

|   |          |
|---|----------|
| <b>1 INTRODUCTION .....</b>   | <b>4</b> |
| <b>2 CONTROLLER'S TECHNICAL INDEX: .....</b>                        | <b>4</b> |
| <b>3 CONTROLLER OPERATIONAL PANEL .....</b>                         | <b>5</b> |
| 3.1 KEYBOARD INDEX .....  | 5        |
| 3.2 LED INDECATOR .....   | 5        |
| <b>4 OPERATION .....</b>  | <b>6</b> |
| 4.1 OPERATION INDICATION.....                                       | 6        |
| 4.2 FUNCTIONS SELECTION .....                                       | 6        |
| 4.3 EDIT FUNCTION.....  | 6        |
| 4.3.1 HOW TO USE THE EDIT BUTTON IN EDIT .....                      | 6        |
| 4.3.2 ESTABLISH A NEW FILE.....                                     | 7        |
| 4.3.3 FILE SELECTION .....  | 7        |
| 4.3.4 FILE DELETION .....   | 7        |
| 4.3.5 FILES NUMBERS .....   | 7        |
| 4.3.6 FILE EDITION.....   | 7        |
| 4.3.6.1 FILE'S CONTENT INPUT .....                                  | 8        |
| 4.3.6.2 INSET ROW, FIELDS, DIGIT IN THE FILE .....                  | 8        |
| 4.3.6.3 DELETE FILE ROW, FIELD, DIGIT .....                         | 9        |
| 4.3.6.4 FILE ROW, FIELD, DIGIT CONSULT .....                        | 9        |
| 4.4 PARAMETER FUNCTION .....  | 9        |
| 4.4.1 PARAMETERS OF THE CONTROLLER .....                            | 9        |
| 4.4.2 PARAMETER CONSULT .....                                       | 11       |
| 4.4.3 PARAMETER MODIFICATION.....                                   | 11       |
| 4.4.3.1 P0 parameter .....  | 11       |
| 4.4.3.2 P3、 P4 PARAMETERS.....                                      | 11       |
| 4.4.3.3 OTHER .....   | 12       |
| 4.5 JOG, SINGLE AND RETURN TO MECHANICAL ORIGIN BY HAND.....        | 12       |
| 4.5.1 MANUAL JOG (MOVE) .....                                       | 12       |
| 4.5.2 MANUAL SINGLE (MOVE).....                                     | 12       |
| 4.5.3 MANUAL RESET .....  | 12       |
| 4.5.4 SWITCH BETWEEN MANUAL JOG, MANUAL SINGLE ANDMANUAL RESET .... | 13       |
| 4.5.5 OTHER ASSISTANT FUNCTIONS .....                               | 13       |
| 4.5.5.1 RESET TO THE PROGRAM ORIGIN.....                            | 13       |
| 4.5.5.2 STEP SELECTION .....  | 13       |

|  |    |
|--|----|
| 4.5.5.3 FEED SPEED AND MULTIPLE .....                  | 13 |
| 4.5.5.4 CONTROL OF SPEED MULTIPLES ON MOVING.....      | 14 |
| 4.5.5.5 PAUSE ON MOVING .....                          | 15 |
| 4.5.5.6 COORDINATES' DISPLAY CLEARING .....            | 15 |
| 4.5.5.7 M FUNCTIONS .....                              | 15 |
| 4.6 AUTOMATIC RUNNING FUNCTION .....                   | 15 |
| 4.6.1 PERFORMED PROGRAM SELECTION .....                | 15 |
| 4.6.2 USER PROGRAM RUNNING .....                       | 16 |
| 4.6.2.1 SERIES RUNNING AND SINGLE RUNNING.....         | 16 |
| 4.6.2.2 SINGLE RUNNING STOP SELECTION .....            | 16 |
| 4.6.2.3 FEED HOLDING IN MOVING .....                   | 16 |
| 4.6.2.4 PROGRAM RUNNING IN MACHINE LOCKED .....        | 16 |
| 4.6.2.5 COORDINATES AND SEGMENTS NUMBER DISPLAYS ..... | 16 |
| 4.6.2.6 MULTIPLE SELECTION .....                       | 16 |
| 4.6.2.7 RETURN TO PROGRAM ORIGIN .....                 | 17 |
| 4.6.2.8 OTHER CONTROL FUNCTIONS .....                  | 17 |
| 4.7 SYSTEM INITIALIZATION .....                        | 17 |
| 5 PROGRAMMING .....                                    | 17 |
| 5.1 PROGRAMMING GENERAL .....                          | 17 |
| 5.1.1 AXIS DEFINITION .....                            | 17 |
| 5.1.2 MECHANICAL ORIGIN .....                          | 17 |
| 5.1.3 PROGRAM ORIGIN .....                             | 17 |
| 5.1.4 PROGRAMMING COORDINATES .....                    | 18 |
| 5.1.4.1 ABSOLUTE COORDINATE .....                      | 18 |
| 5.1.4.2 RELATIVE COORDINATE .....                      | 18 |
| 5.1.4.3 THE UNIT AND SCALE OF THE COORDINATE .....     | 18 |
| 5.1.4.4 INITIALIZAION STATE AND MODE STATE .....       | 18 |
| 5.2 THE CONSTRUCTIONS OF THE PROGRAM AND SEGMENT ..... | 18 |
| 5.2.1 SEGMENT .....                                    | 18 |
| 5.2.2 MAIN PROGRAM AND SUBPROGRAM .....                | 19 |
| 5.3 PREPARATION FUNCTION—G FUNCTION .....              | 20 |
| 5.3.1 G00 POINT POSITIONING WITH FAST SPEED .....      | 20 |
| 5.3.2 G01 POSITIONING WITH FEED SPEED .....            | 21 |
| 5.3.3 G04 DELAY .....                                  | 21 |

|   |    |
|---|----|
| 5.3.4 G27 RETURN TO REFERENCE ORIGIN .....    | 21 |
| 5.3.5 G50 COORDINATE SYSTEM ENACTMENT .....   | 21 |
| 5.3.6 G73 FIX CYCLE OF PECK DRILLING .....    | 21 |
| 5.3.7 G81 DRILLING CYCLE .....                | 22 |
| 5.3.8 G83 DEEP HOLE DRILLING CYCLE .....      | 22 |
| 5.3.9 G22 G80 PROGRAM PART CYCLE .....        | 23 |
| 5.4 ASSISTANT FUNCTION....M FUNCTION .....    | 24 |
| 5.4.1 M00 PAUSE .....                         | 24 |
| 5.4.2 M02 PROGRAM OVER .....                  | 25 |
| 5.4.3 M30 PROGRAM CYCLE RUNNING .....         | 25 |
| 5.4.4 M03 M04 M05 MAIN SPINDLE CONTROL .....  | 25 |
| 5.4.5 M08 M09 COOLANT CONTROL .....           | 25 |
| 5.4.6 M97 TRANSFER WITHOUT CONDITION .....    | 25 |
| 5.4.7 M98 SUBPROGRAM TRANSFER .....           | 25 |
| 5.4.9 M99 SUBPROGRAM ESCAPE .....             | 26 |
| 5.4.10 USER OUTPUT .....                      | 26 |
| 5.4.11 USER INPUT .....                       | 26 |
| 5.5 PROGRAMING PRINCIPLES .....               | 26 |
| 6 ALARM LIST .....                            | 27 |
| 7 DIAGNOSIS .....                             | 27 |
| 7.1 MOTOR INTERFACE DIGNOSIS .....            | 27 |
| 7.2 DIAGNOSIS OF SIGNAL INPUT INTERFACE ..... | 28 |
| 7.3 DIAGNOSIS OF SIGNAL INPUT INTERFACE ..... | 28 |

# GSK991 MONOAXIS CONTROLLER

## OPERATION MANUAL

### 1 INTRODUCTION

GSK991 monoaxis controller, adopts high performance CPU and SPLD as control part with LED display, button operation panel. It is suitable for different step motor driver and servomotor driver. With programming, it can be used to position, control in multipath logical operation. And it can be used as a PLC controller with locate loop, or a control terminal fitting in with epigyny controller.















### 2 CONTROLLER'S TECHNICAL INDEX:

1. Controllable axes: 1 axis, extend to 3 axes
2. Mini enactment unit: 0.01mm
3. Travel: .9999.99~+9999.9mm
4. Fast move speed: 8000mm/min
5. User program capability: 24kB
6. Storage user program numbers: 100
7. Input/output points: input 6 points (photoelectricity insulation), output 7 points( each output point can be controlled by program)
8. Single arrange LED display
9. Auto shift control
10. Standard ISO code segment, relative/ absolute programming
11. Soft/hard overtravel protection, driver alarm check

### 3 CONTROLLER OPERATIONAL PANEL

Re. above figure

#### 3.1 KEYBOARD INDEX

|   |                       |  |
|---|-----------------------|--|
|    | RESET                 | Controller reset   |
|    | ESCAPE                | Escape from local function modular to superior modular   |
| Letters   | ADDRESS               | Program dictate, data, add   |
| 0 . 9   | FIGURE                | Program dictates data, parameter data  |
| Enter   | ENTER                 | Input or selection confirmation  |
| ↑ ↓ ← →   | CURSOR                | Main function switch, move the cursor at editing,<br>move the direct control by hand                       |
|    | MAIN SPINDLE          | Main spindle run   |
|    | MAIN SPINDLE          | Main spindle stop  |
|    | COOLANT               | Coolant on/off   |
|    | BACK REFERENCE ORIGIN | Return to program beginner   |
|    | START                 | Start machining program to auto run  |
|   | MACHINE LOCK/INSET    | Inset a segment in program modular,<br>lock machine in auto module   |
|  | MULTIPLE/DELETE       | Delete a character, a block or a segment in edit module,<br>select speed multiple by hand                  |
|  | FEED.HOLD             | Pause of running   |
|  | SINGLE/FAST FEED      | Single/continuum switch, single stop<br>at continually running, or fast move/ feed state switch<br>by hand |
|  | STEP.SELECT           | Select step at single feed   |

#### 3.2 LED INDECATOR

The 7 LED No. LED1~LED7 indicators on the controller show the different states of this controller as follows:

|      |  |
|------|--|
| LED1 | Driver's state indicate  |
| LED2 | Bright at JOG move, flash at waiting for run   |
| LED3 | Single step/Single segment indicate, bright on single step by hand, or on continuous run at auto                         |
| LED4 | Return mechanical origin/machine lock indicate, bright on returning mechanical origin by hand or locking machine at auto |
| LED5 | Fast move/feed indicate, bright at fast move, dark at feed state   |
| LED6 | Main spindle indicate, bright at spindle run, dark at spindle stop   |
| LED7 | Coolant indicate, bright at coolant pump run, dark at coolant pump stop  |

## 4 OPERATION

### 4.1 OPERATION INDICATION

This controller has 7 functions and their corresponding LED displays as follows:

**E** EDIT, translate and edit user machining program

**P** PARAMETER, setup the parameters of this controller

**J** JOG, Jog control

**S** MANUAL SINGLE STEP, Manually control with enacted step

**Z** MANUAL RESET, Manually turn back to mechanical zero

**A** AUTO RUN, automatically run the user program

**C** CONTROLLER INITIALIZATION, Initialize the parameters and user programs of this controller

### 4.2 FUNCTIONS SELECTION

**4.2.1** '991' will be displayed on the LED of this controller after power on, then it will enter the menu of the functions selection if any button is pressed. It can be back to the '991' display when you press 'Esc' button on the menu of the functions selection.

**4.2.2** The 'E' prompt of the edit function will be displayed if it is the first time to enter the menu of the functions selection. When it escapes from the other functions menu, the corresponding prompt will be displayed.

Press '↑' button, select the last function

Press '↓' button, select the next function

Press 'Enter' button to confirm the selected function, and enter the corresponding function to finish the corresponding function selection.

Press 'Esc', turn back to '991' display

### 4.3 EDIT FUNCTION

Enter the edit function by pressing enter 'Enter' button when function prompt 'E' is flashing on the display. It can establish, edit, amend, and delete the user programs on this menu.

File structure: One user program is called a file named with letter 'O' + two digits. All user programs are managed as files.

#### 4.3.1 HOW TO USE THE EDIT BUTTON IN EDIT

Press '↑' button to move the cursor to the beginning of last row. Press '↓' to move the cursor to the beginning of the next row. Press '←' to move the cursor to the left digit. Press

‘→’ to move the cursor to the right digit. Press ‘Del’ button to delete a digit, a field or a row. Press ‘Ins’ to inset a row.

### 4.3.2 ESTABLISH A NEW FILE

When enter the edit state, if there is no user program in the storage, the controller displays ‘ERR.14’ . Press any button to begin to edit user program with the first file name ‘000’ .

If there have the files in the user program storage, the first file name (serial No.) will be displayed.

Press ‘Ins’ button, the controller displays ‘0’ and wait for input any two digits between ‘00~99’ . It will be waiting again after press ‘←’ or ‘Esc’ button if you want to input again. Press ‘Enter’ button to confirm your correct input, then it displays ‘ERR.’ to appear that is a new file. It displays the first row number and waits for inputting the program contents when press any button.

If there have the file with input name, the controller will display the first row to be edited.

### 4.3.3 FILE SELECTION

Two methods to select the existent file:

Method 1 Press ‘Ins’ button, and input the file name (two digits) to be edited, press ‘Enter’ again to confirm, and then the file can be edited.

Method 2 Press ‘↓’ or ‘↑’ button to select the file in storage. Press ‘↑’ button to display the name of the front file, press ‘↓’ button to display the file of the next file. Press ‘Enter’ again to confirm, and then the file can be edited.

### 4.3.4 FILE DELETION

Press ‘↓’ or ‘↑’ button to select the file name to be deleted, then press ‘Del’ button up to flashing ‘Y’ to prompt if it will need to be deleted.

Please press ‘Enter’ button to delete the file, or else, press any other button to keep the file.

### 4.3.5 FILES NUMBERS

Press ‘→’ button, the files numbers, ‘T.’ and two digits, appear on the LED display. Press ‘Esc’ to escape.

### 4.3.6 FILE EDITION

Select the file name need to edit, press ‘Enter’ to confirm, and begin to edit the file content. Every row of the program begins with the row numbers, ‘N’ + 3 digits, and the follows are the dictate and data. The row numbers is built automatically by the controller,

the others is input by keyboard. The controller insets the row numbers automatically in the conditions as follows:

1. The first row of new file
2. Press 'Enter' button at the end of each row for editing the new file
3. Press 'Enter' button at the end of any file
4. Press 'Ins' button to inset a row

#### 4.3.6.1 FILE'S CONTENT INPUT

##### **NEW FILE INPUT**

After setup a new file, the controller build the first row number automatically, input the file contents in turn. Press 'Enter' button to end the each row input, the controller build the next row number automatically. Repeat above steps up to the last row with pressing 'Esc' button to finish input.

#### 4.3.6.2 INSET ROW, FIELDS, DIGIT IN THE FILE

##### **INSET ROW**

Move the cursor (the flashing character) to any position in the row to inset a new row following this row with '←', '↑', '→', '↓' button. Press 'Ins' button to inset a new row between the row and the next row, and the controller build a new row number automatically, and then can input the new row's contents. Press 'Enter' button to end the row inset.

Repeat above steps to inset more rows

If the inset row is the last row in this file, when press 'Enter' button, the controller build a new row number automatically. Press 'Esc' button to cancel the new row number and end inset.

Note: No need to pay attention to value of the row numbers. The controller will arrange from small to big. Notice the changes of the corresponding row numbers, when run the dictate M97 (move) and M98 (transfer). If the row numbers changed, please amend the corresponding data.

##### **INSET FIELD**

Move the cursor to the row that needs to inset field by pressing '↑', '↓', '→', '←' button. Directly input the letters and data of the new field behind the cursor.

##### **INSET DIGIT**

Move the cursor to the position to need inset digits by pressing '↑', '↓', '→', '←' button, input the digits to inset the digit behind the cursor.



#### 4.3.6.3 DELETE FILE ROW, FIELD, DIGIT

All deletions of the file row, field, and digit are finished with pressing the 'Del' button.

Delete the content at the position of the cursor by pressing 'Del' button.

The digit is deleted by pressing 'Del' button when the cursor flashes at this digit.

Delete the whole field (letters and digits) by pressing 'Del' button when the cursor flashes at these letters except for letter 'N'.

Delete the whole row by pressing 'Del' button when the cursor flashes at the letter 'N'.

#### 4.3.6.4 FILE ROW, FIELD, DIGIT CONSULT

Consult the file row, field, and digit by pressing '↑' '↓' '→' '←' buttons and 'Enter' button.

'↑' button lets the cursor move to the front row, and the controller displays the front row number. The cursor doesn't move if the local row is the first one.

'↓' button lets the cursor move to the next row, and the controller displays the next row number. The cursor doesn't move if the local row is the last one.

'←' button lets the cursor move to the front bit. If the local bit is letter, the controller displays the front field content, and the cursor moves to the last character of the front field.

'←' button lets the cursor move to the next bit. If the local bit is the last bit of field, the controller displays the first character of the next field.

'Enter' button lets the cursor move to the next field, so it is used to consult the whole contents of the file.

**Note: Press 'Enter' button to inset a new field when the cursor is at the last field of the file.**

### 4.4 PARAMETER FUNCTION

When the prompt 'P' flashes on LED, press 'Enter' button to enter parameter function.

The parameter function is used to query or amend the parameters of the controller.

After entering the parameter function, the controller displays the present parameter 'P0', and the cursor flashes at the '0', and wait to input.

#### 4.4.1 PARAMETERS OF THE CONTROLLER

The controller has been set with 10 parameters from 0 to 9. Each parameter is indicated by

‘P’ + parameter number (0~9) with the following instruction:

| Parameter number | Range                          | Parameter name               | Instructions   | Initial values  |
|------------------|--------------------------------|------------------------------|--|-----------------|
| <b>P0</b>        | <b>0000000<br/>~ 1111111</b>   | <b>Bit parameter</b>         | <b>Selection of various control modes(total 7 bits)</b>  | <b>0000000</b>  |
| <b>P1</b>        | <b>0 ~ 2.55</b>                | <b>Reverse clearance</b>     | <b>Reverse clearance of thread<br/>Unit: mm</b>  | <b>0</b>        |
| <b>P2</b>        | <b>0 ~ 255</b>                 | <b>Pulse width of M code</b> | <b>Pulse width of M code when M code is pulse modem<br/>Unit: 50ms</b>                                       | <b>20</b>       |
| <b>P3</b>        | <b>0% ~ 150%,<br/>16 steps</b> | <b>Feed multiplier</b>       | <b>Speed multiplier of manual feed and G01 feed</b>  | <b>100%</b>     |
| <b>P4</b>        | <b>25% ~ 100%<br/>4 steps</b>  | <b>Quick move multiplier</b> | <b>Speed multiplier of manual quick move and G00 quick move</b>  | <b>100%</b>     |
| <b>P5</b>        | <b>0 ~ 6000</b>                | <b>Quick move speed</b>      | <b>Quick move speed<br/>Unit: mm/min</b>   | <b>6000</b>     |
| <b>P7</b>        | <b>0 ~ 65535</b>               | <b>Setup running times</b>   | <b>Setup the running times of the program. Program stops running when running times reach the set value.</b> | <b>0</b>        |
| <b>P6</b>        | <b>0 ~ 65535</b>               | <b>Times to have run</b>     | <b>Times the program has run</b>   | <b>0</b>        |
| <b>P8</b>        | <b>0 ~ 9999.99</b>             | <b>Positive limit</b>        | <b>Positive travel limit<br/>Unit: mm</b>  | <b>9999.99</b>  |
| <b>P9</b>        | <b>.9999.99 ~ 0</b>            | <b>Negative limit</b>        | <b>Negative travel limit<br/>Unit: mm</b>  | <b>.9999.99</b> |

#### Instruction of the parameter P0:

Parameter P0 is bit parameter, it was set as 0 or 1 for different bits, and thus the controller is in the different control modes. There are 7 bits (D6~D0) from left to right:

- D0   0   The running direction of motor is the same as the set direction
- 1   The running direction of motor is the reverse of the set direction
- D1   0   M code is level output
- 1   M code is pulse output

- D2 0 The direction to return to the mechanical origin is positive  
1 The direction to return to the mechanical origin is negative
- D3 0 The program doesn't time  
1 The program times
- D4 0 High level alarm of driver  
1 Low level alarm of driver
- D5~D6 Standby

#### 4.4.2 PARAMETER CONSULT

After enter parameter mode, the controller displays parameter 'P0', and the cursor flashes under the '0' to wait for input

Press '↑' button to select the front parameter round from P9 to P0

Press '↓' button to select the next parameter round from P0 to P9

Press '→' button to display the contents of the current parameter

Press 'Enter' button to confirm the perfected contents of current parameter

Press 'Esc' button to escape from parameter function to function menu

#### 4.4.3 PARAMETER MODIFICATION

Press '↑', '↓' buttons to select the parameter number in need to modify, then press 'Enter' button to enter modification state.

After input the contents of the parameter, press 'Enter' button to confirm, or press 'Esc' button to delete the inputted contents with 'Err.16' display. After press any button, input new contents. The front bit is deleted by pressing '←' button when input new content.

##### 4.4.3.1 P0 parameter

Parameter P0 is bit parameter, each bit is not any other digit but only '0' or '1'. It is allowed only to input the effective bit. Press 'enter' button to confirm, and the current contents are displayed.

e.g. If P0 parameter is 0001011, it is allowed inputting '1011' and 'Enter'.

##### 4.4.3.2 P3、P4 PARAMETERS

P3, P4 parameters are multiple parameters whose grades were set off.

Press '↑' button each time to reduce one grade. If the current grade is the lowest one, press it to jump to the highest grade.

Press '↓' button each time to increase one grade. If the current grade is the highest one, press it to jump to the lowest grade.

Press 'Esc' button to escape from the parameter modification state to the parameter

selection state.

#### 4.4.3.3 OTHER

Select the parameter number to be modified by pressing ‘↑’ , ‘↓’ and ‘Enter’ buttons.

Input the contents of parameter and press ‘Enter’ button to confirm, the controller accepts the contents and displays them. Then it escapes to the selection state.

NOTE: Only the virtual contents of the parameters can be inputted.

### 4.5 JOG, SINGLE AND RETURN TO MECHANICAL ORIGIN BY HAND

Manual JOG, manual Single, and manual Return to mechanical origin functions control the move of the axis and other assistant functions by pressing the buttons by hand. The other operations are same as that.

#### 4.5.1 MANUAL JOG (MOVE)

By pressing ‘Enter’ button when the control displays ‘J’ prompt, the controller enter the Manual Jog function, and displays ‘J’ and current coordinates.

When LED2 lights, press ‘→’ button to move the motor to positive direction ( the coordinates increase), the motor stops as soon as the button is released; press ‘←’ button to move the motor to negative direction (the coordinates reduce), the motor stop as soon as the button is released.

#### 4.5.2 MANUAL SINGLE (MOVE)

When the controller prompts ‘S’ , it can enter manual single function by pressing ‘Enter’ button, displays ‘S’ and the current coordinate, and the LED3 is lighting.

When press ‘→’ button one time, the motor move one step along the positive direction with set step until the travel is zero; when press ‘←’ button one time, the moor move one step along the negative direction with set step until the travel is zero.

#### 4.5.3 MANUAL RESET

When the controller prompts ‘Z’ , it can enter manual reset function y pressing ‘Enter’ button, displays ‘Z’ and the current coordinates, and the LED4 is lighting.

It is set by D2 bit (reset direction) of P0 parameter, the controller performs reset mechanical origin function. The motor doesn’t move when the direction button isn’t set by D2 bit.


When D2=0, the reset direction is positive. When D2=1, the reset direction is negative. The controller rests by pressing the enactment button one time, the motor moves along corresponding direction with the fastest speed, after the speed.down switch is touched, it moves with the enactment speed inside the controller. When it receive the reference signal from exterior, the motor stops at the mechanical origin, at the same time, the coordinates becomes zero.

#### 4.5.4 SWITCH BETWEEN MANUAL JOG, MANUAL SINGLE AND MANUAL RESET


Switches between these three manual functions can be finished directly by pressing the 'F' button without escaping to the function selection mode. When pressing 'F' button, the switch sequence is J.S.Z.J.

#### 4.5.5 OTHER ASSISTANT FUNCTIONS

##### 4.5.5.1 RESET TO THE PROGRAM ORIGIN

The program origin is the coordinates' origin that was defined in the user program by G50 dictate. It can be finished by pressing button in the three manual functions. By pressing the button  one time, the motor moves to the program origin with the fastest speed.

##### 4.5.5.2 STEP SELECTION

It defines the step at single move. By pressing  button to displays the 'L' and the current step.

By pressing '↑' button, the step reduces by degrees. By pressing '↓' button, the step increases by degrees.



The steps inside the controller are as follows:

0.01 0.05 0.1 1.0 5.0 10.0 100 500 mm

##### 4.5.5.3 FEED SPEED AND MULTIPLE

There have two speed of manual feed to be selected:

1. the fastest speed move...fast speed
2. the enactment speed move...feed speed

Switch between the fast speed and feed speed can be finished by pressing  button. When pressing  button one time, the controller switches one time.

When the fast speed is selected, LED5 is lighting, move speed and shifting.up/down are equal to the fast feed (G00) in program dictate.

Fast move multiples are 4 grades as follows:

25% 50% 75% 100%

**current fast move speed = fastest speed defined by parameter P5 \* fast move multiple**


Selected feed speed, LED5 isn't lighting, then the actual move speed is defined by the enactment speed inside the controller and .feed multiple.

Feed multiples are 16 grades from 0% to 150% (distance: 10%) as follows'

| multiple ( % ) | Feed speed ( mm/min ) |
|----------------|-----------------------|
| <b>0</b>       | <b>0</b>              |
| <b>10</b>      | <b>8.4</b>            |
| <b>20</b>      | <b>12.6</b>           |
| <b>30</b>      | <b>20</b>             |
| <b>40</b>      | <b>32</b>             |
| <b>50</b>      | <b>50</b>             |
| <b>60</b>      | <b>79</b>             |
| <b>70</b>      | <b>123</b>            |
| <b>80</b>      | <b>200</b>            |
| <b>90</b>      | <b>280</b>            |
| <b>100</b>     | <b>350</b>            |
| <b>110</b>     | <b>400</b>            |
| <b>120</b>     | <b>600</b>            |
| <b>130</b>     | <b>850</b>            |
| <b>140</b>     | <b>1000</b>           |
| <b>150</b>     | <b>1260</b>           |

**Note: Don't move when the multiple is zero.**

Select the multiple of the fast speed and feed speed:

Press  button to let the controller display 'R ×××'. If current state is fast speed (LED5 lighting), there is the fast multiple (25%~100%, 4 grades) following prompt 'R' ; IF current state is feed speed (LED unlighted), there is the feed multiple (0~150%, 16 grades) following prompt 'R' .

Press '↑' button one time to reduce a grade. If current grade is the lowest one, it jumps to the largest grade.

Press '↓' button one time to increase a grade. I current grade is the largest one, it jumps to the lowest grade.


Press 'Esc' button to escape from the multiple selection.

#### 4.5.5.4 CONTROL OF SPEED MULTIPLES ON MOVING

Only when the motor is moving with feed speed, the multiple can be changed. It can't be changed when the motor is moving with fast speed.

On feeding, adjust the feed speed by pressing ‘↑’ ‘↓’ buttons. Press ‘↑’ button each time to reduce one grade up to 0 multiple (stop). Press ‘↓’ button each time to increase one grade up to 150% multiple.


#### 4.5.5.5 PAUSE ON MOVING


The motor can stop on moving. As soon as press  button (feed hold), the motor speeds down and stops at once.


#### 4.5.5.6 COORDINATES’ DISPLAY CLEARING

Clear current coordinates by pressing ‘Z’ button.

#### 4.5.5.7 M FUNCTIONS

Press  button, the controller outputs M03, the main spindle runs positively, LED6 is lighting.

Press  button, the controller outputs M05, the main spindle stop, LED6 is unlighted.

Press  button, the controller outputs M08/M09, the coolant pump starts/stops. And the switch between start and stop is in effect in controller is running. When coolant pump is running, the LED7 is lighting. When coolant pump stops, LED7 is unlighted.

Press ‘G’ button, the controller output M04

**Note: When D1=1 (main spindle control: pulse) in parameter P0, there hasn’t M04 output but there have M03 and M05 output.**

### 4.6 AUTOMATIC RUNNING FUNCTION

This function lets the controller finish the tasks arranged by the user program. When it displays prompt ‘A’, enter this function by pressing ‘Enter’ button. Escape from this function by pressing ‘Esc’ button.


#### 4.6.1 PERFROMED PROGRAM SELECTION


As soon as enter the auto function, the controller checks if there have the user program automatically. If there haven’t the user program in the storage of the controller, the controller displays ‘Err.14’, and escape from auto function to edit function automatically by pressing any button. If there have the program in the storage of the controller, the controller displays the current program name ‘A.××’. If this isn’t the program to be run automatically, it can be changed by pressing ‘↑’ ‘↓’ buttons. Press ‘↑’ button to select the front program or press ‘↓’ button to select the next program. When the controller displays the program name to be run, the controller explains and checks this program automatically by pressing ‘Enter’ button. If the user program is wrong, the controller displays the wrong row number and wrong type (the wrong types appear in the following chart). If the controller confirms the user program,


it display prompt 'A' and current coordinates.

## 4.6.2 USER PROGRAM RUNNING



### 4.6.2.1 SERIES RUNNING AND SINGLE RUNNING

Single running Press  button one time to run a segment of the user program.


Series running Press  button one time to run all segments of the user program.


Switch between series running and series running is finished by pressing  button. When it is in series running state, LED3 is lighting. When it is in single running, LED is unlighted.

### 4.6.2.2 SINGLE RUNNING STOP SELECTION

In series running, press  button to select single stop. After running up current program, it pauses. Press  button to start the next segments.


### 4.6.2.3 FEED HOLDING IN MOVING

The controller holds feed by pressing  button, the motor speeds down and stops, at the same time, LED2 flashes to wait for inputting next dictate.

If press 'Esc' button then, it escapes from feed holding, and doesn't exist the spare length. If press  button, the motor starts to run from the paused situation. Pressing other button are not in effect.

### 4.6.2.4 PROGRAM RUNNING IN MACHINE LOCKED

When the machine is locked, the controller runs inside with out any outside action (motors, relays and so on).

Switch between running and machine locked states by pressing  button. In machine locked state, LED4 is unlighted. In normal running state, LED4 is lighting.


### 4.6.2.5 COORDINATES AND SEGMENTS NUMBER DISPLAYS

LED display can appear the current coordinates or the current segment number according to actual requirements. Press '←' button to display current segment number, press '→' button to display current coordinates

### 4.6.2.6 MULTIPLE SELECTION

Only the feed multiple can be selected in auto mode. The fast multiple can be adjusted in manual mode, parameter mode.


Select multiple before running:

After press  button, the controller displays the current multiple 'Rxxx' 'xxx'. Press '↑' button to reduce a grade, press '↓' button to increase a grade, press 'Esc' to




escape from the multiple selection. In program running, the feed multiple can be adjusted by pressing ‘↑’ ‘↓’ buttons, and the adjustment is in effect in actual running.

#### 4.6.2.7 RETURN TO PROGRAM ORIGIN


The program origin is defined by dictate G50 in the program. The motor returns from current position to program origin with the fastest speed by pressing  button.

#### 4.6.2.8 OTHER CONTROL FUNCTIONS

Before start auto running, the controller can control the start/stop of main spindle and coolant pump by pressing corresponding button.

Press  button for main spindle to run positively, LED5 is lighting.

Press  button for main spindle to stop, LED5 is unlighted

Press  button for coolant pump to switch between start/stop. LED6 is lighting when coolant pump is running, LED6 is unlighted when coolant pump stop.

**Note: start/stop of coolant pump can be switch by manual in running.**

### 4.7 SYSTEM INITIALIZATION

System initialization resets the inner data, control function, and deletes the all of user data and program in the user program storage and running data storage.

When the controller displays the prompt ‘C’, Press ‘Enter’ button to appear prompt ‘Y’ on the controller display, then if press ‘Enter’ button again to confirm initialization. Or not, press any other button to undo initialization.

## 5 PROGRAMMING

### 5.1 PROGRAMMING GENERAL

#### 5.1.1 AXIS DEFINITION

This controller can control one axis defined as Z axis, and the direction of this axis is defined same as mathematical axis, the left is negative direction, and the coordinate reduces to left, the right is positive direction, and the coordinate increases to right.

#### 5.1.2 MECHANICAL ORIGIN

Mechanical origin is a fix point on the controlled machine. Only with the return to the mechanical origin in manual function, the machine can be return to this point accurately.

#### 5.1.3 PROGRAM ORIGIN

Controller regards the start point of the machine in auto running as a reference point to be defined as the programming coordinates origin.

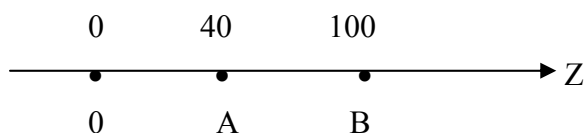
User program defines the coordinate with indicate G50, it is float coordinate.

### 5.1.4 PROGRAMMING COORDINATES

This controller can program with absolute coordinates (Z field) or relative coordinate (Z field).

#### 5.1.4.1 ABSOLUTE COORDINATE

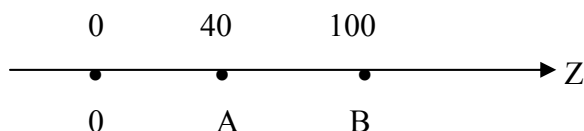
The distance to the coordinate origin is the absolute coordinate.



The dictate of moving from A to B is Z100, the dictate of moving from B to A is Z40

#### 5.1.4.2 RELATIVE COORDINATE

The distance from the front point to local point is relative coordinate.



The dictate of moving from A to B is W60. The dictate of moving from B to A is W.60.

#### 5.1.4.3 THE UNIT AND SCALE OF THE COORDINATE

Min unit of this controller is 0.01mm, programmable scale is .9999.99 ~ +9999.99.

#### 5.1.4.4 INITIALIZAION STATE AND MODE STATE

Initialization state is the programming state before running the user program. Mode state is the state that a corresponding field is in effect before defined again by other segment. Or say, the mode state is that the defined field can be repeated in the following segments without defined again.

## 5.2 THE CONSTRUCTIONS OF THE PROGRAM AND SEGMENT

For controller to control the machine to finish an action, a group of dictates must be endowed to the controller as a segment. The whole of segments to finish a series of actions make up of a program, or called as a file. The order number for identification is entitled the file name.

### 5.2.1 SEGMENT

Each segment consists of some fields as follows

**N3 G2 M2 Z(W)±4.2 F4 D2.2 R2.2 ENTER**

**N3** — Segment number, 3 bits

**G2** — Preparing function, 2 bits

**M2** — Assistant function, 2 bits

**Z ( W ) ±4.2** — Move size, Max. ±4 bits integer and 2 bits decimal

**F4** — Speed function, 4 bits integer, unit: mm/min,  
The target segment number in transfer dictate,  
Cycle times in dictate G22

**D2.2** — Feed per time in fix cycle, 2 bits integer and 2 bits decimal, unit: mm

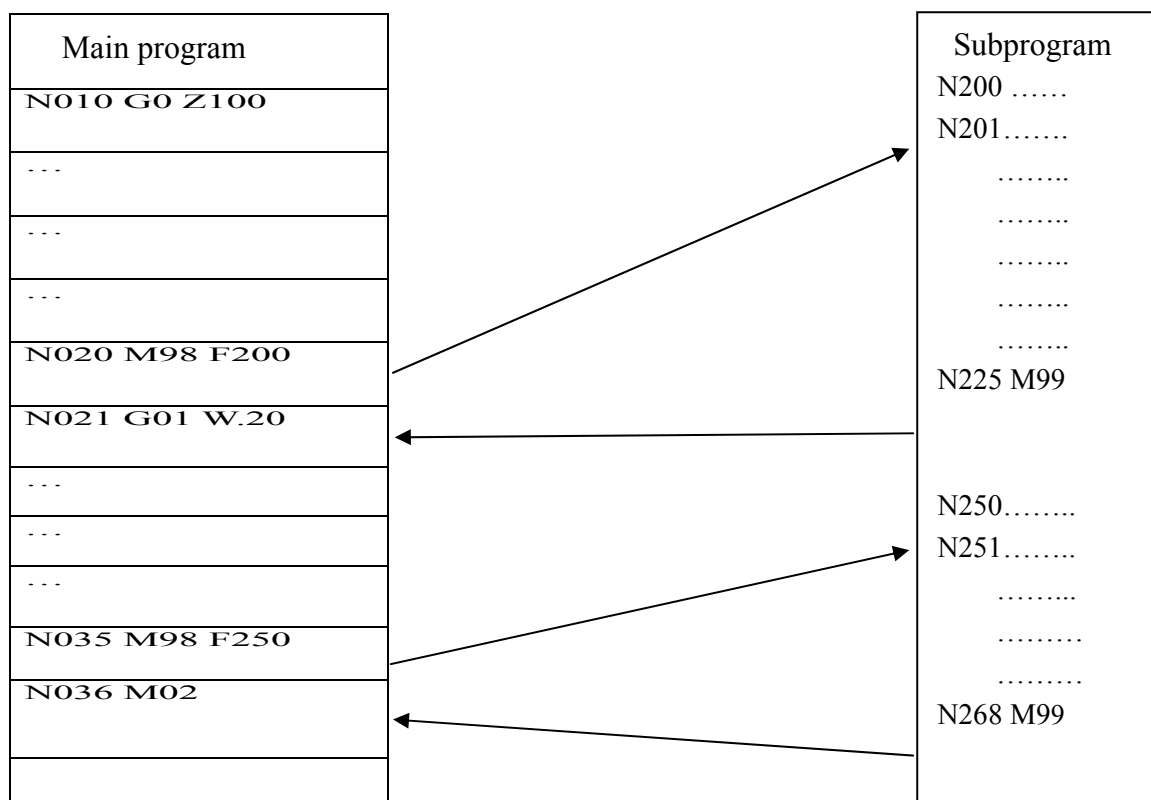
**R2.2** — Untread value in fix cycle, 2 bits integer and 2 bits decimal, unit: mm  
Dictate delay in dictate G04  
Dictate pitch in dictate G84

**ENTER** — End prompt of segment, after finishing input of the segment, press 'Enter' button to end without display

### 5.2.2 MAIN PROGRAM AND SUBPROGRAM

The main program is for the controller to finish the full series functions. If the controller needs to run the same function in different period of main program, we can edit a group of segments for these functions separately as a subprogram. If there is transfer dictate in the running main program, the controller runs the subprogram. After finishing the subprogram, the controller returns to run the main program continuously.

The transfer relationship between main program and subprogram is as follows.



Transfer sketch between main program and subprogram

### 5.3 PREPARATION FUNCTION—G FUNCTION

G function consists of G code and the following 2 digits. The 2 digits define the move contrail of this segment. If the first one is zero, it can be omitted.

For example: the G00 can be written as G0, the G01 can be written as G1

This controller's G codes are as follows:

| Serial No. | G code | Function                          | Mode | Programming format  | Remark                 |
|------------|--------|-----------------------------------|------|---------------------|------------------------|
| 1          | G00    | Point positioning with fast speed | *    | G00 Z ( W )         | Inside appointed speed |
| 2          | G01    | Positioning with feed speed       | *    | G01 Z ( W ) F       | Speed appointed with F |
| 3          | G04    | Delay                             |      | G04 R               |                        |
| 4          | G27    | Return to reference origin        |      | G27                 | Fastest speed          |
| 5          | G50    | Define coordinates                |      | G50 Z               | No move                |
| 6          | G73    | Peck drilling cycle               | *    | G73 Z ( W ) D R F   | Speed appointed with F |
| 7          | G81    | Drilling cycle                    | *    | G81 Z ( W ) R F     | Speed appointed with F |
| 8          | G83    | Deep.hole drilling cycle          | *    | G83 Z ( W ) D R F   | Speed appointed with F |
| 9          | G22    | Start the program part cycle      |      | G22 F               |                        |
| 10         | G80    | End the program part cycle        |      | G80                 |                        |
| 11         | G93    | System offset                     |      | G93 X ( U ) Z ( W ) |                        |

Note: the dictate with \* is mode.

#### 5.3.1 G00 POINT POSITIONING WITH FAST SPEED

**Dictate format: G00 Z ( W )**

Z defines the absolute coordinate of the end point where the axis moves to. W defines the direction and distance that the current axis moves.

G00 fast positioning, the controlled object moves to the appointed position with the fastest speed inside the controller.

The fastest speed is controlled by multiple parameters such that it's of no effect appointing the fastest speed with dictate F.

Dictate G00 is the initial state of the controller. If the G code isn't defined when the controller powers on, the G00 is the default dictate of the controller.

### 5.3.2 G01 POSITIONING WITH FEED SPEED

**Dictate format: G01 Z ( W ) . F**

The controller enables the controlled object to move to the position appointed by Z (W) with the speed defined with dictate F.

Dictate F is in mode state. Unit: mm/min

### 5.3.3 G04 DELAY

**Dictate format: G04 R.**

Pause to run the next segment, the pause time is set with dictate R. Unit: second

### 5.3.4 G27 RETURN TO REFERENCE ORIGIN

**Dictate format: G27**

The controller enables the controlled object to return quickly to the reference origin where the program began to run (it is the work piece coordinate origin set by G50)

### 5.3.5 G50 COORDINATE SYSTEM ENACTMENT

**Dictate format: G50 Z —**

The work piece coordinate origin can be set with this dictate. It decides that the controller defines the position of absolute coordinate dictate with the distance to this coordinate origin.

**Note:** Before each program adopts the dictate G, the coordinate origin must be appointed with dictate G50.

### 5.3.6 G73 FIX CYCLE OF PECK DRILLING

**Dictate format: G73 Z ( W ) D R F**

Z ( W ) — the absolute/relative coordinate of the drilling end

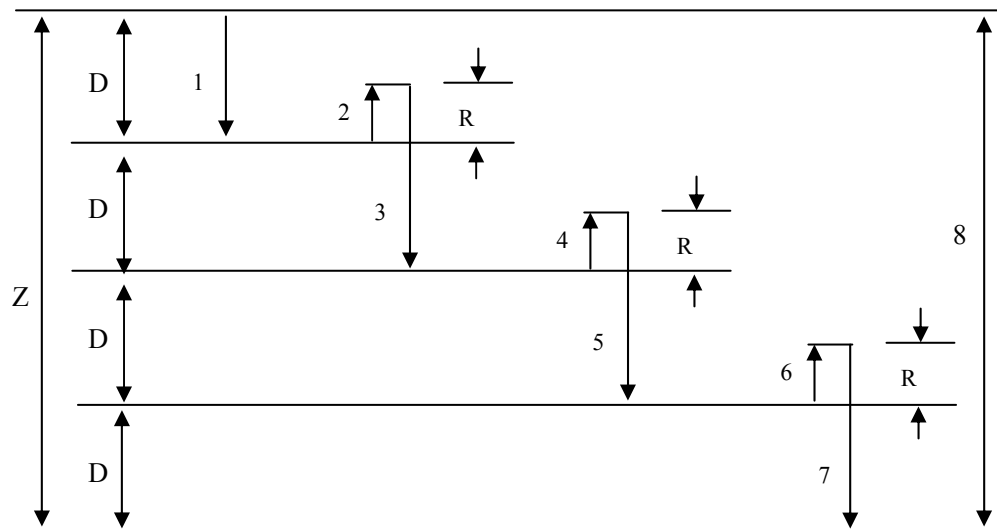
D — Feed of each drilling

R — Withdrawal of each drilling

F — Feed speed

**CYCLE PROCESS:**

1. Feed D length with F speed
2. Withdraw R length quickly
3. Feed R+D length with F speed
4. Repeat above 2, 3 steps up to reach the position appointed by dictate Z (W)
5. Withdraw to the start quickly.

**G73 FIX CYCLE OF PECK DRILLING****5.3.7 G81 DRILLING CYCLE****Dictate format: G81 Z (W) R F**

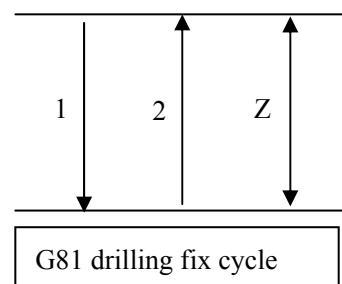
Z (W) — the absolute/relative coordinate of the drilling end

R — the pause time at the hole bottom, unit: second

F — feed speed

**CYCLE PROCESS:**

- 1、 Feed to the Z position with F speed
- 2、 Pause time appointed with dictate R
- 3、 Return to the start

**5.3.8 G83 DEEP HOLE DRILLING CYCLE****Dictate format: G83 Z (W) D R F**

Z (W) — the absolute/relative coordinate of the drilling end

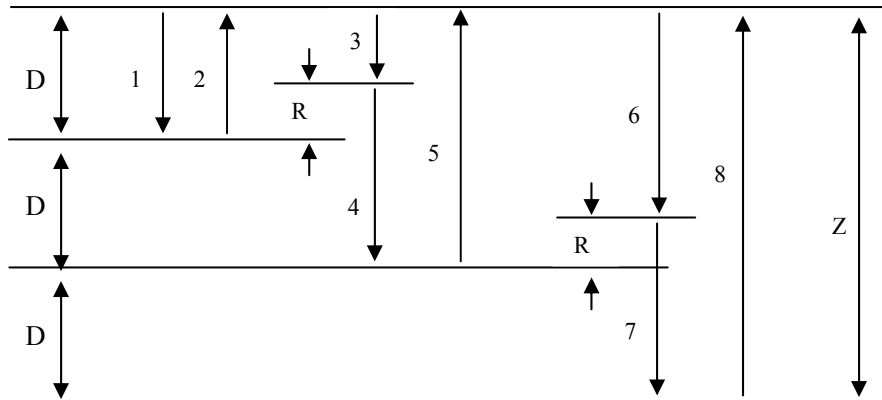
D — feed of each drilling

R — the distance away to the last drilling end

F — feed speed

**CYCLE PROCESS:**

1. Feed d length with speed F
2. Return to the start
3. Fast feed  $i \cdot D \cdot R$  length ( i is the times of feed)
4. Feed (D+R) length with feed speed
5. Repeat above 2, 3, 4 steps up to reach the end position appointed by dictate Z (W)
6. Fast return to the start

**G83 deep hole drilling fix cycle**

**Note:** 1, All G73, G81, G83 fix cycles are in mode. The front dictate is always in effect before the dictate is appointed with new data.

e.g. **N011 G73 Z100 D3.5 R1.2 F400**

**N012 W.5**

**N013 W.3**

After run N011, N012 enables the controlled object to feed W.5 with speed F first, then to run the N010 fix cycle. N013 enables the controlled object to feed W.3 with speed F, then run the N010 fix cycle.

2、When running the fix cycle, the controlled object pauses after running each step, if the 'Single Stop' is selected.

**5.3.9 G22 G80 PROGRAM PART CYCLE**

**Dictate format:** **G22 F**

**G80**

F — cycle times 01 ~ 99, F=1 in default

**CYCLE PROCESS:**

1. Run the program behind G22
2. Run G80 cycle, the cycle times F minus 1. If F is 0, then run the next program. If F isn't 0,

then return to run the program after G22

**Note:** Dictate G22 must be in use with dictate G80 in the same time, and the dictate G22 can't repeat in use, and this cycle is only with relative coordinate.

## 5.4 ASSISTANT FUNCTION...M FUNCTION

M function consists of M code and the following 2 digits. Controller outputs the corresponding control signal, or checks the signal from the controlled object. If the first bit is 0, it can be omitted.

M functions of this controller list as follows:

| Serial No. | M code | Function   | Format | Remark                                |
|------------|--------|--|--------|---------------------------------------|
| 1          | M00    | Pause  | M00    |                                       |
| 2          | M02    | Program over   | M02    |                                       |
| 3          | M30    | Program over, return to the first segment to run continually | M30    |                                       |
| 4          | M03    | Main spindle obverse rotation                                | M03    | M03 M04 interlock                     |
| 5          | M04    | Main spindle reverse rotation                                | M04    |                                       |
| 6          | M05    | Main spindle stop  | M05    |                                       |
| 7          | M08    | Coolant start  | M08    |                                       |
| 8          | M09    | Coolant stop   | M09    |                                       |
| 9          | M97    | Transfer without condition                                   | M97 F  | F is the corresponding program number |
| 10         | M98    | Subprogram transfer  | M98 F  |                                       |
| 11         | M99    | Return from subprogram                                       | M99    |                                       |
| 12         | M21    | 1# valid user output   | M21    |                                       |
| 13         | M22    | 1# invalid user output                                       | M22    |                                       |
| 14         | M23    | 2# valid user output   | M23    |                                       |
| 15         | M24    | 2# invalid user output                                       | M24    |                                       |
| 16         | M81    | 1# valid user input waiting                                  | M81 R  | R delay to canceling signal           |
| 17         | M82    | 1# invalid user input waiting                                | M82 R  |                                       |
| 18         | M83    | 2# valid user input waiting                                  | M83 R  | R delay to canceling signal           |
| 19         | M84    | 2# invalid user input waiting                                | M84 R  |                                       |

### 5.4.1 M00 PAUSE

Dictate format: M00



The controller pauses running to wait for intervention.

Press 'Esc' button to end running

Press any other button to go on

## 5.4.2 M02 PROGRAM OVER

**Dictate format: M02**

Each program must have an ending dictate; this dictate can end the program and return to the beginner of this program.

## 5.4.3 M30 PROGRAM CYCLE RUNNING

**Dictate format: M30**

This dictate can end this program running, return to the beginner of the program and repeat running this program.

## 5.4.4 M03 M04 M05 MAIN SPINDLE CONTROL

**Dictate format: M03**

**M04**

**M05**

M03 — main spindle obverse rotation M04 — main spindle reverse rotation M05 — main spindle stop

Note: M03, M04 are interlocking signal. When M signal is level output, the output pin corresponding to M05 doesn't output.

## 5.4.5 M08 M09 COOLANT CONTROL

**Dictate format: M08**

**M09**

M08 — coolant pump start M09 — coolant stop

## 5.4.6 M97 TRANSFER WITHOUT CONDITION

**Dictate format: M97 F**

F—The number of the target Segment of transfer

## 5.4.7 M98 SUBPROGRAM TRANSFER

**Dictate format: M98 F**

F— the first segment number of the subprogram

## 5.4.9 M99 SUBPROGRAM ESCAPE

### Dictate format: M99

End the transfer of the subprogram, return to main program.

**Note: 1. The F of transfer and transfer dictate must have the corresponding segment number, and it can't construct incurable cycle.**

**2. The subprogram and subprogram transfer must be applied in pairs.**

## 5.4.10 USER OUTPUT

### Dictate format: M21 R

M22 R

M23 R

M24 R

Standing time of the output signal can be defined with R.. After R defined the standing time, the output signal is opposite to original signal, the dictate switches between valid and invalid. If the parameter don't be defined, the signal is long one, cancel it only by corresponding M dictate.

**M21 R ; 1# valid user output, the output abates after R delay**

**M22 R ; 1# invalid user output, the output inures after R delay**

**M23 R ; 2# valid user output, the output abates after R delay**

**M24 R ; 2# invalid user output, the output inures after R delay**

## 5.4.11 USER INPUT

### Dictate format: M81

M82

M83

M84

M81 1# pause when valid user input, run sequentially when invalid input

M82 1# pause when invalid user input, run sequentially when valid input

M83 2# pause when valid user input, run sequentially y when invalid input

M84 2# pause when invalid user input, run sequentially when valid input

## 5.5 PROGRAMING PRINCIPLES


1. Each segment only has one M code or G code, invalid 0 of G、M codes can be defaulted.
2. Each segment number must have 3 bits integer
3. The excrescent parameter can be ignored
4. The positions of the dictate and field in a segment can be arranged at will with sort order
5. It is no need to rewrite the mode dictate
6. After entering the last segment, press 'Esc' to end the input

## 6 ALARM LIST

Please release from alarms referring to this list when the controller alarms

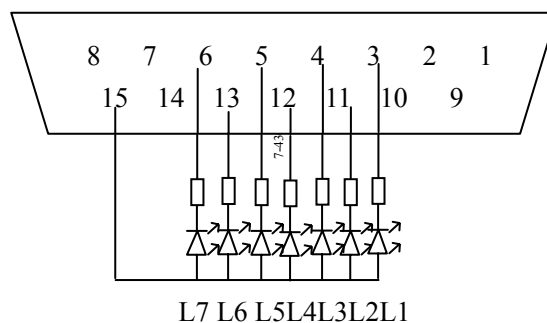
| Err No. | Content                               | Release method   |
|---------|---------------------------------------|--|
| 0       | New file number                       | Press any button to enter new file's content                 |
| 1       | Segment without number                | Input the segment number                                     |
| 2       | incorrect segment number              | Change the segment number to 3 bits integer                  |
| 3       | Repeated dictate or data              | Delete the repeated dictate or data                          |
| 4       | Exceed scale of input data            | Amend input data   |
| 5       | incorrect dictate format              | Input dictate with standard format                           |
| 6       | Irregular dictate                     | Input dictate with standard format                           |
| 7       | program without ending                | End program with M02 or M03                                  |
| 8       | G/M or Z/W concurrence                | Delete excrescent G/M or Z/W                                 |
| 9       | Without dictate G50                   | Input dictate G50 before the first G dictate                 |
| 10      | Necessary parameter lost              | Input the necessary parameter                                |
| 11      | travel limit                          | Amend dictate data or P8, P9 parameter                       |
| 12      | Improper G22/G80, M98/M99             | Input lost dictate   |
| 13      | Transfer without corresponding signal | Amend segment number   |
| 14      | Without file or incorrect source file | Input new file or select existing file                       |
| 15      | Irregular character in file storage   | Clear up the file storage and input again                    |
| 16      | Without parameter input               | Press any button   |
| 17      | Input same file name                  | Input the other file name                                    |
| 18      | Switch limit                          | Reversely move by hand                                       |
| 19      | 0 multiple                            | Reset multiple that isn't 0                                  |
| 20      | Driver alarm                          | Before power on again, check the driver and remove its fault |

## 7 DIAGNOSIS

Press 'Enter' button after reset, input 113 again to diagnosis menu. In diagnosis function, it is needed pressing  instead of 'A' button.

### 7.1 MOTOR INTERFACE DIGNOSIS

Connect circuitry as right drawing



L7 L6 L5L4L3L2L1

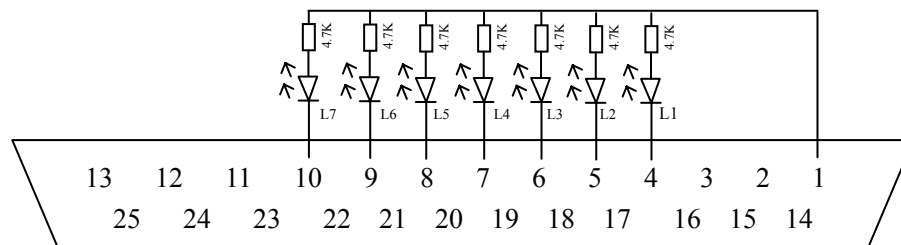
Motor interface diagnosis 27

**Diagnosis menu is as follows:**

Orderly press W A 3 0 1 <ENTER> , L1 lighting ( the others unlighted)  
 Orderly press W A 3 0 2 <ENTER>, L2 lighting ( the others unlighted)  
 Orderly press W A 3 0 4 <ENTER>, L3 lighting ( the others unlighted)  
 Orderly press W A 3 0 8 <ENTER>, L4 lighting ( the others unlighted)  
 Orderly press W A 3 1 0 <ENTER>, L5 lighting ( the others unlighted)  
 Orderly press W A 3 2 0 <ENTER>, L6 lighting ( the others unlighted)  
 Orderly press W A 3 4 0 <ENTER>, L7 lighting ( the others unlighted)

**7.2 DIAGNOSIS OF SIGNAL INPUT INTERFACE**

- 1、2、14、16~23 pins are +24VDC;  
 4、5、6、7、8、9、10 pins are for signal output  
 Connect the circuitry as follows:



Signal output diagnosis

In diagnosis menu,

Orderly press W A 4 0 0 <ENTER>

W A 5 0 0 <ENTER>

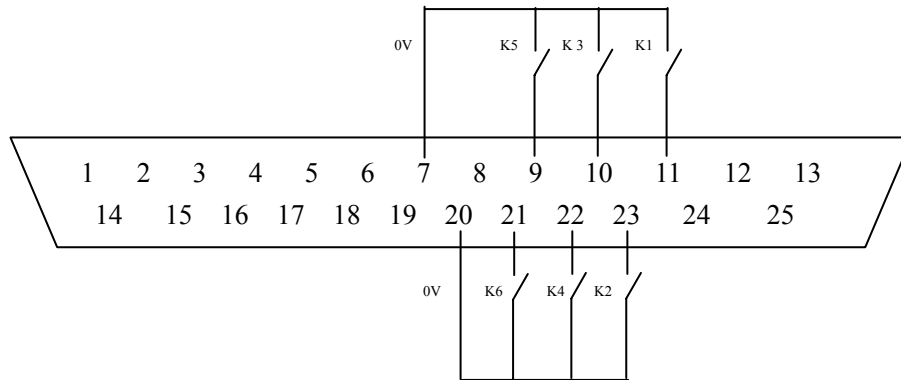
Close all LED indicators

Orderly press W A 4 0 1 <ENTER>, L1 lighting (the others unlighted)  
 Orderly press W A 4 0 2 <ENTER>, L2 lighting (the others unlighted)  
 Orderly press W A 4 0 4 <ENTER>, L3 lighting (the others unlighted).  
 Orderly press W A 4 0 8 <ENTER>, L4 lighting (the others unlighted)  
 Orderly press W A 4 1 0 <ENTER>, L5 lighting (the others unlighted)  
 Orderly press W A 4 2 0 <ENTER>, L6 lighting (the others unlighted)  
 Orderly press W A 4 4 0 <ENTER>, L7 lighting (the others unlighted)

**7.3 DIAGNOSIS OF SIGNAL INPUT INTERFACE**

- 1、6、18、19 pins are +24VDC  
 7、8、13、20、25 pins are 0V  
 9、10、11、21、22、23 pins are for signal input

Connect circuitry as follows:



Signal input diagnosis

In diagnosis menu,

Switch on K1, orderly press R A 2 〈ENTER〉 , then the controller displays FB

Switch on K2, orderly press R A 2 〈ENTER〉 , then the controller displays F7

Switch on K3, orderly press R A 2 〈ENTER〉 , then the controller displays EF

Switch on K4, orderly press R A 2 〈ENTER〉 , then the controller displays BF

Switch on K5, orderly press R A 2 〈ENTER〉 , then the controller displays DF;

Switch on K6, orderly press R A 2 〈ENTER〉 , then the controller displays 7F;

# CNCmakers

[www.CNCmakers.com](http://www.CNCmakers.com)

[info@CNCmakers.com](mailto:info@CNCmakers.com)