HUST H2N CNC

CONNECTING MANUAL

Feb , 2006

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

CONTENTS

1 INTRODUCTION

This manual explains the HUST H2N controller's electrical as well as structural design necessary for connecting the CNC to the machine tool. It also describes the HUST H2N connection signals including input, output, internal signals, such as S-bits and C-bits. This manual also describes the functions and the ladder diagram with corresponding signals. For the functional information of HUST H2N controller, please refer to the HUST H2N Operation Manual. This manual is intended for users with some basic electrical and electronic knowledge.

Chapters 1~5 are for those who are interested in electrical connection and Chapters 6~8 are for those who are familiar with HUST PLC Editor. Fig 1-1 is a typical system application of HUST H2N controller to the external devices. The main features (also see Chap 1 of Operation Manual) of the system are described below.

1. The system can be remotely controlled by PC, HUST CRT screen and PC touch screen through RS232 interface.
2. The system can handle thumb switch signals through PLC ladder program.
3. The system can control both AC servo and stepping motor.
4. The system provides DI/DO=24/16 points. Also included are connections for MPG, spindle and Skip sensor.

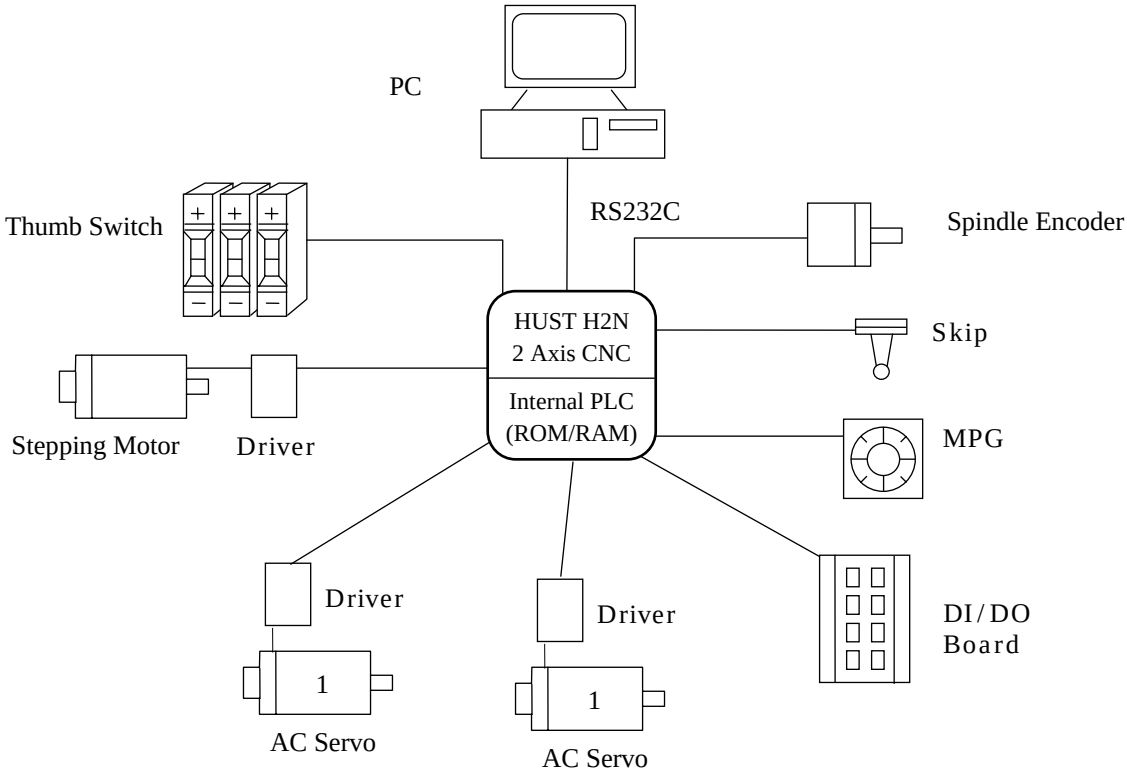


Fig 1-1 Applications of HUST H2N Controller

2 CONFIGURATION

2.1 System Configuration

The HUST H2N system generally has the following configuration as shown in Fig. 2-1. The main components of this configuration are listed below.

1. CPU Main Board
2. Keyboard Unit
3. MPG Hand-wheel
4. RS232 Communication Interface
5. Servo Driver and Motor
6. Power Supply
7. I/O Relay Board
8. Analog Output for Spindle Drive (0~10V)

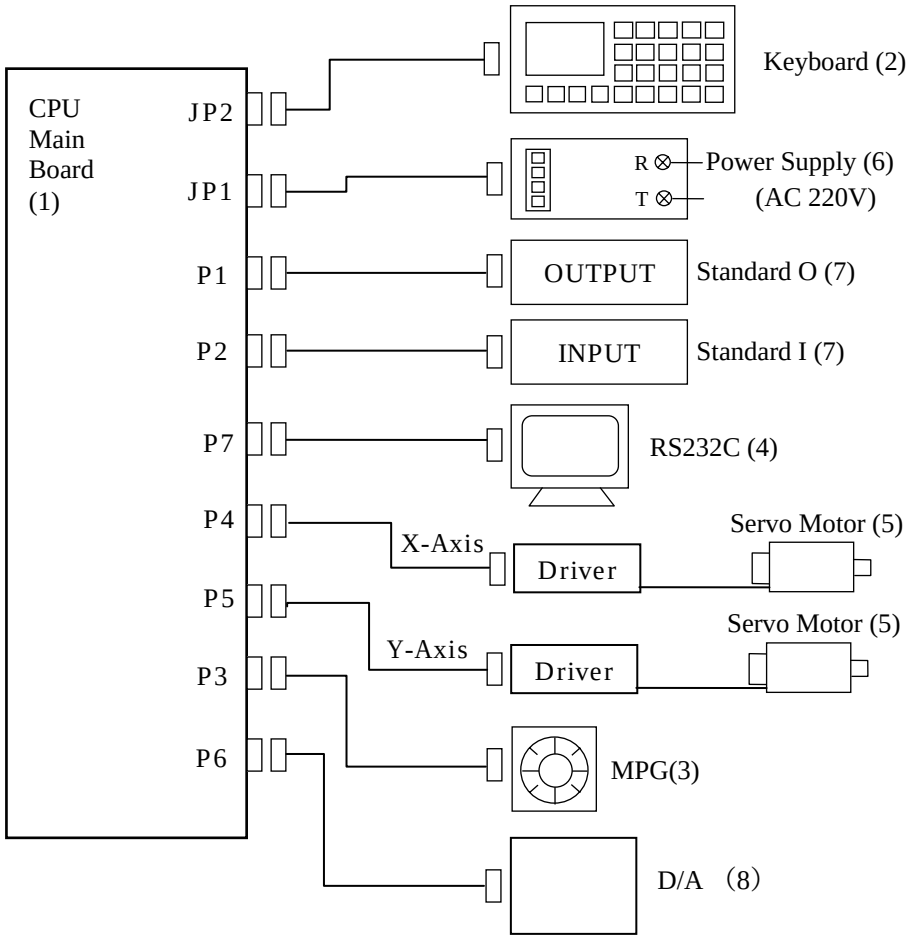


Fig 2-1 System Configuration

3 INSTALLATION AND DIMENSION

3.1 Environmental Requirements

The design of the cabinet to house the CNC unit, the keyboard panel and the LCD display is the responsibility of the machine tool builder. The cabinet must be constructed and the unit installed in an environment satisfying the following conditions.

- Ambient Temperature
  When operating: 0^{o} C to 45^{o} C.
  When storing or transporting the system: -20^{o} C to 55^{o} C.

- Temperature Variation
  Maximum: 1.1^{o} C/minute.

- Humidity
  Normally: 80% RH or less (Relative Humidity).
  For short period: maximum 95% RH.

- Vibration
  When operating: <= 0.075 mm with frequency 5 HZ.

- Noise
  When operating: withstand 2000V/100 nsec pulse width every 10 msec.

- Consult us when installing the system in an environment with high degree of dust, coolant, or organic solution.

3.2 Cabinet Considerations

Pay attention to the following items when designing a cabinet.

- The cabinet must be constructed to house the CNC unit, MDI panel and is always a completely enclosed box.

- Temperature rise within the cabinet should be 10^{o} C or less than the ambient temperature.

- Sealing of the cable inlet/outlet or doors must be done thoroughly.

- To avoid the noise interference, various units, the connecting cable and the AC power source must be mounted at least 100 mm apart. In a magnetic field environment, this distance should be increased to at least 300 mm.

- When mounting the servo amplifier, please refer to the Servo Motor Manual.

3.3 Thermal Design in the Cabinet

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

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

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

Example 1 (with a cooling fan):

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

Example 2 (without a cooling fan):

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

3.4 External Dimensions and Diagrams

- HUST H2N Keyboard Panel and LCD Screen

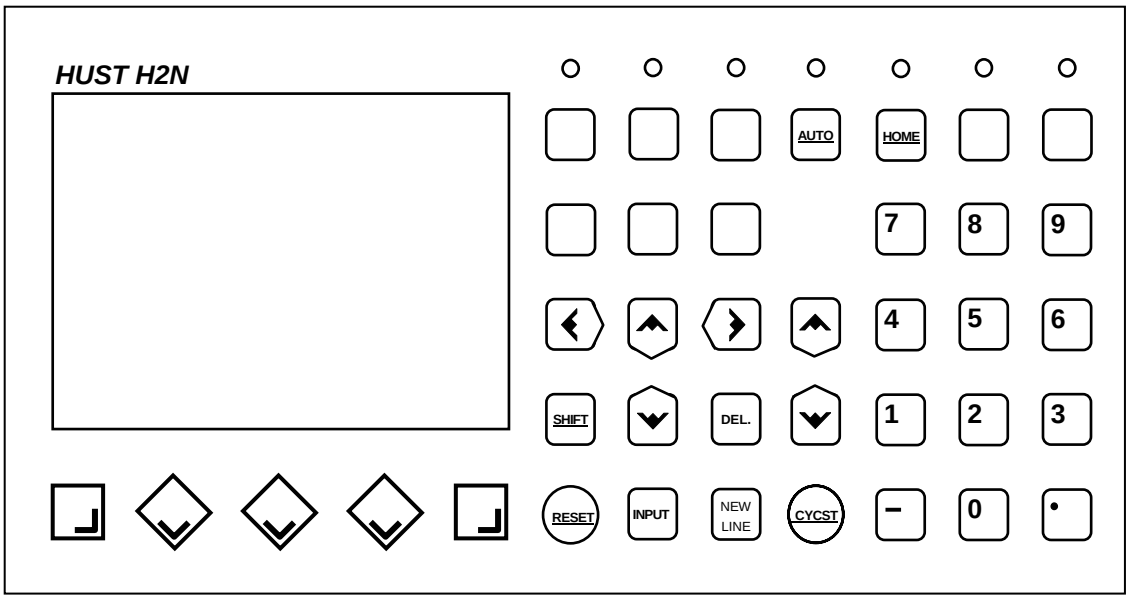
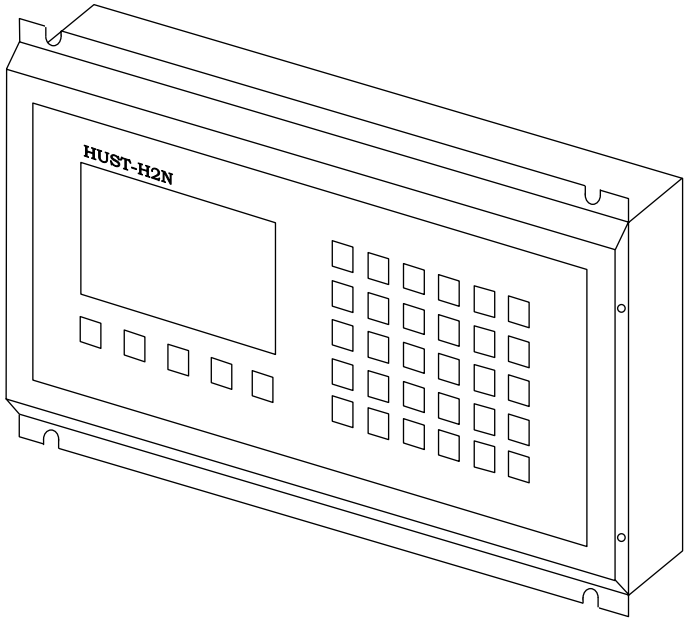


Fig 3-1 HUST H2N Keyboard Panel and LCD Screen



- CPU Main Board Connectors (Rearview)

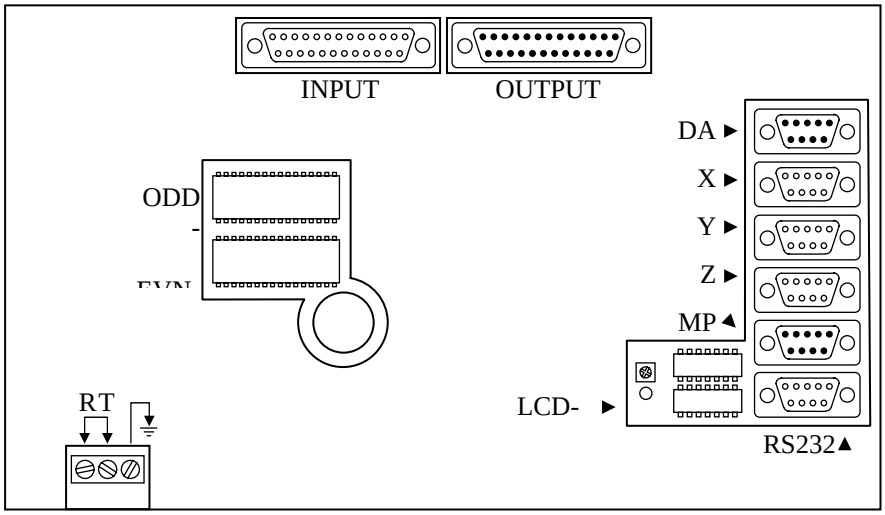


Fig 3-2 Main Board Connectors

- Dimension for H2N Controller Box (Rearview)

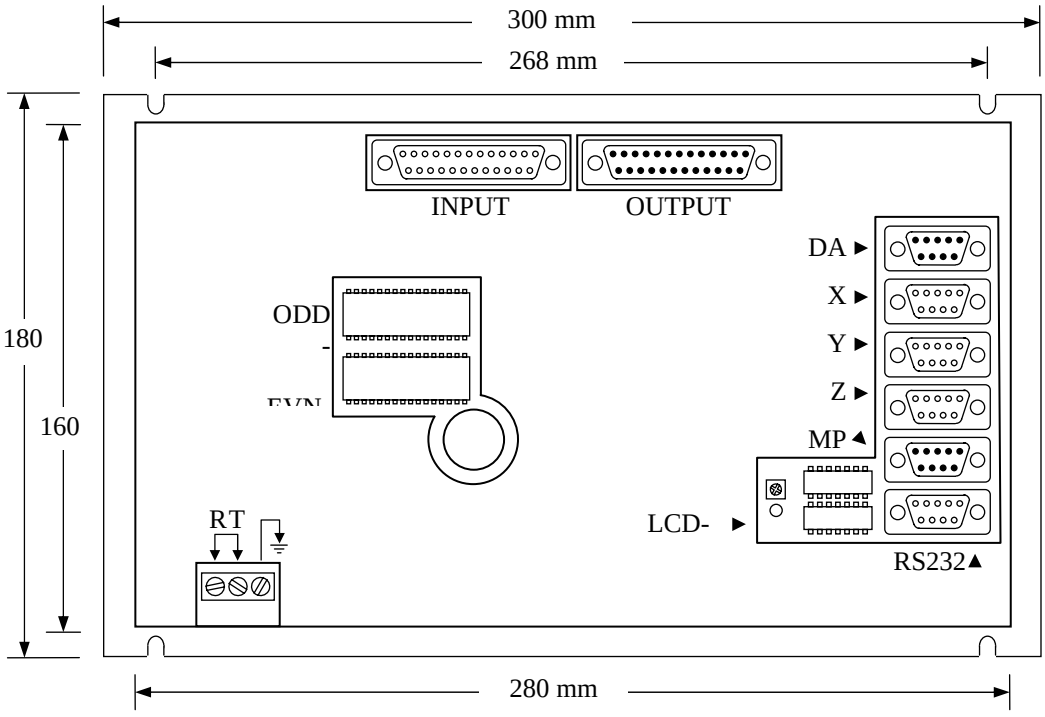


Fig 3-3 Dimension for HUST H2N Controller Box (Rearview)

- Dimension for H2N Controller Box (Top View)

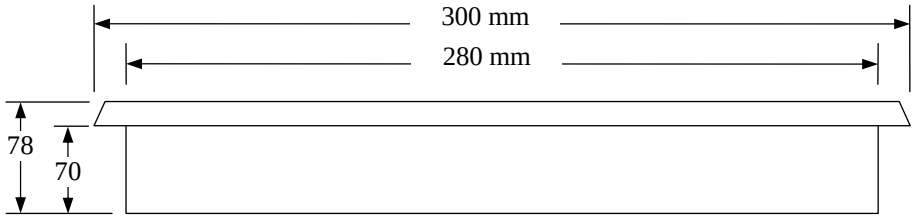


Fig 3-3 Dimension for HUST H2N Controller Box (Top View)

- Cutout Dimension for H2N Controller Installation

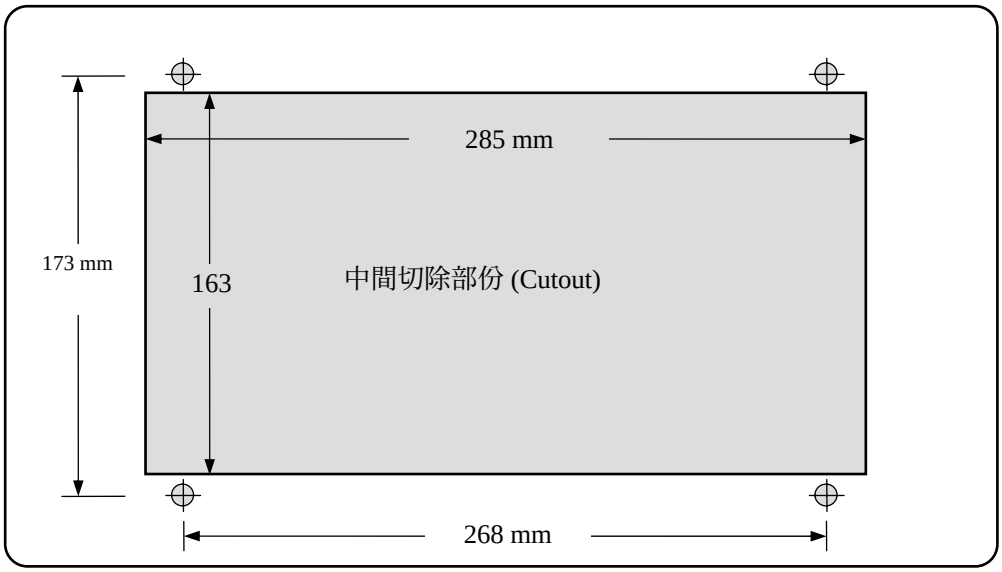


Fig 3-5 Cutout for H2N controller Installation

- Input Board – NPN Type

  Fig 3-6 is a standard NPN type input board. This board provides 24 input points with LED indicators. When the input signal is correctly received, the corresponding LED will be ON. Otherwise, check your program or connections.

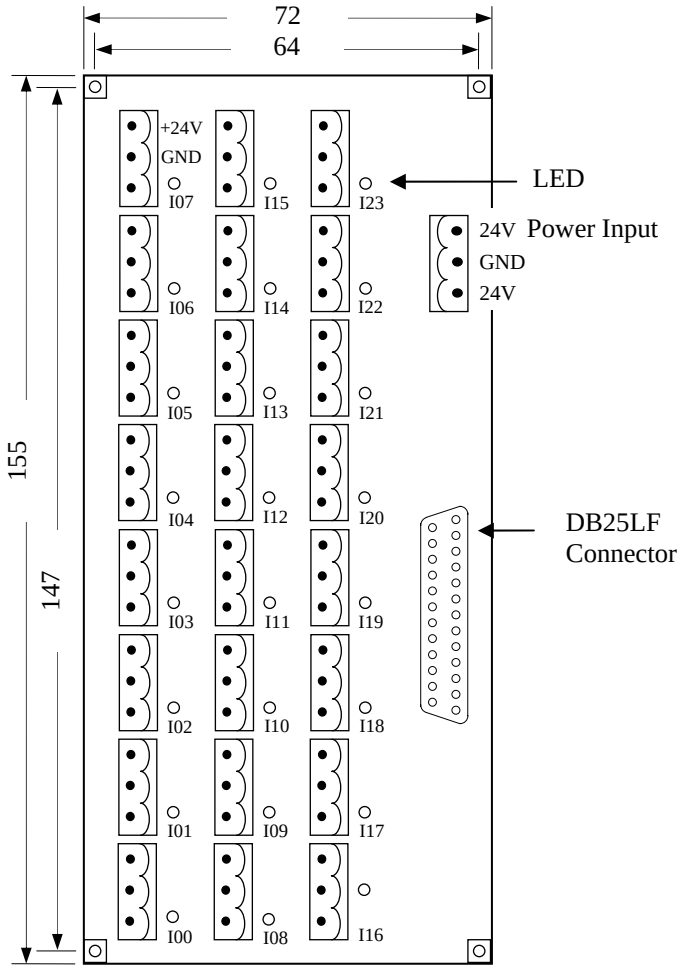
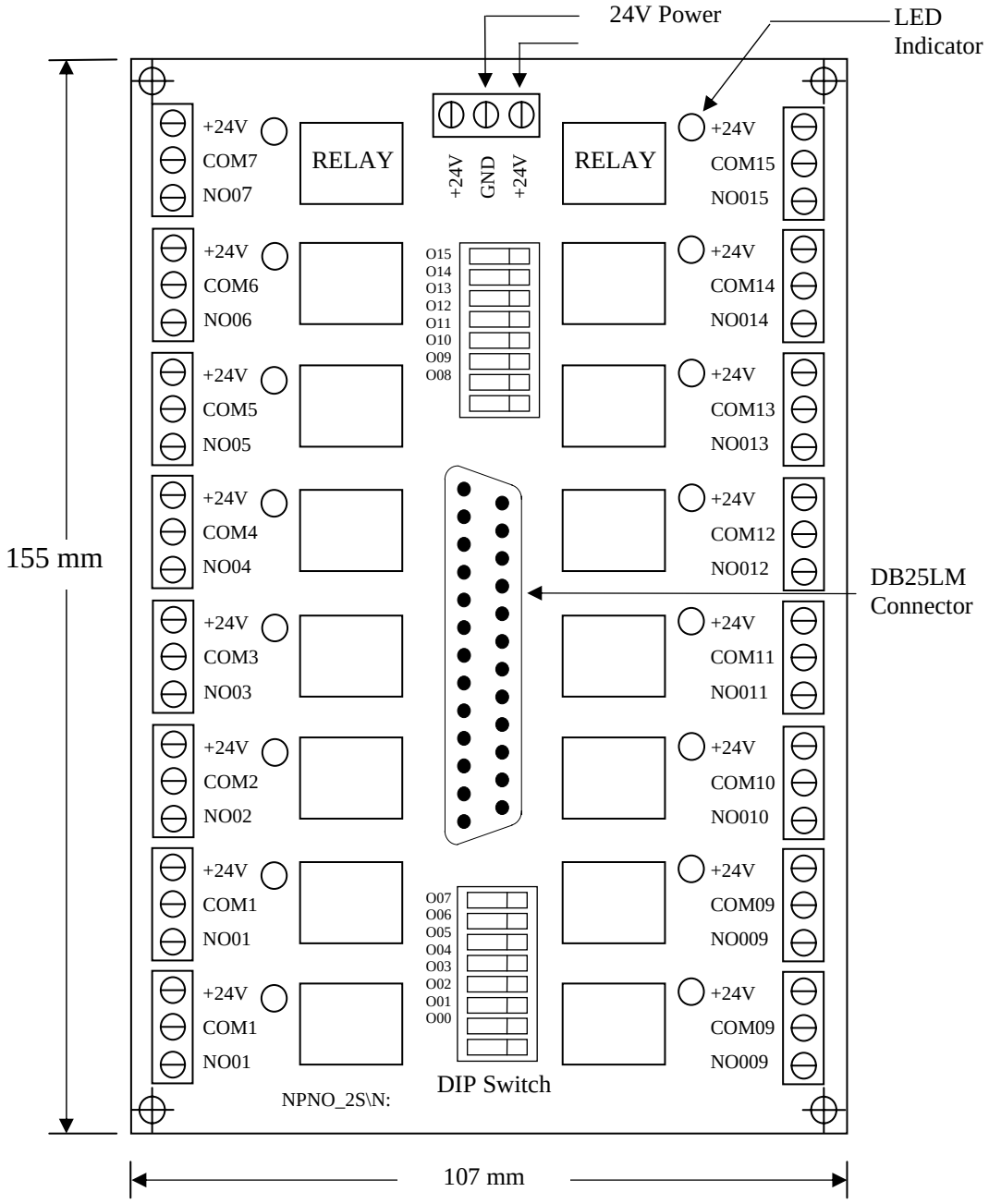


Fig 3-6 HUST Input Board – NPN Type

- Output Relay Board – NPN Type

  Fig 3-7 is a NPN type output relay board and it has 16 outputs with LED indicators. When the output signal is properly received, the corresponding LED will be ON.

  Normally, all DIP switches are at "OFF" position. In this case, all connections on COM and NO will be done in normal manner. If the DIP switch is at “ON” position, COM becomes the ground (GND) of 24V. If you have an external device, such as relay valve or DC motor, that is driven by 24V ground, just push the corresponding DIP switch to “ON” and no 24V grounding connected to COM is necessary.



NO = Normal Open, COM = Common

Fig 3-7 Output Relay Board

4 CONNECTING DIAGRAMS

4.1 Connector Type

The types of connectors provided on the HUST H2N controller box are summarized below. Each connector symbol is followed by a letter of either M (for male) or F (for female).

DB9LF (or M) : D sub type connector 9 pins female (or male).
DB25LF (or M) : D sub type connector 25 pins female (or male).
TBxx : Terminal block with pin number indicated by “xx”

4.2 Connector Designation

The connector designation on HUST H2N controller in printed on the back of the controller and their corresponding types are listed below.

Connector Name	Connector Designation	Type
Output Interface	P1	DB25LM (Male)
Input Interface	P2	DB25LF (Female)
MPG Handwheel	P3	DB9LM (M)
X-axis Servo	P4	DB9LF (F)
Y/Z-axis Servo	P5	DB9LF (F)
D/A Output for Spindle	P6	DB9LM (M)
RS232C Interface	P7	DB9LF (F)
Main Board DC Power	JP1	TB10
Keyboard Connector	JP2	TB20
LCD Connector	JP3	TB20
220V AC Power	R和T	TB3

Table 4-1 Connector Designation and Type

4.3 System Connecting Block Diagram (Fig 4-1)

The RS232 communication cable should not exceed the 15m maximum length. This is to avoid the potential electrical noise due to the electro-magnetic interference.

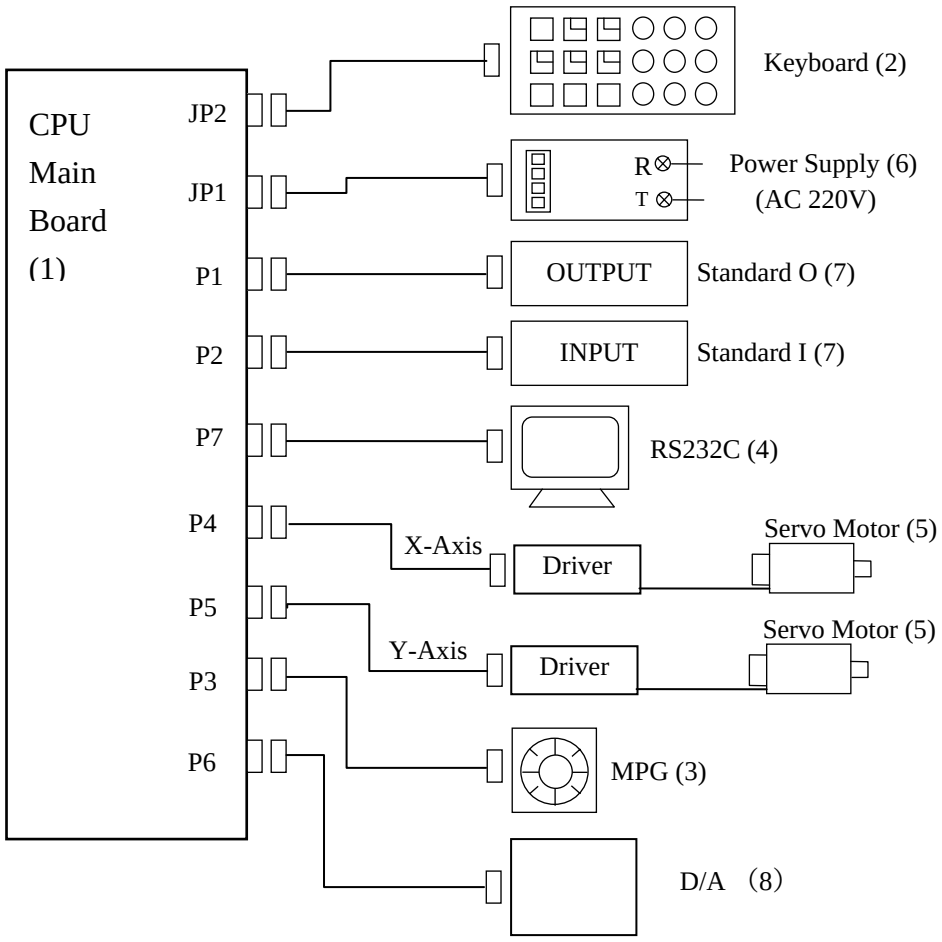


Fig 4-1 System Connecting Diagram

- **AC Power Supply Connecting Diagram (Fig 4-2)**

HUST H2N controller requires an AC110~220V± 10% power. Try to avoid excessive power fluctuation or it will not work properly. All components in Fig 4-2 are inside the control box and it's shown here for reference only.

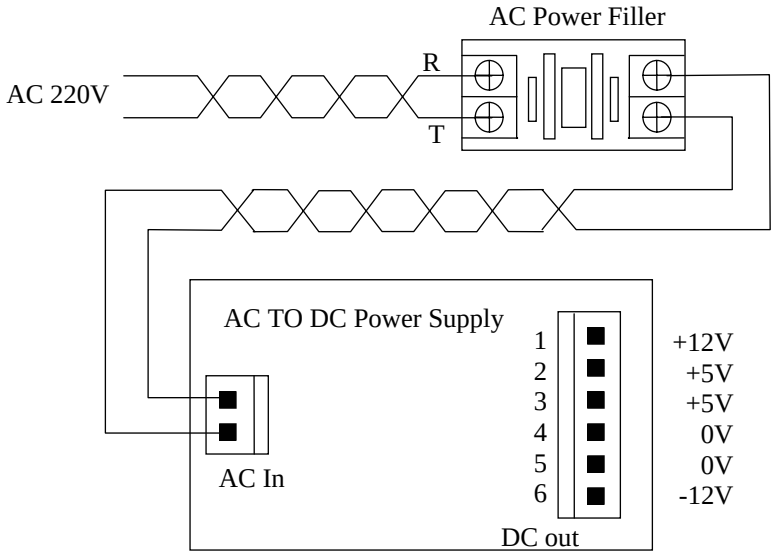


Fig 4-2 AC Power Supply Connection

- **DC Power Supply Connecting Diagram (Fig 4-3)**

The DC power connections have been done in the factory. In the case that it becomes necessary to reconnect yourself, please use Fig 4-3 for reference.

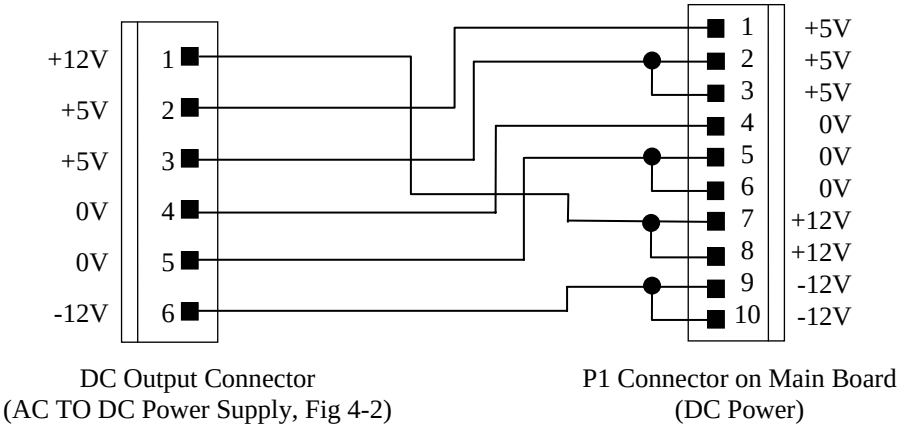


Fig 4-3 DC Power Supply

- **Motion Control (Servo/MPG) Connection (Fig 4-4)**

  X-axis servo driver is connected to P4, Y-axis to P5 and MPG hand-wheel to P3 as shown in Fig 4-4. The connections may vary depending on the type of servo system and the MPG that customer uses.

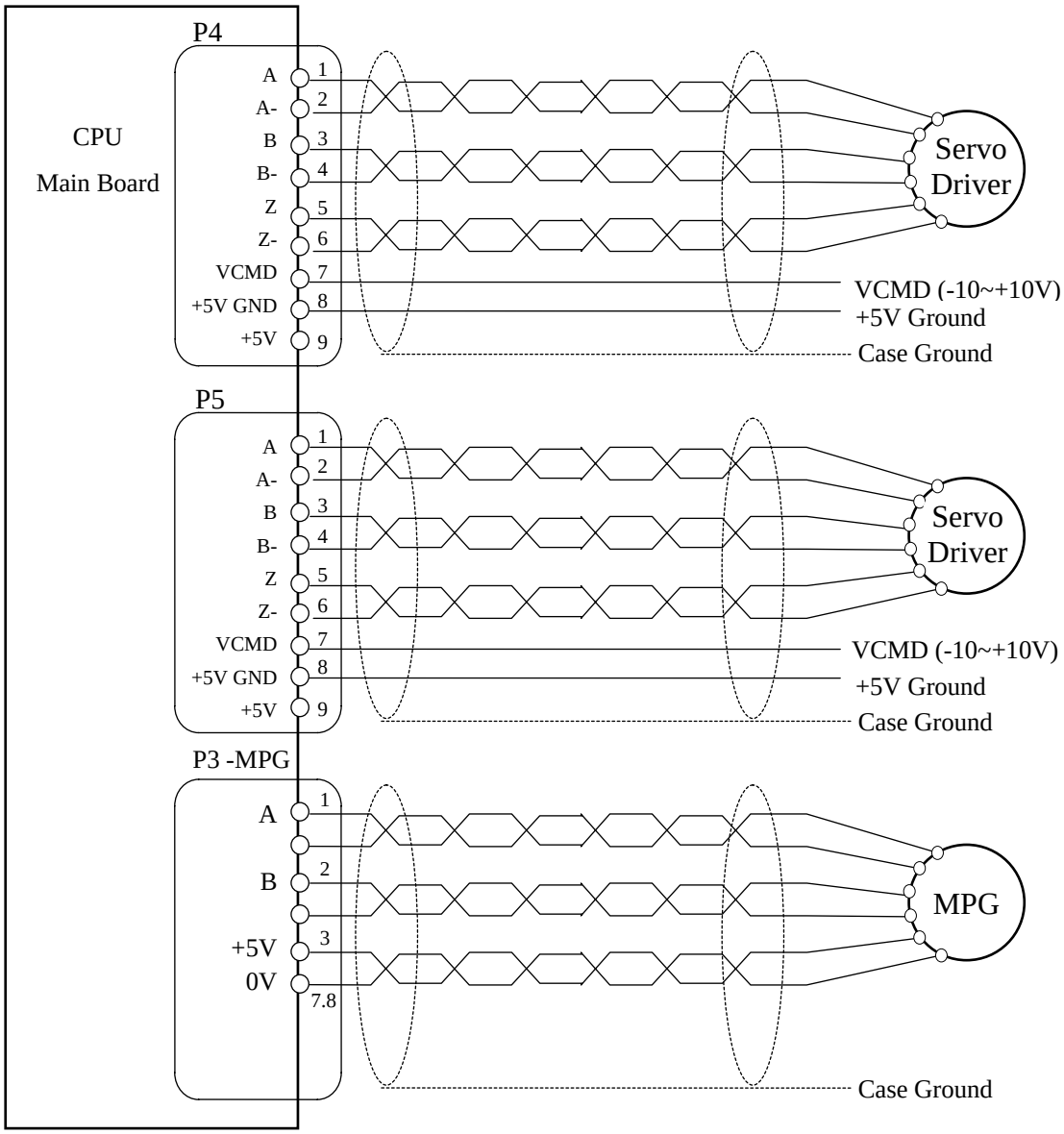


Fig 4-4 Servo and MPG Unit Connection

- **AC Power System Connection (Fig 4-5)**

HUST internal PLC in the controller has a timer to control the servo-on delay. When the set time (about 1.5 seconds) in this timer has elapsed, an output is ON to turn on the servo motors.

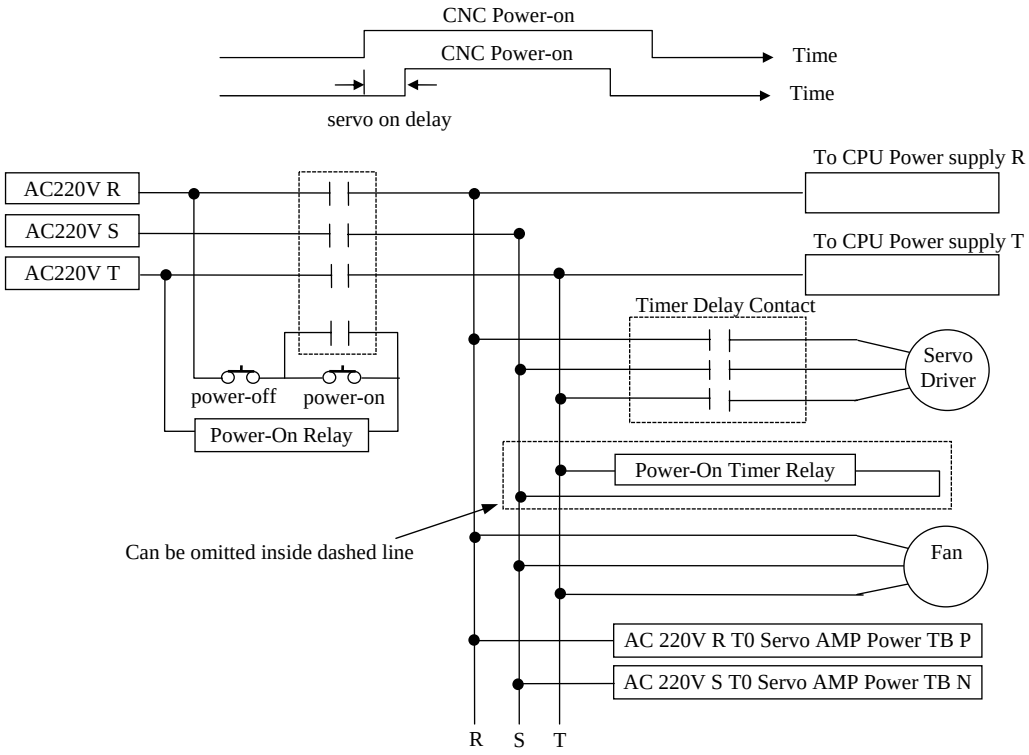


Fig 4-5 AC Power Connection to Servo Driver

- **MPG Hand-wheel Connecting Diagram (Fig 4-6)**

If the tool movement is in the opposite direction to the MPG hand-wheel rotation, please exchange A and B signal connections.

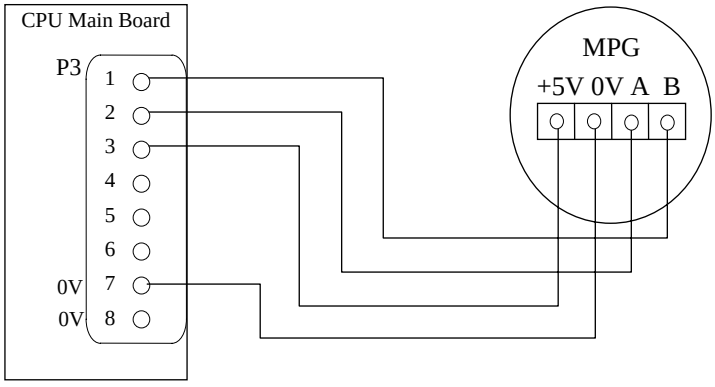


Fig 4-6 MPG Hand-wheel connection

- **CONTRAST, R7**

  This is an adjustable resistor for LCD screen contrast adjustment.

- **D/A Connector**

  This connector is for special application and it needs an IC for it to be effective. In addition, the PLC should includes C100 = 1 and the registers in PLC are R152 and R153.

  R152 = Input signal = 0 ~ 10V
  R153 = Output signal = -10.000V ~ +10.000V
  The decimal format for R153 is 2 integer with 3 decimal points.
  If R153 = 05200, it means 5.2 volts.

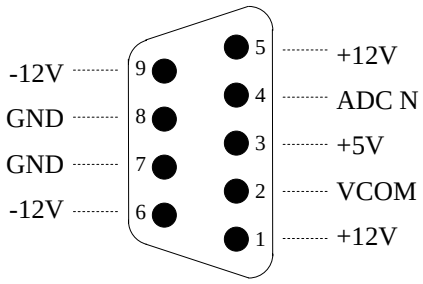
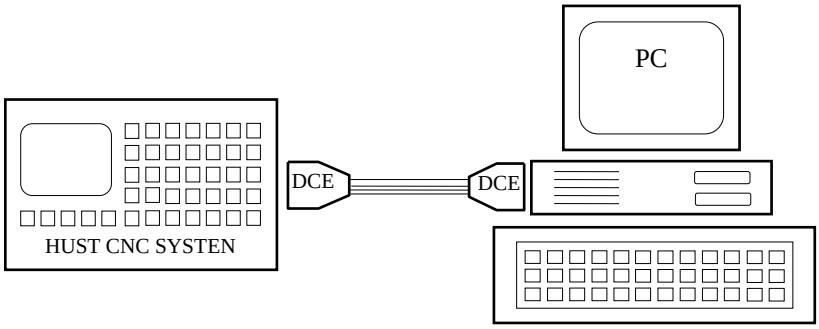


Fig 4-7 D/A Connector

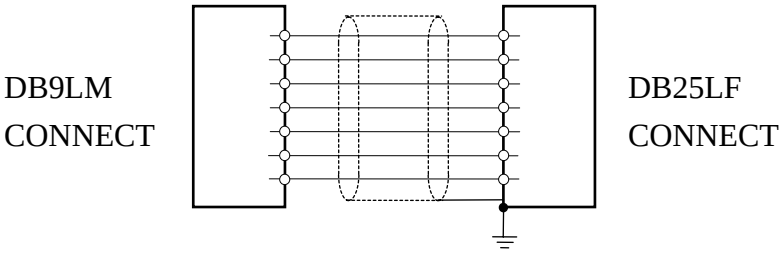
- **RS232 Connector Pin Assignment and Connecting Diagram (Fig 4-7)**

  For the operational details of RS232 communication, please refer to Chapter 9 of HUST H2N Operation Manual.

  1. Do not exceed 15 meters for the cable connecting RS232 port and PC.
  2. The interface voltage on the PC side should be in the range 10 ~ 15V.
  3. Try NOT to work in an environment where electrical noises are common. EDM machine and electric welder are noise generators. Do not use the same power source as EDM machine or electric welder.



Controller Side PC Side - COM2



Controller Side PC Side – COM1

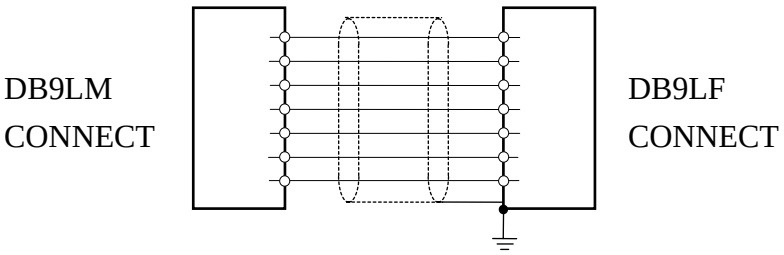


Fig 4-8 RS232 Connector Pin Assignment

5 I/O INTERFACE CONNECTIONS

5.1 I/O Connector Pin Assignment

The pin assignment of 25-pin connector is shown in Fig 5-1. This assignment is standard in HUST controller and can be applied to those connectors on main board, input board, or output relay board. The connector designation on H2N controller is shown in Table 5-1. Note that the input connector has 24 I-points while output connector has 16 O-points.

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

Connector Designation	Connector Type	I/O Pin Assignment
P1	DB25LM (Male)	O 000 ~ O 015
P2	DB25LF (Female)	I 000 ~ I 023

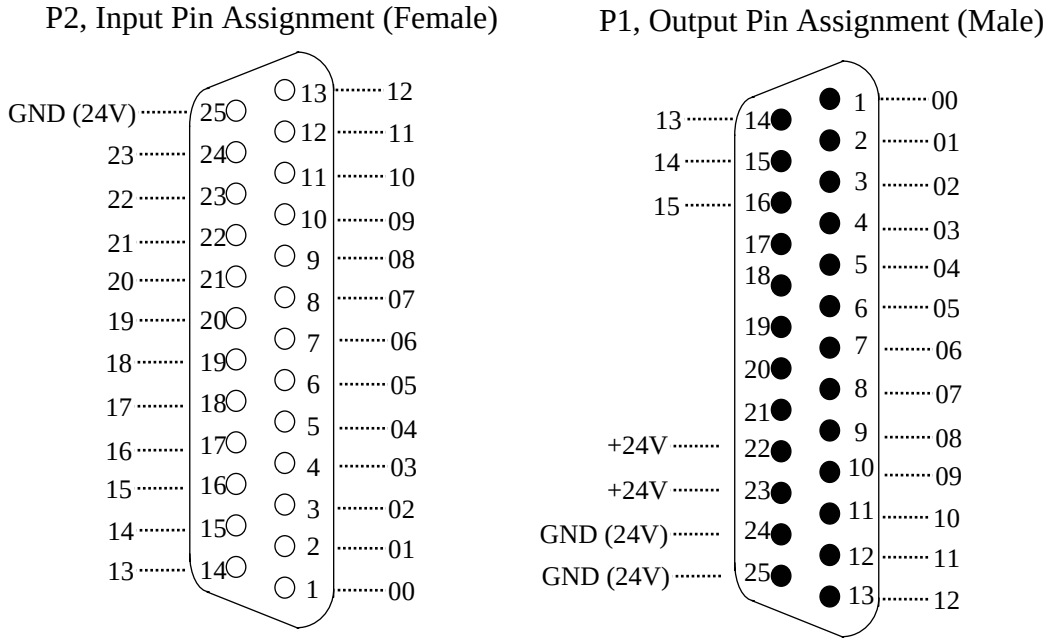


Fig 5-1 I/O Connector Pin Assignment (NPN-type)

I/O signals can be connected directly to P1 and P2 connectors on H2N controller. All output points on HUST H2N main board are of transistor type circuit with open collector. However, it would be advisable to use input board and output relay board before the signal going into the controller. This provides additional protection for main board on CNC from being damaged by electrical surge.

5.2 Input Board and Output Relay Board

The input signal connection can be done in two ways:

1. Connect directly to DB25LF connector in the controller shown in Fig 5-1.

2. Connect input signal to the corresponding terminal of HUST input board, then connect from DB25LF connector on the board to the connector on the controller box. HUST input board is shown in Fig 5-2. The advantage of using input board is to protect the controller circuits from being damaged in case of short circuit. This method of connection is suitable for NPN-type input connector only.

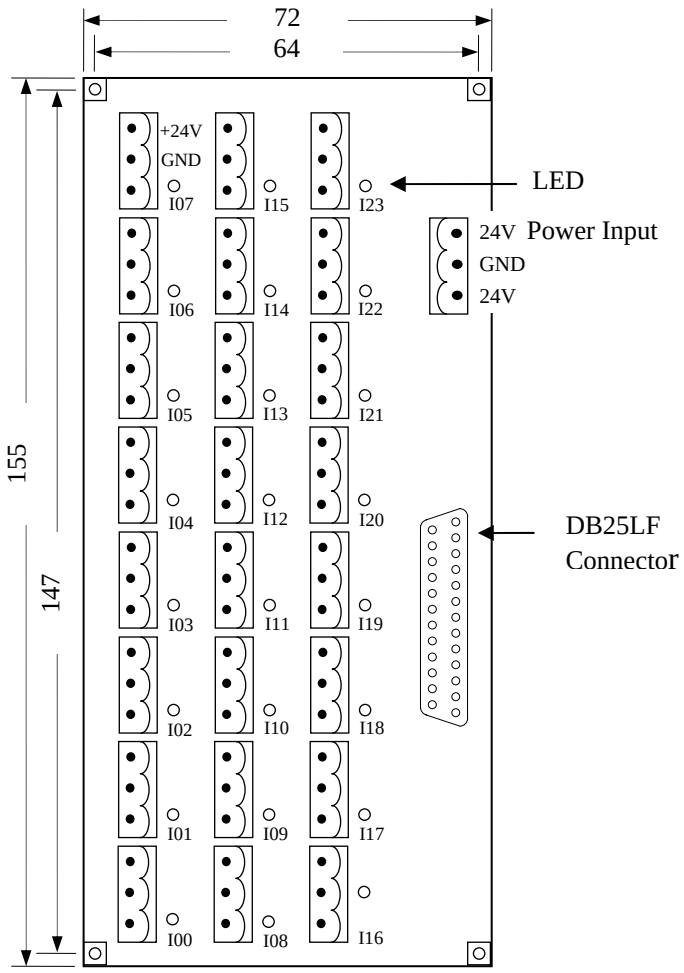


Fig 5-2 HUST Input Board (NPN-type)

The output signal connection can also be done in two ways:

1. Connect directly to DB25LM connector in the controller shown in Fig 5-1.

2. Connect output signal line from external device to the corresponding terminal of output relay board, then connect from DB25LM connector on the board to the connector on the controller box. HUST output board is shown in Fig 5-3. The

advantage of using output relay board is to protect the controller circuits from being damaged in case of short circuit. This method of connection is suitable for NPN-type output connector only.

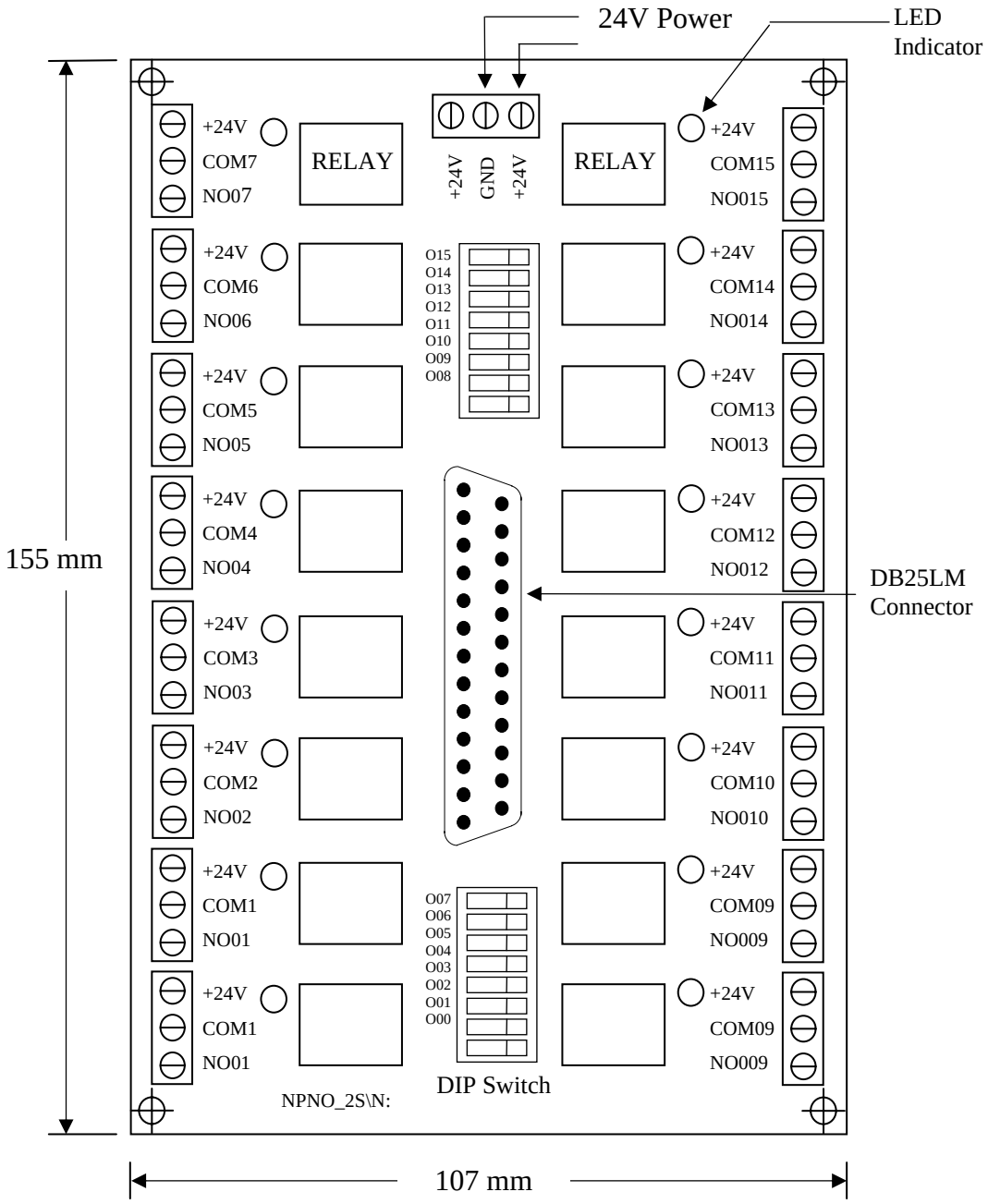


Fig 5-3 HUST Output Relay Board (NPN-type)

5.3 Input Signals

Input signal is the one from external devices to CNC controller, such as push button, sensor, limit switch, proximity switch, etc. There are 24 inputs for HUST H2N controller and their status can be displayed on LCD screen by pressing the blank key (below AUTO key) on the keyboard.

The input signal I007 is specially reserved for G31 skip function. All other 23 input signals and 16 outputs are free for customer's use.

- Input Signal Specification : The voltage to drive the input signal is 24V, but the signal line is connected to the "sink" end (0V). The current is 8 mA.

- Input Signal Directly Connected to CNC (NPN-type)

  Input signal bypasses the input board and is directly connected to P2 connector on CNC.

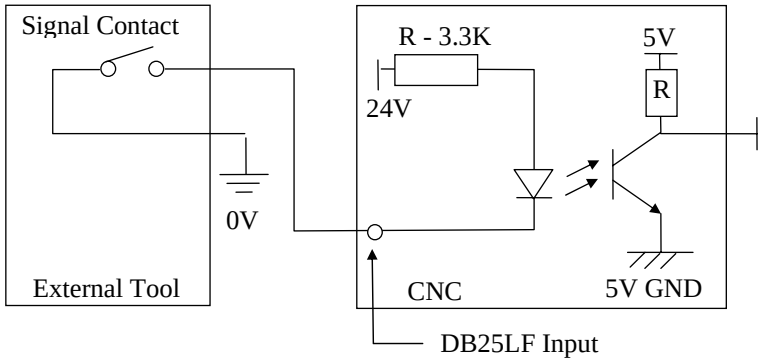


Fig 5-4 Input Signal Directly Connected to CNC (NPN-type)

- Input Signal Connected to Input Board then to CNC (NPN-type)

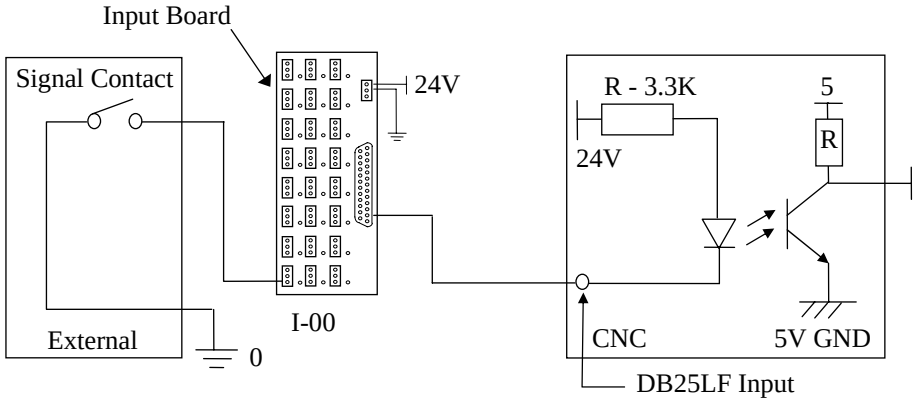


Fig 5-5 Input Signal Connected to Input Board (NPN-type)

5.4 Output Signals

Output signal is the one from CNC controller to an external device. The output circuit of HUST H2N controller is a transistor circuit that can be used to drive a relay or LED, etc.

- Output Signal Specification:

a) With I/O relay board, each connection point on the board can withstand a maximum of AC 250V and 10A current.
b) Without I/O relay board, the output transistor circuit of HUST H2N controller is rated at a maximum of 35V and 500 mA of current.

- Output Signal Directly Connected from CNC to Tool (NPN-type)

When output connector is directly connected to an inductive load, such as relay, it must connect a spark killer diode in parallel and as close to the load as possible. (See Fig 5-6) ll output terminal of HUST H2N controller is open collector transistor circuit.

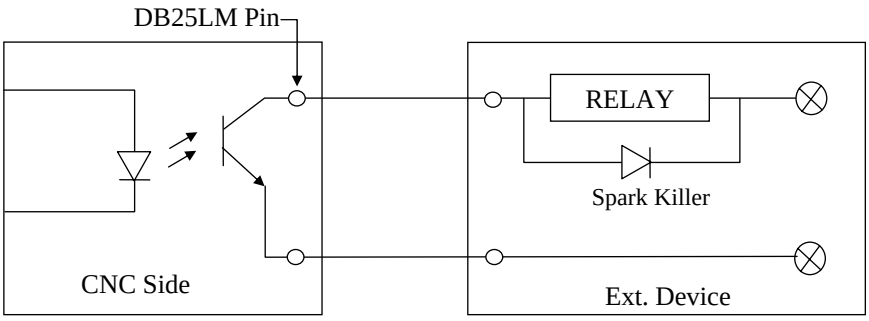


Fig 5-6 Output Signal from CNC without Relay Board (NPN-type)

- Output Signal Connected to Output Relay Board (NPN-type)

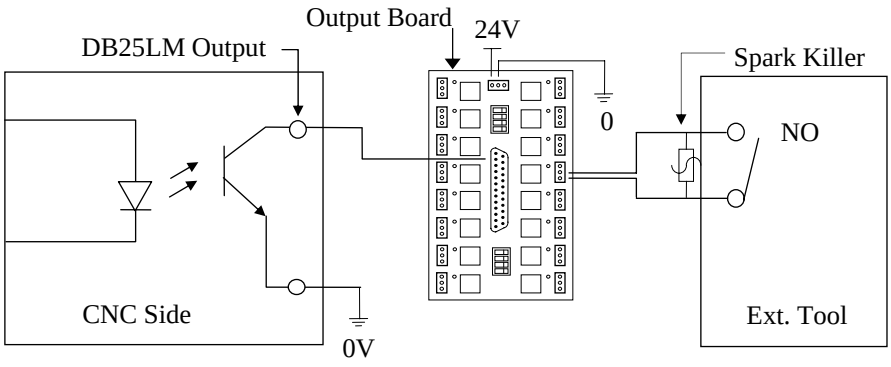


Fig 5-7 Output Signal Connected to Relay Board (NPN-type)

5.5 Emergency-Stop Circuit

The electronic circuit in the controller requires about 100 milli-seconds to reach stable state when power is turned on. During this unstable period, the servo motor should not be turned on. To accomplish this purpose, HUST internal PLC has a "Servo-on Timer" with 1.5 seconds. When the timer is up, the controller sends an O-bit high (1) to turn on the servo driver.

Fig 5-8 is an emergency stop circuit. When the hardware OT limit switch is touched or the E-stop button pushed, the controller will be in a state of emergency stop and the servo-on switch on the servo motor will trip off. For safety reason, the E-stop button, the hardware OT limit switch, and the E-stop relay should be connected in-series and the servo-release button in-parallel with the hardware OT limit switch. Please note that the E-stop button and the hardware OT limit switch are of normal-closed type.

During normal operating condition, the servo-release button is at OFF position. After emergency stop has occurred and all power to servo motor has been cut off, servo-release button can be pushed in to supply power to servo motor to bring the machine tool to normal operating range.

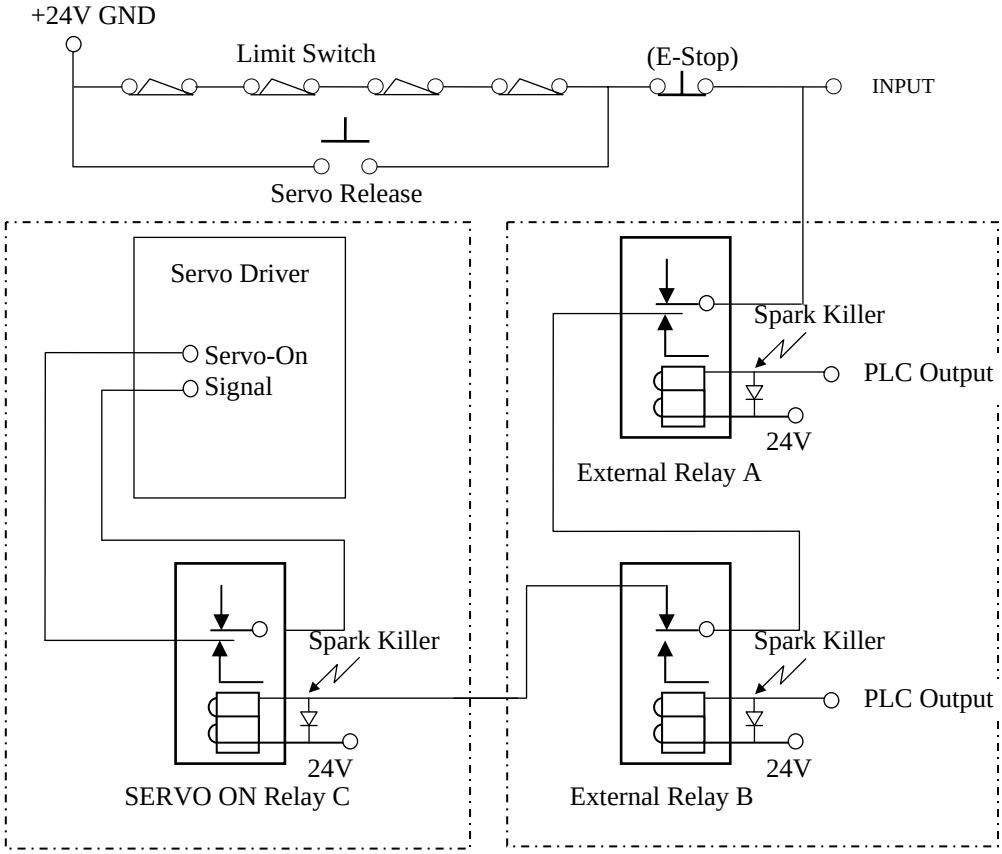


Fig 5-8 Emergency Stop Circuit

Notes on Emergency Stop Circuit (Fig 5-8):

1. Relays A and B are protection circuit during power-on. The purpose is to prevent the tool from being damaged in case of power-on failure that results in a total output.
2. Relay C is a servo-on relay.
3. Generally, Relay C has multiple connecting points for multiple servo. Each servo motor requires independent connecting point. Do NOT connect servo motors in parallel because some servo motors cannot be connected in parallel. See Fig 5-7 and Fig 5-8.

Incorrect Connection:

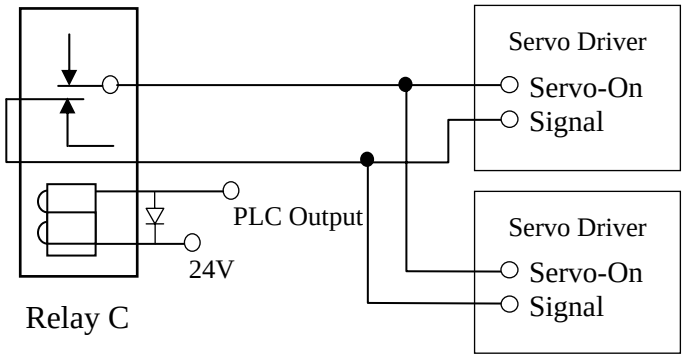


Fig 5-7 Incorrect Connection

Correct Connection:

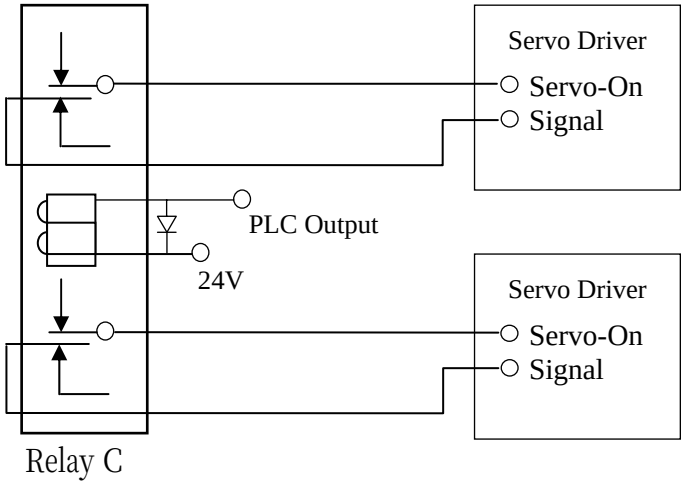


Fig 5-8 Correct Connection

4. Relays A and B are connected to two (2) outputs from PLC. Example of servo-on in PLC is shown in Fig 5-9 below.

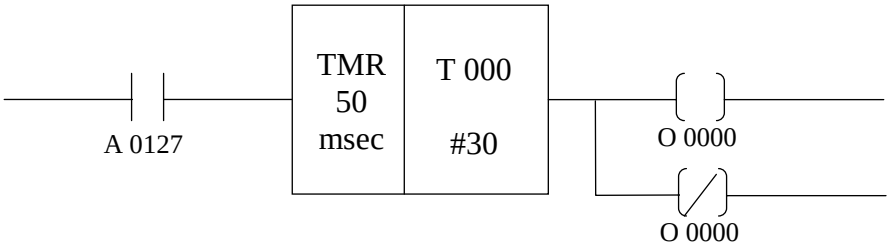
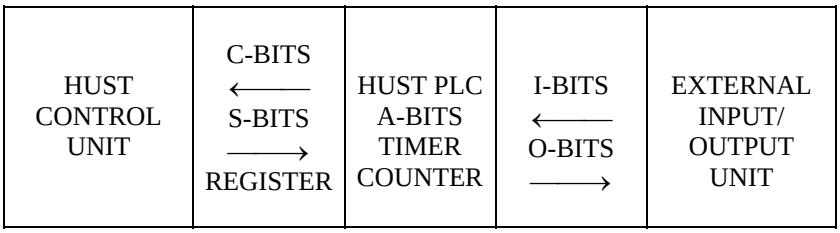


Fig 5-9 Servo-on arrangement in PLC Program

6 INTERFACE BETWEEN CNC CONTROLLER AND PLC

PLC bits used in HUST H2N controller with PLC (Programmable Logic Controller) are described below. Their meanings and applications will be discussed in this chapter. This chapter is written for those who are familiar with HUST PLC program editing.

S-bit : Status signal from CNC controller to PLC unit.
C-bit : Command signal from PLC unit to CNC controller.
I-bit : Input signal from external I/O unit to PLC unit.
O-bit : Output signal from PLC unit to I/O unit for driving external device.
A-bit : Auxiliary bit for PLC unit.
Register : (1). R000~R099 are for user's application in the PLC program.
(2). R100~R255 are reserved for HUST H2N system data, NOT for customer's use. The data in R000 ~ R199 can be saved with power off if MCM #86=256, and the data will NOT be saved if MCM #86=0.



When I-bit (any I-bit) is used for G31 Skip function, the input signal can be directly fed into the controller. Otherwise, all I-bit signals will be processed through the PLC unit before going into the controller.

6.1 HUST H2N Series Status Signal S-Bits

S-bit is the internal status signal from HUST CNC system unit to PLC unit for sequential control of the machine tool. The meaning of each S-bit signal is shown in Table 6-1. Note that S000 to S031 are one-shot bits and are effective for one cycle only. For example, when RESET key is pressed, S001=1 for one cycle.

Table 6-1 HUST H2N Status Code S-Bit Signals

S-Bit Number	Status Signal Description
S000	Send out square wave signal with 0.75 sec ON, 0.25 sec OFF
S001	Reset key
S002	Cycle start key
S003 ~ S007	Reserved
S008	Send out square wave signal with wave period = 0.10 sec.
S009	Send out square wave signal with wave period = 0.25 sec.
S010	Send out square wave signal with wave period = 0.50 sec.
S011	Send out square wave signal with wave period = 1.00 sec.
S012 ~ S015	Reserved
S016	Error code strobe
S017	Key strobe
S018	Key strobe for key being pressed twice in 0.5 sec
S019	Reserved
S020 ~ S021	Cursor up key, Cursor down key
S022 ~ S023	Page up key, Page down key
S024	M-code strobe
S025	T-code strobe
S026	S-code strobe for spindle speed
S027 ~ S063	Reserved
S064	X-axis motor running
S065	Y-axis motor running
S066 ~ S079	Reserved
S080	Program execution in process
S081 ~ S087	Reserved
S088	X/Y-axis Servo motor in motion
S089 ~ S103	Reserved
S104 ~ S111	Status signals from PC or MMI (Man Machine Interface)
S112 ~ S119	Reserved
S120	Battery BT1 down (Flash Rom)
S121	Battery BT2 down (Flash Rom)
S122 ~ S125	Reserved
S126	Key strobe for key being pressed down and kept down
S127	Signal for ladder simulation
S128 ~ S159	Reserved
S160 ~ S161	Home signal for X and Y-axis respectively (Machine coordinate = 0)
S162 ~ S175	Reserved
S176 ~ S177	In-position signal for X and Y-axis when motor in motion.
S178 ~ S191	Reserved
S192 ~ S193	Direction of motor rotation (+/-) for X and Y-axis respectively
S194 ~ S207	Reserved
S208	X-axis traverse execution signal, when C208 0→1, S208 = 1
S209	Y-axis traverse execution signal, when C209 0→1, S209 = 1

6.1.1 S-bit Signal and PLC Cycle Time Chart

Fig 6-1 shows the time chart between the HUST H2N controller and the PLC unit. Note that Bit S017 is synchronized with Bits S000 ~ S023. S000 ~ S031 are all one-shot signals.

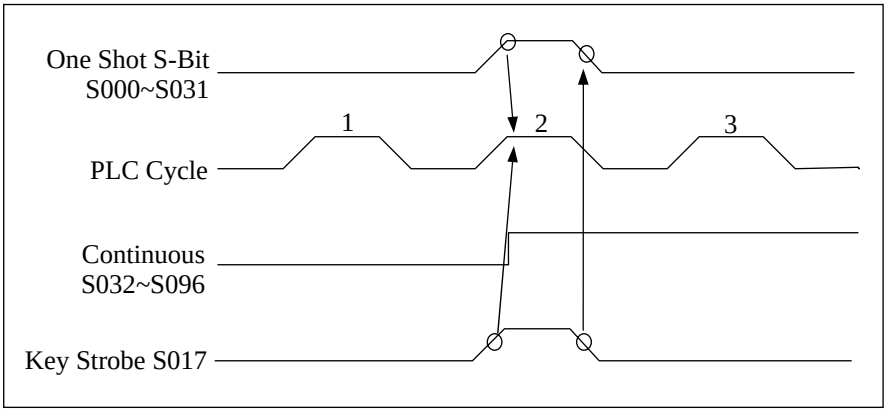
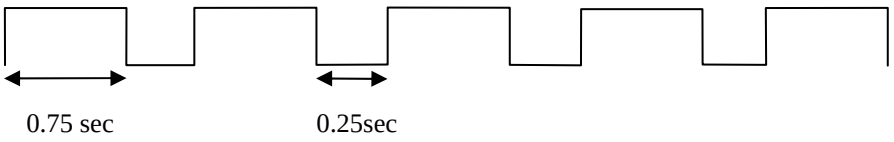


Fig 6-1 Time Chart Between HUST CNC and PLC

6.1.2 Descriptions of S-bits

- **Square Wave Signal with 0.75 sec ON, 0.25 sec OFF (S000)**



- **Reset Key Strobe (S001)**

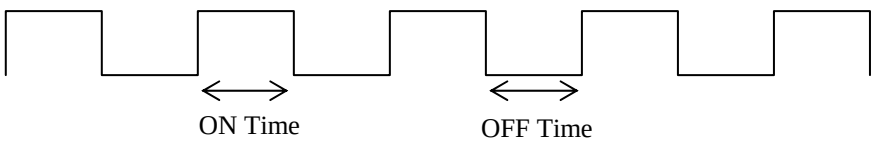
  The controller sends S001 one-shot high (1) strobe to PLC unit when RESET key is pressed.

- **Cycle Start Key Strobe (S002)**

  The controller sends S002 one-shot high (1) strobe to PLC unit when CYCST key is pressed.

- **Square Wave Signal with Wave Period (S008~S011)**

  S008, signal ON = 0.10 sec and OFF = 0.10 sec.
  S009, signal ON = 0.25 sec and OFF = 0.25 sec.
  S010, signal ON = 0.50 sec and OFF = 0.50 sec.
  S011, signal ON = 1.00 sec and OFF = 1.00 sec.



- **CNC Control Unit Alarm Signal (S016)**

  The controller sands Bit S016 one-shot high (1) strobe to inform PLC unit when any alarm occurs.

- **Key Strobe (S017)**

  The controller sets Bit S017 one-shot high (1) strobe to inform PLC unit when any key on HUST keyboard is pressed.

- **Key Strobe for Key Being Pressed Twice in 0.5 sec (S018)**

  When a key on the keyboard is being pressed twice within 0.5 seconds, the controller send S018 = 1 one-shot to PLC.

- **Cursor Up Key Strobe (S020)**
  Cursor Down Key Strobe (S021)

  The controller sets Bit S020 one-shot high (1) strobe to inform PLC unit when CURSOR Up key is pressed. Bit S021 will be one-shot high (1) when CURSOR Down key is pressed.

- **Page Up Key Strobe (S022)**
  Page Down Key Strobe (S023)

  The controller sets Bit S022 one-shot high (1) strobe to inform PLC unit when PAGE Up key is pressed. Bit S023 will be one-shot high (1) when PAGE Down key is pressed.

- **M-, T- and S-Code Strobe (S024~S026),**

  When an M-code is executed in a part program, S024 will be set one-shot high (1) strobe.
  When a T-code is executed in a program, S025 will be set one-shot high (1).
  When an S-code is executed in a part program, S026 will be set one-shot high (1).

  Note that the controller system defined M-codes, such as M00, M01, M02, M30, M98 and M99 will not yield an M-code strobe (S24) but the user defined M-code will.

- **Servo Motor Status (S064~S065, S088)**

  When one of the 2 motors in any axis or 2 motors is in motion, bit S088 is set to high (1). To determine which motor is in motion, check S064~S065.

  S064=1, X-axis motor running.
  S065=1, Y-axis motor running.

- **In Process Status Bit (S080)**

  Whenever HUST control unit is executing a part program including option stop (M01), Bit S080 is set high (1) until the program end (M02, M30, M99).

- **Spindle Status Bit (S085)**

  In spindle closed loop feedback (C117=1), S085 = 1 indicates spindle is in a state of STOP and S085 = 0 means spindle is in motion.

- **Servo Motor Status (S088)** – See S064, S065 above

- **Key Signals from PC or MMI (Man Machine Interface) (S0104~S111)**

  When the special key that was designed on PC or MMI is pressed, a signal through RS232 is sent to the controller which will send the status bits S104 ~ S111 to PLC.

- **Battery (BT1, BT2) Down Signal (S120~S121)**

  When using FLASH-ROM in the controller, S120~S121 is used to check the battery condition.
  If the battery BT1 is down, the controller displays ERROR30.1 and S120=1.
  If the battery BT2 is down, the controller displays ERROR30.2 and S121=1.
  When both batteries are down, the controller displays ERROR 30.A and S120=S121=1.

- **Key Strobe for Key Being Pressed Down and Kept Down (S126)**

  When a key is pressed down and kept down, the controller will send S126 = 1 to PLC. When key is released, S126 = 0.

- **Ladder Simulation Mode (S127)**

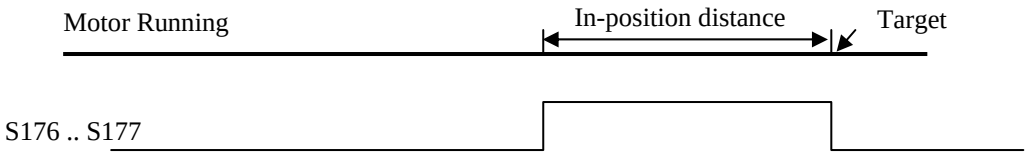
  The S127 bit will be set high (1) when a ladder program is being simulated from a PC.

- **X and Y-axis Home Signal at Machine Coordinate = 0 (S160~S161)**

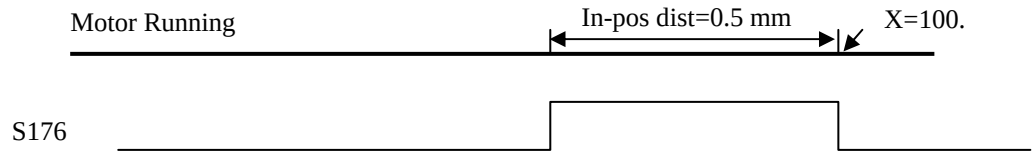
  When the tool is at X-axis home location (machine coordinate X=0), S160 = 1.
  When the tool is at Y-axis home location (machine coordinate Y=0), S161 = 1.

- **In-position Signal for X and Y-axis (S176~S177)**

  When tool moves to a point inside in-position distance on X-axis as set in MCM #87, S176=1. When tool comes to the X-target location, S176=0.
  When tool moves to a point inside in-position distance on Y-axis as set in MCM #88, S177=1. When tool comes to the Y-target location, S177=0.



Example: MCM #87 = 500 = 0.5 mm
Position command: G01 X100.000



- **Direction of Motor Rotation (+/-) for X and Y-axis (S192~S193)**

S-bit	Axis	Positive Rotation (+)	Negative Rotation (-)
S192	X	0	1
S193	Y	0	1

- **Positioning Execution in PLC (S208~S209)**

  When C-bit C208 = 0→1 (Rising edge), the tool on X-axis will move to a location as set by R180x10 with a speed as set by R160. S208=1 when tool in motion, S208=0 when motion completed.

  When C-bit C209 = 0→1 (Rising edge), the tool on Y-axis will move to a location as set by R181x10 with a speed as set by R161. S209=1 when tool in motion, S209=0 when motion completed.

Table 6-2 C-bit Signals Description

Bit Number	C-Bit Signal Description
C000	Machine Lock/Feed hold
C001	Reset
C002	Emergency stop
C003	Reserved
C004	Cycle start
C005	Program No. selection by INPUT key
C006	Single Block
C007	NC alarm & stop NC command
C008~C010	Reserved
C011	Program restart -1
C012~C013	Program restart from the previous program after jumping to other program
C014	Reserved
C015	Dryrun
C016~C019	Reserved
C020~C021	Cursor up / Cursor down command
C022~C023	Page up / Page down command
C024~C025	Reserved
C026	Option stop
C027	Option skip
C028	Speed deceleration selection for G31 Skip function
C029	Reserved
C030	Counter status (On/Off) command
C031	Ladder simulation
C032	M-code finish signal from PLC to CNC
C033	T-code finish signal from PLC to CNC
C034	S-code finish signal from PLC to CNC
C035	Reserved
C036	Round corner non-stop operation
C037~C055	Reserved
C056	MPG hand-wheel test mode
C057	Return to the last program block where execution was interrupted
C058	Program EDIT mode disabled
C059	MCM EDIT mode disabled
C060~C062	Reserved
C063	HOME execution command
C064~C065	HOME limit switch signal for X, Y-axis
C066~C079	Reserved
C080~C081	Set X, Y-axis work coordinate system
C082~C095	Reserved
C096~C097	Clear X, Y-axis program & machine coordinate to zero (0)
C098~C116	Reserved
C117	C117=0, Open loop circuit for spindle C117=1, Closed loop circuit for spindle
C118~C119	Reserved
C120	Command code for program selection from external device
C121	Command code for spindle speed control (in %) from external I/O device
C122	Reserved
C123	Set work coordinate system (G54~G59)
C124	Reserved
C125	M, T, and S-code to be executed in synchronization
C126~C127	Reserved
C128~C129	To enable axial data input (X, Y) when in TEACH mode

C130~C143	Reserved
C144~C145	To enable digital readout mode for X, Y-axis
C146~C159	Reserved
C160~C163	Hardware limit for (+/-) X, (+/-) Y-axis
C164~C191	Reserved
C192~C193	Clear error count to 0 for X, Y-axis
C194~207	Reserved
C208	X-axis. Positioning execution in PLC
C209	Y-axis. Positioning execution in PLC
C210~C223	Reserved
C224~C225	Set G28, G29, G30 mode
C226~C227	Reserved
C228	In JOG mode, axial movement control if R233 or R234 ≠ 0.
C229~C230	Feed-rate adjustment by MPG hand-wheel
C231	When C231=1→0, read the program that was stored in the EPROM/FLASH-ROM
C232	Reserved
C233	When C233=1→0, Clear G54 work coordinates (MCM #1~#2 = 0)
C234~C234	Reserved
C235	Program simulation without V-command output and encoder feedback
C236	For system use only
C237~C241	Reserved
C242	MPG hand-wheel feed (input) for INACTIVE axis during AUTO mode
C243	MMI (Man Machine Interface) or LCD screen in sleep mode
C244	Suppressing alarm ERROR-2 when following count > 4096
C245	User defined keyboard function input
C246	Reserved
C247	LCD display disabled
C248~C254	Reserved
C255	User defined keyboard function output

6.2 PLC Command Signal C-Bits

A C-bit is a command signal from the PLC to the HUST controller for CNC functional control. The C-bits for HUST H2N controller are shown in Table 6-2 and their meanings are discussed in the followings.

- **Machine Lock/FEED-HOLD (C000)**

C000 = 1 (high), All machine tool operation stops.
C000 = 0 (low), Machine tool operation in process.

If bit C000 is set high (1), control unit will stop all CNC operations. It will remain stopped until C000 becomes low again. The relationship between C000 and the machine motion is shown as below.

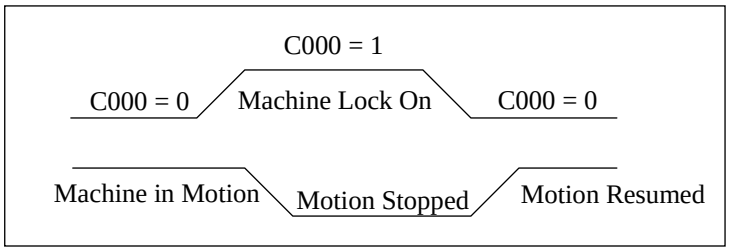
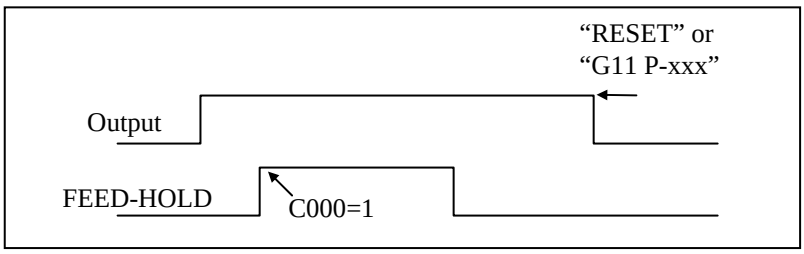


Fig 6-2 Relationship between C000 Signal and Machine

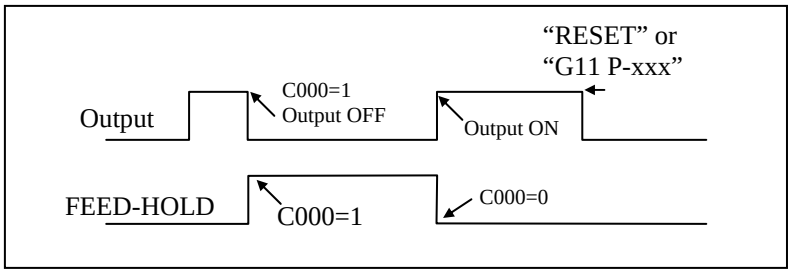
Effect of FEED-HOLD on output control command "G11 Pxxx Lyyy"

G11 Pxxx Lyyy, where xxx = the output number, yyy = 0 or any number.

1. If "yyy"=0 or void, this command is to turn ON the output specified by "xxx" which can be turned OFF either by RESET or by G11 P-xxx. FEED-HOLD will not change the status of output "xxx".



2. If "yyy"= any number, this is to turn ON output "xxx" which can be turned OFF by Feed-Hold. It will remain OFF as long as 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.



Please refer to HUST H2N Operation Manual for more information on G11.

- **RESET (C001)**

  C001 = 1 (high), RESET the whole system to initial status.
  C001 = 0 (low), RESET function is cleared.

- **Emergency Stop (C002)**

  C002 = 1 (high), controller reset and Emergency stop.
  C002 = 0 (low), the function is cleared.

  Bit C002 informs the control unit to execute emergency stop. When bit C002 is set high (1), the control unit is reset and emergency stop is applied to the machine. Emergency stop signal turns off the servo ready signal. Refer to Section 5.5 for suggested connection method.

  When the problem that activated an emergency stop has been resolved, please reset the control unit and execute HOME process before normal operation.

- **Cycle Start (C004)**

  C004 = 1 (high), start program execution. (one pulse)
  C004 = 0 (low), no response.

  Bit C004 informs the control unit to start program execution. However, Cycle-Start signal (C004) is ignored when:

  1. The control unit is already in "IN PROCESS" mode.
  2. The control unit is not at AUTO, SINGLE, MDI, HOME, or TAPE mode.
  3. Emergency stop signal (C002) is at high (1).
  4. The control unit is in alarm state. (C007=1)

- **Program NO. Selection by INPUT Key (C005)**

  When in PRNO mode, Bit C005=1 informs the control unit that program number selection has been completed after pressing [P] key, program number [xxx] and INPUT key.

- **Single Block (C006)**

  C006 = 1 (high), SINGLE mode execution selected.
  C006 = 0 (low), function OFF.

When bit C006 = 1, the current execution is in SINGLE mode. Pre CYCST to execute. If AUTO and SINGLE modes are selected at the same time, the priority of SINGLE mode is higher than AUTO mode.

While in SINGLE mode, one block of data will be read from the memory for execution if a cycle start signal is entered. Operation can be changed to MDI or JOG by the mode selection signal under single block state. Fig 6-8 is a time chart for SINGLE block execution.

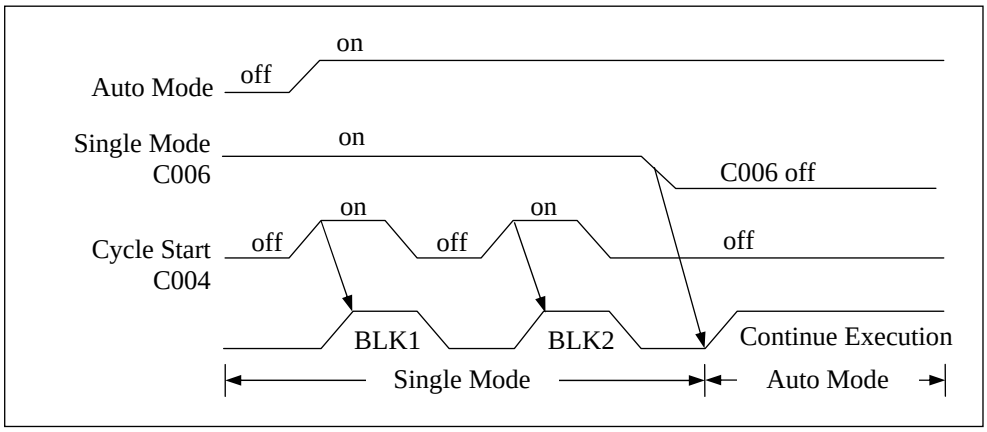


Fig 6-3 Time Chart for Single Block Execution

- **NC Alarm (C007)**

  C007 = 1 (high), NC alarm is ON.
  C007 = 0 (low), NC alarm is OFF.

  Bit C007=1 to inform the CNC unit a machine problem has been detected. When the external device encounters any trouble, such as spindle servo alarm, these alarm signals can be used, through input points, to inform the control unit to stop execution by setting C007 high (1). LCD screen displays ERROR 37 NC ALARM. Resolve the problem, RESET before restart the program.

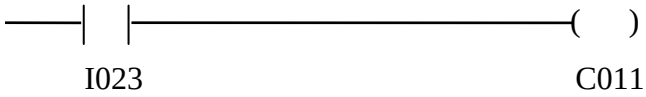
- **Program Restart-1 (C011)**

  C011 = 1 (high), Program Restart mode ON.
  C011 = 0 (low), Program Restart mode OFF.

  If you interrupt the program execution by pressing RESET or Emergence-stop, you can set C011=1 to restart the execution from the program block where the program was interrupted. HUST H2N does not provide this function key on the keyboard and it must be processed through PLC.

  Follow the steps below to restart the interrupted program:

  1. Press RESET. Use MPG to move tool away. If the interruption was caused by Em-stop or servo alarm (Error 2), execute HOME then RESET.
  2. Press user installed [Restart] key to set C011 = 1 in PLC as follow.



3. In AUTO mode, press CYCST key to restart. At the end of M02 or M30, the restart function will be automatically cleared.

Note that the M, T, S-code in the previous program blocks prior to program interruption will be re-executed during restart operation.

Example: See Fig 6-4.

Assuming that the program was interrupted at block N40. When restart function is activated, 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.

N10 S200
N20 G0 X50. Y100.
N30 G1 V-20. F200.
N40 X60. V-20. ..... Program interrupted here and restart from here
N50 V-20.000
N60 X80. V-20.
N70 G0 X250. Y150.
N80 M2

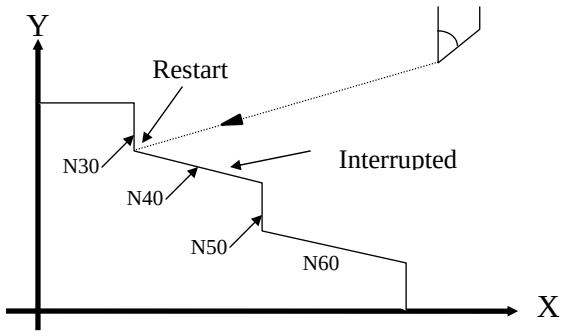


Fig 6-4 Program restart

- **Program Restart –2 (C012~C013)**

Bits C012~C013 are also used to restart the program from where the execution was interrupted. They are applied to the case when you interrupt the execution of the 1st program and jump to the 2nd program. Once you finish the 2nd program, you can use C012~C013 to go back to the 1st program and restart from where it was left. The procedures are as follows:

1. When program #1 is interrupted, C012=1 and the interrupted block number is stored.

2. Jump to the 2nd program or do some other task such as JOG function.
3. Under AUTO mode, select the program number of the interrupted program.
4. Set C013=1 to read the stored block number in step 1.
5. Set C011=1.
6. Press CYCST.

- **Dry-run (C015)**

  C015 = 1 (high), Dry-run mode selected.

  When C015=1, the control unit will execute Dryrun function. The dryrun mode can be set at any time during program execution. When dryrun is set, the feed-rate will change to fast speed at the beginning of next block. When dryrun mode is turned OFF, the feed-rate will return to normal cutting speed at the beginning of next block. Fig 6-5 shows the relationship between dryrun mode and normal execution mode. Notice that G00 manual feed-rate override is also effective.

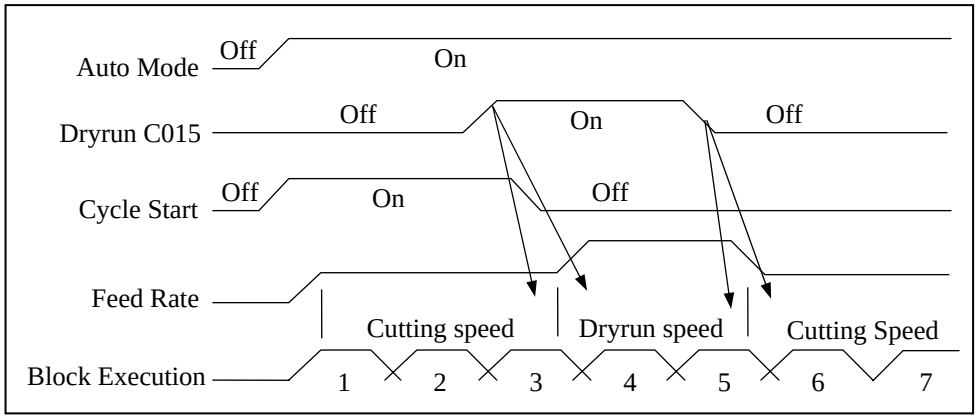


Fig 6-5 Time Chart for Dryrun During Program Execution

- **Cursor Up And Cursor Down (C020~C021)**

  C020 = 0 ~ 1 (rising edge), the cursor moves up one line.
  C021 = 0 ~ 1 (rising edge), the cursor moves down one line.

  Bit C020 informs the control unit to move the cursor one line up. It is effective only at the moment when the signal (C020) rises from low (0) to high (1). This function can be controlled by external input to remotely move the cursor. Bit C021 is for Cursor Down function.

- **Page Up And Page Down Key (C022~C023)**

  C022 = 0 ~ 1 (rising edge), move cursor one page up
  C023 = 0 ~ 1 (rising edge), move cursor one page down

  Bit C022 informs the control unit that Page-up function has been selected. When bit C022 is rising from low (0) to high (1), move cursor one page up. This function can be controlled by external input. Bit C023 is for Page-down function.

- **Option Stop (C026)**

  C026 = 1 (high), Option Stop is ON.
  C026 = 0 (low), Option Stop is OFF.

  If bit C026 is set low (0), M01 block will be ignored during program execution. If bit C026 is set high (1), M01 block will stop program execution until CYCST is pressed again.

- **Option Skip (C027)**

  C027 = 1 (high), Option Skip is ON.
  C027 = 0 (low), Option Skip is OFF.

  If bit C027 is set 1, the block labeled with "/1" will be skipped during program execution. If C027 = 0, the block labeled with "/1" will be executed.

- **Speed Deceleration Selection for G31 Skip Function (C028)**

  Bit C028 informs the control unit how to decelerate the servo motor when G31 skip function is encountered in the program. C028 is used in conjunction with the value in R190.

  C028 = 0 (low), servo motor ignores the value in R190 and decelerates linearly to zero.
  C028 = 1 (high) & R190 = 0, servo motor speed jumps to zero.
  C028 = 1 (high) & R190 ≠0, servo motor will stop in a distance = value in R190.

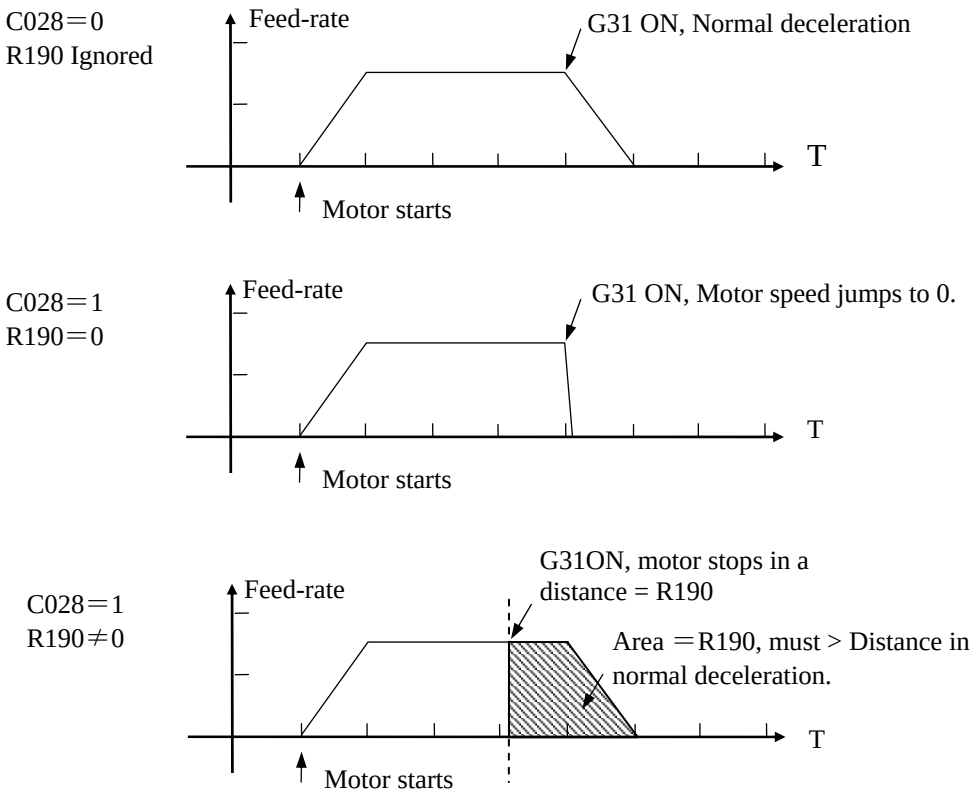


Fig 6-6 Motor Deceleration with C028 and R190

- **Counter Status (C030)**

  C030 = 1 (high), system counter (MCM #65) is ON.
  C030 = 0 (low), system counter is OFF.

  With the system counter active, the number in MCM #65 will increase by one (1) every time when the program execution encounters M02, M30 or M99. When the number in MCM #65 reaches a value equal to the number stored in MCM #66 (Counter limit), the program execution stops.

  If the counter is OFF, the number in MCM #65 will remain the same all the times and its value will never exceed the counter limit. If the counter limit (MCM #66)=0, the execution of the program ended with M99 will never stop even C030=1.

- **PLC Ladder Simulation (C031)**

  C031 = 1 (high), CNC is ready to accept ladder simulation procedures.
  C031 = 0 (low), ladder simulation stops.

  When C031 = 1, a 12K RAM memory in HUST H2N controller will become available for user to test his ladder program that was edited in the personal computer. Note that user's ladder program should be able to process C031 signal for the simulation procedures to be effective. Please refer to Sec. 6 of Chap. 9 in HUST H2N Operation Manual for ladder simulation procedures.

- **M-, T-, S-code Command Finish Signal (C032~C034)**

  C032 = 1 (high), M-code command is finished.
  C033 = 1 (high), T-code command is finished.
  C034 = 1 (high), S-code command is finished.

  When you use an M-code to do work on an external device, it normally takes some times for the external device to finish the work. When the work is finished, the controller will then execute the next block of program. To effect the waiting period, you set C032=0 when the M-code is encountered. When the external work is finished, let C032=1 in your PLC and the execution will continue from the next block. T-code and S-code work the same way. The M-code finish signals are effective only for the range of M003~M499.

- **Round Corner Non-stop Operation (C036)**

  C036 = 1 (high), Round Corner Non-stop operation is ON.
  C036 = 0 (low), Round Corner Non-stop operation is OFF.

  When the tool motion changes from one direction to another, it will produce a sharp corner and the motors will decelerate and accelerate at the intersection. This condition will produce an undesirable effect for some industrial machinery such as glue machine, flame or laser cutting machine. To overcome this problem, set C036=1 to produce a round corner and the motors will also run continuously when cornering.

- **MPG Hand-wheel Test Mode (C056)**

  C056 = 1 (high), MPG Hand-wheel Test Mode ON.
  C056 = 0 (low), MPG Hand-wheel Test Mode OFF.

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

  When MPG Hand-wheel Test Mode is on, the feed-rate is controlled by the speed of hand-wheel and the MFO% knob setting. Refer to Sec. 8.9 of Operation Manual for MPG hand-wheel test.

- **Return to The Last Program Block (C057)**

  C057 = 1 (high), Return to the last program block function ON.
  C057 = 0 (low), Return to the last program block function OFF.

  When this function is turned on in EDIT mode, the tool will be returned to the last program block where the program was interrupted. This function is also effective in TEACH mode.

- **Program EDIT and MCM EDIT Function Disabled (C058~C059)**

  C058 = 1 (high), Program EDIT mode is disabled.
  C059 = 1 (high), MCM parameter EDIT mode is disabled.

- **HOME Execution Command (C063)**
  HOME Limit Switch Signals (C064~C065)

  C063 = 1 (high), execute **HOME** cycle.
  C064 = 1 (high), X-axis limit switch is being touched.
  C065 = 1 (high), Y-axis limit switch is being touched.

  If C063 is set high (1), the control unit will execute HOME cycle according to the axis selected by R232 and the direction by MCM #50~51. Then, the controller will move the tool to find the HOME position according to the following steps. (Use X-axis for explanation)

  Step 1- The tool will go forward until C064 = 1 (high), the machine then stops. While going forward, the feed-rate is by MCM #52 and the direction by MCM #50.

  Step 2- The tool will go backward at a feed-rate equal to 1/4 of the speed in Step 1 until C064 = 0 (low). The servo motor will decelerate to a stop.

  Step 3- The tool will continue going backward at the HOME GRID speed as set by MCM #54 to look for the encoder index. Once the encoder index is detected, the motor stops and the HOME cycle is completed.

To find HOME position for Y-axis, the same procedures described above apply. Fig 6-7 is a typical time chart for HOME execution.

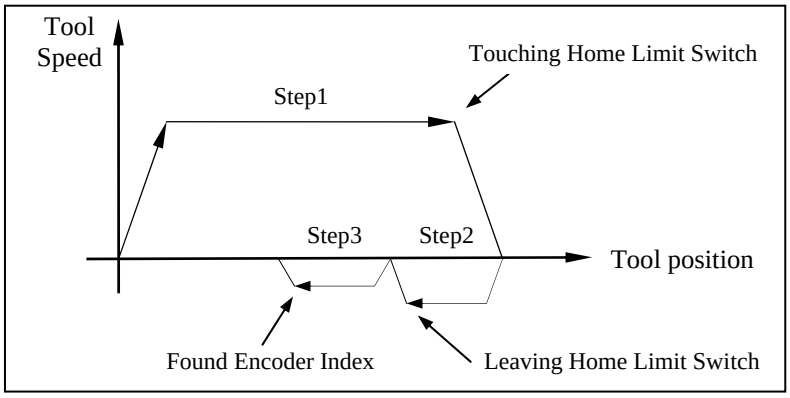


Fig 6-7 Time Chart for HOME Execution

- **Specify Work Coordinate Group as Working Coordinate (C123)**
  Set the Current Program Position as Working Coordinate (C080~C081)

  C123 = 1~0 (falling edge), use the data in MCM parameters as the current work coordinate.
  C080 = 1~0 (falling edge), use the current program position for X-axis as the work coordinate origin.
  C081 = 1~0 (falling edge), use the current program position for Y-axis as the work coordinate origin.

  HUST H2N controller provides 6 groups of work coordinate (G54~G59) with their location (with respect to HOME) data stored in MCM #1~#12 as shown in Table below. Also shown is the group designation that is stored in register R228. (See Table 6-4 for Registers) The default setting of work coordinate system is G54. To change work coordinate, please follow one of the methods described below.

Work Coord.	MCM No.	Group No. by R228
G54	1 2	0
G55	3 4	1
G56	5 6	2
G57	7 8	3
G58	9 10	4
G59	11 12	5

Table 6-3 Work Coordinate

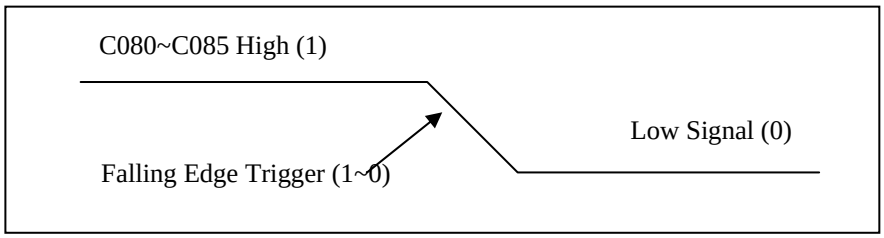
1. When in MDI or AUTO mode, execute one of G54~G59 function in the program. Press “RESET” will restore the work coordinate back to default G54 system. For example, the program below is in G54 system when power-on. It starts executing under AUTO mode and it changes to G55 system at block N10. Once it finishes execution, press RESET will restore to G54 work coordinate system.

```
O01
N10 G55
N20 U100.
N30 M02
```

2. When C123 = 1~0 (falling edge) in PLC, it triggers the controller to use the group number as stored in R228 with the corresponding data in MCM #1~#12 for work coordinate system. Again, press RESET will restore to G54 system. For example, use G55 (2nd group) work coordinate as the current working coordinate.

   Step 1- Let R228 = 1. (Table 6-5)
   Step 2- Set C123 = 1→0 (falling edge) in PLC. This triggers controller to use G55 as work coordinate and the data in MCM #3~#4 as G55 coordinate origin.

3. Bits C080~C081 are also commands to set work coordinate. But they differ from C123 in that they set the current program position as the working coordinate. For example, when bit C080 = 1→0 (falling edge trigger), the controller will set the current program position for X-axis (assuming X=10.) as the working coordinate origin, i.e., X = 0. The machine position of the new origin will be stored in the appropriate MCM parameters.



- **Clear the Current Axial Work and Machine Coordinate to Zero (C096~C097)**

  C096 = 1, Clear the work and machine coordinate at current tool position on X-axis to zero.
  C097 = 1, Clear the work and machine coordinate at current tool position on Y-axis to zero.

  C096~C097 are used to clear both work and machine coordinates when the program is NOT in execution mode. They are not effective when program execution is in process.

- **Spindle Control (C117)**

  Spindle control can be divided in two (2) ways:
  1. Closed loop control (C117=1) with encoder feedback – This allows you to control the spindle position and the speed.
  2. Open loop control (C117=0) without encoder feedback – Control the spindle speed.

- **Command Code for Program Selection from External Device (C120)**

  C120 = 1, function of program selection from external device ON
  C120 = 0, function of program selection from external device OFF

  R226 is the register to store the program number from external device while R255 stores the current program number. When C120 = 1, the number in R255 will be replaced by the number in R226.

- **Command for Spindle Speed Control (in %) from External I/O Device (C121)**

  C121 = 1, Spindle speed percentage = value in R225.
  C121 = 0, Spindle speed percentage = value in R224.

  Ex 1: C121 = 1, use X-axis as spindle by setting R238 = 1, R225 = 60.
  V-command to spindle = 10 * R225 = 10 * 60% = 6 V

  Ex 2: C121 = 0, use Y-axis as spindle by setting R238 = 2, R224 = 50.
  If MCM #92 = 2000 rpm (spindle speed at 10V) and S = 1000 rpm
  V-command to spindle = 10 * (1000/2000) * R224 = 10 * (1/2) * 50% = 2.5V
  Spindle RPM = 1000 * 50% = 2000 * (2.5/10) = 500

- **Set Work Coordinate (C123)**

  When C123 = 1→0 (Falling edge) and R228 = 0, select G54 as work coordinate.
  When C123 = 1→0 (Falling edge) and R228 = 1, select G55 as work coordinate.
  When C123 = 1→0 (Falling edge) and R228 = 2, select G56 as work coordinate.
  When C123 = 1→0 (Falling edge) and R228 = 3, select G57 as work coordinate.
  When C123 = 1→0 (Falling edge) and R228 = 4, select G58 as work coordinate.
  When C123 = 1→0 (Falling edge) and R228 = 5, select G59 as work coordinate.

- **Simultaneous Execution for M-code, S-code, T-code (C125)**

  When C125 =1, all M-code, S-code and T-code will be executed simultaneously.
  For example: G01 X100. S2000 M18
  When C125 =0, the controller will execute M18 first, then start the motor.
  When C125 =1, the controller will execute M18 and start motor simultaneously.

- **Axial Data Input Disabled in TEACH Mode (C128~C129)**

  C128 =1, X-axis data input disabled when in TEACH mode.
  C129 =1, Y-axis data input disabled when in TEACH mode.

- **DRO Mode Enabled (C144~C145)**

  C144 =1, Digital readout mode on X-axis enabled.
  C145 =1, Digital readout mode on Y-axis enabled.

- **Signal for Over-travel (OT) Limit Switch (C160~C163)**

  C160 =1, OT limit switch in the X+ side is touched.
  C161 =1, OT limit switch in the X- side is touched.
  C162 =1, OT limit switch in the Y+ side is touched.
  C163 =1, OT limit switch in the Y- side is touched.

  When the machine tool or the work table touches one of the hardware OT limit switch, the corresponding C-bit sends a high (1) signal to CNC which produce a CNC error signal to reset and stop the machine. When this happens, you have to manually move (MDI mode) the tool away from the limit switch then RESET.

- **Clear Error-count (C192~C193)**

  C192 =1, Clear error-count in X-axis.
  C193 =1, Clear error-count in Y-axis.

- **Positioning Execution in PLC Program (C208~C213)**
  (See Table 6-4 for Registers R160~R161, R180~R181)

  C208 = 0 → 1, tool on X-axis moves to the position R180*10 with a speed of R160. The S-bit S208=1 during command execution and S208=0 when execution finished. This is done in PLC program.
  C209 = 0 → 1, tool on Y-axis moves to the position R181*10 with a speed of R161. The S-bit S209 = 1 during command execution and S209 = 0 when execution finished. This is done in PLC program.

- **Special C-bit for G28, G29 and G30 (C224~C225)**

  In addition to the normal application in the part program, G28, G29 and G30 functions can also be executed through C224~C225 (1 pulse signal) in PLC as follows:

	C225	C224
G28	0	1
G29	1	0
G30	1	1

  When using C224 and C225, remember to specify the axis for G28~G30 in R232. For example when C224=1 and R232=3 (see Table 6-4), G28 will be executed in X- and Y-axis.

- **Axial Motion Command in JOG Mode (C228)**

  When C228 =1 and R233≠0 (or R234≠0) in JOG mode, the designated axis will move
  (JOG feed) right away. The designated axis is determined by the binary bit being stored in R233 or R234. Multiple axes can be put in motion simultaneously. For

example, R233 = 3 = 11 in binary, so X, and Y-axis (1=ON) will move simultaneously.

R233	Bit-1, R233=2	Bit-0, R233=1
Axis	Y+	X+

R234	Bit-1, R234=2	Bit-0, R234=1
Axis	Y-	X-

- **G01 Feed-rate Adjustment by MPG Hand-wheel (C229, C230)**

  When C229 = 1 and C230 =1, G01 feed-rate adjustment by MPG hand-wheel function is ON. When adjustment is completed and C229 = 0, two situations for C230 can exist as follows.

  1. C230 =0, G01 MFO returns to the feed-rate before adjustment.
  2. C230 =1, G01 MFO maintains the feed-rate after adjustment.

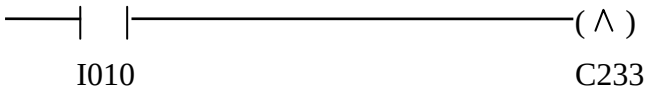
- **Load File from EPROM/FLASH-Rom (C231)**

  C231 = 1~0 (falling edge), Load program file from EPROM or FLASH/Rom.

  C231 bit can be used to load the program file (part program or PLC program) that was burned into the EPROM or FLASH-Rom. In addition, single block command of "G10 P2100" can be used to load program into Flash-Rom, but NOT EPROM.

- **Clear G54 Work Coordinates (C233)**

  C233 = 1→0, clear G54 work coordinates that are stored in MCM #1~#2 to zero (0). This can be done by using an external input as follow. Note that C233 is one shot signal.



- **Program Simulation without V-command and Encoder Feedback (C235)**

  C235 = 1, program simulation without voltage command and encoder feedback.
  C235 = 0, function NOT effective.

- **MPG Hand-wheel Feed Interrupt Mode (C242)**

  C242 = 1 (high), MPG hand-wheel feed interrupt mode ON.
  C242 = 0 (low), MPG hand-wheel feed interrupt mode OFF.

  When the function is on, the work-piece position on the inactive axis during program execution can be moved by the MPG hand-wheel. When the program execution comes to the block where the inactive axis becomes active, the new position is the starting position for that block. When you apply this function, be careful about the

tool path of your program to avoid any potential obstructions and damages to your tool. This function is effective only in AUTO mode, but ineffective in the active axis.

- **MMI (Man Machine Interface) or LCD Screen In Sleep Mode (C243)**

  C243 = 0, sleep mode for MMI or LCD screen OFF
  C243 = 1, sleep mode for MMI or LCD screen ON. This is to extend their life expectancy.

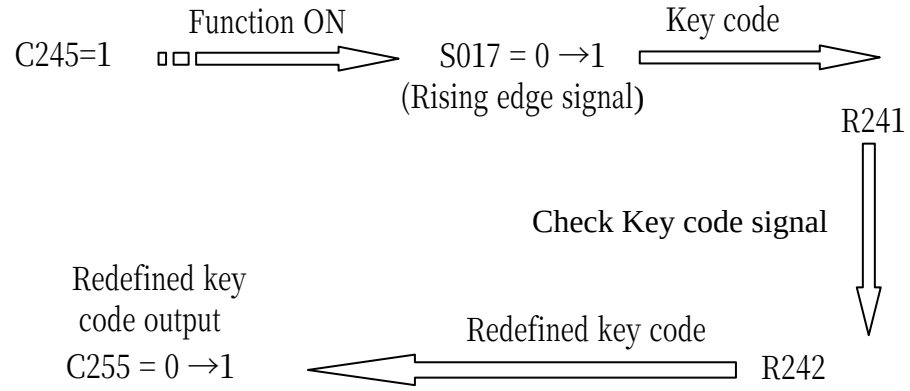
- **Suppressing Alarm ERROR-2 When Following Count > 4096 (C244)**

  C244 =0, controller will send out ERROR 02 when the following-count > 4096.
  C244 =1, controller will NOT send out ERROR 02 when the following-count > 4096.

- **Re-defined Keyboard Key (C245)**

  C245 = 1, Self-defined keyboard function ON. All keys on keyboard have to be redefined.
  C245 = 0, Self-defined keyboard function OFF.

  User can redefine the keys on HUST H2N keyboard by following the steps below. This is done through PLC process. Please see R241 and R242 in next section for more information.



1. Let C245 = 1 to enable redefining function and disable all the key functions (key codes still retained). If only redefining keys 43~47 (see Fig 6-8), it's NOT necessary to set C245 =1.
2. S017 = 0→1 indicates a key has been pressed.
3. Controller places the key code (see Fig 6-8) in step 2 in R241.
4. Based on the S017 signal and R241, store the desired output key number in R242.
5. Let C255 = 0→1 to output the new defined key in R242. (Table 6-5)

Following is a PLC example where key 44 is to be defined as letter "N", key 45 as "M", and key 46 as "single block execution" function. Press key 46 once, function ON and press one more time, function OFF.

- **LCD Display Function Disabled (C247)**

  C247 =0, LCD display disable function OFF
  C247 =1, LCD display disable function ON

- **User Defined Keyboard Function Output (C255)**

  C255 =0, User defined keyboard output function OFF
  C255 =0→1, User defined keyboard output function ON (see C245)

6.3 Description of PLC Register

This section describes the registers that are used in PLC for HUST H2N controller. Table 6-4 is a summary of registers with their corresponding variable number.

1. R000~R099 are for user's application in the PLC program.
2. R100~R255 are reserved for HUST H2N system data, NOT for customer's use. The data in R000 ~ R199 can be saved with power off if MCM #86=256, and the data will NOT be saved if MCM #86=0.

Table 6-4 Registers and Corresponding Variables for HUST H2N Controller

Register	Variable No.	Application / Definition
R100	10100	Execution mode selection
R160~R161	10160~10161	Feed-rate setting for X and Y-axis
R180~R181	10180~10181	Absolute Coordinate setting for X and Y-axis
R190	10190	Type of motor deceleration selection for G31 skip function
R200	10200	Page number for Liquid LCD Screen display
R220	10220	Feed-rate percentage (%) for G00 (G00 MFO %)
R221	10221	Feed-rate percentage (%) for G01, G02, G03
R222	10222	Multiplication factor for MPG pulse (MPG MFO %)
R223	10223	Spindle Gear ratio
R224	10224	When C121=0, spindle speed percentage (%) = value in R224
R225	10225	When C121=1, spindle speed percentage (%) = value in R225
R226	10226	Selecting program number from external I/O device
R227	10227	The selected variable number when editing the variable data
R228	10228	Selecting work coordinate system
R229	10229	Setting the number of decimals for display after power on
R230	10230	Page number for Man Machine Interface display
R231	10231	Reserved
R232	10232	For selecting the axis in JOG and HOME mode
R233	10233	For selecting the axis in JOG and MPG mode, (+) direction
R234	10234	For selecting the axis in JOG and MPG mode, (-) direction
* R235	10235	For displaying the axis that is drivable (enabled)
* R236	10236	For displaying the axis that is on HOME location (machine zero)
* R237	10237	For displaying the axis of which the HOME operation has been executed
R238	10238	For selecting the axis to be the spindle axis
* R241	10241	Storing the keyboard key code as read by the controller
R242	10242	Storing the desired output key code
R250	10250	Input signal type for G31
* R251	10251	For storing the ERROR code
* R252	10252	For storing current M-code
* R253	10253	For storing current T-code
* R254	10254	For storing current S-code
* R255	10255	For storing the current program number

* Data in these registers are “READ only”.

- R100, Execution mode selection according to table below

R100	Execution
0	NONE
1	AUTO
2	HOME
3	EDIT
4	JOG
5	TAPE
6	TEACH
7	MDI

- R160~R161, Feed-rate setting for X and Y-axis
  R180~R181, Absolute coordinate setting for X and Y-axis

  These registers are for positioning in PLC and are used with C208~C209. The coordinates R180~R181 will be multiplied by 10 and their maximum setting values are 32767.
  When C208 = 0→1, X-axis moves to the position of R180*10 with a speed of R160.
  While executing, S208=1 and S208=0 when execution completed.
  When C209 = 0→1, Y-axis moves to the position of R181*10 with a speed of R161.
  While executing, S209=1 and S209=0 when execution completed.

  Ex: R180 = 1000, R160 = 2500.
  When C208 = 0→1, X-axis moves to the location of 10.000 (R180*10=10000) with a speed of 2500 mm/min.

- R190, Type of motor deceleration selection for G31 skip function

  R190 is used in conjunction with C028.
  C028 = 0 (low), servo motor ignores the value in R190 and decelerates linearly to zero.
  C028 = 1 (high) & R190 = 0, servo motor speed jumps to zero.
  C028 = 1 (high) & R190 ≠0, servo motor will stop in a distance = value in R190.

- R200, Page number for liquid LCD screen display

  R200 = 0, Standard system display
  R200 = 1~63, Page number for user designed LCD screen display

- R220, Setting feed-rate percentage for G00 MFO %

  Default = 100 (%), range 0 ~ 100.
  For example, if program G00 feed-rate = 3000 and R220 = 80, the actual G00 feed-rate output = 2400.

- R221, Setting feed-rate percentage for G01, G02, G03 MFO %

Default = 100 (%), range 0 ~ 150. Note that the actual feed-rate output (the program feed-rate * R221) cannot exceed the maximum feed-rate set by MCM #56~#57).
For example, if program G01 feed-rate = 2000 and R221 = 120, the actual G01 feed-rate output = 2400.

- R222, Multiplication factor for MPG pulse

  Default = 100 (%), range 1 ~ 100.
  For example, R222 = 75, the MPG pulse output = MPG pulse spec * 75.

- R223, Spindle gear ratio

  Default = 100, range 1 ~ 100.

- R224, Spindle speed percentage (%)

  When C121=0, spindle speed percentage (%) = value in R224
  Default = 100 (%), range 1 ~ 150.
  For example, select Y-axis as spindle by setting R238=2, program S1000, C121 =0 and R224 = 50, the actual spindle speed = 500 rpm. If MCM #92 = 2000 rpm, the V-command = 10V * (1000/2000) * 50% = 2.5V

- R225, Setting spindle speed percentage (%) from external I/O device

  When C121=1, spindle speed percentage (%) = value in R225
  Default = 100 (%), range 1 ~ 150.
  For example, select X-axis as spindle by setting R238=1, C121 =1 and R225 = 60, the V-command to spindle = 10V * 60% = 6V

- R226, Setting program number from external I/O device

  When C120 = 1, the program number from the external I/O device will be placed in R226. When you start execution, this number in R226 will be placed in R255 where the current program number is stored.

- R227, Setting the selected variable number when editing the variable data

  When in variable data editing, the number in R227 is equal the variable number. When pressing RESET, R227=0. When R227 > 0, you are in variable editing mode. You cannot edit variable data in the following modes -- Program-edit, JOG, and Teach mode.

- R228, Selecting work coordinate

  You can select your work coordinate in the part program or from PLC program by setting C123 = 0→1 (Rising edge signal) and R228 value as described below.

C-bit	R228	Work Coordinate
C123 = 0→1	0	G54
C123 = 0→1	1	G55
C123 = 0→1	2	G56
C123 = 0→1	3	G57
C123 = 0→1	4	G58
C123 = 0→1	5	G59

Note that the work coordinate returns to G54 system whenever you press RESET both in part program or PLC program.

- R229, Setting the number of decimals for display after power on

  Default = 3, range = 1 ~ 4.

R229	Format	Number of Decimals
01	6／1	1
02	5／2	2
03	4／3	3
04	3／4	4

Example:

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

- R230, Page number for Man Machine Interface (MMI) display

  Default = 01, range = 1 ~ 255.

- R232, For selecting the axis in JOG and HOME mode to be drivable

  When in JOG or HOME mode, use R232 to select the axis to be drivable as follows.

BIT	1	0
Chose Axis	Y	X

When R232 = 1 → Bit 1=0 & Bit 0=1, X-axis is drivable.
When R232 = 2 → Bit 1=1 & Bit 0=0, Y-axis is drivable.
When R232 = 3 → Bit 1=1 & Bit 0=1, both X and Y-axis are drivable.

Ex: R232 = 3 (=11 in binary), Bits 0, 1 are ON. If you execute HOME operation (C063=1), both X and Y-axis will move simultaneously. If you execute JOG, both axes (X and Y) will move simultaneously.

- R233, For selecting the axis for JOG feed in positive (+) direction

  When in JOG mode and C228=1, use R233 Bit-0 ~ Bit-1 to select the axis for JOG operation in the positive (+) direction. The value in R233 will affect the direction of MPG. So please set R233=0 when not in use. The JOG speed will be by MCM #44~#45.

  When C228 =1, R233 =1, X-axis will move in (+) direction with JOD speed.
  When C228 =1, R233 =2, Y-axis will move in (+) direction with JOD speed.
  When C228 =1, R233 =3, X and Y-axis will move in (+) direction with JOD speed.

  Ex: R233 = 3 (=11 in binary), Bits 1 and 0 are ON. So, if you execute JOG and C228=1, both X and Y-axis will move simultaneously in the positive (+) direction.

- R234, For selecting the axis for JOG feed in negative (-) direction

  When in JOG mode and C228=1, use R234 Bit-0 ~ Bit-1 to select the axis for JOG operation in the negative (-) direction. The value in R234 will affect the direction of MPG. So please set R234=0 when not in use. The JOG speed will be by MCM #44~#45.

  When C228 =1, R234 =1, X-axis will move in (-) direction with JOD speed.
  When C228 =1, R234 =2, Y-axis will move in (-) direction with JOD speed.
  When C228 =1, R234 =3, X and Y-axis will move in (-) direction with JOD speed.

  Ex: R234 = 3 (=11 in binary), Bits 1 and 0 are ON. So, if you execute JOG and C228=1, both X and Y-axis will move simultaneously in the negative (-) direction.

- R235, For displaying the axis that is drivable (enabled)

  Use R235 to display the drivable axis. <u>This is a system-defined parameter and presented here for read only</u>. Do not attempt to change the value in R235.

  When R235 = 1 → Bit 1=0 & Bit 0=1, X-axis is usable.
  When R235 = 2 → Bit 1=1 & Bit 0=0, Y-axis is usable.
  When R235 = 3 → Bit 1=1 & Bit 0=1, both X and Y-axis are usable.

Ex: R235=3 (=11 in binary), Bits 1 and 0 are ON, X and Y-axis are drivable.

- R236, For displaying the axis that is on HOME location (machine zero) – For read only

Bit	1	0
Axis on HOME Loc	Y	X

Ex: R236 = 3 (=11 in binary), Bit 1 & 0 ON, tool on X & Y-axis are on HOME location.

- R237, Displaying the axis of which the HOME operation has been executed

When R237 = 1→Bit 1=0 & Bit 0=1, X-axis HOME operation has been completed.
When R237 = 2→Bit 1=1 & Bit 0=0, Y-axis HOME operation has been completed.
When R237 = 3→Bit 1=1 & Bit 0=1, X and Y-axis HOME operation have been completed.

Power on default = 0. When Error 01, Error 02, or Error 22 occurs, the value in R237 will be cleared to 0. Press RESET and execute HOME again. R237 is for read only.

Ex: R237 = 3 (=11 in binary), Bits 1 and 0 are ON. This indicates that the HOME operation has been completed in X and Y-axis.

- R238, For selecting the axis to be the spindle axis

Default = 0, spindle NOT specified. Only one axis can be selected as spindle at a time.
R238 = 1, X-axis as spindle.
R238 = 2, Y-axis as spindle.

- R241, Keyboard key code as read by the controller

When S017 = 0→1, the controller will read the key code being pressed and store it in R241. The standard HUST H2N keyboard and the key numbers are shown in Fig 6-8 below. Refer to C245 and C255 for user redefined keyboard key.

- R242, Storing the desired output key code

The definitions of output key code for R242 are in Table 6-5. Refer to C245 and C255 for user redefined keyboard key.

- R250, Input signal type for G31

Use R250 to specify the type of input signal to trigger G31 function as follows. Once the type of input signal is specified, G31 function will be triggered by the correct type of signal only. See Operation Manual for G31.

R250 = 0, Input signal type is “rising edge” (0 → 1).
R250 = 1, Input signal type is “falling edge” (1 → 0).
R250 = 2, Input signal type is “always open” (0).
R250 = 3, Input signal type is “always closed” (1).

- R251, For storing the ERROR code

  In case of ERROR during program execution, the ERROR code will be displayed on the HUST H2N LCD screen. The ERROR code is stored in R251 which is for read only.

- R252, For storing current M-code

  When executing a M-code in a program under AUTO or MDI mode, a one shot signal S24 will be generated and this M-code will be placed in R252. The M-code range is M000 ~ M999. When a M-code is finished, C032 becomes high (1). R252 is for read only.

M000~M499	Waiting M-code. The controller will not execute the next program block unless the M-code is finished (C032 = 1).
M500~M999	Non-waiting M-code. The controller will not wait for the M-code to finish. It will execute the next block immediately.

- R253, For storing current T-code

  When executing a T-code in a program under AUTO or MDI mode, a one shot signal S25 will be generated and this T-code will be placed in R253. The T-code range is T000 ~ T999. When a T-code is finished, C033 becomes high (1). When executing a T-code, the controller will not execute the next program block unless the T-code is finished (C033 = 1). R253 is for read only.

- R254, For storing current S-code

  When executing a S-code in a program under AUTO or MDI mode, a one shot signal S26 will be generated and this S-code will be placed in R254. The S-code range is S000 ~ T999999. When a S-code is finished, C034 becomes high (1). When executing a S-code, the controller will not execute the next program block unless the S-code is finished (C034 = 1). R254 is for read only.

- R255, For storing the current program number

  The current (being executed) main program number is stored in R255. R255 is for read only.

  When C120 = 1, the program number from the external I/O device will be placed in R226. When you start execution, this number in R226 will be placed in R255.

Fig 6-8 HUST H2N Keyboard with Key Code

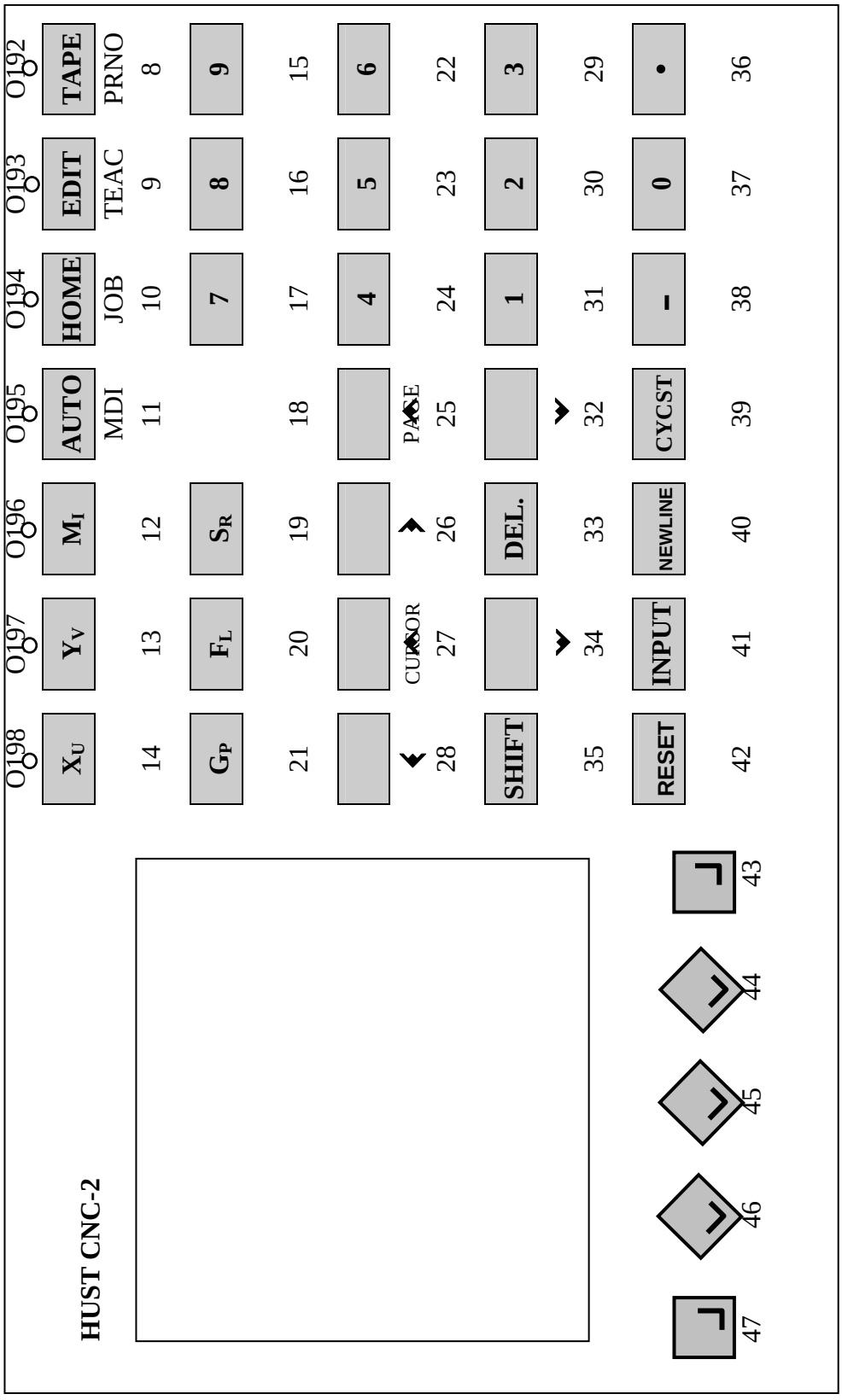


Table 6-5 Definitions of Output Key Code

R242	Output		R242	Output		R242	Output
0	0		30	S		60	
1	1		31	T		61	
2	2		32	U		62	
3	3		33	V		63	
4	4		34	W		64	INPUT
5	5		35	X		65	INSERT
6	6		36	Y		66	AUTO SEQU.
7	7		37	Z		67	FIND LAST "N"
8	8		38	／		68	Delete one block
9	9		39	#		69	Delete current program
10	·		40			70	Delete all programs (*5)
11	—		41			71	MCM parameters reset (*6)
12	A		42			72	Delete current key input signal
13	B		43			73	PAGE-UP
14	C		44			74	PAGE-DOWN
15	D		45			75	CURSOR-UP
16	E		46			76	CURSOR-DOWN
17	F		47			77	Clear variable #1~#9999=0 (*7)
18	G		48	*1		78	Read variables burned into FLASH-ROM #9000 - #9999 (*8)
19	H		49	*2		79	
20	I		50	*3		80	
21	J		51	*4		81	Burn program into FLASH-ROM (*9)
22	K		52			82	Burn MCM parameters into LASH-ROM (*10)
23	L		53			83	Burn PLC ladder into FLASH-ROM (*11)
24	M		54			84	Reserved
25	N		55			85	Burn system data into FLASH-ROM (*12)
26	O		56			86	Burn variables into FLASH-ROM #9000-＃9999 (*13)
27	P		57			87	
28	Q		58			88	
29	R		59			89	

Notes:

*1. R242=48, it means sending a cursor-up signal to MMI (Man Machine Interface).
*2. R242=49, it means sending a cursor-down signal to MMI.
*3. R242=50, it means sending a cursor-left signal to MMI.
*4. R242=51, it means sending a cursor-right signal to MMI.
*5. R242=70, it's equivalent to single block command "G10 P2000".
*6. R242=71, it's equivalent to single block command "G10 P1000".
*7. R242=77, it's equivalent to single block command "G10 P2002".
*8. R242=78, it's equivalent to single block command "G10 P2200".
*9. R242=81, it's equivalent to single block command "G10 P600 L01".
*10. R242=82, it's equivalent to single block command "G10 P600 L02".
*11. R242=83, it's equivalent to single block command "G10 P600 L03".
*12. R242=85, it's equivalent to single block command "G10 P600 L05".
*13. R242=86, it's equivalent to single block command "G10 P600 L06".

6.4 Variables

All data in HUST H2N system is stored in the variables. If you like to check a problem during execution, you can check the variable.

Variable Number	Definition
00001 ... 09999	For user's application
10000 ... 11000	System variables including Register, Counter, Timer
11001 ... 12000	System variables including MCM parameters
12001 ... 13500	System variables including axial positions

1. Variables #10000 ~ #10899

Variable Number	Variable Data	Data Range
10000 ~ 10255	Register	R000 ~ R255
10256 ~ 10511	Counter	Cn000 ~ Cn255
10512 ~ 10767	Timer	Tm000 ~ Tm255
10800 ~ 10807	I - Bit Data	I0000 ~ I0255
10808 ~ 10815	O – Bit Data	O0000 ~ O0255
10816 ~ 10823	C – Bit Data	C0000 ~ C0255
10824 ~ 10831	S – Bit Data	S0000 ~ S0255
10832 ~ 10863	A – Bit Data	A0000 ~ A1023

2. Variables #10900 ~ #11000

Variable Number	Variable Data	Unit
10902	Time required for system to scan one cycle	Millisec.
10903	Time required to finish on cycle of program for M99, M02, M30	Millisec.
10904	Time since power-on	Second
10905	Cumulative time for power-on	Second
10906	Time required to finish on cycle of program for M99, M02, M30	Second
10921	Counter limit for M99, M02, M30	
10922	Current counter for M99, M02, M30	
10923	Current counter for M99	
10924	Current program block number	
10925	Mode status: 0 – NONE, 1 – AUTO, 2 – HOME 3 – EDIT, 4 – JOG, 5 – TAPE 6 – TEACH, 7 – MDI	
10931	G54 .. G59 Work coordinate designation（0 .. 5）	
10932	Current program number	
10933	Production speed ＝ 1 minute ÷#10903	Pc / min

3. Variables #12001 ~ #13500

Variable Number	Variable Data
12001 ~ 12002	Machine Coordinate (Current tool position) *
12021 ~ 12022	Program Coordinate (Current tool position) *
12201 ~ 12202	Machine Coordinate (Planned position) *
12221 ~ 12222	Program Coordinate (Planned position) *
12121 ~ 12122	Following Error
12141 ~ 12142	Unit for Machine Coordinate Resolution (Pulse)
12161 ~ 12162	Unit for Relative Machine Coordinate Resolution (Pulse)

*** Viewing Machine Coordinate and Program Coordinate**

When executing a program in AUTO mode, HUST controller will pre-fetch program blocks (maximum 600 blocks) into buffer and do calculation. However, some command codes, such as G31 and "G65 L50" ~ "G65 L56" will prevent the controller from pre-fetching process. Due to this reason, the machine coordinate and the program coordinate obtained from the variables will be different. Following examples are attempted to explain this difference.

Ex 1:

Viewing the current tool position while executing the program below.
Machine coordinate – V#12001 ~ V#12002
Program coordinate – V#12021 ~ V#12022

```
N001 G01 X300.
N002 G11 P01
N003 X480.
N004 G65 L01 P#1 A#12021 – (Current program coordinate on X-axis)
N005 G01 X0.
N006 M02
```

When you start execution, the tool starts moving while the controller begins pre-fetching process. At block N004, the controller execute "V#1 = V#12021" (current tool position on X-axis) and the tool position at this moment may be at 1.052. If you try to view V#1 at this moment, it'll read V#1=1052.

Ex 2:

Viewing the planned tool position while executing the program below.
Machine coordinate – V#12201 ~ V#12202
Program coordinate – V#12221 ~ V#12222

```
N001 G01 X300.
N002 G11 P01
N003 X480.
N004 G65 L01 P#1 A#12221 – (Planned program coordinate on X-axis)
N005 G01 X0.
N006 M02
```

When you start execution, the tool starts moving while the controller begins pre-fetching process. At block N004, the controller execute “V#1 = V#12221” (planned program position on X-axis) and the tool position at this moment may be at 1.052. The planned program position on X-axis in this case is 480. If you try to view V#1 at this moment, it’ll read V#1=480000.

Ex 3:

Viewing the current tool position while executing the program with G65 L50.
Machine coordinate – V#12001 ~ V#12002
Program coordinate – V#12021 ~ V#12022

```
N001 G01 X300.
N002 G11 P01
N003 X480.
N004 G65 L50 P#99 A1 – (G65 L50 prevents controller from pre-fetching)
N005 G65 L01 P#1 A#12021 – (Current program coordinate on X-axis)
N006 G01 X0.
N007 M02
```

When you start execution, the tool starts moving while the controller begins pre-fetching process. At block N004, the controller stops pre-fetching process and the controller will wait until N003 block is executed. Therefore, when N005 is executed, the current tool position for X-axis (V#12021) is 480. So, V#1=480000.

Ex 4:

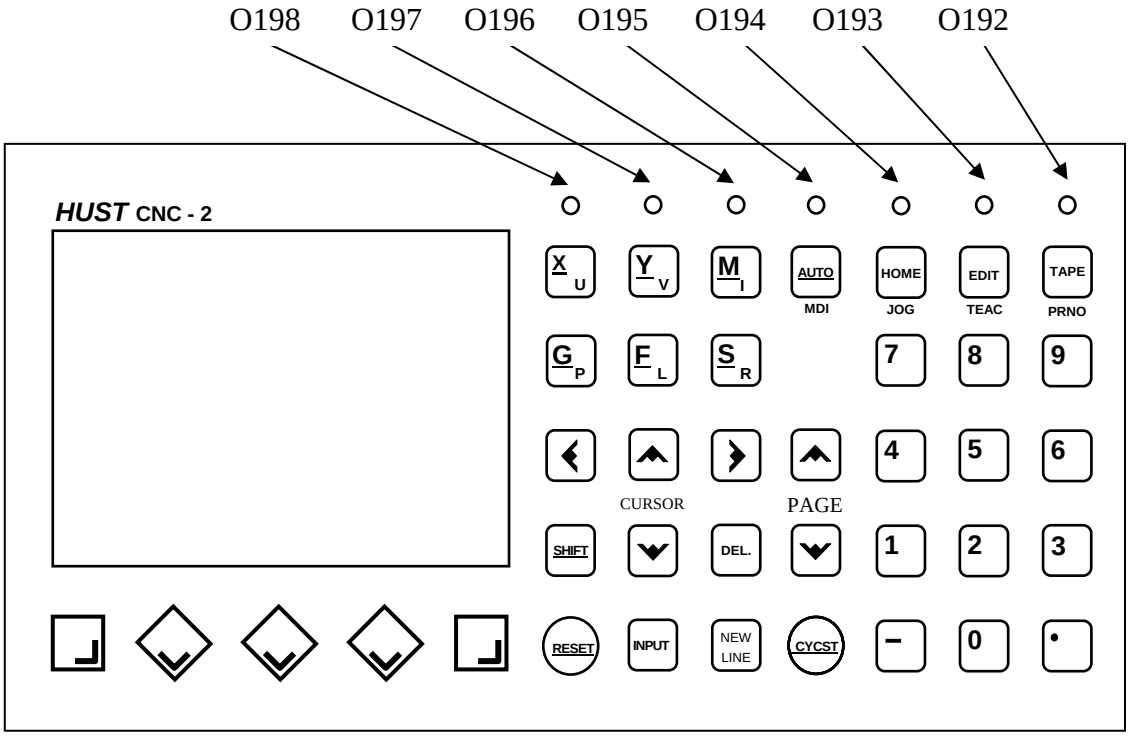
Viewing the current tool position while executing the program with G31.
Machine coordinate – V#12001 ~ V#12002
Program coordinate – V#12021 ~ V#12022

```
N001 G01 X300.
N002 G11 P01
N003 G31 X480. P01
N004 G65 L01 P#1 A#12021 – (Current program coordinate on X-axis)
N005 G01 X0.
N006 M02
```

When you start execution, the tool starts moving while the controller begins pre-fetching process. At block N003, the controller stops pre-fetching process (G31) and the controller will wait until N003 block is executed. Assume G31 causes the tool to jump off at X=425. At block N004, the controller execute "V#1 = V#12021" (current tool position on X-axis) and V#1=425000.

6.5 Input and Output Signal (I-Bits & O-Bits)

1. HUST H2N controller provides 24 I-bits (I00~I23) and 16 O-bits (O00~O15). Among them, I00~I03 are of high speed with a response time of 50 μ sec.
2. All input and output signals are defined as ON = 1 (high) and OFF = 0 (low).
3. HUST H2N keyboard provides 7 LED for user's application. They are output bits and are designated as O192 ~ O198 as shown in the figure below. User can incorporate these bits into PLC for display. They are ready and no external connections are necessary.



7 LADDER DIAGRAM and EPROM

This chapter briefly explains the PLC ladder programs that come as a standard equipment. Users who desire to change the sequential control of the HUST standard PLC Ladder must fully understand the specific purpose of all the S-bits and C-bits that are described in Chap 6 before attempting to make any change. Prior to making any change, lay out and design all the I/O points that are required for the purposes. Note that the S-bits, C-bits, and some I-bits are already defined in HUST H2N controller. Their applications should not be altered or redefined. Otherwise, significant error may result.

7.1 Introduction to Ladder Diagram (PLC or Ladder Program)

A ladder diagram is a sequential control chart for the CNC controller, which in turn issues a voltage command to do some mechanical actions. To edit or revise a PLC program, an optional software called HUST PLC Editor can be used.

Before you start writing a PLC program, a chart depicting the mechanical actions should be laid out, including the inputs and the outputs. Based on this chart, you can start writing your PLC program. When the desired PLC program is completed, you can use the function in the HUST PLC Editor to translate the PLC program to 68000 machine code (*.tsk file). You split this file into *.ODD and *.EVN files and burn it into two blank EPROM as EVEN and ODD and replace the old ones on the CPU main board. More on Eprom in Chap 8.

Following sections will be devoted to discuss the ladder diagrams provided in HUST H2N standard PLC. For PLC Editor, please refer to HUST PLC Operation Manual.

7.1.1 Interface Signal Definitions for PLC Ladder Diagram (see Chap 6)

I-bit : Input signals to the PLC from external I/O devices.
O-bit : Output signals from PLC to external I/O devices.
C-bit : Command signals from PLC to CNC for motion control.
S-bit : CNC status signals to PLC for sequential control.
A-bit : Internal signals used by PLC itself for auxiliary purpose.
Register : R000~R099 for user application and R100~R255 for HUST system data. R000~R199 whose data will be saved when power-off if MCM #86=256 and will not be saved if MCM #86=0.

7.1.2 Maximum Input/Output Points for PLC and PLC Editor

I-bit : 24 input points for PNP type and 144 points for NPN type board.
O-bit : 16 output points for external connections, 256 bits for PLC ladder Editing.
C-bit : 256 bits for HUST H2N PLC and PLC Editing.
S-bit : 256 bits for HUST H2N PLC and PLC Editing.
A-bit : 1024 bits for HUST H2N PLC and PLC Editing.
Timer : 256 timers available with time base of 50 ms, 0.1 sec, 1 sec.
Counter : 256 counters available for count-up, count-down, set, ring-up, ring-down .
Register : 256 registers available.

7.1.3 PLC and PLC Editor Graphical Symbols

The basic graphical symbols used in PLC and the PLC Editor are coil, contact, timer, counter, and register. For the details of these symbols, please refer to HUST PLC Operation Manual.

7.2 Composition and Modification of A Ladder Diagram

Writing a ladder program is a tedious work. To avoid or lessen the potential problems when writing a ladder program, HUST suggests a flow chart (See Fig 7.1) to follow as below:

Step 1: Print the Ladder Diagram
(User can skip this step if writing a new ladder diagram.)

Print out the old ladder diagram (HUST H2N Standard ladder) from print command (PRN) in PLC Editor. User is encouraged to study this old ladder diagram carefully before making any revisions. It is also helpful if the user fully understand the I/O connecting structures for HUST H2N controller. If necessary, please review Chapters 4~6 again.

Step 2: Define Your I-bits, O-bits and the Timing Chart

To control machine tool's motion, the user should fully understand the relationships between the machine tool and the HUST H2N Series I/O interface. At this point, the user must define the I-bits and O-bits to be used, according to the I/O interface connection. In addition, the user should know the exact I/O timing so that the mechanical actions can flow flawlessly in a sequential manners.

Step 3: Use of PLC Editor to Edit Ladder Diagram

When the I/O interface has been assigned and the timing decided, the user can use the PLC Editor to edit, compile and print the ladder diagram for the new model. Once the new model has been completed, user can use the "Ladder Simulation" function to test the revised ladder directly from the PC through RS232C interface. Make sure the new program does not exceed 24-k bytes of memory.

Step 4: Translation of Ladder Diagram to 68000 Machine Code

When the new ladder diagram has been completed, it can be translated to 68000 machine code by using COMPILE command in the HUST PLC Editor. The compiled program will have four (4) different files, appending to user's directory. The one with "*.tsk" is for EPROM or FLASH-ROM writing.

Step 5: Writing and Installing the EPROM / FLASH-ROM

When the jobs in Step 4 have been done, user can use an EPROM writer to write the file "*.tsk" onto EPROM 27C010 or FLASH-ROM 29C010 for HUST controller. The details of EPROM or FLASH-ROM writing will be discussed in Chapter 8.

Install the EPROM or FLASH-ROM to the CNC controller and turn on the power for checking.

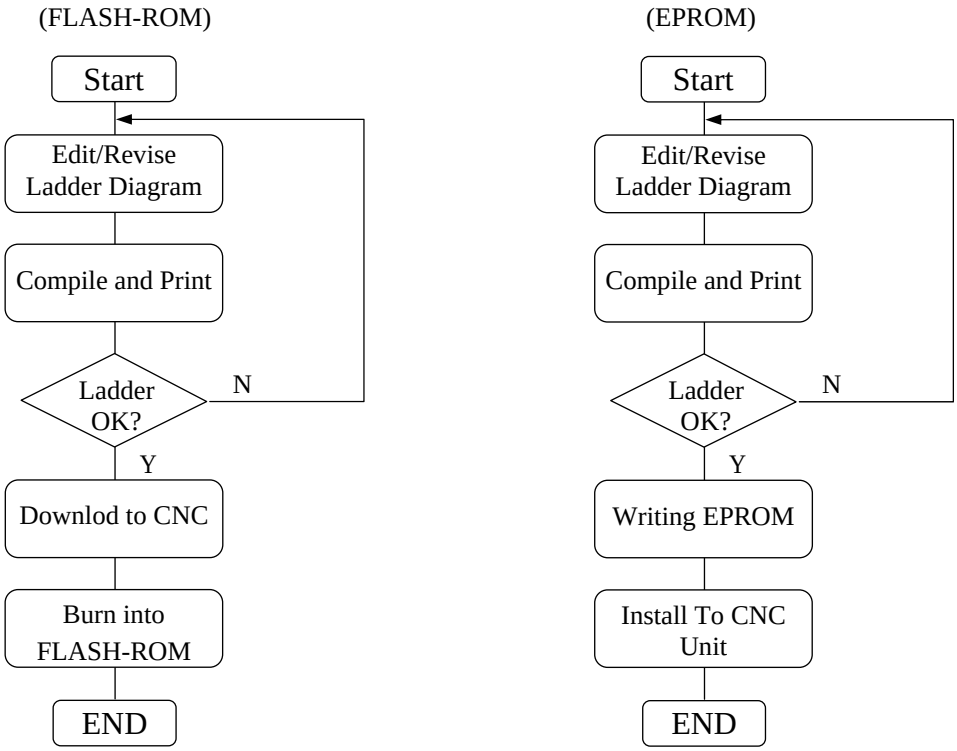


Fig 7.1 Flow Chart for PLC Editor Operation.

8 EPROM INSTALLATION

When the ladder program has been completed, compiled, and translated into 68000 machine languages, the ladder program should then be permanently recorded into the EPROM. This chapter describes how to use the EPROM writer to write the ladder program into the EPROM and how to install the EPROM. For FLASH-ROM installation.

8.1 EPROM Writer

There are several types of EPROM writers available on the market that can write the ladder program into the EPROM in 68000 machine code, for example, HI-LO ALL-03 system, Sunshine, Leap, etc. Regardless of the type of EPROM writer the user has, one common requirement is that the EPROM writer should have the ability to split "*.tsk" file into "*.odd" and "*.evn" file. The other requirement is that it can write the program into the EPROM type of 27C010 (or 27C040) with 128k bytes or 29C010 FLASH-ROM.

8.2 Writing Ladder Program Into EPROM and FLASH-ROM

The PLC Editor translates the ladder diagram to "*.tsk" 68000 machine code. The 68000 CPU is a 16-bit machine, while the EPROM or FLASH-ROM is an 8-bit device. Therefore, the files in 68000 CPU require two EPROMs or two FLASH-ROMs (even and odd) for program storage. The user can use the file splitting function (such as SPLIT2) to split the "*.tsk" file into "*.odd" and "*.evn" file. However, most of the new EPROM writer model can write "*.evn" and "*.odd" file directly into EPROM / FLASH-ROM. Note that the writing steps vary from model to model. Please refer to your FLASH-ROM writer manual for detailed steps. All HUST H2N sold so far were equipped with FLASH-ROM.

As to HUST H2N system file, it is to be splitted into “*.evn” and “*.odd” file also. Both system and PLC “*.evn” files are to be writen onto the same FLASH-ROM and both “*.odd” files onto another FLASH-ROM. The starting address when writing is “00000” for system file and “0C000” for PLC file. Following is an example showing steps to write a system and a PLC ladder diagram onto FLASH-ROM.

Example : LEAP EPROM writer (Model U-1) for writing FLASH-ROM

1. Power-on your PC and bring up FLASH-ROM writer main menu.
2. Select "Type to select IC vendor and type".
   Vendor -- select vendor's name for your FLASH-ROM. Press ENTER.
   Type "29C010". Press ENTER.
   Skip Steps 3~4 if writing onto a <u>new</u> FLASH-ROM.
3. On the main screen, select "Program".
4. Select "B" -- "EPROM device chip erase". Press ENTER to clear the existing program in FLASH-ROM.
5. On the main screen, select “Disk”.
6. Select “L” – “Load Disk Data File to Buffer”. Press ENTER.
7. Select “Binary / Machine code”. Press ENTER.

8. On "Load Disk Data File" screen, select your "*.evn" system (PLC ladder) file with the appropriate directory path. Press ENTER.
9. Start address – 00000 for system file, 0C000 for PLC. Press ENTER. Fill -- N.
   Once "N" has been entered, the "*.evn" file selected in Step 6 will be loaded into the PC memory. Press ENTER to return to the main menu.
10. Repeat Steps 5~9 for PLC file with the starting address = 0C000 for PLC.
11. On the main screen, select "Parameter". Make sure Vcc = 5.0V, Vpp = 5.0V.
12. Insert a blank EPROM into the writer.
13. Press "P" or select "Program" to start writing "*.evn" file into the FLASH-ROM.
14. Remove the EPROM from the writer and mark EVN.
15. Repeat Steps 5~9 and 10~13 for writing "*.odd" file. Mark ODD on the EPROM.
16. Press "Ctrl Esc" to quit.

8.3 EPROM Installation

The 68000 CPU board in HUST H2N controller has two slots for even and odd system and PLC ladder EPROMs. The slot designation for even system and PLC ladder EPROM is EVN and for odd system and PLC ladder EPROM is ODD. Be careful about the locations when installing those EPROMs. They must be correctly installed in the appropriate slot (see Fig 8-1) or the controller will not work properly .

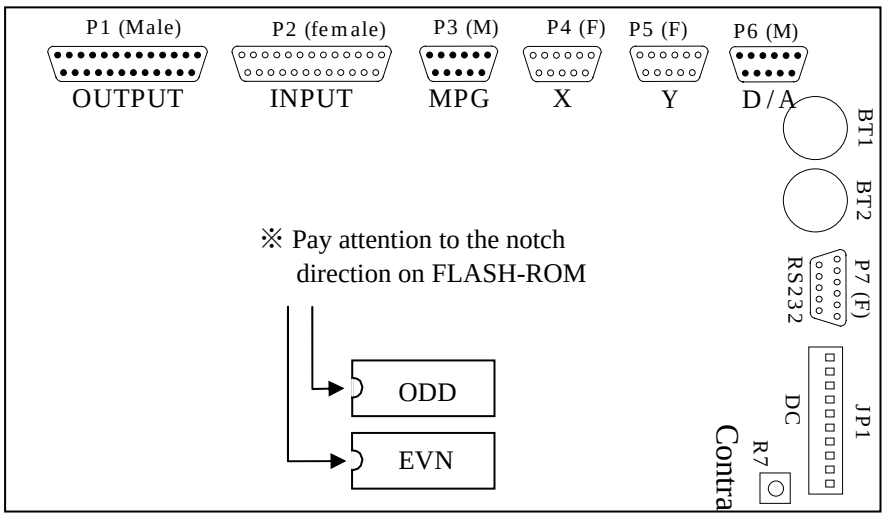


Fig 8-1 Installation of FLASH-ROM

9 Appendix A -- PC Operation with HUST CNC Controller

9.1 Applications

Via the communication ports between the HUST CNC controller and the PC, users can do the following operations from the PC terminal.

1. Command the controller to send data to PC or to execute function from PC.
2. Report or display the CNC machine status on PC screen.
3. Download/upload the MCM parameters from PC.
4. Download/upload or execute a CNC part program from PC.
5. Execute the user designated functions through the command of special S-bit, S104~S111.
6. Download/upload (write/read) the MACRO G65 variables from PC.

9.2 Equipment

1. HUST CNC controller with RS232C communication port.
2. IBM PC or equivalent with RS232 port and CRT monitor.

9.3 Communication Protocol

1. RS232C Asynchronous communication
2. Baud rate : 4800
3. Parity : Even, 2 stop-bits
4. Data bits : 7 bits
5. Data format : ASCII codes
6. RTS/CTS : Handshake
7. DC1/DC2/DC3/DC4 Handshake
   DC1~DC4 are of ASCII codes.
   DC1 represents Hex data of 11H.
   DC2 represents Hex data of 12H.
   DC3 represents Hex data of 13H.
   DC4 represents Hex data of 14H.

The method and the format that are used to establish communication between PC and HUST CNC controller are to be discussed in the following sections. The format specified has to be strictly followed.

9.4 Command the Controller to Send Data to PC or Execute Function

Format:

DC2	---	Command START.
%<CR>		
O9000<CR>	---	O9000, PC command to request data.
N010111000.....<CR>	---	Main command line starting with a letter N followed by On/Off switch (1/0). See below for 0/1 explanation.

Pxxxx<CR>	---	xxxx is the program number if the data requested is not in the current program. Omit, otherwise.
%		
DC4	---	Command END.

DC2 means PC command starts to transmit.
DC4 means PC command stops.

'0' and '1' following N act as On/Off switch for the command requested.
'1' means the requested command is On.
'0' means the requested command is Off.

The command format representations by the 0/1 combination and their meanings are shown in Table A1 and Table A2. The followings are some examples showing the use of command format from PC.

Example 1: The 2nd command, RESET

```
DC2
%<CR>
O9000<CR>
N01<CR>                 ----- Bit 2 On, RESET CNC controller
%
DC4
```

Example 2: The 3rd command, AUTO

```
DC2
%<CR>
O9000<CR>
N001<CR>                ----- Bit 3 On, AUTO mode selected
%
DC4
```

Example 3: The 6th command, CYCLE START

```
DC2
%<CR>
O9000<CR>
N000001<CR>             ----- Bit 6 On, CYCST command
%
DC4
```

Example 4: The 13th command, CNC READ program from PC

```
DC2
%<CR>
O9000<CR>
N1000000000001<CR>      ----- Bit 13 On, Read program from PC
%
DC4
```

Table A1 Command Format Representation

Item No	"On" Switch (1) Position																										
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16~19	20	21	22	23	24	25	26	27	28	29
*1	N	1																									
*2	N	0	1																								
3	N	0	0	1																							
4	N	0	0	0	1																						
5	N	0	0	0	0	1																					
6	N	0	0	0	0	0	1																				
7	N	0	0	0	0	0	0	1																			
8	N	0	0	0	0	0	0	0	1																		
9	N	0	0	0	0	0	0	0	0	1																	
10	N	0	0	0	0	0	0	0	0	0	1																
11	N	0	0	0	0	0	0	0	0	0	0	1															
12	N	0	0	0	0	0	0	0	0	0	0	0	1														
*13	N	1	0	0	0	0	0	0	0	0	0	0	0	1													
*14	N	1	0	0	0	0	0	0	0	0	0	0	0	0	1												
15	N	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1											
20	N	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0 0 0 0	1									
21	N	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0 0 0 0	0	1								
22	N	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0 0 0 0	0	0	1							
23	N	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0 0 0 0	0	0	0	1						
24	N	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0 0 0 0	0	0	0	0	1					
25	N	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0 0 0 0	0	0	0	0	0	1				
26	N	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0 0 0 0	0	0	0	0	0	0	1			
27	N	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0 0 0 0	0	0	0	0	0	0	0	1		
*28	N	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0 0 0 0	0	0	0	0	0	0	0	0	1	
*29	N	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0 0 0 0	0	0	0	0	0	0	0	0	0	1

* Format is fixed for these commands.

Notes:

1. For Item No's 13, 14, 28 and 29, the "stop report" switch (1) must be On also.
2. For Items with '*', command format is fixed (only 1 request). For Items without '*', more than one command can be requested at a time.
   For example, if request the data for program position (Item 7) and machine position (Item 8) at the same time, the command sequence is N00000011.

The meanings of command representations with corresponding item numbers mentioned above are tabulated in Table A2.

Table A2 Meaning of Command Format Representation

Item No	Meaning of Command Representation
1	Stop report
2	RESET control
3	AUTO mode
4	SINGLE mode
5	MDI mode. Input MDI data to controller. The MDI data is typed in the next line as "Gxx X-123.45 Y123.45 F123.45 .....<CR>
6	CYCLE START
7	Current program position during execution
8	Current machine position
9	Tool feedrate
10	Spindle speed
11	Current tool offset
12	Current block counter
13	The controller reads program from PC. Item No 1 'On' also.
14	The controller punches out program to PC. Item No 1 'On' also.
15	Select a part program. Pxxxx<CR> must be in the program also. Otherwise, this request is invalid. This request will be ignored if the CNC controller is currently in process or edit mode.
20	Delete part program specified by Pxxxx<CR>. Without Pxxxx<CR>, delete the current program. This request will be ignored if the controller is currently in process or edit mode.
21	Free memory (available for programming) report to PC
*22	File directory report from HUST CNC to PC
23	Current input-bit status (I-Bit)
24	Current output-bit status (O-Bit)
25	Current command-bit status (C-Bit)
26	Current status-bit status (S-Bit)
27	Current auxiliary-bit status (A-Bit)
28	The controller reads MCM data from PC
29	The controller Punches out MCM data to PC
*30	30 and above are reserved for future expansion.

Note: Items with '*' are not available at the moment.

Example 5 : The 14th command, CNC PUNCH OUT program to PC

DC2
%<CR>
O9000<CR>
N10000000000001<CR> ----- Bit 14 On, Punch out program to PC
%
DC4

Example 6: The 5th/6th command, Download MDI data to controller then Execute

DC2
%<CR>
O9000<CR>
N000011<CR> ----- Bit 5,6 On, download MDI data then execute
G01 X120.0 Y123.45<CR> ----- MDI data is specified in this line

%
DC4

9.5 Report Format of HUST CNC Controller Status

When using O9000 command to request CNC data, the data report will be displayed on the PC according to the format described below. The status report format (for main line) starts with Hxxoooo followed by an ASCII symbol and number, including On/Off (1/0) switch. The meanings and format for all ASCII symbols in HUST CNC controller are shown in Table A3.

Table A3 ASCII Symbols and the Corresponding Report Format

Symbol	Meaning of ASCII Symbol	Format
A	Part program selected does not exist	A
B	I-Bit status report	B010100001000000000.....
C	O-Bit status report	C000001110001000000.....
E	Free memory (maximum 8 digits)	E1234567
F	Tool feedrate	F1234.567
G	Current G code	G02
I	Report X-axis tool offset	I 1234.567
J	Report Y-axis tool offset	J 9876.543
K	Report Z-axis tool offset	K-1234.567
*L	Reserved	
M	Current M code	M02
N	Current block counter	N0020
*O	Reserved	
P	Part program number	P1234
*Q	Reserved	
*R	Reserved	
S	Spindle speed	S1200
T	Tool number	T2
U	X-axis program position	U 1234.567
V	Y-axis program position	V-1234.567
W	Z-axis program position	W-1234.567
X	X-axis machine position	X 5678.999 (blank means +)
Y	Y-axis machine position	Y 9876.432
Z	Z-axis machine position	Z-1234.567
!	B-axis machine position	!-1234.567
"	B-axis program position	"-1234.567
#	C-bit status Report	#010100000000010000000...
$	A-bit status Report	$010100000000010000000...
&	S-bit status Report	&010100000000010000000...

Note that all report data will be send with one line (main line) and end with an ASCII symbol, CR. The symbols with '*' are not available at the moment.

Report Format:

DC2
%<CR>
O9001<CR>
HxxoooX-1234.567Y-1234.890U677.777V999.999B101100000 .....<CR>
%
DC4

Hxxoooo has the following meaning:

"H" is a symbol.
"xx" : Error code, 00~99. See Chap 10 of Operation Manual for details.
"oooo" : Each "o" represents hexadecimal number (0~F) which has an 8-bit (1 byte) information as 0000nnnn. Only the last 4-bits "nnnn" will be used for information description. Each "n" will be either "0"(off) or "1"(on) in binary format. The hexadecimal number for "nnnn" is shown in Table A4. The range for "oooo" is 0000~FFFF. Therefore, "oooo" carries 16 bits information after decoding each "o". For easy explanation, the positions of "nnnn" will be replaced by "a-b-c-d" and the first "o" is designated as "o(1)", the second "o" as "o(2)", the third "o" as "o(3)", the fourth "o" as "o(4)".

Table A4 Hexadecimal Number

Hexadecimal number	nnnn (a-b-c-d)	Hexadecimal number	nnnn (a-b-c-d)
0	0000	8	1000
1	0001	9	1001
2	0010	A	1010
3	0011	B	1011
4	0100	C	1100
5	0101	D	1101
6	0110	E	1110
7	0111	F	1111

a -- 1st n, b -- 2nd n, c -- 3rd n, d -- 4th n.

o(1) --- abcd (nnnn)
a --- Corresponds to HUST S-bit of S080. a=1 means S080=1, which means the program execution is in process.
bcd--- Corresponds to HUST C-bit of C010, C009, and C008 for execution mode. Refer to Chap. 6 of Connecting Manual.

o(2) --- abcd (nnnn)
abc --- Not available for use at this moment.
d --- Corresponds to HUST S-bit of S088. d=1 means S088=1, which means the servo motor is On.

o(3) --- abcd (nnnn)
ab --- Not available for use at this moment.
c --- c=1 means the tool is at HOME position for Y-axis.
d --- d=1 means the tool is at HOME position for X-axis.

o(4) --- abcd (nnnn), not available for use at this moment.

Note: The total number of bytes in the status report should not exceed 256, including those used by DC2, DC4, % and O9001.

Example 1: Machine Position Report

DC2
%<CR>
O9001<CR>
HxxooooX-1234.567Y-1234.890<CR>
%
DC4

Example 2: Program And Machine Position Report

DC2
%<CR>
O9001<CR>
HxxooooX-1234.567Y-1234.890U 677.777V 999.999<CR>
%
DC4

Example 3: Input Bit (I-Bit) Status Report

DC2
%<CR>
O9001<CR>
HxxooooBhhhhhhhhhhhhhhhhhhhhhhhh<CR> --- (main line)
%
DC4

1. In addition to Hxxoooo, the main line of status report has a total of 26 bytes information, which starts with an ASCII symbol 'B' (1 byte) followed by 24 status bytes and a 'CR' (1 byte). Each "h" represents the status of I-bits and is a hexadecimal number as described in Table A4.
2. The first status byte (h) will be I000~I003 and the last status byte, I092~I095.
3. The status of each Ixxx is either '0' (Off) or '1' (On) as in binary.
   Example: If the 2nd "h" is "C", C=1100 from Table A4. Then the status of I007=On, I006=On, I005=Off and I004=Off.

Example 4: Output Bit (O-Bit) Status Report

DC2
%<CR>
O9001<CR>
HxxooooChhhhhhhhhhhhhh.....<CR> --- (main line)
%
DC4

1. In addition to Hxxoooo, the main line of status report has a total of 66 bytes information, which starts with an ASCII symbol 'C' (1 byte) followed by 64 status bytes and a 'CR' (1 byte). Each "h" represents the status of O-bits and is a hexadecimal number as described in Table A4.
2. The first status byte (h) will be O000~O003 and the last byte, O252~O255.
3. The status of each Oxxx is either '0' (Off) or '1' (On) as in binary.
   Example: If the 3rd "h" is "A", A=1010 from Table A4. Then the status of O011=On, O010=Off, O009=On and O008=Off.

Example 5: Command Bit (C-Bit) Status Report

DC2
%<CR>
O9001<CR>
Hxxoooo#hhhhhhhhhhhh.....<CR> --- (main line)
%
DC4

1. In addition to Hxxoooo, the main line of status report has a total of 66 bytes information, which starts with an ASCII symbol '#' (1 byte) followed by 64 status bytes and a 'CR' (1 byte). Each "h" represents the status of C-bits and is a hexadecimal number as described in Table A4.
2. The first status byte (h) will be C000~C003 and the last byte, C252~C255.
3. The status of each Cxxx is either '0' (Off) or '1' (On) as in binary.
   Example: If the 4th "h" is "8", 8=1000 from Table A4. Then the status of C015=On, C014=Off, C013=Off and C012=Off.

Example 6: CNC Status Bit (S-Bit) Status Report

DC2
%<CR>
O9001<CR>
Hxxoooo$hhhhhhhhhhhh.....<CR> --- (main line)
%
DC4

1. In addition to Hxxoooo, the main line of status report has a total of 66 bytes information, which starts with an ASCII symbol '$' (1 byte) followed by 64 status bytes and a 'CR' (1 byte). Each "h" represents the status of S-bits and is a hexadecimal number as described in Table A4.
2. The first status byte (h) will be S000~S003 and the last byte, S252~S255.

3. The status of each Sxxx is either '0' (Off) or '1' (On) as in binary.
   Example: If the 10th "h" is "E", E=1110 from Table A4. Then the status of S039=On, S038=On, S037=On and S036=Off.

Example 7: Auxiliary Bit (A-Bit) Status Report

```
DC2
%<CR>
O9001<CR>
Hxxoooo&hhhhhhhhhhhh.....<CR>      ---     (main line)
%
DC4
```

1. In addition to Hxxoooo, the main line of status report has a total of 66 bytes information, which starts with an ASCII symbol '&' (1 byte) followed by 64 status bytes and a 'CR' (1 byte). Each "h" represents the status of A-bits and is a hexadecimal number as described in Table A4.
2. The first status byte (h) will be A000~A003 and the last byte, A252~A255.
3. The status of each Axxx is either '0' (Off) or '1' (On) as in binary.
   Example: If the 5th "h" is "5", 5=0101 from Table A4. Then the status of A019=Off, A018=On, A017=Off and A016=On.

9.6 MCM Data Download Format (HUST read in)

When download MCM data from PC to HUST controller using MCM download command (Item 28 in Table A2), a DC1 ready signal will be sent from HUST controller to PC. Please do not start downloading MCM data to HUST controller unless you have received DC1 signal. The MCM data download format is a 7-digit integer with negative sign if necessary. The first line data after O9002<CR> will be sent to HUST controller's MCM No 1, 2nd line to MCM No 2, 3rd line to MCM No 3, and so on. Please refer to HUST CNC operator's manual for MCM parameters.

Format:

```
DC2
%<CR>
O9002<CR>
0000000<CR>          MCM No 1
0000100<CR>          MCM No 2
-0000100<CR>         MCM No 3
-0000100<CR>         MCM No 4
0000000<CR>
0000000<CR>          and so forth
.......
.......
%
DC4
```

To write (download) multiple MCM parameters (max. 6) to the controller, use the command format below**.** Use 7-digit integer for data input. Please refer to Macro G65 in Chap. 3 of Operation Manual for integer data input.

```
DC2
%<CR>
O9007<CR>
Nxxxx<CR>        ---   Parameter Index 1~98
+/-xxxxxxx<CR>   ---   Data (7-digit integer)
.....
+/-xxxxxxx<CR>   ---   Data, max. 6 consecutive data
%
DC4
```

9.7 MCM Data Upload Format to PC (HUST punch out)

No DC1 signal will be sent from HUST controller when uploading data to PC. The upload format is identical to download format. The MCM download command is Item 29 in Table A2.

Format:

```
DC2
%<CR>
O9002<CR>
0000000<CR>          MCM No 1
0000100<CR>          MCM No 2
-0000100<CR>         MCM No 3
-0000100<CR>         MCM No 4
0000000<CR>
0000000<CR>          and so forth
.......
.......
%
DC4
```

To read (upload) MCM parameters (max. 48) from controller to PC, use the command format below**.**

```
DC2
%<CR>
O9010<CR>
Nxxxx<CR>        ---   Parameter Beginning Index
Lxxxx<CR>        ---   Total number of parameters to be read (max. 48)
%
DC4
```

Read MCM Report Format:

When use O9010 command to read MCM parameters, the data reported back to the PC are shown in the following format. The number of parameters reported is determined by Lxxxx of O9010 command.

```
DC2
%<CR>
O9011<CR>
+/-xxxxxxx<CR>  ---|
.....           ---|   Parameter number reported = Lxxxx, max. 48
+/-xxxxxxx<CR>  ---|
%
DC4
```

9.8 Download/Upload Format of A Part Program

When downloading a part program from PC to HUST controller using program download command (Item 13 in Table 2), a DC1 ready signal will be sent from HUST controller to PC. Please do not start downloading process unless you have received a DC1 signal. However, no DC1 signal will be sent when uploading a part program. The download and upload formats are identical. Each block of program will occupy a line of space with a <CR> as shown below.

Format:

```
DC2
%<CR>
O0001<CR>       -----  Program O0001, program number ranging O0000 ~
O0999.
N10 G0 X0. Y0.<CR>
N20 G1 X100. Y50. F300.<CR>
......
......
M02<CR>
%
DC4
```

9.9 Special S-Bit Command

The HUST controller provides 8 special S-bits, S104~S111, for user's application from PC terminal. Customers can activate the designated functions by turning on the corresponding S-bit. These functions can be Home, Auto Cycst, MDI, Output-Bit, etc.. Note that these S-bits should be processed through PLC ladder.

The command format is shown below. The "x" in the main line is either 1=On or 0=Off. For example, execute Home by S104=1. The main line will be N10000000 or N1. In the PLC ladder, S104 will be driving the Home execution/axis C-bit, C63 and C062~C060.

DC2
%<CR>
O9005<CR>
N x x x x x x x x <CR> ----- (main line)
| | | | | | | |____ 1 → S111=1
| | | | | | |_____ 1 → S110=1
| | | | | |_____ 1 → S109=1
| | | | |_____ 1 → S108=1
| | | |_____ 1 → S107=1
| | |_____ 1 → S106=1
| |_____ 1 → S105=1
|_____ 1 → S104=1
%
DC4

9.10 Download (Write) MACRO G65 Variable

Customers can use this command to input the G65 variable data to the controller. Data should be sent one at a time with 7-digit integer (+/-) and the variable number (index), 1~4 digit. Please refer to Macro G65 in Chap. 3 of Operation Manual for integer data input.

DC2
%<CR>
O9006<CR>
Nxxxx<CR> --- Variable Index 1~99
+/-xxxxxxx<CR> --- Data (7-digit integer)
.....
+/-xxxxxxx<CR> --- Data, max. 6 datum
%
DC4

9.11 Upload (Read) MACRO G65 Variable

Customers can use this command to check the G65 variables in the controller. The beginning variable is specified by Nxxxx and the number of variables to be read is specified by Lxxxx. If L=0, 1 or omitted, only one variable will be shown on the PC screen.

DC2
%<CR>
O9008<CR>
Nxxxx<CR> --- Variable Start Index 1~127
Lxxxx<CR> --- Number of variable requested after Nxxxx. Max=127
%
DC4

Read-Variable Report Format:

When use O9008 command to read G65 variables, the data reported back to the PC is shown in the following format. The number of variables reported is determined by Lxxxx of O9008 command.

```
DC2
%<CR>
O9009<CR>
+/-xxxxxxx<CR>  ---|
.....           ---|   Data number reported = Lxxxx, max. 48 datum
+/-xxxxxxx<CR>  ---|
%
DC4
```