

H4M

Mill CNC Controller

Manual

March, 2005

HUST Automation Inc.

No. 80 Kon Yei Road, Toufen, Miaoli, Taiwan, R.O.C.

Tel: (037) 623242

Fax: (037) 623241

TABLE OF CONTENTS

1	MAIN FEATURES OF MILL CNC CONTROLLER	1-1
2	INSTRUCTION	2-1
2.1	Basic Instructions	2-1
	Power-On Display	2-1
	Standby Display	2-1
	Auto Mode Display	2-2
	MDI Mode Display	2-3
	Home Origin Mode Display	2-3
	Jog Mode display	2-4
	Edit Mode display	2-5
	Program Mode Display	2-5
	I/O Mode Display	2-7
	Tool Compensation Display	2-7
	Graph Mode Display	2-9
2.2	Programming Introduction	2-10
2.2.1	Part Program	2-10
2.2.2	Methods Of Programming	2-10
2.2.3	The Composition of A Part Program	2-11
2.2.4	Coordinate System	2-14
2.2.5	Mill CNC Control Range	2-19
2.3	Program Editing	2-21
2.3.1	Program Selection	2-21
2.3.2	New Program Edition	2-22
2.3.3	Old Program Revision	2-24
2.3.4	Program Delete	2-28
2.3.5	Decimal Points Entering	2-28
2.3.6	Notes For Edition	2-29
2.3.7	Circular Entering	2-30
3	G/M CODES	3-1
3.1	Explanations	3-1
3.2	Rapid Positioning (Traverse Speed), G00	3-3
3.3	Linear Cutting, G01	3-4

3.4	CNC and Master/Slave Mode	3-6
3.5	Circular (Arc) Cutting Command, G02 and G03	3-11
3.6	Thread Cutting, G17,G18 , G19	3-14
3.7	Feed-Hold, G04	3-16
3.8	Reset Origin of Machine Coordinate, G08	3-16
3.9	Manual Data Input Function, G10	3-17
3.10	English / Metric measuring System Conversion, G20 G21	3-21
3.11	Move To The First Reference Point, G28	3-21
3.12	Return To The Previous Location From The Reference Point, G29	3-22
3.13	Move to The Second (2nd) Reference Point, G30	3-22
3.14	Skip Function, G31	3-22
3.15	Work-piece Dimension Enlarging and Shrinking, G50, G51	3-23
3.16	Work Coordinate System Setting, G54~G59	3-24
3.17	Mirror-effect Cutting, G68, G69	3-28
3.18	Absolute and Increment Coordinate Command, G90, G91	3-29
3.19	Canned Cycle Functions (H4M only),G81~G89, G80	3-29
3.20	G90 or G91-Absolute or Increment Coordinate Setting	3-30
3.21	Canned Cycle Commands, G80, G81~G89	3-31
3.22	G80 Hole Drilling Canned Cycle-Cancel	3-31
3.23	G81 Hole Drilling Canned Cycle-Set	3-31
3.24	G82 Hole Drilling Canned Cycle-Temporary Stop at Bottom	3-32
3.25	G83 Deep Hole Drilling Canned Cycle	3-32
3.26	G84 Thread Tapping Cycle	3-33
3.27	G85 Boring Cycle (Spindle Stop at Bottom)	3-33
3.28	G86 Boring Cycle (Temporary Stop at Bottom)	3-34
3.29	G22 Linear Groove	3-34
3.30	G23 Curve Groove	3-35
3.31	G24 Rectangular Groove	3-35
3.32	G25 Circular Groove	3-36
3.33	Specified Canned Cycle	3-37
3.34	G34 Circular Drill Canned Cycle	3-37
3.35	G35 Angular Drill Canned Cycle	3-38
3.36	G36 Arc Drill Canned Cycle	3-38
3.37	G37 Grid Drill Canned Cycle	3-39
3.38	Auxiliary Functions, M-codes, S-codes	3-39
3.39	Sub-program	3-40
3.40	【MACRO】 Command, G65	3-41

4 MCM PARAMETERS		4-1
4.1	MCM Parameter	4-1
4.2	Description of MCM Machine Constants	4-6
5 CONNECTIONS		5-1
5.1	Connecting System Descriptions	5-1
5.2	System Installation	5-2
5.2.1	Ambient Surroundings	5-2
5.2.2	Cabinet Considerations	5-2
5.2.3	Thermal Design in the Cabinet	5-3
5.2.4	External Dimensions and Diagrams 【H4M Keyboard Panel】	5-4
5.2.5	H4M CPU Main Board Connectors (Rear View)	5-5
5.2.6	Dimension for H4M Controller Box (Rear view)	5-6
5.2.7	Dimension for H4M Controller Box (Front view)	5-6
5.2.8	Cutout Dimension for H4M Controller Installation	5-7
5.3	Input/Output Interface Connection (I/O)	5-8
5.3.1	Input Board / Output Board –Terminal Block Type	5-8
	NPN Type Standard Input Board (I bit)-24 IN	5-8
	NPN Type Standard Output Board (O bit)-16 OUT	5-8
5.3.2	Input Board/Output Board- CE Rules	5-9
	Input Board	5-9
	Output Relay Board	5-10
5.3.3	I/O Interface Connecting Arrangement	5-11
5.3.4	Input Signals	5-11
	Input Signal Type	5-12
	Input Connecting Diagram (Direct Input Controller)	5-12
	Input Connecting Diagram (I bit Input Controller)	5-12
5.3.5	Output Signals	5-12
	Output Signal Type	5-13
	Output Connecting Diagram (Direct Output to Machine)	5-13
	Output Connecting Diagram (Output Board to Machine)	5-13
5.4	Connecting Diagrams	5-13
5.4.1	Connector types	5-13
5.4.2	Indicated numbered connectors	5-14
5.4.3	H4M Connection (Terminal Block)	5-15
	H4M Main Connecting Diagrams	5-15

	Emergency-Stop Circuit-1 (Advised Connecting Diagrams)	5-16
	Emergency-Stop Circuit-2 (Simplified Connecting Diagrams)	5-17
	H4M Spindle Connection	5-18
5.4.4	H4M Connections (C.E. Style Connectors)	5-19
	H4M Main Connecting Diagrams	5-19
	Emergency-Stop Circuit-1 (Advised Connecting Diagrams)	5-20
	Emergency-Stop Circuit-2 (Simplified Connecting Diagrams)	5-21
	Servo motor Connections (Mitsubishi J2S Motor)	5-22
5.4.5	Motion Control (Servo/MPG) Connection (Suitable Machine: H4M)	5-23
5.4.6	AC Power System Connector	5-24
5.4.7	MPG Hand-wheel Connecting Diagram	5-24
5.4.8	RS232 Connector Pin Assignment and Connecting Diagram	5-25

6 ERROR MESSAGES 6-1

7 APPENDIX 7-1

	Input Arrangement	7-1
	Output Arrangement	7-2
	M-Code Versus I/O	7-2

1 MAIN FEATURES OF MILL CNC CONTROLLER

- Controlled Axis: X,Y,Z and Spindle Encoder Feedback
- Program Designed by CAD/CAM on PC. Program input and DNC on-line execution from PC through RS232C interface.
- Memory Capacity for CNC main board - 512k.
- Battery Backup for CNC program storage in case of power-off.
- Backlash error compensation for worn lead screw.
- Provide 40 sets of tool length offset.
- Self-designed MACRO Program.
- Single block and continuous commands.
- Option Skip functions.
- Option Stop and Feed hold functions.
- Simultaneous use of absolute and incremental coordinate in the program.
- Self-diagnostic and error signaling function.
- Direct use of “ R”, “ I” and “ J” incremental value for radius in circular cutting.
- MPG hand-wheel test and collision free function for cutting product at the speed controller by MPG.
- Equipped with 24 standard programmable inputs and 16 outputs.

This operator's manual includes program editing, G/M code, parameter settings, connections and maintenance (plus warn descriptions) with examples and explanations for each command instruction.

If there are any problems in application, please fill out a problem sheet indicating the natures of the problem. Send it by either fax or mail. We will respond to you as soon as possible.

2 INSTRUCTION

2.1 Basic Instructions

Operating Diagrams

● Power-on Display

You will see this image after the power is on like the illustration below:



Fig.2-1

● Standby Display

After 3 seconds, you will enter the standby display. You can also obtain the same image when you press "Reset" key like the image below:

PGM-NO: 000		CNT: 0000000	
X	-0000.000	MPG	: 000
Y	-0000.000	G00 MFO	: 000%
Z	-0000.000	G01 MFO	: 000%
		SSO MFO	: 000%
Spindle	: 0000000	M	: 00000
Feed Rate	: 0000000	T	: 00000
OFFSET-Z	: -0000.000	S	: 00000
Time: 0000000 sec			
Motor not home <input type="checkbox"/> READY <input type="checkbox"/> STOP			

Fig.2-2

● Auto Mode Display

Press key “Auto/ MDI” to enter the auto mode, the display is shown below:

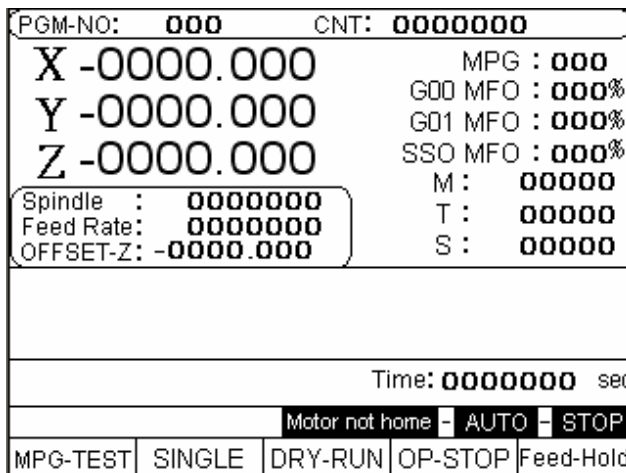


Figure 2-3

Soft keys under the auto mode:

1. Program Feed-Hold: only valid during the program operation.
In the program operation, press the key and the program will stop immediately. You can continue operating the program by press this soft key again or CYCST key.
2. Single Step Execution: users can select the function any time without being limited in the state of operation or stop. This function can only carry one step by each key press of restart instead of executing the whole program continuously.
3. Program Dry Run: only can be selected before the program execution, however, it can be canceled anytime during the execution. When the program dry run is being selected, the speed of F values in the program is in valid if it is executed in the highest speed (MCM79~81).
4. MPG - TEST: users can select the function any time without being limited in the state of operation or stop. When the function is being selected, the movement of all the axis in the program can only be controlled by MPG. If there is no input of MPG, the axis will stop moving.
5. Option Stop: only can be selected before the program execution. When option stop is being selected, M01 commend in the program will be considered as a stop commend. It is meaningless if M01 is not selected.

Part numbers: each execution to M02, M30, M99 will add on one. If users need to return to zero manually, please press the “0” key twice immediately to return zero.

Part time: show the current executing time. After each program end or stop, it will automatically return to zero when it restarts.

MDI Mode Display

Press [Auto/MDI] key twice to enter the MDI mode, the display is shown below:

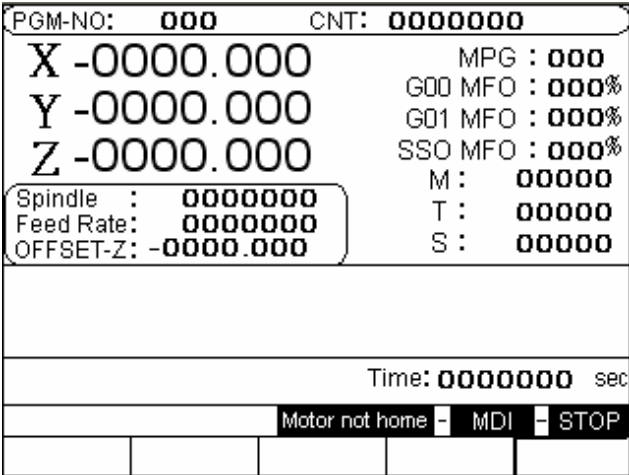


Fig.2-4

Home Origin Mode Display

Press [JOG/HOME] key twice to enter the home origin mode, the display is shown below:

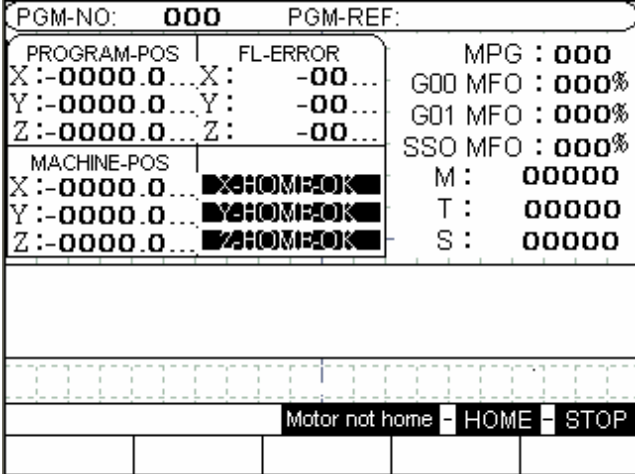


Fig. 2-5

Methods for returning the origin:

1. Select the axis: there are some ways to select the axis. You can either press the English letter “ X ”, “ Y ”, “ Z ” on the right of the screen directly or press the key button” X+”, “X-”, “Y+”, “Y-”, “Z+”, “Z-” to make your selection.

2. Press "CYCST" key

Note: if the three axes come back to its origin at the same time, press the key "Home" in 3 seconds.

Jog Mode display

Press [JOG/HOME] key to enter jog mode, the display is shown below:

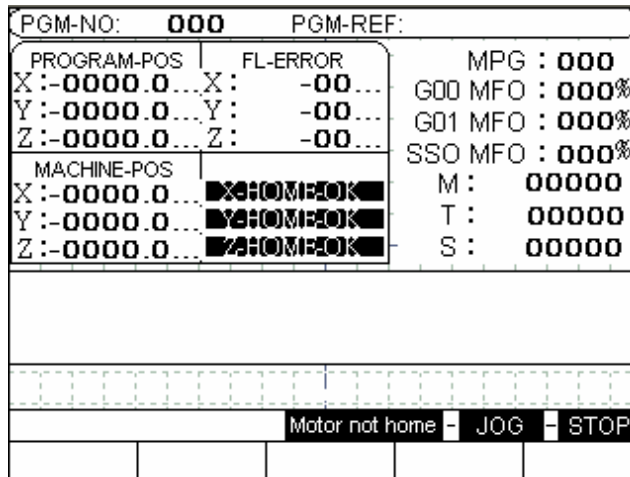


Fig.2-6

There are several functions under the jog mode:

1. Axis positioning:
 - a. Continuous movement: (Single step function is not on)
Continuously press "X+" key and X-axis will do positive movement, X-axis will do negative movement. Z-axis is followed the same way.
 - b. Move single step:
Select your desired distance for each single step such as 0.001,0.01,0.1,1 and press X+, X-,Y+,Y-, Z+,Z-. The system will follow the selection to make the step.
2. Manual Switch (Soft_Key):
 - a. Spindle: Clockwise, Counter Clockwise, Stop.
 - b. Coolant: Press on and off key
 - c. Lubricant: Press the key and it will be provided after 1 second.
LED is the indicator for the operation.

● Edit Mode display

Press "Edit/PRNO" to enter the edit mode, the display is shown below:

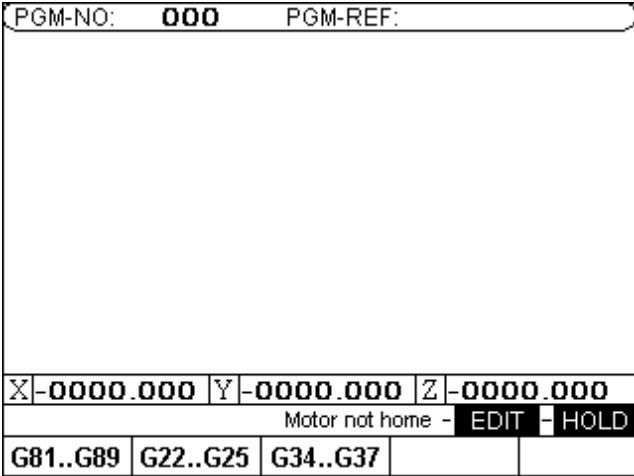


Fig.2-7

This screen mode can be edited directly (Please see the edit chapter for details).

● Program Mode Display

Press twice 『Edit/PRNO』 to enter the program mode, the display is shown below:

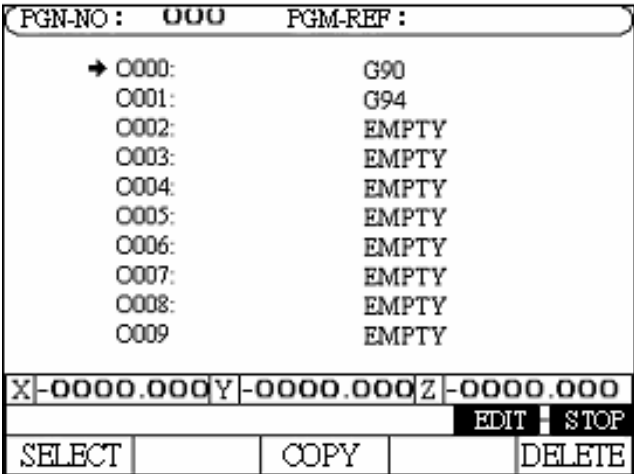


Fig.2-8

Program selecting methods:

1. Select Program:
 - a. Use cursor up and down or page up and down to select the program numbers.
 - b. Press the soft key "Select" or press enter key.
2. Program Note:
 - a. Use cursor up and down or page up and down to select the note numbers.

- b. Enter the English letter or number.
- c. Press enter key.
- 3. Program Delete:
 - a. Use cursor up and down or page up and down to select the delete numbers.
 - b. Press delete key, the dialogue box will appear to confirm your command. Press soft key YES or Y to clear the program. Press NO or N key to cancel the delete program.
- 4. Program Copy:
 - a. Press" copy" key, it shows as follows:

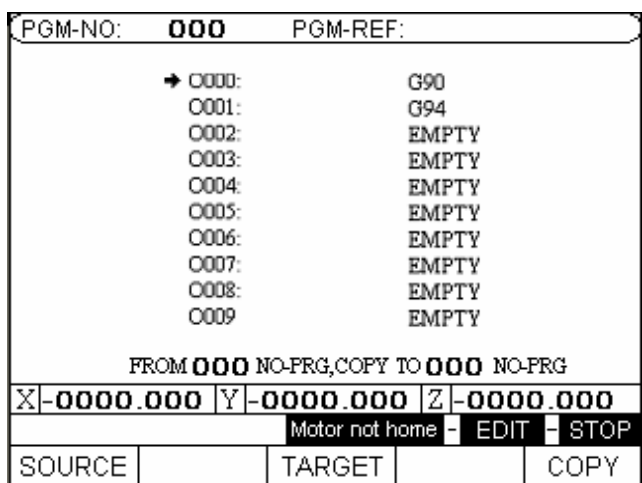


Fig.2-9

- b. Use cursor up and down or page up and down to point at the source program numbers.
- c. Press Source key
- d. Use cursor up and down or page up and down to select the purpose numbers.
- e. Press purpose key
- f. After confirmation for both source and purpose of program numbers, and press executing key. The copy is complete.

I/O Mode Display

Press twice 『GRAPH/ I/O』 key to enter I/O mode, the display is shown below:

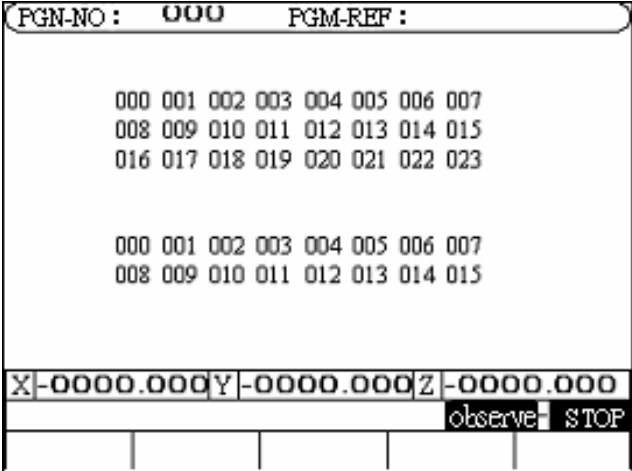


Fig.2-10

000-023 is the input status. (Color reversion shows inputting.)
 000-015 is the output status. (Color reversion shows outputting.)

Tool Compensation Display

Press 『T.Radius / T.Offset』 to enter wear compensation directly in the tool compensation mode.

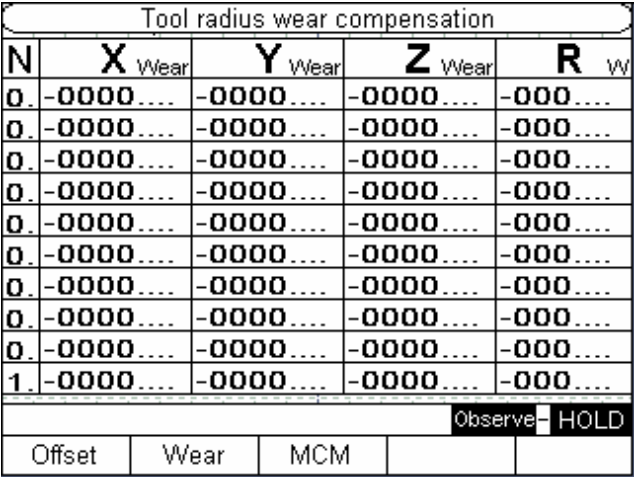


Fig.2-11

Users can utilize the soft key to switch three different screen displays such as tool wear, offset compensations and parameters under this mode. Note that the icon of the cursor blinks the page can be changed.

1. Ways for parameter setting in tool wear compensation are as follows:
 - a. Utilize the cursors to move to the revising parameter.
 - b. Enter numbers.
 - c. Press enter key.
2. Tool offset compensation display is below:

Tool radius wear compensation				
N	X ^{Wear}	Y ^{Wear}	Z ^{Wear}	R ^W
0	-0000	-0000	-0000	-000
0	-0000	-0000	-0000	-000
0	-0000	-0000	-0000	-000
0	-0000	-0000	-0000	-000
0	-0000	-0000	-0000	-000
0	-0000	-0000	-0000	-000
0	-0000	-0000	-0000	-000
0	-0000	-0000	-0000	-000
0	-0000	-0000	-0000	-000
0	-0000	-0000	-0000	-000
1	-0000	-0000	-0000	-000

				Observe	HOLD
Offset	Wear	MCM			

Fig.2-12

- Tool offset compensation setting are as follows:
- a. Utilized the cursors to move to the revising parameter.
 - b. Enter numbers.
 - c. Press enter key.
3. Parameters display is followed:

Parameter	X-AXIS	Y-AXIS	Z-AXIS
Resolution-den	0000000	0000000	0000000
Resolution-num	0000000	0000000	0000000
Encoder factor	0	0	0
Traverse speed	0000000	0000000	0000000
Rotate direction	0	0	0
Home direction	0	0	0
Home speed-1	0000000	0000000	0000000
Home speed-2	0000000	0000000	0000000
Software OT(+)	-00000...	-00000...	-00000...
Software OT(-)	-00000...	-00000...	-00000...
Backlash Comp	0000000	0000000	0000000

					<input checked="" type="checkbox"/>
Offset	Wear	MCM			

Fig.2-13

Note that parameter reversion can cause a serious damage to the machine without professional guidance.

● Graph Mode Display

Single press the key “ Graph/ I/O “ to enter the graph mode display as follows:

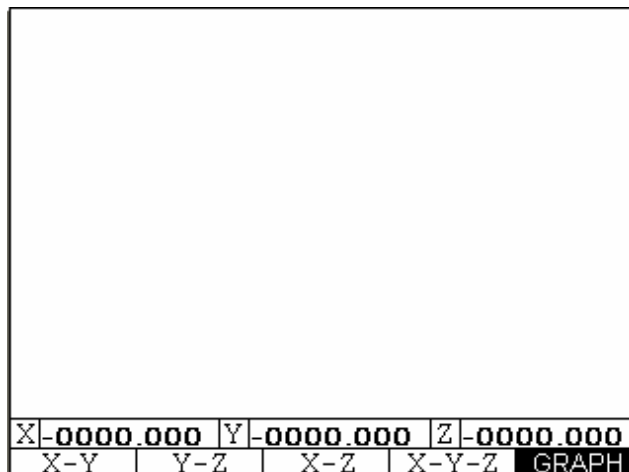


Fig. 2-14

Sign “ + “ in the center of the display shows the zero location. Use cursors to move the zero location.

On the bottom right corner of the screen shows the current coordinate. There are 5 selections to choose 3 different coordinates such as X-Y, Y-Z, X-Z, X-Y-Z.

Number 123 on the bottom right corner of the display shows the current graph of horizontal ratio setting. The ratio can also be changed by page up and down.

The image can be cleared by press the “ CLEAR “ key.

Note that it will automatically switch to auto mode after entering graph mode to operate programs.

2.2 Programming Introduction

2.2.1 A Part Program

Prior to cutting a machine part by using a CNC cutting tool, a computer program, called a part program, must be created to describe the shape of the parts, which is based on some kind of coordinate system. The cutting tool will then follow these coordinates to do exact cutting. To create a part program, a

concise machining plan is a necessity, which includes the coordinates for the machine part, coolant, spindle speed, tool type, I/O-bit, etc.. When design a machining plan, the following factors must be considered:

1. Determine the machining range requirement and select the suitable CNC machine tool.
2. Determine the work-piece loading method and select the appropriate cutting tool and the tool holder.
3. Determine the machining sequence and the tool path.
4. Determine the cutting conditions such as spindle speed (S), federate (F), coolant, etc.

A part program is a group of sequential instructions formulated according to the machining plan. It can be edited either on a personal computer (PC), then transmitted to the CNC controller through RS232C interface or directly on the CNC controller using the editing keys. Lathe can do both. They will be discussed later.

2.2.2 Methods Of Programming

A CNC controller will execute the commands exactly in accordance with the instructions of the part program. So, the program design is the most important task in the whole CNC machining process. There are two ways to design a CNC part program and are to be briefly described as bellows:

I. Manual Programming

Manual programming is a process that the whole process is manually done by hand including the coordinate calculations. It follows this sequence.

- ◆ Machine part drawing.
- ◆ Part shape description includes coordinate calculations.
- ◆ Computer program design includes spindle speed, feed rate, M-code, etc..
- ◆ Keying in the program instructions into the CNC controller or transmitted from PC.
- ◆ Testing the program.

The coordinate calculation is a simple process if the part shape is composed of straight lines or 90-degree angles. For curve cutting, however, the calculation will be more complicated and trigonometry will be required for correct answers. Once all calculations have been completed, the CNC part program is written in the formats to be discussed later.

The main disadvantage of manual programming, particularly when designing for a very complicated part, is time consuming and prone to making errors. In this case, automatic programming becomes more advantageous than the manual methods.

II. Automatic Programming

Automatic programming is a process in which the design work included coordinate calculation that is done by computer. It follows this sequence.

- ◆ Computer added design for part drawing (CAD)
- ◆ Computer added manufacturing for CNC part program (CAM)
- ◆ Transferring program to CNC controller.
- ◆ Testing the program.

By making use of computer's high speed calculating capability, program designer can communicate with the computer in simple language, to describe the shape, size and cutting sequence of the part. The computer will transfer the motions to the machine tool into a part program, which is then transferred into CNC controller through RS232C interface. This process is called CAD/CAM. It is a necessary tool when designing a part program for a 3-D work-piece.

2.2.3 The Composition of A Part Program

A complete part program is composed of program blocks, starting with a program number Oxxx, ended with M2, M30, or M99, and in between with a series of CNC instructions. A CNC instruction is a command to order the cutting tool to move from one location to another with the specified speed, or the peripheral equipment to do some mechanical work. The cutting is done when the cutting tool moves.

An example of a complete part program containing 10 blocks is as follows:

```

N10 G00 X0.000 Y0.000 Z0.000
N20 M3 S1000
N30 G01 X10.000 Y10.000 Z10.000 F200
N40 X20.000
N50 Y20.000
N60 Z20.000
N80 G0 X0.000 Y0.000 Z0.000
N90 M5
N100 M2
    
```

A block of program can have one to several instructions and it has a general form as follow. The block sequence number "Nxx" can be omitted. If you do not key in the block number, Lathe has a special function "Auto-N" to automatically generate the number for you during or after program editing (see chapter 6). The program execution starts from top to bottom block and has nothing to do with the order of block sequence number. Each instruction starts with an English letter (A~Z), followed by a integer or floating number, depending on what type of instruction the number is associated with. If the number represents a coordinate, it can be positive (+) or negative (-).

N ___ G ___ X (U) ___ Y(V) ___ Z(W) ___ F ___ S ___ D(H) ___ M ___

N: block sequence

G: function command

X, Y, Z : coordinate position command (absolute position command)

U, V, W : coordinate position command (incremental position command)

F: Feed rate

S: Spindle speed

D, H: Tool numbers.

M: Auxiliary command (machine control codes)

In general, the program instructions can be divided into four categories.

1. Function command :G-code. A CNC command to instruct the cutting tool to do an action, such as straight, circular or thread cut, compound cut, etc.
2. Position command: X,Y, Z, U,V, W--a coordinate command to instruct the cutting (Motion command) tool to stop the cutting action at the location

specified -- an end point. The end point of the current block is the starting point of the next block.

3. Feed-rate command: F-code. A command to instruct the cutting tool how fast to do the cutting.
4. Auxiliary command :M, S, D, L, etc. A command to instruct the peripheral equipment to do an action, such as spindle speed, coolant on/off, program stop, etc.

Note that not every block is composed from these four parts. Some have only one command. We will have further discussions in chapter 3. In the single block format, most of them are command functions except the series number "N____" of single block. Command function is composed of letters, \pm signs and numbers.

Basic command format (similar with position command):

X-10.000

X : command code

"-" : positive and negative signs(sign + can be omitted)

10.000 : destination point for tool position

Each command code has a fixed format and a special meaning to the CNC controller and it must be strictly followed when designing a program. The system will not accept the command if the format is in error. Otherwise, a machine error will result. Followings are the command codes that are used in H4M.

D :Tool radius compensation number.

F : Feed-rate in mm/min or mm/revolution, a decimal.

G : Function G-code, an integer.

H : Tool offset compensation number.

I : The X-axis component of the arc radius @ the start point, a decimal.

J : The Y-axis component of the arc radius @ the start point, a decimal.

K : The Z-axis component of the arc radius @ the start point, a decimal.

K, L : Repetition counter, integer.

M : Control code for peripheral machine tool, integer.

N : Program block (sequence) number, integer.

- O :Program numbers.
- P : Dwell time; subprogram code; or parameter in canned cycles, integer.
- Q : Parameter in canned cycles, integer.
- R : Arc radius or "R" point in canned cycles, decimal.
- S : Spindle speed, integer.
- U : Incremental coordinate in X-axis, decimal.
- V :Incremental coordinate in Y-axis, decimal.
- W : Incremental coordinate in Z-axis, decimal.
- X : Absolute coordinate in X-axis, decimal.
- Y :Absolute coordinate in Y-axis, decimal.
- Z : Absolute coordinate in Z-axis, decimal.

In program making, each single block has fixed format. Program designing must follow the format. If the format is wrong, the system will not accept. Otherwise, it will cause a great error.

Each serial number of program represents a block. Although it is not necessary to use it, it is recommended to utilize the serial numbers for program searching. Mill has a special function "Auto-N" to automatically generate the number for you during or after program editing (see chapter 6). The program execution starts from top to bottom block and has nothing to do with the order of block sequence number.

Example: N10.....(1) program execution order
 N30.....(2)
 N20.....(3)
 N50.....(4)
 N40.....(5)

2.2.4 Coordinate System

The machining action of a cutting tool is accomplished when the tool is moving along a specific path from point A to point B, which represents the shape or the contour of a machine part. In order for the tool to follow the specific path, a computer program describing the shape of the machine part must be created and the shape or the contour is described by the Cartesian coordinate system.

I. Cartesian Coordinate System

H4M uses the customarily 2-D/3-D Cartesian coordinate system as shown in Fig 2-16, with Z-axis being the center of and parallel to the spindle axis and defined as $x=0$. The other axis is X-axis and $Z=0$ can be anywhere along the Z-axis at some convenient location for coordinate calculation. The intersecting point of the two axis is the origin, $X=0, Z=0$. Depending on the location of the cutting tool with respect to the spindle axis, the sign convention of the coordinate system is shown in Fig 2-15

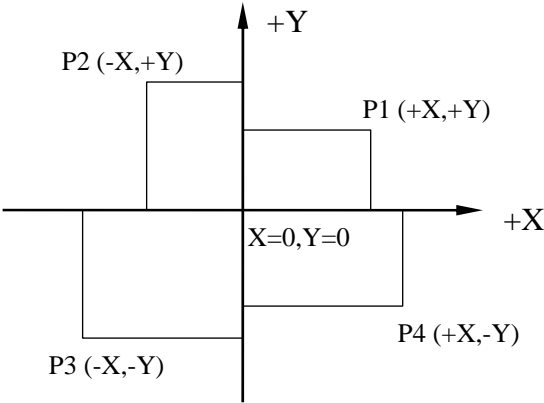


Fig 2-15

Fig 2-15 Cartesian Coordinate System of CNC Mill

Fig 2-16 is 3-D system (right-hand rule) with the intersecting point designated as origin $X=Y=Z=0$. The direction of normal rotation for each axis is indicated by the direction of the four fingers when you grab the axis by the right hand with your thumb pointing to the (+) direction of that axis.

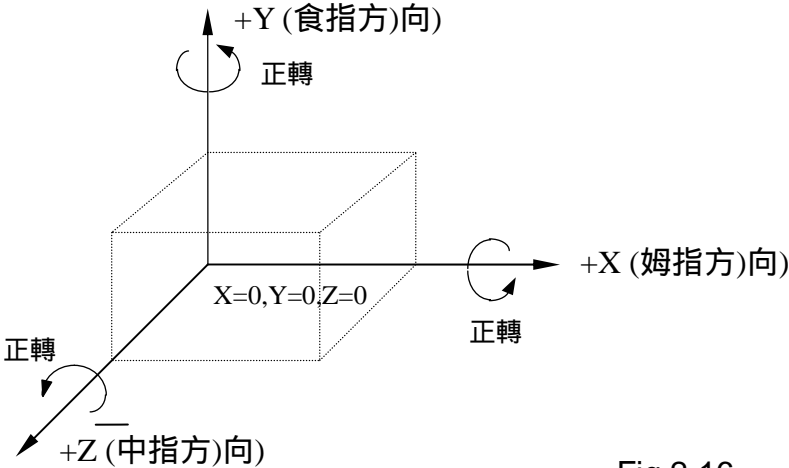


Fig 2-16

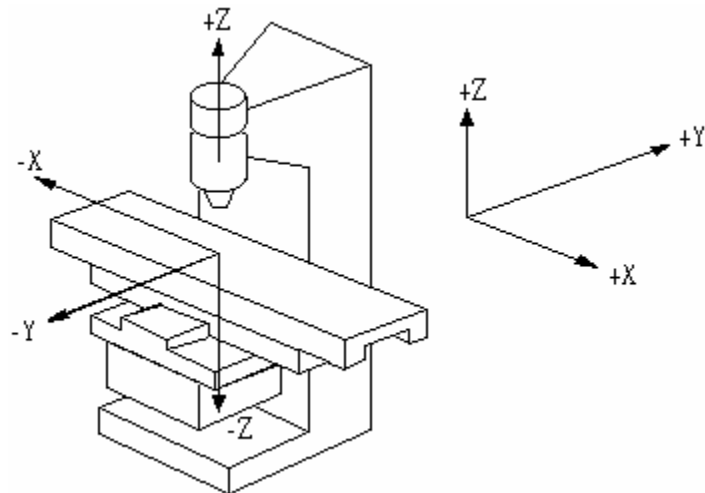


Fig 2-16-1

II. Coordinate of Tool Position Command (Coordinate)

The instruction for tool position command in H4M series can be in either absolute coordinate or incremental coordinate as follows:

X, Y, Z: Absolute coordinate command. The cutting tool moves to the position specified by the absolute coordinate X, Y, Z.

U, V, W: Incremental coordinate command. The cutting tool moves to the position with an incremental amount specified by U, V, W.

III. Absolute Coordinate

The origin is the reference. The coordinates of all points describing the shape of the work-piece (machine part) are calculated from the origin. The coordinates can be positive (+) or negative (-), depending on its relative position with respect to the origin.

IV. Incremental Coordinate

The coordinates of all points describing the shape of the work-piece (machine part) are calculated from the end point of the previous block. They are the amount of coordinate increase from the last point. The incremental coordinates can be either positive (+) or negative (-), depending on its relative position with respect to the end point of the previous block. They are positive (+) if the

cutting tool is going in the direction of U, V, W increment, negative (-), otherwise is in the direction of U, V, W decrement.

X, Y, Z, U, V, W can be mixed in the program. The methods are described below:

Absolute Command:

N10 G00 X0.000 Y0.000	Position to work origin
N20 G90	Set program to absolute value
N30 G1 X12.000 Y12.000 F300.00	P0 to P1
N40 X26.000 Y16.000	P0 to P2
N50 X38.000 Y30.000	P2 to P3
N60 M2	Program end

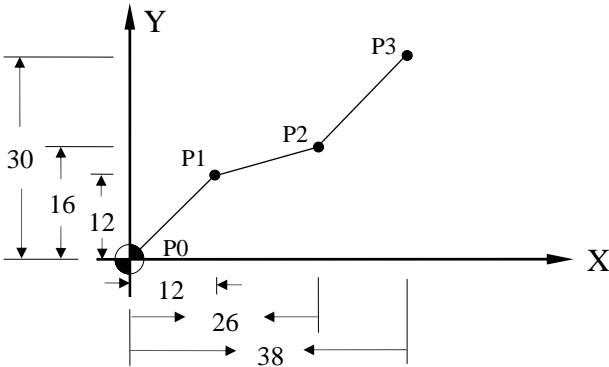


Fig.2-17 Absolute Command

Increment Command:

N10 G00 X0.000 Y0.000	Position to work origin
N20 G91	Set program to increment value
N30 G1 X12.000 Y12.000 F300.00	P0 to P1
N40 X14.000 Y4.000	P0 to P2
N50 X12.000 Y14.000	P2 to P3
N60 M2	Program end

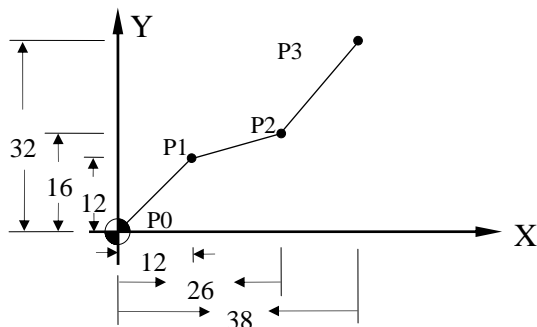


Fig.2-18 Increment Command

Please note that the command G91 sets the X, Y, Z-coordinate as incremental values and the U, V, W will be rendered ineffective with G91. The increment command U,V, W can only be rendered effective with G90.

The power-on setting of H4M series is in absolute mode, the above program can be re-written as:

N10 G00 X0.000 Y0.000	Position to work origin
N20 G1 U12.000 V12.000 F300.00	P0 to P1
N30 U14.000 V4.000	P1 to P2
N40 U12.000 V14.000	P2 to P3
N60 M2	Program end

Mixed Usage:

N10 G00 X0.000 Y0.000	Position to work origin
N30 G1 U12.000 V12.000 F300.00	P0 to P1
N40 U26.000 V4.000	P1 to P2
N50 U38.000 V14.000	P2 to P3
N60 M2	Program end

Or

N10 G00 X0.000 Y0.000	Position to work origin
N30 G1 U12.000 V12.000 F300.00	P0 to P1
N40 U14.000 V16.000	P1 to P2
N50 U12.000 V30.000	P2 to P3
N60 M2	Program end

In the absolute coordinate, the calculation error of one point will not affect the positioning of next point. In the incremental coordinate, however, an error of a point will affect the positioning of all subsequent points. There isn't any rule as to when to use the incremental or the absolute coordinate. If the each point and home are related, it is better to adopt absolute coordinate.

In the command of diagonal or circular movement, it is better to notice the matter of errors in the increment system under the result of trigonometry in each axis value. Basically, we could follow the analysis of part sizes and the need of program design to adopt absolute or increment coordinate.

V. Work Origin/Work Coordinate

The work origin is the coordinate origin as described before. It is also called the program origin. This is the reference point for all coordinate calculations and the coordinate so obtained is called work coordinate. The selection of work origin location needs to be determined by program designers. It can be any point on the table of Mill or tool chucks. Using the coordinate of the part diagram is easier to read.

VI. Machine Origin

The machine origin is the HOME location for the cutting tool. This is the reference point for the coordinate determination of the work origin and the tool offset compensation. The coordinate obtained using the machine origin as calculation base is called the machine coordinate.

The exact location of the machine origin is determined by the location of the home limit switch on each axis. When user executes X,Y , Z Home on a Mill CNC controller, the cutting tool will move to the machine origin. The exact distances between the machine origin and the work origin must be accurately measured using a fine instrument, such as a linear scale. Otherwise, the completed part will be in an error.

When the electric power is interrupted for any reasons, execute HOME on each axis before resuming any cutting.

2.2.5 Control Range

The minimum/maximum programmable range for Mill CNC controller is as follows. Please note that the control range may be limited by the working range of user's machine.

	Metric, mm
Min. setting unit	0.001
Max. setting unit	8000.000
Min. moving unit	0.001
Max. moving unit	8000.000
Min. Time	0.1
Max. Time	8000.000

	Metric Unit / English Unit
G-code	G00~G99 (G01=G1)
M-code	M00~M999 (M01=M1)
S-code	S1~S9999 rpm
F-code	0.01~8000.00mm/spin
X, Z, U, W, I, K, R	0.001~+/- 8000.000 mm
R (Radius)	0.001~+/- 4000.000 mm
G04	0~8000.000 seconds
Memory capacity	128 K
Lead screw compensation	0~255 pulses (related to tool resolution)
Max. Response Speed	500 KPPS

2.3 Program Editing

The following topics will be discussed in this section.

1. Select a program for editing.
2. Edit a new program.
3. Revise an existing program.

2.3.1 Program Selection

H4M controller can store a maximum of 999 programs with number O0~O999. You can select any one of the programs for editing or execution. The program selection process is described as follow.

Press 『EDIT/PRNO』 key twice in 0.5 seconds to enter PRNO mode, move the cursor to the desired program and press the input key. The LCD display is shown as

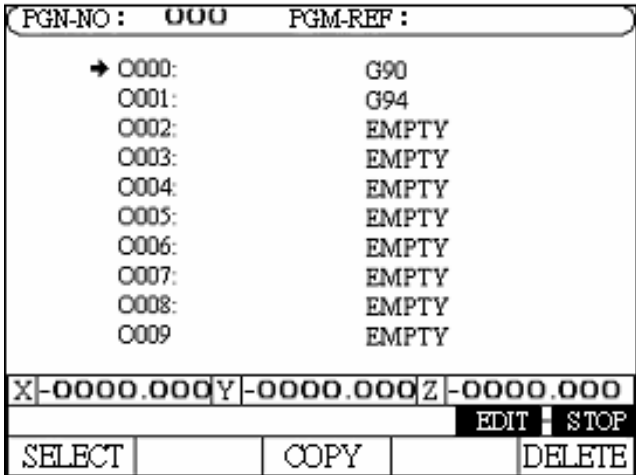


fig.2-20.

Under PRNO mode, the program note can be entered up to 12 different letters and numbers.

Example: If you put the note “TYPE-201” after 001, the instruction is as follows.

1. Move the cursor to 0001
2. Enter the letters and numbers as T Y P E 2 - 0 1
3. Press input

2.3.2 New Program Edition

When a new program has been selected, press EDIT key to be in editing mode. The LCD screen will be blank with cursor pointing at the first line to be entered as in Fig 2-21.

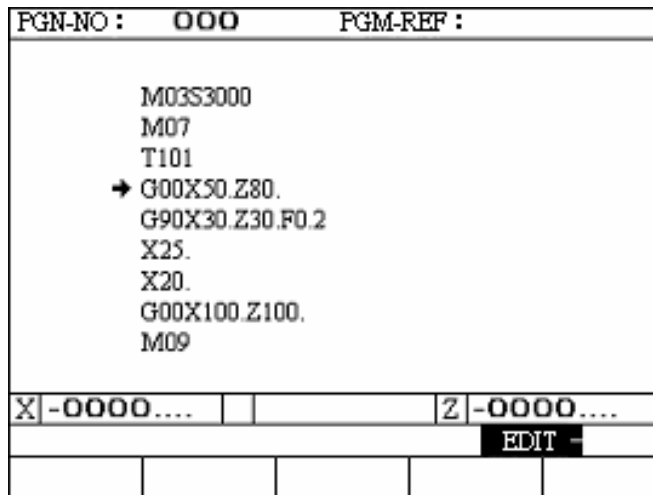


Fig.2-21

During program editing, the following keys will be used.

1. Function keys.
2. Numeric keys, 0~9
3. CURSOR ← and CURSOR → keys for data inspection in the same block.
4. PAGE↑ and PAGE↓ keys for data inspection between lines.
5. NEW LINE key -- Establishing or inserting a new block anywhere in the program.
Key in a function code, then press NEW LINE to establish a new line.
6. INPUT -- For entering a data or a function in the established block.
Key in a function code, then use INPUT to enter more data into the established line.
7. DEL -- For deleting a block (line) of program.

You can edit a program with or without block number. Following is an **example program** to explain the keystrokes required to edit a new program in the controller.

Ex: Program 1
 N10 G0 X0. Y0.
 N20 G4 X1.
 N30 G0 U480. V-480.
 N40 G4 X1
 N50 M99

Keystrokes: (Ignore the sign "-" below. It's there for clarity)

1. Please confirm the edit status and press Edit key to enter in the controller.
N1 G0 X0. Y0.
2. Enter first block information G – 0 – NEW LINE

It is a new establishing block. Thus, users need to enter NEW LINE key.
After this step, the LCD screen is shown as Fig 2-22.

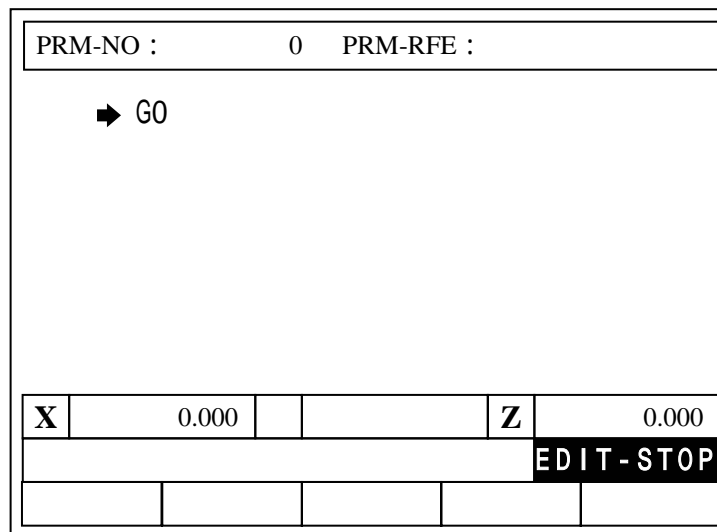


Fig.2-22

And enter:

X 0 INPUT

Y 0 INPUT

Key-strokes for the remaining blocks are as follows.

2. N20 G4 X1.
 - (A) G - 4 - NEW LINE
 - (B) X - 1 - ● - INPUT

3. N30 G0 U480. W-480.
 - (A) G - 0 - NEW LINE
 - (B) U - 4 - 8 - 0 - ● - INPUT
 - V- "-" 4 - 8 - 0 - ● - INPUT

(The negative sign "-" here can be input anywhere before pressing INPUT key)

- 4. N40 G4 X1.
 - (A) G- 4 - NEW LINE
 - (B) X - 1 - ● - INPUT

- 5. N50 M99
 - (A) M - 99 - NEW LINE

During program editing, you can use CURSOR ←, CURSOR → key to check the input data within the block. Use PAGE↑, PAGE↓ to move up and down the block (line). When you finish editing the entire program, press RESET key to exit.

2.3.3 Old Program Revision

Let's use Program O001 of previous section as our example for program revision.

Revise or Add a Function

To revise or add a function, simply key in the function code and the correct number, then press INPUT key.

Ex: Revise N3 U480. V-480.
 To N3 U480. V-480. F300

1. Make sure the system in EDIT mode.
2. Use PAGE↑, PAGE↓ key to move cursor to N3 block.
3. Add a function of F300 by entering data below and LCD will display as in

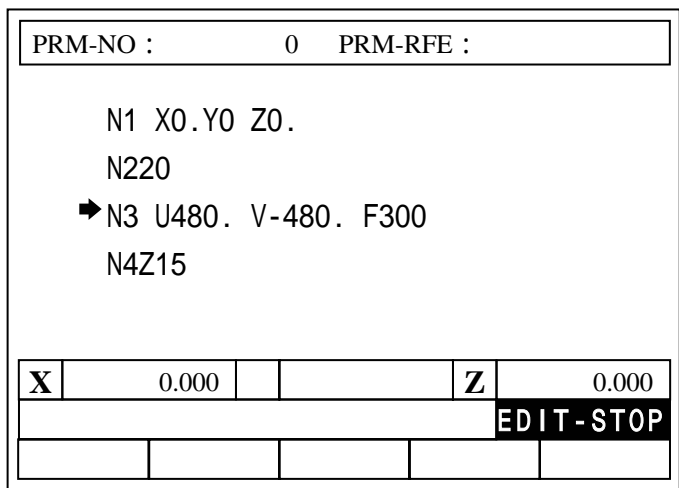


Fig 2-23

F- 3 0 0 - INPUT

- 4. Revise U480. to U360. by keying in

U - 3 - 6 - 0 - • - INPUT

Delete a Function

To delete a function, simply key in the function to be deleted without number, then press INPUT key.

Ex: Revise N30 U480. V-480. F300
 To N30 U480. V-480.

- 1. Make sure the system in EDIT mode.
- 2. Use PAGE↑, PAGE↓ key to move cursor to N3 block.
- 3. Key "F" without numbers and press INPUT key, LCD displays as Fig 2-24.

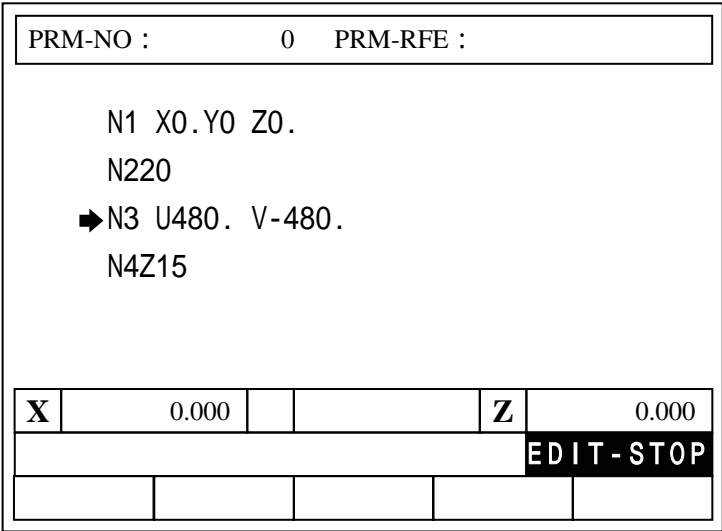


Fig.2-24

Insert a Program Block

To insert a program block, key in the block number (or any function) and use NEW LINE key to establish the block. Then use INPUT key to input the rest of data for the block.

Ex: Insert N31 U20. V-20.

between N3 G0 U480. V-480. and
N4Z15

1. Make sure the system in EDIT mode.
2. Use PAGE↑, PAGE↓ key to move cursor to N30 block.
3. Enter
N 3 1 new line
U 2 0 . input
V- 2 0 . input

The LCD display is shown as fig.2-25.

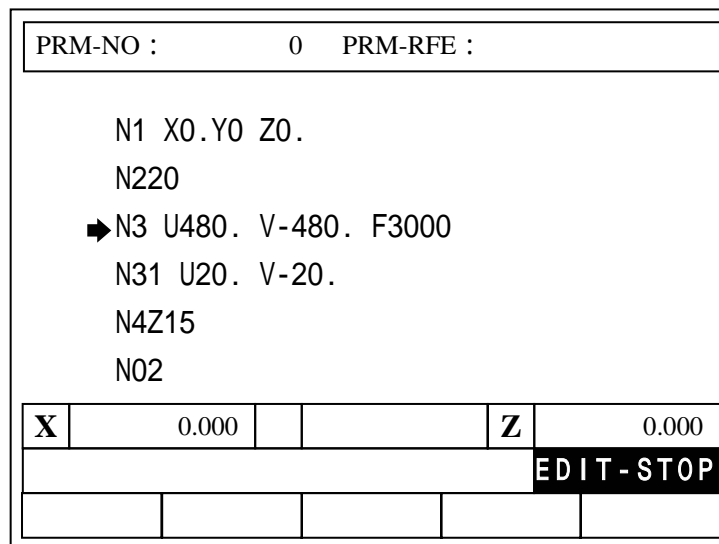


Fig 2-25

Delete a Program Block

To delete a block, use PAGE↑, PAGE↓ key to move cursor to the block that you want to delete and press DEL key. For example: Delete N31 U480 V-480. from last example.

1. Make sure the system in EDIT mode.
2. Use PAGE↑, PAGE↓ key to move cursor to N31 block.
3. Press DEL key and the LCD display is as shown in Fig 2-26 (Block N4)

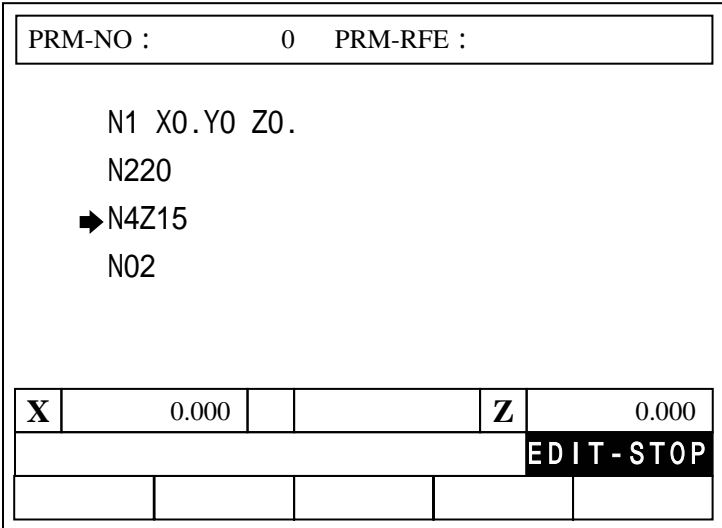


Fig.2-26

2.3.4 Delete a Program

Move the cursor to the program that you want to delete it in PRNO mode and press DEL. The LCD display is shown as fig.2-27

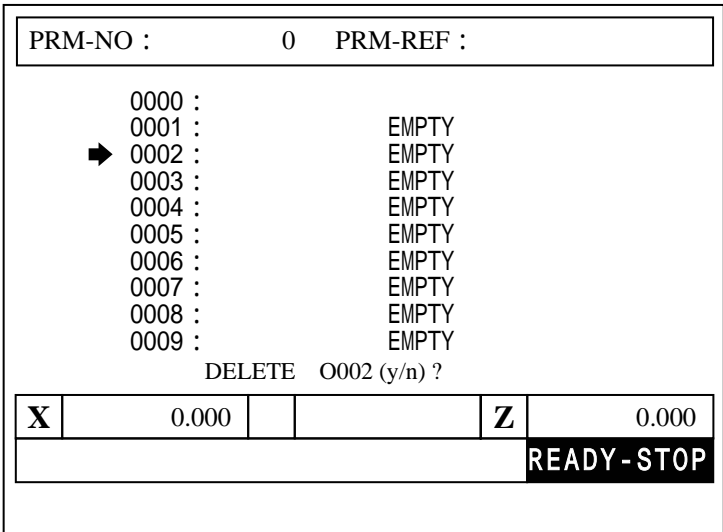


Fig.2-27

In the meantime, press Y and clear the content of the 002 program. The key N remains the same.

If you want to delete all programs- 0~999, follow the procedures below:

Enter MDI mode, and give G10 P2001 command.

Then all the content of the program are cleared immediately.

Note: After completing the procedure, all the program data in memory will be vanished. Therefore, do not use this program if it is not necessary.

2.3.5 Decimal Points Entering

Numerical input has two formats such as integer and decimal with a maximum of 7 digits. If you input the numbers in accordance with the format required by the controller, the number will be entered correctly. You cannot enter a decimal point for a number that requires an integer format. So, the only occasion that may cause error input is the one that you enter an integer for a decimal format. Described more in detail below.

The decimal input such as X, Y, I, J is left blank, the content of the controller will automatically move back to the decimal points of last format with dot at front. The table below shows the decimal numbers recognized by the controller after internal process for some integer inputs.

Input	4/3 Format
X2	X0.002 mm
Y250	Y0.0250mm
Z35	Z0.035mm
U2500	U0.500 mm
V25	V25.0000mm
W125.	W125.000mm
F300	F0.3 mm/min

The numerical formats for the function codes used in Lathe system are listed below. To avoid any potential error, please use the specified format as follow when key in data. The number "0" after decimal point can be omitted.

G, M, N, S-code: Variables	Integer input
X, Y, Z, U, V, W, I, J-code	Decimal input
F-code	Integer input

Note: TO avoid the confusion, apart from integer inputs such G, M, N, S, the rest of the inputs should be entered by decimal points. The number "0" after decimal point can be omitted.

2.3.6 Notes For Edition

Program Block Number

1. Block number N can be omitted, but it's better to have it for the convenience of program inspection later.
2. Block number N is recognized by the editing order not by the block sequence or its value. The numbers by the letter N are merely symbols. For instance, inserting block N35 in Block N30. It will become the following result.

Program 1

```

N10 G0 X0 Y0 .....first block
N20 G4 X1 .....second block
N30 U480 V-480 .....third block
N35 U20 V-20 .....fourth block
N40 G4 X1 .....fifth block
N50 M99.....sixth block

```

If block N35 is changed to block N350, the arrangement of program execution remains the same.

3. Block number is recognized by the number of characters, not by its value. Therefore, N10, N010, N0010 are three different block number.

Program Block

1. Do not use two G-codes in the same block. If more than one G-code exists in a block, only the last one is effective.
2. Do not repeat any position code in the same block. The position codes are X, Y, Z, U, V, I W, J and R.
3. If you specify absolute coordinate and incremental coordinate for the same axis in a block, only the incremental coordinate will be executed. Example: G1 X100. U50. -- U50 will be executed.

- Do not exceed 80 bytes of data input for a single block. Otherwise, the CNC controller will show an error message Err-18 at the bottom of the screen.

2.3.7 Circular Entering

There are three functions on the Edit Screen. These are single drilling, grooving and canned cycle.

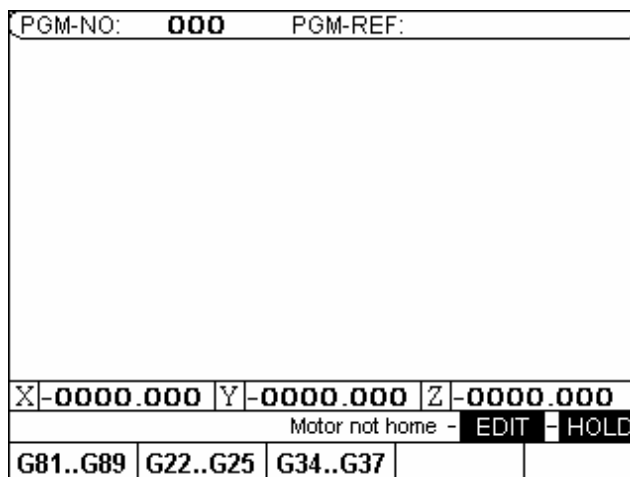


Fig.2-28

2.3.7.1 Single Drilling

After selecting single drilling, the next screen display shows below:

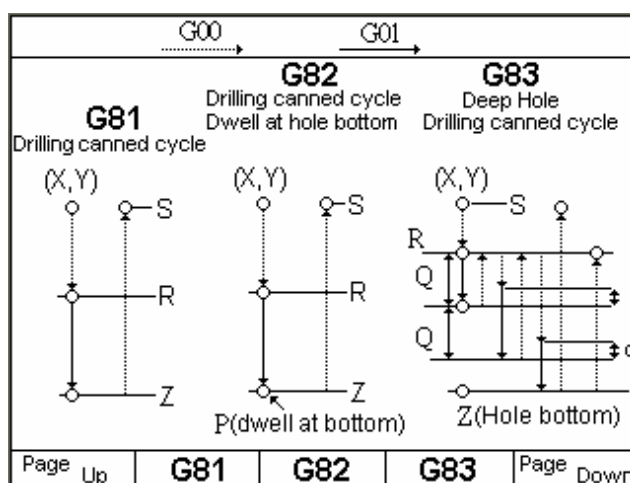


Fig.2-29

Follow the direction of the figure and choose the required function to enter (G81, G83, G85, G86, G89). For example: choose G81 to enter the display below:

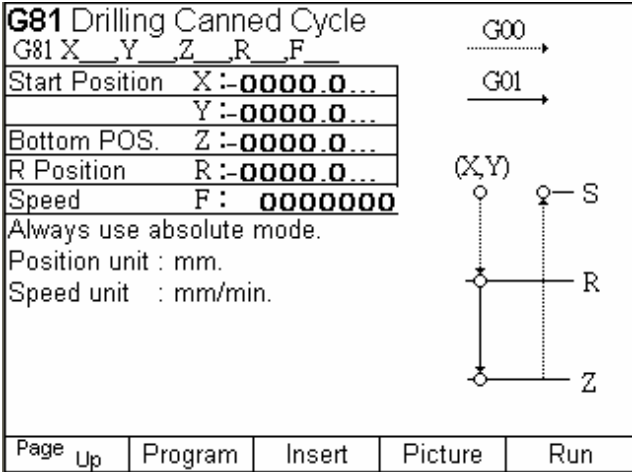


Fig.2-30

And enter the required input value from the format. After the confirmation, and press < Program Newline >. The controller can automatically create a new program from new format values. Use the key “ Program Check” to review the information and check whether there is a program being created.

If the user needs to execute the drilling function directly, press “Direct Execution” key for 3 seconds, it will switch to the auto mode and execute this single step (Non Edit Program). After the execution, it will switch back to the display again.

The figure on the right is a brief format. The user can follow the figure and enter parameters.

Please check the G-functions and descriptions in the format.

2.3.7.2 Grooving

After selecting the function, display shows below:

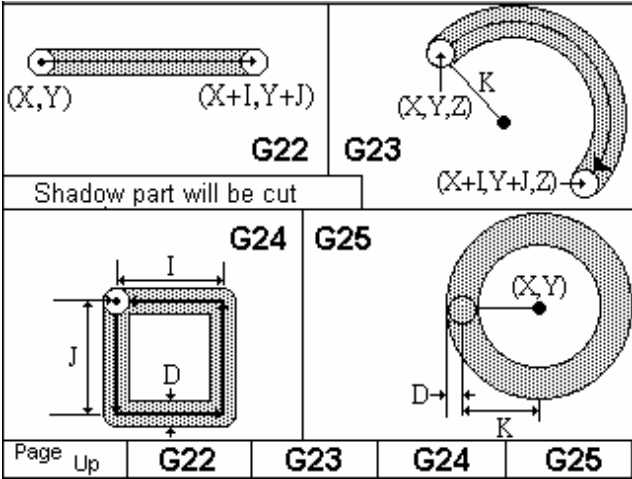


Fig.2-31

Follow the direction of the figure and choose the required functions. For example: choose the circular grooving to enter the display below:

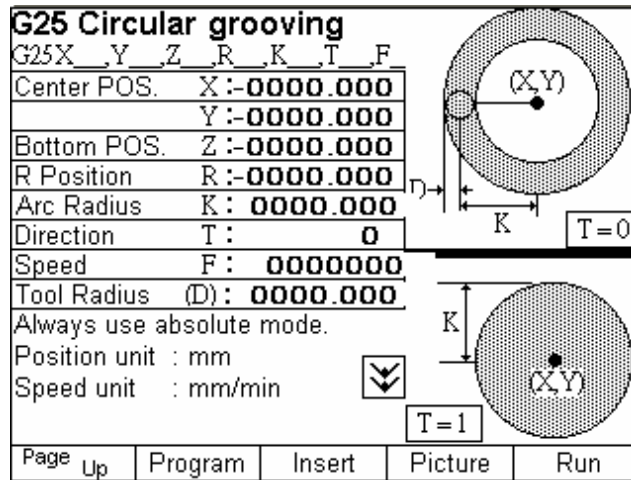


Fig.2-32

And enter the required input value from the format. After the confirmation, and press < Program Newline >. The controller can automatically create a new program from new format values. Use the key “ Program Check” to review the information and check whether there is a program being created.

If the user needs to execute the drilling function directly, press “Direct Execution” key for 3 seconds, it will switch to the auto mode and execute this single step (Non Edit Program). After the execution, it will switch back to the display again.

The figure on the right is a brief format. The user can follow the figure and enter parameters.

Please check the G-functions and descriptions in the format.

2.3.7.3 Canned Cycle

After selecting < canned cycle>, the display shows below:

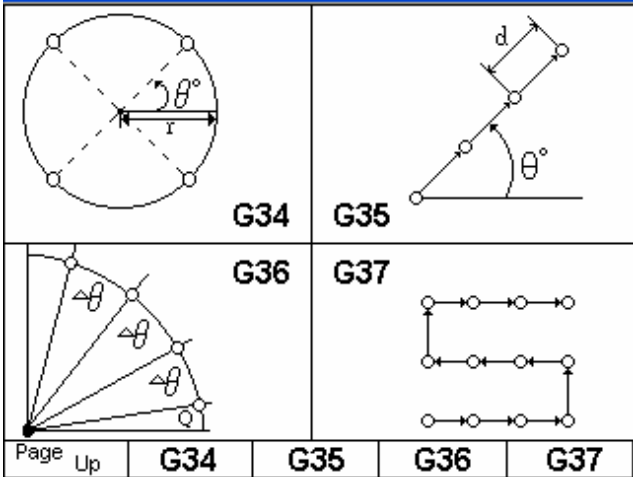


Fig.2-33

Following the direction of the diagram and choose the required function. For example: choose the grid of canned cycle to enter the display below:

G37 Grid drill canned cycle	
G37 X_Y_Z_R_I(ΔX)_P(nx)_ J(ΔY)_K(ny)	
Start Position	X: -0000.0... Y: -0000.0...
Bottom POS.	Z: -0000.0...
R Position	R: -0000.0...
Hole Interval	ΔX : -0000.0...
Line Interval	ΔY : -0000.0...
Amount / Line	nx: 0000000
Total Line	ny: 0000000
Speed	F: 0000000
Speed unit : mm/min	
Page Up	Program Insert Picture Run

The diagram on the right shows a grid of points with horizontal and vertical lines. The horizontal distance between points is labeled ΔX and the vertical distance is labeled ΔY . The total number of points in a row is labeled nx and the total number of rows is labeled ny . A specific point is labeled (X, Y) .

Fig.2-34

And enter the required input value from the format. After the confirmation, and press < Program Newline >. The controller can automatically create a new program from new format values. Use the key “ Program Check” to review the information and check whether there is a program being created.

If the user needs to execute the drilling function directly, press “Direct Execution” key for 3 seconds, it will switch to the auto mode and execute this single step (Non Edit Program). After the execution, it will switch back to the display again.

The figure on the right is a brief format. The user can follow the figure and enter parameters.

Please check the G-functions and descriptions in the format.

3 PREPARATORY FUNCTION (G FUNCTION)

This chapter will discuss the meanings of command codes, such as G, F, M and S-code, and the format of their usage.

3.1 Explanations

G function followed by one (1) or two (2) numbers are special command codes in Mill system and they are ranging from G00 to G99. The first "0" can be omitted. Each G function has its own specific function as shown in Table 3-1. G-code commands are divided into two groups:

1. One-shot G-codes

A One-shot G-code is effective only in the program block where it was encountered. Once program starts executing the next block, it's no longer effective.

Example:

```
N10 G0 X30.000 Y40.000  G0 is mode command
N20 G4 X2.000          .....G04 is one-shot G-code, effective only in this block.
N30 X20.000 Y50.000   .....G04 no longer effective in this block. G0 is.
```

2. Modal G-codes

A Modal G-code is a G-code that remains effective until another G-code in the same group is encountered. In Mill, following G-codes are in the same group.

G00, G01, G02, G03	Same group
G17, G18, G19	Same group
G40, G41, G42	Same group G43, G44, G49 belongs to the same group.
G54~G59	Same group
G80, G89	Same group G90~G91 belongs to the same group.
G98, G99	Same group

Example:

```
N10 G0 X30.000 Z5.000      .... G0 is effective in this block.
N20 X50.000 Z10.000      ....No G-code specified, G0 remains effective.
N30 Y20.000              No G-code specified, G0 remains effective.
N30 G1 X30.000 F200      .... G1 is effective from this block, NOT G0.
```

Normally, only one G-code is allowed in a program block. If several G-codes are accidentally specified in a block, only the last G-code specified is effective.

Example:

```
G00 G1 X10.000 .....Only G01 is effective.
```

Table 3-1 G-Code Definitions

G-code	Function
* 00	Rapid positioning (fast feed-rate)
* 01#	Linear cutting (cutting feed-rate)
* 02	Arc cutting, CW (cutter at rear)
* 03	Arc cutting, CCW (cutter at rear)
04	Temporary stop
08	Clear machine coordinates in all axis
10	Data input
*17	Pitch cutting, X-Y in 2 dimension
*18	Pitch cutting, Z-X in 2 dimension
*19	Pitch cutting, Y-Z in 2 dimension
*20	System measurement in INCH mode
*21	System measurement in METRIC mode
22s	Linear grooving
23s	Arc grooving
24s	Rectangular grooving
25s	Circular grooving
28	Tool moves to the 1st reference point
29	Moves back to the specified position from the ref. point
30	Tool moves to the 2nd reference point (10 groups in total)
31	Skip function
34s	Circular canned cycle
35s	Angular drill canned cycle
36s	Arc drill canned cycle
37s	Grid drill canned cycle
* 40#	Tool radius compensation - cancel
* 41	Tool radius compensation - set (left)
* 42	Tool radius compensation - set (right)
43	Tool length compensation- positive direction
44	Tool length compensation-negative direction
49	Tool length compensation-cancel
*54#	The first work coordinate
*55	The second work coordinate
*56	The third work coordinate
*57	The fourth work coordinate
*58	The fifth work coordinate
*59	The sixth work coordinate

*80\$	Drill canned cycle, cancel
*81\$	Drill canned cycle, set
*82\$	Drill canned cycle-temporary stop at bottom
*83\$	Deep hole drilling canned cycle
*85\$	Boring cycle
*86\$	Boring cycle-spindle stop at bottom
*89\$	Boring cycle-temporary stop at bottom
90	Command for absolute coordinate
91	Command for increment coordinate
<p># -- G-codes with "#" are of power-on default setting. * -- G-codes with "*" are modal G-codes. \$ -- G-codes with "\$" are special functions for H3XMA. % -- G-codes with "%" are optional functions.</p>	

3.2 Rapid Positioning (Traverse Speed)

Format: G00 X(U)____ Y(V)____ Z(W)____

X, Y, Z : Position code in absolute coordinate
 U,V, W : Position code in incremental coordinate

This code is used to move the cutting tool from the current location (or the end point of previous block) to the coordinate specified by X(U),Y(V) Z(W) at high speed while the tool is NOT physically doing any cutting. It can control the movement of 1~3 axes. The moving speed is based on the setting of MCM parameter #79(X-axis) ,#80(Y-axis) ,#81(Z-axis).

The position codes X(U), Y(V), Z(W) associated with G00 command are the target coordinate for the cutting tool to move to. The starting location is the one prior to G00 command or the current position. The moving path will be the resultant of the incremental coordinate between these two points. The position code can be positive (+) or negative (-).

When calculating the moving speed on each axis, the controller will use the longest distance as a base. If the speed on one of the calculated value exceeds the MCM setting value, the controller will use that MCM value to re-calculate the moving speed for the rest of the axes. Following is an example. In the figure 3-1 point S rapid positions to point E.

Caution -- To avoid damages to the cutting tool, be sure there is NOT any obstruction along the tool path when applying G00 command.

G00 Example:

G00 X100.000 Y50.000 Z20.000

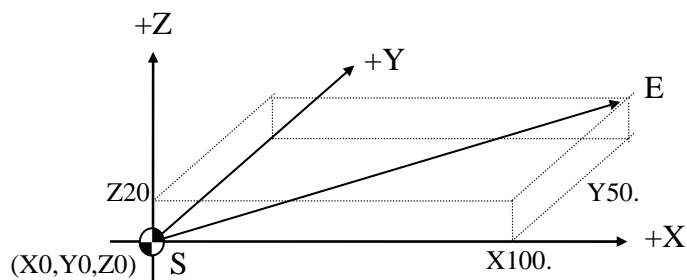


Fig 3-1 G00 Rapid Positioning

The target location is (100,50,20) with the start coordinate of (0,0,0). Therefore, the moving distances are 100 mm, 50mm, and 20mm in X, Y and Z axis, respectively. Assume that the rapid feedrate of MCM parameter 34 is 10000.00 mm/minute. Then, the moving speeds along each axis are:

X-axis – Since X-axis has the longest moving distance (100 mm), the moving speed will be 10000.00mm/minute as set in MCM parameter#79.

Y-axis – Moving distance is 50.000 mm. The moving speed along Y-axis as MCM# 80 will be calculated as (50 mm/100mm) × 10000.00=5000.00 mm/minute. Feed-rate for Y-axis is 5000.00 mm/minute.

Z-axis- Moving distance is 20.000mm. The moving speed along Z-axis as MCM# 81 will be calculated as (20 mm/100 mm) × 10000.00=2000.00 mm/min. Feedrate –rate for Z-axis is 2000.00 mm/min.

3.3 Linear Cutting (Cutting Feed), G01

Format: G01 X(U)____Y(V)____Z(W)____F_____

X, Y, Z : Position code in absolute coordinate
 U, V,W : Position code in incremental coordinate
 F : Cutting feed-rate

G01 is for the linear cutting motion and can control one to three axes at the same time. The cutting speed is determined by F-code. The smallest setting value for F-code is 0.02 mm/min or 0.2 in/min. F-code can be applied with any G-code including G00 block. However, the G00 moving speed will not be affected by F-code setting.

The feed-rate F-code is a modal code. If the cutting rate is a constant for all the program blocks, only one feed-rate in the beginning block needs to be defined and omit the rest. Unless the feed-rate is redefined, the F-code remains effective. The formula of cutting speed calculation for each axis is as below. U, V, W are incremental values.

$$\text{Feed-rate in X-axis,} \quad F_x = \frac{U}{\sqrt{U^2 + V^2 + W^2}} * F \quad (1)$$

$$\text{Feed-rate in Y-axis,} \quad F_y = \frac{V}{\sqrt{U^2 + V^2 + W^2}} * F \quad (2)$$

$$\text{Feed-rate in Z-axis,} \quad F_z = \frac{W}{\sqrt{U^2 + V^2 + W^2}} * F \quad (3)$$

There are three different programs for G01 that are made in the same path of the linear interpolation:

1. G90 absolute program- fig3-2

```
N1 G90
N2 G01 X25.000 Y20.000 Z10.000 F100.00 ...P1
```

N3 X60.000 Y50.000 Z40.000 ...P2

2. G91 increment program- fig3-2

N1 G91
 N2 G01 X25.000 Y20.000 Z10.000 F100.00 ...P1
 N3 X35.000 Y30.000 Z30.000 ...P2

3. G90 increment program- fig3-2

N1 G90
 N2 G01 X25.000 Y20.000 Z10.000 F100.00 ...P1
 N3 X35.000 Y30.000 Z30.000 ...P2

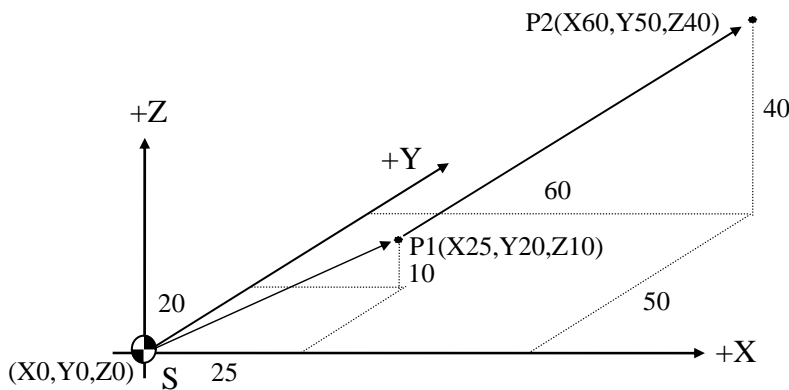


Fig 3-2 G01 Example

3.4 CNC and Master/Slave Mode

When the CNC controller executing a program block, the servo motor is always subjected to a motion sequence as: Acceleration—Constant feed-rate specified in the block—Deceleration. When the controller proceeds to execute the next block, the servo motor will repeat the same motion sequence. The CNC and Master/Slave mode for MILL controller is specially designed to govern how the acceleration/deceleration between blocks is to be connected.

CNC mode—

The servo motor will decelerate to a complete stop at the end of the first block, then the motor will accelerate to the feed-rate on the next block.

Master/Slave mode—

In this mode, the user select one axis as a Master axis and the rest will automatically become slave axes. The acceleration/deceleration connection of motor speed between blocks will NOT come to a complete stop. Instead, the motors for both master and slave axes will decelerate or accelerate to the feed-rate

of the next block from the feed-rate of the current block. The feed-rate (F) in the block is for the master axis and the feed-rate for the slave axes will be calculated according to their displacements. If the feed-rate for the master axis is zero (0), the feed-rate of the slave axis will be used for calculation.

MCM parameter #51 is for setting CNC and Master/ Slave mode as follows:

Setting=0, CNC mode
 Setting=1, X-axis as Master axis, the rest as Slave axes.
 Setting=2, Y-axis as Master axis, the rest as Slave axes.
 Setting=3, Z-axis as Master axis, the rest as Slave axes.
 Setting=4, Nonstop Mode.

MILL system provides three types of motor acceleration/deceleration, i.e. Linear, Exponential, and "S" curve. They are determined by the setting of MCM parameter#124 and will be explained in the examples below.

CNC mode: MCM#51=0

MCM#51setting	MCM#124setting	G00Acc./Dec.type
0	0	Linear
0	1	"S" curve

Example 1: Fig 3-3, CNC mode, motor acceleration/deceleration in exponential curve for G01 (MCM#124=0), Absolute coordinate.

N01 G00 X0. Y0. Z0	---Fx=1000., Fy=Fz=0
N10 G01 X100. F1000.	---Fx=Fy=500., Fz=250.
N20 G01 X200. Y100. Z50. F500	---Fx=250., Fy=Fz=0.
N30 G01 X300. F250.	---Fx=100., Fy=Fz=0.
N35 G01 X350. F100.	

Figure3-3 CNC Mode With G01- Exponential Accel. / Decel.

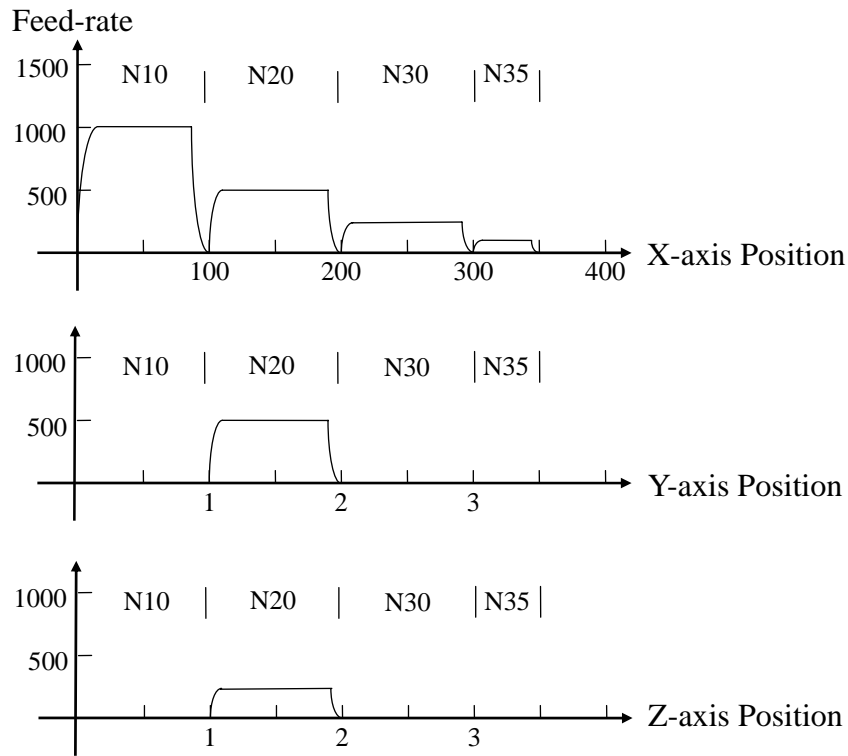


Fig 3-3

Example 2: G1 U100.0 V50.0 F1500.00 (CNC mode)

Calculate the G01 feed-rate for X and Y-axis. Assuming MCM #34 settings as $F_{trx} = 2000.00\text{mm/min}$, $F_{try} = 1000.00\text{mm/min}$.

Composite vector for X and Y-axis = $(100^2 + 50^2)^{1/2} = 111.8$

X-axis Feed-rate, $F_x = (100/111.8) \times 1500.00 = 1341.6$

Y-axis Feed-rate, $F_y = (50/111.8) \times 1500.00 = 670.8$

Both axes are in the G00 limited setting so two axes are calculated by feed-rate.

Example 3: G1 X100.0 Y200.0 F2000.00

Composite vector for X and Y-axis = $(100^2 + 200^2)^{1/2} = 223.6$

X-axis Feed-rate $F_x = 2000.0 \times (100/223.6) = 894.4$

Y-axis Feed-rate $F_y = 2000.0 \times (200/223.6) = 1788.9$

$F_y > TRY (1000.0)$, so the G01 feed-rate will be limited as:

$F_x = (894.4/1788.9) \times 1000 = 500.00$

$F_y = (1788.9/1788.9) \times 1000.00 = 1000.00$

Master/Slave mode MCM#51=1, 2, 3 or 4

MCM#51 setting	MCM#124 setting	G00 Acc./Dec.type
1,2,3,4	0	Linear
1,2,3,4	1	"S" curve

Example 1: Fig 3-4, X-axis as master axis, motor accel./decel. In linear type for G01(MCM#124=0), Absolute coordinate.

Fig 3-4a, X-axis as master axis, motor accel./decel. In "S" type for G01 (MCM#124=1), Absolute coordinate.

Note that the feed-rate in each block is for master axis. The feed-rate for slave axes will be adjusted according to their coordinate increment with respect to the master axis.

```
N10 G01 X100. F1000.
N20 X200. Y100. Z50. F500.
N30 X300. F250.
```

Fig 3-4 Master/ Slave Mode, linear accel./decel.
 Fig 3-4a Master/ Slave Mode. "S" Curve accel./decel.

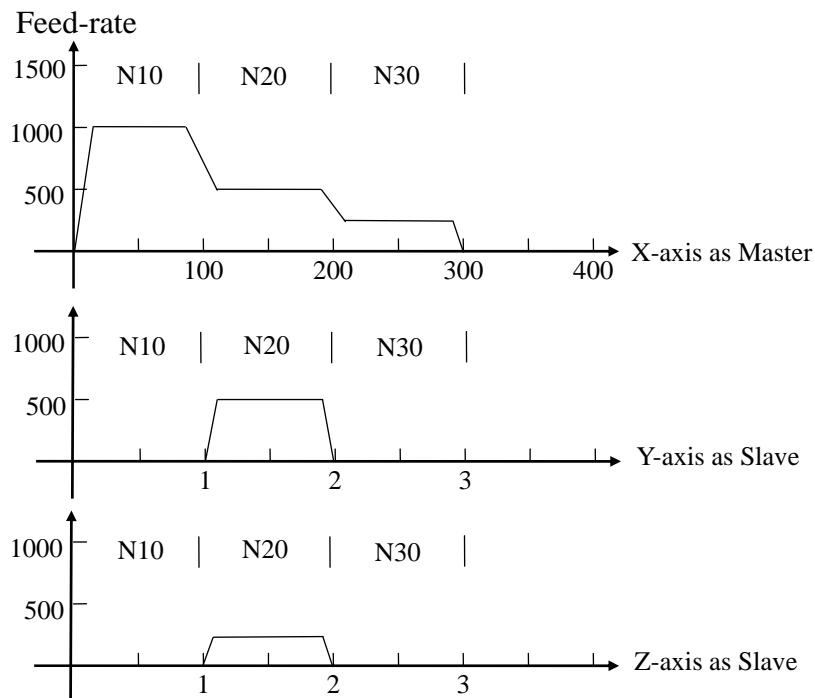


Fig 3-4

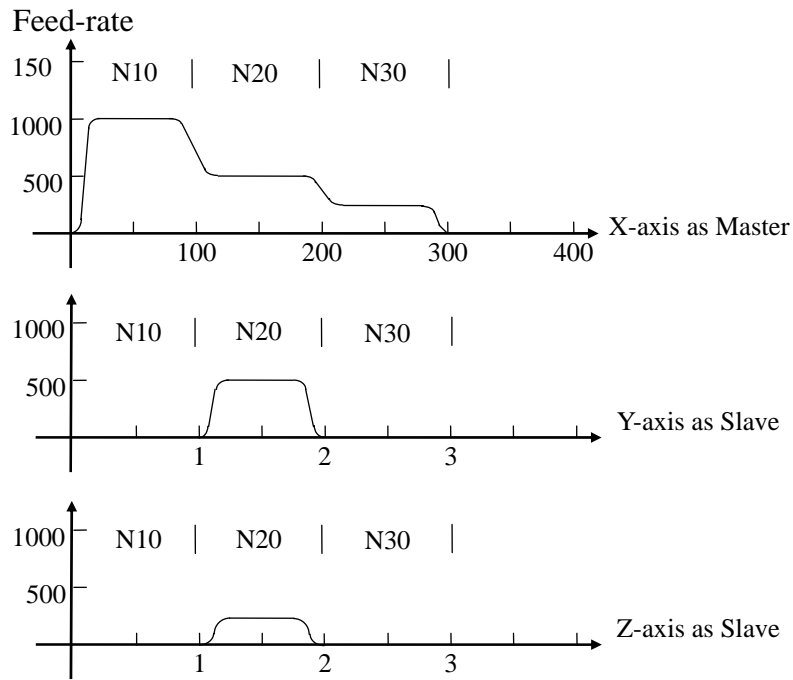


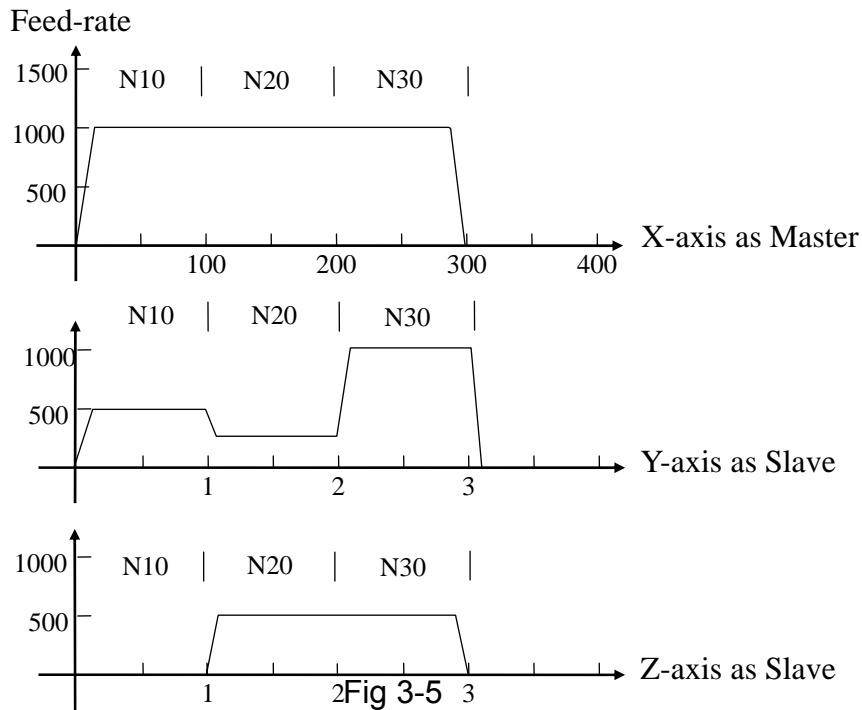
Fig 3-4A

Example 2: Fig 3-5, X-axis as master axis with constant feed-rate and Y, Z-axis as slave axes, motor accel./decel. in linear type for G01 (MCM#124=0), Absolute coordinate.

```

N10 G01 X100 Y50 Z0 F1000
N20 X200 Y75 Z50
N30 X300 Y175 Z100
    
```

Fig 3-5 Master/Slave mode with constant F-rate for master, linear accel./decel.



In example 2, the feed-rates of the slave axes are adjusted according to their incremental ratios. Note the small time required for accel./decel. between blocks and the distance traveled in this small amount of time can be estimated by:

$$\text{Distance} = 0.5 \times \frac{(F1 - F2)}{60} \times \frac{T}{1000}$$

F1, F2 = The feed-rates of 1st and 2nd block of the slave axis

T = The setting value of in MCM# 89

Therefore, if F_{1y} = 500.mm/min(N10 block), F_{2y} = 250.mm/min(N20) and MCM#89=100

There are the examples to calculate the feed-rates for master/slave. Meanwhile, it is necessary to consider their relation to the highest feed-rate settings (MCM#79~81).

In the example, set MCM# 79(TRX) =2000.00, MCM#80(TRY) =4000.00mm/min.

Example 3: G0 U100.0 V50.0 (X-axis as master, MCM#51=1)

Feed-rate for master axis F_x = 2000.00

Feed-rate for slave axis F_y =(50/100) × 2000.00=1000.00

F_y < TRY (4000.00), so the feed-rate is set by TRX setting (X-axis) of MCM#79 as standard.

Example 4: G0 U100.0 V300.0 (X-axis as master, MCM#51=1)

Feed-rate for master axis $F_x = 2000.00$
 Feed-rate for slave axis $F_y = (300/100) \times 2000.00 = 6000.00$
 $F_y < TRY (4000.00)$, thus the speed is limited as below:

Feed-rate for master axis $F_x = (4000/6000) \times 2000.00 = 1333.33$
 Feed-rate for slave axis $F_y = 4000.00$

The feed-rate is set by TRY setting of MCM#80 (Y-axis) as standard.

3.5 Circular (Arc) Cutting Command, G02 and G03

Format:

X-Y plane: G17 --- power-on default
 G02 (or G03) X(U)____Y(V)____ I____J____ F____ (1)
 (R can replace I,J)

X-Z plane: G18
 G02 (or G03) X(U)____Z(W)____ I____K____ F____ (2)
 (R can replace I,K)

Y-Z plane: G19
 G02 (or G03) Y(V)____Z(W)____ J____K____ F____ (3)
 (R can replace J,K)

The arc center variables I-J, I-K or J-K can be replaced by radius "R" and the arc cutting plane is determined by G17 (power-on default), G18 and G19. The arc cutting is the special case of the spiral cutting to be discussed in the next section.

Four (4) elements are required to do a circular cutting:

1. Arc cutting command code -- G02 or G03.

Depending on the arc cutting plane, the direction of circular cutting path is defined by command codes G02 (CW) and G03 (CCW) as in the figure below:

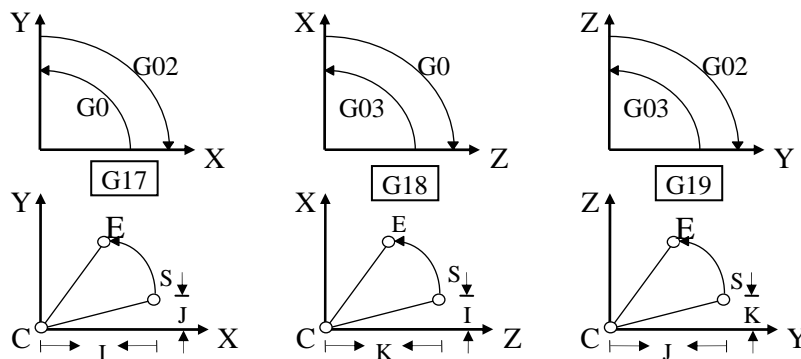


Fig 3-6 Directions of G02 and G03

2. The end point of the arc -- X(U), Y(V), Z(W).

U, V, W are the incremental coordinates from the start point (S) to the end point (E). The start point is the current position or the end point of the last block.

3. The center of the arc -- I, J, K or R

I, J and K are the X, Y, Z-axis components of the arc radius, respectively and R is the radius of the arc. Either representation is acceptable. I, J, K can be (+) or (-) and their meanings are identical to U, V, W. The range for "R" is -4000.~+4000. mm or -400.~+400. inches. Do not use R representation if the arc angle is in the range of $-1^{\circ}\sim+1^{\circ}$ or $179^{\circ}\sim181^{\circ}$.

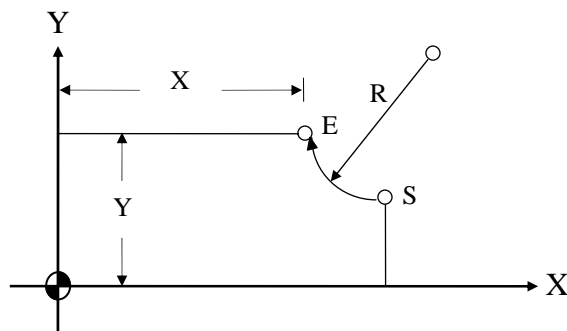


Fig 3-7 G2 Circular Cutting with "R" Specified

4. Arc cutting rate -- F-code

The minimum rate is 0.2 mm/min or 0.02 in/min.

When applying radius R method, be careful in determining the sign of radius R.

Fig 3-8

1. Use "+R" if arc angle $< 180^{\circ}$.
2. Use "-R" if arc angle $> 180^{\circ}$.

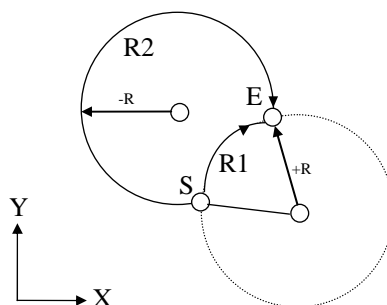


Fig 3-8 Arc Cutting with +R and -R

Example: The following four blocks will do the same arc cutting.

The start point X=50.000, Y=15.000

The end point X=30.000, Y=25.000

Radius R=25.000, or I=0.000, J=25.000

1. G02 X30.0 Y25.0 J25.0 F200.
2. G02 U-20.0 V10.0 J25.0 F200.
3. G02 X30.0 Y25.0 R25.0 F200.
4. G02 U-20.0 V10.0 R25.0 F200.

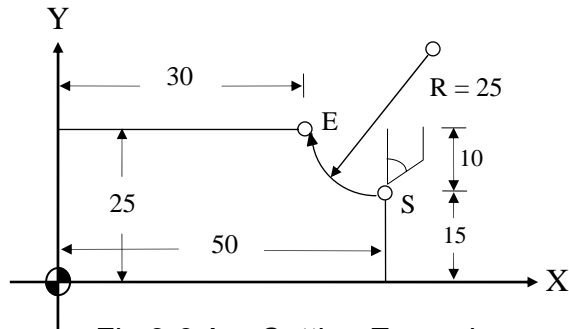


Fig 3-9 Arc Cutting Example

Notes on circular cutting:

1. G02, G03 command block must be followed by a G00 or G01 command block to signal to CNC the completion of the circular cutting. Otherwise, Error 25 will be displayed.
2. When cutting a circle, only the I, J, K method for arc center specification can be used. Radius "R" method will NOT yield satisfactory result.

Example: G90
 G00 X40.0 Y0.0
 G03 X40.0 Y0.0 I50.0 F100.0

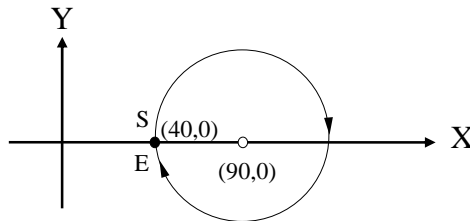


Fig 3-10 Cutting A Circle

3. When the calculated tangential cutting speed for the arc is greater than the programmed F-value, the programmed F-value will be used for the cutting. Otherwise, the calculated value will be used. The max. tangential cutting speed is estimated with the formula:

$$F_c = 85 * \sqrt{R * 1000} \quad \text{mm/min}$$

Where R= Arc radius in mm.

3.6 Thread Cutting, G17, G18, G19

As seen in the last section, these commands stand alone in a block. The meaning is to execute a circular cutting on the specified plane by G17~G19 and do the linear cutting on the third axis. The tool radius compensation is only effective on the

circular cutting plane. The following figure depicts the sign (+/-) convention of the three coordinates.

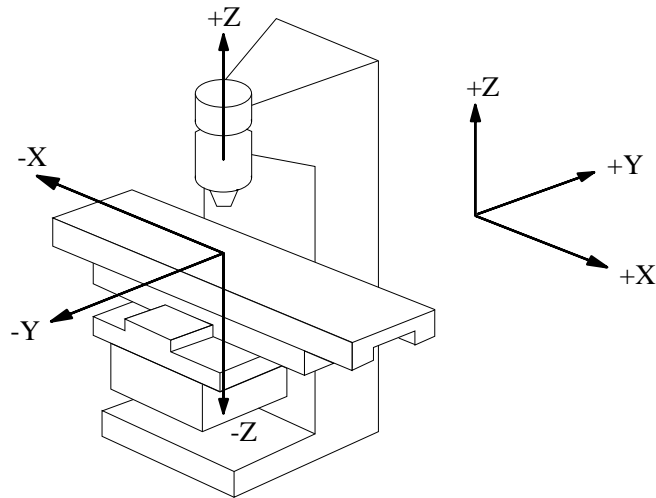


Fig 3-11 Sign (+/-) Convention Of The Three Coordinates (Right-hand Rule)

G17, X-Y Circular Cutting Plane

If you look down the machine from the top (along Z-axis), you have X-Y circular cutting plane and Z-axis as the linear axis.

Format: (see Fig 3-6 and Fig 3-12)

```
G17
G02 (or G03) X___Y___ I___J___ Z___F___      (1)
              (R can replace I,J)
```

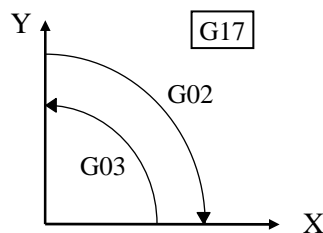


Fig 3-12

G18, Z-X Circular Cutting Plane

If you look at the machine from the back along the Y-axis, you have Z-X circular cutting plane and Y-axis as the linear axis.

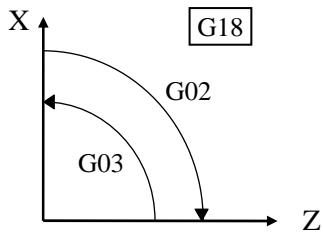


Fig 3-13

Format: (see Fig 3-6 and Fig 3-13)

```
N1 G18
N2 G02 (or G03) Z___X___ K___I___ Y___F___      (2)
              (R can replace K,I)
```

G19, Y-Z Circular Cutting Plane

If you look at the machine from the right along the X-axis, you have Y-Z circular cutting plane and X-axis as the linear axis.

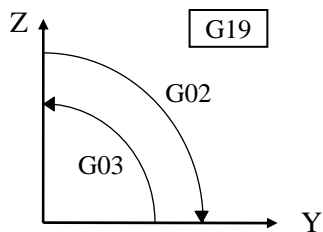


Fig 3-14

Format: (see Fig 3-6 and Fig 3-14)

```
N1 G19
N2 G02 (or G03) Y___Z___ J___K___ X___F___      (3)
              (R can replace J,K)
```

On all three formats, if the displacements are zero on the linear axes, you have circular cutting formats as shown in the last section. Note that the I, J, K representation can be replaced by "R" value also.

```
Example: N1 G17
         N2 G03 X80.000 Y30.000 R30.000 Z40.000 F100
```

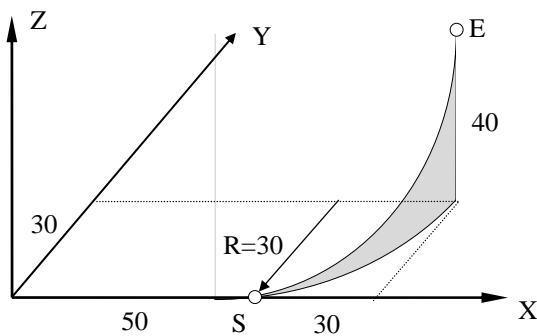


Fig 3-15

3.7 Feed-Hold, G04

Format: G04 X_____

X: Holding period in seconds, ranging 0.01~8000.0 seconds.

Under some circumstances during cutting, it becomes necessary to hold (stop) the cutting action for certain period before proceeding to the next block. In this case, G04 function can be used for this purpose.

Example:

```
N1 G1 X10.000 Y10.000 F100.
N2 G4 X2.000      ..... Hold for 2 seconds, then process to next block
N3 G0 X0.000 Y0.000
```

3.8 Reset Origin of Machine Coordinate, G08

Format: G08 Clear machine coord. in all axes
or G08 X_____Y_____ Clear X and Y-axis
or G08 Z_____	Clear machine coord. in Z-axis.
or G08 X_____Y_____	Clear machine coord. in X and Y-axis.
or G08 any combination of X, Y, Z	

The number associated with X, Y, Z does not have any meaning, but you have to have a number to input X, Y, Z into the CNC buffer. When G08 is used in the program, the machine coordinates accumulated up to the G08 command for the specified axis (axes) will be cleared regardless of what numbers are with X, Y, Z.

3.9 Manual Data Input Function, G10

Table 3-2 G10 Application for HUST H4 Controller

G10 Command	
G10 X___Y___	Set origin for G54~G59 work coordinate
G10 X___Y___P___	Set tool offset compensation data
G10 X___Y___P100	Set in-position data for MCM #87 & #88
G10 P200 L___	Set counter limit for MCM #66
G10 P201	Clear counter to zero, MCM #65=0
G10 P510 L4800	Set the baud rate of RS232 interface on controller @ 4800
G10 P510 L9600	Set the baud rate of RS232 interface on controller @ 9600
G10 P510 L19200	Set the baud rate of RS232 interface on controller @ 19200
G10 P600 L01	Burn the downloaded program into FLASHROM
G10 P600 L02	Burn the downloaded MCM parameters into FLASHROM
G10 P600 L03	Burn the downloaded ladder program into FLASHROM
G10 P600 L04	Burn the downloaded LCD display into FLASHROM
G10 P600 L05	Burn the downloaded system data into FLASHROM
G10 P800 L___	Set the acceleration/deceleration time for G01
G10 P1000	Clear all MCM parameters to factory default values
G10 P2000	Clear the current program
G10 P2001	Clear all programs in the memory
G10 P2002	Clear all variables #1 ~ #9999 to zero

3.9.1 Set Origin for Work Coordinate by G10 (Recommended Method)

Format: G54 ... or G55~G59

G10 X___Y___Z___

Steps to set the origin of the specified work coordinate:

1. Execute HOME to move the tool to home position.
 2. Press JOG key.
 3. Use MPG hand-wheel to move the tool to the desired location for the origin.
 4. Press MDI key, enter G54, then press CYCST.
- 5A. If the coordinates in Step 3 is the desired location for your origin, execute the followings. Otherwise, skip this step and go to Step 5B.

Press G10 Input, X0. Input, Y0. Input Z0. Input.
Press CYCST key to complete the setting process.

- 5B. If the coordinates in Step 3 has some distance (say X=20.0, Y=100.0 Z15. Input) away from the desired origin, execute as follows:

Press G10 Input, X20. Input, Y100. Input Z0 Input.
Press CYCST key to complete the setting process.

The following precautions should be noted when using G10 to set the work origin.

1. Do not add P__ in the G10 block. Otherwise, it becomes an offset compensation command.
2. For setting G55~G59 work origin, use the same method except G54 in step 4 is replaced by G55~G59. If no G54~G59 is specified in step 4 above, the work origin data will be entered into the current work coordinate system.
3. G10 command can be applied in the program.
4. When G54~G59 is executed through G10 command, the corresponding coordinates in MCM #1~#18 will be automatically revised.

3.9.2 Set Tool Offset Data By G10

Format:

1. G10 X___ Y___ Z___ P1__
2. G10 U___ V___ W___ P1__
3. G10 I___ J___ K___ P1__

- P1_ : 1~40 representing the group number in MCM #301~#460.
 X, Y,Z : Setting the Tool Offset Compensation data of X, Y, Z to the existing values of the appropriate MCM #301~#460.
 U, V, W : Setting the Tool Wear Compensation data of U, V, W to the existing values of the appropriate MCM #501~#660.
 I, J, K : Add the Tool Wear Compensation data of I, J, K to the existing values of the appropriate MCM #501~#660.

P -group number	X -axis Tool offset compensation	Y -axis Tool offset compensation	Z -axis Tool offset compensation	R -Tool radius compensation
1	MCM#301	MCM#302	MCM#303	MCM#304
2	MCM#305	MCM#306	MCM#307	MCM#308
3	MCM#309	MCM#310	MCM#311	MCM#312
4	MCM#313	MCM#314	MCM#315	MCM#316
.
.
39	MCM#453	MCM#454	MCM#455	MCM#456
40	MCM#457	MCM#458	MCM#459	MCM#460

P -group number	X -axis Tool wear compensation	Y -axis Tool wear compensation	Z -axis Tool wear compensation	R -Tool radius wear compensation
1	MCM#501	MCM#502	MCM#503	MCM#504
2	MCM#505	MCM#506	MCM#507	MCM#508
3	MCM#509	MCM#510	MCM#511	MCM#512
4	MCM#513	MCM#514	MCM#515	MCM#516
.
.
39	MCM#653	MCM#654	MCM#655	MCM#656
40	MCM#657	MCM#658	MCM#659	MCM#660

Example-1:

Execute the `G10 X0.02 Y0.03 P101` --> Setting the 1 group of Tool Offset Compensation data.

>> `MCM#301 = 0.02` , `MCM#302 = 0.03`

Example-2:

Assume: The original `MCM#505~507` setting is `X=0.02` , `Y=0.03` , `Z=1.25`

Execute the `G10 U0.01 V0.02 W1.72 P102` → Setting the 2 group of Tool Wear Compensation data.

>> `MCM#505 = 0.01` , `MCM#506 = 0.02` , `MCM#507 = 1.72`

Example-3:

Assume: The original `MCM#505~507` setting is `X=0.02` , `Y=0.03` , `Z=1.25`

Execute the `G10 I0.01 J0.02 K1.72 P102` → Add the value into the 2 group of Tool

Wear Compensation data.

- >> $MCM\#505 = 0.02 + 0.01 = 0.03$
- $MCM\#506 = 0.03 + 0.05 = 0.05$
- $MCM\#507 = 1.25 + 1.72 = 2.97$

3.9.3 Set and Clear Counter Limit By G10

Format: G10 P200 L____ (Set)
 G10 P201 _____ (Clear)

L : The counter limit to be stored in MCM #92. The maximum number is 9,999,999.

P200: A fixed command to input the counter limit data into MCM #92.

P201: A fixed command to clear the counter MCM #91 to zero.

The counter limit (MCM #92) is the maximum number of cycles that the program has been repeatedly executed and the counter (MCM #91) is the one that records the actual execution cycles. Every time the program execution encounters M02, M30 or M99, the number in counter MCM #91 will increase by 1. When the number in MCM #91 is increased to a value that is equal to the number in MCM #92, the program execution will stop. If L=0, the counter MCM #91 is unlimited.

For counter limit function to be effective, C30 of the C-bit in PLC program must be high (1). This can be accomplished by installing an input point (a switch) and process by PLC.

3.9.4 Set the Acceleration/Deceleration Time for G01 with G10

The acceleration/deceleration time is stored in MCM #89. This number can be revised or adjusted by one of the following 3 methods. Press "reset" key for revision to be effective.

1. Revise it in MCM EDIT mode.
2. Revise it by G10 in MDI mode. (explain below)
3. Revise it during program execution in AUTO mode.

Format: G10 P800 L_____ where "L" is expressed in milliseconds

This program block can also be blended in the program for AUTO execution as shown in example 2 below.

Ex1: G10 P800 L100

Execute this program in MDI mode will revise MCM #89 to 100 milliseconds. Press "Reset" key for the revised number to be effective.

Ex2: The short program shows how the acceleration/deceleration time (A/D time) can be changed during program execution. The tool's travel distance is stored in variable #1.

A/D time = 30 milliseconds if variable #1 ≤ 100 mm
 A/D time = 50 milliseconds if variable 100 < #1 ≤ 200 mm
 A/D time = 100 milliseconds if variable #1 > 200 mm

```
O001
N001 G65 L85 P005 A#1 B100
N002 G10 P800 L30
N003 M02
N005 G65 L85 P008 A#1 B200
N006 G10 P800 L50
N007 M02
N008 G10 P800 L100
N010 M02
```

3.10 English / Metric measuring System Conversion, G20 G21

Format: G20 -- System measuring unit in INCH mode
 G21 -- System measuring unit in METRIC mode. This is the power-on
 and RESET default.

3.11 Move To The First Reference Point, G28

Format: G28 Move to the 1st ref. point in all axes
 or G28 X___Y___Z___ Move to the 1st ref. point in X,Y, Z-axis

It can be one to three axes in the coordinate commands after G28 command regardless of what the numbers are. The coordinates of the first reference point is set in the MCM parameter from #52~54. The number associated with X ,Y, Z-axis does not have any meaning, but you have to have a number to input X,Y, Z into the CNC buffer. When encountering this command during cutting, the tool will move to the first reference point as set in MCM parameter from #52~54 for the axis specified in the G28 block, regardless of what numbers are with X,Y, Z.

The coordinates of MCM parameter from #52~54 are determined by users, based on the machine origin being at X=Y=Z=0. This reference point is normally selected at some convenient location during machining. Therefore, if X=Y=Z=0 is selected for MCM from #52~54, G28 command will cause the tool moving to machine origin.

Example:

G28 Three axes return simultaneously.
 G28 X___ Single axis return.
 G28 X___Y___ Double axes return.

Note that prior to the G28 command, the tool offset compensation must be canceled and the tool offset compensation cancel command should not be used in the same block as the G28 command.

Example:

```
G40      ..... Tool offset compensation canceled
G28 X10. .... Tool moves to the 1st ref. point in X-axis, no motion in Y, Z-axis.
```

3.12 Return To The Previous Location From The Reference Point, G29

Format:

G29 X____Y____Z____

The G28 command moves the tool to the first reference point. G29 command works just the opposite. It moves the tool from the reference point to the last position, prior to G28 command, as indicated by X, Y, Z in the program block. G29 command cannot be used alone. Instead it is used following a G28 or G30 command. Again, the number associated with X, Y, Z does not have any meaning, but you have to have a number to input X, Y, Z into the CNC buffer.

Example:

```
N1 X60. Y0. Z30.      Tool at the location of X60, Y0., Z30.
N2 G28                Tool moved from (X60, Y0, Z30) to the 1st ref. point.
N3 G29                Tool returns from the ref. point to (X60, Y0, Z30)
```

Like G28 command, it can be one to three axes in the coordinate command regardless of what the numbers are. The tool will follow the command and return to the previous location when it execute to G28. the number associated with X, Y, Z does not have any meaning, but you have to have a number to input X, Y, Z into the CNC buffer.

Example:

```
G29                Three axes return simultaneously.
G29 X____          Single axis return.
G29 X____Y____     Double axes return.
```

3.13 Move To The Second (2nd) Reference Point, G30

Format: G30 X____Y____Z____

The method of application for this command is the same as for G28. The coordinates of this reference point are set in the MCM parameter from #55~57.

3.14 Skip Function, G31

Format: G31 X(U)____Y(V)____Z(W)____

X, Y, Z: Final position in absolute coordinate.
 U,V,W: Final position in increment coordinate.

For Skip function G31 to be effective, it must be used in combination with an I/O signal to be received during the execution of G31 block. Once an I/O signal is detected, the tool will forgo the unfinished operation of current block and starts executing the next block. If no I/O signal is received during the execution of G31 block, the tool will move to the coordinates as specified with the cutting speed as G01.

When G31 is doing linear cutting, the feed-rate will be the one in effect (G00 or G01). G31 is an one-shot G-code.

Example:

```
N40 G40
N50 G31 U100.0 F100.
N60 G01 V25.000
N70 X90.0 Y30.
```

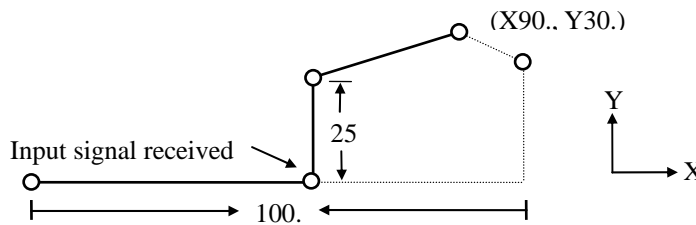


Fig 3-16 G31 Skip Function

In Fig 3-16, the dashed line represents the original path without SKIP function and the solid line is the actual tool path when the SKIP function signal is received. Prior to using SKIP function, do NOT use the tool compensation command.

Note that G31 cannot be used in the state of tool radius compensation. The user must execute G40 to cancel the tool radius compensation before executing G31. Skip function in the program dry run is ineffective on feed-rate adjusting and auto acceleration/deceleration.

3.15 Work-piece Dimension Enlarging and Shrinking, G50, G51

Format: G51 I____J____K____ (Command set)
 G50 (Command cancel)

I, J, K in this command are the proportionalities for the dimension to be enlarged or shrunk. Their ranges are 0.01~100 and the the max. and min. ratio in the block should be kept below 100. G51 can execute 1~3 axis at the same time and the dimension for the axis not specified will remain unchanged.

When using G51 command, it's better for the tool to return to the starting point, then use G50 to cancel it before doing other cutting. The reason is that G51 is a modal G-code and it remains effective unless G50 is used. The danger of this is that the tool may hit the travel limit and cause an accident, particularly if an enlarging scale is used.

Example:

```
N10 G1 X0.0 Y10.0 Z10.0
N20 G51 I2.0 J2.0    --- Dimension in X and Y-axis to be enlarged by 200%
N30 X40.0 Y40.0    --- Actual cutting X=80.0 mm, Y=60.0 mm
N40 G50
```

3.16 Work Coordinate System Setting, G54~G59

There are two coordinate systems for CNC machine tool. They are:

1. Machine Coordinate (Home)
2. Work Coordinate:
 Work Coordinate System (G54~G59)--To be set in the MCM parameters
 (Suggested method)

3.16 .1 Machine Coordinate (Home)

The origin of the machine coordinate system is a fixed point on the machine. Its location is normally determined by the locations of the over-travel limit switches (OTLS). When you execute HOME from the control panel, the tool or the machine table will move toward the OTLS, then reverse back to look for the encoder GRID signal. When it locates the GRID, the tool stops. This location is the HOME position or Machine origin. Machine origin is the calculation basis for all work coordinates and the reference point coordinates. Before you do any cutting, be sure to execute HOME.

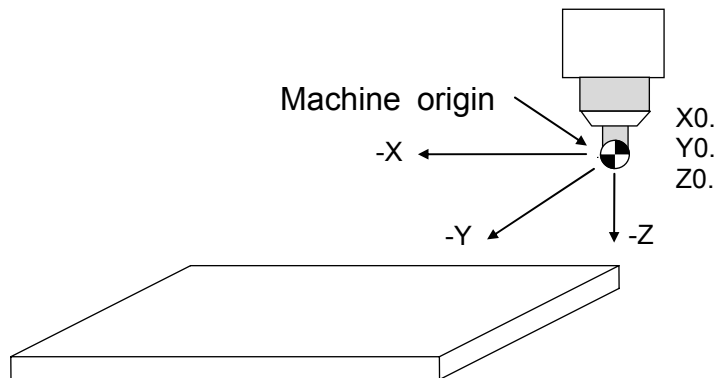


Fig 3-17 Machine Origin

Occasionally, for the convenience of doing cutting, it becomes necessary to set another origin that is slightly shifted from the machine origin. Such origin is called HOME SHIFT. The amount of shift is set in MCM #99~101. When you execute HOME, the tool will rest at the HOME position but the machine coordinate will show the home-shift values. If the setting values in MCM #99~101 is zero (0), the HOME SHIFT is the HOME position. The methods to find HOME position are:

1. Manually execute HOME from the control panel.
2. Use G28 or G30 by setting the reference coordinates in the MCM to zero for all axes.

3.16.2 Work Coordinate System Setting, G54~G59

Work Coordinate System

H3X-M series provides 6 sets of work origin system that are stored in the MCM parameter #1~18, in which the 1st line is the work origin coordinate for G54, the 2nd line for G55, and so forth. The coordinates of these work origins are the coordinates with respect to the machine origin. The coordinate data input into MCM #1~18 can be done by:

1. G10 command in MDI mode
2. Direct input in MCM mode
3. Enter manual jog mode

The application of these work origins in the program is accomplished by G54~G59 command as explained in the examples below. Following the need and the program design, users can choose one to six sets to work. The best of these work coordinates is to simplify the calculations of part program. The examples are followed below:

- G54 represents the work coordinate in the parameter of MCM# 1~3.
- G55 represents the work coordinate in the parameter of MCM# 4~6.
- G56 represents the work coordinate in the parameter of MCM# 7~9.
- G57 represents the work coordinate in the parameter of MCM# 10~12.

G58 represents the work coordinate in the parameter of MCM# 13~15.
G59 represents the work coordinate in the parameter of MCM# 16~18.

The application of these work origins in the program is accomplished by G54~G59 command as explained in the examples below. The advantage of using these work origins is the simplification of the coordinate calculations for the work-piece. Fig 3-19 shows six geographic cutting patterns with six work origins for G54~G59 as follows:

G54, coord. settings in the 1st line of MCM # 1~3 -- X-70.000, Y-10.000
G55, coord. settings in the 2nd line of MCM # 4~6 -- X-80.000, Y-30.000
G56, coord. settings in the 3rd line of MCM # 7~9 -- X-80.000, Y-50.000
G57, coord. settings in the 4th line of MCM #10~12 -- X-70.000, Y-50.000
G58, coord. settings in the 5th line of MCM #13~15 -- X-40.000, Y-60.000
G59, coord. settings in the 6th line of MCM #16~18 -- X-20.000, Y-40.000

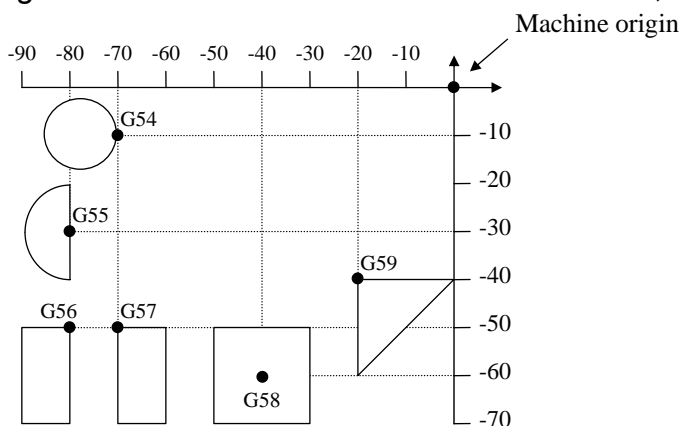


Fig 3-18 G54~G59 Work Origin

Once the work origins have been set in the MCM #1, the cutting patterns can be accomplished using G54~G59 commands as shown below. Only G54 and G55 are shown in the example, but G56~G59 can be done the same way. Note that the program coordinates are also changed when the work origin is changed.

```

N1 G0                -- Feed-rate set at fast move mode
N2 G54 X0. Y0.      -- Move to G54 work origin
N3 G2 I-7.0 F200.0  -- Cut a circle in CW with R=7.0
N4 G0                -- Feed-rate set at fast move mode
N5 G55 X0. Y0.      -- Move to G55 work origin
N6 G1 V10.0 F300.   -- Line cut along Y-axis @ 300 mm/min
N7 G3 V-20.0 R10.0 F300. -- Cut a half-circle in CCW with R=10.0
N8 G1 V10.0 F300.   -- Line cut along Y-axis @ 300 mm/min
N9 G28              -- Move to the 1st reference point
N10 M2              -- Program end

```

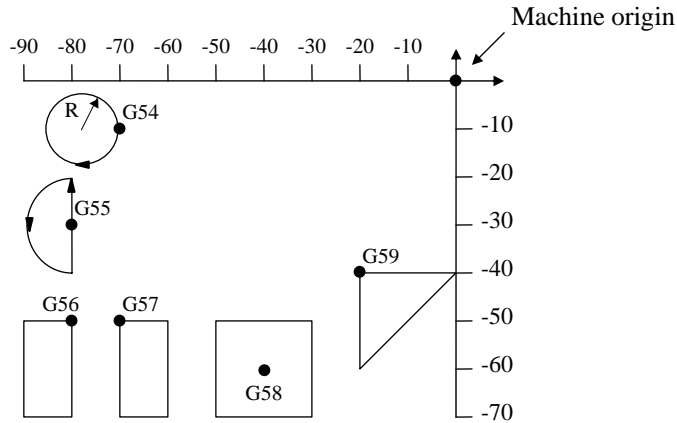


Fig 3-19 The Application of G54~G59

Notes on G54~G59 command:

1. Power-on default is the G54 command.
2. When using G54~G59 command, the associated X, Y, Z-values in the program block are the coordinates with respect to the specified work origin and the tool will move to that location.
3. After using G54~59, the machine coordinate of HOME ORIGIN will be changed by the new one.

3.17 Mirror-effect Cutting, G68, G69

Format: G68 -- Mirror-effect cutting, the X-axis as mirror
 G69 -- Mirror-effect cutting, the Y-axis as mirror

The mirror-effect cutting is by using a sub-program to design a cutting pattern, then using G68, G69 command to accomplish the mirror-effect cutting as shown in Fig 3-21. In application, G68 or G69 forms a single program block. For G68, the sign of the X-coord. will be reversed (+ → - or - → +) and for G69, the sign of the Y-coord. will be reversed.

Therefore, all you have to do in cutting the patterns as in Fig 3-21 is writing a sub-program for pattern 1, then use G68 and G69 commands as follows:

```

M98 P___ -- Sub-program call to cut pattern 1 (+X, +Y)
G68      -- Reverse the sign for X-coord.
M98 P___ -- Sub-program call to cut pattern 2 (-X, +Y)
G69      -- Reverse the sign for Y-coord.
M98 P___ -- Sub-program call to cut pattern 3 (-X, -Y)
G68      -- Reverse the sign for X-coord.
M98 P___ -- Sub-program call to cut pattern 4 (+X, -Y)
G69      -- Reverse the sign for Y-coord and back to the position for
pattern 1.
M2
    
```

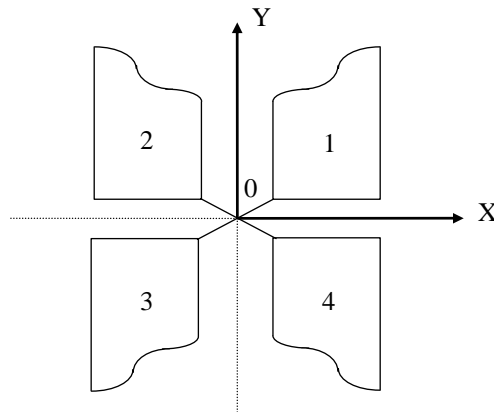


Fig 3-20 Mirror-effect Cutting

Note that G68 and G69 are modal G-code, you have to use one more G69 (or G68) as shown in the program above to reverse the coordinate sign back to normal. The mirror-effect cutting can also be cancelled by RESET key. When mirror-effect cutting is in effect, the "X-mirror" or "Y-mirror" will be shown at the top of CRT screen in reversed cursor.

3.18 Absolute and Incremental Coordinate Command, G90, G91

Format: G90 -- Absolute coord. mode, power-on default

G91 -- Incremental coord. mode for X, Y, Z.(see chapter 2)

The absolute coordinate system (G90) is the power-on default. When in absolute mode, you can use the position command codes (X, Y, Z, or U, V, W) to obtain the desired absolute/ incremental coord. descriptions. However, when G91 command is in effect, all X, Y, Z-coordinates represent incremental values and the U, V, W-codes are rendered ineffective.

Example 1: Absolute coord. setting

```
N1 G90
N2 G1 X20.000 Y15.000    -- P0 to P1
N3 X35.000 Y25.000     -- P1 to P2
N4 X60.000 Y30.000     -- P2 to P3
```

Example 2: Incremental coord. setting (Fig 3-21)

```
N1 G91
N2 G1 X20.000 Y15.000    -- P0 to P1
N3 X15.000 Y10.000     -- P1 to P2
N4 X25.000 Y5.000      -- P2 to P3
```

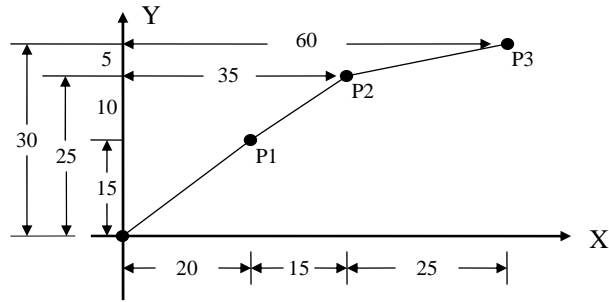


Fig 3-21 G90, G91 Example

3.19 Canned Cycle Command, G80, G81~G89 (Special use for MILL)

These G-code commands are for H4M milling machine only, NOT for other HUST CNC controller. H4M series provides several canned cycle functions to simplified the program design. All these canned cycle functions can be summarized as in Fig 3-23 below.

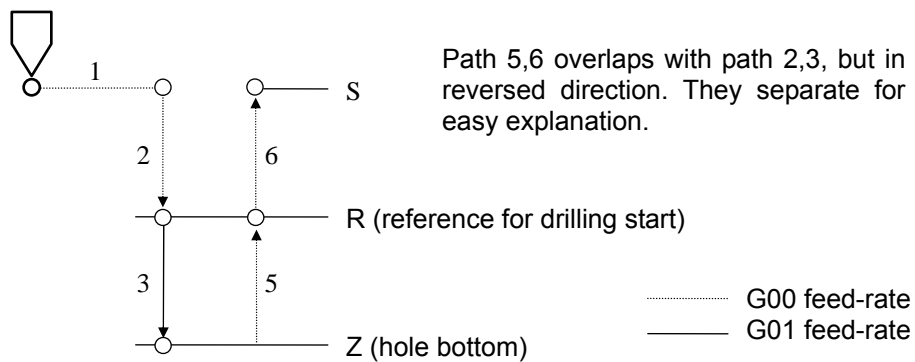


Fig 3-22 Canned Cycle Cutting Sequence

1. Fast move to the start (S) point on X-Y plane.
2. Fast move to the reference point for drilling start (R) along Z-axis.
3. Hole drilling to the bottom (Z) along Z-axis.
4. Mechanical action at the hole bottom -- dwell or spindle rotation reversed.
5. Drill bit retracted to R-point. The moving speed depends on the command specified.
6. Fast move back to the start point S with G00 feed-rate.

When applying the canned cycle function, you must use the followings:

- M03 -- Spindle rotation in the normal direction.
- M04 -- Spindle rotation in the reversed direction
- M05 -- Spindle stopped.

Format For Canned Cycle:

G90 or G91
 G98
 G81~G89 X___Y___Z___P___Q___R___F___K___
 G80 or G00 or G01

- X, Y : Coordinate of hole location, absolute or incremental.
 Z : Coordinate of hole bottom, absolute or incremental.
 P : Dwell time at hole bottom, in 1/1000 seconds.
 Q : Amount of feed for each cut (for G83), in mm.
 R : Coord. of the tool reference point for drilling start or tool retraction, absolute or incremental.
 F : Feed-rate.
 K : Number of repetition for hole drilling.

Note that all the canned cycle commands, position codes, and the associated parameters are modal codes. Unless re-specified, they will remain effective each cycle.

3.20 G90 or G91 -- Absolute or incremental coord. setting

When in G90 absolute mode, R and Z-values are the coord. from Z=0. In G91 mode, they are increments from the start point S. Due to this, the resulting hole depths are different even with the same R, Z parameters, see Fig 3-23.

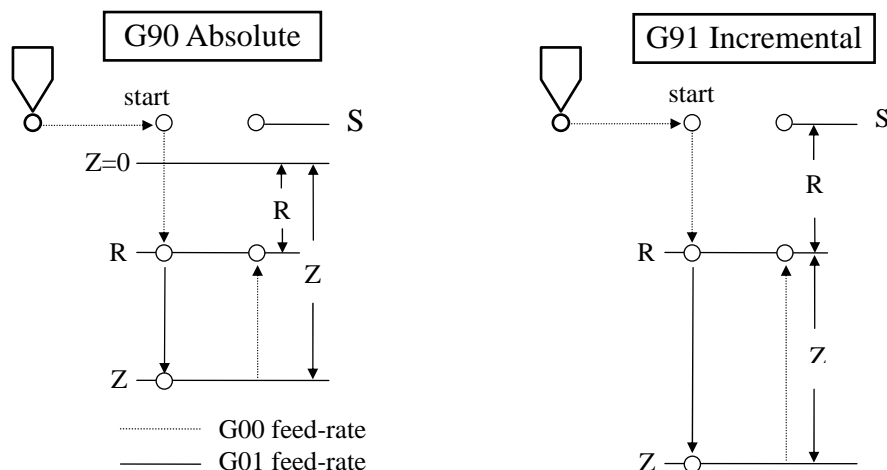


Fig 3-23 G90, G91 Effect On R, Z-value

3.21 G80, G81~G89 -- Canned cycle commands

The application of these commands is tabulated below.

	Application	Drill Rate	Action at Bottom	Retraction Rate
G80	Cycle canceled	---	-----	----
G81	Drilling	G01	--	G00
G82	Drilling	G01	Dwell	G00
G83	Deep hole drilling	G01	--	G00
G85	Boring	G01	--	G01
G86	Boring	G01	Spindle stopped	G01
G89	Boring	G01	Dwell	G01

3.22 G80 -- Canned cycle cancellation.

All canned cycle commands must be cancelled by G80, G00 or G01.

3.23 G81 Drilling Canned Cycle

Format: G81 X___ Y___ Z___ R___ K___ F___

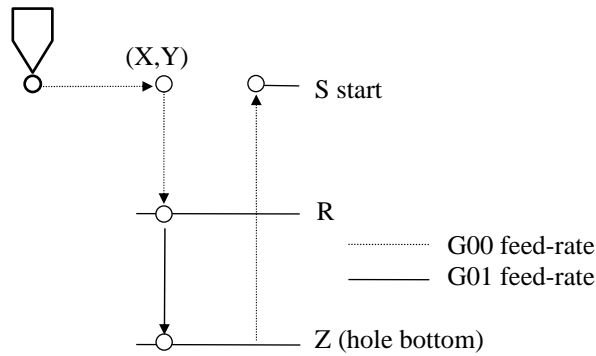


Fig 3-24 G81 Canned Cycle

3.24 G82 Drilling Canned Cycle -- Dwell at hole bottom

Format: G82 X___ Y___ Z___ P___ R___ K___ F___

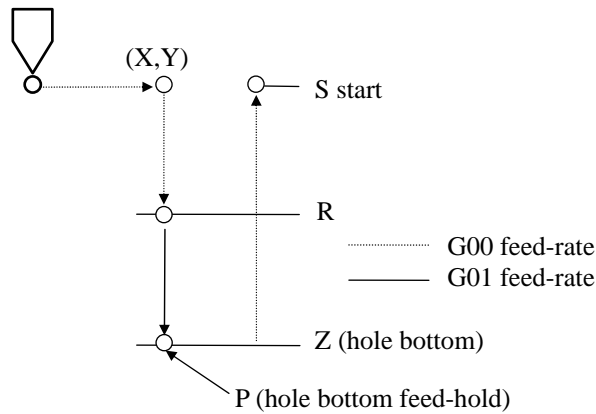


Fig 3-25 G82 Canned Cycle with Dwell at Hole Bottom

The difference between G81 and G82 is that G82 has a dwell time (P) when the drill bit reaches bottom. The dwell time is input as an integer in 1/1000 seconds.

3.25 G83 Deep Hole Drilling Canned Cycle

Format: G83 X___ Y___ Z___ R___ Q___ K___ F___

A deep hole cannot be finished in one forward action. Instead, it'll be finished in several up-and-down repetitions as specified by K. Q is the drilling depth for each repetition. The small "d" in Fig 3-26 is the location where the feed-rate is changed over from G00 to G01 and this number is set in MCM parameter #152.

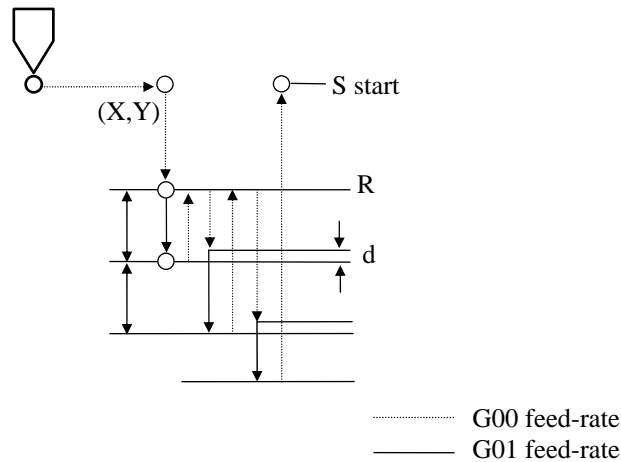


Fig 3-26 G83 Deep Hole Drilling Canned Cycle

3.26 G85 Boring Canned Cycle

Format: G85 X___ Y___ Z___ R___ K___ F___

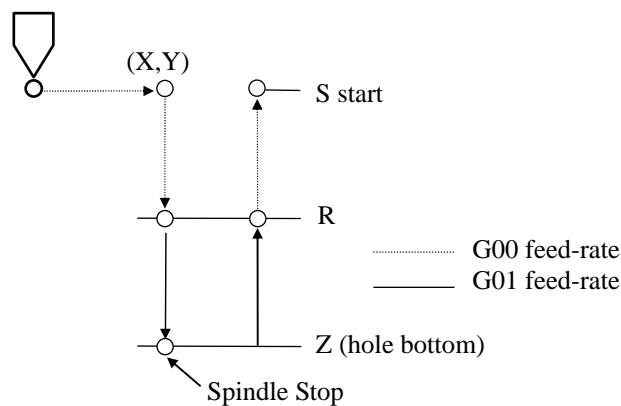


Fig 3-27 G85 Boring Canned Cycle

3.27 G86 Boring Canned Cycle with Spindle Stopped at Hole Bottom

Format: G86 X___ Y___ Z___ R___ K___ F___

Fig 3-28 G86 Boring Canned Cycle with Spindle Stopped at Hole Bottom
The difference between G85 and G86 is that the spindle stops at hole bottom before the drill bit retracts.

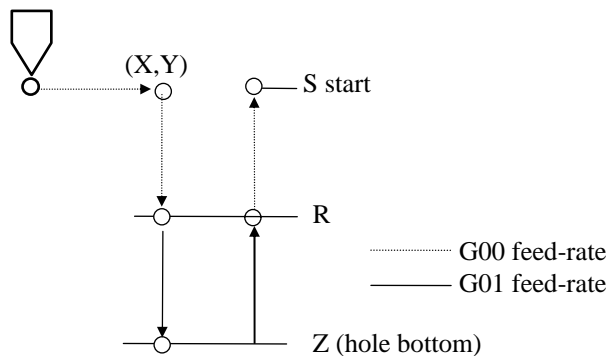


Fig 3-28 G86 Boring

3.28 G89 Boring Canned Cycle with Dwell at Hole Bottom

Format: G89 X___ Y___ Z___ R___ P___ K___ F___

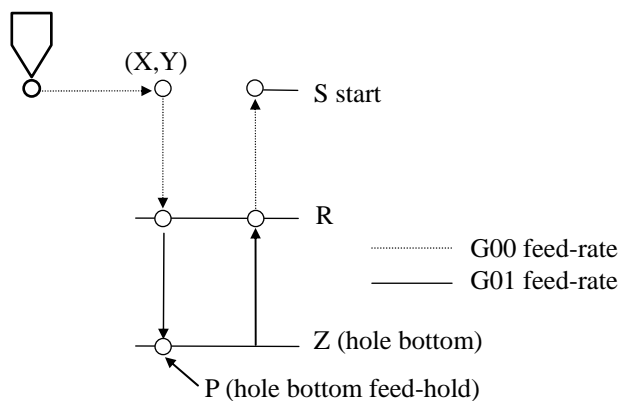


Fig 3-29 G89 Boring Canned Cycle with Spindle Dwell at Hole Bottom

The dwell time (P) is input as an integer in 1/1000 seconds.

3.29 G22 Linear Groove (This function is limited by absolute mode)

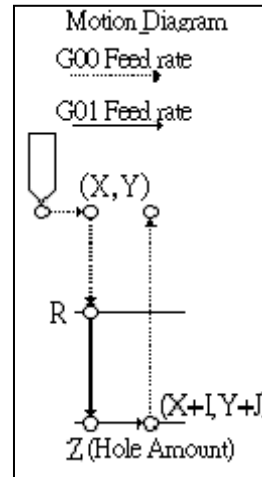
Format:

G22 X___ Y___ Z___ R___ I___ J___ F___

- X Start point of X coordinate
- Y Start point of Y coordinate
- Z Depth of groove
- R Height of outer part
- I The end point in relation to the start point from the incremental X-axis.
- J The end point in relation to the start point from the incremental Y-axis.
- F Groove speed

Explanation:

1. G00 X(x) Y(y)
2. G00 Z(r)
3. G01 Z(z) F(f)
4. G01 X(x+l) Y(y+j)
5. G00 Z(r)



3.30 G23 Curve Groove (This function is limited by the absolute coordinate)

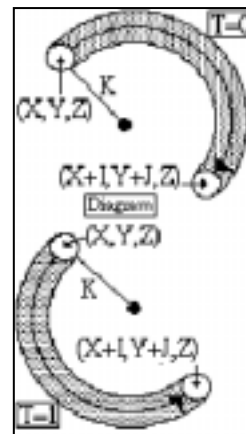
Format:

G23 X__Y__Z__R__I__J__K__T__F__

- X Start point of the coordinate
- Y Start point of the coordinate
- Z Depth of Groove
- R Height of outer part
- I The end point in relation to the start point from the incremental X-axis.
- J The end point in relation to the start point from the incremental Y-axis.
- K Radius of circle
- T Groove type (0~1)
- F Groove speed

Explanation:

1. G00 X(x) Y(y)
2. G00 Z(r)
3. G01 Z(z) F(f)
4. T=0; G02 U(i) V(j) R(k) F(f)
T=1; G03 U(i) V(j) R(k) F(f)
5. G00 Z(r)



The gray area shows the cutting groove.

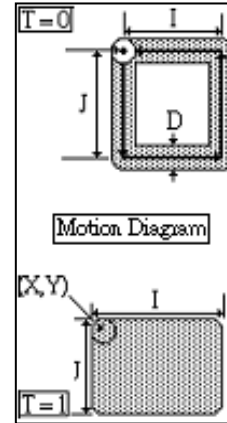
PS. R value is positive when the arc cutting is smaller than 180 degrees.
R value is negative when the arc cutting is bigger than 180 degrees.

3.31 G24 Rectangular Groove (This function is limited by absolute coordinate)

Format:

G24 X__Y__Z__R__I__J__D__T__F__

- X Start point of the coordinate
- Y Start point of the coordinate
- Z Depth of Groove
- R Height of outer part
- I The end point in relation to the start point from the incremental X-axis.
- J The end point in relation to the start point from the incremental Y-axis.
- D Tool radius
- T Groove type (0~1)
- F Groove speed

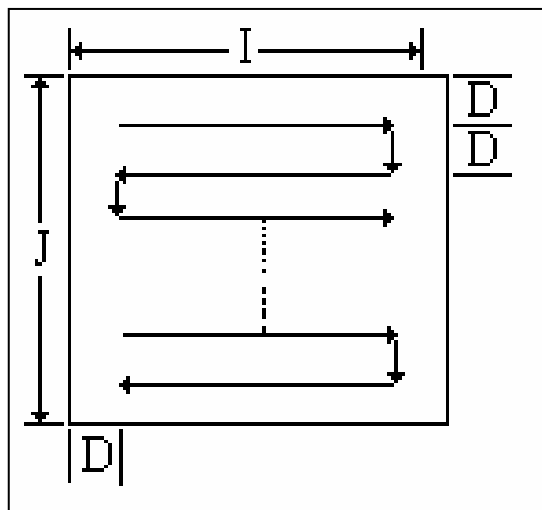


Explanation:

T=0;

1. G00 X(x) Y(y)
2. G00 Z(r)
3. G01 Z(z)
4. G01 U(i)
5. G01 V(j)
6. G01-U(i)
7. G01-V(j)
8. G00 Z(r)

T=1;



The diagram above shows the movement travels through the “S” type of groove in the inner square first and groove the edge once again to cut the missing part from the “S” type of groove.

3.32 G25 Circular Groove(This function is limited by absolute coordinate)

Format:

G24 X__Y__Z__R__I__J__D__T__F__

- X Start point of the coordinate
- Y Start point of the coordinate
- Z Depth of Groove
- R Height of outer part
- I The end point in relation to the start point from the incremental X-axis.
- J The end point in relation to the start point from the incremental Y-axis.

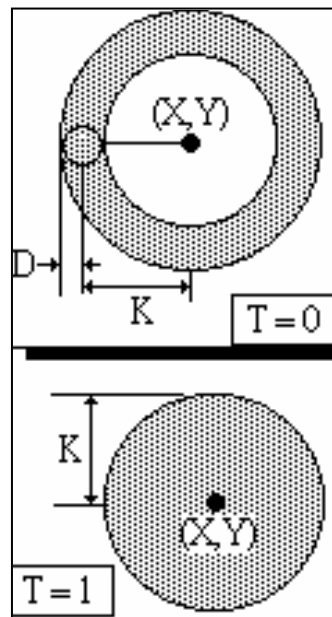
D Tool radius
 T Groove type (0~1)
 F Groove speed

T=0;

1. G00 X(x-k) Y(y)
2. G00 Z(r)
3. G01 Z(z) F(f)
4. G02 I(k) J(0) R(k) F(f)
5. G00 Z(r)

T=1;

1. G00 X(x-k) Y(y)
2. G00 Z(r)
3. G01 U(d) F(f)
4. G01 Z(z) F(f)
5. G02 I(k) J(0) R(k) F(f)
6. IF(k>d) then {[k=k-d]and[go to N3]}
7. G00 Z(r)



3.33 Specified Canned Cycle

The specified canned cycle should go with the command G81~G89. Before using, the user needs to specify the part information of the hole by using the canned cycle. If there is no information on it, the canned cycle can only do the positioning without drillings. Each position from the holes travels at the highest speed (G00).

3.34 G34 Circular Drill Canned Cycle

Format:

G34 X__Y__ I__J__K__F__

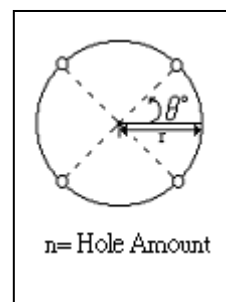
X The center of coordinate
 Y The center of coordinate
 I Radius of circle
 J Angle of the first hole – θ
 K The amount of circular holes – N
 F Drilling speed

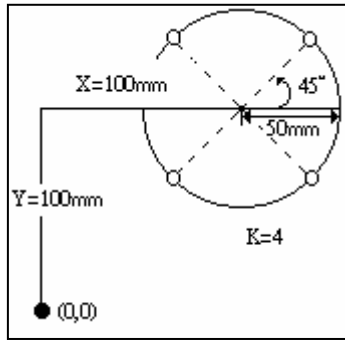
Example:

N001 G81 Z-100. R-50. K0 F100 Definition of drilling resource

N002 G34 X100. Y100. I50. J45. K4

N003 G80 Canned cycle G81 canceled



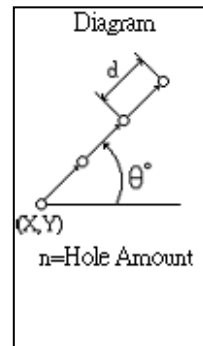


3.35 G35 Angular Drill Canned Cycle

Format:

G35 X__Y__I__J__K__F__

- X Start point of coordinate
- Y Start point of coordinate
- I Drilling Distance
- J Angle of the first hole - θ
- K The amount of circular holes - N
- F Drilling speed

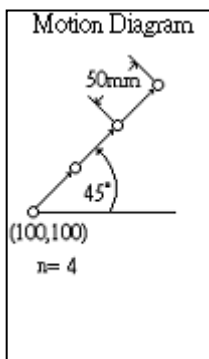


Example:

N001 G81 Z-100. R-50. K0 F100 Definition of drilling resource

N002 G35 X100. Y100. I50. J45. K4

N003 G80 Canned cycle canceled

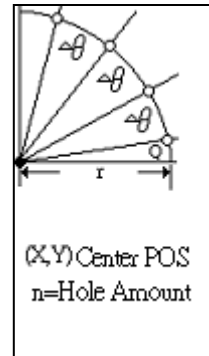


3.36 G36 Arc Drill Canned Cycle

Format:

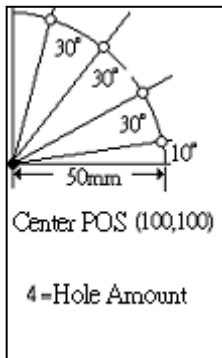
G36 X__Y__I__J__P__K__F__

- X The center of coordinate
- Y The center of coordinate
- I Radius of circle
- J Angle of the first hole – θ
- P Angle of each drilling
- K Drilling amount
- F Drilling speed



Example:

```
N001 G81 Z-100. R-50. K0 F100 Definition of drilling resource
N002 G36 X100. Y100. I50. J10.P30. K4
N003 G80 Canned cycle canceled
```

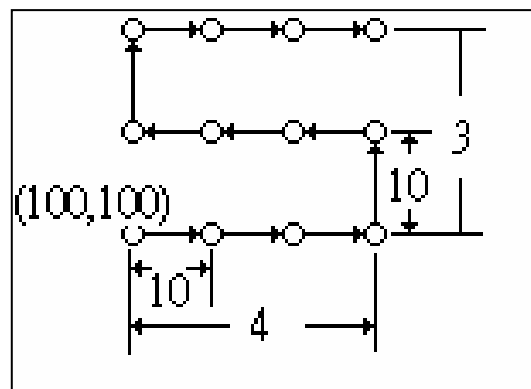


3.37 F37 Grid Drill Canned Cycle

Format:

G37 X__Y__I__P__J__K__F__

- X Start point of coordinate
- Y Start point of coordinate
- I Drilling Distance
- P Drilling numbers of X-axis
- J Distance of Y-axis
- K Drilling numbers of Y-axis
- F Drilling numbers



Example:

```
N001 G81 Z-100. R-50. K0 F100 Definition of drilling resource
N002 G37 X100. Y100. I10.P4 J10. K3
N003 G80 Canned cycle canceled
```


3.38 Auxiliary Functions, M-codes, S-codes

H4M series controller provides M-code functions for users to program certain mechanical actions outside the CNC controller. M-code function consists of a capital letter M followed by a 2-digit number, 00~99. Different M-code represents different action. The following M-codes are the ones used by H4M system and customers should not attempt to change them.

- M00 Program Stop.
When program execution comes to this point, all actions stop, including spindle and the coolant. Press "CYCST" to re-start from where the program was stopped.
- M01 Option Stop. See more details in Sec.6 of Chap 8.
- M02 Program End.
- M30 Program End.
M02 and M30 are identical.
- M98 Sub-program Call.
- M99 Sub-program End.

Except those mentioned above, the remaining M-codes can be defined by users. The following M-codes come with the standard H4M PLC, but users can redefine these M-codes in the PLC if desired.

- M03 Spindle rotation in normal direction.
- M04 Spindle rotation in reversed direction.
- M05 Spindle rotation stops.
- M07 Coolant ON.
- M09 Coolant OFF.

The S-code is used to control the rpm of the spindle rotation. The max. setting is S999999.

Example: S1000 The spindle rpm is 1000 rev/min

For method of decoding the number that associates with M-code and S-code, please refer to Chap 6 of Connecting Manual.

3.39 Sub-program

When a group of program steps will be used repeatedly, these program steps can be grouped in a sub-program that can be called out for execution whenever is required from the main program. In doing this way, the structure of the program can be greatly simplified.

3.39.1 The structure of the sub-program is pretty much the same as the main program except that the sub-program is ended with a M99 as follows:

- 05 Program number (No 5 in this case)
- Program steps
- Program steps

M99 Program end

The sub-program can be independently executed by pressing the "Auto" and "CYCST" button. However, the execution will go round and round to a max. of 8000 times because the sub-program is ended with a M99 function.

3.39.2 Execution of a sub-program from a main program

Format: M98 P____L____

P : Sub-program number

L : Number of execution. If not specified, execute once.

Example: M98 P05 Execute sub-program No 5 once.
 M98 P05 L3 Execute sub-program No 5 three times.

The M98 block can not contain any position code, such as X or Z except those shown in the format. A sub-program can call another sub-program. This stepwise sub-program call can go up to a max. of 5-level for H4M controller as below:

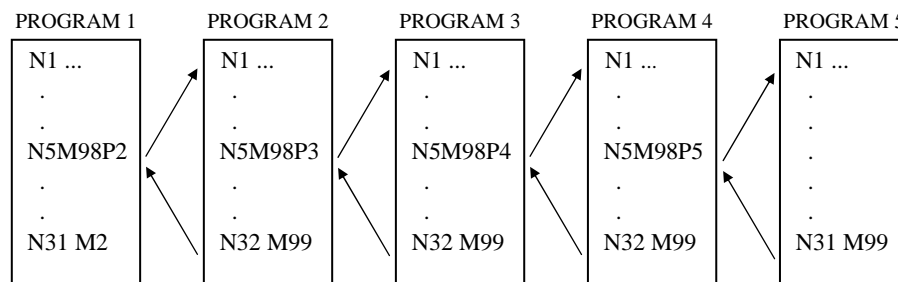


Fig 3-30 Sub-program Call

3.40 MACRO Command, G65

G65 can be used to do some mathematical operations as shown in Table 3-4. It can be applied in the main or sub-program or it can be formed as an independent group of program, to be called upon (M98) from a main program. If you become a master of it, its application is unlimited.

G65 Format: G65 Lm P#i A#j B#k

L, P, A, B : Mathematical command codes in capital letters for G65.

Lm: 'm' is an integer ranging 1~99. 'Lm' represents mathematical operation codes, such as L2 for addition (+), L3 for subtraction (-), L4 for multiplication (*), etc. See Table 3-3 for all 'Lm' definitions.

#i : User defined **variables**, ranging #1~#9999, which are saved when power-off. Variables #10000 and above are controller system variables which you can call out for use but can not change their contents.

1. **P#i** is the location to store the result of mathematical operation. For example P#10=A+B, the result of A+B is stored as variable #10.

- 2. **Pi** (when 'i' is used without a '#' sign) represents the block number for the program execution to branch to if the logic operation is true.
- #j** : Mathematical operand 1. It can be used as either a constant without a '#' sign, i.e. A10, or a variable with a '#' sign, i.e. A#2.
 - 1. **A#j** represents a variable number ranging 1 ~ 9999.
 - 2. **Aj** (when 'j' is used without a '#' sign) represents a constant ranging from "-" 9999999 to "+" 9999999.
- #k** : Mathematical operand 2. It can be used as either a constant without a '#' sign, i.e. B15, or a variable with a '#' sign, i.e. B#7.
 - 1. **B#k** represents a variable number ranging 1 ~ 9999.
 - 2. **Bk** (when 'k' is used without a '#' sign) represents a constant ranging from "-" 9999999 to "+" 9999999.

Ex1: G65 L2 P#10 A#1 B#2

Mathematical meaning is $P\#10 = A\#1 + B\#2$, i.e.

Add the content of variable 1 (A#1) to the content of variable 2 (B#2) and put the result in variable 10 (P#10).

Ex2: G65 L3 P#15 A100 B35

Mathematical meaning is $P\#15 = 100 - 35 = 65$, i.e., put 65 in variable 15.

Ex3: G65 L3 P#15 A100 B#2 (where #2 = 35)

Mathematical meaning is $P\#15 = 100 - 35 = 65$, i.e., put 65 in variable 15.

More Explanations for Variables:

1. Variable #i

- #1~#9999 : User defined **variables**, which are saved when power-off.
- #10000> : Variables #10000 and above are controller system variables which you can call out for use but can not change their contents.

2. All variables '#i, #j, #k' must be integer. '#i' must be positive (+). '#j, #k' can be (+) or (-). If it is negative (-), it means the content value of the variable is reversed before operation.

Ex: If variable #2 = 99, the operation "G65 L01 P#1 A-#2" will result in #1 = -99.

3. The content values of #j, #k must be integer (max 7 digits, + or -). The input unit is depending on decimal format in effect. Refer to Sec. 6.5.

Decimal Point	1 (6/1 format)	2 (5/2 format)	3 (4/3 format)	4 (3/4 format)
Unit	100μm	10μm	1μm	0.1μm
250	25000μm	2500μm	250μm	25μm

Table 3-3 Mathematical Operator Definitions For HUST G65 Command

G-code	L-code	L-code Function	Mathematical Definitions
G65	L1	Equal, Substitution	$\#i = \#j$
G65	L2	Addition	$\#i = \#j + \#k$
G65	L3	Subtraction	$\#i = \#j - \#k$
G65	L4	Multiplication	$\#i = \#j \times \#k$
G65	L5	Division	$\#i = \#j / \#k$
G65	L6	Place Data into Variables	$\#j = \text{Data} = \#i = \#i + 1 = \#i + 2$ $= \dots = \#i + \#k$
G65	L7	Copy Variables	$\#i = \#j$ for consecutive number of $\#k$
G65	L11	Logic OR	$\#i = \#j .\text{OR.} \#k$
G65	L12	Logic AND	$\#i = \#j .\text{AND.} \#k$
G65	L13	Logic exclusive XOR	$\#i = \#j .\text{XOR.} \#k$
G65	L14	ROL, rotate to left	$\#i = \#j .\text{ROL.} \#k$
G65	L15	ROR, rotate to right	$\#i = \#j .\text{ROR.} \#k$
G65	L16	LSL, shift to left	$\#i = \#j .\text{LSL.} \#k$
G65	L17	LSR, shift to right	$\#i = \#j .\text{LSR.} \#k$
G65	L21	Square Root	$\#i = \sqrt{\#j}$
G65	L22	Absolute	$\#i = \#j $
G65	L23	Remainder	$\#i = \#j - \text{trunc}(\#j/\#k) \times \#k$ trunc:(Discard result that is less than 1)
G65	L26	Combined Mul/Div Operation	$\#i = (\#i \times \#j) / \#k$
G65	L31	Sine of an angle (Sin)	$\#i = \#j \times \text{Sin}(\#k)$
G65	L32	Cosine of an angle (Cos)	$\#i = \#j \times \text{Cos}(\#k)$
G65	L34	Arctangent of a value (Tan ⁻¹)	
G65	L50	Obtain Data in Register #j	$\#i = \#j$
G65	L51	Obtain & move I-Bit data to #i	$\#i = \#j$
G65	L52	Obtain & move O-Bit data to #i	$\#i = \#j$
G65	L53	Obtain & move C-Bit data to #i	$\#i = \#j$
G65	L54	Obtain & move S-Bit data to #i	$\#i = \#j$
G65	L55	Obtain & move A-Bit data to #i	$\#i = \#j$
G65	L56	Obtain & move CNT. Operation	$\#i = \#j$
G65	L60	Store Data into Register	$\#i = \#j$
G65	L66	Store Data into Counter	$\#i = \#j$
G65	L80	Unconditional Branching	Execution jumps to block number 'n'

G65	L81	Conditional Branching 1	If #j = #k, Go To n
G65	L82	Conditional Branching 2	If #j ≠ #k, Go To n
G65	L83	Conditional Branching 3	If #j > #k, Go To n
G65	L84	Conditional Branching 4	If #j < #k, Go To n
G65	L85	Conditional Branching 5	If #j ≥ #k, Go To n
G65	L86	Conditional Branching 6	If #j ≤ #k, Go To n
G65	L99	User Designated Error Signals	Error display = i+50 (i=1~49)

Note: The range of computation is from (−9999.999) to (+9999.999).

Mathematical Operation Examples (See Table 3-3)

1. Equal or Substitution, $\#i = \#j$

G65 L1 P#i A#j

Ex: G65 L1 P#10 A150 ($\#10 = 150$)

G65 L1 P#10 A#5 ($\#10 = \#5$. If $\#5=1200$, the result $\#10=1200$.)

G65 L1 P#10 A-#5 ($\#10 = -\#5$, If $\#5=1200$, the result $\#10=-1200$)

2. Addition, $\#i = \#j + \#k$

G65 L2 P#i A#j B#k

Ex: G65 L2 P#1 A#10 B#5

If $\#10=1200$ and $\#5=99$, the result $\#1=1299$.

3. Subtraction, $\#i = \#j - \#k$

G65 L3 P#i A#j B#k

Ex1: G65 L3 P#1 A#10 B#5

If $\#10=1200$ and $\#5=99$, the result $\#1=1101$.

Ex2: G65 L3 P#10 A#10 B5

If $\#10=1200$ before subtraction, then after subtraction $\#10=1200-5=1195$.

4. Multiplication, $\#i = \#j * \#k$

G65 L4 P#i A#j B#k

The result after multiplication should be in the range of (-9999.999~+9999.999). Otherwise, system operation will result in error.

Ex: G65 L4 P#10 A#4 B#30

If $\#4=10$ and $\#30=25$, the result $\#10=10*25=250$.

5. Division, $\#i = \#j / \#k$

G65 L5 P#i A#j B#k

Result that is less than 1 will be discarded.

Ex: G65 L5 P#10 A#5 B#30

If $\#5=130$, $\#30=25$, then $\#10=130/25=5$ (the decimal 0.2 is discarded)

6. Placing Data into Variables, $\#j = \text{Data} = \#i = \#i + 1 = \#i + 2 = \dots = \#i + \#k$

G65 L6 P#i A#j B#k

where: P#i = Starting variable to be operated.

A#j = Data, constant or variable.
 B#k = Number of variables to be operated on, constant or variable.

Meaning: Place the content of variable A#j into a series of variables, starting from the variable P#i and ending at the variable of (P#i + B#k -1).

Ex: Before operation, #10=10, #11=20, #12=30, #13=40, #14=50, #6=5
 G65 L06 P#01 A100 B#6
 After operation, #10= #11= #12= #13= #14=100

7. Copying Variables,

G65 L07 P#i A#j B#k

where: P#i = Starting variable to copy into.
 A#j = Starting variable to be copied from.
 B#k = Number of consecutive variables to be copied.

Ex: Before operation, #01 = 05, #02 = 05, #03 = 05, #04 = 05, #05 = 05,
 #11 = 10, #12 = 20, #13 = 30, #14 = 40, #15 = 50.

G65 L07 P#01 A#11 B5

After operation, #01 = 10, #02 = 20, #03 = 30, #04 = 40, #05 = 50,
 #11 = 10, #12 = 20, #13 = 30, #14 = 40, #15 = 50.

8. Logic OR, #i = #j .OR. #k (operate in binary format)

G65 L11 P#i A#j B#k

Operational principle for logic OR as table below:

A- bit	B-bit	A .OR. B Result
0	0	0
0	1	1
1	0	1
1	1	1

Ex: G65 L11 P#10 A#5 B#20 (#10 = #5 .OR. #20)
 If #5 = 6 = 0000110 in binary and #20 = 100 = 01100100 in binary,
 the result of #10 = #5 .OR. #20 = 01100110 in binary = 102

9. Logic AND, #i = #j .AND. #k (operate in binary format)

G65 L12 P#i A#j B#k

Operational principle for logic AND as table below:

A-bit	B-bit	A .AND. B Result
0	0	0
0	1	0
1	0	0
1	1	1

Ex: G65 L12 P#10 A#5 B#20 (#10 = #5 .AND. #20)

If #5 = 6 = 00000110 in binary and #20 = 100 = 01100100 in binary,
the result of #10 = #5 .AND. #20 = 00000100 in binary = 4

10. Logic XOR, #i = #j .XOR. #k (operate in binary format)

G65 L13 P#i A#j B#k

Operational principle for logic XOR as table below:

A-bit	B-bit	A .XOR. B Result
0	0	0
0	1	1
1	0	1
1	1	0

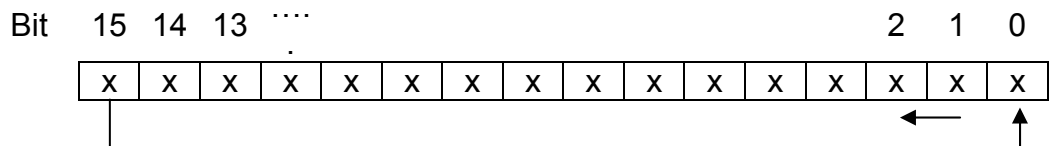
Ex: G65 L13 P#10 A#5 B#20 (#10 = #5 .XOR. #20)

If #5 = 6 = 00000110 in binary and #20 = 100 = 01100100 in binary,
the result of #10 = #5 .XOR. #20 = 01100010 in binary = 98

11. .ROL. #i = #j .ROL. #k (Rotate Left)

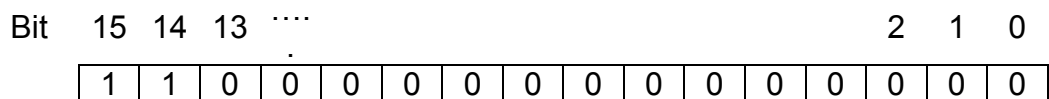
G65 L14 P#i A#j B#k

Meaning: Rotate the 16-bit binary number A#j to the LEFT and place the result in P#i. The number of bits to rotate is indicated by B#k. The bits rotated out to the left will be put at the right.

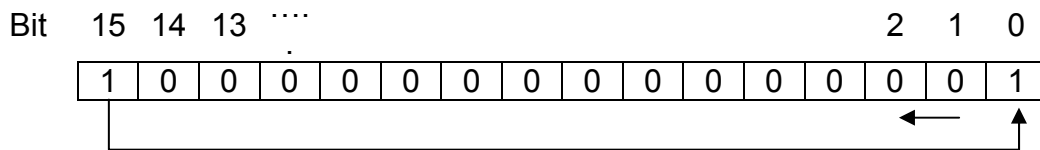


Ex 1: Before G65, variable #10 = 49152
G65 L14 P#12 A#10 B1 (ROL by 1 position)
After G65, variable #12 = 32769

Before:

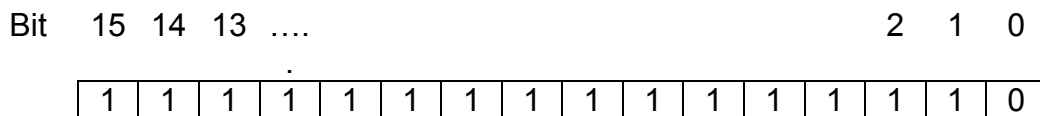


After G65:

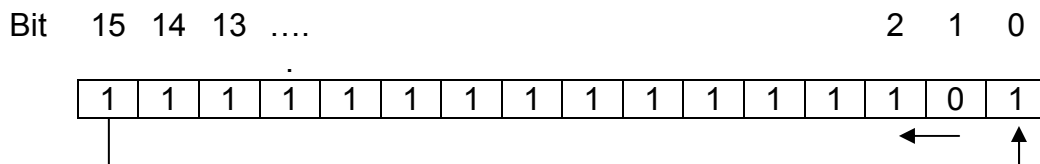


Ex 2: Before G65, variable #10 = -2
 G65 L14 P#12 A#10 B1 (ROL by 1 position)
 After G65, variable #12 = -3

Before:



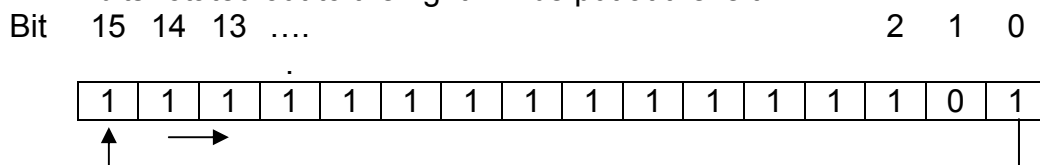
After G65:



12. .ROR. #i = #j .ROR. #k (Rotate Right)

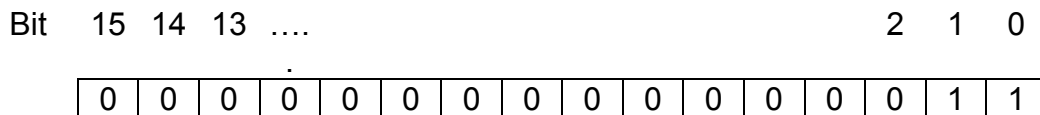
G65 L15 P#i A#j B#k

Meaning: Rotate the 16-bit binary number A#j to the RIGHT and place the result in P#i. The number of bits to rotate is indicated by B#k. The bits rotated out to the right will be put at the left.

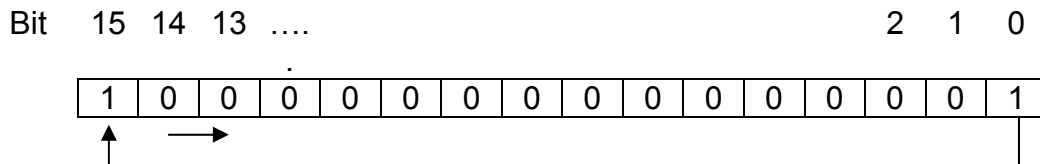


Ex 1: Before G65, variable #10 = 3
 G65 L15 P#12 A#10 B1 (ROR by 1 position)
 After G65, variable #12 = 32769

Before:



After G65:

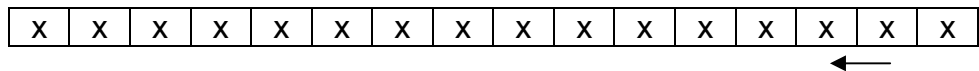


13. .LSL. #i = #j .LSL. #k (Move Left)

G65 L16 P#i A#j B#k

Meaning: Shift the 16-bit binary number A#j to the LEFT and place the result in P#i. The number of bits to shift is indicated by B#k. The bits moved out to the left will be lost and the void spaces at the right will be filled with '0'.

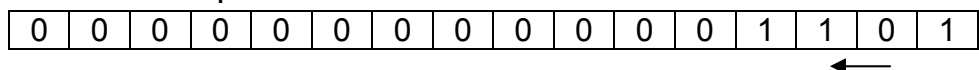
Bit 15 14 13 2 1 0



Ex 1: Before G65, variable #10 = 13
 G65 L16 P#12 A#10 B2 (LSL by 2 positions)
 After G65, variable #12 = 52

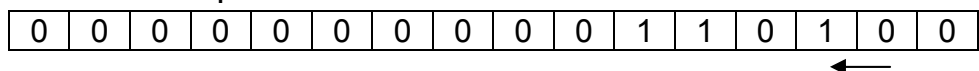
Before:

Bit 15 14 13 2 1 0



After G65:

Bit 15 14 13 2 1 0

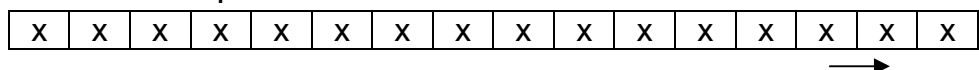


14. LSR #i = #j .LSR. #k (Move Right)

G65 L16 P#i A#j B#k

Meaning: Shift the 16-bit binary number A#j to the RIGHT and place the result in P#i. The number of bits to shift is indicated by B#k. The bits rotated out to the right will be lost and the void spaces at the left will be filled with '0'.

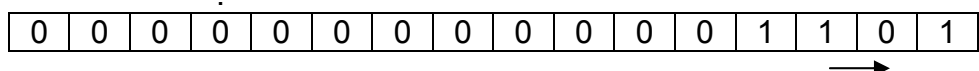
Bit 15 14 13 2 1 0



Ex 1: Before G65, variable #10 = 13
 G65 L17 P#12 A#10 B2 (LSR by 2 positions)
 After G65, variable #12 = 3

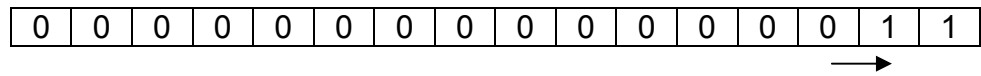
Before:

Bit 15 14 13 2 1 0



After G65:

Bit 15 14 13 2 1 0



15. Square Root, $\#i = \sqrt{\#j}$

G65 L21 P#i A#j (Result that is less than 1 will be discarded)

Ex: G65 L21 P#10 A#5 ($\#10 = \sqrt{\#5}$)
 If #5 = 30, #10 = 5 after G65 operation.

16. Absolute, $\#i = |\#j|$

G65 L22 P#i A#j

Ex: G65 L22 P#10 A#5 ($\#10 = |\#5|$)
 If #5 = -30, #10 = 30 after G65 operation.

17. Remainder, $\#i = \#j - \text{trunc}(\#j / \#k) \times \#k$
 trunc: (Discard result that is less than 1)

G65 L23 P#i A#j B#k

Ex: Find the remainder of (#5/12) with #5 = 99
 G65 L23 P#10 A#5 B12 ($\#10 = \#5 - \text{trunc}(\#5 / 12) \times 12$)
 $\#10 = 99 - \text{trunc}(99 / 12) \times 12 = 99 - (8 \times 12) = 3$

18. Combined Multiplying then Dividing Operation, $\#i = (\#i * \#j) / \#k$

G65 L26 P#i A#j B#k

HUST H4 controller can not handle the multiplied value greater than 9999.999. However, if you use G65 L26 function, the number of digits can exceed 7 digits for the first multiplication operation so long as the final result after division is less than 7 digits. For example, #1=10000, #2=30000, #3=1000, to get the result for (#1)*(#2)/(#3), you thought you could use G65 L04 P#5 A#1 B#2 first, then use G65 L05 P#6 A#5 B#3. But the first operation would yield more than 7 digits and the result would be incorrect. In this case, L26 function can be used as follow to get correct answer.

G65 L26 P#1 A#2 B#3
 $\#1 = (\#1) * (\#2) / (\#3) = 300000$

Ex: G65 L26 P#5 A#10 B#15 ($\#5=120, \#2=15000, \#3=3000$)
 $\#5 = (\#5 * \#10) / \#15 = (120 * 15000) / 3000 = 600$

19. Sine of an Angle, $\#i = \#j \times \text{Sin}(\#k)$

G65 L31 P#i A#j B#k

1. The angle code “k” is in 5/2 format (2 decimals). So, if #k = 4500, it means 45 °.
2. Since Sin(#k) is always less than 1 and HUST H-2 does not operate on decimal (anything smaller than 1 will be discarded), the G65 L31 operation includes a multiplier “#j”.

Ex: Find the value for Sin 60 ° and store it as variable #1

G65 L31 P#1 A1000 B6000

The result P#1 = 1000 * Sin 60 ° = 866

20. Cosine of an Angle, $\#i = \#j \times \text{Cos}(\#k)$

G65 L32 P#i A#j B#k

1. The angle code “k” is in 5/2 format (2 decimals). So, if #k = 4500, it means 45 °.
2. Since Cos(#k) is always less than 1 and HUST H-2 does not operate on decimal (anything smaller than 1 will be discarded), the G65 L31 operation includes a multiplier “#j”.

Ex: Find the value for Cos 45 ° and store it as variable #1

G65 L32 P#1 A1000 B4500

The result P#1 = 1000 * Cos 45 ° = 707

21. Arc-tangent of a number, $\#i = \text{Tan}^{-1} (\#j / \#k)$

G65 L34 P#i A#j B#k

The resulted angle code “i” is in 5/2 format (2 decimals).

Ex: Find the angle for $\text{Tan}^{-1} (577/1000)$ and store it as variable #1

G65 L34 P#1 A577 B1000

The result P#1 = $\text{Tan}^{-1} (577/1000) = 0003000 = 30 °$

22. Obtain Data in Register Number #j and Store in P#i, $\#i = R(\#j)$

G65 L50 P#i A#j (Register #j range = R0 ~ R255)

Ex: G65 L50 P#10 A#5 (#10 = R(#5) = R3, if #5 = 3)

G65 L50 P#10 A31 (#10 = R31)

23. Obtain I-Bit Signal Data in PLC, $\#i = \#j = i(\#j * 16) \dots i(\#j * 16 + 15)$

G65 L51 P#i A#j (#j range = 0 ~ 1 (I000 ~ I023))

16 I-Bit data can be obtained at one time with total of 24 I-bit available according to the value of variable, A#j (0~7) as shown in the example below.

Ex: Obtain data I016~I023 and store in variable #10

G65 L51 P#10 A1

After G65 operation, #10 = I016~I023= 229 (I016 ~ I023 data shown below)

Bit									I23	I22	I21	I20	I19	I18	I17	I16
	0	0	0	0	0	0	0	0	1	1	1	0	0	1	0	1

24. Obtain O-Bit Signal Data in PLC, #I = #j = O(#j * 16)..... O(#j * 16 + 15)

G65 L52 P#I A#j (#j range = 0 (O000 ~ O015))

16 O-Bit data can be obtained at one time with total of 16 O-bit available according to the value of variable, A#j (0) as shown in the example below.

Ex: Obtain data O000~O015 and store in variable #10

G65 L52 P#10 A0

After G65 operation, #10 = O000 ~ 015= 229 (O000 ~ O015 data shown below)

Bit	01	01	01	01	01	01	00	00	00	00	00	00	00	00	00	00
	5	4	3	2		0	9	8	7	6	5	4	3	2	1	0
	0	0	0	0	0	0	0	0	1	1	1	0	0	1	0	1

25. Obtain C-Bit Signal Data in PLC, #I = #j = C(#j * 16)..... C(#j * 16 + 15)

G65 L53 P#I A#j (#j range = 0 ~ 15 (C000 ~ C255))

16 C-Bit data can be obtained at one time with total of 255 C-bit available according to the value of variable, A#j (0~15) as shown in the example below.

Ex: Obtain data C016~C031 and store in variable #10

G65 L53 P#10 A1

After G65 operation, #10 = C016~C031= 229 (C016 ~ C031 data shown below)

Bit	C3	C3	C2	C2	C2	C2	C2	C2	C2	C2	C2	C2	C1	C1	C1	C1
	1	0	9	8	7	6	5	4	3	2	1	0	9	8	7	6
	0	0	0	0	0	0	0	0	1	1	1	0	0	1	0	1

26. Obtain S-Bit Signal Data in PLC, #I = #j = S(#j * 16)..... S(#j * 16 + 15)

G65 L54 P#I A#j (#j range = 0 ~ 15 (S000 ~ S255))

16 S-Bit data can be obtained at one time with total of 255 S-bit available according to the value of variable, A#j (0~15) as shown in the example below.

Ex: Obtain data S016~S031 and store in variable #10

G65 L54 P#10 A1

After G65 operation, #10 =S016~S031= 229 (S016 ~ S031 data shown below)

Bit	S3	S3	S2	S2	S2	S2	S2	S2	S2	S2	S2	S2	S1	S1	S1	S1
	1	0	9	8	7	6	5	4	3	2	1	0	9	8	7	6
	0	0	0	0	0	0	0	0	1	1	1	0	0	1	0	1

27. Obtain A-Bit Signal Data in PLC, #i = #j = A(#j * 16)..... A(#j * 16 + 15)

G65 L55 P#i A#j (#j range = 0 ~ 15 (A000 ~ A255))

16 A-Bit data can be obtained at one time with total of 255 A-bit available according to the value of variable, A#j (0~15) as shown in the examples below.

Ex: Obtain data A016~A031 and store in variable #10

G65 L55 P#10 A1

After G65 operation, #10 =A016~A031= 229 (A016 ~ A031 data shown below)

Bit	A3	A3	A2	A2	A2	A2	A2	A2	A2	A2	A2	A2	A1	A1	A1	A1
	1	0	9	8	7	6	5	4	3	2	1	0	9	8	7	6
	0	0	0	0	0	0	0	0	1	1	1	0	0	1	0	1

Table 3-4 IOCSA-bit with Corresponding A#j-value
For G65 L51, L52, L53, L54, L55 Macro Command

	G65 L51 I-Bit	G65 L52 O-Bit	G65 L53 C-Bit	G65 L54 S-Bit	G65 L55 A-Bit
#j = 0	I000~I015	O000~O015	C000~C015	S000~S015	A000~A015
#j = 1	I016~I023	---	C016~C031	S016~S031	A016~A031
#j = 2	---	---	C032~C047	S032~S047	A032~A047
#j = 3	---	---	C048~C063	S048~S063	A048~A063
#j = 4	---	---	C064~C079	S064~S079	A064~A079
#j = 5	---	---	C080~C095	S080~S095	A080~A095
#j = 6	---	---	C096~C111	S096~S111	A096~A111
#j = 7	---	---	C112~C127	S112~S127	A112~A127
#j = 8	---	---	C127~C143	S127~S143	A127~A143
#j = 9	---	---	C144~C159	S144~S159	A144~A159
#j = 10	---	---	C160~C175	S160~S175	A160~A175
#j = 11	---	---	C176~C191	S176~S191	A176~A191
#j = 12	---	---	C192~C207	S192~S207	A192~A207
#j = 13	---	---	C208~C223	S208~S223	A208~A223
#j = 14	---	---	C224~C239	S224~S239	A224~A239
#j = 15	---	---	C240~C255	S240~S255	A240~A255

28. Obtain Counter Data

G65 L56 P#i A#j (#i = Counter(#j), range 0~255)

Ex: Obtain data (=100) in Counter #10 and store in variable #3.

G65 L56 P#3 A10

After G65 operation, #3 = 100

29. Store Data into Register

G65 L60 P#i A#j (Register #i = #j = 0~255)

Ex: Store data from variable #3 into Register #10. The content of #3 = 100.

G65 L60 P#10 A#3

After G65 operation, Register #10 = 100

30. Store Data into Counter

G65 L66 P#i A#j (Counter #i = #j = 0~255)

Ex: Store data from variable #3 into Counter #10. The content of #3 = 100.

G65 L66 P#10 A#3

After G65 operation, Counter #10 = 100

31. Unconditional Branching

G65 L80 Pn (n = block number)

Program execution jumps to block number 'n'.

Ex: N10 G65 L80 P40

N20 G01 X100.

N30 Y200.

N40 M02

After program executes block N10, execution will skip block N20, N30 and jumps to block number 40. Note that when in G65 the "P" number must match exactly with the program number. For example, P0010 = N0010, but P0010 ≠ N10.

32. Conditional Branching 1 (**Equal**)

G65 L81 Pn A#j B#k (n = block number)

This is equal to the statement "If #j = #k, go to block Pn". Otherwise, execution continues as normal, line by line without interruption.

Ex: N10 G65 L01 P#1 A10

N20 G65 L81 P50 A#1 B10

N30 G01 X100.

N40 Y100.

N50 M02

Set variable #1 = 10 at block N10. After block N20, execution will branch to N50 (skip block N30, N40) because "#1=10" is true.

33. Conditional Branching 2 (**NOT Equal**)

G65 L82 Pn A#j B#k (n = block number)

This is equal to the statement “If #j ≠ #k, go to block Pn”. Otherwise, execution continues as normal, line by line without interruption.

```
Ex: N10 G65 L01 P#1 A20
     N20 G65 L82 P50 A#1 B10
     N30 G01 X100.
N40 Y100.
     N50 M02
```

Set variable #1 = 20 at block N10. After block N20, execution will branch to N50 (skip block N30, N40) because “#1≠10” is true.

34. Conditional Branching 3 (**Greater**)

G65 L83 Pn A#j B#k (n = block number)

This is equal to the statement “If #j > #k, go to block Pn”. Otherwise, execution continues as normal, line by line without interruption.

```
Ex: N10 G65 L01 P#1 A20
     N20 G65 L83 P50 A#1 B10
     N30 G01 X100.
N40 Y100.
     N50 M02
```

Set variable #1 = 20 at block N10. After block N20, execution will branch to N50 (skip block N30, N40) because “#1>10” is true.

35. Conditional Branching 4 (**Smaller**)

G65 L84 Pn A#j B#k (n = block number)

This is equal to the statement “If #j < #k, go to block Pn”. Otherwise, execution continues as normal, line by line without interruption.

```
Ex: N10 G65 L01 P#1 A20
     N20 G65 L84 P50 A#1 B100
     N30 G01 X100.
N40 Y100.
     N50 M02
```

Set variable #1 = 20 at block N10. After block N20, execution will branch to N50 (skip block N30, N40) because “#1<100” is true.

36. Conditional Branching 5 (**Equal or Greater**)

G65 L85 Pn A#j B#k (n = block number)

This is equal to the statement “If #j ≥ #k, go to block Pn”. Otherwise, execution continues as normal, line by line without interruption.

Ex: N10 G65 L01 P#1 A20
N20 G65 L85 P50 A#1 B10
N30 G01 X100.
N40 Y100.
N50 M02

Set variable #1 = 20 at block N10. After block N20, execution will branch to N50 (skip block N30, N40) because “#1≥10” is true.

37. Conditional Branching 6 (**Equal or Smaller**)

G65 L86 Pn A#j B#k (n = block number)

This is equal to the statement “If #j ≤ #k, go to block Pn”. Otherwise, execution continues as normal, line by line without interruption.

Ex: N10 G65 L01 P#1 A20
N20 G65 L86 P50 A#1 B20
N30 G01 X100.
N40 Y100.
N50 M02

Set variable #1 = 20 at block N10. After block N20, execution will branch to N50 (skip block N30, N40) because “#1≤20” is true.

38. User Defined Error Signals (Display = i+50, i=1~49)

G65 L99 Pi

i = 1~49, If 'i' is not in this range, it will display Error 50. User defined error number will be added by 50 because Error 1~49 are HUST system errors.

Ex: G65 L99 P10
Error 60 will be displayed when this block is executed.

Example for G65 Application:

Imagine a roll of plastic (or fabric) being fed into a machine to be cut or perforated according to a fixed pattern on the feed stock. Whole roll of plastic has to be stopped before the cutting action. This example is a part of the program that is used to stop the feed stock for cutting. An electronic light beam is used to detect the recurring pattern so that the feed stock can be stopped. In the program, the I005 bit is the signal from the light beam. When I005=1 (ON), it means the desired pattern is detected. Assume that I005 is the only active signal for I000~ I007 group. So, when I005=1, I000~I007=00100000 in binary = 32 numerically.

Variable: #01 = Total cut length
 #02 = The length for fixed pattern detection (I005)
 #03 = Normal G01 feed-rate
 #04 = Slower feed-rate for fixed pattern detection

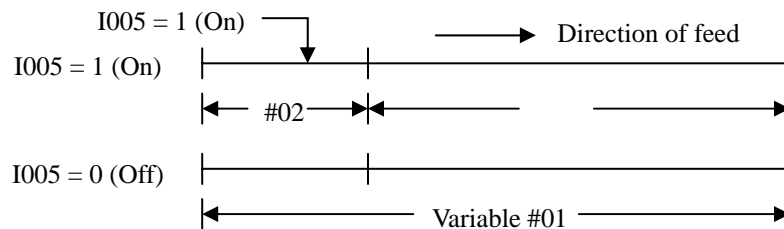


Fig 3-38

G65 L51 P#10 A0	...	Obtain I000~I007 signal
G65 L12 P#11 A#10 B32	...	Check if I005 = 1 (On)
		Note that 32 = 00100000 (binary)
G65 L82 P0010 A#11 B32	...	If I005 ≠ 1, jump to N0010
G65 L84 P0020 A#01 B#02	...	If #01 < #02, jump to N0020
G65 L03 P#12 A#01 B#02	...	#12 = #01 - #02
G01 U#12 F#03		
G31 U#02 F#04	...	Do this when sensor I005 = 1
M02		
N0010 G01 U#01 F#03	...	Do this when sensor I005 = 0
M02		
N0020 G65 L99 P1	...	If #01 < #02, display Error 51
M02		

4 MCM (Machine Constant) PARAMETERS

4.1 MCM Parameter Setting

The MCM parameter allows the user to define certain machine constants that match to the mechanical specifications of the equipment and the machining requirements. The correct and proper setting of these constants enable the user to smoothly operate the equipment. Once they have been set, press RESET key to restart the system.

There is a reference chart of MCM parameters for users to utilize on the next page.

Read and Revise MCM parameters:

(1) To revise or modify MCM parameters on CNC:

1. Enter MDI mode to execute M9999 or press “M” key for 5 seconds to revise the parameter.
2. Each code of parameter should be set from the reference chart on the next page.
3. Press ENTER key to complete the modification. (Some modifications might terminate the power).

PRO-NO :	0	PGM-REF :	
➔ PA#001 = 0.000 PA#002 = 0.000 PA#003 = 0.000 PA#004 = 0.000 PA#005 = 0.000 PA#006 = 0.000 PA#007 = 0.000 PA#008 = 0.000 PA#009 = 0.000 PA#010 = 0.000			
X	0.000	Y	0.000
			Z 0.000
			READY-STOP

Fig4-1

(2) To revise from uploading by RS232C:

Utilizing the software (HCON.exe) of transmission and saving the parameter content as a word file from what has been transmitted into PC such as PE2 or Chinese fonts....etc. After revising the user can download back to the controller.

To clear all the default values

Enter MDI mode to execute G10 P1000

MCM Parameter Setting Table

MCM No.	Setting Axis	Factory Default Setting	Unit	Description	Setting
1	X	0	mm	G54 X-axis 1 st Work coordinate (origin)	
2	Y	0	mm	G54 Y-axis 1 st Work coordinate (origin)	
3	Z	0	mm	G54 Z-axis 1 st Work coordinate (origin)	
4	X	0	mm	G55 X-axis 2 nd Work coordinate (origin)	
5	Y	0	mm	G55 Y-axis 2 nd Work coordinate (origin)	
6	Z	0	mm	G55 Z-axis 2 nd Work coordinate (origin)	
7	X	0	mm	G56 X-axis 3 rd Work coordinate (origin)	
8	Y	0	mm	G56 Y-axis 3 rd Work coordinate (origin)	
9	Z	0	mm	G56 Z-axis 3 rd Work coordinate (origin)	
10	X	0	mm	G57 X-axis 4 th Work coordinate (origin)	
11	Y	0	mm	G57 Y-axis 4 th Work coordinate (origin)	
12	Z	0	mm	G57 Z-axis 4 th Work coordinate (origin)	
13	X	0	mm	G58 X-axis 5 th Work coordinate (origin)	
14	Y	0	mm	G58 Y-axis 5 th Work coordinate (origin)	
15	Z	0	mm	G58 Z-axis 5 th Work coordinate (origin)	
16	X	0	mm	G59 X-axis 6 th Work coordinate (origin)	
17	Y	0	mm	G59 Y-axis 6 th Work coordinate (origin)	
18	Z	0	mm	G59 Z-axis 6 th Work coordinate (origin)	
19	X	0	mm	X-axis, Tool #1 offset compensation	
20	Y	0	mm	Y-axis, Tool #1 offset compensation	
21	Z	0	mm	Z-axis, Tool #1 offset compensation	
22	R	0	mm	Tool #1 radius compensation, NOT available	
23	X	0	mm	X-axis, Tool #2 offset compensation	
24	Y	0	mm	Y-axis, Tool #2 offset compensation	
25	Z	0	mm	Z-axis, Tool #2 offset compensation	
26	R	0	mm	Tool #2 radius compensation, NOT available	
27	X	0	mm	X-axis, Tool #3 offset compensation	
28	Y	0	mm	Y-axis, Tool #3 offset compensation	
29	Z	0	mm	Z-axis, Tool #3 offset compensation	
30	R	0	mm	Tool #3 radius compensation, NOT available	
31	X	0	mm	X-axis, Tool #4 offset compensation	
32	Y	0	mm	Y-axis, Tool #4 offset compensation	
33	Z	0	mm	Z-axis, Tool #4 offset compensation	
34	R	0	mm	Tool #4 radius compensation, NOT available	
35	X	0	mm	X-axis, Tool #5 offset compensation	
36	Y	0	mm	Y-axis, Tool #5 offset compensation	
37	Z	0	mm	Z-axis, Tool #5 offset compensation	
38	R	0	mm	Tool #5 radius compensation, NOT available	
39	X	0	mm	X-axis, Tool #6 offset compensation	
40	Y	0	mm	Y-axis, Tool #6 offset compensation	
41	Z	0	mm	Z-axis, Tool #6 offset compensation	
42	R	0	mm	Tool #6 radius compensation, NOT available	
43	X	0	mm	X-axis, Tool #7 offset compensation	
44	Y	0	mm	Y-axis, Tool #7 offset compensation	
45	Z	0	mm	Z-axis, Tool #7 offset compensation	
46	R	0	mm	Tool #7 radius compensation, NOT available	
47	X	0	mm	X-axis, Tool #8 offset compensation	
48	Y	0	mm	Y-axis, Tool #8 offset compensation	
49	Z	0	mm	Z-axis, Tool #8 offset compensation	
50	R	0	mm	Tool #8 radius compensation, NOT available	
51		0		Master/Slave mode, 0=CNC, 1=X-axis, 2=Y-axis	

MCM No.	Setting Axis	Factory Default Setting	Unit	Description	Setting
52	X	0	mm	X-axis, G28 reference point coordinate	
53	Y	0	mm	Y-axis, G28 reference point coordinate	
54	Z	0	mm	Z-axis, G28 reference point coordinate	
55	X	0	mm	X-axis, G30 reference point coordinate	
56	Y	0	mm	Y-axis, G30 reference point coordinate	
57	Z	0	mm	Z-axis, G30 reference point coordinate	
58	X	0	mm	X-axis, Backlash compensation, 0~255	
59	Y	0	mm	Y-axis, Backlash compensation, 0~255	
60	Z	0	mm	Z-axis, Backlash compensation, 0~255	
61	X	1000	mm/min	X-axis, JOG Feed-rate & power-on G01 speed	
62	Y	1000	mm/min	Y-axis, JOG Feed-rate & power-on G01 speed	
63	Z	1000	mm/min	Z-axis, JOG Feed-rate & power-on G01 speed	
64	X	100	Pulse/rev	X-axis, Denominator, resolution calc. (Encoder pulse)	
65	X	100	μm	X-axis, Numerator, resolution calc. (Ball-screw pitch)	
66	Y	100	Pulse/rev	Y-axis, Denominator, resolution calc. (Encoder pulse)	
67	Y	100	μm	Y-axis, Numerator, resolution calc. (Ball-screw pitch)	
68	Z	100	Pulse/rev	Z-axis, Denominator, resolution calc. (Encoder pulse)	
69	Z	100	μm	Z-axis, Numerator, resolution calc. (Ball-screw pitch)	
70	X	0		X-axis, HOME direction , 0=+ dir., 1=negative dir.	
71	Y	0		Y-axis, HOME direction , 0=+ dir., 1=negative dir.	
72	Z	0		Z-axis, HOME direction , 0=+ dir., 1=negative dir.	
73	X	2500	mm/min	X-axis, HOME speed 1	
74	Y	2500	mm/min	Y-axis, HOME speed 1	
75	Z	2500	mm/min	Z-axis, HOME speed 1	
76	X	40	mm/min	X-axis, Home grid speed during HOME execution	
77	Y	40	mm/min	Y-axis, Home grid speed during HOME execution	
78	Z	40	mm/min	Z-axis, Home grid speed during HOME execution	
79	X	10000	mm/min	X-axis, G00 Traverse speed limit	
80	Y	10000	mm/min	Y-axis, G00 Traverse speed limit	
81	Z	10000	mm/min	Z-axis, G00 Traverse speed limit	
82	X	0		X-axis, Direction of motor rotation, 0=CW, 1=CCW	
83	Y	0		Y-axis, Direction of motor rotation, 0=CW, 1=CCW	
84	Z	0		Z-axis, Direction of motor rotation, 0=CW, 1=CCW	
85	X	4		X-axis, Encoder pulse multiplication factor, 1, 2, or 4	
86	Y	4		Y-axis, Encoder pulse multiplication factor, 1, 2, or 4	
87	Z	4		Z-axis, Encoder pulse multiplication factor, 1, 2, or 4	
88		100	msec	G00 Linear accel./decel. Time, 4~512	
89		100	msec	G01 Linear accel./decel. Time, 4~3072	
90		9600		RS232 Baud rate, 9600, 4800 / EVEN /2 Bit	
91		0		Current counter (M02, M30, M99)	
92		0		Counter limit (Max 9,999,999)	
93	X	9999999	mm	X-axis, Software OT limit, (+) direction	
94	Y	9999999	mm	Y-axis, Software OT limit, (+) direction	
95	Z	9999999	mm	Z-axis, Software OT limit, (+) direction	
96	X	-9999999	mm	X-axis, Software OT limit, (-) direction	
97	Y	-9999999	mm	Y-axis, Software OT limit, (-) direction	
98	Z	-9999999	mm	Z-axis, Software OT limit, (-) direction	
99	X	0	mm	X-axis, HOME shift data	
100	Y	0	mm	Y-axis, HOME shift data	
101	Z	0	mm	Z-axis, HOME shift data	
102				Reserved	
103					

MCM No.	Setting Axis	Factory Default Setting	Unit	Description	Setting
104	X	0		X-axis, Cycle clearing w/ M02, M30, M99	
105	Y	0		Y-axis, Cycle clearing w/ M02, M30, M99	
106	Z	0		Z-axis, Cycle clearing w/ M02, M30, M99	
107	X	1		X-axis,0=incremental coord.,1=absolute coordinate	
108	Y	1		Y-axis,0=incremental coord.,1=absolute coordinate	
109	Z	1		Z-axis,0=incremental coord.,1=absolute coordinate	
110	X	0		Set X-axis as Rotating (1)/Linear axis (0)	
111	Y	0		Set Y-axis as Rotating (1)/Linear axis (0)	
112	Z	0		Set Z-axis as Rotating (1)/Linear axis (0)	
113		0		Reserved	
114	X	64	Pulse	X-axis, Position gain, standard=64	
115	Y	64	Pulse	Y-axis, Position gain, standard=64	
116	Z	64	Pulse	Z-axis, Position gain, standard=64	
117	X	10	Pulse	X-axis, Break-over point for position gain, std=10	
118	Y	10	Pulse	Y-axis, Break-over point for position gain, std=10	
119	Z	10	Pulse	Z-axis, Break-over point for position gain, std=10	
120		1		Flag, R000~R199 to be saved when power off. 0=No, 256=Yes	
121				Reserved	
122				Reserved	
123				Reserved	
124		0		Motor Accel/Decel mode, 0=linear, 1="S" curve	
125		100	msec	Accel/Decel time when in G99 mode (mm/rev)	
126		4096	pulse/rev	Spindle encoder resolution (pulse per revolution)	
127		3000	rpm	Max. spindle rpm @ 10 volts	
128	S	0		Start number for program block number generation	
129	D	0		Increment for program block number generation	
130				Reserved	
131				Reserved	
132		100		Denominator of feed-rate when in MPG test mode	
133		100		Numerator of feed-rate when in MPG test mode	
134				Reserved	
135		100	msec	Time Setting for spindle acceleration	
136	X	0		X-axis, Home grid direction during HOME execution	
137	Y	0		Y-axis, Home grid direction during HOME execution	
138	Z	0		Z-axis, Home grid direction during HOME execution	
139		0	pulse	Servo error when executing in-position function	
140		64	msec	Set Acceleration/Deceleration Time for MPG (4~512)	
141	X	100	Pulse	X-axis, Denominator, MPG resolution calc.	
142	X	100	μm	X-axis, Numerator, MPG resolution calc.	
143	Y	100	Pulse	Y-axis, Denominator, MPG resolution calc.	
144	Y	100	μm	Y-axis, Numerator, MPG resolution calc.	
145	Z	100	Pulse	Z-axis, Denominator, MPG resolution calc.	
146	Z	100	μm	Z-axis, Numerator, MPG resolution calc.	
147		0		0=Metric mode, 1=inch mode	
148		1		Error in Circular Cutting, ideal value=1	
149				Reserved	
:					
160				Reserved	
161	X	0		X-axis allowable compensation of back screw pitch	
162	Y	0		Y-axis allowable compensation of back screw pitch	
163	Z	0		Z-axis allowable compensation of back screw pitch	

MCM No.	Setting Axis	Factory Default Setting	Unit	Description	Setting
164	X	20000		X-axis length compensation of back screw pitch	
165	Y	20000		Y-axis length compensation of back screw pitch	
166	Z	20000		Z-axis length compensation of back screw pitch	
167 : 206			mm	X-axis, Pitch error compensation of each segment.	
207 : 246			mm	Y-axis, Pitch error compensation of each segment.	
247 : 286			mm	Z-axis, Pitch error compensation of each segment.	

Note: press page up and down once can move up to ten pages

The words in bold are common used parameters.

4.2 Description of MCM Machine Constants

In this section the decimal format of the parameters is based on 4/3 format. Item 1~ 18 of MCM parameter is specified as work coordinate from G54~G59. That is to set the location of work origin as its work coordinate. The setting of its work coordinate is with respect to the machine origin. (See Chap 3)

- Format : . , Unit: mm (Default=0.000)
1. G54 (1st) Work Coordinate, X-axis.
 2. G54 (1st) Work Coordinate, Y-axis.
 3. G54 (1st) Work Coordinate, Z-axis.
- Format : . , Unit: mm (Default=0.000)
4. G55 (2nd) Work Coordinate, X-axis.
 5. G55 (2nd) Work Coordinate, Y-axis.
 6. G55 (2nd) Work Coordinate, Z-axis.
- Format : . , Unit: mm (Default=0.000)
7. G56 (3rd) Work Coordinate, X-axis.
 8. G56 (3rd) Work Coordinate, Y-axis.
 9. G56 (3rd) Work Coordinate, Z-axis.
- Format : . , Unit: mm (Default=0.000)
10. G57 (4th) Work Coordinate, X-axis.
 11. G57 (4th) Work Coordinate, Y-axis.
 12. G57 (4th) Work Coordinate, Z-axis.
- Format : . , Unit: mm (Default=0.000)
13. G58 (5th) Work Coordinate, X-axis.
 14. G58 (5th) Work Coordinate, Y-axis.
 15. G58 (5th) Work Coordinate, Z-axis.
- Format : . , Unit: mm (Default=0.000)
16. G59 (6th) Work Coordinate, X-axis.
 17. G59 (6th) Work Coordinate, Y-axis.
 18. G59 (6th) Work Coordinate, Z-axis.
- Format : . , Unit: mm (Default=0.000)
19. X-axis Offset Data, 1st Tool.
 20. Y-axis Offset Data, 1st Tool.
 21. Z-axis Offset Data, 1st Tool.
 22. R-Radius Compensation, Not Available.
- Format : . , Unit: mm (Default=0.000)
23. X-axis Offset Data, 2nd Tool.
 24. Y-axis Offset Data, 2nd Tool.
 25. Z-axis Offset Data, 2st Tool.
 26. R-Radius Compensation, Not Available.
- Format : . , Unit: mm (Default=0.000)

27. X-axis Offset Data, 3rd Tool.
 28. Y-axis Offset Data, 3rd Tool.
 29. Z-axis Offset Data, 3st Tool.
 30. R-Radius Compensation, Not Available.
 Format : . , Unit: mm (Default=0.000)
31. X-axis Offset Data, 4th Tool.
 32. Y-axis Offset Data, 4th Tool.
 33. Z-axis Offset Data, 4st Tool.
 34. R-Radius Compensation, Not Available.
 Format : . , Unit: mm (Default=0.000)
35. X-axis Offset Data, 5th Tool.
 36. Y-axis Offset Data, 5th Tool.
 37. Z-axis Offset Data, 5st Tool.
 38. R-Radius Compensation, Not Available.
 Format : . , Unit: mm (Default=0.000)
39. X-axis Offset Data, 6th Tool.
 40. Y-axis Offset Data, 6th Tool.
 41. Z-axis Offset Data, 6st Tool.
 42. R-Radius Compensation, Not Available.
 Format : . , Unit: mm (Default=0.000)
43. X-axis Offset Data, 7th Tool.
 44. Y-axis Offset Data, 7th Tool.
 45. Z-axis Offset Data, 7st Tool.
 46. R-Radius Compensation, Not Available.
 Format : . , Unit: mm (Default=0.000)
47. X-axis Offset Data, 8th Tool.
 48. Y-axis Offset Data, 8th Tool.
 49. Z-axis Offset Data, 8st Tool.
 50. R-Radius Compensation, Not Available.
 Format : . , Unit: mm (Default=0.000)
51. Master/Slave Mode Setting
 Format : . , (Default=0)
- Setting = 0, CNC mode, Master/Slave mode NOT set.
 = 1, X-axis as master axis, Y-axis; Z-axis as slave axes.
 = 2, Y-axis as master axis, X-axis; Z-axis as slave axes.
 = 3, Z-axis as master axis, X-axis; Y-axis as slave axes.
 = 256 Round Corner Non-stop Operation

MCM parameters #52~#57 are for reference point data. The setting value is the location of each reference point with respect to the machine HOME position.

- 52. G28 Reference Point Data, X-axis.
- 53. G28 Reference Point Data, Y-axis.
- 54. G28 Reference Point Data, Z-axis.
Format : . , Unit: mm (Default=0.000)

- 55. G30 Reference Point Data, X-axis.
- 56. G30 Reference Point Data, Y-axis.
- 57. G30 Reference Point Data, Z-axis.
Format : . , Unit: mm (Default=0.000)

- 58. Backlash Compensation, X-axis.
- 59. Backlash Compensation, Y-axis.
- 60. Backlash Compensation, Z-axis.
Format : . , Unit: pulse (Default=0)
Unit range 0~255 pulses.
If the machine resolution is 1 μm/pulse, setting range = 0~0.255 mm.

- 61. Jog Speed, X-axis and Power-on G01 Speed.
- 62. Jog Speed, Y-axis and Power-on G01 Speed.
- 63. Jog Speed, Z-axis and Power-on G01 Speed.
Format : . , Unit: mm/min (Default=1000)

- 64. Denominator of Machine Resolution, X-axis.
- 65. Numerator of Machine Resolution, X-axis.
- 66. Denominator of Machine Resolution, Y-axis.
- 67. Numerator of Machine Resolution, Y-axis.
- 68. Denominator of Machine Resolution, Z-axis.
- 69. Numerator of Machine Resolution, Z-axis
Format : . , (Default=100)

Denominator (D) = pulses/rev for the encoder on motor.
 Numerator (N) = pitch length (mm/rev) of the ball-screw.
 Gear Ratio (GR) = Tooth No. on ball-screw / Tooth No. on motor.
 Pulse Multiplication Factor (MF) = MCM #85, #87.

$$\text{Machine Resolution} = \frac{(\text{Pitch of Ball - screw})}{(\text{Encoder Pulse}) * (\text{MF})} * \frac{1}{\text{GR}}$$

Ex1: X-axis as linear axis (MCM #110=0), pitch = 5 mm = 5000 μm
 Encoder = 2500 pulses, MCM #85 = 4, and GR = 5 (motor rotates 5 times while ball-screw rotates once)

Machine resolution = 5000/(2500 x 4)/5 = 5000/50000 = 1/10 = 0.1 μm/pulse

Therefore, the setting value for MCM #64 (D) and #69 (N) can be one of the three combinations. They are all correct.

(1) D=10000, N=1000 (2) D=10, N=1 (3) D=100, N=10

Ex2: Y-axis as rotating axis (MCM #111=1), Angle = 360.000 deg/circle
Encoder = 2500 pulses, MCM #86 = 4, and GR = 5 (motor rotates 5 times while ball-screw rotates once)

Machine resolution = $360000 / (2500 \times 4) / 5 = 360000 / 50000 = 36/5 = 72/10$

Therefore, the setting value for MCM #66 (D) and #67 (N) can be one of the three combinations. They are all correct.

(1) D=5, N=36 (2) D=10, N=72 (3) D=50000, N=360000

Caution: When machine resolution $< 1/100$, software travel limit (MCM #93 and #96) should be in the range of $-999999 \sim +999999$. Otherwise, error message will appear. For example, if MCM #64=400 and #65=2, the machine resolution = $2/400 = 1/200 < 1/100$. So, the value for MCM #93 should be less than 999999 and the value for #96 greater than -999999 .

- 70. Home Direction for Tool, X-axis.
 - 71. Home Direction for Tool, Y-axis.
 - 72. Home Direction for Tool, Z-axis.
- Format : , (Default=0)

Setting = 0, Tool returning to HOME in the positive direction.

Setting = 1, Tool returning to HOME in the negative direction

- 73. Home Speed When Tool Going to Home, X-axis.
 - 74. Home Speed When Tool Going to Home, Y-axis.
 - 75. Home Speed When Tool Going to Home, Z-axis.
- Format : , Unit: mm/min (Default=2500)

- 76. Home Grid Speed When Tool Going to Home, X-axis.
 - 77. Home Grid Speed When Tool Going to Home, Y-axis.
 - 78. Home Grid Speed When Tool Going to Home, Z-axis.
- Format : , Unit: mm/min (Default=200)

Set the moving speed when the tool, after having touched the HOME limit switch, is searching for the encoder grid signal during HOME execution. HUST H3X CNC has three (3) different speeds when you execute HOME function as shown by Fig 7.2.

Speed 1: The motor accelerates to Speed 1 and its maximum speed is determined by the settings of MCM #73, #74, #75 and the direction by MCM #70, #71, #72. When tool touches the home limit switch, it starts deceleration to a stop.

Speed 2: The motor accelerates again to speed 2 and its maximum speed is equal to 1/4 of Speed 1 and the direction is by MCM #136, #137, #138. When tool starts leaving the home limit switch, it starts deceleration to a stop.

Speed 3: The motor accelerates to speed 3 and its maximum speed is determined by the settings of MCM #76, #77, #78 and the direction by MCM #136, #137, #138. Once the encoder grid index is found, motor decelerates to a stop. This is the HOME position.

Note that the length of the Home limit switch should be longer than the distance for the deceleration of Speed 1. Otherwise, serious error may result. The equation to calculate the length of the Home limit switch is

$$\text{Length of Home Limit Switch (mm)} \geq \frac{\text{FDCOM} * \text{ACC}}{60000}$$

FDCOM = Speed 1, in mm/min. (MCM #73, #74, #75)
 ACC = Time for acceleration/deceleration, in ms. (MCM #89)
 60000 = 60 seconds = 60 * 1000 milliseconds

When the C-bit C063=1 in PLC program, it commands the controller to do homing operation. Do homing operation for X-axis if R232=1, do Y-axis if R232=2, do Z-axis if R232=4 and do three axes simultaneously if R232=7.

Ex: FDCOM = 3000.00 mm/min, and ACC = 100 ms
 Length of Home Limit Switch = 3000 * 100 / 60000 = 5 mm

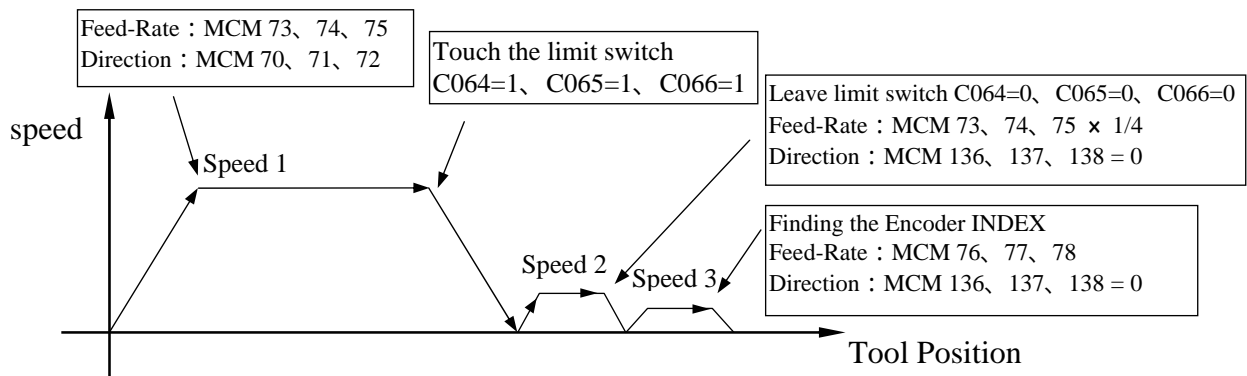


Fig.4-2(A)

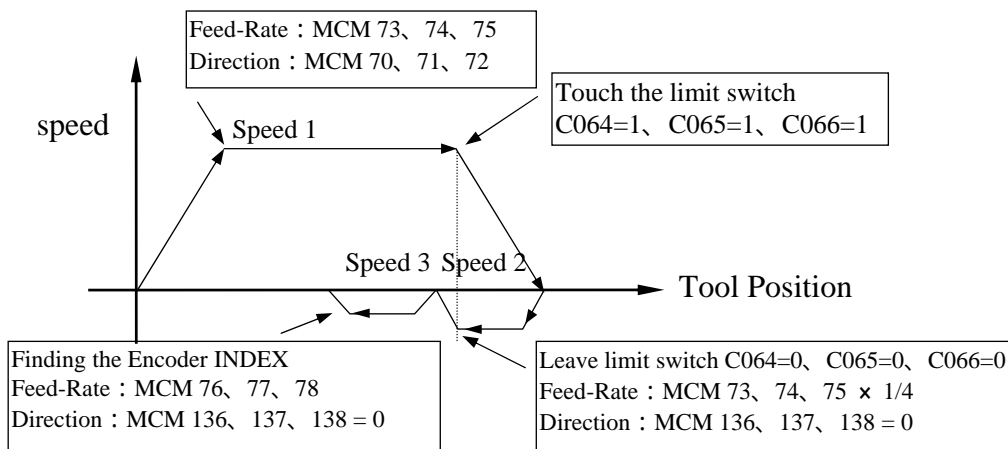


Fig.4-2(B)

79. Traverse Speed Limit, X-axis (G00).
 80. Traverse Speed Limit, Y-axis (G00).
 81. Traverse Speed Limit, Z-axis (G00).
 Format : , Unit: mm/min (Default=10000)

Set the max. moving speed for G00 function in each axis. The traverse speed limit can be calculated from the following equation:

$$F_{max} = 0.95 * RPM * Pitch * GR$$

RPM: The max. rpm of servo motor
 Pitch: The pitch of the ball-screw
 GR : Gear ratio of ball-screw/motor

Ex: Max. rpm = 3000 rpm for X-axis, Pitch = 5 mm/rev, Gear Ratio = 5/1
 $F_{max} = 0.95 * 3000 * 5 / 5 = 2850$ mm/min
 Therefore, it is recommended to set MCM #79=2850.

82. Direction of Motor Rotation, X-axis.
 83. Direction of Motor Rotation, Y-axis.
 84. Direction of Motor Rotation, Z-axis.
 Format : , (Default=0)

Setting = 0, Motor rotates in the positive direction. (CW)
 Setting = 1, Motor rotates in the negative direction. (CCW)

This MCM can be used to reverse the direction of motor rotation if desired. So you don't have to worry about the direction of rotation when installing motor. These parameters will affect the direction of HOME position

Attention: Motor Divergence

Due to the variations in circuit design of the servo drivers that are available from the market, the proper electrical connections from servo encoder to the driver, then to the CNC controller may vary. If the connections do not match properly, the motor RPM may become divergent (Rotate @ HIGH RPM) and damage to the machine may result. For this reason, HUST strongly suggest separate the servo motor and the machine before you are 100% sure the direction of the motor rotation. If a motor divergence occurs, please inter-change the connections of (A and B phase) and (A- and B-phase) on the driver side.

85. Encoder Multiplication Factor, X-axis.
 86. Encoder Multiplication Factor, Y-axis.
 87. Encoder Multiplication Factor, Z-axis.
 Format : , (Default=0)

Only one the following 3 numbers:

Setting = 1, Encoder pulse number is multiplied by 1.
 Setting = 2, Encoder pulse number is multiplied by 2.
 Setting = 4, Encoder pulse number is multiplied by 4.

Ex: If factor = 2 for MCM #85 and the encoder resolution is 2000 pulses/rev,
 then the feed-back signals = $2000 * 2 = 4000$ pulses/rev for Y-axis.

88. Servo Motor Acceleration/Deceleration Time, G00.
 Format : _____ , Unit: millisecond (Default=100)
 Setting Range: 4 ~ 512 millisecond

If MCM #124 setting = 0, type of accel./decel. for G00 = Linear.
 If MCM #124 setting = 1, type of acceleration/deceleration for G00 = "S" curve.
 In this case, the actual acceleration/deceleration time is twice the setting value.

89. Servo Motor Acceleration/Deceleration Time (T), G01.
 Format : _____ , Unit: millisecond (Default=100)
 Setting Range: 4 ~ 1024 millisecond.
 100 milliseconds is the recommended setting for both G00 and G01.

90. RS232C Baud Rate.
 Format : _____ , (Default = 9600)

Set RS232C communication speed. Choose from, 4800, 9600, 19200,38400,
 27600, Speed rate 9600 stands for 9600 bits per second.

In addition, use the following settings for your PC:

Parity -- Even
 Stop Bits-- 2 bits

91. Current Counter.
 Format : _____ , (Default=0)

Record the number of cycles that the current program has been executed.
 Every time the program execution encounters M02, M30 and M99, it will add
 one (1) to the counter. When the number is equal to the number as set in
 MCM #92 (Counter limit), the program execution will stop. This parameter is
 effective only if C030=1 in the PLC. (See Connecting Manual)

Note that when MCM #91 = MCM #92 and you attempt to execute the program,
 error 03 will be the result. When this happens, you have to clear MCM #91 or
 reset MCM #92 to higher value.

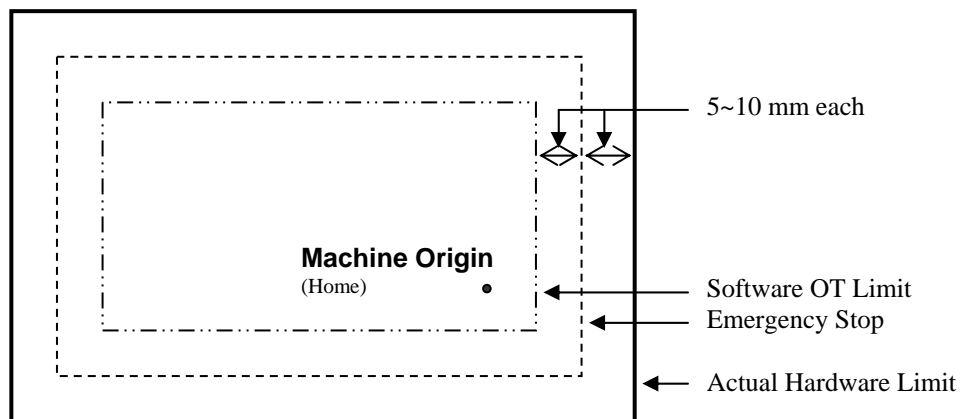
92. Counter Limit.
 Format : _____ , (Default=0)
 Maximum setting value = 9,999,999.

93. Software OT Limit in (+) Direction, X-axis
 94. Software OT Limit in (+) Direction, Y-axis
 95. Software OT Limit in (+) Direction, Z-axis.
 Format : _____ , Unit: mm/min (Default=9999.999)

Set the software over-travel (OT) limit in the positive (+) direction, the setting value is equal to the distance from positive OT location to the machine origin (HOME).

96. Software OT Limit in (-) Direction, X-axis
 97. Software OT Limit in (-) Direction, Y-axis
 98. Software OT Limit in (-) Direction, Z-axis.
 Format : . , Unit: mm/min (Default=-9999.999)

Set the software over-travel (OT) limit in the negative (-) direction, the setting value is equal to the distance from negative OT location to the machine origin (HOME). Figure below shows the relationship among the software OT limit, the emergency stop, and the actual hardware limit.



Setting by MCM#93 98

Fig.4-3

99. Home-Shift Data, X-axis
 100. Home-Shift Data, Y-axis
 101. Home-Shift Data, Z-axis
 Format : . , Unit: mm/min (Default=0.000)

Set the amount of coordinate shift for HOME location (or machine origin). With these settings, the machine coordinate will be shifted by the same amount when you execute "Home". If home shift data are zero for all axes, the machine coordinate after "Home" operation will be zero also. Note that the work coordinate will be shifted by the same amount.

102. Reserved
 103. Reserved
 104. Flag to Clear X-axis Program Coordinate on M02, M30 or M99 Command.
 105. Flag to Clear Y-axis Program Coordinate on M02, M30 or M99 Command.
 106. Flag to Clear Z-axis Program Coordinate on M02, M30, or M99 Command.
 Format : , (Default=0)

Used as flag to clear the coordinate when program execution encounters M02, M30 or M99 function. The following settings are valid for both X and Y-axis.

Setting = 0, Flag is OFF, NOT to clear.

Setting = 1, Flag is ON, YES to clear when encountering M02 and M30.

Setting = 2, Flag is ON, YES to clear when encountering M99.

Setting = 3, Flag is ON, YES to clear when encountering M02, M30 and M99.

107. Set Incremental/Absolute Mode, X-axis coordinate.

Format: , (Default=1)

Setting 0=X Incremental mode, U is not valid.

1=X Absolute mode, U is incremental coordinate.

108. Set Incremental/Absolute Mode, Y-axis coordinate.

Format: , (Default=1)

Setting 0=Y Incremental mode, V is not valid.

1=Y Absolute mode, V is incremental coordinate.

109. Set Incremental/Absolute Mode, Z-axis coordinate.

Format : , (Default=1)

Setting 0=Z Incremental mode, W is not valid.

1=Z Absolute mode, W is incremental coordinate.

110. Set X-axis as Rotating/Linear axis.

111. Set Y-axis as Rotating/Linear axis.

112. Set Z-axis as Rotating/Linear axis.

Format : , (Default=0)

Setting = 0, Use as linear axis.

Setting = 1, Use as rotating axis.

113. Reserved for system and set zero

114. Set Position Gain, X-axis.

115. Set Position Gain, Y-axis.

116. Set Position Gain, Z-axis.

Format : , (Default=64)

Setting Range: 8 ~ 640

Set the position gain for encoder feedback. Recommended setting value is 64.

The proper setting of this MCM is very important in assuring the smooth operation of the servo motor. Do not change once the correct setting is determined.

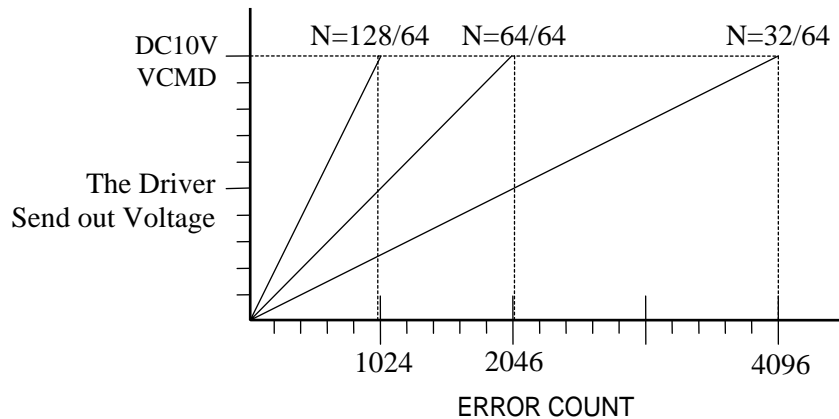


Fig 4-4 The relationship of Driver V-command and Servo Error

Recommended servo motor adjusting procedure:

1. Adjust the servo driver based on its operation manual.
2. Adjust the encoder multiplication factor (MCM # 85 ~# 87). Under the normal condition, the servo error will be blinking between 0~1 if the motor is properly locked. If it's blinking between 4~5, adjust MCM # 85~#87 by lowering its multiplication factor.
3. Adjust the position gain of encoder feed-back. (MCM # 114 ~ # 116)

The equations to calculate position gain and CNC V-command are:

$$\text{Position Gain} = \frac{(\text{Value of MCM\#85 or \#87})}{64}$$

$$\text{CNC V - command} = (\text{Position Gain}) * (\text{Servo Error}) * \left(\frac{10 \text{ V}}{2048}\right)$$

HUST controller adopts a closed-circuit system. The servo-error is the difference in pulses between the CNC V-command and the motor encoder feedback. Then, the next V-command from CNC will be properly adjusted based on this error.

If Servo-error > 4096, a message of "Error 2" will be displayed.

In this case, please increase the setting value for MCM #114~#116 and press RESET. If the problem still exists, please check the motor and the driver for proper connection.

117. Break-over Point (in Error Count) for Position Gain, X-axis.
 118. Break-over Point (in Error Count) for Position Gain, Y-axis.
 119. Break-over Point (in Error Count) for Position Gain, Z-axis.
- Format : , (Default=10)

The proper setting of this parameter will assure smooth start-up of servo motor. When servo error is smaller than the setting value of MCM #117~#119, the position gain is 64. Otherwise, position gain will be calculated based on the

setting value of MCM #114~ #116 and the setting values depend on the frictional load on the motor. If the frictional load is high, setting value is small and vice versa.

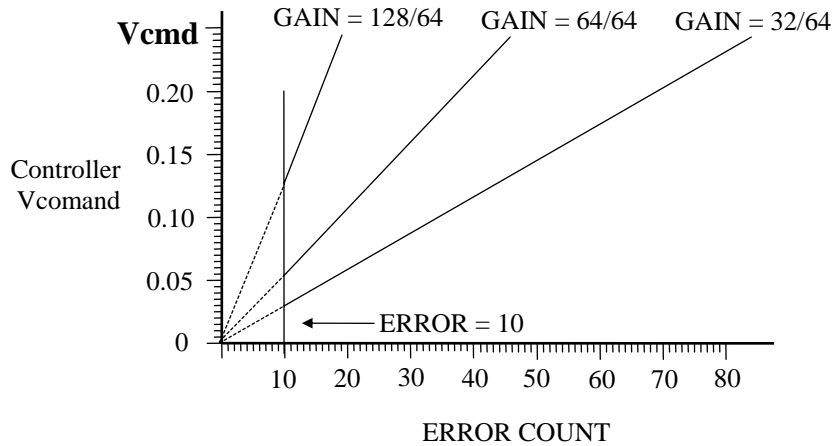


Fig 4-5 Break-over of Position Gain

120. Flag to Save the Data of R000~R199 in PLC when power-off.
 Format : , (Default=0)

In case of power interruption, the important data in your program can be saved through registers R000~R199 in PLC program with proper setting of MCM #120.

Setting = 0, NOT to save.
 Setting = 256, Save R000~R199 data.

121. Reserved
 122. Reserved
 123. Reserved

124. Type of Motor Acceleration/Deceleration
 Format : , (Default=0)

Setting = 0, Linear type.
 Setting = 1, "S" curve.

When the setting = 1, all linear type will be replaced by "S-curve".

125. Acceleration/Deceleration Time for G99 Mode.
 Format : , Unit: Millisecond (Default=100)
 Setting Range: 4 ~ 1024 ms.

126. Spindle Encoder Pulse Per Revolution
 Format : , Unit: Pulse/rev (Default=4096)

Setting value is equal to the encoder spec times the factor from MCM #85.

- I. When Spindle set on X coordinate, the value is equal to Size of the Encoder Pulse x Rate of MCM#85

- II. When Spindle set on Y coordinate, the value is equal to Size of the Encoder Pulse x Rate of MCM#86
- III. When Spindle set on Z coordinate, the value is equal to Size of the Encoder Pulse x Rate of MCM#87

127. Set Spindle Motor RPM When Vcmd = 10 Volt.
Format : , Unit: RPM (Default=3000)

128. Starting Number for Auto Generation of Program Block Number.
Format : S= , (Default=0)

129. Increment for Auto-generation of Program Block Number.
Format : D= , (Default=0)

If D = 0, the block number will NOT be auto-generated. If D ≠ 0, the block number will be automatically generated when you press NEW LINE key under EDIT and TEACH mode. Press RESET will restart the number again from MCM #128, #129.

Ex: MCM #128=0, #129=5, the generated block numbers are 5, 10, 15, 20, 25---.

130. Reserved

131. Reserved

132. Denominator of Feed-rate Multiplication Factor for MPG Test.

133. Numerator of Feed-rate Multiplication Factor for MPG Test.

Format : , (Default=100)

134. Reserved

135. Set Spindle for acceleration / Deceleration
Format: (Default=100) ; Unit: milliseconds

136. Home Grid Speed When Tool Going to Home, X axis.

137. Home Grid Speed When Tool Going to Home, Y axis.

138. Home Grid Speed When Tool Going to Home, Z axis.

Format: , Default=0

Set the moving speed (MCM# 136,#137,#138) when the tool, having touched the HOME limit switch, is searching for the encoder grid signal during HOME execution.

Setting =0, when the encoder from X-axis returning to HOME, Speed 2, 3 is moving away from Limit Switch to search the HOME GRID, the direction is opposite to speed 1. See fig.7-2 (A)

Setting =-1, when the encoder from X-axis returning to HOME, Speed 2, 3 is moving away from Limit Switch to search the HOME GRID, the direction is as the same as speed 1. See fig.7-2 (B)

139. Following Count

Format= , Default=0, Unit: pulse

Setting=0 the function is NOT valid.

Setting 0 (1, 2, 3, 4....) the following count is in effect.

In the case where CNC controller has finished sending all pulses to the motor, yet the motor is falling behind substantially in pulse execution (servo error count). If you start the next block without waiting for motor to finish execution, cutting error may result. To overcome this problem, you set a number for Following Count. When this function is in effect, the CNC execution will wait until the motor execution is within the number pulses you set for.

140. Set Acceleration / Deceleration Time for MPG

Format= , (Default=64), Unit: milliseconds

Setting Range: 4~512 ms.

The motor acceleration / deceleration time is equal to MCM #140 when MPG hand-wheel is used in JOG mode.

141. X-axis Denominator, MPG Hand-wheel Resolution Adjustment. (pulse)

142. X-axis Numerator, MPG Hand-wheel Resolution Adjustment. (μm)

143. Y-axis Denominator, MPG Hand-wheel Resolution Adjustment. (pulse)

144. Y-axis Numerator, MPG Hand-wheel Resolution Adjustment. (μm)

Format : , (Default = 100)

Unit: Denominator = pulses, Numerator = μm

145. Z-axis Denominator, MPG Hand-wheel Resolution Adjustment. (pulse)

146. Z-axis Numerator, MPG Hand-wheel Resolution Adjustment. (μm)

Format : , (Default = 100)

Ex1: For X-axis, MCM #141 = 100 pulses, MCM #142 = 100 μm .

The resolution for X-axis = $100/100 = 1 \mu\text{m}/\text{pulse}$.

If MPG hand-wheel moves 1 notch (=100 pulses), the feed length in X-axis = $100 \times (100/100) = 100 \mu\text{m} = 0.1 \text{ mm}$.

Ex2: For Y-axis, MCM #143 = 200 pulses, MCM #144 = 500 μm .

The resolution for Y-axis = $500/200 = 2.5 \mu\text{m}/\text{pulse}$.

If MPG hand-wheel moves 1 notch (=100 pulses), the feed length in Y-axis = $100 \times (500/200) = 250 \mu\text{m} = 0.25 \text{ mm}$.

147. METRIC/INCH Mode Selection (default = 0)

Format : , (Default = 0)

Setting = 0, Measurement in METRIC unit.

Setting = 1, Measurement in INCH unit.

148. Error in Circular Cutting

Format : , (Default = 1)

Range: 1 ~ 32

In circular cutting, the ideal cutting path is a circular arc, but the actual motor path is along the arc cord (a straight line). Therefore, there is a cutting error as shown in the figure below.

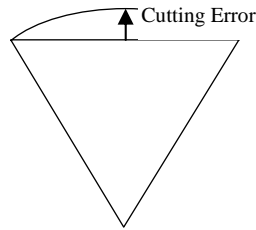


Fig 4-6

This parameter enables the user to adjust acceptable error. The smaller is the setting (=1, the best), the better the circular cutting result. However, the setting should not be too small to the point that it's not able to drive the motor.

MCM# 161~286 are for pitch-error compensation.

The reference point of HUST H3X for pitch error compensation is the machine origin (HOME)

161. Pitch Error Compensation Mode Setting, X-axis.

162. Pitch Error Compensation Mode Setting, Y-axis.

163. Pitch Error Compensation Mode Setting, Z-axis.

Format: , Default=0

Setting=0, Compensation canceled.

Setting=-1, Negative side of compensation.

Setting=1, Positive side of compensation.

X-axis	Y-axis	Explanation
0	0	Compensation cancel
-1	-1	Do compensation when tool is on the (-) side of the reference point
1	1	Do compensation when tool is on the (+) side of the reference point.

Ex:

MCM # 121=-1

The pitch error in the X-axis will not be compensated when the tool travels to the positive side of the X-HOME location. It will be compensated when the tool travels to the negative side of machine origin.

MCM #122=1

The pitch error in the Y-axis will be compensated when the tool travels to the positive side of Y-HOME location. No compensation will be done when it travels to the negative side of machine origin.

164. Segment Length for Pitch Error Compensation, X-axis.

165. Segment Length for Pitch Error Compensation, Y-axis.

166. Segment Length for Pitch Error Compensation, Z-axis.

Format= . , Default=0, Unit=mm

Axis	Corresponding MCM# for Segment Length	Segment Length	Max. Number of Segment
X	MCM#167~#206	20~480mm	40
Y	MCM#207~#246	20~480mm	40
Z	MCM#247~#286		

1. Segment length is the total length of ball-screw divided by the number of segment.

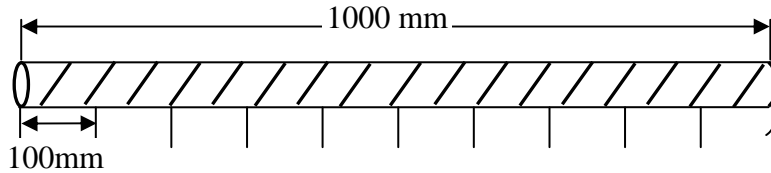


Fig 4-7

Ex: If you want to divide the ball-screw on X-axis, which is 1 meter in length, into 10 segments, the segment length is $1000.00/10=100.00$ mm. This 100.00 mm will be stored in MCM# 164.

2. If the average segment length is less than 20 mm, use 20 mm.
3. When doing compensation, HUST H4 controller will further divide each segment into 8 sections. The amount of compensation for each section is equal to the whole number, in μ m, of 1/8 of the amount in MCM #167~#206. The remainder will be added to the next section.

Ex: Segment length=100.00mm and the amount of compensation is 0.026mm as set in MCM#164. Then, the compensation for each section is $0.026/8=0.00325$ mm. The compensation for this segment will be done in a manner as tabulated below:

Section	Tool Position	Avg. comp. For each section	Actual comp. At each section	Accumulated compensation
1	12.5	0.00325	0.003	0.003
2	25	0.00325	0.003	0.006
3	37.5	0.00325	0.003	0.009
4	50	0.00325	0.004	0.013
5	62.5	0.00325	0.003	0.016
6	75	0.00325	0.003	0.019
7	87.5	0.00325	0.003	0.022
8	100	0.00325	0.004	0.026

167~286. Amount of Compensation for each segment(X.Y.Z-axis) is 40.

The Compensation value is in incremental mode. If the number of segment is less than 40, please fill the uncompensated segments with zero to avoid any potential errors.

Ex: If the segment of compensation is 10, the amount of the compensation from Seg.11 to 40 (MCM# 1787~#126, MCCM#217~#246 of Y-axis, MCM# 257~ #286 of Z-axis) must be set as zero.

167~206. Pitch error compensation of each segment, X-axis. (unit: mm)

MCM# Parameter	Default	Compensation
167	0	
168	0	
169	0	
170	0	
171	0	
172	0	
173	0	
174	0	
175	0	
176	0	
177	0	
178	0	
179	0	
180	0	
181	0	
182	0	
183	0	
184	0	
185	0	
186	0	
187	0	
188	0	
189	0	
190	0	
191	0	
192	0	
193	0	
194	0	
195	0	
196	0	
197	0	
198	0	
199	0	
200	0	
201	0	
202	0	
203	0	
204	0	
205	0	
206	0	

207~246. Pitch error compensation of each segment, Y-axis. (unit: mm)

MCM# Parameter	Default	Compensation
207	0	
208	0	
209	0	
210	0	
211	0	
212	0	
213	0	
214	0	
215	0	
216	0	
217	0	
218	0	
219	0	
220	0	
221	0	
222	0	
223	0	
224	0	
225	0	
226	0	
227	0	
228	0	
229	0	
230	0	
231	0	
232	0	
233	0	
234	0	
235	0	
236	0	
237	0	
238	0	
239	0	
240	0	
241	0	
242	0	
243	0	
244	0	
245	0	
246	0	

247~286. Pitch error compensation of each segment, Z-axis. (unit: mm)

MCM# Parameter	Default	Compensation
247	0	
248	0	
249	0	
250	0	
251	0	
252	0	
253	0	
254	0	
255	0	
256	0	
257	0	
258	0	
259	0	
260	0	
261	0	
262	0	
263	0	
264	0	
265	0	
266	0	
267	0	
268	0	
269	0	
270	0	
271	0	
272	0	
273	0	
274	0	
275	0	
276	0	
277	0	
278	0	
279	0	
280	0	
281	0	
282	0	
283	0	
284	0	
285	0	
286	0	

5 CONNECTIONS

5.1 Connecting System Descriptions

This manual explains the H4M controller's electrical as well as structural design necessary for connecting the CNC to the machine tool. This manual is intended for users with some basic electrical and electronic knowledge.

Fig5-1, represents the common peripheral of H4M series. They are categorized as follows:

1. The system can be remotely controlled by PC, HUST CRT screen and PC touch screen through RS232C interface.
2. The system can handle thumb switch signals through PLC ladder program.
3. The system can control both AC servo and stepping motor.
4. The system provides DI/DO=24/16 points. Also included are connections for MPG, spindle and Skip sensor.

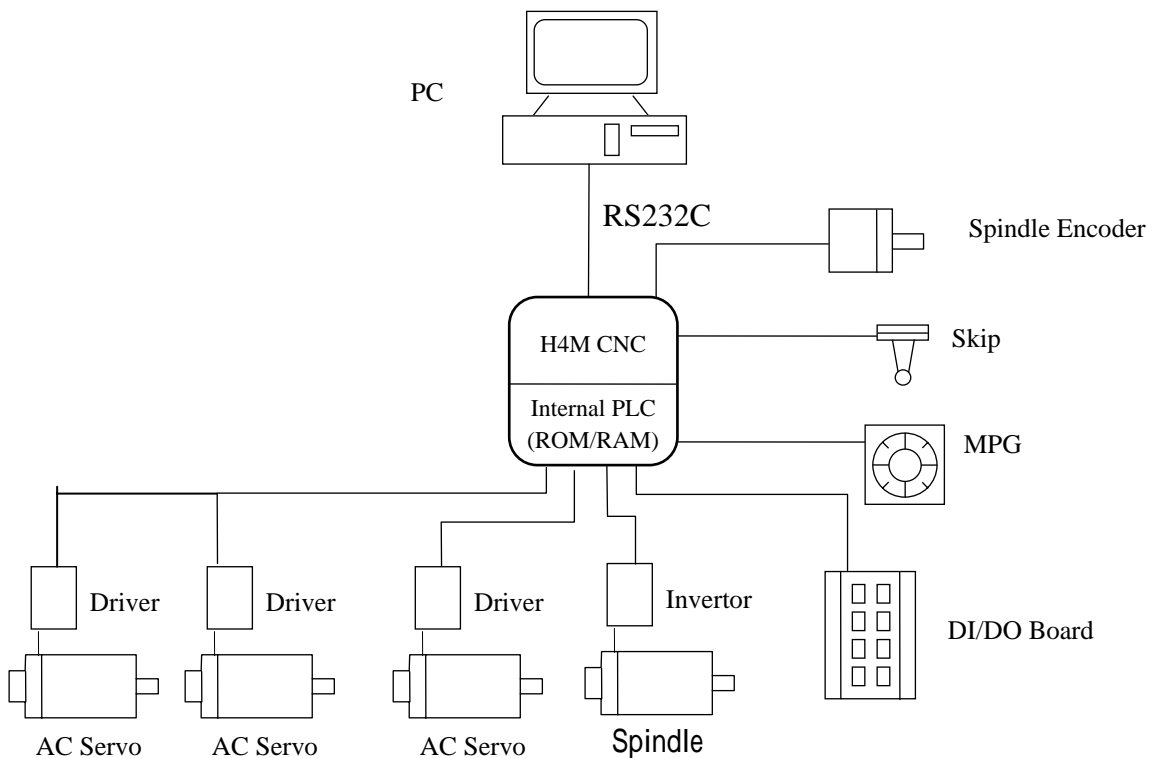


Fig 5-1 Applications of H4M Controller

5.2 System Installation

5.2.1 Ambient Surroundings

H4M series of CNC controllers must be installed and used in the following ambient surroundings. It can cause malfunctioning if the users utilize it outside any of these ambiances.

- Ambient temperature must be between -0 to 45 during the operation and -20 to 55 in the storage and trafficking.
- The maximum temperature change rate should be 1.1 per minute.
- Under the normal condition of humidity, the relative humidity should lower than 80% RH and higher than 95% RH in a short time.
- The vibrant limitation should be at the maximum length of 0.075 mm with the frequency of 5 hertz in operation.
- Noise: the maximum pulse should be $2000 \text{ V} / 0.1 \times 10^{-6}$ seconds per 0.01 second in operation.
- Consult us when installing the system in an environment with high degree of dust, coolant, or organic solution.

5.2.2 Cabinet Considerations

Pay attention to the following items when designing a cabinet.

- The cabinet must be constructed to house the CNC unit, MDI panel and is always a completely enclosed box.
- Temperature rise within the cabinet should be 10° C or less than the ambient temperature.
- Sealing of the cable inlet/outlet or doors must be done thoroughly.
- To avoid the noise interference, various units, the connecting cable and the AC power source must be mounted at least 100 mm apart. In a magnetic field environment, this distance should be increased to at least 300 mm.
- When mounting the servo amplifier, please refer to the Servo Motor Manual.

5.2.3 Thermal Design in the Cabinet

The cabinet for the control unit must be a closed structure, and the temperature rise within the cabinet must be 10°C or less than the ambient temperature. When designing a metal cabinet for a CNC unit, two factors must be considered, which are the heat source and the radiation area. For a CNC unit, the user cannot do too much to change the heat source. Therefore, the one thing that the user can do in controlling the temperature rise is the radiation area. The allowable temperature rise inside a metal cabinet can be estimated as follows:

1. With a cooling fan, the allowable temperature rise is $1^{\circ}\text{C}/6\text{ W}/1\text{ m}^2$.
2. Without a cooling fan, the allowable temperature rise is $1^{\circ}\text{C}/4\text{ W}/1\text{ m}^2$.

This means that a cabinet with a radiation area of 1 m^2 and a cooling fan will have an internal temperature rise of 1°C when a heating unit of 6 W (4 W without a cooling) is inside the cabinet. The radiation area of a cabinet is the entire surface area of the cabinet minus the area contacting the floor.

Example 1 (with a cooling fan):

A cabinet has a 2 m^2 radiation area with a temperature rise of 10°C . The maximum allowable heat value inside the cabinet is $6\text{ W} \times 2 \times 10 = 120\text{ W}$. Therefore, the heat generated inside the cabinet must be kept less than 120 W . If the heat generated is 120 W or more in the cabinet, some other cooling devices, such as cooling fins, must be incorporated in the unit.

Example 2 (without a cooling fan):

A cabinet has a 2 m^2 radiation area with a temperature rise of 10°C . The maximum allowable heat value inside the cabinet is $4\text{ W} \times 2 \times 10 = 80\text{ W}$. Therefore, the heat generated inside the cabinet must be kept less than 80 W . If the heat generated is 80 W or more in the cabinet, some other cooling devices, such as fan, or cooling fins, must be incorporated in the unit.

5.2.4 External Dimensions and Diagrams

- H4M Keyboard Panel and LCD Screen

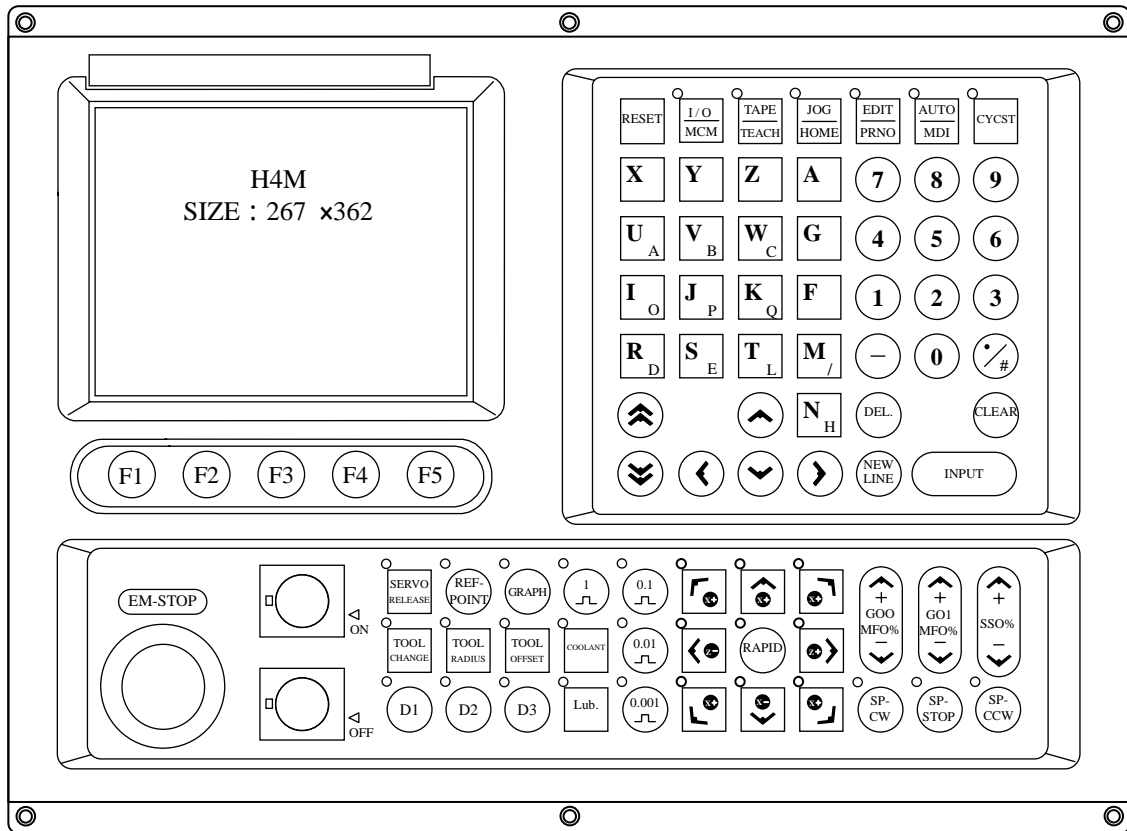


Fig 5-2 H4M Keyboard Panel and LCD Screen

5.2.5 CPU Main Board Connectors (Rear View)

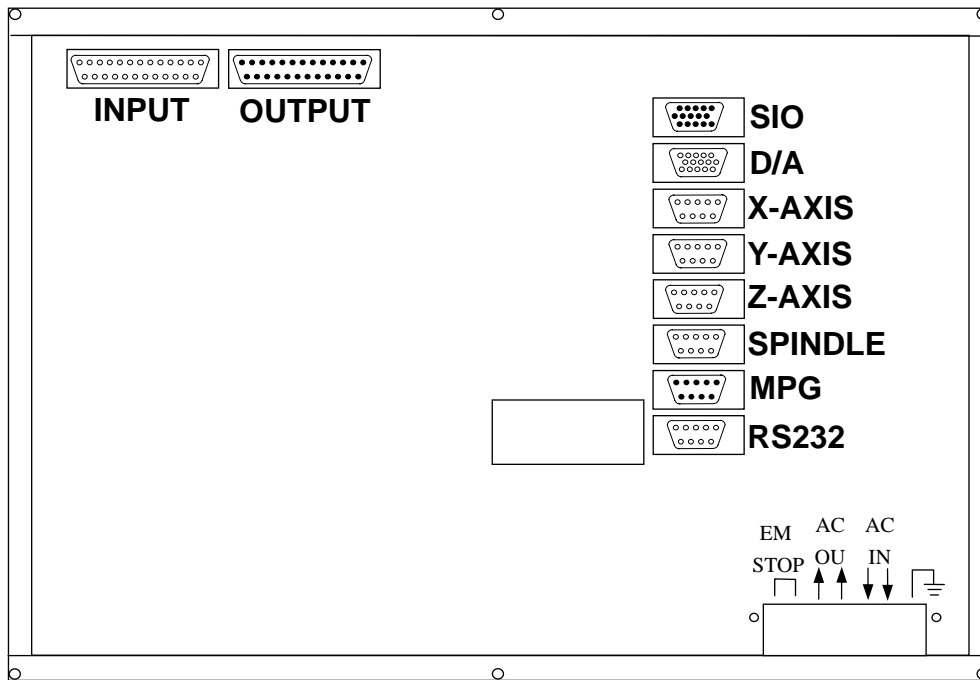


Fig 5-3 H4M CPU Main Board Connectors

5.2.6 Dimension for H4M Controller Box (Rearview)

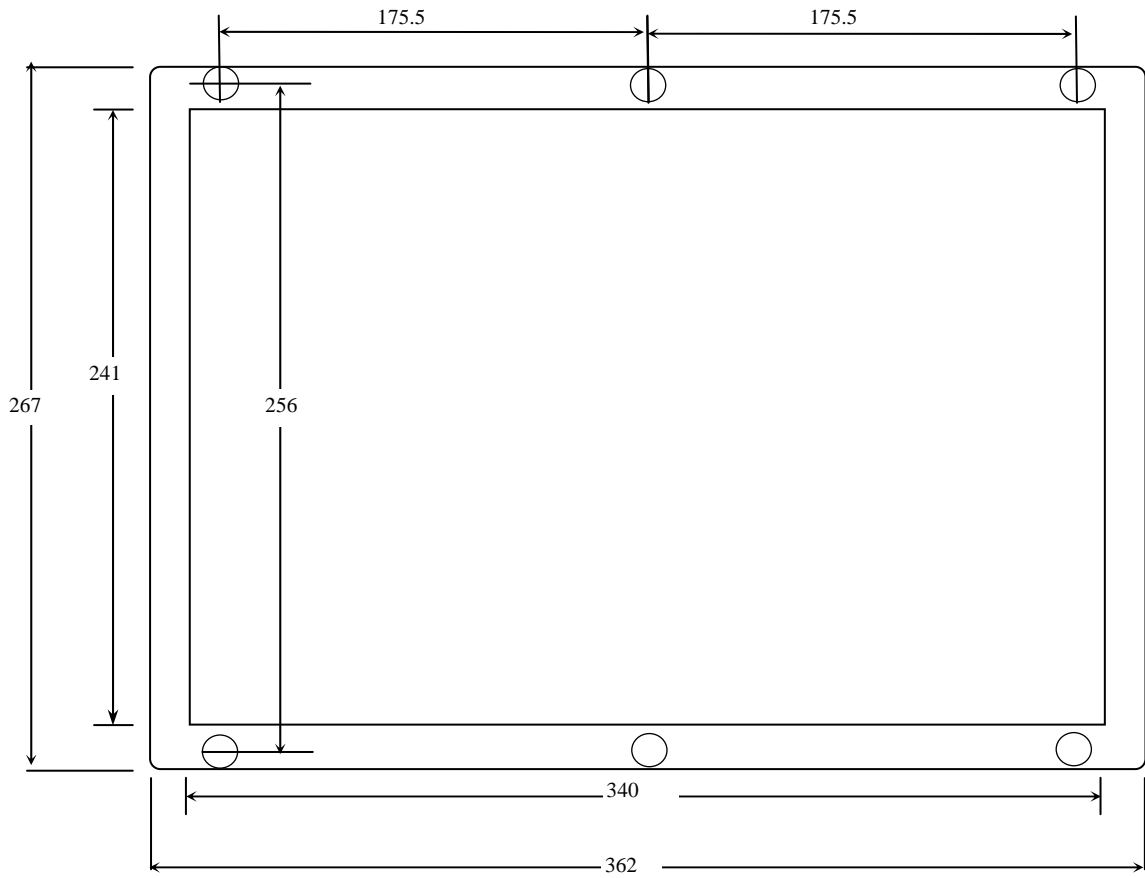


Fig 5-4 H4M Controller Box

5.2.7 Dimension for H4M Controller Box (Front View)

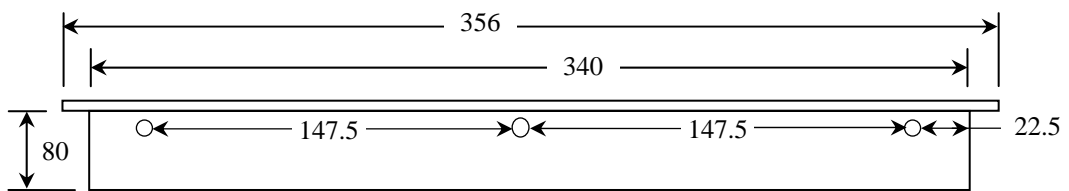


Fig 5-5 H4M Controller Box (Front View)

5.2.8 Cutout Dimension for H4M Controller Installation (Rear View)

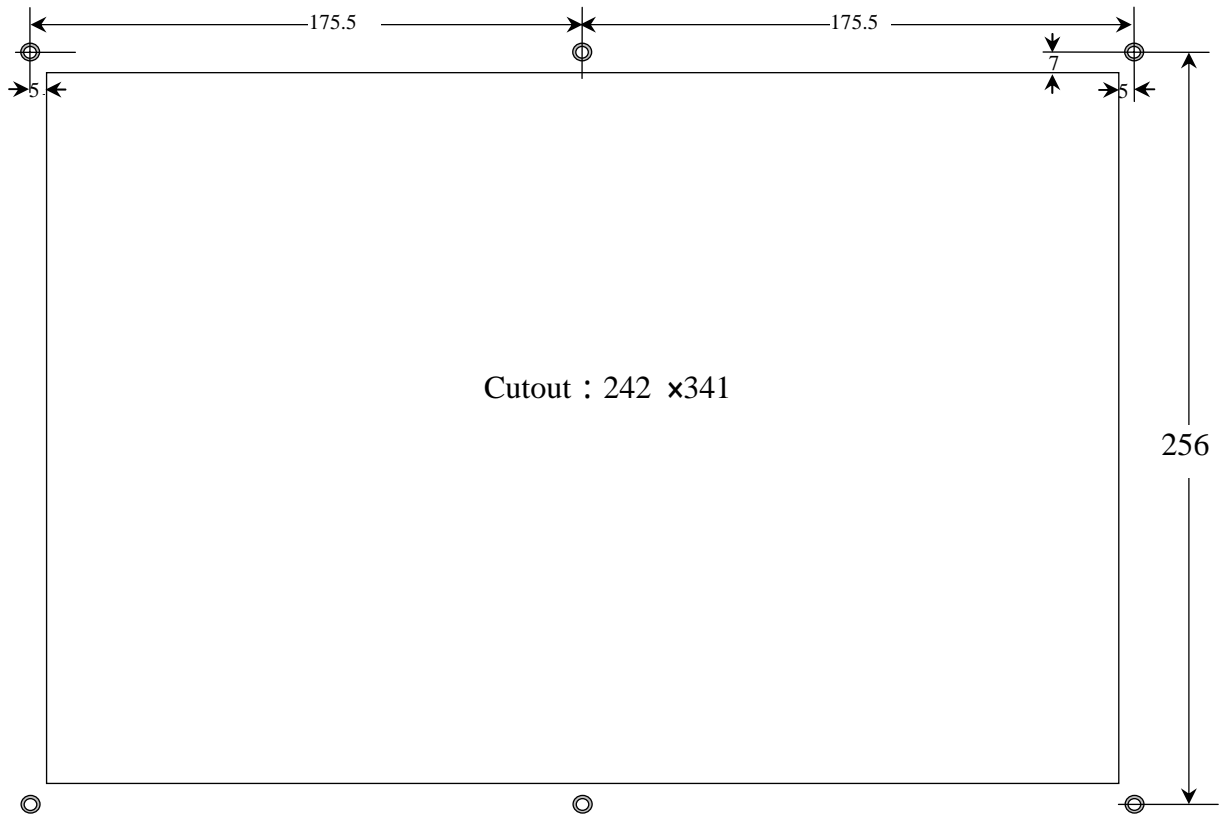


Fig 5-6 Cutout for H4M controller Installation

5.3 Input/Output Interface Connection (I/O)

5.3.1 Input Board / Output Board –Terminal Block Type

NPN Type- Standard Input Board

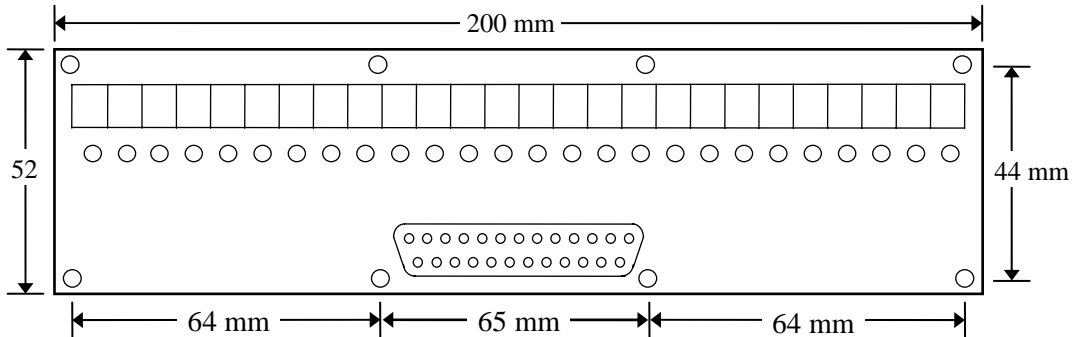


Fig 5-7 NPN type 1 Input Board.

This board provides 24 input points with LED indicators. When the input signal is correctly received, the corresponding LED will be ON. Otherwise, check your program or connections.

Note: When the power with 24 positive volts plugs in the points, it will be taken with 24 positive volts on each terminal in each group on the block.

NPN Type- Standard Output Board

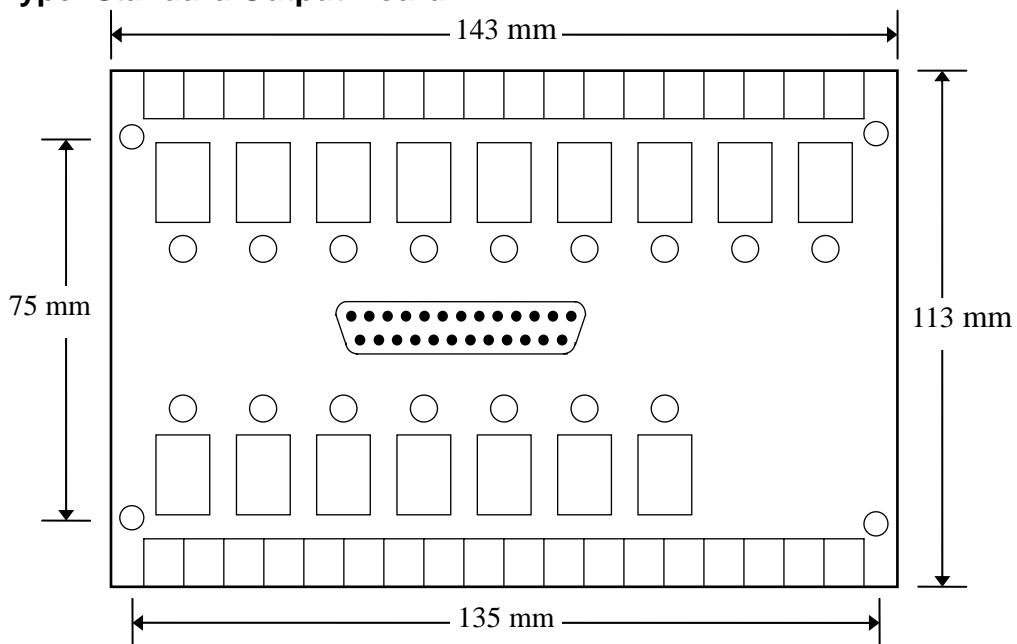


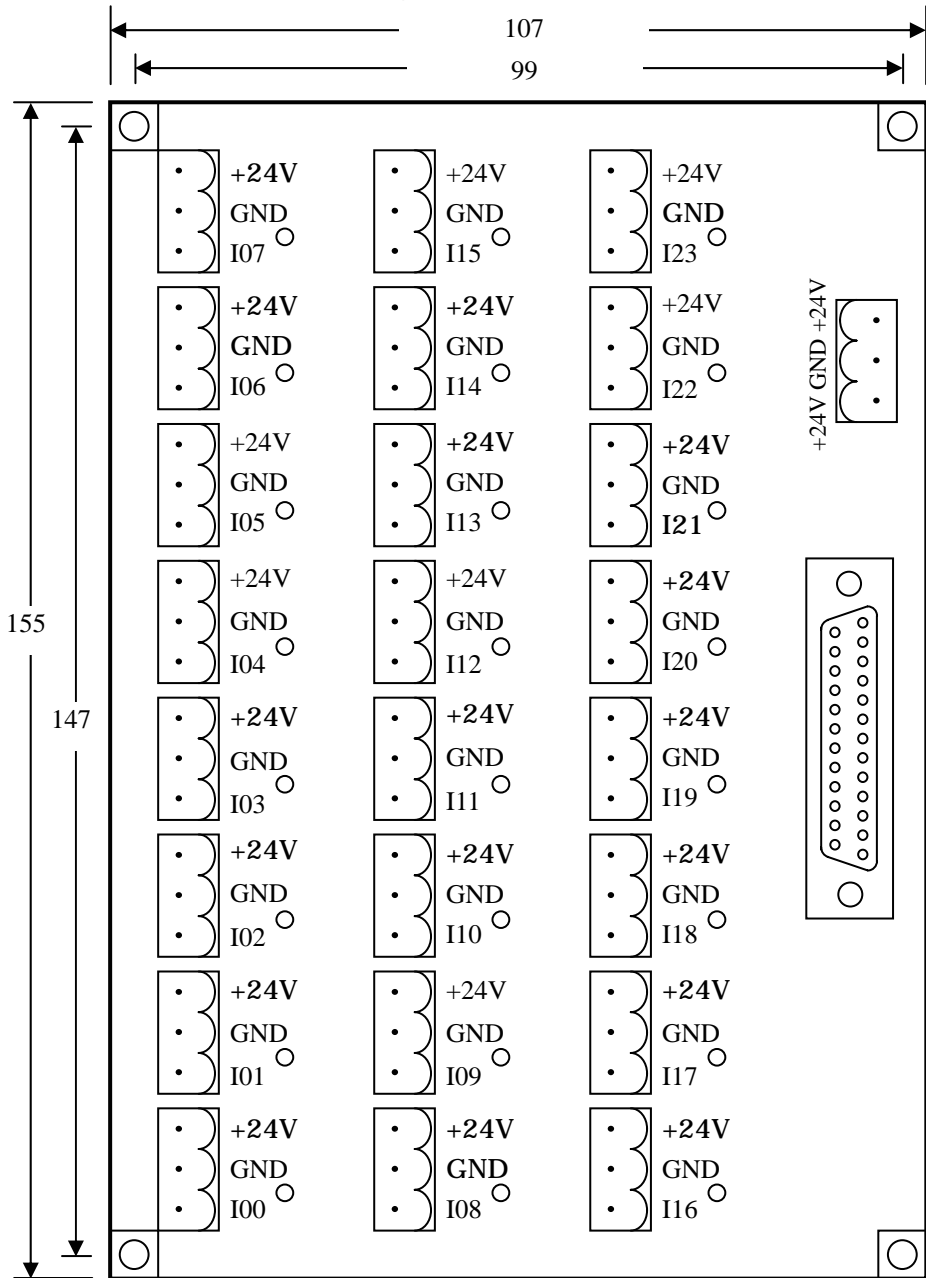
Fig 5-8 NPN type 1 Output Board

This board provides 16 output points with LED indicators. When the output signal is correctly received, the corresponding LED will be ON. Otherwise, check your program or connections.

5.3.2 Input Board/Output Board-CE Rules

Input Board

Standard NPN Input Board provides 24 input points with LED indicators. When the output signal is correctly received, the corresponding LED will be ON. Otherwise, check your program or connections.



Unit : mm

Fig.5-9 Input Board

Note: When the power with 24 positive volts plugs in the points, it will be taken with 24 positive volts on each terminal in each group on the block.

Output Board

Standard NPN Output Board provides 16 output points. When it outputs, the signal indicator will be lighted. If DIP Switch is off, common connectors and normal-open are the ordinary switch connectors. If DIP Switch is on, common connectors can be directly connected with 24V signal ground. Therefore, when the magnetic valve, DC or motor from the normal-open block is driven by 24V signal ground, only to put DIP into on position. It is not necessary to connect common connectors with 24V signal ground.

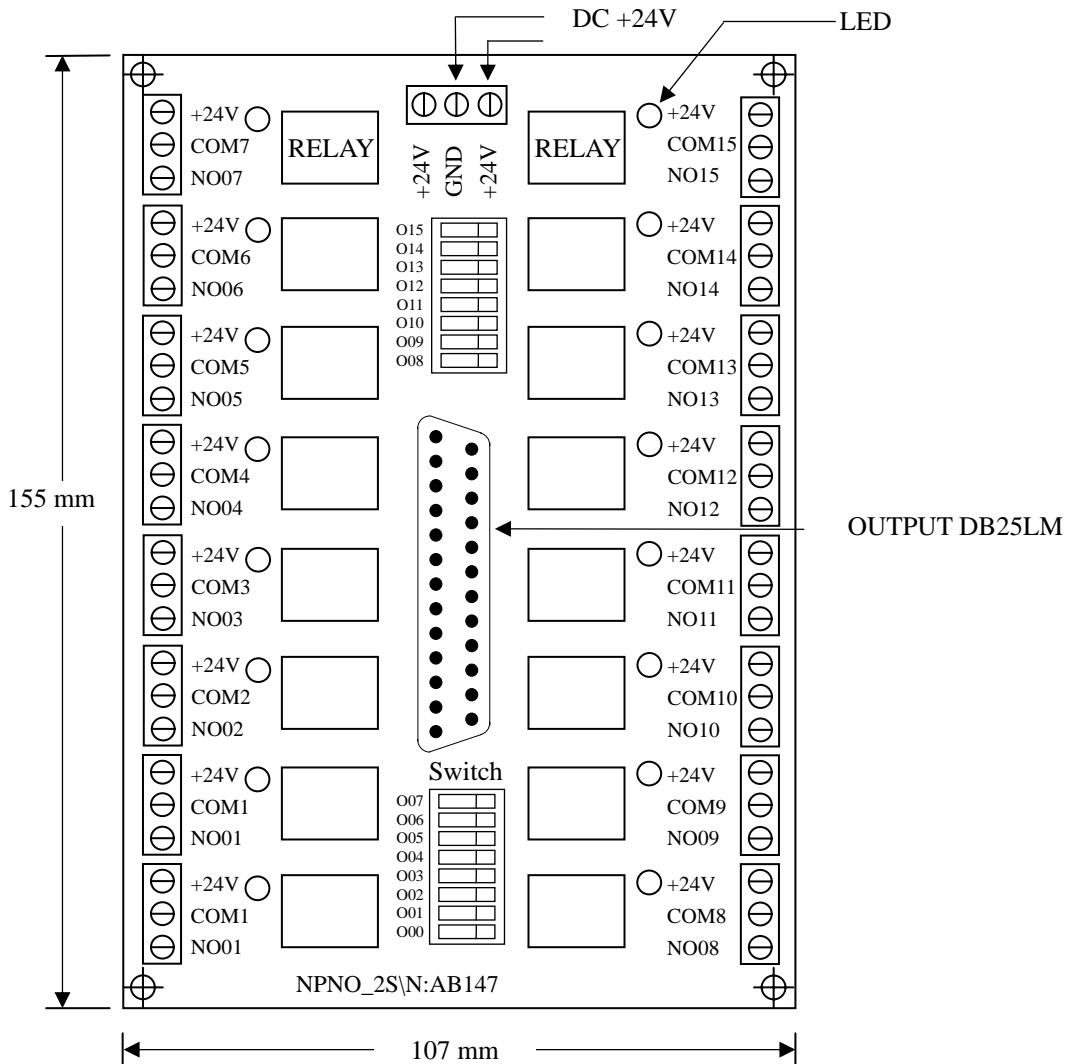


Fig.5-10 Output Board

5.3.3 I/O Connector Pin Assignment

The pin assignment of 25-pin connector is shown in Fig 5-1. This assignment is standard in H4M and can be applied to those connectors on main board, input board, or output relay board. The connector designation on H4M controller is shown in Table 5-1. Note that the input connector has 24 I-points while output connector has 16 O-points.

Table 5-1 I/O Connector Designation and Pin Assignment

Connector Designation	I/O Pin Assignment
P1	O 000 ~ O 015
P2	I000 ~ I023

P2 Input Pin Assignment (Female)

P1 Output Pin Assignment (Male)

H4M P2,Input Pin Assignment (Female)

H4M P1,Output Pin Assignment (Male)

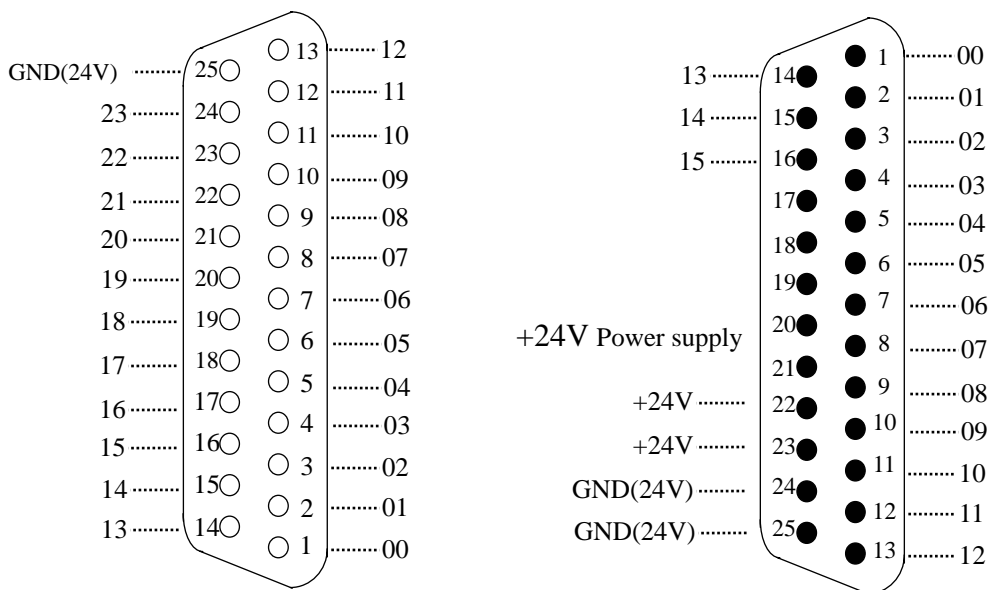


Fig 5-11 I/O Connector Pin Assignment (NPN-type)

I/O signals can be connected directly to P1 and P2 connectors on H3X/P-T controller. All output points on H4M main board are transistor circuit with open collector. However, it would be advisable to use input board and output relay board before the signal going into the controller. This provides additional protection for main board on CNC from being damaged by electrical surge.

5.3.4 Input Signals

Input signal is the one from external devices to CNC controller, such as push button, sensor, limit switch, output relay board or proximity switch, etc. There are 24 inputs

for H4M controller and their status can be displayed on LCD screen by pressing the blank key (below AUTO key) on the keyboard.

- Input Signal Specification : The voltage to drive the input signal is 24V, but the signal line is connected to the “sink” end (0V). The current is 8 mA.
- Input Signal Directly Connected to CNC (NPN-type)

Input signal bypasses the input board and is directly connected to P2.

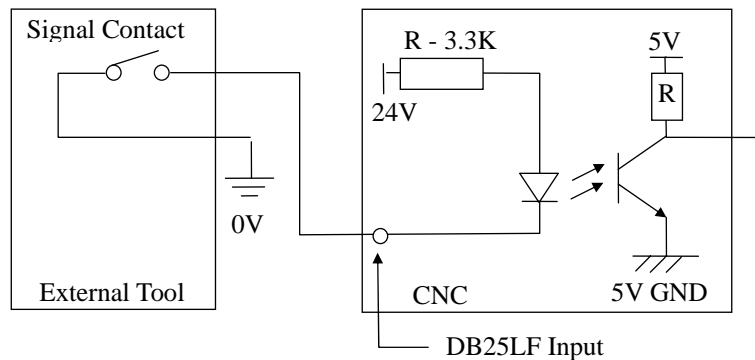


Fig 5-12 Input Signal Directly Connected to CNC (NPN-type)

- Input Signal Connected to Input Board then to CNC (NPN-type)

If input point is in use, the connector distribution is shown at the figure 5-11.

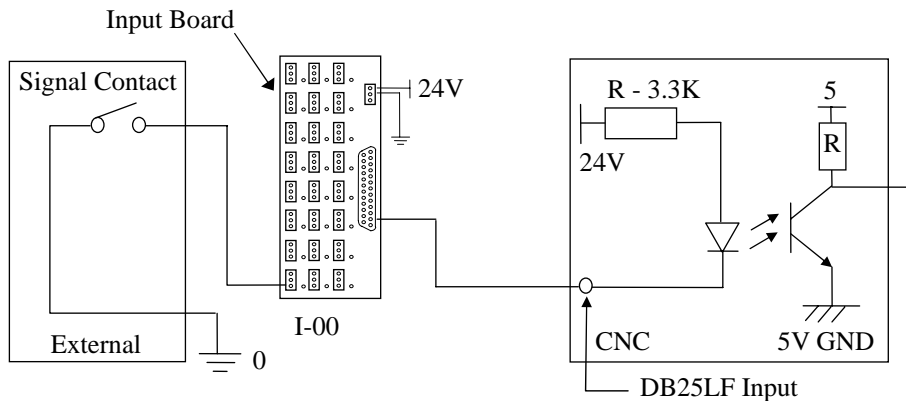


Fig 5-13 Input Signal Connected to Input Board (NPN-type)

5.3.5 Output Signals

Output signal is the one from CNC controller to an external device. The output circuit of H4M controller is a transistor circuit that can be used to drive a relay or LED, etc.

- Output Signal Specification:
 - a) With I/O relay board, each connection point on the board can withstand a maximum of AC 250V and 10A current.

- b) Without I/O relay board, the output transistor circuit of H4M controller is rated at a maximum of 35V and 500 mA of current.
- Output Signal Directly Connected from CNC to Tool (NPN-type)

If the output relay board is not used, the output signal can directly connect to the output points in CNC. P1 is the connector.

When output connector is directly connected to an inductive load, such as relay, it must connect a spark killer diode in parallel and as close to the load as possible. (See Fig 5-14)

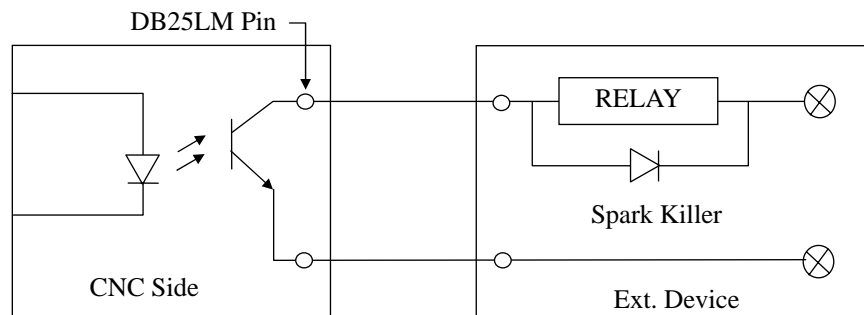


Fig 5-14 Output Signal from CNC without Relay Board (NPN-type)

- Output Signal Connected to Output Relay Board (NPN-type)

If output point is in use, the connector distribution is shown at the figure 5-11.

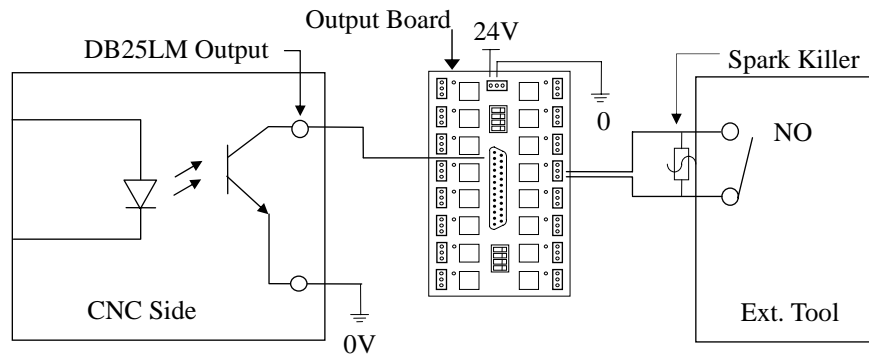


Fig 5-15 Output Signal Connected to Relay Board (NPN-type)

5.4 Connecting Diagrams

5.4.1 Connector Type

The types of connectors provided on the H4M controller box are summarized below. Each connector symbol is followed by a letter of either M (for male) or F (for female).

- DB9LF (or M) : D sub type connector 9 pins female (or male).
 DB25LF (or M) : D sub type connector 25 pins female (or male).
 TBxx : Terminal block with pin number indicated by "xx"

5.4.2 Connector Designation

The connector designation on H4M controller is printed on the back of the controller and their corresponding types are listed below.

Connector Name	Connector Designation	Type
Output Interface	P1	DB25LM (Male)
Input Interface	P2	DB25LF (Female)
D/A Spindle Analog Output	P3	DB9LM (M)
X-axis Servo	P4	DB9LF (F)
Spindle	P5	DB9LF (F)
Z-axis Servo	P6	DB9LF (F)
MPG	P7	DB9LM (M)
I/O	P8	DB9LF(F)
RS232C Interface	P9	DB9LF (F)
Main Board DC Power	JP1	TB6

Table 5-2 H4M Connector Designation and Type

5.4.3 H4M Connection (Terminal Block)

- H4M Main Connecting Diagram

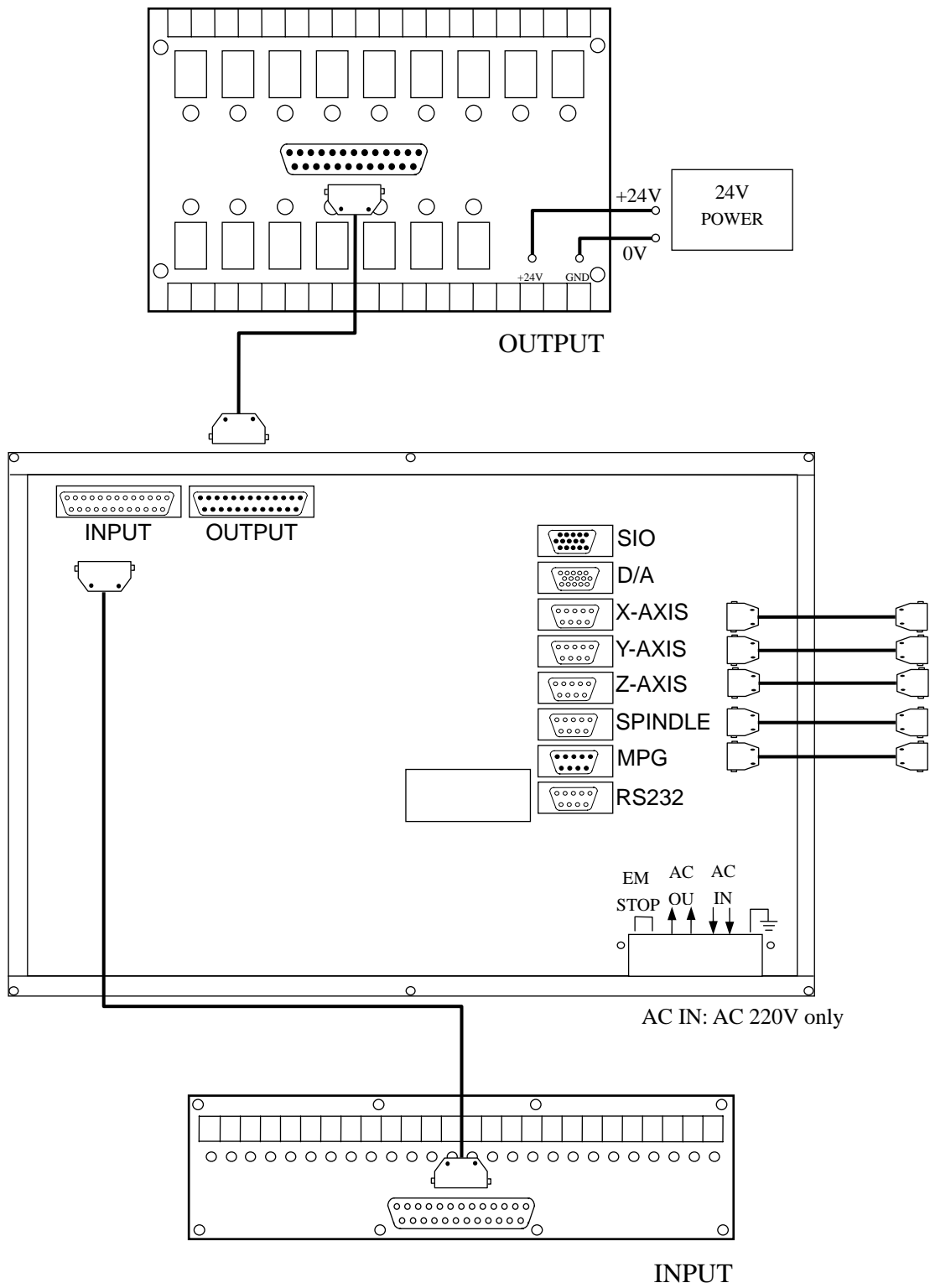


Fig.5-16

- Emergency-Stop Circuit-1

Fig 5-17 is a suggesting emergency stop circuit. This type of connection with both hardware and software attached to the controller, hardware can still turn off the servo if the software causes malfunction after press the button.

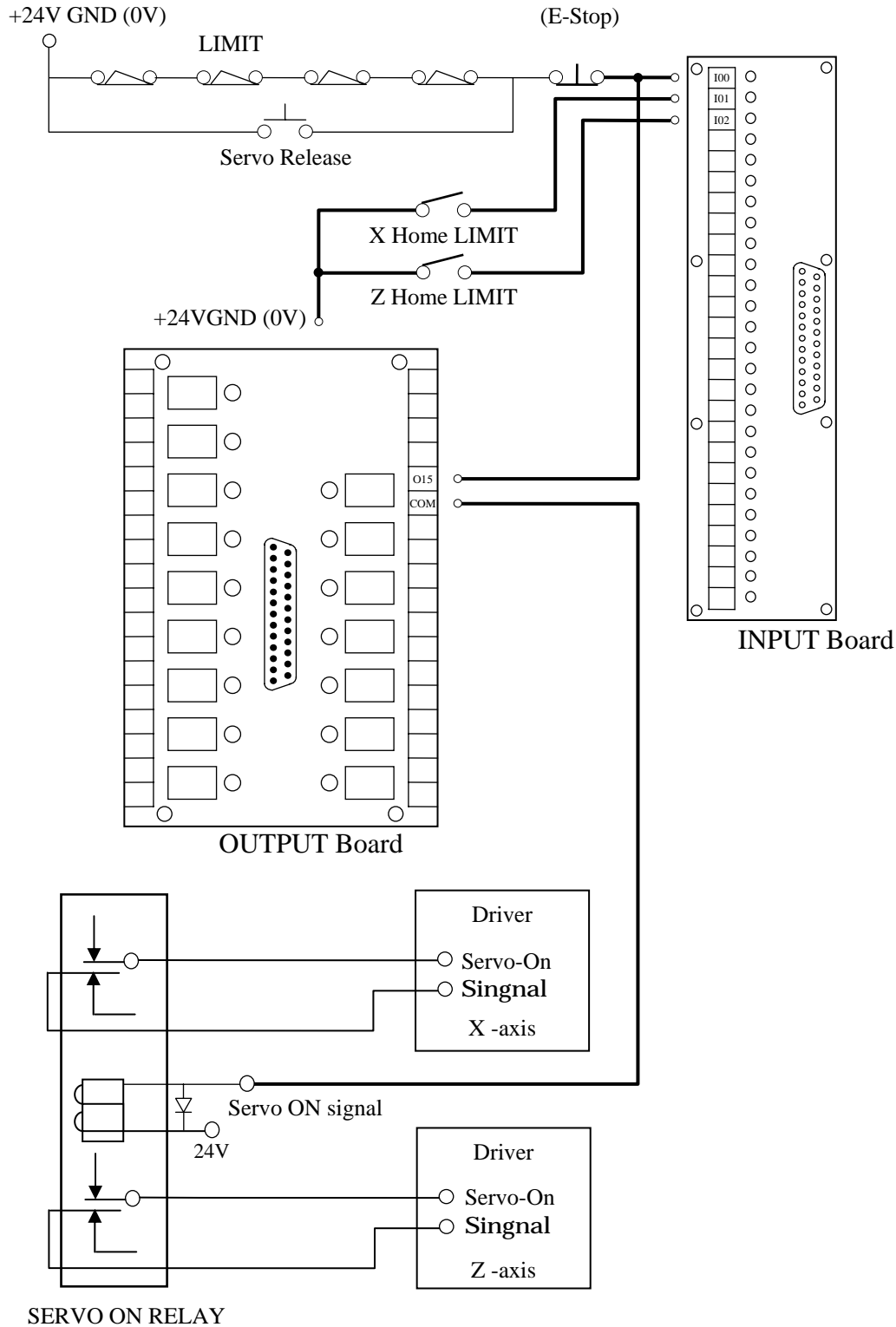


Fig 5-17

- Emergency-Stop Circuit-2

Fig.5-18 is a simplified connection diagram.

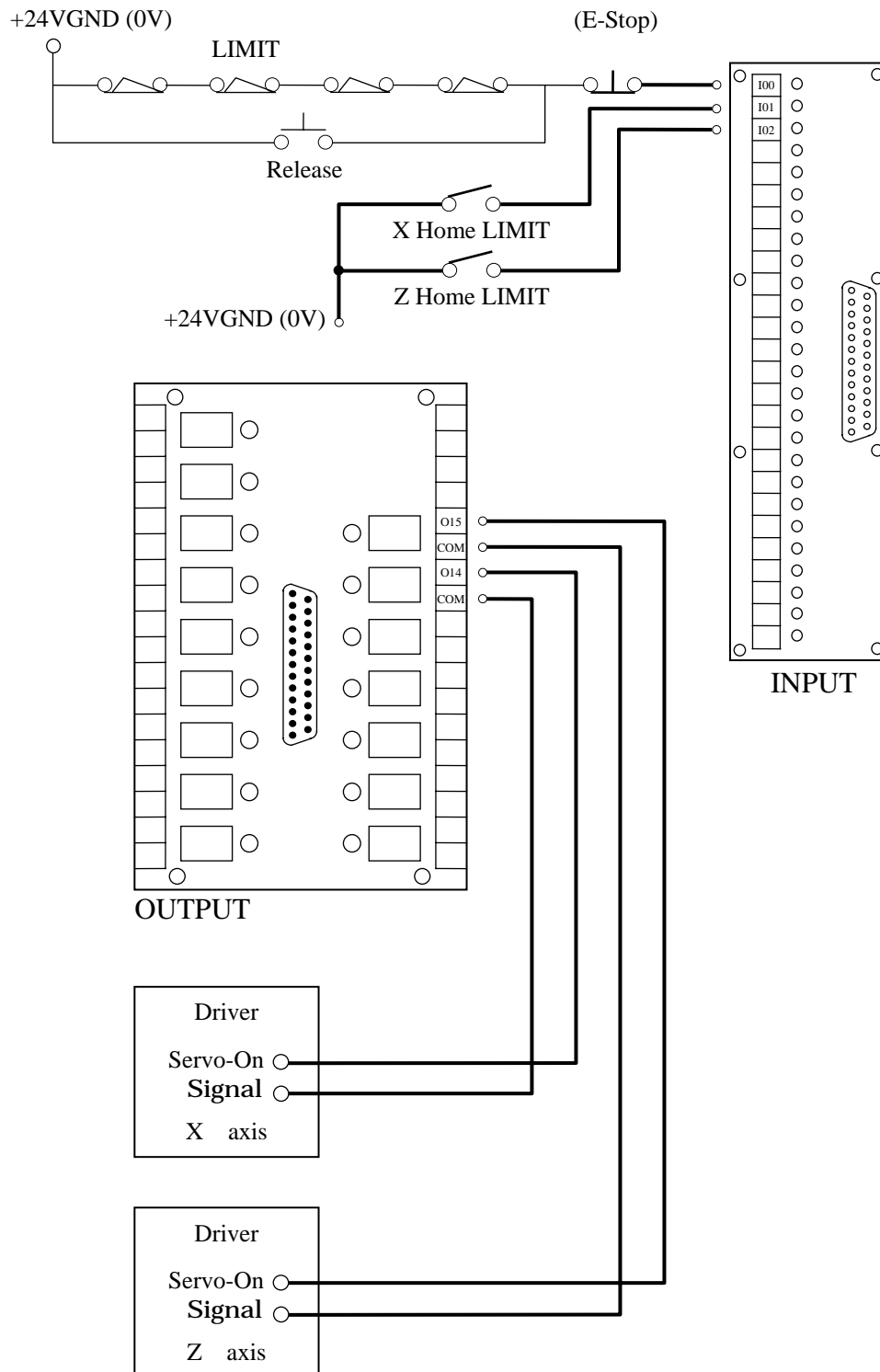


Fig 5-18

● H4M Spindle Connection

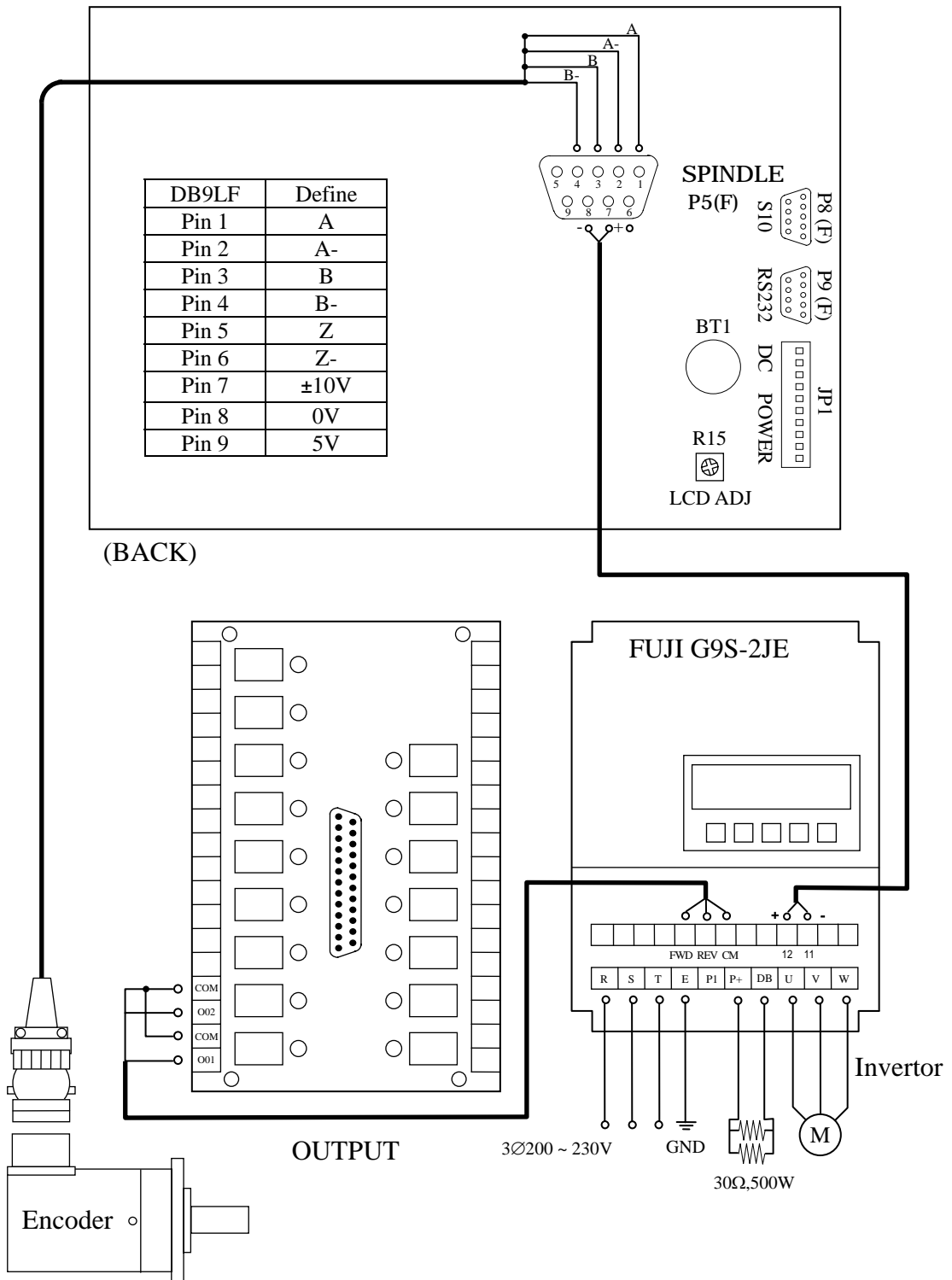


Fig. 5-19

Note: If the users intend to operate the parts with 110V or 220V(Oil, water pump, motor), we suggest connecting 24v only on the output board and from output board to generate the other with another relay addition.

5.4.4 H4M Connections (CE Rules)

H4-M Main Connection Diagram

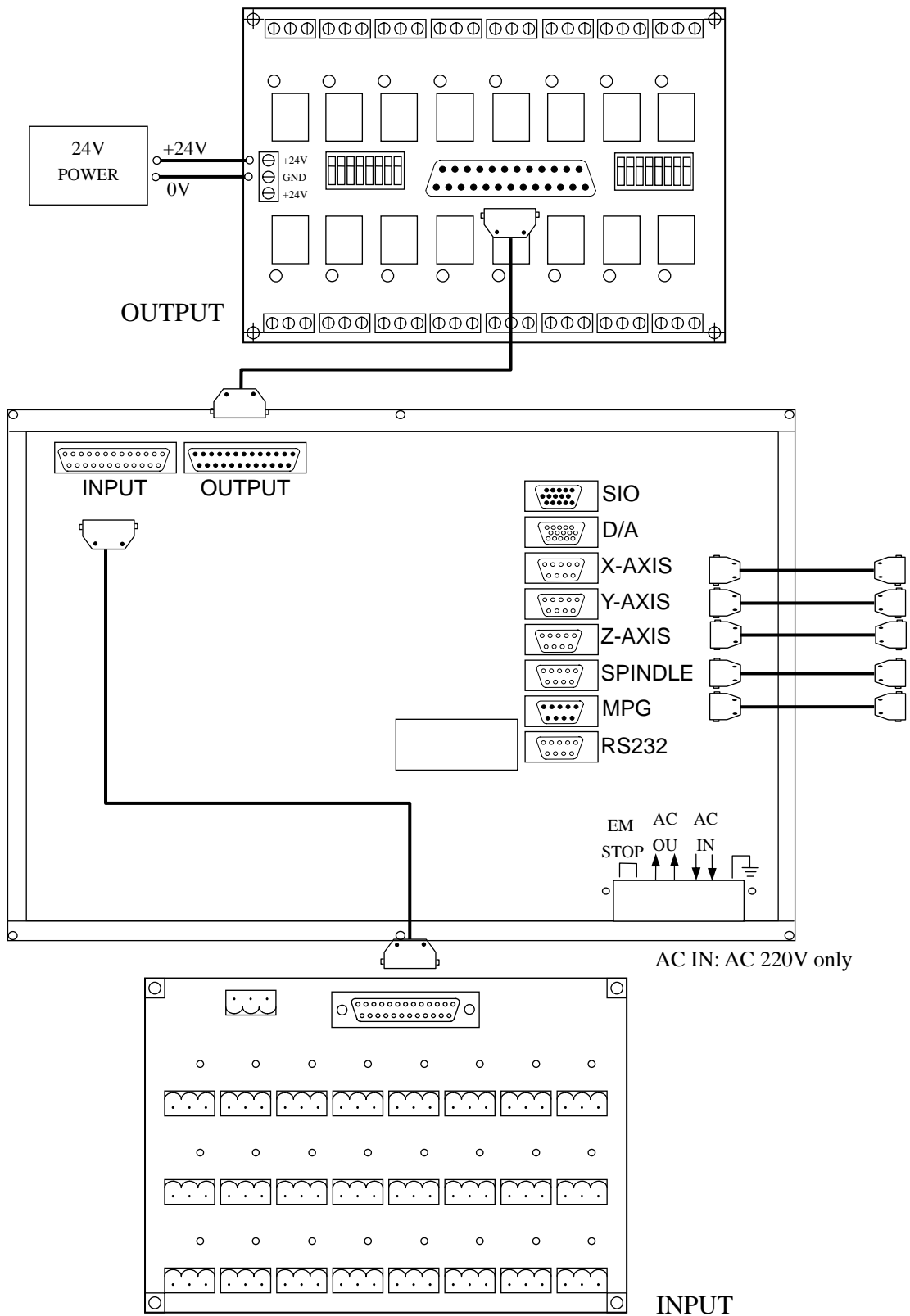


Fig. 5-20

● Emergency-Stop Circuit-1

Fig 5-21 is a suggesting emergency stop circuit. This type of connection with both hardware and software attached to the controller, hardware can still turn off the servo if the software causes malfunction after press the button.

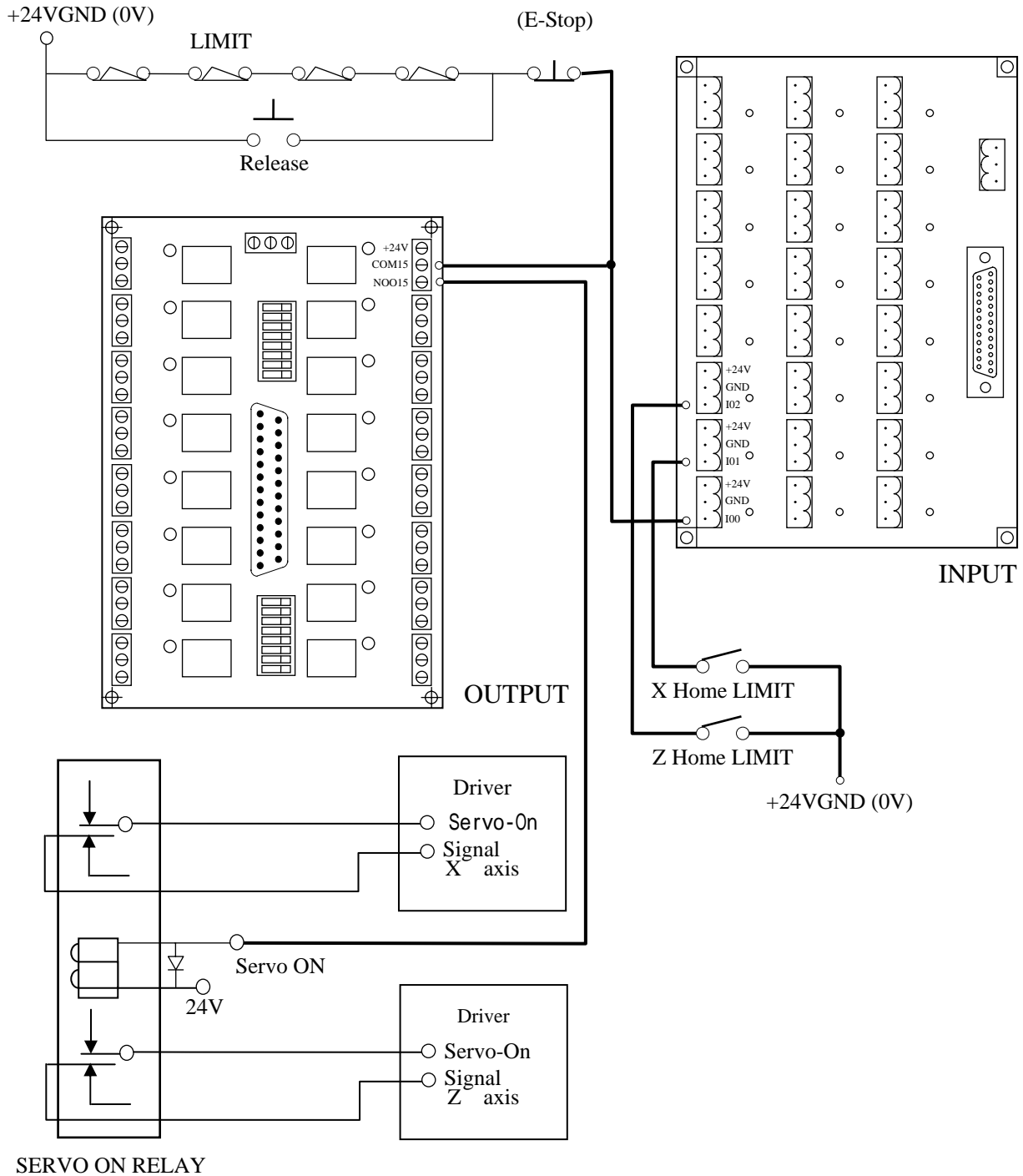


Fig.5-21Emergency-Stop Circuit

● Emergency-Stop Circuit-2

Fig.5-22 is a simplified connection diagram.

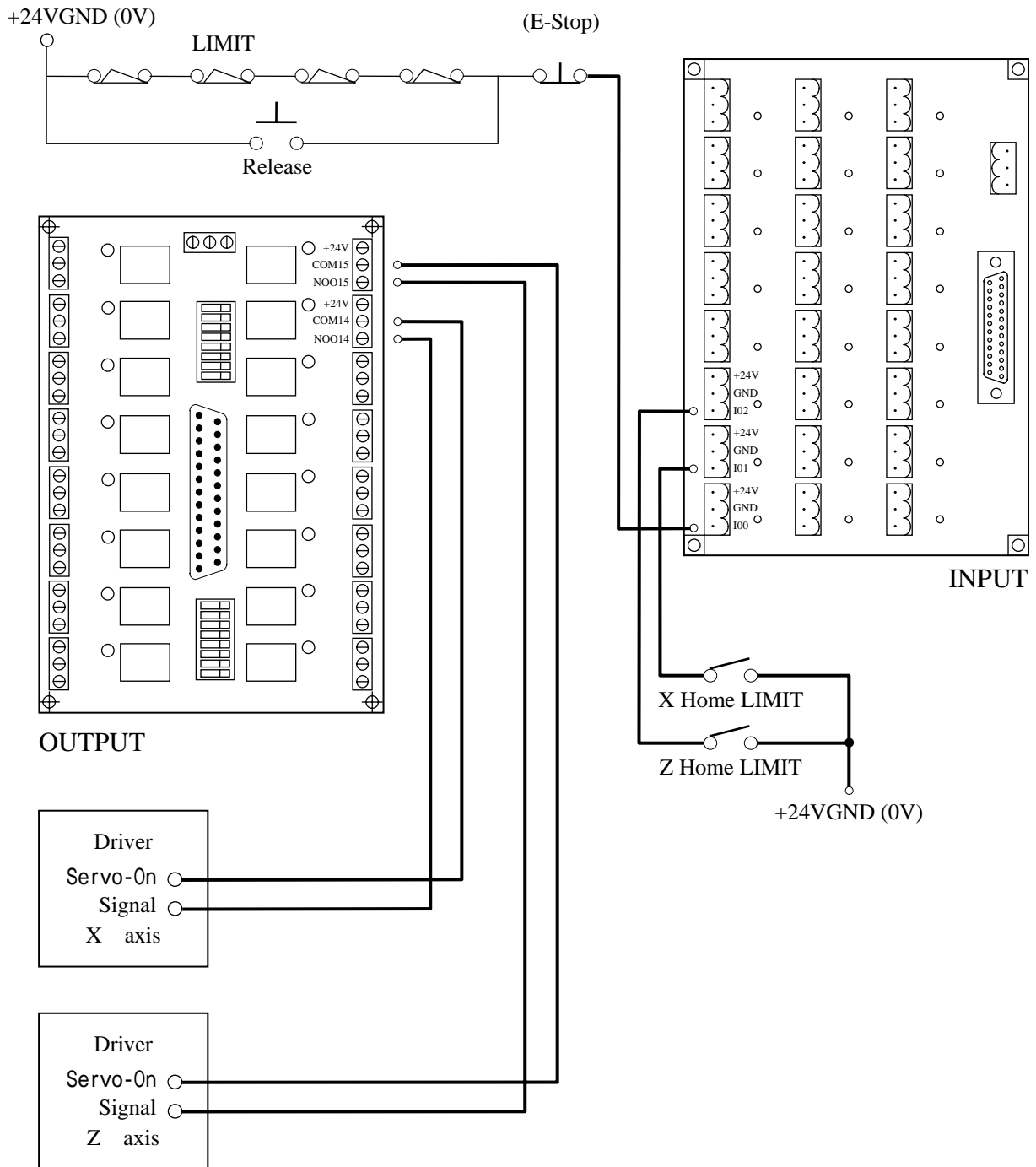


Fig.5-22

- Servo Motor Connection (This is the example of **MITSUBISHI** J2S Motor)

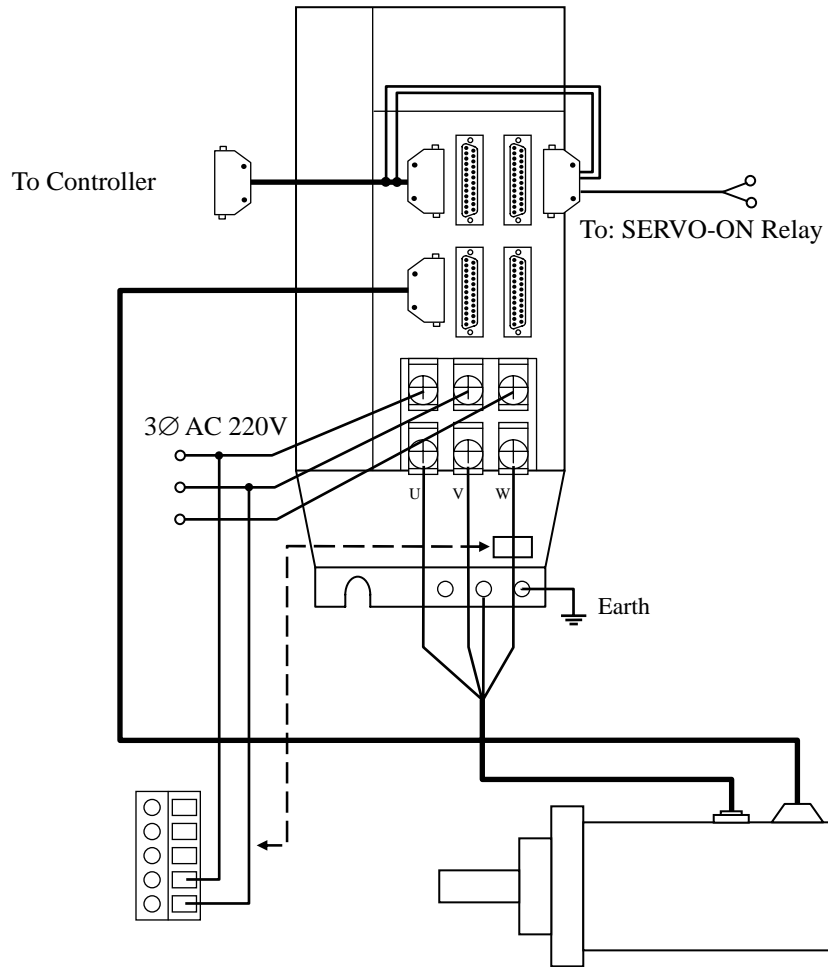
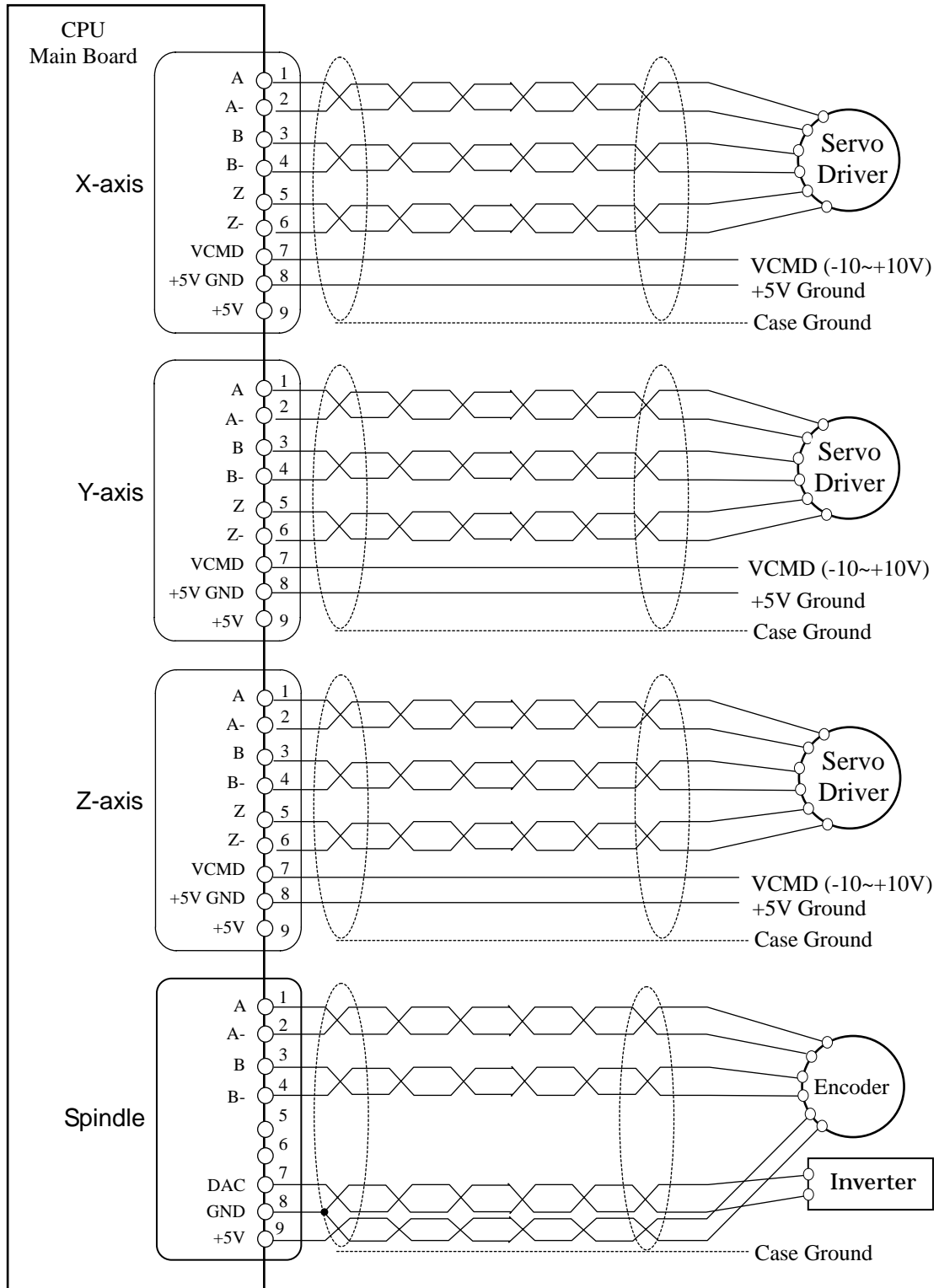


Fig.5-23

5.4.5 Motion Control (Servo/MPG) Connection

Suitable Type for H4M

The connections may vary depending on the type of servo system and the MPG that customer uses.



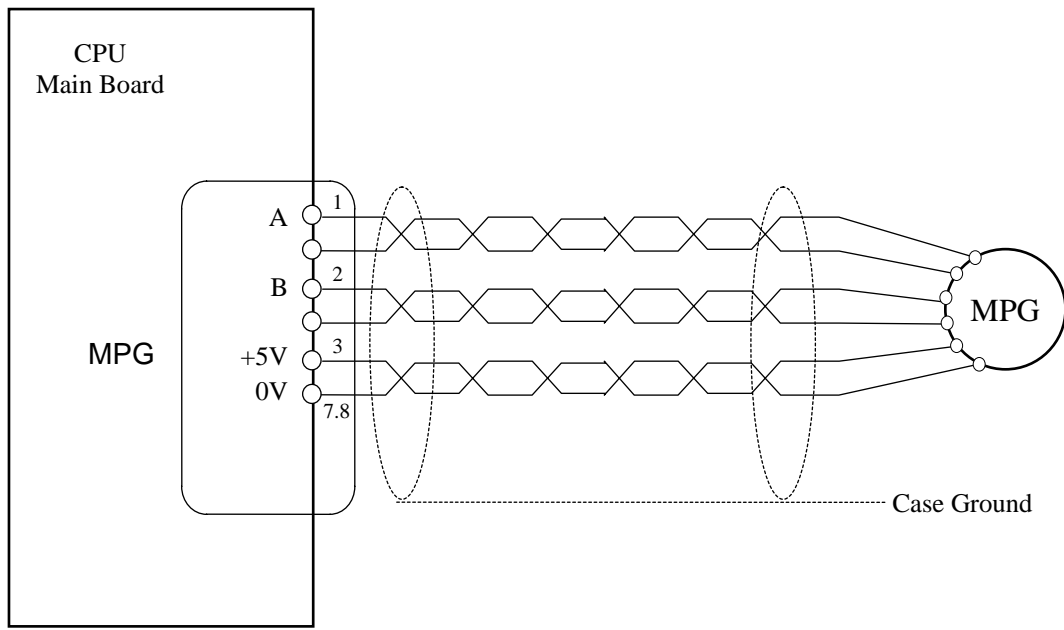


Fig 5-24 Servo and MPG Unit Connection

5.4.6 AC Power System Connection

HUST internal PLC in the controller has a timer to control the servo-on delay. When the set time (about 1.5 seconds) in this timer has elapsed, an output is ON to turn on the servo motors.

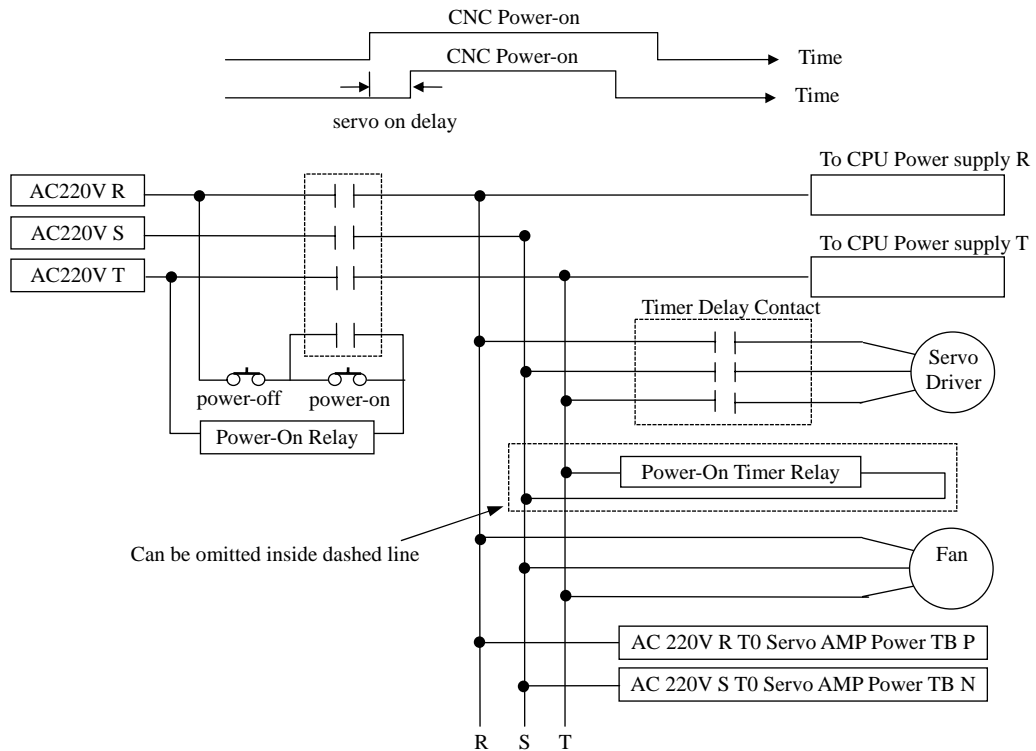


Fig 5-25 AC Power Connection to Servo Driver

5.4.7 MPG Hand-wheel Connecting Diagram

If the tool movement is in the opposite direction to the MPG hand-wheel rotation, please exchange A and B signal connections.

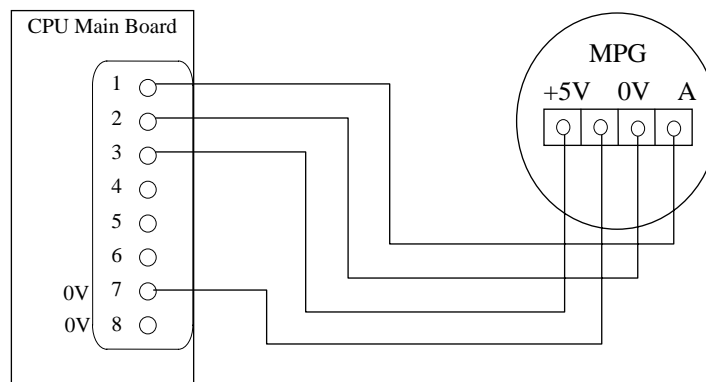


Fig 5-26 MPG Hand-wheel connection

5.4.8 RS232 Connector Pin Assignment and Connecting Diagram

For the operational details of RS232 communication, please refer to Chapter 9 of H4M Operation Manual.

1. Do not exceed 15 meters for the cable connecting RS232 port and PC.
2. The interface voltage on the PC side should be in the range 10 ~ 15V.
3. Try NOT to work in an environment where electrical noises are common. EDM machine and electric welder are noise generators. Do not use the same power source as EDM machine or electric welder.

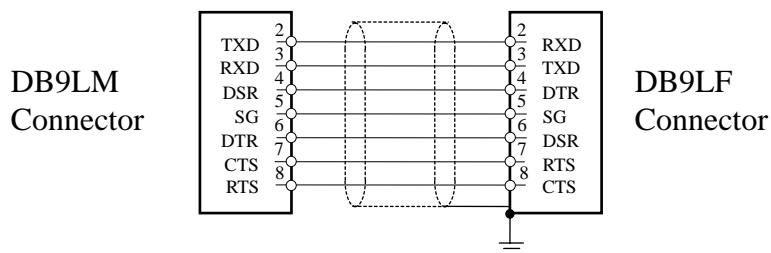
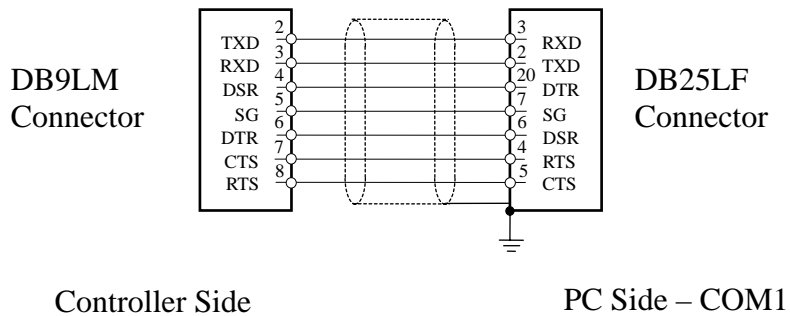
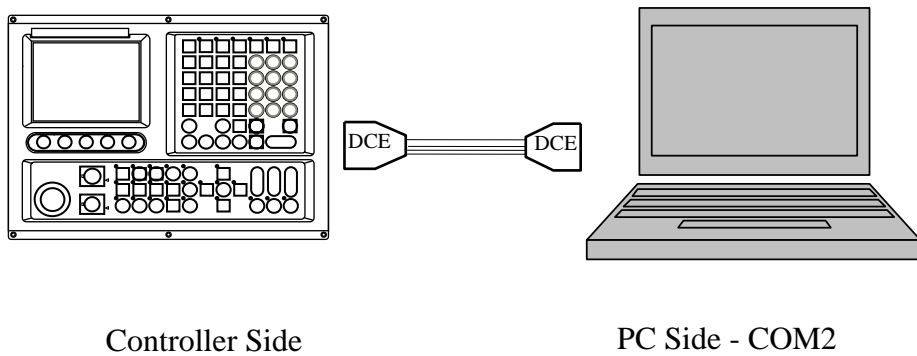


Fig 5-27 RS232 Connector Pin Assignment

6 ERROR MESSAGES

When an error occurs during operation, H4M controller will be stopped and an error message displayed at the top of the screen as shown in Fig 6-1. If there is more than one error, only one error will be displayed at a time. The second error will be displayed when the first problem is resolved. This chapter is to explain the error messages and the method to solve them.

PGM-NO: 000		CNT: 0000000	
X	-0000.000	MPG	: 000
Y	-0000.000	G00 MFO	: 000%
Z	-0000.000	G01 MFO	: 000%
		SSO MFO	: 000%
Spindle	: 0000000	M	: 00000
Feed Rate	: 0000000	T	: 00000
OFFSET-Z	: -0000.000	S	: 00000
Time: 0000000 sec			
Motor not home - READY - STOP			

Fig 6-1 Error Message Display

ERROR 1 -- MCM Data Error Or Battery Fail

Message:

MCM parameter setting is incorrect or the backup battery has failed.

Recommended Remedy:

1. Check if MCM parameter setting data are correct. Otherwise, use "G10 P1000" to clear all MCM data and re-enter all MCM data. Press RESET.
2. If the problem is on the battery, turn on the controller and leave it "on" for 4 hours to recharge the backup battery. If the error persists, please contact the maintenance engineer to change out the battery. If the controller has not been turned on for a month, the data in the memory will be lost.

ERROR 2 -- Servo Alarm, Please Home Again

Message:

Servo position control system (servo feedback) error. Possible causes are:

1. The voltage command from the controller is too fast for the motor to response.

2. The controller does not receive any feedback signal from the servo motor.

Recommended Remedy:

1. Check if the feed-rate F in the part program is too fast.
2. Check if the resolution setting of MCM parameter is correct.
3. Check if the worktable being overloaded, or any obstruction in the motor.
Also check the servo system including the connections.

ERROR 4 – G60 MISSING REPEAT NO. (LA)

Message:

G60 rotation command didn't input the repeat time value. (L)
G60 unused the G54 command.

Recommended Remedy:

Check program and correct the G60 of (L) value.

ERROR 8 -- Exceed 64 Characters For One Block

Message:

Exceeds 64 characters in one program block.

Recommended Remedy:

Check the program block and make sure each single block of program is less than 64 characters.

ERROR 10 – RS232 Error

Message:

RS232 communication error.

Recommended Remedy:

1. Check the baud rate in MCM #90 if compatible with the one in PC.
2. Check the communication cable connection from PC to CNC controller.

ERROR 11 -- Program Memory Error

Message:

Error in program memory due possibly from lack of charge in battery or memory being overloaded.

Recommended Remedy:

1. Execute G10 P2001 in MDI mode to clear all programs.
2. Check battery for memory chip. If the controller has not been turned on for months, the data in the memory will be lost. The controller will show "BT1 or BT2" message. In this case, change the battery.

ERROR 13 -- Error G-Code Command

Message:

Error in G-code function that is not acceptable by HUST H4M system.

Recommended Remedy:

Check the program for the G-code that is not acceptable by HUST H4M controller.

ERROR 14 -- X-axis Over-travel

Message:

The cutting tool traveled beyond the hardware limit in the X-axis.

Recommended Remedy:

Use MPG hand-wheel (or by hand) to manually move the tool in the X-axis within the operating range (or inside the hardware limit switch).

ERROR 14 -- Y-axis Over-travel

Message:

The cutting tool traveled beyond the hardware limit in the Y-axis.

Recommended Remedy:

Use MPG hand-wheel (or by hand) to manually move the tool in the Y-axis within the operating range (or inside the hardware limit switch).

ERROR 14 -- Z-axis Over-travel

Message:

The cutting tool traveled beyond the hardware limit in the Z-axis.

Recommended Remedy:

Use MPG hand-wheel (or by hand) to manually move the tool in the Z-axis within the operating range (or inside the hardware limit switch).

ERROR 18 -- End Of File Not Found

Message:
Error in the program ending.

Recommended Remedy:
Check the ending statement of the program, such as M02 and M30.

ERROR 20 --Software Over-Travel

Message:
The cutting tool has traveled beyond the bounding limit as set by the software.

Recommended Remedy:
Check the program or revise the settings in MCM #93~#98 for software travel limit.

ERROR 22 -- EM-Stop, Home Again

Message: Controller is in emergency stop state.

Recommended Remedy:
Resolve the cause for emergency stop. Restore Emergency-STOP button and press RESET.

ERROR 24 -- M98 Exceed 8 Level

Message:
The sub-program calls exceed 8 levels.

Recommended Remedy:
Revise the part program and make sure the sub-program call does not exceed 8 levels.

ERROR 25 -- Wrong Circle Format or Data Error

Message:
The circular cutting command (G02, G03) or the command format in error.

Recommended Remedy:

Check the part program and recalculate the coordinate of the center of the arc/circle.

ERROR 30.1 –Battery is Low

Message:

The battery (BT1) for data storage is out of charge or service.

Recommended Remedy:

Replace the battery BT1.

ERROR 31 -- None PLC

Message:

There is no PLC ladder program in the memory.

Recommended Remedy:

Check if there is any PLC simulation program in FLASH-ROM and check if it is being properly installed. (EVN, ODD location)

ERROR 35 RS232C Program NO. Error

Message:

Program number error in the downloaded program.

Recommended Remedy:

Check the program number.

ERROR 36 Execution Mode Error

Message:

Execution mode selection in error.

Recommended Remedy:

Check the execution selected if correct.

ERROR 37 -- NC Alarm

Message:

There is a mechanical problem from the machine tool side.

Recommended Remedy:

Check the machine tool for proper operation. Correct the problem and press RESET.

ERROR 50~99

Message:

Error in user defined error message by G65 Macro.

Recommended Remedy:

Check if G65 function is properly applied. If not, have the problem corrected.

7 APPENDIX

*** Input Arrangement**

Input	Description	Note
I00	EM-STOP	NC
I01	X-axis Home Limit	
I02	Y-axis Home Limit	
I03	Z-axis Home Limit	
I04	CYCST	
I05		
I06		
I07		
I08		
I09		
I10		
I11		
I12	MPG×10	I14 must be NO, then it will be work
I13	MPG×100	I14 must be NO, then it will be work
I14	Use MPG hand-wheel	
I15	X-axis of MPG	I14 must be NO, then it will be work
I16	Y-axis of MPG	I14 must be NO, then it will be work
I17	Z-axis of MPG	I14 must be NO, then it will be work
I18	X - LIMIT (+)	NC
I19	X - LIMIT (-)	NC
I20	Y - LIMIT (+)	NC
I21	Y - LIMIT (-)	NC
I22	Z - LIMIT (+)	NC
I23	Z - LIMIT (-)	NC

PARAMATER:

MPG Mode = 0

I14 I17 will be Servo Ready signal

I14 X-SERVO READY (NC)

I15 Y-SERVO READY (NC)

I16 Z-SERVO READY (NC)

I17 SPINDLE SERVO READY (NC)

Parameter	X-AXIS	Y-AXIS	Z-AXIS
MPG-NUM	0000000	0000000	0000000
MPG-DEN	0000000	0000000	0000000
AD Mode	0		
G00 AD Time	0000	Accel / Decel Mode Set	
G01 AD Time	0000		
G99 AD Time	0000		
SP AD Time	0000		
SP Encoder	00...		
SP Factor	0	Spindle	
SP RPM at 10v	0000		
Counter Limit	00000	MPG Mode	0000000
Offset	Wear	MCM	

MPG Mode = 1 (HAND-WHEEL)

I14 I17 will choose the axis signal.

*** Output Arrangement**

Output	Description	Note
O00	Spindle CW	
O01	Spindle CCW	
O02	Coolant	
O03		
O04		
O05	Lubrication	
O06		
O07		
O08		
O09		
O10		
O11		
O12		
O13	Servo-on	
O14	Servo-on	
O15	Servo-on	

*** M-code Versus I/O**

M code	Description	I/O	Note
M03	Spindle CW	003=1	
M04	Spindle CCW	004=1	
M05	Spindle stop	003=0,004=0	
M08	Coolant On	002=1	
M09	Coolant Off	002=0	
M10	Chuck On	005=1	
M11	Chuck Off	005=0	

- Others

In PLC arrangement, the taken parts are the followings:

1. Variables: 1-1000.
2. Temporal Memory Saving Machine 20 or 90 after.
3. A_Bit 0 400, don't use it during the arrangement.

