No patent liability is assumed with respect to the use of the information contained herein. Every precaution has been taken in the preparation of this booklet; however OMRON assumes no responsibility for errors or omissions.

CX - Motion

Version 1.0
12th July 1999

Table of Content

Section 1 Introduction of CX-Motion

Section 2 Position Data

Section 3 G - Code Programming

Table of Content

Section 4 Automatic Downloading Function

Section 5 Monitoring Function

Section 6 Startup Guide

Section 1
Introduction to CX-Motion

This section will guide user on the creation of project and setting up of parameter.

1-1 Introduction

System Requirements

To install the CX-Motion, the basic configuration of the System is as shown:

CX-Motion :

OS: Windows 95/98, Window NT 4.0
RAM: 32MB or more
Harddisk: 10MB or more.
Resolution: SVGA (800x600) or larger

The CX-Motion can support the following MC units.

MC Unit	PLC Type
CS1-MC421/221	CS1 Series
CV500-MC421/221	CV Series
C200H-MC221	C Series

Product Type

Product Description	Specification	Model
Cx-Motion	CD-ROM	WS02-MCTC1-E

1-2 Features

Single-port Multi-access Function
A Windows-based Support Software package called CX-Motion can be used on the same computer and through the same port as the CX-Programmer, enabling multiple programming environments on a single computer.

Servo Information Trace Function
Speed reference values, the present speed, and the error counter can be traced with specified starting conditions and a specified sampling period using the MC Support Software. Up to 500 items can be traced, making it easy to adjust the servo system.

Automatic Loading Function
When it is necessary to use more programs or position data than can be stored in the MC Unit, programs or position data stored in an external memory device at the computer where the MC Support Software is installed can be automatically downloaded to the MC Unit's internal memory.

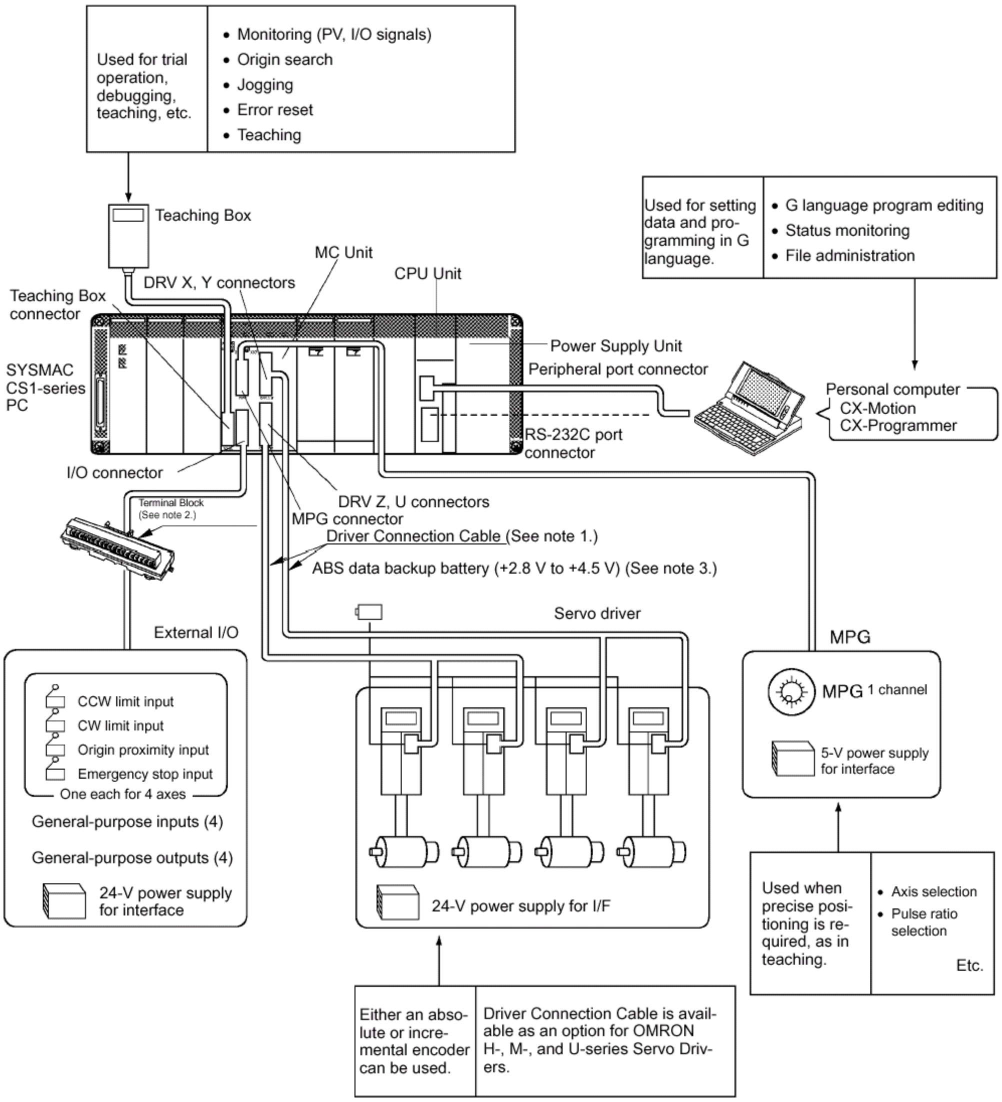
1-3 Functions

Function	Description
Program Editing	Creating, changing, and clearing MC programs.
Position data editing	Creating, changing, and clearing MC programs.
Parameter editing	Creating and changing system parameters, and clearing to defaults.
Transfer and Comparison	Transferring and verifying MC programs, system parameters, and position data contents between MC unit and personal computer.
Printing	Printing MC programs, system parameters and position data contents.

Function	Description
Monitoring	Monitoring MC programs that are being run. Monitoring present values: Reference coordinate system PV (user settings and pulse) Workpiece coordinate system PV Workpiece origin shift amount Error counter value MC unit FAL status MC unit I/O status MC unit error log (CS1W-MC421/MC221 only)
File Management	File display, loading, saving and changing
Use-defined mnemonics	Setting G code by mnemonics
Servo trace	Displaying servo trace data with the MC unit (CS1W-MC421/MC221 only.)
Automatic loading	Automatic downloading of programs to the MC unit (CS1W-MC421/MC2221 only)
File conversion	Converting parameters created by MC Support Software for use by the CS1W-MC421/MC221.

1-4 System Configuration

The following system configuration is applicable to CS1W-MC421 (4 axes). CS1W-MC221 will be used for 2 axes system configuration.


Used for trial operation, debugging, teaching, etc.
• Monitoring (PV, I/O signals)
• Origin search
• Jogging
• Error reset
• Teaching
Teaching Box
MC Unit
CPU Unit
DRV X, Y connectors
Teaching Box connector
SYSMAC CS1-series PC
Used for setting data and programming in G language.
• G language program editing
• Status monitoring
• File administration
Power Supply Unit
Peripheral port connector
Personal computer
CX-Motion
CX-Programmer
RS-232C port connector
I/O connector
Terminal Block (See note 2.)
DRV Z, U connectors
MPG connector
Driver Connection Cable (See note 1.)
ABS data backup battery (+2.8 V to +4.5 V) (See note 3.)
External I/O
Servo driver
MPG
CCW limit input
CW limit input
Origin proximity input
Emergency stop input
One each for 4 axes
General-purpose inputs (4)
General-purpose outputs (4)
24-V power supply for interface
24-V power supply for I/F
MPG 1 channel
5-V power supply for interface
Used when precise positioning is required, as in teaching.
• Axis selection
• Pulse ratio selection
Etc.
Either an absolute or incremental encoder can be used.
Driver Connection Cable is available as an option for OMRON H-, M-, and U-series Servo Drivers.

Note 1. A special Driver Connection Cable is available for OMRON U-, H-, and M-series Servo Drivers. A cable can also be prepared by the user.

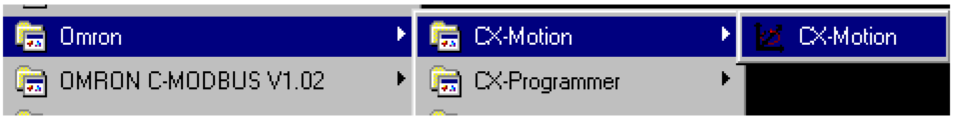
2. A special cable is available for connecting to a Terminal Block. The cable can also be prepared by the user.
3. A data backup is required when using an absolute encoder.

1-5 Creating Project with CX-Motion

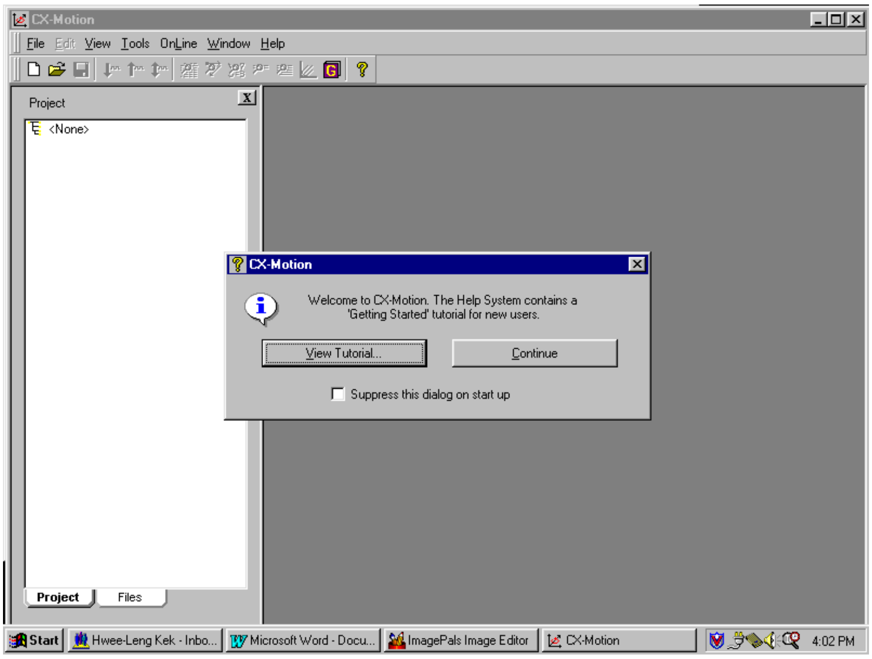
1-5-1 Startup CX-Motion

To start CX-Motion, please perform the following step.

Start - Program - Omron - CX-Motion - Cx-Motion.

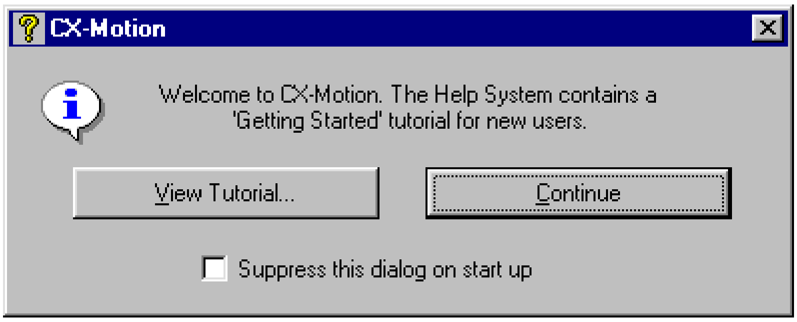


The CX-Motion application window will appear.


CX-Motion
File Edit View Tools OnLine Window Help
Project
<None>
Project Files
Start Hwee-Leng Kek - Inbo... Microsoft Word - Docu... ImagePals Image Editor CX-Motion 4:02 PM
CX-Motion
View Tutorial...
Continue
Suppress this dialog on start up

Welcome to CX-Motion. The Help System contains a 'Getting Started' tutorial for new users.

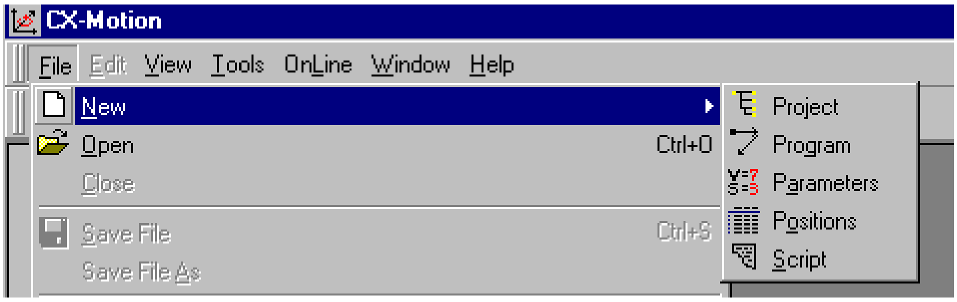
At the following window, click on **Continue** to continue.



Welcome to CX-Motion. The Help System contains a 'Getting Started' tutorial for new users.

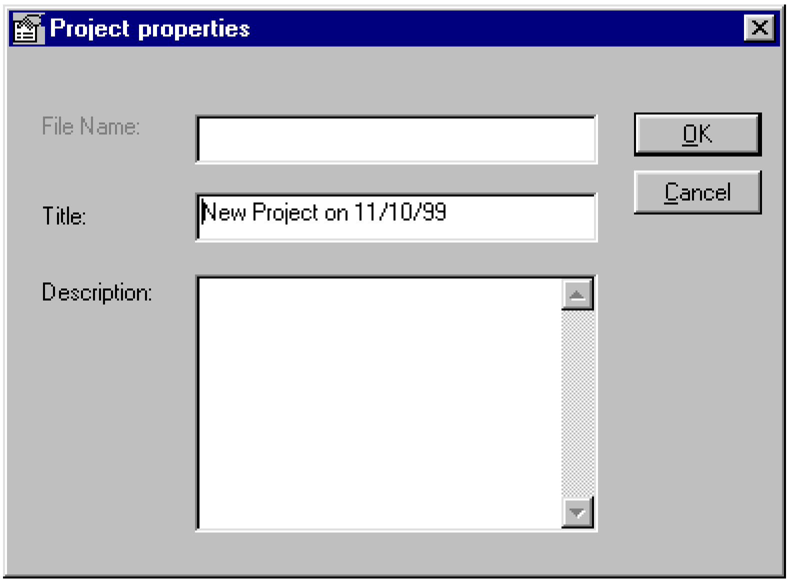
□ Suppress this dialog on start up

From the **File** Menu , select New. At the pop-up menu, select **Project**. (To create a new project.)



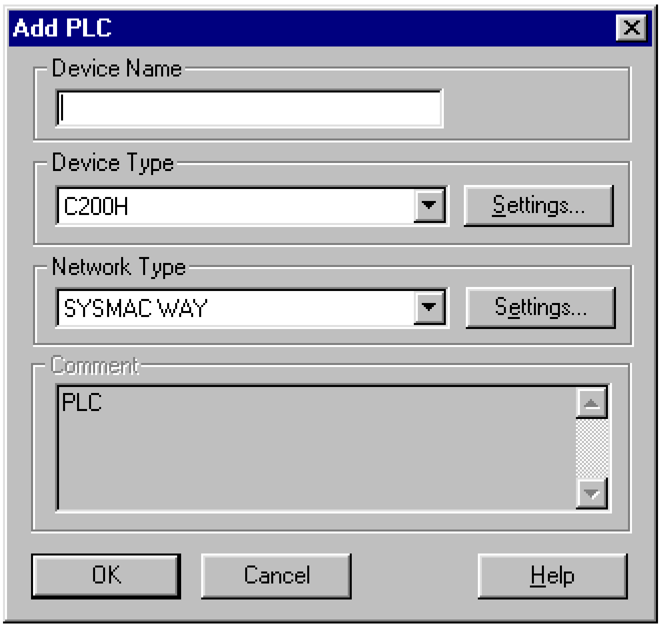
1-4-1 Setting Up a new Project.

Upon the selection, the following dialog box will appear on the screen.



Title: New Project on 11/10/99

Enter the necessary information of the project to be created. Then click on the **OK** button. The following dialog box will appear.



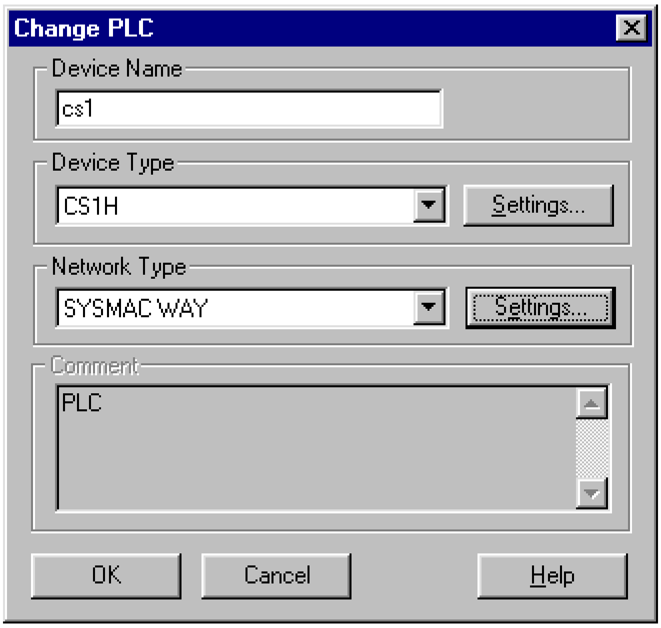
Device Type: C200H

Network Type: SYSMAC WAY

Comment: PLC

Enter a label for the PLC, and provide the necessary information of the PLC. (That is select the PLC type and the communication type.)

When communicating with C200H-MC221, please do the following setting.



Change PLC

Device Name: cs1

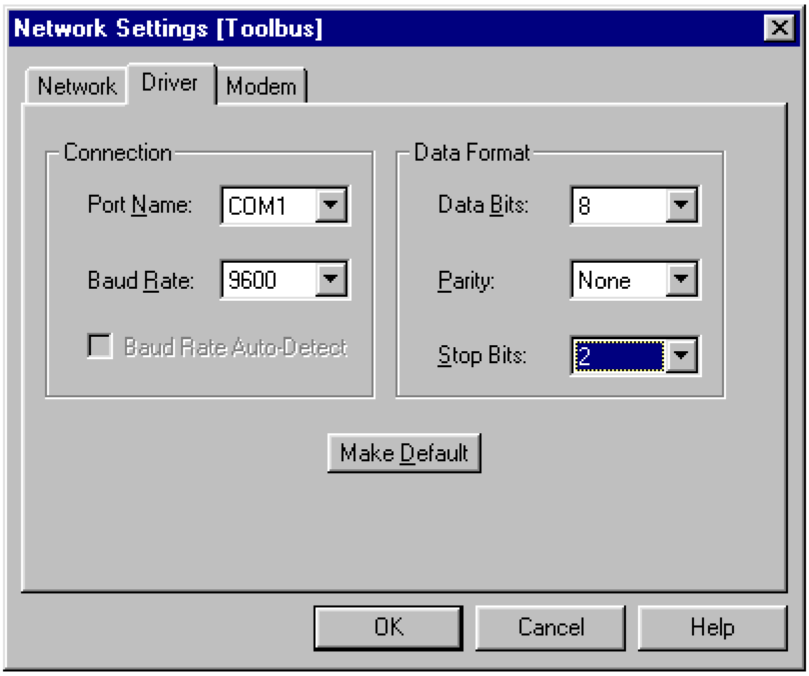
Device Type: CS1H

Network Type: SYSMAC WAY

Comment: PLC

Select **Sysmac Way**, then at the **Driver** tab configure the communication settings as followed:

Communication rate	:	9,600 bps
Parity	:	None
Data Length	:	8 bits
Stop bits	:	2 bits
Response monitor time	:	10 s



Network Settings [Toolbus]

Connection

Port Name: COM1

Baud Rate: 9600

□ Baud Rate Auto-Detect

Data Format

Data Bits: 8

Parity: None

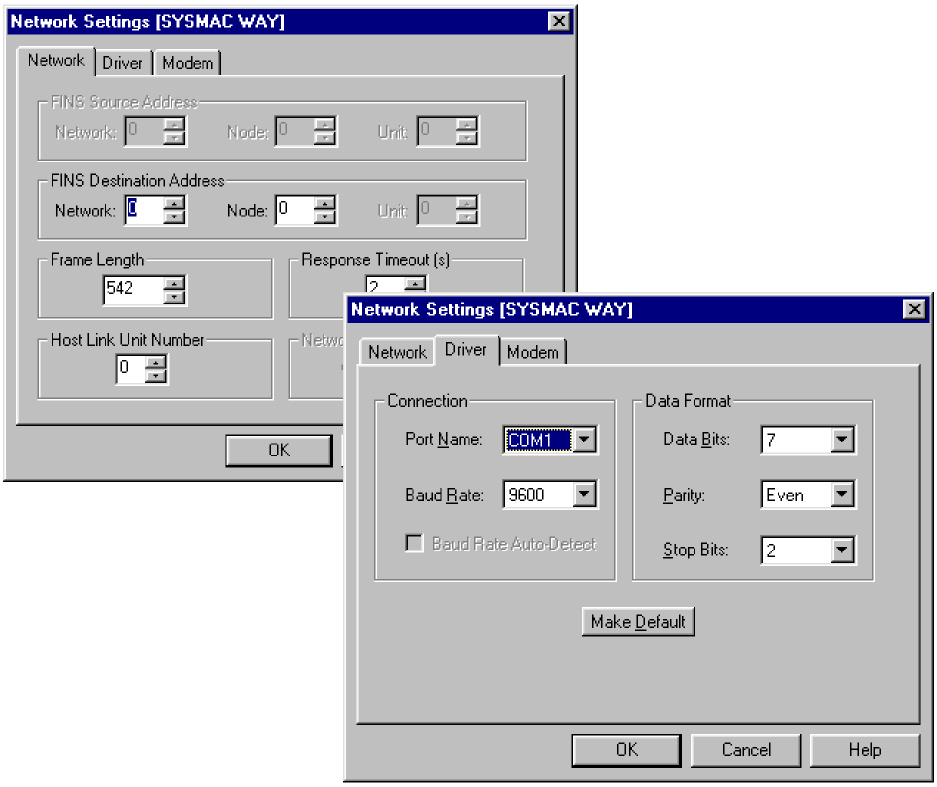
Stop Bits: 2

(The connect for C200H-MC221 will be made via the tool port on the unit, the switch must be slide to HOST.)

As user are able to access to the MC unit on the backplane of CS1W and CV via the I/O bus, uploading and downloading of parameters, position data, programs and etc. can done via the SYSMAC way. That is, connect to the RS232C of the CPU unit.

The configuration are as followed:

• Communication rate	:	9,600 bps
• Parity	:	Even
• Data Length	:	7 bits
• Stop bits	:	2 bits
• Response monitor time	:	2 s



Network Settings [SYSMAC WAY]

Network tab:

FINS Source Address — Network: 0 Node: 0 Unit: 0

FINS Destination Address — Network: 0 Node: 0 Unit: 0

Frame Length: 542

Response Timeout (s): 2

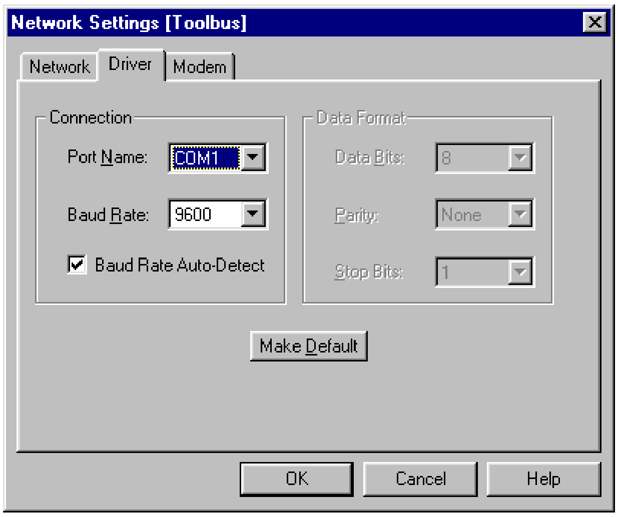
Host Link Unit Number: 0

Driver tab:

Connection — Port Name: COM1; Baud Rate: 9600; □ Baud Rate Auto-Detect

Data Format — Data Bits: 7; Parity: Even; Stop Bits: 2

If **Toolbus** is selected as the Network type for communication, the setting of the driver are as followed.



Network Settings [Toolbus]

Driver tab:

Connection — Port Name: COM1; Baud Rate: 9600; ☑ Baud Rate Auto-Detect

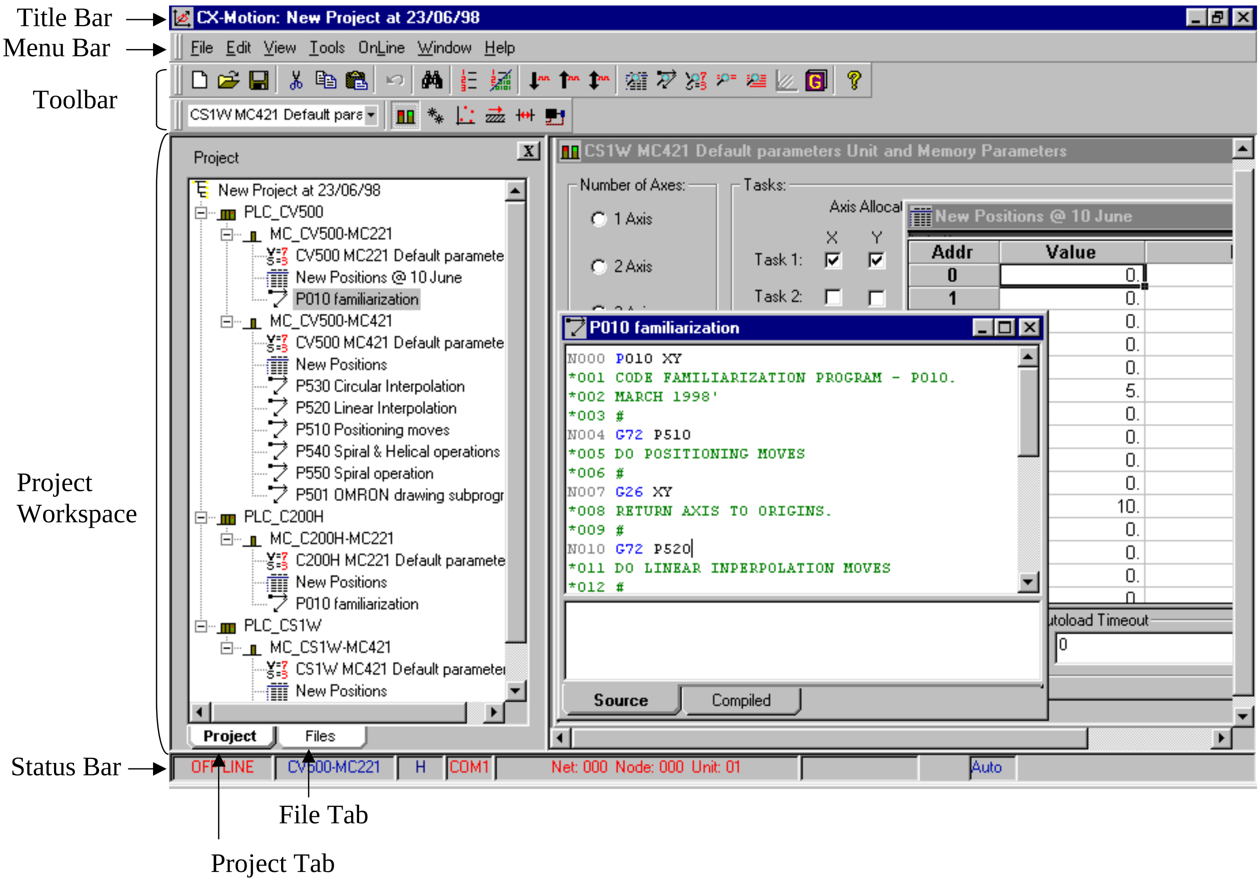
Data Format — Data Bits: 8; Parity: None; Stop Bits: 1

Note 1. The Baud Rate Auto-Detect is only available when using CS series.
2. For detail information on wiring, please refer to the Appendix.

1-5-2 Introduction to CX-Motion workspace

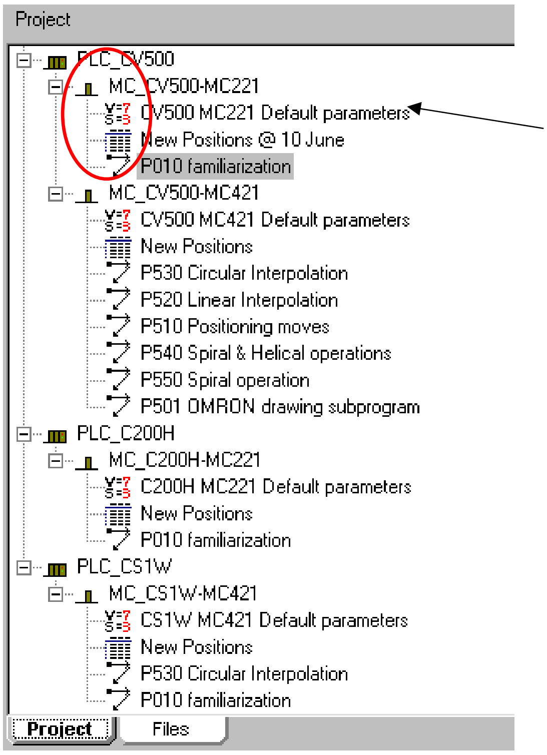
Upon configure the **ADD PLC** dialogue box, the window as shown will appear on screen.

Introduction to CX-Motion window



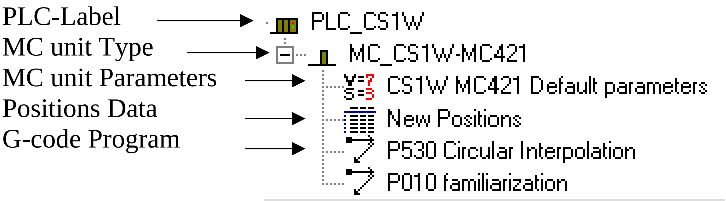
```
N000 P010 XY
*001 CODE FAMILIARIZATION PROGRAM - P010.
*002 MARCH 1998'
*003 #
N004 G72 P510
*005 DO POSITIONING MOVES
*006 #
N007 G26 XY
*008 RETURN AXIS TO ORIGINS.
*009 #
N010 G72 P520
*011 DO LINEAR INPERPOLATION MOVES
*012 #
```

Introduction to Project Workspace

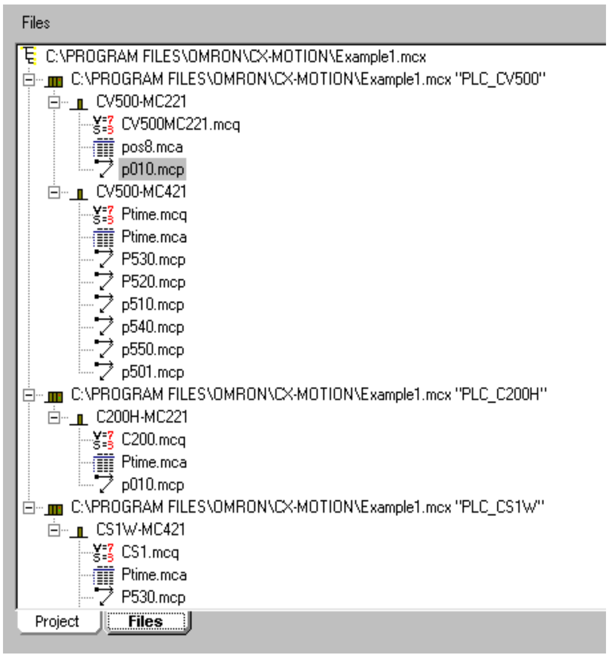


The project workspace consist of two tab, **Project Tab** and **File Tab**. The figure as shown on the left is the **Project Tab**.

Each file types are represented by different icon. In a project file, more than one type of PLC can be configured.
The project tree can be expanded or shrink by click on the "+" with the left mouse click.



Each branch from the PLC-Label will represent the files created for that PLC.

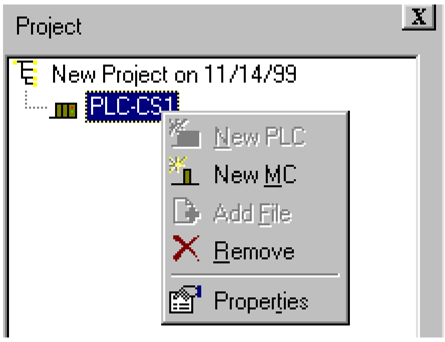


```
Files
C:\PROGRAM FILES\OMRON\CX-MOTION\Example1.mcx
C:\PROGRAM FILES\OMRON\CX-MOTION\Example1.mcx "PLC_CV500"
    CV500-MC221
        CV500MC221.mcq
        pos8.mca
        p010.mcp
    CV500-MC421
        Ptime.mcq
        Ptime.mca
        P530.mcp
        P520.mcp
        p510.mcp
        p540.mcp
        p550.mcp
        p501.mcp
C:\PROGRAM FILES\OMRON\CX-MOTION\Example1.mcx "PLC_C200H"
    C200H-MC221
        C200.mcq
        Ptime.mca
        p010.mcp
C:\PROGRAM FILES\OMRON\CX-MOTION\Example1.mcx "PLC_CS1W"
    CS1W-MC421
        CS1.mcq
        Ptime.mca
        P530.mcp
```

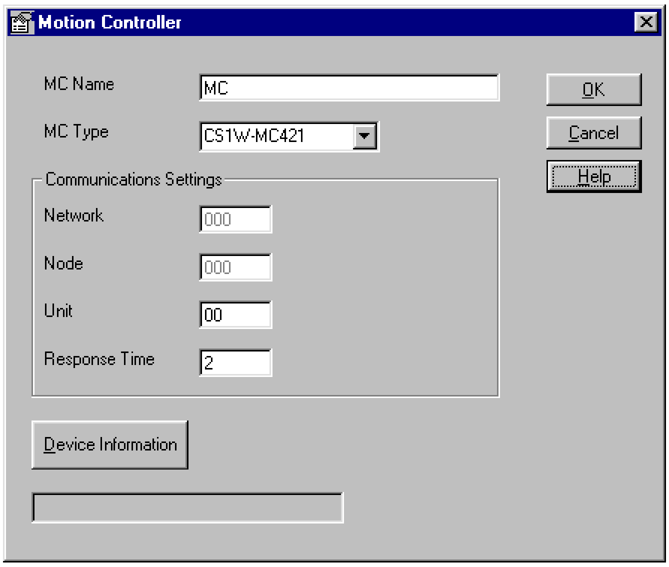
The figure on the left shows the **File Tab** of the project workspace. In the **File Tab**, the filename of the Project file, MC unit parameter, position data file and the G-code program file will be displayed. Hence, the path of the project file will also be displayed in the window.

1-5-3 Inserting MC unit for project

At the **Project Tab**, use the right mouse button to click on the **PLC Type**. The following pop-up dialogue box will appear.



Select **New MC**. The following dialogue box will be shown.



Motion Controller

MC Name: MC

MC Type: CS1W-MC421

Communications Settings

Network: 000

Node: 000

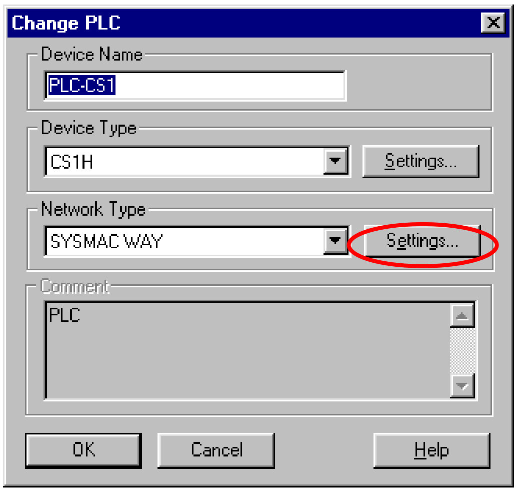
Unit: 00

Response Time: 2

Parameter	**Description**
MC Name	The label for the MC Unit.
MC Type	For CS1 Series , the following type of MC unit can be mounted on the backplane: • CS1W-MC421 • CS1W-MC221 • C200H-MC221 For CV Series, the following type of MC unit can be mounted on the blackplane. • CV500-MC421 • CV500-MC221 For C-Series, only C200H-MC221 is available.
Communications settings:	
Network	The network number of the remote PLC. (See note.)
Node	The node number of the PLC in the remote network. (See note.)
Unit	The MACHINE No. of the MC Unit to communicate to.
Response Time	The timeout response time. (Default 2 seconds.)

Note: CS1 Series and CV Series support remote programming via OMRON Network, example: Ethernet, Controller Link and Sysmac Link . The following are the steps to configure the Network Number and the Node Number.

1,2,3… 1. At the **ADD PLC** or **Change PLC** dialogue box, select **Settings**, as shown.



Change PLC

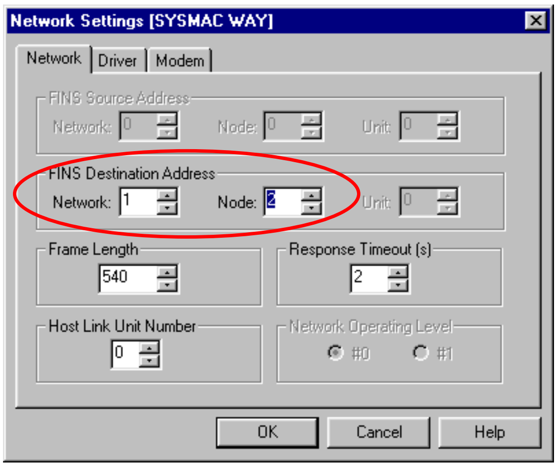
Device Name: PLC-CS1

Device Type: CS1H

Network Type: SYSMAC WAY

Comment: PLC

2. At the **Network Tab**, enter the information of the PLC in the remote network.



Network Settings [SYSMAC WAY]

FINS Source Address — Network: 0 Node: 0 Unit: 0

FINS Destination Address — Network: 1 Node: 2 Unit: 0

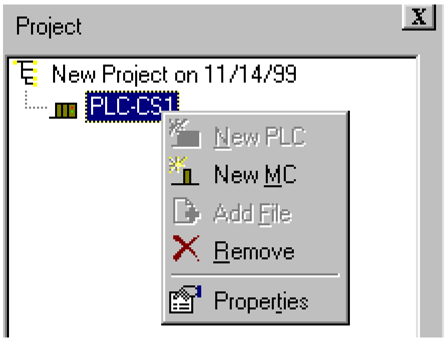
Frame Length: 540

Response Timeout (s): 2

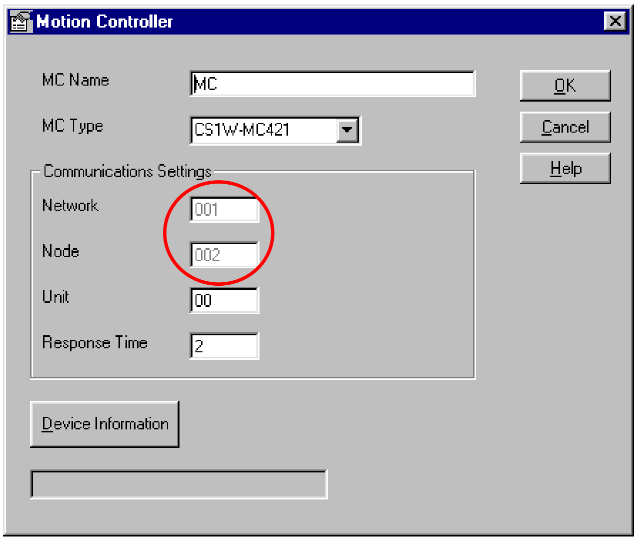
Host Link Unit Number: 0

Network Operating Level: ● #0 ○ #1

3. Upon configuring, click **OK**. Then use the right mouse button to click on the **PLC type**.



4. Select **New MC**, the information of the network will be shown.


Motion Controller

MC Name: MC

MC Type: CS1W-MC421

Communications Settings

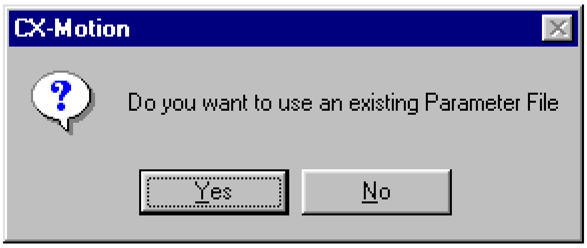
Network: 001

Node: 002

Unit: 00

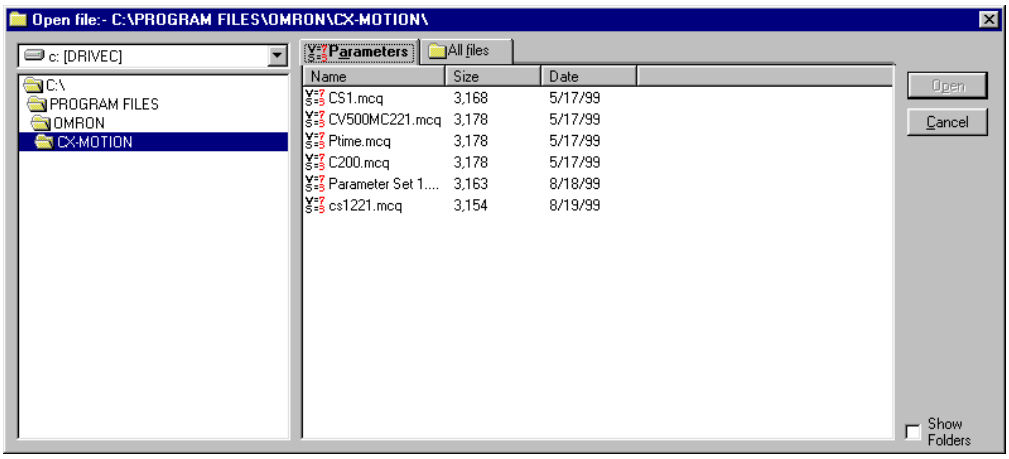
Response Time: 2

Upon completion of entering the parameter in the **Motion Controller** window, click on **OK**. The following dialogue box will be shown.


CX-Motion

Do you want to use an existing Parameter File

If **Yes** is selected. At the window shown, select a file.

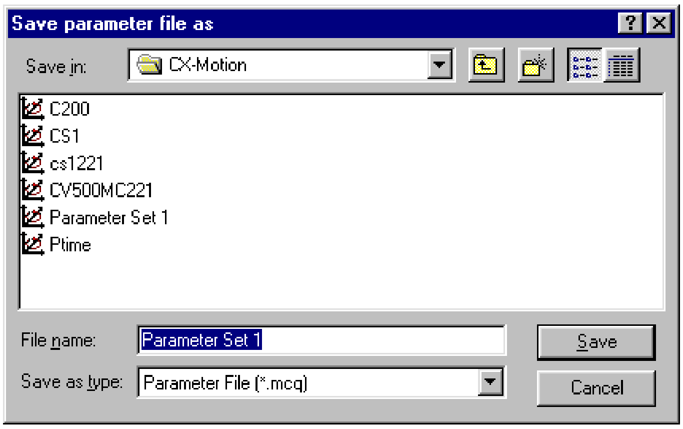

Open file:- C:\PROGRAM FILES\OMRON\CX-MOTION\

c: [DRIVEC]

C:\
PROGRAM FILES
OMRON
CX-MOTION

Name	Size	Date
CS1.mcq	3,168	5/17/99
CV500MC221.mcq	3,178	5/17/99
Ptime.mcq	3,178	5/17/99
C200.mcq	3,178	5/17/99
Parameter Set 1....	3,163	8/18/99
cs1221.mcq	3,154	8/19/99

If **No** is selected, at the window as shown, create a new file.



Save parameter file as

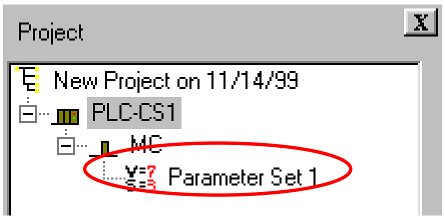
Save in: CX-Motion

- C200
- CS1
- cs1221
- CV500MC221
- Parameter Set 1
- Ptime

File name: Parameter Set 1

Save as type: Parameter File (*.mcq)

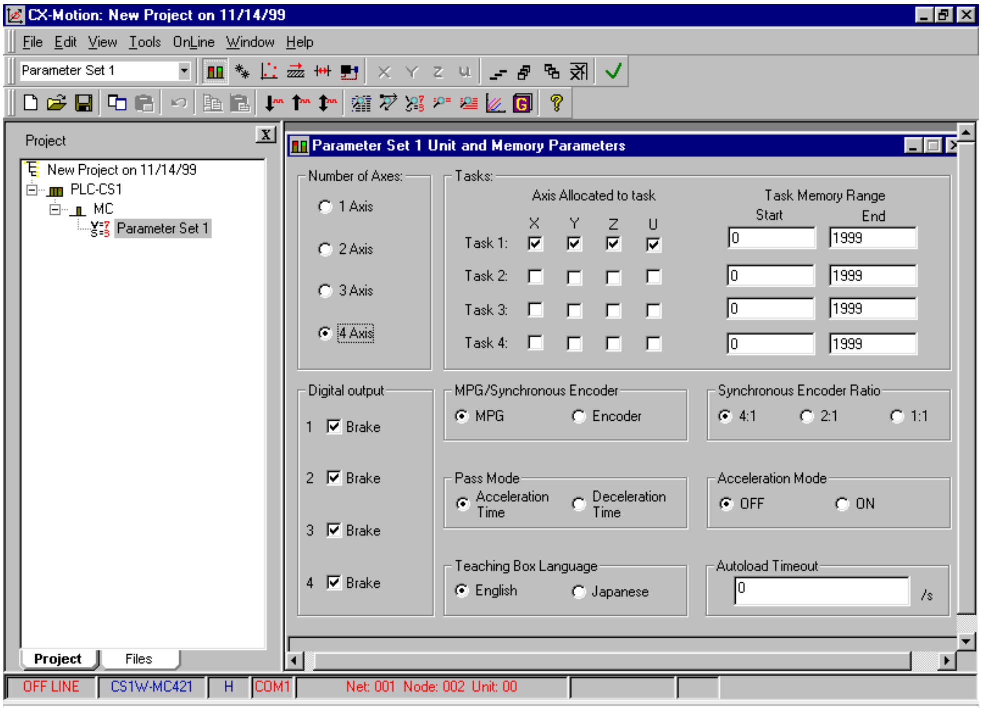
After selecting **Yes** or **No**, the **Parameter Icon** will be shown in the **Project Workspace**.



Project

- New Project on 11/14/99
  - PLC-CS1
    - MC
      - Parameter Set 1

Double click on the Parameter set, at the **Display window**, the parameter setting dialogue box will be shown.



CX-Motion: New Project on 11/14/99

Project

- New Project on 11/14/99
  - PLC-CS1
    - MC
      - Parameter Set 1

Parameter Set 1 Unit and Memory Parameters

Number of Axes: ○ 1 Axis ○ 2 Axis ○ 3 Axis ● 4 Axis

Tasks:

	Axis Allocated to task X	Y	Z	U	Task Memory Range Start	End
Task 1:	☑	☑	☑	☑	0	1999
Task 2:	□	□	□	□	0	1999
Task 3:	□	□	□	□	0	1999
Task 4:	□	□	□	□	0	1999

Digital output
1 ☑ Brake
2 ☑ Brake
3 ☑ Brake
4 ☑ Brake

MPG/Synchronous Encoder: ● MPG ○ Encoder

Synchronous Encoder Ratio: ● 4:1 ○ 2:1 ○ 1:1

Pass Mode: ● Acceleration Time ○ Deceleration Time

Acceleration Mode: ● OFF ○ ON

Teaching Box Language: ● English ○ Japanese

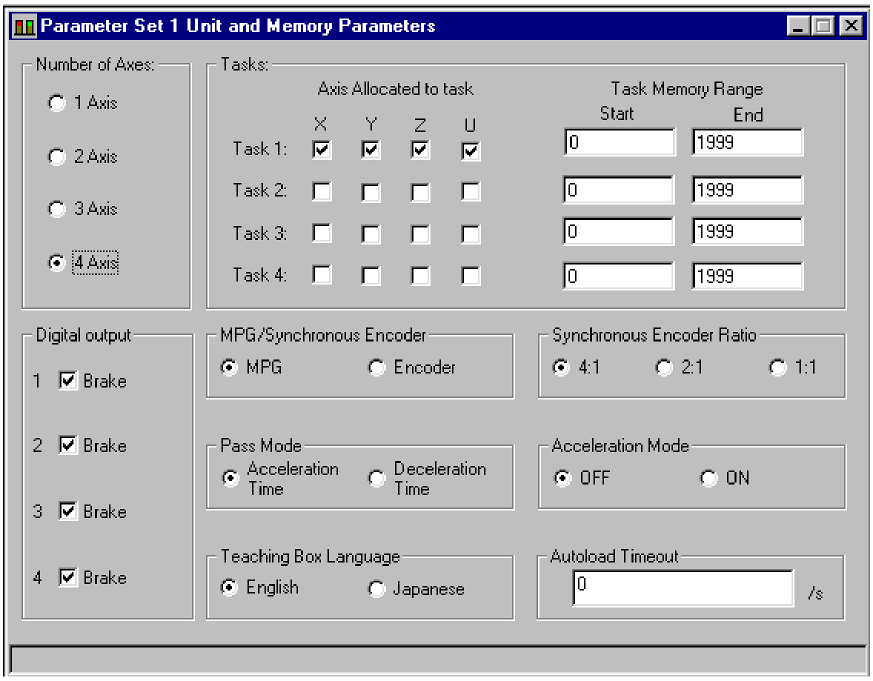
Autoload Timeout: 0 /s

OFF LINE | CS1W-MC421 | H | COM1 | Net: 001 Node: 002 Unit: 00

1-5-4 Setting Parameter for MC unit

Unit and Memory Parameter
Parameter Setting for CS1W-MC421

If CS1W-MC421 is selected for the MC unit in the project, the following parameter window will be shown.



Parameter Set 1 Unit and Memory Parameters

Number of Axes:
○ 1 Axis
○ 2 Axis
○ 3 Axis
● 4 Axis

Tasks:

	Axis Allocated to task X	Y	Z	U	Task Memory Range Start	End
Task 1:	☑	☑	☑	☑	0	1999
Task 2:	□	□	□	□	0	1999
Task 3:	□	□	□	□	0	1999
Task 4:	□	□	□	□	0	1999

Digital output
1 ☑ Brake
2 ☑ Brake
3 ☑ Brake
4 ☑ Brake

MPG/Synchronous Encoder: ● MPG ○ Encoder

Synchronous Encoder Ratio: ● 4:1 ○ 2:1 ○ 1:1

Pass Mode: ● Acceleration Time ○ Deceleration Time

Acceleration Mode: ● OFF ○ ON

Teaching Box Language: ● English ○ Japanese

Autoload Timeout: 0 /s

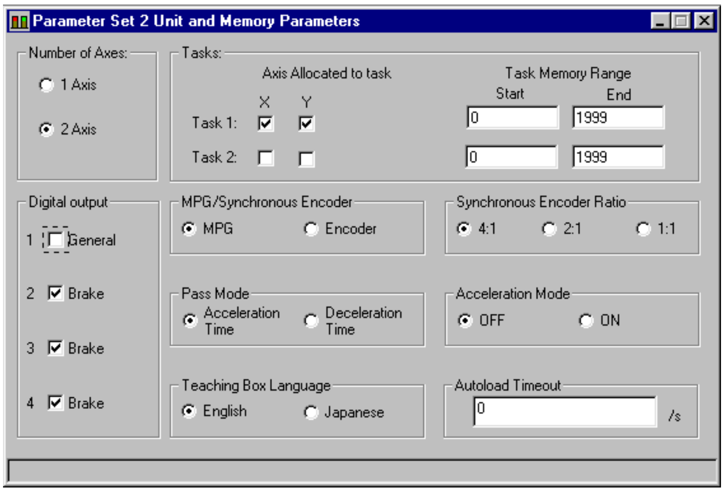
Description of the Parameter setting of the Unit and Memory Window.

Parameter	Description
Number of axis	CS1W-MC421: Max. number of axis 4 CS1W-MC221: Max. number of axis 2 CV500-MC421: Max. number of axis 4 C200H-MC221: Max. number of axis 2
Task Allocated per axis	CS1W-MC421: Max. number of task 4 Max. number of block 2000 Max. number of program 100 CS1W-MC221: Max. number of task 2 Max. number of block 2000 Max. number of program 100 **Note**: Max. no. of block per program 800, including subroutine. Same program number cannot be used for difference tasks. CV500-MC421: Max. number of task 4 Max. number of block 800 Max. number of program 100 CV500-MC221: Max. number of task 2 Max. number of block 800 Max. number of program 100 **Note**: Max. no. of block per program 800, including subroutine. Same program number cannot be used for difference tasks. C200H-MC221: Max. number of task 2 Max. number of block 800 Max. number of program 100

Parameter	Description
Task memory Range	Indication of the block range assign for each task.
Digital Output	Selection of Brake output or general output.
MPG/Synchronous Encoder	Select Line driver output-type, MPG/Sync encode :1
Pass Mode	To specify whether the Pass mode is use at acceleration time or deceleration time.
Teaching Box Language	The type of language display at the teaching box: English or Japanese.
Synchronous Encoder Ratio	Pulse Ratio Selection
Acceleration Mode	To enable or disable the constant acceleration mode.
Autoload Timeout	Max. Time range 0 ~ 180. The parameter is used when batch downloading programs, position data and system parameter to the MC units from external memory device of a personal computer with CX-Motion

Parameter Setting for CS1W-MC221

If CS1W-MC221 is selected for the MC unit in the project, the following parameter window will be shown.



Parameter Set 2 Unit and Memory Parameters

Number of Axes: ○ 1 Axis ● 2 Axis

Tasks:

	Axis Allocated to task X	Y	Task Memory Range Start	End
Task 1:	☑	☑	0	1999
Task 2:	□	□	0	1999

Digital output: 1 □ General; 2 ☑ Brake; 3 ☑ Brake; 4 ☑ Brake

MPG/Synchronous Encoder: ● MPG ○ Encoder

Synchronous Encoder Ratio: ● 4:1 ○ 2:1 ○ 1:1

Pass Mode: ● Acceleration Time ○ Deceleration Time

Acceleration Mode: ● OFF ○ ON

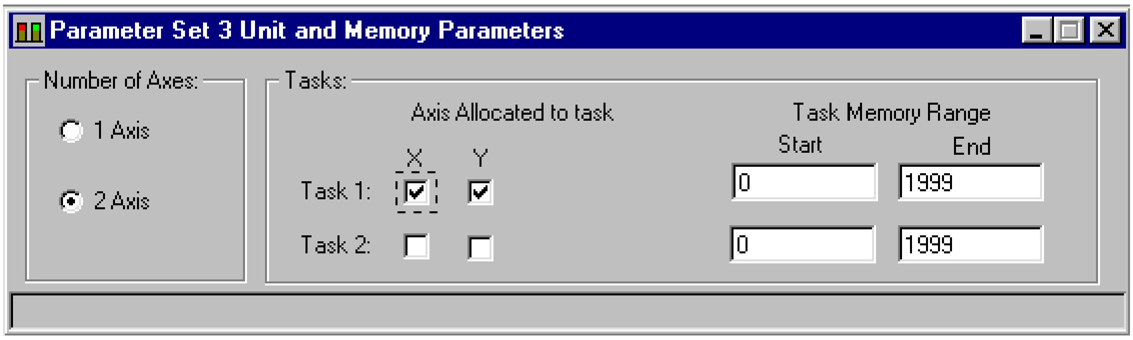
Teaching Box Language: ● English ○ Japanese

Autoload Timeout: 0 /s

The parameter setting of CS1W-MC221 is similar to CS1W-MC421, except that the maximum number of axis available is 2 axis.

Parameter Setting for C200H-MC22 and CV500-MC221

If C200H-MC221 is selected for the MC unit in the project, the following parameter window will be shown.



Parameter Set 3 Unit and Memory Parameters

Number of Axes: ○ 1 Axis ● 2 Axis

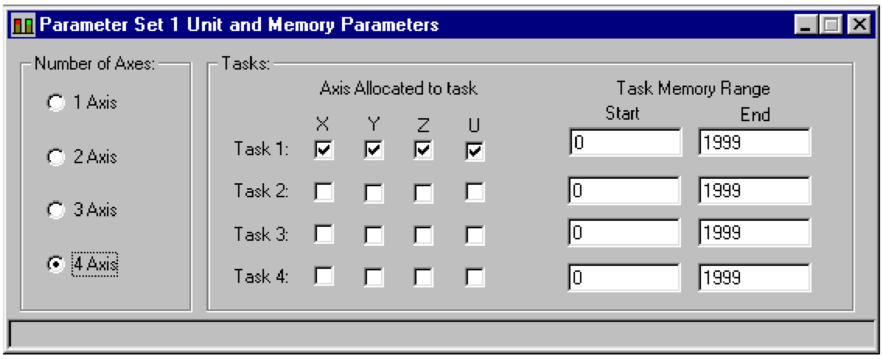
Tasks:

	Axis Allocated to task X	Y	Task Memory Range Start	End
Task 1:	☑	☑	0	1999
Task 2:	□	□	0	1999

Specify the number of axis used, and the block range assign to each task.

Parameter Setting for CV500-MC421

If CV500-MC421 is selected for the MC unit in the project, the following parameter window will be shown.



Parameter Set 1 Unit and Memory Parameters

Number of Axes:

○ 1 Axis
○ 2 Axis
○ 3 Axis
● 4 Axis

Tasks:

	Axis Allocated to task X	Y	Z	U	Task Memory Range Start	End
Task 1:	☑	☑	☑	☑	0	1999
Task 2:	□	□	□	□	0	1999
Task 3:	□	□	□	□	0	1999
Task 4:	□	□	□	□	0	1999

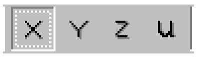
Specify the number of axis used, and the block range assigned for each task.

After configuring the Unit and Memory Parameter setting, it is necessary to set the following parameters for individual axis:

- Machine Parameter
- Coordinate Parameter
- Feed Rate Parameter
- Zone Parameter
- Servo Parameter

To start with, click on at the toolbar for Machine Parameter setting

Note: When any of the parameter setting icon is selected from the toolbar, the following selection icon will also appear.

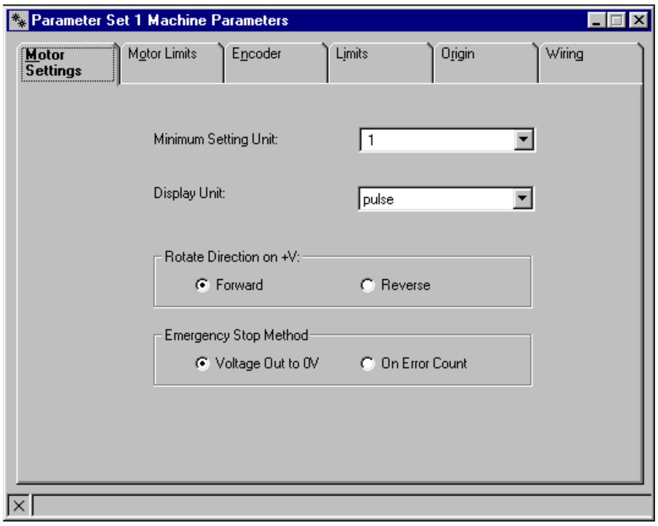


The icon indicates the axis parameters setting. (For example, the figure shown above is X-axis. Hence, all parameters setting, which is being set for X-axis. To set the parameter for Y,Z or U, click on the respective alphabet.) For ***-MC221, only X and Y are selectable.

Machine Parameter

Setting Parameter for CS1W-MC421 and CS1W-MC221

If CS1W-MC421or CS1W-MC221 is selected for the MC unit in the project, the following parameter window will be shown.



Minimum Setting Unit: 1

Display Unit: pulse

Rotate Direction on +V:
● Forward ○ Reverse

Emergency Stop Method
● Voltage Out to 0V ○ On Error Count

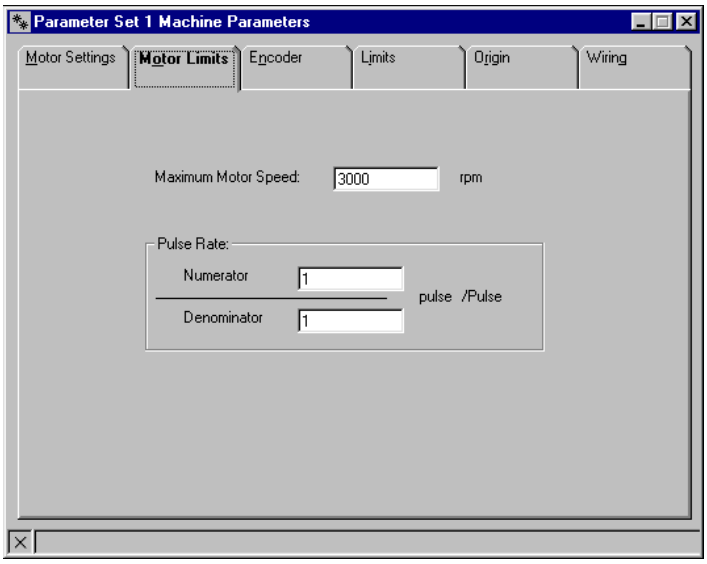
A total of six tabs of forms to be filled up:

- Motor Setting
- Motor Limits
- Encoder
- Limits
- Origin
- Wiring

Motor Setting

Parameter	Description
Minimum setting Unit	1/0.1/0.01/0.001/0,0001 (Default 1)
Display Unit	The following type of display units are available; • mm • inch • deg • pulse
Rotate Direction on +V	The rotating direction when the voltage is positive.
Emergency Stop Method	The parameter will determine how the servomotor will stopped when an emergency stop input signal or CW/CCW limit signal is received. • By dropping the output to 0 v • Accumulated pulses in error counter.

Motor Limits



Maximum Motor Speed: 3000 rpm

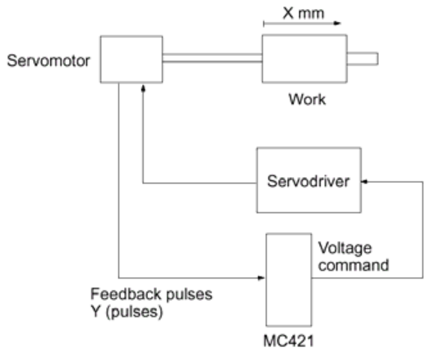
Pulse Rate:

Numerator 1

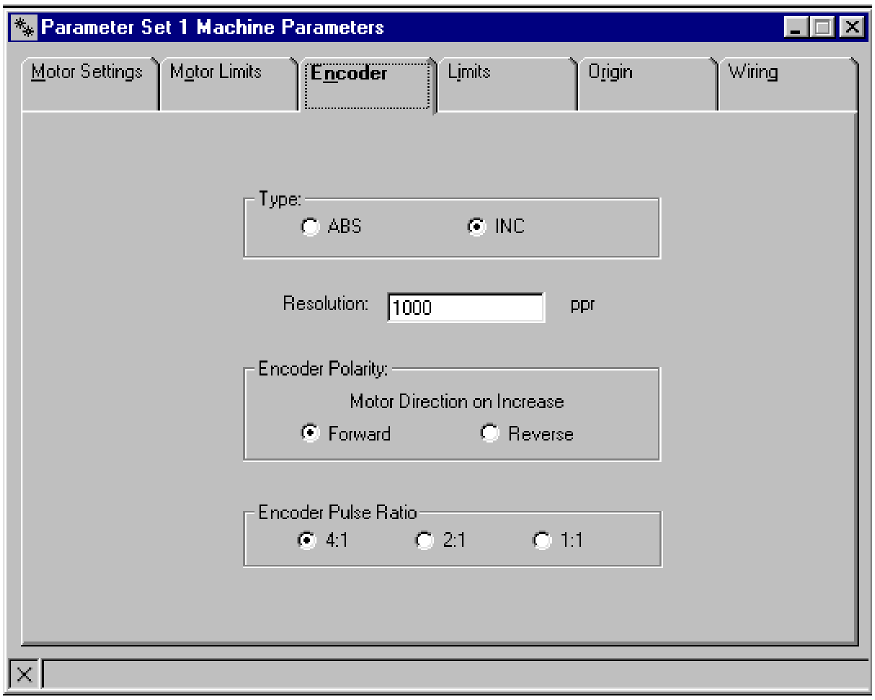
Denominator 1

pulse /Pulse

Parameter	Description
Maximum Motor Speed	Specification of the Motor used. (Speed) Range 1 ~ 32767

Parameter	**Description**
Pulse Rate	The parameter determine the amount that the work-piece is moved per feedback pulse. The pulse rate are given by the following equation. The pulse rate ≤ 1. $\text{Pulse rate} = \frac{X\ (mm)}{Y\ (pulses)\ \times\ 4}$ ← With an encoder ratio of 4  X mm Servomotor Work Servodriver Voltage command Feedback pulses Y (pulses) MC421
Numerator	The value is determine by X (range 1~100000) (X the reading of the ball screw.)
Denominator	The value is determine by Y*4 (range 1~100000) (Y the Encoder Resolution.)

Encoder



Parameter Set 1 Machine Parameters

Type: ○ ABS ● INC

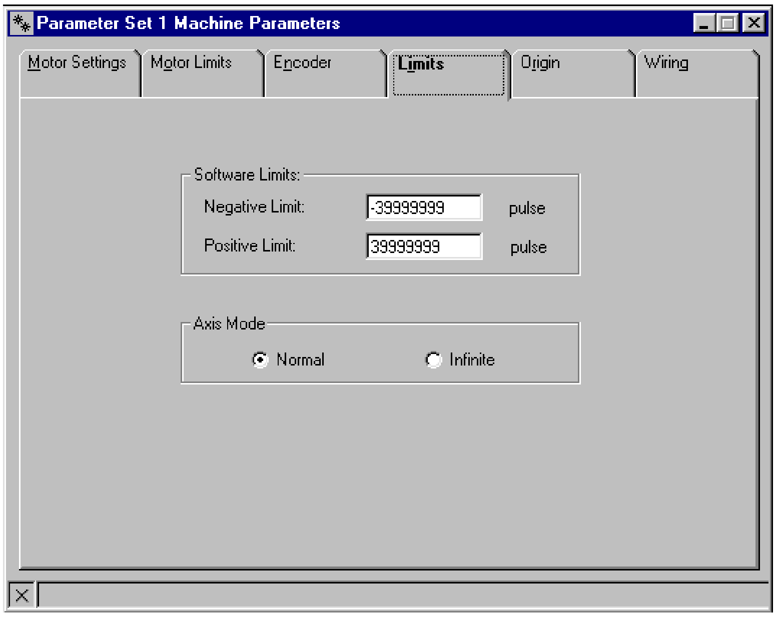
Resolution: 1000 ppr

Encoder Polarity:
Motor Direction on Increase
● Forward ○ Reverse

Encoder Pulse Ratio
● 4:1 ○ 2:1 ○ 1:1

Parameter	**Description**
Type	Type of encoder used : Absolute Encoder or Incremental Encoder
Resolution	Setting the encoder resolution. (Range 1~ 65,535) (The number of pulse that can be output per encoder revolution)
Encoder Polarity	The parameter specified whether the motor will turn forward or reverse when the feedback pulses from the encoder increase.
Encoder Pulse Ratio	Specifies the ratio of the encoder.

Limits


Parameter Set 1 Machine Parameters

Software Limits:

Negative Limit: -39999999 pulse

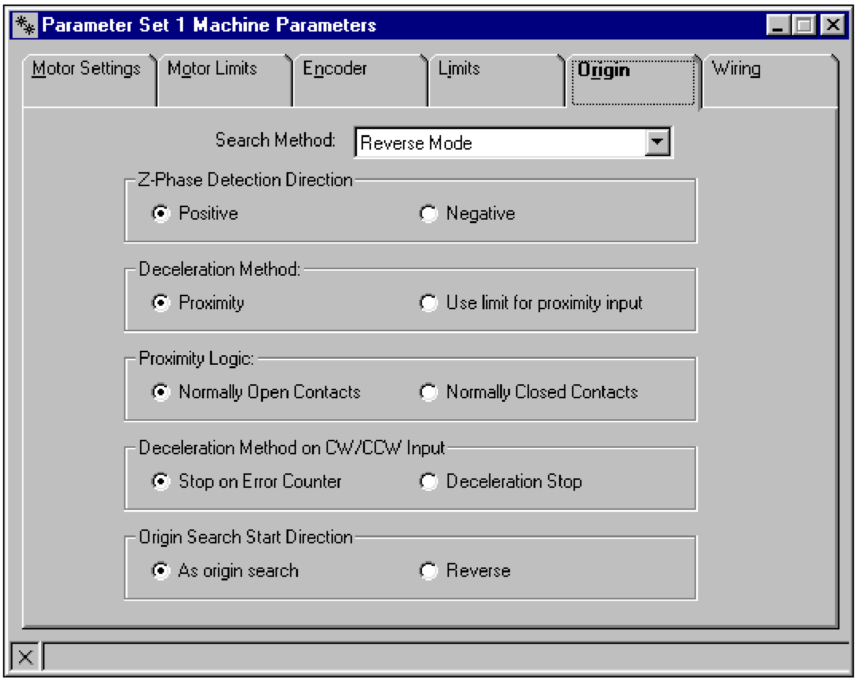
Positive Limit: 39999999 pulse

Axis Mode

● Normal ○ Infinite

Parameter	**Description**
Software Limit	The limit monitor by the software. (The range of the software limit is determined by the positive and negative limit.)
Negative Limit	Minimum setting (-39,999,999)
Position Limit	Maximum setting (39,999,999)
Axis Mode	Two selection are available: • Normal Enables positioning within the software limits after the origin has been established. • Infinite When the Infinite is chosen, the Unlimited Feed Mode is enable, that is the software limits will not operate regardless of whether or not the origin has been determined. Thus, the present position is set by the ring counter method at the value determined by the positive and negative software limits.

Origin



Search Method: Reverse Mode

Z-Phase Detection Direction
● Positive ○ Negative

Deceleration Method:
● Proximity ○ Use limit for proximity input

Proximity Logic:
● Normally Open Contacts ○ Normally Closed Contacts

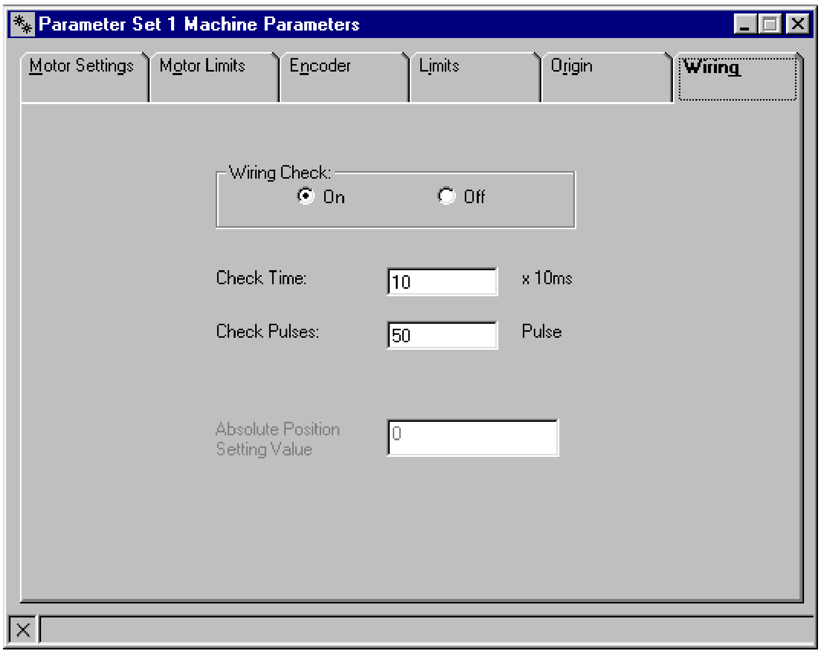
Deceleration Method on CW/CCW Input
● Stop on Error Counter ○ Deceleration Stop

Origin Search Start Direction
● As origin search ○ Reverse

Parameter	Description
Search Method	Origin Search Method. The method available are as followed : • Set origin at Power On. The position of the motor when power is turned ON is automatically defined as the origin. • Reverse Mode The motor search direction is reversed. The origin will be establish when the Z-phase signal goes ON after the origin proximity signal goes from ON to OFF. • Direction search Mode The origin search is performed in the direction set by the origin search direction parameter. An error will occur if a limit input signal goes ON before the origin proximity input signal is received.
Z-phase Detection Direction	Specifies whether to move in the CW or CCW direction to detect Z-phase.
Deceleration Method	Two method are available: Proximity Specifies whether to use origin proximity signal input or limit signal input for deceleration near the origin. Use limit for proximity input. Specifies the limit signal input for deceleration.
Proximity Logic	Specifies whether the origin proximity is normally open or normally closed.
Deceleration Method on CW/CCW input	Two selection: Stop on error counter Uses pulse accumulated in the error counter to stop. Deceleration stop Decelerate to a stop.

Parameter	Description
Origin Search Start Direction	Two selection. As origin search Specifies to start search as in the same direction of phase Z-detection. Reverse Specifies to start search as in the opposite direction of phase Z-detection.

Wiring



Wiring Check: ● On ○ Off

Check Time: 10 x 10ms

Check Pulses: 50 Pulse

Absolute Position Setting Value 0

Parameter	Description
Wiring Check	To enable or disable wiring check function. The wiring check function is designed to check for reverse wiring and disconnected wiring when power supply is turn on in order to prevent motor runaway. The function will output a specified number of pulses in the specified direction and to check whether correct feedback pulses are read.
Check Time	Wiring checking time, the parameter check the feedback pulses if the check pulses are not returned with in the time set, it is consider as failed.(Range 0 to 99)(x10ms)
Check Pulse	Wiring Check Pulse, the parameter specifies the number of pulses output in a wiring check. (Range 0 to 999)
Absolute Position Setting Value	The parameter specifies the correction value for the absolute encoder used when the absolute origin setting is used. (Not available in Teaching Box.)

Setting Parameter for CV500-MC421/MC221 and C200H-MC221

The parameter setting of CV500-MC421,CV500-MC221 and C200H-MC221is about the same except that only two axes are available in CV500-MC221and C200H-MC221.

As similar to the CS1-MC421/221, CV500-MC421/221 and C200H-MC221 had the following tabs for parameters setting :

- Motor Setting
- Motor Limits
- Encoder
- Limits
- Origin
- Wiring

The parameter setting in the Motor Setting Tab and Motor Limits is the same as in CS1W-MC421/221. Thus, the differences will be as shown.

- **Encoder Tab**: The encoder pulse ratio is not available.
- **Limit Tab**: The Axis Mode is not available.
- **Origin Tab**: The Deceleration Method on CW/CCW input and Origin Search Start Direction are not available.
- **Wiring Tab**: For the ABS Encoder, Initial Set Value and Soft Reset is available.

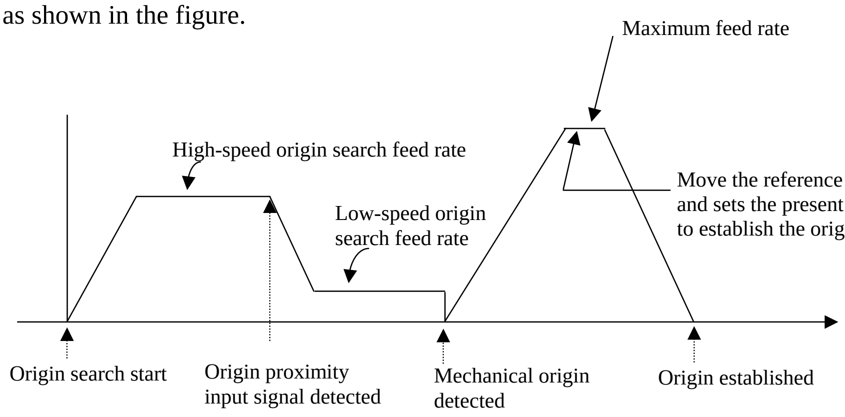
When completion in configuring the Motor Parameter, click on to configure the Coordinate Parameter.

Coordinate Parameter

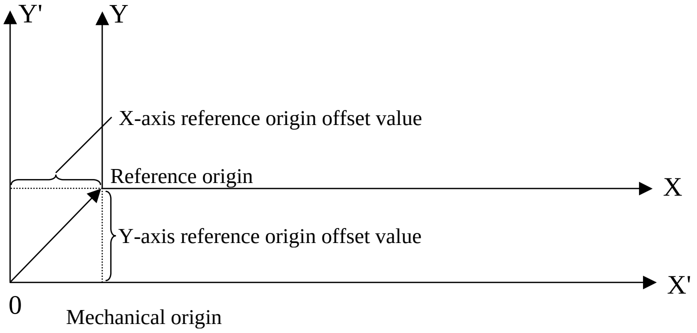
In the Coodinate Parameter window, the Refernce Origin Offset and Workpiece Origin offset is configured.

Reference Origin Offset

A reference origin offset is an offset from the Mechanical origin. It is used in origin searches as shown in the figure.

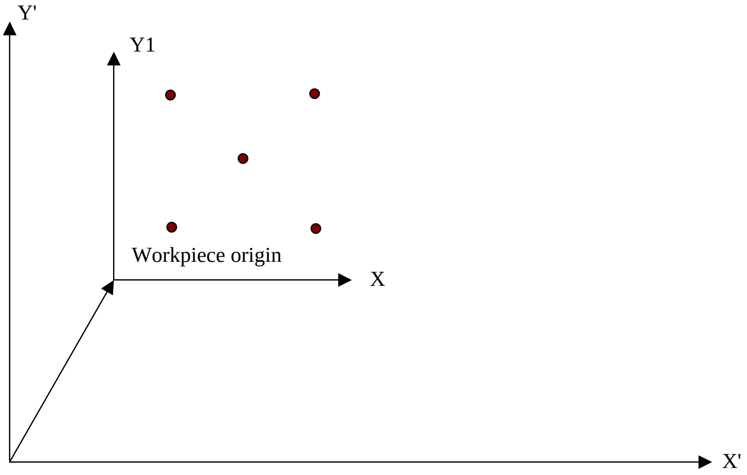

Maximum feed rate
High-speed origin search feed rate
Low-speed origin search feed rate
Move the reference origin offset and sets the present position to 0 to establish the origin
Origin search start
Origin proximity input signal detected
Mechanical origin detected
Origin established

The setting range will vary with the minimum setting unit and pulse rate in the machine parameters.


Y'
Y
X-axis reference origin offset value
Reference origin
X
Y-axis reference origin offset value
X'
0
Mechanical origin

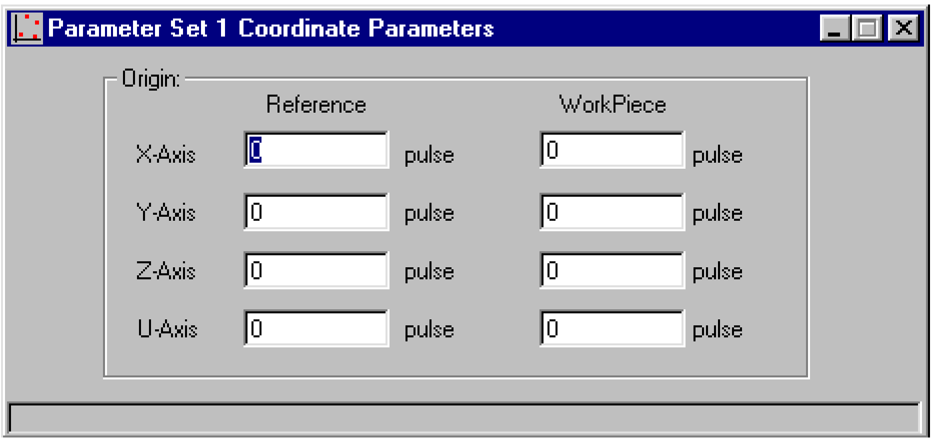
Workpiece Origin Offset Values

A workpiece origin offset is an offset from the reference origin.



It is useful when carrying out operations such as drilling holes in the position as shown above.

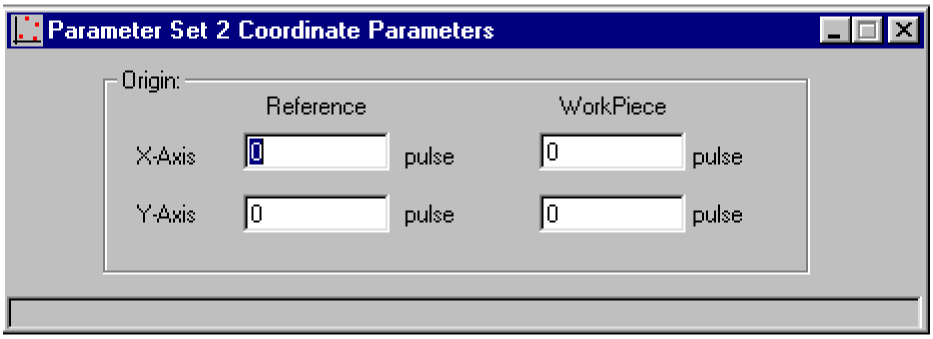
Coordinate Parameter window for 4 axes



Origin:

	Reference		WorkPiece	
X-Axis	0	pulse	0	pulse
Y-Axis	0	pulse	0	pulse
Z-Axis	0	pulse	0	pulse
U-Axis	0	pulse	0	pulse

Coordinate Parameter window for 2 axis.



Origin:

	Reference		WorkPiece	
X-Axis	0	pulse	0	pulse
Y-Axis	0	pulse	0	pulse

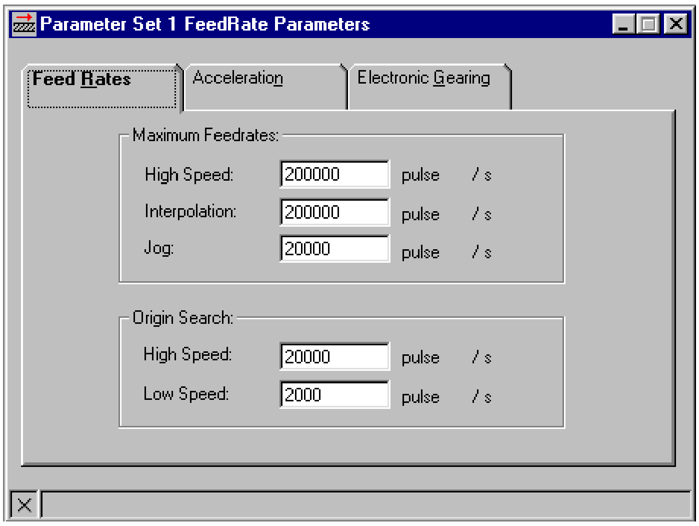
In completion of the Coordinate Parameter, click on ▨ to configure FeedRate parameter.

FeedRate Parameter

In the FeedRate Parameter Window, it consist of three tabs,

- Feed Rates
- Acceleration
- Electronics Gears

Feed Rate CS1W-MC421/221



Maximum Feedrates:
High Speed: 200000 pulse / s
Interpolation: 200000 pulse / s
Jog: 20000 pulse / s

Origin Search:
High Speed: 20000 pulse / s
Low Speed: 2000 pulse / s

The following equation provides possible ranges for high-speed feed rates as well as high and low speed feed rates for origin searches.

$$1 \leq SV \leq Vm \times Ep \times Et \times \frac{Pr}{60}$$

Vm: Max. motor frequency
Ep: Encoder Resolution
Et: Encoder Ratio
Pr: Pulse rate

For example,
Vm: 1000
Ep: 300
Pr: 0.01
Et: ratio of 4

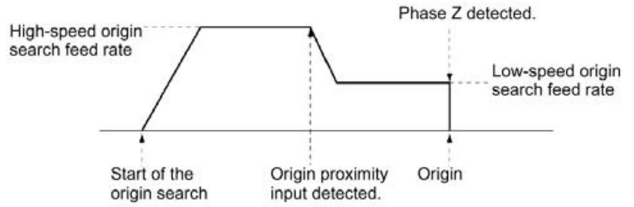
Therefore, the equation yield : 200[mm/s]. Hence the setting range would be 1 to 200. When the minimum setting units is 0.01, the possible setting range would be 0.01 to 200.00. (Units maybe display in terms of the pulse or mm.)

Feed Rate Relationships

The following condition must be met, else error will occur when transferred from CX-Motion.

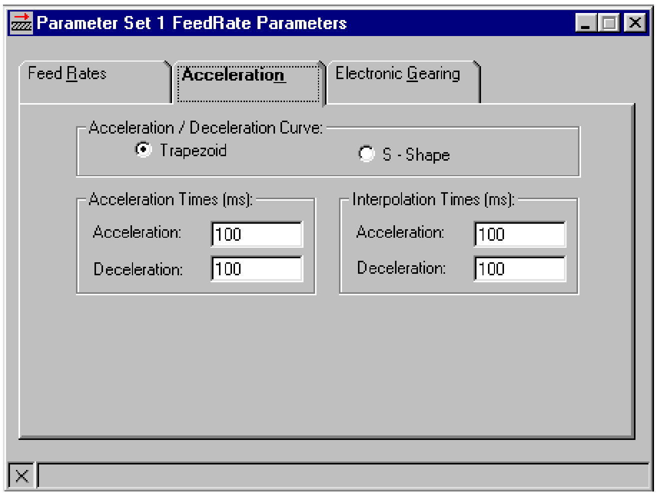
Max. Feed rate ≥ Maximum interpolation feed rate
Max. Feed rate ≥ Maximum jog feed rate
Max. Feed rate ≥ Origin search high-speed feed rate
Max. Feed rate ≥ Origin search low-speed feed rate
Origin search high-speed feed rate ≥ Origin search low-speed feed rate

Parameter	Description
Maximum Feedrate	Setting of feed rate for different operation: • High Speed (Point to Point) • Interpolation (Linear, Circular, Helical circular) • Jog (Jogging rate)

Parameter	**Description**
Origin Search	The high-speed and the low-speed rates for origin search. The high-speed feed rate is the feed rate at which the axis is moved until the origin proximity input signal is detected during an origin search. The origin search low-speed rate is used after the origin proximity input signal is received until phase Z is detected.  Phase Z detected. High-speed origin search feed rate Low-speed origin search feed rate Start of the origin search Origin proximity input detected. Origin

All the parameters in the Feed Rate Tab must be set between 0.0001 to 39,999,999.

Acceleration

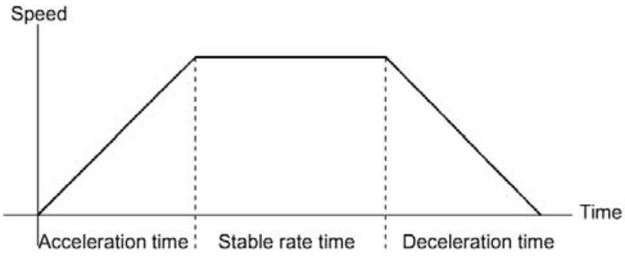
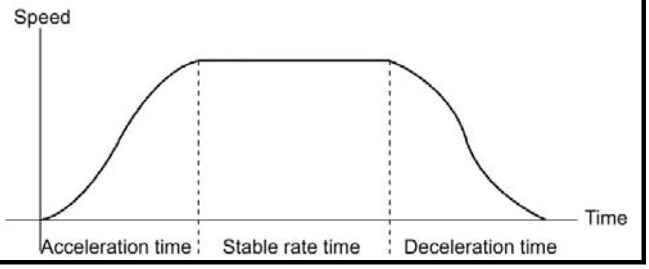


Parameter Set 1 FeedRate Parameters

Acceleration / Deceleration Curve: ● Trapezoid ○ S - Shape

Acceleration Times (ms):
Acceleration: 100
Deceleration: 100

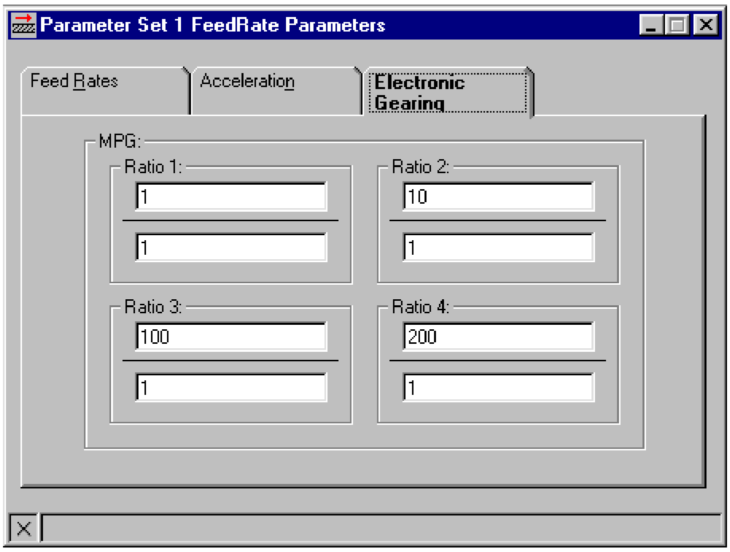
Interpolation Times (ms):
Acceleration: 100
Deceleration: 100

Parameter	**Description**
Acceleration/Deceleration Curve	A trapezoidal curve or a S-curve can be selected. The default setting is for a trapezoidal curve. Trapezoidal Curve  Speed Time Acceleration time Stable rate time Deceleration time S-Curve  Speed Time Acceleration time Stable rate time Deceleration time

Parameter	Description
Acceleration Times (ms) and Interpolation Times (ms)	Acceleration is the time required to reach the maximum feed rate. Deceleration is the time required to decelerate to from maximum speed to zero. The range is from 0 to 10,000 in 2-ms units.

All the parameters in the Acceleration Tab must be set between 0 to100,000

Electronic Gear



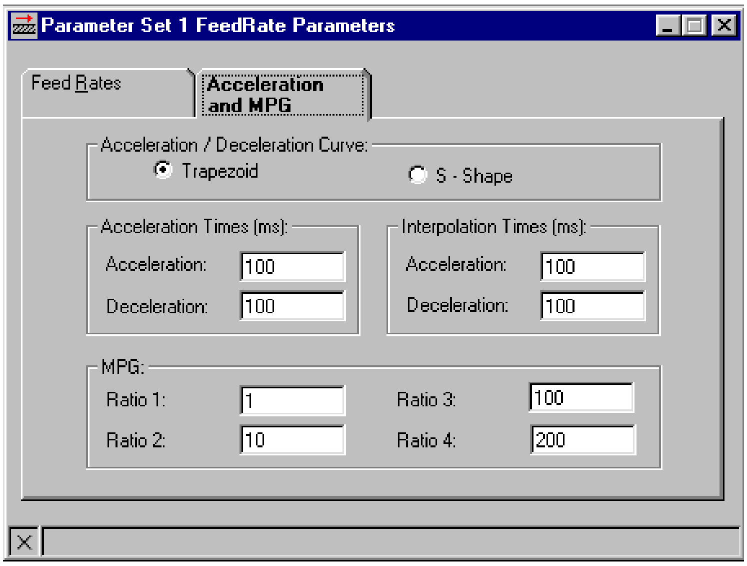
Ratio 1: 1 / 1
Ratio 2: 10 / 1
Ratio 3: 100 / 1
Ratio 4: 200 / 1

Parameter	Description
MPG Ratio	The parameter set the ratio per pulse when using an MPG or Sync encoder. The ratio setting is extremely accurate because the numerator and denominator can be set separately. A maximum of four ratios can set for each axis. Ranges Numerator : -10,000 to 10,000 Denominator : 1 to 10,000 0.0001 ≤ | numerator/denominator | ≤ 10,000

Feed Rate
CV500-MC421/221 & C200H-MC221

The feed rate setting of CV500-MC421/221 and C200H-MC221 is similar as in CS1W-MC421/221.

For CV500-MC421/221 and C200H-MC221, the Electronic Gear Tab is not available. Instead, the MPG Ratio is included in the second tab, which is as followed.



Parameter Set 1 FeedRate Parameters

Feed Rates | **Acceleration and MPG**

Acceleration / Deceleration Curve:
● Trapezoid ○ S - Shape

Acceleration Times (ms):
Acceleration: 100
Deceleration: 100

Interpolation Times (ms):
Acceleration: 100
Deceleration: 100

MPG:
Ratio 1: 1
Ratio 2: 10
Ratio 3: 100
Ratio 4: 200

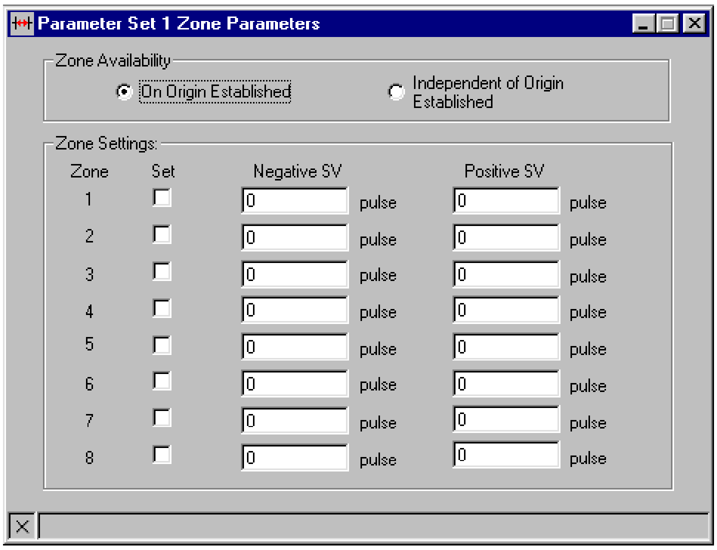
The Acceleration Times and Interpolation Times must be set between 0 to 9999. As for the MPG Ratio, the parameters must be set between 1 to 1000. (Please note, the numerator and denominator is not used.)

To set the Zone Parameter, click on .

Zone Parameter

In the Zone Parameter window, the ranges of positions or values will be defined, and the flags corresponding to the zone will be turned on whenever the present position is within the range.

Zone parameter window for CS1W-MC421/221



Parameter Set 1 Zone Parameters

Zone Availability
● On Origin Established ○ Independent of Origin Established

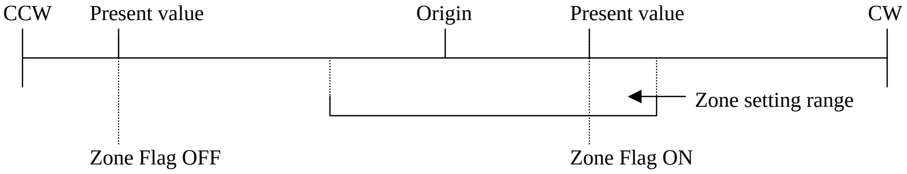
Zone Settings:

Zone	Set	Negative SV		Positive SV	
1	☐	0	pulse	0	pulse
2	☐	0	pulse	0	pulse
3	☐	0	pulse	0	pulse
4	☐	0	pulse	0	pulse
5	☐	0	pulse	0	pulse
6	☐	0	pulse	0	pulse
7	☐	0	pulse	0	pulse
8	☐	0	pulse	0	pulse

Parameter	**Description**
Zone Availability	The validation of when the Zone Setting: On Origin Establish - valid when the origin is establish. Independent of Origin Establish - always valid.
Set	Validation of the zone setting
Negative SV	Setting of the lower range
Positive SV	Setting of the upper range

Positive and Negative SV.

When the present value is within the range set in Positive and Negative SV, the corresponding flags will turn on.

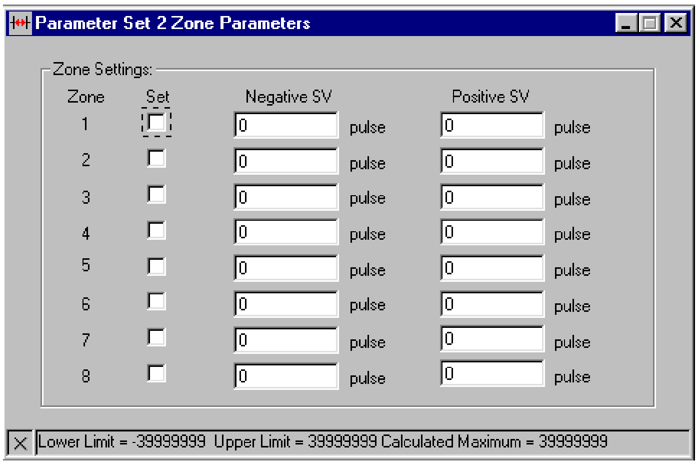


The following table shows the zone flag allocations.

Word		**Axis**	**Bits**	**Bit names**
MC221	**MC421**			
n+22	n+38	X axis	08 to 15	Zone 1 Flag (bit 08) to Zone 8 Flag (Bit 15)
n+25	n+41	Y axis		
	n+44	Z axis		
	n+47	U axis		

The zone ranges that can be set depend on the machine parameter minimum setting unit and pulse rate.

Zone Parameter for CV500-MC421/221 and C200H-MC221



Zone Settings:

Zone	Set	Negative SV		Positive SV	
1	☐	0	pulse	0	pulse
2	☐	0	pulse	0	pulse
3	☐	0	pulse	0	pulse
4	☐	0	pulse	0	pulse
5	☐	0	pulse	0	pulse
6	☐	0	pulse	0	pulse
7	☐	0	pulse	0	pulse
8	☐	0	pulse	0	pulse

Lower Limit = -39999999 Upper Limit = 39999999 Calculated Maximum = 39999999

The parameter setting is the same as in CS1W-MC421/221, except that the Zone availability option is not included.

The following table shows the zone flag area for CV500-MC421/221.

Word		Axis	Bits	Bit names
MC221	MC421			
n+17	n+17	X axis	08 to 15	Zone 1 Flag (bit 08) to Zone 8 Flag (Bit 15)
n+19	n+19	Y axis		
	n+21	Z axis		
	n+23	U axis		

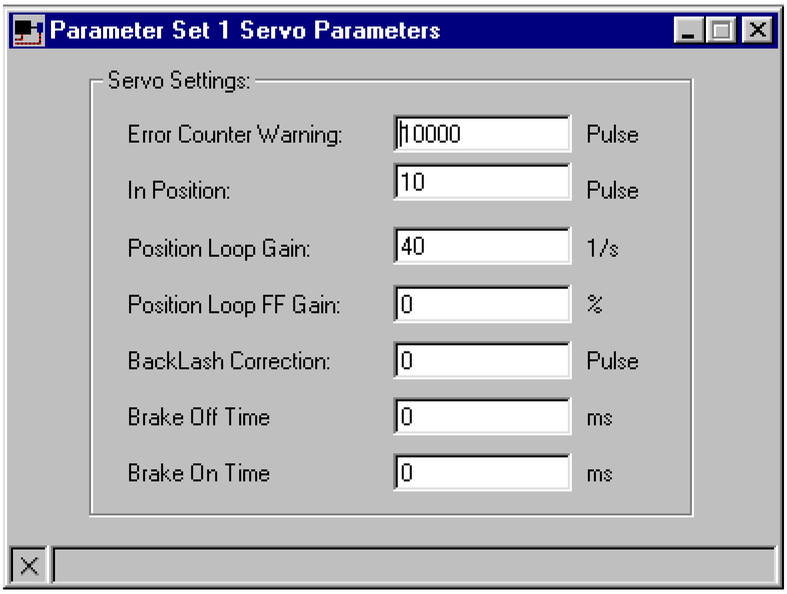
The following table shows the zone flag area for C200H-MC221

Word	Axis	Bits	Bit names
MC221			
n+16	X axis	08 to 15	Zone 1 Flag (bit 08) to Zone 8 Flag (Bit 15)
n+19	Y axis		

To setting the Servo Parameter, click on .

Servo Parameter CS1W-MC421/221

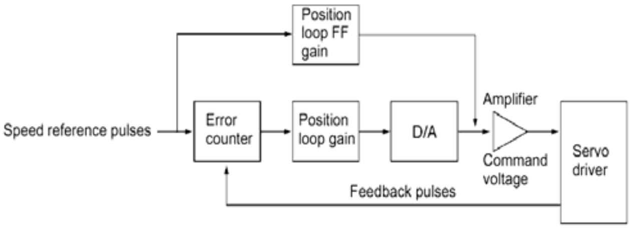
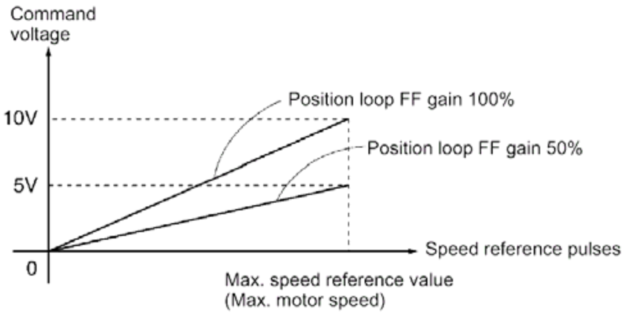
Setting the Servo Parameter, when the parameter is changed, the changed value will become valid in real time.

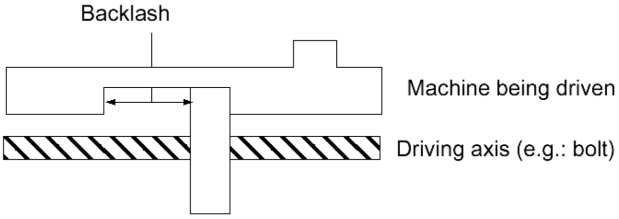


Parameter Set 1 Servo Parameters

Servo Settings:

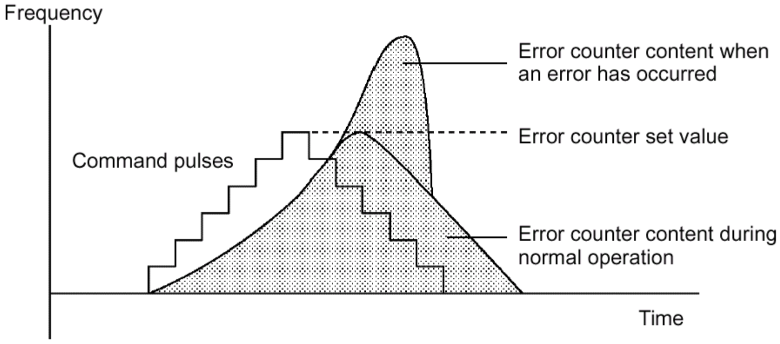
Error Counter Warning:	10000	Pulse
In Position:	10	Pulse
Position Loop Gain:	40	1/s
Position Loop FF Gain:	0	%
BackLash Correction:	0	Pulse
Brake Off Time	0	ms
Brake On Time	0	ms

Parameter	Description
Error Counter Warning	The parameter set the number of pulse used to monitor accumulated pulse in the error counter. (See Note 1.) The range of set value is 0 ~ 65000
In Position	The range within which the system is determined to be at the target position. (See Note 2.) The range of set value is 0~10000
Position Loop Gain	The loop gain can be determined by the following equation: $\text{Position loop gain [1/s]} = \frac{\text{Feed rate [pulses/sec]}}{\text{Accumulated pulses [pulses]}}$ If the position loop gain is too low, the motor responsivity will deteriorate because there will be too many accumulated pulses. If the position loop gain is too high, oscillation and noise might occur. In general, the setting should be : 50 to 70 (1/s) for NC Machine tools 30 to 50 (1/s) for Multi-purpose machines 10 to 30 (1/s) for industrial robots Setting range is 1 to 250.
Position Loop FF Gain	The position loop FF gain process the speed reference pulses and reduces the positioning time by adding directly to the command voltage.  Position loop FF gain Speed reference pulses Error counter Position loop gain D/A Amplifier Command voltage Servo driver Feedback pulses The following diagram shows the relationship between the position loop FF gain and the speed reference pulses 

Parameter	Description
Backlash Correction	The parameter sets backlash correction for the mechanical system. It is a function for pre-registering the backlash space within a range of 0 to 10,000 pulse to minimise the positioning error of the machines moving forward or reverse direction to the same position. A backlash is a sudden backward movement of a driving axis that may be caused due to mechanical looseness as shown.  Backlash Machine being driven Driving axis (e.g.: bolt) The positioning of the machine can be precisely if there is backlash.
Brake Off Time	The delay time before the Positioning Completed Flag turns ON. Setting Range 0 to 10,000ms. (Used when performing Servo Lock.)
Brake On Time	The delay time before the Positioning Completed Flag turns off. Setting Range 0 to 10,000ms. (Used when performing Servo Unlock.)

Note 1. The Error Counter Function Selection (bit 07 to word n) can be used to set either error counter error detection (default setting) or and error counter warnings. If the error detection is set and the number of pulses in the error counter exceeds the warning value, an error counter overflow error will occur and the servo will be turned off. IF the alarm function is set, the Error Counter Warning Flag will be turned on instead.

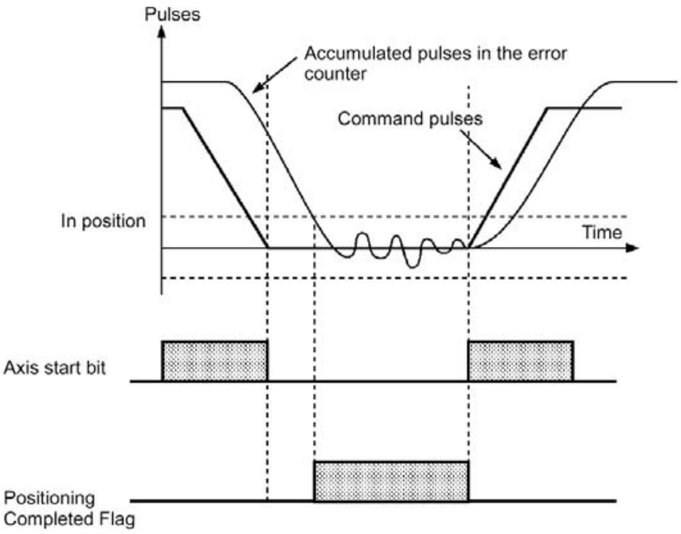
Generally set the error warning counter about 20% above the level of pulses that accumulate during normal operation and adjust to proper setting for the mechanical system. When a problem occurs in the mechanical system or motor, the accumulated pulses in the error counter will exceed the set value so the error can be detected.



The following table shows the Error Counter Warning Flag and Error Counter function selection allocation for CS1W-MC421/221.

Word		Axis	Bits	Bit names
MC221	MC421			
N+22	n+38	X axis	06	Error Counter Warning Flag
N+25	n+41	Y axis		
	n+44	Z axis		
	n+47	U axis		
n		All	07	Error Counter function selection

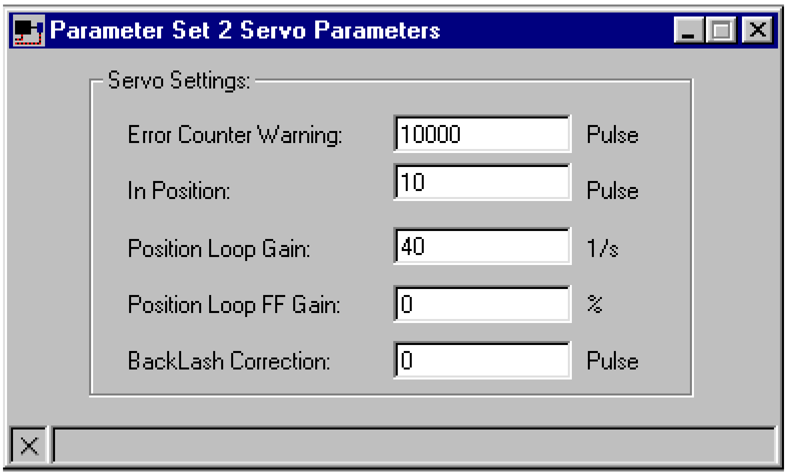
2. When positioning an axis, the Position Completed Flag in the PC interface area is turned ON when the pulse distribution is completed and the axis is in position.


Pulses
Accumulated pulses in the error counter
Command pulses
In position
Time
Axis start bit
Positioning Completed Flag

The following table shows the Axis Start bit and Positioning Completed Flag allocation.

Word		Axis	Bit names
MC221	MC421		
n+22	n+38	X axis	Axis Start Bit (bit 04) Positioning Completed Flag (bit 05)
n+25	n+41	Y axis	
	n+44	Z axis	
	n+47	U axis	

Servo Parameter
CV500-MC421/221 and C200H-MC221


Parameter Set 2 Servo Parameters

Servo Settings:

Error Counter Warning: 10000 Pulse

In Position: 10 Pulse

Position Loop Gain: 40 1/s

Position Loop FF Gain: 0 %

BackLash Correction: 0 Pulse

The configuration of the parameter is similar as in CS1W-MC421/221.

Parameter	Setting Range
Error Counter Warning	0 ~ 65000 (See Note 1)
In position	0 ~ 999 (See Note 2)
Position Loop Gain	5 ~ 150 (Model CV500-MC421/221) 5 ~ 250 (Model C200H-MC221)
Position Loop FF Gain	0 ~ 100
Backlash	0 ~ 999

Note 1. The following table shows the Error Counter Alarm allocation for CV500-MC421/221.

Word		Axis	Bits	Bit names
MC221	MC421			
n+18	n+18	X axis	07	Error Counter Alarm Flag
n+20	n+20	Y axis		
	n+22	Z axis		
	n+24	U axis		

The following table shows the Error Counter Alarm Flag allocation for C200H-MC221.

Word	Axis	Bits	Bit names
MC221			
n+16	X axis	06	Error Counter Alarm Flag
n+19	Y axis		

2. The following table shows the Axis Operating Bit and the Positioning Completed Flag for CV500-MC421/221.

Word		Axis	Bit names
MC221	MC421		
n+18	n+18	X axis	Axis Operating Bit (bit 02) Positioning Completed Flag (bit 03)
n+20	n+20	Y axis	
	n+22	Z axis	
	n+24	U axis	

The following table shows the Axis Operating Bit and the Positioning Completed Flag for C200H-MC221.

Word	Axis	Bit names
MC221		
n+16	X axis	Axis Operating Bit (bit 04) Positioning Completed Flag (bit 05)
n+19	Y axis	

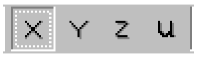
Upon configuring the following parameters for an axis:

- Machine Parameter
- Coordinate Parameter
- Feed Rate Parameter
- Zone Parameter
- Servo Parameter

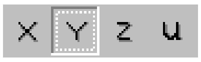
If ✗ is shown at the toolbar, this show that the parameter setting is out of range. Click on the icon and the error will be shown. The error will be shown in red, tips will be shown on the bottom of the parameter window. If ✓ is shown, that means that the parameter setting is all right.

Users are able to copy the parameter settings from one axis to another by the steps are as shown.

To copy the parameter setting from X axis to the other axis(Y, Z or U). Select X (as shown)



Then click on the **Copy Axis** icon . Select thedestination , example Y (as shown) , then click on the **Paste Axis** icon .

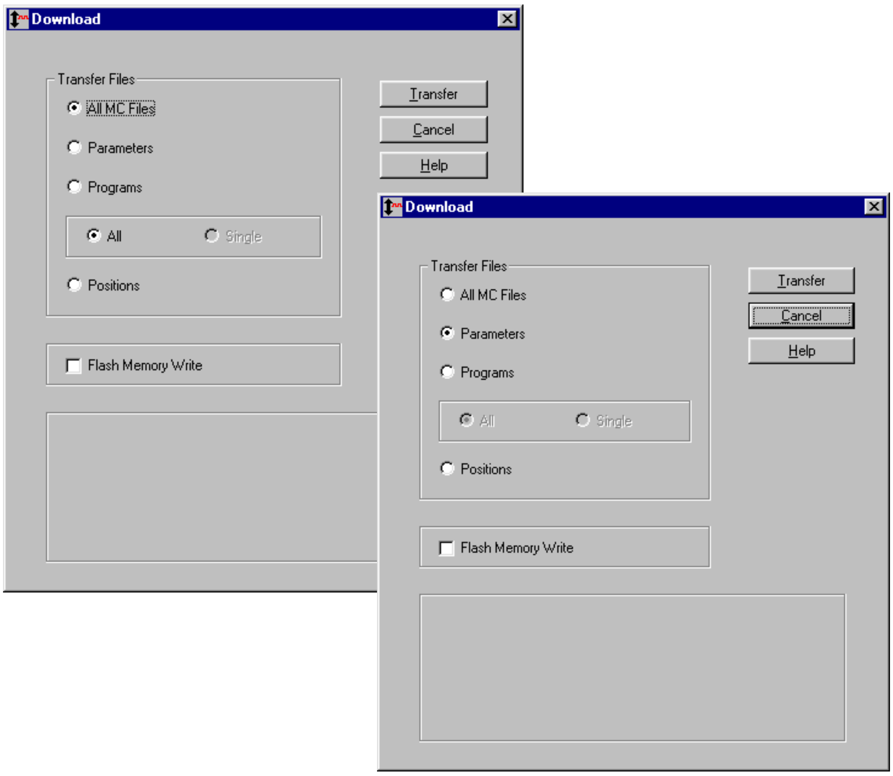


1-6 Downloading and Upload of Parameters

1-6-1 Downloading of Parameters

To download the parameter setting to the MC units, please follow the steps as shown:

Select the MC unit, then click on , the following window will appear.



Download

Transfer Files
- ● All MC Files
- ○ Parameters
- ○ Programs
  - ● All ○ Single
- ○ Positions

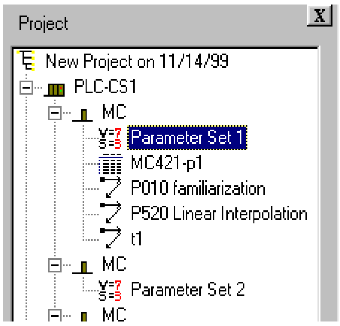
□ Flash Memory Write

Download

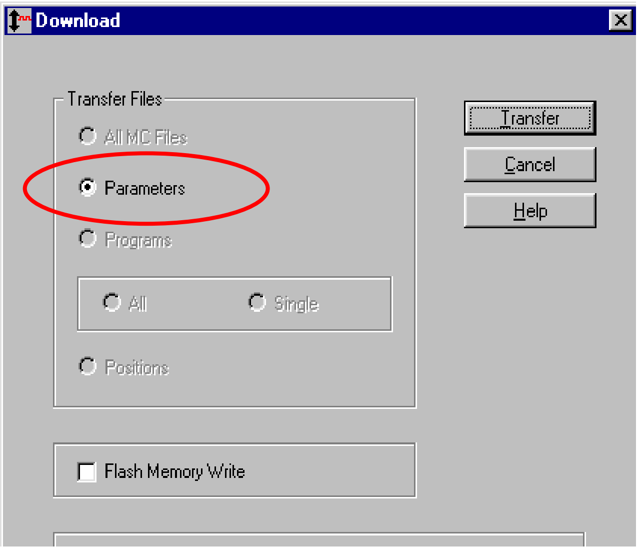
Transfer Files
- ○ All MC Files
- ● Parameters
- ○ Programs
  - ● All ○ Single
- ○ Positions

□ Flash Memory Write

Select **Parameters** to download the parameters Setting only; or at the project tree, select the Parameters setting file, and click . Then, **Download** Dialogue box will appear.



- New Project on 11/14/99
  - PLC-CS1
    - MC
      - Parameter Set 1
      - MC421-p1
      - P010 familiarization
      - P520 Linear Interpolation
      - t1
    - MC
      - Parameter Set 2
    - MC

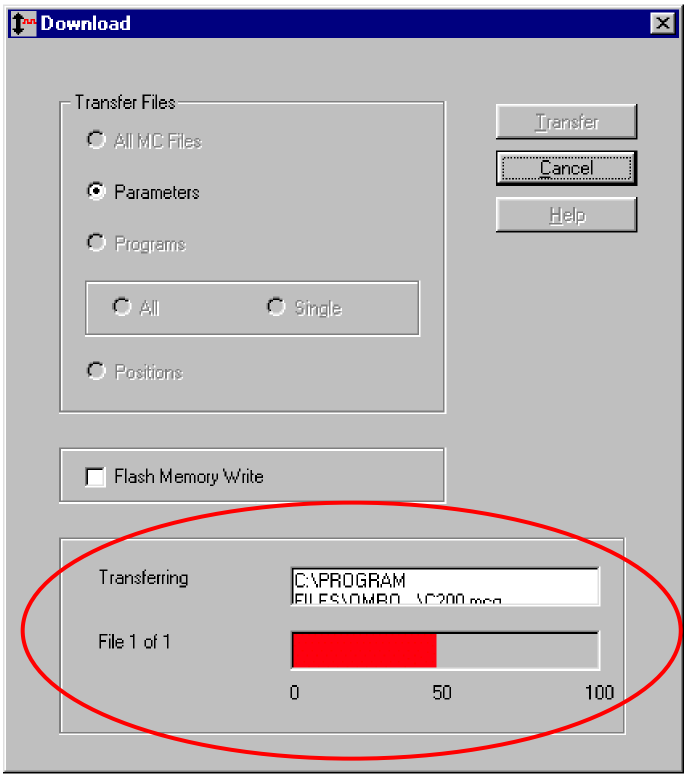


Transfer Files
- ○ All MC Files
- ● Parameters
- ○ Programs
  - ○ All ○ Single
- ○ Positions

□ Flash Memory Write

Note: Only the **Parameters** is selected.

To proceed, click on **Transfer**, to download the Parameters. At the lower portion of the dialogue box, the status of the file transfer will be shown.



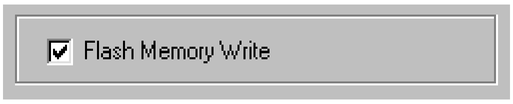
Transfer Files
○ All MC Files
● Parameters
○ Programs
○ All ○ Single
○ Positions

□ Flash Memory Write

Transferring C:\PROGRAM FILES\OMRO...\C200.mcg

File 1 of 1

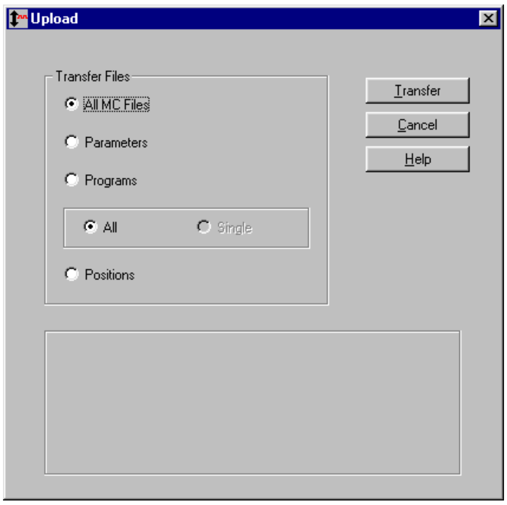
To download the Parameters Settings to the Flash Memory, tick the checkbox of **Flash Memory Write**.


☑ Flash Memory Write

1-6-2 Uploading of Parameters

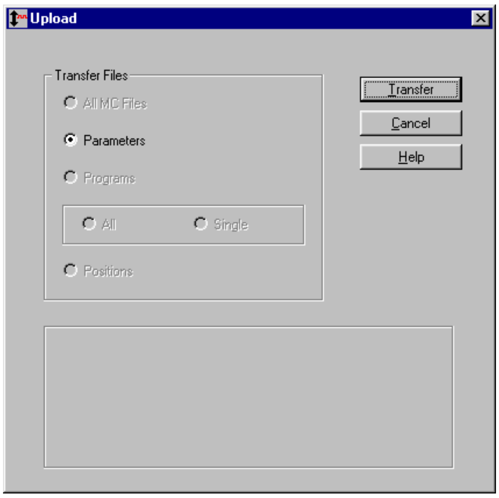
To upload the parameters setting from the MC unit, please proceed as followed.

Select the MC unit to upload, click on , the following window will appear.

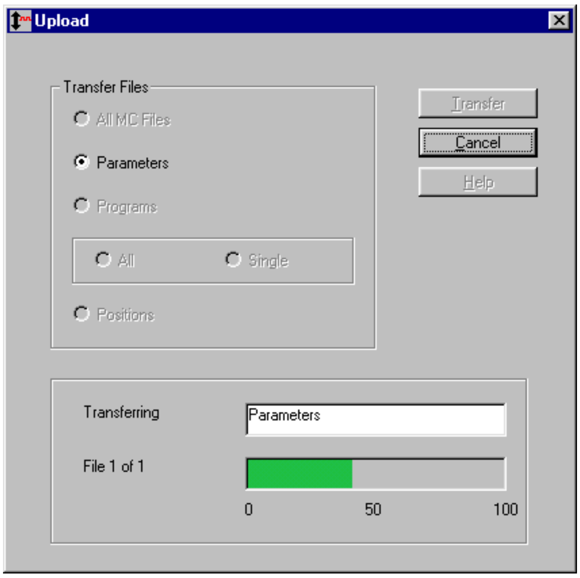


Transfer Files
● All MC Files
○ Parameters
○ Programs
● All ○ Single
○ Positions

Click on **Parameters**.



Then click on **Transfer**. At the lower portion of the Upload window, the file transfer status will be shown.



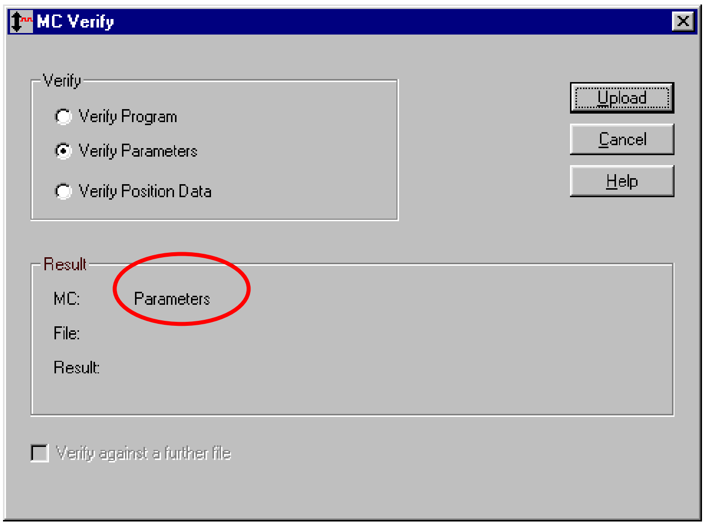
Transferring Parameters

File 1 of 1

Upon completion, click on cancel to proceed. The parameters settings will be shown on screen.

1-6-3 Verify Parameters Settings

To verify the parameters settings of the MC unit, please proceed with the following steps. Select the MC unit, and click on . At the **MC Verify** window, select **Verify Parameters**.

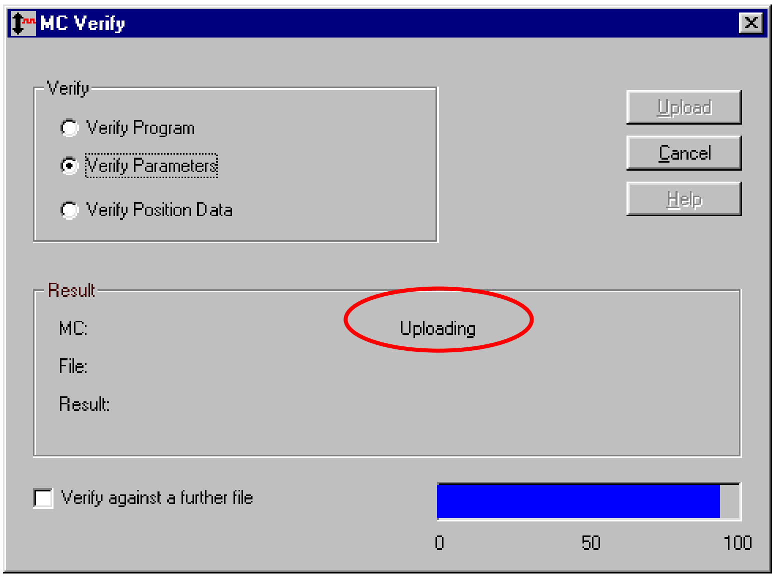


Verify
○ Verify Program
● Verify Parameters
○ Verify Position Data

Result
MC: Parameters
File:
Result:

□ Verify against a further file

Click on **Upload** to proceed, at the **Result** Section , the file transfer status will be shown.

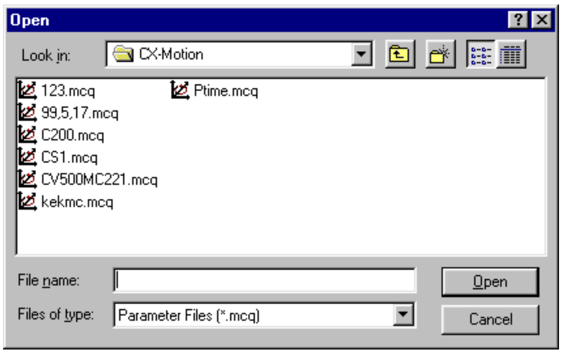


Verify
○ Verify Program
● Verify Parameters
○ Verify Position Data

Result
MC: Uploading
File:
Result:

□ Verify against a further file

Upon uploading of the parameters, the following window will appear, select the file to verify with, and click **Open**.

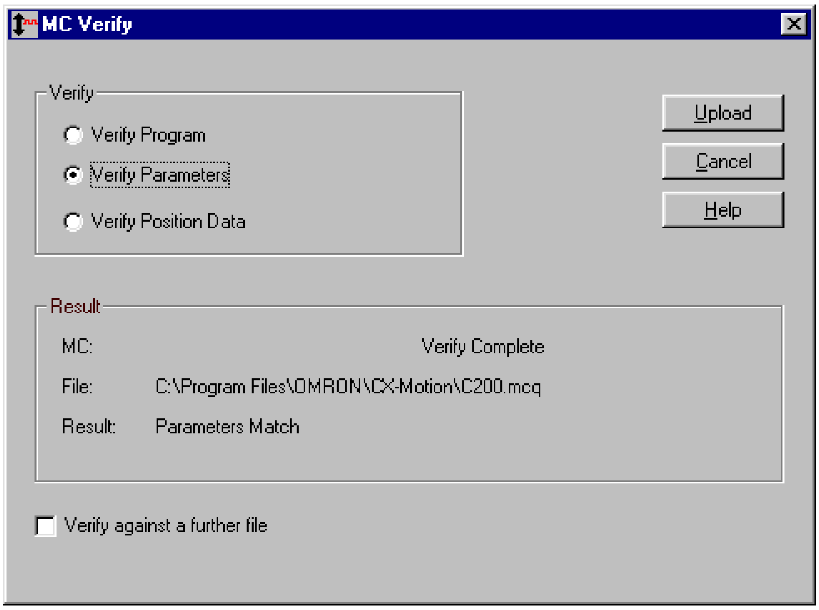


Look in: CX-Motion

123.mcq
99.5.17.mcq
C200.mcq
CS1.mcq
CV500MC221.mcq
kekmc.mcq
Ptime.mcq

File name:
Files of type: Parameter Files (*.mcq)

Then at the **MC Verify** window, the compare result will be shown.



MC: Verify Complete

File: C:\Program Files\OMRON\CX-Motion\C200.mcq

Result: Parameters Match

□ Verify against a further file

To verify with multiple files, check the checkbox at **Verify against a further file**. Then click on **OK**.
The **Open** dialogue box will appear. Select the file to compare will and click **Open**.
The result will be shown.

Section 2
Position Data

In this chapter, the different method of editing the position data will be described.

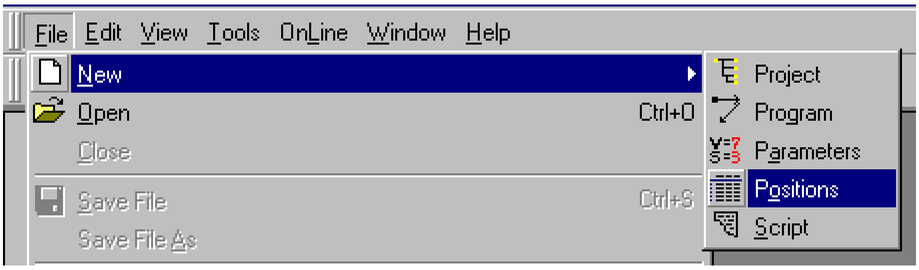
2-1 Position Data

The motion of the machine controlled by an OMRON PLC and MC unit combination is defined using one or more G Code programs. These programs contain sequences of movements and delays which combine across the two or four axes of the controller to perform the job required. The G Code programs work in conjunction with the parameter settings and position data of the project.

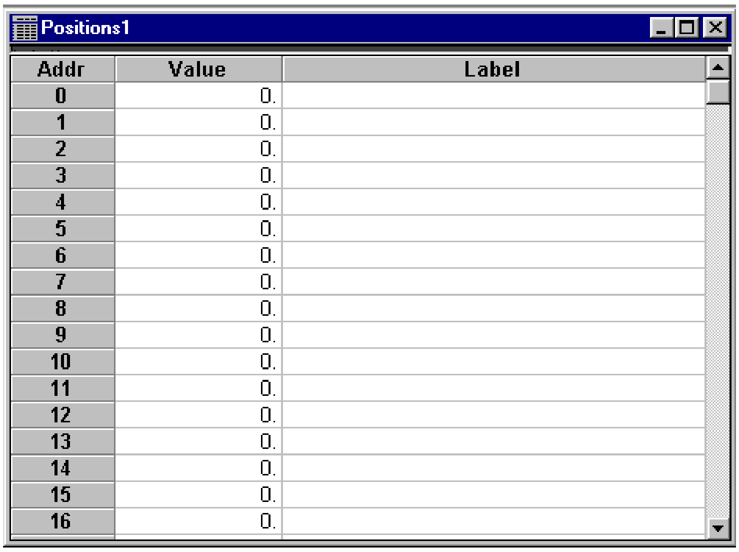
2-1-1 Creating Position Data file

To create position data, please perform the following steps:

File - New - Positions



The Position Window as shown will appear.



Addr	Value	Label
0	0.	
1	0.	
2	0.	
3	0.	
4	0.	
5	0.	
6	0.	
7	0.	
8	0.	
9	0.	
10	0.	
11	0.	
12	0.	
13	0.	
14	0.	
15	0.	
16	0.	

As the MC unit are able to store up to 2000 position data, the position data are addressed using addresses A0000 to A1999in G-Code programs.

The position data will be stored in the following format.

Example:

For 2-axis system: (XY)

A50	First X-position
A51	First Y-position
A52	Second X-position
A53	Second Y-position

For 4-axes system (XYZU)

A50	First X-position
A51	First Y-position
A52	First Z-position
A53	First U-position
A54	Second X-position
A55	Second Y-position
A56	Second Z-position
A57	Second U-position

Example of G-Code program using the position data.

N010 G01 XA1000 MA1001

Explanation:
The content of A1000 (Address 1000 for position data) will be used for X-axis, and the content of A1001 will be used for M code.

Example of G-Code program using the position data

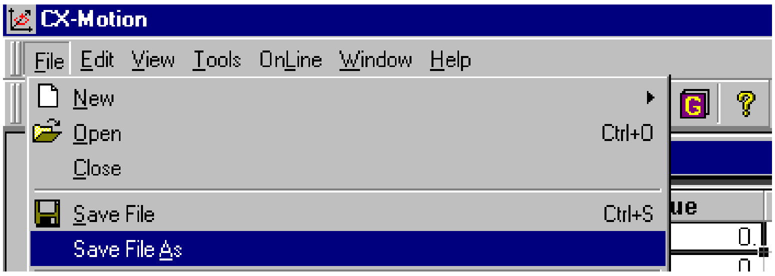
N009 G90 Absolute specification.
N010 G00 XA1000 YA1001 M001

Explanation:
The content of A1000 (Address 1000 for position data) will be used for X-axis, and the content of A1001 will be used for Y-axis.

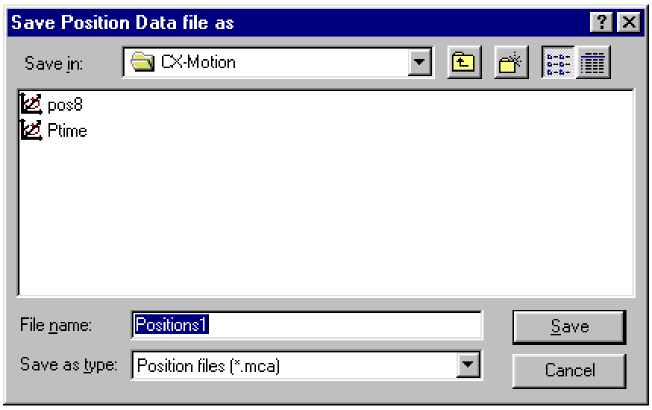
In completion of creating the position data, save the file with an extension ".mca".

The steps are as followed:

File- Save File as



The following dialog box will appear.



File name: Positions1

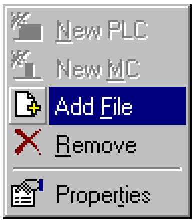
Save as type: Position files (*.mca)

Enter the desire filename and click on **Save**.

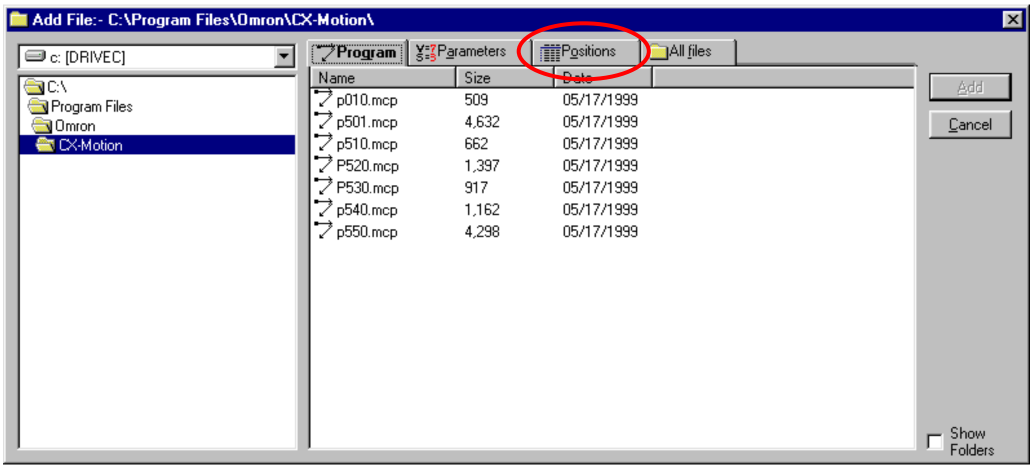
2-1-2 Inserting Position Data file to Project

To insert the position data to a project, please followed the steps as shown.

At the Project Tree, use the right mouse to click on the desire MC card, the following pop-up window will appear.

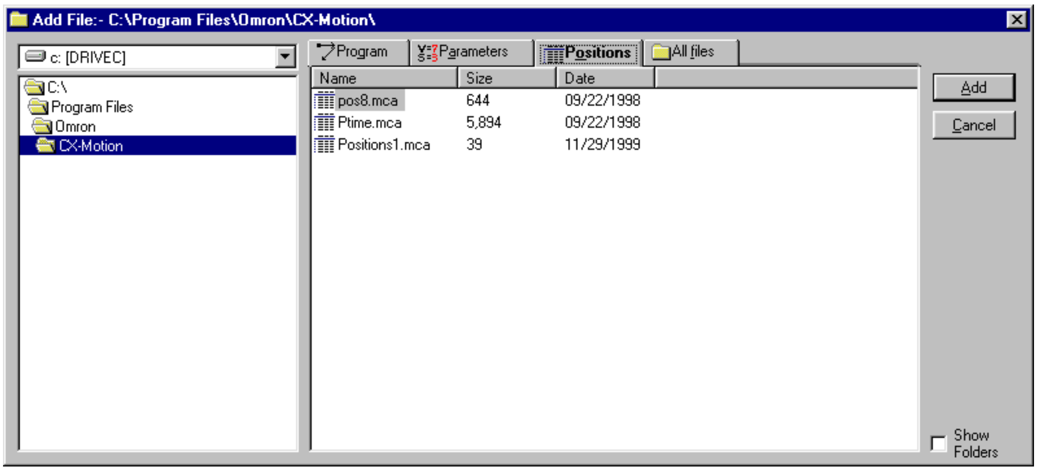


Select **Add File**, from the file selection dialog box, select the Position Tab.



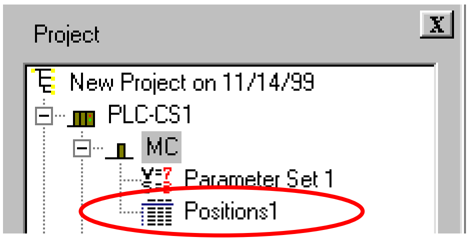
Name	Size	Date
p010.mcp	509	05/17/1999
p501.mcp	4,632	05/17/1999
p510.mcp	662	05/17/1999
P520.mcp	1,397	05/17/1999
P530.mcp	917	05/17/1999
p540.mcp	1,162	05/17/1999
p550.mcp	4,298	05/17/1999

Then select the desire file to insert, then click on **Add**.



Name	Size	Date
pos8.mca	644	09/22/1998
Ptime.mca	5,894	09/22/1998
Positions1.mca	39	11/29/1999

At the project tree, the position file icon will be shown.
Please note that only one position data file are allowed.

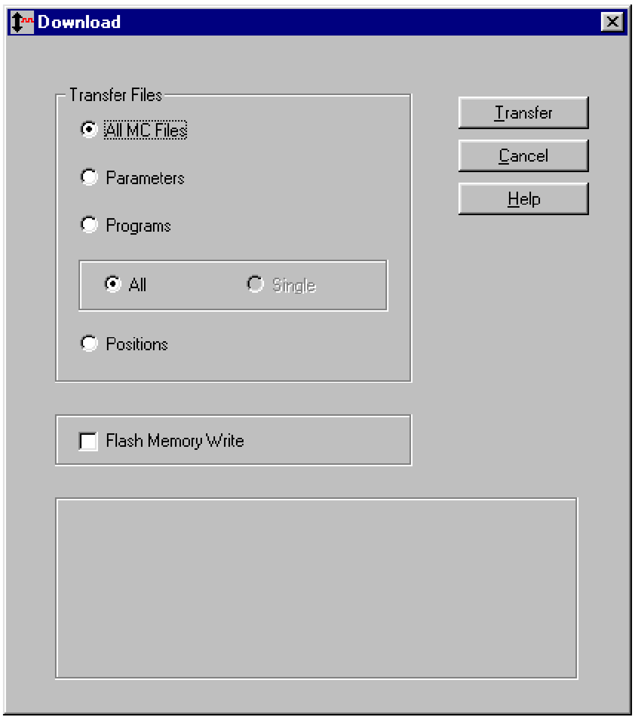


2-2 Downloading and Upload of Position data file

Downloading and Uploading of the Position data are supported by CX-Motion, to execute these functions provided, please follow the steps as shown.

2-2-1 Downloading of Position Data

Select a MC unit to download, click on . (Download Icon.) The Download Dialog box will appear, check the Postions checkbox, or;



Download

Transfer Files

● All MC Files

○ Parameters

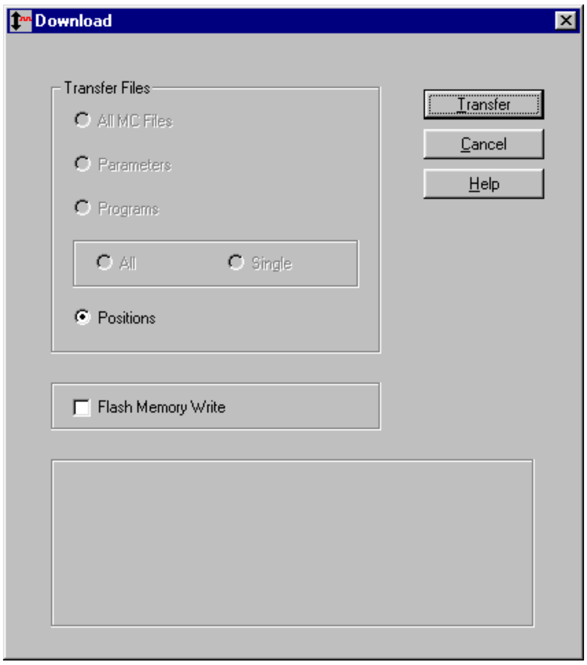
○ Programs

● All ○ Single

○ Positions

☐ Flash Memory Write

At the project tree, click on Position data file, and click on



Download

Transfer Files

○ All MC Files

○ Parameters

○ Programs

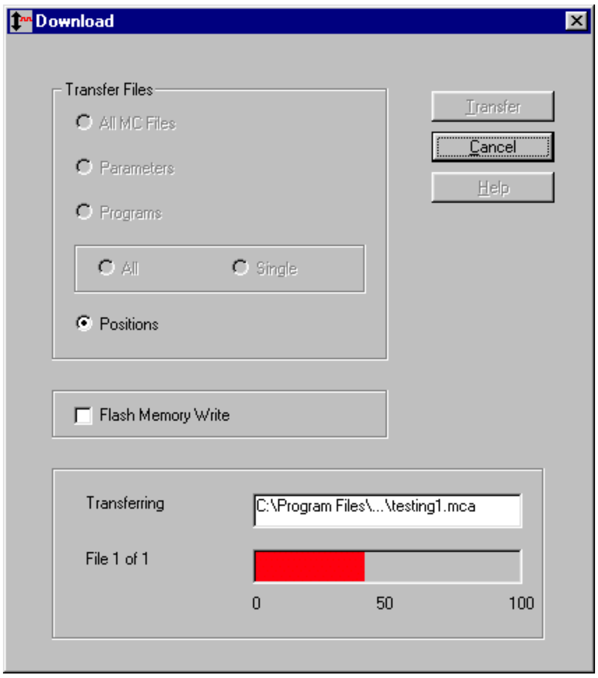
○ All ○ Single

● Positions

☐ Flash Memory Write

Note: Only **Positions** is highlighted

Then select **Transfer**. The downloading status of files will be shown.



Transfer Files

○ All MC Files

○ Parameters

○ Programs

○ All ○ Single

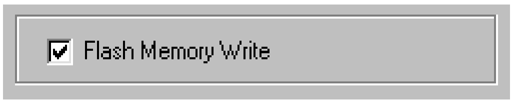
● Positions

□ Flash Memory Write

Transferring C:\Program Files\...\testing1.mca

File 1 of 1

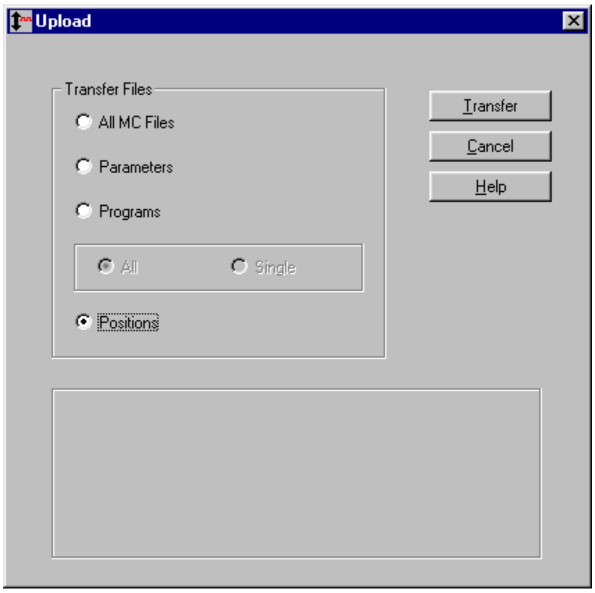
To download the position data to the Flash Memory, check the checkbox at **Flash Memory Write**.



☑ Flash Memory Write

2-2-2 Uploading of Position Data

To upload position data, select the desire MC unit, then click on ; check the checkbox or the Position; or



Transfer Files

○ All MC Files

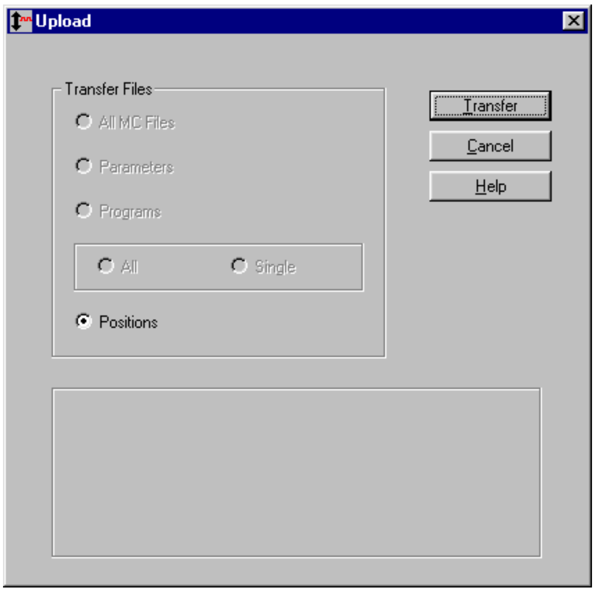
○ Parameters

○ Programs

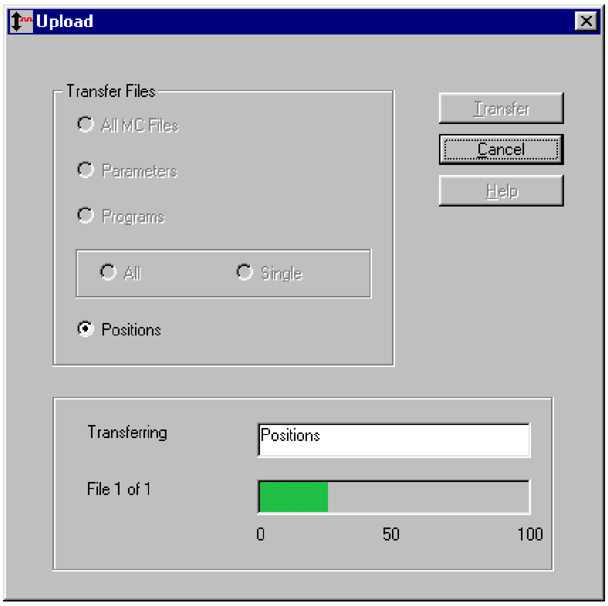
● All ○ Single

● Positions

At the project tree window, click on the Position data file and click ↑.



Then click on **Transfer** to upload the Position data. The status of file transfer will be shown.



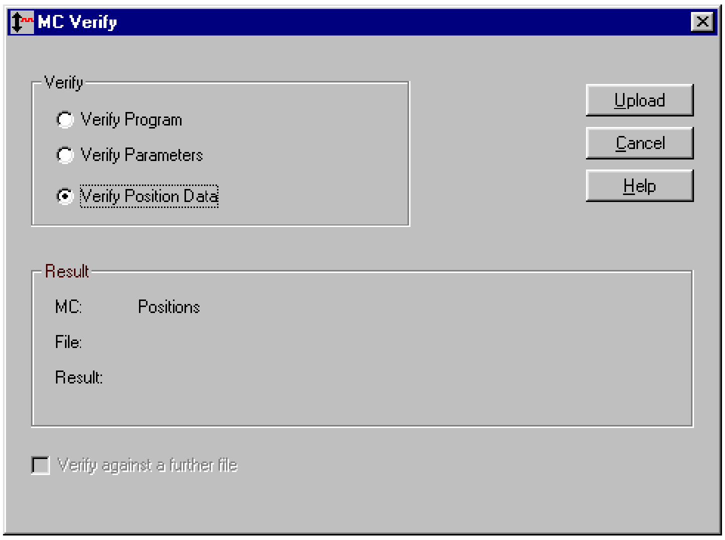
Transferring Positions

File 1 of 1

Note: The MC unit allows only one Position data file.

2-2-3 Verifying of Position Data

To verify the Position data, click on ↕, then select **Verify Position Data**.

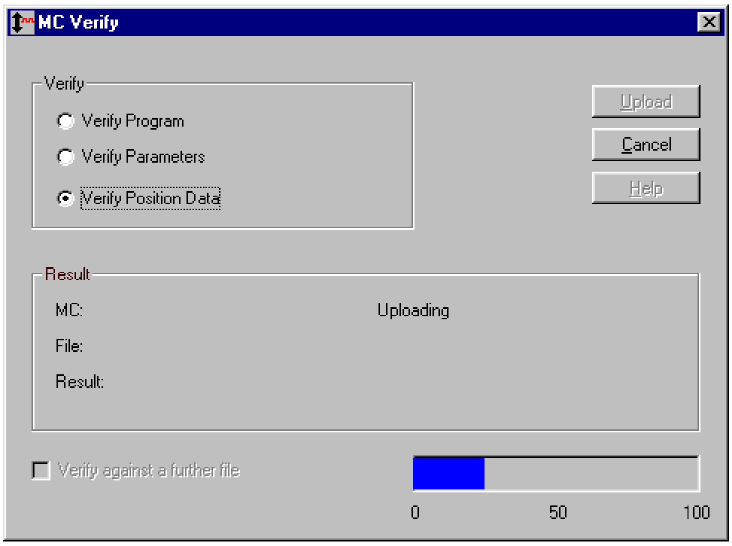


Verify
○ Verify Program
○ Verify Parameters
● Verify Position Data

Result
MC: Positions
File:
Result:

□ Verify against a further file

Then, click on **Upload**. The status of file transfer will be shown at the bottom left corner of the **MC Verify** window.

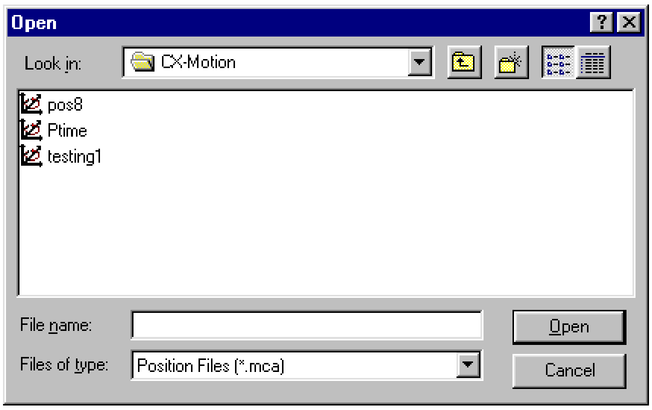


Verify
○ Verify Program
○ Verify Parameters
● Verify Position Data

Result
MC: Uploading
File:
Result:

□ Verify against a further file

When completion, the **Open** dialogue box will appear, thus prompting user to select a file to verify with.

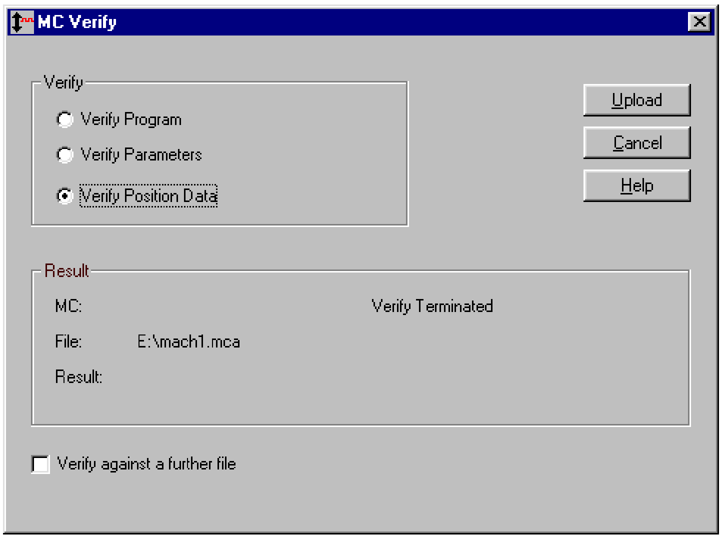


Look in: CX-Motion

pos8
Ptime
testing1

File name:
Files of type: Position Files (*.mca)

Then the selected file is being compare and the result will be shown at the result portion.

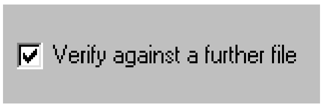


Verify
○ Verify Program
○ Verify Parameters
● Verify Position Data

Result
MC: Verify Terminated
File: E:\mach1.mca
Result:

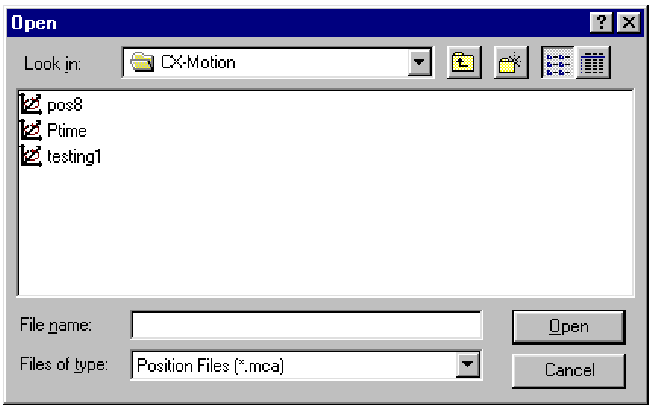
□ Verify against a further file

To verify against a further file, check the checkbox as shown.



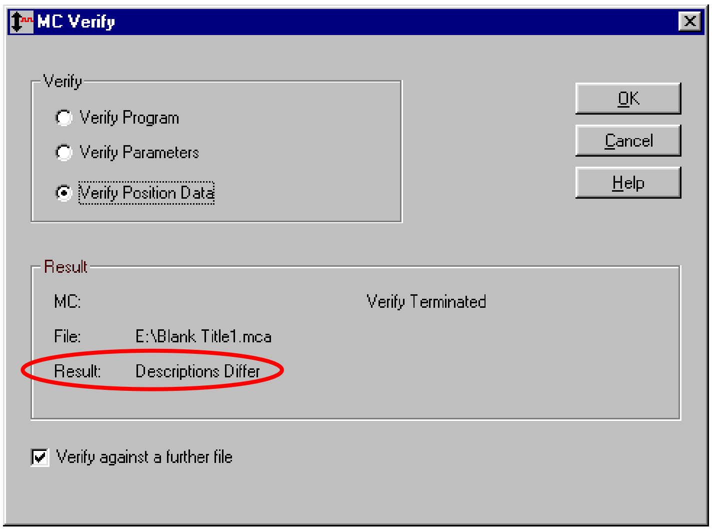
☑ Verify against a further file

Then click on **OK**. At the **Open** dialogue box, select the file to verify against.



Look in: CX-Motion
pos8
Ptime
testing1
File name:
Files of type: Position Files (*.mca)

After verification, the result will be shown.



Verify
○ Verify Program
○ Verify Parameters
● Verify Position Data

Result
MC: Verify Terminated
File: E:\Blank Title1.mca
Result: Descriptions Differ

☑ Verify against a further file

2-3 Changing Position Data via Programming

Position Data can be change via Ladder Programming. In the G code Program, users are able use AXXXX as a variable. Hence, with the help of the Ladder Program, users are able to change the value as required by the application.

2-3-1 CS1W-MC421/221

The procedure of changing the position data are as shown.

1,2,3… 1. Position Data Addressing

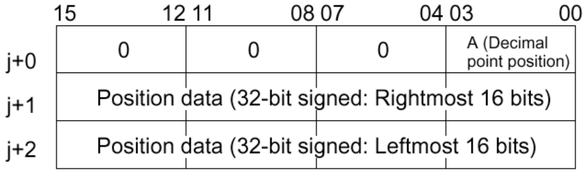
The addresses of the positions data are specified in hexadecimal when using IOWR/IORD commands.

Data	Address Range
Position data	0000 to 07CF (decimal: 0000 to 1999)

The preceding addresses 0000 to 1999 are specified using A0000 to A1999 in G language.

2. Position Data Configuration

Each position is comprised of three words.


15
12 11
08 07
04 03
00
j+0
0
0
0
A (Decimal point position)
j+1
Position data (32-bit signed: Rightmost 16 bits)
j+2
Position data (32-bit signed: Leftmost 16 bits)

A: Decimal Point Position	0: x 1
	1: x 0.1
	2: x 0.01
	3: x 0.001
	4: x 0.0001

For example, the position data is 1234.567 (123456 = 1E240)

j+0 :	0003
j+1 :	E240
j+2 :	0001

Another example, the position data is -387654.32 (-38765432 = FDB07C88)

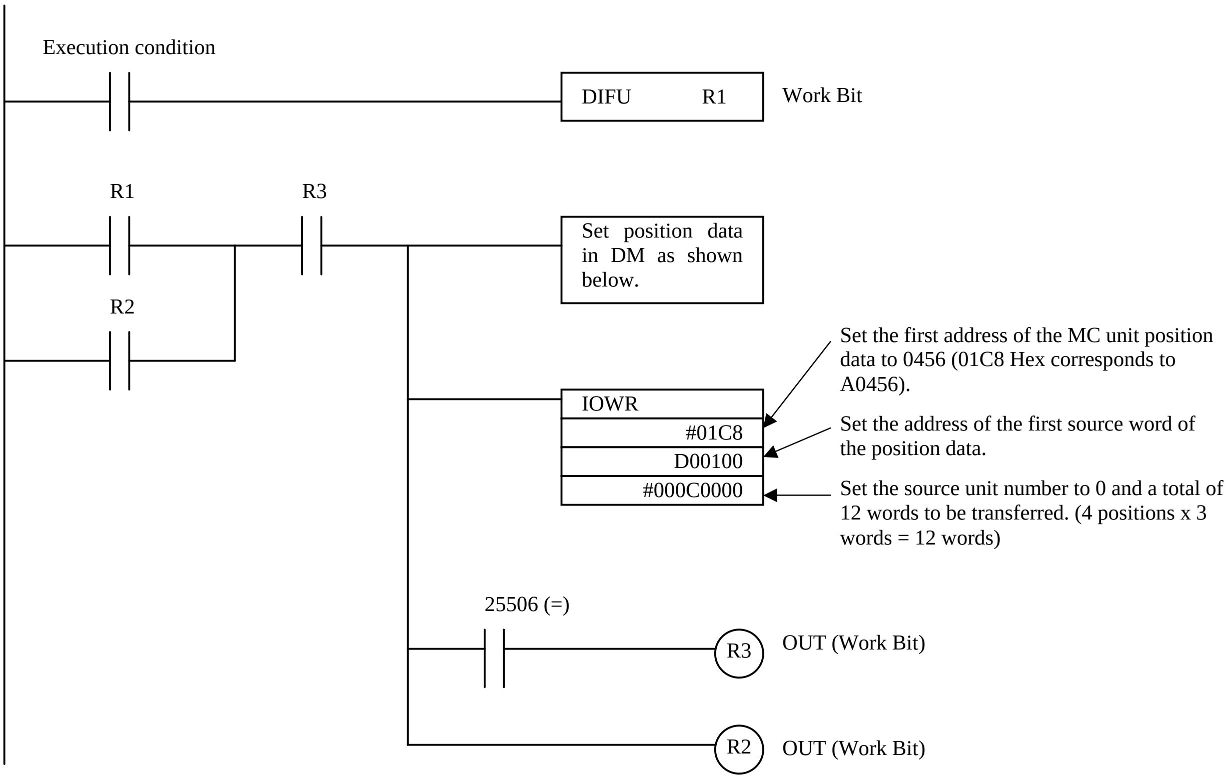
j+0 :	0002
j+1 :	7C88
j+2 :	FDB0

3. Using Ladder Programming.
In these examples, the IOWR operands and transfer data are all assumed to be correct. When debugging, write ladder code that will check the status of the Error Flag and bit 14 of word n + 10/n + 18.

Example 1 Transferring Four Positions to A0456 to A0459 from D0100 to D0111
Position data Position data address

A0456	345678.90
A0457	123456.78
A0458	–387654.32
A0459	123.456

Ladder Program



Set the first address of the MC unit position data to 0456 (01C8 Hex corresponds to A0456).

Set the address of the first source word of the position data.

Set the source unit number to 0 and a total of 12 words to be transferred. (4 positions x 3 words = 12 words)

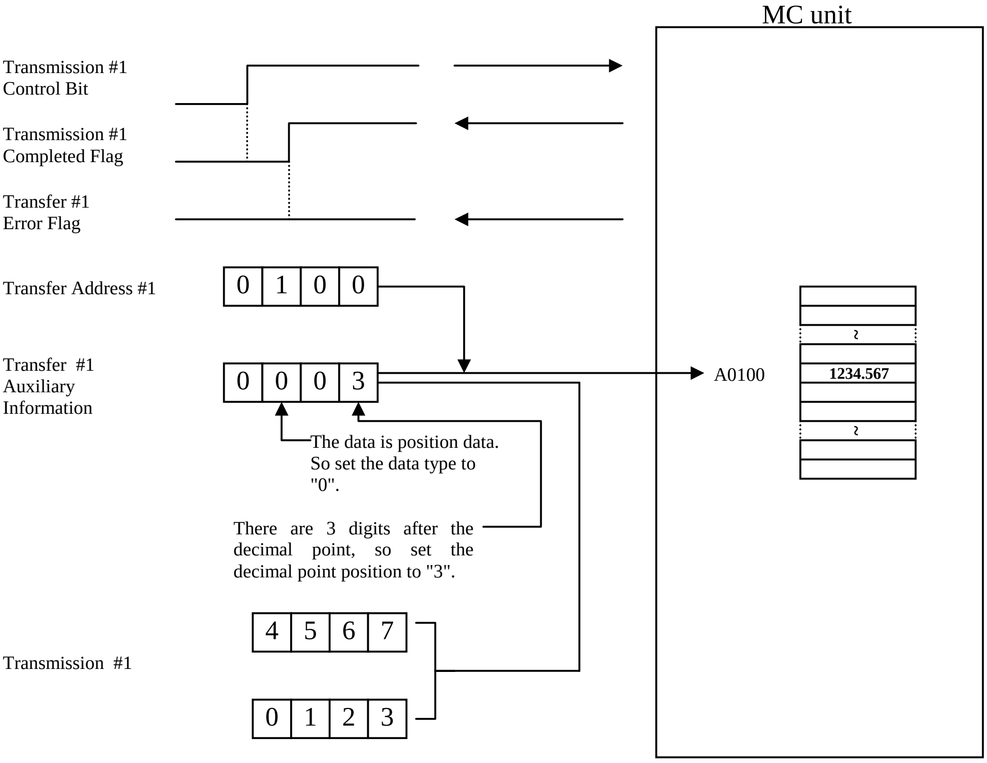
Settings
DM Area

D00100	0002	345678.90
D00101	76D2	
D00102	020F	
D00103	0002	123456.78
D00104	614E	
D00105	00BC	
D00106	0002	-387654.32
D00107	7C88	
D00108	FDB0	
D00109	0003	123.456
D00110	E240	
D00111	0001	

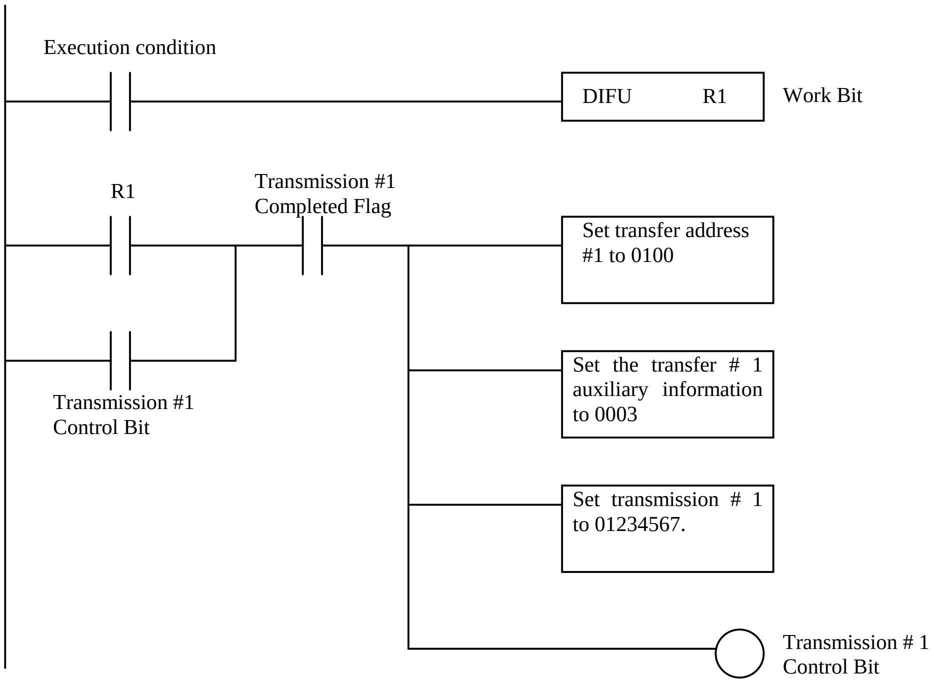
2-3-2 CV500-MC421/MC221

To send new position data to the MC unit, the procedure are as shown.

The following diagram shows how to transmit the value 1234.567 to position data address A0100.


MC unit
Transmission #1 Control Bit
Transmission #1 Completed Flag
Transfer #1 Error Flag
Transfer Address #1
0 1 0 0
Transfer #1 Auxiliary Information
0 0 0 3
The data is position data. So set the data type to "0".
There are 3 digits after the decimal point, so set the decimal point position to "3".
Transmission #1
4 5 6 7
0 1 2 3
A0100
1234.567

Ladder Program

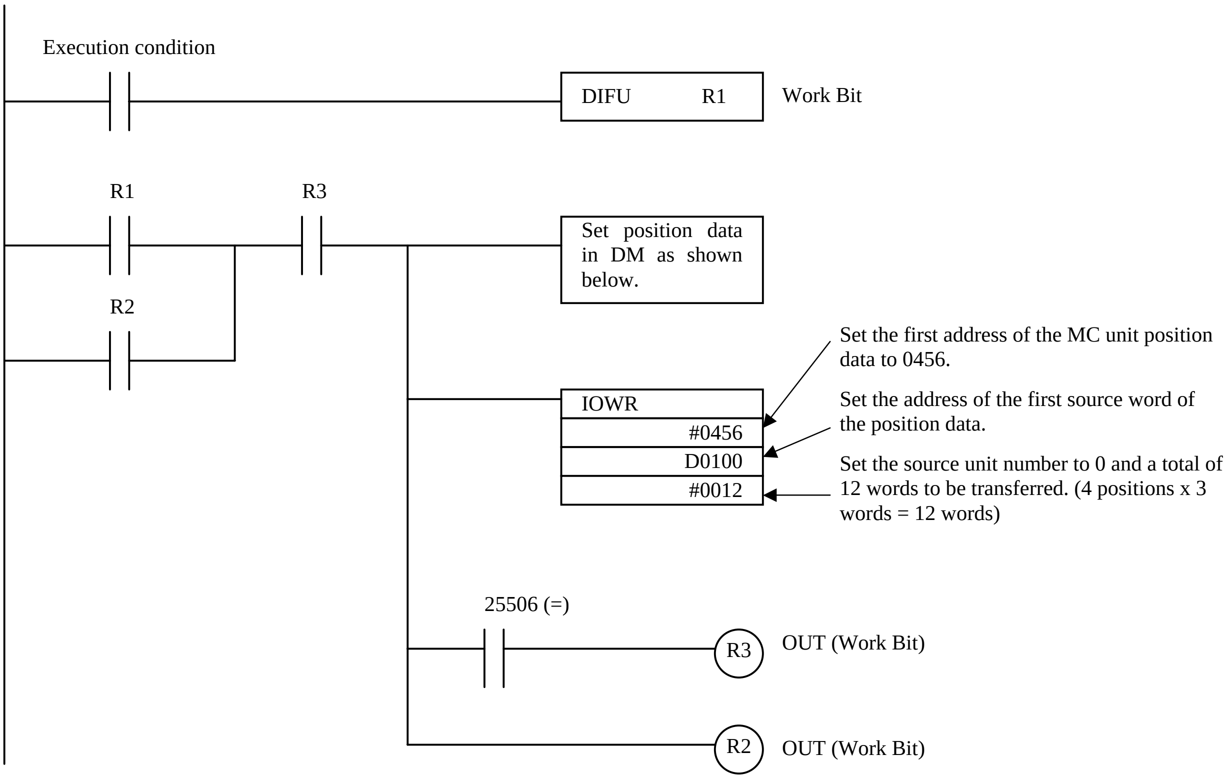

Execution condition
DIFU R1
Work Bit
R1
Transmission #1 Completed Flag
Set transfer address #1 to 0100
Transmission #1 Control Bit
Set the transfer # 1 auxiliary information to 0003
Set transmission # 1 to 01234567.
Transmission # 1 Control Bit

2-3-3 C200H-MC221

The process of sending new position data to C200H-MC221 is similar to CS1, that is , the IOWR instruction is used.

Instead of expressing the data in terms of hexadecimal, the values of the data is express in terms of BCD.

Ladder Diagram


Execution condition
DIFU R1
Work Bit
R1
R3
Set position data in DM as shown below.
R2
IOWR
#0456
D0100
#0012
Set the first address of the MC unit position data to 0456.
Set the address of the first source word of the position data.
Set the source unit number to 0 and a total of 12 words to be transferred. (4 positions x 3 words = 12 words)
25506 (=)
R3
OUT (Work Bit)
R2
OUT (Work Bit)

Settings
DM Area

D00100	0002	345678.90
D00101	7890	
D00102	3456	
D00103	0002	123456.78
D00104	5678	
D00105	1234	
D00106	0002	-387654.32
D00107	5432	
D00108	B876	
D00109	0003	123.456
D00110	3456	
D00111	0012	

Section 3
G-Code Programming

This section will describe how G-Code program can be created in CX-Motion. The detail of individual G-Code description can be found in the Appendix A.

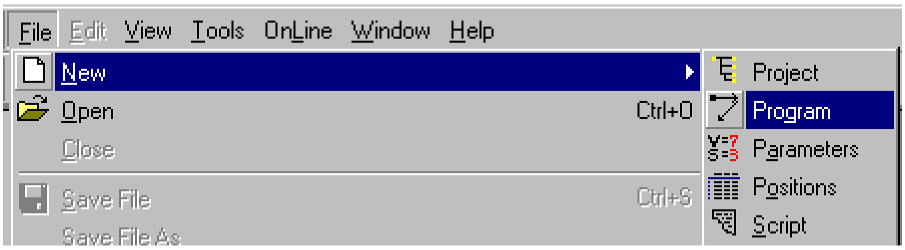
3-1 Program

The motion of the machine controlled by an OMRON PLC and MC unit combination is defined using one or more G Code programs. These programs contain sequences of movements and delays, which combine across the two or four axes of the controller, to perform the job required. The G Code program or programs work in conjunction with the parameter settings and position data of the project. Up to 100 programs can program into the MC unit. Hence, CX-Motion provides the utilities for programmer to write G-code programming for MC unit.

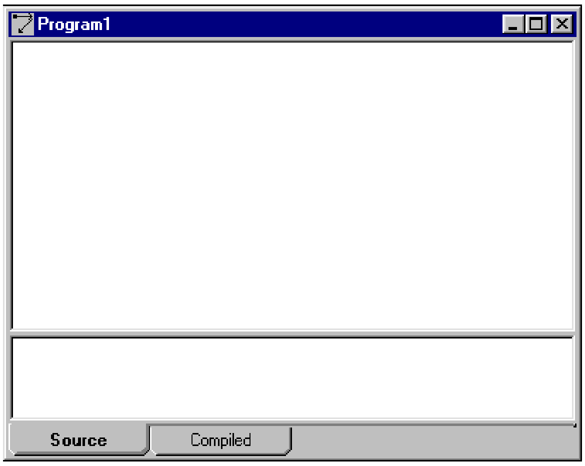
3-1-1 Creating New program

To create a new program file, please follow the steps as shown.

File - New - Program

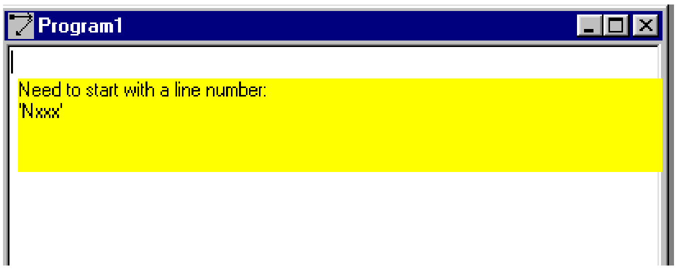


The programming window will appear.



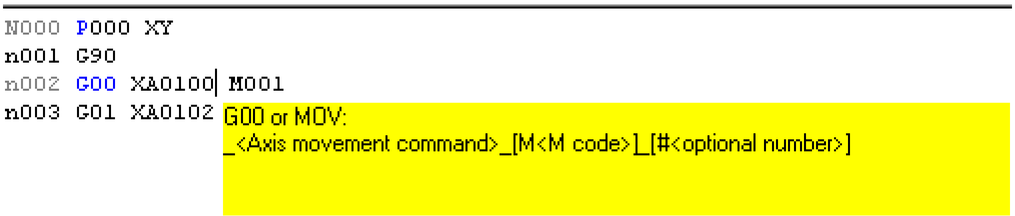
3-1-2 Programming

To start programming, click on the workspace provided.
Online help will appear in a yellow box from time to time, guiding programmer on the programming structure.



Need to start with a line number:
'Nxxx'

Guides on G-code will also be shown.



```
N000 P000 XY
n001 G90
n002 G00 XA0100 M001
n003 G01 XA0102 G00 or MOV:
                _<Axis movement command>_[M<M code>]_[#<optional number>]
```

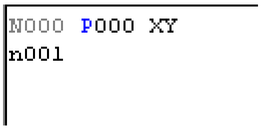
In the program, each command line will be identified by a NXXX (representing the block/line number).
In the first line of a program, it is normal started with a declaration of program number, which is represented by PXXX, and a notification on the usage of the axes (XYZU).

Example:

N000 P000 XY
:
:
N003 G90
:
:

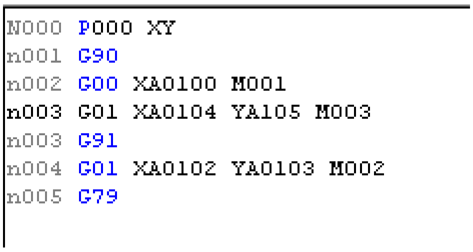
Note: The block number varies from 000 to 800 (maximum per program), and the program number varies from 0 to 100 (which depend on the number of task used.)

In every carriage return at the end of command line, the subsequent line number will automatically appear.



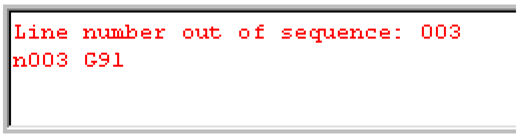
```
N000 P000 XY
n001
```

In the case of inserting a command in between written line, the line number will be jumble up.



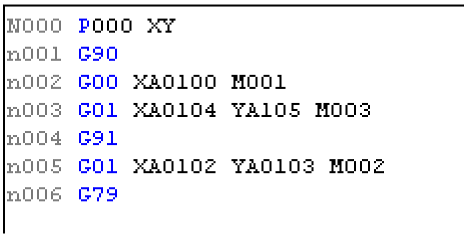
```
N000 P000 XY
n001 G90
n002 G00 XA0100 M001
n003 G01 XA0104 YA105 M003
n003 G91
n004 G01 XA0102 YA0103 M002
n005 G79
```

Or during compilation of the program, the following error will be shown.



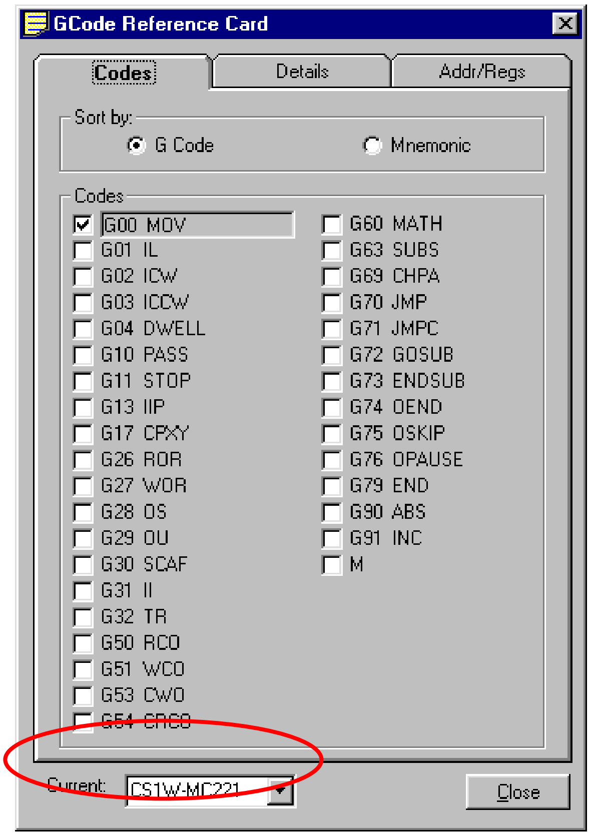
```
Line number out of sequence: 003
n003 G91
```

Click on ☰ to rearrange the line number.



```
N000 P000 XY
n001 G90
n002 G00 XA0100 M001
n003 G01 XA0104 YA105 M003
n004 G91
n005 G01 XA0102 YA0103 M002
n006 G79
```

To acquire details on the G-code, click on G. The G-code reference card dialog box will appear.



GCode Reference Card

Codes | Details | Addr/Regs

Sort by:
● G Code ○ Mnemonic

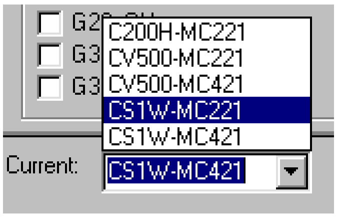
Codes

☑ G00 MOV
□ G01 IL
□ G02 ICW
□ G03 ICCW
□ G04 DWELL
□ G10 PASS
□ G11 STOP
□ G13 IIP
□ G17 CPXY
□ G26 ROR
□ G27 WOR
□ G28 OS
□ G29 OU
□ G30 SCAF
□ G31 II
□ G32 TR
□ G50 RCO
□ G51 WCO
□ G53 CWO
□ G54 CRCO
□ G60 MATH
□ G63 SUBS
□ G69 CHPA
□ G70 JMP
□ G71 JMPC
□ G72 GOSUB
□ G73 ENDSUB
□ G74 OEND
□ G75 OSKIP
□ G76 OPAUSE
□ G79 END
□ G90 ABS
□ G91 INC
□ M

Current: CS1W-MC221

Close

The **Codes** tab will display the G-code available for the particular MC unit, selected by **Current**. To change the MC Model, click on the down-arrow.

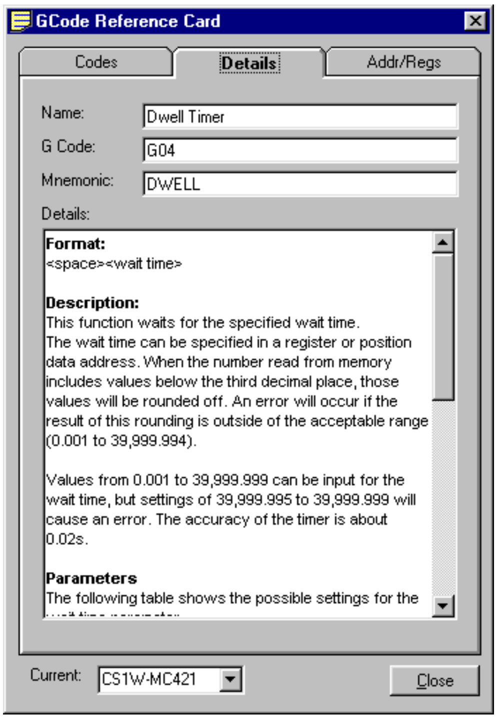


C200H-MC221
CV500-MC221
CV500-MC421
CS1W-MC221
CS1W-MC421

Current: CS1W-MC421

Choose the model of your choice, the G-code reference Card will display the applicable G-code.

To display the description of the G-code, check on the corresponding checkbox and click on the **Details** tab. The details of the selected G-code will appear.



Name: Dwell Timer
G Code: G04
Mnemonic: DWELL
Details:

Format:
<space><wait time>

Description:
This function waits for the specified wait time.
The wait time can be specified in a register or position data address. When the number read from memory includes values below the third decimal place, those values will be rounded off. An error will occur if the result of this rounding is outside of the acceptable range (0.001 to 39,999.994).

Values from 0.001 to 39,999.999 can be input for the wait time, but settings of 39,999.995 to 39,999.999 will cause an error. The accuracy of the timer is about 0.02s.

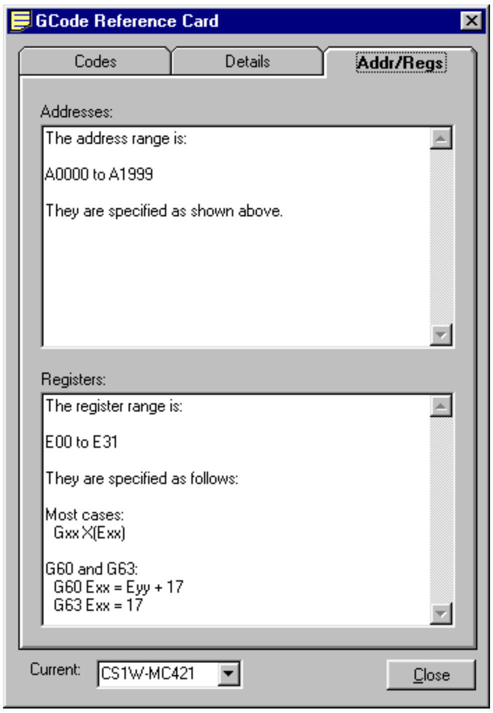
Parameters
The following table shows the possible settings for the

Current: CS1W-MC421

The descriptions includes:

- Format
- Description
- Parameters
- Note
- Caution
- Example Program

To view the address and register available in the MC units, select the desire MC unit and click in the **Addr/Regs** tab.



Addresses:

The address range is:

A0000 to A1999

They are specified as shown above.

Registers:

The register range is:

E00 to E31

They are specified as follows:

Most cases:
  GxxX(Exx)

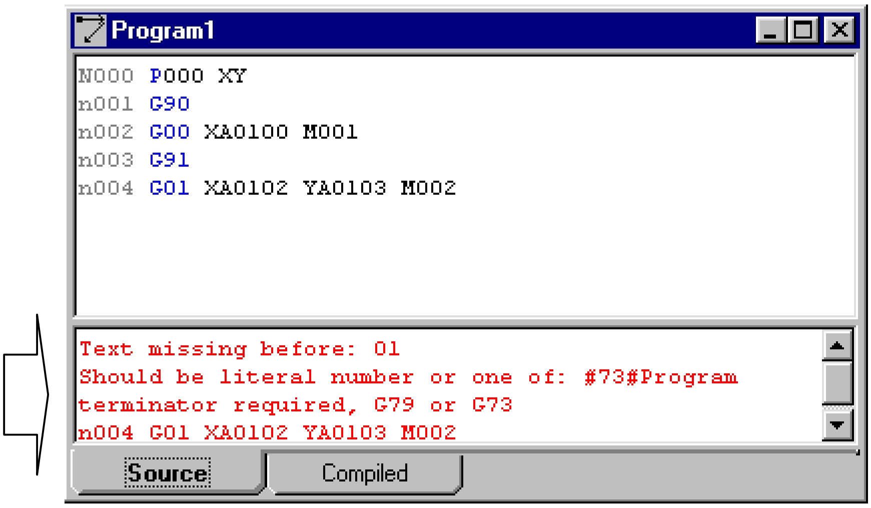
G60 and G63:
  G60 Exx = Eyy + 17
  G63 Exx = 17

Current: CS1W-MC421

3-1-3 Compilation of Program

In completion, users are able to compile the written G-code.

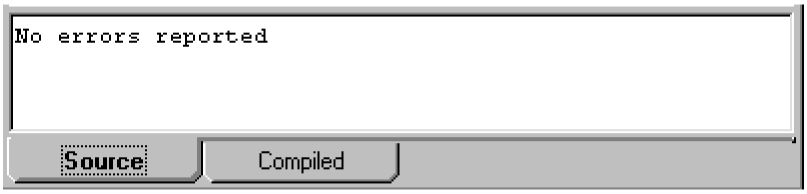
Click on the project workspace, then click on . The program will be compiled. The result of compilation will be shown at the bottom of the program window.



```
N000 P000 XY
n001 G90
n002 G00 XA0100 M001
n003 G91
n004 G01 XA0102 YA0103 M002
```

```
Text missing before: 01
Should be literal number or one of: #73#Program
terminator required, G79 or G73
n004 G01 XA0102 YA0103 M002
```

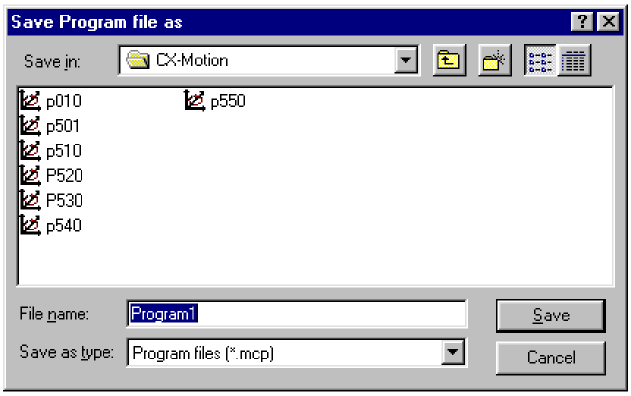
If error occurs, the description will be in red, else "No errors reported" with be displayed.



```
No errors reported
```

3-1-4 Saving of program

To save the written program, select **File** - **Save as**. Then, labelled the file.



Save in: CX-Motion

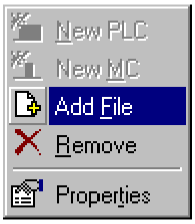
File name: Program1

Save as type: Program files (*.mcp)

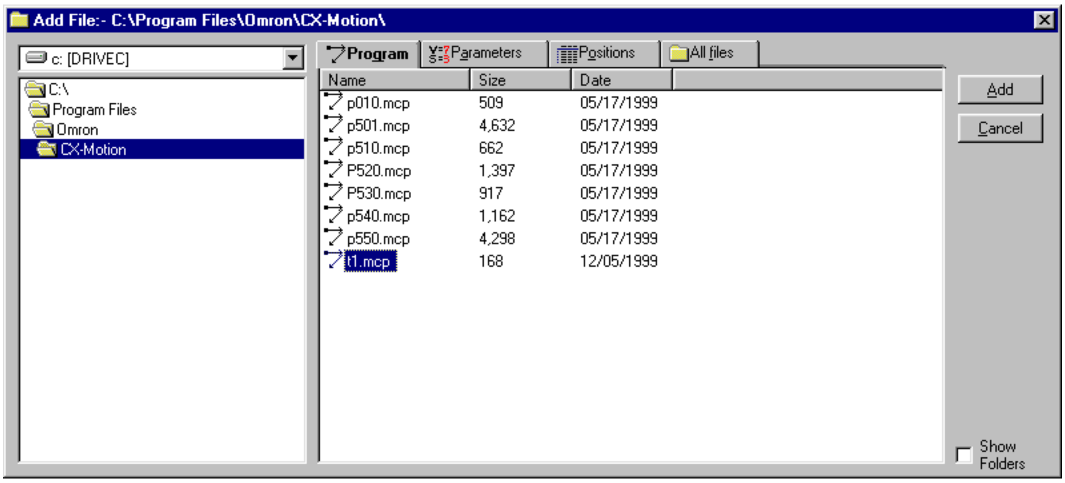
The G-code program file will have an extension ".mcp".

3-1-5 Inserting a file to project.

To insert a program file to a project, use the right mouse click to a select a MC unit. From the pop-up menu, select **Add File**.

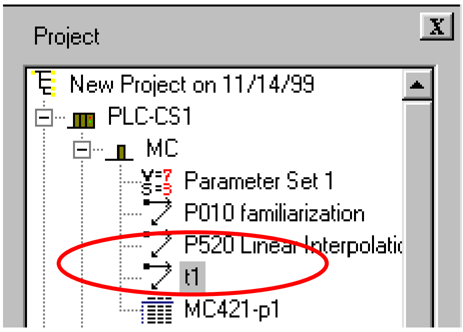


From the dialog box, select the file to include, and click on **Add.**



Name	Size	Date
p010.mcp	509	05/17/1999
p501.mcp	4,632	05/17/1999
p510.mcp	662	05/17/1999
P520.mcp	1,397	05/17/1999
P530.mcp	917	05/17/1999
p540.mcp	1,162	05/17/1999
p550.mcp	4,298	05/17/1999
t1.mcp	168	12/05/1999

At the project tree the filename will be displayed.



3-2 Downloading and uploading of Program Files

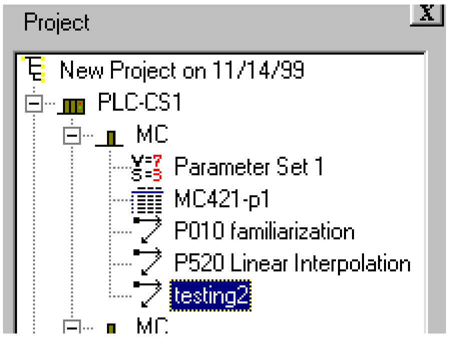
3-2-1 Downloading of Programs

Programs can be downloaded by two method, single program download or batch program download.

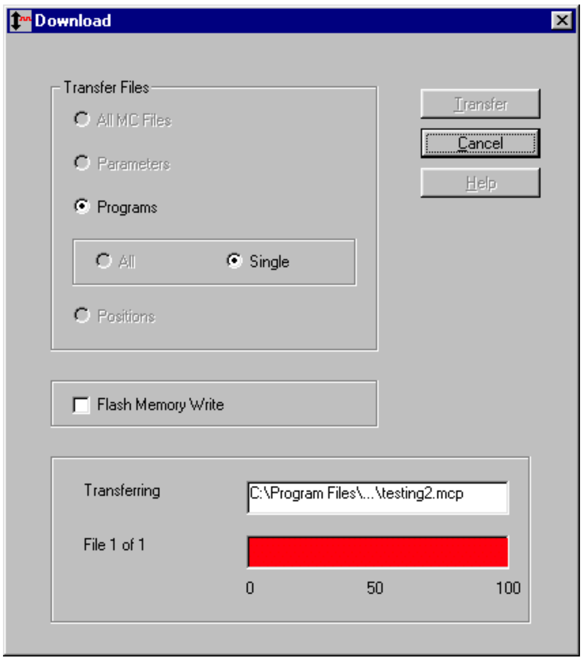
The steps are as shown.

Single Program download

To download a single program, at the project tree, highlight the filename, and click on .



Then at the **Download** Window, click on **Transfer**. The file transfer status will be shown at the lower portion of the **Download** Window.



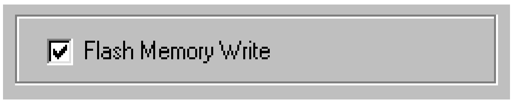
○ All MC Files
○ Parameters
● Programs
○ All ● Single
○ Positions

□ Flash Memory Write

Transferring C:\Program Files\...\testing2.mcp

File 1 of 1

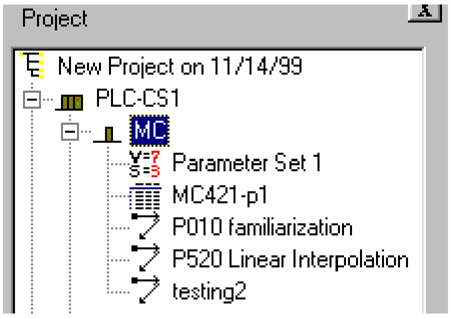
To download the program to the Flash Memory, check the checkbox at the Flash Memory Write.



☑ Flash Memory Write

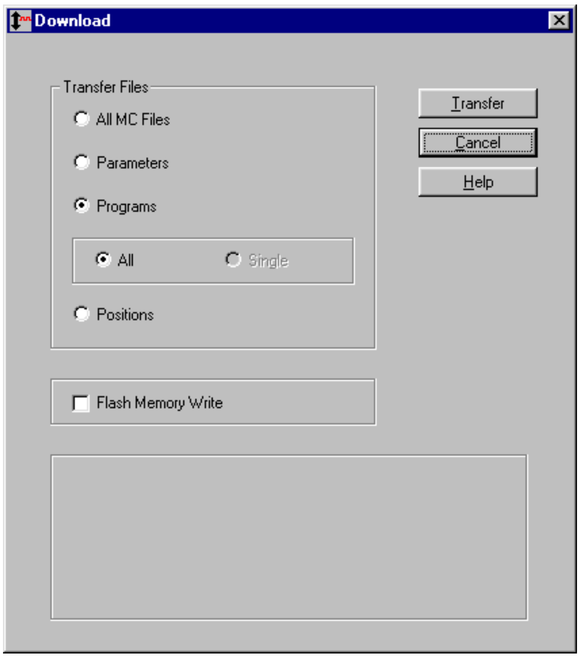
Multiple Programs download

To download all programs in one shot, at the project tree, select the MC unit and click on the download icon.



New Project on 11/14/99
PLC-CS1
MC
Parameter Set 1
MC421-p1
P010 familiarization
P520 Linear Interpolation
testing2

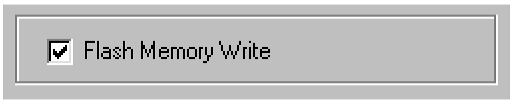
At the **Download** window, select Programs, and click on **Transfer** to proceed.



Transfer Files
○ All MC Files
○ Parameters
● Programs
● All ○ Single
○ Positions

□ Flash Memory Write

To download the program to the Flash Memory, check the checkbox at the Flash Memory Write.

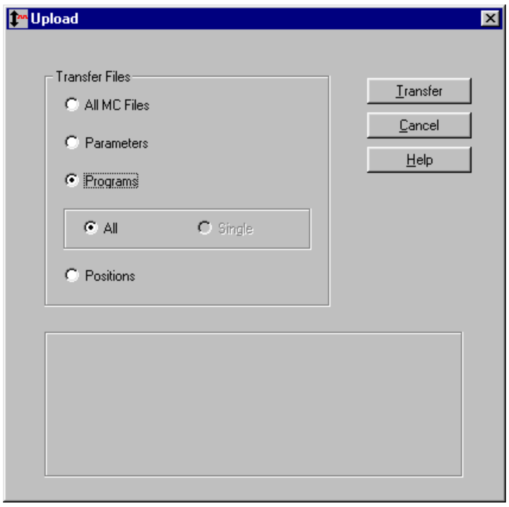


☑ Flash Memory Write

3-2-2 Uploading of Programs

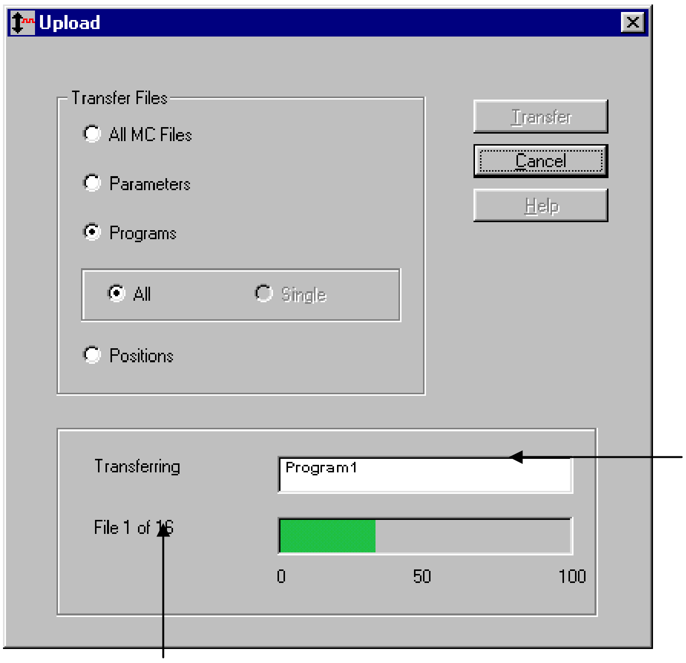
To upload the program from the MC unit, please proceed with the steps as shown.

At the project window, click on ↑. At the **Uploading** Window, select **Programs**, and click **Transfer**.



Transfer Files
○ All MC Files
○ Parameters
● Programs
● All ○ Single
○ Positions

The CX-Motion will upload program files until all files have been uploaded.



Transfer Files
○ All MC Files
○ Parameters
● Programs
● All ○ Single
○ Positions

Transferring Program1

File 1 of 16

Indicating the filename uploading at the moment.

Show the number of program files available in the MC unit

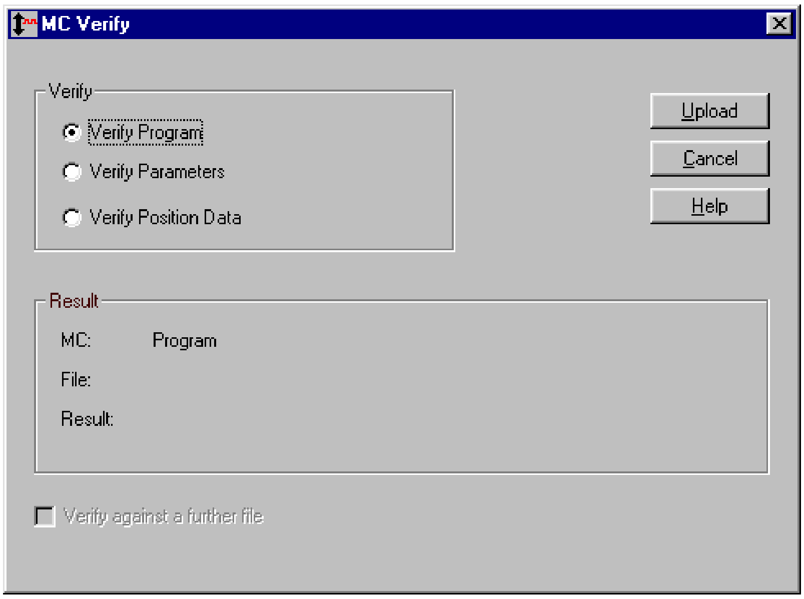
Hence, the **Transferring** box will constantly update the user by displaying the present uploading file.

In completion, all program files will be displayed on the screen.

3-2-3 Verifying of Program

To verify the program, please follow the steps as shown.

Select the MC unit for verification, then click on ↕. Then at the **MC Verify** Window, select **Verify Program**.



Verify

● Verify Program
○ Verify Parameters
○ Verify Position Data

Result

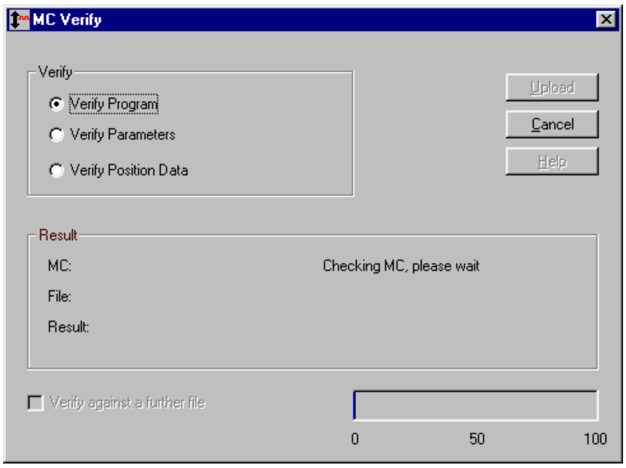
MC: Program

File:

Result:

□ Verify against a further file

Then click on **Upload**. " Checking MC, please wait" will be shown at the **Result** portion. (The CX-Motion will check for the number of program available in the MC unit.)



Verify

● Verify Program
○ Verify Parameters
○ Verify Position Data

Result

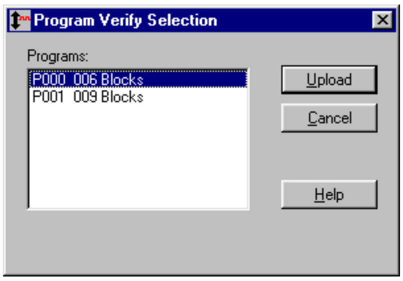
MC: Checking MC, please wait

File:

Result:

□ Verify against a further file

Upon checking, the following window will be shown.

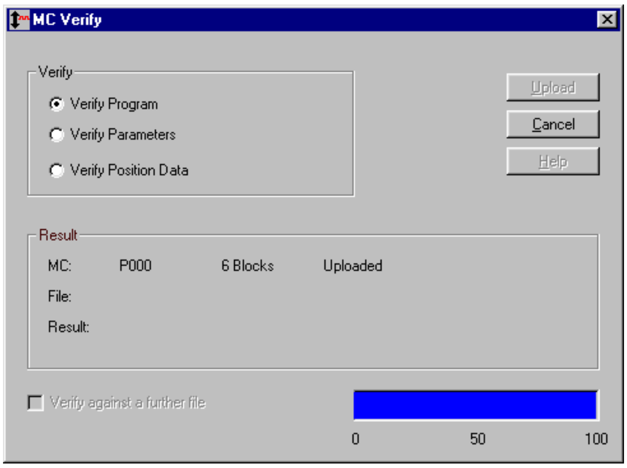


Programs:

P000 006 Blocks
P001 009 Blocks

Select the program to verify and click on **Upload**.

The status of the file transfer will be shown.



MC Verify

Verify
- ● Verify Program
- ○ Verify Parameters
- ○ Verify Position Data

Result

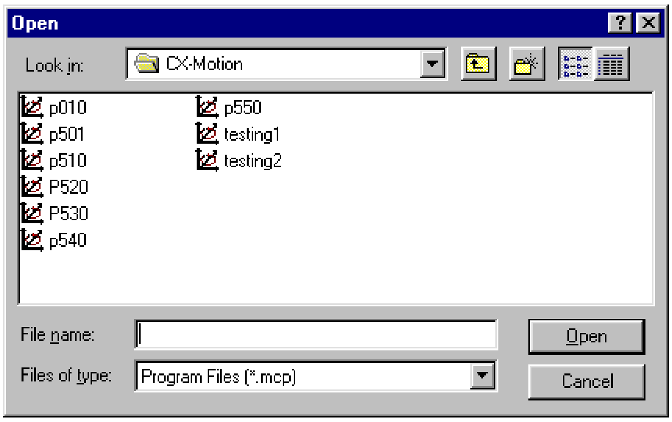
MC: P000 6 Blocks Uploaded

File:

Result:

□ Verify against a further file

Upon completion, the **Open** Window will prompt user to select the file to compare with. Select the appropriate file and click **OK**.



Open

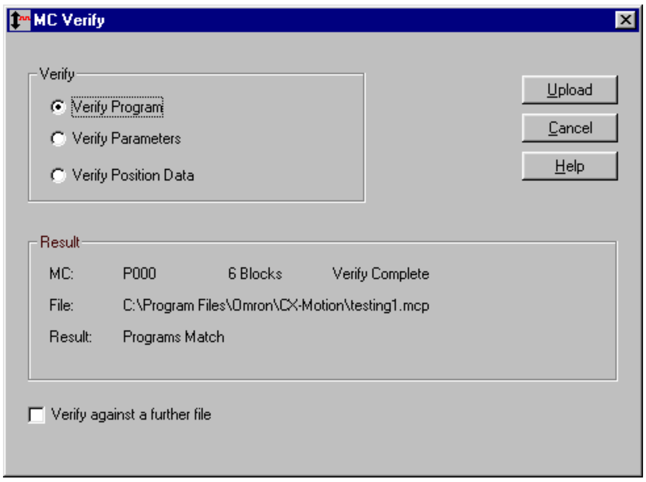
Look in: CX-Motion

p010, p501, p510, P520, P530, p540, p550, testing1, testing2

File name:

Files of type: Program Files (*.mcp)

The file will be verify, and the result will be shown at the result portion.



MC Verify

Verify
- ● Verify Program
- ○ Verify Parameters
- ○ Verify Position Data

Result

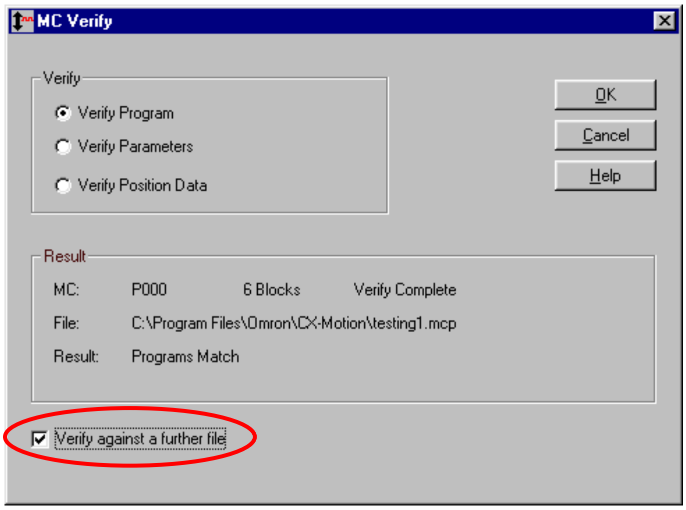
MC: P000 6 Blocks Verify Complete

File: C:\Program Files\Omron\CX-Motion\testing1.mcp

Result: Programs Match

□ Verify against a further file

To verify against a further file, check the checkbox at the **Verify against further file**, and click **OK**

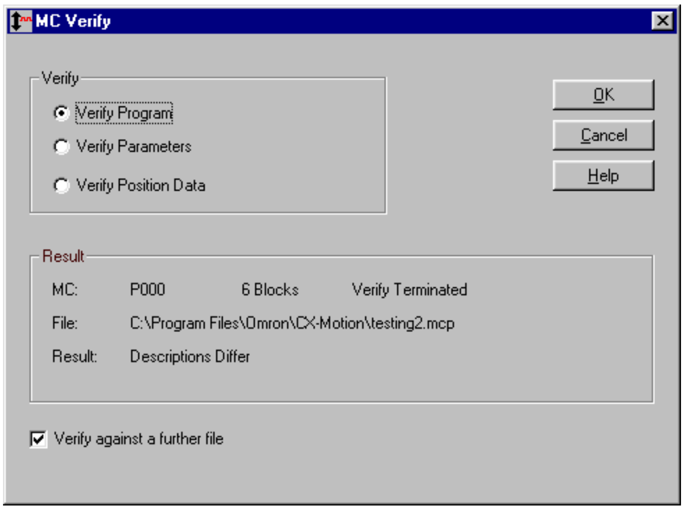


● Verify Program
○ Verify Parameters
○ Verify Position Data

MC: P000 6 Blocks Verify Complete
File: C:\Program Files\Omron\CX-Motion\testing1.mcp
Result: Programs Match

☑ Verify against a further file

The **Open** window will appear for the user to select the file to verify against. Then click OK to continue, the result of the verification will be shown.



● Verify Program
○ Verify Parameters
○ Verify Position Data

MC: P000 6 Blocks Verify Terminated
File: C:\Program Files\Omron\CX-Motion\testing2.mcp
Result: Descriptions Differ

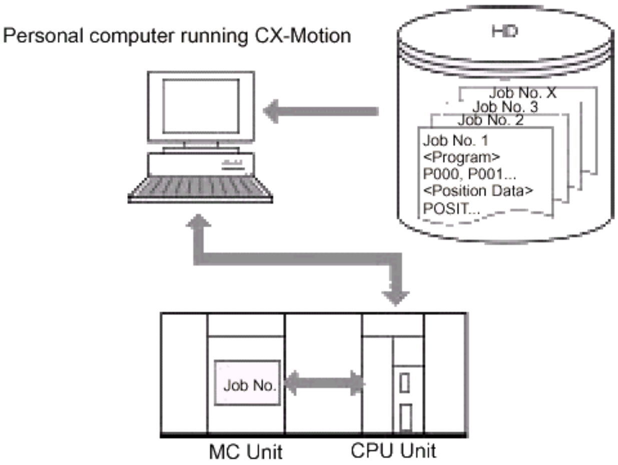
☑ Verify against a further file

Section 4
Automatic Downloading Function

This section will contain description of writing a script for Automatic Downloading of program files or position data. The activation of the Automatic Downloading Function will also be described.

4-1 Script

CS1 MC Unit has an auto-download features that allow a ladder control program running on the CPU to request new G code programs and position data for MC unit. This function allows more programs or program blocks to be executed by treating external memory devices connected to a personal computer running CX-Motion as MC Unit memory.
The automatic loading function is executed by the IOWR instruction for the Command Area. Either Automatic Mode or Manual Mode can be used.
Any number from 1 to 10,000 can be specified as the job number. CX-Motion must be used to create MC programs and position data in advance for job numbers that are to be used.



How does auto-downloading work?

1,2,3…
1. The MC unit's job number is always monitored by CX-Motion.
2. Using the IOWR instruction, a new job number is written to the job number currently in the MC unit.
3. When CX-Motion detects the specified job number, the program and position data are downloaded to the MC unit from the file for that the job number created by CX-Motion.
4. Before downloading the program and position data, CX-Motion deletes from the MC unit all of the programs for all tasks. Then it downloads the program and position data for the specified job number.
5. When the downloading has been completed, the CPU unit is notified from the MC unit. While the program and position are being downloaded, the Auto-loading it in the PC Interface Area turns from OFF to ON, and when the downloading is completed normally it turns from ON to OFF.

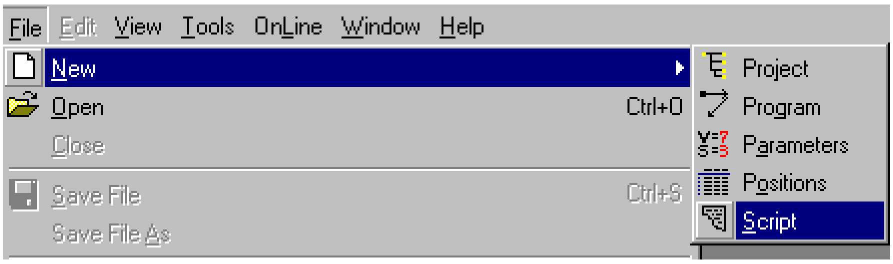
To enable the auto-downloading features, user must create a script file. This script file contains three sections:

- A project file
- A poll interval
- The jobs to process

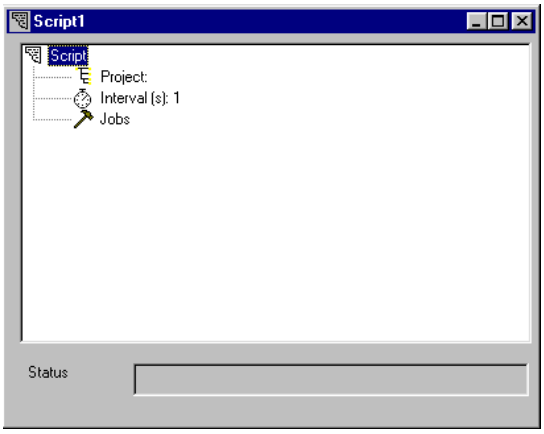
4-1-1 Creating of Script file

To create the script file, follow the steps as shown.

At the **File** menu, select **New** - **Script**.

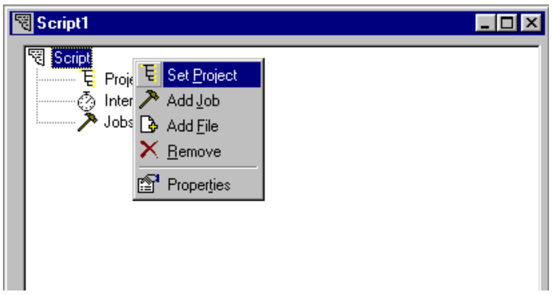


The script window will appear in the workspace.

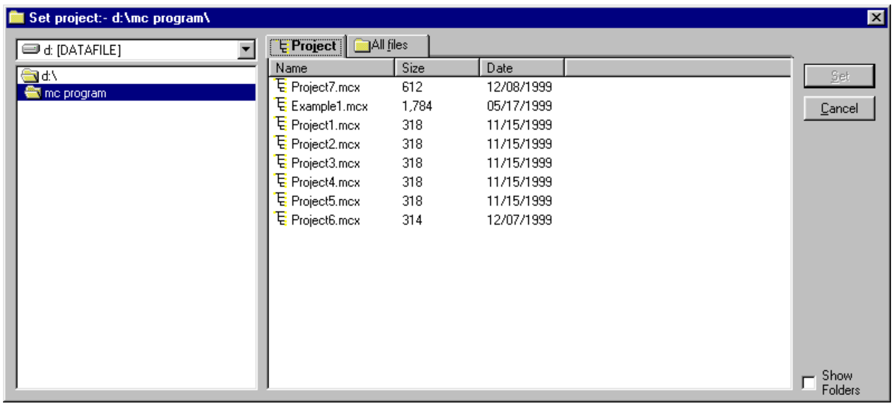


Adding Project File

To set the project file (i.e., information of the MC Unit.), click on the right mouse button, and select **Set Project**.

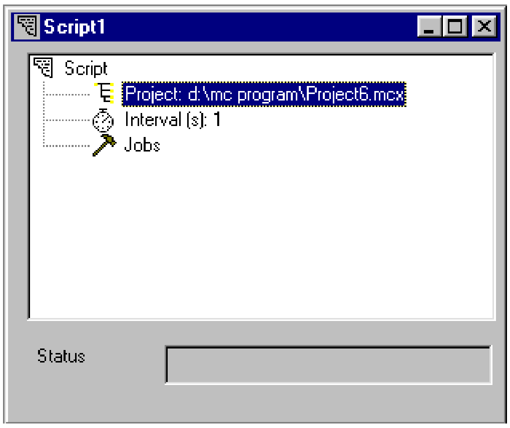


At the following window, select the project to add in, and click on **Set**.



Name	Size	Date
Project7.mcx	612	12/08/1999
Example1.mcx	1,784	05/17/1999
Project1.mcx	318	11/15/1999
Project2.mcx	318	11/15/1999
Project3.mcx	318	11/15/1999
Project4.mcx	318	11/15/1999
Project5.mcx	318	11/15/1999
Project6.mcx	314	12/07/1999

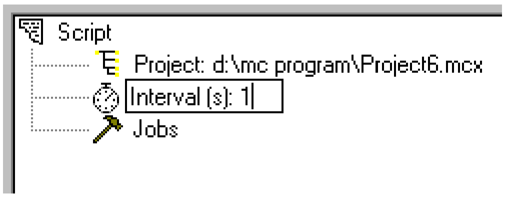
At the Script window, the filename and the path will be shown.



Setting Time Interval

The time interval determines the polling time for the CX-Motion to poll the MC units for the job number. For example, if the script specifies a poll interval of two seconds, and two MC units, each unit will polled once every four seconds. If a job number is specified, the programs and position data corresponding to the job number will be downloaded to the MC unit. (The CX-Motion will delete all programs for all tasks before downloading.)

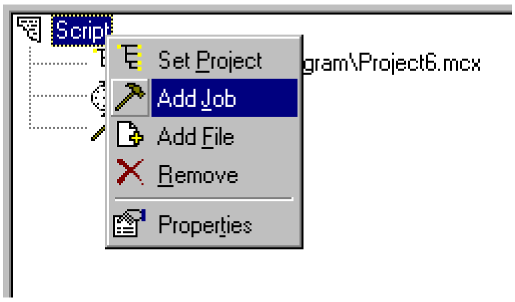
To change the interval time setting, highlight the **Interval (s)**, and click on it, when the blinking cursor appeared, key-in the desire time. The range is between 1 to 60 seconds.



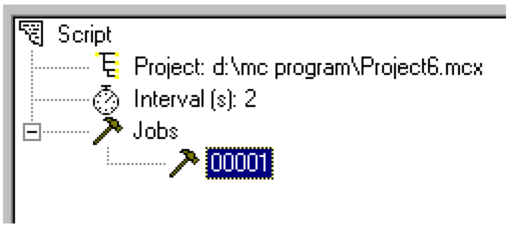
Define Jobs

At the **Jobs**, the job functions will be defined. The job number and the files to download will be set in the section.

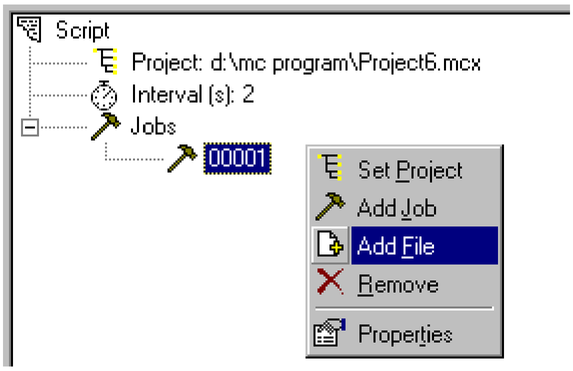
To start with, at the script window, click on the right mouse button, and select **Add Job**.



Then, an icon with a job number will appear under the **Jobs** branches.

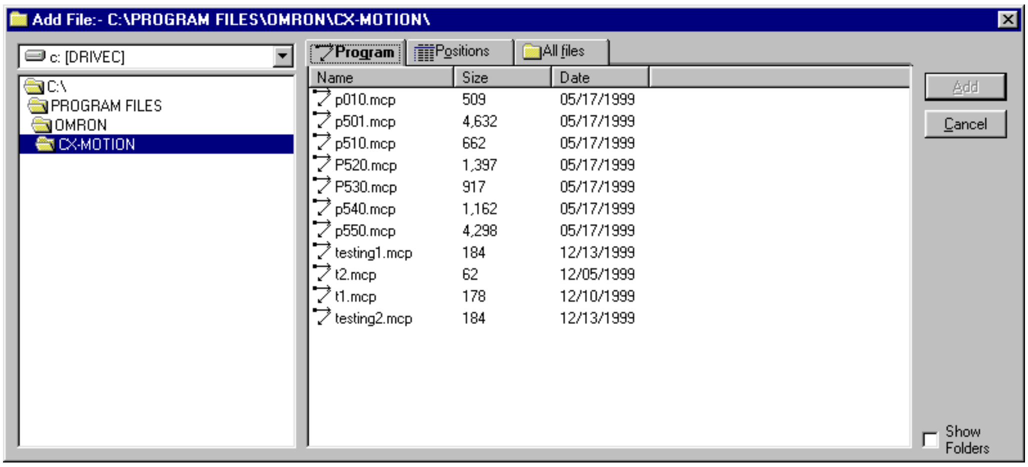


At the Jobs number, set the files to download. Click on the right mouse key, and select **Add File**.



Script
Project: d:\mc program\Project6.mcx
Interval (s): 2
Jobs
00001

From the **Add File** window, select the desire file.



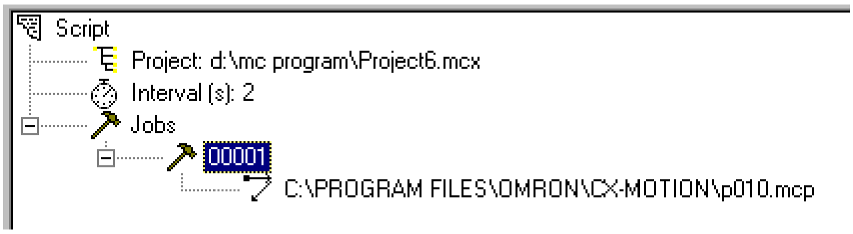
Add File:- C:\PROGRAM FILES\OMRON\CX-MOTION\

c: [DRIVEC]

C:\
PROGRAM FILES
OMRON
CX-MOTION

Name	Size	Date
p010.mcp	509	05/17/1999
p501.mcp	4,632	05/17/1999
p510.mcp	662	05/17/1999
P520.mcp	1,397	05/17/1999
P530.mcp	917	05/17/1999
p540.mcp	1,162	05/17/1999
p550.mcp	4,298	05/17/1999
testing1.mcp	184	12/13/1999
t2.mcp	62	12/05/1999
t1.mcp	178	12/10/1999
testing2.mcp	184	12/13/1999

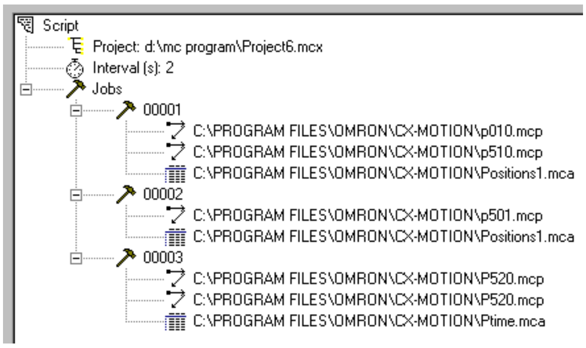
Then, click on **Add**. The path of the file will be shown.



Script
Project: d:\mc program\Project6.mcx
Interval (s): 2
Jobs
00001
C:\PROGRAM FILES\OMRON\CX-MOTION\p010.mcp

To add Jobs and files to each the Jobs task, repeat the steps as shown above.

Note: Only one position data file can be set per task.



Script
Project: d:\mc program\Project6.mcx
Interval (s): 2
Jobs
00001
C:\PROGRAM FILES\OMRON\CX-MOTION\p010.mcp
C:\PROGRAM FILES\OMRON\CX-MOTION\p510.mcp
C:\PROGRAM FILES\OMRON\CX-MOTION\Positions1.mca
00002
C:\PROGRAM FILES\OMRON\CX-MOTION\p501.mcp
C:\PROGRAM FILES\OMRON\CX-MOTION\Positions1.mca
00003
C:\PROGRAM FILES\OMRON\CX-MOTION\P520.mcp
C:\PROGRAM FILES\OMRON\CX-MOTION\P520.mcp
C:\PROGRAM FILES\OMRON\CX-MOTION\Ptime.mca

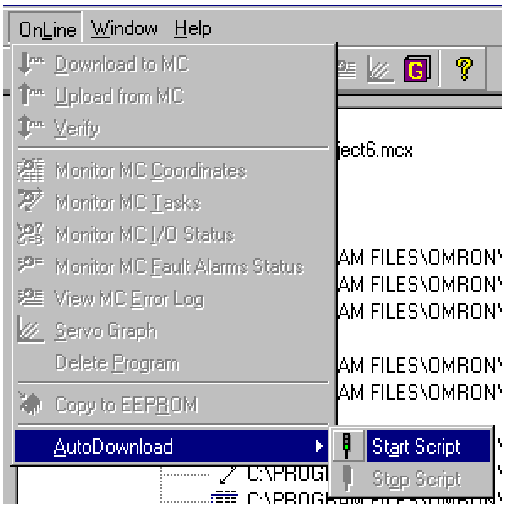
4-1-2 Autodownload

After the creation of the script file, connect the computer with CX-Motion with the MC unit, then RUN the script. It is only possible to run one script at a time, although the script can process many MC units.

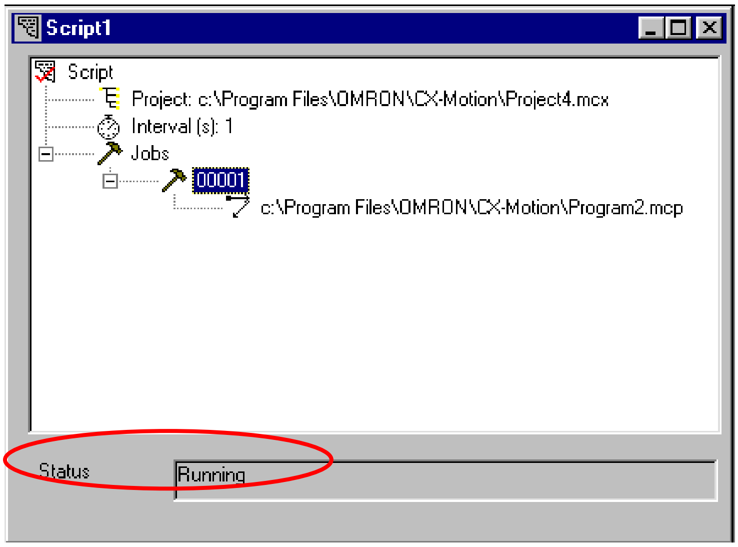
While the script is running, it is still possible to use the main editing and monitoring features of CX-Motion. However, that some online functions may perform slower than normal as the communications connection is now processing and passing more information, thus causing the polling rate between CX-Motion and the MC units to be slower than expected. (Example: System is being used to monitor an MC unit.)

To run the script, followed the steps as shown.

Select **OnLine - AutoDowload - Start Script**



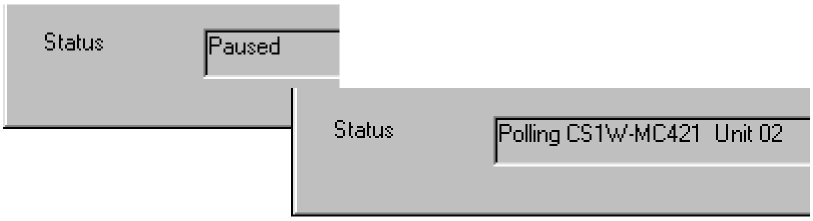
The process can be monitor by the status bar at the **Script** Window.


Script1

Script
- Project: c:\Program Files\OMRON\CX-Motion\Project4.mcx
- Interval (s): 1
- Jobs
  - 00001
    - c:\Program Files\OMRON\CX-Motion\Program2.mcp

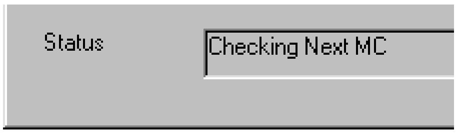
Status Running

At an interval of **1**second, the CX-Motion will poll the specified MC unit.


Status Paused

Status Polling CS1W-MC421 Unit 02

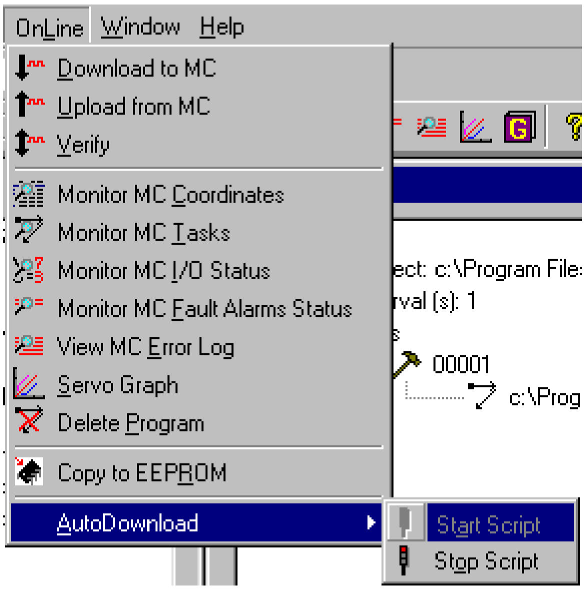
After the specified timeout time, CX-Motion will poll for the next MC unit.



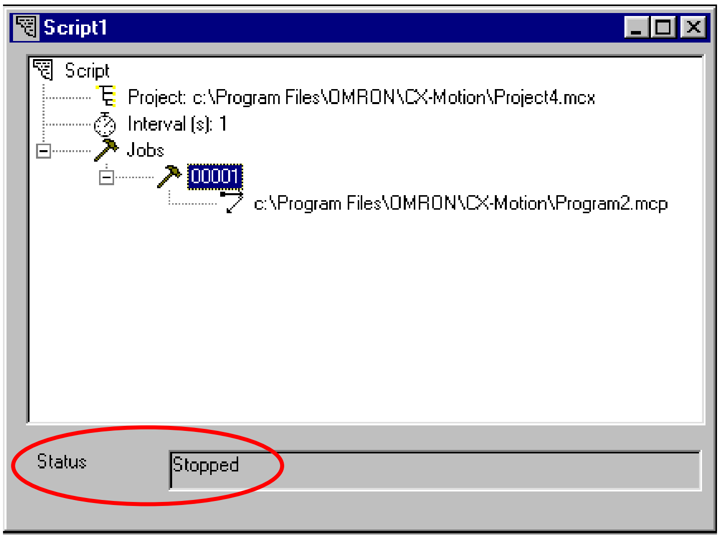
Status: Checking Next MC

To stop the Script, please proceed will the steps as shown.

Select **OnLine** - **AutoDownload** - **Stop Script**.



The **Status** bar at the Script window will display **Stopped**.



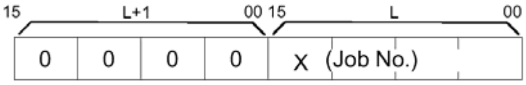
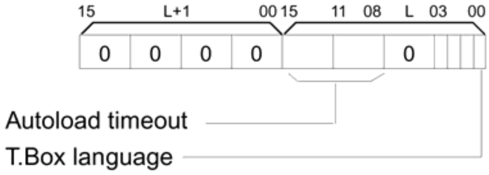
Script
Project: c:\Program Files\OMRON\CX-Motion\Project4.mcx
Interval (s): 1
Jobs
00001
c:\Program Files\OMRON\CX-Motion\Program2.mcp

Status: Stopped

4-2 Executing Automatic Loading

To execute automatic downloading, set one of the following addresses for the control code (C) of the IOWR instruction and then transfer the job number to the MC Unit.

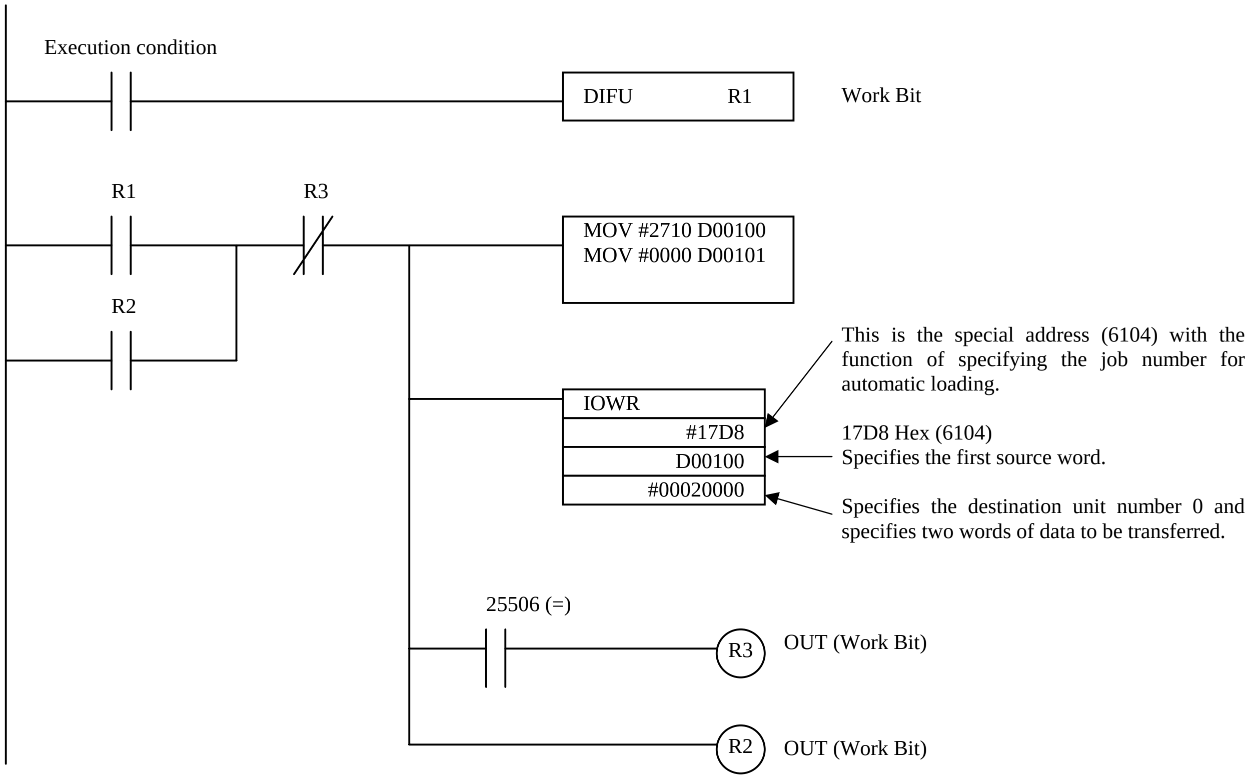
4-2-1 Parameter to access

Address		R/W	Name	Configuration
MC221	MC421			
17D8 (6104)	17D8 (6104)	R/W	Specifying automatic loading	 15 L+1 00 15 L 00 0 0 0 0 X (Job No.) This parameter is used when batch downloading programs and position data to an MC unit from the external memory device of a personal computer with CX-Motion. X (Job No.) Specifies a job number within a range from 0001 to 2710 Hex (1 to 10,000). The following contents can be read from this address using an IORD instruction. X= 0000: Power supply ON, automatic loading completed normally X= 0001 to 2710 Hex (1 to 10,000): Specified jib number or executing job number.
0FA9 (4009)	0FA9 (4009)	R	Teaching Box language/autoload timeout	 15 L+1 00 15 11 08 L 03 00 0 0 0 0 0 Autoload timeout T.Box language T.Box language Specifies whether the T.Box message is displayed in English or Japanese. (Only bit 0 is valid) X=0: English, X=1: Japanese Autoload Timeout Specifies whether the time up time is displayed when the automatic loading function is used. Range: 00 to B4 Hex (0 to 180s) Time up will not be monitored if 00 is set.

Note Programs and position data will not be properly downloaded to the MC Unit if the personal computer cable is disconnected or CX-Motion downloading is interrupted during execution of the automatic downloading function. If that occurs, perform the operation again.

Ladder Programming example In this example, job number 100 is specified. The MC Unit is mounted to a CS1 PC and assigned unit number 0, and the job number to be transferred (job number 100) is stored in words D00100 and D00101. Both the IOWR operands and the data to be transferred are assumed to be normal.

When debugging, check the Error Flags from the ladder program.


Execution condition
DIFU R1
Work Bit
R1
R3
R2
MOV #2710 D00100
MOV #0000 D00101
This is the special address (6104) with the function of specifying the job number for automatic loading.
IOWR
#17D8
D00100
#00020000
17D8 Hex (6104)
Specifies the first source word.
Specifies the destination unit number 0 and specifies two words of data to be transferred.
25506 (=)
R3
OUT (Work Bit)
R2
OUT (Work Bit)

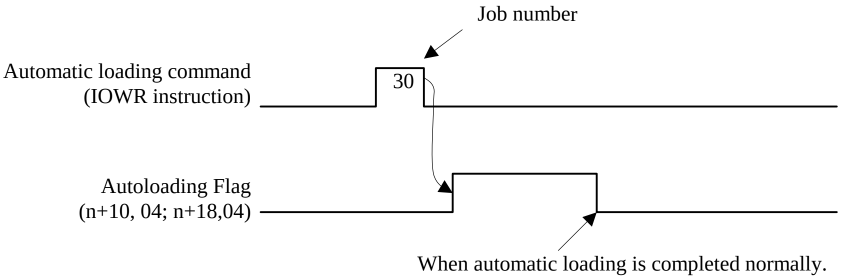
Transfer Data

D00100	0064	← Job Number: 100
D00101	0000	

4-2-2 Related Bits in the PC Interface Area

Bit Name	Model	Words	Bit	Decription
Autoloading Error Reset Bit	MC421 MC221	n n	05	↑: Error reset for automatic loading
Autoloading Flag	MC421 MC221	n+10 n+18	04	↑:Automatic loading begun ↓:Automatic loading finished
Autoloading Error Flag	MC421 MC221	n+10 n+18	05	↑:Error occurred during automatic loading ↓:Automatic loading error reset turned ON.

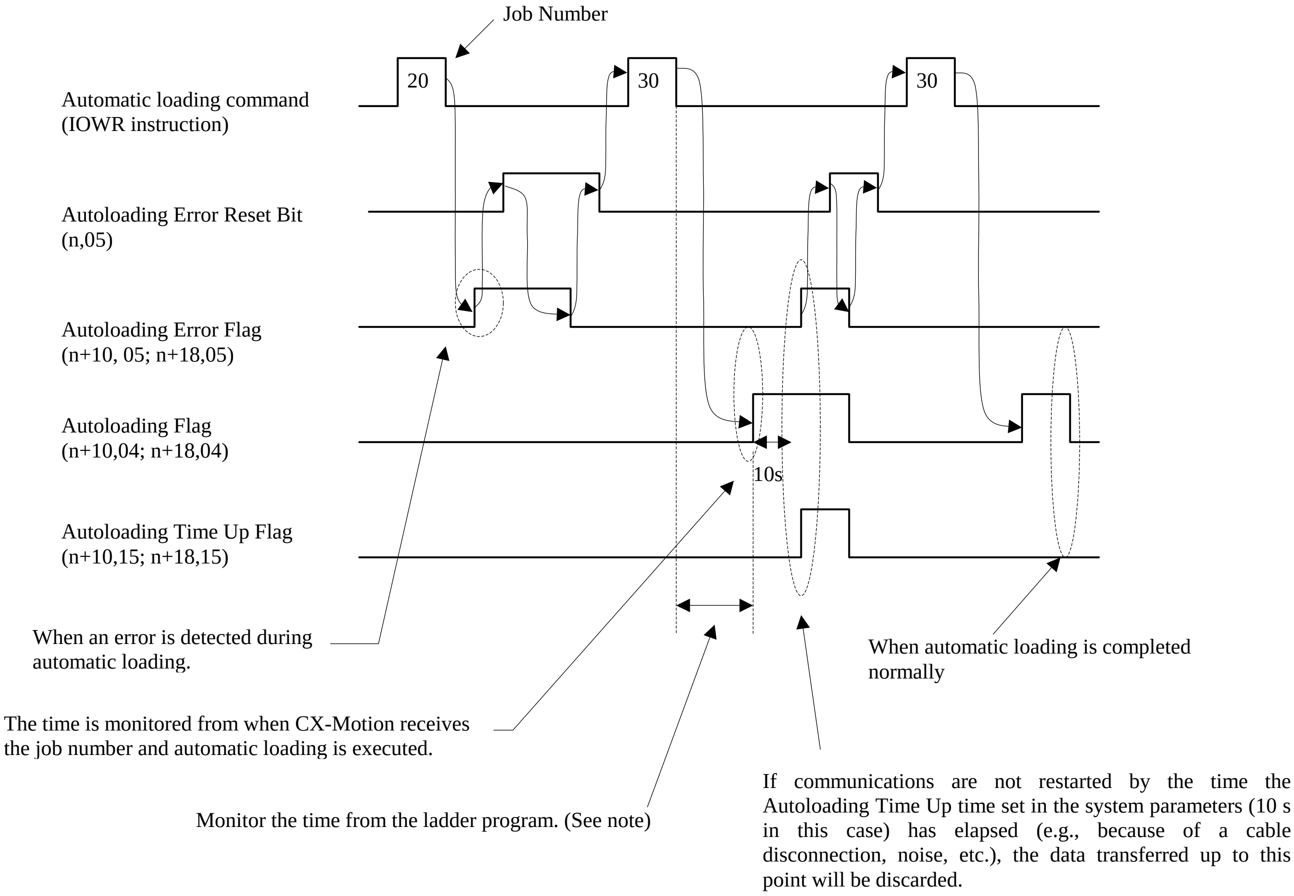
Autoloading Flag This flag turns ON when the data for the specified job number begins to be downloaded as the result of executing the IOWR instruction. It turns OFF when all of the data for that job number has been downloaded.


Job number
Automatic loading command
(IOWR instruction)
30
Autoloading Flag
(n+10, 04; n+18,04)
When automatic loading is completed normally.

Autoloading Error Flag and Error Reset Bit

The Autoloading Error Flag turns ON when an error occurs during automatic loading. When the Autoloading Error Reset Bit is turned ON, the Autoloading Error Bit will turn OFF. Unless the cause of the error has been cleared, however, the Autoloading Error Flag will turn back ON again the next time the automatic loading is executed.

If communications are not restarted by the time the Autoloading Time Up set in the unit parameters has elapsed, an error will occur and the Autoloading Error Flag will turn ON. If that occurs, the data transferred up to that point will be discarded.


Job Number
20
30
30
Automatic loading command
(IOWR instruction)
Autoloading Error Reset Bit
(n,05)
Autoloading Error Flag
(n+10, 05; n+18,05)
Autoloading Flag
(n+10,04; n+18,04)
10s
Autoloading Time Up Flag
(n+10,15; n+18,15)
When an error is detected during automatic loading.
When automatic loading is completed normally
The time is monitored from when CX-Motion receives the job number and automatic loading is executed.
Monitor the time from the ladder program. (See note)
If communications are not restarted by the time the Autoloading Time Up time set in the system parameters (10 s in this case) has elapsed (e.g., because of a cable disconnection, noise, etc.), the data transferred up to this point will be discarded.

Note: The MC Unit does not monitor the time from when it receives the automatic loading command until the Autoloading Flag turns ON. There may be cases where the power to the personal computer is turned OFF or a cable is disconnected, so monitor this time from the ladder program.

Section 5
Monitoring Function

This section describes the monitoring utilities provided by the CX-Motion.

5-1 Monitoring Function

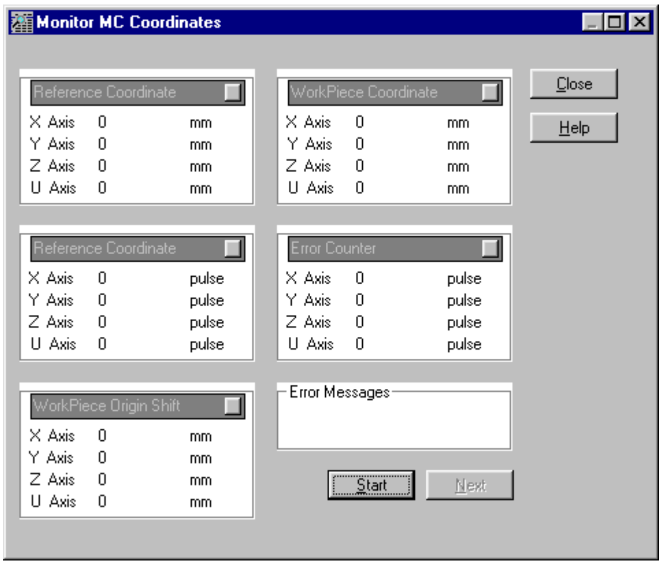
The following Monitoring Functions are provided by the CX-Motion.

<table>
<tr><th>Window Name</th><th colspan="2">Description</th></tr>
<tr><td>MC Coordinates Displays</td><td colspan="2">The following information will be shown:
▪ Reference Coordinate (In pulse)
▪ Reference Coordinate (As selected in the Machine Parameter - Motor Setting - Display Unit.)
▪ WorkPiece Coordinate
▪ WorkPiece Origin Shift
▪ Error Counter
▪ Error Messages</td></tr>
<tr><td>Monitor MC Tasks</td><td colspan="2">The following information will be shown:
▪ Executing Program
▪ Task Status
▪ Error Messages
▪ Axis Monitoring</td></tr>
<tr><td rowspan="2">I/O status Display</td><td colspan="2">ON/OFF status of Input and Output</td></tr>
<tr><td>Input
▪ General Input
▪ Origin Proximity
▪ CW / CCW Limit
▪ Emergency Stop Input
▪ Alarm Input</td><td>Output
▪ General Output/Break Signal Output
▪ Memory Run
▪ Memory Run Stopped
▪ Run Command Output
▪ Alarm Reset Output
▪ Sensor ON Output</td></tr>
<tr><td>MC FAL Status Display</td><td colspan="2">MC unit Error Display
▪ MC FAL
▪ Task FAL
▪ Axis FAL</td></tr>
<tr><td>MC Error Log</td><td colspan="2">The 20 error that occur most recently will be displayed. (If more that 20 error had occur the old error will be deleted.)</td></tr>
<tr><td>Servo Graph Input</td><td colspan="2">Data Trace and General input trace.</td></tr>
</table>

5-1-1 Monitor Coordinates

To monitor the present values of coordinates, please proceed with the steps as shown.

Click on , the following window will appear.



Monitor MC Coordinates

Reference Coordinate		
X Axis	0	mm
Y Axis	0	mm
Z Axis	0	mm
U Axis	0	mm

WorkPiece Coordinate		
X Axis	0	mm
Y Axis	0	mm
Z Axis	0	mm
U Axis	0	mm

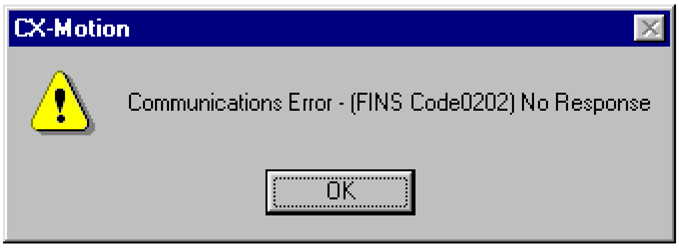
Reference Coordinate		
X Axis	0	pulse
Y Axis	0	pulse
Z Axis	0	pulse
U Axis	0	pulse

Error Counter		
X Axis	0	pulse
Y Axis	0	pulse
Z Axis	0	pulse
U Axis	0	pulse

WorkPiece Origin Shift		
X Axis	0	mm
Y Axis	0	mm
Z Axis	0	mm
U Axis	0	mm

Error Messages

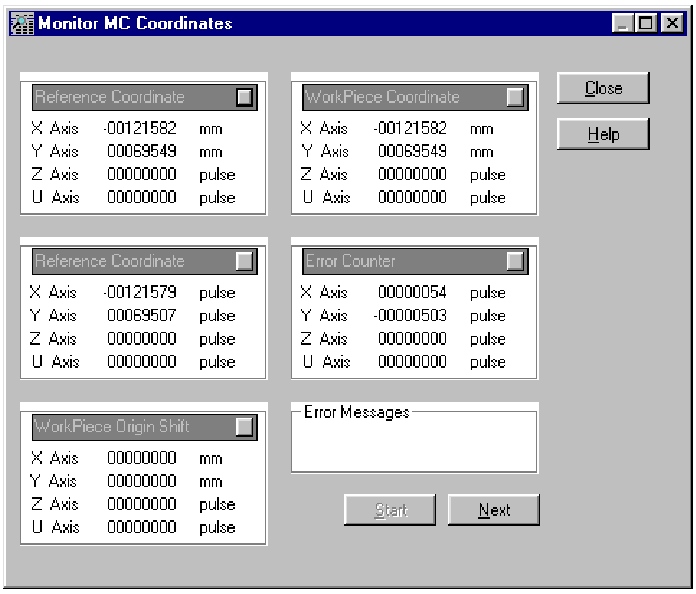
The coordinates displayed represent the respective titled at the boxes.
To start monitoring, click on **Start**. When the icon is click, the CX-Motion will tried to go ONLINE. If communication error occurs, the following dialog box will appear.



CX-Motion

Communications Error - (FINS Code0202) No Response

Else the status of the coordinates will be shown.



Monitor MC Coordinates

Reference Coordinate		
X Axis	-00121582	mm
Y Axis	00069549	mm
Z Axis	00000000	pulse
U Axis	00000000	pulse

WorkPiece Coordinate		
X Axis	-00121582	mm
Y Axis	00069549	mm
Z Axis	00000000	pulse
U Axis	00000000	pulse

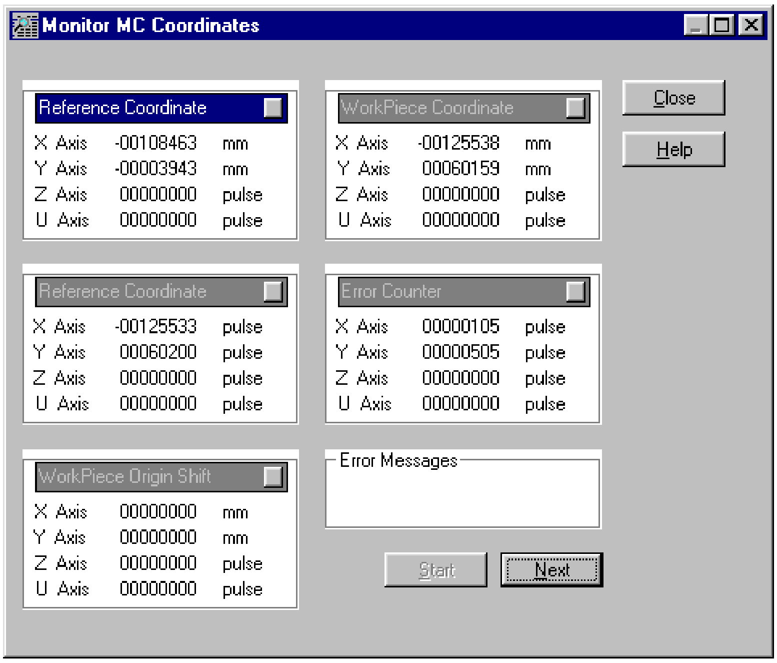
Reference Coordinate		
X Axis	-00121579	pulse
Y Axis	00069507	pulse
Z Axis	00000000	pulse
U Axis	00000000	pulse

Error Counter		
X Axis	00000054	pulse
Y Axis	-00000503	pulse
Z Axis	00000000	pulse
U Axis	00000000	pulse

WorkPiece Origin Shift		
X Axis	00000000	mm
Y Axis	00000000	mm
Z Axis	00000000	pulse
U Axis	00000000	pulse

Error Messages

To monitor individual box, click on the next button. Only the highlighted box will have its coordinates updated. To select the other boxes, click on **Next**.



Reference Coordinate		
X Axis	-00108463	mm
Y Axis	-00003943	mm
Z Axis	00000000	pulse
U Axis	00000000	pulse

WorkPiece Coordinate		
X Axis	-00125538	mm
Y Axis	00060159	mm
Z Axis	00000000	pulse
U Axis	00000000	pulse

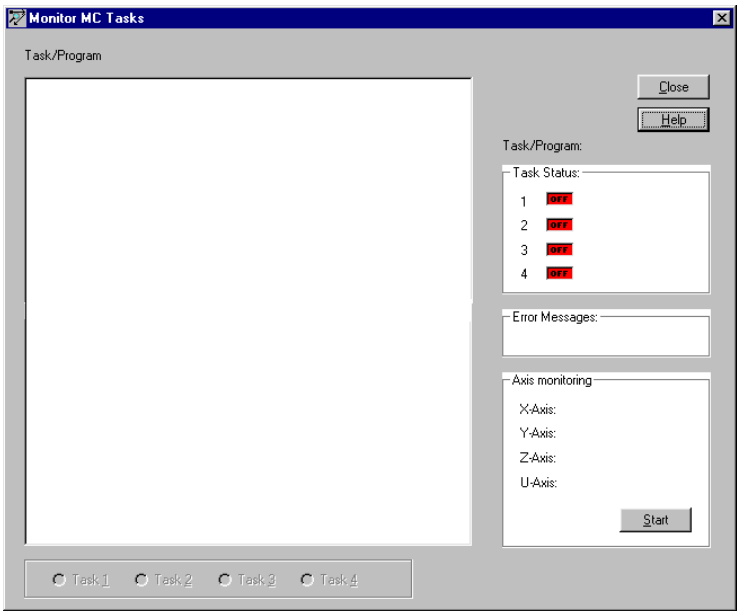
Reference Coordinate		
X Axis	-00125533	pulse
Y Axis	00060200	pulse
Z Axis	00000000	pulse
U Axis	00000000	pulse

Error Counter		
X Axis	00000105	pulse
Y Axis	00000505	pulse
Z Axis	00000000	pulse
U Axis	00000000	pulse

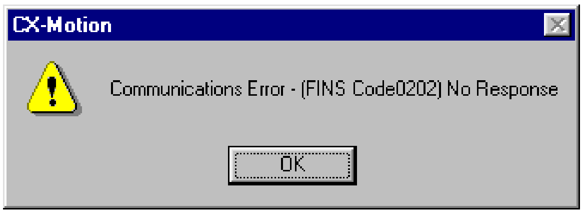
WorkPiece Origin Shift		
X Axis	00000000	mm
Y Axis	00000000	mm
Z Axis	00000000	pulse
U Axis	00000000	pulse

5-1-2 Monitor MC Task

To monitor the present status of the task, click on . The following window will appear on screen.

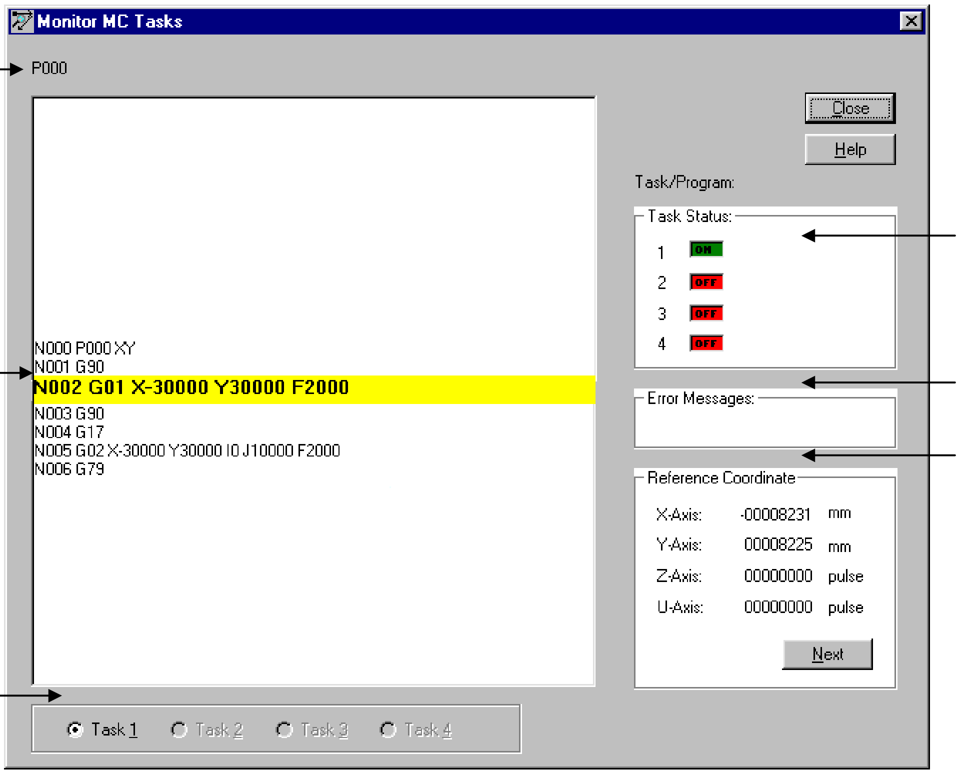


When the icon is click, the CX-Motion will try to connect to the PLC, if communication error occur, the following dialog window will appear.



Communications Error - (FINS Code0202) No Response

To start monitoring, click on the Start button at lower right hand corner of the Monitor MC Task window.



Present program number

P000

Present active block

```
N000 P000 XY
N001 G90
N002 G01 X-30000 Y30000 F2000
N003 G90
N004 G17
N005 G02 X-30000 Y30000 I0 J10000 F2000
N006 G79
```

Present active Task.

● Task 1 ○ Task 2 ○ Task 3 ○ Task 4

Task/Program:

Task Status:

1	ON
2	OFF
3	OFF
4	OFF

Present Active Task

Error Messages:

Error Occurred.

Reference Coordinate

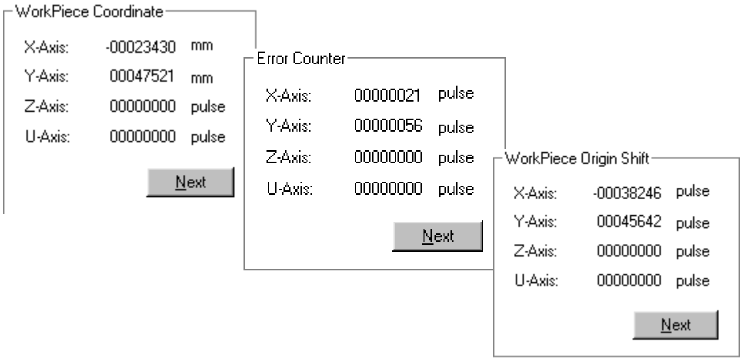
X-Axis:	-00008231	mm
Y-Axis:	00008225	mm
Z-Axis:	00000000	pulse
U-Axis:	00000000	pulse

Present Monitored Coordinate

At the bottom of the window, select the desire task to view (if more than one task is in used). At the **Tasks Status** section, the status of the task will be shown in correspond to ON/OFF. At the **Task/Program** section, the execution steps of the program will be shown, when the particular block number is being highlighted, it implies that the MC unit is executing that G code command.

To see more information, click on the Next button.

When the Next button is being click, users are able to view others information.



WorkPiece Coordinate

X-Axis:	-00023430	mm
Y-Axis:	00047521	mm
Z-Axis:	00000000	pulse
U-Axis:	00000000	pulse

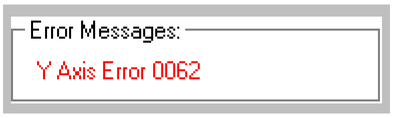
Error Counter

X-Axis:	00000021	pulse
Y-Axis:	00000056	pulse
Z-Axis:	00000000	pulse
U-Axis:	00000000	pulse

WorkPiece Origin Shift

X-Axis:	-00038246	pulse
Y-Axis:	00045642	pulse
Z-Axis:	00000000	pulse
U-Axis:	00000000	pulse

If error occurred while the task is processing, the error number will be shown at Error Messages.

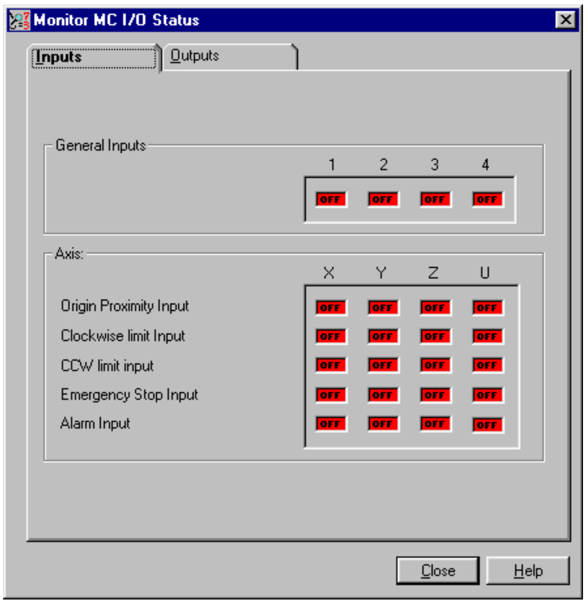


Error Messages:

Y Axis Error 0062

5-1-3 Monitor I/O status

To monitor the I/O status of the MC unit, click on . The following window will appear.



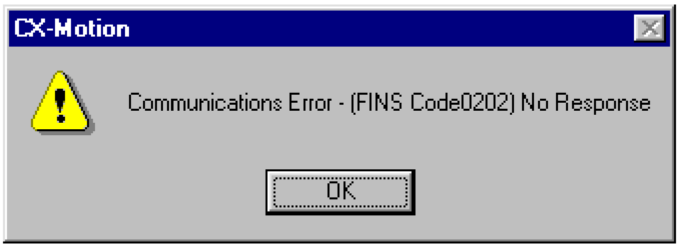
General Inputs

1	2	3	4
OFF	OFF	OFF	OFF

Axis:

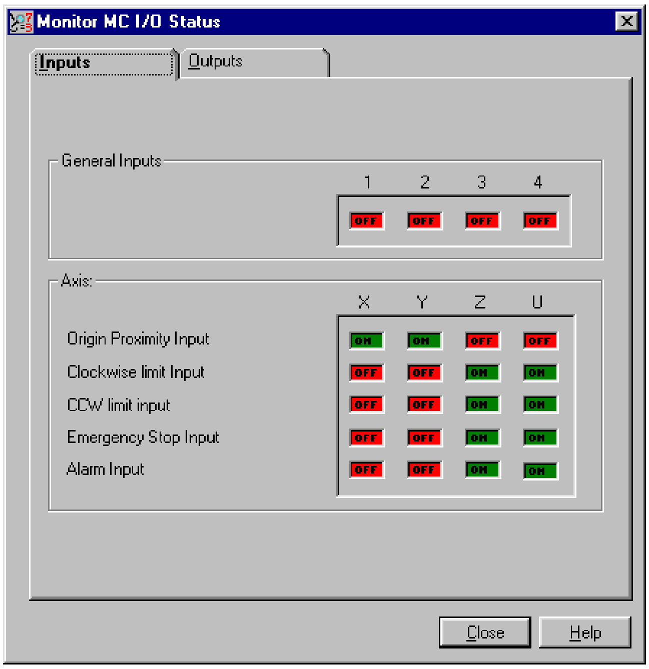
	X	Y	Z	U
Origin Proximity Input	OFF	OFF	OFF	OFF
Clockwise limit Input	OFF	OFF	OFF	OFF
CCW limit input	OFF	OFF	OFF	OFF
Emergency Stop Input	OFF	OFF	OFF	OFF
Alarm Input	OFF	OFF	OFF	OFF

When the icon is clicked, CX-Motion will try to communicate with the MC unit, however if communication error occur, the following window will appear.



Communications Error - (FINS Code0202) No Response

Else, status of the Input will be display in terms of ON/OFF (which is GREEN/RED)



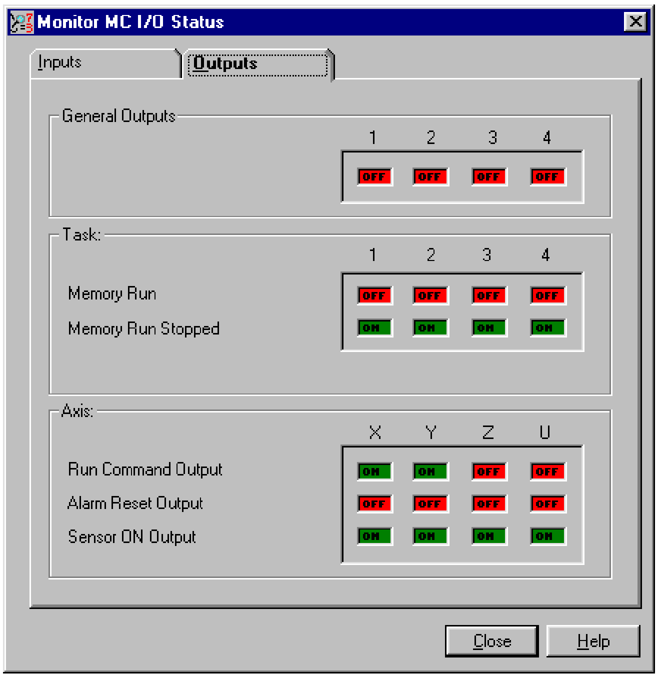
General Inputs

1	2	3	4
OFF	OFF	OFF	OFF

Axis:

	X	Y	Z	U
Origin Proximity Input	ON	ON	OFF	OFF
Clockwise limit Input	OFF	OFF	ON	ON
CCW limit input	OFF	OFF	ON	ON
Emergency Stop Input	OFF	OFF	ON	ON
Alarm Input	OFF	OFF	ON	ON

To view the Output status, click on the Output tab.



Monitor MC I/O Status

General Outputs

1	2	3	4
OFF	OFF	OFF	OFF

Task:

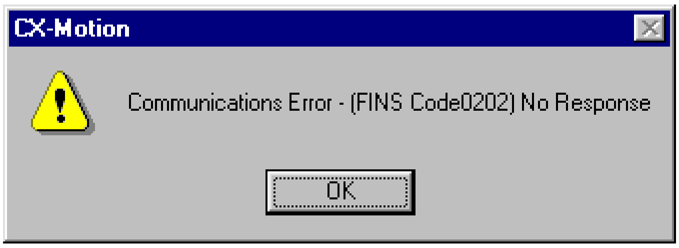
	1	2	3	4
Memory Run	OFF	OFF	OFF	OFF
Memory Run Stopped	ON	ON	ON	ON

Axis:

	X	Y	Z	U
Run Command Output	ON	ON	OFF	OFF
Alarm Reset Output	OFF	OFF	OFF	OFF
Sensor ON Output	ON	ON	ON	ON

5-1-4 MC FAL Status

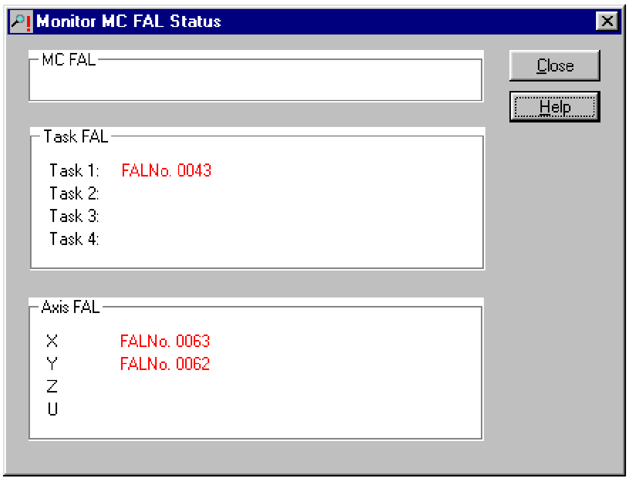
To view the present fault that occur on the MC unit, click on . The following window will appear. The CX-motion will try to go ONLINE with the MC unit when the icon is being click. If communication error occur, the following window will be shown.



CX-Motion

Communications Error - (FINS Code0202) No Response

Else the present fault status will be display.



Monitor MC FAL Status

MC FAL

Task FAL

Task 1: FALNo. 0043
Task 2:
Task 3:
Task 4:

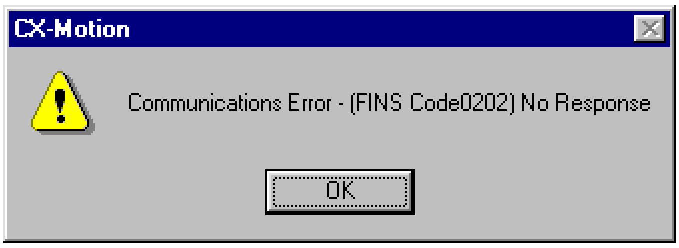
Axis FAL

X FALNo. 0063
Y FALNo. 0062
Z
U

5-1-5 View MC Error Log

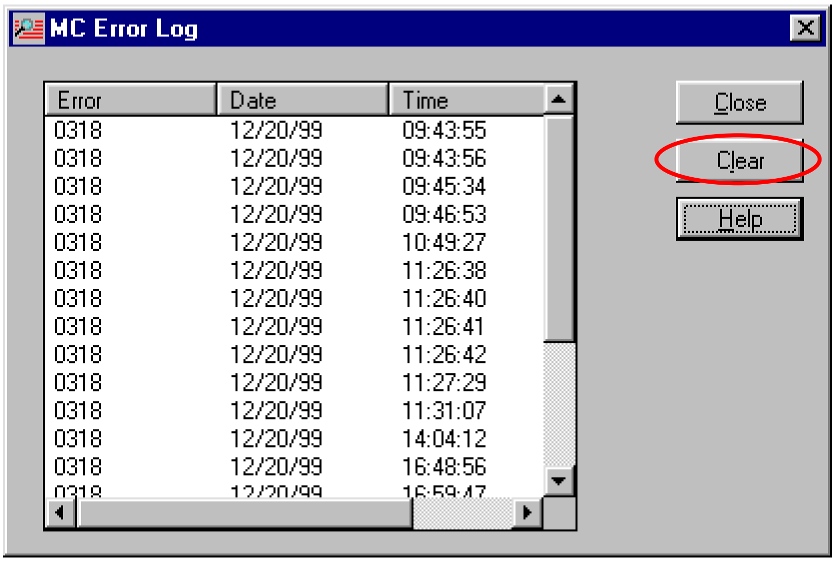
The CX-Motion provide error log viewer for user, The Error Log are able to display up to 20 error that had occurred. If more than 20 errors were accumulated, the old error will be discard.

To view the error log, click on . The CX-Motion will try to go ONLINE, hence if communication error occurs, the following window will appear.



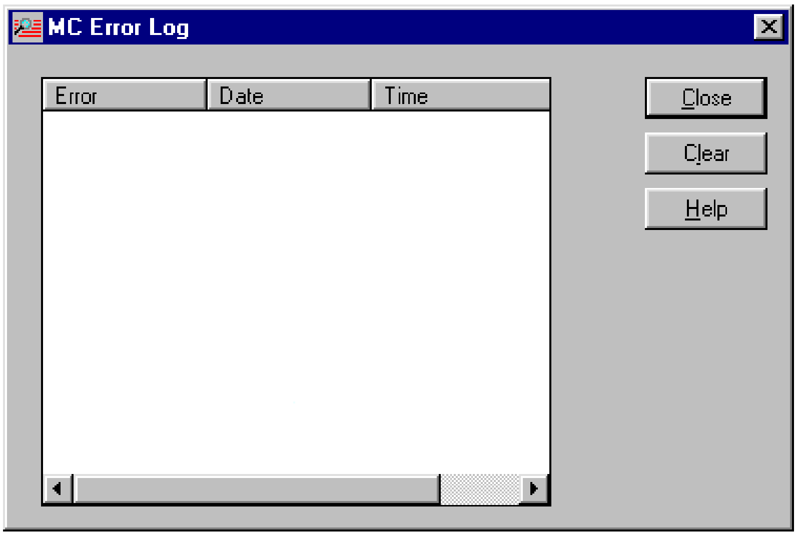
Communications Error - (FINS Code0202) No Response

Else the error log will be displayed.



Error	Date	Time
0318	12/20/99	09:43:55
0318	12/20/99	09:43:56
0318	12/20/99	09:45:34
0318	12/20/99	09:46:53
0318	12/20/99	10:49:27
0318	12/20/99	11:26:38
0318	12/20/99	11:26:40
0318	12/20/99	11:26:41
0318	12/20/99	11:26:42
0318	12/20/99	11:27:29
0318	12/20/99	11:31:07
0318	12/20/99	14:04:12
0318	12/20/99	16:48:56
0318	12/20/99	16:59:47

To clear the error log, click on the clear button.



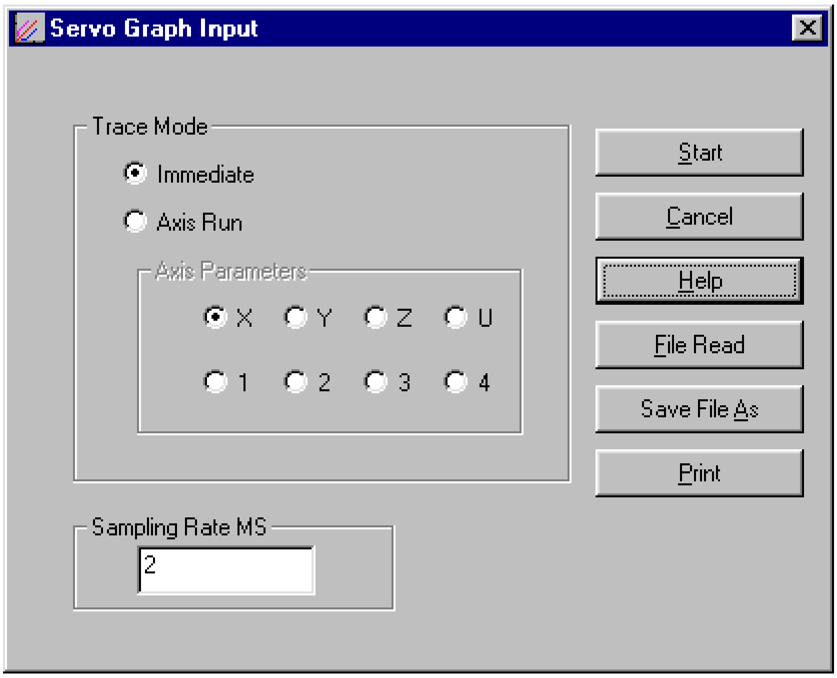
5-2 The Servo Graph

The servo graph monitoring functions are only provided in CS1W-MC421/221. With these functions, users are able to monitor servo data as shown:

For each axis:

Feedback:	Current, measured, velocity
Demand:	Demanded velocity
Error:	Error count

To start, click on , the **Servo Graph Input Window** will appear.



Trace Mode
● Immediate
○ Axis Run

Axis Parameters
● X ○ Y ○ Z ○ U
○ 1 ○ 2 ○ 3 ○ 4

Sampling Rate MS
2

To start monitoring, there are two possible methods to trigger:

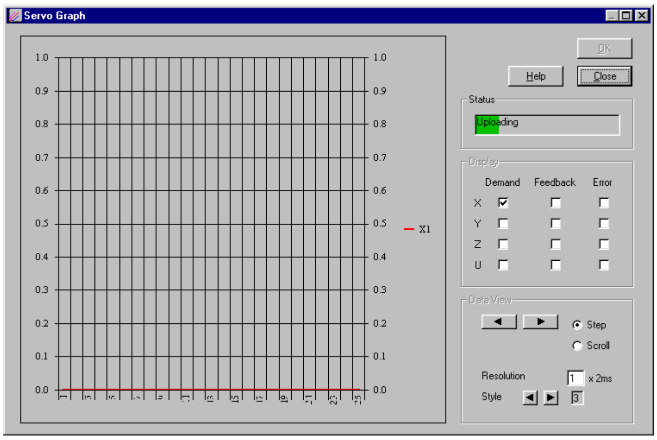
- Immediate
  That is, once the **Start** button has been clicked, the data trace will start.
- Axis Run
  Only when the selected axis or general input goes high, then the data trace will start.

It is also possible to specify the sampling interval in milliseconds (between 2 and 100ms).

After configuring the trigger type, click on **Start**.

5-2-1 Immediate

If immediate is selected, the following window will be shown.

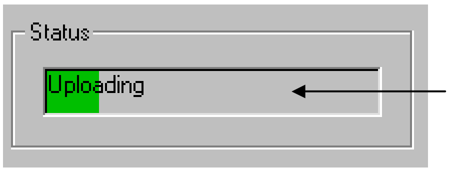


Status
Uploading

Display

	Demand	Feedback	Error
X	☑	□	□
Y	□	□	□
Z	□	□	□
U	□	□	□

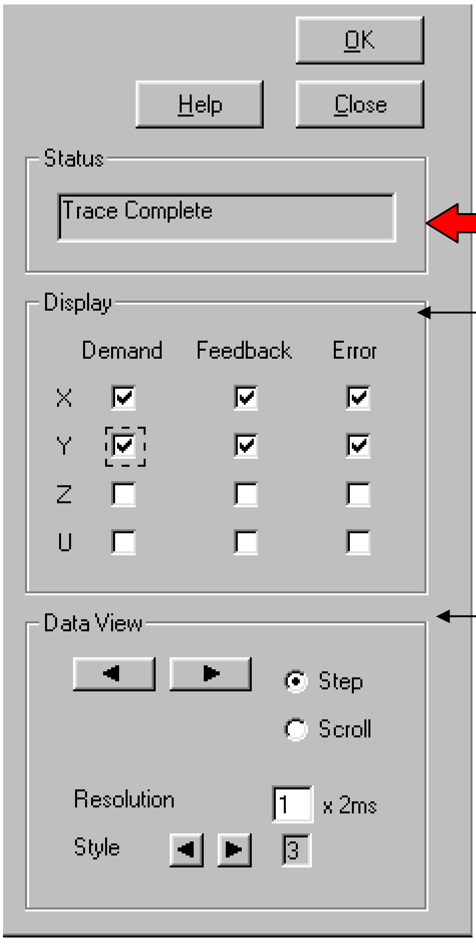
Data View
● Step
○ Scroll
Resolution 1 x 2ms
Style 3



Status: Uploading

Present status of the process.

When completed, the Status box will shown the process status.



Status: Trace Complete

Display

	Demand	Feedback	Error
X	☑	☑	☑
Y	☑	☑	☑
Z	□	□	□
U	□	□	□

Data View: ● Step ○ Scroll

Resolution 1 x 2ms

Style 3

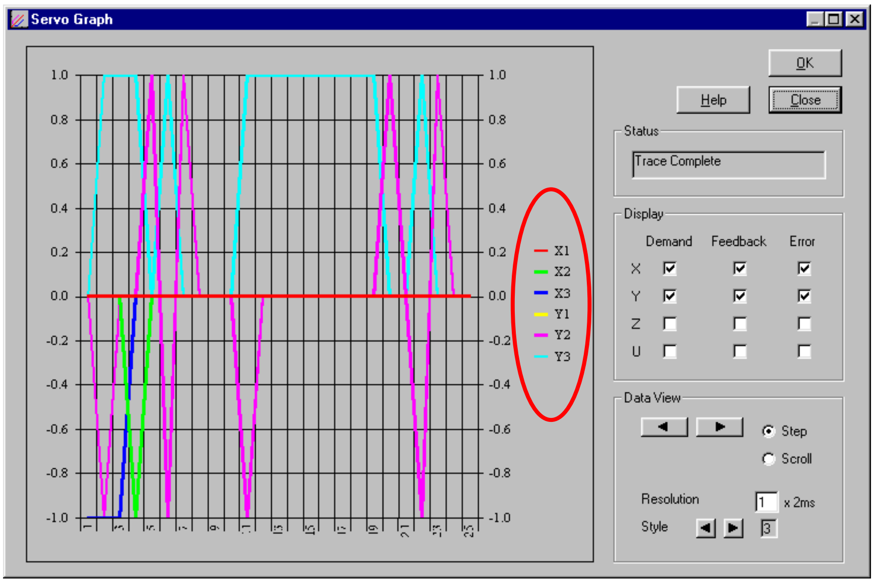
At the **Display** dialogue, select the information to be display in the graph.
The legend of the graph will be as followed:
Example X axis :

X1 : Demand
X2 : Feedback
X3 : Error

At the **Data View** dialogue, select the following:

- display style of the graph
  - maximum 10 styles
- the resolution of the graph
  - setting range 1 - 20.
- the navigation controls type
  - Step
  - Scroll

Upon configuration, click on **OK**. The graph will be display as configured.



Status: Trace Complete

Display

	Demand	Feedback	Error
X	☑	☑	☑
Y	☑	☑	☑
Z	□	□	□
U	□	□	□

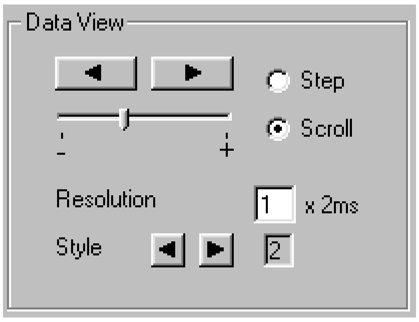
Data View: ● Step ○ Scroll

Resolution 1 x 2ms

Style 3

The legends of the graph are described as above.

If the scroll navigation tools are selected, the scroll bar will appear under the arrow button.



○ Step
● Scroll

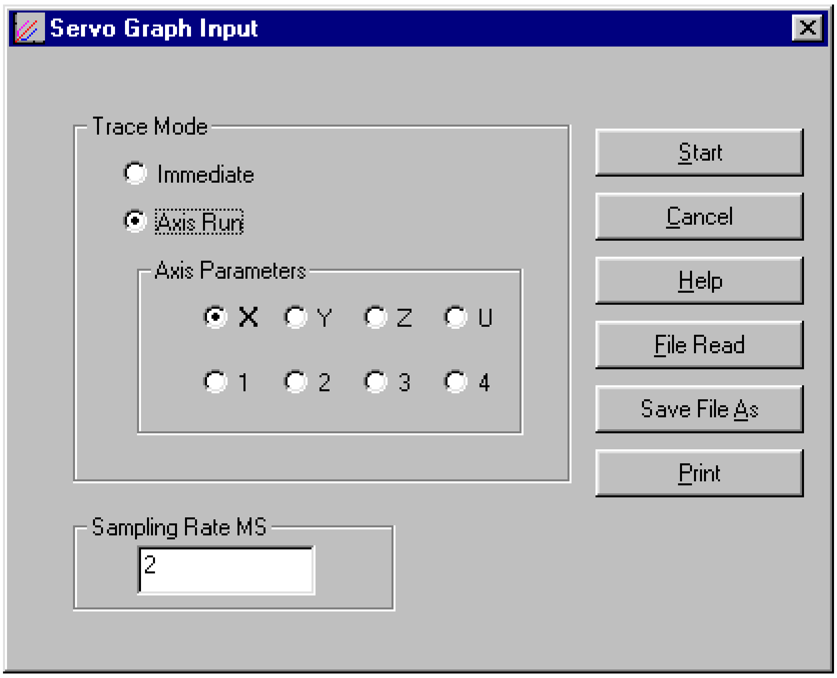
Resolution 1 x 2ms

Style 2

To scroll, click on the selector and move it <-> or <+>. To display continuous data, click on the arrow button, if ⇦ is selected, the graph will display data which go towards the <-> direction. If the ⇨ is selected, the graph will display data which go toward the <+> direction. With scroll navigation tools, a maximum of total of 500 trace data can be displayed. To stop continuous data display, click on **Step**.

5-2-2 Axis Run

When Axis Run is selected as trigger method, select the trigger source, and click on **Start**.



Trace Mode

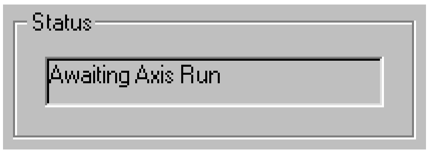
○ Immediate
● Axis Run

Axis Parameters

● X ○ Y ○ Z ○ U
○ 1 ○ 2 ○ 3 ○ 4

Sampling Rate MS 2

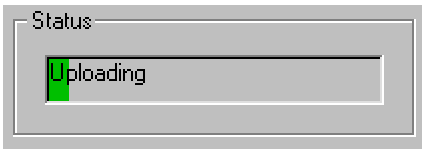
At the **Status** Bar the following information will be shown.



Status

Awaiting Axis Run

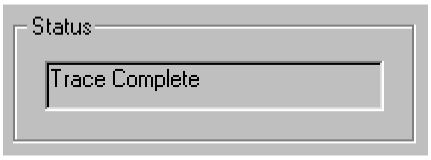
When the axis start running, the Status will show



Status

Uploading

Upon completion, the **Trace Complete** will be display at the status bar.



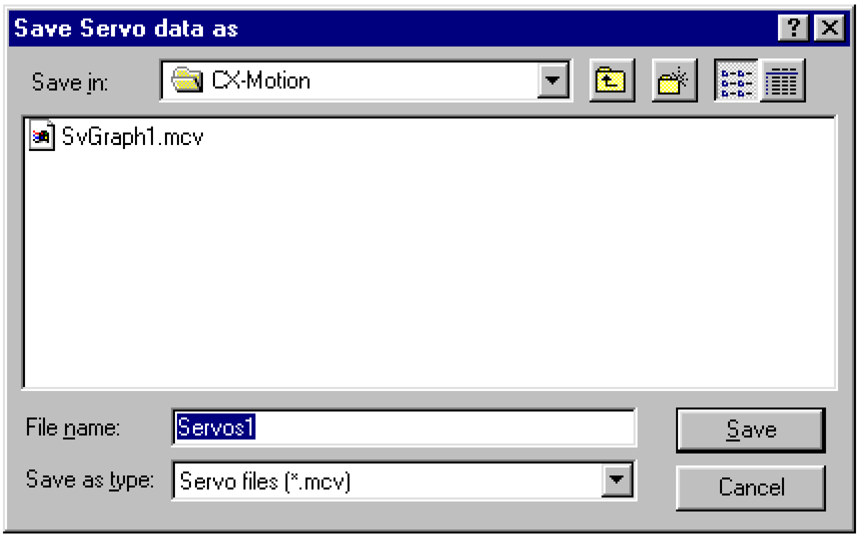
Trace Complete

To display the graph, the steps are the same as **Immediate** method.

5-2-3 Saving File

The trace file can be save for further analysis. To save the file followed the steps below.

At the Servo Graph window, click on the **Close** button. At the **Servo Graph Input**, click on **Save File as**. The following will appear.



Save in: CX-Motion

SvGraph1.mcv

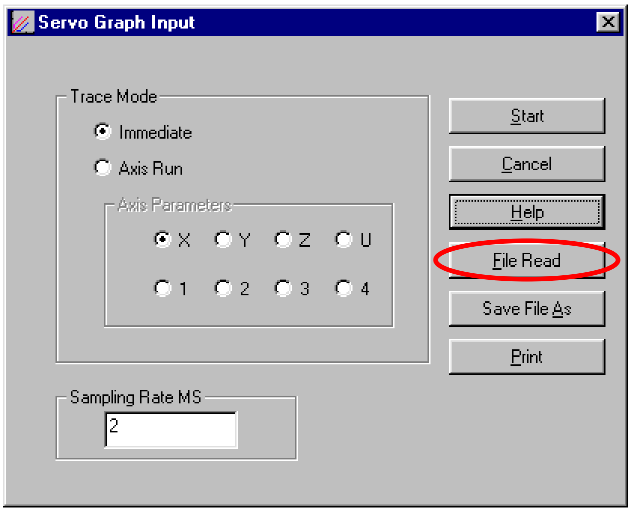
File name: Servos1

Save as type: Servo files (*.mcv)

The Servo files will be save as ***.mcv**.

5-2-4 Reading a save file

To retrieve the file for analysis, at the **Servo Graph Input** , click on the **File Read** button.



Trace Mode

● Immediate

○ Axis Run

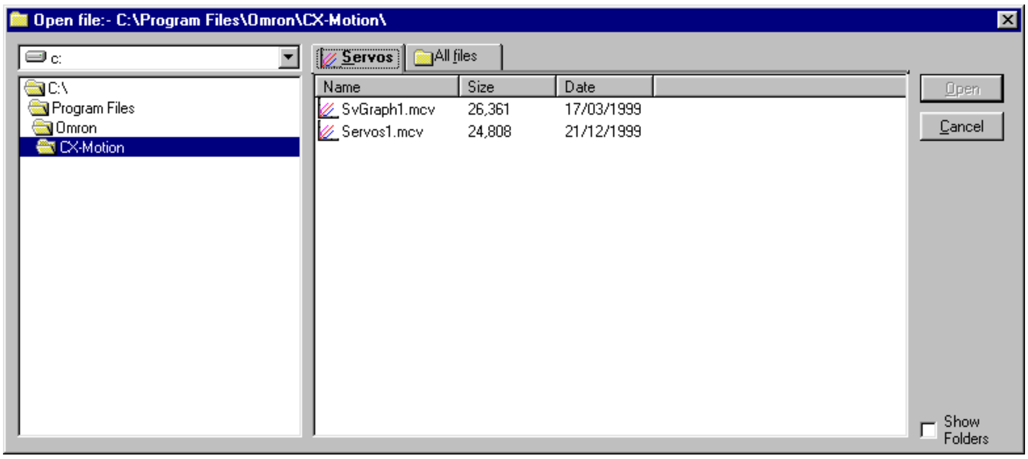
Axis Parameters

● X ○ Y ○ Z ○ U

○ 1 ○ 2 ○ 3 ○ 4

Sampling Rate MS: 2

Select the appropriate file to read.



Open file:- C:\Program Files\Omron\CX-Motion\

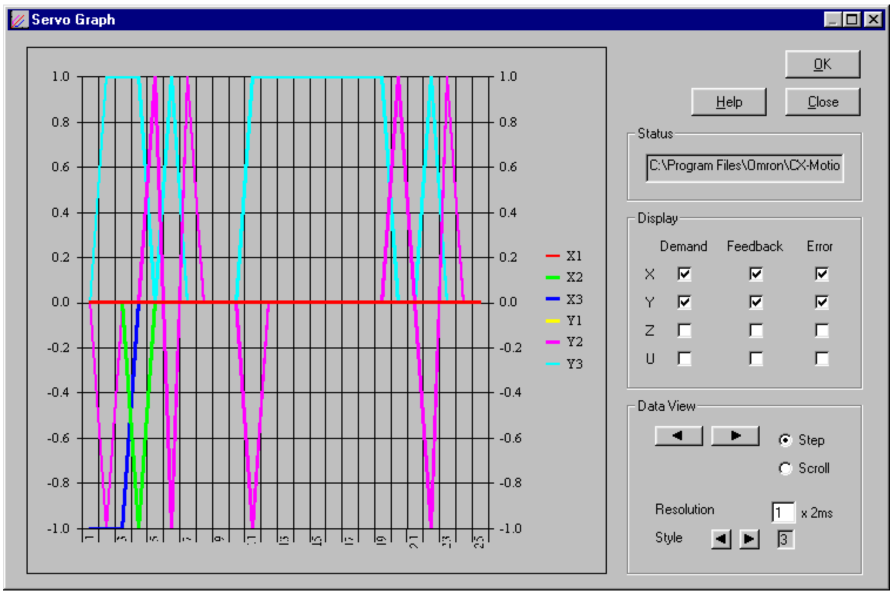
c:

C:\
Program Files
Omron
CX-Motion

Servos | All files

Name	Size	Date
SvGraph1.mcv	26,361	17/03/1999
Servos1.mcv	24,808	21/12/1999

Then, the data will be display in the **Servo Graph Window**.



Status: C:\Program Files\Omron\CX-Motio

Display

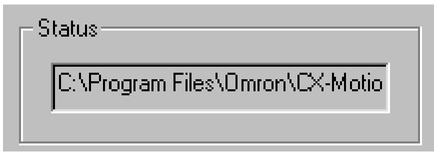
	Demand	Feedback	Error
X	☑	☑	☑
Y	☑	☑	☑
Z	□	□	□
U	□	□	□

Data View: ● Step ○ Scroll

Resolution 1 x 2ms

Style 3

The filename of the read file will be display at the **Status** Bar.



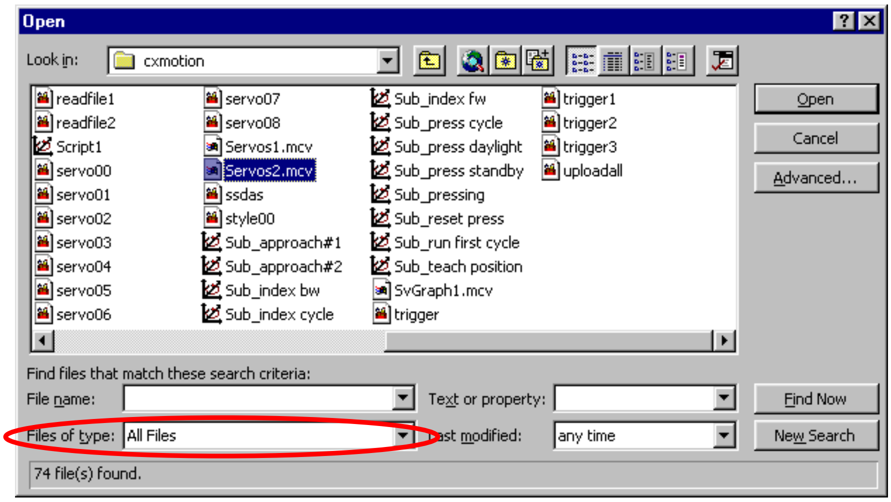
Status

C:\Program Files\Omron\CX-Motio

5-2-5 Retrieving file in Microsoft Excel

Servo Data can also be read in Microsoft Excel, the steps are as shown.

At the Microsoft Excel, perform a file open operation. At he Open File Window, select **All Files** at the **File of type**.



Open

Look in: cxmotion

Find files that match these search criteria:

File name:

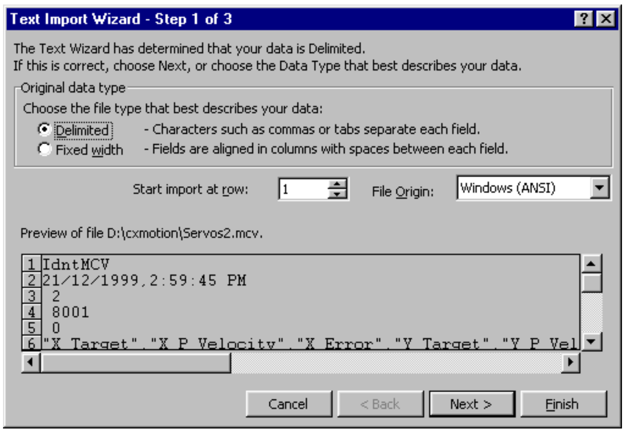
Text or property:

Files of type: All Files

Last modified: any time

74 file(s) found.

Select the desired file and click **Open**. The **Text Import Wizard - Step 1 of 3** window will appear.



Text Import Wizard - Step 1 of 3

The Text Wizard has determined that your data is Delimited.
If this is correct, choose Next, or choose the Data Type that best describes your data.

Original data type

Choose the file type that best describes your data:

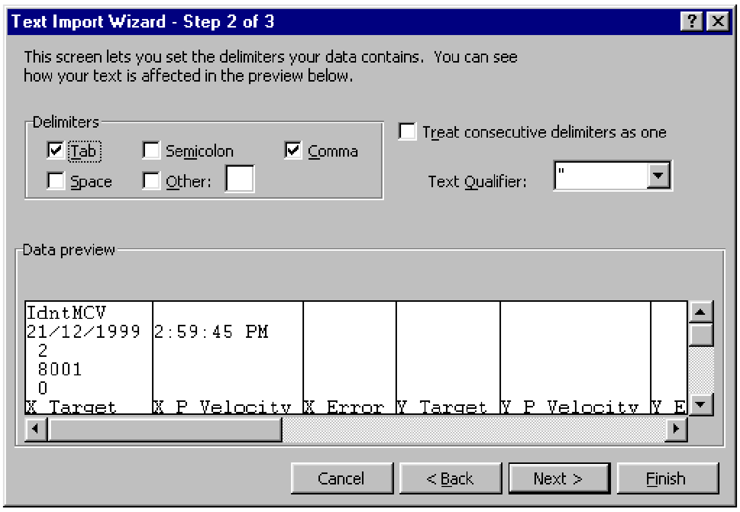
● Delimited - Characters such as commas or tabs separate each field.
○ Fixed width - Fields are aligned in columns with spaces between each field.

Start import at row: 1 File Origin: Windows (ANSI)

Preview of file D:\cxmotion\Servos2.mcv.

```
1 IdntMCV
2 21/12/1999,2:59:45 PM
3  2
4  8001
5  0
6 "X Target","X P Velocity","X Error","Y Target","Y P Vel
```

Click on **Next >** to continue. At the **Text Import Wizard - Step 2 of 3** window, select **Comma** & **Tab** as Delimiters and **"** as Text Qualifier.



Text Import Wizard - Step 2 of 3

This screen lets you set the delimiters your data contains. You can see how your text is affected in the preview below.

Delimiters

☑ Tab ☐ Semicolon ☑ Comma
☐ Space ☐ Other:

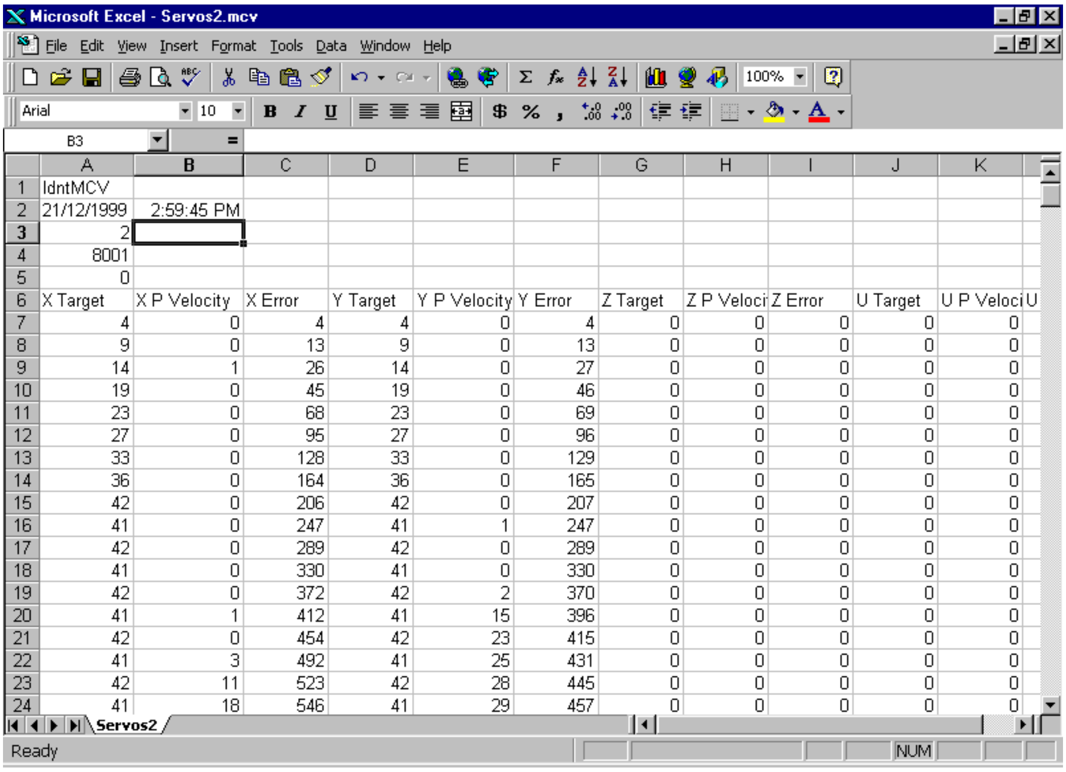
☐ Treat consecutive delimiters as one

Text Qualifier: "

Data preview

```
IdntMCV
21/12/1999 2:59:45 PM
 2
 8001
 0
X Target   X P Velocity X Error Y Target Y P Velocity Y E
```

Click **Next >** to proceed. At **Text Import Wizard - Step 3 of 3** window, click on **Finish** to complete. The servo data will be shown as below.


Microsoft Excel - Servos2.mcv
File Edit View Insert Format Tools Data Window Help
100%
Arial
10
B3
=
Servos2
Ready
NUM

	A	B	C	D	E	F	G	H	I	J	K	
1	IdntMCV											
2	21/12/1999	2:59:45 PM										
3	2											
4	8001											
5	0											
6	X Target	X P Velocity	X Error	Y Target	Y P Velocity	Y Error	Z Target	Z P Veloci	Z Error	U Target	U P Veloci	U
7	4	0	4	4	0	4	0	0	0	0	0	
8	9	0	13	9	0	13	0	0	0	0	0	
9	14	1	26	14	0	27	0	0	0	0	0	
10	19	0	45	19	0	46	0	0	0	0	0	
11	23	0	68	23	0	69	0	0	0	0	0	
12	27	0	95	27	0	96	0	0	0	0	0	
13	33	0	128	33	0	129	0	0	0	0	0	
14	36	0	164	36	0	165	0	0	0	0	0	
15	42	0	206	42	0	207	0	0	0	0	0	
16	41	0	247	41	1	247	0	0	0	0	0	
17	42	0	289	42	0	289	0	0	0	0	0	
18	41	0	330	41	0	330	0	0	0	0	0	
19	42	0	372	42	2	370	0	0	0	0	0	
20	41	1	412	41	15	396	0	0	0	0	0	
21	42	0	454	42	23	415	0	0	0	0	0	
22	41	3	492	41	25	431	0	0	0	0	0	
23	42	11	523	42	28	445	0	0	0	0	0	
24	41	18	546	41	29	457	0	0	0	0	0	

Section 6
Startup Guide

This section will give a basic guide of setting up a servo system, by using CX-Motion to configure the system parameter of the MC unit.

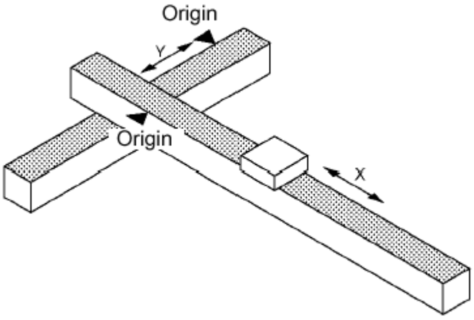
6-1 Getting started

This section explains the MC Unit startup procedures, taking the CS1W-MC221 as an example.

6-1-1 Overview

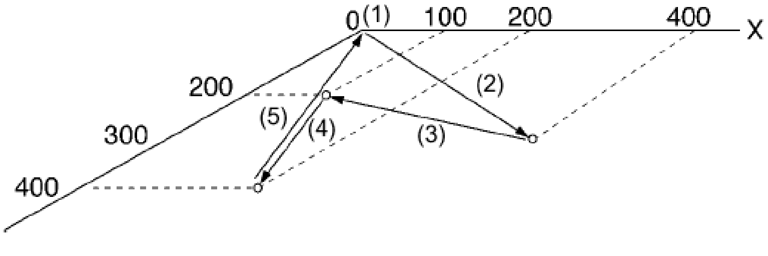
In this operation, a 2-axis mechanical system using the X and Y axes is used for positioning according to an MC program in the G-language executed in Automatic Mode, and for performing servo-lock, origin search, and jogging operations in Manual Mode.

Mechanical System



Automatic Mode Operation

The MC Unit executes the following positioning according to the MC program.



1, 2, 3...

1. Each axis waits at its reference origin (0, 0).
2. The X and Y axes are started to move the drill to a position where the first hole (400, 200) is to be drilled.
3. The robot moves to the next position (100, 200).
4. The robot moves to the last position (200, 400).
5. The robot returns to the reference origin (0, 0).

Manual Mode Operation

Servo-lock
If no other manual command is being executed and if the axes are not already servolocked, the X and Y axes will be servolocked.

Origin Search
If no other manual command is being executed and if the origin has not been defined yet, an origin search will be executed for the X and Y axes.

Jogging
If no other manual command is being executed and if the axes are servolocked, jogging will be executed along the X and Y axes.

6-1-2 Procedure

The procedure for performing these operations is as follows:

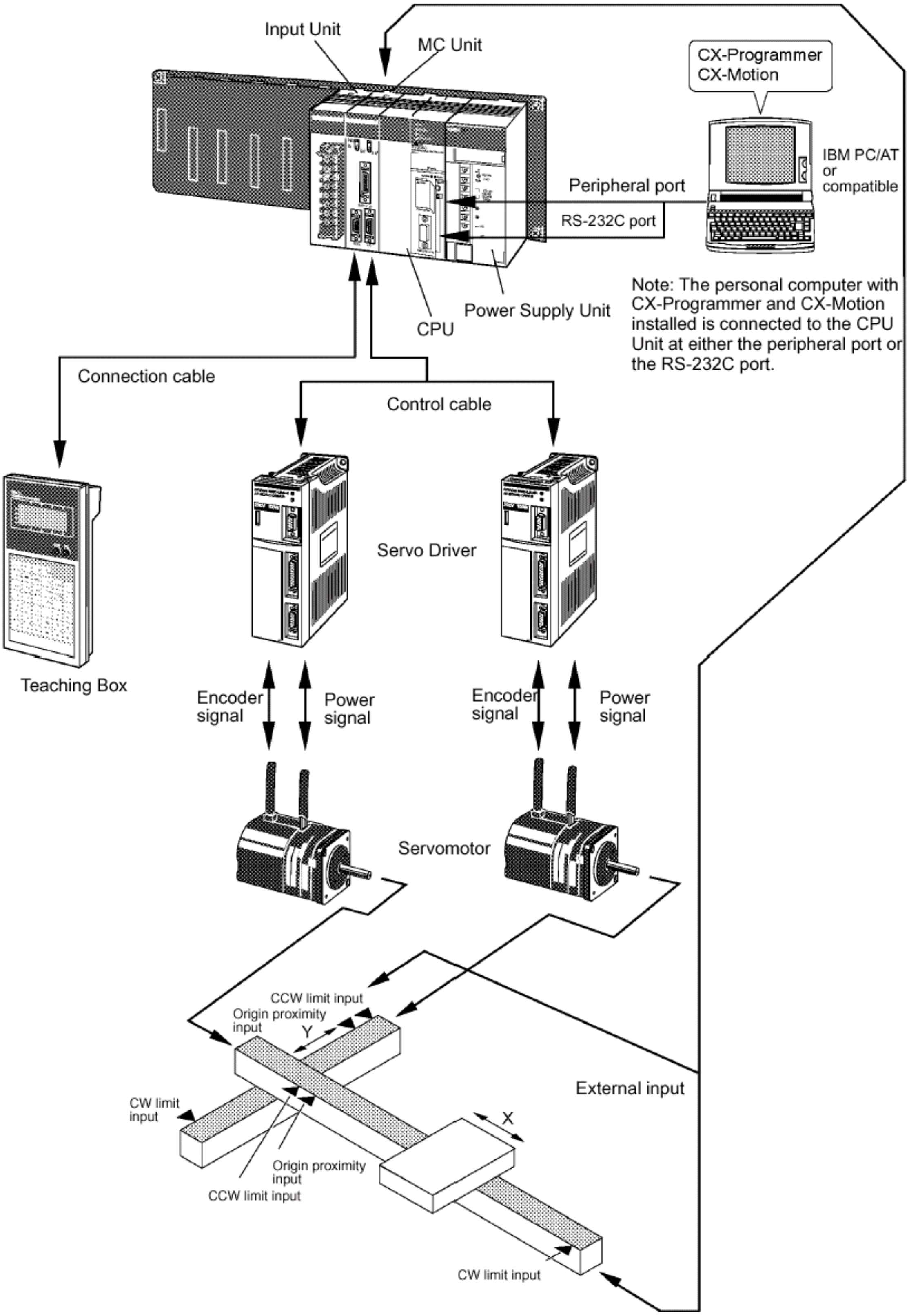
1, 2, 3...

1. Mount the MC Unit.
2. Set the MC Unit's unit number
3. Connect and wire all Units and devices.
   Connect the Programming Devices to the CPU Unit and a personal computer with CX-Programmer and CX-Motion installed. Connect the Teaching Box. Wire the MC Unit external inputs. Wire the Input Unit (for the switchbox), and wire the Servo Driver.
4. Power up the PC and use CX-Programmer or the Programming Console to create I/O tables.
5. Create a new project and add PC (CPU Unit) using CX-Motion.
6. 6. Add MC Unit to the project and set/save the system parameters using CX-Motion.
7. Create, compile, and save MC programs (G language), register in the project, and save the project using CX-Motion.
8. Transfer the system parameters and MC programs (G language) to the MC Unit and save in the flash memory using the CX-Motion.
9. Create a ladder program, transfer it to the CPU Unit, and save it.
10. Conduct trial operation. (Refer to 8)
    Using either CX-Motion or the Teaching Box, check the MC program in Automatic Mode and the servo-lock, origin search, and jogging operations in Manual Mode. Use CX-Motion for to monitor operation.
    When setting the system parameters and conducting the trial operation.

6-2 System Configuration and Wiring

6-2-1 System Configuration

Perform the operations using the following system configuration. Mount the MC Unit to the CPU Backplane and set it to unit number 0.



The model numbers for the Units and devices used in this configuration are provided in the following tables.

Control System

Name	Model
CPU Unit (Sysmac CS1-Series)	CS1H-CPU6_-E/CS1G-CPU4_-E
MC unit	CS1W-MC221
Input	C200H-ID212
Power Supply Unit	C200HW-PA204
CPU Backplane	CS1W-BC103
Power supply for input signals	DC power supply (+24 V)
Teaching Boc	CVM1-PRO01 + CVM1-MP702

Programming Devices

Name	Model
Personal Computer for Programming Devices	IBM PC/AT or compatible
Operating System	Windows 95/98/NT 4.0
CX-Programmer	WS02-CXPC1-E
CX-Motion	WS02-MCTC1-E

Servo Control System

Name	Model
Servo Drivers (two)	R88D-UA03L (for 100 VAC, 30 W) _ 2
Servomotors (two)	R88M-U03030L (for 100 VAC, 30 W) _ 2
Power Cables	R88A-CAU___S (for U Series)
Encoder Cable	R88A-CRU___C (for U Series)
Control Cable	R88A-CPU00_M1 (for U Series)

Note In this example operation, U series Servo Drives and Servomotors are used for the X and Y axes. To use H-series, M-series, or other Servo Drives and Servomotors, refer to the applicable manuals.

Special I/O cable and Terminals

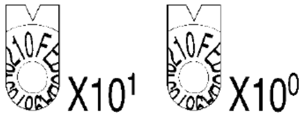
Name	Model
Special MC Unit Cable	XW2Z-100J-F1
Special MC Unit Terminal Block	XW2B-20J6-6

1) Mounting the MC Unit

Mount the MC Unit to the CPU Backplane.

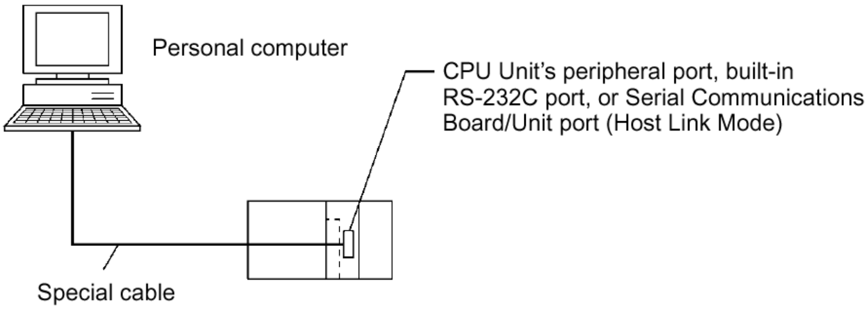
2) Setting the Unit Number

Set the unit number to 0 (MACHINE No. 00) with the rotary switch on the front panel of the MC Unit.



3) Connecting and Wiring Units and Devices

Connecting Programming Devices Connect the CPU Unit with connecting cable to a personal computer running the CX-Programmer and CX-Motion.


Personal computer
CPU Unit's peripheral port, built-in RS-232C port, or Serial Communications Board/Unit port (Host Link Mode)
Special cable

Unit	Port at CPU Unit	Personal Computer	Port at personal Computer	Network Type (serial comunications mode)	Cable model	Cable Length	Remarks
CPU unit	Internal peripheral port	IBM PC/AT or compatible	D-sub 9-pin male	Peripheral bus or host link (SYSMAC WAY)	CS1W-CN226	2 meters	
					CS1W-CN626	6 meters	
	Built-In RS-232C port D-sub 9 pin female				XW2Z-200S-V	2 meters	Use a connector with protection against static electric discharge
					XW2Z-500S-V	5 meters	
Serial Communications Board/Unit	RS-232C port D-sub 9 pin female			Host Link (SYSMAC WAY)	XW2Z-200S-CV	2 meters	
					XW2Z-500S-CV	5 meters	

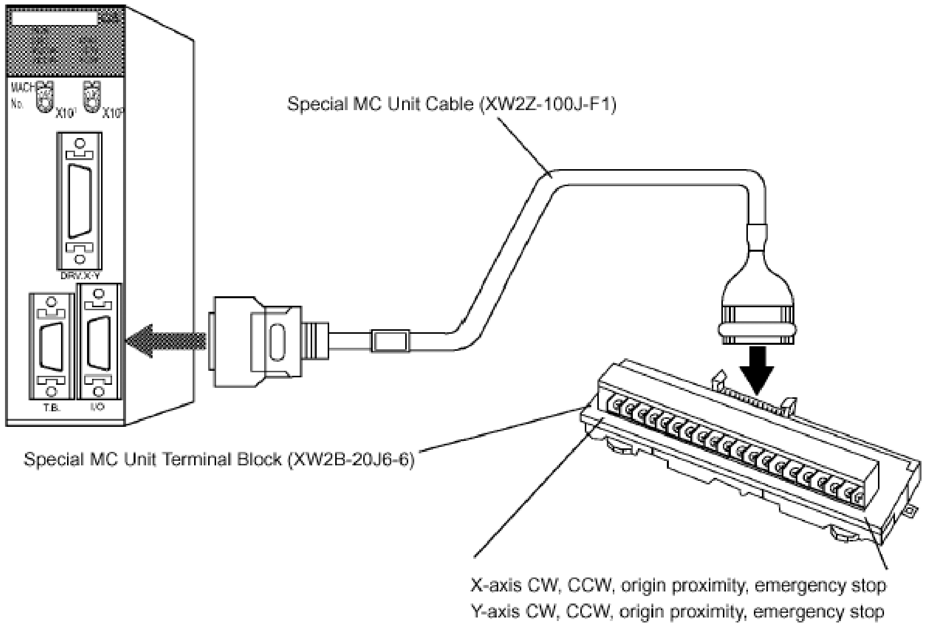
Connecting the Teaching Box

Using the following connecting cable to the teaching box to the MC unit.

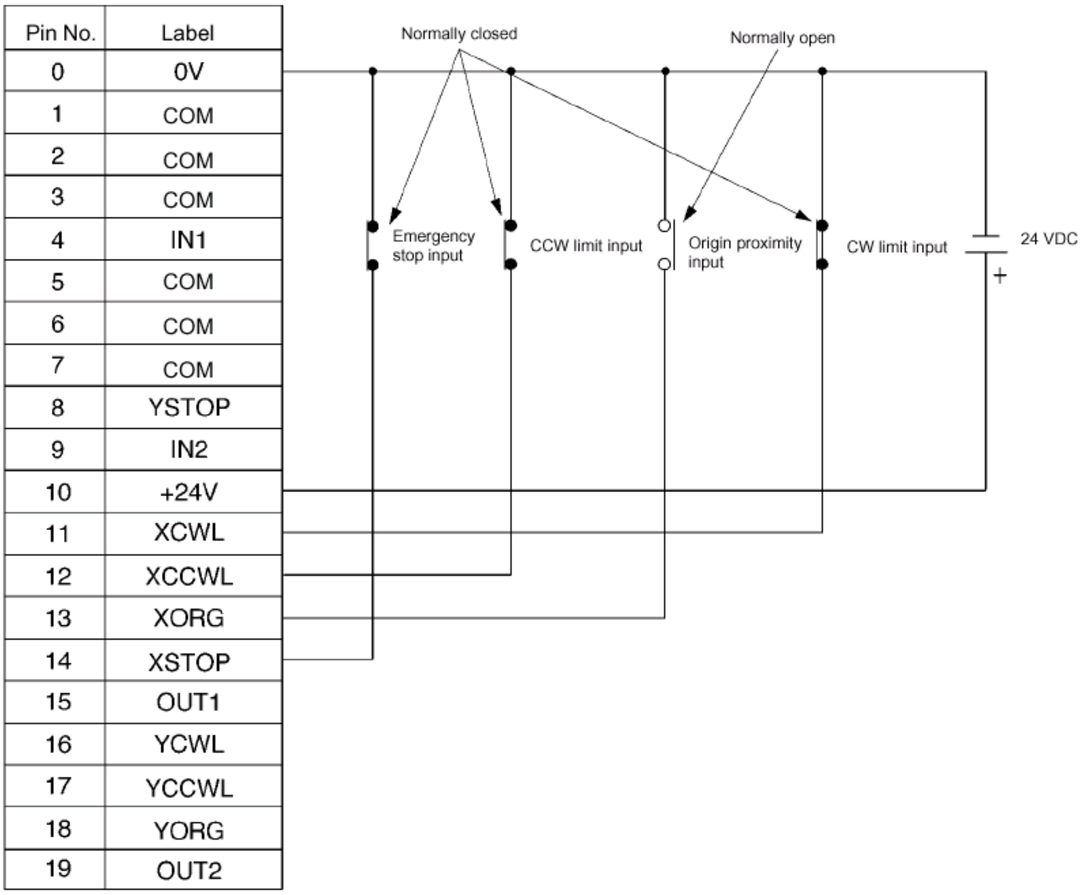
Model	Cable Length
CV500-CN224	2 meters
CV500-CN424	4 meters
CV500-CN634	6 meters

Connecting MC unit External Inputs

The cable and terminal block shown in this example are for the CS1W-MC221.


MACH No.
X10¹
X10⁰
DRV.X-Y
T.B.
I/O
Special MC Unit Cable (XW2Z-100J-F1)
Special MC Unit Terminal Block (XW2B-20J6-6)
X-axis CW, CCW, origin proximity, emergency stop
Y-axis CW, CCW, origin proximity, emergency stop

Connection Diagram

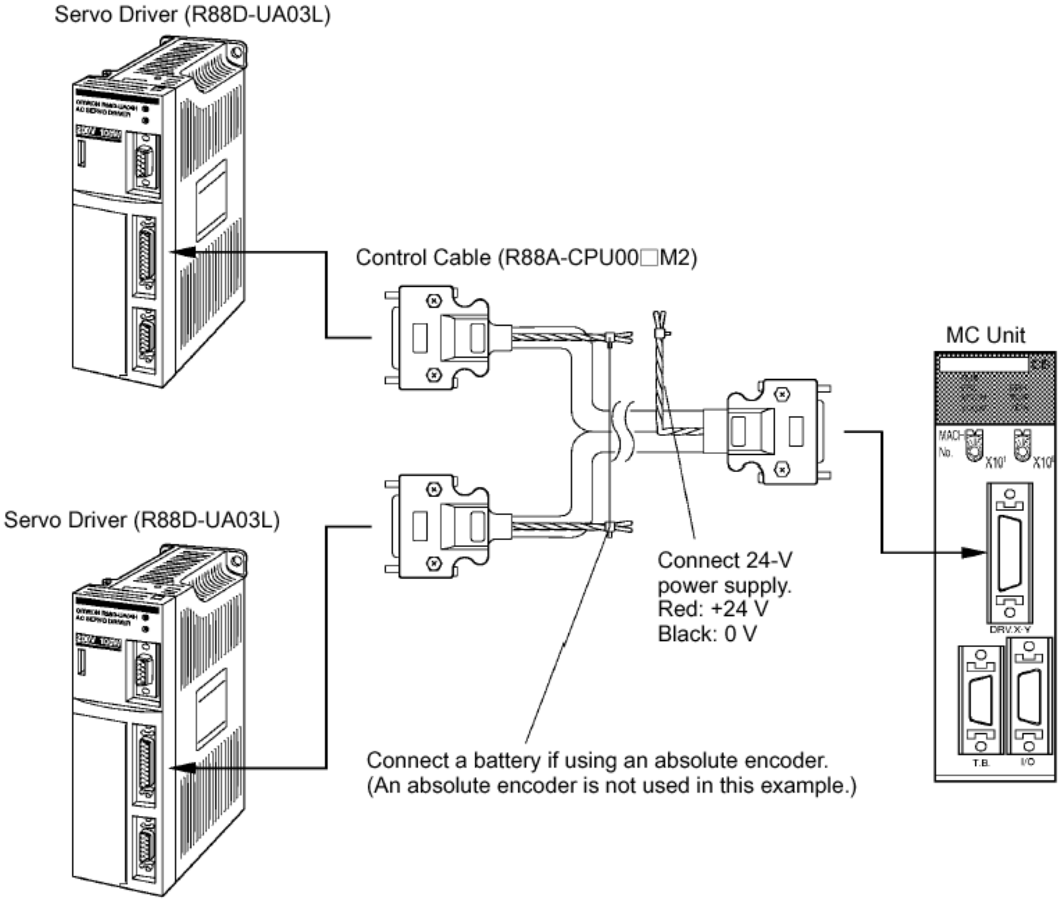


Pin No.	Label
0	0V
1	COM
2	COM
3	COM
4	IN1
5	COM
6	COM
7	COM
8	YSTOP
9	IN2
10	+24V
11	XCWL
12	XCCWL
13	XORG
14	XSTOP
15	OUT1
16	YCWL
17	YCCWL
18	YORG
19	OUT2

Connecting the Servo Drivers

Connect the MC Unit and Servo Drivers using special Control Cable, and connect a +24-V power supply. Then connect the Servo Drivers to the Servomotors using special Power Cable and Encoder Cable. (The Control Cable, Power Cable, and Encoder Cable are all purchased separately.)

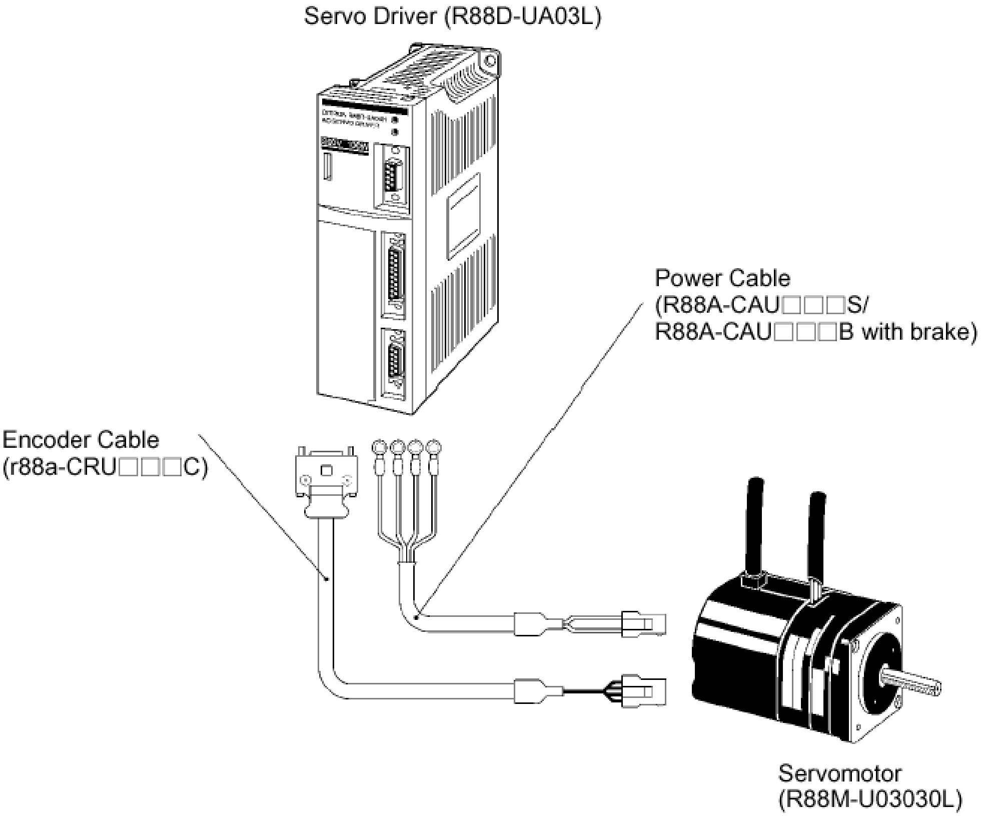
Connecting Control Cables


Servo Driver (R88D-UA03L)
Control Cable (R88A-CPU00□M2)
MC Unit
Servo Driver (R88D-UA03L)
Connect 24-V power supply.
Red: +24 V
Black: 0 V
Connect a battery if using an absolute encoder.
(An absolute encoder is not used in this example.)
MACH No.
X10¹
X10⁰
DRV.X-Y
T.B.
I/O

The following table shows that available Control Cables.

Series	Model Number	Length
R88D-U	R88A-CPU001M1 (for two axes) R88A-CPU002M2(for two axes)	1.0 meters 2.0 meters

Connecting Power Cables and Encoder Cables



The following table shows the available Power Cables and Encoder Cables.

Power Cables

Series	Specification	Model	Length
R88D-U	Without brake	R88A-CA003S R88A-CA005S R88A-CA010S R88A-CA015S R88A-CA020S	3 meters 5 meters 10 meters 15 meters 20 meters
	With brake	R88A-CAU003B R88A-CAU005B R88A-CAU010B R88A-CAU015B R88A-CAU020B	3 meters 5 meters 10 meters 15 meters 20 meters

Encoder Cables

Series	Model	Length
R88D-U	R88A-CRU003C R88A-CRU005C R88A-CRU010C R88A-CRU015C R88A-CRU020C	3 meters 5 meters 10 meters 15 meters 20 meters

Power Supply Wiring Wire the power supply according to the instructions in the applicable Servo Driver manual.

4) Creating I/O Tables

Turn ON the PC and create I/O tables according to the Units mounted to the Backplane. For details on creating I/O tables, refer to the CX-Programmer operation manual.

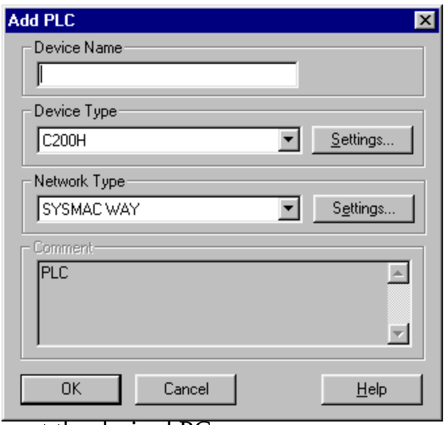
Note It is necessary to create an I/O table after mounting the MC Unit.

5) Creating Projects/Adding PCs (CPU Units)

Note A simple explanation of the operation procedure is given below. For details, refer to Section 1.

1, 2, 3...

1. Turn ON the power supply for the personal computer, start the CX-Motion and select **File/New/Projec**t.
2. The Project Properties Dialog Box will be displayed. The title can changed and explanation can be omitted.
3. Click **OK**. The Add PC Dialog Box will be displayed.



Add PLC

Device Name:

Device Type: C200H

Network Type: SYSMAC WAY

Comment: PLC

4. Input the desired PC name.
5. Click the Down Arrow at the right of the PC Model Field, and select the PC model from the menu. For example, if the CPU Unit is the CS1H-CPU67, select **CS1H**.
6. Click the **Set** button on the right of the PC Model Field. Click the Down Arrow at the right of the CPU type field, and select the type of CPU Unit from the menu. In this example, **CPU 67** is selected.
7. Click the **OK** button. The Add PC Dialog Box will return.
8. Set the Network type. In this example. SYSMAC WAY is selected.
9. Click the **Set** button on the right of the Network Type Field, and click the Driver Tab.
10. In Port Name under Connections, select the port of the personal computer to which the cable is connected (COM1/COM2).
11. Click **OK**. The Add PC Dialog Box will return.
12. Click **OK**. The PC will be added to the project.

6) Adding MC Units/Setting and Saving System Parameters

Note A simple explanation of the operation procedure is given below. For details, refer Section 1

1, 2, 3...

1. In the project workspace, click the icon of the PC that was added in 5) Creating Projects/Adding PCs (CPU Units), and then select **Edit/Add M**C. The MC Unit Dialog Box will be displayed.
2. Input the desired MC Unit name.
3. Click the Down Arrow at the right of the MC Type Field and select **CS1W-MC221** from the menu.
4. Set the unit number under Communications Environment Settings to 0 (unit number 0).
5. Click **OK**.
6. In this operation, new parameters are to be created, so click **No**. The Parameter File Save Type Dialog Box will be displayed.
7. Input the file name and click **Sav**e. The system parameters will be saved, and registered in the project.

System Parameter Settings

1, 2, 3... 1. Double-click the system parameters in the project workspace. The Unit and memory parameters will be displayed.

Set the system parameters that are required for controlling the MC Unit. Some of the defaults set by CX-Motion will be changed. Once the system parameters have been set, transfer them to the MC Unit and save them in the flash memory. For the task axis declaration, use the X and Y axes for task 1.

For detail, please refer to Section 1.

The following parameters require setting. Use default settings for other parameters.

Type of Parameter	Name	Set Value
Machine parameters	Minimum unit	0.01
	Display Unit	Mm
	Pulse Rate	1/100
	Positive/negative software limit	X-axis – direction: –50.00
		X-axis + direction: 600.00
		Y-axis – direction: –50.00
		Y-axis + direction: 600.00
Feed Rate parameters	Maximum feed rate	4000.00
	Maximum interpolation feed rate	4000.00
	High-speed origin search feed rate	400.00
	Low-speed origin search feed rate	40.00
	Maximum jog feed rate	400.00

When no axes are specified, set the same values for the X and Y axes.

7) Creating/Compiling/Saving MC Programs, Registering in Projects, and Saving Projects

Creating Program Using CX-Motion, create the following MC program. For details on creating programs, refer to Section 3..

MC Program

```
N000 P001 XY
*001 SAMPLE PROGRAM
N002 G04 5
N003 G26 XY
N004 G11
N005 G01 X400 Y300 F30
N006 G01 X100 Y200 F10
N007 G01 X200 Y400 F30
N008 G026 XY
N009 G79
```

Program Contents

N000 Declares program number P001 and use of the X and Y axes.
*001 This is a comment.
N002 Waits five seconds.
N003 Returns the X and Y axes to their reference origins.
N004 Selects the stop mode for positioning.
N005 Moves to X400 Y300 at speed 30.
N006 Moves to X100 Y200 at speed 10.
N007 Moves to X200 Y400 at speed 30.
N008 Returns the X and Y axes to their reference origins.
N009 Ends the program.

Saving and Compiling the MC program

Compiling

After creating the program and before transferring it to the MC unit, compile it and check that no errors are generated. For details, please refer to Section 3.

1, 2, 3...

1. Select **File/Propertie**s.
2. Select **CS1W-MC221** for the MC type, and click **O**K.
3. Select **Tool/Compil**e.

Saving

1, 2, 3...

1. Select **File/Save as....**
2. After inputting a file name, click **Sav**e.

Registering in the Project

1, 2, 3...

1. Click on the MC Unit that was added in 5) Creating Projects/Adding PCs (CPU Units, and select **Edit/Add File**).
2. Click on **Progra**m, and click on the file that was saved in Saving above. The MC program will be registered in the project.

Saving the Project Save the project before transferring system parameters and the MC program to the MC Unit.

1, 2, 3...

1. Click on the project in the project workspace, and then select **File/Save project as....**
2. Input a file name, and then select **Save**.

8)Transferring System Parameters and MC Programs to the MC Unit and Saving to Flash Memory

Note A simple explanation of the operation procedure is given below. For details, refer Section 3.

1, 2, 3...

1. Click on the MC Unit in the project workspace, select **Online/Transfer to MC**. The Download Dialog Box will be displayed.
2. Turn ON the **All MC files** and **Write to flash Memory** settings and click **Transfer**.

9) Creating Ladder Programs/Transferring and Saving to the CPU Unit

Creating Ladder Programs

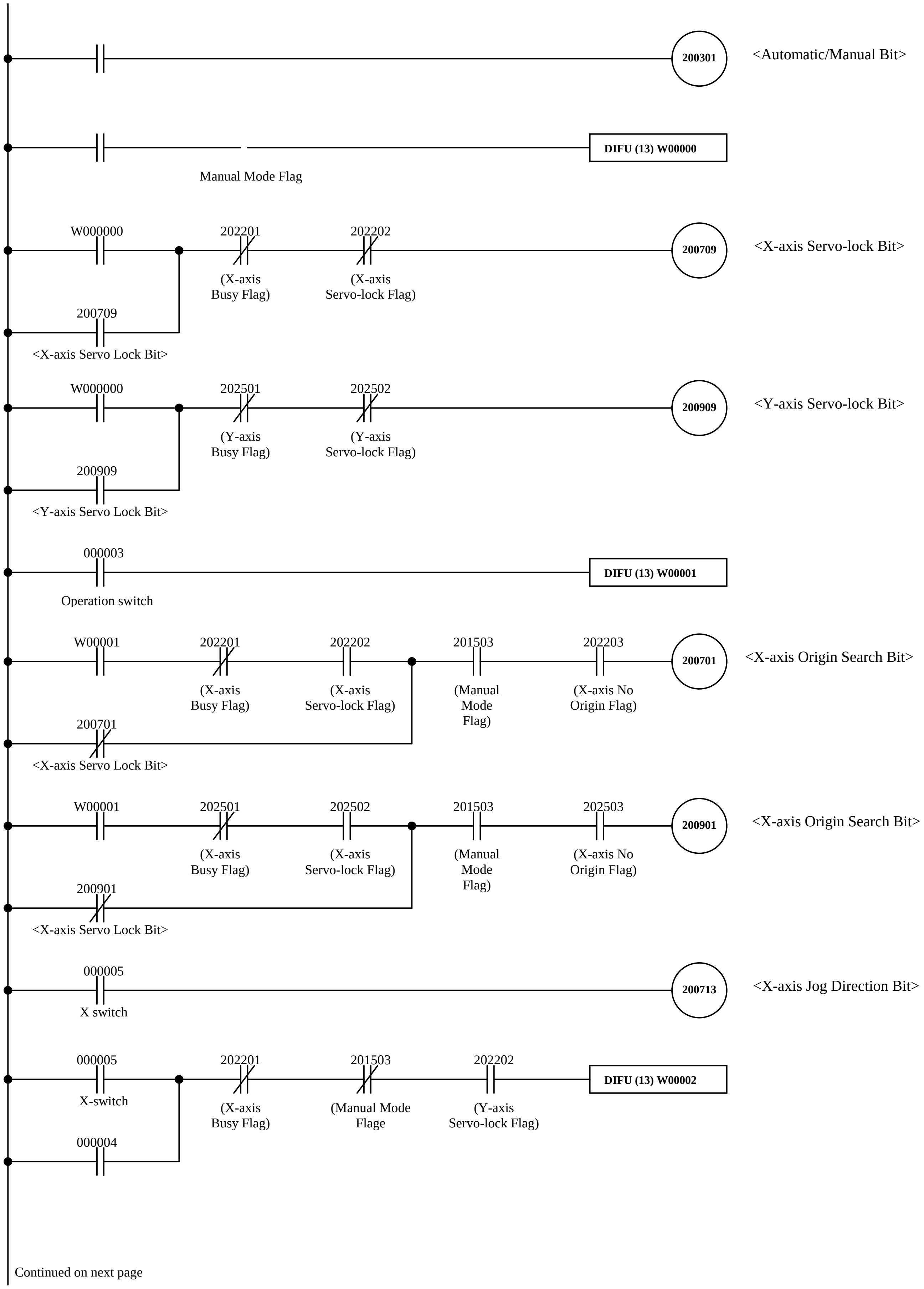
Create ladder programs using CX-Programmer. The programs required to execute servo-lock, origin search and jogging in Manual Mode and to execute the MC program in Automatic Mode are given below. For details about creating, ladder programs refer to the CX-Programmer Startup guide. After creating a ladder program, be sure to transfer it to the CPU Unit.

Notation Used in Ladder Program Diagrams

Bit/flag names in the < > symbols indicate outputs in allocated words. Bit/flag names in the ( ) symbols indicate inputs in allocated words.

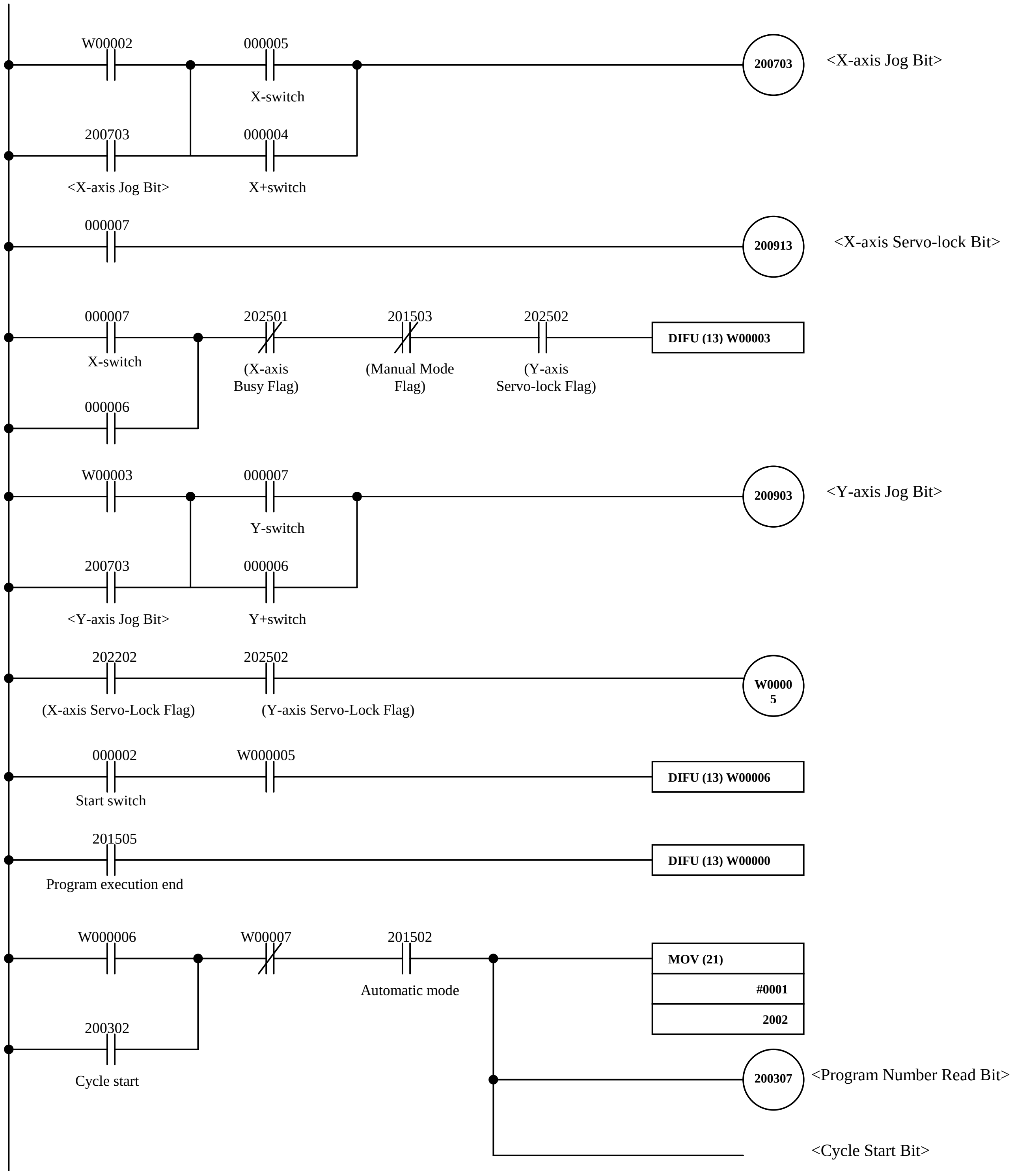
In the following examples, the MC Unit is the CS1W-MC221 and the unit number is 00. Therefore, 30 words will be allocated to the MC Unit starting from CIO 2000. Word Area words are from W00000 to W00008. The Input Unit corresponds to CIO 0000.

Ladder Program Input the ladder program shown below.

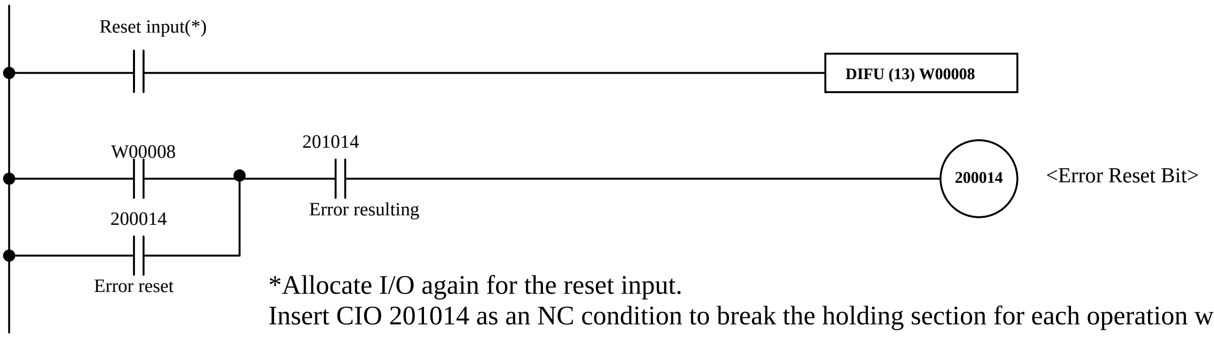


Continued on next page

Continued from previous page



Note A task error reset is required for when an error occurs during trial operation. Add an error reset like the on shown below to the program.



*Allocate I/O again for the reset input.
Insert CIO 201014 as an NC condition to break the holding section for each operation when an error occurs.

Transferring the Program

After the ladder program has been created, transfer it to the CPU Unit. (Perform the procedure using CX-Programmer. Refer to CX-Programmer Startup Guide.)

Saving the Program

Save the ladder program to a data disk. (Perform the procedure using CX-Programmer. Refer to CX-Programmer Startup Guide.)

10) Trial Operation (Operation Check)

Note A simple explanation of the operation procedure is given below. For details, refer to 9-2 Monitoring Operations in the CX-Motion Operation Manual.

Use the following procedures to confirm correct operation of the MC program in Automatic Mode and servo-lock, origin search and jogging operations in Manual Mode.

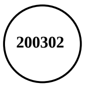
Manual Mode

Power ON
If power is supplied to the MC Unit when the Servo Driver alarm input is already ON, an error will not be generated. An error will occur, however, if the alarm turns OFF and then ON again, or if an instruction that includes axis operation is per-formed with the alarm still ON. This feature allows greater flexibility with the sequence in which power is supplied to the MC Unit and the Servo Drive.
When operating the MC Unit online form a personal computer, the power supply for the personal computer must be ON first. Otherwise an error will occur.

MC Coordinate Displays
Display the MC coordinates in order to check the present position of each axis.

1, 2, 3...
1. Select **Online/MC coordinate display**. (For more detail, please refer to Section 5.)
2. Click **Star**t.

Servo-lock



Set the Automatic/Manual switch to Manual. Servo-lock the X and Y axes by pressing the Operation Buttons. By performing servo-lock, power will be supplied to each axis. While the servo is locked, the CW and CCW indicators will flash. On the MC coordinates display screen, the present position values and the error counter pulse numbers will constantly change in a neighborhood of 0.

Origin Search
Set the Automatic/Manual switch to Manual. Execute an origin search for the X and Y axes by pressing the Origin Buttons. The axes will move to the origin. When origin search has been completed, the reference coordinate system present position values in the MC coordinates display screen will change to values near 0.

Jogging
Check that jogging is performed correctly for each axis. Set the Automatic/Manual Switch to Manual. Press and hold down the X+ button to move the X axis in the + direction. The present position value for the X axis on the MC monitoring screen will increase. Press and hold down the X– button to move the X axis in the – direction. The present position value for the X axis on the MC monitoring screen will decrease. Use the same procedure to confirm correct operation for the Y axis. Be sure that the present position value does not exceed the set software limit. A CW software limit or CCW software limit error will occur.

MC Program

Set the Automatic/Manual switch to Automatic. Press the Start Button to execute the MC program. From the MC task display screen, use the following procedure to display and check the MC program.

1, 2, 3...
1. Set the Automatic/Manual switch to Automatic.
2. Press the Start Button.
3. Select **Online/MC task display**. (For more detail, please refer to Section 5.)
4. Click **Star**t. The MC program will be displayed on the screen. The part of the program currently being executed will be displayed in reverse video.
5. Check that the axes have moved to the coordinates indicated in the lower right-hand area of the screen under **Axis monitoring**.

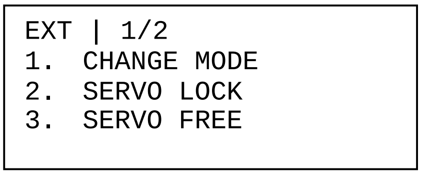
6-2-2 Checking Operation from the Teaching Box

This section explains how to use the Teaching Box to execute the MC program and the jog operation. It explains cyclic execution, in which the specified program is executed from the beginning, and single-block execution, in which the specified program is executed one block at a time.

If servo-lock and origin search have not been executed using the switch box, execute the origin search from the Teaching Box after setting locking the servos before executing the MC program and jog operation.

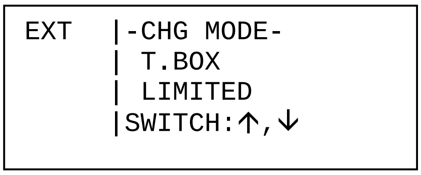
Changing the Mode to T. BOX RESERVED

1, 2, 3...
1. Press the EXT Key.



```
EXT | 1/2
1.  CHANGE MODE
2.  SERVO LOCK
3.  SERVO FREE
```

2. Press 1.

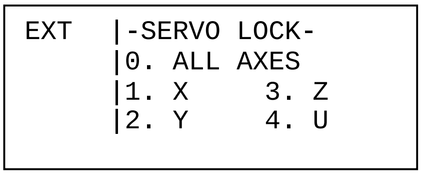


```
EXT  |-CHG MODE-
     | T.BOX
     | LIMITED
     |SWITCH:↑,↓
```

3. Press the Down Key. (Pressing the UP and Down Keys switches among the three modes.)

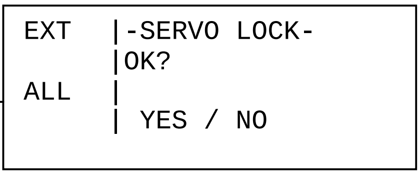
Executing Servo-Lock

1,2,3… 1. Select 2. SERVO LOCK from the Extension Mode Menu.



```
EXT   |-SERVO LOCK-
      |0. ALL AXES
      |1. X     3. Z
      |2. Y     4. U
```

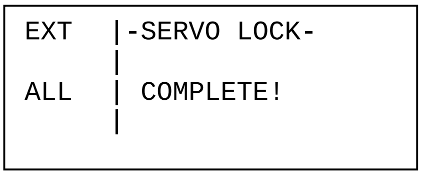
2. Select all the axes.



```
EXT   |-SERVO LOCK-
      |OK?
ALL   |
      | YES / NO
```

The specified axis/axes are shown here.

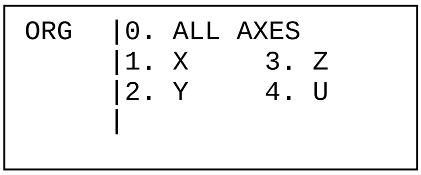
3. Press the YES Key. The Servos will be locked, and the following screen will be displayed.



```
EXT   |-SERVO LOCK-
      |
ALL   | COMPLETE!
      |
```

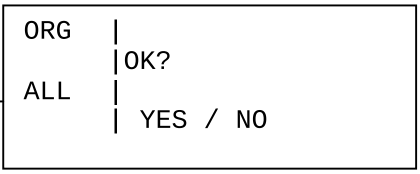
Executing Origin Search for All Axes

1, 2, 3... 1. Press the ORIG SRCH Key.



```
ORG   |0. ALL AXES
      |1. X     3. Z
      |2. Y     4. U
      |
```

2. Select 0. ALL AXES. Press 1 or 2 when selecting individual axis.



```
ORG   |
      |OK?
ALL   |
      | YES / NO
```

The specified axis/axes are shown here.

3. Press the YES Key. An origin search will be started and the present position monitoring screen will be displayed.

```
ORG   |X*         12.00
      |Y*         12.32
ALL   |Z*          0.00
      |U*          0.00
```

The present position will be refreshed for moving axes.
The origin search operations will be performed and the axes will stop at the reference coordinate system origins. The asterisks will disappear at that point, and the present position is shown as 0.

When the origin search operation has been completed for the specified axes, the following screen will be displayed.

```
EXT   |
      |
ALL   | COMPLETE!
      |
```

4. Press the CLR Key. The axis selection screen in step one of this procedure will return.

When an error occurs in the MC Unit or the Servo Driver, clear it by means of the following procedures.

Resetting MC Unit Errors

1, 2, 3... 1. Press the ERR CLR Key.

```
[RT]

     1.MC UNIT
     2.DRIVER
```

2. Press 1.

```
[RT]          MC UNIT

    RESET ERROR ?
     YES / NO
```

3. Press the YES Key. The error will be reset for the task in which the error occurred.

```
[RT]          MC UNIT

    ERROR RESET !
```

Resetting Servo Driver Errors

1, 2, 3... 1. Select 2.DRIVER from the MC Unit’s Error Reset Procedure 1 Menu.

```
[RT]            DRIVER
         0. ALL AXES
         1. X  3. Z
         2. Y  4. U
```

2. Press a number from 0 to 2 to select the number of axes to be reset.

```
[RT]          DRIVER
   <ALL AXES>
 RESET ERROR ?
     YES / NO
```

The specified axis/axes are displayed here.
0: All axes 1: X axis 2: Y axis

3. Press the YES Key. The error will be reset for the selected axes.

```
[RT]          DRIVER

    ERROR RESET !
```

Jogging

1, 2, 3... 1. Press the JOG Key. The Unit will go into Jog Mode and the following screen will be displayed.

```
JOG  |X*        0.00
     |Y*        0.00
     |Z*        0.00
050  |U*        0.00
```

The override values are displayed here. The default value is
50 %. Press the Up and Down Keys to set the values from
10% to 100% in increments of 10%.

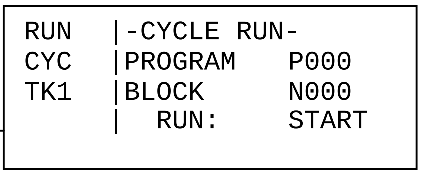
2. To execute jogging along the X or Y axes, press any of the following four jog keys: +X, +Y, –X, –Y. The specified axis will move in the specified direction, and the present position will be refreshed on the screen.

Executing Cyclic Execution

1, 2, 3... 1. Press the PROG EXEC Key.

```
RUN  |1.CYCLE RUN
     |2. SINGLE RUN
     |
     |
```

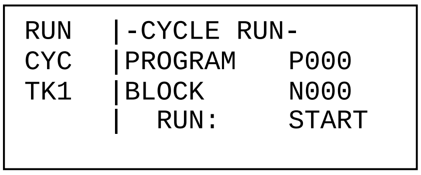
2. Select 1. CYCLE RUN from the menu.



```
RUN  |-CYCLE RUN-
CYC  |PROGRAM   P000
TK1  |BLOCK     N000
     |  RUN:    START
```

Task 1 is displayed.

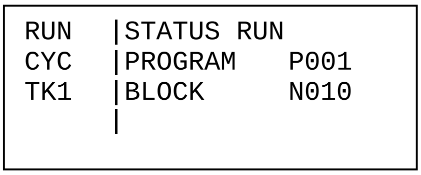
3. Specify the program number. In this case, the program number is 001, so just input 1.



```
RUN  |-CYCLE RUN-
CYC  |PROGRAM   P000
TK1  |BLOCK     N000
     |  RUN:    START
```

If the program is changed from CX-Motion, the program number must be entered again even if it is the same. If it is not entered again, an error will be generated. In this case program execution will begin from the first block.

4. Press the START Key.



```
RUN  |STATUS RUN
CYC  |PROGRAM   P001
TK1  |BLOCK     N010
     |
```

Each status has the following meaning .
RUN: Under memory operation
END: Memory operation ended.
ERR: Error occurring
PAUSE: Temporary suspension

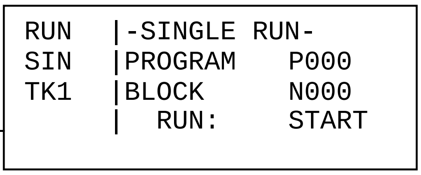
When the PAUSE Key is pressed, program execution will be paused and the previous screen will be restored. Program execution is resumed by pressing the START Key. The program cannot be executed while an error is in effect. If the program number is entered again, program execution will begin from the first block.

5. Press the CLR Key after the program has been completed. The menu for selecting the operation mode will be restored. (The CLR Key is not effective during program execution.)

If the status is either END or ERR, pressing the CLR Key will return to the restart screen.

Executing Single-block Execution

1, 2, 3... 1. Select 2.SINGLE RUN from the menu.



```
RUN  |-SINGLE RUN-
SIN  |PROGRAM   P000
TK1  |BLOCK     N000
     |  RUN:    START
```

Task 1 is displayed.

2. Specify the program number. In this case, the program number is 001, so just input 1. If the program number is already set to 001 after cyclic execution, there is no need to enter anything here.
3. Press the START Key. The first block of program number 001 will be executed. In this case, block N002 will be executed because the first block is a comment.

```
RUN  |STATUS RUN
SIN  |PROGRAM   P001
TK1  |BLOCK     N002
     |
```

To halt the operation in progress, press the PAUSE Key.

3. When the execution of the first block has been completed , PAUSE status will go into effect and program execution will be stopped. The number of the next block to be executed will be displayed.

```
RUN  |STATUS PAUSE
SIN  |PROGRAM   P050
TK1  |BLOCK     N002
     |    RUN:  START
```

Press the PROG EXEC Key to execute the next block. Press the CLR Key to return to the program number input screen. Then press to CLR Key again to return to the operating mode menu.

6-3 Considerations When Starting Up the MC Unit

6-3-1 Setting the System Parameters

The main considerations when setting the machine parameters are explained below, taking U-series Servo Drivers as an example.

An OMRON Servomotor connected to the MC Unit will operate if the default values (forward for positive voltage, and forward for encoder increase) are set.

If the system is set for a right-hand ball screw connection, the default settings will produce a decrease in the present value as the table (workpiece) moves away from the Servomotor. To have the present value increase as the table (work-piece) moves away from the Servomotor with a right-hand ball screw connection, use either of the following two methods.

- Set the MC Unit parameter (rotation direction) so that the motor will reverse for a positive voltage and reverse for an encoder increase.
- Set the Servomotor’s mode to Reverse Mode. For U-series Servomotors, turn ON bit 0 of Cn-02.

Encoder Resolution The encoder resolution parameter sets the number of pulses (encoder frequency dividing ratio) that can be output from the Servo Driver per Servomotor revolution. For U-series Servomotors, set the encoder resolution according to the encoder frequency dividing ratio (Cn-0A), and not the number of encoder pulses parameter.

Note When a U-series Servo Driver with a capacity of 1 kW or more is used, a speed of up to 614.4 kpps is possible (when the encoder resolution is 8,192 and the maximum speed is 4,500 r/m). The maximum control speed of the MC Unit, however, is 500 kpps, so adjust the encoder resolution and the encoder frequency dividing ratio so that the speed will stay within this range.

Pulse Rate

With a ball screw connection, set the ball screw pitch (the distance advanced for one turn of the ball screw) for the numerator. Set the value derived by multiplying the encoder resolution by the ratio for the denominator. This number is a ratio of the number of millimeters advanced per pulse.

Maximum Number of Motor Revolutions

Set the rated number of revolutions for the motor (generally the catalog value).

Note

1. There is a difference between the acceleration/deceleration times set in the MC Unit and the Servo Driver. The MC Unit acceleration/deceleration time defines the time for accelerating and decelerating until the MC Unit’s maximum feed rate or interpolation speed reference. The Servo Driver acceleration/deceleration time, in contrast, sets the delay time for accelerating to the speed reference that is given. When the Servo Driver acceleration/deceleration time is set, it lowers the gain for the system, so do not make this setting when using a Servo Driver in combination with an MC Unit.

2. When an S-curve is set in the MC Unit and a software start is set in the Servo Driver, it puts the MC outputs through a filter and the gain is extremely decreased. Set both the S-curve and software start may produce faulty operation.

Transferring and Saving Data

When using CX-Motion to transfer data to the MC Unit, select Save/Transfer (Computer " MC). Parameters, position data, programs, and so on, are transferred between a file (on a floppy or hard disk) and the MC Unit, so they must be saved or the MC Unit will operate using the old data. After transferring the data, be sure to save it to flash memory before turning OFF the power.

If the machine parameters or Unit parameters are changed and transferred, a system setup error will occur. It can be cleared by turning OFF the power and restarting the Unit. The error will appear to be cleared by just executing an error reset, but internally the old system parameters will remain in effect. The program cannot be correctly transferred while this condition persists, so whenever a system setup error occurs, be sure to turn OFF the power and use the Restart Bit in the CPU Unit to restart the Unit. Then transfer the program after the restart.

Backing Up Data to Flash Memory

With the CS1W-MC421 and CS1W-MC221, memory is backed up to flash memory following the transfer of system parameters and other data. The data in flash memory is always reloaded when the MC Unit is started. If data is not written to flash memory, it will be valid only until the power is turned OFF, so be sure to save it to flash memory after transferring it if it is required.

6-3-2 Considerations When Starting Up or Making Adjustments

An error may occur if the wiring check parameters are left at their default settings. These settings are made assuming a suitable load, but the load in the actual system may be different and a wiring error may be incorrectly detected. If a wiring error does occur, refer to CS1W-MC421/221 Motion Control Units Operation Manual Section 2-5 Wiring Check Troubleshooting and adjust the wiring check parameters.

6-3-3 Changing the Teaching Box Mode for Trial Operation

When the Teaching Box is connected to the MC Unit, it first goes into T. BOX LIMITED mode, in which only monitoring is possible. To execute the program or jog, the mode must be switched to either T. BOX RESERVED or T. BOX ENABLED. The operating mode can be changed only from the Teaching Box itself.

For the procedure for changing the Teaching Box mode to either T. BOX RESERVED or T. BOX ENABLED, refer to Changing the Mode to T. BOX RESERVED in 6-2-2 Checking Operation From the Teaching Box.

To change position data from the Teaching Box, press the EXT Key, select 4. Memory Protect, and use the Up Key or Down Key to clear the memory protection. Operations from the CPU Unit are not possible when the Teaching Box is in any mode other than T. BOX LIMITED. While operations are being executed from the Teaching Box in T. BOX RE-SERVED or T. BOX ENABLED Mode, other operations will not be carried out even if commands are given from the CPU Unit.

For details on Teaching Box operations, refer to the Teaching Box Operation Manual.

Appendix A
G-language Programming

This section describes using the G-language to program motion control in the MC Unit.

Programs and Tasks

1 Overview

The CS1W-MC421 MC Unit can execute up to four tasks (tasks 1 to 4), and the CS1W-MC221 can execute one or two (tasks 1 to 2). (A task is a unit of execution for a program.)

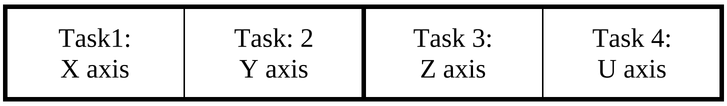
By executing two or four tasks at the same time, the MC Unit can perform the same functions as two or four controllers.

The number of tasks and axes to be used are set in advance using CX-Motion for the unit parameters.

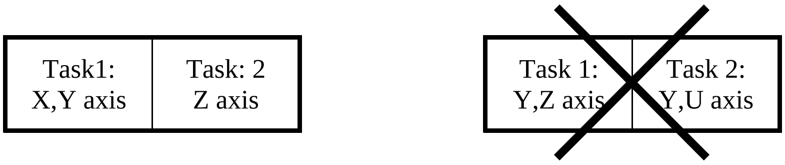
Number of Tasks and Axes

The X and Y axes can be used with the CS1W-MC221, and the X, Y, Z, and U axes can be used with the CS1W-MC421. Each axis can be used for only one task, i.e., any axis assigned to one task cannot be used in another task.

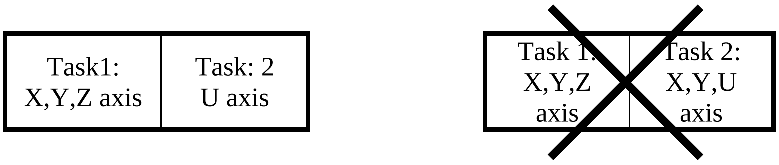
Example: 4 Tasks and 4 Axes



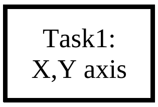
Example 2: 2 Tasks and 3 Axes



Example 3: 2 Tasks and 4 Axes



Example 4: 1 Task and 2 Axes



Tasks and Blocks

The MC Unit is capable of storing a total of 2,000 blocks of program. The maximum number of blocks that can be executed in each task depends on the number of tasks as shown in the following table. These figures include subroutines.

Number of tasks	Maximum number of blocks (CS1W-MC421)	Maximum number of blocks (CS1W-MC221)
1	2,000 blocks	2,000 blocks
2	1,000 blocks/task	1,000 blocks/task
3	666 blocks/task	---
4	500 blocks/task	---

Note A maximum of 800 blocks can be used in one program, including subroutines.

Tasks and Programs

A maximum of 100 programs can be managed by the MC Unit. The number of programs that can be managed per task depends on the number of tasks as shown in the following table. These figures include subroutines.

Number of tasks	Maximum number of blocks (CS1W-MC421)	Maximum number of blocks (CS1W-MC221)
1	100 programs	100 programs
2	50 programs/task	50 programs/tasks
3	33 programs/task	---
4	25 programs/task	---

Note The same program number cannot be used for different tasks.

G Language Overview

1 List of G-language Commands

Code	Name	Function
G00	POSITIONING	Positions up to two or four axes simultaneously with PTP control at the maximum feed rate.
G01	LINEAR INTERPOLATION	Performs linear interpolation on up to two or four axes simultaneously at the specified interpolation feed rate.
G02	CIRCULAR INTERPOLATION (CLOCKWISE)	Performs 2-axis circular interpolation in the clockwise direction at the specified interpolation feed rate.
G03	CIRCULAR INTERPOLATION (COUNTERCLOCKWISE)	Performs 2-axis circular interpolation in the counterclockwise direction at the specified interpolation feed rate.
G04	DWELL TIMER	Waits for the specified period of time.
G10	PASS MODE	Performs operations one-by-one in sequence without deceleration to stop.
G11	STOP MODE	Performs the next operation after completing positioning.
G13	IN-POSITION CHECK OFF MODE	Starts the next operation without waiting for positioning to be completed.
G17	CIRCULAR PLANE SPECIFICATION (X-Y)	Sets the X-Y plane as the plane for circular interpolation.
G18 (See note)	CIRCULAR PLANE SPECIFICATION (X-Z)	Sets the X-Z plane as the plane for circular interpolation.
G19 (See note)	CIRCULAR PLANE SPECIFICATION (Y-Z)	Sets the Y-Z plane as the plane for circular interpolation.
G20 (See note)	CIRCULAR PLANE SPECIFICATION (X-U)	Sets the X-U plane as the plane for circular interpolation.
G21 (See note)	CIRCULAR PLANE SPECIFICATION (Y-U)	Sets the Y-U plane as the plane for circular interpolation.
G22 (See note)	CIRCULAR PLANE SPECIFICATION (Z-U)	Sets the Z-U plane as the plane for circular interpolation.
G26	REFERENCE ORIGIN RETURN	Moves to the reference origin.
G27	WORKPIECE ORIGIN RETURN	Moves to the workpiece origin.
G28	ORIGIN SEARCH	Performs an origin search in the specified axis.
G29	ORIGIN UNDEFINED	Makes the origin undefined.

Code	Name	Function
G30	SPEED CONTROL	Feeds up to two or four axes simultaneously at a controlled speed.
G31	INTERRUPT FEEDING	Performs interrupt feeding operations.
G32	TRAVERSE	Executes traverse operations.
G50	SELECT REFERENCE COORDINATE SYSTEM	Specifies the reference coordinate system.
G51	SELECT WORKPIECE COORDINATE SYSTEM	Specifies the workpiece coordinate system.
G53	CHANGE WORKPIECE ORIGIN OFFSET	Changes the origin of the workpiece coordinate system.
G54	CHANGE REFERENCE COORDINATE SYSTEM PV	Changes the present value in the reference coordinate system.
G60	ARITHMETIC OPERATIONS	Performs arithmetic operations on numerical values, position data, and registers.
G63	SUBSTITUTION	Substitutes numerical values, position data, or registers into other position data or registers.
G69	CHANGE PARAMETER	Changes the specified parameter.
G70	UNCONDITIONAL JUMP	Unconditionally jumps to the specified block.
G71	CONDITIONAL JUMP	Jumps to the specified block when the condition is met.
G72	SUBROUTINE JUMP	Calls the specified subroutine.
G73	SUBROUTINE END	Ends the subroutine.
G74	OPTIONAL END	Ends the block currently being executed when the specified optional input is ON.
G75	OPTIONAL SKIP	Skips the block after this command when the specified optional input is ON.
G76	OPTIONAL PROGRAM PAUSE	Pauses the program when the specified optional input is ON.
G79	PROGRAM END	Ends the main program.
G90	ABSOLUTE SPECIFICATION	Specifies the use of absolute coordinates in axis operations.
G91	INCREMENTAL SPECIFICATION	Specifies the use of relative coordinates in axis operations.

Note Not supported by the CS1W-MC221.

2 G-language Formats

The following notion is used for the operands.

< >	Name of an operand to be specified
[ ]	Optional operand
...	An operand that can be specified more than once
•	A required space
_	Optional space

Name	Code	Operands	
POSITIONING	G00	<Axis movement command ...>	[_M<M code>[/Stopover (Note 2.)]] [_D<D code>[/Stopover (Note 2.)]] [_#<optional number>]
LINEAR INTERPOLATION	G01	<Axis movement command ... >	[_F<speed reference>[/Stopover (Note 2.)]] [_M<M code>[/Stopover (Note 2.)]] [_D<D code>[/Stopover (Note 2.)]] [_#<optional number>]

Name	Code	Operands
CIRCULAR INTERPOLATION (CLOCKWISE)T	G02	[<Axis movement command ... >]_<I to H center coordinate ...> [_F<speed reference>] [_M<M code>[/Stopover (Note 2.)]] [_D<D code>[/Stopover (Note 2.)]] [_L<number of turns>] [_#<optional number>] [<Axis movement command ... >]_<R radius> [_F<speed reference>] [_M<M code>[/Stopover (Note 2.)]] [_D<D code>[/Stopover (Note 2.)]] [_L<number of turns>] [_#<optional number>]
CIRCULAR INTERPOLATION (COUNTERCLOCKWISE)	G03	[<Axis movement command ... >]_<I to H center coordinate ...> [_F<speed reference>] [_M<M code>[/Stopover (Note 2.)]] [_D<D code>[/Stopover (Note 2.)]] [_L<number of turns>] [_#<optional number>] [<Axis movement command ... >]_<R radius> [_F<speed reference>] [_M<M code>[/Stopover (Note 2.)]] [_D<D code>[/Stopover (Note 2.)]] [_L<number of turns>] [_#<optional number>]
DWELL TIMER	G04	• <wait time>
PASS MODE	G10	---
STOP MODE	G11	---
IN-POSITION CHECK OFF MODE	G13	---
CIRCULAR PLANE SPECIFICATION (X-Y)	G17	---
CIRCULAR PLANE SPECIFICATION (X-Z) (See note.)	G18	---
CIRCULAR PLANE SPECIFICATION (Y-Z) (See note.)	G19	---
CIRCULAR PLANE SPECIFICATION (X-U) (See note.)	G20	---
CIRCULAR PLANE SPECIFICATION (Y-U) (See note.)	G21	---
CIRCULAR PLANE SPECIFICATION (Z-U) (See note.)	G22	---
REFERENCE ORIGIN RETURN	G26	<Axis name ... > [_M<M code>[/Stopover (Note 2.)]] [_D<D code>[/Stopover (Note 2.)]]
WORKPIECE ORIGIN RETURN	G27	<Axis name ... > [_M<M code>[/Stopover (Note 2.)]] [_D<D code>[/Stopover (Note 2.)]]
ORIGIN SEARCH	G28	<Axis name ... > [_M<M code>[/Stopover (Note 2.)]] [_D<D code>[/Stopover (Note 2.)]]
ORIGIN UNDEFINED	G29	<Axis name ...>
SPEED CONTROL	G30	<Axis movement command>

Name	Code	Operands
INTERRUPT FEEDING	G31	<Axis movement command> _F <Speed Reference 1> <Axis name> <Coordinate 1>[/<Coordinate 2] [_F <Speed Reference 2>] [_M<M code>[/Stopover (Note 2.)]] [_D<D code>[/Stopover (Note 2.)]] [_S]
SELECT REFERENCE COORDINATE SYSTEM	G50	---
SELECT WORKPIECE COORDINATE SYSTEM	G51	---
CHANGE WORKPIECE ORIGIN OFFSET	G53	<Offset value ... >
CHANGE REFERENCE COORDINATE SYSTEM PV	G54	<Present value ... >
ARITHMETIC OPERATIONS	G60	<First term = Second term Operator Third term>
SUBSTITUTION	G63	<First term = Second term>
CHANGE PARAMETER	G69	<#System parameter type>/<New setting ... >
UNCONDITIONAL JUMP	G70	<Jump destination block number> [/L<Number of loops>]
CONDITIONAL JUMP	G71	<Jump destination block number>/<condition equation>
SUBROUTINE JUMP	G72	<subroutine number>
SUBROUTINE END	G73	---
OPTIONAL END	G74	● <Optional number>
OPTIONAL SKIP	G75	● <Optional number>
OPTIONAL PROGRAM STOP	G76	● <Optional number>
PROGRAM END	G79	---
ABSOLUTE SPECIFICATION	G90	---
INCREMENTAL SPECIFICATION	G91	---

Note 1. Not supported by the CS1W-MC221.
2. The stopover function can be used with either an M code or a D code, but not with both.

3 List of G Symbols

The following table lists the symbols used in G-language programming.

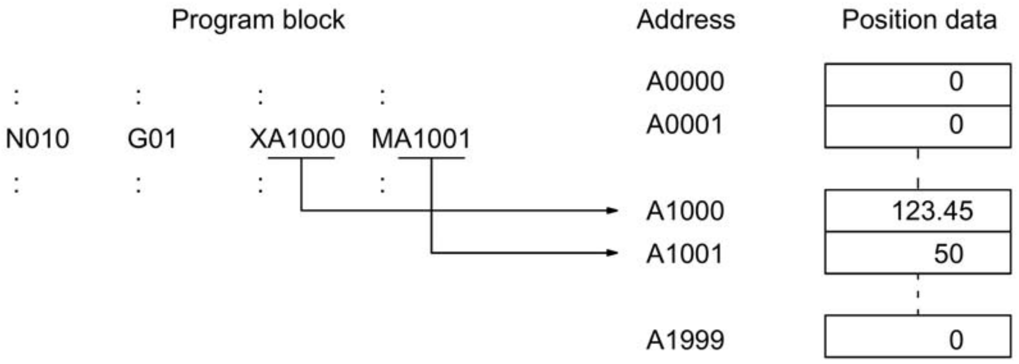
Symbol	Meaning
A	Indicates a position data address. There are 2000 addresses ranging from A0000 to A1999.
D	D code
E	Indicates a register. There are 32 registers ranging from E00 to E31.
F	Indicates the speed when performing interpolation operations.
G	G command
H (See note.)	Arc center coordinate for the U-axis.
I	Arc center coordinate for the X-axis.
J	Arc center coordinate for the Y-axis.
K (See note.)	Arc center coordinate for the Z-axis.
L	Number of loops
M	M Code
N	Block number
O	Specifies the end processing method for G32 winding.
P	Program number
R	Circle radius

Symbol	Meaning
S	Specifies no error when there is no G31 interrupt input.
U (See note.)	U-axis
X	X-axis
Y	Y-axis
Z (See note.)	Z-axis
/	Separator
( )	Indirect specification
#	Optional number, system parameter type
*	Comment

Note Not supported by the CS1W-MC221.

Specifying Position Data Addresses (A0000 to A1999)

It is possible to use the contents of a position data address for position data or an M code by specifying that address in an operand in an axis movement command or M code. For example, when the following program is executed, the contents of A1000 (123.45) will be used for the X-axis data and the contents of A1001 (50) will be used for the M code.



Specifying Registers (E00) to (E31)

Position data can be specified indirectly by specifying a register (E00) to (E31) in an axis movement command or M code. The relationship between the register and position data contents is shown below.

Register		Position data	
E00	1000	A1000:	123.35
E01	1001	A1001:	50

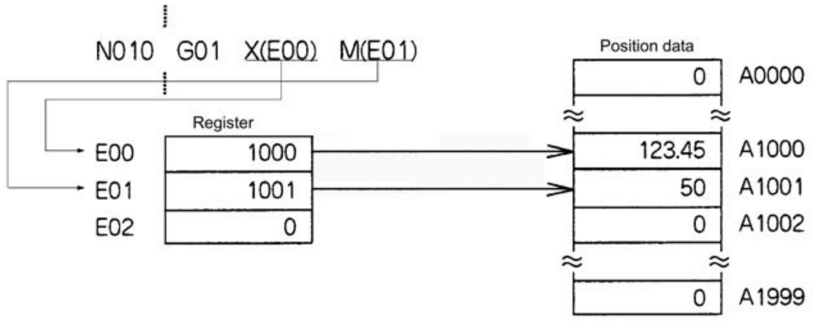
Registers and position data are not assigned together to each task; they can be used independently for different tasks.

The contents of registers are all cleared to zero only when power is turned ON or when a restart is executed. Initialize register contents at the beginning of the pro-gram by executing an command such as the SUBSTITUTION command (G63). Registers can contain values from 0 through 1999. An error will occur if a value greater than 1999 is input in a register. The possible range of values for position data is –39,999,999 through 39,999,999.

Indirect Addressing of Position Data

If the register name is in parentheses, i.e. (E00), the content of register will be treated as a position data address.

For example, when the following program is executed, the contents of A1000 (123.45) will be used for the X-axis data and the contents of A1001 (50) will be used for the M code.

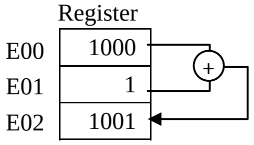


Direct Addressing of Registers

If the register name is not in parentheses, i.e. E00, the command will operate directly on the content that register. In the following example, the content of E00 and E01 are added and the result is placed in E02.

Program block

```
N010     G60     E02=E00+E01
```



4 Declaring Program Numbers and Axes

The program number and axes being used must be declared at the beginning of the program.

Format <program number_axes…>

Operands The following table shows the program numbers and axes that can be declared for main programs and subroutines.

Programs	Program number	Axes
Main programs	P000 through P499	X,Y,Z,U
Subroutines	P500 through P999	

Usage There are two kinds of MC Unit programs, main programs and subroutines. Main programs are assigned program numbers P000 through P499 and subroutines are assigned program numbers P500 through P999.

Every program must begin with a declaration of the program number and the axes being used. Main programs end with a PROGRAM END command (G79) and subroutines end with a SUBROUTINE END command (G73). This declaration is made at the beginning of the program (block number N000); an error (determined by CX-Motion) will occur if the declaration is made in a later block.

The MC Unit can contain up to 100 programs.

Programming Examples

Use the following format for main programs.

```
N000   P100   XY
       :      :
       :      :
N100   G79                    PROGRAM END command
```

Use the following format for subroutines.

```
N000    P510    XY
        :       :
        :       :
N100    G73                 SUBROUTINE END command
```

When executing the main program from the first block, make initial settings such as the following.

1,2,3... 1. Restore the acceleration/deceleration time, interpolation acceleration/deceleration time, and workpiece origin offset value to the system parameter values.

2. Set the operating mode to Pass Mode.

3. Set the coordinate system to the reference coordinate system.

4. Set the command method to absolute.

5 Default Mode and Coordinate System

Various modes and coordinate systems can be used when a G-language program is executed. The following table shows the default settings for these.

The following defaults will always be set before executing the first block of a G-language main program.

Coordinate system or mode		General description
Coordinate system	Reference coordinate system (Default setting)	Axis positions on the reference coordinate system are used.
	Workpiece coordinate system	Axis positions on the workpiece coordinate system are used.
Command Mode	Absolute specification (Default setting)	Absolute coordinates are used in axis operations.
	Incremental specification	Relative coordinates are used in axis operations.
Operating Mode	Stop Mode	For interpolation operations, the next operation is performed only after positioning is completed.
	Pass Mode (Default setting)	For interpolation operations, operations are performed sequentially without decelerating to a stop between them.
	In-position Check OFF Mode	For interpolation operations, the next operation is performed without waiting for positioning to be completed.

G-language Commands

This section provides detailed descriptions of the G-language commands. Command formats, operands, and application are described and programming examples are provided. The command format shows the operands that can be used and shows the correct syntax using the following notation.

< >	Name of an operand to be specified
[ ]	Optional operand
...	An operand that can be specified more than once
●	A required space
_	Optional space

The descriptions provided in this section are for the CS1W-MC421. The CS1W-MC221 supports only the X and Y axes.

1 G00: POSITIONING

Positions up to four axes simultaneously with PTP control at the high-speed feed rate.

Format G00_<Axis movement command ...>[_M<M code>[/Stopover (Note.)]]
[_D<D code>[/Stopover (Note.)]]
[_#<optional number>]

Note The stopover function can be used with either an M code or a D code, but not with both.

Operands The following table shows the possible settings for the operands.

Operand		Possible settings
Axis movement command	Axes	X,Y,Z,U
	Coordinate data	-39,999,999 to +39,999,999 (E00) to (E31) A0000 to A1999 + or -
M code		000 to 999 (E00) to (E31) A0000 to A1999
D code		000(0) to 255 (E00) to (E31) A0000 to A1999
Stopover	Axes	X,Y,Z,U
	Position data	0 to 39999999 (E00) to (E31) A0000 to A1999
Optional number		00 to 19

Note Refer to 7-4 M Code Outputs for details on M codes, and to 7-4-9 D Code Out-puts for details on D codes. Refer to 7-4-10 Stopover Function for details on the stopover function.

Description Positions up to four axes simultaneously with PTP control according to the maximum feed rate, acceleration/deceleration time, and acceleration/deceleration curve settings in the system parameters. The override is valid.

When “+” or “–” is specified for the coordinate data, positioning will be performed to the software limit in the specified direction.

Control is passed on to the next block after all of the specified axes reach their in-position ranges.

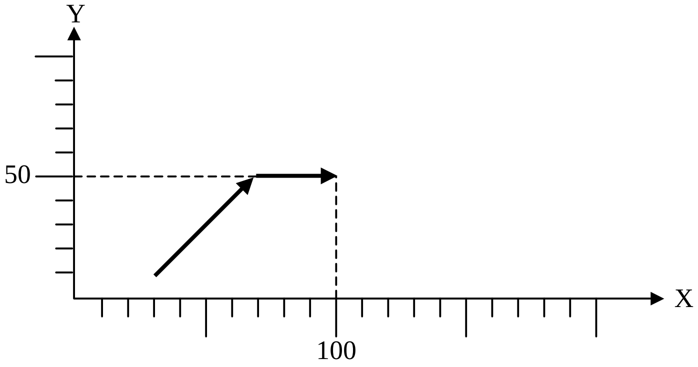
When an optional number is specified, the axis movement command will be carried out when that optional input goes ON. The operation will not be performed until the optional input goes ON. The source of the optional input depends on the optional number specified, as shown below.

0 through 15: Inputs from the PC Interface Area
16 through 19: Inputs from general inputs 1 to 4

Programming Examples

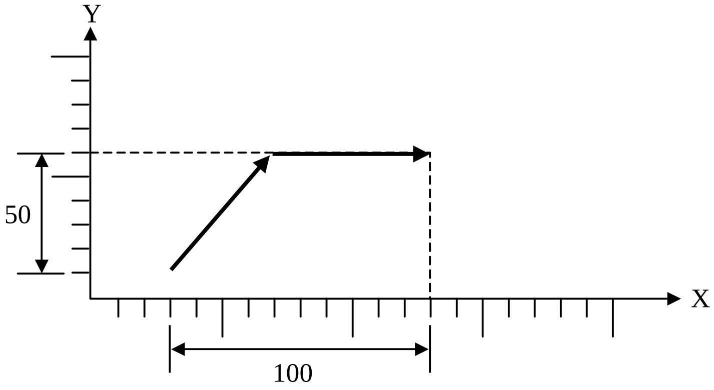
Use the following format for subroutines.

```
        :         :
N010    G90                         Absolute specification
N011    G00       X100    Y50       M001
        :         :
```



The following program is for incremental operation.

```
        :         :
N010    G91                         Incremental specification
N011    G00       X100    Y50       M001
        :         :
```



Note The X and Y axes are operated at the same speed in the above examples.

2 G01: Linear Interpolation

Performs linear interpolation on up to 4 axes simultaneously at the specified interpolation feed rate.

Format G01_<Axis movement command ...> [_F<speed reference>]
[_M<M code>[/Stopover (Note.)]]
[_D<D code>[/Stopover (Note.)]]
[_#<optional number>]

Note The stopover function can be used with either an M code or a D code, but not with both.

Operands The following table shows the possible settings for the operands.

Operand		Possible settings
Axis movement command	Axes	X,Y,Z,U
	Coordinate data	-39,999,999 to +39,999,999 (E00) to (E31) A0000 to A1999
Speed reference		0.0001 to 39,999,999 (E00) to (E31) A0000 to A1999
M code		000 to 999 (E00) to (E31) A0000 to A1999
D code		000(0) to 255 (E00) to (E31) A0000 to A1999
Stopover	Axes	X,Y,Z,U
	Coordinate data	0 to 39999999 (E00) to (E31) A0000 to A1999
Optional number		00 to 19

Note Refer to 7-4 M Code Outputs for details on M codes, and to 7-4-9 D Code Outputs for details on D codes. Refer to 7-4-10 Stopover Function for details on the stopover function.

Description Performs linear interpolation on up to four axes simultaneously at the specified interpolation feed rate.
The interpolation feed rate is calculated with the equations shown below. Four axes are shown, but the same calculations apply when three or fewer axes are being used.

$$Fx = F \text{ x } dLx/L$$
$$Fy = F \text{ x } dLy/L$$
$$Fz = F \text{ x } dLz/L$$
$$Fu = F \text{ x } dLu/L$$

Fx to Fu: Interpolation feed rates for axes X to U
dLx to dLu: Movement distance for axes X to U
L: Total distance between start and end of linear interpolation

$$\left(L = \sqrt{dLx^2 + dLy^2 + dLz^2 + dLu^2} \right)$$

If the interpolation feed rate is not specified, the interpolation will be performed at the last feed rate used in a G01, G02, or G03 command. An error will occur if the interpolation feed rate is not specified and the G01, G02, or G03 command has not been executed before.

⚠ Caution When linear interpolation is performed in Pass Mode and an M code from 0 through 499 or an optional number is specified, the interpolation will be performed in Stop Mode, not Pass Mode. For details on the Pass Mode, refer to 7-3-5 Pass Mode.

If the same interpolation acceleration/deceleration times and override values are not set for all of the axes used in a task, the settings for the axis with the highest priority will be used. Axis X has the highest priority, followed by Y, Z, and U.

The override value can be specified from the Teaching Box or PC Interface Area.

Declared axis configuration for task	Interpolation deceleration time and override value
XYZU	X-axis time and value
YZU	Y-axis time and value
ZU	Z-axis time and value
U	U-axis time and value
XU	X-axis time and value

Refer to 6-8 Operating Modes for details on interpolation acceleration and deceleration times in Pass Mode.

The override value is read only for the first G01 command in Pass Mode or In-Position Check OFF Mode.

When the product of the specified interpolation feed rate and override exceeds the maximum interpolation feed rate specified in system parameters for the axis being moved by linear interpolation, linear interpolation will be performed at the lowest maximum interpolation feed rate among the axes being moved.

Consider the following situation.

```
G01     X1000    Y1000    Z–1000    F3000
X-axis override value:              100.0 [%]
X-axis maximum interpolation feed rate: 4000 [mm/s]
Y-axis maximum interpolation feed rate: 3000 [mm/s]
Z-axis maximum interpolation feed rate: 2000 [mm/s]
```

Interpolation feed rate after override calculation =
3000 x 100.0/100.0=3000 [mm/s]

This exceeds the maximum interpolation feed rate for the Z axis. The feed rate in this case would be 2000 [mm/s].

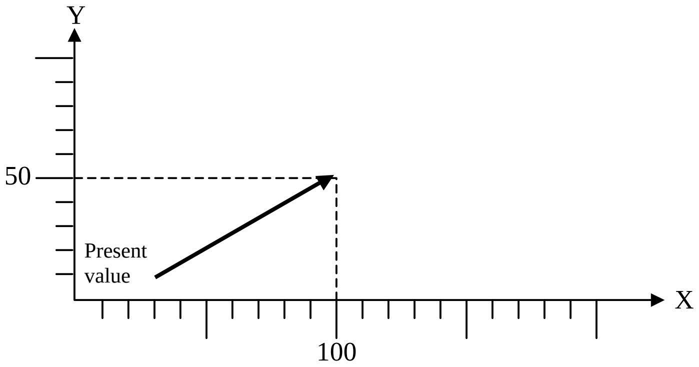
When an optional number is specified, the axis movement command will be carried out when that optional input goes ON. The operation will not be performed until the optional input goes ON. The source of the optional input depends on the optional number specified, as shown below.

0 through 15: Inputs from the PC Interface Area
16 through 19: Inputs from general inputs 1 to 4

Programming Examples

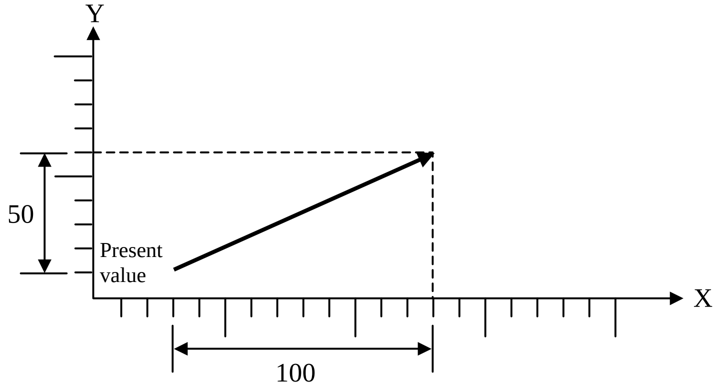
The following example shows absolute positioning.

```
        :        :
N010    G90                       Absolute specification
N011    G01      X100    Y50      F300
        :        :
```



The following program is for incremental operation.

```
          :          :
N010      G91                          Incremental specification
N011      G01        X100     Y50      F300
          :          :
```



3 G02 and G03: Circular Interpolation

Performs two-axis circular interpolation in the clockwise (G02) or counterclockwise (G03) direction at the specified interpolation feed rate. With the CS1W-MC421, it is also possible to perform 2-axis circular interpolation + 1-axis linear interpolation (helical circular interpolation) and 2-axis circular interpolation + 2-axis linear interpolation. In addition, the multiturn function can be added to either circular interpolation or helical circular interpolation for winding machine operations.

Format

```
G02 (G03)_[<Axis movement command ...>]_<center coordinate ...>
                                         [_F<speed reference>]
                                         [_M<M code>[/Stopover (Note.)]]
                                         [_D<D code>[/Stopover (Note.)]]
                                         [_L<number of turns>]
                                         [_#<optional number>]

G02 (G03)_<Axis movement command ...>_<R radius>
                                         [_F<speed reference>]
                                         [_M<M code>[/Stopover (Note.)]]
                                         [_D<D code>[/Stopover (Note.)]]
                                         [_L<number of turns>]
                                         [_#<optional number>]
```

Note The X and Y axes are operated at the same speed in the above examples.

Examples

```
Example 1)  G02 X100 Y90 I0 J40 F300          (Center specification)
Example 2)  G02 X100 Y90 R120 F300            (Radius specification)
Example 3)  G02 X200 Y200 Z200 I0 J0 F200 L10 (Helical circular)
```

Operands The following table shows the possible settings for the operands.

Operand		Possible settings
Axis movement command	Axes	X,Y,Z,U
	Coordinate data	-39,999,999 to +39,999,999 (E00) to (E31) A0000 to A1999
Center coordinate	Axes	I, J, K, H
	Radius	R
	Coordinate and radius data	–39,999,999 to +39,999,999 (E00) to (E31) A0000 to A1999
Speed reference		0.0001 to 39,999,999 (E00) to (E31) A0000 to A1999
M code		000(0) to 999 (E00) to (E31) A0000 to A1999
D code		000(0) to 255 (E00) to (E31) A0000 to A1999
Stopover	Axes	X,Y,Z,U
	Coordinate data	0 to 39,999,999 (E00) to (E31) A0000 to A1999
Number of truns		0 to 39,999,999 A0000 to A1999 (E00) to (E31)
Optional number		00 to 19

I, J, K, and H are the center coordinates (relative position) for the X, Y, Z, and U axes respectively.

Note Refer to 7-4 M Code Outputs for details on M codes, and to 7-4-9 D Code Outputs for details on D codes. Refer to 7-4-10 Stopover Function for details on the stopover function.

Operation This command is used to position two axes by circular interpolation, two axes by circular interpolation and one axis by linear interpolation, or two axes by circular interpolation and two axes by linear interpolation at the specified feed rate. Positioning two axes by circular interpolation and one axis by linear interpolation is referred to as helical circular interpolation. Axes that are not being moved by circular interpolation are called supplemental axes.

Axes that are being moved by circular interpolation are determined by CIRCULAR PLANE SPECIFICATION commands (G17 to G22). With axes specified as the circular plane, the present position of an axis with no axis run command will be treated as the target position. Other axes will not be moved unless axis movements are specified for them.

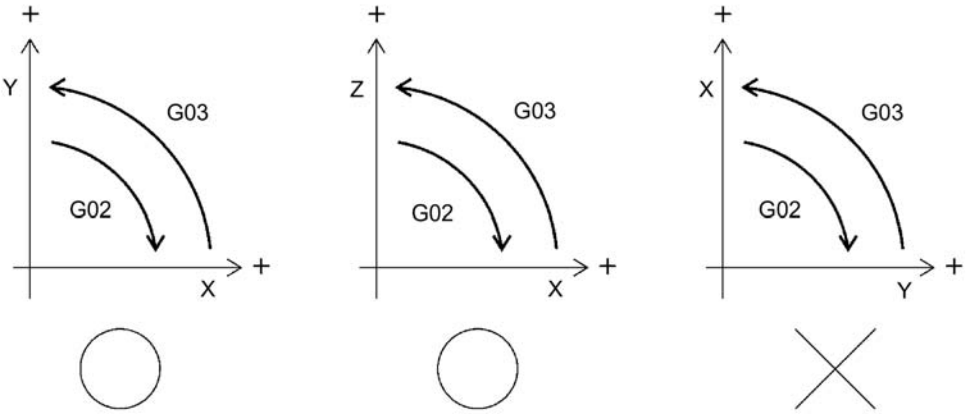
If the interpolation feed rate is not specified, circulation interpolation will be per-formed at the last interpolation feed rate specified for G01, G02 and G03. An error will occur if an interpolation feed rate is not specified when G02 or G03 is first executed.

An error will occur if circular plane is not specified prior to executing circular interpolation.

The axis with the highest priority must be the horizontal axis. The following diagrams show the order of priority.

High ←——→ Low

X Y Z U



If an axis that is not defined as the circular plane is specified, that axis will move by linear interpolation, and the speed of the supplemental axis will be as follows:

Fi = F x dLi/Lc

Fi: Supplemental axis interpolation feed rate
F: Speed reference
dLi: Supplemental axis travel distance
Lc: Circumferential length

An error will occur if the interpolation feed rate of the supplemental axis exceeds the maximum interpolation feed rate set for the axis.

When specifying the circle’s center coordinate, specify the relative distance from the present position (the circle’s starting point) and specify one or two axes. An error will occur if nothing is specified or if three or more axes are specified.

When specifying the radius, circular interpolation smaller than a semicircle will be executed for a positive value and circular interpolation greater than a semi-circle will be executed for a negative value. Either a negative or positive value can be specified for a semicircle.

When specifying a complete circle (360°), the system will operate when either of the following conditions is met in the center coordinate specifications.

- When the axis movement command and present position are the same

- When the axis movement command is omitted

A complete circle cannot be drawn using the radius specification.

⚠ **Caution** When circular interpolation is performed in Pass Mode and an M code from 0 through 499 or an optional number is specified, the interpolation will be per-formed in Stop Mode, not Pass Mode. For details on the Pass Mode, refer to 7-3-5 Pass Mode.

The override value can be set from either the Teaching Box or the PC Interface Area.

The order of priority is as follows:

High ←→ Low
X Y Z U

Declared axis configuration for task	Interpolation deceleration time and override value
XYZU	X-axis time and value
YZU	Y-axis time and value
ZU	Z-axis time and value
U	U-axis time and value
XU	X-axis time and value

In Pass Mode or In-Position Check OFF Mode, the override value is read only for the first G02/G03 command.

When the product of the specified interpolation feed rate and override value exceeds the maximum interpolation feed rate for the axis being moved by circular interpolation (set in the system parameters), circular interpolation will be performed at the lowest maximum feed rate among the axes being moved. For example, consider the following situation.

```
G02    X100    Y90    I0    J40    F3000
```

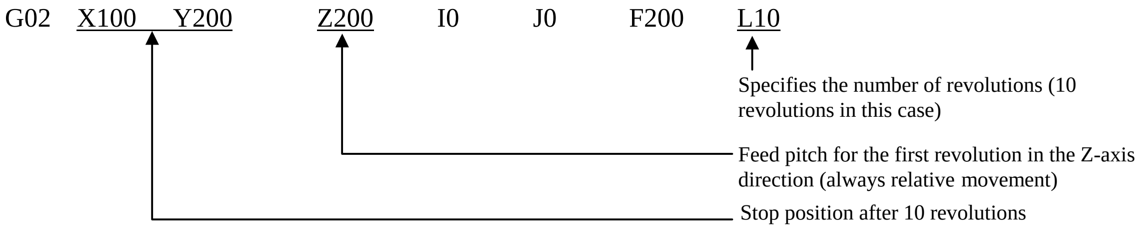
X-axis override value: 100.0 [%]
X-axis max. interpolation feed rate: 4000 [mm/s]
Y-axis max. interpolation feed rate: 2000 [mm/s]
Interpolation feed rate after override calculation = 3000 × 100.0/100.0
= 3000 [mm/s]
The interpolation feed rate of 3000 [mm/s] exceeds the maximum interpolation feed rate for the Y-axis (2000 [mm/s]), so the interpolation feed rate is set to 2000 [mm/s].

Note Internal radius calculations are accurate to seven digits. When the radius exceeds 9,999,999 pulses, circular interpolation will be performed within the specified circle. Keep the radius under 9,999,999 pulses when specifying the radius in 1-pulse units.
If the plane where circular interpolation will be performed is changed while consecutive circular interpolations are being executed, pause the positioning before going on to the next circular interpolation when Pass Mode is specified.
When an optional number is specified, the axis movement command will be carried out when that optional input goes ON. The operation will not be performed until the optional input goes ON. The source of the optional input depends on the optional number specified, as shown below.

0 through 15: Inputs from the PC Interface Area
16 through 19: Inputs from general inputs 1 to 4

Multiturn Function

The multiturn function was added to conventional circular and helical circular interpolations for the purpose of wiring machine operations.

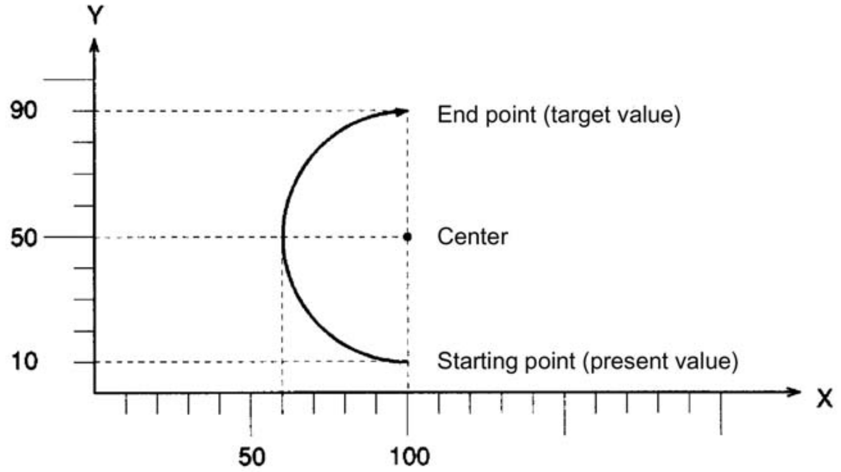

G02 X100 Y200 Z200 I0 J0 F200 L10
Specifies the number of revolutions (10 revolutions in this case)
Feed pitch for the first revolution in the Z-axis direction (always relative movement)
Stop position after 10 revolutions

Note If the multiturn function is specified for a complete circle, the axis will move the number of turns plus one revolution.

Programming Examples

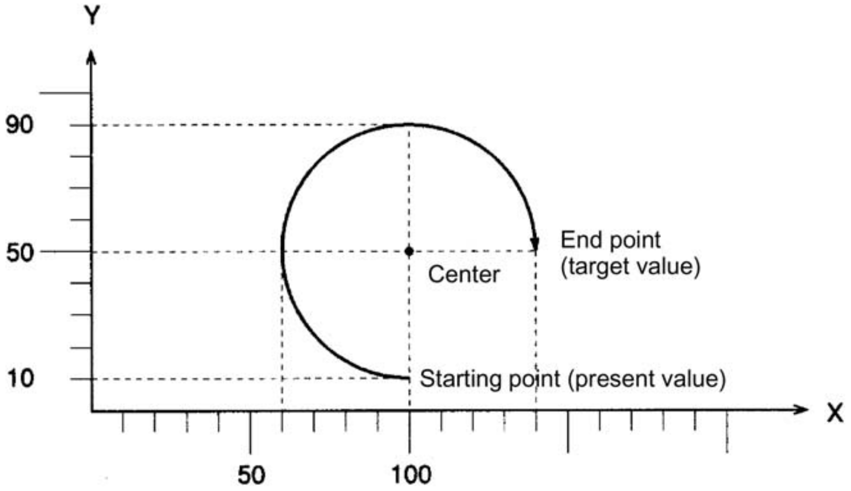
The following programming example shows circular interpolation with center coordinate specification.

```
        :       :
N010    G90                       Absolute specification
N011    G17                       Makes X-Y the circular plane
N010    G02     X100    Y90     I0     J40     F300
        :       :
```



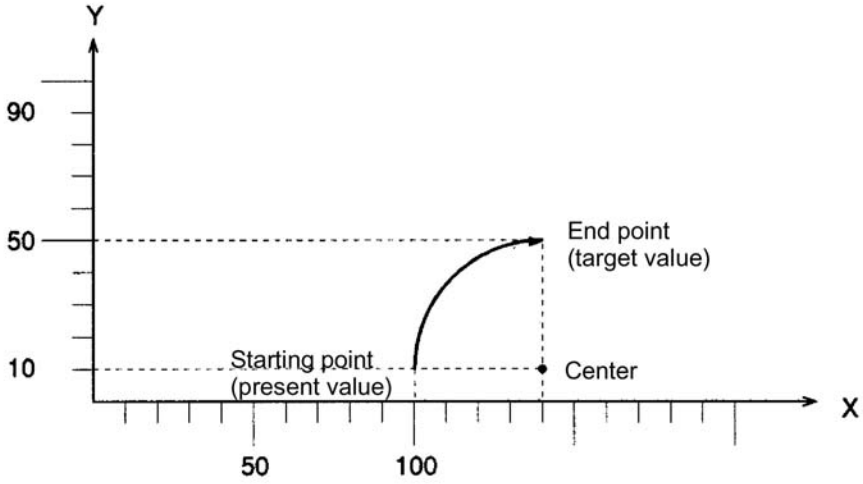
The following program shows circular interpolation with radius specification (R<0). An arc greater than a semicircle will be drawn when R<0.

```
        :       :
N010    G90                       Absolute specification
N011    G17                       Makes X-Y the circular plane
N012    G02     X140    Y50     R-40    F300
        :       :
```



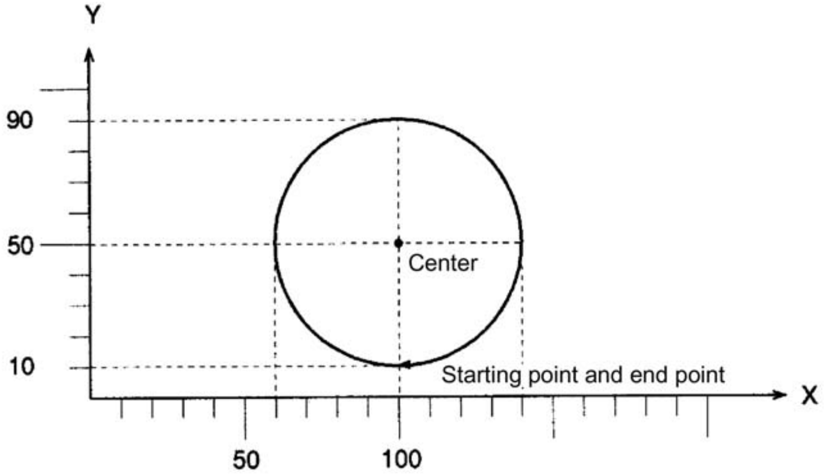
The following program shows circular interpolation with radius specification (R>0). An arc smaller than a semicircle will be drawn when R>0.

```
        :        :
N010    G91                       Incremental specification
N011    G17                       Makes X-Y the circular plane
N012    G02      X40     Y40      R40     F300
        :        :
```



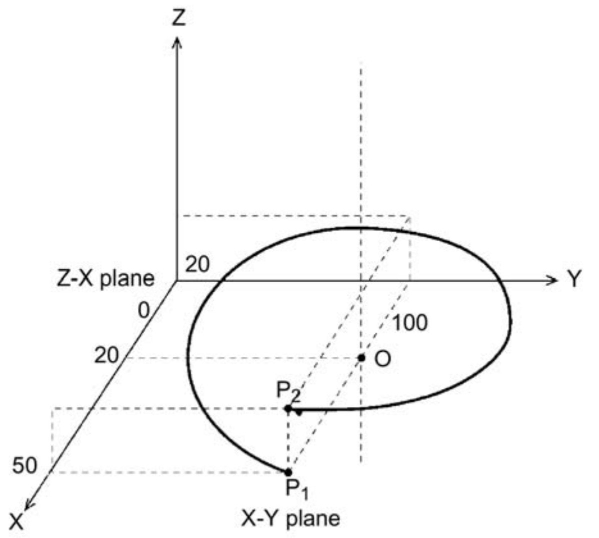
The following program shows circular interpolation of a complete circle.

```
        :        :
N010    G90                         Absolute specification
N011    G17                         Makes X-Y the circular plane
N012    G02      X100    Y10      I0      J40      F500
        :        :
```



Helical Movement

```
        :       :
N010    G90                             Absolute specification
N011    G17                             X-Y the circular plane specification
N012    G02     X50     Y100    Z20     I-30    J0      F500
        :       :
```



Start point coordinate P_1 (50, 100, 0)
End point coordinate P_2 (50, 100, 20)
Center coordinate O (20, 100, 0)

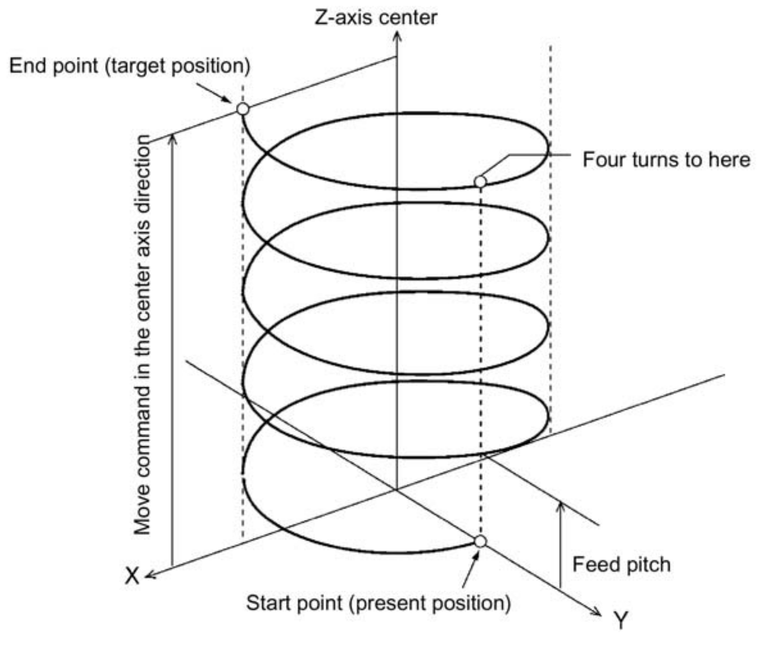
Helical Bundling Operation

```
N010 G90 ..........................................Absolute specification
N011 G17 ..........................................X-Y circular plane specification
N012 G02   X100   Y0       Z20      I0      J0      F200      L4
```

L4: Specifies the number of revolutions (4 revolutions in this case)

Z20: Feed pitch for the first revolution in the Z-axis direction (always relative movement)

X100 Y0: Stop position after 10 revolutions


Z-axis center
End point (target position)
Four turns to here
Move command in the center axis direction
Feed pitch
X
Y
Start point (present position)

4 G04: Dwell Timer

Waits for a specified length of time.

Format G04● <wait time>

Operands The following table shows the possible settings for the wait time operand.

Operand	Possible settings
Wait time	0.001 to 39,999.994 (E00) to (E31) A0000 to A1999

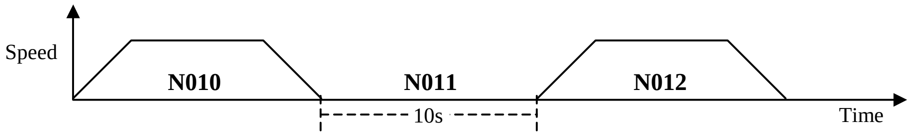
Description This command waits for a specified wait time.
The wait time can be specified by a register or a position data address. If the memory contents have four or more digits below the decimal point, the fourth digit is rounded off. An error will occur if the resulting number is outside of the acceptable range (0.001 to 39,999.994).
Values from 0.001 to 39,999.999 [s] can be input for the wait time, but settings of 39,999.995 to 39,999.999 will cause an error. The accuracy of the timer is approximately 0.02 s. For example, setting “1” [s] will result in an actual wait of 1.00 to 1.02 seconds.

⚠ **Caution** If this command is executed just after a G01, G02, G03, or G32 command, the interpolation will be performed in Stop Mode, not Pass Mode, even if Pass Mode has been selected.

Programming Example

In this example, the dwell timer waits 10 seconds between linear interpolation operations.

```
        :       :
N009    G10
N010    G01     X100    Y100
N011    G04     10
N012    G01     X200    Y200
        :       :
```



Note Program blocks N010 and N012 are not pass operations.

5 G10: Pass Mode

Switches the operating mode to Pass Mode.

Format G10

Operands None

Description This command switches the operating mode to Pass Mode. When this command is executed, any subsequent interpolation (linear or circular) operation will move on to the next positioning operation without decelerating to a stop. The Pass Mode remains in effect until the STOP MODE command (G11) or IN-POSI-TION CHECK OFF MODE command (G13) is executed.
The pass operation may not be executed if a G01, G02, G03, or G32 command is executed with an extremely short travel time.
In Pass Mode, the next G-code command is executed (pre-read execution) without waiting for the completion of a G01, G02, G03, or G32 command that is currently being executed, so the execution block number output to the interface might be different from the actual block number being executed. The subsequent commands will be executed until the next block with a G01, G02, G03, or G32 command.

In the following programming example, blocks N003 to N006 are executed (pre-read execution) while the axis is moving according to the command in block N002. Although the axis is moving according to program block N002, the execution block number will be refreshed with block numbers N003 to N006. Program execution will then wait on standby at block N006.
When command processing for N002 is completed, pre-read execution will start again.

```
N000    P000    XY
N001    G10
N002    G01     X100    F1000
N003    G63     A0000   = 1
N004    G63     A0001   = 2
N005    G63     A0002   = 3
N006    G01     X200
        :       :
```

⚠ **Caution** The following commands will pause pre-reading and switch any axis that is moving to Stop Mode. (The axis will decelerate to a stop when positioning is completed, and an in-position check will be performed).

- G00 (PTP)
- G01, G02 and G03 commands with #optional numbers
- CIRCULAR INTERPOLATION commands (G02, G03) with different circular plane specifications than those for CIRCULAR PLANE SPECIFICATION commands (G17 to G22)
- Pass Mode operation between CIRCULAR INTERPOLATION commands (G02, G03) and TRAVERSE (G32) (which is not possible)
- G04 (DWELL TIMER)
- G26 (WORKPIECE ORIGIN RETURN)
- G27 (REFERENCE ORIGIN RETURN)
- G28 (ORIGIN SEARCH)
- G29 (ORIGIN UNDEFINED)
- G30 (SPEED CONTROL)
- G31 (INTERRUPT FEEDING)
- G54 (CHANGE REFERENCE COORDINATE SYSTEM PV)
- G79 (PROGRAM END)
- M000 to M499 independent commands
- D000 to D256

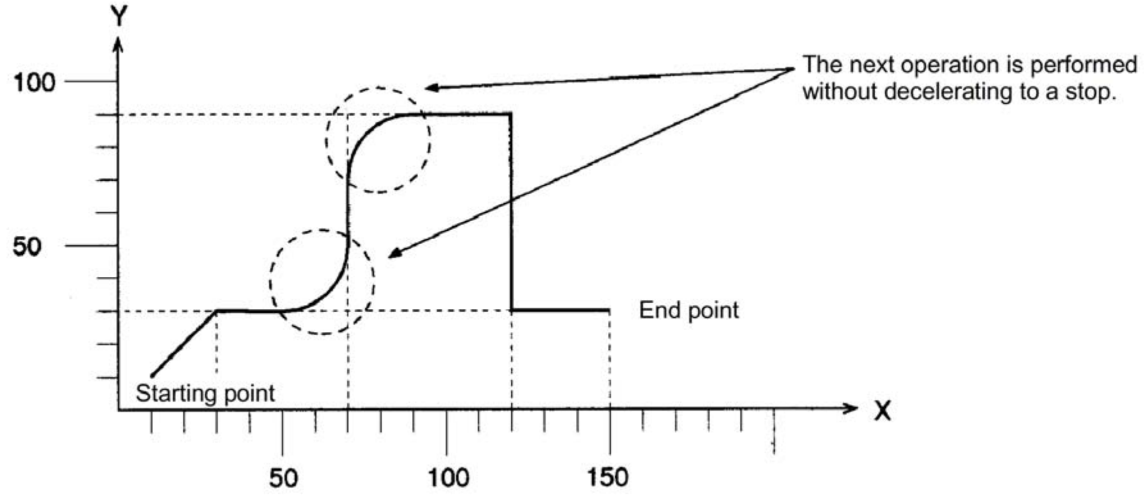
The following commands must be executed in Stop Mode. Pre-reading will be paused.

- G01, G02, G03, G32 commands with an M code from M000 to M499.

Programming Example

In this example, linear interpolation is executed with a pass operation.

```
        :       :
N010    G00     X30     Y30
N011    G10
N012    G01     X70
N013    G01     Y90
N014    G01     X120
N015    G00     Y30
N016    G00     X150
        :       :
```


Y
100
50
X
50
100
150
The next operation is performed without decelerating to a stop.
End point
Starting point

6 G11: Stop Mode

Switches the operating mode to Stop Mode.

Format G11

Operands None

Description This command switches the operating mode to Stop Mode. If this command is executed, subsequent interpolation (linear or circular) commands will be decelerated to a stop at the end point, an in-position check will be performed, and then the next positioning operation will be performed after completion of the positioning is verified.

The Stop Mode will remain in effect until one of the following commands is executed:

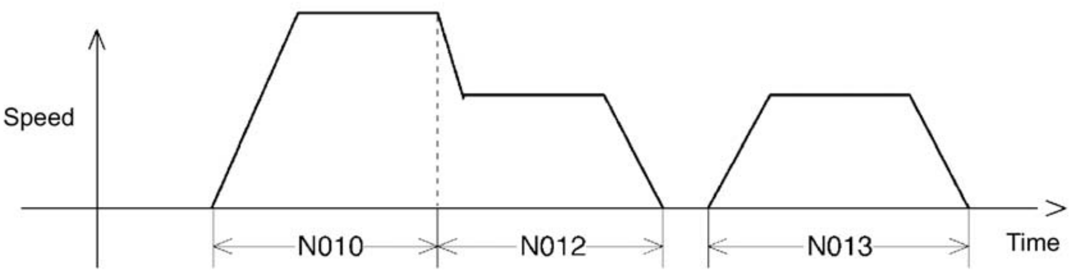
1,2,3...
1. PASS MODE (G10)
2. IN-POSITION CHECK OFF MODE (G13)
3. REFERENCE ORIGIN RETURN (G26)
4. WORKPIECE ORIGIN RETURN (G27)
5. ORIGIN SEARCH (G28)

Note A reset will be performed and the operating mode will be switched to Pass Mode when a REFERENCE ORIGIN RETURN (G26), WORKPIECE ORIGIN RETURN (G27), or ORIGIN SEARCH (G28) command is executed.

Programming Example

The following programming example demonstrates positioning in Stop Mode. The mode is changed to Stop Mode after linear interpolation in Pass Mode. The Unit moves to the next operation after completion of the previous operation is confirmed.

```
        :      :
N009    G10
N010    G01    X200    F200
N011    G11
N012    G01    X300    F100
N013    G01    X400    F100
        :      :
```



In Stop Mode, the Unit moves to the next operation after completion of the previous operation is verified.

7 G13: In-position Check Off Mode

Starts the next positioning operation without waiting for the current one to be completed.

Format G13

Operands None

Description In Stop Mode, the next positioning operation will start only after positioning is completed, but with the In-position Check OFF Mode, the next positioning operation will start without waiting for positioning to be completed.

The following G codes can be used in In-position Check OFF Mode.
- G01 (LINEAR INTERPOLATION)
- G02/G03 (CIRCULAR INTERPOLATION)
- G32 (TRAVERSE)

The In-position Check OFF Mode remains in effect until one of the following is executed.
- PASS MODE (G10)
- STOP MODE (G11)
- REFERENCE ORIGIN RETURN (G26)
- WORKPIECE ORIGIN RETURN (G27)
- ORIGIN SEARCH (G28)

The following commands will pause pre-reading and shift any axis that is moving to Stop Mode. (The axis will decelerate to a stop when positioning is completed, and an in-position check will be performed).
- G00 (PTP)
- G01, G02 and G03 commands with #optional numbers
- CIRCULAR INTERPOLATION commands (G02, G03) with different circular plane specifications than those for CIRCULAR PLANE SPECIFICATION commands (G17 to G22)
- TRAVERSE (G32) after CIRCULAR INTERPOLATION commands (G02, G03)

- G04 (DWELL TIMER)
- G26 (WORKPIECE ORIGIN RETURN)
- G27 (REFERENCE ORIGIN RETURN)
- G28 (ORIGIN SEARCH)
- G29 (ORIGIN UNDEFINED)
- G30 (SPEED CONTROL)
- G31 (INTERRUPT FEEDING)
- G54 (CHANGE REFERENCE COORDINATE SYSTEM PV)
- G79 (PROGRAM END)
- M000 to M499 as independent commands
- D000 to D256

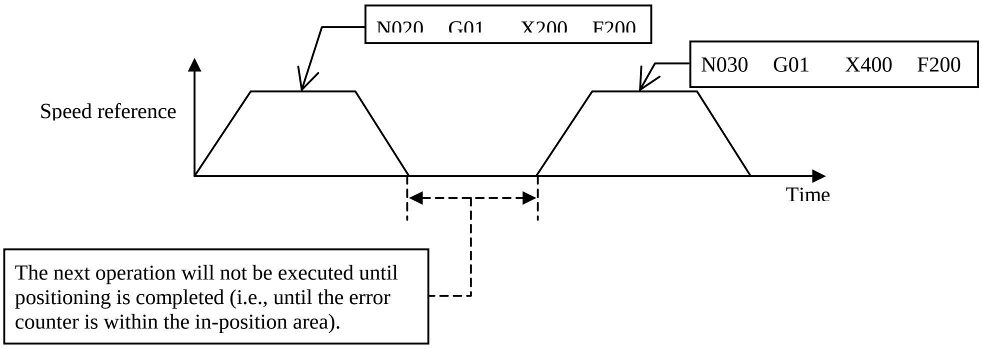
The following commands must be executed in Stop Mode. Pre-reading will be paused.

- G01, G02, G03, G32 commands with an M code from M000 to M499 (including stopover)

Programming Examples

In this programming example, positioning is executed in Stop Mode.

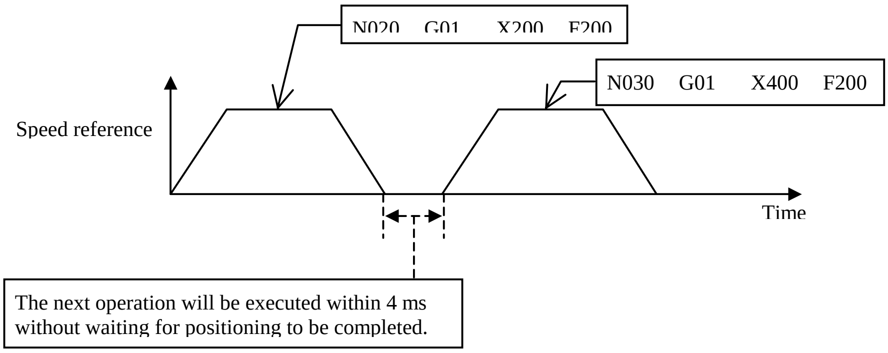
```
        :       :
N010    G11                     Switch to Stop Mode
N020    G01     X200    F200
N030    G01     X400    F200
        :       :
```



The next operation will not be executed until positioning is completed (i.e., until the error counter is within the in-position area).

In this programming example, the In-position Check OFF Mode is enabled.

```
        :       :
N010    G13             Switch to In-position check OFF Mode
N020    G01     X200    F200
N030    G01     X400    F200
        :       :
```



The next operation will be executed within 4 ms without waiting for positioning to be completed.

8 G17 to G22: Circular Plane Specification

Specifies the plane in which circular interpolation is to be performed.

Format

G17
G18 (See note.)
G19 (See note.)
G20 (See note.)
G21 (See note.)
G22 (See note.)

Note G18 to G22 are not available with the CS1W-MC221.

Operands None

Description This command specifies the plane in which circular interpolation is performed, as shown in the following table.

Code	Plane
G17	Specifies the X-Y plane.
G18	Specifies the X-Z plane.
G19	Specifies the Y-Z plane.
G20	Specifies the X-U plane.
G21	Specifies the Y-U plane.
G22	Specifies the Z-U plane.

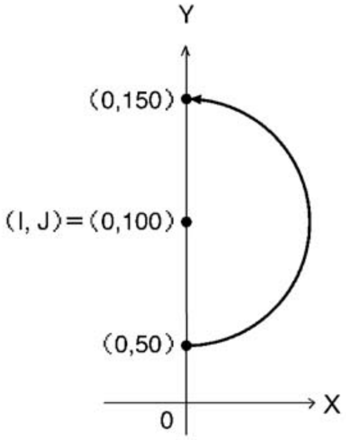
The setting remains in effect until a new plane is specified.

The initial setting is for the X-Y plane.

Programming Examples 1

The following programming example performs circular interpolation in the X-Y plane.

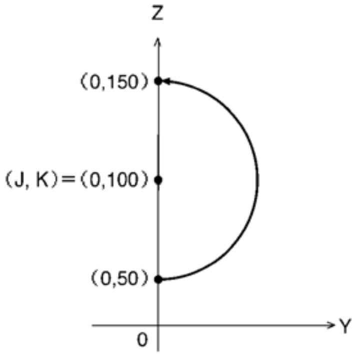
```
          :         :
N010      G90
N011      G17
N012      G00       X0        Y50        Z0
N013      G03       X0        Y150       R50       F300
```



Programming Examples 2

The following example performs circular interpolation in the Y-Z plane.

```
        :        :
N010    G90
N011    G17
N012    G00      X0       Y50      Z50
N013    G03      Y0       Z150     J0       K100     F300
```



9 G26: Reference Origin Return

Moves to the reference origin

Format G26_<Axis name ... > [_M<M code>[/Stopover (Note.)]]
[_D<D code>[/Stopover (Note.)]]

Note The stopover function can be used with either an M code or a D code, but not with both.

Operands The following table shows the possible settings for the operands.

Operand		Possible settings
Axis names		X,Y,Z,U
M code		000 to 999 (E00) to (E31) A0000 to A1999
D code		000(0) to 255 (E00) to (E31) A0000 to A1999
Stopover	Axes	X,Y,Z,U
	Coordinate data	0 to 39999999 (E00) to (E31) A0000 to A1999

Note Refer to 7-4 M Code Outputs for details on M codes, and to 7-4-9 D Code Outputs for details on D codes. Refer to 7-4-10 Stopover Function for details on the stopover function. Refer to Section 10 Establishing the Origin for details on origin searches.

Description This command moves the specified axes to the reference origin by PTP control. The coordinate system and mode settings will be as follows after the return to the reference origin:

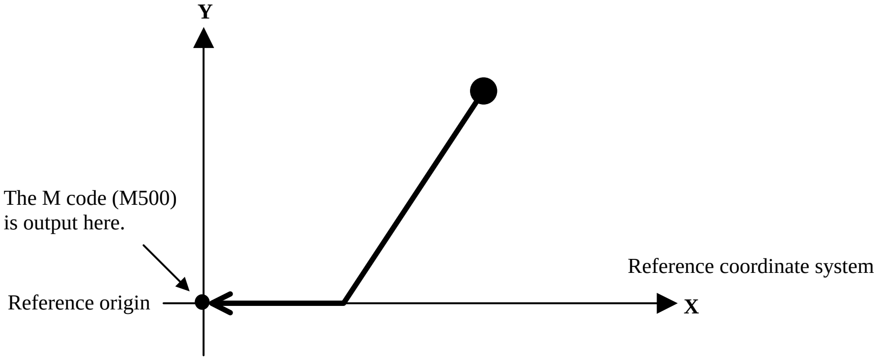
1,2,3...
1. The coordinate system will be set to the reference coordinate system.
2. The command mode will be set to absolute specification.
3. The operating mode will be set to Pass Mode.

The settings will not be switched to the ones above if the G26 command is terminated while in progress by the OPTIONAL END command (G74), FORCED BLOCK END, or other command. The command is also considered to be in progress while waiting for the M code reset.

Programming Example

The following example moves the X and Y axes to the reference origin.

```
        :      :
N010    G26    XY      M500
        :      :
```


Y
The M code (M500) is output here.
Reference origin
Reference coordinate system
X

10 G27: Workpiece Origin Return

Moves to the workpiece origin.

Format G27_<Axis name ... > [_M<M code>[/Stopover (Note.)]]
[_D<D code>[/Stopover (Note.)]]

Note The stopover function can be used with either an M code or D code, but not with both.

Operands The following table shows the possible settings for the operands.

Operand		Possible settings
Axis names		X,Y,Z,U
M code		000 to 999 (E00) to (E31) A0000 to A1999
D code		000(0) to 255 (E00) to (E31) A0000 to A1999
Stopover	Axes	X,Y,Z,U
	Coordinate data	0 to 39999999 (E00) to (E31) A0000 to A1999

Note Refer to 7-4 M Code Outputs for details on M codes, and to 7-4-9 D Code Outputs for details on D codes. Refer to 7-4-10 Stopover Function for details on the stopover function.

Description This command moves the specified axes to the workpiece origin by PTP control. The coordinate system and mode settings will be as follows after the return to the workpiece origin:

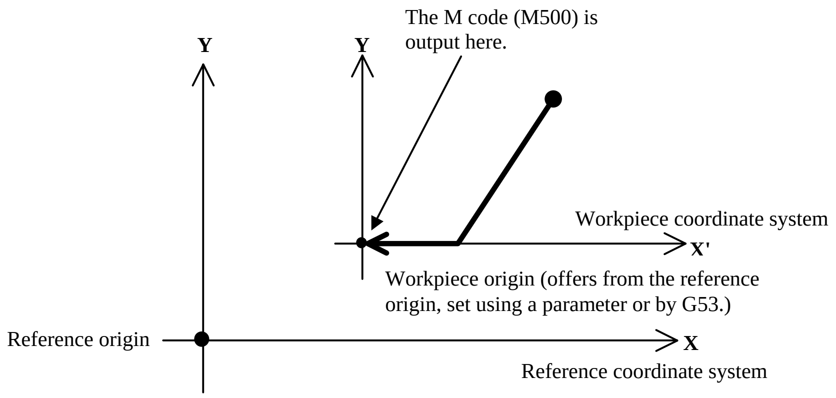
1,2,3... 1. The coordinate system will be set to the workpiece coordinate system.
2. The command mode will be set to absolute specification.
3. The operating mode will be set to Pass Mode.

The settings will not be switched to the ones above if the G27 command is terminated while in progress by the OPTIONAL END command (G74), FORCED BLOCK END, or other command. The command is also considered to be in progress while waiting for the M code reset.

Programming Example

The following example moves the X and Y axes to the workpiece origin.

```
        :          :
N010    G27        XY       M500
        :          :
```


The M code (M500) is output here.
Y
Y
Workpiece coordinate system
X'
Workpiece origin (offers from the reference origin, set using a parameter or by G53.)
Reference origin
X
Reference coordinate system

11 G28: Origin Search

Performs an origin search in the specified axes.

Format G28_<Axis name ... >_[M<M code>[/Stopover (Note.)]]
[_D<D code>[/Stopover (Note.)]]

Note The stopover function can be used with either an M code or D code, but not with both.

Operands The following table shows the possible settings for the operands.

Operand		Possible settings
Axis names		X,Y,Z,U
M code		000 to 999 (E00) to (E31) A0000 to A1999
D code		000(0) to 255 (E00) to (E31) A0000 to A1999
Stopover	Axes	X,Y,Z,U
	Coordinate data	0 to 39999999 (E00) to (E31) A0000 to A1999

Note Refer to 7-4 M Code Outputs for details on M codes, and to 7-4-9 D Code Outputs for details on D codes. Refer to 7-4-10 Stopover Function for details on the stopover function. Refer to Section 10 Establishing the Origin for details on the origin search function.

Description This command performs an origin search in the specified axes.

Axes using Incremental Encoders:

The origin search operation searches for the origin in the specified axes according to the settings of the Origin Search Method, Origin Search Direction, Origin Deceleration Method, and Origin Proximity Input Logic parameters.

For axes in origin mode, the present position is set to zero and fixed as the origin when the origin search method is set. For axes in other modes, the origin is fixed after the origin search operation is executed.

Axes using Absolute Encoders:

As an upgrade to previous models, it is now possible to perform an origin search using an absolute encoder just as with an incremental encoder.

The mode will be as follows after the origin search has been completed.

1,2,3... 1. The coordinate system will be set to the reference coordinate system.
2. The command mode will be set to absolute specification.
3. The operating mode will be set to Pass Mode.

The settings will not be switched to the ones above if the G28 command is terminated while in progress by the OPTIONAL END command (G74), FORCED BLOCK END, or other command. The command is also considered to be in progress while waiting for the M code reset.

Programming Example

The following example performs an origin search for the X, Y, and Z axes.

```
        :      :
N010    G28    XYZ
        :      :
```

12 G29: Origin Undefined

Forcibly sets the origin to an undefined state for the specified axes.

Format G29_<Axis name …>

Operands X,Y,Z, and U axes

Description This command forcibly sets the origin to an undefined state for the specified axes.

- The No Origin Flag will turn ON in the PC Interface Area.
- An asterisk (*) will be displayed for axes with undefined origins on the present value display on the Teaching Box.

The present position will not change with this command, but the present position of the machine will be updated continually. The following will occur if the origin is not defined.

- The software limits will be disabled.
- Zones will be disabled depending on the settings of the system parameters (i.e., they will be disabled when the zones depend on the origin being established).
- Positioning will not be allowed using G00, G01, G02, G03, G26 and G27 commands (when the axis mode is Normal Feed Mode).

Perform one of the following operations to re-establish the origin.

- Execute ORIGIN SEARCH (G28).
- Execute CHANGE REFERENCE COORDINATE SYSTEM PV (G54).

Programming Example

The following example sets the X and Y axes to an undefined state.

```
N000   P000   XY
N001   G00    X0     Y0    (Moves the axis to the system origin.)
N002   G29    XY           (Makes the origin undefined.)
N003   G30    X+     Y+    (Sets unlimited feeding.)
       :      :
```

13 G30: Speed Control Axis Feeding

Performs the feed rate control of a maximum of four axes.

Format G30_<axis movement command …>

Operands The following shows the possible settings for the operands.

Axis movement command	
Axis	**Coordinate data**
X,Y,Z,U	+,-

Description The axis is fed in the positive direction if the coordinate data is set to positive.
The axis is fed in the negative direction if the coordinate data is set to negative.
The feed rate of each axis can be set individually.
X-axis feed rate = X-axis maximum feed rate x X-axis override value/100
Y-axis feed rate = Y-axis maximum feed rate x Y-axis override value/100
Z-axis feed rate = Z-axis maximum feed rate x Z-axis override value/100
U-axis feed rate = U-axis maximum feed rate x U-axis override value/100

The feed rate is set with the system parameters.
Override value: Set with in PC Interface Area or Teaching Box within a range of 0.1% to 100.0%.

Example: The following X-axis feed rate will be obtained if the X-axis maximum feed rate is 10,000 mm/s and the X-axis override value is 50.0%.
X-axis feed rate = 10,000 x 50.0 /100 = 5,000 mm/s.

The acceleration time or deceleration time varies with the override value, as follows:

X-axis acceleration time = X-axis acceleration time* x override value/100.
X-axis deceleration time = X-axis deceleration time* x override value/100
Y-axis acceleration time = Y-axis acceleration time* x override value/100.
Y-axis deceleration time = Y-axis deceleration time* x override value/100
Z-axis acceleration time = Z-axis acceleration time* x override value/100
Z-axis deceleration time = Z-axis deceleration time* x override value/100
U-axis acceleration time = U-axis acceleration time* x override value/100
U-axis deceleration time = U-axis deceleration time* x override value /100
*These values are set with the system parameters.

The feed rate can be changed in real time by changing the override value. The operation will vary as follows depending on whether or not the origin is established:

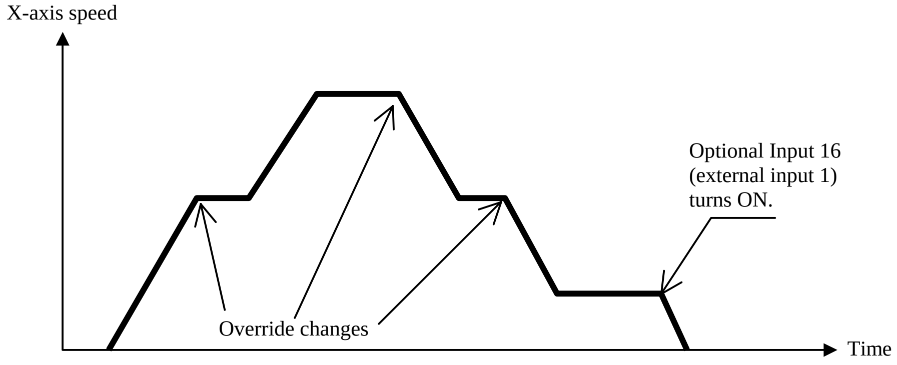
Origin established: The software limit will be enabled and the present position will be always refreshed.

Origin not established: The software limit will be disabled. The present position will be set to 0 at the start of operation if the axis mode is Normal Feed Mode, and always refreshed if the axis mode is Unlimited Feed Mode.

Programming Example

The following example shows executing speed control, adjusting the speed with an override, and stopping with an optional number.

```
        :       :
N010    G74     16
        G30     X+
        :       :
```



Note The preceding G01, G02, G03, and G32 commands will operate in Stop Mode instead of Pass Mode when this command is executed.

14 G31: Interrupt Feeding

Feeds a specified axis for a specified travel distance when a general input turns ON.

Format

G31_<Axis name> <Coordinate 1>[/<Coordinate 2]
[_F <Speed Reference 1>] [_F <Speed Reference 2>]
[_M<M code>[/Stopover (Note.)]]
[_D<D code>[/Stopover (Note.)]]
[_S]

Note The stopover function can be used with either an M code or a D code, but not with both.

Example 1:	G31	X100	F200	
Example 2:	G31	X100	F200	F100
Example 3:	G31	X100/200		F200

Operands The following table shows the possible settings for the operands.

Operand		Possible settings
Axis movement command	Axis names	X,Y,Z,U
	Coordinate Data 1 Coordinate Data 2	–39,999,999 to +39,999,999 (E00) to (E31) A0000 to A1999
Speed Reference 1 Speed Reference 2		0.0001 to +39,999,999 (E00) to (E31) A0000 to A1999
M code		000(0) to 999 (E00) to (E31) A0000 to A1999
D code		000(0) to 255 (E00) to (E31) A0000 to A1999
Stopover	Axes	X,Y,Z,U
	Coordinate data	0 to 39999999 (E00) to (E31) A0000 to A1999

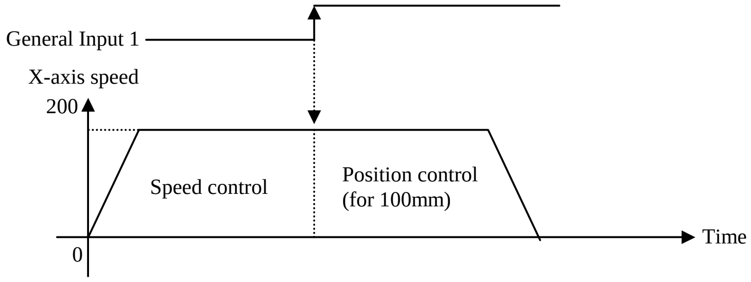
Coordinate Data 1 is the travel distance after the interrupt, and Coordinate Data 2 is the maximum travel distance before the input is received.
Set Speed Reference 1 to a higher value than Speed Reference 2.

Note Refer to 7-4 M Code Outputs for details on M codes, and to 7-4-9 D Code Outputs for details on D codes. Refer to 7-4-10 Stopover Function for details on the stopover function.

Description The specified axis will be moved for the specified distance for positioning after the general input (1 to 4) is turned ON. It is also possible to specify a travel distance for when no general input is received to stop with an error (program execution interrupted) after positioning. An S option can also be used to execute the next block without an error occurring.

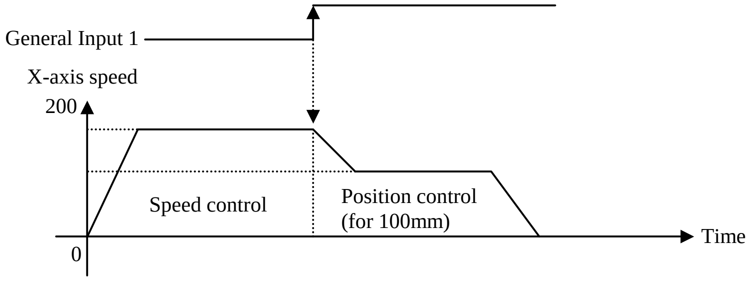
Refer to the following examples.

Example 1: Speed Reference 2 Not Specified
G31 X100 F200 (mm)



The X axis moves at 200 mm/s (speed control) until General Input 1 is turned ON. The X axis then moves for another 100 mm (position control) from the position where General Input 1 was turned ON.

Example 2: Speed Reference 2 Specified
G31 X100 F200 F100 (mm)



The X axis moves at 200 mm/s (speed control) until General Input 1 is turned ON. The speed of the X axis will decelerate to the speed specified by Speed Reference 2 (i.e., 100 mm/s) and the X axis will move for another 100 mm (position control) from the position where General Input 1 was turned ON.

There is no difference in acceleration/deceleration time between the G31 and G00 commands. It is generally set automatically according to the ratio of the speed reference to the maximum feed rate. There are exceptions, however. Refer to the following Operation Examples.

Refer to the following table for the correspondence between interrupt feeding axes and general inputs.

Axis	General input
X	Turning ON of General Input 1
Y	Turning ON of General Input 2
Z	Turning ON of General Input 3
U	Turning ON of General Input 4

The operation will vary as follows depending on whether or not the origin is established.

Origin established: The software limit will be enabled and the present position will be always refreshed.

Origin not established: The software limit will be disabled. The present position will be set to 0 at the start of operation if the axis mode is Normal Feed Mode, and always refreshed if the axis mode is Unlimited Feed Mode.

Note 1. The preceding G01, G02, G03, or G32 command will operate in Stop Mode instead of Pass Mode if this command is executed.

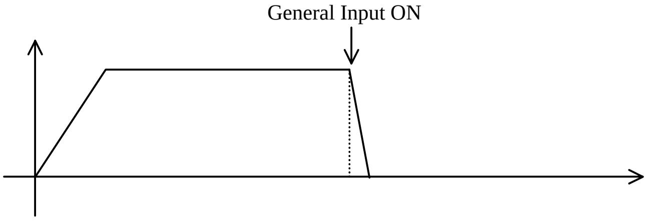
2. An error will result if the G31 command is executed if the value of Speed Reference 2 is the same as or larger than that of Speed Reference 1. Make sure that the value of the Speed Reference 1 is larger than of Speed Reference 2.

3. The value of a speed reference must be the same as or less than the maximum feed rate. If the value is larger than the maximum feed rate, the axis will move at maximum feed rate and the override will be enabled for the speed control but disabled for the position control.

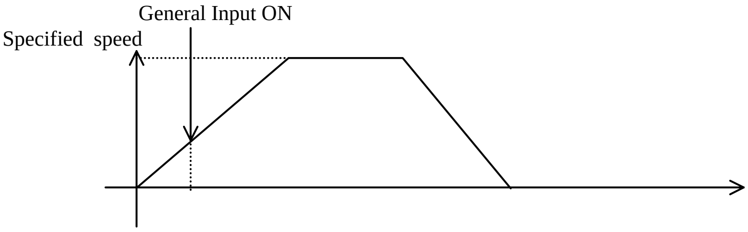
Positioning Curve Examples
The following positioning curves are obtained according to the speed and movement of the axis if a general input is turned ON.

Speed Reference 2 Omitted

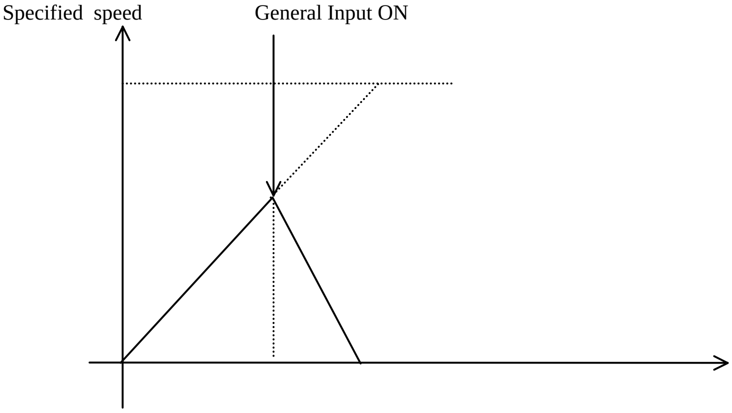
1,2,3… 1. A general input is turned ON while the axis is moving at constant speed, but the deceleration time is extremely short because the speed is high and the movement is small.



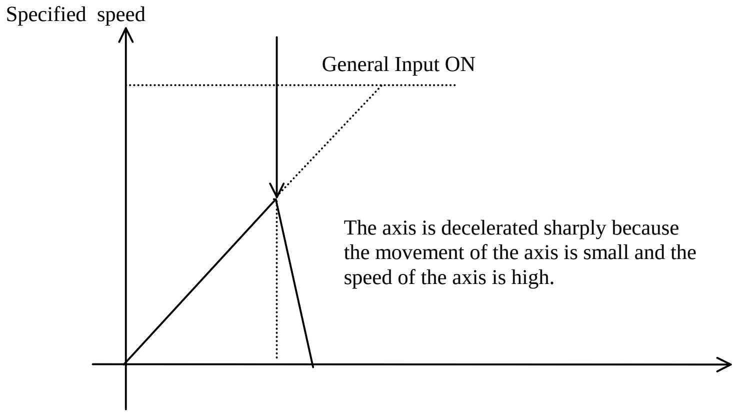
2. A general input is turned ON while the axis is being accelerated. (1)



3. A general input is turned ON while the axis is being accelerated. (2)

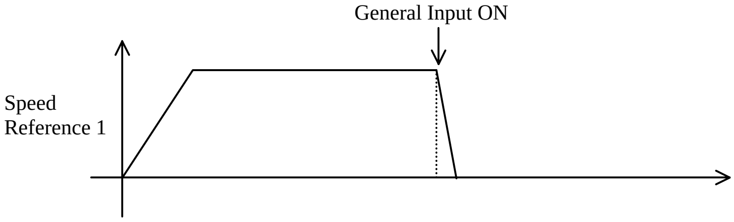


4. A general input is turned ON while the axis is being accelerated. (3)

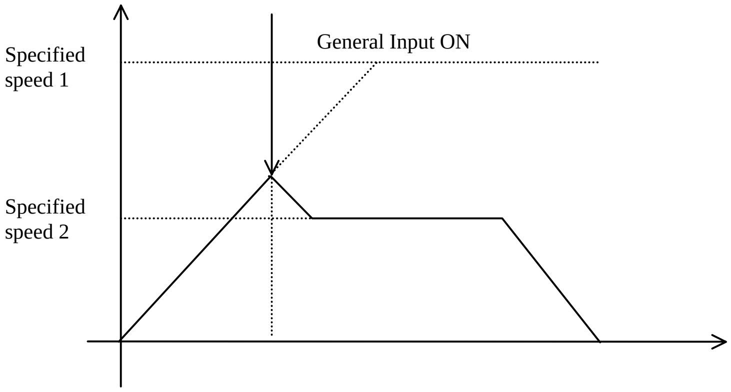


Speed Reference 2 Specified

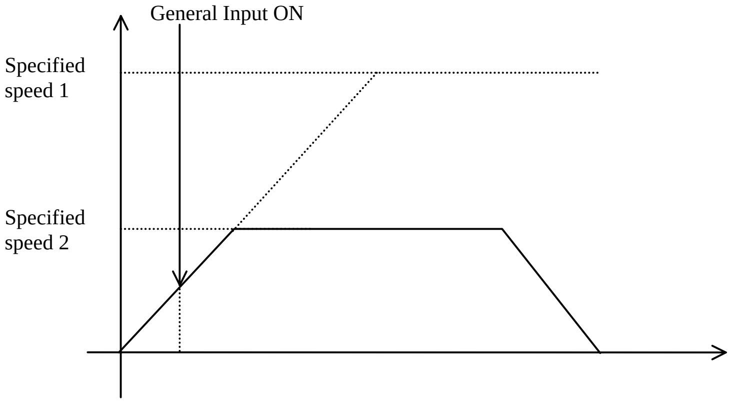
1,2,3... 1. A general input is turned ON while the axis is moving at constant speed, but the deceleration time is extremely short and the speed is not decelerated to what was specified with Speed Reference 2 because the speed of the axis is high and the movement of the axis is small.



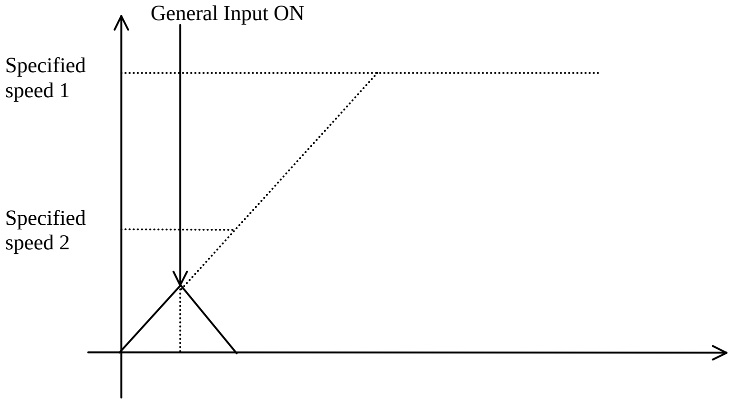
2. A general input is turned ON while the axis is being accelerated. (1)



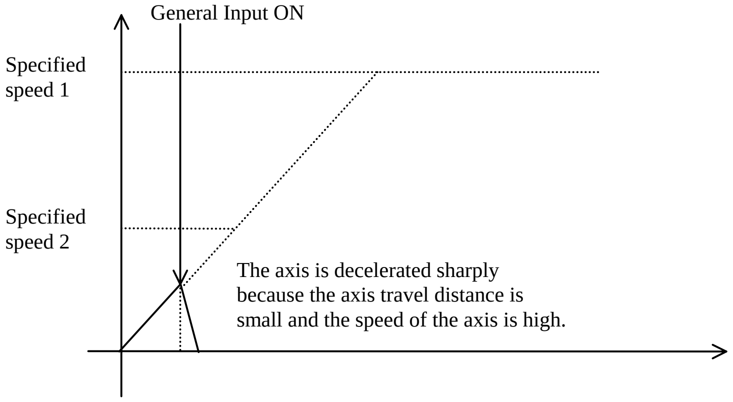
3. A general input is turned ON while the axis is being accelerated. (2)



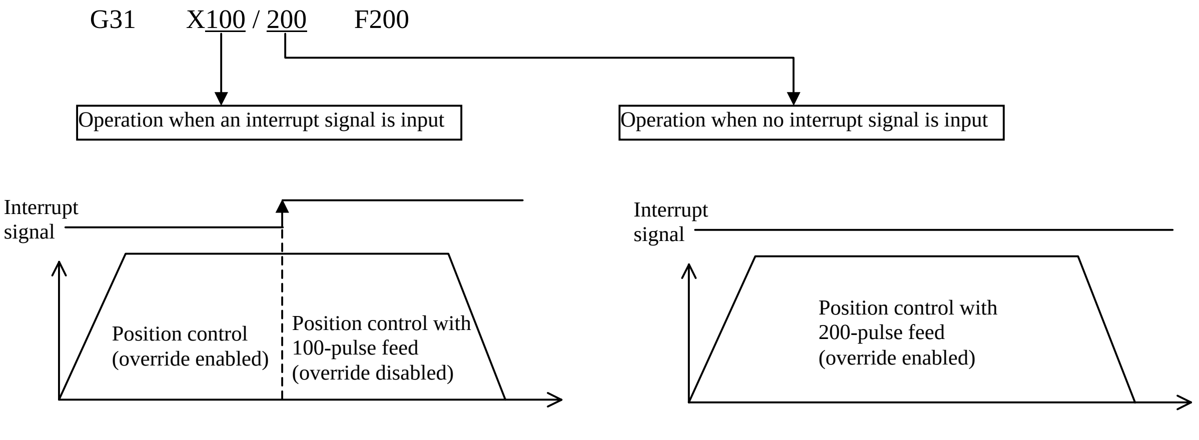
4. A general input is turned ON while the axis is being accelerated. (3)



5. A general input is turned ON when the axis is being accelerated. (4)



Example 3: Specifying the Travel Distance for When No Interrupt is Input
G31 X100/200 F200

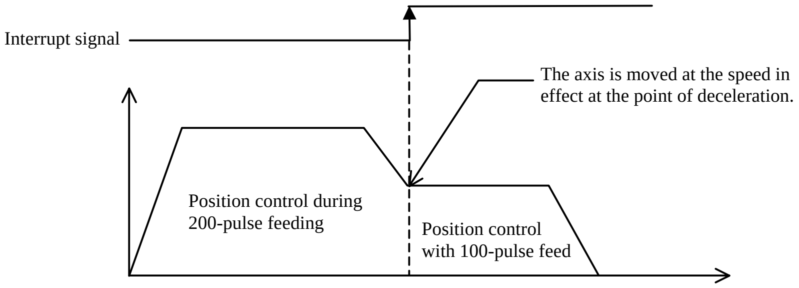

G31 X100 / 200 F200
Operation when an interrupt signal is input
Operation when no interrupt signal is input
Interrupt signal
Position control (override enabled)
Position control with 100-pulse feed (override disabled)
Interrupt signal
Position control with 200-pulse feed (override enabled)

If no interrupt signal is input after positioning is completed, an error will occur and operation will stop (program execution interrupt). If an S option is attached, however, the condition is not judged to be an error and the next block can still be executed.

Example G31 X 100/200 F200 S

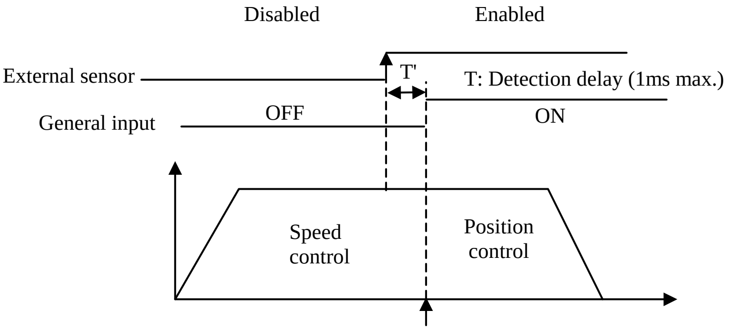
An interrupt input signal can be received at any time. The axis will be fed the reference amount at the going speed if an interrupt is input during deceleration.

Operation when an interrupt signal is input during deceleration


Interrupt signal
The axis is moved at the speed in effect at the point of deceleration.
Position control during 200-pulse feeding
Position control with 100-pulse feed

Note When positioning using interrupt feeding, positions that are determined after external sensor are enabled will vary depending on various conditions, such as the ambient temperature. This variation occurs due to detection delays by external sensors and the general input circuitry of the MC Unit.

The following illustration shows the MC Unit detection delay, and the resulting variation, assuming that there is no detection delay due to external sensors.



Timing for starting actual positioning by the MC Unit.

- The external sensor changes from disabled to enabled.
- The status of this external sensor is taken as the general input. There is a maximum delay of 1 ms in this general input circuit, so the MC Unit starts positioning control at no more than 1 ms after the external sensor is enabled.
- Accordingly, the actual position to be determined by the MC Unit after the external sensor is enabled can be obtained according to the following formula.

  Actual travel distance [pulse] =
  Specified travel distance [pulse] + Feed rate [pps] x Detection delay [s]

  - Actual travel distance refers to the distance from the point where the external sensor is enabled to the point where the machine actually stops moving.
  - Specified travel distance refers to the distance specified by G31.
  - Feed rate refers to the rate specified by G31.
  - The detection delay is 0.001 (s) max.

15 G32: Traverse

This command is provided for traverse, winding machine operations and it can be used in either Pass Mode or In-position Check OFF Mode.

Format G32_<Axis movement command>_<Speed reference>
[_M<M code>[/Stopover (Note 1)]]
[_D<D code>[/Stopover (Note 1)]]
[_O<Trailing end specification>](Note 2)
_L<Number of layers>

Note 1. The stopover function can be used with either an M code or a D code, but not with both.
2. When winding at the end, specify the number of layers. Winding will not be performed at the end if the number of layers is not specified.

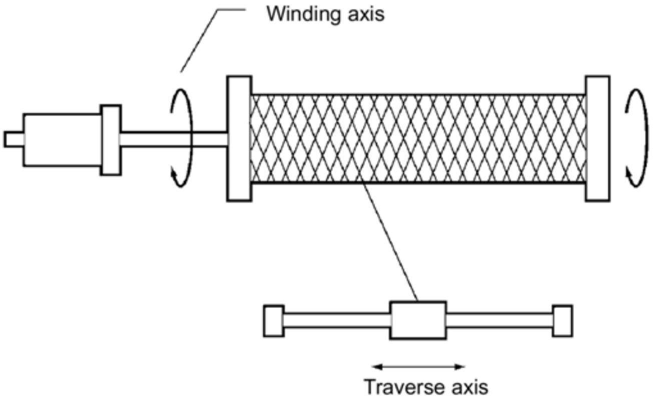
Operands The following table shows the possible settings for the operands.

Operand		Possible settings	
Rotating axis		X	Z
Coordinate data		±39999999 (E00) to (E31) A0 to A1999	±39999999 (E00) to (E31) A0 to A1999
Traverse axis		Y	U
Coordinate data		±39999999 (E00) to (E31) A0000 to A1999	
Speed reference		0.0001 to 39999999 (E00) to (E31) A0000 to A1999	
M code		000(0) to 999 (E00) to (E31) A0000 to A1999	
D code		000(0) to 255 (E00) to (E31) A0000 to A1999	
Stopover	Axes	X, Y, Z, U	
	Coordinate data	0 to 39999999 (E00) to (E31) A0000 to A1999	
End specification		0 to 39999999 (E00) to (E31) A0000 to A1999	
Number of layers		1 to 111111 (E00) to (E31) A0 to A1999	

Note Refer to 7-4 M Code Outputs for details on M codes, and to 7-4-9 D Code Outputs for details on D codes. Refer to 7-4-10 Stopover Function for details on the stopover function.

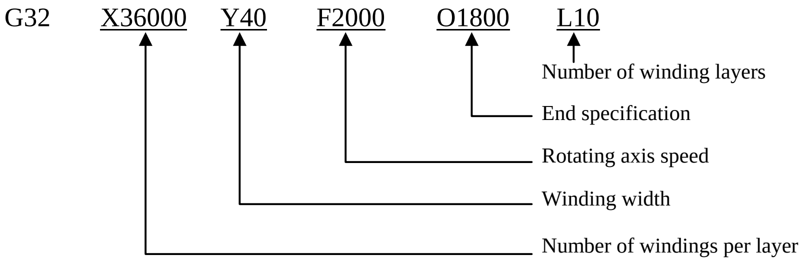
Values for all commands are relative to the present value.

Operation In the following application, the rotating axis of the spool is the X axis and the traverse axis is the Y axis.



G Command Example

The M and D codes are omitted here. Specify the command units in the parameters using [deg.] for the X axis and [mm] for the Y axis.



Number of Windings per Layer

Setting range: –39,999,999 to +39,999,999

Sets the number of windings per layer in [deg.] units. For 100 revolutions, set 36,000 because 360 [deg.] x 100 = 36,000 [deg.].

The windings run clockwise if a positive number is set and counterclockwise if a negative number is set. The minimum setting unit is 0.0001 [deg.].

Only the X and Z axes can be specified as the rotating axis and the rotating axis must be set to Unlimited Feed Mode. Refer to 6-20 Unlimited Feeding for details on this function.

Winding Width

Specifies the traverse width in [mm] units.

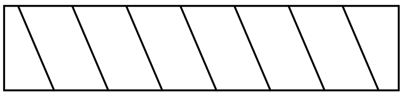
Axis Rotation Speed

Specifies X- or Z-axis rotation speed in [deg./s] units.

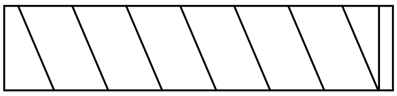
End Specification

Specifies the number of windings at the trailing end in [deg.] units. To set five windings at the end for example, then specify “01800” because 360 (deg.) x 5 =1,800 [deg.]. No winding at the end will be set if the O option is omitted.

No windings at the end (O option omitted)



Windings at the end (for only one layer)



Number of Winding Layers

Specifies the number of winding layers.

Restrictions

- Number of X-axis or Z-axis Rotations

The maximum distance that the X or Z axis can travel in one revolution without stopping must fall within the following equation:

｜Number of windings per layer x Total no. of windings + Number of end windings｜ ≤3 39,999,999 (deg.)

Numbers in the preceding range as well as the feed rate (in pulse units) after the pulse rate is changed must not exceed |7FFFFFFF|.

- Y-axis or U-axis Travel Distance

The maximum distance that the Y axis or the U axis can travel in one revolution without stopping must fall within the following equation:

|Traverse width | 3 39,999,999 (deg.)

Numbers in the preceding range as well as the feed rate (in pulse units) after the pulse rate is changed must not exceed 7FFFFFFF.

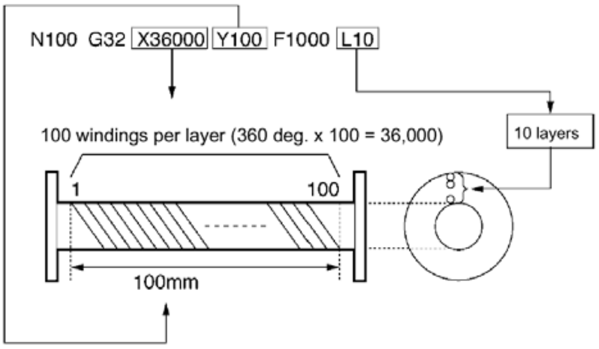
- Number of Winding Layers

The maximum number of windings is 111111 (39,999,999 [deg.]/360 [deg.]).

Programming Examples

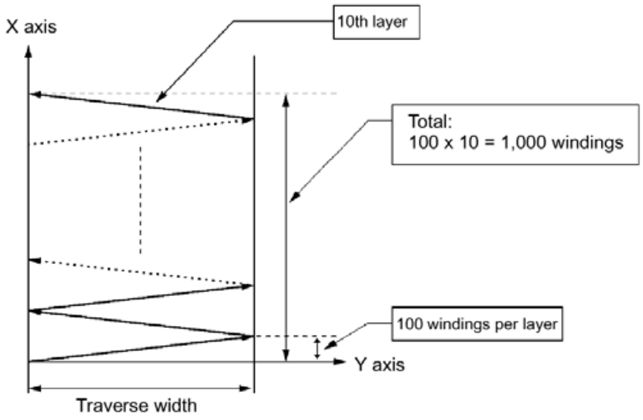
Example 1

In this example, a traverse operation is executed using aligned windings, a traverse width of 100 mm, 100 windings per layer and a total of 10 layers.



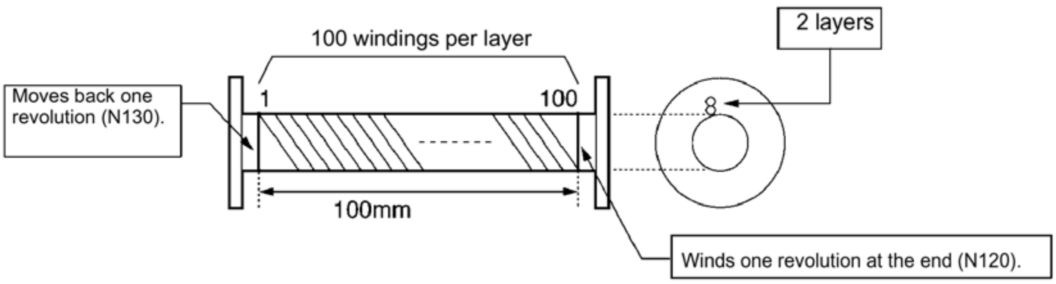
```
N100 G32 X36000 Y100 F1000 L10
```

Movement of each axis

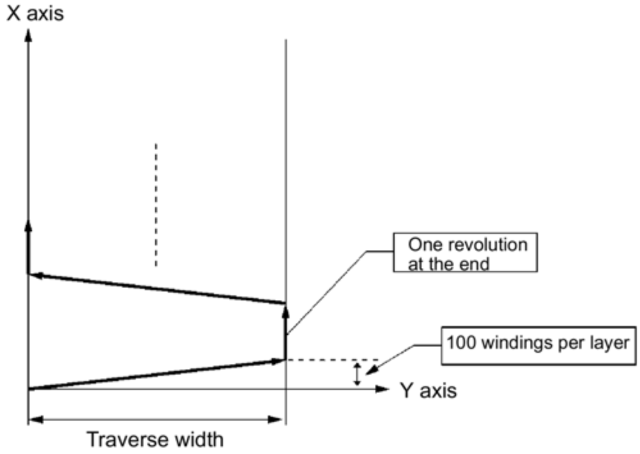


Example 2 In this example, just once winding is executed each time at the end. This is an example of Pass Mode operation combined with G32.

```
N110 G10 ....................................................................................... Specifies Pass Mode
N120 G32       X36000      Y100      F1000      O360      L1 ........... Winds one revolution at the end (0 designation)
N130 G32       X36000      Y-100     F1000      O360      L1 ........... Moves back to the start
```


100 windings per layer
2 layers
Moves back one revolution (N130).
1
100
8
100mm
Winds one revolution at the end (N120).

Movement of each axis



16 G50: Select Reference Coordinate System

Specifies the reference coordinate system as the coordinate system.

Format G50

Operands None

Description This command specifies the reference coordinate system as the coordinate system. The coordinate data for subsequent axis operations will be processed as reference coordinate data.

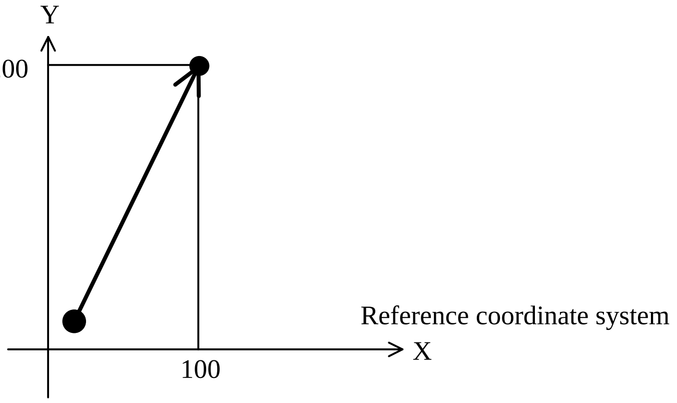
The default setting at the start of program execution will be the reference coordinate system.

After WORKPIECE ORIGIN RETURN (G27) is executed, the workpiece coordinate system will be used.

Programming Example

The following example shows setting the reference coordinate system as the coordinate system and performing linear interpolation.

```
        :       :
N009    G90
N010    G50
N011    G01     X100    Y200
        :       :
```



17 G51: Select Workpiece Coordinate System

Set the workpiece coordinate system as the coordinate system.

Format G51

Operands None

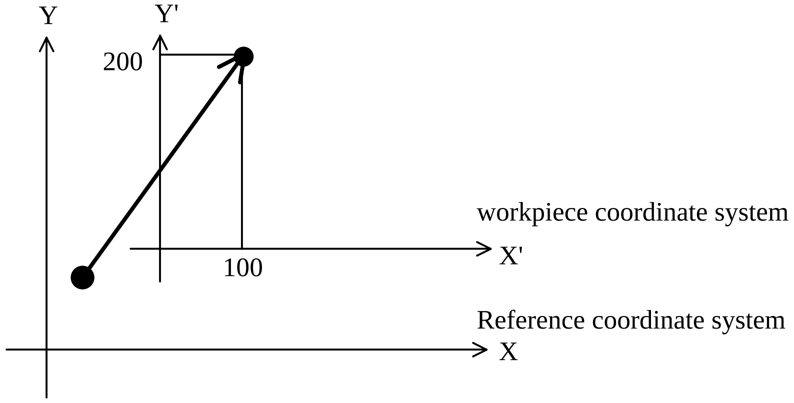
Description This command sets the workpiece coordinate system as the coordinate system. After this command is executed, the coordinate data in all subsequent axis operations is processed as workpiece coordinate data.

The origin for the workpiece coordinate system can be set with the system parameters or with G53. The reference coordinate system will be used after REF-ERENCE ORIGIN RETURN (G26)) or ORIGIN SEARCH (G28) is executed.

Programming Example

The following example shows setting the workpiece coordinate system as the coordinate system and performing linear interpolation.

```
        :       :
N009    G90
N010    G51
N011    G01     X100    Y200
        :       :
```



18 G53: Change Workpiece Origin Offset

Changes the origin of the workpiece coordinate system.

Format G53_<offset value…>

Operands The following table shows the possible settings for the operands.

Operand		Possible settings
Offset value	Axis names	X,Y,Z,U
	Data	±39,999,999 (E00) to (E31) A0000 to A1999

Description This command changes the offset between the reference coordinate system origin and the workpiece coordinate system origin to a specified value. The workpiece coordinate system will not be selected by executing this command. Specify using the workpiece coordinate system with G51.

The default setting for the workpiece coordinate system origin is the workpiece origin offset set in the system parameters when the main program is executed from the beginning.

The workpiece origin offset that is set in system parameters will not be updated by executing this command.

A software limit exceeded error will occur if a specified value exceeds the software limits set in the system parameters.

Programming Example 1

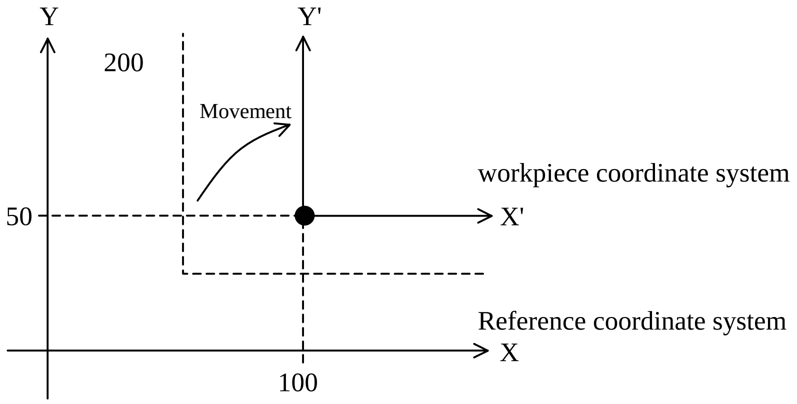
The following example shows changing the origin of the workpiece coordinate system.

```
        :       :
N020    G10                     Sets Pass Mode.
N021    G01     X100    Y100    F100
N022    G01     X200
N023    G53     X100
N024    G01     X300
        :       :
```

Programming Example 2

The following example shows changing the origin of the workpiece coordinate system.

```
        :       :
N010    G53     X100    Y50
        :       :
```



19 G54: Change Reference Coordinate System PV

Changes the present value in the reference coordinate system.

Format G54_<present value…>

Operands The following table shows the possible settings for the operands.

Operand		Possible settings
Present value	Axis name	X,Y,Z,U
	Data	±39,999,999 (E00) to (E31) A0000 to A1999

Description This command changes the present value of the specified axis reference coordinate system to a specified value.

⚠**Caution** The absolute value stored in the absolute encoder will not be changed by this command. Unlock the servo, lock it again and read values to use the present position in the absolute encoder.

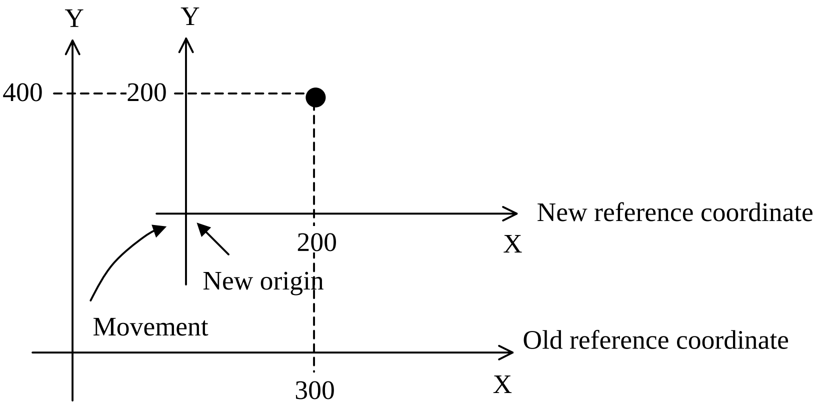
A software limit exceeded error will occur if a specified value exceeds the software limits set in the system parameters.

⚠**Caution** If this command is executed just after a G01, G02, G03, or G32 command, the interpolation will be performed in Stop Mode, not Pass Mode, even if Pass Mode has been selected. The following example shows this effect.

Programming Example

In this programming example, the present value of the reference coordinate sys-tem is changed from (300, 400) to (200, 200).

```
        :       :
N010    G54     X200    Y200
        :       :
```



20 G60: Arithmetic Operations

Performs arithmetic operations on position data and registers.

Format G60_<first term = second term operator third term>

Operands The following table shows the possible settings for the operands.

Operand	Possible settings
First term	A0000 to A1999 E00 to E31
Second term	A0000 to A1999 E00 to E31 ±39,999,999
Operator	+, –, *, or / (addition, subtraction, multiplication, or division)
Third term	A0000 to A1999 E00 to E31 ±39,999,999

Description This command performs arithmetic operations on numerical values, position data, or the contents of registers. When the first term is a register, the second and third terms will be integers. (Non-integer values are rounded to the nearest integer.)

Example

If G60: E00 = 0.4 + 0.4 is executed, then zero will be substituted at E00.

When the first term is a position data address, the second and third terms will be real numbers and values below the fifth decimal place will be rounded off.

Example

If G60: A0000 = 1.2345 x 1.01 is executed, then zero will be substituted at A0000.

The possible values for the result are as follows:

–39,999,999 to –0.0001
0
0.0001 to 39,999,999

An error will occur if the first term's data is not within the acceptable range. The possible values for the first terms are as follows:

When the first term is position data:	–39,999,999 to +39,999,999
When the first term is a register:	0000 to 1999

Registers are cleared to zero if the power is turned ON or the system is restarted.

Substitute values when registers are used.

Programming Example

The following example shows subtracting 500 from the value at A1000 and substituting the result at A0000.

```
        :      :
N010    G60    A0000=A1000–500
        :      :
```

21 G63: Substitution

Substitutes position data and registers

Format G63_<first term = second term>

Operands The following table shows the possible settings for the operands.

Operand	Possible settings
First term	A0000 to A1999 E00 to E31
Second term	A0000 to A1999 E00 to E31 ±39,999,999 X, Y, Z, U

Description This command copies position data, register contents, present values, or numerical values into position data addresses or registers.

When the second term is an axis name, the present position of that axis in the reference coordinate system is copied to the first term. That present position is copied according to the pulse rate and minimum unit setting for that axis specified in the system parameters. However, if the origin has not been fixed, the ORI-GIN UNDEFINED error will occur.

When the first term is a register, the value in the second term will be treated as an integer. Non-integer values are rounded to the nearest integer.

An error will occur if the first term's data is not within the acceptable range. The possible values for the first term are as follows:

When the first term is position data:	–39,999,999 to +39,999,999
When the first term is a register:	0000 to 1999

Registers are cleared to zero only when the power is turned on, so be sure to initialize the register contents when the program is started.

Programming Example

The following example shows **substituting 123.45 for A1000**.

```
        :       :
N010    G63     A1000=123.45
        :       :
```

22 G69: Change Parameter

Changes the setting of the specified parameter.

Format G69_<#parameter type>/<new setting …>

Operands The following table shows he possible settings for the operands.

Operand		Possible settings
Parameter type		1 to 8
New setting	Axis names	X, Y, Z, U
	New setting	Refer to the ranges below for each parameter. (E00) to (E31) A0000 to A1999

Description This command changes the setting of the specified parameter. The following table shows the parameters that can be changed and the parameter type values used to identify them. Parameters #1 to #4 are feed rate parameters, and #5 to #8 are servo parameters.

Parameter type	Parameter	Setting range
#1	Acceleration time	0 to 100,000 (ms)
#2	Deceleration time	0 to 100,000 (ms)
#3	Interpolation acceleration time	0 to 100,000 (ms)
#4	Interpolation deceleration time	0 to 100,000 (ms)
#5	Position loop gain	1 to 250 (rad/s)
#6	Position loop feed forward gain	0 to 100 (%)
#7	In-position	0 to 10,000 (pulses)
#8	Error counter warning	0 to 65,000 (pulses)

The actual settings in the system parameters are not changed by G69.

System parameters will be initialized when the main program is executed starting from the beginning. If a position data address or register is specified and the specified data is not an integer, then the value will be rounded off to the nearest integer. A number range over error will occur if the specified data is not within the allowable range.

The following is supplemental information for each of the parameters.

<Acceleration time, deceleration time, interpolation acceleration time, interpolation deceleration time>

When the operating mode is Stop Mode or In-position Check OFF Mode, the new acceleration/deceleration time will be reflected in the next operation. When the operating mode is Pass Mode and the interpolation acceleration/deceleration time is changed during linear or circular interpolation, the new interpolation acceleration/deceleration time will go into effect between G01, G02, G03, and G32 commands following the G69 command.

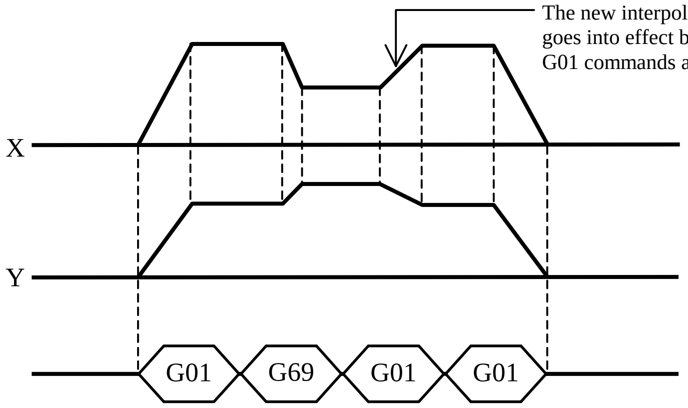
<Position loop gain, position loop feed forward gain, in-position, error counter warning>

In Stop Mode or In-position Check OFF Mode, these values will go into effect when parameters are changed. In Pass Mode, they will go into effect with the next G69 operation command.

Programming Example

In this programming example, the interpolation acceleration time is changed during interpolation.

```
        :       :
N009    G01     X200    Y300
N010    G69     #3/X100
N011    G01     X300    Y500
N012    G01     X500    Y800
        :       :
```



The new interpolation acceleration time goes into effect between the next two G01 commands after the G69 command.

23 G70: Unconditional Jump

Unconditionally jumps to the specified block.

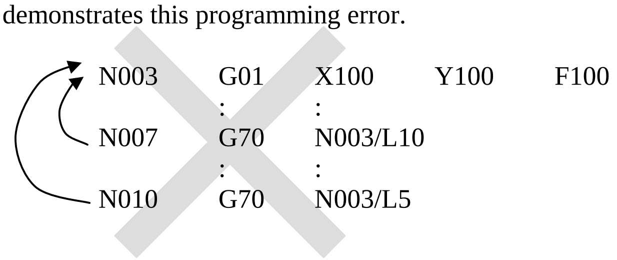
Format G70_<jump destination block number>[/L<number of loops>]

Operands The following table shows the possible settings for the operands.

Operand	Possible settings
Destination block number	N000 to N999
Number of loops	1 to 39,999,999 (E00) to (E31) A0000 to A1999

Description This command unconditionally jumps to the specified block. A fixed number of jumps (a loop) can be created by specifying a number of loops.
When a number of loops is specified, the destination block number must be before the current block number.
If an UNCONDITIONAL JUMP (G70) or a CONDITIONAL JUMP (G71) command is used to exit a loop before the number of loops has counted down to zero and the jump is to a program location after the loop, the number of loops will be cleared when another loop is started.

A second loop cannot be executed while a loop is in progress. The following diagram demonstrates this programming error.



```
N003    G01    X100        Y100    F100
        :      :
N007    G70    N003/L10
        :      :
N010    G70    N003/L5
```

Loops cannot be nested as shown above, but up to six loops can be nested by calling subroutines containing loops. The number of loops in each subroutine is cleared when the subroutine ends. Refer to 7-3-25 G72: SUBROUTINE JUMP for details.
When the number of loops is set indirectly from position data with register specification or position data number specification, the number of loops will be 0 if the value of the position data is 0. Non-integer position data values will be rounded off to the nearest integer value. An error will also occur if the position data is negative.

Programming Example

The following example shows executing another loop when a given condition occurs in the first loop.

```
        :      :
N020    G00    X100
N030    G71    N090/A1000=1
N040    G70    N020/L100 . . . . . . . . . . ●
        :      :
        :      :
N090    G00    X3200
N100    G70    N090/L50  . . . . . . . . . . ●
        :      :
```

In loop ● , the program will jump to N020 up to 100 times while A1000p1, so blocks N020 and N030 will be executed up to 101 times.

In loop ● , the program will jump to N090 up to 50 times, so block N090 will be executed up to 51 times.

If A1000=1 on the 20th execution in loop ● , the program would jump to block N100. The remaining value of 80 jumps in the number of loops would be cleared and the number of loops would be set to the new value of 50 for block N100.

24 G71: Conditional Jump

Jumps to the specified block when the condition is met.

Format G71_<jump destination block number>/<condition equation>

Operands The following table shows the possible settings for the operands.

Operand		Possible settings
Destination block number		N000 to N999
Condition equation	First term	A0000 to A1999 E00 to E31
	Operator	=, <, >, or !
	Second term	A0000 to A1999 E00 to E31 ±39,999,999

Description This command jumps to the specified block when the given condition is met. The “!” operator is the inequality operator (not equal to). The first and second terms will be compared as real numbers, even if registers are specified.

Programming Example

The following example shows repeating until A1000=10

```
        :       :
N008    G91
N009    G63     A1000=0
N010    G00     X500
N011    G60     A1000=A1000+1
N012    G71     N010/A1000 ! 10
        :       :
```

In this programming example, 0 is substituted for A1000 in block N009 and then incremented by 1 in block N011 after the positioning operation in block N010. At N012, the program jumps back to N010 and repeats blocks N010 to N012 while A1000 ≠ 10. The program proceeds when A1000=10.

25 G72: Subroutine Jump

Calls the specified subroutine.

Format G72_<subroutine number>

Operands The following table shows the possible settings for the operands.

Operand	Possible settings
Subroutine number	P500 to P999
	A0000 to A1999

Description This command calls and executes the specified subroutine. Subroutines can be nested five times. An error will occur if a sixth subroutine is called.
If the position data number is specified for the subroutine number, the subroutine stored in the position data will be called and executed (indirect calling).
A no program error will result if the position data does not satisfy the following condition.

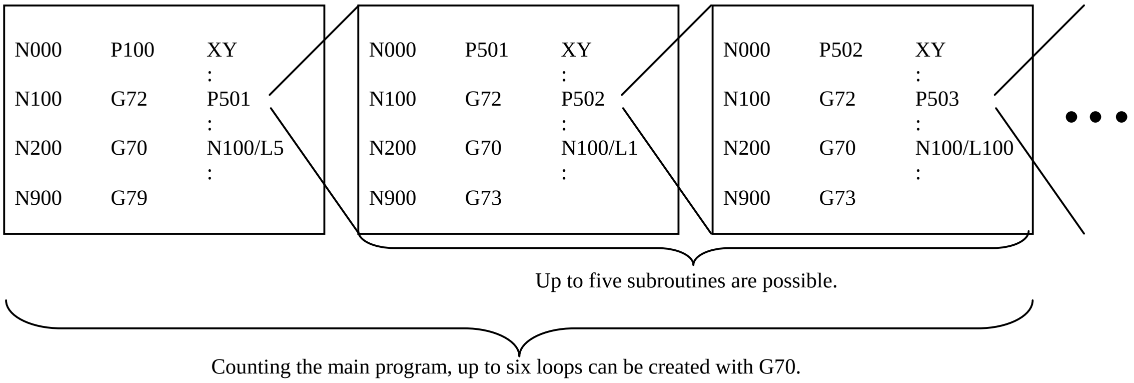
500 ≦ Position data ≦ 999

Programming Example

The following example shows calling a subroutine.

```
        :       :
N010    G72     P500
        :       :
```

Up to six loops can be created by calling subroutines, as shown in the following diagram.



```
N000    P100    XY
                :
N100    G72     P501
                :
N200    G70     N100/L5
                :
N900    G79
```

```
N000    P501    XY
                :
N100    G72     P502
                :
N200    G70     N100/L1
                :
N900    G73
```

```
N000    P502    XY
                :
N100    G72     P503
                :
N200    G70     N100/L100
                :
N900    G73
```

• • •
Up to five subroutines are possible.
Counting the main program, up to six loops can be created with G70.

26 G73: Subroutine End

Ends the subroutine.

Format G73

Operands None

Description This command ends the subroutine and returns control to the block after the one that called the subroutine. This command must be programmed at the end of every subroutine.

Programming Example

```
N000   P510   XY
       :      :
       :      :
       :      :
N100   G73
```

27 G74: Optional End

Ends the block currently being executed when the specified optional input goes ON.

Format G74● <optional number>

Operands The following table shows the possible settings for the operands.

Operand	Possible settings
Optional number	00 to 19

Description If the specified input is received while the block after the one with this command is being executed, execution of that block will be interrupted and the program will proceed to execute the following block.

If an axis operation is being executed, the operation will be decelerated to a stop before proceeding to the next block. If a DWELL TIMER (G04) command is being executed, the command will be interrupted and the remaining time can-celled.

The source of the optional input depends on the optional number specified, as shown below.

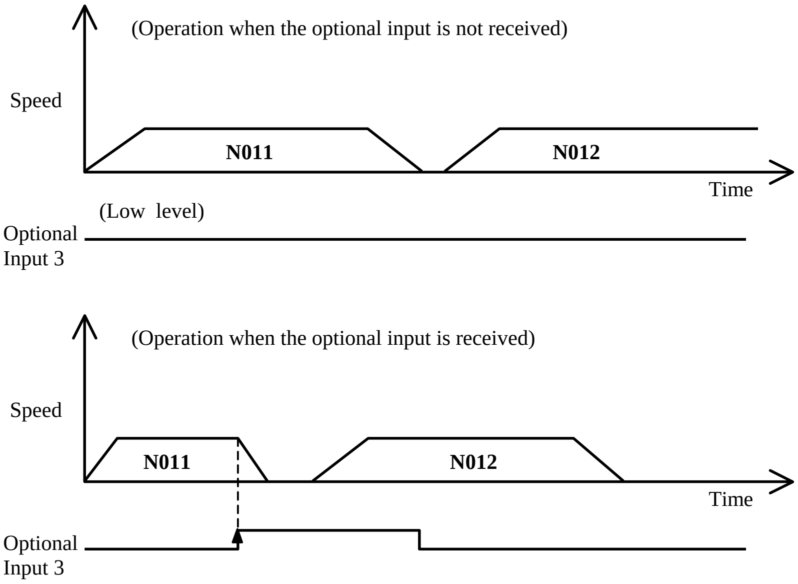
0 through 15:	Inputs from the PC Interface Area
16 through 19:	Inputs from general inputs 1 to 4

This command is ineffective if the following block contains a SUBROUTINE END (G73) or PROGRAM END (G79) command. If the specified optional input is ON already, this command will operate just like the OPTIONAL SKIP (G75) command.

Programming Example

The following example shows stopping linear interpolation in progress by turning ON an optional input.

```
        :       :
N008    G11
N009    G91
N010    G74     3
N011    G01     X100
N012    G01     X100
        :       :
```


(Operation when the optional input is not received)
Speed
N011
N012
Time
(Low level)
Optional Input 3
(Operation when the optional input is received)
Speed
N011
N012
Time
Optional Input 3

28 G75: Optional Skip

Skips the next block when the specified optional input is ON.

Format G75● <optional number>

Operands The following tables shows the possible settings for the operands.

Operand	Possible settings
Optional number	00 to 19

Description If the specified input is ON when this command is executed, the next block will be skipped. The source of the optional input depends on the optional number specified, as shown below.

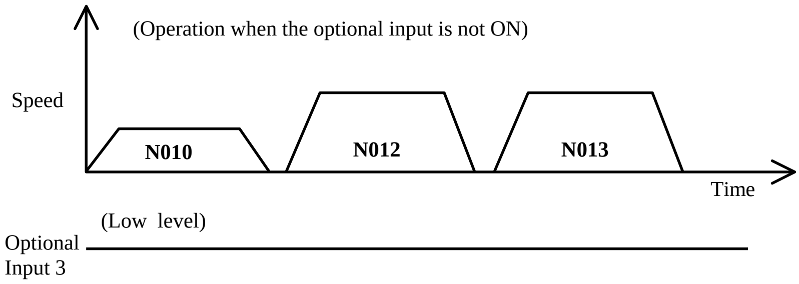
0 through 15:	Inputs from the PC Interface Area
16 through 19:	Inputs from general inputs 1 to 4

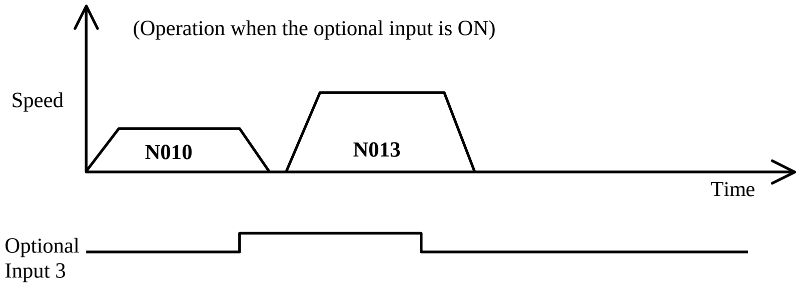
Program execution will continue normally even if the optional input comes ON while the next block is being executed. The OPTIONAL SKIP (G75) command will be disabled if the following block contains a SUBROUTINE END command (G73) or PROGRAM END command (G79).

Programming Example

The following example shows disabling a linear interpolation operation by turning ON an optional input.

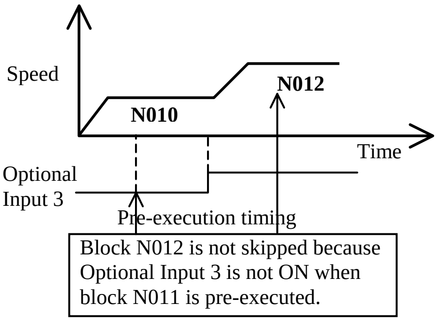
```
        :       :
N008    G11
N009    G91
N010    G01     X50
N011    G75     3
N012    G01     X50
N013    G01     X50
        :       :
```



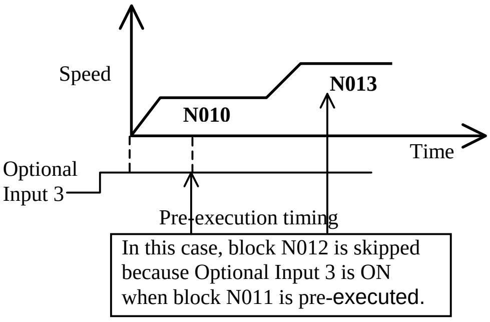


When block N008 contains the Pass Mode (G10) command, blocks N011 and N012 are pre-executed while N010 is being executed, so block N012 will not be skipped if Optional Input 3 is turned ON after execution of block N010.

To ensure that block N012 will be skipped, make sure that Optional Input 3 is ON before block N010 is executed.



Block N012 is not skipped because Optional Input 3 is not ON when block N011 is pre-executed.



In this case, block N012 is skipped because Optional Input 3 is ON when block N011 is pre-executed.

29 G76: Optional Program Pause

Pauses the program when the specified optional input is ON.

Format G76● <optional number>

Operands The following table shows the possible settings for the operands.

Operand	Possible settings
Optional number	00 to 19

Description If the specified input is ON when this command is executed, program execution will be paused. Program execution will continue from the next block when the optional input is reset. The source of the optional input depends on the optional number specified, as shown below.

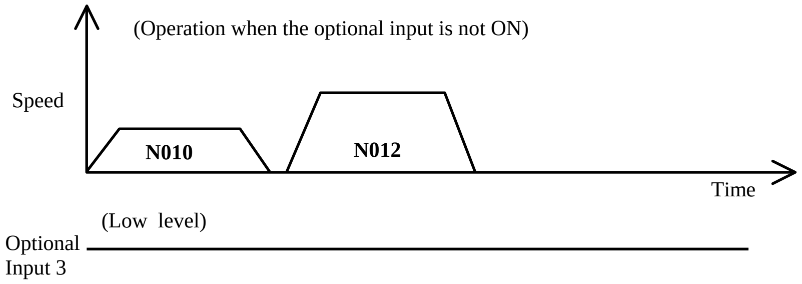
0 through 15:	Inputs from the PC Interface Area
16 through 19:	Inputs from general inputs 1 to 4

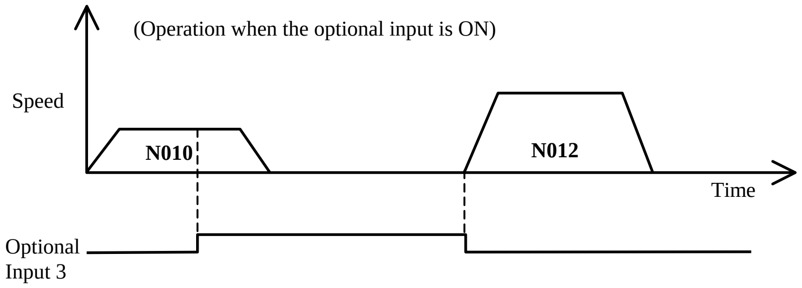
Program execution will continue normally if the optional input comes ON while the next block is being executed.

Programming Example

The following example shows pausing a linear interpolation operation by turning ON an optional input.

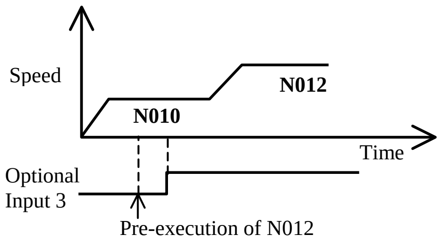
```
        :       :
N008    G11
N009    G91
N010    G01     X100
N011    G76     3
N012    G01     X100
        :       :
```



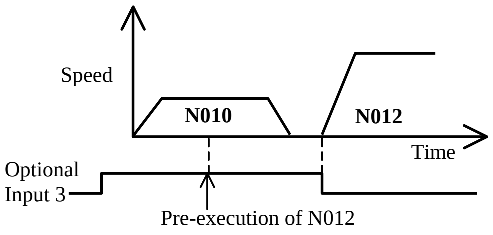


When block N008 contains the PASS MODE command (G10), blocks N011 and N012 are pre-executed while N010 is being executed, so the program will not be paused if Optional Input 3 is turned ON after execution of block N010.

To ensure that the program will be paused, make sure that Optional Input 3 is ON before block N010 is executed.



The program is not paused because Optional Input 3 is not ON when block N011 is pre-executed.



In this case, the program is paused because Optional Input 3 is ON when block N011 is pre-executed.

30 G79: Program End

Ends the main program.

Format G79

Operands None

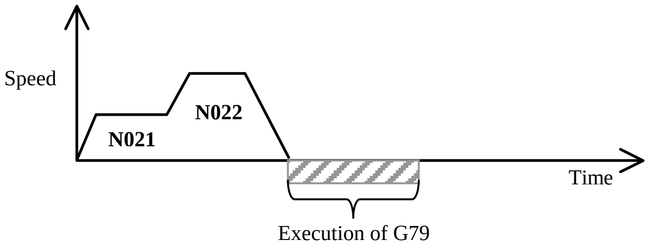
Description This command ends the main program and must be included at the end of the main program.

When G79 is executed and an axis is in operation, the Unit will wait for the axis to be positioned before executing G79. M codes M500 to M999 will be forcibly cleared if they are being output when G79 is executed.

Programming Example

The following example shows how G79 is executed after axis operation is completed in Pass Mode.

```
        :       :
N020    G10                     Sets Pass Mode.
N021    G01     X100    Y100    F100
N022    G01     X200    F200
N023    G79                     Executed after N022 is completed.
        :       :
```



31 G90: Absolute Specification

Specifies the use of the absolute coordinates in axis operations.

Format G90

Operands None

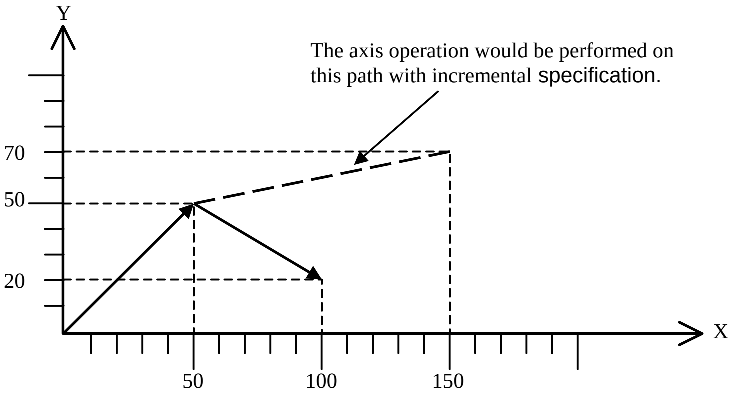
Description This command specifies that the absolute coordinate system for each axis is to be used when for axis positioning. After this command is executed, the coordinate data in axis movement commands is treated as absolute coordinate data.

In addition to G90, the absolute coordinate system is put into effect when a REFERENCE ORIGIN RETURN (G26), WORKPIECE ORIGIN RETURN (G27), or ORIGIN SEARCH (G28) command is executed.

Programming Example

The following example shows positioning with absolute coordinate data.

```
          :          :
N010      G90
N011      G01        X50       Y50
N012      G01        X100      Y20
          :          :
```



The axis operation would be performed on this path with incremental specification.

32 G91: Incremental Specification

Specifies the use of relative coordinates in axis operations.

Format G91

Operands None

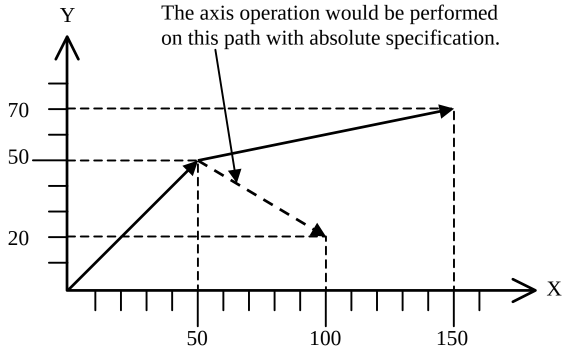
Description This command specifies that positioning axis operations are performed relative to the present position. After this command is executed, the coordinate data in axis movement commands is treated as the distance to be moved from the present position.

In addition to G91, the incremental coordinate system is put into effect when a REFERENCE ORIGIN RETURN (G26), WORKPIECE ORIGIN RETURN (G27), or ORIGIN SEARCH (G28) command is executed.

Programming Example

The following example shows positioning with incremental coordinate data.

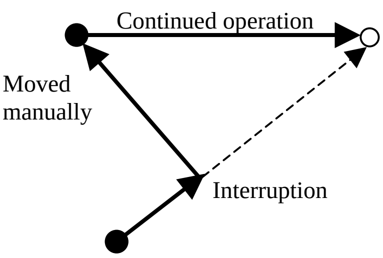
```
          :          :
N010      G91
N011      G01        X50       Y50       F100
N012      G01        X100      Y20
          :          :
```



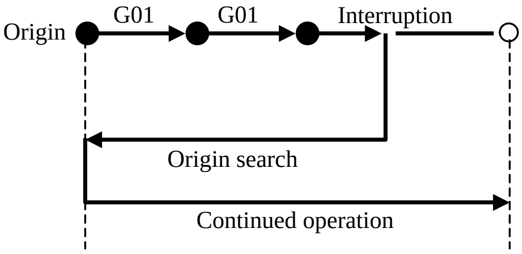
The axis operation would be performed on this path with absolute specification.

If positioning with incremental specification is interrupted by a pause command, the axes will be moved to the original end position after the operation is restarted. Also, if the axes have been moved or an origin search was performed after the operation was interrupted, the axes will still be moved to the original end position.

Example 1



Example 2



M Code outputs

1 Introduction

M codes consist of information used to interlock with external devices in positioning operation processes. External devices are devices directly connected to the general outputs of the CPU Unit and MC Unit.

Format M<M code>

Example 1: M500
Example 2: G01 X100 MA000
Example 3: G01 X200 Y100 M(E00)

Operands The following table shows the possible settings for the operands.

Operand	Possible settings
M code	00 to 999 (E00) to (E32) A0000 to A1999

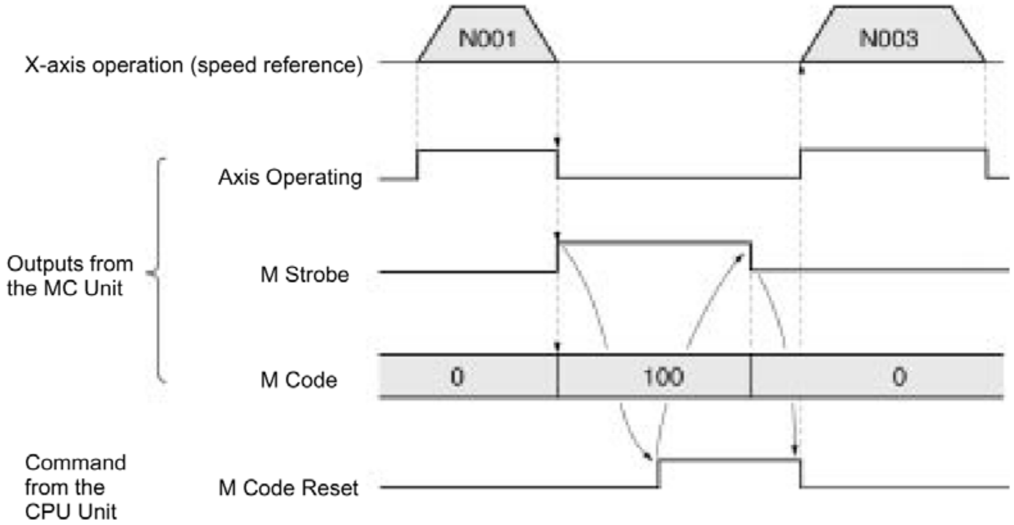
Programming Example

The following program shows an example of M code usage.

```
N000   P100   X ........................ Declares the program number.
N001   G01    X100   F100..... Moves to 100 mm on the X-axis by linear interpolation.
N002   M100 ................................ Outputs M code 100 to the CPU Unit and waits for OK to
                                             perform the next operation. Proceeds to next block when
                                             the M code reset is received from the CPU Unit.
N003   G01    X0 ...................... Moves to 0 mm on the X-axis by linear interpolation.
N004   G79.................................. Declares the end of the program
```

Timing Chart

The following diagram shows the timing of the programming example execution. The Axis Operating, M Strobe, M Code, and M Code Reset Bits are provided in the PC Interface Area.



Description The following list provides a running description of the program execution.

1,2,3...
1. In block N001, the X-axis is moved to 100 mm by linear interpolation.
2. In block N002, M code 100 is output after the X-axis positioning is completed; the M strobe is turned ON at the same time to indicate that the M code has been set.
3. In the CPU Unit, the processes corresponding to M code 100 are performed when the M strobe goes ON. The CPU Unit turns ON the M Code Reset Bit when these processes are completed.
4. In the MC Unit, the M code is cleared to 0 and the M strobe is turned OFF when the CPU Unit turns ON the M Code Reset Bit.
5. When the M Code Reset Bit from the CPU Unit goes OFF, block N003 is executed, moving the X-axis to 0 mm.
6. The program ends in block N004.

2 M Code Data

There are some M codes that interlock as explained on the previous page, and others that do not interlock. These differences are explained below.

When an M code is specified in a register or position data address, the specified value will be rounded to the nearest integer value. An error will occur if the result is not an acceptable M code value (0 to 999).

M Codes 0 to 499

M codes 0 to 499 are used to interlock with external devices. Execution of the program is paused when there is not an M code reset input.

M Codes 500 to 999

Use M codes 500 to 999 when it is not necessary to interlock. These M codes are just output and the program is executed without waiting for an M code reset input.

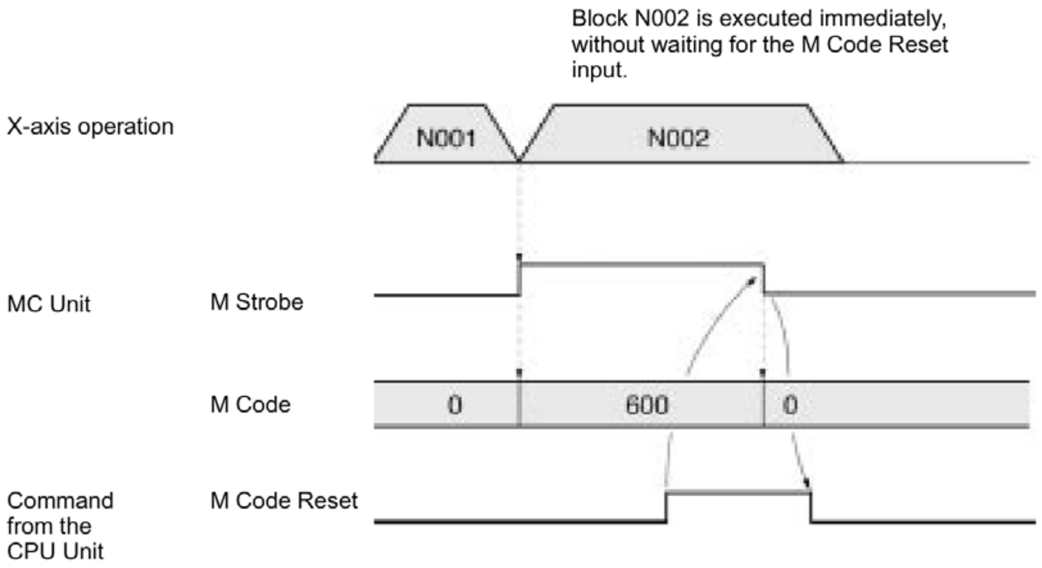
These M codes are cleared when the PROGRAM END command (G79) is executed.

3 M Code Examples

M codes can be used independently or with G codes, such as G00 and G01, that execute axis operations.

Example The following program just outputs M code 600. (M codes over 499 do not interlock.)

```
N000   P000   X
N001   G01    X100   F100   M600
N002   G01    X200   F100
N003   G79
```


Block N002 is executed immediately, without waiting for the M Code Reset input.
X-axis operation
N001
N002
MC Unit
M Strobe
M Code
0
600
0
Command from the CPU Unit
M Code Reset

4 Effect of Mode Changes on M Code Resets

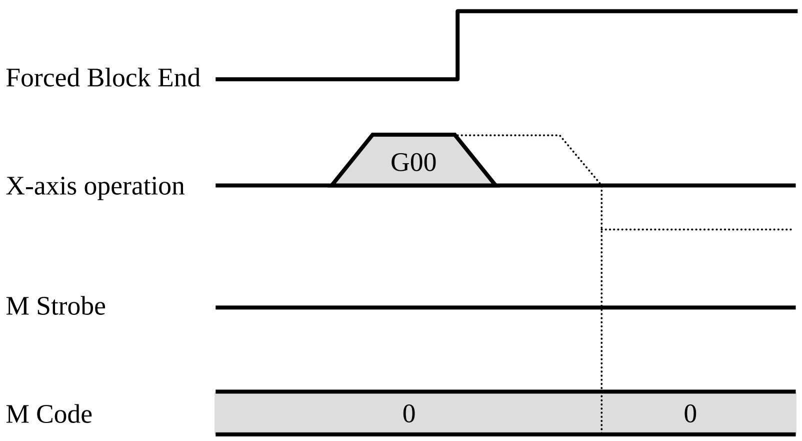
The status of M codes is not affected by switching from manual to automatic mode or vice versa.

5 Clearing M000 to M499 with Forced Block End or G74

The M strobe and M code outputs for M codes 000 to 499 will be cleared if the program block that outputs the M code is cleared with the Forced Block End Bit (PC Interface Area) or the OPTIONAL END (G74) command.

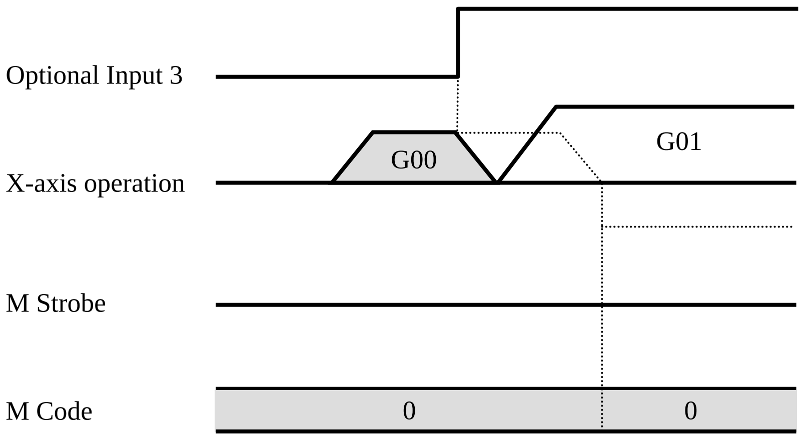
Example 1 The M strobe and M code will not be output if a Forced Block End signal is received before they are output.

G00 X500 M100



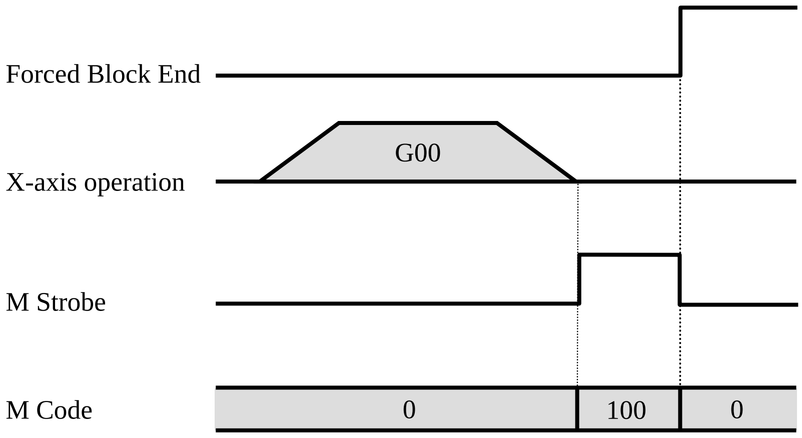
Example 2 The M strobe and M code will not be output if the OPTIONAL END command’s optional input is received before they are output.

G74 3
G00 X500 M100
G01 X1000



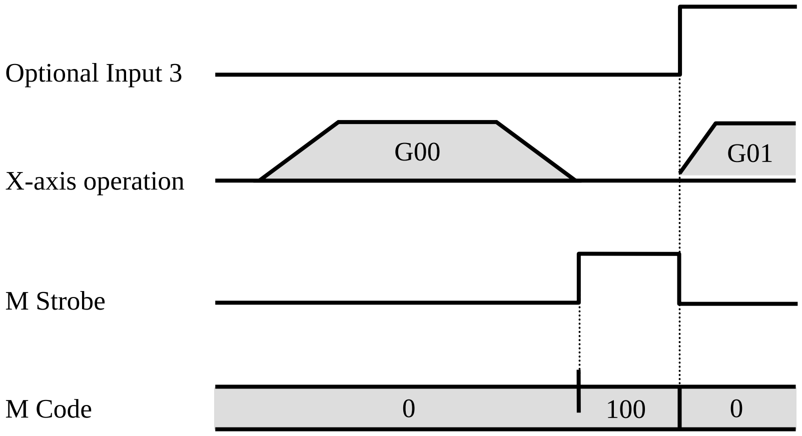
Example 3 When the output has been completed and the system is set to wait for the M code reset, the M strobe will be turned OFF and the M code will be cleared to release the M code reset.

G00 X500 M100



Example 4 The M strobe and M code will be cleared if the OPTIONAL END command's optional input is received while they are being output.

```
G74     3
G00     X500    M100
G01     X1000
```

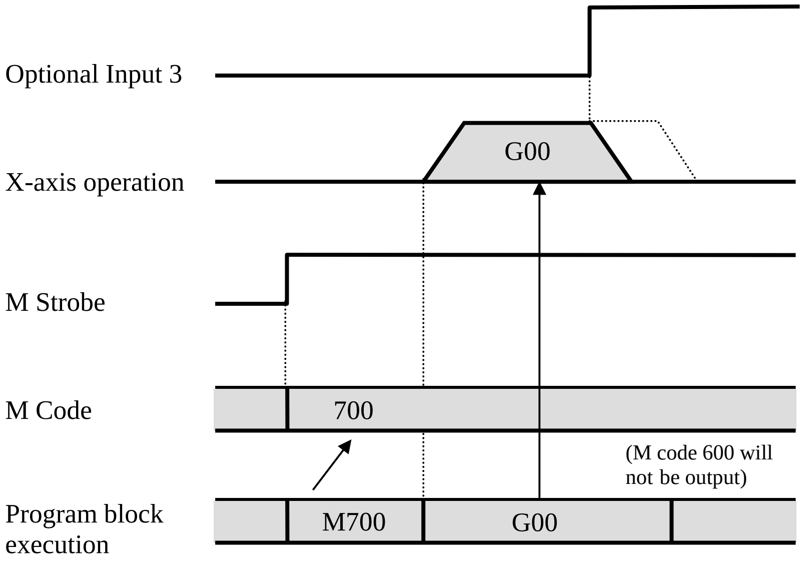


6 Clearing M500 to M999 with Forced Block End or G74

The M strobe and M code outputs for M codes 500 to 999 will not be output cleared if the program block that outputs the M code is cleared with the Forced Block End Bit (PC Interface Area) or the OPTIONAL END (G74) command, but previous M codes will not be cleared.

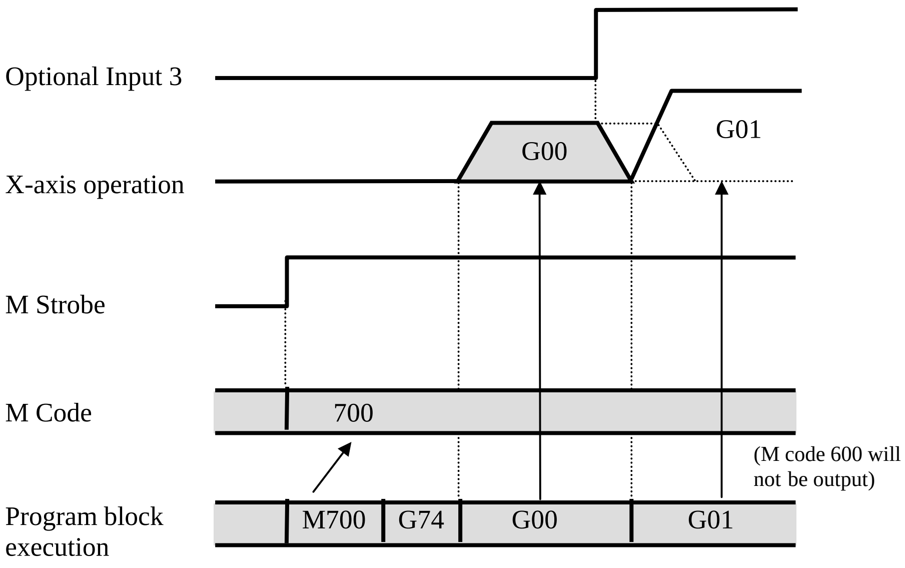
Example 5 The M strobe and M code will not be output if a Forced Block End signal is received before they are output, but a previous M code and M strobe will not be cleared.

```
M700
G00     X500    M600
```



Example 6 The M strobe and M code will not be output if the Optional End command's optional input is received before they are output, but a previous M code and M strobe will not be cleared.

```
M700
G74     3
G00     X500    M600
G01     X1000
```



7 M Code Outputs in Pass Mode

M codes M500 to M999 can be output in Pass Mode while axes are being operated. In this case, M codes will be output at the following positions according to the system parameter settings.

Constant Acceleration Mode OFF

The M code will be output just before the demand position. The output position can be calculated using the following equation.

Demand position - reference speed x pass time / 2

The pass time is determined by the Pass Time Mode Selection settings in the system parameters as follows:

The pass time is the interpolation acceleration time when interpolation acceleration time is selected.

The pass time is the interpolation deceleration time when interpolation deceleration time is selected.

Constant Acceleration Mode ON

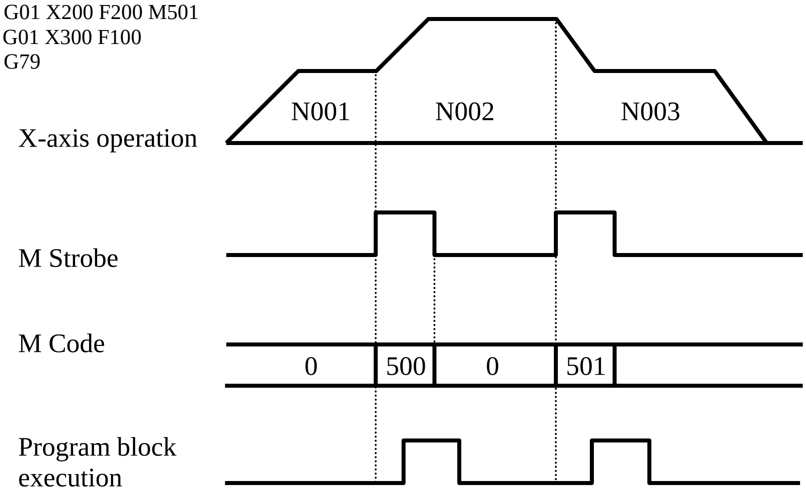
The M code will be output at the demand position.

Refer to 6-8 Operating Modes for details on the Constant Acceleration Mode.

Example 1 **Constant Acceleration Mode OFF**

The following program will output an M code during pass operation. The M code is reset from the CPU Unit.

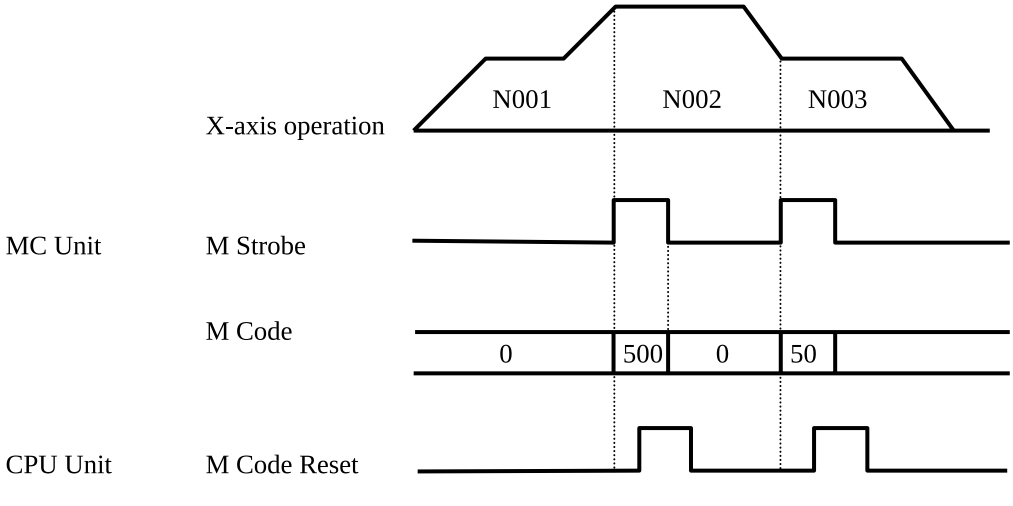
```
N000 P000 X
N001 G01 X100 F100 M500
N002 G01 X200 F200 M501
B003 G01 X300 F100
N004 G79
```



Example 2 **Constant Acceleration Mode ON**

The following program will output an M code during pass operation. The M code is reset from the CPU Unit.

```
N000 P000 X
N001 G01 X100 F100 M500
N002 G01 X200 F200 M501
B003 G01 X300 F100
N004 G79
```



8 Stopover function and Resetting M Codes

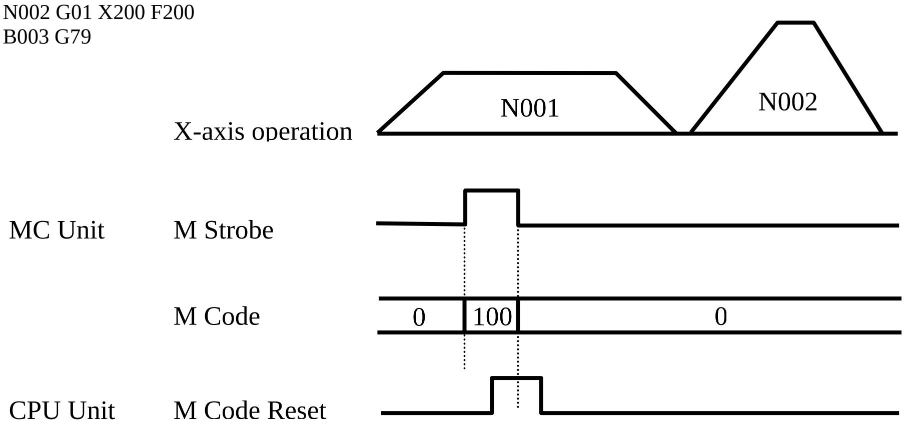
This section describes the reset timing for M codes when the Stopover function is used. Refer to 7-4-3 for details on the Stopover function.

M000 to M499

When an M code between M000 and M499 is output, the next positioning operation will be performed after positioning to the demand position has been completed in either Pass Mode or In-position Check OFF Mode.

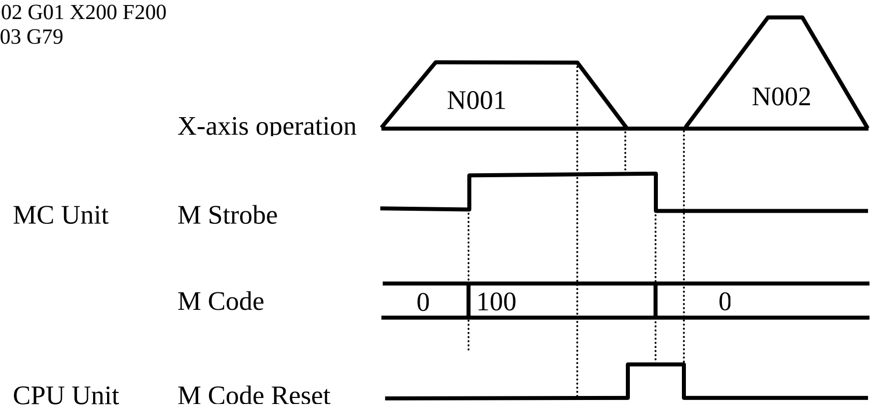
Example 1 The following program outputs M code 100 during linear interpolation (N001). The CPU Unit turns the M Code Reset Flag ON and OFF to reset the M code while N001 is executing. When that happens, the next positioning operation (in N002) is performed after the axis has moved 100 mm on the X axis.

```
N000 P000 X
N001 G01 X100 F100 M100/X30       Outputs M code 100 after the axis has moved 30 mm from the present position.
N002 G01 X200 F200
B003 G79
```



Example 2 The following program outputs M code 100 during linear interpolation (N001). The Unit will wait for the M Code Reset Bit to turn ON after positioning has been completed because the Bit was not input while N001 was executing. The CPU Unit turns the M Code Reset Bit ON and OFF to reset the M code. When that happens, the next positioning operation (N002) is performed.

```
N000 P000 X
N001 G01 X100 F100 M100/X30       Outputs M code 100 after the axis has moved 30 mm from the present position.
N002 G01 X200 F200
B003 G79
```



M500 to M999 The M code is reset with the same timing used to output M code in Pass Mode.

Note Refer to 7-4-7 M-code Outputs in Pass Mode for more details.

9 D Code Outputs

D Codes A D code is a preset code that is output after positioning has been completed. The D code (0 to 255) is output to the CPU Unit as an interrupt task number to executed the interrupt task in the CPU Unit. It can also be output during operation without stopping operation by using the Stopover Function. Refer to 6-10 or Stopover Function for details on the function.

⚠ **Caution** D code outputs can only be used when an MC Unit is mounted on the CPU Back-plane. They cannot be used if the Unit is mounted on an Expansion Backplane.

Format

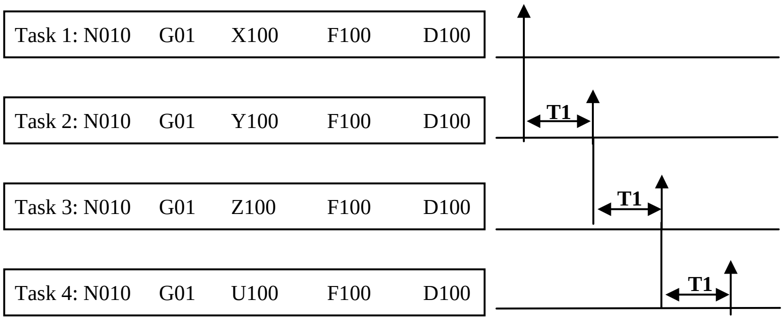
```
D <D code>
Case 1: N010    D100
Case 2: N010    G01    X100    F200    D100
Case 3: N010    G01    X100    F200    M50    D100
```

Operands

D Code
0 to 255
(E00) to (E32)
A0000 to A1999

Description

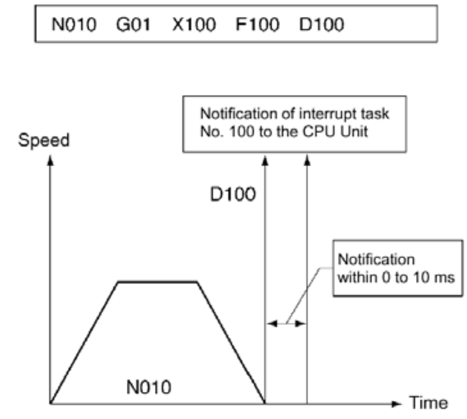
- The D code must be between 0 and 255.
- The following can be written after the D.
  - A number: 0 to 255 (integer)
  - (E00) to (E31)
  - A0000 to A1999
- The output timing for D codes is the same as that for M codes.
- D codes, like M codes, can be used with G codes. Only one or the other can be used at a time with a G code.
- When a D code is output, the CPU Unit determines whether it can receive an interrupt. If it cannot receive an interrupt, then it will monitor status for 10 ms. If reception is still not possible, the CPU Unit will output a D code timeout error and stop.
- If a D code is output from more than one task at the same time, then outputting the D code for each task will be offset in 2 to 10-ms increments.


Task 1: N010 G01 X100 F100 D100
Task 2: N010 G01 Y100 F100 D100
Task 3: N010 G01 Z100 F100 D100
Task 4: N010 G01 U100 F100 D100
T1
T1
T1

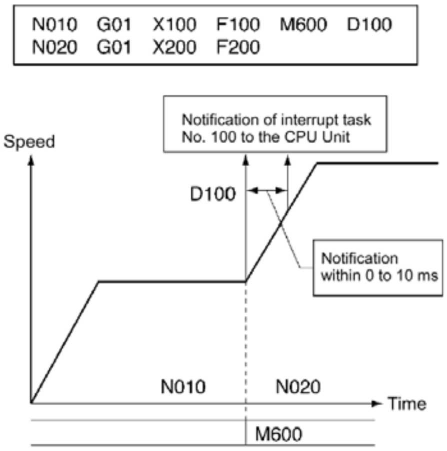
T1, T2, T3: 2 to 10 ms

Operating Patterns

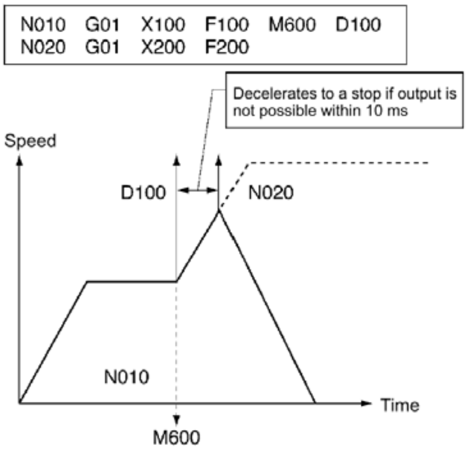
Case 1: Notification after Positioning Is Completed


N010 G01 X100 F100 D100
Notification of interrupt task No. 100 to the CPU Unit
Speed
D100
Notification within 0 to 10 ms
N010
Time

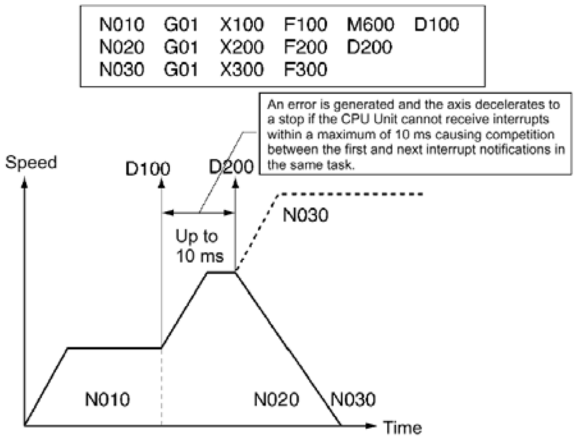
Case 2: Pass Mode Operation


N010 G01 X100 F100 M600 D100
N020 G01 X200 F200
Notification of interrupt task No. 100 to the CPU Unit
Speed
D100
Notification within 0 to 10 ms
N010
N020
Time
M600

Case 3: CPU Unit Cannot Receive Interrupts during Pass Mode


N010 G01 X100 F100 M600 D100
N020 G01 X200 F200
Decelerates to a stop if output is not possible within 10 ms
Speed
D100
N020
N010
Time
M600

Case 4: Continuous Interrupt Tasks Initiated


N010 G01 X100 F100 M600 D100
N020 G01 X200 F200 D200
N030 G01 X300 F300
An error is generated and the axis decelerates to a stop if the CPU Unit cannot receive interrupts within a maximum of 10 ms causing competition between the first and next interrupt notifications in the same task.
Speed
D100
D200
Up to 10 ms
N030
N010
N020
N030
Time

10 Stopover Function

The Stopover function outputs M code or D code without stopping operation when the axis is moved a preset amount (judged from the present position) in axis operation. It is used to control peripheral devices prior to completing a move and to improve tact time. The function can be used with G codes for all operations and with either a D code or a M code, but not both at the same time.

Format M <M code>/<Stopover> or D <D code>/<Stopover>

Case 1: N010 G01 X200 F200 D100/X100 Outputs D code 100 after moving the axis 100 mm on the X axis.

Case 2: N010 G01 X200 F200 M600/X100 Outputs D code 600 after moving the axis 100 mm on the X axis.

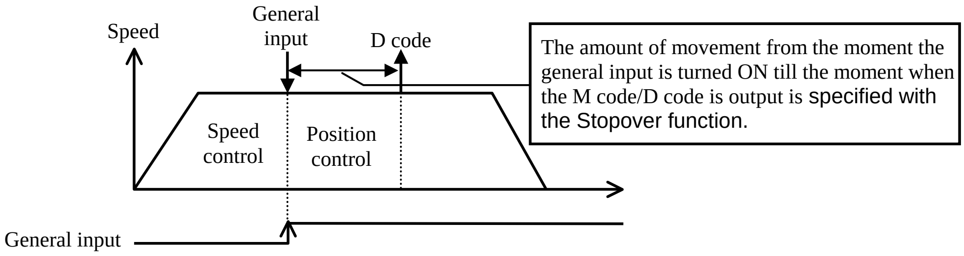
Operands

Stopover	
Axis	Coordinate data (travel distance)
X Y Z U	0 to 39999999 (E00) to (E32) A0000 to A1999

Description The Stopover function outputs a M code or D code when the axis moves a certain amount (judged from the present position) after the axis starts to move from the present position to the demand position. The travel distance (relative movement) is always specified in as a positive value. The Stopover function is used with the following G codes for positioning.

G00	POSITIONING
G01	LINEAR INTERPOLATION
G02	CLOCKWISE CIRCULAR INTERPOLATION
G03	COUNTERCLOCKWISE CIRCULAR INTERPOLATION
G26	REFERENCE ORIGIN RETURN
G27	WORKPIECE ORIGIN RETURN
G28	ORIGIN SEARCH
G31 (See note.)	INTERRUPT FEEDING
G32	TRAVERSE

Note With the G31 code, the Stopover function becomes valid after the general input turns ON.

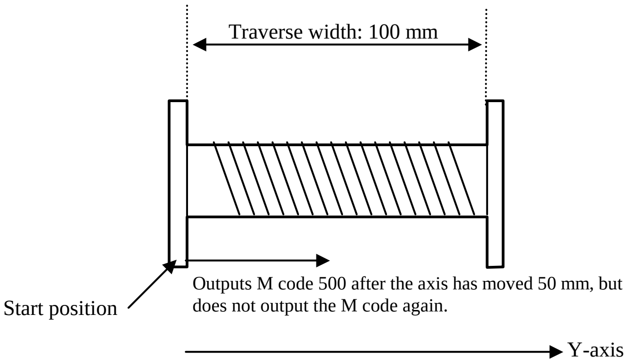


The amount of movement from the moment the general input is turned ON till the moment when the M code/D code is output is specified with the Stopover function.

For G codes in which more than one axis moves, such as in multi-turn circular interpolation and traverse operations, the Stopover function is executed only once during execution of the G code.

Example The following program outputs M code 500 after the axis has moved 50 mm in the Y-axis direction when the traverse operation shown below (traverse width:100 mm, number of windings per layer: 10, number of winding layers: 10) is performed. No D code will not be output beyond that point.

```
G32    X36000   Y100   F1000   L10   M500/Y50
```



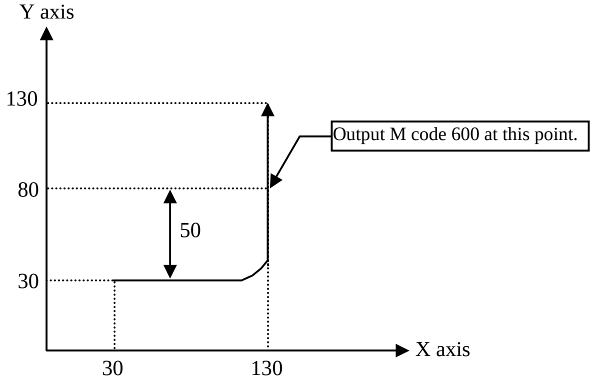
Outputs M code 500 after the axis has moved 50 mm, but does not output the M code again.

The following program outputs a M code or D code when the axis has moved the specified amount from the preceding demand position when the Stopover function is used in Pass Mode or In-position Check OFF Mode.

Example: (X, Y) = (30, 30)

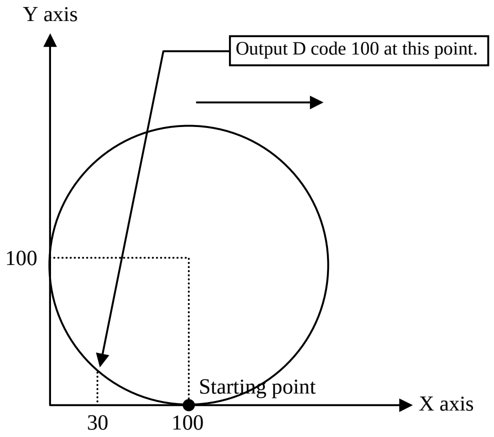
```
        :       :
N110    G10                                Pass Mode specified
N120    G01     X130    Y30     F100       Move to (X, Y) = (130, 30)
N130    G01     X130    Y130    M600/Y50
```

N130 is preread during execution of N120 because the system is in Pass Mode.
M code 600 is thus output after moving 50 mm from Y30 (the demand position for N120) to 80 mm (determined from the present position).



The following occurs with circular interpolation.
The D code is output when the axis has moved –70 mm on the X axis during circular interpolation as shown in the following figure.

```
N010    G02     X100    Y0      I0      J100    F100    D100/X70
```



11 Precautions on the Stopover Function

Heed the following precautions when using the Stopover function.

- Normally, the Stopover function is not valid if the specified distance moved is the same as the next demand position.

Example

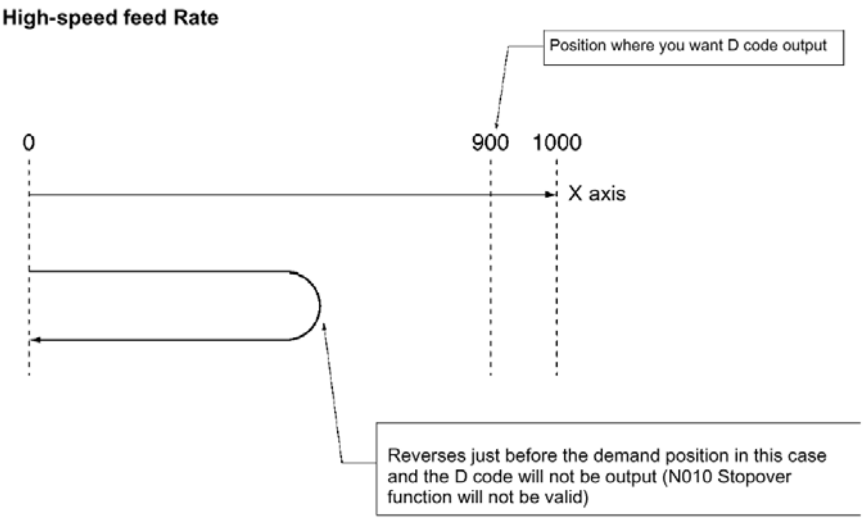
```
N0100   G00   X100
N0110   G00   X200   M500/X100
```

If the specified distance is 100 mm which is the same as the distance from X100 to X200, then the Stopover function will not be valid after the axis has moved to demand position X200. M code 500 will be output if the travel distance exceeds 100 mm (large X-axis overshoot) before the demand position X200 is obtained.

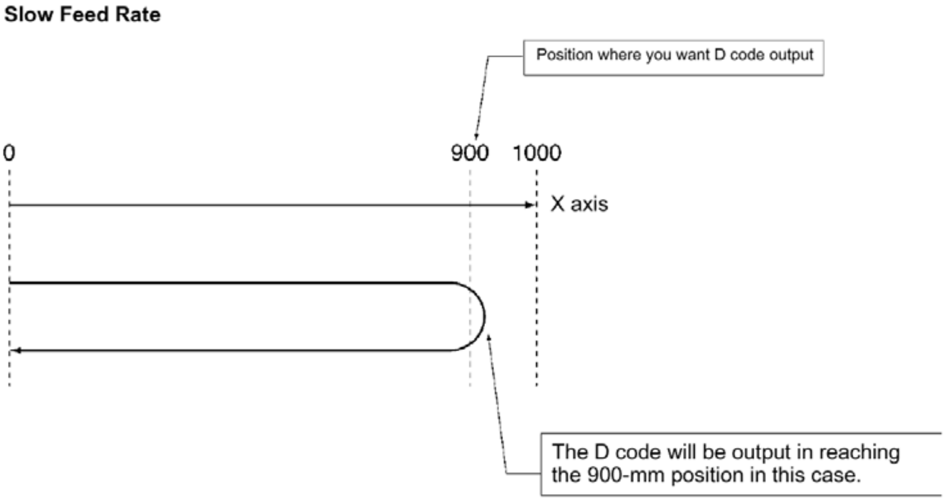
- The Stopover function may not be valid in operations where the axis moves back and forth between two points in Pass Mode.

Example

```
N000   P000   X
N010   G01    X1000F<High-speed feed rate> D100/X900
N020   G01    X0
```



Reverses just before the demand position in this case and the D code will not be output (N010 Stopover function will not be valid)



The D code will be output in reaching the 900-mm position in this case.

Appendix B
G-language Programming Examples

This section provides examples of G-language programming for the CS1W-MC421/221.

Programming Examples

1 Positioning with PTP Control

Operation Three positioning operations are performed and then the reference origin is returned to. An M code is output when positioning is completed.

1,2,3...

```
1. X100   Y50    (Output M code 20.)
2. X250   Y150
3. X300   Y200   (Output M code 700.)
4. Return to reference origin.
```

Programming Example

The following programming example performs the positioning operations listed above.

```
N000   P001   XY
N001   G00    X100   Y50    M20
N002   G00    X250   Y150
N003   G04    2
N004   G00    X300   Y200   M700
N005   G26    XY
N006   G79
```

Explanation

Block N000 declares the program number and axes being used.

N001

Positions the axes to (X100, Y50) by PTP control. Absolute positioning (default) is used, because nothing is specified.

When positioning is completed, M code 20 is output and the Unit waits for a reset (M code reset standby).

N002

When the M code reset is received, block N002 positions the axes to (X250, Y150) by PTP control.

N003

Waits for 2 seconds.

N004

Positions the axes to (X300, Y200) by PTP control. When positioning is completed, M code 700 is output and the next block is executed without waiting for the reset.

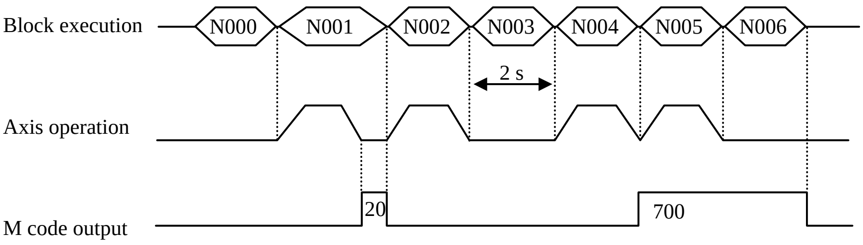
N005

Returns the X and Y axes to the reference origin.

N006

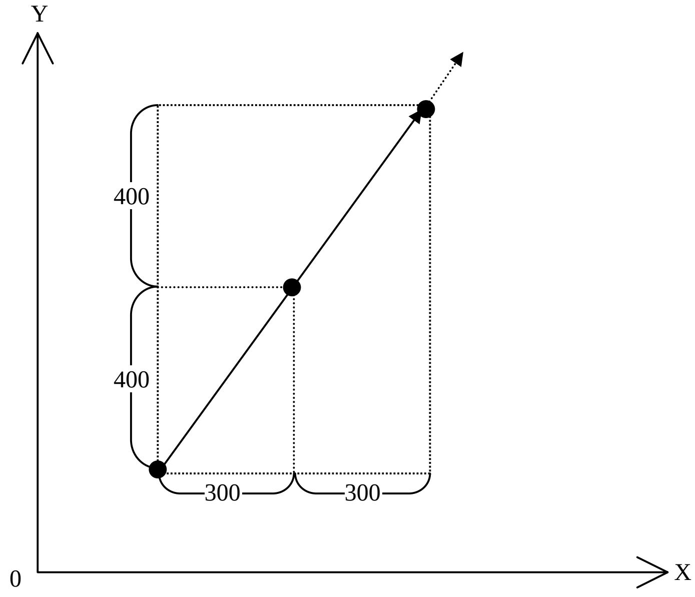
Ends the program. When M code 700 is being output, it is forcibly cleared by the PROGRAM END command.

Timing Chart



2 Positioning with Linear Interpolation

Operation When the optional input turns ON, this program uses linear interpolation to move the X-axis by 300 and the Y-axis by 400 from the present position. This positioning operation will be repeated (up to 21 times) until position data address A1000 contains 1.



Programming Example

```
N000  P002  XY
N001  G91
N002  G01   X300  Y400  F50   #16
N003  G71   N005/A1000=1
N004  G70   N002/L20
N005  G79
```

Explanation

Block N000 declares the program number and axes being used.

N001
Specifies incremental positioning.

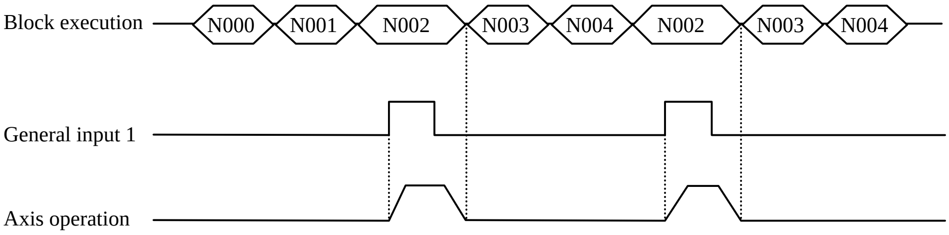
N002
Waits until optional input 16 (general input 1) turns ON. When it turns ON, the X-axis is moved by 300 and the Y-axis by 400 with linear interpolation at a speed of 50.

N003
Checks the content of A1000 and ends the program if it is 1.

N004
Jumps to block N002. Block N004 will jump to N002 20 times (performing 21 positioning operations).

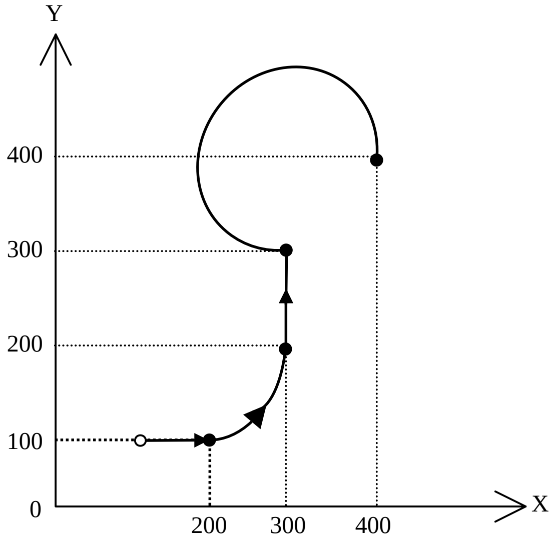
Note The operations above are performed in Stop Mode, and not Pass Mode, because an option is specified.

Timing Chart



3 Positioning with Linear and Circular Interpolation

Operation This program combines linear and circular interpolation to move the X and Y axes in the pattern shown in the following diagram.



Programming Example

```
N000   P003   XY
N001   G17
N002   G01    X200   Y100   F50
N003   G03    X300   Y200   R100
N004   G01    Y300
N005   G02    X400   Y400   R–100
N006   G79
```

Explanation

N001
Sets the X-Y plane as the plane for circular interpolation.

N002
Moves the axes to (X200, Y100) by linear interpolation.

N003
Moves the axes to (X300, Y200) by circular interpolation in the counter-clock-wise direction with a radius of 100. A positive value is specified for the radius, so an arc smaller than a semicircle (1/4 circle) is drawn.

N004
Moves the axes to (X300, Y300) by linear interpolation.

N005
Moves the axes to (X400, Y400) by circular interpolation in the clockwise direction with a radius of 100. A negative value is specified for the radius, so an arc greater than a semicircle (3/4 circle) is drawn.

4 Indirect Addressing with Registers

Operation This program uses registers to indirectly specify position data stored in position data addresses and uses this data in positioning operations.

Programming Example

```
N000   P005   XY
N001   G11
N002   G63    E1=100
N003   G63    E2=101
N004   G01    X(E1) Y(E2) F100
N005   G60    E1=E1+2
N006   G60    E2=E2+2
N007   G75    3
N008   G70    N004/L3
N009   G79
```

Explanation

N001
Specifies Stop Mode.

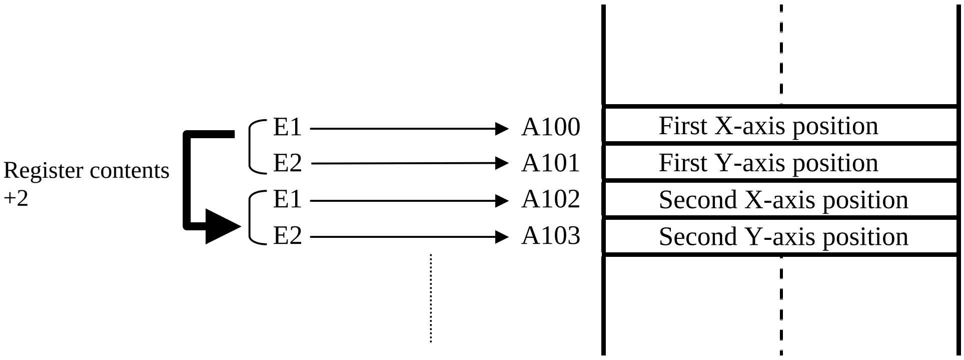
N002 through N003
Set the initial position data addresses in the registers.

N004
Reads the position data from the addresses specified in the registers and moves the axes to this point by linear interpolation.

N005 through N006
Updates the contents of the registers.

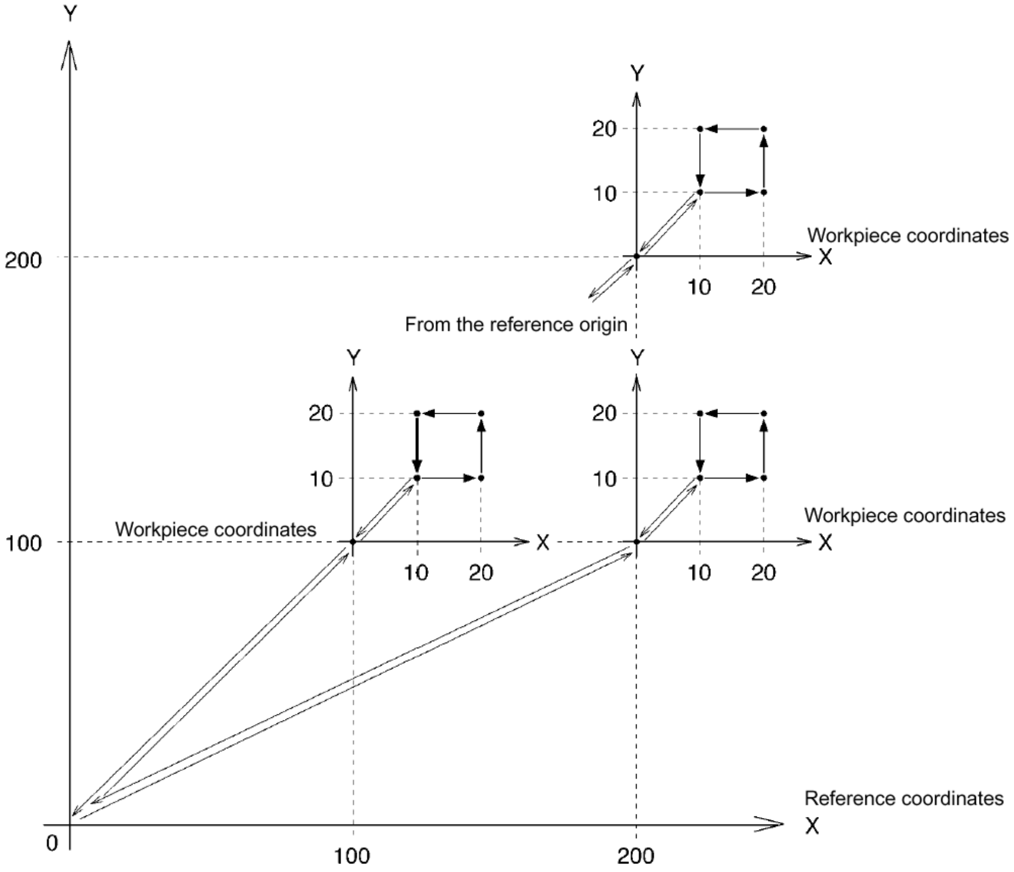
N007
If optional input 3 is ON, the next block will be skipped and the program will end.

N008
Jumps to block N004 and repeats the positioning operation 3 times.



5 Using the Workpiece Origin Offset

Operation Positioning for the same pattern can be performed any number of times by repeatedly changing the workpiece origin offset and calling the subroutine. Using the workpiece coordinate system is useful particularly for absolute positioning



Programming Example

```
N000   P006   XY
N001   G53    X100   Y100
N002   G72    P800
N003   G53    X200
N004   G72    P800
N005   G53    Y200
N006   G72    P800
N007   G79
```

Subroutine

```
N000   P800   XY
N001   G27    XY     M10
N002   G01    X10    Y10    F10
N003   G01    X20
N004   G01    Y20
N005   G01    X10
N006   G01    Y10
N007   G27    XY     M20
N008   G26    XY     M30
N009   G73
```

Explanation

P006: N001
Sets the workpiece origin offset to (X100, Y100).

P006: N002
Calls subroutine P800.

P800: N001
Returns to the workpiece origin. The workpiece coordinate system is selected at this time. M code M10 is output and the Unit waits for the M code reset.

P800: N002 through N006
The subroutine’s series of absolute positioning operations is performed by linear interpolation in Pass Mode.

P800: N007
Returns to the workpiece origin. M code M20 is output and the Unit waits for the M code reset.

P800: N008
When the reset is received, block N008 returns to the reference origin. M code M30 is output and the Unit waits for the M code reset.

P800: N009
When the reset is received, block N009 returns to main program P006.

P006: N003
Changes the workpiece origin offset to X200.

P006: N004
Calls subroutine P800 and repeats the same series of operations.

6 Changing the Interpolation Acceleration Time

Operation This program changes the setting of the interpolation acceleration time parameter and performs linear interpolation.

Programming Example

```
N000   P007   XY
N001   G69    #3/X500
N002   G01    X1000  Y500    F300
N003   G01    X2000  Y–1000
N004   M20
N005   G69    #3/X100
N006   G01    X3000  Y1000
N007   G01    X4000  Y1500
N008   G79
```

Explanation

N001
Changes the setting of the X-axis’ interpolation acceleration time parameter to 500 ms, which becomes the X-axis acceleration time used for interpolation in this task.

N002
Moves to (X1000, Y500) by linear interpolation with an acceleration time of 500 ms and speed of 300.

N003
Moves to (X2000, Y–1000) in Pass Mode with an acceleration time of 500 ms.

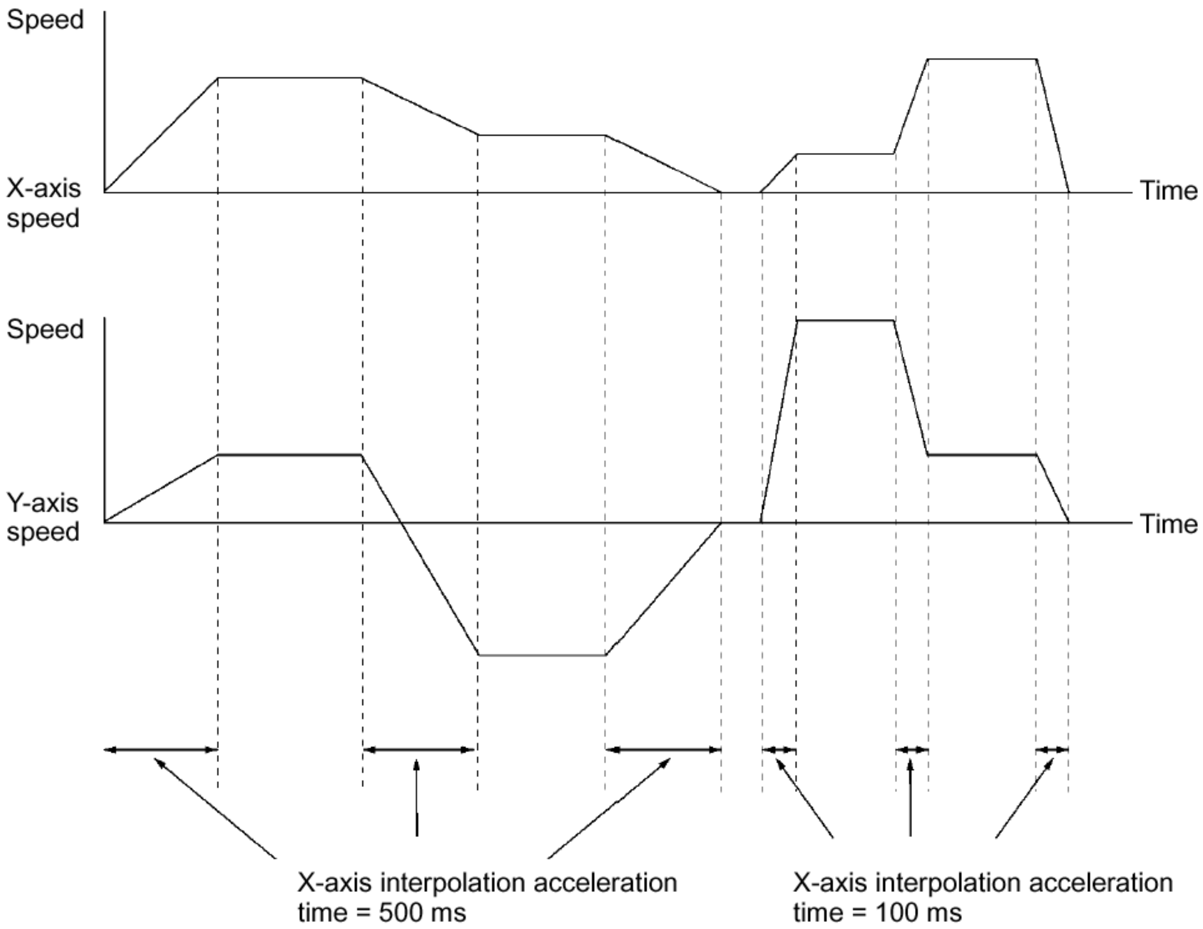
N004
Once the pass operation is completed, N004 outputs M code M20 and waits for the M code reset.

N005
Changes the setting of the interpolation acceleration time to 100 ms.

N006
Moves to (X3000, Y1000) by linear interpolation with an acceleration time of 100 ms.

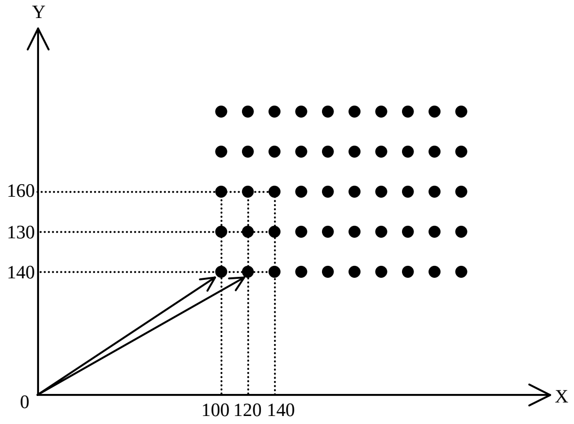
N007
Moves to (X4000, Y1500) in Pass Mode with an acceleration time of 100 ms.

Timing Charts



7 Positioning while Calculating Position Data

Operation After initializing the position data, this program moves the axes 10 times while incrementing the X-axis’ position data by 20. Next, the Y-axis position data is incremented by 30 and the positioning operation is repeated five times. The positioning will be in grid as shown in the following diagram.



Programming Example 1

```
N000   P008   XY
N001   G11
N002   G63    E00=0
N003   G63    A1000=100
N004   G63    A1001=100
N005   G01    XA1000   YA1001   F20   M10
N006   G01    X0       Y0       F50
N007   G60    A1000=A1000+20
N008   G70    N005/L9
N009   G63    A1000=100
N010   G60    A1001=A1001+30
N011   G60    E00=E00+1
N012   G71    N005/E0  !  5
N013   G79
```

Explanation

N001
Specifies Stop Mode.

N002
Initializes register E0 to 0.

N003
Substitutes the X-axis' initial position data into address A1000.

N004
Substitutes the Y-axis' initial position data into address A1001.

N005
Moves the axes by linear interpolation to the X position indicated in A1000 and the Y position indicated in A1001.

N006
Returns to the origin.

N007
Increments the X-axis' position data by 20.

N008
Jumps to block N004 and repeats the above process 9 times.

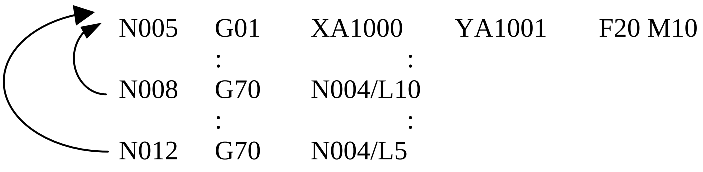
N009
Initializes the X-axis' position data to its original value.

N010
Increments the Y-axis' position data by 30.

N011
Uses register E0 as a loop counter with an initial value of 0. The content of E0 is incremented by 1.

N012
Jumps to N004 as long as the content of E0 is not 5. When E0=5, N012 proceeds to the next block and ends the program.

Note Register E0 is used as a loop counter in block N012 because loops cannot be nested as shown below.



```
N005   G01   XA1000     YA1001    F20 M10
       :           :
N008   G70   N004/L10
       :           :
N012   G70   N004/L5
```

The loops in the example above will not operate properly. A subroutine can also be used for nesting, as shown in the following programming example.

Programming Example 2

```
N000   P008   XY
N001   G11
N002   G63    A1000=100
N003   G63    A1001=100
N004   G72    P700
N005   G63    A1000=100
N006   G60    A1001=A1001+30
N007   G70    N004/L4
N008   G79
```

Subroutine

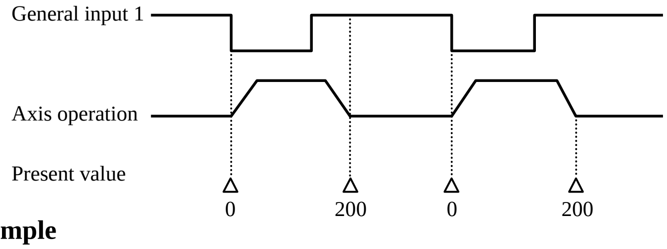
```
N000   P700   XY
N001   G01    XA1000   YA1001   F20   M10
N002   G01    X0       Y0       F50
N003   G60    A1000=A1000+20
N004   G70    N001/L9
N005   G73
```

Explanation

Blocks N005 through N008 in programming example 1 have been converted to subroutine format in subroutine P700, which is called from step N004 in main program P008. SUBROUTINE END (G73) is required at the end of the P700.

8 Stopping a Program with a General Input

Operation This program shows how to stop MC program execution when general input1 turns ON.



Programming Example

```
N000   P009   X
N001   G11
N002   G76    16
N003   G54    X0
N004   G01    X200   F100
N005   G70    N001
N006   G79
```

Explanation

N001

Specifies Stop Mode.

N002

Stops execution when general input 1 turns ON. (Proceeds to the next block when general input 1 is OFF.)

N003

Presets the X-axis’ present value to 0.

N004

Absolute positioning of the X axis.

N005

Unconditionally jumps to block N001, repeating blocks N001 through N004. The process can be repeated indefinitely without an overflow because the present value is preset to 0 each time.

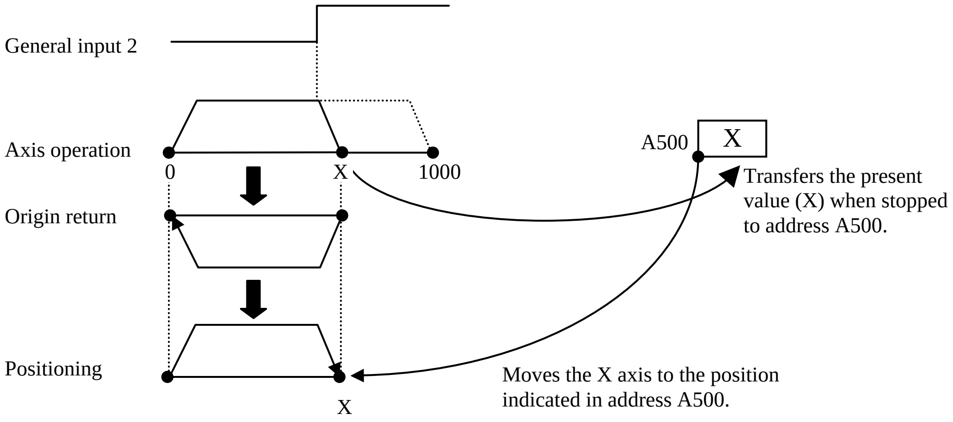
N006

Ends the program.

9 Stopping the Program and Substituting Position Data

Operation This program moves the X axis to X1000 at a speed of 100 by linear interpolation. The movement will be decelerated to a stop by the G74 (OPTIONAL END) command if the target position is reached before the positioning operation is completed.

The X position when the movement was stopped will be stored in address A500 and that position data will be used for later positioning operations. This process is useful for applications in which the position where the operation was stopped will be used for positioning rather than the original target position.


General input 2
Axis operation
0
X
1000
Origin return
Positioning
X
A500
X
Transfers the present value (X) when stopped to address A500.
Moves the X axis to the position indicated in address A500.

Programming Example

```
N000    P010    X
N001    G11
N002    G74     17
N003    G01     X1000 F100
N004    G63     A500=X
N005    G01     X0
N006    G01     XA500
N007    G70     N005
N008    G79
```

Explanation

N001

Specifies Stop Mode.

N002

Execution of the next block is stopped when general input 2 turns ON.

N003

Moves the X axis with a maximum target position of X1000. The positioning operation will be decelerated to a stop if general input 2 turns ON before positioning is completed.

N004

The stopping position is stored in address A500.

N005

Returns to the origin.

N006

Positions the X axis using the position data stored in address A500 in block N004.

N007

Jumps to N005 and repeats the positioning operation.

10 Positioning a Turntable

Operation This operation positions a turntable using the unlimited feed mode and various arithmetic commands. Positioning is performed while judging whether to move clockwise or counterclockwise to arrive at the target position. Target positions are specified between 0° and 360° and the present position will return to zero if a target position is specified outside that range.

Use the following procedure to specify positioning from the PC.

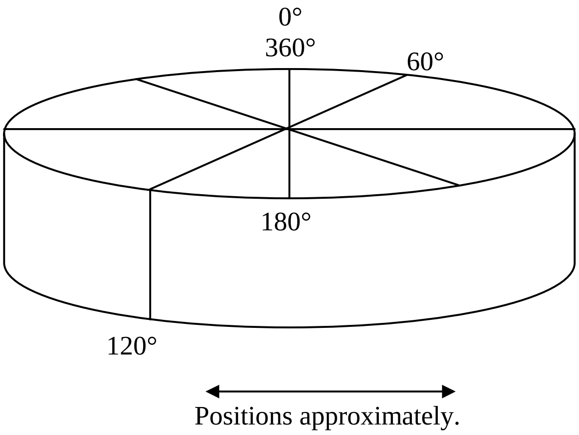
1,2,3...

1. Use the IOWR instruction to write the target position into address A0000 between 0° and 360°.

2. After the target position is specified, turn ON optional input zero and the turntable will be positioned at the target position.

3. M code 0001 will be output after positioning is completed. Check the M code, turn OFF optional input zero and turn the M code reset ON and OFF.

4. Repeat steps 1 to 3.



Positions approximately.

Programming Example

```
N000    P000    X
*001
*002    Target position=A0000, reference position (program setting)=A0001
*003    Present position (used internally)=A0100, work memory=A0101,A0102
*004
*005    Set the target position into A0000 by IOWR instruction.
*006    Waits until optional number zero turns ON.
*007
N008    G75     00
N009    G70     N008
*010
*011    Positions approximately.
*012
N013    G63     A0100=X
N014    G60     A0102=A0000–A0100
N015    G71     N020/A0102<–180
N016    G71     N022/A0102<180
N017    G71     N024/A0102<360
N018    G60     A0102=A0102–360
N019    G70     N015
N020    G60     A0001=A0000+360
N021    G70     N025
N022    G63     A0001=A0000
N023    G70     N025
N024    G60     A0001–A0000–360
N025    G00     XA0001          M001
N026    G70     N008
N027    G79
```

Explanation

Block N000 declares the program number and axis being used.

***001 to *007**

Comments.

N008 and N009

Waits until optional input zero turns ON. When the input turns ON, unconditional jump N009 is disabled and the next block is executed (*010).

***010 to *012**

Comments.

N013

Substitutes the X-axis position into A0100.

N014 to N017

Determines whether to perform positioning in the clockwise or counterclockwise direction based on the target position (A0000) and the present position (A0100), and then jumps to individual processes.

N018 and N019

Corrects to a target position within one revolution if a target position of more than one revolution is specified, and restarts positioning from N014.

N020 to N024

Turns the turntable clockwise or counterclockwise and calculates target position A0001 where the turntable will be positioned.

N025

Positions the turntable at the target position and then outputs M code 0001 and turns ON the M strobe, and then waits for M code reset from the PC.

N026

After the M code reset from the PC has turned ON and OFF, jumps to the process for waiting for optional input zero to turn ON.

System Parameter Settings

The following system parameter settings must be made.

• Minimum setting unit	Set for the user's system.
• Axis mode	Set to Unlimited Feed Mode.
• Software limits	Set between 0° and 360°. If the minimum setting unit is 0.1, then the present position will be updated between 0.0° and 359.9°.

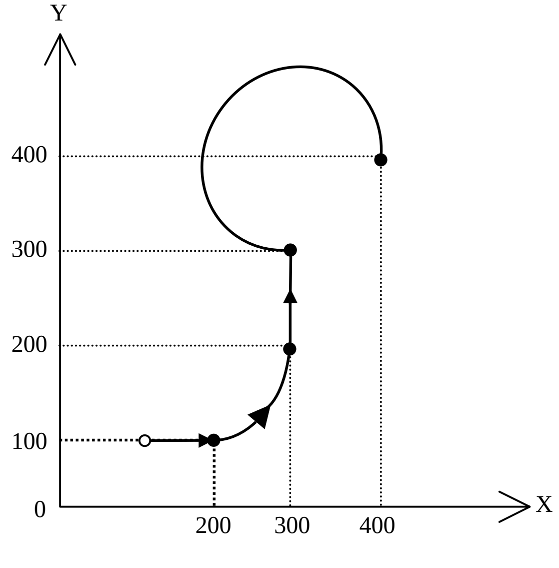
Note In the above application, set the encoder resolution and pulse rate so that a revolution (360°) is an integer when it is converted to pulses.

If positioning that passes though 0°, such as passing through 0° to shift the position from 350° to 10° or passing through 0° to shift the position from 10° to 350°, is performed repeatedly in a system that generates a remainder when a revolution (360°) is converted to pulses, an error of less than one pulse will occur per revolution.

Work within the allowable range for accuracy and execute ORIGIN SEARCH for each operation in systems like that described above.

11 Positioning with Improved Locus Accuracy in Circular Interpolation

Operation This operation is essentially the same as described in 8-1-3 Positioning with Linear and Circular Interpolation except that locus accuracy in the circle will be improved. When performing circular interpolation, the axes will follow the inner side of the expected circle as the feed rate increases. Locus accuracy will be improved by increasing the feed forward gain of the X and Y axes using G69 (change parameter). Normally overshooting will occur when the axes stop or when the target position is inverted if the feed forward gain in the servo system is increased too much. Therefore, set the feed forward gain to 20% for linear interpolation only operation and 60% for circular interpolation-only operation.



Programming Example

```
N000   P003   XY
N001   G17
N002   G69    #6/X20         Y20
N003   G01    X200    Y100   F50
N004   G69    #6/X60         Y60
N005   G02    X300    Y200   R100
N006   G69    #6/X20         Y20
N007   G01    Y300
N008   G69    #6/X60         Y60
N009   G02    X400    Y400   R–100
N010   G79
```

Explanation

N002, N006: Sets feed forward gain for the X and Y axes to 20% for linear interpolation operation.

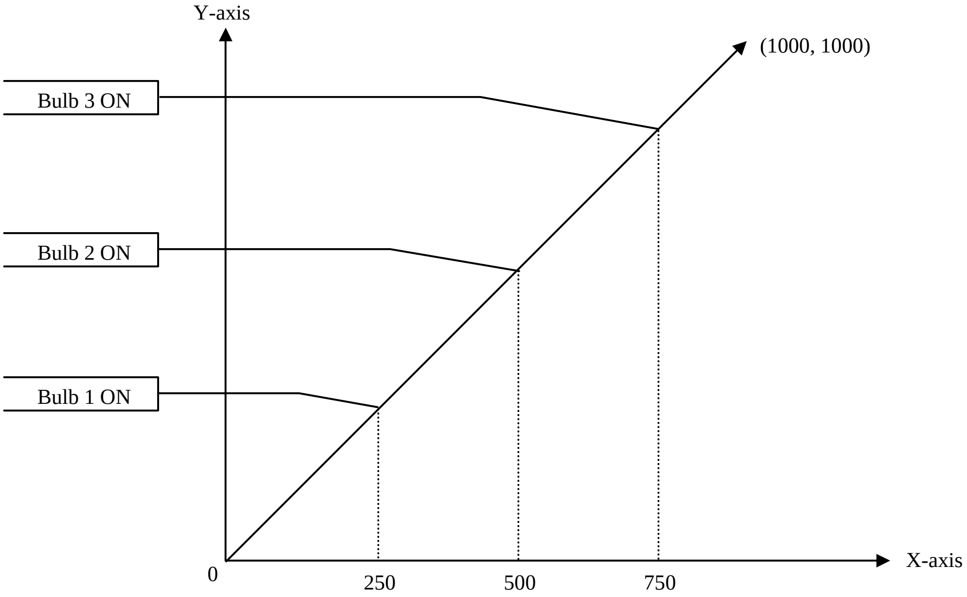
N004, N008: Sets feed forward gain for the X and Y axes to 60% only for circular interpolation operation.

See 8-1-3 Positioning With Linear and Circular Interpolation for details on other settings.

12 Starting Peripheral Devices during Operation

Operation Peripheral devices can be started at high speed from the MC Unit using the MC Unit stopover function or interrupt notification function (D code) as well as a CPU interrupt task. This program allows high-speed synchronous applications to be created, because it is able to control peripheral devices without pausing operation.

The following program turns ON three valves (1, 2 and 3) during two-axis linear interpolation when the present positions on the X axis are 250, 500 and 750 mm.



Programming Example

```
N000   P001   XY
*001   ********************************
*002   Executes interrupt notification.
*003   ********************************
N004   G01    X0      Y0      F100
N005   G75    N005
N007   G01    X250    Y250    F1000
N008   G01    X500    Y500    D100/X0
N009   G01    X750    Y750    D101/X0
N010   G01    X1000   Y1000   D102/X0
N011   G79
```

Explanation

This operation creates a ladder program for the following interrupt tasks in the CPU Unit.

Interrupt Task Number 100:

Turns ON the output on the Output Unit that controls valve 1.

Interrupt Task Number 101:

Turns ON the output on the Output Unit that controls valve 2.

Interrupt task number 102:

Turns ON the output on the Output Unit that controls valve 3.

Positioning will start when optional input zero turns ON.

N004

Moves the axis to the start position.

N005 and N006

Waits until optional input zero turns ON. When the input turns ON, unconditional jump N0006 is disabled and the next block will be executed.

N007

Moves by linear interpolation to (X, Y) = (250, 250) in Pass Mode.

N008

Moves by linear interpolation to (X, Y) = (500, 500) in Pass Mode. This specifies the point where interrupt task number 100 (D100) will start up if the travel distance is zero when the Stopover function is used. The following shows where interrupt task number 100 (D100) will start up since Pass Mode operation is being used in this case.

Present position = preceding X-axis target position (N007) + stopover travel distance
= 250 + 0
= 250

In other words, interrupt task number 100 will start up when the present position is 250 mm.

N009 and N010

Similarly, interrupt task numbers 101 and 102 will start up at present positions 500 mm and 750 mm respectively.

13 Positioning at High Speed

This program reads the present position of the workpiece with a visual sensor, calculates correction and writes the correction value to the MC Unit. After the MC Unit receives the correction, it positions the axis accordingly. The CPU Unit sends two position data items to addresses A0000 and A0001 using the IOWR instruction and the MC Unit moves the axis immediately. Data can be sent to the MC Unit and positioning commands can be executed in a single scan.

Programming Example

```
N000   P000   XY
*001   *********************************
*002   Waits for M code reset.
*003   *********************************
N004   G01    X0       Y0       M500
N005   G75    00
N006   G70    N005
N007   G01    XA0000            YA0001            F1000   M001
N008   G70    N004
N009   G79
```

Explanation

N004

Waits at the reference coordinate system origin and outputs M code 500.

N005 and N006

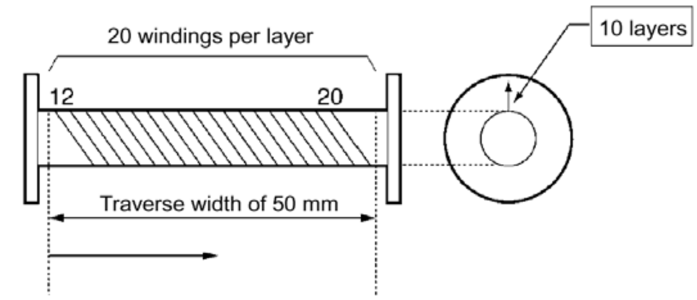
Waits for optional number zero to turn ON. Writes position data from the CPU Unit by IOWR instruction and turns ON optional number zero.

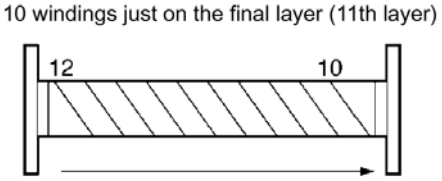
N007

The MC Unit positions the axis at the position specified in A0000 and A0001, outputs M code 0001, and waits for M code reset. When the reset is received from the CPU Unit, the MC Unit moves the axis to the standby position.

14 Shifting from Aligned to Rough Winding

The following example shows what happens when 10 layers are created with wire at the rate of 20 windings per layer, and then 10 windings are applied just on the 11th layer.





Programming Example

```
N000   P000   XY
N010   G32    X7200.0    Y50.0 F2000 L10
N020   G32    X3600.0    Y50.0 F2000 L1
N030   G79
```

Explanation

N010

Specifies 360.0° x 20 revolutions = 7200.0° on the X axis in order to get 20 windings per layer. It also specifies L10 in order to get 10 winding layers and because operation is in Pass Mode (default value for interpolated operation) it executes N020 without a break.

N020

Specifies 360.0° x 10 revolutions = 3600.0° on the X axis in order to get 10 windings per layer.

Executing MC Programs from the Ladder Program

This section describes ladder programs that will execute G-language MC programs. Use the following procedure to execute MC programs.

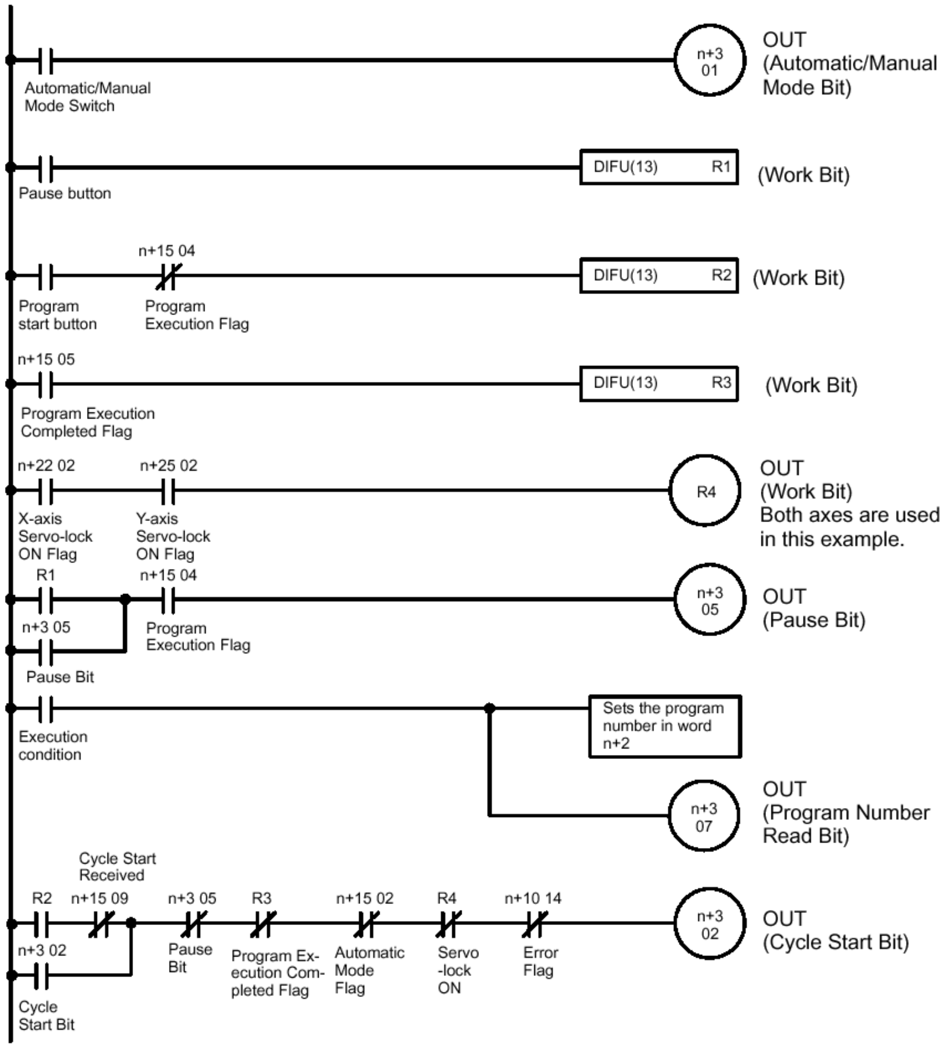
1,2,3...
1. Set the MC Unit to Automatic Mode. (Turn ON bit 01 of word n+3.)
2. Set the program number of the desired MC program. (Word n+2)
3. Read the program number from word n+2 to the MC Unit. (Turn ON bit 07 of word n+3.)
4. Turn ON the Cycle Start Bit to execute the MC program specified in word n+2. (Turn ON bit 02 of word n+3.)

In this example procedure, task 1 is executed. The following table shows the equivalent IR Area control bits and IR Area words for tasks 2 to 4.

Task	Automatic Mode Bit	Program number	Program Number Read Bit	Cycle Start Bit
2	n+5: bit 01	n+4	n+5: bit 07	n+5: bit 02
3	n+7: bit 01	n+6	n+7: bit 07	n+7: bit 02
4	n+9: bit 01	n+8	n+9: bit 07	n+9: bit 02

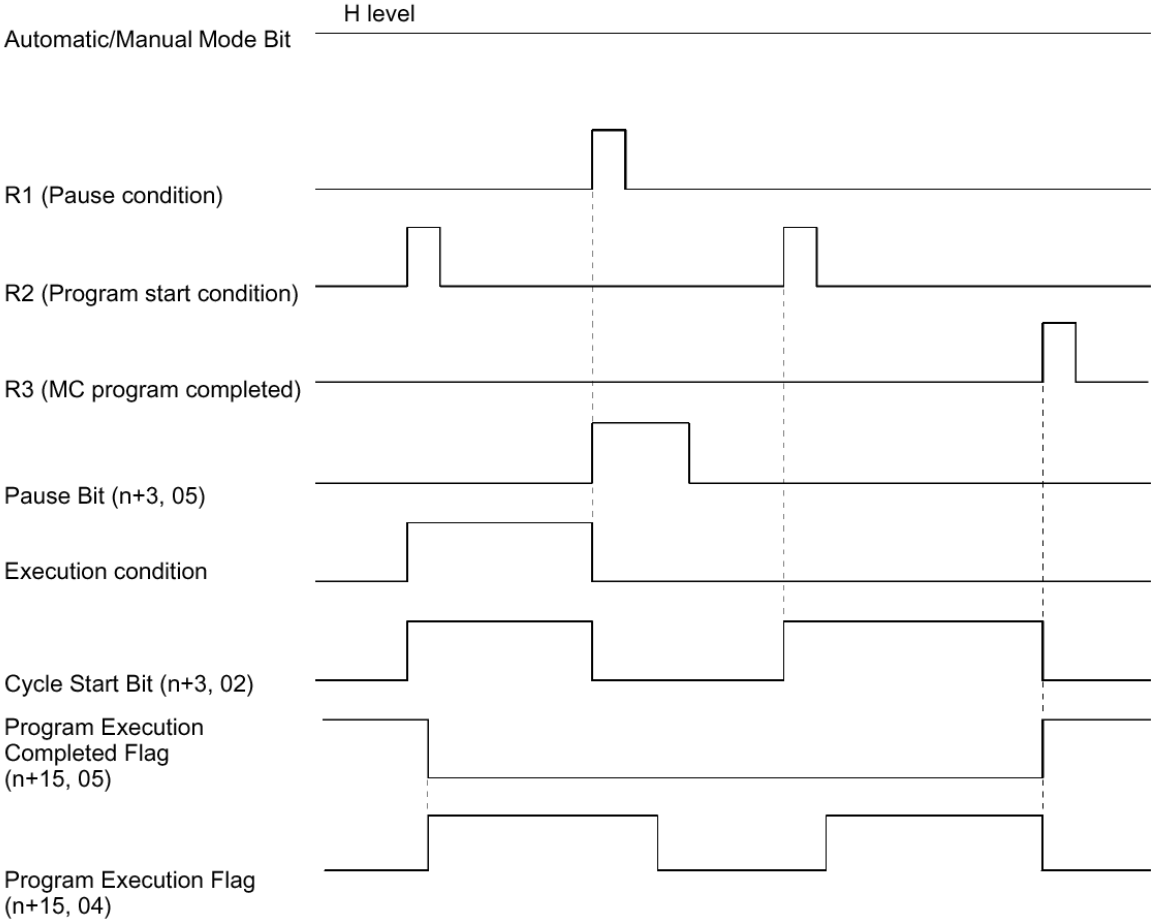
Ladder Programming Example

The following ladder program executes Task 1 in the MC Unit.



Note When executing a positioning operation in the MC program, all of the axes being used must be servo-locked (Servo-lock ON Flags must be turned ON), so be sure to add a condition like R4 in the programming example above.

Timing Chart


H level
Automatic/Manual Mode Bit
R1 (Pause condition)
R2 (Program start condition)
R3 (MC program completed)
Pause Bit (n+3, 05)
Execution condition
Cycle Start Bit (n+3, 02)
Program Execution Completed Flag (n+15, 05)
Program Execution Flag (n+15, 04)

Appendix C
Basic Position Operations

This section explains the basic positioning operations executed by the MC Unit.

Overview

This section describes the operations required for basic positioning with an MC Unit, and explains how to use them. It also explains the functions that have been newly added or changed from those of the earlier C200H-MC221 MC Unit.

There are four ways to execute operations, as follows:
- Running a G-language program.
- Executing operations with the PC Interface Area in the CPU Unit.
- Executing operations with the Command Area.
- Setting system parameters

Notation The bit status is indicated as follows in the explanations of positioning operations:
↑: A transition in the bit from 0 (OFF) to 1 (ON) (See note.)
↓: A transition in the bit from 1 (ON) to 0 (OFF) (See note.)
1: ON
0: OFF

Note Transitions in bits are determined in the MC Unit and are not always detectable when using the DIFU and DIFD instructions in the program in the CPU Unit.

1 Summary of Operations

Operation name	Basic function	Mode	Execution method
PTP control	Positions each axis independently	Automatic	G language (G00)
Linear interpolation	Performs linear interpolation on 1, 2, 3, or 4 axes (1 or 2 axes for MC221). The specified axes move simultaneously. The feed rate can be specified.	Automatic	G language (G01)
Circular interpolation	Positions using two axes, following an arc from the specified starting point to the specified end point.	Automatic	G language (G02/G03)
Helical circular interpolation	Adds another axis in parallel to circular interpolation.	Automatic	G language (G02/G03)
Interrupt feeding	Feeds along an axis for a fixed distance when an external interrupt is input. Feeding is also possible by specifying a given travel distance even when no interrupt is input.	Automatic	G language (G31)
Traverse	Used for winding machine operations	Automatic	G language (G32)
Operating mode	Any of the following modes can be specified for positioning operations: Stop Mode, Pass Mode, or In-position Check OFF Mode.	Automatic	G language (G10, G11, G13), system setup
Change parameter	Changes servo system parameters using G language.	Automatic	G language (G69)
Stopover	Outputs an M code or D code at a specified position during axis operation.	Automatic	G language (G00, G01)
CPU Unit interrupt processing	Executes an interrupt program with the D code (output with the stopover operation) as the interrupt program number.	Automatic	G language (D code output)
Override	Changes the speed by applying a specified factor.	Automatic, manual	Bit designation

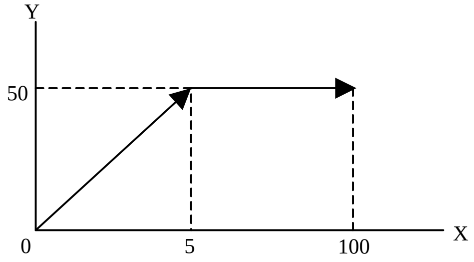
Operation name	Basic function	Mode	Execution method
Error counter reset	Forcibly resets the error counter to zero, and stops axis operation.	Automatic, manual	Bit designation
Servo-lock	Creates a position loop at the MC Unit.	Manual	Bit designation
Servo-unlock	Releases the position loop at the MC Unit.	Automatic, manual	Bit designation
Backlash correction	Compensates for backlash in the mechanical system.	Automatic, manual	Command Area designation, CX-Motion
Automatic loading	Downloads G-language programs, position data, and system parameters to the MC Unit from an external memory device connected to a personal computer running CX-Motion.	Automatic, manual	Command Area designation, CX-Motion
Present position preset	Presets the present position to any specified value.	Manual	Command Area designation
Electronic gear	Sets the per/pulse ratio for an MPG or sync encoder.	Manual	System parameter setting (by IOWR or CX-Motion)
Acceleration/ deceleration curve	Specifies the acceleration and deceleration method for how each axis is started or stopped.	Automatic, manual	System parameter setting (by CX-Motion only)
Unlimited feeding	Specifies control for an axis to be fed Continuously, such as for a conveyer, for example.	Automatic, manual	System parameter setting (by CX-Motion only)
Stop	Describes how to stop the axis operation.	Automatic, manual	Bit designation, software limit (system parameter), external input (emergency stop, limit input), etc.

PTP Control

Overview

Mode: Automatic; **Method:** G language (G00) Positioning each axis independently from the other axes is called PTP control. Each axis moves at the preset feed rates: at the acceleration for the preset acceleration time, the maximum feed rate, and the deceleration for the preset deceleration time.

For example, suppose a control program is executed for moving from the origin to the X-axis coordinate of 100 and Y-axis coordinate of 50. If the feed rates, the acceleration times, and the deceleration times are the same for both axes, X-axis and Y-axis movements will be as illustrated below.



The illustration shows movements when the high-speed feed rate for the X axis is set to the same feed rate as the Y axis.

Both the X-axis and Y-axis move to a coordinate of 50 over the same duration of time. At this point, the Y-axis stops and the X-axis moves to a coordinate of 100.

Jogging in the Manual Mode is also performed using PTP control using the maximum jogging feed rate.

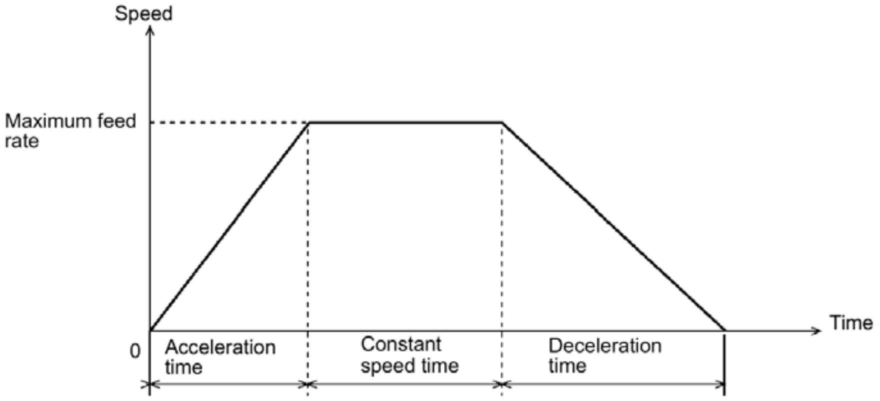
Acceleration Time and Deceleration Time

Acceleration and deceleration times under the PTP control are as follows:

Acceleration time: Time required until the single axis speed reaches the maximum feed rate

Deceleration time: Time required until the speed control voltage drops to zero from the maximum feed rate of the single axis.

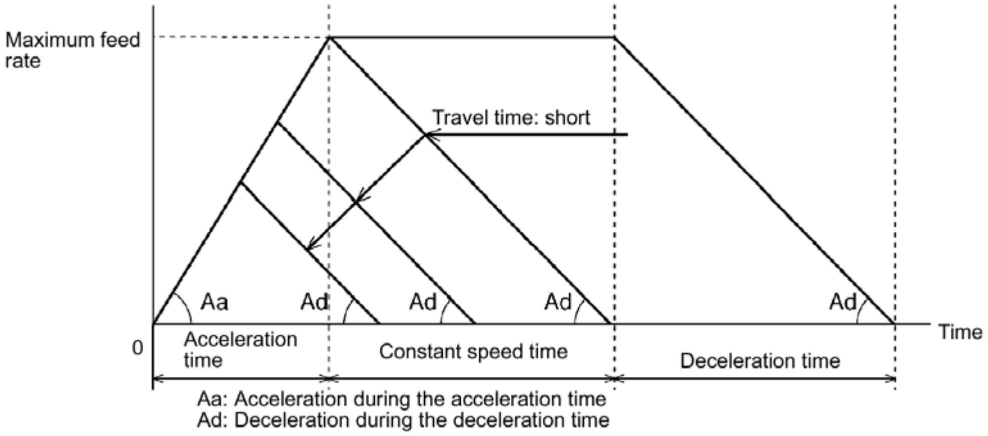
The acceleration time and deceleration time (feed rate parameters) are set either from the CX-Motion or by using IOWR in the ladder program.



Overrides can be set to alter the acceleration time and deceleration time. The acceleration speed, however, will remain constant.

Triangular Control

If the travel time is shorter than the sum of acceleration time and deceleration time, triangular control will occur as shown below.



The ratio of acceleration time and deceleration time in the triangular control will be the same as the ratio of acceleration time and deceleration time set in the feed rate parameters.

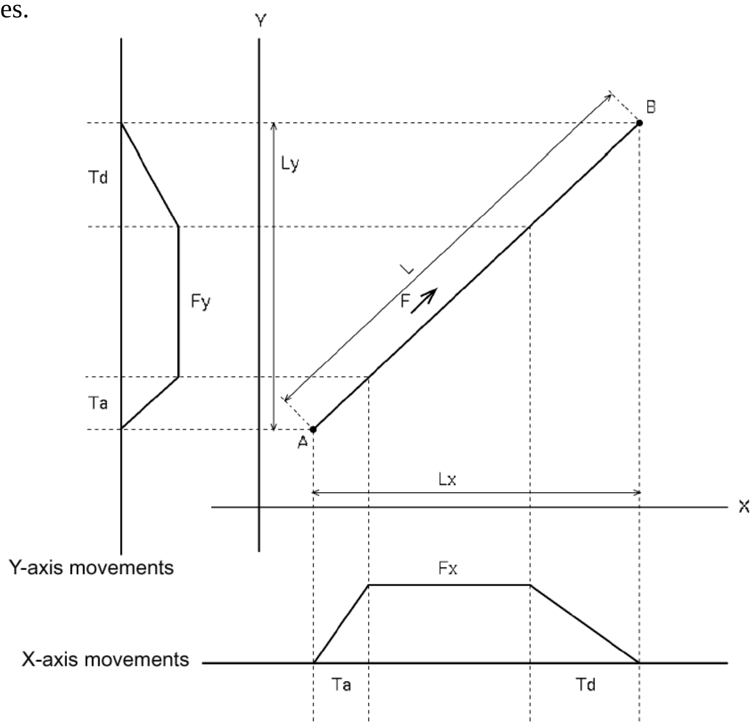
Linear Interpolation

Overview

Mode: Automatic; **Method:** G language (G01)

Positioning with linear interpolation produces a straight line that connects a preset starting point to a preset end point using all specified axes (X to U).

Linear interpolation from the point A to the point B will be as shown below when using the X and Y axes.



F: Designated interpolation feed rate
Fx: Interpolation feed rate of the X axis based on F
tFy: Interpolation feed rate of the Y axis based on F
Ta: Interpolation acceleration time
Td: Interpolation deceleration time
Fx and Fy can be expressed as follows:
Fx= Lx/L x F
Fy= Ly/L x F
Where, L is the travel distance in the specified locus, Lx is the travel distance along the X-axis, and Ly is the travel distance along the Y-axis.

Interpolation Acceleration and Deceleration Times

Interpolation acceleration and deceleration times for linear interpolation are defined as follows:

Interpolation acceleration time: Time required to reach the specified interpolation feed rate on the composite axial locus.

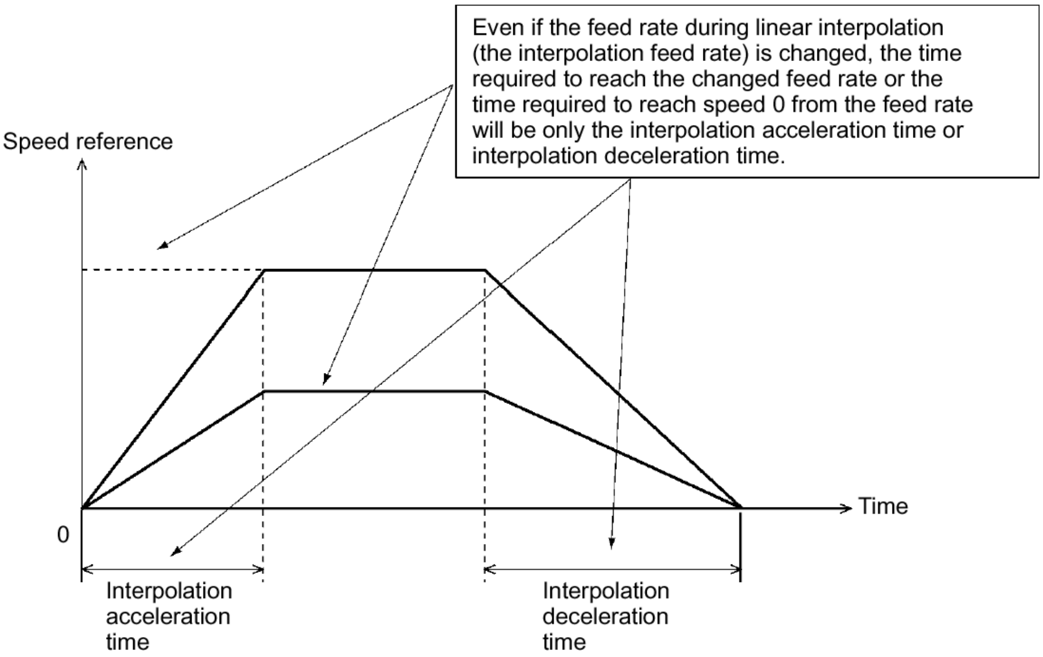
Interpolation deceleration time: Time required until the speed control voltage drops to zero from the specified interpolation feed rate on the composite axial locus.

The acceleration time and deceleration time (feed rate parameters) are set either from the CX-Motion or by using IOWR in the ladder program.

Unlike PTP control, linear interpolation acceleration and deceleration times are not affected by the speed. Acceleration changes according to the movement to satisfy the preset interpolation acceleration and deceleration times.

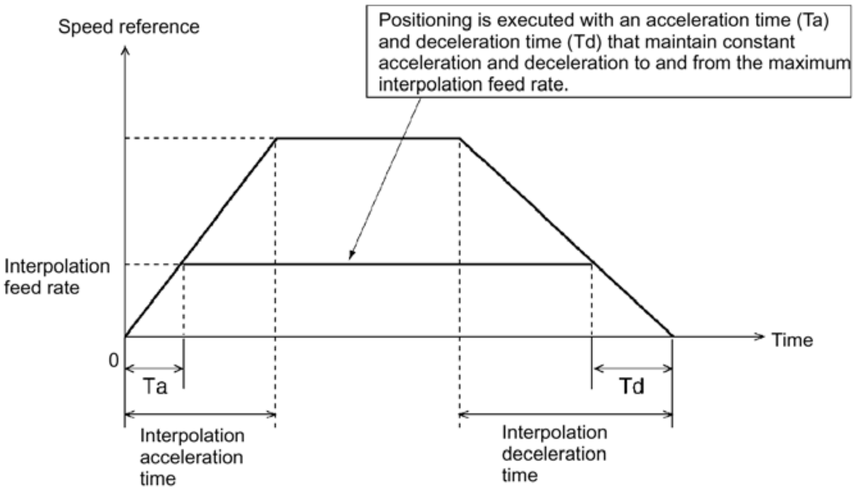
Constant Acceleration Mode

When positioning is executed using linear interpolation, the normal operation (i.e., the CX-Motion default setting) is for positioning to accelerate at the interpolation acceleration time until the specified interpolation feed rate is reached, and for positioning to decelerate at the interpolation deceleration time. Unlike PTP control, the acceleration or deceleration time is not shortened according to the speed (except for triangular control).


Even if the feed rate during linear interpolation (the interpolation feed rate) is changed, the time required to reach the changed feed rate or the time required to reach speed 0 from the feed rate will be only the interpolation acceleration time or interpolation deceleration time.
Speed reference
Time
0
Interpolation acceleration time
Interpolation deceleration time

The Constant Acceleration Mode is provided to shorten the positioning time when only one axis is used for linear interpolation. This mode is normally set to OFF (i.e., it is turned OFF in the CX-Motion default settings). When it is turned ON (enabled), positioning will be executed at acceleration and deceleration times according to the interpolation feed rate, as shown in the following diagram. When linear interpolation is executed for two or more axes, positioning will accelerate and decelerate at the interpolation acceleration and deceleration times (as shown in the above diagram) regardless of the interpolation feed rate and even if the Constant Acceleration Mode is turned ON (enabled).

In the following diagram, linear interpolation is executed for one axis only with the Constant Acceleration Mode turned ON.



Acceleration time Ta = Specified interpolation feed rate x interpolation acceleration time / maximum interpolation feed rate

Deceleration time Td = Specified interpolation feed rate x interpolation deceleration time / maximum interpolation feed rate

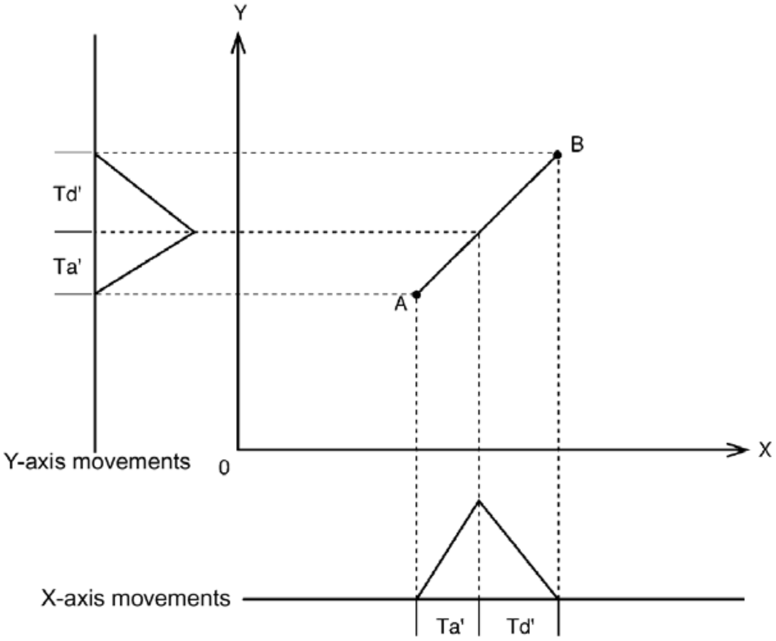
Note 1. The interpolation acceleration time and the interpolation deceleration time use the acceleration and deceleration times set in the system parameters or the times that are changed in the G-language program. (The times can be changed by G69.)

2. The maximum interpolation feed rate uses the value set in the system parameters.

Triangular Control

When the mode is set to the Stop Mode and if the travel time is shorter than the sum of the interpolation acceleration time and the deceleration time, triangular control will be performed, just as it is for PTP control.

When the mode is set to the Pass Mode, the travel time will not become shorter than the preset interpolation acceleration and deceleration times, but the speed will become slower than the designated feed rate.



The ratio between the interpolation acceleration time and the deceleration time for triangular control is equal to the ratio between the preset interpolation acceleration time (Ta) and the deceleration time (Td). Td'/Ta'=Td/Ta

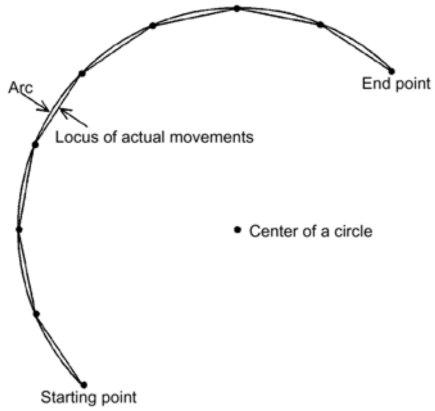
Circular Interpolation

Overview

Mode: Automatic; **Method:** G language (G02, G03)
Positioning for circular interpolation is performed using two axes from a starting point to an ending point and traveling through a circular arc.

Basic Concept

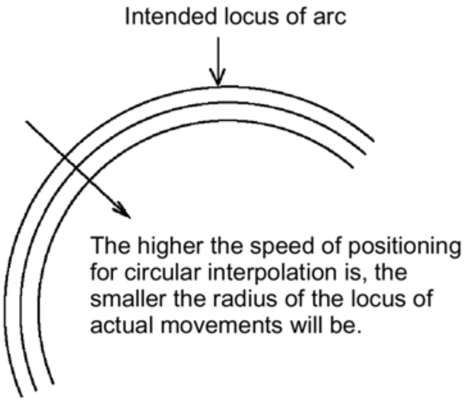
Circular interpolation is achieved by repeating successive linear interpolation along straight lines drawn by dividing a circular arc (in the shape of a polygon). The actual locus of circular interpolation is as shown below. The radius of the locus of actual movements is smaller than the radius of the arc due to accumulated pulses.



The interpolation acceleration time, the interpolation deceleration time, and triangular control are the same as for linear interpolation.

Refer to Linear Interpolation for details on interpolation acceleration time, interpolation deceleration time, and triangular control.

Note If positioning for circular interpolation is performed at high speed, the radius of the locus of actual movements will be reduced. This can also be improved by setting feed forward gain in the servo system. For details, refer to Changing Parameters.


Intended locus of arc
The higher the speed of positioning for circular interpolation is, the smaller the radius of the locus of actual movements will be.

Helical Circular Interpolation

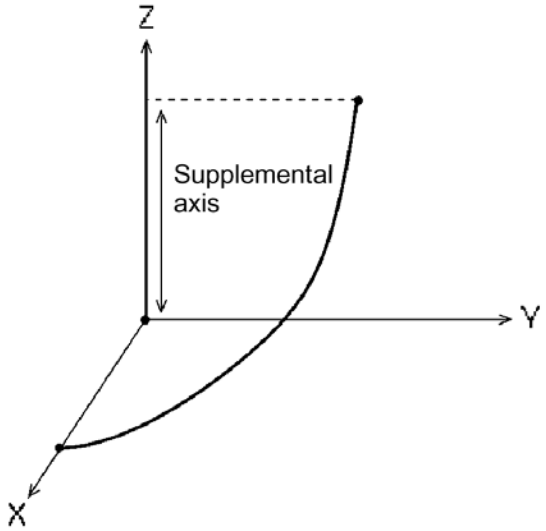
Overview

Mode: Automatic; **Method:** G language (G02, G03)
Helical circular interpolation adds another axis to circular interpolation. It cannot be used with CS1W-MC221 MC Units.

With earlier MC Unit models, helical circular interpolation was limited to a single turn, but the CS1W-MC421's multiturn capability makes helical operations easier.

Basic Concept

Similar to circular interpolation, helical circular interpolation performs control operations by dividing a 3-dimensional circular arc in the shape of a polygon. The helical circular interpolation locus is as shown below for circular interpolation executed for the X and Y axes and the Z-axis added as a supplemental axis.



The Z-axis feed rate is obtained by the following equation:

Feed rate = (Specified interpolation feed rate x Z-axis length)/(Arc length)
The interpolation acceleration time, the interpolation deceleration time, and triangular control are the same as for linear interpolation.

With helical circular interpolation, the travel time is calculated giving priority to the feed rate along the arc. For that reason, the feed rate of the supplemental axis may exceed the maximum interpolation feed rate set in the system parameters. If this occurs, a SUP AXIS SP OVER (supplemental axis speed overflow) error will be generated and operation will be stopped. Lower the feed rate if this error occurs.

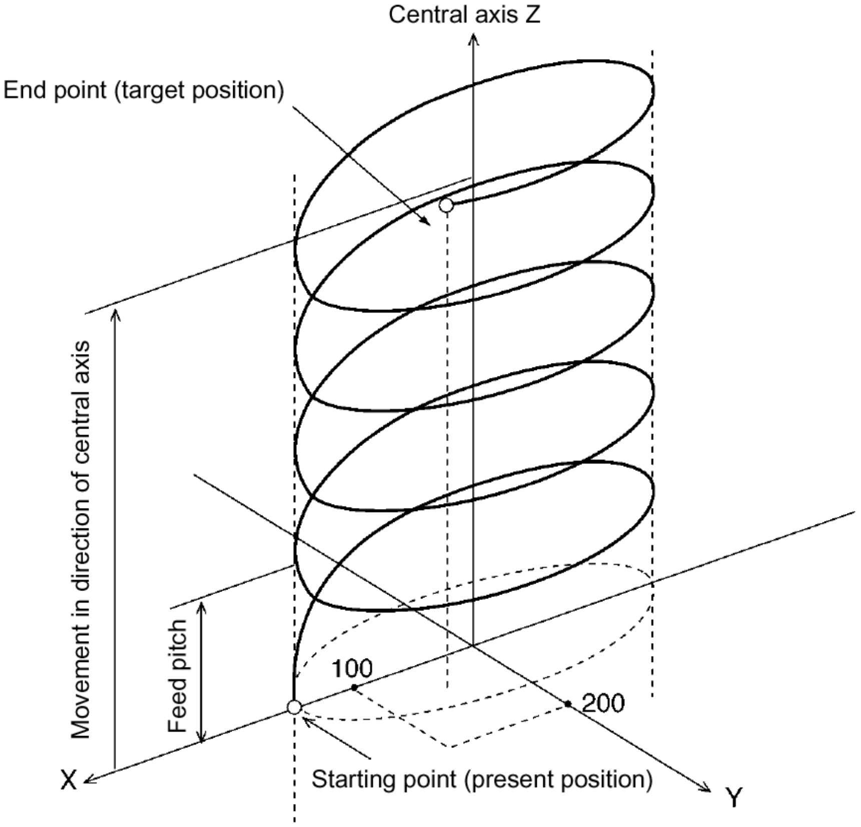
Helical Operation

```
N010    G90. . . . . . . . Absolute specification
N011    G17. . . . . . . . XY circular plane specification
N012    G02      X100   Y200   Z20    I0    J0    F200   L4
```

Number of turns (4 in this case)

Feed pitch for one turn in Z-axis direction

Stop position for 5th turn


Central axis Z
End point (target position)
Movement in direction of central axis
Feed pitch
100
200
X
Starting point (present position)
Y

Interrupt Feeding

Overview

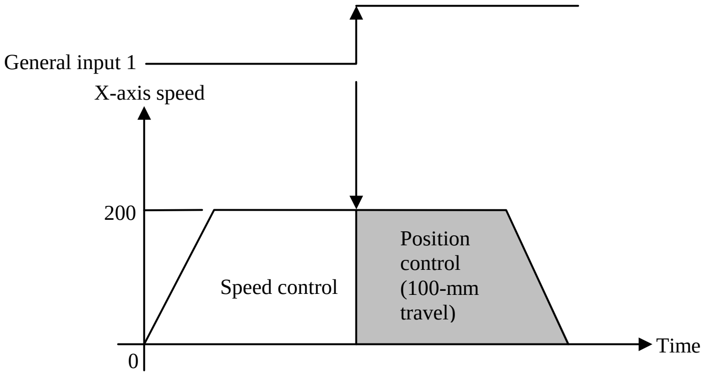
Mode: Automatic; **Method:** G language (G31)
Interrupt feeding is a control operation that feeds an axis for a fixed distance, using speed control until an external signal is input and then switching to position control at that point.

Interrupt feeding can be performed either at constant speed or two-stage speed.

It is also possible to specify a travel distance at which to stop when there is no interrupt input. Operation will normally stop with an error (program execution interrupted) after positioning, but an S option can be used to execute the next block without an error occurring.

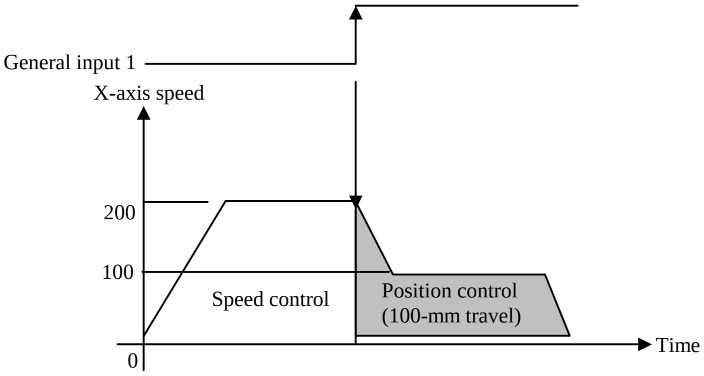
Constant Speed

In this example, only speed reference 1 is specified. The X axis is moved by 100 mm at 200 mm/s (speed reference 1) with speed control.

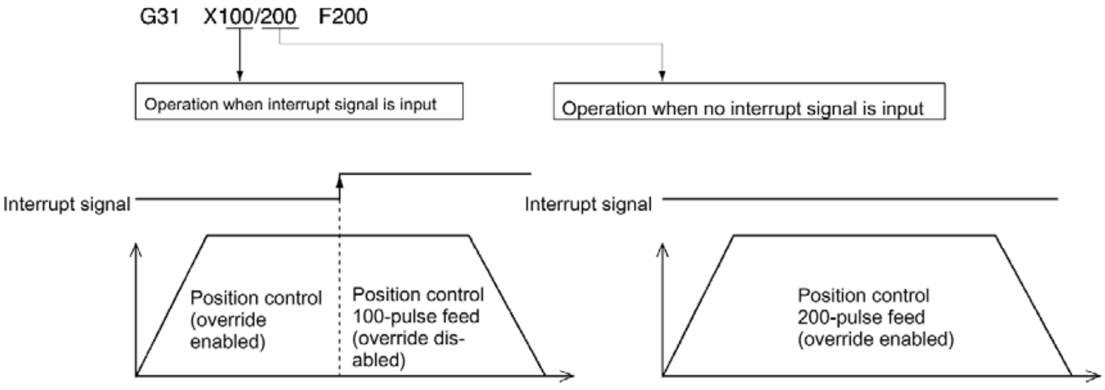


Two-stage Speed

In this example, speed reference 2 is specified. The X axis is moved by 100mm while the speed of the X axis is decelerated to 100mm/s from 200 mm/s.



Specifying an Amount of Travel for When No Interrupt Is Input


G31 X100/200 F200
Operation when interrupt signal is input
Operation when no interrupt signal is input
Interrupt signal
Position control (override enabled)
Position control 100-pulse feed (override dis-abled)
Interrupt signal
Position control 200-pulse feed (override enabled)

If no interrupt signal is received, operation will stop with an error (program execution interrupted) after positioning. If the S option is specified, however, the next block will be executed without an error occurring.

Example: G31 x 100/200 F200 S

Notes
1. Interrupt feeding can be used to move only one axis.
2. An external signal for interrupt feeding can be input via general inputs1 to 4 of the MC Unit.

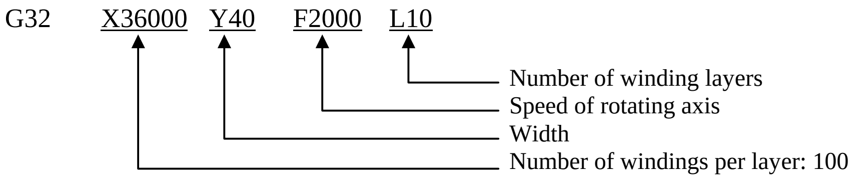
Input number	MC221	MC421
General input 1	For X-axis interrupt feeding	For X-axis interrupt feeding
General input 2	For Y-axis interrupt feeding	For Y-axis interrupt feeding
General input 3	---	For Z-axis interrupt feeding
General input 4	---	For U-axis interrupt feeding

Traverse Function

Overview

Mode: Automatic; **Method:** G language (G32)
The traverse function is a special G-language command (G32) that is provided for winding machine operations.

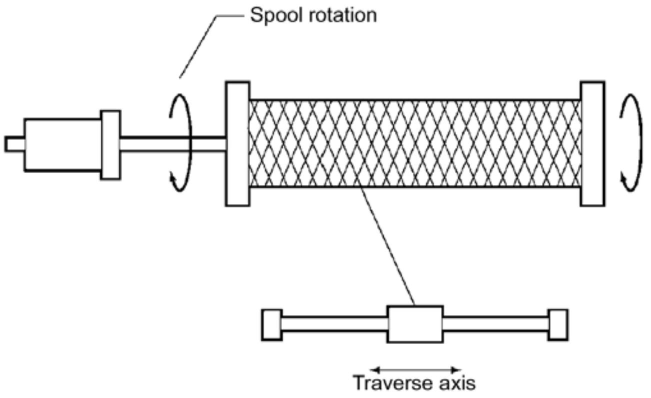
G Code Specifications



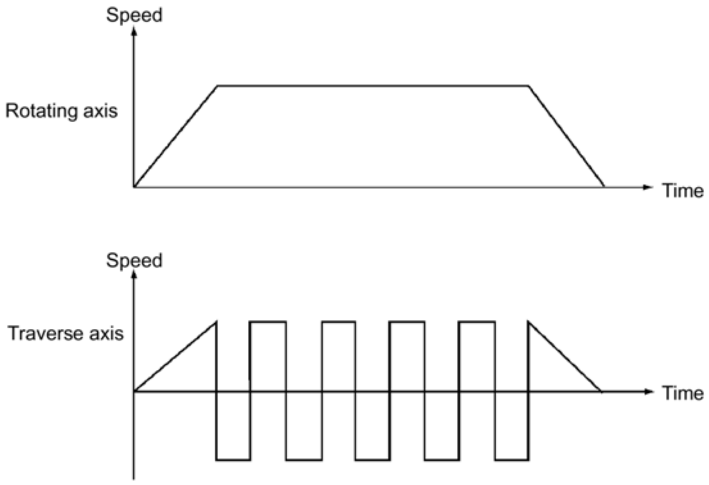
The setting for 100 windings per layer is 36000 because one winding is equivalent to 360°.

Example Application

In this example, the wire is wound around a spool on a winding machine. The spool is turned in a fixed direction and the traverse axis is operated according to the settings for the traverse function.



Speeds for Each Axis during Traverse Operation



Operating Modes

Overview

Mode: Automatic; **Method:** G language (G10, G11, G13), system parameters
There are three operating modes for positioning: Stop Mode (G11), Pass Mode (G10), and In-position Check OFF Mode (G13).

Operating mode	Function
Stop Mode	When continuous operations are specified, the Stop Mode determines when positioning has been completed and then starts the next operation.
Pass Mode	When continuous operations are specified, the Pass Mode proceeds smoothly to the next operation without determining whether the present positioning has been completed.
In-position Check OFF Mode	When positioning is executed, the In-position Check OFF Mode starts the next operation without determining whether positioning has been completed. This is the only difference between this mode and the Stop Mode.

The operating mode can be changed by the operations and G codes shown in the following table. Once the operating mode has been changed, the new mode will remain in effect until it is again changed by one of these methods.

Methods for Changing the Operating Mode

Name	G code	Operation mode
Power supply reset or restarting the MC Unit as a Special I/O Unit	None	Pass Mode
Executing the main program from the beginning	None	Pass Mode
ORIGIN SEARCH	G28	Pass Mode
REFERENCE ORIGIN RETURN	G26	Pass Mode
WORKPIECE ORIGIN RETURN	G27	Pass Mode
PASS MODE	G10	Pass Mode
STOP MODE	G11	Stop Mode
IN-POSITION CHECK OFF MODE	G13	In-position Check OFF Mode

Among those G codes that execute positioning actions, there are some for which any of the three operating modes can be selected and some for which positioning is always executed in Stop Mode. These G code operations are shown in the following table

G code	Name	**Operating mode**		
		Stop Mode	**Pass Mode**	**In-position Check OFF Mode**
G00	POSITIONING	●	---	---
G01	LINEAR INTERPOLATION	●	●	●
G02	CIRCULAR INTERPOLATION (CLOCKWISE)	●	●	●
G03	CIRCULAR INTERPOLATION (COUNTERCLOCKWISE)	●	●	●
G26	REFERENCE ORIGIN RETURN	●	---	---
G27	WORKPIECE ORIGIN RETURN	●	---	---
G28	ORIGIN SEARCH	●	---	---
G30	SPEED CONTROL	●	---	---
G31	INTERRUPT INCHING	●	---	---
G32	TRAVERSE	●	●	●

●: Modes that can be specified.

---: Positioning is always executed in Stop Mode regardless of which operating mode is specified.

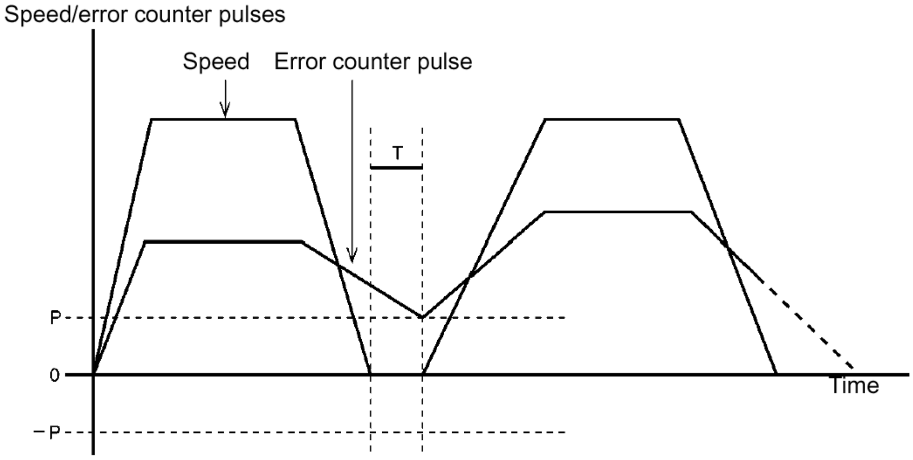
1 Stop Mode

If the Stop Mode is specified for continuous positioning operations, one operation is followed by the subsequent operation after the first positioning operation has been completed.

The following illustration shows changes in the time, speed, and pulses when the axis is controlled as shown below.

Example:

```
N010 G00 X100        Moves X axis to 100-mm point by PTP control
N011 G00 X200        Moves X axis to 200-mm point by PTP control
```



P: In-position width
T: Time required for determining whether positioning is completed or not

Positioning is completed when the error counter pulse falls within the in-position zone. By using the In-position Check OFF function, it is also possible to start the next operation without waiting for positioning to be completed. In this case, the operation is executed within 4ms.

2 Pass Mode

If the Pass Mode is specified for continuous positioning operations, one operation is followed by the subsequent operation without any pause in motion, i.e., without stopping to determine whether positioning is completed or not. When the power is turned ON, this mode is automatically selected.

The following illustrations show changes in the speed in the Stop Mode and the Pass Mode when the continuous operation commands are given as shown be-low.

The Pass Time Mode Selection in the Unit parameters can be used to select, when Pass Mode is used, whether positioning will move to the next operation in the interpolation acceleration time or the interpolation deceleration time of the immediately preceding operation. Only in case of linear interpolation for one axis, acceleration during acceleration or pass operation may be set to a constant speed by selecting the Constant Acceleration Mode.

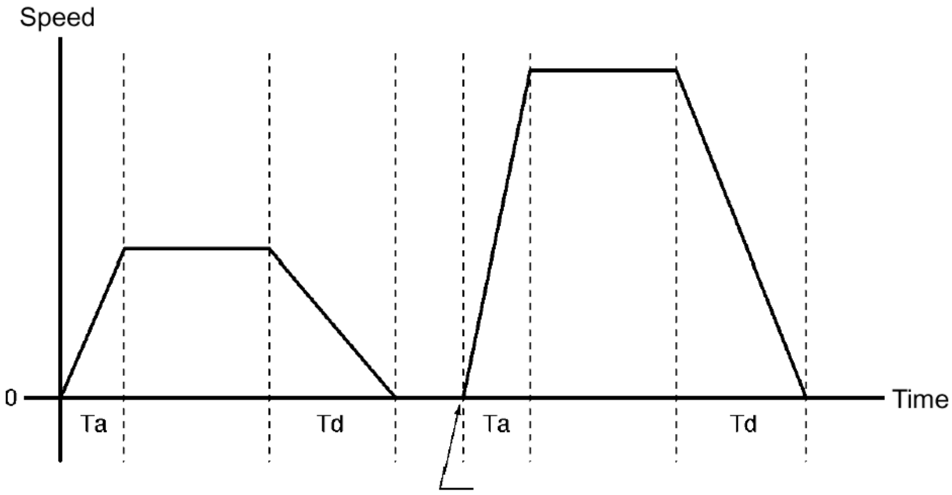
When the Constant Acceleration Mode is enabled, this mode will be given priority for single-axis pass operations regardless of the setting for the Pass Time Mode Selection.

The settings for the Pass Time Mode Selection and the Constant Acceleration Mode are made using CX-Motion. The contents of these settings can only be read from the ladder program using the IORD instruction.

Example:

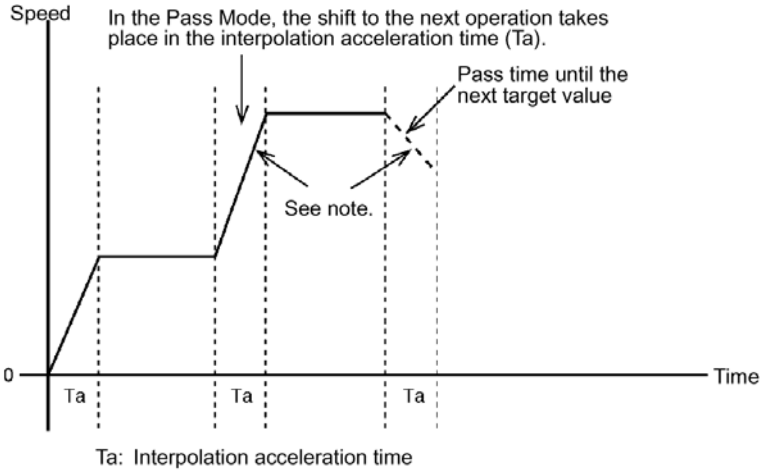
```
      N010 G01 X100 F10
            Moves X axis to 100-mm point at 10 mm/s via linear interpolation
      N011 G01 X300 F20
            Moves X axis to 300-mm point at 20 mm/s via linear interpolation
            ⋮
```

In Stop Mode



In Pass Mode with Acceleration Time

In this example, the Pass Time Mode Selection is set for interpolation acceleration time. Positioning moves to the next operation in the interpolation acceleration time of the immediately preceding operation.

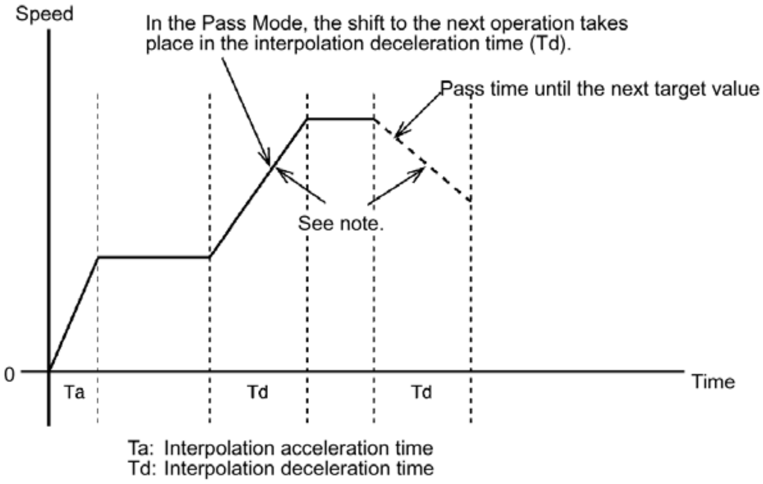


Note: The acceleration/deceleration curve will be trapezoidal even if an S-curve is specified.

When the Pass Mode is specified as the operating mode, the time required for movements is reduced because there is no need to determine whether positioning has been completed.

In Pass Mode with Deceleration Time

In this example, the Pass Time Mode Selection is set for interpolation deceleration time. Positioning moves to the next operation in the interpolation deceleration time of the immediately preceding operation.



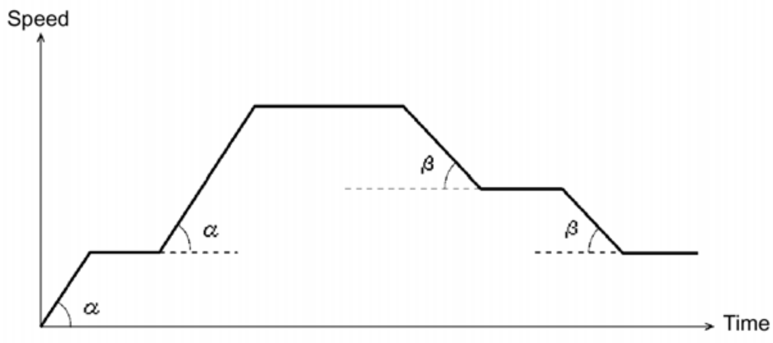
Note: The acceleration/deceleration curve will be trapezoidal even if an S-curve is specified.

When the Pass Mode is selected, the time required for movements is reduced because there is no need to determine whether positioning has been completed.

In Constant Acceleration Mode

The Constant Acceleration Mode is enabled when linear interpolation is executed for one axis only. To use this mode, turn ON the Constant Acceleration Mode setting in the Unit parameters.

If a pass operation is executed in the Constant Acceleration Mode, the acceleration and deceleration times will always be constant as shown in the following diagram.



Acceleration α = Max. interpolation feed rate / interpolation acceleration time
Deceleration β = Max. interpolation feed rate / interpolation deceleration time

Note: 1. The interpolation acceleration time and the interpolation deceleration time use the acceleration and deceleration times set in the system parameters or the times that are changed in the G-language program. (The times can be changed by G69.)

2. The acceleration and deceleration times when pass operations are executed increase or decrease according to the amount of change in the interpolation feed rate. The times can be determined by the following equations:

Acceleration time = Change in interpolation feed rate x Interpolation acceleration time / Maximum interpolation feed rate

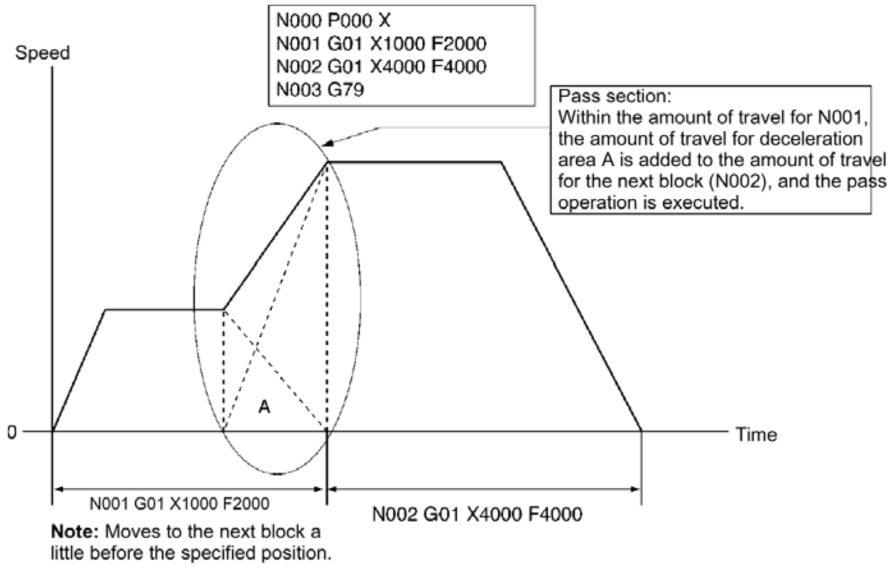
Deceleration time = Change in interpolation feed rate x Interpolation deceleration time / Maximum interpolation feed rate

Block Ends in Pass Mode

In Pass Mode, the ends of G-language blocks are treated as shown below.

When Constant Acceleration Mode Is OFF

Pass operations are executed as shown in the following diagram when the Constant Acceleration Mode is OFF.



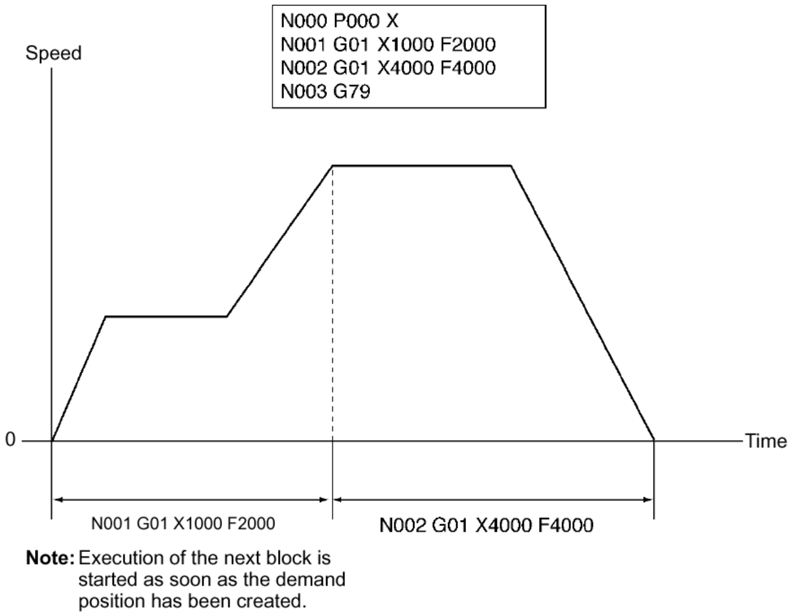
```
N000 P000 X
N001 G01 X1000 F2000
N002 G01 X4000 F4000
N003 G79
```

Pass section:
Within the amount of travel for N001, the amount of travel for deceleration area A is added to the amount of travel for the next block (N002), and the pass operation is executed.

Note: Moves to the next block a little before the specified position.

When Constant Acceleration Mode Is ON

Linear interpolation pass operations are executed as shown in the following diagram when the Constant Acceleration Mode is ON. The next block is executed after the demand position is created.



3 In-position Check OFF Mode

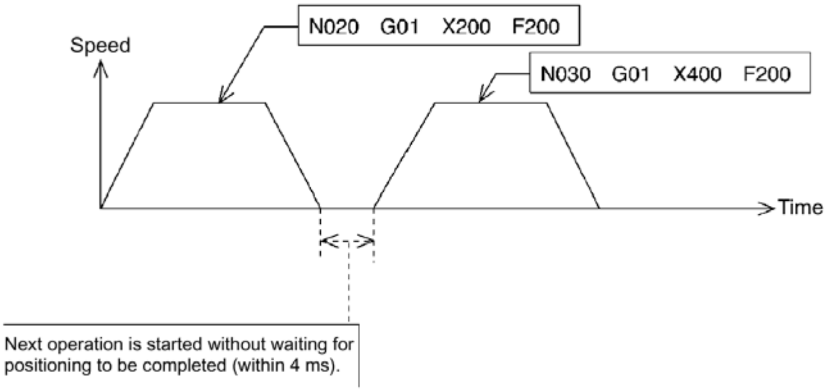
By specifying the G13 code (IN-POSITION CHECK OFF MODE), the next positioning operation can be started without waiting for the present one to be completed. This allows high-speed pick-and-place operations to be performed.

This function contrasts with the Stop Mode, in which the next positioning operation is not started until the present one has been completed.

To have positioning wait until the present operation is completed before the next one is started, execute G11 (STOP MODE).

When In-position Check OFF Mode Is Enabled

```
N010    G13 . . . . . . . . . . . . . . . . . . . . . In-position Check OFF statement
N020    G01     X200    F200
N030    G01     X400    F200
```



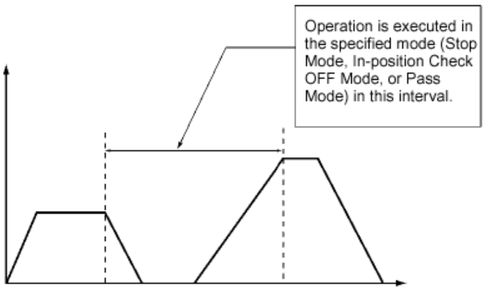
Constant Acceleration Mode Limitations

The Constant Acceleration Mode can be used only for single-axis linear interpolation. When two or more axes are operated following a linear interpolation operation in the Constant Acceleration Mode, the next operation will be executed after positioning has been completed regardless of the operating mode set-ting.

The following diagrams provide four examples of operations in the Constant Acceleration Mode.

Example 1: Operation with One Axis Only

```
G01 X100 F200
G01 X200 F400
```

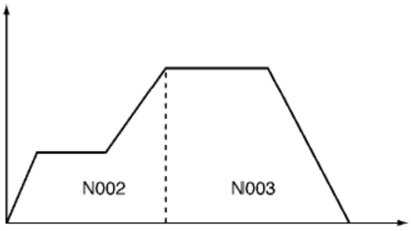


Operation is executed in the specified mode (Stop Mode, In-position Check OFF Mode, or Pass Mode) in this interval.

Example 2: Pass Operation with One Axis Only

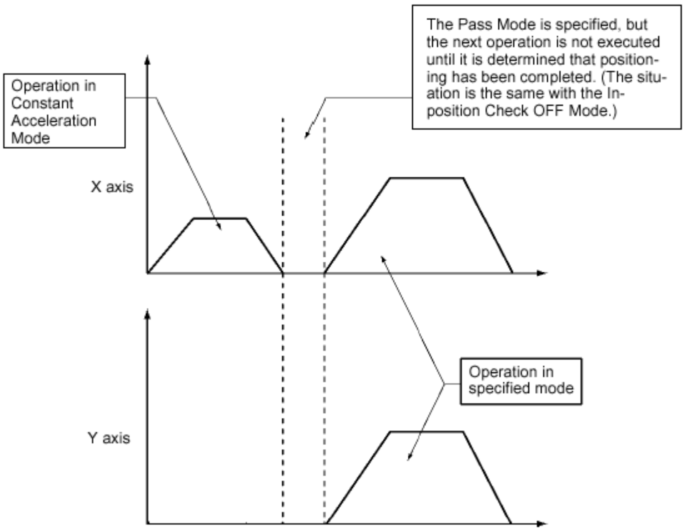
```
N0000 P000 XY
N001 G91
N002 G01 X100 F200
N003 G01 Y0 F400F400
```

Pass operation is executed and the Y axis is not oper-ated even if it is specified. (In this case the amount of relative movement is 0.)



Example 3: Operating from One Axis to Two Axes

```
N000 P000 XY
N001 G01 X100 F200
N002 G01 X200 Y300 X500 F300
```



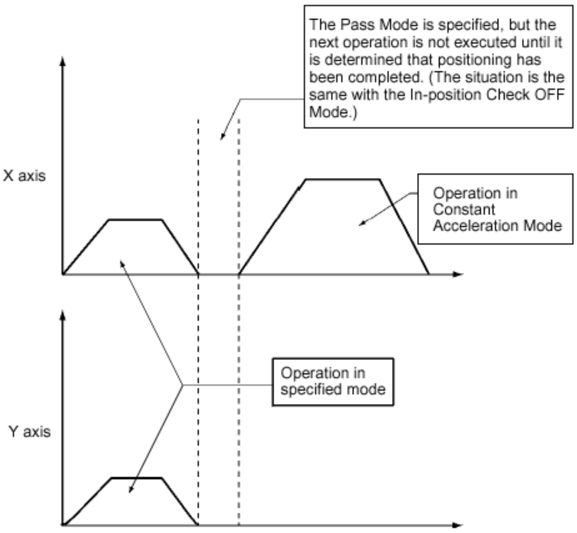
Operation in Constant Acceleration Mode

The Pass Mode is specified, but the next operation is not executed until it is determined that positioning has been completed. (The situation is the same with the In-position Check OFF Mode.)

Operation in specified mode

Example 4: Operating from One Axis to Two Axes

```
N000 P000 XY
N001 G01 X100 Y100 F200
N002 G01 X300 F300
```



The Pass Mode is specified, but the next operation is not executed until it is determined that positioning has been completed. (The situation is the same with the In-position Check OFF Mode.)

Operation in Constant Acceleration Mode

Operation in specified mode

Pass Mode Limitations with Linear Interpolation, Circular Interpolation, and Traverse Operations

Linear interpolation, circular interpolation, and traverse operations are interpolation commands that can be specified for three operating modes. When these interpolation commands are continuously executed in Pass Mode, there are cases where pass operations cannot be performed (depending on the combination of the immediately preceding operation and the present operation). These cases are shown in the following table.

The preceding and present operations for the table shown below are as follows:
Preceding operation: N010 G01 X1000 Y100 F200
Present operation: N020 G01 Y2000 Y-100 F300

Pass: Indicates normal Pass Mode operations. Positioning is executed with pass operations from the preceding operation to the present operation.

Stop: Indicates normal Stop Mode operations. The present operation is executed after the positioning is completed for the preceding operation.

Preceding operation	**Present operation**				
	Linear interpolation	**Circular interpolation**		**Traverse**	
		Same plane	**Different planes**	**Same axis**	**Different axes**[1]
Linear interpolation	Pass	Pass	Pass	Pass	Pass
Circular interpolation	Pass	Pass	Stop[2]	Stop[3]	Stop[3]
Traverse	Pass	Stop[3]	Stop[3]	Stop or Pass[4]	Stop[5]

Note 1. "Different axes" indicates that the axes that are operated are changed, as when four axes are used for one task and linear interpolation is executed for the X and Y axes followed by traverse operation for the Z and U axes.
2. This indicates cases such as executing ZU-plane circular interpolation after XY-plane circular interpolation.
3. This indicates cases such as executing traverse operations after circular interpolation, or circular interpolation after traverse operations.
4. The operating mode for the traverse operation will differ according to the following circumstances.
   If the rotating axes are operated in opposite directions, the preceding traverse operation will be performed in Stop Mode.
   If the rotating axes are operated in the same direction, positioning will be executed in Pass Mode.
5. When four axes are used for one task, the traverse operation is executed first for the X and Y axes and then for the Z and U axes.

Other Limitation

In Pass Mode and In-position Check OFF Mode, the G-language program is read in advance. In addition, the pre-reading stops in the following situations, so the operations are always executed as if in Stop Mode, regardless of the actual operating mode.

Commands where operation stops following command	**Commands where previous interpolation operation stops (G01, G02, G03, G32)**
G01, G02, G03, including M000 to M499	G00 (PTP)
	G01, G02, G03, including # option number specification
	G04 (DWELL TIMER)
	G29 (ORIGIN UNDEFINED)
	G30 (SPEED CONTROL)
	G31 (INTERRUPT FEEDING)
	G54 (CHANGE REFERENCE COORDINATE SYSTEM PV)
	G79 (PROGRAM END)
	Independent commands M000 to M999
	Independent commands D000 to D255

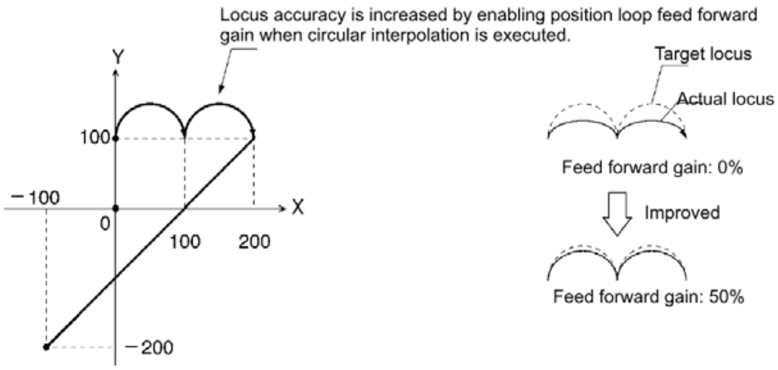
Changing Parameters

Overview

Mode: Automatic; **Method:** G language (G69)
The servo system gain can be changed by using CHANGE PARAMETER (G69).
This function can be used, for example, to increase the accuracy of circular interpolation when position loop feed forward gain is enabled during circular interpolation.

The following parameters can be changed using this function:
Changeable Servo Parameters
#5 Position loop gain
#6 Position loop feed forward gain
#7 In-position
#8 Accumulated pulse warning value
Changeable Feed Rate Parameters
#1 Acceleration time
#2 Deceleration time
#3 Interpolation acceleration time
#4 Interpolation deceleration time
The changed servo parameters and feed rate parameters go into effect immediately after G69 is executed.

```
N010   G01    X0      Y100   F100
N020   G69    #6/50 . . . . . . . . . . . . . . . . . . . . . . . . .Sets the position loop feed forward gain to 50%.
N030   G02    X100    Y100   I50    J0     F200 ⎫  Increases locus accuracy by enabling position loop feed forward
N040   G02    X200    Y100   I50    J0     F300 ⎭  gain when circular interpolation is executed.
N050   G69    #5/X0 . . . . . . . . . . . . . . . . . . . . . . . . Returns the position loop feed forward gain to 0%.
N060   G01    X–100   Y–200  F100
```


Locus accuracy is increased by enabling position loop feed forward gain when circular interpolation is executed.
Y
100
−100
0
X
100
200
−200
Target locus
Actual locus
Feed forward gain: 0%
Improved
Feed forward gain: 50%

Stopover Function

Overview

Mode: Automatic; **Method:** G language (G00, G01, etc.)
The stopover function outputs an M code or D code, without stopping axis operation, when the axis has traveled for a specified distance (determined by encoder feedback present position). Before the operation is completed, tact time can be improved by using this function to control peripheral devices.
This function can be applied to G codes for all operations. Either D codes or M codes can be used, but not both.

Case 1: N010 G01 X100 F200 D100/X50
In this case, D code 100 is output when the X-axis is moved by 50 mm in the direction of the specified position.

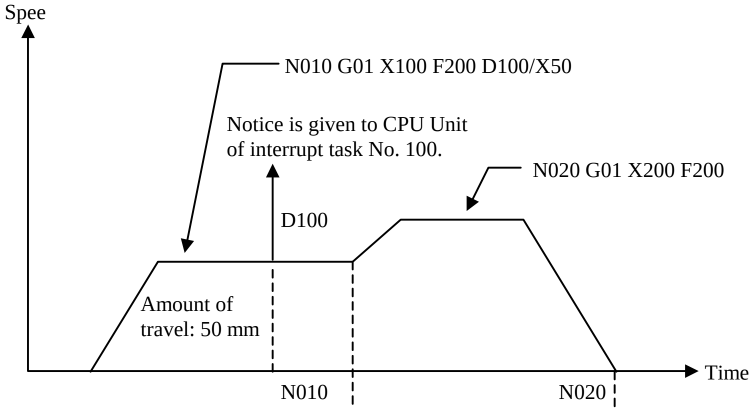
Case 2: N010 G01 X200 F200 M600/X100
In this case, M code 600 is output when the X-axis is moved by 100 mm in the direction of the specified position.

Case 3: N010 G01 X100 F200 M700 D30

In this case, D30 and M700 are output when block N010 is completed (i.e., when the X axis is moved by 200 mm).

Wrong: N010 G01 X100 F200 M800 D100/X50
If the stopover function is used in this way, an error will be generated by CX-Motion.

Operation Example


Spee
N010 G01 X100 F200 D100/X50
Notice is given to CPU Unit of interrupt task No. 100.
N020 G01 X200 F200
D100
Amount of travel: 50 mm
Time
N010
N020

After the D code or M code, place a "/" followed by the axis name and the amount of travel.
Axis name: Specify X, Y, Z, or U.
Amount of travel: Specify the amount of travel, without sign, in the direction of the demand position (determined by the present position). The amount of travel can be written as follows:

- Number (0 to 39,999,999)
- A0000 to A1999
- (E00) to (E31)

CPU Unit Interrupt Processing

Overview

Mode: Automatic; **Method:** G language (D code output)
It is possible to set a D code in the G-language program related to positioning as an interrupt to the CPU Unit. The specified D code is output after positioning is completed, and the CPU Unit executes an interrupt task taking the D code (0 to 255) as the task number.

When the stopover function is used, this D code can be output during operation without stopping the operation. For details on the stopover function, refer to Stopover Function.

Note This interrupt function can be used only when the MC Unit is mounted to the CPU Backplane. It cannot be used if the MC Unit is mounted to an Expansion I/O Rack.

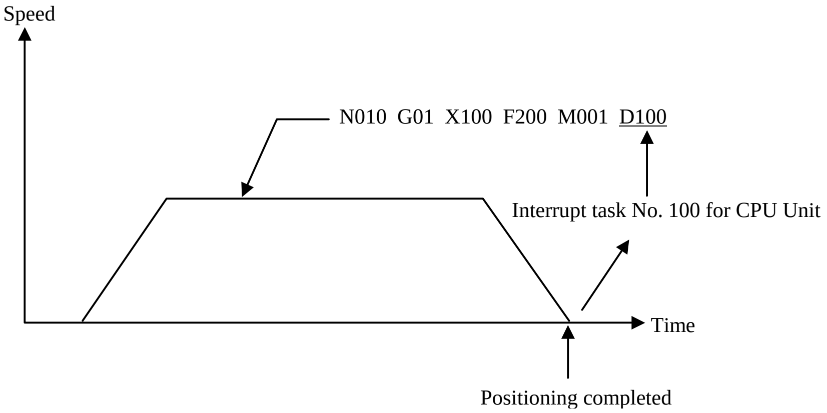
When an interrupt task is used in the CPU Unit, a setting must be made in the PC Setup. For details, refer to the manual for the CPU Unit.

Case 1: N010 D100

Case 2: N010 G01 X100 F200 D100

Case 3: N010 G01 X100 F100 M001 D100

Case 3 Operation


Speed
N010 G01 X100 F200 M001 D100
Interrupt task No. 100 for CPU Unit
Time
Positioning completed

Override Function

Overview

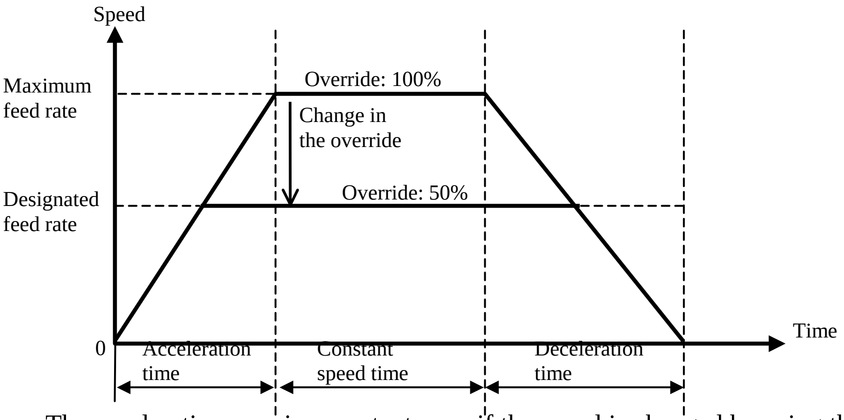
Mode: Automatic or Manual; **Method:** Bit designation

The override function is designed to change the operation speed by multiplying the speed that is set in the system parameters or G commands by a specified factor. Setting methods are different depending on whether the PTP control is used or interpolation is used.

The override function can be used for the following G codes:
G00: PTP CONTROL POSITIONING
G01: LINEAR INTERPOLATION
G02/G03: CIRCULAR INTERPOLATION
G26: REFERENCE ORIGIN RETURN
G27: WORKPIECE ORIGIN RETURN
G28: ORIGIN SEARCH (but only for movement for an origin offset)
G30: SPEED CONTROL
G31: INTERRUPT FEEDING
G32: TRAVERSE

Override in PTP Control

The override in the PTP control can be set to a value between 0.1% and 100.0%. “100%” means the maximum feed rate designated by the feed rate parameter. The override function is used when the maximum feed rate is too fast. For example, if the override is set to 50%, the travel speed drops to half the maximum feed rate.

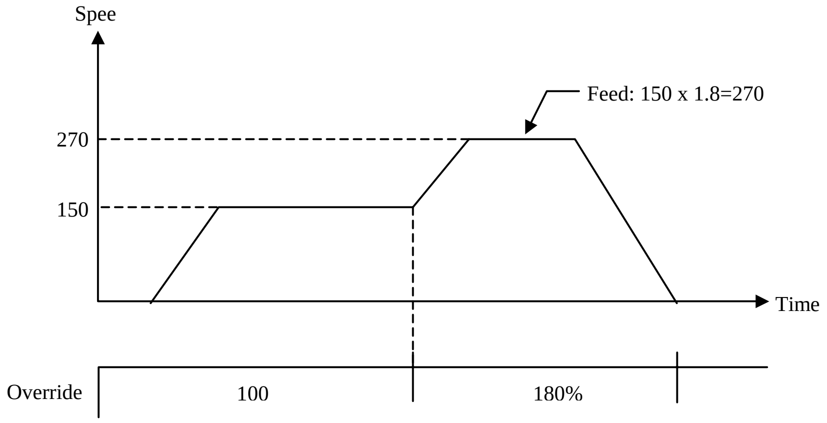


The acceleration remains constant even if the speed is changed by using the override function. As a result, both the acceleration time and the deceleration time are proportionally reduced.

Override in CP Control

In interpolation control, the designated interpolation feed rate is used as the maximum feed rate. The override for the interpolation feed rate can be set to a value between 0.1% and 199.9%. If the result set by using the override function to over 100% exceeds the maximum interpolation feed rate, the maximum interpolation feed rate is used instead.

Example: Execution in Stop Mode



Note With linear or circular interpolation in either the Pass Mode or the In-position Check OFF Mode, the speed cannot be changed in real time during operation.

Setting an Override

To execute an override, set the override in the PC Interface Area and turn ON the Override Setting Bit. The override will then be applied to subsequent operations.

PC Interface Area

Name	MCU Model	Word				Bits	Description
		X axis	Y axis	Z axis	U axis		
Override	MC421 MC221	n+10 n+6	n+12 n+8	n+14 ---	n+16 ---	00 to 15	Override(4 digits Hex) Setting range: 0000 to 07CF Hex (0000 to 1999: 0 to 199.9%)
Override setting	MC421 MC221	n+11 n+7	n+13 n+9	n+15 ---	n+17 ---	12	1: Override enabled 2: Override disabled

Even if an override is set for a given axis, it will be disabled if the Override Setting Bit is turned OFF.

Resetting the Error Counter

Overview

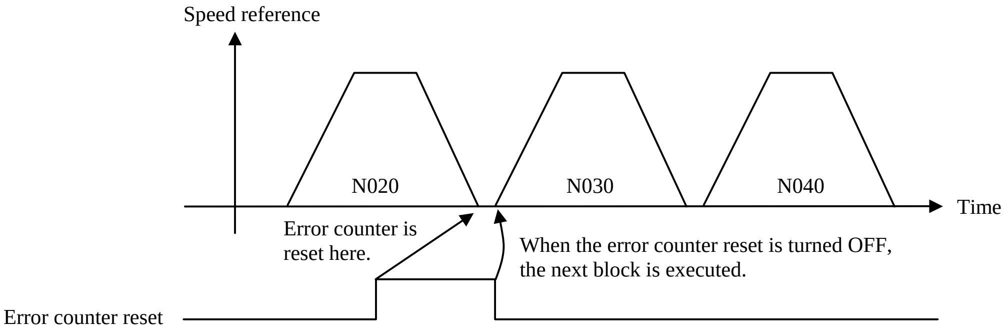
Mode: Automatic or Manual; **Method:** Bit designation

The error counter reset function resets the error counter to 0 and stops axis operation. This function is enabled when no speed reference is being provided to the Servo Driver (after an deceleration command is output). It can be used for molding applications (machine press control).

This function can be used with the MC Unit in either Automatic or Manual Mode, and it is executed from the PC Interface Area. In the examples shown here, the Automatic Mode is used.

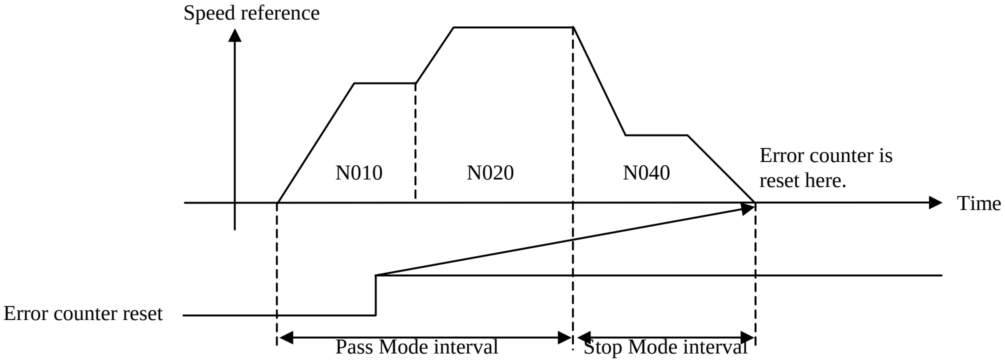
Stop Mode

N000 P000 X
N010 G11 ← Specifies the Stop Mode.
N020 G01 X100 F100
tN030 G01 X200 F100
N040 G01 X300 F100


Speed reference
N020
N030
N040
Time
Error counter is reset here.
When the error counter reset is turned OFF, the next block is executed.
Error counter reset

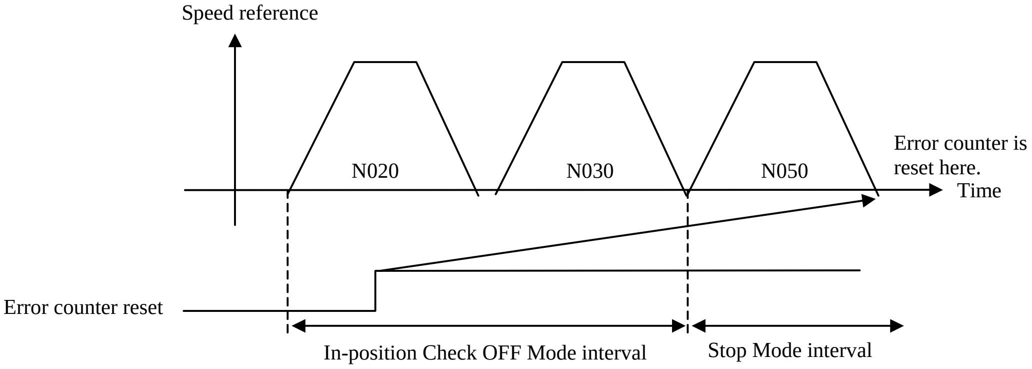
Pass Mode

N000 P000 X
N010 G01 X100 F100 ← Initial value: Pass Mode
N020 G01 X400 F150
N030 G11 ← Specifies the Pass Mode.
N040 G01 X100 F50



In-position Check OFF Mode

N000 P000 X
N010 G13 ← Specifies the In-position Check OFF Mode.
N020 G01 X100 F100
N030 G01 X200 F100
N040 G11 ← Stop Mode designation
N050 G01 X300 F100

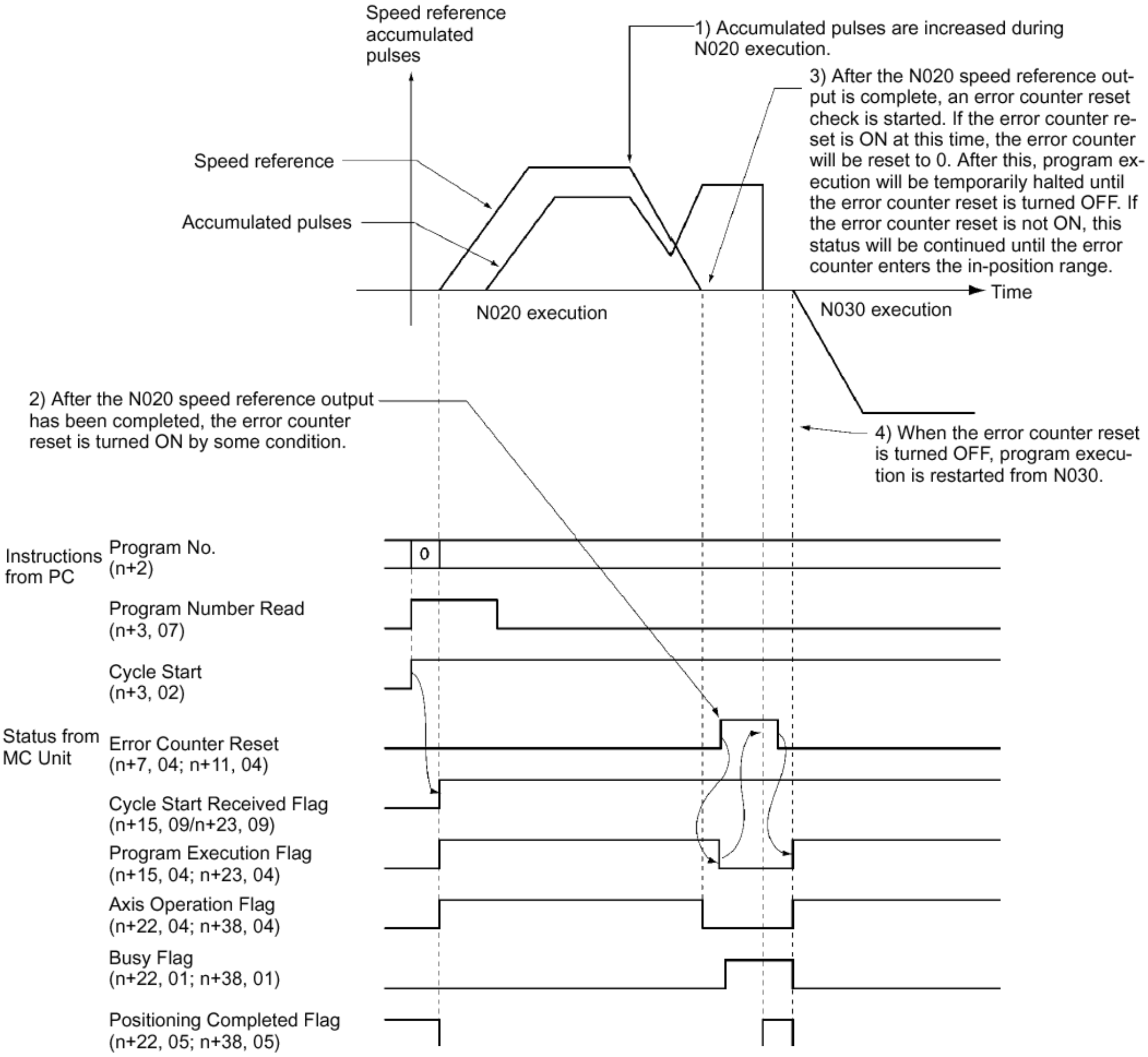


Notes 1. The program cannot be executed while the error counter is ON.

2. The error counter reset can be used when jogging, origin search, and other operations are executed in Manual Mode.

In the following example, linear interpolation is executed in Stop Mode on the X axis. The operation and timing would be the same for either linear or circular interpolation using two or more axes.

```
N000 P000 X
N010 G11 ← Specifies the Stop Mode.
N020 G01 X100 F100
N030 G01 X-100 F100
...
```


Speed reference accumulated pulses
1) Accumulated pulses are increased during N020 execution.
3) After the N020 speed reference output is complete, an error counter reset check is started. If the error counter reset is ON at this time, the error counter will be reset to 0. After this, program execution will be temporarily halted until the error counter reset is turned OFF. If the error counter reset is not ON, this status will be continued until the error counter enters the in-position range.
Speed reference
Accumulated pulses
Time
N020 execution
N030 execution
2) After the N020 speed reference output has been completed, the error counter reset is turned ON by some condition.
4) When the error counter reset is turned OFF, program execution is restarted from N030.
Instructions from PC
Program No. (n+2)
0
Program Number Read (n+3, 07)
Cycle Start (n+3, 02)
Status from MC Unit
Error Counter Reset (n+7, 04; n+11, 04)
Cycle Start Received Flag (n+15, 09/n+23, 09)
Program Execution Flag (n+15, 04; n+23, 04)
Axis Operation Flag (n+22, 04; n+38, 04)
Busy Flag (n+22, 01; n+38, 01)
Positioning Completed Flag (n+22, 05; n+38, 05)

PC Interface Area

Name	MCU Model	Word				Bits	Description
		X axis	Y axis	Z axis	U axis		
Error Counter Reset	MC421 MC221	n+11 n+7	n+13 n+9	n+15 ---	n+17 ---	04	0 to1: Error counter reset 2: Functions can be received.

Servo Lock and Unlock

Overview

Mode: Automatic (servo lock); Automatic or Manual (servo unlock);
Method: Bit designation
This function creates (servo lock) or releases (servo unlock) an MC Unit position loop. In addition to the servo lock and unlock operations available for earlier MC Unit models, it is also possible to set general outputs as brake signal outputs and to turn them ON and OFF in sync with servo lock and unlock. The ON and OFF times can also be set as required.

If the machine parameters are set for a wiring check to be performed, it will be performed at power up and with the first servo lock. If everything is normal, the wiring check will not be performed with the next servo lock. Servo lock and unlock are executed from the PC Interface Area.

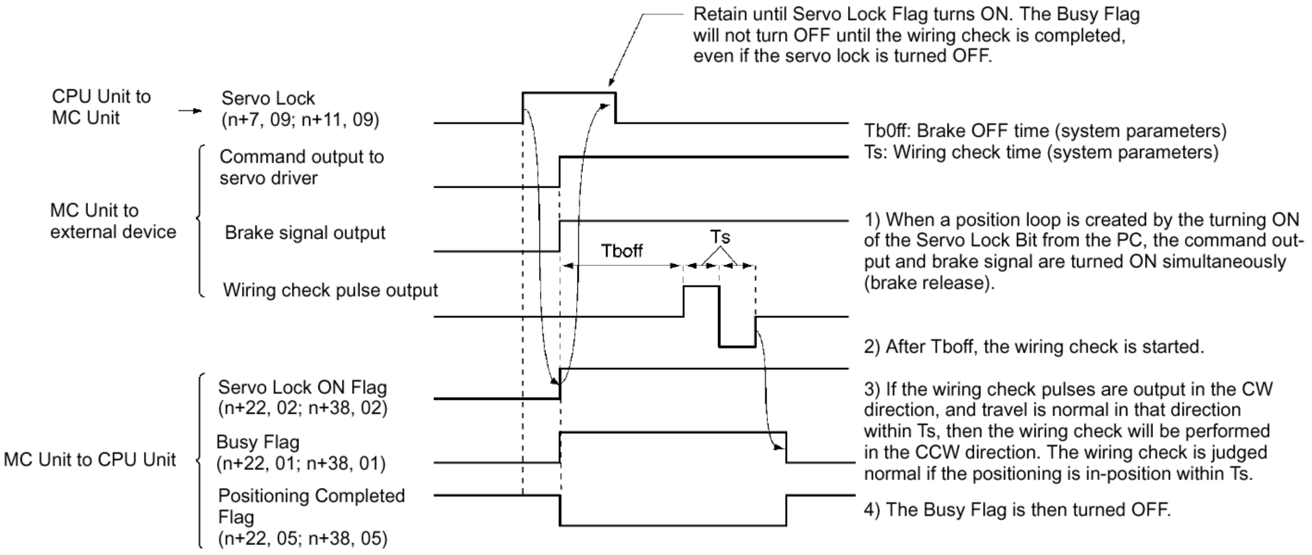
PC Interface Area

Name	MCU Model	Word				Bits	Description
		X axis	Y axis	Z axis	U axis		
Servo Lock	MC421	n+11	n+13	n+15	n+17	09	↑ : Begin servo lock
Servo Unlock	MC221	n+7	n+9	---	---	10	↓ : Begin servo unlock

1 Timing Charts

The following timing charts show the servo lock and servo unlock functions when the brake signal is set and the wiring check is enabled.

Servo Lock


CPU Unit to MC Unit
Servo Lock (n+7, 09; n+11, 09)
MC Unit to external device
Command output to servo driver
Brake signal output
Wiring check pulse output
Tboff
Ts
MC Unit to CPU Unit
Servo Lock ON Flag (n+22, 02; n+38, 02)
Busy Flag (n+22, 01; n+38, 01)
Positioning Completed Flag (n+22, 05; n+38, 05)

Retain until Servo Lock Flag turns ON. The Busy Flag will not turn OFF until the wiring check is completed, even if the servo lock is turned OFF.

Tb0ff: Brake OFF time (system parameters)
Ts: Wiring check time (system parameters)

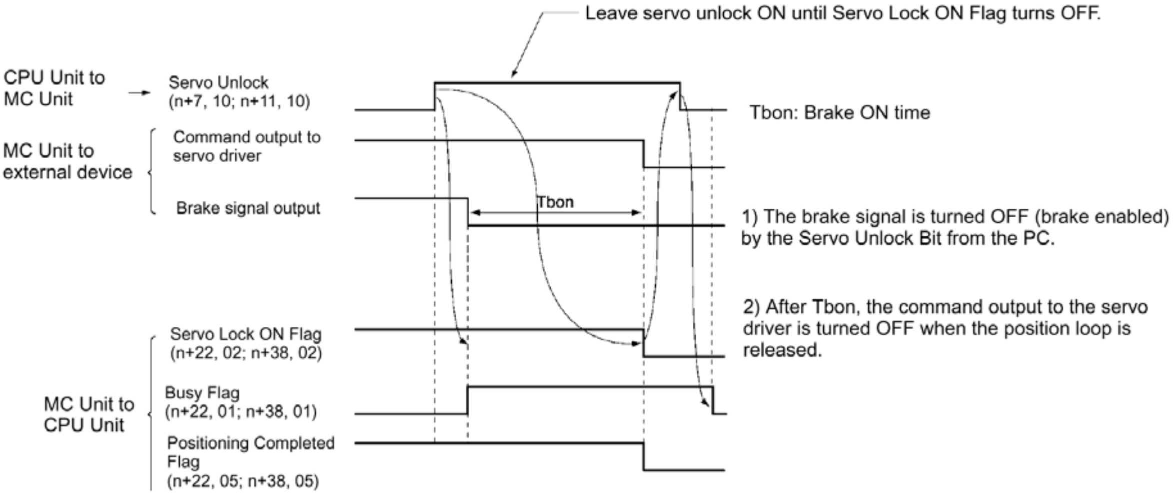
1) When a position loop is created by the turning ON of the Servo Lock Bit from the PC, the command output and brake signal are turned ON simultaneously (brake release).

2) After Tboff, the wiring check is started.

3) If the wiring check pulses are output in the CW direction, and travel is normal in that direction within Ts, then the wiring check will be performed in the CCW direction. The wiring check is judged normal if the positioning is in-position within Ts.

4) The Busy Flag is then turned OFF.

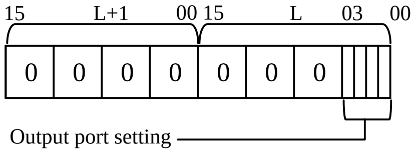
Servo Unlock


Leave servo unlock ON until Servo Lock ON Flag turns OFF.
CPU Unit to MC Unit
Servo Unlock (n+7, 10; n+11, 10)
Tbon: Brake ON time
MC Unit to external device
Command output to servo driver
Brake signal output
Tbon
1) The brake signal is turned OFF (brake enabled) by the Servo Unlock Bit from the PC.
2) After Tbon, the command output to the servo driver is turned OFF when the position loop is released.
Servo Lock ON Flag (n+22, 02; n+38, 02)
MC Unit to CPU Unit
Busy Flag (n+22, 01; n+38, 01)
Positioning Completed Flag (n+22, 05; n+38, 05)

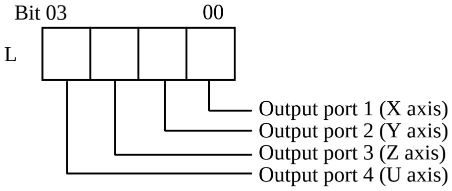
2 Related System Parameters

The system parameters required for the output port settings and the wiring check are all set using CX-Motion. The IORD instruction can be used for reading only. Refer to the addresses shown in the following tables.

Unit Parameters

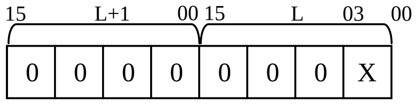
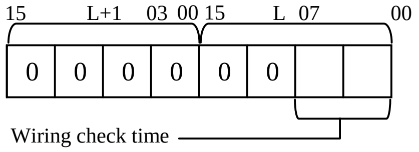
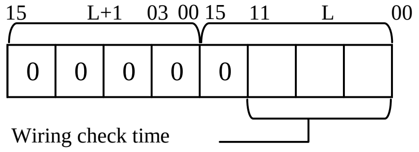
<table>
<tr><th colspan="2">Address</th><th rowspan="2">R/W</th><th rowspan="2">Name</th><th rowspan="2">Configuration</th></tr>
<tr><th>MC221</th><th>MC421</th></tr>
<tr><td>0FA6
(4006)</td><td>0FA6
(4006)</td><td>R</td><td>Output port setting</td><td>


Specifies whether the output ports will be used for general output or brake signal output. Bits 00 to 03 correspond to output ports 1 to 4 if the ports are used for general output, and to axes X through U if the ports are used for brake signal output.



X = 0: General output, X = 1: Brake signal output
</td></tr>
</table>

Unit Parameters

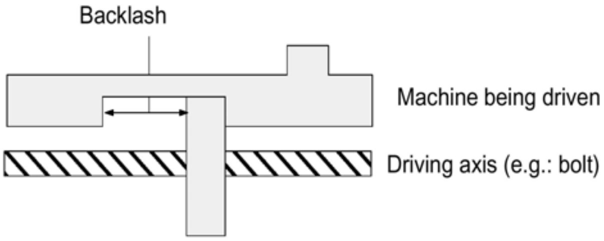
Address				R/W	Name	Configuration
MC221		MC421				
X axis	**Y axis**	**Z axis**	**U axis**			
107B (4219	1094 (4244)	10AD (4269)	10C6 (4294)	R	Wiring check	 15 L+1 00 15 L 03 00 0 0 0 0 0 0 0 X Specifies whether the wiring check function is used. X = 0: The wiring check function is not used. X = 1: The wiring check function is used.
107C (4220)	1095 (4245)	10AE (4270)	10C7 (4295)	R	Wiring check time	 15 L+1 03 00 15 L 07 00 0 0 0 0 0 0 Wiring check time Specifies the wiring check time within a range of 0 to 99 (x 10 ms). Range: 00 to 63 Hex (0 to 99 [x 10 ms])
107D (4221)	1096 (4246)	10AF (4271)	10C8 (4296)	R	Wiring check pulses	 15 L+1 03 00 15 11 L 00 0 0 0 0 0 Wiring check time Specifies the number of pulses to check the wiring within a range of 0 to 999 pulses. Range: 000 to 3E7 Hex (0 to 999 [pulses])

Backlash Correction

Overview

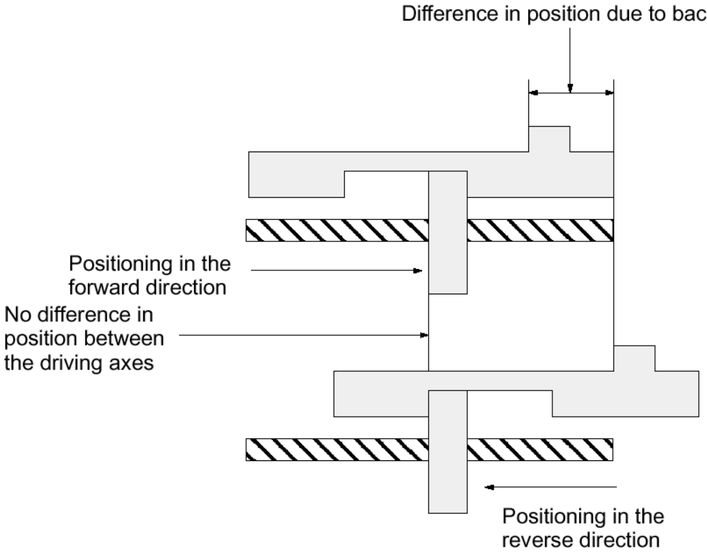
Mode: Automatic or Manual; **Method:** Command Area designation, CX-Motion

A backlash is a sudden backward movement of a driving axis that may be caused due to mechanical looseness as shown in the following illustration.

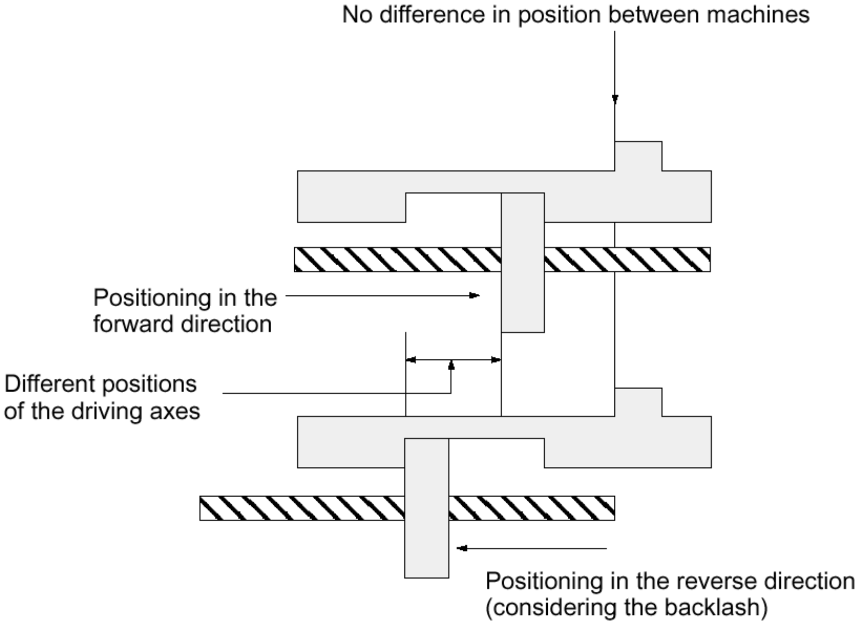


The positioning of a machine cannot be performed precisely if there is backlash.

As shown in the following illustration, for example, the position of a machine moved by 100 mm in the forward direction is different from that of the same machine moved by 100 mm in the reverse direction if there is a 1-mm backlash, even though there is no difference in position between the driving axes.


Difference in position due to backlash
Positioning in the forward direction
No difference in position between the driving axes
Positioning in the reverse direction

Backlash correction is a function for pre-registering the backlash space within a range of 0 to 10,000 pulses to minimize the positioning errors of machines moving in the forward or reverse direction to the same position, as shown in the following illustration. The setting is made with the backlash correction value in the servo parameters.


No difference in position between machines
Positioning in the forward direction
Different positions of the driving axes
Positioning in the reverse direction (considering the backlash)

Backlash correction is executed either at power up or when the axis is moved in the opposite direction from the direction in which it first moves after a restart.

1 Setting Backlash Correction

Either CX-Motion or the IOWR instruction can be used to set backlash correction.

Using CX-Motion

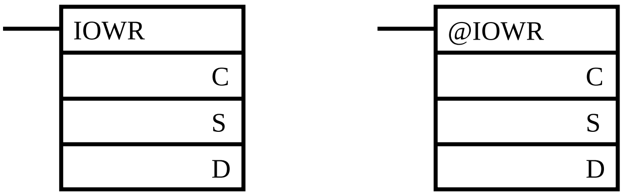
For details on using CX-Motion to make this setting, refer to the Section 1.

Using IOWR

To use IOWR to set the backlash correction, set the backlash correction address in the IOWR control code (C) and then transfer the backlash correction value to the MC Unit.
X axis: Address 11FC Hex (4604)
Y axis: Address 1215 Hex (4629)
Z axis: Address 122E Hex (4654)
U axis: Address 1247 Hex (4679)

Specifications

The following is a basic explanation of the IOWR instruction specifications for setting the backlash correction.

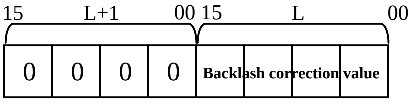


C: Control code, S: First source word, D: Destination unit number and total number of words to transfer.

Operand	Description
C	MC Unit address (Hex) Specifies the first address in the MC Unit to which data will be written. X axis: Address 11FC Hex (4604) Y axis: Address 1215 Hex (4629) Z axis: Address 122E Hex (4654) U axis: Address 1247 Hex (4679)
S	First source word Specifies the first word in the CPU Unit from which data is to be transferred.
D	D+1: Total number of words to transfer (Hex); D: Destination unit number (Hex) Specify the destination unit number to which data will be transferred and the total number of words to transfer. Number of words to transfer: Always 0002 Hex. (Refer to Transfer Data below.) Destination unit number: MC221: 0000 to 005D Hex (Unit numbers 0 to 93) MC421: 0000 to 005B Hex (Unit numbers 0 to 91) Example: #0002003, when two words of data are transferred to Unit #3

Transfer Data

Set the following data in the words beginning at the first source word, and execute the IOWR instruction.

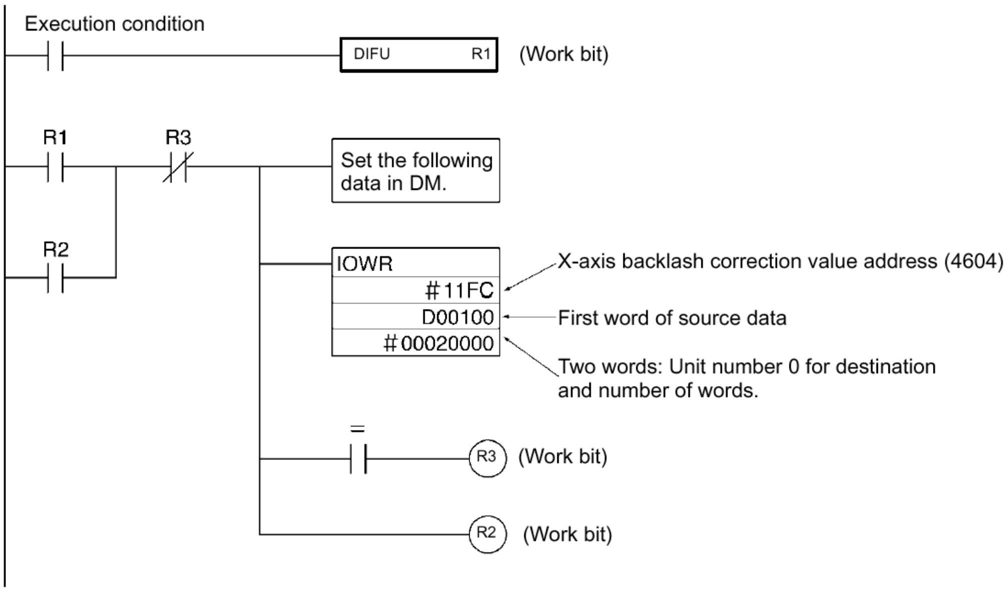


Set the backlash correction value for the X and Y axes between 0000 and 2710 Hex (1 to 10,000).

Programming Example

In the following example for transferring data, the MC Unit is mounted to a CS1 PC and assigned unit number 0. In this example, the backlash correction value is set to 1,000 pulses, and the IOWR operands and the data to be transferred are assumed to be normal.
When debugging, check the Error Flags from the ladder program.

Ladder Program


Execution condition
DIFU R1
(Work bit)
R1
R3
R2
Set the following data in DM.
IOWR
#11FC
D00100
#00020000
X-axis backlash correction value address (4604)
First word of source data
Two words: Unit number 0 for destination and number of words.
=
R3
(Work bit)
R2
(Work bit)

Transfer Data (DM)

D00100	03E8
D00101	0000

Backlash correction: 1000

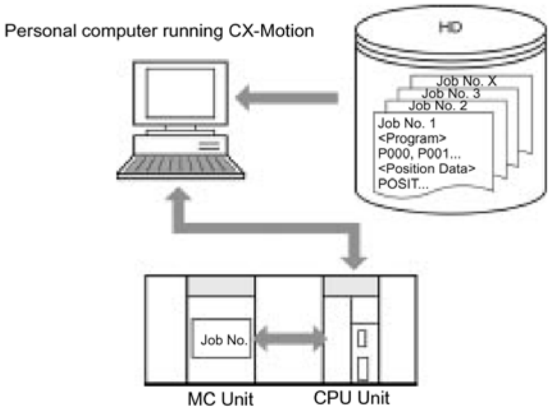
Automatic Loading

Overview

Mode: Automatic or Manual; **Method:** Command Area designation, CX-Motion
A maximum of 100 programs and up to 2,000 blocks in all programs total can be saved. The automatic loading function is provided for additional programs in order to support applications requiring more program capacity. This function allows more programs or program blocks to be executed by treating external memory devices connected to a personal computer running CX-Motion as MC Unit memory.

The automatic loading function is executed by the IOWR instruction for the Command Area. Either Automatic Mode or Manual Mode can be used.

Any number from 1 to 10,000 can be specified as the job number. CX-Motion must be used to create MC programs and position data in advance for job numbers that are to be used.



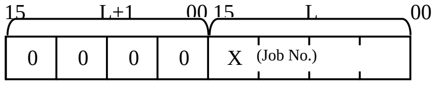
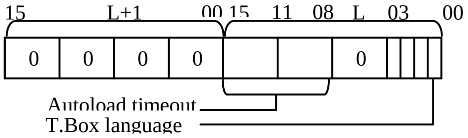
Explanation

1,2,3…
1. The MC Unit’s job number is always monitored by CX-Motion.
2. Using the IOWR instruction, a new job number is written to the job number currently in the MC Unit.
3. When CX-Motion detects the specified job number, the program and position data are downloaded to the MC Unit from the file for that job number created by CX-Motion.
4. Before downloading the program and position data, CX-Motion deletes from the MC Unit all of the programs for all tasks. Then it downloads the program and position data for the specified job number.
5. When the downloading has been completed, the CPU Unit is notified from the MC Unit. While the program and position are being downloaded, the Autoloading Bit in the PC Interface Area turns from OFF to ON, and when the downloading is completed normally it turns from ON to OFF.

1 Executing Automatic Loading

To execute automatic loading, set one of the following addresses for the control code (C) of the IOWR instruction and then transfer the job number to the MC Unit.

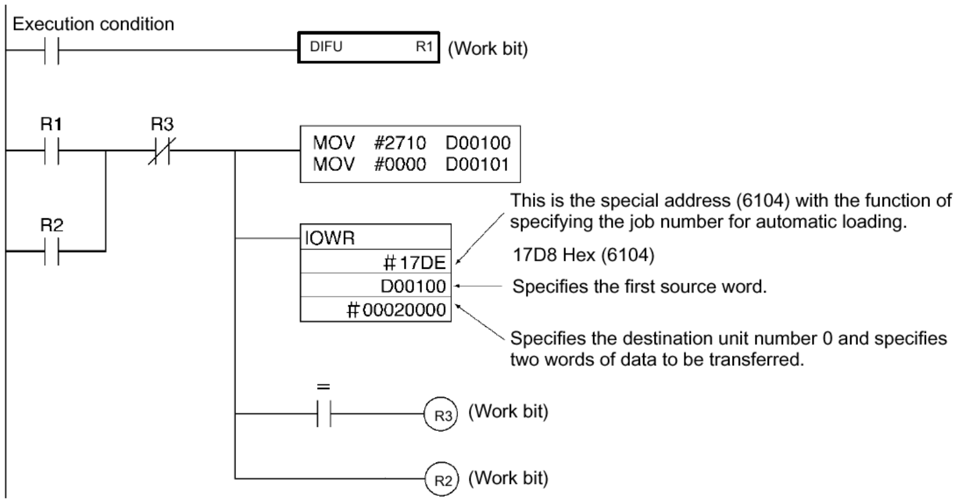
The Autoloading Time Up time is set with CX-Motion. The IORD instruction can be used for reading only.

Address		R/W	Name	Configuration
MC221	MC421			
17D8 (6104)	17D8 (6104)	R/W	Specifying automatic loading	 15 L+1 00 15 L 00 0 0 0 0 X (Job No.) This parameter is used when batch downloading programs and position data to an MC Unit from the external memory device of a personal computer with CX-Motion. X (Job No.) Specifies a job number within a range from 0001 to 2710 Hex (1 to 10,000). The following contents can be read from this address using an IORD instruction. X = 0000: Power supply ON, automatic loading completed normally X = 0001 to 2710 Hex (1 to 10,000): Specified job number or executing job number
0FA9 (4009)	0FA9 (4009)	R	Teaching Box language/ autoload timeout	 15 L+1 00 15 11 08 L 03 00 0 0 0 0 0 Autoload timeout T.Box language T.Box language Specifies whether the T.Box message is displayed in English or Japanese. (Only bit 0 is valid) X = 0: English, X = 1: Japanese Autoload timeout Specifies whether the time up time is displayed when the automatic loading function is used. Range: 00 to B4Hex (0 to 180 s) Time up will not be monitored if 00 is set.

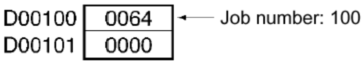
Note Programs and position data will not be properly downloaded to the MC Unit if the personal computer cable is disconnected or CX-Motion downloading is interrupted during execution of the automatic downloading function. If that occurs, perform the operation again.

Ladder Programming Example

In this example, job number 100 is specified. The MC Unit is mounted to a CS1 PC and assigned unit number 0, and the job number to be transferred (job number 100) is stored in words D00100 and D00101. Both the IOWR operands and the data to be transferred are assumed to be normal.
When debugging, check the Error Flags from the ladder program.


Execution condition
DIFU R1
(Work bit)
R1
R3
R2
MOV #2710 D00100
MOV #0000 D00101
IOWR
#17DE
D00100
#00020000
This is the special address (6104) with the function of specifying the job number for automatic loading.
17D8 Hex (6104)
Specifies the first source word.
Specifies the destination unit number 0 and specifies two words of data to be transferred.
=
R3
(Work bit)
R2
(Work bit)

Transfer Data (DM)

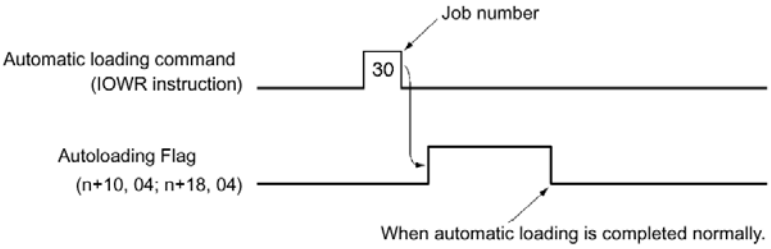


2 Related Bits in the PC Interface Area

Bit name	Model	Words	Bit	Description
Autoloading Error Reset Bit	MC421 MC221	n n	05	↑: Error reset for automatic loading
Autoloading Flag	MC421 MC221	n+10 n+18	04	↑: Automatic loading begun. ↓: Automatic loading finished.
Autoloading Error Flag	MC421 MC221	n+10 n+18	05	↑: Error occurred during automatic loading ↓: Automatic loading error reset turned ON.

Autoloading Flag

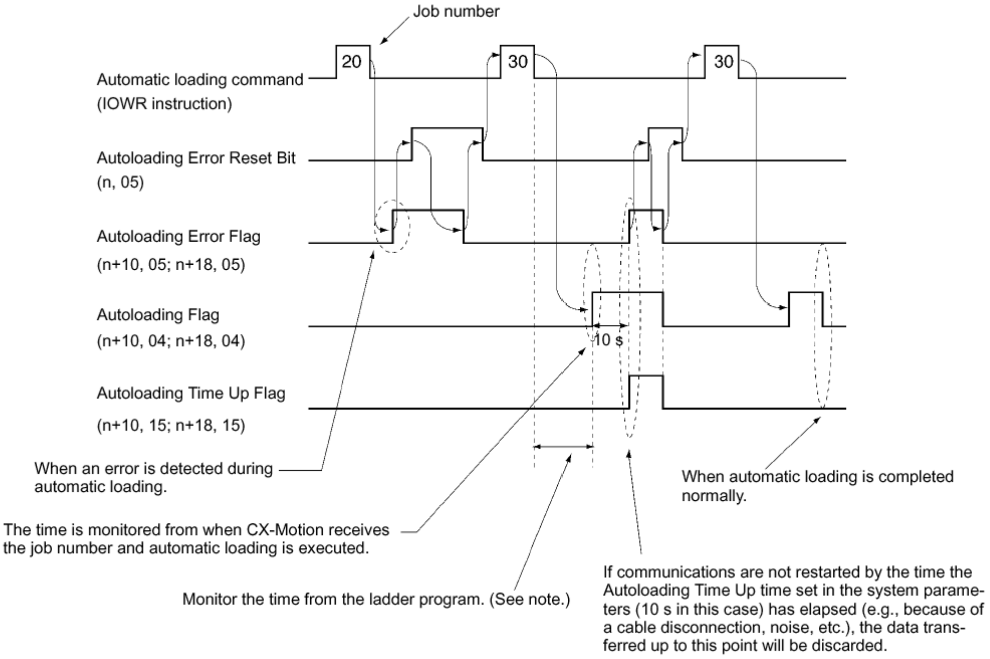
This flag turns ON when the data for the specified job number begins to be down-loaded as the result of executing the IOWR instruction. It turns OFF when all of the data for that job number has been downloaded.



Autoloading Error Flag and Error Reset Bit

The Autoloading Error Flag turns ON when an error occurs during automatic loading. When the Autoloading Error Reset Bit is turned ON, the Autoloading Error Bit will turn OFF. Unless the cause of the error has been cleared, however, the Autoloading Error Flag will turn back ON again the next time the automatic loading is executed.

If communications are not restarted by the time the Autoloading Time Up set in the Unit parameters has elapsed, an error will occur and the Autoloading Error Flag will turn ON. If that occurs, the data transferred up to that point will be discarded.



When an error is detected during automatic loading.

The time is monitored from when CX-Motion receives the job number and automatic loading is executed.

Monitor the time from the ladder program. (See note.)

When automatic loading is completed normally.

If communications are not restarted by the time the Autoloading Time Up time set in the system parameters (10 s in this case) has elapsed (e.g., because of a cable disconnection, noise, etc.), the data transferred up to this point will be discarded.

Note The MC Unit does not monitor the time from when it receives the automatic loading command until the Autoloading Flag turns ON. There may be cases where the power to the personal computer is turned OFF or a cable is disconnected, so monitor this time from the ladder program.

Present Position Preset

Overview

Mode: Manual; **Method:** Command Area designation

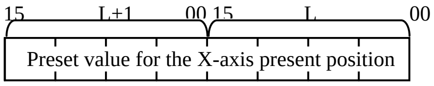
The present position preset function is used to preset the preset position to any given value. With previous MC Unit models, it was executed from the PC Inter-face Area, but with the MC221 and MC421 it is executed by an IOWR instruction using the Command Area. It can only be used in Manual Mode, and will be ignored if used in Automatic Mode.

After the present position has been preset, the reference origin is established automatically, so there is no need to execute an origin search. It is also possible to set the present position as the origin by using the present position preset function to set the present position to “0.”

⚠ **Caution** When present position preset is executed for an axis for which an absolute encoder is used, only the present position stored in the MC Unit will be preset. The present position stored in the absolute encoder will not be preset, and it will be automatically updated with the next servo-lock.

1 Executing Present Position Preset

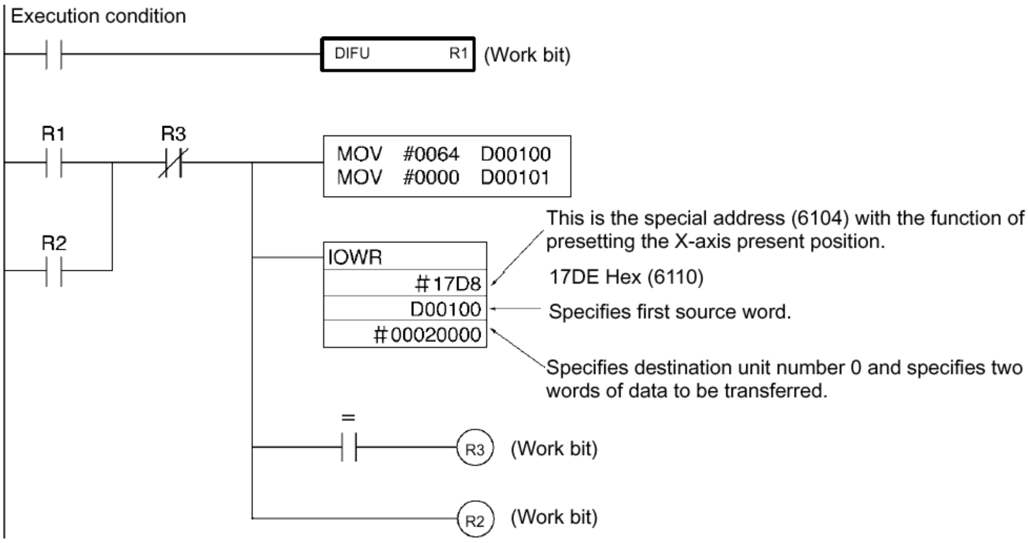
To execute present position preset, set the following addresses for the IOWR instruction’s control code (C) and then transfer the present position preset value to the MC Unit. For details on the Command Area, refer to 3-6 Command Area. For details on IOWR specifications, refer to 4 Data Transfer and Storage.

Address		R/W	Name	Configuration
MC221	MC421			
17DE (6110)	17DE (6110)	W	Presetting the X-axis present position	 15 L+1 00 15 L 00 Preset value for the X-axis present position Sets the X-axis present position to any value and establishes the origin. The data is 32-bit signed binary data. L+1: Leftmost 16 bits, L: Rightmost 16 bits Range: FD9DA601 to 026259FF Hex (–39999999 to 39999999) The minimum setting unit is set to 2 (for 0.01), the display unit is set to 0 (for mm), and the reference origin offset is 3.99 [mm] (i.e., 399 x 0.01) if the data is 399.
17DF (6111)	17DF (6111)	W	Presetting the Y-axis present position	Sets the Y-axis present position to any value and establishes the origin. The data configuration is the same as that for presetting the X-axis present position.
---	17E0 (6112)	W	Presetting the Z-axis present position	Sets the Z-axis present position to any value and establishes the origin. The data configuration is the same as that for presetting the X-axis present position.
---	17E1 (6113)	W	Presetting the U-axis present position	Sets the U-axis present position to any value and establishes the origin. The data configuration is the same as that for presetting the X-axis present position.

Ladder Programming Example

In this example, the X-axis present position is set to 10,000. The MC Unit is mounted to a CS1 PC and assigned unit number 0, and the data to be transferred (i.e., the present position) is stored in words D00100 and D00101. Both the IOWR operands and the data to be transferred are assumed to be normal.

When debugging, check the Error Flags from the ladder program.


Execution condition
DIFU R1
(Work bit)
R1
R3
R2
MOV #0064 D00100
MOV #0000 D00101
IOWR
＃17D8
D00100
＃00020000
This is the special address (6104) with the function of presetting the X-axis present position.
17DE Hex (6110)
Specifies first source word.
Specifies destination unit number 0 and specifies two words of data to be transferred.
=
R3
(Work bit)
R2
(Work bit)

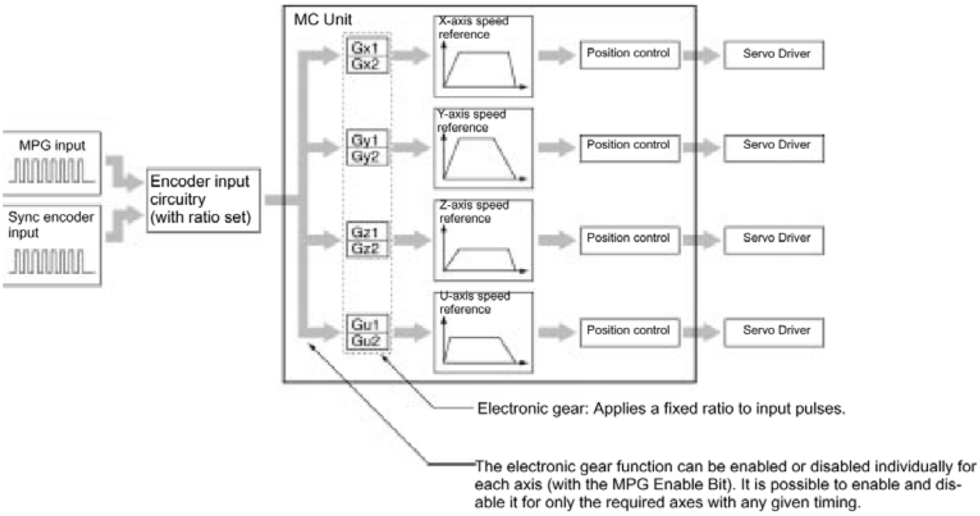
D00100	2710	← Present position preset value: 10000
D00101	0000	

Electronic Gear Function

Overview

Mode: Manual; **Method:** System parameter setting (IOWR or CX-Motion)

The electronic gear function is provided for operations such as using a Manual Pulse Generator (MPG) for handle feeding or using an absolute encoder for synchronizing axis feeding with a conveyer. Both the numerator and denominator can be specified, so this electronic gear function enables more precise feed control than was possible with previous MC Unit models.



Electronic gear: Applies a fixed ratio to input pulses.

The electronic gear function can be enabled or disabled individually for each axis (with the MPG Enable Bit). It is possible to enable and disable it for only the required axes with any given timing.

1 Setting the System Parameters

Setting MPG or Sync Encoder

Using CX-Motion, make the setting in the Unit parameters to determine whether an MPG or a sync encoder is to be used for pulse inputs.

The IORD instruction can also be used to read the parameter settings

This setting specifies the ratio for pulse input devices connected to the MPG or sync encoder. For the MPG, it is used to maintain a ratio of one. When it is set for a sync encoder, the ratio can be specified as 1, 2, or 4.

Setting the Electronic Gear Ratio

When an MPG is used, the electronic gear ratio is the same as it was for previous MC Unit models.

The gear ratio for the MC221 and MC421 is set by CX-Motion or by an IOWR instruction. There are two ways to change the electronic gear ratio:

- Change the electronic gear combination using the PC Interface Area. (A maximum of four can be changed in this way.)
- Specifying the electric gear combination in the PC Interface Area, and then using an IOWR instruction to change that value directly. (The number of combinations is unlimited.)

Regardless of which of these methods is used, the change will go into effect immediately.

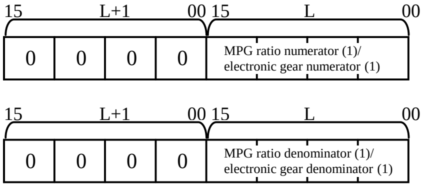
Note With the electronic gear function, when there is a remainder in the total when the input pulses are multiplied by the gear ratio, it is possible to achieve highly accurate synchronization by adding the remaining pulses to the next input pulses. If the electronic gear ratio is changed during a synchronized operation, however, the remainder will be taken as 0 and the calculation will begin with the new electronic gear ratio. For that reason there will be a temporary drop in speed from that which was in effect before the gear ratio was changed.

Using CX Motion

For details on using CX-Motion to set the electronic gear ratio, refer to the Section 1.

Using IOWR

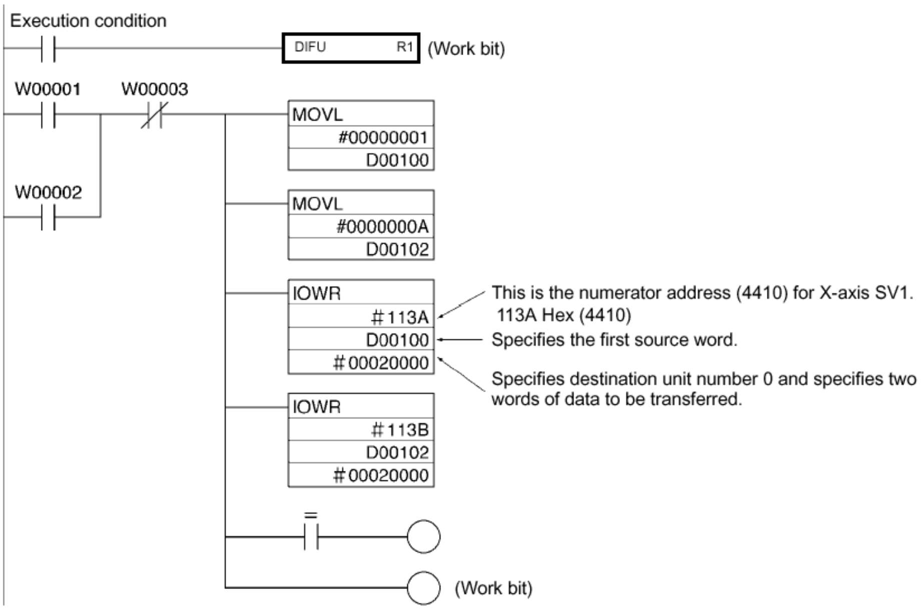
Use the IOWR instruction to specify the following addresses for the control code, and set the numerator and denominator in any area. (Refer to the ladder programming example provided below.) The addresses are shown in hexadecimal, with their decimal number equivalents shown underneath in parentheses.

				R/W	Name	Configuration	Default setting
MC221		MC421					
113A (4410)	1153 (4435)	116C (4460)	1185 (4485)	R/W	MPG ratio numerator (1)/electronic gear numerator (1)	 15 L+1 00 15 L 00 0 0 0 0 MPG ratio numerator (1)/ electronic gear numerator (1) 15 L+1 00 15 L 00 0 0 0 0 MPG ratio denominator (1)/ electronic gear denominator (1) Sets the ratio for an MPG/electronic gear pulse for setting 1. Range (numerator): D8F0 to 2710 Hex (–10000 to 10000) Range (denominator): 0001 to 2710 Hex (1 to 10000) 0.0001 ≤ ∣ Numerator/denominator ∣ ≤ 10000	0000 0001 (1)
113B (4411)	1154 (4436)	116D (4461)	1186 (4486)	R/W	MPG ratio denominator(1)/ electronic gear denominator(1)		0000 0001 (1)
113C (4412)	1155 (4437)	116E (4462)	1187 (4487)	R/W	MPG ratio numerator (2)/electronic gear numerator (2)	Sets the ratio for an MPG/electronic gear pulse for setting 2. The data configuration and range are the same as those for MPG ratio/electronic gear numerator (1)/denominator (1).	0000 000A (10)
113D (4413)	1156 (4438)	116F (4463)	1188 (4488)	R/W	MPG ratio denominator (2)/electronic gear denominator (2)		0000 0001 (1)
113E (4414)	1157 (4439)	1170 (4464)	1189 (4489)	R/W	MPG ratio numerator (3)/electronic gear numerator (3)	Sets the ratio for an MPG/electronic gear pulse for setting 3. The data configuration and range are the same as those for MPG ratio/electronic gear numerator (1)/denominator (1).	0000 0064 (100)
113F (4415)	1158 (4440)	1171 (4465)	118A (4490)	R/W	MPG ratio denominator (3)/electronic gear denominator (3)		0000 0001 (1)
1140 (4416)	1159 (4441)	1172 (4466)	118B (4491)	R/W	MPG ratio numerator (4)/electronic gear numerator (4)	Sets the ratio for an MPG/electronic gear pulse for setting 4. The data configuration and range are the same as those for MPG ratio/electronic gear numerator (1)/denominator (1).	0000 00C8 (200)
1141 (4417)	115A (4442)	1173 (4467)	118C (4492)	R/W	MPG ratio denominator (4)/electronic gear denominator (4)		0000 0001 (1)

Ladder Programming Example

In this example, the electronic gear ratio for set value 1 of the X axis (address 113A Hex) is set to 0.1 (1/10). The MC Unit is mounted to a CS1 PC and assigned unit number 0, and the data to be transferred (i.e., the numerator and denominator) is stored in words D00100 to D00103. Both the IOWR operands and the data to be transferred are assumed to be normal.

When debugging, check the Error Flags from the ladder program.



This is the numerator address (4410) for X-axis SV1. 113A Hex (4410)

Specifies the first source word.

Specifies destination unit number 0 and specifies two words of data to be transferred.

Transfer Data (DM)

D00100	0001	SV1 numerator: 1
D00101	0000	
D00102	000A	SV1 denominator: 10
D00103	0000	

2 Related Bits

The following table shows the methods for selecting set values 1 to 4 and enabling the selected set value.

Bit name	MCU Model	Word				Bit	Description
		X axis	Y axis	Z axis	U axis		
Enable MPG Sync Encoder	MC421 MC221	n+11 n+7	n+13 n+9	n+15 ---	n+17 ---	08	↑:Begin MPG operation 1:Continue MPG operation ↓:Stop MPG operation
MPG Sync Encoder Ratio Specifier 1						14	MPG ratio specifier 2 1 / MPG ratio selection 0 0 Select MPG ratio 1 0 1 Select MPG ratio 2 1 0 Select MPG ratio 3 1 1 Select MPG ratio 4
MPG Sync Encoder Ratio Specifier 2						15	

Acceleration and Deceleration Curves

Overview

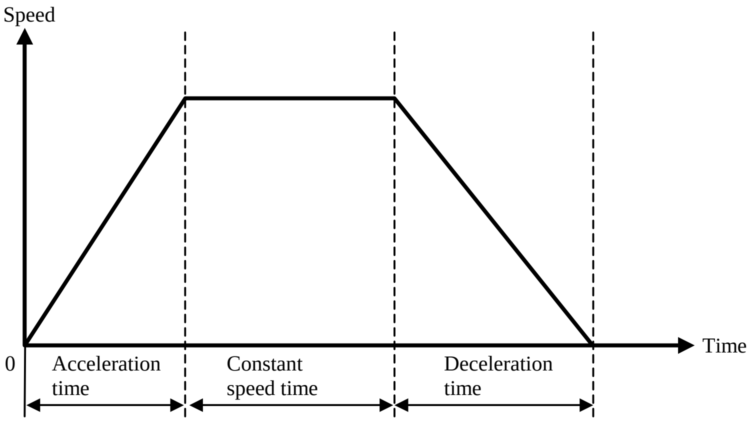
Mode: Automatic or Manual; **Method:** System parameters (CX-Motion only)
In positioning operations, the speed is accelerated gradually at the beginning and decelerated gradually toward the end to achieve smooth movement. For the MC Unit, either a trapezoidal curve or an S curve can be used as the acceleration and deceleration curve for the starting and stopping operations for each axis.

The acceleration time and deceleration time can be set separately.

Refer to the CX-Motion operation manual for details on setting the acceleration and deceleration curve and the acceleration and deceleration times. The acceleration and deceleration times can also be set from the ladder program using the IOWR instruction.

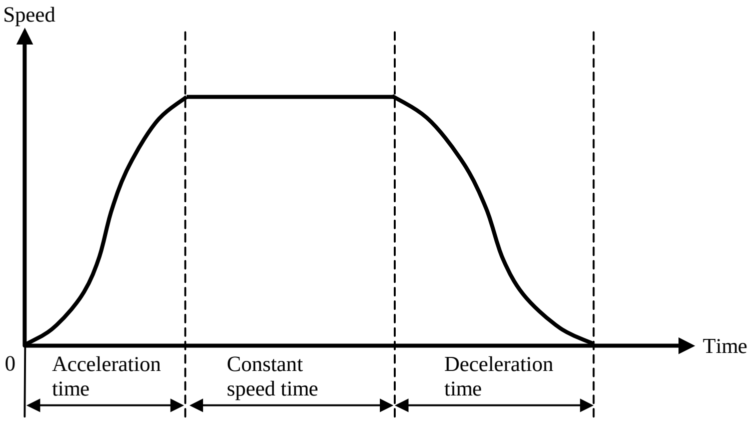
Trapezoidal Curve

With the trapezoidal curve, acceleration is constant during the acceleration time and deceleration time.



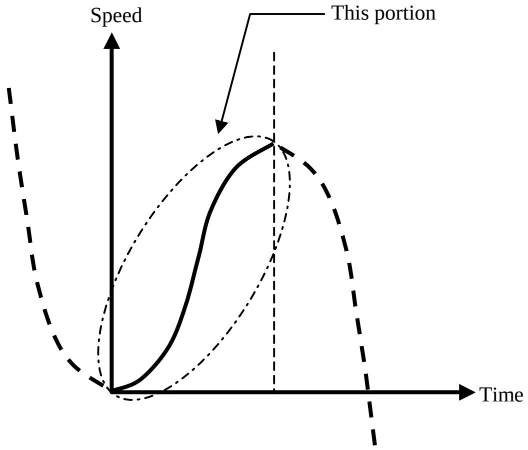
S Curve

With the S curve, acceleration during the acceleration time and deceleration time changes with time. When the initial acceleration speed is reduced for acceleration or deceleration, the acceleration or deceleration time can be shortened using the S curve if there is a leeway in the maximum acceleration speed.



Note If the acceleration time or the deceleration time is the same, the maximum acceleration in the S curve is 1.5 times that in the trapezoidal curve. Therefore, when using a program set for a trapezoidal curve for an S curve, the acceleration and deceleration times set for the trapezoidal curve must be multiplied by 1.5. The maximum acceleration in the S curve will then fall within the acceleration set for the trapezoidal curve, allowing the motor to be driven smoothly.

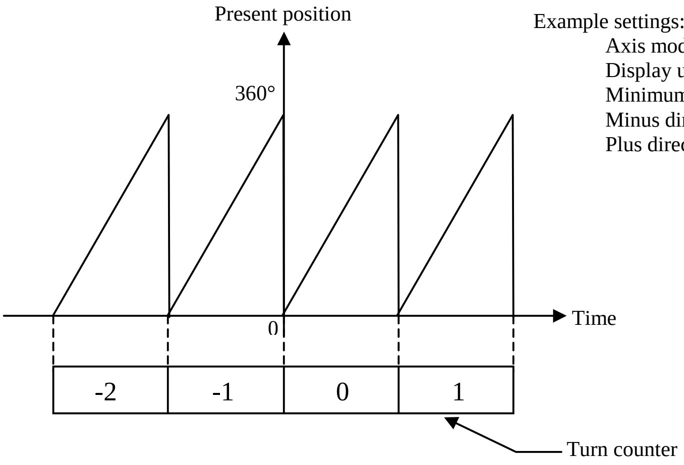
The S curve used by the MC Unit uses a tertiary function, as shown in the following diagram.



Unlimited Feeding

Overview

Mode: Automatic or Manual; **Method:** System parameters (CX-Motion only)
The unlimited feeding function makes it possible to control axes for applications such as turntables or one-direction conveyers, which require unlimited feeding. With axes set for unlimited feeding, the present position can be updated for any set range. In the case of a turntable, for example, the present position will be updated as shown in the following diagram if a range of 0° to 360° is set. The number of turns will be remembered.



Example settings: Machine parameters
Axis mode: Unlimited Mode (See note.)
Display unit: Degrees
Minimum setting unit: 0.1
Minus direction software limit: 0 (See note.)
Plus direction software limit: 360 (See note.)

Note Setting these machine parameters will set unlimited feeding.

1 Setting System Parameters

The following system parameter must be set when using the unlimited feeding function:
- Axis mode (machine parameter): Unlimited Feed Mode

This parameter must be set using the CX-Motion. The IORD instruction can only be used to read the parameter setting.

When Unlimited Feed Mode is set as the axis mode, the software limits will not operate regardless of whether or not the origin has been determined.

The present position is set by the ring counter method at the value determined by the positive and negative software limits, using the minimum setting unit, so in the Unlimited Feed Mode be sure to set the origin (0) between the positive and negative software limits. (Refer to the examples below.)

The positive and negative software limits are set in the same way by CX-Motion. The IORD instruction can be used to read the parameter settings.

Example 1

In this example, the display unit is degrees, the minimum setting unit is 0.1, the negative software limit is 00, and the positive software limit is 360.0. The present position will be between 0.0 and 359.9.

Example 2

In this example, the display unit is degrees, the minimum setting unit is 0.1, the negative software limit is 360.0, and the positive software limit is 360.0. The present position will be between –360.0 and 359.9.

All axis operation commands can be used with the Unlimited Feed Mode. For example, an error would normally be generated for G01 if the origin is not deter-mined, but that command can be used if the axis mode is set to Unlimited Feed Mode.

Note When the present position update range is converted to pulses, select a gear ratio and encoder resolution such that it will be expressed in integers. If it is not in integers, the errors will accumulate whenever the present position update range is exceeded, and correct positioning will not be possible.

For example, when controlling a turntable, suppose that the present position up-date range is 0° to 360°. If the gear ratio is 7/99 and the encoder resolution is 1,000 pulses, the following error will be accumulated with each 360-degree turn.

Gear ratio (degrees/pulse)
= 7 x 360° / (99 x 1,000 pulses)
= 2,520 / 99,000 pulses

360° x 99,000 / 2,520 = 14,142.857 pulses

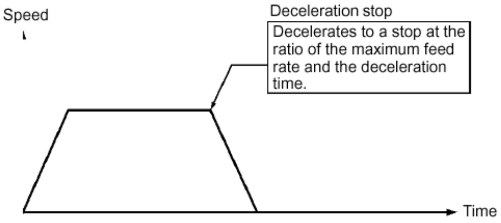
Thus an error of 0.857 pulses will be accumulated with each turn.

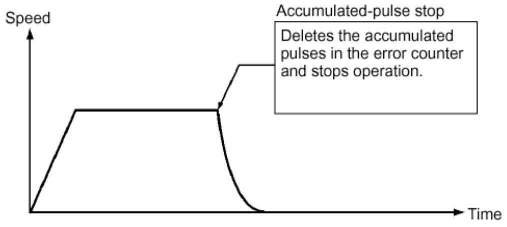
2 Stopping

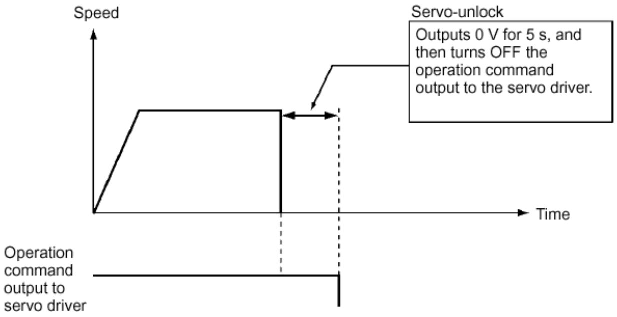
The MC Unit has various ways of stopping axis operation, as shown in the following table.

Method	Activation timing	Remarks
Manual/Automatic Bit	Upon input.	Each axis is decelerated to a stop upon input. During interpolation operations, axis operation is stopped after moving from the interpolation locus.
Cycle Start OFF	When G-language program interpretation is started.	This bit is checked when interpretation of the G-language is started (at the beginning of a block). If it is OFF, the G-language program is not executed. In In-position Check OFF Mode or Pass Mode, however, this bit is not checked until the next block.
Pause ON	Upon input.	Each axis is decelerated to a stop upon input. During interpolation, axis operation is stopped at the interpolation locus.
Forced Block End	Upon input.	Each axis is decelerated to a stop upon input. During interpolation operations, axis operation is stopped after moving from the interpolation locus.
Optional End	Upon input.	Each axis is decelerated to a stop upon input. During interpolation operations, axis operation is stopped after moving from the interpolation locus.

Method	Activation timing	Remarks
Emergency Stop	Upon input.	Each axis is stopped immediately upon input. There are two methods for stopping: accumulated-pulse stop and servo-off stop. The selection is made in the system parameters.
CCW, CW Limits Inputs	Upon input.	Each axis is stopped immediately upon input. There are two methods for stopping: accumulated-pulse stop and servo-unlock stop. The selection is made in the system parameters.
Software Limit ON	When a software limit is exceeded.	Each axis is decelerated to a stop when a software limit is exceeded. During interpolation operations, axis operation is stopped after moving from the interpolation locus.
Other errors	Refer to Section 12 Troubleshooting (W359-E1-1)	


Speed
Deceleration stop
Decelerates to a stop at the ratio of the maximum feed rate and the deceleration time.
Time


Speed
Accumulated-pulse stop
Deletes the accumulated pulses in the error counter and stops operation.
Time


Speed
Servo-unlock
Outputs 0 V for 5 s, and then turns OFF the operation command output to the servo driver.
Time
Operation command output to servo driver