In this user manual we have tried to describe the matters concerning the operation of this CNC system to the greatest extent. However, it is impossible to give particular descriptions for all unnecessary or unallowable operations due to length limitation and products application conditions；Therefore, the items not presented herein should be regarded as “impossible” or “unallowable”.

# PREFACE

## Your Excellency,

GSK980D User Manual is divided into four chapters, the 1st one is the Basis Knowledge, which is mainly introduced the Signal address:, Basis command and Function command, etc. of the PLC; the 2nd one is Programming Instruction, which is mainly introduced the G, F signals based upon the functions; the 3rd one is GSKLadder User Introduction, which is mainly introduced the usage of the PLC program design software; the 4th one is Appendix, which is the List for the G, F Signal Address;

The chapter of the Programming Instruction in is mainly introduced the used signal based upon the function when the GSK980D system PLC programming is performed, which helps the design of the PLC program. The G, F signals and functions mentioned in this chapter are aimed at the GSK980TDb, GSK980TDc, GSK980TTC, GSK980TDi,GSK980MDi,GSK980MDa and GSK980MDc systems.

The described signals in the Programming Instruction are directed against the Max. axis numbers, refer to the following table:

| System Type | Max. Controllable Axis No. |
|---|---|
| GSK980TDb | 5(X axis, Z axis, Y axis, 4th axis, 5th axis,) |
| GSK980MDa | 5(X axis, Y axis, Z axis, 4th axis, 5th axis,) |
| GSK980TDc | 5(X axis, Z axis, Y axis, 4th axis, 5th axis,) |
| GSK980MDc | 5(X axis, Y axis, Z axis, 4th axis, 5th axis,) |
| GSK980TTC | 6(X axis, Z axis, Y axis, 4th axis, 5th axis, 6ix axis,) |
| GSK980TDi | 5(X axis, Z axis, Y axis, 4th axis, 5th axis,) |
| GSK980MDi | 6(X axis, Z axis, Y axis, 4th axis, 5th axis, 6ix axis,) |

**Note: GSK980TTC is the dual-channel CNC controllable system, and its physical axis interface regards as 6; the Max. controllable axis number of each channel treats as 5.**

The expression format of signal in each function includes “Symbol name”, “Signal address:”, which are shown below:

Axis movement signal MV1～MV5 (Fn102＃0～Fn102＃4) and MV1～MV5 are regardes as “Signal name”; Fn102＃0～Fn102＃4 are treated as “Signal address:”.

In order to convenient to explain when the G and F signals relevant to the controllable axis in the Programming instruction chapter can be suitable for both the Turning machine and Milling machine systems; for example: axis movement signals **<F102＃0～F102＃4>** are defined as following in the Turning machine and Milling machine systems:

| Signal | Turning machine system | Milling machine system |
|---|---|---|
| F102#0 | X axis | X axis |
| F102#1 | Z axis | Y axis |
| F102#2 | Y axis | Z axis |
| F102#3 | The 4th axis | The 4th axis |
| F102#4 | The 5th axis | The 5th axis |

In view of the above-mentioned conditions, the following axes names that are not used the programming concretely will uniformally employ the 1st axis, 2nd axis, 3rd axis, 4th axis and 5th axis for the subsequent descriptions; the corresponding cases of the Turning machine and Milling machine systems are shown below:

| Axis name | Turning machine system | Milling machine system |
|---|---|---|
| The 1st aixs | X axis | X axis |
| The 2nd axis | Z axis | Y axis |
| The 3rd axis | Y axis | Z axis |
| The 4th axis | The 4th axis | The 4th axis |
| The 5th axis | The 5th axis | The 5th axis |

When a signal expresses “Gn***＃*” or “Fn***＃*” (“*” means numbers from 0 to 9) which means that this signal is suitable for dual-channel system. For example: Signal address: is Gn100＃0, that is, the addresses are separatedly shown in the dual-channel system and channel:

n=0, at the 1st channel, Signal address: is G100＃0;

n=1, at the 2nd channel, Signal address: is G1100＃0;

This address of this signal at the single-channel system can be directly ignored n, namely, Signal address: is G100＃0.

## WARNING

 Accident may occur by improper connection and operation！This system can only be operated by authorized and qualified personnel.

## Special caution:

The power supply fixed on/in the cabinet is exclusively used for the CNC system made by GSK.

It can't be applied to other purposes, or else it may cause serious danger!

# CAUTION

## ■ Delivery and storage

- Packing box over 6 layers in pile is unallowed.
- Never climb the packing box, stand on it or place heavy objects on it.
- Do not move or drag the products by the cables connected to it.
- Forbid collision or scratch to the panel and display screen.
- Avoid dampness, insolation and drenching.

## ■ Open-package inspection

- Confirm that the products are the required ones.
- Check whether the products are damaged in transit.
- Confirm that the parts in packing box are in accordance with the packing list.
- Contact us in time if any inconsistence, shortage or damage is found.

## ■ Connection

- Only qualified personnel can connect the system or check the connection.
- The system must be earthed, and the earth resistance must be less than 0.1Ω. The earth wire cannot be replaced by zero wire.
- The connection must be correct and firm to avoid any fault or unexpected consequence.
- Connect with surge diode in the specified direction to avoid damage to the system.
- Switch off power supply before plugging out or opening electric cabinet.

## ■ Troubleshooting

- Switch off power supply before troubleshooting or changing components.
- Check the fault when short circuit or overload occurs. Restart can only be done after troubleshooting.
- Frequent switching on/off of the power is forbidden, and the interval time should be at least 1 min.

## ANNOUCEMENT

- This manual describes various possibilities as much as possible. However, operations allowable or unallowable cannot be explained one by one due to so many possibilities that may involve with, so the contents that are not specially stated in this manual shall be considered as unallowable.

## WARNING

- Before installing, connecting, programming and operating, please carefully read the product user manual and the manual from the machine tool manufacturer and strictly operate accordance with the regulations in the manual; otherwise, the product or the machine tool may be damaged, the workpiece may get rejected, even the personal injury may occur.

## CAUTION

- Functions, technical indexes (such as precision and speed) described in this user manual are only for this system. Actual function deployment and technical performance of the machine tool are designed by the machine tool manufacturer, so function configuration and technical indexes are subject to the user manual from the machine tool manufacturer.

  Refer to the user manual from the machine tool manufacturer for function and meaning of each button on the machine panel.

**All specifications and designs herein are subject to change without notice.**

# SAFETY RESPONSIBILITY

## Manufacturer's Responsibility

——Be responsible for the danger which should be eliminated and/or controlled on design and configuration of the provided CNC systems and accessories.

——Be responsible for the safety of the provided CNC systems and accessories.

——Be responsible for the provided message and advice for the users.

## User's Responsibility

——Be responsible for being familiar with and mastering the safety operation procedures through training with the safety operation of the CNC system.

——Be responsible for the dangers caused by adding, changing or altering the original CNC systems and the accessories.

——Be responsible for the dangers caused by failing to observe the provisions in the manual for operation, adjustment, maintenance, installation and storage.

**This manual is kept by the end user.**

**Thank you for supporting us in the use of GSK's products!**

# CONTENT

# I Basic Knowledge

# CHAPTER ONE SEQUENCE PROGRAM

## 1.1 PLC Specification

The different program capacity, treatment speed, function command and nonvolatile memory area addresses are differed based upon the PLC of CNC; the specifications of the GSK980D series PLC are shown below:

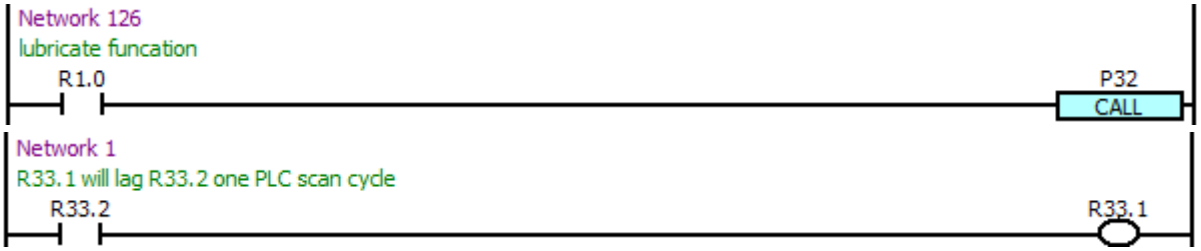
| | | GSK980TDb/MDa | GSK980TDc/MDc | GSK980TTC |
|---|---|---|---|---|
| **Specification** | | 980D-PLC | | |
| **PLC System Version** | | Ladder Diagram | | |
| **Programming Language** | | GSKLadder.exe | | |
| **Programming Software** | | .ldx | .ld2 | |
| **Programming Progression** | | 2 | | |
| **Execution Period of the First-Level Program** | | 8ms | | |
| **Average Processing Time for Basic Instructions** | | <2μs | | |
| **Maximum Steps** | | 5000 step0073 | | |
| **Programming Instructions** | | Basic instructions + function instructions | | |
| **PLC Addresses** | **Internal relay address (R)** | R0~R999 | | R0~R1999 |
| | **Information display request address (A)** | A0~A24 | | A0~A24 |
| | **Timer address (T)** | T0~T99 | | T0~T199 |
| | **Counter address (C)** | C0~C99 | | C0~C199 |
| | **Data table address (D)** | D0~D999 | | D0~D999 |
| | **Holding relay address (K)** | K0~K39 | | K0~K79 |
| | **Counter preset address (DC)** | DC0~DC99 | | DC0~DC199 |
| | **Timer preset address (DT)** | DT0~DT99 | | DT0~DT199 |
| | **Subprogram address (P)** | P0~P9999 | | P0~P9999 |
| | **Label address (L)** | L0~L9999 | | L0~L9999 |
| | **Machine→ PLC address (X)** | X0~X29 | | X0~X127 |
| | **PLC→machine address (Y)** | Y0~Y29 | | Y0~Y127 |
| | **CNC→PLC address (F)** | F0~F255 | | F0~F1999 |
| | **PLC→CNC address (G)** | G0~G255 | | G0~G1999 |

## 1.2 Sequential Program Structuring

Sequential program is a program for logic control to machine tool and relevant devices. Programs are executed by the sequence in PLC. In traditional PLC, programs are written in sequence. However, the GSK980TDc PLC integrates the traditional PLC and modern programming method by using structured programming with which methods such as sub-program, subprogram nesting and conditional branch can be applied. It has distinct advantages compared with traditional PLC.

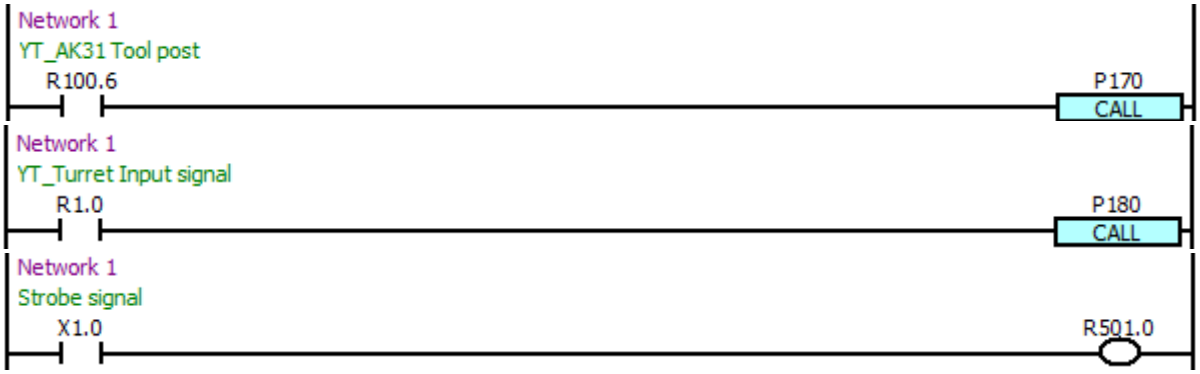
### 1.2.1 Sub-Program

In PLC programs, a specific sub-program can be called if necessary. In the following figure, when the contact R1.0 is closed, sub-programs P0032 are called.




### 1.2.2 Sub-Program Nesting

Sub-programs in CNC PLC can be nested up to 20 levels. The following figure shows that when the contact R100.6 is closed in the main program, sub-program P170 is called; in sub-program P170, when contact R1.0 is closed, sub-program P0180 is called.




### 1.2.3 Conditional Branch

The main program loops and checks whether conditions are fulfilled. If a condition is fulfilled, the corresponding subprogram is executed. Otherwise, the subprogram is skipped。

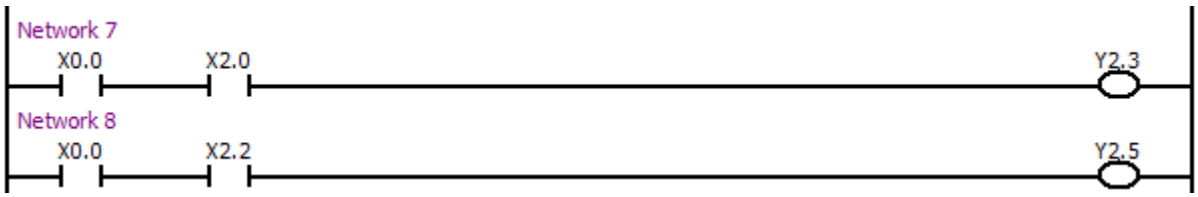
## 1.3 Execution Procedures

A written program (Ladder) can be downloaded via serial ports or U disk; the new ladder diagram will be directly performed after the power is turned on again, therefore, the CNC executes this ladder diagram, and then convert it into one kind of format which the CNC can be indenfied. The CPU, then, can be performed the coding and calculation for it, consequently, its result will be memoried at the RAM; The CPU will read each command from the memory and then perform it based upon the calculation.

### 1.3.1 Execution Procedure of Sequence Program

The PLC sequence control is realized through software, therefore, the working principles is different from the general relay circuit. The working principles of PLC sequence control should be taken into consideration during designing.

In general relay circuit, all relays can work synchronically. The following figure shows that When X0.0 is closed, and X2.0 and X2.2 are closed,Y2.3 and Y2.5 can be output; In PLC sequence control, all outputs are executed in sequence. When X0.0 is closed, and X2.0 and X2.2 are closed, Y2.3 is output in advance; then, Y2.5 is output in shortest delay time. The executions are followed by sequence.




### 1.3.2 Program Loop

PLC program is executed from the beginning to the end, and is re-executed from the beginning when it ends. This process is called program loop.

The time from beginning to the end is called loop processing cycle. Shorter processing cycle enables stronger signal response capacity.

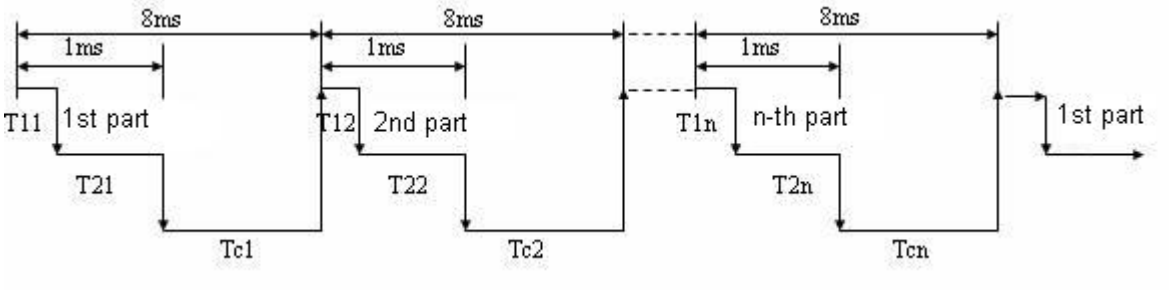
### 1.3.3 Priority of Execution

A sequence program consists of two parts: 1st level sequence and 2nd level sequence. The 1st level sequence part is less than 600 steps and operates every 8ms to process the quickly responded short pulse signal; the 2nd level sequence part operates every 8n (ms). Here n is a dividing number for the 2nd level sequence part. The 2nd level sequence part is divided automatically according to the required execution time. The cycle of execution is 8ms

The 2nd level sequence part must be divided in order to execute the 1st level sequence part. When the dividing number is n, the execution process is shown in the following figure. T11, T12, T1n are the

required time for the execution of the 1st level sequence part every 8ms for the n-th loop; T21, T22 and T2n are the required time for the execution of the 1st, 2nd, n-th division part of the 2nd sequence part for the 1st loop. Tc1, Tc2, and Tcn are the occupied time every 8ms in the 1st loop.

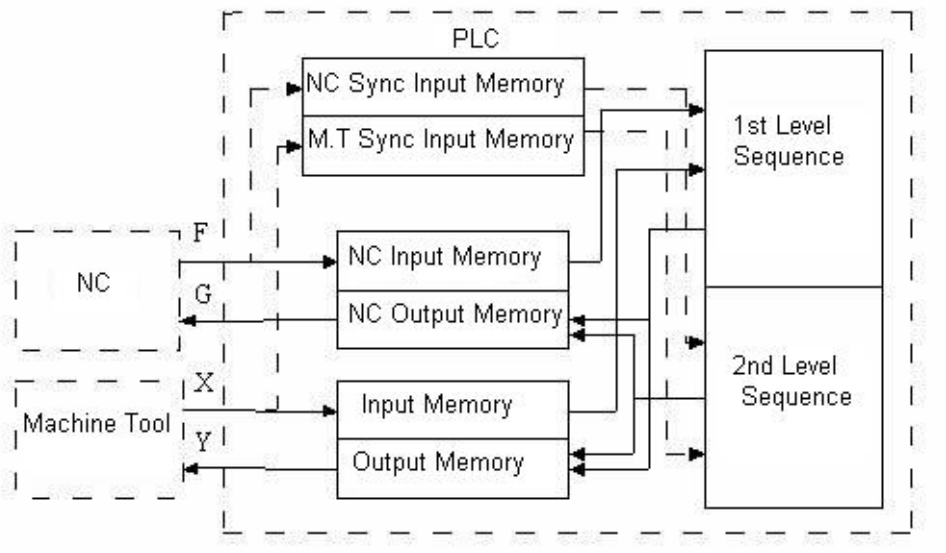
When the last division part of the 2nd sequence part has been executed, the program is re-executed from the beginning. The 1st level sequence part operates every 8ms; the 2nd level sequence part operates every 8n (ms); a loop execution time is 8n (ms).




## 1.4 Output/Input Signal Processing

X signal from machine tool and F signal from NC are input to the corresponding memory in PLC and adopted by the 1st level sequence part; meanwhile, they are input to the machine tool memory and NC memory and adopted by the 2nd level sequence part. The input signals are synchronized only in the 2nd level sequence part. The output signals of the 1st and 2nd level sequence parts are forwarded to the NC and machine tool memories, then to the I/O ports.

The signal status of NC input memory, NC output memory, machine tool input memory and machine tool output memory are displayed on the diagnosis screen.



## 1.4.1 Input Signal Processing

A: Input signal in the 1st level sequence part

F signals from NC are scanned and stored by the NC input memory at intervals of 8 ms. The 1st level sequence part directly applies these signals and process operations.

X signals from machine tool are scanned and stored by the machine tool input memory at intervals of 8 ms. The 1st level sequence part directly applies these signals and process operations.

B: The input signal in the 2nd level sequence part

The input signals in the 2nd level sequence part are the latched input signal in the 1st level sequence part. The F and X signals in the 1st level sequence part are directly adopted, therefore, the input signals in the 2nd level sequence part are lagged behind. The maximum lagging time is the 2nd level sequence part execution time.

C: The difference of input signal status between the 1st level and the 2nd level:

The status of the same input signal may be different in the 1st level and 2nd level sequence. That is, at the 1st level, processing is performed using input signal memory, and at the 2nd level, processing is performed using the 2nd level synchronous input signal memory. Therefore, it is impossible for a 2nd level input signal to delay by a cycle of 2nd level sequence execution at the worst, compared with a 1st level input signal. This must be kept in mind when writing the sequence program.

X1.0＝1 at the 1st 8ms, perform the 1st level program Y1.0＝1. Input the X1.0＝1 to the synchronism memory when the 2nd level program is performed, and then perform the last block after segmenting of the 2nd level program.

X1.0＝0 at the 2nd 8ms, perform the 1st level program Y1.0＝0. Subsequently, perform the 2nd block after segmenting of the 2nd level program, in this case, X1.0 is still the state 1 which is inputted to the memory last time, therefore, the Y2.3＝1 after executing.

## 1.4.2 Output Signal Processing

A: Output signal to NC

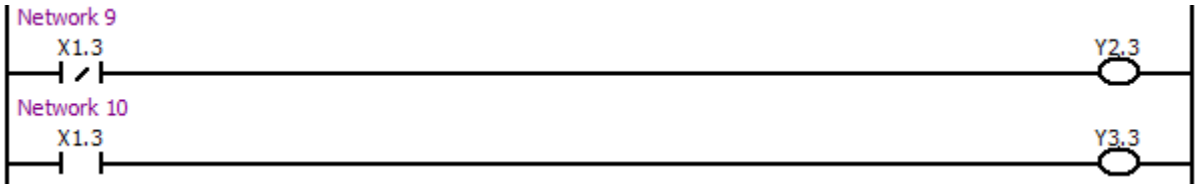
PLC outputs signals to NC memory at the intervals of 8ms, then, NC memory directly outputs the signal to NC.

B: Output signal to machine tool

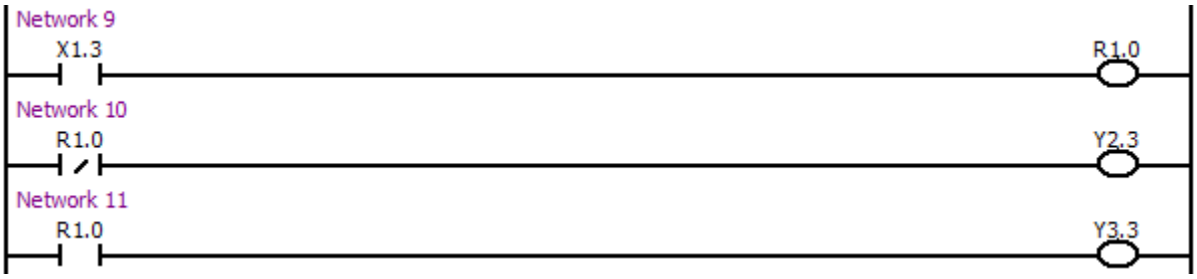
PLC outputs signals to the machine tool memory, then, the memory directly forwards the signals to the machine tool at the intervals of 2ms.

### 1.4.3 Short Pulse Signal Processing

The 1st level sequence program is used to process the short pulse signal. However, if it is less than 8ms, it means the input signal status may be changed during the execution of the 1st level sequence program, which will cause mistake.




As the above-mentioned shows, at the beginning, if the X1.3＝0, X1.3 immediately turns into 1 after Y2.3＝1, in this case, perform the next ladder diagram, so that Y3.3＝1; the Y2.3 and Y3.3 will be simultaneously regarded as 1. In order to avoid this occurrence, the short pulse signal can be treated with synchronization; refer to the following figure:




After the synchronization treatment is performed, when X1.3=1, the Y3.3=1 and the Y2.3=0; when X1.3=0, Y2.3=1, Y3.3=0, therefore, the occurrence of which the Y3.3 and Y2.3 are shared with 1 will not happen.

### 1.4.4 Interlocking

Interlocking is externally important in sequence control safety. Interlocking with the sequence program is necessary. However, interlocking with the end of the electric circuit in the machine tool magnetics cabinet must not be forgotten. Even though logically interlocked with the sequence program (software), the interlock will not work when trouble occurs in the hardware used to execute the sequence program. Therefore, always provide an interlock inside the machine tool magnetics cabinet panel to ensure operator safety and to protect the machine from damage.

## 1.5 Compilation of Sequence Program

Designing a sequence program begins with writing a ladder diagram. The ladder diagram is written using relay contact, symbols and function command codes. Logic written in the ladder diagram is entered as a sequence program. There are two sequence program entry methods. One is the entry method with PLC instructions. The other is the relay symbol method in which the sequence program is entered by

using the relay contact, symbols and the function command symbols of the ladder diagram. When the relay symbol method is used, the ladder diagram format can be used and programming can be performed without understanding the mnemonic languages.

Actually, however, the sequence program entered by the relay symbol method can be realized through the following procedures:

### 1.5.1 Interfaces Assignment (step 1)

After the control object specifications are certain and the number of input/output signal points is calculated, interfaces can be assigned. Refer to the input/output signal interface tables in the GSK980TD Series User Manual for details.

### 1.5.2 Creation of Ladder Diagram (Step 2)

Compile the software “GSKLadder.exe” by CNC ladder diagram; the controllable motions desired by machine tool are then indicated by the ladder diagram. The functions, such as the timer and the counter, can not be expressed by relay symbols, which can be shown by the specified function command symbols. The compiled ladder diagram downloads to CNC by series port, so that the read and execution can be performed.

### 1.5.3 Ladder Diagram Check (step 3)

After the ladder diagram is downloaded to CNC, it can be checked with following methods:

A: Check by simulator

Replace the machine tool with a simulator (consisting of lamps and switches). The ON/OFF of the switch represents the input signal status of machine tool; ON/OFF of lamp indicates the output signal status. Check the output signals on the basis of the activation of the lamps.

B: Check by CNC diagnosis

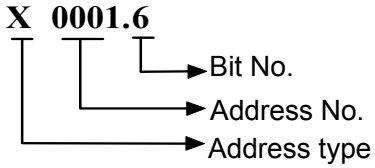
Perform different CNC functions to check whether the signal diagnosis status is consistent with the required function. Check the functions one by one to confirm the correctness of ladder diagram.

C: Check by actual operation

Perform checks by connecting the machine. Since sometimes unexpected operations may happen, arrange for safety before starting operations.

# CHAPTER TWO ADDRESS

Addresses can be distinguished the signals, separately, different addresses are corresponding to the input/output signal of the machine tool side, the input/output signal of CNC side, internal relay, counter, timer, hold-relay and data table. Each address number consists of address type, address number and bit number; refer to the following figure:




Address type: X, Y, R, F, G, K, A, T, DT, DC, C, D, L and P

Address number: Decimal system number means one byte.

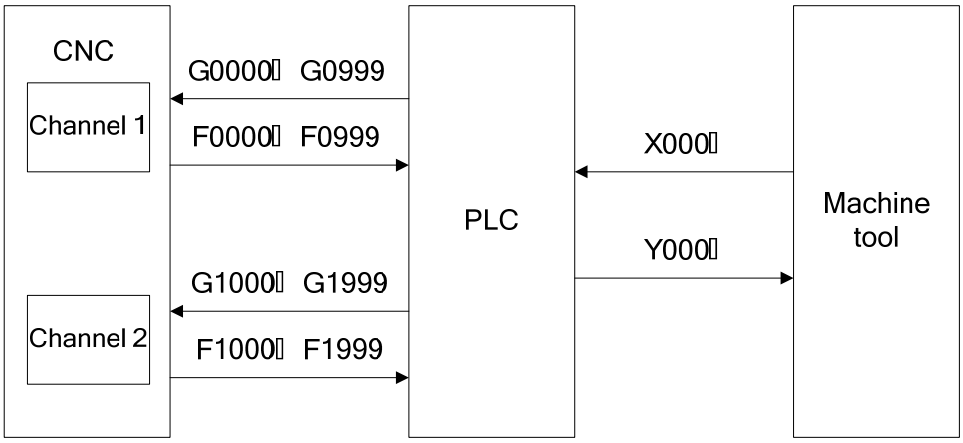
Bit number: Octal system number, 0～7 are separately expressed the bytes 0～7 bits at the front of the address number.

CNC addresses can be divided into fixed one and definable one. The signal of fixed addresses can not be altered, which only can be used by the defined CNC signal; the definable addresses can be defined different function meanings based upon the actual reqiuements; the addresses without special expression is shared with the GSK980TDb, GSK980TDc, GSK980TTC, GSK980MDa and GSK980MDc. The address explanations are shown below:

| **Add.** | **Add. Explanation** | **Add. Range** | |
|---|---|---|---|
| | | **GSK980TDb/TDc/Mda/MDc** | **GSK980TTC** |
| X | Machine tool →PLC | X0~X29 | X0~X127 |
| Y | PLC→Machine tool | Y0~Y29 | Y0~Y127 |
| F | NC→PLC | F0~F255 | F0~F1999 |
| G | PLC→NC | G0~G255 | G0~G1999 |
| R | Intermediate relay | R0~R999 | R0~R1999 |
| D | Data register | D0~D999 | D0~D999 |
| C | Counter | C0~C99 | C0~C199 |
| T | Timer | T0~T99 | T0~T199 |
| DC | Data register of the counter presetting value | DC0~DC99 | DC0~DC199 |
| DT | Data register of the timer presetting value | DT0~DT99 | DT0~DT199 |
| K | Keep/hold relay | K0~K39 | K0~K79 |
| A | Information display request signal | A0~A24 | A0~A24 |
| L | Skip mark | L0~L9999 | L0~L9999 |
| P | Subprogram mark | P0~P9999 | P0~P9999 |

The G, F signal in the above-mentioned addresses are explained as follows:

Each channel in the GSK980TTC is with the individual G, F signals; the G, F signals in the channel 1 are absolutely indentical with the one of the GSK980TDc; the G, F signal address: in the channel adds 1000 based upon the address of channel. Namely, the G, F signals range at the channel 1 is G0000～G0999, F0000～F0999; the G, F signal range at the channel is G1000～G1999, F1000～F1999. Its relationships are shown below:




In the following address expressions, “n” means that the used address position of each channel at the CNC side.

The 1st channel: n=0 (No. 0~999)

The 2nd channel: n=1 (No. 1000～1999)

**Note: 1. If there is no “n” in the expression of the G, F address, which means that the address function is irrelated with the channel.**

**2. The “n” value of the G, F signals of the GSK980TDc, GSK980MDc are regarded as 0, as well as the resolution of the channel 1.**

## 2.1 Machine Tool →PLC Address (X)

The X addresses of CNC are divided into two types: The 1st type is mainly distributed to general-purpose I/O interface of CNC, which includes the fixed addresses and definable addresses; the 2nd type is distributed to input button of the machine tool panel. The mapping address of X input signal, it is better to refer to the User Manual of each system (Installation Connection).

### 2.1.1 The 1st Type X Address

1) Fixed input address

The fixed input address is that the function from the input signal is determined by CNC software; PLC program can not be redefined the signal; for example, the mechanical zero return deceleration signal nDEC of each axis, ESP signal, SKIP signal, etc. which are shown below:

| Signal | Symbol | TD | MD | TD1 | TDa | TDb/ TDc | MDa/ MDc | TTC |
|---|---|---|---|---|---|---|---|---|
| ESP signal | ESP | X0.5 | X0.4 | X0.5 | X0.5 | X0.5 | X0.5 | X0.5 |
| X axis (the 1st axis) mechanical zero return deceleration signal | XDEC(DEC1) | X0.3 | X0.3 | X0.3 | X0.3 | X0.3 | X0.3 | X0.3 |
| Z axis (the 2nd axis) mechanical zero return deceleration signal | ZDEC(DEC2) | X1.3 | X1.2 | X1.3 | X1.3 | X1.3 | X1.3 | X1.3 |
| Y axis (the 3rd axis) mechanical zero return deceleration signal | YDEC(DEC3) | X2.3 | X1.3 | X2.7 | X2.3 | X2.3 | X2.3 | X2.3 |
| The 4th mechanical zero return deceleration signal | DEC4 | — | — | — | X2.4 | X2.4 | X2.4 | X2.4 |
| The 5th mechanical zero return deceleration signal | DEC5 | — | — | — | X2.5 | X2.5 | X2.5 | X2.5 |
| The 6th mechanical zero return deceleration signal | DEC6 | — | — | — | — | — | — | X3.4 |
| SKIP signal | SKIP | — | X1.0 | — | X3.5 | X3.5 | X3.5 | X3.5* |
| Measurement position arrival signal | XAE | — | — | — | X3.6 | X3.6 | X3.6 | X3.6* |
| Measurement position arrival signal | ZAE | — | — | — | X3.7 | X3.7 | X3.7 | X3.7* |
| Spindle alarm signal | ALM5 | — | — | — | — | X5.3 | X5.3 | X5.3 |
| Spindle 2 alarm signal | ALM6 | — | — | — | — | — | — | X8.3 |

**Note: The item with "*" can be set these addresses based upon the CNC data parameter. In the PLC program, the addresses are specified by parameter can not be redefined these signals.**

For example: The treatment of ESP signal can be connected the X0.5 address with signal, the CNC is then directly identified the signal from X0.5 address to judge whether the ESP signal occurs. When the G8.4 is enabled by PLC control, the CNC is generated the ESP alarm accordingly.

Namely: When X0.5 regards 0 which is to be inspected, CNC ESP alarm may occur;

When G8.4 treats as 0 which is to be controlled by PLC, CNC ESP alarm may occur.

2) Customized input address

The customized input address means that these functions of the input signals are defined by PLC program. User can define its function to connect both the external electric circuit and programmed ladder diagram based upon the actual requirements. The point numbers of each input signal from system are slightly different, refer to the following table:

| Product Type | Point No. | Address Range |
|---|---|---|
| GSK980TDb | 42 | X0.0~X3.7, X5.0~X5.3, X6.0~X6.5 |
| GSK980TDc | 42 | X0.0~X3.7, X5.0~X5.3, X6.0~X6.5 |
| GSK980MDa | 42 | X0.0~X3.7, X5.0~X5.3, X6.0~X6.5 |
| GSK980MDc | 42 | X0.0~X3.7, X5.0~X5.3, X6.0~X6.5 |
| GSK980TTC | 52 | X0.0~X3.7, X5.0~X5.3, X6.0~X6.5<br>X7.0~X7.5, X8.0~X8.3 |

### 2.1.2 The 2nd X Address

The button on the system operational panel corresponds to the input address, which is a kind of input signal for identifying the button operations; the scan period of these signals are longer than the input signal in the universal I/O port. In principle, the function of these signals are also defined by PLC program, however, these buttons are already assigned new special functions (which is speicified on the button panel) on the operational panel of the system; user can carry out these function in the PLC program. Refer to the User Manual (Installation & Connection) of each system for the corresponding address for each button.

## 2.2 PLC→Machine Tool Address (Y)

The output signals of the CNC are divided into two types: The 1st type address is mainly distributed to general-purpose I/O interface of CNC of which its functions of these signals can be customized by user. The 2nd address is assorted to indicator of the machine tool operational panel. As the Y output signal mapping address, refer to the User Manual (Installation & Connection) of each system.

### 2.2.1 The 1st Type Y Address

These output signals can be defined its signal meaning to connect both the external electric circuit and programmed ladder diagram based upon the user's actual situation. The point numbers of the output signals from each system are slightly different; refer to the following table:

| Product Type | Point No. | Address Range |
|---|---|---|
| GSK980TDb | 36 | Y0.0~Y3.7, Y5.0~Y5.3 |
| GSK980TDc | 36 | Y0.0~Y3.7, Y5.0~Y5.3 |
| GSK980MDa | 36 | Y0.0~Y3.7, Y5.0~Y5.3 |
| GSK980MDc | 36 | Y0.0~Y3.7, Y5.0~Y5.3 |
| GSK980TTC | 40 | Y0.0~Y3.7, Y5.0~Y5.3, Y8.0~Y8.3 |

### 2.2.2 The 2nd Type Y Address

The output address corresponding to the system operational panel indicator is a kind of output signal when the button is controlled or on one kind of state; the scan period of these signals are longer than the input signal in the universal I/O port. In principle, the function of these signals are also defined by PLC program, however, these buttons are already assigned new special functions (The functions are already defined on the operational panel). Refer to the User Manual (Installation & Connection) of each system for the corresponding address for each button.

## 2.3 PLC→NC Address (G)

G signal is the one from PLC to CNC; generally, it is sent the required signal or state signal to CNC. These signals are set by PLC program, which are read periodically and it is defined by CNC system software; refer to the appendix 1.

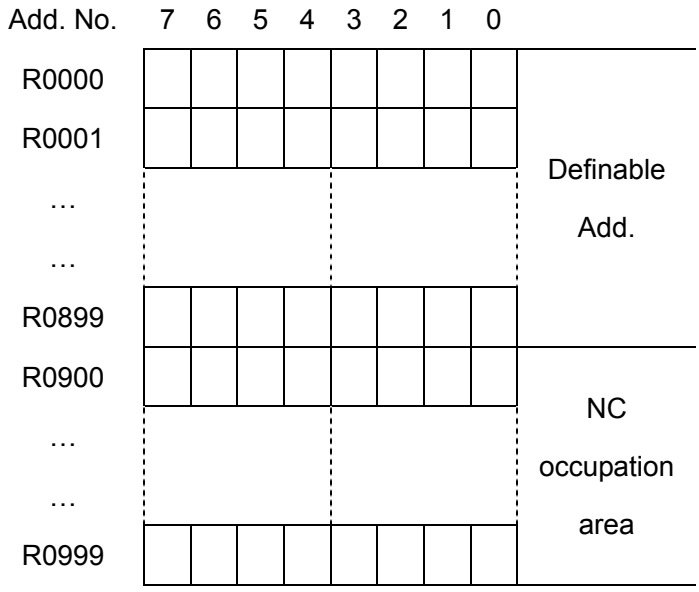
## 2.4 NC→PLC Address (F)

F signal is the one from PLC to CNC; commonly, it is sent the state signal or indication to PLC. These signals can be replaced or cleared by CNC instead of the PLC program, which are defined by CNC system software; refer to the appendix 1.

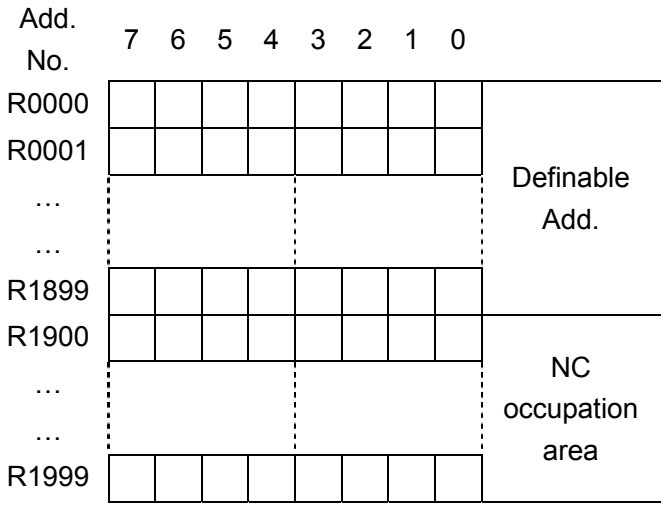
## 2.5 Internal Relay Address (R)

R register is regarded as intermediate logic coil, which is used to save the imtermediate calculation state instead of the external electric input/output. R address is treated as single byte one, its data length is 8-bit. The R register is cleared when the CNC is turned on.

The different types of the system are slightly different of the address space of the R register. The R address space of the GSK980TDb, GSK980TDc, GSK980MDa and GSK980MDc are distributed as follows:

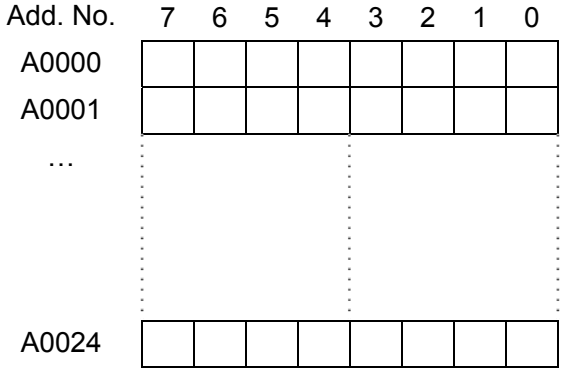


The R address space distribution of the GSK980TTC is as follows:




## 2.6 Information Display Requirement Address (A)

A signal is the one issuing to CNC from PLC, which is used for the requirement display information, for example, its PLC alarm, PLC caution, etc., as well it can be set and reset by PLC program. An address is signle byte one, its data length is 8-digit. The address A is cleared when the CNC is turned on.



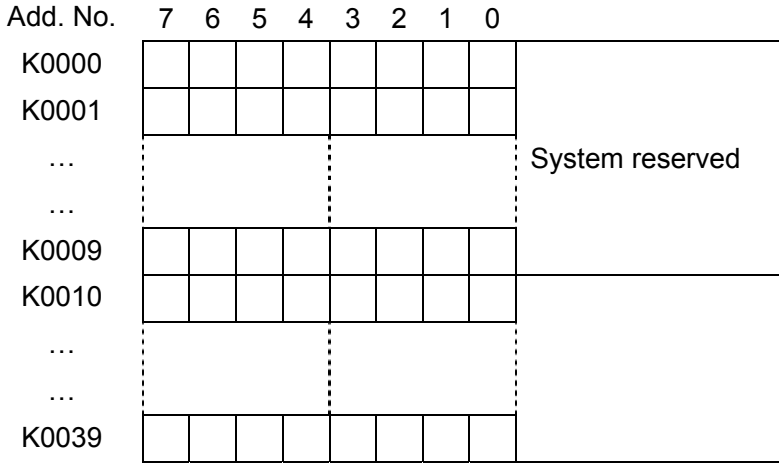

When editing the ladder diagram, each address A corresponding to an alarm number; Whether the corresponding address A is PLC alarm or PLC alert can be performed by setting the alarm numbers within different range. Alarm types from the alarm umber are slightly different based upon the different system types.

| Product Type | Alarm No. | Alarm Type |
|---|---|---|
| GSK980TDb<br>GSK980TDc | 1000～1999 | PLC alarm |
| | 2000～2999 | PLC alert |
| GSK980TTC | 1000～1999 | PLC alarm at the channel 1 |
| | 2000～2999 | PLC alarm, regardless of the channel |
| | 3000～3999 | PLC alarm shared by 2 channels |
| | 4000～4999 | PLC alarm at the channel 2 |

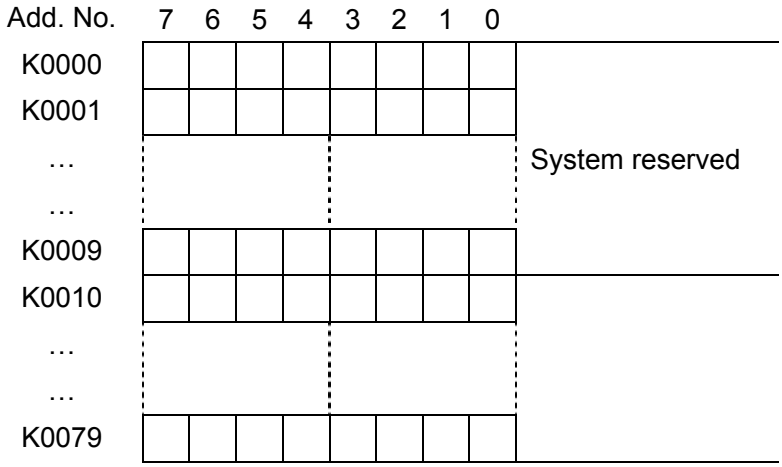
## 2.7 Hold Relay Address (K)

This address is used for holding the relay and setting the PLC parameter, its data will be memorized when the power is turned off. Usually, the K register is used for the PLC parameter and power-down saving state in the PLC; these registered can be not only modified (other than K0~K9) manually, but also replace or reset by PLC program. K address is single one; its data length is 8-bit.

The addresses of K register are slightly different depending on different system types. The K address space of the GSK980TDb, GSK980TDc, GSK980MDa and GSK980MDc are shown below:




The K address space of the GSK980TTC is shown below:




## 2.8 Counter Address (C)

This address area is usded for storing the current counting value of the counter, the data holds when the power is turned off. C address is 4-byte address and its data length is 32-bit.

The addresses spaces of counter C are slightly different depending on different system types. The C addresses space of the GSK980TDb, GSK980TDc, GSK980MDa and GSK980MDc are shown below:

| Add. No. | 31 | 30 | 29 | 28 | | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|
| C0000 | | | | | | | | | |
| C0001 | | | | | | | | | |
| … | | | | | | | | | |
| … | | | | | | | | | |
| C0099 | | | | | | | | | |

The C address space of the GSK980TTC is shown below:

| Add. No. | 31 | 30 | 29 | 28 | | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|
| C0000 | | | | | | | | | |
| C0001 | | | | | | | | | |
| … | | | | | | | | | |
| … | | | | | | | | | |
| C0199 | | | | | | | | | |

## 2.9 Counter Presetting Value Address (DC)

This address area is usded for storing the presetting value of the counter, the data holds when the power is turned off. DC address is 4-byte address and its data length is 32-bit.

The addresses spaces of counter DC are slightly different depending on different system types. The DC addresses space of the GSK980TDb, GSK980TDc, GSK980MDa and GSK980MDc are shown below:

| Add. No. | 31 | 30 | 29 | 28 | | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|
| DC0000 | | | | | | | | | |
| DC0001 | | | | | | | | | |
| … | | | | | | | | | |
| … | | | | | | | | | |
| DC0099 | | | | | | | | | |

The DC address space of the GSK980TTC is shown below:

| Add. No. | 31 | 30 | 29 | 28 | | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|
| DC0000 | | | | | | | | | |
| DC0001 | | | | | | | | | |
| … | | | | | | | | | |
| … | | | | | | | | | |
| DC0199 | | | | | | | | | |

## 2.10 Timer Address (T)

This address area is usded for storing the current numerical value of the timer. T address is 4-byte address and its data length is 32-bit.

The addresses spaces of counter T are slightly different depending on different system types. The T addresses space of the GSK980TDb, GSK980TDc, GSK980MDa and GSK980MDc are shown below:

| Add. No. | 31 | 30 | 29 | 28 | | 3 | 2 | 1 | 0 | |
|---|---|---|---|---|---|---|---|---|---|---|
| T0000 | | | | | | | | | | Powe-on clear |
| T0001 | | | | | | | | | | |
| … | | | | | | | | | | |
| T0079 | | | | | | | | | | |
| T0080 | | | | | | | | | | Power-off memory |
| … | | | | | | | | | | |
| T0099 | | | | | | | | | | |

The T address space of the GSK980TTC is shown below:

| Add. No. | 31 | 30 | 29 | 28 | | 3 | 2 | 1 | 0 | |
|---|---|---|---|---|---|---|---|---|---|---|
| T0000 | | | | | | | | | | Powe-on clear |
| T0001 | | | | | | | | | | |
| … | | | | | | | | | | |
| T0179 | | | | | | | | | | |
| T0180 | | | | | | | | | | Power-off memory |
| … | | | | | | | | | | |
| T0199 | | | | | | | | | | |

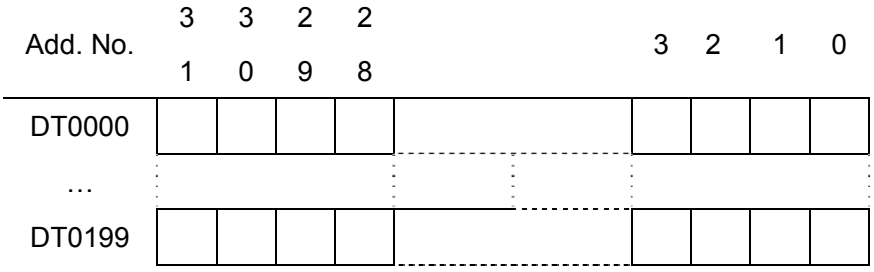
## 2.11 Timer Presetting Value Address (DT)

This address area is usded for storing the presetting value of the timer, data holds when the power is turned off. T address is 4-byte address and its data length is 32-bit.

The addresses spaces of counter DC are slightly different depending on different system types. The DC addresses space of the GSK980TDb, GSK980TDc, GSK980MDa and GSK980MDc are shown below:

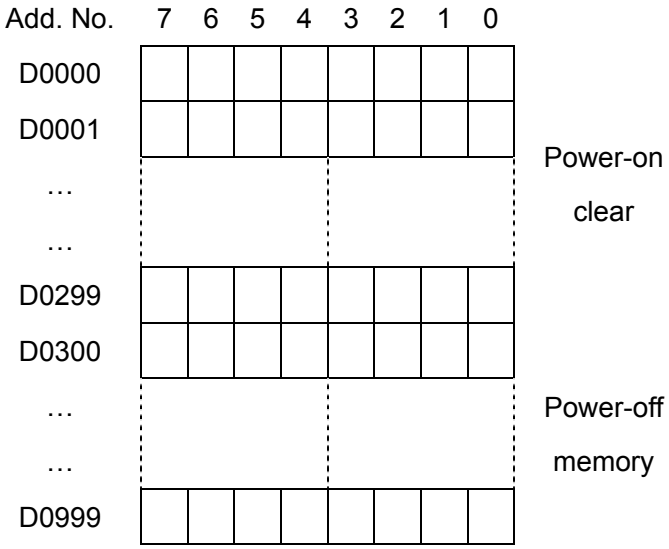
| Add. No. | 31 | 30 | 29 | 28 | | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|
| DT0000 | | | | | | | | | |
| … | | | | | | | | | |
| DT0099 | | | | | | | | | |

The DT address space of the GSK980TTC is shown below:

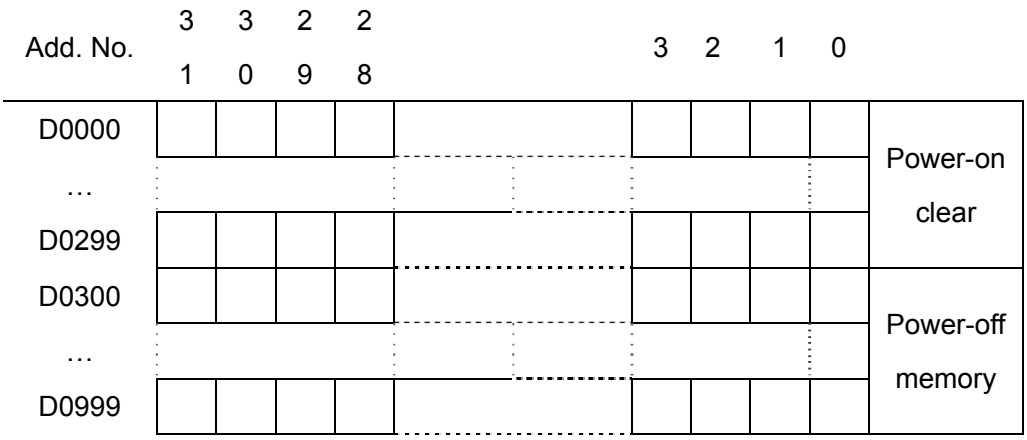



## 2.12 Data Table Address (D)

D0000～D029 address are cleared when the CNC is turned on. D0300～D0999 area data are memorized when the data is turned off. The D address space is slightly different depending on the system type; D address of GSK980TDb or GSK980MDa is single type one and its data length is 8-bit. If it regards as 4-byte address, its data length may 32-bit.




The D address of GSK980TDc, GSK980MDc or GSK980TTC is 4-byte and its data length is 32-bit.



## 2.13 Label Addresses (L)

It is used to specify the label of jump destination in JMPB instruction and the label of LBL instruction. Address range: L0～L9999

## 2.14 Subprogram Numbers (P)

Subprogram numbers are used to specify destination subprogram numbers in the CALL and SP instructions. Address range: P0～P9999

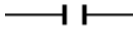
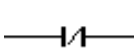
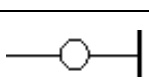
# CHAPTER THREE PLC BASIC INSTRUCTION

The basic commands are the most commonly used one when designing the sequence program, which are performed one-bit calculation. The basic commands of this PLC are shown below:

| Instructions | Functions | Components |
|---|---|---|
| LD | Read normally-open contact status | X, Y, F, G, R, K, A |
| LDI | Read normally-closed contact status | X, Y, F, G, R, K, A |
| OUT | Output coil | Y, G, R, K, A |
| OUTN | Coil output when the condition is not fulfilled | Y, G, R, K, A |
| AND | Normally-open contacts in series | X, Y, F, G, R, K, A |
| ANI | Normally-closed contacts in series | X, Y, F, G, R, K, A |
| OR | Normally-open contacts in parallel | X, Y, F, G, R, K, A |
| ORI | Normally-closed contacts in parallel | X, Y, F, G, R, K, A |
| ORB | Serial circuits in parallel | None |
| ANB | Parallel circuits in series | None |
| MPS | Push logic result to stack | None |
| MRD | Read the top-of-stack result | None |
| MPP | Pop the top-of-stack result | None |

## 3.1 LD, LDI, OUT, OUTN

- Mnemonics and functions

| Mnemonics | Functions | Symbols |
|---|---|---|
| LD | Read normally-open contact status |  |
| LDI | Read normally-closed contact status |  |
| OUT | Output coil |  |
| OUTN | Output NOT |  |

- Instruction explanation:

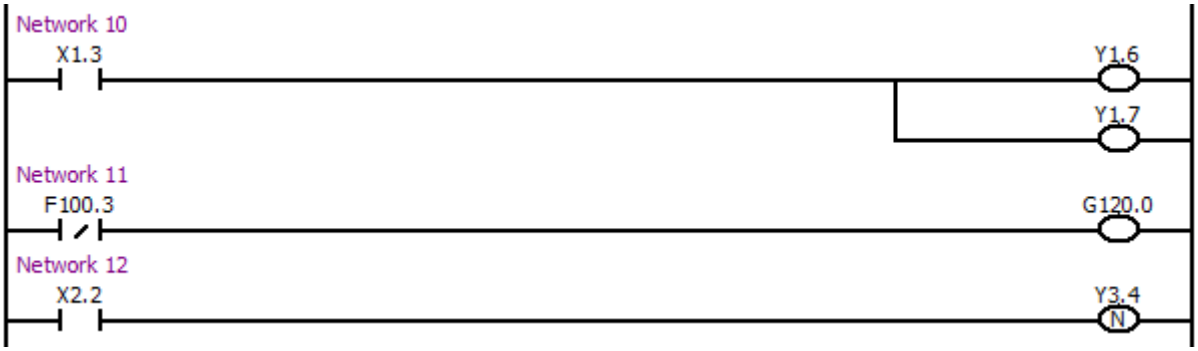
1) LD, LDI instructions are used to connect the contact to the bus. Other functions can be used on the branch start point along with the following ANB instruction.

2) OUT instruction is used to drive the output relay, internal relay coil. It cannot be used in input relay.

3) Parallel OUT instruction can be used repeatedly.

4) OUTN instruction inversely outputs the drive condition. Other usages are the same as OUT.

- Programming example:




Program explanation:

Network 1: Read the state of X2.1; it outputs Y3.6 or Y3.7 of the value is regarded as 1 at the right side; the output coils can be stood side by side;

Network 2: Read the state of F100.3; it outputs G120.0 if the value is treated as 0.

Network 3: Read the state of X2.2; it outputs Y3.4 if the value is set to 0.

Explanation:

Network 1: Read the status of X0002.1, if it is 1, Y3.6, Y3.7 is output; the output coils can be stood side by side.

Network 2: Read the status of F0100.3, if it is 0, G0120.0 is output.

Network 3: Read the status of X2.2, if it is 0, Y3.4 is output.

## 3.2 AND, ANI

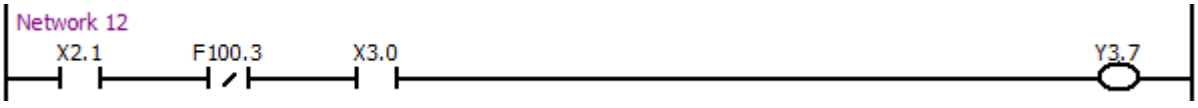
- Mnemonics and functions

| Mnemonics | Functions | Symbols |
|---|---|---|
| AND | Normally-open contacts in series |  |
| ANI | Normally-closed contacts in series |  |

- Instruction explanation:

AND and ANI instructions can connect one contact in series. The serial contact number is not limited. This instruction can be used repeatedly.

- Programming example




Explanation:

Read the status of X0002.1

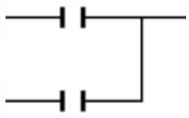
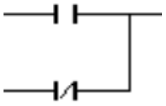
Read the status of F0100.3 and connect it with the status of X0002.1 in parallel.

Read the status of X0003.0 and connect the previous two in parallel.

When the X0002.1 and X0003.0 are 1, and F0100.3 is 0, Y0003.7 is output.

## 3.3 OR, ORI

- Mnemonics and functions

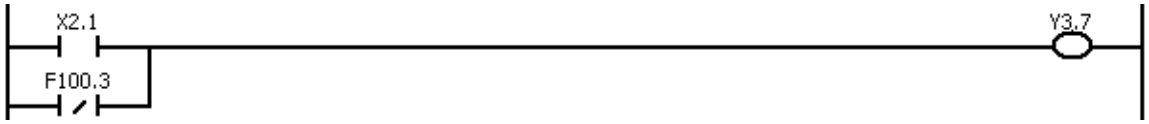
| Mnemonics | Functions | Symbols |
|---|---|---|
| OR | Normally-open contacts in parallel |  |
| ORI | Normally-closed contacts in parallel |  |

- Instruction explanation:

  1) OR, ORI instructions are used to connect one contact in parallel. If more than two contacts are connected in series, and then this kind of circuit is connected with other circuits in parallel, ORB instruction aftermentioned is used.

  2) OR or ORI means the parallel connection between the instruction step and the LD, LDI instruction steps.

- Programming example



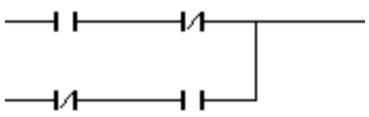

Explanation: Read the status of X0002.1

Read the status of F0100.3 and connect it with X0002.1 in parallel.

When X0002.1 is 1 or F0100.3 is 0, Y0003.7 is output.

## 3.4 ORB

- Mnemonics and functions

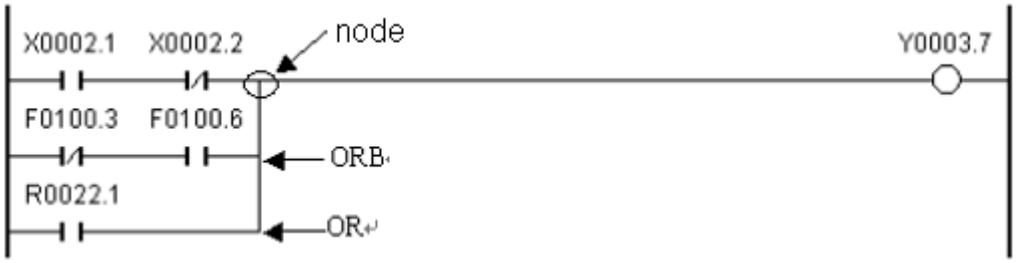
| Mnemonics | Functions | Symbols |
|---|---|---|
| ORB | Serial circuits in parallel |  |

- Instruction explanation:

  1)The circuit which is more than two contacts is connected in series is called serial circuit. When serial circuits are connected in parallel, the LD, LDI instructions are used for branch start, the ORB instruction is used for branch end.

  2)ORB instruction is an independent instruction without address.

- Programming example

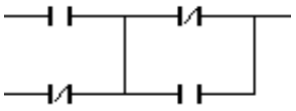



Explanation: There are three branches (0002, 0003, and 0004) from the left bus to the node. Branches 0002 and 0003 are serial circuits. When there are serial circuits connected in parallel between the bus to the node or node to node, ORB instruction is used for the all the branch ends except for the first branch. As branch 0004 is not serial circuit, OR instruction can be used.

ORB and ANB are instructions without components, they indicate the OR, AND relations of circuits.

## 3.5 ANB

- Mnemonics and functions

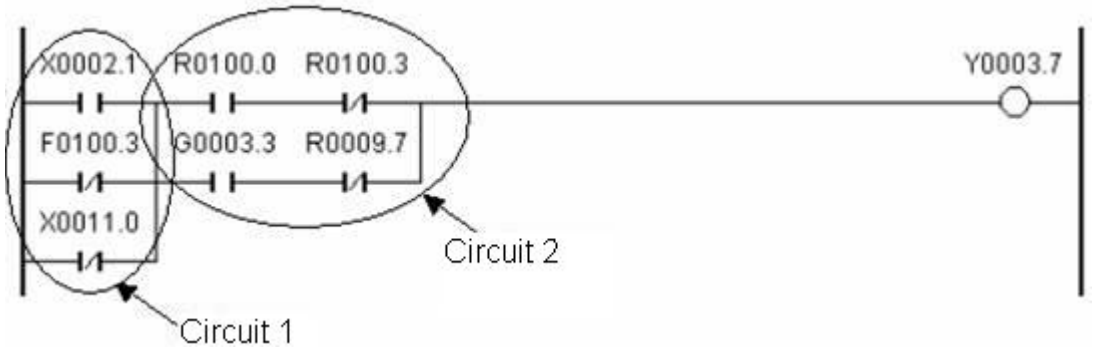
| Mnemonics | Functions | Symbols |
|---|---|---|
| ANB | Parallel circuits in series |  |

- Instruction explanation:

A: When the branch circuit (parallel circuit) is connected with the previous circuits in series, ANB instruction is used. LD, LDI instructions are used for branch start; ANB instruction is used when the serial circuit is ended and is connected in series with the previous circuits.

B: ANB instruction is an independent instruction without address.

- Programming example




Explanation: ORB indicates the serial circuits in parallel in circuit 2; ANB indicates the circuit 1 and circuit 2 are connected in series.

## 3.6 MPS, MRD, MPP

- Instruction explanation:

  1) MPS (Memory Push): This instruction stores the calculation at the top of the stack and moves other values to the bottom of the stack.

  2) MRD (Memory Read): This instruction reads the top of the stack.

  3) MPP (Memory Pop): This instruction reads, draws down the top of the stack and moves other values towards the top of the stack.

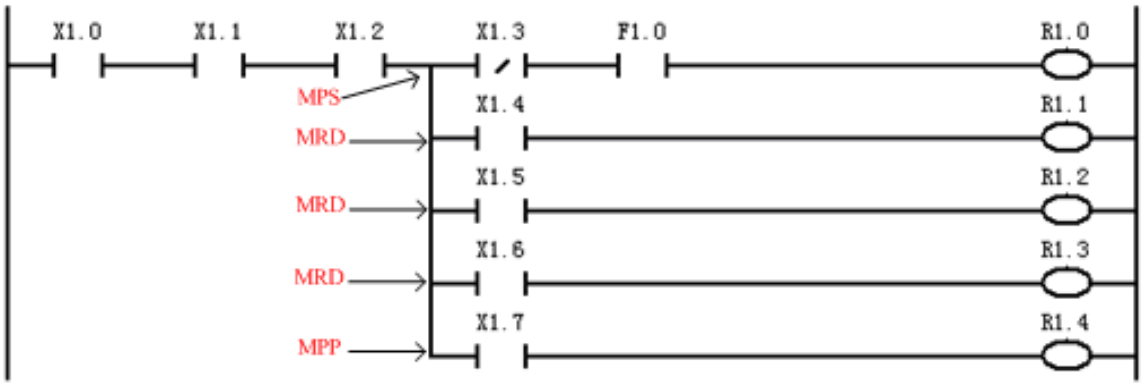
  4) The values in the stack can be used as many times as necessary. MPP is used at the last time;

  5) There is no object component in stack instructions.

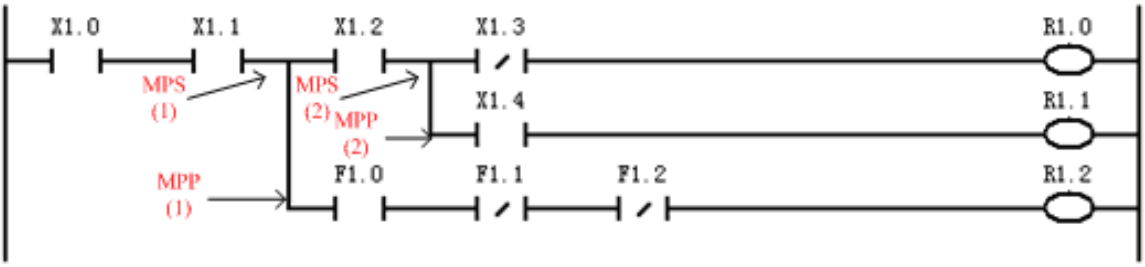
  6) MPS and MPP should be used in pairs.

- Programming example

  example 1：The stack level 1




  example 2：The stack level 2



# CHAPTER FOUR PLC FUNCTION INSTRUCTION

When some functions are hard to complete using the basic commands, the function commands then can be carried out. GSK980D Series PLC owns the function commands, refer to the following table:

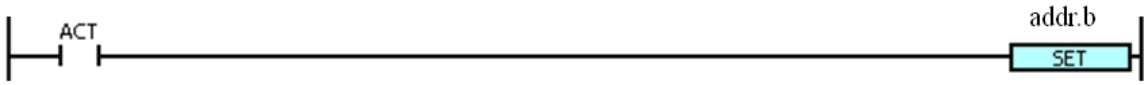
| Instruction | Function |
|---|---|
| SET | Output the logical calculation and address value after logical OR |
| RST | Output the logical calculation inversed result and address value after logical AND |
| CMP | Compare position |
| CTRC | Counter |
| TMRB | Timer |
| MOVN | Data copy |
| PARI | Parity check |
| ALT | Alternative output |
| ROTB | Binary rotation control |
| DECB | Binary decoding |
| CODB | Binary code conversion |
| JMPB | Program jump |
| LBL | Program jump label |
| CALL | Subprogram call |
| DIFU | Rising edge detection |
| DIFD | Falling edge detection |
| MOVE | Logical multiplication |
| ADDB | Binary addition |
| SUBB | Binary subtraction |
| MULB | Binary multiplication |
| DIVB | Binary division |
| WSHL | Binary data shift left |
| WSHR | Binary data shift right |
| WAND | Binary byte AND |
| WOR | Binary data OR |
| WXOR | Binary data XOR |
| WINV | Binary data inversed |
| XMOV | Binary index data transmission |
| XCHG | Binary data exchange |
| MOVBT | Binary data bit transmission |
| DSCHB | Binary data index |

## 4.1 SET

- Function

  Set the assigned address to 1.

- Format




- Control conditions

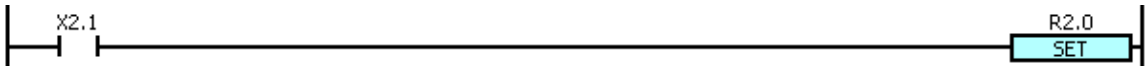
  ACT ＝0: The addr.b status remains unchanged.

  ＝1: addr.b is set to 1.

- Parameters

  addr.b: address bit. It can be contact or output coil. addr= Y, G, R, K, A.

- Example:



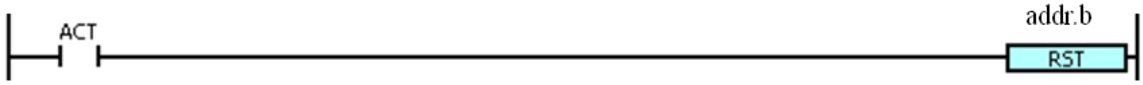

Explanation: When X0002.1 is 1, R0002.0 is set to 1; when X0002.1 is 0, the state of R0002.0 remains unchanged.

## 4.2 RST

- Function

  Set the assigned address to 0.

- Format




- Control conditions

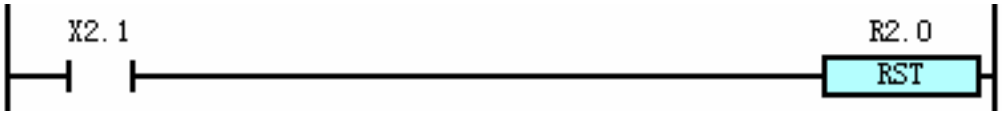
  ACT = 0: The state of addr.b remains unchanged.

  = 1: Addr.b is set to 0.

- Parameters

  addr.b: reset the address bit. It can be contact or output coil. Addr=Y, G, R, K, A.

- Example:



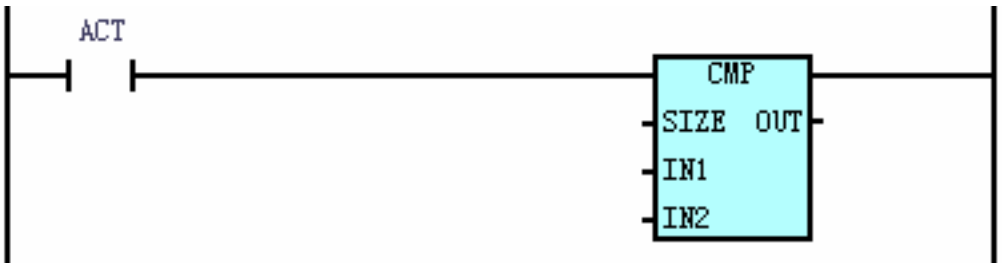

Explanation: When X0002.1 is 0, the state of R0020.0 remains unchanged; when X0002.1 is 1, R0020.0 is set to 0.

## 4.3 CMP（Binary Data Comparison）

- Function

  Compares two data values and outputs the result.

- Format




- Control conditions

  Assume that the address of OUT is represented by addr.b, then

  ACT = 0: addr.b remains unchanged

  = 1: Compares IN1 and IN2, and outputs the following results:

| | addr.(b+2) | addr.(b+1) | addr.(b+0) |
|---|---|---|---|
| IN1> IN2 | 0 | 0 | 1 |
| IN1= IN2 | 0 | 1 | 0 |
| IN1< IN2 | 1 | 0 | 0 |

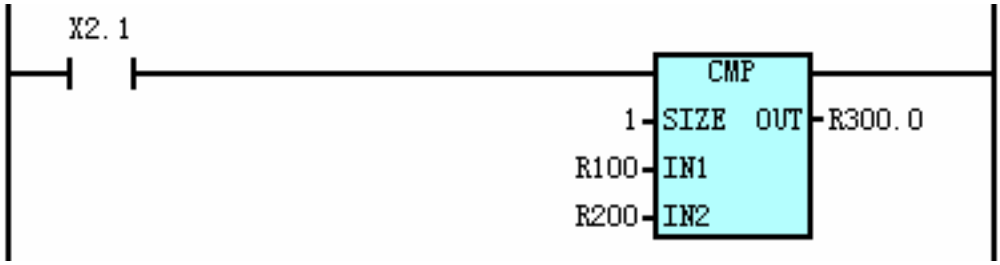
- Parameters

  Size: Specifies the size of the data. When the setting value is 1, 2 or 4, the corresponding data size is 1 byte, 2 bytes or 4 bytes.

  IN1, IN2: Compares the contents of source data 1 and 2. It can be constant or address number (but cannot be address bit, such as addr.b). The address number are R, X, Y, F, G, K, A, D, T, C, DC and DT etc.

  OUT: Compares the output result. It can be R, Y, G, K and A etc.

- Example:




Explanation: When X0002.1 is 0, the comparison is not performed; the states of R0300.0, R0300.1 and R0300.2 remain unchanged.

When X0002.1 is 1, the comparison result is shown as follows:

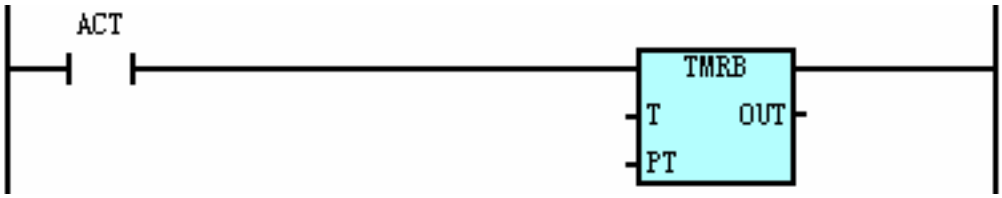
| | R0300.2 | R0300.1 | R0300.0 |
|---|---|---|---|
| R0100>R0200 | 0 | 0 | 1 |
| R0100=R0200 | 0 | 1 | 0 |
| R0100<R0200 | 1 | 0 | 0 |

## 4.4 TMRB（Timer）

- Function

  On-delay timer; the unit is ms.

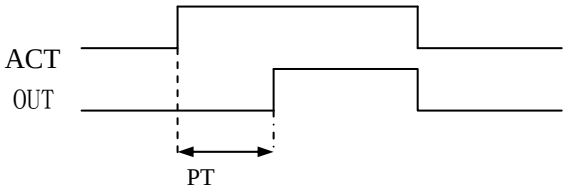
- Format




- Control conditions

  ACT =0: T and OUT are reset

  =1: T starts from 0, when the PT preset time (unit: ms) is reached, OUT=1. The logical relation is shown as follows:



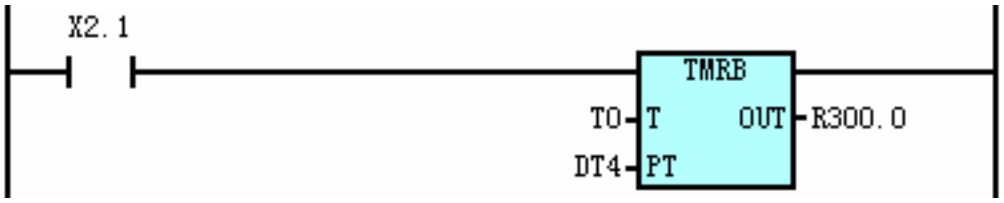

- Parameters

  T: Timer number. Range: T0000 ~ T0099

  PT: Timing constant or the data register started with DT. DT setting range: 0 ~ 21,4748,3647(ms)

  OUT: Timer output address can be R, Y, G, K and A etc.

- Example:




Explanation:

Assume that the current setting value of DT0004 is 100.

When X0002.1 is 0, both T0000 and R0300.0 are 0.

When X0002.1 is 1, after the T0000 starts timing and reaches 100 ms (set by DT0004), R0300.0 is set to 1.

## 4.5 CTRC (Binary Counter)

- Function

The data in this counter are in binary. The following functions are available:

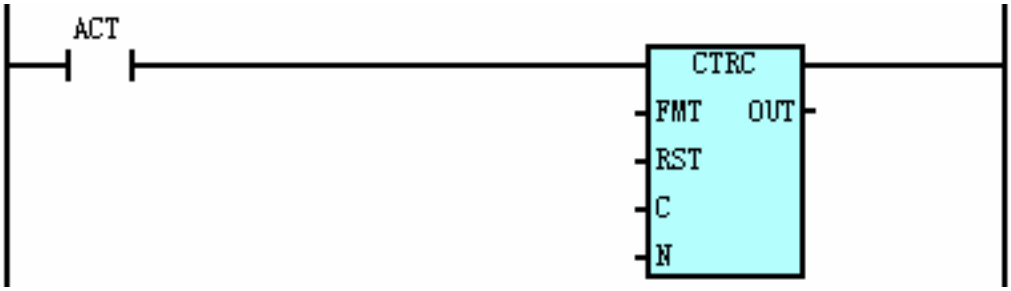
A: Preset counter: Presets the count value and outputs corresponding signal if the count reaches this preset value.

B: Ring counter: Resets to the initial value when the count signal is input after the counter reaches the preset value.

C: Up/down counter: It is the reversible counter to be used as both the up counter and down counter.

D: Selection of the initial value: Either 0 or 1 can be selected as the initial value.

- Format




- Control conditions

When ACT is rising edge (from 0 to 1):

Count up: The count up instruction counts up from the initial value. C counts up each time when rising edge appears. When the current value C reaches the preset value (N), OUT=1; when C is less than N, OUT=0. If the rising edge appears again, C counts from the initial value and meanwhile OUT=0.

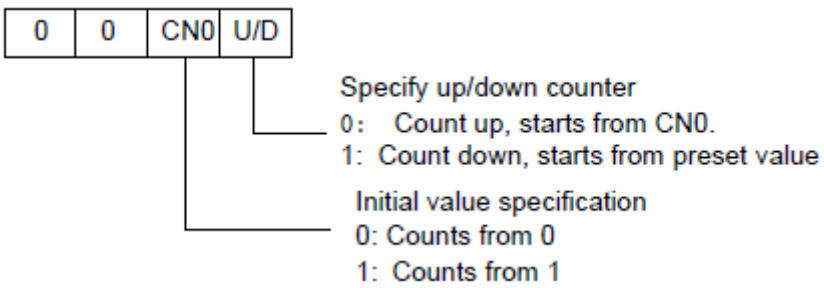
Count down: The count down instruction counts down from the preset value. C counts down each time when rising edge appears. When the current value C reaches the preset value (N), OUT=1; when C is greater than N, OUT=0. If the rising edge appears again, C counts from the initial value and meanwhile OUT=0.

When ACT=0:

C and OUT remain the same.

- Parameters

FMT: Data format



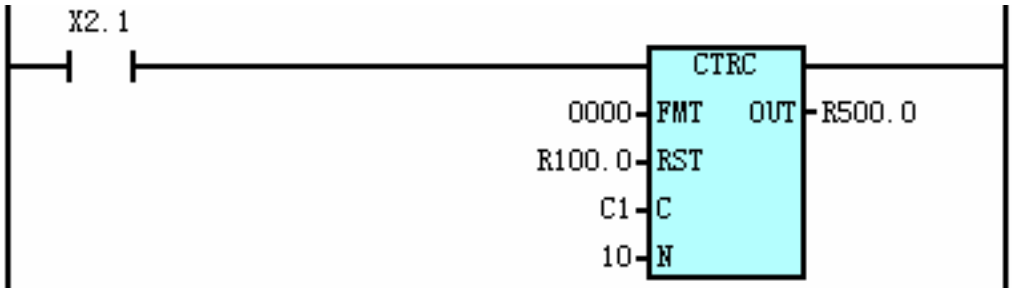
RST: When RST is 1, C=CN0 and OUT=0, (regardless of the state of ACT). RST can be X, Y, G, F, R, K or A etc.

C: Specifies the counter number which is represented with Cxxx, xxx is the number (0~99).

N: Counter preset value. It can be constant or the data register started with DC. If it is constant, the range is from 0 to 21,4748,3647.

OUT: Outputs position 1 when it reaches the count value. OUT can be R, Y, G, K or A etc.

- Example:




Explanation:

When R0100.0 is 1, C0001=0, R0500.0=0;

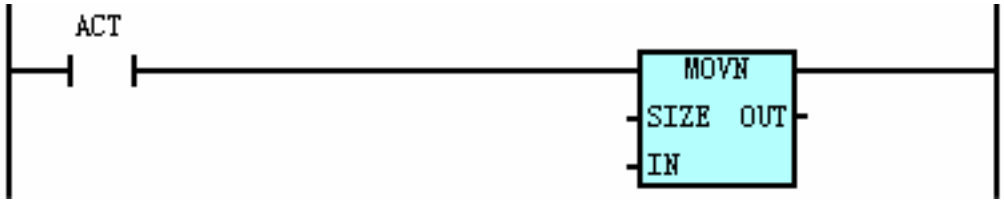
When R0100.0 is 0, each time rising edge appears at X0002.1, C0001 counts up once. When the C reaches 10, R0500.0 is set to 1. When the rising edge appears at X0002.1 again, C is reset to 0 and starts counting, R0500.0 is set to 0.

## 4.6 MOVN (Binary Data Transfer)

- Function

Transfers binary data (data copy) from a specified source address to a specified destination address.

- Format




- Control conditions

ACT = 0: OUT remains the same.

= 1: Copy values or constants from IN to OUT.

- Parameters

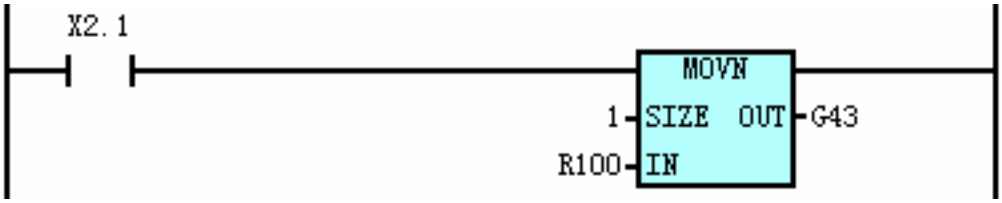
SIZE: Copy the size of data (1, 2, 4 bytes)

IN: The leading byte or constant of source data address. Addresses are R, X, Y, F, G, K, A, D, T, C,

DC, and DT etc.

OUT: The leading byte of destination address. Addresses are R, Y, G, K, A, D, T, C, DC and DT etc.

- Example:




Explanation:

When X0002.1 is 1, it transfers R0100 value (1 byte) to G0043.

## 4.7 DECB (Binary Decoding)

- Function

DECB decodes binary code data. When one of the specified eight consecutive numbers matches the code data, the corresponding output data is 1; if these numbers do not match, the output data is 0. This instruction is used for decoding data of the M or T function.

- Format


- Control conditions

ACT = 0: Reset all the output data bits.

= 1: Compare the values in IN with one of the eight consecutive data started with BASE. If they are equal, the corresponding bit in the output address (OUT) is set to 1.

- Parameters

SIZE: Specifies the size of IN1 address (1, 2, 4 bytes)

IN : Start address of decoding. The addresses are R, X, Y, F, G, K, A, D, T, C, DC and DT etc.

BASE: Compares the basic values of constants.

OUT: Outputs the comparison results. The addresses are R, Y, G, K and A etc.

- Example:


When X0002.1 =1:

If F0010=8, R0010.0=1 ;

If F0010=9, R0010.1=1 ;

……

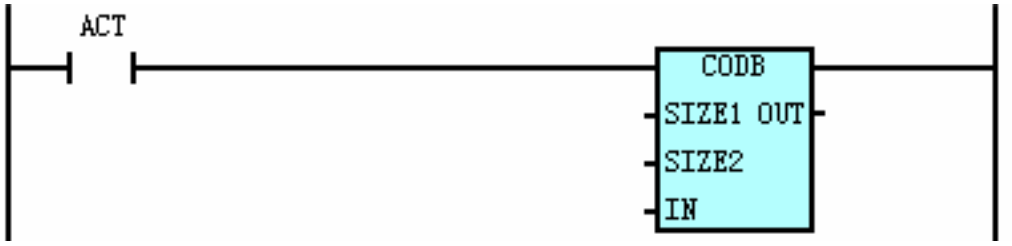
If F0010=15, R0010.7=1

## 4.8 CODB (Binary Code Conversion)

- Function

It converts data in binary format.

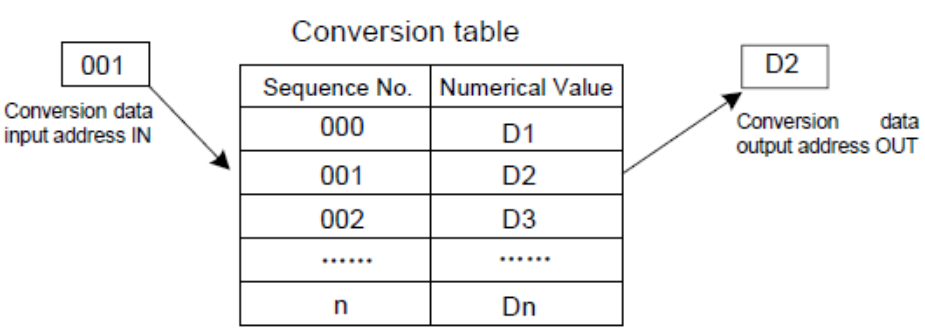
- Format




- Control conditions

ACT = 0: The values in OUT remains unchanged.

= 1: Take the values in “Convert input data address IN” as the sequence number, and obtains the corresponding conversion data from the conversion table, then outputs to the output address (OUT).



- Parameters

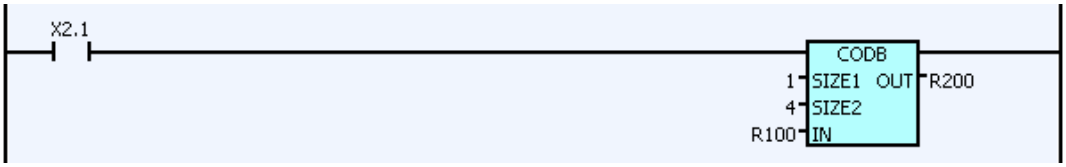
SIZE1: The binary data size and output address size of the conversion data in conversion table (1 byte, 2 bytes, 4 bytes correspondingly).

SIZE2: The size of the conversion table. The size matches with the conversion data.

IN: The input address of conversion data. It only needs one byte data. The addresses are R, X, Y, G, F, A, K and D etc.

OUT: The output address of conversion data. The addresses are R, X, Y, G, F, K, A, D, DT and DC etc.

- Example:




When X0002.1=1,

When X0002.1 = 1, R0100 = 0: R0200 = 1

When X0002.1 = 1, R0100 = 1: R0200 = 2

When X0002.1 = 1, R0100 = 2: R0200 = 3

When X0002.1 = 1, R0100 = 3: R0200 = 4

Conversion Data Table

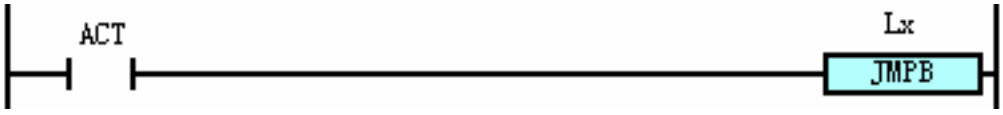
| Sequence No. | Numerical Value |
|---|---|
| 000 | 1 |
| 001 | 2 |
| 002 | 3 |
| 003 | 4 |

## 4.9 JMPB (Label Jump)

- Function

It transfers control to a Ladder immediately after the label set in a Ladder program. It has following additional functions: more than one jump instruction can be coded for the same label; jump out of subprogram is forbidden; jump forward or backward is available.

- Format




- Control conditions

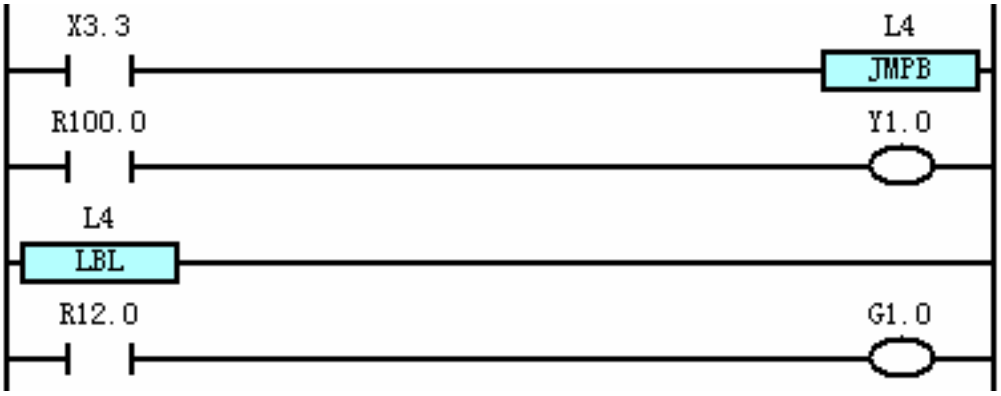
ACT = 0: The next instruction after the JMPB instruction is executed.

= 1: After jump to the specified label, the next instruction after the label is executed.

- Parameters

Lx: Specifies the jump destination. Label number should be started with L address and can be specified one value among L1~L9999.

- Example



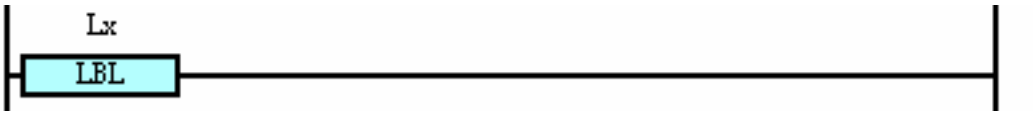

Explanation: When X0003.3 is 1, the program jump over R100.0 and executed from R12.1; if X0003.3 is 0, the execution starts from R100.0.

## 4.10 LBL (Label)

- Function

The LBL function command specifies a label in a Ladder program, i.e. the destination of JUMP. A Lx label can be specified by LBL once.

- Format




- Parameters

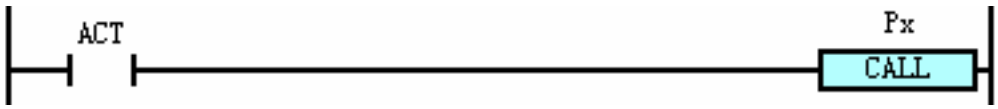
Lx: specifies the jump destination. Label number should start with L address and can be specified one value in L1~L9999.

## 4.11 CALL (Subprogram Call)

- Function

The CALL function command calls the specified subprogram. It has the following features: more than one call instructions can call for the same subprogram; the call instruction can be nested.

- Format



- Control conditions

ACT = 0: The next instruction after CALL is executed

= 1: Call the subprogram which specifies the subprogram number.

- Parameters

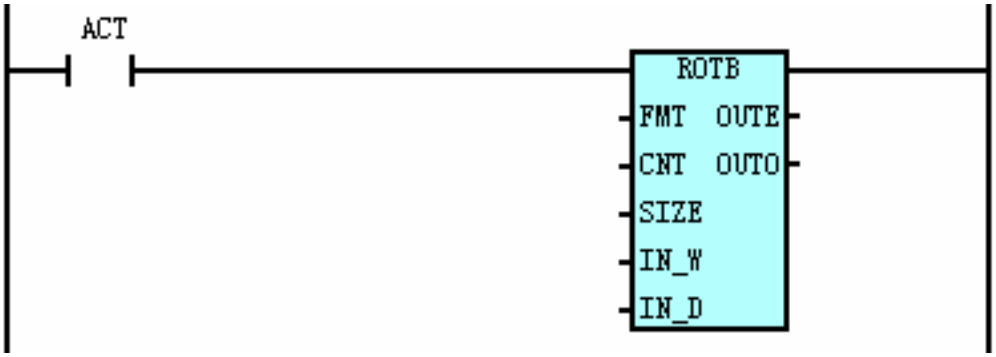
Px: Specifies the subprogram number to be called. The number should be started with P address and can be one value of P1~P9999.

## 4.12 ROTB (Binary Rotation Control)

- Function

It is used to control rotating elements including the tool post, rotary table, etc. The following functions are included: select the rotation direction of the short path; calculate the steps from the current position to destination or the steps from the previous position of current position to that of the object position; calculate the position number of the previous position of the destination.

- Format



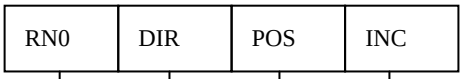

- Control conditions

ACT = 0: The instruction is not executed; OUTE and OUTO remain the same.

= 1: The instruction is executed; the results are output to OUTE and OUTO.

- Parameters

FMT: Data format:




Calculate position number or specify steps
0：calculate position number
1：specify steps

Specify calculation position
0：Calculate object position
1：Calculate the previous position

Select the short path or not:
0: No, the rotation direction is always positive, i.e. OUTO=0
1: Yes, the direction may vary

Start number of rotary table
0: Start from 0
1: Start from 1

CNT: Rotary table indexing position number.
SIZE: Specify the address size of IN-W, IN-D and OUT (1, 2, 4 bytes).

IN_W: Current position address; it used to store the current position number. The addresses are R, X, Y, F, G, K, A, D, DC, and DT, etc.

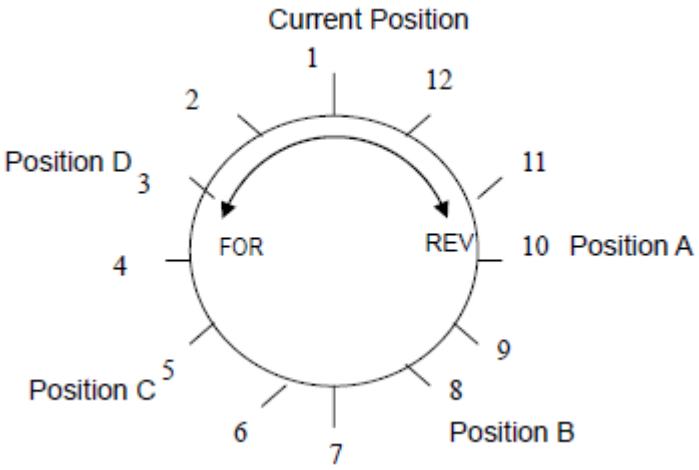
IN_D: Object position address; it is used to store object position number; the addresses are R, X, Y, F, G, K, A, D, DC and DT etc.

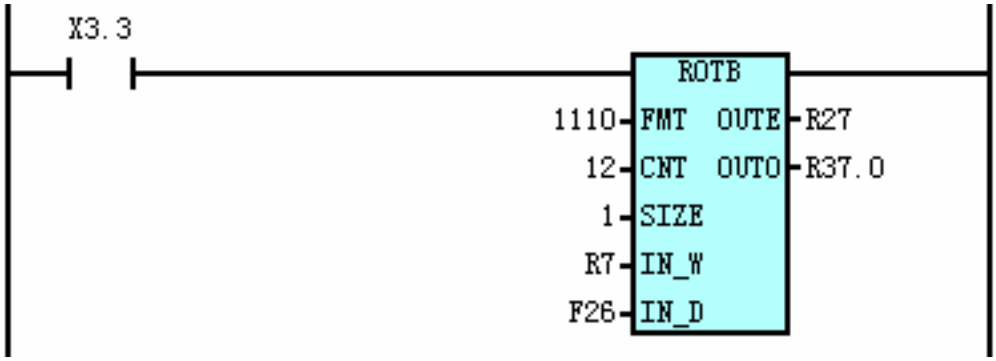
OUTE: Output address of the calculation results.The addresses are R,Y,G,K,A,D,DC and DT etc.

OUTO: Outputs the rotation direction; If the number given to the rotor is ascending, the rotation is FOR; If descending, REV.When OUTO=0, the rotation direction is positive; when OUTO=1, the rotation is inversed; the addresses are R, Y, G, K and A etc.

- Example

The following figure is a rotary table post. The current tool position is 1.






When the short path rotation is performed, the position number of the previous position before the object position is calculated. The current position number R0007=1, rotary table indexing position number CNT=12, then when X0003.3=1:

F0026 = 10, if the object position is A, R0027 = 11,R0037.0 = 1

F0026 = 8, if the object position is B, R0027 = 9,R0037.0 = 1

F0026 = 5, if the object position is C, R0027 = 4,R0037.0 = 0

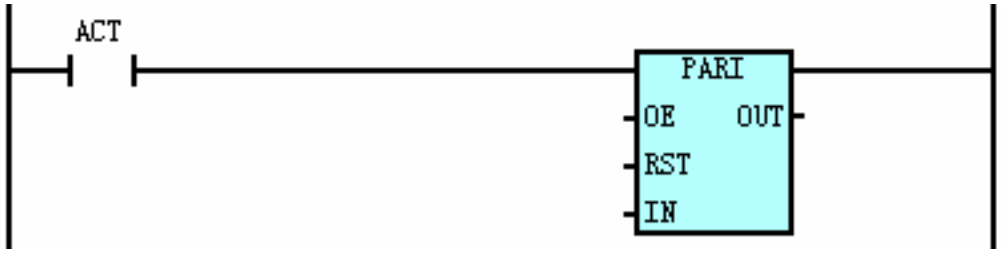
F0026 = 3, if the object position is D, R0027 = 2,R0037.0 = 0

## 4.13 PARI (Parity Check)

- Function

  It checks the parity of the input data. Only one-byte of data (8 bits) can be checked.

- Format




- Control conditions

  ACT=1: Executes the PARI instruction, performing a parity check. If the input data do not match with the one specified by OE, OUT is 1, otherwise OUT is 0.

  ACT=0: Parity checks are not performed. OUT remains the same.

- Parameters

  OE =0: Even-parity check

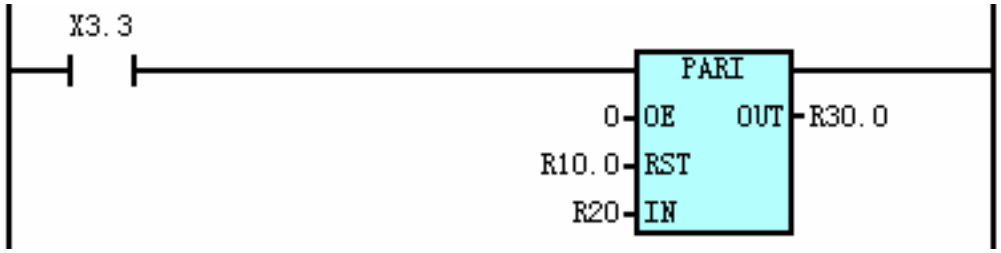
  =1: Odd-parity check

  RST: RST=1, OUT is reset to 0; the address is X, Y, G, R, F, A, and K etc. IN:

  Input data address; the address can be X, Y, G, R, F, A, K and D etc. OUT:

Parity check result output address; it can be Y, G, R, A and K etc.

- Example



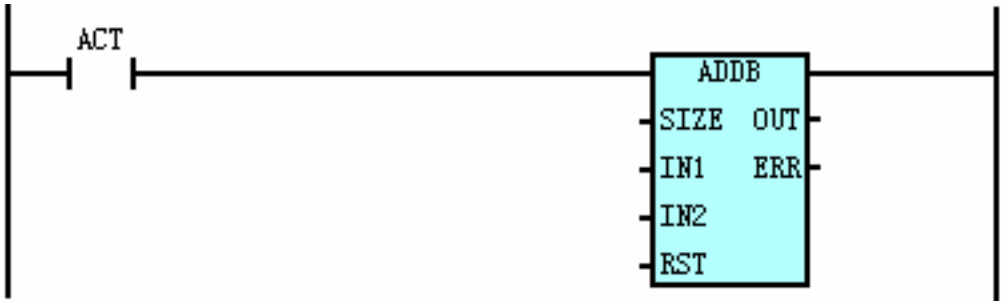

Explanation: If PARI is executed when X0003.3=1, OE=0000, parity check is performed. When R0010.0=1, R0030.0 is reset to 0, parity check is not performed. When R0010.0=0, parity check is performed. R0030.0 is 0 when R0020 data contains even parity; R0030.0 is 1 when R0020 data contains odd parity.

## 4.14 ADDB (Binary Addition)

- Function

It adds the binary data.

- Format




- Control conditions

ACT=1: OUT=IN1+IN2; If error occurs, ERR=1, otherwise ERR=0.

ACT=0: does not execute instruction; OUT and ERR do not change.

- Parameters

SIZE: 1-1 byte, 2-2 bytes 4-4 bytes

IN1: Augend, it can be constant or address. The addresses are R,X,Y,F,G,A,K,D,T,C,DC and DT etc.

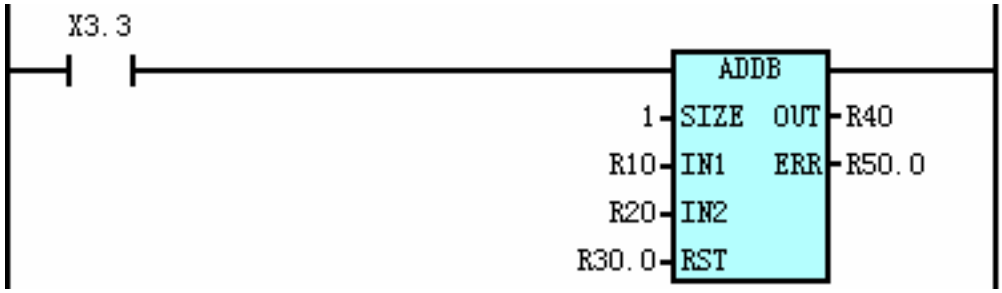
IN2: Addend, it can be constant or address. The addresses are R,X,Y,F,G,A,K,D,T,C,DC and DT etc

RST: When RST=1, the ERR is reset to 0, OUT does not change. The addresses are R,X,Y,F,G,A and K.

OUT: Address of operation result output data.The addresses can be Y,G,R,A,K,DC,DT,D,C and T etc.

ERR: Address of calculation error output, the addresses can be Y, G, R, A, and K.

- Example



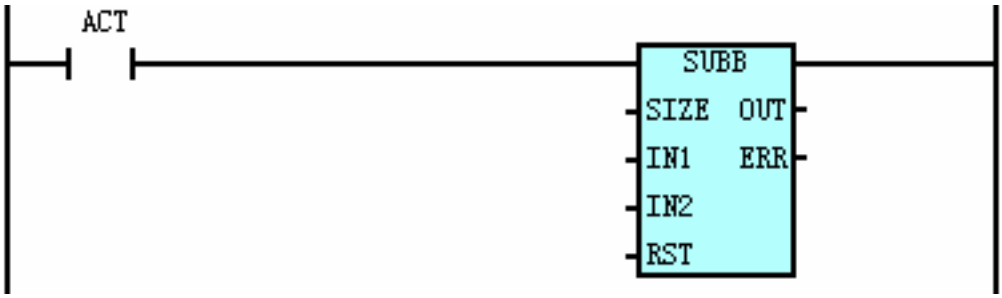

Explanation: When X0003.3=1, ADDB instruction is executed. R0040=R0010+R0020; If the operation is erroneous, R00500.0=1, otherwise R00500.0=0; When R0030.0=1, the state of R0040 remains unchanged, and R0050.0 is reset to 0.

## 4.15 SUBB (Binary Subtraction)

- Function

  It subtracts the binary data.

- Format




- Control conditions

  ACT=1: OUT= IN1-IN2 is executed; if the operation is erroneous, ERR=1, otherwise ERR=0.

  ACT=0: the instruction is not executed. OUT and ERR remain unchanged.

- Parameters

  SIZE: 1-1 byte, 2-2 bytes, 4-4 bytes

  IN1:Subtrahend; it can be constant or address;the addresses are R,X,Y,F,G,A,K,D,T,C,DC and DT etc.

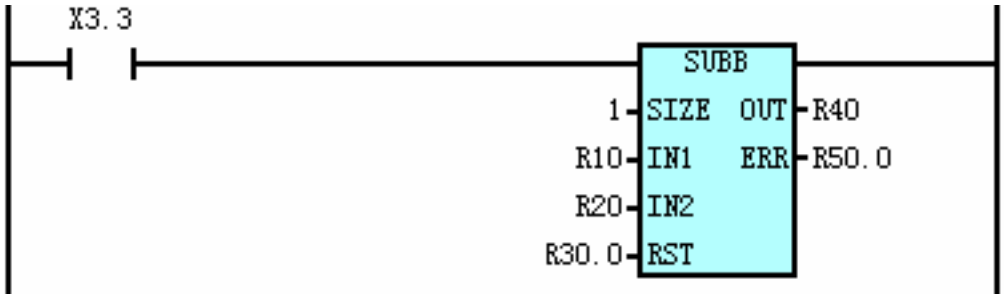
  IN2:Subtractor; it can be constant or address;the addresses are R,X,Y,F,G,A,K,D,T,C,DC and DT etc.

  RST: When RST=1, ERR is reset. The addresses are R, X, Y, F, G, A and K etc.

  OUT: the address of calculation output data. The addresses are Y,G,R,A,K,DC,DT,D,C and T etc.

  ERR: the address of calculation error output. The addresses are Y, G, R, A, and K etc.

- Example



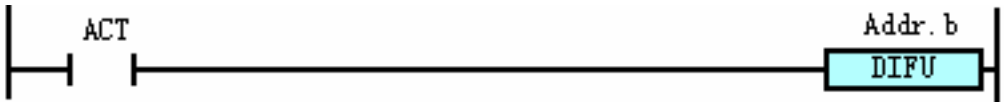

Explanation: When X0003.3=1, SUBB instruction is executed, R0040=R0010-R0020;If the calculation is erroneous, then the R0050.0 is 1, otherwise R0050.0 is 0. When R0030.0 is 1, the state of R0040 remains the same and R0050.0 is reset to 0.

## 4.16 DIFU (Rising Edge Detection)

- Function

It sets the output signal to 1 for one scanning cycle on a rising edge of the input signal.

- Format




- Control conditions

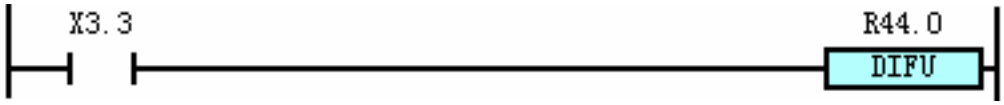
Input signal ACT: On a rising edge (0-1) of the input signal, the output signal is set to 1.

Output signal Addr.b: The output signal level remains at 1 for one scanning cycle of the ladder level where this function command is operating and changes to 0 for next scanning cycle.

- Parameter

Addr.b: calculation output address. It can be Y, G, R, A, and K etc.

- Example



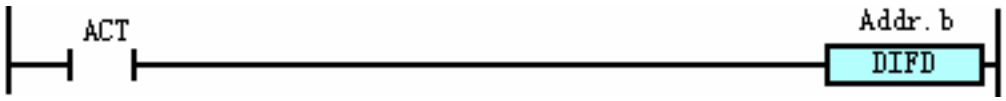

Explanation: When X0003.3 is on rising edge, R0044.0 outputs 1.

## 4.17 DIFD (Falling Edge Detection)

- Function

It sets the output signal to 1 for one scanning period on a falling edge of the input signal.

- Format




- Control conditions

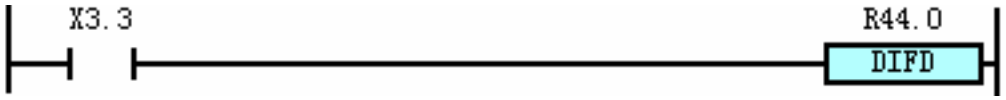
  Input signal ACT: On a falling edge (1-0) of the input signal, the output signal is set to 1.

  Output signal addr.b: The output signal level remains at 1 for one scanning period of the ladder level where this function command is operating, and for the next scanning period it changes to 0.

- Parameters

  Addr.b: the address of calculation output. It can be Y, G, R, A, and K etc.

- Example



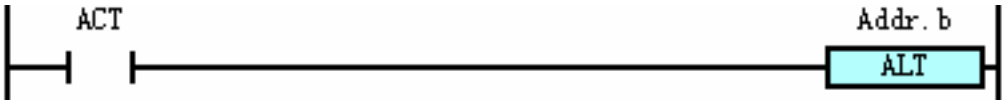

Explanation: When X0003.3 reaches falling edge, the R0044.0 outputs 1.

## 4.18 ALT (Alternative Output)

- Function

  It inversely outputs the output signal when the input signal is changing on the rising edge (0-1).

- Format



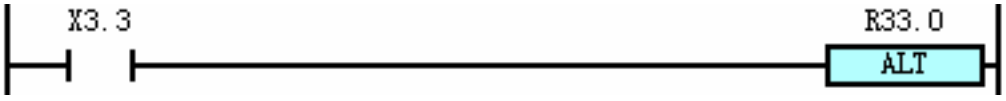

- Control condition

  The output signal addr.b is output inversely each time the input signal ACT is changed from 0 to 1.

- Parameters

  Addr.b: Output signal address; it can be Y, G, R, A and K etc.

- Example




Explanation: On every rising edge of X0003.3, the state of R0033.0 reverses.

## 4.19 MOVE (Logical Multiplication)

- Function

  ANDs logical multiplication data and input data, and outputs the results to a specified address.

- Format


- Control conditions

  ACT=1: ANDs logical multiplication data (H, L) and input data (IN), and output the result to the specified address (OUT). It can remove the unnecessary bit from 8-bit signal in the specified address.

  ACT=0: Out remains unchanged.

- Parameters

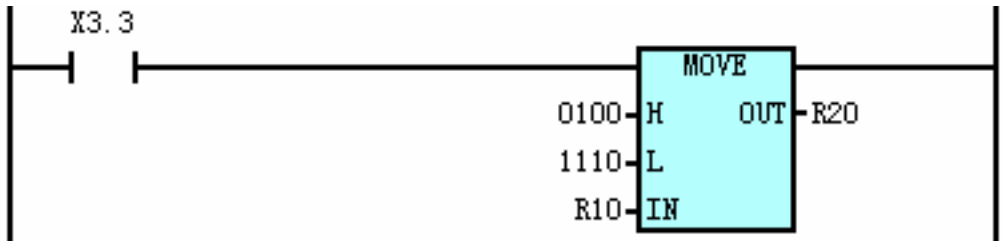
  H : High 4-bit logical multiplicator

  L : Low 4-bit logical multiplicator

  IN : the address of input data; the addresses are R, A, K, X, Y, F, G, and D etc.

  OUT: the address of output data; the addresses are R, A, K, Y, G, and D etc.

- Example



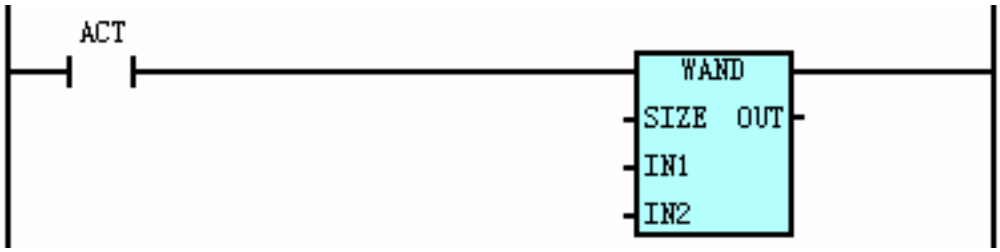

Explanation: When X0003.3 is 1, ANDs the R0010 and 01001110, and stores the result in R0020.

## 4.20 WAND (Binary Byte AND)

- Function

  A logical WAND is performed on two input data (1, 2, 4 bytes); the result is output to the OUT.

- Format




Control conditions

ACT=0: OUT value remains unchanged.

ACT=1: AND is performed on the contents of IN1, IN2, the result is output to OUT.

- Parameters

SIZE: Specifies the size of address IN1, IN2 (1, 2, 4 bytes).

IN1, IN2: The start address or constant of input data, the addresses are R, X, Y, F, G, K, A, D, T, C, DC, DT.

OUT: Address of output result. The addresses can be R, Y, G, K, A, D, T, C, DC, DT.

- Example


Explanation: When X0003.3=1, ANDs the data (8 digits) in X0 and 15 (binary: 00001111), the result is stored in R10. For example, when X0003.0=1, and X0=11000110, after the WAND instruction is executed, the result in R10 is 00000110.

## 4.21 WOR (Binary Byte OR)

- Function

It ORs two input data (1, 2, 4 bytes) by bit. The result is output to the OUT.

- Format


- Control conditions

ACT = 0, OUT value remains unchanged.

ACT = 1, ORs the contents of IN1, IN2, and the result is output to OUT.

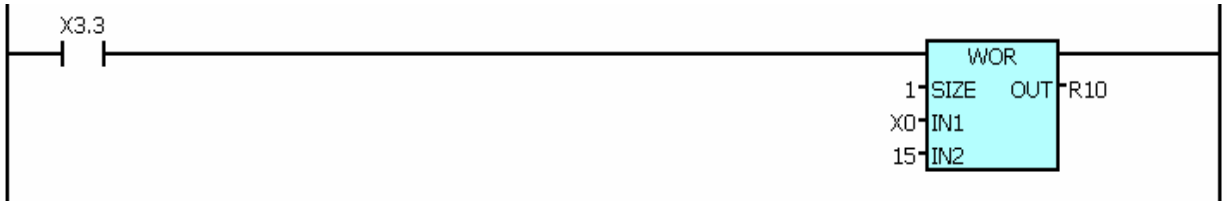
- Parameters

  SIZE: Specifies the size of IN1, IN2 addresses. (1, 2, 4 bytes)

  IN1, IN2: The start address or constant of input data. The addresses can be R, X, Y, F, G, K, A, D, T, C, DC, DT.

  OUT: The address of output result. The addresses can be R, Y, G, K, A, D, T, C, DC, DT.

- Example




Explanation: When X0003.3=1, ORs the data in X0 (8 digits) and 15 (binary: 00001111), and stores the result in R10. For example, when X0003.3=1 and X0 is 11000110, after the WOR instruction is executed, the result in R10 is 00001111.

## 4.22 WXOR (Binary Byte XOR)

- Function

  It XORs two input data (1, 2, 4 bytes) by bit, and outputs the result to OUT.

- Format


- Control conditions

  ACT=0, OUT value remains unchanged.

  ACT=1, XORs the contents of IN1, IN2, and outputs the results to OUT.

- Parameters

  SIZE:Specifies the size of addresses IN1, IN2 (1, 2, 4 bytes)

  IN1,IN2:The leading byte of input address or constant of the data. It can be R,X,Y,F,G,K,A,D,T,C, DC, DT.

  OUT: The address of result output. The address can be R, Y, G, K, A, D, T, C, DC, DT.

Example


Explanation: When X0003.3=1, XORs the data in X0 (8 digits) and 15 (binary: 00001111), the result is stored in R10. For example, when X0003.3=1 and X0 = 11000110, after the WXOR instruction is executed, the result in R10 is 00001001.

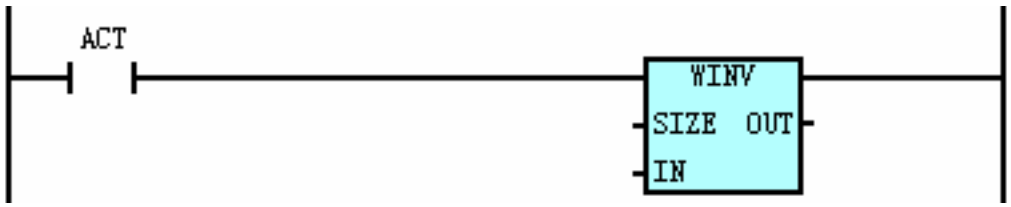


## 4.23 WINV (Binary Byte Inverse)

- Function

  It stores the data or constant of input address inversely into the OUT.

- Format




- Control conditions

  ACT=0, OUT value remains unchanged.

  ACT=1, stores the inversed value of IN into OUT.

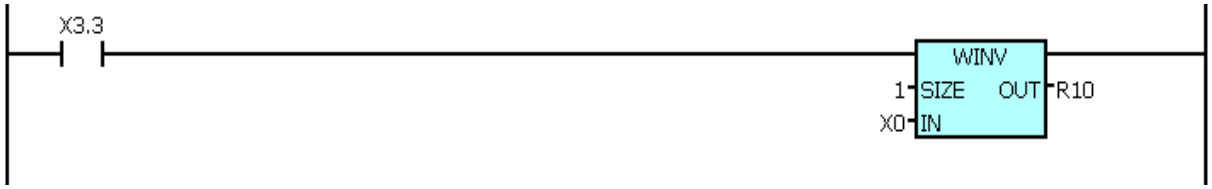
- Parameter

  SIZE: The size of data (1, 2, 4 bytes)

  IN: The leading byte of input address or constant of the data. The address can be R, X, Y, F, G, K, A, D, T, C, DC DT.

  OUT: Output address. It can be R, Y, G, K, A, D, T, C, DC, DT.

- Example




Explanation: When X0003.3=1, inverses the data (8 bits) in X0 and stores the result in R10. For example, when X0003.3=1 and X0=11000110, after the WINV instruction is executed, the result in R10 is 00111001.

## 4.24 WSHL (Binary Data Shift Left)

- Function

It is shift left instruction of two input data (1, 2, 4 bytes) by specified bits. The result is output to the OUT address.

- Format


- Control conditions

ACT=0, OUT value remains the same.

ACT=1, the value in IN is shifted left N bits, and the result is output to OUT.

- Parameters

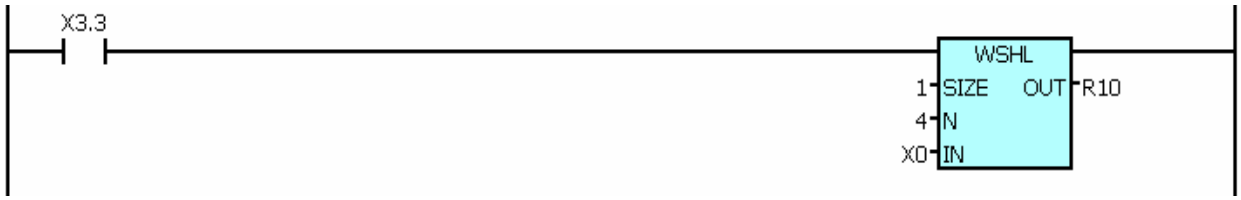
SIZE: Specify the size of data in IN (1, 2, 4 bytes)

N : The address or constant of shifted data. The address can be R,X,Y,F,G,K,A,D,T,C,DC,DT.

IN: The leading byte of input address or constant of the data. It can be R, X, Y, F, G, K, A, D, T, C, DC, DT.

OUT: The address of output result. It can be R, Y, G, K, A, D, T, C, DC, DT.

- Example




Explanation: When X0003.3=1, it shifts left 4 bits of the data (8 bits) in X0, and the result is stored in R10. For example, when X0003.3=1, and X0=11000110, after the WSHL instruction is executed, the result in R10 is 01100000.

## 4.25 WSHR (Binary Data Shift Right)

- Function

It is two input data command instructing shift right (1, 2, 4 bytes) in specified bits. The result is output to the OUT.

- Format


- Control condition

  ACT=0, OUT value remains the same.

  ACT=1, the value in IN is shifted right N bits, and the result is output to OUT.

- Parameters

  SIZE: Specify the size of data in IN (1, 2, 4 bytes)

  N: The address or constant of shifted data. The address can be R, X, Y, F, G, K, A, D, T, C, DC, DT.

  IN:The leading byte of input address or constant of the data. It can be R, X, Y, F, G, K, A, D, T, C, DC, DT.

  OUT: The address of output result. It can be R, Y, G, K, A, D, T, C, DC, DT.

- Example

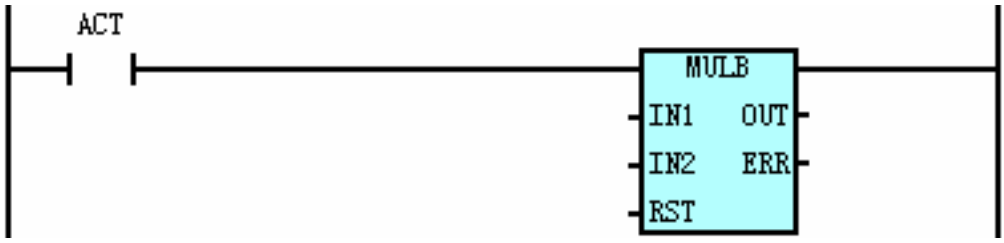

Explanation: When X0003.3=1, it shifts right 4 bits of the data (8 bits) in X0, and the result is stored in R10. For example, when X0003.3=1, and X0=11000110, after the WSHL instruction is executed, the result in R10 is 01100000.

## 4.26 MULB (Binary Data Multiplication)

- Function

  It multiplies two input data (16 bits), and the resulted product (32 bits) is stored in the OUT address (32 bit).

- Format




- Control conditions

  RST = 0: ERR and OUT remain unchanged.

RST = 1: Reset ERR and OUT.

ACT = 0: OUT value remains unchanged.

ACT = 1: Multiplies the values of IN1 and IN2, the result is output to OUT address.

- Parameters

IN1, IN2: The leading byte of input address or constant of a mulplicator; it can be R, X, Y, F, G, K, A, D, T, C, DC, DT; if single-byte address (8 bits),such as R, X, Y, F, G, K, A, D is used, two consecutive bytes is used as the mulplicator; if double-byte address (32 bits) such as T, C, DC, DT is used, the low 16 bits is used as the mulplicator.

OUT: The address of output result. It can be R, Y, G, K, A, D, T, C, DC, DT. RST:

The input address (bit address) of instruction reset signal

ERR: The output address (bit address) of calculation error. It can be R, Y, G, K, A.

- Example

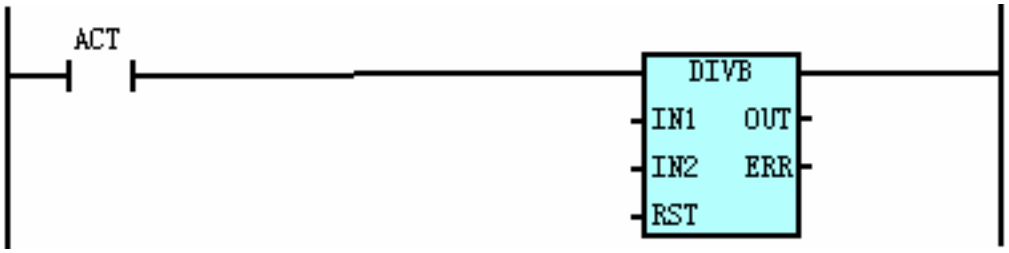

Explanation: When X0003.3=1, the data consisting R100 and R101 (16 bits, R101 takes up high 8 bits, R100 takes up low 8 bits) is multiplied by constant 40000, and the product is stored in the 4 bytes whose start address is R200 (R200, R201, R202, R203, R200 takes up low 8 bits).

## 4.27 DIVB (Binary Data Division)

- Function

It divides two input data (16 bits), and the results (32 bits including high 16-bit remainder and low 16-bit quotient) are stored in the OUT address (32 bit).

- Format




- Instruction format

| DIV | IN1 | IN2 | RST | OUT | ERR |
|---|---|---|---|---|---|

- Control conditions

RST = 0: ERR and OUT remain unchanged.

RST = 1: Reset ERR and OUT.

ACT = 0: OUT value remains unchanged.

ACT = 1: Divides the values of IN1 and IN2, the result is output to OUT address.

Parameters

N1, IN2: The leading byte of input address or constant of a mulplicator; it can be R, X, Y, F, G, K, A, D, T, C, DC, DT; if single-byte address (8 bits),such as R, X, Y, F, G, K, A, D is used, two consecutive bytes is used as the divisor; if double-byte address (32 bits) such as T, C, DC, DT is used, the low 16 bits is used as the divisor.

OUT: The address of output result. It can be R, Y, G, K, A, D, T, C, DC, DT.

RST: The input address (bit address) of instruction reset signal

ERR: The output address (bit address) of calculation error. It can be R, Y, G, K, A.

- Example

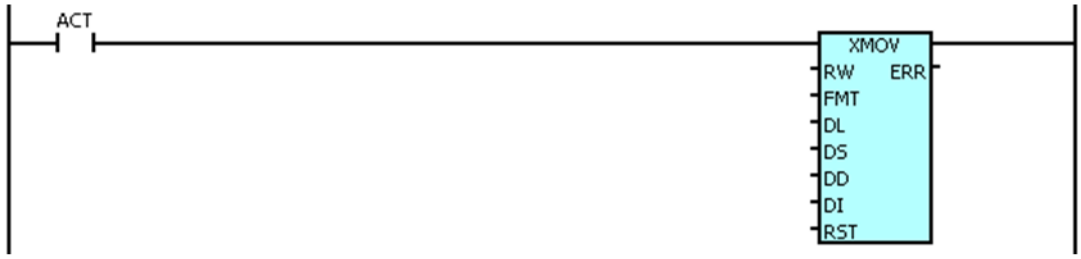

Explanation: When X0003.3=1, it divides the data (16 bits, R101 takes up high 8 bits, R100 takes up low 8 bits) consisted of R100, R101 and constant 1000, and the result is stored in the 4 bytes whose start address is R200 (R200, R201, R202, R203, R200 takes up low 8 bits), the remainder (16 bits) is stored in two bytes (R202, R203, R202 takes up low 8 bits) whose start address is R202.

## 4.28 XMOV (Binary Index Data Transmission)

- Command function:

  Read or change the content of data table.

- Ladder diagram format




Controllable condition

RST=0：DS, ERR and DD are invariable;

RST=1：ERR resetting;

ACT=0：DS and DD are invarible, ERR regards to 0;

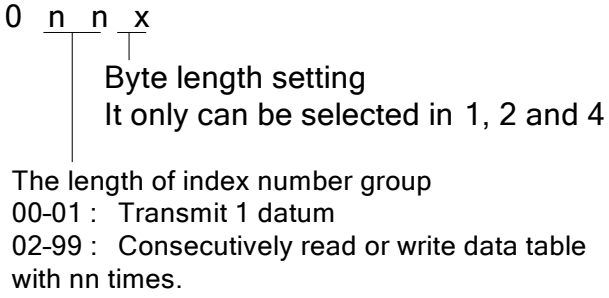
ACT=1: When RW=0, read the data from data table DS to the DD based upon the DI index; when RW=1, write data from DD to the data table DS based upon the DI index. ERR treats as 1 when the execution command is incorrect; it is regarded as 0 without any error.

- Relevant parameter

RW=0: Read data from the initial address data table specified by DS to the one specified by DD.

=1: Write the data to the initial address data table specified by DD to the one by DS.

FMT: Perform the command format specification by 4-bit constant, its setting format:




DL: Specify the data table length based upon the initial address against DS.

The corresponding setting value can be set based upon the speicifed byte length setting in the FMT, refer to the following:

1 byte length：1～255

2 bytes length：1～16384

4 bytes length：1～16384

The setting value is constant or X, Y, F， G, R, K, A, D, C, T, DC, DT (Byte address).

DS: Data table initial address; the $0^{th}$ datum of the data table; specify the initial address of the read/write data table. The byte number to be occupied the register space is: (the specified byte length in FMT) x (the specified data table length by DL), its setting value are Y, G, R, K, A, D, C, T, DC and DT (Byte address).

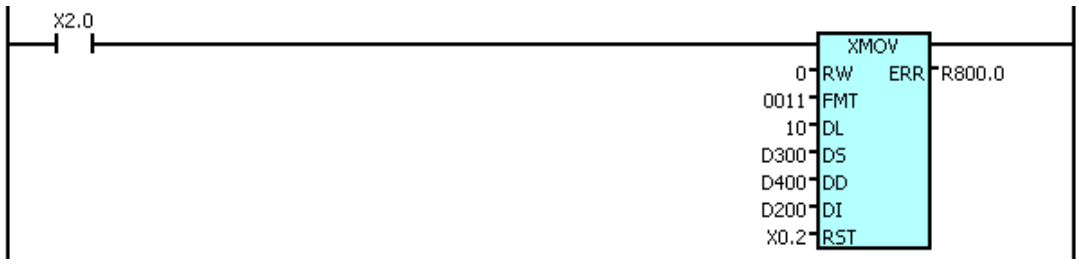
DD: Input/output data storage address. It means the register address of storing the read result during reading; it means the register address for stroing write value during writing. The setting value are Y, G, R, K, A, D, C, T, DC and DT (Byte address).

DI: Set the register address for storing the index value, the data occupies the specified byte length in the FMT and its setting value are Y, G, R, K, A, D, C, T, DC and DT (Byte address).

RST: Resetting input bit, its setting value are X, Y, F, G, R, K and A.

ERR: Command executies the error output; ERR=0 means without error; ERR=1 means error execution; its setting value are Y, G, R, K and A (Bit address)

Program example:



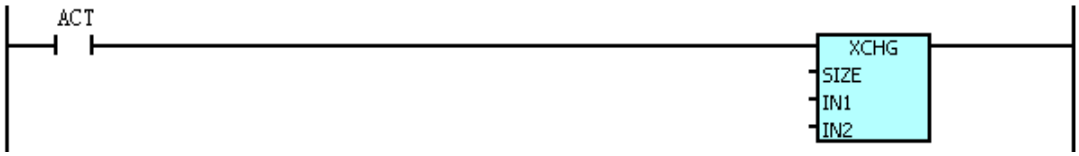

Explanation: It is supposed to D200=3, D303=500, PLC offsets the data value in D200 by the initial address D300 of the resource data when X2.0 sets to 1; the address after offset is regarded as D303. Then, transmit the data from D303 to the destination address D400, that is, D400=D303=500; if the command execution is error, R800.0 sets to 1; the R888.0 is cleared when X0.2 sets to 1.

## 4.29 XCHG (Binary Data Exchange)

- Command function

  Exchange the data between two addresses

- Ladder diagram format




- Controllable condition

  ACT ＝0：The value of the IN1 and IN2 invariable.

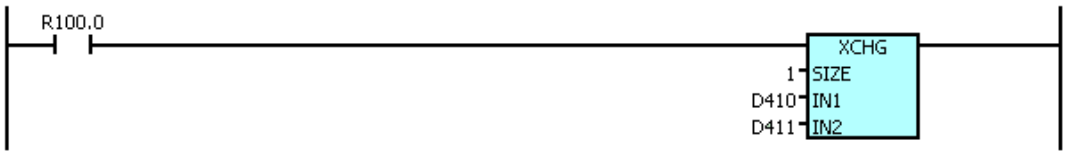
  ＝1：Exchange the value of the IN1 and IN2.

- Relative parameter

  IN1: The data address1 to b exchanged, for example: D410, it means that D is the No.410 parameter. Its addresses are R, Y, G, K, D, T, C, DC and DT.

  IN2: The data address 2 to b exchanged, for example: D411, it means that D is the No.411 parameter. Its addresses are R, Y, G, K, D, T, C, DC and DT.

- Program example:



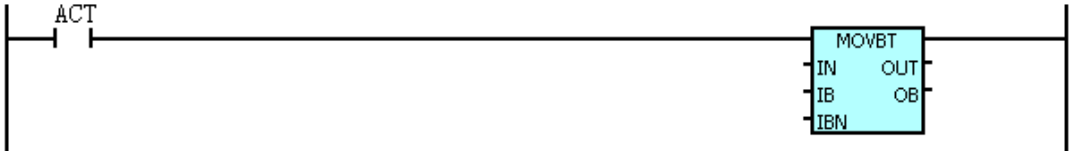

Explanation: When R100.0=0, the data in the D410 and D411 are invariable; when R100.0=1, the data in the D410 and D411 are exchanged; that is, write the data from D410 to the D411; write the data from D411 to D410.

## 4.30 MOVBT (Binary Data Bit Transmission)

- Command function

The specified number of the serial bit data in the transmission bit number; it delivers within the destination address from resource bit position.

- Ladder diagram format




- Controllable condition

ACT ＝0: The value of OUT is invariable.

＝1: Consecutively transmit the multi-bit to the specified position of the OUT address at the IN specified position.

- Relative parameter

IN: Transmit the resource address, for example, X1, it means that the transmission resource address is the No.1 parameter of the X parameter. The addresses can be divided into: X, Y, G, F, R, K and A.

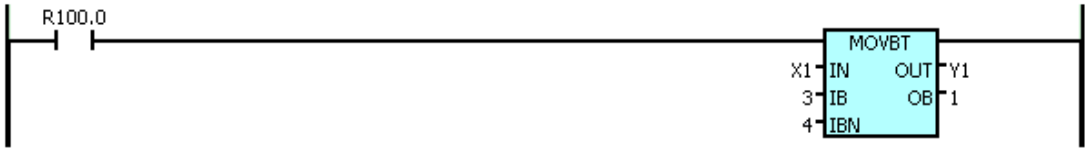
IB: Transmission resource bit, for example, the 3, which means that it transmits from the 3rd bit of the transmission resource address. Its parameter range: 0~7.

IBN: Transmission bit number, for example, the 4, which means that it transmits 4-bit digit from transmission resource bit. Its parameter range: 0~256.

OUT: Transmit the destination address, for example Y1, it means that the transmission destination address is the No.1 parameter of the Y parameter. Its addresses can be divided into: Y, G, R, K and A.

OB: Transmit the destination bit position, for example, the 1, it means that the transmission position starts from the 1st bit from the destination address. Its parameter range: 0~7.

- Program example:



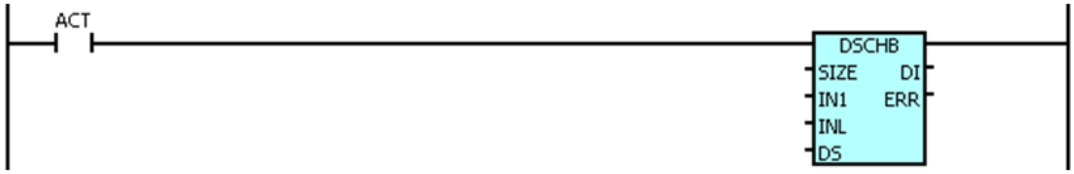

Explanation: When R100.0=0, the state of Y1 is invariable; when R100.0=1, the 4th digit of the 3rd bit of the X1 transmits to the start position at the 1st digit of Y1, that is, Y1.1=X1.3, Y1.2=X1.4, Y1.3=X1.5 and Y1.4=X1.6.

## 4.31 DSCHB (Binary Data Index)

- Command function

DSCHB is indexed the specified data value whether is existed inside the data table. If does, output the first several locates at the data table; if does not, output the information without data.

- Ladder diagram format




Controllable condition:

ACT＝0: The value of DI is invariable.

ACT＝1: Output the position value within the index data table.

- Relative parameter:

IN1: Initial address of data table, for example, the D410, it means the initial address of the data table is the No.410 of the D parameter. Its addresses can be divided into: R, X, Y, F, G, K, A, D, T, C, DC and DT.

INL: Data number within the data, for example, the 20; it means the data table from the initial address D410 to D429. It addresses can be dividied into: R, X, Y, F, G, K, A, D, T, C, DC, DT or constant.

DS: The data to be indexed. Its addresses can be divided into: R, X, Y, F, G, K, A, D, T, C, DC, DT or constant.

DI: Index result outputs an address, the output result is the position of the index data within the data table. Its addresses can be divided into: R, X, Y, F, G, K, A, D, T, C, DC and DT.

ERR: Calculation error output address (bit address), its addresses can be divided into R, Y, G, K and A.

- Program example:


Explanation: When R100.0=1, index whether the same data within the D408 are existed inside the D410-D429, for example: when the data D408 in the D420 is identical, the D407 output data is 10. If this range does not include the same data within the D408, output ERR, R100.1=1, D407 data resets to 0.

# II Programming Guide

# CHAPTER ONE AXIS CONTROL SIGNAL

## 1.1 Axes Moving Status

CNC can send the current axis moving status to PLC, and then PLC works according to the status.

**Axes Moving Signals MV1～MV5（Fn102＃0～Fn102＃4）**

- Signal type: NC→PLC
- Signal function: MV1, MV2, MV3, MV4, MV5 are moving signals for axis 1, 2, 3, 4 ,5 respectively. When an axis is moving, NC sets corresponding axis moving signal to 1. When an axis stops moving, the axis moving signal is 0. PLC works according to the received signal.
- Signal addresses:

| | #7 | #6 | #5 | #4 | #3 | #2 | #1 | #0 |
|---|---|---|---|---|---|---|---|---|
| Fn102 | | | | MV5 | MV4 | MV3 | MV2 | MV1 |

**Axis Moving Direction Signal MVD1～MVD5（Fn106＃0～Fn106＃4）**

- Signal type: NC→PLC
- Signal function: MVD1, MVD2, MVD3, MVD4, MVD5 are axis moving direction signal for axes 1, 2, 3, 4, 5 respectively. When an axis is moving backward, NC sets the axis moving direction signal to 1; when an axis is moving forward, the axis moving direction signal is 0; if an axis stops moving, and the signal will be 1 or 0 according to the moving status before the axis stops.
- Signal addresses:

| | #7 | #6 | #5 | #4 | #3 | #2 | #1 | #0 |
|---|---|---|---|---|---|---|---|---|
| Fn106 | | | | MVD5 | MVD4 | MVD3 | MVD2 | MVD1 |

**Axis enabled signal ABLE1～ABLE5（Fn127＃0～Fn127＃4）**

- Signal type:CNC－>PLC
- Signal function: ABLE1, ABLE 2, ABLE 3, ABLE 4 and ABLE 5 are separately corresponding to 1, 2, 3, 4 and 5, which are respectively corresponding the enables signal of the 1, 2, 3, 4 and 5 axis. When the axis setting is enabled, its signal value is regarded as “1”. When the axis is set to disabled, its signal value is “0”.
- Signal address:

| | #7 | #6 | #5 | #4 | #3 | #2 | #1 | #0 |
|---|---|---|---|---|---|---|---|---|
| Fn127 | | | | ABLE 5 | ABLE 4 | ABLE 3 | ABLE 2 | ABLE 1 |

**Interlocking signal of each axis *IT0～*IT4<G130>**

- Signal type:PLC－>CNC

- Signal function: Forbit moving the the specified axis, each controllable axis has a separated interlocking signal. When any signal of G130#0～G130#4 treats as 0, the interlocking signal of the corresponding axis can be enabled accordingly.

  ITX Interlocking signal of each axis

  0：Enabled

  1：Disabled

| | |
|---|---|
| G130.0 | The 1st axis interlocking signal |
| G130.1 | The 2nd axis interlocking signal |
| G130.2 | The 3rd axis interlocking signal |
| G130.3 | The 4th axis interlocking signal |
| G130.4 | The 5th axis interlocking signal |

- Explanation

1. The interlocking axis movement is forbidden during manual operation; however, the other axes can be moved. If the axes are interlocked during movement, the movement is then stopped after the deceleration of the movement. The axis movement can be performed again after the interlocking is cleared.
2. The overall axes movement should be forbidden when the axis of the command movement is interlocked in the Auto operation. If the axis is interlocked during movement, the overall axes will be stopped after deceleration, this signal is also enabled in the dry run state.



## 1.2 Servo Ready Signal

**SA（Fn000＃6）**

- Signal type:NC→PLC
- Signal function: When NC issues an alarm by receiving an alarm signal from the servo system, it sets SA signal to 0, and informs the PLC that the servo system is not ready and the axis cannot move. When the warning is cancelled, NC sets SA to 1 to move axis again.
- Signal addresses:

| | #7 | #6 | #5 | #4 | #3 | #2 | #1 | #0 |
|---|---|---|---|---|---|---|---|---|
| Fn000 | | SA | | | | | | |

# CHAPTER TWO OPERATION PREPARATION

## 2.1 ESP

**ESP signal ESP(Gn008＃4):**

- Signal type: PLC→NC; valid when it is 0.
- Signal function: When Gn008＃4 is 0 level, NC detects the signal and issues an ESP alarm.
- Signal address:

| | #7 | #6 | #5 | #4 | #3 | #2 | #1 | #0 |
|---|---|---|---|---|---|---|---|---|
| Gn008 | | | | ESP | | | | |

## 2.2 CNC Ready Signal

**MA (Fn001＃7):**

- Signal type: NC→PLC
- Signal function: This signal indicates the CNC is ready for operation.Output condition: After power-on, this signal is set to 1 (usually in a few seconds); if an alarm occurs in CNC or emergency stop is executed, this signal changes to 0.
- Signal address:

| | #7 | #6 | #5 | #4 | #3 | #2 | #1 | #0 |
|---|---|---|---|---|---|---|---|---|
| Fn001 | MA | | | | | | | |

## 2.3 Alarm Signal

**AL(Fn001＃0):**

- Signal type: NC→PLC
- Signal function: When CNC issues an alarm, the alarm will be displayed on the screen and AL is set to 1; after PLC receives this signal, there are three kinds of alarms to be displayed according to the alarm signal status: servo alarm, P/S alarm, overtravel alarm. When CNC is reset, the alarm is cleared and AL is set to 0.
- Signal address:

| | #7 | #6 | #5 | #4 | #3 | #2 | #1 | #0 |
|---|---|---|---|---|---|---|---|---|
| Fn001 | | | | | | | | AL |

## 2.4 Mode Selection

Mode selection signals include MD1, MD2, MD4, DNC1 and ZRN. Six working modes are available: EDIT mode, AUTO mode, MDI mode; JOG mode, HANDLE/INC mode and REF mode. CNC detects signals by outputting the working mode and informs PLC the current working mode.

### 2.4.1 Mode Selection Signal

**MD1,MD2,MD4(G43.0～G43.2),DNC1(G43.5) ZRN(G43.7):**

- Signal type: PLC→NC
- Signal function: The code signal of working mode is shown as follows:

| No. | Code Signal / Working Mode | ZRN | DNC1 | MD4 | MD2 | MD1 |
|---|---|---|---|---|---|---|
| 1 | EDIT mode | 0 | 0 | 0 | 1 | 1 |
| 2 | AUTO mode | 0 | 0 | 0 | 0 | 1 |
| 3 | MDI mode | 0 | 0 | 0 | 0 | 0 |
| 4 | HANDLE/INC mode | 0 | 0 | 1 | 0 | 0 |
| 5 | JOG mode | 0 | 0 | 1 | 0 | 1 |
| 6 | Machine reference point return (REF) mode | 1 | 0 | 1 | 0 | 1 |

PLC assigns values to the code signals after receiving the input signal of working mode, and then, sends it to NC. NC determines the CNC working mode according to the code signal.

- Signal address:

| | #7 | #6 | #5 | #4 | #3 | #2 | #1 | #0 |
|---|---|---|---|---|---|---|---|---|
| Gn043 | ZRN | | DNCI | | | MD4 | MD2 | MD1 |

**Note: DNCI signal is only suitable for the Milling Machien System.**

### 2.4.2 Working Mode Check Signal

**MINC(Fn003＃0), MH(Fn003＃1), MJ(Fn003＃2), MMDI(Fn003＃3), MMEM(Fn003＃5), MEDT(Fn003＃6), MREF(Fn004＃5), MPST(Fn004＃6)**

- Signal type: NC→PLC
- Signal function: When CNC is in a certain working mode, the corresponding F signal is set to 1 and then sent to PLC; PLC works according to the working mode check signal.

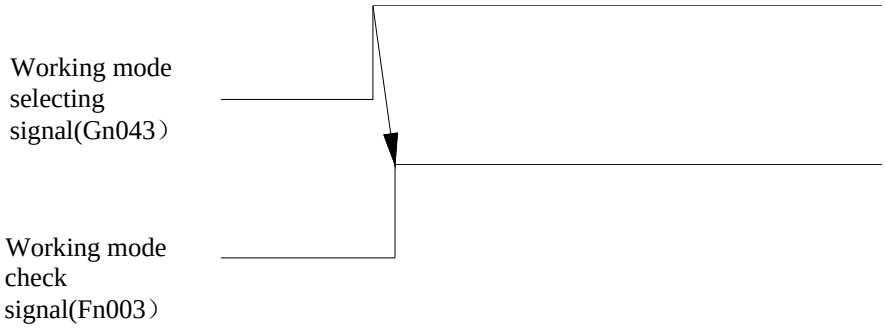
| | |
|---|---|
| INC mode check signal | MINC |
| MPG mode check signal | MH |
| JOG mode check signal | MJ |
| MDI mode check signal | MMDI |
| DNC mode check signal | MRMT |
| AUTO mode check signal | MMEM |
| EDIT mode check signal | MEDT |
| REF mode check signal | MREF |
| Program reference Point return mode check signal | MPRO |

- Signal addresses:

| | #7 | #6 | #5 | #4 | #3 | #2 | #1 | #0 |
|---|---|---|---|---|---|---|---|---|
| Fn003 | | MEDT | MMEM | | MMDI | MJ | MH | MINC |
| Fn004 | | MPRO | MREF | | | | | |



## 2.4.3 Working Mode Signal Sequence




## 2.5 Status Output

**Rapid traverse signal RPDO (Fn002＃1):**

- Signal type: NC→PLC
- Signal function: When CNC is in MANUAL rapid traverse mode, axis movement is executed and RPDO is set to 1.
- Note: RPDO is 1 during rapid traverse and the status remains the same when the feed stops.

  When a mode other than rapid traverse is selected, RPDO signal is reset to 0 after the axis moves.
- Signal address:

| | #7 | #6 | #5 | #4 | #3 | #2 | #1 | #0 |
|---|---|---|---|---|---|---|---|---|
| Fn002 | | | | | | | PRDO | |

**Cutting feed signal CUT (F002#6):**

- Signal type:CNC－>PLC
- Signal function: Only when the CNC is performed the cutting feed (including the linear interpolation, arc interpolation, helical linear interpolation, thread cutting, skip cutting or the cutting in the canned cycle), CNC is set the F2.6 to “1”; the F2.6 becomes “0” in the other conditions.
- Signal address:

| | #7 | #6 | #5 | #4 | #3 | #2 | #1 | #0 |
|---|---|---|---|---|---|---|---|---|
| Fn002 | | CUT | | | | | | |

**Thread cutting signal THRD (Fn002＃3)**

- Signal type: NC→PLC
- Signal function: It indicates that the thread cutting is in-processing.

  This signal is 1 in the following conditions:

  1. In thread cutting mode.

  2. Thread cutting is in-processing.

  This signal is 0 when neither condition 1 nor 2 is fulfilled.
- Signal address:

| | #7 | #6 | #5 | #4 | #3 | #2 | #1 | #0 |
|---|---|---|---|---|---|---|---|---|
| Fn002 | | | | | THRD | | | |

**Tapping signal TAP（Fn001#5）：**

- Signal type:CNC－>PLC
- Signal function: When CNC is on the tapping state, the NC is set the Fn001.5 to “1”. The CNC will set the Fn001.5 to “0” when the system is on the non-tapping mode, as well as input the ESP signal or CNC resetting occurs.
- Signal address:

| | #7 | #6 | #5 | #4 | #3 | #2 | #1 | #0 |
|---|---|---|---|---|---|---|---|---|
| Fn001 | | TAP | | | | | | |

**G04 execution signal CDWL（Fn212#2）：**

- Signal type:CNC－>PLC
- Signal function: When CNC is on the G04 dwell state, the CNC is set the CDWL to “1”; after the G04 dwell is ended, CNC is set the CDWL to “0”.

| | #7 | #6 | #5 | #4 | #3 | #2 | #1 | #0 |
|---|---|---|---|---|---|---|---|---|
| Fn212 | | | | | | CDWL | | |

## 2.6 Overtravel Detection

**Overtravel Signal +L1 ~ +L5 (G114#0~G114#4) , -L1 ~ -L5(G116#0~G116#4)**

- Signal type: PLC→NC
- Signal function: It indicates that the control axis has reached the stroke limit; every direction of each axis has such signal; the symbol "+" or " - " represents the direction, and the numbers correspond to the control axis.
- [ Motion ] When the above signals are 0, the motion of control unit is as follows:

  1)*In AUTO mode, even only one axis overtravel signal changes to 0, all the axes will decelerate and stop, and an alarm is issued. The running is stopped.

  2)*In MANUAL mode, only the corresponding axis stops, and the axis can move reversely after stop.

  3)*The moving direction is stored as long as the overtravel signal becomes 0, and the direction is invalid before the alarm is cleared even if the signal turns into 1.

- Signal addresses:

| | #7 | #6 | #5 | #4 | #3 | #2 | #1 | #0 |
|---|---|---|---|---|---|---|---|---|
| Gn114 | | | | +L5 | +L4 | +L3 | +L2 | +L1 |
| Gn116 | | | | -L5 | -L4 | -L3 | -L2 | -L1 |

# CHAPTER THREE MANUAL OPERATION

Manual feed: In MANUAL mode, when the feed axis signal and direction selection signal on the operation panel are set to 1, the tool continuously moves on the selected axis along the selected direction.

Incremental feed: In incremental mode, when the feed axis signal and direction selection signal on the operation panel are set to 1, the tool moves one step on the selected axis along the selected direction. The minimum movement distance is the least input increment. Four override values (0.001, 0.010, 0.100,1.000) are available.

The only difference between manual feed and incremental feed is the way of selecting feed distance. In manual feed, when the feed axis signals and direction selection signals( + J1, - J1, + J2, - J2, + J3, - J3) are 1, the tool can feed continuously. In incremental feed, the tool feeds in steps.

When manually turn ON the rapid traverse, the tool feeds at the rapid traverse speed. In incremental feed mode, the step distance can be selected by MP1, MP2.



## 3.1.1 Feed Axis signal and Direction Selection Signal

**+ J1 ~ + J5 (G100.0 ~ G100.4), - J1 ~ - J5 (G102.0 ~ G102.4)**

- Signal type: PLC→NC
- Signal function: It selects the feed axis and direction in manual feed mode or incremental feed mode. NC sets the corresponding axis and direction selection signal to 1, and PLC proceeds the control after receiving the signal. Symbol “+” or “-“ indicates the feed direction. The number corresponds to the control axis.
- Note:

  1): In manual feed, CNC makes the selected axis moving continuously. In the incremental feed mode, CNC makes the selected axis moving according to the specified rate defined by MP1, MP2 signals.

  2): When an axis is moving, NC sets the axis and direction selection signal to 1. When the axis stops moving, the signal changes to 0.

- Signal addresses:

| | #7 | #6 | #5 | #4 | #3 | #2 | #1 | #0 |
|---|---|---|---|---|---|---|---|---|
| Gn100 | | | | +J5 | +J4 | +J3 | +J2 | +J1 |
| Gn102 | | | | -J5 | -J4 | -J3 | -J2 | -J1 |

## 3.1.2 Manual Feed Override Signal

**JV00 ~ JV15(G10, G11):**

- Signal type: PLC→NC
- Signal function: It selects the manual feed override. The following table shows the relationship between signals and manual feed override. PLC sets value to G10, G11 and transmits it to NC after receiving the external override input signal. CNC displays the corresponding feedrate.

| Gn011 | Gn010 | Override (%) |
|---|---|---|
| 0000 0000 | 0000 0000 | 150 |
| 0000 0000 | 0000 0001 | 140 |
| 0000 0000 | 0000 0010 | 130 |
| 0000 0000 | 0000 0011 | 120 |
| 0000 0000 | 0000 0100 | 110 |
| 0000 0000 | 0000 0101 | 100 |
| 0000 0000 | 0000 0110 | 90 |
| 0000 0000 | 0000 0111 | 80 |
| 0000 0000 | 0000 1000 | 70 |
| 0000 0000 | 0000 1001 | 60 |
| 0000 0000 | 0000 1010 | 50 |
| 0000 0000 | 0000 1011 | 40 |
| 0000 0000 | 0000 1100 | 30 |
| 0000 0000 | 0000 1101 | 20 |
| 0000 0000 | 0000 1110 | 10 |
| 0000 0000 | 0000 1111 | 0 |

- Signal addresses:

| | #7 | #6 | #5 | #4 | #3 | #2 | #1 | #0 |
|---|---|---|---|---|---|---|---|---|
| Gn010 | JV7 | JV6 | JV5 | JV4 | JV3 | JV2 | JV1 | JV0 |
| Gn011 | JV15 | JV14 | JV13 | JV12 | JV11 | JV10 | JV9 | JV8 |

## 3.1.3 Rapid Traverse Selection Signal

**RT (Gn019＃7)**

- Signal type: PLC→NC
- Signal function: It selects the rapid traverse speed in manual mode. PLC sets RT to 1 after receiving the manual rapid traverse input signal, and transmits it to NC.During manual rapid

traverse, when RT is switched from 1 to 0 or from 0 to 1,the feedrate reduces to 0, and then increases to the specified value. During acceleration/deceleration, the status of feed axis signal and direction selection signal remain unchanged.

- Signal addresses:

| | #7 | #6 | #5 | #4 | #3 | #2 | #1 | #0 |
|---|---|---|---|---|---|---|---|---|
| G19 | RT | | | | | | | |

## 3.2 MPG Feed

In MPG feed mode, the tool can be minutely moved by rotating the handle. An axis can be selected according to the MPG axis selection signal.

### 3.2.1 MPG Feed Axis Selection Signal

**The 1st MPG H1A～H1D (Gn018＃0～Gn018＃3)**

**The 2nd MPG H2A～H2D（Gn018＃4～Gn018＃7）**

- Signal type: PLC→NC
- Signal function: It selects the MPG feed axis. PLC assigns values to HA～HD after receiving the MPG feed axis input signal. NC selects the corresponding axis according to the HA～HD signal status. The relationship between these signals and MPG feed axis is shown as follows:

| HD | HC | HB | HA | Feed Axis |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | None |
| 0 | 0 | 0 | 1 | Axis1 |
| 0 | 0 | 1 | 0 | Axis 2 |
| 0 | 0 | 1 | 1 | Axis 3 |
| 0 | 1 | 0 | 0 | Axis 4 |
| 0 | 1 | 0 | 1 | Axis 5 |
| 0 | 1 | 1 | 0 | Axis 6 |

- Signal address:

| | #7 | #6 | #5 | #4 | #3 | #2 | #1 | #0 |
|---|---|---|---|---|---|---|---|---|
| Gn018 | H2D | H2C | H2B | H2A | H1D | H1C | H1B | H1A |

### 3.2.2 MPG Override Signal

**The 1st MP1(Gn019＃4), MP2(Gn019＃5)**

**The 2nd MP21(Gn087＃0), MP22(Gn087＃1)**

- Signal type: PLC→NC
- Signal function: During the MPG feed, it selects the magnification of the manual MPG feedrate

corresponding to each pulse which is generated by the MPG. PLC assigns values to MP1, MP2 after receiving the MPG override (×1, ×10, ×100, ×1000) input signal, and transmits it to NC. The relationship between MP1, MP2 and the MPG override is shown in the following table:

| MP2 | MP1 | Override | Movement amount |
|---|---|---|---|
| 0 | 0 | 1 | Least input increment ×1 |
| 0 | 1 | 10 | Least input increment ×10 |
| 1 | 0 | 100 | Least input increment ×100 |
| 1 | 1 | 1000 | Least input increment × 1000 |

- Signal addresses:

| | #7 | #6 | #5 | #4 | #3 | #2 | #1 | #0 |
|---|---|---|---|---|---|---|---|---|
| **Gn019** | | | **MP2** | **MP1** | | | | |
| **Gn087** | | | | | | | **MP22** | **MP21** |



## 3.3 MPG Trial Cutting

User can use the MPG function in the Auto operation by the MPG trial function to control the execution velocity of the program, that is, it can be simply and conveniently inspected the error of the program.

### 3.3.1 Inspection Method Signal

**MMOD（G067＃2）**

- Signal type:PLC－>CNC
- Signal function: Shift the signal of MPG trial cutting method. If this signal is “1”, it becomes MPG trial cutting method during automatic operation in the Auto mode. It will not become the MPG trial cutting method in the MDI mode. In addition, this signal sets to “1” in the automatic operation in the Auto mode, and then it turns into the MPG trial cutting method from the start of the next block to be executed. This signal will be set to “0” in the the Auto operation, the MPG trial cutting method becomes disabled from the start of the nect block to be performed.
- Signal address:

| | #7 | #6 | #5 | #4 | #3 | #2 | #1 | #0 |
|---|---|---|---|---|---|---|---|---|
| **Gn067** | | | | | | **MMOD** | | |

### 3.3.2 Signal in Inspection Method

**MMMOD（Fn091＃3）**

- Signal type:CNC－> PLC
- Signal function: Notice the fact in the MPG trial cutting method. This signal is regarded as “1” when entering the MPG trial cutting method; and it is treated as “0” when retreating from the MPG trial cutting method.
- Signal address:

| | #7 | #6 | #5 | #4 | #3 | #2 | #1 | #0 |
|---|---|---|---|---|---|---|---|---|
| Fn091 | | | | | MMMOD | | | |

## 3.4 MPG Interruption

In the Auto operation mode (Auto mode or MDI mode) and the Edit mode, the movement value generated by MPG operation can be overlapped to the movement of the automatic operation based upon the MPG. The axis by the MPG interruption can be selected the signal. The movement distance of the MPG interruption is determined by rotation value and the federate of the MPG.

**MPG interruption selection signal**

**The 1st MPG HS1IA～HS1ID (Gn041＃0～Gn041＃3)**

**The 2nd MPG HS2IA～HS2ID (Gn041＃4～Gn041＃7)**

- Signal type: Signal input
- Signal function: Select the interruption axis in the MPG interruption

| HSnID | HSnIC | HSnIB | HSnIA | Feed axis |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | Without |
| 0 | 0 | 0 | 1 | The 1st axis |
| 0 | 0 | 1 | 0 | The 2nd axis |
| 0 | 0 | 1 | 1 | The 3rd axis |
| 0 | 1 | 0 | 0 | The 4th axis |
| 0 | 1 | 0 | 1 | The 5th axis |
| 0 | 1 | 1 | 0 | The 6th axis |

- Signal address:

| | #7 | #6 | #5 | #4 | #3 | #2 | #1 | #0 |
|---|---|---|---|---|---|---|---|---|
| Gn041 | HS2ID | HS2IC | HS2IB | HS2IA | HS1ID | HS1IC | HS1IB | HS1IA |

# CHAPTER FOUR MECHANICAL ZERO RETURN/PROGRAM ZERO ZERO

## 4.1 Mechanical Zero Return

In the mechanical zero return, generally, the feed axis and direction selection signal are set to “1”, so that the machine tool can be moved along with the specified direction, and then return to the mechanical zero. After the end of the mechanical zero return is performed, the CNC will set the workpiece coordinate system based upon the value set by data parameter.

### 4.1.1 Machine Reference Point Return Completion Signal

**The 1st mechanical zero point ZP1～ZP5(Fn094＃0～Fn094＃4)**

**The 2nd mechanical zero point ZP21～ZP25(Fn096＃0～Fn096＃4)**

**The 3rd mechanical zero point ZP31～ZP35(Fn098＃0～Fn098＃4)**

**The 4th mechanical zero point ZP41～ZP45(Fn100＃0～Fn100＃4)**

- Signal type: CNC－>PLC
- Signal function: When an axis ends the machine reference point return, NC sets the corresponding F signal to 1, and transmits it to PLC. PLC proceeds logic control accordingly. ZPn1, ZPn2, ZPn3, ZPn4 and ZPn5 signals are machine reference point return completion signals for axes 1, 2, 3, 4, 5 respectively.
- Notes: When machine reference point return has already completed (and G28 command returns to zero), and the current position is within the specified area, the machine reference point return completion signal changes to 1. When the tool moves away from machine reference point or when an emergency or drive unit alarm is issued, the machine reference point return completion signal changes to 0.
- Signal addresses:

| | #7 | #6 | #5 | #4 | #3 | #2 | #1 | #0 |
|---|---|---|---|---|---|---|---|---|
| Fn094 | | | | ZP5 | ZP4 | ZP3 | ZP2 | ZP1 |
| Fn096 | | | | ZP25 | ZP24 | ZP23 | ZP22 | ZP21 |
| Fn098 | | | | ZP35 | ZP34 | ZP33 | ZP32 | ZP31 |
| Fn100 | | | | ZP45 | ZP44 | ZP43 | ZP42 | ZP41 |

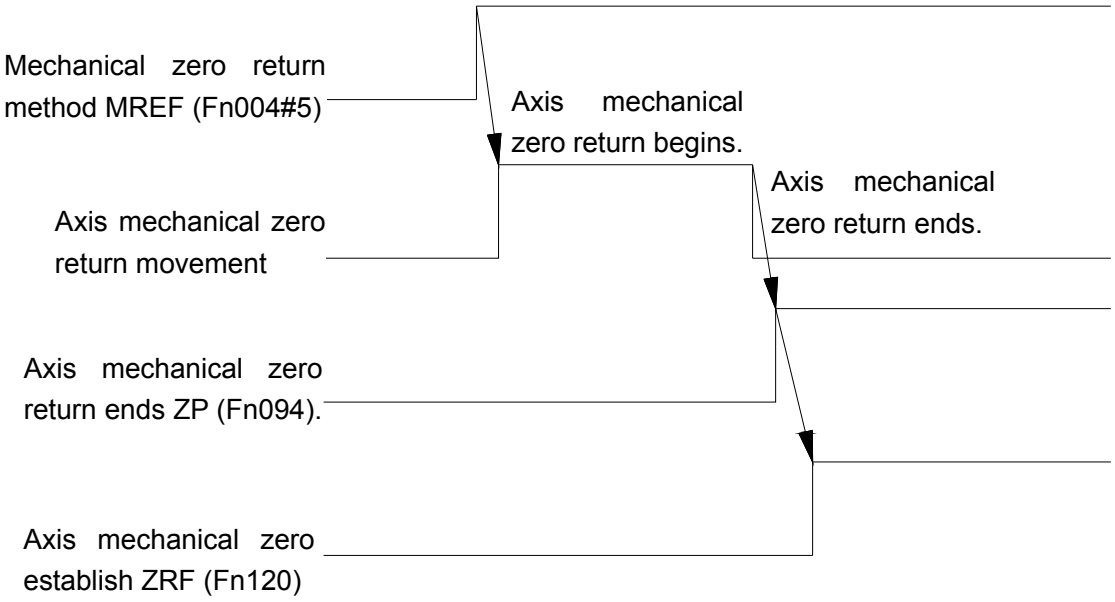
### 4.1.2 Machine Reference Point Setting Signal

**ZRF1～ZRF5(Fn120＃0～Fn120＃4)**

- Signal type: CNC－>PLC
- Signal function: After the machine reference point return is executed and the machine reference point is set, the corresponding machine reference point setting signal is set to 1, and then this signal is transmitted to PLC. PLC proceeds logic control according to the status of the reference point setting signal. ZRF1, ZRF2, ZRF3, ZRF4, ZRF5 are the signals corresponding to axes 1, 2, 3, 4, 5.
- Signal addresses:

| | #7 | #6 | #5 | #4 | #3 | #2 | #1 | #0 |
|---|---|---|---|---|---|---|---|---|
| Fn120 | | | | ZRF5 | ZRF4 | ZRF3 | ZRF2 | ZRF1 |

### 4.1.3 Sequence of Machine Reference Point Signal




## 4.2 Program Reference Point Return

**Program Reference Point Return End Signal PRO1～PRO5（Fn198#0～Fn198#4）**

- Signal type:CNC－>PLC
- Signal function: When program reference point return ends, the tool will stay at the program reference point point. NC will set the program reference point return end signal to 1, and transmit to PLC, which informs PLC that the program reference point return is ended. PRO1, PRO2, PRO3, PRO4 are program reference point return end signals for Axis 1, 2, 3, 4 respectively.

- Signal address:

| | #7 | #6 | #5 | #4 | #3 | #2 | #1 | #0 |
|---|---|---|---|---|---|---|---|---|
| Fn198 | | | | PRO5 | PRO4 | PRO3 | PRO2 | PRO1 |

# CHAPTER FIVE AUTOMATIC OPERATION

## 5.1 Cycle Start/Feed Hold

- Cycle start (start automatic operation)

1) In AUTO or MDI mode, when the automatic operation start signal ST is enabled, program starts running.

Signal ST is ignored under the following conditions:

A: The system is in a mode other than AUTO or MDI mode.

B: Feed hold signal SP is 0.

C: Emergency stop signal ESP is 0. D:

External reset signal ERS is 1.

E: The “RESET” key on the operation panel is pressed.

F: CNC is in alarm state.

G: Automatic operation is already started.

H: Program reset signal SRN is 1.

I : CNC is searching for a sequence number.

2) During automatic operation, the CNC performs feed hold under the following conditions:

A: The feed hold signal SP is 0.

B: The AUTO mode is switched to other modes (MANUAL, HANDWHEEL, STEP, MACHINE REFERENCE RETURN, PROGRAM REFERENCE RETURN modes)

C: During single block execution, the instruction for the single block is finished.

D: The operation is finished in MDI mode.

E: An alarm occurs in CNC system.

F: After the AUTO mode is switched to EDIT mode, the single block instruction is finished.

3) During automatic operation, CNC enters into reset state and stops running under the following conditions:

A: Emergency stop signal ESP is set to 0.

B: External reset signal ERS is 1.

C: “RESET” key on the operation panel is pressed.

- Feed hold (automatic operation interrupted)

During automatic operation, when the feed hold signal SP is 0, CNC enters into paused state and stops running. Meanwhile, cycle start indicator STL is set to 0, and feed hold signal SPL is set to 1. Even if SP is set to 1 again, the automatic operation will not be re-started. Only when SP is set to 1 and ST signal is valid, can the machine be restarted and operated automatically.

During the execution of blocks containing M, S, T function instructions, SP signal is set to 0, STL will be 0 immediately, SPL signal is 1, and CNC enters into paused state. When receiving FIN signal from PLC, CNC continues executing the interrupted blocks. After the block is executed, SPL signal is set to 0 (STL signal is 0), CNC enters into automatic operation paused state.

### 5.1.1 Cycle Start Signal

**ST（Gn007＃2）**

- Signal type: PLC→NC; falling edge is valid
- Signal function: In AUTO or MDI mode, when PLC receives the input signal of cycle start on the panel, G7.2 is set to 1 at first, then set to 0 and transmitted to NC, the automatic operation is started.
- Signal address:

| | #7 | #6 | #5 | #4 | #3 | #2 | #1 | #0 |
|---|---|---|---|---|---|---|---|---|
| Gn007 | | | | | | ST | | |

### 5.1.2 Feed Hold Signal

**SP（Gn008＃5）**

- Signal type: PLC→NC, valid when it is 0.
- Signal function: PLC sets G8.5 to 0 after receiving this signal, and transmits it to NC. The automatic operation is stopped. When the SP input signal is 0, the automatic operation cannot be started.
- Signal addresses:

| | #7 | #6 | #5 | #4 | #3 | #2 | #1 | #0 |
|---|---|---|---|---|---|---|---|---|
| Gn008 | | | SP | | | | | |

### 5.1.3 Cycle Start Signal

**STL（Fn000＃5）**

- Signal type: NC→PLC
- Signal function: CNC sets STL to 1 during automatic operation and then transmits it to PLC; PLC proceeds logic control according to the state of STL.
- Signal address

| | #7 | #6 | #5 | #4 | #3 | #2 | #1 | #0 |
|---|---|---|---|---|---|---|---|---|
| Fn000 | | | STL | | | | | |

## 5.1.4 Feed Hold Signal

**SPL（Fn000＃4）**

- Signal type: NC→PLC
- Signal function: CNC sets SPL to 1 when it is in paused state, and then transmits it to PLC. PLC starts logic control according to the state of SPL.
- Signal address:

| | #7 | #6 | #5 | #4 | #3 | #2 | #1 | #0 |
|---|---|---|---|---|---|---|---|---|
| Fn000 | | | | SPL | | | | |

## 5.1.5 Automatic Operation Signal

**OP（Fn000＃7）**

- Signal type: NC→PLC；
- Signal function: CNC sets OP to 1 during automatic operation and then transmits it to PLC. PLC works according to the state of OP.

| | Cycle start indicator STL | Feed hold indicator SPL | Automatic operation indicator OP |
|---|---|---|---|
| Cycle start state | 1 | 0 | 1 |
| Feed hold state | 0 | 1 | 1 |
| Automatic operation paused state | 0 | 0 | 0 |
| Reset state | 0 | 0 | 0 |

- Signal address:

| | #7 | #6 | #5 | #4 | #3 | #2 | #1 | #0 |
|---|---|---|---|---|---|---|---|---|
| Fn000 | OP | | | | | | | |

# 5.2 Reset

Under the following conditions, CNC enters into reset state:

1) Emergency stop signal ESP is set to 0.

2) External reset signal ERS is 1.

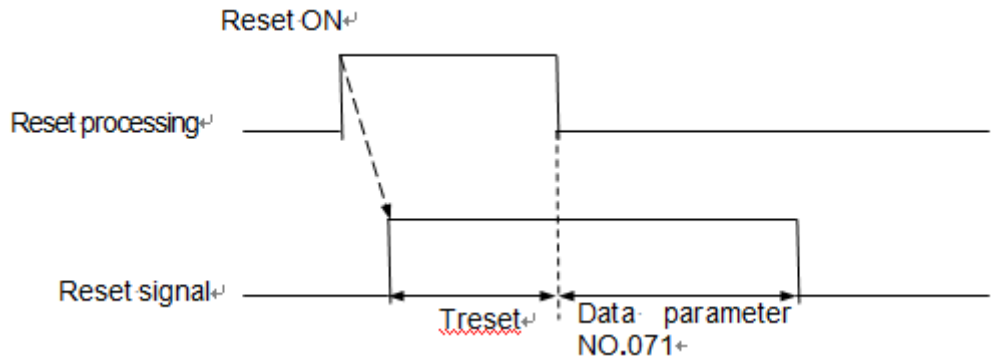
3) “RESET” key on the operation panel is pressed.

Under other conditions, reset signal RST is changed to 0 after the time set by parameter No. 071 has passed.

RST time =T reset (reset processing time) + the setting value of data parameter No.071

During automatic operation, when CNC is reset, the operation stops and control axis decelerates

until it stops. When CNC is reset during the execution the M, S, T instruction, the signal MF, SF, TF are set to 0 within 16ms.




**During the Auto operation, the automatic operation is stopped when CNC is reset; the machine tool decelerates and then stops along with the movement direction of the controllable axis. When the CNC is reset during executing the M, S and T commands, the MF, SF and TF signals within 16ms are set to "0".**



## 5.2.1 External Reset Signal

**ERS（Gn008＃7）**

- Signal type:PLC—>CNC
- Signal function: PLC sets G8.7 to 1 after receiving the external reset input signal, and transmit it to NC. When CNC resets, RST signal is changed to 1.
- Signal address:

| | #7 | #6 | #5 | #4 | #3 | #2 | #1 | #0 |
|---|---|---|---|---|---|---|---|---|
| Gn008 | ERS | | | | | | | |

## 5.2.2 Reset Signal

**RST（Fn001＃1）**

- Signal type:PLC—>CNC
- Signal function: When CNC is in reset state, RST is set to 1. Then the signal is transmitted to PLC, PLC works according to the state of RST.
- Note: RST signal is set to 1 under the following conditions:

  1) The external emergency stop input signal ESP is set to 0.

  2) The external reset signal ERS is 1.

  3) The "RESET" key on the panel is pressed.

In other conditions, the RST is set to 0 when the time set by data parameter No. 071 has passed.

- Signal address:

| | #7 | #6 | #5 | #4 | #3 | #2 | #1 | #0 |
|---|---|---|---|---|---|---|---|---|
| Fn001 | | | | | | | RST | |

## 5.2.3 Resetting & Cursor Return Signal

**RRW（Gn008＃6）**

- Signal type:PLC－>CNC
- Signal function: Reset CNC and return the cursor to the start of the main program.
- Precaution: If you want resetting the CNC and entering the resetting state, set the resetting and cursor return signal RRW as “1”,
- Signal address:

| | #7 | #6 | #5 | #4 | #3 | #2 | #1 | #0 |
|---|---|---|---|---|---|---|---|---|
| Gn008 | | RRW | | | | | | |

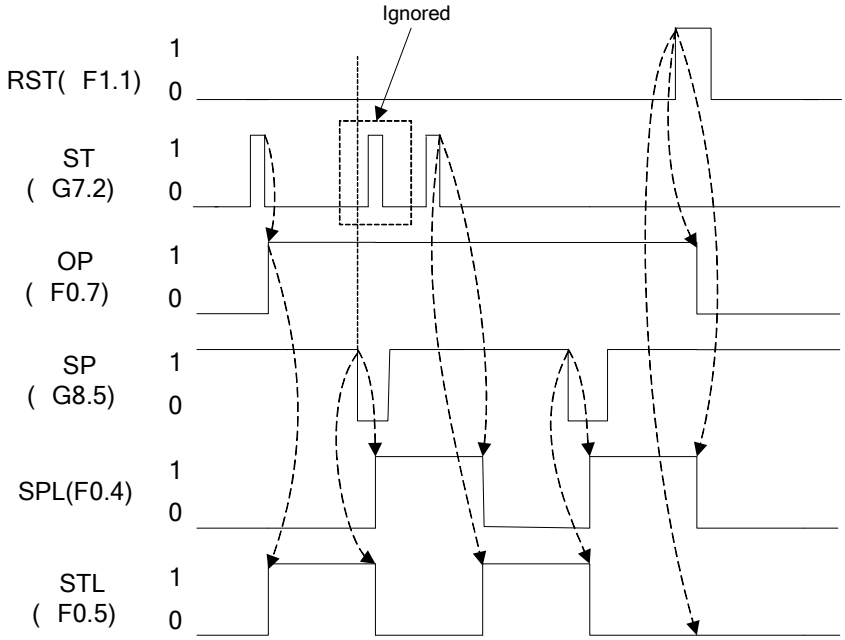
## 5.2.4 Cursor Return Signal

**RWD（Fn000＃0）**

- Signal type:CNC－>PLC
- Signal function: Notice to the PLC, the CNC is already reset and the cursor is then returned to the start of the main program.
- Precaution: When the resetting and cursor return signal RRW is set to “1”, RWD outputs “1” (It turns into “0” after delaying 100ms.)
- Signal address:

| | #7 | #6 | #5 | #4 | #3 | #2 | #1 | #0 |
|---|---|---|---|---|---|---|---|---|
| Fn000 | | | | | | | | RWD |

### 5.2.5 Operation Starting Sequence




## 5.3 Machine Lock

The machine lock can be turned ON when the programs are being checked before machining. The MLK signals (all-axis machine lock signal) are set to 1. During manual or automatic operation, though the execution of inputting pulse to servo motor stops, CNC still assigns instructions; meanwhile the incremental and absolute coordinate systems are updated. The programs can be checked by monitoring the changing of coordinate system.

### 5.3.1 All-Axis Machine Lock Signal

**MIK（Gn044＃1）**

- Signal type: PLC→NC
- Signal function: PLC sets MIK to 1 after receiving all-axis machine lock signal, and then transmits it to NC. All the axes cannot move.
- Note: When MIK is 1, during manual or automatic operation, CNC does not output pulse to servo motor and the worktable does not move.
- Signal address:

| | #7 | #6 | #5 | #4 | #3 | #2 | #1 | #0 |
|---|---|---|---|---|---|---|---|---|
| Gn044 | | | | | | | MLK | |

### 5.3.2 All-Axis Machine Lock Check Signal

**MMLK（Fn004＃1）**

- Signal type: NC→PLC
- Signal function: when all the axes are locked, CNC set all-axis machine lock check signal to 1 and then transmits it to PLC.
- Signal address:

| | #7 | #6 | #5 | #4 | #3 | #2 | #1 | #0 |
|---|---|---|---|---|---|---|---|---|
| **Fn004** | | | | | | | **MMLK** | |

## 5.4 Dry Run

Dry run is valid in automatic operation. The machine works at a constant feedrate rather than the specified feedrate in the program. This function is used to check the machine without workpiece. The dry run speed depends on the manual feed override signal (JV0~JV15).

| | **Program Command** | |
|---|---|---|
| | Rapid traverse | Cutting feed |
| **Rapid traverse button ON** | Rapid traverse | Manual feed top velocity |
| **Rapid traverse button OFF** | Manual feedrate or rapid traverse | Manual feedrate |



### 5.4.1 Dry Run Signal

**DRN（Gn046＃7）：**

- Signal type: PLC→NC
- Signal function: PLC sets DRN to 1 after receiving the dry run input signal, and then, transmits it to NC. CNC enters into dry run state.
- Note:

  A: When DRN is 1, the machine runs at the dry run speed; when it is 0, the machine works normally.

  B: When DRN is changed from 0 to 1 or from 1 to 0 during the operation, the machine speed reduces to 0, and then increases to the specified feedrate.
- Signal address:

| | #7 | #6 | #5 | #4 | #3 | #2 | #1 | #0 |
|---|---|---|---|---|---|---|---|---|
| **Gn046** | **DRN** | | | | | | | |

### 5.4.2 Dry Run Check Signal

**MDRN（Fn002＃7）：**

- Signal type: NC→PLC
- Signal function: CNC sets MDRN to 1 in dry run state and then transmits it to PLC.
- Signal address:

| | #7 | #6 | #5 | #4 | #3 | #2 | #1 | #0 |
|---|---|---|---|---|---|---|---|---|
| Fn002 | MDRN | | | | | | | |

## 5.5 Single Block

Single block execution is valid only during automatic operation, after the single block signal SBK is set to 1, and the current block is executed, the CNC enters into paused state. When the SBK is set to 0 again, the program starts running.



### 5.5.1 Single Block Signal

**SBK（Gn046＃1）：**

- Signal type: PLC→NC
- Signal function: PLC set SBK to 1 after receiving the single block input signal, and transmits it to NC. CNC enters into single block operation state.
- Signal address:

| | #7 | #6 | #5 | #4 | #3 | #2 | #1 | #0 |
|---|---|---|---|---|---|---|---|---|
| Gn046 | | | | | | | SBK | |

### 5.5.2 Single Block Check Signal

**MSBK（Fn004＃3）：**

- Signal type: NC→PLC
- Signal function: CNC sets MSBK to 1 when it is in the single block execution state, and then transmits it to PLC.
- Note:

1): During thread cutting, the SBK signal changes to 1. The operation stops after the first non-thread cutting block is executed.

2): During canned cycle, when the SBK signal is set to 1, the operation stops each time the tool approaches drilling holes or tool retracts (rather than stops at the end of the block).

- Signal address:

| | #7 | #6 | #5 | #4 | #3 | #2 | #1 | #0 |
|---|---|---|---|---|---|---|---|---|
| Fn004 | | | | | MSBK | | | |

## 5.6 Optional Block Skip

During automatic operation, when a slash "/" is specified at the head of a block, and the optional block skip signal BDT is set to 1, the block is skipped during execution.

### 5.6.1 Optional Block Skip Signal

**BDT（Gn044＃0）：**

- Signal type: PLC→NC
- Signal function: PLC sets BDT to 1 after receiving the optional block skip input signal, and then transmits it to NC. CNC enters into the state of optional block skip. In a program, blocks started with "/" will not be executed.
- Signal address:

| | #7 | #6 | #5 | #4 | #3 | #2 | #1 | #0 |
|---|---|---|---|---|---|---|---|---|
| Gn044 | | | | | | | | BDT |

### 5.6.2 Optional Block Skip Check Signal

**MBDT（Fn004＃0）：**

- Signal type: NC→PLC
- Signal function: CNC sets MBDT to 1 during optional block skip execution, and then transmits it to PLC. PLC works according to the state of MBDT.
- Signal address:

| | #7 | #6 | #5 | #4 | #3 | #2 | #1 | #0 |
|---|---|---|---|---|---|---|---|---|
| Fn004 | | | | | | | | MBDT |

# CHAPTER SIX FEEDRATE CONTROL

## 6.1 Rapid Traverse Signal

The rapid traverse speed of all axes can be set by parameter rather than set when programming. It also can be adjusted by controlling the rapid traverse override.

**RPD0（Fn002＃1）:**

- Signal type: NC→PLC
- Signal function: When CNC executes the movement command at the rapid traverse speed, it sets RPDO to 1 and then transmits it to PLC.
- Note:

  1): When RPDO is 1, it means after rapid traverse is selected, an axis starts moving; when RPDO is 0, it means after non-rapid traverse speed is selected, an axis starts moving.

  2): The rapid traverse in automatic operation includes canned cycle positioning, machine reference point return etc. The manual rapid traverse also includes machine reference point return.

  3): Once the rapid traverse is selected, the signal is always 1 (even when the operation stops), till other feedrate is selected.
- Signal address:

| | #7 | #6 | #5 | #4 | #3 | #2 | #1 | #0 |
|---|---|---|---|---|---|---|---|---|
| Fn002 | | | | | | | RPD0 | |

## 6.2 Rapid Traverse Override

An override of four steps (F0, 25%, 50%, 100%) can be applied to the rapid traverse rate. F0 is set by a parameter.In AUTO mode or MANUAL mode, the actual feedrate is the product of override value and the value set by data parameter.

**Rapid Traverse Override Signal ROV1、ROV2（Gn014＃0、Gn014＃1）**

- Signal type: PLC→NC
- Signal function: PLC assigns values to ROV1, ROV2 after receiving rapid traverse override input signal, and then transmits it to NC to determine the rapid traverse speed. The override values corresponding to ROV1, ROV2 are shown in the following table:

| ROV2 | ROV1 | Override Value |
|---|---|---|
| 0 | 0 | 100％ |
| 1 | 0 | 50％ |
| 0 | 1 | 25％ |
| 1 | 1 | FO |

- Signal addresses：

| | #7 | #6 | #5 | #4 | #3 | #2 | #1 | #0 |
|---|---|---|---|---|---|---|---|---|
| Gn014 | | | | | | | ROV2 | ROV1 |

## 6.3 Feedrate Override

A programmed feedrate can be reduced or increased by feedrate override signal. This feature is used to check a program. For example, when a feedrate of 100mm/min is specified in the program, setting the override to 50% can move the tool at a speed of 50mm/min.

**Feedrate Override Signal FV0～FV7(Gn012＃0～Gn012＃7)：**

- Signal type: PLC－>CNC
- Signal function: PLC assigns values to FV0~FV7 after receiving the feedrate override input signal, and then transmits them to NC to determine the feedrate. The override values corresponding to FV0~FV7 is shown as follows:

| FV7～FV0（Gn012.7～Gn012.0) | Feedrate Override Value |
|---|---|
| 0000 0000 | 150％ |
| 0000 0001 | 140％ |
| 0000 0010 | 130％ |
| 0000 0011 | 120％ |
| 0000 0100 | 110％ |
| 0000 0101 | 100％ |
| 0000 0110 | 90％ |
| 0000 0111 | 80％ |
| 0000 1000 | 70％ |
| 0000 1001 | 60％ |
| 0000 1010 | 50％ |
| 0000 1011 | 40％ |
| 0000 1100 | 30％ |
| 0000 1101 | 20% |
| 0000 1110 | 10％ |
| 0000 1111 | 0％ |

- Note: During automatic running, actual feedrate is the product of the specified cutting feedrate and the feedrate override value.

- Signal addresses:

| | #7 | #6 | #5 | #4 | #3 | #2 | #1 | #0 |
|---|---|---|---|---|---|---|---|---|
| Gn012 | FV7 | FV6 | FV5 | FV4 | FV3 | FV2 | FV1 | FV0 |

## 6.4 Override Cancel Signal

**OVC（Gn006＃4）：**

- Signal type: PLC→NC
- Signal function: PLC sets OVC to 1 after receiving the override cancel signal, and then transmits it to NC. The feedrate override is clamped at 100%.
- Note: When OVC is 1, the CNC performs as follows:

  1) The feedrate is clamped at 100%, regardless of the feedrate override signal.

  2) The rapid traverse override and spindle speed override are not affected by the signal.
- Signal address:

| | #7 | #6 | #5 | #4 | #3 | #2 | #1 | #0 |
|---|---|---|---|---|---|---|---|---|
| Gn006 | | | | OVC | | | | |

## 6.5 Feedrate 0% Signal

**COVL（Fn212＃4）：**

- Signal type:CNC—> PLC
- Signal function: During the cutting feed, when the override is “0”, the CNC sets the COVEL as “1”; otherwise, it sets the COVL to “0”;
- Precaution: COVL signal can be only inspected during the cutting feed.
- Signal address:

| | #7 | #6 | #5 | #4 | #3 | #2 | #1 | #0 |
|---|---|---|---|---|---|---|---|---|
| Fn212 | | | | COVL | | | | |

# CHAPTER SEVEN MST FUNCTION

When the numbers followed address M, S, T are specified, the corresponding code signal and strobe signals are sent to PLC. PLC works according to the status of these signals. Shown as follows:

| Function | CNC→PLC | | | PLC→CNC |
|---|---|---|---|---|
| | Code signal | Strobe signal | Distribution end signal | End signal |
| Miscellaneous function M | MB00～MB31 | MF | DEN | FIN |
| Spindle speed function S | SB00～SB31 | SF | | |
| Tool function T | TB00～TB31 | TF | | |

The process is shown as follows: (changing M code to S, T codes is the process of spindle speed function and tool function.)

1) Assume that M XXX is specified in a program, an alarm is issued if CNC does not specify the number that followed.

2) After code signal Mbit00~Mbit31 are specified, the strobe signal MF is set to 1, and the command value XXX is expressed by code signal in binary system. When miscellaneous function is commanded along with other instructions, these functions are executed after the code signal of miscellaneous function is sent.

3) When the strobe signal is 1, PLC reads code signal and executes correspondingly.

4) In a block, when an execution is finished, another execution should be started after the distribution end signal DEN is changed to 1.

5) PLC sets the end signal FIN to 1 after the execution. The FIN signal is used in miscellaneous function, spindle speed function and tool function. If these functions are executed simultaneously, the FIN signal can be set to 1 only after all the execution are finished.

6) Only when the signal FIN is 1 (and should be 1 for a while), can CNC set the strobe signal to 0 and inform PLC the receival of end signal.

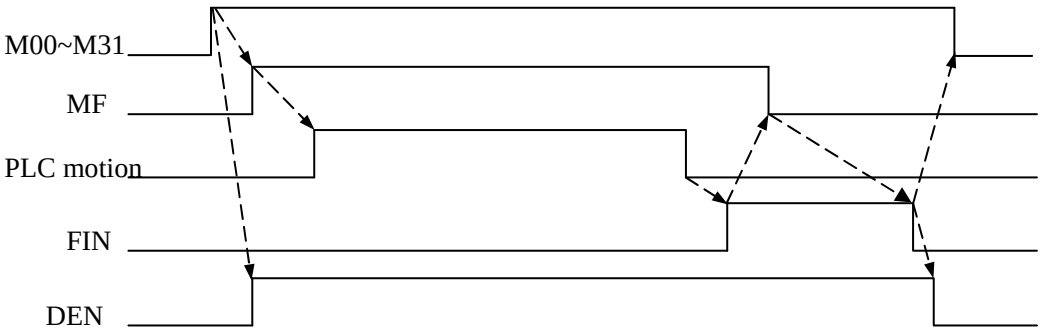
7) When the strobe signal is 0, PLC sets the FIN signal to 0.

8) When the FIN signal is 0, CNC sets all the code signals to 0 and ends all the execution of miscellaneous function.

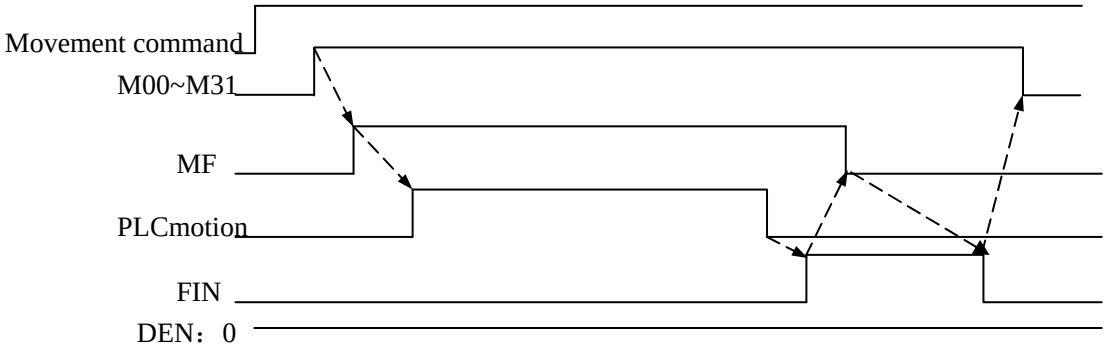
9) When the command execution is finished in a block, CNC preceeds execution to the next block.

In the actual application, the following controllable time-sequence can be selected based upon the logic requirements:

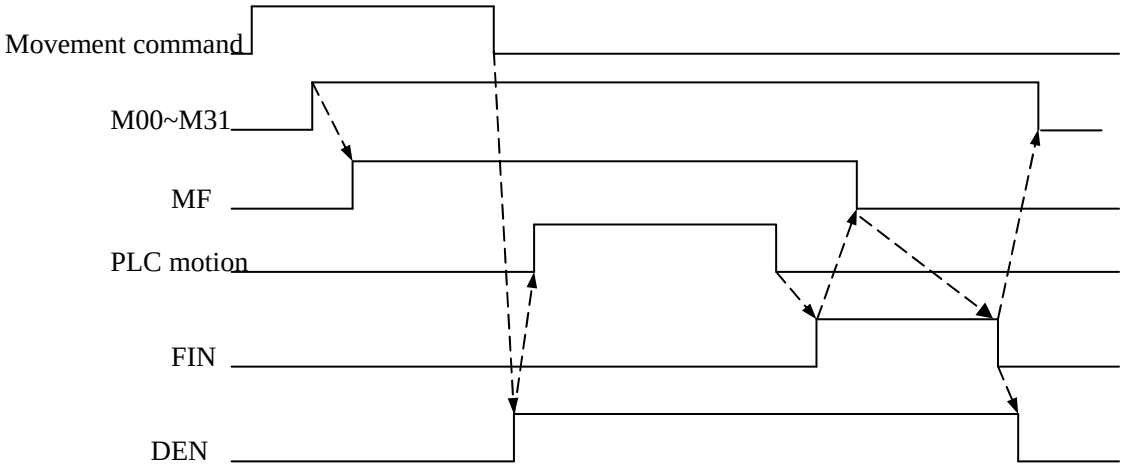
1) Only when one MST function command in the block and without movement command, PLC will use the following time-sequence:




1) When movement command and miscellaneous function are in the same block,the miscellaneous function is executed before the movement command execution is finished




2) When movement command and miscellaneous function are in the same block,the miscellaneous function is executed after the movement command execution is finished




## 7.1 Miscellaneous Function (M Function)

### 7.1.1 Code Signal and Strobe Signal

**Code signal：MB00～MB31（Fn010～Fn013）**

**Strobe signal：MF（Fn007＃0）**

- Signal type: CNC－>PLC
- Signal function: After M code is executed, the corresponding F code signal is set to 1 and MF is set to 1, then these signals are transmitted to PLC. Please refer to the execution

process instruction above for the output condition and execution process. The relationship between M command and code signal is shown as follows:

| F13～F10 | M Command |
|---|---|
| F13, F12, F11, 00000000 | M00 |
| F13, F12, F11, 00000001 | M01 |
| F13, F12, F11, 00000010 | M02 |
| F13, F12, F11, 00000011 | M03 |
| F13, F12, F11, 00000100 | M04 |
| F13, F12, F11, 00000101 | M05 |
| F13, F12, F11, 00000110 | M06 |
| F13, F12, F11, 00000111 | M07 |
| F13, F12, F11, 00001000 | M08 |
| … | … |

- Note:

1: The following miscellaneous function instructions cannot be output even when specified.

A: M98, M99, M198

B: M code for subprogram call

C: M code for custom macro program call

2: Of the miscellaneous function instructions that listed below, decoding signal can also be output in addition to code signal and strobe signal: M00, M01, M02, M30.

3: M00~M31 are in the form of binary code, for example, M5 corresponds to 00000000，00000000, 00000000, 00000101, as listed above.

- Signal addresses:

| | #7 | #6 | #5 | #4 | #3 | #2 | #1 | #0 |
|---|---|---|---|---|---|---|---|---|
| **Fn010** | **MB07** | **MB06** | **MB05** | **MB04** | **MB03** | **MB02** | **MB01** | **MB00** |
| **Fn011** | **MB15** | **MB14** | **MB13** | **MB12** | **MB11** | **MB10** | **MB09** | **MB08** |
| **Fn012** | **MB23** | **MB22** | **MB21** | **MB20** | **MB19** | **MB18** | **MB17** | **MB16** |
| **Fn013** | **MB31** | **MB30** | **MB29** | **MB28** | **MB27** | **MB26** | **MB25** | **MB24** |
| **Fn007** | | | | | | | | **MF** |

## 7.1.2 Decode M Signal

**DM00 (Fn009＃7), DM01 (Fn009＃6), DM02 (Fn009＃5), DM30 (Fn009＃4):**

- Signal type: NC→PLC, valid when it is 1.
- Signal function: When M00, M01, M02, M30 commands are executed, the corresponding decoding signal DM00, DM01, DM02, DM30 are set to 1.

| Program command | Output signal |
|---|---|
| M00 | DM00 |
| M01 | DM01 |
| M02 | DM02 |
| M30 | DM30 |

- Note: The M decoding signal is set to 1 under the following conditions:

  The corresponding miscellaneous function is specified, and other movement command and program stop command have been executed in the same block. (When NC receives the FIN signal prior to the execution of movement command and program stop command, the M decoding signal is not output.)

  The M decoding signal is 0 under the following conditions:

  FIN signal is 1 or CNC is reset.

- Signal addresses:

| | #7 | #6 | #5 | #4 | #3 | #2 | #1 | #0 |
|---|---|---|---|---|---|---|---|---|
| Fn009 | DM00 | DM01 | DM02 | DM30 | | | | |



## 7.2 Spindle Speed Function (S Function)

**Spindle speed code signal SB00～SB31（Fn022～Fn025）、Spindle speed strobe signal SF（Fn007 #2）**

- Signal type: CNC－>PLC
- Signal function: When S command is executed, NC sets the corresponding S code signal to 1, and SF is 1, then NC sends the signal to PLC for logic control. Please refer to the relevant instructions about to the output conditions and process. The relation between S command and binary code of code signal is shown as follows:

| F25～F22 | S Command |
|---|---|
| F25, F24, F23, 00000000 | S00 |
| F25, F24, F23, 00000001 | S01 |
| F25, F24, F23, 00000010 | S02 |
| F25, F24, F23, 00000011 | S03 |
| F25, F24, F23, 00000100 | S04 |
| . | . |

- Signal addresses:

| | #7 | #6 | #5 | #4 | #3 | #2 | #1 | #0 |
|---|---|---|---|---|---|---|---|---|
| Fn022 | SB07 | SB06 | SB05 | SB04 | SB03 | SB02 | SB01 | SB00 |
| Fn023 | SB15 | SB14 | SB13 | SB12 | SB11 | SB10 | SB09 | SB08 |
| Fn024 | SB23 | SB22 | SB21 | SB20 | SB19 | SB18 | SB17 | SB16 |
| Fn025 | SB31 | SB30 | SB29 | SB28 | SB27 | SB26 | SB25 | SB24 |
| Fn007 | | | | | | SF | | |

## 7.3 Tool Function (T Function)

**Tool function code signal**

**TB00～TB31（Fn026～Fn029）, tool function strobe signal TF（Fn007＃3）**

- Signal type: CNC－>PLC
- Signal function: When NC specifies T command, it sets the corresponding T code signal and TF to 1, and then transmits the signal to PLC for logic control. Please refer to the relevant information about to the output conditions and process. The relation between T command and binary code of T code signal is shown as follows:

| F29 | F28 | F27 | F26 | T Command |
|---|---|---|---|---|
| 00000000 | 00000000 | 00000000 | 00000000 | T00 |
| 00000000 | 00000000 | 00000000 | 00000001 | T01 |
| 00000000 | 00000000 | 00000000 | 00000010 | T02 |
| 00000000 | 00000000 | 00000000 | 00000011 | T03 |
| 00000000 | 00000000 | 00000000 | 00000100 | T04 |
| … | … | … | … | … |

- Signal addresses:

| | #7 | #6 | #5 | #4 | #3 | #2 | #1 | #0 |
|---|---|---|---|---|---|---|---|---|
| Fn026 | TB07 | TB06 | TB05 | TB04 | TB03 | TB02 | TB01 | TB00 |
| Fn027 | TB15 | TB14 | TB13 | TB12 | TB11 | TB10 | TB09 | TB08 |
| Fn028 | TB23 | TB22 | TB21 | TB20 | TB19 | TB18 | TB17 | TB16 |
| Fn029 | TB31 | TB30 | TB29 | TB28 | TB27 | TB26 | TB25 | TB24 |
| Fn007 | | | | | TF | | | |

# 7.4 MST Function Completion

## 7.4.1 Completion Signal

**FIN（Gn004＃3）**

- Signal type: PLC－>CNC
- Signal function: When the executions of miscellaneous function, spindle speed function and tool function are finished, PLC sets FIN to 1, and then transmits it to NC.
- Signal address:

| | #7 | #6 | #5 | #4 | #3 | #2 | #1 | #0 |
|---|---|---|---|---|---|---|---|---|
| Gn004 | | | | | FIN | | | |

## 7.4.2 Distribution end signal

**DEN（Fn001＃3）**

- Signal type: CNC－>PLC
- Signal function: When miscellaneous function, spindle speed function, tool function are in the same block with other commands (such as movement command and dwell command), NC sets DEN to 1 after the execution of other commands, and waits for the FIN signal sent by PLC. After the block is executed, DEN is changed to 0.
- Signal address:

| | #7 | #6 | #5 | #4 | #3 | #2 | #1 | #0 |
|---|---|---|---|---|---|---|---|---|
| Fn001 | | | | | DEN | | | |

## 7.4.3 Spindle Function Completion Signal

**SFIN（Gn005＃2）**

- Signal type: PLC－>CNC
- Signal function: When the execution of spindle speed function is finished, PLC sets SFIN to 1, and then transmits it to NC.
- Signal address:

| | #7 | #6 | #5 | #4 | #3 | #2 | #1 | #0 |
|---|---|---|---|---|---|---|---|---|
| Gn005 | | | | | | SFIN | | |

**Note:** When the mult-spindle function or the dual-channel is enabled, perform the S function; confirm whether the spindle function is completed based upon the CNC detection for SFIN.

## 7.5 Miscellaneous Function Lock

### 7.5.1 Miscellaneous Function Lock Signal

**AFL（Gn005＃6）：**

- Signal type:：PLC－>CNC
- Signal function: PLC sets AFL to 1 after receiving the miscellaneous function lock input signal, and then transmits it to NC to prevent the execution of M, S, T functions.
- Note:

  When the AFL signal is 1, CNC works as follows:

  1). During automatic operation or operation in MDI mode, CNC does not execute the specified M, S, T functions, i.e. the code signal and strobe signal are not output.

  2). If AFL is set to 1 after the code signal is output, CNC executes in normal sequence till the end (till it receives the FIN signal and sets the strobe signal to 0).

  3). When AFL is 1, M01, M02, M03 commands can be executed, and the corresponding code signal, strobe signal, decoding signal are output in the normal way.

  4). When AFL is 1, the miscellaneous function M98, M99 are executed normally, but the result is not output.

  5). When AFL is 1, the spindle analog value can still be output.

- Signal address:

| | #7 | #6 | #5 | #4 | #3 | #2 | #1 | #0 |
|---|---|---|---|---|---|---|---|---|
| **Gn005** | | **AFL** | | | | | | |

### 7.5.2 Miscellaneous Function Lock Check Signal

**MAFL（Fn004＃4）：**

- Signal type: CNC－>PLC
- Signal function: When CNC is in the of miscellaneous function locked state, MAFL is set to 1, and then is transmitted to PLC.
- Signal address:

| | #7 | #6 | #5 | #4 | #3 | #2 | #1 | #0 |
|---|---|---|---|---|---|---|---|---|
| **Fn004** | | | | **MAFL** | | | | |

# CHAPTER EIGHT SPINDLE SPEED FUNCTION

## 8.1 Spindle Speed Control

S command is used to specify the analog spindle speed controlled by CNC. For the constant surface speed (in G96 mode), CNC converts the specified surface speed to spindle speed. CNC can output the S command value and SF strobe command to PLC.

**Spindle mode signal SIMSPL（Fn200＃4）:**

- Signal type:CNC－>PLC
- Signal function: CNC sets SIMSPL to 0 or 1, sets control mode of spindle speed，transmits SIMSPL to PLC and informs PLC of current spindle speed control mode.

  SIMSPL “1”： CNC sets analog spindle control; S code is converted into analog value and is transmitted.

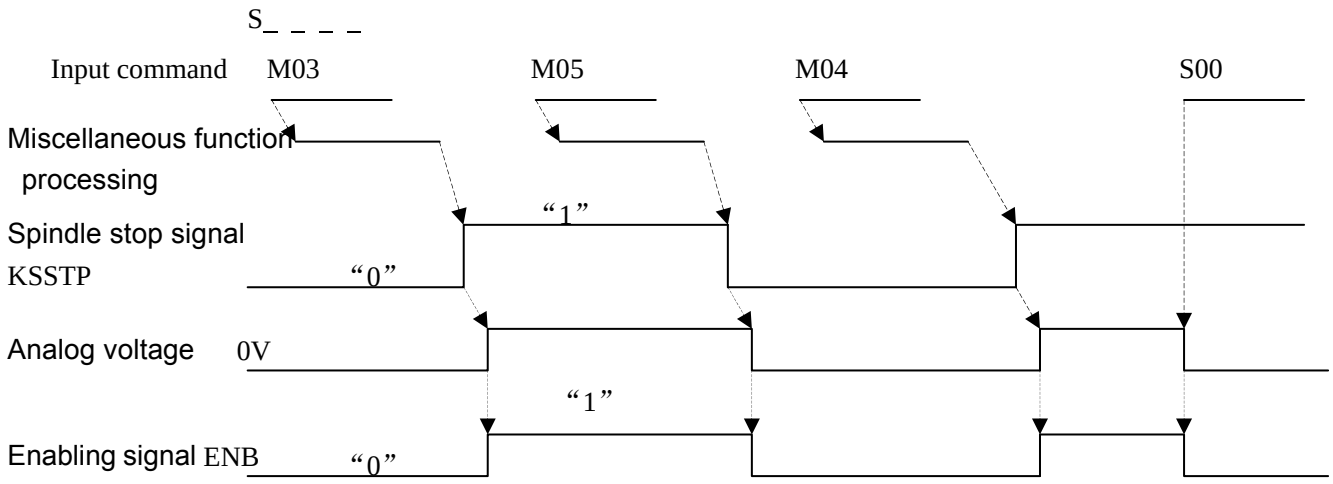
  “0”： CNC sets gear spindle control; S code is converted into switching value and is transmitted.
- Signal address:

| | #7 | #6 | #5 | #4 | #3 | #2 | #1 | #0 |
|---|---|---|---|---|---|---|---|---|
| Fn200 | | | | SIMSPL | | | | |

**Spindle stop signal SSTP KSSTP（Gn029＃6）:**

- Signal type: PLC－>CNC
- Signal function: It stops the output of spindle speed command and sets the S command in NC to 0. The sequence is shown as follows:
- Note: When spindle stop signal *SSTP is 0, the output voltage is changed to 0V. When the signal is 1, the analog voltage output is the command value. When this signal is not used, it is set to 1 so that the CNC can execute spindle speed control.
- Signal address:

| | #7 | #6 | #5 | #4 | #3 | #2 | #1 | #0 |
|---|---|---|---|---|---|---|---|---|
| Gn029 | | KSSTP | | | | | | |




**Spindle speed override signal SOV00～SOV07（Gn030）**

- Signal type: PLC－>CNC
- Signal function: After PLC receives the spindle speed override input signal, it assigns corresponding values to SOV00~SOV07, and then transmits them to NC to set different spindle speed overrides. The relationship between SOV00~SOV07 and the override values is shown as follows:

| SOV3～SOV0 (G30＃3～G30＃0) | The 1st spindle Override | SOV7～SOV4 (G30＃7～G30＃4) | The 2nd spindle Override |
|---|---|---|---|
| 0111 | 50% | 0111 | 50% |
| 0110 | 60% | 0110 | 60% |
| 0010 | 70% | 0010 | 70% |
| 0011 | 80% | 0011 | 80% |
| 0001 | 90% | 0001 | 90% |
| 0000 | 100% | 0000 | 100% |
| 0100 | 110% | 0100 | 110% |
| 0101 | 120% | 0101 | 120% |

- Note: The spindle override function is invalid during tapping cycle and thread cutting.
- Signal address:

| | #7 | #6 | #5 | #4 | #3 | #2 | #1 | #0 |
|---|---|---|---|---|---|---|---|---|
| Gn030 | SOV07 | SOV06 | SOV05 | SOV04 | SOV03 | SOV02 | SOV01 | SOV00 |

**Spindle enable signal ENB（Fn001.4）**

**The Second Spindle enable signal ENB（Fn38.2）**

- Signal type: CNC－>PLC
- Signal function: it represents whether there is the spindle instruction.
- Note: When a non-zero command is output to spindle, the ENB is1; when the command is 0, the ENB signal is changed to logic 0. In analog spindle, even the command output to

spindle is 0 (i.e. the analog voltage is 0V), the spindle motor may work at low speed due to voltage drift of the inverter. In such case, the ENB signal can be used to stop the motor.

- Signal address:

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| **Fn38** | | | | | | **ENB2** | | |
| **Fn1** | | | | **ENB** | | | | |

**Gear selection signal GR1,GR2（Gn28.1，Gn28.2）**

**The second Gear selection signalGR21，GR22(Gn29.0、Gn29.1）**

- Signal type: PLC－>CNC
- Signal function: It informs the CNC the current gear. Refer to the description above for details.
- Signal address:

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| **Gn29** | | | | | | | GR22 | GR21 |
| **Gn28** | | | | | | GR2 | GR1 | |

The corresponding relation between GR1, GR2 and spindle gear is as follows:

| GR2 | GR1 | Gear |
|---|---|---|
| 0 | 0 | 1 |
| 0 | 1 | 2 |
| 1 | 0 | 3 |
| 1 | 1 | 4 |

**Constant surface speed signal CSS（Fn002＃2）**

- Signal type: PLC－>CNC
- Signal function: When the signal is 1, it means the constant surface cutting speed control mode (G96) is executing; when it is 0, it means the mode is not executing.
- Signal address:

| | #7 | #6 | #5 | #4 | #3 | #2 | #1 | #0 |
|---|---|---|---|---|---|---|---|---|
| **Fn002** | | | | | | **CSS** | | |

**Spindle speed arrival signal SAR（Gn029.4）：**

- Signal type:PLC→NC
- Signal function: the signal is used for informing CNC that the spindle speed is in-position.
- Signal address:

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| **Gn29** | | | | **SAR** | | | | |

**Spindle motor speed selection command signal SIND (Gn33.7)**

**The second Spindle motor speed selection command signa SIND2（Gn35.7）**

- Signal type: PLC－>CNC
- Signal function: It is used to select the speed command of spindle motor.

  SIND 1: selects the speed command transmitted from PLC.

  0: selects the speed command transmitted from CNC, i.e. the spindle speed specified by S command.
- Signal address:

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| **Gn33** | **SIND** | | | | | | | |
| **Gn35** | **SIND2** | | | | | | | |

**S12-bit code input signal R01I～R12I（Gn32.0~Gn33.3）**

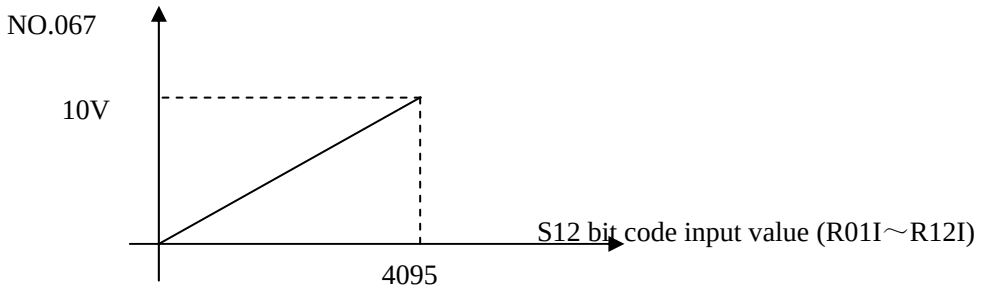
**The second spindle S12-bit code input signal R01I2～R12I2（Gn34.0~Gn35.3）**

- Signal type:PLC→NC
- Signal function: it is instruction value the spindle motor speed transmitted from PLC. It specifies the spindle motor speed in binary and its speed is 0～4095.
- Signal address:

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| **Gn32** | R08I | R07I | R06I | R05I | R04I | R03I | R02I | R01I |
| **Gn33** | | | | | R12I | R11I | R10I | R09I |
| **Gn34** | R08I2 | R07I2 | R06I2 | R05I2 | R04I2 | R03I2 | R02I2 | R01I2 |
| **Gn35** | | | | | R12I2 | R11I2 | R10I2 | R09I2 |

## Instruction signal of spindle gear shifting SFTREV_L，SFTREV_H（F215, F216）

- Signal type: CNC－>PLC
- Signal function: the signal informs PLC of the spindle motor speed instruction in gear shifting and specifies the spindle motor speed in binary. Its speed is 0～4095. Relation between the speed and 10V is as follows（F216,F215）=NO.067×4095/10000）:



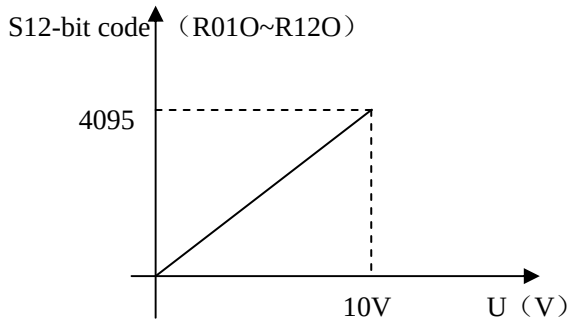

- Signal address:

| | |
|---|---|
| **Fn215** | **SFTREV_L** |
| **Fn216** | **SFTREV_H** |

**S12 bit code signal R01O~R12O（Fn036.0~Fn037.3）R01O2~R12O2（Fn236.0~Fn237.3）**

- Signal type: NC→PLC
- Signal function: This signal shifts the spindle speed command value calculated by NC spindle

controllable function into the data 0～4095, and then output its results to PLC. The relation between 10V and S12-bit code is shown below:




●Signal address:

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Fn36 | R08O | R07O | R06O | R05O | R04O | R03O | R02O | R01O |
| Fn37 | | | | | R12O | R11O | R10O | R09O |
| Fn236 | R08O2 | R07O2 | R06O2 | R05O2 | R04O2 | R03O2 | R02O2 | R01O2 |
| Fn237 | | | | | R12O2 | R11O2 | R10O2 | R09O2 |

**The 1st spindle velocity command value input INP00～INP15 (Fn240＃0～Fn241＃7)**

- Signal type: CNC－>PLC
- Signal function:CNC calculates the code information of the spindle speed command containing the spindle override.
- Signal address:

| | #7 | #6 | #5 | #4 | #3 | #2 | #1 | #0 |
|---|---|---|---|---|---|---|---|---|
| Fn240 | INP07 | INP06 | INP05 | INP04 | INP03 | INP02 | INP01 | INP00 |
| Fn241 | INP15 | INP14 | INP13 | INP12 | INP11 | INP10 | INP09 | INP08 |

**The actual spindle speed signal of the 1st spindle AR00～AR15 (Fn040～Fn041)**

- Signal type: CNC－>PLC
- Signal function: The spindle actual rotation speed of the feedback pulse index outputs to PLC from CNC, which is issued from the position encoder installed on the spindle.
- Signal address:

| | #7 | #6 | #5 | #4 | #3 | #2 | #1 | #0 |
|---|---|---|---|---|---|---|---|---|
| Fn040 | AR07 | AR06 | AR05 | AR04 | AR03 | AR02 | AR01 | AR00 |
| Fn041 | AR15 | AR14 | AR13 | AR12 | AR11 | AR10 | AR09 | AR08 |

## Spindle Jog Signal SPD（Gn200＃0）

- Signal type: PLC－>CNC
- Signal function: The specified function is enabled when the analoy spindle function is valid. SPD sets to “0” when the spindle is normally rotated.
- Precaution: Start the spindle, and then it rotates based upon the data parameter (No.109).

- Signal address:

| | #7 | #6 | #5 | #4 | #3 | #2 | #1 | #0 |
|---|---|---|---|---|---|---|---|---|
| Gn200 | | | | | | | | SPD |

### Spindle Jog Detection Signal MSPHD（Fn199＃0）

- Signal type: CNC－>PLC
- Signal function: when CNC is executing the spindle jog, NC sets MSPHD to 1；when NC is not the spindle jog mode，NC sets MSPHD to 0。
- Signal address:

| | #7 | #6 | #5 | #4 | #3 | #2 | #1 | #0 |
|---|---|---|---|---|---|---|---|---|
| Fn199 | | | | | | | | MSPHD |



## 8.2 Multiple Spindles Control

**Spindle selection signal SWS1（Gn027＃0），SWS2（Gn027＃1）**

- Signal type: PLC－>NC
- Signal function: It indicates whether the S command is output to spindle or not.

**SWS1** 1: Outputs to the 1st spindle

0: Does not output to the 1st spindle.

**SWS2** 1: Outputs to the 2nd spindle

0: Does not output to the 2nd spindle

- Signal address:

| | #7 | #6 | #5 | #4 | #3 | #2 | #1 | #0 |
|---|---|---|---|---|---|---|---|---|
| Gn027 | | | | | | | SWS2 | SWS1 |

### Spindeles stop signal SSTP1（Gn027＃3）, SSTP2（Gn027＃4）

- Signal type: PLC－>NC
- Signal function: It can stops all the spindles. (Only valid for multiple spindles)

**SSTP1** 1: Does not output 0 rotation/min. to the 1st spindle.

0: Outputs 0 rotation/min. to the 1st spindle.

**SSTP2** 1: Does not output 0 rotation/min. to the 2nd spindle

0: Outputs 0 rotation/min. to the 2nd spindle.

- Signal address:

| | #7 | #6 | #5 | #4 | #3 | #2 | #1 | #0 |
|---|---|---|---|---|---|---|---|---|
| Gn027 | | | | SSTP2 | SSTP1 | | | |

**Gear selection signal GR21（Gn029＃0）, GR22（Gn029＃1）**

- Signal type: PLC－>CNC
- Signal function: It selects the gear of the 2nd spindle when the multiple spindles are installed.

  The relationship between GR21、GR22 signal and gear is shown as follows:

| GR22 | GR21 | gear |
|---|---|---|
| 0 | 0 | 1 |
| 0 | 1 | 2 |
| 1 | 0 | 3 |
| 1 | 1 | 4 |

- Signal address:

| | #7 | #6 | #5 | #4 | #3 | #2 | #1 | #0 |
|---|---|---|---|---|---|---|---|---|
| Gn029 | | | | | | | GR22 | GR21 |

**The 2nd spindle motor speed selection command signal SIND2（Gn35＃7）**

- Signal type:PLC－>CNC
- Signal function: It selects the speed command of the 2nd spindle motor.

  **SIND2** 1: selects the speed command from PLC.

  0: selects the speed command from CNC, i.e. the spindle speed specified by S command.

- Signal address:

| | #7 | #6 | #5 | #4 | #3 | #2 | #1 | #0 |
|---|---|---|---|---|---|---|---|---|
| Gn035 | SIND2 | | | | | | | |

**The S12 bit code signal of the 2nd spindle R01O2～R12O2 (Fn236＃0～Fn237＃3)**

- Signal type: CNC－>PLC
- Signal function: This signal shifts the speed speed command into the code signal of the 0～0xFFF calculated by CNC.
- Signal address:

| | #7 | #6 | #5 | #4 | #3 | #2 | #1 | #0 |
|---|---|---|---|---|---|---|---|---|
| Fn236 | R08O2 | R07O2 | R06O2 | R05O2 | R04O2 | R03O2 | R02O2 | R01O2 |
| Fn237 | | | | | R12O2 | R11O2 | R10O2 | R09O2 |

**The 2nd spindle motor speed command input signal R01I2～R12I2（Gn034＃0～Gn035＃3）**

- Signal type:PLC－>CNC
- Signal function: It indicates the input of the 2nd spindle motor speed command from PLC.
- Signal address:

| | #7 | #6 | #5 | #4 | #3 | #2 | #1 | #0 |
|---|---|---|---|---|---|---|---|---|
| Gn034 | R08I2 | R07I2 | R06I2 | R05I2 | R04I2 | R03I2 | R02I2 | R01I2 |
| Gn035 | | | | | R12I2 | R11I2 | R10I2 | R09I2 |

## 8.3 Spindle Position/Speed Switch

**Spindle contour control switching signal CON（Gn027＃7）**

**The second Spindle contour control switching signal CON2（Gn127＃7）**

- Signal type: PLC－>NC
- Signal function: It specifies the Cs contour control function. The control mode can be spindle speed control or Cs contour control. When the signal is 1, the control mode is switched to Cs contour control mode; when the signal is 0, spindle speed control mode is switched back.
- Signal address:

| | #7 | #6 | #5 | #4 | #3 | #2 | #1 | #0 |
|---|---|---|---|---|---|---|---|---|
| Gn027 | CON | | | | | | | |
| Gn127 | CON2 | | | | | | | |

**Spindle contour control switching completion signal FSCSL（Fn044＃1）**

**The second Spindle contour control switching completion signal FSCSL2（Fn144＃1）**

- Signal type:CNC－>PLC
- Signal function: When this signal is 0, it indicates the controlled axis is in spindle speed control mode. When the signal is 1, it indicates the controlled axis is in the Cs contour control mode.
- Signal address:

| | #7 | #6 | #5 | #4 | #3 | #2 | #1 | #0 |
|---|---|---|---|---|---|---|---|---|
| Fn044 | | | | | | | FSCSL | |
| Fn144 | | | | | | | FSCSL2 | |

## 8.4 Rigid Tapping

### 8.4.1 Rigid Tapping Signal

**RGTAP（Gn061＃0）**

- Signal type: PLC－>CNC
- Signal function: PLC sets its side to rigid tapping method based upon the M29 (rigid tapping

method preaparation miscellaneous function) command, and then notice to the CNC after this signal is set to “1”.

- Output condition: It is set to “1” in the following case:

  PLC is already on the rigid tapping method.

  It is set to “0” in the following case:

  PLC does not on the rigid tapping method yet.

- Signal address:

| | #7 | #6 | #5 | #4 | #3 | #2 | #1 | #0 |
|---|---|---|---|---|---|---|---|---|
| Gn061 | | | | | | | | RGTAP |

## 8.4.2 Signal in Rigid Tapping

**RTAP（Fn076＃3）**

- Signal type: CNC－>PLC
- Signal function: This signal is used for noticing the PLC; CNC is the signal in the rigid tapping.
- Output condition: It is set to “1” in the following case:

  CNC is on the rigid tapping method.

  It is set to “0” in the following case:

  CNC is on the non-rigid tapping method.

- Signal address:

| | #7 | #6 | #5 | #4 | #3 | #2 | #1 | #0 |
|---|---|---|---|---|---|---|---|---|
| Fn076 | | | | | RTAP | | | |

## 8.4.3 Spindle Rotation Direction Signal

**RGSPM (Fn065＃1), RGSPP (Fn065＃0)**

- Signal type: CNC－>PLC
- Signal function: These signals are used for noticing the PLC, which is positive or negative of the spindle in the rigid tapping.

  **RGSPP** 1: Spindle positive in rigid tapping method;

  0: Spindle negative in rigid tapping method.

  **RGSPM** 1: Spindle negative in rigid tapping method;

  0: Spindle positive in rigid tapping method;

- Output condition: Signal outputs when the spindle operates in rigid tapping. Therefore, the signal does not output even if the motion locates to the hole position in the rigid tapping method, as well dwells at the bottom of the hole. Certainly, the signal does not output in the feed hold and single block stop. This signal is only enabled in the rigid tapping method; usually, it is not output signal (RGSPP and RGSPM are regarded as “0”)

under the control of the spindle.

- Signal address:

| | #7 | #6 | #5 | #4 | #3 | #2 | #1 | #0 |
|---|---|---|---|---|---|---|---|---|
| Gn065 | | | | | | | RGSPM | RGSPP |

## 8.4.4 Rigid Tapping Spindle Selection Signal

**RGTSP2 (Gn061＃5), RGTSP1 (Gn061＃4)**

- Signal type: PLC－>CNC
- Signal function: These signals can be used for selecting the spindle of the rigid tapping in its rigid tapping method.

| Rigid tapping spindle | RGTSP2 | RGTSP1 |
|---|---|---|
| Alarm occurs | 0 | 0 |
| Spindle 2 | 0 | 1 |
| Spindle 2 | 1 | 0 |

- Precaution: It is better input these signals before specifying the rigid tapping command. In addition, do not shift before the end of the rigid tapping.
- Signal address:

| | #7 | #6 | #5 | #4 | #3 | #2 | #1 | #0 |
|---|---|---|---|---|---|---|---|---|
| Gn061 | | | RGTSP2 | RGTSP1 | | | | |

## 8.5 T-Type Shift

**Spindle gear selection signal GR1(G28.1), GR2(G28.2), GR21(G29.0), GR22(G29.1)**

- Signal type:PLC→NC
- Signal function: Notice the system treating the analog voltage for the corresponding gear.

| | GR2(G28.2) | GR1(G28.1) | Gear |
|---|---|---|---|
| The 1st spindle | 0 | 0 | 1 |
| | 0 | 1 | 2 |
| | 1 | 0 | 3 |
| | 1 | 1 | 4 |
| | GR22(G29.1) | GR21(G29.0) | |
| The 2nd spindle | 0 | 0 | 1 |
| | 0 | 1 | 2 |
| | 1 | 0 | 3 |
| | 1 | 1 | 4 |

- Signal address:

| | #7 | #6 | #5 | #4 | #3 | #2 | #1 | #0 |
|---|---|---|---|---|---|---|---|---|
| G028 | | | | | | GR2 | GR1 | |
| G029 | | | | | | | GR22 | GR21 |

**Note: The signal in this Section is only suitable for the Milling machine system.**

## 8.6 M-Type Gear

**Gear selection output signal GR10(F34.0), GR20(F34.1), GR30(F34.2), GR40(F34.3)**

- Signal type: NC→PLC
- Signal function: Notice the PLC treating the corresponding gear.

| GR10(F34.3) | GR10(F34.2) | GR20(F34.1) | GR10(F34.0) | Selected gear |
|---|---|---|---|---|
| 0 | 0 | 0 | 1 | The 1st gear |
| 0 | 0 | 1 | 0 | The 2nd gear |
| 0 | 1 | 0 | 0 | The 3rd gear |
| 1 | 0 | 0 | 0 | The 4th gear |

- Signal address:

| | #7 | #6 | #5 | #4 | #3 | #2 | #1 | #0 |
|---|---|---|---|---|---|---|---|---|
| F034 | | | | | GR40 | GR30 | GR20 | GR10 |

**Spindle speed strobe signal SF（F7.2）**

- Signal type:NC→PLC
- Signal function: The CNC firstly output the gear selection signal when shifting the gear selection signal. Output SF signal after passing the deleay time of the SF. PMC shifts the gear while receiving the SF signal, and then return to the FIN signal after shifting. The exchange between SF and Fin are same to the output of the S code.
- Signal address:

| | #7 | #6 | #5 | #4 | #3 | #2 | #1 | #0 |
|---|---|---|---|---|---|---|---|---|
| F007 | | | | | | | SF | |

**Note: The signal in this Section is only suitable for the Milling machine system.**

## 8.7 Index Table Function

The index function of the index table is performed the automatic releasing and clamping for the index table after specifying the index axis position (the angle of the rotation axis A, B and C).

**Index table releasing completion feedback signal BEUCL (G38#6)**

- Signal assortment: Signal input
- Signal type:PLC－>NC
- Signal function: G38.6: Notice that the CNC index axis has been completed the releasing of the mechanical clamping, this signal is enabled in the Low Level.
- Signal address:

| | #7 | #6 | #5 | #4 | #3 | #2 | #1 | #0 |
|---|---|---|---|---|---|---|---|---|
| G38 | | BEUCL | | | | | | |

**Index table clamping completion feedback signal BECLP (G38#7)**

- Signal assortment: Signal input
- Signal type:PLC－>NC
- Signal function: Notice the index clamping completion of CNC. This signal is enabled in the Low Level.
- Signal address:

| | #7 | #6 | #5 | #4 | #3 | #2 | #1 | #0 |
|---|---|---|---|---|---|---|---|---|
| G38 | BECLP | | | | | | | |

**Index table releasing signal BUCLP F61#0**

- Signal assortment: Signal output
- Signal type:NC－>PLC
- Signal function: Issue a signal to PMC, and then release the mechanical clamping of the index axis.

  F61.1：Issue a signal to PMC, and then perform the mechanical clamping of the index axis.
- Signal address:

| | #7 | #6 | #5 | #4 | #3 | #2 | #1 | #0 |
|---|---|---|---|---|---|---|---|---|
| F61 | | | | | | | | BUCLP |

**Index table clamping signal BUCLP F61#1**

- Signl assortment: Signal output
- Signal type:NC－>PLC
- Signal function: Issue a signal to PMC, and then perform the mechanical clamping of the index axis.
- Signal address:

| | #7 | #6 | #5 | #4 | #3 | #2 | #1 | #0 |
|---|---|---|---|---|---|---|---|---|
| F61 | | | | | | | BCLP | |

**Note: The signal in this Section is only suitable for the Milling machine system.**

# CHAPTER NINE TOOL FUNCTION

## 9.1 Tool-Change Function

When T code or HDT signal is specified, NC compares the desired tool numbers with the current tool numbers NT00~NT31, if the numbers are consistent, tool change will not be executed; if not, the code signal and strobe signal of the desired tool number are generated, and the machine selects tools accordingly.

In this CNC system, the tool change can be performed by T command in AUTO or MDI mode, or the sequence tool change be performed through HDT signal in MANUAL mode.

**Tool change by T command**

Tool change can be performed through T command in AUTO or MDI modes. CNC sends the tool number signal and tool strobe signal after it decodes the T command, and then waits for the completion of PLC tool change

**Tool-change in manual sequence**

NC adds 1 based upon the current tool number when performing the tool-change by manual, which is performed the tool-change based upon the tool number of the desired next cutter. If the current tool number that adds 1 is more than the overall cutter digit, in this case, the desired next tool number is set to 1.

The execution procedure of the CNC manual sequence tool-change is identical with the one of the T command; NC delivers the next tool-position number and strobe signal, and then waiting for the tool-change completion of the PLC. The delivered tool-position number in the CNC manual sequence tool-change is added 1 based upon the current tool number; however, when the T command performs the tool-change, the issued tool-position number is speicified tool number by the T command.

### 9.1.1 Current Tool Number Signal

**NT00～NT31（Gn201～Gn204）**

- Signal type: PLC－>CNC
- Signal function: After PLC has tested the current tool selection signal, it sets NT00～NT31 to the corresponding value, transmits to NC and informs NC of current tool number. These signals are to represent tool numbers with binary code.
- Signal address:

| | #7 | #6 | #5 | #4 | #3 | #2 | #1 | #0 |
|---|---|---|---|---|---|---|---|---|
| **Gn201** | NT07 | NT06 | NT05 | NT04 | NT03 | NT02 | NT01 | NT00 |
| **Gn202** | NT15 | NT14 | NT13 | NT12 | NT11 | NT10 | NT09 | NT08 |

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| **Gn203** | NT23 | NT22 | NT21 | NT20 | NT19 | NT18 | NT17 | NT16 |
| **Gn204** | NT31 | NT30 | NT29 | NT28 | NT27 | NT26 | NT25 | NT24 |

### 9.1.2 Manually Sequential Tool Change

**HDT（Gn044#7）**

- Signal type: PLC－>CNC
- Signal function: PLC sets HDT to 1 and transmits to NC; NC will execute tool change sequentially according to current tool number.
- Signal address:

| | #7 | #6 | #5 | #4 | #3 | #2 | #1 | #0 |
|---|---|---|---|---|---|---|---|---|
| **Gn044** | **HDT** | | | | | | | |



### 9.1.3 Overall Tool-Position Number

**TMAX（Fn207）：**

- Signal type: CNC－>PLC
- Signal function: The TMAX signal at the CNC channel notices to the PLC of the current oval tool-position number after the CNC modifies the overall tool-position number.
- Signal address:

| | #7 | #6 | #5 | #4 | #3 | #2 | #1 | #0 |
|---|---|---|---|---|---|---|---|---|
| **Fn207** | **T07** | **T06** | **T05** | **T04** | **T03** | **T02** | **T01** | **T00** |

## 9.2 Tool Life Administration

The tool group can be divided into several groups, specify the tool life-span (usage times or time) in each group in advance. In addition, count the life spane when using the tool of each group each time. Automatically select a new tool based upon the arranged sequence beforehand within the same group. Therefore, the tool life span can be administrated while the machining can be performed continuously.

### 9.2.1 Tool Change Signal

**TLCH（Fn064＃0）**

- Signal type: CNC－>PLC
- Signal function: inform PLC that the last tool life ends of in this group
- Output condition: The following case is regarded as “1”:

  The last tool life in one group ends, all other tool life ends

The following cases are treated as “0”:

A: The last tool life span does not end at one group.

B: When the tool resetting signal TLRST change sets to “1”.

- Precaution: The overall tool groups used in the program should be performed a traverse when the end (M02, M30, M99) of the program. If the overall life spans of tools are issued, the TLCH signal is then occurred.
- Signal address:

| | #7 | #6 | #5 | #4 | #3 | #2 | #1 | #0 |
|---|---|---|---|---|---|---|---|---|
| Fn064 | | | | | | | | TLCH |

## 9.2.2 New Tool Selection Signal

**TLNW（Fn064＃1）**

- Signal type: CNC－>PLC
- Signal function: inform a new tool in some tool group has been selected
- Output condition: The following case is regarded as “1”:

  A new tool in some tool group has been selected.

  The following case is treated as “0””

  When the completion of the M.S. function signal sets to “1”.
- Signal address:

| | #7 | #6 | #5 | #4 | #3 | #2 | #1 | #0 |
|---|---|---|---|---|---|---|---|---|
| Fn064 | | | | | | | TLNW | |

## 9.2.3 Tool Change Resetting Signal

**TLRST（Gn048＃7）**

- Signal type: PLC－>CNC
- Signal function: clear all execution data

  When the signal is set to 1, the control unit clears the used tool life data of in all groups; the tool state is reset to the unused.
- Note: when the automatic operation signal OP is “0”, the tool change reset signal TLRST is valid.
- Signal address:

| | #7 | #6 | #5 | #4 | #3 | #2 | #1 | #0 |
|---|---|---|---|---|---|---|---|---|
| Gn048 | TLRST | | | | | | | |

### 9.2.3 Tool Skip Signal

**TLSKP（Gn048＃5）**

- Signal type: PLC－>CNC
- Signal function: It can skip the tool that its life span not reaches to the end yet, and then select the next tool by force.

  Change the tool that does not reach to the life span based upon the one of the following two methods:

  A: Tool group number selects the signal to specify the group number of the tool based upon the setting of the parameter; when the tool skip signal TLSKP becomes to “1”, the next T code command then skips the being used tool of the current group instead of using the one that does not run out the life span in the specified group.

  B: Do not specify the group number by parameter setting. When the current skip signal TLSKP turns into “1”, the machine tool will skip to the next tool at the current used group.
- **N**ote: When cycle start light signal (STL) and feed pause light signal (SPL) must be “0”, inputting TLSKP signal is valid.
- Signal address:

| | #7 | #6 | #5 | #4 | #3 | #2 | #1 | #0 |
|---|---|---|---|---|---|---|---|---|
| Gn048 | | | TLSKP | | | | | |

### 9.2.4 Tool group number selection signal

**TL01～TL16（Gn047＃0～Gn047＃4）**

- Signal type: PLC－>CNC
- Signal function: Specify the tool group number. Which group will perform the tool skip based upon the specification of this signal before inputting the tool skip signal TLSKP. The specified group number is the +1 of the specified value.
- Signal address:

| | #7 | #6 | #5 | #4 | #3 | #2 | #1 | #0 |
|---|---|---|---|---|---|---|---|---|
| Gn047 | | | | TL16 | TL08 | TL04 | TL02 | TL01 |

## 9.3 Tool Measure

### 9.3.1 Tool Compensation Value Write Selection Signal

**GOQSM（Gn039#7）**

- Signal type: PLC－>CNC

- Signal function: Select the write method of the tool compensation value
- Signal address:

| | #7 | #6 | #5 | #4 | #3 | #2 | #1 | #0 |
|---|---|---|---|---|---|---|---|---|
| Gn039 | GOQSM | | | | | | | |

## 9.3.2 Tool Compensation Number Selection Signal

**OFN0~OFN5（Gn039＃0～Gn039＃5）**

- Signal type: PLC－>CNC
- Signal function: Select the tool compensation number. When the tool compensation value write method is selected, the cursor is automatically pointed the selected tool shape compensation number position which is selected by the tool compensation number. The tool compensation number can be selected by 6 code signals (Binary code).
- Signal address:

| | #7 | #6 | #5 | #4 | #3 | #2 | #1 | #0 |
|---|---|---|---|---|---|---|---|---|
| Gn039 | | | OFN5 | OFN4 | OFN3 | OFN2 | OFN1 | OFN0 |

## 9.3.2 Workpiece Coordinate System Offset Value Write Method Selection Signal

**WOQSM（Gn039#6）**

- Signal type: PLC－>CNC
- Signal function: Select the write method of the workpiece coordinate system offset value. When the WOQSM signal sets to “1”, which is became to the write method of the workpiece coordinate system offset value.
- Signal address:

| | #7 | #6 | #5 | #4 | #3 | #2 | #1 | #0 |
|---|---|---|---|---|---|---|---|---|
| Gn039 | | WOQSM | | | | | | |

## 9.3.3 Workpiece Coordinate System Offset Value Write Signal

**WOSET（Gn040#7）**

- Signal type: PLC－>CNC
- Signal function: WOSET signal becomes “1” when selecting the write method of the workpiece coordinate system offset value; the system will be automatically calculated and then set the offset value of the workpiece coordinate system along Z axis.

- Signal address:

| | #7 | #6 | #5 | #4 | #3 | #2 | #1 | #0 |
|---|---|---|---|---|---|---|---|---|
| Gn040 | WOSET | | | | | | | |

### 9.3.4 Position Record Signal

**PRC（Gn040#6）**

- Signal type: PLC－>CNC
- Signal function: This signal is the one when the tool compensation amout measure value is directly input. Try to store the position within the controllable equipment when cutting workpiece by this signal, input the value of the actual workpiece dimension by measuring based upon the described operation, and then store the difference which is regarded as the described tool compensation value. When the PRC signal becomes “1”, the CNC stores the current position of each axis at that moment.
- Signal address:

| | #7 | #6 | #5 | #4 | #3 | #2 | #1 | #0 |
|---|---|---|---|---|---|---|---|---|
| Gn040 | | PRC | | | | | | |

## 9.4 Tool-Magazine Page Signal

**TYN（G176.7）**

- Signal type:PLC->NC
- Signal function:TYN=1：Tool setting page enabled;
  TYN=0：Tool setting page disabled.
- Signal address:

| | #7 | #6 | #5 | #4 | #3 | #2 | #1 | #0 |
|---|---|---|---|---|---|---|---|---|
| G176 | TYN | | | | | | | |

**Note: This signal is only used for Milling Machine System.**

# CHAPTER TEN PLC AXIS CONTROL FUNCTION

## 10.1 Function Overview

Any axes which are independent from CNC management can be directly controlled by PLC signals, and the axis can just be controlled by the signals without CNC program. For example, after the movement amount and the feedrate commands are sent from PLC, the axis can be moved respectively and it is not relative with the other axis controlled by CNC. Therefore, the turret, the tray, the indexing table and the peripheral devices, etc can be controlled by any axis of CNC. Whether each axis is controlled by CNC or PLC is set by the input signals.

The commands can be controlled by PLC axis are as below:

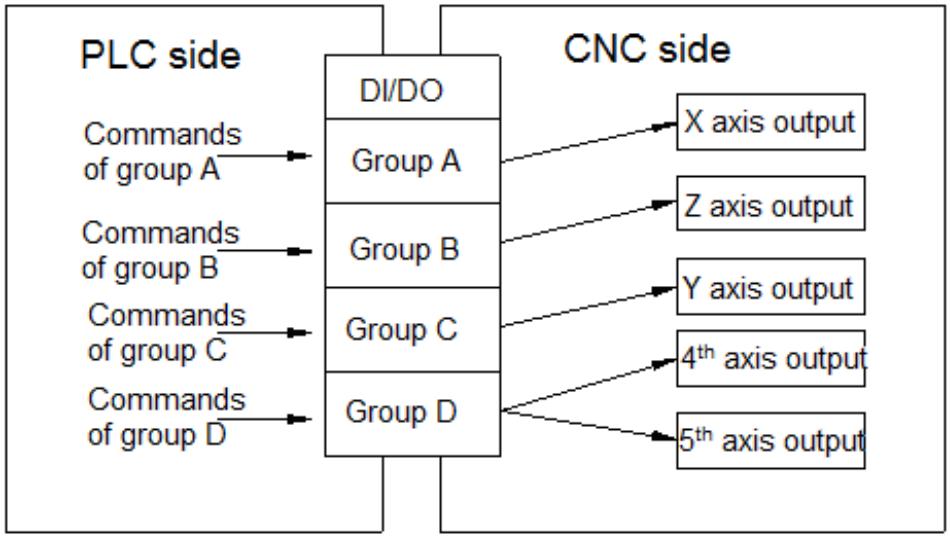
1) Rapid traverse;
2) Cutting feed per minute;
3) Cutting feed per revolution;
4) Dwell;
5) The reference position return;
6) The 1st reference position return;
7) The 2nd reference position return;
8) The 3rd reference position return;
9) The 4th reference position return;
10) Miscellaneous function.



## 10.2 Function Explanation

During PLC axis control, the various operations are controlled by signals. Though the relative commands specified by PLC, the control axis operation can be performed.

CNC provides the input/output signal channels of four groups, the signals of four groups are repectively called as groups A, B, C and D. The “group” of the input and output signals controlled by PLC axis is the unit controlled by PLC axis. Which axis is controlled by the dedicated “group” is set by the data parameters (select DI/DO group for each axis controlled by PLC) in advance. Many axes can be independently controlled by commanding many groups meanwhile.



**Note 1: The relation between the group and the axis is set by the data parameters (select DI/DO group for each axis controlled by PLC);**

**Note 2: Many axes can be controlled by the channel of group 1 meanwhile, which is same with group D.**

**Note 3: Groups A, B, C and D can be independently commanded.**

About the addresses of the signals please refer to the following list:

| DI/DO group | Input signal address | Output signal address |
|---|---|---|
| **Group A** | G142～G148，G150＃5，G150＃0，＃1，＃6，＃7 | F130～F132，F142，F129＃5，＃7 |
| **Group B** | G154～G161，G162＃5，G150＃0，＃1，＃6，＃7 | F133～F135，F145，F129＃5，＃7 |
| **Group C** | G166～G173，G174＃5，G150＃0，＃1，＃6，＃7 | F136～F138，F148，F129＃5，＃7 |
| **Group D** | G178～G185，G186＃5，G150＃0，＃1，＃6，＃7 | F139～F141，F151，F129＃5，＃7 |

In the following description, the input/output signals from 4 channels are respectively group A (channel 1), group B (channel 2), group C (channel 3) and group D (channel 4).

The names of input/output signals for PLC axis control always include the lower-case “g”, like EBUFg. However, such signal EBUFg doesn’t exist. The actual signals represented by EBUFg are EBUFA，EBUFB，EBUFC and EBUFD, which are respectively corresponded to group A (channel 1), group B (channel 2), group C (channel 3) and group D (channel 4) .



## 10.3 Basic Operation Steps

The basic operation steps used by PLC axis control function are as below:

1) The status parameter PLA is set as “1” (Please power on, again after rewriting the parameters) and PLC axis control function is started. Meanwhile, the axis controlled by PLC is valid.
2) Whether any axis in one of groups A～D is used is set by the parameter. Moreover, when many axes are moved meanwhile in one group, the parameters relative with the feedrate (like the rapid traverse rate, the acceleration/deceleration time constant, the linear axis/the rotary axis) should be set with the same conditions.When X axis of the turning machine and other axes are commonly used with one DI/DO channel, please pay attention to setting the radius/diameter programming parameter. The reason is for X axis of the turning machine the dedicated movement amount is relative with the parameter.
3) Firstly, the axis must be switched into PLC control one, that is to say, the selected signals EAX1～EAX5 of the controlled axes are set as “1”.
4) Specifying PLC axis execution movement

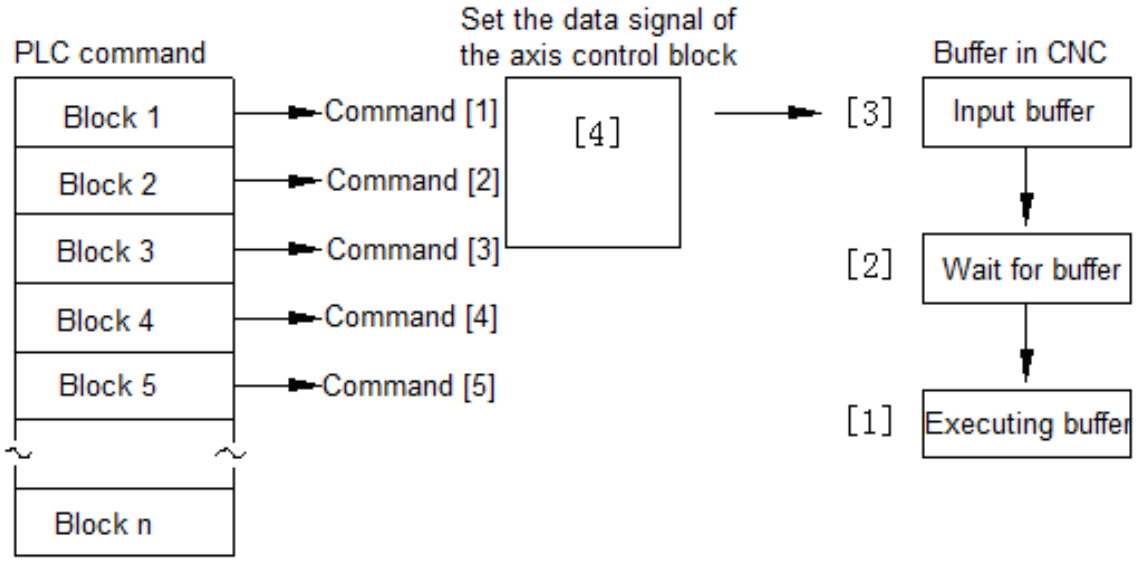
   The axis control signals EC0g～EC06g are for commanding operation. The axis control feeding signals EIF0g～EIF15g are for commanding feedrate. The axis control data signals EID0g～EID31g are for commanding the movement distance and the other data. The signals and the block stop signal ESBKg can be executed one complete operation; it means one block is executed during the automatic operation controlled by CNC. These signals are unified called as the axis control block data signals.

The control signal list in one block controlled by PLC

| Joint name | Signal name | Symbol | Type |
|---|---|---|---|
| Axis control block data signal | Block stop signal | EMSBKg | Bit |
| | Axis control command signal | EC0g～EC6g | Byte |
| | Axis control feedrate signal | EIF0g～EIF15g | Character |
| | Axis control data signal | EID0g～EID31g | 2 characters |

5) When the data of one complete operaton (one block) are confirmed, the logic status (i.e from “0” to “1”, or vise verse) of the control command reading signal of the reversal signal should be set. Therefore, the logic status of the axis control reading finish signal EBSYg must be same as that of EBUFg.

6) Based on PLC axis control function, many continuous movements are executed by PLC, so the blocks in CNC can be executed buffering. Therefore, even one block is being executed, as long as the buffer in CNC is idle, it will read the next block into CNC. During the buffering controlled by PLC axis and in the process of executing command (1), commands [2] and [3] are read into the CNC buffer, command [4] has been in the finish status (setting the axis control block data signal).




**Note:**

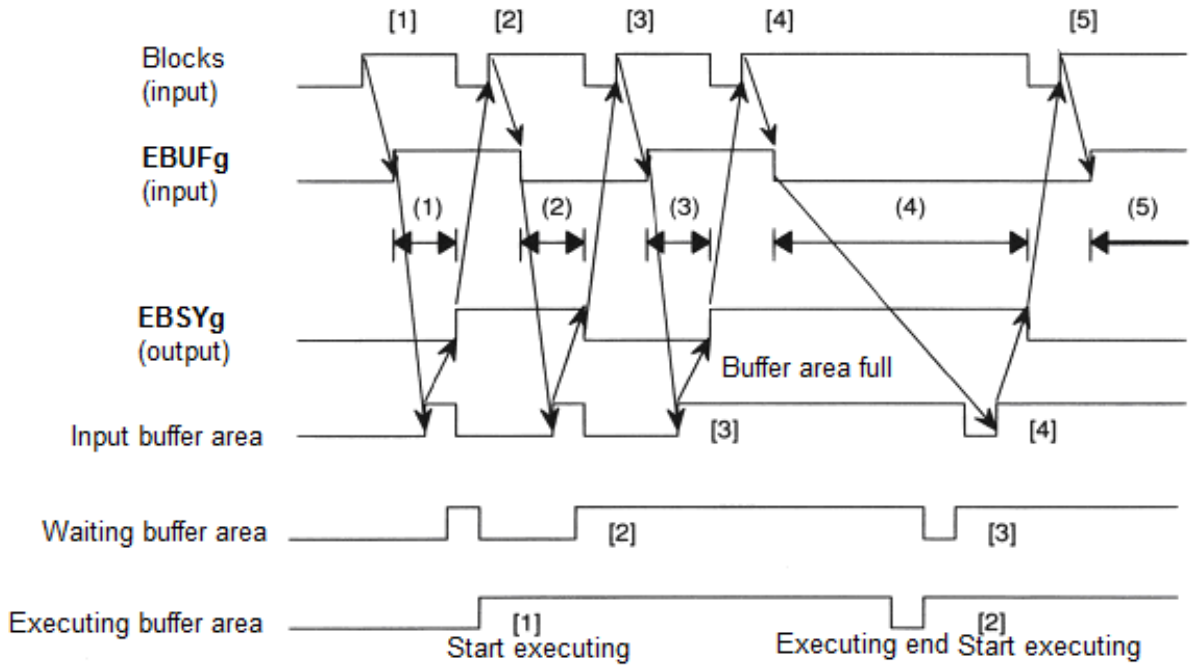
The movements are as below after command [1] execution is completed:

Command [2]: Waiting for the buffer→executing the buffer

Command [3]: Input the buffer→waiting for the buffer

Command [4]: Command the block (the axis control block data signal) → transmission of the input buffer; after the transmission into the command [4] of the input buffer ends, command [5] in CNC (setting the axis control block data signal).

The time chart of the command movement is as below:




(1)，(2)，(3)，(4)，(5)，during the interval, the new block can't be sent (i.e. When EBUFg and EBSYg are in the different logic status, the new block can't be read in).

The buffer area status controlled by PLC axis can be set by OR-AND of the axis control command reading signal EBUFg input by PLC and the axis control command reading finish signal EBSYg output by CNC.

| EBUFg | EBSYg | Add based on bits （X0R） | PLC axis control buffer area status |
|---|---|---|---|
| 0 | 0 | 0 | After the previous block has already been read into the PLC axis control buffer area, PLC can send the next block. |
| 1 | 1 | | |
| 0 | 1 | 1 | When the previous block is being read, the reading or waiting PLC axis control buffer area becomes usable. And PLC doesn't send the next block and doesn't reverse the logic status of EBUFg. If EBUFg status is reversed, the sent block will become invalid. |
| 1 | 0 | | |

Note: Inputting two PLC data must be separated by the period of executing one ladder diagram.

7) Repeatedly execute the steps（3）and（4）till all blocks are sent.

When the last block has been sent, the control signals EAX1～EAX5 are "0". However, before setting these signals as "0", firstly check whether the execution of all programs in PLC axis control buffer area has already been completed or not. When one block is being executed, if the signal is "0", the system alarm occurs (PLC axis is still being operating, the switch can't be executed.).

After comfirming the execution of all programs in PLC axis control buffer area has been completed, check whether the control axis selection status signal *EAXSL is "0" or not. For those axes controlled by PLC, such as: the axes controlling the tool, the switchboard and ATC, the signals EAX1～EAX5 must be always "1" and they are not necessarily set as "0". After the execution of all blocks is completed, PLC axis control will automatically stop executing.

8) When the selection signals EAX1～EAX5 of the controlled axis is changed from “1” to “0”, CNC controls the movement, again.

## 10.4 Function Command

### 10.4.1 List of Commands

The command movements executed by PLC axis control function are shown as the following list; among them, “the command” means the axis control command signals (EC0g～EC6g), decimal system, “data 1” means the axis control feedrate signals (EIF0g～EIF15g), “data 2” means the axis control data signals (EID0g～EID31g).

| Command | Movement | Data 1 | Data 2 | Remark |
|---|---|---|---|---|
| 00 | Rapid traverse | Rapid traverse rate | Movement distance | Execute the movement same with that of “G00” in CNC |
| 01 | Cutting feedrate (per minute) | Cutting feedrate | Movement distance | Execute the movement same with that of “G01 G98 F_” in CNC |
| 02 | Cutting feedrate (per revolution) | Cutting feedrate | Movement distance | Execute the movement same with that of “G01 G99 F_” in CNC |
| 04 | Dwell | --------- | Dwell time | Execute the movement same with that of “G04” in CNC |
| 05 | The reference position return | --------- | --------- | Execute the movement same with the manual reference positon return in CNC |
| 07 | The 1st reference position return | --------- | --------- | Execute the movement same with that of “G28” in CNC |
| 08 | The 2nd reference position return | --------- | --------- | Execute the movement same with that of “G30 P2” in CNC |
| 09 | The 3rd reference position return | --------- | --------- | Execute the movement same with that of “G30 P3” in CNC |
| 10 | The 4th reference position return | --------- | --------- | Execute the movement same with that of “G30 P4” in CNC |
| 18 | Miscellaneous function | --------- | Miscellaneous function code | Execute the movement same with that of the miscellaneous function code in CNC |

**Note 1: When EABSg（G141＃0～＃3） is “1”, “the move distance” is specified by the absolute position; If it is “0”,**

**"the move distance" is specified by the relative positon.**

**Note 2: When 07, 08, 09 or 10 are commanded to return into the reference position, the feedrate is the rapid traverse rate (within the maximum permissible speed of machine zero return) of each axis, and the intemediant position doesn't exist during zero return.**

## 10.4.2 Detailed Explanation of the Command

**1) Rapid traverse（00）**

Execute the movement same with that of "G00" in CNC.

Data of the axis control:

| Signal abbrivation | Signal address (Group A) | Data |
|---|---|---|
| EC0g～EC6g | G143＃0～＃6 | Rapid traverse command |
| EIF0g～EIF15g | G144，G145 | Rapid traverse rate |
| EID0g～EID31g | G146～G149 | Move distance |

Among them:

Feedrate: The feedrate can be set by the status parameter PRPD. When PRPD is set as "1", it is specified by PLC axis feedrate signals EIF0g～EIF15g; When PRPD is set as "0", the feedrate is the rapid traverse rate of each axis set by CNC.

Move distance: When EABSg is "1", the position specified by EID0g～EID31g is absolute one; "0", the position specified by EID0g～EID31g is relative one.

**2) Cutting feedrate（01 / 02）**

Data of axis control

| Signal abbrivation | Signal address (Group A) | Data |
|---|---|---|
| EC0g～EC6g | G143＃0～＃6 | Cutting command (feeding per minute/per revolution) |
| EIF0g～EIF15g | G144，G145 | Cutting feedrate |
| EID0g～EID31g | G146～G149 | Move distance |

Among them:

Cutting command: When the values of EC0g～EC6g are 01, it is the cutting feedrate per minute, and it is equivalent to "G01 G98 F_" executed by CNC;

When the values of EC0g～EC6g are 02, it is the cutting feedrate per revolution, and it is equivalent to "G01 G99 F_" executed by CNC.

Cutting feedrate: During cutting, the feedrate is specified by EIF0g～EIF15g.

Move distance: When EABSg is "1", the position specified by EID0g～EID31g is absolute one; "0", the position specified by EID0g～EID31g is relative one.

**3) Dwell（04）**

Execute the movement same with that of "G04" in CNC.

The axis control data:

| Signal abbrivation | Signal address (Group A) | Data |
|---|---|---|
| EC0g～EC6g | G143＃0～＃6 | Dwell command |
| EID0g～EID31g | G146～G149 | Dwell time |

Among them:

Dwell time: The unit of dwell time is millisecond (ms).

**4) The reference positon return（05）**

The movement is same with the manual reference positon return (mode B) in CNC. The rapid traverse rate, the direction and the zero offset amount of zero return are same with those of the manual reference positon return mode B executed in CNC.

**5) The 1st reference position return (07)**

The 1st reference positon set by the system is returned at the rapid positioning speed (within the zero return high speed). The difference between the 1st reference positon return by the command and that in CNC is that the deceleration and the zero signals aren't detected in the former one (the deceleration and the zero signals should be detected during the 1st reference positon return in CNC.).

During the 1st reference positon return, setting the rapid traverse rate and zero offset is same as setting the parameters in CNC.

**6) The 2nd, 3rd or 4th reference position return (08, 09, 10)**

The 2nd, 3rd or 4th reference positon set by the system is returned at the rapid traverse rate (within the zero return high speed). They are same as each axis returning into the 2nd, 3rd or 4th reference positon.

During the 2nd/3rd/4th reference position return, setting the rapid traverse rate and the zero offset is same as setting the parameter in CNC.

**7) Miscellaneous function（18）**

Execute the movement same as the miscellaneous function in CNC.



## 10.5 Function Signal

### 10.5.1 Control Axis Selection Signal

**EAX1～EAX5（G136＃0～G136＃4）**

- Signal type: PLC－>CNC
- Signal function: When the signal is set as "1", PLC axis control becomes valid; "0", invalid. Only when the control axis selection status signal *EAXSL is "0", the control axis selection signals EAX1～EAX5 can be changed. When *EAXSL is 1, the axis selection signal is selected, the system alarm occurs (PLC axis is working and isn't not allowed to be switched.) The system also alarms when some axis is moving controlled by CNC and if the control selection signal of the axis is set as "1" (When CNC axis is working, it's not allowed to switch.).
- Signal address:

| | #7 | #6 | #5 | #4 | #3 | #2 | #1 | #0 |
|---|---|---|---|---|---|---|---|---|
| G136 | | | | EAX5 | EAX4 | EAX3 | EAX2 | EAX1 |

## 10.5.2 Axis Control Command Signal

**EC0g～EC6g（G143＃0～G143＃6, G155＃0～G155＃6, G167＃0～G167＃6, G179＃0～G179＃6）**

- Signal type: PLC－>CNC
- Signal function: The type of the command for PLC axis control is set; please refer to the relative introduction in chapter 14.
- Signal address:

| | #7 | #6 | #5 | #4 | #3 | #2 | #1 | #0 |
|---|---|---|---|---|---|---|---|---|
| G143 | | EC6A | EC5A | EC4A | EC3A | EC2A | EC1A | EC0A |
| G155 | | EC6B | EC5B | EC4B | EC3B | EC2B | EC1B | EC0B |
| G167 | | EC6C | EC5C | EC4C | EC3C | EC2C | EC1C | EC0C |
| G179 | | EC6D | EC5D | EC4D | EC3D | EC2D | EC1D | EC0D |

## 10.5.3 The Axis Control Feedrate Signal

**EIF0g～EIF15g（G144～G145, G156～G157, G168～G169, G180～G181）**

- Signal type: PLC－>CNC
- Signal function:
  1）Rapid traverse: When the status parameter PRPD（No.185＃1）is set as “1”, the signals EIF0g～EIF15g command the rapid traverse rate in the binary system;
  2）Cutting feed (feeding per minute): Command the feedrate of one axis in the binary system;
  3）Cutting feed (feeding per revolution): Command the tool movement amount per revolution of the spindle in the binary system.
- Signal address:

| | #7 | #6 | #5 | #4 | #3 | #2 | #1 | #0 |
|---|---|---|---|---|---|---|---|---|
| G144 | EIF7A | EIF6A | EIF5A | EIF4A | EIF3A | EIF2A | EIF1A | EIF0A |
| G145 | EIF15A | EIF14A | EIF13A | EIF12A | EIF11A | EIF10A | EIF9A | EIF8A |

| | #7 | #6 | #5 | #4 | #3 | #2 | #1 | #0 |
|---|---|---|---|---|---|---|---|---|
| G156 | EIF7B | EIF6B | EIF5B | EIF4B | EIF3B | EIF2B | EIF1B | EIF0B |
| G157 | EIF15B | EIF14B | EIF13B | EIF12B | EIF11B | EIF10B | EIF9B | EIF8B |

| | #7 | #6 | #5 | #4 | #3 | #2 | #1 | #0 |
|---|---|---|---|---|---|---|---|---|
| G168 | EIF7C | EIF6C | EIF5C | EIF4C | EIF3C | EIF2C | EIF1C | EIF0C |
| G169 | EIF15C | EIF14C | EIF13C | EIF12C | EIF11C | EIF10C | EIF9C | EIF8C |

| | #7 | #6 | #5 | #4 | #3 | #2 | #1 | #0 |
|---|---|---|---|---|---|---|---|---|
| G180 | EIF7D | EIF6D | EIF5D | EIF4D | EIF3D | EIF2D | EIF1D | EIF0D |
| G181 | EIF15D | EIF14D | EIF13D | EIF12D | EIF11D | EIF10D | EIF9D | EIF8D |

## 10.5.4 Axis Control Data Signal

**EID0g～EID31g（G146～G149, G158～G161, G170～G173, G182～G185）**

- Signal type: PLC－>CNC
- Signal function:

1）Rapid traverse;

2）Cutting feed (feeding per minute)；

3）During cutting feed (feeding per revolution) : the coordinates of the end position are specified by these commands EID0g～EID31g in the binary system. However, whether it is the absolute coordinate or the relative coordinate should be set by the status of EABSg signal. When EABSg signal is “1”, it is the absolute coordinate; “0”, the relative coordinate;

4）Dwell: The dwell time is specified by the signal in the binary system, unit: ms;

5）Miscellaneous function: The miscellaneous function codes of PLC axis control are specified by EID0g～EID15g signals in the binary system.

- Signal address:

| | #7 | #6 | #5 | #4 | #3 | #2 | #1 | #0 |
|---|---|---|---|---|---|---|---|---|
| G146 | EID7A | EID6A | EID5A | EID4A | EID3A | EID1A | EID1A | EID0A |
| G147 | EID15A | EID14A | EID13A | EID12A | EID11A | EID10A | EID9A | EID8A |
| G148 | EID23A | EID22A | EID21A | EID20A | EID19A | EID18A | EID17A | EID16A |
| G149 | EID31A | EID30A | EID29A | EID28A | EID27A | EID26A | EID25A | EID24A |

| | #7 | #6 | #5 | #4 | #3 | #2 | #1 | #0 |
|---|---|---|---|---|---|---|---|---|
| G158 | EID7B | EID6B | EID5B | EID4B | EID3B | EID1B | EID1B | EID0B |
| G159 | EID15B | EID14B | EID13B | EID12B | EID11B | EID10B | EID9B | EID8B |
| G160 | EID23B | EID22B | EID21B | EID20B | EID19B | EID18B | EID17B | EID16B |
| G161 | EID31B | EID30B | EID29B | EID28B | EID27B | EID26B | EID25B | EID24B |

| | #7 | #6 | #5 | #4 | #3 | #2 | #1 | #0 |
|---|---|---|---|---|---|---|---|---|
| G170 | EID7C | EID6C | EID5C | EID4C | EID3C | EID1C | EID1C | EID0C |
| G171 | EID15C | EID14C | EID13C | EID12C | EID11C | EID10C | EID9C | EID8C |
| G172 | EID23C | EID22C | EID21C | EID20C | EID19C | EID18C | EID17C | EID16C |
| G173 | EID31C | EID30C | EID29C | EID28C | EID27C | EID26C | EID25C | EID24C |

| | #7 | #6 | #5 | #4 | #3 | #2 | #1 | #0 |
|---|---|---|---|---|---|---|---|---|
| G182 | EID7D | EID6D | EID5D | EID4D | EID3D | EID1D | EID1D | EID0D |
| G183 | EID15D | EID14D | EID13D | EID12D | EID11D | EID10D | EID9D | EID8D |
| G184 | EID23D | EID22D | EID21D | EID20D | EID19D | EID18D | EID17D | EID16D |
| G185 | EID31D | EID30D | EID29D | EID28D | EID27D | EID26D | EID25D | EID24D |

## 10.5.5 Control Axis Coordinate Selection Signal

**EABSg（G141＃0～G141＃3）**

- Signal type: PLC－>CNC
- Signal function: When the signal is set as “1”, the coordinate input by EID0g～EID31g is the absolute one; “0”, the relative coordinate.
- Signal address:

| | #7 | #6 | #5 | #4 | #3 | #2 | #1 | #0 |
|---|---|---|---|---|---|---|---|---|
| G141 | | | | | EABSD | EABSC | EABSB | EABSA |

## 10.5.6 Axis Control Command Reading Signal

**EBUFg（G142＃7, G154＃7, G166＃7, G178＃7）**

- Signal type: PLC－>CNC
- Signal function: Commanding PLC axis control unit reading is for the command data block controlled by PLC axis. The signal is changed from “0” to “1” or “1” to “0”; please refer to “basic steps” for the detailed running situation.
- Signal address:

| | #7 | #6 | #5 | #4 | #3 | #2 | #1 | #0 |
|---|---|---|---|---|---|---|---|---|
| G141 | EBUFA | | | | | | | |
| G154 | EBUFB | | | | | | | |
| G166 | EBUFC | | | | | | | |
| G178 | EBUFD | | | | | | | |

## 10.5.7 Axis Control Command Reading Finish Signal

**EBSYg（F130＃7, F133＃7, F136＃7, F139＃7）**

- Signal type: CNC－>PLC
- Signal function: The signal is sent to inform the system that PLC axis control unit has already read one command data block controlled by PLC and the block has been saved in the buffer area. Please refer to “basic steps” for the detailed running situation.
- Signal address:

| | #7 | #6 | #5 | #4 | #3 | #2 | #1 | #0 |
|---|---|---|---|---|---|---|---|---|
| F130 | EBSYA | | | | | | | |
| F133 | EBSYB | | | | | | | |
| F136 | EBSYC | | | | | | | |
| F139 | EBSYD | | | | | | | |

## 10.5.8 Resetting Signal

**ECLRg（G142＃6, G154＃6, G166＃6, G178＃6）**

- Signal type: PLC－>CNC
- Signal function: Reset the corresponding PLC control axis.

When the signal is set as “1”, the following operation is executed:

1）When the tool is being move along the axis, the deceleration is executed and the tool is stopped;

2）When the tool is dwelled, the operation stops;

3）When the miscellaneous function is being executed, the operation stops.

Meanwhile, all buffer commands are cleared. When the signal is set as “1”, any control commands are ignored.

- Signal address:

| | #7 | #6 | #5 | #4 | #3 | #2 | #1 | #0 |
|---|---|---|---|---|---|---|---|---|
| G142 | | ECLRA | | | | | | |
| G154 | | ECLRB | | | | | | |
| G166 | | ECLRC | | | | | | |
| G178 | | ECLRD | | | | | | |

## 10.5.9 Axis Control Dwell Signal

**ESTPg（G142＃5, G154＃5, G166＃5, G178＃5）**

- Signal type: PLC－>CNC
- Signal function: When the signal is “1”, the following operation is executed:

  1）When the tool is being moved along the axis, the deceleration is executed and the tool is stopped;

  2）When the tool is dwelled, the operation stops;

  3）The operation stops when the miscellaneous function is being executed and the miscellaneous function finish signal EFINg is input. When the signal is set as “0”, the operation is restarted.

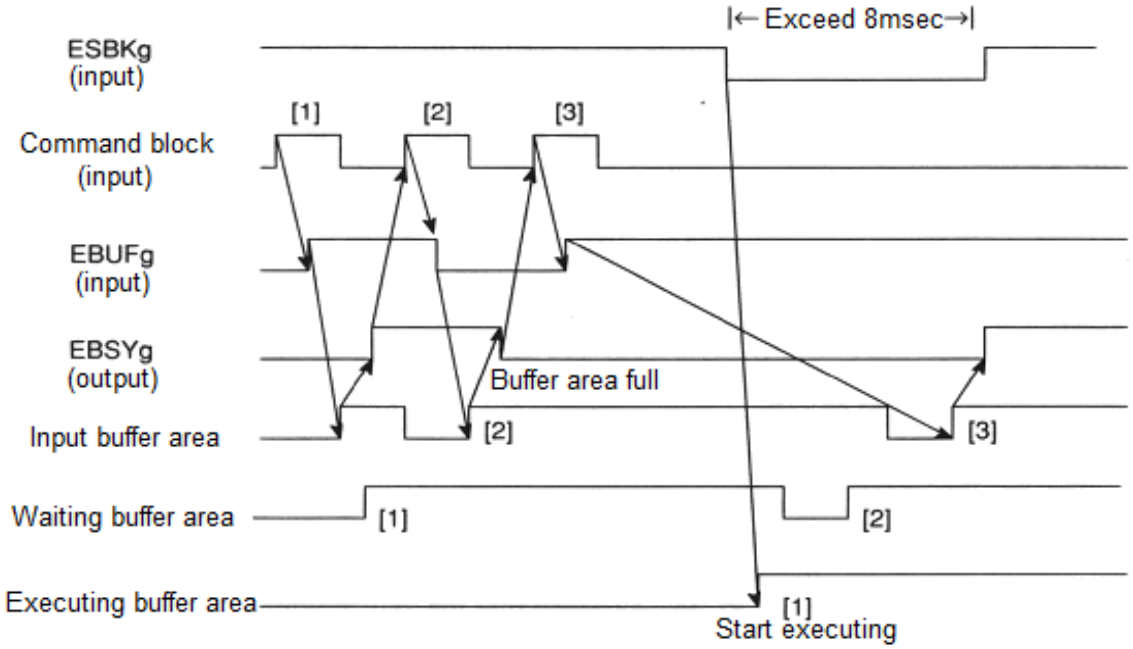
- Signal address:

| | #7 | #6 | #5 | #4 | #3 | #2 | #1 | #0 |
|---|---|---|---|---|---|---|---|---|
| G142 | | | ESTPA | | | | | |
| G154 | | | ESTPB | | | | | |
| G166 | | | ESTPC | | | | | |
| G178 | | | ESTPD | | | | | |

## 10.5.10 Program Stop Signal and Program Stop Invalid Signal

**ESBKg（G142＃3, G154＃3, G166＃3, G178＃3）, EMSBKg（G143＃7, G155＃7, G167＃7, G179＃7）**

- Signal type: PLC－>CNC
- Signal function: During executing the command sent by PLC, when the block stop signal ESBKg is “1”, the axis control stops after the execution of the block is completed. When the signal is set as “0”, the buffer command is executed. In the block, when the block stop invalid signal EMSBKg is set as “1”, the block stop signal ESBKg is invalid.
- Signal time sequence:




- Signal address:

| | #7 | #6 | #5 | #4 | #3 | #2 | #1 | #0 |
|---|---|---|---|---|---|---|---|---|
| G142 | | | | | ESBKA | | | |
| G154 | | | | | ESBKB | | | |
| G166 | | | | | ESBKC | | | |
| G178 | | | | | ESBKD | | | |

| | #7 | #6 | #5 | #4 | #3 | #2 | #1 | #0 |
|---|---|---|---|---|---|---|---|---|
| G143 | EMSBKA | | | | | | | |
| G155 | EMSBKB | | | | | | | |
| G167 | EMSBKC | | | | | | | |
| G179 | EMSBKD | | | | | | | |

## 10.5.11 The Miscellaneous Function Signal

**Miscellaneous function codes signal EM11g～EM48g（F132～F142, F135～F145, F138～F148, F141～F151）**

- Signal type: CNC－>PLC

**The miscellaneous function strobe signal EMFg（F131＃0, F134＃0, F137＃0, F140＃0）**

- Signal type: CNC－>PLC

**The miscellaneous function finish signal EMFINg（G142＃0, G154＃0, G166＃0, G178＃0）**

- Signal type: PLC－>CNC
- Signal function: PLC axis control unit sends the specified miscellaneous function codes in the program into the miscellaneous function code signals EM11g～EM48g and waits for the miscellaneous function finish signal EFINg. When the miscellaneous function finish signal EFINg returns, PLC axis control unit starts to execute the next block. The time

sequence of sending the miscellaneous function code signals and the strobe signals and receiving the miscellaneous function finish signal is same as that of the miscellaneous function (M function) controlled by CNC. About the details, please refer to “miscellaneous function execution signal”.

- Signal address:

| | #7 | #6 | #5 | #4 | #3 | #2 | #1 | #0 |
|---|---|---|---|---|---|---|---|---|
| F132 | EM28A | EM24A | EM22A | EM21A | EM18A | EM14A | EM12A | EM11A |
| F142 | EM48A | EM44A | EM42A | EM41A | EM38A | EM34A | EM32A | EM31A |
| F135 | EM28B | EM24B | EM22B | EM21B | EM18B | EM14B | EM12B | EM11B |
| F145 | EM48B | EM44B | EM42B | EM41B | EM38B | EM34B | EM32B | EM31B |
| F138 | EM28C | EM24C | EM22C | EM21C | EM18C | EM14C | EM12C | EM11C |
| F148 | EM48C | EM44C | EM42C | EM41C | EM38C | EM34C | EM32C | EM31C |
| F141 | EM28D | EM24D | EM22D | EM21D | EM18D | EM14D | EM12D | EM11D |
| F151 | EM48D | EM44D | EM42D | EM41D | EM38D | EM34D | EM32D | EM31D |

| | #7 | #6 | #5 | #4 | #3 | #2 | #1 | #0 |
|---|---|---|---|---|---|---|---|---|
| F131 | | | | | | | | EMFA |
| F134 | | | | | | | | EMFB |
| F137 | | | | | | | | EMFC |
| F140 | | | | | | | | EMFD |

| | #7 | #6 | #5 | #4 | #3 | #2 | #1 | #0 |
|---|---|---|---|---|---|---|---|---|
| G142 | | | | | | | | EMFINA |
| G154 | | | | | | | | EMFINB |
| G166 | | | | | | | | EMFINC |
| G178 | | | | | | | | EMFIND |

## 10.5.12 The Servo Cut-off Signal

**ESOFg（G142＃4，G154＃4，G166＃4，G178＃4）**

- Signal type：PLC－>CNC
- Signal function: When the signal is set as 1, the enable of the servo motor of the corresponding PLC control axis is cut off. When the signal is set as 0, the enable of the servo motor is connected.

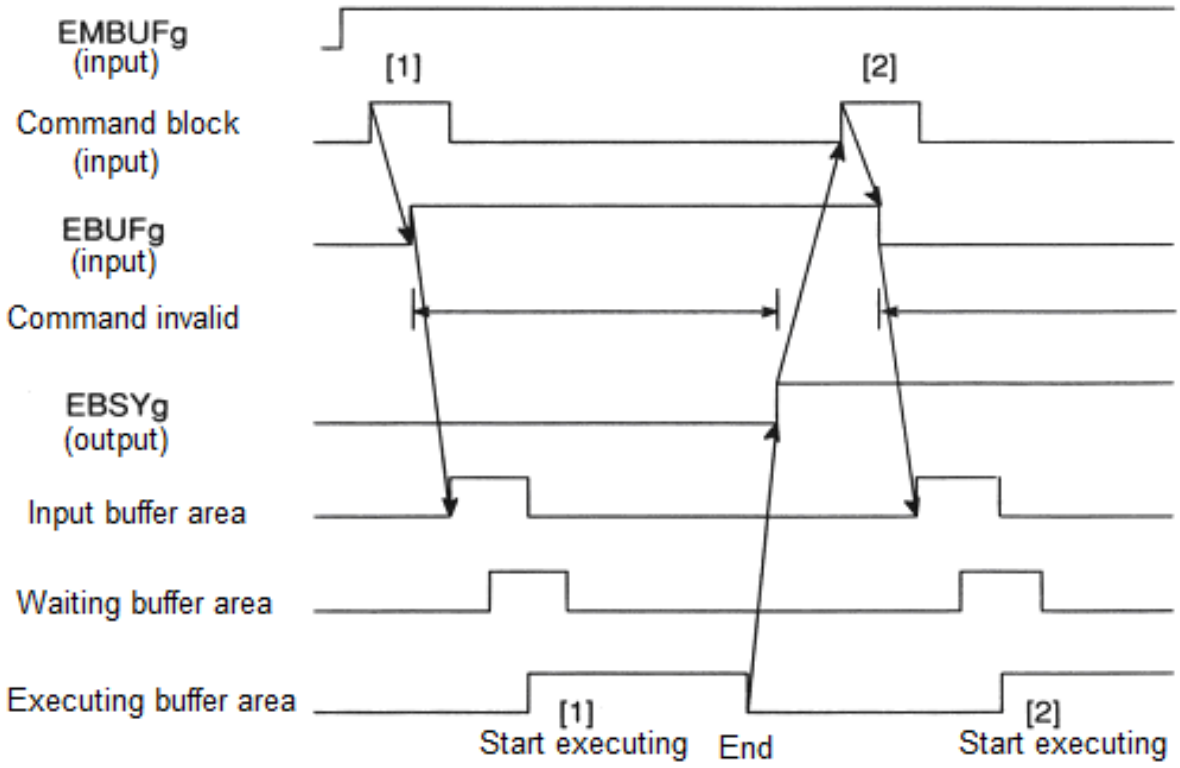
- Signal address:

| | #7 | #6 | #5 | #4 | #3 | #2 | #1 | #0 |
|---|---|---|---|---|---|---|---|---|
| G142 | | | | ESOFA | | | | |
| G154 | | | | ESOFB | | | | |
| G166 | | | | ESOFC | | | | |
| G178 | | | | ESOFD | | | | |

## 10.5.13 Buffer Invalid Signal

**EMBUFg（G142＃2, G154＃2, G166＃2, G178＃2）**

- Singal type: PLC－>CNC
- Signal function: When the signal is set as “1”, the waiting buffer and the input buffer areas are invalid, and the new block can’t be saved. The new control command can be read-in only when the execution buffer includes one block and all buffer areas are idle.
- Signal time sequence:




- Signal address:

| | #7 | #6 | #5 | #4 | #3 | #2 | #1 | #0 |
|---|---|---|---|---|---|---|---|---|
| G142 | | | | | | EMBUFA | | |
| G154 | | | | | | EMBUFB | | |
| G166 | | | | | | EMBUFC | | |
| G178 | | | | | | EMBUFD | | |

## 10.5.14 The Selection Status Signal of the Control Axis

***EAXSL（F129＃7）**

- Signal type: CNC－>PLC
- Signal function: When the signal is “0”, the control axis selection signals EAX1～EAX5 can be changed.

  The signal is “1” in the following situations:

  1）When the tool is being moved along PLC control axis;

  2）When PLC axis control unit buffer area isn’t idle;

  3）When the servo cut-off signal ESOFg is set as “1”;

  4）When the signal is “1”, the axis selection signals EAX1～EAX5 are changed, which will cause the system alarm (PLC axis is working and it’s not allowed to switch).
- Signal address:

| | #7 | #6 | #5 | #4 | #3 | #2 | #1 | #0 |
|---|---|---|---|---|---|---|---|---|
| F129 | *EAXSL | | | | | | | |

## 10.5.15 Axis Movement Signal

**EGENg（F130＃4, F133＃4, F136＃4, F139＃4）**

- Signal type: CNC－>PLC
- Signal function: The signal is “1” when the tool is moved along PLC control axis based on the command.
- Signal address:

| | #7 | #6 | #5 | #4 | #3 | #2 | #1 | #0 |
|---|---|---|---|---|---|---|---|---|
| F130 | | | | EGENA | | | | |
| F133 | | | | EGENB | | | | |
| F136 | | | | EGENC | | | | |
| F139 | | | | EGEND | | | | |

## 10.5.16 The Miscellaneous Function Execution Signal

**EDENg（F130＃3, F133＃3, F136＃3, F139＃3）**

- Signal type: CNC－>PLC
- Signal function: The signal is “1”from sending the miscellaneous function code signals EM11g～EM48g to the miscellaneous function finish signal EFINg return.
- Signal address:

| | #7 | #6 | #5 | #4 | #3 | #2 | #1 | #0 |
|---|---|---|---|---|---|---|---|---|
| F130 | | | | | EDENA | | | |
| F133 | | | | | EDENB | | | |
| F136 | | | | | EDENC | | | |
| F139 | | | | | EDEND | | | |

## 10.5.17 Feedrate Override Signal

***FV0E～*FV7E（G151）**

- Signal type：PLC－>CNC
- Signal function: Like the feedrate override signals *FV0～*FV7 of CNC, these signals are for selecting the override of the cutting feedrate. Totol eight cutting feedrate override signals in the binary system are corresponded to the following override values.
- Command movement: During the tool cutting, the actual feedrate is obtained by the feedrate specified by the cutting feed multiplies the override value selected by these signals:

| ***FV7E～*FV0E(G151＃7～G151＃0)** | **Cutting feedrate override** |
|---|---|
| 00001111 | 0% |
| 00001110 | 10% |
| 00001101 | 20% |
| 00001100 | 30% |
| 00001011 | 40% |
| 00001010 | 50% |
| 00001001 | 60% |
| 00001000 | 70% |
| 00000111 | 80% |
| 00000110 | 90% |
| 00000101 | 100% |
| 00000100 | 110% |
| 00000011 | 120% |
| 00000010 | 130% |
| 00000001 | 140% |
| 00000000 | 150% |

- Signal address:

| | **#7** | **#6** | **#5** | **#4** | **#3** | **#2** | **#1** | **#0** |
|---|---|---|---|---|---|---|---|---|
| **G151** | ***FV7E** | ***FV6E** | ***FV5E** | ***FV4E** | ***FV3E** | ***FV2E** | ***FV1E** | ***FV0E** |

## 10.5.18 The Rapid Traverse Rate Override Signal

**ROV1E，ROV2E（G150＃0，＃1）**

- Signal type: PLC－>CNC
- Signal function: It is the rapid traverse rate override signal.
- Comamnd movement: The encoder signals are corresponding to the following overrides.

| ROV1E | ROV2E | Override value |
|---|---|---|
| 0 | 0 | 100% |
| 0 | 1 | 50% |
| 1 | 0 | 25% |
| 1 | 1 | FO% |

- Signal address:

| | #7 | #6 | #5 | #4 | #3 | #2 | #1 | #0 |
|---|---|---|---|---|---|---|---|---|
| G150 | | | | | | | ROV2E | ROV1E |

## 10.5.19 Buffer Area Full Signal

**EABUFg（F131＃1, F134＃1, F137＃1, F140＃1）**

- Signal type: CNC－>PLC
- Signal function: The signal is “1” when the buffer area is full of the programs.
- Signal address:

| | #7 | #6 | #5 | #4 | #3 | #2 | #1 | #0 |
|---|---|---|---|---|---|---|---|---|
| F131 | | | | | | | EABUFA | |
| F134 | | | | | | | EABUFB | |
| F137 | | | | | | | EABUFC | |
| F140 | | | | | | | EABUFD | |

## 10.5.20 Control signal

**EACNT1～EACNT5（F182＃0～F182＃4）**

- Signal type：CNC－>PLC
- Signal function: After some axis has already been switched into PLC one, the control signal EACNTn of the axis is changed into “1”; when some axis has already been switched into CNC one, the control signal of the axis EACNTn is changed into “0”.

- Signal address:

| | #7 | #6 | #5 | #4 | #3 | #2 | #1 | #0 |
|---|---|---|---|---|---|---|---|---|
| F182 | | | | EACNT5 | EACNT4 | EACNT3 | EACNT2 | EACNT1 |

# CHAPTER ELEVEN DUAL-PATH CONTROL FUNCTION

## 11.1 Brief

The described content in this chapter is suitable for dual-channel control system GSK980TTC system in the CNC control system of the GSK980D. GSK980TTC is the dual-channel control system can not only own the foundamental functions of the standard system, but also offer the special functions; refer to the following items:

1) Channel selection and display
2) Interchannel common memory
3) Wait for M code function
4) Interchannel spindle control function
5) Interchannel interference check function
6) Balanced cutting
7) Synchronous control
8) Mix control
9) Overlap control
10) Imaginary axis control
11) Interchannel single block check function

In the subsequent description, the G, F signals related with the above-mentioned functions will mainly describe.



## 11.2 Channel Selection & Display

For all channels, there is only one set of LCD, keyboard, machine panel. Display and setting of all kinds of data, program input in MDI mode, program edit in program memory of each channel can be switched by channel selection signals.

**Channel selection signal HEAD（G063＃0）**

- Signal type:PLC－>CNC
- Signal function: Channel chage signal. “0” means channel 1; “1” means channel 2.
- Signal address:

| | #7 | #6 | #5 | #4 | #3 | #2 | #1 | #0 |
|---|---|---|---|---|---|---|---|---|
| G063 | | | | | | | | HEAD |

## 11.3 Wait for M Code Function

M code controls wait in the course of processing of Channel 1 and Channel 2. M code used to wait in some channel is executed when automatic run, the system does not pause the next execution till the same M code is executed in other channel, and then the system starts to execute the next block.

### 11.3.1 Wait Ignorance Signal

**NOWT（G063＃1）**（channel's common signal interface）

**NMWT（Gn063＃7）**（channel's alone signal interface）

- Signal type:PLC－>CNC
- Signal function: specify whether to wait based on M code.wait not be based on M code when the signal is "1" . ignore the commanded wait M code in the course of machining.
  Execute the wait based on M code path when the signal is '0'. When the wait M code is commanded in some path, other paths are waited and are commanded by the same M code, and then the next block is executed.
- Signal address:

| | #7 | #6 | #5 | #4 | #3 | #2 | #1 | #0 |
|---|---|---|---|---|---|---|---|---|
| G0063 | NMWT | | | | | | NOWT | |
| G1063 | NMWT | | | | | | | |

### 11.3.2 Waiting Signal

**WATO（Fn063＃6）**

- Signal type:CNC－>PLC
- Signal function: the signal informs that PLC's each channel is based on M code wait. when interchannel is waiting, i.e. after the wait M code is commanded, during the same M code being specified at another channel, the signal is "1". The signal is "0" when interchannel is not waited.
- Signal address:

| | #7 | #6 | #5 | #4 | #3 | #2 | #1 | #0 |
|---|---|---|---|---|---|---|---|---|
| F0063 | | WATO | | | | | | |
| F1063 | | WATO | | | | | | |

## 11.4 Interchannel Spindle Control Function

Interchannel spindle control function can make the spindle command in some channel control a spindle in another channel, or make some channel use the spindle encoder feedback signal of another channel, which ensure two tool posts machine the workpiece mounted on one spindle.

### 11.4.1 Interchannel Spindle Command Selection Signal

**SLSPA（G0063＃2），SLSPB（G0063＃3）**

- Signal type:PLC－>CNC
- Signal function: Specify the spindle's control channel of each channel.

| Signal input SLSPA<G0063.2> | Actual control channel of spindle belonging to Channel 1 |
|---|---|
| 0 | Channel 1 |
| 1 | Channel 2 |

| Signal input SLSPB<G0063.3> | Actual control channel of spindle belonging to Channel 2 |
|---|---|
| 0 | Channel 2 |
| 1 | Channel 1 |

- Signal address:

| | #7 | #6 | #5 | #4 | #3 | #2 | #1 | #0 |
|---|---|---|---|---|---|---|---|---|
| G0063 | | | | | SLSPB | SLSPA | | |

### 11.4.2 Interchannel Spindle Feedback Selection Signal

**SLPCA（G0064＃2），SLPCB（G0064＃3）**

- Signal type:PLC－>CNC
- Signal function: specify the channel of position encoder feedback pulse read by each channel.
- Signal address:

| Signal input SLPCA<G0064.2> | Channel belonged to reading position encoder pulse in Channel 1 |
|---|---|
| 0 | Channel 1 |
| 1 | Channel 2 |

| Signal input SLPCB<G0064.3> | Channel belonged to reading position encoder pulse in Channel 2 |
|---|---|
| 0 | Channel 2 |
| 1 | Channel 1 |

| | #7 | #6 | #5 | #4 | #3 | #2 | #1 | #0 |
|---|---|---|---|---|---|---|---|---|
| G0064 | | | | | SLPCB | SLPCA | | |

### 11.4.3 Interchannel Spindle Command Confirmation Signal

**COSP（F0064＃5）**

- Signal type:CNC－>PLC
- Signal function: the signal informs the channel of the last executed spindle command. the signal is "1" when the spindle command is executed at side of Channel 2.
  the signal is "0" when the spindle command is executed at side of Channel 1 or through none of channel.
- Signal address:

| | #7 | #6 | #5 | #4 | #3 | #2 | #1 | #0 |
|---|---|---|---|---|---|---|---|---|
| F0064 | | | COSP | | | | | |

**Note: when the spindle belonging to some channel is controlled by the spindle command from Channel 1 or 2, the signal can confirm the last executed spindle command's channel.**

## 11.5 Interchannel interference check

Some time, a machine with a dual channel control system needs the tool post controlled by each channel to machine the same workpiece, at the moment, the two tool posts approach very near, so, mistaken programs or other mistaken setting may cause the tool be damaged or other accidences because of tools touching each other. Interchannel tool path interference check function can make tools stop running before touching each, which can avoid the unnecessary losses.

### 11.5.1 Interchannel Interference Checking Signal

**TICHK（F0064＃6）**

- Signal type:CNC－>PLC
- Signal function: the signal informs whether PLC is performing the interchannel interference check when all required conditions for the interchannel interference check on the tool

posts in Channel 1 and in Channel 2 are provided, the signal becomes ‘1’, otherwise, it does ‘0’.

- Signal address:

| | #7 | #6 | #5 | #4 | #3 | #2 | #1 | #0 |
|---|---|---|---|---|---|---|---|---|
| F0064 | | TICHK | | | | | | |

## 11.5.2 Interchannel Interference Alarm Signal

**TIALM（F0064＃7）**

- Signal type:CNC－>PLC
- Signal function: the signal informs PLC interchannel interference check alarm when the interchannel interference check on the tool posts in Channel 1 and in Channel 2 can judge that the two tool posts interfere each other, the signal becomes ‘1’; when the interchannel interference check on the tool posts in Channel 1 and in Channel 2 can judge that the two tool posts do not execute interference or the interchannel interference check is not executed (i.e. when the interchannel interference signal TICHK is ‘0’), the signal becomes ‘0’.
- Signal address:

| | #7 | #6 | #5 | #4 | #3 | #2 | #1 | #0 |
|---|---|---|---|---|---|---|---|---|
| F0064 | TIALM | | | | | | | |

# 11.6 Synchronous control

The synchronous control function makes a axis belonging to other channel or in the same channel with some axis move synchronously.

## 11.6.1 Synchronous Control Axis Selection Signal

**SYNC1～SYNC5（Gn138＃0～Gn138＃4）**

- Signal type:PLC－>CNC
- Signal function: perform the synchronous control. when the signal becomes ‘1’ from ‘0’, the corresponding axis as a slave control axis starts the synchronous control. When it becomes ‘0’ from ‘1’, the synchronous control is released.
- Signal address:

| | #7 | #6 | #5 | #4 | #3 | #2 | #1 | #0 |
|---|---|---|---|---|---|---|---|---|
| G0138 | | | | SYNC5 | SYNC4 | SYNC3 | SYNC2 | SYNC1 |
| G1138 | | | | SYNC5 | SYNC4 | SYNC3 | SYNC2 | SYNC1 |

## 11.6.2 Parking Signal

**PK1～PK5（Gn122＃0～Gn122＃4）**

- Signal type:PLC－>CNC
- Signal function: place each axis in parking state and make each axis not to move when the signal is ‘1’ in synchronous control, the corresponding axis is set to parking state. The signal is invalid in non synchronous control.
- Signal address:

| | #7 | #6 | #5 | #4 | #3 | #2 | #1 | #0 |
|---|---|---|---|---|---|---|---|---|
| G0122 | | | | PK5 | PK4 | PK3 | PK2 | PK1 |
| G1122 | | | | PK5 | PK4 | PK3 | PK2 | PK1 |

## 11.6.3 Synchronous Main Control Axis Confirmation Signal

**SYCM1～SYCM5（Fn341＃0～Fn341＃4）**

- Signal type:CNC－>PLC
- Signal function: the signal informs each axis is the synchronous main control axis. it is ‘1’ when the corresponding axis is the synchronous main control axis, and it is ‘0’ when the corresponding axis is released by synchronous slave control.
- Signal address:

| | #7 | #6 | #5 | #4 | #3 | #2 | #1 | #0 |
|---|---|---|---|---|---|---|---|---|
| F0341 | | | | SYCM5 | SYCM4 | SYCM3 | SYCM2 | SYCM1 |
| F1341 | | | | SYCM5 | SYCM4 | SYCM3 | SYCM2 | SYCM1 |

## 11.6.4 Synchronous Slave Control Axis Confirmation Signal

**SYCS1～SYCS5（Fn342＃0～Fn342＃4）**

- Signal type:CNC－>PLC
- Signal function: the signal informs each axis is the synchronous slave axis. it is ‘1’ when the corresponding axis is the synchronous slave control axis, and it is ‘0’ when the corresponding axis is released by synchronous slave control.
- Signal address:

| | #7 | #6 | #5 | #4 | #3 | #2 | #1 | #0 |
|---|---|---|---|---|---|---|---|---|
| F0341 | | | | SYCS5 | SYCS4 | SYCS3 | SYCS2 | SYCS1 |
| F1341 | | | | SYCS5 | SYCS4 | SYCS3 | SYCS2 | SYCS1 |

## 11.6.5 Parking Axis Confirmation Signal

**SMPK1～SMPK5（Fn346＃0～Fn346＃4）**

- Signal type:CNC－>PLC

- Signal function: the signal informs each axis is the parking axis of synchronous control. it is '1' when the corresponding axis is a parking axis of synchronous control, and it is '0' when the corresponding axis is released by synchronous slave control or parking is released.
- Signal address:

| | #7 | #6 | #5 | #4 | #3 | #2 | #1 | #0 |
|---|---|---|---|---|---|---|---|---|
| F0346 | | | | SMPK5 | SMPK4 | SMPK3 | SMPK2 | SMPK1 |
| F1346 | | | | SMPK5 | SMPK4 | SMPK3 | SMPK2 | SMPK1 |

### 11.6.5 Synchronous/Mix/Overlap Controlling Signal

**SYN1O～SYN5O（Fn118＃0～Fn118＃4）**

- Signal type:CNC－>PLC
- Signal function: the signal informs each axis is in synchronous/mix/overlap control. it is '1' when the corresponding axis is in synchronous/mix/overlap control and it is '0' when it is not in synchronous/mix/overlap control.
- Signal address:

| | #7 | #6 | #5 | #4 | #3 | #2 | #1 | #0 |
|---|---|---|---|---|---|---|---|---|
| F0118 | | | | SYN5O | SYN4O | SYN3O | SYN2O | SYN1O |
| F1118 | | | | SYN5O | SYN4O | SYN3O | SYN2O | SYN1O |

## 11.7 Mix Control

Mix control function can exchange interchannel any axes to realize interchannel axis' movement control.

### 11.7.1 Mix Control Axis Selection Signal

**MIX1～MIX5（Gn128＃0～Gn128＃4）**

- Signal type:PLC－>CNC
- Signal function: perform mix control.the corresponding axis' mix control is started when the signal becomes '1' from '0', the corresponding axis' mix control is released when the signal becomes '0' from '1'.
- Signal address:

| | #7 | #6 | #5 | #4 | #3 | #2 | #1 | #0 |
|---|---|---|---|---|---|---|---|---|
| G0128 | | | | MIX5 | MIX4 | MIX3 | MIX2 | MIX1 |
| G1128 | | | | MIX5 | MIX4 | MIX3 | MIX2 | MIX1 |

### 11.7.2 Mix Axis Confirmation Signal

**MIXO1～MIXO5（Fn343＃0～Fn343＃4）**

- Signal type:PLC－>CNC
- Signal function: the signal informs each axis is a mix control axis. it is '1' when the

corresponding axis is the mix control axis, it is '0' when the corresponding axis is released by the mix control .

- Signal address:

| | #7 | #6 | #5 | #4 | #3 | #2 | #1 | #0 |
|---|---|---|---|---|---|---|---|---|
| F0343 | | | | MIXO5 | MIXO4 | MIXO3 | MIXO2 | MIXO1 |
| F1343 | | | | MIXO5 | MIXO4 | MIXO3 | MIXO2 | MIXO1 |

## 11.8 Overlap Control

The overlap control is that an axis (overlap slave control axis) executing movement command in movement amount adds other axis(overlap main control axis)' movement amount. Similar to other synchronous control, the overlap control can send movement command to the salve control axis. Motion of slave control axis consists of its own command motion and main control axis motion.



### 11.8.1 Overlap Control Axis Selection Signal

**OVLS1～OVLS5（Gn190＃0～Gn190＃4）**

- Signal type:PLC－>CNC
- Signal function: perform the overlap control. when the signal becomes '1' from '0', the corresponding axis as a slave control axis starts the overlap control. When it becomes '0' from '1', the overlap control is released.
- Signal address:

| | #7 | #6 | #5 | #4 | #3 | #2 | #1 | #0 |
|---|---|---|---|---|---|---|---|---|
| F0190 | | | | OVLS5 | OVLS4 | OVLS3 | OVLS2 | OVLS1 |
| F1190 | | | | OVLS5 | OVLS4 | OVLS3 | OVLS2 | OVLS1 |

### 11.8.2 Overlap Main control Axis Confirmation Signal

**OVMO1～OVMO5（Fn344＃0～Fn344＃4）**

- Signal type:CNC－>PLC
- Signal function: the signal informs each axis is the overlap main control axis. it is '1' when the corresponding axis is the overlap main control axis, and it is '0' when the corresponding axis is released by overlap slave control.
- Signal address:

| | #7 | #6 | #5 | #4 | #3 | #2 | #1 | #0 |
|---|---|---|---|---|---|---|---|---|
| F0344 | | | | OVMO5 | OVMO4 | OVMO3 | OVMO2 | OVMO1 |
| F1344 | | | | OVMO5 | OVMO4 | OVMO3 | OVMO2 | OVMO1 |

### 11.8.3 Overlap Slave Control Axis Confirmation Signal

**OVSO1～OVSO5（Fn345＃0～Fn345＃4）**

- Signal type:CNC－>PLC
- Signal function: the signal informs each axis is the overlap slave axis. it is '1' when the

corresponding axis is the overlap slave control axis, and it is '0' when the corresponding axis is released by overlap slave control.

- Signal address:

| | #7 | #6 | #5 | #4 | #3 | #2 | #1 | #0 |
|---|---|---|---|---|---|---|---|---|
| F0345 | | | | OVSO5 | OVSO4 | OVSO3 | OVSO2 | OVSO1 |
| F1345 | | | | OVSO5 | OVSO4 | OVSO3 | OVSO2 | OVSO1 |

## 11.8.4 Synchronous/Mix/overlap Controlling Signal

**SYN1O～SYN5O（Fn118＃0～Fn118＃4）**

- Signal type:CNC－>PLC
- Signal function: the signal informs each axis is in synchronous/mix/overlap control. it is '1' when the corresponding axis is in synchronous/mix/overlap control and it is '0' when it is not in synchronous/mix/overlap control.
- Signal address:

| | #7 | #6 | #5 | #4 | #3 | #2 | #1 | #0 |
|---|---|---|---|---|---|---|---|---|
| F0118 | | | | SYN5O | SYN4O | SYN3O | SYN2O | SYN1O |
| F1118 | | | | SYN5O | SYN4O | SYN3O | SYN2O | SYN1O |

# CHAPTER TWELEVE OTHER FUNCTIONS

## 12.1 Metric/Inch Conversion

**Inch input signal INCH（Fn002＃0）**

- Signal type: CNC－>PLC
- Signal function: When INCH=1, inch input mode (G20) is adopt; INCH=0, metric input mode (G21) is adopted.
- Signal address:

| | #7 | #6 | #5 | #4 | #3 | #2 | #1 | #0 |
|---|---|---|---|---|---|---|---|---|
| Fn002 | | | | | | | | INCH |

## 12.2 Memory protection

**Memory protection signal KEY1（Gn046＃3）**

- Signal type: PLC－>CNC
- Signal function: Set whether the memory data can be altered.

  When the KEY sets to “1”, the memory data can be altered; when the KEY sets to “0”, the memory data can not be modified.
- Signal address:

| | #7 | #6 | #5 | #4 | #3 | #2 | #1 | #0 |
|---|---|---|---|---|---|---|---|---|
| Gn046 | | | | | KEY1 | | | |

## 12.3 User Macro Input/Output

### 12.3.1 User Macro Input Signal

**UI000～UI015（Gn054、Gn055）**

- Signal type: PLC－>CNC
- Signal function: Without function for the controllable unit. These signals, are treated as one of the CNC macro-variable, are read by macro program, and regarded as the interface signal between the macro program and PLC.
- Signal address: The corresponding macro-variable table is shown below:

| | #7 | #6 | #5 | #4 | #3 | #2 | #1 | #0 |
|---|---|---|---|---|---|---|---|---|
| Gn054 | UI07 | UI06 | UI05 | UI04 | UI03 | UI02 | UI01 | UI00 |
| Variable | #1007 | #1006 | #1005 | #1004 | #1003 | #1002 | #1001 | #1000 |

| | #7 | #6 | #5 | #4 | #3 | #2 | #1 | #0 |
|---|---|---|---|---|---|---|---|---|
| Gn055 | UI15 | UI14 | UI13 | UI12 | UI11 | UI10 | UI09 | UI08 |
| Variable | #1015 | #1014 | #1013 | #1012 | #1011 | #1010 | #1009 | #1008 |

## 12.3.2 User Macro Output Signal

**UO000～UO015（Fn054～Fn055）、UO100～UO131（Fn056～Fn059）：**

- Signal type: CNC－>PLC
- Signal function: Without function for the controllable unit. These signals, are treated as one of the CNC variable read or written by macro program, and regarded as the interface signal between the macro program and PLC.
- Signal address: The corresponding CNC macro-variable table is shown below:

| | #7 | #6 | #5 | #4 | #3 | #2 | #1 | #0 |
|---|---|---|---|---|---|---|---|---|
| Fn054 | UO007 | UO006 | UO005 | UO004 | UO003 | UO002 | UO001 | UO000 |
| Variable | #1107 | #1106 | #1105 | #1104 | #1103 | #1102 | #1101 | #1100 |

| | #7 | #6 | #5 | #4 | #3 | #2 | #1 | #0 |
|---|---|---|---|---|---|---|---|---|
| Fn055 | UO015 | UO014 | UO013 | UO012 | UO011 | UO010 | UO009 | UO008 |
| Variable | #1115 | #1114 | #1113 | #1112 | #1111 | #1110 | #1109 | #1108 |

| | #7 | #6 | #5 | #4 | #3 | #2 | #1 | #0 |
|---|---|---|---|---|---|---|---|---|
| Fn056 | UO107 | UO106 | UO105 | UO104 | UO103 | UO102 | UO101 | UO100 |
| Fn057 | UO115 | UO114 | UO113 | UO112 | UO111 | UO110 | UO109 | UO108 |
| Fn058 | UO123 | UO122 | UO121 | UO120 | UO119 | UO118 | UO117 | UO116 |
| Fn059 | UO131 | UO130 | UO129 | UO128 | UO127 | UO126 | UO125 | UO124 |

# III GSKLadder Use Introduction

# CHAPTER ONE BRIEF

## 1.1 GSKLadder Software Explanation

GSKLadder is a kind of CNC configuration software of the GSK-980D, which can be carried out the functions such as program edit, compile, upload and download of PLC ladder diagram program; it is convenient to use based upon its concise interface.

GSKLadder can be performed the operation or compilation of the PLC ladder diagram program in operation system of the WINDOWS98/2000/XP, as well save it as the file storage, and the print the PLC program by printer. The characters of this software are shown below:

- PLC ladder diagram can be compiled the network notes; it is convenient to read the program for the user;
- PLC command system supports the sub-program, program skip operation; as well increase the flexible of the programming.
- It is easy to operate the software based upon the user ambient menu and shortcut buttons
- The current project can be downloaded to CNC or upload the PLC program file to CNC by the serial port communication;

Refer to the *GSK980D Series CNC PLC User Manual – Programming* for the Syntax regulation and component setting of CNC PLC ladder diagram compilation, and the GSKLadder software operation and usage are described in detail in this Manual.

PC machine system requirements:

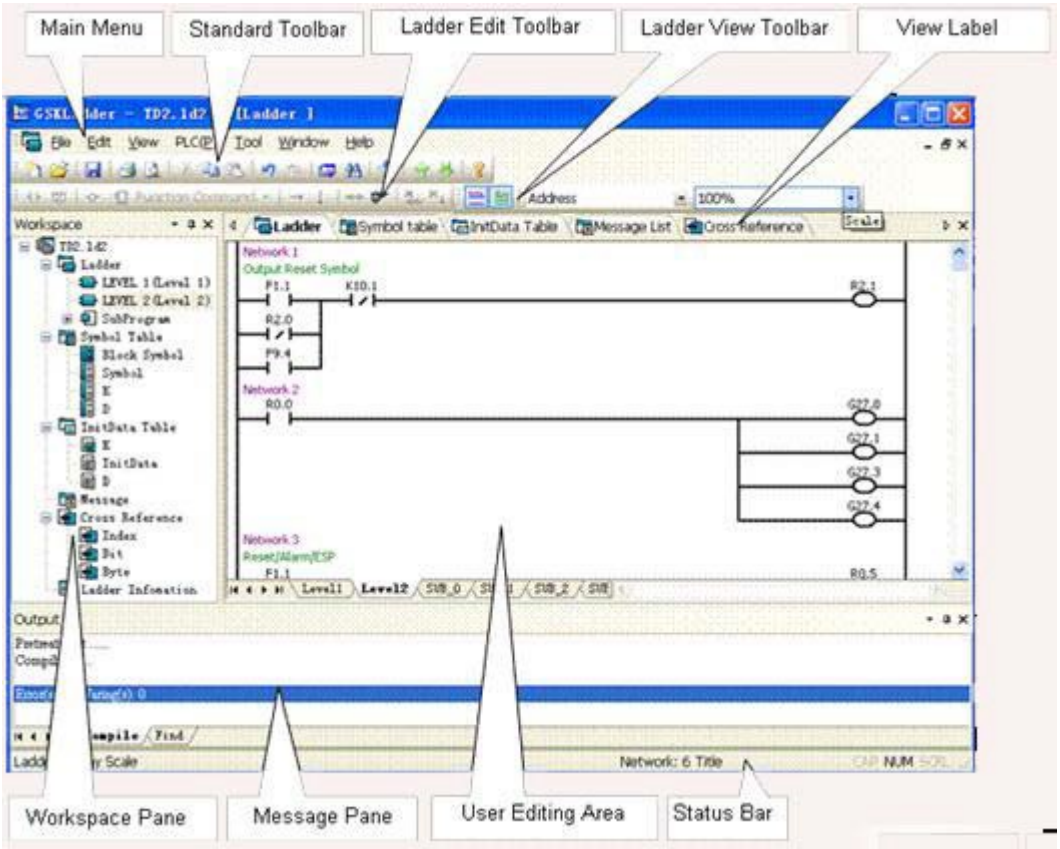
- Operation system: Windows98/2000/XP
- CUP: Pentium 133MHz or more
- Hardware: More than 10M
- Memory: More than 32M
- Display: The recommended resolution 1024×768, 16-bit color
- Keyboard, Mouse
- Serial communication port

## 1.2 Meaning of Project

The project administration PLC program file in this chapter includes the ladder diagram, symbol table, data setting table, display information table, quotation index table and ladder diagram version information.

# CHAPTER TWO MENU COMMAND

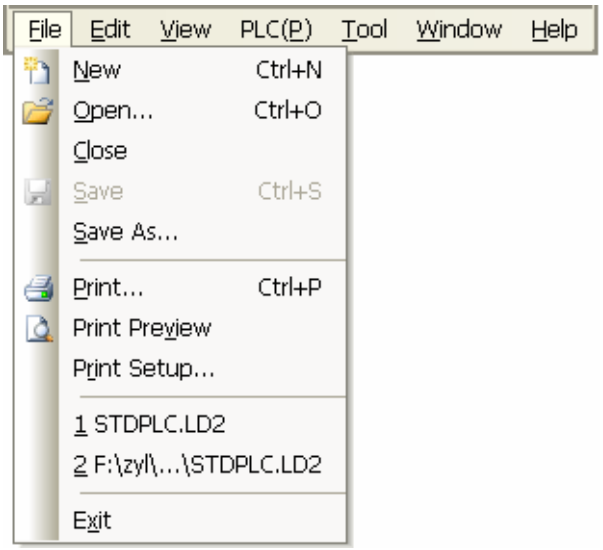
## 2.1 Screen Display




- Main Menu — All the operation commands
- Standard Toolbar — Daily-used commands
- Ladder Edit Toolbar — Ladder edit commands
- Ladder View Toolbar — Ladder display style
- View Label — Different views can be switched
- Workspace Pane — Different project configuration can be managed
- Message Pane — Outputs messages about PLC compiling and searching
- User Editing Area — Different views can be displayed, and the operations such as Ladder, Symbol Table and Initialized Data edit can be executed.
- Status Bar — To display the tool information, keyboard status and current cursor location etc.

## 2.2 Main Menu Commands

### 2.2.1 File Menu



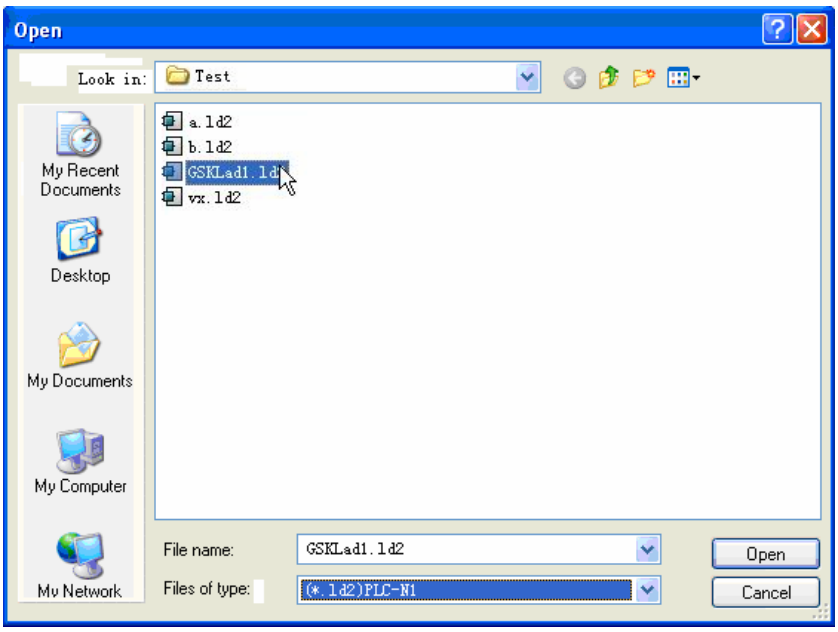

**[New]**

You can create a new project by using the keystroke of Ctrl+N, or clicking  on the standard toolbar.

The newly created project will be named “GSKLad#” (# is a digit). The project should be stored in disk by clicking “Save”, then a “Save as” dialogue box will pop up. Enter a proper name and savepath, and then click “OK” to save.

**[Open]**

You can open an existing project by using keystroke of [Ctrl+O] or clicking  on the standard toolbar, then, a dialogue box will pop up. Select the desired project, then, click [Open] to open the project.




**[Close]**

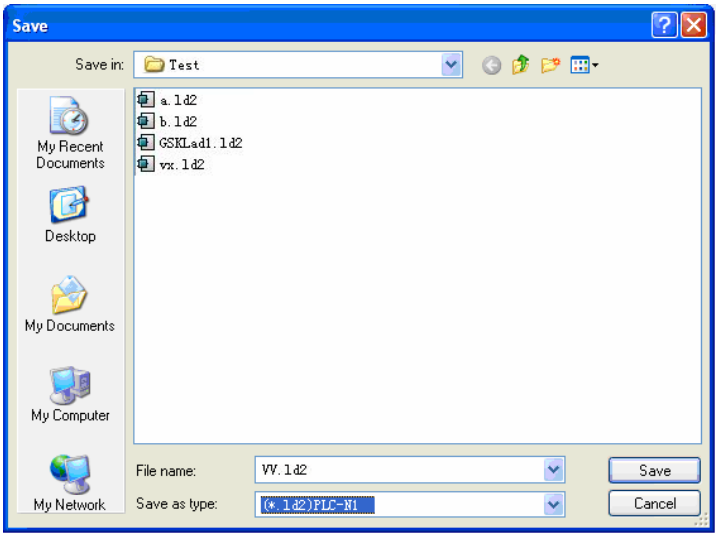
This command is used to close the current open project. If the project is not saved, a hint will pop up to confirm whether to save the current project.

**[Save]**

You can save the current open project by using keystroke of [Ctrl+S] or clicking on the standard toolbar.

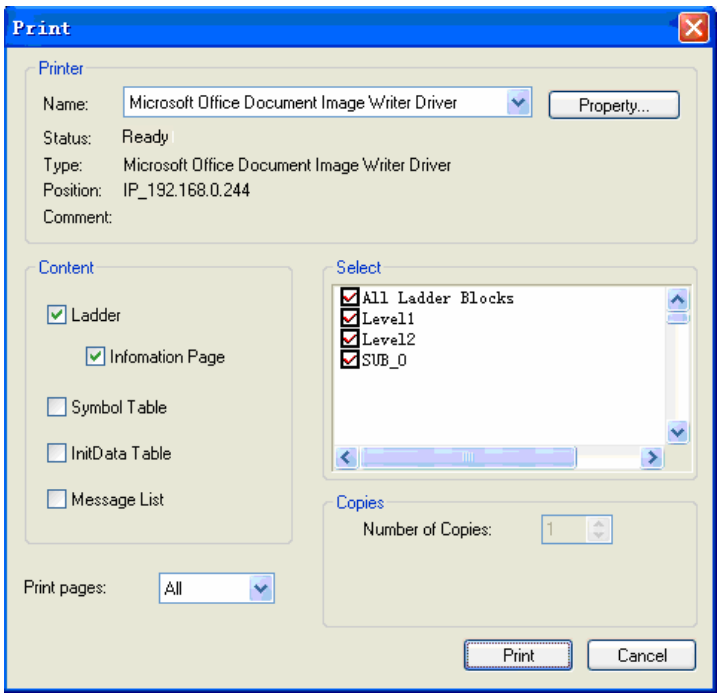
**[Save As]**

The current project can be backed up and saved as another file. When this command is executed, the following dialogue box will pop up. Fill in a proper name and save path, and then click “Save”.




**[Print]**

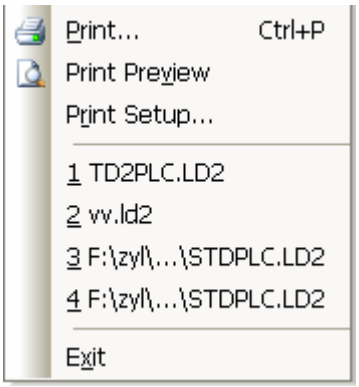
The current file can be printed through keystroke of [Ctrl+P] or clicking on the standard toolbar. In addition, a certain part of the contents within the current file can be selected. If the “Ladder” is selected, the blocks on the right list can also be selected.




**[Print Preview]**

It is used to preview the file before printing. Contents may vary in different views. For example, in the Ladder View, only ladder diagram is displayed; in Symbol Table View, only symbols are displayed. The style of ladder diagram is the same with the current view.

**[Recent Open File List]**

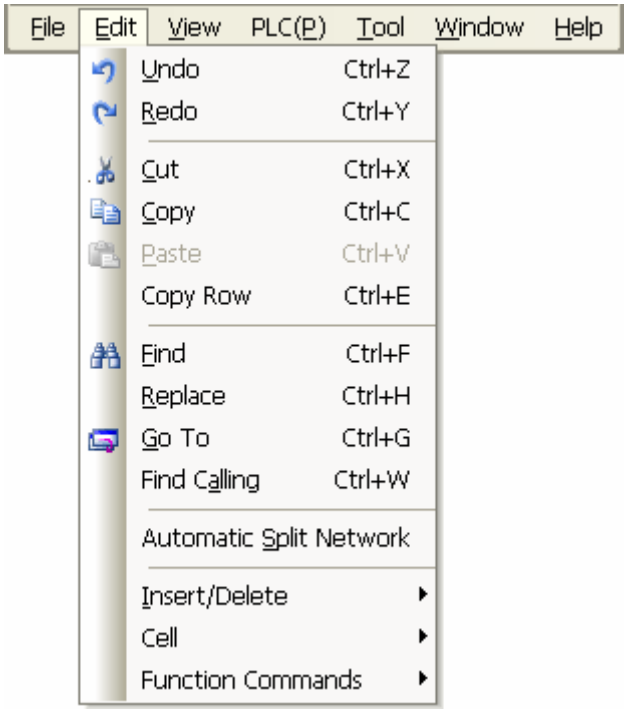



The list is the file names displayed below [Print Setting]. Four recent open projects can be listed and opened directly by clicking.

**[Quit]**

It is used to quit from the current project. If the project is not saved, a hint will pop up to confirm whether to save the current project.

## 2.2.2 Edit Menu




It should be noted that the last three items in “Edit” menu “Insert/Delete”, “Cell” and “Function Commands” are only displayed in the Ladder View.

**[Undo]**

You can undo the recent modified contents (up to 20 times) by using keystroke of [Ctrl+Z] or clicking  on the standard toolbar.

**[Redo]**

You can redo the recent undone operation by using keystroke of [Ctrl+Y] or clicking  on the standard toolbar. If the modification is made after the undo, Redo command cannot be executed.

**[Cut]**

You can cut the selected contents and copy it to the clipboard by using keystroke of [Ctrl+X] or clicking  on the standard toolbar.

**[Copy]**

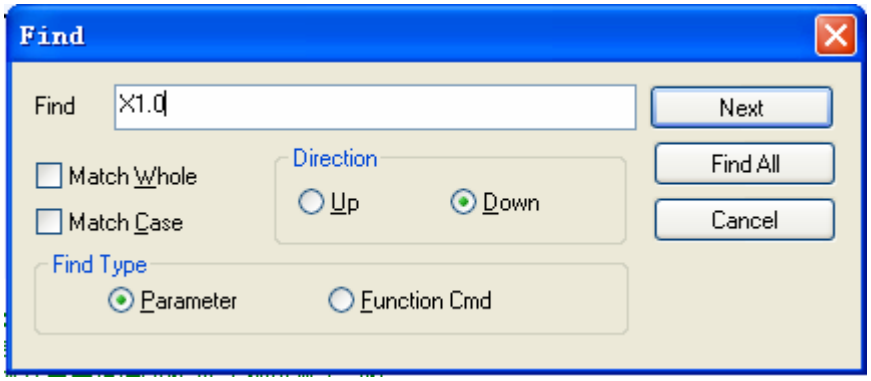
You can copy the selected contents in the clipboard by using keystroke of [Ctrl+C] or clicking  on the standard toolbar.

**[Paste]**

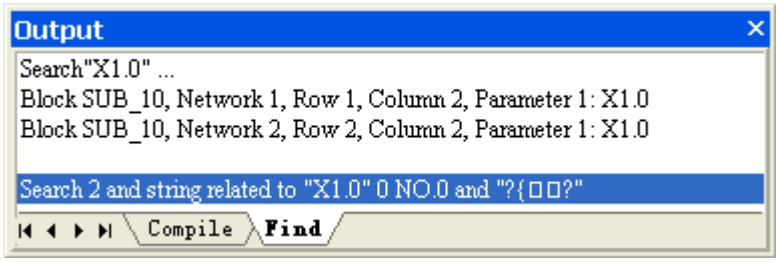
You can paste the contents in the clipboard to the selected position by using keystroke of [Ctrl+V] or  on the standard toolbar.

**[Find]**

You can find contents such as the character string or address by [Find] command. Use keystroke of [Ctrl+F] or click the  on the standard toolbar, then enter the contents to be found in the edit box. In Ladder View, you can select the Find Type the pop-up box, but in other views, it is not available.




Enter the contents to be searched in the edit box, then click [Next], the cursor will be located at the result position; if click [Find All], the results will be displayed in the information output pane; double-click one of the result, the cursor will be located at the corresponding contents. Shown as follows:

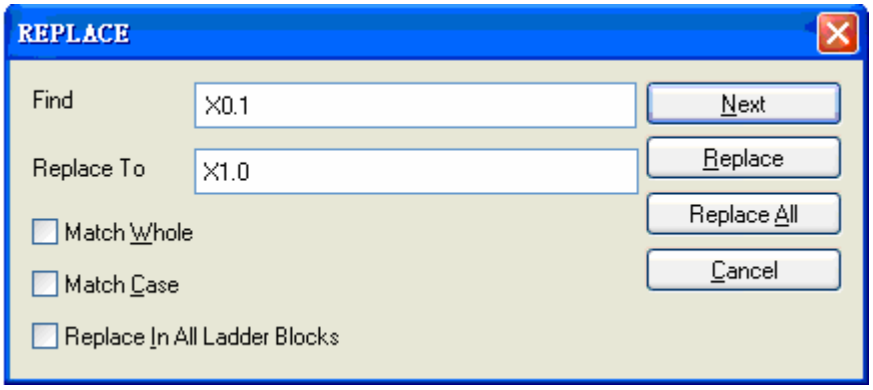



In Ladder View, the type of parameters (constant, addresses or symbols) can be exclusively searched according to the input character string. The options such as “Exact Match” and “Match Case” are available only when the symbols or function commands are searched. “Match Case” is invalid when searching for addresses. Both “Exact Match” and “Match Case” are invalid when searching for constants. For address, formats “x0.1” and “X0000.1” represent the same one and will lead to the same result.

In Table View, all the contents are processed as character strings.

**[Replace]**

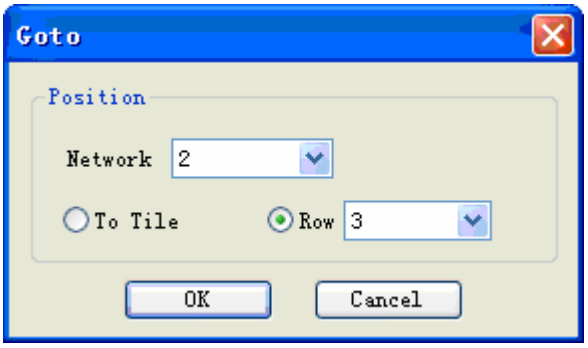
Specified contents can be replaced by new contents by using keystroke of [Ctrl+H]. The following dialogue box will pop up:



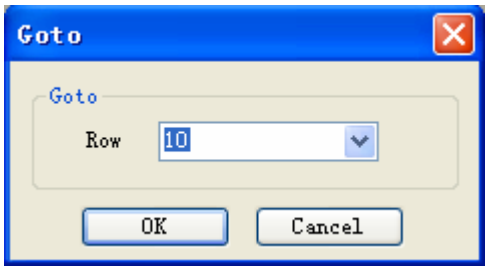

The find function in REPLACE dialogue box is the same as in [Find]. Replacement can be executed only when the search condition is fulfilled and the input content is legal. Address (or bit address) cannot be used to replace constant (or byte address), and vise versa.

**[Goto]**

Go to the designated location by using keystroke of [Ctrl+G] or the on the standard toolbar. A dialogue box will pop up. The dialogue box in Ladder View may be different from the one in other views. Shown as follows:



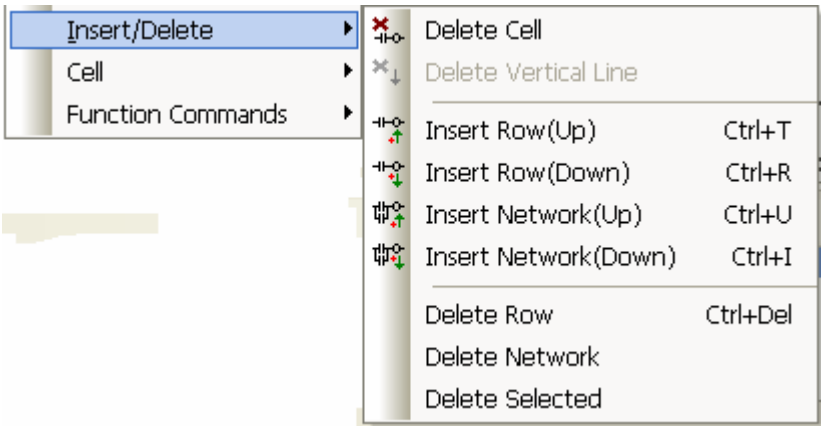

In Ladder View, select or input network position and row position in the dialogue box. The row position can be any row in the network or the network title (network title is the default row). Then, click [OK], the cursor will be located to the desired position.




In Table View, only rows are selectable in the dialogue box. Click [OK] after selection, then, the cursor will be located to the desired position.

**[Insert/Delete]**

There is a sub-menu subject to the [Insert/Delete]:




**---- [Delete Cell]**

Delete a cell of ladder diagram where the cursor located by using key [Delete] or clicking the on the Ladder Edit Toolbar.

**---- [Delete Vertical Line]**

Delete the vertical line on the left side of the cursor by clicking the on Ladder Edit Toolbar.

**---- [Insert Row (Up)]**

Insert a row above the cursor position by using keystroke of [Ctrl+T].

**---- [Insert Row (Down)]**

Insert a row below the cursor position by using keystroke of [Ctrl+ R].

**---- [Inset Network (Up)]**

Insert a network above the cursor position by using keystroke of [Ctrl+U].

**---- [Inset Network (Down)]**

Insert a network below the cursor position by using keystroke of [Ctrl+I].

**---- [Delete Row]**

Delete the row at cursor position. A blank row will be inserted if there is only one row in the current network.

**---- [Delete Selected]**

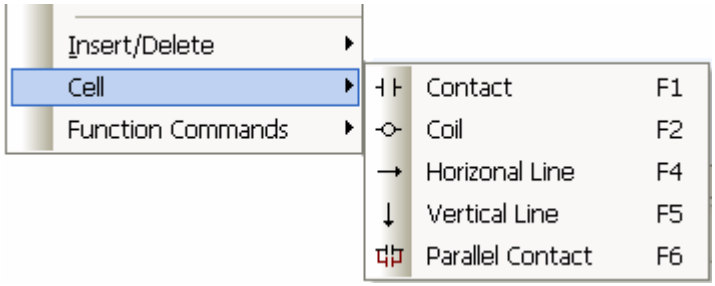
Delete the selected region by using key [Delete]. A network will be inserted if the current block is empty after deletion.

**---- [Delete Network]**

Delete the network at the cursor position.

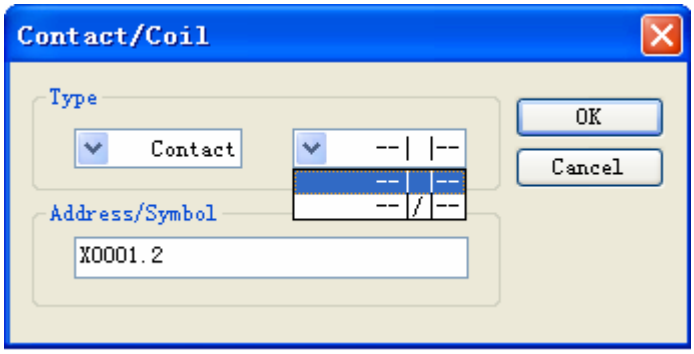
**[Cell]**

There is a sub-menu subject to the [Cell]. Shown as follows:



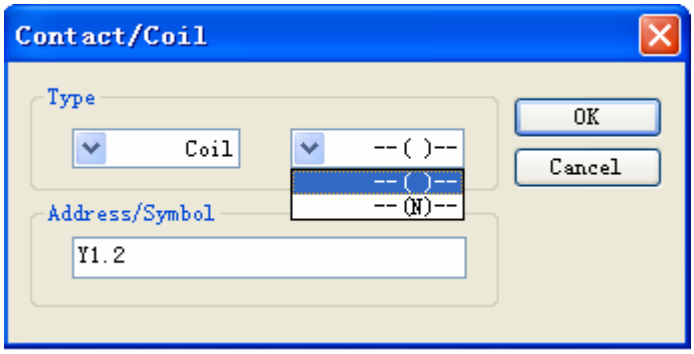

**----[Contact]**

Add a contact (normally-closed/normally-open contacts) by using key [F1] or clicking on the Ladder Edit Toolbar. A dialogue box will pop up for the setting of contact type and address/symbol.




**---- [Coil]**

Add an output coil at the selected position by using key [F2] or clicking on Ladder Edit Toolbar. A dialogue box will pop up for the setting of coil type and address/symbol.




**---- [Horizontal Line]**

Add a horizontal line at the selected position by using key [F4] or clicking on the Ladder Edit Toolbar.

**---- [Vertical Line]**

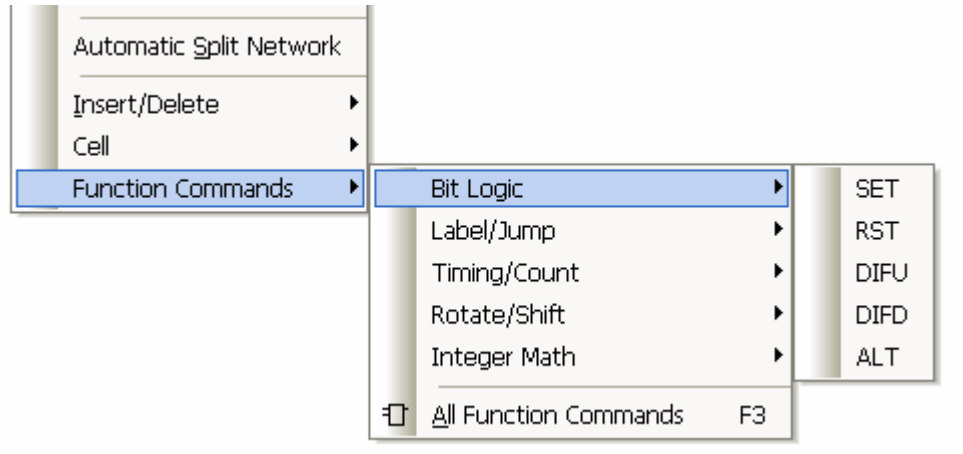
Add a vertical line right to the selected position by using key [F5] or clicking on the Ladder Edit Toolbar.

**---- [Parallel Contacts]**

Add a contact at the selected position and add vertical lines at two sides of the contact, so as to make it parallel to the contact in the above line. Using Key [F6] or clicking on the Ladder Edit Toolbar is the way to realize the operation.

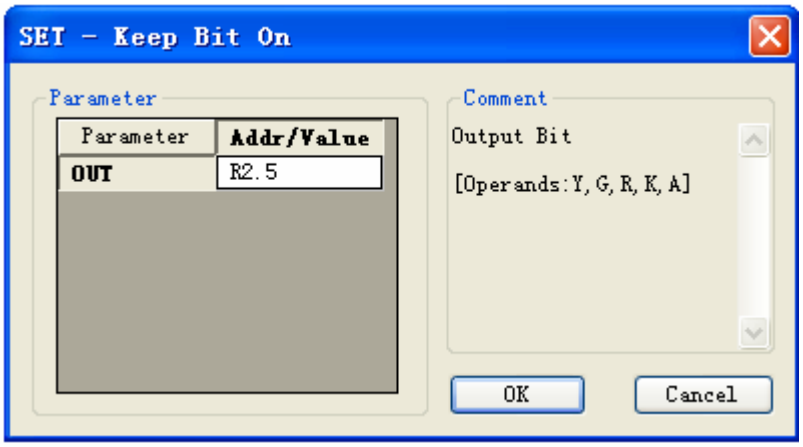
**---- [Function Commands]**

The sub-menu subject to [Function Commands] is shown as follows:



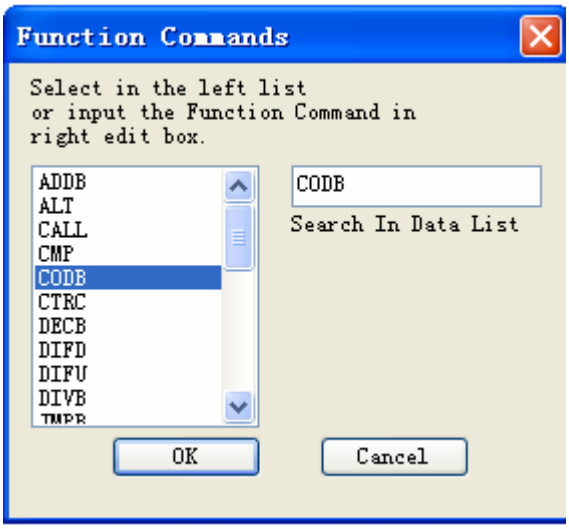

**---- [Submenus]**

There are five submenus including [Bit Logic], [Label/Jump], [Timing/Count], [Rotate/Shift], [Integer Math]. Each submenu contains multiple function commands. When one of the commands is selected, the edit window will pop up. Click [OK] after edition, then, the command will be added to the desired position. Take SET command for example, click [Edit]—[Function Commands]—[Bit Logic]—[SET], then, the following edit window will pop up. The left side of the window is parameter list. Parameter value can be entered in the second column and will be displayed in red when it is erroneous; the right side of the window is the comment for the selected parameter.



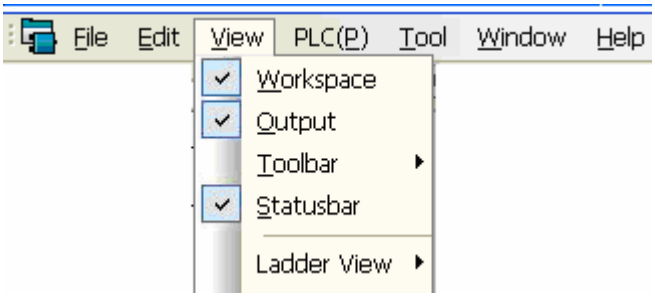

**---- [All Function Commands]**

You can also add function commands by clicking on the Ladder Edit Toolbar. A dialogue box will pop up for selection:




The function commands are selectable in the left side and can also be entered in the edit box. Double-click the command or click [OK] after entering the command. The execution result is the same as executing the commands in submenus.

## 2.2.3 View Menu




The last item in the menu [Ladder View] is displayed only in Ladder View.

**[Workspace]**

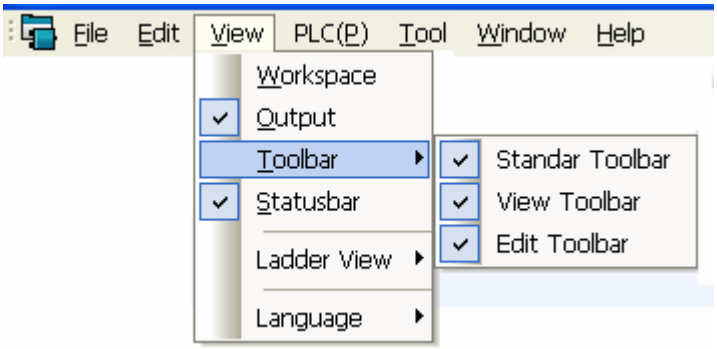
Display/Do not display the workspace pane.

**[Output]**

Display/Do not display the message output pane.

**[Toolbar]**

Display/Do not display the toolbar. The drop-down menu is shown as follows:




**---- [Standard Toolbar]**

Display/Do not display standard toolbar.

**---- [View Toolbar]**

Display/Do not display Ladder View toolbar.

**---- Edit Toolbar]**

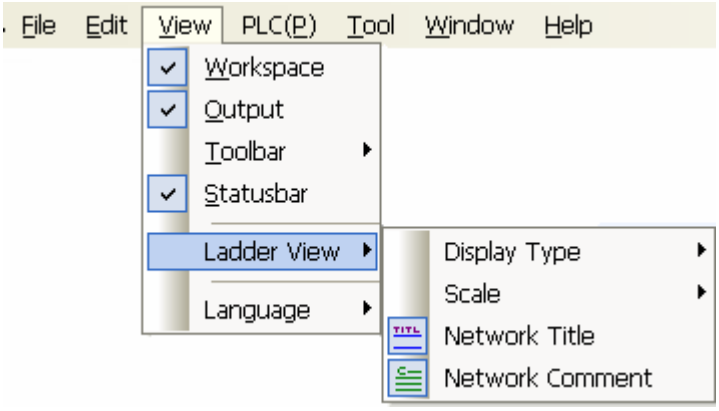
Display/Do not display Ladder Edit Toolbar.

**[Status Bar]**

Display/Do not display the status bar below the main frame window.

**[Ladder View]**

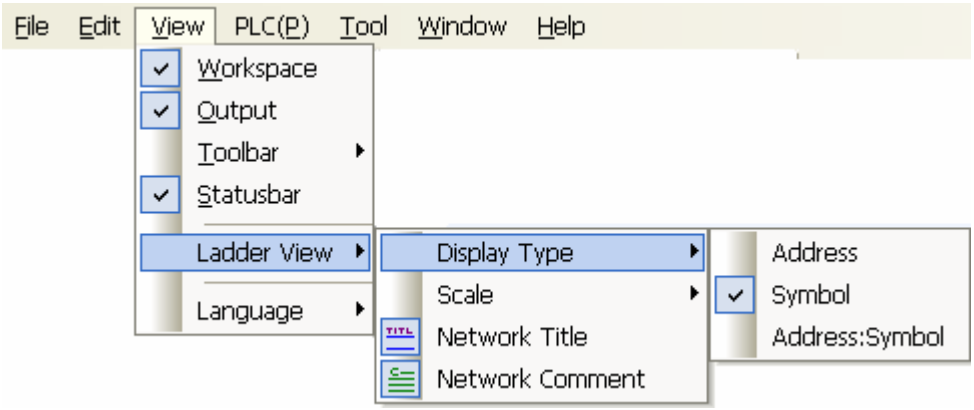
The drop-down menu for the setting of the Ladder View is shown as follows:



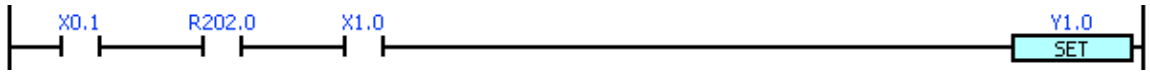



**---- [Display Type]**

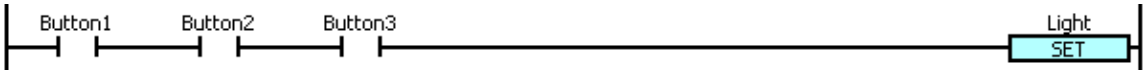
Parameters can be displayed in three types: “Address”, “Symbol”, “Address: Symbol”. They can also be set via the combobox on the Ladder View Toolbar. When display type “Address” is selected, parameters are displayed in addresses, except for those in the format of symbols and have no corresponding addresses (these symbols will be turned into blue if converted to addresses). When display type “Symbol” is selected, parameters are displayed in symbols, except for those whose type cannot be “Symbol”, such as address parameters (in addresses), and constants (in digits).



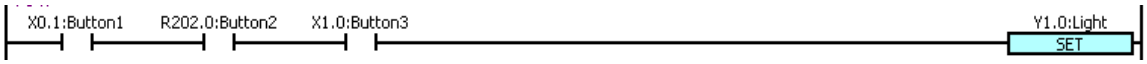

Displayed in “Address” type:



Displayed in “Symbol” type:

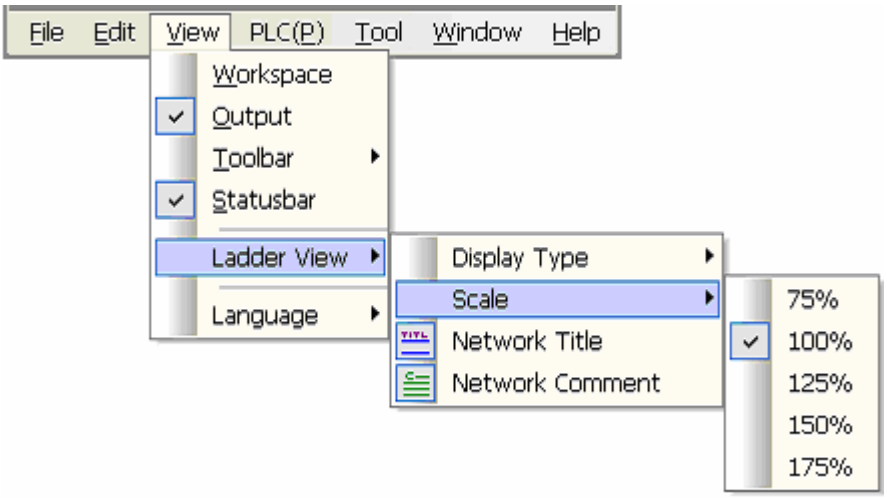



Displayed in “Address: Symbol” type:




**---- [Scale]**

The ladder display scaling ratio can be 75%, 100%, 125%, 150% or 175%. They can be set via the combobox on Ladder View Toolbar.



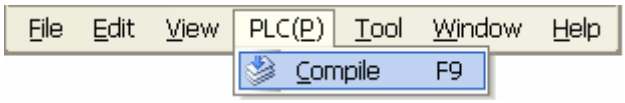

**---- [Network Title]**

Display/Do not display the network title through  on Ladder View Toolbar.

**----[ Network Comment]**

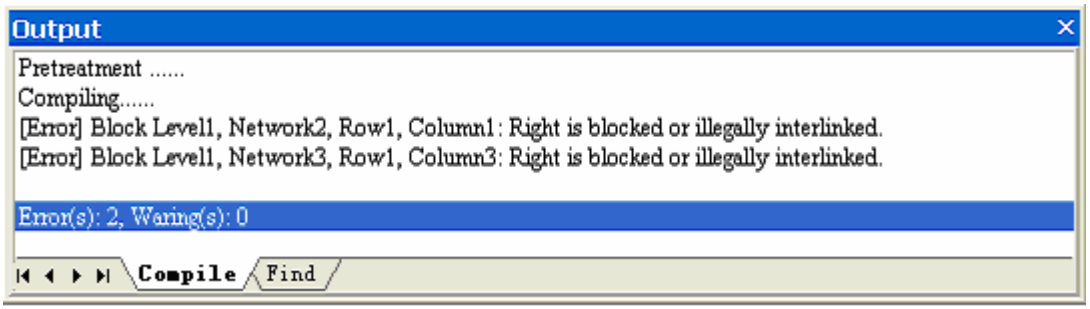
Display/Do not display the network comment through  on Ladder View Toolbar.

## 2.2.4 PLC Menu

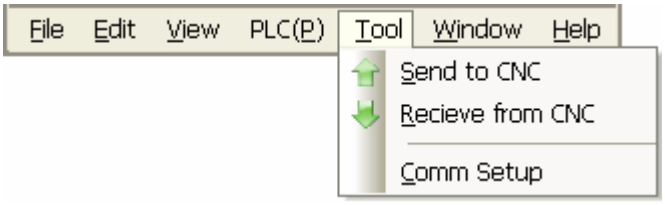



**[Compile]**

Compile the current PLC programs via key F9 or  on the standard toolbar. The information after compilation is displayed in the message output pane. Double-click the displayed error or alarm to trace the source.




## 2.2.5 Tool Menu




**[Send to CNC]**

Send the files in current project to CNC for storage.

**[Receive from CNC]**

Read the PLC files from CNC into PC.

**[Comm Setup]**

Set serial port parameters including serial port number and communication baudrate.

# CHAPTER THREE TOOL COLUMN

## 3.1 Standard Toolbar

 Create a new project

 Open an existing project

 Save the current project

 Print the ladder diagram

 Print preview

 Cut the selected area

 Copy the selected area

 Paste in the selected area

 Undo the last operation

 Redo previously "undone" operations

 Goto the specified position

 Find the designated contents

 PLC compilation

 Send current project to CNC

 Receive PLC files from CNC

 Display program information, version number and copyright.

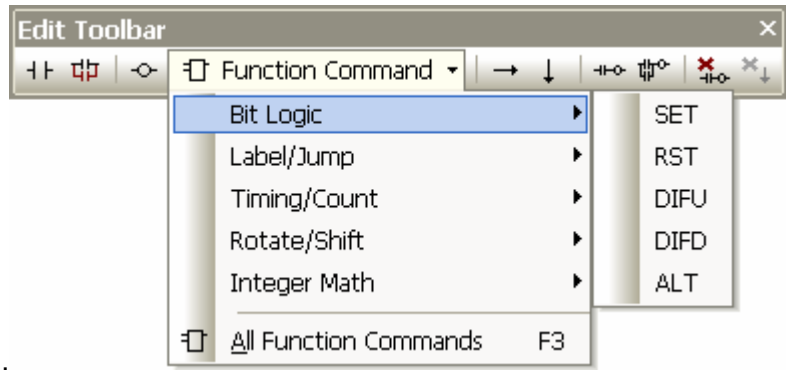
## 3.2 Ladder Edit Toolbar

Add contact at the cursor position (shortcut key F1)

Add serial contacts at the cursor position

Add output coil at the cursor position (shortcut key F2)

Function Command Add function commands at the cursor position. The function commands can be selected in the fly-out list




Add horizontal line at the cursor position (shortcut key F4)

Add vertical line at left side of cursor position (shortcut key F5)

Delete the selected ladder diagram cell (shortcut key Delete)

Delete the vertical line at the left side of the selected cell.

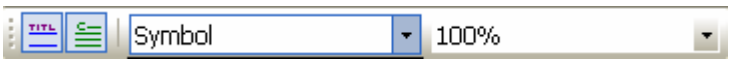
Add one row above the cursor position

Add one row below the cursor position

Add one network above the cursor position

Add one network below the cursor position

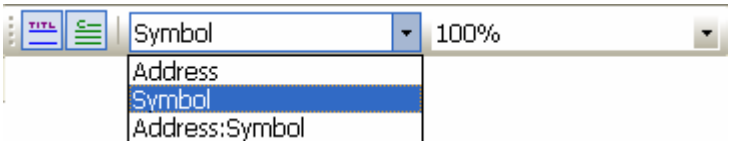
## 3.3 Ladder View Toolbar




Display/Do not display network title

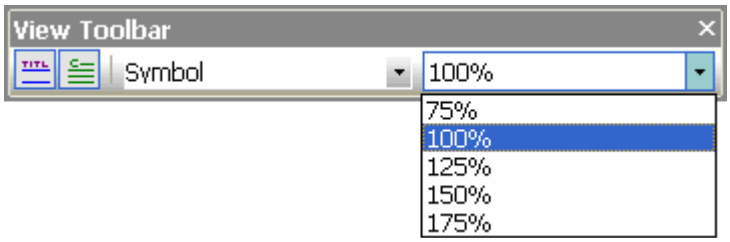
Display/Do not display network commend

Parameter display types combobox




Parameters can be displayed in three types: “Address”, “Symbol”, “Address: Symbol”. They can also be set via the on Ladder View Toolbar. When display type “Address” is selected, parameters are displayed in addresses, except for those in the format of symbols and have no corresponding addresses (these symbols will be turned into blue if converted to addresses). When display type “Symbol” is selected, parameters are displayed in symbols, except for those whose type cannot be “Symbol”, such as address parameters (in addresses), and constants (in digits).

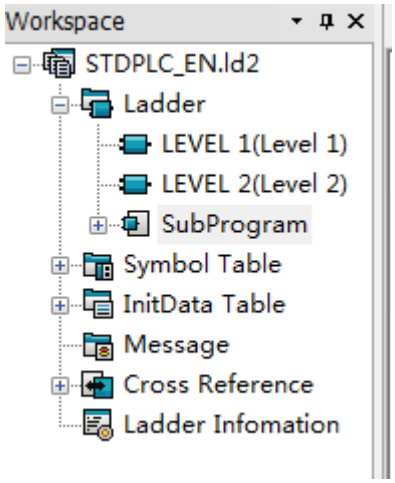
Scaling combobox




The scaling ratio can be 75%, 100%, 125%, 150% or 175%. They can be set via the on Ladder View Toolbar.

# CHAPTER FOUR SOFTWARE USAGE

## 4.1 Project Administrator Window




The workspace is of tree structure. The project name is represented by root node which has 6 children: [Ladder], [Symbol Table], [InitData Table], [Message], [Cross Reference], [Ladder Information]

**[Ladder]**

It consists of [Level 1], [Level 2] and [subprogram]. The tree node number subject to the [subprogram] is not limited and can be added or deleted.

**[Symbol Table]**

It consists of [Block Symbol] and some user-defined symbol table nodes. The nodes in [Block Symbol] are fixed while the node number of user-defined symbol is related to the number of symbol tables.

**[InitData Table]**

It consists of [K Value Setting] and some user-defined symbol table nodes. The nodes in [K] are fixed while the node number of user-defined data table is related to the number of data tables.

**[Message]**

It has no branch node, which means only one message table can be displayed.

**[Cross Reference]**

It consists of three children: [Index], [Bit], [Byte] which cannot be deleted or edited. Node cannot be added to the cross reference.

**[Ladder Information]**

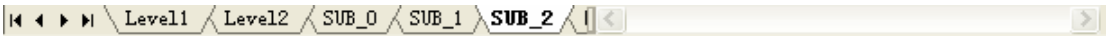
It is a fixed node without child.

### 4.4.1 View Open and Switch

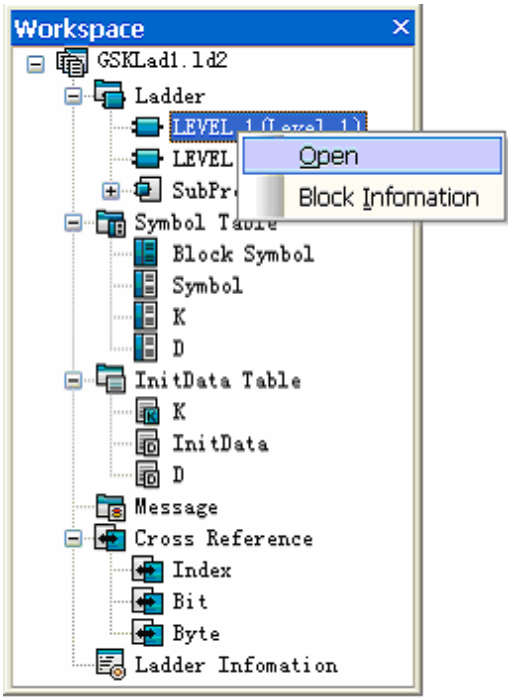
There are three ways to switch among views: double-click the tree node in project manager; click the [Open] in Workspace Pane tree note; click the view switch label on the top or bottom of the user edit area.

View switching label

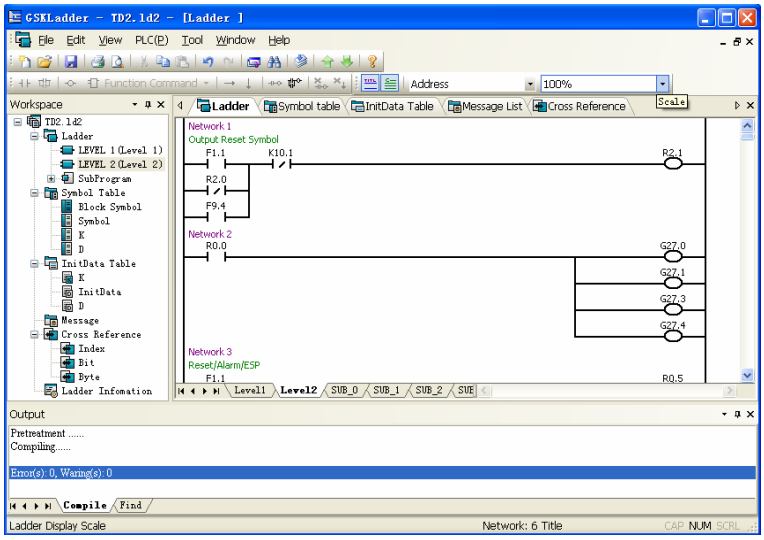

Sub-view switching label




Command [Open] on the Workspace tree note




### 4.4.2 Ladder




After a project is open, the current view is the Level 1 of Ladder. You can switch among different

blocks and views. The operations of menus and toolbars are described in previous sections. The following paragraphs are about the subprogram creation, rename, deletion, block message edit and network note addition.

#### 4.4.2.1 Create, Rename or Delete a Subprogram

**Create a subprogram**

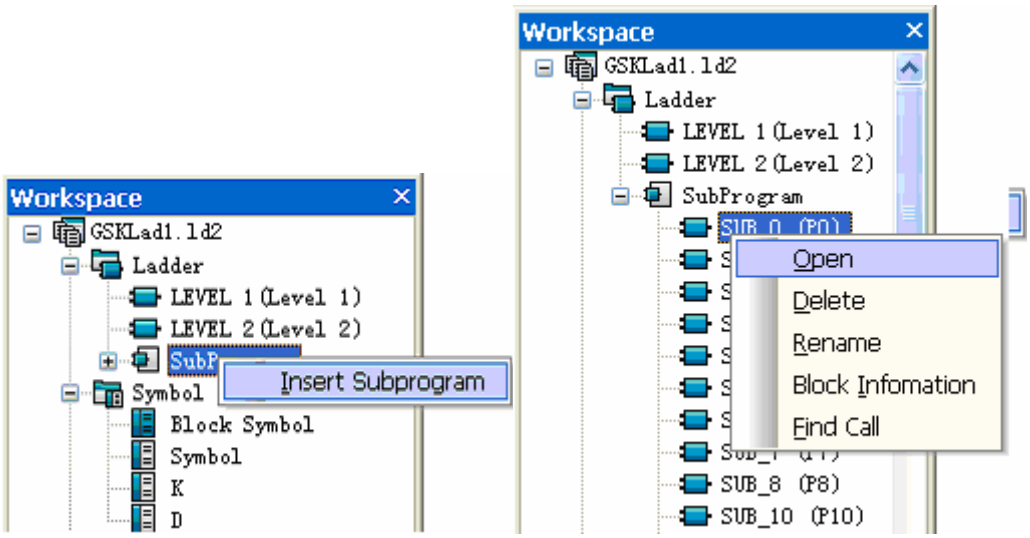
Click command [Insert Subprogram] after right-clicking [Subprogram] node, a new subprogram will be generated, which, at the mean time, enables the generation of a new node and a sub-view label.

**Rename a subprogram**

Expand the [Subprogram] by clicking the + symbol, then, click [Rename] on the fly-out menu, or left-click the sub-node to be renamed, the character string becomes editable, then press “Enter” on the keyboard. Please note that the new name cannot be consistent with other names of blocks (including level 1 and level 2 programs).

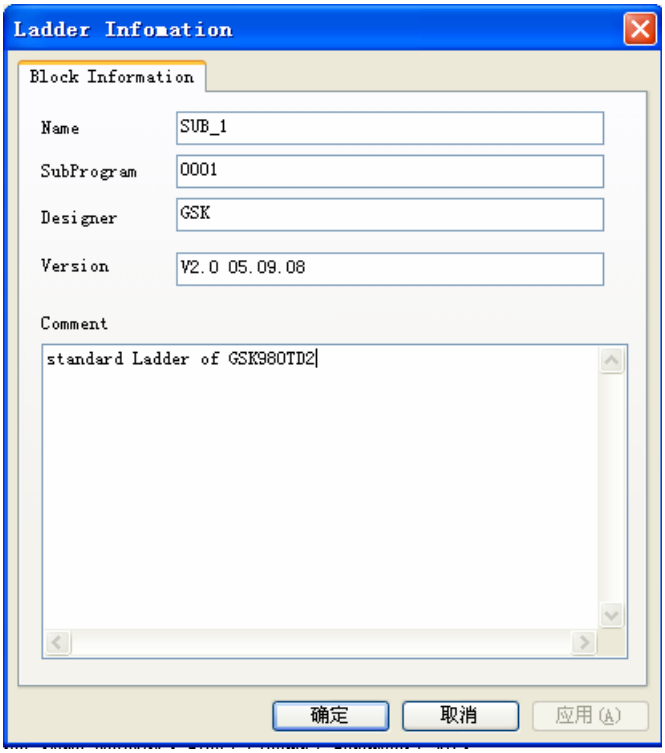
**Delete a subprogram**

Click [Delete], a dialogue box will pop up to confirm the deletion, then, the corresponding subprogram will be deleted if your answer is yes.



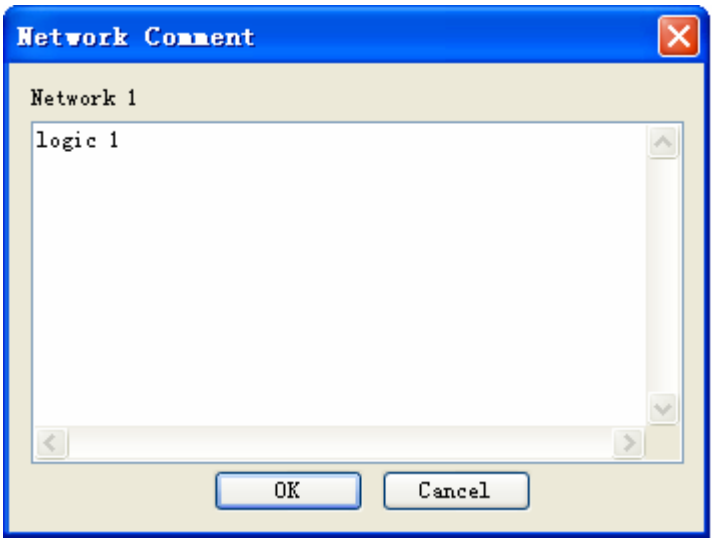

#### 4.4.2.2 Modify Block Information

Click [Block Information] in the fly-out menu, a dialogue box will pop up. Click [OK] after editing proper information, otherwise, click [Cancel] to close the dialogue box.



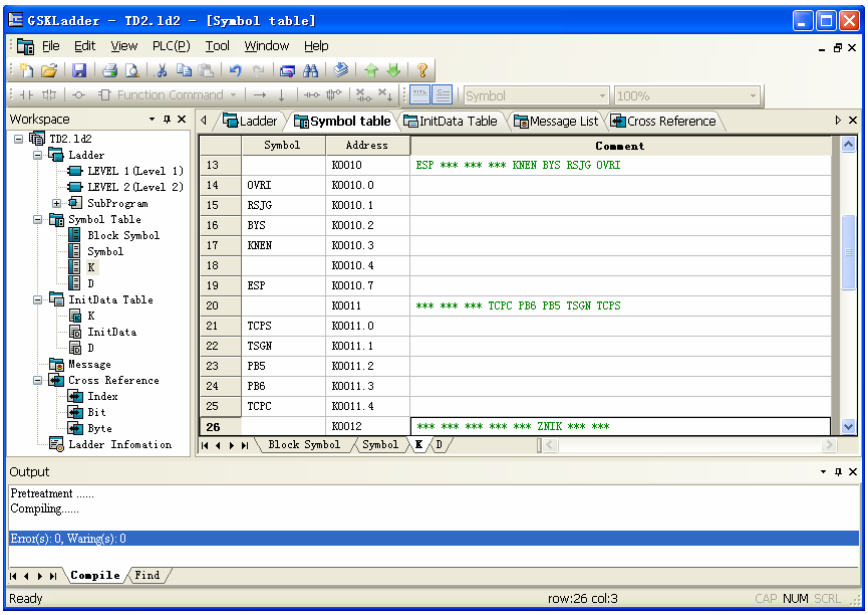

#### 4.4.2.3 Add Network Comment

Double-click the network title in Ladder View; the following dialogue box will pop up. Modify the network comment in edit box and then click [OK] to validate it, or click [Cancel] to close the dialogue box.




### 4.4.3 Symbol Table

Click [Symbol Table] to switch to the symbol table view frame. You can switch among different symbol tables by clicking different sub-view labels.




The main effect of symbol table is to realize the mapping between symbol and address. This kind of mapping relationship enables the user to replace addresses by symbols during PLC programming. The symbol table can be deleted and added, except for the table “Block Symbol” which is fixed and not editable. It is used to display the mapping relationship between subprogram name and subprogram address, thus, the subprogram name can also used as a symbol. Other symbol tables are user-defined. The following paragraphs describe how to create and delete a symbol table.

### 4.4.3.1 Create, Rename and Delete a Symbol Table

**Create a symbol table**

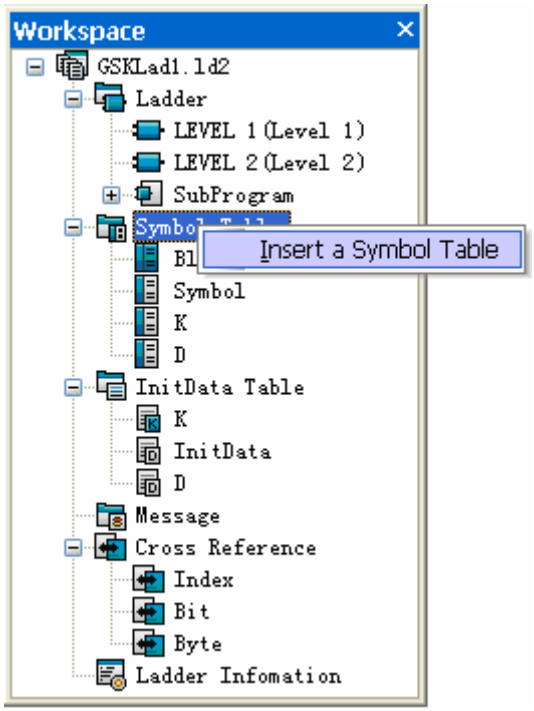
Click command [Insert Symbol Table] after right-clicking [Symbol Table] node, a new symbol table will be generated, which, at the mean time, enables the generation of a new node and a sub-view label.

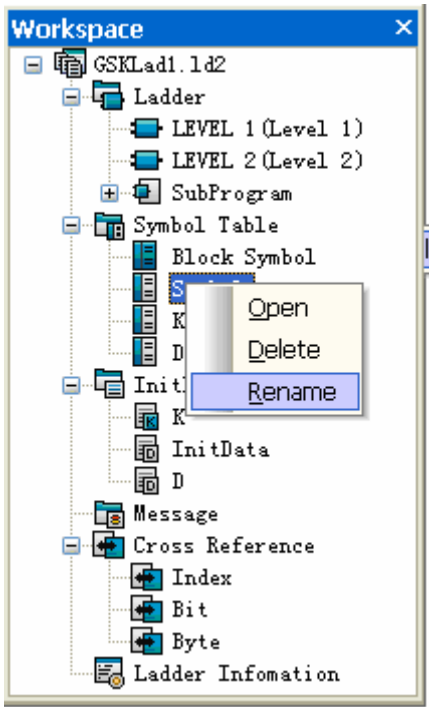
**Rename a symbol table**

Expand the [Symbol Table] by clicking the + symbol, then, click [Rename] on the fly-out menu, or left-click the sub-node to be renamed, the character string becomes editable, then press “Enter” on the keyboard. Please note that the new name cannot be consistent with other names of symbol tables.

**Delete a symbol table**

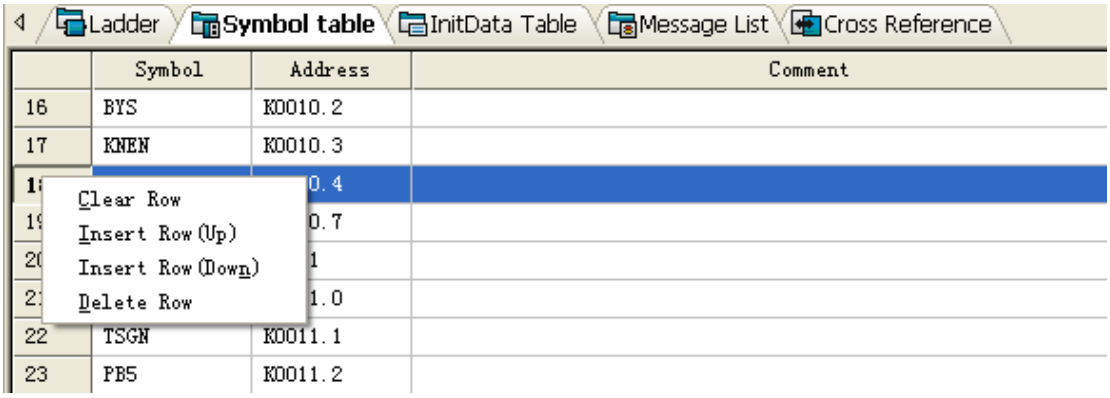
Click [Delete], a dialogue box will pop up to confirm the deletion, then, the corresponding symbol table will be deleted if your answer is yes.




### 4.4.3.2 Symbol Table Edit

**Edit of Rows:** Right-click a row header of the symbol table, a menu will pop up. Click [Clear Row], the contents in the row will then be cleared; click [Insert Row (Up)] to insert a row above the current position; click [Insert a Row (Down)] to insert a row below the current position; click [Delete Row] to delete the selected row.




**Symbol input**

The format of input symbol is limited within letters, digits, underlines and Chinese characters. Digit should not be the head and the total length of a symbol should not exceed 32 characters. Symbols should not be identical to each other; otherwise, a hint will remind you the existence of such symbol.

**Address input**

The format of address is also limited. The format of byte address is: type (letter)+address number (digit); the formation of bit address is: type (letter) + address number (digit)+ “.” + bit number (digit). The allowable input types are: A, X, Y, R, K, F, C, T, D, DT, DC. The addresses should not be the

same; otherwise, the same addresses will be displayed in green for identification.

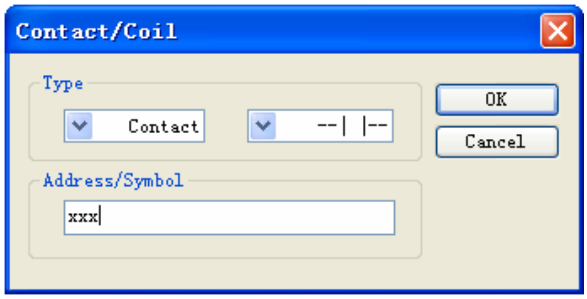
**Comment input**

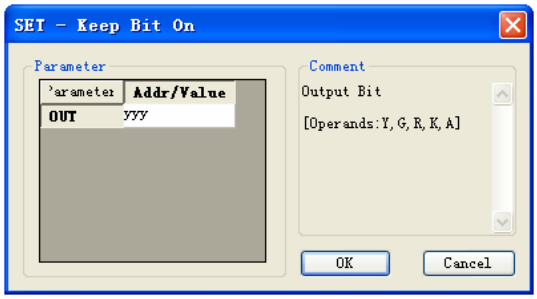
Note should be limited within 127 bytes, but the contents and format are not limited and can be empty as well.

There is exceptionality: it is allowable to input address and note without symbol. It can be regarded as note for the address; however, it is not allowed to input symbol instead of address, the symbol is regarded as invalid.

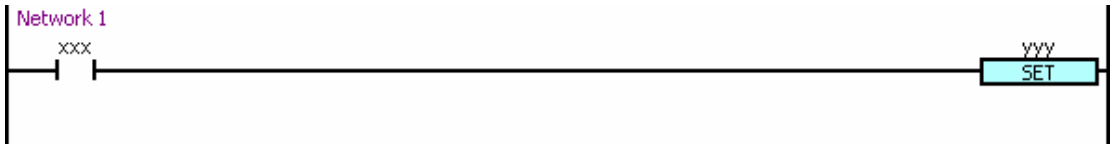
#### 4.4.3.3 Usage of Symbols

The usage of symbol is pretty easy. When you edit cell, just input a symbol as a parameter. Symbols can be used before they are defined.




When a parameter is displayed in “Address” view, the symbols which represent the parameter are displayed in blue; if the symbols are invalid or undefined, they are displayed in red. In “Symbol” view or “Address: Symbol” view, symbols are black when the parameter is correct; when the mapping address type is not the required one, the symbol is orange; symbols are red when they are incorrect or undefined.




### 4.4.4 InitData Table

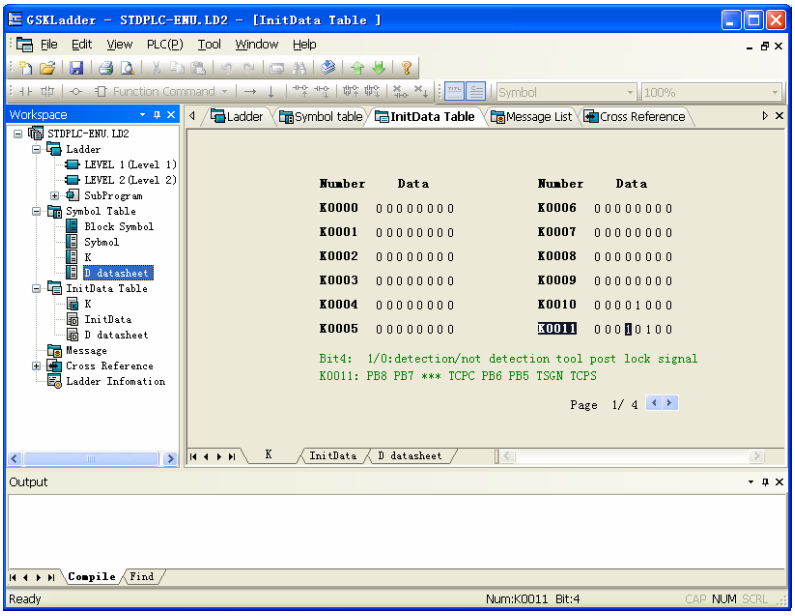
Click [InitData Table] label to switch to the corresponding frame window which includes two different edit screens: parameter screen and table screen. Parameter K screen is used to set the K value and data table screen is used to input the initialized data of D, DT and DC. The default screen is “K value setting” which is fixed and un-removable (see Fig. 1-17). Except for K value, other values are addable and deletable.

#### 4.4.4.1 K Value Setting

The parameter page for K value setting consists of parameters sorted by column. Each parameter consists of sequence number and data. As the parameters cannot be displayed in one page, they may be divided into several pages, and the page number and parameter numbers of each page depend on

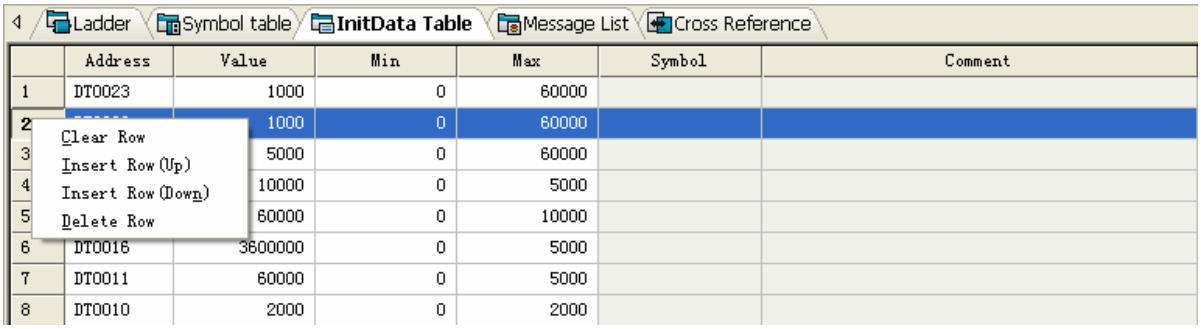
the size of view area. Turn the pages by clicking the forward or backward buttons on the lower-right corner or pressing [PageUp] and [PageDown]. There are two lines of notes in green at the bottom of the page, one is the note for bit, and the other is the notes for individual parameter. These notes are not preset in the software but user-defined, which means users can add or edit these notes in symbol table.

K value setting is edited in bits. To modify a bit, you need to double-click the bit or move cursor the bit then press [Enter].






### 4.4.4.2 Edit of InitData Table (D, DT, DC)




**Edit of Rows**

Right-click a row header of the InitData Table, a menu will pop up. Click [Clear Row], the contents in the row will then be cleared; click [Insert Row (Up)] to insert a row above the current position; click [Insert a Row (Down)] to insert a row below the current position; click [Delete Row] to delete the selected row.

**Address Input**

The address input is similar to the input in the symbol table, but only types D, DC, DT are supported in InitData Table.

**Data Input**

The input data should be set between the data range and will be clamped at the upper limit or the lower limit if exceeds. Integers from -2147483647 to 2147483647 can be input if no upper or

lower limit is set.

**Minimum Data Input**

Integers from -2147483647 to 2147483647 can be input if no upper or lower limit is set. If the maximum data exists, the input data should be limited within -2147483647 to the maximum data. If the input data is smaller than -2147483647, it is clamped at -2147483647; if it is greater than the maximum data, it is clamped at the maximum data. If the modified minimum data is greater than the input data, the input data will be re-set to the modified one.

**Maximum Data Input**

Integers from -2147483647 to 2147483647 can be input if no upper or lower limit is set. If the minimum data exists, the input data should be limited within the minimum data to2147483647. If the input data is greater than 2147483647, it is clamped at 2147483647; if it is smaller than the minimum data, it is clamped at the minimum data. If the modified maximum data is smaller than the input data, the input data will be re-set to the modified one.

#### 4.4.4.3 Create, Rename or Delete InitData Table

**Create an InitData Table**

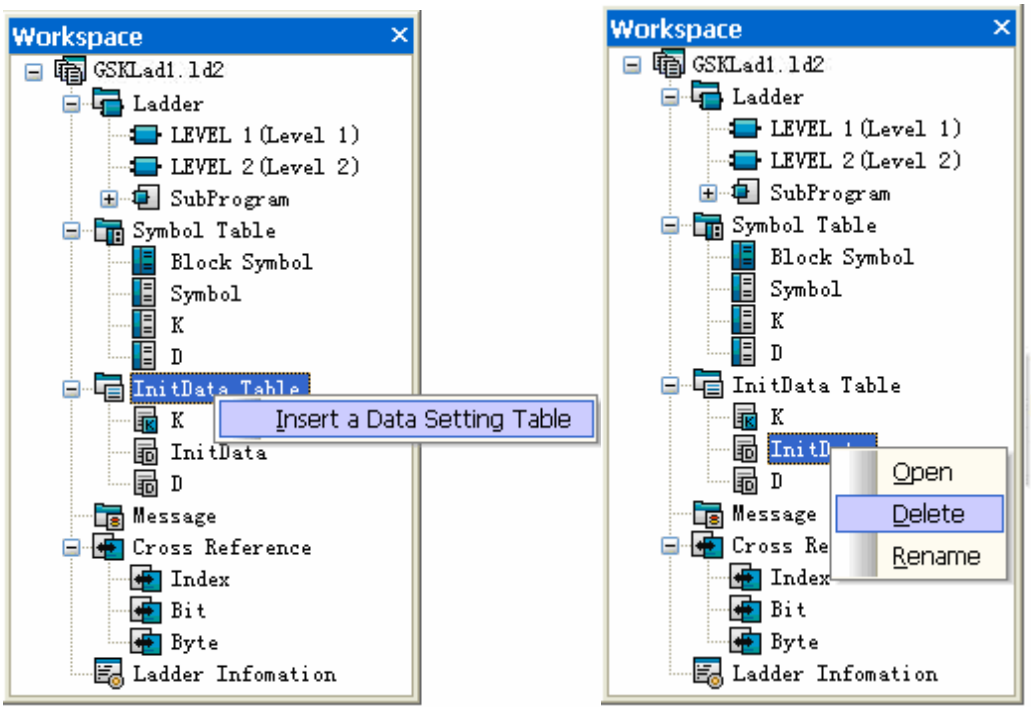
Click [Insert a Data Setting Table] after right-clicking the [InitData Table] node, a new subprogram will be generated, which, at the mean time, enables the generation of a new node and a sub-view label.

**Rename an InitData Table**

Expand the [InitData Table] by clicking the + symbol, then, click [Rename] on the fly-out menu, or left-click the sub-node to be renamed, the character string becomes editable, then press “Enter” on the keyboard. Please note that the new name cannot be consistent with other names of tables (including “K value setting” table).

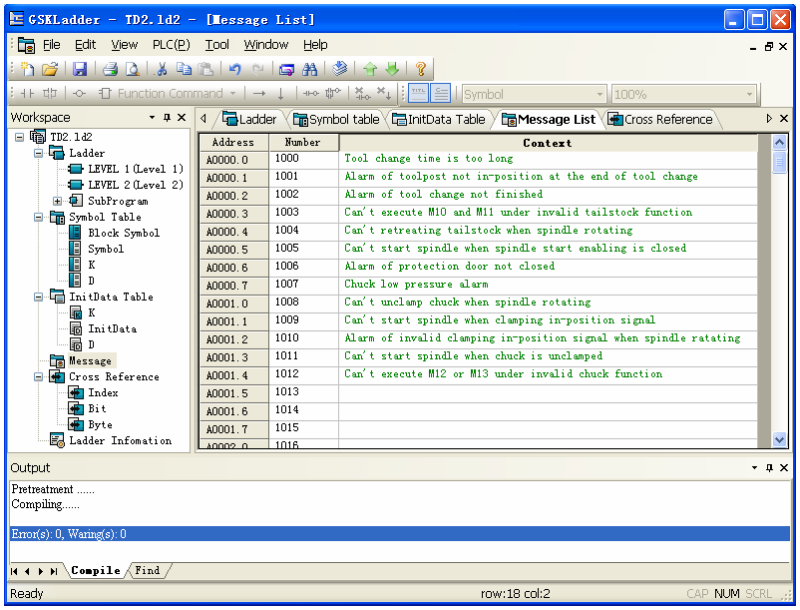
**Delete an InitData Table**

Click [Delete], a dialogue box will pop up to confirm the deletion, then, the corresponding table will be deleted if your answer is yes.




## 4.4.5 Message List

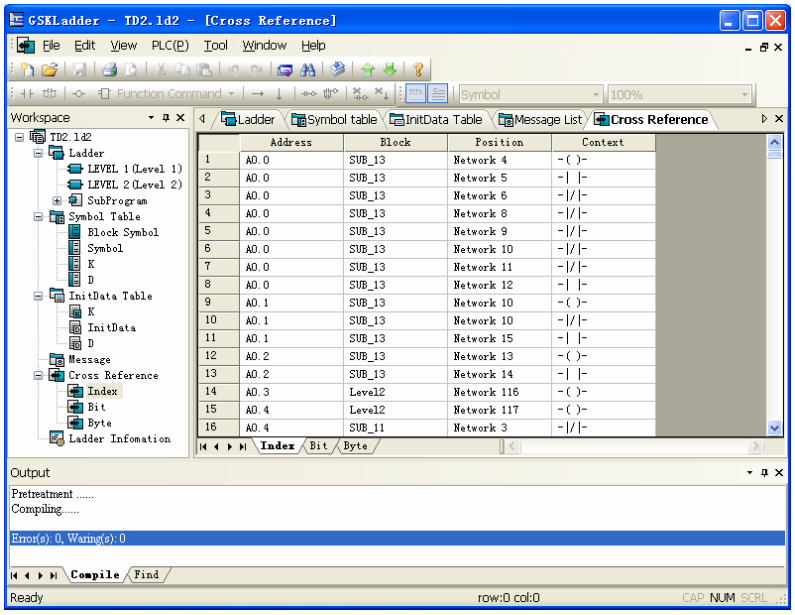
Click [Message List] label. The displayed message list contains 200 rows which cannot be added or deleted. The address is listed from A0000.0 to A0024.7, and cannot be added or deleted neither. The alarm number range is 1000~9999 without the same data. Both the alarm number and displayed contents should be input indispensably; otherwise, an alarm will occur during compiling.




## 4.4.6 Cross Reference List

Click [Cross Reference] label. It is used to show the addresses using and assignment conditions in PLC. It includes “Index”, “Bit” and “Byte” three lists which are uneditable and usually empty. The relevant

information is generated only after compilation. The contents in the three lists will be cleared once the projected is modified.




### 4.4.6.1 Index List

It is used to display the context of referenced address, so that a user can find the address position with ease. There are five columns in the list: row header, address, block, position and context. Double-click the corresponding table cells to goto the desired position.

### 4.4.6.2 Bit List

It is used to display the bit address condition in PLC. The row header of the list indicates the byte part of the address and the eight columns that followed indicate the condition of bits. If a column is marked with 'X', it means the corresponding bit address is occupied. For example, "A0000._", the last column of the row, i.e. the column headed with "0", is marked with "X", indicating that address A0000.0 is occupied. Please note that not all the bit addresses are list unless one of the bits is occupied. If an address is not listed, it means the address is not used.

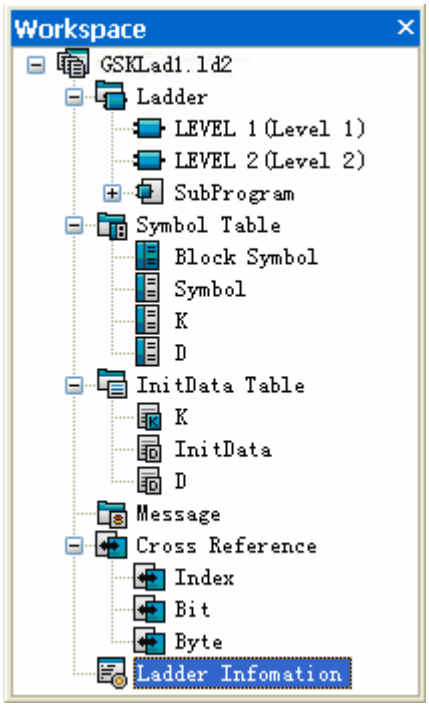
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| A0000._ | X | X | X | X | X | X | X | X |
| A0001._ | | X | X | X | X | X | X | X |
| A0002._ | X | X | X | X | X | | | X |
| F0000._ | | X | X | X | | | | |
| F0001._ | | | X | | X | | X | X |
| F0002._ | X | | | | X | | | |
| F0003._ | | X | X | | X | X | X | X |

#### 4.4.6.3 Byte List

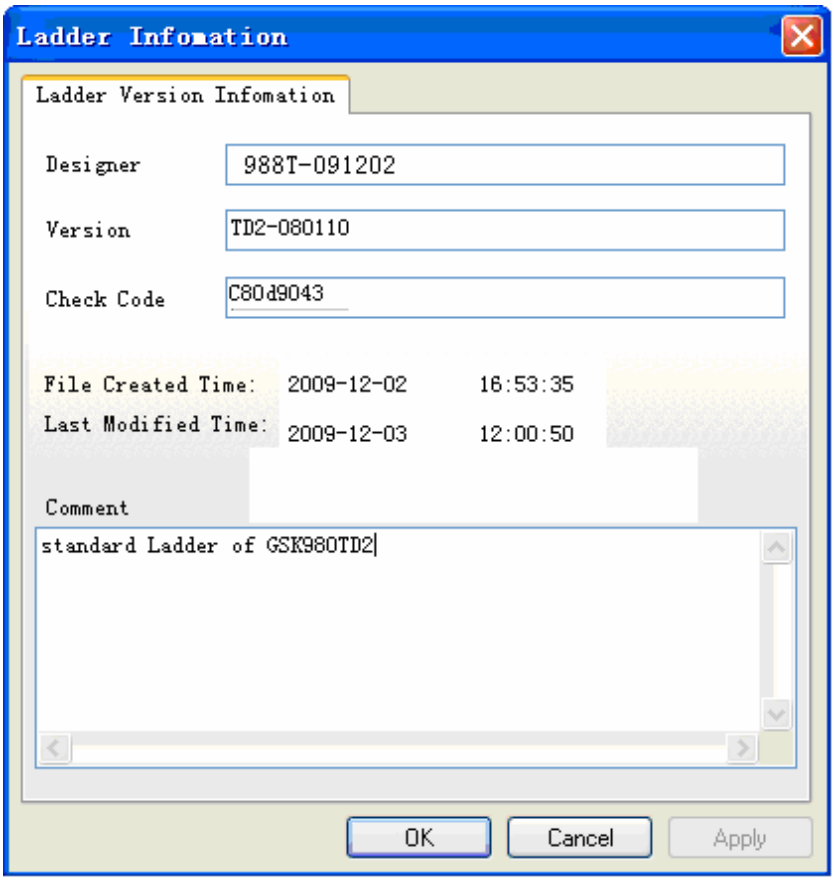
It is used to display the byte address condition in PLC. The row header of the list indicates the part aside from single digits and the ten columns that followed indicate single digits. For example,"C000_", the column header of the row is 9, indicating that the cell represents address C0009. An occupied cell is marked with "X". Please note that not all the addresses are list unless it is occupied. If an address is not listed, it means the address is not used.

| 字 节 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|---|---|---|---|---|---|---|---|---|---|---|
| C000_ | | | | | X | | | | | |
| F001_ | X | | | | | | | | | |
| F002_ | | | X | | | | X | | | |
| F020_ | | | | | | | | X | | |
| F021_ | | | | | | X | | | | |
| G001_ | X | | X | | | | | | | |
| G003_ | X | | X | | | | | | | |
| G004_ | | | | X | | | | | | |

### 4.4.7 Ladder Information




Double-click [Ladder Information] on the workspace tree node, or right-click it, then click command [Open], a dialogue box will pop up (see the following figure). You can enter the information in "Designer", "Version" and "Comment". The input format is not restricted but the character number is limited (63 characters in "Designer", 19 characters in "Version", 511 characters in "Comment"). The Check Code is the 32-bit CRC checksum of PLC file, and it can be seen only when the project is not modified or is saved after modification.


Ladder Infomation
Ladder Version Infomation
Designer
988T-091202
Version
TD2-080110
Check Code
C80d9043
File Created Time:
2009-12-02
16:53:35
Last Modified Time:
2009-12-03
12:00:50
Comment
standard Ladder of GSK980TD2
OK
Cancel
Apply

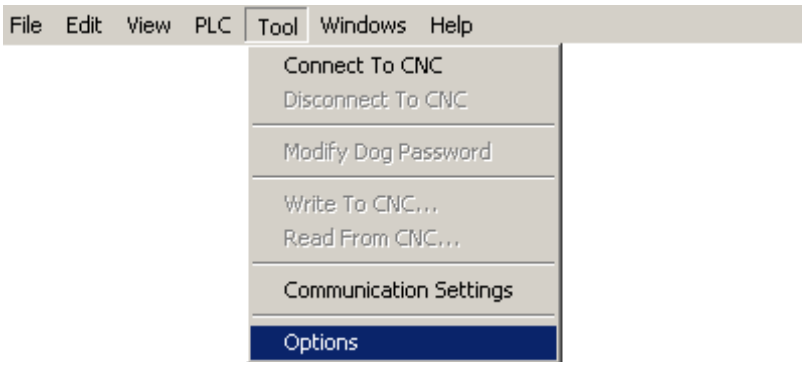
# CHAPTER FIVE COMMUNICATION

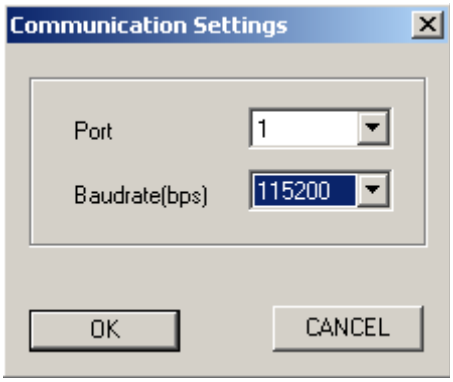
GSKLadder communicates with CNC by RS232, during which ladder of current project and parameters can be

transmitted to CNC or ladder and parameters can be read from CNC.

Rightly connect COMM port of PC with CNC communication interface, and set the corresponding parameters of GSKLadder. Baud rate should be set same as that of CNC as follows:

Select [Tool]—[Communication Settings] to pop-up a dialog box, and set the port and baud rate. Start the next operation after confirming.



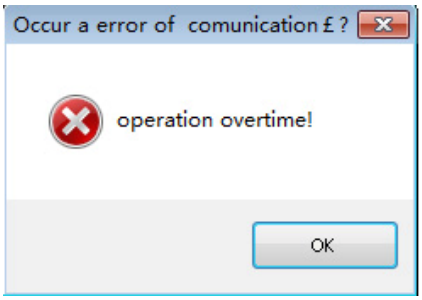




GSKLadder adopts the method that the communication is immediately transited while connecting. Namely, it is disconnected between GSKLadder and CNC when the transmission does not perform. It is only connected based upon the demand of transmission; the connection can be automatically cut off after the transmission is completed. Therefore, it is enough to perform the “transmit to CNC” or “transmit from CNC” when transmitting.

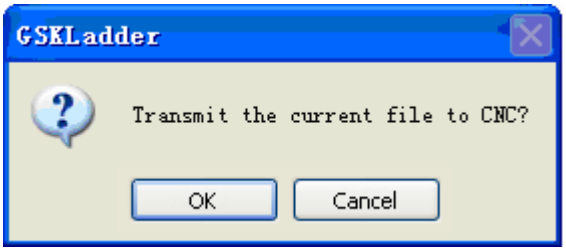
## 5.1 Transmit File to CNC

Perform the menu “transit to CNC” or press the button at the fundamental tool column. There are several seconds responseless state may occur during connection. The prompt “operation overtime”

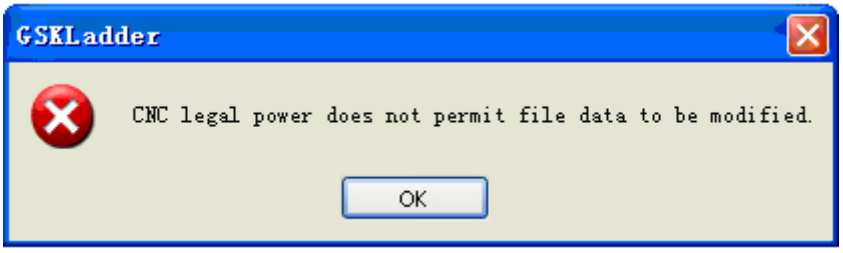
may occur after waiting for a while, which means that the connection is failed. In this moment, it is necessary to detect the hardware connection and communication setting, etc, and then perform the transition command after the elimination is removed.



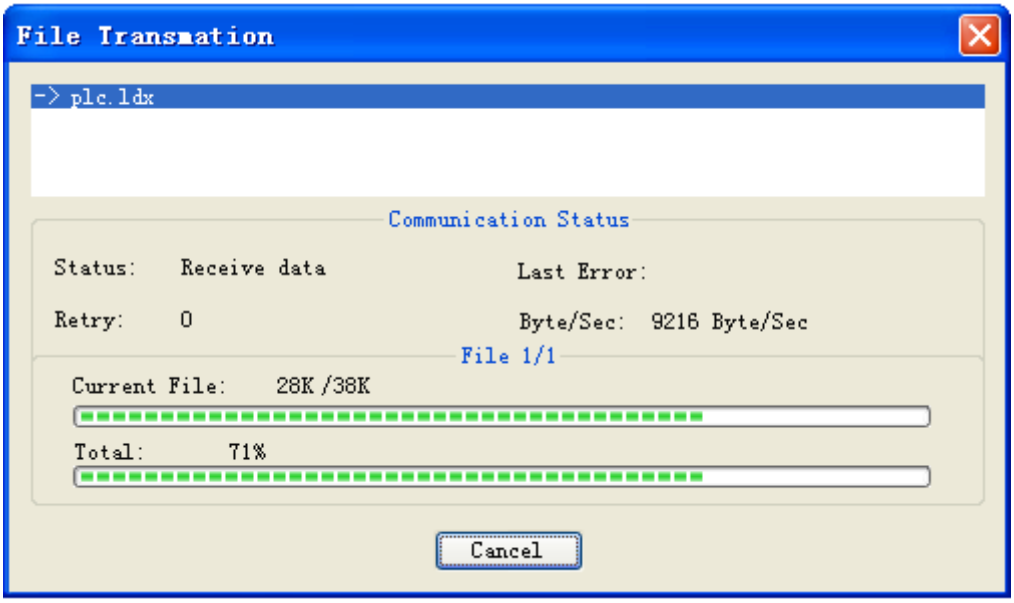

The inquiry dialog frame “Upload or not” may appear if the connection is succeeded, the file transits by [YES]. If the receiver CNC of the file is the system that does not support the current file format of the 980D, etc; the GSKLadder will then convert the format of the current project file, and then upload it. In this case, it is necessary to ensure the ladder diagram is correct, and out of the conetent from the former system; for example, output branch, new command, etc.




The setting of CNC user level should be more than 2 levels when the file is being transmited, otherwise, the prompt “CNC authority prohibits to modify the file data” occurs.

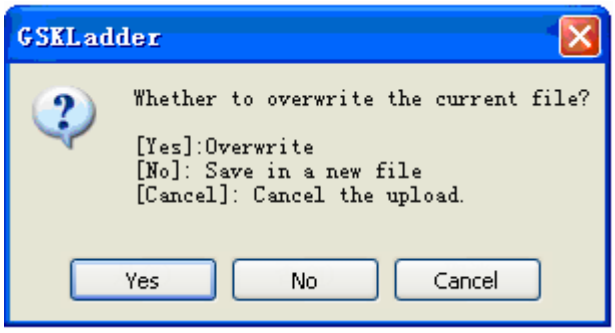



Spring out the file transition dialog after the transition is performed. The communication state and transition schedule can be viewed from the dialog; the selected file in the edit frame is the being transited file name at the current situation. User can intermit the transition by the “CANCEL” button.




## 5.2 File Read from CNC

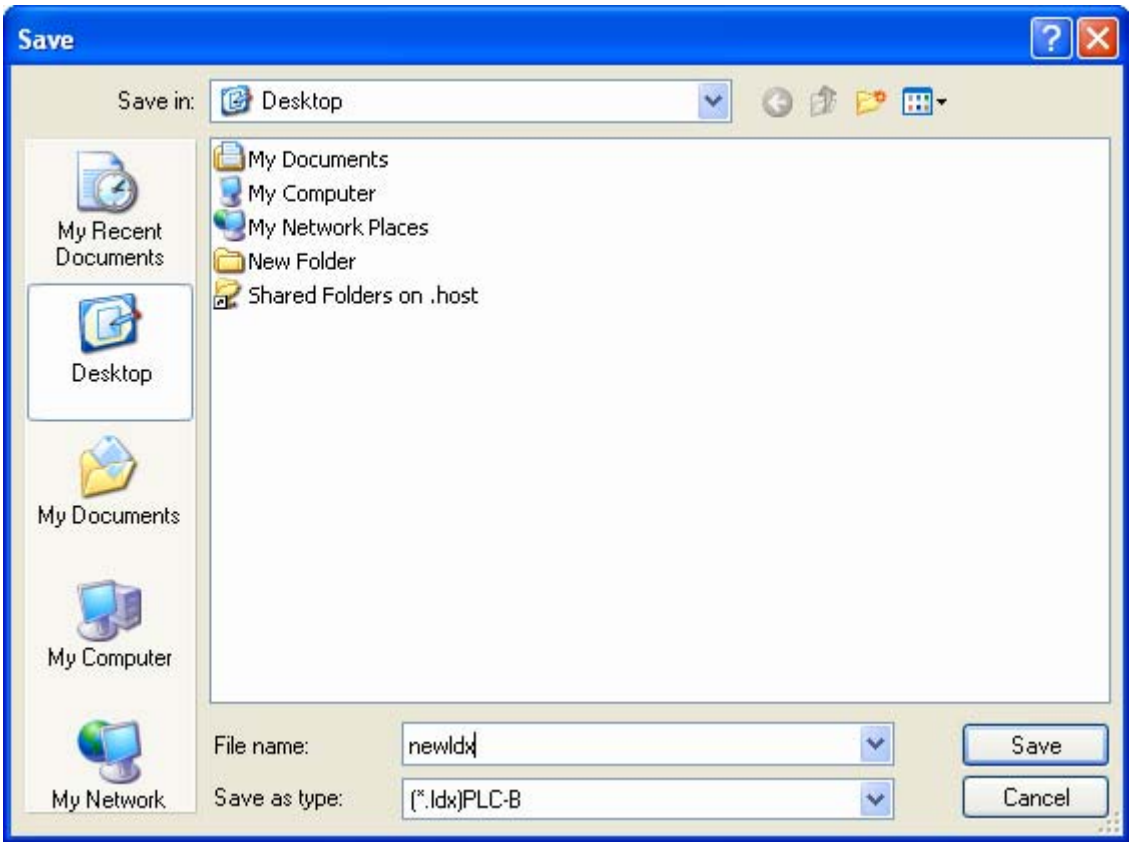
The situation, similarly, such as “It can not be connected” may occur when performing the menu “Introduction from CNC” or control the button  of the fundamental tool column. However, there is no authority limit from introducing by CNC, and the download can be performed as long as the connection is correct.




The inquiry window may occur after connecting, and then the save method file read can be selected:

**1) Cover the current project:** It is better to select this method after the file is correctly received; the current project file will be covered by the introduced file and then open it; the overall modifications by the current project will be abandoned;

**2) New project:** The file dialog frame from a new project will be shown by selecting this method; it receives after the project is saved. Close the opened project at the current situation after correctly receiving, and then open the new project.


Save
Save in:
Desktop
My Documents
My Computer
My Network Places
New Folder
Shared Folders on .host
My Recent Documents
Desktop
My Documents
My Computer
My Network
File name:
newldx
Save as type:
(*.ldx)PLC-B
Save
Cancel

# IV Appendix

Appendix

# APPENDIX ONE G SIGNAL LIST

| Address | Signal | Symbol | Section |
|---|---|---|---|
| G4.3 | Auxiliary function completion signal | FIN | |
| G5.6 | Miscellaneous function lock signal | AFL | |
| G6.4 | Override cancel signal | OVC | |
| G7.2 | Cycle start signal | ST | |
| G8.4 | Emergency stop signal | ESP | |
| G8.5 | Feed dwell signal | SP | |
| G8.6 | Resetting and cursor return signal | RRW | |
| G8.7 | External reset signal | ERS | |
| G10，G11 | Manual rapid traverse override signal | JV0～JV15 | |
| G12 | Feedrate override signal | FV0～FV7 | |
| G14.0， G14.1 | Rapid traverse override signal | ROV1, ROV2 | |
| G18.0～G18.3 | MPG1 feed axis selection signal | HA～HD | |
| G19.4，G19.5 | MPG/STEP override signal | MP1, MP2 | |
| G19.7 | Manual rapid traverse selection | RT | |
| G27.0 | The 1st spindle selection signal | SSW1 | |
| G27.1 | The 2nd spindle selection signal | SSW2 | |
| G27.3 | The 1st spindle stop signal | SSTP1 | |
| G27.4 | The 2nd spindle stop signal | SSTP2 | |
| G27.7 | Spindle contour control switch signal | CON | |
| G28.1，G28.2 | Gear selection signal | GR1，GR2 | |
| G29.0，G29.1 | The 2nd spindle gear selection | GR21，GR22 | |
| G29.4 | Spindle speed arrival signal | SAR | |
| G29.6 | Spindle stop signal | KSSTP | |
| G30 | Spindle override signal | SOV0～SOV7 | |
| G32.0～G32.7<br>G33.0～G33.3 | Spindle motor speed command signal inputted from PLC | R01I～R12I | |
| G33.7 | Spindle motor speed selection command signal | SIND | |
| G34.0～G34.7<br>G35.0～G35.3 | PLC inputs the 2nd spindle motor speed command signal | R01I2～R12I2 | |

| Address | Signal | Symbol | Section |
|---|---|---|---|
| G35.7 | The 2nd spindle motor speed selection command signal | SIND2 | |
| G39.0～G39.5 | Tool compensation number selection number | OFN0~OFN5 | |
| G39.6 | Workpiece coordinate system offset value write selection signal | WOQSM | |
| G39.7 | Tool offset write selection signal | GOQSM | |
| G40.6 | Absolute coordinate position record signal | PRC | |
| G40.7 | Workpiece coordinate offset value write signal | WOSET | |
| G43.0～G43.2、G43.5、G43.7 | Mode selection signal | MD1、MD2、MD4、DNC1、ZRN | |
| G44.0 | Optional block skip signal | BDT | |
| G44.1 | All-axis machine lock signal | MlK | |
| G44.7 | Manual sequence tool-change signal | HDT | |
| G46.1 | Single block signal | SBK | |
| G46.3 | Memory protection signal | KEY1 | |
| G46.7 | Dry run signal | DRN | |
| G47 | Tool group number selection signal | TL01～TL128 | |
| G48.5 | Tool skip signal | TLSKP | |
| G48.7 | Tool change resetting signal | TLRST | |
| G54，G55 | Macro input signal | UI01～UI15 | |
| G61.0 | Rigid tapping signal | RGTAP | |
| G61.4，G61.5 | Rigid tapping spindle selection | RGTSP1，RGTSP2 | |
| G67.2 | MPG try cutting method inspection signal | MMOD | |
| G74.4 | Spindle CW signal of rigid tapping | SFRB | |
| G74.5 | Spindle CCW signal of rigid tapping | SRVB | |
| G100.0～G100.4 | Feed axis and direction selection signal | +J1～+J5 | |
| G102.0～G102.4 | | -J1～-J5 | |
| G114.0～ | Overtravel signal | +L1～+L5 | |

| Address | Signal | Symbol | Section |
|---|---|---|---|
| G114.4 | | | |
| G116.0 ～ G116.4 | | -L1～-L5 | |
| G127.7 | The 2nd spindle Cs outline control shifting signal | CON2 | |
| G136.0 ～ G136.4 | PLC control axis selection signal | EAX1～EAX5 | |
| G141.0 ～ G141.3 | PLC axis control is absolute/relative coordinate | EABSA~EABSD | |
| G142.0 | PLC Miscellaneous Function Completion Signal | EFINA | |
| G154.0 | | EFINB | |
| G166.0 | | EFINC | |
| G178.0 | | EFIND | |
| G142.2 | PLC The Buffer Invalid Signal | EMBUFA | |
| G154.2 | | EMBUFB | |
| G166.2 | | EMBUFC | |
| G178.2 | | EMBUFD | |
| G142.3 | PLC The Program Stop Signal | ESBKA | |
| G154.3 | | ESBKB | |
| G166.3 | | ESBKC | |
| G178.3 | | ESBKD | |
| G142.4 | PLC The Servo Cut-off Signal | ESOFA | |
| G154.4 | | ESOFB | |
| G166.4 | | ESOFC | |
| G178.4 | | ESOFD | |
| G142.5 | The PLC Axis Control Dwell Signal | ESTPA | |
| G154.5 | | ESTPB | |
| G166.5 | | ESTPC | |
| G178.5 | | ESTPD | |
| G142.6 | The PLC axis Resetting Signal | ECLRA | |
| G154.6 | | ECLRB | |
| G166.6 | | ECLRC | |
| G178.6 | | ECLRD | |
| G142.7 | The PLC Axis Control Command Reading Signal | EBUFA | |
| G154.7 | | EBUFB | |

| Address | Signal | Symbol | Section |
|---|---|---|---|
| G166.7 | | EBUFC | |
| G178.7 | | EBUFD | |
| G143.7 | PLC the Program Stop Invalid Signal | EMSBKA | |
| G155.7 | | EMSBKB | |
| G167.7 | | EMSBKC | |
| G179.7 | | EMSBKD | |
| G143.0~G143.6 | PLC The Axis Control Command Signal | EC0A～EC6A | |
| G155.0~G155.6 | | EC0B～EC6B | |
| G167.0~G167.6 | | EC0C～EC6C | |
| G179.0~G179.6 | | EC0D～EC6D | |
| G144～G145 | PLC The Axis Control Feedrate Signal | EIF0A ～ EIF15A | |
| G156～G157 | | EIF0B ～ EIF15B | |
| G168～G169 | | EIF0C ～ EIF15C | |
| G180～G181 | | EIF0D ～ EIF15D | |
| G146～G149 | PLC The Axis Control Data Signal | EID00A ～ EID31A | |
| G158～G161 | | EID00B ～ EID31B | |
| G170～G173 | | EID00C ～ EID31C | |
| G182～G185 | | EID00D ～ EID31D | |
| G150.0 ～ G150.1 | PLC The Rapid Traverse Rate Override Signal | ROV1E ～ ROV2E | |
| G151 | PLC The Feedrate Override Signal | FV0E～FV7E | |
| G200.0 | Spindle JOG function signal | SPD | |
| G201～G204 | Current tool-position signal | NT00～NT31 | |

# APPENDIX TWO F SIGNAL LIST

| Add. | Signal Name | Symbol | Chapter |
|---|---|---|---|
| F0.0 | State signal return of resetting and cursor | RWD | |
| F0.4 | Feed dwell signal | SPL | |
| F0.5 | Cycle start signal | STL | |
| F0.6 | Servo ready signal | SA | |
| F0.7 | Auto running signal | OP | |
| F1.0 | Alarm signal | AL | |
| F1.1 | Reset signal | RST | |
| F1.3 | Distribution completion signal | DEN | |
| F1.4 | Spindle enable signal | ENB | |
| F1.5 | Tapping signal | TAP | |
| F1.7 | CNC ready signal | MA | |
| F2.0 | Inch input signal | INCH | |
| F2.1 | Rapid traverse signal | RPDO | |
| F2.2 | Constant surface speed signal | CSS | |
| F2.3 | Thread cutting signal | THRD | |
| F2.6 | Cutting feed signal | CUT | |
| F2.7 | Dry run check signal | MDRN | |
| F3.0 | STEP mode check signal | MINC | |
| F3.1 | MPG mode check signal | MH | |
| F3.2 | MANUAL mode check signal | MJ | |
| F3.3 | MDI mode check signal | MMDI | |
| F3.5 | AUTO mode check signal | MMEM | |
| F3.6 | EDIT mode check signal | MEDT | |
| F4.0 | Optional block skip check signal | MBDT1 | |
| F4.1 | All-axes machine lock check signal | MMLK | |
| F4.2 | Manual absolute check signal | MABSM | |
| F4.3 | Single block check signal | MSBK | |
| F4.4 | Miscellaneous function lock check signal | MAFL | |
| F4.5 | Machine zero return mode check | MREF | |
| F4.6 | Program zero return detection signal | MPST | |
| F7.0 | Miscellaneous function strobe signal | MF | |
| F7.2 | Spindle speed function strobe signal | SF | |
| F7.3 | Tool function strobe signal | TF | |

| Add. | Signal Name | Symbol | Chapter |
|---|---|---|---|
| F9.4 | Decode M signal | DM30 | |
| F9.5 | | DM02 | |
| F9.6 | | DM01 | |
| F9.7 | | DM00 | |
| F10～F13 | Miscellaneous function code signal | MB00～MB31 | |
| F22～F25 | Spindle speed code signal | SB00～SB31 | |
| F26～F29 | Tool function code signal | TB00～TB31 | |
| F36.0 ～ F37.3 | The 1st spindle S12-digit signal | R01O～R12O | |
| F38.2 | The 2nd spindle enable signal | ENB2 | |
| F40F41 | The 1st spindle actual speed signal | AR00～AR15 | |
| F44.1 | Spindle outline control shifting end signal | FSCSL | |
| F54～F59 | Macro output signal | U000～U131 | |
| F64.0 | Tool used up signal of cutter life span administration | TLCH | |
| F64.1 | New tool selection signal of cutter life span administration | TLNW | |
| F65.0 | Spindle CW in rigid tapping | RGSPP | |
| F65.1 | Spindle CCW in rigid tapping | RGSPM | |
| F76.3 | Rigid tapping method signal | RTAP | |
| F91.3 | MPG inspection method signal | MMMOD | |
| F94.0 ～ F94.4 | Machine zero return end signal | ZP1～ZP5 | |
| F96.0 ～ F96.4 | The 2nd reference point return end signal | ZP21～ZP25 | |
| F98.0 ～ F98.4 | The 3rd reference point return end signal | ZP31～ZP35 | |
| F100.0 ～ F100.4 | The 4th reference point return end signal | ZP41～ZP45 | |
| F102.0 ～ F102.4 | Axis moving signal | MV1～MV5 | |
| F106.0 ～ F106.4 | Axis moving direction signal | MVD1～MVD5 | |
| F120.0 ～ F120.4 | Reference point setting signal | ZRF1～ZRF5 | |

| Add. | Signal Name | Symbol | Chapter |
|---|---|---|---|
| F127.0 ～ F127.4 | Axis enabled signal | ABLE1 ～ ABLE5 | |
| F129.7 | PLC controlled axis selection status signal | KEAXSL | |
| F130.3 | PLC The Miscellaneous Function Execution Signal | EDENA | |
| F133.3 | | EDENB | |
| F136.3 | | EDENC | |
| F139.3 | | EDEND | |
| F130.4 | PLCmiscellaneous function execution signal | EGENA | |
| F133.4 | | EGENB | |
| F136.4 | | EGENC | |
| F139.4 | | EGEND | |
| F130.7 | Axis control command read completed signal | EBSYA | |
| F133.7 | | EBSYB | |
| F136.7 | | EBSYC | |
| F139.7 | | EBSYD | |
| F131.0 | PLC miscellaneous function 2 strobe signal | EMFA | |
| F134.0 | | EMFB | |
| F137.0 | | EMFC | |
| F140.0 | | EMFD | |
| F131.1 | PLC buffer full signal | EABUFA | |
| F134.1 | | EABUFB | |
| F137.1 | | EABUFC | |
| F140.1 | | EABUFD | |
| F132，F142 | PLC Miscellaneous function codes signal | EM11A ～ EM48A | |
| F135，F145 | | EM11B ～ EM48B | |
| F138，F148 | | EM11C ～ EM48C | |
| F141，F151 | | EM11D ～ EM48D | |
| F144.1 | The 2nd spindle Cs outline control shifting end signal | FSCSL2 | |
| F182.0 ～ | PLC The Control signal | EACNT1 ～ | |

| Add. | Signal Name | Symbol | Chapter |
|---|---|---|---|
| F182.4 | | EACNT5 | |
| F198.0 ～ F198.4 | End signal of program reference point return | PR01～PR05 | |
| F199.0 | Spindle jog detection signal | MSPHD | |
| F200.4 | Analog spindle is valid | SIMSPL | |
| F207 | Overall tool-position number | TMAX | |
| F212.2 | G04 command is being performed | CDWL | |
| F212.4 | Override of 0% | C0VL | |
| F214.0 ～ F214.4 | Feed axis direction selection | VAL1～VAL5 | |
| F215.0 ～ F216.3 | Spindle gear-shifting speed command signal | SGR00 ～ SGR11 | |
| F236.0 ～ F237.3 | S12-bit code signal | R01O2 ～ R12O2 | |
| F240 ～ F241 | Spindle input speed command value | INP00～INP15 | |