HUST H2N

CNC CONTROLLER

OPERATION MANUAL

Dec , 2006

HUST Automation Inc.
No. 80 Industry Rd., Toufen, Miaoli, Taiwan, R.O.C.
Tel: 886 37 623242 Fax: 886 37 623241

CONTENTS

1 MAIN FEATURES OF HUST H2N CONTROLLER

- Controllable axis – 2 axes designated as X and Y.
- Voltage-driven servo system with maximum speed of response 500 KPPS (i.e. 30 meters/min with 1μ resolution).
- Capable of setting operation in CNC mode or Master/Slave mode.
- Through RS232C interface, H2N controller can be connected to a PC for display of controller status on PC screen.
- Through RS232C interface, H2N controller can be connected to a EASYVIEW-PLC-Operator-Panel for display controller status .
- Providing internal programmable PLC to suit your requirement.
- User can download/upload program from PC to the controller and/or execute the program in its entirety or step by step. User can edit his program using common software editor such as PE2.exe or HE5.exe.
- The CAD/CAM designed program can be downloaded to HUST H2N controller or the program can be executed from PC through RS232 interface.
- Keyboard can be customized to suit user's application or used as thumb switch.
- Designed the LCD display by yourself .
- H2N controller Standard I/O=24/16 , the 24 sets Input include 4 groups of inputs at high speed, with a maximum response of 50μ sec.
- Providing 1 set of ANALOG TO DIGITAL signal option .
- H2N controller has the capability of fly-cut.
- H2N controller can achieve accurate feed-length by comparing the feedback from
- The passive encoder and the roller.
- 128 K of program memory.
- Battery backup in case of power-off.
- MCM (machine constants) parameter table let user customize his specific machining requirements.
- Backlash error compensation for worn ball-screw.
- Providing 6 sets of user defined work coordinate to simplify program design.
- Providing data storage for 8 sets of tool offset compensation.
- User can customize Macro function G65.
- Feed-rate control either by mm/minute or mm/revolution.
- Continuous program execution or single block at a time.

- Option skip, option stop, and feed-hold function.
- Interchangeable absolute or incremental coordinate in programming.
- Self-diagnostic and error function.
- Circular programming by radius "R" as well as I and J values.
- MPG hand-wheel interface for program testing. (MPG test)

This operation manual includes programming basics, G-codes, keyboard operation, program editing, MCM parameters, manual operation and special functions, RS232 interface, and error functions. Table 1-1 shows the G-code summary available from HUST H2N controller.

Table 1-1 HUST H2N G-code Command

G-code	Function	G-code	Function
* G00 #	Rapid Positioning	* G43	Tool Offset Compensation ON
* G01 #	Linear Cut	* G49 #	Offset Compensation OFF
* G02	Circular Cut (CW)		
* G03	Circular Cut (CCW)	* G54 #	1st Work Coordinate
		* G55	2nd Work Coordinate
G04	Pause	* G56	3rd Work Coordinate
G10	Data Input	* G57	4th Work Coordinate
		* G58	5th Work Coordinate
G11	Simple output control	* G59	6th Work Coordinate
G12	Simple input control		
G14	Simple I/O control	G65	MACRO command
G28	Go to the 1st Ref Point	* G98 #	Feed-rate with mm/min
G29	Return to last location from ref. point	* G99	Feed-rate with mm/revolution
G30	Go to the 2nd Ref Point		
G31	Skip Function		

* Modal G-code.
# Power-on default G-code (Either G00 or G01 can be chosen as power-on default through proper MCM parameter setting).

2 PROGRAMMING BASICS

2.1 A Part Program

Prior to cutting a mechanical part by using a CNC machine, the shape and the dimension of the part must be drawn and accurately calculated. A computer program called a part program is then created to describe the shape of the part using a specific 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 designing a machining plan, the following factors must be considered:

- Determine the machining requirement and select the suitable CNC machine tool.
- Determine the work-piece loading method and select the appropriate cutting tool and the tool holder.
- Determine the machining sequence and the tool path.
- Determine the cutting conditions such as spindle speed (S), feed-rate (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 RS232 interface or directly on the CNC controller using the editing keys. HUST H2N series can do both. They will be discussed later.

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 below:

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

   - Drawing of the mechanical part.
   - Part shape description including coordinate calculations.
   - CNC part program design including spindle speed, feed rate, M-code, etc..
   - Editing the program instructions on CNC controller or PC.
   - Testing the program.

   The coordinate calculation is easy if the shape of the part is composed of lines or 90 degree angles. For curve cutting or slant lines, 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 complicate part, is time consuming and prone to making errors. In this case, automatic programming becomes more advantageous than the manual method.

2. Automatic Programming
   Automatic programming is a process in which the program design including coordinate calculation 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 of the machine tool into a part program, which is then transferred into CNC controller through RS232 interface. This process is called CAD/CAM. It is a necessary tool when designing a part program for a 3-D work-piece.

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 a specified speed, or to instruct the peripheral equipment to do some mechanical work. The cutting action is done when the cutting tool moves.

An example of a complete part program containing nine (9) blocks is as follows:

```
N10 G00 X40.000 Y10.000
N20 G00 X30.000 Y5.000
N30 M3
N40 G1 X10.000 Y20.000 F20.00
N50 V-5.0
N60 X30.000 Y-10.000
N70 G0 X40.000 Y10.000
N80 M5
N90 M2
```

A block of program can have one to several instructions and it has a general form as follow.

N____G____X(U)____Y(V)____F____M____

The block number "Nxxx" can be omitted. If you do not key in the block number, HUST controller can automatically generate the number for you by proper MCM parameter settings (see Chap 6). The program execution starts from top to bottom and has nothing to do with the value of block number. Each instruction starts with an English letter (A~Z), followed by an integer or floating number, depending on the type of instruction the number is associated with. If the number represents a coordinate, it can be positive (+) or negative (-). In general, the program instructions can be divided into four categories.

1. Function command : G-code. A CNC command to instruct the tool to do a work, such as straight/circular cutting, moving, etc.
2. Position command : X, Y, U, and V. Coordinate function to instruct the cutting tool to move from the current location to the next location.
3. Feed-rate command : F-code. A command specifying the cutting speed for cutting tool.
4. Auxiliary command : M, L, etc. A command to instruct the peripheral equipment to do an action, such as valve or coolant on/off, etc.

Each command (or function) code has a fixed format and a special meaning to the CNC controller and it must be strictly followed when writing a program. The system will not accept the command if the format is in error. Otherwise, a machine error will result. More on part program and function codes in Chapter 3. Followings are the command codes that are used in HUST H2N controller.

A, B : Variable #1 and #2 in G65 Macro function.
F : Feed-rate, decimal.
G : Function G-code, integer.
I, J : The X- and Y-axis component of the arc radius @ the start point, decimal.
L : Repetition counter, or operation mode designation in G65, integer.
M : Function code for peripheral equipment, integer.
N : Program block (sequence) number, integer.
O : Program number, integer.
P : Subprogram code, or variable #3 of G65 command, integer.
R : Arc radius, decimal.
S : Spindle speed, integer.
U, V : Incremental command in X- and Y-axis respectively, decimal.
X, Y : Position command in X- and Y-axis respectively, decimal.

2.4 Coordinate System

The cutting action is accomplished when the tool is moving along a specific path from point A to point B, which represent the shape or the contour of a mechanical part. For the tool to follow the specific path, a computer program describing the shape of the part must be created. The shape or the contour is described by the Cartesian coordinate system.

2.4.1 Cartesian Coordinate

HUST H2N series uses the customarily 2-D Cartesian coordinate system as shown in Fig 2-1 with X and Y as designated axes. The intersecting point of these axes is the coordinate **origin** (also known as **work origin**), that is X=0, Y=0.

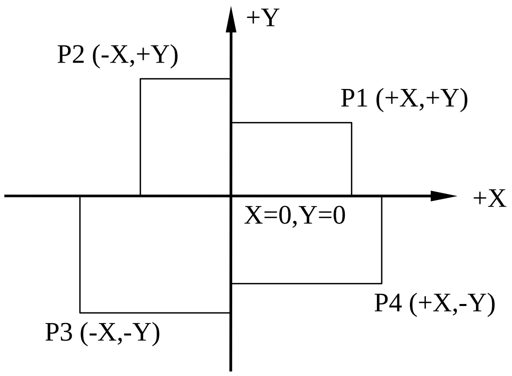


Fig 2-1 2-D Coordinate System

X and Y-axis can be used as **linear** axis or **spindle** axis. When applied as spindle axis, use your right hand to help you to determine the direction of rotation. By pointing your thumb in the +X or +Y, the direction of the rest 4 fingers will be pointing in the positive direction of rotation. We'll have more discussion on spindle axis, such as indexing table in Chapter 3.

2.4.2 Position Command (Coordinate)

The tool position function in HUST CNC controller can be expressed in either absolute or incremental coordinate, depending on the settings of MCM #77 and #78. The factory default settings are 1 for both MCM #77 and #78.

MCM #77 Setting	0	X-axis: Incremental	U-value: meaningless
	1	X-axis: Absolute	U-value: Incremental
MCM #78 Setting	0	Y-axis: Incremental	V-value: meaningless
	1	Y-axis: Absolute	V-value: Incremental

- Absolute Coordinate command

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

- Incremental Coordinate command

  The coordinates of all points describing the shape of the work-piece 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 moving in the positive direction, negative (-), otherwise.
  In programming, the absolute and the incremental coordinates are inter-changeable. Please refer to Fig 2-2 for following examples.

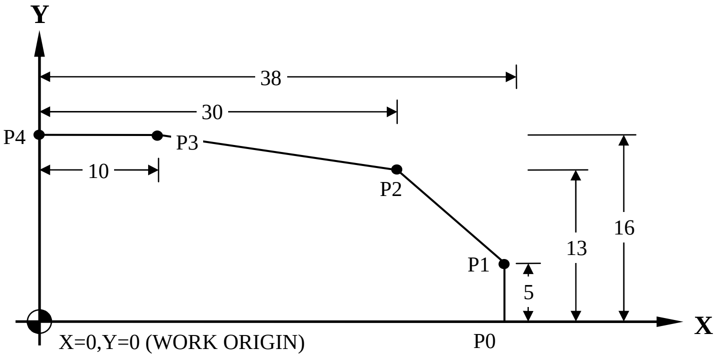


Fig 2-2 Absolute Coordinate

Absolute coordinate:

P0 to P1	G01 Y5.000 F200.00
P1 to P2	X30.000 Y13.000
P2 to P3	X10.000 Y16.000
P3 to P4	X0.000

Incremental coordinate:

P0 to P1	G01 V5.000 F200.00
P1 to P2	U-8.000 V8.000
P2 to P3	U-20.000 V3.000
P3 to P4	U-10.000

Mixed use of absolute and incremental coordinate:

P0 to P1	G01 Y5.000 F200.00
P1 to P2	U-8.000 Y13.000
P2 to P3	X10.000 V3.000
P3 to P4	U-10.000

In an absolute coordinate, a calculation error of one point will not affect the positioning of next point. In an incremental coordinate, however, an error at a location will affect the positioning of all subsequent points. There isn't any rule as to when to use the incremental or the absolute coordinate. The mixed use of both coordinates appears to be the most convenient.

2.4.3 Work Origin/Work Coordinate

The work origin is the coordinate origin and 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 reason to call it as work origin is to differentiate it from the machine origin (HOME location) to be discussed in the next section.

The work origin can be anywhere inside the machine working range. The user should determine the location of this point before making any coordinate calculations. Once the origin is selected, store the coordinate of this point with respect to the machine origin in MCM parameter #1~ #12 (see Chap 7). The best selection is the one that will make the coordinate calculation simple and easy.

2.4.4 Machine Origin (HOME Location)

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 (see Chap 4). The coordinate obtained using the machine origin as the 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 HOME (one axis at a time) on a CNC controller, the cutting tool will move to the machine origin. If the X- or Y-axis is used as rotating axis, the HOME location is equal to zero (0) degree.

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

2.5 HUST H2N Control Range

The minimum/maximum programmable range for HUST H2N controller is as follows. Note that the range of control may be limited by the working range of user's machine.

	Metric, mm
Min. setting unit	0.001
Max. setting unit	9999.999
Min. moving unit	0.001
Max. moving unit	9999.999
Max. travel distance	9999.999

	Metric Unit
G-code	G00~G99 (The 1st 0 may be omitted)
M-code	M000~M999 (The 1st 0 may be omitted)
S-code	S1~S999,999 rpm
F-code	0.01~80,000.00 mm/min
X, Y, U, V, I, J, R	0.001~±9,999.999 mm
G04	0~8,000.000 seconds
Program number	0~999
Memory capacity	72 K
Ball-screw compensation	0~255 pulses (related to tool resolution)
Max. response speed	500 KPPS

3 FUNCTION CODES

This chapter discusses the meanings and applications of function codes, such as G, F, M and S-code, and the format of their usage.

3.1 G-code Definition

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

1. <u>One-shot G-codes</u>

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 X30.0 Y40.0
N20 G4 X2.0              ..... G04 is one-shot G-code, effective only in this
block.
N30 X20.0 Y50.0          ..... G04 no longer effective in this block. G0 is.
```

2. <u>Modal G-codes</u>

A modal G-code is a G-code that remains effective until another G-code in the same group is encountered. Following G-codes are in the same group for HUST H2N series.

G00, G01, G02, G03 ..... Same group
G43, G49 ..... Same group
G54~G59 ..... Same group
G98~G99 ..... Same group

Example:

```
N10 G0 X30.0 Y5.0        .....  G0 is effective in this block.
N20 X50.0 Y10.0          .....  No G-code specified, G0 remains effective.
N30 G1 X30.0 F200.0      .....  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	G-code	Function
* G00 #	Rapid Positioning	* G43	Tool Offset Compensation ON
* G01 #	Linear Cut	* G49 #	Offset Compensation OFF
* G02	Circular Cut (CW)		
* G03	Circular Cut (CCW)	* G54 #	1st Work Coordinate
		* G55	2nd Work Coordinate
G04	Pause	* G56	3rd Work Coordinate
G08	Reset origin of machine coord. (Home)	* G57	4th Work Coordinate
G10	Data Input	* G58	5th Work Coordinate
G11	Simple output control	* G59	6th Work Coordinate
G12	Simple input control		
G14	Simple I/O control	G65	MACRO command
G28	Go to the 1st Reference Point	* G98 #	Feed-rate with mm/min
G29	Return to last location from ref. point	* G99	Feed-rate with mm/revolution
G30	Go to the 2nd Reference Point		
G31	Skip Function		

* Modal G-code.
\# Power-on default G-code (Either G00 or G01 can be chosen as power-on default through proper MCM parameter setting).

3.2 Rapid Positioning (Traverse Speed), G00

Format: G00 X(U)____Y(V)____

X, Y : Position code (End point) in absolute coordinate.
U, V : Position code in incremental coordinate.

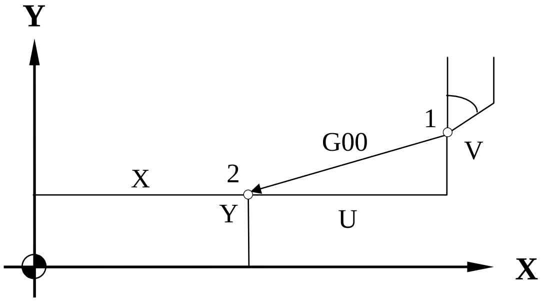


Fig 3-1 Rapid Positioning

X and Y can also be used as incremental command if MCM #77, #78=0. (See Chap 7) In this case, U and V are rendered meaningless. If absolute coordinates are desired for X and Y, set MCM #77 and #78 =1. G0 can be applied in CNC or Master/Slave mode. (Refer to Sec 3.4)

MCM #77 Setting	0	X-axis: Incremental	U-value: meaningless
	1	X-axis: Absolute	U-value: Incremental
MCM #78 Setting	0	Y-axis: Incremental	V-value: meaningless
	1	Y-axis: Absolute	V-value: Incremental

G00 is used to move the tool from the current location (the end point of previous block) to the coordinate specified by (X,Y) or (U,V) at high speed while the tool is NOT physically doing any cutting. It can control the movement of 1~2 axes. The moving speed is based on the combined values of MCM parameter #56 and #57 (See Chapter 7) and the Register #220. It should not exceed the max traverse speed calculated as shown in the next section. The factory default settings for MCM #56 and #57 are 10,000 mm/sec and for R#220 is 100% (R#220 range = 0~100).

For single axis function (G00 X___ or G00 Y___):
G00 speed (X-axis) = (MCM #56 value) * (R #220 in %)
G00 speed (Y-axis) = (MCM #5& value) * (R #220 in %)

For two axes function (G00 X___ Y___), the G00 speed will be calculated internally and the detail is shown in the example elow:

Ex: Fig 3-2, G00 move from point A to point B. (MCM #77 #78=1)
G00 X5.60 Y2.00 ….. X, Y in absolute coordinate
G00 U-3.05 V-3.00 ….. X, Y in incremental coordinate
G00 U3.05 Y2.00 ….. X, Y in mixed coordinate

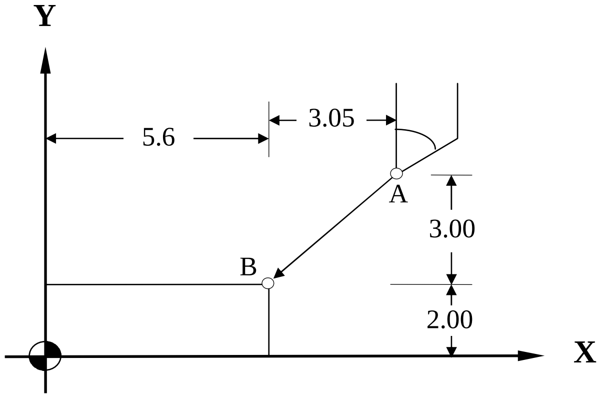


Fig 3-2 G00 Example

The tool moves rapidly to (X5.60,Y2.00) in the direction as shown by arrow in Fig 3-2. The speed calculations for each axis are done internally by the CNC controller, based on the settings of MCM #56 and #57. If the calculated value exceeds the MCM value (#56, #57), the controller will use that MCM value to re-calculate the traverse speed for the other axis. Following is an example.

Ex1: Assuming the values of MCM #56 and #57 are set as:
X-axis=3000.00 mm/min, Y-axis=5000.00 mm/min,

For G00 U-3.05 V-3.00 in Fig 3-2, the controller select the slower 3000.0 as the base to calculate the traverse speeds for other axes:

Fx = 3000.00
Fy = 3000.00 * (3.00/3.05) = 2950.82 (<5000 for MCM #57)

Both axes are within the settings of MCM #56 and #57, X-axis will move at 3000 and Y-axis at 2950.82 mm/min.

Ex2: Assuming the program is changed to G00 U-3.05 V-6.00.
With the same MCM #56, #57 setting, then

Fx = 3000.00
Fy = 3000.00 * (6.00/3.05) = 5901.64 (>5000 for MCM #57)

The calculated Fy is greater than MCM #57. Therefore, controller will re-calculate Fx based on Fy=5000 and the tool will move at the speed as follow.

Fy = 5000.00
Fy = 5000.00 * (3.05/6.00) = 2451.67 (<3000 for MCM #56)

3.2.1 Maximum Traverse Speed for G00

The maximum traverse speed allowed for the servo motor depends on the motor RPM, pitch length of the ball-screw, and the gear ratio (GR). It can be calculated using the equation below.

Fmax = 0.95 * Motor RPM * Pitch / GR
Where GR = Tooth number on ball-screw / Tooth number on motor.
0.95 = Recommended safety factor.

Ex: On X-axis, the motor rated 3000 rpm, ball-screw pitch = 5 mm, and GR = 5. The max traverse speed is

Fmax = 0.95 * 3000 * 5 / 5 = 2850 mm/min. The setting for MCM #56 = 2850.

Note that the maximum slope ratio of X/Y, Y/Z or Z/X for G00 traverse speed is 10000:1. This means if the F=2000.0 mm/min with machine resolution of 1 μm, the error in cutting path will be less than 0.2 %.

3.3 Linear Cutting, G01

Format: G01 X(U)____Y(V)____ F____

X, Y : The ending position code in absolute coordinate
U, V : The ending position code in incremental coordinate
F : Cutting speed. Can be used with any G-code, including G00, but will not affect G00 speed. F-code is a modal code. It affects the cutting speed for the blocks immediately followed until a new F-code is specified.

G01 is for the linear cutting motion and can control 1~2 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. The maximum cutting speed is limited by the setting of MCM #56 and #57. The actual cutting speed is determined by F-code and the Register #221 as follow. The factory default for MCM #56 and #57 are 10,000 and 100% for R#221.

F (actual) = (F-code value) * (R#221 value in %)

The current position of the tool is the starting point and the ending point is specified by X and Y position codes. The feed-rate (F-code) is a modal code. If the cutting rate is a constant for all program blocks, only one feed-rate in the beginning block needs to be defined. Unless the feed-rate is redefined, the previous F-code remains effective. The specified F-code is the rate along the cutting path (CNC mode, See Sec 3.4) and its component for each axis is obtained as below. The max ratio for Fx/Fy is 10000:1. U and V are of incremental values.

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

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

Following is a G01 example in absolute and incremental coordinate. Both programs will do the same cutting. (Fig 3-3)

Ex: Assuming the current tool position is X=4.60, Y=1.0.

G01 X2.01 Y2.0 F3000. ….. Absolute coordinate
G01 U-2.59 V1.00 F3000. ….. Incremental coordinate

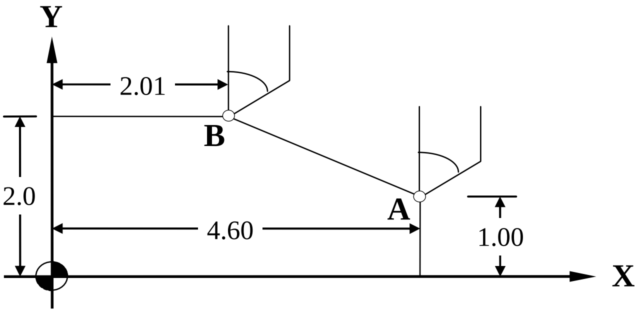


Fig 3-3 G01 Example

3.4 CNC and Master/Slave Mode

When CNC controller executing a program block, the servo motor is always subjected to a motion sequence as: Accelerate to speed at the starting point -- Maintain constant feed-rate specified in the block – Decelerate near target till stop or different speed at target. When the controller proceeds to execute the next block, the servo motor will repeat the same motion sequence. HUST controller provides 2 types of acceleration/deceleration for motor, namely CNC and Master/Slave mode. The shape of acceleration/deceleration can be either straight line or S-curve. This is done by proper settings of MCM parameters as shown below.

Acc/Dec Mode	MCM #37	MCM #89	G00 Acc/Dec	G01, G02 Acc/Dec
CNC Mode	0	0	Linear	Linear
	0	1	“S” curve	“S” curve
Master/Slave Mode X-axis as master	1	0	Linear	Linear
	1	1	“S” curve	“S” curve
Master/Slave Mode Y-axis as master	2	0	Linear	Linear
	2	1	“S” curve	“S” curve

CNC mode --

The servo motor will decelerate to a complete stop at the end of each program block, then the motor will accelerate again to the feed-rate specified in 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 current feed-rate. 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.

<u>MCM parameter #37</u> is for setting CNC and Master/Slave mode as follows:

Setting=0, CNC mode.
Setting=1, X-axis as Master axis, Y-axis as Slave axis.
Setting=2, Y-axis as Master axis, X-axis as Slave axis.

•**<u>CNC mode</u>**: MCM #37=0. Motor comes to complete stop at the end of each block. MCM #89 is used to determine the type of acceleration/deceleration as shown below.

MCM #37 setting	MCM #89 setting	G00 Acc./Dec.	G1, G2 Acc./Dec.
0	0	Linear	Linear
0	1	"S" curve	"S" curve

Ex 1: Fig 3-4, CNC mode (MCM #37=0), motor acceleration/deceleration in linear curve for G01 (MCM #89=0), absolute coordinate.

```
N10 G01 X100. F1000.          ….. Feed-rate Fx=1000, Fy=0
N20 G01 X200. Y100. F500.     ….. Feed-rate Fx=Fy=500.
N30 G01 X300. F250.           ….. Feed-rate Fx=250, Fy=0
N35 G01 X350. F100.           ….. Feed-rate Fx=100, Fy=0
```

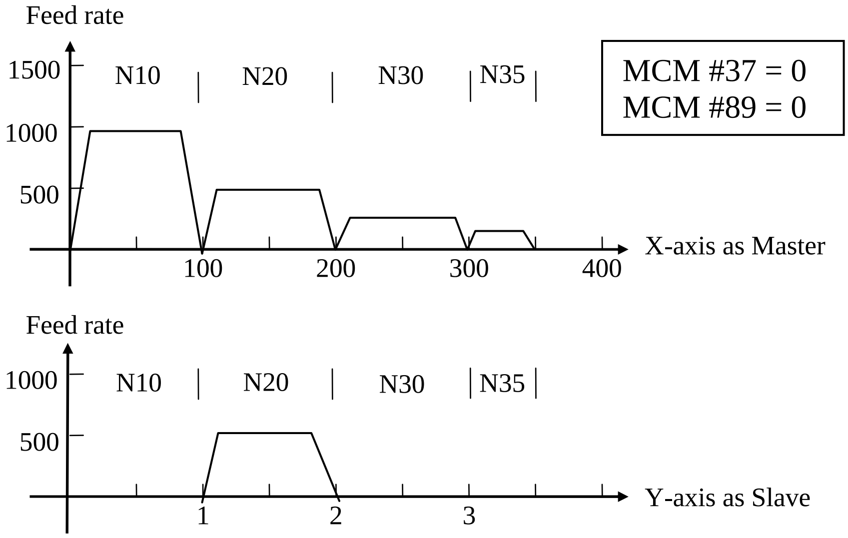


Fig 3-4 CNC mode with G01, Linear Acceleration./Deceleration

Ex 2: G01 U100.0 V200.0 F2000.00 (CNC mode)
Calculate the G01 feed-rate for X and Y-axis. Assuming MCM #56, #57 settings for G00 speed limit as: Ftrx=2000.0mm/min (X-axis), Ftry=1000.0 mm/min (Y-axis)

Composite vector for X and Y-axis = $(100^2 + 200^2)^{1/2}$ = 223.6
X-axis Feed-rate Fx = 2000.0 * (100/223.6) = 894.4
Y-axis Feed-rate Fy = 2000.0 * (200/223.6) = 1788.9
1788.9 > 1000.0 (Ftry), so the G01 feed-rate will be limited as:
Fy = Ftry = 1000.00
Fx = (894.4/1788.9) * 1000.0 = 500.00

- **Master/Slave mode**: MCM #37=1, X-axis as master and Y-axis as slave
  MCM #37=2, Y-axis as master and X-axis as slave

MCM #37	MCM #89	G00 Acc./Dec.	G1, G2 Acc./Dec.
1 or 2	0	Linear	Linear
1 or 2	1	"S" curve	"S" curve

In this mode, 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 current feed-rate. The feed-rate (F) in the block is for the master axis and the feed-rate of the slave axes will be calculated according to their displacements. Note that when in master/slave mode, there will be a minor error for the starting and the ending location of a circular cut.

Ex 1: N10 G01 X100. F1000.
N20 X200. Y100. F500.
N30 X300. F250.

Fig 3-5, X-axis as master axis, motor accel./decel. in Linear type for G01 (MCM #89=0), absolute coordinate.

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

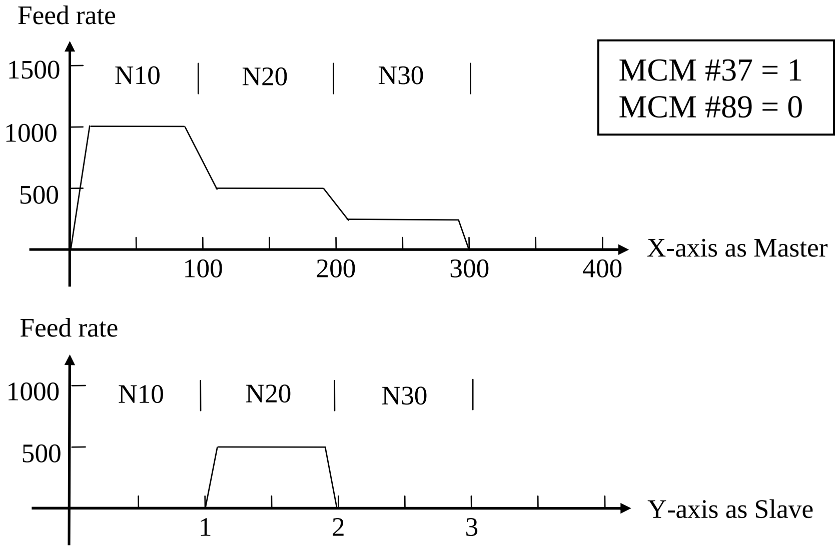


Fig 3-5 Master/Slave Mode, Linear Accel./Decel.

Fig 3-5A, X-axis as master axis, motor accel./decel. in "S" type for G01 (MCM #89=1), absolute coordinate.

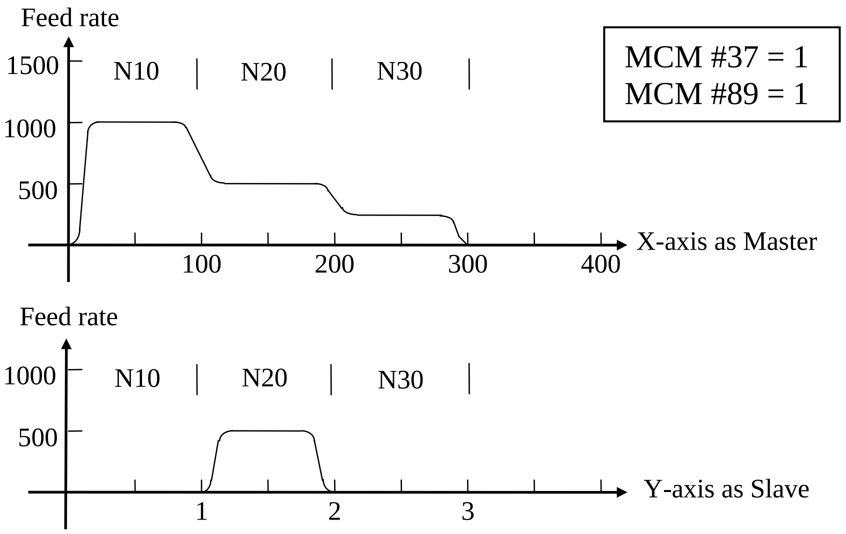


Fig 3-5A Master/Slave Mode, "S" Curve Accel./Decel.

Ex 2: N10 G01 X100. Y50. F1000.
N20 X200. Y75.
N30 X300. Y175.

Fig 3-6, X-axis as master axis with constant feed-rate, motor accel./decel. in linear type for G01 (MCM #89=0), absolute coordinate.

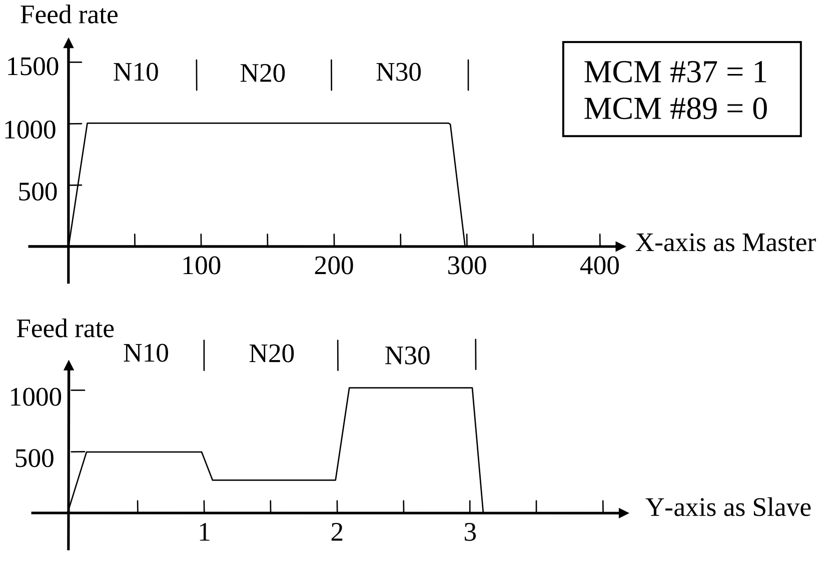


Fig 3-6 Master/Slave Mode with Constant F-rate for Master, Linear Accel./Decel.

In Example 2, the feed-rate of the slave axis is adjusted according to their incremental ratio. Note the small time required for accel./decel. between blocks and the distance traveled in this small amount of time can be estimated by:

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

F1, F2 = The feed-rates of 1st and 2nd block of the Slave axis, mm/min
T = The setting value of G01 in MCM #63

Therefore, if F1y=500. mm/min (N10 block), F2y=250. mm/min (N20) and MCM #63= 500 ms, the distance traveled at the beginning of N20 block for Y-axis is 1.04 mm. You can reduce this distance by reducing the setting of MCM #63.

Ex 3: G00 U100.0 V50.0 (X-axis as master, MCM #37=1)
Calculate the G00 feed-rate for master and slave axis with MCM #56, #57 settings as: Ftrx=2000.00 mm/min, Ftry=4000.00 mm/min.

Master axis	Fx = 2000.0
Slave axis	Fy = (50/100)*2000.00 = 1000.00

Fy < Ftry, so the G00 feed-rate will be based on the setting of MCM #56, Fx=2000.

Ex 4: G00 U100.0 V300.0 (X-axis as master, MCM #37=1)
Calculate the G00 feed-rate for master and slave axis with MCM #56, #57 settings as: Ftrx=2000.00 mm/min, Ftry=4000.00 mm/min.

Master axis	Fx = 2000.0
Slave axis	Fy = (300/100)*2000.00 = 6000.00

Fy > Ftry, so the G00 feed-rate will be limited by Ftry=4000 as:

Master axis	Fx = (4000/6000)*2000.00 = 1333.33
Slave axis	Fy = 4000.00

3.5 Circular (Arc) Cutting, G02 and G03

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

1. Arc cutting command code -- G02 (CW) or G03 (CCW)

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

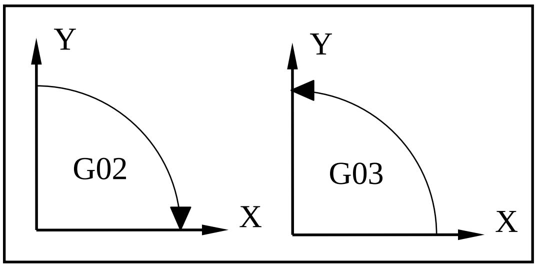


Fig 3-7 Directions of G02 and G03

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

U and V 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 or R

I, J are the X, Y-axis components of the arc radius, respectively and R is the arc radius. Either representation is acceptable. I, J can be (+) or (-) and their meanings are identical to U, V. The range for "R" is -9999.~+9999. mm. Do not use R representation if the arc angle is in the range of -1º~+1º or 179º~181º.

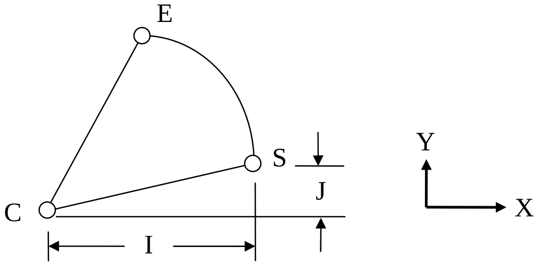


Fig 3-8 Circular Cutting with "I, J" Specified

4. Arc cutting rate -- F-code. The minimum feed-rate is 0.2 mm/min.

The actual feed-rate Ft = (F-value) * (Register R#221 value in %)

Format: (CW, clockwise)

G02 X(U)____Y(V)____ I____J____ F____

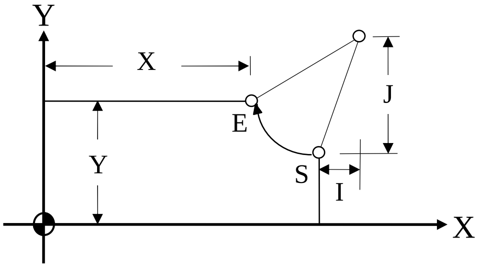


Fig 3-9 G2 Circular Cutting

Format: (CCW, counter-clockwise)

G03 X(U)____Y(V)____ I____J____ F____

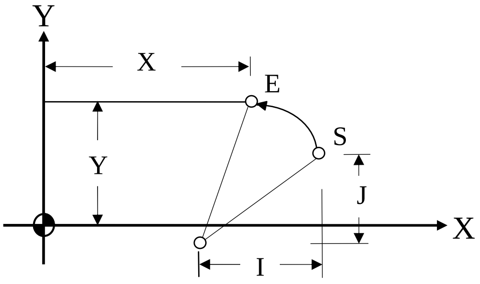


Fig 3-10 G3 Circular Cutting

Format: (With radius R method)

G02 X(U)____Y(V)____ R____F____

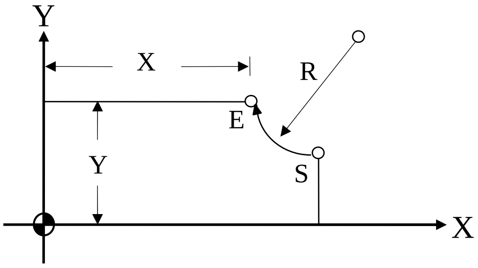


Fig 3-11 Circular Cutting with Radius "R"

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

The start point	X=50.0,	Y=20.0
The end point	X=30.0,	Y=30.0
Radius	R=25.0, or	I=0.0, J=25.0

1. G02 X30.0 Y30.0 J25.0 F300.
2. G02 U-20.0 V10.0 J25.0 F300.
3. G02 X30.0 Y30.0 R25.0 F300.
4. G02 U-20.0 V10.0 R25.0 F300.

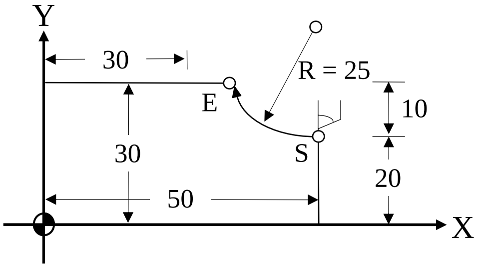


Fig 3-12 Arc Cutting Example

When applying radius R method, be careful in determining the sign of radius R. The range for "R" is -9999.~+9999. mm.

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

Ex: G02 X60.00 Y20.00 R50.00 F300.

As shown in the figure, this program will make a small arc cutting (less than 180 deg) in clockwise direction. If R=-50.00, the arc will follow R2 path.

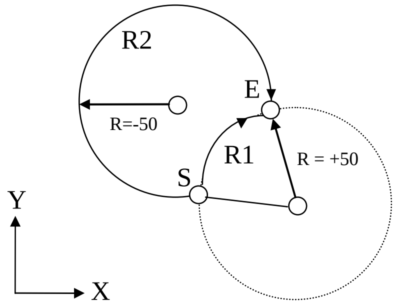


Fig 3-13 Arc Cutting with +R and -R

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. The F-value is the tangential cutting speed at the cutting point, which will be affected by the length of the arc radius. The reason is that the HUST CNC system adopts a constant max. error of 1 μm for arc cord height.

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 maximum tangential cutting speed is estimated with the formula:

$$Fc = 85 * \sqrt{R * 1000} \qquad \text{mm/min}$$

Where R= Arc radius in mm.

Ex 1: G02 X0.250 Y0.500 J0.25 F2000
Fc = 1344 mm/mim from formula above, which is smaller than the specified speed of 2000. So, the actual cutting speed is 1344.

Ex 2: G02 X0.250 Y0.500 J0.25 F1000
Fc = 1344 mm/mim from formula above, which is greater than the specified speed of 1000. So, the actual cutting speed is 1000.

3.6 Dwell (Hold), G04

Format: G04 X____ or G04 P____

X: Time unit in seconds, ranging 0.01~8000.0 seconds
P: Time unit in milliseconds.

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.

P-value will be used if both P and X-value exist. For decimal format (see Sec6.4) of 3/4, 5/2, or 6/1, use P-value. For 4/3 decimal format, both P and X-value are acceptable.

Example:

```
G01 X10.0 Y10.0 F1000.0
G04 X2.00                    ..... Hold for 2 seconds, then process to next block
G00 X0.0 Y0.0
```

3.7 Reset Origin of Machine Coordinate, G08

Format: G08 or G08 X___Y___

G08 is used to reset the machine coordinate of current location to zero. Another words, the current location becomes the origin of machine coordinate (or Home location). G08 is also applied to one axis as G08 X___ or G08 Y___. In Fig 3-14, point A is the origin of machine coordinate before executing G08 and point B is the current tool location. Once G08 is executed, point B becomes the origin of the machine coordinate.

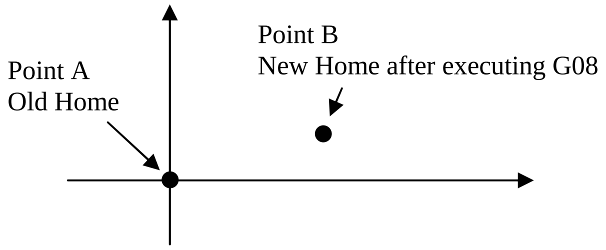


Fig 3-14

Ex 1: Point A -- The original Home location, which is also the G54 work origin as set by
MCM #1 =MCM #2 = 0.
Point B -- The current tool location with work coordinate (X=02, Y=35). The machine coordinate for point B is also (02, 35) since G54 work origin and Home location are the same. After executing G08 at point B, point B becomes new Home location. Since MCM #1 =MCM #2 = 0, point B is also the new work origin with (X=0, Y=0).

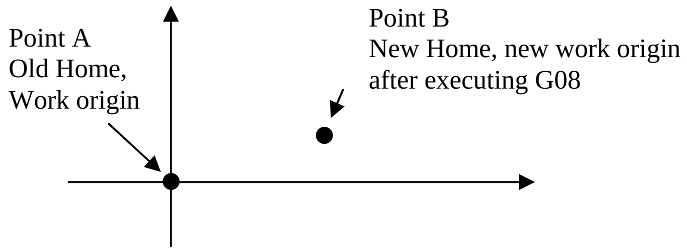


Fig 3-15

Ex 2: Point A -- The original Home location.
Point B -- G54 work origin which has a coordinate of (10, 10) with respect to old Home (point A) as set by MCM #1 =MCM #2 = 10.

Point C -- The current tool location with work coordinate of (X=15, Y=15). So, the machine coordinate for point C is (25, 25) with respect to point A. After executing G08 at point C, point C becomes new Home location. Since MCM #1 =MCM #2 = 10, point D is the new work origin with (X=10, Y=10) with respect to point C. So, the new work coordinate for point C is (-10, -10).

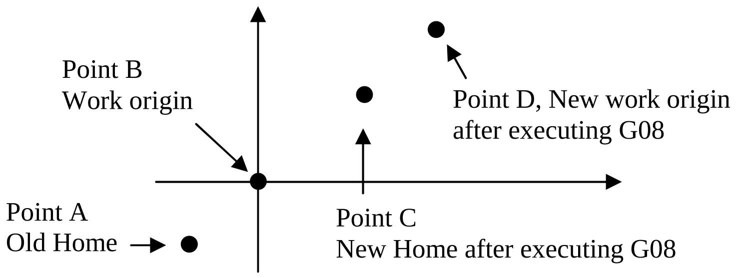


Fig 3-16

3.8 Manual Data Input Function, G10

G10 is a special function that can be used to input data in MDI mode. (Ref Sec 5.1.1 for MDI mode) Table 3-2 summarizes the G10 applications.

Table 3-2 G10 Application for HUST H2N Controller

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
G10 P2100	Download part program to memory from FLASHROM

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

Format: G54 ... or G55~G59
G10 X____Y____

Steps to set the origin of the specified work coordinate:

1. Execute HOME to move the tool to home position. (see Sec 8.1.1)
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.
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) away from the desired origin, execute as follows:

Press G10 Input, X20. Input, Y100. 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~#12 will be automatically revised. (See Chap 7)

3.8.2 Set Tool Offset Data By G10

Format: G10 X____ Y____ P____
or G10 U____ V____P____

P : 1~8 representing the group number in MCM #13~#36.
X, Y : Store the offset data of X, Y into the appropriate MCM #13~#36.
U, V : <u>Add</u> the offset data of U, V to the existing values of the appropriate MCM #13~#36.

Ex 1: G10 X0.02 Y0.03 P1

Press G10 Input, X0.02 Input, Y0.03 Input, P2 Input
Press CYCST key to complete the setting process. The offset data of X=0.02 and Y=0.03 are now stored in MCM #13 and #14, respectively.

Ex 2: G10 U0.01 V0.02 P2 (Assuming the existing value MCM #16=0.02, #17=0.03)

When you finish this operation, the offset data for MCM #16=0.02+0.01 = 0.03 and MCM #17= 0.03+0.02 = 0.05.

3.8.3 Set In-position Data By G10

Format: G10 X____ Y____ P100

X, Y : The in-position data to be stored in MCM #87, #88 respectively.
P100 : A fixed command for in-position data input.

Ex: G10 X0.02 Y0.05 P100
This command will cause the in-position data 0.02 to be stored in MCM #87 and 0.05 in MCM #88.

The in-position data are used in conjunction with the S-bit in PLC program. Assuming that the in-position data have been stored as the example above and that the tool is moving toward a target location. When the tool moves to the point with X=0.02mm and Y=0.05mm from the target location, S176 (for X-axis) and S177 (for Y-axis) will be set high (1), respectively. When the tool moves to the exact target location, S176 and S177 will become low (0). Following is one example showing the relationship between tool's in-position and S176 or S177 bit in PLC.

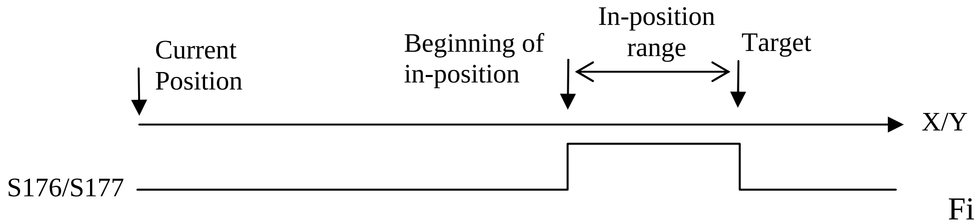


Fig 3-17

Ex: G01 X100.00 F1000.0 and MCM #87 = 500 μm = 0.5 mm

When the tool moves from current position to the point near 99.5, S176=0. When tool is at 99.5, S176=1 and it remains high (1) until X=100. Once the tool reaches 100 mm mark, S176=0.

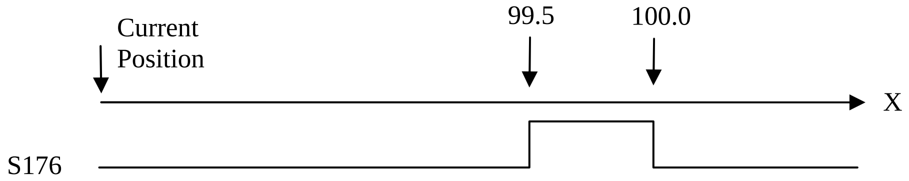


Fig 3-18

3.8.4 Set and Clear Counter Limit By G10

Format: G10 P200 L___ (Set)
G10 P201 (Clear)

L : The counter limit to be stored in MCM #66. The maximum number is 9,999,999.
P200 : A fixed command to input the counter limit data into MCM #66.
P201 : A fixed command to clear the counter MCM #65 to zero.

The counter limit (MCM #66) is the maximum number of cycles that the program has been repeatedly executed and the counter (MCM #65) is the one that records the actual execution cycles. Every time the program execution encounters M02, M30 or M99, the number in counter MCM #65 will increase by 1. When the number in MCM #65 is increased to a value that is equal to the number in MCM #66, the program execution will stop. If L=0, the counter MCM #65 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.

3.8.5 Set the Acceleration/Deceleration Time for G01 with G10

The acceleration/deceleration time is stored in MCM #63. 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 #63 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 $\#1 \leq 100$ mm
A/D time = 50 milliseconds if variable $100 < \#1 \leq 200$ mm
A/D time = 100 milliseconds if variable $\#1 > 200$ mm
(See Sec3.15 for G65 function)

```
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
```

3.9 Input/Output Control, G11, G12, G14

G11, G12, and G14 are the command codes that can be used to control ON and OFF for input and output signals during program execution. G11 is for output signals while G12 for input signals. G14 is for output but with a time delay. Table 3-3 summarizes the applications of input/output control with G11, G12, and G14.

Table 3-3 Summary of G11, G12, and G14 Function

Format	Function
G11	
G11 Pxxx	Turn ON output “xxx”, can be turned OFF by RESET.
G11 Pxxx Lyyy	Turn ON output “xxx”, can be turned OFF by FEED HOLD
G11 P-xxx	Turn OFF output “xxx”.
G11 P1xxx	Turn ON output “xxx”, can NOT be turned OFF by RESET.
G11 P-1xxx	Turn OFF output “xxx”, RESET is NOT effective.
G11 P2xxx Lyyy	Program execution on hold till the specified output “xxx” gone thru “yyy” times of ON/OFF cycles (2 msec/cycle).
G11 P-2xxx Lyyy	Execution of subsequent program block synchronizing with ON/OFF signal of output “xxx” by “yyy” cycles.
G11 P3xxx Lyyy	Output “xxx” remains ON, to be turned OFF when the input “yyy” is ON.
G12	
G12 Pxxx	Program execution of next block is on hold and to start when the input signal (Ixxx) is ON.
G12 Pxxx Lyyy	Program execution of next block is on hold and to start with the rising edge of the input signal (Ixxx).
G12 P-xxx	Program execution of next block is on hold and to start when the input signal (Ixxx) is OFF.
G12 P-xxx Lyyy	Program execution of next block is on hold and to start with the falling edge of the input signal (Ixxx).
G 14	
G14 Pxxx	In Master/Slave mode, the ON timing of output “xxx” will be delayed by ½ of the accel /decel time in the MCM parameter.
G14 P-xxx	In Master/Slave mode, the OFF timing of output “xxx” will be delayed by ½ of the accel /decel time in the MCM parameter.
G14 P1xxx	Similar to “G14 Pxxx”, but RESET signal is NOT effective.
G14 P-1xxx	Similar to “G14 P-xxx”, but RESET signal is NOT effective.

3.9.1 Output Control, G11

1. **G11 Pxxx, G11 P-xxx, G11 Pxxx Lyyy**

   "xxx" represents the output number, ranging 000~015.

   - G11 Pxxx: To turn ON the output specified by "xxx" which can be turned OFF either by RESET or by G11 P-xxx.
   - G11 P-xxx: To turn OFF the output "xxx".
   - G11 Pxxx Lyyy: If "yyy"=0 or void, this command is the same as G11 Pxxx. If "yyy"= any number, this command is to turn ON output "xxx" which can be turned OFF by Feed-Hold function. It will remain OFF as long as the Feed-Hold is on. When Feed-Hold is released, output "xxx" will be back ON. Use RESET or G11 P-xxx to turn output "xxx" OFF.

Ex1:
N10 G00 X30. F1000.
N20 G11 P13
N30 G00 X60.
N40 G00 X100.
N50 M30

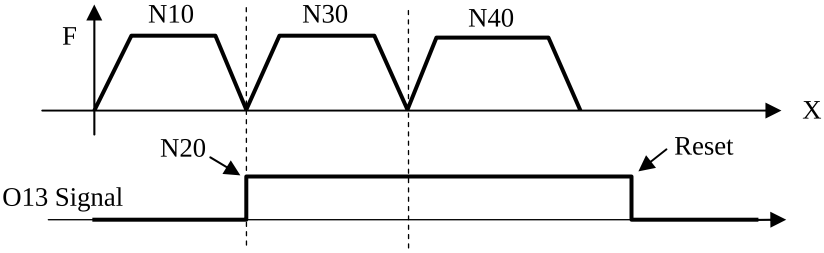


Fig 3-19

Ex2:
N10 G00 X30. F1000.
N20 G11 P13 L1
N30 G00 X60.
N40 G00 X100.
N50 M30

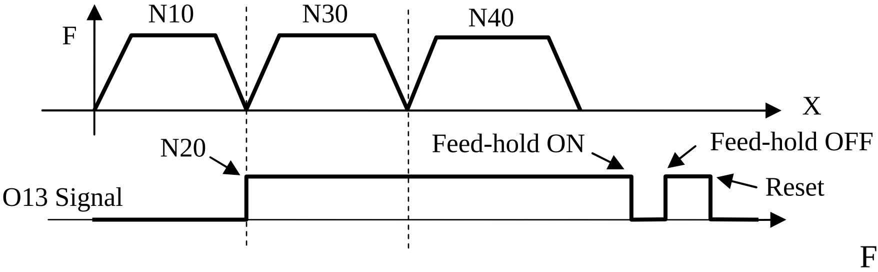


Fig 3-20

Ex3:
N10 G00 X30. F1000.
N20 G11 P13
N30 G00 X60.
N40 G11 P-13
N50 G00 X100.
N60 M30

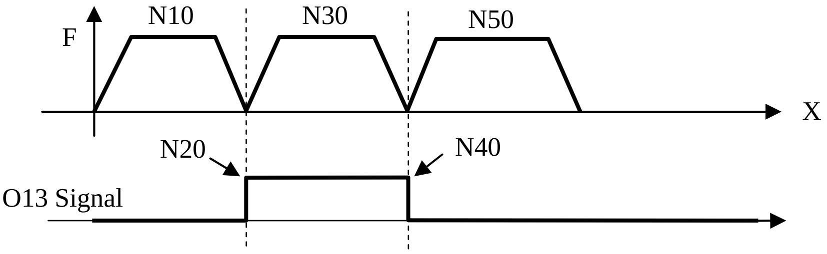


Fig 3-21

2. G11 P1xxx, G11 P-1xxx

"xxx" represents the output number, ranging 000~015.

G11 Pxxx is used to turn ON the output specified by "xxx" which can be turned OFF by P-xxx only. RESET is not effective.

Ex1: N10 G00 X30. F1000.
N20 G11 P1010
N30 G00 X60.
N40 G11 P-1010
N50 G00 X100.
N60 M30

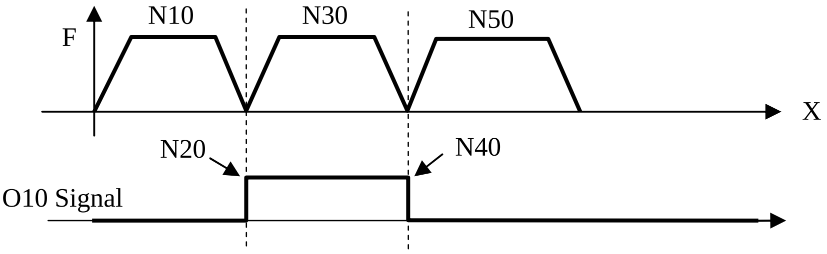


Fig 3-22

Ex2: N10 G00 X30. F1000.
N20 G11 P1010
N30 G00 X60.
N40 G00 X100.
N50 M30

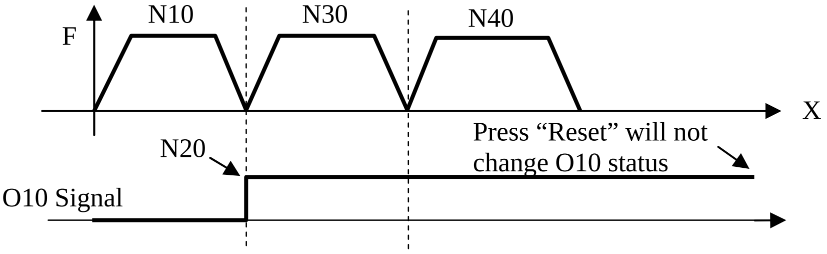


Fig 3-23

3. G11 P2xxx Lyyy (Wait mode)

"xxx" represents the output number, ranging 000~015.
"yyy" represents the number of ON/OFF cycle (every 2 milliseconds).

G11 P2xxx is used to force the program execution to wait till the specified output (Oxxx) ON/OFF the number of cycles specified by Lyyy is reached, then execute the next block. Note that it take 2 msec for each On/Off cycle.

```
Ex1:  N10 G00 X30. F1000.
      N20 G11 P2005 L20        (O005 On/Off 20 times or 40 milli-seconds)
      N30 G00 X60.
      N40 G00 X100.
      N50 M30
```

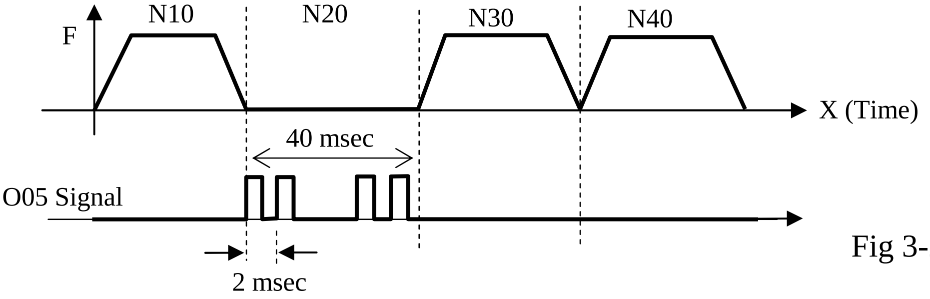


Fig 3-24

4. G11 P-2xxx Lyyy

"xxx" represents the output number, ranging 000~015.
"yyy" represents the number of ON/OFF cycle (every 2 milliseconds).

G11 P-2xxx is used to force the program block immediate below to be executed in synchronization with the specified output (Oxxx) ON/OFF the number of cycles specified by Lyyy. Note that it take 2 msec for each On/Off cycle. Another words, the duration of program execution for the affected block is equal to = 2 msec * Lyyy.

```
Ex1:  N10 G00 X30. F1000.
      N20 G11 P-2008 L15       (O005 On/Off 15 times or 30 milli-seconds)
      N30 G00 X60.
      N40 G00 X100.
      N50 M30
```

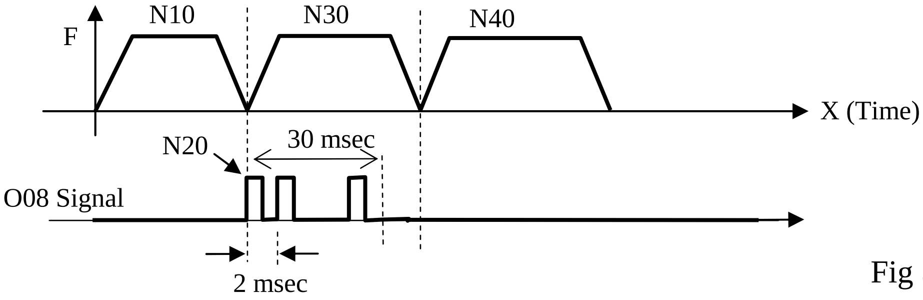


Fig 3-25

5. G11 P3xxx Lyyy

"xxx" represents the output number, ranging 000~015.
"yyy" represents the input number, ranging 000~023.

G11 P3xxx is used to command the specified output (Oxxx) to be turned OFF when the specified input signal Lyyy is ON.

```
Ex1:  N10 G00 X30. F1000.
      N20 G11 P3003 L20
      N30 G00 X60.
      N40 G00 X100.
      N50 M30
```

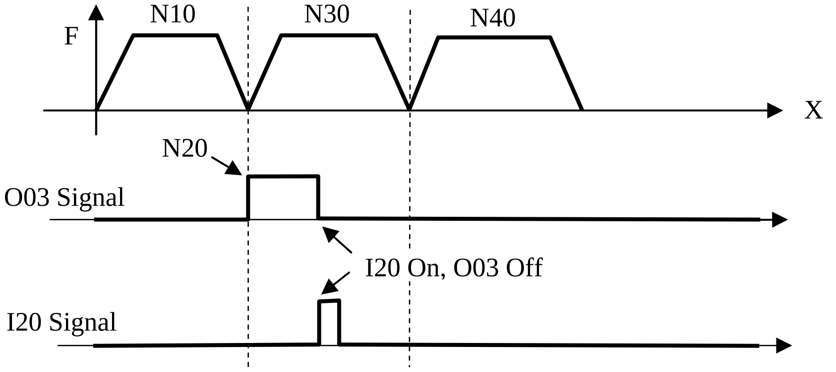


Fig 3-26

3.9.2 Input Control, G12

1. G12 Pxxx Lyyy

"xxx" represents the input number, ranging 000~023.
"yyy" represents a certain number (any value).

G12 Pxxx is to force the program execution of next block to start at the time when the input signal (Ixxx) is ON. If Lyyy has some value, the program execution will resume at the rising edge of input signal Ixxx.

```
Ex1:  N10 G00 X30. F1000.
      N20 G12 P3
      N30 G00 X60.
      N40 G00 X100.
      N50 M30
```

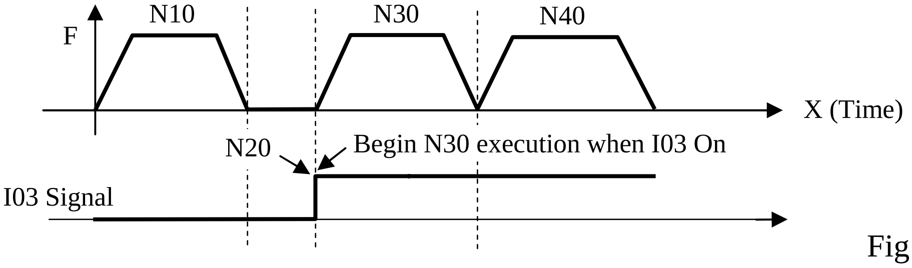


Fig 3-27

2. G12 P-xxx Lyyy

"xxx" represents the input number, ranging 000~023.
"yyy" represents a certain number (any value).

G12 Pxxx is force the program execution of next block to start at the time when the input signal (Ixxx) is OFF. If Lyyy has some value, the program execution will resume at the falling edge of input signal Ixxx.

Ex1:
```
N10 G00 X30. F1000.
N20 G12 P-4
N30 G00 X60.
N40 G00 X100.
N50 M30
```

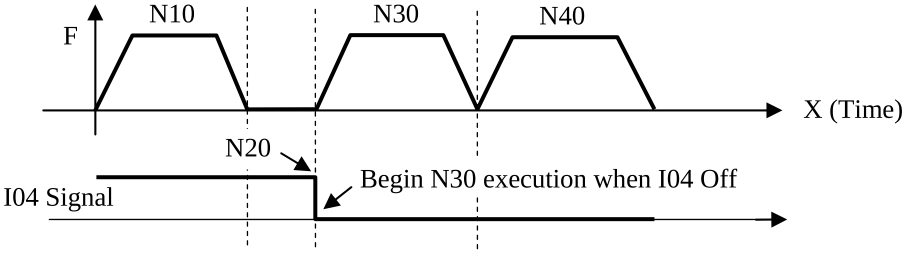


Fig 3-28

3.9.3 Output / Input Control, G14

1. G14 Pxxx and G14 P-xxx

"xxx" represents the output number, ranging 000~015.

Pxxx is used to turn ON the output number "xxx" while P-xxx is to turn it OFF. G14 is mainly used to control the ON/OFF timing of the specified output "xxx". In the standard CNC mode, the ON/OFF timing of output "Oxxx" with G14 Pxxx and P-xxx is the same as that with G11 Pxxx and P-xxx, respectively. In the Master/Slave mode, however, the ON/OFF timing of output "xxx" will be delayed by ½ of the acceleration /deceleration time as set in the MCM parameter #62~#63. (see Chapter 7) The example below explains the G14 Pxxx and P-xxx effect.

Ex: Assume that the Acceleration/Deceleration time in MCM = 30 msec.

For Fig 29 and Fig 30	For Fig 31
G00 X30. F1000.	G00 X30. F1000.
G11 P18	G14 P18
G00 X60.	G00 X60.
G11 P-18	G14 P-18
G00 X100.	G00 X100.
M30	M30

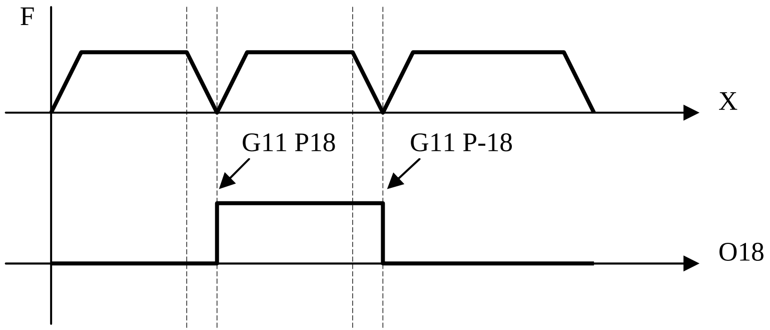


Fig 3-29 CNC Mode (G11)

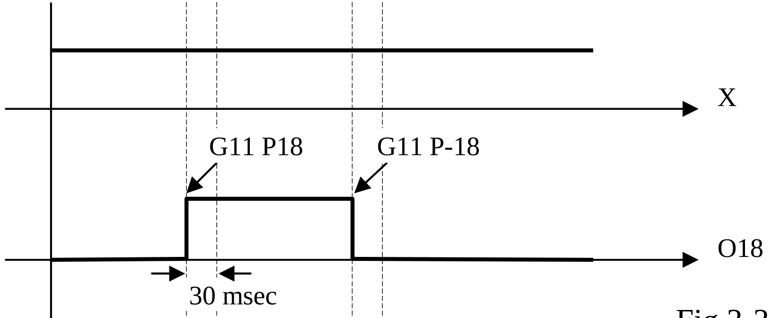


Fig 3-30 Master/Slave Mode (G11)

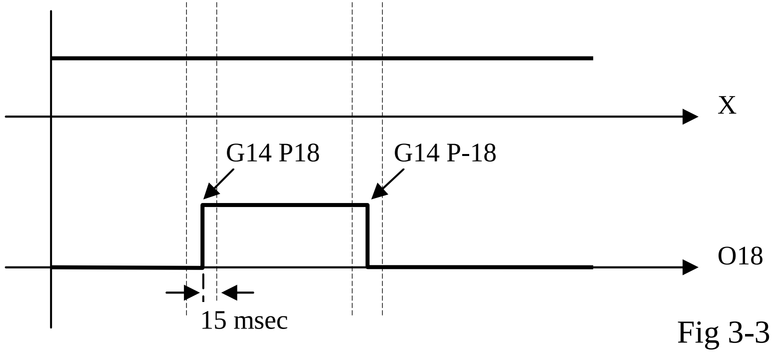


Fig 3-31 Master/Slave Mode (G14)

2. G14 P1xxx and G14 P-1xxx

“xxx” represents the output number, ranging 000~015.

P1xxx is used to turn ON the output number “xxx” while P-1xxx is to turn it OFF. These functions are similar to Pxxx and P-xxx. The only difference is that the RESET signal is not effective and will not alter the ON/OFF status of output xxx.

In the standard CNC mode, the ON/OFF timing of output “Oxxx” with G14 P1xxx and P-1xxx is the same as that with G11 P1xxx and P-1xxx, respectively. In the Master/Slave mode, however, the ON/OFF timing of output “xxx” will be delayed by ½ of the acceleration/deceleration time as set in the MCM parameter #62~#63. (see Chapter 7) The example below explains the G14 effect.

Ex: Assume that the Acceleration/Deceleration time in MCM = 30 msec.

<u>For Fig 32, Fig 33</u>	<u>For Fig 34</u>
G00 X30. F1000.	G00 X30. F1000.
G11 P1020	G14 P1020
G00 X60.	G00 X60.
G00 X100.	G00 X100.
M30	M30

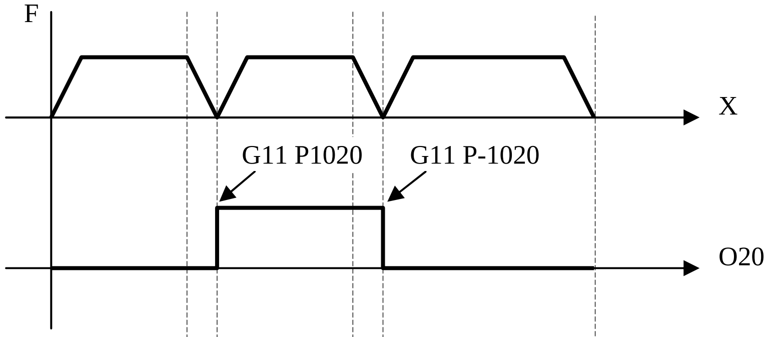


Fig 3-32 CNC Mode (G11)

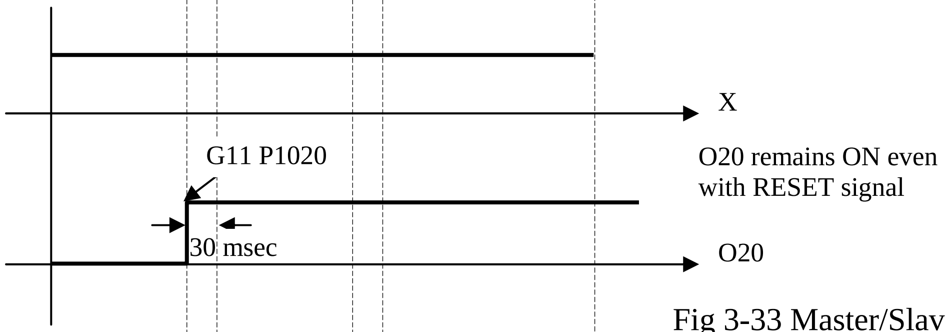


Fig 3-33 Master/Slave Mode (G11)

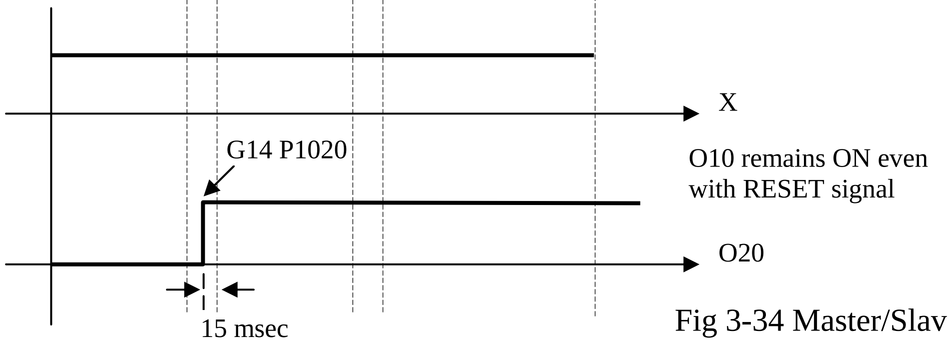


Fig 3-34 Master/Slave Mode (G14)

3.10 Move To The First Reference Point, G28

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

The coordinates of the first reference point is stored in MCM parameters #38~#39. The number associated with X, Y does not have any meaning, but you have to have a number to input X, Y into the CNC buffer. When encountering this command during cutting, the tool will move to the first reference point as set in MCM parameter

#38~#39 for the axis specified in the G28 block, regardless of what numbers are with X, Y-axis.

The coordinates of MCM parameters #38~#39 are determined by users, based on the machine origin being at X=Y=0. This reference point is normally selected at some convenient location during machining. Therefore, if X=Y=0 is selected for MCM #38~#39, G28 command will cause the tool moving to machine origin.

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 G28 block.

Example:

G49	..... Tool offset compensation canceled
G28 X10.	..... Tool moves to the 1st ref. point in X-axis, no motion in Y-axis.

3.11 Return To Previous Location From Reference Point, G29

Format:	G29	..... Return from the ref. point in all axes
or	G29 X____Y____	..... Return from the ref. point in X and Y-axis

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 code, as indicated by X, Y in the program block. G29 command can not be used alone, instead it is used following a G28 or G30 command. Again, the number associated with X, Y does not have any meaning, but you have to have a number to input X, Y into the CNC buffer.

Example: (only G29 in the block)

G01 X60.00 Y0.00	..... Tool at the location of X60., Y0., Z30.
G28	..... Tool moved from (X60, Y0.) to the 1st ref. point.
G29	..... Tool returns from the ref. point to (X60, Y0.)

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

Format: G30 X____Y____

The method of application for this command is the same as for G28. The coordinates of this reference point are set in the MCM parameters #40 and #41.

3.13 Skip Function, G31

Format: G31 X(U)____Y(V)____L____P____

X, Y : Position code in absolute coordinate
U, V : Position code in incremental coordinate
L : Input signal repetition number. L0, L1 or void means that the input signal is to be received only once for G31 to be effective. So, L2 means the signal is to be received 2 times before G31 becomes effective.

P : Input signal designation. Total 8 input signals are available as shown below. If P-code is not specified, the input signal is treated as “I07”.

P___	P0	P1	P2	P3	P4	P5	P6	P7
Input	I0	I01	I02	I03	I04	I05	I06	I07

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

When G31 is doing linear cutting, the feed-rate will be the one in effect. If F-rate is not specified in the program, the value in MCM #44, #45 will be used. G31 is a one-shot G-code.

Example:
```
N10 G1 X10. Y10.
..........
N40 G49
N50 G31 U100.0 F100. L2    ..... I07 to be received twice to cause SKIP
N60 G1 V25.00
N70 X90.00 Y30.00
```

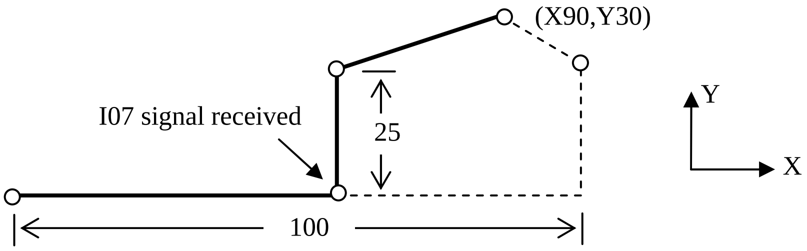


Fig 3-35 G31 Skip Function

In Fig 3-35, the dashed line represents the original path without SKIP function and the solid line is the actual tool path when the SKIP function is established. Prior to using SKIP function, do NOT use the tool compensation command. The input signals can be anything such as ON/OFF of electric current, light or laser beam, etc.

The potential applications of SKIP function are enormous. This function is to be processed through PLC program. Followings are some examples:

1. The length measurement of a work-piece using a metal tongue. When the metal tongue is touched, SKIP function is established and the measurement is taken.
2. Graphical paper cutting. Paper cutting by length measurement in not accurate due to stretching. The better way is done by sensing the recurring pattern on the paper to produce a SKIP function which then induces a cutting action.
3. Coil spring cut-off during manufacturing process. The details are not to be elaborated here.

3.14 Work Coordinate System, G54~G59

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

1. Machine Coordinate (Home)
2. Work Coordinate (G54~G59)

3.14.1 Machine Coordinate (Home)

The origin of the machine coordinate system is a fixed point on the machine. Its location is 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 <u>HOME position</u> or <u>Machine origin</u>. 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 for each axis.

Occasionally for the convenience of cutting operation, it becomes necessary to set another origin that is slightly shifted from the machine origin. Such origin is called <u>HOME SHIFT</u>. The amount of shift is set in <u>MCM #71~#72</u>. 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 #71~#72 are zero (0), the HOME SHIFT is the HOME position. The methods to find HOME position are:

1. Manually execute HOME from the control panel. (See Sec 8.1.1)
2. Use G28 or G30 by setting the reference coordinates in the MCM to zero for all axes.

3.14.2 Work Coordinate System, G54~G59

Format: G54~G59 X____Y____

X, Y: The coordinate (with respect to the specified work origin) for the current tool to move to.

HUST H2N series provides 6 sets of work coordinate system with their origins being stored in the <u>MCM parameter #1~#12</u>. The coordinates of these work origins are the coordinates with respect to the <u>machine origin</u>. The coordinate data or these origins can be entered into the proper MCM number by one of the following methods:

1. G10 command	--- To be discussed in this section
2. Direct input in MCM mode	--- See Chap 7
3. Input in the PLC program	--- See Connecting Manual

The application of G54~G59 command is explained in the example below. The advantage of using these work origins is the <u>simplification of the coordinate calculations</u> for the work-piece. Fig 3-36 shows six geometric cutting patterns with six work origins for G54~G59 as follows:

Work Coord.	MCM #	Machine Coord. X-axis Setting	Machine Coord. Y-axis Setting
G54	1 (X), 2 (Y)	-70.0	-10.0
G55	3 (X), 4 (Y)	-80.0	-30.0
G56	5 (X), 6 (Y)	-80.0	-50.0
G57	7 (X), 8 (Y)	-70.0	-50.0
G58	9 (X), 10 (Y)	-40.0	-60.0
G59	11 (X), 12 (Y)	-20.0	-40.0

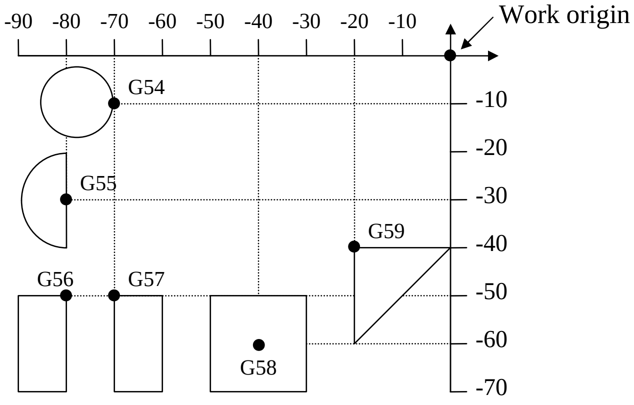


Fig 3-36 G54~G59 Work Origin

Once the work origins have been set in the MCM #1~#12, 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 G54                      -- Select G54
N2 G0 X0. Y0.               -- Move to G54 work origin (Machine Coord, -
70/-10)
N3 G2  I-7.0 F200.0         -- Cut a circle in CW with R=7.0
N4 G55                      -- Select G55 work coord.
N5 G0 X0. Y0.               -- Move to G55 work origin (Machine Coord, -
80/-30)
N6 G1 V10.0 F300.           -- Line cut along Y-axis for 10 mm @ 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 for 10 mm @ 300
mm/min
N9 G28                      -- Move to the 1st reference point
N10 M2                      -- Program end
```

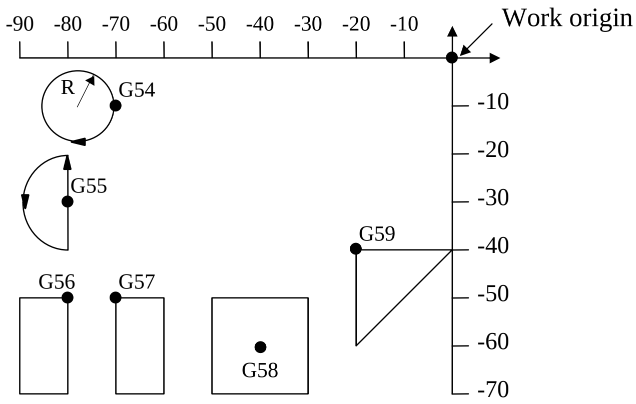


Fig 3-37 The Application of G54~G59

Notes on G54~G59 command:

1. When power-on or pressing RESET key, the default is the G54 command.
2. When G54~G59 command is executed, the machine coordinate of the new origin is also changed accordingly.
3. To change the location of current work origin, simply execute G10 X___Y___.

3.15 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 3-4 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 x #k
G65	L5	Division	#i = #j / #k
G65	L6	Place Data into Variables	#j =Data= #i = #i +1 = #i +2 = ...=#i + #k
G65	L7	Copy Variables	#i = #j for consecutive number of #k
G65	L11	Logic OR	#i = #j .OR. #k
G65	L12	Logic AND	#i = #j .AND. #k
G65	L13	Logic exclusive XOR	#i = #j .XOR. #k
G65	L14	ROL, rotate to left	#i = #j .ROL. #k
G65	L15	ROR, rotate to right	#i = #j .ROR. #k
G65	L16	LSL, shift to left	#i = #j .LSL. #k
G65	L17	LSR, shift to right	#i = #j .LSR. #k
G65	L21	Square Root	$\#i = \sqrt{\#j}$
G65	L22	Absolute	#i = |#j|
G65	L23	Remainder	#i = #J - trunc(#j/#k) x #k trunc:(Discard result that is less than 1)
G65	L26	Combined Mul/Div Operation	#i = (#i x #j) / #k
G65	L31	Sine of an angle (Sin)	#i = #j x Sin(#k)
G65	L32	Cosine of an angle (Cos)	#i = #j x 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 3-4)

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 = Data = #i = #i +1 = #i +2 = ...=#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, #1340, #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 = 00000110 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.

Bit	15	14	13	…..										2	1	0
	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

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

Before:

Bit	15	14	13	…..										2	1	0
	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0

After G65:

Bit	15	14	13	…..										2	1	0
	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

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

Before:

Bit	15	14	13	…..										2	1	0
	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0

After G65:

Bit	15	14	13	…..										2	1	0
	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1

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.

Bit	15	14	13	…..										2	1	0
	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1

↑ →

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

Before:

Bit	15	14	13	…..										2	1	0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1

After G65:

Bit	15	14	13	…..										2	1	0
	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

↑ →

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
	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

←

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

After G65: ←

Bit	15	14	13	…..										2	1	0
	0	0	0	0	0	0	0	0	0	0	1	1	0	1	0	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
	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

→

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

After G65: →

Bit	15	14	13	…..									2	1	0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1

→

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 - trunc(#j / #k) x #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 - trunc(#5 / 12) x 12)
#10 = 99 - trunc(99 / 12) x 12 = 99-(8*12) = 3

18. Combined Multiplying then Dividing Operation, #i = (#i * #j) / #k
G65 L26 P#i A#j B#k

HUST H2N 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
P#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 x 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 ^{o}.
2. Since Sin(#k) is always less than 1 and HUST H2N 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 ^{o} and store it as variable #1
G65 L31 P#1 A1000 B6000
The result P#1 = 1000 * Sin 60 ^{o} = 866

20. Cosine of an Angle, #i = #j x 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 ^{o}.
2. Since Cos(#k) is always less than 1 and HUST H2N 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 ^{o} and store it as variable #1
G65 L32 P#1 A1000 B4500
The result P#1 = 1000 * Cos 45 ^{o} = 707

21. Arc-tangent of a number, #i = 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 Tan^{-1} (577/1000) and store it as variable #1
G65 L34 P#1 A577 B1000
The result P#1 = Tan^{-1} (577/1000) = 0003000 =30 ^{o}

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)….. 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	015	014	013	012	011	010	009	008	007	006	005	004	003	002	001	000
	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	C31	C30	C29	C28	C27	C26	C25	C24	C23	C22	C21	C20	C19	C18	C17	C16
	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	S31	S30	S29	S28	S27	S26	S25	S24	S23	S22	S21	S20	S19	S18	S17	S16
	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	A31	A30	A29	A28	A27	A26	A25	A24	A23	A22	A21	A20	A19	A18	A17	A16
	0	0	0	0	0	0	0	0	1	1	1	0	0	1	0	1

Table 3-5 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

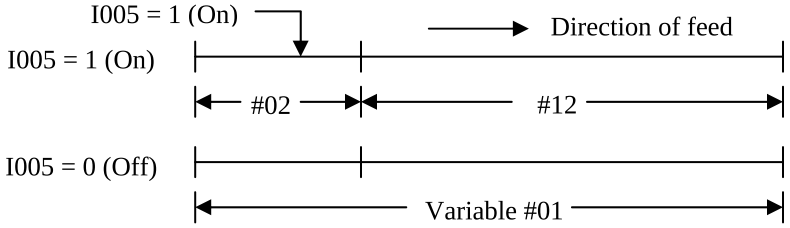


Fig 3-38

```
         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
```

3.16 Feed-rate Mode Control, G98, G99

Format: G98 --- Feed-rate per minute, mm/min
G99 --- Feed-rate per revolution, mm/revolution

HUST H2N provides two types of feed-rate, mm/min and mm/rev. G98 is the power-on default setting. Generally, G98 will suffice most applications. In special application such as coiling machine, G99 will be more useful. The following rules must be observed when applying G99.

1. G99 can be applied only with G01.
2. The rotating spindle (Y-axis) for coiling must be equipped with a feedback encoder. This feedback signal is then used to control the G99 feed-rate on X-axis, mm/revolution. So, there will be no feed on X-axis if no encoder signal is received from Y-axis.
3. When in G99 mode, the MCM #37~#38 settings are no longer effective and the motor acceleration/deceleration mode is always of linear type.
4. When in G99 mode, be sure the following MCM settings are correct.

MCM #	Explanation
90	Acceleration/Deceleration time
91	Spindle encoder pulse number
92	Spindle motor speed at 10 volts

Ex: G99 ... mm/rev mode
G01 X30.0 F0.2 ... X-axis will move 0.2 mm per spindle revolution and will have 150 revolutions to X=30.0.

3.17 Working Program

3.17.1 Linear and Circular Repetitive Indexing, G00, G01, G02, G03

Format: (Recommend to use incremental coord, U, V)

G00 X(U)____ Y(V)____ L____ M____
G01 X(U)____ Y(V)____ L____ M____
G02 X(U)____ Y(V)____ I____ J____ R____L____ M____
G03 X(U)____ Y(V)____ I____ J____ R____L____ M____

X, Y : Absolute coordinate.
U, V : Incremental coordinate.
I, J : X and Y-axis component to determine the arc center.
R : Arc radius.
L : The number of indexing for the length specified.
M : User defined M-code for indexing tool. Note that the first indexing will be executed before the first tool move.

1. Use G00, G01 for linear indexing and G02, G03 for circular indexing.
2. The first indexing by M-code will be executed before the first tool move.
3. The distance between indexing is specified by the incremental coordinate.

4. This function block must start with G00, G01, G2, G3. Otherwise, it will not work.

Ex: G00 X12.00 L12 M5, with current position at 0.0.

Execute 1st M5 indexing.
1st move to location = 12.
Execute 2nd M5 indexing.
2nd move to location = 24.
Execute 3rd M5 indexing.
3rd move to location = 36.
....
Execute 12th M5 indexing.
12th move to location = 144.

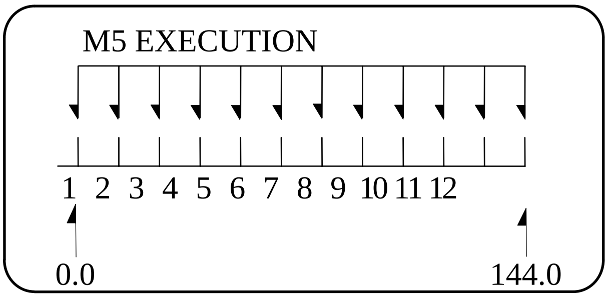


Fig 3-39 Linear Indexing

3.17.2 Auxiliary Functions, M-codes, S-codes

HUST CNC 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 followings are HUST CNC system M-codes and users 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. The program will stop only when the C-bit, C026=1. See 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. Please refer to PLC interface in "HUST H2N Connection Manual" for details. Please note the followings about M-codes.

1. When executing any of M-codes, the controller will send the M-code strobe signal S024=1 to PLC.
2. When executing any of M00~M499, the PLC in the controller will send an M-code finish signal C032=1 to the controller. Another words, the controller will not execute the next line of program unless it receives this signal. For M500~M999, the controller will not wait for this M-code finish signal.

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

3.17.3 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 required from the main program. In doing this way, the structure of the program can be greatly simplified. 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:

O005	Program number (No 5 in this case)
....	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 8,000,000 times because the sub-program is ended with a M99 function.

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 Y 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 8-level for HUST CNC controller as below:

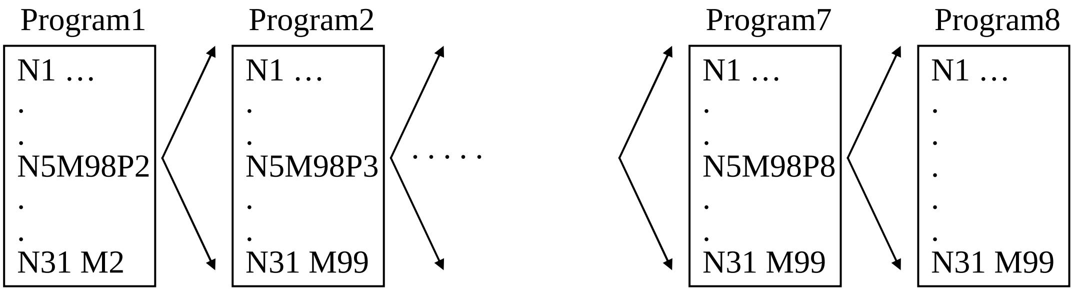


Fig 3-40 Sub-program Call

3.17.4 Variable Value Input in Sub-program from Main Program (G65)

Format: G65 Qxxx Axxx Bxxx Cxxx …..

Qxxx: Sub-program number

Axxx ~ Zxxx: The variable values that are to be transferred into the corresponding variables in the sub-program. The A~Z codes in G65 main program and the corresponding variable assignments in sub-program are in the table below.

G65 Main	Sub-program		G65 Main	Sub-program
Axxxx	#13101		Nxxxx	#13114
Bxxxx	#13102		Oxxxx	#13115
Cxxxx	#13103		Pxxxx	#13116
Dxxxx	#13104		Qxxxx	#13117
Exxxx	#13105		Rxxxx	#13118
Fxxxx	#13106		Sxxxx	#13119
Gxxxx	#13107		Txxxx	#13120
Hxxxx	#13108		Uxxxx	#13121
Ixxxx	#13109		Vxxxx	#13122
Jxxxx	#13110		Wxxxx	#13123
Kxxxx	#13111		Xxxxx	#13124
Lxxxx	#13112		Yxxxx	#13125
Mxxxx	#13113		Zxxxx	#13126

Ex: Main Program:

```
O001
G65 Q05 A10.000 B12.000 C2500.
M02
```

Sub-program:

```
O005
G01 X#13101 Y#13102 F#13103
G04 X4.
M99
```

The resulting values in the variables after being called from G65 are #13101=10.000, #13102=12.000, and #13103=2500.

4 TOOL COMPENSATION

The tool compensation in HUST H2N system is divided into two types as follows:

1. Tool Offset Compensation
2. Tool Radius Compensation -- Currently not available.

4.1 Tool Offset Compensation, G43, G49

Prior to applying tool offset compensation in cutting operation, tool offset data must be entered and stored in MCM parameter #13~#36, which can accommodate offset data for 8 sets of cutting tools. Currently, the tool radius compensation is not available.

All tools have some difference in length (or offset) after being installed on the machine, see Fig 4-1. The difference in tool length causes a minor shift of the coordinates for the tool tips. This difference must be figured in the program if you use more than one tool to cut the same work-piece. Normally, you designate a tool as standard tool and use it to set the work origin. The tip of the standard tool is normally designated as the HOME position, the offset data for other tools will become the machine coordinates with respect to the HOME position. These data can be stored in MCM #13~#36 by G10 function as discussed in Chapter 3.

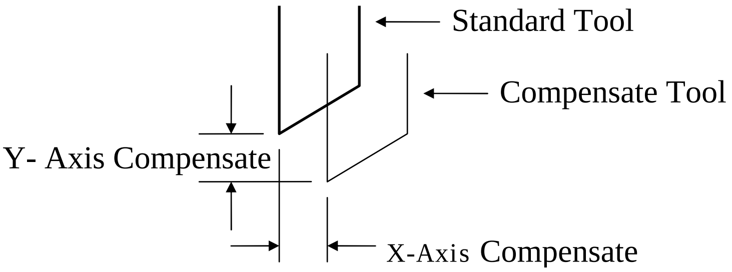


Fig 4-1 Tool Offset Compensation

<u>Offset Compensation Function</u>, G43, G49

Format: G43 P__ (Offset compensation in effect)
G49 (Offset compensation cancellation)
P__: 1~8, Tool number as shown in MCM table (#13~#36) in Chap 7.
P1 as 1st tool group, P2 as 2nd tool group, ... etc.

When G43 is in effect, the compensation data for the designated group which includes X and Y axis will be automatically added to (or subtract from) the program coordinate. The actual tool movement is accordingly adjusted.

Actual tool movement = Program coordinate + (or -) Offset

compensation
When tool offset compensation is no longer required, use G49 function to cancel it.

```
Ex:   N00 G43 P1       ....   1st tool offset compensation (MCM #13~#14) in effect
      N10 X50.000 Y100.000
      N20 Y200.000
      N30 X100.000 Y250.000
      N40 G49          ....   Tool offset compensation canceled
```

When using functions G43 and G49, the following rules must be observed:

1. Tool offset compensation is automatically canceled when power-on.
2. G43 is a modal code. It remains in effect when G43 is encountered in the program. You have to use G49 to turn it off.
3. When use G43 without indicating 'P" number or with P0, the CNC controller will use P1, the first tool offset compensation data.

4.2 Tool Offset Data Input and Revision

The tool offset data stored in MCM parameter #13~#36 can be revised by one the following methods:

1. Direct input/edit in MCM mode. (See Chap 7)
2. Using G10 method as described in Chap 3. (Sec 3.8.2)
3. Incorporated into PLC program. (See HUST H2N Connection Manual)

5 KEYBOARD AND LCD DISPLAY

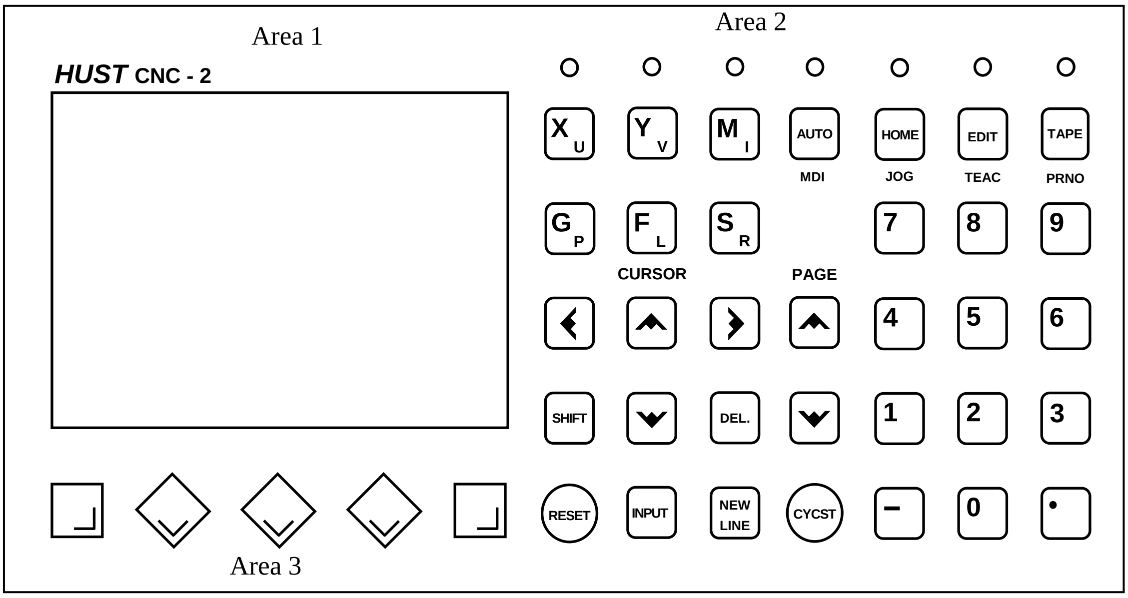


Fig 5.1 HUST H2N Keyboard and LCD Display

HUST H2N keyboard is shown in Fig 5.1 and can be roughly divided into three areas. They are to be discussed in the following sections.

Area 1: LCD display for coordinate, program, MCM parameter, operation status, etc.
Area 2: Execution mode, editing function key and numerical key area.
Area 3: Special function key area.

HUST H2N controller is equipped with LCD screen for display. If necessary, customers can design their own screen display. This can be accomplished by using DIY screen editing software developed by HUST (See Appendix B), DNC10 software (See Chap 9) and processed through PLC program. (See H2N Connection Manual)

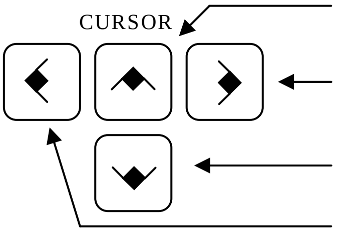
In addition to the standard function, the keyboard can be customized and processed through PLC program to suit customer's special requirement.

Please refer to Chap 6 for program editing and refer to Chap 8 for program execution, manual operation and manual data input.

5.1 Keyboard Description

When there are two words on a key, press the key once to access the word on the top left corner of the key. Press the same key twice within 0.5 seconds to access the word on the lower right corner or the function below the key. The red light above the key will blink if the function beneath the key is accessed. Followings are description for keyboard keys.

X U **Y** V **M** I **G** P **F** L **S** R	These keys are for function code (letter) input.
AUTO MDI	AUTO mode for program execution. MDI Mode for manual data input operation. (Press key twice in 0.5 sec)
HOME JOG	Return the machine tool to HOME position. Put the machine in manual JOG mode. (Press key twice in 0.5 sec)
EDIT TEAC	Edit mode for program editing. Edit program in TEACH mode. (Press key twice in 0.5 sec)
TAPE PRNO	Download or upload program (or data) between PC and the controller. Select a program for editing or execution. (Press key twice in 0.5 sec)
	This blank key is for I/O test display. Press this key twice in 0.5 sec to display MCM parameters.
– • 0 ~ 9	Numeric keys including negative sign and decimal point for data input.



Move cursor up one data item.

Move cursor or data display to the right.

Move cursor down one data item or one line.

Move cursor or data display to the left.

PAGE ▲	Move cursor one line (or one page) up.
▼	Move cursor one line (or one page) down.
SHIFT	Dummy key if press once. Press twice in 0.5 sec to display variable symbol “#”.
DEL	Delete data input during program edit.

RESET	Reset the CNC controller to power-on conditions.
INPUT	Enter data during programming or MCM data editing.
NEW LINE	Insert a program block during programming.
CYCST	Press this key to START program execution.
☐ ◇ ◇ ◇ ☐	No function has been assigned yet.

5.2 Description of LCD Display

HUST H2N controller provides eleven (11) different display modes, including the power-on display. Only one mode can be displayed at a time.

5.2.1 Power-on Display

When you turn on the power or press the RESET key, the controller will display HUST H2N model as shown in Fig 5.2. This controller has an internal PLC. In addition the controller will reset the MCM parameters and some G-codes to their default values.

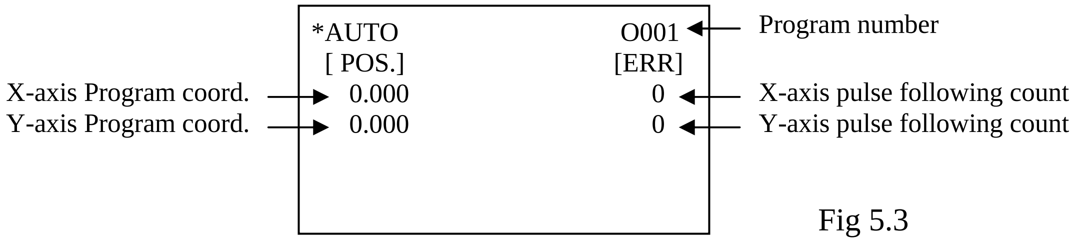
```
HUST
H02B-02
CONTROLLER
```

Fig 5.2

5.2.2 Coordinate Display

Program Coordinate Display

Press AUTO key once to get in AUTO mode and the LCD displays the current program coordinates as shown in Fig 5.3. When in AUTO mode, you can execute the program by pressing CYCST key. (See Sec 8.3)

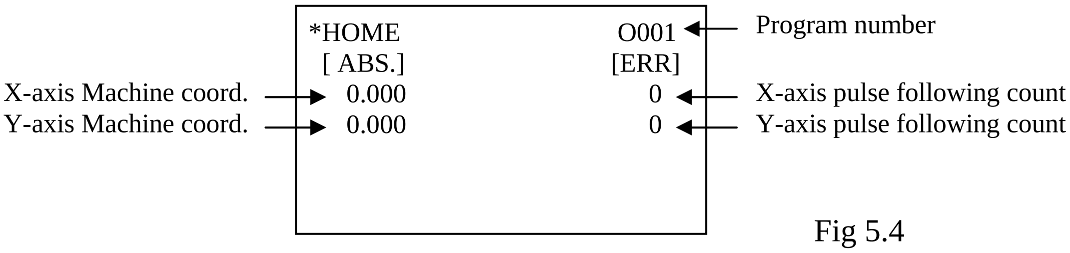


```
*AUTO              O001
 [ POS.]          [ERR]
   0.000              0
   0.000              0
```

Fig 5.3

HOME Mode (Machine Coordinate Display)

Press HOME key once to display the current machine coordinates of the tool as shown in Fig 5.4. When in HOME mode, press CYCST to execute HOME process for machine tool. (See Sec 8.1.1)

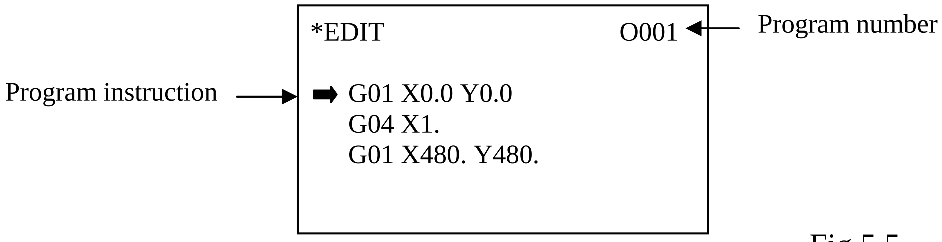


```
*HOME                 O001
 [ ABS.]              [ERR]
   0.000                 0
   0.000                 0
```

Fig 5.4

5.2.3 EDIT Mode

Press EDIT key once to get program EDIT mode with LCD display as in Fig 5.5. Program editing is detailed in Chapter 6. When in EDIT mode, use CURSOR →, CURSOR ← key to view the next data in the same block and use PAGE↑, PAGE↓ to view program block one above (or below).

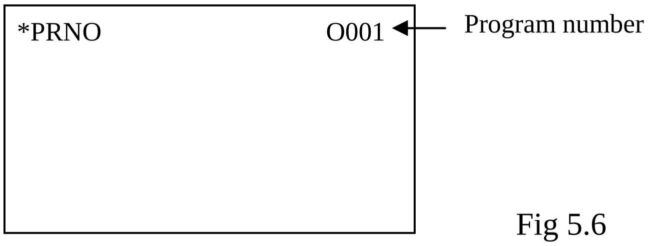


```
*EDIT                 O001

➡ G01 X0.0 Y0.0
  G04 X1.
  G01 X480. Y480.
```

Fig 5.5

5.2.4 Program Number (PRNO) Display Mode

Press TAPE key twice in 0.5 seconds to display the current program number (PRNO) as shown in Fig 5.6. Use CURSOR↑, CURSOR↓ key to see other programs in the memory. To select the desired program for execution or editing, press the "G" key twice in 0.5 seconds to get "P" then followed by program number and press INPUT key. (See Sec 6.1)



```
*PRNO                 O001
```

Fig 5.6

5.2.5 RS232 Data Upload/Download Mode

Press TAPE keys to get in RS232 mode as shown in Fig 5.7.

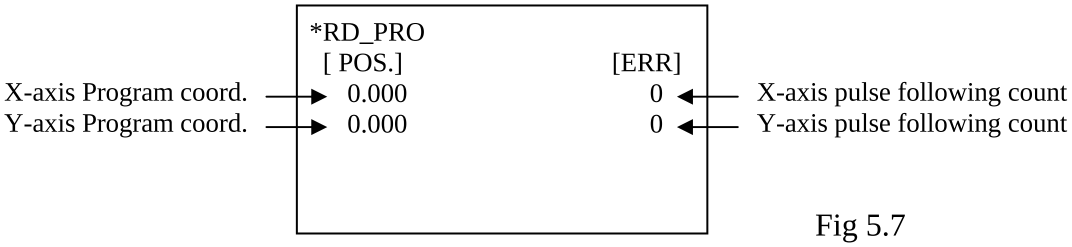


Fig 5.7

Use CURSOR↑, CURSOR↓ key to select upload/download mode you desire as below. When the desired operation is shown on screen, press CYCST to start operation. (See Chap 9)

1.	RD_PRO	Download program from PC to the controller.
2.	PUNCH_P	Upload the current program from the controller to PC.
3.	PUNCH_A	Upload all programs from the controller to PC.
4.	RD_MCM	Download MCM data from PC to the controller.
5.	PUNCH_C	Upload MCM data from the controller to PC.
6.	RD_LAD	Download PLC program from PC to the controller for PLC Simulation.
7.	RD_TBL	Download the customized LCD display data from PC to controller.
8.	RD_SYS	Download controller system data from PC to controller.
9.	DNC_EXE	Download and execute the program simultaneously from PC to the controller.

5.2.6 JOG Mode

Press HOME key twice in 0.5 seconds to get into JOG mode and the LCD displays as shown in Fig 5.8. (See Sec 8.1.2)

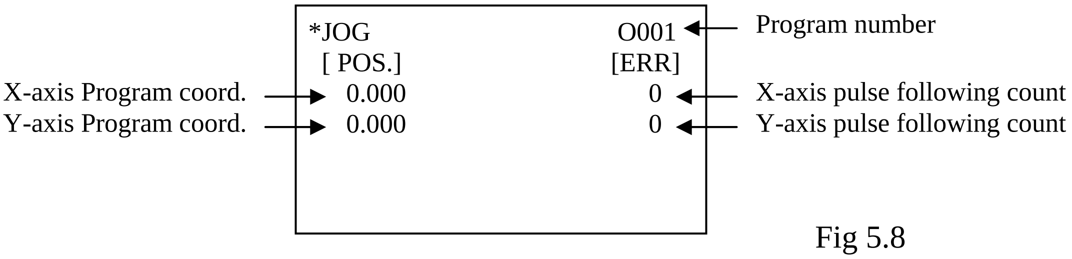
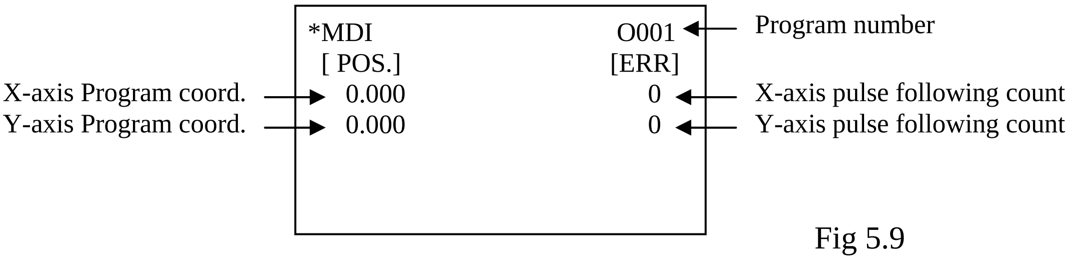


Fig 5.8

When in JOG mode, use CURSOR↑, CURSOR↓ key to select the axis. Then press PAGE↑ key to execute JOG operation, press PAGE↓ will move the tool in other direction.

5.2.7 MDI Mode

Press AUTO key twice in 0.5 seconds to get into MDI mode and the LCD displays as shown in Fig 5.9. (See Sec 8.2)

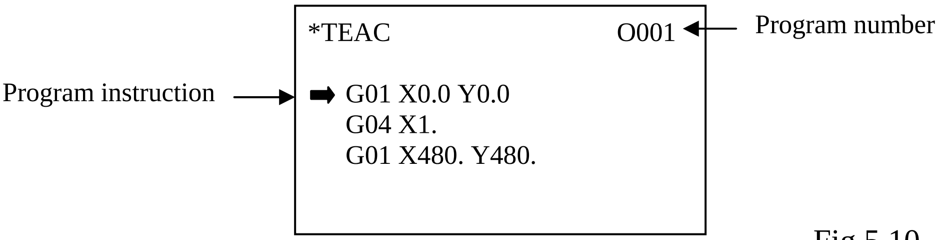


```
*MDI                O001
 [ POS.]           [ERR]
   0.000               0
   0.000               0
```

Fig 5.9

5.2.8 TEACH Mode

Press EDIT key twice in 0.5 seconds to get program TEACH mode and the LCD displays as in Fig 5.10. (See Sec 6.4)

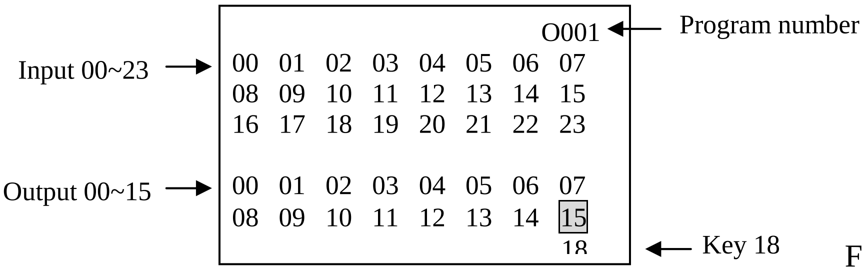


```
*TEAC               O001

➡ G01 X0.0 Y0.0
  G04 X1.
  G01 X480. Y480.
```

Fig 5.10

5.2.9 I/O Test and Key Mode

Press the blank key once to get I/O test and key mode as shown in Fig 5.11.

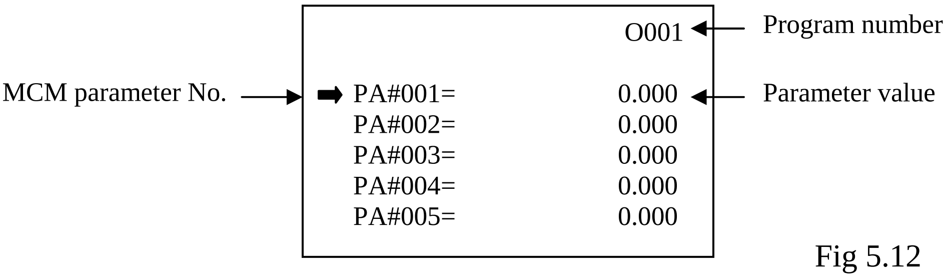


```
                          O001
00  01  02  03  04  05  06  07
08  09  10  11  12  13  14  15
16  17  18  19  20  21  22  23

00  01  02  03  04  05  06  07
08  09  10  11  12  13  14  15
                            18
```

Fig 5.11

5.2.10 MCM Parameter Mode

Press the blank key twice in 0.5 seconds to get MCM mode. The LCD screen displays the first page of MCM data (#1~#5) as shown in Fig 5.12.



```
                     O001

➡ PA#001=           0.000
  PA#002=           0.000
  PA#003=           0.000
  PA#004=           0.000
  PA#005=           0.000
```

Fig 5.12

When in MCM mode, use CURSOR↑, CURSOR↓ key to move the cursor to the desired parameter. To revise the value, simply key in the new data at the desired location and press INPUT key. Use PAGE↑, PAGE↓ to move 5 parameters at a time. More on MCM parameters in Chap 7.

6 PROGRAM EDITING

The following topics will be discussed in this chapter.

1. Select a program for editing.
2. Edit a new program.
3. Revise an existing program.
4. Edit a program in TEACH mode.

6.1 Program Selection

HUST H2N 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.

1. Press TAPE key twice in 0.5 seconds to be in PRNO mode and the LCD display is shown as Fig 6.1

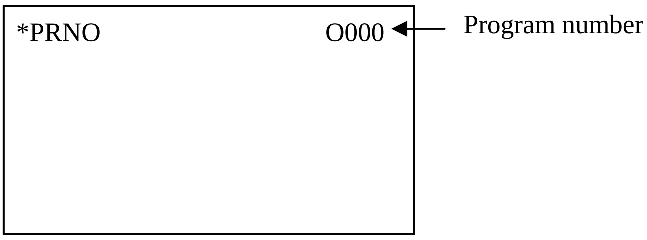


Fig 6.1

2A. There are two ways to select the program number for editing. The 1st method is to press letter "G" twice to get "P" - Program number (1 for example), then press INPUT key to complete the program selection. Program #1 has been selected in Fig 6.2.

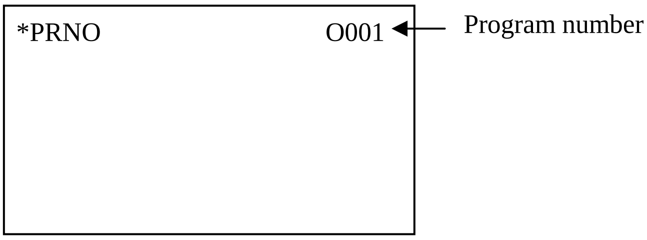


Fig 6.2

If the program selected is not in the memory, it is a new program and the entire screen is a blank except the top PRNO line. If the program selected already exists, the first 4 lines of program will be shown in the LCD screen. In either case, press EDIT key to start editing or revising the program.

2B. The 2nd method is, after step 1 above, to use CURSOR↑, CURSOR↓ to step through the programs in the memory. When the desired number shows up on the LCD screen as shown in Fig 6.2, simply press EDIT key. The controller is now ready for program editing. Note that the program number is NOT necessary in sequence.

6.2 New Program Editing

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 6.3.

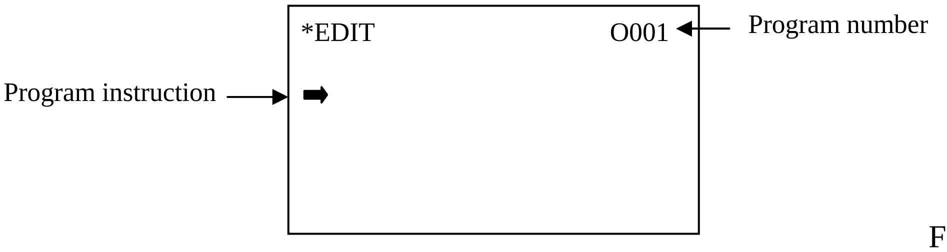


Fig 6.3

During program editing, the following keys will be used.

1. All the function keys and numeric keys on the keypad as shown in Fig 5.1.
2. CURSOR ← and CURSOR → keys for data inspection in the same block.
3. PAGE↑ and PAGE↓ keys for data inspection between lines.
4. NEW LINE key -- Establishing or inserting a new block anywhere in the program. Key in a function code (G01 or X10), then press NEW LINE to establish a new line.
5. INPUT -- For entering a data or a function in the established block. Key in a function code (Y10.0 or F100, etc.), then use INPUT to enter more data into the established line.
6. DEL -- For deleting a block (line) of program.

<u>Auto-generation of Block Number</u> (Auto-N)

You can edit a program with or without block number. If you do not intend to use block number, please set the MCM #94 = 0 (See Chap 7). Otherwise, every time you press NEW LINE key, the block number will be automatically generated according to the values of MCM #93 and #94. Following is an **<u>example program</u>** to explain the keystrokes required to edit a new program in the controller.

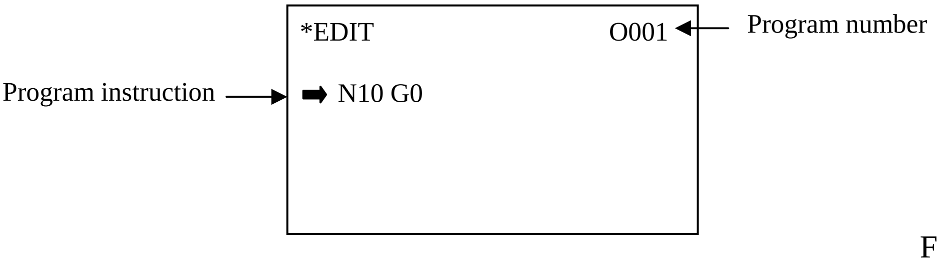
Ex: Program O001
N10 G0 X0. Y0.
N20 G4 X1.
N30 G0 U480. V-80.
N40 G4 X1.
N50 M99

Assume that program 1 is selected according to the steps in Sec. 6.1. MCM #93=10 and MCM #94=10. (See Chap 7) Make sure the controller is in EDIT mode.

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

1. N10 G0 X0. Y0.
   (A) G – 0 – NEW LINE

Use NEW LINE key here to establish a block. N10 is generated automatically because MCM #93=10. After this step, the LCD screen is shown as Fig 6.4.



```
*EDIT                    O001
➡ N10 G0
```

Fig 6.4

(B) X – 0 – 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. V-80.
   (A) G - 0 - NEW LINE
   (B) U - 4 - 8 - 0 - • - INPUT
   V - "–" - 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.

6.3 Program Revision

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

Revise or Add a Function

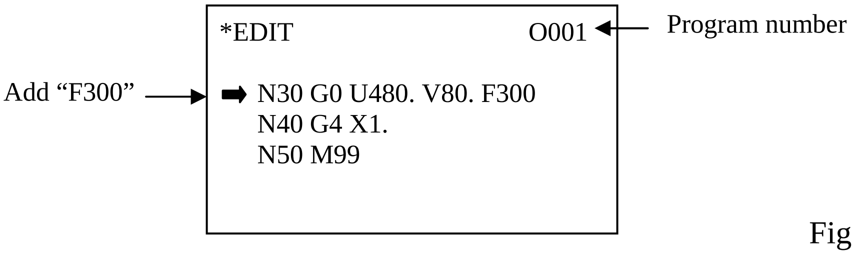
To revise or add a function, simply key in the function code and the correct number, then press INPUT key.

Ex: Revise N30 U480. V-80.
To N30 U480. V-80. F300

1. Make sure the system in EDIT mode.
2. Use PAGE↑, PAGE↓ key to move cursor to N30 block.

3. Add a function of F300. by entering data below and LCD will display as in Fig 96.5.

F - 3 - 0 - 0 - INPUT



```
*EDIT                O0001

➡ N30 G0 U480. V80. F300
  N40 G4 X1.
  N50 M99
```

Fig 6.5

4. Revise U480. to U360. by keying in

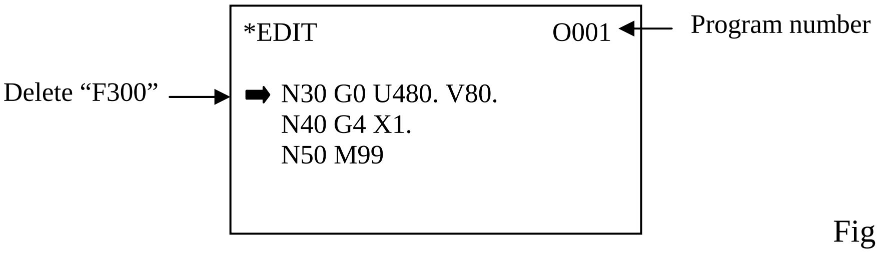
U - 3 – 6 - 0 - • - INPUT

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-80. F300
To N30 U480. V-80.

1. Make sure the system in EDIT mode.
2. Use PAGE↑, PAGE↓ key to move cursor to N30 block.
3. Key "F" without numbers and press INPUT key, LCD displays as Fig 6.6.



```
*EDIT                O0001

➡ N30 G0 U480. V80.
  N40 G4 X1.
  N50 M99
```

Fig 6.6

Insert a Program Block

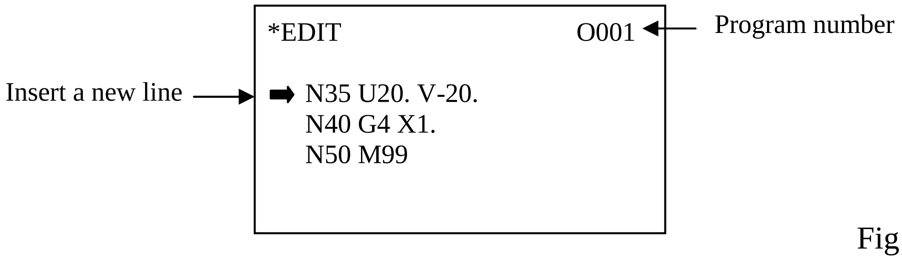
To insert a program block, key in the block number (or any function) and use NEW LINE key to establish the block. Then use INPUT key to input the rest of data for the block.

Ex: Insert N35 U20. V-20.
between N30 U480. V-80. and
N40 G4 X1.

1. Make sure the system in EDIT mode.

2. Use PAGE↑, PAGE↓ key to move cursor to N30 block.
3. Enter data as below and LCD display is as Fig 6.7. Note that the letter “N35” can not be keyed in from keyboard. Depending on the values of MCM #93 and #94, the actual block number may be different.

U - 2 - 0 - • - NEW LINE
V - "-" - 2 - 0 - • - INPUT



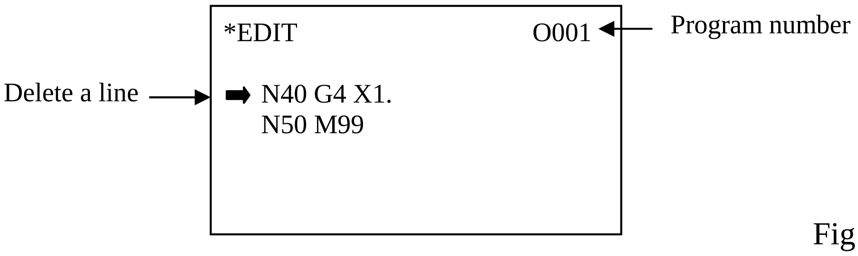
```
*EDIT                O001
➡ N35 U20. V-20.
  N40 G4 X1.
  N50 M99
```

Fig 6.7

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 N35 U20. V-20. from last example.

1. Make sure the system in EDIT mode.
2. Use PAGE↑, PAGE↓ key to move cursor to N35 block.
3. Press DEL key and The LCD displays as shown in Fig 6.8. (Block N40)



```
*EDIT                O001
➡ N40 G4 X1.
  N50 M99
```

Fig 6.8

Delete a Program

To delete the current program being edited, go to the procedures below. To delete one of the programs in memory, select the program using PRNO mode described in Sec 6.1 then follow the procedures below.

1. Get into MDI mode by pressing AUTO key twice in 0.5 seconds.
2. Key in function code G10 P2000, then press CYCST key.

Delete All Programs in the memory

To delete all programs (0~999) in the memory, use the following steps.

1. Get into MDI mode by pressing AUTO key twice in 0.5 seconds.
2. Key in function code G10 P2001, then press CYCST key.

Once you have executed this operation, all programs will be cleared and nothing will be left in the memory. Be careful before you do this.

6.4 Program Edit by TEACH mode

Occasionally during program editing, it's difficult to obtain the X or Y coordinate. One easy way to solve this problem is to use the TEACH function in HUST H2N controller. When the system is in TEACH mode, you can use MPG hand-wheel to move the tool to the desired location. Then press INPUT key to transfer the coordinates to the program. TEACH function is the same as EDIT except that you use MPG hand-wheel to find the coordinates in TEACH mode. Therefore, all the keys used in EDIT mode as discussed in last section are also used for editing program in TEACH mode.

When use TEACH function for a large and long work-piece, it's more convenient to make a hand-carry type TEACH box that contains a MPG hand-wheel, NEW LINE, DEL, and INPUT keys. This can be done by using the appropriate C-bits in PLC program. (Please refer to Chapter 6 of HUST H2N Connecting Manual)

Note that every time the INPUT key is pressed, the current tool coordinate will be transferred into the program when in TEACH mode. If TEACH function will be required for part of your program, it’s advisable to do your entire program in TEACH mode to avoid confusions or mistakes. Followings are steps to edit (or revise) a program in TEACH mode.

1. Press EDIT key twice in 0.5 seconds to get in TEACH mode.
2. Key in function code (such as G00, G01 etc.) then press NEW LINE key to establish a new block.
3. Use CURSOR↑, CURSOR↓ key to select X-axis for input. Use MPG hand-wheel to move tool to the desired X-axis location. Then press INPUT key. Repeat this step for Y-axis if desired. Use PAGE↓ to display the current tool coordinate on LCD screen.
4. Repeat Steps 2~3 to complete the whole program. Finish the program with M02, M30 or M99 function.

6.5 Rules for Numerical Input

Numerical input takes two formats, i.e. integer and decimal with a maximum of 7 digits (the 8th digit for "-" sign). 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.

The standard numerical format for HUST H2N controller is 4/3 (7 digits with 3 decimal points) with optional 3/4, 5/2 or 6/1 format which requires special settings of C-bits in the PLC. If you accidentally enter an integer for a decimal format, the controller will place a decimal point at the location corresponding to the recognized format. The feed-rate F-value does not recognize decimals. The table below shows the decimal numbers recognized by the controller after internal process for some integer inputs.

Input	3/4 Format	4/3 Format	5/2 Format	6/1 Format
X2	X0.0002 mm	X0.002 mm	X0.02 mm	X0.2 mm
Y250	Y0.0250 mm	Y0.250 mm	Y2.50 mm	Y25.0 mm
U2500	U0.2500 mm	U2.500 mm	U25.00 mm	U250.0 mm
V25.	V25.0000 mm	V25.000 mm	V25.00 mm	V25.0 mm
F300	F300 mm/min	F300 mm/min	F300 mm/min	F300 mm/min

The numerical formats for the function codes used in HUST H2N 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, G65 Variable	Integer input
X, Y, U, V, I, J-code	Decimal input
F-code	Integer input

6.6 Notes on Program Edit

Program Block Number

1. Block number can be omitted, but it’s better to have it for the convenience of program inspection later.

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

3. The ranking of the block number has nothing to do with the program execution sequence. The program is executed in a top-down fashion.

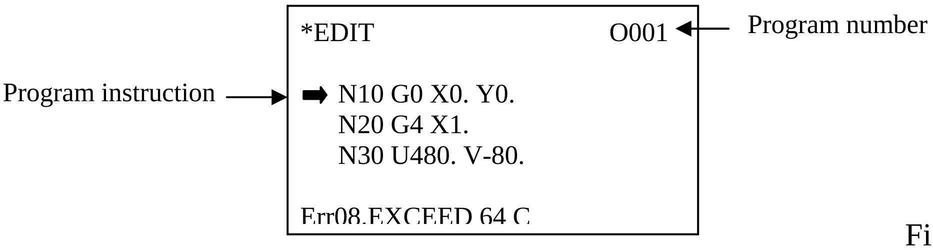
Program Block

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, U, V, I, 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.

4. Do not exceed 64 bytes of data input for a single block. Otherwise, the CNC controller will show an error message at the bottom of screen as in Fig 6.9



```
*EDIT                    O001

➡ N10 G0 X0. Y0.
  N20 G4 X1.
  N30 U480. V-80.

Err08.EXCEED 64 C
```

Fig 6.9

7 MCM (Machine Constant) PARAMETERS

7.1 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 is very important for smooth operation of equipment. Once they have been set, press RESET key to restart the system.

Read and Revise MCM Parameters

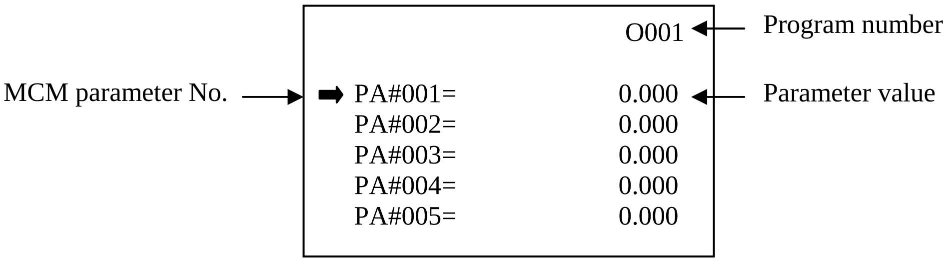
HUST H2N provides two ways for MCM parameter input or revision.

- Direct Input from Keyboard

1. Press RESET. Press the blank key twice in 0.5 seconds to get in MCM mode. The data of MCM parameter #1~#5 shows up on the LCD screen as shown in Fig 7.1.

2. Use PAGE↑ or PAGE↓ key to move MCM parameter 5 items at a time.

3. USE CURSOR↑ or CURSOR↓ key to move cursor to the desired parameter. Key in the correct value, then press INPUT key to complete revision.



```
                          O001

➡ PA#001=                0.000
  PA#002=                0.000
  PA#003=                0.000
  PA#004=                0.000
  PA#005=                0.000
```

Fig 7.1

- Revise on PC then Download to Controller through RS232

  The software “DNC10.EXE” is required for download/upload operation through RS232 interface. The TAPE mode on controller provides download/upload operation. Please see Chapter 9 for RS232 operation.

To Clear All Parameters to Factory Default Settings

1. Get into MDI mode by pressing AUTO key twice in 0.5 seconds.
2. Key in G10 P1000, then press CYCST key.

7.2 Description of MCM Machine Constants

The decimal format for MCM data in this section is based on 4/3 format.

MCM #1~#12 are for G54~G59 work coordinates data. The setting value is the distance between the origin of each work coordinate system and the machine HOME position. All input data have the same format and unit as shown below:

Format：□.□□□, Unit: mm (Default=0.000)

1. G54 (1st) Work Coordinate, X-axis.
2. G54 (1st) Work Coordinate, Y-axis.

3. G55 (2nd) Work Coordinate, X-axis.
4. G55 (2nd) Work Coordinate, Y-axis.

5. G56 (3rd) Work Coordinate, X-axis.
6. G56 (3rd) Work Coordinate, Y-axis.

7. G57 (4th) Work Coordinate, X-axis.
8. G57 (4th) Work Coordinate, Y-axis.

9. G58 (5th) Work Coordinate, X-axis.
10. G58 (5th) Work Coordinate, Y-axis.

11. G59 (6th) Work Coordinate, X-axis.
12. G59 (6th) Work Coordinate, Y-axis.

MCM parameters #13~#36 are for tool offset compensation data (in 8 groups). The setting value is equal to the offset with respect to the reference tool. (See Chap 4) The radius compensation R is not available at present for HUST H2N system. All input data have the same format and unit as shown below:

Format：□.□□□, Unit: mm (Default=0.000)

13. X-axis Offset Data, 1st Tool.
14. Y-axis Offset Data, 1st Tool.
15. R-Radius Compensation, Not Available.

16. X-axis Offset Data, 2nd Tool.
17. Y-axis Offset Data, 2nd Tool.
18. R-Radius Compensation, Not Available.

19. X-axis Offset Data, 3rd Tool.
20. Y-axis Offset Data, 3rd Tool.
21. R-Radius Compensation, Not Available.

22. X-axis Offset Data, 4th Tool.
23. Y-axis Offset Data, 4th Tool.
24. R-Radius Compensation, Not Available.

25. X-axis Offset Data, 5th Tool.
26. Y-axis Offset Data, 5th Tool.
27. R-Radius Compensation, Not Available.

28. X-axis Offset Data, 6th Tool.
29. Y-axis Offset Data, 6th Tool.
30. R-Radius Compensation, Not Available.

31. X-axis Offset Data, 7th Tool.
32. Y-axis Offset Data, 7th Tool.
33. R-Radius Compensation, Not Available.

34. X-axis Offset Data, 8th Tool.
35. Y-axis Offset Data, 8th Tool.
36. R-Radius Compensation, Not Available.

37. Master/Slave Mode Setting
Format：□.□□□, (Default=0)

Setting = 0, CNC mode, Master/Slave mode NOT set.
= 1, X-axis as master axis, Y-axis as slave axis.
= 2, Y-axis as master axis, X-axis as slave axis.

MCM parameters #38~#41 are for reference point data. The setting value is the location of each reference point with respect to the machine HOME position.

38. G28 Reference Point Data, X-axis.
39. G28 Reference Point Data, Y-axis.
Format：□.□□□, Unit: mm (Default=0.000)

40. G30 Reference Point Data, X-axis.
41. G30 Reference Point Data, Y-axis.
Format：□.□□□, Unit: mm (Default=0.000)

42. Backlash Compensation, X-axis.
43. Backlash Compensation, Y-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.

44. Jog Speed, X-axis and Power-on G01 Speed.
45. Jog Speed, Y-axis and Power-on G01 Speed.
Format：□.□□□, Unit: mm/min (Default=1000)

46. Denominator of Machine Resolution, X-axis.
47. Numerator of Machine Resolution, X-axis.
48. Denominator of Machine Resolution, Y-axis.
49. Numerator of Machine Resolution, Y-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 #60, #61.

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

Ex1: X-axis as linear axis (MCM #79=0), pitch = 5 mm = 5000 μm
Encoder = 2500 pulses, MCM #60 = 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 #46 (D) and #47 (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 #80=1), Angle = 360.000 deg/circle
Encoder = 2500 pulses, MCM #60 = 4, and GR = 5 (motor rotates 5 times while ball-screw rotates once)

Machine resolution = 360000/(2500 x 4)/5 = 360000/50000 = 36/5 =72/10

Therefore, the setting value for MCM #48 (D) and #49 (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 #67 and #69) should be in the range of –999999 ~ +999999. Otherwise, error message will appear. For example, if MCM #46=400 and #47=2, the machine resolution=2/400 =1/200 < 1/100. So, the value for MCM #67 should be less than 999999 and the value for #69 greater than –999999.

50. Home Direction for Tool, X-axis.
51. Home Direction for Tool, Y-axis.
Format：□, (Default=0)

Setting = 0, Tool returning to HOME in the positive direction.
Setting = 1, Tool returning to HOME in the negative direction

52. Home Speed When Tool Going to Home, X-axis.
53. Home Speed When Tool Going to Home, Y-axis.
Format：□□□□, Unit: mm/min (Default=2500)

54. Home Grid Speed When Tool Going to Home, X-axis.
55. Home Grid Speed When Tool Going to Home, Y-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 H2N 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 #52, #53 and the direction by MCM #50, #51. 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 #101, #102. 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 #54, #55 and the direction by MCM #101, #102. Once the encoder grid index is found, motor decelerates to a stop. This is the HOME position.

Note: Speed 2A & 3A for MCM #101, #102 = 0, Speed 2B & 3B for MCM #101, #102 =1.

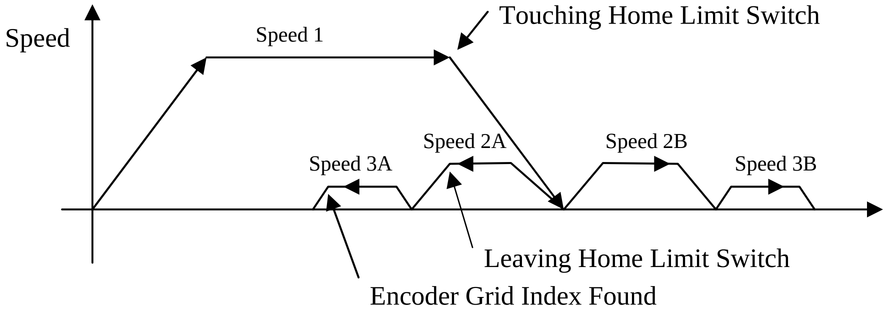


Fig 7.2 Tool's moving speed when searching for HOME location

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 #52, #53)
ACC = Time for acceleration/deceleration, in ms. (MCM #62)
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, and do both axes simultaneously if R232=3.

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

56. Traverse Speed Limit, X-axis (G00).
57. Traverse Speed Limit, Y-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:

Fmax = 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
Fmax = 0.95 * 3000 * 5 / 5 = 2850 mm/min
Therefore, it is recommended to set MCM #56=2850.

58. Direction of Motor Rotation, X-axis.
59. Direction of Motor Rotation, Y-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

IMPORTANT: 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.
(This statement has nothing to do with MCM #58, #59, but it's very important when connecting electrical motor.)

60. Encoder Multiplication Factor, X-axis.
61. Encoder Multiplication Factor, Y-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 #61 and the encoder resolution is 2000 pulses/rev, then the feed-back signals = 2000 * 2 = 4000 pulses/rev for Y-axis.

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

If MCM #89 setting = 0, type of accel./decel. for G00 = Linear.
If MCM #89 setting = 1, type of acceleration/deceleration for G00 = "S" curve. In this case, the actual acceleration/deceleration time is twice the setting value.

63. Servo Motor Acceleration/Deceleration Time (T), G01.
Format：□□□, Unit: millisecond (Default=100)
Setting Range: 4 ~ 3072 millisecond.
100 milliseconds is the recommended setting for both G00 and G01.

64. RS232C Baud Rate.
Format：□□□□, (Default = 9600)

Set RS232C communication speed. Choose from 9600, and 4800.
In addition, use the following settings for your PC:
Parity -- Even
Stop Bits -- 2 bits

65. 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 #66 (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 #65 = MCM #66 and you attempt to execute the program, error 03 will be the result. When this happens, you have to clear MCM #65 or reset MCM #66 to higher value.

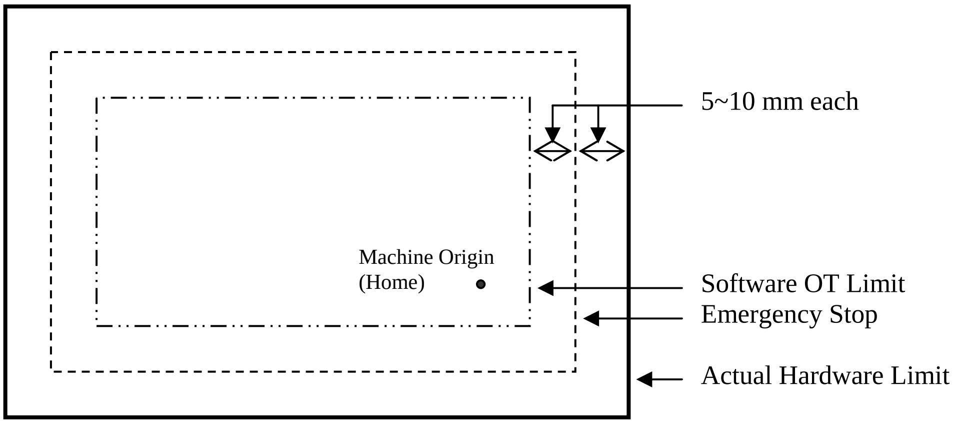
66. Counter Limit.
Format：□□□□□□□, (Default=0)
Maximum setting value = 9,999,999.

67. Software OT Limit in (+) Direction, X-axis
68. Software OT Limit in (+) Direction, Y-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).

69. Software OT Limit in (-) Direction, X-axis
70. Software OT Limit in (-) Direction, Y-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.



71. Home-Shift Data, X-axis
72. Home-Shift Data, Y-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.

73. Reserved
74. Reserved

75. Flag to Clear X-axis Program Coordinate on M02, M30 or M99 Command.
76. Flag to Clear Y-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.

77. Set Incremental/Absolute Mode, X-axis coordinate.
78. Set Incremental/Absolute Mode, Y-axis coordinate.
    Format：☐, (Default=1)

    Setting = 0, Coordinate is in incremental mode. U and V are NOT valid.
    Setting = 1, Coordinate is in absolute mode. U and V are valid.

79. Set X-axis as Rotating/Linear axis
80. Set Y-axis as Rotating/Linear axis
    Format：☐, (Default=1)

    Setting = 0, Use as linear axis.
    Setting = 1, Use as rotating axis.

81. Reserved

82. Set Position Gain, X-axis.
83. Set Position Gain, Y-axis.
    Forma：☐☐☐, (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.

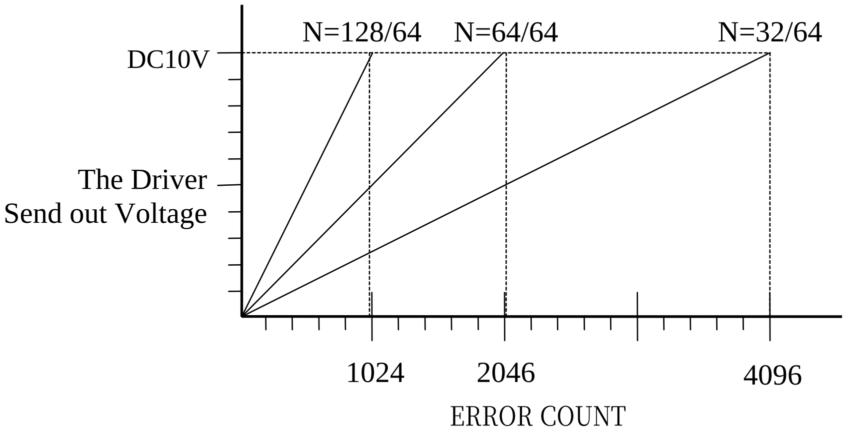


Fig 7-3 The relationship of Driver V-command and Servo Error

Recommended servo motor adjusting procedure:

1. Adjust the servo driver based on its operation manual.
2. Adjust the encoder multiplication factor (MCM #60~#61). 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 #60~#61 by lowering its multiplication factor.
3. Adjust the position gain of encoder feed-back. (MCM #82, #83)

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

$$\text{Position Gain} = \frac{(\text{Value of MCM\#82 or \#83})}{64}$$

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

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 #82~#83 and press RESET. If the problem still exists, please check the motor and the driver for proper connection.

84. Break-over Point (in Error Count) for Position Gain, X-axis.
85. Break-over Point (in Error Count) for Position Gain, Y-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 #84~#85, the position gain is 64. Otherwise, position gain will be calculated based on the setting value of MCM #82~ #83 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.

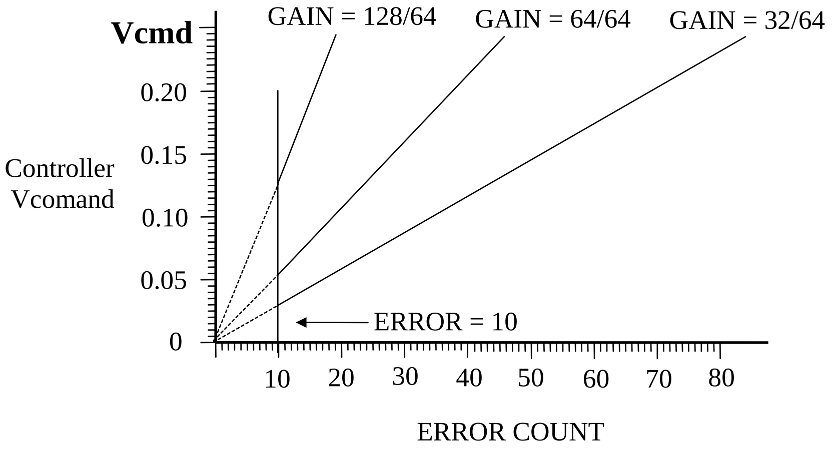


Fig 7.4 Break-over of Position Gain

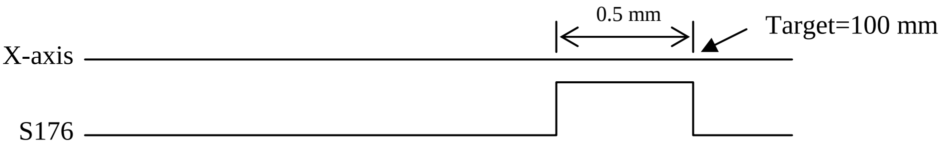
86. Flag to Save the Data of R000~R199 in PLC when power-off.
Forma：□, (Default=0)

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 #86.
Setting = 0, NOT to save.
Setting = 256, Save R000~R199 data.

87. In-Position Distance, X-axis.
88. In-Position Distance, Y-axis.
Format：□.□□□, Unit: mm (Default=0.000)

In-Position distance is the distance from the In-position to the target location. When cutting tool reaches the In-position, the controller sends a status signal S176~S177=1. When tool reaches the target location, S176~S177=0.

Ex: MCM #87 = 500 = 0.5 mm with function G01 X100.00, S176 = 1 when tool arrives at (100.0-0.5) = 99.5 mm and S176 = 0 when reaches target of 100.



89. 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".

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

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

Setting value is equal to the encoder spec times the factor from MCM #61.

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

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

94. 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 #93, #94.

Ex: MCM #93=0, #94=5, the generated block numbers are 0, 5, 10, 15, 20, ---.

95. Amount of Pitch Error Compensation per 100 mm of Lead-screw, X-axis.
96. Amount of Pitch Error Compensation per 100 mm of Lead-screw, Y-axis.
Format：□.□□□, Unit: mm (Default=0.000)

97. Denominator of Feed-rate Multiplication Factor for MPG Test.
98. Numerator of Feed-rate Multiplication Factor for MPG Test.
Format：□□□□, (Default=100)

The feed-rate of MPG test can be adjusted by the values of MCM #97 and #98. Maximum number of digit for these parameters is 5. Do not use zero (0). For example, if the setting value #97 = 100 and #98 = 200, the MPG test rate will be doubled.

99. Reserved
100. Reserved

101. Direction of Motor Rotation When Looking for Home Grid, X-axis.
102. Direction of Motor Rotation When Looking for Home Grid, Y-axis.
Format：D= □, (Default=0)

These parameters are to be used for setting direction of motor rotation when looking for home grid during HOME execution. (See MCM #54~#55, Speed2&3)

Setting = 0, The direction is opposite to the one for Speed 1. (Fig 7.2, Speed 2A, 3A)
Setting = 1, The direction is the same as that for Speed 1. (Fig 7.2, Speed 2B, 3B)

103. In-Position Count (Following Count)
Format：□, Unit: pulse (Default=0)

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, set a number for In-Position Count. When this function is in effect, the CNC execution will wait until the motor execution is within the number of pulses as set in MCM #103.

Setting = 0, In-position Count function is NOT valid.
Setting = 1,2,3 ... Any servo error count and the function is in effect.

104. Set Acceleration/Deceleration Time for MPG
Format：□□□, Unit: msec (Default=64)
Setting Range: 4 ~ 512 msec.

This is used to set the motor acceleration/deceleration time when MPG hand-wheel is used.

105.	X-axis Denominator,	MPG Hand-wheel Resolution Adjustment. (pulse)
106.	X-axis Numerator,	MPG Hand-wheel Resolution Adjustment. (μm)
107.	Y-axis Denominator,	MPG Hand-wheel Resolution Adjustment. (pulse)
108.	Y-axis Numerator,	MPG Hand-wheel Resolution Adjustment. (μm)

Format：□□□□, (Default = 100)
Unit: Denominator = pulses, Numerator = μm

Ex1: For X-axis, MCM #105 = 100 pulses, MCM #106 = 100 μm.
The resolution for X-axis = 100/100 = 1 μm/pulse.
If MPG hand-wheel moves 1 notch (=100 pulses), the feed length in X-axis = 100 x (100/100) = 100 μm = 0.1 mm.

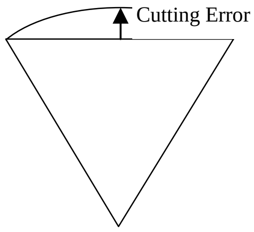
Ex2: For Y-axis, MCM #107 = 200 pulses, MCM #108 = 500 μm.
The resolution for X-axis = 100/100 = 1 μm/pulse.
If MPG hand-wheel moves 1 notch (=100 pulses), the feed length in X-axis = 100 x (500/200) = 250 μm = 0.25 mm.

109. METRIC/INCH Mode Selection (default = 0)
Format：□□□, (Default = 0)

Setting = 0, Measurement in METRIC unit.
Setting = 1, Measurement in INCH unit.

110. 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.



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.

Table 7-2 HUST H2N MCM Parameters

MCM No.	Setting Axis	Unit	Description	Setting
1	X	mm	G54 X-axis 1st Work coordinate (origin)	
2	Y	mm	G54 Y-axis 1st Work coordinate (origin)	
3	X	mm	G55 X-axis 2nd Work coordinate (origin)	
4	Y	mm	G55 Y-axis 2nd Work coordinate (origin)	
5	X	mm	G56 X-axis 3rd Work coordinate (origin)	
6	Y	mm	G56 Y-axis 3rd Work coordinate (origin)	
7	X	mm	G57 X-axis 4th Work coordinate (origin)	
8	Y	mm	G57 Y-axis 4th Work coordinate (origin)	
9	X	mm	G58 X-axis 5th Work coordinate (origin)	
10	Y	mm	G58 Y-axis 5th Work coordinate (origin)	
11	X	mm	G59 X-axis 6th Work coordinate (origin)	
12	Y	mm	G59 Y-axis 6th Work coordinate (origin)	
13	X	mm	X-axis, Tool #1 offset compensation	
14	Y	mm	Y-axis, Tool #1 offset compensation	
15	R	mm	Tool #1 radius compensation, NOT available	
16	X	mm	X-axis, Tool #2 offset compensation	
17	Y	mm	Y-axis, Tool #2 offset compensation	
18	R	mm	Tool #2 radius compensation, NOT available	
19	X	mm	X-axis, Tool #3 offset compensation	
20	Y	mm	Y-axis, Tool #3 offset compensation	
21	R	mm	Tool #3 radius compensation, NOT available	
22	X	mm	X-axis, Tool #4 offset compensation	
23	Y	mm	Y-axis, Tool #4 offset compensation	
24	R	mm	Tool #4 radius compensation, NOT available	
25	X	mm	X-axis, Tool #5 offset compensation	
26	Y	mm	Y-axis, Tool #5 offset compensation	
27	R	mm	Tool #5 radius compensation, NOT available	
28	X	mm	X-axis, Tool #6 offset compensation	
29	Y	mm	Y-axis, Tool #6 offset compensation	
30	R	mm	Tool #6 radius compensation, NOT available	
31	X	mm	X-axis, Tool #7 offset compensation	
32	Y	mm	Y-axis, Tool #7 offset compensation	
33	R	mm	Tool #7 radius compensation, NOT available	
34	X	mm	X-axis, Tool #8 offset compensation	
35	Y	mm	Y-axis, Tool #8 offset compensation	
36	R	mm	Tool #8 radius compensation, NOT available	
37			Master/Slave mode, 0=CNC, 1=X-axis, 2=Y-axis	
38	X	mm	X-axis, G28 reference point coordinate	
39	Y	mm	Y-axis, G28 reference point coordinate	
40	X	mm	X-axis, G30 reference point coordinate	
41	Y	mm	Y-axis, G30 reference point coordinate	
42	X	pulse	X-axis, Backlash compensation, 0~255	
43	Y	pulse	Y-axis, Backlash compensation, 0~255	
44	X	mm/min	X-axis, JOG Feed-rate & power-on G01 speed	
45	Y	mm/min	Y-axis, JOG Feed-rate & power-on G01 speed	
46	X	Pulse/rev	X-axis, Denominator, resolution calc. (Encoder pulse)	
47	X	μm	X-axis, Numerator, resolution calc. (Ball-screw pitch)	
48	Y	Pulse/rev	Y-axis, Denominator, resolution calc. (Encoder pulse)	
49	Y	μm	Y-axis, Numerator, resolution calc. (Ball-screw pitch)	
50	X		X-axis, HOME direction , 0=+ dir., 1=negative dir.	

MCM No.	Setting Axis	Unit	Description	Setting
51	Y		Y-axis, HOME direction , 0=+ dir., 1=negative dir.	
52	X	mm/min	X-axis, HOME speed 1	
53	Y	mm/min	Y-axis, HOME speed 1	
54	X	mm/min	X-axis, Home grid speed during HOME execution	
55	Y	mm/min	Y-axis, Home grid speed during HOME execution	
56	X	mm/min	X-axis, G00 Traverse speed limit	
57	Y	mm/min	Y-axis, G00 Traverse speed limit	
58	X		X-axis, Direction of motor rotation, 0=CW, 1=CCW	
59	Y		Y-axis, Direction of motor rotation, 0=CW, 1=CCW	
60	X		X-axis, Encoder pulse multiplication factor, 1, 2, or 4	
61	Y		Y-axis, Encoder pulse multiplication factor, 1, 2, or 4	
62		msec	G00 Linear accel./decel. Time, 4~512	
63		msec	G01 Linear accel./decel. Time, 4~3072	
64			RS232 Baud rate, 9600, 4800 / EVEN /2 Bit	
65			Current counter (M02, M30, M99)	
66			Counter limit (Max 9,999,999)	
67	X	mm	X-axis, Software OT limit, (+) direction	
68	Y	mm	Y-axis, Software OT limit, (+) direction	
69	X	mm	X-axis, Software OT limit, (-) direction	
70	Y	mm	Y-axis, Software OT limit, (-) direction	
71	X	mm	X-axis, HOME shift data	
72	Y	mm	Y-axis, HOME shift data	
73~74			Reserved	
75	X		X-axis, Cycle clearing w/ M02, M30, M99, 0=N, 1=Y	
76	Y		Y-axis, Cycle clearing w/ M02, M30, M99, 0=N, 1=Y	
77	X		X-axis, 0=incremental coord., 1=absolute coordinate	
78	Y		Y-axis, 0=incremental coord., 1=absolute coordinate	
79	X		Set X-axis as Rotating (1)/Linear axis (0)	
80	Y		Set Y-axis as Rotating (1)/Linear axis (0)	
81			Reserved	
82	X	Pulse	X-axis, Position gain, standard=64	
38	Y	Pulse	Y-axis, Position gain, standard=64	
84	X	Pulse	X-axis, Break-over point for position gain, std=10	
85	Y	Pulse	Y-axis, Break-over point for position gain, std=10	
86			Flag, R000~R199 to be saved when power off. 0=No, 256=Yes	
87	X	mm	X-axis, In-position distance	
88	Y	mm	Y-axis, In-position distance	
89			Motor Accel/Decel mode, 0=linear, 1=”S” curve	
90		msec	Accel/Decel time when in G99 mode (mm/rev)	
91		pulse/rev	Spindle encoder resolution (pulse per revolution)	
92		rpm	Max. spindle rpm @ 10 volts	
93	S		Start number for program block number generation	
94	D		Increment for program block number generation	
95	X	mm	Amount of Pitch Error Compensation per 100 mm of Ball-screw, X-axis.	
96	Y	mm	Amount of Pitch Error Compensation per 100 mm of Ball-screw, Y-axis.	
97			Denominator of feed-rate when in MPG test mode	
98			Numerator of feed-rate when in MPG test mode	
99~100			Reserved	

MCM No.	Setting Axis	Unit	Description	Setting
101	X	0/1	X-axis, Home grid direction during HOME execution	
102	Y	0/1	Y-axis, Home grid direction during HOME execution	
103		pulse	Servo error when executing in-position function	
104		msec	Set Acceleration/Deceleration Time for MPG (4~512)	
105	X		X-axis, Denominator, MPG resolution calc.	
106	X		X-axis, Numerator, MPG resolution calc.	
107	Y		Y-axis, Denominator, MPG resolution calc.	
108	Y		Y-axis, Numerator, MPG resolution calc.	
109			0=Metric mode, 1=inch mode	
110			Error in Circular Cutting, ideal value=1	

8 MANUAL OPERATION

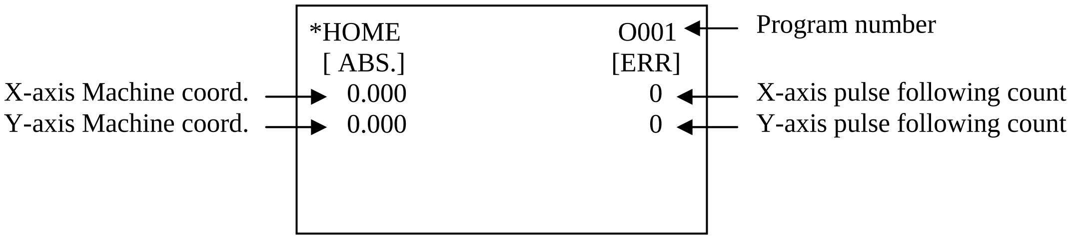
This Chapter is to discuss some functions that you can operate manually. When you become familiar with these operations, you'll feel ease to operate HUST CNC controller.

8.1 Manual operation

8.1.1 HOME Operation (Machine Origin)

1. Press RESET key to put the controller in a power-on status.
2. Press HOME key to enable the function. Notice the mode status display "HOME" on top left corner of LCD screen, as well as the current machine coordinate and pulse following count for both X and Y-axis.
3. Press CURSOR↑, CURSOR↓ to select the desired axis, X or Y. Notice the shift of red indicator light at the top of keyboard. Execute HOME one axis at a time.
4. Press CYCST to execute HOME process.

After HOME execution, the LCD screen will display either 0.0 or the home-shift value (MCM #71~#72) as shown in Fig 8-1.



```
*HOME                O001
 [ ABS.]             [ERR]
   0.000                0
   0.000                0
```

Fig 8-1 Coordinate Display for HOME (X-axis only)

1. Perform "HOME operation" after the controller is powered-on.
2. "HOME operation" is performed one axis at a time.
3. If the tool exceeds the Home-limit switch, move the tool manually inside the Home-limit switch through the application of JOG function.
4. The direction of tool travel for " HOME operation " is set in MCM #50~#51.
5. The homing speed is by MCM #52~#55 and the direction while looking for GRID is by MCM #101 and #102.

HUST H2N CNC has three (3) different speeds when execute HOME as shown in Fig 8.2.

Speed 1: The motor accelerates to Speed 1 and its maximum speed is determined by the settings of MCM #52, #53 and the direction by MCM #50, #51. 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 #101, #102. 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 #54, #55 and the direction by MCM #101, #102.

Once the encoder grid index is found, motor decelerates to a stop. This is the HOME position.

Note: Speed 2A & 3A for MCM #101, #102 = 0, Speed 2B & 3B for MCM #101, #102 =1.

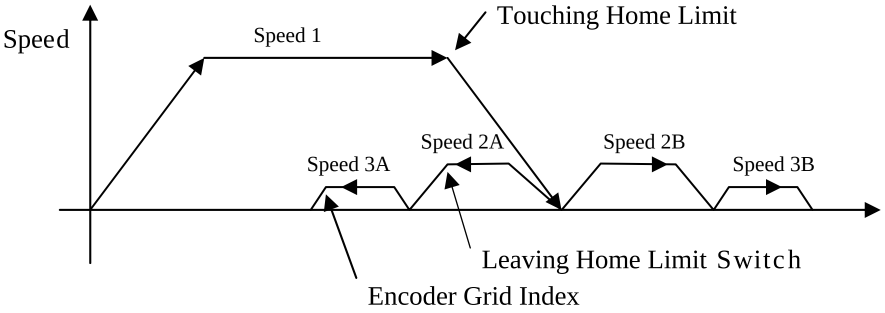


Fig 8.2 Tool's moving speed when searching for HOME location

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 #52, #53)
ACC = Time for acceleration/deceleration, in ms. (MCM #62)
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, and do both axes simultaneously if R232=3.

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

8.1.2 Manual JOG Feed Operation

There are two ways for manual JOG feed operation.

1. Use the external switch and the signal processed by PLC ladder.
2. Use a MPG hand-wheel.

Operation steps using HUST keyboard:

1. Press HOME key twice in 0.5 seconds to enable the JOG feed function.
2. Press CURSOR↑, CURSOR↓ to select the desired axis, X or Y.
3. Press PAGE↑ and keep pressing down to Jog-feed in (+) direction. Release key to stop Jog-feed. Press PAGE↓ and keep pressing down to Jog-feed in (-) direction.

Release key to stop Jog-feed. (HUST standard PLC) JOG-feed speed is determined by MCM #44, #45.

Operation steps using MPG hand-wheel:

On the back of H2N controller, there is a DB9-pin connector for connecting MPG hand-wheel.

1. Press HOME key twice in 0.5 seconds to enable the JOG feed function.
2. Press CURSOR↑, CURSOR↓ to select the desired axis, X or Y.
3. Turn MPG hand-wheel and the tool will move in the direction as indicated on the hand-wheel. When stop turning, motion stops also. If tool moves in the direction opposite to the hand-wheel, inter-change A- and B-connection in the MPG hand-wheel.

The tool's moving speed can be increased by a factor of 1, 10, or 100 which is processed by the internal PLC and the switch for 1, 10, 100 is externally installed.

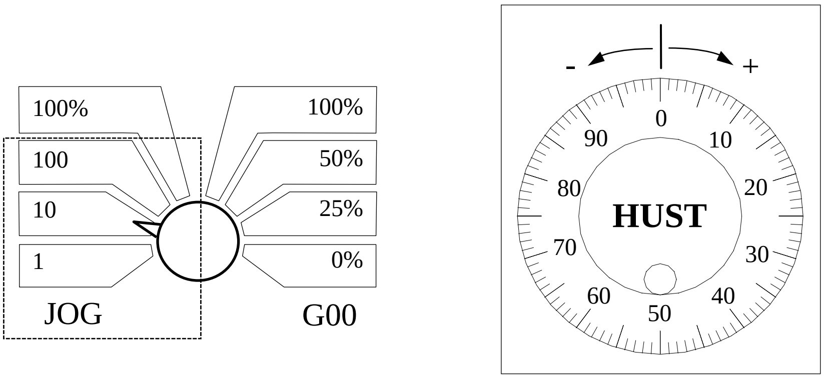


Fig 8-3 MPG Hand-wheel

8.1.3 G01 Manual Feed-rate Override (MFO %)

HUST H2N manual feed-rate override is processed by R221 in the PLC ladder with an external control knob (not provided) as shown in Fig 8-4. The range of control is 0~150%. (See Connection Manual for details) HUST PLC is normally set at 100% if you don't install an external control knob. During cutting operation (G01, G02, G03), you can turn this knob to control its speed anytime. For example, if the feed-rate = 100 mm/min and the MFO knob is pointing at 120%, the actual cutting rate is 120 mm/min.

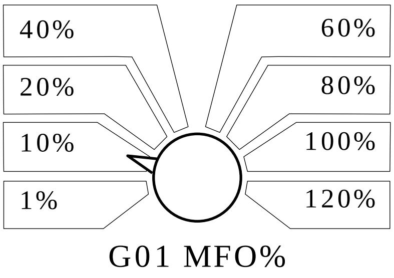


Fig 8-4 G01 Manual Feed-rate Override (MFO%)

8.1.4 G00 Manual Feed-rate Override

The adjustment knob is installed by the user as shown in Fig 8-5. The input signals are to be processed by the PLC Register R220 in ladder program. User can adjust the feed-rate any time during the machining process. The adjustment range is 0 ~ 100% and the standard PLC is fixed at 100%. Please see "Connection Manual" for details.

Ex: If G00 traverse speed has been set at 5000 mm/min and the control knob is pointing at 50%, the actual traverse speed is = 5000 * 50% = 2500 mm/min.

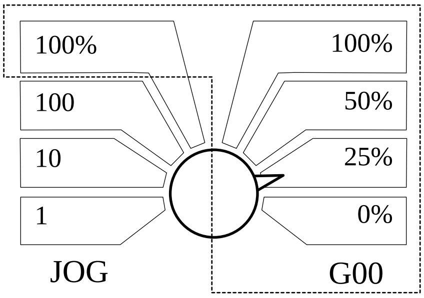


Fig 8-5 G00 Manual Feed-rate Override

8.2 MDI Single Block Operation, MDI

MDI function can be activated by pressing AUTO key twice in 0.5 seconds or it can be installed by user and processed through PLC. MDI function enables the user to enter a single block of program and execute it, or enter data into the controller by G10. (See Sec 3.8) Once the operation is finished, MDI command disappears. Followings are steps illustrating the application of MDI operation.

1. Press AUTO key twice in 0.5 seconds to enable MDI function.
2. Key in function to be executed
   Example: Set baud rate of 4800 for RS232.
   The function is “G10 P510 L4800”.
   Key in: G10 INPUT
   P510 INPUT
   L4800 INPUT
3. Press CYCST key to execute function.

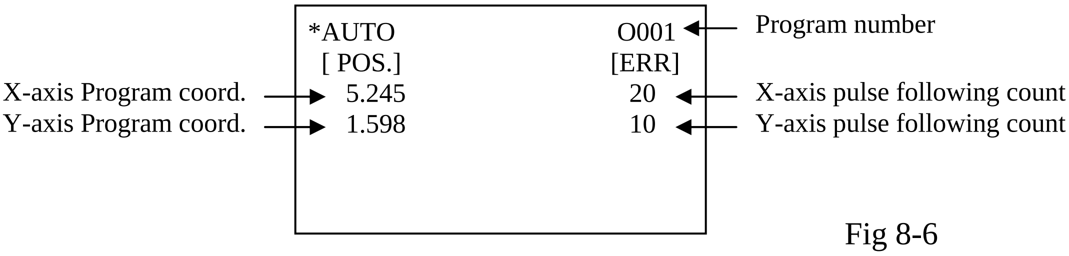
8.3 Auto Execution, AUTO

The "AUTO" function enables user to execute a program continuously until the end. While executing, the current X/Y-coordinate and the current following count will be displayed as shown in Fig 8-6. When executing this function, be sure that the tool is within the hardware and software over-travel limit and that there is no obstruction along the tool path.

Execution steps:

1. Use PRNO to select the desired program. (See Sec 6.1)

2. Press AUTO key to enable the function.
3. Press CYCST key to start execution. The whole program will be executed until the end.



```
*AUTO                O001
 [ POS.]             [ERR]
   5.245               20
   1.598               10
```

Fig 8-6

8.4 Single Block Execution in AUTO Mode, AUTO SINGLE

This function allows user to execute a program one block at a time until the end if desired. User has to install a key for this function and process it through PLC. (C006=1) Please see connection manual for details.

Execution steps:

1. Use PRNO to select the desired program. (See Sec 6.1)
2. Press AUTO SINGLE key to enable the function.
3. Press CYCST key to start the execution of the first block. When the execution stops, press CYCST again to execute the next block. Repeat the same procedure until the end.

8.5 Feed Hold (FEHOLD)

The FEHOLD key must be installed by users and processed through PLC. (C000=1) When FEHOLD key is pressed during AUTO mode operation, the program execution will be put on HOLD unless you cancel it. User can use this function to inspect program or work-piece in the middle of execution.

Execution steps:

1. The program is in AUTO execution mode.
2. Press FEHOLD key, the program execution will be put on HOLD.
3. Press CYCST key, the program execution will resume from the point where the program was stopped in step 2.

8.6 Option-Stop (OPST)

The Option-Stop key must be installed by users and processed through PLC. (C026=1) This function is valid only when M1 command code is present in the program. When OPST function is enabled, the program execution will stop at the block with M1 command. When the CYCST key is pressed again, the program execution will resume from the M1 block.

Execution steps:

1. Enter M1 code in the program where you want the program execution to stop.
2. Press OPST key to enable the option stop function.
3. Press AUTO and CYCST key to execute the program.
4. When the execution runs into M1 block, the execution stops.
5. Press CYCST key to resume program execution.

Note that if the OPST function is NOT enabled, the execution will ignore and skip the M1 block and continue executing the next block.

Example:

```
N10 G0 X20.0 Y20.0
N20 G1 V-20.0 F200.
N30 M1                 .....  If OPST function is enabled, the execution will stop at
                              this block. Press CYCST key will resume execution
                              from N40
N40 G1 X30.0 F300.
N50 X50.0 Y0.0
M60 M2
```

8.7 Skip Function, SKIP

The SKIP key must be installed by users and processed through PLC. (C027=1) When SKIP function is in effect, the program execution will ignore and skip the block containing the "/1" code and continue execution from the next block. Otherwise, the block with "/1" code will be executed as a normal block.

Execution steps:

1. Add the "/1" code in the program block to be skipped.
2. Press SKIP key to enable the skip function.
3. Press AUTO and CYCST key to execute the program.
4. The execution will skip the block containing the "/1" code and continue execution from the next block.

Example:

```
N10 G0 X20. Y0.
N20 G1 V-20. F200.
N30 X25. V-5. /1          .....  If the SKIP function is enabled, execution will skip
                                 this block and continue from the next block, N40.
N40 G1 X30. F300.
N50 G0 X50. Y0.
N60 M2
```

8.8 Program DRYRUN

The DRYRUN key must be installed by users and processed through PLC. (C015=1) DRYRUN function is used to test a program with G00 high speed and it can be activated anytime during program execution. If DRYRUN key is pressed during program execution, the controller will finish the current block with current feed-rate. After that, it will ignore all programmed feed-rates (F-values) and execute at G00 high speed until the program end. When DRYRUN function is canceled, the controller will return to the programmed F-speed at the end of the current block. Before activating this function, be sure that there is no any obstruction along the tool path in order to avoiding any potential damages to the equipment.

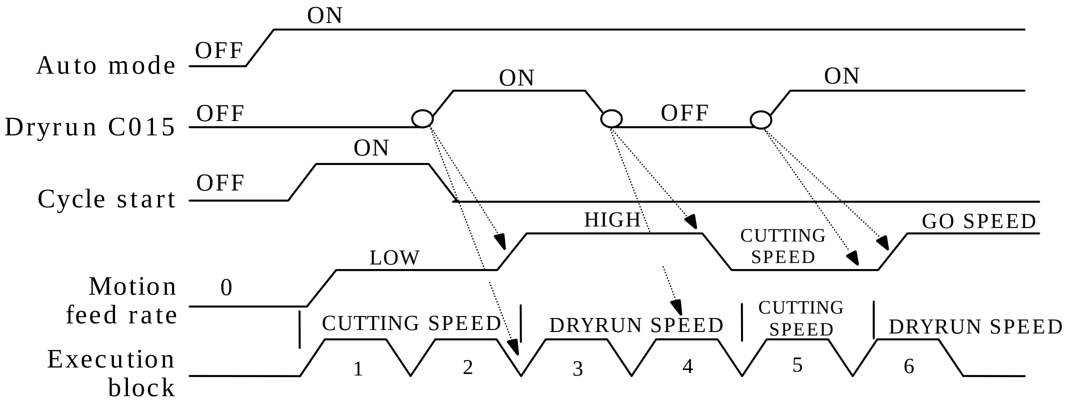


Fig 8-7 DRYRUN Function Timing Relation

8.9 MPG Hand-wheel Testing

In addition to the DRYRUN function for program testing, HUST H2N controller also provides another testing function by a MPG hand-wheel.

1. The advantage of the MPG testing is that user can do actual cutting using a MPG hand-wheel speed. Any errors in the program can be detected and the product inspected before making mass production.

2. MPG test must be done in AUTO mode, i.e. the Register R100=1 in PLC. The MPG Test key must be installed by users and processed through PLC, i.e. C056=1 in PLC to turn on the test function. During testing, the feed-rate by MPG hand-wheel is determined from the MFO% switch. Please refer to "Connecting Manual" for details.

MPG testing function steps:

1. Use PRNO function to select the program for MPG test.
2. Press MPG Test key (C056=1) to enable MPG testing function.
3. Select the feed-rate (MFO% switch)
4. Press AUTO and CYCST.
5. Rotate MPG to start testing. When you stop MPG, the testing also stops. You can control the test speed by controlling the rotating speed of MPG.

6. If you cancel the MPG testing mode, the controller will execute the program with the normal feed-rate.

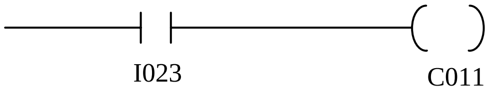
8.10 Program Re-start, RE-STA

- Program restart – Case 1

The program restart function allows the user to restart the execution from where the program was interrupted. User must know the exact location of program interruption when applying the RE-STA function. Its function key must be installed externally and processed by C011 bit in the PLC. When C011=1, RE-STA function is enabled.

Execution steps:

1. Press RESET key and use MPG (JOG function) to move the tool away from the work-piece. If the interruption is due to EM-stop or servo error (Error 2), execute "HOME" prior to pressing RESET.
2. Press RE-STA key (I023) to enable the restart function.
   For example if you use I023=1 in PLC, the bit C011=1 as shown below:



3. Press AUTO key.
4. Use Page↑, or Page↓ key to move the cursor to the block where the program was interrupted.
5. Press CYCST to start the RE-STA function. The restart function will be automatically cleared when program execution comes to M02 or M30 block.

Notes: During RE-STA execution, the M-, T-, S-code in the program before the interrupted block will be executed again.

Example: Program 2 (Fig 8-8)

Work origin at X=-150.0, Z=-250.0.
Execute "HOME" to move the tool to the machine origin.

```
N10 S200
N20 G0 X50. Y100.
N30 G1 V-20. F200.
N40 X60. V-20.       ..... program was interrupted at this block, restart from here.
N50 V-20.
N60 X80. V-20.
N70 G0 X250. Y150.
N80 M2
```

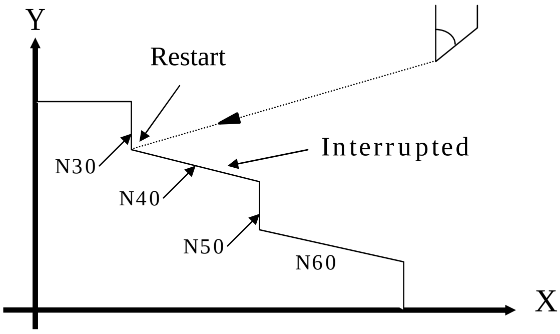


Fig 8-8 Program Restart (RE-STA)

Move cursor to N40 and activate the restart function. The controller will calculate the coordinate change from N10 to N30, then move the tool to the end of N30 and continue the program execution from there.

- Program restart – Case 2

Suppose that you are executing program “A”. For some reason you want to temporarily stop the current execution and switch to program “B”. After you have finished job with program “B”, you can use C012 and C013 in PLC to return to program “A” and continue execution from where it was stopped. The steps to achieve this operation are as follows.

1. Stop execution of program “A” and make C012=1 which would cause controller to store the block number where it was stopped.
2. Switch to program “B” and execute program.
3. When finished with program “B”, use PRNO key or process through PLC to return to program “A”.
4. Use C013=1 to cause controller to read the stored block number in step 1.
5. Use C011=1 to activate Re-start function.
6. Press CYCST to start execution from where program “A” was interrupted.

8.11 Round Corner Non-stop Operation

When executing two program blocks with the tool going in the different direction, the intersection normally forms a sharp angle and the motor will go through deceleration and acceleration. With this operating condition, some machine such as glue machine, flame or laser machine can not obtain a satisfactory result. To overcome this problem, HUST controller provides a round corner non-stop operation.

There is no function key available on the HUST keyboard. However, customer can use the input point to enable the bit C036=1 in the PLC. When C036=1, the round corner non-stop operation is enabled and the cutting goes through the arc SE as in Fig 8-9. The distance d (SP = PE) can be calculated from the equation below:

d = 0.5 * feed-rate, F * acceleration/deceleration time

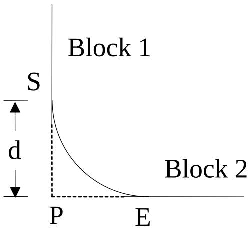


Fig 8-9 Round Corner Operation

Example: F = 500. mm/min. Time for acceleration/Deceleration = 300 ms
d = 0.5 * 500/60 * 300/1000 = 1.25 mm

9 PC ON-LINE OPERATION - RS232

Through TAPE function, HUST H2N can do the following PC (personal computer) on-line operations via RS232 interface. Through MDI mode, you can execute G10 function as shown in Table 9-1 as well as burn the transmitted program, MCM parameters, PLC simulation program, LCD screen display data, and system data into Flash-ROM in the controller.

1* Transfer part program from PC (personal computer) to CNC controller.
2. Transfer a part program from CNC controller to PC.
3. Transfer all part programs from CNC controller to PC in one try.
4* Transfer MCM data from PC to CNC controller.
5* Transfer MCM data from CNC controller to PC.
6* Transfer PLC ladder program from PC to CNC controller and test the ladder program.
7* Transfer LCD screen display data from PC to CNC controller.
8* Transfer controller system data from PC to CNC controller.
9. Transfer part program from PC to CNC controller and execute the program.

Items (with asterisk) 1, 4, 5, 6, 7, and 8 can be done from PC side when the controller is under power-on mode but NOT in TAPE mode.

Table 9-1 Online Operation with G10 Function through RS232 Interface

G10 P510 L4800	Set the baud rate for RS232 interface at 4800 bps
G10 P510 L9600	Set the baud rate for RS232 interface at 9600 bps
G10 P510 L19200	Set the baud rate for RS232 interface at 19200 bps
G10 P510 L38400	Set the baud rate for RS232 interface at 38400 bps
G10 P600 L01	Burn the externally transmitted program into Flash-ROM
G10 P600 L02	Burn the externally transmitted MCM parameters into Flash-ROM
G10 P600 L03	Burn the externally transmitted ladder program into Flash-ROM
G10 P600 L04	Burn the externally transmitted LCD screen display data into Flash-ROM
G10 P600 L05	Burn the externally transmitted controller system data into Flash-ROM
G10 P2100	Load the part program from Flash-ROM to memory

9.1 Program Transfer from PC to CNC Controller (RD_PRO)

Format for program transfer:

```
%
O001                    ..... Program number
N10 G0 X0. Y0.
N20 G1 X50. Y50.
N30 U30. V-30.                Program content
N40 G0 X0. Y0.
N50 M2
%
```

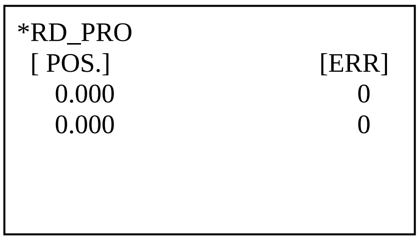
Notes:
1. The program must start and end with a symbol of “%”.
2. Program number range = 000~999

3. One line contains one program block only. Do NOT write multiple program block on the same line as “N10 G0. Y0. N20 G1 X10. Y10. N30 G0 X0. Y0. N40 M2”.

4. If program number (Oxxx) is NOT specified while transferring from PC to CNC, it’ll write over the current program in the controller. If the program number is specified, the program will write over the corresponding program in the memory.

Steps for Program Transfer to CNC:

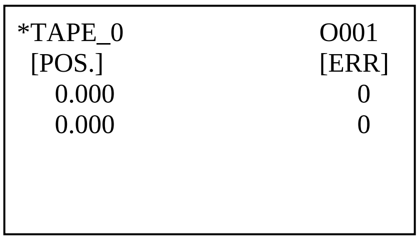
1. On PC side, execute "DNC10.exe" (see Sec 9.10) to bring up the main menu and do the necessary steps to get it ready for transmission. Make sure the communication protocol is correct. Select “Transmit Data” and do the followings in sequence.
   - Select “PC → HUST CNC”.
   - Select “PC send program data to CNC”.
   - Type in program name with directory path. Press ENTER.
2. When finish step 1, the following 3 cases may occur on CNC side.
   - If CNC is NOT in execution mode (S080=0) and NOT in TAPE mode, no additional step is necessary on the CNC side and the program will be downloaded to CNC immediately.
   - If CNC is in execution mode (S080=1), wait until CNC finish execution (S080=0). Put CNC in TAPE mode RD_PRO (see Fig 9-1) and press CYCST key to start transferring.
   - If CNC is already in TAPE mode, make sure it’s on RD_PRO page and then press CYCST key to start transferring.



```
*RD_PRO
 [ POS.]                [ERR]
    0.000                   0
    0.000                   0
```

Fig 9-1 TAPE function - RD_PRO

3. When the transferring is finished, LCD will display "TAPE_0" as in Fig 9-2. If you execute G10 P600 L01 command under MDI mode after step 3, the newly transmitted program will be burned into Flash-ROM.



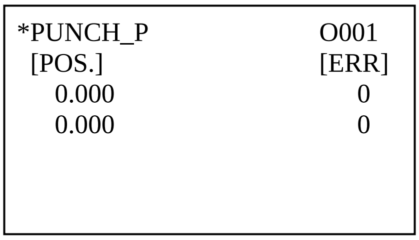
```
*TAPE_0                 O001
 [POS.]                 [ERR]
    0.000                   0
    0.000                   0
```

Fig 9-2 TAPE function - Program Transmission Completed

If the program on PC does NOT have a program number (as O002), it will write over the current program in CNC memory.

9.2 Program Transfer from CNC Controller to PC (PUNCH_P)

1. On PC side, execute "DNC10.exe" (See Sec 9.10) to bring up the main menu and do the necessary steps to get it ready for transmission. Make sure the communication protocol is correct. Select “Transmit Data” and do the followings in sequence.
    - Select “HUST CNC → PC”.
    - Select “CNC transmit program to PC”.
    - Type in the program name with directory path. Press ENTER.
2. On the CNC side, select the program number (PRNO) that is to be transferred.
3. Then, press RESET key.
4. Press TAPE key. Press Cursor↓ key to display PUNCH_P as in Fig 9-3.



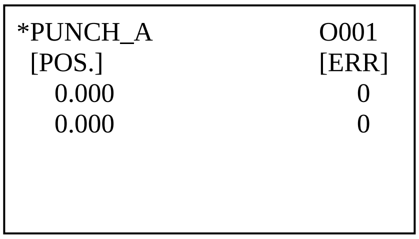
```
*PUNCH_P                O001
 [POS.]                 [ERR]
    0.000                  0
    0.000                  0
```

Fig 9-3 TAPE Function - PUNCH_P

5. Press CYCST to start program transmission to the PC. When the transmission is completed, a message of "Tape_0" as in Fig 9-2 will be displayed.

9.3 All Programs Transferring from CNC Controller to PC (PUNCH_A)

1. On PC side, execute "DNC10.exe" (See Sec 9.10) to bring up the main menu and do the necessary steps to get it ready for transmission. Make sure the communication protocol is correct. Select “Transmit Data” and do the followings in sequence.
    - Select “HUST CNC → PC”.
    - Select “CNC transmit program to PC”.
    - Type in the program name with directory path. Press ENTER.
2. On the CNC controller side, press RESET key.
3. Press TAPE key. Press Cursor↓ key twice to display PUNCH_A as in Fig 9-4.



```
*PUNCH_A                O001
 [POS.]                 [ERR]
    0.000                  0
    0.000                  0
```

Fig 9-4 TAPE Function - PUNCH_A

4. Press CYCST to start program transmission to the PC. When the transmission is completed, a message of "Tape_0" as in Fig 9-2 will be displayed.

9.4 Transfer MCM Data from PC to Controller (RD_MCM)

Format for MCM Data Transfer:

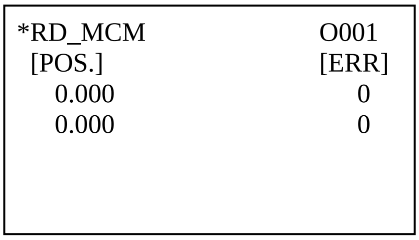
%	
O9002	Program number 9002 designated for MCM data
0000000	MCM #1
0005000	MCM #2
0001500	MCM #3
0000300	MCM #4
0000000	MCM #5
....	....
%	

Notes:

1. The MCM data must start and end with a symbol of %.
2. Program number for MCM data is 9002.
3. No decimal point for MCM data transferred. The unit is 1/1000th of a second for time and μm for length or speed. One line for one MCM data only (7-digit).
   For example, HOME speed on X-axis = 2500.00, it'll show 0250000 after transferred. Software limit on Y-axis = -9999.999 cm, it’ll show –9999999.

Steps for MCM Data Transfer to CNC:

1. On PC side, execute "DNC10.exe" (See Sec 9.10) to bring up the main menu and do the necessary steps to get it ready for transmission. Make sure the communication protocol is correct. Select “Transmit Data” and do the followings in sequence.
   - Select “PC → HUST CNC”.
   - Select “PC send MCM data to CNC”.
   - Type in the file name for MCM data with directory path. Press ENTER.
2. When finish step 1, the following 3 cases may occur on CNC side.
   - If CNC is NOT in execution mode (S080=0) and NOT in TAPE mode, no additional step is necessary on the CNC side and the MCM data will be downloaded to CNC immediately.
   - If CNC is in execution mode (S080=1), wait until CNC finish execution (S080=0). Put CNC in TAPE mode RD_MCM (see Fig 9-5) and press CYCST key to start transferring.
   - If CNC is already in TAPE mode, make sure it’s on RD_MCM page and then press CYCST key to start transferring.



```
*RD_MCM                 O001
 [POS.]                 [ERR]
   0.000                   0
   0.000                   0
```

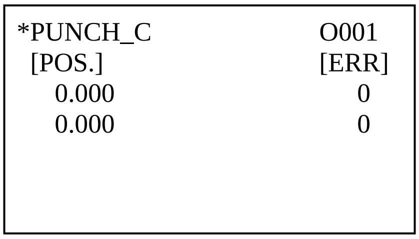
Fig 9-5 MCM Data Transfer from PC to CNC

3. When the transmission is completed, a message of "Tape_0" will be displayed as in Fig 9-2.

Note: If you execute G10 P600L02 under MDI mode after completing the last step, the MCM data will be burned into FLASH-ROM.

9.5 Transfer MCM Data from Controller to PC (PUNCH_C)

1. On PC side, execute "DNC10.exe" (See Sec 9.10) to bring up the main menu and do the necessary steps to get it ready for transmission. Make sure the communication protocol is correct. Select “Transmit Data” and do the followings in sequence.
   - Select “HUST CNC → PC”.
   - Select “CNC transmit program to PC”.
   - Type in the file name for MCM data with directory path. Press ENTER.
2. When finish step 1, the following 3 cases may occur on CNC side.
   - If CNC is NOT in execution mode (S080=0) and NOT in TAPE mode, no additional step is necessary on the CNC side and the MCM data will be uploaded to PC immediately.
   - If CNC is in execution mode (S080=1), wait until CNC finish execution (S080=0). Put CNC in TAPE mode PUNCH_C (see Fig 9-6) and press CYCST key to start transferring.
   - If CNC is already in TAPE mode, make sure it’s on PUNCH_C page and then press CYCST key to start transferring.



```
*PUNCH_C                 O001
 [POS.]                  [ERR]
   0.000                    0
   0.000                    0
```

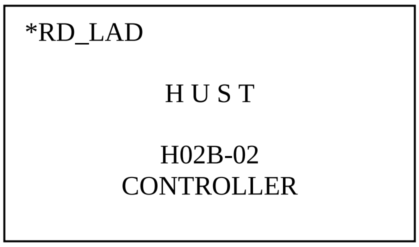
Fig 9-6 MCM Data Transfer from CNC to PC

3. When the transmission is completed, a message of "Tape_0" as in Fig 9-2 will be displayed.

9.6 PLC Ladder Transfer from PC to CNC (RD_LAD) (Ladder Simulation)

1. On PC side, execute "DNC10.exe" (See Sec 9.10) to bring up the main menu and do the necessary steps to get it ready for transmission. Make sure the communication protocol is correct. Select “Transmit Data” and do the followings in sequence.
   - Select “PC → HUST CNC”.
   - Select “PC send LADDER data to CNC”.
   - Type in the file name (*.tsk) for ladder data with directory path. Press ENTER.
2. When finish step 1, the following 3 cases may occur on CNC side.

   - If CNC is NOT in execution mode (S080=0) and NOT in TAPE mode, no additional step is necessary on the CNC side and the PLC data will be downloaded to CNC immediately.
   - If CNC is in execution mode (S080=1), wait until CNC finish execution (S080=0). Put CNC in TAPE mode RD_LAD (see Fig 9-7) and press CYCST key to start transferring.
   - If CNC is already in TAPE mode, make sure it’s on RD_LAD page and then press CYCST key to start transferring.



*RD_LAD

H U S T

H02B-02
CONTROLLER

Fig 9-7 Ladder Transfer from PC to CNC

3. When the transmission is completed, a message of "Tape_0" as in Fig 9-2 will be displayed.

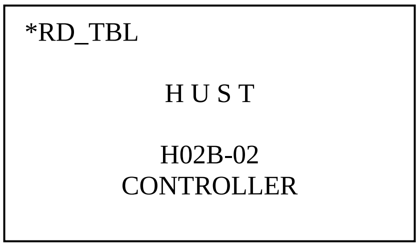
When the transmission is completed, you can proceed to do ladder simulation. If you decide to keep this PLC ladder in the controller, execute G10 P600 L03 to burn the transferred PLC ladder into FLASH-ROM. If you don’t do this step, the transferred PLC will be lost when power-off.

Notes:
1. After transferred to CNC, the PLC ladder will write over all the programs in the memory and leaves 4K of memory for writing new program.
2. C031 in the ladder must be ON (C031=1) for simulation to be effective. If C031=0, no simulation can be performed.
3. If you finish simulation, please execute G10 P2001 to clear all programs in the memory under MDI mode.

9.7 Transfer LCD Screen Display Data from PC to CNC Controller (RD_TBL)

1. On PC side, execute "DNC10.exe" (See Sec 9.10) to bring up the main menu and do the necessary steps to get it ready for transmission. Make sure the communication protocol is correct. Select “Transmit Data” and do the followings in sequence.
   - Select “PC → HUST CNC”.
   - Select “PC send PAGE data to CNC”.
   - Type in the file name (*.tbl) for PAGE data with directory path. Press ENTER.
2. When finish step 1, the following 3 cases may occur on CNC side.
   - If CNC is NOT in execution mode (S080=0) and NOT in TAPE mode, no additional step is necessary on the CNC side and the screen PAGE data will be downloaded to CNC immediately.

- If CNC is in execution mode (S080=1), wait until CNC finish execution (S080=0). Put CNC in TAPE mode RD_TBL (see Fig 9-8) and press CYCST key to start transferring.
- If CNC is already in TAPE mode, make sure it’s on RD_TBL page and then press CYCST key to start transferring.



*RD_TBL

H U S T

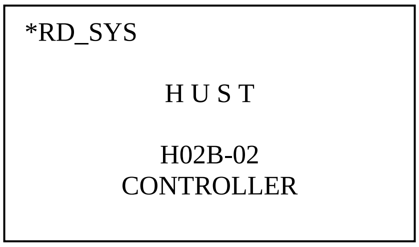
H02B-02
CONTROLLER

Fig 9-8 LCD Screen Data Transfer from PC to CNC

3. When the transmission is completed, execute G10 P600 L04 to burn the newly transferred LCD screen data into FLASH-ROM in MDI mode.

9.8 Transfer Controller System Data from PC to CNC Controller (RD_SYS)

1. On PC side, execute "DNC10.exe" (See Sec 9.10) to bring up the main menu and do the necessary steps to get it ready for transmission. Make sure the communication protocol is correct. Select “Transmit Data” and do the followings in sequence.
   - Select “PC → HUST CNC”.
   - Select “PC send SYSTEM data to CNC”.
   - Type in the file name (*.tsk) for SYSTEM data with directory path. Press ENTER.
2. When finish step 1, the following 3 cases may occur on CNC side.
   - If CNC is NOT in execution mode (S080=0) and NOT in TAPE mode, no additional step is necessary on the CNC side and the screen PAGE data will be downloaded to CNC immediately.
   - If CNC is in execution mode (S080=1), wait until CNC finish execution (S080=0). Put CNC in TAPE mode RD_SYS (see Fig 9-9) and press CYCST key to start transferring.
   - If CNC is already in TAPE mode, make sure it’s on RD_SYS page and then press CYCST key to start transferring.



*RD_SYS

H U S T

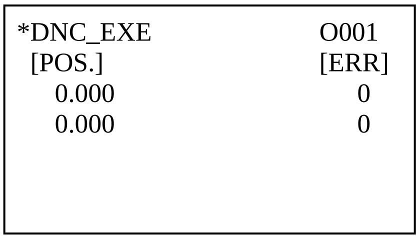
H02B-02
CONTROLLER

Fig 9-9 System Data Transfer from PC to CNC

3. When the transmission is completed, execute G10 P600 L05 to burn the newly transferred system data into FLASH-ROM.

9.9 Transfer Part Program from PC to Controller and Execute the Program (DNC_EXE)

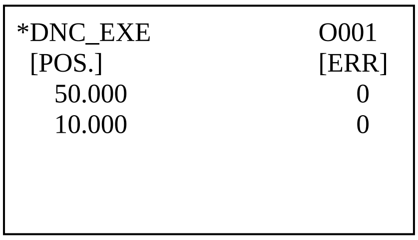
1. On the CNC controller side, press RESET key.
2. Press TAPE key. Press Cursor↑ key to display **DNC_EXE** as in Fig 9-10.



```
*DNC_EXE                O001
 [POS.]                 [ERR]
   0.000                   0
   0.000                   0
```

Fig 9-10 TAPE function – DNC_EXE

3. On PC side, execute "DNC10.exe" (see Sec 9.10) to bring up the main menu and do the necessary steps to get it ready for transmission. Make sure the communication protocol is correct. Select “Transmit Data” and do the followings in sequence.
   - Select “PC → HUST CNC”.
   - Select “PC send program data to CNC”.
   - Type in program name with directory path. Press ENTER.
4. Press CYCST to start program transmission to the controller which will start executing the program block by block. During program execution, the LCD screen will display the axis being executed and its coordinate as in Fig 9-11.



```
*DNC_EXE                O001
 [POS.]                 [ERR]
   50.000                  0
   10.000                  0
```

Fig 9-11 Display of DNC_EXE.

5. When message "Tape_0" as in Fig 9-2 is displayed, it means the program transfer is completed. But the program execution may still be in process. Watch the coordinates display. When execution completed, press RESET to terminate DNC_EXE.

Notes:

1. The program number on the PC side can be omitted for DNC_EXE.
2. Do not use sub-program call in the program.
3. The program must be ended with M02 or M30. (NOT M99)

When you press Cursor↑ or Cursor↓ key under TAPE mode, RD_PRO, PUNCH_P, PUNCH_A, RD_MCM, PUNCH_C, RD_LAD, *RD_Vxx, *PUNCH_F, RD_TBL, RD_SYS & DNC_EXE will appear on screen in sequence. RD_LAD will not appear if C031=0 in PLC. RD_Vxx and PUNCH_F are not for user’s application.

9.10 RS232 Interface -- HUST's DNC10.EXE Software

HUST provides "DNC10.EXE" software for program/data communication between PC and HUST controller. This software is of menu driven and easy to use. Followings are brief descriptions for this software.

9.10.1 Working Environment for DNC10.EXE

DNC.EXE installation

- Personal Computer : IBM PC or compatible with DOS 3.30 or above
- Software : HUST's "DNC10.EXE"
- Copy DNC10.EXE to your PC as follows:
  C:\>md dnc <ENTER>
  C:\>cd dnc <ENTER>
  C:\DNC>copy a:\dnc10.exe <ENTER>

DNC10.EXE program execution

Be sure you are in the DNC directory. Key in DNC and press ENTER. The main menu will be on the screen as shown in Fig 9-12 below:

```
********************************************
**                                        **
**       1. COMMUNICATION PROTOCOL        **
**                                        **
**       2. TRANSMIT PROGRAM              **
**                                        **
**       3. TYPE TRANSMIT DATA            **
**                                        **
**       4. SEARCH FILE.NCD               **
**                                        **
**       PRESS ---- ESC ---- TO QUIT      **
**                                        **
********************************************
      ++++++++++++++++++++++++++++++++
      + Choose:                      +
      ++++++++++++++++++++++++++++++++
```

Fig 9-12

Main menu description:

1. Communication protocol -- For setting port, baud rate, parity.
2. Transmit Program -- Transmittal function selection.
3. Type Transmit Data -- Display the content of the desired program on PC screen.
4. Search File.NCD -- List all “*.NCD” files.

9.10.2 Description of Main Menu of DNC10 Software

1. COMMUNICATION PROTOCOL

On the main menu as shown in Fig 9-12, type “1” on keyboard to bring up the communication protocol as shown in Fig 9-13. You can change item 1 or 2 to match you computer while items 3, 4, and 5 are fixed data and should not be changed. Please note that the baud rate for item 2 should be the same as the one in MCM #64

```
********************************************
**      Revise communication              **
**                                        **
**      1. RS232 PORT  :  1               **
**                                        **
**      2. BAUD RATE   :  9600 bps        **
**                                        **
**      3. PARITY CHECK:  EVEN            **
**                                        **
**      4. WORD LENGTH :  7 bits          **
**                                        **
**      5. STOP BITS   :  2 bits          **
**                                        **
**      PRESS ---- ESC ---- TO QUIT       **
********************************************
      ++++++++++++++++++++++++++++++++
      + Choose:                      +
      ++++++++++++++++++++++++++++++++
```

Fig 9-13

2. TRANSMIT PROGRAM

On the main menu as shown in Fig 9-12, type “2” to bring up the sub-menu of “Transmit Program” as shown in Fig 9-14 where you can choose either “1” if you want to transmit program/data from PC to controller or “2” if transmit from controller to PC.

```
********************************************
**                                        **
**       1. PC -> HUST CNC                **
**           • PROGRAM                    **
**           • MCM  - DATA                **
**           • PLC  - DATA                **
**           • PAGE – DATA                **
**           • SYS  - DATA                **
**                                        **
**       2. HUST CNC -> PC                **
**           • PROGRAM                    **
**           • MCM  - DATA                **
**                                        **
**       PRESS ---- ESC ---- TO QUIT      **
**                                        **
********************************************
      ++++++++++++++++++++++++++++++++
      + Choose:                      +
      ++++++++++++++++++++++++++++++++
```

Fig 9-14

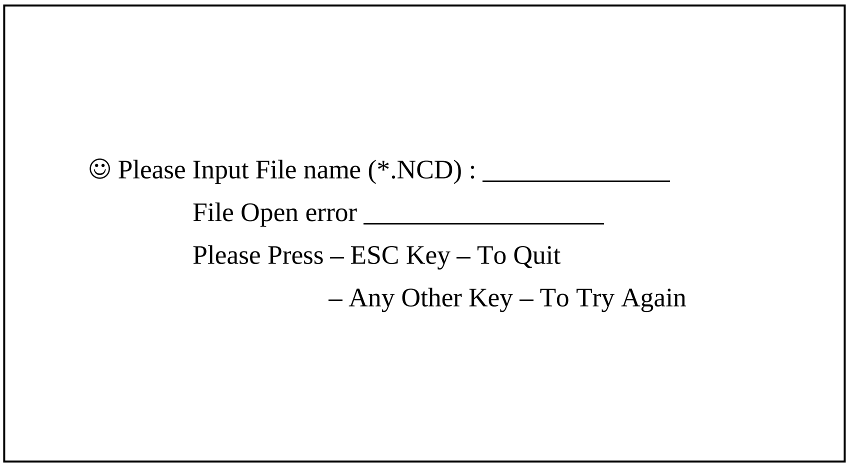
- **PC → HUST CNC** – Transmission from PC to HUST controller

  The available operations are shown in Fig 9-15 which is the screen when you type “1” in Fig 9-14. The option 4 (Send Page Data) means screen display data and the rest of the options are self-explanatory.

```
********************************************
**                                        **
**   1. PC SEND PROGRAM   DATA TO CNC     **
**                                        **
**   2. PC SEND MCM       DATA TO CNC     **
**                                        **
**   3. PC SEND LADDER    DATA TO CNC     **
**                                        **
**   4. PC SEND PAGE      DATA TO CNC     **
**                                        **
**   5. PC SEND SYSTEM    DATA TO CNC     **
**                                        **
**   PRESS ---- ESC ---- TO QUIT          **
**                                        **
********************************************
        ++++++++++++++++++++++++++++++++
        + Choose:                      +
        ++++++++++++++++++++++++++++++++
```

Fig 9-15

  After choosing any one of the 5 options in Fig 9-15, the next screen will ask for a file name for transmission. Please <u>include the appropriate directory path when type in file name</u>. If there is any error in file input, a message page is shown as in Fig 9-16. Press ESC key to get back to the main menu or press any other key to retry your file name.



☺ Please Input File name (*.NCD) : ______________

File Open error ___________________

Please Press – ESC Key – To Quit

– Any Other Key – To Try Again

Fig 9-16

- **HUST CNC → PC** - Transmission from HUST controller to PC

  When type “2” on the screen in Fig 9-14, you get the screen as in Fig 9-17. You have 2 options here, sending program or MCM data to PC. After choosing either option, the next screen will ask for a file name. This is the file name you want it to be stored in PC. Please <u>include the appropriate directory path</u>.

```
****************************************************
**                                                **
**      1. CNC TRANSMIT PROGRAM   TO PC           **
**                                                **
**      2. CNC TRANSMIT MCM DATA  TO PC           **
**                                                **
**      PRESS ---- ESC ---- TO QUIT               **
**                                                **
****************************************************
        ++++++++++++++++++++++++++++++++
        + Choose :                     +
        ++++++++++++++++++++++++++++++++
```

Fig 9-17

3. TYPE TRANSMIT DATA

This is option 3 on the main menu as shown in Fig 9-12. This option allows you to review the content of the program on PC screen before downloading from HUST controller. Typing in “3” on the main menu gives you the next screen as in Fig 9-18. Then type in file name and the content will be displayed after you punch ENTER key. Note that you have to have “PE2” software in the same directory as “DNC10.EXE” to review the content.

```
****************************************************
*                                                  *
*                List NCD File Data                *
*                                                  *
****************************************************
  ☺ Please Input File name (*.NCD) : ____________________
```

Fig 9-18

4. SEARCH FILE.NCD

This is option 4 on the main menu as shown in Fig 9-12. This option allows you to review all the file names with ending “NCD”. Typing in “4” on the main menu will show all “*.NCD” files as shown at the bottom line in Fig 9-19.

```
                    dir : c:\  : *.NCD

                    File 3
< KOP-63.NCD >  < P60.NCD >  < KOP-61.NCD >
```

Fig 9-19

9.11 RS232 Connection

A proper connection between PC and HUST controller is shown in Fig 9-20. Please refer to Connecting Manual for more information. When making connection, please be aware of the followings:

1. The connecting cable should not exceed 15 meters to minimize the potential noise interference. The voltage at the PC interface should be in the range of 10~15 volts.

2. Avoid working in an environment where is under the direct noise interference from the machines such as EDM, electric welder, etc. Do not use the same power outlet as for EDM and electric welder. Twisting the cable may help in noise reduction.

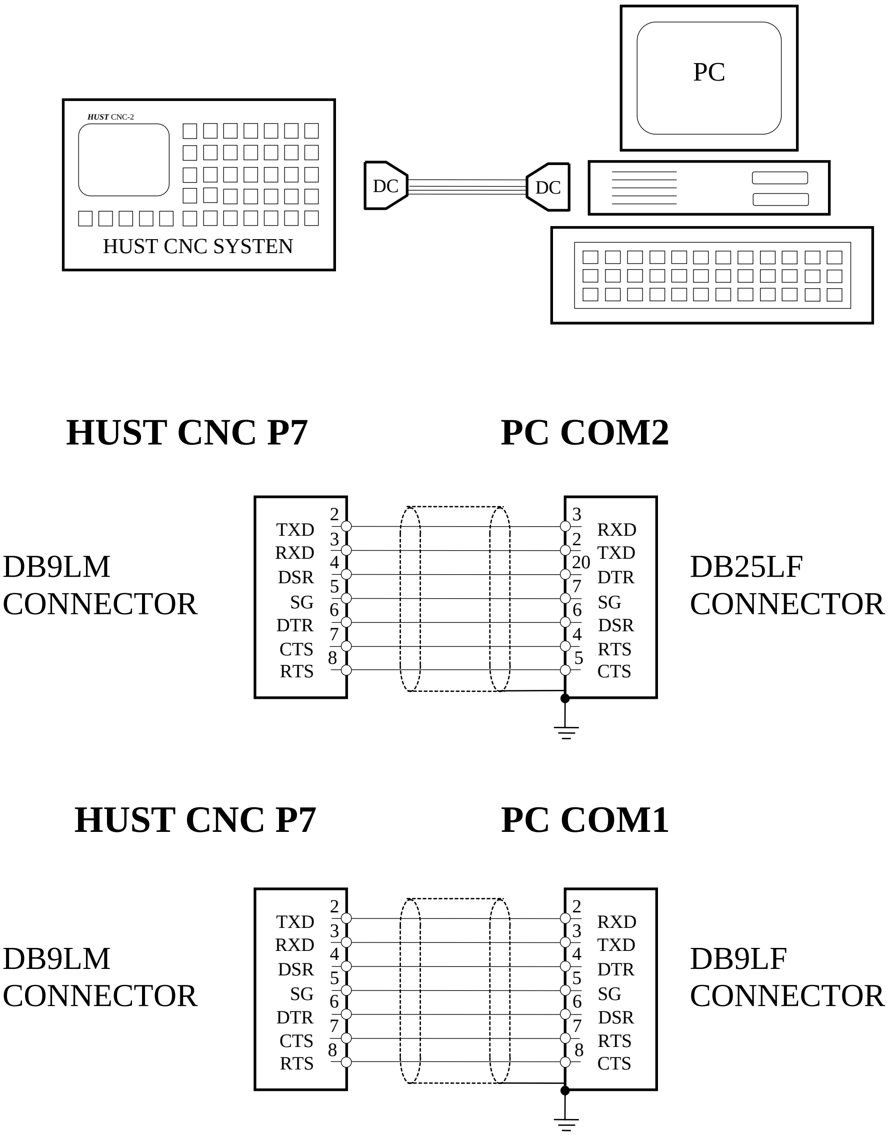
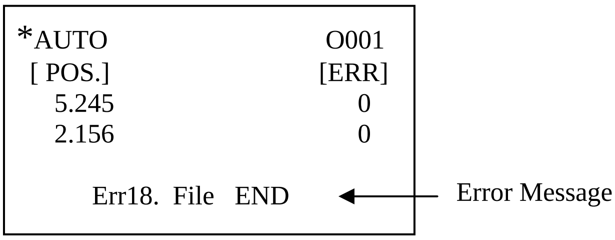


Fig 9-20 RS232 Connection

10 ERROR MESSAGES

When an error occurs during operation, HUST controller will be stopped and an error message displayed on the bottom of LCD screen as shown in Fig 10.1. This chapter explains the error messages and the method to correct them.



```
*AUTO                  O001
 [ POS.]               [ERR]
   5.245                  0
   2.156                  0

      Err18.  File   END
```

Fig 10-1 Error Message Display

ERROR-01 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. Or, execute G10 P1000 in MDI mode to clear all parameters and reenter new data.
2. 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-02 Servo Alarm

Message:
Servo position control (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 settings of MCM parameters are correct. (MCM #46~#49)
3. Check if the work table being overloaded, or any obstruction in the motor. Also check the servo system including the connections.

ERROR-03 Counter Limit

Message: The number in the Counter (MCM #65) for counting M02, M30, and M99 exceeds the one specified by MCM #66.

Recommended Remedy:
1. Clear the number in the Counter or adjust the number in MCM #66 to a larger one. Press RESET key.
2. Under Auto or MDI mode, execute “G10 P201” to clear MCM #65, then press RESET key.

ERROR-04 G60 Missing Repeat No. (LA)

Message: The repetition command code (L) is missing or the G60 function is not being applied under G54 coordinate system.

Recommended Remedy:
Check and revise the missing code (L) in G60 block and the coordinate system.

ERROR-08 Exceeds 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 #64 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 H2N system.

Recommended Remedy:
Check the program for the G-code that is not acceptable by HUST H2N 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-15 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-18 End of File Not Found
Message: Error in the program ending.

Recommended Remedy:

1. Check the ending statement of the program, such as M02, M30, and M99.

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 #67~#70 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 BATT.LOW

Message: The battery (BT1) for data storage is out of charge or service.

Recommended Remedy:
Replace the battery BT1.

ERROR-30.2 BATT.LOW

Message: The battery (BT2) for data storage is out of charge or service.

Recommended Remedy:
Replace the battery BT2.

ERROR-30.A BATT.LOW

Message: The batteries (BT1 and BT2) for data storage are out of charge or service.

Recommended Remedy:
Replace the batteries BT1 and BT2.

ERROR-31 Non 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.

11 Appendix A

11.1 Selection of Servo Motor with Compatible Moment of Inertia

According to manufacturer's general catalogue for servo motor, the maximum load that a servo motor can carry is about 10 times of its moment of inertia. Mathematically, it can be expressed as ($10 * J_M > J_L$).

J_M = Moment of Inertia for Servo Motor (Obtainable from manufacturer's brochure)
J_L = Moment of Inertia for load. (See next section for calculation).

However, 10 times of its moment of inertia is too heavy and the operation is very sluggish. Therefore, it's normally set at 5 times as ($5 * J_M > J_L$).

11.1.1 Calculation of Moment of Inertia for Load

1. Moment of Inertia for Cylindrical Load

Fig 11-1 is a typical setup with cylindrical load. The moment of inertia for the gears "A & B" and the cylindrical load "C" can be calculated by the general equation as follow.

$$Jga, Jgb, \text{ or } Jc = 0.5 * MR^2 \qquad (\text{in kg-cm}^2 \text{ or kg-m}^2 \text{ x}10^{-4}) \qquad (\text{Eq } 11\text{-}1)$$

Where: Jga, Jgb, Jc = Moment of inertia for gear A, B or load C, respectively.
M = Mass for gear A, B or load C, respectively.
R = Radius for gear A, B or load C, respectively.

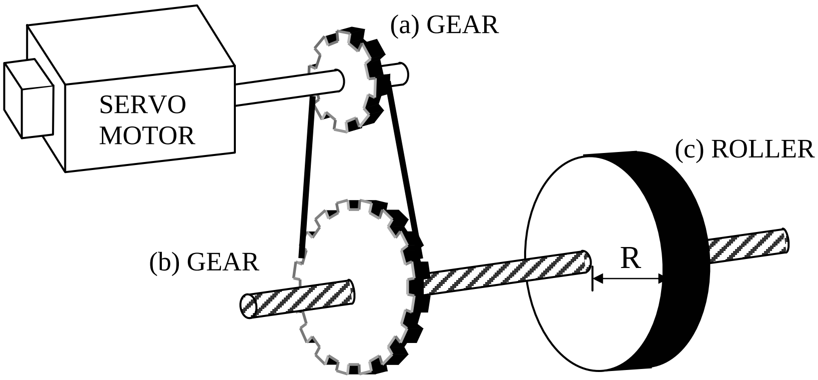


Fig 11-1 Cylindrical Load

The equation to calculate the combined moment of inertia for cylindrical load including gears A and B is as follow.

$$J_L = \frac{(Jgb + Jc)}{(GR)^2} + Jga , \quad (\text{in kg-cm}^2 \text{ or kg-m}^2 \text{x}10^{-4}) \qquad (\text{Eq } 11\text{-}2)$$

Where: GR = gear ratio = (Tooth number of gear B) / (Tooth number of gear A)

2. Moment of Inertia for Square Work Table

Fig 11-2 is a typical setup of a machine tool with work table. Again, the moment of inertia for the gears “A & B” and the ball-screw “C” can be calculated by equation 11-1 and the moment of inertia for square work table “D” is by the following equation.

$$Jtd = M * (P/2\pi)^{2} \qquad \text{(in kg-cm}^{2}\text{ or kg-m}^{2}\text{ x}10^{-4}\text{)} \qquad \text{(Eq 11-3)}$$

Where: Jtd = Moment of inertia for work table D.
M = Mass for work table D.
P = Pitch length of ball-screw C.
$\pi = 3.1416$

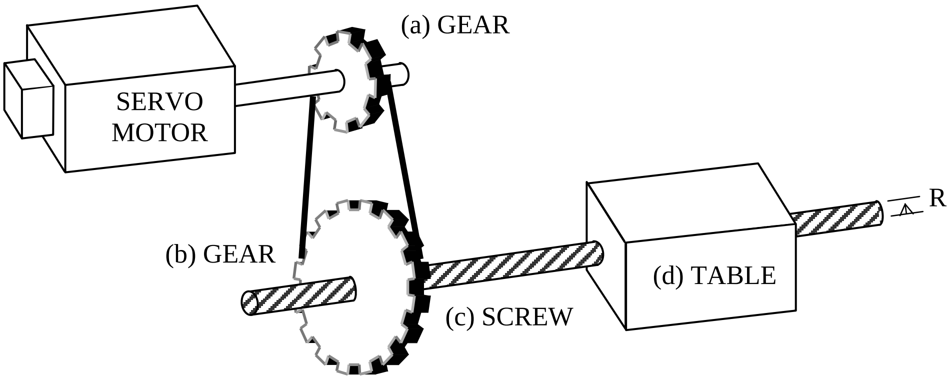


Fig 11-2 Moment of Inertia with Work Table

The equation to calculate the combined moment of inertia for machine tool with square table including gears A, B and ball-screw is as follow.

$$J_L = \frac{(Jgb + Jbc + Jtd)}{(GR)^2} + Jga \,, \quad \text{(in kg - cm}^{2}\text{ or kg - m}^{2}\text{x}10^{-4}\text{)} \qquad \text{(Eq 11 - 4)}$$

Where: Jbc = Moment of inertia for ball-screw C.
GR = gear ratio = (Tooth number of gear B) / (Tooth number of gear A)

Table 11-1 is a table of moment of inertia for square work table with various mass and the pitch length for ball-screw. Tables 11-2 and 11-3 show the moment of inertia for ball-screw with various sizes and weights.

Table 11-1 Moment of Inertia for Square Work Table

	Work Table Weight (kg)									
Pitch(mm)	50	100	150	200	250	300	350	400	450	500
3	0.114	0.228	0.342	0.456	0.570	0.684	0.798	0.912	1.026	1.140
4	0.203	0.405	0.608	0.811	1.013	1.216	1.418	1.621	1.824	2.026
5	0.317	0.633	0.950	1.267	1.583	1.900	2.216	2.533	2.850	3.166
6	0.456	0.912	1.368	1.824	2.280	2.736	3.192	3.648	4.104	4.559
7	0.621	1.241	1.862	2.482	3.103	3.724	4.344	4.965	5.585	6.206
8	0.811	1.621	2.432	3.242	4.053	4.863	5.674	6.485	7.295	8.106
9	1.026	2.052	3.078	4.104	5.129	6.155	7.181	8.207	9.233	10.259
10	1.267	2.533	3.800	5.066	6.333	7.599	8.866	10.132	11.399	12.665
11	1.532	3.065	4.597	6.130	7.662	9.195	10.727	12.260	13.792	15.325
12	1.824	3.648	5.471	7.295	9.119	10.943	12.766	14.590	16.414	18.238
13	2.140	4.281	6.421	8.562	10.702	12.842	14.983	17.123	19.264	21.404
14	2.482	4.965	7.447	9.929	12.412	14.894	17.377	19.859	22.341	24.824
15	2.850	5.699	8.549	11.399	14.248	17.098	19.948	22.797	25.647	28.497
16	3.242	6.485	9.727	12.969	16.211	19.454	22.696	25.938	29.181	32.423
17	3.660	7.320	10.981	14.641	18.301	21.961	25.622	29.282	32.942	36.602
18	4.104	8.207	12.311	16.414	20.518	24.621	28.725	32.828	36.932	41.035
19	4.572	9.144	13.716	18.288	22.861	27.433	32.005	36.577	41.149	45.721
20	5.066	10.132	15.198	20.264	25.330	30.396	35.462	40.528	45.595	50.661
21	5.585	11.171	16.756	22.341	27.927	33.512	39.097	44.683	50.268	55.853
22	6.130	12.260	18.390	24.520	30.650	36.780	42.910	49.039	55.169	61.299
23	6.700	13.400	20.100	26.799	33.499	40.199	46.899	53.599	60.299	66.999
24	7.295	14.590	21.885	29.181	36.476	43.771	51.066	58.361	65.656	72.951
25	7.916	15.831	23.747	31.663	39.579	47.494	55.410	63.326	71.241	79.157
26	8.562	17.123	25.685	34.247	42.808	51.370	59.931	68.493	77.055	85.616
27	9.233	18.466	27.699	36.932	46.164	55.397	64.630	73.863	83.096	92.329
28	9.929	19.859	29.788	39.718	49.647	59.577	69.506	79.436	89.365	99.295
29	10.651	21.303	31.954	42.606	53.257	63.908	74.560	85.211	95.863	106.51
30	11.399	22.797	34.196	45.595	56.993	68.392	79.790	91.189	102.59	113.99

Table 11-2 Moment of Inertia for Ball-screw with Weight Known

	Ball-Screw Diameter (㎜)						
Weight (Kg)	20	25	30	35	40	45	50
1	0.5	0.781	1.125	1.531	2	2.531	3.125
1.5	0.75	1.172	1.688	2.297	3	3.797	4.688
2	1	1.563	2.250	3.063	4	5.063	6.250
2.5	1.25	1.953	2.813	3.828	5	6.328	7.813
3	1.5	2.344	3.375	4.594	6	7.594	9.375
3.5	1.75	2.734	3.938	5.359	7	8.859	10.938
4	2	3.125	4.500	6.125	8	10.125	12.500
4.5	2.25	3.516	5.063	6.891	9	11.391	14.063
5	2.5	3.906	5.625	7.656	10	12.656	15.625
5.5	2.75	4.297	6.188	8.422	11	13.922	17.188
6	3	4.688	6.750	9.188	12	15.188	18.750
6.5	3.25	5.078	7.313	9.953	13	16.453	20.313
7	3.5	5.469	7.875	10.719	14	17.719	21.875
7.5	3.75	5.859	8.438	11.484	15	18.984	23.438
8	4	6.250	9.000	12.250	16	20.250	25.000
8.5	4.25	6.641	9.563	13.016	17	21.516	26.563
9	4.5	7.031	10.125	13.781	18	22.781	28.125
9.5	4.75	7.422	10.688	14.547	19	24.047	29.688
10	5	7.813	11.250	15.313	20	25.313	31.250
10.5	5.25	8.203	11.813	16.078	21	26.578	32.813
11	5.5	8.594	12.375	16.844	22	27.844	34.375
11.5	5.75	8.984	12.938	17.609	23	29.109	35.938
12	6	9.375	13.500	18.375	24	30.375	37.500
12.5	6.25	9.766	14.063	19.141	25	31.641	39.063
13	6.5	10.156	14.625	19.906	26	32.906	40.625
13.5	6.75	10.547	15.188	20.672	27	34.172	42.188
14	7	10.938	15.750	21.438	28	35.438	43.750
14.5	7.25	11.328	16.313	22.203	29	36.703	45.313
15	7.5	11.719	16.875	22.969	30	37.969	46.875
15.5	7.75	12.109	17.438	23.734	31	39.234	48.438
16	8	12.500	18.000	24.500	32	40.500	50.000
16.5	8.25	12.891	18.563	25.266	33	41.766	51.563
17	8.5	13.281	19.125	26.031	34	43.031	53.125
17.5	8.75	13.672	19.688	26.797	35	44.297	54.688
18	9	14.063	20.250	27.563	36	45.563	56.250
18.5	9.25	14.453	20.813	28.328	37	46.828	57.813
19	9.5	14.844	21.375	29.094	38	48.094	59.375
19.5	9.75	15.234	21.938	29.859	39	49.359	60.938
20	10	15.625	22.500	30.625	40	50.625	62.500

Table 11-3 Moment of Inertia for Ball-screw with Length Known

	Ball-Screw Diameter (mm)								
Length(mm)	10	15	20	25	30	35	40	45	50
200	0.0153	0.0775	0.2450	0.5983	1.241	2.298	3.921	6.280	9.572
400	0.0306	0.1551	0.4901	1.1965	2.481	4.596	7.841	12.560	19.144
600	0.0459	0.2326	0.7351	1.7948	3.722	6.895	11.762	18.841	28.716
800	0.0613	0.3101	0.9802	2.3930	4.962	9.193	15.683	25.121	38.288
1000	0.0766	0.3877	1.2252	2.9913	6.203	11.491	19.604	31.401	47.860
1200	0.0919	0.4652	1.4703	3.5895	7.443	13.789	23.524	37.681	57.432
1400	0.1072	0.5427	1.7153	4.1878	8.684	16.088	27.445	43.962	67.004
1600	0.1225	0.6203	1.9604	4.7860	9.924	18.386	31.366	50.242	76.576
1800	0.1378	0.6978	2.2054	5.3843	11.165	20.684	35.286	56.522	86.148
2000	0.1532	0.7753	2.4504	5.9825	12.405	22.982	39.207	62.802	95.720
2200	0.1685	0.8529	2.6955	6.5808	13.646	25.281	43.128	69.082	105.292
2400	0.1838	0.9304	2.9405	7.1790	14.886	27.579	47.048	75.363	114.864
2600	0.1991	1.0079	3.1856	7.7773	16.127	29.877	50.969	81.643	124.437
2800	0.2144	1.0855	3.4306	8.3755	17.368	32.175	54.890	87.923	134.009
3000	0.2297	1.1630	3.6757	8.9738	18.608	34.474	58.811	94.203	143.581
3200	0.2450	1.2405	3.9207	9.5720	19.849	36.772	62.731	100.483	153.153
3400	0.2604	1.3181	4.1658	10.1703	21.089	39.070	66.652	106.764	162.725
3600	0.2757	1.3956	4.4108	10.7685	22.330	41.368	70.573	113.044	172.297
3800	0.2910	1.4731	4.6558	11.3668	23.570	43.667	74.493	119.324	181.869
4000	0.3063	1.5507	4.9009	11.9650	24.811	45.965	78.414	125.604	191.441
4200	0.3216	1.6282	5.1459	12.5633	26.051	48.263	82.335	131.885	201.013
4400	0.3369	1.7057	5.3910	13.1616	27.292	50.561	86.256	138.165	210.585
4600	0.3523	1.7833	5.6360	13.7598	28.532	52.860	90.176	144.445	220.157
4800	0.3676	1.8608	5.8811	14.3581	29.773	55.158	94.097	150.725	229.729
5000	0.3829	1.9383	6.1261	14.9563	31.013	57.456	98.018	157.005	239.301

11.2 How to Calculate the Electric Current Requirement

Watt is a unit to measure power and it is often used to measure electric power. An electric device uses 1 watt when 1 volt electric potential drives 1 ampere of current through it (1 watt = 1 VA or Volt-Amp). The equation to measure power of an electric device is, therefore,

P (Power in Watt) = V (Voltage in Volt) * I (Current in Ampere)

The table below shows that the current requirement is less if the source is a 3-phase current.

Power (Watt)	Voltage (220 V)	Calculation	Current (Amp)
3000 VA	1-phase	3000 VA / 220 V	13.63
3000 VA	3-phase	3000 VA / (220 * $(3)^{0.5}$)	7.87

Example: A machine tool (making capacitor) requires following powers.

1. 9 servo motors with 400W each,	9 * 400 = 3600 W
2. 2 servo motors with 750W each,	2 * 750 = 1500 W
3. 1 spot welder, 4A at 24 volt,	24 * 4 = 96 W
4. 1 Burn-off power, 3A at 35 volt,	35 * 3 = 105 W
5. 2 Controller, 3A at 5 volt,	5 * 3 * 2 = 30 W
6. 1 24V power supply w/ 8 amp max.	24 * 8 = 192 W
7. 1 5V power supply w/ 3 amp max.	5 * 3 = 15 W

Assuming a design factor of 1.5 (from experience), then the maximum power required is
Maximum Power = (3600+1500+96+105+30+192+15) * 1.5 = 8307 W = 8307 VA.

The current required from a 3-phase alternating current with voltage = 415 V is
Current required = 8307 / (415 * $(3)^{0.5}$) = 8307 / (415 * 1.732) = 11.55 Amp.

If an electric wire with a cross-sectional area of 1 mm^2 can carries 8 amperes of current, the minimum wire size for 11.55 amperes of current is 1.45 mm^2.

11.3 Passive Encoder

Two special functions from passive encoder are to be discussed in the following sections.

1. Fly Cut
    - Drum type fly cut
    - Table type fly cut (Fig 11-3)
2. Length Compensation (Fig 11-4)

11.3.1 Table Type Fly Cut – Passive Encoder

Fig 11-3 shows a configuration of table type fly cut. A roll of material on the left is being continuously fed into the cutting table. The X-axis of the CNC controller controls the cutting action while the Y-axis is being used as a passive encoder to calculate the length of feed. Once the length is assured, the cutting action is activated.

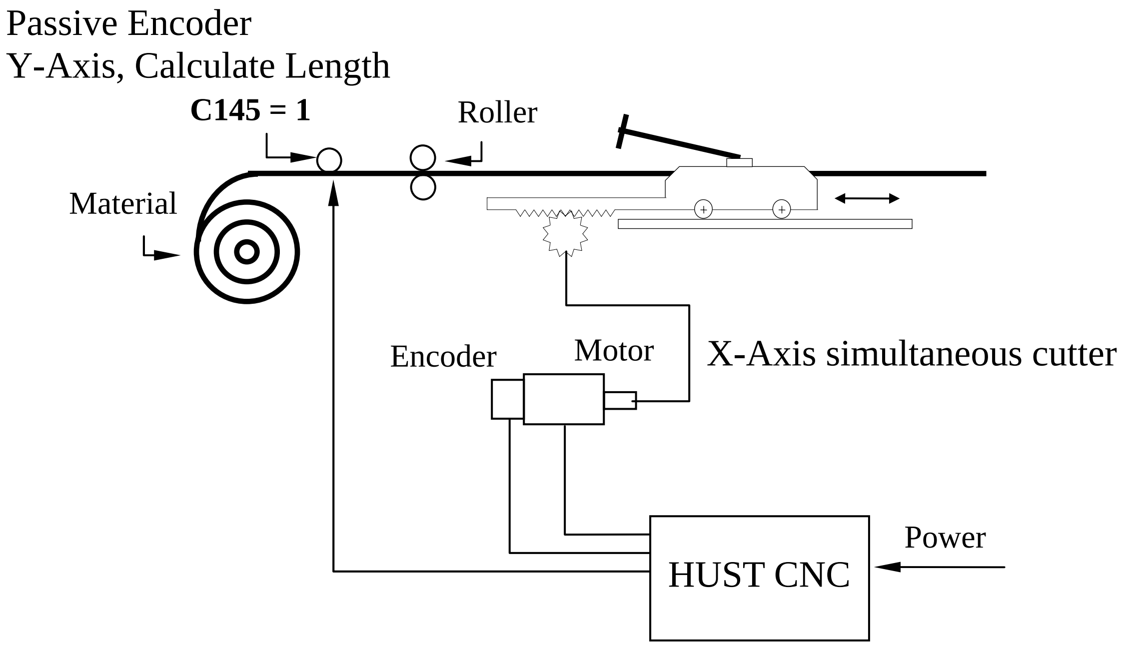


Fig 11-3 Passive Encoder –Table Type Fly Cut

11.3.2 Length Compensation – Passive Encoder

Fig 11-4 shows the application of length compensation using passive encoder. A roll of material on the left is being fed into the machine by the friction force from the two rotating wheels on the right. The rotation of these wheels is controlled by the X-axis of the controller, which is the primary control of the material length being fed into the machine. A passive encoder is also used to measure the length of feed. If this length is the same as that from X-axis, no compensation is necessary. If the lengths are different, compensation (within allowable amount of compensation) on X-axis will be performed as shown in Fig 11-5.

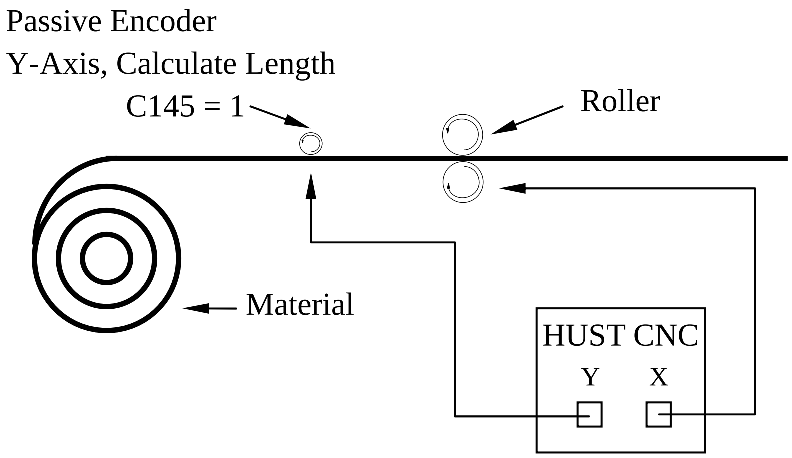
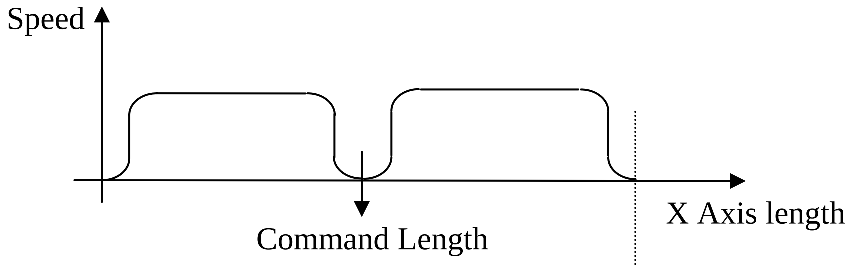


Fig 11-4 Length Compensation – Passive Encoder

⁎ Command Length



⁎ Compensation Length

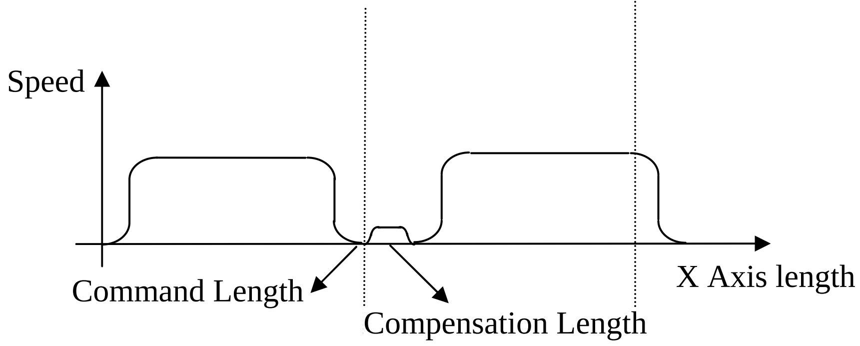


Fig 11-5 Length Compensation on X-axis

Example: Length compensation by passive encoder.

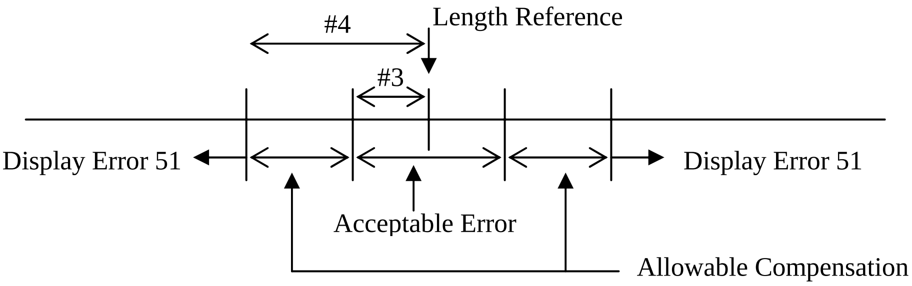
This example shows how you can write a program to do feed length compensation by passive encoder, using G65 function. The example problem is described below:

1. Sheet metal being fed into the machine through rotating wheels, which are controlled by the X-axis of the controller. (Primary length measurement)

2. Passive encoder feedback through Y-axis (Secondary length measurement)
3. Y-axis on LCD screen is used as DRO (digital readout) mode. (C145=1)
4. Use G65 MACRO function to set allowable error.
5. Define variable #1 = Length, #2 = Feed speed, #3 = Acceptable error, #4=Allowable compensation range.
6. System variable #12021 = Program coordinate on X-axis.
   System variable #12022 = Program coordinate on Y-axis.

Example Program:

```
N10 G01 X#1 F#2
N20 G65 L3 P#10 A#12021 B#12022    --- Store the difference (X-Y) in #10.
N30 G65 L22 P#11 A#10              --- Store the absolute value of #10 in #11.
N40 G65 L86 P100 A#11 B#3          --- If |X-Y|≥acceptable error, execute N100.
N50 M30
N100 G65 L86 P200 A#11 B#4         --- If |X-Y||≥allowable compensation,
execute N200.
N110 G01 X#11 F#2                  --- Execute length compensation.
N120 M99
N200 G65 L99 P1                    --- Alarm “Error 51”, |X-Y| exceeds
                                       allowable compensation.
```



12 Appendix B

12.1 LCD Screen Display Editing

LCD screen display of HUST controller is to be edited on PC and the edited data will be compiled and stored in the controller memory. So, a software called “DIY.exe” is required to do screen display editing. Follow the steps below to get into editing mode. You can edit up to a maximum of 63 pages of customized LCD screen display.

1. Getting into the directory of “DIY.exe” in your PC.
2. Type in DIY and press ENTER key to execute “DIY.exe”. The screen appears with the File Name. Type in the desired file name that can be either a new file that you are going to edit or an existing old file for revision.

    - If it’s a new file, a blank window is displayed as shown in Fig 12-1.
    - If it’s an existing file, a simulated display of the old file will appear.
    - To exit the execution program, simply press ESC key or ENTER key without typing any file name.

12.2 Description of Main Function

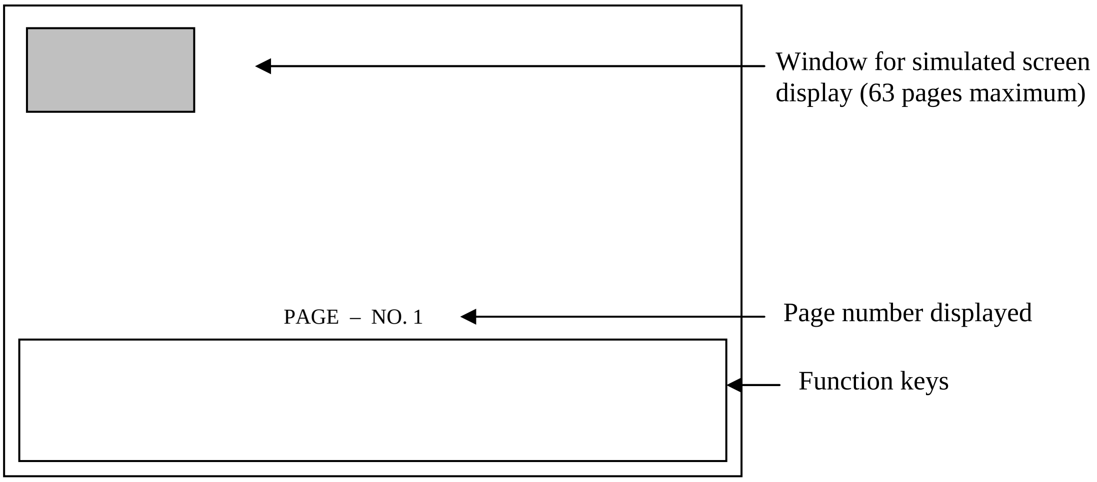


Fig 12-1 Main Function Page

<u>Function Key Description</u>:

The main function keys at the bottom of the screen are as followings.

[F1] - For string data editing
[F2] - To set the width and the height of LCD display area
[F3] - For “object pattern code” editing
[F5] - To bring up the “convert BMP file” page and convert “*.bmp” file to “*.dat” file.
[TAB] - For “object data” editing
[ESC] - To exit the edit mode

Note: When the selected LCD screen size (see next section) is “128 * 64”, you can press [F6] key to enlarge the display by 200% for easier viewing as shown in Fig 12-2.

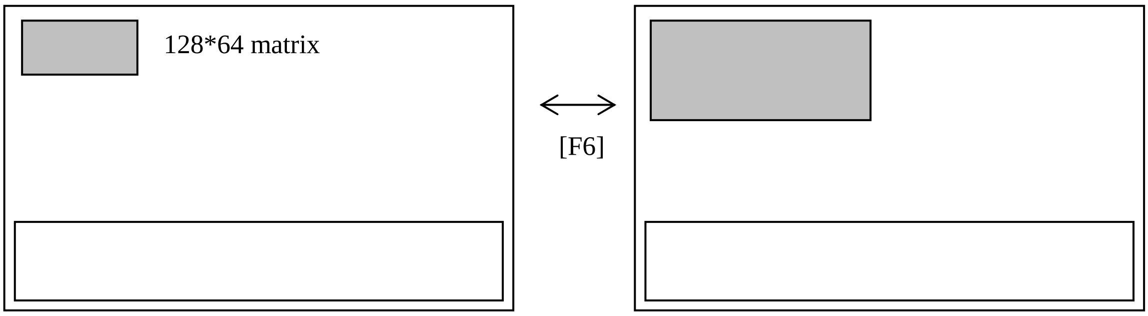
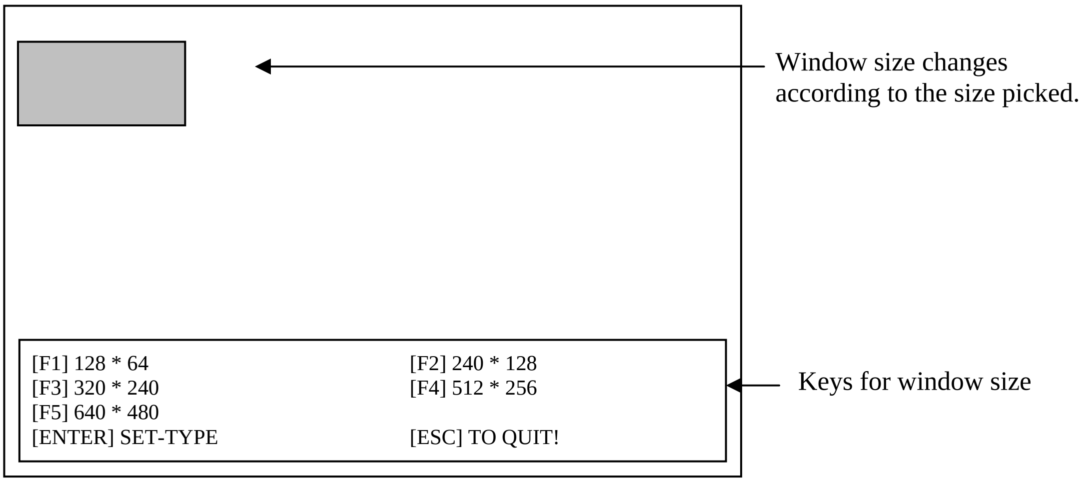


Fig 12-2 Use [F6] to Enlarge the Display Area for Easier Viewing

12.3 Setting Width and Height of LCD Display

Press [F2] key on main function page (Fig 12-1) to bring up the <u>LCD screen size</u> selection page as shown in Fig 12-3.



[F1] 128 * 64 [F2] 240 * 128
[F3] 320 * 240 [F4] 512 * 256
[F5] 640 * 480
[ENTER] SET-TYPE [ESC] TO QUIT!

Window size changes according to the size picked.

Keys for window size

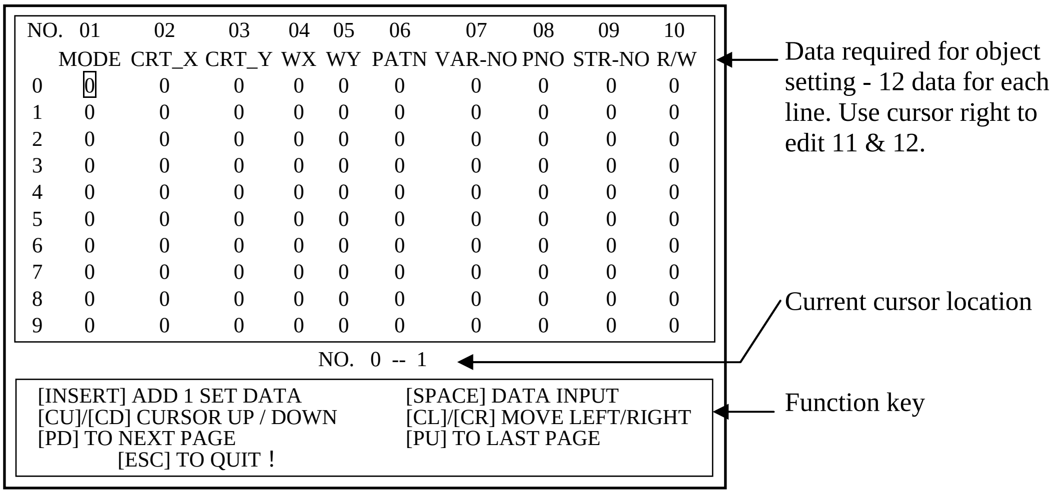
Fig 12-3 Setting Width and Height of LCD Display

[F1] LCD size 128 * 64 [F2] LCD size 240 * 128
[F3] LCD size 320 * 240 [F4] LCD size 512 * 256
[F5] LCD size 640 * 480
[ENTER] To set size selected [ESC] To exit

Press one of the function keys [F1] ~ [F5] to select LCD display size, then press ENTER key. To exit this screen, press [ESC] key. The default size is 128 * 64.

12.4 “Edit Object” Page

Press [TAB] key on main function page (Fig 12-1) to bring up the “edit object” page as shown in Fig 12-4. The object can be numerals, character string, or bit map files. The **<u>data</u>** required for object setting will be discussed in detail in this section.



```
NO.  01     02     03    04   05    06      07     08    09    10
    MODE  CRT_X  CRT_Y  WX   WY   PATN  VAR-NO  PNO  STR-NO  R/W
0    0      0      0     0    0     0       0      0     0     0
1    0      0      0     0    0     0       0      0     0     0
2    0      0      0     0    0     0       0      0     0     0
3    0      0      0     0    0     0       0      0     0     0
4    0      0      0     0    0     0       0      0     0     0
5    0      0      0     0    0     0       0      0     0     0
6    0      0      0     0    0     0       0      0     0     0
7    0      0      0     0    0     0       0      0     0     0
8    0      0      0     0    0     0       0      0     0     0
9    0      0      0     0    0     0       0      0     0     0
                          NO.  0 -- 1
 [INSERT] ADD 1 SET DATA            [SPACE] DATA INPUT
 [CU]/[CD] CURSOR UP / DOWN         [CL]/[CR] MOVE LEFT/RIGHT
 [PD] TO NEXT PAGE                  [PU] TO LAST PAGE
          [ESC] TO QUIT !
```

Data required for object setting - 12 data for each line. Use cursor right to edit 11 & 12.

Current cursor location

Function key

Fig 12-4 "Edit Object" Page for Object Variable Editing

Function Key Description:

[INSERT]:	Add 1 set of data or object	[SPACE]:	Data input
[CU]/[CD]:	Cursor up / Cursor down	[CL]/[[CR]:	Cursor left / Cursor right
[PD]:	Page down	[PU]:	Page up
[ESC]:	Exit to main function page		

＊ **Data #1, Mode** – Specify the type of object being edited

- Numerals: To display the variable data in the controller.
- Character String: To display English characters.
- Bit Map File: To display bit map.

This is a 4-digit variable. Each digit has its meaning as explained below:

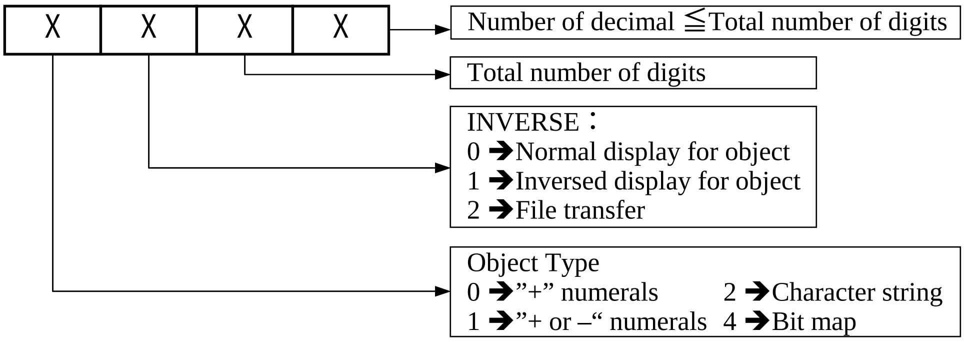


Fig 12-5 Object Edit – Data #1, Mode setting

Ex: 0073 – Object is a positive number, normal display, 7 digits with 3 decimal.
1140 – Object is a number (+ or -), inversed display, 4-digit integer.
2000 – Object is a character string, normal display.
4100 – Object is a bit map, reversed display.

⁎ **Data #2 & #3, “CRT_X” & “CRT_Y” –** Specify the starting location for object

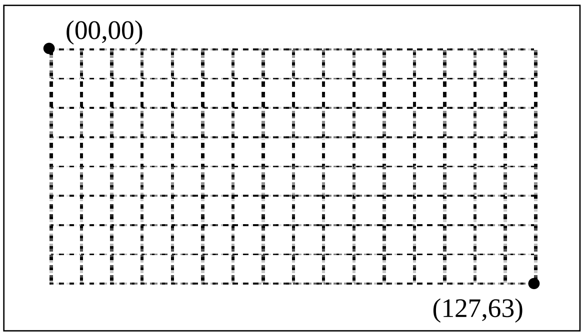


Fig 12-6 Layout of 128 * 64 Dot Matrix for LCD Display

Ex:Specify a character string (TEST) with character size (width=8 and height=8) and the starting location is X=8 & Y=16. The setting is Mode = 2000, CRT_X = 8, CRT_Y =16 and the resulting display is shown in Fig 12-7.

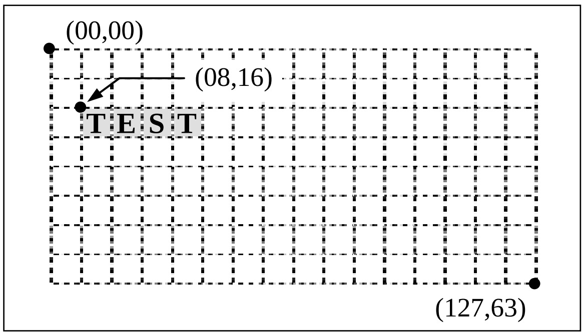


Fig 12-7 Resulting Display of “TEST” on LCD Screen

Note: If CRT_X or CRT_Y is greater than the maximum width (127) or height (63), the character will not be displayed.

⁎ **Data #4 & #5, “WX” & “WY” –** Not available at this time

⁎ **Data #6, “PATN” –** Specify character code or bit map file code

- When Mode setting = numeral or character, “PATN” represents character code. Its setting range is 0 ~ 5.

PATN = “5”,	the character size is 24*24.
PATN = “4”,	the character size is 16*16.
PATN = “3”,	the character size is 12*24.
PATN = “2”,	the character size is 08*16.
PATN = “1”,	the character size is 08*12.
PATN = “0”,	the character size is 08*08.

- When Mode setting = bit map file, “PATN” represents bit map file code. Its setting range is 6 ~ 255.

The procedures to display a bit map is as follows:

Step 1: Use MSPAINT to draw a map as in Fig 12-8. Save it with a file name HUST.BMP.

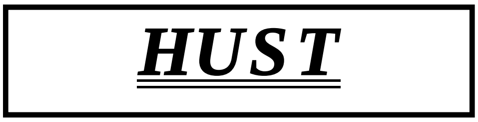
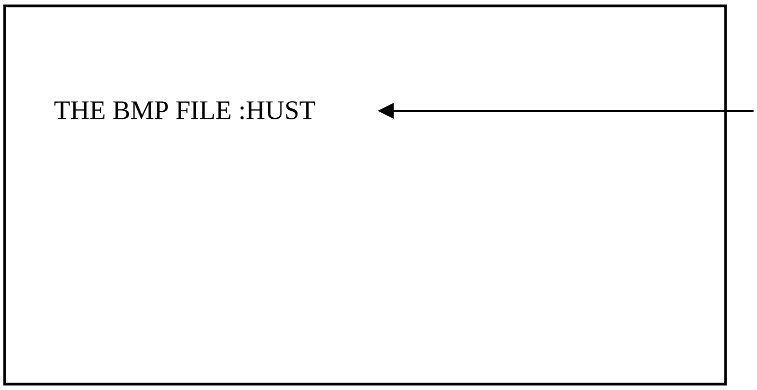


Fig 12-8 Map Drawn Using MSPAINT

Step 2: Execute DIY.exe file and press [F5] key on main function key page to get in “edit object pattern code”. Type in “*.BMP” file name and press ENTER to convert the file to “*.DAT” file as in Fig 12-9. Press any key to return to main function page.

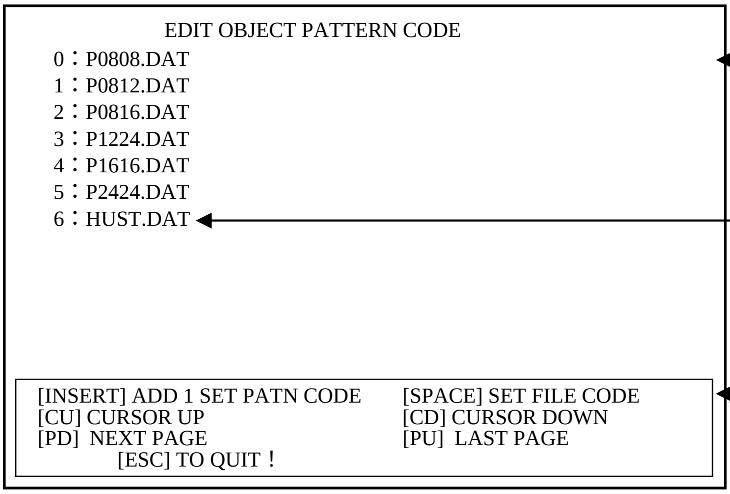


```
THE BMP FILE :HUST
```

Convert “*.BMP” file to “*.DAT” file by typing in the desired file name

Fig 12-9 Convert “*.BMP” file to “*.DAT” file

Step 3: On main function page, press [F3] key to get in “edit object pattern code” area. Press SPACE bar to bring up the code corresponding to the cursor location, then type in the desired “*.DAT” file name (HUST in this case) as in Fig 12-10.



```
              EDIT OBJECT PATTERN CODE
  0：P0808.DAT
  1：P0812.DAT
  2：P0816.DAT
  3：P1224.DAT
  4：P1616.DAT
  5：P2424.DAT
  6：HUST.DAT

[INSERT] ADD 1 SET PATN CODE      [SPACE] SET FILE CODE
[CU] CURSOR UP                    [CD] CURSOR DOWN
[PD]  NEXT PAGE                   [PU]  LAST PAGE
          [ESC] TO QUIT！
```

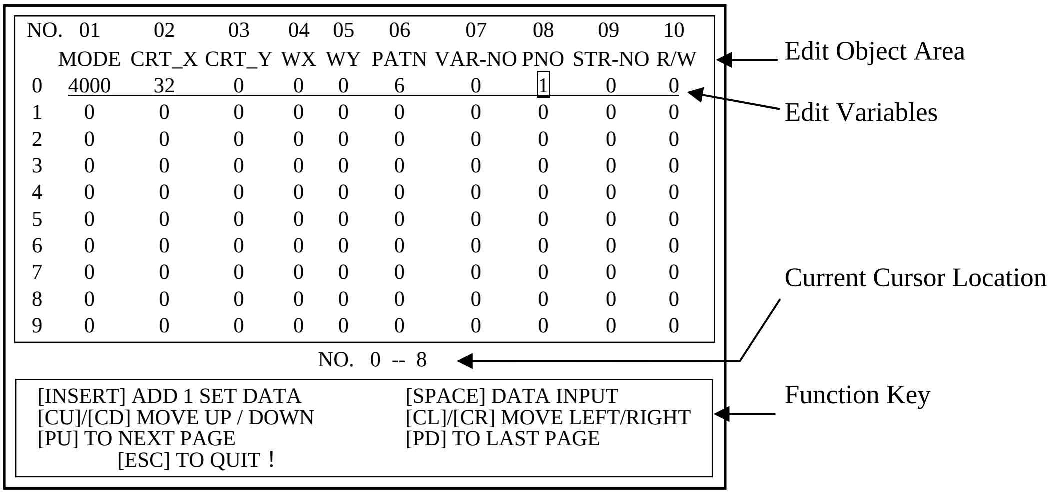
Object Pattern Code Area

HUST.DAT Code on Left

Function Key

Fig 12-10 “Object Pattern Code” area for Code Editing

Step 4: On main function page, press [TAB] key to get in "edit object data" area. Move cursor to PATN, press SPACE bar, type in the code assigned in step 3 and ENTER. Set Mode = 4000 or 4100. See Fig 12-11.



NO.	01	02	03	04	05	06	07	08	09	10
	MODE	CRT_X	CRT_Y	WX	WY	PATN	VAR-NO	PNO	STR-NO	R/W
0	4000	32	0	0	0	6	0	1	0	0
1	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	0	0

NO. 0 -- 8

[INSERT] ADD 1 SET DATA [SPACE] DATA INPUT
[CU]/[CD] MOVE UP / DOWN [CL]/[CR] MOVE LEFT/RIGHT
[PU] TO NEXT PAGE [PD] TO LAST PAGE
[ESC] TO QUIT !

Fig 12-11 "Edit Object" Area

Step 5: Press ESC key to return to main function page. The edited map for LCD display will be shown as in Fig 12-12.

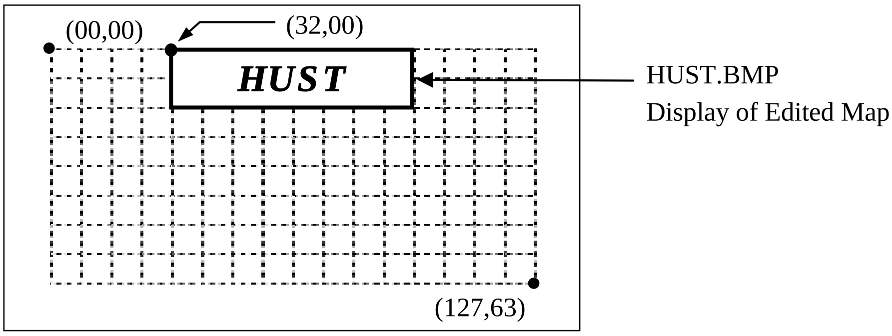


Fig 12-12 LCD Display of Edited Map

* **Data 7, "VAR-NO"** – Specify a variable number to display its content

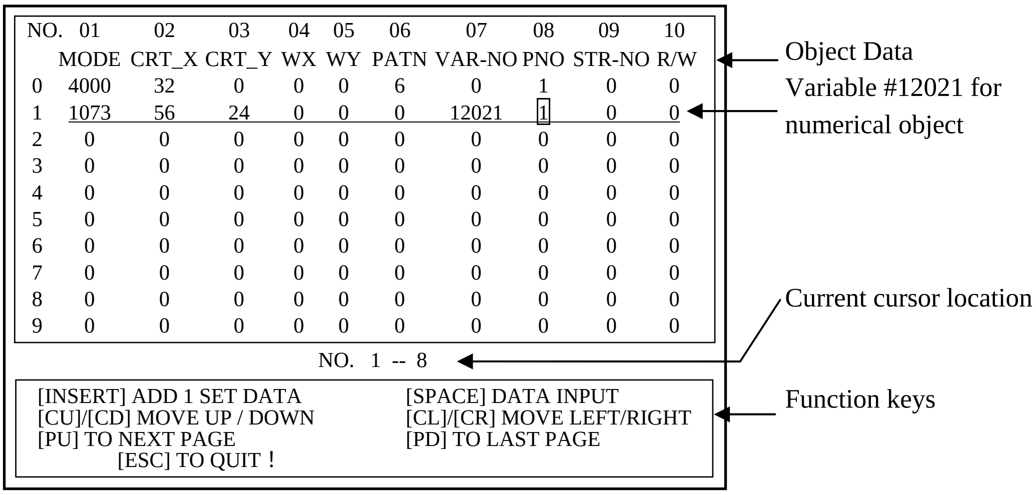
The data in HUST controller are stored in memory as variables.

Variable #1 ~ #9999	User defined variables
#10000 ~ #10255	For registers
#10256 ~ #10511	For counter
#10512 ~ #10767	For timer
#11001 ~ #12000	For MCM parameters
#12000 ~ up	Controller system data

Note: Please refer to Section 6.4 in Connection Manual for variable definition.

Followings are steps to display a numerical object on LCD screen. Use variable #12021 for explanation.

Step 1: On main function page, press [TAB] key. Move cursor to the column VAR-NO on the appropriate line. Press SPACE bar, type in 12021 and press ENTER as shown in Fig 12-13.



NO.	01	02	03	04	05	06	07	08	09	10
	MODE	CRT_X	CRT_Y	WX	WY	PATN	VAR-NO	PNO	STR-NO	R/W
0	4000	32	0	0	0	6	0	1	0	0
1	1073	56	24	0	0	0	12021	1	0	0
2	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	0	0

NO. 1 -- 8

[INSERT] ADD 1 SET DATA [SPACE] DATA INPUT
[CU]/[CD] MOVE UP / DOWN [CL]/[CR] MOVE LEFT/RIGHT
[PU] TO NEXT PAGE [PD] TO LAST PAGE
[ESC] TO QUIT !

Fig 12-13 Variable Number Input for Numerical Object

Step 2: Press ESC to return to main function page and the edited content is displayed as shown as Fig 12-14.

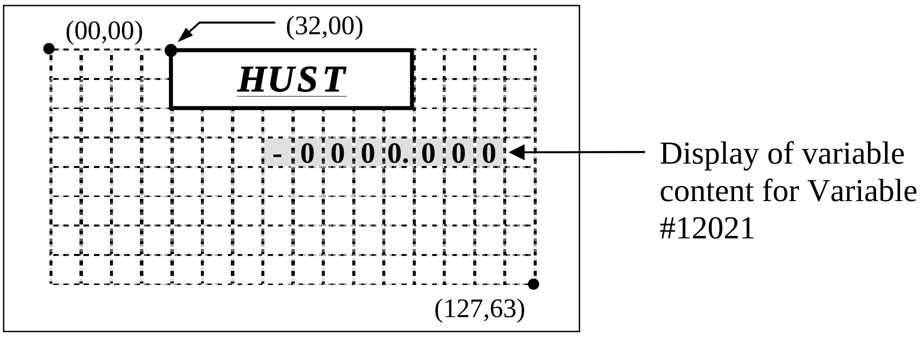


Fig 12-14 LCD Display for Page 1 Object

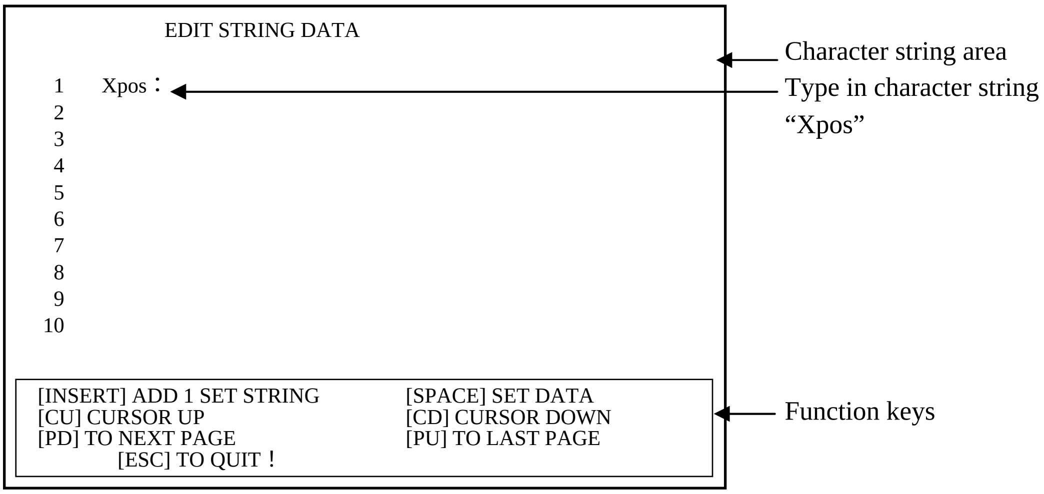
＊ **Data 8, "PNO"** – Specify the page number for the object to be displayed

The page number is ranging from 0 ~ 63.

＊ **Data 9, "STR-NO"** – Specify the character string code for the object to be displayed

The procedure to display a character string is as follows.

Step 1: On main function screen, press [F1] key to get into character string editing area. Press SPACE bar, type character string and press ENTER as in Fig 12-15.



EDIT STRING DATA

1 Xpos :
2
3
4
5
6
7
8
9
10

[INSERT] ADD 1 SET STRING
[SPACE] SET DATA
[CU] CURSOR UP
[CD] CURSOR DOWN
[PD] TO NEXT PAGE
[PU] TO LAST PAGE
[ESC] TO QUIT !

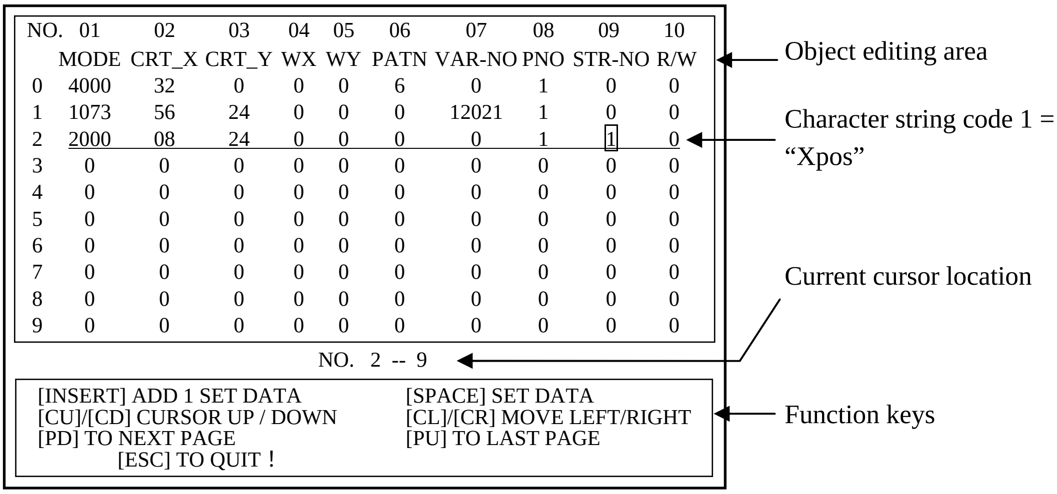
Character string area

Type in character string "Xpos"

Function keys

Fig 12-15 Character String Editing Page

Step 2: Return to main function screen. Press [TAB] key to get into object editing area. Edit object data including "STR-NO" as in Fig 12-16.



NO.	01	02	03	04	05	06	07	08	09	10
	MODE	CRT_X	CRT_Y	WX	WY	PATN	VAR-NO	PNO	STR-NO	R/W
0	4000	32	0	0	0	6	0	1	0	0
1	1073	56	24	0	0	0	12021	1	0	0
2	2000	08	24	0	0	0	0	1	1	0
3	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	0	0

NO. 2 -- 9

[INSERT] ADD 1 SET DATA
[SPACE] SET DATA
[CU]/[CD] CURSOR UP / DOWN
[CL]/[CR] MOVE LEFT/RIGHT
[PD] TO NEXT PAGE
[PU] TO LAST PAGE
[ESC] TO QUIT !

Object editing area

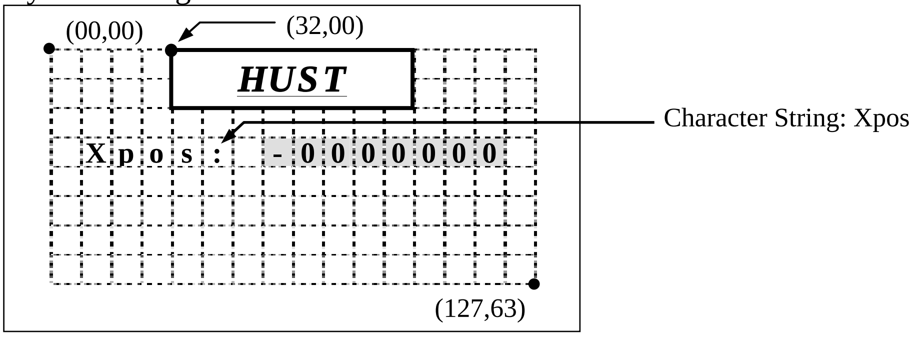
Character string code 1 = "Xpos"

Current cursor location

Function keys

Fig 12-16 Object Editing Including "STR-NO"

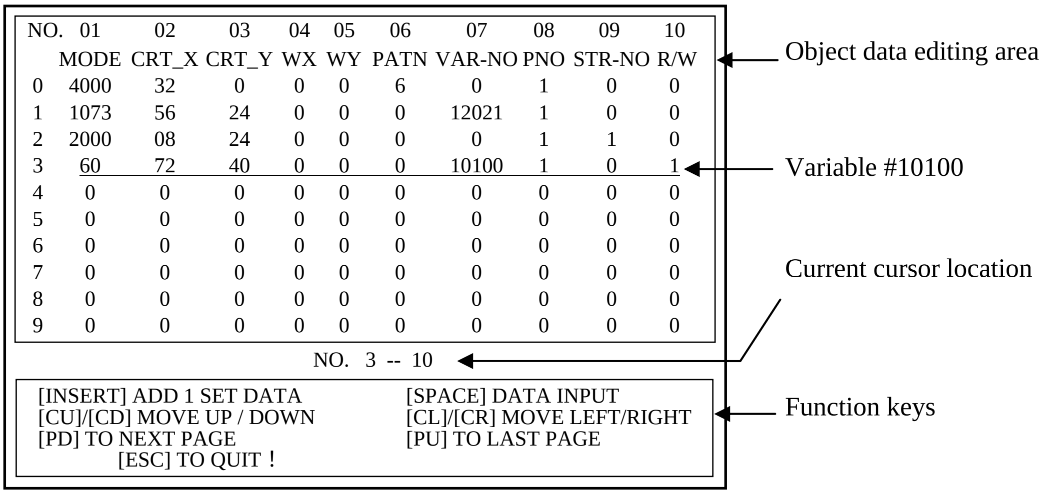
Step 3: Return to main function screen and the edited content for page 1 is displayed as in Fig 12-17.



Character String: Xpos

Fig 12-17 LCD Display for Page 1 Object

- **Data #10, "R/W"** – Specify if the number displayed can be changed



NO.	01	02	03	04	05	06	07	08	09	10
	MODE	CRT_X	CRT_Y	WX	WY	PATN	VAR-NO	PNO	STR-NO	R/W
0	4000	32	0	0	0	6	0	1	0	0
1	1073	56	24	0	0	0	12021	1	0	0
2	2000	08	24	0	0	0	0	1	1	0
3	60	72	40	0	0	0	10100	1	0	1
4	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	0	0

NO. 3 -- 10

[INSERT] ADD 1 SET DATA
[SPACE] DATA INPUT
[CU]/[CD] MOVE UP / DOWN
[CL]/[CR] MOVE LEFT/RIGHT
[PD] TO NEXT PAGE
[PU] TO LAST PAGE
[ESC] TO QUIT !

Fig 12-18 Object Editing Including "R/W"

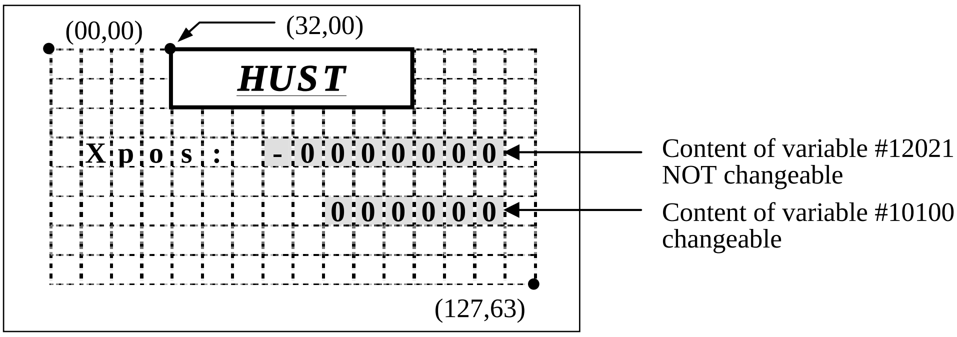


Fig 12-19 LCD Display for Page 1 Object

- **Data #11 & #12, "MIN" & "MAX"** – Specify the minimum and maximum value

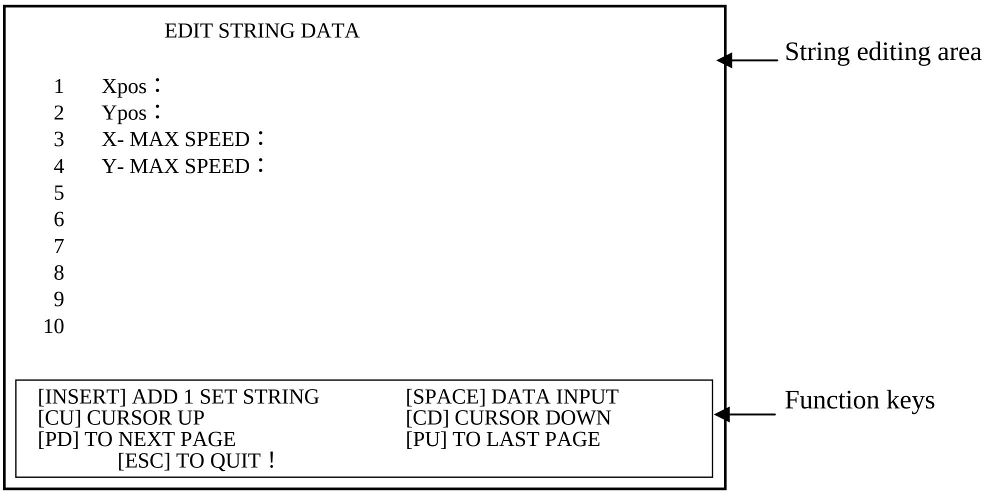
  When the value is equal to 0 (zero), no limit is imposed. When the value is smaller than the minimum or greater than the maximum, LCD will display *****.

12.5 "Edit String Data" Page

On main function screen, press [F1] key to get into character string editing area as shown in Fig 12-20. To edit a string data, simply move cursor (or use insert key) to the desired place, press SPACE bar, type in the characters and ENTER. The maximum number of characters permissible in a character string is 31. For LCD spec of 128*64 (HUST H2 controller), a line of space can display 16 characters.

Function Key Description:

[INSERT]:	Add 1 set of character string	[SPACE]:	Data input
[CU]:	Cursor up	[CD]:	Cursor down
[PD]:	Page down	[PU]:	Page up
[ESC]:	Exit to main function page		

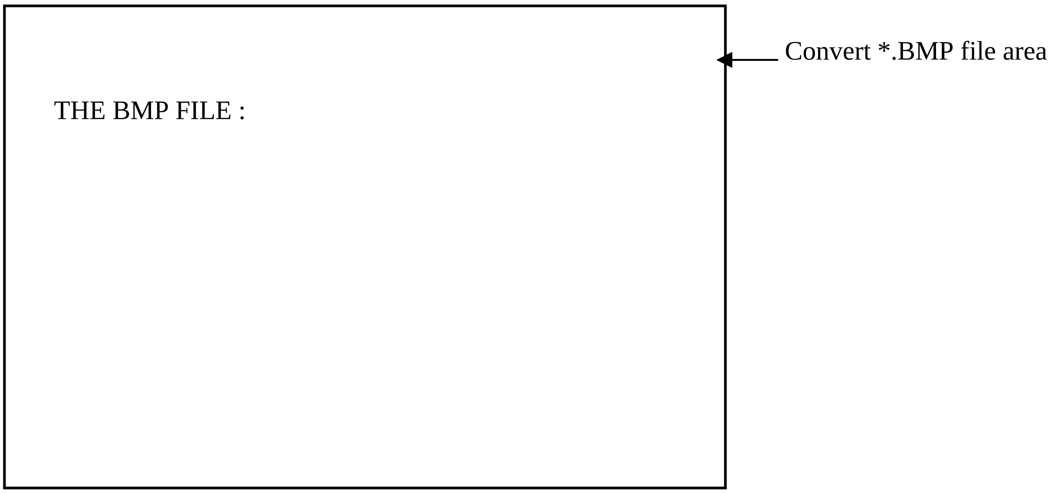


```
                EDIT STRING DATA

  1     Xpos：
  2     Ypos：
  3     X- MAX SPEED：
  4     Y- MAX SPEED：
  5
  6
  7
  8
  9
 10

 [INSERT] ADD 1 SET STRING          [SPACE] DATA INPUT
 [CU] CURSOR UP                     [CD] CURSOR DOWN
 [PD] TO NEXT PAGE                  [PU] TO LAST PAGE
         [ESC] TO QUIT！
```

Fig 12-20 Character String Editing Area

12.6 “Convert BMP File” Page

On main function page, press [F5] key to get into “*.BMP” file conversion page as shown in Fig 12-21. To convert “*.BMP” file, type in the file name and press ENTER key. Then press any key to return to the main function page. After conversion, it generates two files, “*.DAT” and “*.PRJ”.



```
THE BMP FILE :
```

Fig 12-21 “Convert *.BMP File” Page

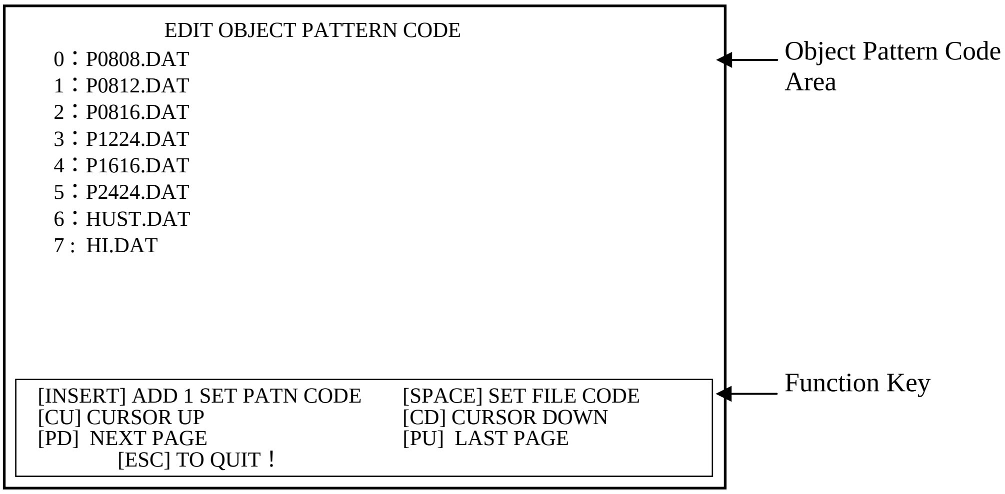
12.7 “Edit Object Pattern Code” Page

On main function page, press [F3] key to get into object pattern code editing area as shown in Fig 12-22. Press SPACE bar to bring up the code corresponding to the cursor location, then type in the desired “*.DAT” file name. Note that you have to convert the map “*.BMP” to “*.DAT” before assigning a code in this page.

<u>Function Key Description</u>:

[INSERT]:	Add a map file	[SPACE]:	For file name input

[CU]:	Cursor up	[CD]:	Cursor down
[PD]:	Page down	[PU]:	Page up
[ESC]:	Exit to main function page		



EDIT OBJECT PATTERN CODE

0 : P0808.DAT
1 : P0812.DAT
2 : P0816.DAT
3 : P1224.DAT
4 : P1616.DAT
5 : P2424.DAT
6 : HUST.DAT
7 : HI.DAT

[INSERT] ADD 1 SET PATN CODE [SPACE] SET FILE CODE
[CU] CURSOR UP [CD] CURSOR DOWN
[PD] NEXT PAGE [PU] LAST PAGE
[ESC] TO QUIT !

Fig 12-22 “Object Pattern Code” area for Code Editing

12.8 Example Display

Assuming that you would like to display the tool’s positions and speeds in both X- and Y-axis during program execution as shown in figures Fig 12-23 and Fig 12-24. To design these two pages of display, the object data, pattern code, and character string data are to be edited as shown in Fig 12-27, Fig 12-26, and Fig 12-25.

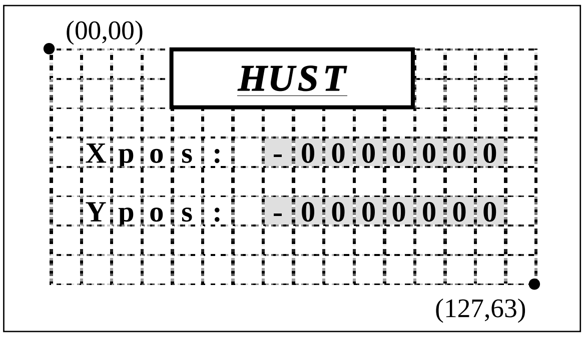


Fig 12-23 Tool Position Display on LCD Screen (Page 1)

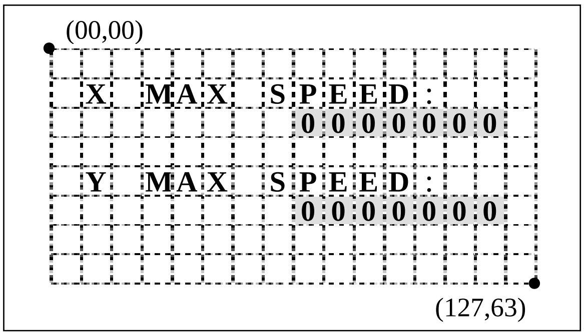
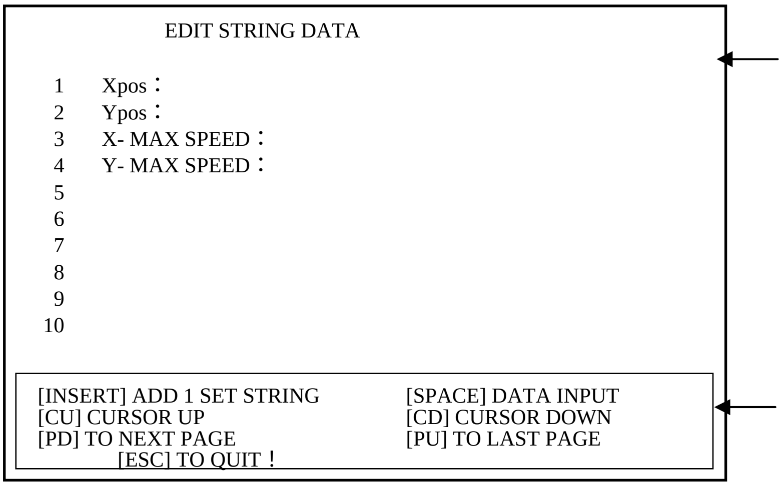


Fig 12-24 Tool Speed Display on LCD Screen (Page 2)

Character String Data



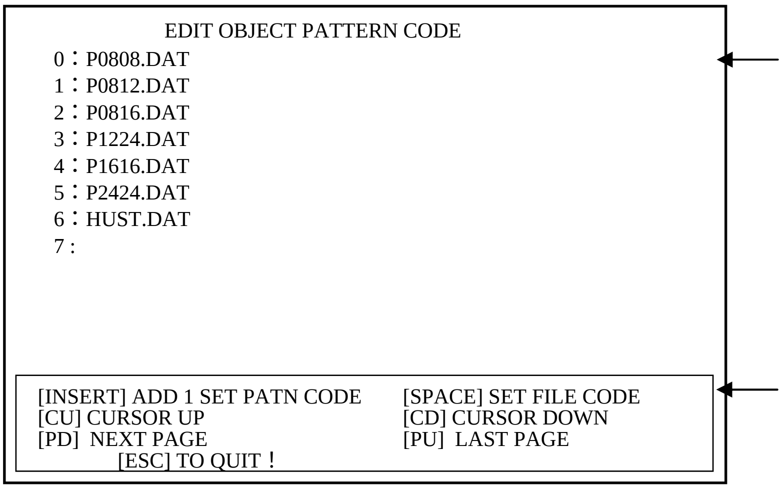
```
                EDIT STRING DATA

  1     Xpos：
  2     Ypos：
  3     X- MAX SPEED：
  4     Y- MAX SPEED：
  5
  6
  7
  8
  9
 10

 [INSERT] ADD 1 SET STRING          [SPACE] DATA INPUT
 [CU] CURSOR UP                     [CD] CURSOR DOWN
 [PD] TO NEXT PAGE                  [PU] TO LAST PAGE
         [ESC] TO QUIT！
```

String editing area

Function keys

Fig 12-25 Character String Editing Page

Object Pattern Code Page



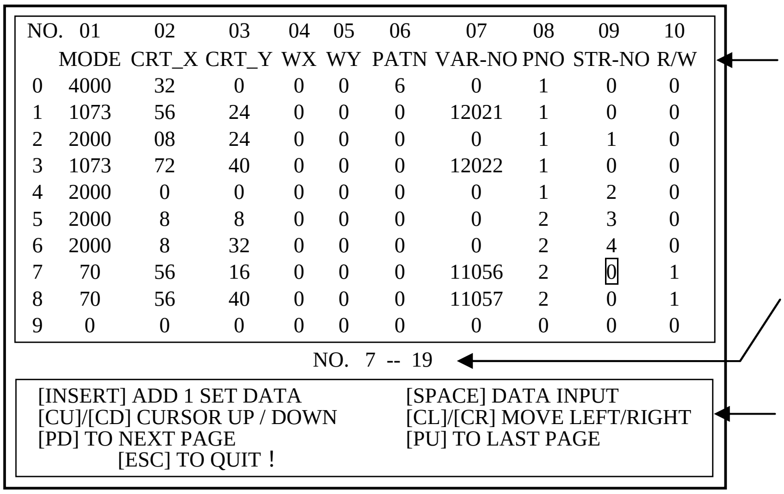
```
                EDIT OBJECT PATTERN CODE
  0：P0808.DAT
  1：P0812.DAT
  2：P0816.DAT
  3：P1224.DAT
  4：P1616.DAT
  5：P2424.DAT
  6：HUST.DAT
  7 :

 [INSERT] ADD 1 SET PATN CODE       [SPACE] SET FILE CODE
 [CU] CURSOR UP                     [CD] CURSOR DOWN
 [PD]  NEXT PAGE                    [PU]  LAST PAGE
         [ESC] TO QUIT！
```

Object Pattern Code Area

Function Key

Fig 12-26 Object Pattern Code Editing Page

Object Data Page



```
NO.  01      02     03    04   05    06      07     08    09     10
    MODE  CRT_X  CRT_Y  WX   WY   PATN  VAR-NO  PNO  STR-NO  R/W
0   4000    32      0    0    0     6       0     1     0      0
1   1073    56     24    0    0     0   12021     1     0      0
2   2000    08     24    0    0     0       0     1     1      0
3   1073    72     40    0    0     0   12022     1     0      0
4   2000     0      0    0    0     0       0     1     2      0
5   2000     8      8    0    0     0       0     2     3      0
6   2000     8     32    0    0     0       0     2     4      0
7     70    56     16    0    0     0   11056     2     0      1
8     70    56     40    0    0     0   11057     2     0      1
9      0     0      0    0    0     0       0     0     0      0

                      NO.  7  --  19

 [INSERT] ADD 1 SET DATA            [SPACE] DATA INPUT
 [CU]/[CD] CURSOR UP / DOWN         [CL]/[CR] MOVE LEFT/RIGHT
 [PD] TO NEXT PAGE                  [PU] TO LAST PAGE
         [ESC] TO QUIT！
```

Object Data

Current cursor location

Function Keys

Fig 12-27 Object Data Area