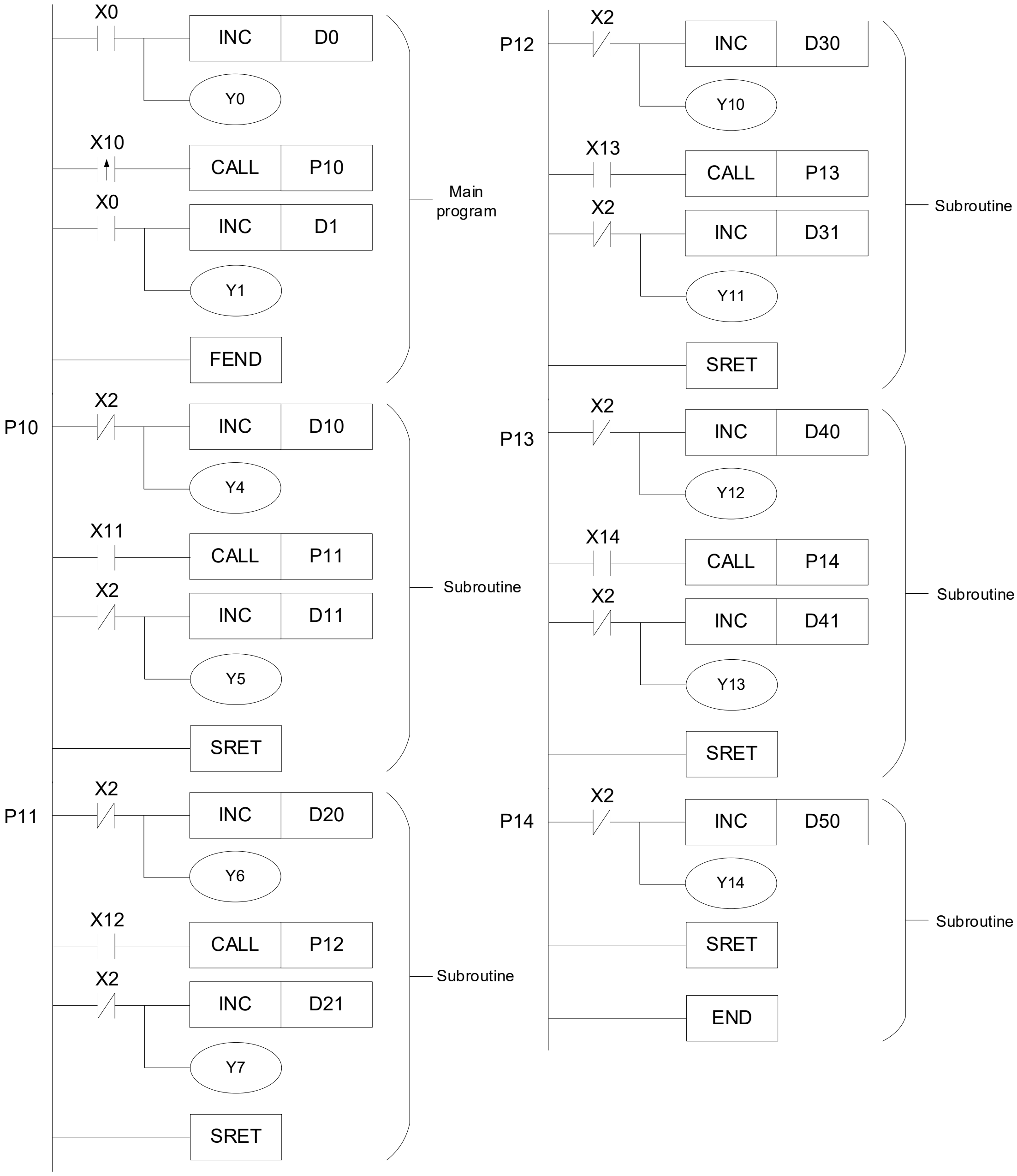
When X0 is ON, execute the CALL instruction and jump to P2 to execute the specified subroutine. When SRET is executed, return to address 24 and then continues to execute what follows.




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.

4


X0
INC D0
Y0
X10
CALL P10
X0
INC D1
Y1
FEND
Main program
P10
X2
INC D10
Y4
X11
CALL P11
X2
INC D11
Y5
SRET
Subroutine
P11
X2
INC D20
Y6
X12
CALL P12
X2
INC D21
Y7
SRET
Subroutine
P12
X2
INC D30
Y10
X13
CALL P13
X2
INC D31
Y11
SRET
Subroutine
P13
X2
INC D40
Y12
X14
CALL P14
X2
INC D41
Y13
SRET
Subroutine
P14
X2
INC D50
Y14
SRET
END
Subroutine

4

## ■ 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.

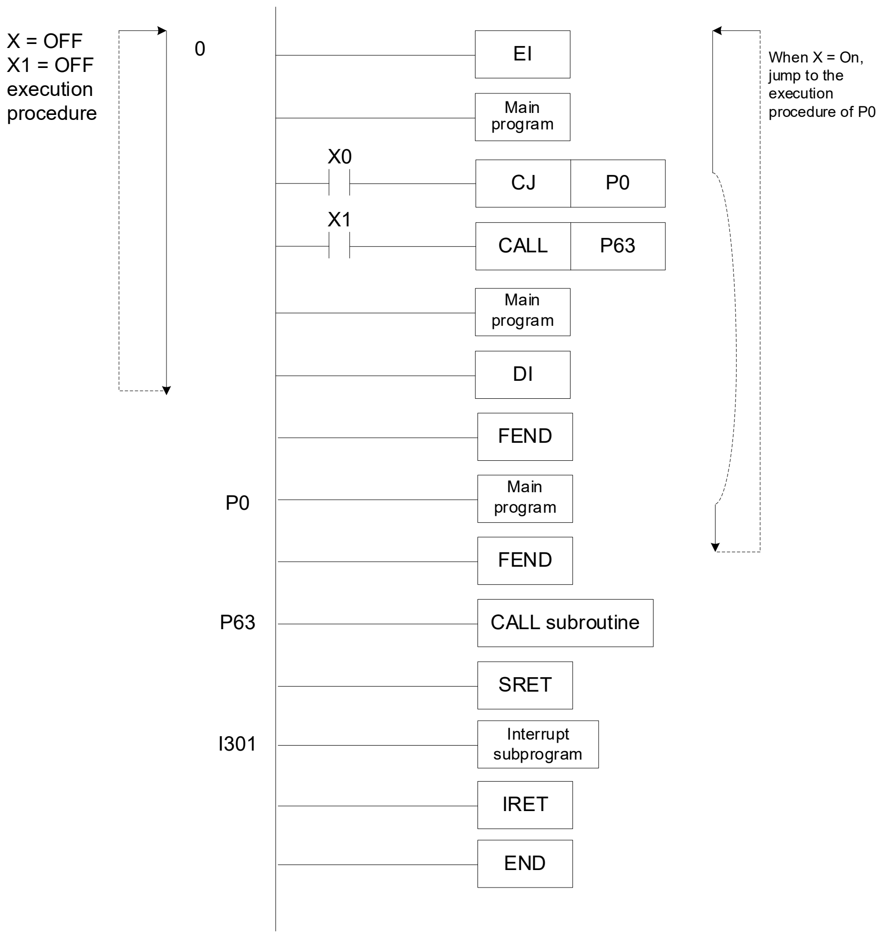
Instruction description:

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:

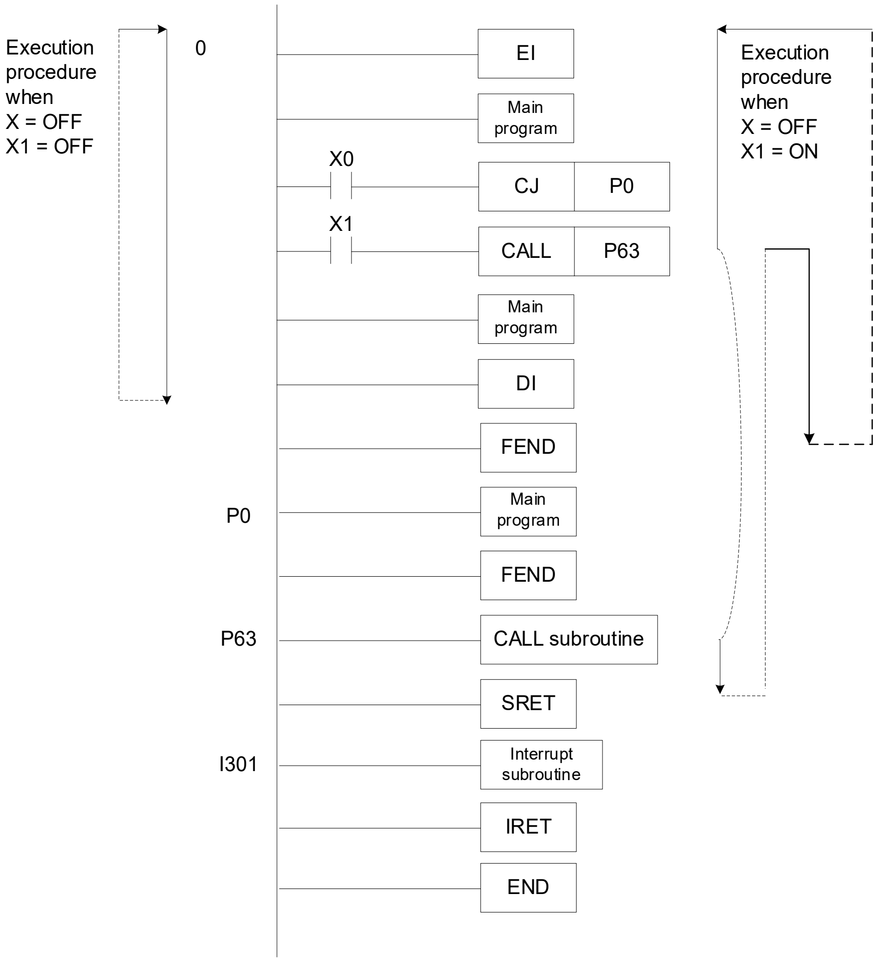
1. After the CALL instruction is executed, FEND is executed before SRET.
2. After the FOR instruction is executed, FEND is executed before NEXT.

Execution procedure of CJ instruction:



4

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 |
| S | | | | | | | | | | | | | | | |

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 times the loop is executed.

## ■ API-08 NEXT: End of nested loop

<table>
<tr><td>API</td><td></td><td rowspan="2">NEXT</td><td colspan="4" rowspan="2">-</td><td colspan="4" rowspan="2">End of nested loop</td><td colspan="5" rowspan="2">NC series</td></tr>
<tr><td>08</td><td>-</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>

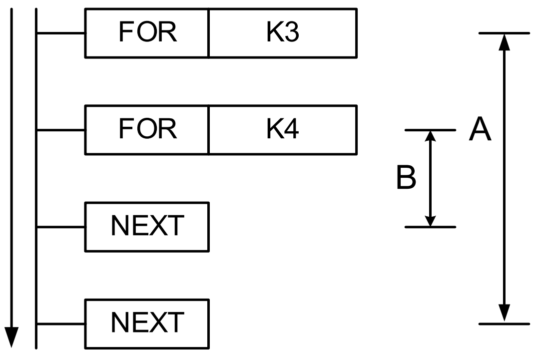
16-bit instruction: NEXT continuous execution type (1 step).
32-bit instruction: none.
Flag: none.
Notes on the use of operands: no operand.
Contact for activating the instruction is not required.

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 $\leqq$ 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.

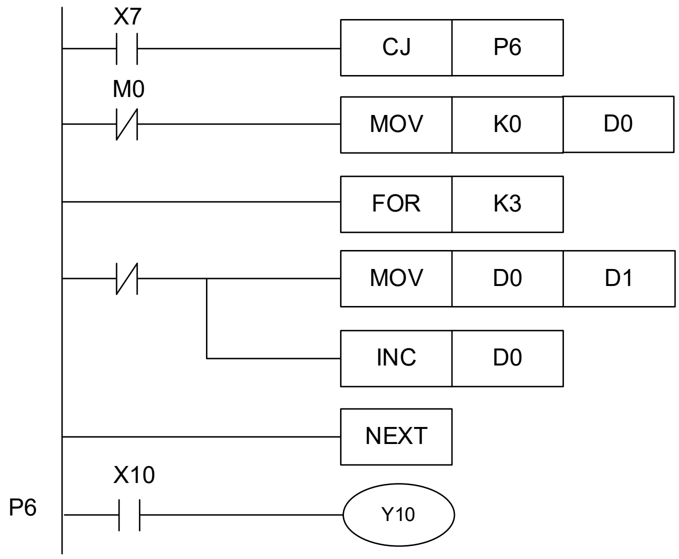
Program example 1:

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.



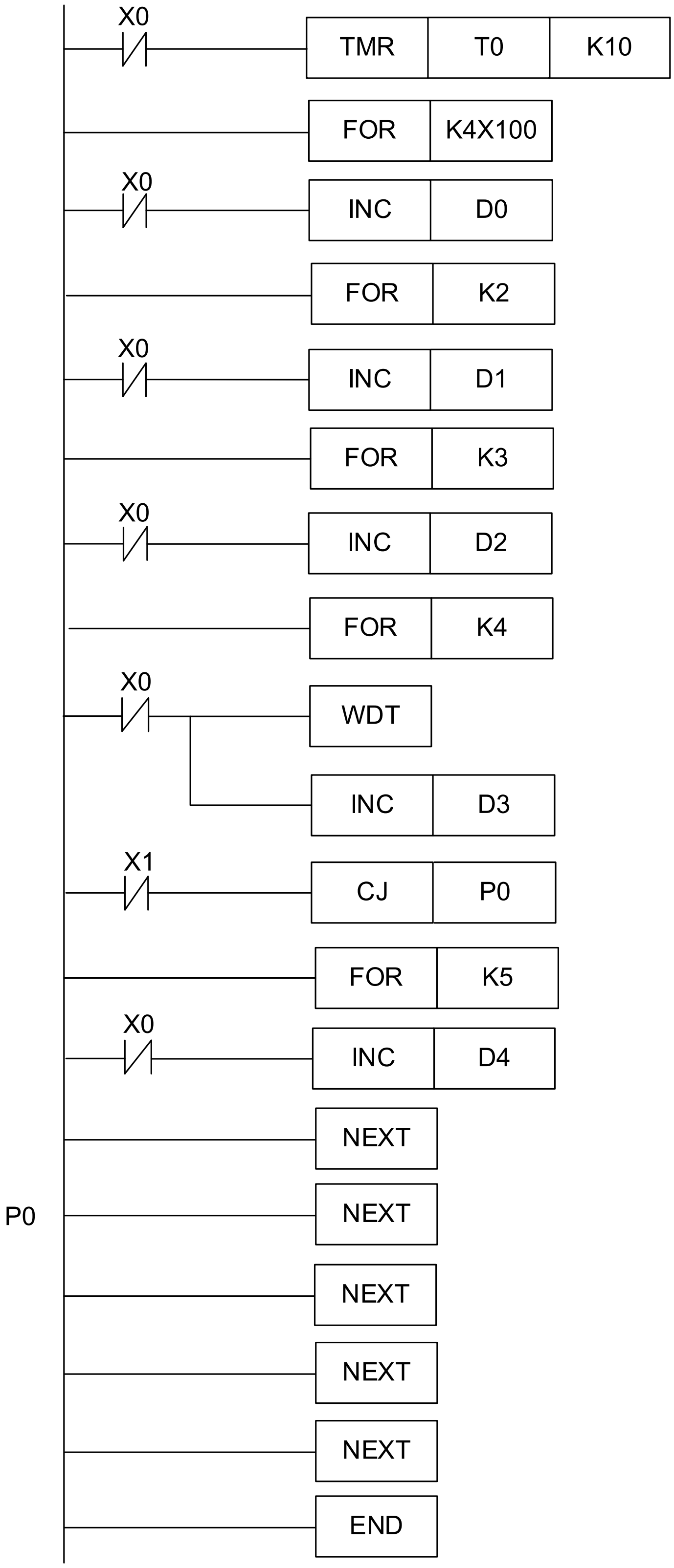
Program example 2:

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

# 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 content 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:

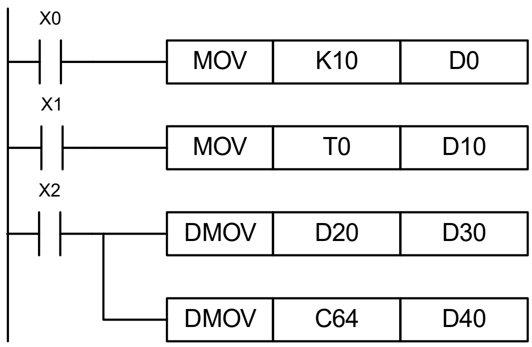
To move the 16-bit data, use the MOV instruction.

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).



4

## ■ 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:

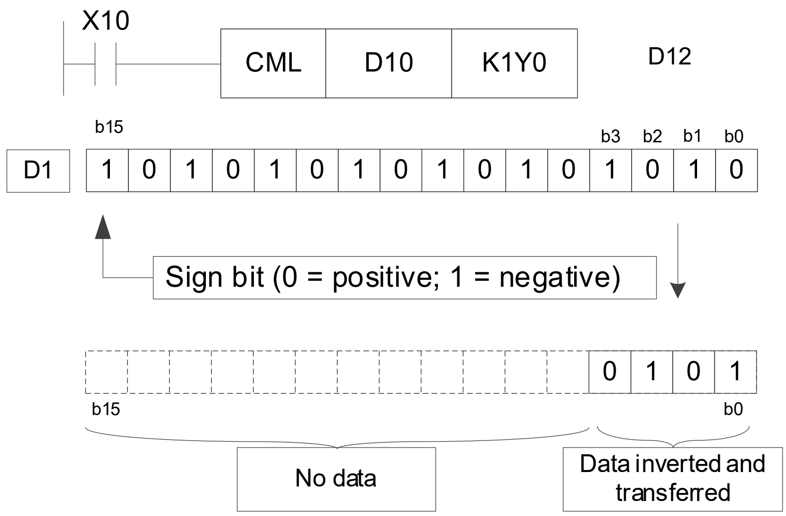
S: data source. D: destination device for the data to transfer to.

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.



4

## ■ 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.

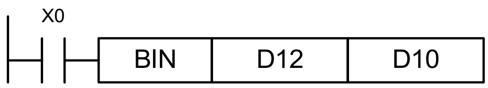
Converts the contents of data source S (BCD: 0 - 9,999) to BIN data and saves the result in D.

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 | S1, S2, 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.

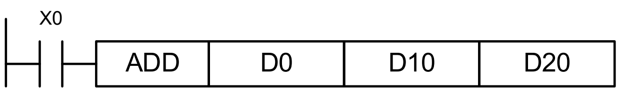
32-bit BIN addition:

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:

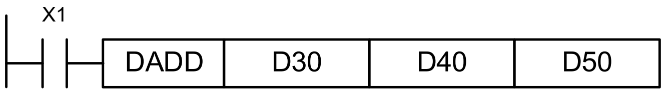
When X0 is ON, add the summand D0 and addend D10 and save the result in D20.



Program example 2:

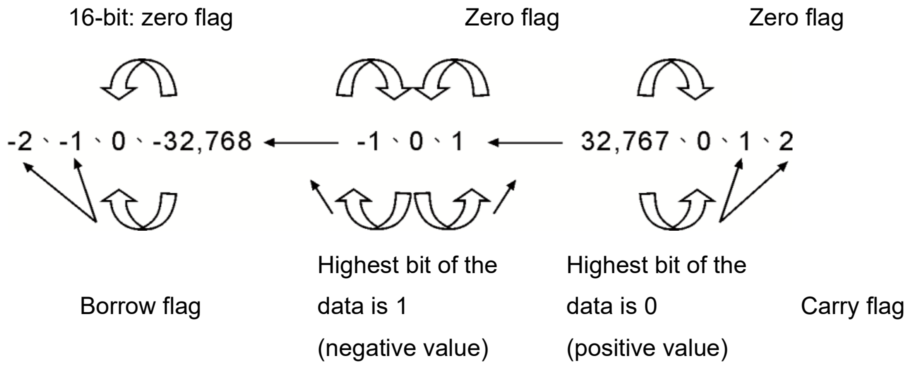
32-bit BIN addition:

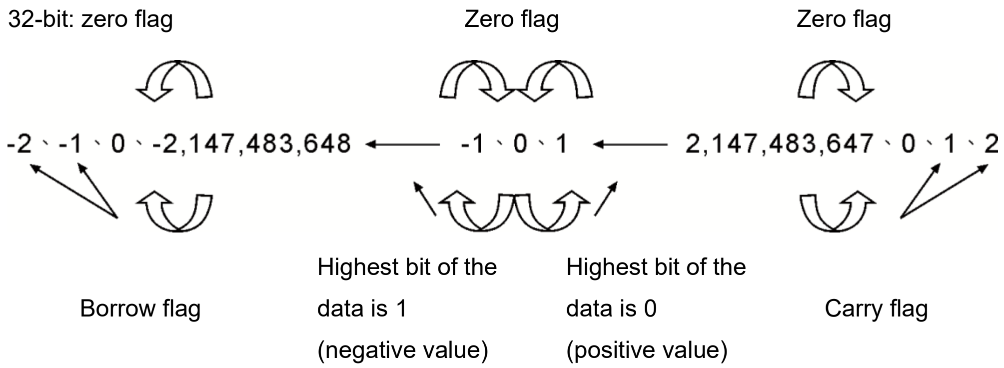
When X1 is ON, add the content in (D31, D30) (summand) and the content in (D41, D40) (addend), and save the sum in (D51, D50). (D30, D40 and D50 are the lower 16-bit data, whereas D31, D41 and D51 are the higher 16-bit data.)




Supplementary note:

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



## ■ 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.

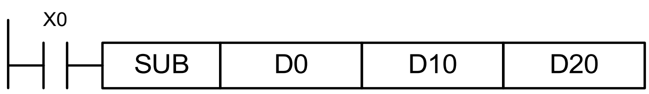
32-bit BIN subtraction:

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.)

X1

| DSUB | D30 | D40 | D50 |
|---|---|---|---|

## ■ API-15 MUL: BIN multiplication

| API | | | | MUL | | S1, S2, 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.

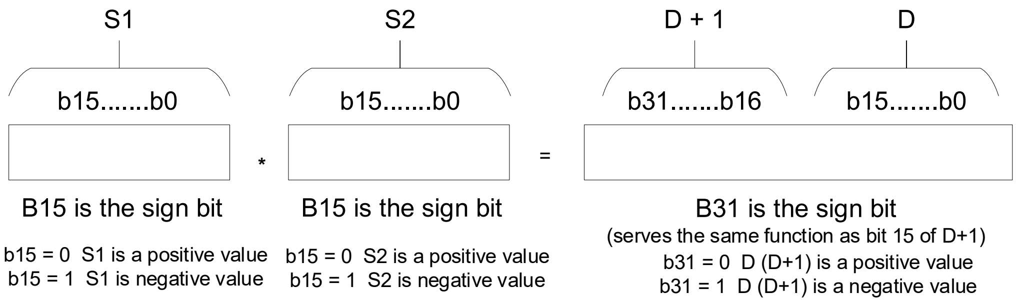
Instruction description:

$S_1$: 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:

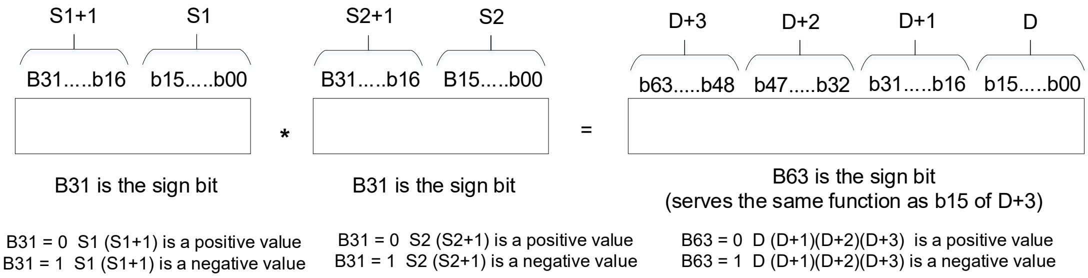
If D serves as a bit device, you can assign K1 - K4 as 16-bit, occupying consecutive 2 sets of 16-bit devices.



4

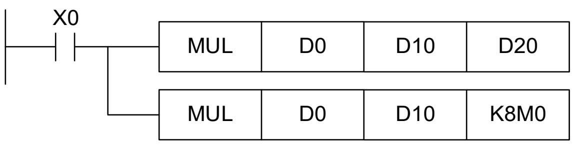
32-bit BIN multiplication:

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




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.

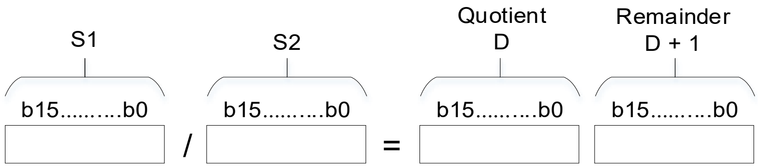
Instruction description:

$S_1$: 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).

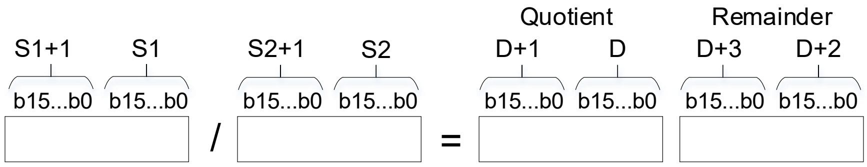
16-bit BIN division:

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.



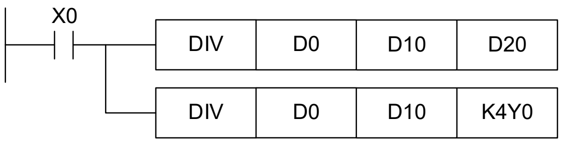

32-bit BIN division:

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.

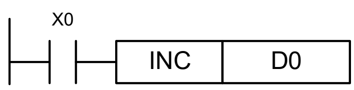
Instruction description:

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:

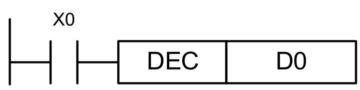
D: destination device

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 | S1, S2, 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 S1, S2 and D are used in register Z, only the16-bit instruction is applicable.

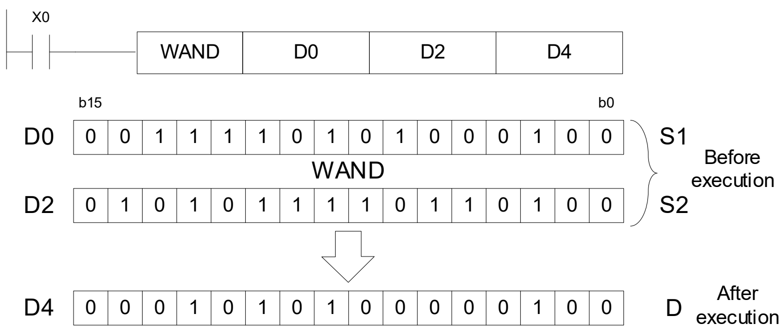
Instruction description:

$S_1$: 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 if $S_1$ or $S_2$ is 0.

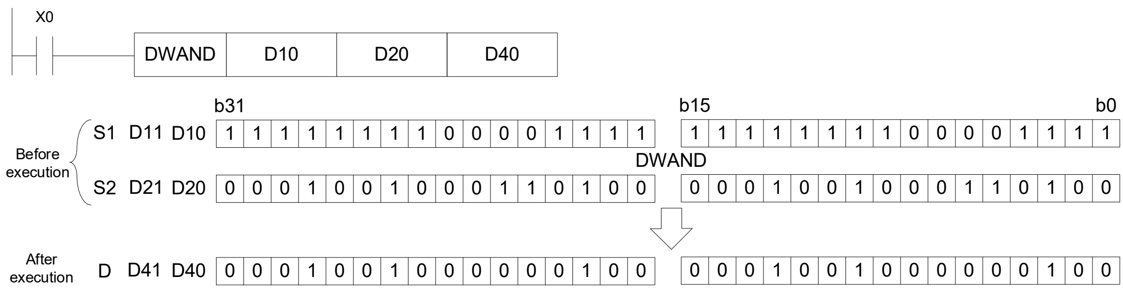
Program example 1:

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




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).



## ■ 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.

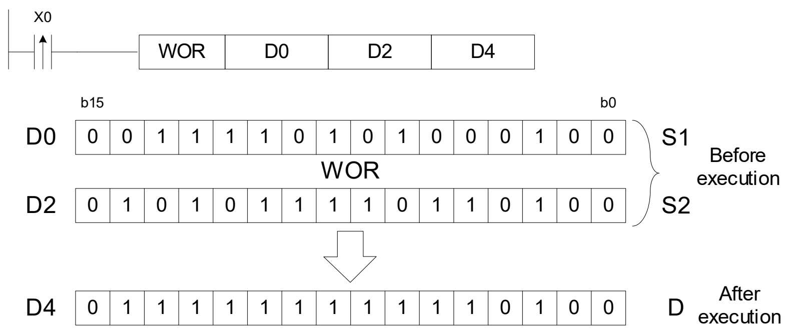
Instruction description:

$S_1$: 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 if $S_1$ or $S_2$ is 1.

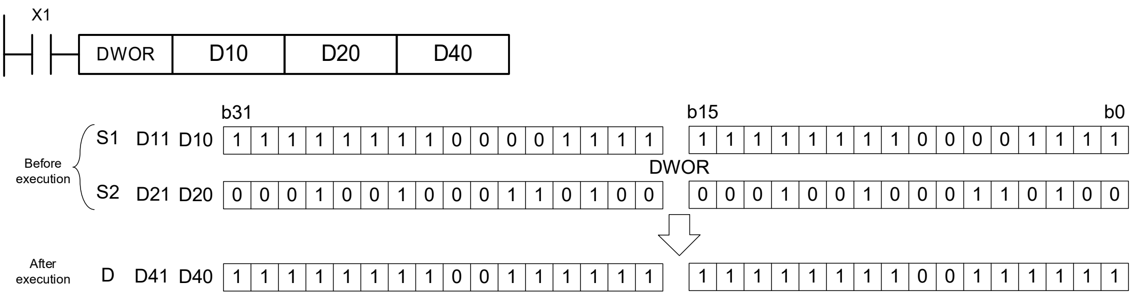
Program example 1:

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




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).



## ■ API-21 WXOR: XOR operation

| API | | WXOR | $S_1$, $S_2$, 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.

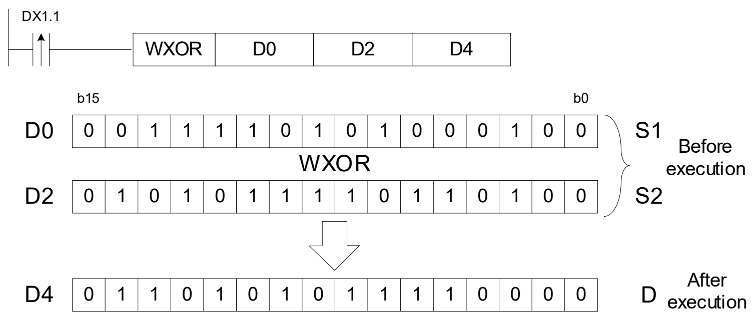
Instruction description:

$S_1$: 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.

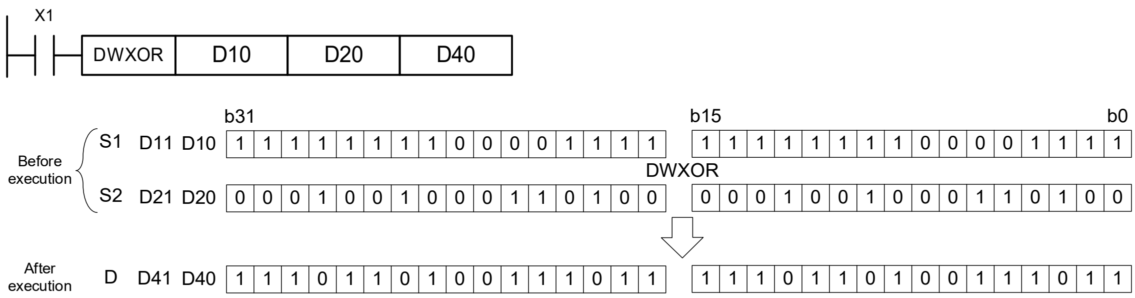
Program example 1:

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




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).



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

| API | | | NEG | | D | | | | Take the negative number (Two's complement) | | | NC series | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 22 | D | | | | | | | | | | | | | | |
| | Bit device | | | | Word device | | | | | | | | | | |
| | X | Y | M | A | K | F | KnX | KnY | KnM | KnA | T | C | D | V | Z |
| D | | | | | | | | | | | | | * | | |

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.

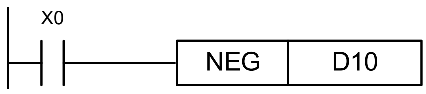
Instruction description:

D: the device requiring two's complement

Converts a negative BIN value to an absolute value.

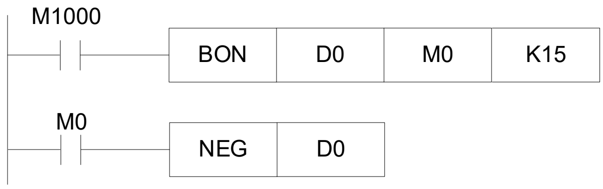
Program example 1:

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:

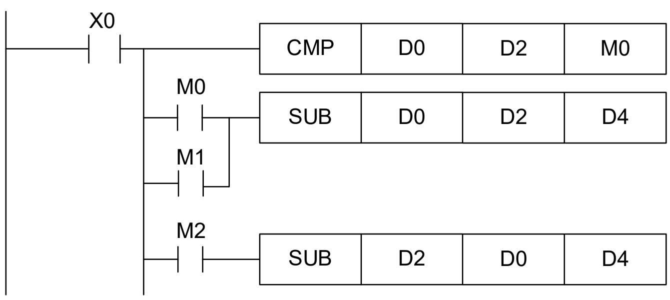
Get the negative absolute value: when the $15^{th}$ bit of D0 is 1, M0 is ON, meaning D0 is a negative value. When M0 is ON, use the NEG instruction to get two's compliment of D0 and then get its absolute value.




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 ⇒ $(\overline{D0}) + 1 = 1$

| | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |

(D0) = -2 ⇒ $(\overline{D0}) + 1 = 2$

| | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |

(D0) = -3 ⇒ $(\overline{D0}) + 1 = 3$

| | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 1 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 |

(D0) = -4 ⇒ $(\overline{D0}) + 1 = 4$

| | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |

(D0) = -5 ⇒ $(\overline{D0}) + 1 = 5$

| | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 1 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 |

.
.
.

(D0) = -32,765 ⇒ $(\overline{D0}) + 1 = 32,765$

| | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 |
| 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 1 |

(D0) = -32,766 ⇒ $(\overline{D0}) + 1 = 32,766$

| | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 |

(D0) = -32,767 ⇒ $(\overline{D0}) + 1 = 32,767$

| | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

(D0) = -32,768 ⇒ $(\overline{D0}) + 1 = -32,768$

| | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 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

| API | | ROR | D, n | | | | Rotate right | | | | NC series | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 23 | 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: 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).

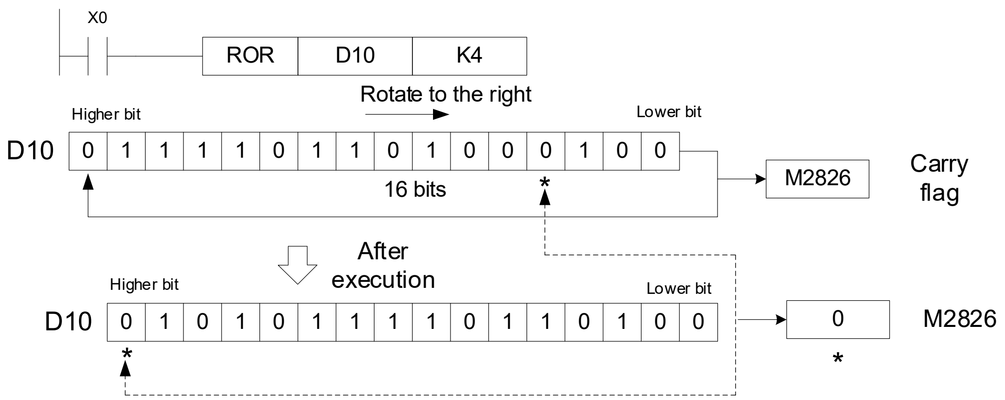
Instruction description:

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 | | | | |

| | X | Y | M | A | K | F | KnX | KnY | KnM | KnA | T | C | D | V | Z |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Bit device | | | | Word device | | | | | | | | | | |
| 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 | D1, D2 | 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 $D_2$.

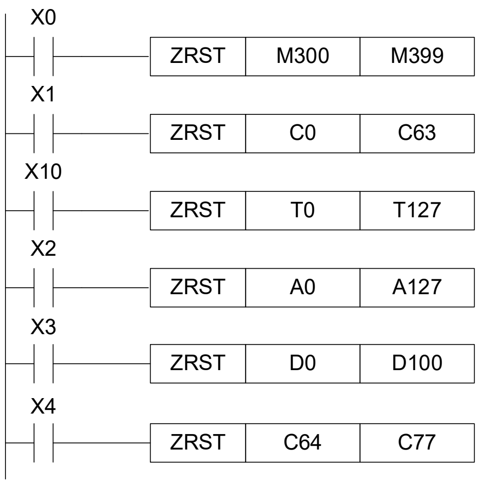
Instruction description:

$D_1$: 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.

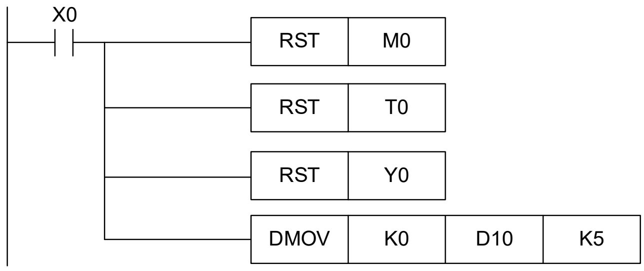
Program example:

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.

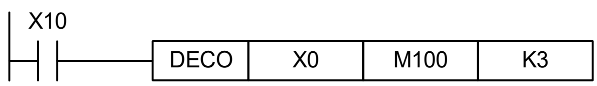
Instruction description:

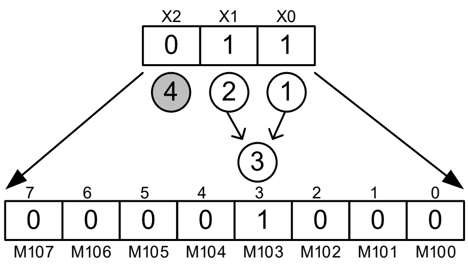
S: source device for decoding; D: device for saving the decoded result; n: length of the decoded bits

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.)



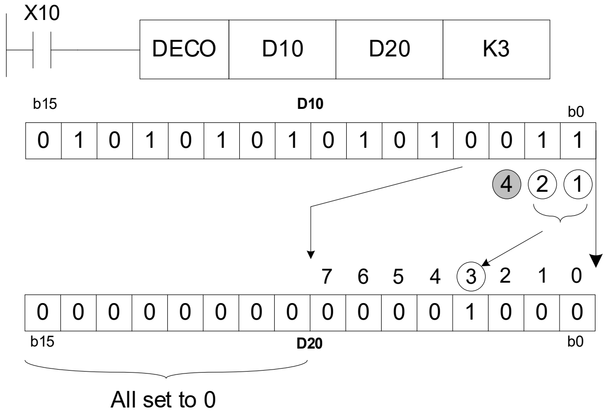

a. When X10 goes from OFF to ON, the DECO instruction decodes values in X0 - X2 to M100 - M107.

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.

4

## ■ 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.

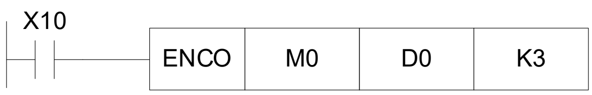
D: device for saving the encoded value.

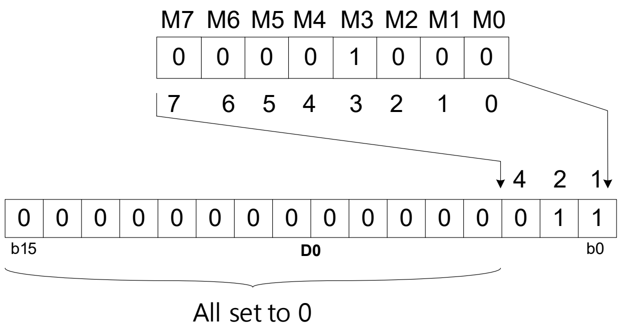
n: length of the encoded bits

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).



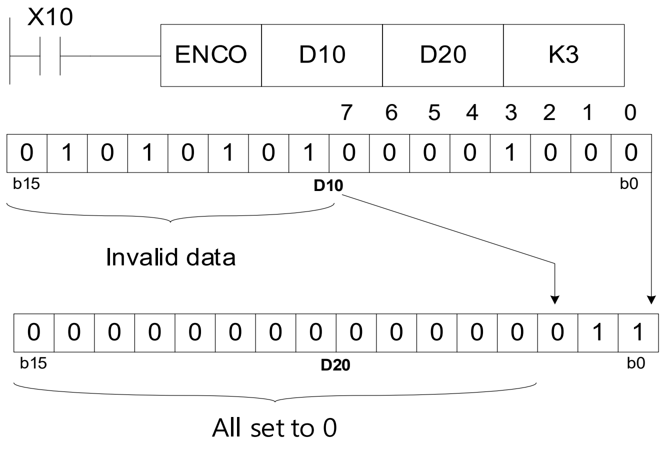




a. When X0 goes from OFF to ON, the ENCO instruction encodes $2^3$ bits data in (M0 - M7) and saves the result in the lower 3 bits (b2 - b0) of D0. The unused bits (b15 - b3) in D0 are all set to 0.

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 | | | | |

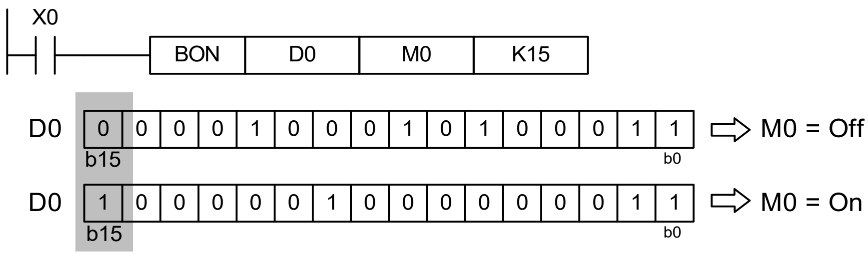
| | Bit device | | | | Word device | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | X | Y | M | A | K | F | KnX | KnY | KnM | KnA | T | C | D | V | Z |
| S | | | | | * | | | | | | * | * | * | | |
| D | | * | * | * | | | | | | | | | | | |
| n | | | | | * | |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

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:




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.

# 4.6 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.

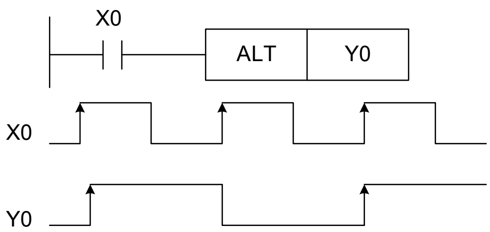
Instruction description:

D: destination device

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

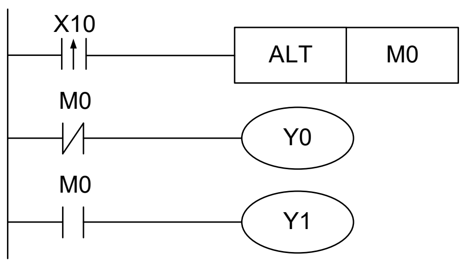
Program example 1:

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.



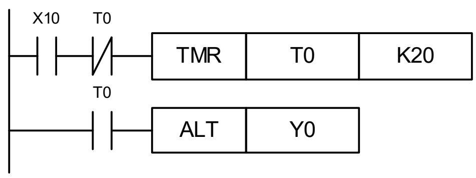

Program example 2:

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.



# 4.7 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 | | | | |

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

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.

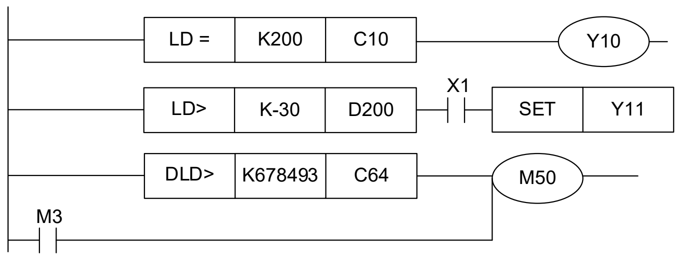
| API No. | 16-bit instruction | 32-bit instruction | Continuity condition | No-continuitycondition |
|---|---|---|---|---|
| 39 | LD = | DLD = | S1 = S2 | S1 ≠ S2 |
| 40 | LD > | DLD > | S1 > S2 | S1 ≦ S2 |
| 41 | LD < | DLD < | S1 < S2 | S1 ≧ S2 |
| 42 | LD < > | DLD < > | S1 ≠ S2 | S1 = S2 |
| 43 | LD < = | DLD < = | S1 ≦ S2 | S1 > S2 |
| 44 | LD > = | DLD > = | S1≧ S2 | S1 < S2 |

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 | | | | |

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

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

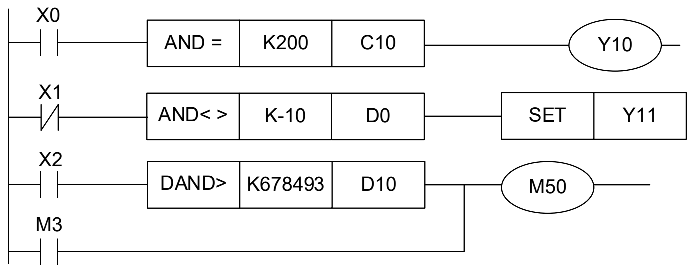
1. Compares the contents in $S_1$ and $S_2$. Take the API-45 (AND=) instruction as an example: when $S_1$ equals $S_2$, the continuity of the instruction is enabled; when $S_1$ does not equal $S_2$, the continuity of the instruction is disabled.
2. AND※ is a comparison instruction that connects to contacts 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 > | $S_1 > S_2$ | $S_1 \leqq S_2$ |
| 47 | AND < | DAND < | $S_1 < S_2$ | $S_1 \geqq S_2$ |
| 48 | AND < > | DAND < > | $S_1 \neq S_2$ | $S_1 = S_2$ |
| 49 | AND < = | DAND < = | $S_1 \leqq S_2$ | $S_1 > S_2$ |
| 50 | AND > = | DAND > = | $S_1 \geqq S_2$ | $S_1 < S_2$ |

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※ | | | S₁, S₂ | | | | | 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

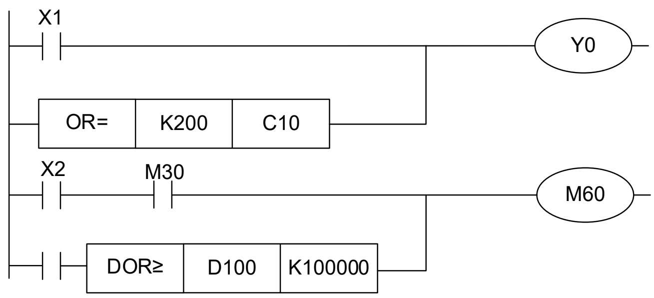
1. Compares contents in $S_1$ and $S_2$. Take the API-51(OR=) instruction as an example. When the $S_1$ equals $S_2$, the continuity of the instruction is enabled; when $S_1$ does not equal $S_2$, the instruction is disabled.
2. OR※ is a comparison instruction that connects to contacts in parallel, as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

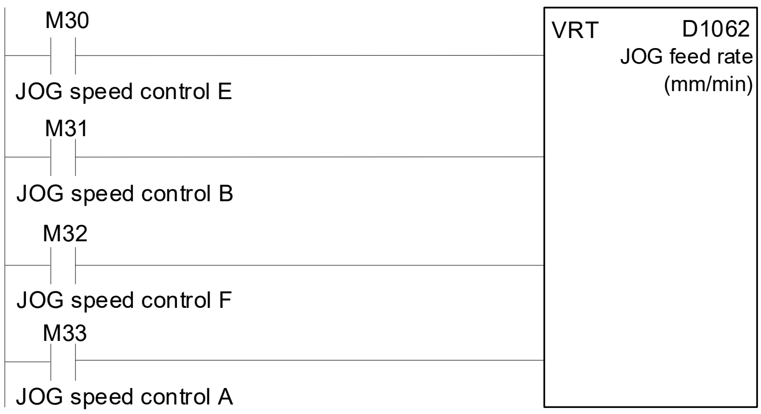
16-bit instruction: VRT continuous execution type (70 steps).
32-bit instruction: DVRT continuous execution type (134 steps).
Flag: none.
Notes on the use of operands: none.

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:




| BINCD | +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 | | | | |
| … | | | | | |
| 63 | | | | | |

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

## 4.8 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.

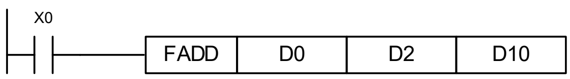
Instruction description:

$S_1$: 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.

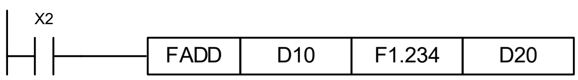
Program example 1:

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).



4

## ■ 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.

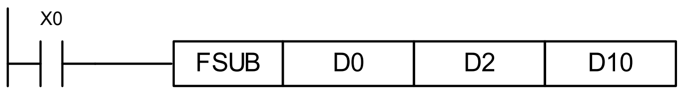
Instruction description:

$S_1$: 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.

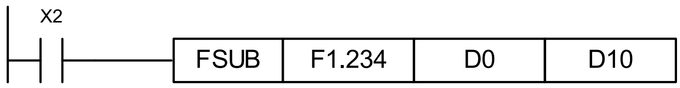
Program example 1:

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 | S1, S2, 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.

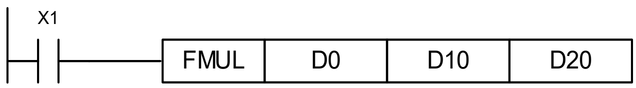
Instruction description:

$S_1$: 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.

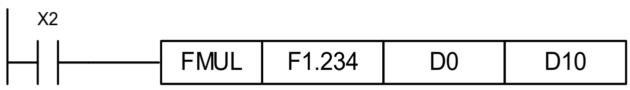
Program example 1:

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).



4

## ■ 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.

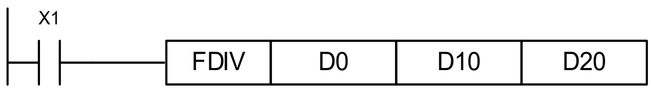
Instruction description:

$S_1$: 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.

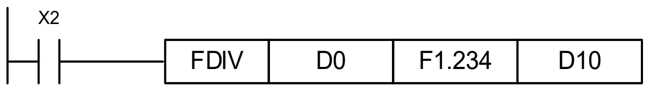
Program example 1:

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 | | S₁, S₂, 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.

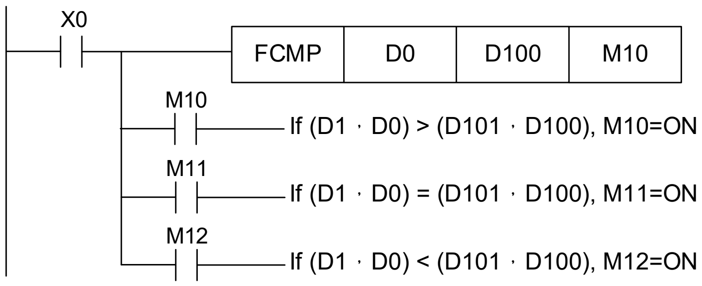
Instruction description:

$S_1$: 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.



## ■ 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.

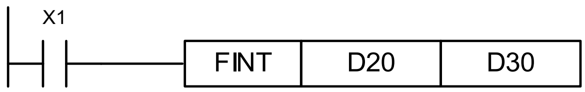
Instruction description:

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.

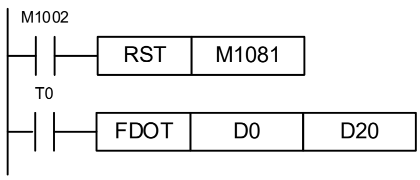
Instruction description:

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.

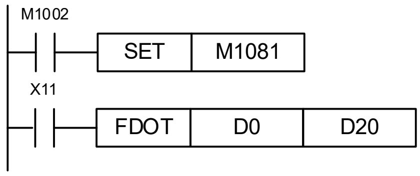
Program example 1:

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 ONand 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

| API | - | | | FRAD | S, D | | | | Convert degrees to radians | | | | NC series | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 65 | | | | | | | | | | | | | | | |
| | 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: 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.

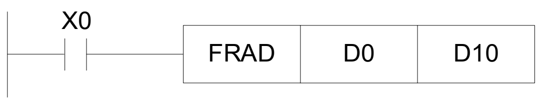
Instruction description:

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.




| | | | |
|---|---|---|---|
| S | D1 | D0 | Radian<br>Binary floating-point |
| | ⇩ | | |
| D | D11 | D10 | Radian (degree x π/180)<br>Binary floating-point |

## ■ 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.

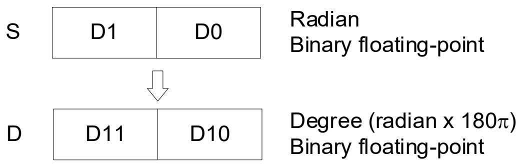
Instruction description:

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.

# 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

## 5.1 Definition of MLC special M relay and special D register

The motion logic control (MLC) 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 functions related to the system and servo axis. The MLC's 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, being 0 (OFF) and 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- prefixes followed by five 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)

There are each 65,535 address numbers for the M and D registers in the NC5 series controller. The range between M20000 to M24999, M30000 to M34999, D20000 to D24999 and D30000 to D34999 are the special M and D registers in the system. Therefore, all of these special M and D register will separated in different groups according to the NC5 series controller support multi-channel control, as shown in the table below.

| Channel Index | System Special M and D<br>MLC to System (NC) | | System Special M and D<br>System (NC) to MLC | |
|---|---|---|---|---|
| General | M20000 ~ M20999 | D20000 ~ D20999 | M30000 ~ M30999 | D30000 ~ D30999 |
| Channel 1 | M21000 ~ M21999 | D21000 ~ D21999 | M31000 ~ M31999 | D31000 ~ D31999 |
| Channel 2 | M22000 ~ M22999 | D22000 ~ D22999 | M32000 ~ M32999 | D32000 ~ D32999 |
| Channel 3 | M23000 ~ M23999 | D23000 ~ D23999 | M33000 ~ M33999 | D33000 ~ D33999 |
| Channel 4 | M24000 ~ M24999 | D24000 ~ D24999 | M34000 ~ M34999 | D34000 ~ D34999 |

| Channel Index | For HMI Special M and D MLC to System (HMI) | | For HMI Special M and D System (HMI) to MLC | |
|---|---|---|---|---|
| General | M49000 ~ M49099 | D49000 ~ D49099 | M59000 ~ M59099 | D59000 ~ D59099 |
| Channel 1 | M49100 ~ M49199 | D49100 ~ D49199 | M59100 ~ M59199 | D59100 ~ D59199 |
| Channel 2 | M49200 ~ M49299 | D49200 ~ D49299 | M59200 ~ M59299 | D59200 ~ D59299 |
| Channel 3 | M49300 ~ M49399 | D49300 ~ D49399 | M59300 ~ M59399 | D59300 ~ D59399 |
| Channel 4 | M49400 ~ M49499 | D49400 ~ D49499 | M59400 ~ M59499 | D59400 ~ D59499 |

For the special M and D register, when the tens of thousands digit is 2, it means the MLC is sending a value to the NC system. If the digit is 3, it means the NC system is updating its status for the MLC to read and check. If the digit is 4, it means the MLC is sending a value to the HMI system. If the digit is 5, it means the HMI system is updating its status for the MLC to read and check. When the tens of thousands digit is 2 or 3 and the thousands digit is 0, it means this register is for system general purpose use. If the digit is 1 to 4, it represents the specific NC channel respectively. The remaining three digits from 000 to 999 are for the functional index. The chapters below will give detailed function descriptions for each available special M and D.

# 5.2 List of special M and special D

## 5.2.1 List of special M (MLC to system)

### 5.2.1.1 M21000 to M29999

<table>
<tr><th>Function Name</th><th>Special M</th><th>Description</th><th>Device</th></tr>
<tr><td>Spindle1 DO Control</td><td>M20018</td><td>When this function is ON, the system will output the digital signal on the Pin 9 of Spindle 1 connector.</td><td>R/W</td></tr>
<tr><td>Spindle2 DO Control</td><td>M20019</td><td>When this function is ON, the system will output the digital signal on the Pin 9 of Spindle 2 connector.</td><td>R/W</td></tr>
<tr><td>Enable 3 MPG Control</td><td>M20024</td><td>Enable 3 sets of MPG function control. The controller can process three sets of MPG pulse signals and control the three axes respectively.<br>When this special M is enabled, the $2^{nd}$ and $3^{rd}$ MPG will also be enabled at the same time.</td><td>R/W</td></tr>
<tr><td>Cycle Start</td><td>M2x000</td><td>Informs the NC system to execute Cycle Start and execute the NC program.</td><td>R/W</td></tr>
<tr><td>Feed Hold</td><td>M2x001</td><td>Pause the NC process in the controller system. After pausing, the system can resume the procedure with “Cycle Start” or stop all actions with “Reset”.</td><td>R/W</td></tr>
<tr><td>NC Reset</td><td>M2x004</td><td>Tells the NC system to execute the Reset procedure; all actions will be stopped.</td><td>R/W</td></tr>
<tr><td>Enable Dry Run</td><td>M2x005</td><td>Enable dry run mode. The system will execute NC programs, including regular feed and rapid command, at dry run speed. The dry run speed will refer to parameter [N1.013 Bit18 Dry run speed mode] and [N1.66 Dry run feed rate] for the speed reference.
<table>
<tr><th>Parameter</th><th>Setting</th><th>Actual Speed</th></tr>
<tr><td>N1.013 Bit18</td><td>0</td><td>N1.66 Setting</td></tr>
<tr><td>N1.013 Bit18</td><td>1</td><td>N1.66 x D2x002</td></tr>
</table></td><td>R/W</td></tr>
<tr><td>MPG Simulation</td><td>M2x006</td><td>During program execution, users can use the MPG to control the speed of movement trajectories.</td><td>R/W</td></tr>
<tr><td>Disable Hardware Limit</td><td>M2x007</td><td>The limit signal of each axis is ignored when this function is enabled.</td><td>R/W</td></tr>
<tr><td>Single Block</td><td>M2x008</td><td>In AUTO mode, the program stops after one block is executed.</td><td>R/W</td></tr>
<tr><td>Optional Stop</td><td>M2x009</td><td>Enable the 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>M2x010</td><td>The program skips the block containing the symbol '/' when this function is enabled.</td><td>R/W</td></tr>
<tr><td>M, S, and T Codes Lock</td><td>M2x011</td><td>The program skips any block containing M, S, T code when this function is enabled.</td><td>R/W</td></tr>
<tr><td>Servo ON/ OFF</td><td>M2x012</td><td>Servo ON or OFF for all of connected servo drives in the corresponding channel.</td><td>R/W</td></tr>
<tr><td>Enable Emergency Stop</td><td>M2x013</td><td>When this function is ON, the NC will trigger an emergency stop.</td><td>R/W</td></tr>
<tr><td>Enable Synchronization Adjustment</td><td>M2x014</td><td>Manually trigger synchronization adjustment.</td><td>R/W</td></tr>
<tr><td>M, S, and T Codes Finished</td><td>M2x016</td><td>Triggering this signal informs the NC system that the procedures for M, S and T codes are complete.</td><td>R/W</td></tr>
<tr><td>M96 Program Interruption</td><td>M2x019</td><td>After the NC executes M96, if this function is triggered, the NC system interrupts the main program and jumps to the subprogram to execute it.</td><td>R/W</td></tr>
</table>

<table>
<tr><th>Function Name</th><th>Special M</th><th>Description</th><th>Device</th></tr>
<tr><td>M, S, and T Codes Call Macro Lock</td><td>M2x021</td><td>When this function is ON, the NC will not execute macro program including M code process regarding M, S, and T codes in the NC program.</td><td>R/W</td></tr>
<tr><td>Macro Call Activation</td><td>M2x025</td><td>Activates macro call.<br>(Only works with the correct macro-ID in AUTO mode)</td><td>R/W</td></tr>
<tr><td>M99 Call Stop</td><td>M2x026</td><td>When this function is ON, the NC system will slow down to stop and then stop all process when executing M99.</td><td>R/W</td></tr>
<tr><td>1st Macro Call Preparation</td><td>M2x032</td><td rowspan="16">Initializes macro call.<br>(A successful call will only occur when the correct macro number [D2x064~D2x079] and the corresponding macro program both exist)</td><td>R/W</td></tr>
<tr><td>2nd Macro Call Preparation</td><td>M2x033</td><td>R/W</td></tr>
<tr><td>3rd Macro Call Preparation</td><td>M2x034</td><td>R/W</td></tr>
<tr><td>4th Macro Call Preparation</td><td>M2x035</td><td>R/W</td></tr>
<tr><td>5th Macro Call Preparation</td><td>M2x036</td><td>R/W</td></tr>
<tr><td>6th Macro Call Preparation</td><td>M2x037</td><td>R/W</td></tr>
<tr><td>7th Macro Call Preparation.</td><td>M2x038</td><td>R/W</td></tr>
<tr><td>8th Macro Call Preparation</td><td>M2x039</td><td>R/W</td></tr>
<tr><td>9th Macro Call Preparation</td><td>M2x040</td><td>R/W</td></tr>
<tr><td>10th Macro Call Preparation</td><td>M2x041</td><td>R/W</td></tr>
<tr><td>11th Macro Call Preparation</td><td>M2x042</td><td>R/W</td></tr>
<tr><td>12th Macro Call Preparation</td><td>M2x043</td><td>R/W</td></tr>
<tr><td>13th Macro Call Preparation</td><td>M2x044</td><td>R/W</td></tr>
<tr><td>14th Macro Call Preparation</td><td>M2x045</td><td>R/W</td></tr>
<tr><td>15th Macro Call Preparation</td><td>M2x046</td><td>R/W</td></tr>
<tr><td>16th Macro Call Preparation</td><td>M2x047</td><td>R/W</td></tr>
<tr><td rowspan="4">Robot Coordinate System</td><td>M2x048</td><td rowspan="4">The robot coordinate system can be set as shown in the table below.<br>The CNC controller needs to be in manual mode. The D2x016 defines the coordinate system, e.g., 1 = G54, 2 = G55, …, 5 = G58 and 6 = G59.
<table>
<tr><th></th><th>PCS (Piece)</th><th>TCS (Tool)</th><th>JCS (Joint)</th></tr>
<tr><td>M2x048</td><td>1</td><td>1</td><td>0</td></tr>
<tr><td>M2x049</td><td>1</td><td>1</td><td>0</td></tr>
<tr><td>M2x050</td><td>1</td><td>0</td><td>0</td></tr>
<tr><td>M2x051</td><td>1</td><td>1</td><td>0</td></tr>
<tr><td>D2x016</td><td>1 ~ 6</td><td>1 ~ 6</td><td>1 ~ 6</td></tr>
</table></td><td rowspan="4">R/W</td></tr>
<tr><td>M2x049</td></tr>
<tr><td>M2x050</td></tr>
<tr><td>M2x051</td></tr>
</table>

5

| Function Name | Special M | Description | Device |
|---|---|---|---|
| Tool Magazine 1 Move Forward | M2x064 | Moves tool magazine 1 forward. When this special M relay is triggered, the tool pot deviation **[D3x039]** is decreased by 1, and the standby tool pot number **[D3x038]** is increased by 1. | R/W |
| Tool Magazine 1 Move Backward | M2x065 | Moves tool magazine 1 backward. When this special M relay is triggered, the tool pot deviation **[D3x039]** is increased by 1, and the standby tool pot number **[D3x038]** is decreased by 1. | R/W |
| Tool 1 Exchange | M2x066 | Exchanges tool data in tool magazine 1. | R/W |
| Tool Magazine 1 Reset | M2x067 | When this special M relay is triggered, the tool data in tool magazine 1 is reset. | R/W |
| Tool Magazine 2 Move Forward | M2x072 | Moves tool magazine 1 forward. When this special M relay is triggered, the tool pot deviation **[D3x045]** is decreased by 1, and the standby tool pot number **[D3x044]** is increased by 1. | R/W |
| Tool Magazine 2 Move Backward | M2x073 | Moves tool magazine 1 backward. When this special M relay is triggered, the tool pot deviation **[D3x045]** is increased by 1, and the standby tool pot number **[D3x044]** is decreased by 1. | R/W |
| Tool 2 Exchange | M2x074 | Exchanges tool data in tool magazine 2. | R/W |
| Tool Magazine 2 Reset | M2x075 | When this special M relay is triggered, the tool data in tool magazine 2 is reset. | R/W |
| Canceling Tapping Interrupt Status | M2x080 | When the NC system is in the tapping interrupt status, users can trigger this M relay to cancel the interrupt status. | R/W |
| MLC to NC Variable 1 | M2x128 | The system will move the binary status of this special M to NC variable #25000 as 0 or 1. | R/W |
| MLC to NC Variable 2 | M2x129 | The system will move the binary status of this special M to NC variable #25001 as 0 or 1. | R/W |
| MLC to NC Variable 3 | M2x130 | The system will move the binary status of this special M to NC variable #25002 as 0 or 1. | R/W |
| MLC to NC Variable 4 | M2x131 | The system will move the binary status of this special M to NC variable #25003 as 0 or 1. | R/W |
| MLC to NC Variable 5 | M2x132 | The system will move the binary status of this special M to NC variable #25004 as 0 or 1. | R/W |
| MLC to NC Variable 6 | M2x133 | The system will move the binary status of this special M to NC variable #25005 as 0 or 1. | R/W |
| MLC to NC Variable 7 | M2x134 | The system will move the binary status of this special M to NC variable #25006 as 0 or 1. | R/W |
| MLC to NC Variable 8 | M2x135 | The system will move the binary status of this special M to NC variable #25007 as 0 or 1. | R/W |
| MLC to NC Variable 9 | M2x136 | The system will move the binary status of this special M to NC variable #25008 as 0 or 1. | R/W |
| MLC to NC Variable 10 | M2x137 | The system will move the binary status of this special M to NC variable #25009 as 0 or 1. | R/W |
| MLC to NC Variable 11 | M2x138 | The system will move the binary status of this special M to NC variable #25010 as 0 or 1. | R/W |
| MLC to NC Variable 12 | M2x139 | The system will move the binary status of this special M to NC variable #25011 as 0 or 1. | R/W |
| MLC to NC Variable 13 | M2x140 | The system will move the binary status of this special M to NC variable #25012 as 0 or 1. | R/W |
| MLC to NC Variable 14 | M2x141 | The system will move the binary status of this special M to NC variable #25013 as 0 or 1. | R/W |
| MLC to NC Variable 15 | M2x142 | The system will move the binary status of this special M to NC variable #25014 as 0 or 1. | R/W |

| Function Name | Special M | Description | Device |
|---|---|---|---|
| MLC to NC Variable 16 | M2x143 | The system will move the binary status of this special M to NC variable #25015 as 0 or 1. | R/W |
| MLC to NC Variable 17 | M2x144 | The system will move the binary status of this special M to NC variable #25016 as 0 or 1. | R/W |
| MLC to NC Variable 18 | M2x145 | The system will move the binary status of this special M to NC variable #25017 as 0 or 1. | R/W |
| MLC to NC Variable 19 | M2x146 | The system will move the binary status of this special M to NC variable #25018 as 0 or 1. | R/W |
| MLC to NC Variable 20 | M2x147 | The system will move the binary status of this special M to NC variable #25019 as 0 or 1. | R/W |
| MLC to NC Variable 21 | M2x148 | The system will move the binary status of this special M to NC variable #25020 as 0 or 1. | R/W |
| MLC to NC Variable 22 | M2x149 | The system will move the binary status of this special M to NC variable #25021 as 0 or 1. | R/W |
| MLC to NC Variable 23 | M2x150 | The system will move the binary status of this special M to NC variable #25022 as 0 or 1. | R/W |
| MLC to NC Variable 24 | M2x151 | The system will move the binary status of this special M to NC variable #25023 as 0 or 1. | R/W |
| MLC to NC Variable 25 | M2x152 | The system will move the binary status of this special M to NC variable #25024 as 0 or 1. | R/W |
| MLC to NC Variable 26 | M2x153 | The system will move the binary status of this special M to NC variable #25025 as 0 or 1. | R/W |
| MLC to NC Variable 27 | M2x154 | The system will move the binary status of this special M to NC variable #25026 as 0 or 1. | R/W |
| MLC to NC Variable 28 | M2x155 | The system will move the binary status of this special M to NC variable #25027 as 0 or 1. | R/W |
| MLC to NC Variable 29 | M2x156 | The system will move the binary status of this special M to NC variable #25028 as 0 or 1. | R/W |
| MLC to NC Variable 30 | M2x157 | The system will move the binary status of this special M to NC variable #25029 as 0 or 1. | R/W |
| MLC to NC Variable 31 | M2x158 | The system will move the binary status of this special M to NC variable #25030 as 0 or 1. | R/W |
| MLC to NC Variable 32 | M2x159 | The system will move the binary status of this special M to NC variable #25031 as 0 or 1. | R/W |
| MLC to NC Variable 33 | M2x160 | The system will move the binary status of this special M to NC variable #25032 as 0 or 1. | R/W |
| MLC to NC Variable 34 | M2x161 | The system will move the binary status of this special M to NC variable #25033 as 0 or 1. | R/W |
| MLC to NC Variable 35 | M2x162 | The system will move the binary status of this special M to NC variable #25034 as 0 or 1. | R/W |
| MLC to NC Variable 36 | M2x163 | The system will move the binary status of this special M to NC variable #25035 as 0 or 1. | R/W |
| MLC to NC Variable 37 | M2x164 | The system will move the binary status of this special M to NC variable #25036 as 0 or 1. | R/W |
| MLC to NC Variable 38 | M2x165 | The system will move the binary status of this special M to NC variable #25037 as 0 or 1. | R/W |
| MLC to NC Variable 39 | M2x166 | The system will move the binary status of this special M to NC variable #25038 as 0 or 1. | R/W |
| MLC to NC Variable 40 | M2x167 | The system will move the binary status of this special M to NC variable #25039 as 0 or 1. | R/W |
| MLC to NC Variable 41 | M2x168 | The system will move the binary status of this special M to NC variable #25040 as 0 or 1. | R/W |
| MLC to NC Variable 42 | M2x169 | The system will move the binary status of this special M to NC variable #25041 as 0 or 1. | R/W |

| Function Name | Special M | Description | Device |
|---|---|---|---|
| MLC to NC Variable 43 | M2x170 | The system will move the binary status of this special M to NC variable #25042 as 0 or 1. | R/W |
| MLC to NC Variable 44 | M2x171 | The system will move the binary status of this special M to NC variable #25043 as 0 or 1. | R/W |
| MLC to NC Variable 45 | M2x172 | The system will move the binary status of this special M to NC variable #25044 as 0 or 1. | R/W |
| MLC to NC Variable 46 | M2x173 | The system will move the binary status of this special M to NC variable #25045 as 0 or 1. | R/W |
| MLC to NC Variable 47 | M2x174 | The system will move the binary status of this special M to NC variable #25046 as 0 or 1. | R/W |
| MLC to NC Variable 48 | M2x175 | The system will move the binary status of this special M to NC variable #25047 as 0 or 1. | R/W |
| MLC to NC Variable 49 | M2x176 | The system will move the binary status of this special M to NC variable #25048 as 0 or 1. | R/W |
| MLC to NC Variable 50 | M2x177 | The system will move the binary status of this special M to NC variable #25049 as 0 or 1. | R/W |
| MLC to NC Variable 51 | M2x178 | The system will move the binary status of this special M to NC variable #25050 as 0 or 1. | R/W |
| MLC to NC Variable 52 | M2x179 | The system will move the binary status of this special M to NC variable #25051 as 0 or 1. | R/W |
| MLC to NC Variable 53 | M2x180 | The system will move the binary status of this special M to NC variable #25052 as 0 or 1. | R/W |
| MLC to NC Variable 54 | M2x181 | The system will move the binary status of this special M to NC variable #25053 as 0 or 1. | R/W |
| MLC to NC Variable 55 | M2x182 | The system will move the binary status of this special M to NC variable #25054 as 0 or 1. | R/W |
| MLC to NC Variable 56 | M2x183 | The system will move the binary status of this special M to NC variable #25055 as 0 or 1. | R/W |
| MLC to NC Variable 57 | M2x184 | The system will move the binary status of this special M to NC variable #25056 as 0 or 1. | R/W |
| MLC to NC Variable 58 | M2x185 | The system will move the binary status of this special M to NC variable #25057 as 0 or 1. | R/W |
| MLC to NC Variable 59 | M2x186 | The system will move the binary status of this special M to NC variable #25058 as 0 or 1. | R/W |
| MLC to NC Variable 60 | M2x187 | The system will move the binary status of this special M to NC variable #25059 as 0 or 1. | R/W |
| MLC to NC Variable 61 | M2x188 | The system will move the binary status of this special M to NC variable #25060 as 0 or 1. | R/W |
| MLC to NC Variable 62 | M2x189 | The system will move the binary status of this special M to NC variable #25061 as 0 or 1. | R/W |
| MLC to NC Variable 63 | M2x190 | The system will move the binary status of this special M to NC variable #25062 as 0 or 1. | R/W |
| MLC to NC Variable 64 | M2x191 | The system will move the binary status of this special M to NC variable #25063 as 0 or 1. | R/W |
| MLC to NC Variable 65 | M2x192 | The system will move the binary status of this special M to NC variable #25064 as 0 or 1. | R/W |
| MLC to NC Variable 66 | M2x193 | The system will move the binary status of this special M to NC variable #25065 as 0 or 1. | R/W |
| MLC to NC Variable 67 | M2x194 | The system will move the binary status of this special M to NC variable #25066 as 0 or 1. | R/W |
| MLC to NC Variable 68 | M2x195 | The system will move the binary status of this special M to NC variable #25067 as 0 or 1. | R/W |
| MLC to NC Variable 69 | M2x196 | The system will move the binary status of this special M to NC variable #25068 as 0 or 1. | R/W |

| Function Name | Special M | Description | Device |
|---|---|---|---|
| MLC to NC Variable 70 | M2x197 | The system will move the binary status of this special M to NC variable #25069 as 0 or 1. | R/W |
| MLC to NC Variable 71 | M2x198 | The system will move the binary status of this special M to NC variable #25070 as 0 or 1. | R/W |
| MLC to NC Variable 72 | M2x199 | The system will move the binary status of this special M to NC variable #25071 as 0 or 1. | R/W |
| MLC to NC Variable 73 | M2x200 | The system will move the binary status of this special M to NC variable #25072 as 0 or 1. | R/W |
| MLC to NC Variable 74 | M2x201 | The system will move the binary status of this special M to NC variable #25073 as 0 or 1. | R/W |
| MLC to NC Variable 75 | M2x202 | The system will move the binary status of this special M to NC variable #25074 as 0 or 1. | R/W |
| MLC to NC Variable 76 | M2x203 | The system will move the binary status of this special M to NC variable #25075 as 0 or 1. | R/W |
| MLC to NC Variable 77 | M2x204 | The system will move the binary status of this special M to NC variable #25076 as 0 or 1. | R/W |
| MLC to NC Variable 78 | M2x205 | The system will move the binary status of this special M to NC variable #25077 as 0 or 1. | R/W |
| MLC to NC Variable 79 | M2x206 | The system will move the binary status of this special M to NC variable #25078 as 0 or 1. | R/W |
| MLC to NC Variable 80 | M2x207 | The system will move the binary status of this special M to NC variable #25079 as 0 or 1. | R/W |
| MLC to NC Variable 81 | M2x208 | The system will move the binary status of this special M to NC variable #25080 as 0 or 1. | R/W |
| MLC to NC Variable 82 | M2x209 | The system will move the binary status of this special M to NC variable #25081 as 0 or 1. | R/W |
| MLC to NC Variable 83 | M2x210 | The system will move the binary status of this special M to NC variable #25082 as 0 or 1. | R/W |
| MLC to NC Variable 84 | M2x211 | The system will move the binary status of this special M to NC variable #25083 as 0 or 1. | R/W |
| MLC to NC Variable 85 | M2x212 | The system will move the binary status of this special M to NC variable #25084 as 0 or 1. | R/W |
| MLC to NC Variable 86 | M2x213 | The system will move the binary status of this special M to NC variable #25085 as 0 or 1. | R/W |
| MLC to NC Variable 87 | M2x214 | The system will move the binary status of this special M to NC variable #25086 as 0 or 1. | R/W |
| MLC to NC Variable 88 | M2x215 | The system will move the binary status of this special M to NC variable #25087 as 0 or 1. | R/W |
| MLC to NC Variable 89 | M2x216 | The system will move the binary status of this special M to NC variable #25088 as 0 or 1. | R/W |
| MLC to NC Variable 90 | M2x217 | The system will move the binary status of this special M to NC variable #25089 as 0 or 1. | R/W |
| MLC to NC Variable 91 | M2x218 | The system will move the binary status of this special M to NC variable #25090 as 0 or 1. | R/W |
| MLC to NC Variable 92 | M2x219 | The system will move the binary status of this special M to NC variable #25091 as 0 or 1. | R/W |
| MLC to NC Variable 93 | M2x220 | The system will move the binary status of this special M to NC variable #25092 as 0 or 1. | R/W |
| MLC to NC Variable 94 | M2x221 | The system will move the binary status of this special M to NC variable #25093 as 0 or 1. | R/W |
| MLC to NC Variable 95 | M2x222 | The system will move the binary status of this special M to NC variable #25094 as 0 or 1. | R/W |
| MLC to NC Variable 96 | M2x223 | The system will move the binary status of this special M to NC variable #25095 as 0 or 1. | R/W |

5

| Function Name | Special M | Description | Device |
|---|---|---|---|
| MLC to NC Variable 97 | M2x224 | The system will move the binary status of this special M to NC variable #25096 as 0 or 1. | R/W |
| MLC to NC Variable 98 | M2x225 | The system will move the binary status of this special M to NC variable #25097 as 0 or 1. | R/W |
| MLC to NC Variable 99 | M2x226 | The system will move the binary status of this special M to NC variable #25098 as 0 or 1. | R/W |
| MLC to NC Variable 100 | M2x227 | The system will move the binary status of this special M to NC variable #25099 as 0 or 1. | R/W |
| MLC to NC Variable 101 | M2x228 | The system will move the binary status of this special M to NC variable #25100 as 0 or 1. | R/W |
| MLC to NC Variable 102 | M2x229 | The system will move the binary status of this special M to NC variable #25101 as 0 or 1. | R/W |
| MLC to NC Variable 103 | M2x230 | The system will move the binary status of this special M to NC variable #25102 as 0 or 1. | R/W |
| MLC to NC Variable 104 | M2x231 | The system will move the binary status of this special M to NC variable #25103 as 0 or 1. | R/W |
| MLC to NC Variable 105 | M2x232 | The system will move the binary status of this special M to NC variable #25104 as 0 or 1. | R/W |
| MLC to NC Variable 106 | M2x233 | The system will move the binary status of this special M to NC variable #25105 as 0 or 1. | R/W |
| MLC to NC Variable 107 | M2x234 | The system will move the binary status of this special M to NC variable #25106 as 0 or 1. | R/W |
| MLC to NC Variable 108 | M2x235 | The system will move the binary status of this special M to NC variable #25107 as 0 or 1. | R/W |
| MLC to NC Variable 109 | M2x236 | The system will move the binary status of this special M to NC variable #25108 as 0 or 1. | R/W |
| MLC to NC Variable 110 | M2x237 | The system will move the binary status of this special M to NC variable #25109 as 0 or 1. | R/W |
| MLC to NC Variable 111 | M2x238 | The system will move the binary status of this special M to NC variable #25110 as 0 or 1. | R/W |
| MLC to NC Variable 112 | M2x239 | The system will move the binary status of this special M to NC variable #25111 as 0 or 1. | R/W |
| MLC to NC Variable 113 | M2x240 | The system will move the binary status of this special M to NC variable #25112 as 0 or 1. | R/W |
| MLC to NC Variable 114 | M2x241 | The system will move the binary status of this special M to NC variable #25113 as 0 or 1. | R/W |
| MLC to NC Variable 115 | M2x242 | The system will move the binary status of this special M to NC variable #25114 as 0 or 1. | R/W |
| MLC to NC Variable 116 | M2x243 | The system will move the binary status of this special M to NC variable #25115 as 0 or 1. | R/W |
| MLC to NC Variable 117 | M2x244 | The system will move the binary status of this special M to NC variable #25116 as 0 or 1. | R/W |
| MLC to NC Variable 118 | M2x245 | The system will move the binary status of this special M to NC variable #25117 as 0 or 1. | R/W |
| MLC to NC Variable 119 | M2x246 | The system will move the binary status of this special M to NC variable #25118 as 0 or 1. | R/W |
| MLC to NC Variable 120 | M2x247 | The system will move the binary status of this special M to NC variable #25119 as 0 or 1. | R/W |
| MLC to NC Variable 121 | M2x248 | The system will move the binary status of this special M to NC variable #25120 as 0 or 1. | R/W |
| MLC to NC Variable 122 | M2x249 | The system will move the binary status of this special M to NC variable #25121 as 0 or 1. | R/W |
| MLC to NC Variable 123 | M2x250 | The system will move the binary status of this special M to NC variable #25122 as 0 or 1. | R/W |

| Function Name | Special M | Description | Device |
|---|---|---|---|
| MLC to NC Variable 124 | M2x251 | The system will move the binary status of this special M to NC variable #25123 as 0 or 1. | R/W |
| MLC to NC Variable 125 | M2x252 | The system will move the binary status of this special M to NC variable #25124 as 0 or 1. | R/W |
| MLC to NC Variable 126 | M2x253 | The system will move the binary status of this special M to NC variable #25125 as 0 or 1. | R/W |
| MLC to NC Variable 127 | M2x254 | The system will move the binary status of this special M to NC variable #25126 as 0 or 1. | R/W |
| MLC to NC Variable 128 | M2x255 | The system will move the binary status of this special M to NC variable #25127 as 0 or 1. | R/W |
| Synchronous Control Enable | M2x256 | This special M must be set to ON when using synchronous functions to allow the NC to enable synchronous control. | R/W |
| Command Transfer Enable | M2x257 | This special M must be set to ON when using command transfer functions to allow the NC to enable command transfer. | R/W |
| Axes Oscillation Control Enable | M2x264 | This special M must be set to ON when using axes oscillation functions. The system will execute axis oscillation commands based on the settings in **[D2x018 ~ D2x023]**. If the axis is already performing path interpolation, the oscillation command will be added to the original position command.<br>Only available on modes other than HOME and EDIT. | R/W |
| X Axis Servo OFF | M2x272 | Trigger this special M to set the X axis to Servo OFF status. | R/W |
| Y Axis Servo OFF | M2x273 | Trigger this special M to set the Y axis to Servo OFF status. | R/W |
| Z Axis Servo OFF | M2x274 | Trigger this special M to set the Z axis to Servo OFF status. | R/W |
| A Axis Servo OFF | M2x275 | Trigger this special M to set the A axis to Servo OFF status. | R/W |
| B Axis Servo OFF | M2x276 | Trigger this special M to set the B axis to Servo OFF status. | R/W |
| C Axis Servo OFF | M2x277 | Trigger this special M to set the C axis to Servo OFF status. | R/W |
| U Axis Servo OFF | M2x278 | Trigger this special M to set the U axis to Servo OFF status. | R/W |
| V Axis Servo OFF | M2x279 | Trigger this special M to set the V axis to Servo OFF status. | R/W |
| W Axis Servo OFF | M2x280 | Trigger this special M to set the W axis to Servo OFF status. | R/W |
| $10^{th}$ Axis Servo OFF | M2x281 | Trigger this special M to set the $10^{th}$ axis to Servo OFF status. | R/W |
| $11^{th}$ Axis Servo OFF | M2x282 | Trigger this special M to set the $11^{th}$ axis to Servo OFF status. | R/W |
| $12^{th}$ Axis Servo OFF | M2x283 | Trigger this special M to set the $12^{th}$ axis to Servo OFF status. | R/W |
| $13^{th}$ Axis Servo OFF | M2x284 | Trigger this special M to set the $13^{th}$ axis to Servo OFF status. | R/W |
| $14^{th}$ Axis Servo OFF | M2x285 | Trigger this special M to set the $14^{th}$ axis to Servo OFF status. | R/W |
| $15^{th}$ Axis Servo OFF | M2x286 | Trigger this special M to set the $15^{th}$ axis to Servo OFF status. | R/W |
| $16^{th}$ Axis Servo OFF | M2x287 | Trigger this special M to set the $16^{th}$ axis to Servo OFF status. | R/W |

| Function Name | Special M | Description | Device |
|---|---|---|---|
| X Slave Axis Follows the Master Axis | M2x288 | Sets the X axis as the slave axis for synchronous control. The master axis ID must be set with **[N2.015]** in advance. | R/W |
| Y Slave Axis Follows the Master Axis | M2x289 | Sets the Y axis as the slave axis for synchronous control. The master axis ID must be set with **[N2.015]** in advance. | R/W |
| Z Slave Axis Follows the Master Axis | M2x290 | Sets the Z axis as the slave axis for synchronous control. The master axis ID must be set with **[N2.015]** in advance. | R/W |
| A Slave Axis Follows the Master Axis | M2x291 | Sets the A axis as the slave axis for synchronous control. The master axis ID must be set with **[N2.015]** in advance. | R/W |
| B Slave Axis Follows the Master Axis | M2x292 | Sets the B axis as the slave axis for synchronous control. The master axis ID must be set with **[N2.015]** in advance. | R/W |
| C Slave Axis Follows the Master Axis | M2x293 | Sets the C axis as the slave axis for synchronous control. The master axis ID must be set with **[N2.015]** in advance. | R/W |
| U Slave Axis Follows the Master Axis | M2x294 | Sets the U axis as the slave axis for synchronous control. The master axis ID must be set with **[N2.015]** in advance. | R/W |
| V Slave Axis Follows the Master Axis | M2x295 | Sets the V axis as the slave axis for synchronous control. The master axis ID must be set with **[N2.015]** in advance. | R/W |
| W Slave Axis Follows the Master Axis | M2x296 | Sets the W axis as the slave axis for synchronous control. The master axis ID must be set with **[N2.015]** in advance. | R/W |
| $10^{th}$ Slave Axis Follows the Master Axis | M2x297 | Sets the $10^{th}$ axis as the slave axis for synchronous control. The master axis ID must be set with **[N2.015]** in advance. | R/W |
| $11^{th}$ Slave Axis Follows the Master Axis | M2x298 | Sets the $11^{th}$ axis as the slave axis for synchronous control. The master axis ID must be set with **[N2.015]** in advance. | R/W |
| $12^{th}$ Slave Axis Follows the Master Axis | M2x299 | Sets the $12^{th}$ axis as the slave axis for synchronous control. The master axis ID must be set with **[N2.015]** in advance. | R/W |
| $13^{th}$ Slave Axis Follows the Master Axis | M2x300 | Sets the $13^{th}$ axis as the slave axis for synchronous control. The master axis ID must be set with **[N2.015]** in advance. | R/W |
| $14^{th}$ Slave Axis Follows the Master Axis | M2x301 | Sets the $14^{th}$ axis as the slave axis for synchronous control. The master axis ID must be set with **[N2.015]** in advance. | R/W |
| $15^{th}$ Slave Axis Follows the Master Axis | M2x302 | Sets the $15^{th}$ axis as the slave axis for synchronous control. The master axis ID must be set with **[N2.015]** in advance. | R/W |
| $16^{th}$ Slave Axis Follows the Master Axis | M2x303 | Sets the $16^{th}$ axis as the slave axis for synchronous control. The master axis ID must be set with **[N2.015]** in advance. | R/W |
| X Axis Receives Command from The Master Axis | M2x304 | Specifies the X axis as the slave axis to receive the transfer command from the master axis. The master axis ID must be set with **[N2.015]** in advance. | R/W |
| Y Axis Receives Command from The Master Axis | M2x305 | Specifies the Y axis as the slave axis to receive the transfer command from the master axis. The master axis ID must be set with **[N2.015]** in advance. | R/W |
| Z Axis Receives Command from The Master Axis | M2x306 | Specifies the Z axis as the slave axis to receive the transfer command from the master axis. The master axis ID must be set with **[N2.015]** in advance. | R/W |

| Function Name | Special M | Description | Device |
|---|---|---|---|
| A Axis Receives Command from The Master Axis | M2x307 | Specifies the A axis as the slave axis to receive the transfer command from the master axis. The master axis ID must be set with **[N2.015]** in advance. | R/W |
| B Axis Receives Command from The Master Axis | M2x308 | Specifies the B axis as the slave axis to receive the transfer command from the master axis. The master axis ID must be set with **[N2.015]** in advance. | R/W |
| C Axis Receives Command from The Master Axis | M2x309 | Specifies the C axis as the slave axis to receive the transfer command from the master axis. The master axis ID must be set with **[N2.015]** in advance. | R/W |
| U Axis Receives Command from The Master Axis | M2x310 | Specifies the U axis as the slave axis to receive the transfer command from the master axis. The master axis ID must be set with **[N2.015]** in advance. | R/W |
| V Axis Receives Command from The Master Axis | M2x311 | Specifies the V axis as the slave axis to receive the transfer command from the master axis. The master axis ID must be set with **[N2.015]** in advance. | R/W |
| W Axis Receives Command from The Master Axis | M2x312 | Specifies the W axis as the slave axis to receive the transfer command from the master axis. The master axis ID must be set with **[N2.015]** in advance. | R/W |
| $10^{th}$ Axis Receives Command from The Master Axis | M2x313 | Specifies the $10^{th}$ axis as the slave axis to receive the transfer command from the master axis. The master axis ID must be set with **[N2.015]** in advance. | R/W |
| $11^{th}$ Axis Receives Command from The Master Axis | M2x314 | Specifies the $11^{th}$ axis as the slave axis to receive the transfer command from the master axis. The master axis ID must be set with **[N2.015]** in advance. | R/W |
| $12^{th}$ Axis Receives Command from The Master Axis | M2x315 | Specifies the $12^{th}$ axis as the slave axis to receive the transfer command from the master axis. The master axis ID must be set with **[N2.015]** in advance. | R/W |
| $13^{th}$ Axis Receives Command from The Master Axis | M2x316 | Specifies the $13^{th}$ axis as the slave axis to receive the transfer command from the master axis. The master axis ID must be set with **[N2.015]** in advance. | R/W |
| $14^{th}$ Axis Receives Command from The Master Axis | M2x317 | Specifies the $14^{th}$ axis as the slave axis to receive the transfer command from the master axis. The master axis ID must be set with **[N2.015]** in advance. | R/W |
| $15^{th}$ Axis Receives Command from The Master Axis | M2x318 | Specifies the $15^{th}$ axis as the slave axis to receive the transfer command from the master axis. The master axis ID must be set with **[N2.015]** in advance. | R/W |
| $16^{th}$ Axis Receives Command from The Master Axis | M2x319 | Specifies the $16^{th}$ axis as the slave axis to receive the transfer command from the master axis. The master axis ID must be set with **[N2.015]** in advance. | R/W |
| X Axis Homing | M2x320 | Trigger this special M for X axis homing procedure. | R/W |
| Y Axis Homing | M2x321 | Trigger this special M for Y axis homing procedure. | R/W |
| Z Axis Homing | M2x322 | Trigger this special M for Z axis homing procedure. | R/W |
| A Axis Homing | M2x323 | Trigger this special M for A axis homing procedure. | R/W |
| B Axis Homing | M2x324 | Trigger this special M for B axis homing procedure. | R/W |
| C Axis Homing | M2x325 | Trigger this special M for C axis homing procedure. | R/W |
| U Axis Homing | M2x326 | Trigger this special M for U axis homing procedure. | R/W |
| V Axis Homing | M2x327 | Trigger this special M for V axis homing procedure. | R/W |
| W Axis Homing | M2x328 | Trigger this special M for W axis homing procedure. | R/W |
| $10^{th}$ Axis Homing | M2x329 | Trigger this special M for $10^{th}$ axis homing procedure. | R/W |
| $11^{th}$ Axis Homing | M2x330 | Trigger this special M for $11^{th}$ axis homing procedure. | R/W |
| $12^{th}$ Axis Homing | M2x331 | Trigger this special M for $12^{th}$ axis homing procedure. | R/W |
| $13^{th}$ Axis Homing | M2x332 | Trigger this special M for $13^{th}$ axis homing procedure. | R/W |
| $14^{th}$ Axis Homing | M2x333 | Trigger this special M for $14^{th}$ axis homing procedure. | R/W |

5

| Function Name | Special M | Description | Device |
|---|---|---|---|
| 15th Axis Homing | M2x334 | Trigger this special M for 15th axis homing procedure. | R/W |
| 16th Axis Homing | M2x335 | Trigger this special M for 16th axis homing procedure. | R/W |
| Lock X Axis Movement in Positive Direction | M2x336 | When this special M is ON, the axis will not be able to move in the positive direction. | R/W |
| Lock Y Axis Movement in Positive Direction | M2x337 | When this special M is ON, the axis will not be able to move in the positive direction. | R/W |
| Lock Z Axis Movement in Positive Direction | M2x338 | When this special M is ON, the axis will not be able to move in the positive direction. | R/W |
| Lock A Axis Movement in Positive Direction | M2x339 | When this special M is ON, the axis will not be able to move in the positive direction. | R/W |
| Lock B Axis Movement in Positive Direction | M2x340 | When this special M is ON, the axis will not be able to move in the positive direction. | R/W |
| Lock C Axis Movement in Positive Direction | M2x341 | When this special M is ON, the axis will not be able to move in the positive direction. | R/W |
| Lock U Axis Movement in Positive Direction | M2x342 | When this special M is ON, the axis will not be able to move in the positive direction. | R/W |
| Lock V Axis Movement in Positive Direction | M2x343 | When this special M is ON, the axis will not be able to move in the positive direction. | R/W |
| Lock W Axis Movement in Positive Direction | M2x344 | When this special M is ON, the axis will not be able to move in the positive direction. | R/W |
| Lock 10th Axis Movement in Positive Direction | M2x345 | When this special M is ON, the axis will not be able to move in the positive direction. | R/W |
| Lock 11th Axis Movement in Positive Direction | M2x346 | When this special M is ON, the axis will not be able to move in the positive direction. | R/W |
| Lock 12th Axis Movement in Positive Direction | M2x347 | When this special M is ON, the axis will not be able to move in the positive direction. | R/W |
| Lock 13th Axis Movement in Positive Direction | M2x348 | When this special M is ON, the axis will not be able to move in the positive direction. | R/W |
| Lock 14th Axis Movement in Positive Direction | M2x349 | When this special M is ON, the axis will not be able to move in the positive direction. | R/W |
| Lock 15th Axis Movement in Positive Direction | M2x350 | When this special M is ON, the axis will not be able to move in the positive direction. | R/W |
| Lock 16th Axis Movement in Positive Direction | M2x351 | When this special M is ON, the axis will not be able to move in the positive direction. | R/W |
| Lock X Axis Movement in Negative Direction | M2x352 | When this special M is ON, the axis will not be able to move in the negative direction. | R/W |
| Lock Y Axis Movement in Negative Direction | M2x353 | When this special M is ON, the axis will not be able to move in the negative direction. | R/W |
| Lock Z Axis Movement in Negative Direction | M2x354 | When this special M is ON, the axis will not be able to move in the negative direction. | R/W |
| Lock A Axis Movement in Negative Direction | M2x355 | When this special M is ON, the axis will not be able to move in the negative direction. | R/W |
| Lock B Axis Movement in Negative Direction | M2x356 | When this special M is ON, the axis will not be able to move in the negative direction. | R/W |
| Lock C Axis Movement in Negative Direction | M2x357 | When this special M is ON, the axis will not be able to move in the negative direction. | R/W |

| Function Name | Special M | Description | Device |
|---|---|---|---|
| Lock U Axis Movement in Negative Direction | M2x358 | When this special M is ON, the axis will not be able to move in the negative direction. | R/W |
| Lock V Axis Movement in Negative Direction | M2x359 | When this special M is ON, the axis will not be able to move in the negative direction. | R/W |
| Lock W Axis Movement in Negative Direction | M2x360 | When this special M is ON, the axis will not be able to move in the negative direction. | R/W |
| Lock 10$^{th}$ Axis Movement in Negative Direction | M2x361 | When this special M is ON, the axis will not be able to move in the negative direction. | R/W |
| Lock 11$^{th}$ Axis Movement in Negative Direction | M2x362 | When this special M is ON, the axis will not be able to move in the negative direction. | R/W |
| Lock 12$^{th}$ Axis Movement in Negative Direction | M2x363 | When this special M is ON, the axis will not be able to move in the negative direction. | R/W |
| Lock 13$^{th}$ Axis Movement in Negative Direction | M2x364 | When this special M is ON, the axis will not be able to move in the negative direction. | R/W |
| Lock 14$^{th}$ Axis Movement in Negative Direction | M2x365 | When this special M is ON, the axis will not be able to move in the negative direction. | R/W |
| Lock 15$^{th}$ Axis Movement in Negative Direction | M2x366 | When this special M is ON, the axis will not be able to move in the negative direction. | R/W |
| Lock 16$^{th}$ Axis Movement in Negative Direction | M2x367 | When this special M is ON, the axis will not be able to move in the negative direction. | R/W |
| Disable X Axis 1$^{st}$ Software Limit | M2x368 | Trigger this special M to disable the X axis 1$^{st}$ software limit. | R/W |
| Disable Y Axis 1$^{st}$ Software Limit | M2x369 | Trigger this special M to disable the Y axis 1$^{st}$ software limit. | R/W |
| Disable Z Axis 1$^{st}$ Software Limit | M2x370 | Trigger this special M to disable the Z axis 1$^{st}$ software limit. | R/W |
| Disable A Axis 1$^{st}$ Software Limit | M2x371 | Trigger this special M to disable the A axis 1$^{st}$ software limit. | R/W |
| Disable B Axis 1$^{st}$ Software Limit | M2x372 | Trigger this special M to disable the B axis 1$^{st}$ software limit. | R/W |
| Disable C Axis 1$^{st}$ Software Limit | M2x373 | Trigger this special M to disable the C axis 1$^{st}$ software limit. | R/W |
| Disable U Axis 1$^{st}$ Software Limit | M2x374 | Trigger this special M to disable the U axis 1$^{st}$ software limit. | R/W |
| Disable V Axis 1$^{st}$ Software Limit | M2x375 | Trigger this special M to disable the V axis 1$^{st}$ software limit. | R/W |
| Disable W Axis 1$^{st}$ Software Limit | M2x376 | Trigger this special M to disable the W axis 1$^{st}$ software limit. | R/W |
| Disable 10$^{th}$ Axis 1$^{st}$ Software Limit | M2x377 | Trigger this special M to disable the 10$^{th}$ axis 1$^{st}$ software limit. | R/W |
| Disable 11$^{th}$ Axis 1$^{st}$ Software Limit | M2x378 | Trigger this special M to disable the 11$^{th}$ axis 1$^{st}$ software limit. | R/W |
| Disable 12$^{th}$ Axis 1$^{st}$ Software Limit | M2x379 | Trigger this special M to disable the 12$^{th}$ axis 1$^{st}$ software limit. | R/W |
| Disable 13$^{th}$ Axis 1$^{st}$ Software Limit | M2x380 | Trigger this special M to disable the 13$^{th}$ axis 1$^{st}$ software limit. | R/W |
| Disable 14$^{th}$ Axis 1$^{st}$ Software Limit | M2x381 | Trigger this special M to disable the 14$^{th}$ axis 1$^{st}$ software limit. | R/W |

5

| Function Name | Special M | Description | Device |
|---|---|---|---|
| Disable 15th Axis 1st Software Limit | M2x382 | Trigger this special M to disable the 15th axis 1st software limit. | R/W |
| Disable 16th Axis 1st Software Limit | M2x383 | Trigger this special M to disable the 16th axis 1st software limit. | R/W |
| X Axis JOG Forward | M2x384 | Trigger this special M for X axis forward JOG or INC operation. | R/W |
| Y Axis JOG Forward | M2x385 | Trigger this special M for Y axis forward JOG or INC operation. | R/W |
| Z Axis JOG Forward | M2x386 | Trigger this special M for Z axis forward JOG or INC operation. | R/W |
| A Axis JOG Forward | M2x387 | Trigger this special M for A axis forward JOG or INC operation. | R/W |
| B Axis JOG Forward | M2x388 | Trigger this special M for B axis forward JOG or INC operation. | R/W |
| C Axis JOG Forward | M2x389 | Trigger this special M for C axis forward JOG or INC operation. | R/W |
| U Axis JOG Forward | M2x390 | Trigger this special M for U axis forward JOG or INC operation. | R/W |
| V Axis JOG Forward | M2x391 | Trigger this special M for V axis forward JOG or INC operation. | R/W |
| W Axis JOG Forward | M2x392 | Trigger this special M for W axis forward JOG or INC operation. | R/W |
| 10th Axis JOG Forward | M2x393 | Trigger this special M for 10th axis forward JOG or INC operation. | R/W |
| 11th Axis JOG Forward | M2x394 | Trigger this special M for 11th axis forward JOG or INC operation. | R/W |
| 12th Axis JOG Forward | M2x395 | Trigger this special M for 12th axis forward JOG or INC operation. | R/W |
| 13th Axis JOG Forward | M2x396 | Trigger this special M for 13th axis forward JOG or INC operation. | R/W |
| 14th Axis JOG Forward | M2x397 | Trigger this special M for 14th axis forward JOG or INC operation. | R/W |
| 15th Axis JOG Forward | M2x398 | Trigger this special M for 15th axis forward JOG or INC operation. | R/W |
| 16th Axis JOG Forward | M2x399 | Trigger this special M for 16th axis forward JOG or INC operation. | R/W |
| X Axis JOG Reverse | M2x400 | Trigger this special M for X axis reverse JOG or INC operation. | R/W |
| Y Axis JOG Reverse | M2x401 | Trigger this special M for Y axis reverse JOG or INC operation. | R/W |
| Z Axis JOG Reverse | M2x402 | Trigger this special M for Z axis reverse JOG or INC operation. | R/W |
| A Axis JOG Reverse | M2x403 | Trigger this special M for A axis reverse JOG or INC operation. | R/W |
| B Axis JOG Reverse | M2x404 | Trigger this special M for B axis reverse JOG or INC operation. | R/W |
| C Axis JOG Reverse | M2x405 | Trigger this special M for C axis reverse JOG or INC operation. | R/W |
| U Axis JOG Reverse | M2x406 | Trigger this special M for U axis reverse JOG or INC operation. | R/W |
| V Axis JOG Reverse | M2x407 | Trigger this special M for V axis reverse JOG or INC operation. | R/W |
| W Axis JOG Reverse | M2x408 | Trigger this special M for W axis reverse JOG or INC operation. | R/W |

| Function Name | Special M | Description | Device |
|---|---|---|---|
| 10th Axis JOG Reverse | M2x409 | Trigger this special M for 10th axis reverse JOG or INC operation. | R/W |
| 11th Axis JOG Reverse | M2x410 | Trigger this special M for 11th axis reverse JOG or INC operation. | R/W |
| 12th Axis JOG Reverse | M2x411 | Trigger this special M for 12th axis reverse JOG or INC operation. | R/W |
| 13th Axis JOG Reverse | M2x412 | Trigger this special M for 13th axis reverse JOG or INC operation. | R/W |
| 14th Axis JOG Reverse | M2x413 | Trigger this special M for 14th axis reverse JOG or INC operation. | R/W |
| 15th Axis JOG Reverse | M2x414 | Trigger this special M for 15th axis reverse JOG or INC operation. | R/W |
| 16th Axis JOG Reverse | M2x415 | Trigger this special M for 16th axis reverse JOG or INC operation. | R/W |
| MLC X Axis Control Mode | M2x416 | When this special M is ON, X axis will be in speed mode. When this special M is OFF, X axis will be in position mode. | R/W |
| MLC Y Axis Control Mode | M2x417 | When this special M is ON, Y axis will be in speed mode. When this special M is OFF, Y axis will be in position mode. | R/W |
| MLC Z Axis Control Mode | M2x418 | When this special M is ON, Z axis will be in speed mode. When this special M is OFF, Z axis will be in position mode. | R/W |
| MLC A Axis Control Mode | M2x419 | When this special M is ON, A axis will be in speed mode. When this special M is OFF, A axis will be in position mode. | R/W |
| MLC B Axis Control Mode | M2x420 | When this special M is ON, B axis will be in speed mode. When this special M is OFF, B axis will be in position mode. | R/W |
| MLC C Axis Control Mode | M2x421 | When this special M is ON, C axis will be in speed mode. When this special M is OFF, C axis will be in position mode. | R/W |
| MLC U Axis Control Mode | M2x422 | When this special M is ON, U axis will be in speed mode. When this special M is OFF, U axis will be in position mode. | R/W |
| MLC V Axis Control Mode | M2x423 | When this special M is ON, V axis will be in speed mode. When this special M is OFF, V axis will be in position mode. | R/W |
| MLC W Axis Control Mode | M2x424 | When this special M is ON, W axis will be in speed mode. When this special M is OFF, W axis will be in position mode. | R/W |
| MLC 10th Axis Control Mode | M2x425 | When this special M is ON, 10th axis will be in speed mode. When this special M is OFF, 10th axis will be in position mode. | R/W |
| MLC 11th Axis Control Mode | M2x426 | When this special M is ON, 11th axis will be in speed mode. When this special M is OFF, 11th axis will be in position mode. | R/W |
| MLC 12th Axis Control Mode | M2x427 | When this special M is ON, 12th axis will be in speed mode. When this special M is OFF, 12th axis will be in position mode. | R/W |
| MLC 13th Axis Control Mode | M2x428 | When this special M is ON, 13th axis will be in speed mode. When this special M is OFF, 13th axis will be in position mode. | R/W |
| MLC 14th Axis Control Mode | M2x429 | When this special M is ON, 14th axis will be in speed mode. When this special M is OFF, 14th axis will be in position mode. | R/W |

5

| Function Name | Special M | Description | Device |
|---|---|---|---|
| MLC 15th Axis Control Mode | M2x430 | When this special M is ON, 15th axis will be in speed mode. When this special M is OFF, 15th axis will be in position mode. | R/W |
| MLC 16th Axis Control Mode | M2x431 | When this special M is ON, 16th axis will be in speed mode. When this special M is OFF, 16th axis will be in position mode. | R/W |
| NC / MLC Axis Switching (X Axis) | M2x432 | When this special M is ON, the axis will be controlled by MLC. When this special M is OFF, the axis will be controlled from NC command. | R/W |
| NC / MLC Axis Switching (Y Axis) | M2x433 | When this special M is ON, the axis will be controlled by MLC. When this special M is OFF, the axis will be controlled from NC command. | R/W |
| NC / MLC Axis Switching (Z Axis) | M2x434 | When this special M is ON, the axis will be controlled by MLC. When this special M is OFF, the axis will be controlled from NC command. | R/W |
| NC / MLC Axis Switching (A Axis) | M2x435 | When this special M is ON, the axis will be controlled by MLC. When this special M is OFF, the axis will be controlled from NC command. | R/W |
| NC / MLC Axis Switching (B Axis) | M2x436 | When this special M is ON, the axis will be controlled by MLC. When this special M is OFF, the axis will be controlled from NC command. | R/W |
| NC / MLC Axis Switching (C Axis) | M2x437 | When this special M is ON, the axis will be controlled by MLC. When this special M is OFF, the axis will be controlled from NC command. | R/W |
| NC / MLC Axis Switching (U Axis) | M2x438 | When this special M is ON, the axis will be controlled by MLC. When this special M is OFF, the axis will be controlled from NC command. | R/W |
| NC / MLC Axis Switching (V Axis) | M2x439 | When this special M is ON, the axis will be controlled by MLC. When this special M is OFF, the axis will be controlled from NC command. | R/W |
| NC / MLC Axis Switching (W Axis) | M2x440 | When this special M is ON, the axis will be controlled by MLC. When this special M is OFF, the axis will be controlled from NC command. | R/W |
| NC / MLC Axis Switching (10th Axis) | M2x441 | When this special M is ON, the axis will be controlled by MLC. When this special M is OFF, the axis will be controlled from NC command. | R/W |
| NC / MLC Axis Switching (11th Axis) | M2x442 | When this special M is ON, the axis will be controlled by MLC. When this special M is OFF, the axis will be controlled from NC command. | R/W |
| NC / MLC Axis Switching (12th Axis) | M2x443 | When this special M is ON, the axis will be controlled by MLC. When this special M is OFF, the axis will be controlled from NC command. | R/W |
| NC / MLC Axis Switching (13th Axis) | M2x444 | When this special M is ON, the axis will be controlled by MLC. When this special M is OFF, the axis will be controlled from NC command. | R/W |
| NC / MLC Axis Switching (14th Axis) | M2x445 | When this special M is ON, the axis will be controlled by MLC. When this special M is OFF, the axis will be controlled from NC command. | R/W |
| NC / MLC Axis Switching (15th Axis) | M2x446 | When this special M is ON, the axis will be controlled by MLC. When this special M is OFF, the axis will be controlled from NC command. | R/W |

| Function Name | Special M | Description | Device |
|---|---|---|---|
| NC / MLC Axis Switching (16th Axis) | M2x447 | When this special M is ON, the axis will be controlled by MLC. When this special M is OFF, the axis will be controlled from NC command. | R/W |
| Trigger X Axis' Movement (MLC Axis) | M2x448 | Trigger this special M for MLC X axis movement. | R/W |
| Trigger Y Axis' Movement (MLC Axis) | M2x449 | Trigger this special M for MLC Y axis movement. | R/W |
| Trigger Z Axis' Movement (MLC Axis) | M2x450 | Trigger this special M for MLC Z axis movement. | R/W |
| Trigger A Axis' Movement (MLC Axis) | M2x451 | Trigger this special M for MLC A axis movement. | R/W |
| Trigger B Axis' Movement (MLC Axis) | M2x452 | Trigger this special M for MLC B axis movement. | R/W |
| Trigger C Axis' Movement (MLC Axis) | M2x453 | Trigger this special M for MLC C axis movement. | R/W |
| Trigger U Axis' Movement (MLC Axis) | M2x454 | Trigger this special M for MLC U axis movement. | R/W |
| Trigger V Axis' Movement (MLC Axis) | M2x455 | Trigger this special M for MLC V axis movement. | R/W |
| Trigger W Axis' Movement (MLC Axis) | M2x456 | Trigger this special M for MLC W axis movement. | R/W |
| Trigger 10th Axis' Movement (MLC Axis) | M2x457 | Trigger this special M for MLC 10th axis movement. | R/W |
| Trigger 11th Axis' Movement (MLC Axis) | M2x458 | Trigger this special M for MLC 11th axis movement. | R/W |
| Trigger 12th Axis' Movement (MLC Axis) | M2x459 | Trigger this special M for MLC 12th axis movement. | R/W |
| Trigger 13th Axis' Movement (MLC Axis) | M2x460 | Trigger this special M for MLC 13th axis movement. | R/W |
| Trigger 14th Axis' Movement (MLC Axis) | M2x461 | Trigger this special M for MLC 14th axis movement. | R/W |
| Trigger 15th Axis' Movement (MLC Axis) | M2x462 | Trigger this special M for MLC 15th axis movement. | R/W |
| Trigger 16th Axis' Movement (MLC Axis) | M2x463 | Trigger this special M for MLC 16th axis movement. | R/W |
| MLC Axis Command Type of X Axis | M2x464 | When this special M is ON, the **D2x256** will be incremental movement. | R/W |

| Function Name | Special M | Description | Device |
|---|---|---|---|
| MLC Axis Command Type of Y Axis | M2x465 | When this special M is ON, the **D2x258** will be incremental movement. | R/W |
| MLC Axis Command Type of Z Axis | M2x466 | When this special M is ON, the **D2x260** will be incremental movement. | R/W |
| MLC Axis Command Type of A Axis | M2x467 | When this special M is ON, the **D2x262** will be incremental movement. | R/W |
| MLC Axis Command Type of B Axis | M2x468 | When this special M is ON, the **D2x264** will be incremental movement. | R/W |
| MLC Axis Command Type of C Axis | M2x469 | When this special M is ON, the **D2x266** will be incremental movement. | R/W |
| MLC Axis Command Type of U Axis | M2x470 | When this special M is ON, the **D2x268** will be incremental movement. | R/W |
| MLC Axis Command Type of V Axis | M2x471 | When this special M is ON, the **D2x270** will be incremental movement. | R/W |
| MLC Axis Command Type of W Axis | M2x472 | When this special M is ON, the **D2x272** will be incremental movement. | R/W |
| MLC Axis Command Type of 10$^{th}$ Axis | M2x473 | When this special M is ON, the **D2x274** will be incremental movement. | R/W |
| MLC Axis Command Type of 11$^{th}$ Axis | M2x474 | When this special M is ON, the **D2x276** will be incremental movement. | R/W |
| MLC Axis Command Type of 12$^{th}$ Axis | M2x475 | When this special M is ON, the **D2x278** will be incremental movement. | R/W |
| MLC Axis Command Type of 13$^{th}$ Axis | M2x476 | When this special M is ON, the **D2x280** will be incremental movement. | R/W |
| MLC Axis Command Type of 14$^{th}$ Axis | M2x477 | When this special M is ON, the **D2x282** will be incremental movement. | R/W |
| MLC Axis Command Type of 15$^{th}$ Axis | M2x478 | When this special M is ON, the **D2x284** will be incremental movement. | R/W |
| MLC Axis Command Type of 16$^{th}$ Axis | M2x479 | When this special M is ON, the **D2x286** will be incremental movement. | R/W |
| 1$^{st}$ Spindle Forward Rotation | M2x704 | Sets the 1$^{st}$ spindle to rotate in forward direction. | R/W |
| 1$^{st}$ Spindle Reverse Rotation | M2x705 | Sets the 1$^{st}$ spindle to rotate in reverse direction. | R/W |
| 1$^{st}$ Spindle Positioning Control | M2x706 | Positioning the 1$^{st}$ spindle. | R/W |
| 1$^{st}$ Spindle Retraction After Tapping | M2x707 | Retracts the 1$^{st}$ spindle after tapping. | R/W |
| Switching C / S Axis of 1$^{st}$ Spindle | M2x708 | Trigger to switch between C or S axes for the spindle. | R/W |
| 1$^{st}$ Spindle Command Source | M2x710 | When this special M is ON, the speed command for the 1$^{st}$ spindle will refer to **D2x024**. When this special M is OFF, the speed command will be based on the S command from the NC program. | R/W |
| 1$^{st}$ Spindle Alarm | M2x711 | When the MLC receive an alarm signal from digital input and then trigger this special M, the system will stop NC process and return alarm code [0x0C0A MLC activate spindle alarm] of 1$^{st}$ spindle. | R/W |
| 2$^{nd}$ Spindle Forward Rotation | M2x720 | Sets the 2$^{nd}$ spindle to rotate in forward direction. | R/W |
| 2$^{nd}$ Spindle Reverse Rotation | M2x721 | Sets the 2$^{nd}$ spindle to rotate in reverse direction. | R/W |
| 2$^{nd}$ Spindle Positioning Control | M2x722 | Positioning the 2$^{nd}$ spindle. | R/W |

| Function Name | Special M | Description | Device |
|---|---|---|---|
| 2nd Spindle Retraction After Tapping | M2x723 | Retracts the 2nd spindle after tapping. | R/W |
| Switching C / S Axis of 2nd Spindle | M2x724 | Trigger to switch between C or S axes for the spindle. | R/W |
| 2nd Spindle Command Source | M2x726 | When this special M is ON, the speed command for the 2nd spindle will refer to **D2x030**. When this special M is OFF, the speed command will be based on the S command from the NC program. | R/W |
| 2nd Spindle Alarm | M2x727 | When the MLC receive an alarm signal from digital input and then trigger this special M, the system will stop NC process and return alarm code [0x0C0A MLC activate spindle alarm] of 2nd spindle. | R/W |
| 3rd Spindle Forward Rotation | M2x736 | Sets the 3rd spindle to rotate in forward direction. | R/W |
| 3rd Spindle Reverse Rotation | M2x737 | Sets the 3rd spindle to rotate in reverse direction. | R/W |
| 3rd Spindle Positioning Control | M2x738 | Positioning the 3rd spindle. | R/W |
| 3rd Spindle Retraction After Tapping | M2x739 | Retracts the 3rd spindle after tapping. | R/W |
| Switching C / S Axis of 3rd Spindle | M2x740 | Trigger to switch between C or S axes for the spindle. | R/W |
| 3rd Spindle Command Source | M2x742 | When this special M is ON, the speed command for the 3rd spindle will refer to **D2x320**. When this special M is OFF, the speed command will be based on the S command from the NC program. | R/W |
| 3rd Spindle Alarm | M2x743 | When the MLC receive an alarm signal from digital input and then trigger this special M, the system will stop NC process and return alarm code [0x0C0A MLC activate spindle alarm] of 3rd spindle. | R/W |
| 4th Spindle Forward Rotation | M2x752 | Sets the 4th spindle to rotate in forward direction. | R/W |
| 4th Spindle Reverse Rotation | M2x753 | Sets the 4th spindle to rotate in reverse direction. | R/W |
| 4th Spindle Positioning Control | M2x754 | Positioning the 4th spindle. | R/W |
| 4th Spindle Retraction After Tapping | M2x755 | Retracts the 4th spindle after tapping. | R/W |
| Switching C / S Axis of 4th Spindle | M2x756 | Trigger to switch between C or S axes for the spindle. | R/W |
| 4th Spindle Command Source | M2x758 | When this special M is ON, the speed command for the 4th spindle will refer to **D2x326**. When this special M is OFF, the speed command will be based on the S command from the NC program. | R/W |
| 4th Spindle Alarm | M2x759 | When the MLC receive an alarm signal from digital input and then trigger this special M, the system will stop NC process and return alarm code [0x0C0A MLC activate spindle alarm] of 4th spindle. | R/W |
| 5th Spindle Forward Rotation | M2x768 | Sets the 5th spindle to rotate in forward direction. | R/W |
| 5th Spindle Reverse Rotation | M2x769 | Sets the 5th spindle to rotate in reverse direction. | R/W |
| 5th Spindle Positioning Control | M2x770 | Positioning the 5th spindle. | R/W |
| 5th Spindle Retraction After Tapping | M2x771 | Retracts the 5th spindle after tapping. | R/W |

| Function Name | Special M | Description | Device |
|---|---|---|---|
| Switching C / S Axis of 5th Spindle | M2x772 | Trigger to switch between C or S axes for the spindle. | R/W |
| 5th Spindle Command Source | M2x774 | When this special M is ON, the speed command for the 5th spindle will refer to **D2x332**. When this special M is OFF, the speed command will be based on the S command from the NC program. | R/W |
| 5th Spindle Alarm | M2x775 | When the MLC receive an alarm signal from digital input and then trigger this special M, the system will stop NC process and return alarm code [0x0C0A MLC activate spindle alarm] of 5th spindle. | R/W |
| 6th Spindle Forward Rotation | M2x784 | Sets the 6th spindle to rotate in forward direction. | R/W |
| 6th Spindle Reverse Rotation | M2x785 | Sets the 6th spindle to rotate in reverse direction. | R/W |
| 6th Spindle Positioning Control | M2x786 | Positioning the 6th spindle. | R/W |
| 6th Spindle Retraction After Tapping | M2x787 | Retracts the 6th spindle after tapping. | R/W |
| Switching C / S Axis of 6th Spindle | M2x788 | Trigger to switch between C or S axes for the spindle. | R/W |
| 6th Spindle Command Source | M2x790 | When this special M is ON, the speed command for the 6th spindle will refer to **D2x338**. When this special M is OFF, the speed command will be based on the S command from the NC program. | R/W |
| 6th Spindle Alarm | M2x791 | When the MLC receive an alarm signal from digital input and then trigger this special M, the system will stop NC process and return alarm code [0x0C0A MLC activate spindle alarm] of 6th spindle. | R/W |
| 7th Spindle Forward Rotation | M2x800 | Sets the 7th spindle to rotate in forward direction. | R/W |
| 7th Spindle Reverse Rotation | M2x801 | Sets the 7th spindle to rotate in reverse direction. | R/W |
| 7th Spindle Positioning Control | M2x802 | Positioning the 7th spindle. | R/W |
| 7th Spindle Retraction After Tapping | M2x803 | Retracts the 7th spindle after tapping. | R/W |
| Switching C / S Axis of 7th Spindle | M2x804 | Trigger to switch between C or S axes for the spindle. | R/W |
| 7th Spindle Command Source | M2x806 | When this special M is ON, the speed command for the 7th spindle will refer to **D2x344**. When this special M is OFF, the speed command will be based on the S command from the NC program. | R/W |
| 7th Spindle Alarm | M2x807 | When the MLC receive an alarm signal from digital input and then trigger this special M, the system will stop NC process and return alarm code [0x0C0A MLC activate spindle alarm] of 7th spindle. | R/W |
| 8th Spindle Forward Rotation | M2x816 | Sets the 8th spindle to rotate in forward direction. | R/W |
| 8th Spindle Reverse Rotation | M2x817 | Sets the 8th spindle to rotate in reverse direction. | R/W |
| 8th Spindle Positioning Control | M2x818 | Positioning the 8th spindle. | R/W |
| 8th Spindle Retraction After Tapping | M2x819 | Retracts the 8th spindle after tapping. | R/W |
| Switching C / S Axis of 8th Spindle | M2x820 | Trigger to switch between C or S axes for the spindle. | R/W |

| Function Name | Special M | Description | Device |
|---|---|---|---|
| 8$^{th}$ Spindle Command Source | M2x822 | When this special M is ON, the speed command for the 8$^{th}$ spindle will refer to **D2x350**. When this special M is OFF, the speed command will be based on the S command from the NC program. | R/W |
| 8$^{th}$ Spindle Alarm | M2x807 | When the MLC receive an alarm signal from digital input and then trigger this special M, the system will stop NC process and return alarm code [0x0C0A MLC activate spindle alarm] of 8$^{th}$ spindle. | R/W |
| Time Pulse (500 ms) | M29032 | This special M will continuously switch between ON and OFF for 500 ms each. | R |
| Time Pulse (50 ms) | M29033 | This special M will continuously switch between ON and OFF for 50 ms each. | R |
| Remaining OFF | M29034 | After the controller system is ready, this special M will always remain OFF. | R |
| Remaining ON | M29035 | After the controller system is ready, this special M will always remain ON. | R |

### 5.2.1.2 M49000 to M49899

| Function Name | Special M | Description | Device |
|---|---|---|---|
| Program Lock | M49000 | When this special M is ON, the system will lock the current main program and not allow changes or edits. | R/W |

5

## 5.2.2 List of special M (System status)

### 5.2.2.1 M30000 to M38999

| Function Name | Special M | Description | Device |
|---|---|---|---|
| EtherCAT Communication Status | M30000 | After EtherCAT communication is initialized and without errors, this special M will be ON.<br>Note: this special M status only corresponds to the EtherCAT communication status, it does not indicate Servo ON. | R |
| HSI 1 Status | M30016 | When the system executes the skip command G31P1, this special M will be also set to ON by the NC. | R |
| HSI 2 Status | M30017 | When the system executes the skip command G31P2, this special M will be also set to ON by the NC. | R |
| HSI 3 Status | M30018 | When the system executes the skip command G31P3, this special M will be also set to ON by the NC. | R |
| HSI 4 Status | M30019 | When the system executes the skip command G31P4, this special M will be also set to ON by the NC. | R |
| HSI 5 Status | M30020 | When the system executes the skip command G31P5, this special M will be also set to ON by the NC. | R |
| HSI 6 Status | M30021 | When the system executes the skip command G31P6, this special M will be also set to ON by the NC. | R |
| HSI 7 Status | M30022 | When the system executes the skip command G31P7, this special M will be also set to ON by the NC. | R |
| HSI 8 Status | M30023 | When the system executes the skip command G31P8, this special M will be also set to ON by the NC. | R |
| EIO 1 Connection Status | M30032 | This special M is ON when the EtherCAT remote module node ID 501 is connected. | R |
| EIO 2 Connection Status | M30033 | This special M is ON when the EtherCAT remote module node ID 502 is connected. | R |
| EIO 3 Connection Status | M30034 | This special M is ON when the EtherCAT remote module node ID 503 is connected. | R |
| EIO 4 Connection Status | M30035 | This special M is ON when the EtherCAT remote module node ID 504 is connected. | R |
| EIO 5 Connection Status | M30036 | This special M is ON when the EtherCAT remote module node ID 505 is connected. | R |
| EIO 6 Connection Status | M30037 | This special M is ON when the EtherCAT remote module node ID 506 is connected. | R |
| EIO 7 Connection Status | M30038 | This special M is ON when the EtherCAT remote module node ID 507 is connected. | R |
| EIO 8 Connection Status | M30039 | This special M is ON when the EtherCAT remote module node ID 508 is connected. | R |
| EIO 9 Connection Status | M30040 | This special M is ON when the EtherCAT remote module node ID 509 is connected. | R |
| EIO 10 Connection Status | M30041 | This special M is ON when the EtherCAT remote module node ID 510 is connected. | R |
| EIO 11 Connection Status | M30042 | This special M is ON when the EtherCAT remote module node ID 511 is connected. | R |
| EIO 12 Connection Status | M30043 | This special M is ON when the EtherCAT remote module node ID 512 is connected. | R |
| EIO 13 Connection Status | M30044 | This special M is ON when the EtherCAT remote module node ID 513 is connected. | R |
| EIO 14 Connection Status | M30045 | This special M is ON when the EtherCAT remote module node ID 514 is connected. | R |
| EIO 15 Connection Status | M30046 | This special M is ON when the EtherCAT remote module node ID 515 is connected. | R |

| Function Name | Special M | Description | Device |
|---|---|---|---|
| EIO 16 Connection Status | M30047 | This special M is ON when the EtherCAT remote module node ID 516 is connected. | R |
| EIO 17 Connection Status | M30048 | This special M is ON when the EtherCAT remote module node ID 517 is connected. | R |
| EIO 18 Connection Status | M30049 | This special M is ON when the EtherCAT remote module node ID 518 is connected. | R |
| EIO 19 Connection Status | M30050 | This special M is ON when the EtherCAT remote module node ID 519 is connected. | R |
| EIO 20 Connection Status | M30051 | This special M is ON when the EtherCAT remote module node ID 520 is connected. | R |
| NC Channel 1 Enable Status | M30080 | This special M is ON when the 1st NC channel is enabled. | R |
| NC Channel 2 Enable Status | M30081 | This special M is ON when the 2nd NC channel is enabled. | R |
| NC Channel 3 Enable Status | M30082 | This special M is ON when the 3rd NC channel is enabled. | R |
| NC Channel 4 Enable Status | M30083 | This special M is ON when the 4th NC channel is enabled. | R |
| Spindle 1 DI Status | M30096 | This special M is normal close type relay, means it will always stay ON until Pin 8 on the Spindle 1 connector has turn ON, and this special M will be OFF instead. | R |
| Spindle 2 DI Status | M30097 | This special M is normal close type relay, means it will always stay ON until Pin 8 on the Spindle 2 connector has turn ON, and this special M will be OFF instead. | R |
| G900 Command Control M_1 | M30200 | These Special M relay can be controlled by the G900 P_ Q_ command. This G900 will not stop look-ahead or pause the NC process.<br>P: command value indicates the M address directly. From 30200 as M30200 to 30399 as M30399. When the value exceeds this range, the system will not return an error, but not the other special M will be triggered as well.<br>Q: control the M relay status by 0 as OFF and any not 0 value will set as ON. | R |
| G900 Command Control M_2 | M30201 | | R |
| G900 Command Control M_3 | M30202 | | R |
| G900 Command Control M_4 | M30203 | | R |
| G900 Command Control M_5 | M30204 | | R |
| G900 Command Control M_6 | M30205 | | R |
| G900 Command Control M_7 | M30206 | | R |
| G900 Command Control M_8 | M30207 | | R |
| G900 Command Control M_9 | M30208 | | R |
| G900 Command Control M_10 | M30209 | | R |
| G900 Command Control M_11 | M30210 | | R |
| G900 Command Control M_12 | M30211 | | R |
| G900 Command Control M_13 | M30212 | | R |
| G900 Command Control M_14 | M30213 | | R |
| G900 Command Control M_15 | M30214 | | R |

| Function Name | Special M | Description | Device |
|---|---|---|---|
| G900 Command Control M_16 | M30215 | These Special M relay can be controlled by the G900 P_ Q_ command. This G900 will not stop look-ahead or pause the NC process.<br>P: command value indicates the M address directly. From 30200 as M30200 to 30399 as M30399. When the value exceeds this range, the system will not return an error, but not the other special M will be triggered as well.<br>Q: control the M relay status by 0 as OFF and any not 0 value will set as ON. | R |
| G900 Command Control M_17 | M30216 | | R |
| G900 Command Control M_18 | M30217 | | R |
| G900 Command Control M_19 | M30218 | | R |
| G900 Command Control M_20 | M30219 | | R |
| G900 Command Control M_21 | M30220 | | R |
| G900 Command Control M_22 | M30221 | | R |
| G900 Command Control M_23 | M30222 | | R |
| G900 Command Control M_24 | M30223 | | R |
| G900 Command Control M_25 | M30224 | | R |
| G900 Command Control M_26 | M30225 | | R |
| G900 Command Control M_27 | M30226 | | R |
| G900 Command Control M_28 | M30227 | | R |
| G900 Command Control M_29 | M30228 | | R |
| G900 Command Control M_30 | M30229 | | R |
| G900 Command Control M_31 | M30230 | | R |
| G900 Command Control M_32 | M30231 | | R |
| G900 Command Control M_33 | M30232 | | R |
| G900 Command Control M_34 | M30233 | | R |
| G900 Command Control M_35 | M30234 | | R |
| G900 Command Control M_36 | M30235 | | R |
| G900 Command Control M_37 | M30236 | | R |
| G900 Command Control M_38 | M30237 | | R |
| G900 Command Control M_39 | M30238 | | R |
| G900 Command Control M_40 | M30239 | | R |
| G900 Command Control M_41 | M30240 | | R |
| G900 Command Control M_42 | M30241 | | R |

| Function Name | Special M | Description | Device |
|---|---|---|---|
| G900 Command Control M_43 | M30242 | These Special M relay can be controlled by the G900 P_ Q_ command. This G900 will not stop look-ahead or pause the NC process.<br>P: command value indicates the M address directly. From 30200 as M30200 to 30399 as M30399. When the value exceeds this range, the system will not return an error, but not the other special M will be triggered as well.<br>Q: control the M relay status by 0 as OFF and any not 0 value will set as ON. | R |
| G900 Command Control M_44 | M30243 | | R |
| G900 Command Control M_45 | M30244 | | R |
| G900 Command Control M_46 | M30245 | | R |
| G900 Command Control M_47 | M30246 | | R |
| G900 Command Control M_48 | M30247 | | R |
| G900 Command Control M_49 | M30248 | | R |
| G900 Command Control M_50 | M30249 | | R |
| G900 Command Control M_51 | M30250 | | R |
| G900 Command Control M_52 | M30251 | | R |
| G900 Command Control M_53 | M30252 | | R |
| G900 Command Control M_54 | M30253 | | R |
| G900 Command Control M_55 | M30254 | | R |
| G900 Command Control M_56 | M30255 | | R |
| G900 Command Control M_57 | M30256 | | R |
| G900 Command Control M_58 | M30257 | | R |
| G900 Command Control M_59 | M30258 | | R |
| G900 Command Control M_60 | M30259 | | R |
| G900 Command Control M_61 | M30260 | | R |
| G900 Command Control M_62 | M30261 | | R |
| G900 Command Control M_63 | M30262 | | R |
| G900 Command Control M_64 | M30263 | | R |
| G900 Command Control M_65 | M30264 | | R |
| G900 Command Control M_66 | M30265 | | R |
| G900 Command Control M_67 | M30266 | | R |
| G900 Command Control M_68 | M30267 | | R |
| G900 Command Control M_69 | M30268 | | R |

| Function Name | Special M | Description | Device |
|---|---|---|---|
| G900 Command Control M_70 | M30269 | These Special M relay can be controlled by the G900 P_ Q_ command. This G900 will not stop look-ahead or pause the NC process.<br>P: command value indicates the M address directly. From 30200 as M30200 to 30399 as M30399. When the value exceeds this range, the system will not return an error, but not the other special M will be triggered as well.<br>Q: control the M relay status by 0 as OFF and any not 0 value will set as ON. | R |
| G900 Command Control M_71 | M30270 | | R |
| G900 Command Control M_72 | M30271 | | R |
| G900 Command Control M_73 | M30272 | | R |
| G900 Command Control M_74 | M30273 | | R |
| G900 Command Control M_75 | M30274 | | R |
| G900 Command Control M_76 | M30275 | | R |
| G900 Command Control M_77 | M30276 | | R |
| G900 Command Control M_78 | M30277 | | R |
| G900 Command Control M_79 | M30278 | | R |
| G900 Command Control M_80 | M30279 | | R |
| G900 Command Control M_81 | M30280 | | R |
| G900 Command Control M_82 | M30281 | | R |
| G900 Command Control M_83 | M30282 | | R |
| G900 Command Control M_84 | M30283 | | R |
| G900 Command Control M_85 | M30284 | | R |
| G900 Command Control M_86 | M30285 | | R |
| G900 Command Control M_87 | M30286 | | R |
| G900 Command Control M_88 | M30287 | | R |
| G900 Command Control M_89 | M30288 | | R |
| G900 Command Control M_90 | M30289 | | R |
| G900 Command Control M_91 | M30290 | | R |
| G900 Command Control M_92 | M30291 | | R |
| G900 Command Control M_93 | M30292 | | R |
| G900 Command Control M_94 | M30293 | | R |
| G900 Command Control M_95 | M30294 | | R |
| G900 Command Control M_96 | M30295 | | R |

| Function Name | Special M | Description | Device |
|---|---|---|---|
| G900 Command Control M_97 | M30296 | These Special M relay can be controlled by the G900 P_ Q_ command. This G900 will not stop look-ahead or pause the NC process.<br>P: command value indicates the M address directly. From 30200 as M30200 to 30399 as M30399. When the value exceeds this range, the system will not return an error, but not the other special M will be triggered as well.<br>Q: control the M relay status by 0 as OFF and any not 0 value will set as ON. | R |
| G900 Command Control M_98 | M30297 | | R |
| G900 Command Control M_99 | M30298 | | R |
| G900 Command Control M_100 | M30299 | | R |
| G900 Command Control M_101 | M30300 | | R |
| G900 Command Control M_102 | M30301 | | R |
| G900 Command Control M_103 | M30302 | | R |
| G900 Command Control M_104 | M30303 | | R |
| G900 Command Control M_105 | M30304 | | R |
| G900 Command Control M_106 | M30305 | | R |
| G900 Command Control M_107 | M30306 | | R |
| G900 Command Control M_108 | M30307 | | R |
| G900 Command Control M_109 | M30308 | | R |
| G900 Command Control M_110 | M30309 | | R |
| G900 Command Control M_111 | M30310 | | R |
| G900 Command Control M_112 | M30311 | | R |
| G900 Command Control M_113 | M30312 | | R |
| G900 Command Control M_114 | M30313 | | R |
| G900 Command Control M_115 | M30314 | | R |
| G900 Command Control M_116 | M30315 | | R |
| G900 Command Control M_117 | M30316 | | R |
| G900 Command Control M_118 | M30317 | | R |
| G900 Command Control M_119 | M30318 | | R |
| G900 Command Control M_120 | M30319 | | R |
| G900 Command Control M_121 | M30320 | | R |
| G900 Command Control M_122 | M30321 | | R |
| G900 Command Control M_123 | M30322 | | R |

| Function Name | Special M | Description | Device |
|---|---|---|---|
| G900 Command Control M_124 | M30323 | These Special M relay can be controlled by the G900 P_ Q_ command. This G900 will not stop look-ahead or pause the NC process.<br>P: command value indicates the M address directly. From 30200 as M30200 to 30399 as M30399. When the value exceeds this range, the system will not return an error, but not the other special M will be triggered as well.<br>Q: control the M relay status by 0 as OFF and any not 0 value will set as ON. | R |
| G900 Command Control M_125 | M30324 | | R |
| G900 Command Control M_126 | M30325 | | R |
| G900 Command Control M_127 | M30326 | | R |
| G900 Command Control M_128 | M30327 | | R |
| G900 Command Control M_129 | M30328 | | R |
| G900 Command Control M_130 | M30329 | | R |
| G900 Command Control M_131 | M30330 | | R |
| G900 Command Control M_132 | M30331 | | R |
| G900 Command Control M_133 | M30332 | | R |
| G900 Command Control M_134 | M30333 | | R |
| G900 Command Control M_135 | M30334 | | R |
| G900 Command Control M_136 | M30335 | | R |
| G900 Command Control M_137 | M30336 | | R |
| G900 Command Control M_138 | M30337 | | R |
| G900 Command Control M_139 | M30338 | | R |
| G900 Command Control M_140 | M30339 | | R |
| G900 Command Control M_141 | M30340 | | R |
| G900 Command Control M_142 | M30341 | | R |
| G900 Command Control M_143 | M30342 | | R |
| G900 Command Control M_144 | M30343 | | R |
| G900 Command Control M_145 | M30344 | | R |
| G900 Command Control M_146 | M30345 | | R |
| G900 Command Control M_147 | M30346 | | R |
| G900 Command Control M_148 | M30347 | | R |
| G900 Command Control M_149 | M30348 | | R |
| G900 Command Control M_150 | M30349 | | R |

| Function Name | Special M | Description | Device |
|---|---|---|---|
| G900 Command Control M_151 | M30350 | | R |
| G900 Command Control M_152 | M30351 | | R |
| G900 Command Control M_153 | M30352 | | R |
| G900 Command Control M_154 | M30353 | | R |
| G900 Command Control M_155 | M30354 | | R |
| G900 Command Control M_156 | M30355 | | R |
| G900 Command Control M_157 | M30356 | | R |
| G900 Command Control M_158 | M30357 | | R |
| G900 Command Control M_159 | M30358 | | R |
| G900 Command Control M_160 | M30359 | | R |
| G900 Command Control M_161 | M30360 | | R |
| G900 Command Control M_162 | M30361 | These Special M relay can be controlled by the G900 P_ Q_ command. This G900 will not stop look-ahead or pause the NC process.<br>P: command value indicates the M address directly. From 30200 as M30200 to 30399 as M30399. When the value exceeds this range, the system will not return an error, but not the other special M will be triggered as well.<br>Q: control the M relay status by 0 as OFF and any not 0 value will set as ON. | R |
| G900 Command Control M_163 | M30362 | | R |
| G900 Command Control M_164 | M30363 | | R |
| G900 Command Control M_165 | M30364 | | R |
| G900 Command Control M_166 | M30365 | | R |
| G900 Command Control M_167 | M30366 | | R |
| G900 Command Control M_168 | M30367 | | R |
| G900 Command Control M_169 | M30368 | | R |
| G900 Command Control M_170 | M30369 | | R |
| G900 Command Control M_171 | M30370 | | R |
| G900 Command Control M_172 | M30371 | | R |
| G900 Command Control M_173 | M30372 | | R |
| G900 Command Control M_174 | M30373 | | R |
| G900 Command Control M_175 | M30374 | | R |
| G900 Command Control M_176 | M30375 | | R |
| G900 Command Control M_177 | M30376 | | R |

5

| Function Name | Special M | Description | Device |
|---|---|---|---|
| G900 Command Control M_178 | M30377 | These Special M relay can be controlled by the G900 P_ Q_ command. This G900 will not stop look-ahead or pause the NC process.<br>P: command value indicates the M address directly. From 30200 as M30200 to 30399 as M30399. When the value exceeds this range, the system will not return an error, but not the other special M will be triggered as well.<br>Q: control the M relay status by 0 as OFF and any not 0 value will set as ON. | R |
| G900 Command Control M_179 | M30378 | | R |
| G900 Command Control M_180 | M30379 | | R |
| G900 Command Control M_181 | M30380 | | R |
| G900 Command Control M_182 | M30381 | | R |
| G900 Command Control M_183 | M30382 | | R |
| G900 Command Control M_184 | M30383 | | R |
| G900 Command Control M_185 | M30384 | | R |
| G900 Command Control M_186 | M30385 | | R |
| G900 Command Control M_187 | M30386 | | R |
| G900 Command Control M_188 | M30387 | | R |
| G900 Command Control M_189 | M30388 | | R |
| G900 Command Control M_190 | M30389 | | R |
| G900 Command Control M_191 | M30390 | | R |
| G900 Command Control M_192 | M30391 | | R |
| G900 Command Control M_193 | M30392 | | R |
| G900 Command Control M_194 | M30393 | | R |
| G900 Command Control M_195 | M30394 | | R |
| G900 Command Control M_196 | M30395 | | R |
| G900 Command Control M_197 | M30396 | | R |
| G900 Command Control M_198 | M30397 | | R |
| G900 Command Control M_199 | M30398 | | R |
| G900 Command Control M_200 | M30399 | | R |
| AUTO | M3x000 | This special M is ON when the NC system is in AUTO mode. | R |
| EDIT | M3x001 | This special M is ON when the NC system is in EDIT mode. | R |
| MDI | M3x002 | This special M is ON when the NC system is in MDI mode. | R |
| MPG | M3x003 | This special M is ON when the NC system is in MPG mode. | R |

| Function Name | Special M | Description | Device |
|---|---|---|---|
| JOG | M3x004 | This special M is ON when the NC system is in JOG mode. | R |
| RAPID | M3x005 | This special M is ON when the NC system is in RAPID mode. | R |
| INC | M3x006 | This special M is ON when the NC system is in INC mode. | R |
| HOME | M3x007 | This special M is ON when the NC system is in HOME mode. | R |
| Cycle Start Status | M3x016 | This special M is ON when the NC system is running NC program. | R |
| Feed Hold Status | M3x017 | This special M is ON when the NC system pause NC process. | R |
| Emergency Stop Status | M3x018 | This special M is ON when the EMG button or signal is triggered. | R |
| Reset Finished | M3x019 | This special M is ON when the NC system finishes reset procedures. | R |
| Break Point Searching | M3x020 | This special M is ON when the NC system is searching for the break point. | R |
| Program End Finished | M3x021 | This special M is ON when the NC system finishes the last block, M02 or M30. | R |
| M02 Executed | M3x022 | This special M is ON when the NC system finishes executing the M02 command and cycle stop. | R |
| M30 Cycle Stop and Index Reset | M3x023 | This special M is ON when the NC system finishes executing the M30 command, cycle stop and program line index back to first line. | R |
| Single Block Hold | M3x024 | This special M is ON when the NC system is holding on a single block. | R |
| NC Error | M3x025 | This special M is ON when the NC system encounters an error. | R |
| Macro Call Status | M3x027 | This special M is ON when the macro call is in execution. | R |
| Macro Call Ready | M3x028 | After users trigger the macro call preparation and then system finishes the preparation, this special M will be ON. When this special M is ON, users will need to switch the NC system to AUTO mode and finish the remaining actions to start the macro. | R |
| Macro Call Error | M3x029 | Indicates a macro call error. | R |
| M96 (Program Interruption) in Execution | M3x031 | This special M is ON when M96 (program interruption) is in execution. | R |
| System Mode Switching | M3x032 | When the system is switching between operation modes such as AUTO or JOG, this special M will be triggered. | R |
| Main Program Lock | M3x033 | When the system locks the current NC main program and it is not allowed to change, this special M will be triggered. | R |
| Servo Drive Error | M3x034 | When any of the connected servo drives encounter an error, this special M will be triggered. | R |
| Axes Auto Servo ON Status | M3x035 | When the system parameter **[N1.10 Axes Manual Servo ON]** is set to 0, the system will set all axes and this special M to servo ON after initialization is finished. | R |
| System Ready and Servo ON | M3x036 | When the system is successful initialized and servo drives are set to servo ON, this special M will be triggered. | R |

5

| Function Name | Special M | Description | Device |
|---|---|---|---|
| M00 System Hold | M3x037 | After the system executes M00 and pauses the procedure, this special M will be triggered. | R |
| M01 Optional Stop | M3x038 | After the system executes M01 and stops the procedure, this special M will be triggered. | R |
| Block Finished on Single Block Mode | M3x039 | When the system is in single block mode and then finishes the current block, this special M will be triggered. | R |
| User Define Alarm Status | M3x040 | This special M is ON when a user defined alarm occurs. | R |
| NC Program Scanning | M3x044 | This special M is ON when the system is scanning the NC program. | R |
| NC Program Scan finished | M3x045 | This special M is ON when the system finishes the NC program scanning. | R |
| M99 Call Stop Status | M3x047 | When the **M2x026** is ON and execute the M99, the NC system will trigger this special M after slowing down the motion and stop the process.<br>This special M will be reset to OFF when the cycle start triggered again, M30 executed, or system reset. | R/W |
| 1st Macro Call Initial Finished | M3x048 | This special M is ON when macro call finishes. | R |
| 2nd Macro Call Initial Finished | M3x049 | This special M is ON when macro call finishes. | R |
| 3rd Macro Call Initial Finished | M3x050 | This special M is ON when macro call finishes. | R |
| 4th Macro Call Initial Finished | M3x051 | This special M is ON when macro call finishes. | R |
| 5th Macro Call Initial Finished | M3x052 | This special M is ON when macro call finishes. | R |
| 6th Macro Call Initial Finished | M3x053 | This special M is ON when macro call finishes. | R |
| 7th Macro Call Initial Finished | M3x054 | This special M is ON when macro call finishes. | R |
| 8th Macro Call Initial Finished | M3x055 | This special M is ON when macro call finishes. | R |
| 9th Macro Call Initial Finished | M3x056 | This special M is ON when macro call finishes. | R |
| 10th Macro Call Initial Finished | M3x057 | This special M is ON when macro call finishes. | R |
| 11th Macro Call Initial Finished | M3x058 | This special M is ON when macro call finishes. | R |
| 12th Macro Call Initial Finished | M3x059 | This special M is ON when macro call finishes. | R |
| 13th Macro Call Initial Finished | M3x060 | This special M is ON when macro call finishes. | R |
| 14th Macro Call Initial Finished | M3x061 | This special M is ON when macro call finishes. | R |
| 15th Macro Call Initial Finished | M3x062 | This special M is ON when macro call finishes. | R |
| 16th Macro Call Initial Finished | M3x063 | This special M is ON when macro call finishes. | R |

| Function Name | Special M | Description | Device |
|---|---|---|---|
| M Code Execution | M3x064 | When the M code is executed in the program (not including M00, M01, M02, M30, M98, M99), this special M will be triggered. When the M, S, and T codes complete their execution and then the MLC triggers the **M2x016**, this special M will be set to OFF. This action does not include an M code that is used for macro calls. | R |
| S Code Execution | M3x065 | When the S code is executed in the program, this special M will be triggered. When the M, S, and T codes complete their execution and then the MLC triggers the **M2x016**, this special M will be set to OFF. The NC does not trigger this special M when an S code is used for macro call. | R |
| T Code Execution | M3x066 | When the T code (Standby tool number) is executed in the program, this special M will be triggered. When the M, S, and T codes complete their execution and then the MLC triggers the **M2x016**, this special M will be set to OFF. The NC does not trigger this special M when a T code is used for macro call. This special M is related to the tool pot setting of the tool magazine, and the special M will be triggered only when the T code value is set within the specified range of tool number for the tool magazine parameter. | R |
| Connection Status of 1st Spindle | M3x096 | This special M is ON when the 1st spindle is connected. | R |
| Connection Status of 2nd Spindle | M3x097 | This special M is ON when the 2nd spindle is connected. | R |
| Connection Status of 3rd Spindle | M3x098 | This special M is ON when the 3rd spindle is connected. | R |
| Connection Status of 4th Spindle | M3x099 | This special M is ON when the 4th spindle is connected. | R |
| Connection Status of 5th Spindle | M3x100 | This special M is ON when the 5th spindle is connected. | R |
| Connection Status of 6th Spindle | M3x101 | This special M is ON when the 6th spindle is connected. | R |
| Connection Status of 7th Spindle | M3x102 | This special M is ON when the 7th spindle is connected. | R |
| Connection Status of 8th Spindle | M3x103 | This special M is ON when the 8th spindle is connected. | R |
| NC Variable to MLC 1 | M3x128 | The NC system will update the status of variable #25256 into this special M as ON or OFF. | R |
| NC Variable to MLC 2 | M3x129 | The NC system will update the status of variable #25257 into this special M as ON or OFF. | R |
| NC Variable to MLC 3 | M3x130 | The NC system will update the status of variable #25258 into this special M as ON or OFF. | R |
| NC Variable to MLC 4 | M3x131 | The NC system will update the status of variable #25259 into this special M as ON or OFF. | R |
| NC Variable to MLC 5 | M3x132 | The NC system will update the status of variable #25260 into this special M as ON or OFF. | R |
| NC Variable to MLC 6 | M3x133 | The NC system will update the status of variable #25261 into this special M as ON or OFF. | R |
| NC Variable to MLC 7 | M3x134 | The NC system will update the status of variable #25262 into this special M as ON or OFF. | R |
| NC Variable to MLC 8 | M3x135 | The NC system will update the status of variable #25263 into this special M as ON or OFF. | R |
| NC Variable to MLC 9 | M3x136 | The NC system will update the status of variable #25264 into this special M as ON or OFF. | R |

| Function Name | Special M | Description | Device |
|---|---|---|---|
| NC Variable to MLC 10 | M3x137 | The NC system will update the status of variable #25265 into this special M as ON or OFF. | R |
| NC Variable to MLC 11 | M3x138 | The NC system will update the status of variable #25266 into this special M as ON or OFF. | R |
| NC Variable to MLC 12 | M3x139 | The NC system will update the status of variable #25267 into this special M as ON or OFF. | R |
| NC Variable to MLC 13 | M3x140 | The NC system will update the status of variable #25268 into this special M as ON or OFF. | R |
| NC Variable to MLC 14 | M3x141 | The NC system will update the status of variable #25269 into this special M as ON or OFF. | R |
| NC Variable to MLC 15 | M3x142 | The NC system will update the status of variable #25270 into this special M as ON or OFF. | R |
| NC Variable to MLC 16 | M3x143 | The NC system will update the status of variable #25271 into this special M as ON or OFF. | R |
| NC Variable to MLC 17 | M3x144 | The NC system will update the status of variable #25272 into this special M as ON or OFF. | R |
| NC Variable to MLC 18 | M3x145 | The NC system will update the status of variable #25273 into this special M as ON or OFF. | R |
| NC Variable to MLC 19 | M3x146 | The NC system will update the status of variable #25274 into this special M as ON or OFF. | R |
| NC Variable to MLC 20 | M3x147 | The NC system will update the status of variable #25275 into this special M as ON or OFF. | R |
| NC Variable to MLC 21 | M3x148 | The NC system will update the status of variable #25276 into this special M as ON or OFF. | R |
| NC Variable to MLC 22 | M3x149 | The NC system will update the status of variable #25277 into this special M as ON or OFF. | R |
| NC Variable to MLC 23 | M3x150 | The NC system will update the status of variable #25278 into this special M as ON or OFF. | R |
| NC Variable to MLC 24 | M3x151 | The NC system will update the status of variable #25279 into this special M as ON or OFF. | R |
| NC Variable to MLC 25 | M3x152 | The NC system will update the status of variable #25280 into this special M as ON or OFF. | R |
| NC Variable to MLC 26 | M3x153 | The NC system will update the status of variable #25281 into this special M as ON or OFF. | R |
| NC Variable to MLC 27 | M3x154 | The NC system will update the status of variable #25282 into this special M as ON or OFF. | R |
| NC Variable to MLC 28 | M3x155 | The NC system will update the status of variable #25283 into this special M as ON or OFF. | R |
| NC Variable to MLC 29 | M3x156 | The NC system will update the status of variable #25284 into this special M as ON or OFF. | R |
| NC Variable to MLC 30 | M3x157 | The NC system will update the status of variable #25285 into this special M as ON or OFF. | R |
| NC Variable to MLC 31 | M3x158 | The NC system will update the status of variable #25286 into this special M as ON or OFF. | R |
| NC Variable to MLC 32 | M3x159 | The NC system will update the status of variable #25287 into this special M as ON or OFF. | R |
| NC Variable to MLC 33 | M3x160 | The NC system will update the status of variable #25288 into this special M as ON or OFF. | R |
| NC Variable to MLC 34 | M3x161 | The NC system will update the status of variable #25289 into this special M as ON or OFF. | R |
| NC Variable to MLC 35 | M3x162 | The NC system will update the status of variable #25290 into this special M as ON or OFF. | R |
| NC Variable to MLC 36 | M3x163 | The NC system will update the status of variable #25291 into this special M as ON or OFF. | R |

| Function Name | Special M | Description | Device |
|---|---|---|---|
| NC Variable to MLC 37 | M3x164 | The NC system will update the status of variable #25292 into this special M as ON or OFF. | R |
| NC Variable to MLC 38 | M3x165 | The NC system will update the status of variable #25293 into this special M as ON or OFF. | R |
| NC Variable to MLC 39 | M3x166 | The NC system will update the status of variable #25294 into this special M as ON or OFF. | R |
| NC Variable to MLC 40 | M3x167 | The NC system will update the status of variable #25295 into this special M as ON or OFF. | R |
| NC Variable to MLC 41 | M3x168 | The NC system will update the status of variable #25296 into this special M as ON or OFF. | R |
| NC Variable to MLC 42 | M3x169 | The NC system will update the status of variable #25297 into this special M as ON or OFF. | R |
| NC Variable to MLC 43 | M3x170 | The NC system will update the status of variable #25298 into this special M as ON or OFF. | R |
| NC Variable to MLC 44 | M3x171 | The NC system will update the status of variable #25299 into this special M as ON or OFF. | R |
| NC Variable to MLC 45 | M3x172 | The NC system will update the status of variable #25300 into this special M as ON or OFF. | R |
| NC Variable to MLC 46 | M3x173 | The NC system will update the status of variable #25301 into this special M as ON or OFF. | R |
| NC Variable to MLC 47 | M3x174 | The NC system will update the status of variable #25302 into this special M as ON or OFF. | R |
| NC Variable to MLC 48 | M3x175 | The NC system will update the status of variable #25303 into this special M as ON or OFF. | R |
| NC Variable to MLC 49 | M3x176 | The NC system will update the status of variable #25304 into this special M as ON or OFF. | R |
| NC Variable to MLC 50 | M3x177 | The NC system will update the status of variable #25305 into this special M as ON or OFF. | R |
| NC Variable to MLC 51 | M3x178 | The NC system will update the status of variable #25306 into this special M as ON or OFF. | R |
| NC Variable to MLC 52 | M3x179 | The NC system will update the status of variable #25307 into this special M as ON or OFF. | R |
| NC Variable to MLC 53 | M3x180 | The NC system will update the status of variable #25308 into this special M as ON or OFF. | R |
| NC Variable to MLC 54 | M3x181 | The NC system will update the status of variable #25309 into this special M as ON or OFF. | R |
| NC Variable to MLC 55 | M3x182 | The NC system will update the status of variable #25310 into this special M as ON or OFF. | R |
| NC Variable to MLC 56 | M3x183 | The NC system will update the status of variable #25311 into this special M as ON or OFF. | R |
| NC Variable to MLC 57 | M3x184 | The NC system will update the status of variable #25312 into this special M as ON or OFF. | R |
| NC Variable to MLC 58 | M3x185 | The NC system will update the status of variable #25313 into this special M as ON or OFF. | R |
| NC Variable to MLC 59 | M3x186 | The NC system will update the status of variable #25314 into this special M as ON or OFF. | R |
| NC Variable to MLC 60 | M3x187 | The NC system will update the status of variable #25315 into this special M as ON or OFF. | R |
| NC Variable to MLC 61 | M3x188 | The NC system will update the status of variable #25316 into this special M as ON or OFF. | R |
| NC Variable to MLC 62 | M3x189 | The NC system will update the status of variable #25317 into this special M as ON or OFF. | R |
| NC Variable to MLC 63 | M3x190 | The NC system will update the status of variable #25318 into this special M as ON or OFF. | R |

| Function Name | Special M | Description | Device |
|---|---|---|---|
| NC Variable to MLC 64 | M3x191 | The NC system will update the status of variable #25319 into this special M as ON or OFF. | R |
| NC Variable to MLC 65 | M3x192 | The NC system will update the status of variable #25320 into this special M as ON or OFF. | R |
| NC Variable to MLC 66 | M3x193 | The NC system will update the status of variable #25321 into this special M as ON or OFF. | R |
| NC Variable to MLC 67 | M3x194 | The NC system will update the status of variable #25322 into this special M as ON or OFF. | R |
| NC Variable to MLC 68 | M3x195 | The NC system will update the status of variable #25323 into this special M as ON or OFF. | R |
| NC Variable to MLC 69 | M3x196 | The NC system will update the status of variable #25324 into this special M as ON or OFF. | R |
| NC Variable to MLC 70 | M3x197 | The NC system will update the status of variable #25325 into this special M as ON or OFF. | R |
| NC Variable to MLC 71 | M3x198 | The NC system will update the status of variable #25326 into this special M as ON or OFF. | R |
| NC Variable to MLC 72 | M3x199 | The NC system will update the status of variable #25327 into this special M as ON or OFF. | R |
| NC Variable to MLC 73 | M3x200 | The NC system will update the status of variable #25328 into this special M as ON or OFF. | R |
| NC Variable to MLC 74 | M3x201 | The NC system will update the status of variable #25329 into this special M as ON or OFF. | R |
| NC Variable to MLC 75 | M3x202 | The NC system will update the status of variable #25330 into this special M as ON or OFF. | R |
| NC Variable to MLC 76 | M3x203 | The NC system will update the status of variable #25331 into this special M as ON or OFF. | R |
| NC Variable to MLC 77 | M3x204 | The NC system will update the status of variable #25332 into this special M as ON or OFF. | R |
| NC Variable to MLC 78 | M3x205 | The NC system will update the status of variable #25333 into this special M as ON or OFF. | R |
| NC Variable to MLC 79 | M3x206 | The NC system will update the status of variable #25334 into this special M as ON or OFF. | R |
| NC Variable to MLC 80 | M3x207 | The NC system will update the status of variable #25335 into this special M as ON or OFF. | R |
| NC Variable to MLC 81 | M3x208 | The NC system will update the status of variable #25336 into this special M as ON or OFF. | R |
| NC Variable to MLC 82 | M3x209 | The NC system will update the status of variable #25337 into this special M as ON or OFF. | R |
| NC Variable to MLC 83 | M3x210 | The NC system will update the status of variable #25338 into this special M as ON or OFF. | R |
| NC Variable to MLC 84 | M3x211 | The NC system will update the status of variable #25339 into this special M as ON or OFF. | R |
| NC Variable to MLC 85 | M3x212 | The NC system will update the status of variable #25340 into this special M as ON or OFF. | R |
| NC Variable to MLC 86 | M3x213 | The NC system will update the status of variable #25341 into this special M as ON or OFF. | R |
| NC Variable to MLC 87 | M3x214 | The NC system will update the status of variable #25342 into this special M as ON or OFF. | R |
| NC Variable to MLC 88 | M3x215 | The NC system will update the status of variable #25343 into this special M as ON or OFF. | R |
| NC Variable to MLC 89 | M3x216 | The NC system will update the status of variable #25344 into this special M as ON or OFF. | R |
| NC Variable to MLC 90 | M3x217 | The NC system will update the status of variable #25345 into this special M as ON or OFF. | R |

| Function Name | Special M | Description | Device |
|---|---|---|---|
| NC Variable to MLC 91 | M3x218 | The NC system will update the status of variable #25346 into this special M as ON or OFF. | R |
| NC Variable to MLC 92 | M3x219 | The NC system will update the status of variable #25347 into this special M as ON or OFF. | R |
| NC Variable to MLC 93 | M3x220 | The NC system will update the status of variable #25348 into this special M as ON or OFF. | R |
| NC Variable to MLC 94 | M3x221 | The NC system will update the status of variable #25349 into this special M as ON or OFF. | R |
| NC Variable to MLC 95 | M3x222 | The NC system will update the status of variable #25350 into this special M as ON or OFF. | R |
| NC Variable to MLC 96 | M3x223 | The NC system will update the status of variable #25351 into this special M as ON or OFF. | R |
| NC Variable to MLC 97 | M3x224 | The NC system will update the status of variable #25352 into this special M as ON or OFF. | R |
| NC Variable to MLC 98 | M3x225 | The NC system will update the status of variable #25353 into this special M as ON or OFF. | R |
| NC Variable to MLC 99 | M3x226 | The NC system will update the status of variable #25354 into this special M as ON or OFF. | R |
| NC Variable to MLC 100 | M3x227 | The NC system will update the status of variable #25355 into this special M as ON or OFF. | R |
| NC Variable to MLC 101 | M3x228 | The NC system will update the status of variable #25356 into this special M as ON or OFF. | R |
| NC Variable to MLC 102 | M3x229 | The NC system will update the status of variable #25357 into this special M as ON or OFF. | R |
| NC Variable to MLC 103 | M3x230 | The NC system will update the status of variable #25358 into this special M as ON or OFF. | R |
| NC Variable to MLC 104 | M3x231 | The NC system will update the status of variable #25359 into this special M as ON or OFF. | R |
| NC Variable to MLC 105 | M3x232 | The NC system will update the status of variable #25360 into this special M as ON or OFF. | R |
| NC Variable to MLC 106 | M3x233 | The NC system will update the status of variable #25361 into this special M as ON or OFF. | R |
| NC Variable to MLC 107 | M3x234 | The NC system will update the status of variable #25362 into this special M as ON or OFF. | R |
| NC Variable to MLC 108 | M3x235 | The NC system will update the status of variable #25363 into this special M as ON or OFF. | R |
| NC Variable to MLC 109 | M3x236 | The NC system will update the status of variable #25364 into this special M as ON or OFF. | R |
| NC Variable to MLC 110 | M3x237 | The NC system will update the status of variable #25365 into this special M as ON or OFF. | R |
| NC Variable to MLC 111 | M3x238 | The NC system will update the status of variable #25366 into this special M as ON or OFF. | R |
| NC Variable to MLC 112 | M3x239 | The NC system will update the status of variable #25367 into this special M as ON or OFF. | R |
| NC Variable to MLC 113 | M3x240 | The NC system will update the status of variable #25368 into this special M as ON or OFF. | R |
| NC Variable to MLC 114 | M3x241 | The NC system will update the status of variable #25369 into this special M as ON or OFF. | R |
| NC Variable to MLC 115 | M3x242 | The NC system will update the status of variable #25370 into this special M as ON or OFF. | R |
| NC Variable to MLC 116 | M3x243 | The NC system will update the status of variable #25371 into this special M as ON or OFF. | R |
| NC Variable to MLC 117 | M3x244 | The NC system will update the status of variable #25372 into this special M as ON or OFF. | R |

5

| Function Name | Special M | Description | Device |
|---|---|---|---|
| NC Variable to MLC 118 | M3x245 | The NC system will update the status of variable #25373 into this special M as ON or OFF. | R |
| NC Variable to MLC 119 | M3x246 | The NC system will update the status of variable #25374 into this special M as ON or OFF. | R |
| NC Variable to MLC 120 | M3x247 | The NC system will update the status of variable #25375 into this special M as ON or OFF. | R |
| NC Variable to MLC 121 | M3x248 | The NC system will update the status of variable #25376 into this special M as ON or OFF. | R |
| NC Variable to MLC 122 | M3x249 | The NC system will update the status of variable #25377 into this special M as ON or OFF. | R |
| NC Variable to MLC 123 | M3x250 | The NC system will update the status of variable #25378 into this special M as ON or OFF. | R |
| NC Variable to MLC 124 | M3x251 | The NC system will update the status of variable #25379 into this special M as ON or OFF. | R |
| NC Variable to MLC 125 | M3x252 | The NC system will update the status of variable #25380 into this special M as ON or OFF. | R |
| NC Variable to MLC 126 | M3x253 | The NC system will update the status of variable #25381 into this special M as ON or OFF. | R |
| NC Variable to MLC 127 | M3x254 | The NC system will update the status of variable #25382 into this special M as ON or OFF. | R |
| NC Variable to MLC 128 | M3x255 | The NC system will update the status of variable #25383 into this special M as ON or OFF. | R |
| Servo Connection Status of X Axis | M3x256 | This special M is ON when the servo drive of the X axis is connected. This connection status can be monitored from the [Connect] status light of the [Servo Monitor Table] page. | R |
| Servo Connection Status of Y Axis | M3x257 | This special M is ON when the servo drive of the Y axis is connected. This connection status can be monitored from the [Connect] status light of the [Servo Monitor Table] page. | R |
| Servo Connection Status of Z Axis | M3x258 | This special M is ON when the servo drive of the Z axis is connected. This connection status can be monitored from the [Connect] status light of the [Servo Monitor Table] page. | R |
| Servo Connection Status of A Axis | M3x259 | This special M is ON when the servo drive of the A axis is connected. This connection status can be monitored from the [Connect] status light of the [Servo Monitor Table] page. | R |
| Servo Connection Status of B Axis | M3x260 | This special M is ON when the servo drive of the B axis is connected. This connection status can be monitored from the [Connect] status light of the [Servo Monitor Table] page. | R |
| Servo Connection Status of C Axis | M3x261 | This special M is ON when the servo drive of the C axis is connected. This connection status can be monitored from the [Connect] status light of the [Servo Monitor Table] page. | R |
| Servo Connection Status of U Axis | M3x262 | This special M is ON when the servo drive of the U axis is connected. This connection status can be monitored from the [Connect] status light of the [Servo Monitor Table] page. | R |
| Servo Connection Status of V Axis | M3x263 | This special M is ON when the servo drive of the V axis is connected. This connection status can be monitored from the [Connect] status light of the [Servo Monitor Table] page. | R |

| Function Name | Special M | Description | Device |
|---|---|---|---|
| Servo Connection Status of W Axis | M3x264 | This special M is ON when the servo drive of the W axis is connected. This connection status can be monitored from the [Connect] status light of the [Servo Monitor Table] page. | R |
| Servo Connection Status of 10th Axis | M3x265 | This special M is ON when the servo drive of the 10th axis is connected. This connection status can be monitored from the [Connect] status light of the [Servo Monitor Table] page. | R |
| Servo Connection Status of 11th Axis | M3x266 | This special M is ON when the servo drive of the 11th axis is connected. This connection status can be monitored from the [Connect] status light of the [Servo Monitor Table] page. | R |
| Servo Connection Status of 12th Axis | M3x267 | This special M is ON when the servo drive of the 12th axis is connected. This connection status can be monitored from the [Connect] status light of the [Servo Monitor Table] page. | R |
| Servo Connection Status of 13th Axis | M3x268 | This special M is ON when the servo drive of the 13th axis is connected. This connection status can be monitored from the [Connect] status light of the [Servo Monitor Table] page. | R |
| Servo Connection Status of 14th Axis | M3x269 | This special M is ON when the servo drive of the 14th axis is connected. This connection status can be monitored from the [Connect] status light of the [Servo Monitor Table] page. | R |
| Servo Connection Status of 15th Axis | M3x270 | This special M is ON when the servo drive of the 15th axis is connected. This connection status can be monitored from the [Connect] status light of the [Servo Monitor Table] page. | R |
| Servo Connection Status of 16th Axis | M3x271 | This special M is ON when the servo drive of the 16th axis is connected. This connection status can be monitored from the [Connect] status light of the [Servo Monitor Table] page. | R |
| Servo Enable Status of X Axis | M3x272 | This special M is ON when the servo drive of the X axis is servo ON. This connection status can be monitored from the [Ready] status light of the [Servo Monitor Table] page. | R |
| Servo Enable Status of Y Axis | M3x273 | This special M is ON when the servo drive of the Y axis is servo ON. This connection status can be monitored from the [Ready] status light of the [Servo Monitor Table] page. | R |
| Servo Enable Status of Z Axis | M3x274 | This special M is ON when the servo drive of the Z axis is servo ON. This connection status can be monitored from the [Ready] status light of the [Servo Monitor Table] page. | R |
| Servo Enable Status of A Axis | M3x275 | This special M is ON when the servo drive of the A axis is servo ON. This connection status can be monitored from the [Ready] status light of the [Servo Monitor Table] page. | R |
| Servo Enable Status of B Axis | M3x276 | This special M is ON when the servo drive of the B axis is servo ON. This connection status can be monitored from the [Ready] status light of the [Servo Monitor Table] page. | R |
| Servo Enable Status of C Axis | M3x277 | This special M is ON when the servo drive of the C axis is servo ON. This connection status can be monitored from the [Ready] status light of the [Servo Monitor Table] page. | R |
| Servo Enable Status of U Axis | M3x278 | This special M is ON when the servo drive of the U axis is servo ON. This connection status can be monitored from the [Ready] status light of the [Servo Monitor Table] page. | R |

| Function Name | Special M | Description | Device |
|---|---|---|---|
| Servo Enable Status of V Axis | M3x279 | This special M is ON when the servo drive of the V axis is servo ON. This connection status can be monitored from the [Ready] status light of the [Servo Monitor Table] page. | R |
| Servo Enable Status of W Axis | M3x280 | This special M is ON when the servo drive of the W axis is servo ON. This connection status can be monitored from the [Ready] status light of the [Servo Monitor Table] page. | R |
| Servo Enable Status of $10^{th}$ Axis | M3x281 | This special M is ON when the servo drive of the $10^{th}$ axis is servo ON. This connection status can be monitored from the [Ready] status light of the [Servo Monitor Table] page. | R |
| Servo Enable Status of $11^{th}$ Axis | M3x282 | This special M is ON when the servo drive of the $11^{th}$ axis is servo ON. This connection status can be monitored from the [Ready] status light of the [Servo Monitor Table] page. | R |
| Servo Enable Status of $12^{th}$ Axis | M3x283 | This special M is ON when the servo drive of the $12^{th}$ axis is servo ON. This connection status can be monitored from the [Ready] status light of the [Servo Monitor Table] page. | R |
| Servo Enable Status of $13^{th}$ Axis | M3x284 | This special M is ON when the servo drive of the $13^{th}$ axis is servo ON. This connection status can be monitored from the [Ready] status light of the [Servo Monitor Table] page. | R |
| Servo Enable Status of $14^{th}$ Axis | M3x285 | This special M is ON when the servo drive of the $14^{th}$ axis is servo ON. This connection status can be monitored from the [Ready] status light of the [Servo Monitor Table] page. | R |
| Servo Enable Status of $15^{th}$ Axis | M3x286 | This special M is ON when the servo drive of the $15^{th}$ axis is servo ON. This connection status can be monitored from the [Ready] status light of the [Servo Monitor Table] page. | R |
| Servo Enable Status of $16^{th}$ Axis | M3x287 | This special M is ON when the servo drive of the $16^{th}$ axis is servo ON. This connection status can be monitored from the [Ready] status light of the [Servo Monitor Table] page. | R |
| Oscillation Control Status of $1^{st}$ Axis | M3x304 | This special M is ON when the $1^{st}$ axis has enabled the oscillation function and set movement command. | R |
| Oscillation Control Status of $2^{nd}$ Axis | M3x305 | This special M is ON when the $2^{nd}$ axis has enabled the oscillation function and set movement command. | R |
| Oscillation Control Status of $3^{rd}$ Axis | M3x306 | This special M is ON when the $3^{rd}$ axis has enabled the oscillation function and set movement command. | R |
| Oscillation Control Status of $4^{th}$ Axis | M3x307 | This special M is ON when the $4^{th}$ axis has enabled the oscillation function and set movement command. | R |
| Oscillation Control Status of $5^{th}$ Axis | M3x308 | This special M is ON when the $5^{th}$ axis has enabled the oscillation function and set movement command. | R |
| Oscillation Control Status of $6^{th}$ Axis | M3x309 | This special M is ON when the $6^{th}$ axis has enabled the oscillation function and set movement command. | R |
| Oscillation Control Status of $7^{th}$ Axis | M3x310 | This special M is ON when the $7^{th}$ axis has enabled the oscillation function and set movement command. | R |
| Oscillation Control Status of $8^{th}$ Axis | M3x311 | This special M is ON when the $8^{th}$ axis has enabled the oscillation function and set movement command. | R |
| Oscillation Control Status of $9^{th}$ Axis | M3x312 | This special M is ON when the $9^{th}$ axis has enabled the oscillation function and set movement command. | R |
| Oscillation Control Status of $10^{th}$ Axis | M3x313 | This special M is ON when the $10^{th}$ axis has enabled the oscillation function and set movement command. | R |

| Function Name | Special M | Description | Device |
|---|---|---|---|
| Oscillation Control Status of 11th Axis | M3x314 | This special M is ON when the 11th axis has enabled the oscillation function and set movement command. | R |
| Oscillation Control Status of 12th Axis | M3x315 | This special M is ON when the 12th axis has enabled the oscillation function and set movement command. | R |
| Oscillation Control Status of 13th Axis | M3x316 | This special M is ON when the 13th axis has enabled the oscillation function and set movement command. | R |
| Oscillation Control Status of 14th Axis | M3x317 | This special M is ON when the 14th axis has enabled the oscillation function and set movement command. | R |
| Oscillation Control Status of 15th Axis | M3x318 | This special M is ON when the 15th axis has enabled the oscillation function and set movement command. | R |
| Oscillation Control Status of 16th Axis | M3x319 | This special M is ON when the 16th axis has enabled the oscillation function and set movement command. | R |
| Axis Homed Status of X Axis | M3x320 | This special M is ON when the X axis is homed, and the controller’s POS page shows the origin complete symbol. | R |
| Axis Homed Status of Y Axis | M3x321 | This special M is ON when the Y axis is homed, and the controller's POS page shows the origin complete symbol. | R |
| Axis Homed Status of Z Axis | M3x322 | This special M is ON when the Z axis is home, and the controller's POS page shows the origin complete symbol. | R |
| Axis Homed Status of A Axis | M3x323 | This special M is ON when the A axis is homed, and the controller's POS page shows the origin complete symbol. | R |
| Axis Homed Status of B Axis | M3x324 | This special M is ON when the B axis is homed, and the controller's POS page shows the origin complete symbol. | R |
| Axis Homed Status of C Axis | M3x325 | This special M is ON when the C axis is homed, and the controller's POS page shows the origin complete symbol. | R |
| Axis Homed Status of U Axis | M3x326 | This special M is ON when the U axis is homed, and the controller's POS page shows the origin complete symbol. | R |
| Axis Homed Status of V Axis | M3x327 | This special M is ON when the V axis is homed, and the controller's POS page shows the origin complete symbol. | R |
| Axis Homed Status of W Axis | M3x328 | This special M is ON when the W axis is homed, and the controller's POS page shows the origin complete symbol. | R |
| Axis Homed Status of 10th Axis | M3x329 | This special M is ON when the 10th axis is homed, and the controller's POS page shows the origin complete symbol. | R |
| Axis Homed Status of 11th Axis | M3x330 | This special M is ON when the 11th axis is homed, and the controller's POS page shows the origin complete symbol. | R |
| Axis Homed Status of 12th Axis | M3x331 | This special M is ON when the 12th axis is homed, and the controller's POS page shows the origin complete symbol. | R |
| Axis Homed Status of 13th Axis | M3x332 | This special M is ON when the 13th axis is homed, and the controller's POS page shows the origin complete symbol. | R |
| Axis Homed Status of 14th Axis | M3x333 | This special M is ON when the 14th axis is homed, and the controller's POS page shows the origin complete symbol. | R |
| Axis Homed Status of 15th Axis | M3x334 | This special M is ON when the 15th axis is homed, and the controller's POS page shows the origin complete symbol. | R |

| Function Name | Special M | Description | Device |
|---|---|---|---|
| Axis Homed Status of 16th Axis | M3x335 | This special M is ON when the 16th axis is homed, and the controller's POS page shows the origin complete symbol. | R |
| Homing Finished Status of X Axis | M3x336 | This special M is ON when the X axis has finished its homing procedure. | R |
| Homing Finished Status of Y Axis | M3x337 | This special M is ON when the Y axis has finished its homing procedure. | R |
| Homing Finished Status of Z Axis | M3x338 | This special M is ON when the Z axis has finished its homing procedure. | R |
| Homing Finished Status of A Axis | M3x339 | This special M is ON when the A axis has finished its homing procedure. | R |
| Homing Finished Status of B Axis | M3x340 | This special M is ON when the B axis has finished its homing procedure. | R |
| Homing Finished Status of C Axis | M3x341 | This special M is ON when the C axis has finished its homing procedure. | R |
| Homing Finished Status of U Axis | M3x342 | This special M is ON when the U axis has finished its homing procedure. | R |
| Homing Finished Status of V Axis | M3x343 | This special M is ON when the V axis has finished its homing procedure. | R |
| Homing Finished Status of W Axis | M3x344 | This special M is ON when the W axis has finished its homing procedure. | R |
| Homing Finished Status of 10th Axis | M3x345 | This special M is ON when the 10th axis has finished its homing procedure. | R |
| Homing Finished Status of 11th Axis | M3x346 | This special M is ON when the 11th axis has finished its homing procedure. | R |
| Homing Finished Status of 12th Axis | M3x347 | This special M is ON when the 12th axis has finished its homing procedure. | R |
| Homing Finished Status of 13th Axis | M3x348 | This special M is ON when the 13th axis has finished its homing procedure. | R |
| Homing Finished Status of 14th Axis | M3x349 | This special M is ON when the 14th axis has finished its homing procedure. | R |
| Homing Finished Status of 15th Axis | M3x350 | This special M is ON when the 15th axis has finished its homing procedure. | R |
| Homing Finished Status of 16th Axis | M3x351 | This special M is ON when the 16th axis has finished its homing procedure. | R |
| X Axis at Origin Position | M3x352 | This special M is ON when the X axis's machine position is 0. | R |
| Y Axis at Origin Position | M3x353 | This special M is ON when the Y axis's machine position is 0. | R |
| Z Axis at Origin Position | M3x354 | This special M is ON when the Z axis's machine position is 0. | R |
| A Axis at Origin Position | M3x355 | This special M is ON when the A axis's machine position is 0. | R |
| B Axis at Origin Position | M3x356 | This special M is ON when the B axis's machine position is 0. | R |
| C Axis at Origin Position | M3x357 | This special M is ON when the C axis's machine position is 0. | R |
| U Axis at Origin Position | M3x358 | This special M is ON when the U axis's machine position is 0. | R |
| V Axis at Origin Position | M3x359 | This special M is ON when the V axis's machine position is 0. | R |
| W Axis at Origin Position | M3x360 | This special M is ON when the W axis's machine position is 0. | R |

| Function Name | Special M | Description | Device |
|---|---|---|---|
| 10th Axis at Origin Position | M3x361 | This special M is ON when the 10th axis's machine position is 0. | R |
| 11th Axis at Origin Position | M3x362 | This special M is ON when the 11th axis's machine position is 0. | R |
| 12th Axis at Origin Position | M3x363 | This special M is ON when the 12th axis's machine position is 0. | R |
| 13th Axis at Origin Position | M3x364 | This special M is ON when the 13th axis's machine position is 0. | R |
| 14th Axis at Origin Position | M3x365 | This special M is ON when the 14th axis's machine position is 0. | R |
| 15th Axis at Origin Position | M3x366 | This special M is ON when the 15th axis's machine position is 0. | R |
| 16th Axis at Origin Position | M3x367 | This special M is ON when the 16th axis's machine position is 0. | R |
| X Axis Switch to MLC Axis Finished | M3x432 | This special M is ON when **M2x432** is triggered, and the X axis has switched to MLC control mode. | R |
| Y Axis Switch to MLC Axis Finished | M3x433 | This special M is ON when **M2x433** is triggered, and the Y axis has switched to MLC control mode. | R |
| Z Axis Switch to MLC Axis Finished | M3x434 | This special M is ON when **M2x434** is triggered, and the Z axis has switched to MLC control mode. | R |
| A Axis Switch to MLC Axis Finished | M3x435 | This special M is ON when **M2x435** is triggered, and the A axis has switched to MLC control mode. | R |
| B Axis Switch to MLC Axis Finished | M3x436 | This special M is ON when **M2x436** is triggered, and the B axis has switched to MLC control mode. | R |
| C Axis Switch to MLC Axis Finished | M3x437 | This special M is ON when **M2x437** is triggered, and the C axis has switched to MLC control mode. | R |
| U Axis Switch to MLC Axis Finished | M3x438 | This special M is ON when **M2x438** is triggered, and the U axis has switched to MLC control mode. | R |
| V Axis Switch to MLC Axis Finished | M3x439 | This special M is ON when **M2x439** is triggered, and the V axis has switched to MLC control mode. | R |
| W Axis Switch to MLC Axis Finished | M3x440 | This special M is ON when **M2x440** is triggered, and the W axis has switched to MLC control mode. | R |
| 10th Axis Switch to MLC Axis Finished | M3x441 | This special M is ON when **M2x441** is triggered, and the 10th axis has switched to MLC control mode. | R |
| 11th Axis Switch to MLC Axis Finished | M3x442 | This special M is ON when **M2x442** is triggered, and the 11th axis has switched to MLC control mode. | R |
| 12th Axis Switch to MLC Axis Finished | M3x443 | This special M is ON when **M2x443** is triggered, and the 12th axis has switched to MLC control mode. | R |
| 13th Axis Switch to MLC Axis Finished | M3x444 | This special M is ON when **M2x444** is triggered, and the 13th axis has switched to MLC control mode. | R |
| 14th Axis Switch to MLC Axis Finished | M3x445 | This special M is ON when **M2x445** is triggered, and the 14th axis has switched to MLC control mode. | R |
| 15th Axis Switch to MLC Axis Finished | M3x446 | This special M is ON when **M2x446** is triggered, and the 15th axis has switched to MLC control mode. | R |
| 16th Axis Switch to MLC Axis Finished | M3x447 | This special M is ON when **M2x447** is triggered, and the 16th axis has switched to MLC control mode. | R |
| X Axis Target Reached (MLC Axis) | M3x448 | This special M is ON when the X axis reaches the target position. If the MLC is in speed mode, then the speed represented by **M3x448** will be written to the special M. | R |
| Y Axis Target Reached (MLC Axis) | M3x449 | This special M is ON when the Y axis reaches the target position. If the MLC is in speed mode, then the speed represented by **M3x449** will be written to the special M. | R |

| Function Name | Special M | Description | Device |
|---|---|---|---|
| Z Axis Target Reached (MLC Axis) | M3x450 | This special M is ON when the Z axis reaches the target position. If the MLC is in speed mode, then the speed represented by **M3x450** will be written to the special M. | R |
| A Axis Target Reached (MLC Axis) | M3x451 | This special M is ON when the A axis reaches the target position. If the MLC is in speed mode, then the speed represented by **M3x451** will be written to the special M. | R |
| B Axis Target Reached (MLC Axis) | M3x452 | This special M is ON when the B axis reaches the target position. If the MLC is in speed mode, then the speed represented by **M3x452** will be written to the special M. | R |
| C Axis Target Reached (MLC Axis) | M3x453 | This special M is ON when the C axis reaches the target position. If the MLC is in speed mode, then the speed represented by **M3x453** will be written to the special M. | R |
| U Axis Target Reached (MLC Axis) | M3x454 | This special M is ON when the U axis reaches the target position. If the MLC is in speed mode, then the speed represented by **M3x454** will be written to the special M. | R |
| V Axis Target Reached (MLC Axis) | M3x455 | This special M is ON when the V axis reaches the target position. If the MLC is in speed mode, then the speed represented by **M3x455** will be written to the special M. | R |
| W Axis Target Reached (MLC Axis) | M3x456 | This special M is ON when the W axis reaches the target position. If the MLC is in speed mode, then the speed represented by **M3x456** will be written to the special M. | R |
| 10$^{th}$ Axis Target Reached (MLC Axis) | M3x457 | This special M is ON when the 10$^{th}$ axis reaches the target position. If the MLC is in speed mode, then the speed represented by **M3x457** will be written to the special M. | R |
| 11$^{th}$ Axis Target Reached (MLC Axis) | M3x458 | This special M is ON when the 11$^{th}$ axis reaches the target position. If the MLC is in speed mode, then the speed represented by **M3x458** will be written to the special M. | R |
| 12$^{th}$ Axis Target Reached (MLC Axis) | M3x459 | This special M is ON when the 12$^{th}$ axis reaches the target position. If the MLC is in speed mode, then the speed represented by **M3x459** will be written to the special M. | R |
| 13$^{th}$ Axis Target Reached (MLC Axis) | M3x460 | This special M is ON when the 13$^{th}$ axis reaches the target position. If the MLC is in speed mode, then the speed represented by **M3x460** will be written to the special M. | R |
| 14$^{th}$ Axis Target Reached (MLC Axis) | M3x461 | This special M is ON when the 14$^{th}$ axis reaches the target position. If the MLC is in speed mode, then the speed represented by **M3x461** will be written to the special M. | R |
| 15$^{th}$ Axis Target Reached (MLC Axis) | M3x462 | This special M is ON when the 15$^{th}$ axis reaches the target position. If the MLC is in speed mode, then the speed represented by **M3x462** will be written to the special M. | R |
| 16$^{th}$ Axis Target Reached (MLC Axis) | M3x463 | This special M is ON when the 16$^{th}$ axis reaches the target position. If the MLC is in speed mode, then the speed represented by **M3x463** will be written to the special M. | R |
| X Axis Is Moving | M3x464 | This special M is ON when the X axis is in motion, regardless of the type of mode. | R |

| Function Name | Special M | Description | Device |
|---|---|---|---|
| Y Axis Is Moving | M3x465 | This special M is ON when the Y axis is in motion, regardless of the type of mode. | R |
| Z Axis Is Moving | M3x466 | This special M is ON when the Z axis is in motion, regardless of the type of mode. | R |
| A Axis Is Moving | M3x467 | This special M is ON when the A axis is in motion, regardless of the type of mode. | R |
| B Axis Is Moving | M3x468 | This special M is ON when the B axis is in motion, regardless of the type of mode. | R |
| C Axis Is Moving | M3x469 | This special M is ON when the C axis is in motion, regardless of the type of mode. | R |
| U Axis Is Moving | M3x470 | This special M is ON when the U axis is in motion, regardless of the type of mode. | R |
| V Axis Is Moving | M3x471 | This special M is ON when the V axis is in motion, regardless of the type of mode. | R |
| W Axis Is Moving | M3x472 | This special M is ON when the W axis is in motion, regardless of the type of mode. | R |
| $10^{th}$ Axis Is Moving | M3x473 | This special M is ON when the $10^{th}$ axis is in motion, regardless of the type of mode. | R |
| $11^{th}$ Axis Is Moving | M3x474 | This special M is ON when the $11^{th}$ axis is in motion, regardless of the type of mode. | R |
| $12^{th}$ Axis Is Moving | M3x475 | This special M is ON when the $12^{th}$ axis is in motion, regardless of the type of mode. | R |
| $13^{th}$ Axis Is Moving | M3x476 | This special M is ON when the $13^{th}$ axis is in motion, regardless of the type of mode. | R |
| $14^{th}$ Axis Is Moving | M3x477 | This special M is ON when the $14^{th}$ axis is in motion, regardless of the type of mode. | R |
| $15^{th}$ Axis Is Moving | M3x478 | This special M is ON when the $15^{th}$ axis is in motion, regardless of the type of mode. | R |
| $16^{th}$ Axis Is Moving | M3x479 | This special M is ON when the $16^{th}$ axis is in motion, regardless of the type of mode. | R |
| X Axis Moving Forward | M3x480 | This special M is ON when the X axis is moving in the positive direction. | R |
| Y Axis Moving Forward | M3x481 | This special M is ON when the Y axis is moving in the positive direction. | R |
| Z Axis Moving Forward | M3x482 | This special M is ON when the Z axis is moving in the positive direction. | R |
| A Axis Moving Forward | M3x483 | This special M is ON when the A axis is moving in the positive direction. | R |
| B Axis Moving Forward | M3x484 | This special M is ON when the B axis is moving in the positive direction. | R |
| C Axis Moving Forward | M3x485 | This special M is ON when the C axis is moving in the positive direction. | R |
| U Axis Moving Forward | M3x486 | This special M is ON when the U axis is moving in the positive direction. | R |
| V Axis Moving Forward | M3x487 | This special M is ON when the V axis is moving in the positive direction. | R |
| W Axis Moving Forward | M3x488 | This special M is ON when the W axis is moving in the positive direction. | R |
| $10^{th}$ Axis Moving Forward | M3x489 | This special M is ON when the $10^{th}$ axis is moving in the positive direction. | R |
| $11^{th}$ Axis Moving Forward | M3x490 | This special M is ON when the $11^{th}$ axis is moving in the positive direction. | R |
| $12^{th}$ Axis Moving Forward | M3x491 | This special M is ON when the $12^{th}$ axis is moving in the positive direction. | R |

5

| Function Name | Special M | Description | Device |
|---|---|---|---|
| 13$^{th}$ Axis Moving Forward | M3x492 | This special M is ON when the 13$^{th}$ axis is moving in the positive direction. | R |
| 14$^{th}$ Axis Moving Forward | M3x493 | This special M is ON when the 14$^{th}$ axis is moving in the positive direction. | R |
| 15$^{th}$ Axis Moving Forward | M3x494 | This special M is ON when the 15$^{th}$ axis is moving in the positive direction. | R |
| 16$^{th}$ Axis Moving Forward | M3x495 | This special M is ON when the 16$^{th}$ axis is moving in the positive direction. | R |
| X Axis Moving Backward | M3x496 | This special M is ON when the X axis is moving in the negative direction. | R |
| Y Axis Moving Backward | M3x497 | This special M is ON when the Y axis is moving in the negative direction. | R |
| Z Axis Moving Backward | M3x498 | This special M is ON when the Z axis is moving in the negative direction. | R |
| A Axis Moving Backward | M3x499 | This special M is ON when the A axis is moving in the negative direction. | R |
| B Axis Moving Backward | M3x500 | This special M is ON when the B axis is moving in the negative direction. | R |
| C Axis Moving Backward | M3x501 | This special M is ON when the C axis is moving in the negative direction. | R |
| U Axis Moving Backward | M3x502 | This special M is ON when the U axis is moving in the negative direction. | R |
| V Axis Moving Backward | M3x503 | This special M is ON when the V axis is moving in the negative direction. | R |
| W Axis Moving Backward | M3x504 | This special M is ON when the W axis is moving in the negative direction. | R |
| 10$^{th}$ Axis Moving Backward | M3x505 | This special M is ON when the 10$^{th}$ axis is moving in the negative direction. | R |
| 11$^{th}$ Axis Moving Backward | M3x506 | This special M is ON when the 11$^{th}$ axis is moving in the negative direction. | R |
| 12$^{th}$ Axis Moving Backward | M3x507 | This special M is ON when the 12$^{th}$ axis is moving in the negative direction. | R |
| 13$^{th}$ Axis Moving Backward | M3x508 | This special M is ON when the 13$^{th}$ axis is moving in the negative direction. | R |
| 14$^{th}$ Axis Moving Backward | M3x509 | This special M is ON when the 14$^{th}$ axis is moving in the negative direction. | R |
| 15$^{th}$ Axis Moving Backward | M3x510 | This special M is ON when the 15$^{th}$ axis is moving in the negative direction. | R |
| 16$^{th}$ Axis Moving Backward | M3x511 | This special M is ON when the 16$^{th}$ axis is moving in the negative direction. | R |
| Diameter or Radius Mode of X Axis | M3x512 | In the lathe system, this special M is ON when the X axis is in the diameter mode or OFF when it is in the radius mode. | R |
| Diameter or Radius Mode of Y Axis | M3x513 | In the lathe system, this special M is ON when the Y axis is in the diameter mode or OFF when it is in the radius mode. | R |
| Diameter or Radius Mode of Z Axis | M3x514 | In the lathe system, this special M is ON when the Z axis is in the diameter mode or OFF when it is in the radius mode. | R |
| Diameter or Radius Mode of A Axis | M3x515 | In the lathe system, this special M is ON when the A axis is in the diameter mode or OFF when it is in the radius mode. | R |
| Diameter or Radius Mode of B Axis | M3x516 | In the lathe system, this special M is ON when the B axis is in the diameter mode or OFF when it is in the radius mode. | R |

| Function Name | Special M | Description | Device |
|---|---|---|---|
| Diameter or Radius Mode of C Axis | M3x517 | In the lathe system, this special M is ON when the C axis is in the diameter mode or OFF when it is in the radius mode. | R |
| Diameter or Radius Mode of U Axis | M3x518 | In the lathe system, this special M is ON when the U axis is in the diameter mode or OFF when it is in the radius mode. | R |
| Diameter or Radius Mode of V Axis | M3x519 | In the lathe system, this special M is ON when the V axis is in the diameter mode or OFF when it is in the radius mode. | R |
| Diameter or Radius Mode of W Axis | M3x520 | In the lathe system, this special M is ON when the W axis is in the diameter mode or OFF when it is in the radius mode. | R |
| Diameter or Radius Mode of 10th Axis | M3x521 | In the lathe system, this special M is ON when the 10th axis is in the diameter mode or OFF when it is in the radius mode. | R |
| Diameter or Radius Mode of 11th Axis | M3x522 | In the lathe system, this special M is ON when the 11th axis is in the diameter mode or OFF when it is in the radius mode. | R |
| Diameter or Radius Mode of 12th Axis | M3x523 | In the lathe system, this special M is ON when the 12th axis is in the diameter mode or OFF when it is in the radius mode. | R |
| Diameter or Radius Mode of 13th Axis | M3x524 | In the lathe system, this special M is ON when the 13th axis is in the diameter mode or OFF when it is in the radius mode. | R |
| Diameter or Radius Mode of 14th Axis | M3x525 | In the lathe system, this special M is ON when the 14th axis is in the diameter mode or OFF when it is in the radius mode. | R |
| Diameter or Radius Mode of 15th Axis | M3x526 | In the lathe system, this special M is ON when the 15th axis is in the diameter mode or OFF when it is in the radius mode. | R |
| Diameter or Radius Mode of 16th Axis | M3x527 | In the lathe system, this special M is ON when the 16th axis is in the diameter mode or OFF when it is in the radius mode. | R |
| 1st Spindle Speed Reach | M3x704 | This special M is ON when the 1st spindle's speed reaches the target value. | R |
| 1st Spindle Zero Speed | M3x705 | This special M is ON when the 1st spindle's speed reaches zero. | R |
| 1st Spindle Target Reach | M3x706 | This special M is ON when the 1st spindle reaches the target position. | R |
| 1st Spindle Is in The Rigid Tapping | M3x707 | This special M is ON when the 1st spindle is executing the rigid tapping. | R |
| 1st Spindle Is in Position Axis Mode | M3x709 | This special M is ON when the 1st spindle is switching from S axis to C axis. | R |
| 1st Spindle Ready | M3x710 | This special M is ON when the 1st spindle is ready to use. | R |
| 2nd Spindle Speed Reach | M3x720 | This special M is ON when the 2nd spindle's speed reaches the target value. | R |
| 2nd Spindle Zero Speed | M3x721 | This special M is ON when the 2nd spindle's speed reaches zero. | R |
| 2nd Spindle Target Reach | M3x722 | This special M is ON when the 2nd spindle reaches the target position. | R |
| 2nd Spindle Is in The Rigid Tapping | M3x723 | This special M is ON when the 2nd spindle is executing the rigid tapping. | R |
| 2nd Spindle Is in Position Axis Mode | M3x725 | This special M is ON when the 2nd spindle is switching from S axis to C axis. | R |

| Function Name | Special M | Description | Device |
|---|---|---|---|
| 2nd Spindle Ready | M3x726 | This special M is ON when the 2nd spindle is ready to use. | R |
| 3rd Spindle Speed Reach | M3x736 | This special M is ON when the 3rd spindle's speed reaches the target value. | R |
| 3rd Spindle Zero Speed | M3x737 | This special M is ON when the 3rd spindle's speed reaches zero. | R |
| 3rd Spindle Target Reach | M3x738 | This special M is ON when the 3rd spindle reaches the target position. | R |
| 3rd Spindle Is in The Rigid Tapping | M3x739 | This special M is ON when the 3rd spindle is executing the rigid tapping. | R |
| 3rd Spindle Is in Position Axis Mode | M3x741 | This special M is ON when the 3rd spindle is switching from S axis to C axis. | R |
| 3rd Spindle Ready | M3x742 | This special M is ON when the 3rd spindle is ready to use. | R |
| 4th Spindle Speed Reach | M3x752 | This special M is ON when the 4th spindle's speed reaches the target value. | R |
| 4th Spindle Zero Speed | M3x753 | This special M is ON when the 4th spindle's speed reaches zero. | R |
| 4th Spindle Target Reach | M3x754 | This special M is ON when the 4th spindle reaches the target position. | R |
| 4th Spindle Is in The Rigid Tapping | M3x755 | This special M is ON when the 4th spindle is executing the rigid tapping. | R |
| 4th Spindle Is in Position Axis Mode | M3x757 | This special M is ON when the 4th spindle is switching from S axis to C axis. | R |
| 4th Spindle Ready | M3x758 | This special M is ON when the 4th spindle is ready to use. | R |
| 5th Spindle Speed Reach | M3x768 | This special M is ON when the 5th spindle's speed reaches the target value. | R |
| 5th Spindle Zero Speed | M3x769 | This special M is ON when the 5th spindle's speed reaches zero. | R |
| 5th Spindle Positioning Control | M3x770 | This special M is ON when the 5th spindle reaches the target position. | R |
| 5th Spindle Is in The Rigid Tapping | M3x771 | This special M is ON when the 5th spindle is executing the rigid tapping. | R |
| Switching C / S Axis of 5th Lathe Spindle | M3x773 | This special M is ON when the 5th spindle is switching from S axis to C axis. | R |
| 5th Spindle Ready | M3x774 | This special M is ON when the 5th spindle is ready to use. | R |
| 6th Spindle Speed Reach | M3x784 | This special M is ON when the 6th spindle's speed reaches the target value. | R |
| 6th Spindle Zero Speed | M3x785 | This special M is ON when the 6th spindle's speed reaches zero. | R |
| 6th Spindle Target Reach | M3x786 | This special M is ON when the 6th spindle reaches the target position. | R |
| 6th Spindle Is in The Rigid Tapping | M3x787 | This special M is ON when the 6th spindle is executing the rigid tapping. | R |
| 6th Spindle Is In Position Axis Mode | M3x789 | This special M is ON when the 6th spindle is switching from S axis to C axis. | R |
| 6th Spindle Ready | M3x790 | This special M is ON when the 6th spindle is ready to use. | R |
| 7th Spindle Speed Reach | M3x800 | This special M is ON when the 7th spindle's speed reaches the target value. | R |
| 7th Spindle Zero Speed | M3x801 | This special M is ON when the 7th spindle's speed reaches zero. | R |

| Function Name | Special M | Description | Device |
|---|---|---|---|
| 7th Spindle Target Reach | M3x802 | This special M is ON when the 7th spindle reaches the target position. | R |
| 7th Spindle Is in The Rigid Tapping | M3x803 | This special M is ON when the 7th spindle is executing the rigid tapping. | R |
| 7th Spindle Is in Position Axis Mode | M3x805 | This special M is ON when the 7th spindle is switching from S axis to C axis. | R |
| 7th Spindle Ready | M3x806 | This special M is ON when the 7th spindle is ready to use. | R |
| 8th Spindle Speed Reach | M3x816 | This special M is ON when the 8th spindle's speed reaches the target value. | R |
| 8th Spindle Zero Speed | M3x817 | This special M is ON when the 8th spindle's speed reaches zero. | R |
| 8th Spindle Target Reach | M3x818 | This special M is ON when the 8th spindle reaches the target position. | R |
| 8th Spindle Is in The Rigid Tapping | M3x819 | This special M is ON when the 8th spindle is executing the rigid tapping. | R |
| 8th Spindle Is in Position Axis Mode | M3x821 | This special M is ON when the 8th spindle is switching from S axis to C axis. | R |
| 8th Spindle Ready | M3x822 | This special M is ON when the 8th spindle is ready to use. | R |

5

## 5.2.3 List of special D (MLC to system)

### 5.2.3.1 D20000 to D28999

| Function Name | Special D | Description | Device | Type | Range |
|---|---|---|---|---|---|
| NC Channel Selection | D20000 | Setting the current NC channel for the controller and screen interface. When the value is 1, it means the system is using and showing NC channel 1; when the value is 2, it means the system is using and showing NC channel 2, and so forth. | R/W | Decimal | 0 ~ 65,535 |
| Spindle Analog Voltage Output Port 1 | D20160 | When the spindle is in EtherCAT mode, users can use this special D register to output the analog voltage.<br>Unit: 0.01V. | R/W | Decimal | -1,000 ~ +1,000 |
| Spindle Analog Voltage Output Port 2 | D20161 | | R/W | Decimal | -1,000 ~ +1,000 |
| 1st Remote Module Output | D20200<br>D20201<br>D20202<br>D20203 | For connected modules that do not include digital input/output (DI/O), the system will set the arrangement order according to the EtherCAT module and will set the special D module with the special D based on the connection order (such as analog modules). After the remote modules get these data, it will output the results as transferred data.<br>Taking the first module as an example, D20200 corresponds to the first set of values on the module, D20201 corresponds to the second set of values on the module, and so forth. | R/W | Decimal | 0 ~ 65,535 |
| 2nd Remote Module Output | D20204<br>D20205<br>D20206<br>D20207 | | R/W | Decimal | 0 ~ 65,535 |
| 3rd Remote Module Output | D20208<br>D20209<br>D20210<br>D20211 | | R/W | Decimal | 0 ~ 65,535 |
| 4th Remote Module Output | D20212<br>D20213<br>D20214<br>D20215 | | R/W | Decimal | 0 ~ 65,535 |
| 5th Remote Module Output | D20216<br>D20217<br>D20218<br>D20219 | | R/W | Decimal | 0 ~ 65,535 |
| 6th Remote Module Output | D20220<br>D20221<br>D20222<br>D20223 | | R/W | Decimal | 0 ~ 65,535 |
| 7th Remote Module Output | D20224<br>D20225<br>D20226<br>D20227 | | R/W | Decimal | 0 ~ 65,535 |
| 8th Remote Module Output | D20228<br>D20229<br>D20230<br>D20231 | | R/W | Decimal | 0 ~ 65,535 |

<table>
<tr><th>Function Name</th><th>Special D</th><th>Description</th><th>Device</th><th>Type</th><th>Range</th></tr>
<tr><td>NC Mode Switching</td><td>D2x000</td><td>This special D is for NC channels to switch to different operation modes.<br>0: AUTO<br>1: EDIT<br>2: MDI<br>3: MPG<br>4: JOG<br>5: RAPID<br>6: INC<br>7: HOME</td><td>R/W</td><td>Decimal</td><td>0 ~ 7</td></tr>
<tr><td>Feed Rate Percentage</td><td>D2x002</td><td>Setting for feed rate percentage of NC program speed.<br>Unit: %<br>Ex: When the NC program speed is 1000 mm/min and this D2x002 is 50, this means the NC system will execute the axes interpolation speed as 1000 x 50% = 500 mm/min.</td><td>R/W</td><td>Decimal</td><td>0 ~ 150</td></tr>
<tr><td>Rapid Speed Percentage</td><td>D2x004</td><td>Setting NC rapid speed percentage for G00 command.<br>Unit: %<br>Ex: When the NC rapid speed [N1.030] is 6000 mm/min and this D2x004 is 50, this means the NC system will execute the axes rapid speed as 6000 x 50% = 3000 mm/min.</td><td>R/W</td><td>Decimal</td><td>0 ~ 65,535</td></tr>
<tr><td>Speed Override for JOG and INC</td><td>D2x006</td><td>Setting JOG and INC speed.<br>When the system is in JOG or INC mode, it will take two different mode types to determine the moving speed. If the JOG speed mode [N1.011 Bit26] is set as 0, the system will take the JOG maximum speed [N2.030] and then multiply this the value in this special D (unit: %) as the JOG moving speed.<br>If the JOG speed mode [N1.011 Bit26] is set as 1, the system will take the value of this special D as the JOG moving speed directly. The unit of this special D will refer to parameter axes control type [N2.001].
<table>
<tr><th colspan="2">N2.001</th><th>Unit</th></tr>
<tr><td colspan="2">Bit 2~4 (Linear axis)</td><td>mm/min</td></tr>
<tr><td rowspan="2">Bit 2~4 (Rotary axis)</td><td>Bit 11 = 0</td><td>Deg/min</td></tr>
<tr><td>Bit 11 = 1</td><td>RPM</td></tr>
</table></td><td>R/W</td><td>Decimal</td><td>Percentage Mode 0 ~ 100<br><br>Constant Mode 0 ~ 65,535</td></tr>
</table>

5

| Function Name | Special D | Description | Device | Type | Range |
|---|---|---|---|---|---|
| 1st MPG Axes Selection | D2x008 | This special D is for the NC system to switch the 1st MPG axis in MPG mode.<br>0: X axis<br>1: Y axis<br>2: Z axis<br>3: A axis<br>…<br>15: 16th axis | R/W | Decimal | 0 ~ 15 |
| 1st MPG Ratio Selection | D2x009 | This special D is for the NC system to switch the 1st MPG axis moving ratio, which can be 1, 10 or 100. The system will take the unit setting **[N9.013]** as the smallest movement and then multiply it by this ratio to derive the final movement on each scale of the MPG hand wheel.<br>Ex: When D2x009 is 1 and the unit setting [N9.013] is 0.001, with 3 decimal places, the minimum movement of the MPG hand wheel will be 0.001 × 1 = 0.001 mm. | R/W | Decimal | 0 ~ 65,535 |
| 2nd MPG Axes Selection | D2x010 | This special D is for the NC system to switch the 2nd MPG axis in MPG mode.<br>0: X axis<br>1: Y axis<br>2: Z axis<br>3: A axis<br>...<br>15: 16th axis<br>*Set M20024 **[Enable 3 MPG Control]** to ON to enable this function. | R/W | Decimal | 0 ~ 15 |
| 2nd MPG Ratio Selection | D2x011 | This special D is for the NC system to switch the 2nd MPG axis moving ratio, which can be 1, 10 or 100. The system will take the unit setting **[N9.013]** as the smallest movement and then multiply it by this ratio to derive the final movement on each scale of the MPG hand wheel.<br>Ex: When D2x009 is 1 and the unit setting **[N9.013]** is 0.001, with 3 decimal places, the minimum movement of the MPG hand wheel will be 0.001 × 1 = 0.001 mm.<br>*Set M20024 **[Enable 3 MPG Control]** to ON to enable this function. | R/W | Decimal | 0 ~ 65,535 |

| Function Name | Special D | Description | Device | Type | Range |
|---|---|---|---|---|---|
| 3rd MPG Axes Selection | D2x012 | This special D is for the NC system to switch the MPG axis in the 3rd MPG mode.<br>0: X axis<br>1: Y axis<br>2: Z axis<br>3: A axis<br>...<br>15: 16th axis<br>*Set M20024 **[Enable 3 MPG Control]** to ON to enable this function. | R/W | Decimal | 0 ~ 15 |
| 3rd MPG Ratio Selection | D2x013 | This special D is for the NC system to switch the 3rd MPG axis moving ratio, which can be 1, 10 or 100. The system will take the unit setting **[N9.013]** as the smallest movement and then multiply it by this ratio to derive the final movement on each scale of the MPG hand wheel.<br>Ex: When D2x009 is 1 and the unit setting **[N9.013]** is 0.001, with 3 decimal places, the minimum movement of the MPG hand wheel will be 0.001 × 1 = 0.001 mm.<br>*Set M20024 **[Enable 3 MPG Control]** to ON to enable this function. | R/W | Decimal | 0 ~ 65,535 |
| Axes Movement in INC Mode | D2x014 | When the NC system is in INC mode and the JOG motion trigger special M (M2x384 ~ M2x415) is enabled, the system will take this special D and then multiply the unit setting **[N9.013]** to derive the target movement.<br>Ex: When D2x014 is 1234 and unit setting [N9.013] is 0.001, with 3 decimal places, the axes will move 1.234 mm every time the special M of JOG motion trigger enabled. | R/W | Decimal | 0 ~ 4,294,967,295 |
| Robot's Coordinate System Switch in Manual Mode | D2x016 | Setting the Robot's coordinate system when the system is in manual mode.<br>0: Not using coordinate manual mode<br>1 ~ 6: corresponding to G54 ~ G59. | R/W | Decimal | 0 ~ 6 |
| Robot's Tool System Switch in Manual Mode | D2x017 | Setting the Robot's tool offset system when the system is in manual mode.<br>0: Not using tool offset<br>1 ~ N: corresponds to tool number | R/W | Decimal | 0 ~ N<br>N will be based on N0.408 ~ N0.411 |
| Axes Oscillation Amplitude | D2x018<br>D2x019 | When the axes oscillation control **[M2x264]** is enabled, the activated axes will oscillate based on the amplitude here.<br>Unit: mm | R/W | Float | -2,147,483,648 ~ +2,147,483,647 |

5

| Function Name | Special D | Description | Device | Type | Range |
|---|---|---|---|---|---|
| Axes Oscillation Cycle Time | D2x020<br>D2x021 | When the axes oscillation control **[M2x264]** is enabled, the system will take this special D as the cycle time for each oscillation command.<br>Unit: ms | R/W | Float | 2,147,483,648<br>~<br>+2,147,483,647 |
| Oscillation Axes Enable | D2x022 | The oscillation axes use this 16-bit special D as the mask to determine whether to enable the function on each specific axis.<br>For example, if this set to 5, it means the X and Z axes will both have their oscillation function activated. | R/W | Decimal | 0 ~ 65,535 |
| Axes Oscillation Wave Type | D2x023 | Setting for axes oscillation wave type.<br>0: Start position as middle position of the SIN wave.<br>1: Start position as base position (max or min) of the SIN wave. | R/W | Decimal | 0 ~ 1 |
| 1st Spindle Speed | D2x024<br>D2x025 | Write the 1st spindle's speed through the special D (in accordance with **M2x710**). | R/W | Decimal | 0 ~<br>4,294,967,295 |
| 1st Spindle Speed Rate | D2x026 | Setting the spindle's speed ratio.<br>Ex: When the program speed is S1000 and this special D is 30, then the NC system will execute the spindle speed as 1000 x 30% = 300 RPM. | R/W | Decimal | 0 ~ 65,535 |
| 1st Spindle Gear Ratio Selection | D2x027 | Select in accordance with **N0.1034 ~ N0.1041**. Ex: When this special D is set to 1, the system will set the 1st spindle to 1st gear, when it is set to 2, it will be in 2nd gear, and so forth. | R/W | Decimal | 0 ~ 65,535 |
| 2nd Spindle Speed | D2x030<br>D2x031 | Write the 2nd spindle's speed through the special D (in accordance with **M2x726**). | R/W | Decimal | 0 ~<br>4,294,967,295 |
| 2nd Spindle Speed Rate | D2x032 | Setting the spindle's speed ratio.<br>Ex: When the program speed is S1000 and this special D is 30, then the NC system will execute the spindle speed as 1000 x 30% = 300 RPM. | R/W | Decimal | 0 ~ 65,535 |
| 2nd Spindle Gear Ratio Selection | D2x033 | Select in accordance with **N0.1084 ~ N0.1091**. Ex: When this special D is set to 1, the system will set the 2nd spindle to 1st gear, when it is set to 2, it will be in 2nd gear, and so forth. | R/W | Decimal | 0 ~ 65,535 |
| Tool Number<br>Tool Magazine 1 | D2x036 | Write the tool number through the special D<br>(in accordance with **N1.010**). | R/W | Decimal | 0 ~ 65,535 |
| Standby Tool Number<br>Tool Magazine 1 | D2x037 | Write the standby tool number through the special D<br>(in accordance with **N1.010**). | R/W | Decimal | 0 ~ 65,535 |

| Function Name | Special D | Description | Device | Type | Range |
|---|---|---|---|---|---|
| Command Tool Number<br>Tool Magazine 1 | D2x038 | Write the command tool number through the special D<br>(in accordance with **N1.010**). | R/W | Decimal | 0 ~ 65,535 |
| Tool Number<br>Tool Magazine 2 | D2x042 | Write the tool number through the special D<br>(in accordance with **N1.010**). | R/W | Decimal | 0 ~ 65,535 |
| Standby Tool Number<br>Tool Magazine 2 | D2x043 | Write the standby tool number through the special D<br>(in accordance with **N1.010**). | R/W | Decimal | 0 ~ 65,535 |
| Command Tool Number<br>Tool Magazine 2 | D2x044 | Write the command tool number through the special D<br>(in accordance with **N1.010**). | R/W | Decimal | 0 ~ 65,535 |
| 1st Macro Call Macro Number | D2x064 | Specify the macro call function's O macro file number, such as O9xxx.<br>Ex: When this special D is written to K9100 and the 1st macro call is triggered with special **M2x032**, the NC system will execute the O9100 program. | R/W | Decimal | 1 ~ 65,535 |
| 2nd Macro Call Macro Number | D2x065 | Specify the macro call function's O macro file number, such as O9xxx.<br>Ex: When this special D is written to K9100 and the 2nd macro call is triggered with special **M2x033**, the NC system will execute the O9100 program. | R/W | Decimal | 1 ~ 65,535 |
| 3rd Macro Call Macro Number | D2x066 | Specify the macro call function's O macro file number, such as O9xxx.<br>Ex: When this special D is written to K9100 and the 3rd macro call is triggered with special **M2x034**, the NC system will execute the O9100 program. | R/W | Decimal | 1 ~ 65,535 |
| 4th Macro Call Macro Number | D2x067 | Specify the macro call function's O macro file number, such as O9xxx.<br>Ex: When this special D is written to K9100 and the 4th macro call is triggered with special **M2x035**, the NC system will execute the O9100 program. | R/W | Decimal | 1 ~ 65,535 |
| 5th Macro Call Macro Number | D2x068 | Specify the macro call function's O macro file number, such as O9xxx.<br>Ex: When this special D is written to K9100 and the 5th macro call is triggered with special **M2x036**, the NC system will execute the O9100 program. | R/W | Decimal | 1 ~ 65,535 |
| 6th Macro Call Macro Number | D2x069 | Specify the macro call function's O macro file number, such as O9xxx.<br>Ex: When this special D is written to K9100 and the 6th macro call is triggered with special **M2x037**, the NC system will execute the O9100 program. | R/W | Decimal | 1 ~ 65,535 |

5

| Function Name | Special D | Description | Device | Type | Range |
|---|---|---|---|---|---|
| 7th Macro Call Macro Number | D2x070 | Specify the macro call function's O macro file number, such as O9xxx. Ex: When this special D is written to K9100 and the 7th macro call is triggered with special **M2x038**, the NC system will execute the O9100 program. | R/W | Decimal | 1 ~ 65,535 |
| 8th Macro Call Macro Number | D2x071 | Specify the macro call function's O macro file number, such as O9xxx. Ex: When this special D is written to K9100 and the 8th macro call is triggered with special **M2x039**, the NC system will execute the O9100 program. | R/W | Decimal | 1 ~ 65,535 |
| 9th Macro Call Macro Number | D2x072 | Specify the macro call function's O macro file number, such as O9xxx. Ex: When this special D is written to K9100 and the 9th macro call is triggered with special **M2x040**, the NC system will execute the O9100 program. | R/W | Decimal | 1 ~ 65,535 |
| 10th Macro Call Macro Number | D2x073 | Specify the macro call function's O macro file number, such as O9xxx. Ex: When this special D is written to K9100 and the 10th macro call is triggered with special **M2x041**, the NC system will execute the O9100 program. | R/W | Decimal | 1 ~ 65,535 |
| 11th Macro Call Macro Number | D2x074 | Specify the macro call function's O macro file number, such as O9xxx. Ex: When this special D is written to K9100 and the 11th macro call is triggered with special **M2x042**, the NC system will execute the O9100 program. | R/W | Decimal | 1 ~ 65,535 |
| 12th Macro Call Macro Number | D2x075 | Specify the macro call function's O macro file number, such as O9xxx. Ex: When this special D is written to K9100 and the 12th macro call is triggered with special **M2x043**, the NC system will execute the O9100 program. | R/W | Decimal | 1 ~ 65,535 |
| 13th Macro Call Macro Number | D2x076 | Specify the macro call function's O macro file number, such as O9xxx. Ex: When this special D is written to K9100 and the 13th macro call is triggered with special **M2x044**, the NC system will execute the O9100 program. | R/W | Decimal | 1 ~ 65,535 |
| 14th Macro Call Macro Number | D2x077 | Specify the macro call function's O macro file number, such as O9xxx. Ex: When this special D is written to K9100 and the 14th macro call is triggered with special **M2x045**, the NC system will execute the O9100 program. | R/W | Decimal | 1 ~ 65,535 |

| Function Name | Special D | Description | Device | Type | Range |
|---|---|---|---|---|---|
| 15th Macro Call Macro Number | D2x078 | Specify the macro call function's O macro file number, such as O9xxx.<br>Ex: When this special D is written to K9100 and the 15th macro call is triggered with special **M2x046**, the NC system will execute the O9100 program. | R/W | Decimal | 1 ~ 65,535 |
| 16th Macro Call Macro Number | D2x079 | Specify the macro call function's O macro file number, such as O9xxx.<br>Ex: When this special D is written to K9100 and the 16th macro call is triggered with special **M2x047**, the NC system will execute the O9100 program. | R/W | Decimal | 1 ~ 65,535 |
| MLC to NC Variable 1 | D2x128 | The system will move data from this special D to NC variable #25128. | R/W | Decimal | 0 ~ 65,535 |
| MLC to NC Variable 2 | D2x129 | The system will move data from this special D to NC variable #25129.<br>*Both this special D and the variable # are NOT available to use when using MLC floating mode **[N1.010 Bit7=1]**. Otherwise, the system will return error 0x235. | R/W | Decimal | 0 ~ 65,535 |
| MLC to NC Variable 3 | D2x130 | The system will move data from this special D to NC variable #25130. | R/W | Decimal | 0 ~ 65,535 |
| MLC to NC Variable 4 | D2x131 | The system will move data from this special D to NC variable #25131.<br>*Both this special D and the variable # are NOT available to use when using MLC floating mode **[N1.010 Bit7=1]**. Otherwise, the system will return error 0x235. | R/W | Decimal | 0 ~ 65,535 |
| MLC to NC Variable 5 | D2x132 | The system will move data from this special D to NC variable #25132. | R/W | Decimal | 0 ~ 65,535 |
| MLC to NC Variable 6 | D2x133 | The system will move data from this special D to NC variable #25133.<br>*Both this special D and the variable # are NOT available to use when using MLC floating mode **[N1.010 Bit7=1]**. Otherwise, the system will return error 0x235. | R/W | Decimal | 0 ~ 65,535 |
| MLC to NC Variable 7 | D2x134 | The system will move data from this special D to NC variable #25134. | R/W | Decimal | 0 ~ 65,535 |
| MLC to NC Variable 8 | D2x135 | The system will move data from this special D to NC variable #25135.<br>*Both this special D and the variable # are NOT available to use when using MLC floating mode **[N1.010 Bit7=1]**. Otherwise, the system will return error 0x235. | R/W | Decimal | 0 ~ 65,535 |

| Function Name | Special D | Description | Device | Type | Range |
|---|---|---|---|---|---|
| MLC to NC Variable 9 | D2x136 | The system will move data from this special D to NC variable #25136. | R/W | Decimal | 0 ~ 65,535 |
| MLC to NC Variable 10 | D2x137 | The system will move data from this special D to NC variable #25137.<br>*Both this special D and the variable # are NOT available to use when using MLC floating mode **[N1.010 Bit7=1]**. Otherwise, the system will return error 0x235. | R/W | Decimal | 0 ~ 65,535 |
| MLC to NC Variable 11 | D2x138 | The system will move data from this special D to NC variable #25138. | R/W | Decimal | 0 ~ 65,535 |
| MLC to NC Variable 12 | D2x139 | The system will move data from this special D to NC variable #25139.<br>*Both this special D and the variable # are NOT available to use when using MLC floating mode **[N1.010 Bit7=1]**. Otherwise, the system will return error 0x235. | R/W | Decimal | 0 ~ 65,535 |
| MLC to NC Variable 13 | D2x140 | The system will move data from this special D to NC variable #25140. | R/W | Decimal | 0 ~ 65,535 |
| MLC to NC Variable 14 | D2x141 | The system will move data from this special D to NC variable #25141.<br>*Both this special D and the variable # are NOT available to use when using MLC floating mode **[N1.010 Bit7=1]**. Otherwise, the system will return error 0x235. | R/W | Decimal | 0 ~ 65,535 |
| MLC to NC Variable 15 | D2x142 | The system will move data from this special D to NC variable #25142. | R/W | Decimal | 0 ~ 65,535 |
| MLC to NC Variable 16 | D2x143 | The system will move data from this special D to NC variable #25143.<br>*Both this special D and the variable # are NOT available to use when using MLC floating mode **[N1.010 Bit7=1]**. Otherwise, the system will return error 0x235. | R/W | Decimal | 0 ~ 65,535 |
| MLC to NC Variable 17 | D2x144 | The system will move data from this special D to NC variable #25144. | R/W | Decimal | 0 ~ 65,535 |
| MLC to NC Variable 18 | D2x145 | The system will move data from this special D to NC variable #25145.<br>*Both this special D and the variable # are NOT available to use when using MLC floating mode **[N1.010 Bit7=1]**. Otherwise, the system will return error 0x235. | R/W | Decimal | 0 ~ 65,535 |
| MLC to NC Variable 19 | D2x146 | The system will move data from this special D to NC variable #25146. | R/W | Decimal | 0 ~ 65,535 |

| Function Name | Special D | Description | Device | Type | Range |
|---|---|---|---|---|---|
| MLC to NC Variable 20 | D2x147 | The system will move data from this special D to NC variable #25147.<br>*Both this special D and the variable # are NOT available to use when using MLC floating mode **[N1.010 Bit7=1]**. Otherwise, the system will return error 0x235. | R/W | Decimal | 0 ~ 65,535 |
| MLC to NC Variable 21 | D2x148 | The system will move data from this special D to NC variable #25148. | R/W | Decimal | 0 ~ 65,535 |
| MLC to NC Variable 22 | D2x149 | The system will move data from this special D to NC variable #25149.<br>*Both this special D and the variable # are NOT available to use when using MLC floating mode **[N1.010 Bit7=1]**. Otherwise, the system will return error 0x235. | R/W | Decimal | 0 ~ 65,535 |
| MLC to NC Variable 23 | D2x150 | The system will move data from this special D to NC variable #25150. | R/W | Decimal | 0 ~ 65,535 |
| MLC to NC Variable 24 | D2x151 | The system will move data from this special D to NC variable #25151.<br>*Both this special D and the variable # are NOT available to use when using MLC floating mode **[N1.010 Bit7=1]**. Otherwise, the system will return error 0x235. | R/W | Decimal | 0 ~ 65,535 |
| MLC to NC Variable 25 | D2x152 | The system will move data from this special D to NC variable #25152. | R/W | Decimal | 0 ~ 65,535 |
| MLC to NC Variable 26 | D2x153 | The system will move data from this special D to NC variable #25153.<br>*Both this special D and the variable # are NOT available to use when using MLC floating mode **[N1.010 Bit7=1]**. Otherwise, the system will return error 0x235. | R/W | Decimal | 0 ~ 65,535 |
| MLC to NC Variable 27 | D2x154 | The system will move data from this special D to NC variable #25154. | R/W | Decimal | 0 ~ 65,535 |
| MLC to NC Variable 28 | D2x155 | The system will move data from this special D to NC variable #25155.<br>*Both this special D and the variable # are NOT available to use when using MLC floating mode **[N1.010 Bit7=1]**. Otherwise, the system will return error 0x235. | R/W | Decimal | 0 ~ 65,535 |
| MLC to NC Variable 29 | D2x156 | The system will move data from this special D to NC variable #25156. | R/W | Decimal | 0 ~ 65,535 |

| Function Name | Special D | Description | Device | Type | Range |
|---|---|---|---|---|---|
| MLC to NC Variable 30 | D2x157 | The system will move data from this special D to NC variable #25157.<br>*Both this special D and the variable # are NOT available to use when using MLC floating mode **[N1.010 Bit7=1]**. Otherwise, the system will return error 0x235. | R/W | Decimal | 0 ~ 65,535 |
| MLC to NC Variable 31 | D2x158 | The system will move data from this special D to NC variable #25158. | R/W | Decimal | 0 ~ 65,535 |
| MLC to NC Variable 32 | D2x159 | The system will move data from this special D to NC variable #25159.<br>*Both this special D and the variable # are NOT available to use when using MLC floating mode **[N1.010 Bit7=1]**. Otherwise, the system will return error 0x235. | R/W | Decimal | 0 ~ 65,535 |
| MLC to NC Variable 33 | D2x160 | The system will move data from this special D to NC variable #25160. | R/W | Decimal | 0 ~ 65,535 |
| MLC to NC Variable 34 | D2x161 | The system will move data from this special D to NC variable #25161.<br>*Both this special D and the variable # are NOT available to use when using MLC floating mode **[N1.010 Bit7=1]**. Otherwise, the system will return error 0x235. | R/W | Decimal | 0 ~ 65,535 |
| MLC to NC Variable 35 | D2x162 | The system will move data from this special D to NC variable #25162. | R/W | Decimal | 0 ~ 65,535 |
| MLC to NC Variable 36 | D2x163 | The system will move data from this special D to NC variable #25163.<br>*Both this special D and the variable # are NOT available to use when using MLC floating mode **[N1.010 Bit7=1]**. Otherwise, the system will return error 0x235. | R/W | Decimal | 0 ~ 65,535 |
| MLC to NC Variable 37 | D2x164 | The system will move data from this special D to NC variable #25164. | R/W | Decimal | 0 ~ 65,535 |
| MLC to NC Variable 38 | D2x165 | The system will move data from this special D to NC variable #25165.<br>*Both this special D and the variable # are NOT available to use when using MLC floating mode **[N1.010 Bit7=1]**. Otherwise, the system will return error 0x235. | R/W | Decimal | 0 ~ 65,535 |
| MLC to NC Variable 39 | D2x166 | The system will move data from this special D to NC variable #25166. | R/W | Decimal | 0 ~ 65,535 |

| Function Name | Special D | Description | Device | Type | Range |
|---|---|---|---|---|---|
| MLC to NC Variable 40 | D2x167 | The system will move data from this special D to NC variable #25167.<br>*Both this special D and the variable # are NOT available to use when using MLC floating mode **[N1.010 Bit7=1]**. Otherwise, the system will return error 0x235. | R/W | Decimal | 0 ~ 65,535 |
| MLC to NC Variable 41 | D2x168 | The system will move data from this special D to NC variable #25168. | R/W | Decimal | 0 ~ 65,535 |
| MLC to NC Variable 42 | D2x169 | The system will move data from this special D to NC variable #25169.<br>*Both this special D and the variable # are NOT available to use when using MLC floating mode **[N1.010 Bit7=1]**. Otherwise, the system will return error 0x235. | R/W | Decimal | 0 ~ 65,535 |
| MLC to NC Variable 43 | D2x170 | The system will move data from this special D to NC variable #25170. | R/W | Decimal | 0 ~ 65,535 |
| MLC to NC Variable 44 | D2x171 | The system will move data from this special D to NC variable #25171.<br>*Both this special D and the variable # are NOT available to use when using MLC floating mode **[N1.010 Bit7=1]**. Otherwise, the system will return error 0x235. | R/W | Decimal | 0 ~ 65,535 |
| MLC to NC Variable 45 | D2x172 | The system will move data from this special D to NC variable #25172. | R/W | Decimal | 0 ~ 65,535 |
| MLC to NC Variable 46 | D2x173 | The system will move data from this special D to NC variable #25173.<br>*Both this special D and the variable # are NOT available to use when using MLC floating mode **[N1.010 Bit7=1]**. Otherwise, the system will return error 0x235. | R/W | Decimal | 0 ~ 65,535 |
| MLC to NC Variable 47 | D2x174 | The system will move data from this special D to NC variable #25174. | R/W | Decimal | 0 ~ 65,535 |
| MLC to NC Variable 48 | D2x175 | The system will move data from this special D to NC variable #25175.<br>*Both this special D and the variable # are NOT available to use when using MLC floating mode **[N1.010 Bit7=1]**. Otherwise, the system will return error 0x235. | R/W | Decimal | 0 ~ 65,535 |
| MLC to NC Variable 49 | D2x176 | The system will move data from this special D to NC variable #25176. | R/W | Decimal | 0 ~ 65,535 |

5

| Function Name | Special D | Description | Device | Type | Range |
|---|---|---|---|---|---|
| MLC to NC Variable 50 | D2x177 | The system will move data from this special D to NC variable #25177.<br>*Both this special D and the variable # are NOT available to use when using MLC floating mode **[N1.010 Bit7=1]**. Otherwise, the system will return error 0x235. | R/W | Decimal | 0 ~ 65,535 |
| MLC to NC Variable 51 | D2x178 | The system will move data from this special D to NC variable #25178. | R/W | Decimal | 0 ~ 65,535 |
| MLC to NC Variable 52 | D2x179 | The system will move data from this special D to NC variable #25179.<br>*Both this special D and the variable # are NOT available to use when using MLC floating mode **[N1.010 Bit7=1]**. Otherwise, the system will return error 0x235. | R/W | Decimal | 0 ~ 65,535 |
| MLC to NC Variable 53 | D2x180 | The system will move data from this special D to NC variable #25180. | R/W | Decimal | 0 ~ 65,535 |
| MLC to NC Variable 54 | D2x181 | The system will move data from this special D to NC variable #25181.<br>*Both this special D and the variable # are NOT available to use when using MLC floating mode **[N1.010 Bit7=1]**. Otherwise, the system will return error 0x235. | R/W | Decimal | 0 ~ 65,535 |
| MLC to NC Variable 55 | D2x182 | The system will move data from this special D to NC variable #25182. | R/W | Decimal | 0 ~ 65,535 |
| MLC to NC Variable 56 | D2x183 | The system will move data from this special D to NC variable #25183.<br>*Both this special D and the variable # are NOT available to use when using MLC floating mode **[N1.010 Bit7=1]**. Otherwise, the system will return error 0x235. | R/W | Decimal | 0 ~ 65,535 |
| MLC to NC Variable 57 | D2x184 | The system will move data from this special D to NC variable #25184. | R/W | Decimal | 0 ~ 65,535 |
| MLC to NC Variable 58 | D2x185 | The system will move data from this special D to NC variable #25185.<br>*Both this special D and the variable # are NOT available to use when using MLC floating mode **[N1.010 Bit7=1]**. Otherwise, the system will return error 0x235. | R/W | Decimal | 0 ~ 65,535 |
| MLC to NC Variable 59 | D2x186 | The system will move data from this special D to NC variable #25186. | R/W | Decimal | 0 ~ 65,535 |

| Function Name | Special D | Description | Device | Type | Range |
|---|---|---|---|---|---|
| MLC to NC Variable 60 | D2x187 | The system will move data from this special D to NC variable #25187.<br>*Both this special D and the variable # are NOT available to use when using MLC floating mode **[N1.010 Bit7=1]**. Otherwise, the system will return error 0x235. | R/W | Decimal | 0 ~ 65,535 |
| MLC to NC Variable 61 | D2x188 | The system will move data from this special D to NC variable #25188. | R/W | Decimal | 0 ~ 65,535 |
| MLC to NC Variable 62 | D2x189 | The system will move data from this special D to NC variable #25189.<br>*Both this special D and the variable # are NOT available to use when using MLC floating mode **[N1.010 Bit7=1]**. Otherwise, the system will return error 0x235. | R/W | Decimal | 0 ~ 65,535 |
| MLC to NC Variable 63 | D2x190 | The system will move data from this special D to NC variable #25190. | R/W | Decimal | 0 ~ 65,535 |
| MLC to NC Variable 64 | D2x191 | The system will move data from this special D to NC variable #25191.<br>*Both this special D and the variable # are NOT available to use when using MLC floating mode **[N1.010 Bit7=1]**. Otherwise, the system will return error 0x235. | R/W | Decimal | 0 ~ 65,535 |
| MLC to NC Variable 65 | D2x192 | The system will move data from this special D to NC variable #25192. | R/W | Decimal | 0 ~ 65,535 |
| MLC to NC Variable 66 | D2x193 | The system will move data from this special D to NC variable #25193.<br>*Both this special D and the variable # are NOT available to use when using MLC floating mode **[N1.010 Bit7=1]**. Otherwise, the system will return error 0x235. | R/W | Decimal | 0 ~ 65,535 |
| MLC to NC Variable 67 | D2x194 | The system will move data from this special D to NC variable #25194. | R/W | Decimal | 0 ~ 65,535 |
| MLC to NC Variable 68 | D2x195 | The system will move data from this special D to NC variable #25195.<br>*Both this special D and the variable # are NOT available to use when using MLC floating mode **[N1.010 Bit7=1]**. Otherwise, the system will return error 0x235. | R/W | Decimal | 0 ~ 65,535 |
| MLC to NC Variable 69 | D2x196 | The system will move data from this special D to NC variable #25196. | R/W | Decimal | 0 ~ 65,535 |

5

| Function Name | Special D | Description | Device | Type | Range |
|---|---|---|---|---|---|
| MLC to NC Variable 70 | D2x197 | The system will move data from this special D to NC variable #25197.<br>*Both this special D and the variable # are NOT available to use when using MLC floating mode **[N1.010 Bit7=1]**. Otherwise, the system will return error 0x235. | R/W | Decimal | 0 ~ 65,535 |
| MLC to NC Variable 71 | D2x198 | The system will move data from this special D to NC variable #25198. | R/W | Decimal | 0 ~ 65,535 |
| MLC to NC Variable 72 | D2x199 | The system will move data from this special D to NC variable #25199.<br>*Both this special D and the variable # are NOT available to use when using MLC floating mode **[N1.010 Bit7=1]**. Otherwise, the system will return error 0x235. | R/W | Decimal | 0 ~ 65,535 |
| MLC to NC Variable 73 | D2x200 | The system will move data from this special D to NC variable #25200. | R/W | Decimal | 0 ~ 65,535 |
| MLC to NC Variable 74 | D2x201 | The system will move data from this special D to NC variable #25201.<br>*Both this special D and the variable # are NOT available to use when using MLC floating mode **[N1.010 Bit7=1]**. Otherwise, the system will return error 0x235. | R/W | Decimal | 0 ~ 65,535 |
| MLC to NC Variable 75 | D2x202 | The system will move data from this special D to NC variable #25202. | R/W | Decimal | 0 ~ 65,535 |
| MLC to NC Variable 76 | D2x203 | The system will move data from this special D to NC variable #25203.<br>*Both this special D and the variable # are NOT available to use when using MLC floating mode **[N1.010 Bit7=1]**. Otherwise, the system will return error 0x235. | R/W | Decimal | 0 ~ 65,535 |
| MLC to NC Variable 77 | D2x204 | The system will move data from this special D to NC variable #25204. | R/W | Decimal | 0 ~ 65,535 |
| MLC to NC Variable 78 | D2x205 | The system will move data from this special D to NC variable #25205.<br>*Both this special D and the variable # are NOT available to use when using MLC floating mode **[N1.010 Bit7=1]**. Otherwise, the system will return error 0x235. | R/W | Decimal | 0 ~ 65,535 |
| MLC to NC Variable 79 | D2x206 | The system will move data from this special D to NC variable #25206. | R/W | Decimal | 0 ~ 65,535 |

| Function Name | Special D | Description | Device | Type | Range |
|---|---|---|---|---|---|
| MLC to NC Variable 80 | D2x207 | The system will move data from this special D to NC variable #25207.<br>*Both this special D and the variable # are NOT available to use when using MLC floating mode **[N1.010 Bit7=1]**. Otherwise, the system will return error 0x235. | R/W | Decimal | 0 ~ 65,535 |
| MLC to NC Variable 81 | D2x208 | The system will move data from this special D to NC variable #25208. | R/W | Decimal | 0 ~ 65,535 |
| MLC to NC Variable 82 | D2x209 | The system will move data from this special D to NC variable #25209.<br>*Both this special D and the variable # are NOT available to use when using MLC floating mode **[N1.010 Bit7=1]**. Otherwise, the system will return error 0x235. | R/W | Decimal | 0 ~ 65,535 |
| MLC to NC Variable 83 | D2x210 | The system will move data from this special D to NC variable #25210. | R/W | Decimal | 0 ~ 65,535 |
| MLC to NC Variable 84 | D2x211 | The system will move data from this special D to NC variable #25211.<br>*Both this special D and the variable # are NOT available to use when using MLC floating mode **[N1.010 Bit7=1]**. Otherwise, the system will return error 0x235. | R/W | Decimal | 0 ~ 65,535 |
| MLC to NC Variable 85 | D2x212 | The system will move data from this special D to NC variable #25212. | R/W | Decimal | 0 ~ 65,535 |
| MLC to NC Variable 86 | D2x213 | The system will move data from this special D to NC variable #25213.<br>*Both this special D and the variable # are NOT available to use when using MLC floating mode **[N1.010 Bit7=1]**. Otherwise, the system will return error 0x235. | R/W | Decimal | 0 ~ 65,535 |
| MLC to NC Variable 87 | D2x214 | The system will move data from this special D to NC variable #25214. | R/W | Decimal | 0 ~ 65,535 |
| MLC to NC Variable 88 | D2x215 | The system will move data from this special D to NC variable #25215.<br>*Both this special D and the variable # are NOT available to use when using MLC floating mode **[N1.010 Bit7=1]**. Otherwise, the system will return error 0x235. | R/W | Decimal | 0 ~ 65,535 |
| MLC to NC Variable 89 | D2x216 | The system will move data from this special D to NC variable #25216. | R/W | Decimal | 0 ~ 65,535 |

5

| Function Name | Special D | Description | Device | Type | Range |
|---|---|---|---|---|---|
| MLC to NC Variable 90 | D2x217 | The system will move data from this special D to NC variable #25217.<br>*Both this special D and the variable # are NOT available to use when using MLC floating mode **[N1.010 Bit7=1]**. Otherwise, the system will return error 0x235. | R/W | Decimal | 0 ~ 65,535 |
| MLC to NC Variable 91 | D2x218 | The system will move data from this special D to NC variable #25218. | R/W | Decimal | 0 ~ 65,535 |
| MLC to NC Variable 92 | D2x219 | The system will move data from this special D to NC variable #25219.<br>*Both this special D and the variable # are NOT available to use when using MLC floating mode **[N1.010 Bit7=1]**. Otherwise, the system will return error 0x235. | R/W | Decimal | 0 ~ 65,535 |
| MLC to NC Variable 93 | D2x220 | The system will move data from this special D to NC variable #25220. | R/W | Decimal | 0 ~ 65,535 |
| MLC to NC Variable 94 | D2x221 | The system will move data from this special D to NC variable #25221.<br>*Both this special D and the variable # are NOT available to use when using MLC floating mode **[N1.010 Bit7=1]**. Otherwise, the system will return error 0x235. | R/W | Decimal | 0 ~ 65,535 |
| MLC to NC Variable 95 | D2x222 | The system will move data from this special D to NC variable #25222. | R/W | Decimal | 0 ~ 65,535 |
| MLC to NC Variable 96 | D2x223 | The system will move data from this special D to NC variable #25223.<br>*Both this special D and the variable # are NOT available to use when using MLC floating mode **[N1.010 Bit7=1]**. Otherwise, the system will return error 0x235. | R/W | Decimal | 0 ~ 65,535 |
| MLC to NC Variable 97 | D2x224 | The system will move data from this special D to NC variable #25224. | R/W | Decimal | 0 ~ 65,535 |
| MLC to NC Variable 98 | D2x225 | The system will move data from this special D to NC variable #25225.<br>*Both this special D and the variable # are NOT available to use when using MLC floating mode **[N1.010 Bit7=1]**. Otherwise, the system will return error 0x235. | R/W | Decimal | 0 ~ 65,535 |
| MLC to NC Variable 99 | D2x226 | The system will move data from this special D to NC variable #25226. | R/W | Decimal | 0 ~ 65,535 |

| Function Name | Special D | Description | Device | Type | Range |
|---|---|---|---|---|---|
| MLC to NC Variable 100 | D2x227 | The system will move data from this special D to NC variable #25227.<br>*Both this special D and the variable # are NOT available to use when using MLC floating mode **[N1.010 Bit7=1]**. Otherwise, the system will return error 0x235. | R/W | Decimal | 0 ~ 65,535 |
| MLC to NC Variable 101 | D2x228 | The system will move data from this special D to NC variable #25228. | R/W | Decimal | 0 ~ 65,535 |
| MLC to NC Variable 102 | D2x229 | The system will move data from this special D to NC variable #25229.<br>*Both this special D and the variable # are NOT available to use when using MLC floating mode **[N1.010 Bit7=1]**. Otherwise, the system will return error 0x235. | R/W | Decimal | 0 ~ 65,535 |
| MLC to NC Variable 103 | D2x230 | The system will move data from this special D to NC variable #25230. | R/W | Decimal | 0 ~ 65,535 |
| MLC to NC Variable 104 | D2x231 | The system will move data from this special D to NC variable #25231.<br>*Both this special D and the variable # are NOT available to use when using MLC floating mode **[N1.010 Bit7=1]**. Otherwise, the system will return error 0x235. | R/W | Decimal | 0 ~ 65,535 |
| MLC to NC Variable 105 | D2x232 | The system will move data from this special D to NC variable #25232. | R/W | Decimal | 0 ~ 65,535 |
| MLC to NC Variable 106 | D2x233 | The system will move data from this special D to NC variable #25233.<br>*Both this special D and the variable # are NOT available to use when using MLC floating mode **[N1.010 Bit7=1]**. Otherwise, the system will return error 0x235. | R/W | Decimal | 0 ~ 65,535 |
| MLC to NC Variable 107 | D2x234 | The system will move data from this special D to NC variable #25234. | R/W | Decimal | 0 ~ 65,535 |
| MLC to NC Variable 108 | D2x235 | The system will move data from this special D to NC variable #25235.<br>*Both this special D and the variable # are NOT available to use when using MLC floating mode **[N1.010 Bit7=1]**. Otherwise, the system will return error 0x235. | R/W | Decimal | 0 ~ 65,535 |
| MLC to NC Variable 109 | D2x236 | The system will move data from this special D to NC variable #25236. | R/W | Decimal | 0 ~ 65,535 |

| Function Name | Special D | Description | Device | Type | Range |
|---|---|---|---|---|---|
| MLC to NC Variable 110 | D2x237 | The system will move data from this special D to NC variable #25237.<br>*Both this special D and the variable # are NOT available to use when using MLC floating mode **[N1.010 Bit7=1]**. Otherwise, the system will return error 0x235. | R/W | Decimal | 0 ~ 65,535 |
| MLC to NC Variable 111 | D2x238 | The system will move data from this special D to NC variable #25238. | R/W | Decimal | 0 ~ 65,535 |
| MLC to NC Variable 112 | D2x239 | The system will move data from this special D to NC variable #25239.<br>*Both this special D and the variable # are NOT available to use when using MLC floating mode **[N1.010 Bit7=1]**. Otherwise, the system will return error 0x235. | R/W | Decimal | 0 ~ 65,535 |
| MLC to NC Variable 113 | D2x240 | The system will move data from this special D to NC variable #25240. | R/W | Decimal | 0 ~ 65,535 |
| MLC to NC Variable 114 | D2x241 | The system will move data from this special D to NC variable #25241.<br>*Both this special D and the variable # are NOT available to use when using MLC floating mode **[N1.010 Bit7=1]**. Otherwise, the system will return error 0x235. | R/W | Decimal | 0 ~ 65,535 |
| MLC to NC Variable 115 | D2x242 | The system will move data from this special D to NC variable #25242. | R/W | Decimal | 0 ~ 65,535 |
| MLC to NC Variable 116 | D2x243 | The system will move data from this special D to NC variable #25243.<br>*Both this special D and the variable # are NOT available to use when using MLC floating mode **[N1.010 Bit7=1]**. Otherwise, the system will return error 0x235. | R/W | Decimal | 0 ~ 65,535 |
| MLC to NC Variable 117 | D2x244 | The system will move data from this special D to NC variable #25244. | R/W | Decimal | 0 ~ 65,535 |
| MLC to NC Variable 118 | D2x245 | The system will move data from this special D to NC variable #25245.<br>*Both this special D and the variable # are NOT available to use when using MLC floating mode **[N1.010 Bit7=1]**. Otherwise, the system will return error 0x235. | R/W | Decimal | 0 ~ 65,535 |
| MLC to NC Variable 119 | D2x246 | The system will move data from this special D to NC variable #25246. | R/W | Decimal | 0 ~ 65,535 |

| Function Name | Special D | Description | Device | Type | Range |
|---|---|---|---|---|---|
| MLC to NC Variable 120 | D2x247 | The system will move data from this special D to NC variable #25247.<br>*Both this special D and the variable # are NOT available to use when using MLC floating mode **[N1.010 Bit7=1]**. Otherwise, the system will return error 0x235. | R/W | Decimal | 0 ~ 65,535 |
| MLC to NC Variable 121 | D2x248 | The system will move data from this special D to NC variable #25248. | R/W | Decimal | 0 ~ 65,535 |
| MLC to NC Variable 122 | D2x249 | The system will move data from this special D to NC variable #25249.<br>*Both this special D and the variable # are NOT available to use when using MLC floating mode **[N1.010 Bit7=1]**. Otherwise, the system will return error 0x235. | R/W | Decimal | 0 ~ 65,535 |
| MLC to NC Variable 123 | D2x250 | The system will move data from this special D to NC variable #25250. | R/W | Decimal | 0 ~ 65,535 |
| MLC to NC Variable 124 | D2x251 | The system will move data from this special D to NC variable #25251.<br>*Both this special D and the variable # are NOT available to use when using MLC floating mode **[N1.010 Bit7=1]**. Otherwise, the system will return error 0x235. | R/W | Decimal | 0 ~ 65,535 |
| MLC to NC Variable 125 | D2x252 | The system will move data from this special D to NC variable #25252. | R/W | Decimal | 0 ~ 65,535 |
| MLC to NC Variable 126 | D2x253 | The system will move data from this special D to NC variable #25253.<br>*Both this special D and the variable # are NOT available to use when using MLC floating mode **[N1.010 Bit7=1]**. Otherwise, the system will return error 0x235. | R/W | Decimal | 0 ~ 65,535 |
| MLC to NC Variable 127 | D2x254 | The system will move data from this special D to NC variable #25254. | R/W | Decimal | 0 ~ 65,535 |
| MLC to NC Variable 128 | D2x255 | The system will move data from this special D to NC variable #25255.<br>*Both this special D and the variable # are NOT available to use when using MLC floating mode **[N1.010 Bit7=1]**. Otherwise, the system will return error 0x235. | R/W | Decimal | 0 ~ 65,535 |
| Target Position of X axis<br>(MLC Axis) | D2x256<br>D2x257 | Specifies the target position of the X axis in MLC axis mode.<br>Unit: mm or inch. | R/W | Float | -2,147,483,648<br>~<br>+2,147,483,647 |

5

| Function Name | Special D | Description | Device | Type | Range |
|---|---|---|---|---|---|
| Target Position of Y axis (MLC Axis) | D2x258 D2x259 | Specifies the target position of the Y axis in MLC axis mode. Unit: mm or inch. | R/W | Float | -2,147,483,648 ~ +2,147,483,647 |
| Target Position of Z axis (MLC Axis) | D2x260 D2x261 | Specifies the target position of the Z axis in MLC axis mode. Unit: mm or inch. | R/W | Float | -2,147,483,648 ~ +2,147,483,647 |
| Target Position of A axis (MLC Axis) | D2x262 D2x263 | Specifies the target position of the A axis in MLC axis mode. Unit: mm or inch. | R/W | Float | -2,147,483,648 ~ +2,147,483,647 |
| Target Position of B axis (MLC Axis) | D2x264 D2x265 | Specifies the target position of the B axis in MLC axis mode. Unit: mm or inch. | R/W | Float | -2,147,483,648 ~ +2,147,483,647 |
| Target Position of C axis (MLC Axis) | D2x266 D2x267 | Specifies the target position of the C axis in MLC axis mode. Unit: mm or inch. | R/W | Float | -2,147,483,648 ~ +2,147,483,647 |
| Target Position of U axis (MLC Axis) | D2x268 D2x269 | Specifies the target position of the U axis in MLC axis mode. Unit: mm or inch. | R/W | Float | -2,147,483,648 ~ +2,147,483,647 |
| Target Position of V axis (MLC Axis) | D2x270 D2x271 | Specifies the target position of the V axis in MLC axis mode. Unit: mm or inch. | R/W | Float | -2,147,483,648 ~ +2,147,483,647 |
| Target Position of W axis (MLC Axis) | D2x272 D2x273 | Specifies the target position of the W axis in MLC axis mode. Unit: mm or inch. | R/W | Float | -2,147,483,648 ~ +2,147,483,647 |
| Target Position of $10^{th}$ axis (MLC Axis) | D2x274 D2x275 | Specifies the target position of the $10^{th}$ axis in MLC axis mode. Unit: mm or inch. | R/W | Float | -2,147,483,648 ~ +2,147,483,647 |
| Target Position of $11^{th}$ axis (MLC Axis) | D2x276 D2x277 | Specifies the target position of the $11^{th}$ axis in MLC axis mode. Unit: mm or inch. | R/W | Float | -2,147,483,648 ~ +2,147,483,647 |
| Target Position of $12^{th}$ axis (MLC Axis) | D2x278 D2x279 | Specifies the target position of the $12^{th}$ axis in MLC axis mode. Unit: mm or inch. | R/W | Float | -2,147,483,648 ~ +2,147,483,647 |
| Target Position of $13^{th}$ axis (MLC Axis) | D2x280 D2x281 | Specifies the target position of the $13^{th}$ axis in MLC axis mode. Unit: mm or inch. | R/W | Float | -2,147,483,648 ~ +2,147,483,647 |
| Target Position of $14^{th}$ axis (MLC Axis) | D2x282 D2x283 | Specifies the target position of the $14^{th}$ axis in MLC axis mode. Unit: mm or inch. | R/W | Float | -2,147,483,648 ~ +2,147,483,647 |
| Target Position of $15^{th}$ axis (MLC Axis) | D2x284 D2x285 | Specifies the target position of the $15^{th}$ axis in MLC axis mode. Unit: mm or inch. | R/W | Float | -2,147,483,648 ~ +2,147,483,647 |

| Function Name | Special D | Description | Device | Type | Range |
|---|---|---|---|---|---|
| Target Position of 16th axis (MLC Axis) | D2x286 D2x287 | Specifies the target position of the 16th axis in MLC axis mode. Unit: mm or inch. | R/W | Float | -2,147,483,648 ~ +2,147,483,647 |
| Target Velocity of X axis (MLC Axis) | D2x288 D2x289 | Specifies the target velocity of the X axis in MLC axis mode. Unit: mm/min, inch/min. | R/W | Float | -2,147,483,648 ~ +2,147,483,647 |
| Target Velocity of Y axis (MLC Axis) | D2x290 D2x291 | Specifies the target velocity of the Y axis in MLC axis mode. Unit: mm/min, inch/min. | R/W | Float | -2,147,483,648 ~ +2,147,483,647 |
| Target Velocity of Z axis (MLC Axis) | D2x292 D2x293 | Specifies the target velocity of the Z axis in MLC axis mode. Unit: mm/min, inch/min. | R/W | Float | -2,147,483,648 ~ +2,147,483,647 |
| Target Velocity of A axis (MLC Axis) | D2x294 D2x295 | Specifies the target velocity of the A axis in MLC axis mode. Unit: RPM. | R/W | Float | -2,147,483,648 ~ +2,147,483,647 |
| Target Velocity of B axis (MLC Axis) | D2x296 D2x297 | Specifies the target velocity of the B axis in MLC axis mode. Unit: RPM. | R/W | Float | -2,147,483,648 ~ +2,147,483,647 |
| Target Velocity of C axis (MLC Axis) | D2x298 D2x299 | Specifies the target velocity of the C axis in MLC axis mode. Unit: RPM. | R/W | Float | -2,147,483,648 ~ +2,147,483,647 |
| Target Velocity of U axis (MLC Axis) | D2x300 D2x301 | Specifies the target velocity of the U axis in MLC axis mode. Unit: mm/min, inch/min. | R/W | Float | -2,147,483,648 ~ +2,147,483,647 |
| Target Velocity of V axis (MLC Axis) | D2x302 D2x303 | Specifies the target velocity of the V axis in MLC axis mode. Unit: mm/min, inch/min. | R/W | Float | -2,147,483,648 ~ +2,147,483,647 |
| Target Velocity of W axis (MLC Axis) | D2x304 D2x305 | Specifies the target velocity of the W axis in MLC axis mode. Unit: mm/min, inch/min. | R/W | Float | -2,147,483,648 ~ +2,147,483,647 |
| Target Velocity of 10th axis (MLC Axis) | D2x306 D2x307 | Specifies the target velocity of the 10th axis in MLC axis mode. Unit: mm/min, inch/min. | R/W | Float | -2,147,483,648 ~ +2,147,483,647 |
| Target Velocity of 11th axis (MLC Axis) | D2x308 D2x309 | Specifies the target velocity of the 11th axis in MLC axis mode. Unit: mm/min, inch/min. | R/W | Float | -2,147,483,648 ~ +2,147,483,647 |
| Target Velocity of 12th axis (MLC Axis) | D2x310 D2x311 | Specifies the target velocity of the 12th axis in MLC axis mode. Unit: mm/min, inch/min. | R/W | Float | -2,147,483,648 ~ +2,147,483,647 |
| Target Velocity of 13th axis (MLC Axis) | D2x312 D2x313 | Specifies the target velocity of the 13th axis in MLC axis mode. Unit: mm/min, inch/min. | R/W | Float | -2,147,483,648 ~ +2,147,483,647 |

5

| Function Name | Special D | Description | Device | Type | Range |
|---|---|---|---|---|---|
| Target Velocity of 14th axis (MLC Axis) | D2x314 D2x315 | Specifies the target velocity of the 14th axis in MLC axis mode. Unit: mm/min, inch/min. | R/W | Float | -2,147,483,648 ~ +2,147,483,647 |
| Target Velocity of 15th axis (MLC Axis) | D2x316 D2x317 | Specifies the target velocity of the 15th axis in MLC axis mode. Unit: mm/min, inch/min. | R/W | Float | -2,147,483,648 ~ +2,147,483,647 |
| Target Velocity of 16th axis (MLC Axis) | D2x318 D2x319 | Specifies the target velocity of the 16th axis in MLC axis mode. Unit: mm/min, inch/min. | R/W | Float | -2,147,483,648 ~ +2,147,483,647 |
| 3rd Spindle Speed | D2x320 D2x321 | Write the 3rd spindle's speed through the special D (in accordance with **M2x742**). | R/W | Decimal | 0 ~ 4,294,967,295 |
| 3rd Spindle Speed Rate | D2x322 | Setting the spindle's speed ratio. Ex: When the program speed is S1000 and this special D is 30, then the NC system will execute the spindle speed as 1000 x 30% = 300 RPM. | R/W | Decimal | 0 ~ 65,535 |
| 3rd Spindle Gear Ratio Selection | D2x323 | Select in accordance with **N0.1134 ~ N0.1141**. Ex: When this special D is set to 1, the system will set the 3rd spindle to 1st gear, when it is set to 2, it will be in 2nd gear, and so forth. | R/W | Decimal | 0 ~ 65,535 |
| 4th Spindle Speed | D2x326 D2x327 | Write the 4th spindle's speed through the special D (in accordance with **M2x758**). | R/W | Decimal | 0 ~ 4,294,967,295 |
| 4th Spindle Speed Rate | D2x328 | Setting the spindle's speed ratio. Ex: When the program speed is S1000 and this special D is 30, then the NC system will execute the spindle speed as 1000 x 30% = 300 RPM. | R/W | Decimal | 0 ~ 65,535 |
| 4th Spindle Gear Ratio Selection | D2x329 | Select in accordance with **N0.1184 ~ N0.1191**. Ex: When this special D is set to 1, the system will set the 4th spindle to 1st gear, when it is set to 2, it will be in 2nd gear, and so forth. | R/W | Decimal | 0 ~ 65,535 |
| 5th Spindle Speed | D2x332 D2x333 | Write the 5th spindle's speed through the special D (in accordance with **M2x774**). | R/W | Decimal | 0 ~ 4,294,967,295 |
| 5th Spindle Speed Rate | D2x334 | Setting the spindle's speed ratio. Ex: When the program speed is S1000 and this special D is 30, then the NC system will execute the spindle speed as 1000 x 30% = 300 RPM. | R/W | Decimal | 0 ~ 65,535 |
| 5th Spindle Gear Ratio Selection | D2x335 | Select in accordance with **N0.1234 ~ N0.1241**. Ex: When this special D is set to 1, the system will set the 5tht spindle to 1st gear, when it is set to 2, it will be in 2nd gear, and so forth. | R/W | Decimal | 0 ~ 65,535 |

| Function Name | Special D | Description | Device | Type | Range |
|---|---|---|---|---|---|
| 6th Spindle Speed | D2x338<br>D2x339 | Write the 6th spindle's speed through the special D (in accordance with **M2x790**). | R/W | Decimal | 0 ~ 4,294,967,295 |
| 6th Spindle Speed Rate | D2x340 | Setting the spindle's speed ratio.<br>Ex: When the program speed is S1000 and this special D is 30, then the NC system will execute the spindle speed as 1000 x 30% = 300 RPM. | R/W | Decimal | 0 ~ 65,535 |
| 6th Spindle Gear Ratio Selection | D2x341 | Select in accordance with **N0.1284 ~ N0.1291**. Ex: When this special D is set to 1, the system will set the 6th spindle to 1st gear, when it is set to 2, it will be in 2nd gear, and so forth. | R/W | Decimal | 0 ~ 65,535 |
| 7th Spindle Speed | D2x344<br>D2x345 | Write the 7th spindle's speed through the special D (in accordance with **M2x806**). | R/W | Decimal | 0 ~ 4,294,967,295 |
| 7th Spindle Speed Rate | D2x346 | Setting the spindle's speed ratio.<br>Ex: When the program speed is S1000 and this special D is 30, then the NC system will execute the spindle speed as 1000 x 30% = 300 RPM. | R/W | Decimal | 0 ~ 65,535 |
| 7th Spindle Gear Ratio Selection | D2x347 | Select in accordance with **N0.1334 ~ N0.1341**. Ex: When this special D is set to 1, the system will set the 7th spindle to 1st gear, when it is set to 2, it will be in 2nd gear, and so forth. | R/W | Decimal | 0 ~ 65,535 |
| 8th Spindle Speed | D2x350<br>D2x351 | Write the 8th spindle's speed through the special D (in accordance with **M2x822**). | R/W | Decimal | 0 ~ 4,294,967,295 |
| 8th Spindle Speed Rate | D2x352 | Setting the spindle's speed ratio.<br>Ex: When the program speed is S1000 and this special D is 30, then the NC system will execute the spindle speed as 1000 x 30% = 300 RPM. | R/W | Decimal | 0 ~ 65,535 |
| 8th Spindle Gear Ratio Selection | D2x353 | Select in accordance with **N0.1384 ~ N0.1391**. Ex: When this special D is set to 1, the system will set the 8th spindle to 1st gear, when it is set to 2, it will be in 2nd gear, and so forth. | R/W | Decimal | 0 ~ 65,535 |

5

### 5.2.3.2 D49000 to D49899

| Function Name | Special D | Description | Device | Type | Range |
|---|---|---|---|---|---|
| Complete Process Amount | D49x00<br>D49x01 | Available to set this special D from system interface or MLC. | R/W | Decimal | 0 ~ 4,294,967,295 |
| Process Target Amount | D49x02<br>D49x03 | Available to set this special D from system interface or MLC. | R/W | Decimal | 0 ~ 4,294,967,295 |
| Total Process Time | D49x04<br>D49x05 | When the system parameter **[N6.032 Process Time Record]** is 1, the system will automatically record the total process time, in units of seconds. | R/W | Decimal | 0 ~ 4,294,967,295 |
| Single Process Time | D49x06<br>D49x07 | When the system parameter **[N6.032 Process Time Record]** is 1, the system will automatically record the single process time, in units of seconds. | R/W | Decimal | 0 ~ 4,294,967,295 |

## 5.2.4 List of special D (System status)

### 5.2.4.1 D30000 to D38999

| Function Name | Special D | Description | Device | Type | Range |
|---|---|---|---|---|---|
| Pulse Feedback of Spindle 1 | D30000 | Pulse feedback of spindle 1 connector. | R | Decimal | 0 ~ 65,535 |
| Pulse Feedback of Spindle 2 | D30001 | Pulse feedback of spindle 2 connector. | R | Decimal | 0 ~ 65,535 |
| Z Phase Pulse Feedback of Spindle 1 | D30002 | Z phase pulse feedback of spindle 1 connector. | R | Decimal | 0 ~ 65,535 |
| Z Phase Pulse Feedback of Spindle 2 | D30003 | Z phase pulse feedback of spindle 2 connector. | R | Decimal | 0 ~ 65,535 |
| MPG Pulse Feedback | D30004 | Pulse feedback of MPG connector. | R | Decimal | 0 ~ 65,535 |
| Pulse Output of Spindle 1 | D30008 | Pulse output of spindle 1 connector. | R | Decimal | 0 ~ 65,535 |
| Pulse Output of Spindle 2 | D30009 | Pulse output of spindle 2 connector. | R | Decimal | 0 ~ 65,535 |
| Spindle 1 Pulse Feedback Coordinate | D30174<br>D30175 | Display the pulse feedback coordinate from the Spindle 1 connector.<br>The NC system will calculate the pulse feedback from the Spindle 1 connector as the 1st spindle's feedback coordinate, which based on parameter setting of **[N0.030 ~ N0.032]**.<br>When the [N0.030] set as rotary axis, this special D register will display the position between 0 ~ 359.999. | R | Float | -2,147,483,648 ~ +2,147,483,647 |
| Spindle 2 Pulse Feedback Coordinate | D30176<br>D30177 | Display the pulse feedback coordinate from the Spindle 2 connector.<br>The NC system will calculate the pulse feedback from the Spindle 2 connector as the 2nd spindle's feedback coordinate, which based on parameter setting of **[N0.035 ~ N0.037]**.<br>When the [N0.035] set as rotary axis, this special D register will display the position between 0 ~ 359.999. | R | Float | -2,147,483,648 ~ +2,147,483,647 |
| 1st Remote Module Input | D30200<br>D30201<br>D30202<br>D30203 | For connected modules that do not include digital input/output (DI/O), the system will set the arrangement order according to the EtherCAT module and will set the special D module with the special D based on the connection order (such as analog modules).<br>Taking the first module as an example, D30200 corresponds to the first set of values on the module, D30201 corresponds to the second set of values on the module, and so forth. | R | Decimal | 0 ~ 65,535 |
| 2nd Remote Module Input | D30204<br>D30205<br>D30206<br>D30207 | | R | Decimal | 0 ~ 65,535 |
| 3rd Remote Module Input | D30208<br>D30209<br>D30210<br>D30211 | | R | Decimal | 0 ~ 65,535 |

5

| Function Name | Special D | Description | Device | Type | Range |
|---|---|---|---|---|---|
| 4th Remote Module Input | D30212<br>D30213<br>D30214<br>D30215 | For connected modules that do not include digital input/output (DI/O), the system will set the arrangement order according to the EtherCAT module and will set the special D module with the special D based on the connection order (such as analog modules).<br>Taking the first module as an example, D30200 corresponds to the first set of values on the module, D30201 corresponds to the second set of values on the module, and so forth. | R | Decimal | 0 ~ 65,535 |
| 5th Remote Module Input | D30216<br>D30217<br>D30218<br>D30219 | | R | Decimal | 0 ~ 65,535 |
| 6th Remote Module Input | D30220<br>D30221<br>D30222<br>D30223 | | R | Decimal | 0 ~ 65,535 |
| 7th Remote Module Input | D30224<br>D30225<br>D30226<br>D30227 | | R | Decimal | 0 ~ 65,535 |
| 8th Remote Module Input | D30228<br>D30229<br>D30230<br>D30231 | | R | Decimal | 0 ~ 65,535 |
| Torque Feedback of 1st Axis | D30240 | Torque feedback for the axis. The axis index is set according to the NC channel axis sequence setting.<br>PDO address 0x6077H<br>Unit: 0.1% | R | Decimal | -32,768 ~ +32,767 |
| Torque Feedback of 2nd Axis | D30241 | | R | Decimal | -32,768 ~ +32,767 |
| Torque Feedback of 3rd Axis | D30242 | | R | Decimal | -32,768 ~ +32,767 |
| Torque Feedback of 4th Axis | D30243 | | R | Decimal | -32,768 ~ +32,767 |
| Torque Feedback of 5th Axis | D30244 | | R | Decimal | -32,768 ~ +32,767 |
| Torque Feedback of 6th Axis | D30245 | | R | Decimal | -32,768 ~ +32,767 |
| Torque Feedback of 7th Axis | D30246 | | R | Decimal | -32,768 ~ +32,767 |
| Torque Feedback of 8th Axis | D30247 | | R | Decimal | -32,768 ~ +32,767 |
| Torque Feedback of 9th Axis | D30248 | | R | Decimal | -32,768 ~ +32,767 |
| Torque Feedback of 10th Axis | D30249 | | R | Decimal | -32,768 ~ +32,767 |
| Torque Feedback of 11th Axis | D30250 | | R | Decimal | -32,768 ~ +32,767 |
| Torque Feedback of 12th Axis | D30251 | | R | Decimal | -32,768 ~ +32,767 |
| Torque Feedback of 13th Axis | D30252 | | R | Decimal | -32,768 ~ +32,767 |
| Torque Feedback of 14th Axis | D30253 | | R | Decimal | -32,768 ~ +32,767 |
| Torque Feedback of 15th Axis | D30254 | | R | Decimal | -32,768 ~ +32,767 |
| Torque Feedback of 16th Axis | D30255 | | R | Decimal | -32,768 ~ +32,767 |

| Function Name | Special D | Description | Device | Type | Range |
|---|---|---|---|---|---|
| Torque Feedback of 17th Axis | D30256 | Torque feedback for the axis. The axis index is set according to the NC channel axis sequence setting.<br>PDO address 0x6077H<br>Unit: 0.1% | R | Decimal | -32,768 ~ +32,767 |
| Torque Feedback of 18th Axis | D30257 | | R | Decimal | -32,768 ~ +32,767 |
| Torque Feedback of 19th Axis | D30258 | | R | Decimal | -32,768 ~ +32,767 |
| Torque Feedback of 20th Axis | D30259 | | R | Decimal | -32,768 ~ +32,767 |
| Torque Feedback of 21st Axis | D30260 | | R | Decimal | -32,768 ~ +32,767 |
| Torque Feedback of 22nd Axis | D30261 | | R | Decimal | -32,768 ~ +32,767 |
| Torque Feedback of 23rd axis | D30262 | | R | Decimal | -32,768 ~ +32,767 |
| Torque Feedback of 24th Axis | D30263 | | R | Decimal | -32,768 ~ +32,767 |
| Torque Feedback of 25th Axis | D30264 | | R | Decimal | -32,768 ~ +32,767 |
| Torque Feedback of 26th Axis | D30265 | | R | Decimal | -32,768 ~ +32,767 |
| Torque Feedback of 27th Axis | D30266 | | R | Decimal | -32,768 ~ +32,767 |
| Torque Feedback of 28th Axis | D30267 | | R | Decimal | -32,768 ~ +32,767 |
| Torque Feedback of 29th Axis | D30268 | | R | Decimal | -32,768 ~ +32,767 |
| Torque Feedback of 30th Axis | D30269 | | R | Decimal | -32,768 ~ +32,767 |
| Torque Feedback of 31st Axis | D30270 | | R | Decimal | -32,768 ~ +32,767 |
| Torque Feedback of 32nd Axis | D30271 | | R | Decimal | -32,768 ~ +32,767 |
| Velocity Feedback of 1st Axis (mm/min) | D30272<br>D30273 | Axis speed feedback for the axis. The axis index is set according to the NC channel axis sequence setting.<br>PDO address 0x606CH<br>Unit: mm/min | R | Float | -2,147,483,648 ~ +2,147,483,647 |
| Velocity Feedback of 2nd Axis (mm/min) | D30274<br>D30275 | | R | Float | -2,147,483,648 ~ +2,147,483,647 |
| Velocity Feedback of 3rd Axis (mm/min) | D30276<br>D30277 | | R | Float | -2,147,483,648 ~ +2,147,483,647 |
| Velocity Feedback of 4th Axis (mm/min) | D30278<br>D30279 | | R | Float | -2,147,483,648 ~ +2,147,483,647 |
| Velocity Feedback of 5th Axis (mm/min) | D30280<br>D30281 | | R | Float | -2,147,483,648 ~ +2,147,483,647 |
| Velocity Feedback of 6th Axis (mm/min) | D30282<br>D30283 | | R | Float | -2,147,483,648 ~ +2,147,483,647 |
| Velocity Feedback of 7th Axis (mm/min) | D30284<br>D30285 | | R | Float | -2,147,483,648 ~ +2,147,483,647 |

5

| Function Name | Special D | Description | Device | Type | Range |
|---|---|---|---|---|---|
| Velocity Feedback of 8th Axis (mm/min) | D30286 D30287 | | R | Float | -2,147,483,648 ~ +2,147,483,647 |
| Velocity Feedback of 9th Axis (mm/min) | D30288 D30289 | | R | Float | -2,147,483,648 ~ +2,147,483,647 |
| Velocity Feedback of 10th Axis (mm/min) | D30290 D30291 | | R | Float | -2,147,483,648 ~ +2,147,483,647 |
| Velocity Feedback of 11th Axis (mm/min) | D30292 D30293 | | R | Float | -2,147,483,648 ~ +2,147,483,647 |
| Velocity Feedback of 12th Axis (mm/min) | D30294 D30295 | | R | Float | -2,147,483,648 ~ +2,147,483,647 |
| Velocity Feedback of 13th Axis (mm/min) | D30296 D30297 | | R | Float | -2,147,483,648 ~ +2,147,483,647 |
| Velocity Feedback of 14th Axis (mm/min) | D30298 D30299 | | R | Float | -2,147,483,648 ~ +2,147,483,647 |
| Velocity Feedback of 15th Axis (mm/min) | D30300 D30301 | | R | Float | -2,147,483,648 ~ +2,147,483,647 |
| Velocity Feedback of 16th Axis (mm/min) | D30302 D30303 | Axis speed feedback for the axis. The axis index is set according to the NC channel axis sequence setting.<br>PDO address 0x606CH<br>Unit: mm/min | R | Float | -2,147,483,648 ~ +2,147,483,647 |
| Velocity Feedback of 17th Axis (mm/min) | D30304 D30305 | | R | Float | -2,147,483,648 ~ +2,147,483,647 |
| Velocity Feedback of 18th Axis (mm/min) | D30306 D30307 | | R | Float | -2,147,483,648 ~ +2,147,483,647 |
| Velocity Feedback of 19th Axis (mm/min) | D30308 D30309 | | R | Float | -2,147,483,648 ~ +2,147,483,647 |
| Velocity Feedback of 20th Axis (mm/min) | D30310 D30311 | | R | Float | -2,147,483,648 ~ +2,147,483,647 |
| Velocity Feedback of 21st Axis (mm/min) | D30312 D30313 | | R | Float | -2,147,483,648 ~ +2,147,483,647 |
| Velocity Feedback of 22nd Axis (mm/min) | D30314 D30315 | | R | Float | -2,147,483,648 ~ +2,147,483,647 |
| Velocity Feedback of 23rd Axis (mm/min) | D30316 D30317 | | R | Float | -2,147,483,648 ~ +2,147,483,647 |
| Velocity Feedback of 24th Axis (mm/min) | D30318 D30319 | | R | Float | -2,147,483,648 ~ +2,147,483,647 |
| Velocity Feedback of 25th Axis (mm/min) | D30320 D30321 | | R | Float | -2,147,483,648 ~ +2,147,483,647 |

| Function Name | Special D | Description | Device | Type | Range |
|---|---|---|---|---|---|
| Velocity Feedback of 26th Axis (mm/min) | D30322<br>D30323 | Axis speed feedback for the axis. The axis index is set according to the NC channel axis sequence setting.<br>PDO address 0x606CH<br>Unit: mm/min | R | Float | -2,147,483,648<br>~<br>+2,147,483,647 |
| Velocity Feedback of 27th Axis (mm/min) | D30324<br>D30325 | | R | Float | -2,147,483,648<br>~<br>+2,147,483,647 |
| Velocity Feedback of 28th Axis (mm/min) | D30326<br>D30327 | | R | Float | -2,147,483,648<br>~<br>+2,147,483,647 |
| Velocity Feedback of 29th Axis (mm/min) | D30328<br>D30329 | | R | Float | -2,147,483,648<br>~<br>+2,147,483,647 |
| Velocity Feedback of 30th Axis (mm/min) | D30330<br>D30331 | | R | Float | -2,147,483,648<br>~<br>+2,147,483,647 |
| Velocity Feedback of 31st Axis (mm/min) | D30332<br>D30333 | | R | Float | -2,147,483,648<br>~<br>+2,147,483,647 |
| Velocity Feedback of 32nd Axis (mm/min) | D30334<br>D30335 | | R | Float | -2,147,483,648<br>~<br>+2,147,483,647 |
| Velocity Feedback of 1st Axis (RPM) | D30336<br>D30337 | Axis speed feedback for the axis. The axis index is set according to the NC channel axis sequence setting.<br>PDO address 0x606CH<br>Unit: RPM | R | Float | -2,147,483,648<br>~<br>+2,147,483,647 |
| Velocity Feedback of 2nd Axis (RPM) | D30338<br>D30339 | | R | Float | -2,147,483,648<br>~<br>+2,147,483,647 |
| Velocity Feedback of 3rd Axis (RPM) | D30340<br>D30341 | | R | Float | -2,147,483,648<br>~<br>+2,147,483,647 |
| Velocity Feedback of 4th Axis (RPM) | D30342<br>D30343 | | R | Float | -2,147,483,648<br>~<br>+2,147,483,647 |
| Velocity Feedback of 5th Axis (RPM) | D30344<br>D30345 | | R | Float | -2,147,483,648<br>~<br>+2,147,483,647 |
| Velocity Feedback of 6th Axis (RPM) | D30346<br>D30347 | | R | Float | -2,147,483,648<br>~<br>+2,147,483,647 |
| Velocity Feedback of 7th Axis (RPM) | D30348<br>D30349 | | R | Float | -2,147,483,648<br>~<br>+2,147,483,647 |
| Velocity Feedback of 8th Axis (RPM) | D30350<br>D30351 | | R | Float | -2,147,483,648<br>~<br>+2,147,483,647 |
| Velocity Feedback of 9th Axis (RPM) | D30352<br>D30353 | | R | Float | -2,147,483,648<br>~<br>+2,147,483,647 |
| Velocity Feedback of 10th Axis (RPM) | D30354<br>D30355 | | R | Float | -2,147,483,648<br>~<br>+2,147,483,647 |
| Velocity Feedback of 11th Axis (RPM) | D30356<br>D30357 | | R | Float | -2,147,483,648<br>~<br>+2,147,483,647 |

| Function Name | Special D | Description | Device | Type | Range |
|---|---|---|---|---|---|
| Velocity Feedback of 12th Axis (RPM) | D30358 D30359 | Axis speed feedback for the axis. The axis index is set according to the NC channel axis sequence setting.<br>PDO address 0x606CH<br>Unit: RPM | R | Float | -2,147,483,648 ~ +2,147,483,647 |
| Velocity Feedback of 13th Axis (RPM) | D30360 D30361 | | R | Float | -2,147,483,648 ~ +2,147,483,647 |
| Velocity Feedback of 14th Axis (RPM) | D30362 D30363 | | R | Float | -2,147,483,648 ~ +2,147,483,647 |
| Velocity Feedback of 15th Axis (RPM) | D30364 D30365 | | R | Float | -2,147,483,648 ~ +2,147,483,647 |
| Velocity Feedback of 16th Axis (RPM) | D30366 D30367 | | R | Float | -2,147,483,648 ~ +2,147,483,647 |
| Velocity Feedback of 17th Axis (RPM) | D30368 D30369 | | R | Float | -2,147,483,648 ~ +2,147,483,647 |
| Velocity Feedback of 18th Axis (RPM) | D30360 D30371 | | R | Float | -2,147,483,648 ~ +2,147,483,647 |
| Velocity Feedback of 19th Axis (RPM) | D30372 D30373 | | R | Float | -2,147,483,648 ~ +2,147,483,647 |
| Velocity Feedback of 20th Axis (RPM) | D30374 D30375 | | R | Float | -2,147,483,648 ~ +2,147,483,647 |
| Velocity Feedback of 21st Axis (RPM) | D30376 D30377 | | R | Float | -2,147,483,648 ~ +2,147,483,647 |
| Velocity Feedback of 22nd Axis (RPM) | D30378 D30379 | | R | Float | -2,147,483,648 ~ +2,147,483,647 |
| Velocity Feedback of 23rd Axis (RPM) | D30380 D30381 | | R | Float | -2,147,483,648 ~ +2,147,483,647 |
| Velocity Feedback of 24th Axis (RPM) | D30382 D30383 | | R | Float | -2,147,483,648 ~ +2,147,483,647 |
| Velocity Feedback of 25th Axis (RPM) | D30384 D30385 | | R | Float | -2,147,483,648 ~ +2,147,483,647 |
| Velocity Feedback of 26th Axis (RPM) | D30386 D30387 | | R | Float | -2,147,483,648 ~ +2,147,483,647 |
| Velocity Feedback of 27th Axis (RPM) | D30388 D30389 | | R | Float | -2,147,483,648 ~ +2,147,483,647 |
| Velocity Feedback of 28th Axis (RPM) | D30390 D30391 | | R | Float | -2,147,483,648 ~ +2,147,483,647 |
| Velocity Feedback of 29th Axis (RPM) | D30392 D30393 | | R | Float | -2,147,483,648 ~ +2,147,483,647 |

| Function Name | Special D | Description | Device | Type | Range |
|---|---|---|---|---|---|
| Velocity Feedback of 30th Axis (RPM) | D30394 D30395 | Axis speed feedback for the axis. The axis index is set according to the NC channel axis sequence setting.<br>PDO address 0x606CH<br>Unit: RPM | R | Float | -2,147,483,648 ~ +2,147,483,647 |
| Velocity Feedback of 31st Axis (RPM) | D30396 D30397 | | R | Float | -2,147,483,648 ~ +2,147,483,647 |
| Velocity Feedback of 32nd Axis (RPM) | D30398 D30399 | | R | Float | -2,147,483,648 ~ +2,147,483,647 |
| Torque Peak of 1st Axis | D30400 | The system will monitor and record the maximum axis torque shown in **D30240** during Servo ON. Once the axis enters Servo OFF, these special D will be cleared as well. Unit: 0.1%. | R | Decimal | 0 ~ 4,294,967,295 |
| Torque Peak of 2nd Axis | D30401 | The system will monitor and record the maximum axis torque shown in **D30241** during Servo ON. Once the axis enters Servo OFF, these special D will be cleared as well. Unit: 0.1%. | R | Decimal | 0 ~ 4,294,967,295 |
| Torque Peak of 3rd Axis | D30402 | The system will monitor and record the maximum axis torque shown in **D30242** during Servo ON. Once the axis enters Servo OFF, these special D will be cleared as well. Unit: 0.1%. | R | Decimal | 0 ~ 4,294,967,295 |
| Torque Peak of 4th Axis | D30403 | The system will monitor and record the maximum axis torque shown in **D30243** during Servo ON. Once the axis enters Servo OFF, these special D will be cleared as well. Unit: 0.1%. | R | Decimal | 0 ~ 4,294,967,295 |
| Torque Peak of 5th Axis | D30404 | The system will monitor and record the maximum axis torque shown in **D30244** during Servo ON. Once the axis enters Servo OFF, these special D will be cleared as well. Unit: 0.1%. | R | Decimal | 0 ~ 4,294,967,295 |
| Torque Peak of 6th Axis | D30405 | The system will monitor and record the maximum axis torque shown in **D30245** during Servo ON. Once the axis enters Servo OFF, these special D will be cleared as well. Unit: 0.1%. | R | Decimal | 0 ~ 4,294,967,295 |
| Torque Peak of 7th Axis | D30406 | The system will monitor and record the maximum axis torque shown in **D30246** during Servo ON. Once the axis enters Servo OFF, these special D will be cleared as well. Unit: 0.1% | R | Decimal | 0 ~ 4,294,967,295 |
| Torque Peak of 8th Axis | D30407 | The system will monitor and record the maximum axis torque shown in **D30247** during Servo ON. Once the axis enters Servo OFF, these special D will be cleared as well. Unit: 0.1%. | R | Decimal | 0 ~ 4,294,967,295 |
| Torque Peak of 9th Axis | D30408 | The system will monitor and record the maximum axis torque shown in **D30248** during Servo ON. Once the axis enters Servo OFF, these special D will be cleared as well. Unit: 0.1%. | R | Decimal | 0 ~ 4,294,967,295 |
| Torque Peak of 10th Axis | D30409 | The system will monitor and record the maximum axis torque shown in **D30249** during Servo ON. Once the axis enters Servo OFF, these special D will be cleared as well. Unit: 0.1%. | R | Decimal | 0 ~ 4,294,967,295 |

| Function Name | Special D | Description | Device | Type | Range |
|---|---|---|---|---|---|
| Torque Peak of 11th Axis | D30410 | The system will monitor and record the maximum axis torque shown in **D30250** during Servo ON. Once the axis enters Servo OFF, these special D will be cleared as well. Unit: 0.1%. | R | Decimal | 0 ~ 4,294,967,295 |
| Torque Peak of 12th Axis | D30411 | The system will monitor and record the maximum axis torque shown in **D30251** during Servo ON. Once the axis enters Servo OFF, these special D will be cleared as well. Unit: 0.1%. | R | Decimal | 0 ~ 4,294,967,295 |
| Torque Peak of 13th Axis | D30412 | The system will monitor and record the maximum axis torque shown in **D30252** during Servo ON. Once the axis enters Servo OFF, these special D will be cleared as well. Unit: 0.1%. | R | Decimal | 0 ~ 4,294,967,295 |
| Torque Peak of 14th Axis | D30413 | The system will monitor and record the maximum axis torque shown in **D30253** during Servo ON. Once the axis enters Servo OFF, these special D will be cleared as well. Unit: 0.1%. | R | Decimal | 0 ~ 4,294,967,295 |
| Torque Peak of 15th Axis | D30414 | The system will monitor and record the maximum axis torque shown in **D30254** during Servo ON. Once the axis enters Servo OFF, these special D will be cleared as well. Unit: 0.1%. | R | Decimal | 0 ~ 4,294,967,295 |
| Torque Peak of 16th Axis | D30415 | The system will monitor and record the maximum axis torque shown in **D30255** during Servo ON. Once the axis enters Servo OFF, these special D will be cleared as well. Unit: 0.1%. | R | Decimal | 0 ~ 4,294,967,295 |
| Torque Peak of 17th Axis | D30416 | The system will monitor and record the maximum axis torque shown in **D30256** during Servo ON. Once the axis enters Servo OFF, these special D will be cleared as well. Unit: 0.1%. | R | Decimal | 0 ~ 4,294,967,295 |
| Torque Peak of 18th Axis | D30417 | The system will monitor and record the maximum axis torque shown in **D30257** during Servo ON. Once the axis enters Servo OFF, these special D will be cleared as well. Unit: 0.1%. | R | Decimal | 0 ~ 4,294,967,295 |
| Torque Peak of 19th Axis | D30418 | The system will monitor and record the maximum axis torque shown in **D30258** during Servo ON. Once the axis enters Servo OFF, these special D will be cleared as well. Unit: 0.1%. | R | Decimal | 0 ~ 4,294,967,295 |
| Torque Peak of 20th Axis | D30419 | The system will monitor and record the maximum axis torque shown in **D30259** during Servo ON. Once the axis enters Servo OFF, these special D will be cleared as well. Unit: 0.1%. | R | Decimal | 0 ~ 4,294,967,295 |
| Torque Peak of 21st Axis | D30420 | The system will monitor and record the maximum axis torque shown in **D30260** during Servo ON. Once the axis enters Servo OFF, these special D will be cleared as well. Unit: | R | Decimal | 0 ~ 4,294,967,295 |
| Torque Peak of 22nd Axis | D30421 | The system will monitor and record the maximum axis torque shown in **D30261** during Servo ON. Once the axis enters Servo OFF, these special D will be cleared as well. Unit: 0.1%. | R | Decimal | 0 ~ 4,294,967,295 |

| Function Name | Special D | Description | Device | Type | Range |
|---|---|---|---|---|---|
| Torque Peak of 23rd Axis | D30422 | The system will monitor and record the maximum axis torque shown in **D30262** during Servo ON. Once the axis enters Servo OFF, these special D will be cleared as well. Unit: 0.1%. | R | Decimal | 0 ~ 4,294,967,295 |
| Torque Peak of 24th Axis | D30423 | The system will monitor and record the maximum axis torque shown in **D30263** during Servo ON. Once the axis enters Servo OFF, these special D will be cleared as well. Unit: 0.1%. | R | Decimal | 0 ~ 4,294,967,295 |
| Torque Peak of 25th Axis | D30424 | The system will monitor and record the maximum axis torque shown in **D30264** during Servo ON. Once the axis enters Servo OFF, these special D will be cleared as well. Unit: 0.1%. | R | Decimal | 0 ~ 4,294,967,295 |
| Torque Peak of 26th Axis | D30425 | The system will monitor and record the maximum axis torque shown in **D30265** during Servo ON. Once the axis enters Servo OFF, these special D will be cleared as well. Unit: 0.1%. | R | Decimal | 0 ~ 4,294,967,295 |
| Torque Peak of 27th Axis | D30426 | The system will monitor and record the maximum axis torque shown in **D30266** during Servo ON. Once the axis enters Servo OFF, these special D will be cleared as well. Unit: 0.1%. | R | Decimal | 0 ~ 4,294,967,295 |
| Torque Peak of 28th Axis | D30427 | The system will monitor and record the maximum axis torque shown in **D30267** during Servo ON. Once the axis enters Servo OFF, these special D will be cleared as well. Unit: 0.1%. | R | Decimal | 0 ~ 4,294,967,295 |
| Torque Peak of 29th Axis | D30428 | The system will monitor and record the maximum axis torque shown in **D30268** during Servo ON. Once the axis enters Servo OFF, these special D will be cleared as well. Unit: 0.1%. | R | Decimal | 0 ~ 4,294,967,295 |
| Torque Peak of 30th Axis | D30429 | The system will monitor and record the maximum axis torque shown in **D30269** during Servo ON. Once the axis enters Servo OFF, these special D will be cleared as well. Unit: 0.1%. | R | Decimal | 0 ~ 4,294,967,295 |
| Torque Peak of 31st Axis | D30430 | The system will monitor and record the maximum axis torque shown in **D30270** during Servo ON. Once the axis enters Servo OFF, these special D will be cleared as well. Unit: 0.1%. | R | Decimal | 0 ~ 4,294,967,295 |
| Torque Peak of 32nd Axis | D30431 | The system will monitor and record the maximum axis torque shown in **D30271** during Servo ON. Once the axis enters Servo OFF, these special D will be cleared as well. Unit: 0.1%. | R | Decimal | 0 ~ 4,294,967,295 |
| 1st Spindle Torque Feedback | D30432 | Display the spindle current torque feedback according to the spindle ID setting in the channel.<br>Unit: 0.1% | R | Decimal | -32,768 ~ +32,767 |
| 2nd Spindle Torque Feedback | D30433 | | R | Decimal | -32,768 ~ +32,767 |
| 3rd Spindle Torque Feedback | D30434 | | R | Decimal | -32,768 ~ +32,767 |
| 4th Spindle Torque Feedback | D30435 | | R | Decimal | -32,768 ~ +32,767 |

5

| Function Name | Special D | Description | Device | Type | Range |
|---|---|---|---|---|---|
| 5th Spindle Torque Feedback | D30436 | Display the spindle current torque feedback according to the spindle ID setting in the channel. Unit: 0.1% | R | Decimal | -32,768 ~ +32,767 |
| 6th Spindle Torque Feedback | D30437 | | R | Decimal | -32,768 ~ +32,767 |
| 7th Spindle Torque Feedback | D30438 | | R | Decimal | -32,768 ~ +32,767 |
| 8th Spindle Torque Feedback | D30439 | | R | Decimal | -32,768 ~ +32,767 |
| 1st Spindle Torque Feedback Peak | D30440 | The NC system will record the maximum torque feedback which showing in the **D30432** when the spindle is enabled. Value cleared when the spindle is disabled. Unit: 0.1% | R | Decimal | -32,768 ~ +32,767 |
| 2nd Spindle Torque Feedback Peak | D30441 | The NC system will record the maximum torque feedback which showing in the **D30433** when the spindle is enabled. Value cleared when the spindle is disabled. Unit: 0.1% | R | Decimal | -32,768 ~ +32,767 |
| 3rd Spindle Torque Feedback Peak | D30442 | The NC system will record the maximum torque feedback which showing in the **D30434** when the spindle is enabled. Value cleared when the spindle is disabled. Unit: 0.1% | R | Decimal | -32,768 ~ +32,767 |
| 4th Spindle Torque Feedback Peak | D30443 | The NC system will record the maximum torque feedback which showing in the **D30435** when the spindle is enabled. Value cleared when the spindle is disabled. Unit: 0.1% | R | Decimal | -32,768 ~ +32,767 |
| 5th Spindle Torque Feedback Peak | D30444 | The NC system will record the maximum torque feedback which showing in the **D30436** when the spindle is enabled. Value cleared when the spindle is disabled. Unit: 0.1% | R | Decimal | -32,768 ~ +32,767 |
| 6th Spindle Torque Feedback Peak | D30445 | The NC system will record the maximum torque feedback which showing in the **D30437** when the spindle is enabled. Value cleared when the spindle is disabled. Unit: 0.1% | R | Decimal | -32,768 ~ +32,767 |
| 7th Spindle Torque Feedback Peak | D30446 | The NC system will record the maximum torque feedback which showing in the **D30438** when the spindle is enabled. Value cleared when the spindle is disabled. Unit: 0.1% | R | Decimal | -32,768 ~ +32,767 |
| 8th Spindle Torque Feedback Peak | D30447 | The NC system will record the maximum torque feedback which showing in the **D30439** when the spindle is enabled. Value cleared when the spindle is disabled. Unit: 0.1% | R | Decimal | -32,768 ~ +32,767 |

| Function Name | Special D | Description | Device | Type | Range |
|---|---|---|---|---|---|
| Current Multi Head Status of Z axis | D3x014 | This special D will show the Z axes index currently in use when the Multi head function is enabled. When the value is 12, it means Z1 and Z2 are working. When the value is 123, it means Z1, Z2, and Z3 are working. | R | Decimal | 0 ~ 4,294,967,295 |
| Current Coordinate System | D3x016 | Shows the coordinate system the system is currently using. This coordinate system can be different according to the channel machine type setting such as standard type and Robot.<br>1. When the channel set as standard machine type such as milling or lathe, the system will show as below working coordinate system. 1~6: corresponding to G54~G59. 7~262: corresponding to G54 P1~G54 P256.<br>2. When the channel set as Robot machine type, the system will show as below working coordinate system. 1~6: corresponding to G54~G59. | R | Decimal | Standard<br>1 ~ 262<br><br>Robot<br>1~6 |
| Current Robot Tool Coordinate System | D3x017 | Shows the robot tool coordinate system.<br>0: Not using tool offset<br>1 ~ n: corresponds to tool number | R | Decimal | 0 ~ 65,535 |
| Speed Command of 1st Spindle | D3x024 | When the 1st spindle S code is executed in a program, the command value will be sent in this special D.<br>Unit: RPM | R | Float | -2,147,483,648 ~ +2,147,483,647 |
| 1st Spindle Speed Feedback | D3x026 | Shows the 1st spindle's speed. The data source is from the spindle's command speed. | R | Float | -2,147,483,648 ~ +2,147,483,647 |
| 1st Spindle Actual Degree | D3x028 | Shows the 1st spindle's actual degree. The data source is from the spindle's actual degree. | R | Float | -2,147,483,648 ~ +2,147,483,647 |
| Speed Command of 2nd Spindle | D3x030 | When the 2nd spindle S code is executed in a program, the command value will be sent in this special D.<br>Unit: RPM | R | Float | -2,147,483,648 ~ +2,147,483,647 |
| 2nd Spindle Speed Feedback | D3x032 | Shows the 2nd spindle's speed. The data source is from the spindle's command speed. | R | Float | -2,147,483,648 ~ +2,147,483,647 |
| 2nd Spindle Actual Degree | D3x034 | Shows the 2nd spindle's actual degree. The data source is from the spindle's actual degree. | R | Float | -2,147,483,648 ~ +2,147,483,647 |
| Current Tool Number Tool Magazine 1 | D3x036 | Current tool number in the tool magazine 1. | R | Decimal | 0 ~ 65,535 |
| Standby Tool Number Tool Magazine 1 | D3x037 | Current standby tool number in the tool magazine 1. (The latest T code) | R | Decimal | 0 ~ 65,535 |
| Standby Tool Pot Tool Magazine 1 | D3x038 | Current standby tool pot number in the tool magazine 1. | R | Decimal | 0 ~ 65,535 |

5

| Function Name | Special D | Description | Device | Type | Range |
|---|---|---|---|---|---|
| Tool Pot Deviation Tool Magazine 1 | D3x039 | The deviation between the positions specified for the current tool and command tool in tool magazine 1.<br>Users can determine the rotation direction by whether this value is positive or negative. When the tool magazine rotates forward (**M2x064**) or backward (**M2x065**) during tool exchange, the tool magazine needs to rotate according to the value to compensate for the offset. | R | Decimal | 0 ~ 65,535 |
| Current Tool Number Tool Magazine 2 | D3x042 | Current tool number in the tool magazine 2. | R | Decimal | 0 ~ 65,535 |
| Standby Tool Number Tool Magazine 2 | D3x043 | Current standby tool number in the tool magazine 2. (The latest T code) | R | Decimal | 0 ~ 65,535 |
| Standby Tool Pot Tool Magazine 2 | D3x044 | Current standby tool pot number in the tool magazine 2. | R | Decimal | 0 ~ 65,535 |
| Tool Pot Deviation Tool Magazine 2 | D3x045 | The deviation between the positions specified for the current tool and command tool in tool magazine 2.<br>Users can determine the rotation direction by whether this value is positive or negative. When the tool magazine rotates forward (**M2x072**) or backward (**M2x073**) during tool exchange, the tool magazine needs to rotate according to the value to compensate for the offset. | R | Decimal | 0 ~ 65,535 |
| 1st M Code Data | D3x048 | When the 1st M code is executed in the program (Not including M00, M01, M02, M30, M98, M99), the M code value will be mapped to this register. When the M code is used to call the macro, this special D will keep the previous value. | R | Decimal | 0 ~ 4,294,967,295 |
| 1st S Code Data | D3x050 | When the 1st spindle's 1st S code is executed in the program, the 1st spindle's S code will be mapped to this register. Unit: RPM. | R | Decimal | 0 ~ 4,294,967,295 |
| 1st T Code Data | D3x052 | When the 1st T code is executed in the program, the T code will be mapped to this register. When the 1st T code is used to call the macro, this special D will keep the previous value.<br>This data is related to the tool pot setting of the tool magazine, and the T code will be shown only when the T code value is set within the specified range of tool number for the tool magazine parameter. | R | Decimal | 0 ~ 4,294,967,295 |
| Information Monitoring 1 Sort 1 | D3x096<br>D3x097 | This special D is for users to monitor the information of system devices according to the variable settings. In order to use this special D, users need to set up [N1.321] for information type and [N1.322] for data sort 1 in advance. *Please check section 6.17 for more details. | R | Decimal | 0 ~ 65,535 |

| Function Name | Special D | Description | Device | Type | Range |
|---|---|---|---|---|---|
| Information Monitoring 1 Sort 2 | D3x098 D3x099 | This special D is for users to monitor the information of system devices according to the variable settings. In order to use this special D, users need to set up [N1.321] for information type and [N1.323] for data sort 2 in advance. *Please check section 6.17 for more details. | R | Decimal | 0 ~ 65,535 |
| Information Monitoring 1 Sort 3 | D3x100 D3x101 | This special D is for users to monitor the information of system devices according to the variable settings. In order to use this special D, users need to set up [N1.321] for information type and [N1.324] for data sort 3 in advance. *Please check section 6.17 for more details. | R | Decimal | 0 ~ 65,535 |
| Information Monitoring 1 Sort 4 | D3x102 D3x103 | This special D is for users to monitor the information of system devices according to the variable settings. In order to use this special D, users need to set up [N1.321] for information type and [N1.325] for data sort 4 in advance. *Please check section 6.17 for more details. | R | Decimal | 0 ~ 65,535 |
| Information Monitoring 2 Sort 1 | D3x104 D3x105 | This special D is for users to monitor the information of system devices according to the variable settings. In order to use this special D, users need to set up [N1.326] for information type and [N1.327] for data sort 1 in advance. *Please check section 6.17 for more details. | R | Decimal | 0 ~ 65,535 |
| Information Monitoring 2 Sort 2 | D3x106 D3x107 | This special D is for users to monitor the information of system devices according to the variable settings. In order to use this special D, users need to set up [N1.326] for information type and [N1.328] for data sort 2 in advance. *Please check section 6.17 for more details. | R | Decimal | 0 ~ 65,535 |
| Information Monitoring 2 Sort 3 | D3x108 D3x109 | This special D is for users to monitor the information of system devices according to the variable settings. In order to use this special D, users need to set up [N1.326] for information type and [N1.329] for data sort 3 in advance. *Please check section 6.17 for more details. | R | Decimal | 0 ~ 65,535 |
| Information Monitoring 2 Sort 4 | D3x110 D3x111 | This special D is for users to monitor the information of system devices according to the variable settings. In order to use this special D, users need to set up [N1.326] for information type and [N1.330] for data sort 4 in advance. *Please check section 6.17 for more details. | R | Decimal | 0 ~ 65,535 |

5

| Function Name | Special D | Description | Device | Type | Range |
|---|---|---|---|---|---|
| NC Variable to MLC 1 | D3x128 | The system will move data from NC variable #25384 to this special D. | R | Decimal / Float | (Default) -32,768 ~ +32,767 / (N1.010 Bit7=1) -2,147,483,648 ~ +2,147,483,647 |
| NC Variable to MLC 2 | D3x129 | The system will move data from NC variable #25385 to this special D. *Both this special D and the variable # are NOT available to use when using MLC floating mode **[N1.010 Bit7=1]**. Otherwise, the system will return error 0x235. | R | | |
| NC Variable to MLC 3 | D3x130 | The system will move data from NC variable #25386 to this special D. | R | | |
| NC Variable to MLC 4 | D3x131 | The system will move data from NC variable #25387 to this special D. *Both this special D and the variable # are NOT available to use when using MLC floating mode **[N1.010 Bit7=1]**. Otherwise, the system will return error 0x235. | R | | |
| NC Variable to MLC 5 | D3x132 | The system will move data from NC variable #25388 to this special D. | R | | |
| NC Variable to MLC 6 | D3x133 | The system will move data from NC variable #25389 to this special D. *Both this special D and the variable # are NOT available to use when using MLC floating mode **[N1.010 Bit7=1]**. Otherwise, the system will return error 0x235. | R | | |
| NC Variable to MLC 7 | D3x134 | The system will move data from NC variable #25390 to this special D. | R | | |
| NC Variable to MLC 8 | D3x135 | The system will move data from NC variable #25391 to this special D. *Both this special D and the variable # are NOT available to use when using MLC floating mode **[N1.010 Bit7=1]**. Otherwise, the system will return error 0x235. | R | | |
| NC Variable to MLC 9 | D3x136 | The system will move data from NC variable #25392 to this special D. | R | | |
| NC Variable to MLC 10 | D3x137 | The system will move data from NC variable #25393 to this special D. *Both this special D and the variable # are NOT available to use when using MLC floating mode **[N1.010 Bit7=1]**. Otherwise, the system will return error 0x235. | R | | |
| NC Variable to MLC 11 | D3x138 | The system will move data from NC variable #25394 to this special D. | R | | |
| NC Variable to MLC 12 | D3x139 | The system will move data from NC variable #25395 to this special D. *Both this special D and the variable # are NOT available to use when using MLC floating mode **[N1.010 Bit7=1]**. Otherwise, the system will return error 0x235. | R | | |
| NC Variable to MLC 13 | D3x140 | The system will move data from NC variable #25396 to this special D. | R | | |

| Function Name | Special D | Description | Device | Type | Range |
|---|---|---|---|---|---|
| NC Variable to MLC 14 | D3x141 | The system will move data from NC variable #25397 to this special D. *Both this special D and the variable # are NOT available to use when using MLC floating mode **[N1.010 Bit7=1]**. Otherwise, the system will return error 0x235. | R | Decimal / Float | (Default) -32,768 ~ +32,767 / (N1.010 Bit7=1) -2,147,483,648 ~ +2,147,483,647 |
| NC Variable to MLC 15 | D3x142 | The system will move data from NC variable #25398 to this special D. | R | | |
| NC Variable to MLC 16 | D3x143 | The system will move data from NC variable #25399 to this special D. *Both this special D and the variable # are NOT available to use when using MLC floating mode **[N1.010 Bit7=1]**. Otherwise, the system will return error 0x235. | R | | |
| NC Variable to MLC 17 | D3x144 | The system will move data from NC variable #25400 to this special D. | R | | |
| NC Variable to MLC 18 | D3x145 | The system will move data from NC variable #25401 to this special D. *Both this special D and the variable # are NOT available to use when using MLC floating mode **[N1.010 Bit7=1]**. Otherwise, the system will return error 0x235. | R | | |
| NC Variable to MLC 19 | D3x146 | The system will move data from NC variable #25402 to this special D. | R | | |
| NC Variable to MLC 20 | D3x147 | The system will move data from NC variable #25403 to this special D. *Both this special D and the variable # are NOT available to use when using MLC floating mode **[N1.010 Bit7=1]**. Otherwise, the system will return error 0x235. | R | | |
| NC Variable to MLC 21 | D3x148 | The system will move data from NC variable #25404 to this special D. | R | | |
| NC Variable to MLC 22 | D3x149 | The system will move data from NC variable #25405 to this special D. *Both this special D and the variable # are NOT available to use when using MLC floating mode **[N1.010 Bit7=1]**. Otherwise, the system will return error 0x235. | R | | |
| NC Variable to MLC 23 | D3x150 | The system will move data from NC variable #25406 to this special D. | R | | |
| NC Variable to MLC 24 | D3x151 | The system will move data from NC variable #25407 to this special D. *Both this special D and the variable # are NOT available to use when using MLC floating mode **[N1.010 Bit7=1]**. Otherwise, the system will return error 0x235. | R | | |
| NC Variable to MLC 25 | D3x152 | The system will move data from NC variable #25408 to this special D. | R | | |

5

| Function Name | Special D | Description | Device | Type | Range |
|---|---|---|---|---|---|
| NC Variable to MLC 26 | D3x153 | The system will move data from NC variable #25409 to this special D.<br>*Both this special D and the variable # are NOT available to use when using MLC floating mode **[N1.010 Bit7=1]**. Otherwise, the system will return error 0x235. | R | Decimal / Float | (Default) -32,768 ~ +32,767 / (N1.010 Bit7=1) -2,147,483,648 ~ +2,147,483,647 |
| NC Variable to MLC 27 | D3x154 | The system will move data from NC variable #25410 to this special D. | R | | |
| NC Variable to MLC 28 | D3x155 | The system will move data from NC variable #25411 to this special D.<br>*Both this special D and the variable # are NOT available to use when using MLC floating mode **[N1.010 Bit7=1]**. Otherwise, the system will return error 0x235. | R | | |
| NC Variable to MLC 29 | D3x156 | The system will move data from NC variable #25412 to this special D. | R | | |
| NC Variable to MLC 30 | D3x157 | The system will move data from NC variable #25413 to this special D.<br>*Both this special D and the variable # are NOT available to use when using MLC floating mode **[N1.010 Bit7=1]**. Otherwise, the system will return error 0x235. | R | | |
| NC Variable to MLC 31 | D3x158 | The system will move data from NC variable #25414 to this special D. | R | | |
| NC Variable to MLC 32 | D3x159 | The system will move data from NC variable #25415 to this special D.<br>*Both this special D and the variable # are NOT available to use when using MLC floating mode **[N1.010 Bit7=1]**. Otherwise, the system will return error 0x235. | R | | |
| NC Variable to MLC 33 | D3x160 | The system will move data from NC variable #25416 to this special D. | R | | |
| NC Variable to MLC 34 | D3x161 | The system will move data from NC variable #25417 to this special D.<br>*Both this special D and the variable # are NOT available to use when using MLC floating mode **[N1.010 Bit7=1]**. Otherwise, the system will return error 0x235. | R | | |
| NC Variable to MLC 35 | D3x162 | The system will move data from NC variable #25418 to this special D. | R | | |
| NC Variable to MLC 36 | D3x163 | The system will move data from NC variable #25419 to this special D.<br>*Both this special D and the variable # are NOT available to use when using MLC floating mode **[N1.010 Bit7=1]**. Otherwise, the system will return error 0x235. | R | | |
| NC Variable to MLC 37 | D3x164 | The system will move data from NC variable #25420 to this special D. | R | | |

| Function Name | Special D | Description | Device | Type | Range |
|---|---|---|---|---|---|
| NC Variable to MLC 38 | D3x165 | The system will move data from NC variable #25421 to this special D.<br>*Both this special D and the variable # are NOT available to use when using MLC floating mode **[N1.010 Bit7=1]**. Otherwise, the system will return error 0x235. | R | Decimal / Float | (Default) -32,768 ~ +32,767 / (N1.010 Bit7=1) -2,147,483,648 ~ +2,147,483,647 |
| NC Variable to MLC 39 | D3x166 | The system will move data from NC variable #25422 to this special D. | R | | |
| NC Variable to MLC 40 | D3x167 | The system will move data from NC variable #25423 to this special D.<br>*Both this special D and the variable # are NOT available to use when using MLC floating mode **[N1.010 Bit7=1]**. Otherwise, the system will return error 0x235. | R | | |
| NC Variable to MLC 41 | D3x168 | The system will move data from NC variable #25424 to this special D. | R | | |
| NC Variable to MLC 42 | D3x169 | The system will move data from NC variable #25425 to this special D.<br>*Both this special D and the variable # are NOT available to use when using MLC floating mode **[N1.010 Bit7=1]**. Otherwise, the system will return error 0x235. | R | | |
| NC Variable to MLC 43 | D3x170 | The system will move data from NC variable #25426 to this special D. | R | | |
| NC Variable to MLC 44 | D3x171 | The system will move data from NC variable #25427 to this special D.<br>*Both this special D and the variable # are NOT available to use when using MLC floating mode **[N1.010 Bit7=1]**. Otherwise, the system will return error 0x235. | R | | |
| NC Variable to MLC 45 | D3x172 | The system will move data from NC variable #25428 to this special D. | R | | |
| NC Variable to MLC 46 | D3x173 | The system will move data from NC variable #25429 to this special D.<br>*Both this special D and the variable # are NOT available to use when using MLC floating mode **[N1.010 Bit7=1]**. Otherwise, the system will return error 0x235. | R | | |
| NC Variable to MLC 47 | D3x174 | The system will move data from NC variable #25430 to this special D. | R | | |
| NC Variable to MLC 48 | D3x175 | The system will move data from NC variable #25431 to this special D.<br>*Both this special D and the variable # are NOT available to use when using MLC floating mode **[N1.010 Bit7=1]**. Otherwise, the system will return error 0x235. | R | | |
| NC Variable to MLC 49 | D3x176 | The system will move data from NC variable #25432 to this special D. | R | | |

5

| Function Name | Special D | Description | Device | Type | Range |
|---|---|---|---|---|---|
| NC Variable to MLC 50 | D3x177 | The system will move data from NC variable #25433 to this special D.<br>*Both this special D and the variable # are NOT available to use when using MLC floating mode **[N1.010 Bit7=1]**. Otherwise, the system will return error 0x235. | R | Decimal / Float | (Default) -32,768 ~ +32,767 / (N1.010 Bit7=1) -2,147,483,648 ~ +2,147,483,647 |
| NC Variable to MLC 51 | D3x178 | The system will move data from NC variable #25434 to this special D. | R | | |
| NC Variable to MLC 52 | D3x179 | The system will move data from NC variable #25435 to this special D.<br>*Both this special D and the variable # are NOT available to use when using MLC floating mode **[N1.010 Bit7=1]**. Otherwise, the system will return error 0x235. | R | | |
| NC Variable to MLC 53 | D3x180 | The system will move data from NC variable #25436 to this special D. | R | | |
| NC Variable to MLC 54 | D3x181 | The system will move data from NC variable #25437 to this special D.<br>*Both this special D and the variable # are NOT available to use when using MLC floating mode **[N1.010 Bit7=1]**. Otherwise, the system will return error 0x235. | R | | |
| NC Variable to MLC 55 | D3x182 | The system will move data from NC variable #25438 to this special D. | R | | |
| NC Variable to MLC 56 | D3x183 | The system will move data from NC variable #25439 to this special D.<br>*Both this special D and the variable # are NOT available to use when using MLC floating mode **[N1.010 Bit7=1]**. Otherwise, the system will return error 0x235. | R | | |
| NC Variable to MLC 57 | D3x184 | The system will move data from NC variable #25440 to this special D. | R | | |
| NC Variable to MLC 58 | D3x185 | The system will move data from NC variable #25441 to this special D.<br>*Both this special D and the variable # are NOT available to use when using MLC floating mode **[N1.010 Bit7=1]**. Otherwise, the system will return error 0x235. | R | | |
| NC Variable to MLC 59 | D3x186 | The system will move data from NC variable #25442 to this special D. | R | | |
| NC Variable to MLC 60 | D3x187 | The system will move data from NC variable #25443 to this special D.<br>*Both this special D and the variable # are NOT available to use when using MLC floating mode **[N1.010 Bit7=1]**. Otherwise, the system will return error 0x235. | R | | |
| NC Variable to MLC 61 | D3x188 | The system will move data from NC variable #25444 to this special D. | R | | |

| Function Name | Special D | Description | Device | Type | Range |
|---|---|---|---|---|---|
| NC Variable to MLC 62 | D3x189 | The system will move data from NC variable #25445 to this special D. *Both this special D and the variable # are NOT available to use when using MLC floating mode **[N1.010 Bit7=1]**. Otherwise, the system will return error 0x235. | R | Decimal / Float | (Default) -32,768 ~ +32,767 / (N1.010 Bit7=1) -2,147,483,648 ~ +2,147,483,647 |
| NC Variable to MLC 63 | D3x190 | The system will move data from NC variable #25446 to this special D. | R | | |
| NC Variable to MLC 64 | D3x191 | The system will move data from NC variable #25447 to this special D. *Both this special D and the variable # are NOT available to use when using MLC floating mode **[N1.010 Bit7=1]**. Otherwise, the system will return error 0x235. | R | | |
| NC Variable to MLC 65 | D3x192 | The system will move data from NC variable #25448 to this special D. | R | | |
| NC Variable to MLC 66 | D3x193 | The system will move data from NC variable #25449 to this special D. *Both this special D and the variable # are NOT available to use when using MLC floating mode **[N1.010 Bit7=1]**. Otherwise, the system will return error 0x235. | R | | |
| NC Variable to MLC 67 | D3x194 | The system will move data from NC variable #25450 to this special D. | R | | |
| NC Variable to MLC 68 | D3x195 | The system will move data from NC variable #25451 to this special D. *Both this special D and the variable # are NOT available to use when using MLC floating mode **[N1.010 Bit7=1]**. Otherwise, the system will return error 0x235. | R | | |
| NC Variable to MLC 69 | D3x196 | The system will move data from NC variable #25452 to this special D. | R | | |
| NC Variable to MLC 70 | D3x197 | The system will move data from NC variable #25453 to this special D. *Both this special D and the variable # are NOT available to use when using MLC floating mode **[N1.010 Bit7=1]**. Otherwise, the system will return error 0x235. | R | | |
| NC Variable to MLC 71 | D3x198 | The system will move data from NC variable #25454 to this special D. | R | | |
| NC Variable to MLC 72 | D3x199 | The system will move data from NC variable #25455 to this special D. *Both this special D and the variable # are NOT available to use when using MLC floating mode **[N1.010 Bit7=1]**. Otherwise, the system will return error 0x235. | R | | |
| NC Variable to MLC 73 | D3x200 | The system will move data from NC variable #25456 to this special D. | R | | |

5

| Function Name | Special D | Description | Device | Type | Range |
|---|---|---|---|---|---|
| NC Variable to MLC 74 | D3x201 | The system will move data from NC variable #25457 to this special D. *Both this special D and the variable # are NOT available to use when using MLC floating mode **[N1.010 Bit7=1]**. Otherwise, the system will return error 0x235. | R | Decimal / Float | (Default) -32,768 ~ +32,767 / (N1.010 Bit7=1) -2,147,483,648 ~ +2,147,483,647 |
| NC Variable to MLC 75 | D3x202 | The system will move data from NC variable #25458 to this special D. | R | | |
| NC Variable to MLC 76 | D3x203 | The system will move data from NC variable #25459 to this special D. *Both this special D and the variable # are NOT available to use when using MLC floating mode **[N1.010 Bit7=1]**. Otherwise, the system will return error 0x235. | R | | |
| NC Variable to MLC 77 | D3x204 | The system will move data from NC variable #25460 to this special D. | R | | |
| NC Variable to MLC 78 | D3x205 | The system will move data from NC variable #25461 to this special D. *Both this special D and the variable # are NOT available to use when using MLC floating mode **[N1.010 Bit7=1]**. Otherwise, the system will return error 0x235. | R | | |
| NC Variable to MLC 79 | D3x206 | The system will move data from NC variable #25462 to this special D. | R | | |
| NC Variable to MLC 80 | D3x207 | The system will move data from NC variable #25463 to this special D. *Both this special D and the variable # are NOT available to use when using MLC floating mode **[N1.010 Bit7=1]**. Otherwise, the system will return error 0x235. | R | | |
| NC Variable to MLC 81 | D3x208 | The system will move data from NC variable #25464 to this special D. | R | | |
| NC Variable to MLC 82 | D3x209 | The system will move data from NC variable #25465 to this special D. *Both this special D and the variable # are NOT available to use when using MLC floating mode **[N1.010 Bit7=1]**. Otherwise, the system will return error 0x235. | R | | |
| NC Variable to MLC 83 | D3x210 | The system will move data from NC variable #25466 to this special D. | R | | |
| NC Variable to MLC 84 | D3x211 | The system will move data from NC variable #25467 to this special D. *Both this special D and the variable # are NOT available to use when using MLC floating mode **[N1.010 Bit7=1]**. Otherwise, the system will return error 0x235. | R | | |
| NC Variable to MLC 85 | D3x212 | The system will move data from NC variable #25468 to this special D. | R | | |

| Function Name | Special D | Description | Device | Type | Range |
|---|---|---|---|---|---|
| NC Variable to MLC 86 | D3x213 | The system will move data from NC variable #25469 to this special D.<br>*Both this special D and the variable # are NOT available to use when using MLC floating mode **[N1.010 Bit7=1]**. Otherwise, the system will return error 0x235. | R | Decimal / Float | (Default) -32,768 ~ +32,767 / (N1.010 Bit7=1) -2,147,483,648 ~ +2,147,483,647 |
| NC Variable to MLC 87 | D3x214 | The system will move data from NC variable #25470 to this special D. | R | | |
| NC Variable to MLC 88 | D3x215 | The system will move data from NC variable #25471 to this special D.<br>*Both this special D and the variable # are NOT available to use when using MLC floating mode **[N1.010 Bit7=1]**. Otherwise, the system will return error 0x235. | R | | |
| NC Variable to MLC 89 | D3x216 | The system will move data from NC variable #25472 to this special D. | R | | |
| NC Variable to MLC 90 | D3x217 | The system will move data from NC variable #25473 to this special D.<br>*Both this special D and the variable # are NOT available to use when using MLC floating mode **[N1.010 Bit7=1]**. Otherwise, the system will return error 0x235. | R | | |
| NC Variable to MLC 91 | D3x218 | The system will move data from NC variable #25474 to this special D. | R | | |
| NC Variable to MLC 92 | D3x219 | The system will move data from NC variable #25475 to this special D.<br>*Both this special D and the variable # are NOT available to use when using MLC floating mode **[N1.010 Bit7=1]**. Otherwise, the system will return error 0x235. | R | | |
| NC Variable to MLC 93 | D3x220 | The system will move data from NC variable #25476 to this special D. | R | | |
| NC Variable to MLC 94 | D3x221 | The system will move data from NC variable #25477 to this special D.<br>*Both this special D and the variable # are NOT available to use when using MLC floating mode **[N1.010 Bit7=1]**. Otherwise, the system will return error 0x235. | R | | |
| NC Variable to MLC 95 | D3x222 | The system will move data from NC variable #25478 to this special D. | R | | |
| NC Variable to MLC 96 | D3x223 | The system will move data from NC variable #25479 to this special D.<br>*Both this special D and the variable # are NOT available to use when using MLC floating mode **[N1.010 Bit7=1]**. Otherwise, the system will return error 0x235. | R | | |
| NC Variable to MLC 97 | D3x224 | The system will move data from NC variable #25480 to this special D. | R | | |

5

| Function Name | Special D | Description | Device | Type | Range |
|---|---|---|---|---|---|
| NC Variable to MLC 98 | D3x225 | The system will move data from NC variable #25481 to this special D.<br>*Both this special D and the variable # are NOT available to use when using MLC floating mode **[N1.010 Bit7=1]**. Otherwise, the system will return error 0x235. | R | Decimal / Float | (Default) -32,768 ~ +32,767 / (N1.010 Bit7=1) -2,147,483,648 ~ +2,147,483,647 |
| NC Variable to MLC 99 | D3x226 | The system will move data from NC variable #25482 to this special D. | R | | |
| NC Variable to MLC 100 | D3x227 | The system will move data from NC variable #25483 to this special D.<br>*Both this special D and the variable # are NOT available to use when using MLC floating mode **[N1.010 Bit7=1]**. Otherwise, the system will return error 0x235. | R | | |
| NC Variable to MLC 101 | D3x228 | The system will move data from NC variable #25484 to this special D. | R | | |
| NC Variable to MLC 102 | D3x229 | The system will move data from NC variable #25485 to this special D.<br>*Both this special D and the variable # are NOT available to use when using MLC floating mode **[N1.010 Bit7=1]**. Otherwise, the system will return error 0x235. | R | | |
| NC Variable to MLC 103 | D3x230 | The system will move data from NC variable #25486 to this special D. | R | | |
| NC Variable to MLC 104 | D3x231 | The system will move data from NC variable #25487 to this special D.<br>*Both this special D and the variable # are NOT available to use when using MLC floating mode **[N1.010 Bit7=1]**. Otherwise, the system will return error 0x235. | R | | |
| NC Variable to MLC 105 | D3x232 | The system will move data from NC variable #25488 to this special D. | R | | |
| NC Variable to MLC 106 | D3x233 | The system will move data from NC variable #25489 to this special D.<br>*Both this special D and the variable # are NOT available to use when using MLC floating mode **[N1.010 Bit7=1]**. Otherwise, the system will return error 0x235. | R | | |
| NC Variable to MLC 107 | D3x234 | The system will move data from NC variable #25490 to this special D. | R | | |
| NC Variable to MLC 108 | D3x235 | The system will move data from NC variable #25491 to this special D.<br>*Both this special D and the variable # are NOT available to use when using MLC floating mode **[N1.010 Bit7=1]**. Otherwise, the system will return error 0x235. | R | | |
| NC Variable to MLC 109 | D3x236 | The system will move data from NC variable #25492 to this special D. | R | | |

| Function Name | Special D | Description | Device | Type | Range |
|---|---|---|---|---|---|
| NC Variable to MLC 110 | D3x237 | The system will move data from NC variable #25493 to this special D. *Both this special D and the variable # are NOT available to use when using MLC floating mode **[N1.010 Bit7=1]**. Otherwise, the system will return error 0x235. | R | Decimal / Float | (Default) -32,768 ~ +32,767 / (N1.010 Bit7=1) -2,147,483,648 ~ +2,147,483,647 |
| NC Variable to MLC 111 | D3x238 | The system will move data from NC variable #25494 to this special D. | R | | |
| NC Variable to MLC 112 | D3x239 | The system will move data from NC variable #25495 to this special D. *Both this special D and the variable # are NOT available to use when using MLC floating mode **[N1.010 Bit7=1]**. Otherwise, the system will return error 0x235. | R | | |
| NC Variable to MLC 113 | D3x240 | The system will move data from NC variable #25496 to this special D. | R | | |
| NC Variable to MLC 114 | D3x241 | The system will move data from NC variable #25497 to this special D. *Both this special D and the variable # are NOT available to use when using MLC floating mode **[N1.010 Bit7=1]**. Otherwise, the system will return error 0x235. | R | | |
| NC Variable to MLC 115 | D3x242 | The system will move data from NC variable #25498 to this special D. | R | | |
| NC Variable to MLC 116 | D3x243 | The system will move data from NC variable #25499 to this special D. *Both this special D and the variable # are NOT available to use when using MLC floating mode **[N1.010 Bit7=1]**. Otherwise, the system will return error 0x235. | R | | |
| NC Variable to MLC 117 | D3x244 | The system will move data from NC variable #25500 to this special D. | R | | |
| NC Variable to MLC 118 | D3x245 | The system will move data from NC variable #25501 to this special D. *Both this special D and the variable # are NOT available to use when using MLC floating mode **[N1.010 Bit7=1]**. Otherwise, the system will return error 0x235. | R | | |
| NC Variable to MLC 119 | D3x246 | The system will move data from NC variable #25502 to this special D. | R | | |
| NC Variable to MLC 120 | D3x247 | The system will move data from NC variable #25503 to this special D. *Both this special D and the variable # are NOT available to use when using MLC floating mode **[N1.010 Bit7=1]**. Otherwise, the system will return error 0x235. | R | | |
| NC Variable to MLC 121 | D3x248 | The system will move data from NC variable #25504 to this special D. | R | | |

5

| Function Name | Special D | Description | Device | Type | Range |
|---|---|---|---|---|---|
| NC Variable to MLC 122 | D3x249 | The system will move data from NC variable #25505 to this special D.<br>*Both this special D and the variable # are NOT available to use when using MLC floating mode **[N1.010 Bit7=1]**. Otherwise, the system will return error 0x235. | R | Decimal / Float | (Default) -32,768 ~ +32,767 / (N1.010 Bit7=1) -2,147,483,648 ~ +2,147,483,647 |
| NC Variable to MLC 123 | D3x250 | The system will move data from NC variable #25506 to this special D. | R | | |
| NC Variable to MLC 124 | D3x251 | The system will move data from NC variable #25507 to this special D.<br>*Both this special D and the variable # are NOT available to use when using MLC floating mode **[N1.010 Bit7=1]**. Otherwise, the system will return error 0x235. | R | | |
| NC Variable to MLC 125 | D3x252 | The system will move data from NC variable #25508 to this special D. | R | | |
| NC Variable to MLC 126 | D3x253 | The system will move data from NC variable #25509 to this special D.<br>*Both this special D and the variable # are NOT available to use when using MLC floating mode **[N1.010 Bit7=1]**. Otherwise, the system will return error 0x235. | R | | |
| NC Variable to MLC 127 | D3x254 | The system will move data from NC variable #25510 to this special D. | R | | |
| NC Variable to MLC 128 | D3x255 | The system will move data from NC variable #25511 to this special D.<br>*Both this special D and the variable # are NOT available to use when using MLC floating mode **[N1.010 Bit7=1]**. Otherwise, the system will return error 0x235. | R | | |
| Look Ahead Remaining Command | D3x320<br>D3x321 | When the system is executing an NC program, it will preview the current program and planning path, and the remaining number of previewed and executable blocks will be written to this special D. | R | Decimal | 0 ~ 4,294,967,295 |
| Rigid Tapping Max Error of 1st Spindle | D3x350<br>D3x351 | The system will record the maximum absolute difference distance between the feed axis feedback and spindle feedback when the 1st spindle executes the tapping function.<br>This special D will be reset prior to every time the tapping function is triggered.<br>Unit: mm | R | Float | -2,147,483,648 ~ +2,147,483,647 |
| Rigid Tapping Max Error of 2nd Spindle | D3x352<br>D3x353 | The system will record the maximum absolute difference distance between the feed axis feedback and spindle feedback when the 2nd spindle executes the tapping function.<br>This special D will be reset prior to every time the tapping function is triggered.<br>Unit: mm | R | Float | -2,147,483,648 ~ +2,147,483,647 |

| Function Name | Special D | Description | Device | Type | Range |
|---|---|---|---|---|---|
| Target Feed of 1st Spindle | D3x354 D3x355 | Target feed movement of the 1st spindle in each revolution.<br>Unit: mm/rev. | R | Float | -2,147,483,648 ~ +2,147,483,647 |
| Target Feed of 2nd Spindle | D3x356 D3x357 | Target feed movement of the 2nd spindle in each revolution.<br>Unit: mm/rev. | R | Float | -2,147,483,648 ~ +2,147,483,647 |
| Actual Feed of 1st Spindle | D3x358 D3x359 | Actual feed movement of the 1st spindle in each revolution.<br>Unit: mm/rev. | R | Float | -2,147,483,648 ~ +2,147,483,647 |
| Actual Feed of 2nd Spindle | D3x360 D3x361 | Actual feed movement of the 2nd spindle in each revolution.<br>Unit: mm/rev. | R | Float | -2,147,483,648 ~ +2,147,483,647 |
| Speed Command of 3rd Spindle | D3x362 D3x363 | When the 3rd spindle S code is executed in a program, the command value will be sent in this special D.<br>Unit: RPM | R | Float | -2,147,483,648 ~ +2,147,483,647 |
| 3rd Spindle Speed Feedback | D3x364 D3x365 | Shows the 3rd spindle's speed. The data source is from the spindle's command speed. | R | Float | -2,147,483,648 ~ +2,147,483,647 |
| 3rd Spindle Actual Degree | D3x366 D3x367 | Shows the 3rd spindle's actual degree. The data source is from the spindle's actual degree. | R | Float | -2,147,483,648 ~ +2,147,483,647 |
| Rigid Tapping Max Error of 3rd Spindle | D3x368 D3x369 | The system will record the maximum absolute difference distance between the feed axis feedback and spindle feedback when the 3rd spindle executes the tapping function.<br>This special D will be reset prior to every time the tapping function is triggered.<br>Unit: mm | R | Float | -2,147,483,648 ~ +2,147,483,647 |
| Target Feed of 3rd Spindle | D3x370 D3x371 | Target feed movement of the 3rd spindle in each revolution.<br>Unit: mm/rev. | R | Float | -2,147,483,648 ~ +2,147,483,647 |
| Actual Feed of 3rd Spindle | D3x372 D3x373 | Actual feed movement of the 3rd spindle in each revolution.<br>Unit: mm/rev. | R | Float | -2,147,483,648 ~ +2,147,483,647 |
| Speed Command of 4th Spindle | D3x374 D3x375 | When the 4th spindle S code is executed in a program, the command value will be sent in this special D.<br>Unit: RPM | R | Float | -2,147,483,648 ~ +2,147,483,647 |
| 4th Spindle Speed Feedback | D3x376 D3x377 | Shows the 4th spindle's speed. The data source is from the spindle's command speed. | R | Float | -2,147,483,648 ~ +2,147,483,647 |
| 4th Spindle Actual Degree | D3x378 D3x379 | Shows the 4th spindle's actual degree. The data source is from the spindle's actual degree. | R | Float | -2,147,483,648 ~ +2,147,483,647 |
| Rigid Tapping Max Error of 4th Spindle | D3x380 D3x381 | The system will record the maximum absolute difference distance between the feed axis feedback and spindle feedback when the 4th spindle executes the tapping function.<br>This special D will be reset prior to every time the tapping function is triggered.<br>Unit: mm | R | Float | -2,147,483,648 ~ +2,147,483,647 |

5

| Function Name | Special D | Description | Device | Type | Range |
|---|---|---|---|---|---|
| Target Feed of 4th Spindle | D3x382<br>D3x383 | Target feed movement of the 4th spindle in each revolution.<br>Unit: mm/rev. | R | Float | -2,147,483,648<br>~<br>+2,147,483,647 |
| Actual Feed of 4th Spindle | D3x384<br>D3x385 | Actual feed movement of the 4th spindle in each revolution.<br>Unit: mm/rev. | R | Float | -2,147,483,648<br>~<br>+2,147,483,647 |
| Speed Command of 5th Spindle | D3x386<br>D3x387 | When the 5th spindle S code is executed in a program, the command value will be sent in this special D.<br>Unit: RPM | R | Float | -2,147,483,648<br>~<br>+2,147,483,647 |
| 5th Spindle Speed Feedback | D3x388<br>D3x389 | Shows the 5th spindle's speed. The data source is from the spindle's command speed. | R | Float | -2,147,483,648<br>~<br>+2,147,483,647 |
| 5th Spindle Actual Degree | D3x390<br>D3x391 | Shows the 5th spindle's actual degree. The data source is from the spindle's actual degree. | R | Float | -2,147,483,648<br>~<br>+2,147,483,647 |
| Rigid Tapping Max Error of 5th Spindle | D3x392<br>D3x393 | The system will record the maximum absolute difference distance between the feed axis feedback and spindle feedback when the 5th spindle executes the tapping function.<br>This special D will be reset prior to every time the tapping function is triggered.<br>Unit: mm | R | Float | -2,147,483,648<br>~<br>+2,147,483,647 |
| Target Feed of 5th Spindle | D3x394<br>D3x395 | Target feed movement of the 5th spindle in each revolution.<br>Unit: mm/rev. | R | Float | -2,147,483,648<br>~<br>+2,147,483,647 |
| Actual Feed of 5th Spindle | D3x396<br>D3x397 | Actual feed movement of the 5th spindle in each revolution.<br>Unit: mm/rev. | R | Float | -2,147,483,648<br>~<br>+2,147,483,647 |
| Speed Command of 6th Spindle | D3x398<br>D3x399 | When the 6th spindle S code is executed in a program, the command value will be sent in this special D.<br>Unit: RPM | R | Float | -2,147,483,648<br>~<br>+2,147,483,647 |
| 6th Spindle Speed Feedback | D3x400<br>D3x401 | Shows the 6th spindle's speed. The data source is from the spindle's command speed. | R | Float | -2,147,483,648<br>~<br>+2,147,483,647 |
| 6th Spindle Actual Degree | D3x402<br>D3x403 | Shows the 6th spindle's actual degree. The data source is from the spindle's actual degree. | R | Float | -2,147,483,648<br>~<br>+2,147,483,647 |
| Rigid Tapping Max Error of 6th Spindle | D3x404<br>D3x405 | The system will record the maximum absolute difference distance between the feed axis feedback and spindle feedback when the 6th spindle executes the tapping function.<br>This special D will be reset prior to every time the tapping function is triggered.<br>Unit: mm | R | Float | -2,147,483,648<br>~<br>+2,147,483,647 |
| Target Feed of 6th Spindle | D3x406<br>D3x407 | Target feed movement of the 6th spindle in each revolution.<br>Unit: mm/rev. | R | Float | -2,147,483,648<br>~<br>+2,147,483,647 |
| Actual Feed of 6th Spindle | D3x408<br>D3x409 | Actual feed movement of the 6th spindle in each revolution.<br>Unit: mm/rev. | R | Float | -2,147,483,648<br>~<br>+2,147,483,647 |

| Function Name | Special D | Description | Device | Type | Range |
|---|---|---|---|---|---|
| Speed Command of 7th Spindle | D3x410<br>D3x411 | When the 7th spindle S code is executed in a program, the command value will be sent in this special D.<br>Unit: RPM | R | Float | -2,147,483,648<br>~<br>+2,147,483,647 |
| 7th Spindle Speed Feedback | D3x412<br>D3x413 | Shows the 7th spindle's speed. The data source is from the spindle's command speed. | R | Float | -2,147,483,648<br>~<br>+2,147,483,647 |
| 7th Spindle Actual Degree | D3x414<br>D3x415 | Shows the 7th spindle's actual degree. The data source is from the spindle's actual degree. | R | Float | -2,147,483,648<br>~<br>+2,147,483,647 |
| Rigid Tapping Max Error of 7th Spindle | D3x416<br>D3x417 | The system will record the maximum absolute difference distance between the feed axis feedback and spindle feedback when the 7th spindle executes the tapping function.<br>This special D will be reset prior to every time the tapping function is triggered.<br>Unit: mm | R | Float | -2,147,483,648<br>~<br>+2,147,483,647 |
| Target Feed of 7th Spindle | D3x418<br>D3x419 | Target feed movement of the 7th spindle in each revolution.<br>Unit: mm/rev. | R | Float | -2,147,483,648<br>~<br>+2,147,483,647 |
| Actual Feed of 7th Spindle | D3x420<br>D3x421 | Actual feed movement of the 7th spindle in each revolution.<br>Unit: mm/rev. | R | Float | -2,147,483,648<br>~<br>+2,147,483,647 |
| Speed Command of 8th Spindle | D3x422<br>D3x423 | When the 8th spindle S code is executed in a program, the command value will be sent in this special D.<br>Unit: RPM | R | Float | -2,147,483,648<br>~<br>+2,147,483,647 |
| 8th Spindle Speed Feedback | D3x424<br>D3x425 | Shows the 8th spindle's speed. The data source is from the spindle's command speed. | R | Float | -2,147,483,648<br>~<br>+2,147,483,647 |
| 8th Spindle Actual Degree | D3x426<br>D3x427 | Shows the 8th spindle's actual degree. The data source is from the spindle's actual degree. | R | Float | -2,147,483,648<br>~<br>+2,147,483,647 |
| Rigid Tapping Max Error of 8th Spindle | D3x428<br>D3x429 | The system will record the maximum absolute difference distance between the feed axis feedback and spindle feedback when the 8th spindle executes the tapping function.<br>This special D will be reset prior to every time the tapping function is triggered.<br>Unit: mm | R | Float | -2,147,483,648<br>~<br>+2,147,483,647 |
| Target Feed of 8th Spindle | D3x430<br>D3x431 | Target feed movement of the 8th spindle in each revolution.<br>Unit: mm/rev. | R | Float | -2,147,483,648<br>~<br>+2,147,483,647 |
| Actual Feed of 8th Spindle | D3x432<br>D3x433 | Actual feed movement of the 8th spindle in each revolution.<br>Unit: mm/rev. | R | Float | -2,147,483,648<br>~<br>+2,147,483,647 |

5

# 5.3 Special M and special D functions

## 5.3.1 Mode switching

The NC5 controller provides 8 different types of system modes, which can be switched by setting the MLC special D2x000 for any of them. Special M can be used to monitor the mode status. The x digit in D2x000 represents different NC channels. For instance, to change the NC mode in channel 1, users need to set D21000 to switch NC mode.

| Function Name | Special M | Function Name | Special M |
|---|---|---|---|
| AUTO | M3x000 | JOG | M3x004 |
| EDIT | M3x001 | RAPID | M3x005 |
| MDI | M3x002 | INC | M3x006 |
| MPG | M3x003 | HOME | M3x007 |

### 5.3.2 NC processing

The NC5 controller provides various special M relays and special D registers related to the machining statuses. Users 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 |
|---|---|---|---|
| Cycle Start | M2x000 | Single Block | M2x008 |
| Feed Hold | M2x001 | Optional Stop | M2x009 |
| NC Reset | M2x004 | Single Block Skip (‘/’) | M2x010 |
| Enable Dry Run | M2x005 | M96 Program Interruption | M2x019 |
| MPG Simulation | M2x006 | - | - |

| Function Name | Special M | Function Name | Special M |
|---|---|---|---|
| Cycle Start Status | M3x016 | M30 Cycle Stop and Index Reset | M3x023 |
| Feed Hold Status | M3x017 | Single Block Hold | M3x024 |
| Emergency Stop Status | M3x018 | M96 (Program Interruption) in Execution | M3x031 |
| Reset Finished | M3x019 | Main Program Lock | M3x033 |
| Break Point Searching | M3x020 | M00 System Hold | M3x037 |
| Program End Finished | M3x021 | M01 Optional Hold | M3x038 |
| M02 Executed | M3x022 | Block Finished on Single Block Mode | M3x039 |

| Function Name | Special D | Function Name | Special D |
|---|---|---|---|
| Feed Rate Percentage | D2x002 | Feed Rate Override for JOG, INC and Dry Run Mode | D2x006 |
| Rapid Speed Percentage | D2x004 | - | - |

5

### 5.3.3 Axis status

The NC5 controller provides the corresponding special M relays and special D registers for each axis. With these special M relays and special D registers, users can lock the axis, cancel the 1st software limit, or access the status, position, or current status of the axis.

| Function Name | Special M | Function Name | Special M |
|---|---|---|---|
| Disable Hardware Limit | M2x007 | Lock 15th Axis Movement in Positive Direction | M2x350 |
| X Axis Servo OFF | M2x272 | Lock 16th Axis Movement in Positive Direction | M2x351 |
| Y Axis Servo OFF | M2x273 | Lock X Axis Movement in Negative Direction | M2x352 |
| Z Axis Servo OFF | M2x274 | Lock Y Axis Movement in Negative Direction | M2x353 |
| A Axis Servo OFF | M2x275 | Lock Z Axis Movement in Negative Direction | M2x354 |
| B Axis Servo OFF | M2x276 | Lock A Axis Movement in Negative Direction | M2x355 |
| C Axis Servo OFF | M2x277 | Lock B Axis Movement in Negative Direction | M2x356 |
| U Axis Servo OFF | M2x278 | Lock C Axis Movement in Negative Direction | M2x357 |
| V Axis Servo OFF | M2x279 | Lock U Axis Movement in Negative Direction | M2x358 |
| W Axis Servo OFF | M2x280 | Lock V Axis Movement in Negative Direction | M2x359 |
| 10th Axis Servo OFF | M2x281 | Lock W Axis Movement in Negative Direction | M2x360 |
| 11th Axis Servo OFF | M2x282 | Lock 10th Axis Movement in Negative Direction | M2x361 |
| 12th Axis Servo OFF | M2x283 | Lock 11th Axis Movement in Negative Direction | M2x362 |
| 13th Axis Servo OFF | M2x284 | Lock 12th Axis Movement in Negative Direction | M2x363 |
| 14th Axis Servo OFF | M2x285 | Lock 13th Axis Movement in Negative Direction | M2x364 |
| 15th Axis Servo OFF | M2x286 | Lock 14th Axis Movement in Negative Direction | M2x365 |
| 16th Axis Servo OFF | M2x287 | Lock 15th Axis Movement in Negative Direction | M2x366 |
| Lock X Axis Movement in Positive Direction | M2x336 | Lock 16th Axis Movement in Negative Direction | M2x367 |
| Lock Y Axis Movement in Positive Direction | M2x337 | Disable X Axis 1st Software Limit | M2x368 |
| Lock Z Axis Movement in Positive Direction | M2x338 | Disable Y Axis 1st Software Limit | M2x369 |
| Lock A Axis Movement in Positive Direction | M2x339 | Disable Z Axis 1st Software Limit | M2x370 |
| Lock B Axis Movement in Positive Direction | M2x340 | Disable A Axis 1st Software Limit | M2x371 |
| Lock C Axis Movement in Positive Direction | M2x341 | Disable B Axis 1st Software Limit | M2x372 |
| Lock U Axis Movement in Positive Direction | M2x342 | Disable C Axis 1st Software Limit | M2x373 |

| Function Name | Special M | Function Name | Special M |
|---|---|---|---|
| Lock V Axis Movement in Positive Direction | M2x343 | Disable U Axis 1st Software Limit | M2x374 |
| Lock W Axis Movement in Positive Direction | M2x344 | Disable V Axis 1st Software Limit | M2x375 |
| Lock 10th Axis Movement in Positive Direction | M2x345 | Disable W Axis 1st Software Limit | M2x376 |
| Lock 11th Axis Movement in Positive Direction | M2x346 | Disable 10th Axis 1st Software Limit | M2x377 |
| Lock 12th Axis Movement in Positive Direction | M2x347 | Disable 11th Axis 1st Software Limit | M2x378 |
| Lock 13th Axis Movement in Positive Direction | M2x348 | Disable 12th Axis 1st Software Limit | M2x379 |
| Lock 14th Axis Movement in Positive Direction | M2x349 | Disable 13th Axis 1st Software Limit | M2x380 |
| Disable 14th Axis 1st Software Limit | M2x381 | Disable 16th Axis 1st Software Limit | M2x383 |
| Disable 15th Axis 1st Software Limit | M2x382 | - | - |

5

| Function Name | Special M | Function Name | Special M |
|---|---|---|---|
| Servo Enable Status of X Axis | M3x272 | X Axis Moving Forward | M3x480 |
| Servo Enable Status of Y Axis | M3x273 | Y Axis Moving Forward | M3x481 |
| Servo Enable Status of Z Axis | M3x274 | Z Axis Moving Forward | M3x482 |
| Servo Enable Status of A Axis | M3x275 | A Axis Moving Forward | M3x483 |
| Servo Enable Status of B Axis | M3x276 | B Axis Moving Forward | M3x484 |
| Servo Enable Status of C Axis | M3x277 | C Axis Moving Forward | M3x485 |
| Servo Enable Status of U Axis | M3x278 | U Axis Moving Forward | M3x486 |
| Servo Enable Status of V Axis | M3x279 | V Axis Moving Forward | M3x487 |
| Servo Enable Status of W Axis | M3x280 | W Axis Moving Forward | M3x488 |
| Servo Enable Status of $10^{th}$ Axis | M3x281 | $10^{th}$ Axis Moving Forward | M3x489 |
| Servo Enable Status of $11^{th}$ Axis | M3x282 | $11^{th}$ Axis Moving Forward | M3x490 |
| Servo Enable Status of $12^{th}$ Axis | M3x283 | $12^{th}$ Axis Moving Forward | M3x491 |
| Servo Enable Status of $13^{th}$ Axis | M3x284 | $13^{th}$ Axis Moving Forward | M3x492 |
| Servo Enable Status of $14^{th}$ Axis | M3x285 | $14^{th}$ Axis Moving Forward | M3x493 |
| Servo Enable Status of $15^{th}$ Axis | M3x286 | $15^{th}$ Axis Moving Forward | M3x494 |
| Servo Enable Status of $16^{th}$ Axis | M3x287 | $16^{th}$ Axis Moving Forward | M3x495 |
| X Axis Is Moving | M3x464 | X Axis Moving Backward | M3x496 |
| Y Axis Is Moving | M3x465 | Y Axis Moving Backward | M3x497 |
| Z Axis Is Moving | M3x466 | Z Axis Moving Backward | M3x498 |
| A Axis Is Moving | M3x467 | A Axis Moving Backward | M3x499 |
| B Axis Is Moving | M3x468 | B Axis Moving Backward | M3x500 |
| C Axis Is Moving | M3x469 | C Axis Moving Backward | M3x501 |
| U Axis Is Moving | M3x470 | U Axis Moving Backward | M3x502 |
| V Axis Is Moving | M3x471 | V Axis Moving Backward | M3x503 |
| W Axis Is Moving | M3x472 | W Axis Moving Backward | M3x504 |
| $10^{th}$ Axis Is Moving | M3x473 | $10^{th}$ Axis Moving Backward | M3x505 |
| $11^{th}$ Axis Is Moving | M3x474 | $11^{th}$ Axis Moving Backward | M3x506 |
| $12^{th}$ Axis Is Moving | M3x475 | $12^{th}$ Axis Moving Backward | M3x507 |
| $13^{th}$ Axis Is Moving | M3x476 | $13^{th}$ Axis Moving Backward | M3x508 |
| $14^{th}$ Axis Is Moving | M3x477 | $14^{th}$ Axis Moving Backward | M3x509 |
| $15^{th}$ Axis Is Moving | M3x478 | $15^{th}$ Axis Moving Backward | M3x510 |
| $16^{th}$ Axis Is Moving | M3x479 | $16^{th}$ Axis Moving Backward | M3x511 |

5

### 5.3.4 Homing origin

The NC5 controller homing process is triggered by special M relays. Users can determine the homing sequence according to the machine design.

| Function Name | Special M | Function Name | Special M |
|---|---|---|---|
| X Axis Homing | M2x320 | W Axis Homing | M2x328 |
| Y Axis Homing | M2x321 | 10th Axis Homing | M2x329 |
| Z Axis Homing | M2x322 | 11th Axis Homing | M2x330 |
| A Axis Homing | M2x323 | 12th Axis Homing | M2x331 |
| B Axis Homing | M2x324 | 13th Axis Homing | M2x332 |
| C Axis Homing | M2x325 | 14th Axis Homing | M2x333 |
| U Axis Homing | M2x326 | 15th Axis Homing | M2x334 |
| V Axis Homing | M2x327 | 16th Axis Homing | M2x335 |

| Function Name | Special M | Function Name | Special M |
|---|---|---|---|
| Axis Homed Status of X Axis | M3x320 | Homing Finished Status of W Axis | M3x344 |
| Axis Homed Status of Y Axis | M3x321 | Homing Finished Status of 10th Axis | M3x345 |
| Axis Homed Status of Z Axis | M3x322 | Homing Finished Status of 11th Axis | M3x346 |
| Axis Homed Status of A Axis | M3x323 | Homing Finished Status of 12th Axis | M3x347 |
| Axis Homed Status of B Axis | M3x324 | Homing Finished Status of 13th Axis | M3x348 |
| Axis Homed Status of C Axis | M3x325 | Homing Finished Status of 14th Axis | M3x349 |
| Axis Homed Status of U Axis | M3x326 | Homing Finished Status of 15th Axis | M3x350 |
| Axis Homed Status of V Axis | M3x327 | Homing Finished Status of 16th Axis | M3x351 |
| Axis Homed Status of W Axis | M3x328 | X Axis At Origin Position | M3x352 |
| Axis Homed Status of 10th Axis | M3x329 | Y Axis At Origin Position | M3x353 |
| Axis Homed Status of 11th Axis | M3x330 | Z Axis At Origin Position | M3x354 |
| Axis Homed Status of 12th Axis | M3x331 | A Axis At Origin Position | M3x355 |
| Axis Homed Status of 13th Axis | M3x332 | B Axis At Origin Position | M3x356 |
| Axis Homed Status of 14th Axis | M3x333 | C Axis At Origin Position | M3x357 |
| Axis Homed Status of 15th Axis | M3x334 | U Axis At Origin Position | M3x358 |
| Axis Homed Status of 16th Axis | M3x335 | V Axis At Origin Position | M3x359 |
| Homing Finished Status of X Axis | M3x336 | W Axis At Origin Position | M3x360 |
| Homing Finished Status of Y Axis | M3x337 | 10th Axis At Origin Position | M3x361 |
| Homing Finished Status of Z Axis | M3x338 | 11th Axis At Origin Position | M3x362 |
| Homing Finished Status of A Axis | M3x339 | 12th Axis At Origin Position | M3x363 |
| Homing Finished Status of B Axis | M3x340 | 13th Axis At Origin Position | M3x364 |
| Homing Finished Status of C Axis | M3x341 | 14th Axis At Origin Position | M3x365 |
| Homing Finished Status of U Axis | M3x342 | 15th Axis At Origin Position | M3x366 |
| Homing Finished Status of V Axis | M3x343 | 16th Axis At Origin Position | M3x367 |

5

### 5.3.5 JOG control

The jog operations and jogging speed of each axis are determined by special M relays and special D registers. Users can change the speed and direction (forward or reverse) with the ladder diagram at any time.

Note: this action control is only available when the system is in JOG or INC mode.

| Function Name | Special M | Function Name | Special M |
|---|---|---|---|
| X Axis JOG Forward | M2x384 | X Axis JOG Reverse | M2x400 |
| Y Axis JOG Forward | M2x385 | Y Axis JOG Reverse | M2x401 |
| Z Axis JOG Forward | M2x386 | Z Axis JOG Reverse | M2x402 |
| A Axis JOG Forward | M2x387 | A Axis JOG Reverse | M2x403 |
| B Axis JOG Forward | M2x388 | B Axis JOG Reverse | M2x404 |
| C Axis JOG Forward | M2x389 | C Axis JOG Reverse | M2x405 |
| U Axis JOG Forward | M2x390 | U Axis JOG Reverse | M2x406 |
| V Axis JOG Forward | M2x391 | V Axis JOG Reverse | M2x407 |
| W Axis JOG Forward | M2x392 | W Axis JOG Reverse | M2x408 |
| 10th Axis JOG Forward | M2x393 | 10th Axis JOG Reverse | M2x409 |
| 11th Axis JOG Forward | M2x394 | 11th Axis JOG Reverse | M2x410 |
| 12th Axis JOG Forward | M2x395 | 12th Axis JOG Reverse | M2x411 |
| 13th Axis JOG Forward | M2x396 | 13th Axis JOG Reverse | M2x412 |
| 14th Axis JOG Forward | M2x397 | 14th Axis JOG Reverse | M2x413 |
| 15th Axis JOG Forward | M2x398 | 15th Axis JOG Reverse | M2x414 |
| 16th Axis JOG Forward | M2x399 | 16th Axis JOG Reverse | M2x415 |

| Function Name | Special D | Function Name | Special D |
|---|---|---|---|
| Feed Rate Override for JOG, INC and Dry Run Mode | D2x006 | Axes Movement in INC Mode | D2x014 |

### 5.3.6 MPG control

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

| Function Name | Special M | Function Name | Special M |
|---|---|---|---|
| MPG Simulation | M2x006 | - | - |

| Function Name | Special D | Function Name | Special D |
|---|---|---|---|
| MPG Axes Selection | D2x008 | MPG Ratio Selection | D2x009 |

5

### 5.3.7 Skip control (G31)

When using the G31 Skip command, the NC system will send the special M signals for users to check if the command signal is triggered.

| Function Name | Special M | Function Name | Special M |
|---|---|---|---|
| HSI 1 | M30016 | HSI 5 | M30020 |
| HSI 2 | M30017 | HSI 6 | M30021 |
| HSI 3 | M30018 | HSI 7 | M30022 |
| HSI 4 | M30019 | HSI 8 | M30023 |

### 5.3.8 One-button macro call

The NC5 controller provides a convenient function, which can call a specific program for machining with a one button trigger.

Note: this function is only available when the system is in AUTO mode.

| Function Name | Special M | Function Name | Special M |
|---|---|---|---|
| Macro Call Activation | M2x025 | 9th Macro Call Preparation | M2x040 |
| 1st Macro Call Preparation | M2x032 | 10th Macro Call Preparation | M2x041 |
| 2nd Macro Call Preparation | M2x033 | 11th Macro Call Preparation | M2x042 |
| 3rd Macro Call Preparation | M2x034 | 12th Macro Call Preparation | M2x043 |
| 4th Macro Call Preparation | M2x035 | 13th Macro Call Preparation | M2x044 |
| 5th Macro Call Preparation | M2x036 | 14th Macro Call Preparation | M2x045 |
| 6th Macro Call Preparation | M2x037 | 15th Macro Call Preparation | M2x046 |
| 7th Macro Call Preparation. | M2x038 | 16th Macro Call Preparation | M2x047 |
| 8th Macro Call Preparation | M2x039 | - | - |

| Function Name | Special M | Function Name | Special M |
|---|---|---|---|
| Macro Call Status | M3x027 | 8th Macro Call Initial Finished | M3x055 |
| Macro Call Ready | M3x028 | 9th Macro Call Initial Finished | M3x056 |
| Macro Call Error | M3x029 | 10th Macro Call Initial Finished | M3x057 |
| 1st Macro Call Initial Finished | M3x048 | 11th Macro Call Initial Finished | M3x058 |
| 2nd Macro Call Initial Finished | M3x049 | 12th Macro Call Initial Finished | M3x059 |
| 3rd Macro Call Initial Finished | M3x050 | 13th Macro Call Initial Finished | M3x060 |
| 4th Macro Call Initial Finished | M3x051 | 14th Macro Call Initial Finished | M3x061 |
| 5th Macro Call Initial Finished | M3x052 | 15th Macro Call Initial Finished | M3x062 |
| 6th Macro Call Initial Finished | M3x053 | 16th Macro Call Initial Finished | M3x063 |
| 7th Macro Call Initial Finished | M3x054 | - | - |

| Function Name | Special D | Function Name | Special D |
|---|---|---|---|
| 1st Macro Call Macro Number | D2x064 | 9th Macro Call Macro Number | D2x072 |
| 2nd Macro Call Macro Number | D2x065 | 10th Macro Call Macro Number | D2x073 |
| 3rd Macro Call Macro Number | D2x066 | 11th Macro Call Macro Number | D2x074 |
| 4th Macro Call Macro Number | D2x067 | 12th Macro Call Macro Number | D2x075 |
| 5th Macro Call Macro Number | D2x068 | 13th Macro Call Macro Number | D2x076 |
| 6th Macro Call Macro Number | D2x069 | 14th Macro Call Macro Number | D2x077 |
| 7th Macro Call Macro Number | D2x070 | 15th Macro Call Macro Number | D2x078 |
| 8th Macro Call Macro Number | D2x071 | 16th Macro Call Macro Number | D2x079 |

5

### 5.3.9 MLC axis control

Aside from the NC program, JOG or INC axes control, or AUTO mode, the NC5 controller can also control axes through MLC. Users can move the axis to a specific position or set the axis to constantly move at a specific speed.

| Function Name | Special M | Function Name | Special M |
|---|---|---|---|
| MLC X Axis Control Mode | M2x416 | NC / MLC Axis Switching (C Axis) | M2x437 |
| MLC Y Axis Control Mode | M2x417 | NC / MLC Axis Switching (U Axis) | M2x438 |
| MLC Z Axis Control Mode | M2x418 | NC / MLC Axis Switching (V Axis) | M2x439 |
| MLC A Axis Control Mode | M2x419 | NC / MLC Axis Switching (W Axis) | M2x440 |
| MLC B Axis Control Mode | M2x420 | NC / MLC Axis Switching (10th Axis) | M2x441 |
| MLC C Axis Control Mode | M2x421 | NC / MLC Axis Switching (11th Axis) | M2x442 |
| MLC U Axis Control Mode | M2x422 | NC / MLC Axis Switching (12th Axis) | M2x443 |
| MLC V Axis Control Mode | M2x423 | NC / MLC Axis Switching (13th Axis) | M2x444 |
| MLC W Axis Control Mode | M2x424 | NC / MLC Axis Switching (14th Axis) | M2x445 |
| MLC 10th Axis Control Mode | M2x425 | NC / MLC Axis Switching (15th Axis) | M2x446 |
| MLC 11th Axis Control Mode | M2x426 | NC / MLC Axis Switching (16th Axis) | M2x447 |
| MLC 12th Axis Control Mode | M2x427 | Trigger X Axis' Movement (MLC Axis) | M2x448 |
| MLC 13th Axis Control Mode | M2x428 | Trigger Y Axis' Movement (MLC Axis) | M2x449 |
| MLC 14th Axis Control Mode | M2x429 | Trigger Z Axis' Movement (MLC Axis) | M2x450 |
| MLC 15th Axis Control Mode | M2x430 | Trigger A Axis' Movement (MLC Axis) | M2x451 |
| MLC 16th Axis Control Mode | M2x431 | Trigger B Axis' Movement (MLC Axis) | M2x452 |
| NC / MLC Axis Switching (X Axis) | M2x432 | Trigger C Axis' Movement (MLC Axis) | M2x453 |
| NC / MLC Axis Switching (Y Axis) | M2x433 | Trigger U Axis' Movement (MLC Axis) | M2x454 |
| NC / MLC Axis Switching (Z Axis) | M2x434 | Trigger V Axis' Movement (MLC Axis) | M2x455 |
| NC / MLC Axis Switching (A Axis) | M2x435 | Trigger W Axis' Movement (MLC Axis) | M2x456 |
| NC / MLC Axis Switching (B Axis) | M2x436 | Trigger 10th Axis' Movement (MLC Axis) | M2x457 |
| Trigger 11th Axis' Movement (MLC Axis) | M2x458 | MLC Axis Command Type of C Axis | M2x469 |
| Trigger 12th Axis' Movement (MLC Axis) | M2x459 | MLC Axis Command Type of U Axis | M2x470 |
| Trigger 13th Axis' Movement (MLC Axis) | M2x460 | MLC Axis Command Type of V Axis | M2x471 |
| Trigger 14th Axis' Movement (MLC Axis) | M2x461 | MLC Axis Command Type of W Axis | M2x472 |
| Trigger 15th Axis' Movement (MLC Axis) | M2x462 | MLC Axis Command Type of 10th Axis | M2x473 |
| Trigger 16th Axis' Movement (MLC Axis) | M2x463 | MLC Axis Command Type of 11th Axis | M2x474 |
| MLC Axis Command Type of X Axis | M2x464 | MLC Axis Command Type of 12th Axis | M2x475 |
| MLC Axis Command Type of Y Axis | M2x465 | MLC Axis Command Type of 13th Axis | M2x476 |
| MLC Axis Command Type of Z Axis | M2x466 | MLC Axis Command Type of 14th Axis | M2x477 |
| MLC Axis Command Type of A Axis | M2x467 | MLC Axis Command Type of 15th Axis | M2x478 |
| MLC Axis Command Type of B Axis | M2x468 | MLC Axis Command Type of 16th Axis | M2x479 |

| Function Name | Special M | Function Name | Special M |
|---|---|---|---|
| X Axis Switch to MLC Axis Finished | M3x432 | X Axis Target Reached (MLC Axis) | M3x448 |
| Y Axis Switch to MLC Axis Finished | M3x433 | Y Axis Target Reached (MLC Axis) | M3x449 |
| Z Axis Switch to MLC Axis Finished | M3x434 | Z Axis Target Reached (MLC Axis) | M3x450 |
| A Axis Switch to MLC Axis Finished | M3x435 | A Axis Target Reached (MLC Axis) | M3x451 |
| B Axis Switch to MLC Axis Finished | M3x436 | B Axis Target Reached (MLC Axis) | M3x452 |
| C Axis Switch to MLC Axis Finished | M3x437 | C Axis Target Reached (MLC Axis) | M3x453 |
| U Axis Switch to MLC Axis Finished | M3x438 | U Axis Target Reached (MLC Axis) | M3x454 |
| V Axis Switch to MLC Axis Finished | M3x439 | V Axis Target Reached (MLC Axis) | M3x455 |
| W Axis Switch to MLC Axis Finished | M3x440 | W Axis Target Reached (MLC Axis) | M3x456 |
| 10$^{th}$ Axis Switch to MLC Axis Finished | M3x441 | 10$^{th}$ Axis Target Reached (MLC Axis) | M3x457 |
| 11$^{th}$ Axis Switch to MLC Axis Finished | M3x442 | 11$^{th}$ Axis Target Reached (MLC Axis) | M3x458 |
| 12$^{th}$ Axis Switch to MLC Axis Finished | M3x443 | 12$^{th}$ Axis Target Reached (MLC Axis) | M3x459 |
| 13$^{th}$ Axis Switch to MLC Axis Finished | M3x444 | 13$^{th}$ Axis Target Reached (MLC Axis) | M3x460 |
| 14$^{th}$ Axis Switch to MLC Axis Finished | M3x445 | 14$^{th}$ Axis Target Reached (MLC Axis) | M3x461 |
| 15$^{th}$ Axis Switch to MLC Axis Finished | M3x446 | 15$^{th}$ Axis Target Reached (MLC Axis) | M3x462 |
| 16$^{th}$ Axis Switch to MLC Axis Finished | M3x447 | 16$^{th}$ Axis Target Reached (MLC Axis) | M3x463 |

| Function Name | Special D | Function Name | Special D |
|---|---|---|---|
| Target Position of X axis (MLC Axis) | D2x256 | Target Velocity of X axis (MLC Axis) | D2x288 |
| Target Position of Y axis (MLC Axis) | D2x258 | Target Velocity of Y axis (MLC Axis) | D2x290 |
| Target Position of Z axis (MLC Axis) | D2x260 | Target Velocity of Z axis (MLC Axis) | D2x292 |
| Target Position of A axis (MLC Axis) | D2x262 | Target Velocity of A axis (MLC Axis) | D2x294 |
| Target Position of B axis (MLC Axis) | D2x264 | Target Velocity of B axis (MLC Axis) | D2x296 |
| Target Position of C axis (MLC Axis) | D2x266 | Target Velocity of C axis (MLC Axis) | D2x298 |
| Target Position of U axis (MLC Axis) | D2x268 | Target Velocity of U axis (MLC Axis) | D2x300 |
| Target Position of V axis (MLC Axis) | D2x270 | Target Velocity of V axis (MLC Axis) | D2x302 |
| Target Position of W axis (MLC Axis) | D2x272 | Target Velocity of W axis (MLC Axis) | D2x304 |
| Target Position of 10$^{th}$ axis (MLC Axis) | D2x274 | Target Velocity of 10$^{th}$ axis (MLC Axis) | D2x306 |
| Target Position of 11$^{th}$ axis (MLC Axis) | D2x276 | Target Velocity of 11$^{th}$ axis (MLC Axis) | D2x308 |
| Target Position of 12$^{th}$ axis (MLC Axis) | D2x278 | Target Velocity of 12$^{th}$ axis (MLC Axis) | D2x310 |
| Target Position of 13$^{th}$ axis (MLC Axis) | D2x280 | Target Velocity of 13$^{th}$ axis (MLC Axis) | D2x312 |
| Target Position of 14$^{th}$ axis (MLC Axis) | D2x282 | Target Velocity of 14$^{th}$ axis (MLC Axis) | D2x314 |
| Target Position of 15$^{th}$ axis (MLC Axis) | D2x284 | Target Velocity of 15$^{th}$ axis (MLC Axis) | D2x316 |
| Target Position of 16$^{th}$ axis (MLC Axis) | D2x286 | Target Velocity of 16$^{th}$ axis (MLC Axis) | D2x318 |

### 5.3.10 M, S, T codes

When the M, S, and T codes are executed in the NC program, the NC system sends the corresponding special M signals to the MLC.

For example, when M03 is executed in the program and M3x064 is set to ON, and then the first M code command ‘3’ data will written to the corresponding special D3x048 register.

| Function Name | Special M | Function Name | Special M |
|---|---|---|---|
| M, S, and T Codes Lock | M2x011 | M Code Execution | M3x064 |
| M, S, and T Codes Finished | M2x016 | S Code Execution | M3x065 |
| - | - | T Code Execution | M3x066 |

| Function Name | Special D | Function Name | Special D |
|---|---|---|---|
| 1st M Code Data | D3x048 | 1st T Code Data | D3x052 |
| 1st S Code Data | D3x050 | - | - |

### 5.3.11 Axes synchronous control

The NC5 controller provides the synchronous axis control functions, which can let master and slave axes have same tool path and also acceleration and deceleration. The parameter **[N2.015 Synchronous master axis ID]** definitions the master axis number of each axis. Then, it is turned OFF or ON through the following special M.

| Function Name | Special M | Function Name | Special M |
|---|---|---|---|
| Enable Synchronization Adjustment | M2x014 | V Slave Axis Follows the Master Axis | M2x295 |
| Synchronous Control Enable | M2x256 | W Slave Axis Follows the Master Axis | M2x296 |
| X Slave Axis Follows the Master Axis | M2x288 | 10th Slave Axis Follows the Master Axis | M2x297 |
| Y Slave Axis Follows the Master Axis | M2x289 | 11th Slave Axis Follows the Master Axis | M2x298 |
| Z Slave Axis Follows the Master Axis | M2x290 | 12th Slave Axis Follows the Master Axis | M2x299 |
| A Slave Axis Follows the Master Axis | M2x291 | 13th Slave Axis Follows the Master Axis | M2x300 |
| B Slave Axis Follows the Master Axis | M2x292 | 14th Slave Axis Follows the Master Axis | M2x301 |
| C Slave Axis Follows the Master Axis | M2x293 | 15th Slave Axis Follows the Master Axis | M2x302 |
| U Slave Axis Follows the Master Axis | M2x294 | 16th Slave Axis Follows the Master Axis | M2x303 |

### 5.3.12 Axes command transfer control

The NC5 controller provides the transfer functions for the axis commands, which transfers the NC command originally expected to run on the master axis to the slave axis for execution. The slave axis' parameter **[N2.015 Synchronous master axis ID]** definitions of the master axis number. Then, it is turned OFF or ON through the following special M.

| Function Name | Special M | Function Name | Special M |
|---|---|---|---|
| Command Transfer Enable | M2x257 | W Axis Receives Command from The Master Axis | M2x312 |
| X Axis Receives Command from The Master Axis | M2x304 | 10th Axis Receives Command from The Master Axis | M2x313 |
| Y Axis Receives Command from The Master Axis | M2x305 | 11th Axis Receives Command from The Master Axis | M2x314 |
| Z Axis Receives Command from The Master Axis | M2x306 | 12th Axis Receives Command from The Master Axis | M2x315 |
| A Axis Receives Command from The Master Axis | M2x307 | 13th Axis Receives Command from The Master Axis | M2x316 |
| B Axis Receives Command from The Master Axis | M2x308 | 14th Axis Receives Command from The Master Axis | M2x317 |
| C Axis Receives Command from The Master Axis | M2x309 | 15th Axis Receives Command from The Master Axis | M2x318 |
| U Axis Receives Command from The Master Axis | M2x310 | 16th Axis Receives Command from The Master Axis | M2x319 |
| V Axis Receives Command from The Master Axis | M2x311 | - | - |

5

### 5.3.13 Spindle

Users 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 | M2x704 | 5th Spindle Speed Reach | M2x768 |
| 1st Spindle Reverse Rotation | M2x705 | 5th Spindle Zero Speed | M2x769 |
| 1st Spindle Positioning Control | M2x706 | 5th Spindle Positioning Control | M2x770 |
| 1st Spindle Retraction After Tapping | M2x707 | 5th Spindle Is In The Rigid Tapping | M2x771 |
| Switching C / S Axis of 1st Lathe Spindle | M2x708 | Switching C / S Axis of 5th Lathe Spindle | M2x772 |
| 1st Spindle Command Source | M2x710 | 5th Spindle Ready | M2x774 |
| 2nd Spindle Forward Rotation | M2x720 | 6th Spindle Speed Reach | M2x784 |
| 2nd Spindle Reverse Rotation | M2x721 | 6th Spindle Reverse Rotation | M2x785 |
| 2nd Spindle Positioning Control | M2x722 | 6th Spindle Positioning Control | M2x786 |
| 2nd Spindle Retraction After Tapping | M2x723 | 6th Spindle Retraction After Tapping | M2x787 |
| Switching C / S Axis of 2nd Lathe Spindle | M2x724 | Switching C / S Axis of 6th Lathe Spindle | M2x788 |
| 2nd Spindle Command Source | M2x726 | 6th Spindle Command Source | M2x790 |
| 3rd Spindle Forward Rotation | M2x736 | 7th Spindle Forward Rotation | M2x800 |
| 3rd Spindle Reverse Rotation | M2x737 | 7th Spindle Reverse Rotation | M2x801 |
| 3rd Spindle Positioning Control | M2x738 | 7th Spindle Positioning Control | M2x802 |
| 3rd Spindle Retraction After Tapping | M2x739 | 7th Spindle Retraction After Tapping | M2x803 |
| Switching C / S Axis of 3rd Lathe Spindle | M2x740 | Switching C / S Axis of 7th Lathe Spindle | M2x804 |
| 3rd Spindle Command Source | M2x742 | 7th Spindle Command Source | M2x806 |
| 4th Spindle Forward Rotation | M2x752 | 8th Spindle Forward Rotation | M2x816 |
| 4th Spindle Reverse Rotation | M2x753 | 8th Spindle Reverse Rotation | M2x817 |
| 4th Spindle Positioning Control | M2x754 | 8th Spindle Positioning Control | M2x818 |
| 4th Spindle Retraction After Tapping | M2x755 | 8th Spindle Retraction After Tapping | M2x819 |
| Switching C / S Axis of 4th Lathe Spindle | M2x756 | Switching C / S Axis of 8th Lathe Spindle | M2x820 |
| 4th Spindle Command Source | M2x758 | 8th Spindle Command Source | M2x822 |

| Function Name | Special M | Function Name | Special M |
|---|---|---|---|
| 1st Spindle Speed Reach | M3x704 | 5th Spindle Speed Reach | M3x768 |
| 1st Spindle Zero Speed | M3x705 | 5th Spindle Zero Speed | M3x769 |
| 1st Spindle Target Reach | M3x706 | 5th Spindle Positioning Control | M3x770 |
| 1st Spindle Is In The Rigid Tapping | M3x707 | 5th Spindle Is In The Rigid Tapping | M3x771 |
| 1st Spindle Is In Position Axis Mode | M3x709 | Switching C / S Axis of 5th Lathe Spindle | M3x773 |
| 1st Spindle Ready | M3x710 | 5th Spindle Ready | M3x774 |
| 2nd Spindle Speed Reach | M3x720 | 6th Spindle Speed Reach | M3x784 |
| 2nd Spindle Zero Speed | M3x721 | 6th Spindle Zero Speed | M3x785 |
| 2nd Spindle Target Reach | M3x722 | 6th Spindle Target Reach | M3x786 |
| 2nd Spindle Is In The Rigid Tapping | M3x723 | 6th Spindle Is In The Rigid Tapping | M3x787 |
| 2nd Spindle Is In Position Axis Mode | M3x725 | 6th Spindle Is In Position Axis Mode | M3x789 |
| 2nd Spindle Ready | M3x726 | 6th Spindle Ready | M3x790 |
| 3rd Spindle Speed Reach | M3x736 | 7th Spindle Speed Reach | M3x800 |
| 3rd Spindle Zero Speed | M3x737 | 7th Spindle Zero Speed | M3x801 |
| 3rd Spindle Target Reach | M3x738 | 7th Spindle Target Reach | M3x802 |
| 3rd Spindle Is In The Rigid Tapping | M3x739 | 7th Spindle Is In The Rigid Tapping | M3x803 |
| 3rd Spindle Is In Position Axis Mode | M3x741 | 7th Spindle Is In Position Axis Mode | M3x805 |
| 3rd Spindle Ready | M3x742 | 7th Spindle Ready | M3x806 |
| 4th Spindle Speed Reach | M3x752 | 8th Spindle Speed Reach | M3x816 |
| 4th Spindle Zero Speed | M3x753 | 8th Spindle Zero Speed | M3x817 |
| 4th Spindle Target Reach | M3x754 | 8th Spindle Target Reach | M3x818 |
| 4th Spindle Is In The Rigid Tapping | M3x755 | 8th Spindle Is In The Rigid Tapping | M3x819 |
| 4th Spindle Is In Position Axis Mode | M3x757 | 8th Spindle Is In Position Axis Mode | M3x821 |
| 4th Spindle Ready | M3x758 | 8th Spindle Ready | M3x822 |

| Function Name | Special D | Function Name | Special D |
|---|---|---|---|
| Spindle Analog Voltage Output Port 1 | D20160 | 5th Spindle Speed (written through the special D) | D2x332 |
| Spindle Analog Voltage Output Port 2 | D20161 | 5th Spindle Speed Rate | D2x334 |
| 1st Spindle Speed (written through the special D) | D2x024 | 5th Spindle Gear Ratio Selection | D2x335 |
| 1st Spindle Speed Rate | D2x026 | 6th Spindle Speed (written through the special D) | D2x338 |
| 1st Spindle Gear Ratio Selection | D2x027 | 6th Spindle Speed Rate | D2x340 |
| 2nd Spindle Speed (written through the special D) | D2x030 | 6th Spindle Gear Ratio Selection | D2x341 |
| 2nd Spindle Speed Rate | D2x032 | 7th Spindle Speed (written through the special D) | D2x344 |
| 2nd Spindle Gear Ratio Selection | D2x033 | 7th Spindle Speed Rate | D2x346 |
| 3rd Spindle Speed (written through the special D) | D2x320 | 7th Spindle Gear Ratio Selection | D2x347 |
| 3rd Spindle Speed Rate | D2x322 | 8th Spindle Speed (written through the special D) | D2x350 |
| 3rd Spindle Gear Ratio Selection | D2x323 | 8th Spindle Speed Rate | D2x352 |
| 4th Spindle Speed (written through the special D) | D2x326 | 8th Spindle Gear Ratio Selection | D2x353 |
| 4th Spindle Speed Rate | D2x328 | - | - |
| 4th Spindle Gear Ratio Selection | D2x329 | - | - |

| Function Name | Special D | Function Name | Special D |
|---|---|---|---|
| Speed Command of 1st Spindle | D3x024<br>D3x025 | Actual Feed of 2nd Spindle | D3x360<br>D3x361 |
| 1st Spindle Speed Feedback | D3x026<br>D3x027 | Speed Command of 3rd Spindle | D3x362<br>D3x363 |
| 1st Spindle Actual Degree | D3x028<br>D3x029 | 3rd Spindle Speed Feedback | D3x364<br>D3x365 |
| Target Feed of 1st Spindle | D3x354<br>D3x355 | 3rd Spindle Actual Degree | D3x366<br>D3x367 |
| Actual Feed of 1st Spindle | D3x358<br>D3x359 | Tapping Error of 3rd Spindle | D3x368<br>D3x369 |
| Speed Command of 2nd Spindle | D3x030<br>D3x031 | Target Feed of 3rd Spindle | D3x370<br>D3x371 |
| 2nd Spindle Speed Feedback | D3x032<br>D3x033 | Actual Feed of 3rd Spindle | D3x372<br>D3x373 |
| 2nd Spindle Actual Degree | D3x034<br>D3x035 | Speed Command of 4th Spindle | D3x374<br>D3x375 |
| Tapping Error of 1st Spindle | D3x350<br>D3x351 | 4th Spindle Speed Feedback | D3x376<br>D3x377 |
| Tapping Error of 2nd Spindle | D3x352<br>D3x353 | 4th Spindle Actual Degree | D3x378<br>D3x379 |
| Target Feed of 2nd Spindle | D3x356<br>D3x357 | Tapping Error of 4th Spindle | D3x380<br>D3x381 |
| Target Feed of 4th Spindle | D3x382<br>D3x383 | Speed Command of 5th Spindle | D3x386<br>D3x387 |
| Actual Feed of 4th Spindle | D3x384<br>D3x385 | 5th Spindle Speed Feedback | D3x388<br>D3x389 |
| 5th Spindle Actual Degree | D3x390<br>D3x391 | 7th Spindle Speed Feedback | D3x412<br>D3x413 |
| Target Feed of 5th Spindle | D3x394<br>D3x395 | 7th Spindle Actual Degree | D3x414<br>D3x415 |
| Actual Feed of 5th Spindle | D3x396<br>D3x397 | Target Feed of 7th Spindle | D3x418<br>D3x419 |
| Speed Command of 6th Spindle | D3x398<br>D3x399 | Actual Feed of 7th Spindle | D3x420<br>D3x421 |
| 6th Spindle Speed Feedback | D3x400<br>D3x401 | Speed Command of 8th Spindle | D3x422<br>D3x423 |
| 6th Spindle Actual Degree | D3x402<br>D3x403 | 8th Spindle Speed Feedback | D3x424<br>D3x425 |
| Target Feed of 6th Spindle | D3x406<br>D3x407 | 8th Spindle Actual Degree | D3x426<br>D3x427 |
| Actual Feed of 6th Spindle | D3x408<br>D3x409 | Target Feed of 8th Spindle | D3x430<br>D3x431 |
| Speed Command of 7th Spindle | D3x410<br>D3x411 | Actual Feed of 8th Spindle | D3x432<br>D3x433 |

5

### 5.3.14 Tool magazine

The NC5 controller provides tool magazine management functions. Aside from using macros for tool changes, users also 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 Move Forward | M2x064 | Tool Magazine 2 Move Forward | M2x072 |
| Tool Magazine 1 Move Backward | M2x065 | Tool Magazine 2 Move Backward | M2x073 |
| Tool 1 Exchange | M2x066 | Tool 2 Exchange | M2x074 |
| Tool Magazine 1 Reset | M2x067 | Tool Magazine 2 Reset | M2x075 |

| Function Name | Special D | Function Name | Special D |
|---|---|---|---|
| Tool Magazine 1 Tool Number | D2x036 | Tool Magazine 2 Tool Number | D2x042 |
| Tool Magazine 1 Standby Tool Number | D2x037 | Tool Magazine 2 Standby Tool Number | D2x043 |
| Tool Magazine 1 Command Tool Number | D2x038 | Tool Magazine 2 Command Tool Number | D2x044 |

| Function Name | Special D | Function Name | Special D |
|---|---|---|---|
| Tool Magazine 1 Current Tool Number | D3x036 | Tool Magazine 2 Current Tool Number | D3x042 |
| Tool Magazine 1 Standby Tool Number | D3x037 | Tool Magazine 2 Standby Tool Number | D3x043 |
| Tool Magazine 1 Standby Tool Pot | D3x038 | Tool Magazine 2 Standby Tool Pot | D3x044 |
| Tool Magazine 1 Tool Pot Deviation | D3x039 | Tool Magazine 2 Tool Pot Deviation | D3x045 |

5

### 5.3.15 NC system

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

| Function Name | Special M | Function Name | Special M |
|---|---|---|---|
| NC Reset | M2x004 | Time Pulse (500 ms) | M29032 |
| Servo On / Off | M2x012 | Time Pulse (50 ms) | M29033 |
| Enable Emergency Stop | M2x013 | Remaining OFF | M29034 |
| - | - | Remaining ON | M29035 |

| Function Name | Special M | Function Name | Special M |
|---|---|---|---|
| EIO 1 Connection Status | M30032 | Emergency Stop Status | M3x018 |
| EIO 2 Connection Status | M30033 | Reset Finished | M3x019 |
| EIO 3 Connection Status | M30034 | NC Error | M3x025 |
| EIO 4 Connection Status | M30035 | System Mode Switching | M3x032 |
| EIO 5 Connection Status | M30036 | Main Program Lock | M3x033 |
| EIO 6 Connection Status | M30037 | Servo Drive Error | M3x034 |
| EIO 7 Connection Status | M30038 | System Ready and Servo ON | M3x036 |
| EIO 8 Connection Status | M30039 | User Define Alarm Status | M3x040 |
| EIO 9 Connection Status | M30040 | Servo Connection Status of X Axis | M3x256 |
| EIO 10 Connection Status | M30041 | Servo Connection Status of Y Axis | M3x257 |
| EIO 11 Connection Status | M30042 | Servo Connection Status of Z Axis | M3x258 |
| EIO 12 Connection Status | M30043 | Servo Connection Status of A Axis | M3x259 |
| EIO 13 Connection Status | M30044 | Servo Connection Status of B Axis | M3x260 |
| EIO 14 Connection Status | M30045 | Servo Connection Status of C Axis | M3x261 |
| EIO 15 Connection Status | M30046 | Servo Connection Status of U Axis | M3x262 |
| EIO 16 Connection Status | M30047 | Servo Connection Status of V Axis | M3x263 |
| EIO 17 Connection Status | M30048 | Servo Connection Status of W Axis | M3x264 |
| EIO 18 Connection Status | M30049 | Servo Connection Status of 10$^{th}$ Axis | M3x265 |
| EIO 19 Connection Status | M30050 | Servo Connection Status of 11$^{th}$ Axis | M3x266 |
| EIO 20 Connection Status | M30051 | Servo Connection Status of 12$^{th}$ Axis | M3x267 |
| NC Channel 1 Enable Status | M30080 | Servo Connection Status of 13$^{th}$ Axis | M3x268 |
| NC Channel 2 Enable Status | M30081 | Servo Connection Status of 14$^{th}$ Axis | M3x269 |
| NC Channel 3 Enable Status | M30082 | Servo Connection Status of 15$^{th}$ Axis | M3x270 |
| NC Channel 4 Enable Status | M30083 | Servo Connection Status of 16$^{th}$ Axis | M3x271 |
| - | - | Program Lock | M49000 |

5

| Function Name | Special D | Function Name | Special D |
|---|---|---|---|
| NC Channel Selection | D20000 | Torque Feedback of 25th Axis | D30264 |
| Torque Feedback of 1st Axis | D30240 | Torque Feedback of 26th Axis | D30265 |
| Torque Feedback of 2nd Axis | D30241 | Torque Feedback of 27th Axis | D30266 |
| Torque Feedback of 3rd Axis | D30242 | Torque Feedback of 28th Axis | D30267 |
| Torque Feedback of 4th Axis | D30243 | Torque Feedback of 29th Axis | D30268 |
| Torque Feedback of 5th Axis | D30244 | Torque Feedback of 30th Axis | D30269 |
| Torque Feedback of 6th Axis | D30245 | Torque Feedback of 31th Axis | D30270 |
| Torque Feedback of 7th Axis | D30246 | Torque Feedback of 32th Axis | D30271 |
| Torque Feedback of 8th Axis | D30247 | Velocity Feedback of 1st Axis (mm/min) | D30272<br>D30273 |
| Torque Feedback of 9th Axis | D30248 | Velocity Feedback of 2nd Axis (mm/min) | D30274<br>D30275 |
| Torque Feedback of 10th Axis | D30249 | Velocity Feedback of 3rd Axis (mm/min) | D30276<br>D30277 |
| Torque Feedback of 11th Axis | D30250 | Velocity Feedback of 4th Axis (mm/min) | D30278<br>D30279 |
| Torque Feedback of 12th Axis | D30251 | Velocity Feedback of 5th Axis (mm/min) | D30280<br>D30281 |
| Torque Feedback of 13th Axis | D30252 | Velocity Feedback of 6th Axis (mm/min) | D30282<br>D30283 |
| Torque Feedback of 14th Axis | D30253 | Velocity Feedback of 7th Axis (mm/min) | D30284<br>D30285 |
| Torque Feedback of 15th Axis | D30254 | Velocity Feedback of 8th Axis (mm/min) | D30286<br>D30287 |
| Torque Feedback of 16th Axis | D30255 | Velocity Feedback of 9th Axis (mm/min) | D30288<br>D30289 |
| Torque Feedback of 17th Axis | D30256 | Velocity Feedback of 10th Axis (mm/min) | D30290<br>D30291 |
| Torque Feedback of 18th Axis | D30257 | Velocity Feedback of 11th Axis (mm/min) | D30292<br>D30293 |
| Torque Feedback of 19th Axis | D30258 | Velocity Feedback of 12th Axis (mm/min) | D30294<br>D30295 |
| Torque Feedback of 20th Axis | D30259 | Velocity Feedback of 13th Axis (mm/min) | D30296<br>D30297 |
| Torque Feedback of 21th Axis | D30260 | Velocity Feedback of 14th Axis (mm/min) | D30298<br>D30299 |
| Torque Feedback of 22th Axis | D30261 | Velocity Feedback of 15th Axis (mm/min) | D30300<br>D30301 |
| Torque Feedback of 23th axis | D30262 | Velocity Feedback of 16th Axis (mm/min) | D30302<br>D30303 |
| Torque Feedback of 24th Axis | D30263 | Velocity Feedback of 17th Axis (mm/min) | D30304<br>D30305 |
| Velocity Feedback of 18th Axis (mm/min) | D30306<br>D30307 | Velocity Feedback of 9th Axis (RPM) | D30352<br>D30353 |
| Velocity Feedback of 19th Axis (mm/min) | D30308<br>D30309 | Velocity Feedback of 10th Axis (RPM) | D30354<br>D30355 |
| Velocity Feedback of 20th Axis (mm/min) | D30310<br>D30311 | Velocity Feedback of 11th Axis (RPM) | D30356<br>D30357 |
| Velocity Feedback of 21th Axis (mm/min) | D30312<br>D30313 | Velocity Feedback of 12th Axis (RPM) | D30358<br>D30359 |

| Function Name | Special D | Function Name | Special D |
|---|---|---|---|
| Velocity Feedback of 22th Axis (mm/min) | D30314<br>D30315 | Velocity Feedback of 13th Axis (RPM) | D30360<br>D30361 |
| Velocity Feedback of 23th Axis (mm/min) | D30316<br>D30317 | Velocity Feedback of 14th Axis (RPM) | D30362<br>D30363 |
| Velocity Feedback of 24th Axis (mm/min) | D30318<br>D30319 | Velocity Feedback of 15th Axis (RPM) | D30364<br>D30365 |
| Velocity Feedback of 25th Axis (mm/min) | D30320<br>D30321 | Velocity Feedback of 16th Axis (RPM) | D30366<br>D30367 |
| Velocity Feedback of 26th Axis (mm/min) | D30322<br>D30323 | Velocity Feedback of 17th Axis (RPM) | D30368<br>D30369 |
| Velocity Feedback of 27th Axis (mm/min) | D30324<br>D30325 | Velocity Feedback of 18th Axis (RPM) | D30360<br>D30371 |
| Velocity Feedback of 28th Axis (mm/min) | D30326<br>D30327 | Velocity Feedback of 19th Axis (RPM) | D30372<br>D30373 |
| Velocity Feedback of 29th Axis (mm/min) | D30328<br>D30329 | Velocity Feedback of 20th Axis (RPM) | D30374<br>D30375 |
| Velocity Feedback of 30th Axis (mm/min) | D30330<br>D30331 | Velocity Feedback of 21th Axis (RPM) | D30376<br>D30377 |
| Velocity Feedback of 31th Axis (mm/min) | D30332<br>D30333 | Velocity Feedback of 22th Axis (RPM) | D30378<br>D30379 |
| Velocity Feedback of 32th Axis (mm/min) | D30334<br>D30335 | Velocity Feedback of 23th Axis (RPM) | D30380<br>D30381 |
| Velocity Feedback of 1st Axis (RPM) | D30336<br>D30337 | Velocity Feedback of 24th Axis (RPM) | D30382<br>D30383 |
| Velocity Feedback of 2nd Axis (RPM) | D30338<br>D30339 | Velocity Feedback of 25th Axis (RPM) | D30384<br>D30385 |
| Velocity Feedback of 3rd Axis (RPM) | D30340<br>D30341 | Velocity Feedback of 26th Axis (RPM) | D30386<br>D30387 |
| Velocity Feedback of 4th Axis (RPM) | D30342<br>D30343 | Velocity Feedback of 27th Axis (RPM) | D30388<br>D30389 |
| Velocity Feedback of 5th Axis (RPM) | D30344<br>D30345 | Velocity Feedback of 28th Axis (RPM) | D30390<br>D30391 |
| Velocity Feedback of 6th Axis (RPM) | D30346<br>D30347 | Velocity Feedback of 29th Axis (RPM) | D30392<br>D30393 |
| Velocity Feedback of 7th Axis (RPM) | D30348<br>D30349 | Velocity Feedback of 30th Axis (RPM) | D30394<br>D30395 |
| Velocity Feedback of 8th Axis (RPM) | D30350<br>D30351 | Velocity Feedback of 31th Axis (RPM) | D30396<br>D30397 |
| Velocity Feedback of 32th Axis (RPM) | D30398<br>D30399 | Torque Peak of 22th Axis | D30421 |
| Torque Peak of 1st Axis | D30400 | Torque Peak of 23th Axis | D30422 |
| Torque Peak of 2nd Axis | D30401 | Torque Peak of 24th Axis | D30423 |
| Torque Peak of 3rd Axis | D30402 | Torque Peak of 25th Axis | D30424 |
| Torque Peak of 4th Axis | D30403 | Torque Peak of 26th Axis | D30425 |
| Torque Peak of 5th Axis | D30404 | Torque Peak of 27th Axis | D30426 |
| Torque Peak of 6th Axis | D30405 | Torque Peak of 28th Axis | D30427 |
| Torque Peak of 7th Axis | D30406 | Torque Peak of 29th Axis | D30428 |
| Torque Peak of 8th Axis | D30407 | Torque Peak of 30th Axis | D30429 |
| Torque Peak of 9th Axis | D30408 | Torque Peak of 31th Axis | D30430 |

5

| Function Name | Special D | Function Name | Special D |
|---|---|---|---|
| Torque Peak of 10th Axis | D30409 | Torque Peak of 32th Axis | D30431 |
| Torque Peak of 11th Axis | D30410 | Current Status of Multi-Z axis | D3x014 |
| Torque Peak of 12th Axis | D30411 | Current Coordinate System | D3x016 |
| Torque Peak of 13th Axis | D30412 | Information Monitoring 1 Sort 1 | D3x096<br>D3x097 |
| Torque Peak of 14th Axis | D30413 | Information Monitoring 1 Sort 2 | D3x098<br>D3x099 |
| Torque Peak of 15th Axis | D30414 | Information Monitoring 1 Sort 3 | D3x100<br>D3x101 |
| Torque Peak of 16th Axis | D30415 | Information Monitoring 1 Sort 4 | D3x102<br>D3x103 |
| Torque Peak of 17th Axis | D30416 | Information Monitoring 2 Sort 1 | D3x104<br>D3x105 |
| Torque Peak of 18th Axis | D30417 | Information Monitoring 2 Sort 2 | D3x106<br>D3x107 |
| Torque Peak of 19th Axis | D30418 | Information Monitoring 2 Sort 3 | D3x108<br>D3x109 |
| Torque Peak of 20th Axis | D30419 | Information Monitoring 2 Sort 4 | D3x110<br>D3x111 |
| Torque Peak of 21th Axis | D30420 | Look Ahead Remaining Command | D3x320<br>D3x321 |

| Function Name | Special D | Function Name | Special D |
|---|---|---|---|
| Complete Process Amount | D49x00<br>D49x01 | Total Process Time | D49x04<br>D49x05 |
| Process Target Amount | D49x02<br>D49x03 | Single Process Time | D49x06<br>D49x07 |

5

### 5.3.16 NC read bit status from special M

Users can use dedicated NC # variables to obtain special M relays' status from the MLC system. This means these # variables below can be edited by the MLC, so users can control the NC program's conditions through the MLC program. Variables from #25000 to #25127 are paired with MLC bit type relays M2x128 to M2x255 respectively (MLC to NC). For example, #25000 in NC channel 1 is paired with M21128. If M21128 is ON, the value of the variable #25000 in the NC program of channel 1 will be 1. On the other hand, if M21128 is OFF, the value of #25000 will be 0.

Refer to the following table for the bits for the output points in the MLC system and their corresponding # variables in the NC system.

| Function Name | Special M | # variable | Function Name | Special M | # variable |
|---|---|---|---|---|---|
| MLC to NC Variable 1 | M2x128 | #25000 | MLC to NC Variable 33 | M2x160 | #25032 |
| MLC to NC Variable 2 | M2x129 | #25001 | MLC to NC Variable 34 | M2x161 | #25033 |
| MLC to NC Variable 3 | M2x130 | #25002 | MLC to NC Variable 35 | M2x162 | #25034 |
| MLC to NC Variable 4 | M2x131 | #25003 | MLC to NC Variable 36 | M2x163 | #25035 |
| MLC to NC Variable 5 | M2x132 | #25004 | MLC to NC Variable 37 | M2x164 | #25036 |
| MLC to NC Variable 6 | M2x133 | #25005 | MLC to NC Variable 38 | M2x165 | #25037 |
| MLC to NC Variable 7 | M2x134 | #25006 | MLC to NC Variable 39 | M2x166 | #25038 |
| MLC to NC Variable 8 | M2x135 | #25007 | MLC to NC Variable 40 | M2x167 | #25039 |
| MLC to NC Variable 9 | M2x136 | #25008 | MLC to NC Variable 41 | M2x168 | #25040 |
| MLC to NC Variable 10 | M2x137 | #25009 | MLC to NC Variable 42 | M2x169 | #25041 |
| MLC to NC Variable 11 | M2x138 | #25010 | MLC to NC Variable 43 | M2x170 | #25042 |
| MLC to NC Variable 12 | M2x139 | #25011 | MLC to NC Variable 44 | M2x171 | #25043 |
| MLC to NC Variable 13 | M2x140 | #25012 | MLC to NC Variable 45 | M2x172 | #25044 |
| MLC to NC Variable 14 | M2x141 | #25013 | MLC to NC Variable 46 | M2x173 | #25045 |
| MLC to NC Variable 15 | M2x142 | #25014 | MLC to NC Variable 47 | M2x174 | #25046 |
| MLC to NC Variable 16 | M2x143 | #25015 | MLC to NC Variable 48 | M2x175 | #25047 |
| MLC to NC Variable 17 | M2x144 | #25016 | MLC to NC Variable 49 | M2x176 | #25048 |
| MLC to NC Variable 18 | M2x145 | #25017 | MLC to NC Variable 50 | M2x177 | #25049 |
| MLC to NC Variable 19 | M2x146 | #25018 | MLC to NC Variable 51 | M2x178 | #25050 |
| MLC to NC Variable 20 | M2x147 | #25019 | MLC to NC Variable 52 | M2x179 | #25051 |
| MLC to NC Variable 21 | M2x148 | #25020 | MLC to NC Variable 53 | M2x180 | #25052 |
| MLC to NC Variable 22 | M2x149 | #25021 | MLC to NC Variable 54 | M2x181 | #25053 |
| MLC to NC Variable 23 | M2x150 | #25022 | MLC to NC Variable 55 | M2x182 | #25054 |
| MLC to NC Variable 24 | M2x151 | #25023 | MLC to NC Variable 56 | M2x183 | #25055 |
| MLC to NC Variable 25 | M2x152 | #25024 | MLC to NC Variable 57 | M2x184 | #25056 |
| MLC to NC Variable 26 | M2x153 | #25025 | MLC to NC Variable 58 | M2x185 | #25057 |
| MLC to NC Variable 27 | M2x154 | #25026 | MLC to NC Variable 59 | M2x186 | #25058 |
| MLC to NC Variable 28 | M2x155 | #25027 | MLC to NC Variable 60 | M2x187 | #25059 |
| MLC to NC Variable 29 | M2x156 | #25028 | MLC to NC Variable 61 | M2x188 | #25060 |
| MLC to NC Variable 30 | M2x157 | #25029 | MLC to NC Variable 62 | M2x189 | #25061 |
| MLC to NC Variable 31 | M2x158 | #25030 | MLC to NC Variable 63 | M2x190 | #25062 |
| MLC to NC Variable 32 | M2x159 | #25031 | MLC to NC Variable 64 | M2x191 | #25063 |
| MLC to NC Variable 65 | M2x192 | #25064 | MLC to NC Variable 97 | M2x224 | #25096 |
| MLC to NC Variable 66 | M2x193 | #25065 | MLC to NC Variable 98 | M2x225 | #25097 |

| Function Name | Special M | # variable | Function Name | Special M | # variable |
|---|---|---|---|---|---|
| MLC to NC Variable 67 | M2x194 | #25066 | MLC to NC Variable 99 | M2x226 | #25098 |
| MLC to NC Variable 68 | M2x195 | #25067 | MLC to NC Variable 100 | M2x227 | #25099 |
| MLC to NC Variable 69 | M2x196 | #25068 | MLC to NC Variable 101 | M2x228 | #25100 |
| MLC to NC Variable 70 | M2x197 | #25069 | MLC to NC Variable 102 | M2x229 | #25101 |
| MLC to NC Variable 71 | M2x198 | #25070 | MLC to NC Variable 103 | M2x230 | #25102 |
| MLC to NC Variable 72 | M2x199 | #25071 | MLC to NC Variable 104 | M2x231 | #25103 |
| MLC to NC Variable 73 | M2x200 | #25072 | MLC to NC Variable 105 | M2x232 | #25104 |
| MLC to NC Variable 74 | M2x201 | #25073 | MLC to NC Variable 106 | M2x233 | #25105 |
| MLC to NC Variable 75 | M2x202 | #25074 | MLC to NC Variable 107 | M2x234 | #25106 |
| MLC to NC Variable 76 | M2x203 | #25075 | MLC to NC Variable 108 | M2x235 | #25107 |
| MLC to NC Variable 77 | M2x204 | #25076 | MLC to NC Variable 109 | M2x236 | #25108 |
| MLC to NC Variable 78 | M2x205 | #25077 | MLC to NC Variable 110 | M2x237 | #25109 |
| MLC to NC Variable 79 | M2x206 | #25078 | MLC to NC Variable 111 | M2x238 | #25110 |
| MLC to NC Variable 80 | M2x207 | #25079 | MLC to NC Variable 112 | M2x239 | #25111 |
| MLC to NC Variable 81 | M2x208 | #25080 | MLC to NC Variable 113 | M2x240 | #25112 |
| MLC to NC Variable 82 | M2x209 | #25081 | MLC to NC Variable 114 | M2x241 | #25113 |
| MLC to NC Variable 83 | M2x210 | #25082 | MLC to NC Variable 115 | M2x242 | #25114 |
| MLC to NC Variable 84 | M2x211 | #25083 | MLC to NC Variable 116 | M2x243 | #25115 |
| MLC to NC Variable 85 | M2x212 | #25084 | MLC to NC Variable 117 | M2x244 | #25116 |
| MLC to NC Variable 86 | M2x213 | #25085 | MLC to NC Variable 118 | M2x245 | #25117 |
| MLC to NC Variable 87 | M2x214 | #25086 | MLC to NC Variable 119 | M2x246 | #25118 |
| MLC to NC Variable 88 | M2x215 | #25087 | MLC to NC Variable 120 | M2x247 | #25119 |
| MLC to NC Variable 89 | M2x216 | #25088 | MLC to NC Variable 121 | M2x248 | #25120 |
| MLC to NC Variable 90 | M2x217 | #25089 | MLC to NC Variable 122 | M2x249 | #25121 |
| MLC to NC Variable 91 | M2x218 | #25090 | MLC to NC Variable 123 | M2x250 | #25122 |
| MLC to NC Variable 92 | M2x219 | #25091 | MLC to NC Variable 124 | M2x251 | #25123 |
| MLC to NC Variable 93 | M2x220 | #25092 | MLC to NC Variable 125 | M2x252 | #25124 |
| MLC to NC Variable 94 | M2x221 | #25093 | MLC to NC Variable 126 | M2x253 | #25125 |
| MLC to NC Variable 95 | M2x222 | #25094 | MLC to NC Variable 127 | M2x254 | #25126 |
| MLC to NC Variable 96 | M2x223 | #25095 | MLC to NC Variable 128 | M2x255 | #25127 |

5

## 5.3.17 NC write bit control to special M

Users can use dedicated NC # variables to write the bit status to special M registers from the NC system. This means these # variables below can control the MLC directly, so users can set MLC conditions for specific purposes. Variables from #25256 to #25383 are paired with M relays M3x128 to M3x255 respectively (NC to MLC).

For example, #25256 in NC channel 1 is paired with M31128. If #25256 is set to 1 in the NC program of channel 1, the value of the M31128 relay in the MLC will be ON. The M31128 will be updated right after #25256 has changed.

Refer to the following table for the bits for the input points in the MLC system and their corresponding # variables in the NC system.

| Function Name | Special M | # variable | Function Name | Special M | # variable |
|---|---|---|---|---|---|
| NC Variable to MLC 1 | M3x128 | #25256 | NC Variable to MLC 33 | M3x160 | #25288 |
| NC Variable to MLC 2 | M3x129 | #25257 | NC Variable to MLC 34 | M3x161 | #25289 |
| NC Variable to MLC 3 | M3x130 | #25258 | NC Variable to MLC 35 | M3x162 | #25290 |
| NC Variable to MLC 4 | M3x131 | #25259 | NC Variable to MLC 36 | M3x163 | #25291 |
| NC Variable to MLC 5 | M3x132 | #25260 | NC Variable to MLC 37 | M3x164 | #25292 |
| NC Variable to MLC 6 | M3x133 | #25261 | NC Variable to MLC 38 | M3x165 | #25293 |
| NC Variable to MLC 7 | M3x134 | #25262 | NC Variable to MLC 39 | M3x166 | #25294 |
| NC Variable to MLC 8 | M3x135 | #25263 | NC Variable to MLC 40 | M3x167 | #25295 |
| NC Variable to MLC 9 | M3x136 | #25264 | NC Variable to MLC 41 | M3x168 | #25296 |
| NC Variable to MLC 10 | M3x137 | #25265 | NC Variable to MLC 42 | M3x169 | #25297 |
| NC Variable to MLC 11 | M3x138 | #25266 | NC Variable to MLC 43 | M3x170 | #25298 |
| NC Variable to MLC 12 | M3x139 | #25267 | NC Variable to MLC 44 | M3x171 | #25299 |
| NC Variable to MLC 13 | M3x140 | #25268 | NC Variable to MLC 45 | M3x172 | #25300 |
| NC Variable to MLC 14 | M3x141 | #25269 | NC Variable to MLC 46 | M3x173 | #25301 |
| NC Variable to MLC 15 | M3x142 | #25270 | NC Variable to MLC 47 | M3x174 | #25302 |
| NC Variable to MLC 16 | M3x143 | #25271 | NC Variable to MLC 48 | M3x175 | #25303 |
| NC Variable to MLC 17 | M3x144 | #25272 | NC Variable to MLC 49 | M3x176 | #25304 |
| NC Variable to MLC 18 | M3x145 | #25273 | NC Variable to MLC 50 | M3x177 | #25305 |
| NC Variable to MLC 19 | M3x146 | #25274 | NC Variable to MLC 51 | M3x178 | #25306 |
| NC Variable to MLC 20 | M3x147 | #25275 | NC Variable to MLC 52 | M3x179 | #25307 |
| NC Variable to MLC 21 | M3x148 | #25276 | NC Variable to MLC 53 | M3x180 | #25308 |
| NC Variable to MLC 22 | M3x149 | #25277 | NC Variable to MLC 54 | M3x181 | #25309 |
| NC Variable to MLC 23 | M3x150 | #25278 | NC Variable to MLC 55 | M3x182 | #25310 |
| NC Variable to MLC 24 | M3x151 | #25279 | NC Variable to MLC 56 | M3x183 | #25311 |
| NC Variable to MLC 25 | M3x152 | #25280 | NC Variable to MLC 57 | M3x184 | #25312 |
| NC Variable to MLC 26 | M3x153 | #25281 | NC Variable to MLC 58 | M3x185 | #25313 |
| NC Variable to MLC 27 | M3x154 | #25282 | NC Variable to MLC 59 | M3x186 | #25314 |
| NC Variable to MLC 28 | M3x155 | #25283 | NC Variable to MLC 60 | M3x187 | #25315 |
| NC Variable to MLC 29 | M3x156 | #25284 | NC Variable to MLC 61 | M3x188 | #25316 |
| NC Variable to MLC 30 | M3x157 | #25285 | NC Variable to MLC 62 | M3x189 | #25317 |
| NC Variable to MLC 31 | M3x158 | #25286 | NC Variable to MLC 63 | M3x190 | #25318 |
| NC Variable to MLC 32 | M3x159 | #25287 | NC Variable to MLC 64 | M3x191 | #25319 |
| NC Variable to MLC 65 | M3x192 | #25320 | NC Variable to MLC 97 | M3x224 | #25352 |

| Function Name | Special M | # variable | Function Name | Special M | # variable |
|---|---|---|---|---|---|
| NC Variable to MLC 66 | M3x193 | #25321 | NC Variable to MLC 98 | M3x225 | #25353 |
| NC Variable to MLC 67 | M3x194 | #25322 | NC Variable to MLC 99 | M3x226 | #25354 |
| NC Variable to MLC 68 | M3x195 | #25323 | NC Variable to MLC 100 | M3x227 | #25355 |
| NC Variable to MLC 69 | M3x196 | #25324 | NC Variable to MLC 101 | M3x228 | #25356 |
| NC Variable to MLC 70 | M3x197 | #25325 | NC Variable to MLC 102 | M3x229 | #25357 |
| NC Variable to MLC 71 | M3x198 | #25326 | NC Variable to MLC 103 | M3x230 | #25358 |
| NC Variable to MLC 72 | M3x199 | #25327 | NC Variable to MLC 104 | M3x231 | #25359 |
| NC Variable to MLC 73 | M3x200 | #25328 | NC Variable to MLC 105 | M3x232 | #25360 |
| NC Variable to MLC 74 | M3x201 | #25329 | NC Variable to MLC 106 | M3x233 | #25361 |
| NC Variable to MLC 75 | M3x202 | #25330 | NC Variable to MLC 107 | M3x234 | #25362 |
| NC Variable to MLC 76 | M3x203 | #25331 | NC Variable to MLC 108 | M3x235 | #25363 |
| NC Variable to MLC 77 | M3x204 | #25332 | NC Variable to MLC 109 | M3x236 | #25364 |
| NC Variable to MLC 78 | M3x205 | #25333 | NC Variable to MLC 110 | M3x237 | #25365 |
| NC Variable to MLC 79 | M3x206 | #25334 | NC Variable to MLC 111 | M3x238 | #25366 |
| NC Variable to MLC 80 | M3x207 | #25335 | NC Variable to MLC 112 | M3x239 | #25367 |
| NC Variable to MLC 81 | M3x208 | #25336 | NC Variable to MLC 113 | M3x240 | #25368 |
| NC Variable to MLC 82 | M3x209 | #25337 | NC Variable to MLC 114 | M3x241 | #25369 |
| NC Variable to MLC 83 | M3x210 | #25338 | NC Variable to MLC 115 | M3x242 | #25370 |
| NC Variable to MLC 84 | M3x211 | #25339 | NC Variable to MLC 116 | M3x243 | #25371 |
| NC Variable to MLC 85 | M3x212 | #25340 | NC Variable to MLC 117 | M3x244 | #25372 |
| NC Variable to MLC 86 | M3x213 | #25341 | NC Variable to MLC 118 | M3x245 | #25373 |
| NC Variable to MLC 87 | M3x214 | #25342 | NC Variable to MLC 119 | M3x246 | #25374 |
| NC Variable to MLC 88 | M3x215 | #25343 | NC Variable to MLC 120 | M3x247 | #25375 |
| NC Variable to MLC 89 | M3x216 | #25344 | NC Variable to MLC 121 | M3x248 | #25376 |
| NC Variable to MLC 90 | M3x217 | #25345 | NC Variable to MLC 122 | M3x249 | #25377 |
| NC Variable to MLC 91 | M3x218 | #25346 | NC Variable to MLC 123 | M3x250 | #25378 |
| NC Variable to MLC 92 | M3x219 | #25347 | NC Variable to MLC 124 | M3x251 | #25379 |
| NC Variable to MLC 93 | M3x220 | #25348 | NC Variable to MLC 125 | M3x252 | #25380 |
| NC Variable to MLC 94 | M3x221 | #25349 | NC Variable to MLC 126 | M3x253 | #25381 |
| NC Variable to MLC 95 | M3x222 | #25350 | NC Variable to MLC 127 | M3x254 | #25382 |
| NC Variable to MLC 96 | M3x223 | #25351 | NC Variable to MLC 128 | M3x255 | #25383 |

5

## 5.3.18 NC read data status from special D

Users can use dedicated NC # variables to obtain special D registers' data from the MLC system. This means these # variables below can be edited by the MLC, so users can control the NC program's variables through the MLC program. Variables from #25128 to #25255 are paired with MLC word data type register D2x128 to D2x255 respectively (MLC to NC).

For example, #25128 in NC channel 1 is paired with D21128. If D21128 set to 100, the value of the variable #25128 in the NC program of channel 1 will be 100. The variable #25128 will be updated right after D21128 has changed.

Refer to the following table for the bits for the registers in the MLC system and their corresponding # variables in the NC system.

| Function Name | Special D | # variable | Function Name | Special D | # variable |
|---|---|---|---|---|---|
| MLC to NC Variable 1 | D2x128 | #25128 | MLC to NC Variable 33 | D2x160 | #25160 |
| MLC to NC Variable 2 | D2x129 | #25129 | MLC to NC Variable 4 | D2x161 | #25161 |
| MLC to NC Variable 3 | D2x130 | #25130 | MLC to NC Variable 35 | D2x162 | #25162 |
| MLC to NC Variable 4 | D2x131 | #25131 | MLC to NC Variable 36 | D2x163 | #25163 |
| MLC to NC Variable 5 | D2x132 | #25132 | MLC to NC Variable 37 | D2x164 | #25164 |
| MLC to NC Variable 6 | D2x133 | #25133 | MLC to NC Variable 38 | D2x165 | #25165 |
| MLC to NC Variable 7 | D2x134 | #25134 | MLC to NC Variable 39 | D2x166 | #25166 |
| MLC to NC Variable 8 | D2x135 | #25135 | MLC to NC Variable 40 | D2x167 | #25167 |
| MLC to NC Variable 9 | D2x136 | #25136 | MLC to NC Variable 41 | D2x168 | #25168 |
| MLC to NC Variable 10 | D2x137 | #25137 | MLC to NC Variable 42 | D2x169 | #25169 |
| MLC to NC Variable 11 | D2x138 | #25138 | MLC to NC Variable 43 | D2x170 | #25170 |
| MLC to NC Variable 12 | D2x139 | #25139 | MLC to NC Variable 44 | D2x171 | #25171 |
| MLC to NC Variable 13 | D2x140 | #25140 | MLC to NC Variable 45 | D2x172 | #25172 |
| MLC to NC Variable 14 | D2x141 | #25141 | MLC to NC Variable 46 | D2x173 | #25173 |
| MLC to NC Variable 15 | D2x142 | #25142 | MLC to NC Variable 47 | D2x174 | #25174 |
| MLC to NC Variable 16 | D2x143 | #25143 | MLC to NC Variable 48 | D2x175 | #25175 |
| MLC to NC Variable 17 | D2x144 | #25144 | MLC to NC Variable 49 | D2x176 | #25176 |
| MLC to NC Variable 18 | D2x145 | #25145 | MLC to NC Variable 50 | D2x177 | #25177 |
| MLC to NC Variable 19 | D2x146 | #25146 | MLC to NC Variable 51 | D2x178 | #25178 |
| MLC to NC Variable 20 | D2x147 | #25147 | MLC to NC Variable 52 | D2x179 | #25179 |
| MLC to NC Variable 21 | D2x148 | #25148 | MLC to NC Variable 53 | D2x180 | #25180 |
| MLC to NC Variable 22 | D2x149 | #25149 | MLC to NC Variable 54 | D2x181 | #25181 |
| MLC to NC Variable 23 | D2x150 | #25150 | MLC to NC Variable 55 | D2x182 | #25182 |
| MLC to NC Variable 24 | D2x151 | #25151 | MLC to NC Variable 56 | D2x183 | #25183 |
| MLC to NC Variable 25 | D2x152 | #25152 | MLC to NC Variable 57 | D2x184 | #25184 |
| MLC to NC Variable 26 | D2x153 | #25153 | MLC to NC Variable 58 | D2x185 | #25185 |
| MLC to NC Variable 27 | D2x154 | #25154 | MLC to NC Variable 59 | D2x186 | #25186 |
| MLC to NC Variable 28 | D2x155 | #25155 | MLC to NC Variable 60 | D2x187 | #25187 |
| MLC to NC Variable 29 | D2x156 | #25156 | MLC to NC Variable 61 | D2x188 | #25188 |
| MLC to NC Variable 30 | D2x157 | #25157 | MLC to NC Variable 62 | D2x189 | #25189 |
| MLC to NC Variable 31 | D2x158 | #25158 | MLC to NC Variable 63 | D2x190 | #25190 |
| MLC to NC Variable 32 | D2x159 | #25159 | MLC to NC Variable 64 | D2x191 | #25191 |

5

| Function Name | Special D | # variable | Function Name | Special D | # variable |
|---|---|---|---|---|---|
| MLC to NC Variable 65 | D2x192 | #25192 | MLC to NC Variable 97 | D2x224 | #25224 |
| MLC to NC Variable 66 | D2x193 | #25193 | MLC to NC Variable 98 | D2x225 | #25225 |
| MLC to NC Variable 67 | D2x194 | #25194 | MLC to NC Variable 99 | D2x226 | #25226 |
| MLC to NC Variable 68 | D2x195 | #25195 | MLC to NC Variable 100 | D2x227 | #25227 |
| MLC to NC Variable 69 | D2x196 | #25196 | MLC to NC Variable 101 | D2x228 | #25228 |
| MLC to NC Variable 70 | D2x197 | #25197 | MLC to NC Variable 102 | D2x229 | #25229 |
| MLC to NC Variable 71 | D2x198 | #25198 | MLC to NC Variable 103 | D2x230 | #25230 |
| MLC to NC Variable 72 | D2x199 | #25199 | MLC to NC Variable 104 | D2x231 | #25231 |
| MLC to NC Variable 73 | D2x200 | #25200 | MLC to NC Variable 105 | D2x232 | #25232 |
| MLC to NC Variable 74 | D2x201 | #25201 | MLC to NC Variable 106 | D2x233 | #25233 |
| MLC to NC Variable 75 | D2x202 | #25202 | MLC to NC Variable 107 | D2x234 | #25234 |
| MLC to NC Variable 76 | D2x203 | #25203 | MLC to NC Variable 108 | D2x235 | #25235 |
| MLC to NC Variable 77 | D2x204 | #25204 | MLC to NC Variable 109 | D2x236 | #25236 |
| MLC to NC Variable 78 | D2x205 | #25205 | MLC to NC Variable 110 | D2x237 | #25237 |
| MLC to NC Variable 79 | D2x206 | #25206 | MLC to NC Variable 111 | D2x238 | #25238 |
| MLC to NC Variable 80 | D2x207 | #25207 | MLC to NC Variable 112 | D2x239 | #25239 |
| MLC to NC Variable 81 | D2x208 | #25208 | MLC to NC Variable 113 | D2x240 | #25240 |
| MLC to NC Variable 82 | D2x209 | #25209 | MLC to NC Variable 114 | D2x241 | #25241 |
| MLC to NC Variable 83 | D2x210 | #25210 | MLC to NC Variable 115 | D2x242 | #25242 |
| MLC to NC Variable 84 | D2x211 | #25211 | MLC to NC Variable 116 | D2x243 | #25243 |
| MLC to NC Variable 85 | D2x212 | #25212 | MLC to NC Variable 117 | D2x244 | #25244 |
| MLC to NC Variable 86 | D2x213 | #25213 | MLC to NC Variable 118 | D2x245 | #25245 |
| MLC to NC Variable 87 | D2x214 | #25214 | MLC to NC Variable 119 | D2x246 | #25246 |
| MLC to NC Variable 88 | D2x215 | #25215 | MLC to NC Variable 120 | D2x247 | #25247 |
| MLC to NC Variable 89 | D2x216 | #25216 | MLC to NC Variable 121 | D2x248 | #25248 |
| MLC to NC Variable 90 | D2x217 | #25217 | MLC to NC Variable 122 | D2x249 | #25249 |
| MLC to NC Variable 91 | D2x218 | #25218 | MLC to NC Variable 123 | D2x250 | #25250 |
| MLC to NC Variable 92 | D2x219 | #25219 | MLC to NC Variable 124 | D2x251 | #25251 |
| MLC to NC Variable 93 | D2x220 | #25220 | MLC to NC Variable 125 | D2x252 | #25252 |
| MLC to NC Variable 94 | D2x221 | #25221 | MLC to NC Variable 126 | D2x253 | #25253 |
| MLC to NC Variable 95 | D2x222 | #25222 | MLC to NC Variable 127 | D2x254 | #25254 |
| MLC to NC Variable 96 | D2x223 | #25223 | MLC to NC Variable 128 | D2x255 | #25255 |

### 5.3.19 NC write data to special M

Users can use dedicated NC # variables to write data to special D registers from the NC system. This means these # variables below can access the MLC directly, so users can send data to the MLC program for specific purposes. Variables from #25384 to #25511 are paired with MLC word data type register D3x128 to D3x255 respectively (NC to MLC).

For example, #25384 in NC channel 1 is paired with D31128. If the #25384 in the NC program of channel 1 set to 101, the value of the register D31128 in the MLC will be 101. The D31128 will be updated right after #25384 has changed.

When the system parameter **[N1.010 Bit7 MLC variable type]** is 1 for MLC double words floating mode, these odd D registers and paired # variables will be occupied. The system will return alarm code 0x235 if they are used in the program.

| Function Name | Special D | # variable | Function Name | Special D | # variable |
|---|---|---|---|---|---|
| NC Variable to MLC 1 | D3x128 | #25384 | NC Variable to MLC 33 | D3x160 | #25416 |
| NC Variable to MLC 2 | D3x129 | #25385 | NC Variable to MLC 34 | D3x161 | #25417 |
| NC Variable to MLC 3 | D3x130 | #25386 | NC Variable to MLC 35 | D3x162 | #25418 |
| NC Variable to MLC 4 | D3x131 | #25387 | NC Variable to MLC 36 | D3x163 | #25419 |
| NC Variable to MLC 5 | D3x132 | #25388 | NC Variable to MLC 37 | D3x164 | #25420 |
| NC Variable to MLC 6 | D3x133 | #25389 | NC Variable to MLC 38 | D3x165 | #25421 |
| NC Variable to MLC 7 | D3x134 | #25390 | NC Variable to MLC 39 | D3x166 | #25422 |
| NC Variable to MLC 8 | D3x135 | #25391 | NC Variable to MLC 40 | D3x167 | #25423 |
| NC Variable to MLC 9 | D3x136 | #25392 | NC Variable to MLC 41 | D3x168 | #25424 |
| NC Variable to MLC 10 | D3x137 | #25393 | NC Variable to MLC 42 | D3x169 | #25425 |
| NC Variable to MLC 11 | D3x138 | #25394 | NC Variable to MLC 43 | D3x170 | #25426 |
| NC Variable to MLC 12 | D3x139 | #25395 | NC Variable to MLC 44 | D3x171 | #25427 |
| NC Variable to MLC 13 | D3x140 | #25396 | NC Variable to MLC 45 | D3x172 | #25428 |
| NC Variable to MLC 14 | D3x141 | #25397 | NC Variable to MLC 46 | D3x173 | #25429 |
| NC Variable to MLC 15 | D3x142 | #25398 | NC Variable to MLC 47 | D3x174 | #25430 |
| NC Variable to MLC 16 | D3x143 | #25399 | NC Variable to MLC 48 | D3x175 | #25431 |
| NC Variable to MLC 17 | D3x144 | #25400 | NC Variable to MLC 49 | D3x176 | #25432 |
| NC Variable to MLC 18 | D3x145 | #25401 | NC Variable to MLC 50 | D3x177 | #25433 |
| NC Variable to MLC 19 | D3x146 | #25402 | NC Variable to MLC 51 | D3x178 | #25434 |
| NC Variable to MLC 20 | D3x147 | #25403 | NC Variable to MLC 52 | D3x179 | #25435 |
| NC Variable to MLC 21 | D3x148 | #25404 | NC Variable to MLC 53 | D3x180 | #25436 |
| NC Variable to MLC 22 | D3x149 | #25405 | NC Variable to MLC 54 | D3x181 | #25437 |
| NC Variable to MLC 23 | D3x150 | #25406 | NC Variable to MLC 55 | D3x182 | #25438 |
| NC Variable to MLC 24 | D3x151 | #25407 | NC Variable to MLC 56 | D3x183 | #25439 |
| NC Variable to MLC 25 | D3x152 | #25408 | NC Variable to MLC 57 | D3x184 | #25440 |
| NC Variable to MLC 26 | D3x153 | #25409 | NC Variable to MLC 58 | D3x185 | #25441 |
| NC Variable to MLC 27 | D3x154 | #25410 | NC Variable to MLC 59 | D3x186 | #25442 |
| NC Variable to MLC 28 | D3x155 | #25411 | NC Variable to MLC 60 | D3x187 | #25443 |
| NC Variable to MLC 29 | D3x156 | #25412 | NC Variable to MLC 61 | D3x188 | #25444 |
| NC Variable to MLC 30 | D3x157 | #25413 | NC Variable to MLC 62 | D3x189 | #25445 |
| NC Variable to MLC 31 | D3x158 | #25414 | NC Variable to MLC 63 | D3x190 | #25446 |
| NC Variable to MLC 32 | D3x159 | #25415 | NC Variable to MLC 64 | D3x191 | #25447 |
| NC Variable to MLC 65 | D3x192 | #25448 | NC Variable to MLC 97 | D3x224 | #25480 |
| NC Variable to MLC 66 | D3x193 | #25449 | NC Variable to MLC 98 | D3x225 | #25481 |

5

| Function Name | Special D | # variable | Function Name | Special D | # variable |
|---|---|---|---|---|---|
| NC Variable to MLC 67 | D3x194 | #25450 | NC Variable to MLC 99 | D3x226 | #25482 |
| NC Variable to MLC 68 | D3x195 | #25451 | NC Variable to MLC 100 | D3x227 | #25483 |
| NC Variable to MLC 69 | D3x196 | #25452 | NC Variable to MLC 101 | D3x228 | #25484 |
| NC Variable to MLC 70 | D3x197 | #25453 | NC Variable to MLC 102 | D3x229 | #25485 |
| NC Variable to MLC 71 | D3x198 | #25454 | NC Variable to MLC 103 | D3x230 | #25486 |
| NC Variable to MLC 72 | D3x199 | #25455 | NC Variable to MLC 104 | D3x231 | #25487 |
| NC Variable to MLC 73 | D3x200 | #25456 | NC Variable to MLC 105 | D3x232 | #25488 |
| NC Variable to MLC 74 | D3x201 | #25457 | NC Variable to MLC 106 | D3x233 | #25489 |
| NC Variable to MLC 75 | D3x202 | #25458 | NC Variable to MLC 107 | D3x234 | #25490 |
| NC Variable to MLC 76 | D3x203 | #25459 | NC Variable to MLC 108 | D3x235 | #25491 |
| NC Variable to MLC 77 | D3x204 | #25460 | NC Variable to MLC 109 | D3x236 | #25492 |
| NC Variable to MLC 78 | D3x205 | #25461 | NC Variable to MLC 110 | D3x237 | #25493 |
| NC Variable to MLC 79 | D3x206 | #25462 | NC Variable to MLC 111 | D3x238 | #25494 |
| NC Variable to MLC 80 | D3x207 | #25463 | NC Variable to MLC 112 | D3x239 | #25495 |
| NC Variable to MLC 81 | D3x208 | #25464 | NC Variable to MLC 113 | D3x240 | #25496 |
| NC Variable to MLC 82 | D3x209 | #25465 | NC Variable to MLC 114 | D3x241 | #25497 |
| NC Variable to MLC 83 | D3x210 | #25466 | NC Variable to MLC 115 | D3x242 | #25498 |
| NC Variable to MLC 84 | D3x211 | #25467 | NC Variable to MLC 116 | D3x243 | #25499 |
| NC Variable to MLC 85 | D3x212 | #25468 | NC Variable to MLC 117 | D3x244 | #25500 |
| NC Variable to MLC 86 | D3x213 | #25469 | NC Variable to MLC 118 | D3x245 | #25501 |
| NC Variable to MLC 87 | D3x214 | #25470 | NC Variable to MLC 119 | D3x246 | #25502 |
| NC Variable to MLC 88 | D3x215 | #25471 | NC Variable to MLC 120 | D3x247 | #25503 |
| NC Variable to MLC 89 | D3x216 | #25472 | NC Variable to MLC 121 | D3x248 | #25504 |
| NC Variable to MLC 90 | D3x217 | #25473 | NC Variable to MLC 122 | D3x249 | #25505 |
| NC Variable to MLC 91 | D3x218 | #25474 | NC Variable to MLC 123 | D3x250 | #25506 |
| NC Variable to MLC 92 | D3x219 | #25475 | NC Variable to MLC 124 | D3x251 | #25507 |
| NC Variable to MLC 93 | D3x220 | #25476 | NC Variable to MLC 125 | D3x252 | #25508 |
| NC Variable to MLC 94 | D3x221 | #25477 | NC Variable to MLC 126 | D3x253 | #25509 |
| NC Variable to MLC 95 | D3x222 | #25502 | NC Variable to MLC 127 | D3x254 | #25510 |
| NC Variable to MLC 96 | D3x223 | #25503 | NC Variable to MLC 128 | D3x255 | #25511 |

### 5.3.20 EtherCAT communication information

The NC5 controller allows users to monitor the status of connected slave devices. Therefore, users can designed their alarms and actions according to these special M and D registers.

5

| Function Name | Special M | Function Name | Special M |
|---|---|---|---|
| EtherCAT Communication Status | M30000 | - | - |

| Function Name | Special D | Function Name | Special D |
|---|---|---|---|
| 1st Remote Module Input | D20200<br>D20201<br>D20202<br>D20203 | 5th Remote Module Input | D20216<br>D20217<br>D20218<br>D20219 |
| 2nd Remote Module Input | D20204<br>D20205<br>D20206<br>D20207 | 6th Remote Module Input | D20220<br>D20221<br>D20222<br>D20223 |
| 3rd Remote Module Input | D20208<br>D20209<br>D20210<br>D20211 | 7th Remote Module Input | D20224<br>D20225<br>D20226<br>D20227 |
| 4th Remote Module Input | D20212<br>D20213<br>D20214<br>D20215 | 8th Remote Module Input | D20228<br>D20229<br>D20230<br>D20231 |

### 5.3.21 Robot arm control

The NC5 controller can allocate any NC channel for robot arm mechanism control. Users can define different robot mechanisms, system modes and system coordinates.

| Function Name | Special M | Function Name | Special M |
|---|---|---|---|
| Servo On / Off | M2x012 | C Axis JOG Forward | M2x389 |
| Robot Coordinate System | M2x048<br>M2x049<br>M2x050<br>M2x051 | X Axis JOG Reverse | M2x400 |
| X Axis JOG Forward | M2x384 | Y Axis JOG Reverse | M2x401 |
| Y Axis JOG Forward | M2x385 | Z Axis JOG Reverse | M2x402 |
| Z Axis JOG Forward | M2x386 | A Axis JOG Reverse | M2x403 |
| A Axis JOG Forward | M2x387 | B Axis JOG Reverse | M2x404 |
| B Axis JOG Forward | M2x388 | C Axis JOG Reverse | M2x405 |

| Function Name | Special D | Function Name | Special D |
|---|---|---|---|
| NC Mode Switching | D2x000 | Robot’s Coordinate System Switch in Manual Mode | D2x016 |
| Feed Rate Override for JOG, INC and Dry Run Mode | D2x006 | Robot’s Tool System Switch in Manual Mode | D2x017 |
| MPG Axes Selection | D2x008 | Current Coordinate System | D3x016 |
| MPG Ratio Selection | D2x009 | Current Robot Tool Coordinate System | D3x017 |

# MLC Application Examples 6

This chapter provides commonly used MLC applications, including examples such as analog spindle gear switch, one-button macro call, MLC axes switching and synchronous control.

6

# 6.1 Mode Switch

The NC5 controller provides 8 different types of system modes, which can be switched by setting the MLC special D2x000, and the special M can be used to monitor the mode status.

## ■ MLC special D

| NC Mode Switch | Value | Mode | Mode Status |
|---|---|---|---|
| D2x000 | 0 | AUTO | M3x000 |
| | 1 | EDIT | M3x001 |
| | 2 | MDI | M3x002 |
| | 3 | MPG | M3x003 |
| | 4 | JOG | M3x004 |
| | 5 | RAPID | M3x005 |
| | 6 | INC | M3x006 |
| | 7 | HOME | M3x007 |

**[NC Mode Switch] D2x000**

The NC system uses this D2x000 to switch between different NC modes. The x in D2x000 represents different NC channels.

**[Mode Status] M3x000 ~ M3x007**

When the NC successfully switches to a different mode, the NC mode statuses M3x000 to M3x007 will respond within 4ms.

## ■ MLC Example

In the below MLC program example, a scenario is demonstrated where users use a real hardware button to switch the NC mode of channel 1 to general-purpose use. Two different types of methods are described, including button and knob.

**Button Type:**

The mode switch signal is triggered only when the user presses the button as a DI signal. Therefore, at each DI trigger, the MLC writes the corresponding decimal value to D21000 with the MOV instruction. In addition, M31000 can be used to obtain whether the system is in AUTO mode. If yes, MLC will output a DO signal.

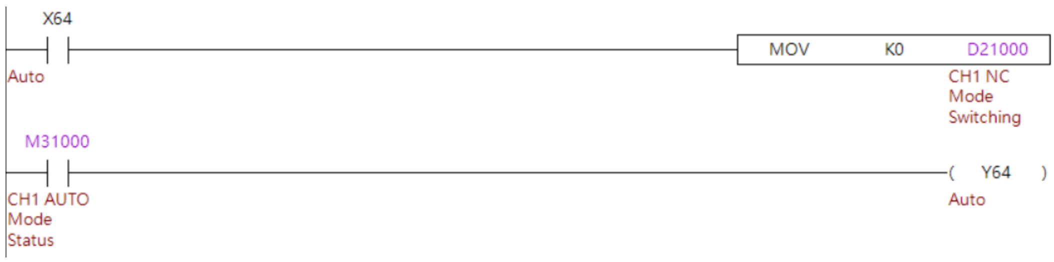



Figure 6.1.1 Mode switch by DI button

**Knob Type:**

The trigger signals input to the MLC continuously. Therefore, when users 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 to the D21000 to switch the system mode and the system uses the **[Mode Status]** special M to output signals.

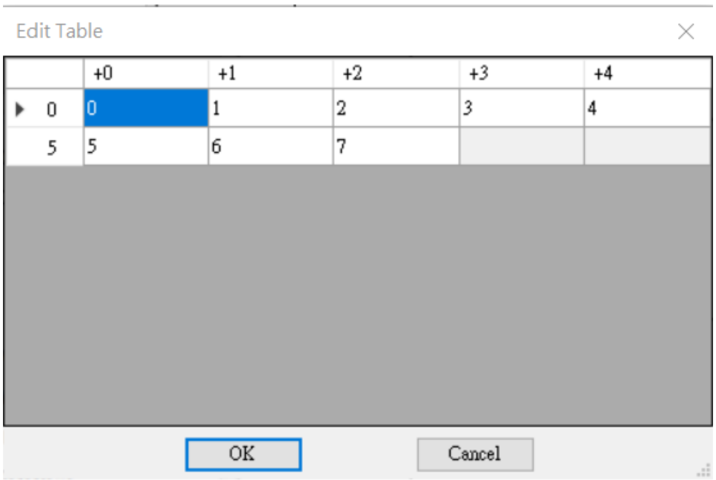



Figure 6.1.2 VRT table


Figure 6.1.3 Mode switching by knob

6

## 6.2 Machining, single block, and feed rate override

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

### ■ MLC special D

| | | | |
|---|---|---|---|
| Cycle Start | M2x000 | Cycle Start Status | M3x016 |
| Feed Hold | M2x001 | Feed Hold Status | M3x017 |
| NC Reset | M2x004 | Program End Finished | M3x021 |
| Enable Dry Run | M2x005 | M02 Cycle Stop | M3x022 |
| MPG Simulation | M2x006 | M30 Cycle Stop and Index Reset | M3x023 |
| Single Block | M2x008 | Single Block Hold | M3x024 |
| Optional Stop | M2x009 | M00 System Hold | M3x037 |
| Single Block Skip (‘/’) | M2x010 | M01 Optional Stop | M3x038 |

| | | | |
|---|---|---|---|
| Feed Rate Percentage | D2x002 | Feed Rate for JOG, INC and Dry Run Mode | D2x006 |
| Rapid Speed Percentage | D2x004 | - | - |

**[Cycle Start, Cycle Start Status] M2x000, M3x016**

When the controller is in AUTO or MDI mode, if the **[Cycle Start]** is triggered, the system will load the main program or MDI program and then start execution. The controller will execute the NC program and the **[Cycle Start Status]** will be set to ON.

- **[Cycle Start]** This trigger must stay as ON for at least one MLC cycle to activate the function.
- **[Cycle Start Status]** After this status changes to ON, the system will automatically reset to OFF in the following situations.
  a. When the NC program is processing but **[Single Block]** is triggered. After the current block is finished this status will be OFF.
  b. When the NC program finishes the M00, M02 or M30 command.
  c. When the NC program is processing but **[NC Reset]** is triggered.
  d. When the NC program finishes the last block command.
  e. When the NC program is processing but **[NC Mode switch]** changes.

**[Feed Hold, Feed Hold Status] M2x001, M3x017**

When the controller is in AUTO or MDI mode, if **[Feed Hold]** is triggered, the system will pause the current process and then set **[Feed Hold Status]** to ON.

- **[Feed Hold]** This trigger must stay as ON for at least one MLC cycle to activate the process.
- **[Feed Hold Status]** After this status changes to ON, it will automatically reset to OFF in the following situations.
  a. After **[Cycle Start]** is triggered again, the system will continue the previous process.
  **b.** After **[NC Reset]** is triggered, the system will stop and reset all statuses.

**[Dry Run, Feed Rate] M2x005, D2x006**

When the controller is in AUTO or MDI mode, if **[Dry Run]** is triggered, the system will load the main program or MDI program and then start the execution with the **[Feed Rate]** speed.

- When **[Dry Run]** is triggered as ON, the system will take **[Feed Rate]** as the process speed.
- When the system is executing in dry run mode and the **[Dry Run]** is reset to OFF, the system will continue running but at the NC programed speed.

**[MPG Simulation] M2x006**

When the controller is in AUTO or MDI mode, if the **[MPG Simulation]** is triggered, the system will execute the entire block command with the MPG rotate speed.

- Besides motion type commands, all other NC programs will execute as normal when the **[MPG Simulation]** is set to ON.
- If the **[MPG Simulation]** is triggered and the system is executing the NC program, the controller system will pause the NC process immediately.
- After the **[MPG Simulation]** is triggered during the system executing the NC program and the system pauses the process, if the **[MPG Simulation]** is reset to OFF, the system will resume the process right away.

**[Single Block, Single Block Hold] M2x008, M3x024**

When the controller is in AUTO or MDI mode, users can switch **[Single Block]** to ON at any time as long as it is before processing or during the NC executing. The system will execute one block command each time **[Cycle Start]** is triggered.

- The **[Single Block]** is enabled before the **[Cycle Start]**. The system will execute one block command once **[Cycle Start]** is triggered, and then hold the process and set **[Single Block Hold]** to ON after each block command is finished.
- When the **[Single Block]** is enabled during NC execution, the system will finish its current block command and then hold the process and set **[Single Block Hold]** to ON.
- When the **[Single Block]** is enabled and the system holds the process, users can trigger the **[Cycle Start]** to continue to the next command block.
- The **[Single Block]** is disabled when the system is in hold status. The system will set **[Single Block Hold]** to OFF automatically after users set **[Cycle Start]** to ON and continue to the next process.

**[Optional Stop, M01 Optional Stop] M2x009, M3x038**

When the controller is in AUTO or MDI mode, if users set **[Optional Stop]** to ON, the system will pause the process or the system will set **[M01 Optional Stop]** to ON once the NC block command is M01.

- When the **[Optional Stop]** is ON and the system holds the process due to execute the M01 command, users can trigger the **[Cycle Start]** again to continue the process.
- When the system is in hold status and releases the **[Optional Stop]** to OFF, the system will continue the process after users trigger the **[Cycle Start]**.
- After the **[M01 Optional Stop]** is ON, it will automatically reset to OFF in the following situations.
  a. After the **[Cycle Start]** has been set to ON again.
  b. After the **[NC Reset]** has been triggered.

**[Single Block Skip] M2x010**

When the controller is in AUTO or MDI mode, if users set **[Single Block Skip]** to ON, the system will skip every command block that starts with the '/' symbol.

**[Program End Finished, M02 Cycle Stop, M30 Cycle Stop and Index Reset, M00 System Hold] M3x021, M3x022, M3x023, M3x037**

When the controller is in AUTO or MDI mode, these special M will be triggered when the last command block is M02, M30, M00 or there are no further blocks.

- When the system executes the last block as M02, M30 or there are no further blocks, regardless of whether it is a sub-program or main program, the **[Program End Finished]** will be ON automatically.
  When the last block is M99, the **[Program End Finished]** will NOT be ON due to the M99 being a continuous command.
- After **[Program End Finished, M02 Cycle Stop, M30 Cycle Stop and Index Reset, M00 System Hold]** are triggered, they will be reset to OFF in the following situations.
  a. After the **[Cycle Start]** has been set to ON again.
  b. After the **[NC Reset]** has been triggered.

**[Feed Rate Percentage] D2x002**

When the controller is in AUTO or MDI mode, the system will take the program speed or dry run speed as a reference to multiply this **[Feed Rate Percentage]** and derive the final process feed. For example, the program feed is 1000 mm/min and **[Feed Rate Percentage]** is 50. The system will execute the NC program as feed 1000 x 50% = F500 mm/min.

- **[Feed Rate Percentage]** this percentage is only available up to 150%.
- The final feed speed will be limit by the parameter N1.033 (Feed Maximum) (mm/min).
- The system's current process feed will activate right away when this percentage changes.

■ **Relevant Parameter**

**Feed rate setting:**

N1.033 (Feed Maximum)

- When the controller is in AUTO or MDI mode, the system will take the program speed or dry run speed as a reference to multiply this **[Feed Rate Percentage]** as the final process feed but this final speed will be limited by the parameter N1.033.

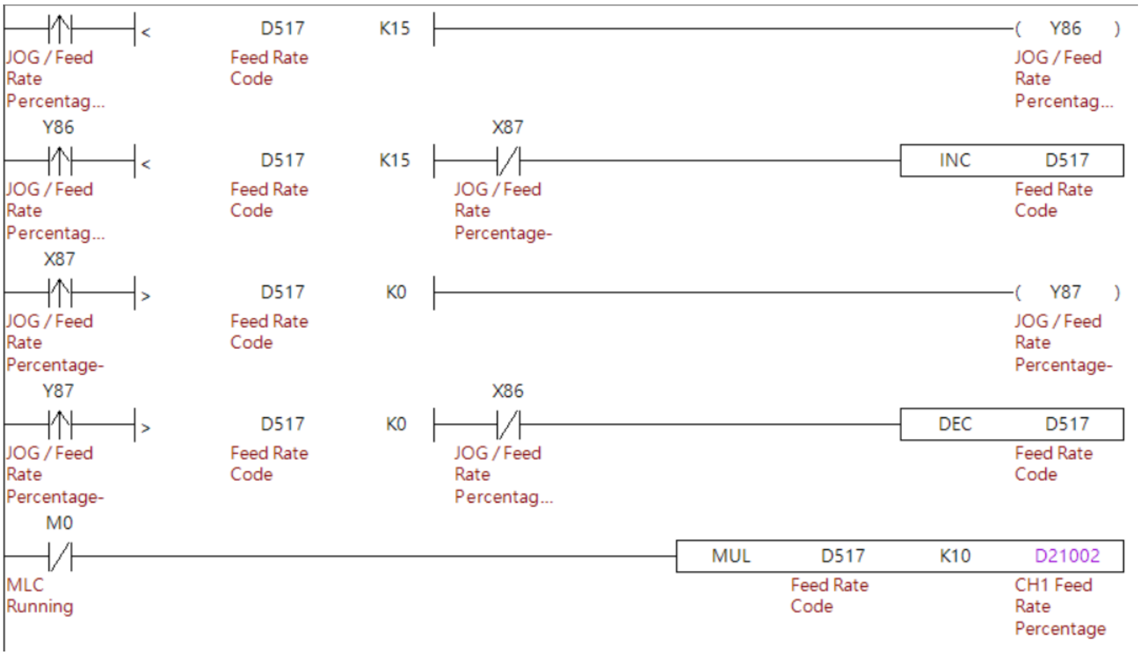
## ■ MLC Example


**Program execution procedure (based on NC channel 1)**

1. Use the button 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 **[Cycle Start Status]** to ON, and then sets the cycle start DO to ON.
3. The system can trigger **[Feed Hold]** only when the program is being executed.
4. When the program pauses, the program feed hold 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 Skip]** 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]** function.
3. Use the rising edge of the key signal to trigger ALT and to switch the special M for enabling/disabling the **[Optional Hold]** 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]**.

6




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 **[Feed Rate Percentage]**.

# 6.3 MPG

On the NC5 controller, the axis and feed rate for MPG operation is switched with special D registers.



## ■ MLC special D register

| MPG Axes Selection | D2x008 | MPG Ratio Selection | D2x009 |
|---|---|---|---|

**[MPG Axes Selection] D2x008**

Users can switch the MPG axis by setting this special D to the indicated axes as below.

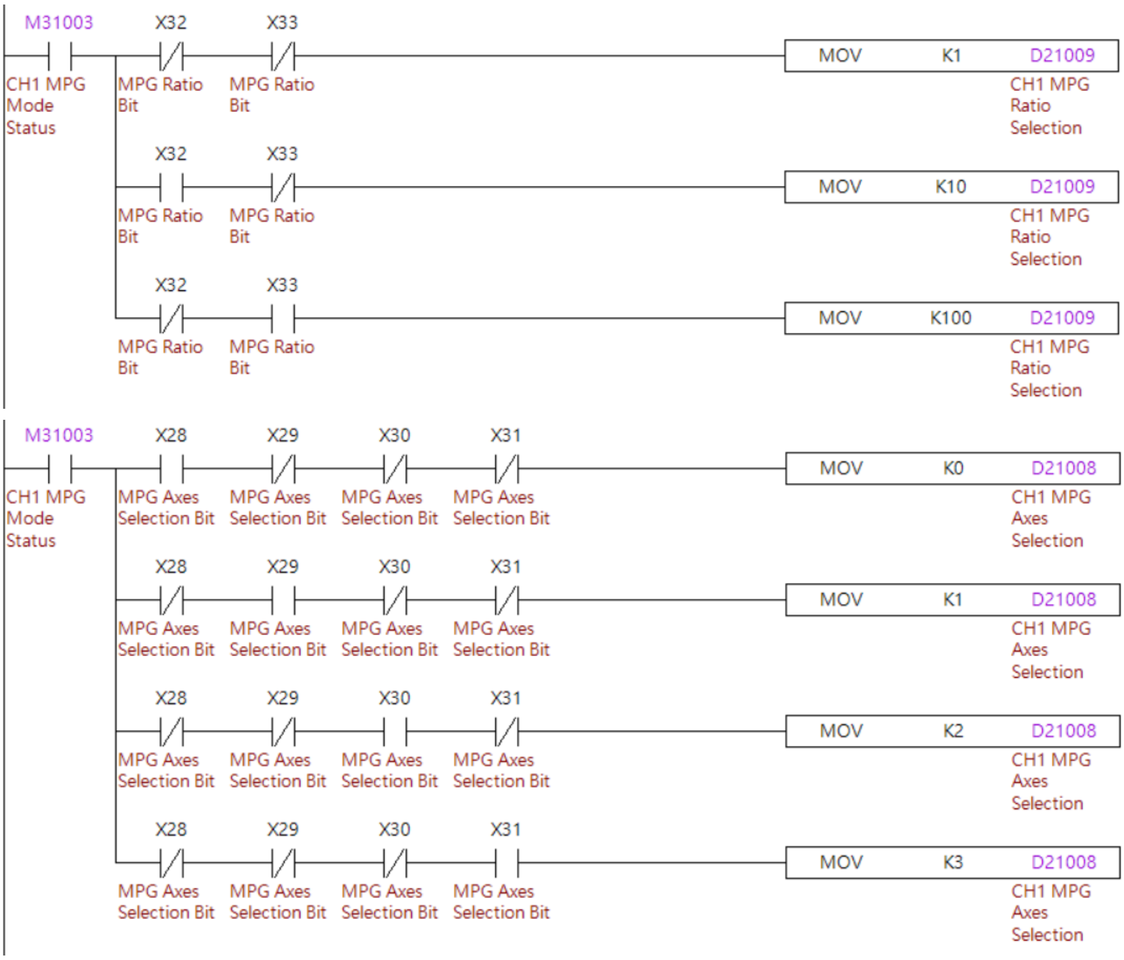
0 = X Axis, 1 = Y Axis, 2 = Z Axis, 3 = A Axis, 4 = B Axis, and so forth to the 16$^{th}$ Axis.

**[MPG Ratio Selection] D2x009**

This **[MPG Ratio Selection]** can be set to 1, 10 or 100. When users rotate one unit from the wheel, the system will obtain 1 pulse from the MPG device. The system will take the parameter **N9.013 (Unit Decimal Point)** as the command unit for each pulse. In addition, the system will multiply the **[MPG Ratio Selection]** as the ratio for the final command sent to a specific axis. For example: **[MPG Ratio Selection]** is 10 and the N9.013 (Unit Decimal Point) set to 0.001mm. The minimum movement of the MPG control will be 0.001 × 10 = 0.01 mm.

6

## ■ MLC Example




**Program execution procedure**

1. In MPG mode, use the MPG signal to choose the **[MPG Ratio Selection]** to use.
2. Use the MPG signal to select the axis to move.
3. **[MPG Axes Selection]** Operate the MPG to move the axis.

## 6.4 JOG move

On the NC5 controller, the speed and movement for JOG operation is determined by the MLC. JOG applications are as follows.

### ■ MLC special D

| Axis | JOG Forward | JOG Reverse | Lock Axis Forward Movement | Lock Axis Reverse Movement | Axis Moving | Moving in Positive Direction | Moving in Reverse Direction | Feed Rate Override |
|---|---|---|---|---|---|---|---|---|
| X Axis | M2x384 | M2x400 | M2x336 | M2x352 | M3x464 | M3x480 | M3x496 | D2x006 |
| Y Axis | M2x385 | M2x401 | M2x337 | M2x353 | M3x465 | M3x481 | M3x497 | |
| Z Axis | M2x386 | M2x402 | M2x338 | M2x354 | M3x466 | M3x482 | M3x498 | |
| A Axis | M2x387 | M2x403 | M2x339 | M2x355 | M3x467 | M3x483 | M3x499 | |
| B Axis | M2x388 | M2x404 | M2x340 | M2x356 | M3x468 | M3x484 | M3x500 | |
| C Axis | M2x389 | M2x405 | M2x341 | M2x357 | M3x469 | M3x485 | M3x501 | |
| U Axis | M2x390 | M2x406 | M2x342 | M2x358 | M3x470 | M3x486 | M3x502 | |
| V Axis | M2x391 | M2x407 | M2x343 | M2x359 | M3x471 | M3x487 | M3x503 | |
| W Axis | M2x392 | M2x408 | M2x344 | M2x360 | M3x472 | M3x488 | M3x504 | |
| $10^{th}$ Axis | M2x393 | M2x409 | M2x345 | M2x361 | M3x473 | M3x489 | M3x505 | |
| $11^{th}$ Axis | M2x394 | M2x410 | M2x346 | M2x362 | M3x474 | M3x490 | M3x506 | |
| $12^{th}$ Axis | M2x395 | M2x411 | M2x347 | M2x363 | M3x475 | M3x491 | M3x507 | |
| $13^{th}$ Axis | M2x396 | M2x412 | M2x348 | M2x364 | M3x476 | M3x492 | M3x508 | |
| $14^{th}$ Axis | M2x397 | M2x413 | M2x349 | M2x365 | M3x477 | M3x493 | M3x509 | |
| $15^{th}$ Axis | M2x398 | M2x414 | M2x350 | M2x366 | M3x478 | M3x494 | M3x510 | |
| $16^{th}$ Axis | M2x399 | M2x415 | M2x351 | M2x367 | M3x479 | M3x495 | M3x511 | |

**[JOG Forward] M2x384 ~ M2x399**

When the controller is in JOG mode, users can move any of the axes forward or stop by setting the **[JOG Forward]** to ON or OFF.

**[JOG Reverse] M2x400 ~ M2x415**

When the controller is in JOG mode, users can move any of the axes reverse or stop by setting the **[JOG Reverse]** to ON or OFF.

**[Lock Axis Forward Movement] M2x336 ~ M2x351**

When the **[Lock Axis Forward Movement]** is ON, the axis will not be able to move in the forward direction. Reset it to OFF to disable the limitation.

**[Lock Axis Reverse Movement] M2x352 ~ M2x367**

When the **[Lock Axis Reverse Movement]** is ON, the axis will not be able to move in the reverse direction. Reset it to OFF to disable the limitation.

**[Axis Moving] M3x464 ~ M3x479**

When any of the axes is in motion, no matter which direction, the system will set this **[Axis Moving]** to ON. Otherwise, this **[Axis Moving]** will be OFF when the corresponding axis is still.

**[Moving in Positive Direction] M3x480 ~ M3x495**

When the axis is moving forward, the system will set this **[Moving in Positive Direction]** to ON. On the other hand, when the axis is still or moving backward, this **[Moving in Positive Direction]** will be OFF.

**[Moving in Reverse Direction] M3x496 ~ M3x511**

When the axis is moving a reverse direction, the system will set this **[Moving in Reverse Direction]** to ON. On the other hand, when the axis is still or moving forward, this **[Moving in Reverse Direction]** will be OFF.

**[Feed Rate Override] D2x006**

When the controller is in JOG mode, the system will take **N2.030 (JOG Maximum Speed)** and then multiply the **[Feed Rate Override]** to derive the JOG speed. The linear axes will move in the mm/min and the rotary axes will reference the parameter N2.001 (Rotary Axes' Unit) and then move in the unit RPM or degree/min.

- **[Feed Rate Override]** value range 0 ~ 100.

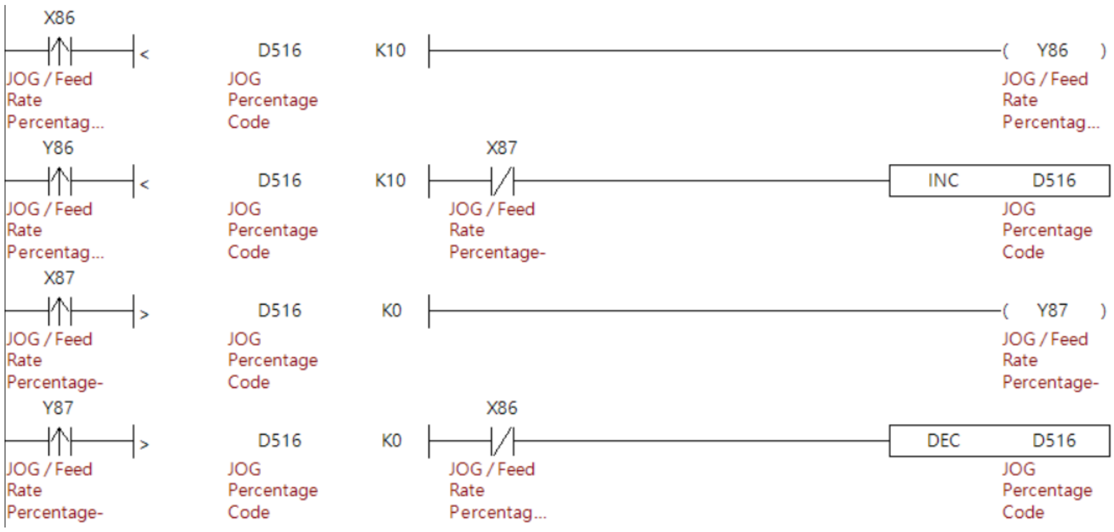
## ■ Relevant Parameter

**Speed parameter setting:**

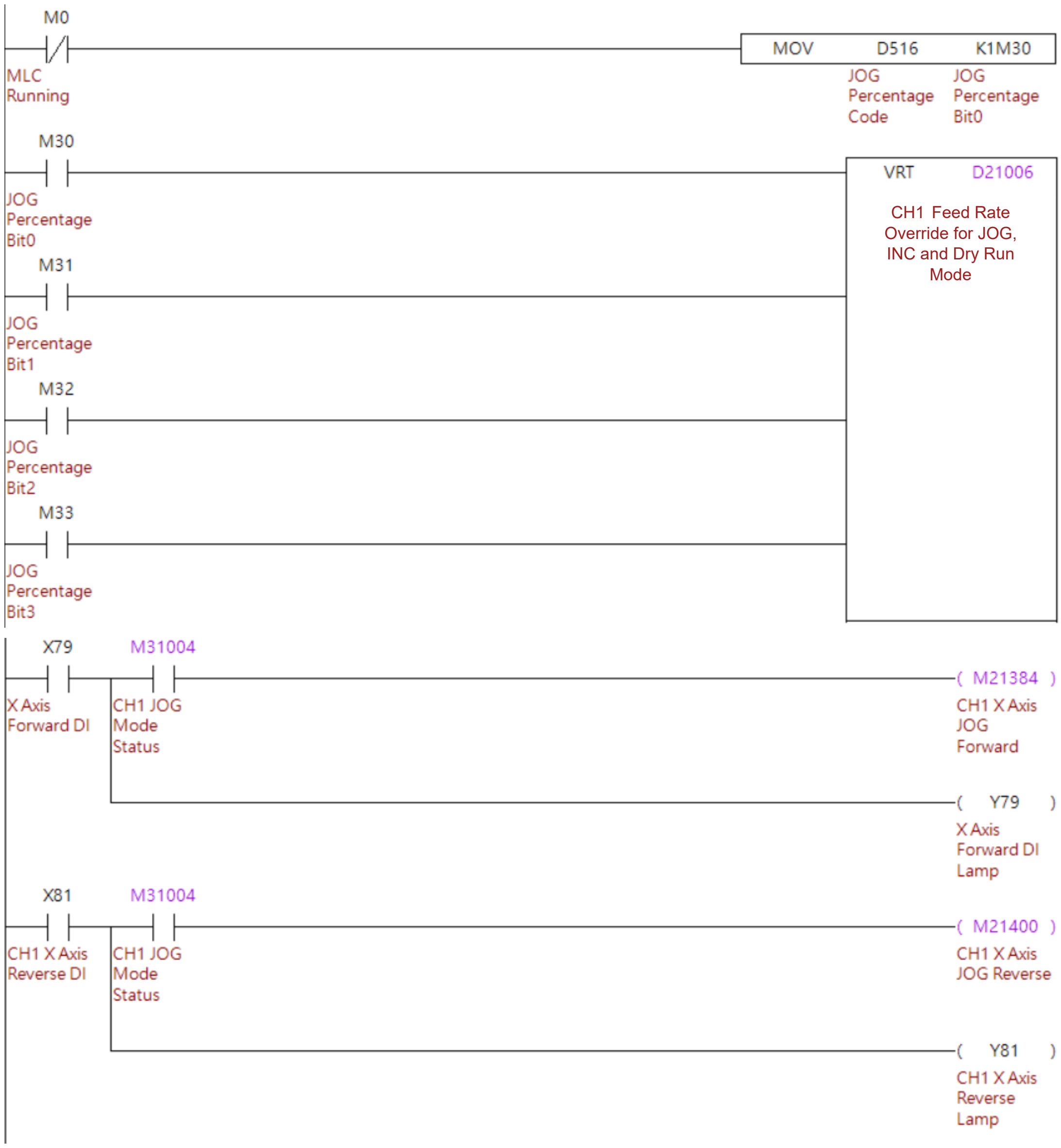
This JOG speed will be effected by N2.030 (JOG Maximum Speed), N2.031 (JOG Acc and Dec Time) and N2.032 (JOG S Curve Time).

## ■ MLC Example

The **[Feed Rate Override]** can be edited at any time even if the axes are moving.



6




**Program execution procedure**

1. Use the key signal to increase or decrease the value of **[Feed Rate Override]**, with the maximum as 10 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 D21006.
4. Perform the X-axis jog operation using the X-axis positive / negative JOG keys.

6

## 6.5 INC move

On the NC5 controller, the speed and movement for INC operations are determined by the MLC. INC applications are as follows.

### ■ MLC special D

| Axis | JOG Forward | JOG Reverse | Lock Axis Forward Movement | Lock Axis Reverse Movement | Axis Moving | Moving in Positive Direction | Moving in Reverse Direction | Feed Rate Override | Axes Movement in INC Mode |
|---|---|---|---|---|---|---|---|---|---|
| X Axis | M2x384 | M2x400 | M2x336 | M2x352 | M3x464 | M3x480 | M3x496 | D2x006 | D2x014 |
| Y Axis | M2x385 | M2x401 | M2x337 | M2x353 | M3x465 | M3x481 | M3x497 | | |
| Z Axis | M2x386 | M2x402 | M2x338 | M2x354 | M3x466 | M3x482 | M3x498 | | |
| A Axis | M2x387 | M2x403 | M2x339 | M2x355 | M3x467 | M3x483 | M3x499 | | |
| B Axis | M2x388 | M2x404 | M2x340 | M2x356 | M3x468 | M3x484 | M3x500 | | |
| C Axis | M2x389 | M2x405 | M2x341 | M2x357 | M3x469 | M3x485 | M3x501 | | |
| U Axis | M2x390 | M2x406 | M2x342 | M2x358 | M3x470 | M3x486 | M3x502 | | |
| V Axis | M2x391 | M2x407 | M2x343 | M2x359 | M3x471 | M3x487 | M3x503 | | |
| W Axis | M2x392 | M2x408 | M2x344 | M2x360 | M3x472 | M3x488 | M3x504 | | |
| 10th Axis | M2x393 | M2x409 | M2x345 | M2x361 | M3x473 | M3x489 | M3x505 | | |
| 11th Axis | M2x394 | M2x410 | M2x346 | M2x362 | M3x474 | M3x490 | M3x506 | | |
| 12th Axis | M2x395 | M2x411 | M2x347 | M2x363 | M3x475 | M3x491 | M3x507 | | |
| 13th Axis | M2x396 | M2x412 | M2x348 | M2x364 | M3x476 | M3x492 | M3x508 | | |
| 14th Axis | M2x397 | M2x413 | M2x349 | M2x365 | M3x477 | M3x493 | M3x509 | | |
| 15th Axis | M2x398 | M2x414 | M2x350 | M2x366 | M3x478 | M3x494 | M3x510 | | |
| 16th Axis | M2x399 | M2x415 | M2x351 | M2x367 | M3x479 | M3x495 | M3x511 | | |

**[JOG Forward] M2x384 ~ M2x399**

When the controller is in INC mode, users can move any of the axes forward with a movement by setting the **[JOG Forward]** to ON.

**[JOG Reverse] M2x400 ~ M2x415**

When the controller is in INC mode, users can move any of the axes reverse with a movement by setting the **[JOG Reverse]** to ON.

**[Lock Axis Forward Movement] M2x336 ~ M2x351**

When the **[Lock Axis Forward Movement]** is ON, the axis will not be able to move in the forward direction. Reset it to OFF to disable the limitation.

**[Lock Axis Reverse Movement] M2x352 ~ M2x367**

When the **[Lock Axis Reverse Movement]** is ON, the axis will not be able to move in the reverse direction. Reset it to OFF to disable the limitation.

**[Axis Moving] M3x464 ~ M3x479**

When any of the axes is in motion, no matter which direction, the system will set this **[Axis Moving]** to ON. Otherwise, this **[Axis Moving]** will be OFF when the corresponding axis is still.

**[Moving in Positive Direction] M3x480 ~ M3x495**

When the axis is moving forward, the system will set this **[Moving in Positive Direction]** to ON. On the other hand, when the axis is still or moving backward, this **[Moving in Positive Direction]** will be OFF.

**[Moving in Reverse Direction] M3x496 ~ M3x511**

When the axis is moving a reverse direction, the system will set this **[Moving in Reverse Direction]** to ON. On the other hand, when the axis is still or moving forward, this **[Moving in Reverse Direction]** will be OFF.

**[Feed Rate Override] D2x006**

When the controller is in INC mode, the system will take **N2.030 (JOG Maximum Speed)** and then multiply the **[Feed Rate Override]** to derive the INC movement speed. The linear axes will move in the mm/min and the rotary axes will reference the parameter N2.001 (Rotary Axes' Unit) and then move in the unit RPM or degree/min.

- **[Feed Rate Override]** value range 0~100.

**[Axes Movement in INC Mode] D2x014**

When the controller is in INC mode, the system will take **[Axes Movement in INC Mode]** and then multiply the parameter **N9.013 (Unit Decimal Point)** to derive the movement distance when the **[JOG Forward]** or **[JOG Reverse]** is triggered.

- The INC movement will only take place once after the move command **[JOG Forward]** or **[JOG Reverse]** is triggered until motion finished or stopped. This INC movement does not support the position change function.
- **[Axes Movement in INC Mode]** is a 32-bit data type special D. Therefore, D2x015 is occupied as high-word data.

## ■ Relevant Parameter

**Speed parameter setting:**

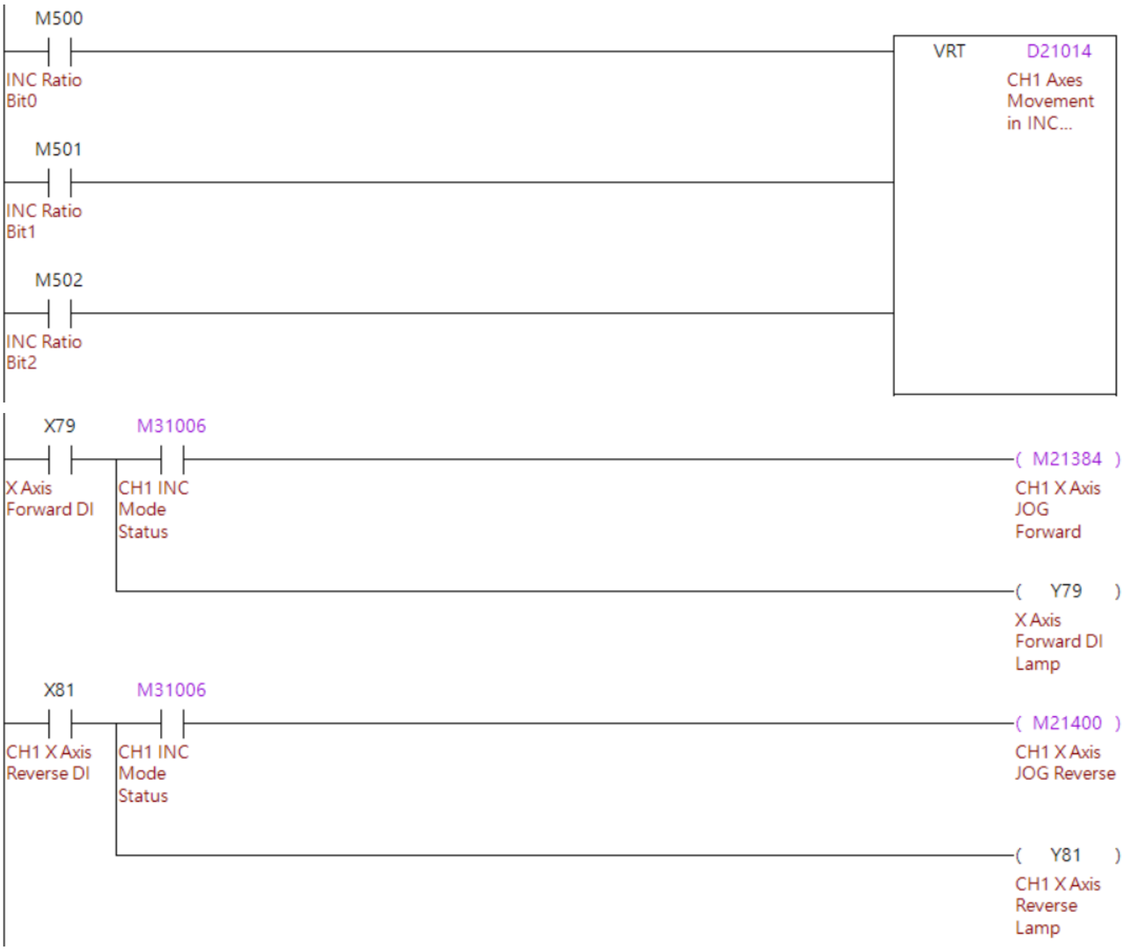
This INC speed will be effected by N2.030 (JOG Maximum Speed), N2.031 (JOG Acc and Dec Time) and N2.032 (JOG S Curve Time).

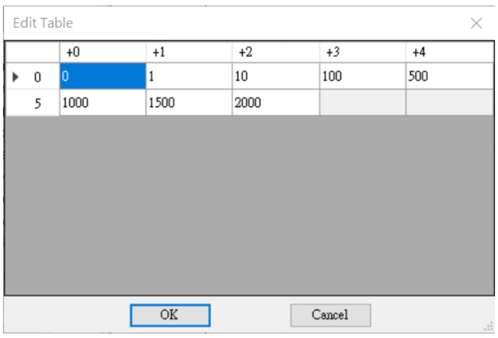
The movement takes parameter **N9.013 (Unit Decimal Point)** and the system will take **[Axes Movement in INC Mode]** and then multiply the parameter N9.013 (Unit Decimal Point) to derive the movement distance.

6

## ■ MLC Example

**[Axes Movement in INC Mode]** needs to be set before the movement command. If this **[Axes Movement in INC Mode]** is changed during motion, it will be available at the next command trigger.





Edit Table

| | +0 | +1 | +2 | +3 | +4 |
|---|---|---|---|---|---|
| 0 | 0 | 1 | 10 | 100 | 500 |
| 5 | 1000 | 1500 | 2000 | | |

OK Cancel

**Program execution procedure**

1. Use VRT table to convert M500 – M502 into decimal values and move the data into D21014 as **[Axes Movement in INC Mode]**.
2. Perform the X-axis jog operation using the X-axis positive / negative JOG keys.

## 6.6 Rapid move

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. Rapid traverse applications are as follows.

### ■ MLC special D register

| | |
|---|---|
| Rapid Speed Percentage | D2x004 |

**[Rapid Speed Percentage] D2x004**

When the controller is in RAPID mode or executing the G00 command in the NC program, the system will refer to parameter **N1.030 (G00 Velocity)** and N2.020 (G00 Max Velocity) as full speed in 100%. In addition, the system will multiply this **[Rapid Speed Percentage]** as a ratio to execute the process.

- **[Rapid Speed Percentage]** is available in the range 1~100.
- During rapid command executing, the actual speed of each axis, which is multiplied by the **[Rapid Speed Percentage]**, will be limited by parameter N2.020 (G00 max velocity).
- When the value of **[Rapid Speed Percentage]** changes, the rapid actual speed will change right away.

### ■ Relevant Parameter

**Rapid speed setting in AUTO mode:**

G00 rapid command speed will refer to following parameters:

N1.030 (G00 Velocity), N1.031 (G00 Acc and Dec Time), N1.032 (G00 S Curve Time), N2.020 (Axis G00 Max Velocity), N2.021 (Axis Acc and Dec Time of G00 Command) and N2.022 (Axis S Curve Time of G00 Command).

N1.030, N1.031, N1.032 are the rapid speed configuration of all interpolation axes.

N2.020, N2.021, N2.022 are the rapid speed configuration of specific axis.

The NC5 controller will, based on these parameters, interpolate the proper path control.

**Rapid speed setting in MDI mode:**

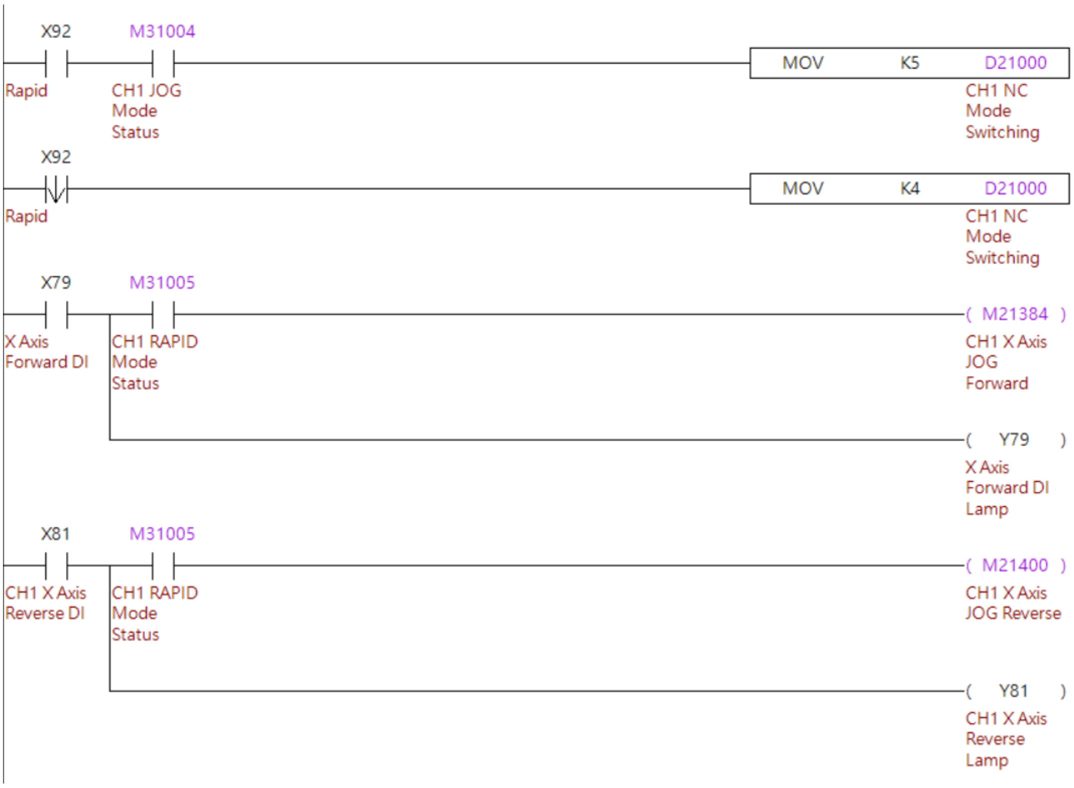
When the controller is in MDI mode, the rapid speed will refer to the following parameters:

N2.030 (Axis JOG Max Velocity), N2.031 (Axis JOG Acc and Dec Time) and N2.032 (Axis JOG S Curve Time).

6

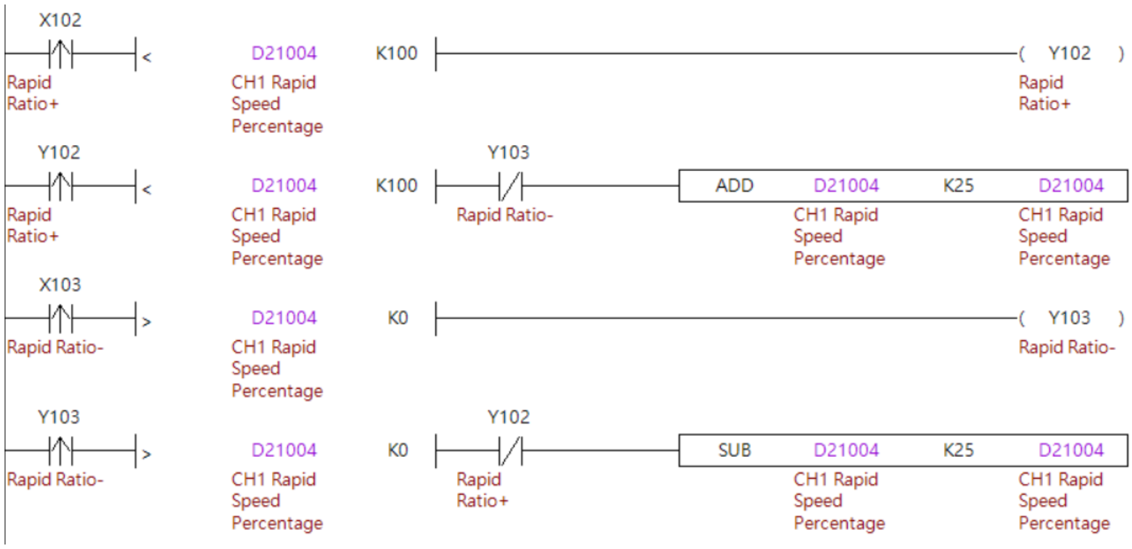
## ■ MLC Example

The following demonstrates the switching of RAPID mode and speed override.




**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 often 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 users press the X axis direction key, the system moves the axis based on the setting of N2.022 (Axis S Curve Time of G00 Command).




**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 the **[Rapid Speed Percentage]**.

6

## 6.7 Homing

On the NC5 controller, this homing procedure needs to be triggered under HOME mode. Users will also need to use the special M to activate the specific axis execute the homing procedure.

### ■ MLC special M relays

| Axis | Axis Homing | Homing Finished | Axis Homed Status |
|---|---|---|---|
| X Axis | M2x320 | M3x336 | M3x320 |
| Y Axis | M2x321 | M3x337 | M3x321 |
| Z Axis | M2x322 | M3x338 | M3x322 |
| A Axis | M2x323 | M3x339 | M3x323 |
| B Axis | M2x324 | M3x340 | M3x324 |
| C Axis | M2x325 | M3x341 | M3x325 |
| U Axis | M2x326 | M3x342 | M3x326 |
| V Axis | M2x327 | M3x343 | M3x327 |
| W Axis | M2x328 | M3x344 | M3x328 |
| $10^{th}$ Axis | M2x329 | M3x345 | M3x329 |
| $11^{th}$ Axis | M2x330 | M3x346 | M3x330 |
| $12^{th}$ Axis | M2x331 | M3x347 | M3x331 |
| $13^{th}$ Axis | M2x332 | M3x348 | M3x332 |
| $14^{th}$ Axis | M2x333 | M3x349 | M3x333 |
| $15^{th}$ Axis | M2x334 | M3x350 | M3x334 |
| $16^{th}$ Axis | M2x335 | M3x351 | M3x335 |

**[Axis Homing] M2x320 ~ M2x335**

When the controller is in HOME mode, users can set **[Axis Homing]** to ON to activate the homing procedure.

**[Homing Finished] M3x336 ~ M3x351**

When the controller finishes the homing procedure, the system will set **[Homing Finished]** to ON automatically.

- After **[Homing Finished]** is set to ON, the system will reset it to OFF automatically in the following situations.
  a. When axes are moving in JOG or MPG mode.
  b. When the system is executing the NC program in AUTO or MDI mode.
  c. When a non-absolute motor is used and the parameter N2.050 (Homing Mode) is not set to mode 5, and the controller is powered on again.
  d. Once an axis loses its origin, this **[Homing Finished]** will reset automatically.

**[Axis Homed Status] M3x320 ~ M3x335**

After axes have finished the homing procedure and the origin position has been defined, the system will set **[Axis Homed Status]** permanently. However, the system will still reset it to OFF in the following situations.

- After **[Axis Homed Status]** is set to ON, the system will reset it to OFF automatically in the following situations.
  a. When a non-absolute motor is used and the parameter N2.050 (Homing Mode) is not set to mode 5, and the controller is powered on again.
  b. Once an axis loses its origin, this **[Homing Finished]** will reset automatically.

## ■ Relevant Parameter

**Homing mode setting:**

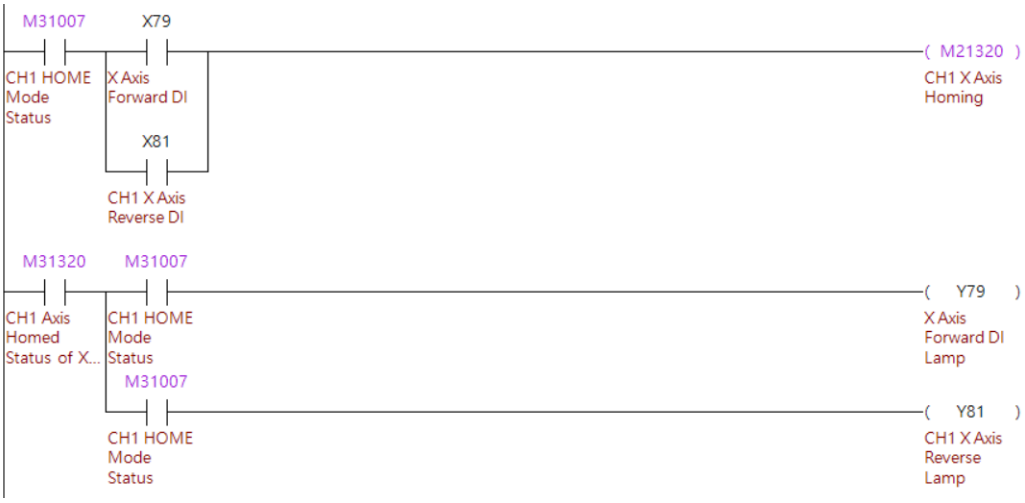
Homing procedure for home switch and Z phase defined as **N2.050 (Homing Mode)** and **N2.051 (Origin Type)**.

**Homing speed:**

Homing speed is defined as **N2.053 ($1^{st}$ Searching Speed)** and **N2.054 ($2^{nd}$ Searching Speed)**.

## ■ MLC Example

The system needs to switch to HOME mode and then use special M and D to start the procedure.




**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.

# 6.8 M / S / 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 an M, S, or T code, it triggers the M, S or T code execution. This special M relay must be confirmed and reset to OFF by the MLC. The functions of M, S and T codes are as follows.

## ■ MLC special D

| | | | |
|---|---|---|---|
| M, S, and T Codes Finished | M2x016 | 1st M Code Data | D3x048 |
| M Code Execution | M3x064 | 1st S Code Data | D3x050 |
| S Code Execution | M3x065 | 1st T Code Data | D3x052 |
| T Code Execution | M3x066 | - | - |

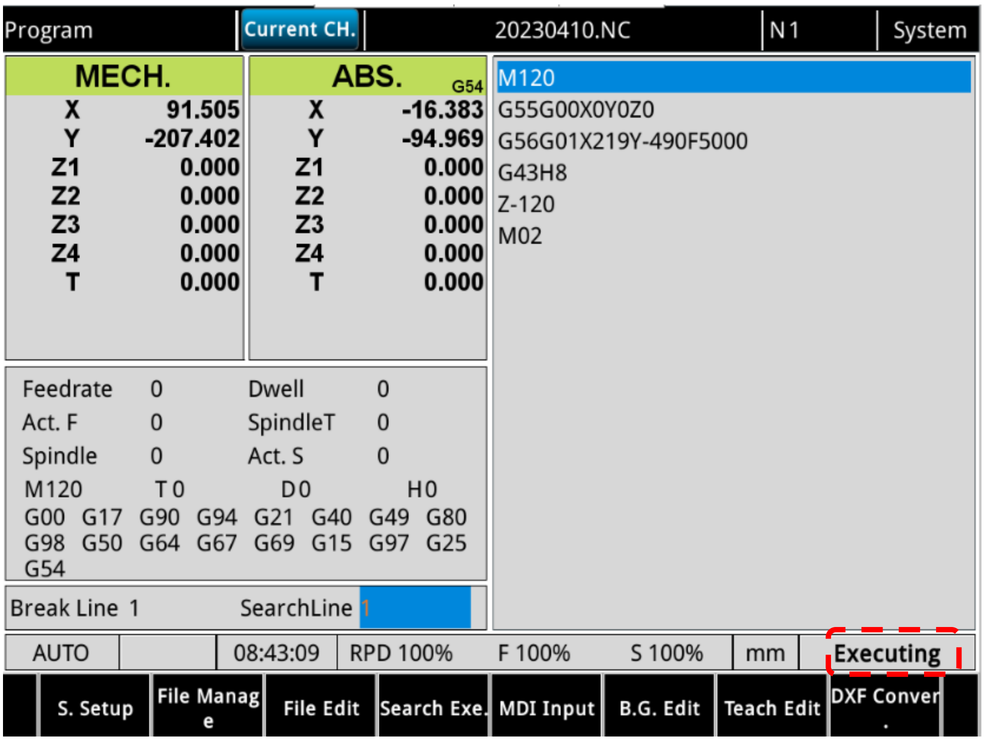



Figure 6.8.1 M, S, and T codes in execution

**[M, S and T Codes Finished] M2x016**

When the system is executing the program and the M, S, or T code is executed, the system pauses executing the program, which keeps the state as "Executing" (as shown in Figure 6.8.1) and sets the **[M, S and T Codes Execution]** flags to ON. After the MLC program finishes these M, S and T codes' actions, the MLC will need to set **[M, S and T Codes Finished]** to ON. Once the system receives the trigger of **[M, S and T Codes Finished]**, it will reset **[M, S and T Codes Execution]** and then continue with the rest of the NC program.

**[M Codes Execution] M3x064**

When the system is executing the program and the M code is executed, the system pauses executing the program, which keeps the state as "Executing" (as shown in Figure 6.8.1) and sets the **[M Codes Execution]** flags to ON. After the MLC program finishes these M codes' actions, the MLC will need to set **[M, S and T Codes Finished]** to ON. Once the system receives the trigger of **[M, S and T Codes Finished]**, it will reset **[M Codes Execution]** and then continue with the rest of the NC program.

**[S Codes Execution] M3x065**

When the system is executing the program and the S code is executed, the system pauses executing the program, which keeps the state as “Executing” (as shown in Figure 6.8.1) and sets the **[S Codes Execution]** flags to ON. After the MLC program finishes these M codes’ actions, the MLC will need to set **[M, S and T Codes Finished]** to ON. Once the system receives the trigger of **[M, S and T Codes Finished]**, it will reset **[S Codes Execution]** and then continue with the rest of the NC program.

**[T Codes Execution] M3x066**

When the system is executing the program and the T code is executed, the system pauses executing the program, which keeps the state as “Executing” (as shown in Figure 6.8.1) and sets the **[T Codes Execution]** flags to ON. After the MLC program finishes these M codes’ actions, the MLC will need to set **[M, S and T Codes Finished]** to ON. Once the system receives the trigger of **[M, S and T Codes Finished]**, it will reset **[T Codes Execution]** and then continue with the rest of the NC program.

**[1st M Code Data] D3x048**

When the system is executing the program and the M code is executed, the system pauses executing the program, which keeps the state as “Executing” (as shown in Figure 6.8.1) and sets the **[1st M Code Data]** to the 1st M code in the command block.

- For example: When the system is executing M13, the value of **[1st M Code Data]** will be 13.
- If the 1st M code in the block is defined as macro call, this **[1st M Code Data]** will stay as the previous value.

**[1st S Code Data] D3x050**

When the system is executing the program and the S code is executed, the system pauses executing the program, which keeps the state as “Executing” (as shown in Figure 6.8.1) and sets the **[1st S Code Data]** to the 1st S code in the command block.

- For example: When the system is executing S4000, the value of **[1st S Code Data]** will be 4000.

**[1st T Code Data] D3x052**

When the system is executing the program and the T code is executed, the system pauses executing the program, which keeps the state as “Executing” (as shown in Figure 6.8.1) and sets the **[1st T Code Data]** to the 1st T code in the command block.

- For example: When the system is executing T5, the value of **[1st T Code Data]** will be 5.
- The first T code command number must also be defined in the range of the tool magazine, and it will respond in the **[1st T Code Data]**. When the system executes the T code command and the **“Continue Index”** is not set to 1, the system will return an error and this T code will not respond in the **[1st T Code Data]**. If users execute the corresponding T code, which is disabled, the system will return an alarm to acknowledge.

## Relevant Parameter

**M code call macro setting:**

**N1.123 (M Code Call Macro Stater M Index):**

Setting for the first code number for M command.

**N1.124 (M Code Call Macro Stater Macro File Number):**

Setting the first Macro file name as number corresponding to the parameter N1.123.

**N1.125 (M Code Call Macro Amount):**

Setting the M code call Macro amount start from N1.123.

If the M code in the block is defined as a macro call function, this M code data will not update to the **[M Code Data]** and the **[M Codes Execution]** will not be triggered. It will be based on parameter N1.123 to N1.125 to execute the responded macro program.

- If the N1.125 is set to 0, it means the M code call macro function is disabled.
- For example, when N1.23 is set to 10, N1.124 is set to 9100 and N1.125 is set to 15, this means the system will execute Macro O9100 when M10 is executed, execute Macro O9101 when M11 is executed, etc. until M24, the system will execute Macro O9114 and for M25 it will execute normal M code procedures.
- If the macro call function's M code is programed inside the corresponded macro, this M code will progress as a normal M code procedure.

**T code call macro O9000 setting:**

**N1.010 (T Code Call Macro O9000)**

- N1.010 Bit 22 set to 0, function disabled.
- When **N1.010 Bit 22** is set to 1, the system will execute the O9000 macro directly instead of triggering the **[T Codes Execution]** or updating the T code to **[T Code Data]**, and instead will execute O9000.
  If the current program is O9000, the system will not execute O9000 again. It will instead execute using the regular method, which will trigger the **[T Codes Execution]** and update the T code to **[T Code Data]**.

Halt M code setting:

**N1.118 (Beginning M Code of Halt Function):**

The first M code of the halt function.

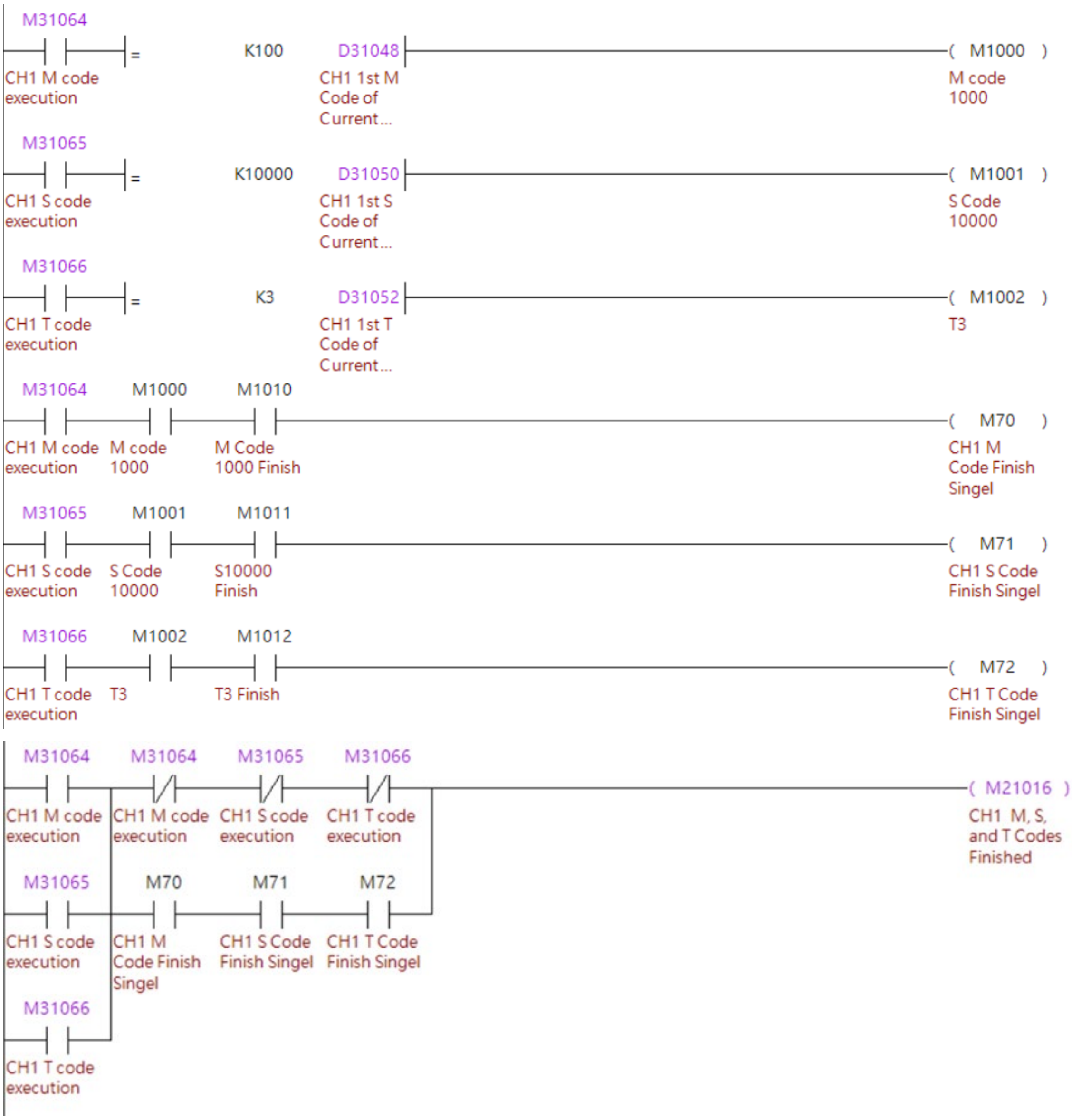
**N1.119 (Amount of Halt M Code):**

The number of halt M code function.

- When the controller is executing the program, the system automatically plans the path by looking ahead of the subsequent program, which has not been executed. When the M code is set to halt the look-ahead of the 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 N1.118 is set to 100 and N1.119 is set to 10, this means M100 to M109 has enabled the halt M code function.
- When the controller is executing the program and reads the Halt M code:
  a. The system continues to execute the program that follows and sets the state to "Executing".
  b. The system sets **[M Code Execution]** to ON and writes the executed M code value to **[M Code Data]**.
  c. Users can plan the MLC actions for the M code normally. The **[M, S and T Codes Finished]** has to be set to ON to execute 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 and writes 100 to **[1st M Code Data]**.
2. When the conditions are met, M1000 in the MLC is set to ON, so users can use M1000 to trigger the required MLC actions.
3. Once MLC has finished the planned MLC actions, set M1010 of the MLC to ON.
4. When you set M1010 to ON, M70 is set to ON because the conditions are met.
5. When M70 is ON, the **[M, S, and T Codes Finished]** flag is set to ON because the conditions are met.
6. When the system confirms that the **[M, S, and T Codes Finished]** flag is ON, it considers the M code execution to be complete and sets **[M Code Execution]** to OFF.
7. M code actions complete.

6

S-code execution procedure

1. When the system is executing the program and executes S10000, it sets **[S Code Execution]** to ON and writes 10000 to **[1st S Code Data]**.
2. When the conditions are met, M1001 in the MLC is set to ON, so users can use M1001 to trigger the required MLC actions.
3. Once MLC has finished the planned MLC actions, set M1011 of the MLC to On.
4. When you set M1011 to ON, M71 is set to ON because the conditions are met.
5. When M71 is ON, the **[M, S, and T Codes Finished]** flag is set to ON because the conditions are met.
6. When the system confirms that the **[M, S, and T Codes Finished]** flag is ON, it considers the S code execution to be complete and sets **[S Code Execution]** to OFF.
7. 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 and writes 3 to **[1st T code data]**.
2. When the conditions are met, M1002 in the MLC is set to ON, so users can use M1002 to trigger the required MLC actions.
3. Once MLC has finished the planned MLC actions, set M1012 of the MLC to On.
4. When you set M1012 to ON, M72 is set to ON because the conditions are met.
5. When M72 is ON, the **[M, S, and T Codes Finished]** flag is set to ON because the conditions are met.
6. When the system confirms that the **[M, S, and T codes execution complete]** flag is ON, it considers the T code execution to be complete and sets **[T code execution]** to OFF.
7. T code actions complete.

**Important:**

the examples in this section are only based on T code. For more details about the tool magazine and T code functions, refer to Section 6.13.

## 6.9 1st software limit / 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. Users 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 axis motion commands. To resume the movement, users must cancel the hardware limit.

### ■ MLC special M relays

| Axis | Disable 1st Software Limit | Disable Hardware Limit |
|---|---|---|
| X Axis | M2x368 | M2x007 |
| Y Axis | M2x369 | |
| Z Axis | M2x370 | |
| A Axis | M2x371 | |
| B Axis | M2x372 | |
| C Axis | M2x373 | |
| U Axis | M2x374 | |
| V Axis | M2x375 | |
| W Axis | M2x376 | |
| 10th Axis | M2x377 | |
| 11th Axis | M2x378 | |
| 12th Axis | M2x379 | |
| 13th Axis | M2x380 | |
| 14th Axis | M2x381 | |
| 15th Axis | M2x382 | |
| 16th Axis | M2x383 | |

**[Disable 1st Software Limit] M2x368 ~ M2x383**

When **[Disable 1st Software Limit]** set to ON, the 1st software limit of this axis will be canceled.

**[Disable Hardware Limit] M2x007**

When **[Disable Hardware Limit]** set to ON, the hardware limits of all axes are canceled.

### ■ Relevant Parameter

**Software limit setting:**

N2.006 (1st positive software limit), N2.007 (1st negative software limit),
N2.008 (2nd positive software limit), N2.009 (2nd negative software limit): When axes reach different limit positions, the system will return the corresponding alarm. The software limit will be active after the axes' origin position is established.

- Therefore, if axes are using incremental motors and system has just started up, it will need to finish its home procedure.

**Hardware limit setting:**

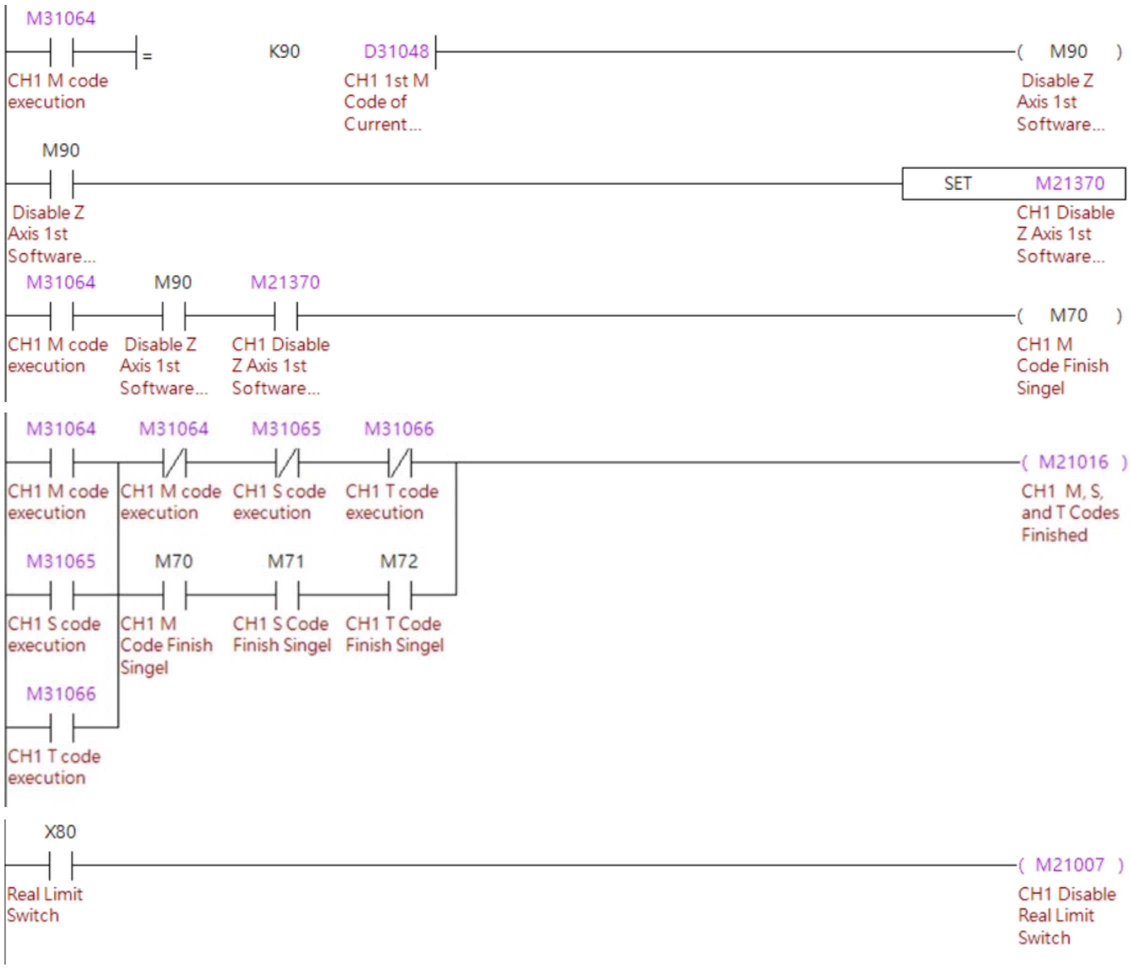
Users can set up the limit sensor type such as normal open or normal close and parameter N2.010 for position or negative limit. The system can correctly identify the hardware limit status of each axis.

6

## ■ 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 | N2.006: The 1st software positive limit is set to 70.<br>N2.008: The 2nd software positive limit is set to 140. |
| Program execution | G90G54X100.<br>G0Z-50.<br>M90<br>G0Z134.<br>M30 |



**Program execution procedure:**

Software limit

1. When the system is executing the main program and executes M90, the M90 is set to ON.
2. When M90 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 mechanism reaches the hardware limit in AUTO, MDI or any other mode of operation. Users 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. Users can move the mechanism 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

## 6.10 Spindle control

On the controller, the spindle is controlled by multiple special M relays and special D registers. The descriptions for forward/reverse operation, positioning, and speed override of the spindle are as follows.

### ■ MLC special D

| Spindle | 1st | 2nd | 3rd | 4th | 5th | 6th | 7th | 8th |
|---|---|---|---|---|---|---|---|---|
| Forward Rotation | M2x704 | M2x720 | M2x736 | M2x752 | M2x768 | M2x784 | M2x800 | M2x816 |
| Reverse Rotation | M2x705 | M2x721 | M2x737 | M2x753 | M2x769 | M2x785 | M2x801 | M2x817 |
| Positioning | M2x706 | M2x722 | M2x738 | M2x754 | M2x770 | M2x786 | M2x802 | M2x818 |
| Speed Reach | M3x704 | M3x720 | M3x736 | M3x752 | M3x768 | M3x784 | M3x800 | M3x816 |
| Zero Speed | M3x705 | M3x721 | M3x737 | M3x753 | M3x769 | M3x785 | M3x801 | M3x817 |
| Target Reach | M3x706 | M3x722 | M3x738 | M3x754 | M3x770 | M3x786 | M3x802 | M3x818 |
| Command Source | M2x710 | M2x726 | M2x742 | M2x758 | M2x774 | M2x790 | M2x806 | M2x822 |
| Speed Control | D2x024 | D2x030 | D2x320 | D2x326 | D2x332 | D2x338 | D2x344 | D2x350 |
| Speed Rate | D2x026 | D2x032 | D2x322 | D2x328 | D2x334 | D2x340 | D2x346 | D2x352 |
| Speed Command | D3x024 | D3x030 | D3x362 | D3x374 | D3x386 | D3x398 | D3x410 | D3x422 |
| Speed Feedback | D3x026 | D3x032 | D3x364 | D3x376 | D3x388 | D3x400 | D3x412 | D3x424 |
| Actual Degree | D3x028 | D3x034 | D3x366 | D3x378 | D3x390 | D3x402 | D3x414 | D3x426 |

**[Forward Rotation, Reverse Rotation] M2x704 ~ M2x705, M2x720 ~ M2x721, M2x736 ~ M2x737, M2x752 ~ M2x753, M2x768 ~ M2x769, M2x784 ~ M2x785, M2x800 ~ M2x801, M2x816 ~ M2x817**

When **[Forward Rotation]** or **[Reverse Rotation]** is ON, the spindle starts rotating forward or reversely referring to the S code data.

**[Spindle Positioning] M2x706, M2x722, M2x738, M2x754, M2x770, M2x786, M2x802, M2x818**

When **[Spindle Positioning]** is ON, the spindle uses the Z pulse as the zero degree and positions based on the setting of N0.1013 (Spindle positioning offset). When the spindle completes positioning, it stops rotating.

**[Spindle Speed Reach] M3x704, M3x720, M3x736, M3x752, M3x768, M3x784, M3x800, M3x816**

When the spindle rotates forward or reversely and reaches the target speed, the system sets **[Spindle Speed Reach]** to ON.

- When the rotation speed changes, **[Spindle Speed Reach]** is set to OFF until the spindle reaches the target speed, when **[Spindle Speed Reach]** is set to ON again.
- When the speed command is 0 and the spindle stops, **[Spindle Speed Reach]** is set to ON.
- The duration when **[Spindle Speed Reach]** is ON is influenced by N0.1018 (Spindle Target Speed Deviation).

**[Spindle Zero Speed] M3x705, M3x721, M3x737, M3x753, M3x769, M3x785, M3x801, M3x817**

When the spindle speed reaches zero or the spindle stops, the system sets **[Spindle Zero Speed]** to ON.

- When the spindle starts rotating forward or reversely, [Spindle Zero Speed] is set to OFF.
- When the rotation command becomes a non-zero value, [Spindle Zero Speed] is set to OFF.

**[Spindle Target Reach] M3x706, M3x722, M3x738, M3x754, M3x770, M3x786, M3x802, M3x818**

When **[Spindle Positioning]** is ON and the spindle completes the positioning based on **N0.1013 (Spindle Positioning Offset)**, the system sets M2258 **[Spindle Target Reach]** to ON.

- When the spindle starts rotating, [Spindle Target Reach] is set to OFF.

**[Spindle Command Source] M2x710, M2x726, M2x742, M2x758, M2x774, M2x790, M2x806, M2x822**

When **[Spindle Command Source]** is ON, the spindle speed command will refer to **[Spindle Speed]**. When **[Spindle Command Source]** is set to OFF, the spindle speed command will refer to the S code as program speed.

**[Spindle Speed] D2x024, D2x030, D2x320, D2x326, D2x332, D2x338, D2x344, D2x350**

When **[Spindle Command Source]** is set to ON, the spindle speed command will refer to **[Spindle Speed]**.

- The applicable input rage is 0 to 2,147,483,647.
- The spindle command speed is limited by parameter N0.1008 (Spindle Maximum Speed).
- When the value of [Spindle Speed] is changed, the spindle speed changes immediately.

**[Spindle Speed Rate] D2x026, D2x032, D2x322, D2x328, D2x334, D2x340, D2x346, D2x352**

When the spindle starts rotating forward or reversely, users can refer to the S code as the programed speed and **[Spindle Speed Rate]** to adjust the spindle speed with the speed override.

- The applicable input rage for **[Spindle Speed Rate]** is 0 to 120.
- The spindle command speed is limited by parameter N0.1008 (Spindle Maximum Speed).
- When the value of **[Spindle Speed Rate]** is changed, the spindle speed changes immediately.

6

**[Speed Command] D3x024, D3x030, D3x362, D3x374, D3x386, D3x398, D3x410, D3x422**

When the controller executes the S code in the program, it writes the value of this S code to **[1st S code data]** as well as **[Speed Command]**.

**[Speed Feedback] D3x026, D3x032, D3x364, D3x376, D3x388, D3x400, D3x412, D3x424**

When the spindle rotates forward or reversely, **[Speed Feedback]** displays the spindle's actual speed.

**[Actual Degree] D3x028, D3x034, D3x366, D3x378, D3x390, D3x402, D3x414, D3x426**

When the spindle rotates forward or reversely, **[Actual Degree]** displays the spindle's actual degree.

## ■ Relevant Parameter

**Spindle setting:**

According to system parameter setting "**Channel Setting**", users can determine whether to enable the spindle function and spindle ID.

**Spindle Function:**

N0.1000 (Spindle Setting):

| Spindle Function | Description |
|---|---|
| Spindle Control Type | 0: Communication Mode<br>1: Reserved<br>2: Analog Mode |
| Analog Close Loop Control | 0: Disable<br>1: Enable |
| Analog Spindle Command Source | 0: Reference from command speed<br>1: Reference from encoder |
| Analog Spindle Encoder Feedback Source | 0: Reference from spindle encoder<br>1: Reference from end motor encoder |
| Spindle Speed Reference | 0: Reference from NC program<br>1: Reference from N0.1006 (Spindle Initial Speed) |
| Spindle High Speed Command Check | 0: Disable<br>1: Enable |
| Multi-Spindle Encoder Feedback Function Switch | 0: Disable<br>1: Enable |
| Spindle Target Speed Error Unit | 0: RPM<br>1: % |

**Speed parameter setting:**

Spindle's rotary speed will be based on the below parameter setting:

N0.1006 (Spindle Initial Speed): S code initialized value.

N0.1008 (Spindle Maximum Speed): When the spindle is set to communication mode, this will limit the spindle maximum speed. When the spindle is set to analog mode, this will be the voltage ratio of the spindle speed.

N0.1009 (Spindle Acc and Dec Time): Spindle acceleration and deceleration time.

N0.1010 (Spindle S Curve Time): Spindle S curve time when acceleration and deceleration.

N0.1017 (Spindle Zero Speed Error): When the speed difference between actual speed and zero speed is lower than parameter N0.1017, the system will set **[Spindle Zero Speed]** to ON automatically.

- When the N0.1017 is 100 and the actual speed is ≤100 RPM, the system will set **[Spindle Zero Speed]** to ON automatically.

N0.1018 (Spindle Target Speed Error): When the speed difference between actual speed and command speed is lower than parameter N0.1018, the system will set [Spindle Speed Reach] to ON automatically.

- When **[Speed Command]** is 1000 and N0.1018 is 100, when the actual speed is 900 to 1100 RPM, the system will set **[Spindle Speed Reach]** to ON automatically.

**Spindle positioning setting:**

N0.1013 (Spindle Positioning Offset): When the positioning command procedure is finished, the system will be based on the position of the Z phase and then add the offset configure from parameter N1.1013.

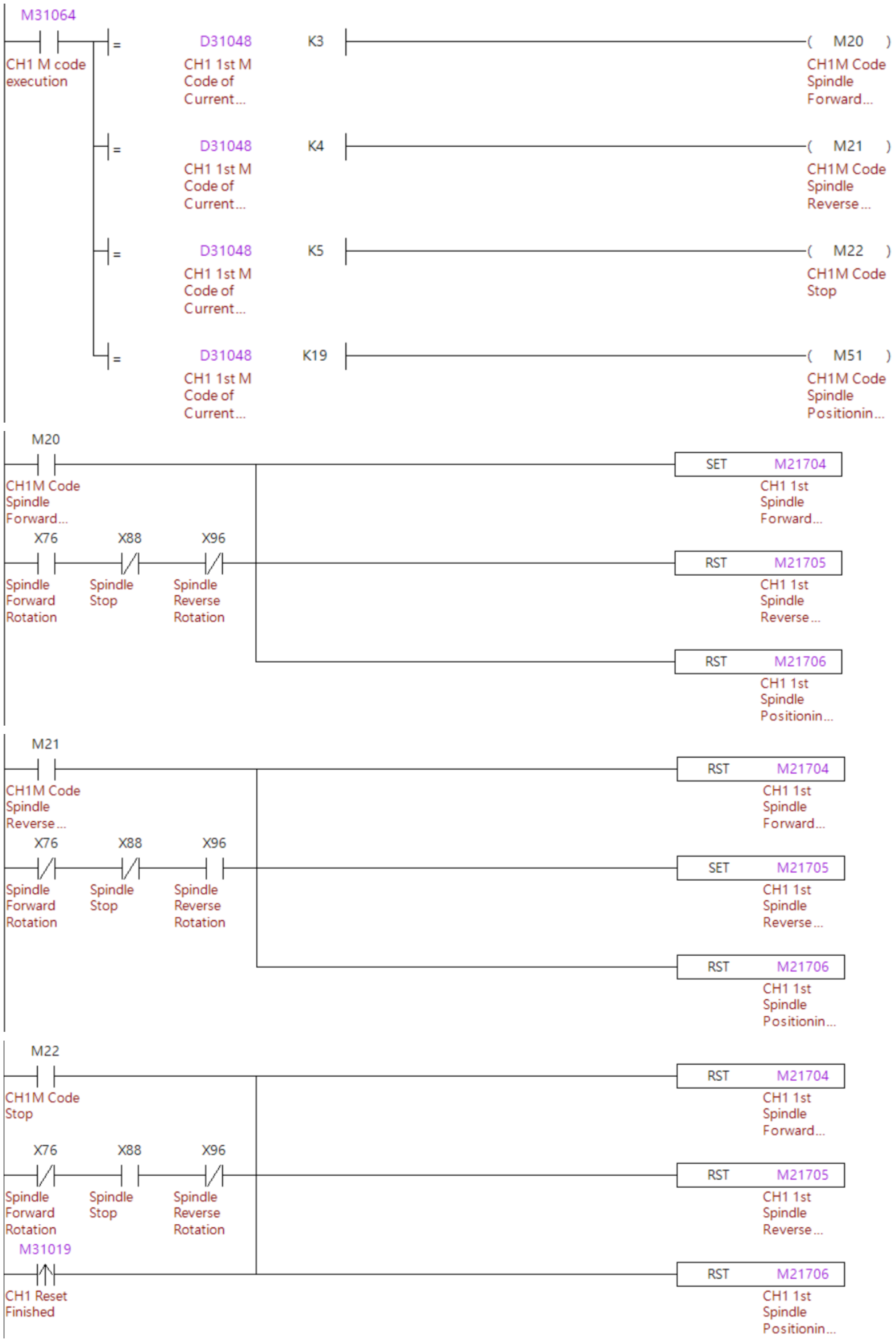
N0.1019 (Spindle Positioning Error): When the spindle is executing positioning procedures and the distance between actual position and N0.1013 (Spindle Positioning Offset) is lower than parameter N0.1019, the system will set **[Spindle Speed Reach]** to ON automatically.

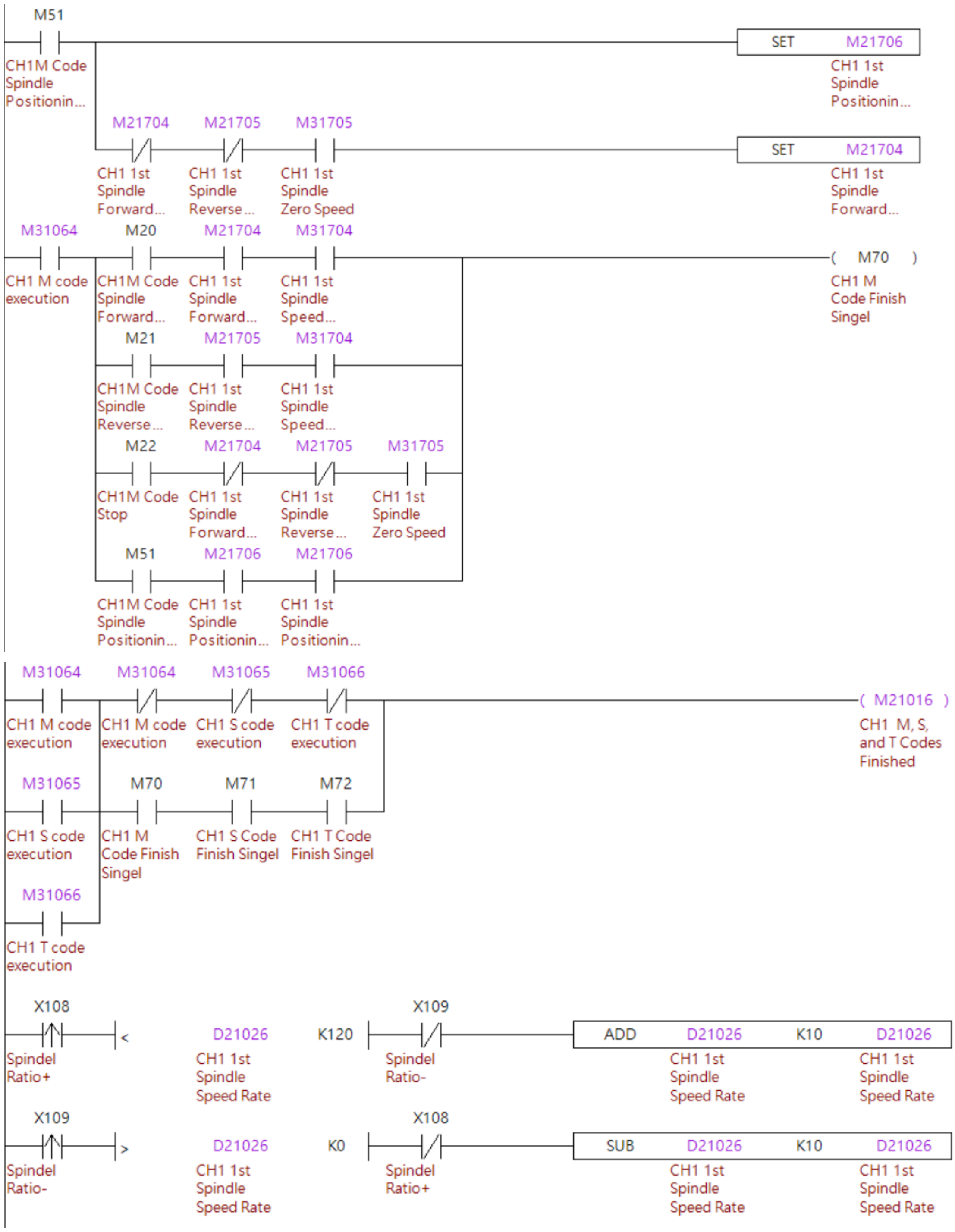
- If the parameter N0.1013 is 1000 and N0.1019 is 500 (Unit: 0.01 degrees), the **[Spindle Speed Reach]** will be ON when the spindle feedback degree is between 5 to 15.

6

## ■ MLC Example

The following illustrates the usage of spindle forward/reverse operation, stop, positioning, and speed override.




M51
CH1M Code Spindle Positionin...
SET M21706
CH1 1st Spindle Positionin...
M21704 M21705 M31705
CH1 1st Spindle Forward...
CH1 1st Spindle Reverse...
CH1 1st Spindle Zero Speed
SET M21704
CH1 1st Spindle Forward...
M31064 M20 M21704 M31704
CH1 M code execution
CH1M Code Spindle Forward...
CH1 1st Spindle Forward...
CH1 1st Spindle Speed...
M70
CH1 M Code Finish Singel
M21 M21705 M31704
CH1M Code Spindle Reverse...
CH1 1st Spindle Reverse...
CH1 1st Spindle Speed...
M22 M21704 M21705 M31705
CH1M Code Stop
CH1 1st Spindle Forward...
CH1 1st Spindle Reverse...
CH1 1st Spindle Zero Speed
M51 M21706 M21706
CH1M Code Spindle Positionin...
CH1 1st Spindle Positionin...
CH1 1st Spindle Positionin...
M31064 M31064 M31065 M31066
CH1 M code execution
CH1 M code execution
CH1 S code execution
CH1 T code execution
M21016
CH1 M, S, and T Codes Finished
M31065 M70 M71 M72
CH1 S code execution
CH1 M Code Finish Singel
CH1 S Code Finish Singel
CH1 T Code Finish Singel
M31066
CH1 T code execution
X108
Spindel Ratio+
< D21026 K120
CH1 1st Spindle Speed Rate
X109
Spindel Ratio-
ADD D21026 K10 D21026
CH1 1st Spindle Speed Rate
CH1 1st Spindle Speed Rate
X109
Spindel Ratio-
> D21026 K0
CH1 1st Spindle Speed Rate
X108
Spindel Ratio+
SUB D21026 K10 D21026
CH1 1st Spindle Speed Rate
CH1 1st Spindle Speed Rate

6

**Program execution procedure**

Forward / reverse operation and stop.

1. If users press the key to spindle forward operation, reverse operation, or stop operation, the corresponding **[Forward Rotation]** or **[Reverse Rotation]** is set to 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 **[Forward Rotation]** or **[Reverse Rotation]** to ON or OFF to have the spindle rotate forward, reversely, or stopped.
3. Confirm the spindle status with **[Speed Reach]** and **[Zero Speed]** and end the M code procedure.

Spindle positioning.

1. When M19 is executed in the program, the M code procedure sets **[Positioning]** to ON.
2. Confirm the spindle positioning is complete with **[Spindle Target Reach]** and end the M code procedure.

Spindle speed rate.

Use the key signal to increase or decrease the spindle speed rate. 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 Rate]**.

## 6.11 Spindle gear ratio switch

In the controller, there are four sets of spindle gear ratio parameters that need to be switched with the MLC. The description of the spindle gear ratio switch is as follows.

### ■ MLC special D register

| | | |
|---|---|---|
| 1st Spindle Gear Ratio Selection | D2x027 | 1: Take N0.1034, N0.1035 as Ratio<br>2: Take N0.1036, N0.1037 as Ratio<br>3: Take N0.1038, N0.1039 as Ratio<br>4: Take N0.1040, N0.1041 as Ratio |
| 2nd Spindle Gear Ratio Selection | D2x033 | 1: Take N0.1084, N0.1085 as Ratio<br>2: Take N0.1086, N0.1087 as Ratio<br>3: Take N0.1088, N0.1089 as Ratio<br>4: Take N0.1090, N0.1091 as Ratio |
| 3rd Spindle Gear Ratio Selection | D2x323 | 1: Take N0.1134, N0.1135 as Ratio<br>2: Take N0.1136, N0.1137 as Ratio<br>3: Take N0.1138, N0.1139 as Ratio<br>4: Take N0.1140, N0.1141 as Ratio |
| 4th Spindle Gear Ratio Selection | D2x329 | 1: Take N0.1184, N0.1185 as Ratio<br>2: Take N0.1186, N0.1187 as Ratio<br>3: Take N0.1188, N0.1189 as Ratio<br>4: Take N0.1190, N0.1191 as Ratio |
| 5th Spindle Gear Ratio Selection | D2x335 | 1: Take N0.1234, N0.1235 as Ratio<br>2: Take N0.1236, N0.1237 as Ratio<br>3: Take N0.1238, N0.1239 as Ratio<br>4: Take N0.1240, N0.1241 as Ratio |
| 6th Spindle Gear Ratio Selection | D2x341 | 1: Take N0.1284, N0.1285 as Ratio<br>2: Take N0.1286, N0.1287 as Ratio<br>3: Take N0.1288, N0.1289 as Ratio<br>4: Take N0.1290, N0.1291 as Ratio |
| 7th Spindle Gear Ratio Selection | D2x347 | 1: Take N0.1334, N0.1335 as Ratio<br>2: Take N0.1336, N0.1337 as Ratio<br>3: Take N0.1338, N0.1339 as Ratio<br>4: Take N0.1340, N0.1341 as Ratio |
| 8th Spindle Gear Ratio Selection | D2x353 | 1: Take N0.1384, N0.1385 as Ratio<br>2: Take N0.1386, N0.1387 as Ratio<br>3: Take N0.1388, N0.1389 as Ratio<br>4: Take N0.1390, N0.1391 as Ratio |

**[Spindle Gear Ratio Selection] D2x027, D2x033, D2x323, D2x329, D2x335, D2x341, D2x347, D2x353**

When users need to switch the spindle gear ratio, set **[Spindle Gear Ratio Selection]** to the group 1 to 4 as preferred. (As shown in the above table)

## Relevant Parameter

**Spindle gear ratio setting:**

N0.1034 ~ N0.1041: When users set **[1st Spindle Gear Ratio Selection]** and the ratio can be defined as below parameters.

- N0.1034 (Numerator of 1st Gear Ratio) and N0.1035 (Denominator of 1st Gear Ratio)
- N0.1036 (Numerator of 2nd Gear Ratio) and N0.1037 (Denominator of 2nd Gear Ratio)
- N0.1038 (Numerator of 3rd Gear Ratio) and N0.1039 (Denominator of 3rd Gear Ratio)
- N0.1040 (Numerator of 4th Gear Ratio) and N0.1041 (Denominator of 4th Gear Ratio)

N0.1084 ~ N0.1091: When users set **[2nd Spindle Gear Ratio Selection]** and the ratio can be defined as below parameters.

- N0.1084 (Numerator of 1st Gear Ratio) and N0.1085 (Denominator of 1st Gear Ratio)
- N0.1086 (Numerator of 2nd Gear Ratio) and N0.1087 (Denominator of 2nd Gear Ratio)
- N0.1088 (Numerator of 3rd Gear Ratio) and N0.1089 (Denominator of 3rd Gear Ratio)
- N0.1090 (Numerator of 4th Gear Ratio) and N0.1091 (Denominator of 4th Gear Ratio)

N0.1134 ~ N0.1141: When users set **[3rd Spindle Gear Ratio Selection]** and the ratio can be defined as below parameters.

- N0.1134 (Numerator of 1st Gear Ratio) and N0.1135 (Denominator of 1st Gear Ratio)
- N0.1136 (Numerator of 2nd Gear Ratio) and N0.1137 (Denominator of 2nd Gear Ratio)
- N0.1138 (Numerator of 3rd Gear Ratio) and N0.1139 (Denominator of 3rd Gear Ratio)
- N0.1140 (Numerator of 4th Gear Ratio) and N0.1141 (Denominator of 4th Gear Ratio)

N0.1184 ~ N0.1191: When users set **[4th Spindle Gear Ratio Selection]** and the ratio can be defined as below parameters.

- N0.1184 (Numerator of 1st Gear Ratio) and N0.1185 (Denominator of 1st Gear Ratio)
- N0.1186 (Numerator of 2nd Gear Ratio) and N0.1187 (Denominator of 2nd Gear Ratio)
- N0.1188 (Numerator of 3rd Gear Ratio) and N0.1189 (Denominator of 3rd Gear Ratio)
- N0.1190 (Numerator of 4th Gear Ratio) and N0.1191 (Denominator of 4th Gear Ratio)

N0.1234 ~ N0.1241: When users set **[5th Spindle Gear Ratio Selection]** and the ratio can be defined as below parameters.

- N0.1234 (Numerator of 1st Gear Ratio) and N0.1235 (Denominator of 1st Gear Ratio)
- N0.1236 (Numerator of 2nd Gear Ratio) and N0.1237 (Denominator of 2nd Gear Ratio)
- N0.1238 (Numerator of 3rd Gear Ratio) and N0.1239 (Denominator of 3rd Gear Ratio)
- N0.1240 (Numerator of 4th Gear Ratio) and N0.1241 (Denominator of 4th Gear Ratio)

N0.1284 ~ N0.1291: When users set **[6th Spindle Gear Ratio Selection]** and the ratio can be defined as below parameters.

- N0.1284 (Numerator of 1st Gear Ratio) and N0.1285 (Denominator of 1st Gear Ratio)
- N0.1286 (Numerator of 2nd Gear Ratio) and N0.1287 (Denominator of 2nd Gear Ratio)
- N0.1288 (Numerator of 3rd Gear Ratio) and N0.1289 (Denominator of 3rd Gear Ratio)
- N0.1290 (Numerator of 4th Gear Ratio) and N0.1291 (Denominator of 4th Gear Ratio)

N0.1334 ~ N0.1341: When users set **[7th Spindle Gear Ratio Selection]** and the ratio can be defined as below parameters.

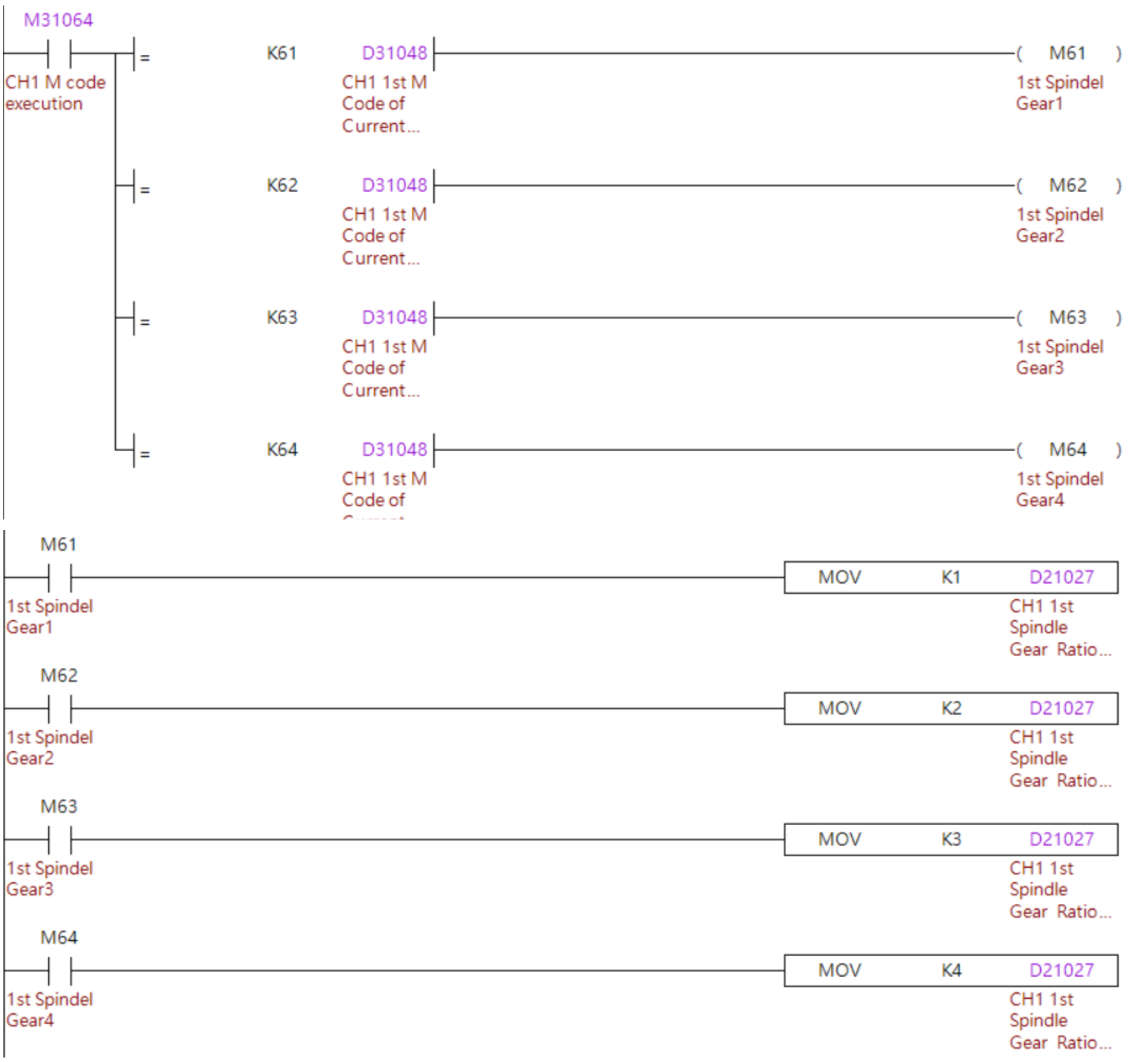
- N0.1334 (Numerator of 1st Gear Ratio) and N0.1335 (Denominator of 1st Gear Ratio)
- N0.1336 (Numerator of 2nd Gear Ratio) and N0.1337 (Denominator of 2nd Gear Ratio)
- N0.1338 (Numerator of 3rd Gear Ratio) and N0.1339 (Denominator of 3rd Gear Ratio)
- N0.1340 (Numerator of 4th Gear Ratio) and N0.1341 (Denominator of 4th Gear Ratio)

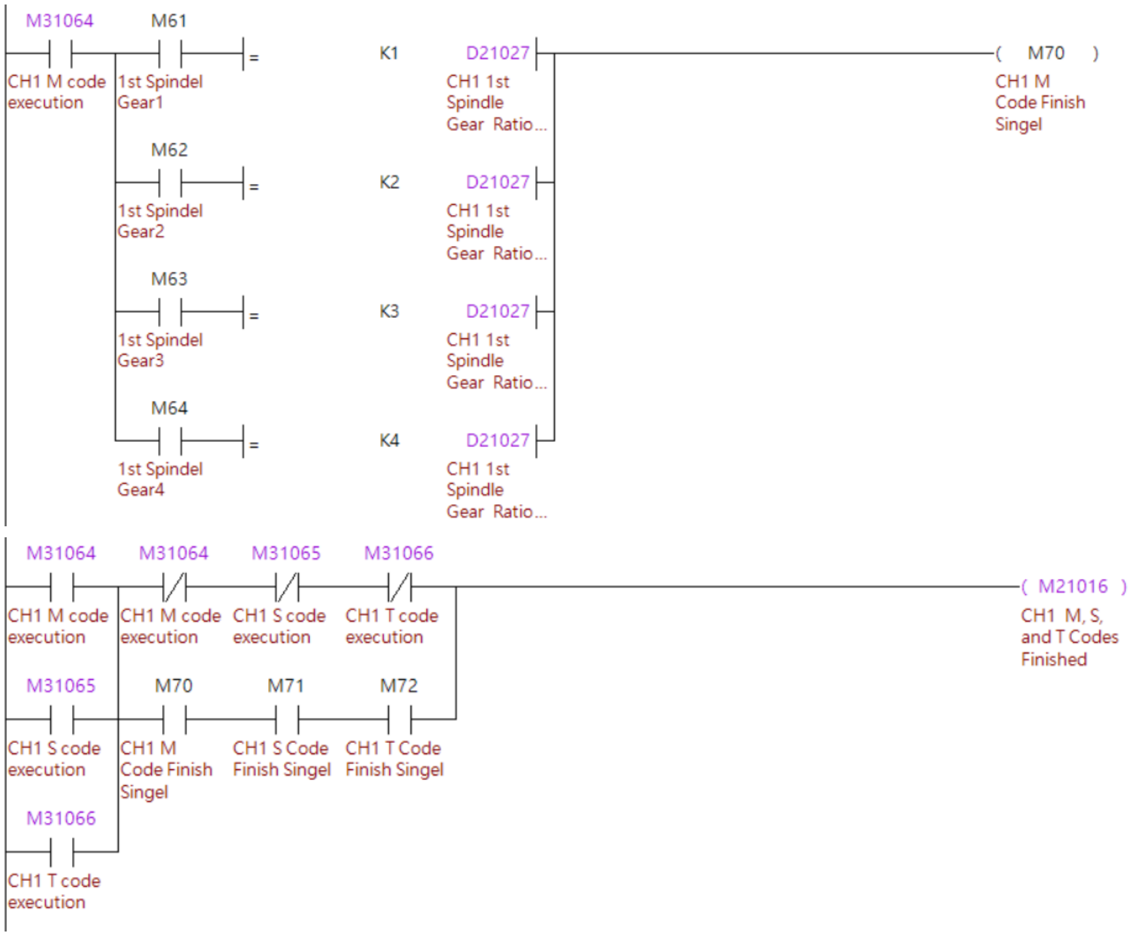
N0.1384 ~ N0.1391: When users set **[8th Spindle Gear Ratio Selection]** and the ratio can be defined as below parameters.

- N0.1384 (Numerator of 1st Gear Ratio) and N0.1385 (Denominator of 1st Gear Ratio)
- N0.1386 (Numerator of 2nd Gear Ratio) and N0.1387 (Denominator of 2nd Gear Ratio)
- N0.1388 (Numerator of 3rd Gear Ratio) and N0.1389 (Denominator of 3rd Gear Ratio)
- N0.1390 (Numerator of 4th Gear Ratio) and N0.1391 (Denominator of 4th 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]**. An example of using the M code to switch the gear ratio is as follows.






**Program execution procedure:**

1. When the NC program executes M61, the MLC will set 1 to **[1st Spindle Gear Ratio Selection].**
2. In this example the MLC uses the M61 relay to switch the ratio and check the status. After the process is finished, the MLC will acknowledge system M code finished.
3. When the NC program executes M62, MLC will set 2 to **[1st Spindle Gear Ratio Selection].** In this example the MLC uses the M62 relay to switch ratio and check status.
4. After the process is finished, the MLC will acknowledge system M code finished.
5. When the NC program executes M63, MLC will set 3 to **[1st Spindle Gear Ratio Selection].** In this example the MLC uses the M63 relay to switch ratio and check status.
6. After the process is finished, the MLC will acknowledge system M code finished.
7. When the NC program executes M64, MLC will set 4 to **[1st Spindle Gear Ratio Selection].** In this example the MLC uses the M64 relay to switch ratio and check status.
8. After the process is finished, the MLC will acknowledge system M code finished.

**Important:**

If you use the M code to switch the gear ratio with **[Spindle Gear Ratio Selection],** it only switches the spindle speed command. If there is a physical mechanical part for the gear switch, you need to compose the corresponding MLC and the output DO, so the external mechanical part can correctly change the gear ratio.

## 6.12 One-button macro call

The one-button 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 D

| | | | |
|---|---|---|---|
| Macro Call Activation | M2x025 | Macro Call Status | M3x027 |
| 1st Macro Call Preparation | M2x032 | Macro Call Ready | M3x028 |
| 2nd Macro Call Preparation | M2x033 | Macro Call Error | M3x029 |
| 3rd Macro Call Preparation | M2x034 | 1st Macro Call Initial Finished | M3x048 |
| 4th Macro Call Preparation | M2x035 | 2nd Macro Call Initial Finished | M3x049 |
| 5th Macro Call Preparation | M2x036 | 3rd Macro Call Initial Finished | M3x050 |
| 6th Macro Call Preparation | M2x037 | 4th Macro Call Initial Finished | M3x051 |
| 7th Macro Call Preparation | M2x038 | 5th Macro Call Initial Finished | M3x052 |
| 8th Macro Call Preparation | M2x039 | 6th Macro Call Initial Finished | M3x053 |
| 9th Macro Call Preparation | M2x040 | 7th Macro Call Initial Finished | M3x054 |
| 10th Macro Call Preparation | M2x041 | 8th Macro Call Initial Finished | M3x055 |
| 11th Macro Call Preparation | M2x042 | 9th Macro Call Initial Finished | M3x056 |
| 12th Macro Call Preparation | M2x043 | 10th Macro Call Initial Finished | M3x057 |
| 13th Macro Call Preparation | M2x044 | 11th Macro Call Initial Finished | M3x058 |
| 14th Macro Call Preparation | M2x045 | 12th Macro Call Initial Finished | M3x059 |
| 15th Macro Call Preparation | M2x046 | 13th Macro Call Initial Finished | M3x060 |
| 16th Macro Call Preparation | M2x047 | 14th Macro Call Initial Finished | M3x061 |
| - | - | 15th Macro Call Initial Finished | M3x062 |
| - | - | 16th Macro Call Initial Finished | M3x063 |

| | | | |
|---|---|---|---|
| 1st Macro Call Macro Number | D2x064 | 9th Macro Call Macro Number | D2x072 |
| 2nd Macro Call Macro Number | D2x065 | 10th Macro Call Macro Number | D2x073 |
| 3rd Macro Call Macro Number | D2x066 | 11th Macro Call Macro Number | D2x074 |
| 4th Macro Call Macro Number | D2x067 | 12th Macro Call Macro Number | D2x075 |
| 5th Macro Call Macro Number | D2x068 | 13th Macro Call Macro Number | D2x076 |
| 6th Macro Call Macro Number | D2x069 | 14th Macro Call Macro Number | D2x077 |
| 7th Macro Call Macro Number | D2x070 | 15th Macro Call Macro Number | D2x078 |
| 8th Macro Call Macro Number | D2x071 | 16th Macro Call Macro Number | D2x079 |

**[Macro Call Activation] M2x025**

When the controller is in AUTO mode, first set **[Macro Call Initial Finished]** to ON and then set **[Macro Call Activation]** to ON, and the system will execute the O macro corresponding to the number of **[Macro Call Macro Number]**.

**[Macro Call Preparation] M2x032 ~ M2x047**

When users set **[Macro Call Preparation]** to ON, the system will start to load the O macro and file name as **[Macro Call Macro Number].**

- The corresponding O macro must be stored in the correct channel folder of folder **[O_MACRO]** or the corresponding macro of the location INTER.

**[Macro Call Status] M3x027**

When the user sets **[Macro Call Preparation]** to ON, the system will set **[Macro Call Status]** to ON right away.

**[Macro Call Ready] M3x028**

Regardless of whether the controller system is in AUTO mode, the system will set **[Macro Call Ready]** to ON after it finishes the macro initialization and then set **[Macro Call Initial Finished]** to ON. This is in order to notify users to switch the system mode to AUTO, which is able to execute the macro program.

- When the **[Macro Call Activation]** is ON, the **[Macro Call Ready]** will change to OFF.
- After the **[Macro Call Ready]** is ON, it will reset to OFF once the **[NC Reset]** has triggered.

**[Macro Call Error] M3x029**

The **[Macro Call Error]** will be set to ON when the **[Macro Call Preparation]** has triggered and the system is not in the AUTO mode or **[Macro Call Macro Number]** is 0.

**[Macro Call Initial Finished] M3x048 ~ M3x063**

When the **[Macro Call Preparation]** is ON, the system will start to prepare the macro and set **[Macro Call Initial Finished]** to ON right after the process is finished.

- After the **[Macro Call Initial Finished]** is ON, the **[Macro Call Initial Finished]** will reset to OFF once the **[NC Reset]** has triggered.

**[Macro Call Macro Number] D2x064 ~ D2x079**

When the **[Macro Call Preparation]** is ON, the system will access the O macro to refer to the value of **[Macro Call Macro Number].**

- Available range between 1 to 65535.
- When the **[Macro Call Macro Number]** is set to 10000, the system will load O10000. Or, when it is set to 12345, the system will load O12345.
- When the value is set between 1 to 9999, the system will refer to the parameter N8.022 (Macro Call File Source) as a different location.
  a. When N8.022 = 0, the system will load the macro that is stored in the INTER location's O_MACRO folder of the correct channel folder.
  b. When N8.022 = 1, the system will load the macro that is stored in the SD card location's O_MACRO folder of the correct channel folder.
- When the value is set between 10000 to 65535, the system will refer to the parameter N8.022 (Macro Call File Source) as a different root location.
  a. When N8.022 = 0, the system will load the macro that is stored in the root INTER location of the correct channel folder.
  b. When N8.022 = 1, the system will load the macro that is stored in the root SD card location of the correct channel folder.

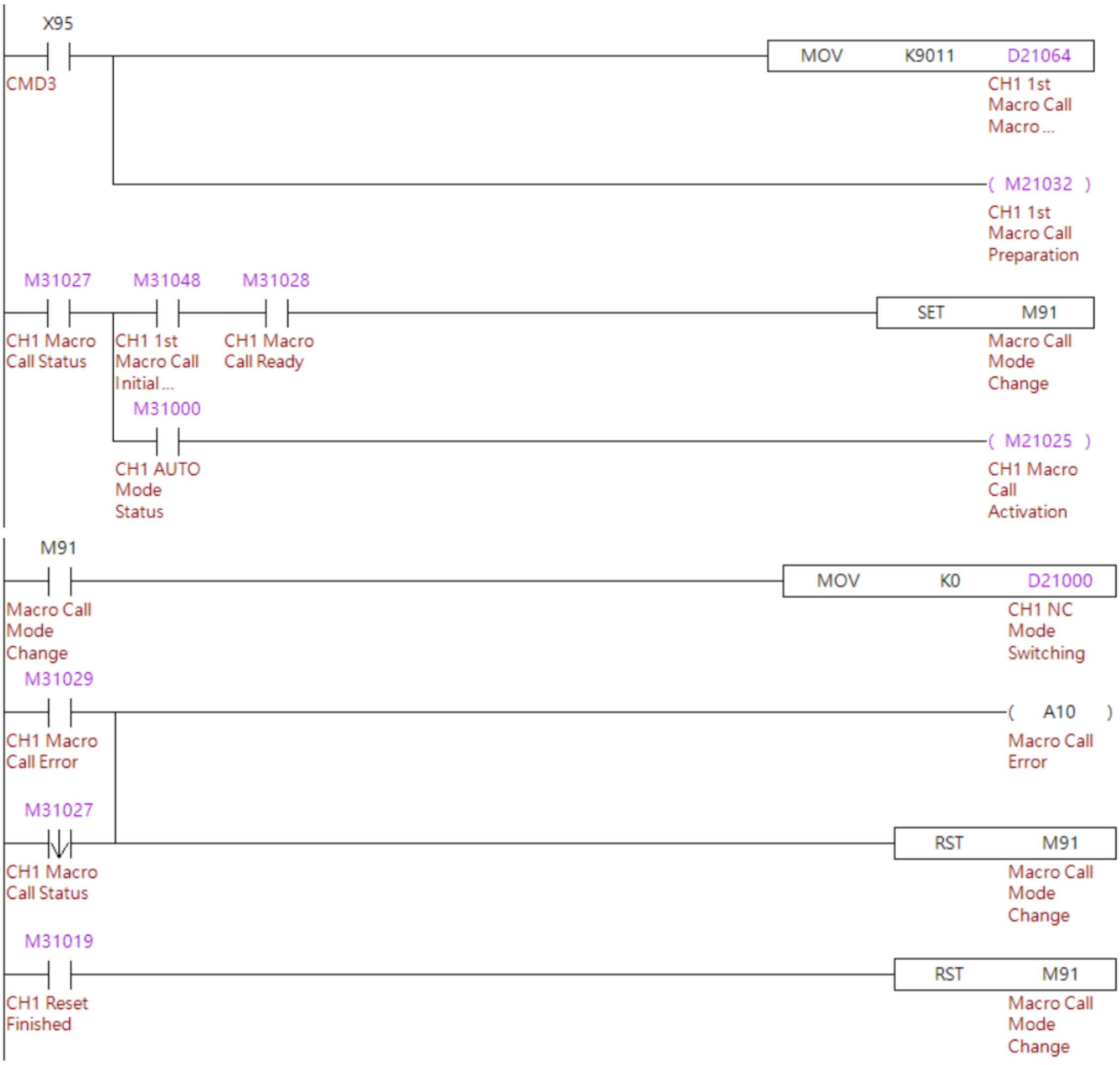
## ■ Relevant Parameter

**Setting macro call file source:**

N8.022 (Macro Call File Source): the O macro file source to call.

- **[Macro Call Macro Number]** value between 1 to 9999:
  When N8.022 = 0, the system will load the macro that is stored in the INTER location's O_MACRO folder of the correct channel folder.
  When N8.022 = 1, the system will load the macro that is stored in the SD card location's O_MACRO folder of the correct channel folder.
- **[Macro Call Macro Number]** value between 10000 to 65535:
  When N8.022 = 0, the system will load the macro that is stored in the root INTER location of the correct channel folder.
  When N8.022 = 1, the system will load the macro that is stored in the root SD card location of the correct channel folder.

## ■ MLC Example




**Program execution procedure:**

1. Call the specific macros with the X95 signals.
2. When X95 is on, the MLC will give a value to D21064 and then set **[Macro Call Preparation]** to ON. The system will set **[Macro Call Status]** to ON automatically.

3. When the system finishes the macro preparation, it will set **[Macro Call Initial Finished]** to ON and then set **[Macro Call Status]** to ON as well. Once this signal status is finished, in this example MLC will set M91 to ON and then switch the system to AUTO mode and activate the **[Macro Call Activation]**.
4. After the macro is finished, the system will set **[Macro Call Status]** to OFF and then MLC will set M91 to OFF.

**Important:**

Executing this one-button macro call function is not recommended when the system is in AUTO or MDI mode. This is because during the macro preparation, the system will be initializing some system inner status, which could affect the macro start in unexpected ways.

## 6.13 Tool magazine control with I/O

The controller changes tools with an external tool exchanger and transmits data or status with the I/O. The following describes how to control the tool magazine with the I/O.

### ■ MLC special D

| Tool Magazine 1 | | Tool Magazine 2 | |
|---|---|---|---|
| Tool Magazine 1 Move Forward | M2x064 | Tool Magazine 2 Move Forward | M2x072 |
| Tool Magazine 1 Move Backward | M2x065 | Tool Magazine 2 Move Backward | M2x073 |
| Tool 1 Exchange | M2x066 | Tool 2 Exchange | M2x074 |
| Tool Magazine 1 Reset | M2x067 | Tool Magazine 2 Reset | M2x075 |
| Current Tool Number Tool Magazine 1 | D3x036 | Current Tool Number Tool Magazine 2 | D3x042 |
| Standby Tool Number Tool Magazine 1 | D3x037 | Standby Tool Number Tool Magazine 2 | D3x043 |
| Standby Tool Pot Tool Magazine 1 | D3x038 | Standby Tool Pot Tool Magazine 2 | D3x044 |
| Tool Pot Deviation Tool Magazine 1 | D3x039 | Tool Pot Deviation Tool Magazine 2 | D3x045 |

**[Tool Magazine Move Forward] M2x064, M2x072**

When **[Tool Magazine Move 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 and set **[Tool Magazine Move Forward]** to ON again, the standby tool pot number and standby tool number will become 1.

- The operation is the same for both tool magazine 1 and tool magazine 2.

**[Tool Magazine Move Backward] M2x065, M2x073**

When **[Tool Magazine Move 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 and set **[Tool Magazine Move Backward]** to ON again, the standby tool pot number and standby tool number will become the maximum numbers.

- The operation is the same for both tool magazine 1 and tool magazine 2.

**[Tool Exchange] M2x066, M2x074**

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] M2x067, M2x075**

To reset the tool magazine, in addition to using the tool setting function in the OFS screen, users 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 Reset]** is triggered, the spindle tool number refers to the **[Continue]** setting in the **[Tool Magazine Setting Component]**.
  When **[Continue]** is 0, the current spindle tool number will be 0 after reset. When **[Continue]** is 1, the current spindle tool number will refer to the settings of **[StartNumber]** and **[PotAmount]**, which will increase maximum tool number by 1 (where tool numbers are arranged in ascending order).
  For example, when the tool magazine 1 sets **[StartNumber]** to 3, **[PotAmount]** to 16 and **[Continue]** to 1, the current tool number will be 19, 1$^{st}$ tool pot will be 3 and 2$^{nd}$ tool pot will be 4.
- After reset, the system will set default standby tool pot to **[StandbyPot]** in the **[Tool Magazine Setting Component]**.
- After reset, the system will set command tool number to **[StartNumber]** in the **[Tool Magazine Setting Component]** and set standby tool number to the command tool number plus 1. Then, all the other tool pots in this tool magazine will increase by 1 in the tool number setting.
- The operation is the same for both tool magazine 1 and tool magazine 2.

**[Current Tool Number] D3x036, D3x042**

This shows the current in use tool number according to the corresponding **[Current Tool Number]** based on the channel and tool magazine.

**[Standby Tool Number] D3x037, D3x043**

When the controller is in AUTO or MDI mode, once the T code block has been executed, the system will execute the tool change procedure. At the same time, the tool number will be set to the **[Standby Tool Number]**, which is based on the setting of **[Tool Pot Amount]** in the **[Tool Magazine Setting Component]** (as shown in the relevant parameters).

- The operation is the same for both tool magazine 1 and tool magazine 2.

**[Standby Tool Pot] D3x038, D3x044**

This shows the current standby tool pot number according to the corresponding **[Standby Tool Pot]** based on the channel and tool magazine.

**[Tool Pot Deviation] D3x039, D3x045**

This shows the difference between the current tool number and command tool number. Users can rotate the tool magazine forward or backward through positive or negative values here. After the **[Tool Magazine Move Forward] or [Tool Magazine Move Backward]** is ON, the system will update the new difference to this **[Tool Pot Deviation].**

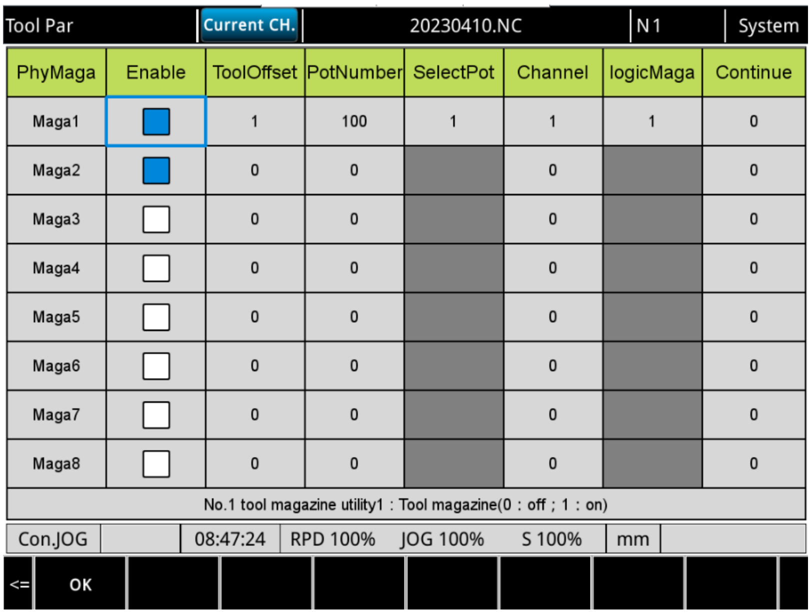
- The operation is the same for both tool magazine 1 and tool magazine 2.

6

## ■ Relevant Parameter

**Tool magazine setting component:**

Tool magazines relate parameters are in this **[Tool Magazine Setting Component]** page. Users can set the machine's requirements as described below.




| PhyMaga | Enable | ToolOffset | PotNumber | SelectPot | Channel | logicMaga | Continue |
|---|---|---|---|---|---|---|---|
| Maga1 | ☑ | 1 | 100 | 1 | 1 | 1 | 0 |
| Maga2 | ☑ | 0 | 0 | | 0 | | 0 |
| Maga3 | ☐ | 0 | 0 | | 0 | | 0 |
| Maga4 | ☐ | 0 | 0 | | 0 | | 0 |
| Maga5 | ☐ | 0 | 0 | | 0 | | 0 |
| Maga6 | ☐ | 0 | 0 | | 0 | | 0 |
| Maga7 | ☐ | 0 | 0 | | 0 | | 0 |
| Maga8 | ☐ | 0 | 0 | | 0 | | 0 |

**PhyMaga**: the actual physical sequence of the tool magazine.

**Enable**: to determine whether to use this tool magazine.

- If users execute a T code that is disabled, the system will return an alarm to acknowledge.

**SrartNumber**: to determine the beginning tool number after tool magazine reset.

- After reset, the system will set the command tool number to **[StartNumber]** and set the standby tool number to the command tool number plus 1. The tool magazine is based on the **[PotAmount]** and the allocated number is increased starting from the **[StartNumber]**.

**PotAmount**: Set the total tool number of the tool magazine.

**StandbyPot**: Set the standby tool pot number after tool magazine reset.

- After tool magazine reset, the standby tool pot number will be 2 if this setting is 2.

**Channel**: Set the channel of the physical tool magazine.

- When set to 1, this physical tool magazine will be allocated to channel 1.

**LogicMaga**: Set the sequence number of the physical tool magazine.

- When set to 2, this tool magazine will be the 2nd logic magazine in the channel.

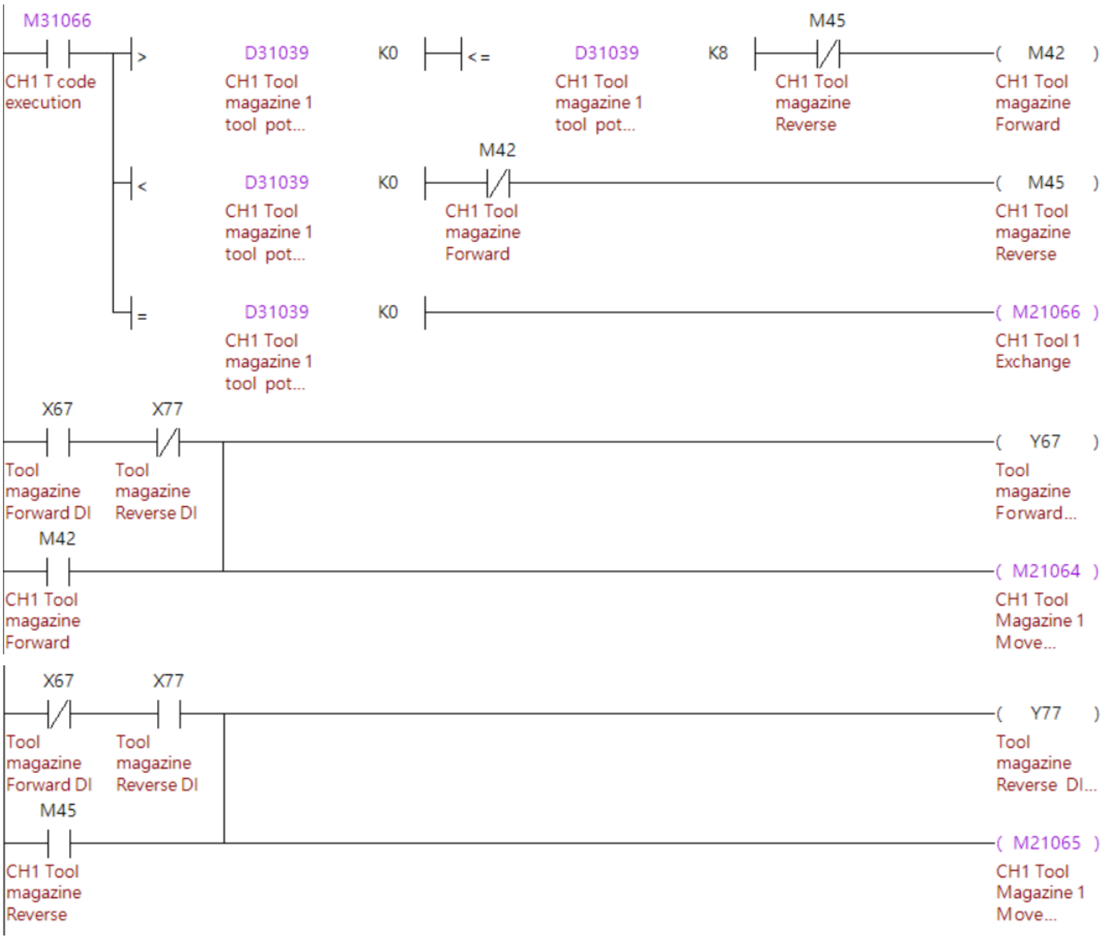
**Continue**: Set the current tool number after the tool magazine has been reset.

- When **[Continue]** is 0, the current spindle tool number will be 0 after reset. When **[Continue]** is 1, the current spindle tool number will refer to the settings of **[StartNumber]** and **[PotAmount]**, which will apply the maximum tool number increases by 1 (where tool numbers are arranged in ascending order).

6

## ■ MLC Example

The following example illustrates the program execution with tool magazine 1 set to hold 16 tools.




**Program execution procedure:**

1. When reading the T code in the program, the system sets **[T code execution]** to ON and automatically write the difference between the current standby tool pot and command tool pot to **[Tool Pot Deviation]**. 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.
2. The system determines in which direction and how many positions the tool magazine must move by performing logic statements. Then, it sets **[Tool Magazine Move Forward]** or **[Tool Magazine Move Backward]** to ON by referring to the corresponding signals such as tool magazine moves forward and external tool counting.
3. When the value of **[Tool Pot Deviation]** is 0, the tool data is exchanged. Additionally, the tools of an external mechanical part can be changed with the MLC.

## 6.14 MLC axes control

Users 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, users can perform position control, speed control, and applications requiring rotations like the spindle rotation or positioning control.

### ■ MLC special D

| Axis | Trigger Movement | Command Type | Control Mode | Target Position | Target Velocity | Target Reached | Axis Moving |
|---|---|---|---|---|---|---|---|
| X Axis | M2x448 | M2x464 | M2x416 | D2x256 | D2x288 | M3x448 | M3x464 |
| Y Axis | M2x449 | M2x465 | M2x417 | D2x258 | D2x290 | M3x449 | M3x465 |
| Z Axis | M2x450 | M2x466 | M2x418 | D2x260 | D2x292 | M3x450 | M3x466 |
| A Axis | M2x451 | M2x467 | M2x419 | D2x262 | D2x294 | M3x451 | M3x467 |
| B Axis | M2x452 | M2x468 | M2x420 | D2x264 | D2x296 | M3x452 | M3x468 |
| C Axis | M2x453 | M2x469 | M2x421 | D2x266 | D2x298 | M3x453 | M3x469 |
| U Axis | M2x454 | M2x470 | M2x422 | D2x268 | D2x300 | M3x454 | M3x470 |
| V Axis | M2x455 | M2x471 | M2x423 | D2x270 | D2x302 | M3x455 | M3x471 |
| W Axis | M2x456 | M2x472 | M2x424 | D2x272 | D2x304 | M3x456 | M3x472 |
| 10th Axis | M2x457 | M2x473 | M2x425 | D2x274 | D2x306 | M3x457 | M3x473 |
| 11th Axis | M2x458 | M2x474 | M2x426 | D2x276 | D2x308 | M3x458 | M3x474 |
| 12th Axis | M2x459 | M2x475 | M2x427 | D2x278 | D2x310 | M3x459 | M3x475 |
| 13th Axis | M2x460 | M2x476 | M2x428 | D2x280 | D2x312 | M3x460 | M3x476 |
| 14th Axis | M2x461 | M2x477 | M2x429 | D2x282 | D2x314 | M3x461 | M3x477 |
| 15th Axis | M2x462 | M2x478 | M2x430 | D2x284 | D2x316 | M3x462 | M3x478 |
| 16th Axis | M2x463 | M2x479 | M2x431 | D2x286 | D2x318 | M3x463 | M3x479 |

| Axis | NC / MLC Axis Switching | Switch to MLC Axis Finished |
|---|---|---|
| X Axis | M2x432 | M3x432 |
| Y Axis | M2x433 | M3x433 |
| Z Axis | M2x434 | M3x434 |
| A Axis | M2x435 | M3x435 |
| B Axis | M2x436 | M3x436 |
| C Axis | M2x437 | M3x437 |
| U Axis | M2x438 | M3x438 |
| V Axis | M2x439 | M3x439 |
| W Axis | M2x440 | M3x440 |
| 10th Axis | M2x441 | M3x441 |
| 11th Axis | M2x442 | M3x442 |
| 12th Axis | M2x443 | M3x443 |
| 13th Axis | M2x444 | M3x444 |
| 14th Axis | M2x445 | M3x445 |
| 15th Axis | M2x446 | M3x446 |
| 16th Axis | M2x447 | M3x447 |

**[Trigger Movement] M2x448 ~ M2x463**

When the axis is in the MLC control mode and users trigger the **[Trigger Movement]** to ON, the specific axis will move based on the position command setting. Once the **[Trigger Movement]** is set to OFF the axis will stop actions.

- Before triggering the **[Trigger Movement]**, users must set **[Target Position]** and **[Target Velocity]** at least one MLC scan cycle earlier.
- If the **[Target Position]** is changed during the MLC axis movement, the new target position will be available until the next **[Trigger Movement]** command is triggered.
- If the **[Target Position]** changed one MLC scan cycle before the **[Trigger Movement]** command is triggered, this new target position will be activated and available.

**[MLC Command Type] M2x464 ~ M2x479**

When the axis is in the MLC position control mode, this **[MLC Command Type]** can switch the **[Target Position]** as an absolute or relative command. Once the **[MLC Command Type]** is OFF, axes will move based on the absolute command and work on the machine coordinate system.

On the other hand, when the **[MLC Command Type]** is ON, axes will move based on the relative command.

**[MLC Control Mode] M2x416 ~ M2x431**

The controller will be using position control mode when **[MLC Control Mode]** is OFF and then using **[Target Position]** and **[Target Velocity]** for the target position and speed command. On the other hand, when the **[MLC Control Mode]** is ON, the system will take **[Target Velocity]** as the target speed and then control the axis with a continuous rotary speed.

**[MLC Target Position] D2x256 ~ D2x286**

When the axis is in the MLC position control mode, the MLC axes can be commanded in absolute or relative mode.

In absolute command mode, the **[Target Position]** will be the position command based on machine coordinate system. In relative command mode, the **[Target Position]** will be the position command, which is the actual movement after motion is triggered.

- Attention: this special D uses float format; thus, it will take two sequence special D addresses.
- The **[Target Position]** will not be available when the axis is in the velocity mode.

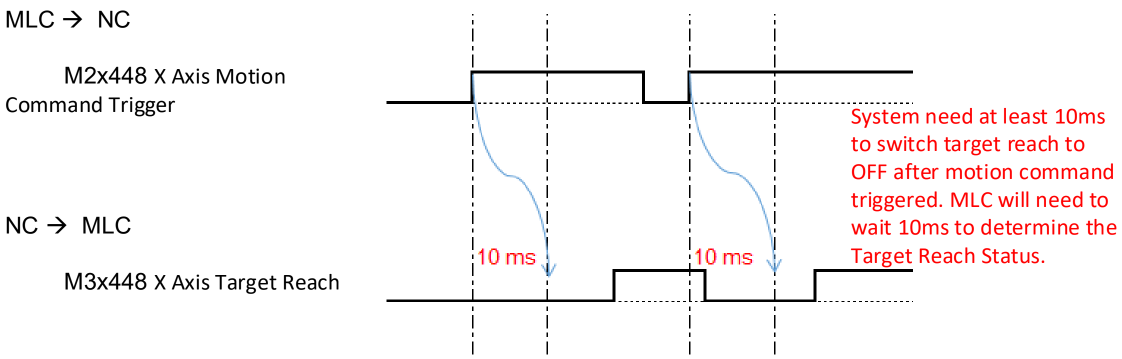
**[MLC Target Velocity] D2x288 ~ D2x318**

When any of the X, Y or Z axis is in the MLC velocity control mode, the axis will be defined as a linear axis and the velocity command unit will only accept mm/min. When other axes such as A, B, C, U, V, W to 16th axis are in the MLC velocity control mode, the axis can define as linear axis or rotary axis from parameter N2.001 Bit2 to 4 (Axis Command Mode). If the axis is defined as a rotary axis, the velocity command unit can be switched in the parameter N2.001 Bit11 as RPM or deg/min.

- When the MLC axis is in the position mode, the system will be using **[Target Position]** and **[Target Velocity]** for the motion command.
- When the MLC axis is in the velocity mode, the system will be using **[Target Velocity]** for the motion command.
- Attention: this special D uses float format; thus, it will take two sequence special D addresses.
- The value of **[Target Velocity]** is available to change and is active during motion once the MLC scans the new value command.

**[Target Reached] M3x448 ~ M3x463**

- When the axis is in the MLC position control mode, the **[Target Reached]** status reflects whether the axis has finished the command and reached the command position.
- When the axis is in the MLC velocity control mode, the **[Target Reached]** status reflects whether the axis has finished the command and reached the command speed.
- When programing these special M, users need to note the time response of these flags. After **[Trigger Movement]** is set to ON, the system will need 10ms to update the **[Target Reached]** status to ON. Therefore, users need to delay at least 10ms and then check whether the axes have reached their target command.




**[Axis Moving] M3x464 ~ M3x479**

When any of the axes is in motion, this **[Axis Moving]** status will be ON.

6

**[NC / MLC Axis Switching] M2x432 ~ M2x447**

The NC5 controller allows users to switch axes from NC axis to MLC axis or vice versa. Setting **[NC / MLC Axis Switching]** to ON will change the axis to MLC axis mode. Alternatively, set it to OFF to change the axis to NC axis mode.

- Users need to configure the axis to NC axis by setting **[Type]** as 1 in the **[Channel Setting]** page. (As shown in the below Relevant Parameter section)
- This function is only available when the system is in AUTO or MDI mode and is switched through the Halt M Code.
- The axis must be motionless before the switch.

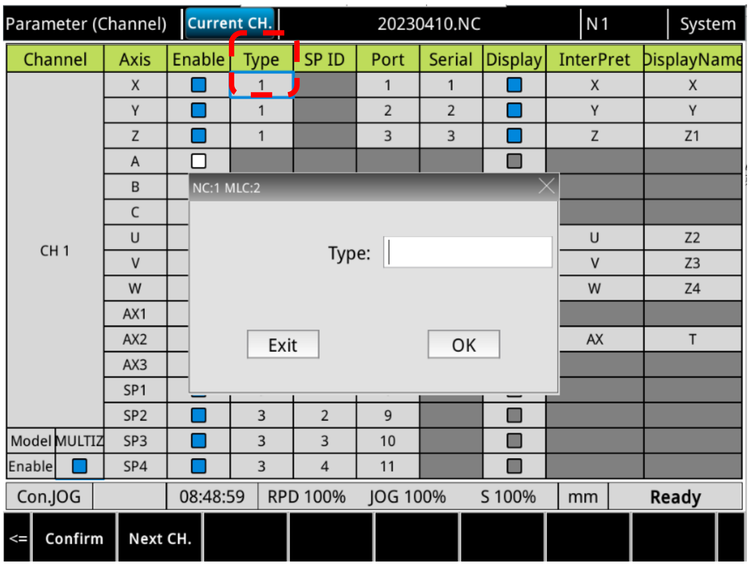
**[NC / MLC Axis Switching Finished] M3x432 ~ M3x447**

After setting the **[NC / MLC Axis Switching]** to ON to change the axis to MLC axis mode, the **[NC / MLC Axis Switching Finished]** will change to ON automatically once the switching is finished. Similarly, the **[NC / MLC Axis Switching Finished]** will change to OFF automatically if users set the **[NC / MLC Axis Switching]** to OFF.

## ■ Relevant Parameter

**Channel setting:**

When the setting in the **[Type]** is set to 1, the specific axis will be one of the NC axes and it will be allowed to switch between MLC or NC axis. When the setting in the **[Type]** is set to 2, the specific axis will be MLC axis and will not be able to switch to NC axis mode.

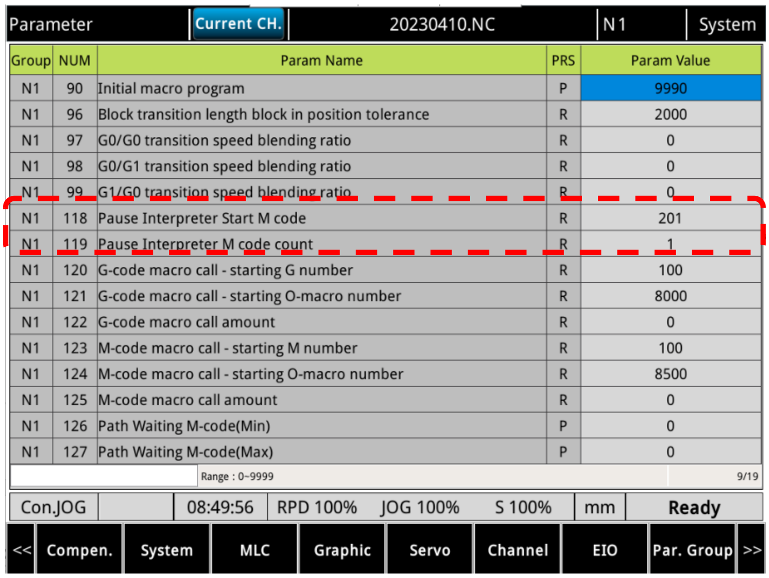



**Speed parameter setting:**

The MLC axis moving speed will refer to N2.023 (G01 Max Velocity), N2.024 (G01 Acc and Dec Time), N2.025 (G01 S Curve Time).

**Operate parameter setting:**

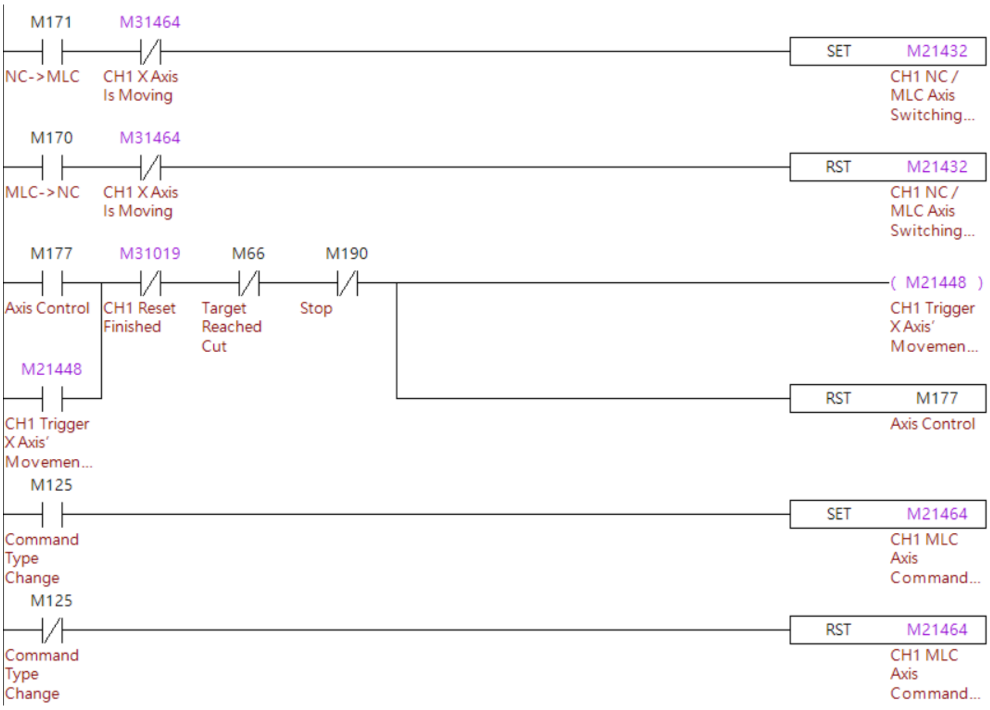
This function is only available when the system is in AUTO or MDI mode and uses the Halt M Code to switch. The Halt M Code is defined by N1.118 (Beginning M Code of Halt Function) and N1.119 (Amount of Halt M Code).



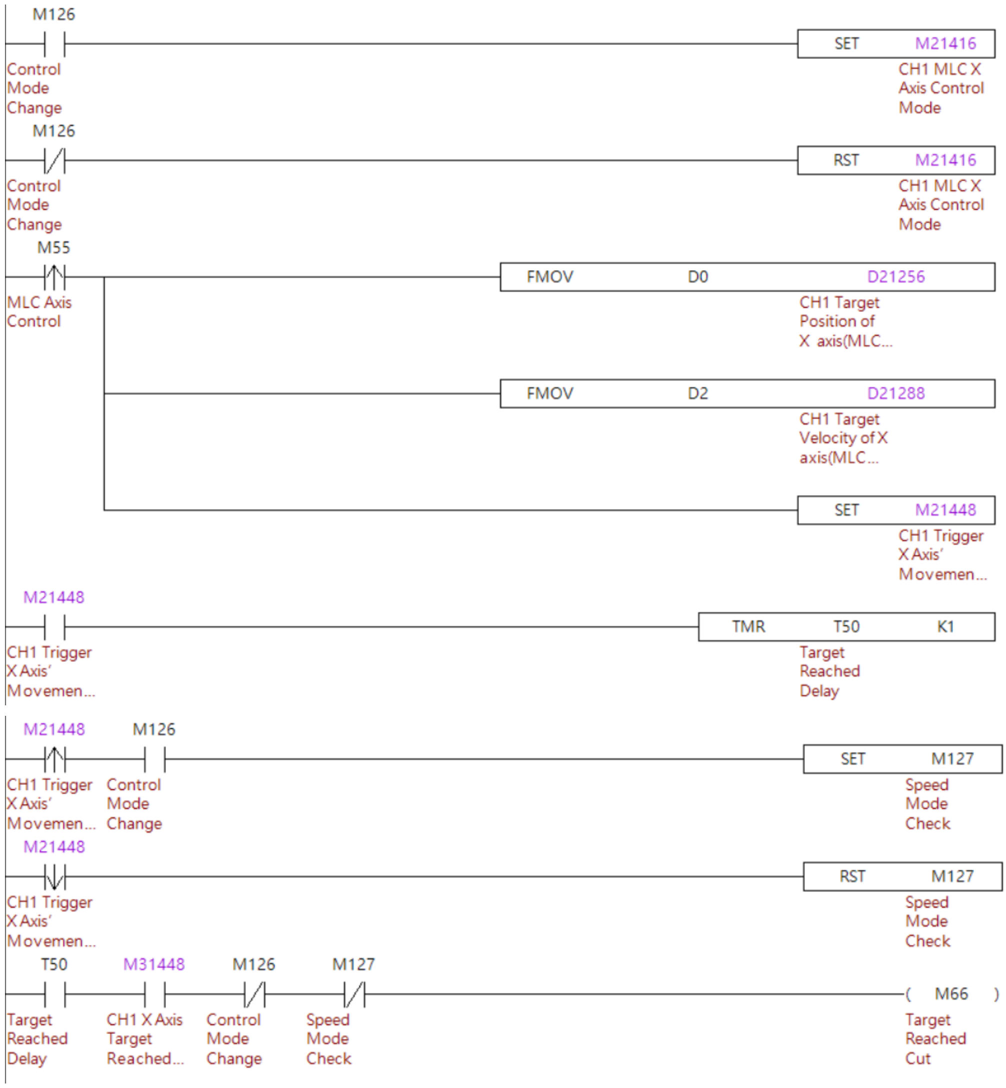

| Group | NUM | Param Name | PRS | Param Value |
|---|---|---|---|---|
| N1 | 90 | Initial macro program | P | 9990 |
| N1 | 96 | Block transition length block in position tolerance | R | 2000 |
| N1 | 97 | G0/G0 transition speed blending ratio | R | 0 |
| N1 | 98 | G0/G1 transition speed blending ratio | R | 0 |
| N1 | 99 | G1/G0 transition speed blending ratio | R | 0 |
| N1 | 118 | Pause Interpreter Start M code | R | 201 |
| N1 | 119 | Pause Interpreter M code count | R | 1 |
| N1 | 120 | G-code macro call - starting G number | R | 100 |
| N1 | 121 | G-code macro call - starting O-macro number | R | 8000 |
| N1 | 122 | G-code macro call amount | R | 0 |
| N1 | 123 | M-code macro call - starting M number | R | 100 |
| N1 | 124 | M-code macro call - starting O-macro number | R | 8500 |
| N1 | 125 | M-code macro call amount | R | 0 |
| N1 | 126 | Path Waiting M-code(Min) | P | 0 |
| N1 | 127 | Path Waiting M-code(Max) | P | 0 |



## ■ MLC Example



6




**Program execution procedure:**

Position mode:

1. M171 switches the X axis from an NC axis to MLC axis. (This action is not available when the axis is set to an MLC axis in the **channel setting** page)
2. M125 Sets the axis to absolute or incremental mode.
3. Set M126 to OFF and the axis is in position mode.
4. Set D0 for the position command value.
5. Set D2 for the speed command value.
6. Trigger M55 to update the position and speed values and activate the axis.

Speed mode:

1. M171 switches the X axis from an NC axis to MLC axis. (This action is not available when the axis is set to an MLC axis in the **channel setting** page)
2. Set M126 to ON and the axis is in speed mode.
3. Set D2 for the speed command value.
4. Trigger M55 to update the speed value and activate the axis.

**Important:**

1. The homing speed for the rotation axis refers to the settings of N2.053 (1st Searching Speed) and N2.054 (2nd Searching Speed) (unit: RPM).
2. Pay attention to the execution timing of the program. The **[M, S and T Codes Finished]** must be executed after the **[NC / MLC Axis Switching]** to ensure correct operation.
3. When a switching program is executed but the system does not perform any actions, check that the special M relays **[NC / MLC Axis Switching Finished]** are ON.
4. When a motion program is executed but the system does not perform any actions, check that the values **[Target Position]** and **[Target Velocity]** are floating-point values.
5. The corresponding halt M code parameter is required in AUTO mode when the axis is switched from MLC to NC axis mode.
6. Users need to re-trigger **[Trigger Movement]** to have the updated value take effect, except for the speed command **[Target Velocity]** which takes immediate effect after being modified.
7. The axes must stay in still when changing NC axis into MLC axis.

6

## 6.15 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 command transfer function is only available when the system is in AUTO mode. The following describes the two functions.

### ■ MLC special M relays

| Axis | Synchronous Control Enable | Enable as Slave Axis of the Synchronous Control | Command Transfer Enable | Enable as Slave Axis of the Command Transfer Control |
|---|---|---|---|---|
| X Axis | M2x256 | M2x288 | M2x257 | M2x304 |
| Y Axis | | M2x289 | | M2x305 |
| Z Axis | | M2x290 | | M2x306 |
| A Axis | | M2x291 | | M2x307 |
| B Axis | | M2x292 | | M2x308 |
| C Axis | | M2x293 | | M2x309 |
| U Axis | | M2x294 | | M2x310 |
| V Axis | | M2x295 | | M2x311 |
| W Axis | | M2x296 | | M2x312 |
| $10^{th}$ Axis | | M2x297 | | M2x313 |
| $11^{th}$ Axis | | M2x298 | | M2x314 |
| $12^{th}$ Axis | | M2x299 | | M2x315 |
| $13^{th}$ Axis | | M2x300 | | M2x316 |
| $14^{th}$ Axis | | M2x301 | | M2x317 |
| $15^{th}$ Axis | | M2x302 | | M2x318 |
| $16^{th}$ Axis | | M2x303 | | M2x319 |

**[Synchronous Control Enable] M2x256**

Set the **[Synchronous Control Enable]** to ON to enable the function of axes synchronous control. Users still need to switch ON the **[Slave Axis of the Synchronous Control]** for each slave axis.

**[Slave Axis of the Synchronous Control] M2x288 ~ M2x303**

To enable the axes synchronous control, users will need to set the **[Synchronous Control Enable]** to ON and the **[Slave Axis of the Synchronous Control]** to ON for each slave axis.

**[Command Transfer Enable] M2x257**

Set the **[Command Transfer Enable]** to ON to enable the function of axes command transfer control. Users still need to switch ON the **[Slave Axis of the Command Transfer Control]** for each slave axis.

**[Slave Axis of the Command Transfer Control] M2x304 ~ M2x319**

To enable the axes command transfer control, users will need to set the **[Command Transfer Enable]** to ON and the **[Slave Axis of the Command Transfer Control]** to ON for each slave axis.

## ■ Relevant Parameter

N1.128 (M Code to Enable Halt Function for Synchronous and Command Transfer)
N1.129 (M Code to Disable Halt Function for Synchronous and Command Transfer)

- When the controller is in AUTO or MDI mode, the synchronous and command transfer function needs to utilize the M code of halt to enable or disable the function. Controllers provide a specific halt M code for users to configure based on their preferences.
- Available range 0 to 65,535.
- When the N1.128 is set to 10, the M10 will be the halt function and specific for enable or disable synchronous or command transfer function.

N2.015 (Master Axis of Synchronous and Command Transfer):

Set for the master axis for synchronous and command transfer function.

- Value set to 1 means X axis; value set to 2 means Y axis …, etc. For example, when this parameter is set to 3, the system will use the Z axis as master axis.

N2.016 (Slave Direction of Synchronous and Command Transfer):

Set for the slave direction when the synchronous and command transfer is enabled.

- When set to 0, the slave axes will follow the same direction as the master axis.
  When set to 1, the slave axes will follow the opposite direction of the master axis.

N2.051 (Homing Action of Synchronous and Command Transfer):

Set for slave axis home action mode when the synchronous and command transfer is enabled.

- When set to 0, the slave axes will follow up the master axis during the home procedure.
  When set to 1, the slave axes and master axis are separated and proceed with the home procedure independently.

N8.009 (Slave Coordinate Setting of Synchronous and Command Transfer):

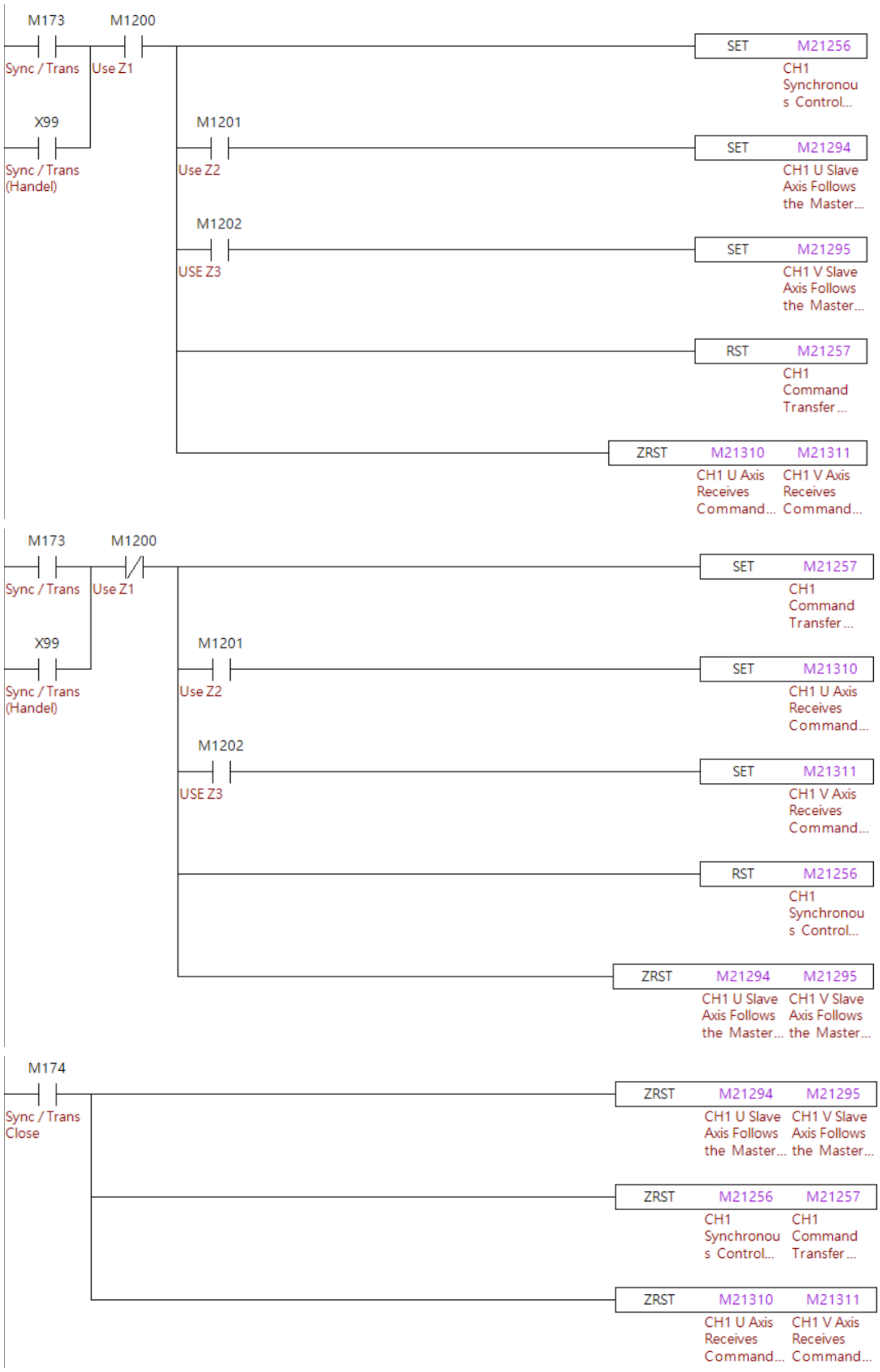
Set for whether to show slave axes’ coordinates (Bit 0) and work coordinates (Bit 2).

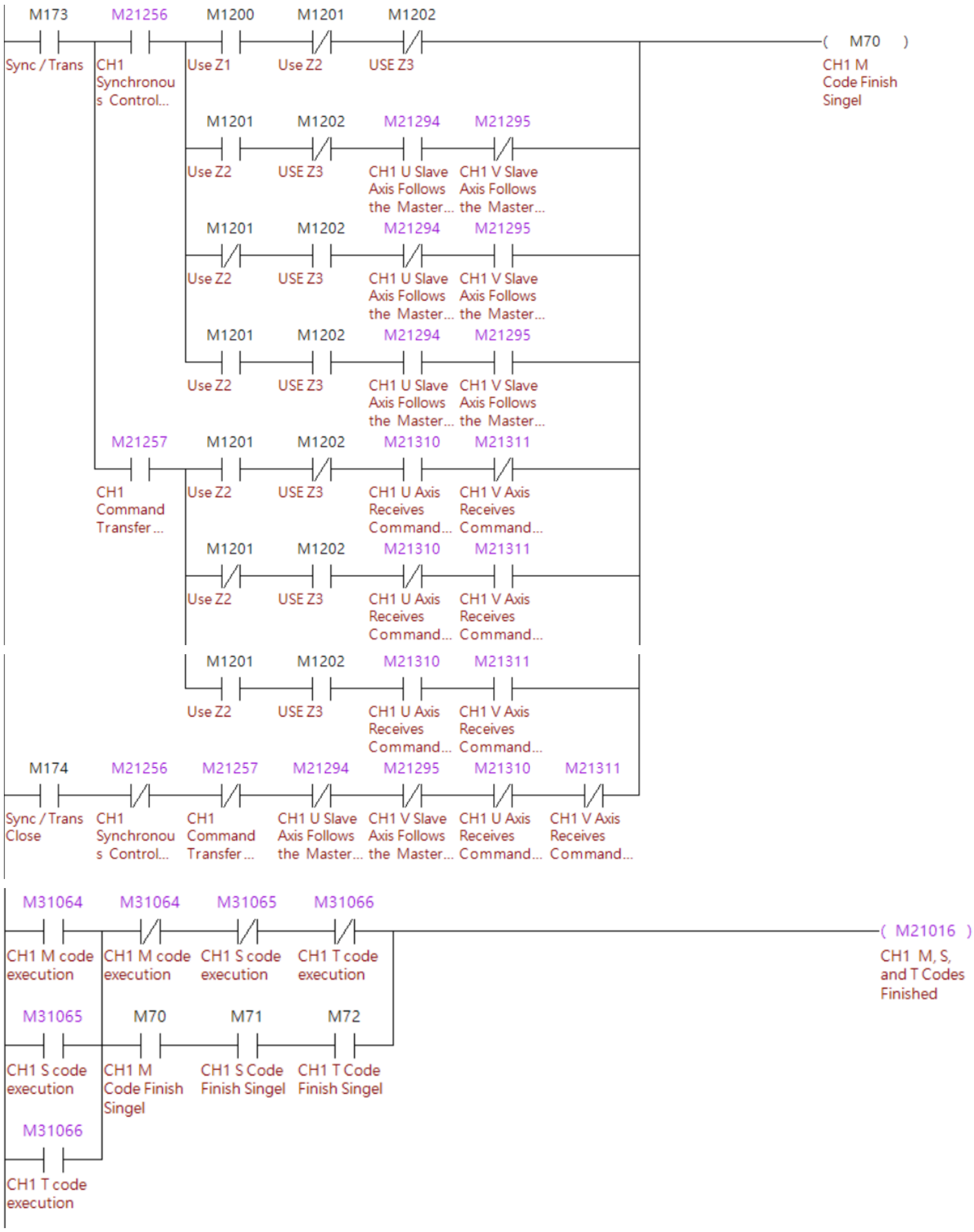
- When set to 0, not display the coordinates. When set to 1, display the coordinates.

6

## ■ MLC Example

The Z, U, and V axes are used in the following example. M1200 enables Z axis, M1201 enables U axis and M1202 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.




M173
Sync / Trans
M21256
CH1 Synchronous Control...
M1200
Use Z1
M1201
Use Z2
M1202
USE Z3
M70
CH1 M Code Finish Singel
M21294
CH1 U Slave Axis Follows the Master...
M21295
CH1 V Slave Axis Follows the Master...
M21257
CH1 Command Transfer...
M21310
CH1 U Axis Receives Command...
M21311
CH1 V Axis Receives Command...
M174
Sync / Trans Close
M31064
CH1 M code execution
M31065
CH1 S code execution
M31066
CH1 T code execution
M21016
CH1 M, S, and T Codes Finished
M70
CH1 M Code Finish Singel
M71
CH1 S Code Finish Singel
M72
CH1 T Code Finish Singel

**Program execution procedure:**

- JOG or MPG mode
  1. When X99 is ON in JOG or MPG mode, the system determines whether to execute the synchronous control or command transfer function depending on whether the Z axis is enabled.
  2. When the synchronous control function is enabled, the command transfer function must be disabled to avoid errors, and vice versa.
  3. The synchronous control or command transfer function can be disabled through RESET.
- AUTO or MDI mode
  1. When the program reads M171 in AUTO or MDI mode, the system determines whether to execute the synchronous control or command transfer function depending on whether the Z axis is enabled.
  2. When the synchronous control function is enabled, the command transfer function must be disabled to avoid errors, and vice versa.
  3. After the function is enabled, the M code procedure is complete.
  4. When the program reads M172, the synchronous control or command transfer function is disabled.

**Important:**

1. All of the M code must be defined as halt M code function in order to enable or disable the synchronous control or command transfer function in AUTO and MDI modes.
2. 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, MDI: when the M code procedure is complete.
   b. JOG, MPG: at all times.
   c. HOME
      Synchronous control: After **[Slave Axis of the Synchronous Control]** and **[Slave Axis of the Synchronous Control]** are set to ON or OFF, the system will check the corresponding axis and activate the related function once the **[Axis Homing]** is triggered.
   d. EDIT: the system does not check for the enabling of the function and operates according to the mode users switch to.
3. One axis cannot be the master axis and slave axis at the same time.
4. Multiple slave axes can follow the same master axis for synchronous control or command transfer at the same time.
5. When the synchronous control or command transfer function is enabled, if the program reads the movement command for the slave axis, the slave axis will not move, and the movement command is skipped.
6. The command transfer function supports the cutting cycle command for Z axis.
7. When N2.050 (Origin Search Mode) is 0 - 5 and the synchronous control function is enabled, if N2.051 (Origin Search Mode for Sync. Motion) is 0, the slave axes and the

master axis will perform the homing procedure synchronously.

8. The settings of the machine parameters and homing modes for the synchronous axes should be consistent.
9. When the 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 N2.001 (Rotation Axis Feed Mode) for A, B, C, U, V, and W axes must be 5. When the A, B, C, U, V, and W axes are the master axes, the setting of N2.001 for these axes must be consistent.

6

## 6.16 Synchronous gantry control

Users can use the synchronous gantry control function with M code in AUTO or 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 relays

| Axis | Synchronous Control Enable | Enable as Slave Axis of the Synchronous Control |
|---|---|---|
| X Axis | M2x256 | M2x288 |
| Y Axis | | M2x289 |
| Z Axis | | M2x290 |
| A Axis | | M2x291 |
| B Axis | | M2x292 |
| C Axis | | M2x293 |
| U Axis | | M2x294 |
| V Axis | | M2x295 |
| W Axis | | M2x296 |
| $10^{th}$ Axis | | M2x297 |
| $11^{th}$ Axis | | M2x298 |
| $12^{th}$ Axis | | M2x299 |
| $13^{th}$ Axis | | M2x300 |
| $14^{th}$ Axis | | M2x301 |
| $15^{th}$ Axis | | M2x302 |
| $16^{th}$ Axis | | M2x303 |

**[Synchronous Control Enable] M2x256**

Set the **[Synchronous Control Enable]** to ON to enable the function of axes synchronous control. Users still need to switch ON the **[Slave Axis of the Synchronous Control]** for each slave axis.

**[Slave Axis of the Synchronous Control] M2x288 ~ M2x303**

To enable the axes synchronous control, users will need to set the **[Synchronous Control Enable]** to ON and the **[Slave Axis of the Synchronous Control]** to ON for each slave axis.

## Relevant Parameter

N1.128 (M Code to Enable Halt Function for Synchronous and Command Transfer)

N1.129 (M Code to Disable Halt Function for Synchronous and Command Transfer)

- When the controller is in AUTO or MDI mode, the synchronous and command transfer function needs to utilize the M code of halt to enable or disable the function. Controllers provide a specific halt M code for users to configure based on their preferences.
- The parameter names “enable” and “disable” do not turning functions ON or OFF, but are only for the user to distinguish whether the two M codes are ON or OFF.
- Available range 0 to 65,535.
- When the N1.128 is set to 10, the M10 will be the halt function and specific for enable or disable synchronous or command transfer function.

N2.015 (Master Axis of Synchronous and Command Transfer):

Set for the master axis for synchronous and command transfer function.

- Value set to 1 means X axis; value set to 2 means Y axis …, etc. For example, when this parameter is set to 3, the system will use the Z axis as master axis.
- 

N2.016 (Slave Direction of Synchronous and Command Transfer):

Set for the slave direction when the synchronous and command transfer is enabled.

- When set to 0, the slave axes will follow the same direction as the master axis.
  When set to 1, the slave axes will follow the opposite direction of the master axis.

N2.051 (Homing Action of Synchronous and Command Transfer):

Set for slave axis home action mode when the synchronous and command transfer is enabled.

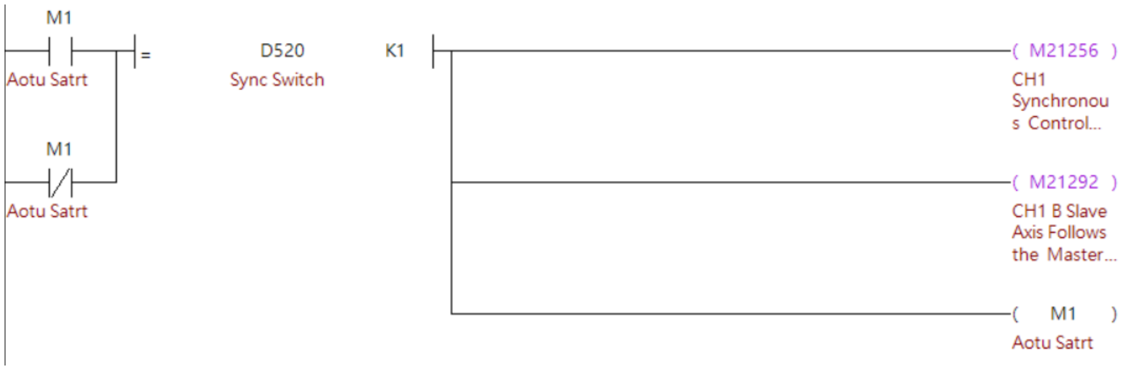
- When set to 0, the slave axes will follow up the master axis during the home procedure.
  When set to 1, the slave axes and master axis are separated and proceed with the home procedure independently.

N8.009 (Slave Coordinate Setting of Synchronous and Command Transfer):

Set for whether to show slave axes’ coordinates (Bit 0) and work coordinates (Bit 2).

- When set to 0, not display the coordinates. When set to 1, display the coordinates.

## ■ MLC Example




**Program execution procedure:**

As soon as the system is powered on, if the switch for synchronous gantry control is 0, M1 is constantly ON, and the system sets **[Synchronous Control Enable]** and **[Slave Axis of the Synchronous Control]** to ON.

**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 at the same time.
4. When the synchronous control function is enabled, if the program reads the movement command for the slave axis, the slave axis will not move, and the movement command is skipped.
5. When N2.050 (Origin Search Mode) is 0 - 5 and the synchronous control function is enabled, if N2.051 (Origin Search Mode for Sync. Motion) is 0, the slave axes and the master axis will perform the homing procedure synchronously.
6. The settings of the machine parameters and homing modes for the synchronous axes should be consistent.
7. When the 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 N2.001 (Rotation Axis Feed Mode) for A, B, C, U, V, and W axes must be 5. When the A, B, C, U, V, and W axes are the master axes, the setting of N2.001 for these axes must be consistent.

# 6.17 Devices information monitor

6

The system provides special D for users to monitor or compare the specific status of the controller through MLC. This specific status will be maintained and increased as a field request in the future.

## ■ MLC special D register

| | | | |
|---|---|---|---|
| Information Monitoring 1 Sort 1 | D3x096<br>D3x097 | Information Monitoring 2 Sort 1 | D3x104<br>D3x105 |
| Information Monitoring 1 Sort 2 | D3x098<br>D3x099 | Information Monitoring 2 Sort 2 | D3x106<br>D3x107 |
| Information Monitoring 1 Sort 3 | D3x100<br>D3x101 | Information Monitoring 2 Sort 3 | D3x108<br>D3x109 |
| Information Monitoring 1 Sort 4 | D3x102<br>D3x103 | Information Monitoring 2 Sort 4 | D3x110<br>D3x111 |

**[Information Monitoring 1 Sort 1] D3x096, D3x097**

The system will be showing the slave devices' information based on the configured parameter N1.321 (Information Monitoring Category 1) and N1.322 (Information Monitoring Sort 1).

- This is a 32-bit special D data, which will occupy two sequence D addresses.

**[Information Monitoring 1 Sort 2] D3x098, D3x099**

The system will be showing the slave devices' information based on the configured parameter N1.321 (Information Monitoring Category 1) and N1.323 (Information Monitoring Sort 1).

- This is a 32-bit special D data, which will occupy two sequence D addresses.

**[Information Monitoring 1 Sort 3] D3x100, D3x101**

The system will be showing the slave devices' information based on the configured parameter N1.321 (Information Monitoring Category 1) and N1.324 (Information Monitoring Sort 1).

- This is a 32-bit special D data, which will occupy two sequence D addresses.

**[Information Monitoring 1 Sort 4] D3x102, D3x103**

The system will be showing the slave devices' information based on the configured parameter N1.321 (Information Monitoring Category 1) and N1.325 (Information Monitoring Sort 1).

- This is a 32-bit special D data, which will occupy two sequence D addresses.

**[Information Monitoring 2 Sort 1] D3x104, D3x105**

The system will be showing the slave devices' information based on the configured parameter N1.326 (Information Monitoring Category 2) and N1.327 (Information Monitoring Sort 1).

- This is a 32-bit special D data, which will occupy two sequence D addresses.

**[Information Monitoring 2 Sort 2] D3x106, D3x107**

The system will be showing the slave devices' information based on the configured parameter N1.326 (Information Monitoring Category 2) and N1.328 (Information Monitoring Sort 1).

- This is a 32-bit special D data, which will occupy two sequence D addresses.

**[Information Monitoring 2 Sort 3] D3x108, D3x109**

The system will be showing the slave devices' information based on the configured parameter N1.326 (Information Monitoring Category 2) and N1.329 (Information Monitoring Sort 1).

- This is a 32-bit special D data, which will occupy two sequence D addresses.

**[Information Monitoring 2 Sort 4] D3x110, D3x111**

The system will be showing the slave devices' information based on the configured parameter N1.326 (Information Monitoring Category 2) and N1.330 (Information Monitoring Sort 1).

- This is a 32-bit special D data, which will occupy two sequence D addresses.

■ **Relevant Parameter**

N1.321 (Information Monitoring Category 1):

Users can set this to determine the data type of 1st category to read.

- The input range is 0 – 65535.
- When set to 1, the system will allocate the 1st category as the axes' machine coordinates.

N1.322 (Information Monitoring Sort 1 of Category 1):

The special D can show specific monitored information in accordance with the setting of N1.321 (Information Monitoring Category 1).

- The input range is 0 – 65535.
- When N1.321 is set to 1 and this parameter is set to 1, the system will update the X axis's machine coordinate to the corresponding special D. If this parameter is set to 2, the system will update the Y axis's information to the corresponding special D.

N1.323 (Information Monitoring Sort 2 of Category 1):

The special D can show specific monitored information in accordance with the setting of N1.321 (Information Monitoring Category 1).

- The input range is 0 – 65535.
- When N1.321 is set to 1 and this parameter is set to 1, the system will update the X axis's machine coordinates to the corresponding special D. If this parameter is set to 2, the system will update the Y axis's information to the corresponding special D.

N1.324 (Information Monitoring Sort 3 of Category 1):

The special D can show specific monitored information in accordance with the setting of N1.321 (Information Monitoring Category 1).

- The input range is 0 – 65535.
- When N1.321 is set to 1 and this parameter is set to 1, the system will update the X axis's machine coordinates to the corresponding special D. If this parameter is set to 2, the system will update the Y axis's information to the corresponding special D.

N1.325 (Information Monitoring Sort 4 of Category 1):

The special D can show specific monitored information in accordance with the setting of N1.321 (Information Monitoring Category 1).

- The input range is 0 – 65535.
- When N1.321 is set to 1 and this parameter is set to 1, the system will update the X axis's machine coordinates to the corresponding special D. If this parameter is set to 2, the system will update the Y axis's information to the corresponding special D.

N1.326 (Information Monitoring Category 2):

Users can set this to determine the data type of 2nd category to read.

- The input range is 0 – 65535.
- When set to 1, the system will allocate the 2nd category as the axes' machine coordinate.

N1.327 (Information Monitoring Sort 1 of Category 2):

The special D can show specific monitored information in accordance with the setting of N1.321 (Information Monitoring Category 2).

- The input range is 0 – 65535.
- When N1.326 is set to 1 and this parameter is set to 1, the system will update the X axis's machine coordinates to the corresponding special D. If this parameter is set to 2, the system will update the Y axis's information to the corresponding special D.

N1.328 (Information Monitoring Sort 2 of Category 2):

The special D can show specific monitored information in accordance with the setting of N1.321 (Information Monitoring Category 2).

- The input range is 0 – 65535.
- When N1.326 is set to 1 and this parameter is set to 1, the system will update the X axis's machine coordinates to the corresponding special D. If this parameter is set to 2, the system will update the Y axis's information to the corresponding special D.

N1.329 (Information Monitoring Sort 3 of Category 2):

The special D can show specific monitored information in accordance with the setting of N1.321 (Information Monitoring Category 2).

- The input range is 0 – 65535.
- When N1.326 is set to 1 and this parameter is set to 1, the system will update the X axis's machine coordinates to the corresponding special D. If this parameter is set to 2, the system will update the Y axis's information to the corresponding special D.

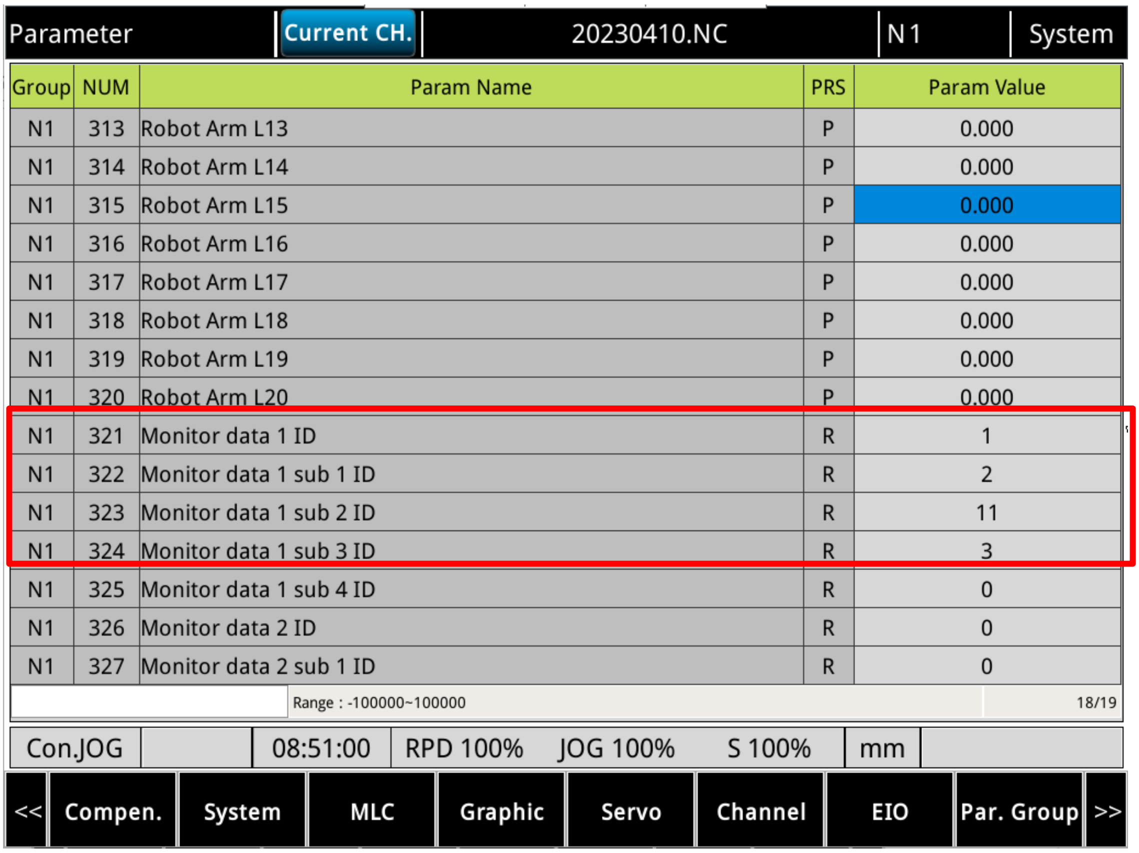
N1.330 (Information Monitoring Sort 4 of Category 2):

The special D can show specific monitored information in accordance with the setting of N1.321 (Information Monitoring Category 2).

- The input range is 0 – 65535.
- When N1.326 is set to 1 and this parameter is set to 1, the system will update the X axis's machine coordinates to the corresponding special D. If this parameter is set to 2, the system will update the Y axis's information to the corresponding special D.

## ■ MLC Example

When the parameter N1.321 (Information Monitoring Category) is set to 1 and N1.322 to N1.324 (Information Monitoring Sort 1 to 3) are set to 1, 2 and 3 respectively, users can read the machine coordinates of X, Y and Z through the special D in the MLC.



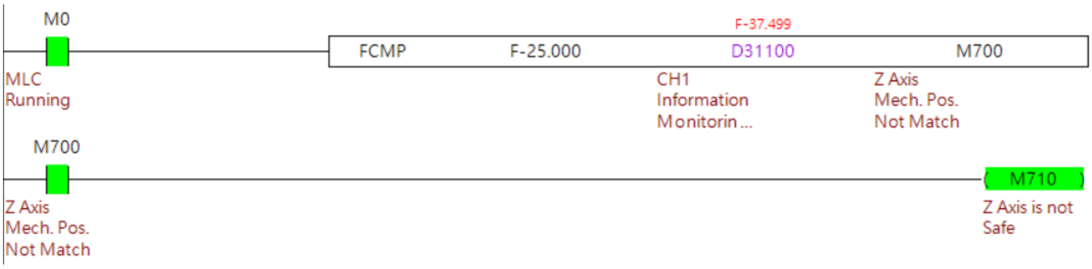
Parameter | Current CH. | 20230410.NC | N1 | System

| Group | NUM | Param Name | PRS | Param Value |
|---|---|---|---|---|
| N1 | 313 | Robot Arm L13 | P | 0.000 |
| N1 | 314 | Robot Arm L14 | P | 0.000 |
| N1 | 315 | Robot Arm L15 | P | 0.000 |
| N1 | 316 | Robot Arm L16 | P | 0.000 |
| N1 | 317 | Robot Arm L17 | P | 0.000 |
| N1 | 318 | Robot Arm L18 | P | 0.000 |
| N1 | 319 | Robot Arm L19 | P | 0.000 |
| N1 | 320 | Robot Arm L20 | P | 0.000 |
| N1 | 321 | Monitor data 1 ID | R | 1 |
| N1 | 322 | Monitor data 1 sub 1 ID | R | 2 |
| N1 | 323 | Monitor data 1 sub 2 ID | R | 11 |
| N1 | 324 | Monitor data 1 sub 3 ID | R | 3 |
| N1 | 325 | Monitor data 1 sub 4 ID | R | 0 |
| N1 | 326 | Monitor data 2 ID | R | 0 |
| N1 | 327 | Monitor data 2 sub 1 ID | R | 0 |

Range : -100000~100000 18/19

Con.JOG | 08:51:00 | RPD 100% JOG 100% S 100% | mm

<< Compen. System MLC Graphic Servo Channel EIO Par. Group >>




**Program execution procedure:**

In this MLC example, FCMP instructions are used to compare the Z axis's machine position. In addition, the M700 will be triggered due to the Z axis's actual position being smaller than the setting position.

# 6.18 Robot arm control



The NC5 controller provides multi-axes and multi-channel functions. On the different channels, users can define each channel with different mechanisms, such as a Cartesian coordinate system or robot coordinate system. Furthermore, users can use the machine for regular processes and utilize the robot arm for load and unload procedures.

## ■ MLC special D

| | | | |
|---|---|---|---|
| Servo On / Off | M2x012 | MPG Ratio Selection | D2x009 |
| Robot Coordinate System | M2x048<br>M2x049<br>M2x050<br>M2x051 | Robot's Coordinate System Switch in Manual Mode | D2x016 |
| NC Mode Switch | D2x000 | Robot's Tool System Switch in Manual Mode | D2x017 |
| Feed Rate Override for JOG, INC and Dry Run Mode | D2x006 | Current Coordinate System | D3x016 |
| MPG Axes Selection | D2x008 | Current Robot Tool Coordinate System | D3x017 |

**[Servo ON / OFF] M2x012**

Servo ON or OFF for all connected servo drives in the corresponding channel.

- The parameter **[N1.010 Axes Manually Enable after Startup]** must be set to 1.

**[Robot Coordinate System Switching] M2x048 ~ M2x051**

Robot coordinate system can be switched as shown in the table below. CNC controllers need to be in manual mode.

| Robot Mode Control | M2x048 | M2x049 | M2x050 | M2x051 | D2x016 |
|---|---|---|---|---|---|
| PCS (Piece) | 1 | 1 | 1 | 1 | - |
| TCS (Tool) | 1 | 1 | 0 | 1 | - |
| JCS (Joint) | 0 | 0 | 0 | 0 | - |

**[NC Mode Switch] D2x000**

The NC system uses this D2x000 to switch between different NC modes. The x in D2x000 represents different NC channels.

| NC Mode Switch | Value | Mode | Mode Status |
|---|---|---|---|
| D2x000 | 0 | AUTO | M3x000 |
| | 1 | EDIT | M3x001 |
| | 2 | MDI | M3x002 |
| | 3 | MPG | M3x003 |
| | 4 | JOG | M3x004 |
| | 5 | RAPID | M3x005 |
| | 6 | INC | M3x006 |
| | 7 | HOME | M3x007 |

**[Dry Run, Feed Rate] M2x005, D2x006**

When the controller is in AUTO or MDI mode, if **[Dry Run]** is triggered, the system will load the main program or MDI program and then start the execution with the **[Feed Rate]** speed.

- When **[Dry Run]** is triggered as ON, the system will take **[Feed Rate]** as the process speed.
- When the system is executing in dry run mode and the **[Dry Run]** is reset to OFF, the system will continue running but at the NC programed speed.

**[MPG Axes Selection] D2x008**

Users can switch the MPG axis by setting this special D to the indicated axes as below.
When the channel is configured as a robot system, the system will determine whether the robot is in the PCS (Piece Coordinate) or TCS (Tool Coordinate) mode, and it will switch MPG axis control accordingly as
0 = X Axis, 1 = Y Axis, 2 = Z Axis, 3 = A Axis, 4 = B Axis or 5 = C Axis.
When the robot system is in JCS (Joint Coordinate) mode, the system will switch the MPG axis control accordingly as
0 = J1 Axis, 1 = J2 Axis, 2 = J3 Axis, 3 = J4 Axis, 4 = J5 Axis or 5 = J6 Axis.

**[MPG Ratio Selection] D2x009**

This **[MPG Ratio Selection]** can be set to 1, 10 or 100. When users rotate one unit from the wheel, the system will obtain 1 pulse from the MPG device. The system will take the parameter **N9.013 (Unit Decimal Point)** as the command unit for each pulse. In addition, the system will multiply the **[MPG Ratio Selection]** as the ratio for the final command sent to a specific axis.
For example: **[MPG Ratio Selection]** is 10 and the N9.013 (Unit Decimal Point) set to 0.001mm. The minimum movement of the MPG control will be 0.001 × 10 = 0.01 mm.

**[Robot's Coordinate System Switch in Manual Mode] D2x016**

When the robot system is in manual mode, users can use this special D to switch between different work coordinate systems.
1 ~ 6: corresponding to G54 to G59.

**[Robot's Tool System Switch in Manual Mode] D2x017**

When the robot system is in manual mode, users can use this special D to switch between different tool coordinate systems.
0: Not using tool offset.
1 ~ n: corresponding to the tool number.

**[Current Coordinate System] D3x016**

This shows the current applied working coordinate system.

- When the **[model]** in the **[Channel Setting]** page is not set to Robot, this special D will show the current coordinate system.
  1 ~ 6: corresponding to G54 to G59.
  7: corresponding to G54 P1 to G54 P256
- When the **[model]** in the **[Channel Setting]** page is set to Robot, this special D will show the current robot coordinate system.
  1 ~ 6: corresponding to G54 to G59.

**[Current Coordinate System] D3x017**

This shows the current applied tool coordinate system.

0: Not using tool offset

1 ~ n: corresponding to the tool number.

## ■ Relevant Parameter

**N1.301~N1.320 (Robot Arm Length 1~20):**

The NC5 controller provides a Robot arm control mechanism, users will need to provide corresponding hardware mechanical arm length parameters (MDH definition) for the system's algorithm.

- Unit: mm, available range: -100000 to +100000

**N1.064 MPG Path Acc and Dec Time**

The system will use this parameter as the path filter to smooth out the machining speed between each command. The smoother the machining speed, the longer the total process time. Larger value settings could cause the command deceleration to become too long, which can cause the machine to continue moving for a short distance after MPG is stopped.

**N1.065 MPG Path S Curve Time**

The system will use this parameter as the path filter to smooth out the machining speed between each command. Larger value settings could cause a loss in precision.

**N.915 Enable the Robot Teaching Programming**

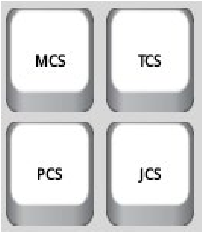
Users can easily teach and program the robot control program directly on the controller.

After moving and confirming the robot to the teaching point using MPG or JOG control, users can click on the linear move (G01.1) or point-to-point (G00.1) buttons on the lower left corner. The system will insert the relative command to the user’s NC program. The teaching position is based on the Robot’s current coordinate mode, which is PCS, TCS or JCS. In the PCS and TCS modes, the point command is based on XYZABC coordinates. Furthermore, the XYZABC coordinates will describe the values of the machine coordinates when it is in JCS mode.

6

## ■ Robot mode switching example

User can manage the system mode of the robot arm directly on the controller HMI.




In this example, the robot mode switch is demonstrated using the mandatory button, utilizing the corresponding *SYSVRM to ON/OFF button source with after call macro to complete the mode switch function. The following description illustrates the mode switching and macro content with *SYSVRW.

- PCS (Piece Coordinate System): *SYSVRW_0_40.2

  Macro content: Setting [**Robot Coordinate System manual switch**] to 0 to disable manual mode, and then set **[Robot Coordinate System Switching]** to PCS (Piece Coordinate) and then reset other button sources to OFF.

```
$D_0_21016 = 0
BITON M_0_21048
BITON M_0_21049
BITON M_0_21050
BITON M_0_21051
BITOFF *SYSVRW_0_40.4
BITOFF *SYSVRW_0_40.3
BITOFF *SYSVRW_0_40.1
```

- TCS (Tool Coordinate System): *SYSVRW_0_40.3

  Macro content: Setting **[Robot Coordinate System Switching]** to TCS (Tool Coordinate) and then reset other button sources to OFF.

```
BITON M_0_21048
BITON M_0_21049
BITOFF M_0_21050
BITON M_0_21051
BITOFF *SYSVRW_0_40.1
BITOFF *SYSVRW_0_40.2
BITOFF *SYSVRW_0_40.4
```

- JCS (Joint Coordinate): *SYSVRW_0_40.4

  Macro content: Setting **[Robot Coordinate System Switching]** to JCS (Joint Coordinate) and then reset other button sources to OFF.

```
BITOFF M_0_21048
BITOFF M_0_21049
BITOFF M_0_21050
BITOFF M_0_21051
BITOFF *SYSVRW_0_40.2
BITOFF *SYSVRW_0_40.3
BITOFF *SYSVRW_0_40.1
```

## ■ Instruction description

When the system is in AUTO or MDI mode, the robot is controlled by G code instructions, as shown below.

G00.1 **MovP** point-to-point moving instruction.

| | |
|---|---|
| Format | G00.1 X_Y_Z_A_B_C_ P0000 H_R_Q_F_ |
| Example | G00.1 X400. Y0. Z300. A180. B0. C0. P0000 H0 R1 Q1 F20 |

G01.1 **MovL** linear moving instruction.

| | |
|---|---|
| Format | G01.1 X_Y_Z_A_B_C_ P0000 H_R_Q_F_ |
| Example | G01.1 X400. Y0. Z300. A180. B0. C0. P0000 H0 R1 Q1 F20 |

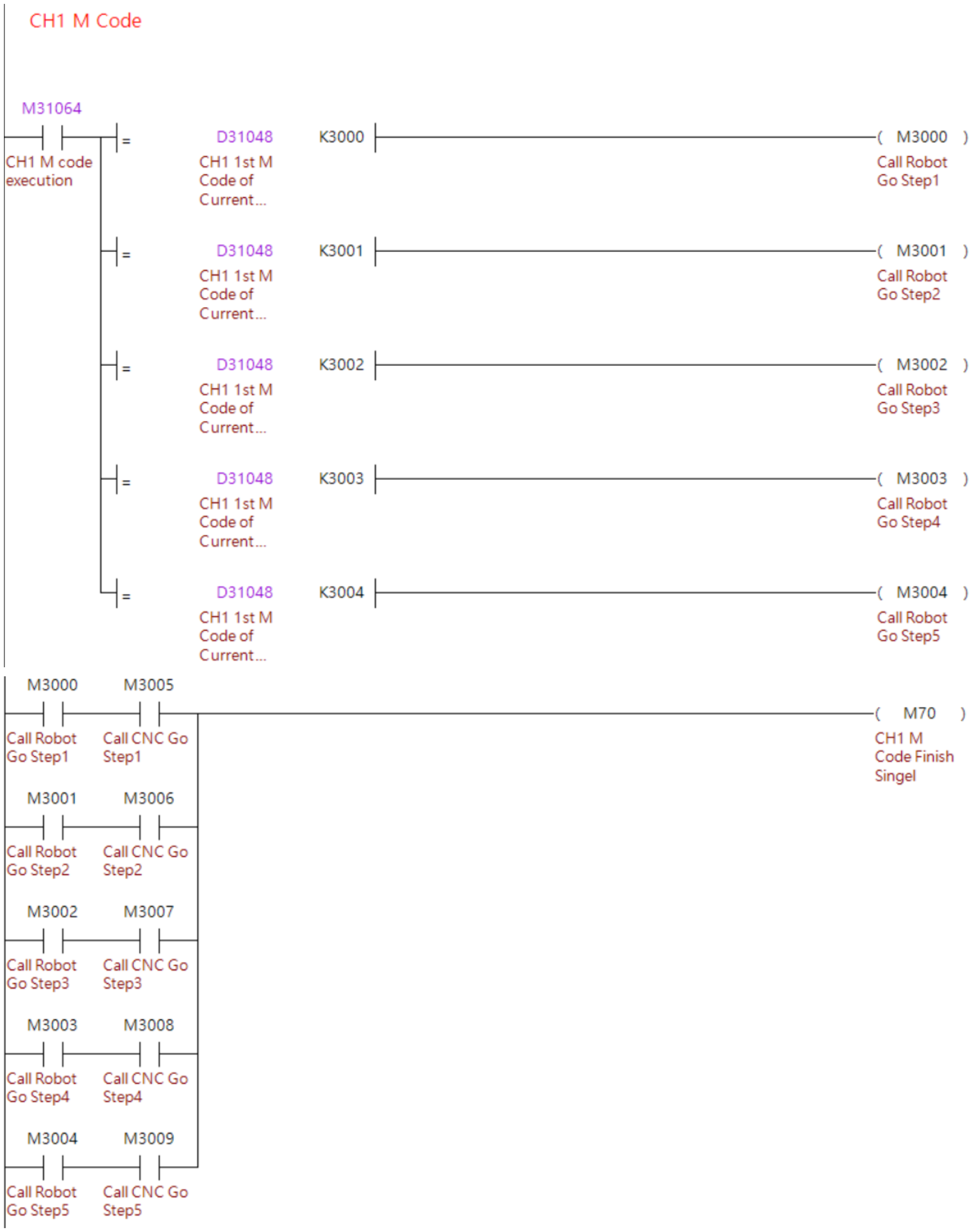
- Instruction Description

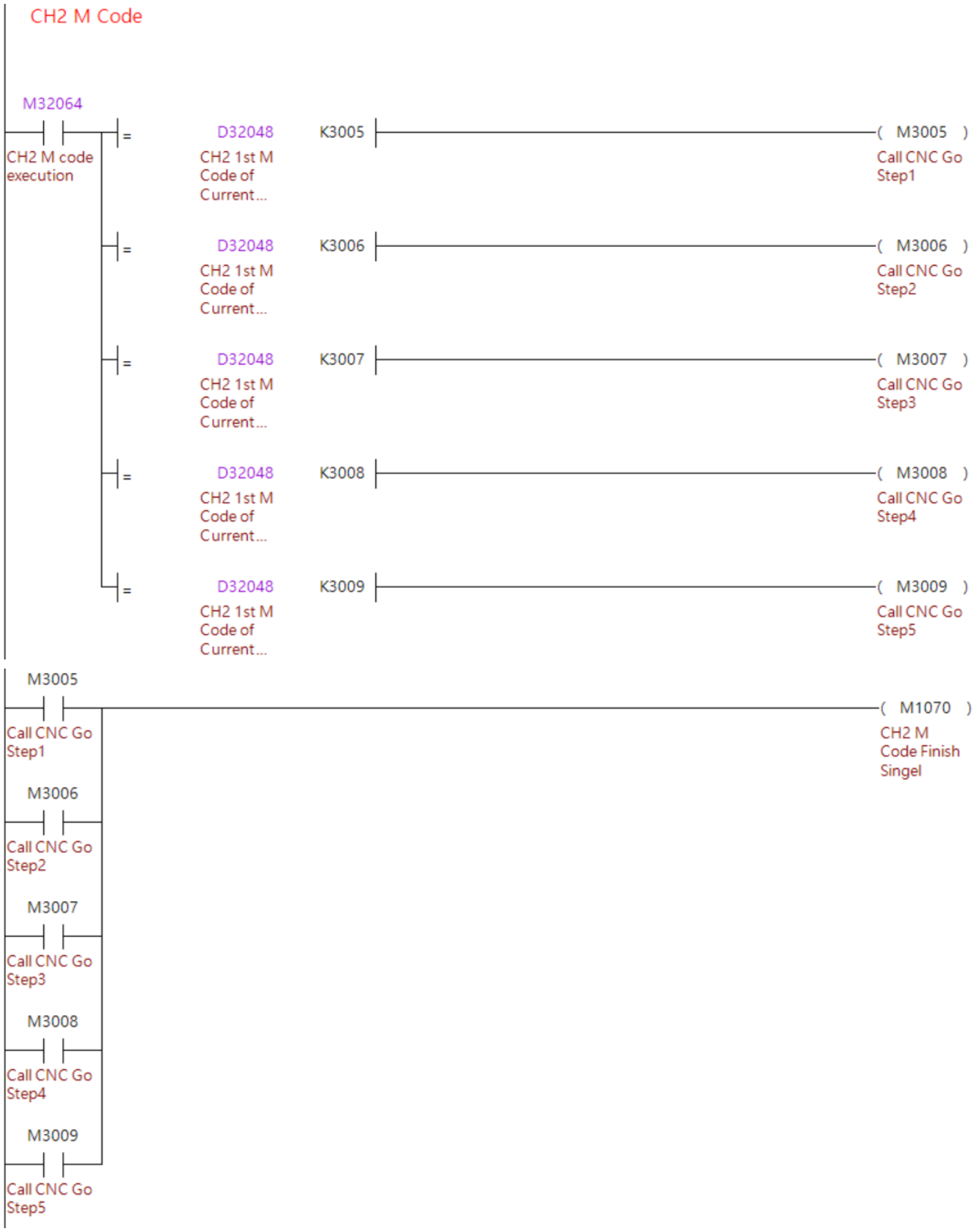
<table>
<tr><th>Instruction</th><th>Description</th></tr>
<tr><td>XYZABC</td><td>Refers to Q coordinate settings, if the coordinate system is PCS or TCS (Q = 0 to 2), the XYZABC represent space coordinates for each direction. If the coordinate system is JCS (Q = 3), the XYZABC represent joint coordinates for J1 to J6.</td></tr>
<tr><td>P0000</td><td>Posture setting of Robot arm control can be defined as below, in bits.
<table>
<tr><td>Bit</td><td>0</td><td>1</td><td>2</td><td>3</td></tr>
<tr><td>Robot posture</td><td>PS*</td><td>Shoulder</td><td>Elbow</td><td>Flip</td></tr>
</table>
Note: represents applying the current posture and ignores any other setting.</td></tr>
<tr><td>H</td><td>Tool coordinate system<br>0: Not using tool offset<br>1 ~ n: corresponding to the tool number</td></tr>
<tr><td>R</td><td>Working coordinate<br>1 ~ 6: corresponding to G54 to G59.</td></tr>
<tr><td>Q</td><td>Coordinate system setting<br>1: PCS (Piece coordinate)<br>2: TCS (Tool coordinate)<br>3: JCS (Joint coordinate)</td></tr>
<tr><td>F</td><td>Speed percentage (%)</td></tr>
</table>

6

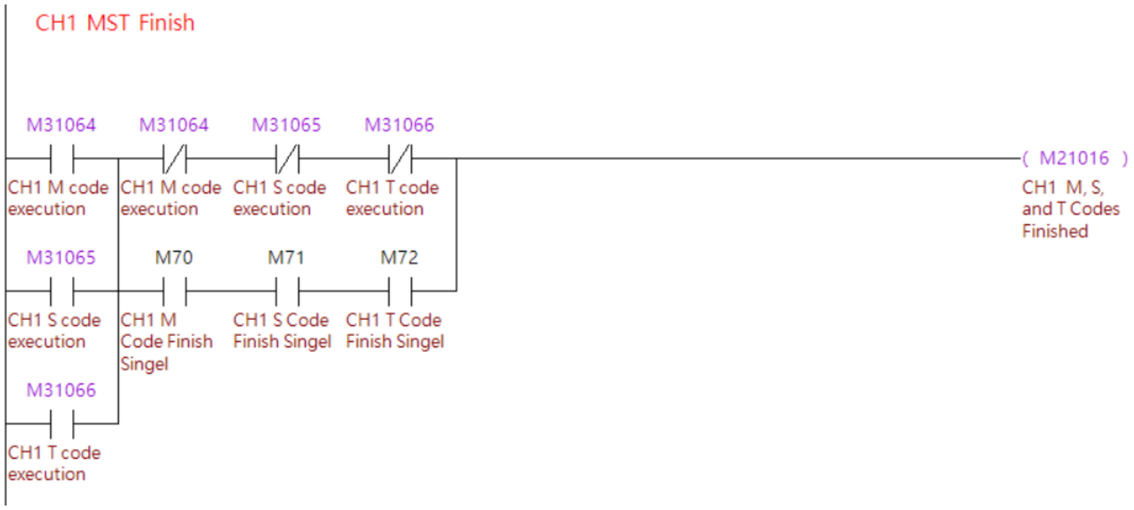
## ■ MLC Example

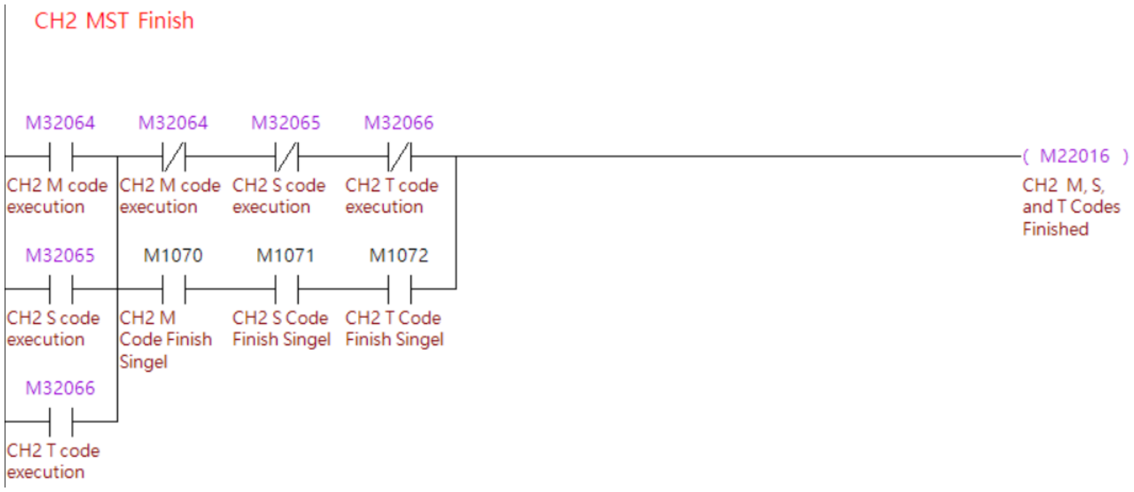
The following example provides a two channels application as multi-channel. Therefore, the same MLC can access two different CH1 and CH2 and exchange data or whether to execute or to hold execution based on the other channel's status.




CH2 M Code
M32064
CH2 M code execution
D32048
CH2 1st M Code of Current...
K3005
M3005
Call CNC Go Step1
K3006
M3006
Call CNC Go Step2
K3007
M3007
Call CNC Go Step3
K3008
M3008
Call CNC Go Step4
K3009
M3009
Call CNC Go Step5
M1070
CH2 M Code Finish Singel

6




**Program execution procedure:**

The 1st channel (CH1) is defined as a milling machine and the 2nd channel (CH2) is defined as a robot arm. Users can use M / S / T code to exchange status or data and hold procedures based on the status of each other. For more information on the M / S / T functions, please refer to section 6.8.

The above example demonstrates that when the CH1 milling machine finishes the process it will activate M3000 to acknowledge the CH2 to start the robot action. Furthermore, after CH2 finishes the procedure with the command as G00.1, G01.1 or M code, it will acknowledge CH1 to continue work by M3005.

**Important:**

In order to enable the teaching function, users need to set up the below items.

1. Set N.915 (Enable the Robot Teaching Programming) to 1.
2. Set up two function keys as function number 9226 (Linear move) and function number 9227 (Point-to-point move).

# 6.19 Axes Oscillation Control

The NC5 controller provides axes oscillation control, which is special design for grinding solution. Controller system will add a SIN wave position directly on the configured axes. In the grinding field, this oscillation can greatly improve performance and the grinding wheel can efficiently uniform the usage.

## ■ MLC Special M & Special D

| | | | |
|---|---|---|---|
| Axes Oscillation Control Enable | M2x264 | - | - |
| Axes Oscillation Amplitude | D2x018<br>D2x019 | Axes Oscillation Cycle Time | D2x020<br>D2x021 |
| Oscillation Axes Enable | D2x022 | Axes Oscillation Wave Type | D2x023 |

**[Axes Oscillation Control Enable]: M2x264**

Set this **[Axes Oscillation Control Enable]** to ON to enable the axes oscillation function, and the system will execute axis oscillation commands based on the settings in **[Axes Oscillation Amplitude]** and **[Axes Oscillation Cycle Time]**. The **[Oscillation Axes Enable]** defines the axes number to be performed the oscillation function.

- If the axis is already performing path interpolation, the oscillation command will be added to the original position command.
- The system will enable the function when **[Axes Oscillation Control Enable]** is ON instead of raising edge. During the function is enabled, the system will continuously add oscillation command based on **[Axes Oscillation Amplitude]** and **[Axes Oscillation Cycle Time]** and return an alarm code **[0602 Over speed acceleration]** once axes speed acceleration is too fast.
- When the **[Axes Oscillation Control Enable]** is ON and function is enabled, any alarm occur will stop the oscillation function immediately. If users want to enable the function again after the system reset, they will need to set **[Axes Oscillation Control Enable]** to OFF and then turn it ON again.
- When the **[Axes Oscillation Wave Type]** set as mode 0, the system will stop adding oscillation command after axes pass the beginning position by set the **[Axes Oscillation Control Enable]** from ON to OFF to turn off the function.
- When the **[Axes Oscillation Wave Type]** set as mode 1, the system will stop adding oscillation command after axes at the beginning position by set the **[Axes Oscillation Control Enable]** from ON to OFF to turn off the function.
- When the **[Axes Oscillation Control Enable]** is ON, the current position of each axis is showing the added new position.
- Only available on modes other than HOME and EDIT.

**[Axes Oscillation Amplitude]: D2x018 ~ D2x019**

When the axes oscillation control **[Axes Oscillation Control Enable]** is enabled, the activated axes will oscillate based on the amplitude here.

- Value range -2,147,483,648 ~ +2,147,483,647
- Unit: mm
- When the **[Axes Oscillation Wave Type]** set as mode 0 and **[Axes Oscillation Amplitude]** is positive, the axes will oscillate forward first and then backward continuously. On the other hands, it will oscillate backward first and then forward continuously when the value is negative.
- When the **[Axes Oscillation Wave Type]** set as mode 1 and **[Axes Oscillation Amplitude]** is positive, the axes will oscillate forward first to the amplitude value and then backward to the beginning position continuously. On the other hands, it will oscillate backward first and then forward continuously when the value is negative.

**[Axes Oscillation Cycle Time]: D2x020 ~ D2x021**

The system will take this **[Axes Oscillation Cycle Time]** as the cycle time for each oscillation command.

- Unit: ms
- The cycle time is defined as the axes pass through one cycle distance from original position to the amplitude movement and then back to the original position.

**[Oscillation Axes Enable]: D2x022**

The oscillation axes use this 16-bit **[Oscillation Axes Enable]** as the mask to determine whether to enable the function on each specific axis.

For example, if this set to 5, it means the X and Z axes will both have their oscillation function activated.

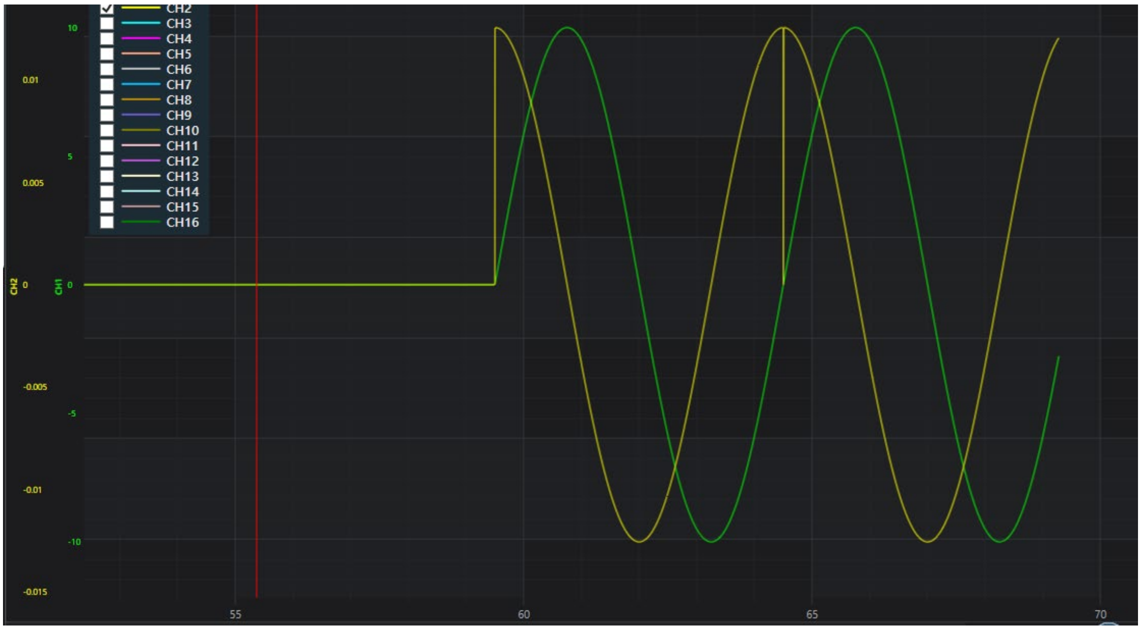
- Value range 0 ~ 65535.
- For example, if this set to 5, it means the X and Z axes will both have their oscillation function activated.

6

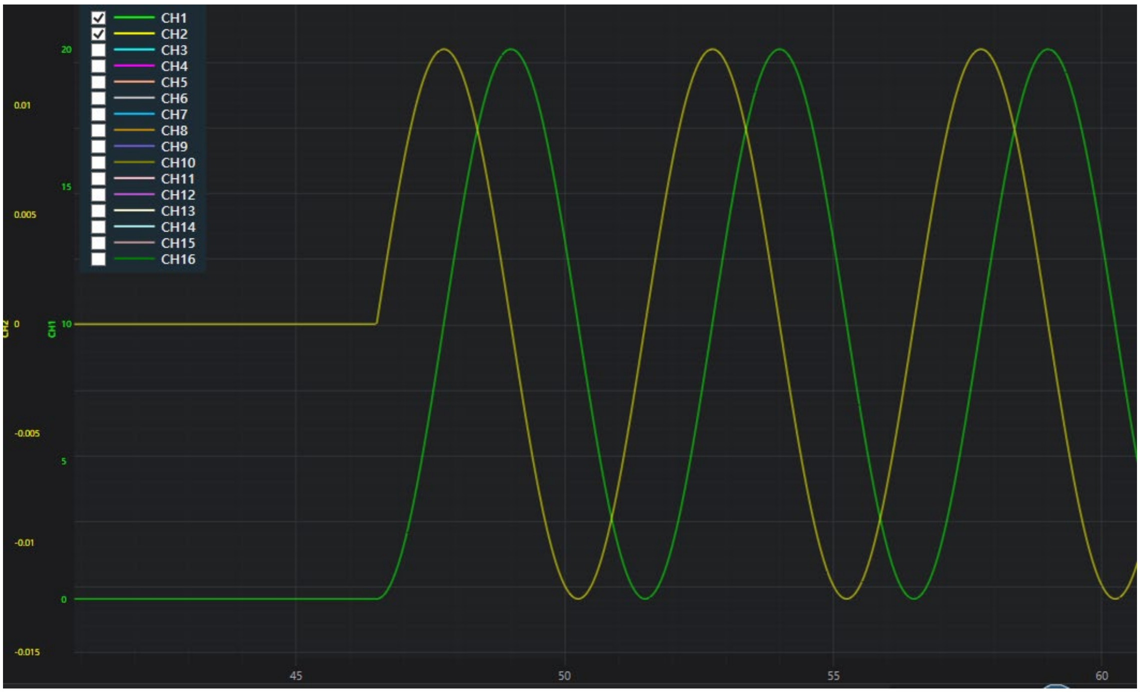
**[Axes Oscillation Wave Type]: D2x023**

Setting for axes oscillation wave type.

- Value range 0 ~ 65535.
- When set as 0, the system will apply [Asin(w)] as overlay oscillation wave. In this [Asin(W)] formula, the added command will take 0 as median incremental position and then add positive and negative additional movement continuously as shown below in green line.



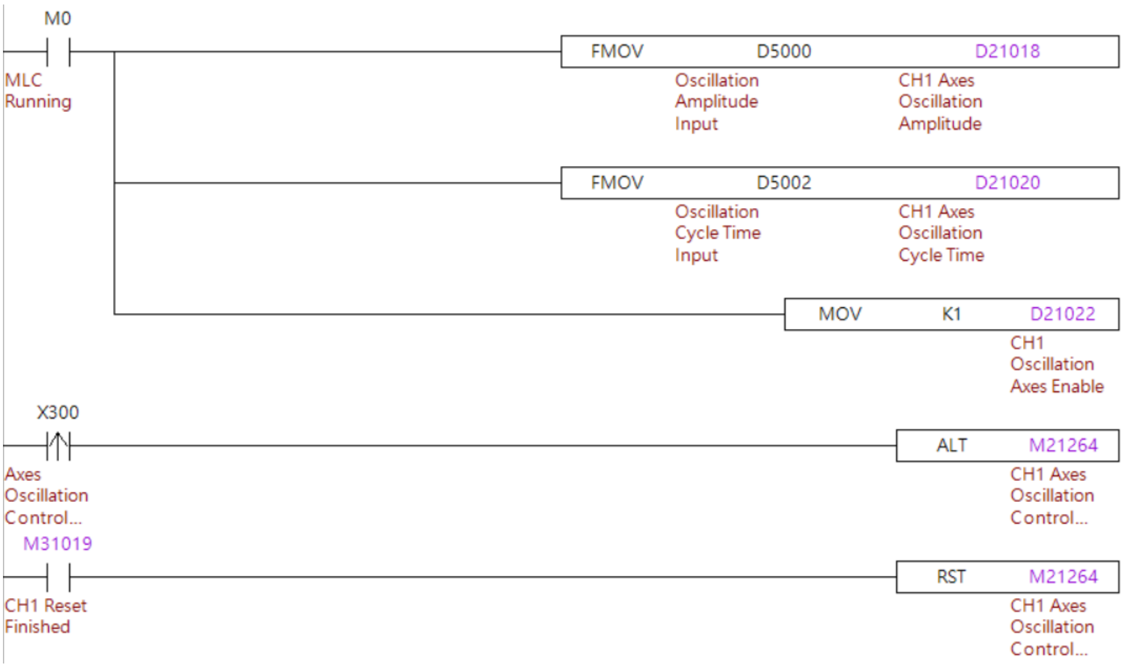

- When set as 0, the system will apply [A-cos(w) + A] as overlay oscillation wave. In this [A-cos(w) + A] formula, the added command will take 0 as based incremental position and then add positive or negative as one direction additional movement continuously as shown below in green line.



6

## ■ MLC example

The following example provides oscillation function enable through real digital input and then indicated to the X axis.




**Program execution procedure:**

1. The example MLC will continuously move user provide amplitude and cycle time into special D register.
2. When users triggered the digital input signal (X300), the MLC will enable the oscillation function on the X axis.
3. After trigger the system Reset, the MLC will disable the oscillation function on the X axis.

**Important:**

Only available on modes other than HOME and EDIT.

# CNC Alarm Information A

This appendix provides the information about the alarms and troubleshooting methods for the NC system. Search this appendix for the methods of handling the NC system related malfunctions.

# A.1 CNC Alarm Categories

The CNC alarms can be divided into three categories, which are system alarms, user-defined alarms, and user-defined macro alarms. This appendix only describes the **[system alarms]** while the rest are user-defined.

| Alarm Category | Alarm Code | Alarm Description |
|---|---|---|
| System alarms | - | The system alarms caused by system error or operation error. Some system alarms include information, which has different meanings according to the different alarms. |
| User-defined alarms | A _ | The user-defined alarms which are programed in the MLC. When the A_ device is triggered, the alarm corresponding to the A_ device defined in **[DOPSoft - User-defined alarm]** will reported as well.<br>Range: A0~A511 |
| User-defined macro alarms | MR_ | The user-defined macro alarm works with NC program # variable. When the command as #20020=_, the alarm triggered and have the corresponding macro alarm defined in **[DOPSoft - Macro-defined alarm]** will displayed on the controller.<br>Range: M1~M1000 |

# A.2 CNC System Alarms

The system alarms are divided into MLC related alarms and NC related alarms by function.

| System alarm category | Alarm code range | Abnormal action | Description |
|---|---|---|---|
| NC alarms | 0x0000 ~ 0x1FFF | NC error | The alarms in this range are the error code that terminate the execution when an error occurs during the operation of the NC system, mainly divided into system abnormalities or operation error alarms. If the reason of the error cannot be identified, please report to Delta for assistance. |
| MLC alarms | 0xA000 ~ 0xAFFF | MLC error | The alarms in this range are the warning errors reported when the MLC and HMI screen inside the controller make errors during operation. If there is a related alarm, powering on the CNC again first, and if it still cannot be resolved, please contact Delta or supplier for assistance. |
| HMI alarms | 0x8000 ~ 0x8FFF | HMI error | |
| Servo related alarms | 0xF002 | - | The error code is an alarm when the errors of servo axes occur during system operation. When the error occurs, the servo alarm message will display on the screen. |

## A.2.1 NC Instruction Errors

| Alarm code (Hex) | Name | Cause and correction |
|---|---|---|
| 0x0000 | NC internal error. | An internal error occurred in the control system, please contact Delta or supplier. |
| 0x0002 | The specified line number or N line number cannot be found. | The specified file message cannot be found. |
| 0x0004 | Macro file call error. | Macro number error. |
| 0x0005 | Macro call mode error. | System mode error. |
| 0x0201 | Undefined operator. | Incorrect syntax. (Undefined symbol)<br>Wrong example:<br>G01O100.; “O” is undefined. |
| 0x0202 | Repeated operators command in one line. | Incorrect syntax. Please correct the syntax.<br>Wrong example:<br>G01 X100 G01 X50. |
| 0x0203 | Operators that must be command at the beginning of the block are not written at the beginning of the line. | Specific commands need to be used at the beginning of the line.<br>Wrong example:<br>G01 IF ➔ IF misused.<br>G01 WHILE ➔ WHILE is misused. |
| 0x0204 | Operators that are forbidden to be command at the beginning of the block are written at the beginning of the line. | Specific commands cannot be used at the beginning of the line.<br>Ex: DO, THEN |
| 0x0205 | Operators that must be command at the end of the block are not written at the end of the line. | Specific commands are not used at the end of the line.<br>Wrong example:<br>END 1 X100. ➔ END misused. |
| 0x0206 | Operators’ syntax error. | Wrong example:<br>#100=*3 ➔ character * misused. |
| 0x0207 | Command syntax error. | Wrong example:<br>X10. G1 ➔ command “X10.” misused. |
| 0x0208 | Command syntax error.<br>Statements and motion command cannot program in the same line. | Wrong example:<br>#1=2 G01 X10.<br>➔”#1=2” and “G01 X10.” cannot be on the same line. |
| 0x0209 | Command syntax error.<br>Wrong use of brackets. | “[“ and ”]” are not paired. |
| 0x020A | Command syntax error.<br>Wrong command characters. | Example error:<br>#1=BIT[2,SIN[3]]. The command content is incorrect with a comma. Only numbers or ‘#’ for arithmetic operations are allowed. |
| 0x020B | Command syntax error.<br>Cannot resolve multi-input operators. | Multi-input instructions such as ATAN can’t recognized by the system when it comes to negative numbers or the minus sign. |
| 0x020C | Command syntax error.<br>Newline character error. | Newline character incomplete. |
| 0x020D | Command syntax error. Violation of judgement operator in command syntax. | Syntax error. Correct the syntax format.<br>Invalid syntax for judgement operators such as ‘>’, ‘<’, ‘<>’, ‘=’.<br>Error example:<br>#1>G01. |
| 0x020E | Command syntax error.<br>IF command error. | Syntax error. Correct the syntax format.<br>There is no condition after the IF judgement. |
| 0x020F | Command syntax error. WHILE condition statement error. | Syntax error. Correct the syntax format.<br>There is no condition after the WHILE statement. |

A

| Alarm code (Hex) | Name | Cause and correction |
|---|---|---|
| 0x0210 | Command syntax error. Divisor is 0. | Syntax error. Correct the syntax format.<br>Expression divided by zero, unable to compute. |
| 0x0211 | Command syntax error. Assignment operator used incorrectly. | Syntax error. Correct the syntax format.<br>Incorrect syntax for the ‘=’ operator.<br>Error example:<br>#1=G01. |
| 0x0212 | The macro program is being called more times than the allowed limit. | Subprogram or macro program stack exceeds 8 levels. |
| 0x0213 | Command syntax error. WHILE loop command without END instruction. | Syntax error. Correct the syntax format.<br>The WHILE statement lacks the END instruction. |
| 0x0214 | Command syntax error. The loop count for DO is not an integer. | The character following DO can only be an integer. |
| 0x0215 | Command syntax error. The WHILE command lacks a DO instruction in its syntax. | Missing DO instruction in the syntax of the WHILE command. |
| 0x0216 | Command syntax error. The relevant instruction is missing the D parameter. | Missing usage of the D parameter in the command syntax.<br>Error example:<br>G42 X100. |
| 0x0217 | Command syntax error. The value of D exceeds the maximum limit. | The value of D parameter exceeds the maximum limit.<br>Error example:<br>G42 D1234567. |
| 0x0218 | Command syntax error. The relevant instruction is missing the H parameter. | Missing usage of the H parameter in the command syntax.<br>Error example:<br>G43 X100. |
| 0x0219 | Command syntax error. The value of H exceeds the maximum limit. | The value of H parameter exceeds the maximum limit.<br>Error example:<br>G43 H1234567. |
| 0x021A | The T value of lathe tool compensation exceeds the maximum limit. | Error example:<br>M06 T1234567. |
| 0x021B | Error in using the G53.1 command. | The G53.1 command must be used after the G68.2 command. |
| 0x021C | The T value exceeds the maximum limit. | Error example:<br>M06 T1234567. |
| 0x021D | Not supported G code instruction. | Syntax error. Correct the syntax format.<br>G code number not supported. |
| 0x021E | The coordinates instruction of robot system command is incorrect. | Syntax of [X, Y, Z] or [J1, J2, J3] command error. Correct the syntax format.<br>Correct format:<br>G01.1 X_Y_Z_A_B_C_ P0000 H0 R0 Q0<br>When Q is 0,1 or 2, use XYZABC.<br>G01.1 J1_ J2_ J3_ P0000 H0 R0 Q3<br>When Q is 3, use J1, J2, …, J6. |
| 0x021F | The tool number of robot system command is incorrect. | Syntax of H command error of H. Correct the syntax format.<br>Correct format:<br>G01.1 X_Y_Z_A_B_C_ P0000 H0 R0 Q0 |
| 0x0220 | The workpiece number of robot system command is incorrect. | Syntax of R command error. Correct the syntax format.<br>Correct format:<br>G01.1 X_Y_Z_A_B_C_ P0000 H0 R0 Q0 |

| Alarm code (Hex) | Name | Cause and correction |
|---|---|---|
| 0x0221 | The coordinates setting of robot system command is incorrect. | Syntax of Q command error. Correct the syntax format.<br>Correct format:<br>G01.1 X_Y_Z_A_B_C_ P0000 H0 R0 Q0<br>Q: setting range from 0 to 3. |
| 0x0222 | Command syntax error.<br>No value specified in the instruction. | Syntax error. Correct the syntax format.<br>Error example:<br>G01 X Y100. ➔ No value specified after X. |
| 0x0223 | Command syntax error.<br>Coordinate command value exceeds the range. | Syntax error. Correct the syntax format.<br>Correct format:<br>G01 X987654.321<br>G01 X98765432.1<br>The maximum number of digits is 9. |
| 0x0224 | Command syntax error.<br>GOTO statement error. | Syntax error. Correct the syntax format.<br>Error example:<br>GOTO 2<br>N2 can't be found in the NC program. |
| 0x0225 | Command syntax error.<br>END statement error. | Syntax error. Correct the syntax format.<br>Error example:<br>END<br>No value specified after END. |
| 0x0226 | Command syntax error.<br>DO statement error. | Syntax error. Correct the syntax format.<br>Error example:<br>WHILE DO<br>No value specified after DO. |
| 0x0229 | Operational error.<br>The specified file line number can't be found. | Error example:<br>M98 Q10<br>Specifies line 10, but the NC program only contains 9 lines of content. |
| 0x022A | Operational error.<br>The specified N line number can't be found. | Error example:<br>M98 H10<br>Specifies line 10, but there is no N10 line number instruction in the NC program. |
| 0x022B | Operational error.<br>The specified DO loop instruction can't be found. | Syntax error. Correct the syntax format.<br>The END command doesn't have a corresponding DO. |
| 0x022C | Operational error.<br>The specified END can't be found. | The specified number of DO layers doesn't have corresponding END layer number.<br>For example: While using WHILE DO 2, it should be ended with END 2. |
| 0x022D | Operational error.<br>Subprogram calling layers exceeds the maximum limit. | The maximum layer of subprogram call or macro stack is 8. |
| 0x022E | Operational error.<br>Missing P command in M98 subprogram calling function. | Missing P_ definition for the subprogram in M98 subprogram calling function. |
| 0x022F | Operational error.<br>Missing B command after G122. | B_ is not used with G122 command. |
| 0x0230 | Operational error.<br>G122 B command misused. | B_ misused in G122 command. |
| 0x0231 | Operational error.<br>The axis of drilling cycle is incorrect. | The axis used in the drilling cycle command doesn't exist. |
| 0x0232 | Operational error.<br>Missing P_ or Q_ in the drilling cycle command. | The drilling cycle command doesn't apply P or Q. |

A

| Alarm code (Hex) | Name | Cause and correction |
|---|---|---|
| 0x0233 | Operational error.<br>The drilling cycle command can't find the specified Q label. | The Q label defined by the drilling cycle command can't be found in the NC program. |
| 0x0234 | Operational error.<br>Unable to find the specified N line number or file line number during breakpoint search. | While using breakpoint search, there is no corresponding N line number or file line number in the NC program. |
| 0x0235 | The # variable number exceeds the usage range. | Using an undefined # variable number or when the odd addresses are used between the # variable range #25128 ~ #25255 and #25384 ~ #25511 while using the floating-point MLC variable type with **[N1.010 Bit7=1]**, which occupies 32 bits of memory space. |
| 0x0236 | Error in the format of the extended workpiece coordinates. | The extended value for workpiece coordinates must be an integer.<br>G54 P_ ➔ The parameter of P command must be an integer. |
| 0x0237 | The extended workpiece coordinates exceed the range. | The extended workpiece coordinates exceed the maximum limit.<br>The range is G54 P1 to G54 P256. |
| 0x0238 | IF command error. | Correct the syntax of IF. |
| 0x0239 | THEN command error. | Correct the syntax of IF and THEN. |
| 0x023A | ELSEIF command error. | Correct the syntax of IF and ELSEIF. |
| 0x023B | ELSE command error. | Correct the syntax of IF and ELSE. |
| 0x023C | ENDIF command error. | Correct the syntax of IF and ENDIF. |
| 0x023D | Cannot find the specified ENDIF. | The NC program has IF statement but doesn't have the corresponding ENDIF. |
| 0x023E | The layers used of IF command exceed the maximum limit. | The maximum layers used of IF command is 5. |
| 0x0241 | The P command is not marked. | The P command is not marked with the corresponding number. |
| 0x0242 | The value of P command exceeds the maximum value. | The limit value of P command is 1 to 256. |
| 0x0243 | The P command value is incorrect. | Syntax error. Correct the syntax format. |
| 0x0244 | The setting value of the # variable exceeds the limit range. | The setting value of the special # variable exceeds the allowed range.<br>Error example:<br>#20020 = 32768<br>(The value range of #20020 is 0 to 32767). |
| 0x0245 | The command P or Q is not used in the nested loop. | Syntax error. Correct the syntax format. |
| 0x0246 | The cutting amount in the roughing cycle is invalid. | Syntax error. Correct the syntax format. |
| 0x0247 | The nested loop must be followed by 2 consecutive blocks of instructions. | Syntax error. Correct the syntax format. |
| 0x0248 | The number of M codes used in a single block instruction exceeds the limit. | The maximum number of M codes allowed in a single block instruction is 4. |
| 0x0249 | M code usage error. | Syntax error. Correct the syntax format.<br>Error example: M98 M1 M3 |
| 0x024B | Call instruction usage error. | Error example:<br>M98 G65<br>One block can only have one call command. |
| 0x024D | Multi Z axis application with tool compensation cannot specify the movement amount of the slave axis. | Syntax error. Correct the syntax format. |

| Alarm code (Hex) | Name | Cause and correction |
|---|---|---|
| 0x024E | Usage error of the HSI value of multi-Z axis. | The trigger contact of G31 contains undefined contacts. |
| 0x024F | Usage error of V_ or XYZABCUVW_ in the G31 skip function of multi-Z axis. | G31 command for the slave axis was used in the NC program. |
| 0x0250 | Usage error of the input function of the multi-Z axis in the G10 programmable parameter function. | Syntax error of G10 L50. |
| 0x0251 | Error parameters of the call subprogram function. Only P_, H(Q)_, L_, can be used. | Error example:<br>M98 K5<br>K5 is not supported by M98. |
| 0x0252 | Usage error of read and write servo parameters of G10 programmable parameters function. | Syntax error of G10 L40 / G10 L41. |
| 0x0253 | Data error of the intermedia point for returning to the reference point. | Syntax error of G28. |
| 0x0254 | Attempt to assign a value to a ready-only # variable. | Error example:<br>#20005 = 10<br>#20005 is a constant value π (3.1415) and cannot be written. |
| 0x0255 | The number of interpolation axes in a single block exceeds the limit. | The maximum number of interpolation axes in a single section is 16. |
| 0x0256 | Usage error of the arithmetic command. | Syntax error of the arithmetic command.<br>Error example:<br>#100=BIT[1]<br>The BIT command requires 2 input values for calculation, but only one set of value is provided. |
| 0x0257 | Input numerical error in arithmetic command. | Syntax numerical error of the arithmetic command.<br>Error example:<br>B=ASIN[A]<br>The range of A value is not between -1 to 1. |
| 0x0258 | Decimal point is not allowed. | The value of the command does not allow decimal points.<br>Error example:<br>G04 P1.2<br>The parameter of P does not support decimal values. |
| 0x0259 | The radius of the arc command is incorrect. | The distance between the center and the end point calculated by the arc command exceeds the arc radius tolerance of **N1.040**. |
| 0x025A | The center of the arc command is incorrect. | The distance between the center and the end point calculated by the arc command exceeds the arc radius tolerance of **N1.040**. |
| 0x025B | 3-dimension arc command error. | The distance between the center and the end point calculated by the 3-dimension arc command exceeds the arc radius tolerance of **N1.040**. |
| 0x025C | Blending function data registration error | G10 L60 blending function data registration error. Please check the value of the P command. |
| 0x025D | Working plane setting error. | While enabling synchronize axis function, the plane selections such as G17, G18 and G19 have the same axis, or the plane selection is incorrect. |

A

| Alarm code (Hex) | Name | Cause and correction |
|---|---|---|
| 0x025E | System obtained NULL pointers. | RBTask obtained NULL pointers. Controller malfunction, please contact Delta or supplier for assistance. |
| 0x025F | Not supported instruction for the scaling function. | When using the G68 rotation command, it is not allowed to use 3-dimension arc command, G28/G29/G30, and G54~G59 commands simultaneously. If you need to use them, you must first cancel the G68 function with G69. |
| 0x0260 | Insufficient information for circle command by three points method. | The input information used for circle command by three points is insufficient. |
| 0x0261 | Subprogram calling function missing H_ statement. | Syntax error. Correct the syntax format. |
| 0x0262 | Switching arm gesture is prohibited during arm moving. | Switching hand orientation is prohibited during motion. |
| 0x0263 | File not found. | Cannot find the specified NC program file. |
| 0x0264 | The P_Q_ command does not exist in looping command. | Make sure the NC program file includes the N line number or file line number for specified by P_Q_. |
| 0x0265 | G53 or G28/ G30 commands error | G53 and G28/G30 commands cannot be used in the same block. |

A

## A.2.2 NC Axis Alarms

| Alarm code (Hex) | Name | Cause and correction |
|---|---|---|
| 0x0401 | Emergency stop signal is triggered. | Exclude the EMG signal. |
| 0x0402 | Hardware limit signal is triggered. | Exclude the hardware limit signal. |
| 0x0601 | Axes reference information error | Communication is disconnected after established.<br>Reset after reconnecting the physical cable.<br>When the alarm occurs, the origin state of the axis will be canceled if it's an incremental type of axis. |
| 0x0602 | Overspeed protection. | System overspeed protection.<br>The axis will servo off when the alarm occurs. It will be automatically servo on again after reset command and clearing the alarm. |
| 0x0A01 | The 1st segment of software limit is triggered. | The axis has reached the software limit position of 1st segment. The machine must move away from the limit position before system reset. |
| 0x0A02 | The status of 1st segment of software limit needs to be cleared. | After moving away from the 1st segment of software limit, this alarm will be displayed. Please reset the machine. |
| 0x0A03 | The 2nd segment of software limit is triggered. | The axis has reached the software limit position of 2nd segment. The machine must move away from the limit position before system reset. |
| 0x0A04 | The status of 2nd segment of software limit needs to be cleared. | After moving away from the 2nd segment of software limit, this alarm will be displayed. Please reset the machine. |
| 0x0A05 | Hardware limit triggered. | The axis has reached the hardware limit position. The machine must move away from the limit position before system reset. |
| 0x0A06 | The status of hardware limit needs to be cleared. | After moving away from the hardware limit, this alarm will be displayed. Please reset the machine. |
| 0x0A07 | Servo is not ready. | The axis is not servo ON. |
| 0x0A08 | Origin status of the axis is missing. | Please execute the homing operation. |
| 0x0C01 | Spindle positioning failed. | Verify the spindle positioning process or check if the spindle positioning time setting is incorrect.<br>**N0.1025**, Positioning check time for 1st spindle.<br>**N0.1075**, Positioning check time for 2nd spindle.<br>**N0.1125**, Positioning check time for 3rd spindle.<br>…<br>**N0.1375**, Positioning check time for 8th spindle. |
| 0x0C02 | Spindle is not rotating before cutting. | **[N1.011 bit8 the spindle enables checks before cutting]** is set to 1, and the spindle is not rotating when executing the motion command. |
| 0x0C03 | The spindle speed setting exceeds the maximum speed. | **[N0.1000 bit7 the spindle maximum speed command check]** is set to 1, and the S value in the NC instruction exceeds the maximum spindle speed setting in **N0.1008**. |
| 0x0C04 | Incorrect switching of the spindle and C-axis. | Make sure all channel parameters are set, whether the C-axis and the spindle are both configured as active axes. |
| 0x0C05 | The spindle C-axis is not set as rotational axis. | Check the C axis setting of **N2.001 bits 2 to 4** for the rotational axis feed mode. |
| 0x0C06 | The spindle is not enabled. | Please check if the spindle is enabled. |

A

| | | |
|---|---|---|
| 0x1601 | The hardware limit is triggered during execution of home mode 4.<br>The search direction does not match. | The hardware limit is triggered during homing. Please check the following:<br>1. Make sure the machine is out of any limit positions before executing the homing procedure.<br>2. Verify the wiring signals for the hardware limits are functioning correctly.<br>3. Chek if the homing search direction is correct. |
| 0x1602 | The hardware limit alarm not cleared of the home mode 4. | This alarm will be triggered after clearing 0x1601 hardware limit alarm. |
| 0x1603 | Oscillation function is not allowed in the home mode. | The M2x264 oscillation function cannot be enabled in the homing mode. |
| 0x1604 | The home mode is not supported. | Please confirm the parameter settings of **N2.50** homing mode. |
| 0x1605 | The axis is set as non-rotary axis in home mode 6. | Please confirm the parameter settings of **N2.01** axis configuration. |
| 0xF002 | The alarm codes of Delta servo. | The alarm of 0xF002 needs to be interpreted in conjunction with the alarm codes of the Delta servo.<br>For example, if the Delta servo displays an alarm code 0x09 (following error), the controller will display 0xF002 and indicate the servo’s alarm code as 0x09. |

A

## A.2.3 NC Motion Interpolation Alarms

| Alarm code (Hex) | Name | Cause and correction |
|---|---|---|
| 0x1001 | Feed rate is not specified. | The NC program needs to specify F or set default feed rate value. |
| 0x1201 | Block command error. | Command block error. |
| 0x1202 | The repeated points of tool compensation have exceeded the maximum limit. | The repeated points exceed the maximum limit. |
| 0x1203 | Plane changing after enabling tool compensation. | Changing the plane is prohibited after enabling tool compensation. |
| 0x1204 | Unable to calculate the tool compensation path at arc-to-linear command. | Unable to calculate the tool compensation path. |
| 0x1205 | Tool compensation path interference occurred at arc-to-linear. | Tool compensation result has path interference.<br>For example:<br>The arc entry is smaller than the tool diameter. |
| 0x1206 | Changing tool diameter during arc command performing. | Tool diameter cannot be changed during arc command while tool compensation is enabled. |
| 0x1207 | Unable to calculate the tool compensation path at arc-to-arc command. | Unable to calculate the tool compensation path. |
| 0x1208 | Tool compensation path interference occurred between linear-to-arc. | Tool compensation result has path interference.<br>For example:<br>The arc entry is smaller than the tool diameter. |
| 0x1209 | Unable to calculate the tool compensation path at linear-to-arc command. | Unable to calculate the tool compensation pat. |
| 0x120A | The arc radius is less than 0 after tool compensation. | Tool compensation result has path interference. |
| 0x120B | Tool compensation path interference occurred at two linear commands. | Tool compensation result has path interference. |
| 0x120C | Calculation error at the arm path interpolation. | The endpoint position is outside the working range of the robot arm. |
| 0x120D | The P value of G05 command does not correspond to HMI recipe. | The corresponding recipe values must be filled in within the HMI interface. |
| 0x120E | Path interpolation distance exceeds the maximum travel distance. | Motion coordinate point exceeds the maximum distance.<br>Error example:<br>X[tan[90]] ➔ tan[90] is a infinity value<br>Please correct the coordinate position of the motion command. |

A

## A.2.4 NC Kernel Alarms

| Alarm code (Hex) | Name | Cause and correction |
|---|---|---|
| 0x1401 | System kernel construction failed. | Controller malfunction, please contact Delta or supplier for assistance. |
| 0x1402 | System kernel RT construction failed. | Controller malfunction, please contact Delta or supplier for assistance. |
| 0x1403 | System kernel RT memory error. | Controller malfunction, please contact Delta or supplier for assistance. |
| 0x1404 | System RT deployment configuration error. | Controller malfunction, please contact Delta or supplier for assistance. |
| 0x1405 | Path UTY construction failed. | Controller malfunction, please contact Delta or supplier for assistance. |
| 0x1406 | Path NRT construction failed. | Controller malfunction, please contact Delta or supplier for assistance. |
| 0x1407 | Path RT construction failed. | Controller malfunction, please contact Delta or supplier for assistance. |
| 0x1408 | Path UTY memory error. | Controller malfunction, please contact Delta or supplier for assistance. |
| 0x1409 | Path NRT memory error. | Controller malfunction, please contact Delta or supplier for assistance. |
| 0x140A | Path RT memory error. | Controller malfunction, please contact Delta or supplier for assistance. |
| 0x140B | Retentive # variable error. | Controller malfunction, please contact Delta or supplier for assistance. |
| 0x140C | Path UTY deployment parameter error. | Parameter error. |
| 0x140D | Path UTY deployment configuration error. | Parameter error. |
| 0x140E | Path NRT association error. | Controller malfunction, please contact Delta or supplier for assistance. |
| 0x140F | Path NRT deployment parameter error. | Parameter error. |
| 0x1410 | Path NRT deployment configuration error. | Parameter error. |
| 0x1411 | Path RT association error. | Controller malfunction, please contact Delta or supplier for assistance. |
| 0x1412 | Path RT deployment parameter error. | Parameter error. |
| 0x1413 | Path RT deployment configuration error. | Parameter error. |
| 0x1414 | Path NRT registration # variable error. | Controller malfunction, please contact Delta or supplier for assistance. |
| 0x1415 | Virtual axis type error. | The system axis parameters have both physical and virtual axis setting on the same axis. |
| 0x1416 | The slave address does not match. | The slave address is different from the system address setting. Please confirm the slave address or reset. |
| 0x1417 | The slave address is duplicated. | Duplicate slave addresses are being used. Please check the address setting of the slave devices and eliminate the duplicate addresses. |
| 0x1418 | The number of channels in the multi-channel is incorrect. | The NC multi-channel must be used sequentially and cannot be used with only channel one and channel three. |
| 0x1419 | NC axis number setting error. | Controller malfunction, please contact Delta or supplier for assistance. |
| 0x141A | Spindle number setting error. | Controller malfunction, please contact Delta or supplier for assistance. |
| 0x141B | Coding of the tool magazine setting error. | Controller malfunction, please contact Delta or supplier for assistance. |

| | | |
|---|---|---|
| 0x141C | Setting error of the axis type. | The axis type setting, such as NC axis or PLC axis, is incorrect. Please contact Delta or supplier for assistance. |
| 0x141D | The number of Cartesian axes and joint axes in the path dose not match. | Controller malfunction, please contact Delta or supplier for assistance. |
| 0x141E | System PDO mapping mismatch. | Controller malfunction, please contact Delta or supplier for assistance.<br>The system DAT configuration does not match the connected products. |
| 0x141F | The system interrupt execution time is not supported. | The servo does not support the current interpolation time. |
| 0x1420 | EtherCAT communication establish failed. | Check if the communication cables are properly connected to the slave devices. |
| 0x1426 | The system remote module (EIO) failed. | Verify if there are any conflicts with the settings in the controller EIO interface and confirm the changes to save the modified parameter values. |
| 0x1427 | Failed to trigger the EtherCAT communication OP mode. | Check if the fieldbus is properly connected. |
| 0x1428 | Failed to switch the EtherCAT communication to OP mode. | Check if the fieldbus is properly connected. |
| 0x1429 | EtherCAT communication initialization error. | Please restart the system. |
| 0x142A | The number of axes defined in the channel exceeds the limit. | Please check the parameter definition of the channel axis. |
| 0x142B | EtherCAT slave devices not found. | Check if the cables are properly connected. |
| 0x142C | Conflicting setting between multi-Z and synchronization function. | Check for any conflicts between the multi-Z parameters and the synchronization control parameter **N2.015**. |
| 0x142D | Setting error of the multi-Z machine parameters. | Check if the setting of multi-Z parameters is correct. |
| 0x142E | The system remote module (EIO) connection device type mismatch with the settings. | Check if the EIO module type settings match the devices on the fieldbus. |
| 0x142F | The corresponding port number for the remote module (EIO) connection was not found. | Check if the port number setting for the EIO module is correct. |

# Special Memory

# B

This chapter introduces CNC system definition device list. It supplies users to check MLC special # variable action correct or not in CNC system.

# B CNC Special # Variable

This appendix provides functional mapping tables of # variables in the NC 5 controller system. Variables are independently working in each channel of NC, mainly used in operation, read and system setting of NC program.

System # variable corresponding table define as following:

| Variable | Description | Type |
|---|---|---|
| #0 | system variable of value is null (NULL) | R/-/- |

1. # variable name, beginning as # and number from 0 to 223999. Number after 20000 are system variables or reserved variables.
2. # variable function description.
3. # variable operation attribute and its attribute description:

   R: # variable read only.

   W: # variable writable.

   ●: When NC system execute this # variable, it will stop look-ahead and block preview which ensure the system can obtain the current state of the NC system.

## B.1 Constant Variable

### B.1.1 Null Variable (#0)

<table>
<tr><th>Variable</th><th>Description</th><th>Type</th></tr>
<tr><td>#0</td><td>A null and read only variable (Null).<br>Users can move #0 value into specified variable, and then set this specified variable as null. (The value will display as NULL).<br>Null (NULL) and 0 is different in usage, the following table describes the differences when the #100 is given in NULL or 0:
<table>
<tr><th rowspan="2">Judgement</th><th colspan="2">Value of #100</th></tr>
<tr><th>NULL</th><th>0</th></tr>
<tr><td>#100 == #0</td><td>True</td><td>False</td></tr>
<tr><td>#100 != #0</td><td>False</td><td>True</td></tr>
<tr><td>#100 >= #0</td><td>True</td><td>True</td></tr>
<tr><td>#100 <= #0</td><td>True</td><td>True</td></tr>
<tr><td>#100 > 0</td><td>False</td><td>False</td></tr>
<tr><td>#100 < 0</td><td>False</td><td>False</td></tr>
<tr><td>#100 != 0</td><td>True</td><td>False</td></tr>
<tr><td>G90X99Y#100</td><td>G90 X99</td><td>G90 X99 Y0</td></tr>
</table></td><td>R/-/-</td></tr>
</table>

B

## B.1.2 Constant Variable (#20000~#20012)

| Variable | Description | Type |
|---|---|---|
| #20000 | $e \cong 2.718281828$ | R/-/- |
| #20001 | $\log_2 e \cong 1.442695041$ | R/-/- |
| #20002 | $\log_{10} e \cong 0.434294482$ | R/-/- |
| #20003 | $\ln 2 \cong 0.693147181$ | R/-/- |
| #20004 | $\ln 10 \cong 2.302585093$ | R/-/- |
| #20005 | $\pi \cong 3.141592654$ | R/-/- |
| #20006 | $\frac{\pi}{2} \cong 1.570796327$ | R/-/- |
| #20007 | $\frac{\pi}{4} \cong 0.785398163$ | R/-/- |
| #20008 | $\frac{1}{\pi} \cong 0.318309886$ | R/-/- |
| #20009 | $\frac{2}{\pi} \cong 0.636619772$ | R/-/- |
| #20010 | $\frac{2}{\sqrt{\pi}} \cong 1.128379167$ | R/-/- |
| #20011 | $\sqrt{2} \cong 1.414213562$ | R/-/- |
| #20012 | $\frac{1}{\sqrt{\pi}} \cong 0.707106781$ | R/-/- |

B

# B.2 Alarm Variable

## B.2.1 Macro Defined Alarm (#20020)

| Variable | Description | Type |
|---|---|---|
| #20020 | Macro defined alarm (MR)<br>When non-zero value is written into this variable, the system will display Macro defined alarm on the controller screen.<br>Value range: from 0 to 32767. If the value is not in the range, the system will return alarm 0x0244.<br>The error message from MR1 to MR1000 are configured in the **[CNCSoft]** – **[DOPSoft]**.<br>For example: #20020 = 100; the system displays MR100. | R/W/● |

# B.3 Field Special Variables

## B.3.1 User-Defined Coordinate (#20021)

| Variable | Description | Type |
|---|---|---|
| #20021 | Special for woodworking machine, used for displaying the working coordinate of the multi-channel.<br>*Need to be used with file sequence function | R/-/● |

## B.3.2 Multi-Z Axis Synchronize (#20022)

| Variable | Description | Type |
|---|---|---|
| #20022 | Slave axes number shielding of multi-Z axis.<br>This value is associated with D3X014. | R/-/● |

B

# B.4 NC System Status

## B.4.1 Servo Parameter Read and Write (#20023, #20024)

| Variable | Description | Type |
|---|---|---|
| #20023 | This variable store the servo parameter which return from the slave parameters reading instruction.<br>Servo parameters reading instruction includes G10 L40, L41, L45 and L46.<br>For example:<br>After executing G10 L40 I3 P4 D5, this variable will store the value of P4-5 from the servo drive of address number 3.<br>After executing G10 L45 I3 P4 D5, this variable will store the value of [OD-code 4] [Sub OD code 5] from the slave device of address number 3. | R/-/● |
| #20024 | This variable store the execution result (error code) after executing the slave parameters reading instruction. Servo parameters reading instruction includes G10 L40, L41, L45 and L46.<br>For example:<br>After executing G10 L40 I3 P4 D5, this variable will store the execution result (error code).<br>When executing G10 L45 I3 P4 D5, this variable will store the execution result (error code). | R/-/● |

## B.4.2 System Time (#20025, #20026)

| Variable | Description | Type |
|---|---|---|
| #20025 | System time:  YYMMDD<br>For example: controller system time is 2023/04/17, the value of #20025 is 230417 | R/-/● |
| #20026 | System time:  hhmmss<br>For example: controller system time is 17:30:01, the value of #20026 is 173001 | R/-/● |

## B.4.3 NC System Mode (#20300~#20330)

| Variable | Function description (Milling machine type G code) | Type |
|---|---|---|
| #20300 | G code group [0] mode: Temporary G command information | R/-/- |
| #20301 | G code group [1] mode: Interpolation mode (G00~G03) | R/-/- |
| #20302 | G code group [2] mode: Plane selection (G17~G19) | R/-/- |
| #20303 | G code group [3] mode: Absolute/ incremental instruction (G90, G91). | R/-/- |
| #20304 | G code group [4] mode: Procedure check (G22, G23). | R/-/- |
| #20305 | G code group [5] mode: Feeding mode G94, G95. | R/-/- |
| #20306 | G code group [6] mode: Metric/ inch mode (G20, G21). | R/-/- |
| #20307 | G code group [7] mode: Tool radius compensation (G40, G41, G42). | R/-/- |
| #20308 | G code group [8] mode: Tool length compensation (G43, G44, G49). | R/-/- |
| #20309 | G code group [9] mode: Circulation instruction (G80). | R/-/- |
| #20310 | G code group [10] mode: Drilling return mode (G98, G99). | R/-/- |
| #20311 | G code group [11] mode: Ratio mode (G50, G51). | R/-/- |
| #20312 | G code group [12] mode: Workpiece coordinate (G54~G59). | R/-/- |
| #20313 | G code group [13] mode: Cutting mode (G61, G64). | R/-/- |
| #20314 | G code group [14] mode: Macro call (G66, G67). | R/-/- |
| #20315 | G code group [15] mode: Coordinate rotation (G68, G69). | R/-/- |
| #20316 | G code group [16] mode: Polar coordinate instruction (G15, G16). | R/-/- |
| #20317 | G code group [17] mode: Cut speed (G96, G97). | R/-/- |
| #20318 | G code group [18] mode: Mirror function (G24, G25). | R/-/- |
| #20319<br>…#20330 | System reserved. Do not use this sector variable. | -/-/- |
| #20331 | G code group [31] mode: Extend work coordinate (G54 P1-P64). | R/-/- |

B

### B.4.4 NC Command Status (#20400~#20411)

| Variable NO. | Function description | Type |
|---|---|---|
| #20400 | Current execution feed rate F.<br>This variable records the last F command before the specified search break block when executing breakpoint search function. | R/-/- |
| #20401 | Current tool compensation H value<br>This variable records the last H command before the specified search break block when executing breakpoint search function. | R/-/- |
| #20402 | Current tool compensation D value<br>This variable records the last D command before the specified search break block when executing breakpoint search function. | R/-/- |
| #20403 | Current tool number T value<br>This variable records the last T command before the specified search break block when executing breakpoint search function. | R/-/- |
| #20404 | 1st spindle current program speed S. | R/-/● |
| #20405 | 2nd spindle current program speed S. | R/-/● |
| #20406 | 3rd spindle current program speed S. | R/-/● |
| #20407 | 4th spindle current program speed S. | R/-/● |
| #20408<br>…<br>#20411 | System reserved.<br>Do not use variables in this range. | R/-/● |

### B.4.5 Coordinate Information (#21000~#21079)

| | Machine Coordinate | Absolute Coordinate | Target Position | G31 Machine | G31 Absolute | Axis Compensate Feedback | Type |
|---|---|---|---|---|---|---|---|
| 1st axis (X) | #21000 | #21016 | #21032 | #21048 | #21064 | #21080 | R/-/● |
| 2nd axis (Y) | #21001 | #21017 | #21033 | #21049 | #21065 | #21081 | R/-/● |
| 3rd axis (Z) | #21002 | #21018 | #21034 | #21050 | #21066 | #21082 | R/-/● |
| 4th axis (A) | #21003 | #21019 | #21035 | #21051 | #21067 | #21083 | R/-/● |
| 5th axis (B) | #21004 | #21020 | #21036 | #21052 | #21068 | #21084 | R/-/● |
| 6th axis (C) | #21005 | #21021 | #21037 | #21053 | #21069 | #21085 | R/-/● |
| 7th axis (U) | #21006 | #21022 | #21038 | #21054 | #21070 | #21086 | R/-/● |
| 8th axis (V) | #21007 | #21023 | #21039 | #21055 | #21071 | #21087 | R/-/● |
| 9th axis (W) | #21008 | #21024 | #21040 | #21056 | #21072 | #21088 | R/-/● |
| 10th axis | #21009 | #21025 | #21041 | #21057 | #21073 | #21089 | R/-/● |
| 11th axis | #21010 | #21026 | #21042 | #21058 | #21074 | #21090 | R/-/● |
| 12th axis | #21011 | #21027 | #21043 | #21059 | #21075 | #21091 | R/-/● |
| 13th axis | #21012 | #21028 | #21044 | #21060 | #21076 | #21092 | R/-/● |
| 14th axis | #21013 | #21029 | #21045 | #21061 | #21077 | #21093 | R/-/● |
| 15th axis | #21014 | #21030 | #21046 | #21062 | #21078 | #21094 | R/-/● |
| 16th axis | #21015 | #21031 | #21047 | #21063 | #21079 | #21095 | R/-/● |

B

## B.4.6 G54-G59 Workpiece Coordinate (#22000~#23183)

| | Offset Position | G54 | G55 | G56 | G57 | G58 | G59 | Type |
|---|---|---|---|---|---|---|---|---|
| 1st axis (X) | #22000 | #22002 | #22003 | #22004 | #22005 | #22006 | #22007 | R/W/● |
| 2nd axis (Y) | #22078 | #22080 | #22081 | #22082 | #22083 | #22084 | #22085 | R/W/● |
| 3rd axis (Z) | #22156 | #22158 | #22159 | #22160 | #22161 | #22162 | #22163 | R/W/● |
| 4th axis (A) | #22234 | #22236 | #22237 | #22238 | #22239 | #22240 | #22241 | R/W/● |
| 5th axis (B) | #22312 | #22314 | #22315 | #22316 | #22317 | #22318 | #22319 | R/W/● |
| 6th axis (C) | #22390 | #22392 | #22393 | #22394 | #22395 | #22396 | #22397 | R/W/● |
| 7th axis (U) | #22468 | #22470 | #22471 | #22472 | #22473 | #22474 | #22475 | R/W/● |
| 8th axis (V) | #22546 | #22548 | #22549 | #22550 | #22551 | #22552 | #22553 | R/W/● |
| 9th axis(W) | #22624 | #22626 | #22627 | #22628 | #22629 | #22630 | #22631 | R/W/● |
| 10th axis | #22702 | #22704 | #22705 | #22706 | #22707 | #22708 | #22709 | R/W/● |
| 11th axis | #22780 | #22782 | #22783 | #22784 | #22785 | #22786 | #22787 | R/W/● |
| 12th axis | #22858 | #22860 | #22861 | #22862 | #22863 | #22864 | #22865 | R/W/● |
| 13th axis | #22936 | #22938 | #22939 | #22940 | #22941 | #22942 | #22943 | R/W/● |
| 14th axis | #23014 | #23016 | #23017 | #23018 | #23019 | #23020 | #23021 | R/W/● |
| 15th axis | #23092 | #23094 | #23095 | #23096 | #23097 | #23098 | #23099 | R/W/● |
| 16th axis | #23170 | #23172 | #23173 | #23174 | #23175 | #23176 | #23177 | R/W/● |

| | G54 Offset Position | G55 Offset Position | G56 Offset Position | G57 Offset Position | G58 Offset Position | G59 Offset Position | Type |
|---|---|---|---|---|---|---|---|
| 1st axis (X) | #22008 | #22009 | #22010 | #22011 | #22012 | #22013 | R/W/● |
| 2nd axis (Y) | #22086 | #22087 | #22088 | #22089 | #22090 | #22091 | R/W/● |
| 3rd axis (Z) | #22164 | #22165 | #22166 | #22167 | #22168 | #22169 | R/W/● |
| 4th axis (A) | #22242 | #22243 | #22244 | #22245 | #22246 | #22247 | R/W/● |
| 5th axis (B) | #22320 | #22321 | #22322 | #22323 | #22324 | #22325 | R/W/● |
| 6th axis (C) | #22398 | #22399 | #22400 | #22401 | #22402 | #22403 | R/W/● |
| 7th axis (U) | #22476 | #22477 | #22478 | #22479 | #22480 | #22481 | R/W/● |
| 8th axis (V) | #22554 | #22555 | #22556 | #22557 | #22558 | #22559 | R/W/● |
| 9th axis(W) | #22632 | #22633 | #22634 | #22635 | #22636 | #22637 | R/W/● |
| 10th axis | #22710 | #22711 | #22712 | #22713 | #22714 | #22715 | R/W/● |
| 11th axis | #22788 | #22789 | #22790 | #22791 | #22792 | #22793 | R/W/● |
| 12th axis | #22866 | #22867 | #22868 | #22869 | #22870 | #22871 | R/W/● |
| 13th axis | #22944 | #22945 | #22946 | #22947 | #22948 | #22949 | R/W/● |
| 14th axis | #23022 | #23023 | #23024 | #23025 | #23026 | #23027 | R/W/● |
| 15th axis | #23100 | #23101 | #23102 | #23103 | #23104 | #23105 | R/W/● |
| 16th axis | #23178 | #23179 | #23180 | #23181 | #23182 | #23183 | R/W/● |

# B.5 Tool Management and Breakpoints

## B.5.1 Tool Magazine (#24001~#24004)

| Variable | Function description | Type |
|---|---|---|
| #24001 | Activated tool number from the $1^{st}$ tool magazine. | R/-/● |
| #24002 | Activated tool number from the $2^{nd}$ tool magazine. | R/-/● |
| #24003 | Tool exchange number of the $1^{st}$ tool magazine.<br>In the $1^{st}$ tool magazine, after search the [#24003 specified number] and get the current tool slot, user can exchange the [#24001 activated tool number] and the [#24003 Tool exchange number] and then the tool number in the tool slot will be changed automatically.<br>if #24003 specified tool number is not in the tool magazine, the system will not exchange the tool number. | R/W/● |
| #24004 | Tool exchange number of the $2^{nd}$ tool magazine.<br>In the $2^{nd}$ tool magazine, after search the [#24004 specified number] and get the current tool slot, user can exchange the [#24002 activated tool number] and the [#24004 Tool exchange number] and then the tool number in the tool slot will be changed automatically.<br>if #240044 specified tool number is not in the tool magazine, the system will not exchange the tool number. | R/W/● |

## B.5.2 Breakpoint Search (#24039~#24095)

After executing the breakpoint row-searching or label execution, the NC system will direct current line number to the specified breakpoint, and the status information of M, S, T, F and axis coordinates are saved as following table:

| Variable | Function Description | Type |
|---|---|---|
| #24042<br>…<br>#24076 | Last M code record before breakpoint.<br>When using breakpoint search function, those used M code before breakpoint line number or breakpoint label will record in #24042 to #24076 variables.<br>* If there are more than 35 M codes are used before the breakpoint line number, the system retains only the last 35 sets of M codes based on the first-in-first-out principle.<br>* If less than 35 M codes are used before the breakpoint line number, the corresponding variable will be NULL. | R/-/- |
| #24077 | Last S code record before breakpoint.<br>When using breakpoint search function, the last S code before breakpoint line number or breakpoint label will record in #24077 variable.<br>*If there are no S code has been used before the breakpoint line number, this variable will be null (NULL). | R/-/- |
| #24078 | Last T code record before breakpoint.<br>When using breakpoint search function, those used T code before breakpoint line number or breakpoint label will record in #24078 to #24079 variables.<br>* If there are more than 2 T codes are used before the breakpoint line number, the system retains only the last 2 sets of T codes based on the first-in-first-out principle.<br>* If less than 2 T codes are used before the breakpoint line number, the corresponding variable will be NULL. | R/-/- |
| #24079 | | R/-/- |

<table>
<tr><th>Variable</th><th colspan="2">Function description</th><th>Type</th></tr>
<tr><td>#24080</td><td>1st axis (X)</td><td rowspan="16">Target position of each axis record before breakpoint line.<br>When using breakpoint searching function, the target position before breakpoint line or breakpoint label of each axis will be store in #24080 to #24095 variables.<br>*If the axis is not used before the breakpoint line number, the corresponding variable will be null (NULL).</td><td>R/-/-</td></tr>
<tr><td>#24081</td><td>2nd axis (Y)</td><td>R/-/-</td></tr>
<tr><td>#24082</td><td>3rd axis (Z)</td><td>R/-/-</td></tr>
<tr><td>#24083</td><td>4th axis (A)</td><td>R/-/-</td></tr>
<tr><td>#24084</td><td>5th axis (B)</td><td>R/-/-</td></tr>
<tr><td>#24085</td><td>6th axis (C)</td><td>R/-/-</td></tr>
<tr><td>#24086</td><td>7th axis (U)</td><td>R/-/-</td></tr>
<tr><td>#24087</td><td>8th axis (V)</td><td>R/-/-</td></tr>
<tr><td>#24088</td><td>9th axis (W)</td><td>R/-/-</td></tr>
<tr><td>#24089</td><td>10th axis</td><td>R/-/-</td></tr>
<tr><td>#24090</td><td>11th axis</td><td>R/-/-</td></tr>
<tr><td>#24091</td><td>12th axis</td><td>R/-/-</td></tr>
<tr><td>#24092</td><td>13th axis</td><td>R/-/-</td></tr>
<tr><td>#24093</td><td>14th axis</td><td>R/-/-</td></tr>
<tr><td>#24094</td><td>15th axis</td><td>R/-/-</td></tr>
<tr><td>#24095</td><td>16th axis</td><td>R/-/-</td></tr>
<tr><td>#24100</td><td colspan="2" rowspan="4">Multiple T code record before breakpoint line.<br>When using breakpoint searching function, the T code command before breakpoint line or breakpoint label will be store in #24100 to #24103 variables.<br>Maximum 4 sets of T code will be recorded.<br>*If the multiple T code is not used before the breakpoint line number, the corresponding variable will be null (NULL).</td><td>R/-/-</td></tr>
<tr><td>#24101</td><td>R/-/-</td></tr>
<tr><td>#24102</td><td>R/-/-</td></tr>
<tr><td>#24103</td><td>R/-/-</td></tr>
</table>

B

# B.6 MLC Exchange Variable

## B.6.1 MLC M Relay Write to # Variable (#25000~#25127)

| | +0 | +1 | +2 | +3 | +4 | Type |
|---|---|---|---|---|---|---|
| #25000 | M2X128 | M2X129 | M2X130 | M2X131 | M2X132 | R/-/- |
| #25005 | M2X133 | M2X134 | M2X135 | M2X136 | M2X137 | R/-/- |
| #25010 | M2X138 | M2X139 | M2X140 | M2X141 | M2X142 | R/-/- |
| #25015 | M2X143 | M2X144 | M2X145 | M2X146 | M2X147 | R/-/- |
| #25020 | M2X148 | M2X149 | M2X150 | M2X151 | M2X152 | R/-/- |
| #25025 | M2X153 | M2X154 | M2X155 | M2X156 | M2X157 | R/-/- |
| #25030 | M2X158 | M2X159 | M2X160 | M2X161 | M2X162 | R/-/- |
| #25035 | M2X163 | M2X164 | M2X165 | M2X166 | M2X167 | R/-/- |
| #25040 | M2X168 | M2X169 | M2X170 | M2X171 | M2X172 | R/-/- |
| #25045 | M2X173 | M2X174 | M2X175 | M2X176 | M2X177 | R/-/- |
| #25050 | M2X178 | M2X179 | M2X180 | M2X181 | M2X182 | R/-/- |
| #25055 | M2X183 | M2X184 | M2X185 | M2X186 | M2X187 | R/-/- |
| #25060 | M2X188 | M2X189 | M2X190 | M2X191 | M2X192 | R/-/- |
| #25065 | M2X193 | M2X194 | M2X195 | M2X196 | M2X197 | R/-/- |
| #25070 | M2X198 | M2X199 | M2X200 | M2X201 | M2X202 | R/-/- |
| #25075 | M2X203 | M2X204 | M2X205 | M2X206 | M2X207 | R/-/- |
| #25080 | M2X208 | M2X209 | M2X210 | M2X211 | M2X212 | R/-/- |
| #25085 | M2X213 | M2X214 | M2X215 | M2X216 | M2X217 | R/-/- |
| #25090 | M2X218 | M2X219 | M2X220 | M2X221 | M2X222 | R/-/- |
| #25095 | M2X223 | M2X224 | M2X225 | M2X226 | M2X227 | R/-/- |
| #25100 | M2X228 | M2X229 | M2X230 | M2X231 | M2X232 | R/-/- |
| #25105 | M2X233 | M2X234 | M2X235 | M2X236 | M2X237 | R/-/- |
| #25110 | M2X238 | M2X239 | M2X240 | M2X241 | M2X242 | R/-/- |
| #25115 | M2X243 | M2X244 | M2X245 | M2X246 | M2X247 | R/-/- |
| #25120 | M2X248 | M2X249 | M2X250 | M2X251 | M2X252 | R/-/- |
| #25125 | M2X253 | M2X254 | M2X255 | - | - | R/-/- |

## B.6.2 MLC Word D Register Write to # Variable (#25128~#25255)

B

This function effected by the parameter **[N1.010 Bit7 MLC variable status]**. When the parameter set as 0 single word type, the D register uses 16-bit signed integer is converted into the # variable, for example: D2X128 as -19, the #25128 will read out of -19.000; if set parameter as 1 floating-point type, D register uses 32-bit floating-point format conversion passed in the # variable and odd-numbered of # variables are forbidden to operate, for example: set the value of D2X128, D2X129 as 19.999, read #25128 out of 19.999 and the #25129 is not available to operate.

| | +0 | +1 | +2 | +3 | +4 | Type |
|---|---|---|---|---|---|---|
| D2X128 | #25128 | #25129note | #25130 | #25131note | #25132 | R/-/- |
| D2X133 | #25133note | #25134 | #25135note | #25136 | #25137note | R/-/- |
| D2X138 | #25138 | #25139note | #25140 | #25141note | #25142 | R/-/- |
| D2X143 | #25143note | #25144 | #25145note | #25146 | #25147note | R/-/- |
| D2X148 | #25148 | #25149note | #25150 | #25151note | #25152 | R/-/- |
| D2X153 | #25153note | #25154 | #25155note | #25156 | #25157note | R/-/- |
| D2X158 | #25158 | #25159note | #25160 | #25161note | #25162 | R/-/- |
| D2X163 | #25163note | #25164 | #25165note | #25166 | #25167note | R/-/- |
| D2X168 | #25168 | #25169note | #25170 | #25171note | #25172 | R/-/- |
| D2X173 | #25173note | #25174 | #25175note | #25176 | #25177note | R/-/- |
| D2X178 | #25178 | #25179note | #25180 | #25181note | #25182 | R/-/- |
| D2X183 | #25183note | #25184 | #25185note | #25186 | #25187note | R/-/- |
| D2X188 | #25188 | #25189note | #25190 | #25191note | #25192 | R/-/- |
| D2X193 | #25193note | #25194 | #25195note | #25196 | #25197note | R/-/- |
| D2X198 | #25198 | #25199note | #25200 | #25201note | #25202 | R/-/- |
| D2X203 | #25203note | #25204 | #25205note | #25206 | #25207note | R/-/- |
| D2X208 | #25208 | #25209note | #25210 | #25211note | #25212 | R/-/- |
| D2X213 | #25213note | #25214 | #25215note | #25216 | #25217note | R/-/- |
| D2X218 | #25218 | #25219note | #25220 | #25221note | #25222 | R/-/- |
| D2X223 | #25223note | #25224 | #25225note | #25226 | #25227note | R/-/- |
| D2X228 | #25228 | #25229note | #25230 | #25231note | #25232 | R/-/- |
| D2X233 | #25233note | #25234 | #25235note | #25236 | #25237note | R/-/- |
| D2X238 | #25238 | #25239note | #25240 | #25241note | #25242 | R/-/- |
| D2X243 | #25243note | #25244 | #25245note | #25246 | #25247note | R/-/- |
| D2X248 | #25248 | #25249note | #25250 | #25251note | #25252 | R/-/- |
| D2X253 | #25253 | #25254 | #25255 | - | - | R/-/- |

Note: parameter **[N1.010 Bit7 MLC variable status]** set as 1 floating-point type, the odd # variable number are prohibited.

B

## B.6.3 # Variable Write to MLC M Relay (#25256~#25383)

| | +0 | +1 | +2 | +3 | +4 | Type |
|---|---|---|---|---|---|---|
| #25256 | M2X256 | M2X257 | M2X258 | M2X259 | M2X260 | R/W/- |
| #25261 | M2X261 | M2X262 | M2X263 | M2X264 | M2X265 | R/W/- |
| #25266 | M2X266 | M2X267 | M2X268 | M2X269 | M2X270 | R/W/- |
| #25271 | M2X271 | M2X272 | M2X273 | M2X274 | M2X275 | R/W/- |
| #25276 | M2X276 | M2X277 | M2X278 | M2X279 | M2X280 | R/W/- |
| #25281 | M2X281 | M2X282 | M2X283 | M2X284 | M2X285 | R/W/- |
| #25286 | M2X286 | M2X287 | M2X288 | M2X289 | M2X290 | R/W/- |
| #25291 | M2X291 | M2X292 | M2X293 | M2X294 | M2X295 | R/W/- |
| #25296 | M2X296 | M2X297 | M2X298 | M2X299 | M2X300 | R/W/- |
| #25301 | M2X301 | M2X302 | M2X303 | M2X304 | M2X305 | R/W/- |
| #25306 | M2X306 | M2X307 | M2X308 | M2X309 | M2X310 | R/W/- |
| #25311 | M2X311 | M2X312 | M2X313 | M2X314 | M2X315 | R/W/- |
| #25316 | M2X316 | M2X317 | M2X318 | M2X319 | M2X320 | R/W/- |
| #25321 | M2X321 | M2X322 | M2X323 | M2X324 | M2X325 | R/W/- |
| #25326 | M2X326 | M2X327 | M2X328 | M2X329 | M2X330 | R/W/- |
| #25331 | M2X331 | M2X332 | M2X333 | M2X334 | M2X335 | R/W/- |
| #25336 | M2X336 | M2X337 | M2X338 | M2X339 | M2X340 | R/W/- |
| #25341 | M2X341 | M2X342 | M2X343 | M2X344 | M2X345 | R/W/- |
| #25346 | M2X346 | M2X347 | M2X348 | M2X349 | M2X350 | R/W/- |
| #25351 | M2X351 | M2X352 | M2X353 | M2X354 | M2X355 | R/W/- |
| #25356 | M2X356 | M2X357 | M2X358 | M2X359 | M2X360 | R/W/- |
| #25361 | M2X361 | M2X362 | M2X363 | M2X364 | M2X365 | R/W/- |
| #25366 | M2X366 | M2X367 | M2X368 | M2X369 | M2X370 | R/W/- |
| #25371 | M2X371 | M2X372 | M2X373 | M2X374 | M2X375 | R/W/- |
| #25376 | M2X376 | M2X377 | M2X378 | M2X379 | M2X380 | R/W/- |
| #25381 | M2X381 | M2X382 | M2X383 | - | - | R/W/- |

## B.6.4 # Variable Write to MLC D Register (#25384~#25511)

This function effected by the parameter **[N1.010 Bit7 MLC variable status]**, when set parameter as 0 single word type, the # variable uses 16-bit signed integer conversion is passed to the D register, for example: set #25384 as -19.999, read D3X128 out of -19; if set parameter as 1 floating-point type, # variable uses 32-bit floating-point format in D register and odd-numbered of # variables are forbidden to operate, for example: set the value of #25328 as 19.999, read D3X128 and D3X129 out of 19.999 and the #25329 is not available to operate.

| | +0 | +1 | +2 | +3 | +4 | Type |
|---|---|---|---|---|---|---|
| D3X128 | #25384 | #25385 note | #25386 | #25387 note | #25388 | R/W/- |
| D3X133 | #25389 note | #25390 | #25391 note | #25392 | #25393 note | R/W/- |
| D3X138 | #25394 | #25395 note | #25396 | #25397 note | #25398 | R/W/- |
| D3X143 | #25399 note | #25400 | #25401 note | #25402 | #25403 note | R/W/- |
| D3X148 | #25404 | #25405 note | #25406 | #25407 note | #25408 | R/W/- |
| D3X153 | #25409 note | #25410 | #25411 note | #25412 | #25413 note | R/W/- |
| D3X158 | #25414 | #25415 note | #25416 | #25417 note | #25418 | R/W/- |
| D3X163 | #25419 note | #25420 | #25421 note | #25422 | #25423 note | R/W/- |
| D3X168 | #25424 | #25425 note | #25426 | #25427 note | #25428 | R/W/- |
| D3X173 | #25429 note | #25430 | #25431 note | #25432 | #25433 note | R/W/- |
| D3X178 | #25434 | #25435 note | #25436 | #25437 note | #25438 | R/W/- |
| D3X183 | #25439 note | #25440 | #25441 note | #25442 | #25443 note | R/W/- |
| D3X188 | #25444 | #25445 note | #25446 | #25447 note | #25448 | R/W/- |
| D3X193 | #25449 note | #25450 | #25451 note | #25452 | #25453 note | R/W/- |
| D3X198 | #25454 | #25455 note | #25456 | #25457 note | #25458 | R/W/- |
| D3X203 | #25459 note | #25460 | #25461 note | #25462 | #25463 note | R/W/- |
| D3X208 | #25464 | #25465 note | #25466 | #25467 note | #25468 | R/W/- |
| D3X213 | #25469 note | #25470 | #25471 note | #25472 | #25473 note | R/W/- |
| D3X218 | #25474 | #25475 note | #25476 | #25477 note | #25478 | R/W/- |
| D3X223 | #25479 note | #25480 | #25481 note | #25482 | #25483 note | R/W/- |
| D3X228 | #25484 | #25485 note | #25486 | #25487 note | #25488 | R/W/- |
| D3X233 | #25489 note | #25490 | #25491 note | #25492 | #25493 note | R/W/- |
| D3X238 | #25494 | #25495 note | #25496 | #25497 note | #25498 | R/W/- |
| D3X243 | #25499 note | #25500 | #25501 note | #25502 | #25503 note | R/W/- |
| D3X248 | #25504 | #25505 note | #25506 | #25507 note | #25508 | R/W/- |
| D3X253 | #25509 note | #25510 | #25511 note | - | - | R/W/- |

Note: parameter **[N1.010 Bit7 MLC variable status]** set as 1 floating-point type, the odd # variable number are prohibited.

B

# B.7 Tool Management

## B.7.1 Tool Length (#26000~#41999)

| | T1 | T2 | T3 | T4 | ~ | T1000 | Type |
|---|---|---|---|---|---|---|---|
| Axis 1 | #26000 | #26001 | #26002 | #26003 | ~ | #26999 | R/W/- |
| Axis 2 | #27000 | #27001 | #27002 | #27003 | ~ | #27999 | R/W/- |
| Axis 3 | #28000 | #28001 | #28002 | #28003 | ~ | #28999 | R/W/- |
| Axis 4 | #29000 | #29001 | #29002 | #29003 | ~ | #29999 | R/W/- |
| Axis 5 | #30000 | #30001 | #30002 | #30003 | ~ | #30999 | R/W/- |
| Axis 6 | #31000 | #31001 | #31002 | #31003 | ~ | #31999 | R/W/- |
| Axis 7 | #32000 | #32001 | #32002 | #32003 | ~ | #32999 | R/W/- |
| Axis 8 | #33000 | #33001 | #33002 | #33003 | ~ | #33999 | R/W/- |
| Axis 9 | #34000 | #34001 | #34002 | #34003 | ~ | #34999 | R/W/- |
| Axis 10 | #35000 | #35001 | #35002 | #35003 | ~ | #35999 | R/W/- |
| Axis 11 | #36000 | #36001 | #36002 | #36003 | ~ | #36999 | R/W/- |
| Axis 12 | #37000 | #37001 | #37002 | #37003 | ~ | #37999 | R/W/- |
| Axis 13 | #38000 | #38001 | #38002 | #38003 | ~ | #38999 | R/W/- |
| Axis 14 | #39000 | #39001 | #39002 | #39003 | ~ | #39999 | R/W/- |
| Axis 15 | #40000 | #40001 | #40002 | #40003 | ~ | #40999 | R/W/- |
| Axis 16 | #41000 | #41001 | #41002 | #41003 | ~ | #41999 | R/W/- |

## B.7.2 Tool Wear (#42000~#57999)

| | T1 | T2 | T3 | T4 | ~ | T1000 | Type |
|---|---|---|---|---|---|---|---|
| Axis 1 | #42000 | #42001 | #42002 | #42003 | ~ | #42999 | R/W/- |
| Axis 2 | #43000 | #43001 | #43002 | #43003 | ~ | #43999 | R/W/- |
| Axis 3 | #44000 | #44001 | #44002 | #44003 | ~ | #44999 | R/W/- |
| Axis 4 | #45000 | #45001 | #45002 | #45003 | ~ | #45999 | R/W/- |
| Axis 5 | #46000 | #46001 | #46002 | #46003 | ~ | #46999 | R/W/- |
| Axis 6 | #47000 | #47001 | #47002 | #47003 | ~ | #47999 | R/W/- |
| Axis 7 | #48000 | #48001 | #48002 | #48003 | ~ | #48999 | R/W/- |
| Axis 8 | #49000 | #49001 | #49002 | #49003 | ~ | #49999 | R/W/- |
| Axis 9 | #50000 | #50001 | #50002 | #50003 | ~ | #50999 | R/W/- |
| Axis 10 | #51000 | #51001 | #51002 | #51003 | ~ | #51999 | R/W/- |
| Axis 11 | #52000 | #52001 | #52002 | #52003 | ~ | #52999 | R/W/- |
| Axis 12 | #53000 | #53001 | #53002 | #53003 | ~ | #53999 | R/W/- |
| Axis 13 | #54000 | #54001 | #54002 | #54003 | ~ | #54999 | R/W/- |
| Axis 14 | #55000 | #55001 | #55002 | #55003 | ~ | #55999 | R/W/- |
| Axis 15 | #56000 | #56001 | #56002 | #56003 | ~ | #56999 | R/W/- |
| Axis 16 | #57000 | #57001 | #57002 | #57003 | ~ | #57999 | R/W/- |

B

## B.7.3 Tool Radius/ Status (#42000~#57999)

| | T1 | T2 | T3 | T4 | ~ | T1000 | Type |
|---|---|---|---|---|---|---|---|
| Tool radius | #58000 | #58001 | #58002 | #58003 | ~ | #58999 | R/W/- |
| Radius wear | #59000 | #59001 | #59002 | #59003 | ~ | #59999 | R/W/- |
| Tool status | #60000 | #60001 | #60002 | #60003 | ~ | #60999 | R/W/- |

## B.7.4 Tool Tolerance (#61000~#76999)

Tool tolerance # variable can be defined in the DOPSoft's tool table and then can be applied to tool wear, end tool detection by macro program.

| | T1 | T2 | T3 | T4 | ~ | T1000 | Type |
|---|---|---|---|---|---|---|---|
| Axis 1 | #61000 | #61001 | #61002 | #61003 | ~ | #61999 | R/W/- |
| Axis 2 | #62000 | #62001 | #62002 | #62003 | ~ | #62999 | R/W/- |
| Axis 3 | #63000 | #63001 | #63002 | #63003 | ~ | #63999 | R/W/- |
| Axis 4 | #64000 | #64001 | #64002 | #64003 | ~ | #64999 | R/W/- |
| Axis 5 | #65000 | #65001 | #65002 | #65003 | ~ | #65999 | R/W/- |
| Axis 6 | #66000 | #66001 | #66002 | #66003 | ~ | #66999 | R/W/- |
| Axis 7 | #67000 | #67001 | #67002 | #67003 | ~ | #67999 | R/W/- |
| Axis 8 | #68000 | #68001 | #68002 | #68003 | ~ | #68999 | R/W/- |
| Axis 9 | #69000 | #69001 | #69002 | #69003 | ~ | #69999 | R/W/- |
| Axis 10 | #70000 | #70001 | #70002 | #70003 | ~ | #70999 | R/W/- |
| Axis 11 | #71000 | #71001 | #71002 | #71003 | ~ | #71999 | R/W/- |
| Axis 12 | #72000 | #72001 | #72002 | #72003 | ~ | #72999 | R/W/- |
| Axis 13 | #73000 | #73001 | #73002 | #73003 | ~ | #73999 | R/W/- |
| Axis 14 | #74000 | #74001 | #74002 | #74003 | ~ | #74999 | R/W/- |
| Axis 15 | #75000 | #75001 | #75002 | #75003 | ~ | #75999 | R/W/- |
| Axis 16 | #76000 | #76001 | #76002 | #76003 | ~ | #76999 | R/W/- |

B

## B.7.5 Tool Lifetime Target (#77000~92999)

These variables are tool lifetime setting for multi head machine.

Lathe and milling machine, please refer to tool lifetime description (#192000~#195999)

| | T1 | T2 | T3 | T4 | ~ | T1000 | Type |
|---|---|---|---|---|---|---|---|
| Axis 1 | #77000 | #77001 | #77002 | #77003 | ~ | #77999 | R/W/- |
| Axis 2 | #78000 | #78001 | #78002 | #78003 | ~ | #78999 | R/W/- |
| Axis 3 | #79000 | #79001 | #79002 | #79003 | ~ | #79999 | R/W/- |
| Axis 4 | #80000 | #80001 | #80002 | #80003 | ~ | #80999 | R/W/- |
| Axis 5 | #81000 | #81001 | #81002 | #81003 | ~ | #81999 | R/W/- |
| Axis 6 | #82000 | #82001 | #82002 | #82003 | ~ | #82999 | R/W/- |
| Axis 7 | #83000 | #83001 | #83002 | #83003 | ~ | #83999 | R/W/- |
| Axis 8 | #84000 | #84001 | #84002 | #84003 | ~ | #84999 | R/W/- |
| Axis 9 | #85000 | #85001 | #85002 | #85003 | ~ | #85999 | R/W/- |
| Axis 10 | #86000 | #86001 | #86002 | #86003 | ~ | #86999 | R/W/- |
| Axis 11 | #87000 | #87001 | #87002 | #87003 | ~ | #87999 | R/W/- |
| Axis 12 | #88000 | #88001 | #88002 | #88003 | ~ | #88999 | R/W/- |
| Axis 13 | #89000 | #89001 | #89002 | #89003 | ~ | #89999 | R/W/- |
| Axis 14 | #90000 | #90001 | #90002 | #90003 | ~ | #90999 | R/W/- |
| Axis 15 | #91000 | #91001 | #91002 | #91003 | ~ | #91999 | R/W/- |
| Axis 16 | #92000 | #92001 | #92002 | #92003 | ~ | #92999 | R/W/- |

## B.7.6 Tool Lifetime Accumulation (#93000~108999)

These variables are tool actual lifetime setting for multi head machine.

Lathe and milling machine, please refer to tool lifetime description (#192000~#195999)

| | T1 | T2 | T3 | T4 | ~ | T1000 | Type |
|---|---|---|---|---|---|---|---|
| Axis 1 | #93000 | #93001 | #93002 | #93003 | ~ | #93999 | R/W/- |
| Axis 2 | #94000 | #94001 | #94002 | #94003 | ~ | #94999 | R/W/- |
| Axis 3 | #95000 | #95001 | #95002 | #95003 | ~ | #95999 | R/W/- |
| Axis 4 | #96000 | #96001 | #96002 | #96003 | ~ | #96999 | R/W/- |
| Axis 5 | #97000 | #97001 | #97002 | #97003 | ~ | #97999 | R/W/- |
| Axis 6 | #98000 | #98001 | #98002 | #98003 | ~ | #98999 | R/W/- |
| Axis 7 | #99000 | #99001 | #99002 | #99003 | ~ | #99999 | R/W/- |
| Axis 8 | #100000 | #100001 | #100002 | #100003 | ~ | #100999 | R/W/- |
| Axis 9 | #101000 | #101001 | #101002 | #101003 | ~ | #101999 | R/W/- |
| Axis 10 | #102000 | #102001 | #102002 | #102003 | ~ | #102999 | R/W/- |
| Axis 11 | #103000 | #103001 | #103002 | #103003 | ~ | #103999 | R/W/- |
| Axis 12 | #104000 | #104001 | #104002 | #104003 | ~ | #104999 | R/W/- |
| Axis 13 | #105000 | #105001 | #105002 | #105003 | ~ | #105999 | R/W/- |
| Axis 14 | #106000 | #106001 | #106002 | #106003 | ~ | #106999 | R/W/- |
| Axis 15 | #107000 | #107001 | #107002 | #107003 | ~ | #107999 | R/W/- |
| Axis 16 | #108000 | #108001 | #108002 | #108003 | ~ | #108999 | R/W/- |

## B.7.7 Tool Lifetime (#192000~195999)

| | T1 | T2 | T3 | T4 | ~ | T1000 | Type |
|---|---|---|---|---|---|---|---|
| Tool target lifetime | #192000 | #192001 | #192002 | #192003 | ~ | #192999 | R/W/- |
| Tool actual use time | #193000 | #193001 | #193002 | #193003 | ~ | #193999 | R/W/- |
| Tool target lifetime count | #194000 | #194001 | #194002 | #194003 | ~ | #194999 | R/W/- |
| Tool actual used count | #195000 | #195001 | #195002 | #195003 | ~ | #195999 | R/W/● |

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 2, Building A, 18-24 Ricketts Road,
Mount Waverley, Victoria 3149 Australia
Mail: IA.au@deltaww.com
TEL: +61-1300-335-823 / +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 Ltd.**
Estrada Velha Rio-São Paulo, 5300 Eugênio de
Melo - São José dos Campos CEP: 12247-004 - 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 2921 987 238

**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

**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