

**Declaration!**

- We try to describe all the various matters as much as possible in this manual. However, it is impossible to give detailed descriptions to all the unnecessary or unallowable operations because there are too many possibilities. Therefore, the matters not specially described herein should be considered as “impossible” or “unallowable”.

**Warning !**

- Before installing, connecting, programming and operating the product, please read this manual and the manual provided by the machine tool builder carefully, and operate the product according to these manuals. Otherwise, the operation may cause damage to the product and machine tool, or even cause personal injury.

**Caution !**

- The functions and specifications (e.g., precision and speed) described in this manual are only for this product itself. For those CNC machine tools with this product installed, the actual function configuration and specifications depend on the designs of the machine tool builders. Moreover, the function configuration and specifications of the CNC machine tool are subject to the manual provided by the machine tool builder.
- Please refer to the user manual issued by the machine tool builder for the function and meaning of each key on the panel.

**All specifications and designs in this manual are subject to change without notice.**

## II Precautions

### 1) Inspection and acceptance



- Caution • It is not allowed to use damaged or defective products.

### 2) Transport and storage



- Note • Guard the products against moisture during transit and storage; do not climb up or stand on the packages of the products, or place heavy objects on the packages; do not pile up the packages more than 5 layers; avoid impact and scratch to the front panel and LCD screen.

### 3) Installation



- Caution • Protect the system from sunlight and raindrops because the shell of the system is not waterproof.



- Note
- Prevent dust, corrosive air, liquid, conductors and inflammable substances from entering the system.
  - Keep the system away from inflammable and explosive substances. Avoid places where there is powerful electromagnetic interference.
  - Install the system firmly in case of vibration.

### 4) Connection



- Warning • Only qualified persons can connect the system or check the connection. No damage should be caused to the connecting wires. Do not press or open the cover of the system with power on.

**Caution**

- The voltage and the polarity of connecting plugs must accord with the manual.
- Wet hands are dangerous to grasp the plug or the switch.

**Note**

- The connection must be proper and firm.
- The system must be earthed.

## 5) Debugging

**Warning**

- Make sure that the parameters of the system are correct before running.
- No parameter should be beyond the setting limit in the manual.

## 6) Operation

**Warning**

- Only qualified operators can operate the system.
- Ensure the switch is OFF before connecting the power supply.

**Warning**

- The operator can not leave the system to work alone.
- Make sure the connection is correct before Power On.
- The emergency stop button should be able to cut off all power supplies when the system breaks down. Do not switch on/off the system frequently.

**Warning**

- Prevent the system from environmental interference.

## 7) Troubleshooting

**Caution**

- Unqualified persons cannot repair the system.

**Warning**

- After an alarm occurs, do not restart the system until the breakdown is fixed.

### **III Safety and precautions for programming**

#### **1) Coordinate system**

Incorrect coordinate system may cause the machine not to work as expected even if the instruction is correct, which may injure the operator, and damage the machine as well as its tool and workpiece.

#### **2) G00 rapid traverse**

G00 rapid traverse performs nonlinear motion between its starting point and end point. Make sure that the path for the tool is safe before G00 rapid traverse starts, otherwise the tool, the machine and the workpiece may be damaged, and even the operator injured.

#### **3) Use of this manual**

This manual introduces in details all functions of the system, including optional functions and max. controllable ranges, which are subject to change with the machine. Therefore, some functions described in this manual may not be applicable to a specific machine tool. If there is any doubt, please read the instruction for the machine.

#### **4) Functions of the CNC and machine tool**

The functions of CNC machines not only depend on CNC systems, but also power voltage cabinets, servo systems, CNC and the operator panels. It is hard to explain all the integrated functions, programming and operation. Do not use integrated instructions not included in the manual until they have been tested successfully.

## IV Precautions and warnings for operation

### 1) Before machining a part

First check whether the machine tool works normally. Make sure that the machine tool works normally by means of trial run before machining, with no workpiece and tool mounted on the machine tool.

### 2) Before operating the machine tool

Check the input data of the system carefully before operating the machine. Incorrect input data may cause the machine to work improperly, and thus damage the workpiece and the tool, as well injure the operator.

### 3) Make sure the system input feedrate is suitable for the expected operation.

In general, there is a maximum feedrate for each machine tool. The proper feedrate varies with different operations. Please refer to the user manual to determine the maximum feedrate. If the user does not operate the machine tool at a proper speed, the machine tool may work incorrectly, thus causing damage to the workpiece or the machine tool itself, or even cause personal injury.

### 4) Compensation function

When tool compensation is needed, check the direction and the amount of the compensation. Improper compensation causes the machine to work wrongly, so as to damage the workpiece and the tool, as well injure the operator.

### 5) Manual operation

If the machine is to run in Manual Mode, check the current position of the tool and the workpiece, and correctly specify the moving axis, moving direction and the feedrate. During MPG feed, rotating the MPG (previously called electronic handwheel) with a large override, such as 100%, causes the tool and worktable to move rapidly. In such a case, the tool and worktable will not stop immediately even when the MPG is not rotated. Therefore, MPG movement with a large override may cause damage to the tool or machine, or even injury to the operator.

### 6) Manual reference point return

If manual reference point return is required, make sure that the machine has been equipped with the detecting element for the reference point. If the manual reference point return is performed without installing the detecting element, the tool keeps moving until it hits the stroke limit, which may cause damage to the machine, workpiece and tool, or even injury to the operator.

## Safety responsibility

### **Manufacturer Responsibility**

- Be responsible for the danger which should be eliminated on the design and configuration of the provided CNC systems
- Be responsible for the safety of the provided CNC and its accessories
- Be responsible for the provided information and advice

### **User Responsibility**

- Be trained with the safety operation of CNC system operation procedures and familiar with the safety operation.
- Be responsible for the dangers caused by adding, changing or modifying the original CNC systems and accessories.
- Be responsible for the danger caused by failing to observe the operation, maintenance, installation and storage in the manual.

This user manual shall be kept by the end user.

**Thank you for your support when you are using the products**

## **I OPERATION**

**introduces the operation methods, technical specifications and parameter setting for GSK928TD turning machine CNC system.**

## **II PROGRAMMING**

**introduces the instruction codes and program formats of the CNC system.**

## **III Connection**

**introduces the installation and connection of the CNC system.**

## **IV Appendix**

**introduces the supplementary explanations for the installation and connection of the CNC system.**



# CONTENTS

<b>OPERATION</b> .....	<b>3</b>
<b>CHAPTER ONE OVERVIEW</b> .....	<b>3</b>
<b>CHAPTER TWO TECHNICAL SPECIFICATION</b> .....	<b>5</b>
2.1 928TD Technical Specifications .....	5
<b>CHAPTER THREE OPERATION PANEL</b> .....	<b>7</b>
3.1 LCD .....	7
3.2 LED Status Indicator .....	7
3.3 Keyboard .....	7
3.3.1 Character Key .....	7
3.3.2 Operation Mode Select Key .....	8
3.3.3 Function Key .....	8
3.3.4 Cycle Start Key and Cycle Pause Key (Feed Hold key) .....	9
3.3.5 Manual Axis Control Key .....	9
3.3.6 Manual Auxiliary Function Key .....	10
3.3.7 Edit Key .....	11
3.3.8 Reset Key .....	12
<b>CHAPTER FOUR SYSTEM OPERATION</b> .....	<b>13</b>
4.1 System Power-on, Power-off, Initial State, Modal State, and Safety Protection .....	13
4.1.1 Power On .....	13
4.1.2 Power Off .....	14
4.1.3 Initial State and Modal State of System and Program .....	14
4.1.3.1 Initial State and Modal State of System .....	14
4.1.3.2 Initial State and Modal State of Program .....	15
4.1.4 Safety Protection .....	15
4.1.4.1 Hard Limit Protection .....	15
4.1.4.2 Soft Limit Protection .....	16
4.1.4.3 Emergency Stop Alarm (Stopping System Emergently) .....	17
4.1.4.4 Drive Unit Alarm .....	19
4.1.4.5 Other Alarms .....	19
4.1.4.6 Power Off .....	20
4.1.4.7 Reset Operation .....	20
4.2 Operation Mode Selection for CNC System .....	21
4.3 Edit Operation Mode .....	21
4.3.1 Part Program Directory Search .....	22
4.3.2 Selecting, Creating, Deleting, Renaming and Copying a Part Program .....	23
4.3.2.1 Selecting and Creating a Part Program .....	23
4.3.2.2 Deleting a Part Program .....	24
4.3.2.3 Deleting All Part Programs .....	24
4.3.2.4 Renaming a Part Program .....	24
4.3.2.5 Copying a Part Program .....	25
4.3.3 Part Program Communication .....	25
4.3.3.1 Sending Part Program (CNC→PC, CNC→USB, CNC→CNC) .....	25
4.3.3.2 Receiving Part Programs (PC→CNC, USB→CNC, CNC→CNC) .....	26
4.3.3.3 Standard Format of TXT Part Program on PC .....	27
4.3.4 Inputting and Editing the Contents of Part Program .....	28
4.3.4.1 Inputting Program Contents .....	31
4.3.4.2 Inserting a Block .....	32
4.3.4.3 Deleting a Block .....	32
4.3.4.4 Inserting a Character in a Block .....	33
4.3.4.5 Deleting a Character in a Block .....	33
4.3.4.6 Altering Contents of a Block .....	33
4.3.4.7 Inserting a Macro String .....	34
4.3.4.8 Storage Capacity for Programs .....	34
4.3.4.9 Operating No. 253 Program .....	34
4.3.4.10 Operating No. 254 program .....	35

4.3.5	Function of <b>hp5</b> Key	35
4.3.5.1	Help for Part Program Command	35
4.3.5.2	Help for Obtaining Relative Parameters of Arc	36
4.3.5.3	Rearrangement of Program Line Numbers	37
4.3.5.4	Replacement of Strings	37
4.3.5.5	Cursor Position	37
4.3.5.6	Cursor Movement by MPG	37
4.3.6	Compiling a Part Program	38
4.3.6.1	<b>hp3</b> Compiling Command	38
4.3.6.2	Result Analysis of Program Compilation	38
4.3.6.3	Prompts of Program Compound Check	39
4.4	JOG Operation Mode	40
4.4.1	Coordinate Axis Movement	42
4.4.1.1	JOG movement	42
4.4.1.2	Step Movement	42
4.4.1.3	MPG Control Movement	43
4.4.1.4	Selecting Rapid Traverse Rate	44
4.4.1.5	Selecting Speed for Low-speed Feed	45
4.4.1.6	Inputting a Word to Move, Setting Feedrate	45
4.4.1.7	Drive Unit Enabling Control	47
4.4.1.8	Alarm Prompts for Coordinate Axis Movement	47
4.4.2	Establishing a Coordinate System	48
4.4.2.1	Establishing Machine Coordinate System—Machine Zero Return (Machine Reference Point Return)	48
4.4.2.2	Establishing Machine Coordinate System— without Machine Zero (No Machine Reference Point)	50
4.4.2.3	Setting Workpiece Coordinate System	50
4.4.2.4	Setting Program Reference Point	52
4.4.2.5	Program Reference Point Return	52
4.4.2.6	Recovering Workpiece Coordinate System and Program Reference Point	53
4.4.3	Spindle Control Function	53
4.4.3.1	Spindle Start/Stop Control	53
4.4.3.2	Spindle S Command – Gear Shift Control	55
4.4.3.3	Spindle S Command— Rotating Speed Control	56
4.4.4	Coolant Control	59
4.4.5	Manual Tool Change Control	59
4.4.6	Manual Tool Change	61
4.4.7	Hydraulic Chuck Control Function	65
4.4.8	Hydraulic Tailstock Control Function	67
4.4.9	Other Option Functions	69
4.4.9.1	Triple-color Indicator Control	69
4.4.9.2	Lubricant Control	70
4.4.9.3	Machine Electricity Delay Power-on Control	70
4.4.9.4	Safety Door Detection Function	70
4.4.9.5	Low-pressure Detection Function	71
4.4.10	Viewing Operation Information in Manual Mode	71
4.4.11	Appendix Table	71
4.4.11.1	List of M function Commands Controlled by MDI Input	71
4.5	Auto Operation Mode	73
4.5.1	System Working States in Auto Operation Mode	74
4.5.2	Explanations for Function Key Operation in Auto Operation Mode	74
4.5.2.1	Switching between Single and Continuous Operation	74
4.5.2.2	Switching between Dry Run and Machining Run	75
4.5.2.3	Switching between Coordinate Display and Graph Display	76
4.5.2.4	Starting Execution from First Block of Program	76
4.5.2.5	Starting Execution from a Specified Block	76
4.5.3	Display during Program Execution	77
4.5.3.1	Definition of Graph Display Data	77
4.5.3.2	Inputting Graph Display Data	78
4.5.3.3	Part Count and Timing	79
4.5.4	Manual Operation for Machine Auxiliary Functions	80
4.5.5	Speed Override Adjustment in Auto Operation Mode	80
4.5.5.1	Speed Override Adjustment	80

## Contents

4.5.5.2	MPG Speed Control	81
4.5.6	Interruption Operation during Program Execution	82
4.5.6.1	Interruption with Keys during Program Execution	82
4.5.6.2	External feed/Spindle hold Knob	83
4.5.6.3	External Start and Pause Signals	84
4.5.6.4	Feeding Device Alarm Function	84
4.5.7	Modifying Tool Offset during Program Execution	84
4.5.7.1	Methods of Modifying Tool Offsets during Program Execution	84
4.5.7.2	Modifying Validity of Tool Offsets during Program Execution	85
4.5.8	Viewing Running Information in AUTO Operation Mode	85
4.5.9	Return to Program Reference Point in AUTO Operation Mode	87
4.5.10	System Reset Key and Emergency Stop Signal Processing in Auto Mode	87
4.5.11	Adjusting Brightness of LCD screen in AUTO, MANUAL Operation Mode	87
4.5.12	Displaying Executing States of M Commands in Auto, Manual Operation Mode	88
4.5.13	Additional Operation in Auto Operation Mode	88
4.6	Parameter Operation Mode	89
4.6.1	Parameter Overview	90
4.6.1.1	Parameter Authority	90
4.6.1.2	Entering an Operation Level	90
4.6.1.3	Parameter Management	90
4.6.2	Modifying Parameters	92
4.6.2.1	Searching Parameters	92
4.6.2.2	Modifying Parameters	92
4.6.3	Parameter hp6 Function	93
4.6.3.1	Parameter Communication and Standard Format	93
4.6.3.2	Parameter Extraction and Solidification	96
4.6.3.3	System Software Upgrade and Internal Memory Update	97
4.6.3.4	Function Command Authority	98
4.6.4	Description of Parameters	98
4.6.4.1	Parameters of Reference Point, Soft Limit __ P000~P020	98
4.6.4.2	Parameters of Zero Return Function __ P021~P026, P109~P111, P406~P407	99
4.6.4.3	Parameters of Movement Speed and Acceleration Time __ P100~P108, P112~P119	101
4.6.4.4	Parameters of Drive and Compensation P200~P209, P411, P1000~P1905	103
4.6.4.5	Parameters of Spindle and Coolant __ P300~P317, P326, P329, P341, P410	106
4.6.4.6	Parameters of Tool Post __ P318~P325, P408	109
4.6.4.7	Parameters of Chuck and Tailstock __ P327~P328, P409	112
4.6.4.8	Bit Parameters of Running and Efficiency __ P400~P401	113
4.6.4.9	Relationship between Path and Parameters of Running and Efficiency	115
4.6.4.10	Bit Parameters of Safety and Debugging __ P402~P404, P419	116
4.6.4.11	Bit Parameter of Motor Driver __ P405	121
4.6.4.12	Parameters of Other Interfaces __ P412, P330~P332	121
4.6.4.13	Other Parameters __ P413~P416, P333	123
4.6.4.14	Parameters of Interface __ P500~P556	126
4.6.4.15	Initial Values of Variables __ P600~P639	127
4.6.4.16	Parameters of G76 __ P336~P339	127
4.6.5	Appendix Parameter List	128
4.6.5.1	Reference Parameter List	128
4.6.5.2	Motion Parameter List	129
4.6.5.3	Drive Parameter List	130
4.6.5.4	Auxiliary Parameter List	130
4.6.5.5	Bit Parameter List	131
4.6.5.6	Interface Parameter List	132
4.6.5.7	Variable Initial Value List	133
4.6.5.8	Pitch Error Compensation Parameter List	134
4.6.5.9	List of Parameters Relative to Command Disabling	134
4.6.5.10	List of Parameters Relative to Output Interface Releasing	135
4.6.5.11	List of Parameters Related to Input Interface Releasing	135
4.7	Tool Offset Operation Mode	137
4.7.1	Searching Tool Offset Value	138
4.7.2	Inputting Tool Offset Data from Keyboard	138

- 4.7.3 Clearing Offset Values of Each Group ..... 139
- 4.7.4 Tool Compensation hp6 Function ..... 139
  - 4.7.4.1 Communication and Standard Format of Tool Offset Data ..... 140
  - 4.7.4.2 Tool Compensation Data Clearing ..... 141
- 4.8 Diagnosis Operation Mode ..... 142
  - 4.8.1 Searching Interface Signal ..... 142
  - 4.8.2 Explanations for Display of Interface Signals Names ..... 143
  - 4.8.3 Explanation of Input Interface Diagnosis ..... 143
  - 4.8.4 Explanation of Output Interface Diagnosis ..... 143
  - 4.8.5 Output Interface Operation Function ..... 144
  - 4.8.6 Spindle Encoder and Spindle Speed Detection ..... 144
  - 4.8.7 Diagnosis hp6 Function ..... 144
    - 4.8.7.1 Display of Alarm Record ..... 145
    - 4.8.7.2 Searching Alarm Record ..... 146
    - 4.8.7.3 Alarm Record hp6 Function ..... 147
  - 4.8.8 Machine Auxiliary Function Control ..... 148

**CHAPTER FIVE RS232 AND USB SYSTEM COMMUNICATION ..... 149**

- 5.1 RS232 Communication ..... 149
  - 5.1.1 Communication between CNC and PC ..... 149
  - 5.1.2 Communication between CNC and CNC ..... 150
- 5.2 USB Communication ..... 150
  - 5.2.1 USB Operation ..... 151
  - 5.2.2 USB File Directory Requirements ..... 151

**PROGRAMMING ..... 155**

**CHAPTER ONE PROGRAMMING FUNDAMENTAL ..... 155**

- 1.1 Coordinate Axis and Its Direction ..... 155
- 1.2 Machine Coordinate System, Machine Zero ..... 156
- 1.3 Program Reference Point ..... 156
- 1.4 Machine 2nd, 3rd Program Reference Point ..... 156
- 1.5 Workpiece Coordinate System ..... 156
- 1.6 Programming Coordinate ..... 157
  - 1.6.1 Absolute Coordinate Values ..... 157
  - 1.6.2 Relative Coordinate Values ..... 157
  - 1.6.3 Compound Coordinate Values ..... 158
- 1.7 Diameter Programming and Radius Programming ..... 158
- 1.8 Interpolation Function ..... 158

**CHAPTER 2 PROGRAM CONFIGURATION ..... 160**

- 2.1 Character ..... 160
- 2.2 Word ..... 160
- 2.3 Block Number ..... 161
- 2.4 Block ..... 161
- 2.5 Block Skip Symbol and Comment ..... 162
- 2.6 Program Structure ..... 163

**CHAPTER 3 MSTF COMMANDS AND FUNCTIONS ..... 164**

- 3.1 M — Auxiliary Function (Command List) ..... 164
  - 3.1.1 M00 — Pause ..... 165
  - 3.1.2 M02 — End of Program ..... 165
  - 3.1.3 M20 — Program End Cycle Machining ..... 165
  - 3.1.4 M30 — End of Program, Spindle OFF, Cooling OFF ..... 166
  - 3.1.5 M03, M04, M05 — Spindle Control ..... 166
  - 3.1.6 M08, M09 — Coolant Control ..... 166
  - 3.1.7 M10, M11, M12 — Clamping/Releasing Chuck, Cancelling Chuck Output Signal ..... 167
  - 3.1.8 M32, M33 — Lubricating ON/OFF ..... 167
  - 3.1.9 M41, M42, M44, M43 — Spindle Automatic Gear Shift Control ..... 167
  - 3.1.10 M78, M79, M80 — Tailstock Advancing and Retracting, Tailstock Output Signal Cancelling ..... 168
  - 3.1.11 M96 — Calling Cycle Execution ..... 168
  - 3.1.12 M97 — Program Transfer ..... 169
  - 3.1.13 M98, M99 — Subprogram Call and Subprogram Return ..... 169

## Contents

3.1.14	M21, M22, M23, M24 —User Output Control	171
3.1.15	M91, M92, M93, M94 — User Input Control	171
3.1.16	M60~M74 — Custom Command	172
3.2	M81, M82, M83—User Input/Output Condition Control	172
3.2.1	M82— Output Control and Detection	173
3.2.2	M81—Control According to Input Signal State	173
3.2.3	M83—Control According to Output Signal State	174
3.3	S function — Spindle Function	174
3.3.1	Spindle Motor Controlled by Gear Shift	174
3.3.2	Variable-frequency Motor Controlled by Speed	175
3.4	T Function — Tool Function	176
3.4.1	Tool Offset Execution Mode-Moving Slide Carriage	176
3.4.2	Tool Offset Execution Mode- Modifying Coordinates	177
3.5	F Function — Feedrate Function	177
<b>CHAPTER FOUR G COMMANDS AND FUNCTIONS</b>		<b>180</b>
4.1	G00 — Rapid Positioning G00	180
4.2	G01 — Linear Interpolation	182
4.3	G02, G03, G05 —Circular Interpolation	183
4.4	Thread Cutting Command	188
4.4.1	G33 — Thread Cutting	188
4.5	G32 —Tapping Cycle	196
4.6	G50 — Setting Workpiece Coordinate System	197
4.7	G51 — Recovering Workpiece Coordinate System Setting	198
4.8	G26 — X, Z Reference Point Return	199
4.9	G28 — Return to Machine Zero (Machine Reference Point)	200
4.10	G30 — 2 <sup>nd</sup> , 3 <sup>rd</sup> Program Reference Point Return	201
4.11	G04 — Timing Delay	201
4.12	G96 —Constant Surface Speed Control, G97 —Constant Surface Speed Cancel	202
4.13	Single Canned Cycle	205
4.13.1	G90 —Outer Cylinder Face Turning Cycle (Axial Cutting Cycle)	205
4.13.2	G92 —Thread Cutting Cycle	208
4.13.3	G94 — Inner/outer End (Taper) Face Turning Cycle	216
4.13.4	G74 —End Face Deep Hole Machining Cycle	219
4.13.5	G75 —Grooving Cycle	221
4.14	Compound Cycle	223
4.14.1	G71 —Axial Plane Rough and Finish Command Group	223
4.14.2	G72 —End Face Roughing/Finishing Command Group	228
4.14.3	G73 — Closed-loop Cutting Cycle Command Group	232
4.14.4	G76 —Multiple Repetitive Threading Cycle	236
4.15	G22, G80 —Program Local Cycle	241
4.16	G98 — Feed per Minute, G99 — Feed per Revolution	243
4.17	G31 — Skip Function	244
4.18	G66 -Memorizing Current Coordinates, G67-Return to Memorized Coordinates	246
4.19	Appendix: G function and Its Explanation Table (Table 4-3)	246
4.20	Appendix: G and Its Relative Parameter Explanation (Table 4-4)	248
<b>CHAPTER FIVE GENERAL PROGRAMMING RULES AND EXAMPLES</b>		<b>249</b>
5.1	General Programming Rules	249
5.2	Programming Rules for Commands in One Block	250
5.3	Command Execution Sequence	251
5.4	Programming Example	253
5.4.1	Outer Machining Example	253
5.4.2	Thread Machining Example	254
5.4.3	Compound Machining Example	257
<b>CHAPTER SIX ALARM MESSAGE</b>		<b>261</b>
6.1	Emergency Stop Alarm	261
6.2	Alarm Table in PARAMETER, OFFSET Operation Mode (i.e. E001~E009)	261
6.3	Table of Alarm in Edit Operation Mode(i.e. E100~ E199)	263
6.4	Table of Alarms Relative to Program (i.e.E200~ E299, E600~ E699)	265
6.4.1	Alarm in Program Command (i.e. E200~299)	265
6.4.2	Alarm in Program Comprehensive Check Alarm (E600~E699)	268
6.5	Table of Alarm in JOG or AUTO Operation Mode (i.e. E300~ E499)	270

6.5.1	Alarm in Executing Relative Operations (i.e. E300~E399)	270
6.5.2	Relative Alarm in Executing Statement (E400~ E499)	274
<b>CHAPTER SEVEN STATEMENT PROGRAMMING (not for GSK928TD)</b>		<b>276</b>
7.1	Variable	276
7.1.1	Variable Expression Method	276
7.1.2	Classification of Variable	276
7.1.2.1	Command Variable	277
7.1.2.2	Pointer Variable	279
7.1.2.3	Interface Variable	280
7.1.2.4	Keyboard Scan Register r5001	281
7.1.2.5	Display Window Register r5002	282
7.1.2.6	Display Value Register r5003	285
7.1.2.7	Graph Refresh Register r5004	285
7.1.2.8	Program Control Register r5008	286
7.1.2.9	System Special Variable Group 1	287
7.1.2.10	System Special Variable Group 2	288
7.2	Statement	289
7.2.1	Assignment Statement	289
7.2.2	Conditional Statement	290
7.2.3	Statement Program Example	290
7.3	Process Monitoring and Execution	292
7.3.1	Process Monitor Description (r7000)	292
7.3.2	Process Monitor ON/OFF	293
7.3.3	Monitor Program Example	295
7.3.4	Pulse Monitoring (r7100)	297
7.3.5	Pulse Monitoring Programming Example	298
7.3.6	Variable Transfer Register (r7900)	299
7.4	Attached List	300
7.4.1	ASCII List	300
7.4.2	Corresponding List between Common Colors and Code Values	300
<b>CHAPTER EIGHT CUSTOMIZED COMMAND PROGRAMMING</b>		<b>301</b>
8.1	Customized Command	301
8.1.1	Programming Format of Customized Command	301
8.2	Customized Command Library (P254)	301
8.2.1	Programming Format and Debugging of Customized Command Library	302
8.2.2	Use of Customized Command Library	302
8.3.	Foot Switch of M61 command	303
<b>CONNECTION</b>		<b>307</b>
<b>CHAPTER ONE INTERFACE OVERVIEW</b>		<b>307</b>
1.1	Rear Cover Interface Layout	307
1.2	Overall Frame	308
<b>CHAPTER TWO INTERFACE TABLE</b>		<b>309</b>
<b>CHAPTER THREE CNC DEVICE CONNECTION</b>		<b>311</b>
3.1	Communication Interface	311
3.1.1	USB Interface	311
3.2	X1, X2 Interface	311
3.2.1	X1, X2 Interface Signal Definition	311
3.2.2	Connection Method of Input Signal	314
3.2.3	Connection Method of Output Signal	316
3.2.4	Input/Output Signal Specification	318
3.3	Machine Zero Return Function and Connection	318
3.4	Tool Change Control Function and Connection	321
3.4.1	Definition of Tool Change Control Signal	321
3.4.2	Signal Connection	321
3.4.3	Function Description	321
3.4.3.1	Tool Change Mode 0	322
3.4.3.2	Tool Change Mode 1	322
3.4.3.3	Tool Change Mode 2	323
3.4.3.4	Tool Change Mode 3	323

3.4.3.5	Tool Change Mode 4 .....	325
3.4.4	Tool Signal Check and Parameter Setting .....	326
3.4.4.1	Default mode (P408_d7=0) .....	326
3.4.4.2	Table Checking Mode .....	327
3.5	X3 Motor Interface .....	328
3.5.1	Signal Definition .....	328
3.5.2	Technical Specifications .....	329
3.5.3	Equivalent Circuit .....	329
3.5.3.1	Drive Unit Alarm Signal Xalm, Zalm .....	329
3.5.3.2	Enable Signal Xen, Zen .....	330
3.5.3.3	Pulse Signal and Direction Signal .....	330
3.5.4	Connection between CNC System and Drive Unit of Compound Stepper Motor .....	330
3.5.5	Connection between CNC and Drive Unit of Reaction Stepper Motor .....	333
3.5.6	Connection between CNC and AC Servo Drive Unit .....	335
3.5.7	Connection Diagram between CNC and Panasonic Drive Unit .....	337
3.5.8	Connection Diagram between CNC and Yaskawa Drive Unit .....	338
3.6	X4 Spindle Interface .....	339
3.6.1	Signal Definitions .....	339
3.6.2	Converter Technical Specification .....	339
3.6.3	Encoder Technical Specifications .....	340
3.6.4	Connection Diagram of Converter Analog Voltage Interface .....	340
3.6.5	Encoder Interface Principle .....	340
3.6.6	Encode Interface Connection Diagram .....	340
3.7	X5 MPG Interface .....	341
3.7.1	Signal Definition .....	341
3.7.2	MPG Interface Principle .....	341
3.7.3	MPG Interface Connection Diagram .....	341
<b>CHAPTER FOUR USE AND MAINTENANCE INFORMATION .....</b>		<b>342</b>
4.1	Ambient Condition .....	342
4.2	Earthing .....	342
4.3	Power Supply Requirements .....	342
4.4	Protection .....	342
4.5	Use after Long-time Unuse .....	342
<b>Appendix 1 CNC System Electrical Symbol Explanations .....</b>		<b>345</b>
<b>Appendix 2 CNC System Tool Post Controller Circuit Method Layout .....</b>		<b>346</b>
<b>Appendix 3 Interface Schematic Circuit .....</b>		<b>347</b>
<b>Appendix 4 External Control Connection Diagram .....</b>		<b>350</b>
<b>Appendix 5 GSK928TD CNC System Appearance Installation Dimension .....</b>		<b>351</b>
<b>Appendix 6 GSK928TD-L CNC System Appearance Installation Dimension .....</b>		<b>352</b>
<b>Supplementary Explanation .....</b>		<b>353</b>
<b>1. Modified Functions and Commands .....</b>		<b>353</b>
1.1	Newly-added Interface Parameter P538, P539, P540—Z/X/Y Move Limit .....	353
1.2	Newly-added Interface Parameter P351 — Alarm of Lubrication Check before Machining .....	353
1.3	G76 Command Modification .....	353
1.4	Diagnosis Operation Mode .....	354
1.5	AUTO Operation Mode .....	354
<b>2. Newly-added M Command: M50—M59, M84 .....</b>		<b>354</b>
2.1	Customized Commands Expanded to M50-M74 from M60-M74 .....	354
2.2	Newly-added Function of Calling M50-M72 before Machining .....	354
2.3	M84 — Input Signal Check within a Specified Time .....	355
<b>3. Newly-added G Commands and Functions .....</b>		<b>355</b>
3.1	G38 — Rigid Taping, Threading .....	355
3.2	G21, G20— Metric, Inch Input .....	357
3.3	Detailed Explanation of Metric/Inch Switch .....	358
3.3.1	Metric/Inch Switch Parameter .....	358

# I OPERATION





# OPERATION

## CHAPTER ONE OVERVIEW

The GSK928TD system employs a 32-bit high performance CPU and a complex programmable logic device of very-large-scale programmable array integrated circuits as its control center, thus realizing the movement control with a  $\mu\text{m}$ -level precision. The product, equipped with a true color LCD with resolution of  $480\times 234$ , uses the international standard NC language, also known as ISO codes, to write part programs. It is characterized by the full-screen program editing, Chinese/English operation interface, real-time track and display of the part graph and simple operation as well as the high cost performance. It can be matched with stepper motors or AC servo drive units, and by means of programming, it is capable of machining outer cylinders, end faces, grooves, tapers, circular arcs, threads, etc.

### Technical specifications

- ✓ Link axes: 2 (X, Z axes), short linear smooth interpolation at a high speed realizable; Interpolation precision: 0.001mm, max. rapid traverse: 15m/min
- ✓ Flexible and convenient programming
- ✓ USB interface communication, fast and easy to operate
- ✓ Least command increment: 0.001mm, electronic gear ratio: (1~99999) / (1~99999)
- ✓ Realizing controls like automatic tool post, spindle automatic gear shift.
- ✓ Backlash compensation, tool length compensation
- ✓ Exponential acceleration/deceleration control, applicable to high-speed and high-precision machining
- ✓ Tapping function
- ✓ Available to cut inch/metric thread, end face thread, continuous thread; with thread high-speed run-out
- ✓ Full-screen part program editing, capable of storing 255 programs; a capacity of 4 MB for No. 253 program
- ✓ True color LCD with a large screen, color profiles selected by parameters
- ✓ Real-time tracking and display of MSTF status during processing
- ✓ Multi-level passwords, convenient for equipment management
- ✓ Parameter backup function
- ✓ Communication of parameters and tool compensation data
- ✓ Support for two-way communication between CNC and CNC, and between CNC and PC; CNC software upgraded through a serial port
- ✓ Support for two-way communication between CNC and USB; CNC software upgraded through USB

- ✓ Installation dimension, electrical characteristics and part of interfaces compatible with GSK928TC turning machine NC system

Note

1. Neither the parameters nor the functions of Y axis described in this manual are valid.
2. The interface RS232 of the system has not been led out; to lead it out from the inside of the system, special tools and professional technician are required.

## CHAPTER TWO TECHNICAL SPECIFICATION

## 2.1 928TD Technical Specifications

<b>Motion control</b>	Simultaneously controlled axes (interpolation axes): 2 (X, Y axes)
	Interpolation function: linear, circular, and thread interpolation of X and Z axes
	Position command range: -9999.999 mm~9999.999mm; least command increment: 0.001mm
	Electronic gear: Command multiplier coefficient 1~99999, command division coefficient 1~99999
	Rapid traverse rate: Max. 15000mm/min; rapid override: F25%, 50%, 75%, 100% four-level real time adjustment
	Cutting feedrate: Max. 4000mm/min; feedrate override: 16-level real time adjustment from 0~150% (increment 10%)
	Manual feedrate: 0mm/min~1260mm/min 16-level real time adjustment, or user-defined speed in real time
	MPG feed: Three gears, 0.001mm, 0.01mm, 0.1mm
	Acceleration/deceleration: Either exponential or linear acceleration/deceleration can be selected for cutting feed.
<b>G codes</b>	G codes: G00, G01, G02, G03, G04, G05, (G22/G80), G26, G28, G30, G31, G32, G33, G50, G51, G66, G67, G71, G72, G73, G74, G75, G76, G90, G92, G94, G96, G97, G98, G99
<b>Thread machining</b>	Capable of machining single /multiple metric/inch straight thread, taper thread and end face thread; Thread run-out length, angle and speed characteristics settable, with high-speed run-out processing; thread pitch: 0.001mm~500mm or 0.06 teeth/inch ~ 25400 teeth/inch; tapping function available
	Spindle encoder: Setting range of encoder lines: 100 p/r~5000p/r; drive ratio between encoder and spindle: 1: 1
<b>Precision compensation</b>	Backlash compensation: 0 mm~10.000mm
	Tool compensation: 16 tool numbers, 64 groups of tool length compensations Tool setting mode: Trial tool setting, fixed point tool setting; Tool compensation execution mode: tool compensation executed by modifying coordinates, tool compensation executed by moving tool post
<b>M codes</b>	M00, M02, M20, M30, M03, M04, M05, M08, M09, M10, M11, M12, M32, M33, M41, M42, M43, M44, M78, M79, M80, M81, M82, M83, M96, M97, M98, M99, M91, M92, M93, M94, M21, M22, M23, M24; user defined M codes: M60~M74;
<b>T codes</b>	Up to 16 tool numbers (T01□□~T16□□), control process of tool change is selected by setting tool post type parameters; tool post type is set to 0 when using a line-up tool.
<b>Spindle speed control</b>	Speed switch value control mode: The output range of S command 4-gear direct control is S01~S04; or the output range of the 16-gear BCD code is S00~S15.
	Speed analog voltage control mode: S commands specify the spindle speed per minute or cutting linear speed (constant surface speed control); the CNC outputs 0V~10V voltage to the spindle frequency converter; spindle stepless speed variation; support for 4 spindle mechanical gears M41~M44
<b>I/O function</b>	I/O function diagnosis display
	I/O interface: 23 points input/18 points output
<b>Display interface</b>	Display: 480×234 lattice true color LCD, with LED or CCFL backlight
	Display mode: Chinese or English display interface set by parameters; real-time display of machining path
<b>Program edit</b>	Program number: up to 255 programs, program storage capacity: 4400KB
	Edit mode: Full-screen editing, support for incremental/absolute coordinate mixed programming, program calling, and subprogram multi-level nesting

<b>Communication</b>	With USB interface; bidirectional transmission of programs, parameters and tool compensations between CNC and USB; support for system software upgrade by USB download
<b>Adaptive driver</b>	GSK DA98 series digital AC servo or DY3 series step drive device, with pulse + direction signal input

## CHAPTER THREE OPERATION PANEL

This turning machine CNC system (abbreviated to system or CNC) employs an operation panel made from aluminum alloy.

### 3.1 LCD

LCD: Human-machine interface, with resolution of 480×234, lattice true color LCD

### 3.2 LED Status Indicator

LED indicators are used to indicate the current working states of the system. There are 15 function keys with a LED indicator. When the indicator lights up, the corresponding function of the key is enabled; when it goes out, the function is disabled.

### 3.3 Keyboard

According to the standard of GSK, the function keys with the visible signs below are designed for the system. The corresponding function of a function key is enabled when it is pressed. The meaning of each key is as follows:

#### 3.3.1 Character Key

Character keys consist of numbers, letters and some signs.

In the Edit mode, each letter key can switch between two or three keycodes; in other modes, each letter key only indicates one keycode. (E.g. Though I and P are on the same key, the system will automatically identify the keycode (I or P) to be used after pressing this key.)







**Numeric key:** Inputs data (0 ~9)

**Letter key:** Inputs letters

**Sign key:** Inputs signs such as + (plus sign), - (minus sign), \* (multiplication sign), / (division sign), + (positive sign), - (negative sign), . (decimal point), > (is greater than), = (is equal to), < (is less than), and, or, as well as ( ).

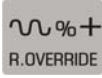
### 3.3.2 Operation Mode Select Key

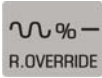
The keys are identified by a sign and letters. The user can finish the corresponding function by pressing an operation mode select key. The meaning of each key is as follows:


-  **selects Edit operation mode**
-  **selects Manual operation mode**
-  **selects AUTO operation mode**
-  **selects Parameter operation mode**
-  **selects Tool Offset operation mode**
-  **selects Diagnosis operation mode**


### 3.3.3 Function Key


The keys are indicated by a sign and letters. With a function key pressed, its function is enabled. The meaning of each key is as follows:

 **Rapid override increase** increases the rapid traverse override in MANUAL operation mode, and increases the speed override of G00 command in AUTO operation mode.

 **Rapid override decrease** decreases the rapid traverse override in MANUAL operation mode, and decreases the speed override of G00 command in AUTO operation mode.

 **Feedrate override increase** increases the feedrate override in MANUAL operation mode, and increases the speed override of G01 command in AUTO operation mode.

 **Feedrate override decrease** decreases the feedrate override in MANUL operation mode, and decreases the speed override of G01 command in AUTO operation mode.

 **X axis reference point return** is only valid in MANUAL/AUTO operation mode. (In this manual, the program reference point is also called the program zero point)



**Z axis program reference point return** is only valid in MANUAL/AUTO operation mode.



**X axis machine reference point return** only valid in MANUAL operation mode (In this manual, the machine reference point is also called the machine zero point).



**Z axis machine reference point return** only valid in MANUL operation mode.



**Dry run key** When Dry Run is selected in AUTO operation mode to execute commands, whether M, S, T commands are valid is determined by bit parameter P401\_d7. After the Dry Run state is exited, the coordinate of each axis of the system automatically resumes to the one before Dry Run.



**Single/Continuous key** selects Single/Continuous mode in AUTO operation mode; in other operations, it is for the **hp** function.

### 3.3.4 Cycle Start Key and Cycle Pause Key (Feed Hold key)

In AUTO operation mode, they are respectively used to start and suspend the program execution. The meaning of each key is as follows:



**Cycle Start key** starts the program in AUTO operation mode, and then the program is executed automatically; moves the coordinate axis in MANUAL operation mode.



**Cycle Pause key (Feed Hold key)** suspends the execution in MANUAL or AUTO operation mode; in other operation modes, it means the **hp** function.

#### 【Note】

There are two letters “hp” (help) on the upper right corners of some keys. In total, there are 7 Help keys, which are hp0~hp6; in different operation modes, when the main key is invalid, the hp is valid.

### 3.3.5 Manual Axis Control Key

In MANUAL operation mode, the meanings of manual axis control keys are as follows:





In MANUAL operation mode, X axis is moved in negative direction.



In MANUAL operation mode, X axis is moved in positive direction.



In MANUAL operation mode, Z axis is moved in negative direction.



In MANUAL operation mode, Z axis is moved in positive direction.



**RAPID/FEED key** In MANUAL operation mode, it switches between rapid traverse rate and feedrate.



In MANUAL operation mode, it is **Single Step/MPG Step Width Selection**; in other operation modes, it is the **hp** function.



In MANUAL operation mode, it is **MPG control selection and axis selection**; in other operation modes, it is the **hp** function.



In MANUAL operation mode, it is **Z/Y axis selection**, which is invalid for **928TD**; in other operation modes, it is the **hp** function.



**Step/JOG key** It selects Step/JOG operation in MANUAL operation mode.

### 3.3.6 Manual Auxiliary Function Key

The keys below are used for controlling and completing a variety of auxiliary functions of the machine tool. The meaning of each key is as follows:



**Spindle CW rotation** The spindle rotates in CW direction (viewed from the tailstock to the chuck)



**Spindle stop** The spindle stops rotating.



**Spindle CCW rotation** The spindle rotates in CCW direction (viewed from the tailstock to the chuck)



**Coolant control** switches between coolant ON/OFF.




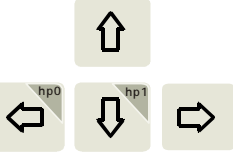

**Spindle gear shift** selects the speed of each spindle gear for the machine tools equipped with multi-speed spindle motors and control circuits.



**Tool change key** selects the next tool whose tool number is adjacent to the current one.

### 3.3.7 Edit Key

Key	Designation	Function
	ENTER key	Press this key for confirmation after input.
	INPUT key	Input the desired contents
	ALTER key	In Edit operation, press it to switch between word insertion/alternation; In other operations, it has specific meanings.
	DELETE key	In Edit operation mode, press it to delete words, blocks or a whole program. In other operations, it has specific meanings.
	ESCAPE key	Press it to cancel the current input data, or exit the operation state; Press it to quit the current operation or setting.
	HOME key	In AUTO operation mode, it indicates “Dry Run”; In EDIT operation mode, it moves the cursor to the beginning of the current line.
	END key	In MANUAL operation, it indicates “STEP”; In EDIT operation, it moves the cursor to the end of the current line.

Key	Designation	Function
	SINGLE BLOCK key	In AUTO operation mode, it means “Single/Continuous” program execution. In EDIT operation mode, it means “Single/Continuous” analog program execution. In other operations, it means the <b>hp</b> function.
	Cursor move keys	In Edit/Parameter/Tool Offset operation mode, they control the movement of the cursor; In other operation modes, they are <b>hp</b> functions or have other specific meanings.
	Page change keys	In Edit/Parameter/Tool Offset operation mode, they are used to turn pages up and down. In Manual/Auto operation, they have specific meanings.

### 3.3.8 Reset Key



**Reset** causes the system to enter RESET state.

## CHAPTER FOUR SYSTEM OPERATION

This chapter describes in detail the operation and precautions of each functional module of the CNC system. Before operating the machine, please thoroughly read this chapter.

### 4.1 System Power-on, Power-off, Initial State, Modal State, and Safety Protection

#### 4.1.1 Power On

There is no power switch on the operation panel of this CNC system. The user may install a power switch if necessary, so as to prevent the system from electric impact.

**Before turning on the system, please confirm that:**

- 1) The machine is in normal status;
- 2) The voltage of the power supply is in accordance with the requirements;
- 3) The connection is correct and firm.

**The operation procedures for System Power On are as follows:**

- 1) First turn on the power main switch;

Turn on the system power switch, then the system displays the initial screen as shown in fig. 4-1. In


this state, press any key other than  key to enable the system to enter EDIT operation mode.



Fig. 4-1 System initial screen

- 2) The following steps are completed in order after system power on.

- System control program loading
- System self-check and initialization
- System parameter loading and check
- I/O interface initialization
- User program loading and check

**【Note】**

- 1) Do not press any key on the operation panel during system power on. If the system enters the key test interface, please press RESET key to exit.

### 4.1.2 Power Off

**The operation procedures for system power off are as follows:**

- 1) Press the system power switch to cut off the power.
- 2) Turn off the main power switch of the machine.

**Before turning off the system, please confirm that:**

- 1) The X and Z axes of the system are at the halted state;
- 2) The auxiliary functions (e.g. spindle, cooling) are OFF;
- 3) The power is cut off.

**【Note】**

- 1) In general, if it is the first time that the system is powered on, self-check and initialization should be performed first.( This procedure is done by the machine tool builder rather than by the user. Otherwise, the parameters set by the machine tool builder may be lost. )
- 2) For the procedures of turning off the power of the machine, please see the user manual provided by the machine tool builder.

### 4.1.3 Initial State and Modal State of System and Program

#### 4.1.3.1 Initial State and Modal State of System

The initial state of the system is the specific state to which a function is automatically set upon system power on; therein, none of the auxiliary functions has actual output.

The modal state of the system is the state that a function remains in after the system executes this function.

Initial state and modal state of the system:

System state	Initial state of the system	Modal state
System machine coordinate system	Remains in the state of last Power-On.	Keeps until being changed.
System tool nose coordinate system	Remains in the state of last Power-On.	Keeps until being changed.
Cutting feedrate: F	In Auto operation: 30mm/min	Keeps until being changed.
	In Manual operation: Remains in the state of last Power-On	
Frequency-conversion spindle speed: S	S200	Keeps until being changed
Spindle gear	Gear-shift spindle gear: S0	Keeps until being changed
	Frequency-conversion spindle gear: M41	
Manual slow feed/rapid feed state	Slow feed	Keeps until being changed
Feedrate override	Remains in the state of last Power-On.	Keeps until being changed.
Rapid override	Remains in the state of last Power-On.	Keeps until being changed
Spindle state	M05 Spindle stop	Keeps until being changed
Coolant state	M06 coolant OFF	Keeps until being changed
Chuck state	M11 Chuck release	Keeps until being changed
Lubricant state	M33 Coolant OFF	Keeps until being changed
T tool number state	Remains in the state of last Power-On.	Keeps until being changed
Tailstock state	M79 Tailstock run-out state	Keeps until being changed

### 4.1.3.2 Initial State and Modal State of Program

The initial state of the program is the state automatically set by the system by means of initialization before a machining program is executed; i.e., the initial default state for the default programming word and speed word when the system executes the first command of the machining program.

Program initial states of the system are as follows:

G function: G00, G97, G98;

Cutting speed: 30mm/min;

Auxiliary function: Current state

System coordinates: Current coordinates, which are the ones after the last automatic program execution or after the last manual operation.

G function modal state means that once a word is set, it keeps unchanged until it is changed by other G modal commands belonging to the same group. The meaning of the modal state: after a G command is executed, it is unnecessary to input it again if it needs to be used in the following blocks.

The modal G commands are classified into the following three groups, and in each group, only one command is in the modal state.

Group 1: G00, G01, G02, G03, G05	(Initial state: G00)
Group 2: G96, G97;	(Initial state: G97)
Group 3: G98, G99;	(Initial state: G98 F30)

The non-modal commands are only effective in the current block. They must be specified each time they are used.

#### 【Note】

In AUTO operation mode, when the system executes the first command, or executes the first command after executing M20, or selects a block between the first lock and the last block as its first command, it will automatically resumes to the program initial state.

### 4.1.4 Safety Protection

A series of safety protection is set for the CNC system, so as to protect the operator as well as the machine tool. **(Y axis function is invalid)**

#### 4.1.4.1 Hard Limit Protection

For the machine tools mounted with limit switches, the system can detect the switches. When the slide carriage of the machine tool moves and presses down a limit switch, the system will stop feeding without turning off the other auxiliary functions, and then the program execution is stopped, with the **hard limit alarm** being displayed on the screen.

After the stroke limit alarm is issued, the user can select MANUAL operation mode and then press the **axis move key** that moves the carriage in the reverse direction to exit the stroke limit, then the alarm displayed on the screen will disappear.

#### 【Explanation】

- 1) The positive stroke limit detection of X, Y and Z axes shares the same pin +TL, and the negative limit detection shares the same pin -TL; If a positive hard limit alarm occurs, all the axes cannot be moved in positive direction other than in negative direction; and vice versa.
- 2) When the limit switch hits the limit stopper, the limit signal is generated; The length of the effective part of the hardware stopper signal should be greater than 300 mm, to prevent the signal effective area being exceeded.
- 3) If the "limit emergency stop" mode is set by bit parameter **P402\_d7=1**, there may be a large error between the coordinates displayed on the system and the actual ones after the limit stopper is hit. In such a case, the machine coordinates must be readjusted.

**【Related parameters】**

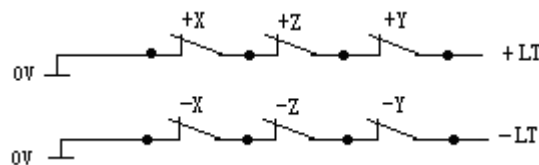
Bit parameter: **P402\_d7, P404\_d6, P404\_d1.**

Therein: Bit parameter **P402\_d7** sets the modes of the hard limit alarm;

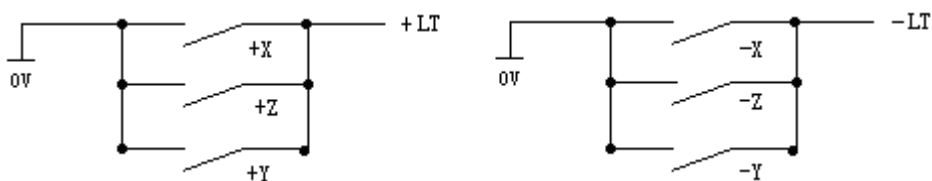
Bit parameter **P404\_d6** sets whether the hard limit alarm is detected;

Bit parameter **P404\_d1** sets the level of the hard limit alarm for each axis.

When bit parameter **P404\_d1=1**, the alarm is set to high level. The positive limit switch +X, +Y or +Z of each axis should be a normal closed contact, and connected to X/Z/Y positive limit input interface +LT in series (an alarm occurs if open ); the negative limit switch -X, -Y, -Z of each axis should also be a normal closed contact, and connected to X/Z/Y negative limit input interface -LT in series; it is recommended that the user select a normal closed contact for the hard limit of each axis. The connection is shown as the figure below:



When bit parameter **P404\_d1=0**, the alarm is set to low level. The positive limit switch +X, +Y or +Z of each axis should be a normal open contact, and connected to X/Z/Y positive limit input interface +LT in parallel (an alarm occurs if closed ); the negative limit switch -X, -Y, -Z of each axis should also be a normal open contact, and connected to X/Z/Y negative limit input interface -LT in parallel; The connection is shown as the figure below:



4.1.4.2 Soft Limit Protection

- 1) Mechanical soft limit protection

Mechanical soft limit protection is used to limit the movement range of the machine coordinates, in order to prevent the slide carriage from moving beyond the range. If the machine position (machine coordinates) exceeds the range, the **mechanical soft limit alarm** is issued.

Method of releasing the overtravel alarm: In MANUAL operation mode, move the carriage in the opposite direction (e.g. if the positive overtravel occurs, move the carriage in the negative direction; and vice versa).

#### 2) Tool nose soft limit protection

Tool nose software limit protection is used to limit the movement range of the tool nose coordinates, to prevent the range from being exceeded by the tool nose. If the tool nose position (tool nose coordinates) exceeds the range, the **tool nose soft limit alarm** is issued.

Method of releasing the overtravel alarm: In MANUAL operation mode, move the carriage in the opposite direction (e.g. if the positive overtravel occurs, move the carriage in the negative direction; and vice versa).

#### 【Explanation】

During the movement, if the coordinates exceed the software limit range, the axes will decelerate to stop.

#### 【Relative parameters】

**P009, P010:** The max. stroke of tool nose software limit in the positive/negative direction of Z axis;

**P011, P012:** The max. stroke of tool nose software limit in the positive/negative direction of X axis;

**P013, P014:** The max. stroke of tool nose software limit in the positive/negative direction of Y axis;

**P015, P016:** The max. stroke of mechanical software limit in the positive/negative direction of Z axis;

**P017, P018:** The max. stroke of mechanical software limit in the positive/negative direction of X axis;

**P019, P020:** The max. stroke of mechanical software limit in the positive/negative direction of Y axis;

Bit parameter **P404\_d4** and **P404\_d3** respectively set whether the mechanical and tool nose software limit alarms are valid.

#### 4.1.4.3 Emergency Stop Alarm (Stopping System Emergently)

There is an external emergency stop input terminal ESP among the system interfaces. The user should connect the **normal closed contact** of the red mushroom emergency button on the panel to the terminal. In case of emergency, press the **Emergency Stop button** to cause the system to enter emergency stop state, then the system will stop all the feed and turn off the spindle and cooling, issuing the alarm “Emergency Stop” (If there are other popups displaying on the screen, the emergency stop function is executed firstly, then the alarm “Emergency Stop” is displayed later).

After the cause of emergency stop is removed, press the Emergency Stop button in the direction of arrow, then the button will lift automatically, cancelling the emergency stop signal.

When the system is in the emergency stop state, if the external emergency stop signal has been cancelled, press RESET key to exit the state and return to the operation mode before emergency stop.

When the system is in the emergency stop state, if the external emergency stop signal is not cancelled, it is forbidden to operate the system in Manual, Auto and Diagnosis operations; but the user



can press **RESET** key to remove the alarm window, and switch the control to Edit, Parameter or Tool Offset operation mode; if the emergency signal is not cancelled in Edit, Parameter or Tool Offset operation, press **RESET** key to remove the alarm window, then it is available to operate system.

If Emergency Stop occurs during movement, there may be a large error between the system displayed coordinates and the actual ones. Thus the user must readjust the machine coordinates. When the Emergency Stop button is pressed, the system will execute a series of emergency treatment, which is shown as follows:

- 1) During emergency stop, the system stops all the feed; program execution stops; the spindle stops and the cooling and lubricating are turned off.
- 2) During emergency stop, the system automatically sets the states of chuck and tailstock saved in the memory to M11 and M79 respectively. After the emergency stop is released, the states of the system chuck and tailstock are M10 and M78 respectively when their foot switches are pedaled for the first time.
- 3) During emergency stop, if the system is executing commands such as tool change, chuck, tailstock or gear shift, the execution will be stopped immediately (tool post positive/negative rotation signal is cancelled, and whether chuck and tailstock are cancelled depends on the parameter setting); here, the system assumes that the tool post, chuck, tailstock and gear are in uncertain positions, and issues red flashing. Only after the emergency stop alarm is released, can the system recover to the normal state by re-performing the operation successfully, or by turning off the system and then on.
- 4) During emergency stop, the interface output signal of MDLY keeps unchanged. How to process the interface control signals other than those of MDLY, spindle, coolant and lubricant is determined by parameters. If **P403\_d3** is set to 0, only the output signals of the spindle, coolant and lubricant are turned off; if **P403\_d3** is set to 1, all the interface control signals including the chuck and tailstock are turned off.
- 5) After the emergency stop alarm is released, if the system is in the Dry Run state in Auto operation mode, it will exit the Dry Run state; if the system is in Rapid Traverse state in MANUAL operation mode, it will automatically enter Feed state; the set F value keeps unchanged; and the spindle analog voltage keeps unchanged; the states of other functions except the above are in program initial states.
- 6) After the emergency stop alarm is released, the time counting of low-pressure detection function and automatic lubricating control function will be restarted.

#### **【Special attentions】**

- 1) The standard Emergency Stop function executed by the system is the function of turning on or turning off the system output signals uniformly. The user can set it as follows: In Manual/Auto operation mode, after Emergency Stop button is pressed, and the system executes the standard Emergency Stop function, the system executes an additional M74 custom command once (**only when in MANUAL/AUTO operation mode and when there is a solidified M74 command in the system, the execution is available**). This function is applicable to some special machine components and is used only when some of the output signals are required to turn off and some

maintained. If there is an alarm occurs during the execution of the M74 custom command, the execution is stopped. After the emergency stop button is pressed, if the solidified M74 custom command is being executed, the system will automatically terminate the command when it proceeds to an axis move command or tool change command,

- 2) In Manual/Auto operation mode, when the M74 custom command can be executed in Emergency Stop state by setting a parameter (**P412\_d1=1**), there is a prompt “+M74” ( indicating the M74 is being executed) added on the emergency stop window (if there is a solidified M74 custom command in the system).
- 3) For the programming, debugging and solidification of M74 custom command, please see Chapter Eight “Custom Command Programming” in this manual.
- 4) Be careful to use M74 function during Emergency Stop. The function is only applicable to some special machine tools.

#### 【Relative parameters】

When bit parameter **P404\_d7** is set to 0, the external emergency stop signal is valid; when it is set to 1, the signal is invalid.

When bit parameter **P403\_d3=0**: If the emergency stop alarm is valid, the system only turns off the output signals of the spindle, cooling and lubricating.

When bit parameter **P403\_d3=1**: If the emergency stop alarm is valid, the system turns off the output signals of all auxiliary functions.

Parameter **P404\_d7** is set for the convenience of system debugging. In the on-line state, it must be set to the valid state. Otherwise, it cannot provide protection.

Parameter **P412\_d1** sets whether M74 custom command is executed in the emergency stop state.

#### 4.1.4.4 Drive Unit Alarm

When the CNC system is connected to the alarm output signal of a drive unit, and the drive unit alarm is issued, the system will automatically stop all the feed, and prompt **Z/X axis drive unit alarm** on the screen. All the axes, as well as the program execution, are stopped immediately. Here, check the drive unit and relative components to remove the fault and then power on the system after power-off.

If the alarm occurs during movement, there may be a large error between the system displayed coordinates and the actual ones. Therefore, the user must readjust the machine coordinates.

In MANUL operation mode, if the alarm occurs, all the axes movement is stopped.

In AUTO operation mode, if the alarm occurs, it is forbidden to execute the program.

#### 【Relative parameters】

When bit parameter **P404\_d5** is set to 0, the drive unit alarm is detected.

Bit parameter **P405\_d4**, **P405\_d3** and **P405\_d2** set the alarm level of the drive unit respectively for Z, X and Y axes.

#### 4.1.4.5 Other Alarms

When other alarms are issued in the system, they are displayed on the screen. Here, the user can handle them according to prompts and the countermeasures described in Chapter Six Alarm Message.

#### 4.1.4.6 Power Off

In case of emergency during machine running, the user can turn off the machine power immediately, so as to prevent accidents.

However, if the power is turned off during coordinate axes movement, please note that there may be a large error between the system-displayed coordinates and the actual ones after power-on. The user must readjust the coordinate coordinates by performing machine zero point return or other means, in order to make the system displayed coordinates coincide with the actual ones.

#### 4.1.4.7 Reset Operation

When abnormal system output or coordinate axis move occurs, press **RESET** key to cause the system to enter Reset state as follows:

- 1) All axes decelerate to stop.
- 2) Bit parameter **P403\_d2** sets whether M function (spindle, coolant) output is valid.
- 3) Automatic operation ends, and modal functions and states are kept.
- 4) The system is in G00, G97, G98 state; F value, as well as the spindle analog voltage output, keeps unchanged.
- 5) The system terminates the operation in progress, and returns to the initial interface of the current operation mode.

#### 【Special attentions】

- 1) The standard Reset function executed by the system is the function of turning on or turning off the system output signals uniformly. The following function can be set: In Manual/Auto operation, after Reset button is pressed, and the system executes the standard Reset function, the system executes an additional M73 custom command (**only when in Manual/Auto operation mode and when there is a solidified M73 command in the system, the execution is available**). This function is applicable to some special machine components and is used when only some of the output signals are required to turn off and some maintained. If there is an alarm occurs during the execution of the M73 custom command, the execution is stopped.
- 2) In Manual/Auto operation mode, when the M73 custom command can be executed in Emergency Stop state by setting a parameter (**P412\_d2=1**), there is a prompt "+M73" ( indicating the M73 is being executed) added on the reset window (if there is a solidified M73 custom command in the system).
- 3) For the programming, debugging and fixing of M73 custom command, please see Chapter Eight "Custom Command Programming" in this manual.
- 4) Be careful to use M73 custom function during Reset. The function is only applicable to some special machine tools.

#### 【Relative parameters】

When bit parameter **P403\_d2=0**: After pressing Reset key, the system will turn off the output signals of the spindle and cooling.

When bit parameter **P403\_d2=1**: After pressing Reset key, the system will keep the output states of the spindle and cooling before Reset.

Parameter **P412\_d2** sets whether M73 custom command is executed during Reset.

## 4.2 Operation Mode Selection for CNC System

The operation modes are directly selected by the operation mode keys in this system. It is available to switch different modes directly, thus realizing simple, convenient and visible operation.

The screen shown in Fig. 4-1 is displayed after system power-on. If no key is pressed, the screen keeps unchanged. When a key on the panel is pressed, the system enters Edit operation mode.

## 4.3 Edit Operation Mode

**Edit operation mode** is to operate the workpiece programs by using the system operation panel. For each step, the system provides corresponding intelligent prompt messages. Also, the user can press the hp2 key on the upper right corner of the system to view the list of the system operation keys in Edit operation mode.


The explanations for the input format and examples about related setting and operation in the manual are as follows: The meanings and use of the function keys to be pressed are described at the beginning of the manual; the letter keys, numeric keys to be input are indicated by an underline; the prompt messages of the system are indicated by a rectangle.

When setting or inputting an item, or during the man-machine dialogue, if ESC key is pressed before confirmation, the current operation is quit.

### ◆ The major functions of the Edit operation mode are:

- ☆ To select, add, rename, copy and delete the workpiece programs;
- ☆ To input, insert, modify and delete the contents of the workpiece programs selected;
- ☆ To transfer workpiece programs between U disk and system through the USB interface.
- ☆ To transfer workpiece programs between external computer and system through the RS232 communication interface;
- ☆ To transfer workpiece programs between two CNC systems through the RS232 communication interface;
- ☆ To compile and save programs
- ☆ To input variables and macro character strings



Press the operation mode select key  to enter EDIT operation mode. The EDIT operation mode consists of two main pages, which are the program directory search page and program editing page. The program directory search page is shown as Fig. 4-2.

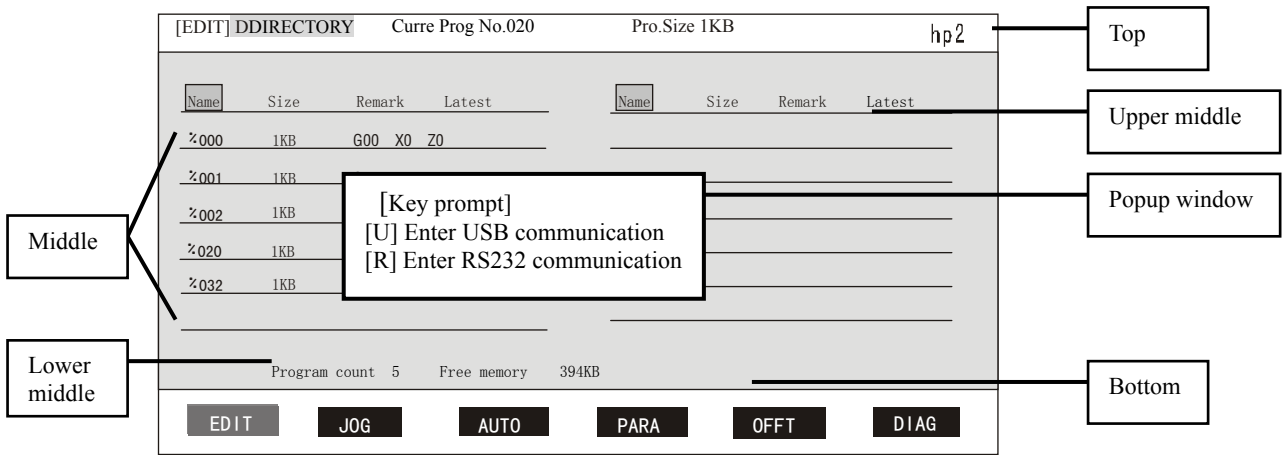


Fig. 4-2 Program directory search

◆ Explanations for the contents displaying on each area of the page

- Top:** Current program number and occupied storage capacity (program size), prompt key **hp2** for system function operation method;
- Upper middle:** The system arranges the programs in terms of the name, size, remark or latest.
- Middle:** Displaying the program names, sizes and remarks saved in the system
- Lower middle:** Operation prompt messages
- Bottom:** Displaying the number of programs stored in the system (up to 255) and the remaining storage capacity for programs.
- Popup window:** Displaying the operation prompt messages.

**【Note】**




When pressing **hp2** key, the system prompts “Program directory interface message prompt”, which introduces the functions of the keys to be used.

4.3.1 Part Program Directory Search

The part program directory search page displays the number of programs stored in the system. All the programs can be arranged in the following four orders:

- 1) Sort by name: Sorting by the program numbers from the top to bottom, and from the left to right;
- 2) Sort by size: Sorting by the size of the programs, from top to bottom, and from left to right;
- 3) Sort by remarks: Sorting by the size of the first 12 strings in the first line of the programs, from top to bottom, and from left to right;
- 4) Sort by latest: Sorting by the time that the programs are input, from top to bottom, and from left to right.

The user can retrieve all the programs by pressing  and  cursor keys and  and  page keys. 12 programs can be displayed on one screen; pressing  key turns the

screen to the previous page, and pressing  turns the screen to the next page; Pressing  or  key sorts the programs by name, size, remark or latest.

### 4.3.2 Selecting, Creating, Deleting, Renaming and Copying a Part Program

It means the operation of selecting, creating, deleting, renaming or copying a part program.

#### 【Note】

1. Up to 255 programs can be operated, from number %000~%254. When operating a program number bigger than 254, the system prompts "**E160 Input program number error**".
2. If there are no part programs in the system, or it is the first time that the system is used, the system automatically creates and selects No. %000 program as the current program. If there are part programs in the system, the system arranges the programs which existed before last power-off according to their numbers.
3. The system supports multiple inputs, with the leading zero omissible. E.g. inputs the program with name %003. Press **INPUT** key, and then input **0 0 3**; or **0 3**; or **3**.

#### 4.3.2.1 Selecting and Creating a Part Program

The steps of selecting a part program or creating a new part program are as follows:

- ① Press **INPUT** key in EDIT operation mode;
- ② Input the program number to be selected using the keyboard, or input a program number not included in the program directory as a new program number;
- ③ Press **ENTER** key;
- ④ Then the part program selection or creation is completed. The screen displays the contents of the program, and the system enters the program edit page.

#### 【Note】

- 1) After selecting a program, the user can change the desired program only through the above steps; once a program is selected, it keeps unchanged even if the power is turned off.
- 2) If the input part program does not exist in the program directory, a new program with the input program number is created and taken as the current program.

#### 【Example】

Example 1: The procedures of creating a part program with number %20 when it is not included in the program directory:

Press keys: **INPUT 2 0 ENTER**. The new program with name %020 is created, and the system enters program edit page.

Example 2: Procedures of selecting the part program with number %001 when it is included in the program directory:

Press keys: **INPUT 1 ENTER**. The program with number %001 is selected, and the system enters program edit page.

#### 4.3.2.2 Deleting a Part Program

The procedures of deleting a part program are as follows:

- ① Press **INPUT** key in Edit operation mode;
- ② Input the program name to be deleted using the keyboard;
- ③ Press **DELETE** key, then the system prompts: **ENTER-Confirm Deletion ESC-Exit Deletion**;
- ④ Press **ENTER** key to delete the part program whose number is input; or press **ESC** key to cancel the deletion, and returns the control to the Edit operation mode.

##### 【Note】

- 1) If the program to be deleted does not exist, the system prompts "**E100 The program to be deleted does not exist**";
- 2) If the program to be deleted already exists: when it is not the current program, it is deleted from the program list; when it is the current program, it is deleted from the program list, and the system takes the program with the smallest number on the list as the current program; when there are no programs existing after deletion, the system automatically creates an empty program with name 000 as the current program.



##### 【Example】

Example: The procedures of deleting the program with number **%003** are as follows:

Press keys: **ENTER 3 DELETE ENTER**. The program with number **%003** is deleted from the part program storage area.

#### 4.3.2.3 Deleting All Part Programs

It is the operation of deleting the program area in the program directory search page, i.e. deleting all the programs once; the procedures are as follows:

- ① Press **INPUT** key in the state of part program directory search;
- ② Input  ,  from the keyboard;
- ③ Press **DELETE** key, then the system prompts: **ENTER-Confirm to delete all programs ESC-Escape deletion**;
- ④ Press **ENTER** key to delete all the programs; press **ESC** key to cancel the deletion, and return the control to the Edit operation mode.

##### 【Note】

After all the programs are deleted, the system will create an empty program with a name 0000 as the current program.

#### 4.3.2.4 Renaming a Part Program

The part program renaming is to replace the name of the current program by a new one. The new program is taken as the current program. The procedures are as follows:

- ① Press **INPUT** key;

- ② Input a program number which does not exist in the program directory, and press **ALTER** key to change the current program name to the one input.

**【Note】**

If the input program number already exists, the system prompts "**E166 The program to be renamed already exists**". Input another name after the prompt disappears.

**【Example】**

Example: The procedures of renaming the current program %000 as %005 are as follows:

Press keys: **INPUT** **5** **ALTER**. Then the renaming is completed.

## 4.3.2.5 Copying a Part Program

It is to copy the contents of the current program to a new program. The newly created program is taken as the current program. The procedures are as follows:

- ① Press **Input** key;
- ② Input a nonexistent program name as a new program, then press **INPUT** key to copy the contents of the current program to the new program. The new program is taken as the current program.

**【Note】**

If the input program name already exists, the system prompts "**E161 The program to be copied already exists**". It is available to input another program name again after the prompt disappears.

**【Example】**

Example: Procedures of copying the current program %000 to program %005 are as follows:

Press keys: **INPUT** **5** **INPUT**. Then the copy is completed.

## 4.3.3 Part Program Communication

The communication of part programs consists of part program sending and part program receiving. The part programs can be sent from system to computer (CNC→PC), from system to U disc (CNC→USB), or between two CNC systems (CNC→CNC); in addition, the system can receive the part programs from a computer (PC→CNC), from another system(CNC→CNC), or from a U disc (USB→CNC).

When pressing **hp6** key on the program directory search page, the system prompts the part program communication interface.

## 4.3.3.1 Sending Part Program (CNC→PC, CNC→USB, CNC→CNC)

**Mode 1: RS232 serial communication**

- 1) Before transferring the files, set the communication baudrates of parameter **P414\_d7** and **P414\_d6**. The communication baudrate is decided by the setting of the sender. Setting range:



9600, 19200, 38400 (unit: bps). Factory standard setting: 9600 bps. For how to set the baudrate, see Section 4.6 “Parameter Operation Mode” in Part I Programming.

- 2) Output the part programs stored in the system to an external computer or CNC system:
  - ① Connect the communication cable with the system power OFF, then turn on the system.
  - ② On program directory search page, press keys: **hp6** → **R** → **hp2** in order;
  - ③ Select the programs to be sent according to the help information of **hp2**.
  - ④ Press **ENTER** key to send the programs;
  - ⑤ Enable the external computer or CNC system to remain in receiving state; (For details, see Chapter Five “RS232 and USB System Communication”)
  - ⑥ The system sends the selected programs, and displays the progress bar at the same time till the sending is completed; if multiple programs are selected, the system sends the programs one by one in the ascending order according to their program numbers.
  - ⑦ It is possible to interrupt the sending by pressing **ESC** key.

**Mode 2:** USB communication;

- 1) After entering USB communication mode, the system first checks whether a U disc has been inserted. If no U disc is inserted, the system displays a prompt message box, prompting **No USB device inserted**. If a USB disc containing a folder named “C001PRO” has been inserted, and there are CNCxxx.TXT files in the folder, the files will be listed on the USB file directory box. If there are no CNCxxx.TXT files in the C001PRO folder, the system prompts: **USB device specified directory: C001 PRO has no programs**. If the folder named “C001PRO” does not exist in the U disc, the system prompts: **No specified directory in USB device: C001 PRO**.
- 2) Output the part programs stored in the system to the U disc:
  - ① Insert an UBS disc into the USB interface of the system;
  - ② The system automatically opens the file directory of the U disc (when the U disc contains a folder “C001PRO”, and the folder contains files “CNCxxx.TXT”). If the folder named “C001PRO” does not exist in the USB root directory, the system will create one.
  - ③ On program directory search page, press keys: **hp6** → **U** → **EDIT** in order;
  - ④ Select the programs to be sent according to the help information of **hp2**;
  - ⑤ Press **ENTER** key to send the programs, and select the send mode according to the system prompts.
  - ⑥ The system sends the programs according to the selected mode, and displays the progress bar till the sending is completed;
  - ⑦ Press **ESC** key to exit the U disc.

#### 4.3.3.2 Receiving Part Programs (PC→CNC, USB→CNC, CNC→CNC)

**Mode 1:** RS232 serial communication;

- 1) Set the baudrate and communication port of the communication software; (for details, see Chapter Five “RS232 and USB System Communication” in Part I Programming)

- 2) Input the part programs stored in the external computer to the system; or transfer programs between CNC systems; the procedures are as follows:
  - ① Connect the communication cable with the system power OFF, then turn on the system;
  - ② On the program directory search page, press keys: **hp6** → **R** → **EDIT** in order;
  - ③ Select the receive mode according to the system prompts;
  - ④ After confirming the system is in receive state, input the programs stored in the external computer or CNC system to the CNC system.
  - ⑤ The system inputs the programs according to the selected receive mode, and displays the progress bar until the receiving is completed;
  - ⑥ It is possible to stop receiving programs pressing **ESC** key in the progress of receiving.

#### Mode 2: USB communication;

- 1) After entering USB communication mode, the system first checks the U disc, and then opens the folder “C001PRO” to list the “CNCxxx.TXT” files in the folder.
- 2) Output the part programs stored in the U disc to the CNC system:
  - ① Insert the U disc into the USB interface of the system;
  - ② On the program directory search page, press keys: **hp6** → **U** in order;
  - ③ The system automatically opens the file directory of the U disc;
  - ④ Select the programs to be received according the help information of **hp2**;
  - ⑤ Press **ENTER** key to receive programs, and select the send mode according to the system prompts;
  - ⑥ The system receives part programs according to the selected receive mode, and displays the progress bar until the receiving is completed;
  - ⑦ Press **ESC** key to exit the U disc.

#### 【Note】

If the name of a program to be sent to the system already exists in the system, the system prompts whether to replace the original one. Once replaced, the original program will be replaced by the sent program.

#### 4.3.3.3 Standard Format of TXT Part Program on PC

On a personal computer, the part program can be edited using TXT or LST text. However, the file name and its contents must be edited in the standard format required in the system, so that the program can be correctly sent to the system. The specifications are as follows:

- 1) On a personal computer, the user should name the part program file with suffix TXT or LST, e.g. “CNC008.TXT”; It is recommended that the user use suffix TXT for the convenience of the part program operation on the PC.
- 2) The first line of the TXT file must indicate the program number, in the format of “% XXX”, i.e. an one-digit, two-digit or three digit number behind the percent, within a range of 0~254. No other contents can be contained in the first line. The range of the program number must be within 0~254. Otherwise, the system fails to receive the programs and prompts corresponding error

messages. The program numbers are determined by the following two modes (USB and RS232) after the CNC receives the programs:

- ◆ In system RS232 communication, the program name to be stored is subject to the program number of the first line; i.e. the Arabic numerals XXX of the string “%XXX” in the first line of the program sent by the PC.
- ◆ The number of the program transferred by the USB is the Arabic numerals xxx of CNCxxx.TXT in the folder “C001PRO” in the U disc root directory.

**Note:** In system USB communication, The Arabic numerals XXX in string “%XXX” of the first line in the program should be the same as the Arabic numerals XXX in CNCxxx.TXT in the folder “C001PRO” of the U disc root directory.

- 3) The second line and the following are blocks. The format of a block must accord with the requirement, and each block, ended with ENTER key, cannot contain more than 250 characters. Otherwise, the system reports an error message “Block overlong in received program”
- 4) The comment area of the block can contain comments in Chinese.
- 5) The size of the TXT file cannot exceed the limit of the program storage capacity of the system.

Standard format of part program communication on PC:

TXT file format	Explanation
%099 N0000 G50 X100 Z100 ; Setting coordinate system	<b>1. When the system is receiving a program, the program name %099 cannot be omitted; the first line must be a three-digit number between 0~254.</b> <b>2. N**** is the line number of a block. Blocks without N**** have no line numbers.</b> <b>3. The beginning of each line is a space.</b> <b>4. For the block with a line number, there is a space between the line number and the command.</b> <b>5. / indicates the block is skipped;</b> <b>6. ; The contents behind it are comments.</b>
N0010 G00 X20 Z90 ; Rapid positioning G01 X10 Z80 ; Linear cutting	
..... /N0250 G02 X30 Z20 R5 ; Circular cutting	
N0260 ; T22 ; Tool change	
N0262 M05	
N0270 G04 D8 ..... M20	

#### 4.3.4 Inputting and Editing the Contents of Part Program

The input part program consists of a number of blocks, and each block includes a block number, commands, data, etc. The format of the program should be consistent with the general programming rules described in Part II Programming, and no alarm should occur during program compiling. For the alarm prompts, see Chapter Six Alarm Message in Part II Programming. Only after the correct program contents are input according to order of the technological requirements, can the machine tool produce qualified parts.

The system employs the full-screen edit mode, and the program edit page is shown in Fig. 4-3:

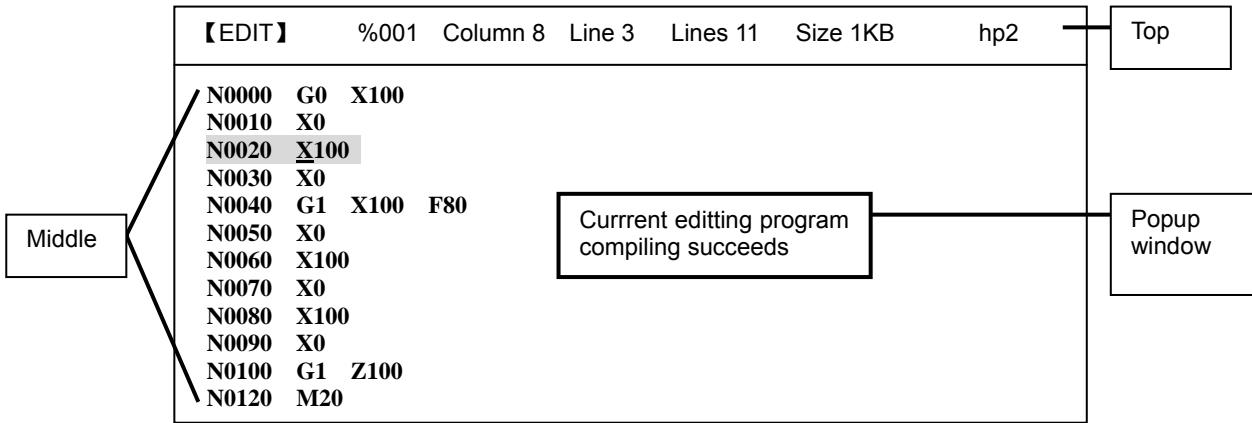


Fig. 4-3 Program editing

◆ Explanations for each area of the interface

Top: Program number and program size of the current program, number of lines, line and column at which the edit cursor (prompt symbol of editable character on current position ) is located, prompt key hp2 for system function operation;

Middle: Program edit window;

Popup window: Displaying prompt information for operation.



**【Note】**

1. After pressing hp2 key, the system prompts “Program editing Help message prompts”, introducing the functions of the help keys to be used.
2. When bit parameter P416\_d0 is set to 1, it is forbidden to edit or alter a program, and the system displays alarm message **E174 Part programs are locked, and forbidden to modify**; when it is required to edit and alter a program, set the bit parameter P416\_d0 to 0.











**Meanings and use of edit keys on program edit page**


- 1)  ,  cursor up and down keys


By pressing a cursor key once, the cursor moves up (or down) vertically till it reaches the uppermost (or lowest) line, taking the current cursor column value as the reference movement value. By holding the cursor key, the cursor moves up (or down) continuously till the first (or last) block is reached, or till the key is released. When using the string search function (hp5), the two keys are used for retrieving the desired string up and down.


- 2)  ,  Cursor left and right move keys

By pressing a cursor key once, the cursor moves one character towards left (or right). By holding the cursor key, the cursor continuously moves to the left (or right) till the first (or last) character of the block is reached, or till the key is released.

- 3)  Line Home key: The cursor rapidly moves to the beginning of a line or the beginning of the first word of the line. When repeatedly pressing the Line Home key, the cursor location is switched between the beginning of a line and the first word of the line. By pressing the Line Home key and Delete key together, the current line is deleted.
  
-  Line End key: The cursor rapidly moves to the end of a line.
  
- 4)  Insert/Alter key:  
 Changing the edit input mode: By pressing the ALTER key once, the input mode toggles between Insert and Alter, with the cursor displaying in different shapes. In Insert mode, the cursor is a flicking underscore, while in Alter mode, the cursor becomes a flicking and highlighted block cursor.
  
- 5)  Input key: It switches the program edit state to the program directory search state, displaying `Please input program number;`.
  
- 6)   Page up and Page down keys  
 They are used for turning pages to display programs. When used in the function of `hp5` key, they move the cursor to Home/End page; when used in the function of `hp3` key, they are used for other purposes.
  
- 7)  Enter key:  
 It inserts a new line when the cursor is located at the beginning or the end of a line, with the cursor moving to the new line; when the cursor is located in other position, it moves to the beginning of the next block by pressing `Enter` key.
  
- 8)  Delete key: It is for deleting a whole program or the characters in a block.
  
- 9)  `hp2` key: Help information for program editing.
  
-  `hp3` key: Current program compiling.

10 )  **hp4** key: On program edit page, the user can control the movement of the cursor using a MPG, browsing the programs rapidly.
















 **hp5** key: Help for system commands, part program string search, etc. If the program number of the current program is 253 or 254, the operation of the program with number 253 or 254 is added.

 **hp6** key: It prompts to display macro string list.

**Rules for multi-function definition key**

- 1) If the first letter in a line is a capital, the first key has the priority; if the first letter is lowercase letter, the third or second key has the priority.
- 2) If a letter or string is input behind a number (0~9), a space between them will be created automatically.
- 3) After a string is input, the cursor stays in the position that is the most convenient for inputting.

Table of values for multi-function definition keys

Panel display	First key value	Second key value	Third key value	Panel display	First key value	Second key value	Third key value
	G	r			F	N	( )
	M	H	=		I	P	
	X	J	>		K	Space	
	Z	Q	<		D	V	and
	S	if			R	Y	or
	T	L	then		.	*	
	U	E	else		-	+	
	W	/	;				


4.3.4.1 Inputting Program Contents

**Note:** Set the value of parameter P333 to 10 (program number automatic creation, similarly hereinafter)

On program edit page, input the contents of a part program by the following steps:

- ① Create a new program by means of creating a new part program;
- ② After the block number **N0000** is displayed, input a line of program contents from the keyboard;
- ③ After the input, press **ENTER** key to end the input of this line;
- ④ The system automatically creates the number of the next block for user to continue inputting;
- ⑤ After the last line of the program is input, press **ESC** key to end the input of the program.

#### 【Note】

- 1) The first column of each line must be the space;
- 2) Up to 60 characters can be displayed for one block. If there are more than 60 characters, the remaining ones are not displayable. If the cursor is at the end of a line, the cursor moves a character towards left by pressing  key.
- 3) The first column of each line is a space, which is created automatically by the system; the number of the first column is 1, and that of the last column is 252; Only the cursor rather than the character can be displayed at the first and last column; up to 250 characters can be edited in one block.

#### 4.3.4.2 Inserting a Block

Insert a block or multiple blocks between two blocks; or insert a block or multiple blocks before the current block. The procedures are as follows:

- ① For two adjacent blocks, move the cursor to the end of the first block or to the beginning of the second block;
- ② By pressing **Enter** key, the system automatically creates a new block number between the current block and the next block (the increment of the sequence number is the integer which is 1/4 of the value set in parameter P333. If the numbers are not enough, modify the number of the next block.) and leaves one blank line.
- ③ Input the desired contents to the block;
- ④ After the input, if it is necessary to insert multiple blocks, press **ENTER** key; if only one block is required to insert, this operation is unnecessary.

#### 【Example】

Example: The operation procedures of inserting a new block **M3** between block **N0020** and block **N0030** are as follows:

Move the cursor to the end of block **N0020** or to the beginning of block **N0030**; Press **Enter** key, then input **M3** for the new block.

#### 4.3.4.3 Deleting a Block

The procedures of deleting all contents (including the block name) in one line are as follows:

- ① Move the cursor to the beginning of the block to be deleted;
- ② Press **DELETE** key;
- ③ Delete all the contents of the selected block.

#### 4.3.4.4 Inserting a Character in a Block

The procedures of inserting a character in a block are as follows:

- ① Pressing **ALTER** key to switch the input mode to Insert mode, i.e. the cursor is displayed as an underscore;
- ② Move the cursor to the character behind the position where the contents are to be inserted;
- ③ Input the desired contents
- ④ Insert the desired contents before the character where the cursor is located.

#### 【Note】

According to the requirement of the system, a space must be left between two words (a word means a letter followed by digits) to separate them. The space can be created automatically during program editing. However, it may not be created during insertion operation. In this case, the operator is required to input the space, in order to keep the program complete.

#### 【Example】

Example: The procedures of inserting **1** between **X** and **0** in the block **N0020 GO X0.0 Z0.0** are as follows:

Move the cursor to the bottom of **0** behind **X**, then input **1**. Then the screen displays **N0020 GO X10.0 Z0.0**

#### 4.3.4.5 Deleting a Character in a Block

The procedures of deleting characters in a block are as follows:

- ① Move the cursor to the character to be deleted;
- ② Press **DELETE** key to delete the character where the cursor is located.

#### 4.3.4.6 Altering Contents of a Block

Change the contents of a block. There are two methods depending on the input mode (Insert/Alter)

In **Insert mode**: Procedures of altering a block using the combination of insertion and deletion:

- ① Move the cursor to the character to be altered;
- ② Input the new contents;
- ③ Press **DELETE** key to delete the unnecessary contents;

In **Alter mode**: Procedures of directly altering the contents where the cursor is located:

- ① Press **ALTER** key to switch the cursor to Alter mode (the character where the cursor is located is highlighted by a rectangle);
- ② Move the cursor to the character to be altered;
- ③ Input the new contents, then the cursor moves to the next character.

#### 【Example】



Example: The procedures to alter **X** in block **N0020 G0 X0.0 Z0.0** to **U** are as follows:  
Switch the input mode to **ALTER** mode, move the cursor to the bottom of **X**, and then input **U**.  
After alteration, the block becomes **N0020 G0 U 0.0 Z0.0**.

#### 4.3.4.7 Inserting a Macro String

On program edit page, the procedures of inserting a macro string are as follows:

- ① Press **hp6** key to display on the screen the macro string list;
- ② Select the contents to be input by pressing keys according to prompts.

##### 【Example】

Example: Press **hp6** key, then **G** key, and then input program contents: **r = r \* r / r**. Here, the cursor stays behind **r**.

##### 【Note】

For details about the variables and statement programming, please see Chapter Seven Statement Programming in Part II Programming.

#### 4.3.4.8 Storage Capacity for Programs

The system provides 400k storage capacity for storing programs from No.0~ No.252, and No.254. Therefore, the size of a single program can reach 400K theoretically. For No.253, the system provides another 4 M storage space.

##### 【Explanation】

- 1) The screen displays the remaining program storage capacity, as well as the remaining storage size of the system program area.
- 2) If the size of the program (No. 0~252, 254) being edited currently is greater than the max. storage capacity (400k), the program cannot be saved. and the system prompts that the program storage space is full: **Edit area overflow**. If the remaining storage space is insufficient, delete some old programs.
- 3) The edit storage space for No. 253 program is 4 M, but the program cannot be saved.
- 4) If the size of a program is large, it takes longer time to save the program.

#### 4.3.4.9 Operating No. 253 Program

As the size of No. 253 program can be up to 4M, the operation of this program is special. The procedures are as follows:

- 1) The No. 253 program cannot be saved into the system. It is lost after power-off.
- 2) When No. 253 program is changed to other programs, the No. 253 program cannot be saved neither, i.e. it will be lost;
- 3) The communication of No. 253 program is unavailable by using USB or RS232 mode.

##### 【Note】

- 1) No.253 program cannot be saved.

- 2) No.253 program cannot be copied or renamed.

#### 4.3.4.10 Operating No. 254 Program

After selecting No. 254 program on program edit page, press **hp5** key, then the help information about how to edit, solidify and read No. 254 program appears. The operation procedures are as follows:

- 1) Key **5**, for editing and solidifying No. 254 program:  
When compiling No. 254 program, if the compiling fails, the system issues an alarm; if the compiling succeeds, the program is saved to the solidification area (FLASH)
- 2) Key **4**, for extracting No. 254 program:  
It reads No. 254 program saved in the solidification area (FLASH) to the edit buffer area, to update the program.
- 3) Key **6**, for deleting the custom command of No. 254 program  
If there is a custom command in the solidification area, the command will be displayed on the screen. After the custom command in No. 254 program is deleted, there is no custom command in the system solidification area; the help information dialog box for No. 254 program prompts “No custom command in solidification area”
- 4) Press **ESC** key to exit the current state.

#### 4.3.5 Function of **hp5** Key

**hp5** help key includes help for system commands, help for obtaining relative parameters of an arc, rearrangement of line numbers, replacement of strings, cursor location, MPG cursor movement, etc.

If the program number of the current program is 253 or 254, **hp5** help key adds the prompts for the operation of No. 253 program or No. 254 program.

##### 4.3.5.1 Help for Part Program Command

When **hp5** **1** are input on program edit page, the system prompts “Command help introduction” interface; this function is capable of retrieving all the commands of the system, including G, M, S, T, F commands. The operation procedures are as follows:

- 1) Press G, M, S, T, or F key to respectively view the corresponding explanation for the G, M, S, T or F command.
- 2) Press **INPUT** key, then input the command number to be searched; the system displays the meaning, function, format as well as explanation for the command number.

#### 【Example】

Example: Searching the command help for G05

Press keys: **hp5** **1** **INPUT** **G** **05** **ENTER**, then the system prompts the meaning, function, format and explanation for G05.

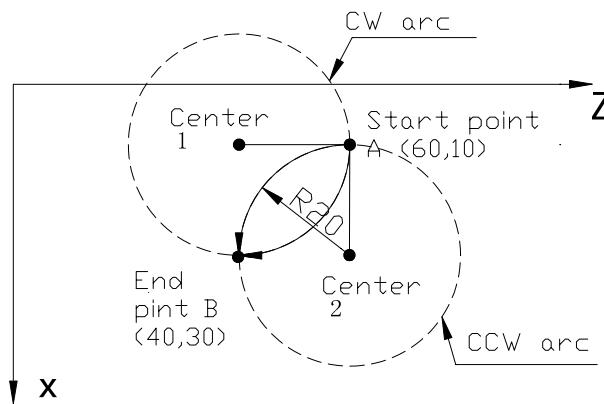
4.3.5.2 Help for Obtaining Relative Parameters of Arc

Input **hp5 2** on program edit page, then the system prompts “Please input relative arc parameters” interface; this function is for obtaining the parameters of an arc.

Input [start point coordinates] , [end point coordinates] , [circle radius ], then the system will automatically calculate the relative parameters of the arc.

**【Example】**

As shown in the figure below: Coordinates of start point A (Z60, X10), coordinates of start point B (Z40, X30).



[Please input relative arc parameters] (Explanation: X direction is the radial programming, **P413\_d6=1** )

[Start point coordinates] Z: 60 X: 10

[End point coordinates] Z: 40 X: 30

[Circle radius] R: 20

After inputting the data above, the system will automatically calculate the following data:

CW circle center 1: Z: 40 X: 10 (The values of Z and X are: the coordinate values of CW circle center 1.)

CCW circle center 2: Z: 60 X: 30 (The values of Z and X are: the coordinate values of CW circle center 2.)

Start point -> Circle center 1: Z: -20 X: 0 (The values of Z and X are: the Z and X vectors of start point A pointing to circle center 1.)

End point -> Circle center 1: Z: 0 X: -20 (The values of Z and X are: the Z and X vectors of end point B pointing to circle center 1.)

Start point -> Circle center 2: Z: 0 X: 20 (The values of Z and X are: the Z and X vectors of start point A pointing to circle center 2)

End point-> Circle center 2: Z: 20 X: 0 (The values of Z and X are: the Z and X vectors of end point B pointing to circle center 2)

#### 4.3.5.3 Rearrangement of Program Line Numbers

Input **hp5 3** on program edit page, then the system rearranges the programs, with the rearranged program numbers increased based on the multiple of 10. (Parameter P333 is set to 10)

##### 【Note】

- 1) After the block numbers are rearranged, if the skip command is used in programming, the program skip error may occur.
- 2) When the value of parameter P333 is 0, the program number rearrangement function is invalid.





#### 4.3.5.4 Replacement of Strings

Input **hp5 R** on program edit page, then the system prompts “String replacement” page; perform the operation according to the system prompts. After the replacement of the string, all the characters to be replaced are replaced from the current character where the cursor is located to the last character.

#### 4.3.5.5 Cursor Position

The string search function provided by the system is used for locating the contents to be searched, which is convenient for the user to search the desired contents. Press **hp5** key on program edit page,

then the system prompts the functions of , , ,  and **E** key as follows:

- 1) Press  key to move the cursor to the first page of the current program.
- 2) Press  key to move the cursor to the last page of the current program.
- 3) After pressing **E** key, input the string to be searched in the current program, then press **ENTER** key. After that, the cursor is located at the searched string, highlighted in red. In inputting the string to be searched, the system automatically records the latest 10 times of string search. If there is no record, nothing is displayed.
- 4) According to the system prompts, press  or  to move the cursor up or down to search the desired string in the current program. If the desired string does not exist in the current program, the system prompts: **Search completed, string not found**

#### 4.3.5.6 Cursor Movement by MPG

After the MPG is connected to the system, press **MPG** key on the program edit page. You can quickly browse the programs by rotating the MPG to control the movement of the cursor when the MPG LED indicator on the operation panel lights up. By pressing **MPG** key again, the LED indicator goes out. In this case, it is unavailable to perform MPG operation. For the connection of MPG, see Part III Connection.

### 4.3.6 Compiling a Part Program

The compiling command key **hp3** in the system is used for compiling a part program, and checking the grammar error, logic error, and whether the coordinate data causes overtravel according to the execution path of the part program, in a bid to reduce the alarm errors when the program is executed in AUTO operation mode, and improve the safety of executing the part program.

By pressing the compiling command key, the system checks and compiles the part program from the first block according to the execution path, and generates an object code which is convenient for execution. If an error in the program is detected, the system stops the compiling, displaying the word position and error number in the block where the source program error occurs, and prompting the user to modify the error.

#### 4.3.6.1 **hp3** Compiling Command

On program edit page, press **hp3** key, the system compiles the current program line by line. If an error message is detected, a window 『Program Alarm 』 pops up. If no error exists in all commands, the system displays: **Current program compiling succeeded**

The information of 『Program Alarm』 consists of:

Error: indicates error codes (View Chapter Six Alarm Message in PART II Programming according to the codes);

Program: indicates contents in the error block;

Position: indicates error letter or word in the error block.

#### 【Explanation】

- 1) Only those programs successfully compiled by **hp3** can be executed in Auto operation mode.
- 2) The system automatically completes the compiling when the control is switched from edit operation mode to other operation modes.
- 3) After 『Program Alarm』 appears during the compiling by **hp3** key, the cursor automatically moves to the error block by pressing any key.
- 4) During the execution of **hp3** compiling command, the system starts execution from the first block of the current program, assuming that the axes of the machine take the position of the current workpiece coordinates as the start point. Therefore, for some special programs, the stop position of each axis may affect the compiling; it is suggested that each axis stop in advance at the machining start point.

#### 4.3.6.2 Result Analysis of Program Compilation

On program edit page, there are two types of alarm due to program compiling error: [Program Alarm] and [Program Compound Check Alarm]. Only when the above types of alarm are not issued, can the compiling succeed.

Therein:

**Program alarm:** means the alarms caused by the command error in the programming. The alarm can be removed by inputting a correct command. It is not closely relative to the parameter setting.

**Program compound check alarm:** means the alarm which occurs during the program command check (relevance check according to the setting of auxiliary parameters and interface parameters). It is required to analyze the program in accordance with the setting of auxiliary parameters and interface parameters, and then modify the program and parameter setting to remove the alarm.

**【Example】**

**[Example for program alarm]:**

When pressing **hp3** key to compile a program, the alarm is displayed as follows:

```
[ Program alarm ]
Error E206: Missing message
Program: N0100 G92 Z300 P1
Position: P
```

**[Example for program compound check alarm]**

After the current program is compiled successfully by pressing hp3 key, and the control is switched from the Edit operation mode to the other modes, the alarm is displayed as follows:

```
[ Program comound check alarm ]
Error E610: Illegal use of M78
           This function is invalid.
```

#### 4.3.6.3 Prompts of Program Compound Check

After the program is compiled, if no error is detected, in general, the program can be executed in Auto operation mode. However, under the following conditions, the system displays the prompts of the program compound check.

- 1) The range of tool nose soft limit or machine soft limit is exceeded.

During the execution of the program, if the range of tool nose soft limit or machine soft limit is exceeded, the system displays the program compound check prompts when the control is switched from EDIT operation mode to AUTO operation mode.

- 2) Inconsistent with the tool setting record

If tool setting inconsistency occurs during the execution of the program, the system displays the program compound check prompts when the control is switched from the EDIT operation mode to the AUTO operation mode.

**【Example】**


**[Example for program compound check alarm]**

E.g. When the tool is T0408 during tool setting, and there is T0308 in the program, it is inconsistent with the tool setting record; When bit parameter **P403\_d4** is set to 1, an alarm message **T0308 inconsistent with tool setting record T0408** pops up when the control is switched from the EDIT operation mode to the AUTO operation mode.


## 4.4 JOG Operation Mode

In JOG (Manual) operation mode, the user can perform a certain operation pressing function keys directly, or perform a certain setting or operation pressing letter keys. For each step, the system offers corresponding intelligent prompt messages.

Descriptions for the relative setting or the input format and examples of relative operation are as follows: The function keys required to press are indicated by a sign; the letter keys or numeric keys are indicated by an underline; the prompt messages of the system are indicated by a rectangle.

If incorrect data is input during the input of letters or figures, press  key to delete it, and then input the correct data

When setting or inputting an item, or during the man-machine dialogue, if ESC key is pressed before confirmation, it means the current operation is quit

In Manual operation mode,  key is displayed on the upper right corner of the screen; After pressing this key, a window pops up, displaying the directory of the operation keys in Manual mode; by pressing this key again, the window is closed; by pressing other function keys directly, the window is automatically closed.

Press operation mode select key  to enter Manual operation mode.

For the CNC machine tool, a large amount of work, such as the installation and debugging of a variety of electrical components, debugging of motion performance, establishment of the coordinate system and tool preparation, is completed in Manual operation mode.

After the Manual page is entered, the system performs necessary analysis and pre-check in accordance with the user parameter list and tool offset values. If the precheck finds that the Manual operation may result in serious accidents, the system locks the function of Manual operation, and pops up a window displaying an alarm message; In this case, the user should modify relative parameters according to the alarm message, and then perform Manual operation.

This system provides multiple modes for the execution of part programs. Before the execution, the user can perform necessary setting in Manual operation mode, to ensure the safety during machining.

◆ The main functions of Manual operation mode include:

- ☆ The axes can be moved in JOG mode, STEP mode or MPG mode.
- ☆ The axes can be moved by absolute or incremental programming.
- ☆ Establishment of machine coordinate system, establishment of workpiece coordinate system
- ☆ Auxiliary functions such as spindle, chuck, cooling, or tool post
- ☆ Tool setting
- ☆ Real-time display of the states of the machine tool, real-time alarm displayed in a popup window.

Note: **928TD system has no Y axis function.**

The Manual page is displayed as fig. 4-4:

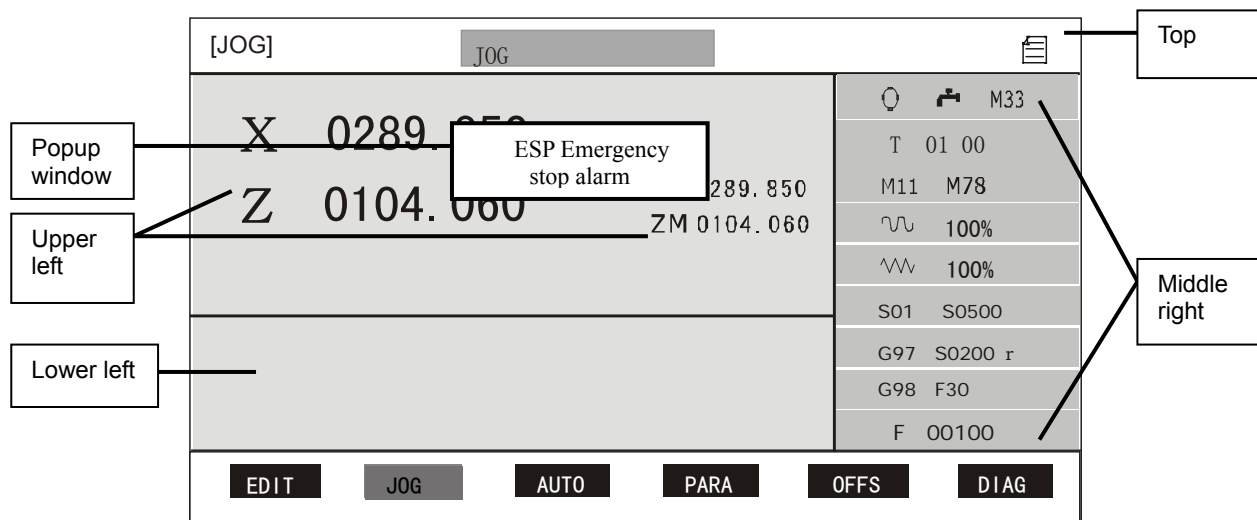



Fig. 4-4, Manual operation mode

◆ Explanation for each area of the interface:

Top: Displaying the manual feed operation mode, including JOG, STEP, X/Z MPG; prompt key  for system function operation methods;

Upper left: Displaying tool nose coordinates as well as machine coordinates;

Lower left: MDI input and alarm prompt area;

Middle right: Displaying the current states of the machine, including messages about the spindle, cooling, lubricating, chuck, tailstock, rotating speed and cutting speed;

Popup window: Displaying the system alarm messages.

◆ Explanations for the display of auxiliary function states:

- 1) The states of auxiliary functions are indicated by an icon or corresponding instruction code sign;
- 2) A black sign indicates the current state; e.g. spindle, cooling, etc;
- 3) A red sign indicates the function is being executed;
- 4) A blinking red sign indicates that the function was unsuccessfully executed or interrupted (e.g. reset, emergency stop) last time, and the system assumes that the corresponding function is in an uncertain state. When the tool or chuck is in an uncertain state, it is unavailable to start the machining program; Only by performing a successful operation or turning the power OFF then back ON, can the system recover to the normal state.
- 5) When there is constant response detection for the chuck/tailstock, the green signal indicates a normal state, while the yellow sign indicates an abnormal state;
- 6) The S behind the spindle gear indicates the spindle speed detected in real time;
- 7) Pressure detection icon  $\Delta$ : It is green when it is normal. It becomes half-empty yellow as soon as the low pressure is detected; when the time that the low pressure lasts exceeds half of the time set in P332, it turns to entirely-empty yellow;
- 8) G96/G97 and S indicate whether the mode is the constant surface speed cutting mode; S indicates the set surface speed or rotating speed, unit: m/min or r/min;
- 9) G98/G99 indicates whether the mode is feed per rotation or feed per minute; F is the set cutting speed;
- 10) F indicates the actual movement speed of the coordinate axis.



4.4.1 Coordinate Axis Movement



Before moving the coordinate axis, please determine the movement speed, movement distance first. In case of emergency, press down the Emergency Stop button immediately.

4.4.1.1 JOG movement



By pressing **STEP** key, the control can be switched from STEP or MPG mode to JOG mode.

【**JOG movement**】 means holding a coordinate axis movement key to cause the machine slide carriage to move continuously; When the key is released, the machine carriage decelerates to stop.

The meanings of axis movement keys are as follows:




X axis – direction movement key;




X axis + direction movement key



Z axis/Y axis – direction movement key (That the indicator of  key lights up is for Y axis.)



Z axis/Y axis + direction movement key (That the indicator of  key lights up is for Y axis.)

◆ Switching between Z and Y axes:

By pressing Z/Y axis switch key, the user can switch the operation between Z axis and Y axis; that the Z/Y axis switch indicator lights up is for Y axis operation. (when bit parameter **P405\_d1=1**, the machine tool has Y axis, then the switch is valid). Note: **928TD** system has no Y axis function.

When the machine tool is switched to Y axis operation, it is also the Y axis that is valid for operations such as program reference point return and machine zero point return.

【**Attention**】

- 1) When the motor is running at a high speed, if the coordinate axis movement key has been released, the machine carriage keeps moving instead of stopping immediately, because of the automatic acceleration/deceleration of the system. The movement distance varies with the max. motor speed, system acceleration/deceleration time and feedrate. The higher the speed is, and the longer the acceleration/deceleration time is, the longer the distance the slide carriage moves; and vice versa.

4.4.1.2 Step Movement



By pressing **STEP** key, the system can be switched from JOG mode or MPG mode to STEP mode.

【**STEP movement**】 indicates that the slide carriage moves a pre-set step width pressing a coordinate axis movement key once. The movement speed is executed at the selected rapid traverse

rate or feedrate. However, the max. movement speed of Z, X, Y axis is respectively limited by parameter P100, P101, P102.

By holding the key, the slide carriage feeds continuously by step width. The last step width is moved when the key is released. The step width of Step movement is displayed in the black background.

The step width of STEP movement is divided into 6 levels: **0.001** **0.01** **0.1** **1.0** **10.0** **50.0**. Press STEP REGULATION key repeatedly to select a level.

**【Note】**

- 1) The movement can be stopped by pressing CYCLE PAUSE key during movement. After the key is pressed, the slide carriage decelerates to stop, without retaining the remaining step width.

#### 4.4.1.3 MPG Control Movement



Pressing MPG key, the user can switch the control from JOG or STEP mode to MPG mode. That the indicator lights up indicates the MPG mode is selected.

**【MPG movement】** indicates that the system receives the pulse signals generated from the manual pulse generator (MGP or handwheel) to control the movement of the axis. The user can slightly control the movement of the coordinate axis by rotating the MPG.

◆ **Determining the movement amount per scale for the MPG before hand**

When the MPG dial rotates one scale, the coordinate axis moves one step width. There are three gears (0.001mm, 0.01mm, 0.1mm) selectable for the step width. The user can press STEP REGULATION key to switch among the three gears.

◆ **Determining the coordinate axis controlled by the MPG before hand:**

Pressing MPG key repeatedly, the user can select one coordinate axis (between X or Z/Y) controlled by the MPG. The coordinate of the selected axis is highlighted. When the indicator of Y/Z axis switch lights up, by repeatedly pressing MPG key, the MPG-controlled axis is switched between Y axis and X axis.

After selecting the coordinate axis to be moved, rotate the MPG to move the selected axis.

Rotating the MPG clockwise moves the coordinate axis in the positive direction. Rotating the MPG counterclockwise moves the coordinate axis in the negative direction.

**【Note】**


- 1) **The MPG rotation speed should be less than 5 r/s. If it is exceeded, the motor cannot stop immediately even if the MPG stops rotating, resulting in the MPG rotating scale differing from the movement amount of the slide carriage.**
- 2) **In MPG mode, neither JOG or STEP movement function, nor the program zero point return and machine zero point return are valid.**
- 3) **In MPG mode, when operating the functions relative to coordinate axis movement, e.g. when inputting word absolute/incremental movement, the functions relative to the MPG is forbidden temporarily. In this case, the MPG operation is invalid, with its LED indicator flicking. After the above-mentioned functions are executed, the MPG functions**

automatically resumes.

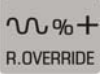

- 4) When a large override ( $\times 0.1$  mm gear) is selected, if the MPG is rotated at a high speed, the slide carriage moves at a high speed. Here, even if the MPG is not rotated, the slide carriage still keeps moving instead of stopping immediately. The moving distance varies with the max. motor speed, system acceleration/deceleration time, feedrate and MPG rotation speed. The higher the max. speed is, and the longer the acceleration/deceleration time is, and the faster the MPG is rotated, the longer the distance that the slide carriage moves before it stops by deceleration, and vice versa.
- 5) When parameter P400\_d4 is set to 0, step width 0.1 is valid; When parameter P400\_d4 is set to 1, step width 0.1 is invalid.
- 6) When p400\_d1 is set to 1, neither the control knob of the external MPG, nor the Y/Z axis select key and step regulation key are valid.

#### 4.4.1.4 Selecting Rapid Traverse Rate

##### ◆ Selecting manual rapid traverse or low-speed feed state

In Manual state, the speed of each axis in the negative/positive direction can be selected as either rapid traverse mode or cutting feed (low-speed traverse) mode. By pressing  key, the user can switch the state between manual rapid traverse and manual low-speed feed. That the speed indicator lights up indicates the rapid traverse state is selected.

##### ◆ Rapid traverse override

There are four gears for the rapid traverse override, which are 25%, 50%, 75%, 100%. Pressing  key once, the override is increased by one gear till 100%. Pressing  key once, the rapid traverse override decreases by one gear till 25%.

In the manual rapid traverse, the actual traverse speed is determined by the rapid traverse rate and the rapid traverse override.

**Actual traverse speed of Z (X, Y) axis = P100 (P101, P102) × rapid traverse override**

##### ◆ The manual movement operation affected by rapid traverse override and feedrate override is as follows:

JOG movement operation: When the speed indicator lights up, it is affected by rapid traverse override; when the indicator goes out, it is affected by feedrate override.

STEP movement operation: When the speed indicator lights up, it is affected by rapid traverse override; when the indicator goes out, it is affected by feedrate override.

Input word movement operation: When the speed indicator lights up, it is affected by rapid traverse override; when the indicator goes out, it is affected by feedrate override.


Program reference point return operation: It is affected by rapid traverse override.

Machine zero point return operation: It is affected by rapid traverse override.

**【Note】**

- 1) In manual Jog movement mode, first select the rapid traverse override, and then press the axis movement key.
- 2) In manual Step movement mode, the user can select the rapid traverse override first, or adjust the rapid traverse override during movement, with the traverse speed changed accordingly.

## 4.4.1.5 Selecting Speed for Low-speed Feed

When the speed indicator goes out by pressing  key, the low-speed feed state is selected.

**◆ System built-in feedrate**

When the F value of the input word is 0, the system uses the built-in feedrate.



The manual feedrate consists of 16 gears from **0~150%** (increment: 10%). The corresponding feedrate of each gear is as follows:

Feedrate override	Feedrate (mm/ min )	Feedrate override	Feedrate (mm/ min )
0%	0	80%	240
10%	7.5	90%	300
20%	22	100%	420
30%	38	110%	525
40%	60	120%	675
50%	82	130%	850
60%	110	140%	1000
70%	180	150%	1260

**【Note】**

- 1) There is an error between the speed in the table and the system actual speed. The system actual speed prevails.
- 2) When the feedrate override is 0, the system prompts “Feedrate override is 0”; which indicates the command is being executed, and the slide carriage is in the standstill state. To cause the slide carriage to move immediately, adjust the feedrate override to a value other than 0.

**◆ Feedrate override**

There are 16 gears (**0%~150%** with an increment of 10%) selectable for the feedrate override; By pressing  key once, the feedrate override increases by one gear till 150%; By pressing  key, the feedrate override decreases by one gear till 0%.

## 4.4.1.6 Inputting a Word to Move, Setting Feedrate

In Manual operation mode, it is available to move the axes according to the input length and direction, or to directly move the axes to the input coordinate position from the current position. The operation methods are as follows:

## ◆ Words relative to movement

The corresponding operation of each axis is:

X axis movement \_\_ X word (position of X axis), or U word (relative movement amount of X axis)

Y axis movement \_\_ Y word (position of Y axis), or V word (relative movement amount of Y axis)

Z axis movement \_\_ Z word (position of Z axis), or W word (relative movement amount of Z axis)

Range of word X, Z, Y, U, W, V: -99999.999mm~99999.999mm;

Feedrate \_\_ F word (F0000~F4000, the leading zero can be omitted, unit: mm/min)

## 【Format of movement word】

**Z(W)\_ F\_** ; Z axis moves alone. If F is omitted, the speed is determined by the state of system rapid traverse/feedrate (the same below).

**Y(V)\_ F\_** ; Y axis moves alone.

**Z(W)\_ X(U)\_ F\_** ; Z/X axes move simultaneously.

**X(U)\_ Y(V)\_ F\_** ; X/Y axes move simultaneously.

**Z(W)\_ Y(V)\_ F\_** ; Z/Y axes move simultaneously.

## 【Note】

- 1) In Manual operation mode, up to two axes can move simultaneously, at the selected manual traverse speed.
- 2) When Y axis is enabled by setting the system parameter, Y (V) word can be input.

## 【Example 1】

Input: W Move W -5.2 Enter Run? START (or ESC cancel); this means that Z axis moves 5.2 mm in the negative direction.

Input: X Move X 40 Enter Run? START (or ESC cancel); this means that X axis moves to the position of which the coordinate is 40.

## 【Example 2】

Input: **Z200 U50** ; Z axis moves to 200, X moves 50 in the positive direction. F speed is not specified, so it is determined by the state of rapid traverse/feed.

Input: **U20 W-50 F80** ; X axis moves 20 in the positive direction, Z axis moves 50 in the negative direction. Feedrate is 80, affected by the feedrate override.

Input: **F200** ; The feedrate is set to 200. The system is switched to Feed state.

Input: **F0** ; The feedrate is set to 0. The system is switched to Feed state and the built-in speed is used.

## ◆ Explanations for the movement speed:


- 1) If word F is not input, when the current speed indicator lights up, the speed is rapid traverse, which is affected by the rapid override; when the indicator goes out, the speed is low-speed movement, affected by the feedrate override.
- 2) If word F is input, the system automatically switches to the low-speed feed state, with the speed indicator off; Feedrate override gear is the current gear; if the input F value is greater than the one set in **P113**, the latter one prevails.

- 3) If the input value of word F is 0, the system uses the built-in feedrate.
- 4) Both JOG movement operation and STEP movement operation are controlled by word F even if the speed indicator goes out.
- 5) If word F is not input, when the indicator goes out, the speed of the low-speed feed is limited by parameter P113 (max. cutting feedrate); if the speed of the low-speed feed is greater than the one set in P113, the latter prevails.
- 6) If word F is not input, when the indicator goes out and the speed is double-axis low-speed feed, its movement mode is the interpolation movement mode, which is the same as that in executing G01 command in AUTO operation mode. The two axes rapid traverse proportionally and stop proportionally at the same time.
- 7) If word F is not input, when the indicator lights up, the speed of the single-axis rapid traverse is determined by parameters **P100~P102** (rapid traverse rate) and rapid traverse override.
- 8) If word F is not input, when the indicator lights up, the movement of the double-axis rapid traverse is determined by parameter **P400\_d3**, which is the same as that in executing G00 command in Auto operation mode. When bit parameter P400\_d3=0, the movement mode is independent movement mode that each axis rapidly and independently traverses. The resultant speed displayed on the screen is greater than the max. traverse speed of each axis. When bit parameter **P400\_d3=1**, the mode is the interpolation movement mode, i.e. the double axes rapid traverse proportionally and stop proportionally at the same time. The system takes both movement parameters and slope of the movement line segments into consideration, ensuring no step out occurs during movement. It is normal that the rapid traverse rate varies with the line segments with different slopes (priority is given to the axis which moves slower)
- 9) When the feedrate override is 0, and the system is in the low-speed state or there is word F in the block, no movement is allowed. After Enter key is pressed, the system prompts Feedrate override is 0 until the feedrate override is adjusted to another value.
- 10) Word F is input using G98 command. The system cannot input G99 command.

#### ◆ Calling a word for execution

During the movement by inputting a word, the system automatically saves the latest 8 times of command record that has been executed.



By pressing  key, the system displays a window showing the record list; the operator can call a record by pressing the numeric keys, and then modify or execute it directly.

#### 4.4.1.7 Drive Unit Enabling Control

Set bit parameter **P416\_d4** to 1, and press DELETE key twice successively in Manual or Auto mode, then the servo unit is turned off, and the motor is in the free state. By pressing DELETE key once when the drive unit is turned off, the unit is turned on, and the motor is in the working state.

#### 4.4.1.8 Alarm Prompts for Coordinate Axis Movement

When operating the axis movement, if the currently moving axis touches the soft limit of tool nose

coordinates (tool nose soft limit), the axis stops moving, and the screen displays a corresponding alarm. If the axis touches the machine coordinate soft limit (mechanical soft limit), the axis can only be moved in the opposite direction, the axis can only be moved in the opposite direction. However, Manual Zero Point Return is not restricted by the range of the soft limit.

#### 【Note】

In JOG feed or STEP feed mode, when the movement axis reaches the soft limit, the system issues the soft limit alarm. When a command is input to move Z/X axes in input word movement mode, if the specified value exceeds the range, the system prompts the soft limit alarm and does not perform execution. Whether the alarm for the machine or tool nose soft limit is valid is set by parameter P404\_d4 or P404\_d3 respectively.

### 4.4.2 Establishing a Coordinate System

#### 4.4.2.1 Establishing Machine Coordinate System—Machine Zero Return (Machine Reference Point Return)


##### ◆ Machine zero

The machine coordinate system is the reference coordinate system for CNC to calculate the coordinate position. It is the inherent coordinate system of the machine tool. After the system is installed, you should establish the machine coordinate system first.


The reference point of the machine coordinate system is referred to as the machine zero. (or machine reference point, or mechanical zero). For each machine, there is a fixed point taken as the machine reference point. Each time the machine returns to the machining start point after returning to machine zero, the error of the machine coordinate system caused by power-off and stepout can be removed. After unexpected power-off, you can retrieve the machine coordinate system and workpiece coordinate system by performing zero return, without setting the tool again.

In most cases, the system seeks the machine reference point using the deceleration switch and zero switch installed on the machine; you can install the deceleration switch only, then use the one-revolution signal of the servo motor to seek the machine reference point. The deceleration switches are generally installed on the max. stroke of Z/X axes in the positive direction.

##### ◆ Operations for machine zero return

Press  key to cause X axis to perform machine zero return;

Press  key to cause Z axis to perform machine zero return.

In Manual operation mode, by pressing  key, Z axis rapidly traverses to Z axis machine zero at the selected rapid traverse rate.

##### ◆ The process of performing zero return by the system is as follows:

**Zero return mode 1:** When there is a deceleration signal and a zero signal, the system performs execution by mode 1; the procedures of machine zero return are as follows:

- Step 1: The axes rapidly traverse at the rapid traverse rate in the specified direction, and decelerate to stop when the system detects the start point of the deceleration signal after the deceleration switch is pressed down.
- Step 2: The axes traverse at the specified zero return speed in the reverse direction, and decelerate to stop when the start point of the deceleration signal is detected.
- Step 3: If the specified zero offset is not 0, another zero offset will be moved.
- Step 4: The axes continue to move at the specified zero return speed, and then decelerate to stop when the zero signal is detected.
- Step 5: Then the zero return and detection is completed; finally, the system automatically modifies the machine coordinates of the current point to the “zero position coordinates” set by the parameter.

**Zero return mode 2:** When there is no zero signal but only a deceleration signal, the system performs execution by mode 2.

Compared with mode 1, mode 2 has no zero signals, so step 3 and step 4 are unnecessary; Only step 1, step 2 and step 5 are required to complete the zero return. However, the precision of the zero return of this mode is poorer than that of mode 1.

**Zero return mode 3:** When there is no deceleration signal, but only a zero signal, the system performs execution by mode 3.

Compared with mode 1, mode 3 has no deceleration signal, so step 1, step 2 and step 3 are unnecessary; Only step 4 and step 5 are required to complete the zero return. In this mode, the operator needs to move the axes to a specified position manually, and then performs the zero return. Otherwise, the result is incorrect.

**Zero return mode 4:** Where there is neither a deceleration signal nor a zero signal, the system performs execution by mode 4.


If there is no machine zero detection device (including zero switch and deceleration switch) installed on the machine, set the relative parameter to 0;

If the machine zero return function is performed at this moment, the system does not detect the zero signal and deceleration signal till the zero coordinate position of the axis is returned to.

**【Note】**

- 1) **All the axes return to the machine zero in the zero return direction. Therefore, if the “zero return direction” is set to the positive direction before the zero return, the axes should stay at the negative direction of the machine zero.**
- 2) **During the machine zero return, the rapid traverse rate of the coordinate axis is restricted by the rapid traverse override.**
- 3) **During the machine zero return, the movement of the axis is not restricted by the soft limit parameters.**
- 4) **For the parameters relative to the machine zero return, see Parameter Setting Operation Mode in Part II Programming.**



- 5) For the connection and zero return modes of the machine zero return, see **Machine Zero Return and Connection in Part III Connection**.
- 6) After the execution of the machine zero return, behind the corresponding machine coordinates on the screen is displayed the blue icon  (indicating the machine zero return) as a prompt.

#### 4.4.2.2 Establishing Machine Coordinate System— without Machine Zero (No Machine Reference Point)

(Note: This operation is invalid for those axes equipped with zero detection devices)

For those axes without zero detection devices (without a deceleration signal and zero signal), the operations below can be used to create a machine coordinate system.

##### 【Format】

Input: **INPUT** **U** **New coordinate** **ENTER**. To modify the machine coordinate of X axis to the new coordinate.

Input: **INPUT** **W** **New coordinate** **ENTER**. To modify the machine coordinate of Z axis to the new coordinate.

#### 4.4.2.3 Setting Workpiece Coordinate System

This system employs the floating workpiece coordinate system. The floating workpiece coordinate system is the reference for the tool setting and relative dimension. After determining the machine coordinate system, you should set the workpiece coordinate system.

##### 【Format】

Input: **INPUT** **X** **New coordinate** **ENTER**. To modify the current tool nose coordinate of X axis to the new coordinate.

Input: **INPUT** **Z** **New coordinate** **ENTER**. To modify the current tool nose coordinate of Z axis to the new coordinate.

##### ◆ The procedures of setting workpiece coordinate system are as follows:

Fix the trial-cut workpiece on the machine, then select any tool (in general, the first tool used in the machining)

1. Set the workpiece coordinate in the X direction:
  - 1) Select an appropriate spindle speed to start the spindle.
  - 2) Move the tool to cut a small step on the trial-cut workpiece, and keep the X axis position unchanged.
  - 3) Move the tool to a safe position in the Z direction and then stop the spindle. Measure the diameter of the cut step.
  - 4) Press **INPUT** key, the screen displays **Setting**, then press **X** key, the screen displays **Set**

**workpiece coordinate system X**; input the measured diameter (input radius if in radius programming) , press **ENTER** key, then the system automatically sets the workpiece coordinate of X axis.

## 2. Setting the workpiece coordinate in Z axis direction

1) Start the spindle and move the tool to cut an end face on the workpiece. Keep the position of Z axis unchanged.

2) Move the tool to a safe position in the X direction, then stop the spindle.

Select a point as the reference point (it is suggested that this point be a fixed point on the machine, such as the chuck end face or other reference faces, so that the newly established workpiece coordinate system coincides with the original one after the latter one is broken) . Measure the distance from the cut end face to the selected reference point in the Z axis direction.

3) Press **ENTER** key, the screen displays **Setting**, then press **Z** key, the screen displays **Set** **workpiece coordinate system Z**; input the measured data, then press **ENTER** key, then the system automatically sets the workpiece coordinate of Z axis.

**Note: After the above operations, the establishment of the system workpiece coordinate system is completed.**

### 【Explanations】

- 1) The operations of setting the workpiece coordinate system are to modify the tool nose coordinates of the current point under the condition that the machine coordinates and tool compensation are not changed. The result of the operation is equal to that after resetting the offset amount between workpiece coordinate system and machine coordinate system.
- 2) In general, the operations of setting the workpiece coordinate system are performed once after the system initialization, or after the replacement of the workpiece type (when all the tool compensation values are cleared). It is unnecessary to set the workpiece coordinate system afterwards.

### 【Note】

**When the actual position of the tool is inconsistent with the one of the workpiece coordinate system due to the step out caused by some special causes, it is inappropriate to use the method of resetting the workpiece coordinate system, because both workpiece coordinate system and machine coordinate system are changed after step-out. In this case, if only the workpiece coordinate system is corrected, without correcting the machine coordinate system, an unexpected “machine coordinate soft limit alarm” may occur.**

**After the motor is out of step, the appropriate operations are as follows:**

- 1) Choose a reference point for tool setting (a position easy for the tool nose to reach and easy to observe), and measure Z and X coordinates of this point;
- 2) Move the tool nose to a reference point (the coordinates of the reference point are known)
- 3) If the coordinates of the tool nose are inconsistent with those of the reference point, press **DELETE** key twice successively to turn off the drive unit;

- 4) Input word movement to make the tool nose coordinates consistent with the reference point coordinates (the coordinates are changed without changing actual tool nose);
- 5) Press **DELETE** key again to turn on the drive unit.

After the above operations, both the machine coordinate system and the workpiece coordinate system are corrected.

#### 4.4.2.4 Setting Program Reference Point

In the machine coordinate system, the operator should determine a position. When the tool nose stays at this position, it is safe and convenient to change the tool and clamp the workpiece. When the tool post stays at this position to set the program reference point, this position is referred to as the program reference point (or program zero point). The coordinates of the program reference point are relative to the machine coordinate system.

##### 【Format】

By pressing **INPUT** key, the screen displays **Setting**; then by pressing **0** key, the screen displays **Set program reference point?**; here, press **ENTER** key to confirm this point as the program reference point of Z/X/Y axes.


If the workpiece coordinate system is reset after setting the program reference point, the coordinates of the original reference point is unchanged in the new workpiece coordinate system. In this case, reset the program reference point. The initial values of the program reference point are X=150 Z=200.


After setting the program reference point, both the operations of program reference point return command (G26) and program zero point return using keys on the system operation panel return to this point, regardless of where the slide carriage is.

#### 4.4.2.5 Program Reference Point Return

**Before performing program reference point return, the operator must determine the position of the program reference point. Otherwise, unexpected accidents may occur.**


In Manual operation mode, perform this operation by directly pressing function keys. After pressing the keys, the corresponding axes rapidly return to the program reference point. When the axis change indicator lights up, the operation is for Y axis.

By pressing  key, X axis rapidly returns to the program reference point from the current point.

By pressing  key, Z axis rapidly returns to the program reference point from the current point.

##### 【Note】

- 1) When waiting for machining, each axis should stay at the program reference point generally.

- 2) After performing program reference point return, the green icon  for the program reference point return is displayed as a prompt before the corresponding machine coordinates on the screen.

#### 4.4.2.6 Recovering Workpiece Coordinate System and Program Reference Point

In Manual operation mode, the workpiece coordinate system and program reference point have been set. If the blocks containing G50 command are executed in Auto operation mode, the workpiece coordinate system and program reference point are changed. With the following operations, the operator can recover the workpiece coordinate system and program reference point set in the Manual operation mode.

##### 【Format】


Key in: **G 5 1 ENTER** , to recover the workpiece coordinate system and program reference point set in the Manual operation mode


#### 4.4.3 Spindle Control Function


##### 4.4.3.1 Spindle Start/Stop Control

###### ◆ Spindle Start/Stop Operation

In Manual operation mode, it is available to control spindle CW rotation, CCW rotation and spindle stop by directly pressing the function keys on the panel or by keying in M03/M04/M05 command (Feed/Spindle hold knob is invalid in Manual operation mode).





Press  key, or key in **M 3 ENTER** ; the spindle rotates in CW direction. The screen displays the state of the spindle, and the LED indicator on the key lights up.

Press  key, or key in **M 5 ENTER** ; the spindle stops.

Press  key, or key in **M 4 ENTER** ; the spindle rotates in CCW direction. The screen displays the state of the spindle, and the LED indicator on the key lights up.

###### ◆ Spindle JOG control

It is available to switch to the spindle JOG control using the spindle stop key.

When the spindle stops, by pressing  key, the system switches to the spindle JOG control state, highlighting the spindle state icon on the screen. By pressing the  key once again, the system switches to the normal state. In the spindle JOG control state, by pressing  or  key, the spindle rotates for a specified period of time at the specified speed and then stops. (If the



specified time is too long, you can also press **S.STOP** key to stop it). When the spindle is in the JOG state, the spindle control commands M03, M04, M05 input by MDI are invalid. The spindle JOG speed is set by parameter P309, and the JOG rotation time by parameter P308. After the JOG time is up, the spindle stops automatically, with the LED indicator on the key OFF.

◆ Interlock relationship between spindle start/stop and chuck:

When **P402\_d5=0**, the interlock relationship between hydraulic chuck control and spindle control is as follows:

- 1) When the chuck is not clamped, it is forbidden to start the spindle; otherwise, "Chuck is not clamped, spindle start forbidden" is issued.
- 2) When the spindle is rotating, it is forbidden to operate the chuck; otherwise, "Spindle is not stopped, chuck operation forbidden" is issued.

◆ Interlock relationship between spindle start/stop and tailstock:

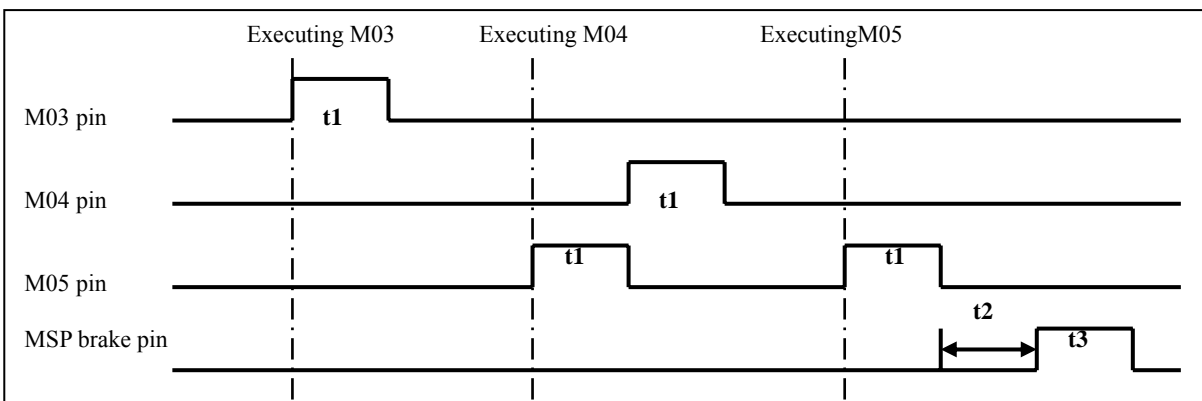
When **P402\_d3=0**, the interlock relationship between tailstock control and spindle control is as follows:

When the spindle is rotating, it is forbidden to operate the tailstock; otherwise, "Spindle is not stopped, tailstock operation forbidden" is issued.

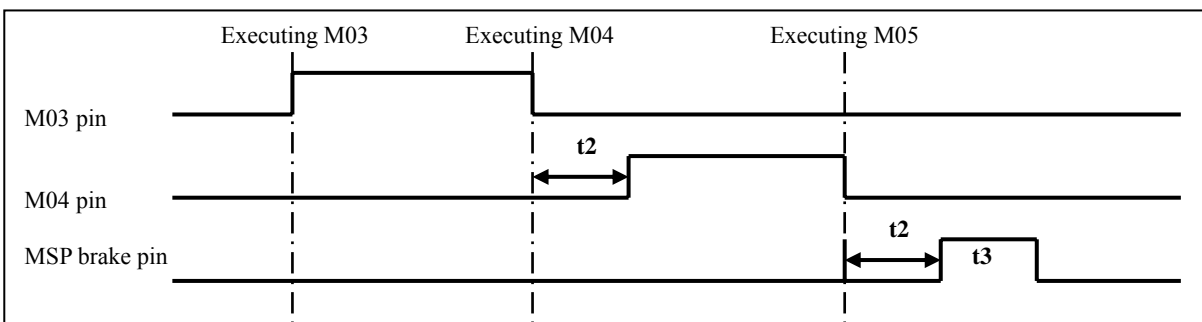
◆ Spindle start/stop execution and signal output time sequence:

**Note:** Select the spindle control output signal by parameter P410\_d7. When P410\_d7 is set to 0, it is the spindle control level output. When P410\_d7 is set to 1, it is spindle control pulse output. The time sequence relationship between spindle brake signal MSP and spindle start/stop signal is as follows:

- 1) In pulse control mode, the output time sequence of M3, M4, M5, MSP:



- 2) In level control mode, output time sequence of M3, M4, M5, MSP (here, M5 pin does not output, which can be used for other purpose):



- t1: Hold time output by signal M3, M4, M5 in the pulse control mode, set by parameter P326;
- t2: Spindle stop brake delay time, set by parameter P315;
- t3: Hold time output by spindle brake signal MSP, set by parameter P316.

#### 4.4.3.2 Spindle S Command – Gear Shift Control

(Note: Those users who employ a variable frequency spindle are not required to read this section.)

If the machine is not equipped with a variable frequency spindle, set parameter **P410\_d6** to 0, in order to use S function to shift the spindle gear. The standard format of word S consists of an S and a two-digit number, which indicates the spindle gear number.

**【Format of S code】**

**Sx** ;  
**Sxx** ;

**【Example】**

Select the second gear speed of the spindle:

Key in: **S 0 2 ENTER** ; the system outputs S02 signal, and the screen displays gear state **S02**.

**【Explanations】**

- 1) When parameter **P410\_d5** is set to 0, the gear control signal is directly output by the bit, and the range of S code is S00~S04. Each gear signal corresponds to one output point. S0 means all the outputs are invalid.
- 2) When parameter **P410\_d5** is set to 1, the gear control signal is the coded output, and the range of S code is S00~S15. The coded output is as follows:

Code Output point	Code															
	S00	S01	S02	S03	S04	S05	S06	S07	S08	S09	S10	S11	S12	S13	S14	S15
<b>S01</b>		★		★		★		★		★		★		★		★
<b>S02</b>			★	★			★	★			★	★			★	★
<b>S03</b>					★	★	★	★					★	★	★	★
<b>S04</b>									★	★	★	★	★	★	★	★

Note: “★” in the table indicates that the output of the corresponding output point is valid.

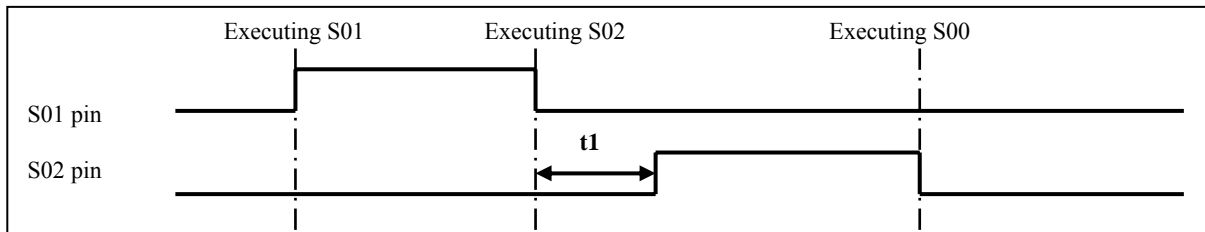
- 3) The line number controlled by the **actual output** of the spindle gear is set by parameter P310.
  - P310=4, the actual output control points are S01, S02, S03, S04;
  - P310=3, the actual output control points are S01, S02, S03; S04 is released, which can be used for other purpose;
  - P310=2, the actual output control points are S01, S02; S04, S03 are released, which can be used for other purpose;
  - P310=1, the actual output control point is S01; S04, S03, S02 are released, which can be used for other purpose;

P310=0, S code does not output; S04, S03, S02, S01 are released, which can be used for other purpose;

- 4) During the coded output of the gear control signal (parameter **P410\_d5** is set to 1), if the control lines specified by parameter P310 are less than 4, only the low-gear control lines are valid, the high-gear ones are not output. Thus the high-gear coded control lines are released, which are not affected by the gear control.

◆ Gear shift execution of spindle S commands and signal output time sequence:


At system power on, the default is S00, and S01~S04 are invalid. When executing any one of S01, S02, S03, S04, the corresponding S signal output is valid and retained, while the other three S signal outputs are cancelled. When executing S00 command, the outputs of S01~S04 are cancelled. Only one of S01~S04 is valid at a time.





t1: Interval time for spindle gear shift (P313).

◆ Function key operation for spindle gear shift

In addition to the above method of executing spindle gear shift, the operator can also change the

spindle gear by pressing spindle gear shift key .

By pressing  key once, the spindle speed is output in the order of S01~S04 or S00~S15

(coded output) recurrently. If there is only the four-gear spindle speed, you need to press  key three times to switch the gear to S01 from S02.

### 4.4.3.3 Spindle S Command— Rotating Speed Control

(Note: Those users who do not employ a variable frequency spindle are not required to read this section.)

If the machine is equipped with a variable frequency spindle, set parameter **P410\_d6** to 1. To solve the problem that the frequency converter has insufficient torque at a low speed, the system is provided with four levels of automatic gear shift output signal. With the converter working in a high frequency, the machine obtains a higher cutting torque at a lower speed. The system uses M41/M42/M43/M44 command for the spindle gear control; and uses S code for the rotation speed control.

◆ Gear Control of the Variable Frequency Spindle

【Format】

M41 ;  
 M42 ;  
 M43 ;  
 M44 ;

## 【Explanations】

- 1) M41, M42, M43, M44 output the gear control signals. Each gear signal corresponds to one output point of S01, S02, S03, S04 respectively.
- 2) The control lines of the actual output of the spindle gear is set by parameter P310.  
 P310=4, the actual output control points are S01, S02, S03, S04;  
 P310=3, the actual output control points are S01, S02, S03; S04 is released, which can be used for other purpose;  
 P310=2, the actual output control points are S01, S02; S04, S03 are released, which can be used for other purpose;  
 P310=1, the actual output control point is S01; S04, S03, S02 are released, which can be used for other purpose;  
 P310=0, the actual output control points do not output; S04, S03, S02, S1 are released, which can be used for other purpose;
- 3) The initial gear at system power on is M41.

## 【Execution process and signal output time sequence of spindle M gear shifting:】

At CNC power on, whether the spindle gear is memorized is controlled by parameter **P400\_d6** (spindle gear memory):

- 1) When the parameter is set to 0, the spindle gear is not memorized at power on after power off. The default is the first spindle gear, and M41~M44 have no output.
- 2) When the parameter is set to 1, the spindle gear is memorized at power on after power off. If the specified gear is consistent with the current one, no gear shift is performed. If the specified gear is inconsistent with the current one, the process of gear shift is as follows:
  - ① Execute any one of M41, M42, M43 and M44 commands, and output the analog voltage to the spindle servo or frequency converter according to the value (unit: mv) set by data parameter **P314** (the voltage output during spindle gear shift);
  - ② After delaying the time set in data parameter **P311** (gear shift time 1 of the variable frequency spindle), turn off the output signal of the original gear;
  - ③ After delaying P313 (gear switch interval of the variable frequency spindle), output the new gear shift signal;
  - ④ When the system is connected to the detection of gear shift in-position signals M41I, M42I, M43I, M44I; If the gear shift is not in-position, the CNC does not proceed to the next step until the gear shift in-position signal is detected; when the system is not connected to the detection of gear shift in-position signals, it directly proceeds to the next step; M41I~M44I input signals are defined in the interface parameters.



- ⑤ After delaying data parameter **P312** (gear shift time 2 of the variable frequency spindle), output the spindle analog voltage by the value set in data parameters P300~P303 (corresponding to gears 1~4 respectively) according to the current gear, to complete the gear shift.

◆ Rotating speed control of the variable frequency spindle

If the machine is equipped with a variable frequency spindle, control the spindle speed using S commands. The standard format of S command consists of an S and a 4-digit number. The two-digit number indicates the spindle gear number. There two ways to input spindle speed:

- 1) S sets the fixed speed of the spindle (r/min); the spindle speed keeps unchanged if S value is not changed, which is referred to as constant rotation speed control.
- 2) S sets the surface speed (m/min) of the tool nose relative to the outer circle of the workpiece, which is referred to as constant surface speed control. In constant surface speed control mode, the spindle speed varies with the coordinate of X axis during cutting feed. See “Constant Speed Control G96, Constant Speed Control Cancel G97 ” in Part II Programming.

【Code format】

- G96           ;   Setting the constant surface speed cutting state;
- G96 S\_\_       ;   Setting the constant surface speed cutting state and specifying the value of the surface speed; range: 0~9999 m /min;
- G97           ;   Cancelling the constant surface speed state; G97 is a modal command;
- G97 S\_\_       ;   Cancelling the constant surface speed state and specifying the value of the rotating speed; range: 0~9999 r /min;
- S\_\_           ;   The type of speed depends on the current state; it is either rotating speed value or constant surface value.

【Example】

Key in: S 2 0 0 ENTER; the system converts the rotating speed to the analog voltage of 0~10 V and outputs it to the frequency converter via the output interface.

【Explanations】

- 1) When executing the S command, the system calculates the analog voltage corresponding to the rotating speed taking the max. spindle speed of the current spindle gear as a reference, and then outputs it to the spindle servo or frequency converter.
- 2) In order to keep the spindle actual speed and the speed set by the S command the same, the actual max. spindle speed for each gear should be set by parameters **P300~P303** (when the output analog voltage is 10 V); Setting method: First input S\_\_ according to the setting values of parameters **P300~P303**, then modify the setting of parameters **P300~P303** according to the spindle speed displayed by the system.
- 3) The analog voltage is 0 V at system power on. After executing S command, the system outputs a corresponding voltage; then the voltage keeps unchanged (unless it is in the cutting feed state of constant surface speed control and the coordinate of X axis is changed). After executing S0,

the analog voltage output is 0V. The analog voltage output remains unchanged at CNC reset and emergency stop.

#### 4.4.4 Coolant Control

In Manual operation mode, the coolant ON/OFF operations can be controlled by directly pressing the function keys on the panel or inputting M08/M09 command.

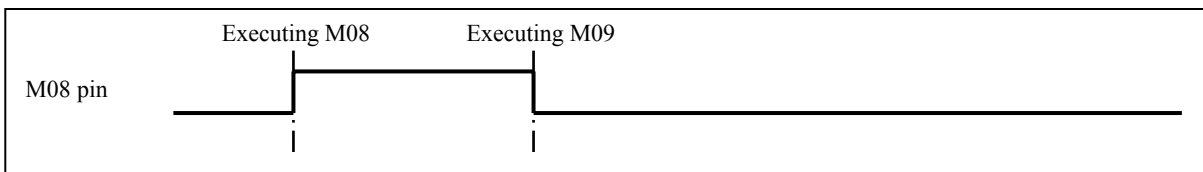


By pressing **COOLING** key, the coolant operation is switched between ON and OFF; The state icon on the screen, and the LED indicator on the key indicate corresponding states. When the coolant is ON, the LED indicator lights up; when the coolant is OFF, the indicator goes out.

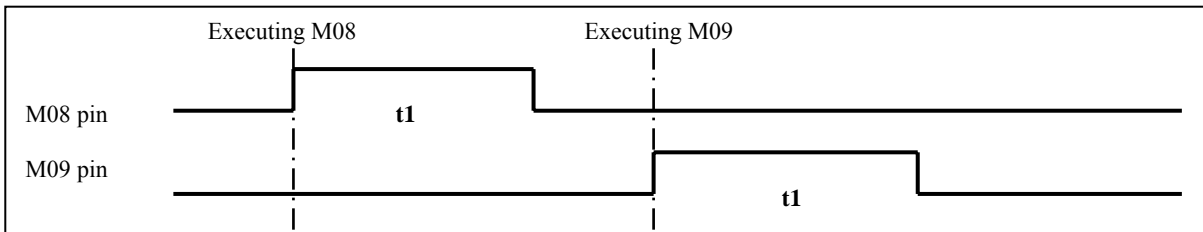
Key in: **M 8 ENTER**; the coolant is ON.

Key in: **M 9 ENTER** ; the coolant is OFF.

- 1) Output time sequence of M8, M9 in level control mode: (M9 pin does not output, which can be used for other purpose)



- 2) Output time sequence of M8, M9 in pulse control mode:



t1: Hold time of M08, M09 signal output in pulse control mode, set by parameter P326.

#### 【Relative parameters】

P410\_d7: When it is set to 1, the coolant is controlled by the system pulse output; when it is set to 0, the coolant is controlled by the system level output. This parameter is shared by the spindle control output parameter.

#### 4.4.5 Manual Tool Change Control

In Manual operation mode, the tool change control can be performed by directly pressing the tool change function key on the panel or inputting T function commands.

##### ◆ Operation of tool change function key



By pressing **T.CHANGE** key once, the tool post rotates to the next tool number, which will be displayed on the screen (if parameter **P403\_d5** is set to “confirmation” required, press **ENTER** key afterwards.)

◆ Input format of T function command

The standard format of the tool function word in this system includes a T and a 4-digit number. The first two digits indicates the tool number, and the last two digits indicates the tool offset number. It is not required that all the four digits be input. The operator can use 2~4 digits depending on the conditions.

【Format】

- Txx     \_\_\_ The first digit means the tool number, and the second number means the tool offset number;
- Txxx    \_\_\_ The first digit means the tool number, and the second two digits means the tool offset number;
- Txxxx   \_\_\_ The first two digits means the tool number, the last two digits means the tool offset number.

【Explanations】

The range of the tool number is determined by parameter **P319** (Max. tool number: 1~16); if **P319** is 4, the tool number can be 0~4. If the tool number input is 0, the current tool number is maintained.);

Range of tool offset number: 0~64; if the input tool offset number is 0, the tool offset is cancelled.

【Example】

- Key in: T 4 6     Enter ; Changing to tool number 4, and executing tool offset number 6
- Key in: T 3 0     Enter ; Changing to tool number 3, and cancelling tool offset
- Key in: T 0 6     Enter ; Maintaining the current tool number, and executing tool offset number 6
- Key in: T 8 1 2   Enter ; Changing to tool number 8, and executing tool offset number 12
- Key in: T 4 0 5   Enter ; Changing to tool number 4, and executing tool offset number 5
- Key in: T 0 6 0 8 Enter ; Changing to tool number 6, and executing tool offset number 8
- Key in: T 0 0     Enter ; No tool changing, and canceling tool offset
- Key in: T 0 4 0   Enter ; Maintaining the current tool number, and executing tool offset number 40

【Note】

- 1) Example: Inputting T400 means changing to tool number 4 and cancelling the tool offset. (Note: T400 cannot be input as T040 here.)
- 2) If the electric tool post malfunctions, the system displays the alarm message “Tool number signal detection overtime” on the screen, which indicates that the corresponding tool number cannot be found within a specified period of time.
- 3) The CNC system employs the absolute tool change mode, so the position of each tool number is fixed on the tool post when an electric tool post is used. Make sure the tool number where the tool post is located is the same as that displayed on the screen during installation and debugging.

- 4) When parameter P318 is set to 0, the selected tool post is the line-up tool post, which does not output the tool change signal during tool change.
- 5) The tool offset can be executed by moving the slide carriage or modifying the coordinate system. It is set by parameter P403\_d6.  
When P403\_d6 is set to 1, the slide carriage is moved without changing the coordinates during tool offset execution.  
When P403\_d6 is set to 0, the coordinates displayed are modified without moving the slide carriage during tool offset execution.
- 6) If the tool change fails or it is interrupted (by reset, emergency stop), the system assumes that the tool is in an uncertain position, and keeps the tool number flicking read as a prompt. In this case, the machining program cannot be started; Only by performing a successful tool change or powering on the system after power off, can the system recover to the normal state.
- 7) During tool change, if the target tool number is the current one, only the tool offset is changed without performing tool change output action, with the exception of the following two cases:
  - After tool change fails, the tool number is flicking red on the screen, which means the tool number displayed is not necessarily the same as the actual tool number. If the target tool number in the tool change command to be executed next time is the current tool number, the system executes one tool change action.
  - During the first tool change after system power on, if the target tool number is the current tool number displayed on the screen, the system executes one tool change action.

◆ Execution process and signal output sequence time of T function command:

The system is provided with multiple tool change modes. For details, please refer to Section 3.4 “Tool Change Control Function and Connection” in Part III Connection in this manual.

#### 4.4.6 Manual Tool Change

In general, it is necessary to use different tools to machine a workpiece. Since the tool installation and tool dimension are different, the position where the tool nose of a tool is located does not coincide with others when the tool is rotated to the cutting position. In such a case, there exists an offset.

Tool setting is used for the system to automatically store the offset to the specified tool offset number.

After tool setting, the user can edit the part program according to the workpiece drawing and machining technique by specifying the corresponding tool offset number in the tool change command of the program, regardless of the difference between the tools.

The tool offset list of the system is capable of storing 64 groups of tool offset from number 1~64, with each offset number corresponding to one group. For each group, five items, i.e. Z offset, X offset, tool radius, imaginary tool nose number, tool setting record are recorded (see Tool Offset Operation Mode).

Through tool setting, the system modifies Z offset, X offset and tool setting record in the tool offset number specified by the user. However, the other two items, the tool radius and imaginary tool number,

should be input in advance, because the system needs to reference to these two items during tool setting. Otherwise, unexpected results may occur (if the tool is not ball-shape, both the two items should be set to 0 in advance. For details, see Tool Offset Operation Mode).

Tool setting record item means the system automatically records the tool number when the number is set to the tool offset number. For safety reasons, the system will scan the machining program using tool setting record. E.g. the T command is T0409 during tool setting, but the machining program includes T0309, so it is inconsistent with the tool setting record. It is highly dangerous in this case, and the system prompts an alarm message.

#### 【Precautions before tool setting】

- 1) According to the above description, determine the offset number where the system stores the offset value in advance , and determine the contents of the imaginary tool nose item of the offset number in advance.
- 2) For the same tool nose, only Z offset and X offset can be stored to the same offset number. Otherwise, unexpected accidents may occur.
- 3) In general, it is recommended that tool number 1 use offset number 1, and tool number 2 use tool offset number 2 for convenience.
- 4) It is suggested that the operator execute the offset number first, and then perform tool setting, making the operation visible. E.g. if the offset value in tool number 4 needs to be stored to offset number 9, first execute T49 command.
- 5) The tool setting must be performed on the condition that the workpiece coordinate system is normal. Otherwise, unexpected accidents may occur.

The system is provided with trial-cut tool setting and fixed-point tool setting, which can be chosen by the user. The explanations for them are as follows:

#### ◆ Trial-cut tool setting (method 1):

##### 【Format】

Key in: **I** Measured value **ENTER** [Tool offset number] **Enter**. To modify the current tool nose coordinate of X axis to the new coordinate.

Key in: **K** Measured value **ENTER** [Tool offset number] **Enter**. To modify the current tool nose coordinate of Z axis to the new coordinate.

##### 【The procedures for trial-cut tool setting are as follows:】

After fixing the workpiece on the machine, the user can perform tool setting to any tool with the above method till all the tools are set. This operation is convenient and quick when a tool is abraded or needs to be adjusted.

#### (1) Tool setting in X direction:

- 1) Fix the trial-cut workpiece on the machine, and select any tool (select the first tool used in the machining generally).
- 2) Select an appropriate spindle speed, and start the spindle. Move the tool to cut a step along the workpiece surface in Manual operation mode.

- 3) Without moving X axis, move the tool to a safe position in Z direction and stop the spindle. Measure the diameter of the cut step.
- 4) By pressing **I** key, the screen displays **Tool setting X**; Input the diameter measured, then press **ENTER** key.
- 5) The system prompts **Confirm tool offset No.: XX**; and automatically presets an offset number. If this offset number is the same as the one to be input, press **ENTER** key directly. Otherwise, press **ENTER** key after inputting the offset number. The system then automatically calculates the tool offset value of X axis and stores it to the specified offset number.



(2) Tool setting in Z direction:

- 1) Start the spindle again, move the tool to cut an end face along the workpiece surface.
- 2) Without moving Z axis, move the tool to a safe position in X direction and stop the spindle.
- 3) Select a point as the reference point, measure the distance from the cut end face to the selected reference point in Z direction.
- 4) By pressing **K** key, the screen displays **Tool setting Z**; input the measured data, then press **ENTER** key.
- 5) The system prompts **Confirm tool offset No.: XX**; and automatically presets an offset number. If this offset number is the same as the one to be input, press **ENTER** key directly. Otherwise, press **ENTER** key after inputting the offset number. The system then automatically calculates the tool offset value of Z axis and stores it to the specified offset number.



◆ **Trial-cut tool setting (method 2):**

【The procedures are as follows:】

(1) Tool setting in X direction:

- 1) Fix the trial-cut workpiece on the machine, and then select any tool (select the first tool used in the machining generally).
- 2) Select an appropriate spindle speed, and start the spindle. Move the tool to cut a step along the workpiece surface in Manual operation mode.
- 3) Without moving X axis, by pressing  key, the system automatically memorizes the tool nose position, with tool setting icon  flickering on the screen; then move the tool out of X axis to a safe position and stop the spindle. Measure the diameter of the step measured.
- 4) By pressing **I** key, the screen displays **Tool setting X**; input the measured diameter, then press **ENTER** key.
- 5) The system prompts **Confirm tool offset No.: XX**; and automatically presets an offset number. If this offset number is the same as the one to be input, press **ENTER** key directly. Otherwise, press **ENTER** key after inputting the offset number. The system then automatically calculates the tool offset value of X axis and stores it to the specified offset number; the system automatically cancels the tool setting icon as well.

## (2) Tool setting in Z direction:

- 1) Start the spindle again, move the tool to cut an end face along the workpiece surface.
- 2) Without moving Z axis, by pressing  key, the system automatically memorizes the tool nose position, with tool setting icon  flickering on the screen; then move the tool out of Z axis to a safe position and stop the spindle.
- 3) Select a point as the reference point, measure the distance from the cut end face to the selected reference point in Z direction.
- 4) By pressing **K** key, the screen displays **Tool setting Z**; input the measured data, then press **ENTER** key. The system then prompts **Confirm tool offset No.: XX**; and automatically presets an offset number. If this offset number is the same as the one to be input, press **ENTER** key directly. Otherwise, press **ENTER** key after inputting the offset number. The system then automatically calculates the tool offset value of Z axis and stores it to the specified offset number; the system automatically cancels the tool setting icon as well.

## 【Explanations】

- 1) When the tool setting icon is flickering, it is available to perform operations such as spindle start/stop, coordinate axis movement; if tool setting is performed, the system automatically cancels the tool setting icon, and does not memorize the original tool setting point.
- 2) If **K** or **I** is directly pressed without the tool setting icon appearing, the system assumes the current point as the tool setting point.

◆ **Fixed-point tool setting:**

After determining a reference point on the machine (or on the workpiece), first set a group of coordinates (Z, X) for the point by means of actual measurement or using other methods. Then move all the tool noses to this reference point to obtain this group of coordinates, with the system memorizing the tool offsets automatically. This type of tool setting is called the fix-point tool setting. I.e. first execute “Input reference point coordinates” or “Modify reference point coordinates” to determine the reference point coordinates; then move the tool nose to the reference point and execute “Fixed-point tool setting” to complete the tool setting.

【Fix-point tool setting by pressing **hp1** key】

## 1) Inputting the reference point coordinates:

It means inputting the current tool nose coordinates as the coordinates of the reference point; in general, this method is applied when the coordinates of the reference point is unknown, but the tool nose coordinates of the current tool is considered as correct. Move the tool nose to the reference point, and then execute “Input reference point coordinates” to complete the input operation. After that, the system automatically stores the current tool nose coordinates as the coordinates of the reference point. If a tool is broken or needs to be re-mounted, select any tool which has been set as the reference.

## 2) Modifying the reference point coordinates:

Input the coordinates of Z/X reference point, then press **ENTER** key to complete the modification of the reference point coordinates; in general, this method is applied when the coordinates of the reference point is known.

## 3) Fixed-point tool setting:

Move the tool to be set to the reference point, and execute “Fixed-point tool setting”, then press **ENTER** key to complete the tool setting of the current tool. The system will automatically generate the tool offset.

**【Note】**

- 1) If using an optical tool setting gauge, the user can set the tool setting point at the intersection of the reticle of the gauge without starting the spindle; the other operations are the same.
- 2) The tool offset automatically generated by the system can be viewed and modified in the tool offset operation mode. See Tool Offset Operation Mode.
- 3) When using the line-up tool, if the tool is located at the other side of the workpiece, the measured value in the X direction input during trial-cut tool setting should be negative.

## 4.4.7 Hydraulic Chuck Control Function

## ◆ Chuck operation

In Manual operation mode, key in M10/M11 command to control the chuck clamping/releasing

Key in: **M 1 0 ENTER**; the chuck is clamped. The screen displays the spindle state.

Key in: **M 1 1 ENTER**; the chuck is released.

Key in: **M 1 2 ENTER**; the chuck control signal is cancelled. (M12 is used for some special chuck devices)

**【Relative parameters】**

When parameter **P409\_d7** is set to 0, the system has hydraulic chuck control function.

When **P402\_d5**=0, there is an interlock relationship between hydraulic chuck control and spindle control.

When **P402\_d4**=0, the chuck response signal consecutive detection is turned off.

When **P402\_d4**=1, the chuck response signal consecutive detection is turned on.

When **P409\_d6**=0, the hydraulic chuck is the outer chuck mode.

When **P409\_d6**=1, the hydraulic chuck is in the inner chuck mode.

When **P409\_d5**=1, the hydraulic chuck requires the response detection; the response signal is green when it is normal, and is yellow when it is abnormal.

When **P409\_d5**=0, the hydraulic chuck does not require the response detection.

When **P409\_d3**=0, the hydraulic chuck control signal is level control;

When **P409\_d3**=1, the hydraulic chuck control signal is pulse control; the pulse width is set by parameter P327.

When **P409\_d1**=0, the footswitch input of the hydraulic chuck is valid;

When **P409\_d1**=1, the footswitch of the hydraulic chuck is invalid

## ◆ Execution of chuck commands

In the outer chuck mode, after executing M10 command, the system outputs the chuck clamping signal from M10 pin (Whether to output a pulse signal or level signal is selected by a parameter), then



the chuck clamping action is completed when the response detection signal is not needed; If the response detection is needed, the system waits until the chuck clamping is in-position. If the chuck clamping signal (interface pin RM10 is low level, and RM11 is high level) is detected within the specified period of time (set by parameter **P329**: M command response detection time specification), the chuck clamping action is completed; otherwise, the alarm “**Chuck clamping response detection overtime**” is issued.

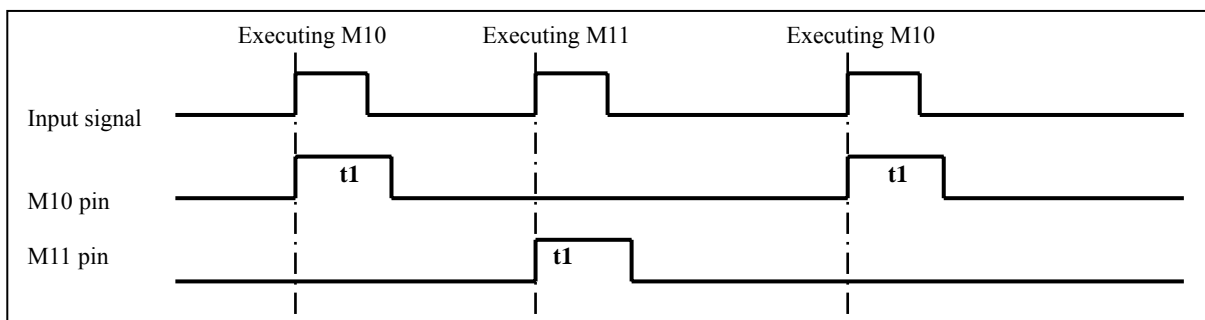
After executing M11 command, the system outputs the chuck releasing signal from pin M11 (whether to output a pulse signal or level signal is selected by a parameter). If the response detection signal is not required, the chuck releasing action is completed. If the response detection signal is required, the system waits until the chuck clamping is in-position. After the chuck releasing in-position signal (interface pin RM11 is low level, and RM10 is high level) is detected, the chuck releasing action is completed. Otherwise, the alarm “**Chuck releasing response detection overtime**” is issued.

In the inner chuck mode, after executing M10 command, the system outputs the chuck clamping signal from pin M11; after executing M11 command, the system outputs the chuck releasing signal from pin M10. Therefore, only the output pins are opposite to those of the outer chuck mode, and the others are the same.

In addition to controlling the hydraulic chuck with commands, the user can also control it with an external footswitch. By pedaling the footswitch once, the system switches between clamping and releasing once according to the control mode of the clamping/releasing command M10/M11. Before switching to Manual or Auto operation mode from other modes, it is required that the chuck footswitch be released (disconnected). Otherwise, the system issues an alarm of abnormality.

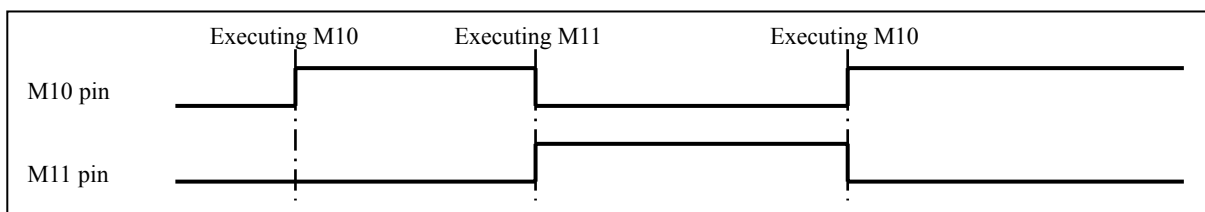
◆ Output time sequence of hydraulic chuck control signal:

- 1) Output time sequence of M10, M11 in pulse control mode:



t1: M10, M11 signal output hold time in pulse control mode, set by parameter P327;

- 2) Output time sequence of M10, M11 in level control mode:



**【Note】**

- 1) When the hydraulic chuck function is valid, the system is in the chuck releasing state by default after power on or emergency stop; When the first hydraulic chuck footswitch input is valid, the system outputs the chuck clamping signal.
- 2) When there is an interlock relationship between chuck and spindle: it is forbidden to operate the chuck during spindle running, or an alarm occurs; it is forbidden to start the spindle in the chuck releasing state, or an alarm occurs.
- 3) The operation of the footswitch is invalid during automatic continuous execution, regardless of whether the spindle is running or not.
- 4) If the operation of the chuck fails or is interrupted (by reset, emergency stop), the system assumes that the chuck is at an uncertain position, and displays a red flicker, prompting the chuck state (M10 or M11). In this case, it is unavailable to start the machining program; only by performing a successful chuck operation or powering on the system after power off, can the system recover to the normal state.
- 5) Chuck response signal consecutive detection (bit parameter P409\_d5=1) is to continue detecting whether there is abnormal releasing occurring in the normal state or machining state.
- 6) When the chuck signal is cancelled (in M12 state), there is an underscore being displayed under the chuck state (M10 or M11), i.e. M10 or M11 are displayed.

#### 4.4.8 Hydraulic Tailstock Control Function

##### ◆ Tailstock operation

In Manual operation mode, key in M78/M79 command to control tailstock advancing/retracting

Key in: M 7 8 Enter; The tailstock advances.

Key in: M 7 9 Enter; The tailstock retracts.

Key in: M 8 0 Enter; The tailstock control signal is cancelled. (M80 is used for some special tailstock devices)

##### 【Relative parameters】

When parameter **P409\_d4** is set to 0, the system has the hydraulic tailstock control function.

When **P402\_d3**=0, there is an interlock relationship between hydraulic tailstock control and spindle control.

When **P402\_d2**=0, the hydraulic tailstock response signal consecutive detection is off;

When **P402\_d2**=1, the hydraulic tailstock response signal consecutive detection is on.

When **P409\_d2**=0, the hydraulic tailstock control signal is level control.

When **P409\_d2**=1, the hydraulic tailstock control signal is pulse control; the pulse width is determined by the time set in parameter P328.

When **P409\_d0**=0, the footswitch input of the hydraulic tailstock is valid;

When **P409\_d0**=1, the footswitch input of the hydraulic tailstock is invalid;

◆ Execution process and signal output time sequence of tailstock commands:

If the detection of the tailstock in-position signal is required, define pin **RM78** or **RM79** in interface parameter **P519** or **P520**.

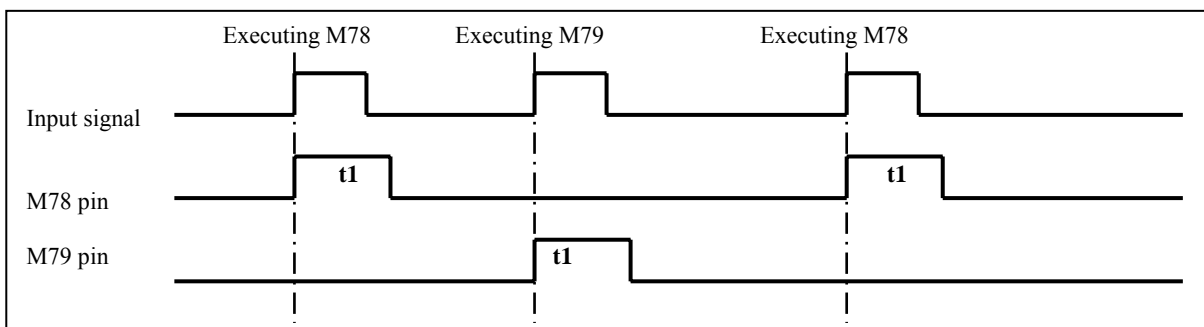
After executing M78 command, the system outputs the tailstock advancing signal from pin M78 (whether to output a pulse signal or level signal is set by the parameter); then the tailstock advancing action is completed; if the response detection signal is required, the system waits for the tailstock advancing in-position signal. After the tailstock advancing in-position signal (interface pin RM78 is low level, and RM79 is high level) is detected within a specified period of time (set by parameter P329: M command response detection time specification), the operation is completed; otherwise, the alarm “**Tailstock advancing in-position response detection overtime**” is issued.

After executing M79 command, the system outputs the tailstock retracting signal from pin M79 (whether to output a pulse signal or level signal is set by the parameter), then the tailstock retracting action is completed; if the response detection signal is required, the system waits for the tailstock retracting in-position signal. After the tailstock retracting in-position signal (interface pin RM79 is low level, and RM78 is high level) is detected within a specified period of time, the operation is completed; otherwise, the alarm “**Tailstock retracting in-position detection overtime**” is issued.

In addition to controlling the hydraulic tailstock with commands, the user can also control the tailstock with an external footswitch. By pedaling the footswitch once, the system switches between tailstock advancing and retracting once according to the control mode of the tailstock advancing/retracting command M78/M79.

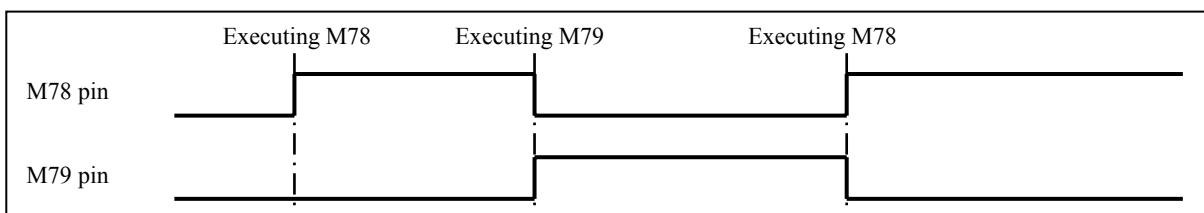
Before switching to Manual or Auto operation mode from other modes, it is required that the tailstock footswitch be released (disconnected); otherwise, the system issues an abnormality alarm.

1) Output time sequence of M78, M79 in pulse control mode:



t1: Signal M78, M79 output hold time in pulse control mode, set by parameter P328;

2) Output time sequence of M78, M79 in level control mode:



**【Note】**

- 1) When the hydraulic tailstock function is valid, the system is in the tailstock retracting state by default after power on or emergency stop; When the first hydraulic tailstock footswitch input is valid, the system outputs the tailstock advancing signal.
- 2) When there is an interlock relationship between tailstock and spindle: it is forbidden to operate the tailstock during spindle running, or an alarm occurs.
- 3) The tailstock control input is invalid during automatic continuous execution, regardless of whether the spindle is running or not.
- 4) If the operation of the tailstock fails or is interrupted (by reset, emergency stop), the system assumes that the tailstock is at an uncertain position, and displays a red flicker, prompting the tailstock state (M10 or M11). In this case, it is unavailable to start the machining program; only by performing a successful tailstock operation or powering on the system after power off, can the system recover to the normal state.
- 5) Tailstock response signal consecutive detection (set by parameters P519, P520) indicates the system continues detecting whether the tailstock is released abnormally in the normal state or machining state. If the function is set to alarm (bit parameter P402\_d2=1), the program machining is stopped and the spindle is turned off when the tailstock is released during machining.
- 6) When the tailstock signal is cancelled (in M80 state), there is an underscore displayed under the tailstock state (M78 or M79), i.e. M78 or M79.

## 4.4.9 Other Option Functions

The option function is the non-standard pin input/output control function.

If a function is required, it is necessary to define its pin in the interface parameters and correctly connect the lines. For details on the definitions of interface parameters, refer to Section 4.6 Parameter Operation Mode; For details on the connection, refer to Chapter Three “CNC Device Connection” in Part III Connection.

**【Warning】**

**The definition of pins must be accomplished by the machine tool builder because incorrect definition may cause damage to the system and the machine tool electric.**

## 4.4.9.1 Triple-color Indicator Control

If the function is required, define its output pin in the interface parameters and correctly connect the lines; the system will output a signal at the corresponding pin.

**P502: LMP3:** Green light (program execution signal light 3);

**P503: LMP2:** Yellow light (program execution signal light 2);

**P504: LMP1:** Red light, alarm light (alarm light control signal 1) .

**【Function description】**

- 1) The triple-color indicator is valid in Manual/Auto operation mode; in other operation modes, it is invalid.
- 2) During program execution, if the green light lights up, it indicates the normal operation.

- 3) If an alarm occurs, the green light goes out, and the red light lights up.
- 4) When the execution of the program is stopped, and no alarm occurs, the red and green lights go out, while the yellow light lights up.

#### 4.4.9.2 Lubricant Control

If the function is required, define its output pin in the interface parameters, and connect the lines correctly; the system will output the signal at the corresponding pin.

**P506: M32O:** Lubricant control output signal

##### 【Function description】

##### 1) Non-auto lubrication

When **P330** is set to 0: the lubrication is non-automatic. The lubricant ON/OFF is controlled with commands.

In Manual/Auto operation mode, Lubricant ON/OFF command M32/M33 is valid if input.

After executing M32, the lubricant output is ON; after executing M33, the lubricant output is OFF.

##### 2) Automatic lubrication:

When **P330** is set to 0: Automatic timing lubrication.

Lubrication start time and lubrication interval are settable. After the system is powered on, the system starts the lubrication and lasts a period of time set by P330, then stops output; after delaying the time set by P331, the system outputs lubricant again, and so on.

##### 【Note】

If the system enables the automatic lubrication function, the values of parameter **P330** and **P331** should be greater than 1s. If the values are smaller than 1s, they are processed as 1s.

#### 4.4.9.3 Machine Electricity Delay Power-on Control

If the function is required, define its output pin in the interface parameters, and connect the lines correctly; the system outputs the signal at the corresponding pin.

**P505: MDLY:** Machine electricity delay power-on control signal.

##### 【Function description】

If the machine electricity delay power-on control signal is defined in interface parameters, the system outputs the signal from the self-defined pin after delaying 3s, and then retains. During the 3-second delay, it is unavailable to press keys to operate the system.

#### 4.4.9.4 Safety Door Detection Function

If the function is required, define its output pin in the interface parameters, and connect the lines correctly; the system outputs the signal at the corresponding pin.

**P511: SAGT:** Safety door detection signal.

**【Function description】**

- 1) When SAGT signal is connected to 0V, the system confirms that the safety door is closed.
- 2) In Auto operation mode, if the system detects the safety door is open, the alarm “Safety door not closed” occurs.
- 3) During automatic operation, if the system detects the safety door is open, all the axes feed is stopped, and then the spindle is stopped, the coolant is turned off, with an alarm being issued.
- 4) The safety door detection function is valid only in Auto operation mode.

## 4.4.9.5 Low-pressure Detection Function

**【Relative parameters】**

When **P412\_d5** is set to 1, the low-pressure detection function is valid.

**P412\_d4** sets the low pressure alarm level; when P412\_d4=1, the low-level alarm is set, when P412\_d4=0, the high-level alarm is set.

**P332** sets duration for low pressure alarm.

**【Function description】**

- 1) After choosing the low pressure alarm detection function, the pressure detection icon  $\triangle$  is displayed on the right of the status bar in Manual and Auto operation modes. When the pressure is normal, the icon is displayed as a green solid triangle  $\blacktriangle$ . Once the system detects the low-pressure alarm signal PRES is valid, the screen displays a half-empty yellow triangle if the low pressure hold time is less than half of the time set in P332. If the low pressure hold time exceeds half of the time set in P332, the screen displays a yellow full-empty triangle  $\triangle$ . If the signal hold time exceeds the value set in P332, the icon is displayed as a red empty triangle  $\triangle$ , with the alarm “Low pressure alarm” being issued ; in this case, all axes feed is suspended, the spindle is stopped, and the automatic cycle cannot start.

## 4.4.10 Viewing Operation Information in Manual Mode

See Section 4.5.8 “Auto Operation Mode” in Part I Operation For details.

## 4.4.11 Appendix Table

## 4.4.11.1 List of M function Commands Controlled by MDI Input


- ◆ The M commands which can be input and executed in Manual mode are as follows:

Type	Command	Function	Remarks
Spindle control	M03, M04, M05	Spindle CW rotation, CCW rotation, and stop	Function interlock, state retained
Coolant	M08, M09	Coolant ON/OFF	
Chuck	M10, M11, M12	Clamping, releasing, cancelling chuck output signal	
Lubricant	M32, M33	Lubricant ON/OFF	' Function interlock, state retained
Tailstock	M78, M79, M80	Tailstock advancing, tailstock retracting, tailstock cancelling output signal	
User output 1	M21, M22		Function interlock, state retained
User output 2	M23, M24		Function interlock, state retained
Spindle gear	M41, M42, M43, M44	Switching spindle gear from gear 1, 2, 3, 4	Function interlock, state retained
User-defined command	M60 ~ M74		
Output control and detection	M82	E.g. M82 Q17.0 D3 or M82 Q17.0	


**Note:** When inputting an M command, if its first digit is 0, the 0 can be omitted. The function of this M command is the same as that in Auto operation mode. See the explanations for commands in Part II Programming.

## 4.5 Auto Operation Mode

In Auto operation mode, the descriptions for the input format of relative setting and operation as well as the examples are as follows: The function keys to be pressed are indicated with an icon; the letter keys, numeric keys to be pressed are indicated with an underscore; the system prompt messages are indicated with a rectangle.

In Auto operation mode, on the right corner of the screen is displayed the  page key; by pressing this key, the directory window displaying the operation keys in Auto mode pops up; by pressing the key again, the pop-up window is closed; by directly pressing other function keys, the window is automatically closed.



Press operation mode select key  to enter Auto operation mode. In Auto operation mode, the system completes the workpiece machining specified by the machining program; the system executes the program from the first line in accordance with the logic path until the end of the program.


After entering the Auto page, the system performs necessary analysis and pre-check to the part program in terms of the user parameter list and tool offsets. If the system detects the execution of the program results in serious consequences after precheck, it will lock the CYCLE START key, and prompt a window displaying an alarm message; in this case, it is unavailable to press CYCLE START key to execute the program; the user should modify the program or relative parameters correctly in accordance with the alarm message, and then perform the execution.

This system provides multiple modes for executing the workpiece program. Before the execution, the user can perform necessary setting to ensure the safety during machining.

◆ The main functions of Auto operation mode include:

- 1) Setting **Single/Continuous** program execution
- 2) Setting **Dry Run (without output)** to check the execution, the program execution can be accelerated in Dry Run state.
- 3) Prechecking the soft limit alarm before program execution
- 4) **Setting the program block**, starting the execution from any block between first block and last block
- 5) Pressing keys to control the spindle and coolant
- 6) Performing **Pause, Block Stop, End Stop, Cycle Stop** during program execution
- 7) Adjusting **Cutting Speed Override** proportion
- 8) Modifying tool offsets during program execution
- 9) Displaying the states of the machine in real time, popping up the real-time alarm

◆ The explanations for the areas on the screen are shown in fig. 4-5:

Top: Displaying the execution mode (Single/Continuous, Dry Run), current program number, part count, machining time, and key  (prompting operation methods of system functions)



Upper left: Displaying the tool nose coordinates and machine coordinates, or tool nose path;

Lower left: Displaying machining blocks (the pointer points at the current block)

Right middle: Displaying the current states of the machine, including the spindle, coolant, lubricant, tool post, chuck, tailstock, rotation speed and cutting speed;

Pop-up window: Displaying the alarm message of the program being executed.

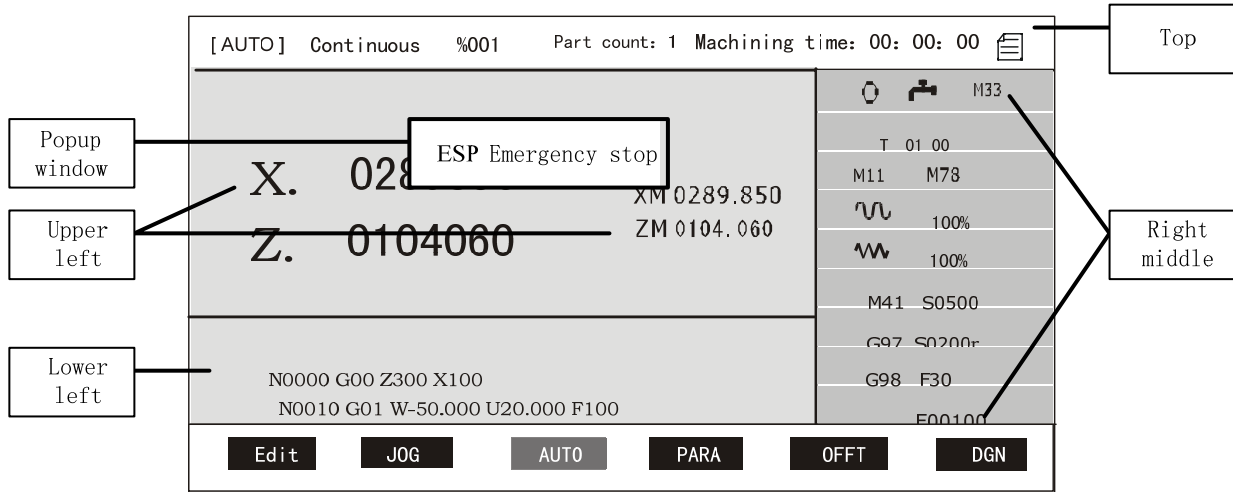


Fig. 4-5 Auto operation mode

#### 4.5.1 System Working States in Auto Operation Mode

In Auto operation mode, the system may be in the following states from the perspective of program machining; the functions which can be operated by the user vary with the system state; in this manual, several simple concepts are introduced to describe these states.

Initial state: The execution has not been started, and the pointer points at the first block of the program; When switched to Auto operation mode from other modes, the system is in the initial state; after completing the execution, or after an alarm, the system returns to the initial state.

Running state: The system is executing blocks, and the axes are moving.

Pause state: The execution is suspended when the axis move commands are being executed; the system waits for user's operation by pressing keys.


Block stop state: The execution stops after the current block has been executed and before the next block is executed; the system waits for user's operation by pressing keys.

#### 4.5.2 Explanations for Function Key Operation in Auto Operation Mode

##### 4.5.2.1 Switching between Single and Continuous Operation

Single/Continuous operation switch:



By pressing  key, the system switches between Single/Continuous operation recurrently; (valid in any state)

In the Continuous execution, the user can also press this key to switch to Single mode; then after the current block is executed, the execution is stopped; by pressing **CYCLE START** key, the user can continue the execution.

In Continuous mode, by pressing **CYCLE START** key once, the program is executed from the beginning to the end once.

In Single mode, by pressing **CYCLE START** key once, one block is executed (for the cycle command, only one operation is executed; press **CYCLE START** key again to perform another operation)

#### 4.5.2.2 Switching between Dry Run and Machining Run

Check all the contents of the machining program using Dry Run, preventing the workpiece from being damaged due to a programming data error in the program.

**Dry Run/Machining Run mode switch:**



By pressing **DRY** key, the system switches between Dry Run/Machining Run recurrently.

If the system is set to Dry Run when executing commands, whether M, S, T are valid is set by parameters. After the system exits Dry Run state, the coordinate of each axis automatically recovers to the one before Dry Run.

Parameters which are valid in Dry Run mode:

Parameter **P401\_d7**:

- 0: When executing auxiliary functions, the system outputs and detects signals as usual.
- 1: When executing auxiliary functions, the system does not output and detect signals.

Parameter **P401\_d6**:

- 0: The execution speed of the feed command is set by the program as usual.
- 1: The execution speed of the feed command is out of the control of the program. The system demonstrates the program path at the max. speed of cutting feed (P113).

#### **【Note】**

- 1) The Dry Run key is valid only when the program executes the **initial state**. This key is invalid during the program execution, or before the end of the execution, or before the execution state is exited.
- 2) When P401\_d7=0, all the M, S, T auxiliary function commands are executed in Dry Run state; after the dry run state is exited, the system does not recover to the previous state.
- 3) When P401\_d7=1, the system does not output and detect signals when executing auxiliary functions; however, when executing T functions, the tool offset number is executed (e.g. if the original one is T11, it becomes T13 after executing T33); after exiting Dry Run state, the original state recovers.
- 4) In Dry Run state, all M60~M74 commands are executed normally; if the tool offset is modified, the offset will be changed. If the tool offset is modified, the tool nose coordinates of the corresponding offset number will be changed after exiting Dry Run state.

5) In Dry Run state, the part counter does not increase by 1 automatically.

#### 4.5.2.3 Switching between Coordinate Display and Graph Display

This function is valid in all operation modes

The system enters AUTO operation mode after the first power-on, and it automatically selects the coordinate display as the display mode.

In Auto operation mode, press I key to switch between Coordinate Display and Graph Display anytime.

In Graph Display mode, by pressing S key, the graph display path can be cleared.

#### 4.5.2.4 Starting Execution from First Block of Program

After entering Auto operation mode, the system is in the initial state, and the pointer always points at the first block of the current program. By pressing CYCLE START key, the user can start the program automatic execution.





During the execution, the block being executed is displayed in poor color and blinks; the first line is the block that has been executed, and the third line is the one to be executed; if the program being executed is a conditional command, and the jump or call object is not known, the third line may not be displayed.

#### 4.5.2.5 Starting Execution from a Specified Block

In some special cases, if the user needs to start the execution from a block in the middle of the program, first select the start block using this function.(Valid in the initial state)

Procedures of selecting a block:

1) By press INPUT key, the system displays the program browser window displaying the current program, and the pointer points at the first line.

2) By pressing  ,  ,  , or  key, the system displays the last (next) block or last (next) page of programs. By pressing ESC key, the system exits the selection and displays the original program.

3) When the pointer points to the block to be selected, press ENTER key, then the window prompts Run ?, and waits for the next operation.

4) At this time, if CYCLE START key is pressed, the system starts execution from the block which the pointer points to; if ESC key is pressed, the system exits the selection, and the pointer points at the first block.

#### 【Note】

1) **The specified block cannot be in the canned cycle, compound cycle or subprogram; otherwise, unexpected accidents may occur. Generally, select G00 command, or the tool change command before G00 to start execution.**

- 2) It is suggested that the selected block from which the execution is started be the linear movement or M, S, T commands. When selecting G02/G03/G05 command for execution, make sure the coordinates of the tool and machine stay at the start point of the arc; otherwise, the arc machined may not meet the requirements.
- 3) During the execution, by pressing **ENTER** key, the system also displays the browser window, but it is forbidden to select a block.

### 4.5.3 Display during Program Execution

During the program execution, on the screen are displayed the running states, dynamic running coordinates, as well as the real-time tool nose running path, so as to monitor the running states of the machine and program. The contents displayable are as follows:

- ◆ Dynamic coordinates or dynamic tool nose movement path during the program execution.
- ◆ Contents of the block being executed.
- ◆ Auxiliary function states, e.g. spindle, coolant, lubricant, tool, rotation speed, chuck, tailstock.
- ◆ Feedrate override, rapid traverse override.
- ◆ Machining time.
- ◆ Part count

#### 4.5.3.1 Definition of Graph Display Data

Due to the limitation of the display area, for the workpieces of different sizes, different kinds of scaling are required to display their complete shapes. Therefore, the system defines four types of data, which are the workblank length, tool initial offset, display scaling, graph area type (customized or system-created). When the system is in the initial state, the user can define the above data pressing



key. The display is as fig. 4-6:

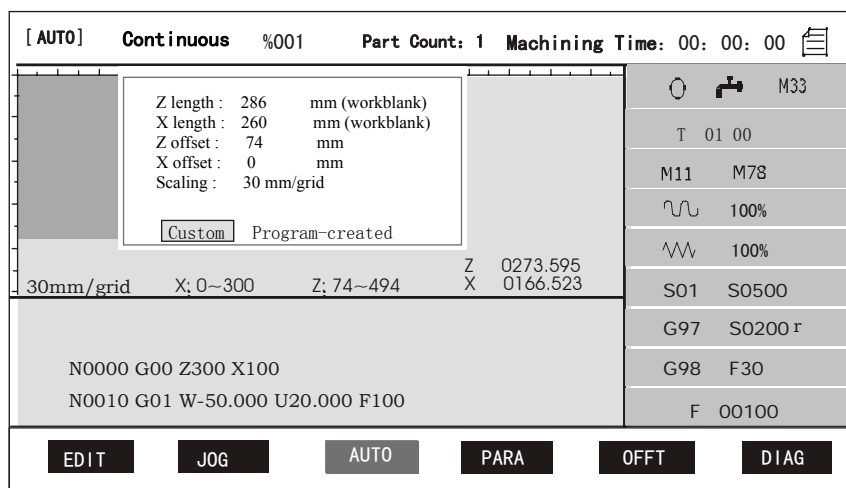


Fig. 4-6 Definition of graph display data

In the graph display, the horizontal scale mark indicates the coordinate dimension in Z direction, the vertical scale mark indicates the coordinate dimension in X direction; determine the area that is to display the graph first. The custom graph display area is as follows:

Z: (Z offset ~ Z offset + display scaling × 14 );  
 X: (X offset ~ X offset + display scaling × 5 × 2);  
 e.g. X: 300 ← 600      Z: -200 ← 220

Hereinto:







- 1) The coordinates of the intersection point between Z scale mark and X scale mark are referred to as Z offset and X offset (unit: mm).
- 2) Both Z scale mark and X scale mark are divided into a fixed number of equal grids.
- 3) The length of each grid is referred to as the display scaling, which is the scaling of the display graph. It has nothing to do with the actual machining scaling.
- 4) If the size of the workpiece is too large, select scaling down; if the size is too small, select scaling up, in order to view the graph clearly; the actual value range of X/Z offset: -9999 ~ 9999

After entering system graph display, the total lengths of Z scale mark and X scale mark of the part program workblank are Z length and X length respectively. (Unit: mm)

Z: Scaling × Number of grids that the tool nose path graph occupies on Z axis;  
 X: Scaling × Number of grids that the tool nose path graph occupies on X axis × 2

The graph area is classified into the custom type and program-created type. When selecting the custom type, the length, offset, scaling can be modified; when selecting the program-created type, the length, offset, and scaling cannot be modified. Here, the area range values pre-set by the system is the complete graph of the program movement path, so it is unnecessary to modify the length, offset and scaling. The system enters the program to set the execution state.

#### 4.5.3.2 Inputting Graph Display Data

The graph display data is shown as follows by pressing  key in Auto operation mode. First press  key or  key to select the custom type "Custom", and then press  key or  key to select the data to be modified. Then the screen displays the previously defined data as follows. Move the cursor to the data to be modified, then press  key to delete the original data and input the desired data.

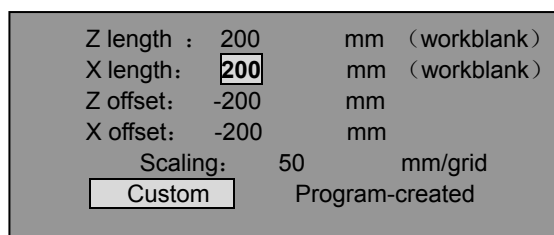








Fig. 4-7 Graph data display

- Inputting data (without a decimal): Press  key to delete the original data, then input the new data. When pressing  key repeatedly, the graph data display interface is displayed recurrently.
- Modifying the scaling: When the cursor points at Scaling, press  key or  key to scale down or scale up. The built-in scaling in the system contains 16 levels: 5, 10, 20, 30, 40, 50, 60, 70, 80, 90, 100, 200, 300, 400, 500, 600. The user can select an appropriate level to obtain the optimal display effect depending on the actual conditions.
- Two types of graph area: Press  key or  key to select custom type or program-created type. When selecting the program-created type, the data cannot be modified; when selecting the custom type, the data can be modified. At this moment, the user can input the new data as required.

After completing the modification of the data, press ESC key or ENTER key to exit, and return the system to Auto operation mode. If it is in the graph display mode, the system refreshes the graph display range according to the set display data. When the set display data exceeds the displayable range, the system prompts X/Z overlimit. At this moment, re-adjust the display data or reduce the display scaling.


**【Note】**


- 1) To correctly display the tool path, the start position of the tool should be beyond the display range of the workblank, or the machining progress cannot be correctly displayed.
- 2) During the program execution, if the coordinate display is switched to the graph display, the system may not correctly display the workpiece shape until the next cycle starts.

#### 4.5.3.3 Part Count and Timing

**Part count:** After the program executes the program end command M02, M20 or M30 once, the part count increases by one. The max. count is 99999. If it is exceeded, it is automatically cleared. When M99 command is used to end a program, the part count also increases by 1 after the program is executed once.

**Machining time:** Total program execution time. When CYCLE START key is pressed to start program execution, the system starts timing until the program ends. During the execution, if a pause operation is performed, the timing is stopped simultaneously. After the execution recovers, the timing is started again. In Single operation, the system only records the execution time of each block. The max. execution time: 99 hours 59 minutes 59 seconds; if the time reaches the maximum value, it is automatically cleared.

**Part count and machining time clearing:** In the initial state, by pressing  key twice,

the work count is automatically cleared; by pressing  key twice, the machining time is cleared.

#### 4.5.4 Manual Operation for Machine Auxiliary Functions

In Auto operation mode, by pressing function keys, the user can operate some of the auxiliary functions, which are the same as those in MANUAL operation mode. However, the auxiliary functions allowed to operate by pressing keys vary with the system state; The rules are as follows:

- 1) In any state, the coolant ON/OFF switch key is valid.
- 2) When bit parameter **P400\_d5** is set to 0, the spindle CW, CCW, Stop keys are valid. (invalid in the running state).
- 3) In the initial state, the spindle gear shift key (speed switch key) is valid.
- 4) In the initial state, when the hydraulic chuck control function is valid, the user can control the clamping and releasing of the chuck with an external button or footswitch. The chuck clamping/releasing action interlocks with the spindle.
- 5) In the initial state, when the hydraulic tailstock control function is valid, the user can control the clamping and releasing of the tailstock with an external button or footswitch. The tailstock clamping/releasing action interlocks with the spindle ..

#### 4.5.5 Speed Override Adjustment in Auto Operation Mode

##### 4.5.5.1 Speed Override Adjustment

In Auto operation mode, the feedrate override key and rapid override key are valid in any state.

In Auto operation mode, the program execution speed can be changed by changing the speed override, without altering the program and the speed set in parameters.

- **Feedrate override** adjusting the value set in speed word F in the program:

$$\text{Actual feedrate} = F \times \text{feedrate override}$$

There are 16 gears from 0%~150% (increment: 10%) for the feedrate override, which controls all the commands controlled by the feedrate during the program execution. When the feedrate override is 0, the program execution is stopped.

- **Rapid override** adjusting the speed of rapid traverse commands such as G00 in the program.

$$\text{Z axis actual rapid traverse rate} = \text{P100} \times \text{rapid traverse override}$$

$$\text{X axis actual rapid traverse rate} = \text{P101} \times \text{rapid traverse override}$$

The rapid traverse override contains four gears: **25%**, **50%**, **75%**, **100%**. It controls all the rapid traverse commands and operations during the program execution.

The feedrate override can be changed with the rapid traverse +/-, keys/feedrate +/-keys. During the program execution, the actual speed of the slide carriage varies with the speed override.

The program execution is stopped when the feedrate override is 0 during the program execution, and the system prompts "Feedrate override is 0". By adjusting the feedrate override to a nonzero value, the program execution is continued.

#### 4.5.5.2 MPG Speed Control

In Auto operation mode, the user can also use a MPG to control the rapid traverse/feedrate override. When bit parameter **P402\_d1**=1, the rapid traverse/feedrate override controlled by an MPG is valid. The actual feedrate and rapid traverse rate are as follows:

$$\text{Actual feedrate} = F \times \text{feedrate override} \times \text{MPG override}$$

$$\text{Z axis actual rapid traverse rate} = P100 \times \text{rapid traverse override} \times \text{MPG override}$$

$$\text{X axis actual rapid traverse rate} = P101 \times \text{rapid traverse override} \times \text{MPG override}$$

$$\text{MPG override range: } 0 \sim 100\%.$$

Two types of MPG-controlled rapid traverse/feedrate override:

##### ◆ Type 1:

When bit parameter **P402\_d0**=0, the system enters MPG-controlled rapid traverse/feedrate override by type 1; the system sets the current position (marked as point A) of the MPG as the position where the MPG override is 0%. Within the 100 scales counted clockwise from point A, the MPG override increases by 1 when the MPG is rotated one scale clockwise, and decreases by 1 when the MPG is rotated one scale counterclockwise.

##### ◆ Type 2:

When bit parameter **P402\_d0**=1, the system enters MPG-controlled rapid traverse/feedrate override by type 2; the MPG override is set to 0%, and the MPG override varies with the speed at which the MPG is rotated clockwise. When the user rotates the MPG clockwise from slow to fast, the MPG override changes between 0%~ 100%. To avoid impacts to the machine due to the uneven MPG speed, the increment should not be less than 10% each time. If the MPG is stopped after CW rotation, the MPG override becomes 0%; if the MPG is immediately rotated CCW after CW rotation (or is stopped after CCW rotation), the MPG override keeps the CCW instantaneous override until it is rotated CW (or is stopped after CW rotation).

The use of the MPG-controlled rapid traverse/feedrate override is as follows:

In Auto operation mode, by pressing **【MPG】** key before executing the program or in the state of program pause, single stop, cycle stop or feed hold (including the external feedrate/spindle hold), the **【MPG】** key lights up, which indicates that the system is in the MPG-controlled rapid traverse/feedrate override mode. At this moment, on the upper right corner are displayed the MPG control mode and current MPG override with yellow ground. After completing the execution of the program, the system automatically cancels the MPG-controlled rapid traverse/feedrate override mode.

##### 【Note】

In thread machining commands (**G32, G33, G92, G76**), the feedrate speed is determined by the spindle speed instead of value F. Thus the MPG override has no effect on these commands.



## 4.5.6 Interruption Operation during Program Execution

### 4.5.6.1 Interruption with Keys during Program Execution

The interruption operations below can be performed during the program execution:

**Emergency stop:** The execution is stopped immediately, which cannot be continued.

**Pause** The execution can be continued by pressing **CYCLE START** key.

**Single stop:** The execution is stopped after the current block is executed, which can be continued by pressing **CYCLE START** key.

**Cycle stop:** The execution is suspended after the current cycle is finished, which can be continued by pressing **CYCLE START** key.

#### ◆ Pause

- 1) By pressing **CYCLE PAUSE** key, the execution is stopped during program execution. After the system responds, all the move axes are stopped, with **PAUSE** displayed on the lower left corner of the screen.
- 2) In the pause state, by pressing **CYCLE START** key, the system recovers the program and then continues executing the remaining blocks; by pressing **ESC** key, the system exits the program and then returns to the **initial state** automatically, with the pointer pointing at the first block of the current program.

#### 【Caution】

- 1) After pause, the spindle and tailstock are controllable. Before pressing **CYCLE START** key to perform execution, make sure whether the spindle is started and whether the chuck and tailstock are ready. Otherwise, it may cause damage to the machine or results in personal injury.
- 2) In executing G32, G33, G92, G76 command, this key is invalid when the system is in the block where the spindle machining thread is tracked.

#### ◆ Single stop

- 1) During Continuous program execution, by pressing **SINGLE** key, the system switches to single execution mode. After the current block is executed, the screen displays **Single stop**.
- 2) After Single stop, the program execution is continued if **CYCLE START** key is pressed; and the system returns to the initial state with the pointer pointing at the first block of the current program if **ESC** key is pressed.

#### 【Note】

- 1) In executing the canned cycle commands, the Single Stop is valid after each step in the cycle is completed.

#### ◆ Cycle end stop

- 1) During Continuous program execution, the screen displays **Cycle stop: ON** by pressing **hp6** key. The execution is stopped after M20 is executed, and the screen displays **Cycle stop**

#### 4.5.6.2 External feed/Spindle hold Knob

The External feed/Spindle hold knob is effective only in Auto operation mode.

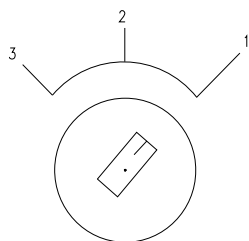
Parameter **P412\_d6** sets whether the External feed/Spindle hold knob is effective.

When **P412\_d6=1**, the system External feed/Spindle hold knob is effective; the input signals are led in through pin MXZ1, MXZ2.

When **P412\_d6=0**, the External feed/Spindle hold knob is ineffective; the input signal pins are used for other purposes.

##### ◆ Introduction to External feed/Spindle hold knob

This CNC system contains an interface for the External feed/Spindle hold knob. When the knob is rotated to different positions, the movement of the spindle and slide carriage can be enabled or disabled. When debugging the program, it is very convenient to control the movement and stop of the spindle and carriage. The External feed/Spindle hold knob has three positions, whose functions are as follows:



Feed hold knob

**Note:** For the sign indicating the the External feed/Spindle hold knob, refer to the user manual provided by the machine tool builder.

Position 1 enables the spindle rotation and the slide carriage movement.

Position 2 enables the spindle rotation, and disables the slide carriage movement.

Position 3 disables the spindle rotation, and enables the slide carriage movement.

##### ◆ Use of External feed/Spindle hold knob

**Before executing the program:**

When the External feed/Spindle hold knob is at position 1 and position 2, the spindle start/stop can be controlled by pressing keys manually; however, when it is at position 3, the spindle cannot be started.

**In Single operation mode:**

When the External feed/Spindle hold knob is located at position 1, all the commands are executed as usual; when it is at position 2, the spindle control commands can be executed rather than the move commands of X, Z axes. These move commands can be executed only when the knob is rotated back to position 1. When it is at position 3, no block can be executed. To execute the blocks, rotate the knob back to position 2 or position 1.

**In Continuous execution mode:**

After starting the program, the user can rotate the External feed/Spindle hold knob anytime to control the spindle and the movement of the slide carriage.

When the knob is located at position 1, the program is executed normally.

When the knob is rotated to position 2 from position 1, the slide carriage enters the pause state and stops moving, but the spindle state is maintained.

When the knob is rotated to position 3 from position 2, the spindle stops rotating.

When the knob is rotated to position 2 from position 3, the spindle recovers to the original state.

When the knob is rotated to position 1 from position 2, the carriage recovers to the movement state.

During the feed hold and spindle stop, if **ESC** key or **RESET** key is pressed, the system returns to the automatic initial state. Then the previous spindle state and the slide carriage unfinished commands cannot be maintained. The user can continue the machining only by restarting the program.

#### 4.5.6.3 External Start and Pause Signals

The external cycle start/pause signal, which is set by parameter **P412\_d7**, is effective only in AUTO operation mode.

When **P412\_d7=1**, the system external cycle start/pause signal is effective; the input signals are led in through pin **ST**, **SP**.

When **P412\_d7=0**, the system external cycle start/pause signal is ineffective; the input signal pins are used for other purposes.

The function of the external pause operation key signal (SP) is the same as that of the feed hold key (cycle pause key) on the panel; and the external cycle start key signal (ST) is the same as that of the cycle start key on the panel. Both SP and ST are output to the system from the machine, with the low level valid.

Before the system is switched to Auto operation mode from other modes, it is required that the “external start button” and “external pause button” be released (disconnected). Otherwise, the system issues an abnormality alarm.

For the connection, refer to Chapter Three CNC Device Connection in Part III Connection.

#### 4.5.6.4 Feeding Device Alarm Function

If the function is required, define its input pin in interface parameters, and perform the connection correctly; the system detects the signal at the corresponding pin.

**P512: Dalm: Detection signal of feeding device alarm.**

##### 【Function descriptions】

- 1) When executing M20 command, the system automatically stops and issues an alarm after detecting the signal.
- 2) If the program does not end with M20 command, but with M02 or M30 command, the feeding device alarm function is invalid.







#### 4.5.7 Modifying Tool Offset during Program Execution

In the machining, it is available to modify the tool offsets in a pop-up window. However, this operation should be performed with caution.

##### 4.5.7.1 Methods of Modifying Tool Offsets during Program Execution

###### ◆ Methods of modifying tool offsets

- 1) In the automatic operation, by pressing **OFFSET** key, the system displays a window, showing the tool offset modification page; by pressing **OFFSET** key again, the system closes the window, and so on.

- 2) Select the tool offset number to be modified by pressing ,  or ,  key; then select the tool offset item to be modified by pressing  or  key (See Section 4.7 “Tool Offset Operation Mode” in Part I Operation for details).
- 3) Press **ENTER** key, then input the data; if the input data is required to directly replace the original data, press **ENTER** key; if it is necessary to add the input data to the original data, press **ALTER** key; if it is required to cancel the data, press **ESC** key.
- 4) After the input, press **OFFSET** key or **ESC** key to exit the tool offset display page.

**【Note】**

- 1) In inputting the data, the user can switch back to display page of Auto operation mode anytime. Then after switching to the tool offset display page again, the previously-input data which has been confirmed by pressing **ENTER** key is retained, and the user can continue the data input.
- 2) On the tool offset modification page, the operations in Auto operation mode, such as dwell and feedrate override modification, are still valid. During dwell, the user can also switch to the tool offset modification page to modify the tool offsets.

#### 4.5.7.2 Modifying Validity of Tool Offsets during Program Execution

**【Warning】**

The tool offset data modified is valid only after the tool change is executed. If it is the tool offset data corresponding to the current tool offset number that is modified, the modified data is valid after the next tool change is executed. If it is the tool offset corresponding to the tool offset number which has not been executed that is modified, the modified data is valid in this execution.

If the tool offset of the current tool offset number is modified in the program without a tool change command, the modified tool offset becomes valid after executing M02, M30, M20, “Reset” or interruption.

#### 4.5.8 Viewing Running Information in AUTO Operation Mode

This function is valid in any state in Auto and Manual operation modes.










In the process of machining a part program automatically, a window can be displayed for the user to view three kinds of data, i.e. macro variable, I/O variable and others during the system operation. The explanations are as follows:

**Macro variable:** Viewing and modifying all the common variables; **(unsupported)**

**I/O variable:** Viewing the values (i.e. system interface states) of the system interface variables; **(unsupported)**

**Others:** Viewing the blocks that have been executed, spindle fluctuation range in threading, nesting call level in subprogram and program cycle.

◆ Operation methods are as follows:

- 1) In Auto operation mode, press  key to cause the system to display a window indicating a three-item (macro variable, I/O variable, others) page; by pressing ESC key, the system closes the window.
- 2) Press  or  key to select the item to be viewed. The item selected is displayed in black.
- 3) When there are many macro variables to be viewed, press  key to select a variable (pointed by the pointer). Here, the macro variable cannot be modified. Then press  or  key to view the last or the next macro variable, or press  or  key to view the last page or next page of macro variables (seven lines per page); press  key to return, then the macro variable is displayed in black.
- 4) In automatic initial state, the common variable at which the pointer points can be modified by pressing ENTER key.
- 5) After viewing the variables, press ESC key to exit the display page.

【Explanations】

- 1) Macro variable: Displaying the common variables being executed in the program where they are edited, including the variable sequence number, variable name, variable value and state. Therein, the variable values varying with the program execution are dynamically displayed, and the sequence number as well as the variable name are arranged in ascending order. Variables r001~r040 are displayed in brown, variables r041~r99 are in orange, and variables r100~r199 are in green.
- 2) I/O variable: Dynamically displaying the interface state of the current program. The two state values of the input interface variable are: 0 (low level) or 1 (high level); The low level indicates that the external signal is valid, and the pin is connected to 0 V; the high level indicates that the external signal is invalid. Variable state of the input interface: When “0” is output, the external can form a return circuit. When “1” is output, the external cannot form a return circuit. r1001~r1032 are the states of the input interfaces, and r2001~r2032 are the states of the output interfaces; These states are classified into 8 groups. The first line displays the first group: r1008, r1007, r1006, r1005, r1004, r1003, r1002, r1001, and so on.
- 3) Others: Viewing the total number of blocks that have been executed, spindle fluctuation range in threading, nesting call level of the subprogram and program cycle.

- 4) Before starting the program, the macro variables r100~r199 can be modified manually. Modification method: Select the macro variable to be modified, press **ENTER** key to input the desired value, and then press **ENTER** key again to complete the modification.

#### 4.5.9 Return to Program Reference Point in AUTO Operation Mode

This function is valid in the **initial state** in Auto operation mode.

This system allows the program to be started wherever the slide carriage is after the user sets the workpiece coordinate system and program reference point. In this case, both the operations of program zero point return by pressing keys and program reference point return by G commands return to the position of the set program reference point.

After returning to the program reference point using a G command, if it is required to continue the machining, use G00 command to simultaneously position the absolute coordinates of the two axes, in order to ensure the machining is correct.

After returning to the program zero point by pressing keys manually, the system automatically points to the first block of the program. Here, by pressing **CYCEL START** key, the system starts execution from the first block of the program.

#### 4.5.10 System Reset Key and Emergency Stop Signal Processing in Auto Mode

In Auto operation mode, the system enters the Reset state by pressing Reset key. See Section 4.1.4.7 "Reset Operation" in Part I Operation.

For details about the emergency stop signal in Auto operation, see Section 4.1.4.3 "Emergency Stop Alarm" in Part I Operation.

##### 【Note】



- 1) **Before releasing the emergency stop alarm, first confirm the fault has been removed;**
- 2) **Pressing down the emergency stop button before Power On and Power Off can reduce the surging to the equipment.**
- 3) **Re-perform machine zero return after releasing the emergency stop alarm, to ensure the coordinate position is correct (if the machine is not equipped with the machine zero, it is forbidden to perform machine zero return);**
- 4) **The external emergency stop is valid only when bit parameter P404\_d7 is set to 0.**

#### 4.5.11 Adjusting Brightness of LCD screen in AUTO, MANUAL Operation Mode

This function is valid in the **initial state** in AUTO operation mode.

##### ◆ The operation method is as follows:

- 1) Press **9** key twice rapidly to cause the system to display the brightness adjusting window; the windows is closed by pressing any key other than the brightness adjusting key.

- 2) The brightness adjusting window contains 0~10 levels: level 0 is the darkest, and level 10 is the brightest; Press brightness adjusting key  to increase the LCD brightness, press brightness adjusting key  to reduce the LCD brightness.

**【Note】**

- 1) When the LCD has LED backlight, the brightness adjusting function is valid for the LCD screen; when the LCD has CCFL backlight, the function is invalid for the screen.
- 2) The brightness adjusting window will disappear in 10 seconds if no key operation is performed.
- 3) It is also available to adjust the LCD screen brightness in Manual operation mode in the same way as in Auto operation mode.

#### 4.5.12 Displaying Executing States of M Commands in Auto, Manual Operation Mode

The displayed M commands are classified into the following 6 groups: M21/22, M23/24, M61/62, M63/64, M65/66, other M commands: M60, M67~M74, M81/82/83. In Manual/Auto operation mode, if the M codes above are executed, the screen displays corresponding prompts. The M commands being executed are displayed in red; the M commands which have been executed are in green; the M commands which are terminated during operation are in yellow.

#### 4.5.13 Additional Operation in Auto Operation Mode


When executing the program in Auto operation mode, if there is an interlock relationship between spindle and chuck, execute M3 to start the spindle during M11; there are two choices after an alarm occurs:

- 1) Press ESC key to exit to the initial state in Auto operation mode.
- 2) Pedal the footswitch to execute M10; or execute M3 to start the spindle by pressing CYCLE START key; or exit to the initial state in Auto operation mode by pressing ESC key.

### 4.6 Parameter Operation Mode

The functions of **Parameter operation mode** include: parameter input, parameter extraction, parameter solidification and parameter transfer; for each operation, the system provides corresponding intelligent prompt messages. Also, the user can press the hp2 key on the upper right corner of the system to view the list of the system parameter operation keys.

The descriptions for the input formats and examples of related setting and operation in the manual are as follows: the function keys to be pressed are indicated by a sign; the letter keys, numeric keys to be input are indicated by an underline; the prompt messages of the system are indicated by a rectangle.


In inputting a letter or a numeric, if a wrong one is input, press  key to delete it, and then input the correct one.

When setting or inputting an item, or during the man-machine dialogue, if ESC key is pressed before confirmation, it means the current operation is quit.

**Special note: Before modifying the parameters, save all the parameter setting values stored in the system into a PC or laptop. Once the parameters are modified by mistake or system faults occur, the user can recover the system using the stored data.**

By the easy to search and remember principle, the parameters are classified into the following types: reference coordinate parameters, motion parameters, drive parameters, auxiliary function parameters, bit-control parameters, interface parameters, variable initial value parameters and pitch error compensation parameters. Each parameter has a definite meaning, which determines the operation mode of the CNC system and machine. When installing and debugging the machine, it is necessary to modify some parameters depending on the actual conditions.



Press operation mode select key  to enter Parameter operation mode page. (If the system displays a window requiring passwords, input the user password or directly press Enter key to enter the Parameter interface).

[ PARAMETER]	Programmer	hp2
M-REFER PARAM	<u>P000</u> Zprogram reference point	303.698
X-MOTION PARAM	P001 X program reference point	203.698
Z-DRIVE PARAM	P002 Y program reference point	0.000
S-AUXILIARY PARA	P003 Z axis 2nd prog refer point	200.000
T-BIT PARAM	P004 Z axis 2nd prog refer point	200.000
U-INTRFACE	P005 Y axis 2nd prog refer point	200.000
W-VARIABLE	P006 Z axis 3rd prog refer point	200.000
F-PITCH PARAM	P007 X axis 2nd prog refer point	200.000
EDIT	JOG	AUTO
PARAM    OFFT    DNG		

Fig. 4-8 Parameter operation mode



### 4.6.1 Parameter Overview

The operation characteristics of the parameters include:

- 1) According to the level of the password that the user input, the parameters corresponding to this level can be modified.
- 2) If **ENTER** key is pressed without inputting any password, the operation level is level 4. The user can enter the parameter interface to view the parameters here rather than modify them.
- 3) The input and display format of the parameter: a decimal should include a decimal point, and a negative number should include a negative sign; for the convenient and safe operation, the system limits the number of effective digits.
- 4) The user can open the message window which prompts the input range of the parameter data.
- 5) The system self-checks the parameter data at power on, and prompts automatic initialization after detecting the data is in mess.
- 6) The system provides abundant and applicable safety parameters. Therefore, the user can reduce the misoperation by setting the parameters appropriately, so as to prevent accidents.

#### 4.6.1.1 Parameter Authority

The parameter authority is the password level to modify parameters. For convenient management, the system provides the parameter authority setting function. The current operation level is displayed on the prompt column on the top of the parameter page.

Arranged from high to low, the parameter password levels are as follows:

- |          |      |                        |   |
|----------|------|------------------------|---|
| Level 1: | **** | Machine tool builder,  | range of alterable parameters: Parameter level $\geq 1$ ; |
| Level 2: | **** | Device administrator,  | range of alterable parameters: Parameter level $\geq 2$   |
| Level 3: | 3333 | Machine tool operator, | range of alterable parameters: Parameter level $\geq 3$   |
| Level 4: | **** | No password input,     | range of alterable parameters: Parameter level $\geq 4$   |

For the parameter levels of each type of parameter, see the parameter lists in the appendix.

#### 4.6.1.2 Entering an Operation Level

The procedures of entering different operation levels are as follows:

- ① Enter parameter password input page;
- ② Input the operation password (each time a number is input, a sign "\*" is displayed.)
- ③ After the input, press **ENTER** key to enter the operation level corresponding to this password.

#### 4.6.1.3 Parameter Management

The parameter management consists of the parameter display, parameter authority, and parameter initialization, solidification, extraction, sending and receiving according to the parameter authority.

##### 【Parameter display】

##### Meanings of the parameter colors:

In the current authority, the parameters allowed to modify are displayed in yellow, the parameters not allowed to modify are displayed in white;

On the parameter page, some operation function options are also relative to the authority; e.g. the function options which are not operable after pressing **hp6** key are displayed in grey-white.

#### **Prompt message display:**

When inputting a parameter, the user can open or close the prompt message about the parameter value range by setting parameter **P415\_d7**.

#### **Other displays:**



If the user tries to modify the parameters with a higher authority than the current one, the system displays the alarm message **Without modification authority!** on the parameter setting area.

On the parameter page, if a parameter is modified successfully, a sign “\*” is marked in front of its parameter number, indicating the parameter modification succeeds.

After completing the some operations of **hp6**, the system displays the operation result, and marks a sign “\*” ahead of the parameter number which is successfully modified, indicating the parameter modification succeeds.

#### **【Parameter authority】**

In different authorities, the alterable parameters are displayed in yellow, and the unalterable parameters are displayed in white. The parameter update (e.g. parameter update through a serial port or USB) can only modify the parameter data of the current user authority or lower.

The sign “” before a parameter means the parameter is alterable under the current password level; “” means the parameter is unalterable.

#### **Authority modification**

The authority is modified using passwords, i.e. inputting the password when entering the parameter password input page. If the password input page does not appear when entering the parameter page, the user can set whether to memorize the password by setting bit parameter **P416\_d7**. When **P416\_d7=0**, the password is not memorized. In this case, after entering the parameter page from other operation modes, the parameter password input page will appear. The operations are as follows:

- 1) Set bit parameter **P416\_d7** to 0;
- 2) Press any operation mode key other than **PARAMETER** operation key;
- 3) Press **PARAMETER** key to enter the parameter password input page;
- 4) Input the operation authority password directly;
- 5) After the input, press **ENTER** key to enter the operation level corresponding to the password.

#### **【Parameter saving】**

The successfully-modified parameter is automatically saved into the CNC system. In addition, when exiting from the parameter page (i.e. pressing other operation mode keys to enter other operation mode pages), the system saves all the parameters. The parameter data saved in the system are read each time the system is powered on. If the system detects a parameter exceeds the range during parameter reading, the system replaces the value by the one within the range and issues a prompt. If the parameters read during Power On are in chaos, the system prompts whether to read the original solidified parameters; if there is no solidified parameter, the system prompts the user to select

Step/Servo parameters to initialize the parameters, and save them to the system. The main difference between step parameters and servo parameters is the motion parameter value.

## 4.6.2 Modifying Parameters

The system parameters have been initialized in factory. After installing the system to the machine, please modify and debug the parameters appropriately depending on the actual conditions.

After a parameter number is selected, it is highlighted by the system.







### 4.6.2.1 Searching Parameters

The parameter search is to search the desired parameter; there are two methods:

#### Method 1:

- ◆ On the parameter page, select: M - reference parameters, X - motion parameters, Z - drive parameters, S - auxiliary parameters, T - bit parameters, U - interface parameters, W - variable initial values or F – pitch error compensation parameters, as required.

E.g. select M – reference parameters, then press **M** key to enter the reference parameter page.

- ◆ Press ,  key or ,  key to move the highlighted block cursor to the parameter number to be searched; for bit parameters, press  or  key to move the cursor leftwards or rightwards to select different bits, with the meaning of the selected bit displayed at the same time.


#### Method 2:

Locate the desired parameter number directly. The procedures are as follows:

Press **P** key, and input the parameter number to be searched, then press **ENTER** key. The system then displays the desired parameter with a highlighted block cursor. E.g., to search P208, first input **P**, and input **2** **0** **8**, then press **ENTER** key to seek parameter P208.

### 4.6.2.2 Modifying Parameters

The procedures of modifying parameters are as follows:

- ① Seek the parameter to be modified using the method above.
- ② Press **ENTER** key, then input the parameter data; or directly input the parameter data.
- ③ If wrong data are input, press  key to delete them, then input the right ones.
- ④ Press **ENTER** key.

#### 【Note】

- 1) If the data input exceeds the specified parameter range, the data is invalid, and the contents of the parameter are not changed.
- 2) After inputting the data, the input data is invalid if **ESC** key is pressed.
- 3) The procedures of inputting bit parameters are as follows:

- ① After selecting the parameter to be modified, select the parameter bit to be modified using left and right keys (the explanation for the currently selected bit is displayed on the bottom of the screen).
- ② Modifying a single bit: Directly input the desired data (only “0” or “1”).
- ③ Modifying all bits: the method is almost the same as that of manual setting of common parameters. The only difference is that the data is input from left to right. E.g. input 11, and press **ENTER** key, then the parameter is modified to: 00000011; input 11000000, and press **ENTER** key, then the parameter is modified to 11000000.
- ④ Bit parameter **P411\_d6**: if it is changed, the pitch error compensation parameters P1000~P1899 are initialized to 0, i.e. all the values of previous pitch error compensation parameters are changed to 0.

### 4.6.3 Parameter **hp6** Function

Press **hp6** key on the parameter page to display the following:

Key U – USB interface operation Key R – RS232 interface operation Key I –Parameter extraction Key K – Parameter solidification Key F – Software update Key D –Memory update
--

According to the password level, the user can communicate, extract and solidify the parameters as well as update the system software and memory; “I” is for parameter extraction; “K” for parameter fixture; “F” for software update (the system software is upgraded with USB or RS232); “D” for internal memory update.

During data solidification and extraction, please do not turn off the power, and do not perform other operations before the operation is completed. The data solidification and extraction have no effect on the part programs stored in the system.

#### 4.6.3.1 Parameter Communication and Standard Format

There are two types of communication depending on the parameter transmission direction: Parameter sending and parameter receiving. The parameter receiving includes three modes: **PC→CNC**, **USB→CNC**, **CNC→CNC**; the parameter sending also includes three modes: **CNC→PC**, **CNC→USB**, **CNC→CNC**.

**Parameter sending:** (Operation level: all levels)

The operators of all levels can send the parameters to a U disc or a PC through RS232 serial port.

**Parameter receiving:** (Operation level: Machine tool builder, device administrator, machine operator)

The operators of level 3 or above can receive the parameters from a U disc or PC (RS232). However, it is only valid for those parameters which can be modified in the corresponding level.

For the operation of the communication software that uses RS232 for transmission, see Chapter Five RS232 and USB System Communication in Part I Operation. GSKCOM\_C001.EXE can realize the file transmission between PC and CNC, which is easy to operate and has high communication efficiency and reliability.

RS232 and USB interface function: for transmitting parameter data.

Press **R** key to transmit data by means of RS232 communication mode.

### I . RS232 parameter receiving: PC→CNC, CNC→CNC

1. Set the baudrate and communication port of the communication software; see Chapter Five RS232 and USB System Communication in Part I Operation.
2. Input the parameters saved in the external computer to the CNC system; or transmit the parameters between CNC systems.
  - 1) Connect the communication cable between CNC system and computer, or between CNC systems at Power Off.
  - 2) Turn on the CNC system and select the Parameter setting operation mode. Set a corresponding authority in the CNC, and the received parameters can only modify the parameters with a corresponding authority level; Before inputting a password, the CNC disables the parameter receiving; note: The transmission between CNC systems only transfers the parameters with the same password level.
  - 3) First press **hp6** key, then press **R** key
  - 4) Press **1** key to receive parameters.
  - 5) The operation is completed, and the system prompts the receiving succeeds

### II . RS232 Parameter sending: CNC→PC, CNC→CNC

1. RS232 baudrate setting

Before transmitting files, set the baudrates of bit parameter **P414\_d7** and **P414\_d6**. The communication baudrate is subject to the setting of the sender. Setting range: 9600, 19200, 38400 (unit: bps)
2. Input the parameters stored in the external computer to the CNC system; or transmit parameters between CNC systems.
  - 1) Connect the communication cable between CNC system and computer, or between CNC systems at Power Off.
  - 2) Turn on the system and enter the parameter page.
  - 3) First press **hp6** key; then press **R** key.
  - 4) Press **2** key to send the parameters.
  - 5) The operation is completed, and the system prompts the sending succeeds.

### III USB operation: USB→CNC, CNC→USB

First press **hp6** key, then **U** key to transmit data using USB communication mode. The user should select the transmission direction as required.

When using a U disc to transmit parameters, create a folder with file name “C001PAR” in its root directory, and the receiving and sending of parameters are in this folder. Format of file name: “PAR” + file number (three digits) + “.TXT”

**IV Standard format of TXT parameter file on PC**

On a PC, the parameter files can be edited with TXT or LST text. However, to send the files correctly to the system, the file name and the contents of the file must be in accordance with the requirements of the system. The requirements are as follows according to the parameter file pattern output by the system.

- 1) On a PC, the suffix of the file name should be TXT or LST, e.g. **PAR099.TXT**; it is suggested that the user use TXT suffix, for the convenience of the operation on PC.
- 2) The first line of the contents of the TXT file must be the parameter mark: “CNC\_GSKC001”; this is an essential part.
- 3) The second line is the comment part, in front of which should be placed “//”; this item is omissible.
- 4) The third line and the following lines are the contents of the parameter. The contents must be in accordance with the requirements of the standard format.

E.g. **P000**: 00000.000 // Z program reference point

Hereinto, **P000** is the parameter number; the format of the parameter number includes three parts: “P + number + :”. If any part of them is missing, the parameter number is incorrect. **00000.000** is the contents of the parameter; “//” is placed in front of the parameter comment.

- 5) The contents of the file can be part of all the parameters.
- 6) The pitch error compensation parameters include equal-interval pitch compensation parameters and unequal-interval pitch compensation parameters. Select the appropriate format for pitch error compensation parameters according to the parameter types to be updated. The selection of formats of the pitch error compensation parameter is relative to bit parameter **P411\_d6**: When **P411\_d6**=0, the pitch error compensation has equal interval, when **P411\_d6**=1, the space of the pitch error compensation has unequal interval. If the selected pitch error compensation parameter does not match corresponding bit parameter **P411\_d6**, the pitch error compensation parameters cannot be updated successfully. The format is as follows:

◆ **Format of equal-interval pitch error compensation parameter**

```
// Pitch error compensation parameter; starts with P1000 in TXT file
P1000:    0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0
P1010:    0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0
P1020:    0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0
.....
P1880:    0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0
P1890:    0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0
P1900:    0 // Machine coordinate of Z axis start point
P1901:    0 // Machine coordinate of X axis start point
P1902:    0 // Machine coordinate of Y axis start point
P1903:    0 // Pitch error compensation interval of Z axis
P1904:    0 // Pitch error compensation interval of X axis
P1905:    0 // Pitch error compensation interval of Y axis
```

◆ Format of unequal-interval pitch compensation Parameter:

```
// Pitch error compensation parameter; starts with P2000 in TXT file
P2000:    0000.000, 0, 0000.000, 0, 0000.000, 0, 0000.000, 0, 0000.000, 0
P2010:    0000.000, 0, 0000.000, 0, 0000.000, 0, 0000.000, 0, 0000.000, 0
.....
P2880:    0000.000, 0, 0000.000, 0, 0000.000, 0, 0000.000, 0, 0000.000, 0
P2890:    0000.000, 0, 0000.000, 0, 0000.000, 0, 0000.000, 0, 0000.000, 0
P2900:    0000.000 // Machine coordinate of Z axis start point
P2901:    0000.000 // Machine coordinate of X axis start point
P2902:    0000.000 // Machine coordinate of Y axis start point
P2903:    0000.000 // Pitch error compensation interval of Z axis
P2904:    0000.000 // Pitch error compensation interval of X axis
P2905:    0000.000 // Pitch error compensation interval of Y axis
```

7) Standard format of parameter file communication on PC:

TXT file format	Remarks
<pre>CNC_GSKC001 //reference coordinate; P000:    00200.000 // Z program reference point ..... P027:    0 // Y zero offset //motion parameters; P100:    6000 // Z max. rapid traverse rate ..... P209:    1200 // Spindle encoder lines //Auxiliary function; P300:    1000 // M41 gear max. speed ..... P344:    0 //bit-control parameter; P400:    00000000 // Running setting ..... //interface parameter; P512:    0 // Feed device alarm detection Dalm ..... //variable initial value; P600:    0 // variable r01 ..... //Pitch error compensation parameter; P1000:    0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 ..... ..... P1905:    0 //Y pitch compensation interval</pre>	<p><b>Mark for checking parameters, which cannot be omitted.</b></p> <p><b>The contents behind // is the comment, which can be omitted.</b></p> <p><b>PXXX parameter number, P000: the first parameter number cannot be omitted.</b></p> <p><b>PXXX: the parameter numbers after the first number can be omitted, each of which is regarded as the parameter following the last one.</b></p> <p><b>0: the parameter contents can be omitted.</b></p> <p><b>PXXX: the parameter number and colon : must not be separated.</b></p> <p><b>.....: indicates other parameters, which are not listed. The parameters can be partly received.</b></p>

4.6.3.2 Parameter Extraction and Solidification

The parameters which can be viewed on the interface are saved in the SRAM memory of the system. The memory has the power failure protection function; if the battery of the mainboard fails , the parameters will be lost.

The system is provided with the parameter initialization function; please perform the initialization operation depending on the matched motor drive unit; for the system matched with a DA98 series drive unit, perform the servo initialization, and for the system matched with a DY3 series drive unit, perform the step initialization. The difference between servo and step initialization parameters lies primarily in the

X- motion parameter. For the difference between servo/step values, see the motion parameter list in the appendix.

The initialization parameters are not applicable to all the machines. The machine tool builder should modify the parameters, such as the spindle and tool post, according to the specific machine configuration.

To prevent the parameters from being lost, the user should perform the solidification command. That is, to solidify the parameters that have been modified to the FLASH memory of the system as a backup. The FLASH memory is capable of storing the data permanently without any battery. If the current parameters are lost, recover them by extracting the solidified parameters.

The user can perform the parameter extraction and solidification according to the prompts displayed in a pop-up dialog box. The parameter extraction commands include the step parameter initialization, servo parameter initialization and machine tool parameter extraction.

#### 【Explanations】

- 1) Before performing the parameter solidification, the system checks corresponding parameters. If the check is not passed, the system prompts an alarm message requiring the user to modify the corresponding parameters; if the check is passed, the user can save the parameters to the FLASH of the system for solidification.
- 2) If no parameters are solidified before, the user cannot perform solidified parameter extraction.

#### 4.6.3.3 System Software Upgrade and Internal Memory Update

The system software upgrade is to upgrade the system software, i.e. to replace the old version of software by the new version; so as to improve the system functions.

The system update is to update and improve the system software, so as to make the system more stable. It does no harm to the system, but incorrect operation may result in the system update failure. If the update fails, the system cannot be started up, or even the system hardware will be damaged. Therefore, we suggest the user not upgrade the system software without authorization. It is recommended that the upgrade service be provided by our professional staff with our after-sale service center.

The system memory upgrade is to upgrade the system memory, including the memory occupied by the system software.

**If the memory upgrade fails, serious consequences may occur. It is suggested that the operation be performed by our professional technician instead of the user.**

The methods of upgrading the system and memory include: USB mode and RS232 mode; the operation can be performed only with the authority of the machine tool builder.

- 1) System software upgrade using USB mode:

After inserting a U disc, operate the system based on the system prompts; then the system automatically completes the software upgrade. When using USB mode for upgrade, create a folder with the name "C001DATA" in the root directory of the U disc. The sending and receiving of codes are in the folder. Format of the file name: "DATA" + file number (three digits) + ".TXT". Range of the file number: 0~254.



2) System software upgrade using RS232 mode:

The system can be upgraded by using the communication software GSKCOM\_C001.EXE. For the operation of the communication software with RS232 mode, see Part I Operation: Chapter Five “RS232 and USB System Communication” .

When upgrading the memory with USB mode, create a folder with the name “C001MEMO” in the root directory of the U disc. The receiving and sending of codes are in the folder. Format of the file name: “MEMO”+ file number (three digits) +“.TXT”. Range of the file number: 0~254. When upgrading the memory with RS232 mode, complete the operation using the communication software GSKCOM\_C001.EXE.

**【Note】**

- 1) After the upgrade using USB mode succeeds, if **ESC** key or **RESET** key is pressed, the user needs to return the control to Edit operation mode to re-edit the current program, or the system issues an alarm.

4.6.3.4 Function Command Authority

On the parameter page, some operation function options are also relative to the authority; it is not allowed to operate those function options displayed in grey white.

Correspondence between function options and password authorities

Operation authority Operation option	Level 1 Machine tool builder	Level 2 Device administrator	Level 3 Machine tool operator	Level 4 No password
Step, servo parameter initialization	★			
Extracting parameters of machine tool builder	★	★	★	
Parameter solidification operation	★	★	★	
Parameters received with USB and RS232	According to parameter level	According to parameter level	According to parameter level	According to parameter level
Parameters sent with USB and RS232	All	All	All	All
System software update and memory update	★			

The “★” in the table indicates the required authority; the space indicates no authority is required.

4.6.4 Description of Parameters

The system parameters are classified based on their use and functions. Their functions are as follows: (Y axis function is invalid)

4.6.4.1 Parameters of Reference Point, Soft Limit \_\_ P000~P020

The reference coordinate parameters include the important coordinate position points of machine Z/X/Y axes. The movement of each axis is based on these points.

**【Program reference point of Z/X/Y axis】 \_\_ Reference coordinate parameters P000, P001, P002**

This parameter is used for setting the position of the program reference point. In Manual/Auto operation mode, the machine returns to this position after performing the program reference point return. The setting of the position can be modified by inputting a command (in Manual operation mode) or executing G50 command (in Auto operation mode) .

The coordinates of the program reference point are the machine coordinates, which are not affected by the tool offset.

**【Second, third program reference point of Z/X/Y axis】 \_\_ Reference coordinate parameters P003, P004, P005, P006, P007, P008**

The second and third program reference points resemble the program reference point: in Auto operation mode, after specifying G30 command, the machine returns to the second or third program reference point specified by G30 command. For details, see Section 4.11 “G30-Return to 2nd, 3rd Program Reference Point” in Part II Programming.

**【Positive, negative tool nose software limit of Z/X/Y axis】 \_\_Reference coordinate parameters P009, P010, P011, P012, P013, P014**

This parameter is used to limit the motion range of the tool nose coordinates. It determines the max. strokes of the tool post on the positive and negative direction of Z/X/Y axis.

In Manual/Auto operation mode, if the tool nose coordinates of Z/X/Y axis are greater than or equal to the positive soft limit values of the tool nose, the tool cannot be moved in the positive direction any more. It can only be moved in the negative direction. Otherwise, the system issues an alarm: **Positive tool nose software limit alarm**. This also applies to negative movement.

**【Positive, negative machine software limit of Z/X/Y axis】 \_\_ Reference coordinate parameters P015, P016, P017, P018, P019, P020**

This parameter is used for limiting the motion range of the machine coordinates. It determines the max. strokes of the machine in the positive and negative direction of Z/X/Y axis.

In Manual/Auto operation mode, if the machine coordinates of Z/X/Y axis are greater than or equal to the positive soft limit values of the machine, the tool cannot be moved in the positive direction any more. It can only be moved in the negative direction. Otherwise, the system issues an alarm: **Machine positive soft limit alarm**. This also applies to negative movement.

**4.6.4.2 Parameters of Zero Return Function \_\_ P021 ~ P026, P109 ~ P111, P406 ~****P407****【Machine zero coordinates of Z/X/Y axis】 \_\_Reference coordinate parameters P021, P022, P023**

This parameter determines the coordinates of the machine zero point position. If the machine is equipped with the device to detect the machine zero point, and P407\_d1 is set to 0, after the “Machine Zero Return” (or G28) is executed in Manual/Auto operation mode and the system detects the “Zero

Signal”, the system automatically modifies the current machine coordinates to the values set in P021/P022/P023.

**【Zero offset value of Z/X/Y】** \_\_ Reference coordinate parameters P024, P025, P026

When using a servo motor, in general, the zero return detection is to detect the deceleration signal and the zero point rotation signal of the motor encoder; When the deceleration signal installed on the axis coincides with the zero point rotation signal, the zero return precision may be affected (the difference may be one revolution); in this case, set the offset to 2mm to avoid 2mm, then perform the detection.

**【Zero-return low speed of Z/X/Y axis】** \_\_ Motion parameters P109, P110, P111

Zero-return low speed is the traverse speed of the axis when detecting the zero signal during the execution of machine zero function. When the zero return speed is higher than the minimum start speed, the zero return is performed at the minimum start speed. When the zero-return speed is lower than the start speed, the zero return is performed at the zero-return speed. After setting the zero-return speed, do not modify it freely. Otherwise, the zero-return precision may be affected.

**【Zero setting 1】** \_\_ Bit parameter P406 (password level: 1)

d7	d6	d5	d4	d3	d2	reserved	d0
----	----	----	----	----	----	----------	----

d7\_\_ With/without Z deceleration signal

d6\_\_ With/without X deceleration signal

d5\_\_ With/without Y deceleration signal

0: Without. // If the machine is not equipped with a deceleration switch or stopper, set it to 0;

1: With. // If the machine is equipped with a deceleration switch and stopper, set it to 1;

d4\_\_ With/without Z zero point signal

d3\_\_ With/without X zero point signal

d2\_\_ With/without Y zero point signal

0: Without. // If the machine is not equipped with a machine zero switch or stopper, set it to 0;

1: With. // If the machine is equipped with a machine zero switch and stopper, set it to 1;

// The system provides four types of zero return (see CONNECTION for details) as follows:

Zero return type	Deceleration signal	Zero signal	Remarks
1	With	With	Return to machine zero using both the deceleration signal and zero return signal
2	With	Without	Return to machine zero only using the deceleration signal
3	Without	With	Return to machine zero only using the zero signal.
4	Without	Without	Return to the zero coordinates set by parameters (without a machine zero)

**d0\_\_ Without machine zero, the zero-return key is:**

**0: Disabled.** // When the system has no machine zero, the “Machine zero return key” on the panel is disabled, but G28 zero return in the program is still enabled.

**1: Enabled.** // When the system has no machine zero, the “Machine zero return key” on the panel is enabled, and the system returns to the machine zero by means of type-4 zero return.

// When neither the machine zero switch and stopper, nor the deceleration switch and stopper are equipped, the system returns to the machine zero by type-4 zero return, i.e. returning to the zero coordinates set by parameters when there is no machine zero.

**【Zero setting 2】 \_\_ Bit parameter P407 (password level: 1)**

d7	d6	d5	d4	d3	d2	d1	Reserved
----	----	----	----	----	----	----	----------

**d7\_\_ Zero return direction of Z axis****d6\_\_ Zero return direction of X axis****d5\_\_ Zero return direction of Y axis**

**0: Positive.** // If the deceleration switch is installed at the end of the positive direction for the coordinate axis, the parameter is set to 0, and the system moves towards the positive direction to detect the zero point;

**1: Negative.** // If the deceleration switch is installed at the end of the negative direction for the coordinate axis, the parameter is set to 1, and the system moves towards the negative direction to detect the zero point;

**d4\_\_ Deceleration signal level of Z axis****d3\_\_ Deceleration signal level of X axis****d2\_\_ Deceleration signal level of Y axis**

**0: Low.** // The system decelerates to stop during axis zero return when detecting the deceleration signal is the low level.

**1: High.** // The system decelerates to stop during axis zero return when detecting the deceleration signal is the high level.

**d1\_\_ Whether to modify machine coordinates after the machine zero is returned.**

**0: Modify machine coordinates** // In general, set it to 0. The machine coordinates are modified after zero return.

**1: Not modify machine coordinates** // Set it to 1 when debugging zero return function, then the zero return precision can be known visually.

#### 4.6.4.3 Parameters of Movement Speed and Acceleration Time \_\_P100~P108, P112~P119

To enhance machining efficiency, the system can be applied to motors of different types or machine tools of different loads by adjusting parameters P100~P116, etc. E.g. When using a stepper motor, adjust the parameters to a lower value to prevent stepout; when using a servo motor, adjust the parameters to a higher value appropriately.

For details, refer to Section 4.6.5 Appendix in Part I Operation. The stepper initial values differ from the servo initial values in the motion parameter list

**【Max. traverse speed limit of Z/X/Y】 \_\_Motion parameters P100, P101, P102**

Parameters **P100, P101, P102** determine the rapid traverse rate for X, Y, Z axis respectively in manual rapid traverse and G00 command. Meanwhile, the actual rapid traverse rate of Z, X, Y axis is also controlled by the rapid override.

Actual rapid traverse rate of Z axis = P100 × rapid override (unit: mm/min)

Actual rapid traverse rate of X axis = P101 × rapid override (unit: mm/min)

Actual rapid traverse rate of Y axis = P102 × rapid override (unit: mm/min)

**【Minimum initial speed of Z/X/Y】 \_\_Motion parameters P103, P104, P105**

Parameters **P103, P104, P105** determine the minimum initial speed for Z, X, Y axis respectively when the axis movement is performed in MANUAL operation mode or G00 is performed in AUTO operation mode; When the speed of Z, X, Y axis is lower than the value set in **P103, P104, P105**, the value set in P103, P104, P105 takes precedence.

Adjust the value of this parameter to a proper value based on the actual loading of the machine tool.  
(Unit: mm/min)

**【Z/X/Y rapid feed acceleration/deceleration time (ms)】 \_\_Motion parameters P106, P107, P108**

Parameters **P106, P107, P108** determine the time that the speed linearly increases from 0 to 15m/min for, X, Y, Z axis respectively when the axis movement is performed in MANUAL operation mode or G00 is performed in AUTO operation mode; (X axis is programmed in radius ). The greater the values of **P106, P107, P108**, the longer the acceleration process of Z, X, Y axis. To improve machining efficiency, decrease the values of P106, P107, P108 as much as possible on the condition that the load characteristics are satisfied. (Unit: ms)

**【Initial speed of cutting feed】 \_\_Motion parameter P112**

P112 cutting initial speed. Parameter **P112** determines the initial speed of cutting commands such as G01, G02, G03, G05 during system automatic machining. (Unit: mm/min)

**【Max. speed limit of cutting feed】 \_\_Motion parameter P113**

P113 maximum speed of cutting feed. P113 determines the maximum speed of cutting commands such as G01, G02, G03, G05 during system automatic machining. When the F speed value specified by the program is greater than the one set in P113, the latter takes precedence. When the thread cutting speed exceeds P113, the system issues an alarm, terminating the thread machining. (Unit: mm/min)

**【Cutting feed linear/exponential acceleration/deceleration time (ms)】 \_\_Motion parameters P114, P115**

**P114** Feed linear acceleration/deceleration time; **P115** Feed exponential acceleration/deceleration time;

Parameters **P114 and P115** determine the time that the speed of cutting commands such as G01, G02, G03, G05 accelerates from 0 to 15m/min during system automatic machining. (Unit: ms)

**【Thread cutting acceleration/deceleration time (ms)】 \_\_Motion parameter P116**

Parameter **P116** determines the time that the speed of the thread cutting axis accelerates from 0 to 15m/min. The smaller the P116, the shorter the generated distance. If **P116** is too small, the stepper motor may step out. (Unit: ms)

**【Thread run-out acceleration/deceleration (ms)】 \_\_Motion parameter P117**

When there is run-out in thread cutting, parameter **P117** determines the acceleration/deceleration time is the time that the speed accelerates from 0 to 15m/min. The smaller the P117, the steeper the run-out. If **P117** is too small, the stepper motor may step out. (Unit: ms)

**【G99 initial lines】 \_\_Motion parameter P118**

If the parameter is set to 0, the system detects the lines of the spindle encoder each time it executes G99 command. It starts to execute the commands following G99 when the encoder rotates to the lines set in P118, i.e. the spindle rotates to a specific angle.

If the setting value of the parameter is not 0, the system does not detect the lines of the encoder, but directly executes the commands following G99.

The lines that the system detects is four times that of the spindle encoder.

E.g. If the lines of the encoder installed is 1200, the lines that the system detects changes between 0~4800 recurrently. This parameter should be set between 0~4800 as well, or the system keeps waiting because G99 command cannot be detected.

**【Delay time when positioning converts to cutting (ms)】 \_\_Motion parameter P119**

If the last command is a rapid positioning command, and its following one is a cutting command, the delay set in P119 is automatically inserted between these two commands;

This parameter is used for preventing the taper streak on the workpiece due to the excessively fast command transition. If the parameter is too large, the machining efficiency may be affected. In general, set it between 0~100.

Example 1:

G00 U-50 ; Rapid positioning

G01 W-10 F100 ; The following command (this one ) is a cutting command. Before executing this command, delay P119 is inserted; to prevent taper streak generating on the front end.

Example 2:

G00 U-50 ; Rapid positioning

S1000

G01 W-10 F100 ; Since this command does not follow the rapid positioning command closely (another command separates them ), no delay is inserted during execution.

#### 4.6.4.4 Parameters of Drive and Compensation P200~P209, P411, P1000~P1905

**【Command pulse multiplier of Z/X/Y】 \_\_Drive parameters P203, P205, P207**

**【Command pulse division coefficient of Z/X/Y】 \_\_Drive parameters P204, P206, P208**

**P203** — Z multiplier coefficient: Numerator of Z axis electronic gear. (Range: 1~99999)

**P204** — Z division coefficient: Denominator of Z axis electronic gear. (Range: 1~99999)

**P205** —X multiplier coefficient: Numerator of X axis electronic gear. (Range: 1~99999)

**P206** — X division coefficient: Denominator of X axis electronic gear. (Range: 1~99999)

**P207** — Y multiplier coefficient: Nominator of Y axis electronic gear. (Range: 1~99999)

**P208** —Y division coefficient: Denominator of Y axis electronic gear. (Range: 1~99999)

**【Note】**

- 1) When setting parameters of drive, the ratio of the multiplier coefficient to the division coefficient should be limited between 1/128 ~128, or the system issues an alarm.
- 2) When the ratio of multiplier coefficient to division coefficient is 1:1:
  - Z/Y axis: The system outputs one pulse each time 0.001mm is moved;
  - X axis: In system radius programming, each time 0.001mm is moved, two pulses are output. (In system diameter programming, each time 0.001 mm is moved, one pulse is output.)
- 3) When the ratio of the multiplier coefficient to division coefficient is not 1:1:
  - Z/Y axis: Each time 0.001 mm is moved, the pulse that the system outputs is: multiplier coefficient/division coefficient
  - X axis: In system radius programming, each time 0.001 mm is moved, the pulse output is: multiplier coefficient \* 2/division coefficient. (In system diameter programming, each time 0.001 mm is moved, the pulse output is: multiplier coefficient/division coefficient. )
- 4) The maximum pulse output frequency of the system is 511pps/ms. This value cannot be exceeded anytime, or an alarm occurs during movement. I.e. System parameter P100 (maximum traverse speed) × multiplier coefficient/division coefficient should not be greater than 30000mm/min.

**【Spindle encoder lines】 \_\_Drive parameter P209**

This parameter determines lines with settable range between 100~5000 for the spindle encoder. When the spindle is ON and the diagnosis operation mode is entered, if the set value is inconsistent with the lines of the spindle encoder, the diagnosis check prompt : **Encoder lines inconsistent with parameter** is displayed in the diagnosis mode. Unit: line/rev.

**【Backlash value for Z/X/Y】 \_\_ Drive parameters P200, P201, P202**

These parameters respectively determine the backlash values of mechanical drive for Z, X, Y axes. Unit: mm.

Because gaps exist in the driving parts such as a screw and speed reducer on the machine tool, errors occur when the tool post moves back and forth. To compensate the errors caused by the gaps, Parameters P200, P201, P202 are provided. By setting these three parameters, the system automatically compensates the gap errors when the machine changes its direction during movement.

The mechanical drive gap of the machine can be measured with following method (an example for Z axis)

- 1) Select MANUAL operation mode and proper feedrate.
- 2) Mount the dial-indicator to a proper position on the machine, move the tool post till it touches the measuring head, and then set the pointer of the dial-indicator to 0.

- 3) Select manual STEP operation mode, with a step width of 1.0mm.
- 4) Press Z axis coordinate axis move key to move the tool post towards the dial-indicator, then make the pointer of the dial-indicator rotate one revolution and point at 0.
- 5) Press Z axis coordinate axis move key to move in the reverse direction, the dial-indicator rotates back because the pointer cannot return to 0 position due to the gap. Here, the difference between the position where the pointer points and 0 position is the backlash value of Z axis.

**【Note】**

- 1) It is necessary to repeat the above steps several times to measure the gap correctly.
- 2) The measuring method for X axis is the same, but the measured value should be multiplied by 2 to convert to the diameter amount.
- 3) The backlash compensation speed of Z, X, Y axis is set in **P411\_d1**.

**Note:** For **GSK928TD**, pitch error compensation function and tool nose radius compensation function are not provided.

**【Precision compensation】 \_\_Bit parameter P411 (password level: 2)**

d7	d6	d5	d4	Reserved	Reserved	d1	Reserved
----	----	----	----	----------	----------	----	----------

**d7\_\_Ptich error compensation function**

- 0: Disabled.** // The pitch error compensation function is disabled.
- 1: Enabled.** // The pitch error compensation function is enabled.

**d6\_\_ Representation of pitch error compensation value**

- 0: Unequal space.** // also referred to as corner description method. 150 groups of corner information can be input for each axis.
- 1: Equal space.** // By using equal-space description method, 300 compensation points can be input for each axis; the interval between points is the same.

Since the compensation data formats for corner description method and equal-space description method are different, the system prompts whether to confirm the modification when modifying this bit parameter. If the modification is confirmed, the original compensation data will be cleared, and the user needs to re-input the data in terms of the new format.

**d5\_\_ Tool nose radius compensation function**

- 0: Disabled.** // Tool radius compensation commands G41, G42 are disabled.
- 1: Enabled.** // Tool radius compensation commands G41, G42 are enabled.

**d4\_\_ Tool radius compensation mode**

- 0: Line transition.** // The system processes the transition mode for the sharp corner as the line transition in executing tool nose radius compensation.
- 1: Arc transition** // The system processes the transition mode for sharp corner as the arc transition in executing tool nose radius compensation.

**d1\_\_Backlash compensation mode**

- 0: Low speed.** // Z/X/Y axis performs backlash compensation at the low speed, which is set by P103, P104, P105.
- 1: High speed.** // Z/X/Y axis performs backlash compensation at the high speed, which is set by P100, P101, P102.



**【F—Pitch error compensation parameter】 \_\_Pitch error compensation parameters P1000~P1905**

The pitch error compensation parameters can be divided into unequal-space representation and equal-space representation, based on the setting of bit parameter P411\_d6.

**4.6.4.5 Parameters of Spindle and Coolant \_\_ P300~P317, P326, P329, P341, P410**

**【Spindle configuration】 \_\_Bit parameter P410 (password level: 1)**

d7	d6	d5	d4	d3	d2	Reserved	Reserved
----	----	----	----	----	----	----------	----------

**d7\_\_ Spindle control output**

- 0: Level.** // The spindle (M03/M04/M05) and coolant (M08/M09) are controlled by system level output.
- 1: Pulse.** // The spindle (M03/M04/M05) and coolant (M08/M09) are controlled by system pulse output. P326 is the pulse width.

**d6\_\_ Spindle S control**

- 0: Gear shift.** // The spindle is the gear-shift spindle, using S commands to shift gears; M41~M44 are forbidden.
- 1: Frequency conversion.** // The spindle is the frequency-conversion spindle, the speed of which is commanded by S, with gears M41~M44.

//For details, refer to Section 4.4.3.2 Spindle S Command -- Gear Shift Control and Section 4.4.3.3 Spindle S command – Speed Control in Part I Operation

**d5\_\_ S Gear shift output**

- 0: Direct.** // The control signals for the spindle gear are direct outputs S01~S04. It is forbidden to use S05~S15 commands.
- 1: Coded.** // The control signals of the spindle gear are coded outputs S00~S15.  
 // This parameter is used together with P310 (Occupied lines of the spindle gear control ). If it is set to direct output, each control line controls one gear, and the total number of gears is the same as that in P310; If it is set to coded output, it performs coded output based on the number of control lines in use. The total number of control gears is  $2^{P310}$  ; up to 4-channel control line output can be set as gear control signal.

**d4\_\_ Switching between spindle and Y axis**

- 0: Not performed.** //The spindle cannot operate in position control mode. It is forbidden to use M47/M48 command.
- 1: Performed.** // 928TD does not support this function.

**d3\_\_ Spindle brake signal output**

- 0: Used.** // When executing M5, the brake signal MSP is output.
- 1: Not used.** //When executing M5, the brake signal MSP is not output. The MSP signal interface can be used for other interface controls.

**d2\_\_ Spindle CCW signal output**

- 0: Used.** // To output spindle CCW signal M04  
**1: Not used.** // Not to output spindle CCW signal M04

**【Maximum speed of gear M41, M42, M43, M44】 \_\_ Auxiliary parameter P300, P301, P302, P303**

This parameter is the maximum speed of gear M41, M42, M43, M44. When the system employs a frequency converter to control the spindle, and the spindle gear is in gear M41, M42, M43 or M44, it is the maximum speed corresponding to the machine when the system outputs 10V analog voltage. When the spindle is controlled by a spindle multi-gear switch value, parameters P300, P301, P302 and P303 are disabled. (Unit: r/min)

**【Spindle minimum speed in constant surface speed】 \_\_ Auxiliary parameter P304**

This parameter limits the minimum speed in the constant surface speed cutting. According to the computational formula for the constant surface speed cutting:

$$\text{Surface speed} = \text{Spindle speed} * |X| * \pi / 1000 \quad (\text{Unit for X: mm, unit for spindle speed: r/min})$$

The X in the above formula means the absolute value (diameter value) of the X-axis tool nose coordinate. When X is smaller than a certain value, the spindle speed calculated is smaller than the value set in P304. In such a case, the spindle speed is clamped to the speed set in P304.

**【Spindle maximum speed in constant surface speed】 \_\_ Auxiliary parameter P305**

This parameter limits the maximum speed in the constant surface speed cutting. According to the computation formula for the constant surface speed cutting:

$$\text{Surface speed} = \text{Spindle speed} * |X| * \pi / 1000 \quad (\text{Unit for X: mm, unit for spindle speed: r/min})$$

The X in the above formula means the absolute value (diameter value) of the X-axis tool nose coordinate. When X is greater than a certain value, the spindle speed calculated is greater than the value set in P304. In such a case, the spindle speed is clamped to the speed set in P305.

**【Thread smooth speed boundary】 \_\_ Auxiliary parameter P306**

During thread machining, the system has two modes: One is the high-speed machining, the other is the low-speed machining. Parameter **P306** is the boundary between the two machining speeds (unit: r/min ). In general, P306 should be greater than 100.

If P306 is set to 300, make a judgment on the mode to be used based on the currently-detected actual spindle speed before the system machines the thread. If the actual speed is greater than 300, use the high-speed machining mode. Otherwise, use the low-speed machining mode.

When using the high-speed machining mode, the system following performance for the spindle speed fluctuation is the highest, but the motor running stability may be reduced; When the spindle speed fluctuation is intensive, there may be vibrating streak generated on the machining surface, or step-out occurs on the stepper motor.

When using the low-speed machining mode, the system following performance for the spindle speed fluctuation is slightly lower, but the motor running stability may be enhanced; When the spindle speed fluctuation is intensive, the vibrating streak is not obvious and the stepper motor is not easy to be out of step.

**【Thread spindle fluctuation alarm】 \_\_Auxiliary parameter P307**

During thread cutting, if the spindle speed fluctuation exceeds the value set in **P307**, the system issues an alarm after the current thread block machining is completed, and suspends the workpiece machining; by pressing **CYCLE START** key, the user can proceed to the next step of the machining program.

This parameter is valid for G33, G92, G32 command. (Unit: r/min)

**【Spindle JOG time (ms)】 \_\_ Auxiliary parameter P308**

In spindle JOG mode, it is the time for starting the spindle, then the spindle stops automatically after the time is up. When parameter P308 is 0, the spindle JOG function is disabled. (Unit: ms)

**【Speed in spindle JOG】 \_\_ Auxiliary parameter P309**

When the spindle is in the JOG state in MANUAL mode, it is the speed after spindle start key is pressed. (Unit: r/min)

If P309=0, the JOG output speed is the same as the speed at which M03/M04 is executed.

**【Occupied lines of spindle gear control】 \_\_Auxiliary parameter P310**

The parameter limits the number of output control lines used for spindle gear control. Up to 4 control lines can be used. When the control lines are less than 4, only the low-digit control lines can be used, and the unused high-digit control lines can be used for other purposes.

When the occupied number of lines =0, no output is available.

When the occupied number of lines =1, only S01 output is available.

When the occupied number of lines =2, only S02 and S01 outputs are available.

When the occupied number of lines =3, only S03, S02 and S01 outputs are available.

When the occupied number of lines =4, S04, S03, S02, S01 outputs are available.

**【Frequency-conversion spindle gear shift time 1, 2 (ms)】 \_\_ Auxiliary parameters P311, P312;**

The spindle gear shift time for M41~M44 when the spindle is the frequency-conversion spindle. (Unit: ms). For details, refer to Section 4.4 Manual Operation Mode in Part I Operation.

**【Interval time for spindle gear shift (ms)】 \_\_ Auxiliary parameter P313**

This parameter determines the time interval from cancelling the original gear signal to outputting the new gear signal. (Unit: ms)

**【Output voltage during spindle gear shift (mV)】 \_\_ Auxiliary parameter P314**

It is the output voltage during spindle gear shift. (Unit: mV)

**【Delay time from spindle stop to spindle brake (ms)】 \_\_ Auxiliary parameter P315**

This parameter determines the delay time from sending the spindle stop signal to sending the spindle brake signal. (Unit: ms)

**【Spindle brake output time (ms)】 \_\_Auxiliary parameter P316**

This parameter means the brake signal output time. In general, it is set to the time required for the spindle motor to stop completely from the braking. (Unit: ms)

**【Spindle maximum speed limit】 \_\_Auxiliary parameter P317**

This parameter limits the spindle maximum speed. (Unit: r/min)

If the S speed set by the program command is greater than the value given by P317, the spindle speed is clamped to the one set in P317.

**【Spindle control pulse time (ms)】 \_\_Auxiliary parameter P326**

This parameter determines the hold time of the pulse signal when the spindle (M03/M04/M05), coolant (M08/M09) output is the pulse control mode. (Unit: ms)

**【M code response detection time (ms)】 \_\_ Auxiliary parameter P329**

This parameter determines the limit for M code response detection time. If the M code response signal is not detected within this period of time, the system prompts the corresponding alarm message. (Unit: ms). E.g. When executing M10 in MANUAL operation mode, if the chuck clamping response signal is not detected after waiting for the period of time set in P329, the system issues: **Chuck clamping response detection overtime alarm**.

**【Cutting enabled when rotation speed reaches certain percentage】 \_\_Auxiliary parameter P341**

If the cutting is started immediately after the spindle is started or the gear is shifted, the tool life and workpiece will be affected because the spindle has not reached the preset speed.

This parameter is used for the automatic identification of the spindle speed arrival. E.g. P341=80 means the cutting is allowed when the spindle actual speed is within the speed range (100%±20%) set by the program, i.e. within the range between 80%~120% ; If the programming is S2000, the cutting is allowed when the actual speed detected by the system is within the range of 1600~2400. Otherwise, the system has to wait.

When P341=0, this function is disabled. When P341 is not set to 0, the system automatically detects the actual speed in MANUAL or Auto mode, with the state of the spindle actual speed being displayed on the screen; if the spindle speed Sxxxx is displayed in yellow, it means the preset range has not been reached; if it is displayed in green, it means the preset range has been reached.

Use: When using either a frequency-conversion spindle or gear-shift spindle, this parameter is valid.

When using the frequency-conversion spindle, the system makes a judgment by using the programmed speed S and actual speed.

When using the gear-shift spindle, the spindle speeds of gears S1~S4 respectively correspond to the set values of parameters P300~P303; e.g. when the gear of the gear-shift spindle is S1, the system makes a judgment by using parameter P300 and the actual speed. P341 is invalid for other gears, so they are not restricted by P341 during cutting.

This parameter is valid only for the cutting commands, e.g. G01, G02, G33; and it is also valid for the cutting in compound commands. When cutting commands are executed continuously, only the first cutting command is detected. If the spindle speed is displayed in yellow, and the cutting command does not move, it means the system is waiting for the spindle to reach the preset speed.

#### 4.6.4.6 Parameters of Tool Post \_\_ P318~P325, P408

**【Tool post setting】 \_\_Bit parameter P408 (password level: 2)**

d7	d6	d5	d4	d3	d2	Reserved	Reserved
----	----	----	----	----	----	----------	----------

**d7\_\_ Tool number signal detection mode**

**0: Default** // The tool number detection method is preset inside the system.

**1: Table checking** // For some special tool posts, it is necessary to use the table checking method, which requires to set the tool number detection signals in parameters P541~P556 . For details, refer to Section 3.4.4 in Part III Connection.

**d6\_\_ Tool post clamping in-position signal**

**0: Without.** // Without the tool post clamping in-position signal (input interface 9 can be released).

**1: With.** // With the tool post clamping in-position signal (input interface 9 is occupied).

**d5\_\_ Tool post clamping signal level**

**0: Low** // Tool post clamping signal low level is valid.

**1: High** // Tool post clamping signal high level is valid.

**d4\_\_ Tool post overheat detection level (this parameter is invalid)**

**0: Low** // Tool post temperature control switch (TGR signal) low level is valid.

**1: High** // Tool post temperature control switch (TGR signal) high level is valid

**d3\_\_ Strobe signal level**

**0: Low** // Tool post encode strobe signal low level is valid.

**1: High** // Tool post encoder strobe signal high level is valid.

**d2\_\_ Pre-indexing in-position detection level**

**0: Low** // Tool post pre-indexing switch low level is valid.

**1: High** // Tool post pre-indexing switch high level is valid.

**【Tool post type】 \_\_ Auxiliary parameter P318 (maximum value: 9)**

This parameter is determined by the tool post type installed on the machine. The system performs the tool change mode according to this parameter.

**P318=0**, the machine installs the line-up tool post, which does not occupy any input/output interface of the system.

**P318=1 or 2**, the machine installs the common electric rotation tool post; performs tool change mode 1 or mode 2.

**P318=3**, it is applicable for the 8-station tool posts of Taiwan Genyang machine tool CLT-63~CLT300 series; performs tool change mode 3

**P318=4**, it is applicable to the tool post type which can change the tool nearby in the forward/reverse direction; performs tool change mode 4.

**P318=9**, use M60 to self-define a command to change the tool. See the example in Section 3.4 “Tool Change Control Function and Connection” in Part III Connection.

If **P318** is not the numbers described above, the system performs tool change in terms of tool change mode 1.

**【Maximum tool number】 \_\_Auxiliary parameter P319**

This parameter determines the maximum tool number on the electric tool post of the machine. The

standard configuration of the system is 8-station electric tool post. The tool number signal can be extended to 12~16 stations electric tool post according to the specific coded input. See Section 3.4 “Tool Change Control Function and Connection” in Part III Connection

**【Number of lines occupied by tool number signal】 \_\_ Auxiliary parameter P320**

This parameter determines the number of system hardware interfaces occupied by the tool number signal. The maximum number is 8, i.e., up to 8 interfaces can be occupied. In general, it is set in combination with the maximum tool number. The interface occupation can be displayed in the Diagnosis operation mode.

**【Tool change time T1 T2 T3 (ms)】 \_\_ Auxiliary parameter P321, P322, P323**

T1\_ Delay time from “stopping CW rotation” to “starting tool post CCW rotation” after tool post CW rotation detects the tool number signal during tool change. (Unit: ms)

T2\_ During the tool pose CCW rotation, if it is necessary to detect the tool post clamp in-position signal TCP, P323 is the delay time from “after the tool number clamp signal is detected” to “when the tool post CCW signal stops” .

T3\_ Reserved. (Unit: ms)

**【Tool post CCW rotation clamping time (ms)】 \_\_ Auxiliary parameter P324**

During tool change of the electric tool post, this parameter determines the duration of the tool post CCW signal if it is not necessary to detect the tool post clamping in-position signal TCP; and the tool post maximum CCW clamping time if it is necessary to detect TCP. (Unit: ms)

**Note: The value set in P324 should be adjusted properly depending on the actual electric tool post. An excessively large value may cause overheat or even damage to the motor. An excessively small value may not clamp the tool post firmly. Therefore, the user should use different values for debugging and then choose a proper one during debugging.**

**【Tool change rotation time limit (ms)】 \_\_ Auxiliary parameter P325**

Maximum duration from “when the tool post CW rotation starts” to “when tool number signal of the target tool is detected”. (Unit: ms)

The below parameters are the dedicated interface parameters for tool change mode 3. Refer to Section 3.4.3.4 Tool Change Mode 3 in Part III Connection.

**【Tool post brake output: TZD】 \_\_ Interface parameter P507**

This parameter controls the brake device of the tool post motor.

**【Tool post pre-indexing output: TFD】 \_\_ Interface parameter P508**

This parameter controls the pre-indexing proximity switch.

**【Pre-indexing in-position detection: TFDC】 \_\_ Interface parameter P528**

This parameter detects the in-position signal of the pre-indexing proximity switch.

**【Tool number strobe signal: TXT】 \_\_ Interface parameter P529**

This parameter is the strobe signal for tool post encoder. It controls the tool number output together with the binary absolute value of the encoder. Up to 12 positions can be controlled.

**【Tool post overheat detection: TGR】 \_\_Interface parameter P530 (this parameter is invalid)**

This parameter detects the temperature control signal of the tool post. When the temperature is higher than 120°C, the temperature control switch is turned off.

**4.6.4.7 Parameters of Chuck and Tailstock \_\_ P327~P328, P409**

**【Chuck control pulse time (ms)】 \_\_Auxiliary parameter P327**

If the chuck is in the pulse control mode, this parameter determines the duration of the pulse signal output by executing the chuck command (M10/M11). (Unit: ms)

**【Tailstock control pulse time (ms)】 \_\_Auxiliary parameter P328**

If the tailstock is in the pulse control mode, this parameter determines the duration of the pulse signal output by executing tailstock command (M78/M79). (Unit: ms)

**【Chuck tailstock】 \_\_Bit parameter P409 (password level: 1)**

d7	d6	d5	d4	d3	d2	d1	d0
----	----	----	----	----	----	----	----

**d7\_\_ Chuck control function**

- 0: Enabled. // The system has the hydraulic chuck control function.
- 1: Disabled. // The chuck control function is disabled. It is forbidden to use M10/M11 command.

**d6\_\_ Chuck clamping mode**

- 0: Outer chuck. // The hydraulic chuck is the outer chuck mode.
- 1: Inner chuck. // The hydraulic chuck is the inner mode. For the execution of M10/ M11 command, refer to Section 4.4 “Manual Operation Mode” in Part I Operation.

**d5\_\_ Chuck response detection**

- 0: No performed.
- 1: Performed. // If the response detection is required, input interfaces RM10 and RM11 are respectively used as clamping and release in-position signal inputs

**d4\_\_ Tailstock control function**

- 0: Enabled. // The system has the hydraulic tailstock control function.
- 1: Disabled. // The system disables the M78/M79 command, so it has no hydraulic tailstock control function.

**d3\_\_ Chuck control output**

- 0: Level. // Level mode control M10, M11 signal output
- 1: Pulse. // In pulse control mode, the output holding time of signals M10, M11 is set by parameter P327.

**d2\_\_ Tailstock control output**

- 0: Level. // Level mode control M78, M79 signal output.
- 1: Pulse. // Pulse control mode, the output holding time of signals M78, M79 is set by parameter P328.

**d1\_\_ Hydraulic chuck foot switch input**

- 0: Required. // In addition to controlling the hydraulic chuck using a command, it is also available to control the hydraulic chuck by an external foot switch.

1: Not required.

#### d0\_\_Hydraulic tailstock foot switch input

0: Required. // In addition to controlling the hydraulic tailstock using a command, it is also available to control the hydraulic tailstock by an external foot switch.

1: Not required.

#### 4.6.4.8 Bit Parameters of Running and Efficiency\_\_ P400~P401

**【Running setting】 \_\_Bit parameter P400 (password level: 3)**

Reserved	d 6	d 5	d 4	d 3	d 2	d 1	Reserved
----------	-----	-----	-----	-----	-----	-----	----------

#### d6\_\_Spindle gear memory

0: Disabled. // Spindle gears M41~M44 are not memorized in MANUAL or AUTO state. The gear is M41 after the power is turned off and then back on.

1: Enabled. // Spindle gears M41~M44 are memorized in MANUAL or AUTO state. The gear after power on is the one before the last power-off.

#### d5\_\_Spindle in AUTO mode

0: Is controlled by keys. // In the state that no program is executed in AUTO mode, Spindle CW, Spindle CCW, Spindle Stop keys are enabled.

1: Is not controlled by keys. // In AUTO operation mode, Spindle CW, Spindle CCW, Spindle Stop keys are disabled.

#### d4\_\_MPG 0.1mm step width

0: Enabled. //In MANUAL operation mode, it is allowed to select a large override (×0.1mm gear) to rotate the MPG.

1: Disabled. // In MANUAL operation mode, it is not allowed to select a large override (×0.1mm gear) to rotate the MPG.

#### d3\_\_Rapid positioning mode

0: Independent traverse. // G00 performs rapid positioning respectively by the rapid traverse mode for each axis.

1: Interpolation traverse. // G00 performs rapid positioning simultaneously by the scaling mode.

#### d2\_\_Arc across vertex process mode

0: Precise. // Precisely processing the backlash in the arc across the vertex.

1: Smooth. // Smoothly processing the backlash in arc across the vertex.

#### d1\_\_MPG axis select

0: Selecting axis with keyboard. // The axis select key on the keyboard is enabled.

1: External axis select. // The external MPG control knob is enabled, and the axis select key on the keyboard is disabled.

// In MANUAL operation mode, after pressing MPG key to switch to MPG mode, the indicator lights up, and on the screen are displayed the states of the external MPG control knob, axis select key and movement amount knob. The coordinate axis selected is highlighted.



**【Efficiency setting】 \_\_Bit parameter P401 (password level: 3)**

d7	d6	d5	d4	d3	d2	d1	d0
----	----	----	----	----	----	----	----

**d7\_\_ TMS command in dry run**

**0: Executed.** // In the dry run state, when executing the auxiliary function command, the signals are output and detected as normal machining.

**1: Skipped.** // In the dry run state, when executing the auxiliary function command, the signals are not output and detected.

**d6\_\_ Feed command in dry run**

**0: Normal speed.** // In the dry run state, the executed speed of the feed command is set by the program, just as normal machining.

**1: Acceleration.** // In the dry run state, the executed speed of the feed command is out of the control of the program. The system demonstrates the program path at the maximum speed (P113) for the cutting feed.

**d5\_\_ Short linear high speed**

**0:** High speed connection. // Short linear connection transition uses the high-speed connection.

**1:** Not used. //Short linear connection transition does not use the high-speed connection.

// When **P401\_d5=0**, for the successive short line blocks (no empty line, notation line, or other block between blocks) with short movement distance and fast executed speed, the CNC realizes the optimal speed connection transition by adopting the advanced preview control mode. Up to 80 blocks can be pre-read.

**d4\_\_ Cutting command**

**0:** Continuous smooth transition.

**1:** Deceleration to 0.

// In G01, G02, G03 cutting feed, when bit parameter **P401\_d4=0**, the system employs post-acceleration/deceleration. At the intersection of the paths of the two adjacent cutting feed blocks, an arc transition is formed due to the acceleration/deceleration. In such a case, the intersection of the two paths cannot be precisely positioned, so there is an error between actual path and programmed path. To prevent such a contour error, insert a delay command (G04) between the two blocks or set the system bit parameter **P401\_d4** to 1. At this moment, the former block decelerates to 0 when it moves to the block end point, precisely positioning at the end point. Then the system proceeds to the next cutting feed block; in this mode, however, because each block starts to accelerate from 0 speed, and the next block is not executed until it decelerates to 0 at the end point. Thus, the program run time will increase, reducing the machining efficiency.

**d3\_\_ Execution sequence of multiple commands in the same block**

**0: step by step.**

**1: Simultaneously.** //It is referred to as multiple blocks in the same block when a block contains several commands. The commands in the same block are executed at the same time; after all the commands are executed, the next block is executed

// When **P401\_d3=1**, the execution is simultaneous execution; in this mode, the irrelative commands are executed almost at the same time, without following the execution sequence of step by step or waiting for each other. However, for those M codes with interlock relationship, the system automatically arranges the execution sequence according to the parameter.

// When **P401\_d3=0**, the execution is performed step by step; the non-simultaneous execution sequence is as follows: after a function command is executed, the next function command can be executed; till all of them are executed.

#### **d2\_\_ Rapid block skip execution function**

**0: OFF**

**1: ON.** // Under the premise that **P401\_d3=1**, **P401\_d2=1** is valid. When **P401\_d2=1**, the **rapid block skip execution is allowed**. In the mode, as long as the axis move commands in the current block are executed (without waiting for the completion of the execution of other MST commands), the system rapidly skips to the next block to execute.

#### **d1\_\_G92 Run-out**

**0: Optimized.** //When there is run-out in G92, the run-out axis directly retracts to the start point of G92 if there is only I; thus enhancing the efficiency and improving the run-out effect.

**1: Common.** // The run-out is performed according to the G92 run-out mode described in Section 4.13.2 in Part II Programming.

#### **d0\_\_Cutting efficiency**

**0: Normal state.** // The cutting speed is the normal value set by cutting feed command F.

**1: Optimized.** // The cutting efficiency is improved by 5% by optimizing the speed; which applies to the situation that F is not modified, but efficiency needs to be improved.

#### 4.6.4.9 Relationship between Path and Parameters of Running and Efficiency

1) When **P401\_d4=0**, it is the cutting command continuous smooth transition. The system acceleration/deceleration after interpolation is valid:

##### **Processing of G01:**

When **P401\_d5=0**, G01 command has the acceleration/deceleration function before interpolation, and the connection transition between G01 and G02 is performed at the optimal speed; then the system performs the acceleration/deceleration processing after interpolation; When the cutting speed is high, the line segment path is accurate, and the transition point is smooth. In this case, the acceleration/deceleration function after interpolation does not have much effect on the path precision.

When **P401\_d5=1**, G01 command does not employ high-speed connection. It directly performs the acceleration/deceleration after interpolation instead of acceleration/deceleration before interpolation.

##### **Processing of an arc**

When **P400\_d2=0**, it is the path precise processing when the arc crosses the vertex. The arc command has the acceleration/deceleration function before interpolation. The speed decelerates to

the start speed when the quadrant is crossed, then the backlash is performed, and the machining path transits to the next quadrant, then accelerates, and then decelerates before the end. Finally the acceleration/deceleration after interpolation is performed.

When **P400\_ d2=1**, it is the smooth processing when the arc passes the vertex; the arc command directly performs the acceleration/deceleration after interpolation without performing the acceleration/deceleration before interpolation.

2) When **P401\_ d4=1**, the cutting command decelerates to 0; the deceleration function after interpolation is invalid:

**Processing of G01:**

When **P401\_ d5=0**, G01 command has the acceleration/deceleration function before interpolation, and the connection transition between G01 and G02 is performed at the optimal speed; all the G01 commands decelerate to 0 speed when moving to the end point of the block, and are positioned at the end point precisely; when the cutting speed is high, the line segment path is accurate, and the transition point is smooth.

When **P401\_ d5=1**, G01 command has the acceleration/deceleration function before interpolation. Each cutting command decelerates to 0 after it is executed.

**Processing of an arc**

When **P400\_ d2=0**, it is the path precise processing when the arc passes the vertex. The arc command has the acceleration/deceleration function before interpolation. The speed decelerates to the start speed when the quadrant is passed, then the backlash is performed, and the machining path transits to the next quadrant, then accelerates, and then decelerates before the end.

When **P400\_ d2=1**, it is the smooth processing when the arc passes the convex; the arc command has the acceleration/deceleration before interpolation. If the backlash is 0, the deceleration is not performed when the quadrant is passed.

**4.6.4.10 Bit Parameters of Safety and Debugging\_\_ P402~P404, P419**

**【Safety setting 1】\_ Bit parameter P402 (password level: 3)**

d7	d6	d5	d4	d3	d2	d1	d0
----	----	----	----	----	----	----	----

**d7\_\_ Hard limit alarm**

// This parameter specifies the processing mode when the hard limit alarm signal is detected during axis movement.

**0: Deceleration.** // When it set to 0, the move axis decelerates to stop if the hard limit alarm occurs. The coordinate is the same as the actual position.

**1: Emergency stop.** // When it is set to 1, the move axis immediately stops after the hard limit alarm occurs. The coordinate may not be the same as the actual position.

**d6\_\_ Spindle abnormal stop detection**

**0: Performed.** // If the spindle stops during cutting, the system stops the feed and turns off the spindle, with an alarm being issued.

**1: Not performed.**

// In MANUAL/AUTO operation mode, after the spindle is started, the system automatically detects the spindle speed. If the spindle is not started normally, the system prompts "Spindle rotation abnormality". If the spindle stops rotating by accident during cutting feed, the system will stop the feed, suspend the program execution and turn off the spindle, issuing an alarm.

// When the spindle is required to work at a very low speed (lower than 1 r/min), the system may issue an alarm by mistake. In this case, the user should set the parameter to 1. For the frequency-conversion spindle, when the programmed speed is lower than S0, the system will not perform detection or give an alarm.

**d5\_\_ Interlock between chuck and spindle**

**0: Is performed.** // When it interlocks with the spindle, the chuck cannot be controlled before the spindle stops; the spindle cannot be started neither when the chuck is not clamped tightly.

**1: Is released.** // When the interlock between chuck and spindle is released, neither the spindle starting is affected by the state of the chuck, nor the chuck control is affected by the state of the spindle.

**d4\_\_ Chuck response signal consecutive detection alarm is**

**0: Not issued.**

**1: Issued.** // This bit indicates whether to detect the state of the chuck in real time. Anytime the chuck is released, an alarm is issued. This control bit is valid only when there is a response signal for the chuck (P409\_ d5=1).

**d3\_\_ Interlock between tailstock and spindle**

**0: Is performed.** // When the tailstock interlocks with the spindle, the tailstock cannot be controlled before the spindle stops; the spindle cannot be started neither when the tailstock is not clamped tightly.

**1: Is released.** // When the interlock between tailstock and spindle is released, neither the spindle starting is affected by the state of the tailstock, nor the tailstock control is affected by the state of the spindle.

**d2\_\_ Tailstock response signal consecutive detection alarm is**

**0. Not issued.**

**1: Issued** // This bit indicates whether to detect the state of the tailstock in real time. Anytime the tailstock is released, an alarm is issued. This control bit is valid only when there is a response signal for the tailstock (setting parameter P519, P520).

**d1\_\_ Auto MPG control**

**0: Disabled.** // In AUTO operation mode, using MPG to control the rapid/feed override is invalid.

**1: Enabled.** // In AUTO operation mode, using MPG to control the rapid/feed override is valid.

**d0\_\_ Auto MPG mode**

**0: Mode 1.** // In AUTO operation mode, use mode 1 to control MPG rapid/feed override.

**1: Mode 2.** // In AUTO operation mode, use mode 2 to control MPG rapid/feed override. For details about mode 1 and mode 2, please refer to Section 4.5 Auto Operation Mode

in Part I Operation.

**【Safety setting 2】 \_\_Bit parameter P403 (password level: 3)**

d7	d6	d5	d4	d3	d2	d1	d0
----	----	----	----	----	----	----	----

**d7\_\_ Tool change operation**

- 0: Allowed.** // Tool change is allowed in MANUAL and AUTO states.
- 1: Forbidden.** // Tool change is forbidden in MANUAL and AUTO states.

**d6\_\_ Tool Offset operation mode is**

- 0: Coordinate modification.** // The tool offset operation mode in MANUAL state is machine coordinate modification.
- 1: Tool post movement.** // Tool offset operation mode in MANUAL state is tool post movement.

**d5\_\_ Manual tool change**

- 0: Confirmed by a key.** // During tool change in MANUAL state, the system prompts **Confirm tool change?**, then you can execute tool change by pressing **ENTER** key.
- 1: Performed immediately.** // Tool change is performed immediately in MANUAL state.

**d4\_\_ Tool setting record check**

- 0: OFF.** // The program compound check prompt alarm caused by inconsistency with the tool setting record is not displayed.
- 1: ON.** // The tool setting record has been set in the tool offset number. When the other tool numbers still use this tool offset number during programming, the system displays [Program compound check prompt alarm]. E.g. **T0205 is inconsistent with the tool setting record T0105.**

**d3\_\_ M output during emergency stop**

- 0: Only the spindle and coolant are turned off.** // When the emergency stop alarm is valid, the system only turns off the output signals of the spindle, coolant and lubricant.
- 1: All are turned off.** //When the emergency stop alarm is valid, the system turns off all the output signals of the auxiliary functions.

**d2\_\_ By pressing Reset key, M function output is**

- 0: Turned off.** // After pressing Reset key, the system turns off the output signals of M03, M04, M08, M32.
- 1: Not turned off.** // After pressing Reset key, the system does not turn off the output signals of M03, M04, M08, M32.

**d1\_\_ During M function alarm**

- 0: The program is terminated.** // The program is terminated during M10 M11 M78 M79 Txx function alarm.
- 1: Selection is prompted.** // The system prompts whether to retry during M10 M11 M78 M79 Txx function alarm.

// When the conditions that **P401\_d3=0** (execution step by step in a block with multiple commands) and **P403\_d1=1** (Prompt in M function alarm) are satisfied simultaneously, the retry function is valid.

Otherwise, the system issues an alarm and terminates the machining program when the relative commands cannot be completed.

In the process of automatic machining, the commands that can be retried are: T commands, chuck (M10/M11) control and tailstock (M78/M79) control.

When executing these commands, if the corresponding valid input signal is not detected, the commands cannot be completed within a specified period of time, then the system prompts whether to retry. After the retry message appears, the system is at the pause state; at this moment, the operator can check relative input signals and remove the fault in terms of the prompted message.

After the fault is removed, press **R** key to re-execute the command which caused the fault. After the retry operation is correctly completed, the system is at the pause state, and the user can continue the machining by pressing **CYCLE START** key. If the retry execution is still incorrect, you can perform the retry operation several times; if the command cannot be executed correctly after multiple retry operations, you can exit the machining program by pressing **ESC** key.

**d0\_\_ Waiting for the speed to be stable before thread machining is**

**0: Not performed.** // Whether the spindle speed is stable is not detected before thread machining.

**1: Performed.** // Whether the spindle speed is stable is detected by the system automatically before thread machining. The system machines threads after the speed is stable.

// This parameter is valid for G33, G92, G32 commands.

// The system detects the spindle speed in an instant. If it detects the spindle is in the progress of acceleration or deceleration, it does not machine the threads until the progress is finished. If the thread command is executed immediately after the speed is changed, the function of detecting whether the spindle speed is stable may be invalid.

E.g., in S1000 stable state, when executing S200, if G33 command is executed immediately, the system will immediately execute the program for detecting whether the speed is stable. In this case, the system misjudges the speed is stable in S1000 stable state; because there is a time difference between system command and spindle speed actual change. To prevent the misjudgment, specify a delay of 0.3s between S200 and G33.

When the spindle is required to work at a very low speed (lower than 10 r/min) to cut threads, the system may remain in the state of detecting whether the spindle speed is stable for a long time; in this case, set this parameter to 0 to cancel this function.

**【Safety setting 3】 \_\_Bit parameter P419 (password level: 3)**

d7	d6	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
----	----	----------	----------	----------	----------	----------	----------

**d7\_\_ Turning off spindle during driver alarm is**

**0: Not performed.** // When the system detects the driver unit alarm, it does not turn off the spindle.

**1: Performed.** // When the system detects the drive unit alarm, it turns off the spindle.

**d6\_\_ M state change during machining is**

**0: Not prompted.** // The system does prompt the state change of the spindle, chuck and tailstock.

**1: Prompted.** // The system displays a window to prompt the state change of the spindle, chuck and tailstock.

// In AUTO machining, it is require to pause sometimes, e.g., after the pause, turn off the spindle for observation, then turn on the spindle, and press **CYCLE START** key to continue the machining. If the execution is continued without turning off the spindle, unexpected accidents may occur. Setting bit parameter P419\_d6 to 1 can provide protection to some extent. When P419\_d6=1, at the beginning of the single block or pause, the system needs to memorize the states of the spindle (M3/M4/M5), chuck(M10/M11) and tailstock (M78/M79). If some operations are performed afterwards, the system first judges whether the state is consistent with the previous one after **CYCLE START** key is pressed. If the states are inconsistent with each other, a window pops up to prompt the changed contents. In such a case, you can recover the state by another operation, or press **Y** key to continue the execution; after **N** key is pressed to exit, the window displaying prompts for M3/M4/M5, M10/M11, M78/M79 disappears, and the system returns to the pause state.

**【Debugging setting】 \_\_Bit parameter P404 (password level: 2)**

d7	d6	d5	d4	d3	d2	d1	d0
----	----	----	----	----	----	----	----

This parameter is set for the convenience of the system debugging. In the online state, it must be set to the valid state. Otherwise, it cannot provide protection.

**d7\_\_Emergency stop alarm is**

**0: Detected.** // The external emergency stop signal function is valid.

**1: Shielded.** // The external emergency stop signal function is invalid.

**d6\_\_Hard limit alarm**

**0: Detected.** // The hard limit alarm function is valid.

**1: Shielded.** // The hard limit alarm function is invalid.

**d5\_\_Drive unit alarm**

**0: Detected.** // The drive unit alarm function is valid.

**1: Shielded.** // The drive unit alarm function is invalid.

**d4\_\_Mechanical soft limit alarm**

**0: Valid.** // The mechanical soft limit alarm function is valid.

**1: Invalid.** // The mechanical soft limit alarm function is invalid.

**d3\_\_Tool nose soft limit alarm**

**0: Valid.** // The tool nose soft limit alarm function is valid.

**1: Invalid.** // Tool nose soft limit alarm function is invalid.

**d2\_\_254 Program limit**

**0: Forbidding move command.** // All G codes (other than G04) in M60~M74 blocks are forbidden.

**1: Allowed.** // M60~M74 are user-defined commands. When the program is written according to the program format, it is allowed that the blocks contain G move codes.

**d1\_\_Hard limit alarm level**

**0: Low.** // The hard limit alarm low level is valid.

**1: High.** // The hard limit alarm high level is valid.

**d0\_\_ No. 254 program solidification limit**

**0: Forbidden.** // It is forbidden to modify or re-solidify No. 254 program, i.e. it is forbidden to modify the self-defined command.

**1: Allowed.** // It is allowed to modify and re-solidify No. 254 program, i.e. it is allowed to modify the self-defined command.

Whether to modify and then re-solidify the solidified program is determined by the machine tool builder.

**4.6.4.11 Bit Parameter of Motor Driver\_\_ P405**

**【Motor driver】 \_\_Bit parameter P405 (password level: 2)**

d7	d6	d5	d4	d3	d2	d1	d0
----	----	----	----	----	----	----	----

This parameter sets the motor working state and whether there is the Y axis control.

**d7\_\_Z axis motor direction;**

**d6\_\_X axis motor direction;**

**d5\_\_Y axis motor direction;**

**0: Positive.**

**1: Negative.** // By setting the parameter for the motor direction, it is available to change the motor rotation direction without changing other external conditions. Make the tool post actual movement direction the same as the system-defined direction.

**d4\_\_Z axis drive unit alarm level;**

**d3\_\_X axis drive unit alarm level;**

**d2\_\_Y axis drive unit alarm level;**

**0: High.** //When the input signal of drive unit alarm for Z, X, Y axis is the high level, "Drive unit alarm" is issued.

**1: Low.** // When the input signal of drive unit alarm for Z, X, Y axis is the low level, "Drive unit alarm" is issued.

**d1\_\_Control axis**

**0: Without Y axis.** // It is forbidden to use Y axis move command.

**1: With Y axis.** // The control axis has Y axis, and Y axis move command is allowed. **Note: 928TD has no Y axis function.**

**d0\_\_Motor acceleration/deceleration control**

**0: Linear.** // The motor acceleration/deceleration adopts the linear control.

**1: Exponential.** //The motor acceleration/deceleration curve is an exponential curve or adjusted exponential curve.

**4.6.4.12 Parameters of Other Interfaces\_\_ P412, P330~P332**

**【relative Interface】 \_\_Bit parameter P412 (password level: 2)**



d7	d6	d5	d4	Reserved	d2	d1	d0
----	----	----	----	----------	----	----	----

**d7\_\_External start/pause signal**

**0: Disabled.** // The external start and pause signals are invalid.

**1: Enabled.** // The external start and pause signals are valid.

**d6\_\_External feed hold knob**

**0: Disabled.** // The system external interface for the feed/spindle hold knob is invalid.

**1: Enabled.** // The system external interface for the feed/spindle hold knob interface is valid.

**d5\_\_Function of detecting low pressure**

**0: Not provided.**

**1: Provided.** //The system is provided with the low pressure detection function; **P412\_d4** sets the alarm level.

**d4\_\_Low pressure alarm level**

**0: High.** // Disconnected with 0 V.

**1: Low.** // Connected with 0 V. After the system low pressure alarm detection function is selected, the **Low pressure alarm** is issued once the system detects the low pressure alarm signal PRES is valid and the signal hold time exceeds the set value of data parameter **P332**; at this moment, the axis feed pauses, the spindle stops, and the automatic cycle cannot start.

**d2\_\_Executing M73 in MANUAL/AUTO reset is**

**0: Not performed.** // In MANUAL/AUTO operation mode, by pressing the reset key, the system only executes the standard reset function.

**1: Performed.** // In MANUAL/AUTO operation mode, by pressing the reset key, the system executes the solidified M73 self-defined command after executing the standard reset function.

**d1\_\_Executing M74 in MANUAL/AUTO emergency stop**

**0: Not performed.** // In MANUAL/AUTO operation mode, the system only executes the standard emergency stop function when the emergency stop alarm is valid.

**1: Performed.** // In MANUAL/AUTO operation mode, the emergency stop alarm is valid. The system will execute the solidified M74 self-defined command after executing the standard emergency stop function.

**d0\_\_Macro programming for auxiliary function output pin**

**0: Disabled.** // It is forbidden to perform the statement programming to the defined dedicated pins of the input signals. It is only allowed to perform the statement programming to the pins which display “UO” on the diagnosis page. It is forbidden for M81, M82, M83 commands to program the defined dedicated pins of input signals, but to program the pins which display “UO” on the diagnosis page.

**1: Enabled.** // It is allowed to perform the statement programming to the pins of all input signals.

**【Automatic lubrication start time (s)】 \_\_Auxiliary parameter P330:**

This parameter determines the lubrication start time. (Unit: s)

**【Automatic lubrication interval time (s)】 \_\_Auxiliary parameter P331**

This parameter determines the automatic lubrication time interval. (Unit: s)

**【Hold time of low pressure alarm (s)】 \_\_Auxiliary parameter P332**

This parameter determines if the low pressure signal hold time exceeds the value set in P332, the system issues an alarm. (Unit: s)

## 4.6.4.13 Other Parameters\_\_ P413~P416, P333

**【Interface language】 \_\_Bit parameter P413 (password level: 2)**

d7	d6	d5	Reserved	d3	Reserved	Reserved	Reserved
----	----	----	----------	----	----------	----------	----------

**d7\_\_Interface language**

**0: Chinese.** // The interface is displayed in Chinese.

**1: English.** // The interface is displayed in English.

**d6\_\_System parameter and programming**

**0: Diameter.** //The command value of X axis in the program is input in diameter, and the coordinate of X axis is displayed in diameter.

**1: Radius.** //The command value of X axis in the program is input in radius, and the coordinate of X axis is displayed in radius.

// The coordinate value of X axis in the system can be expressed in radius and diameter. When modifying this bit parameter, the system will prompt whether to convert the parameter. If the conversion is confirmed, the system will convert the reference coordinate parameters P001, P004, P007, P011, P012, P017, P018, P022 and P025.

// After modifying this bit parameter, it is required to re-perform tool setting and programming. Otherwise, the data in X axis direction is incorrect.

**d5\_\_Coordinate system setting**

**0: Front tool post.** // The definition of the front tool post is shown in figure 1-1 in Chapter One in Part II Programming.

**1: Rear tool post.** // The definition of the rear tool post is shown in figure 1-2 in Chapter One in Part II Programming.

// This parameter is used only for displaying the icon of the imaginary tool nose number visually in manual tool setting; during manual tool setting, the system uses this parameter to judge whether the front tool post coordinate system or rear tool post coordinate system is used, in order to correctly display the position relationship between tool nose center and imaginary tool nose.

**// 928TD does not support the tool nose radius compensation function.**

**d3\_\_Lease command increment**

**0: Metric.** // The unit of the command value in the program is mm.

**1: Inch.** // The unit of the command value in the program is inch. (This option is invalid currently.)

**【Communication interface】 \_\_Bit parameter P414 (password level: 2)**

d7	d6	Reserved	Reserved	Reserved	Reserved	Reserved	d0
----	----	----------	----------	----------	----------	----------	----

**d7\_\_Baudrate select (valid when P414\_d6=0)**

0: 9600。

1: 19200。

**d6\_\_Baudrate select (prior)**

0: Decided by d7.

1: 38400。

Bit select	Baudrate
d7 =0, d6 =0	9600
d7 =1, d6 =0	19200
d7 =0, d6=1	38400
d7 =1, d6=1	

**d0\_\_Bit of debugging function**

0: Normal state. // The user generally sets it to the normal state.

1: Debugging state. // Used when debugging the software.

**【Display interface】 \_\_Bit parameter P415 (password level: 3)**

d7	Reserved	Reserved	Reserved	d3	Reserved	Reserved	Reserved
----	----------	----------	----------	----	----------	----------	----------

**d7\_\_ Prompting range in parameter interface**

0: OFF. // The parameter range prompt is turned off;

1: ON. // The parameter range prompt is turned on.

**d3\_\_ Diagnosis interface prompt**

0: OFF. // The prompt message in diagnosis mode is turned off.

1: ON. // The prompt message in diagnosis mode is turned on.

**【Function switch】 \_\_Bit parameter P416 (password level: 3)**

d7	d6	d 5	d 4	Reserved	Reserved	d1	d0
----	----	-----	-----	----------	----------	----	----

**d7\_\_ Memorizing the current password level**

0: Not performed. // The parameter password level is not memorized. When entering the parameter password input interface, Please input user password is displayed.

1: Performed. // The parameter password level is memorized. The last parameter setting interface memorized by the system is entered.

**d6\_\_ Operation of deleting all programs in EDIT mode**

0: Enabled. // In EDIT mode, it is allowed to delete all the programs in the system.

1: Disabled. // In EDIT mode, it is forbidden to delete all the programs in the system.

**d5\_\_ Operation of clearing all tool offset in tool offset mode**

0: Allowed. // In tool offset mode, it is allowed to delete all the tool offset values.

1: Disabled. // In tool offset mode, it is forbidden to delete all the tool offset values.

**d4\_\_ Turning off driver unit enabling in MANUAL mode**

0: Disabled. // It is forbidden to turn off the drive unit by pressing “DELETE” key in MANUAL state.

1: Enabled. // It is allowed to turn off the drive unit by pressing “DELETE” key in MANUAL state.

**d1\_\_ Parameter display select**

- 0: All are displayed.** // According to the parameter password level, both the parameters alterable and unalterable are displayed.
- 1: Alterable items are displayed. // According to the parameter password level, only the alterable parameters rather than all parameters can be displayed.

**d0\_\_Program modification in EDIT mode**

- 0: Enabled.** // The program lock function is invalid, and it is allowed to modify the program using keys on the operation panel in EDIT mode.
- 1: Disabled.** // The program lock function is valid. It is forbidden to modify the program using keys in EDIT mode. Otherwise, the system issues an alarm.

**【Program line number auto interval】 \_\_Auxiliary parameter P333**

In the Edit operation mode, this parameter determines the increment from last program line number to current one, i.e. the difference value between two line numbers.

**【Y axis rotation clear coordinate】 \_\_ Reference coordinate parameter P027**

**928TD does not support this function.**

When Y axis is used for spindle or worktable rotation control, the user should set the counting range of the rotation coordinate in this parameter. E.g. When Y axis rotates 360 degrees, P027 should be set to 360.000; when Y axis is moving, its tool nose and machine coordinates change circularly within the range of 0~359.999.

When P027 is 0, the rotation axis coordinate clearing function is invalid. Range of P207: 0~9999.

**【Record alarm】 \_\_Bit parameter P417 (password level: 2)**

d7	d6	d 5	d 4	d3	Reserved	Reserved	Reserved
----	----	-----	-----	----	----------	----------	----------

Parameter P417 is for setting whether to record the contents of alarms. When the bit parameter is set to 0, the system records the alarm messages; when changing a part program or during the emergency alarm or alarms in MANUAL/AUTO operation mode, the system automatically records the alarm without bit parameter setting; alarm messages of the system are classified into the following types:

P417 bit parameter (record or not)	P418 bit parameter (display or not)	Alarm message
d7	d7	Alarms in parameter/tool offset operation mode (E001~ E099)
d6	d6	Alarms in EDIT operation mode (E101~ E199)
d5	d5	Alarms relative to program (E201~ E299, E601~ E699)
d4	d4	M03/M04/M05 start/stop operation
d3	d3	Tool change T operation
	d0	Emergency stop alarm, alarms in MANUAL/AUTO mode (E301~ E499)
		Program change in EDIT operation mode

**d7\_\_Recording parameter/tool compensation is**

- 0: Performed.** // The alarm messages in parameter/tool offset operation mode are recorded.
- 1: Not performed.** // The alarm messages in parameter/tool offset operation mode are not recorded.

**d6, d5, d4, d3\_\_ Recording edit alarm, program alarm, spindle alarm, tool change alarm is**

- 0: Performed.**
- 1: Not performed.**

**【Display alarm】 \_\_Bit parameter P418 (password level: 3)**

d7	d6	d5	d4	d3	Reserved	Reserved	d0
----	----	----	----	----	----------	----------	----

Parameter P418 sets whether to display the alarm messages recorded by the system. When the bit parameter is set to 0, the system displays the alarm messages; the system automatically displays all the messages about program change, without bit parameter setting; The system displays the alarm messages as follows:

**d7\_\_ Displaying parameter/tool compensation is**

- 0: Performed.** // The alarm messages recorded by the system in parameter/tool offset operation mode are displayed.
- 1: Not performed.** // The alarm messages recorded by the system in parameter/tool offset operation mode are not displayed.

**d6, d5, d4, d3, d0\_\_ Displaying edit/program/spindle/ tool change alarm, emergency stop in MANUAL/AUTO is**

- 0: Performed.**
- 1: Not performed.**

**【Note】**

- 1) Alarm record is only a search function provided in some special cases; it is suggested that the operator not operate it without special reasons.

**4.6.4.14 Parameters of Interface \_\_P500~P556**

In addition to the control and detection functions for the main devices, like spindle, chuck, tailstock and tool post, this system also provides the control and detection functions for the additional devices such as the signal lamp, lubrication and safety door. The former occupies the fixed signal pins. However, the system does not set fixed signal pins for the additional devices due to the limited number of pins. They are set by the machine tool builder based on the actual conditions.

If the unused signals for the main devices are released, it is available to add the signals required by the additional devices by means of interface parameter setting. In this case, the system can realize the control and detection for the additional devices.

First judge what the interface parameter occupies is the input or output interface. P500~P510 are output interface parameters, and P511~P540 are input interface parameters.

Parameters P541~P556 are used for the multiple-station tool number signal encoding. When bit parameter **P408\_d7** is set to 1 (tool number signal detection mode: Table checking), parameters P541~P556 are used for setting the tool number check signal.

If the interface parameter value is not 0, the input or output interface is occupied; If the set value is the sequence number of a common signal name, it means the pin corresponding to the common signal name of the occupied input or output. If the initial value of the interface parameter is 0, the function of this parameter is not used, i.e. the pin is not occupied.

If the pin of input or output is already occupied by other functions, the interface parameter cannot be set to the occupied common signal name, and the system prompts: [Parameter alarm] – [Illegal I/O setting, this input or output pin has been occupied].

It is allowed to view those unoccupied pins, which are displayed in white and whose names are displayed with **common signal names**, on the diagnosis page. The pin numbers of input or output on the diagnosis page are arranged from top to bottom, and from left to right, which increase from number 1.

For the standard definition and use of the interface parameters, please refer to Section 4.6.5.8 Interface Parameter List in the appendix of this chapter.

#### 【Example】

Setting P511 SAGT to 5 indicates that SAGT occupies the output interface UI05; System UI05 pin inputs the safety door detection function.

Setting P506 M32O to 9 indicates that M32O occupies the output interface UO09; The system UO09 pin outputs the lubrication control function.

#### 4.6.4.15 Initial Values of Variables \_\_P600~P639

The initial values of variable parameters P600~P639 correspond to the variable initial values r001~r040; for details about the variable explanation, please refer to Chapter Seven Statement Programming in Part II Programming

**Note: 928TD has no statement programming function.**

#### 4.6.4.16 Parameters of G76 \_\_P336~P339

When using G76 command, the information about the thread machining can be notified in advance, then the system automatically arranges the thread cutting, and finally machines the threads conforming to the requirements.

##### 【G76 tool angle P】 \_\_Auxiliary parameter P336

This parameter determines the angle between two adjacent teeth threads. The actual thread angle is decided by the tool angle. Therefore, P336 should be the same as the tool angle. Unit: Degree.

##### 【G76 fine turning times L】 \_\_Auxiliary parameter P337

This parameter determines the number of times for the thread fine turning.

**【G76 fine turning cutting amount R】 \_\_ Auxiliary parameter P338**

This parameter determines the cutting amount for the thread fine turning. Unit: mm.

**【G76 rough turning minimum cutting amount Q】 \_\_ Auxiliary parameter P339**

This parameter determines the minimum cutting amount in the thread rough turning. Unit: mm.

**【Note】**

- 1) When the system is executing G76 command ①, it will automatically set the values of words P, L, R, Q of G76 command ① into their corresponding parameters P336, P337, P338, P339 respectively, because these parameter items need to be used in executing command ②.
- 2) When P, L, R, Q in G76 command ① are entirely or partially omitted, the execution of G76 ② is performed by the set values of parameters P336, P337, P338, P339.

**4.6.5 Appendix Parameter List**

**4.6.5.1 Reference Parameter List**

Parameter No.	Parameter level	Parameter designation	Unit	Initial value	Range	User backup
P000	3	Z program reference point	mm	200.000	-99999.999~ +99999.999	
P001	3	X program reference point	mm	150.000		
P002	3	Y program reference point	mm	100.000		
P003	3	2 <sup>nd</sup> program reference point of Z axis	mm	210.000		
P004	3	2 <sup>nd</sup> program reference point of X axis	mm	160.000		
P005	3	2 <sup>nd</sup> program reference point of Y axis	mm	110.000		
P006	3	3 <sup>rd</sup> program reference point of Z axis	mm	220.000		
P007	3	3 <sup>rd</sup> program reference point of X axis	mm	170.000		
P008	3	3 <sup>rd</sup> program reference point of Y axis	mm	120.000		
P009	3	+ direction tool nose soft limit of Z axis	mm	8000.000		
P010	3	- direction tool nose soft limit of Z axis	mm	-8000.000		
P011	3	+ direction tool nose soft limit of X axis	mm	8000.000		
P012	3	- direction tool nose soft limit of X axis	mm	-8000.000		
P013	3	+ direction tool nose soft limit of Y axis	mm	8000.000		
P014	3	- direction tool nose soft limit of Y axis	mm	-8000.000		
P015	2	+ direction mechanical soft limit of Z axis	mm	8000.000		
P016	2	- direction mechanical soft limit of Z axis	mm	-8000.000		
P017	2	+ direction mechanical soft limit of X axis	mm	8000.000		
P018	2	- direction mechanical soft limit of X axis	mm	-8000.000		
P019	2	+ direction mechanical soft limit of Y axis	mm	8000.000		

P020	2	- direction mechanical soft limit of Y axis	mm	-8000.000		
P021	1	Z axis zero coordinate	mm	300.000		
P022	1	X axis zero coordinate	mm	200.000		
P023	1	Y axis zero coordinate	mm	200.000		
P024	2	Z axis zero point offset value	mm	0	0~10	
P025	2	X axis zero point offset value	mm	0	0~10	
P026	2	Y axis zero point offset value	mm	0	0~10	
P027	2	Y axis rotation zeroing coordinate		0	0~9999	

4.6.5.2 Motion Parameter List

Parameter No.	Parameter level	Parameter designation	Unit	Stepper initial value	Servo initial value	Range	User backup
P100	2	Maximum rapid traverse rate limit for Z axis	mm/min	5000	8000	1~15000	
P101	2	Maximum rapid traverse rate limit for X axis	mm/min	2500	4000	1~15000	
P102	2	Maximum rapid traverse rate limit for Y axis	mm/min	5000	6000	1~15000	
P103	2	Minimum start speed for Z axis	mm/min	300	600	1~10000	
P104	2	Minimum start speed for X axis	mm/min	150	300	1~10000	
P105	2	Minimum start speed for Y axis	mm/min	300	600	1~10000	
P106	2	Z axis rapid feed acceleration/deceleration time (ms)	ms	400	150	1~12000	
P107	2	X axis rapid acceleration/deceleration (ms)	ms	400	150	1~12000	
P108	2	Y axis rapid feed acceleration/deceleration time (ms)	ms	400	150	1~12000	
P109	1	Zero return low speed of Z axis	mm/min	100	100	0~2000	
P110	1	Zero return low speed of X axis	mm/min	100	100	0~2000	
P111	1	Zero return low speed of Y axis	mm/min	100	100	0~2000	
P112	2	Cutting feed initial speed	mm/min	100	200	1~2000	
P113	2	Maximum speed limit for Cutting feed	mm/min	4000	6000	1~5000	
P114	2	Linear acceleration/deceleration time for cutting feed (ms)	ms	500	200	1~12000	
P115	2	Exponential acceleration/deceleration time for cutting feed (ms)	ms	500	200	1~12000	
P116	2	Acceleration/deceleration time for thread cutting (ms)	ms	400	150	1~8000	
P117	2	Acceleration/deceleration time for thread run-out (ms)	ms	300	150	1~8000	
P118	2	G99 initial lines		0	0	0~30000	
P119	2	Delay when positioning is switched to cutting (ms)	ms	100	100	0~8000	



4.6.5.3 Drive Parameter List

Parameter No.	Parameter level	Parameter designation	Unit	Initial value	Range	User backup
P200	2	Z axis backlash	mm	0.000	0~10.000	
P201	2	X axis backlash	mm	0.000		
P202	2	Y axis backlash	mm	0.000		
P203	1	Z axis command pulse multiplier		1	1~99999	
P204	1	Z axis command pulse division coefficient		1		
P205	1	X axis command pulse multiplier		1		
P206	1	X axis command pulse division coefficient		1		
P207	1	Y axis command pulse multiplier		1		
P208	1	Y axis command pulse division coefficient		1		
P209	1	Spindle encoder lines		1200	100~5000	
P210	0	Undefined (reserved)		0.000	-1.000~1.000	

4.6.5.4 Auxiliary Parameter List

Parameter No.	Parameter level	Parameter designation	Unit	Initial value	Range	User backup
P300	2	Maximum speed for spindle M41 gear	r/min	3000	0~99999	
P301	2	Maximum speed for spindle M42 gear	r/min	3000		
P302	2	Maximum speed for spindle M43 gear	r/min	3000		
P303	2	Maximum speed for spindle M44 gear	r/min	3000		
P304	2	Spindle minimum speed limit in constant surface speed	r/min	100		
P305	2	Spindle maximum speed limit in constant surface speed	r/min	8000		
P306	2	Thread smooth speed boundary	r/min	100	1~9999	
P307	2	Thread spindle fluctuation alarm	r/min	300	1~99999	
P308	2	Spindle JOG time (ms)	ms	0	0~99999	
P309	2	Speed in spindle JOG	r/min	200	0~99999	
P310	1	Lines occupied by spindle gear control		4	0~4	
P311	2	Gear shift time 1 of the variable frequency spindle (ms)	ms	100	1~99999	
P312	2	Gear shift time 2 of the variable frequency spindle (ms)	ms	100	1~99999	
P313	2	Interval time for spindle gear shift ((ms)	ms	100	1~99999	
P314	2	Output voltage during spindle gear shift (mV	mV	0	0~10000	
P315	2	Delay time from spindle stop to spindle brake (ms)	ms	100	1~99999	
P316	2	Spindle brake output time (ms)	ms	1000	1~99999	
P317	2	Spindle maximum speed limit	r/min	8000	1~99999	
P318	1	Tool post type (0_9)		1	0~9	
P319	1	Maximum tool number		4	1~16	
P320	1	Lines occupied by tool post signal		4	1~8	
P321	2	Tool change time T1 (ms)	ms	100	1~99999	
P322	2	Tool change time T2 (ms)	ms	100	1~99999	

P323	2	Tool change time T3 (ms)	ms	100	1~99999	
P324	2	Tool post CCW rotation clamping time (ms)	ms	1000	1~99999	
P325	2	Tool change rotation time limit (ms)	ms	10000	1~99999	
P326	2	Spindle control pulse time (ms)	ms	10	1~99999	
P327	2	Chuck control pulse time (ms)	ms	10	1~99999	
P328	2	Tailstock control pulse time (ms)	ms	10	1~99999	
P329	2	M code response detection time (ms)	ms	5000	1~99999	
P330	3	Automatic lubrication start time (s)	s	60	0~99999.999	
P331	3	Automatic lubrication interval time (s)	s	600	0~99999.999	
P332	3	Hold time of low pressure alarm (s)	s	600	0.001~99999.999	
P333	3	Program line number automatic interval		10	0~100	
P334	1	Automatic machining time limit	h	0	0~100000	
P336	3	G76 tool angle P	deg	0.000	0~99.000	
P337	3	G76 fine turning times L		1	1~99	
P338	3	G76 fine turning cutting amount R	mm	0.000	0~99.999	
P339	3	G76 rough turning minimum cutting amount Q	mm	0.000	0~99.999	
P341	2	Cutting enabled when rotation speed reaches the percentage		0	0~90	

#### 4.6.5.5 Bit Parameter List

The corresponding bits of the bit parameters are set to 0 or 1 to realize different control functions, thus satisfying the requirements of different machine tools.

Parameter No.	Parameter level	Parameter designation	Initial value	Range	User backup
P400	3	Running setting	0000100	00000000~11111111	
P401	3	Efficiency setting	00000000		
P402	2	Safety setting 1	01000000		
P403	2	Safety setting 2	00100001		
P404	2	Debugging setting	00000000		
P405	2	Motor drive	00011100		
P406	1	Zero point setting 1	00000000		
P407	1	Zero point setting 2	00000000		
P408	2	Tool post setting	00000000		
P409	1	Chuck tailstock (hydraulic system)	00000000		
P410	1	Spindle configuration	00000000		
P411	2	Precision compensation	00000010		
P412	2	Other interfaces	00000000		
P413	2	Interface language	00000000		
P414	2	Communication interface	00000000		
P415	3	Display interface	10001000		
P416	3	Function switch	00000000		
P417	2	Record alarm	00000000		
P418	3	Display alarm	00000000		
P419	2	Safety setting 3	00000000		

4.6.5.6 Interface Parameter List

**I  
OPERATION**

Parameter No.	Parameter level	Signal designation	Function explanation	Standard definition and use	I/O	Initial value	Range	User backup
P500	1	M21O	User command output		O	17	0~99	
P501	1	M23O	User command output		O	18		
P502	1	LMP3	Program run signal light 3	For tricolor light control (green light)	O	0		
P503	1	LMP2	Alarm light control signal 2	For tricolor light control (yellow light)	O	0		
P504	1	LMP1	Alarm light control signal 1	For tricolor light control (red light)	O	0		
P505	1	MDLY	Machine electricity delay power-on control signal		O	0		
P506	1	M32O	Lubrication control switch	For controlling the lubrication switch on the machine tool	O	0		
P507	1	TZD	Tool post worktable brake output	Applicable to SWD120, AK31, SBWD-80 tool post				
P508	1	TFD	Tool post pre-indexing output	Applicable to AK31, SBWD-80 tool post				
P509	1	A001	Reserved					
P510	1	A002	Reserved					
P511	1	SAGT	Safety door detection	For detecting the state of the machine safety door	I	0		
P512	1	Dalm	Feed device alarm detection	For detecting the state of the feed device when executing M230	I	0		
P513	1	M41I	Gear shift in-position signal	Using the frequency-conversion spindle in-position detection signal M41	I	0		
P514	1	M42I	Gear shift in-position signal	Using the frequency-conversion spindle in-position detection signal M42	I	0		
P515	1	M43I	Gear shift in-position signal	Using the frequency-conversion spindle in-position detection signal M43	I	0		
P516	1	M44I	Gear shift in-position signal	Using the frequency-conversion spindle in-position detection signal M44	I	0		
P517	1	M91I	User command input		I	7		
P518	1	M93I	User command input		I	8		

P519	1	RM78	Tailstock advancing in-position detection	Used when using a hydraulic tailstock	I	0		
P520	1	RM79	Spindle retracting in-position detection	Used when using a hydraulic tailstock	I	0		
P521	1	Wsp	External MPG emergency stop	Applicable to an external MPG	I	0		
P522	1	WsY	External MPG axis select Y	Applicable to an external MPG	I	0		
P523	1	WsX	External MPG axis select X	Applicable to an external MPG	I	0		
P524	1	WsZ	External MPG axis select Z	Applicable to an external MPG	I	0		
P525	1	Wbk2	External MPG override 2	Applicable to an external MPG	I	0		
P526	1	Wbk1	External MPG override 1	Applicable to an external MPG	I	0		
P528	1	TFDC	Pre-indexing in-position detection	Applicable to AK31, SBWD-80 tool post	I	0		
P529	1	TXT	Tool number strobe signal	Applicable to AK31, SBWD-80 tool post	I	0		
P530	1	TGR	Tool post overheat detection	Applicable to AK31, SBWD-80 tool post	I	0		
P532	1	G31I	G31 input detection	G31 input interface	I	6		
P533	1	M61I	M61/M62 execution switch	M61/M62 foot switch	I	0	0~32 and 99	
P534	1	M63I	M63/M64 execution switch	M63/M64 foot switch	I	0		
P535	1	M65I	M65/M66 execution switch	M65/M66 foot switch	I	0		
P541 ~ P556	1		Number 1~16 tool number detection signal	Used for multiple-station tool post signal encoding	I	0	0~255	

**【Note】**

- 1) When bit parameter P409\_d4 =1 (the tailstock control function is invalid), the interfaces RM78, RM79 for tailstock advancing/retracting in-position detection are invalid.
- 2) When bit parameter P410\_d6 =0 (spindle S gear shift control), the interfaces M41I, M42I, M43I, M44I for gear shift in-position signals are invalid.

4.6.5.7 Variable Initial Value List

Parameter No.	Parameter level	Parameter designation	Initial value	Range	User backup
P600~P639	3	r001~r040	0	-99999999~99999999	

#### 4.6.5.8 Pitch Error Compensation Parameter List

PARA. No.	Parameter designation (in equal-space mode)	Parameter range	Parameter designation (in unequal-space mode)	Parameter range
P1000	Compensation value (um)	-1000~+1000	1st compensation position point of Z axis (mm)	0~9999.999
P1001	Compensation value (um)	-1000~+1000	Compensation value of 1st point of Z axis (um)	-1000~+1000
P1002 ~ P1299	Compensation value (um)	-1000~+1000	Z axis compensation position point (mm) ~ Z axis compensation value (um)	0-9999.999 ~ -1000~+1000
P1300	Compensation value (um)	-1000~+1000	1st compensation position point of X axis (mm)	0-9999.999
P1301	Compensation value (um)	-1000~+1000	Compensation value of 1st point of X axis (um)	-1000~+1000
P1600	Compensation value (um)	-1000~+1000	1st compensation position point of Y axis (mm)	0~9999.999
P1601	Compensation value (um)	-1000~+1000	Compensation value of 1st point of Y axis (mm)	-1000~+1000
P1900	Position of Z axis pitch measurement start point on machine coordinate system (mm)	-9999.999~9999.999	With the same names as the left parameters	With the same ranges as the left parameters
P1901	Position of X axis pitch measurement start point on machine coordinate system (mm)	-9999.999~9999.999		
P1902	Position of Y axis pitch measurement start point on machine coordinate system (mm)	-9999.999~9999.999		
P1903	Z axis pitch compensation interval (mm)	0~999.999		
P1904	X axis pitch compensation interval (mm)	0~999.999		
P1905	Y axis pitch compensation interval (mm)	0~999.999		

- 【Note】** : 1. T level of all pitch error compensation parameters is 1.  
 2. Pitch error compensation interval range for P1903~P1905: 0~999.999, with the interval not smaller than 0.256 mm.

#### 4.6.5.9 List of Parameters Relative to Command Disabling

Some command names in this system are forbidden to use due to the parameter setting relative to them; once the parameter setting satisfies the conditions to forbid a command name, the system will forbid the use of corresponding command and its function. The parameters relative to command disabling are as follows:

Command name	Command function	Explanation: When the condition below is satisfied, the command is disabled.
M41~M44	Automatic spindle gear shift control	P410_d6=0: Spindle S control: Gear shift
S05~S16	Spindle S control	P410_d6=0 and P410_d5=0: only input S00~S04 in spindle gear shift output
M47~M48		P410_d4=0: Spindle is switched to Y axis
M32~M33	Lubrication function	Interface P506=0
M04	Spindle CCW rotation	P410_d2=1: Spindle CCW rotation signal output is not used.

M10~M11	Chuck clamping/releasing	P409_d7=1: The chuck control function is invalid.
M78~M79	Tailstock advancing/retracting	P409_d4=1: The tailstock control function is invalid.
Command relative to Y axis		P405_d1=0: Without Y axis
Pin programming command	r1001~r1032	P412_d0=0: The pin programming for auxiliary function output is not allowed.

#### 4.6.5.10 List of Parameters Relative to Output Interface Releasing

By parameter setting, the interfaces are not necessarily used for special signals. In this case, the released interfaces can be used as common output interfaces; the interface parameters can be defined as the released output interfaces. The parameters related to output interface releasing are as follows:

Special signal name	Function	Common signal name	Variable name	Released as a common signal by a parameter
M79	Tailstock retraction	UO16	r2016	P409_d4=1: Tailstock control function is invalid.
M78	Tailstock advance	UO15	r2015	P409_d4=1: Tailstock control function is invalid.
M10	Chuck clamping	UO14	r2014	P409_d7=1: The chuck control function is invalid.
M11	Chuck releasing	UO13	r2013	P409_d7=1: The chuck control function is invalid.
TL+	Tool post CW rotation output signal	UO12	r2012	P318=0: Line-up tool P318=9 : M60 tool change by a self-defined command
TL-	Tool post CCW rotation output signal	UO11	r2011	P318=0: Line-up tool P318=9 : M60 tool change by a self-defined command
M8	Coolant ON	UO10	r2010	
M9	Coolant OFF	UO09	r2009	P410_d7=0: Spindle control: Level mode
MSP	Spindle brake signal	UO08	r2008	P410_d3=1: Without spindle brake signal output
M3	Spindle CW rotation	UO07	r2007	
M4	Spindle CCW rotation	UO06	r2006	P410_d2=1: Spindle CCW rotation signal unused
M5	Spindle stop	UO05	r2005	P410_d7=0: Spindle control: Level mode
S04/M44	Spindle gear output	UO04	r2004	P310<4: Releasing this signal, with 0/1/2/3 channel occupied
S03/M43	Spindle gear output	UO03	r2003	P310<3: Releasing this signal, with 0/1/2 channel occupied
S02/M42	Spindle gear output	UO02	r2002	P310<2: Releasing this signal, with 0/1 channel occupied
S01/M41	Spindle gear output	UO01	r2001	P310=0: Releasing this signal, with 0 channel occupied

#### 4.6.5.11 List of Parameters Related to Input Interface Releasing


By parameter setting, the interfaces are not necessarily used for special signals. In this case, the released interfaces can be used as common input interfaces; the interface parameters can be defined as the released input interfaces. The parameters related to input interface releasing are as follows:

Special signal name	Function	Common signal name	Variable name	Released as a common signal by a parameter
+LT	Z/X/Y axis + direction limit	UI32	r1032	P404_d6=1: Shielding the hard limit alarm
-LT	Z/X/Y axis - direction limit	UI31	r1031	P404_d6=1: Shielding the hard limit alarm
DecY	Y axis deceleration signal	UI30	r1030	P405_d1=0: Without Y axis P406_d5=0: Without Y axis machine deceleration switch
DecX	X axis deceleration signal	UI29	r1029	P406_d6=0: Without X axis machine deceleration switch
DecZ	Z axis deceleration signal	UI28	r1028	P406_d7=0: Without Z axis machine deceleration switch
SP	External pause signal	UI27	r1027	P412_d7=0: Without external start/pause signal
ST	External cycle start signal	UI26	r1026	P412_d7=0: Without external start/pause signal
MXZ1	Feed hold signal	UI16	r1016	P412_d6=0: Without an external feed/spindle hold knob
MXZ2	External feed/spindle hold signal	UI15	r1015	P412_d6=0: Without external feed/spindle hold knob
RM10	Chuck clamp in-position detection	UI14	r1014	P409_d7=1:: Chuck control function is invalid P409_d5=0: No chuck response detection
RM11	Chuck releasing in-position detection	UI13	r1013	P409_d7=1: Chuck control function is invalid. P409_d5=0: No chuck response detection
TPS	Input signal of hydraulic tailstock foot switch	UI12	r1012	P409_d4=1: Tailstock control function is invalid. P409_d0=1: No hydraulic tailstock foot switch input
SHL	Input signal of hydraulic chuck foot switch	UI11	r1011	P409_d7=1: Chuck control function is invalid. P409_d1=1: No hydraulic chuck foot switch input.
PRES	Low pressure detection	UI10	r1010	P412_d5=0: No low pressure detection
TCP	Tool post clamp in-position signal	UI09	r1009	P318=0: Line-up tool P318=9: M60 tool change by a self-defined command P408_d6=0: No tool post clamp in-position signal
T1	Tool post tool number signal T1	UI01	r1001	P318=0 or 9
T2	Tool post tool number signal T2	UI02	r1002	P318=0 or 9: P320<2 is released
T3	Tool post tool number signal T3	UI03	r1003	P318=0 or 9: P320<3 is released
T4	Tool post tool number signal T4	UI04	r1004	P318=0 or 9: P320<4 is released
T5	Tool post tool number signal T5	UI05	r1005	P318=0 or 9: P320<5 is released
T6	Tool post tool number signal T6	UI06	r1006	P318=0 or 9: P320<6 is released.
T7	Tool post tool number signal T 7	UI07	r1007	P318=0 or 9: P320<7 is released.
T8	Tool post tool number signal T8	UI08	r1008	P318=0 or 9: P320<8 is released.

## 4.7 Tool Offset Operation Mode

**Tool offset operation mode:** The system compensates the tool offset. For each step of operation, the system provides corresponding intelligent prompt messages. It is also available to press the **hp2** key on the upper right corner of the system to view the operation key list for the system tool compensation.

The explanations for the input format and examples about related setting and operation in the manual are as follows: The function keys required to press are indicated by a sign; the letter keys, numeric keys to be input are indicated by an underline; the prompt messages of the system are indicated by a **rectangle**.

If incorrect data are input during the input of letters or figures, press  key to delete them, and then input the correct data


When setting or inputting an item, or during the man-machine dialogue, if **ESC** key is pressed before confirmation, the current operation is quit.

There are 64 groups of tool offset values from T01~T64 set in this system, with each tool offset number corresponding to one group. In each group are recorded five data, which are Z offset value, X offset value, R tool radius, T tool shape, S tool setting record. Therein, by the manual tool setting operation, the number of tool offset groups can be the same as the number of used tools. Other tool offset data can only be input from the keyboard.

◆ The main functions of the tool offset operation mode consist of:

- ☆ Selecting, modifying and clearing tool offset data;
- ☆ Through a USB interface, transmitting the tool offset data between U disc and CNC system.
- ☆ Through RS232 communication interface, transmitting the tool offset data between external computer and CNC system;
- ☆ Through **RS232** communication interface, transmitting the tool offset data between two CNC systems.



Press operation mode select key  to enter Tool offset operation mode, as displayed in fig. 4-9. (Y axis offset value is invalid)



[OFFSET]						hp2
OFFSET NO.	Z	X	R	T	S	Y
01	0001.111	-0001.111	0000.100	0	00	0001.111
02	0002.222	-0002.222	0000.200	0	00	0002.222
03	0003.333	-0003.333	0000.300	0	00	0002.222
04	0004.444	-0004.444	0000.000	0	00	0002.222
05	0000.000	0000.000	0000.000	0	00	0000.000
06	0000.000	0000.000	0000.000	0	00	0000.000
07	0000.000	0000.000	0000.000	0	00	0000.000

EDIT
JOG
AUTO
PARA
OFFT
DGN





Fig. 4-9 Tool compensation operation mode

### 4.7.1 Searching Tool Offset Value

The search of the tool offset value is to search the desired tool offset value; there are two methods:

#### Method one: Scanning

The contents of each tool offset value can be viewed in Tool Offset operation mode. By pressing

 ,  key, the last or next tool offset value can be searched. By pressing  ,  key, the last page or next page of tool offset values can be searched, with each page containing 7 lines.

#### Method two: Searching

**P** + Offset number + **ENTER** .

### 4.7.2 Inputting Tool Offset Data from Keyboard

Range of word X, Z: -8000.000 mm~8000.000mm; Range of word R: 0 mm~8000.000mm;







Range of word T: Integer 0~8; Range of word S: Integer 0~16.


There are two ways to input tool offset data from the keyboard: absolute input and relative input.

The operations are as follows:


#### Absolute input of tool offset data:

1) Select Tool Offset operation mode.

2) Press  ,  ,  ,  key to move the highlighted block cursor to the tool offset number to be changed; or find the tool offset number to be changed using SEARCHING method; then press  ,  key to move the highlighted block cursor to the Z offset value, X offset value, R tool radius, T tool shape or S tool setting record to be changed,

- 3) Press **INPUT** key, or directly input the desired data.
- 4) Input the desired data from the keyboard. During the data input, if incorrect data is input, press  key to cancel it then input the right value.
- 5) Press **ENTER** key to store the input data into the offset value of the currently-selected tool offset number.
- 6) If sign “\*” appears ahead of the data to be changed, it means the data modification succeeds

**Relative input of tool offset data:**

- 1) Select Tool Offset operation mode; then move the highlighted block cursor to the data to be changed.
- 2) Press **INPUT** key, or directly input the desired data from the keyboard.
- 3) Input the desired data from the keyboard. During the data input, if the input data is incorrect, press  key to cancel it then input the right value. By pressing **ALTER** key, the system performs addition operation by adding the input data to the original value of the selected offset value.

### 4.7.3 Clearing Offset Values of Each Group

The procedures of clearing the offset values of each group are as follows:

- 1) Select Tool Offset operation mode.
- 2) Move the highlighted block cursor to the S tool setting record of the tool offset number to be cleared.
- 3) Press Delete key to clear the Z offset value, X offset value, R tool radius, T tool shape and S tool setting record of the tool offset number.

### 4.7.4 Tool Compensation **hp6** Function

By pressing hp6 key on the page of Tool Offset operation mode, the display is as follows:

Tool offset transfer menu  
 Key U – USB interface operation  
 Key R – RS232 interface operation  
 Key Z – Clearing all tool offse data  
 Key ESC – Exit

#### 4.7.4.1 Communication and Standard Format of Tool Offset Data

##### 1. Tool offset data transmission through USB interface:

In the root directory of the U disc, a folder with name "C001OFT" needs to be created. The naming rule of the tool compensation file includes 10 characters, i.e. "OFT"+File number (3 digits) + ".TXT". The file number cannot be greater than 666, or the system will not list the file when reading the U disc. The tool compensation files must be placed in the folder C001OFT.

##### 2. Tool offset data transmission through RS232 interface

For details about the operation of the transmission software using RS232, please see Part I Operation: Chapter Five RS232 and USB System Communication. The transmission of the tool compensation can be completed according to the system operation prompts.

##### 3. Standard format of TXT Tool Compensation File on PC

On a PC, it is possible to edit tool compensation files using TXT, LST text. but the file name and its contents must be edited in the standard format required in the system, so that the files can be sent to the system correctly. The specifications are as follows:

- 1) On a personal computer, the user should name the tool compensation file with suffix TXT or LST, e.g. "OFT088.TXT"; It is recommended that the user use suffix TXT for the convenience of the parameter file operation on the PC.
- 2) The first line of the contents of the TXT file must be the tool compensation mark: "CNC\_GSKC001"; which is indispensable.
- 3) The contents behind "/" are the annotation, which lists in turn the offset number, Z offset value, X offset value, tool radius, tool shape, tool setting record.
- 4) The contents of the tool compensation must be consistent with the requirement of the standard format.

E.g. T01: 00000.000, 00000.000, 00000.000, 0, 00

Thereinto:

- ◆ Range of tool number: 01~64. If it is exceeded, the system processes it as an error;
- ◆ Offset number format: Three parts: T + Number (01~64) + : , which are indispensable.
- ◆ A comma must be used to separate different tool compensation data in a line;
- ◆ There are five compensation data for each line. More than five data indicates there are illegal characters in the tool compensation data;
- ◆ If there are less than five tool compensation data in a line (i.e. not all Z offset value + X offset value + R tool radius + T tool shape + S tool setting record are included), the system processes the data as Z offset value , X offset value , R tool radius, T tool shape and S tool setting record from left to right; if the data at the end of a line need to be modified, the data at the beginning should be added first. The values of Z, X, R, T and S cannot exceed their respectively-specified ranges.

## 【Example】

CNC\_GSKC001

// Tool No.	Z offset	X offset	Tool nose radius	Imaginary tool nose	Clamp tool NO.
T01:	00000.000,	00000.000,	00002.000,	0,	00
T02:	00000.000,	00000.000,	00000.000,	0,	00
T03:	00000.000,	00000.000,	00000.000,	0,	00
T04:	00000.000,	00000.000,	00000.000,	0,	00
T05:	00000.000,	00000.000,	00000.000,	0,	00
T06:	00000.000,	00000.000,	00000.000,	0,	00
T07:	00000.000,	00000.000,	00000.000,	0,	00
.....					
T63:	00000.000,	00000.000,	00000.000,	0,	00
T64:	00000.000,	00000.000,	00000.000,	0,	00

## 4.7.4.2 Tool Compensation Data Clearing

First set bit parameter **P416\_d5=0** (Clearing all tool compensation values in Tool Offset mode is allowed), then press **hp6** key in the Tool Offset operation mode, finally press **Z** key to clear all the tool compensation data.


## 4.8 Diagnosis Operation Mode

Diagnosis operation mode: The system detects and displays the states of input/output (I/O) signals, spindle speed, encoder lines in real time. For details about the operation key list for the system diagnosis, press **hp2** key on the upper right corner of the system.

◆ The main functions of the diagnosis mode consist of:

- ☆ Self-diagnosis function;
- ☆ Displaying the states of the input/output (IO) signals;
- ☆ Diagnosis of spindle control function input, output signals.
- ☆ Diagnosis of tool post control function input, output signals;
- ☆ Diagnosis of hard limit signals for each axis;
- ☆ Diagnosis of machine zero point (machine reference point ) signal for each axis;
- ☆ Diagnosis of spindle speed and encoder lines







Press operation mode select key  to enter the page of Diagnosis operation mode, as displayed in figure 4-10:



[ DIAGNOSIS ]				Input interface	hp2
UI01 1	UI09 1	UI17 1	UI25 1	ALZ 1	Spindle speed 0000
UI02 1	UI10 1	UI18 1	UI26 1	ALX1	
UI03 1	SHL 1	UI19 1	UI27 1	ALY 1	Encoder lines 0000
UI04 1	TPS 1	UI20 1	UI28 1	PCZ 1	
UI05 1	UI13 1	UI21 1	UI29 1	PCX 1	
UI06 1	UI14 1	UI22 1	UI30 1	PCY 1	
UI07 1	UI15 1	UI23 1	-LT 1	PA 1	
UI08 1	UI16 1	UI24 1	+LT 1	API 1	
<div style="display: flex; justify-content: space-around;"> <span>EDIT</span> <span>JOG</span> <span>AUTO</span> <span>PARA</span> <span>OFFT</span> <span>DNG</span> </div>					

Fig. 4-10 Diagnosis operation mode

### 4.8.1 Searching Interface Signal

First set bit parameter **P415\_d3=1** (the diagnosis page prompt is ON). In the Diagnosis operation mode, not only the explanations of each I/O but also the common signal name, hardware interface and pin number of each I/O can be viewed; if the bit parameter is set to OFF, it is unavailable to view the relative explanations.

Press  ,  ,  ,  key to search each I/O. The I/O searched is highlighted.

Press  or  key to display the pages of input interface and output interface alternately.

#### 4.8.2 Explanations for Display of Interface Signals Names

This system has 23-channel switch value input signals and 18-channel switch value output signals; each signal has a name, which indicates the meaning of the signal.

**Common signal name:** The names of input signals are UI01~UI32, and those of output signals are UO01~UO32, with each signal corresponding to one pin. Thereinto, UI17~UI25 signals and UO19~UO32 signals are not led out in the interfaces.

**Special signal name:** For a specific machine tool, once a signal is occupied by a special function, the signal is assigned a special name. The system sets a standard special signal name for the signal to be used by each function.

On the display of Diagnosis, the signal used by a special function is indicated by a special signal name; it means this function is set enabled in the parameter. The initialization parameters in the system has enabled most of the auxiliary functions. Thus most of pin signals are displayed with a special signal name, while the unoccupied signals are displayed with a common signal name.

For the explanations and connection of command signal names and special signal names, please refer to Part III Connection in this manual.

#### 4.8.3 Explanation of Input Interface Diagnosis

On the display of input interface diagnosis, when an external signal is valid, its corresponding bit is displayed as 0; when an external signal is invalid, its corresponding bit is displayed as 1. The diagnosis of input interface signals is performed recurrently all the time, displaying the current signal states anytime.

The definition of the input interface signal is determined by parameter. Therefore, the definition varies with the corresponding setting of the parameter.

##### 【Example】

The standard configuration of the electric tool post is 4 stations. When electric tool post configuration of the system is 4 stations, the diagnosis messages are shown as follows:

The auxiliary parameter P320 is 4; parameter P319 is 4; parameter P318 is 1: Electric tool post.  
The leftmost column on the diagnosis input page displays the diagnosis messages of T1 ~ T4.

#### 4.8.4 Explanation of Output Interface Diagnosis

When the bit in the output interface diagnosis is displayed as 0, the output of the corresponding bit is valid; and when the bit is displayed as 1, the output is invalid. The output interface diagnosis displays the current hold state of each output bit. If the signal is pulse, and the pulse time is short, the output of the bit is valid, but it is still displayed as 1.

The definition of the output interface signal is also determined by the parameter, and the definition varies with the setting of the parameter.

#### 4.8.5 Output Interface Operation Function

Move the cursor to the input signal to be modified using direction keys, and change the value of the output signal where the cursor is located by pressing 1 or 0 key. If the value differs from the current value, it is displayed in red; if the value is the same as the current value, it is displayed in yellow.

##### 【Note】

**Diagnosis output interface operation is used for system debugging. For system safety, the operation is available only when the parameter password level is higher than 2.**

#### 4.8.6 Spindle Encoder and Spindle Speed Detection

The system can detect and display the pulse number per revolution for the spindle encoder, and display the number of lines of the encoder automatically according to the detection result.

The number of lines of the encoder indicates the number of pulses per revolution of the encoder in use.

The spindle speed indicates the current spindle actual speed (unit: r/min).

##### 【Explanation】

- 1) If the spindle is not started, the spindle speed is displayed as 0.
- 2) When the spindle is started, and the initial page of the Diagnosis operation mode is entered, if the detected number of lines of the encoder is inconsistent with the number of lines of the spindle encoder, the system prompts: [Diagnosis check prompt]: Encoder line number inconsistent with parameter.
- 3) The spindle encoder rotates simultaneously with the spindle: I.e. each time the spindle rotates one revolution, the encoder also rotates one revolution; otherwise, the spindle speed detected is inconsistent with the actual value.

#### 4.8.7 Diagnosis hp6 Function

Press hp6 key on the diagnosis page, then the display is as follows:

Key 0 – Viewing alarm messages	Key 1 –Viewing color coding
Key 2 – Entering keyboard test	Key 3 –Viewing CPLD pulse number
Key 4 – Viewing version info.	Key 5 –RAM import/export
ESC – Exit	

Key 0 – Viewing alarm messages: Displaying the system alarm record;

Key 1 – Viewing color coding: Displaying 256 kinds of color and code, e.g. 00 indicates black, FF

indicates white;

Key 2 – Keyboard test: Testing the keyboard;

Key 3 – Viewing CPLD pulse number: Displaying the number of pulses of X/Y/Z;

Key 4 – Version information: Displaying the system version information: System software, CPLD version information and software version re-loading operation;

Key 5 – RAM import and export: The system RAM is transmitted through U disc or serial port RS232.

### 1. RAM data transmission through USB interface

It is required to create a folder with name “C001RAM” in the root directory of U disc. The naming rule of the system RAM file includes 10 characters, which are “RAM”+File number (3 digits) +“.TXT”. The file number cannot be greater than 999, or the system does not list the file when reading the U disc. The memory file must be placed in the folder C001RAM. If there is no such a folder, the system prompts: **No RAM file in the directory** after detection. After opening the RAM directory page, the system can only send the RAM files.

### 2. RAM data transmission through RS232 interface

This CNC system can transmit RAM data through RS232 interface. It is possible to complete the RAM transmission in terms of the operation prompts provided by the system.

#### 【Note】

- 1) It is suggested that the user not use **hp6** function key.
- 2) For the system safety, it is allowed to operate the RAM and load and upgrade the version only when the parameter password level is higher than level 2.

#### 4.8.7.1 Display of Alarm Record

In the Diagnosis operation mode, press **hp6** key, then **0** key. Then the system displays the alarm record messages, as shown in figure 4-11.

**Alarm record function:** The system records the alarm messages in detail. For each step of operation, the system provides corresponding intelligent prompt messages. It is also available to view the operation key list for the system alarm record function by pressing the **hp2** key on the upper right corner of the system.

This system is capable of storing up to 4031 alarm messages. When the alarm record number exceeds 4031, the system will automatically delete the oldest alarm messages, and display the newest alarm messages. Therefore, it is necessary for the operator to download and save the alarm record, lest the old alarm record cannot be referred to.

#### ◆ The main functions of the alarm record consist of:

- ☆ Searching the number of startup times, alarm messages
- ☆ Displaying the total alarm record counts, the record counts required to list according to parameter setting, and the current record number.
- ☆ Transmitting the alarm record to a U disc through USB interface.
- ☆ Transmitting the alarm record to the external computer through RS232 communication interface.



Total records: 890		Displayed records: 220		No. 0220		hp2	
NO.125	90:59:02	E346	Tailstock retracting response detection overtime alarm				
	1001:20:59	M05	Spindle stop				
	2001:05:59	%028	Part program change				
NO.126	00:01:59	E367	Chuck unclamp in-position signal abnormal				
	01:11:20	E606	Illegal use of M91; M91 not defined in interface parameter				
	01:21:59	E302	Z driver alarm				
NO.129	00:40:20	E303	X driver alarm				
	120:40:20	E317	X tool nose - direction soft limit alarm				

Fig. 4-11 Alarm record display

**【Example】**

E.g. latest alarm record: 120:40:20 E317 X tool nose – direction soft limit alarm

Thereinto, the system startup times is the NO.129 displayed in the last alarm record, i.e. the time this alarm record (220<sup>th</sup>) occurs is 120 hours 40 minutes and 20 seconds after the 129<sup>th</sup> startup. The contents of the alarm is “E317: X tool nose – direction soft limit alarm”.

**【Note】**

- 1) When replacing a program, the alarm record will display the program number; just like %028 in figure 4-13, which indicates the part program is changed to %028 after 2001 hour 05 minute 59 second after the 125<sup>th</sup> startup.
- 2) The maximum time displayed for the alarm record is 99999 hours 59 minutes and 59 seconds; the maximum number of startup times displayed for the alarm record is 999.
- 3) The current alarm record is highlighted, and on the right side of the scrollbar is displayed the position where the current alarm record is located among all the displayed alarm records.
- 4) The alarm record is a search function provided in some special cases; the operator should not operate it without special reasons.

4.8.7.2 Searching Alarm Record

The alarm record search is to search the specified alarm record, view the identical alarm records, or count the number of error numbers which is the same as the error number (alarm number in system alarm message, e.g. E001) of the current alarm record. There are three methods:

**Method one: Scanning**

It is possible to view the contents of the record on the alarm record display page. By pressing



key, you can view the last or next record. By pressing



key, you can

view the last or next page of records, with each page displaying up to 8 lines.

**Method two: Positioning**

**P** + **Record number** + **ENTER**. (Record number means: N<sup>th</sup> piece of alarm record)

**Method three: Others**

**F** + **Searching type** + **Number** + **ENTER**

The searching type includes: E—by alarm error number, M— by spindle command, T— By tool change number, P— by program number; press **E**, **M**, **T** or **P** key to enter the search modes of different types.

In addition, the system provides the function of viewing the identical alarm records. By pressing



, key, you can view the alarm records which is identical with the current one upward and downward.



By pressing key, you can count the number of alarm records which is identical with the current one; The display window disappears in 5 seconds, or it disappears immediately within 5 seconds if any other key is pressed.

**【Relative parameters】**

Parameters relative to the alarm record function: **P417, P418**.

**4.8.7.3 Alarm Record hp6 Function**

By pressing hp6 key on the alarm record display page, the display is as follows:

```
Alarm record sending mode select
Key U – USB interface transmission
Key R – RS232 interface transmission
Key ESC – Return
```

**1. Sending alarm records through USB interface:**


The alarm records sent through the USB interface are saved in the folder “C001ERR” in the root directory of the U disc. The naming rule of the alarm record files includes 10 characters, which are “ERR”+ File number (3 digits) +“.TXT”. The file number cannot be greater than 254.


**2. Alarm records sent through RS232 interface**


For the operation of the communication software using RS232 for transmission, see Chapter Five RS232 and USB System Communication in Part I Operation. GSKCOM\_C001.EXE can realize the file transmission between PC and CNC, which is easy to operate and has high communication efficiency and high reliability. The transmission of alarm records can be completed by following the system operation prompts.


## 4.8.8 Machine Auxiliary Function Control


You can execute the machine auxiliary functions by using the auxiliary function keys on the operation panel, rather than by inputting commands.


Press  key: The spindle rotates clockwise. When bit parameter **P410\_d7=0**, the LED indicator lights up, and the corresponding bit of M3 in the output interface displays 0. (When bit parameter **P410\_d7=1**, the LED indicator lights up, and M3 in the output interface firstly outputs validity. After the pulse output, the corresponding bit of M3 displays 1.)

Press  key: The spindle stops. (When bit parameter **P410\_d7=1**, the diagnosis output interface page displays M5.)

Press  key: The spindle rotates counterclockwise. When bit parameter **P410\_d7=0**, **P410\_d2=0**, the LED indicator lights up, and the corresponding bit of M41 in the output interface displays 0. (When **P410\_d7=1**, **P410\_d2=0**, the LED indicator lights up, M4 in the output interface firstly outputs validity. After the pulse output, the corresponding bit of M4 displays 1. When bit parameter **P410\_d2=1**, the diagnosis output interface page does not display M4.)

Press  key: The coolant switches between ON/OFF. When bit parameter **P410\_d7=0** and the coolant is ON, the LED indicator lights up, and the corresponding bit of M8 in the output interface displays 0; when the coolant is OFF, the LED indicator goes out, and the corresponding bit of M8 in the output port displays 1. (When **P410\_d7=1** and the coolant is ON, the LED indicator lights up, and the corresponding bit of M8 in the output port firstly outputs validity. After the pulse output, the corresponding bit of M8 displays 1. When the coolant is OFF, the indicator goes out, M9 in the output interface firstly outputs validity. After the pulse output, the corresponding bit of M9 displays 1.)

Press  key: Used for spindle gear cycle. The spindle motor executes S01~S04 or S00~S15 (based on the setting of parameter **P410\_d6**, **P410\_d5**)

Press  key: The tool post rotates to the next tool number, and the tool number state is displayed on the corresponding position of input interfaces T1~T4.

## CHAPTER FIVE RS232 AND USB SYSTEM COMMUNICATION

This system can transmit the part programs, system parameters, system software and tool offsets through RS232 and USB interfaces; for the operation of the part program transmission, see Section 4.3.3 “Edit Operation Mode” in Part I Operation; for the operation of the transmission of system parameters and system software, see Section 4.6.3 “Parameter Operation Mode” in Part I ; for the operation of tool offset transmission, see Section 4.7.4 “Tool Offset Operation Mode” in Part I .

### 5.1 RS232 Communication

RS232 communication is the serial port communication mode, which realizes data transmission between CNC and PC, or between CNC and CNC, through the serial port communication cable (with three lines).

#### 5.1.1 Communication between CNC and PC

The file sending and receiving between CNC and PC is realized by the communication software GSKCOM\_C001.EXE. The software is easy to operate, and characterized by high communication efficiency and reliability.

1) PC configuration:

Hardware: Common-use PC with RS232 serial port; serial port communication cable (with three lines)

Operation system: Microsoft Windows 98/2000/2003/XP.

2) Introduction to GSKCOM\_C001.EXE communication software:

See the communication software GSKCOM\_C001.EXE on PC.

#### 【Preparation before communication】

1) Connect the communication cable on the condition that both PC and CNC are powered off: Insert DB9 female plug into the RS232 communication interface on the front cover of the CNC, and insert the other DB9 female plug into the 9-pin serial port (COM1 or COM2) on the PC;

2) Select the port and baudrate for communication on PC. The communication baudrate is determined by the sender.

◆ **Setting the port of GSKCOM\_C001.EXE communication software on PC**

After running the communication software, leftclick to select “port number”, selecting the communication serial port..

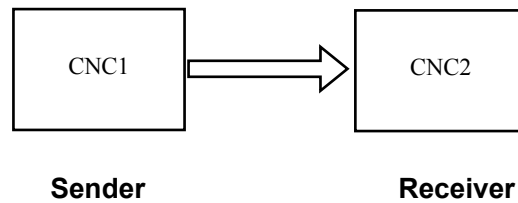
◆ **Setting the baudrate of GSKCOM\_C001.EXE communication software on PC**

After running the communication software, leftclick the “baudrate” to select the communication baudrate (Set to: 38400 (unit: b/s))

- 3) Set the communication baudrate of bit parameters **P414\_d7**, **P414\_d6**: See Section 4.6 “Parameter Operation Mode” in Part I Operation.

### 5.1.2 Communication between CNC and CNC

For convenient operation, it is allowed to transmit data between two CNC systems (both are GSK928TD) . The CNC for sending data is referred to as the sender; while the CNC for receiving the data is the receiver, as shown in the figure below:



#### 【Preparation before communication】

- 1) Connect the communication cable on the condition that both CNC systems are powered off: Insert the DB9 female plug into the RS232C communication interface on the front cover of each CNC.
- 2) Set the communication baudrate of bit parameters **P414\_d7**, **P414\_d6**: See Section 4.6 “Parameter Operation Mode” in Part I Operation.

#### 【Precautions for data communication between two CNC systems】

- 1) The sender and receiver are in the same operation mode (Edit, Parameter, Tool Compensation);
- 2) During parameter sending or receiving, the sender and receiver must enter corresponding operation authority.
- 3) The operation procedures are the same as those of “data sending” and “data receiving” of the communication software GSKCOM\_C001.EXE.

#### 【Note】

- 1) To stop the transmission, press “Cancel” button on the communication software; or press **RESET** key or **ESC** key on the CNC.
- 2) Do not turn off the power during the data transmission,. Otherwise, the data transmission error may occur.
- 3) The communication cable for the communication between CNC and PC is the same as that for the communication between CNC and CNC.

## 5.2 USB Communication

This system supports USB communication mode, thus realizing the data transmission between CNC and U disc with the USB interface.

### 5.2.1 USB Operation

When performing USB operation, insert the U disc directly into the USB interface on the system panel. If the U disc contains the folder and file names created in accordance with the system requirements in its rooted directory, the system automatically identifies and opens the U disc. At the same time, on the system interface is displayed the USB icon.

#### 【Note】

- 1) After the U disc operation, the user must press **ESC** key to turn off the U disc before pulling it out. Then when the USB icon displayed on the system interface disappears, pull the U disc out from the USB interface. Otherwise, the system hardware and U disc may be damaged.
- 2) Never attempt to pull the U disc out when its indicator is blinking, because it is reading or writing data at this moment. Pulling it out halfway may cause damage to the hardware or result in data loss.
- 3) A U disc generally has a write-protect switch. The switch should be performed before the U disc is inserted into the system interface rather than during the U disc running.
- 4) The storage size of the U disc should be as small as possible, in case that the communication speed between system and U disc is affected. It is suggested that the user not use those removable storage devices with a size larger than 8G. Otherwise, the system hardware and the devices may be damaged.
- 5) When the U disc operation fails, press **RESET** key on the system, and then re-insert the U disc after pulling it out.

### 5.2.2 USB File Directory Requirements

When transmitting different kinds of data during USB communication, the system requires different names for the USB files. The folder names required to create in the rooted directory of U disc are as follows:

Name type \ Data type	Part program	Parameter data	Tool offset data	System software upgrade
Folder name in rooted directory of U disc	C001PRO	C001PAR	C001OFT	C001DATA
File name in folder	CNCxxx.TXT	PARxxx.TXT	OFTxxx.TXT	DATAxxx.TXT
Range of XXX in file name	0~254	0~999	0~999	0~254

Name type \ Data type	Memory whole upgrade	Alarm history	System RAM
Folder name in U disc rooted directory	C001MEMO	C001ERR	C001RAM
File name in folder	MEMOxxx.TXT	ERR xxx.TXT	RAMxxx.TXT
XXX range in file name	0~254	0~999	0~999



---

## II PROGRAMMING





# PROGRAMMING

## CHAPTER ONE PROGRAMMING FUNDAMENTAL

The automatic machining of CNC machine is the course of automatically running edited part programs. The programming is defined that the drawing and the technology of machining workpiece are described with CNC language and are edited to the part programs. Here describes the definition of command and the programming mode of CNC part programs. Please read carefully these contents before programming.

### 1.1 Coordinate Axis and Its Direction

This system has defined the controlled axis and its motion according to GB/T 19660—2005/ISO 841:2001 *Industrial Automation System and Integration-Numerical Control of Machines-Coordinate System and Motion Nomenclature*. The two coordinate axes are named with X and Z, which are perpendicular to each other to form X—Z plane rectangular coordinate system as Fig. 1-1:

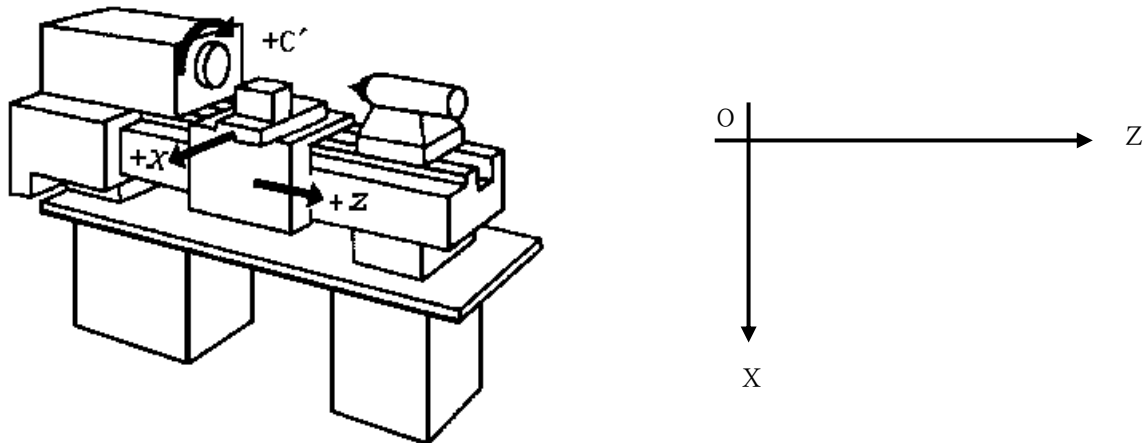


Fig. 1-1 X—Z plane rectangular coordinate system 1 (front tool post coordinate system)

- X: It is defined to be perpendicular with the rotary centerline of spindle. X positive direction is the one in which the tool leaves from the rotary center of spindle.
- Z: It is defined to be coincident with the rotary centerline of spindle and Z positive direction is the one that the tool leaves from the headstock.



Fig. 1-2 X-Z plane rectangular coordinate system 2 (rear tool post coordinate system)

The coordinate system in the manual defines X, Z based on Fig. 1-1 (front tool post coordinate system).

The execution course of G02,G03 commands is different from that in front tool post coordinate system when Fig.1-2 (rear tool post coordinate system) is used to define X, Z (see G02, G03 commands):

## 1.2 Machine Coordinate System, Machine Zero

Machine coordinate system is the reference for the system to count coordinates, and is fixed on the machine.

Machine zero (or **machine reference point, or mechanical zero**) is a fixed point on the machine. Generally, it is set at the position of X, Z max. stroke in the positive direction, with the machine zero check device installed on the machine. Do not use the machine zero return function or set bit parameters **P406\_d7, P406\_d6, P406\_d5, P406\_d4, P406\_d3, P406\_d2** to 0, i.e. “no machine zero” when the machine zero check device is not installed.

## 1.3 Program Reference Point

In the machine coordinate system, the operator should set a position that is convenient and safe for tool change and workpiece fixing when the tool post stops at it. Program reference point is set when the tool post stops at the position, which is called program reference point (or **program zero**). Program reference point coordinates are relative to the machine coordinate system.

Once the reference point is determined, the tool can return to the reference point by executing the reference point return function in JOG operation mode or AUTO operation mode. Even if the system is switched off, the reference point remains valid. If the stepper motor is employed, there may be slight error caused by the motor vibrating upon system power on. Execute the reference point return once to avoid the error.

The program reference point is automatically set to X=150, Z=150 without setting the program reference point after the system is switched on for the first time.

## 1.4 Machine 2nd, 3rd Program Reference Point

In the machine coordinate system, the operator can also set the 2nd, 3rd program reference point. Their functions are similar to the program reference point, and the system has corresponding commands to move the worktable to the 2nd, 3rd program reference point.

## 1.5 Workpiece Coordinate System

A coordinate system with a certain point on the workpiece set as its origin is referred to as a workpiece coordinate system. Its axes are separately parallel with X, Z axis in the same direction.

After the workpiece coordinates is created, all absolute coordinate values in programming are the coordinate values in the workpiece coordinate system. Generally, Z axis in the workpiece coordinate system is set on the rotary center of workpiece.

According to the actual condition in programming, define the workpiece coordinate origin, i.e. the programming origin in the workpiece drawing and the coordinate origin of CNC system command. The workpiece coordinate system is created by setting workpiece coordinates.

### 1.6 Programming Coordinate

Programming coordinate is for workpiece coordinate system.

This system can use absolute coordinates (**X, Z word**), relative coordinates (**U, W word**) or compound coordinates (**X/W, U/Z word**) for programming .

#### 1.6.1 Absolute Coordinate Values

The absolute coordinate value is the distance to the coordinate origin, i.e. the coordinate position of the end point of tool movement, as Fig. 1-3

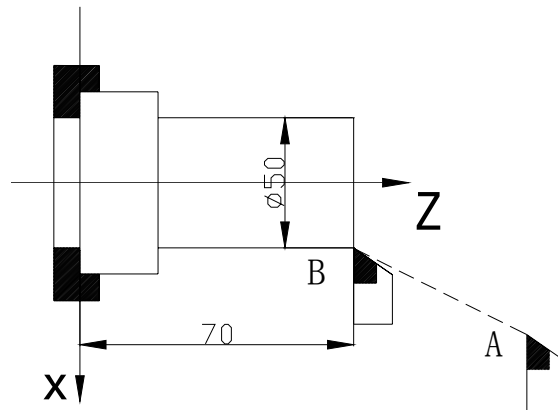


Fig.1-3 Absolute coordinate value

Tool traversing from A to B uses B coordinate values as follows: **X 50 Z 70**.

#### 1.6.2 Relative Coordinate Values

The relative coordinate value is the distance from the previous position to the next one, i.e. the actual traversing distance of the tool as Fig. 1-4:

II PROGRAMMING

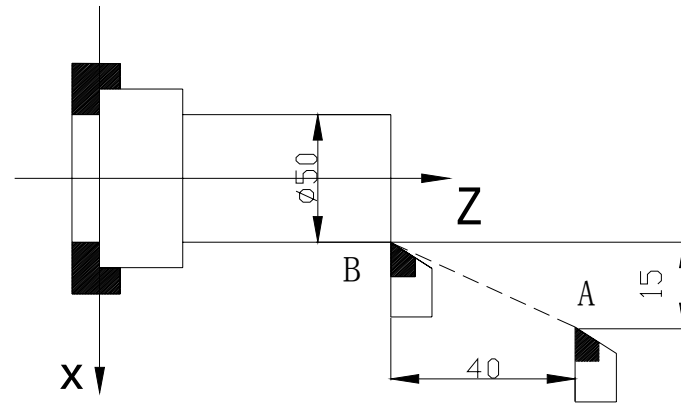


Fig.1-4 Incremental coordinate values

Tool traversing from A to B uses the incremental coordinates as follows: **U -30 W -40** (X with diameter programming)

### 1.6.3 Compound Coordinate Values

The relative coordinates and the absolute coordinates can be applied at the same time, but one coordinate axis in one block can only be defined by one method, i.e. **X , W** or **U , Z** can be applied, but the **X, U** or **Z, W** cannot be applied.

For tool traversing from A to B as Fig. 1-4, X is applied with the absolute coordinate and Z with the incremental coordinate as : **X 50 W -40**.

## 1.7 Diameter Programming and Radius Programming

X coordinate value is input with diameter or radius in programming: **diameter programming, radius programming.**

**Diameter programming:** when **P413\_d6** bit is set to 0, X command value is input with diameter, and X coordinate value is with diameter and its traverse speed is with radius in the system (program, parameter, tool offset, coordinates).

**Radius programming:** when **P413\_d6** bit is set to 1, X command value is input with radius, and X coordinate value is with radius and its traverse speed is with radius in the system (program, parameter, tool offset, coordinates).

#### 【Notes】

- 1) Diameter programming rules in the manual are described except for special explanations.
- 2) Arc radius is unrelated to diameter programming or radius programming.
- 3) Execute the tool setting and programming again after **P413\_d6** is modified, otherwise, X data is not correct.

## 1.8 Interpolation Function

**Interpolation** is defined as a planar or three dimensional contour formed by path of 2 or multiple axes moving at the same time, also called **contour control**. The axes controlled during interpolation are

called link axes. The moving distance, direction and speed of the link axes are controlled synchronously in the course of movement to form the required composite motion path. **Positioning control** is defined that only motion end point of one axis or multiple axes and only the motion path in the course of movement are controlled.

This system is a two-axis linkage CNC, with X and Z as its link axes. It provides includes linear, circular and thread interpolation functions .

Linear interpolation: Composite motion path of X, Z axis is a straight line from starting point to end point.

Circular interpolation: Composite motion path of X, Z axis is an arc with its radius specified by R (or by center by I, K) from starting point to end point; or an arc formed by coordinates I, K of a mediate point on the curve.

Thread interpolation: Move amount of X or Z or both X and Z is determined by the rotation angle of the spindle to form a helical cutting path on the workpiece surface (rotating along with the spindle) to realize the thread cutting.

## CHAPTER 2 PROGRAM CONFIGURATION

A group of commands edited according to the requirement of moving a machine is referred to as **program**. According to the sequence of command, the system can move the tool along a straight line or a circular arc, start/stop the spindle and cooling, or select a tool, etc. The command sequence is edited according to sequence required by the workpiece technology.

### 2.1 Character

Character is the basic unit to compose the program. The characters in the system are English letters, digits and some symbols.

An English letter is the address character of a command or data: **D E F G H I J K L M N P Q R S T U V W X Y Z r**

Digits are the specific data of an address character: **0, 1, 2, 3, 4, 5, 6, 7, 8, 9**

Macro word: **If then else and or = < > ()**

**If then else** : select statement; example: **If(x>y) then (z=0) else(z=1);**

**and: logic and ;**

**or: logic or ;**

**>: more than ;**

**<: less than;**

**=: equal to ;**

Sign: **% - + \* . 冫 /;**

**%** : Start sign of program name ;

**—** : Negative data or reduction;

**+** : Addition;

**\*** : Multiplication or modification;

**.** : Decimal point;

**/** : Division in assignment statement, and block skip sign in others;

**冫** : Blank space;

**;** : Comment.

### 2.2 Word

A word consists of an address character and the following digits. For example: N0100 X12.8 W-23.45. Rules are as follows:

- ◆ Each word must have an address character (an English letter) and the following number character string.
- ◆ The invalid 0 of digital character string can be omitted.

- ◆ The leading zero of a command can be omitted. For example: G00 can be written as G0.
- ◆ The positive sign must be omitted, but the negative sign must not be omitted.

## 2.3 Block Number

The block number is divided into two formats: the one with a line number and the one without a line number; block number is the line number of a block which is either automatically created or manually input and modified. A block number consists of the letter N and the following 4-digit integer (range: 0000-9999). The sequence of block number can be free, and its interval can also be unequal. It is suggested to increase the block number in ascending order for convenient program search and analysis.

When manual input is executed, block number N \* \* \* \* (\* is 0~9) can be directly input at the beginning of program line. When the integer following block number N needs to be modified, please modify the integer according to the block content modification mode described in Section 4.1 Edit Operation Mode in Part I Operation.

When automatic block number creation is selected, and parameter **P333** is not set to 0, the line number can be automatically created. If **P333** is not set to 0, the line number cannot be created automatically. Please refer to Section 4.6 Parameter Operation Mode in Part I Operation for parameter setting; set **P333** to 0; each program includes many blocks, and each block begins with block number "N\*\*\*\*". After a new program is created, the system automatically creates the first block number "N0000", and after each block is input and **ENTER** is pressed, the system automatically creates the next block number. The program number increment is determined by **P333**. The system automatically creates block numbers according to the increment which is 1/4 integer of **P333** content.

### 【Note】

- 1) When the system judges that current line has no line number, it will automatically create a new line number "N1000".
- 2) When commands M98, M97, M91, M92, M93, M94 related to block number are used, there should be no repetitive block number. Otherwise, the system will issue an alarm. A block number can be repeated without executing the above commands.

## 2.4 Block

A block with a line number is composed of a block number and several words; a block without a line number has no block number. One block can contain 255 characters at most (including spaces between words). The block number is indispensable for a block. It can be either created automatically by the system, or modified in EDIT state.

A block can contain several words or no words. When a block contain many words, one or more blank space must be input between words.



A complete block is as follows:

N0120 G1 X130 W-40 F50

N0120           Block number

G1               Preparatory function

X130 W-40      Motion data

F50             Motion speed

          Block end, not displayed on the screen. However, each program ends with **ENTER**.

#### 【Note】

- 1) Each word of a block is separated with a blank space generated automatically by the system; but it is necessary to input the blank space manually by user when this system cannot distinguish words.
- 2) A word can be placed anywhere in a block.

## 2.5 Block Skip Symbol and Comment

Insert skip symbol “/” or comment symbol “;” before the block which needs not to be executed (and not be deleted) during execution. Then the system skips this block in the program being executed.

When a comment symbol is added behind each block, the user can add comments for each block; the comments are edited by only English letters and digitals on CNC, and can be edited by Chinese on PC. The CNC can display Chinese comments after programs are downloaded to CNC.

The system skips the block with “/” at the line beginning to execute the next one.

The system skips the block with “;” at the line beginning or block beginning to execute the next one.

Input “;” at the end of a line and then input simple comments.

#### 【Note】

- 1) Press **W** key to insert “;” or “/”.
- 2) Block contents following “;” will become green.
- 3) Insert “/” in front of the head of the block, the block will turn into green.
- 4) Except for the first condition above, the block where the cursor is located becomes light green; and becomes orange in other conditions.
- 5) Chinese comments following “;” can be input only through serial port or USB instead of the system keyboard, but the system can display Chinese.
- 6) “;” must be input in half-width format, when comments are input through serial port or USB, because the system does not support full-width input format.

2.6 Program Structure

A **block** consists of a group of commands realizing one or several technology operations in the course of machining. A **machining program** (also referred to as **part program**) consists of multiple blocks arranged according to the machining technology. A **block number (line number)** is used for identifying different blocks, and a **program name** is used for identifying programs.

Each part program consists of one program number and some blocks (up to 9999 blocks). A block number is composed of letter N and the following 4-digit integer. A program name is comprised of sign % and a program number (3-digit integer). The general structure of a program is as Fig. 2-1.

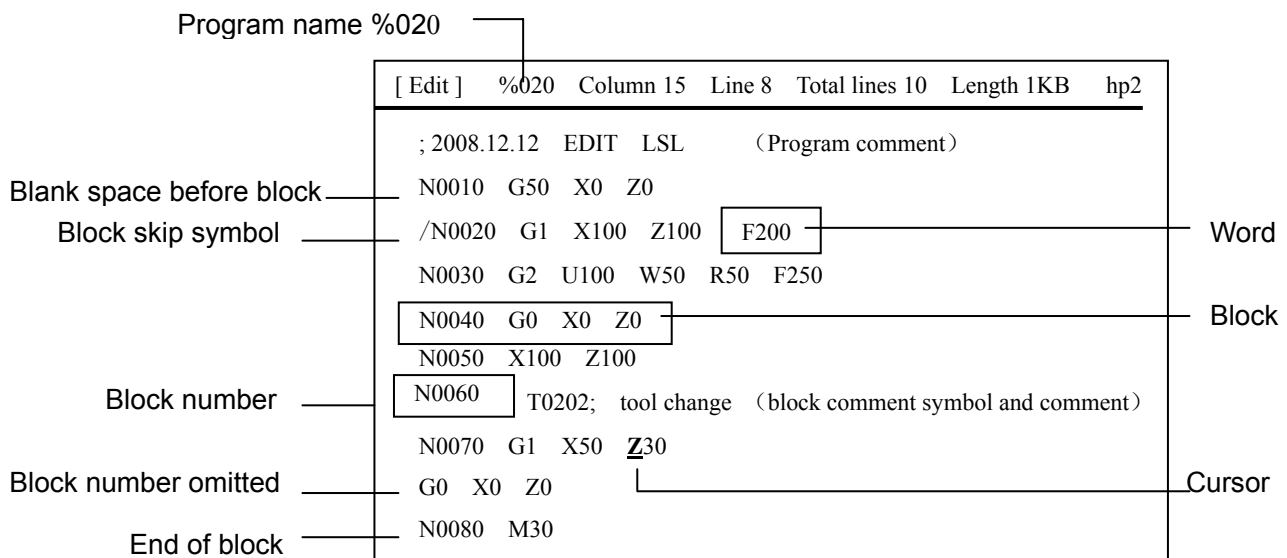


Fig. 2-1 General structure of program

According to the figure above, a program consists of program comments and blocks; each block begins with a block number (omissible), followed by characters and words; and there is a block skip symbol and comments in a block.

**【Example】**

Program (program name: %012)	Remarks
<pre> N0000 G50 X200 Z300 G00 X200 Z200 N0020 T11 N0030 ; G00 X100 Z100 /N0040 G02 X150 Z150 R50 N0030 G01 X100 Z100 ; 2008 N0050 M30                     </pre>	<p>Blocks containing N**** are the blocks with a line number, blocks not containing N**** are the blocks without a line number;</p> <p>Home of each line is a blank space;</p> <p>There is a blank space between line number and code for the program with line numbers;</p> <p>/ Block skip</p> <p>; The followings are comments.</p>

## CHAPTER 3 MSTF COMMANDS AND FUNCTIONS

This chapter introduces in details all **MSTF** command functions and their use in this CNC system.

### 3.1 M — Auxiliary Function (Command List)

M function, which consists of address symbol M and its following two-digit integer, is used to control the ON/OFF of some machine operations and execution sequence of the machine program. The used M functions in the system are as follows:

Command	Function	Programming format	Explanation
M00	Pause to wait for starting	M00	Press CYCLE START to restart
M02	End of program	M02	Return to No.1 block
M20	End of program	M20 L	Return to No. 1 block to execute cycle machine, L being cycle machine times
M30	End of program for spindle OFF and cooling OFF	M30	
M03	Spindle CW	M03	
M04	Spindle CCW	M04	
M05	Spindle OFF	M05	
M08	Cooling ON	M08	
M09	Cooling OFF	M09	
M10	Workpiece clamp	M10	
M11	Workpiece release	M11	
M12	Chuck output signal cancel	M12	
M32	Lubricating ON	M32	
M33	Lubricating OFF	M33	
M41	Spindle gearing No. 1	M41	
M42	Spindle gearing No. 2	M42	
M43	Spindle gearing No. 3	M43	
M44	Spindle gearing No. 4	M44	
M78	Tailstock forward	M78	
M79	Tailstock backward	M79	
M80	Tailstock output signal cancel	M80	
M96	Call cycle execution	M96 P Q L	
M97	Program transfer	M97 P	Transfer entry block number specified by P
M98	Subprogram call	M98 P L	
M99	Subprogram return	M99	M98/M99 call times specified by L
M21	Set No. 1 user output to be valid	M21 D	Output signal keeping for the time specified by D, and the signal is cancelled after the time is up.
M22	Set No. user output to be invalid	M22 D	
M23	Set No. 2 user output to be valid	M23 D	
M24	Set No. 2 user output to be invalid	M24 D	
M91	No. 1 user input is invalid until input is invalid.	M91 P	Transfer entry block number specified by P
M92	Wait when No. 1 user input is valid till the input is invalid	M92 P	
M93	Wait when No. 2 user input is valid till the input is invalid	M93 P	

M94	Wait when No. 2 user input is valid till the input is invalid	M94 P	
M60~M74	Customized commands	Refer to Chapter 8 Custom Command Programming in Part II Programming	
M81	Control according to input signal state	See Section 3.2 in Part II Programming for introduction to M81, M82 and M83.	
M82	Output control and detection		
M83	Control according to output signal state		

**【Note】**

- 1) The leading zero of M command can be omitted; Example: M00 can be abbreviated to M0, which is the same as M00.
- 2) Bit parameter **P403\_d3**, **P403\_d2**: When Emergency Stop button or Reset key is pressed, they are used to set the M function output state.
- 3) M commands which must be in a separate block without other commands are:  
M00, M02, M20, M30, M96, M97, M98, M99;  
M21, M22, M23, M24;  
M91, M92, M93, M94;  
M81, M82, M83;  
M60~M74 (can be in the same block with F, S command)

3.1.1 M00 — Pause

**【Command format】**

**M00** ; Program pause

**【Explanation】**

M00 suspends the program execution temporarily and the system prompts “PAUSE” in flash; the program execution is continued after **CYCLE START** is pressed.

Press **ESC** to exit from program running.

M00 function is different from that of **CYCLE START** key. M00 is used to perform pause before a specified block in advance, and **CYCLE START** is used to perform pause at random.

3.1.2 M02 — End of Program

**【Command format】**

**M02** ; End of program

**【Explanation】**

M02 indicates that the program ends, and the system returns the control to the first block to wait.

3.1.3 M20 — Program End Cycle Machining

**【Command format】**

**M20** L ; Return to the first block to cycle the machining, L is the cycle machining times, range: 1 ~ 9999.

**【Explanation】**

L is the cycle machine times (when L is equal to 3, the actual machined workpiece is 4PCS)

The system defaults the infinite cycle machine when L is omitted.

M20 indicates that the program ends and the system returns the control to the first block to execute repetitively; M20 is used when the system or machine is tested.

### 3.1.4 M30 — End of Program, Spindle OFF, Cooling OFF

**【Command format】**

**M30** ; Endo of program, spindle OFF, cooling OFF

**【Explanation】**

M30 indicates the program ends, spindle is OFF and cooling is OFF and the system returns the control to the first block to wait.

### 3.1.5 M03, M04, M05 —Spindle Control

**【Command format】**

**M03** ; Spindle CW

**M04** ; Spindle CCW

**M05** ; Spindle stop

**【Explanation】**

The system is in M05 state after power-on.

When M03/M04 is executed, M03/M04 output is valid and the spindle ON state remains unchanged. Turn OFF the spindle when M05 is executed. Cancel M03/M04 signal output during system emergency stop.

See Section 4.4.3.1 Spindle Start/Stop Control in Part I Operation for details.

### 3.1.6 M08, M09 —Coolant Control

**【Command format】**

**M08** ; Cooling ON

**M09** ; Cooling OFF

**【Explanation】**

The system is in M09 state upon power-on.

When M08 is executed, its output is valid, and the cooling is ON; when M09 is executed, M08 output is cancelled, and the cooling is OFF. M08 signal output is cancelled when the system is in the emergency stop state.

See Section 4.4.4 Coolant Control in Part I Operation for details.

### 3.1.7 M10, M11, M12 — Clamping/Releasing Chuck, Cancelling Chuck Output Signal

**【Command format】**

**M10** ; Clamping chuck

**M11** ; Releasing chuck

**M12** ; Cancelling chuck output signal; (only some special chuck devices can use M12)

**【Explanation】**

M11 output is valid while M10 output is invalid when the system is turned on.

Whether M10, M11 are valid, whether they require response detection, whether they are pulse control output or level control output, and whether they are in inner or outer chuck control mode can be set by parameters. Interlock between chuck and spindle is decided by **P402\_d5**: When **P402\_d5=0**, they are interlocked; when **P402\_d5=1**, their interlock is released.

See Section 4.4.7 “Hydraulic Chuck Control Function” in Part I Operation for details.

### 3.1.8 M32, M33 — Lubricating ON/OFF

**【Command format】**

**M32** ; Lubricating ON

**M33** ; Lubricating OFF

**【Explanation】**

M33 output is valid while M32 output is invalid upon power-on.

For details, see Section 4.4.9.2 Lubricating Control in Part I Operation.

### 3.1.9 M41, M42, M44, M43 — Spindle Automatic Gear Shift Control

**【Command format】**

**M41** ; Spindle (variable-frequency type) automatically shifts to No. 1 gear

**M42** ; Spindle (variable-frequency type) automatically shifts to No. 2 gear

**M43** ; Spindle (variable-frequency type) automatically shifts to No. 3 gear

**M44** ; Spindle (variable-frequency type) automatically shifts to No. 4 gear

**【Explanation】**

M41, M42, M43, M44 can be used only when the machine employs the variable-frequency spindle. See Section 4.4.3.3 Spindle S Command—Rotation Speed Control in Part I Operation for details.

### 3.1.10 M78, M79, M80 —Tailstock Advancing and Retracting, Tailstock Output Signal Cancelling

**【Command format】**

**M78** ; Tailstock advancing

**M79** ; Tailstock retracting

**M80** ; Cancelling tailstock output signal (only some special tailstock devices can use M80)

**【Explanation】**

Whether M78, M79 are valid, whether they require response detection, and whether they are pulse control output or level control output can be set by parameters. The tailstock is interlocked with the spindle. See Section 4.4.8 Hydraulic Tailstock Control Function in Part I Operation for details.

### 3.1.11 M96 —Calling Cycle Execution

**【Command format】**

**M96** P \* \* \* \* Q \* \* \* \* L \* \* \* \* ; Calling cycle execution

**【Word】**

P — Entry block number of called program. The leading zero of the block number can be omitted.

Q — Last block number of called program. The leading zero of the block number can be omitted.

L — Call times. The number of call times is 1 when L is omitted or is 1. L is within 1~9999 times.

**【Explanation】**

Program group called by M96 can contain M96, M98/M99, G22/G80, and allows nesting.

The embedded use of M96 and M97 can prevent the program returning to the block following M96 and then rerunning after M96 is executed.

**【Example】**

Method 1	Method 2
N0010 G00 X100 Z100	N0010 G00 X100 Z100
N0020 M96 P70 Q80 L3	N0020 M96 P40 Q50 L3
N0030 G01 W-5	N0030 M97 P0060
N0040 U5 W-5	N0040 G02 U5 W-5 R5 F300
N0050 W-5	N0050 G03 U5 W-5 R5
N0060 M30	N0060 G01 W-5
N0070 G02 U5 W-5 R5 F300	N0070 U5 W-5
N0080 G03 U5 W-5 R5	N0080 W-5
	N0090 M30

Method 1: After M96 specifies that the specified program is called three times, the cursor returns to N0030 and the subsequent blocks are executed till the program ends.

Method 2: After M96 specifies that the specified program is called three times, the cursor returns to N0030 and then the system uses M97 to continue executing the subsequent blocks till the program ends. The results of method 1 and method 2 are the same.

## 3.1.12 M97 — Program Transfer

## 【Command format】

**M97**    **P**    ; Program transfer

## 【Word】

**P** —Block number which is transferred to. The leading zero of the block number can be omitted.

## 【Explanation】

M97 command causes the program to transfer from the current block to the block specified by P and then continues the execution. Block number specified by P should exist in the program. Otherwise, the program issues the alarm "E215: Line number missing".

The block number specified by P cannot be M97 block.

Prevent the endless loop when using M97 command.

## 【Example】

```
N0030 G00 X100
N0040 M98 P0060
N0050 M97 P0090
N0060 G01 U2
N0070 W-5
N0080 M99
N0090 M02
```

When executing N0050, the program directly executes N0090 instead of N0060.

## 3.1.13 M98, M99 — Subprogram Call and Subprogram Return

## 【Command format】

**M98**    **P** \* \* \* \*    **L** \* \*

**M99**

## 【Word】

**P** — Block number where the subprogram is located. The leading zero of the block number can be omitted.

**L** — Call times of subprogram. The subprogram is called once if L is omitted or is 1. L is within 1~9999.

## 【Explanation】

If a fixed sequence which appears repetitively in the program, the sequence can be taken as a subprogram, so it can be called instead of being compiled when it is needed again.

When M98 calls the subprogram and there is M99 in the execution of the subprogram, the subprogram call ends and the program returns to the main program and proceeds the block following the subprogram block.



**【Note】**

- 1) M98 should be used together with M99. When M98 is used alone, the subprogram can only be called once instead of L (L > 1) times.
- 2) When the program has M99 without M98, the program ends after executing M99.
- 3) M98/M99 can be contained in the subprogram call, i.e. they can be nested; up to 18 levels of M98/M99 subprogram call can be nested.
- 4) The subprogram is usually behind M02/M30 of the main program, and the last block of this program must be M99 (subprogram return). If the subprogram is not behind the main program, M97 command should be used for program transfer. (see the following example)

**【Example】**

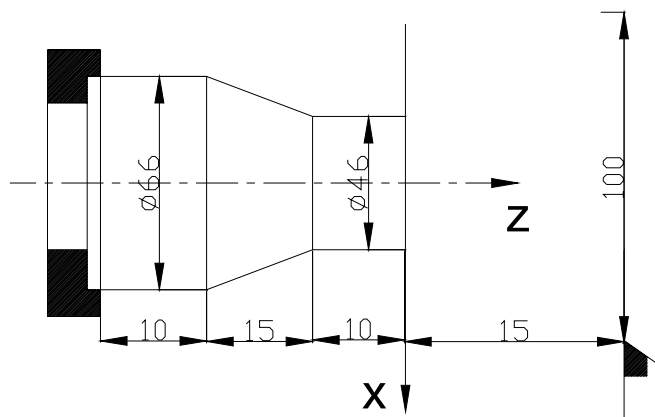


Fig. 3-1 M98 M99 subprogram call cutting example

Two programming methods as Fig.3-1:

Method 1		Method 2		
N0010 G00 X100 Z50		N0010 G00 X100 Z50		
N0020 M03 S01		N0020 M03 S01		
N0030 G0 X50 Z15 F500		N0030 G0 X66 Z15 F500		
N0040 M98 P0060 L5	Call subprogram	N0040 M98 P0080 L5	Call subprogram	
N0050 M97 P0130	Transfer program	N0050 G0 X100 Z50		
N0060 G0 U-4	Subprogram is inserted in the main program	N0060 M05	Subprogram follows the main program	
N0070 G01 Z-10 F80		N0070 M02		End of main program
N0080 U20 Z-25		N0080 G0 U-4		
N0090 Z-35		N0090 G01 Z-10 F80		
N0100 G0 U2 Z1		N0100 U20 Z-25		
N0110 U-22	N0110 Z-35			
N0120 M99	N0120 G0 U2 Z1			
N0130 G0 X100 Z50	N0130 U-22			
N0140 M05	N0140 M99			
N0140 M02	End of main program			
Transfer the program with M97. Otherwise, the program does not accord to the actual machining requirement.		The subprogram must follow the main program without M97		

Method 1: When the system executes N0040, it calls the subprogram and executes N0060—N0120 five times. Then it executes N0050, and the program skips to N0130 and executes the following blocks.

Method 2: When the system executes N0040, it calls the subprogram and executes N0080—N0140 five times, and then executes blocks from N0050 to N0070, and so the program ends.

## 3.1.14 M21, M22, M23, M24 —User Output Control

## 【Command format】

<b>M21</b>	<b>D</b>	; The output of No.1 user output signal is valid (output low level)
<b>M22</b>	<b>D</b>	; The output of No.1 user output signal is invalid (output OFF)
<b>M23</b>	<b>D</b>	; The output of No.2 user output signal is valid (output low level)
<b>M24</b>	<b>D</b>	; The output of No.2 user output signal is invalid (output OFF)

## 【Word】

D — Signal hold time. (unit: s; range: 0~9999.999). When D is omitted, the output signal is always being held.

## 【Explanation】

M21, M22, M23, M24 have no concrete definitions specified the system, so the user can specify their definitions according to the specific requirements. M21, M22, M23, M24 separately correspond to two user-defined output points. By using corresponding commands, the states of output signals can be changed.

User output point 1 and 2 are defined in the interface parameters (**M210**, **M230**). See Part III Connection after defining the output interfaces.

## 【Note】

- 1) M21, M22, M23 and M24 cannot be in the same block with other commands. Each of them must be specified in a separate block.
- 2) M21~M24 can contain parameter D: when M21~M24 are executed with parameter D, the time specified by D is delayed after corresponding output is executed, and then the previous output is cancelled. If the command has no D, the output signal is being kept.

## 3.1.15 M91, M92, M93, M94 — User Input Control

## 【Command format】

<b>M91</b>	<b>P</b>	;
<b>M92</b>	<b>P</b>	;
<b>M93</b>	<b>P</b>	;
<b>M94</b>	<b>P</b>	;

## 【Word】

P—Block number of the target block which is jumped to. It does not skip when P is omitted. The leading zero of the block number specified by P can be omitted.

## 【Explanation】

The definitions of input signals corresponding to M91, M92, M93 and M94 are not determined, so the user can define them according to the requirements. M91, M92, M93, M94 separately correspond to two user-defined input points. User input point 1 and 2 are defined in the interface parameters (**M911**, **M931**). See Part III Connection after defining the output interfaces.

When P is omitted, and the state of the check input point meets the command requirement, the system orderly executes the next block, and it keeps waiting when the requirement is not met.

When P≠0, the system skips to the block specified by P when the input point state meets the command requirement, otherwise orderly executes the next block.

When P is omitted:

- M91 checks the state of No. 1 user input. When the state is valid (the input terminal is connected to 0V), the systems waits till the input is invalid.
- M92 checks the state of No. 1 user input. When the state is invalid (the input terminal is disconnected to 0V), the system waits till the input is valid.
- M93 checks the state of No. 2 user input. When the state is valid (the input terminal is connected to 0V), the system waits till the input is invalid.
- M94 checks the state of No. 2 user input. When the state is invalid (the input terminal is disconnected to 0V), the system waits till the input is valid.

When P≠0:

- M91 checks the state of No. 1 user input. When the state is valid (input terminal is connected to 0V), the system skips to the block specified by P. Otherwise the next block is executed.
- M92 checks the state of No. 1 user input. When the state is invalid (input terminal is disconnected with 0V), the system skips to the block specified by P; otherwise the next block is executed.
- M93 checks the state of No. 2 user input. When the state is valid (input terminal is connected to 0V), the system skips to the block specified by P; otherwise the next block is executed.
- M94 checks the state of No. 2 user input. When the state is invalid (input terminal is disconnected to 0V), the system skips to the block specified by P; otherwise the next block is executed.

### 3.1.16 M60~M74 — Custom Command

【Command format】

```
M60 ;
...
M74 ;
```

【Explanation】

See Chapter 8 Customized Command Programming in Part II Programming for details.

### 3.2 M81, M82, M83—User Input/Output Condition Control

M81, M82, M83 are a group of condition control commands based on input/output signal state.

Representations of signal point, level and holding time are: for example: I8.1, Q17.0, R18.0, D5, etc.

- 1) Letters I, Q, R represent the input signal, output signal and output signal state respectively (two states: when 0 is output, the external can form a conducted circuit; when 1 is output, it means high impedance state, and the external cannot form a conducted circuit.); D represents the signal holding time.
- 2) The integral part after letters represents signal points, corresponding to the pin of input signal UIxx or output signal UOxx pin; the range is 1~32 point.
- 3) .0 behind the decimal point represents low level (0, which is the system default, can be omitted); .1 represents high level.
- 4) D: Signal holding time; (unit: s range: 0~9999.999); when D is omitted, the output signal remains.
- 5) Relevant parameter: Bit parameter **P412\_d0**. M81, M82, M83 are affected by this parameter.

### 3.2.1 M82— Output Control and Detection

**【Code format. Example】**

- M82 Q17.0** ; UO17 pin outputs low level (conducted)
- M82 Q17.1** ; UO17 pin outputs high level (high-impedance)
- M82 Q17.0 D3** ; UO17 pin outputs low level, and the output is cancelled after delaying time D.
- M82 Q17.0 I8.0** ; UO17 outputs low level, and the next is executed after detecting UI18 pin is at low level;
- M82 Q17.0 I8.0 D3** ; UO17 pin outputs low level, after delay time D, the output is cancelled; the next is executed after detecting UI08 pin is low level;

**【Note. Example】**

M81, M83 cannot be executed in MANUAL mode, and M82 can be executed in following two formats:

**M82 Q17.0 D3 or M82 Q17.0**

For example: **M82 Q17.0 D5** ; UO17 pin outputs low level (conducted), then the output is cancelled after 5 seconds delay. In MANUAL mode, it can be simply input as M Q 1 7 D 5.

### 3.2.2 M81—Control According to Input Signal State

**【Code format. Example】**

- M81 I8.0** ; If the input UI08 pin is at low level, the system proceeds to the next command; otherwise, it keeps waiting.
- M81 I8.1** ; If the input UI08 pin is at high level, the system proceeds to the next command; otherwise, it keeps waiting.
- M81 I8.0 P1000** ; If the input UI08 pin is at low level, the system transfers to P1000; otherwise, it proceeds to the next command.
- M81 I8.0 Q17.0** ; If UI08=0, then, UO17 outputs 0, and the system proceeds to the next command; otherwise, it keeps waiting.
- M81 I8.0 Q17.0 D3** ; If UI08=8, UO17 outputs 0, the signal holds for D3, and the system proceeds to next command; otherwise, it keeps waiting

### 3.2.3 M83—Control According to Output Signal State

#### 【Code format. Example】

**M83 R8.0** ; If the input UO8 pin is at low level, the system proceeds to the next command; otherwise, it keeps waiting.

**M83 R8.0 P1000** ; If the input UO8 pin is at low level, the system transfers to P1000; otherwise, it goes to the next command.

**M83 R8.0 Q17.0** ; If UO8 pin is at low level, UO17 pin outputs 0, and system proceeds to the next command; otherwise, it keeps waiting.

**M83 R8.0 Q17.0 D3** ; If UO8 pin is at low level, UO17 outputs 0, the signal holds D3, and the system proceeds to the next command; otherwise, it keeps waiting.

### 3.3 S function — Spindle Function

S and the following data are used to control the spindle speed and there are two modes as follows:

- 1) Spindle speed switch value control mode: Sx or Sxx, the system outputs the gear signal to the machine to realize spindle speed step change.
- 2) Spindle speed analog voltage control mode: Sxxxx specifies the actual speed, and the system outputs 0~10V analog voltage signal to the spindle servo device or converter to realize the spindle speed stepless change.

According to the machine configuration, whether the spindle function is used for controlling multi-speed or variable-frequency motor is set by bit parameter **P410\_d6**.

#### 3.3.1 Spindle Motor Controlled by Gear Shift

##### 【Command format】

**S01 ~ S04** ; 4-gear direct outputs, leading zero can be omitted.

**S00 ~ S15** ; 16-gear BCD code outputs, leading zero can be omitted.

##### 【Explanation】

When **P410\_d6=0**, S function controls the multi-speed spindle motor; when it controls the multi-speed spindle motor, whether to directly output 4-gear control signal or output 16-gear BCD code signal can be selected.

**P410\_d5=0**: Spindle control is the 4-gear direct control output S01~S04, each gear corresponds to one output point;

**P410\_d5=1** Spindle control is 16-gear BCD code output S00~S15.

## 3.3.2 Variable-frequency Motor Controlled by Speed

**P410\_d6=1**: S function controls the variable-frequency motor. The system outputs 0~10V DC analog voltage signal to control the converter to realize the stepless speed regulating for the spindle motor.

## ◆ Gear control of variable-frequency spindle

## 【Command format】

M41 ; Corresponds to output point S01  
 M42 ; Corresponds to output point S02  
 M43 ; Corresponds to output point S03  
 M44 ; Corresponds to output point S04

## ◆ Speed control of variable-frequency motor

## 【Command format】

G96 ; Setting constant surface cutting state;  
 G96 S\_\_ ; Setting constant surface cutting state, and specifying the surface speed value;  
 range: 0~9999 m /min;  
 G97 ; Cancelling constant surface speed state;  
 G97 S\_\_ ; Cancelling constant surface speed state, specifying speed value;  
 range: 0~9999 r /min  
 S\_\_ ; According to the current state; speed value or surface speed value cutting.

## 【Explanation】

- 1) In JOG and AUTO operation mode, the screen displays the actual spindle speed S\_.
- 2) In JOG and AUTO operation mode, the screen displays the cutting constant surface speed control G96 S\_\_m, or constant speed control G97 S\_\_r.
- 3) Parameters related to the spindle speed analog voltage control are as follows:
  - Data parameter **P305**: Max. spindle speed of constant surface speed;
  - Data parameter **P304**: Min. spindle speed of constant surface speed;
  - Data parameter **P300~P303**: Max. spindle speed in the spindle gear 1~4 (corresponding to M41~M44).
- 4) See Section 4.4.3.3 Spindle S Command—Rotation Speed Control in Part I Operation for details.

### 3.4 T Function — Tool Function

Generally, machining one workpiece requires several different tools. This system can control the motor tool post with 4~16 stations. The position of each tool is different when it is at the cutting position because of the installation error or abrasion. To prevent the programming being affected by these factors, the tool change and tool compensation function are provided.

Tool setting operation is executed before machining to obtain the position offset data of each tool (referred to as **tool offset**). After T command is executed in program execution, the system automatically executes the tool offset. Therefore, each tool is programmed according to the workpiece drawing dimension without considering the position relationship among the tools in the machine coordinate system.

If the deviation of machining dimension caused by tool wear occurs, the user can modify the tool offset according to the dimension deviation.

#### 【Command format】

- Txx**     \_\_\_    The first digit is the tool number; the second one is the tool offset number;  
**Txxx**    \_\_\_    The first digit is the tool number, the last 2 digits is the tool offset number;  
**Txxxx**   \_\_\_    The first digit is the tool number and the last two digits is the tool offset number.

#### 【Word】

- 1) Tool number range: Determined by **P319** (maximum tool number: 1~16); E.g. if **P319** is 4, the tool number is 0~4; if the input tool number is 0, it means keeping the current tool number.
- 2) Range of tool offset number: 0~64; if the input tool offset number is 0, it means cancelling the tool compensation.

#### 【Explanation】

- 1) The system can select up to 16 tools and **P319** sets the maximum tool number.
- 2) When the system executes the manual tool setting operation, it automatically matches the tool offset number to the tool number and saves it to the tool compensation table. When **P403\_d4** is set to 1, the system automatically identifies the relationship between tool setting record and tool number and tool offset number of the tool change command in the program; if inconsistency occurs, the system issues the Program Compound Check Alarm but does not lock the program execution.
- 3) The system executes the tool setting operation based on the trial cutting mode or fixed point mode in Jog operation mode. See Section 4.4.6 Manual Tool Setting in Part I Operation.

#### 3.4.1 Tool Offset Execution Mode-Moving Slide Carriage

When executing T command, the system first changes the tool to the tool number specified by T, then executes the tool compensation of T command. In such a case, the system modifies the tool nose coordinates. The system uses two methods according to the parameters:

- P403\_d6 = 0:** After the system executes the tool compensation and modifies the tool nose coordinates, it does not move the slide carriage, i.e. the execution is completed;

**P403\_d6 =1:** After the system executes the tool compensation and modifies the tool nose coordinates, it moves the slide carriage to recover the tool nose coordinates (move to the position before modification, making the tool nose coordinates unchanged).

**Move the machine slide carriage:**

- 1) Move the carriage in JOG operation mode to recover the tool nose coordinates. X, Y axes rapidly traverse simultaneously.
- 2) In AUTO operation mode, when T command is a separate block, its execution mode is the same as that in JOG operation mode, i.e. the system moves the carriage to recover the tool nose coordinates. X, Y axes rapidly traverse simultaneously.
- 3) In AUTO operation mode, when T and G00/G01 are in the same block, the system executes the tool change first, and then adds the tool compensation value to X, Z coordinates of G00/G01 for movement. The system performs the rapid traverse in G00 and cutting speed in G01.

### 3.4.2 Tool Offset Execution Mode- Modifying Coordinates

When **P403\_d6=0**, the tool compensation execution mode is to modify the system coordinates.

In JOG and AUTO operation modes, the slide carriage does not move after the system executes the tool compensation and modifies the tool nose coordinates.

In AUTO operation mode, after the system executes the tool compensation, the workpiece program is in undefined state if the tool nose coordinates are changed and not recovered. The solutions are as follows when the system executes different subsequent commands:

- 1) When the subsequent command is G00 Z/X absolute coordinate positioning, the tool directly traverses to G00 positioning point;
- 2) When the subsequent command is G00 W/U relative coordinate movement, the tool directly traverses one relative amount;
- 3) When the subsequent command is G00 single-axis positioning, the other axis does not move.

#### 【Note】

- 1) Correct programming method: After the system executes the tool change command, use G00 to perform Z/X double-axis re-positioning, or use G00 single-axis repositioning in order. Prevent unexpected alarms during machining.
- 2) To improve the machining efficiency, T command and G0 must be in the same block. For example, `G0 X100 Z200 T0202`.
- 3) Adding the tool compensation to program command value is available only when G0/G1 and the tool compensation command are in the same block.
- 4) For G0/G1 in the compound cycle, no T command is allowed in the same block.

### 3.5 F Function — Feedrate Function

It defines the tool cutting feedrate. I.e. the feedrate function (G98/G99, F command).



## 【Command format】

**G98** F \* \* \* \* . \* \* \* ; Feed per minute

**G99** F \* \* \* \* . \* \* \* ; Feed per rev

## 【Explanation】

**Cutting feed:** The system can simultaneously control X, Z motion, making the tool motion path and the path defined by commands (straight line, arc) consistent, and the instantaneous speed in the tangent direction of motion path and F word consistent. This kind of motion control is called **cutting feed** or **interpolation**. The cutting feedrate is specified by word F. When the system executes the interpolation command (cutting feed), it divides the cutting feedrate specified by F into X, Z direction according to the programming path, and simultaneously controls X, Z instantaneous speed to make the vector resultant speed of speed in the two directions equal to F command value.

$$f_x = \frac{d_x}{\sqrt{d_x^2 + d_z^2}} \bullet F$$

$$f_z = \frac{d_z}{\sqrt{d_x^2 + d_z^2}} \bullet F$$

F is the vector resultant speed of instantaneous speed in X/Y direction;  
 $d_x$  is the instantaneous increment of X axis,  
 $f_x$  is the instantaneous speed in X direction; the speed of X axis is the speed on the radius.  
 $d_z$  is the instantaneous (increment of Z axis,  
 $f_z$  is the instantaneous speed of Z axis.

The command determines the cutting feedrate of the tool.

Feedrate function in feed per minute (G98) is expressed with F \* \* \* \* . \* \* \* . (range: 0~4000.000 unit: mm/min)

Feedrate function in feed per rev (G99) is expressed with F \* \* \* \* . \* \* \* . (range: 0~4000.000 unit: mm/r)

F is a modal value. Once specified, it is not necessary to input it again. It is the feed per minute (G98) state upon power-on, and the tool actual traverse rate is controlled by F and feedrate override.

Tool actual cutting speed = Fx feedrate override (feed per minute)

Tool actual cutting speed = Fx spindle speed x feedrate override (feed per rev)

Reduction formula between feed per rev and feed per min:  $F_m = F_r \times S$

$F_m$ : Feed per min (mm/min);

$F_r$ : Feed per rev (mm/r);

S: Spindle speed (r/min)

F value keeps unchanged after F command is executed, and is not changed during system reset and emergency stop. The system provides 16-level feedrate override (0~150%, increment of 10%). In MANUAL and AUTO operation modes, the feed override keys on the system panel can regulate the feedrate in real time, and the actual feedrate can be regulated within 0~150% of the command speed. In addition, the feedrate override is memorized after power-off. For details about feedrate override regulation, refer to Section 4.4.1.5 Selecting Speed for Low-Speed Feed in Part I Operation.

**【Relative parameters】**

- 1) System parameter **P112**: Initial speed in cutting feed.
- 2) System parameter **P113**: Max. speed limit of cutting feed.
- 3) System parameter **P114**: Linear acceleration/deceleration of cutting feed.
- 4) System parameter **P115**: Exponential acceleration/deceleration of cutting feed.
- 5) System parameter **P118**: G99 initial lines.

**【Note】**

Uneven cutting feedrate occurs when the spindle speed is lower than 1 r/min (in G99 modal state); and following error occurs in the actual cutting feedrate when there is fluctuation in the spindle speed. To ensure the machining quality, it is recommended that the selected spindle speed should not be lower than min. speed of the effective torque output by the spindle servo or the converter.

## CHAPTER FOUR G COMMANDS AND FUNCTIONS

Here describes the functions and use of the G commands in the system

**Note: The Y command function is invalid for GSK928TD.**

### 4.1 G00 — Rapid Positioning G00

The tool rapidly traverses to the specified position in G00.

#### 【Command format】

**G00 Z (W)\_ X (U)\_** ; Two axes rapid traverse  
**G00 Z (W)\_** ; Z independent rapid traverse  
**G00 X (U)\_** ; X independent rapid traverse

#### 【Word】

X, Z: Absolute coordinates of end point;

U, W: Relative movement from starting point to end point;

Use relative or absolute coordinates; the coordinates not to be moved can be omitted; G00 without coordinate words cannot be specified.

#### 【Word range】

X, Z, U, W: (-9999.999~9999.999) mm

#### 【Explanation】

Z/X can select single-axis movement and simultaneous two-axis movement.

If the tool change command and tool offset command is in the same block with G00, the tool offset value is added to G00 movement value, in order to improve the work efficiency. In general, the tool change and tool offset commands should be specified in the same block with G00.

G00 actual run speed is controlled by the rapid override. For example, when G00 rapidly traverses in the independent movement mode, the actual speed is set by **P100, P101**.

$$Z \text{ actual rapid speed} = P100 \times \text{rapid override}$$

$$X \text{ actual rapid speed} = P101 \times \text{rapid override}$$

The actual max. speed of the machine depends on its actual condition and matched motor. For particular parameters, please see the manual issued by the machine tool builder. G00 is a modal command, which can be omitted in the next block in which it is used. G00 can be abbreviated to G0, and G0 and G00 are equivalent.

#### 【Relative parameters】

Parameters related to G00: **P100, P101, P103, P104, P106, P107, P112, P114, P400\_d3**.

Each axis can move at its respective rapid traverse rate or simultaneously rapidly move according to the proportion, and their movement mode is defined by **P400\_d3**: **P400\_d3=0**, each axis performs positioning at rapid traverse rate separately (independent movement mode) in G00; **P400\_d3=1**, each axis performs positioning rapidly and simultaneously (interpolation movement mode) in the proportion mode in G00.

**When P400\_d3=0**, the independent movement mode is executed as follows: (an example of Z axis)

- ① Speed raising stage: Z axis raises the speed at the initial speed of **P103**;
- ② Raise the speed to max. speed set by **P100**; the acceleration time in the speed raising stage is **P106**.

When each axis moves at the rapid speed, the actual speed displayed by the system is the resultant rapid traverse speed; when two axes or more move simultaneously, the actual rapid traverse speed displayed will be greater than the values set in **P100**, **P101** and **P102**.

**P400\_d3=1**: interpolation movement mode is executed as follows:

- ① Raising speed stage: Raising the speed at the initial speed of **P112**;
- ② The acceleration time in the speed raising phase is **P114**.
- ③ Here, the traverse speed is the resultant speed of the system. The movement axes rapidly perform positioning according to the proportion, and their actual speeds are controlled by **P100**, **P101**.

**【Note】**

Ensure the tool is placed at a safe position to prevent the tools collision when G00 is executed.

**【Example】**

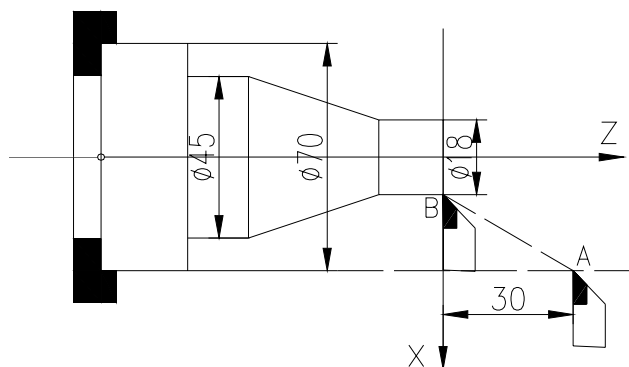


Fig. 4-1 G00 rapid positioning

As Fig.4-1, the tool traverses from A to B using G00 mode.

Absolute programming: N0010 G00 X18 Z0 ;

Relative programming: N0100 G00 U-52 W-30 ;

## 4.2 G01 — Linear Interpolation

## 【Command format】

**G01 Z (W)\_ X (U)\_ F\_** ; Z/X two-axis simultaneous feed  
**Z (W)\_ F\_** ; Z single-axis feed  
**G01 X (U)\_ F\_** ; X single-axis feed

G01 command causes the tool to move along the line from current point to X (U) , Z (W) specified point at the set speed, and to reach the specified end point simultaneously.

## 【Word】

X, Z: Absolute coordinates of end point;

U, W: Relative move amount from starting point to end point.

F: Cutting feedrate. F value is modal, which can be omitted when it is not changed. At the same time, it is controlled by the feedrate override.

## 【Word range】

X, Z, U, W: (-9999.999~9999.999) mm

F: 0.001mm/min~4000mm/min

## 【Explanation】

Z/X can select single-axis feed and two-axis simultaneous feed.

The traverse speed of G01 is specified by F and controlled by the feedrate override.

$$\text{Actual feedrate} = F \times \text{feedrate override}$$

G01 is a modal command which can be omitted in the next block if used. G01 can be abbreviated to G1, and G1 and G01 are equivalent.

## 【Relative parameters】

Parameters related to G01: **P112, P113, P114, P401\_d5, P401\_d4**

Interpolation movement execution is as follows:

- ① Raising speed stage: Raising speed at the initial speed of **P112**;
- ② The acceleration time of raising speed stage is **P114**; at the same time, the system checks whether the feedrate (F× feedrate override) exceeds **P113** limit. If it does, the feedrate is P113.

**P401\_d5, P401\_d4** set in the cutting machining: Whether the system uses high-speed connection mode, and whether uses continuous smooth transition or deceleration to zero.

## 【Example】

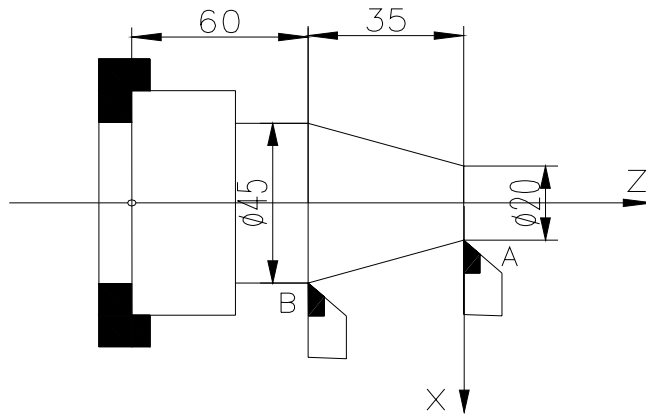


Fig. 4-2 G01 linear interpolation

The tool traverses from A to B in G01 at the speed of 150 mm/min, as Fig. 4-2:

Absolute programming: N0100 G01 X45 Z-35 F150 ;

Relative programming: N0100 G01 U25 W-35 F150 ;

## 【Note】

F cutting feed command in the examples of this chapter is executed with feed per minute (G98) except for special explanations; F word can be 0 during programming; when it is 0, and the program is executed in AUTO operation mode, the system issues an alarm: Feedrate is zero.

## 4.3 G02, G03, G05 —Circular Interpolation

## 【Command format】

**G02** Z (W)\_ X (U)\_ I\_ K\_ F\_ ; Circle center coordinate programming, CCW arc

**G02** Z (W)\_ X (U)\_ R\_ F\_ ; Arc radius programming, CCW arc

**G03** Z (W)\_ X (U)\_ I\_ K\_ F\_ ; Circle center coordinate programming, CW arc

**G03** Z (W)\_ X (U)\_ R\_ F\_ ; Arc radius program, CW arc

**G05** Z (W)\_ X (U)\_ I\_ K\_ F\_ ; Arbitrary point (I, K) on arc programming

The **G02**, **G03**, **G05** causes the tool to traverse along the arc from the current point to X(U), Z(W) specified point at the set speed. The arc radius is R or the vector from circle center to starting point is I, K (G02, G03), or arbitrary point I, K (G05) on the arc.

## 【Word】

X, Z: Absolute coordinates of end point;

U, W: Relative move amount from starting point to end point;

F: Cutting feedrate;

R: Arc radius;

K: It is Z different value between circle center and starting point of arc in G02/G03;

It is Z coordinate value of an arbitrary point on the arc in G05;

I: It is X different value between circle center and starting point of arc in G02/G03;

It is X coordinate value of an arbitrary point on the arc in G05.

**【Word range】**

X, Z, U, W: -9999.999 mm~9999.999mm

I, K: -1000000.000 mm~1000000.000mm in G02, G03; -9999.999 mm~9999.999mm in G05

F: 0.001 mm/min~4000mm/min

R:  $0 < R \leq 100000.000$ mm

**【Explanation】**

**G02, G03, G05** can only be Z/X programming; the traverse rate is set by F and controlled by the feedrate override.

F value is modal, which can be omitted when it is not changed. Meanwhile, it is controlled by the feedrate override.

$$\text{Actual feedrate} = F \times \text{feedrate override}$$

**G02, G03** and **G05** are modal commands, which can be omitted when used in the next block. G02 can be abbreviated to G2 and they are equivalent; G03 can be abbreviated to G3 and they are equivalent; G05 can be abbreviated to G5 and they are equivalent. The words are explained as follows:

Table 4-1

Word	Specified content	Meaning
G02	Arc rotation direction	CCW arc in front tool post coordinate system/ CW arc in rear tool post coordinate system
G03	Arc rotation direction	CW arc in front tool post coordinate system/ CCW arc in rear tool post coordinate system
X,Z	Absolute coordinate	Absolute coordinates of arc end point
U,W	Relative coordinate	Distance from starting point to end point of arc
I,K (G02,G03)	Circle center coordinates	Vector from circle center to starting point of arc in G02 ,G03
R	Arc radius	Distance between any point and circle center
F	Feedrate	Arc cutting speed
I,K (G05)	An intermediate point on arc	An arbitrary point on arc ( three points determine one circle)

G02, G03 cause the tool to move along the specific arc path at the set speed. G02/G03 arc direction definition: Taking the circle center as the reference point, their directions are reverse in the front/rear tool post coordinate system as follows:

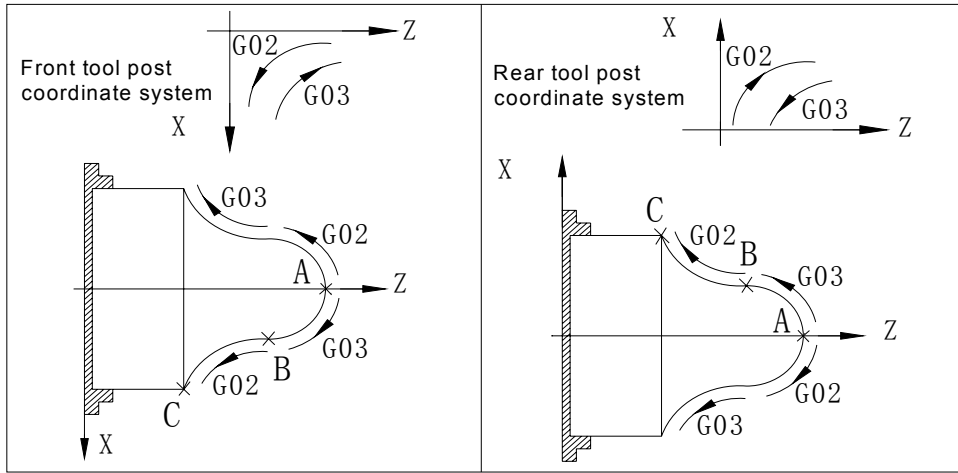


Fig. 4-3 G02/G03 arc direction definition

X and Z or U and W specify the end point of arc. The end point is presented with either absolute or relative coordinates. The relative coordinate is the distance from starting point to end point of arc.

For G02/G03: I, K specify the circle center coordinates of arc. I and K respectively correspond to X and Z vector from starting point to the circle center. I is X vector component (in diameter), K is Z component. I, K are positive when their directions are the same those of X, Z. Otherwise, they are negative. Use radius R programming when I, K programming is not used. See Fig. 4-4:

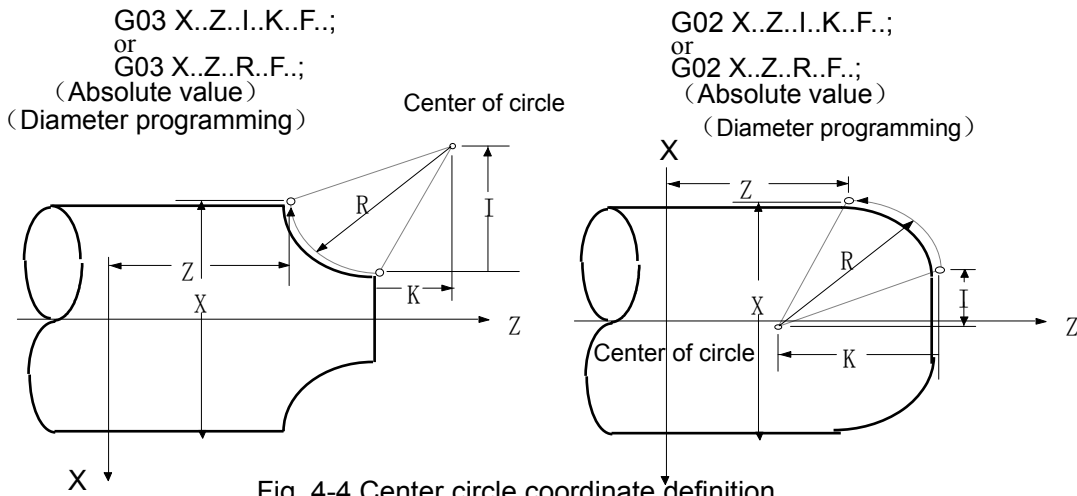


Fig. 4-4 Center circle coordinate definition

G05: I, K specify the coordinates of an intermediate point on the arc. Three points determine one circle: Starting point, end point and an intermediate point on arc as Fig. 4-5:

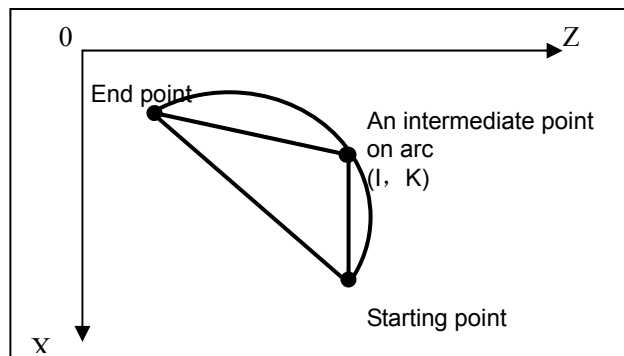


Fig. 4-5 Definition of coordinates of an intermediate point on arc



**【Related parameters】**

Parameters related to arc interpolation: **P112, P113, P114, P401\_d4, P400\_d2.**

Interpolation movement execution is as follows:

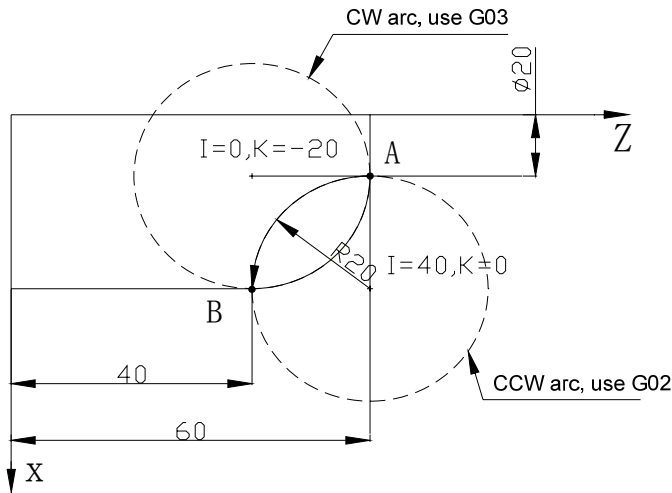
- ① Raising speed stage: Raising the speed at the initial speed of **P112**;
- ② The acceleration time of raising speed stage is **P114**; meanwhile, the system checks whether the feedrate (F x feedrate override) exceeds **P113** limit, if it does, the feedrate is **P113**.

**P401\_d4, P400\_d2** set in the cutting machining: Whether to use continuous smooth transition or deceleration to zero, whether to use front/post acceleration/deceleration, and whether to use arc crossing vertex precision execution.

**【Note】**

- 1) I value is expressed with the diameter. The value calculated according to the drawing is generally in radius. It should be multiplied by 2 in programming.
- 2) Max. arc radius is not more than 100000mm. Otherwise, it may be incorrect in the system.
- 3) For the arc with very small radius R, when the value of programming speed x feedrate is too large, the step angle in interpolation should not be more than 1.5 degree. Otherwise, the system automatically decelerates and there is the difference value between the actual feedrate and programmed speed.
- 4) The system automatically decelerates when the programmed speed is too high in machining small arc.
- 5) When using K, I programming in G02, G03, the system checks the current coordinates (starting point), end point coordinates and circle center coordinates; when the end point is not on the circle, and Z deviation is more than 0.015mm or X deviation is more than 0.03mm mm, the system issues: E238 Distances between circle center to two points are not equal.
- 6) when using G05 programming, the system checks the current coordinates (starting point), end point coordinates and coordinates of an intermediate point on the arc; the system issues the alarm: E239 Three points are in one line and cannot consist of an arc when the three points cannot form a triangle. However, when the specific intermediate point, arc start point and end point are the same, the system processes the tool path as one point.
- 7) The system alarms: E237 chord length is greater than the diameter when the diameter 2R is less than the distance between the current point (starting point) and end point in R programming.
- 8) G02, G03, G05 cannot be in the same block with T, or the system issues: E205 Command incompatible.
- 9) The arc cannot be more than 180° in R programming.

**【Example 1】**



Draw the full circle along the arc direction to judge whether the mached arc is CCW or CW interpolation. When it is CW, the system executes G03; when it is CCW, the system uses G02, as shown in the left figure.

Fig. 4-6 arc interpolation

Clockwise interpolation from A to B at feedrate of 100mm/min as Fig. 4-6:

```

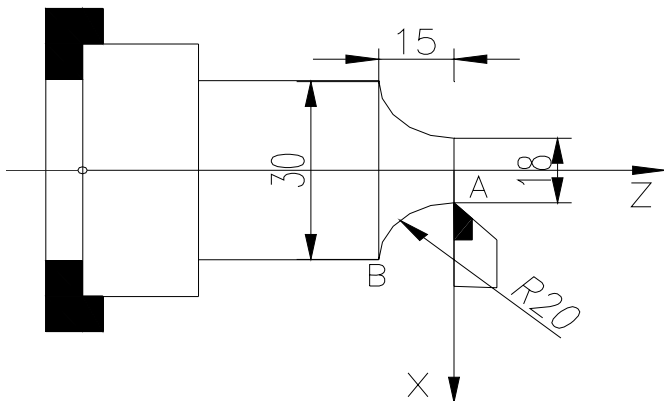
N0100 G00 X20 Z60 ; Rapid positioning to arc machining start point
N0110 G03 X60 Z40 I0 K-20 F100 ; Circle center programming
N0110 G03 X60 Z40 R20 F100 ; Radius programming
Or N0110 G03 U40 W-20 I0 K-20 F100 ;
N0110 G03 U40 W-20 R20 F100 ;
    
```

Counterclockwise interpolation from A to B at the feedrate of 100mm/min:

```

N0100 G00 X20 Z60 ; Rapid positioning to arc machining start point
N0110 G02 X60 Z40 I40 K0 F100 ; Circle center programming
N0110 G02 X60 Z40 R20 F100 ; Radius programming
Or N0110 G02 U40 W-20 I40 K0 F100 ;
N0110 G02 U40 W-20 R20 F100 ;
    
```

**【Example 2】**



```

Absolute programming:
N0000 G0 X18 Z0 ;
N0010 G02 X30 Z-15 R20 F100 ;
N0020 M30

Relative programming:
N0000 G0 X18 Z0 ;
N0010 G02 U12 W-15 R20 F100 ;
N0020 M30
    
```

Fig. 4-7 Circular interpolation

## 4.4 Thread Cutting Command

The CNC system can execute many kinds of thread cutting function including machining inch/metric single, multi-thread, variable pitch thread and tapping cycle; the thread run-out length and the angle can be changed, and the cycle thread cutting can perform cutting by single side, which protects the tool and improve the surface smoothness. The thread function includes: continuous thread cutting command G33, tapping cycle cutting command G32, thread cycle cutting command G92, multiple thread cutting cycle command group G76.

The machine with the thread cutting function must be installed with a spindle encoder whose number of lines is set by **P209**. In thread cutting, the system starts to move X or Z axis to execute the thread machining after it receives the one-turn signal of the spindle encoder. Therefore, the system can execute the roughing, finishing many times to complete machining of the same thread without changing the spindle speed.

The system has many kinds of thread cutting function used to machine the thread without the tool retraction groove. However, there is large pitch error due to the acceleration/deceleration of X axis and Z axis at the beginning and end of the thread cutting. Therefore, it is necessary to leave the thread lead-in length and the tool retraction distance at the actual thread start point and the end.

When the thread pitch is determined, X, Z traverse speed is determined by the spindle speed in thread cutting and is not related to feedrate override. When the spindle override control in the thread cutting is valid and the spindle speed changes, there is pitch errors because of X, Z acceleration/deceleration. Therefore, never attempt to adjust the spindle speed or stop the spindle in thread cutting. Stopping the spindle may cause damage to the tool and workpiece. The thread cutting command can only be Z/X programming.

### 4.4.1 G33 — Thread Cutting

#### 【Command format】

**G33 X (U)\_ Z (W)\_ P (E)\_ K\_ I\_ Q\_ H\_ ;** Thread cutting  
**G33 Z (W)\_ P (E)\_ K\_ I\_ Q\_ H\_ ;** Axial straight thread cutting  
**G33 X (U)\_ P (E)\_ K\_ I\_ Q\_ H\_ ;** End face straight thread cutting

#### 【Word】

P — Metric thread lead.

E — Inch thread lead.

It is the axial thread and Z axis is the thread axis when P/E is a positive value; it is the end face thread and X axis is the thread axis when it is a negative value.

X(U) /Z(W) — Absolute/relative coordinates of thread end point.

For the axial thread, Z movement amount cannot be 0; it is the axial straight-thread if X is omitted.

For the end face thread, X movement amount cannot be 0; it is the end face thread when Z is omitted.

**K** — It is the length from the thread run-out starting point to the thread end point in the thread machining axis.

K cannot be negative and must be less than the movement amount of thread machining axis.

**I** — Total movement amount in thread run-out axis direction during thread run-out. (there is no thread run-out when it is omitted).

For straight-thread, the thread runs-out in the positive direction when I is positive; the thread runs-out in the negative direction when I is negative.

For taper thread, the thread run-out direction is same as that of the taper, the sign of I has no effect.

For axial taper thread, the thread runs-out in the positive direction when U is positive; the thread runs-out in the negative direction when U is negative.

For end face thread, the thread runs-out in the positive direction when W is positive; the thread runs-out in the negative direction when W is negative.

**Q** — Initial angle, which is 0° if not specified; it is used to thread part.

**H** — Execution mode symbol, each bit has different meaning (H\_d7 ~ H\_d2: reserved) ; it is 00000000 if it defaults

H\_d0 = 0: Retraction is limited by K value;

H\_d0 = 1: Thread run-out starts when the thread cutting axis reduces speed, and K value is invalid.

H\_d1: It selects the continuous thread machining raising/reducing speed mode

H\_d1 = 0: In continuous thread cutting, between two neighboring threads, the thread machining axis has a process of reducing the speed from machining speed to initial speed, and then raising the speed from initial speed to thread machining speed. The thread pitch length changes in the transition stage of speed raising and reducing

H\_d1 = 1: In continuous thread cutting, between two neighboring threads, the machining speed of the current thread suddenly changes to the one of the next thread without the process of reducing the speed from machining speed to initial speed, and then raising the speed from initial speed to thread machining speed. Therefore, it is not suitable to machine those taper threads with a large taper using this mode. Otherwise, the motor may step out.

#### 【Word range】

X, Z, U, W: -9999.999mm~9999.999mm

I: -9999.999mm~9999.999mm

K: 0~9999.999mm

P: 0.001mm~500.000mm (the negative sign can be added at the front of the range. The positive value indicates the axial thread, and the negative value indicates the end face thread.)

E: (0.060~25400.000) tooth/inch(the negative sign can be added to the front of the range. The

positive value means to the axial thread, and the negative value means to the end face thread.)

Q: 0°~360.000°

H: 00000000~11111111

**【Relative parameters】**

Parameters related to G33: **P100, P101, P103, P104, P106, P107, P113, P116, P117, P209, P306, P307, P403\_d0.**

Taking axis thread for example, the relationship between G33 and relative parameters:

- ① Before the system enters raising speed stage, it firstly checks whether the thread cutting speed exceeds **P113**, if it does, the system issues an alarm to terminate the thread machining;
- ② Raising speed phrase: Z axis raises speed at **P103** initial speed; Acceleration time of raising speed stage is **P116**;
- ③ When there is thread run-out, X axis rapidly executes the thread run-out at P101 speed;
- ④ When there is thread run-out, its acceleration/deceleration time is **P117**;
- ⑤ The boundary of two thread machining modes is **P306**;
- ⑥ During thread cutting, the system detects the spindle speed through spindle encoder line number P209. If the spindle rotation speed exceeds P307, an alarm occurs after the current thread machining block is finished, and then the machining is suspended; by pressing **CYCLE START** key, the system continues executing the next step of the machining program.
- ⑦ **P403\_d0=0**: the system does not check whether the spindle speed is stable before the thread machining.

**P403\_d0=1**: the system checks whether the spindle speed is stable before the thread machining. It does not machine threads until the speed becomes stable.

**【G33 execution process】**

G33 execution process is as follows (taking axial thread cutting as an example):

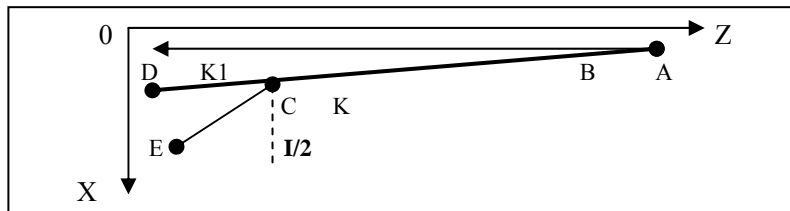


Fig. 4-8a G33 execution path 1

**I=0, no run-out at the end of thread:**

- ① Check the spindle speed, count the reference speed of thread machining according to the rotation speed and the lead P. When the reference speed exceeds P113 (max. cutting feedrate), the system issues an alarm. Before this, the user must ensure the spindle speed is stable; otherwise, there may be confused machining teeth.
- ② Check one-turn signal of coder from the initial angle; Z axis raises speed from the current

position A (G33 starting point), which makes the traverse speed reach the reference speed. At the moment, Z moves to B; the lead of AB section maybe not standard (in the actual machining, the front of AB section must reserve an idle distance which does not touch the workpiece), the higher the reference speed is, the longer AB section is.

- ③ The system follows the spindle speed from B to execute the cutting to C; the cutting speed changes with the spindle speed; the lead of BC section is standard.
- ④ Z reduces speed from C till D position Z (W); CD lead maybe not standard, CD length is equal to AB section (when the reference speed is lower than the initial speed, raising/reducing speed is not required, the lengths of CD and AB are 0); therefore, X should execute the thread run-out in advance when the end of thread has no thread run-out groove.

#### **I≠0, thread end has run-out:**

- ① Z movement is the same as the above; X starts executing the thread run-out at different position in advance according to the value of H\_d0; X thread run-out is performed based on G00 speed, it also has a process of raising speed, constant speed and reducing speed; the longer the X distance I is, the better the effect is. If permitted, the speed should be more than 40mm.
- ② When H\_d0=1, and Z moves to the reducing speed point C, X starts executing the thread run-out in advance, which is not limited by K; it finally moves to E.
- ③ If H\_d0=0, and Z moves to the position where the distance to the end point D is K (K point), X starts executing the thread run-out in advance till E. If K is big, the system executes the thread run-out in the middle of BC section, which will result in untimely run-out (see Fig. 4-8b for its path); when K is small, the system executes the thread run-out in the middle of CD section.



Fig. 4-8b G33 execution path 2

#### **【Explanation】**

- 1) G33 can machine metric/inch constant straight, taper, internal and external threads.
- 2) In the spindle CW rotation, the positive cutting is the right-hand thread, and the negative is the left-hand thread. For the spindle CCW rotation, they are reverse.
- 3) Generally, it is necessary to rough turn and finish turn the same path many times during thread machining. The thread cutting starts after 1-turn signal from the spindle coder is detected, so the cutting points on the circumference of the workpiece are the same after multiple thread cuttings. However, the spindle speed must keep unchanged; otherwise there is an error in the thread.
- 4) The thread machining cutting speed is controlled by P113 (max. cutting feedrate); when the speed exceeds the max. feedrate, the system issues an alarm. It is suggested that the thread machining cutting speed should be less than 3000 mm/ min; if the speed is too high, the motor may be slow to response, causing confused thread teeth.

The thread feedrate formula is as follows:

Inch thread speed =  $N \times 25.4 / E$

Metric thread speed =  $N \times P$

N — Speed (unit: r/min) max. speed is less than 2000r/min.

P — Thread lead (unit: mm). It is switched into the metric unit to count when it is in the inch thread.

- 5) In thread cutting start and end, the lead is not correct because of raising/reducing speed, so, the commanded thread length should be longer than the actual required length. Generally, the length for raising speed > 1.3 mm.
- 6) For axial taper thread, the pitch P/E is the pitch of the thread axis direction (Z); for the end face thread, it is the pitch of the thread axis direction (X).

**【Note】**

- 1) Start the spindle before machining the thread; otherwise, the system keeps waiting all the time (not system crash).
- 2) In the course of thread cutting, the feed hold key and the feedrate override are invalid; the feed stops as the spindle stops.
- 3) As soon as the spindle starts, do not machine the thread because of the unstable spindle speed, otherwise, the thread machining precision may be affected.
- 4) In thread cutting, the photoelectric coder with 1200 or 1024 lines are generally installed to rotate with the spindle synchronously. The selected coder lines must be the same as those of the actually-installed coder. When the coder lines are 1200, P209 should be set to 1200; when the coder lines are 1024, P209 should be set to 1024. If P209 setting is wrong, the pitch will be incorrect when the thread is machined.
- 5) For the thread with run-out, the effect of thread run-out is affected by the spindle speed, the pitch, the acceleration time of the run-out axis, the initial speed and I value. The higher the speed is, and the bigger the pitch is, and the longer the acceleration time is, and the lower the initial speed is, and the smaller I value is, the poorer the thread run-out effect will be.
- 6) When the previous block and the current one are both the thread cutting command, the system does not detect the thread head signal (only one per revolution) but directly starts movement. This function can realize the continuous thread machining.  
 E.g. G33 W-20 P3; The system detects 1-turn signal in thread cutting.  
 G33 W-30 P2; The system does not detect 1-turn signal in thread cutting.
- 7) The command must not be in the same block with other commands. It should be an independent block.

G33 thread cutting programming modes (diameter programming)

	<p>1) G33 axial straight thread:                  P is positive;                  I sign determines the direction of the thread run-out;                  there is no thread run-out when I is omitted;                  Example: G00 Z100 X100                  G33 W-40 K1 I5 P2                  M30                  End point (Z60, X105)</p>
--	--

	<p>2) G33 axial straight thread:                  P is positive;                  I sign determines the direction of the thread run-out;                  there is no thread run-out when I is omitted;                  Example: G00 Z100 X100</p>
--	---

	<p>3) G33 axial taper thread:                  P is positive;                  U sign determines the direction of the thread run-out;                  I sign is invalid;                  Example: G00 Z100 X20                            G33 U6.2 W-40 K3 I5 P2                            M30                  End point (Z60, X31.20)</p>
--	--

	<p>4) G33 end face straight thread:                  P is negative;                  I sign determines the direction of the thread run-out;                  when I is omitted, there is no thread run-out;                  Example: G00 Z100 X20                            G33 U20.7 K3 I1.6 P-2                            M30                  End point (Z101.60, X40.70)</p>
--	---

	<p>5) G33 end face taper thread:                  P is negative;                  W sign determines the direction of the thread run-out;                  I sign is invalid;                  Example: G00 Z100 X20                            G33 U20.7 W-8.3 K3 I1.6 P-2                            M30                  End point (Z90.10, X40.70)</p>
--	---

**【Example】**

1) G33 axial straight thread: as Fig. 4-9



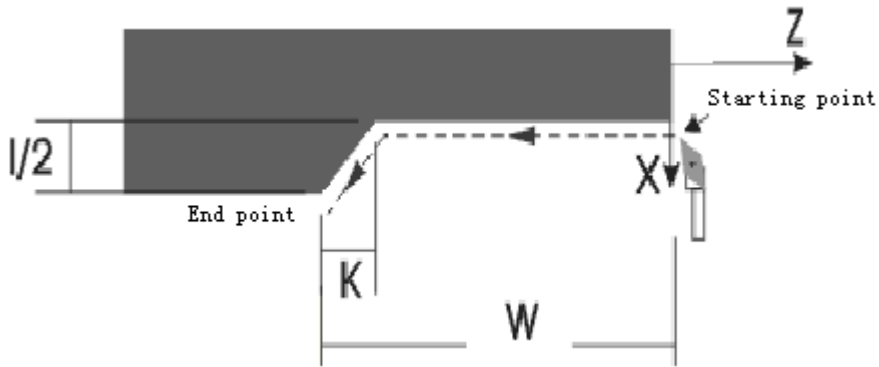


Fig. 4-9

I sign determines the direction of thread run-out; there is no thread run-out if I is omitted;

Example: G00 X100 Z100

G33 W-40 K3 I5 P2 ; End point (Z60, X105)

M30

**II PROGRAMMING**

2) G33 end face taper thread: as Fig. 4-10

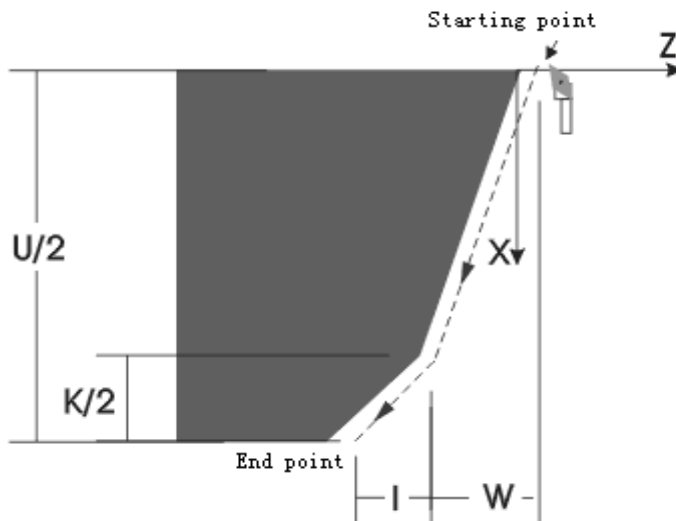


Fig. 4-10

P is negative; W sign determines the direction of the thread run-out; I sign is invalid;

Example: G00 Z100 X20

G33 U20.7 W-8.3 K3 I1.6 P-2 ; end point (Z90.10, X40.70)

M30

3) G33 axial taper thread: as Fig. 4-11

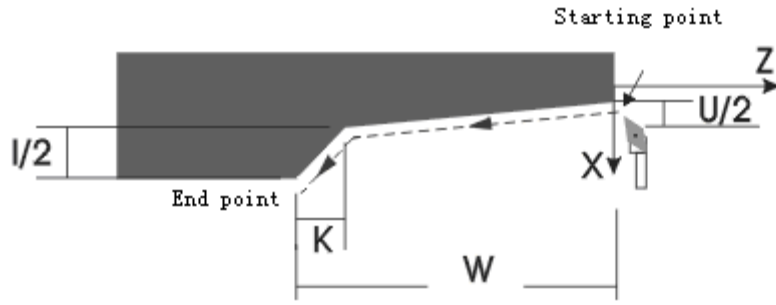


Fig. 4-11

U sign determines the direction of the thread run-out; I sign is invalid;

Example: G00 Z100 X20

G33 U6.2 W-40 K3 I5 P2 ; End point (Z60, X31.20)

M30

4) G33 end face straight thread: as Fig. 4-12

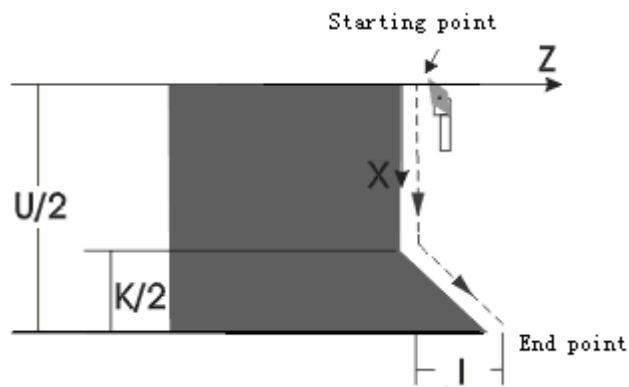


Fig. 4-12

P is negative; I sign determines the direction of the thread run-out; there is no run-out if I is omitted;

Example: G00 Z100 X20

G33 U20.7 K3 I1.6 P-2 ; End point (Z101.60, X40.70)

M30

5) Comprehensive example: as Fig. 4-13

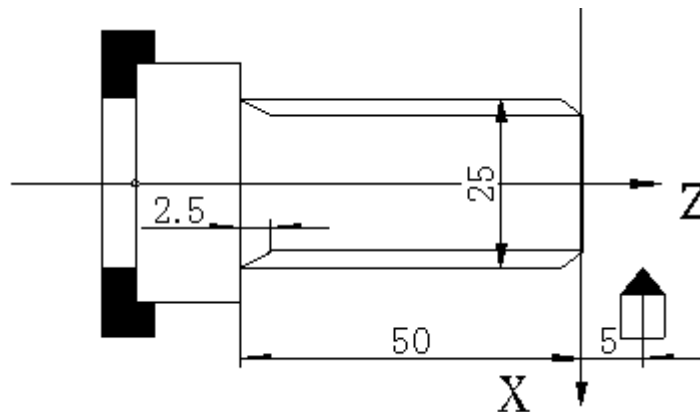


Fig. 4-13 Thread cutting

Example: thread lead: 2mm, cutting depth: 2.5mm(diameter value, cutting twice) as Fig. 4-13:

```

N0000 G00 X25 Z5 ; Approaching the workpiece
N0010 G01 X23.5 F100 ; Tool infeed 1.5 mm (diameter programming)
N0020 G33 Z-50 P2 K2.5 I5.0 ; First thread cutting
N0030 G00 X26 ;
N0040 Z5 ; Return to the starting point
N0050 G01 X22.5 ; Second tool infeed 1mm
N0060 G33 Z-50 P2 K2.5 I5.0 ; Second thread cutting
N0070 G00 X26 ;
N0080 Z5 ; Z returns to starting point
N0090 M30

```

## 4.5 G32 —Tapping Cycle

### 【Command format】

**G32 Z (W)\_ P (E)\_ H\_ ; Z axis tapping**

### 【Word】

Z(W): Position which the tapping is performed to; Z is the absolute coordinate of Z axis, and W is the relative coordinate of Z axis; the user can use either relative coordinate or absolute coordinate. The relative coordinate is the movement amount relative to the current position

P: Metric thread lead;

E: Inch thread lead;

H: Execution mode symbol (reserved)

### 【Word range】

Z, W: -9999.999mm~9999.999mm

P: 0.001mm~500.000mm

E: 0.060 teeth/inch~25400.000 teeth/inch

H: 00000000~11111111

### 【Relative parameters】

Parameters related to G32: **P100, P103, P106, P112, P113, P114.**

### 【Explanation】

**G32 Z tapping cycle execution process:**

- ① Spindle rotation starts; Z axis feeds for tapping.
- ② Spindle OFF.

- ③ Waits the spindle to exactly stop.
- ④ The spindle rotates in the reverse direction
- ⑤ Z axis retracts to the starting point of the cycle.
- ⑥ The spindle stops.

**【Note】**

- 1) Determine the spindle rotation direction according to the direction in which tapping is possible before tapping. The spindle will stop after the tapping ends. Restart the spindle if you need to continue the machining.
- 2) There is a deceleration time for the spindle after the spindle is turned off. At the moment, Z axis still rotates along with the spindle till the spindle stops completely. Therefore, the thread bottom hole in actual machining is deeper than the actually-required one. The actual depth should be determined by the spindle speed in tapping and by whether the spindle brake is installed or not.
- 3) The system forbids executing G32 in DRY RUN mode.
- 4) The other cautions are the same as those of G33.

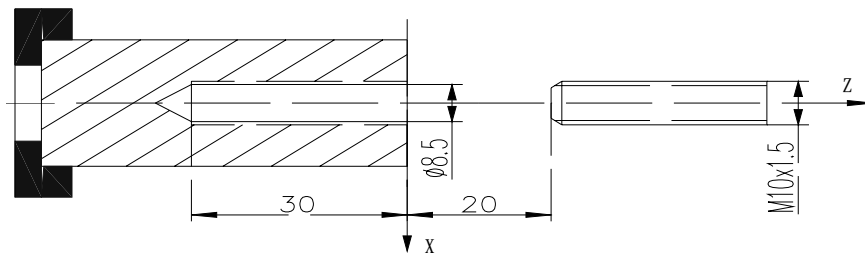
**【Example】**


Fig. 4-15 tapping

Example: Single thread with 1.5 mm lead

N0010	G00	X0	Z20	; Rapidly positioning the starting point of workpiece
N0020	M3	S01		; Spindle CW rotation
N0030	G01	Z2	F500	; Z axis approaches the workpiece
N0040	G32	Z-30	P1.5	; Z axis infeed tapping
N0050	G00	Z20		; Moving away from the workpiece and returning to program start point
N0060	M02			; End of program

## 4.6 G50 — Setting Workpiece Coordinate System

The system directly modifies the current tool nose coordinates into the coordinate values set by G50 when executing G50 and sets the current machine coordinates to the program reference point, which is equivalent to modifying the tool nose coordinates in Manual operation mode and then re-setting the program reference point.

After executing G50 command, there is a green icon  being displayed ahead of the corresponding axis, which prompts the program reference point return.

The coordinate system created by G50 is referred to as the workpiece coordinate system. After the workpiece coordinate system is created, the position of absolute coordinates of the following commands is the coordinate values in this coordinate system.

In creating workpiece system, the X axis in the workpiece coordinate system is generally defined at the workpiece rotation center, while the Z axis in the workpiece coordinate system can be defined at the chuck end face or workpiece end face according to your custom.

#### 【Command format】

- G50 Z\_ X\_** ; Two-axis workpiece coordinate system setting  
**G50 X\_** ; X single-axis workpiece coordinate system setting  
**G50 S\_** ; Max. spindle speed limit in constant surface speed state. See G96, G97

#### 【Word】

Z, X, Y are absolute coordinate values. S limits the max. spindle speed (r/min). Refer to G96, G97 for details.

#### 【Word range】

X, Z: -9999.999mm~9999.999mm

#### 【Explanation】

- 1) Z/X/ cannot be in the same block with S. Z/X can select single-axis or simultaneous two-axis to set the workpiece coordinate system.
- 2) G50 must be specified in an independent block. It cannot be in the same block with other commands.
- 3) Because there has been one workpiece coordinate system and one program reference point in Jog operation mode, the system uses the new workpiece coordinate system and program reference point in Auto and Jog operation modes after the system executes G50 till they are replaced.

#### 【Relative parameter】

Parameters related to G50: **P000, P001.**

#### 【Example】

G50 X100 Z100 ; Two-axis workpiece coordinate system setting

## 4.7 G51 — Recovering Workpiece Coordinate System Setting

#### 【Command format】

**G51**

**【Explanation】**

G51 must be specified in an independent block. It cannot be in the same block with other commands.

Recover the workpiece coordinate system and reference point set in Jog operation mode. When the workpiece coordinate and the program reference point set in Jog operation mode are replaced after G50 is executed, use G51 to recover them. After recovering, the system uses the previous workpiece coordinate system and program reference point in Jog and Auto operation modes till they are replaced.

## 4.8 G26 — X, Z Reference Point Return

Return to the program reference point through the intermediate point at rapid traverse rate by G00.

**【Command format】**

**G26 Z (W)\_ X (U)\_** ; Z/X rapidly and simultaneously move to the program reference point through the intermediate point

**G26 Z (W)\_** ; Z axis rapidly moves to the program reference point without moving other axes.

**【Word】**

X, Z: Absolute coordinates of the intermediate point;

U, W: Relative movement amount from starting point to middle point.

**【Word range】**

X, Z, U, W: -9999.999mm~9999.999mm

**【Explanation】**

- 1) After this command is executed, the specified coordinate axes move to the point defined by G50. The axes move to the program reference point defined in the Jog operation mode if there is no G50 in the program. After the corresponding axis executes the reference point return, ahead of the machine coordinates of this axis is displayed a green icon as a prompt.
- 2) The command and other commands cannot in the same block.
- 3) The execution mode of this command is the same as that of G00; refer to G00, G50 for relative parameters.
- 4) Z/X can select single-axis, two-axis simultaneous program reference point return.

**【Example】**

G26 Z150 X100 ; Return to program reference point through the intermediate point (Z150, X100)

G26 U0 ; X axis directly returns to program reference point without moving other axes

## 4.9 G28 — Return to Machine Zero (Machine Reference Point)

## 【Command format】

**G28 Z(W) \_** ; Z axis rapidly moves to machine zero through the intermediate point without moving other axes

**G28 X(U) \_** ; X axis rapidly moves to machine zero through the intermediate point without moving other axes

## 【Word】

X, Z: Absolute coordinates of the intermediate point;

U, W: Relative movement amount from starting point to intermediate point.


## 【Word range】

X, Z, U, W: -9999.999mm ~ 9999.999mm

## 【Relative parameter】

Main parameters related to G28: **P021 ~ P026, P109, P110, P406 and P407.**

## 【Explanation】

- 1) When the machine zero detection devices (zero switch and deceleration switch) are not installed on the corresponding coordinate axis, the corresponding bit of P406 should be set to 0; in executing G28, the system does not detect the zero signal and deceleration signal, but moves the axis to the intermediate point and then returns to the zero coordinate position of this axis.
- 2) When the machine zero detection devices (zero switch and deceleration switch) are installed on the corresponding coordinate axis, the corresponding bit of P406 should be set to 1; in executing G28, the axis directly returns to the machine zero at the rapid traverse rate from the start point without passing the intermediate point; which is equivalent to the machine zero return in Jog operation mode.
- 3) For G28 machine zero return, see Section 4.4 Machine Zero Return in Jog Operation Mode in Part II Programming.
- 4) G28 is a one-shot G command; its execution mode is the same as that of G00; for other relative parameters, refer to G00.
- 5) After the corresponding axis executes the machine zero return, there is a blue icon  being displayed behind its machine coordinate
- 6) This command must be specified in an independent block. It cannot be in the same block with other commands.
- 7) Z/X can only perform the single-axis machine zero return.

## 【Example】

G28 U0 ; X axis rapidly returns to the machine zero without moving other axes

G28 W0 ; Z axis rapidly returns to the machine zero without moving other axes

4.10 G30 — 2<sup>nd</sup>, 3<sup>rd</sup> Program Reference Point Return

## 【Command format】

**G30 P2 Z (W)\_ X (U)\_** ; Z/X rapidly return to the 2<sup>nd</sup> program reference point via the intermediate point

**G30 P3 Z (W)\_ X (U)\_** ; Z/X rapidly return to the 3<sup>rd</sup> program reference point via the intermediate point

**G30 P2 Z (W)\_** ; Z axis returns to the 2<sup>nd</sup> program reference point via the intermediate point without moving the other axes

## 【Word】

X, Z: Absolute coordinates of the intermediate point;

U, W: Relative movement amount from starting point to intermediate point.

P2 specifies the 2<sup>nd</sup> program reference point; P3 specifies the 3<sup>rd</sup> program reference point.

## 【Word range】

X, Z, U, W: -9999.999mm~9999.999mm

P: 2 or 3

## 【Relative parameters】

Main parameters related to G30: **P003~P008**.

## 【Explanation】

- 1) The execution mode of the command is the same as that of G00; for other relative parameters, refer to G00.
- 2) This command must be specified in an independent block. It cannot be in the same block with other commands.
- 3) Z/X can select single-axis, two-axis simultaneous 2<sup>nd</sup>, 3<sup>rd</sup> program reference point return.

## 【Example】

**G30 P2 Z150 X100** ; Return to 2<sup>nd</sup> program reference point via the intermediate point (Z150, X100)

**G30 P3 Z150 X100** ; Return to 3<sup>rd</sup> program reference point via the intermediate point (Z150, X100)

**G30 P3 W0** ; Z axis rapidly returns to 3<sup>rd</sup> program reference point directly.

## 4.11 G04 — Timing Delay

## 【Command format】

**G04 D\_\_** ; Timing delay



## 【Word】

D — Dwell time. (unit: s)      G04 defines the time interval between two blocks

## 【Word range】

D: 0~9999.999 s

## 【Explanation】

- 1) G04 cannot be in the same block with other commands except S, F command.
- 2) In executing G04, pressing **CYCLE START** can terminate the delay, and then the system proceeds to the next command.

## 【Example】

Example : G04 D2.5 ; 2.5s is delayed

## 4.12 G96 —Constant Surface Speed Control, G97 —Constant Surface Speed Cancel

## 【Command format】

**G96 S** ; Constant surface speed control

**G97 S** ; Cancelling constant surface speed control

## 【Word】

S in G96 specifies the constant surface speed. (unit: m/min)

S in G97 specifies spindle speed after the constant surface speed is cancelled. (unit: r/min)

## 【Word range】

S: 0~9999 r/min

0~9999 m/min

0~4 (multi-gear spindle)

## 【Explanation】

G96, G97 are the modal words in the same group, only one of which is valid. G97 is the initial word and it is valid by default when the system is switched on.

When the machine tool is turning the workpiece, the workpiece usually rotates taking the spindle axis as the center line, the cutting point of tool cutting workpiece is a circle motion around the spindle axis, and the instantaneous speed in the circumference tangent direction is called **cutting surface speed** (or **surface speed** for short). Different workpieces and tools with different materials require different surface speed.

The constant speed control is valid only when the spindle speed analog voltage control function is valid. In the constant speed control, the spindle speed changes with the absolute value of X tool nose coordinate in the programmed path (tool length compensation is ignored). When the absolute value of X tool nose coordinate is increased, the spindle speed is reduced, and when absolute value is decreased,

the spindle speed is increased, which ensures the cutting surface speed keeps unchanged at S value. The use of constant surface speed control function for workpiece cutting can make the smoothness of workpiece surface with diameter change even.

$$\text{Surface speed} = \text{spindle speed} \times |X| \times \pi \div 1000 \quad (\text{unit: m/min})$$

Spindle speed: r/min;

|X|: Absolute value of X tool nose coordinate value (diameter value), unit: mm;

$\pi \approx 3.14$ .

In constant surface speed control, the spindle speed changes along with the absolute value of X tool nose coordinate of programmed path only in the process of cutting feed (interpolation).

In constant surface speed control, Z coordinate axis of workpiece coordinate system must coincide with the spindle axis (rotary axis of workpiece). Otherwise, there is different between actual surface speed and specified speed.

In constant surface speed control, G50 S\_ can limit the max. spindle speed (r/min). The actual spindle speed is the limit value of max. spindle speed when the spindle speed counted by the surface speed and X coordinate value is more than the max. spindle speed set by G50 S\_. After the system is switched on, the max. spindle speed limit value is **P305** (max. spindle speed of constant surface speed).

Max. spindle speed limit value defined by G50 S\_ is maintained before it is defined again and its function is valid in G96. Max. spindle speed defined by G50 S\_ is invalid in G97 but its limit value is still maintained.

#### 【Relative parameters】

Main parameters related to G96, G97: **P410\_d6, P304, P305**;

**P410\_d6** setting determines whether the system executes the constant surface speed control in G96;

Min. and Max. speed limits of constant surface speed control are **P304, P305** respectively.

#### 【Note】

- 1) The system issues **E272- S value cannot be 0 in G50** when G50 S0 is programmed.
- 2) The constant surface speed function can actually control the spindle speed only when the variable-frequency spindle is used, i.e. bit parameter P410\_d6 = 1. If the gear-shift spindle is used, the constant surface control cannot be performed even if G96 command is specified.
- 3) The constant surface speed control can be performed anytime when executing **G00, G01, G02, G03, G05, G90, G94, G71, G72 or G73**; the constant surface speed control is invalid during thread cutting when executing **G32, G33, G34, G76 or G92**; when X coordinate is changed in G50 or executing the tool offset, the spindle speed adjustment is valid in the next command.
- 4) Min./max. speed of constant surface speed control is set by **P304/P305** respectively.
- 5) The rotary axis in the constant surface speed must be set in Z axis (X=0) of the workpiece coordinate.

- 6) G96 is modal command. When G96 is valid, the single S command is taken as the new surface speed data.
- 7) In DRY RUN mode, the constant surface speed control is valid. Whether the single S command is executed in Dry run is controlled by **P401\_d7**: **P401\_d7= 0**: T, M, S command is executed in DRY RUN; **P401\_d7=1**: T, M, S command is not executed.
- 8) The constant surface speed control is invalid to get the constant spindle speed in thread cutting.
- 9) In G96, when the system defaults S word, the system automatically counts the surface speed as S value according to the current spindle speed (the last speed is taken as the current speed when the spindle does not start, or the spindle initial speed is taken as the current speed when the system is switched on just now) and X absolute coordinate value of tool nose.
- 10) In G97, when the system defaults S word, the system automatically counts the speed as S value according to the current constant surface linear speed and X absolute value of tool nose.
- 11) In AUTO operation mode, if the initial state of the program execution is G96, the system automatically changes G96 into G97 after **CYCLE START** key is pressed. The constant surface cutting is valid in machining the arc and straight line in the course of machining.
- 12) In Jog operation mode, move X coordinate in feed state (rapid/feed indicator OFF) in G96, G96 is valid, the spindle speed changes along with X coordinate value; and is invalid in rapid feed state(rapid/feed indicator ON).
- 13) S value of G96 is limited by S value of G50 Sxxxx and P304, P305. The S value of G50 is initialized to P305 upon power on, and their limit relationship is as follows:
  - **G50 Sxxxx<P304**: **G50 Sxxxx** is invalid, the spindle speed is adjustable between **P304~P305** in G96 .
  - **G50 Sxxxx>=P304**: **G50 Sxxxx** is valid. The min. value of spindle speed range is limited by P304 and max. value is limited by G50 Sxxx or P305, whichever is smaller.

**【Example】**

P304 value is the lower limit of the spindle speed in constant surface speed control. When the spindle speed counted by the surface speed and X coordinate value is lower than P304 value, the lower limit of the spindle speed is the actual spindle speed.

Example: as Fig. 4-17; surface speed is 300m/min which can count the spindle speed annotated in Table 4-2.

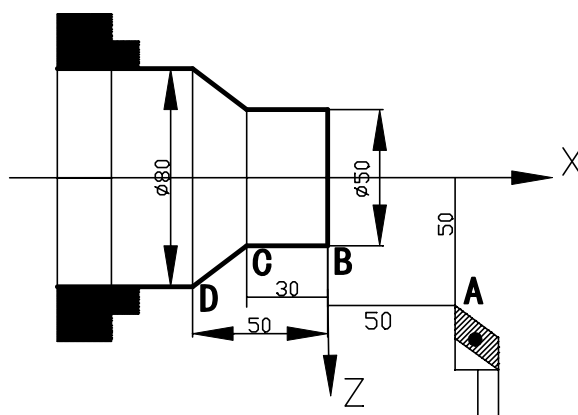


Fig. 4-15

Table 4-2

N0010	M3 G96 S300	; (Spindle rotates clockwise, the constant surface speed control is valid and the surface speed is 300 m/min)
N0020	G0 X100 Z50	; (Rapidly traverse to A point at the spindle speed of 955 r/min)
N0030	G0 X50 Z0	; (Rapidly traverse to B point at the spindle speed of 1910 r/min)
N0040	G1 W-30 F200	; (Cut from B to C at the spindle speed of 1910 r/min)
N0050	X80 W-20 F150	; (Cut from C to D, with the spindle speed changing from 1910 r/min to 1194 r/min)
N0060	G0 X100 Z100	; (Rapidly retract to A point at the spindle speed of 955 r/min)
N0110	M30	; (End of program, spindle stopping and coolant OFF)

### 4.13 Single Canned Cycle

It is necessary to cut the same machining path repeatedly in the course of some special roughing. To simplify the programming and improve the programming and the machining efficiency, the canned cycle is set. The tool will automatically return to the coordinate position before execution when executing the canned cycle once. If the cycle needs to be executed again, you only need to program the feeding data again instead of re-writing the canned command. The control is returned to the starting point of the cycle after the system executes the cycle. If other G commands are contained in the block behind the canned cycle, the cycle automatically ends. The single canned cycle only uses Z/X programming.

#### 4.13.1 G90 —Outer Cylinder Face Turning Cycle (Axial Cutting Cycle)

##### 【Command format】

**G90 Z (W)\_ X (U)\_ R\_ F\_** ; Inner/outer cylinder face turning cycle;

**X (U)\_ F\_** ; G90 cycle subsequent command; which indicates cycling once again by G90 contour.

##### 【Word】

X(U) Z(W) — Cylinder(taper) face end point; two axis coordinates should be given;

R —Diameter difference between starting point and end point of cycle. It is the axis surface cutting if R is omitted.

F —Feedrate.

##### 【Word range】

X, Z, U, W, R: -9999.999mm~9999.999mm

F: 0.001mm/min~4000mm/min

##### 【Command execution process】

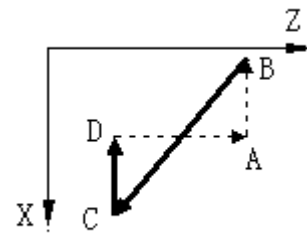
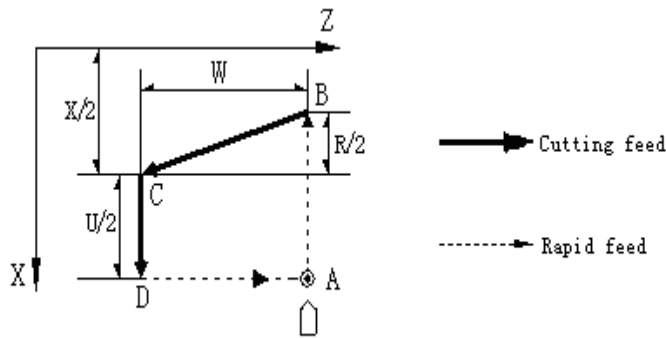


Fig. 4-16a Outer/inner cylinder (taper) face turning cycle

Fig. 4-16b Incorrect programming contour, causing system alarm

**G90 cycle process: (Fig. 4-16a)**

- ① X axis rapidly moves from A to B.
- ② X, Z axes cut from B to C at F speed (X does not move without R).
- ③ X axis cuts from C to D at F speed.
- ④ Z axis rapidly moves from D to A.

**【Relative parameter】**

In the process of G90 cycle execution, refer to G00 for the relative parameters of rapid traverse, and refer to G01 for relative parameters of cutting feed.

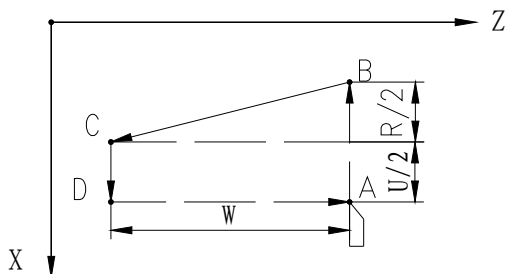
**【Explanation】**

- 1) The directions of inner, outer machining and cutting feed are determined automatically by the cycle start position and X, Z coordinates.
- 2) The tool stops at the cycle starting point after the cycle ends. The sign of U is determined by X direction from A to B when the system uses the relative coordinates. The sign of W is determined by Z direction from B to C. In taper face cutting cycle, the sign of R is determined by X direction from C to B
- 3) In programming, A B C D must form one quadrangle or a triangle in sequence; i.e. B, C must be on the same side of straight line AD, otherwise, an alarm is issued (Fig. 4-16b). When C coincides with D or A coincides with B, they form a triangle.
- 4) G90 can be followed by the cycle subsequent commands, which sets again X end point coordinate, and the system executes the cycle once again based on G90 contour; U in the subsequent command is relative to the cycle starting point. For example, when the system only redefines X end point coordinate (X/U), it repeats the above cycle according to the new X (U) coordinate. The cycle contour of the subsequent command must be consistent with the direction of G90.
- 5) There is only the single X (U) word (with F command) in the cycle subsequent command; the next block of the subsequent command can also be the subsequent command; if the next

block is not the single X (U) command but other G commands instead, the system cancels the cycle; if the next block is not the single X (U) but M, S, T command, the system prompts the alarm message.

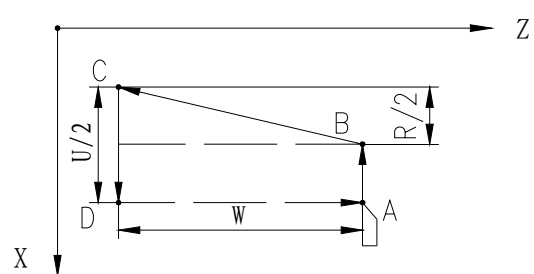
- 6) In single block execution, by pressing **CYCLE START** key once, one cycle step is executed. The system stops at the end point of each step during single block stop.
- 7) This command cannot be in the same block with other commands. It can only be specified in an independent block.
- 8) The explanations of the cycle contour of G94, G92 are the same as those of G90.
- 9) Relationships between the data behind U, W, R and the tool path are as follows:

(1)  $U < 0, W < 0, R < 0$



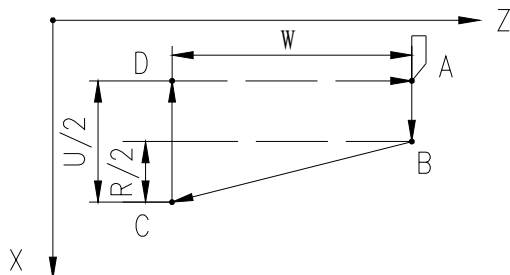
X from A to B is negative, so  $U < 0$ ;  
 Z from B to C is negative, so  $W < 0$ ;  
 X from C to B is negative, so  $R < 0$ .

(2)  $U < 0, W < 0, R > 0$



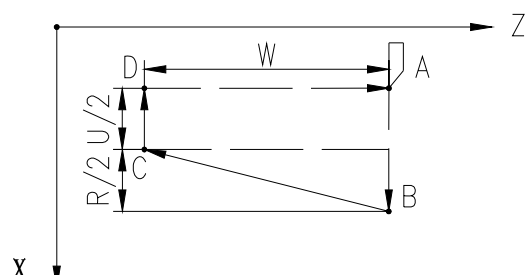
X from A to B is negative, so  $U < 0$ ;  
 Z from B to C is negative, so  $W < 0$ ;  
 X from C to B is positive, so  $R > 0$ .

(3)  $U > 0, W < 0, R < 0$



X from A to B is positive, so  $U > 0$ ;  
 Z from B to C is negative, so  $W < 0$ ;  
 X from C to B is negative, so  $R < 0$ .

(4)  $U > 0, W < 0, R > 0$



X from A to B is positive, so  $U > 0$ ;  
 Z from B to C is negative, so  $W < 0$ ;  
 X from C to B is positive, so  $R > 0$ .

Fig. 4-17 Relationships between signs of U, W, R and tool path in G90

**【Example】**

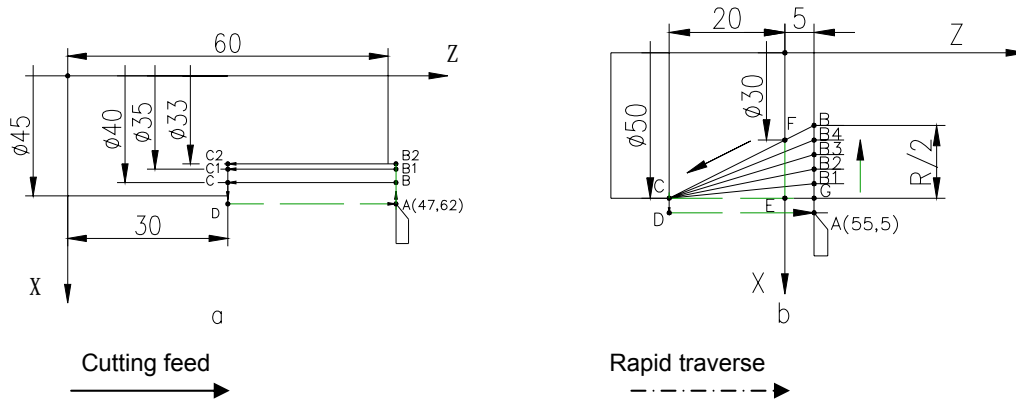


Fig. 4-18 G90 cutting example

**Example 1:** Fig. 4-18a outer cylindrical face: the first cutting feed= 5 mm, the second cutting feed= 2 mm, cutting feedrate F=100 mm/min.

```

N0010 G00 X47 Z62 ; Rapid positioning to A
N0020 G90 X40 Z30 F100 ; Cycle A B C D A once
N0030 X35 ; The first tool infeed executes E A B1 C1 D A
           cycle once
N0040 X33 ; The second tool infeed executes A B2 C2 D A
           cycle
N0050 M30
    
```

The tool is still on A after executing the above-mentioned blocks.

**Example 2:** Fig. 4-18b taper surface: the cutting feed R=- 5 mm once, F=100 mm/min.

```

N0010 G00 X55 Z5 ; Rapidly positioning to A
N0020 G90 X50 Z-20 R-5 F100 ; Executing cycle A B1 C D A
N0030 G90 X50 Z-20 R-10 ; Executing cycle A B2 C D A
N0040 G90 X50 Z-20 R-15 ; Executing cycle A B3 C D A
N0050 G90 X50 Z-20 R-20 ; Executing cycle A B4 C D A
N0060 G90 X50 Z-20 R-25 ; Executing cycle A B C D A
N0070 M30
    
```

The tool is still on A after executing the above blocks.

### 4.13.2 G92 —Thread Cutting Cycle

Thread cutting cycle is realizable using G92. The system rapidly traverses to the thread start point from the start point, then performs the cutting of the thread face, and then retracts from the thread face rapidly, finally returns to the start point rapidly. The whole motion path forms an enclosed path. If the thread is a multiple thread, another single-thread cutting cycle is performed. If a subsequent command

with another cutting depth follows G92, another cutting is also performed. Increasing the cutting depth gradually using the subsequent command can machine the threads which accord to the depth requirement.

**【Command format】**

**G92 X (U)\_ Z (W)\_ P (E)\_ K\_ I\_ R\_ L\_ H\_ ;** Thread cutting cycle.

**【Word】**

P —— Metric thread lead; E —— Inch thread lead.

When P/E is positive, Z axis is the thread axis; When P/E is negative, X axis is the thread axis.

X(U), Z(W)—— Absolute/relative coordinates of the thread end point;

Z movement amount of axial thread cannot be 0; X movement amount of end face thread cannot be 0 either.

R —— For the axial thread, it is the difference of X coordinate between thread start point and thread end point (it is the straight thread when R is omitted).

For the end face thread, it is Z coordinate difference between thread start point and thread end point.

For taper thread, R determines the direction of the thread run-out; i.e. the direction is positive when R is negative, and the direction is negative when it is positive.

K —— It is the length from the start point of the thread run-out to the thread end point in thread machining axis.

K cannot be negative and must be less than the movement amount of thread machining axis.

I —— It is the total movement amount in the run-out axis direction in thread run-out. (There is no thread run-out when it is omitted).

For the straight thread, the thread run-out is positive when I is positive; the thread run-out is negative when I is negative.

For the taper thread, the direction of thread run-out is the same as that of the taper; the sign of I is invalid.

L —— Thread number of multi-thread (it is single thread when L is omitted).

H —— Sign of execution mode, each bit has different meaning (H\_d7~H\_d3: reserved) ; the default is 00000000 when it is omitted.

H\_d0=1: Thread run-out starts when the thread machining axis reduces speed, which is not limited by K value.

H\_d0=1: The system executes the thread run-out according to the proportion between K and I. (if one of K, I is 0, the bit is invalid; H\_d1 is prior to H\_d0).

Note: When the system executes the thread run-out according to K and I proportion, if I/K ratio is big, the run-out axis speed raises rapidly, which can cause the stepper motor step-out, and the alarm of the servo motor.



H\_d2=1: It indicates precession; the programming rules for the thread precession function (feeding from the middle of the workpiece) are as follows:

- 1) In G92 command, I and K indicate the depth of precession, which are not omissible.
- 2) H\_d1=1 in G92: When the run-out is performed using I, K linkage mode, the conditions of precession are the same as those of run-out.
- 3) H\_d1=0 in G92: When the run-out is performed respectively using I, K rapid traverse mode, the conditions of precession differ from those of run-out.

**【Word range】**

X,Z,U,W,R: -9999.999mm~9999.999mm

I: -9999.999mm~9999.999mm;

K: 0~9999.999mm

P: 0.001mm~500.000mm (An negative sign can be added in front of this range. The positive means the axial thread, the negative means the end face thread)

E: (0.060~25400.000) teeth/inch (An negative sign can be added in front of this range. The positive means the axial thread, the negative means the end face thread)

L: 1~99

H: 00000000~11111111

**【Command execution process】**

The execution process of G92 thread cycle is as Fig. 4-19 (taking axial thread cutting for example)

- ① Check the spindle speed, count the reference speed of thread machining according to the speed and the pitch P. The system issues an alarm when the value exceeds P113 value (max. cutting feed speed). Before it, the user must ensure the spindle speed is stable; otherwise, it may cause to be the confused thread teeth.
- ② X axis rapidly traverses from A to B: X (U) + R position (Thread start point B).
- ③ X, Z execute the thread cutting from B to C.( including thread run-out); the process is the same as that of G33.
- ④ X rapidly returns to D.
- ⑤ Z rapidly returns to A ( starting point).
- ⑥ Multi-thread, repeat the above step 2~4 to execute the multi-thread cutting.

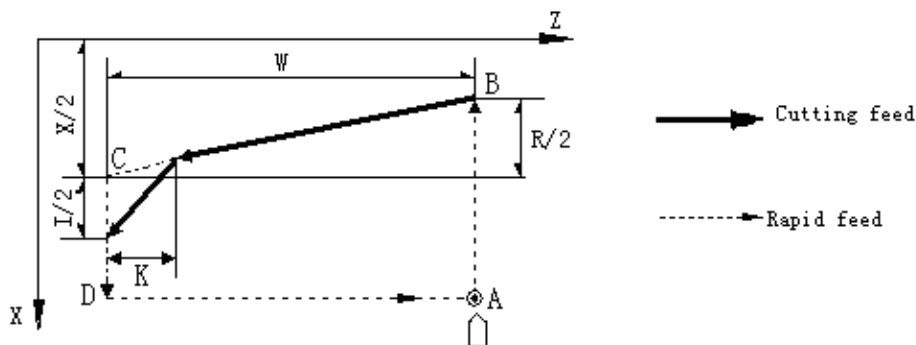


Fig. 4-19 G92 Thread cutting cycle

**【Relative parameter】**

In G92, refer to G00 for the relative parameters of rapid traverse, and refer to G33 for the relative parameters of thread cutting

**【Explanation】**

- 1) The thread cutting generally requires several times feeding. At the moment, the user only needs to alter X coordinate value of end point of cutting feed (or the increment value relative to the starting point). The coordinate position is still on the starting point when the thread cycle ends.
- 2) The command cannot be in the same block with other commands. It must be specified in an independent block.
- 3) G92 recycle subsequent command means the system executes one time the contour in G92 again; when P(E) is positive, G92 recycle subsequent command can only be X(U) ; when P(E) is negative, G92 recycle subsequent command can only be Z(W) .
- 4) The axial thread machining is limited by the diameter difference between the starting point and the end point of the thread in cutting taper thread; it is limited by Z coordinate difference between the starting point and end point of the cycle in cutting end face thread.
- 5) Notes are the same as those of G33 thread cutting.
- 6) For axial taper thread, the pitch P/E is the pitch in the thread axis (Z) direction; for the end face thread, it is the pitch in the thread axis (X) direction.
- 7) The relationship between R, K, P and tool path is as follows:

**◆ P/E is positive (axial thread, Z is the thread axis):**

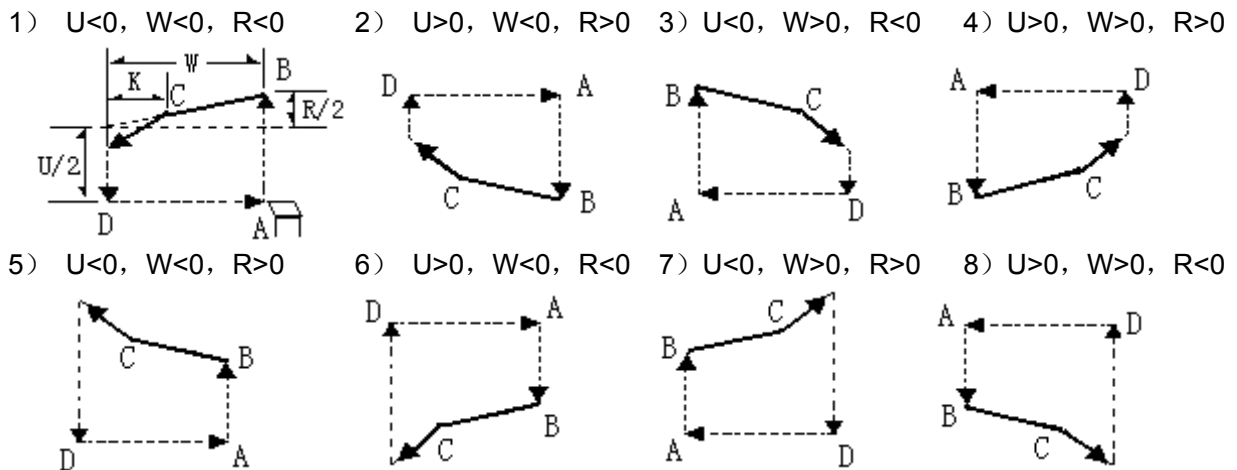
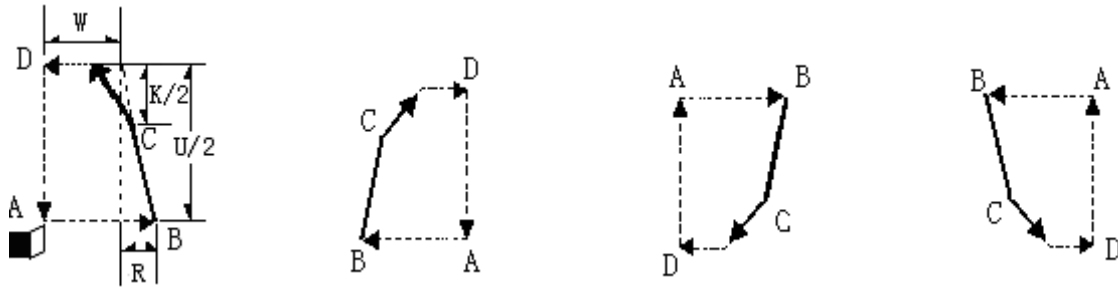


Fig. 4-20 G92 thread cutting tool path

◆ P/E is negative (end face thread, X is the thread axis):

- 1)  $U < 0, W > 0, R > 0$     2)  $U < 0, W < 0, R < 0$     3)  $U > 0, W > 0, R > 0$     4)  $U > 0, W < 0, R < 0$



- 5)  $U < 0, W > 0, R < 0$     6)  $U < 0, W < 0, R > 0$     7)  $U > 0, W > 0, R < 0$     8)  $U > 0, W < 0, R > 0$

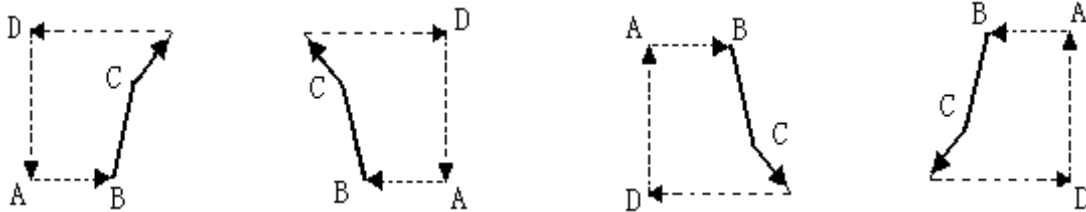


Fig. 4-21 G92 thread cutting tool path

G92 thread cutting programming forms: (diameter programming)

	<p>1) G92 axial face straight thread:                  P is positive;                  R=0;                  U&lt;0;                  The thread run-out is in the positive direction when I is positive; there is no thread run-out when there is no I; the end position does not change;</p>
--	--

	<p>2) G92 axial face straight thread:                  P is positive;                  R=0;                  U&lt;0;                  The thread run-out is in the negative direction when I is negative; there is no thread run-out when there is no I; the end position does not change;</p>
--	--

	<p>3) G92 axial face taper thread:                  P is positive;                  The thread run-out is in the positive direction when R is negative;                  U&lt;0;                  Example: G00 Z100 X100                  G92 W-60 U-50 R-10 K3 I6 P2                  U-51                  M30</p>
--	--

	<p>4) G92 end face taper thread:                  P is negative;                  R is positive;                  There is no the thread run-out when there is no I;                  Example: G00 Z100 X10                  G92 W-10 U60 R10 K3 I1.5 P-2                  W-10.5                  W-11                  M30</p>
--	--

	<p>5) G92 end face thread:                  P is negative;                  R=0;                  The thread run-out is in the positive direction when I is positive;                  Example: G00 Z100 X10                  G92 W-30 U80 K5 I2.5 P-2                  W-31.5                  W-32                  M30</p>
--	---

The programming example of G92 thread precession function machining is as follows:

	<p>1. G92 axial precession straight thread:                  P is a positive value; the sign of I determines the direction of run-out; I and K are not omissible; the linkage precession and linkage run-out are performed by I/K proportion.                  E.g. G00 Z100 X100                  G92 W-60 U-50 K3 I6 P2 H00000110                  M30</p>
--	--

	<p>2.G92 axial precession straight thread:                  P is a positive value; the sign of I determines the direction of run-out, I and K are not omissible; rapid precession and rapid run-out.                  E.g. G00 Z100 X100                  G92 W-60 U-50 K3 I6 P2 H00000100                  M30</p>
--	---

**【Example】**

1) G92 axial straight thread: as Fig. 4-22

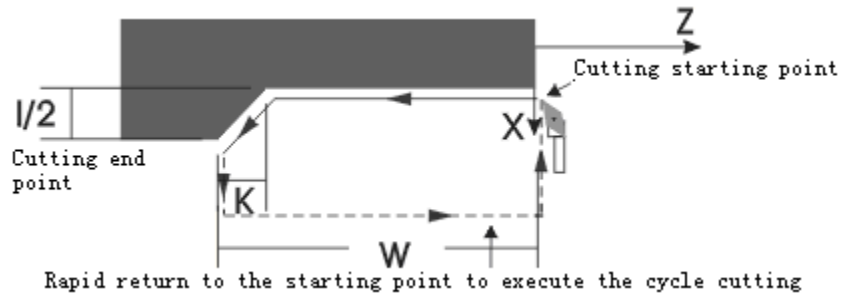


Fig. 4-22

P is positive; the sign of I determines the run-out direction; there is no the thread run-out when I is omitted;

- 2) G92 axial taper thread: as Fig. 4-23, P is positive; the sign of R determines the run-out direction; the sign of I is invalid.

Example: G00 Z100 X100

G92 W-60 U-50 R-10 K3 I6 P2

U-50.5

U-51

M30

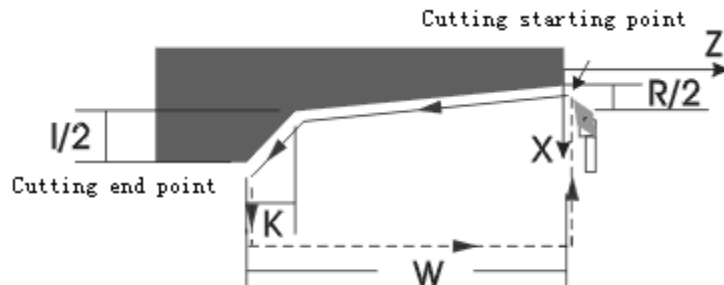


Fig. 4-23

- 3) G92 end face thread: as Fig. 4-24

P is negative; the sign of I determines the run-out direction; there is no run-out if I is omitted.

Example: G00 Z100 X10

G92 W-30 U80 K5 I2.5 P-2

W-31.5

W-32

M30

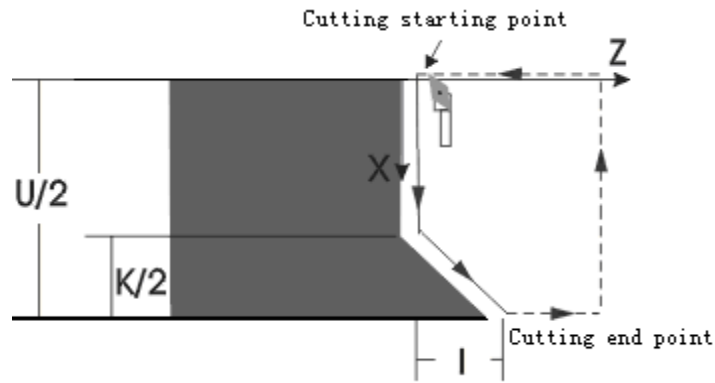


Fig. 4-24

4) G92 end face thread: as Fig. 4-25

P is negative; the sign of I determines the run-out direction; there is no run-out if I is omitted;

Example: G00 Z100 X10

G92 W-30 U80 K5 I2.5 P-2

W-31.5

W-32

M30

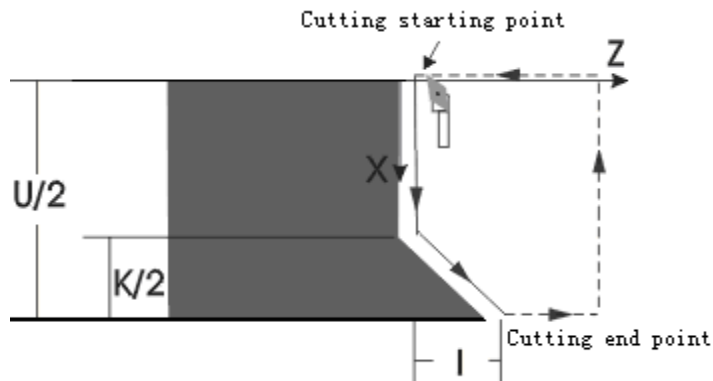


Fig. 4-25

5) Compound example: as Fig. 4-28

Example 1: Metric cylinder thread as Fig. 4-26 a. (G1  $\frac{1}{4}$ : teeth 11, d=41.910, d2=40.431, d1=38.952)

```

N0010 G00 X45 Z5 ; Rapidly position A
N0020 M03 S600 ; Spindle CW, 600r/min
N0030 G92 X41 Z-50 E11 ; First tool infeed, cutting 0.91 mm
N0040 X40.2 ; Second cutting 0.8mm
N0050 X39.6 ; Third cutting 0.6mm
N0060 X39.2 ; Fourth cutting 0.4mm
N0070 X38.952 ; Fifth cutting to the required dimension
N0080 M30
    
```

The tool is still on A after executing the above-mentioned blocks.

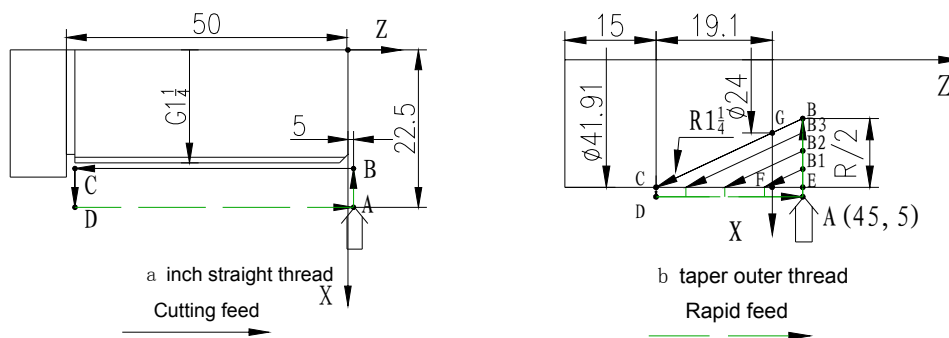


Fig. 4-26

Example 2: Taper Outer thread as Fig. 4-26 b ( $R1\frac{1}{4}$ :d=41.910, d2=40.431, d1=38.952, P=2.309, thread valid length is 19.1)

```

N0010 G00 X45 Z5 ; Rapid positioning to A
N0020 M03 S600 ; Spindle CW, 600r/min
N0030 G92 X41 Z-19.1 P2.309 R-1.506 ; First tool infeed cutting
N0040 X40.3 ; Second tool infeed cutting
N0050 X39.9 ; Third tool infeed cutting
N0060 X39.352 ; Fourth tool infeed cutting
N0070 M30
    
```

The tool is still on A after executing the above-mentioned blocks.

### 4.13.3 G94 — Inner/outer End (Taper) Face Turning Cycle

**【Command format】**

G94 Z (W)\_ X (U)\_ R\_ F\_ ; Inner/outer end (taper) face turning cycle  
 Z (W)\_ F\_ ; G94 recycle subsequent command; which means G94  
 coutour is cycled once again.

**【Word】**

X (U) Z (W) — End point coordinate. The coordinates of two axes must be given.  
 R — Z coordinate difference between starting point and end point of taper face. It is the  
 end face cutting if R is omitted.  
 F — Cutting speed

**【Word range】**

X, Z, U, W, R: -9999.999mm~9999.999mm  
 F: 0.001mm/min~4000mm/min

**【Command execution process】**

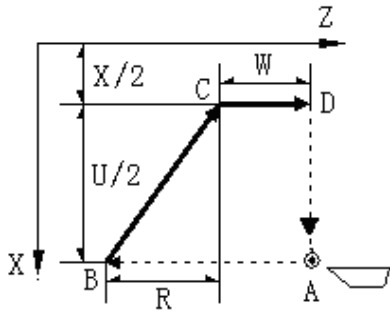


Fig. 4-27a G94 end face turning cycle

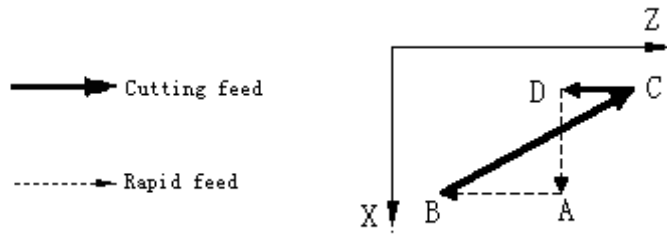


Fig. 4-27b Incorrect programmed contour which causes system alarm

G94 cycle execution process (Fig. 4-27a):

- ① Z axis rapidly moves from A to B.
- ② X, Z axes move at F speed from B to C (Z does not move when R is omitted).
- ③ Z axis moves at F speed from C to D.
- ④ X axis rapidly moves from D to A.

**【Relative parameter】**

In G94, refer to G00 for the relative parameters of rapid traverse, and refer to G01 for the relative parameters of cutting feed.

**【Explanation】**

- 1) The G94 can be followed by the recycle subsequent command, which means the Z end point coordinate is respecified, and the system recycles one more time according to the contour of G94; W in the subsequent command is relative to the cycle starting point. E.g. After the system redefines Z end point coordinate(Z/W), it repeats the cycle process according to the new Z (W) coordinate. The subsequent command cycle contour must be consistent with that of G94, otherwise, the system alarms.
- 2) There is only the independent Z(W) word in the cycle subsequent command; the next block of the subsequent command can also be the subsequent command; if the next block is not the independent Z (W) word but other G commands, the cycle is cancelled; when it is not the independent Z (W) word but M,S,T, the system prompts the alarm message.
- 3) This command cannot be in the same block with other commands. It must be specified in an independent block.
- 4) Other explanations of G94 are the same those of G90.
- 5) Relationships between the data behind U, W, R and the tool path are as follows:



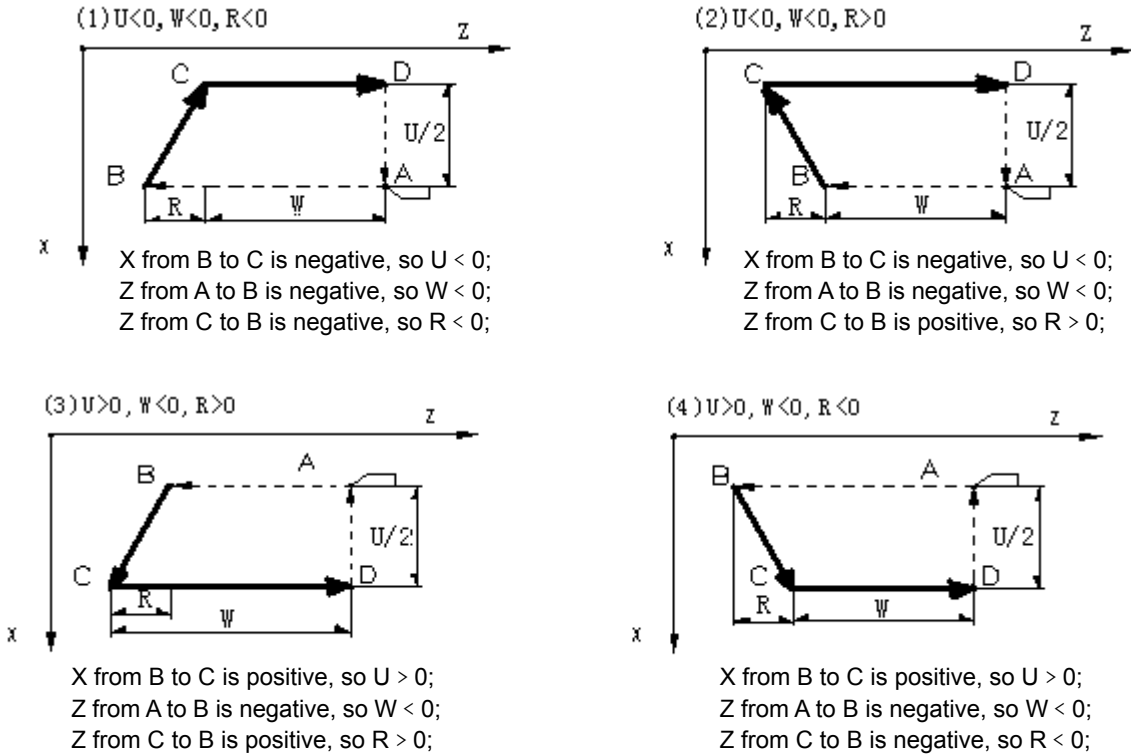


Fig. 4-28 G94 relationship between U, W, R sign and tool path

【Example】

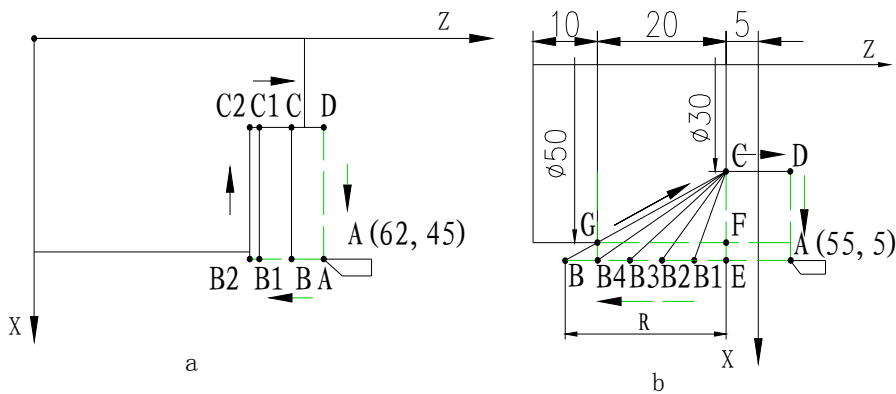


Fig. 4-29 G94 Cutting example

**Example 1:** Fig. 4-29a, the first feed= 5 mm, the second feed= 1.5 mm, cutting speed =80 mm/min, and its programming is as follows:

```

N0010 G00 X62 Z45 ; Rapid positioning to A
N0020 G94 X25 Z40 F80 ; 1st cutting cycle of end face cycle cutting A
                          B C D A
N0030 Z35 ; Tool infeed 5mm, the 2nd cycle A B1 C1 D A
N0040 Z33.5 ; Tool infeed 1.5m, the 3rd cycle A B2 C2 D A
N0050 M30
    
```

**Example 2:** Fig. 4-29b, feed R=-5 mm once, cutting speed =100 mm/min and its programming is as follows:

```

N0010 G00 X55 Z5 ; Rapid positioning to A
N0020 G94 X30 Z-5 R-5 F100 ; 1st cycle cutting of end face cycle cutting A
                                B1 C D A
N0030 G94 X30 Z-5 R-10 ; Tool infeed 5mm, 2nd cycle A B2 C D A
N0040 G94 X30 Z-5 R-15 ; Tool infeed 5mm, 3rd cycle A B3 C D A
N0050 G94 X30 Z-5 R-20 ; Tool infeed 5mm, 4th cycle A B4 C D A
N0060 G94 X30 Z-5 R-25 ; Tool infeed 5mm, 5th cycle A B C D A
N0070 M30
    
```

#### 4.13.4 G74 —End Face Deep Hole Machining Cycle

**【Command format】**

**G74 X (U)\_ Z (W)\_ I\_ K\_ R\_ E\_ F\_ ;** End face deep hole machining cycle

**【Word】**

X(U) Z(W)— Coordinates of hole bottom. It is the deep hole drilling cycle when X is omitted.

I — Each Z tool infeed amount (Z axis)

K — Each Z tool retraction amount (Z axis)

R — Pecking cycle or deep hole cycle selection. When R is omitted or R=0, the distance of retraction each time is only K, i.e. pecking cycle. When R=1, the tool retracts to the starting point of the first hole drilling each time, i.e. deep hole drilling cycle

E — Each offset value in X direction (diameter value) .

F — Feedrate.

**【Word range】**

X, Z, U, W: -9999.999mm~9999.999mm

I, K: 0~9999.999mm

E: Deep hole drilling cycle is 0~9999.999 mm; End face deep hole machining cycle is: 0.001 mm~9999.999 mm

R: 0 or 1

F: 0.001mm/min~4000mm/min

**【Explanation】**

R=1, the cycle process of G74 command is as follows:

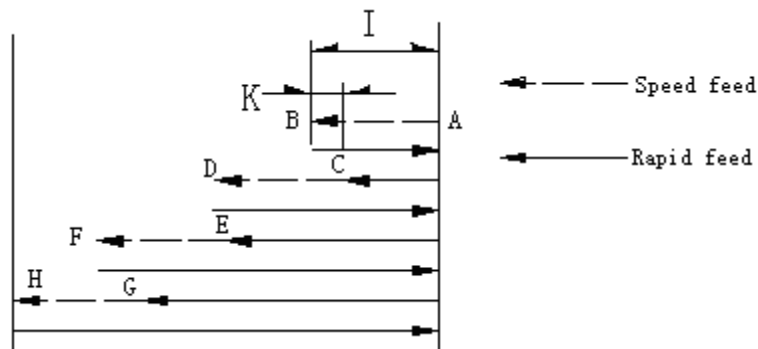


Fig. 4-30 G74 Cycle- Deep hole drilling cycle(R=1)

- ① Z feeds from A to B at F speed.
- ② Z returns to A at the rapid traverse rate.
- ③ Z feeds I-K to C at the rapid traverse rate.
- ④ Z feeds I+K at F speed from C to D.
- ⑤ Z returns to A at the rapid traverse rate.
- ⑥ Z feeds 2I-K to E at the rapid traverse rate
- ⑦ Z feeds I+K from E to F at F speed.
- ⑧ Z returns to A at the rapid traverse rate
- ⑨ Z feeds 3I-K to G at the rapid traverse rate.
- ⑩ Z feeds I+K from G to H at F speed.
- ⑪ Z returns to A at the rapid traverse rate.

When R is omitted or 0, the execution process of G74 end face deep hole machining cycle is as follows:

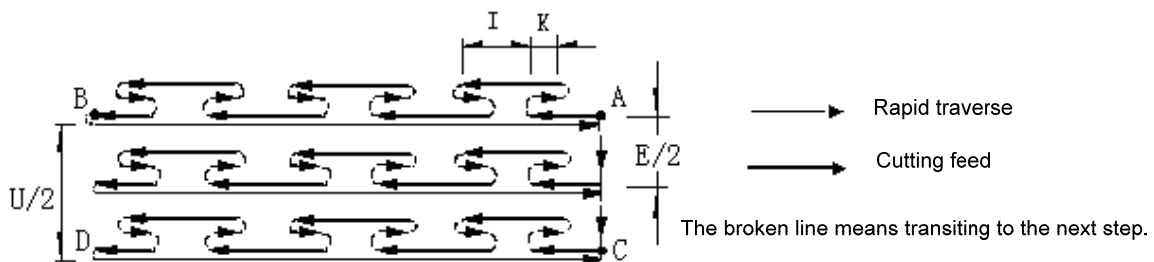


Fig. 4-31 G74 end face deep hole machining cycle

- ① Z feeds a distance of I from A point at F speed.
- ② Z retracts a distance of K.
- ③ Z feeds a distance of I+K at F speed
- ④ The above-mentioned steps 2~3 are repeated until Z feeds to B
- ⑤ Z rapidly retracts to A.
- ⑥ If X is not 0, X rapidly offsets a distance of E.
- ⑦ Repeat steps 1~4 until X, Z feed to D
- ⑧ Z rapidly returns to C, and then X rapidly returns to A.
- ⑨ The tool still stops at the starting point of the cycle when G74 cycle ends.

**【Relative parameter】**

In G74, refer to G00 for the relative parameters of rapid traverse, and refer to G01 for the relative parameters of cutting feed.

**【Note】**

- 1) In the command, the tool width has not been taken into account, so the X coordinate of the end point should be the actual end point coordinate plus or minus the tool width.(depending on the feeding direction.)

- 2) None of I, K, E has signs. The system executes the operation according to  $K = I$  when K is greater than I.
- 3) G74 cannot be in the same block without other commands. It must be specified in an independent block.
- 4) In the end face deep hole machining cycle, E cannot be 0, otherwise, the system issues No. E269 alarm; in the deep hole drilling cycle, E can be 0, which is valid.

**【Example】**

Executing end face deep hole machining cycle as Fig. 4-32: tool width= 5 mm, each tool infeed =6 mm, tool retraction= 2 mm, each offset= 5 mm, F=100 mm/min.

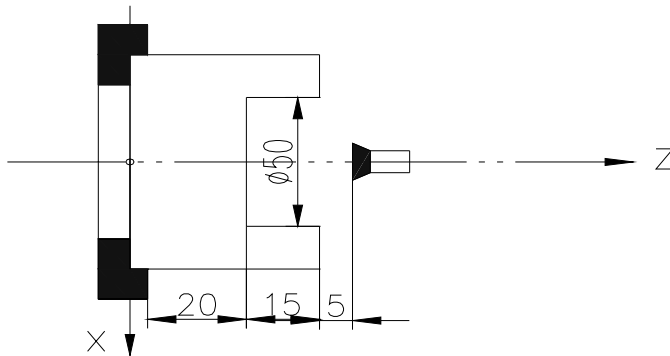


Fig. 4-34 End face deep hole machining cycle example (R=0)

```

N0010 G0 X0 Z40 ; Positioning to the tool infeed point;
N0020 G74 X22.5 Z20 I6 K2 E5 F100 ; End face machining cycle. The programming end point is
the coordinate of X end point plus tool width.
N0030 M30
    
```

4.13.5 G75 —Grooving Cycle

**【Command format】**

**G75 X (U)\_ Z (W)\_ I\_ K\_ E\_ F\_ ; Grooving cycle**

**【Word】**

- X (U) Z (W) —Groove end point coordinates.
- I — Each X tool infeed.
- K — Each X tool retraction.
- E —Each Z offset.
- F — Feedrate.

**【Word range】**

- X, Z, U, W: -9999.999mm~9999.999mm
- I, K, E: 0~9999.999mm
- F: 0.001mm/min~4000mm/min

**【Explanation】**

Execution process of G75 grooving cycle is as Fig. 4-33:

- ① X feeds a distance of I at F speed.
- ② X rapidly retracts a distance of K.
- ③ X feeds a distance of I+K at F speed.
- ④ The steps 2~3 are repeated until X feeds to B.
- ⑤ X rapidly feeds to A.
- ⑥ If Z is not 0, Z rapidly offsets a distance of E.
- ⑦ Repeat steps 1~4 until Z, X feed to D.
- ⑧ X rapidly returns to C and then Z rapidly returns to A.
- ⑨ The tool still stops at the starting point of the cycle after G75 is completed.

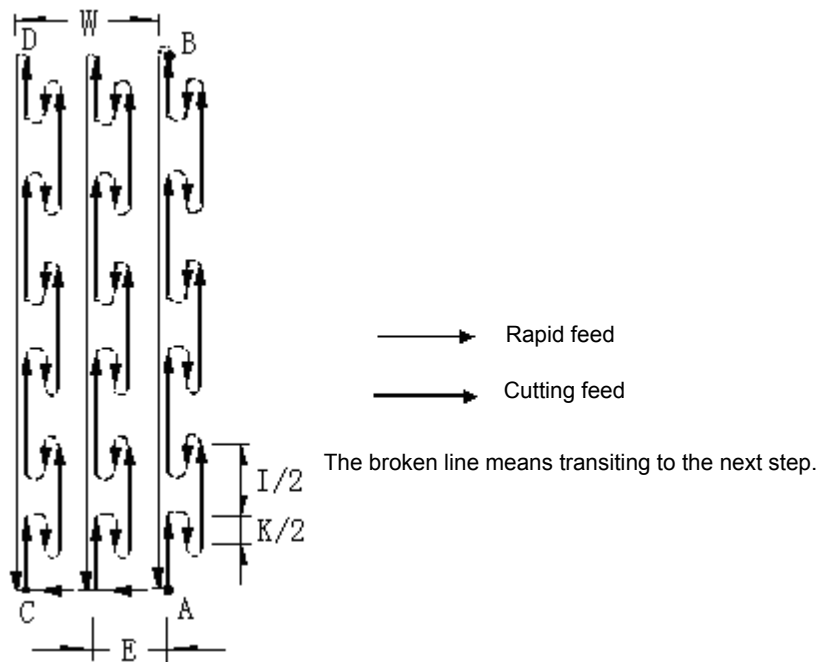


Fig. 4-33 G75 grooving cycle

**【Relative parameter】**

They are the same those of G74

**【Note】**

- 1) In G75, the tool width has not been taken into account, so the Z coordinate of the end point should adds or subtracts the tool width depending on the actual condition (it is determined by the tool infeed direction).
- 2) None of I, K, E has signs. The system executes the operation according to  $K = I$  when K is greater than I.
- 3) This command cannot be in the same block with other commands. It must be specified in an independent block.
- 4) In the outer grooving cycle, E cannot be 0. Otherwise, the system issues E269 alarm; in the outer machining cycle, E can be 0, which is valid.

**【Example】**

Executing the grooving cycle as Fig. 4-34: tool width= 5 mm, each tool infeed = 6 mm, each retraction= 2mm, each offset= 5 mm, F=150 mm/min

```

N0010 G0 X125 Z100 ; Positioning to the starting point;
N0020 G75 X80 Z35 I6 K2 E5 F150 ; Grooving cycle. The tool width should be
                                   added to the end point coordinates.
N0030 M30
    
```

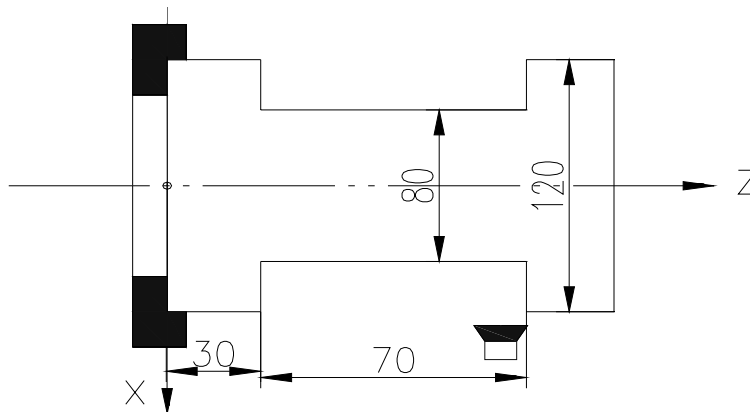


Fig. 4-34 grooving cycle

4.14 Compound Cycle

To simply the programming and reduce the calculation, the compound cycle is applied. Only the path of the machining shape needs to be given during programming, then the system can automatically determines the tool path of roughing halfway.

4.14.1 G71 —Axial Plane Rough and Finish Command Group

Using the command group can realize the compound cycle rough and finish cutting of one group of slope command collection. The system firstly starts from the starting point to rough the blank along the axial direction (the tool is parallel to Z axis), and then cuts along the command group path, and finally returns to the starting point to finish the roughing. If the roughing offset value (roughing allowance) is set in advance anytime before executing the roughing command, the system automatically offsets one offset value to execute the roughing. Anytime after executing the roughing command, the finishing tool can be used along with G710 to execute the finishing.

G71 includes the following 3 commands:

- ① G71 U W ; Using U, W to specify roughing offset value
- ② G71 X(U) I K F P Q ; G71 axial plane roughing cycle
- ③ G710 X(U) P Q F ; G710 finishing

The following introduces the above 3 commands separately.

◆ **G71** — axial plane roughing cycle

G71② is called the axial roughing cycle command, which realizes the compound cycle roughing of one group of slope command group. The system executes the gradual roughing in the axial direction (referred to as sidestep roughing), and then executes the cutting along the command group path (referred to as contour roughing), and at last returns to the initial point to complete the roughing. The section describes the execution process of G71 roughing command.(assuming the roughing offset value of command ① is: W0, U0) .

【Command format】

**G71 X (U)\_ I\_ K\_ F\_ P\_ Q\_ ;** Axial roughing cycle

【Word】

X (U) —X coordinate value of machining contour starting point.

I —Each X tool infeed, without sign; I cannot be 0.

K — Each X tool retraction, without sign.

P, Q —Block line numbers describing the starting and final path of the machining contour.

F —Roughing cutting speed.

【Word range】

X, U: -9999.999mm~9999.999mm

I: 0.001mm~9999.999mm

K: 0~9999.999mm

P, Q: 1~9999

F: 0.001mm/min~4000mm/min

【Command execution process】

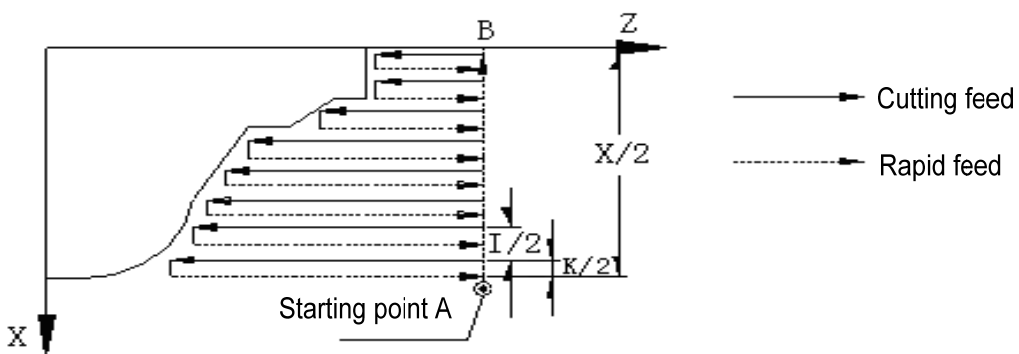


Fig. 4-35 Inner/outer roughing compound cycle

**G71 execution process:**

- ① X rapidly traverses a distance of I from starting point A in G71.
- ② Cutting feed is performed along Z axis at F speed, and the end point of the feed is calculated by the system automatically.
- ③ X retracts a distance of K at F speed.
- ④ Z rapidly returns to the starting point.
- ⑤ X feeds another distance of I+K.

- ⑥ Repeat steps ② ~ ⑤ to rough the blank until X reaches the specified machining contour starting point specified by X in the command.
- ⑦ X, Z execute the cutting at their respectively-specified speed according to the final path and machine the shape described by the final path.
- ⑧ Finally, Z rapidly returns to starting point, and then X rapidly returns to the starting point.

#### 【Relative parameter】

In G71, refer to G00 for the relative parameters of rapid traverse, and refer to the linear and circular interpolation for the relative parameters of cutting feed.

#### 【Explanation】

- ① When executing G71, the system automatically searches and executes P~Q blocks; after executing them, it proceeds to the block following G71. However, if P~Q blocks are compiled following G71, the system returns to the starting point and then executes the block following Q.
- ② Before roughing, the system will calculate the rough turning point according to programmed contour. To prevent imprint on roughing points, there should be an allowance between roughing point and contour: 0.03 in X direction, 0.05 in Z direction.
- ③ In P~Q blocks can only include G commands: G00, G01, G02, G03, G05, G04, G96, G97, G98, G99; and common input/output control M commands; other commands are not allowed (e.g. T, transfer, call command). The system only uses Z/X programming. The number of paths in P~Q blocks cannot exceed 1000.
- ④ F, S command in P~Q blocks are invalid when executing the sidestep roughing, and they are valid only in the contour path; thus, F speed should be specified in advance or F is programmed in the same block with G71 in the sidestep roughing.
- ⑤ In P~Q blocks, X, Z dimension data must be monotonous slope-type change (always increasing or reducing) ; and X must starts from the machining contour starting point B to monotonously change to G71 starting point A.
- ⑥ None of address I, K has signs, and the tool infeed direction is automatically determined by the system. The smaller I is, the more the roughing layers are.
- ⑦ I range is related to X(U) range; I, U range must meet the requirement that U/I is smaller than 10000.
- ⑧ When executing feed hold, single block, the system pauses after it executes the path end point of the current step.
- ⑨ This command cannot be in the same block with other commands. It must be specified in an independent block.
- ⑩ There are four cases when using G71 command to cut the workpiece, as in Fig. 4-36; For each one, the tool cuts from G71 starting point A and roughs the blank in parallel with Z till B is reached.



- ⑪ Generally, X coordinate of BCD section should be within the range between B and A; when D exceeds A, the system does not rough the exceeded part.
- ⑫ There are four cases for the shapes cut by G71 command: in Fig. 4-36, A is G71 starting point, B is the starting point of machining contour, D is end point of machining contour, BCD section is the machining path.

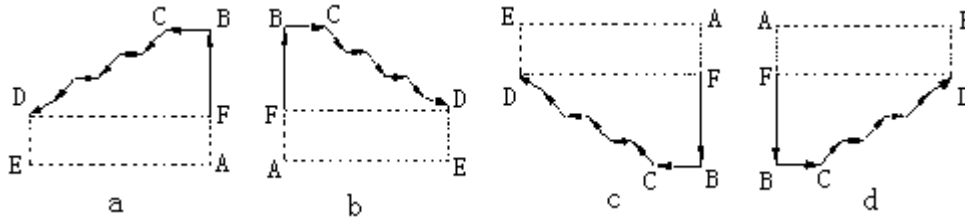


Fig. 4-36

**【Example】**

As Fig. 4-37: bar stock  $\Phi 82$ , each tool infeed = 4 mm, each tool retraction = 2.5 mm, F = 60mm/min

```

N0000 G00 X115 Z155 ; Positioning to the starting point
N0010 M3 S02 ; Start the spindle, set to spindle high speed
N0020 M8 ; Cooling ON
N0030 G00 X83 ; X tool infeed to approach workpiece
N0040 G71 X0 I4 K2.5 F100 P0090 Q0160 ; Self defining roughing cycle parameter
N0050 G00 X115 Z155 ; Return to tool starting point
N0060 M5 ; Stop the spindle
N0070 M9 ; Cooling OFF
N0080 M2 ; End of program
N0090 G01 Z145 F60 ;
N0100 X15 ;
N0110 W-30 ;
N0120 G03 X55 W-20 I0 K-20 ; Defining the final path
N0130 G01 W-25 ;
N0140 G01 X80 W-20 ;
N0150 W-50 ;
N0160 X81 ;
    
```

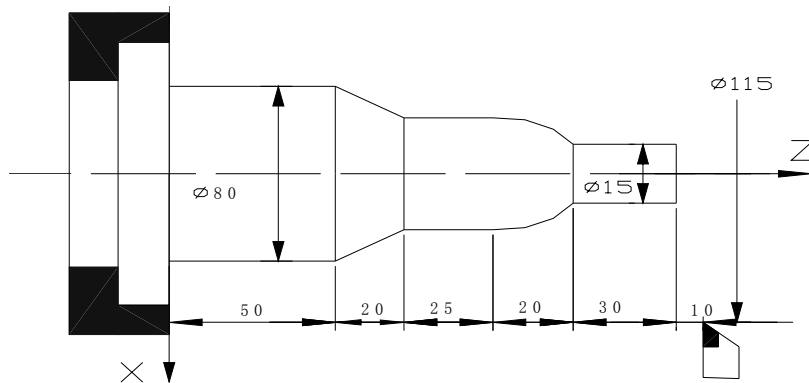


Fig. 4-37 G71 cutting example

### ◆ G71 roughing offset command and G710 finishing command

G71 includes 3 commands. When using command ① to set the roughing offset value (roughing allowance) in advance anytime before executing the roughing command ②, the coordinate axis firstly offsets one offset value automatically and then executes the roughing path during the execution of command ②, i.e. the roughing path creates the whole offset to leave the allowance for the next finishing. When command ① is omitted, the roughing offset value is W0,U0. Anytime after the roughing, the user can use a finishing tool to perform finishing by using command ③.

#### 【Command ① format and definition】

G71 U\_ W\_ ; Specifying the roughing offset value

Thereinto: U— X offset value; U: -9999.999mm~9999.999mm;

W— Z offset value; W: -9999.999mm~9999.999mm.

#### 【Command ③ format and definition】

**G710 X(U)\_ P\_ Q\_ F\_ ;** G710 finishing

Or **G710** ; The omission of the subsequent words indicates the words are the same as those of roughing command ②.

X(U) — X coordinate value of starting point of finishing contour.

P, Q — Block line numbers describing the initial to the final path of finishing contour.

F —Finish cutting speed.

#### 【Explanation】

- 1) In executing command ①, the coordinate axis does not move actually, the system automatically memorizes the offset value to the internal and the value keeps valid; it is 0 by default when it is not specified.
- 2) The offset value set by G71 command ① is only valid for G71 command ②.
- 3) U, W words in G71 command ① must be specified simultaneously.
- 4) X(U), P, Q words in G71 command ③ should be the same as those of G71 command ②; if they are re-input, they must be input completely; the word range is the same as that of command ②.
- 5) In G71 command ③, the command programming rules in P~Q blocks are the same as those of G71 command ②.
- 6) When executing the roughing G71 command ②, the system automatically memorizes X(U), P, Q data of the latest G71; if X(U), P, Q are omitted in G710, the system directly refers the data of the internal memory. Therefore, the words following G710 can be omitted when the system executes finishing to the latest roughed G71 path.
- 7) When the system executes G710 finishing, it performs the cutting along the command group path, and finally returns to the initial point, thus completing the finishing.

- 8) The tool nose position before executing G710 finishing should be consistent with that before G71 roughing
- 9) Coordinate offset direction in reserving roughing allowance: U,W in command ① expresses the coordinate offset and cut-in direction in finishing; there are four kinds of combination according to the signs of U, W (see Fig. 4-38): B→C is finishing path, B'→C' is roughing contour, A is the start-up point.

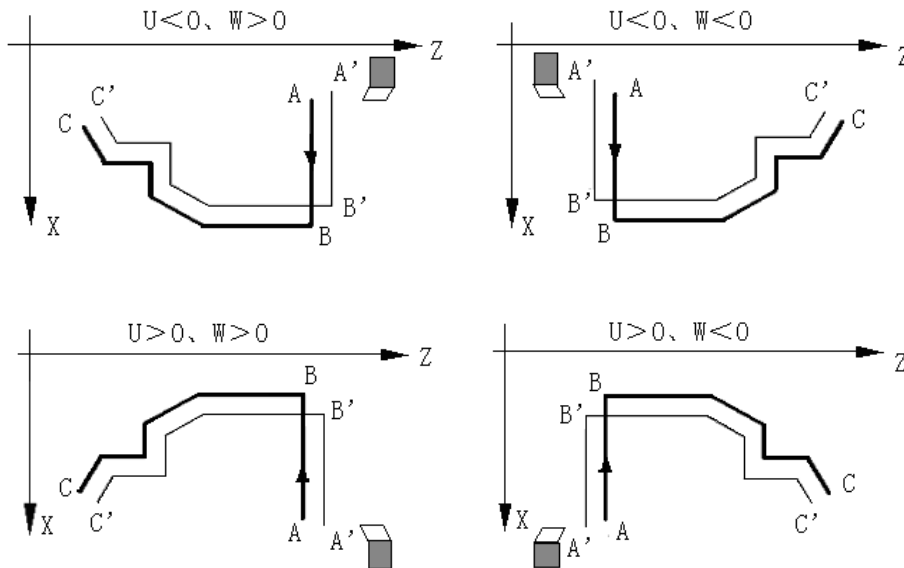


Fig. 4-38

II PROGRAMMING

#### 4.14.2 G72 —End Face Roughing/Finishing Command Group

Using the command group can realize the compound cycle rough and finish cutting of one group of slope command collection. The system firstly starts from the starting point to rough the blank along the axial direction, and then cuts along the command group path, and finally returns to the starting point to complete the roughing. When the roughing offset value (roughing allowance) is set in advance before executing the roughing command, the system automatically offsets one offset value and then performs roughing. The finishing tool can be used along with G720 to execute the finishing anytime after executing the roughing command.

G72 command group includes 3 commands:

- ① G72 U W ; Specifying roughing offset value using U, W
- ② G72 Z(W) I K F P Q ; G72 end face roughing cycle
- ③ G720 Z(W) P Q F ; G720 finishing

They are respectively described as follows:

◆ **G72 —end face roughing cycle**

The ② command in G72 command group is called the axial roughing cycle command, which can realize the compound cycle roughing of one group of slope command group. The system executes the gradual roughing along the radial direction (called sidestep roughing), and then executes the cutting

along the command group path (called contour roughing), and finally returns to the initial point to complete the roughing. This section describes the execution process of G72 roughing command (assuming the roughing offset value of command ① is: W0, U0).

**【Command format】**

**G72 Z(W) \_ I\_ K\_ F\_ P\_ Q\_ ;** End Face Roughing Cycle

**【Word】**

Z(W) — Z coordinate value of the starting point of machining contour

I — Each Z tool infeed, without sign; I cannot be 0.

K — Each Z tool retraction, without sign.

P, Q — Block line numbers describing initial and final path of machining contour.

F — Roughing cutting feedrate.

**【Word range】**

Z, W: -9999.999mm~9999.999mm

I: 0.001mm~9999.999mm

K: 0~9999.999mm

P, Q: 1~9999

F: 0.001mm/min~4000mm/min

**【Relative parameters】**

They are the same as those of G71 command

**【Explanation】**

- 1) The reserved allowance between G72 command roughing point and contour: 0.05 in X direction, 0.03 in Z direction; other explanations are the same as those of G71.

**【Command execution process】**

G72 cycle execution process is as Fig. 4-39:

- ① Start from G72 starting point A, Z rapidly traverses a distance of I.
- ② X cuts along X axis at F speed, and the feeding end point is counted automatically by the system.
- ③ Z retracts a distance of K at F speed.
- ④ X rapidly returns to the starting point;
- ⑤ Z feeds another distance of I+K;
- ⑥ Repeat steps ② ~ ⑤ to rough the blank gradually till Z reaches the starting point B of the machining contour specified by Z in the command.
- ⑦ X, Z perform cutting according to the final path at their respectively-specified speed and machine the shape described as the final path.
- ⑧ Finally, X rapidly returns to the starting point, and then Z rapidly returns to the starting point.

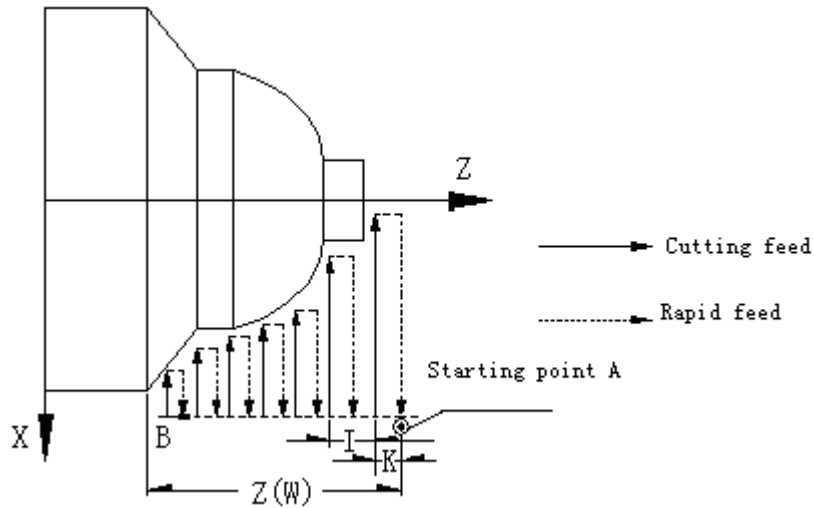


Fig. 4-39 G72 end face roughing compound cycle

【Example】

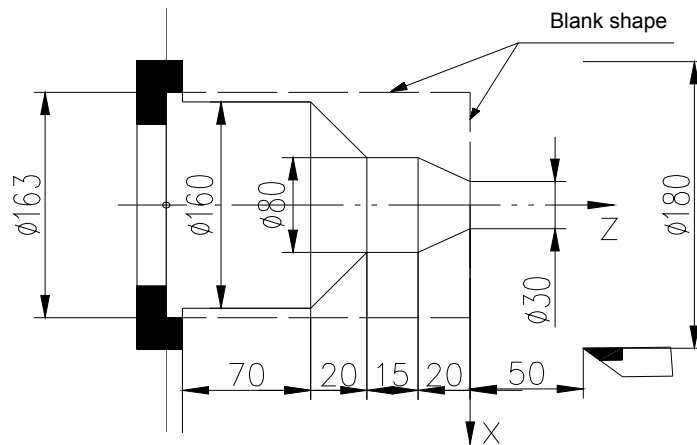


Fig. 4-40 G72 cutting example

As Fig. 4-40 : bar  $\Phi 163$ , each tool infeed =5mm, each tool retraction =3 mm, F=80mm/m

```

N0000 G00 X180 Z50 ; Positioning to the starting point
N0010 M3 S02 ; Start the spindle, set to spindle speed 2
N0020 M8 ; Cooling ON
N0030 G00 X165 Z5 ; Tool infeed to approach workpiece
N0040 G72 Z-125 I5 K3 F80 P0100 Q0150 ; Defining roughing cycle parameter
N0050 G00 Z50 ; Z returns to machining starting point
N0060 X180 ; X returns to machining starting point
N0070 M5 ; Spindle stop
N0080 M9 ; Cooling OFF
N0090 M2 ; End of program
N0100 G01 X160 ;
N0110 Z-55 ;
N0120 X80 Z-35 ;
N0130 Z-20 ;
N0140 X30 Z0 ;
N0150 X0 ;
    
```

} Defining final path

◆ **G72 roughing offset command and G720 finishing command**

G72 includes 3 commands. When using command ① to set the roughing offset value (roughing allowance) in advance anytime before executing the roughing command ②, the coordinate axis firstly offsets one offset value automatically and then executes the roughing path during the execution of command ②, i.e. the roughing path creates the whole offset to leave the allowance for the next finishing. When command ① is omitted, the roughing offset value is W0,U0. Anytime after the roughing, the user can use a finishing tool to perform finishing by using command ③.

【Command ① format and meaning】

**G72 U\_ W\_ ;** Specifying the roughing offset value  
 Thereinto: U — X offset value; U range: -9999.999mm~9999.999mm;  
 W — Z offset value; W range: -9999.999mm~9999.999mm.

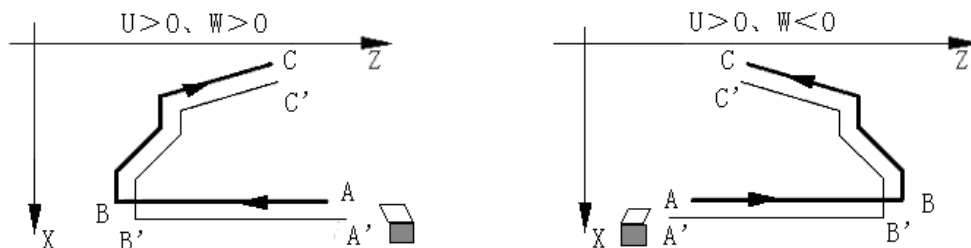
【Command ③ format and meaning】

**G720 Z(W)\_ P\_ Q\_ F\_ ;** G720 finishing  
 Or **G720 ;** The omission of the subsequent words indicates they are the same as those of roughing command ②

Z(W) — Z coordinate value of starting point of finishing contour.  
 P, Q — Block line numbers describing the initial to final path of the finishing contour.  
 F — Finishing cutting speed.

【Explanation】

- 1) G720 gradually cuts the workpiece according to the tool being parallel with X axis.
- 2) The other explanations of G720 are same as those of G71 and G710.
- 3) Coordinate offset direction in reserving roughing allowance: U,W in command ① expresses the coordinate offset and cut-in direction in finishing; there are four kinds of combination according to the signs of U, W, as Fig. 4-41: B→C is finishing path, B'→C' is roughing contour, A is the start-up point.



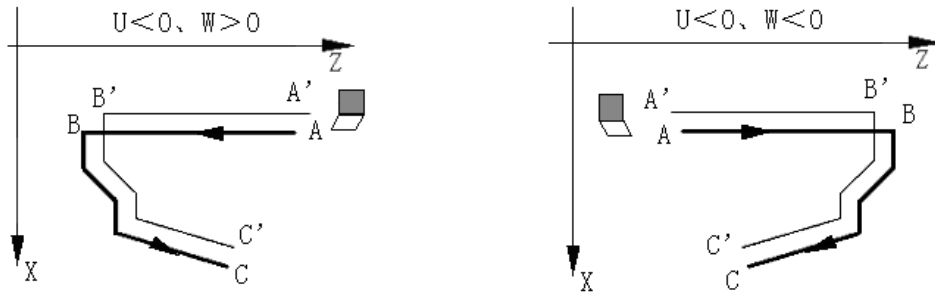


Fig. 4-41

### 4.14.3 G73 — Closed-loop Cutting Cycle Command Group

This command group can realize the closed-loop compound cycle roughing and finishing. The system gradually cuts the blank and then cuts along the command group path, and finally returns to the starting point, to complete the roughing. When the roughing offset value (roughing allowance) is set in advance before executing the roughing command, the system automatically offsets one offset value to execute the roughing. The user can use a finishing tool along with G730 command to execute finishing anytime after executing the roughing command.

G73 starting point is identical with the end point, and the command is applicable to the roughing of formed blank. G73 is a one-shot command and its path is as Fig. 4-42.

- ① G73 U W ; Specifying roughing offset value using U, W
- ② G73 X(U) Z(W) I K F P Q L ; G73 closed-loop roughing cycle
- ③ G730 X(U) Z(W) P Q F ; G730 finishing

The commands above are respectively described as follows:

#### ◆ G73 — closed roughing cycle

The ② command in G73 command group is called the closed-loop roughing cycle command which can realize the compound cycle roughing of one group of closed-loop command group. The system firstly cuts the blank (called step roughing) gradually from the starting point, and then cuts along the command group path (called contour roughing) and finally returns to the initial point, completing the roughing.

The section describes the execution process of G73 roughing command (assuming roughing offset value of command ① is  $W=0, U=0$ ).

#### 【Command format】

**G73 X (U)\_ Z(W)\_ I\_ K\_ L\_ P\_ Q\_ F\_ ;** Closed-loop roughing cycle

#### 【Word】

X (U) — X coordinate value of starting point B of roughing contour.

Z (W) —Z coordinate value of starting point B of roughing contour.

- P — Initial block line number describing the roughing contour.
- Q — Final block line number describing the roughing contour.
- F — Roughing cutting feedrate.
- I — Retraction value on X axis during the first roughing (with sign). Refer to the distance I in Fig. 4-44.
- K — Retraction value on Z axis during the first roughing (with sign). Refer to the distance K in Fig. 4-44.
- L — Cutting times, i.e. the number of gradual cutting layers.

X(U), Z(W), P, Q describe the ABCA closed path in the figure, i.e. the tool rapidly positions from A to B, and then the system performs PQ blocks to C, and finally the tool returns to A rapidly. ABCA is the path of the last roughing.

I, K in the figure below describe the system firstly offsets I then K value and then executes the first roughing; i.e. the system executes  $A_1B_1C_1A_2$  path. L describes that L layers need to be performed from the first roughing before the last roughing is performed; the infeed amount for each layer should be even, i.e. the amount of each layer is  $I/L$  in X direction, and is  $K/L$  in Z direction. The tool gradually approaches the ABC layer from the  $A_1B_1C_1$  layer. The actual cutting times are  $L+1$ .

**【Word range】**

- X, U, Z, W: -9999.999mm~9999.999mm
- I, K: -9999.999mm~9999.999mm
- P, Q : 1~9999
- L: 1~9999
- F: 0.001mm/min~4000mm/min

**【Execution process】** as Fig.4-44. Suppose  $L=2$  layers.

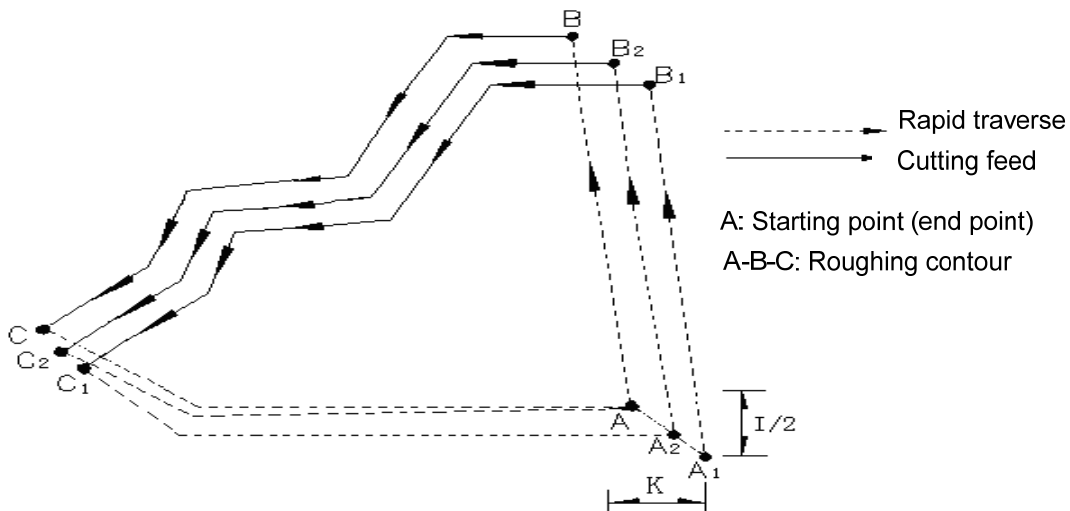


Fig. 4-42 G73 command execution path



- ① A→A<sub>1</sub>: Rapidly traverse, offset I, K, to A<sub>1</sub>;
- ② **The first roughing, A<sub>1</sub>→B<sub>1</sub>→C<sub>1</sub> :**  
 A<sub>1</sub>→B<sub>1</sub>: Rapidly traverse to B<sub>1</sub>;  
 B<sub>1</sub>→C<sub>1</sub>: Cutting feed;  
 C<sub>1</sub>→A<sub>2</sub>: Rapidly traverse;
- ③ **The second roughing, A<sub>2</sub>→B<sub>2</sub>→C<sub>2</sub> :**  
 A<sub>2</sub>→B<sub>2</sub>: Rapidly traverse;  
 B<sub>2</sub>→C<sub>2</sub>: Cutting feed;  
 C<sub>2</sub>→A: Rapidly traverse;
- ④ **The last roughing:**  
 A→B: Rapidly traverse;  
 B→C: Cutting feed;  
 C→A: Rapidly traverse to the starting point.

#### 【Command explanation】

- 1) P~Q blocks are different from those of G71/G72. It is unnecessary to ensure the dimension data monotonous change in X and Z directions.
- 2) The tool retraction direction is determined by the signs of I, K.
- 3) The F, S commands in P~Q blocks are valid for both workblank roughing and finishing.
- 4) For other explanations, refer to G71, G72.

#### ◆ G73 roughing offset command and G730 finishing command

G73 includes 3 commands. When using command ① to set the roughing offset value (roughing allowance) in advance anytime before executing the roughing command ②, the coordinate axis firstly offsets one offset value automatically and then executes the roughing path during the execution of command ②, i.e. the roughing path creates the whole offset to leave the allowance for the next finishing. When command ① is omitted, the roughing offset value is W0,U0. Anytime after the roughing, the user can use a finishing tool to perform finishing by using command ③.

#### 【Command ① format】

**G73 U\_ W\_ ;** Specifying the roughing offset value  
 U— X offset value ; U range: -9999.999mm~9999.999mm;  
 W— Z offset value; W range: -9999.999mm~9999.999mm。

#### 【Command ① word definition】

U: X offset value (roughing allowance), which means X coordinate offset of the last roughing path relative to the finishing path;

W: Z offset value (roughing allowance), which means Z coordinate offset of the last roughing path relative to the finishing path.

**【Command ③ format and definition】**

**G730 X(U)\_ Z(W)\_ P\_ Q\_ F\_ ;** G730 finishing

Or **G730** ; The omission of the subsequent words indicates they are the same as those of roughing command ②

X (U) — X coordinate value of starting point of finishing contour.

Z (W) —Z coordinate value of starting point of finishing contour.

P, Q — Block line numbers describing the initial and final path of the finishing contour.

F —Finishing cutting feedrate

**【Explanation】**

- 1) The other explanations of G730 are the same as those of G71 G72 and those of G710 and G720.
- 2) When executing G730 finishing, the tool cuts along the command group path, and finally returns to the starting point to complete the finishing.
- 3) When the finishing positioning point does not coincide with the roughing positioning point, the tool execution path will automatically coincide.
- 4) Coordinate offset direction in reserving roughing allowance:

I, K, U, W express the coordinate offset and cut-in direction in roughing and finishing; In general, the sign of I is the same as that of U, while the sign of K is the same as that of W; generally, there are four kinds of combination for I, K, U, W, as fig. 4-43: A is the start-up point, B→C is the workpiece contour, B'→C' is the roughing contour, and B''→C'' is the finishing path.

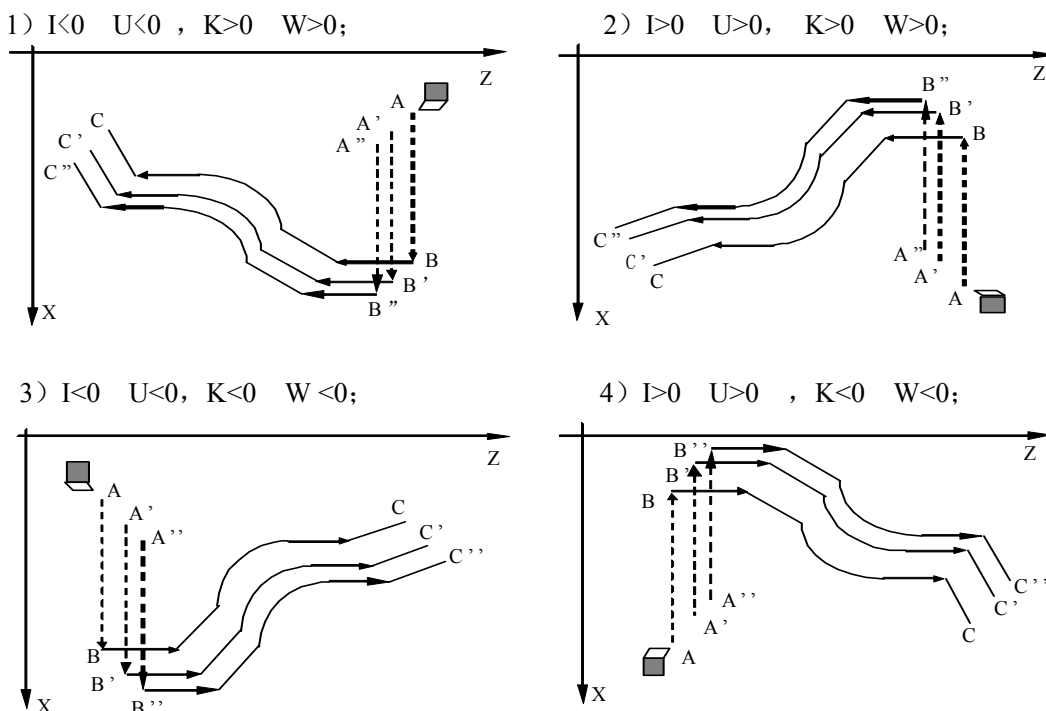


Fig. 4-43

【Example】

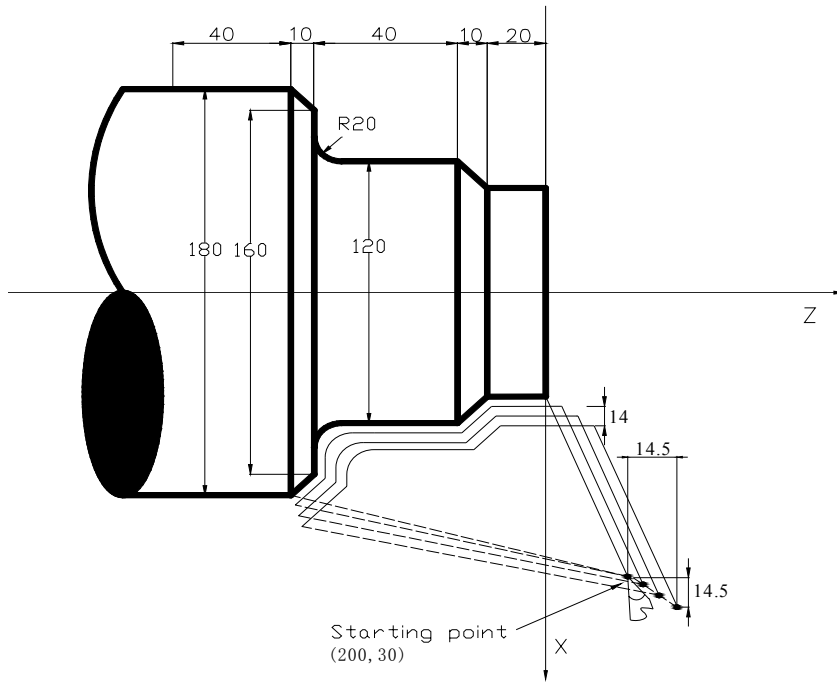


Fig. 4-44

Cutting figure is shown as the shape in fig. 4-44:

```

N0010 T11
N0020 G00 X200 Z30 ; Positioning starting point
N0022 M03 S500 ; Start the spindle
N0030 G73 U0.5 W0.5 ; An allowance of 0.5 mm for X, 0.5 mm for Z
N0040 G73 X80 Z0 P50 Q100 I14 K14 L5 F300 ; X tool retraction 14mm, Z 14mm in roughing

N0050 G01 Z-20 F100
N0070 X120 Z-30
N0080 Z-50
N0090 G02 X160 Z-70 R20
N0100 G01 X180 Z-80
N0110 T22
N0120 G00 X200 Z30
N0130 G730 ; Finishing with finishing tool
    
```

} ; Workpiece shape blocks

4.14.4 G76 —Multiple Repetitive Threading Cycle

The cutting path of G76 threading cycle is almost the same as that of G92. The difference is that G92 completes the cutting depth by the subsequent command while G76 informs the last cutting face and the relative information about the gradual cutting depth in advance and then the gradual cutting depth is arranged by the system to machine the qualified thread.

G76 command group includes 2 commands:

- ① G76 D\_ I\_ Q\_ P\_ L\_ R\_ ; G76 cycle information
- ② G76 X (U) \_ Z (W) \_ P (E) \_ I\_ K\_ R\_ L\_ H\_ ; G76 threading

The commands above are respectively described as follows:

◆ **G76** — multiple repetitive threading cycle command ②

The ② command of G76 command group is called the multiple repetitive threading cycle command, which can realize the multiple thread cutting.

When the thread angle is 0°, in the process of each threading cycle of G76 from the starting point to end point, the thread axis does not move, and the non-thread axis moves to the cut-in point independently. The cutting path is the same as that of G92.

【Command ② format】

**G76 X (U) \_ Z (W) \_ P (E) \_ I\_ K\_ R\_ L\_ H\_ ; G76 threading**

【Command ② definition explanations】

Word definitions, data range of this command are the same as those of G92, as Fig. 4-47:

**Starting point (end point):** Position before the block is executed and after the execution ends, which is expressed by point A.

**Thread end point:** Threading end point defined by X(U) \_ Z(W) \_, which is expressed by point C.

**Thread starting point:** Z absolute coordinate is same as that of A point, and the difference between X absolute coordinate and that of C point is the thread taper, expressed with E. When the defined thread angle is not 0°, the tool cannot reach E point in cutting.

**Tool retraction end point:** Radial (X axis) retraction end point after threading in each thread roughing cycle or finishing cycle, which is expressed by D point.

◆ **G76** — multiple repetitive threading cycle command ①

The ① command of G76 command group is used for informing relative information about threading in advance.

【Command ① format and definition】

**G76 D\_ I\_ Q\_ P\_ L\_ R\_ ; G76 cycle information**

D—Tooth depth, total depth range: (0.001~9999.999) mm ; Not omissible; D cannot be negative

I—The first cut depth in roughing range:( 0.001~9999.999) mm; Not omissible; I cannot be negative

Q—Least cutting amount in roughing range: (0~99.999) mm ; Omissible, corresponding to P339

P— Tool angle range: (0~99.000) degree ; Omissible, corresponding to P336

- L — Number of finishing times      range: 1~99      ; Omissible, corresponding to P337  
 R — Finishing cutting amount      range: (0~99.999) mm      ; Omissible, corresponding to P338

**【Command ① relative definitions】**

- D: Thread tooth depth, and the thread total cutting depth. The system issues an alarm when D is not input.
- I: The first threading depth. The system issues an alarm when I is not input.
- Q: Least cutting amount in thread roughing. n is the cycle times in the current roughing; when  $I \times \sqrt{n}$  - the last cutting depth) < Q, Q is taken as the cutting amount of this roughing, i.e. this thread cutting depth is (the last cutting depth+Q). Setting Q is to prevent too small roughing cutting amount and too many roughing times because of the thread cutting amount gradually decreasing.
- P: Angle between neighboring two threads. The actual thread angle is determined by the tool angle, and so P value should be consistent with the tool angle.
- L: Thread finishing times.
- R: Cutting amount in thread finishing, which is equal to the difference between the X absolute value of the thread finishing cut-in point  $B_e$  and that of the last thread roughing cut-in point  $B_f$ . In the thread finishing, the first cutting amount is R and the following finishing amount is 0.

**【Command ① explanation】**

- 1) When the system executes the command, it loads P, L, R, Q values to their corresponding parameters, because these parameters are used when the system executes the command ②.
- 2) When the system executes the command, D, I are memorized to the system internal and are used in G76 ②.
- 3) The system must firstly execute one G76① command, and then can execute one or more G76 ② commands. Otherwise, an alarm is issued because of lacking some necessary information.
- 4) All P, L, R, Q can be omitted. When they are omitted, the system runs according to the setting values of P336, P337, P338, P339 in executing G76②; D, I, however, cannot be omitted.

**Relative definitions:** (as Fig. 4-45)

**Thread cut depth reference point:** Z absolute coordinate of the thread cut depth reference point is the same as that of E, and the difference between its X absolute coordinate and X absolute coordinate of E is D (total thread cut depth). The thread cut depth reference point is expressed by B. Thread cut depth of B is 0, and is the reference point for the system to count each thread cut depth.

**Thread cut depth:** Cut depth of each threading cycle. The cut depth of the thread is the difference (without sign) between X absolute value of the intersection point (between the reserve extension line of each threading path and line BE) and that of B. The thread cut

depth of each roughing is  $\sqrt{n} \times I$ . Thereinto,  $n$  is the current roughing cycle times, and  $I$  is the thread cut depth of the first roughing.

**Threading amount:** The difference between current thread cut depth and last one:  $(\sqrt{n} - \sqrt{n-1}) \times I$ . The cutting amount cannot be less than the least cutting amount  $Q$ .

**Thread cut-in point:** The point from which the actual threading starts in each thread roughing and finishing, which is expressed by  $B_n$  ( $n$  is the current roughing cycle times).  $B_1$  is the cut-in point of the first thread roughing,  $B_f$  is the cut-in point of the last thread roughing, and  $B_e$  is the cut-in point of the thread finishing. The displacement formula of  $B_n$  relative to  $X$ ,  $Z$  of  $B$  is as follows:

$$\operatorname{tg} \frac{P}{2} = \frac{Z \text{ axis displacement}}{X \text{ axis displacement}}$$

$P$ : Thread angle

#### 【Command ② execution process】

- ① The tool rapidly traverse to  $B_1$  from the starting point, and the first thread cut depth is  $I$ . If  $P=0$ , only  $X$  axis moves; if  $P \neq 0$ ,  $X$  and  $Z$  move simultaneously, and their movement directions are the same as that of  $A \rightarrow C$ ;
- ② Threading to the intersection with  $C \rightarrow D$  in the direction parallel with  $E \rightarrow C$  (when there is run-out, there is the thread run-out process);
- ③  $X$  axis rapidly traverses to  $D$ ;
- ④  $Z$  axis rapidly traverses to  $A$ , then the single roughing cycle is completed;
- ⑤ The tool rapidly traverses again to  $B_n$  ( $n$  is the current roughing cycle times), and the cut depth is  $(\sqrt{n} \times I)$  or (the last cut depth +  $Q$ ), whichever is greater. If the cut depth is less than  $(D-R)$ , the system shifts to ②; if the cut depth is greater than or equal to  $(D-R)$ , the tool feeds the cut depth  $(D-R)$  to  $B_f$ , and the system shifts to ⑥ to execute the last thread roughing;
- ⑥ Threading to the intersection with  $C \rightarrow D$  in the direction parallel with  $E \rightarrow C$  (when there is run-out, there is the thread run-out process);
- ⑦  $X$  axis rapidly traverses to  $D$ ;
- ⑧  $Z$  axis rapidly moves to  $A$ , then the thread roughing cycle ends and the thread finishing starts;
- ⑨ The thread finishing is executed after the tool rapidly moves to  $B_e$  (thread cut depth is  $D$ , the cutting amount is  $R$ ), and finally the tool returns to  $A$ , completing one thread finishing cycle;
- ⑩ If the finishing cycle times is less than  $L$ , the system shifts to ⑨ execute the next finishing cycle, and the thread cut depth is  $D$ , the cut amount is  $0$ ; if the finishing cycle times is  $L$ , G76 compound thread machining cycle ends.

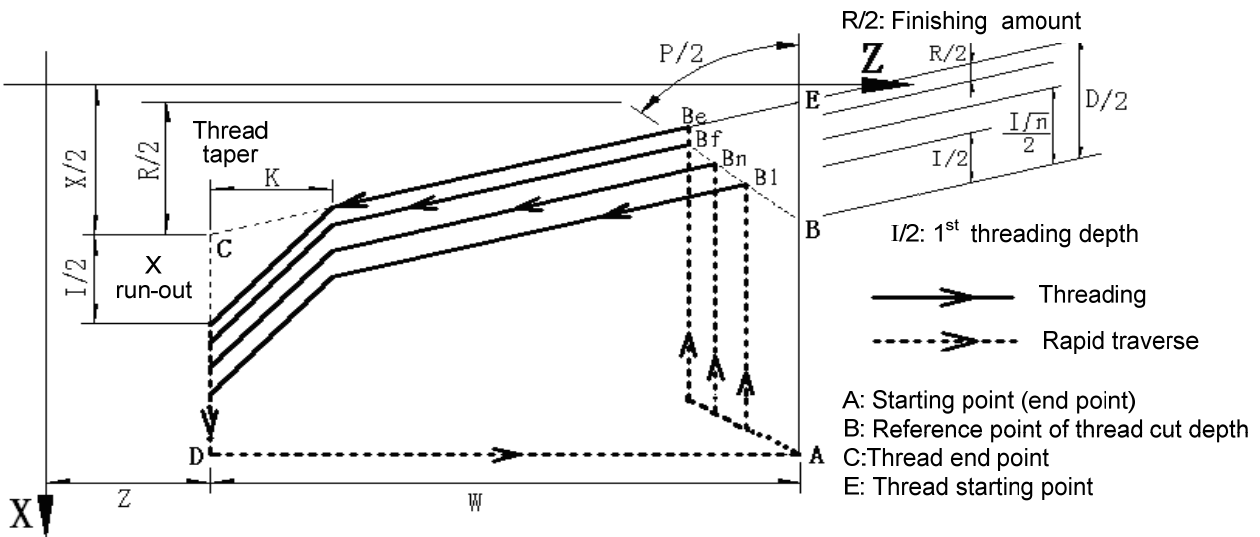


Fig. 4-45

The cut-in method is shown in Fig. 4-46:

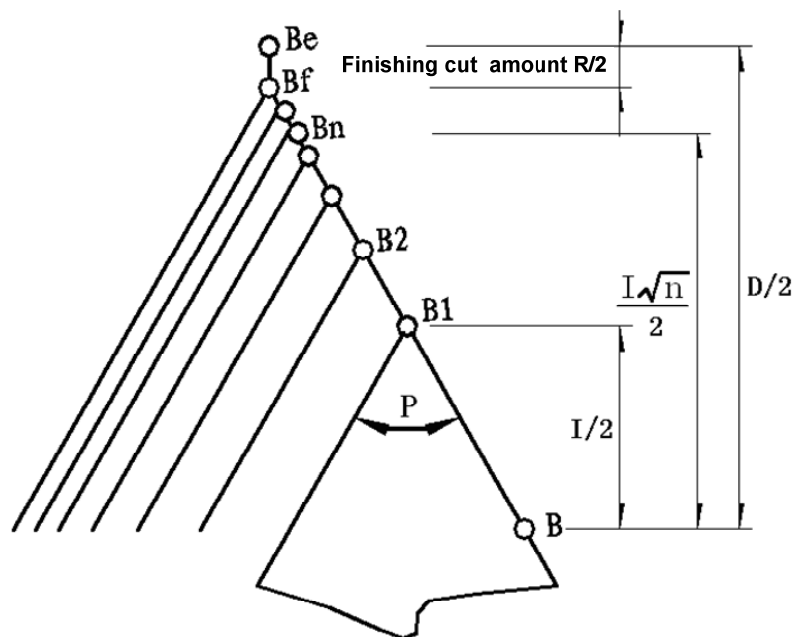


Fig. 4-46

**【Command ② explanation】**

- 1) Using G76 thread cycle cutting command group can complete the thread cutting with specified tooth height (total cut depth) by multiple thread roughing and finishing; when the thread angle defined by the command ① is not 0°, the cut-in point of thread roughing gradually moves to the thread tooth bottom from the thread tool top, so that the angle between two neighboring teeth is the specified thread angle.
- 2) G76 command group can machine the straight and taper thread with the thread run-out and can realize the single-side tool edge threading, which protects the tool and improves the thread precision.

- 3) G76 command group can also machine multiple thread and end face thread. When G76 machines the multiple thread, each time the cut depth is fed, each thread is cut the same cut depth till the multiple thread is completed. Then the system executes the next cut depth.
- 4) If the thread angle defined by G76 command ① is not 0°, the difference between G76 command ② and G92 is: in the process of each thread cycle of G76 ② from the starting point to cut-in point, X, Z move simultaneously at first, and the thread axis reaches the cut-in point, and then the non-thread axis moves to the cut-in point independently; the other cut path is the same as that of G92 . See Fig. 4-46.

**【Example】**

Example: As fig. 4-47, the thread is M68×6.

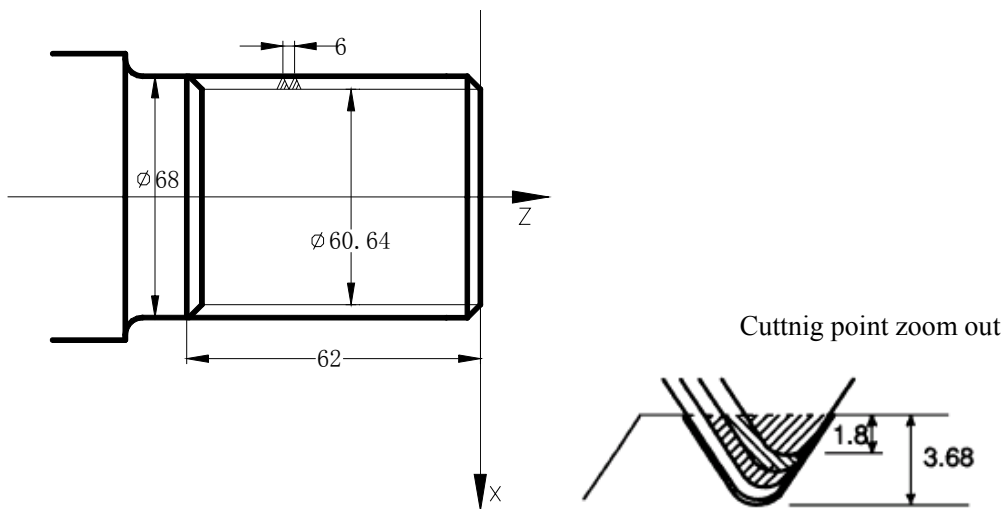


Fig. 4-47

Program:

```
G50 X100 Z50 M3 S300 ; Sets workpiece coordinate system, starts the spindle
                        and specify its speed
G00 X80 Z10 ; Rapidly moves to the machining start point
G76 D7.36 I3.6 Q0.3 P60 L2 R0.2 ; Thread tooth height 7.36, the 1st thread cut depth 3.6;
                        ; The least cut-in depth 0.3, tool angle 60°,
                        ; Repetitive finishing 2 times, finishing cutting amount 0.2.
G76 X60.64 Z-62 P6 I0.5 L3 ; Pitch 6, 3-head thread
G00 X100 Z50 ; Returns to program start point
M30 ; End of program
```

#### 4.15 G22, G80 —Program Local Cycle

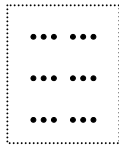
In the course of actual machining, for the workpiece part of which needs to be repetitively machined, or the basically-formed workpiece, the user can employ local cycle command to simplify the programming. The cycle body of the local cycle is programmed by the user, and the end point



coordinates after execution are determined after the program execution.

**【Command format】**

**G22 L\_**



Cycle body program

**G80**

**【Word】**

**G22** defines the starting of cycle body;

**L:** defines the cycle times, range 1~9999;

**G80** defines the end of cycle body.

**【Explanation】**

**Command execution process:**

- ① G22 defines the starting of cycle body and L defines the cycle times.
- ② Execute the cycle body program.
- ③ When G80 cycle body ends, if L is not 0, it reduces by 1 and then the system executes the cycle body program again; if L is 0, the cycle ends, and the system executes the following program in order.

**【Note】**

- 1) Only when G22 and G80 are used together, can the cycle body be formed; there can be G22 and G80 in the cycle body, i.e. they can be nested.
- 2) The subprogram can be called in the loop body, which can also include M96 and M97. For the parts whose shapes have been determined and which need to be roughed, using G22 and G80 to program is very convenient and can improve the machining efficiency.
- 3) For the axis needing to create the cycle offset, the program in the cycle body uses the relative programming. And by making an offset between the exit coordinates and entry coordinates of the cycle body, the user can realize the purpose that each time the cycle program and machining contour are the same, but the machining path is different.
- 4) G22, G80 can nest G90, G92, G94, G71, G72 and other cycle commands.

**【Example】**

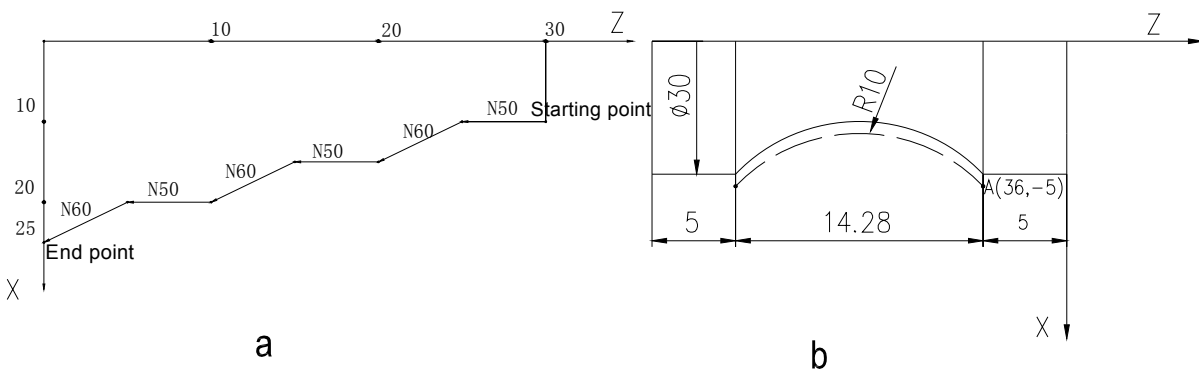


Fig. 4-48

It is convenient to program using G22, G80 cycle when machining the workpiece whose shape is shown in Fig. 4-48a. The program is as follows:

```

N0000 G50 X100 Z100 ; Defines coordinate system
N0010 M3 S01 ; Starts the spindle, sets to the spindle low speed
N0020 M8 ; Cooling ON
N0030 G00 X10 Z30 ; Rapid positioning to cycle start point
N0040 G22 L3 ; Program cycle starts and cycles 3 times
N0050 G01 W-5 F50 ; Z negatively cuts 5mm, F=50mm/min
N0060 U5 W-5 ; X positively cuts 5mm, Z negatively cuts 5mm
N0070 G80 ; End of cycle body
N0080 G26 X100 Z100 ; X, Z rapidly retract to program reference point
N0090 M5 ; Stops the spindle
N0100 M9 ; Cooling OFF
N0110 M2 ; End of program

```

When machining the arc shown in fig. 4-48b, the programming using G22, G80 cycle is as follows:

```

N0000 G00 X36 Z-5 ; Rapid positioning to the arc start point
N0010 G22 L3 ; Program cycle starts and cycles 3 times
N0020 G01 U-2 F50 ; X feeds 1mm
N0030 G02 W-14.28 R10 ; X, Z feed to cut the arc
N0040 G01 W14.28 F500 ; Z retracts to the arc start point
N0050 G80 ; End of cycle body
N0060 M30

```

The above two examples are programmed according to the actual workpiece shape. The programming method in Fig. 4-48a is applicable to the roughing like smithing and casting, which can improve the machining efficiency, while the one in Fig. 4-48b is applicable to the bars.

#### 4.16 G98 — Feed per Minute, G99 — Feed per Revolution

##### 【Command format】

**G98** F \* \* \* \* . \* \* \* ; Feed per minute

**G99** F \* \* \* \* . \* \* \* ; Feed per revolution

##### 【Word】

G98 F \* \* \* \* . \* \* \* ; F unit: mm/min;

F - Sets the feedrate for the interpolation command following it in mm/min.

G99 F \* \* \* \* . \* \* \* ; F unit: mm/r;

F - Sets the feedrate for the interpolation command following it in mm/r.

**【Word range】**

F: 0~4000.000 mm/min  
0~4000.000 mm/r

**【Note】**

- 1) The value format of G98/G99 must be matched with that of F word. Otherwise, the system issues: Missing information.
- 2) G98/G99 is the modal command and it keeps valid before being changed. The single F command can be taken as the new feedrate.
- 3) G98 is the initial state of the system and the system automatically sets to G98 during initialization (mm/min).
- 4) When F=0, the system suspends and prompts an alarm message when executing the machining the program.
- 5) In G99, when the spindle speed is 0, the feed automatically pauses and the system displays "Pause: the spindle speed is 0". It keeps the state till the spindle speed is not zero, then, by pressing CYCLE START key again, the system recovers the execution.
- 6) In G99, generally, when the spindle is just started, the system should delay a proper period of time to wait the spindle speed to become stable, and then executes the cutting. Otherwise, the system fails to detect the spindle speed, displaying "Pause: the spindle speed is 0".

**【Example】**

```

...
N0100  G98 F800      ; Sets the feed per minute, F feedrate: 800 mm/min;
...
N0160  F50          ; F feedrate: 50 mm/min;
...
N0200  G99 F2.1     ; Set the feed per revolution, F feedrate: 2.1 mm/r;
...
N0250  F0.56        ; F feedrate 0.56mm/r。
...

```

## 4.17 G31 — Skip Function

The execution of G31 is almost the same as that of G01, and the difference is that G31 constantly detects the state of the external input interface signal (G31I). If the signal meets the condition, the coordinate axis stops the feed and then continues executing the next command; if the signal cannot meet the condition, the coordinate axis feeds to the end point, then the system goes to the next command.

**【Command format】**

**G31 Z (W)\_ X (U)\_ F\_ H\_** ; Z/X two-axis simultaneously feed  
**G31 X (U)\_ F\_ H\_** ; X single-axis feed

**【Word】**

X, Z, Y: End point coordinates after the command ends.

U, W, V: Incremental value counted by the theory end point.

F: Feedrate. F is modal and can be omitted when it is not changed; it is controlled by the feedrate override.

H: Command bit parameter(H\_d7 ~ H\_d1: reserved) ; the system defaults it to be 0 when it defaults.

H\_d0=0: G31I LOW meets the skip condition;

H\_d0=1: G31I HIGH meets the skip condition;

**【Word range】**

X, Z, U, W: (-9999.999~9999.999) mm

F: 0.001mm/min~4000mm/min

H: 00000000~11111111

**【Explanation】**

Z/X can select single-axis or two-axis simultaneous feed.

G31 traverse speed is specified by F and controlled by the feedrate override.

**Actual feedrate = F x feedrate override**

G31 is a non-modal command, which cannot be omitted in the next block if used.

**【Note】**

- 1) G31 input check function is non-standard pin input control; when the function is needed, the user needs to define its pin in the interface parameter P532 (G31I) and connects the lines correctly. For the details about the interface parameter definitions, see Section 4.6 Parameter Operation Mode in Part I Programming; for the wiring connection, see Chapter 3 CNC Device Connection in Part III Connection.
- 2) To ensure the position is correct, the execution decelerates to stop once the system detects the external input interface signal change. Therefore, the actual feedrate in G31 cannot be too high and it is suggested that it should be less than 1000mm/min. Otherwise, G31 cannot realize the skip function because it cannot check the change of the interface signal .

**【Relative parameters】**

Interface parameter P532 defines the common signal names of the input interfaces.

Refer to G01 for the relative parameters of cutting feed.

**【Example】**

Current: X=100 Z=100;

G31 Z30 F500;

The move distance is Z70, Z moves at the speed of 500 mm/min till it reaches Z30 if G31I signal does not meet the skip condition; if Z=68.37, G31I meets the skip condition and Z immediately stops running and the system executes the next command.

4.18 G66 -Memorizing Current Coordinates, G67-Return to Memorized Coordinates

**【Command format】**

**G66** ; Memorizes the current coordinate of tool nose on Z, X, Y axes;  
**G67 W0 U0** ; Returns to the G66 position. The axis uncommanded in G66 does not move;

**【Explanation】**

X, Z: If they are specified using absolute coordinates, it means the axes rapidly traverse to those absolute coordinate points.

U, W: If the relative coordinates which are not 0 are used, it means the axes rapidly traverse to the new coordinates of G66 position plus the increment.

Z/X/Y can select single-axis, two-axis rapid traverse. Absolute coordinates and relative coordinates cannot be used at the same time. The axis that does not move can be omitted, but G67 must be specified with coordinate words.

G66 is to store the current tool nose coordinates; G67 is to command the axis to move to the coordinates rapidly.

G67 should be used only when G66 has been used once. Otherwise, the system issues an alarm. In the program, G66 can be used several times, and G67 is to command to specified axis to rapidly traverse to the latest coordinate position stored by G66.

G66/G67 cannot be in the same block with other commands. It must be specified in an independent block. The movement path of G67 is the same as that of G00.

**【Example】**

N0010 G00 Z350 X280 ; Defines the machining original point and position to this point;  
 N0010 T11  
 N0100 G00 Z102 X120 ;  
 N0200 G66 ; Stores the current coordinates Z102, X120  
 N0200 T22 ; The tool nose coordinates are uncertain after tool changing;  
 N0300 G67 X120 W10 ; X axis rapidly moves to 120, and Z axis rapidly moves to 102+10.  
 N0200 M02

4.19 Appendix: G function and Its Explanation Table (Table 4-3)

Code	Programming format and explanation	Function	Group	Remarks
<b>G00</b>	G00 X (U) Z (W) (single or two-axis)	Rapid traverse	1	Initial G
<b>G01</b>	G01 X (U) Z (W) F (single or two-axis)	Linear interpolation		Modal G
<b>G02</b>	G02 X (U) Z (W) R F	CCW circular interpolation		
	G02 X (U) Z (W) I K F			
<b>G03</b>	G03 X (U) Z (W) R F	CW circular interpolation		
	G03 X (U) Z (W) I K F			

<b>G05</b>	G05 X (U) Z (W) I K F (I, K) an intermediate on the arc	Circular interpolation			
<b>G32</b>	G32 Z (W) P (E) H	Taping cycle	2	One-shot G	
<b>G33</b>	G33 X (U) Z (W) P (E) I K Q H	Constant thread cutting			
<b>G26</b>	G26 X (U) Z (W) (single or two-axis) Return to the program reference point via the intermediate point; rapid traverse by G00.	Program reference point return		4	One-shot G
<b>G28</b>	G28 X (U) or G28 Z (W)	Return to machine zero			
<b>G30</b>	G30 P2 X (U) Z (W) (single or two-axis) G30 P3 X (U) Z (W)	Return to 2 <sup>nd</sup> , 3 <sup>rd</sup> program reference point			
<b>G50</b>	G50 X Z (single or two-axis)	Workpiece coordinate system setting			
<b>G51</b>	G51	Workpiece coordinate system recovery			
<b>G31</b>	G31 X (U) Z (W) F H (single or two-axis)	Skip function			
<b>G66, G67</b>	G66	Memorizing current t coordinate point			
	G67 W0 U0 V0	Return to the coordinate point memorized by G66			
<b>G74</b>	G74 X (U) Z (W) I K R E F	End face drilling cycle	3	One-shot G	
<b>G75</b>	G75 X (U) Z (W) I K E F	Outer/inner grooving cycle			
<b>G90</b>	G90 X (U) Z (W) R F	Outer/inner cylinder face cycle			
<b>G92</b>	G92 X (U) Z (W) P (E) I K R L H	Thread cutting cycle			
<b>G94</b>	G94 X (U) Z (W) R F	Outer/inner taper cycle			
<b>G22</b>	G22 L	Local cycle start	6	One-shot G	
<b>G80</b>	G80	Local cycle end		One-shot G	
<b>G04</b>	G04 D	Delay	5	One-shot G	
<b>G71</b>	G71 U W G71 X (U) I K F P Q G710 X (U) P Q F	Outer roughing, finishing cycle command group			
	<b>G72</b>	G72 U W G72 Z (W) I K F P Q G720 Z (W) P Q F			End face roughing, finishing cycle command group
		<b>G73</b>			G73 U W G73 X (U) Z (W) I K F P Q L G730 X (U) Z (W) P Q F
<b>G76</b>	G76 D I Q P L R G76 X (U) Z (W) P (E) I K R L H	Multi threading cycle command group			
<b>G96</b>	G96 S	Constant surface speed ON	7	Modal G	
<b>G97</b>	G97 S	Constant surface speed OFF		Initial G	
<b>G98</b>	G98 F	Feed per minute	8	Initial G	
<b>G99</b>	G99 F	Feed per revolution		Modal G	

**Note:** The system is in the G00, G97, G98 states upon power-on and reset.

4.20 Appendix: G and Its Relative Parameter Explanation (Table 4-4)

Code	Function	Relative parameter explanation	Remarks
G00	Rapid traverse	P100~P108, P112, P114, P400_d3	
G01	Linear interpolation	P112, P113, P114, P401_d5, P401_d4	
G02	CCW circular interpolation	P112, P113, P114, P401_d4, P400_d2	
G03	CW circular interpolation		
G05	Circular interpolation		
G32	Taping cycle	P100, P103, P106, P112, P113, P114	
G33	Constant thread cutting	P100, P101, P103, P104, P106, P107, P113, P116, P117, P209, P306, P307, P403_d0	
G26	program reference point return	Refer to G00 and G50	
G28	Return to machine zero	P021~P026, P109, P110, bit parameter P406 and P407. Refer to G00 for others	
G30	2 <sup>nd</sup> , 3 <sup>rd</sup> program reference point return	P003~P008. Refer to G00 for others	
G50	Workpiece coordinate system setting	P000, P001, P002	
G51	Workpiece coordinate system recovery		
G74	End face drilling cycle	Refer to G00 and G01	
G75	Outer/inner grooving cycle	The same as those of G74	
G76	Multiple thread cutting cycle command group	P336, P337, P338, P339. Refer for G00 and G33 for others	
G90	Outer/inner cylinder face cycle	Refer to G00 and G01	
G92	Threading cycle	Refer to G00 and G33	
G94	Outer/inner taper cycle	Refer to G00 and G01	
G22	Local cycle start		
G80	Local cycle end		
G04	Delay		
G71	Outer roughing, finishing cycle command group	Refer to G00 for the relative parameters of rapid traverse, and refer to linear and circular interpolation for those of cutting feed	
G72	End roughing , finishing cycle	The same as those G71	
G73	Closed-loop cutting cycle command group	Refer to G71, G72	
G96	Constant surface speed ON	P410_d6, P304, P305	
G97	Constant surface speed OFF		
G98	Feed per minute		
G99	Feed per revolution		
G31	Skip function	P532. Refer to G01 for others	
G66	Memorizing the current coordinates		
G67	Return to coordinates memorized by G66		

## CHAPTER FIVE GENERAL PROGRAMMING RULES AND EXAMPLES

## 5.1 General Programming Rules

The command compiling must meet the rules in one block, so that the system can identify them conveniently; it is suggested that the command character should be placed in front of the data words. When the system executes the **Programming Rule Check**, only the program meeting the programming rules can be compiled; if an error is detected, an alarm is issued. Please observe the following rules:

- 1) No repetitive command is allowed in one block; otherwise, the system issues "E202: Repetitive command";  
Incorrect example: N0200 G00 G00 Z30;
- 2) The command character must be followed by the valid digital command; otherwise, the system issues "E201: Illegal command";  
Incorrect example: N0200 G23 Z30 ; No such a command.
- 3) There must be no repetitive data word in one block, otherwise, the system issues "E234: the data word is repetitive" ;  
Incorrect example: N0200 G00 Z20 Z30;
- 4) There must be no conflicting data words; otherwise the system issues "E210; the excessive words";  
Incorrect example: N0200 G00 Z20 W30;
- 5) The command character and word character must be followed by valid digits without a blank space; otherwise the system issues "E204: command format error";  
Incorrect example: N0200 G00 Z 20 ; No blank space between Z and 20  
N0200 G 00 Z30 ; No blank space between G and 00
- 6) The required data in the block cannot be omitted, otherwise, the system issues "E206: missing message";  
Incorrect example: N0200 G90 X100 ; The coordinates of the two axes must be given. Z coordinate data is missing here.
- 7) There must be no words, letters and digits unrelated to the command; otherwise, the system issues "E203: Illegal message";  
Incorrect example: N0200 G00 X W 100 ; Unnecessary character W
- 8) The values of the data word must be within the valid range; otherwise, the system issues "E211: Data exceeding the range"  
Incorrect example: N0200 G00 X100 Z99999 ; the data 9999 exceeds the range.
- 9) The first digit can be omitted when it is zero in the command.  
Example: G00 can be abbreviated to G0.



## 5.2 Programming Rules for Commands in One Block

Programming for multiple commands in one block means that there are many commands in one block simultaneously, but all these commands cannot be in the same block. When the system executes **Programming rule check**, only the programs meeting the programming rules can pass the compiling; otherwise, the system issues an alarm and prompts **Some commands need to be used independently**. And the rules are as follows:

1) Commands only specified in an independent block:

G50, G51, G26, G28, G30, G31;

G32, G33;

G04;

Program local cycle: G22, G80;

Single canned cycle: G92, G74, G75;

Single canned cycle: G90, G94;

Compound cycle: G71, G72, G73, G76;

M00, M02, M20, M30, M96, M97, M98, M99;

M21, M22, M23, M24;

M91, M92, M93, M94;

M81, M82, M83; M60~M74 (Self-defined commands can be in the same block with F, S command).

2) The system cannot make a judgment for some commands below have mutually-contradictory operations or the same data. To avoid the above, M, G commands which cannot be in the same block are divided into many groups and only the commands in the different group can be in the same block as follows:

M commands which can be in the same block are divided into many groups as follows:

Group 1: M03, M04, M05;

Group 2: M08, M09;

Group 3: M10, M11, M12;

Group 4: M78, M79, M80;

Group 5: M32, M33;

Group 6: M41, M42, M43, M44.

M commands which can be in the same block are divided into many groups as follows:

Group 1: G00, G01, G02, G03, G05; (Thereinto, G02, G03, G05 cannot be in the same block with T command)

Group 2: G96, G97;

Group 3: G98, G99.

**【Note】**

- 1) The commands which must be specified in an independent block cannot be in the same block with other commands; otherwise, the system issues “E205: Incompatible commands”, and they do not meet the rules for the commands in the same block;
- 2) The commands in the same group cannot be in the same block; otherwise, the system issues “E205: Incompatible commands”;
- 3) T command cannot be in the same block with **G02, G03, G05**; otherwise, the system issues “E205: Incompatible commands”; and they do not meet the rules for the commands in the same block;

5.3 Command Execution Sequence

**【Relative parameters】**

**P401 bit parameter** efficiency setting (3-level)

**P401\_d3:** Execution sequence of many commands in the same block: 0-substep 1-synchronism

**P401\_d2:** Rapid skip execution function: 0-ON 1-OFF

◆ Substep execution for many commands in the same block

**P401\_d3=0**, substep; at the moment, **P401\_d2** is not active.

When many commands in the same block are executed, their execution sequence is as follows: after one functional command is executed, the system executes the next one till all are performed.

Table 7-1 V Command execution sequence table

Command execution sequence(from top to down)	Function
M32	Cooling ON
M10	Chuck clamping
M78	Tailstock advancing
M41, M42, M43, M44	Spindle gear shifting 1 <sup>st</sup> , 2 <sup>nd</sup> , 3 <sup>rd</sup> , 4 <sup>th</sup> gear
S function	Spindle gear shifting or rotating speed
M03, M04	Spindle ON
M08	Cooling ON
G98/G99 F function	
G96/G97 S function	
T function	Tool change
G40/G41/G42	
G00/G01/G02/G03/G05	Motion command
M05	Spindle OFF
M09	Cooling OFF
M33	Lubricating OFF
M79	Tailstock retracting
M11	Workpiece releasing
M12	
M80	

**【Example】**

N1000 T11 M03 M10 G00 X50 M08

Substep execution sequence: M10, M03, M08, T11, G00 ;

Total used time for executing the whole block is equal to the sum of each single command execution time.

**◆ Synchronous execution for many commands in the same block**

**P401\_d3=1**, synchronism; **P401\_d2=0**: forbid the execution of rapid block jump function

In the synchronous execution in table 7-1, M, S, T before the motion command are executed at the same time with the motion command, and M command after the motion command cannot be executed till the motion command is executed. The system executes the next block after all are executed.

The commands which are not relative to each other are almost executed simultaneously. They are not executed orderly. The system automatically arranges the execution sequence for those M codes with interlock relationship according to the parameters.

**【Example】**

N1000 T11 M03 M10 G00 X50 M08

When the parameter is set to starting the spindle after clamping (**P402\_d5=0**), the system follows the interlock relationship and its execution process is as follows:

- 1) Simultaneously start executing T11/ G00 X50/M10/M08;
- 2) Execute M03 after M10 is executed;
- 3) Execute the next block after all are executed.

When the parameter is set to starting the spindle without firstly being clamped, the execution process is as follows:

- 1) Simultaneously start executing T11/ G00/M10/M08/ M03;
- 2) Execute the next block after all are executed.

The total used time for executing the whole block is equal to the longest single execution time.

**◆ Synchronous execution for many commands in the same block and rapid block jump execution allowed**

**P401\_d3=1**, synchronism; **P401\_d2=1**, rapid block jump execution is allowed.

In the mode, the system rapidly skips to the next block after the axis motion commands of the current block are executed (do not wait for the end of execution of other M, S, T).

**【Note】**

- 1) The system follows the rule: wait the others to be executed when other commands in the same group are executed.
- 2) When the system meets "The command only specified in an independent block" , it waits to execute it after the all the previous commands are executed; and it executes the next block after "The command only specified in an independent block" is executed.

- 3) When the user needs to execute the next after all the previous commands are executed, he needs to insert one G04 D0 command.
- 4) When rapid block jump execution is allowed, note that the distance between tool change point and workpiece must be sufficient, to prevent the tool colliding with the workpiece when the tool is being changed and approaching the workpiece.

### 5.4 Programming Example

All the following programming examples use the tool as Fig. 5-1:

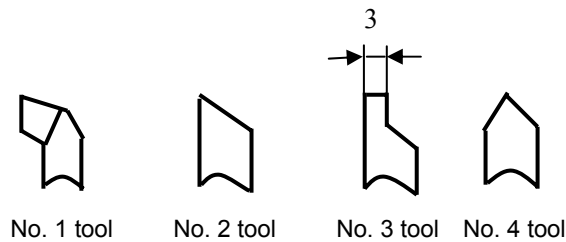


Fig. 5-1 Programming example tool shape

#### 5.4.1 Outer Machining Example

**Example 1:** Workpiece in Fig. 5-2. Bar:  $\Phi 64 \times 105$  mm. No. 1 tool is used for roughing, No. 2 for finishing. (G90 inner/outer cylindrical surface turning cycle).

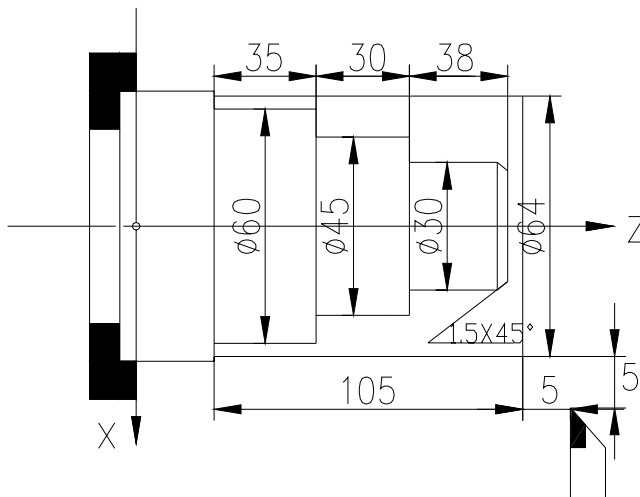


Fig. 5-2 Outer machining example

```

N0000 G50 X100 Z150 ; Sets a the workpiece coordinate system
N0010 M3 S01 ; Starts the spindle and set to the spindle speed gear 1
N0020 M8 ; Cooling ON
    
```

```

N0030 T11 ; Executes NO.1 tool change and execute its
           ; compensation
N0040 G00 Z103 X65 ; Tool rapidly approaches the workpiece
N0050 G01 X0 F60 ; Cuts the end face at the speed of 60mm/min
N0060 G00 Z105 ; Tool leaves the end face of workpiece
N0070 X60.5 ; Positions the tool with the allowance 0.5mm
N0080 G01 Z0 F60 ; Turns outer Φ60.5
N0090 G00 X62 ; Tool leaves the surface of workpiece
N0100 Z105 ; Tool rapidly traverses to the workpiece
N0110 X60.5 ;
N0120 G90 X56.5 Z35 F100 ; Turns outer Φ45mm with cylindrical surface
cycle command
N0130 X52.5 ; Tool infeed 4mm and cycles again
N0140 X48.5 ; Tool infeed 4mm and cycles again
N0150 X45.5 ; Tool infeed 3mm and cycles again
N0155 G00 X45.5 ; X rapidly traverses the starting point of next
cycle
N0160 G90 X40.5 Z65 ; Turns Φ30mm outer with cylindrical surface
cycle command
N0170 X35.5 ; Tool infeed 5mm and cycles again
N0180 X30.5 ; Tool infeed 5mm and cycles again
N0190 G00 X100 Z150 ; Retracts tool to a safety position
N0200 T22 ; Changes the No.2 tool
N0210 S2 ; Sets to the spindle high speed
N0220 G00 Z103 ; Tool positions to the vicinity of the workpiece
N0230 X32 ;
N0240 G01 X27 ; Tool infeed to starting point of chamfer
N0250 X30 Z101.5 F60 ; Finishes chamfer 1.5mm
N0260 Z65 ; Finishes outer 30mm
N0270 X45 ;
N0280 Z35 ; Finishes outer 45mm
N0290 X60 ;
N0300 Z0 ; Finishes outer 60mm N0310 G00 X100
Z150 ; Tool returns the origin point
N0320 M5 ; Spindle OFF
N0330 M9 ; Cooling off
N0340 M2 ; End of program

```

#### 5.4.2 Thread Machining Example

**Example 2:** as Fig. 5-3. Bar: Φ30×100 mm, No. 1 is roughing tool, No. 3 is parting tool, No. 4 for 60° threading tool.

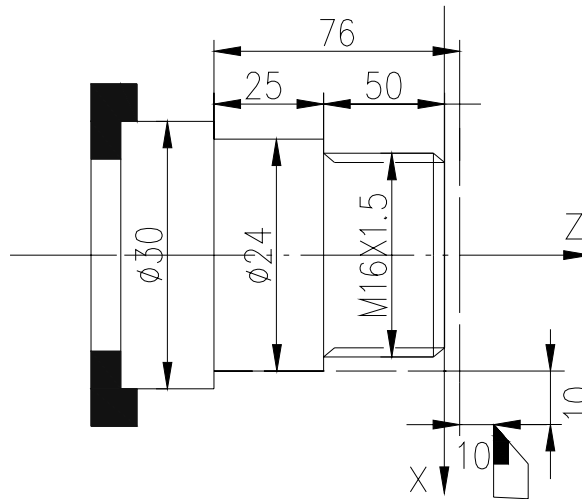


Fig. 5-3 Thread machining example

N0000	G50	X50	Z11						; Sets a workpiece coordinate system
N0010	M3	S2							; Starts the spindle, sets the spindle speed to No.2 gear
N0020	M8								; Cooling ON
N0030	T11								; Executes No. 1 tool change and executes its tool compensation
N0040	G00	Z0	X35						; Tool approaches the workpiece
N0050	G01	X0	F60						; Cuts the end face at the speed of 60mm/min
N0060	G00	W2							; Tool leaves the end face of workpiece
N0070	X24								; Tool positions to the vicinity of the workpiece
N0080	G01	Z-78	F60						; Turns outer $\Phi 24$ and reserves the cut width
N0090	G00	X26							; Tool leaves the workpiece surface
N0100	Z0								; Tool positions to the vicinity of the workpiece
N0110	X24								;
N0120	G90	X21	Z-50	F100					; Turns outer $\Phi 16$ mm with cylindrical surface cycle command
N0130	X18								; Tool infeeds 3mm and cycles again
N0140	X15.9								; Tool infeeds 2.1mm and cycles again, outer is less than 0.1mm
N0150	G01	X14	Z0						; Tool positions to the starting point of chamfer
N0160	X16	W-1							; Cuts a chamfer
N0170	G00	X50	Z50						; Retracts the tool to a safety position
N0180	T44								; Change No.4 thread tool
N0190	S2								; Sets the spindle to high speed (the speed is not more than 800n/min)
N0200	G00	X19	Z2						; Positions to the thread starting point of the first tool
N0220	G92	X15	Z-50	P1.5	K2				; Machines the thread, length of run out 2mm
N0230	X14.2								; Tool infeed 0.8mm to the 2 <sup>nd</sup> thread machining
N0240	X13.8								; Tool infeed 0.4mm to the 3 <sup>rd</sup> thread machining
N0250	X13.6								; Tool infeed 0.2mm to the 4 <sup>th</sup> thread machining
N0260	G00	X60	Z50						; Retracts the tool to a safety position
N0270	T33								; Change No. 3 tool with tool width of 3mm
N0280	G00	X26	Z-78						; Positions to the cut-off point

```

N0290 G01 X0 F50 ; Cut off
N0300 G26 ; X, Z return to the starting point of machining
N0310 M9 ; Cooling OFF
N0320 M5 ; Spindle OFF
N0330 M2 ; End of program
    
```

**Example 3:** Cut the multiple threads with M98 subprogram call and M99 subprogram return commands.

Fig. 5-4a: metric multiple thread. Fig.5-4b: inch multiple thread with three threads. The program is as follows:

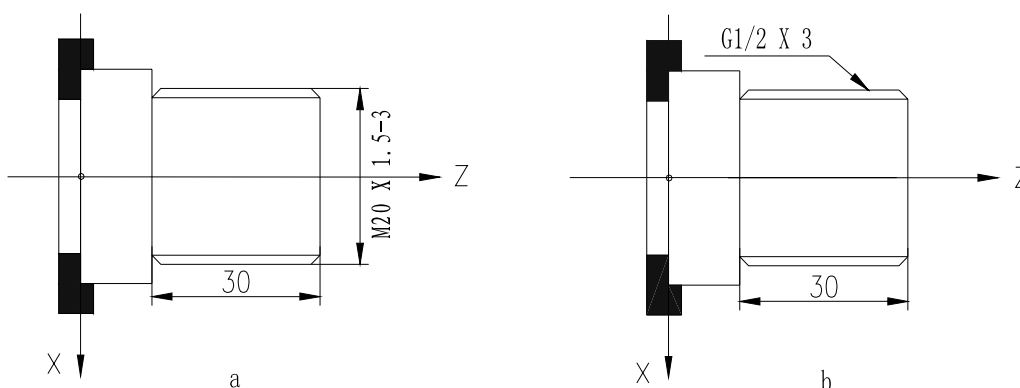


Fig. 5-4 Multiple thread cutting example

a (Metric multiple threads) :

```

N0010 G50 X100 Z50 ; Sets a workpiece coordinate system
N0020 M03 S600 ; Spindle rotates (CW) with 600 r/min
N0030 T44 ; Changes No. 4 tool and executes its offset
N0040 G00 X25 Z5 ; Rapidly approaches the workpiece
N0050 G92 X19.5 Z-30 P4.5 L3 ; Executes No. 1 thread cycle, P= thread lead
N0060 X19 ; Tool infeed 0.5mm, executes the 2nd thread machining
N0070 X18.5 ; Tool feed 0.5mm, executes the 3rd thread machining
N0080 X18.35 ; Tool feed 0.15mm and executes the 4th thread machining
N0090 G00 X100 Z50 ; Rapidly returns to program origin
N0100 M05 ; Stops the spindle
N0110 M02 ; End of program
    
```

b (Inch multiple threads) :

```

N0010 G50 X100 Z50 ; Sets a workpiece coordinate system
N0020 M03 S600 ; Spindle rotates(CW) with 600n/min
N0030 T44 ; Changes No. 4 tool and executes its offset
N0040 G00 X25 Z5 ; Rapid approaches the workpiece
N0050 G92 X19.5 Z-30 E4.667 L3 ; Executes the 1st thread cycle
N0060 X19 ; Tool infeed 0.5mm, executes the 2nd thread machining
N0070 X18.5 ; Tool infeed 0.5mm, executes the 3rd thread machining
N0080 X18.35 ; Tool infeed 0.15mm, executes the 4th thread machining
N0090 G00 X100 Z50 ; Rapidly returns to program reference point
N0100 M05 ; Stops the spindle
N0110 M02 ; End of program
    
```

5.4.3 Compound Machining Example

**Example 4:** Compound machining as Fig. 5-5(G71, G92).

Bar  $\Phi 135 \times 178$  mm. 4 tools are used for machining:

No. 1 is outer roughing tool; No.2 is outer finishing tool; No.3 is for grooving with 3mm width; No.4 is  $60^\circ$  threading tool. The machining allowance in finish machining is defined by the offset parameter T8.

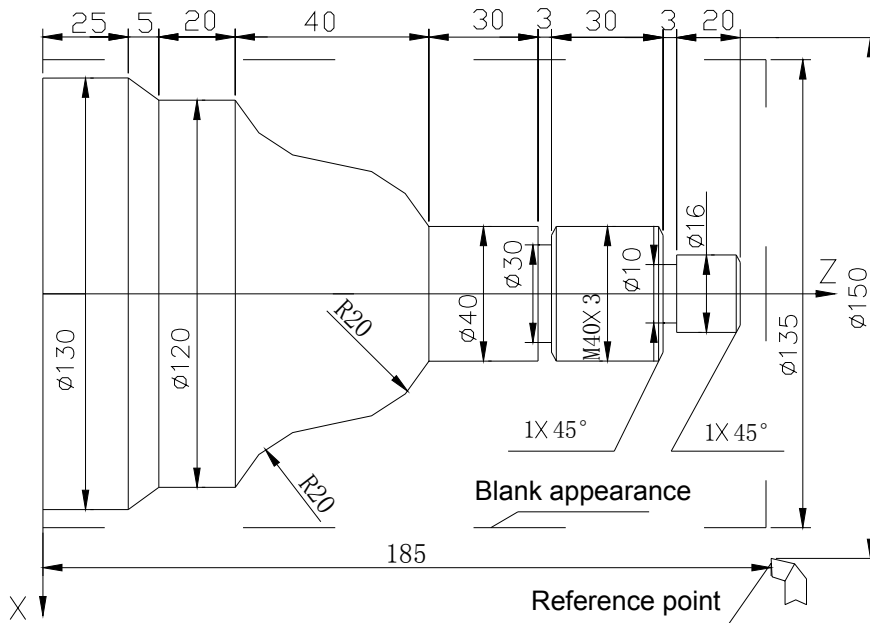


Fig. 5-5 Machining example

```

N0000 G50 X150 Z250 ; Sets a workpiece coordinate system
N0010 M3 S01 ; Starts the spindle and sets its speed to gear 1
N0020 M8 ; Cooling ON
N0030 T08 ; Executes the tool compensation with machining allowance
N0040 G00 X136 Z180 ; Approaches the workpiece
N0050 G71 X0 I4 K2.5 L10 F80 ; Outer compound cycle
N0060 G01 W-4 ; Approaches the face of workpiece
N0070 X16 ; Turns the end face
N0080 W-23 ; Turns outer  $\Phi 16$ 
N0090 X40 ; Turns the end face
N0100 W-63 ; Turns outer  $\Phi 40$ 
N0110 G02 X80 W-20 R20 ; Turns convex arc
N0120 G03 X120 W-20 R20 ; Turns concave arc
N0130 G01 W-20 ; Turns outer  $\Phi 120$ 
N0140 G01 X130 W-5 ; Turns taper
N0150 G01 W-25 ; Turns outer  $\Phi 130$ 
N0155 G00 X150 ;
N0160 G26 X136 Z180 ; Returns to the starting point after roughing
N0180 T22 ; Changes No. 2 tool to finish the outer
N0190 S02 ; Sets to the spindle to high speed
    
```



N0200	G00	X0	Z178						; Rapidly approach the workpiece
N0210	G01	Z176	F50						; Approaches the end face of workpiece
N0220	G01	X14							; Turns the end face to the chamfer
N0230	X16	W-1							; Chamfer 1×45°
N0240	W-22								; Finishes outer Φ16
N0250	X37								; Finishes the end face Φ40
N0260	X40	W-1.5							; Chamfer 1×45°
N0270	W-61.5								; Finishes outer Φ40
N0280	G02	X80	W-20	I0	K-20				; Finishes convex circle
N0290	G03	X120	W-20	I40	K0				; Finishes concave circle
N0300	G01	W-20							; Turns outer Φ120
N0310	X130	W-5							; Turns taper
N0320	W-25								; Finishes outer Φ130
N0325	G00	X150							; X rapidly retracts the tool
N0330	G26	X136	Z180						; Returns to the starting point of machining
N0340	T33								; Changes No. 3 tool to groove
N0350	G00	X42	Z120						; Approaches the workpiece
N0360	G01	X30	F50						; Groove Φ30
N0370	G01	X40							; Retracts the tool
N0380	G01	Z121.5							; Positions to the starting point of chamfer
N0390	X37	Z121.5							; Chamfer 1×45°
N0400	X41								; Retracts the tool
N0410	G00	Z153							; Rapidly traverses
N0414	G01	X20	F200						; Approaches the workpiece
N0418	X10	F50							; Groove Φ10
N0420	G00	X100							; X rapidly retracts the tool
N0425	G26	X136	Z180						; Returns to the starting point of machining
N0430	T44	S01							; Changes No. 4 tool to cut the thread and set to the spindle low speed
N0440	G00	X42	Z155						; Approaches the workpiece
N0460	G92	X39	W-34	P3					; Executes the thread machining cycle
N0470	X38.2								; Feeds 0.8 and executes the 2 <sup>nd</sup> tool machining
N0480	X37.7								; Feeds 0.5 and executes the 3 <sup>rd</sup> tool machining
N0485	G00	X100							; X rapidly retracts the tool
N0490	G26	X136	Z180						; Returns to the starting point of machining
N0500	T11								; Changes No. 1 tool
N0510	M05								; Spindle OFF
N0520	M09								; Cooling OFF
N0530	M02								; End of program

**Example 5:** Fig. 5-6. Bar: Φ50×100. Machining with 3 tools: No.1: outer turning tool; No.2: pointed tool; No.3: grooving tool with 3mm width.

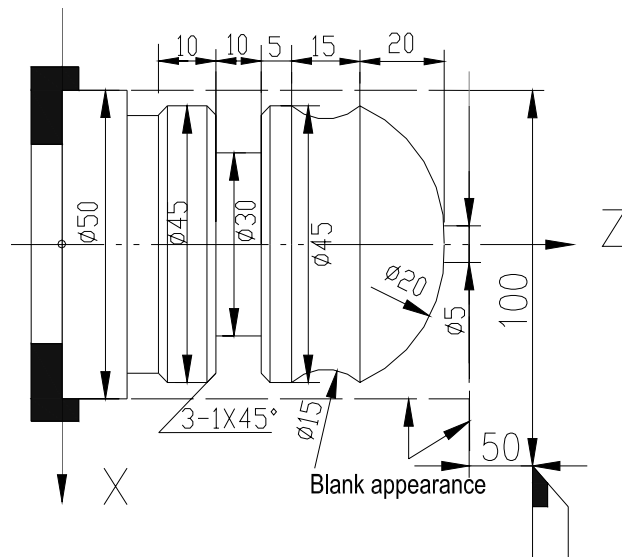


Fig. 5-6 Machining example

```

N0010 G50 X100 Z50           ; Sets a workpiece coordinate system
N0020 M4 S600                ; Spindle rotates(CW) with 600 rev/min
N0030 T11                    ; Changes No.1 tool and executes its offset
N0040 M 8                    ; Cooling ON
N0050 G00 X50 Z3            ; Approaches the workpiece
N0060 G71 X0 I3 K2 P0070 Q0100 F50 ; Executes the outer compound cycle
N0070 G01 Z0                 ; Approaches the workpiece
N0080 X5                     ; Turns the end face
N0090 G03 X45 Z-20 R20       ; Turns the arc R20
N0100 G01 Z-65               ; Turns the outer Φ45
N0110 G00 X50                ; Rapidly traverses to X50 safety position
N0111 G26 X100 Z50           ; Rapidly returns to reference point (program starting
                               point X100 Z50)
N0120 T22                    ; Changes No. 2 and execute its offset
N0130 G00 X51 Z-20           ; Rapidly traverse
N0140 M98 P0160 L3           ; Calls subprogram
N0150 M97 P0200              ; Program skips to N0200
N0160 G01 U-2 F40            ; X feeds 2, approaches to workpiece
N0170 G02 U0 W-15 R7.5 F35   ; Turns R7.5 arc
N0180 G01 W15 F500           ; Return to arc starting point
N0190 M99                    ; End of subprogram
N0200 G26 X100                ; X rapidly returns to program reference point (i.e. X100)
N0210 G26 Z50                 ; Z rapidly returns to program reference point (i.e. Z50)
N0220 T33                    ; Uses No. 3 grooving tool and executes its tool offset
N0230 G00 X50 Z-43           ; Rapidly positions to cycle starting point( tool width 3mm)
N0240 G75 X30 Z-50 I5 K2 E3 F50 ; Executes the grooving cycle command
N0250 G01 X45 Z-42 F50       ; Chamfer
N0260 X43 Z-43               ; Chamfer
N0270 X30                    ; Moves to X30 outer to execute finishing
N0280 Z-50                   ; Turns Φ30 outer
    
```

N0290 G00 X45 ; Positions to the chamfer position  
N0300 G01 Z-51 F50  
N0310 X43 Z-50 ; Chamfer  
N0320 G00 X46 ; Position the cut-down point  
N0330 Z-62  
N0340 G01 X42 Z-63 F50 ; Chamfer  
N0350 X0  
N0360 G26 X100 ; Return to X reference point  
N0370 G26 Z50 ; Return to Z reference point  
N0380 T10 ; Uses the reference tool T10  
N0390 M5 ; Spindle OFF  
N0400 M9 ; Cooling OFF  
N0400 M2 ; End of program

## CHAPTER SIX ALARM MESSAGE

All the alarm messages in this CNC system are prompted in the screen by Chinese characters. Please performs corresponding handling according to the prompting content. Each alarm meaning is shown in the following table. “Workpiece program” is also referred to as “machining program” in the following table.

The alarm number starts with capital letter E followed by a three-digit number. The classification is as follows:

E001~E009: Alarms in PARAMETER, OFFSET operation mode;

E100~ E199: Alarms in EDIT operation mode;

E200~ E299: Alarms in PROGRAM COMMAND;

E600~ E699: Alarms in PROGRAM CHECK;

E300~ E399: Alarms in JOG, AUTO operation mode when executing relative operation;

E400~ E499: Related Alarms in JOG, AUTO operation mode when executing statements.

## 6.1 Emergency Stop Alarm

For the details about the stop operation in system emergency, please refer to Section 4.1.4.3 Emergency Stop Alarm in Part I Operation.

## 6.2 Alarm Table in PARAMETER, OFFSET Operation Mode (i.e. E001~E009)

Alarm No.	Alarm prompt	Alarm cause	Troubleshooting
E001	Not support USB	USB device isn't supported by this system software	Use USB device supported by the system
E002	USB Removed suddenly	Device is removed suddenly in making USB device check	Incorrect operation. Forbid this operation
E003	No USB inserted	Don't USB device is inserted	Insert USB device
E004	Fail to find valid catalogue or file	USB device doesn't create the specified catalogue and file	Create the valid catalogue and file in USB device
E005	USB file saving failure	USB file saving failure	Save again or check USB device
E006	Not enough memory in USB	Not enough U disc rest room	Use a bigger U disc or check U disc to release more space
E007	Create content fail	U disc fails in creating the specified catalogue	Check USB device and create the appointed catalogue again
E008	Create file fail	Failure in creating file	Check USB device or create the specified format file again
E009	USB read file fail	USB is failed in reading file	Check the USB device or read file again
E010	Fail to find dedicated content or file	USB is failed in reading catalogue list	Check the USB device or create this catalogue again
E011	File is too big	Too large file	File is too large, modify the file
E012	Fail to open file	Can't open the file in specified catalogue or file is destroyed	Check the file name ,suffix if they can match the rule or check the USB equipment, create the file again
E013	No dedicated file	The file in specified list does not exist or has been destroyed	Create the file in specified file

Alarm No.	Alarm prompt	Alarm cause	Troubleshooting
E014	Invalid Para NO	Parameter number exceeds the allowed range	Change the parameter number into the available parameter number range
E015	Para NO not standard	Invalid character in parameter number, does not meet the standard	Delete the invalid character in the parameter number
E016	Specify Para NO before data	Must have parameter number in the first parameter	Add the parameter number of the first parameter
E017	Error in Para data	Parameter data exceeds the system specified range	Modify the parameter data
E018	Repeatedly input Para NO	Parameter number can't be input repeatedly	Check and delete the repeat parameter number
E019	Invalid characters in the data	Can't have illegal character in file	Delete the illegal character
E020	Wrong Para file formula	Parameter file format can't meet the requirement	Create the parameter file according to the parameter file format or add the parameter file sign
E021	S-upgrade code proof error	Software promoting command checking error	Contact the supplier
E022	Stop communication	Delete manual, stop transmission, RS232 communication is failed in sending and receiving	Check the command control unit serial software tool or hardware connection
E023	Illegal I/O port set pin NO out of range	Interface parameter setting exceeds the maximal I/O pin number	Modify the I/O to the specified range
E024	Illegal I/O port set input pin NO which used	Parameter setting and I/O input pin collision	Release this I/O or use other I/O pin
E025	Illegal I/O port set output pin NO which used (33)	Parameter setting and I/O input pin collision	Release this I/O or use other I/O pin
E027 ~ E029	Z/X/Y thread invalid	Z/X/Y axis pitch setting value is wrong, doesn't meet the standard	Check the pitch compensation position and modify
E030	Fail to read Para, initialization	Failure in SRAM saving parameter, can't read	Save again
E031	Illegal character OFT.	Illegal character in offset number of tool compensation transmission file	Delete the illegal character
E032	Tool compensation NO not standard	There is no tool compensation number behind the T of tool compensation transmission file, or the offset number lack T sign	Completing the number behind the T sign in file, or add the sign of offset number
E034	OFT. NO exceed 1—64	The tool offset number in tool compensation transmission file exceeds (1~64) range	Modify or delete the exceeded tool compensation number, change them to the specified range
E035	Illegal data	Tool compensation data in tool compensation transmission file has illegal character in	Delete the illegal character
E036	Input data error	The tool compensation input data in tool compensation transmission file has mistake	Check data according the tool compensation communication file format and correct it
E037	File symbol error	File type mark in tool compensation transition file is wrong	Modify the file type according to the tool compensation file format
E038	Send OFT fail	sending tool compensation file failure or be manually cancelled	Sending tool compensation file according to the correct tool compensation file format and operation method

Alarm No.	Alarm prompt	Alarm cause	Troubleshooting
E039	Receive OFT fail	Fail in receiving tool compensation file or delete manual	Receive the tool compensation file according the correct tool compensation file format and operation mode
E040	File is too big	The length of received file is out of the maximal length of buffer zone	Modify the file according the file format, correct the received file according the requirement
E041 ~ E043	Z/X/Y pitch compensation value out of range	Known from system internal arithmetic, in the 0.254mm length of Z/X/Y axis, the calculated departure value is out of 7um	Change the pitch compensation value of pitch compensation parameter to small
E044 ~ E046	Ratio of magnify & coefficient exceed (1/128-128)	Ratio between multiplication and division is out of (1/128~128) range	Modify parameter: multiplication or division
E047	Para set conflict, input port was used	The corresponding pin in function is occupied	Release the occupied input pin in interface parameter, or set other free input pin
E048	Para set conflict, output port was used (33)	The corresponding pin in inputting function is occupied	Release the occupied output pin in interface parameter, or set other free output pin
E049	Abnormal, ESC	In reading the USB process, it is disordered	Operate the USB device over again
E050	Not support USB file formula	The system only reads FAT16 or FAT32 file format	Check or change USB device
E051	USB memory is too big	The system only supports the maximal 8G USB device	Change the USB device
E053	USB fails to transmit, reset	Failure in USB data transmission process	Press system reset key and try again, or operate the USB device again
E054	Function not provided in this system	928TD system does not provide this function.	Only the 928TEa provides this function

## 6.3 Table of Alarm in Edit Operation Mode(i.e. E100~ E199)

Alarm No.	Alarm prompt	Alarm reason	Troubleshooting
E100	Program to be deleted doesn't exist	The program list hasn't the program to be deleted	wrong operation
E101	Block is too long in received program	There are blocks exceeded the length in the program of system sent	Modify the program to be sent
E102	Program which will send doesn't exist	The program catalogue hasn't the program to be sent	Rebuilt the program to be sent
E103	No specified program	Don't choose program in program catalogue	Choose the needed program in program catalogue
E110	RS232-fail to send program	RS232- failure in communication sending	Check the command control unit serial system tool or hardware interface
E111	Received program NO is wrong	The program number range doesn't in (0~254)	Check the receiving program's number
E112	Received program is void	No content in program	Check and modify the program
E113	RS232-fail to receive program	RS232- failure in communication receiving	Check the command control unit serial system tool or hardware interface
E120	Not support USB	The system can't identify this USB	Use the USD equipment which can

Alarm No.	Alarm prompt	Alarm reason	Troubleshooting
		device	be identified by the system
E121	USB is removed	The system is identifying the USB equipment, the equipment is moved suddenly	wrong operation, forbidden these operation
E122	Not insert USB	Don't insert the USB equipment	Insert the USB device
E123	Stop opening USB	Manual operation, press emergency or reset key	Manual operation, stop to open the USB device
E124	Fail to save program in USB	The USB is failed in saving program	Check the USB device
E126	Surplus space in USB is not enough	The U disc doesn't have enough rest room	Change the bigger U disc or tidy U disc to release more room
E127	Contents fail	The U disc creates specified catalogue failure	Check the USB device
E128	Fail to set specified content	The program in specified catalogue can't be opened	Check the USB device or the program is destroyed and must to be created over again
E129	Fail to set program	The U disc is failed in creating program	Check the USB device
E130	Stop sending program	Manual operation, press emergency or reset key	Manual operation, exit the sending
E131	No specified content in USB: C928PRO	The USB device doesn't have the specified catalogue: C928PRO	Create this catalogue
E132	USB fails to transmit & to send now	The USB device communication error	Press reset key and then insert the USB device again
E133	Stop receiving files	Manual operation, press emergency or reset key	Manual operation, exit the receiving
E134	Fail to read program in USB	The USB is failed in reading program	Check the USB device or create this program
E136	Program is too big	The communication program capacity is too large	The program is too large, modify the program
E137	Fail to open program	Can't open program, maybe it's destroyed	Check the USB device or create this program
E142	Program is too long, delete extra part	Program error	Check the program error again
E143	Program is too long, delete it	Program error	Check the program error again
E144	Current program is void, so can't rename	The current program is empty and can't change the name	Edit the program first
E145	USB-program to receive doesn't exist	The USB receiving program does not exist	No this program in the USB equipment, create this program again
E147	Not support file system in USB, use FAT32	The U disc isn't FAT16 or FAT32 file system	Suggest using FAT32 file system
E160	Program NO is wrong	Only (000~254) program can be input	Modify the input program number
E161	Copied program exist	The program to be copied has existed	The program has existed, this operation can't go on
E162	Program is void & can't copy	The current program is empty and can't be copied	Edit the program first
E163	Editing 253 program exceed memory	The 4M program editing storage room is full	Can't go on the next edit, except the editing program can release some storage room
E164	Exceed program memory, fail to save	Power-off protection program storage area is full and can't protect any more	Delete some programs in system
E165	No deleting all	The system parameter enactment	Modify the enactment parameter

Alarm No.	Alarm prompt	Alarm reason	Troubleshooting
	programs	can't delete all program	according to the privilege
E166	Renamed program exist	The program to be changed the name has been existed, can't change the name	Please delete the old program firstly
E167	Fail to erase FLASH	System hardware: storage CMOS chip has problem	Contact the supplier, change the storage CMOS chip
E168	Fail to write FLASH	System hardware: storage CMOS chip has problem	Contact the supplier, change the storage CMOS chip
E169	Editing program exceeds memory	The storage room of 800k program editing area is full	Can't go on the next edit, except the editing program can release some storage room
E171	No command in help	Command input error, or don't support this command	Check the command to be looking for, input the correct command again
E172	Program line too long	Edit and replace error	Check the replacing content again
E173	Replace failed, no content replace	Edit and replace error	Check the replacing content again
E174	Processing program locked, no rewriting	Set the bit parameter P416_d0 to be 1	Set the bit parameter P416_d0 to be 0
E175	Para set no line number, no arrange again	Set the P333 to be 0, the system compositor again function is invalid	Set the P333 to be non zero value again

6.4 Table of Alarms Relative to Program (i.e.E200~ E299, E600~ E699)

The types of alarm relative to programs are divided to: [Alarm in PROGRAM COMMAND], [Alarm in PROGRAM CHECK] .

6.4.1 Alarm in Program Command (i.e. E200~299)

Alarm in program command means the alarm due to incorrect command in workpiece program. It can be eliminated by inputting the correct command, and it has little relation to the setting of parameters.

Alarm List in programming command

Alarm No.	Alarm prompt	Alarm reason	Troubleshooting
E201	Illegal command	The system undefined command appears in block	Input again according system command list
E202	Command repeat	The same command is input repeatedly in the program	Delete the repeated command
E203	Illegal information	The system unidentified command appears in block	Delete the wrong command, Input the correct command
E204	Command format wrong	The command format is wrong in the block	Input again according the correct command format
E205	Command isn't compatible	Two or more commands incompatibility appear in the same block	Delete the unwanted command or line input
E206	Lack of information	Missed the command needed content in the block	Input the command needed content according the correct format
E207	Tool tip radius too much	tool nose radius is too big or path is too short, can't make the tool nose radius compensation	Modify program or tool nose radius(happen in running)
E208	Brackets is miss	Without brackets in the statement	Brackets are added to the



Alarm No.	Alarm prompt	Alarm reason	Troubleshooting
			source program
E209	Too many commands & characters	The command character in same block is over 20	Modify source program, or line input
E210	Surplus characters	Unwanted field is input in the block	Delete the unwanted field in the block
E211	Data exceeding range	The value in the block is over the specified range	Modify the is wrong data again
E212	Errors in data	The data don't meet the standard	Modify the is wrong data again
E213	Single block is too long	The whole line number of part program is over the range	Modify the workpiece program
E214	Use command independently	some commands are in an alone block without others	Modify the source program
E215	Lack of line number	The specified program line number of command G71, G72, M96, M97, M98 doesn't exist in this program i.e. the system has not found the line number for the call or the transfer	Modify the source program, input the correct block line number
E216	Line number repeat	Repeated program line number in program makes the block G71/G72/M96/M97/M98 not be identified i.e. the program has more than two call or transfer needed line number	There is repeated program line number in modifying program
E217	Program stock data has err	System EMS memory error or program error	Need the professional to check the system all sided
E218	Errors in variable NO	The variable number exceeds the set range	Using the usable variable number
E219	Edit area is unusable	System battery is shortage	Need to clear out program, Edit the program again
E220	Sentences calculate, divisor is 0	In the statement operation, the divisor is zero	Modify the divisor in macro command, not to be zero
E221	Not assign a value to variable	Input interface variable is read only, can't evaluate it i.e. input	Modify source program
E222	Program executes, information lost	The data has problem in program translate and edit process	Power on again
E224	Buffer area file too big	The editing single program is too big	Modify source program
E225	Orbit data can't compensate	Edit error, can't make the tool nose radius compensation	Modify source program
E226	Coordinate fields repeat	Coordinate field repeat	Delete one field of them
E227	Data format wrong	Data format error (such as: F1r1, G1r, M1r, T2r), or commands like G1r, M1r, T2r are used.	Modify source program
E228	Program hasn't end command	The program don't have the ending command M02 or M20 or M30	Modify source program, add the ending command
E232	Errors in command data after G92	The followed command data don't meet the format	Modify source program
E233	No user-defined command	At first make program for the needed user-defined command M60, then harden it	Write M60 program, then harden it
E234	Data field repeat	The same field in block is input repeatedly	Delete the repeat data field
E235	G71/G72 not inc & dec relations	The coordinate data in block which is followed command G71/G72 doesn't meet the condition of increase or decrease by degrees	Modify source program
E236	Forbid com exist G71/G72 (forbid commands exist in	The block followed G71/G72 command doesn't include the motion command beside G01/G02/G03	Modify source program

Alarm No.	Alarm prompt	Alarm reason	Troubleshooting
	G71/G72)		
E237	Chord is longer than diameter	Arc uses R program, the distance from starting point to end point is longer than diameter	Modify source program
E238	Dis (cen to 2 points) not equal (distances from center to 2 points aren't equal)	When the arc command G02 and G03 use K.I program, the data can't form the correct arc section	Modify source program
E239	G05 data cannot form arc	The data by G05 can't form the correct arc section	Modify source program
E240	Too many program nested calling layers	Excessive nested call in the program	Modify source program
E241	#254 not use traverse com	254 program has used motion command	Modify command or system parameter
E243	Too many data after decimal	The data behind the decimal point exceeds the limited range	At most three data is behind the decimal point
E244	Too many decimal in data	Wrong data is input i.e. 0.343.44	Modify data
E245	Lack of data after decimal	The decimal point is input, but no data followed	Delete the decimal point
E246	Not input nega (negative) NO in the field	The negative is used in this field	Check the notebook, modify the data
E247	Too much leading zero	The data like 000033 is used	Omit the needless zero in the front
E248	Not cancel tool radius compensation	Don't cancel the tool nose radius compensation function before using compound command	The cancel command G40 is added in the proper place
E249	No change in tip radius compensation	In the state of tool nose radius compensation, there is operation of changing tool	Delete the radius compensation, then change the tool
E250	G00/G01 cancel radius compensation	The command beside G00/G01 is used to delete the tool nose radius compensation	Modify source command
E251	G00/G01 sets radius compensation	The command beside G00/G01 is used to create the tool nose radius compensation	Modify source command
E252	No replacing variable in M command	Commands such as M81 I r1 are used.	Modify source command
E253	No orbit com (command) in G71/G72 cycle	The command which can produce the comparatively moving coordinate is lacked in cycle	Check the block P, Q in G71, G72 are correct or not
E254	Forbidden M3/M4 com in M47	The M3/M4 command is used in M47 state	Using the command after change the state
E255	Spindle allows to use M47	The spindle don't stop	Stop the spindle
E256	Lack of module G command	When compound command is consecutively cancelled, the G command is not used	G command is added
E257	No using the type of variable	In the special situation, this kind of variable is used in illegal	Modify source command
E258	Beveling data is too big	So large chamfer data can't be carried in this path	Reduce the chamfer data
E259	Editing is cut off	Reset key is pressed in program	
E260	No Y axis traverse in	When the Y axis has exchanged the	Modify the program according

Alarm No.	Alarm prompt	Alarm reason	Troubleshooting
	M48	spindle, the moving command can't be used in Y axis	to the real situation
E261	Drawing is cut off	The reset key is stopped in the drawing demonstration	
E262	No chamfering to full circle	Whole circle is used in chamfer	Divide into two semi-circles
E263	Use pointer variable wrongly	The pointer variable has special explanation, it must be used carefully	Modify program
E264	Arc radius is too big	The arc radius by I/K program is over the range	Modify program
E265	No calling in monitor macro	Call command is used in process monitor r7000 ~r7009 commands	Using jumping P or other method to control
E266	Pointer variable points invalid NO	The pointer variable points to the forbidden variable	Check the program, then modify it
E267	no compensate (compensating) to full circle C	Tool nose radius compensation, the program path has the whole circle program	Divide into two semi-circles
E268	Traverse is 0, retract too big (traverse amount is 0 or retraction is too big)	When making the G90 and G94 command program, the starting point and end point don't meet the regulation	Modify program
E269	Traverse amount but no offset (amount)	When making the G74 and G75 command program, it doesn't meet the regulation	Modify program
E270	Orbit (amount) bigger than (stipulated) value	When making G71 and G72 command program, the path exceeds the specified data	Modify program
E271	Start & orbit points not meet (the requirements)	When making the G71 and G72 command program, it doesn't meet the regulation	Modify program
E272	S value can't be 0 in G50	Used the G50 command, it doesn't meet the program regulation	Modify the S value
E273	Macro value is negative	The variable value is negative, but the command need it be positive number	Modify program
E275	Center of cycle isn't in coordinate	When the arc interpolation uses R program, the circle center doesn't in the stated range	Modify program
E276	Roughing blocks exceed specified value	Not conforming to the programming regulation in compound command programming	Modify program
E278	Forbidden command in G73 canned cycle	Not conforming to the programming regulation in G73 command programming.	Modify program
E279	Nesting call in this command forbidden	Not conforming to the programming regulation in G73 command programming.	Modify program
E280	No relevant command in executed program	When executing G66/G67, G67 can be executed only after G66 is executed at least once; or G67 can be used when G66 is used at least once in programming,	Re-select program; or modify program
E281	R must be 0 when using precession function	Not conforming to the programming regulation When using precession function in G92 command	Modify program

### 6.4.2 Alarm in Program Comprehensive Check Alarm (E600~E699)

Alarm in program comprehensive check alarm means the alarm due to the use of the command

which conflicts with the system parameters. When the workpiece program is checked independently, there is no problem. Therefore, it is necessary to analyze the program from setting of the whole auxiliary parameters and interface parameters, then to modify the program and parameter setting to eliminate the alarm.

[List of alarm in program comprehensive check]

Alarm No.	Alarm prompt	Alarm reason	Troubleshooting
E601	Illegal use Sxxxx : Sxxxx out of gear range	The Sxxxx command is over the parameter assured gear position range	Modify it to the gear position value in correct range or modify the P410 parameter
E602	Illegal use M21: No define M21 pin Para	The M21 interface parameter doesn't specify the exact chip pin	Set the interface parameter P500 is effective, or change the program command
E603	Illegal use M22: No define M22 pin Para	The M22 interface parameter doesn't specify the exact chip pin	Set the interface parameter P500 is effective, or change the program command
E604	Illegal use M23: No define M23 pin Para	The M23 interface parameter doesn't specify the exact chip pin	Set the interface parameter P501 is effective, or change the program command
E605	Illegal use M24: No define M24 pin Para	The M24 interface parameter doesn't specify the exact chip pin	Set the interface parameter P501 is effective, or change the program command
E606	Illegal use M91: No define M91 pin Para	The M91 interface parameter doesn't specify the exact chip pin	Set the interface parameter P517 is effective, or change the program command
E607	Illegal use M92: No define M92 pin Para	The M92 interface parameter doesn't specify the exact chip pin	Set the interface parameter P517 is effective, or change the program command
E608	Illegal use M93: No define M93 pin Para	The M93 interface parameter doesn't specify the exact chip pin	Set the interface parameter P518 is effective, or change the program command
E609	Illegal use M94: No define M94 pin Para	The M94 interface parameter doesn't specify the exact chip pin	Set the interface parameter P518 is effective, or change the program command
E610	Illegal use M78: Command invalid	The function of M78 is set to be invalid	Set the interface parameter P409_d4 is effective, or change the program command
E611	Illegal use M79: Command invalid	The function of M79 is set to be invalid	Set the interface parameter P409_d4 is effective, or change the program command
E612	Illegal use M41: Command invalid	The function of M41 is set to be invalid	Modify P410_d6=1(frequency conversion) or P410_d7=0(electrical level), or change the program command
E613	Illegal use M42: Command invalid	The function of M42 is set to be invalid	Modify P410_d6=1(frequency conversion) or P410_d7=0(electrical level), or change the program command
E614	Illegal use M43: Command invalid	The M43 function is set to be invalid	Modify P410_d6=1(frequency conversion) or P410_d7=0(electrical level), or change the program command
E615	Illegal use M44: Command invalid	The M44 function is set to be invalid	Modify P410_d6=1(frequency conversion) or P410_d7=0(electrical level), or change the program command
E616	Illegal use M32: Command invalid	The M32 function is not expanded to be effective	Set P506=1, or change the program command
E617	Illegal use M33: Command invalid	The M33 function is not expanded to be effective	Set P506=1, or change the program command

Alarm No.	Alarm prompt	Alarm reason	Troubleshooting
E618	Illegal use M10: Command invalid	The M10 function is set to be invalid	Modify P409_d7 to be effective , or change the program command
E619	Illegal use M11: Command invalid	The M10 function is set to be invalid	Modify P409_d7 to be effective , or change the program command
E620	Illegal use r10xx: No program pin	The input chip pin by variable is occupied by other function, or the parameter forbids to use the input chip pin statement program	Modify P412_d1 to be effective, or change the variable corresponding input chip pin or change the program command
E621	Illegal use r20xx: No program pin	The input chip pin by variable is occupied by other function, or the parameter forbids to use the output chip pin statement program , or used the user-defined changing tool method	Modify P412_d1 to be effective, or change the variable corresponding input chip pin or change the program command, or modify parameter P318=9
E622	No traverse command in M61 program	The parameter setting: the M61 program forbids the motion command	Change the position parameter P404_d2 to be allowed, or change the program to don't include motion command
E623	Illegal use M47	The switch setting in spindle and Y axis is not	Modify P404_d4=1, or change the program command
E624	Illegal use M48	The switch setting in spindle and Y axis is not	Modify P404_d4=1, or change the program command
E625	Illegal use M04	The spindle reversion signal is set to be without	Modify P404_d2=1, or change the program command
E628	Tool type is 9, T command need single block	When the tool post type is 9, the T command doesn't have separate section	Modify program
E629	Illegal use Y command	The control axis is set to be Y axis	Modify P405_d1=1, or change the program command
E630	Cut NO. exceed	In the program the tool number is over the setting range	Modify P319, or change the program command
E631	Illegal use G96	When the spindle S control is set to be gear shifting mode, the constant line speed cutting	Modify P410_d6=1, or change the program command
E632	No define G31I input interface, invalid	Don't set the G311 interface parameter, or the setting is wrong	Set the P532 to be effective, or change the program command

### 6.5 Table of Alarm in JOG or AUTO Operation Mode (i.e. E300~ E499)

The Alarm in JOG and AUTO is divided into two kinds: [Alarm in executing relative operation] and [Relative Alarm in executing statement]

#### 6.5.1 Alarm in Executing Relative Operations (i.e. E300~E399)

Alarm No.	Alarm prompt	Alarm reason	Troubleshooting
E302	Z driver unit alarm	Z drive unit alarm	Check Z axis drive unit, resolve the failure to power-on again
E303	X driver unit alarm	X drive unit alarm	Check t X axis drive unit, resolve the failure to power-on again
E304	Y driver unit alarm	Y drive unit alarm	Check Y axis drive unit, resolve the failure to power-on again

Alarm No.	Alarm prompt	Alarm reason	Troubleshooting
E305	Positive hardware limit alarm	X or Z positive limit switch is closed	X or Z negatively moves in JOG working mode till the alarm is released
E306	Negative hardware limit alarm	X or Z negative limit switch is closed	X or Z positively moves in JOG working mode till the alarm is released
E307	Z machine (+) software limit alarm	Z machine coordinate(Zm) exceeds the positive software limit alarm value (P015)	X or Z positively moves in JOG working mode, or set the machine coordinate to zero again, release the alarm
E308	X machine (+) software limit alarm	X machine coordinate(Zm) exceeds the positive software limit alarm value(P017)	X negatively moves in JOG working mode, or set the machine coordinate to zero again, release the alarm
E309	Y machine (+) software limit alarm	Y machine coordinate(Zm) exceeds the positive software limit alarm value(P019)	Y negatively moves in JOG working mode, or set the machine coordinate to zero again, release the alarm
E310	Z machine (-) software limit alarm	X machine coordinate(Zm) exceeds the negative software limit alarm value(P016)	Z positively moves in JOG working mode, or set the machine coordinate to zero again, release the alarm
E311	X machine (-) software limit alarm	X machine coordinate(Zm) exceeds the negative software limit alarm value(P018)	X positively moves in JOG working mode, or set the machine coordinate to zero again, release the alarm
E312	Y machine (-) software limit alarm	Y machine coordinate(Zm) exceeds the negative software limit alarm value(P020)	Y positively moves in JOG working mode, or set the machine coordinate/machine zero return again, release the alarm
E313	Z nose (+) software limit alarm	Z tool nose coordinate exceeds the positive limit alarm value(P009)	Z negatively moves in JOG working mode, or execute the tool setting/program reference point return again, release the alarm
E314	X nose (+) software limit alarm	X tool nose coordinate exceeds the positive limit alarm value(P011)	X axis negatively moves in JOG working mode, or execute the tool setting/program reference point return again, release the alarm
E315	Y nose (+) software limit alarm	Y tool nose coordinate exceeds the positive limit alarm value(P013)	Y negatively moves in JOG working mode, or execute the tool setting/program reference point return again, release the alarm
E316	E316: Z nose (-) software limit alarm	Z tool nose coordinate exceeds the negative limit alarm value(P010)	Y positively moves in JOG working mode, or execute the tool setting/program reference point return again, release the alarm
E317	X nose (-) software limit alarm	X tool nose coordinate exceeds the negative limit alarm value(P012)	X positively moves in JOG working mode, or execute the tool setting/program reference point return again, release the alarm
E318	E318:Y nose (-) software limit alarm	Y tool nose coordinate exceeds the negative limit alarm value(P014)	Y positively moves in JOG working mode, or execute the tool setting/program reference point return again, release the alarm
E319	Output frequency too high	Z/X/Y pulse output frequency exceeds the system specified range	Reduce the moving speed or modify the pulse multiplication ratio, division coefficient(P203, P204) to the proper value
E320	Thread spindle speed too high	the value of spindle speed multiplying the thread pitch(P) is more than max. speed limit of cutting feed (P113)	Reduce the feedrate or modify P113 value to the enough
E321	Can't execute G32 in DRY	The system cannot execute G32 in DRY RUN working mode	Deleted G32, then execute the program in DRY RUN working mode

Alarm No.	Alarm prompt	Alarm reason	Troubleshooting
E322	G34: Screw pitch is out of range	Before G34 thread machining, the pitch is changing, and it exceeds the permitted range or be a negative value when the end point is reached.	Modify the machining program
E330	Tool lock signal does not use	The set tool post type need the clamping signal, but the parameter P408_d6 doesn't set it to be the clamping signal	Set P408_d6 to be the using tool post clamping signal
E331	Chuck release P507 not define	The system has not defined the tool post using releasing signal (P507)	Set the parameter P507 to be the corresponding interface signal
E332	Chuck in-position P508 not define	The system has not defined the tool post in-position using signal(P508)	Set the parameter P508 to be the corresponding interface signal
E333	Tool-post gating P529 not define	The system has not defined the tool selection using strobe signal (P529)	Set the parameter P529 to be the corresponding interface signal
E334	Tool-post locking is overtime	In tool post rotation(CCW) time in tool change, it can't inspect the tool post clamping signal	Check the tool post clamping signal is in normal or not, or delay the reversion time(P324)
E335	Tool-post NO alarm	After tool change completed, the tool selection number isn't the target number	Check the target tool selection signal is in normal or not
E336	No changing tool	The parameter set the tool changing forbidden	Modify the forbidden tool changing parameter P403_d7
E337	Detect tool post signal overtime	In tool post rotation (CCW) time in tool change, it can't inspect the tool selection number	Check the target tool selection signal is in normal or not, or delay the time for tool changing in properly
E338	Chuck doesn't clamp, no spindle	Forbid starting the spindle when the chuck is released in spindle and chuck locking each other	Execute the chuck clamping operation to clamp the workpiece
E339	Spindle doesn't stop, no chuck	When the spindle and chuck are locking each other, don't operate the spindle without stopped the spindle	Stop the spindle, then operate the chuck
E340	Forbid function of chuck	The parameter set chuck control forbidden	Modify parameter P409_d7
E341	Spindle doesn't stop, no tail	When the spindle and chuck are locking each other, don't operate the tailstock without stopped the spindle	Stop the spindle, the operate the tailstock
E342	Forbid function of tail stock	The parameter set tailstock control forbidden	Modify the parameter P409_d4
E343	M10 no respond	The system cannot check the valid chuck clamping in position signal RM10 in controlling chuck clamping	Check the chuck clamp selection signal is in normal or not
E344	M11 no respond	The system cannot check the valid chuck releasing selection signal RM11 in controlling chuck releasing	Check the chuck releasing selection signal is in normal or not
E345	M78 no respond	the system cannot check the valid tailstock going-forward in-position signal RM78 in controlling tailstock going-forward	Check the tailstock signal is in normal or not
E346	M79 no respond	The system cannot check the valid tailstock going-backward in-position signal alarm 79 in controlling tailstock going-backward	Check the tailstock going-backward signal is in normal or not
E347	The non-frequency spindle	The system can't execute the constant surface speed command G96 or gear shifting command M41~M44 in non-frequency spindle	Modify the spindle collocated parameter P410_d6
E348	The program is not ready	Program error, execution is forbidden	Modify the program, resolve error



Alarm No.	Alarm prompt	Alarm reason	Troubleshooting
E349	Use 【 return machine zero 】 key	When the machine is back to zero function, the modifying manual is forbidden	Using “ MACHINE ZERO RETURN” key to zero
E350	S exceed limit	The Sxxx value exceeds the gear shifting range in the non-frequency spindle gear shifting	Modify the S value to the correct gear value
E351	Spindle is controlled by Y axis	When the spindle is controlled by Y axis, the start and stop of the spindle can't be controlled in JOG	Execute M48 to get back the spindle function
E352	Spindle Y, levels can't be changed	The spindle can't execute the gear shifting command when it is controlled by Y axis	Execute M48 to get back the spindle function
E353	M47/M48 function doesn't work	The M47/M48 function forbidden is set in parameter when there is no Y axis(P405_d1) or Y axis can't be changed (P410_d4)	Modify parameter P405_d1 and P410_d4
E354	Spindle doesn't stop, no M47/M48	The M47/M48 is forbidden to execute when the spindle don't stop	Stop the spindle, then execute M47/M48
E355	254 program error	254 program error, the machining program is forbidden to execute	Modify 254 program, resolve error
E357	Feeding override is 0.	The override is zero in feedrate	Adjusted to feedrate override nonzero
E358	Spindle running is abnormal.	The spindle rotate speed is checked to be zero after the spindle is started	Check the spindle relative part, resolve the error or modify parameter P402_d6 to forbid its function
E359	Alarm: pressure is low.	The system has checked the low pressure and lasted the parameter P332 specified time	Check the hydraulic pressure equipment, resolve the error
E360	Alarm: protecting door isn't closed	The safety door don't close in AUTO	Close the safety door
E361	Material convey alarm	The system has checked it's out of rod in M20, automatism stop alarm	Change the rod, then go on the machining
E362	Lubricating doesn't start	The system executes the M32/M33 command without lubricating function	Set the parameter P506 to be the corresponding lubricating control interface
E363	E363: Tool NO exceeds limit	The target tool number exceeds the parameter P319 specified value in changing tool	Modify tool number or modify P319 to be the maxima system I tool number
E364	Tool post release overtime	The tool pose champing signal is effective always in tool changing and releasing tool post	Check the tool pose champing signal and resolve error
E365	Y axis is speed control mode	The spindle and Y axis allow change(P410_d4), but the system don't allow to control the Y axis moving in current spindle non-position control mode	Execute M47 first, change the spindle to location control mode
E366	Clamp in-position signal abnormal	When the following chuck respond signal checking alarm function is open, the chuck champing in-position signal is invalidation	Control the chuck champing again or resolve the chuck error
E367	Release in-position signal abnormal	When the following chuck respond signal checking alarm function is open, the chuck champing in-position signal is invalidation	Control the chuck champing again or resolve the chuck error
E368	Tail stock FW signal abnormal	When the following tailstock respond signal checking alarm function is open, the tailstock going-forward in-position signal is invalidation	Control the tailstock going-forward again or resolve the chuck error



Alarm No.	Alarm prompt	Alarm reason	Troubleshooting
E369	Tail stock BK signal abnormal	When the following tailstock respond signal checking alarm function is open, the tailstock going-backward in-position signal is invalidation	Control the tailstock going-backward again or resolve the chuck error
E370	P319 (number of tools) and P320 (tool position signal number) are not equal	The parameter setting P319≠P320 in tool changing, the position parameter P408_d7=0	Set the parameter P408_d7=0-1 again or let P319 equal to P320
E371	External start signal abnormal	The outside starting key always isn't released when the system changes to JOG or AUTO working mode	Check the outside startup signal and resolve the error
E372	Chuck foot switch signal abnormal	The chuck foot switch always isn't released when the system changes to JOG or AUTO working mode	Check the chuck foot switch signal and resolve the error
E373	Tail foot switch signal abnormal	The tailstock foot switch always isn't released when the system changes to JOG or AUTO working mode	Check the tailstock foot switch signal and resolve the error
E374	G31 monitor interface undefined	The parameter P532 is undefined(G31 input checking)	Set interface parameter P532 as a fixed pin
E375	Spindle doesn't stop	When the system is in the tapping in JOG working mode, the spindle doesn't stop or the rotate speed isn't zero	It's allowed to enter the tapping in JOG working mode when the spindle has stopped and the rotate speed is zero
E376	Manual tapping speed is too high	The manual tapping rotate speed is too fast	Reduce the tapping rotate speed in properly
E377	Commands have been undefined	Used the undefined command M60~M74	Modify the program
E378	External pause signal abnormal	The external pause key is not released during MANUAL/AUTO mode switching	Check the external pause signal
E379	Feed hold switch is not reset	Workpiece program cannot be executed when the feed hold switch is not cancelled	Check feed hold signal
E380	Spindle/feed hold switch is not reset	Workpiece program cannot be executed when the spindle/feed hold switch is not cancelled	Check spindle/feed hold signal
E382	Tailstock is not ready; spindle operation is forbidden	Spindle is operated when tailstock is not ready	Check the tailstock

### 6.5.2 Relative Alarm in Executing Statement (E400~ E499)

Alarm in statement program means the alarm due to an incorrect statement command in the workpiece program when using statement programming. The alarm can be resolved by inputting the correct command.

Alarm No.	Alarm prompt	Alarm reason	Troubleshooting
E400	Monitor variable not described	The system doesn't evaluate and set the judgment condition to the process monitor before startup	Modify workpiece program
E401	In macro, divisor	The divisor is zero in statement operation	Modify the divisor in macro command, let it don't be zero

Alarm No.	Alarm prompt	Alarm reason	Troubleshooting
	is 0		
E402	not support monitor calculation	The process monitor assignment statement has assignment operation which is not supported by the system, i.e. square root, sine operation	Modify program
E403	r5008 data error	Evaluate the variable r5008 illegal value	Modify program
E404	Variable data r6xxx error	Illegally used the undefined r6xxx series variable	Modify program
E405	Variable data r4xxx error	Illegally used the undefined r4xxx series variable	Modify program
E406	r4003 can't be written	The illegal written operation is made to variable r4003(spindle real-time speed)	Modify program
E407	r4007 can't be written	The illegal written operation is made to variable r4007(spindle real-time speed)	Modify program
E408	r4008 can't be written	The illegal written operation is made to variable r4008(spindle real-time speed)	Modify program
E409	Not assign a value to r6xxx in monitor	The condition statement in process monitor evaluates the r6xxx series variable	Modify program
E410	r711x Invalid data	Write the undefined value to r711x series variable	Modify program
E411	r4002 can't be written	Written operation is made to the read only machining time variable r4002	Modify program
E412	The variable doesn't support this kind of operation	The system non-supported operator is used in the assignment statement	Modify program
E413	The unknown process monitor description	The system non-supported judgment condition is used in the statement of process monitor description	Modify program
E414	Unknown pointer variable	The pointer variable points to the non-common variable or the exceeded range pointer variable is used	Modify program
E415	The variable r4004 can't be written	Making written operation to reading variable r4004	Modify program
E416	The variable r4012 can't be written	Making written operation to reading variable r4012	Modify program

## CHAPTER SEVEN STATEMENT PROGRAMMING (not for GSK928TD)

This system has provided the programming method similar to the high level language, which realizes the variable assignment, arithmetic operation, logic judgment and conditional transfer. Programming using the statement and variable can realize the functions which can't be achieved by the common G and M commands.

### 7.1 Variable

#### 7.1.1 Variable Expression Method

The variable value can be assigned by the program command or set by keyboard directly. Multiple variables can be used in a program and they can be distinguished by the variable number.

##### The variable expression method:

Using lowercase letter "r" + variable number( four-digit integer) to express, the leading zero can be omitted.

Such as:

r5: referred to as number 5 variable; r1003: referred to number 1003 variable

#### 7.1.2 Classification of Variable

According to the function and purpose, the variables in this system can be divided into: common variables, pointer variables, input/output interface variables, special variables in system internal, keyboard scan command register variables, display command register variables, function command register variables, process monitor/ process monitor management variables, pulse signal monitor /pulse signal monitor management variables, etc.

Different types of variable have different functions, and their use and value ranges are different too. The types are described below:

Variable type	Variable range	Remarks
Common variable	r001~r299	
Pointer variable	r9000~r9049	
Input interface variable	r1001~r1032	
Output interface variable	r2001~r2032	
Register variable	r5001~r5004, r5008	
System variable	r4001~r4012, r6001~r6006, r6101~r6164, r6201~r6264	
Process monitor variable	r7000~r7009	
Process monitor management variable	r7010~r7019	
Pulse signal monitor variable	r7100~r7103	
Pulse signal monitor management variable	r7110~r7113	
Variable transfer register variable	r7900	

### 7.1.2.1 Command Variable

The variable number range of common variable is from 001 to 299 ( 299 variables in total).

The common variable is used to store a data which can participate in operation and also be quoted by data word of G command, etc.

The data value range of command variable: -999999999~999999999

#### ◆ Common variable reference

In the workpiece program, the system can use the variable to replace the data word values

Format: <address> + “r”, means to treat the variable value as the word value

For example:

Fr003 ; when r003 equals 15000, its function is same as that of F15 command.

Zr010 ; when r010 equals 2500, its function is same as that of Z-2.5 command.

K r 010 ; when r010 equals 2500, its function is same as that of K2.5 command.

S r 003 ; when r003 equals 2000000, its function is same as that of Sr003 command.

#### 【For example】

N0010 r001=-3700

N0020 r002=150000

N0030 G00 Z100 X80

N0040 G01 Ur001 Fr002 ; the same as U-3.700 F150.000 command function

N0050 G01 Wr001





N0060 M02

#### 【Explanation】

- 1) The variable value doesn't include decimal point, and is obtained by the representative value multiplied by 1000. When it is replaced to the words with decimal point, e.g. coordinates, divide the variable value by 100, then the coordinate data including three decimals is obtained;
- 2) In the command symbol G, M, T, the variable can't be quoted; Gr003 can't be used for programming;
- 3) The words for the program line number (i.e. P, Q etc) can't quote variables;
- 4) The word L indicating times and word H indicating location sign cannot quote variables;
- 5) Only the common variables can be quoted by command word; otherwise, an alarm occurs;
- 6) When the program is in AUTO operation mode, if the variable used in the main program is also used in the subprogram(i.e.: M60~M74 command), please note the intercross use and change of the variable in the main program and subprogram will affect the result of program execution.

#### ◆ The characteristics of common variable:

When the system executes the program in AUTO operation mode, the variable value may change with the program path. The common variable has the following characteristics:

- 1) The system performs different initiation process according to the different sections of variable number.
  - r001~r040: The variable parameter table offers the initial values for them (set by the user as required)
  - r041~r099: This group of variables are assigned 0 in the **initial state** of program execution in AUTO mode.
  - r100~r199: They are cleared upon power-on, and are not cleared afterwards.
- 2) In the AUTO operation mode, before executing the first program, by pressing the **CYCLE START** key, the system will automatically load the contents of P600~P639 in the parameter table into variable r001~r040 as the initial value; the variable r041~r099 is cleared.
- 3) After executing M20, in the recycle, r001~r040 are assigned the initial value by the P600~P639 in parameter table again; r041~r099 and r100~r199 are not cleared.
- 4) The initial values of r100~r199 are zero after power-on, later they are not cleared automatically. However, their values can be changed in JOG or AUTO operation mode in real time. Their initial values can be set by the following method ( When using this group of variables for programming, if the variables are not assigned the initial values before using, there is uncertainty in the program, so the user should be careful). When the user is using these variables and needs the initial values, the initial values of r100~r199 can be set in JOG operation mode. In the JOG or AUTO operation mode, before the execution of the program, press  to call the macro variable display window which displays all the macro variables in the program, then press  and ,  to move the pointer to the variable to be modified, press **ENTER** key to choose the variable and input the data, then press **ENTER** key again. It is forbidden to modify the variable value during program execution.

#### ◆ Double-precision floating-point variable \_ common variable r200~r299

The double-precision floating-point variable is the common variable; its range is between 200~299, (100 variables in total).


As data overflow is easy to occur in the original command variables (r001~r199), the use of double-precision floating-point variable can solve this problem. The data range of double-precision floating-point variable is: 64-digit number with sign.

#### 【Explanation】

- 1) Double-precision floating-point variable cannot be quoted by a code word; otherwise, an alarm will occur.
- 2) For the other explanations, refer to common variable (r001~r199).

#### 【Characteristics】

In AUTO mode, the value of double-precision floating-point variable may vary with the program path. The double-precision floating-point variable has following characteristics:

- 1) In AUTO mode, in the initial state of program execution, this group of variables is assigned 0.
- 2) After the execution of M20, the variable is not cleared when it is cycled.
- 3) In MANUAL or AUTO mode, press  to call macro variable display window which does not display double-precision floating-point variables.

### 7.1.2.2 Pointer Variable

The variable number range of the pointer variable is from 9000 to 9049 (50 variables in total)

The pointer variable is specially used to point to the address of a common variable. To assignment to it means assigning the value to the pointed common variable; and its value means the value of the pointed common variable. The pointer variable can participate in the operation or reading judgment.

#### ◆ The method of creating and changing pointer

The pointer must be created correctly and point to a specific common variable before using the pointer variable. The pointing sign is indicated by “-“ and “>” and a blank is allowed between them.

The variable format of creating and changing pointer: Pointer variable name -> Expression

E.g. r 9001 -> 1 ; r9001 pointer variable points to r001 common variable;

r 9002 -> 199 ; r9002 pointer variable points to r199 common variable;

r 9003 -> r100 ; when r100 equals 5, r9003 points to r005 common variable

r 9003 -> r 9003+1 ; r9003 points to next common variable, if it points r5 at first, then now it points to r6;

r 9003 -> r 9003-1 ; r9003 points to the last common variable, when it points r6 at first, then now it points to r5;

#### ◆ The characteristic of pointer variable:

- 1) First set up the pointer variable and make it point to a certain specific common variable, then perform the read/ written operation to it or use it for judgment; otherwise, an alarm occurs.
- 2) In setting up the pointer variable, its value range is from 1 to 199 (i.e. it can only point to r1~r199).An alarm occurs when the range is exceeded;
- 3) The statement of setting up or changing the pointer variable can only be written in a independent block; otherwise an alarm occurs; the expression in the right side of pointing symbol can be the addition and subtraction operation between two of them;
- 4) The pointer variable can only be used in the numerical value operation, and it cannot be quoted by the data word like G command, etc; e.g. for Ur9000, an alarm occurs;
- 5) In the assignment statement, conditional judge statement, the assignment of the pointer variable means assigning the pointed common variable; while reading the pointer variable means reading the value of the pointed common variable.

#### ◆ The use of pointer variable:

##### 【Program example】

In the following program, each time the Z/X axis moves a position, the coordinates of the position are saved to the command variables (there are 10 groups of coordinates in total); then these 10 groups

of coordinates are taken out one by one and return to the initial position by the original path. The following is realized by the pointer variable.

```

N0000 G00 Z200 X200          ;
N0010 r9000 -> 1             ; r9001 pointer variable points to r001 common variable;;
N0020 M98 L10 P1000         ;
N0030 M98 L10 P2000         ;
N0040 M02                   ;
N1000 r9000= r6004          ; In the first calling, stores the Z coordinate of this position to r1
N1010 r9000 -> r9000+1      ; In the first calling, the pointer +1 and points to r2
N1020 r9000= r6005          ; In the first calling, stores the X coordinate of this position to r2
N1030 r9000 -> r9000+1      ; In the first calling, the pointer +1 and points to r3
N1040 G00 W-25              ;
N1050 G00 U-15              ;
N1060 M99
N2000 r9000 -> r9000-1      ; In the first calling, the pointer -1 and points to r20
N2010 r190= r9000           ; In the first calling, takes out the X coordinate of the last but
                             ; two point from r20 and stores it to r190
N2020 r9000 -> r9000-1      ; In the first calling, the pointer -1 points to r19
N2030 r191= r9000           ; In the first calling, takes out the Z coordinate of the last but two
                             ; point from r19 and stores it to r191
N2040 G01 Xr190 F1000       ;
N2050 G01 Zr191             ;
N2060 M99

```

### 7.1.2.3 Interface Variable

This group of variables corresponds to input/output pins. They are used to check and control the input/output pin signals of the system. The interface variables are divided into input interface variables and output interface variables.

#### ◆ Input interface variable (read only)

The system has 32 input interface variables, each variable corresponds to one input signal pin ( the variable number r1001 ~r1032 corresponds to the signal pin number U1001~U1032 respectively)

Note: The variable number r1000 is the integer state of 32 input interfaces; corresponding to signal pin U1001 ~U1032 from low to high.

The two state values of the input interface variable is 0 (low level) or 1(high level). The low level means the external signal is effective, which indicates the pin is connected to 0V; the high level means the external signal is ineffective.

The input interface variable is a group of read-only variables, which cannot be assigned values (written); otherwise, an alarm occurs.

After reading the value of input interface signal r1001~r1032 through the programming, the signal level corresponding to the pin can be known. Then the system can perform program skip using the level as a judgment condition.

#### 【For example:】

```
r8=r1002 ; Reading UI02 pin level; r8=0 or 1.
```

if (r1009=0) then(P1600) ; if the UI09 pin read is the low level, then the system goes to P1600.  
r1007= r8 ; System issues the alarm: “ Assignment cannot be done to read-only variables”.

◆ Output interface variable (readable/writable)

The system has 32 output interface variables, each variable corresponds to one output signal pin ( the variable number r1001 ~r1032 corresponds to the signal pin number U1001~U1032 respectively). the output state of the corresponding pin can be changed by assigning a value to the output interface variable.

When the system outputs “0”, the outside can form the conductive loop;

When the system outputs “1”, the interface is in the high resistance state, and the outside cannot form the conductive loop.

The current assignment state of the output interface can be obtained by reading the output interface variable.

It does not mean that all the output signal pins can be read; when bit parameter **P412\_d1=0**, the statement programming can only be performed to the pins which display “UO” on the Diagnosis page. If the programming is performed to the output signal pins which have been defined, an alarm is issued.

【Example】

R2002=0 ; Outputs “0” to UO02 pin, the outside can form the conductive loop.

R2016=1 ; Outputs “1” to UO16 pin, the interface is in the high resistance state, and the outside cannot form the conductive loop .

r2007= r8 ; Determines the UO07 output state according to the content of r8 ( zero or non zero)

if (r2016=1) then(P0100) ; If the output of current UO16 pin read is “1”, then the system goes to P0100.

【Explanation】

- 1) The assignment output is only allowed for the variables of the output pins which are released by the standard function definition due to the system limitation; otherwise, an alarm occurs; i.e. the statement programming is only allowed for the pins with “UO” displayed on the Diagnosis page. If programming is performed to the output signal pins which have been defined, an alarm occurs.
- 2) In the system boot-strap and power-on beginning, the system will set r2001~r2032 to be “1”, and the outside cannot form the conductive loop.

#### 7.1.2.4 Keyboard Scan Register r5001

Keyboard scan register: r5001 value range: 32 digits without sign.

The different numerical values written to the register have different meanings; the system executes the corresponding function command according to the written numerical value. Please refer to the table below:

Table of r5001 executable keyboard scan commands: (the system forbids using this variable in the process monitor description statement)



Command No.	Meaning
1	Keyboard scans once; When the key is pressed, the r5001 equals the key value ; when there is no key pressed, r5001 equals zero;
2	Waiting for the keyboard to input a character ; waiting until the key is pressed, and then recording it to r5001;
3	Waiting for the keyboard to input a character, with cursor and character displayed; the value is recorded to r5001;
4	Waiting for the keyboard to input a character and releasing the press key;
5	Waiting for the keyboard to input a character and releasing the press key, with cursor and character displayed;
6	Waiting for the keyboard to input a numerical value, and ending with ENTER; or you can quit by pressing ESC, then the input value will be zero.
<b>Note: If the assignment is not within 1~6, it is ineffective.</b>	

**【Use】**

R5001 assignment is the operation mode for setting keyboard register, the key value obtained after executing the command is stored in register r5001, and can be read or used for the conditional judgment.

**【Example】**

r5001=4; The system is waiting for the keyboard to input a character, it will not proceed to the next until the key is released.

**【Explanation】**

- 1) If the keyboard scan command 3 or 5 is chosen, the scanned key will be displayed in the display window when the window is open; if the display window is not opened, the key cannot be displayed after this operation.
- 2) In waiting for key pressing, if the ESC key is pressed, the press key input is cancelled, and the next statement is executed.
- 3) The acceptable key symbols are shown in below table, the other key input is ineffective. Note: The key value is the corresponding ASCII value of this key symbol.

Correspondence table between r5001 acceptable key symbols and obtained key values:

key symbol	0	1	2	3	4	5	6	7	8	9	.	-	ENTER	ESC
Key value	48	49	50	51	52	53	54	55	56	57	45	46	13	01
Key symbol	G	M	X	Z	S	T	U	W	F	I	K	D	R	
Key value	71	77	88	90	83	84	85	87	70	73	75	68	82	

7.1.2.5 Display Window Register r5002

Display window register: r5002 (the system forbids using this variable in the process monitor description statement)

Through the operation to the display window register, the system can construct a window on the display screen to prompt the program execution steps, alarm etc information.

The different values written to the register have different meanings; the system executes the corresponding function command according to the written value. Please refer to the following table:

Table of r5002 executable commands:

Command No.	Meaning (the user information can only be displayed on the display window)
0	Close the display window;
1	Open an empty display window;
2	When the window is not opened, call the display window which has opened last time;
10	Set the cursor position, the cursor points to the first line home;
11	Set the cursor position, the cursor points to the next line home;
12	Erase the content from the current cursor position to the line end; the cursor position does not change;
13	Erase all the content in window, the cursor points to the first line home;
32~126	Send an ASCII character to the display window; (see ASCII code table)
201~209	The cursor is positioned to the specified line, the range is from 1 to 9;
210~230	The cursor is positioned to the specified line, the range is from 10 to 30;
1000~1099	Send a group of character strings to the display window; the mantissa 00~99 are character string numbers. Up to 100 groups of character strings can be displayed;
2000~2999	Send a alarm number to the display window: displaying "alarm XXX", the range of XXX is from 0 to 999;
110001~110009	Set the line number of the display window (line 1 to 9); (the system default is 6)
110010~110040	Set the displayed character number in each line ( 10 to 40); (the system default is 30)
120000~120255	Set the window grounding or character grounding, 0 to 255 can be chosen; ( the system default is white, and the value is 255). See correspondence table between colors and code values
130000~130255	Set the character color, 0 to 255 are available ( the system default is black, the value is 0)
140000	Set the size of the character: 16*16
140001	Set the size of the character:: 8*16 (system default)
Note: when the assignment exceeds the range, it is ineffective.	

**【Use】**

r5002 assignment is to set the command for displaying window register; the data is uncertain when reading the register, so it cannot be used as the conditional judgment.

**【Example】**

```
r5002=1      ; Opens an empty display window
r5002=49     ; Sends the character "1" to the display window
r5002=49+r3  ; Sends the character which deviates r3 position from "1" to the display window
r5002=2001   ; The display window displays " Alarm 001"
```

**【Explanation】**

- 1) Set the size and color of the window before opening the display window; otherwise, they are the system defaults;
- 2) When the current window is opened, if the parameters such as the window size is re-set, the window content will be cleared automatically, and then the new setting takes effect;
- 3) For the selection of grounding and font color, refers to the attached table in this chapter: the corresponding table between usually-used colors and command values;
- 4) In the single block mode in AUTO operation, if there is a window displaying characters currently, the window is closed by pressing **ENTER** key is pressed, then the program exits the execution.

## ◆ Expression of character string:

In the block, one of the below two expressions can be used to input the character string:

## 1) Common expression: Format: Serial number+ blank + “S” + “display character string” ;

Example: If the contents to be displayed of the character string 00 is “ 1~9 please press a key to choose the working procedure to be repeated:”, the block content is as follows:

```
N5000 00 S 1~9 please press the key to choose working procedure to be repeated;
N5010 r5002=1000 ; Displays the set 0 character string r5010
```

## 2) ASCII code decimal expression

Format: Serial number + blank + “D” + character code + blank space +character code +blank space +character code...

Example: If the contents of 04 character string to be displayed is “1-9 GongXu:”, the block contents are as follows:

```
N5000 04 D 49 95 57 32 71 111 110 103 88 117 58
N5010 r5002=1004 ; Displays the fourth group of character strings
```

## 【Explanation】

- 1) The serial number must be a two-digit number, the range is from 00 to 99 (100 groups of character strings in total), and the lowered two digit of the command word must correspond to the serial number, then the contents of this character string can be displayed;
- 2) The contents behind the S blank is regarded as the contents of the character string by the system; No comment can be added behind the character string;
- 3) Each group of character strings cannot exceed 40 characters (20 Chinese characters)
- 4) The character string can be written before or behind the program; Generally, it is written in the end, which will not affect the program execution speed.

## 【Program Example】

It mainly demonstrates the use of keyboard scan register and display window register in the following program. Suppose there are three working procedures, and each of them moves to different position from start point, then goes back to start point. However, the three working procedures are chosen by pressing keys according to the pop-up window prompt rather than being executed by turn.

```
%101
N0000 G00 Z0 X0
N0010 G00 Z100 X100
N0020 r5002 = 110003 ; Sets the number of lines for the display window
N0030 r5002 = 110016 ; Sets the number of displayed characters of each line
N0200 r5002 = 1 ; Opens an empty display window
N0210 r5002 = 140001 ; Set the character size: 8*16
N0220 r5002 = 130000 ; Sets the character color: Black
N0230 r5002 = 1004 ; Displays the 004 character string
N0240 r5002 = 11 ; Sets the cursor position, the cursor points to the home of next
line
N0250 r5002 = 140000 ; Sets the character size: 16*16
N0260 r5002 = 1006 ; Displays the 006 character string
N0270 r5002 = 11 ; Changes line
```

```

N0280 r5002 = 130006      ; Changes color: Blue
N0290 r5002 = 1005       ; Displays the 005 character string
N0300 r5002 = 130168     ; Changed color: Red
N0500 r5001 = 4          ; Waits for the keyboard to input a character, then executes the
                          next block after releasing the key
N0510 if(r5001 = 49) then(P1000) ; If the character "1" is input, executes P1000
N0520 if(r5001 = 50) then(P2000) ; If the character "2" is input, executes P2000
N0530 if(r5001 = 51) then(P3000) ; If the character "3" is input, executes P3000
N0540 if(r5001 = 48) then(P600)  ; If the character "0" is input, executes P600, ends
N0550 M97 P500
N0600 M02
N1000 r5002 = r5001      ; Displays the character "1" input from the keyboard
N1020 G01 Z50 X10 F2000
N1030 G00 Z100 X100
N1040 M97 P200
N2000 r5002 = r5001      ; Displays the character "2" input from the keyboard
N2020 G01 Z120 X50 F2000
N2030 G00 Z100 X100
N2040 M97 P200
N3000 r5002 = r5001      ; Displays the character "3" input from the keyboard
N3010 G01 Z160 X90 F2000
N3020 G00 Z100 X100
N3040 M97 P200
N4000 ; No comment can be added behind the character string
N4000 ; The character string can be written finally, which will not affect the execution speed
N5000 04 D 49 95 51 32 71 111 110 103 88 117 58
N5010 05 SINPUT
N5030 06 S Three working procedures in total

```

### 7.1.2.6 Display Value Register r5003

Display value register: r5003 (it is forbidden to use this variable in the description statement of process monitor)

#### 【Use】

The r5003 assignment is a command to display this value(it can be viewed when the display window is opened. The data is uncertain when reading the register, so it cannot be used as the conditional judgment.

#### 【Example】

```

r5003=r032      ; Displays the value in r032
r5003=r03+r01   ; Displays the value of r03 plus r01

```

### 7.1.2.7 Graph Refresh Register r5004

Graph refresh register: r5004 (It is forbidden to use this variable in the statement description of process monitor)

**【Use】**

The r5004 assignment is a command to clear the machining graph display area ((it can be viewed when the graph display window is opened). the contents of the system machining graph display area are cleared if any data is written to r5004.

**【Example】**

r5004= 0 ; Clears the contents of the display area, i.e. the machining graph path

Note: In the cycle machining, the last machining graph is not cleared; in order to observe the new machining path and graph conveniently, execute this command to display the next machining path and graph again at the end of the machining program,

7.1.2.8 Program Control Register r5008

Program control register: r5008 (usually used for description program of process monitor; do not used in common programs). If the data is uncertain when reading this register, it cannot be used as the reading or conditional judgment.

Different values written in the register have different meanings; the system executes corresponding function command according to the written value.

Table of r5008 executable commands

No.	Meaning
1	Pause: movement stops (equals to key CYCLE PAUSE); press CYCLE START to resume;
2	Switch between SINGLE/CONTINUOUS mode (equals to key SINGLE); press CYCLE START to resume; (usually used for program description during monitoring. If it is used in common program, M00 command is advised. )
3	Stop after cycle (movement stop after M20); press CYCLE START to resume;
4	Cycle starts (equals to key CYCLE START); usually used for description of process monitoring;
5	Set to SINGLE mode; press CYCLE START to resume;
6	Set to CONTINUOUS mode
11	Foot switch is allowed for chuck operation afterwards; M10/M11 switching; If spindle and chuck are interlocked, it is valid only under M05 command;
12	Foot switch is allowed for chuck operation afterwards; M10/M11 switching; If spindle and chuck are interlocked, it is valid only under M05 command and the actual speed is 0;
13	Foot switch is not allowed for chuck operation afterwards (except for SINGLE PAUSE); if the chuck operation is not finished yet, the next program cannot be executed;
14	Foot switch is allowed for chuck operation afterwards; M78/M79 switching; If spindle and chuck are interlocked, it is valid only under M05 command;
15	Foot switch is allowed for chuck operation afterwards; M78/M79 switching; If spindle and chuck are interlocked, it is valid only under M05 command and the actual speed is 0;
16	Foot switch is not allowed for chuck operation afterwards (except for SINGLE PAUSE); if the chuck operation is not finished yet, the next program cannot be executed;
17	Goes to next program if M10 is not finished; does not take up time;
18	Goes to next program if M11 is not finished; does not take up time;
19	Goes to next program if M12 is not finished; does not take up time;
20	Goes to next program if M78 is not finished; does not take up time;
21	Goes to next program if M79 is not finished; does not take up time;
22	Goes to next program if M80 is not finished; does not take up time;
23	Goes to next program if M3 is not finished; does not take up time;
24	Goes to next program if M4 is not finished; does not take up time;
25	Goes to next program if M5 is not finished; does not take up time;
40	Does not go to next program till all M commands are finished;

**【Explanation】**

- 1) During program execution, the operation of chuck and tailstock controlled by foot switch equals to inserting a work during the execution of machining program, which does not take up time. If the command in the same group is executed, it is invalid; it is valid by pedaling the foot switch again after the execution.
- 2) In statement or whole process monitor, the operation of chuck and tailstock controlled by r5008 equals to inserting a work during the execution of machining program, which does not take up time;
- 3) The operation of chuck and tailstock controlled by r5008 may be invalid; for instance, when M11 is being executed, and the chuck is clamped using r5008, the operation is invalid;
- 4) If the spindle is not interlocked with the chuck and tailstock, the operation of chuck and tailstock controlled by r5008 is valid regardless the spindle state and spindle speed in the above table 2.
- 5) In program end, reset or emergency stop state, the current command is cancelled automatically.

**【Use】**

Assignment to r5008 means choosing program pause and start mode; usually it is used together with monitoring description; see the example of monitor description;

**【Example】**

- r5008=1 ; Program pauses immediately; press CYCLE START key to resume;  
 r5008=3 ; Pause after cycle ends; press CYCLE START key to resume;

7.1.2.9 System Special Variable Group 1

System special variable group 1: readable/ writable (in the process monitor description statement, read only, write forbidden).

Variable No.	meaning
r6001	Z machine coordinate
r6002	X machine coordinate
r6003	Y machine coordinate
r6004	Z tool nose coordinate
r6005	X tool nose coordinate
r6006	Y tool nose coordinate
r6101...6164	Z tool offset 01...64
r6201...6264	X tool offset 01...64

**【Use】**

For the writable variables, the assignment is a command to modify the parameter, at the same time the variables save the assignments and can be used as the reading or conditional judgment. Usually, the system inner special variables should not be used, and the use must be careful when using them.

**【Example】**

- r6001=150000 ; Modifies the current Z axis machine coordinate to 150.000  
 when (r6005>3000) then P1500 ; If the current X tool nose coordinate is greater than 3.000, then turns to P1500  
 r6201= r6201+20 ; Adds 0.020 to the X tool compensation of the first group tool offset number

The value range of the system inner special variables: 32 digits with sign.

7.1.2.10 System Special Variable Group 2

System special variable group 2 (used for the monitor description)

Variable No.	Meaning	Property
r4001	Workpiece counter: the workpiece count displayed on screen In the program execution, the counter will add 1 automatically when it meets a M02 or M30 or M20.	Read/write
r4002	Time counter: workpiece processing time; unit: ms	Read only
r4003	Spindle real-time speed recorder : the real-time speed detected by the system; unit: r/min	Read only
r4004	Spindle program rotate speed recorder : in executing the S command, the system will automatically program the speed to this variable first.	Read only
r4005	Target tool number recorder: the system will input the tool position number to this variable automatically in executing T command;	Read/write
r4006	The target tool offset number recorder: the system will input the tool offset number to this variable automatically in executing T command; 0≤tool offset number≤64	Read/write
r4007	The current tool number recorder: after the tool changing finished every time, the system will input the tool position number to this variable automatically.	Read only
r4008	The current tool offset number recorder: after the tool changing finished every time, the system will input the tool position number to this variable automatically. 0≤tool offset number≤64;	Read only
r4009	Instantaneous interference recorder: In the AUTO operation mode, when the system has checked the recently press key “G” or “M”, it will input the corresponding ASCII value 71/77 to r4009; and display the character on the top left corner of the screen. Write random number except the 71/77, or press X key, or enter the AUTO operation mode, the variable value will be cleared to zero. Using r4009, two kinds of program branch transfer function can be added, which are used for the specific manual interference temporally during machining.	Read/write
r4010	Timer : In the AUTO operation mode, when the variable value is not zero, it will automatically reduce by 1 per ms till it is zero; it can be the timer used for conditional judge ;	Read/write
r4011	Timer: In the AUTO operation mode, when the variable value is not zero, it will automatically reduce by 1 per ms till it is zero;	Read/write
r4012	Spindle coder : When reading the spindle coder, the value range read is : among 0~ (four times of actual coder LINEs); when the coder LINEs (parameter P209) are set to 1200, the value range read will be 0~4800.	Read only

**【Use】**

For the writable variables, the assignment is a command to modify the parameter, at the same time the variables save their assignments, and can be used as reading or condition judgment.

**【Example】**

if (r4008=1) then P0060 ; If the current offset is No. 1 tool offset, turns to P0060  
 if (r4001=1000) then P1500 ; If the machining number equals 1000, turns P1500000,  
 if (r4009=71) then P0050 ; If the current special transfer function key is “G”, turns P0050 to execute

**【Explanation】**

- 1) The workpiece recorder can be read and written.
- 2) When the program has used r4009 instantaneous interference function, on the upper left corner of display screen is displayed “G” or “M” character or blank, indicating which program branch transfer function is used at present; “G” indicates r4009=71, “M” indicates r4009=77, the blank indicates r4009=0. Here, the “G” or “M” is not related to command M, and is only the sign of instantaneous interference function key. Assigning 71 or 77 to r4009 is equivalent to inputting “G” or “M” by keys, the branch transfer is effective as well.

- 3) The r4010 and r4011 cannot be assigned a negative value, and they are meaningless if assigned 0. Assignment range: 0~9999999 (i.e. 9999.999s). They can be used for read or conditional judgment. I.e. r1=r4010 or if(r4010>1000) then P0020.
- 4) In general, the r4005~r4008 is used for tool changing distinguishing. I.e. using (parameter P318=9) in M60 user-defined tool changing program

## 7.2 Statement

This system has assignment operation statement and conditional judgment statement, which are described as follows.

### 7.2.1 Assignment Statement

This system provides assignment operation statement that can realize the operation between two variables or values. For details about expression and operational symbols, see the table below.

Assignment operation statement table 9-1

Command format	Function	Definition
rN=rA	Assignment	Assigning the value of rA to rN variable
rN=rA + rB	Decimal addition operation	
rN=rA - rB	Decimal subtraction operation	
rN=rA * rB	Decimal multiplication operation	
rN=rA / rB	Decimal division operation	
rN=rA or rB	Binary or operation	
rN=rA and rB	Binary and operation	
rN=rA xor rB	Binary exclusive or operation	
rN=sqr (rA)	Decimal square root	$rN = \sqrt{rA}$ (rA does not support negative)
rN=abs(rA)	Decimal absolute value	$rN =  rA $
rN=rA % rB	Decimal remainder	rN = the remainder of (rA ÷ rB)
rN = rA*rB/rC	Decimal multiplication and division operation	
rN=sqrf (rA rB)	Compound square root	$rN = \sqrt{rA^2 + rB^2}$
rN=rA*sin(rB)	Sine	
rN=rA*cos(rB)	Cosine	
rN=rA*tan(rB)	Tangent	
rN=rA*atan(rB/ rC)	Arc tangent	

In the above table: rN indicates the variable name which the operation result is stored to; rA, rB, rC indicate the variables for running the operation, or constants.

#### 【Explanation】

- 1) None of the value, variable value and operation result contains decimal point, the unit is 0.001; For example: the assignment of Angle 45° should be rB=45000.
- 2) In the angle assignment operation statement, the rA cannot be omitted; otherwise, an alarm occurs; it can be written to rN=1\*sine(rB).
- 3) In the PROGRAM EDIT operation mode, all the special expressions can be obtained by pressing the hp6 key on the panel.
- 4) Each assignment statement can only have one operator (except the operation of multiplication and then division)



Incorrect example:  $r001 = r002 + r003 - r004$ .

The correct writing is :  $r001 = r002 + r003$

$r001 = r001 - r004$

**【Example】**

- $r3 = r2 * r41$  ; The result of r2 multiplied by r41 sends to r3
- $r2 = \text{sqr}(100)$  ; The data after extracting 100 sends to r2
- $r1 = 0$  ; The initialization r1 is angle 0°
- $r51 = 1000 * \sin(r1)$  ; The sine value of angle r1 multiplied by 1000 sends to r51
- $r1 = r1 + 1000$  ; The angle increases by 1°

7.2.2 Conditional Statement

The if conditional statement can form the branch structure. It makes judgment according to the given conditions, to decide which branch block is executed.

The basic format of if condition is:

if ( relationship expression) then statement 1 else statement 2

Its meaning is: if the expression is established, execute the statement 1 behind the then, otherwise execute the statement 2 behind the else.

**【Explanation】**

- 1) In the relation expression, the relation operators are  $<$ ,  $>$ ,  $=$  ; (less than, more than, equal to); e.g.  $rA < rB$ ,  $rA > rB$ ,  $rA = rB$ ;
- 2) The statement 1 and statement 2 can be one of transfer statement P, calling statement H, and assignment statement;
- 3) Behind the transfer symbol P and calling symbol H is the block line number;
- 4) The operator in assignment statement can be only +, - (addition, subtraction); e.g.  $rN = rC + rD$ ,  $rN = rC - rD$ .

**【The basic format and meaning of if statement:】**

**if (rA = rB) then P1 else P2 ; If rA=rB, executes P1, otherwise executes P2;**  
**if (rA = rB) then P1 else rN=rC + rD ; If rA=rB, executes P1, otherwise executes rN=rC+rD**  
**if (rA = rB) then rN=rC + rD else P2 ; If rA=rB, executes rN=rC+rD, otherwise executes P2;**

Thereinto:

rN: Variable name;

rA, rB, rC, rD: Variable name, or constant;

P1, P2: Transferred block line number ;

H1, H2: The called block line number(in the above format, the place transferred by P can also use call).

7.2.3 Statement Program Example

**【Program example 1】**

In the following program, it mainly demonstrates the method of automatically modifying tool compensation in the program. In the batch machining, suppose each time 20 pieces are machined, the tool T11 will fray one thread in the X direction regularly; when the workpiece recorder is the integer multiple of 20, the program will modify the tool compensation automatically.

```
N0010 G00 Z200 X100
.....
N1000 r1=r4001 % 20          ; Obtains the remainder of workpiece recorder divided by 20

N1010 if(r1=0) then P1030    ; If the remainder is zero, it is the integer multiple of 20 N1020
M20
N1030 r6201=r6201-10        ; No.1 X tool offset reduces one thread
N1040 M20                    ;Recycles
```

**【Program Example 2】**

In the following program, it mainly demonstrates the method of temporary press key interference and call in the program. Suppose a receiver is controlled by the pins of UO31 and UO32. In the batch cycle machining, if the operator needs to randomly sample the workpieces for test, press **G** key, then the receiver collects one sample automatically.

```
%104
N0010 G00 Z200 X100          ;
N0020 G00 Z180 X80           ;
N0030 G01 Z100 F200          ; Cuts outer circle
N1000 if (r4009=71) then (H2000) ; If "G" key is pressed before, then calls program P2000 to
                                execute
N1010 G01 X0 F200             ; Cut off
N1020 r2031=1                 ; Receiver returns to the original position
N1030 r2032=1                 ;
N1040 M20
N2000 r2031=0                 ; Receiver goes forward
N2010 r2032=0                 ;
N2010 r4009=0                 ; Clears "G" state
N2020 M99
```

**【Program example 3】**

In the following program, it mainly demonstrates the function operation. According to the elliptic parameter equation, figure out that the point on the ellipse and use the short line approaching method to move along the path of the ellipse.

The parameter equation of ellipse is  $X=50*\sin(a)$ ,  $Z=100*\cos(a)$ ; because X is diameter programming,  $X=100*\sin(a)$ .

```
%103
N0010 G00 Z0 X0
```

```

N0020 G00 Z200 X200
N0022 G00 Z190 X110      ; Determines the start point of G01 path
N0030 r1=0000            ; The initial value of r1 is 0°
N0040 r12=100000*sin(r1) ; The angle sine value is multiplied by 100000
N0050 r13=100000*cos(r1) ; The angle cosine value is multiplied by 100000
N0060 r12=r12-110000     ; Figures out the relative difference between ellipse path start
                          ; point and G01 start point

N0070 r13=r13-190000
N1000 r2=100000*sin(r1)  ; The angel sine value multiplied by 100000 sends to r2
N1010 r3=100000*cos(r1) ; The angel cosine value multiplied by 100000 sends to r3
N1020 r2 = r2 - r12
N1030 r3 = r3 - r13
N1040 G1 Xr2 Zr3 F500    ; Moves along the ellipse path
N1050 if(r1=360000) then P1080 ; Exits if the angle is not equal to 360°
N1060 r1=r1+1000        ; The value of r1 plus 1000 (angel plus 1)
N1070 M97 P1000         ; Cycles
N1080 M02

```

### 7.3 Process Monitoring and Execution

The common part program is executed according to the pre-designed path one by one, so quick response can not be performed in real time if some events occur during the command execution

In the system, there are ten monitors which are numbered from 0# to 9#, called 0# monitor, 1# monitor... 9# monitor. Their work process is called process monitor; in the same time of executing the common part program, they specially track the events which occur in real time in the part program execution process. However, the user must describe the task for the monitors, and tell them which event is to be tracked and how to deal with it after the event happens. The monitor without task is in sleep state.

Each monitor has a monitor register and a monitor management register, the corresponding sequence number is as follows:

Monitor number: 0#~9# sequence number;

Monitor register number: r7000~r7009 corresponding to the monitor sequence number;

Monitor management register number: r7010~r7019 corresponding to the monitor sequence number.

In the progress of machining, if there are progress monitors enabled, on the right side of the program status bar is displayed "M-n". Thereinto, M indicates the process monitor, and n indicates the number of enabled process monitors. The operator can also view the states of process monitors currently being used on the macro variable window.

E.g. M-2 indicates there are two process monitors enabled.

#### 7.3.1 Process Monitor Description (r7000)

To use the process monitor, first describe the process monitor according to the rules. The process monitor description includes an assignment statement and a if condition statement, neither of which can be omitted; the assignment statement specifies the monitor object, the if condition statement makes the judgment according to the obtained information by monitor, and decides whether to execute some branch block or not.

**【Description method of process monitor】**

## 1) Assignment description:

The so-called assignment description is to specify the monitor object with an assignment statement; all assignment statements which aim at monitor registers (r7000~r7009) are called monitor description statements.

E.g. r7000= r1001 or r1002; 0# monitor r7000 objects are interface input signal UI01, UI02  
r7001=r4003 ( rotation speed); 1# monitor r7001 comes from spindle real-time speed

## 2) Condition description

The so-called condition description is to specify how to make judgment according to the monitored information using an if statement;

The format of the condition description statement is the same as the basic format of common if statement, and the program rules are also basically the same.

if (relation expression) then statement 1 else statement 2;

Thereinto: In the condition description statement, the left item of if () relation expression is the monitor register name;

In the condition description statement, the call cannot be used in then, else, only the assignment or transfer can be used.

E.g. if (r7000=1) then P1500 ; if the input signal satisfies the condition, then turns p1500  
if (r7001>6000) then r3=r3+1 ; if the condition is satisfied, r3 adds 1

**【Explanation】**

- 1) The system specifies: all assignment statements that aim at r7000~r7009, are the description of monitor 0#~9#; according to the program execution path, the latest description is effective.
- 2) The operator of assignment description statement only can be one of "+", "-", "or", "and"; otherwise, an alarm occurs.
- 3) It is allowed to perform the assignment description using the conditional judgment statement, such as if (r5004=3) then r7000= r1001 + r1002, is a legal statement.
- 4) system specifies: if r7000~r7009 appears on the left of relational expression of if conditional statement, it is regarded as the conditional description to 0#~9# monitor. The call cannot be used in then, else, only the assignment or transfer can be used.
- 5) The assignment description and if conditional description must appear in pair; if there is no assignment description, or the variable which cannot be monitored is assigned, then r7000=0; if there is no if conditional description, it's allowed but the monitor function cannot be provided.
- 6) In the if condition description, the system does not execute the description statement immediately, but waits for monitor function of corresponding serial number to be enabled, then starts monitoring.
- 7) If the assignment meets the condition, it's not the conditional transfer behind, but the common variable assignment, such as if (r7000=100) then r3=r3+1, the monitor will not close.

**7.3.2 Process Monitor ON/OFF**

After describing the process monitor according to the rules, assign the monitor management register (r7010~r7019), then the monitor can be turned on or off. There are several working modes for turning on the monitor.

Assignment	Event process mode
0	Turn off the monitor which has been turned on, cause the monitor to enter the sleep state; keep its task, can open again and continue monitoring;
1	Turn on the monitor; in the monitor process, make the judgment according to the relation expression of the event. When the condition is established, the process mode is as follows: 1) If the monitor executes the assignment to common variable or output variable, it continues the monitor without affecting the execution of common program command; 2) If the monitor is to execute the special assignment, then turn off the monitor and stop the current motion command immediately, wait for the execution completion of all MST commands being executed, execute special assignment; 3) If the monitor needs to execute transfer, then turn off the monitor, stop the current motion command immediately, wait for the execution completion of all MST commands being executed, execute the transfer;
2	Turn on the monitor; in the monitor process, make the judgment according to the relation expression of the event. When the condition is established, the process mode is as follows: 1) If the monitor executes the assignment to common variable or output variable, it continues the monitor without affecting the execution of common program command; 2) If the monitor is to execute the special assignment, then turn off the monitor, wait for the execution completion of all MST commands being executed, execute special assignment; 3) If the monitor needs to execute transfer, then turn off the monitor, wait for the execution completion of all MST commands being executed, execute the transfer;
<b>Note: In execution, when meeting the M20/M02, turn off all the monitors, clear the task description; Special assignment is to assign values to the program control register r5008, system special variable group1 and group 2.</b>	

II PROGRAMMING

【Example】

- r7010= 0 ; If the system writes zero to r7010, then cancels the r7010 monitor;
- r7016= 1 ; Turns on r7016 monitor; if the condition is satisfied, stops the current motion command immediately, executes the assignment or transfer;
- r7012= 2 ; Turns on r7012 monitor; if the condition is satisfied, executes the assignment or transfer after executing the current motion command;

【Explanation】

- 1) Once the monitor is turned on, while executing the common part program, actually there are two description statements alternating and cycling in the system; until the monitor is turned off.
- 2) The if statement is allowed to turn on or off the process monitor; such as if (r1=0) then r7010=1.
- 3) It's allowed that one monitor turns on another one without turning off itself; such as if (r7001=0) then r7010=1, 0# monitor is turned off after 1# monitor meets the condition. A this time 1# monitor is still ON.
- 4) In programming, before turning on the monitor, the user must describe the monitor; if a monitor which is not described (without task) is turned on, the system prompts "E400: process monitoring variable undescribed".
- 5) After turning on a monitor, if its monitor description is changed without satisfying the condition or stopping the monitor, the old monitor description is replaced by the new one, and the monitor is still ON, so the restart is unnecessary. However, it is very dangerous in this state. The user should turn off the monitor first, and then change its monitor description, finally turn it on again.
- 6) After satisfying the monitor condition, if the command executed in monitor condition is jump command (Pxxxx), the monitor is turned off automatically; if the executed command is assignment or calculation command, the monitor is not turned off.

- 7) Monitor management register cannot be used as condition judgment; otherwise, the system alarm occurs; for example, if (r7010 > 1) then (r002 = 2), system issues “machining abnormal! E413: unknown process monitoring description”.
- 8) If a monitor is turned on, under the PAUSE/Block stop/Cycle stop state, the monitor is still effective; its assignment or calculation command are executed after meeting the condition, the skip command is executed by pressing CYCLE START key.
- 9) Once the monitor is turned on, inside the system, because its two description statements are alternating and cycling, its execution speed is faster than the common program; for the calculation statement which executes assignment after meeting the condition, it may have been executed many times, so the result of this variable is uncertain (using it in this way is not recommended). Please note that in the common part program, this variable must be used carefully.  
For example, if (r7000=100) then r3=r3+1 ; // if the condition is satisfied, r3 adds 1. At this time, note that the value of r3 is very uncertain (the accumulation frequency is very high); if the call of subprogram or transfer in the program uses r3, it will cause the uncertainty of the program execution. Therefore, in such a case, it is not suggested to use such uncertain variables like r3 as the actual function of each kind of command
- 10) In the common program, read the monitor register, monitor management register, it's value is zero or uncertain; such as r5= r7000, the value of r5 is uncertain.
- 11) If only one monitor is turned on, the signal change of more than 3ms width can be detected; if ten monitors are turned on, the signal change of 30ms width can be detected.

### 7.3.3 Monitor Program Example

#### 【Program example 1】

The following program introduces the use of process monitor.

In the batch machining, if the lengths of rough bars are not the same, it's hard to determine the machining start point. If a sensor is installed on the tool, then the system can use the sensor to determine the machining start point.

In the following program, if the system detection signal is connected to the corresponding pin of UI05; when the tool is far away from the workpiece, UI05="1"; when the tool is near the workpiece, UI05="0"; when the tool approaches the workpiece from a far-away position, it is stopped at the position of UI05="0", and the system determines this point to be the machining start point.

```

%105
N0010 r7000=r1005      ; 0# monitor detects the state of input signal r1005, and it forms the
                        ; monitor description together with the judgment statement below
N0020 if(r7000=0) then P1000 ; If the signal r1005 is monitored to be "Zero", then turns to
                        ; P1000 program running
N0030 G00 Z300 X200   ; First positioning to the initial point
N0040 G00 X180       ;
N0050 r7010=1        ; Turns on 0# monitor by mode 1, if the condition is satisfied, stops the

```

```

motion immediately, and turns to P1000
N0060 G00 W-100 F1000 ; The tool approaches the workpiece from a far-away position
N0070 r7010=0 ; If the last command is finished, but nothing is checked, then turns
off the 0# monitor
N0080 G00 Z300 X200 ; Returns to the initial point
N0080 M02
N1000 G50 Z200 X180 ; Sets this point to be the machining start point, resets the workpiece
coordinate system
N1020 G01 Z90 F500 ; Cuts outer circle
N1030 G01 X0 ; Cut off
N1040 G51 ; Recovers the workpiece coordinate system
N1050 G00 Z300 X200 ; Returns to the initial point
N1060 M02 ;

```

### 【Program Example 2】

The following program introduces the use of process monitor.

The system has the external cycle start/pause function, so the operator can realize the start/pause function with the external buttons from a position which is far away from the system. However, in the position which is far away the system, it is hard to realize the “single block stop”. In the following program, this function is realized through the process monitor.

In the program, it is supposed that the button switch signal is connected to the pin corresponding to UI05; when pressing the switch, UI05=“0”, when releasing the switch, UI05=“1”. After turning on the monitor, this press-button can switch between single block mode and continuous execution mode.

```

%106
N0030 r7000 =r1005 ; 0# monitor detects the state of input signal r1005
N0040 if(r7000 =0) then r5008=2 ; When the signal r1005 is monitored to be “zero”, then
pauses
N0030 r7001 =r1005 ; 1# monitor detects the state of input signal r1005
N0040 if(r7001 =1) then r7010=2 ; When the press-button is released, then 0# monitor is
turned on again
N0070 r7011=2 ; Turns on 1# monitor by mode 2
N0070 r7010=2 ; Turns on 0# monitor by mode 2
N0010 G0 X100 Z200
N0020 G00 X50 Z190
N0050 G1 X50 Z160 F300
N0060 G2 W-80 R100
N0080 G1 U10 W-20
N0080 G1 W-20
N0080 G1 U10 W-20
N0090 G0 X100 Z200
N0160 M02

```

### 【Program Example 3】

The following program introduces the use of process monitor.

In the following program, it is supposed that the detection signal is connected to the corresponding pins of UI05 and UI06, and the output signal is UO05; in the machining process, when both UI05 and UI06 are “0”, the system is required to output “0” from UO05 pin, and output “1” in other instances.

```

%107
N0010 r7000=r1005 or r1006 ; 0# monitor detects the state of input signal r1005, r1006
N0020 if(r7000=0) then r2005=0 ; If the two signals r1005, r1006 are "zero" at the same time,
    UO05 outputs "zero";
N0030 r7010=1 ; Turns on 0# monitor by mode 1, if the condition is satisfied,
    UO05 outputs "zero"

N1000 G00 Z300 X200 ; Positioning to the initial point
N1010 G00 X180 ;
N1020 G01 Z90 F500 ; Cuts outer circle
N1030 G01 X0 ; Cut off
N1050 G00 Z300 X200 ; Returns to the initial point
N1060 M97 P1000 ;
N1070 M02 ;

```

### 7.3.4 Pulse Monitoring (r7100)

In this system, there are another four pulse monitor variables numbered with 0#~3#. They are specially used for the input signal pulse count. Each pulse monitor variable has a monitoring register and a management register. The corresponding sequence numbers are as following:

Pulse monitoring variable number: 0#~3# sequence number;

Pulse monitoring register number: r7100~r7103 correspond to the monitor variable sequence numbers;

Pulse monitoring management register number: r7110~r7113 correspond to the monitor variable sequence numbers

The so-called pulse monitor belongs to process monitor too, and its description method, ON/OFF method and the rules are the same as those of process monitor.

In the machining process, if there are monitors enabled, on the right of the program status bar is displayed "P-n". The user can also view the state of pulse monitor currently being used in the system on the macro variable window. For example, P-1, indicates one pulse monitoring is enabled.

#### ◆The use of pulse monitor:

- 1) First describe the pulse monitoring variable; (all assignment statements for r7100 are description)

Example: r7100 = r1001; r7100 adds 1 to the pulse signal of interface UI01

- 2) Then describe the pulse monitoring condition;

Example: if (r7100>50) then P1500 ; If the r7100 count value is greater than 50, then turns to P1500

- 3) And then start the monitor through the assignment of pulse monitoring manager;

Through the assignment of pulse monitoring management register(r7110~r7113) , the monitor can be turned on or off.



Thereinto: When 0, 1, 2 is assigned, it is the same as the process monitor.

When 5 is assigned, the pulse monitor counter is cleared.

When 6 or 7 is assigned, the pulse monitor count mode is set.

Assignment	Event process mode
0	Turn off the monitor which has been turned on, which is the same as that of the process monitor
1	Turn on the monitor; the process mode is the same as that of the process monitor
2	Turn on the monitor; the process mode is the same as that of the process monitor
5	When the 5 is assigned, the pulse monitor counter is cleared.
6	Set the counter mode: when the operation result is changed from zero to non zero, the inner counter increases by 1.
7	Set the counting mode: when the operation result is changed from non-zero to zero, the inner counter increases by 1.

**◆How is the pulse monitor counted;**

- 1) After the pulse monitor is turned on, it executes the assignment description statement repeatedly, when the last operation result compares with the operation result of this time, it matches the following skip, the inner counter increases by 1.  
 If set to 6, when the last operation result is 0, and this operation result is non- zero, the inner counter increases by 1.  
 If set to 7, when the last operation result is non-zero, and this operation result is zero, the inner counter increases by 1.
- 2) In the initialization when the pulse monitoring is turned on, the counter is automatically cleared.
- 3) If only one pulse monitor is turned on, the system can check the signal change of more than 3ms width; if four pulse monitors are turned on, the system can check the signal change with more than 12ms width.

7.3.5 Pulse Monitoring Programming Example

**【Program example】**

The following program introduces the use of pulse monitor.

In the following program, it is supposed that the detection signal is connected to the pin corresponding to U105; in the machining, when the number of times of U105 changing from “0” to “1” is greater than 10, the program ends.

```

%108
N0010 r7100 = r1005          ; r1005 pulse signal plus 1
N0020 if(r7100>10) then P2000; If the number of monitor pulses is greater than 10, then turns
                                to P2000 running
N0030 r1=0                    ;
N0040 r7110=6                 ; Sets the count mode, when it is changed from zero to
                                non-zero, the inner counter plus 1.
    
```

```

N0050 r7110=2          ; Chooses the monitor working mode 2 to turn on
N1000 G00 Z300 X200    ; Positioning to the initial point
N1010 G00 X180         ;
N1020 G01 Z90 F500     ; Cuts outer circle
N1030 G01 X0           ; Cuts off
N1050 G00 Z300 X200    ; Returns to the initial point
N1060 M97 P1000       ;
N2000 M02              ;

```

### 7.3.6 Variable Transfer Register (r7900)

Variable transfer register: r7900; (the system forbid using this variable in the process monitor description statement)

#### 【Use example】

```
r7900= 1 ;
```

Writing any number to this register means all the common variables in monitor backup area are copied to common variable storage area. The following will explain this register function.

In general, in order to avoid the collision, if a certain common variable is used in the monitor, the user should avoided using this variable in the common block. Because the system has a common variable storage area, and a monitor variable backup area, the variable value change in these two sections may not be synchronous.

The change of common variable r3 is explained as follows:

- 1) If assignment statement of r3 is executed in a common program, r3 is modified, then the r3 in the monitor will be modified immediately as well;
- 2) If the r3 assignment statement is executed in the monitor, it only modifies the r3 in the monitor backup area; and the r3 in the common program is not modified.
- 3) After the variable transfer statement (i.e.r7900=1) is executed, the system will copy all common variables in the monitor backup area to the common variable storage area; at this moment, the r3 in the common program is the same as that in the monitor.

#### 【Program Example】

In the following program, it is supposed that the detection signal is connected to the pin corresponding to U105; in the machining, when the times of U105 changing from “0” to “1” is greater than 10, the program ends after the cycle execution is finished.

```

%109
N0010 r7100 = r1005      ; The pulse signal of r1005 plus 1
N0020 if(r7100>10) then r1=1 ; If the number of pulses is greater than 10, r1 equals 1, and it is
                           ; the r1 in the monitor backup area that is actually modified
N0030 r1=0              ; Sets the initial value
N0040 r7110=6          ; Sets the count mode, when changing from zero to non-zero, the
                           ; inner counter plus one.
N0050 r7110=2          ; Chooses the monitor working mode 2 to turn on

```

```

N1000 G00 Z300 X200      ; Positioning to the initial point
N1010 G00 X180          ;
N1020 G01 Z90 F500      ; Cuts outer circle
N1030 G01 X0            ; Cuts off
N1050 G00 Z300 X200      ; Returns to the initial point
N0060 r7900=1          ; Read the common variable value in the monitor backup area
N0020 if(r1=1) then P2000 ; If r1 equals 1, then turns to P2000
N1060 M97 P1000        ;
N2000 M02              ;
    
```

## 7.4 Attached List

### 7.4.1 ASCII List

character	ASCII code	character	ASCII code	character	ASCII code	character	ASCII code	character	ASCII code	character	ASCII code
Space	32	0	48	@	64	P	80	`	96	p	112
!	33	1	49	A	65	Q	81	a	97	q	113
"	34	2	50	B	66	R	82	b	98	r	114
#	35	3	51	C	67	S	83	c	99	s	115
\$	36	4	52	D	68	T	84	d	100	t	116
%	37	5	53	E	69	U	85	e	101	u	117
&	38	6	54	F	70	V	86	f	102	v	118
,	39	7	55	G	71	W	87	g	103	w	119
(	40	8	56	H	72	X	88	h	104	x	120
)	41	9	57	I	73	Y	89	i	105	y	121
*	42	:	58	J	74	Z	90	j	106	z	122
+	43	;	59	K	75	[	91	k	107	{	123
,	44	<	60	L	76	\	92	l	108		124
-	45	=	61	M	77	]	93	m	109	}	125
.	46	>	62	N	78	^	94	n	110	~	126
/	47	?	63	O	79	-	95	o	111		

### 7.4.2 Corresponding List between Common Colors and Code Values

Name	Code	Name	Code	Name	Code	Name	Code	Name	Code	Name	Code
Black	0	Cyan	41	Purple	85	Red	168	Yellow	245	Grey	251
Blue	6	Light blue	54	Light green	105	Pink	172	Golden	248	Hoar	253
Green	35	Green yellow	80	Brown	126	Orange	231	Silver white	250	White	255

## CHAPTER EIGHT CUSTOMIZED COMMAND PROGRAMMING

## 8.1 Customized Command

The system has a group of customized commands except the standard M commands. The machine manufacturer can set some commands to realize the control of additional equipment, and the operator can conveniently operate the additional equipment by inputting this command.

The system provides 15 customized commands from M60 to M74, and the number of the commands is edited by the machine manufacturer. The customized commands must be edited in %254 program. First debug %245 program successfully with single block in AUTO operation mode, then solidify the program in EDIT operation mode; after the program is solidified successfully, the solidified customized command can be used in JOG mode or other programs (%000~%253); otherwise, the system issues an alarm prompting no customized command.

## 8.1.1 Programming Format of Customized Command

In No. %254 program, the programming format of the system customized command is: input a group blocks which starts with start sign“-M60”and ends with “M99”. This group of blocks constitutes M60 command. Other blocks which starts with “-M61 ~ -M74” and ends with “M99” constitute M61~M74 command.

## 【Format】

N1000	-M61			; M61command start sign
N1010	M8	M3		; Cooling ON, Spindle ON
N1020	G4	D3		; Timing delay
N1030	M99			; M61 end

## 【Explanation】

- 1) If G command is required to add in the customized command, set the position bit parameter **P404\_d2** to 1, i.e. G motion command is allowed in M61 command; otherwise, the system issues an alarm

## 8.2 Customized Command Library (P254)

The customized command is solidified in No. %254 program, so No. %254 program is also called the user-defined command library: P254 mainly includes the M60~M74 commands which are formed by statements and blocks, and the system will perform special compiling to P254 during compiling.

### 8.2.1 Programming Format and Debugging of Customized Command Library

**【Format】**

P254 program format:

```

N0010 M98 P1000 ; Starts executing No.%254 program
N0020 M98 P2000 ; Calls M61 command
N0030 M98 P3000 ; Calls M62 command
N0040 M02 ;

N1000 -M60 ; The start of M60
...
N1100 M99 ; The end of M60

N2000 -M61 ; The start of M61
... ; Contents of customized M61
N2100 M99 ; The end of M61

N3000 -M62 ; The start of M62
... ; Contents of customized M62
N3100 M99 ; The end of M62
    
```

The debugging steps of customized command library are as follows:

- 1) Edit the user desired customized command according to the above programming format of customized command library;
- 2) Debug the program by single block mode in AUTO operation mode;
- 3) After debugging the program, solidify the program in EDIT mode, by pressing keys: **hp5 5** ; the machine manufacturer determines whether the solidified program can be modified and then be solidified again, i.e. set the position parameter **P404\_d0=1**.
- 4) After the system prompts that the compiling and solidification of P254 succeeds, press “ESC” key.

**【Explanation】**

After the compiling and solidification of No.%254 program succeeds, the user can also check whether P254 is correct by inputting and then executing the customized command in P254 in JOG operation mode.

### 8.2.2 Use of Customized Command Library

In the JOG operation mode, input the customized command directly when it is necessary to execute the solidified customized command in P254. In editing program, the solidified customized command in P254 can be used. Thus the operation is simple and convenient.

**【Format of using customized command】**

```

%001 ; Executes the customized command in the programs except No.%254
program
N0010 M61 ; Executes M61 command
N0020 G04 D3 ;
N0030 M62 ; Executes M62 command
    
```

```

N0040 G04 D3 ;
...
...
N0050 M02 ;

```

**【Explanation】**

- 1) The M60~M74 can be directly used in JOG operation mode or other programs only after the No.% 254 program has been solidified to FLASH; otherwise, the system issues an alarm; if the corresponding customized command is not written in No. % 254 program, but this command is used in JOG operation mode or other programs, the system also issues an alarm.
- 2) For the operation of how to solidify the No.%254 program to FLASH, please refer to the explanation of the No.%254 program. For details, refer to Section 4.3 Edit Operation Mode in Part I Operation
- 3) After the No. %254 program is modified, it must be solidified to FLASH again; otherwise, the M60~M74 command called in JOG operation mode or other programs are not the latest customized command after modification.
- 4) M60~M74 command must be used in an independent block; if not, an alarm occurs.

**【Note】**

- 1) When the customized command is executed in AUTO mode, by pressing **CYCLE PAUSE** key, the system will not stop the blocks in the customized command as the normal pause function, but it will continue the execution after a short pause.
- 2) When the customized command is executed in AUTO mode, if SINGLE BLOCK is valid, SINGLE BLOCK function is valid only for the whole customized command rather than the blocks in the command.
- 3) When the customized command is executed in MANUAL mode, by pressing **CYCLE PAUSE** key, the system will perform pause function to the blocks in this command.

## 8.3. Foot Switch of M61 command

**◆ Foot switch function of M61/ M62, M63/ M64, M65/ M66 commands:**

When you need to use the foot switch to control the reverse motion commands in pair such as M10/M11, M79/M78, you can use M61/M62. For example, program M61 as feeding-in, and M62 as feeding-back, then set the corresponding I/O name of the input pin of the foot switch in parameter P533, and connect them correctly. Then, M61/M62 can be executed alternately by pedaling the switch.

All of M61/62, M63/64, M65/66 can realize the pedal control function. You can set the corresponding I/O name of the input pin of the foot switch in parameter P533~P535 respectively, and set the value to the sequence number of the common signal name. Note that the corresponding M61~M66 must be solidified.

**◆ Key control M61~M66:**

When the system input interfaces are not enough, set parameters P533~P535 to 99; in AUTO or

JOG mode, you can press keys to control the execution of customized commands M61~M66.

When P533 is 99, key 1 corresponds to M61, key 2 to M62; when P534 is 99, key 3 to M63, key 4 to M64; when P535 is 99, key 5 to M65, key 6 to M66.

**【Explanation】**

- 1) In MANUAL/AUTO mode, key control M61~M66 function can only be used when the system is free or no program is executed. It cannot be used in SINGLE stop mode and when a window pops up.
- 2) Upon the system power-on, M61/M63/M65 is always executed when the foot switch is used for the first time.
- 3) In the process of executing M61/M63/M65, if the execution is interrupted by RESET operation, the system will execute M62/M64/M66 by pressing down the foot switch again.

## III CONNECTION





# CONNECTION

## CHAPTER ONE INTERFACE OVERVIEW

### 1.1 Rear Cover Interface Layout

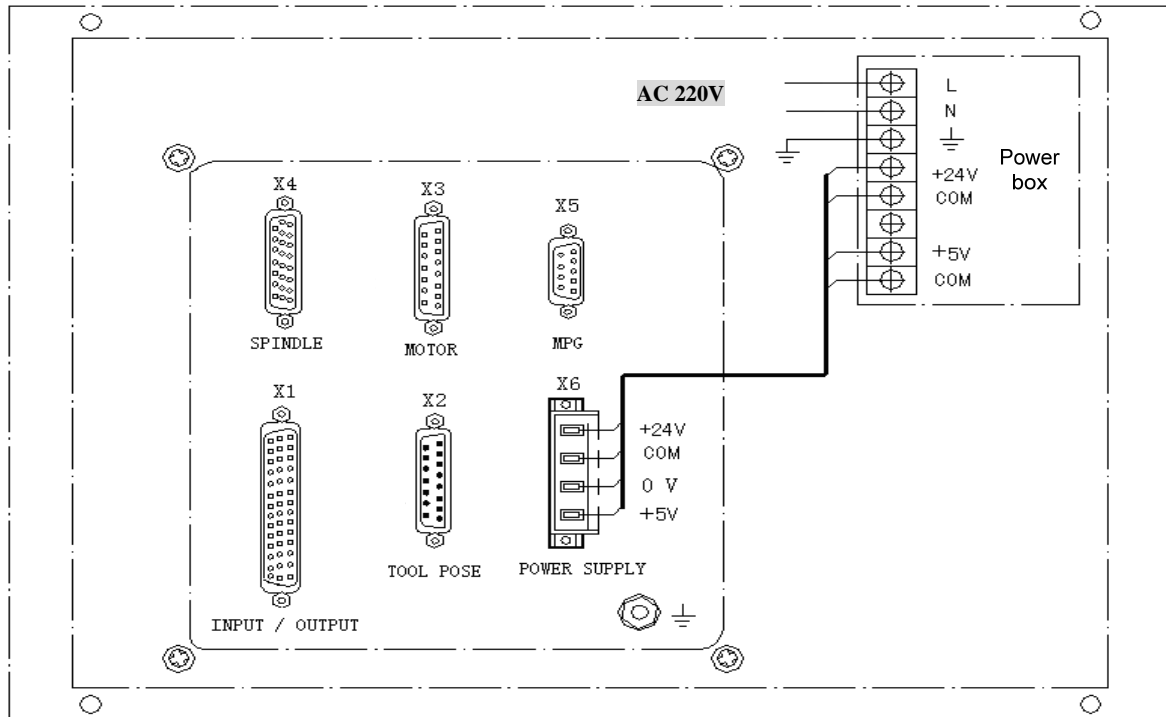


Fig. 1-1 Rear Cover Interface Layout

#### 【Explanation】

- X1: Input/output, 44-pin D-type female socket, interface through which CNC receives machine signals or CNC signals are output to the machine
- X2: Tool pose, 15-pin D-type male socket, which is the input/output interface
- X3: Motor, 5-pin D-type female socket, which is the interface of X/Z drive unit
- X4: Spindle, 26-pin D-type female socket, which is the interface for spindle coder, spindle frequency converter and Y drive unit
- X5: MPG, 9-pin D-type female socket, which connects with MPG
- X6: Input power supply, green 4-pin female socket. The connection between the power box and the system X6 interface has been performed, the user only needs to connect 220V power supply.
- Power supply box: adopts GSK-PB or GSK-PC2 power supply (Note that their connecting terminals are different); Fig. 1-1 shows GSK-PB power supply, which provides GND, +5V, +24V power.

1.2 Overall Frame

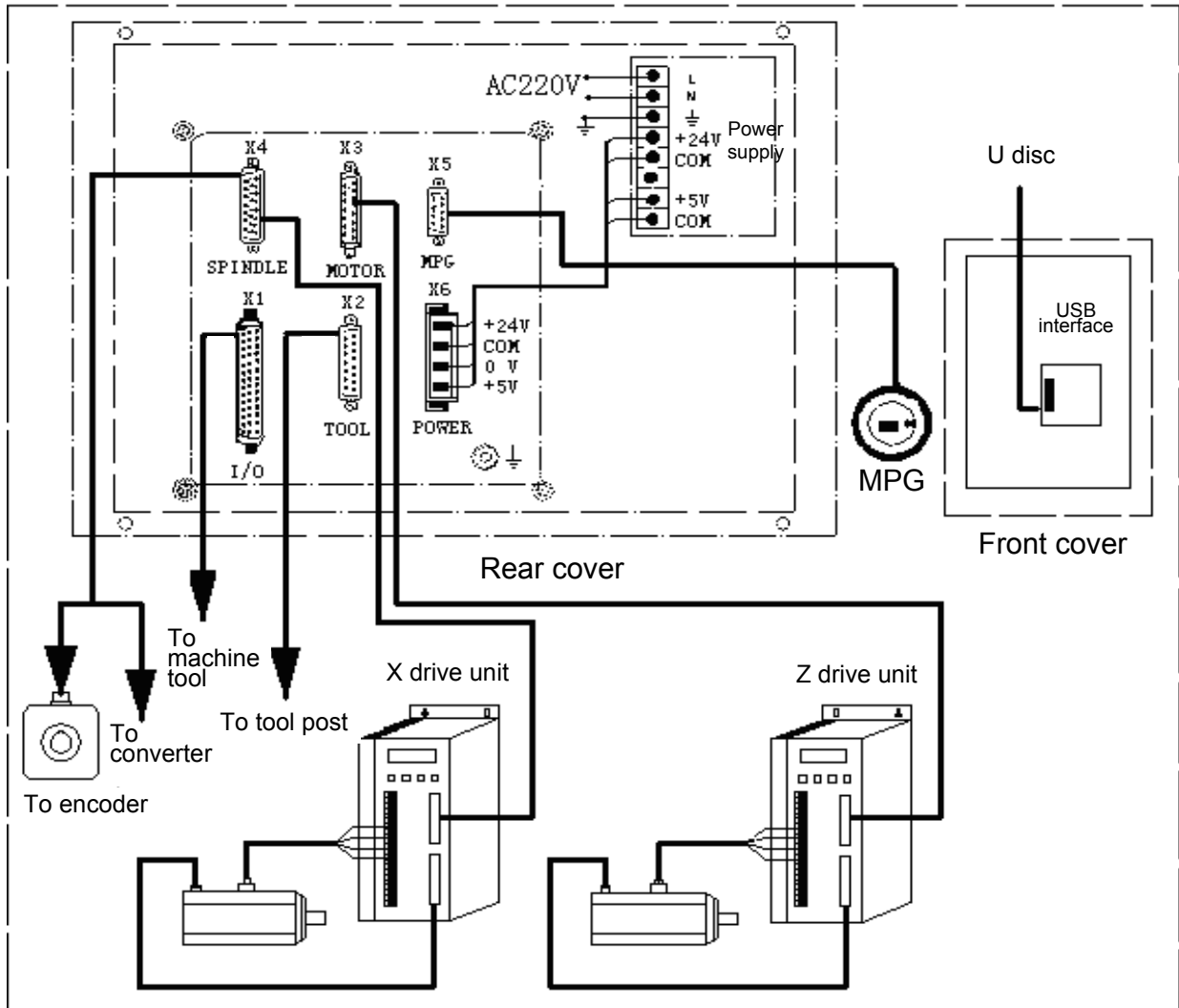


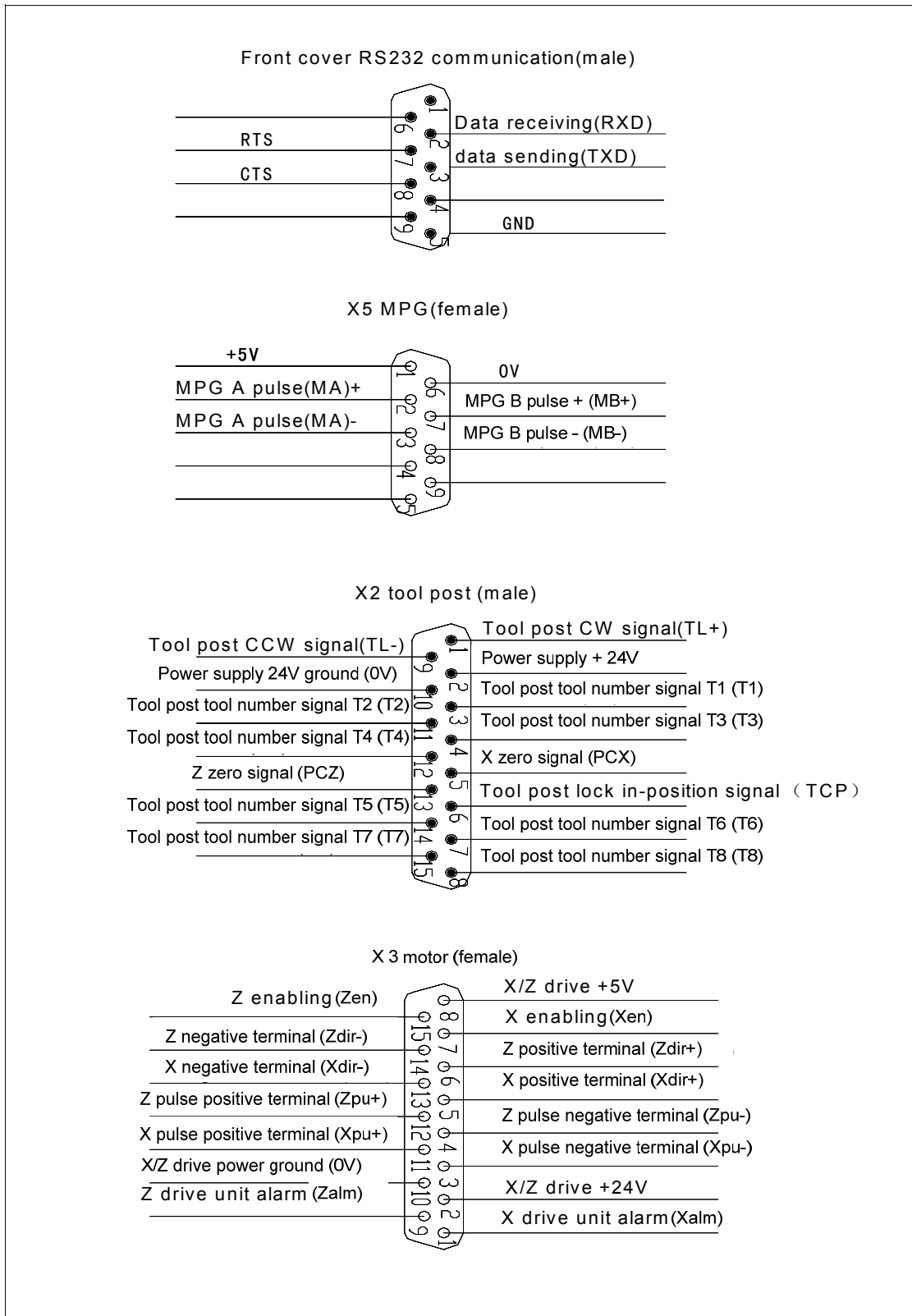
Fig. 1-2 Overall connection layout

**【Note】**

GSK928TD has no Y axis function, so it is unnecessary to connect the interfaces relative to Y axis.

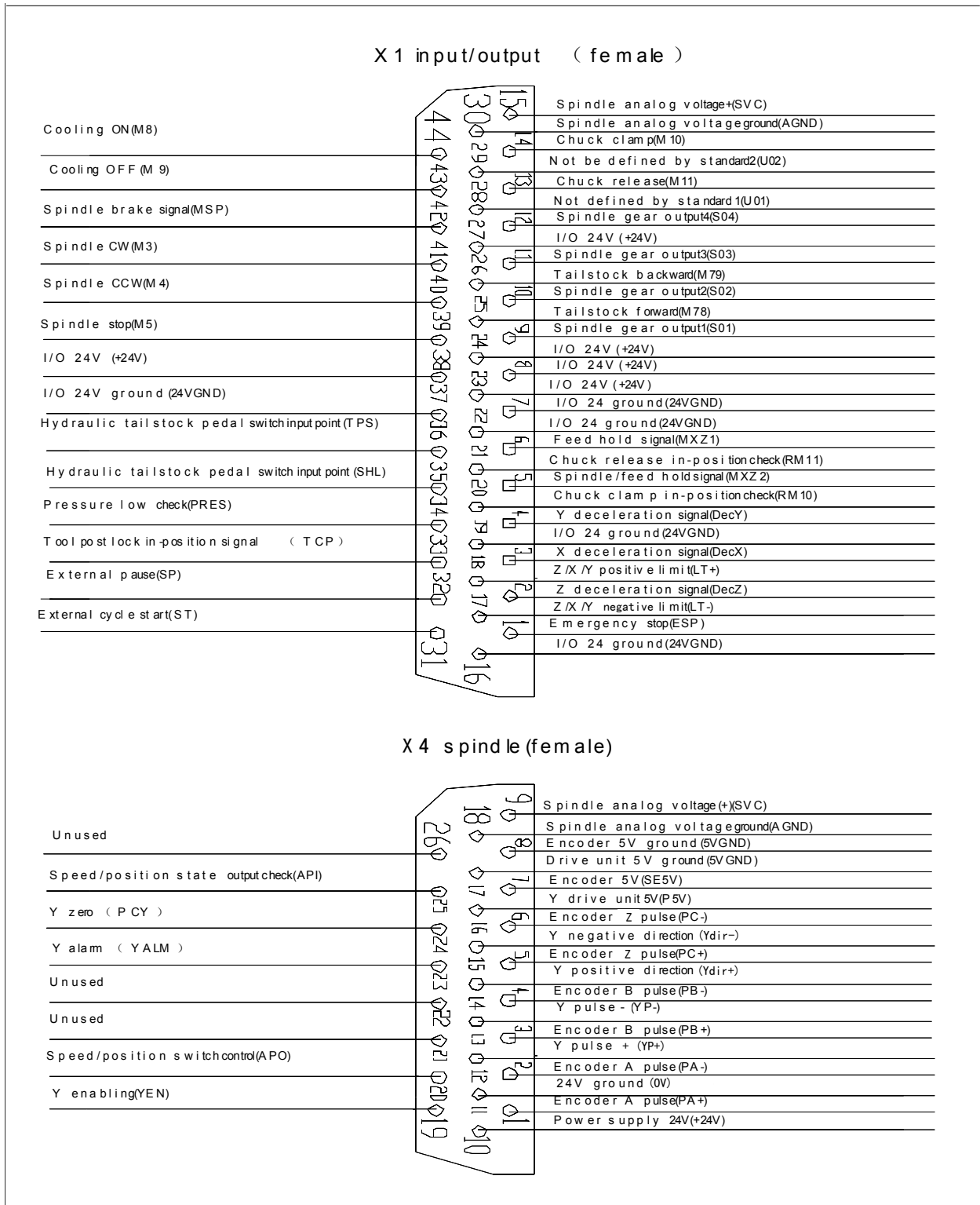
CHAPTER TWO INTERFACE TABLE

2.1 Interface Table I



III CONNECTION

2.2 Interface Table II



## CHAPTER THREE CNC DEVICE CONNECTION

### 3.1 Communication Interface

The CNC system can connect to the external PC (RS232 interface) or another CNC system through the serial port to realize the data exchange or transmission; it can also realize the data exchange or transmission with an external disk through the USB interface.

#### 【Explanation】

- 1) Serial communication interface: RS232 communication, for connecting with PC RS232 interface.
- 2) USB interface: for connecting with U disk

**Note:** The RS232 interface of 928TD system has not been led out; the interface needs to be led out with special tools and be performed by professional technicians.

#### 3.1.1 USB Interface

USB interface is used to connect the CNC system and U disk. The system not only sends the data to the U disk, but also receives the data from the U disk via the USB interface. Directly insert the U disk into the USB interface on the system panel, and the system automatically identifies and opens the content of the U disk when the U disk creates the file and file name in the root directory according to the system file requirements.

#### 【Note】

- 1) The system USB interface must be covered if not used. Do not expose it in the atmosphere for long time; otherwise, the surface metal will be oxidized and interface sensitivity may be reduced.
- 2) Do not leave the U disk unused in the system USB interface for a long time; otherwise, the system interface will be aged, thus damaging the U disc.
- 3) After using the U disc, be sure to turn off the U disc by pressing **ESC** key before unplugging it. When the USB icon displayed on the screen disappears, you can pull the U disc out from the system USB interface. Otherwise, the system hardware and U disc may be damaged.

### 3.2 X1, X2 Interface

The CNC system X1, X2 interfaces have 23-channel digital input and 18-channel digital output, all of which use the photoelectric isolation.

#### 3.2.1 X1, X2 Interface Signal Definition

Important explain:

- 1) The 23-channel digital input signal and 18-channel digital output signal correspond to 23 input pins and 18 output pins. Each pin has a “pin number” and two “signal names”, i.e. special signal name and common signal name”. Input common signal names are numbered with UI01~UI32, output common signal names are numbered with UO01~UO32; thereinto, UI17~UI25, UO19~UO32 are not led out (unusable; displayed in grey in DIAGNOSIS mode). If a signal is not specified, it is displayed in white UI\*\* or UO\*\*, representing it can be used

otherwise.

- 2) This system has already set control and detection pin for most of the devices. These pins are called “special signal name”, for example, spindle, chuck, tailstock, tool post etc. As the table below, set parameters correctly before using these functions. In DIAGNOSIS mode, the special signal names are displayed in yellow, representing it cannot be used otherwise.
- 3) During system parameter initialization, some commands (such as M91, M21, G31) have been set pins i.e. the interface parameters P500 (M210, P501, (M230), P517 (M91I), P518 (M93I), P532 (G31I) have been defined; their signal names are displayed in green in DIAGNOSIS mode; the machine tool builder can perform connection according to these interface parameters, and the system will detect the corresponding signals on pins.
- 4) For some other devices (see Section 4.4.9 Other Option Functions in Part I Operation, like safety door detection, triple-color indicator, etc.), there is no pin set for their functions. They can be set to any pin by the machine tool builder with this method: Fill the sequence number of unused “common signal name” in the “interface parameter” in PARAMETER mode. The newly added signals in “interface parameters” are displayed in green in DIAGNOSIS mode. For example: Inputting signal UI05 is invalid for a 4-position tool post. Set parameter P511 to 5, then it turns to be the detection signal of safety door (SAGT). The system detects UI05 signal (pin 14 on X2 interface) during safety door detection.
- 5) Both “Pin 15 on X1 interface” and “pin 9 on X4 interface” are “spindle analog voltage +” signal (SVC). Their functions are the same.
- 6) Both “Pin 30 on X1 interface” and “pin 18 on X4 interface” are “spindle analog voltage ground” signal (AGND). Their functions are the same.
- 7) Both “Pin 33 on X1 interface” and “pin 6 on X2 interface” are “tool post clamping in-position signal (TCP)” . Their functions are the same.

The definitions of 23-channel digital input signal and 18-channel digital output signal are as follows:

**Table 3-1**

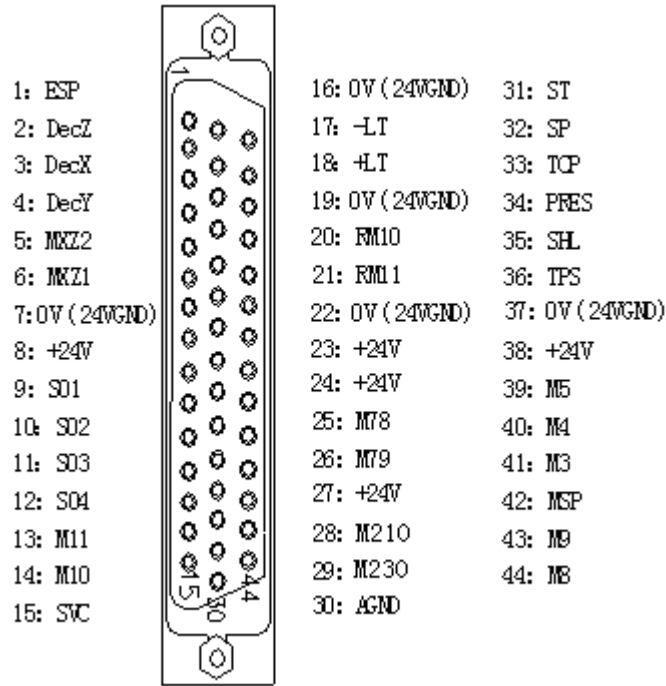
Interface	Pin No.	Special signal name	Common signal name	Variable	Function explanation of special signal name
X1 interface	1	ESP			Emergency stop (emergency stop signal)
	2	DecZ	UI28	r1028	Z axis deceleration signal
	3	DecX	UI29	r1029	X axis deceleration signal
	4	DecY	UI30	r1030	Y axis deceleration signal
	5	MXZ2	UI15	r1015	External feed/spindle hold signal
	6	MXZ1	UI16	r1016	Feed hold signal
	7	0V (24VGND)			Power supply 24V ground
	8	+24V			Power supply 24V
	9	S01	UO01	r2001	Spindle gear output 1
	10	S02	UO02	r2002	Spindle gear output 2
	11	S03	UO03	r2003	Spindle gear output 3
	12	S04	UO04	r2004	Spindle gear output 4
	13	M11	UO13	r2013	Chuck releasing
	14	M10	UO14	r2014	Chuck clamping
	15	SVC			Spindle analog voltage +
	16	0V (24VGND)			Power supply 24V ground
	17	-LT	UI31	r1031	Z/X negative limit
	18	+LT	UI32	r1032	Z/X positive limit
	19	0V (24VGND)			Power supply 24V ground

20	RM10	UI14	r1014	Chuck clamping in-position check
21	RM11	UI13	r1013	Chuck releasing in-position check
22	0V (24VGND)			Power supply 24V ground
23	+24V			Power supply 24V
24	+24V			Power supply 24V
25	M78	UO15	r2015	Tailstock advancing
26	M79	UO16	r2016	Tailstock retracting
27	+24V			Power supply 24V
28	M210	UO17	r2017	P500 set as customized command output
29	M230	UO18	r2018	P501 set as customized command output
30	AGND			Spindle analog voltage ground
31	ST	UI26	r1026	External cycle start
32	SP	UI27	r1027	External pause
33	TCP	UI09	r1009	Tool post clamping in-position signal
34	PRES	UI10	r1010	Pressure LOW check
35	SHL	UI11	r1011	Hydraulic chuck foot switch input
36	TPS	UI12	r1012	Hydraulic tailstock foot switch input
37	0V (24VGND)			Power supply 24V ground
38	+24V			Power supply 24V
39	M5	UO05	r2005	Spindle stop
40	M4	UO06	r2006	Spindle CCW
41	M3	UO07	r2007	Spindle CW
42	MSP	UO08	r2008	Spindle brake signal
43	M9	UO09	r2009	Cooling OFF
44	M8	UO10	r2010	Cooling ON

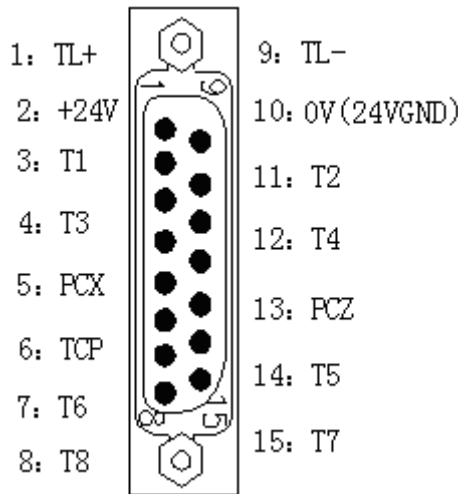
Table 3-2

Interface	Pin No.	Special signal name	Common signal name	Variable name	Function explanation of special signal name
X2 interface	1	TL+	UO12	r2012	Tool post (CW) output signal
	2	+24V			Power supply 24V
	3	T1	UI01	r1001	Tool post tool position signal T1
	4	T3	UI03	r1003	Tool post tool position signal T3
	5	PCX			X axis zero signal
	6	TCP	UI09	r1009	Tool post clamping in-position signal
	7	T6	UI06	r1006	Tool post tool position signal T6
	8	T8	UI08	r1008	Tool post tool position signal T8
	9	TL-	UO11	r2011	Tool post CCW output signal
	10	0V (24VGND)			Power supply 24V ground
	11	T2	UI02	r1002	Tool post tool position signal T2
	12	T4	UI04	r1004	Tool post tool position signal T4
	13	PCZ			Z axis zero signal
	14	T5	UI05	r1005	Tool post tool position signal T5
	15	T7	UI07	r1007	Tool post tool position signal T7





X1: Input/output, 44-pin D-type female socket



X2: Input/output, 15-pin D-type male socket

### 3.2.2 Connection Method of Input Signal

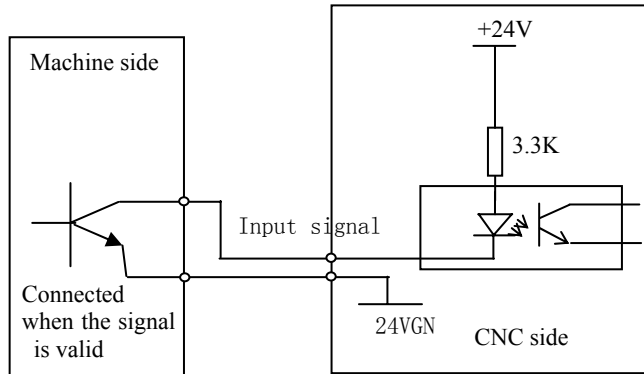
The input signal includes UI01~UI16 ,UI26~UI32 and zero signal (PCX, PCZ, PCY). CNC checks the machine states by the input signal; signal direction: machine to CNC. The input signal is valid in the low level (24VGND).

There are two types of external input of the input signal: one uses the machine contact switch input, by which the signal is from the machine side press key, the limit switch and relay contact; the other uses the electronic proximity switch (transistor) input without the contact.

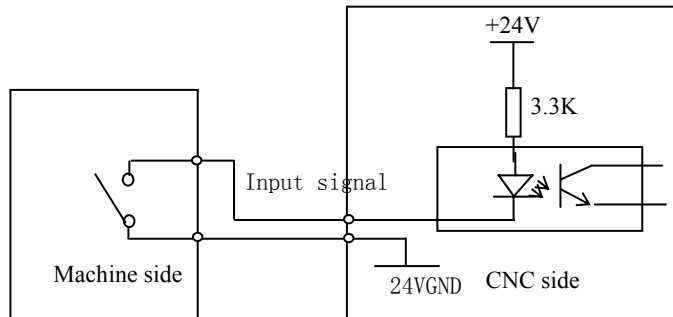
- 1) The input signal can use the normally-open contact input of the machine contact, or use the electronic proximity switch input without the contact (low level is output in approaching). The

interface connection method is as follows:

NPN connection method without contact:



Machine contact connection method:



The contact of the input signal at the machine side should meet the following conditions:

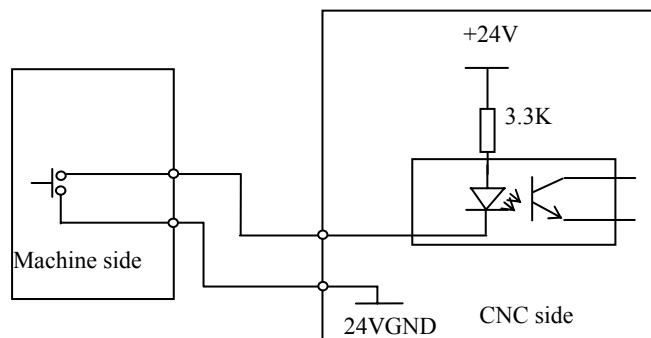
Contact capacity: more than DC28V, 16mA

Leakage current between the contacts in open circuit: below 1mA.

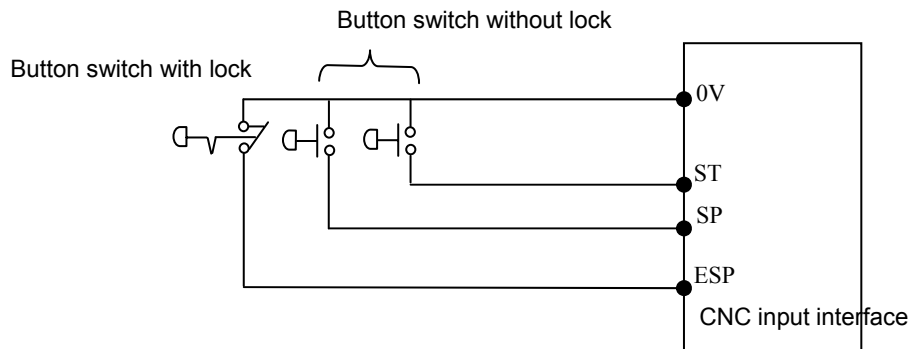
Voltage drop between contacts in close-circuit: below 1V (current 8.5mA, including the voltage drop of the cable).

- 2) The input signal UI26, UI27 (SP, ST are special signals) use the normally-open contact of the machine contact switch; ESP signal uses the normally-closed contact of the machine contact switch with self-lock.

Circuit diagram:



Connection example:



### 3) Special signal name definition of input interface

SP: External pause operation key signal.

ST: External cycle start key signal.

ESP: Emergency stop key signal (the signal must be connected) .

SHL(chuck foot switch): Foot switch input signal of hydraulic chuck.

TPS(tailstock foot switch): Foot switch input signal of hydraulic tailstock.

PRES: Pressure low check, the hydraulic system pressure low check input signal

DecX, DecZ, DecY: X, Z, Y axis machine reference point return deceleration signal.

PCX, PCZ, PCY: X, Z, Y axis machine reference point return zero signal

LT+,LT-: Z/X positive/negative limit switch signals. Connect X, Z positive/negative limit signal to the signal.

Note: For the connection of LT+, LT- , see Section 4.1 Safety Protection in Part I Operation.

MXZ1: Feed hold signal. When it is valid (i.e. switch contact ON), the feed is locked (i.e. X, Z stop).

MXZ2: External feed/spindle hold signal. When it is valid (i.e. switch contact ON), the spindle, X, Z are locked at the same time (i.e. spindle, X, Z stop). Note: For the control of MXZ1, MXZ2, see Auto Operation Mode in Part I Operation.

RM10: Chuck clamping( outer chuck)/releasing(inner chuck) in-position detection.

RM11: Chuck releasing( outer chuck)/clamping(inner chuck) in-position detection.

Note: For the control of the hydraulic chuck and tailstock, see JOG Operation Mode in Part I Operation.

T1~T8: Tool post tool position signal, connecting with the tool post; 4~16 station tool post is selectable; when the tool number is more than 8, it must be sent to the system via an coder.

TCP: Tool post clamping in-position signal, connecting with tool post.

### 3.2.3 Connection Method of Output Signal

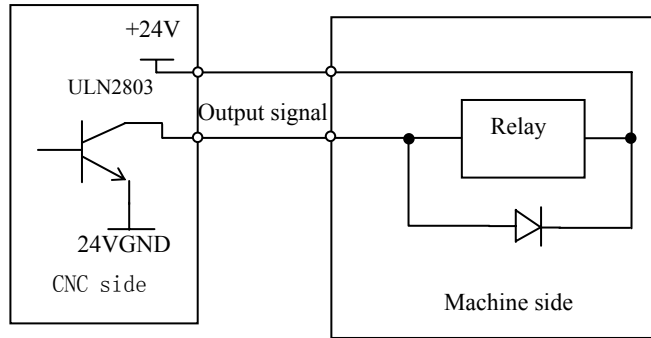
The output signal (UO01~UO18) is used to control the relative operations of the machine; signal direction: CNC to machine. The signal can drive the relay and the indicator at the machine side. When the output signal is connected with 24VGND, the output function is valid; when it is disconnected with

24VGND, the output function is invalid. X1, X2 interfaces have 18-channel output interface, all of which have the same structure. The output signal connection method is as the following figure.

1) output signal connection

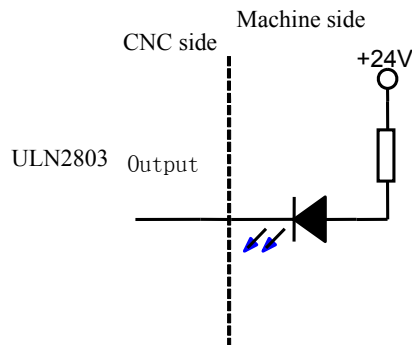
● Driving inductive load:

Use ULN 2803 output to drive the inductive load, at the moment, connect the fly-wheel diode in the vicinity of the coil to protect the output circuit and reduce the interference.



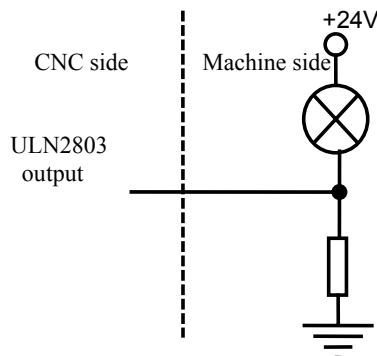
● Driving LED:

Use ULN2803 output to drive LED with a serial resistor to limit the current (about 10mA) through the LED.



● Driving filament indicator

Use ULN2803 output to drive the filament indicator, and connect an external preheat resistor to reduce the current impact. The preheat resistor value is based on the condition that the indicator does not light up.



2) Output special signal

- The output special signal are driven by ULN2803 transistor array, and the max. load instantaneous current is 200mA. When the signal is valid, the transistor is connected and common terminal is

+24V.

- When the machine side is connected with the relay or other inductive load, the spark suppressor which should be placed near the load (within 20cm) must be used. And the serial current-limiting resistance must be used when the machine side is connected with the capacitance load.
- The output special signal M8, M9, M3, M4, M5, M10, M11, M78, M79 can be set to the level control mode or pulse control mode by the bit parameter, and whether M4, MSP special signals are output is set by the parameter; For the time sequence of the signal output, refer to JOG Operation Mode in Part I Operation.
- For the gear shifting execution process and signal output time sequence of output special signal S01, S02, S03, S04, M41, M42, M43, M44, see JOG operation mode in Part I Operation.
- Output special signal TL+ and TL- are tool post CW/CCW signal respectively.

### 3.2.4 Input/Output Signal Specification

- Supply voltage: +24V
- Max. load instantaneous current of the output signal: 200mA

**【Note】**

- 1) When the transistor of the electronic switch is connected, the output voltage should be within 1V; when the transistor is turned off, the output voltage should be more than 23V.
  - 2) When the input function is valid, the input signal is connected with 24VGND; when it is invalid, the signal is disconnected with 0V.
  - 3) When the output function is valid, the signal is connected with 24VGND; when it is invalid, the signal is in high-impedance state.
- ⚠ It is suggested that the input/output cable should use the shield cable, and the shield layer be connected with the plug metal cover and the machine to improve the anti-interference ability.

### 3.3 Machine Zero Return Function and Connection

Interface signals relative to machine zero return are the X1, X2, X4 interfaces in the following table:

1) X1 interface:

Pin 2	DECZ	Z deceleration signal
Pin 3	DECX	X deceleration signal
Pin 4	DECY	Y deceleration signal

2) X2 interface:

Pin 5	PCX	X zero signal
Pin 13	PCZ	Z zero signal

3) X4 interface:

Pin 24	PCY	Y zero signal
--------	-----	---------------

**【Note】**

1) 928TD has no Y axis function, so it is unnecessary to connect Y deceleration signal and Y zero signal.

**【Explanation】**

- Bit parameter P406\_d7, P406\_d6, P406\_d5, P406\_d4, P406\_d3, P406\_d2 set the machine zero return mode.
- Setting bit parameter to 0 means there is no deceleration signal or zero signal; setting bit parameter to 1 means there is the deceleration signal or zero signal.
- There are four zero return methods. For the parameter setting methods, see Parameter Operation Mode in Part I Operation; for the description of the zero return execution process, see JOG Operation Mode in Part I Operation.

Connection example:

1) The two figures below show two kinds of connection examples of deceleration signal: travel switch, NPN proximity switch; it is suggested that the user should use the travel switch; the connection example is as follows:

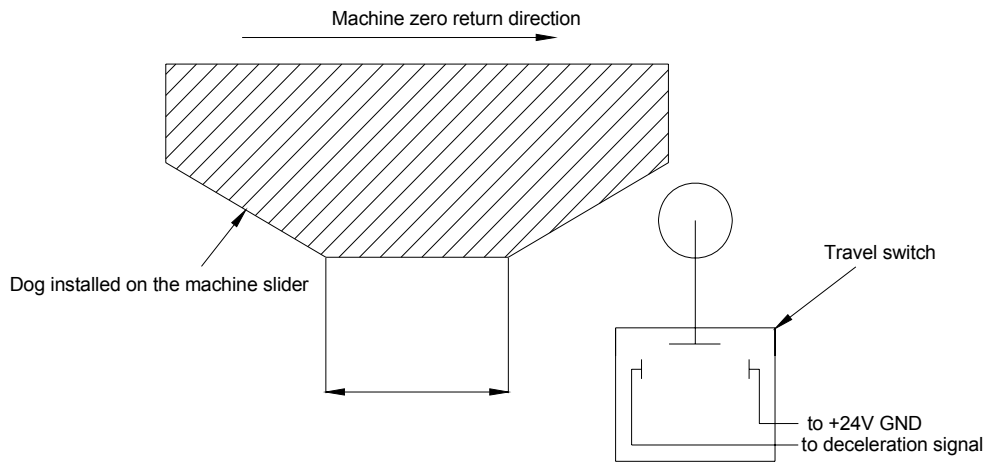


Fig. 3-1 Deceleration signal connected to travel switch

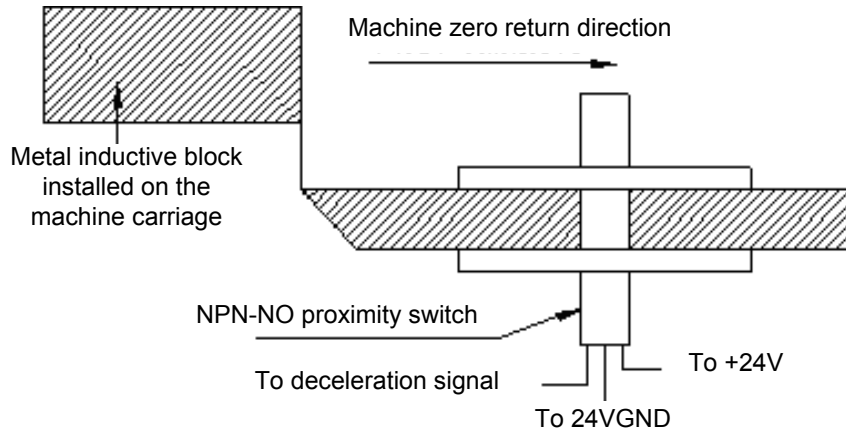


Fig. 3-2 Deceleration signal connected to NPN proximity switch

- 2) When the bit parameter **P407\_d7**, **P407\_d6**, **P407\_d5**, **P407\_d4**, **P407\_d3**, **P407\_d2** are all set to 0, the machine zero return direction is positive, and the deceleration signal low level is valid. The figures below are the connection example of the deceleration signal and zero signal respectively; the one-turn signal of servo motor is taken as the zero signal when the system is connected to the servo motor.

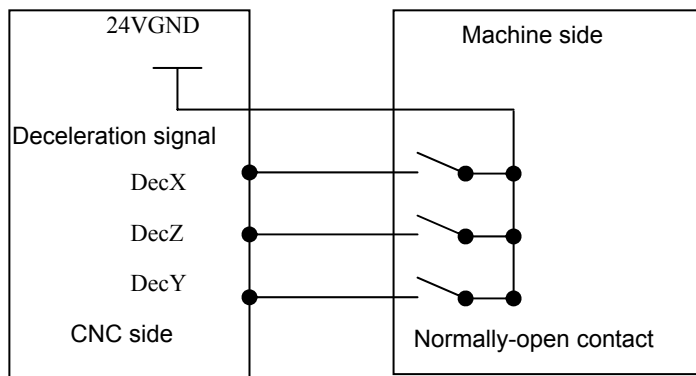


Fig. 3-3 Connection of deceleration signal

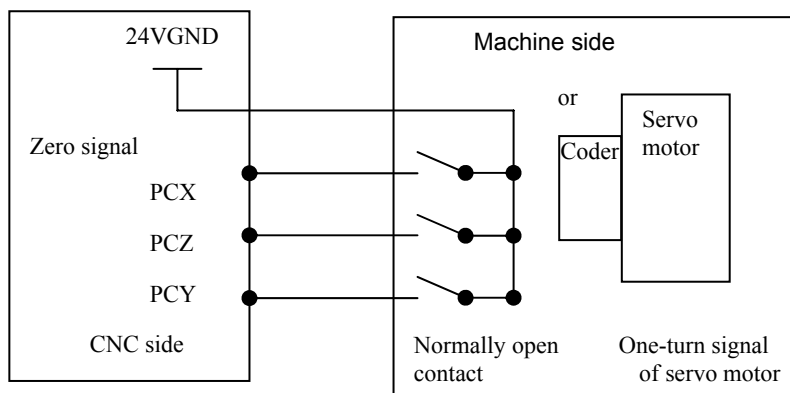


Fig. 3-4 Connection of zero signal

### 3.4 Tool Change Control Function and Connection

#### 3.4.1 Definition of Tool Change Control Signal

Interfaces relative to tool change control are as follows:

**Table 3-3**

X2 interface	Common signal name	Special signal name	Special signal function	Remarks
Pin 8	UI08	T08	Tool post position signal T8	Common input interface UI08
Pin 15	UI07	T07	Tool post position signal T7	Common I input interface UI07
Pin 7	UI06	T06	Tool post position signal T6	Common input interface UI06
Pin 14	UI05	T05	Tool post position signal T5	Common I input interface UI05
Pin 12	UI04	T04	Tool post position signal T4	Common I input interface UI04
Pin 4	UI03	T03	Tool post position signal T3	Common Common input interface UI03
Pin 11	UI02	T02	Tool post position signal T2	Common input interface UI02
Pin 3	UI01	T01	Tool post position signal T1	Common input interface UI01
	O	TZD	Tool post worktable brake	Applicable to SWD120,AK31,SBWD-80 tool post
	O	TFD	Tool post worktable pre-indexing output	Applicable to AK31,SBWD-80 tool post
Pin 1	UO12	TL+	Tool post (CW) output signal	Can be used as common output interface UO12
Pin 9	UO11	TL-	Tool post (CCW) output signal	Can be used as common output interface UO11
	I	TFDC	Pre-indexing in-position signal	Applicable to AK31,SBWD-80 tool post
	I	TXT	Tool position strobe signal	Applicable to AK31,SBWD-80 tool post
	I	TGR	Tool post overheat check	Applicable to AK31,SBWD-80 tool post
Pin 6	UI09	TCP	Tool post clamping in-position signal	Common input interface UI09

Note: For the explanation of AK31, SBWD-80 tool post, refer to the user manuals of Yantai AK31 series tool post and Changzhou SBWD-80 series tool post .

#### 3.4.2 Signal Connection

The connection method of common input/output (UI01~UI09,UO11~UO12) interface accords to the X1, X2 interface connection principle above. When the tool number signal is connected to the machine external, and the tool number is in position, the corresponding tool number signal is connected to 24VGND.

#### 3.4.3 Function Description

The tool change mode of this system is set by P318. The tool change mode and the control time sequence are as follows:



3.4.3.1 Tool Change Mode 0

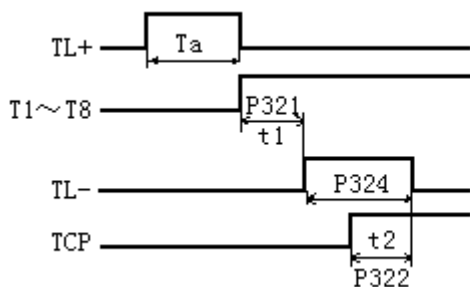
**P318=0: tool change mode 0**, which is applicable to the line-up tool post.

3.4.3.2 Tool Change Mode 1

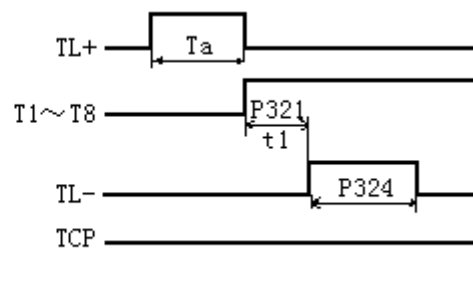
**P318=1: tool post type 1, tool change mode 1**, which is applicable to the common 4-satation electric rotating tool post, tool changing (CW) and tool clamping (CCW). (if **P318=5, 6, 7, 8**, the system also changes the tool by tool change mode 1).

Execution steps are as follows:

- 1) The system calculates the new tool nose coordinates according to tool offset number, and displays them;
- 2) If the current tool is the target tool, and the detected tool number signal is correct, the tool change output control is not performed; otherwise, the system goes to next step.
- 3) The system outputs the tool post (CW) signal TL+ and checks the tool number signal of the target tool, and then turns off TL+ output after it checks the signal; if the tool number signal is not detected within the period time set by P325 (tool change time limit), it turns off TL+ output and issues “**Check tool number signal overtime**”
- 4) The system outputs tool post (CCW) signal TL- after delaying the time set by data parameter P321 (t1);
- 5) Different processings are performed based on whether P408\_d6 has the tool post clamping in-position signal TCP:
  - The system delays the time set by **P324** (tool post (CCW) clamping time) if the tool post clamping in-positioni signal TCP( **P408\_d6= 0**) is not checked; the next step is executed after the time is up.
  - The system checks TCP within the time set by P324 if the tool post clamping in-position signal TCP (**P408\_d6=1**) needs to be checked; the system turns off TL- output and issues “**Tool post clamping overtime**” if it has not received TCP; the system delays the time set by **P322** (t2) if it receives the TCP signal, and then proceeds to the next step.
- 6) The system turns off the tool post (CCW) output signal (TL-), and the tool change is completed.
- 7) Finally the sytem confirms the signal, i.e. checks whether the current tool number input signal is consistent with the current tool number; if so, the tool change is completed; if not, it issues an alarm.



TCP clamping signal check required



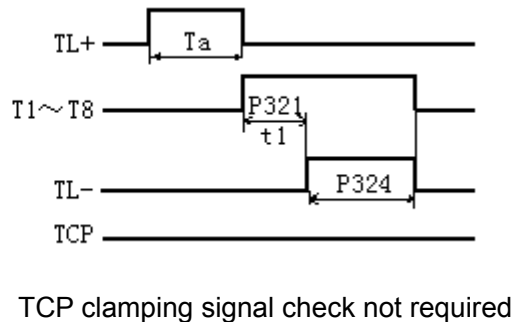
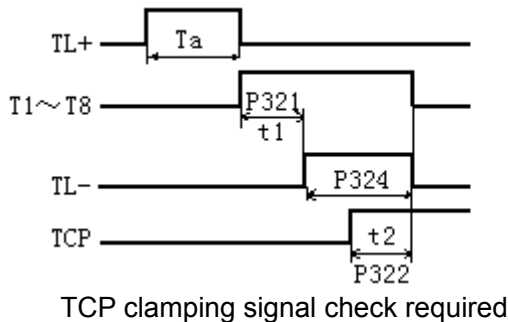
TCP clamping signal check not required

3.4.3.3 Tool Change Mode 2

**P318=2: tool post type 2, tool change mode 2**, which is applicable to the common 4-station electric rotating tool post, tool changing (CW) and tool clamping (CCW).

The execution steps are as follows:

- 1) The system calculates the new tool nose coordinates according to tool offset number, and displays them;
- 2) If the current tool number is the target tool number, the following tool change operation is not performed (e.g. if the current tool is T11, and T11 is executed again, the tool change operation is not performed), the tool change is completed.
- 3) The system outputs the tool post (CW) signal TL+ and checks the tool number signal of the target tool, and then turns off TL+ output after it detects the signal; if the tool number signal is not detected within the period of time set by P325 (tool change time limit), it turns off TL+ output and issues "Check tool number signal overtime"
- 4) The system outputs tool post (CCW) signal TL- after delaying the time set by data parameter P321 (t1);
- 5) Different processings are performed based on whether P408\_d6 has the tool post clamping in-position signal TCP:
  - The system delays the time set by **P324** (tool post (CCW) clamping time) if the tool post clamping in-position signal TCP( **P408\_d6= 0**) is not checked; the next step is executed after the time is up.
  - The system checks TCP within the time set by **P324** if the tool post clamping in-position signal TCP( **P408\_d6=1**) needs to be checked; the system turns off TL- output and issues "**Tool post clamping overtime**" if it does not receive TCP; the system delays the time set by **P322** (t2) if it receives the TCP signal, and then proceeds to the next step.
- 6) The system turns off the tool post (CCW) output signal (TL-), and the tool change is completed.



3.4.3.4 Tool Change Mode 3

**P318=3: tool post type 3, tool change mode 3**, which is applicable to Taiwan Hengyang

CLT-63~CLT300 turning machine tool tower series, the system nearby executes the tool change CW/CCW.

In tool change mode 3, max. tool quantity (P319) and the lines occupied by (P320) are set by the corresponding tool post type.

Used signal definitions are as follows:

Table 3-4

Tool post signal	CNC signal	Signal explanation
SensorA	T1	Tool post tool number signal T1
SensorB	T2	Tool post tool number signal T2
SensorC	T3	Tool post tool number signal T3
SensorD	T4	Tool post tool number signal T4
SensorE	TXT (P529)	Input tool inductive signal (tool number strobe signal)
SensorF	TCP	Tool post clamping in-position signal
Sol A1	TZD (P507)	Tool post releasing control signal(tool post worktable brake output)
Sol A2	TFD (P508)	Tool post clamping control signal (tool post worktable pre-indexing output)
Sol B	TL+	Tool post CW output signal
Sol C	TL-	Tool post CCW output signal

The tool change signals are explained as follows:

- 1) Sensor A, B, C, D are only for tool number check. They are not used as the starting signals of any operations;
- 2) Sensor E: It senses once each time the tool position is changed, and it is the starting signal of the tool post clamping;
- 3) Sensor F: Releasing/clamping confirmation signal. High level for clamping, low level for releasing;  
Releasing means the tool post is released and the the tool post can start to rotate;  
Clamping means the oil pressure motor stops rotating;
- 4) Sol A1, Sol A2: Two-head electromagnetic valve. When Sol A1 is turned on and Sol A2 is turned off, the tool post is released; when Sol A1 is turned off and Sol A2 is turned on, the tool post is clamped;
- 5) Sol B, Sol C: Two-head electromagnetic valve. It controls the CW/CCW rotation of the oil pressure motor, and it returns to the middle position after the clamping is confirmed, then the motor stops rotating.

Tool change sequence as follows:

- 1) The system calculates the new tool nose coordinates according to tool offset number, and displays them;
- 2) If the current tool number is the target tool number, the following tool change operation is not performed;

- 3) Sol A1 is ON, SolA2 is OFF, and the tool post is released;
- 4) When the sensor F is invalid, the tool post is released. If the system cannot detect that the sensor F is invalid within the time set by P325, it issues “**Tool post releasing overtime**”;
- 5) Control the tool post to rotate (CW/CCW) to execute the tool change nearby by judging the tool change direction according to the current tool number and the target tool number;
- 6) After the SensorE falling edge which has checked the tool post rotating to the tool signal before the target tool number appears, the system executes the next step;
- 7) After the SensorE drising edge which has checks the tool post rotating to the tool signal before the target tool number appear, SolA1 is turned off, SolA2 is turned on, the tool post is clamped (at the moment, Sol B still keeps ON, and the motor normally rotates); when the system has not detected the tool number signal before the target tool number or the tool number signal of the target tool within the time set by P325 tool change time limit, the system alarms: “**Check tool signal overtime**”;
- 8) When the system checks the SensorF is valid, Sol B is turned off, the motor stops rotating, and the tool change is completed; the system alarms “**Tool post clamping overtime**” when it has not checked SensorF is valid in the t time set by **P324**.

**【Example】**

Example: the 8-station turret executes the tool change from No.1 to No.4 tool.

- 1) Sol A1 is turned on, Sol A2 is turned off, and the tool post is released;
- 2) The system waits the Sensor F is invalid, judges the nearby tool change direction, Sol B is turned on and motor rotates;
- 3) The system starts to check the tool signal (Note: SensorE generates the valid signal when it is in the No. 1, No. 2, No.3 tool, but the clamping operation is not performed if the 4<sup>th</sup> tool is not reached); when it checks No. 3 tool signal, it sets SensorE preparatory operation to make that the tool post is rotate to No. 4 tool; Once SolA1 is valid, SolA1 is turned off and SolA2 is turned on immediately and the tool post is clamped (at the moment, So1 B still keeps ON and the motor normally rotates).
- 4) When the system has checked Sensor F is valid, So1 B is turned off, the motor stops rotating and the tool chang is performed.

3.4.3.5 Tool Change Mode 4

**P318=4: tool post type 4, tool change mode 4.** It is applied to the tool post type which executes the nearby tool change (CW/CCW).

**Table 3-5**

CNC signal	Signal explanation
T1~T8	Tool post tool positon signal
TCP	Tool clamping in-position signal
TFD (P508)	Tool post clamping control signal (tool post worktable pre-indexing)
TL+	Tool post rotation (CW) outputl signal

TL-	Tool post rotation (CCW) outputl signal
-----	---

Execution steps are as follows:

- 1) The system calculates the new tool nose coordinates according to tool offset number, and displays them;
- 2) If the current tool number is the target tool number, and the tool number signal detected is also correct, the following tool change operation is not performed; otherwise, the next step is executed.
- 3) The system outputs corresponding tool post rotation output signal (TL+ or TL-) and then starts to check the tool number signal of the targe tool, by judging the tool change direction according to the current tool number and targe tool number. if the tool number signal is not detected within the period of time set by P325 (tool change time limit), it turns off the tool post rotation signal and issues "Check tool number signal overtime"
- 4) After detecting the tool in-position signal, the system outputs the tool post clamping control signal (TFD) to casue the tool post to enter the clamping control state, with the tool post rotation signal remaining ON.
- 5) Different processings are performed based on whether P408\_d6 has the tool post clamping in-position signal TCP:
  - Delay the time set by **P324** (tool post (CCW) clamping time) if the tool post clamping in-positioni signal TCP ( **P408\_d6= 0**) is not checked; the next step is executed after the time is up.
  - The system checks TCP within the time set by **P324** if the tool post clamping in-position signal TCP (**P408\_d6=1**) needs to be checked; the system turns off tool post rotation signal and issues "**Tool post clamping overtime**" if it does not receive TCP; the system delays the time set by **P322** (t2) if it receives the TCP signal, and then proceeds to the next step.
- 6) The system turns off the tool post rotation output signal (TL+ or TL-) and tool post clampoing control signal (TFD);
- 7) Finally the system confirms the signal, i.e. checks whether the current tool number input signal is consistent with the current tool number; if so, the too change is completed; if not, the CNC issues an alarm.

### 3.4.4 Tool Signal Check and Parameter Setting

The tool signal check is controlled by **P408\_d7**. When **P408\_d7=0**, the tool signal check is the default mode; when **P408\_d7=1**, the tool signal check is the table checking mode. Note: line-up tool (**P318=0**) is not influenced by **P408\_d7**.

#### 3.4.4.1 Default mode (**P408\_d7=0**)

In the default mode, P319 must be equal to P320. Otherwise, the system prompts "**P319** is not

equal to **P320** in tool change.

The definition of tool number signal in the default mode is as follows:

The number of tool number signal lines defined by P320 separately corresponds the tool numbers defined by P319, each signal line corresponds to one tool number, and the valid level of the tool signal is the low level. When a tool number signal is valid, the other tool number signals are invalid.

Example: When P319=4 and P320=4, the tool signals are as follows:

	T04	T03	T02	T01
No.1 tool	1	1	1	0
No.2 tool	1	1	0	1
No.3 tool	1	0	1	1
No.4 tool	0	1	1	1

Note: 1 in the table stands for the high level, 0 for the low level

Parameter setting is as follows:

Parameter	Parameter setting value	Remarks
<b>P318</b>	1	Tool change mode 1
<b>P319</b>	4	4-station tool post
<b>P320</b>	4	4 tool number signals
<b>P408_d7</b>	0	Bit parameter is set to the default mode

### 3.4.4.2 Table Checking Mode

In the table checking mode, P319 can be unequal to P320. The tool number signal is defined by P541~P556.

Tool number signal definition in table checking mode is as follows:

P541~P556 correspond to tool signal codes of No. 1~16 tool, and the tool code in the parameter is expressed by the decimal. The encoding method is as follows: all used tool signals constitute binary codes ( $T_{P320}...T1$ ) from the high to the low, and then the binary code corresponding to each tool number is converted to the decimal number to fill in the position corresponding to P541~P556. The binary code of the tool signal can also be seen in DIAGNOSIS mode.

Example: Genyuan CLT-63~CLT300 turning machine turret serial 8-station tool post uses 4 tool number signals, and the corresponding tool codes are as follows:

Tool Number	Tool code(T04...T01)	Decimal	Corresponding parameter
No. 1 tool	0010	2	P541
No. 2 tool	1000	8	P542
No. 3 tool	0001	1	P543
No. 4 tool	0100	4	P544
No. 5 tool	0111	7	P545
No. 6 tool	1101	13	P546
No. 7 tool	1011	11	P547
No. 8 tool	1110	14	P548

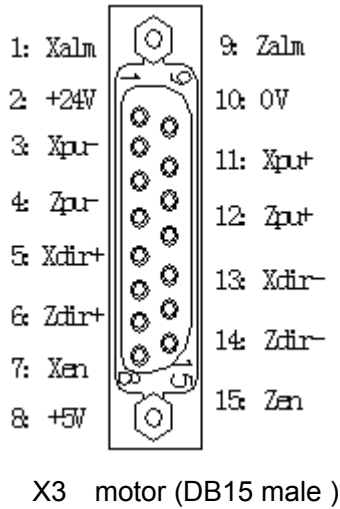
Parameter setting is as follows:

Parameter	Parameter setting value	Remarks
<b>P318</b>	3	Tool change mode 3
<b>P319</b>	8	8-station tool post
<b>P320</b>	4	4 lines
<b>P408_d7</b>	1	Bit parameter is set to table checking mode
<b>P541</b> tool number 1 check signal	2	Tool post tool number signal T1
<b>P542</b> tool number 2 check signal	8	Tool post tool number signal T2
<b>P543</b> tool number 3 check signal	1	Tool post tool number signal T3
<b>P544</b> tool number 4 check signal	4	Tool post tool number signal T4
<b>P545</b> tool number 5 check signal	7	Tool post tool number signal T5
<b>P546</b> tool number 6 check signal	13	Tool post tool number signal T6
<b>P547</b> tool number 7 check signal	11	Tool post tool number signal T7
<b>P548</b> tool number 8 check signal	14	Tool post tool number signal T8

### 3.5 X3 Motor Interface

The system can be matched with the drive units of the reaction stepper motor, the compound stepper motor and AC servo motor via the motor drive unit interface X3.

#### 3.5.1 Signal Definition



Pin	Special signal name	Special signal name explanation	Signal direction
1	Xalm	X drive unit alarm	Drive unit → CNC
2	+24V	X/Z drive power supply +24V	
3	XP-	X pulse negative terminal	CNC → Drive unit
4	ZP-	Z pulse negative terminal	CNC → Drive unit
5	Xdir+	X positive terminal	CNC → Drive unit
6	Zdir+	Z positive terminal	CNC → Drive unit
7	Xen	X axis enabled	CNC → Drive unit
8	+5V	X/Z drive power supply +5V	
9	Zalm	Z drive unit alarm	Drive unit → CNC
10	0V	Drive power ground	
11	Xpu+	X pulse positive terminal	CNC → Drive unit
12	Zpu+	Z pulse positive terminal	CNC → Drive unit
13	Xdir-	X negative terminal	CNC → Drive unit
14	Zdir-	Z negative terminal	CNC → Drive unit
15	Zen	Z axis enabled	CNC → Drive unit

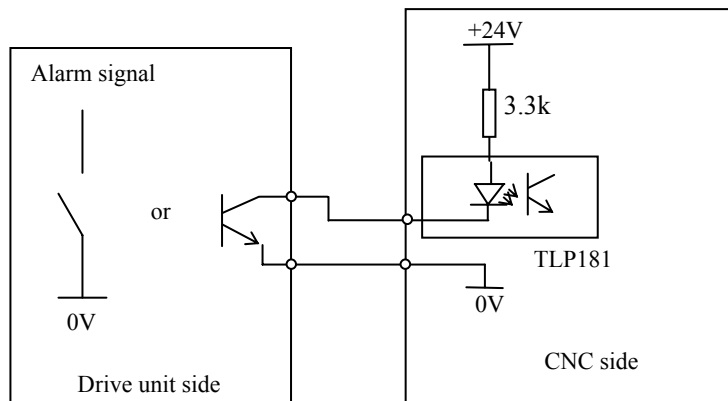
### 3.5.2 Technical Specifications

- Max. pulse output frequency: 500kHz
- Pulse width: 2μs

### 3.5.3 Equivalent Circuit

#### 3.5.3.1 Drive Unit Alarm Signal Xalm, Zalm

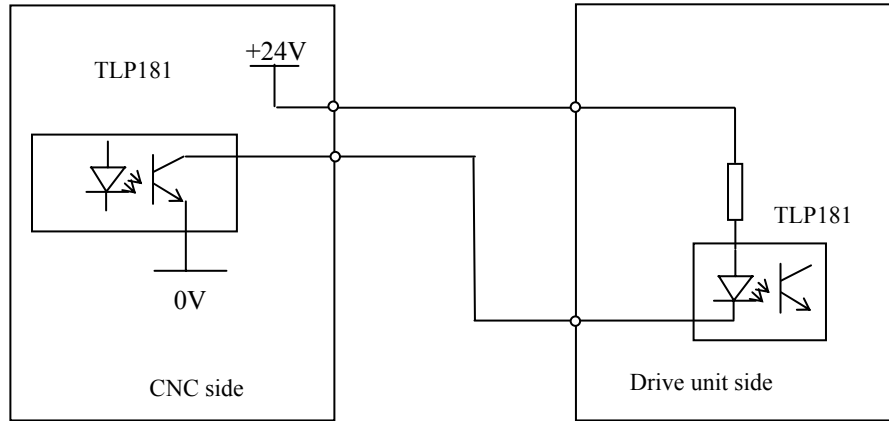
**P405\_d4, P405\_d3** set the drive unit alarm level is LOW or HIGH. The drive unit must use the following two methods to provide the signals:





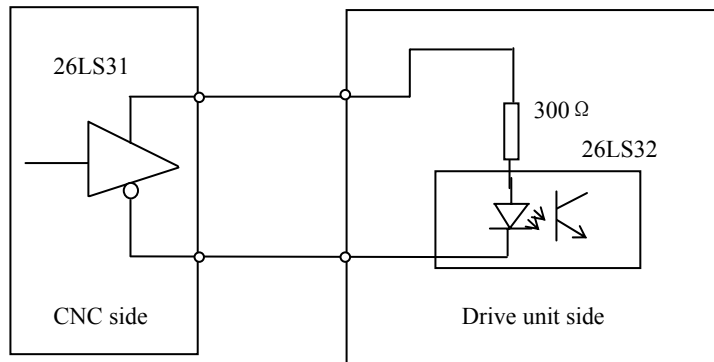
### 3.5.3.2 Enable Signal Xen, Zen

When the CNC is working normally, the enable signal output is valid (the enable signal is connected with 0V). The schematic circuit diagram is shown as follows:



### 3.5.3.3 Pulse Signal and Direction Signal

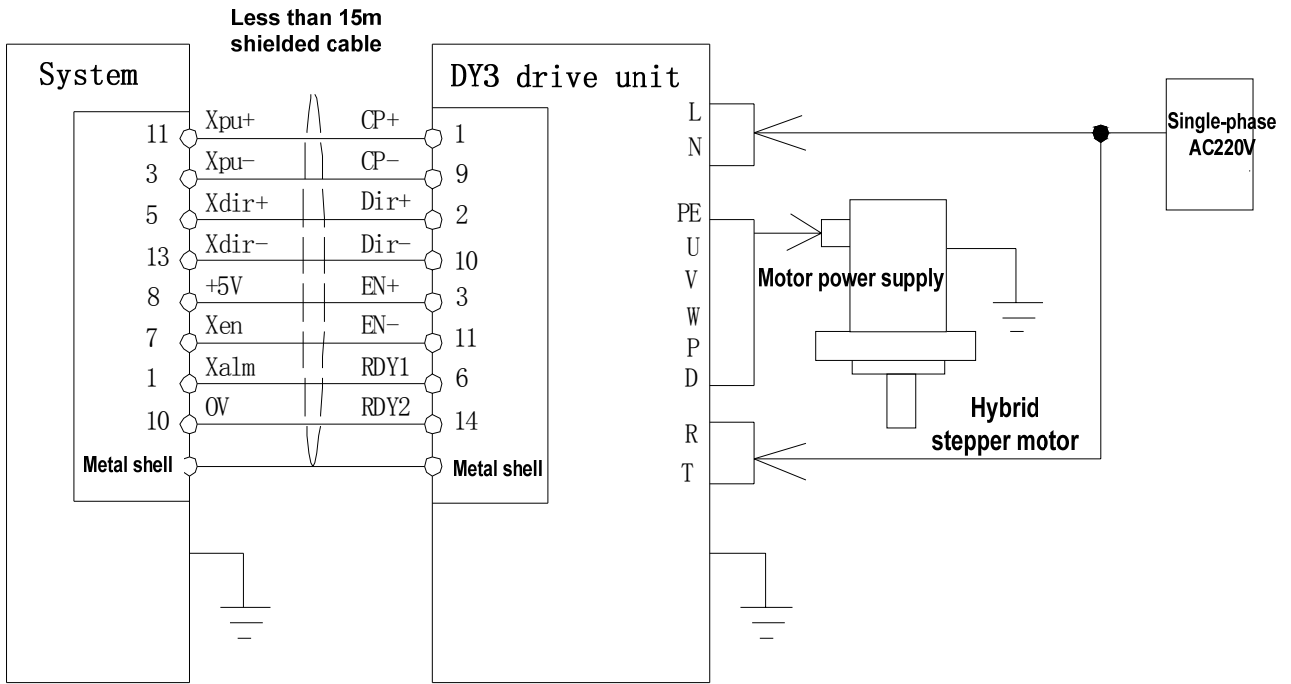
Xpu+, Xpu-, Zpu+, Zpu- are pulse signals, Xdir+, Xdir-, Zdir+, Zdir- are direction signals. The signals in the two groups are differential output, and it is suggested that the external should use 26LS32 to receive. The schematic circuit diagram is as follows:



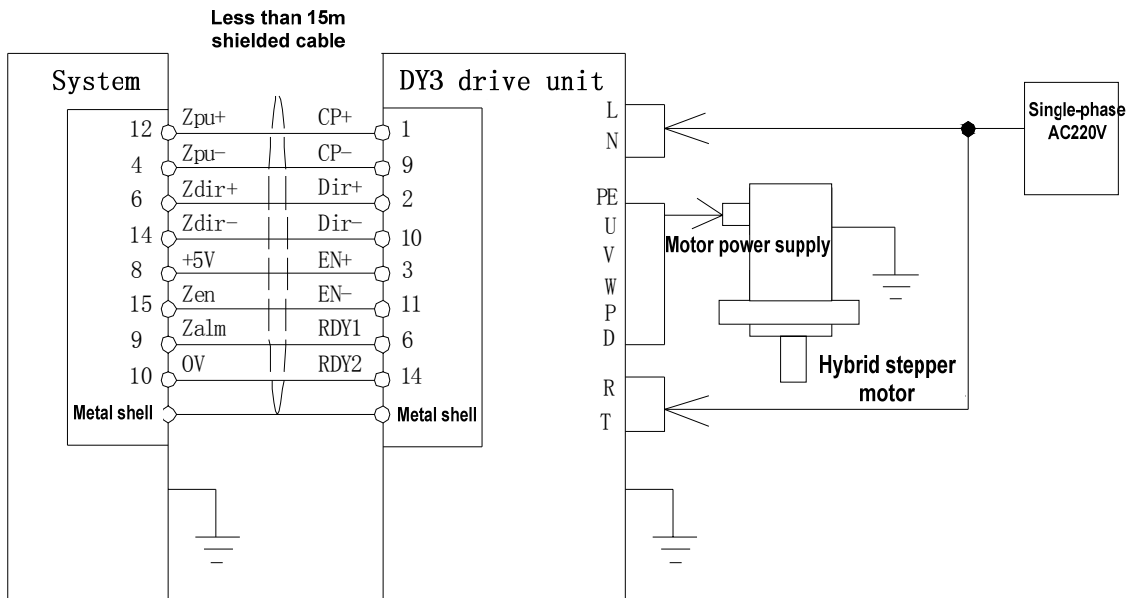
### 3.5.4 Connection between CNC System and Drive Unit of Compound Stepper Motor

Connection diagram between CNC and GSK DY3:

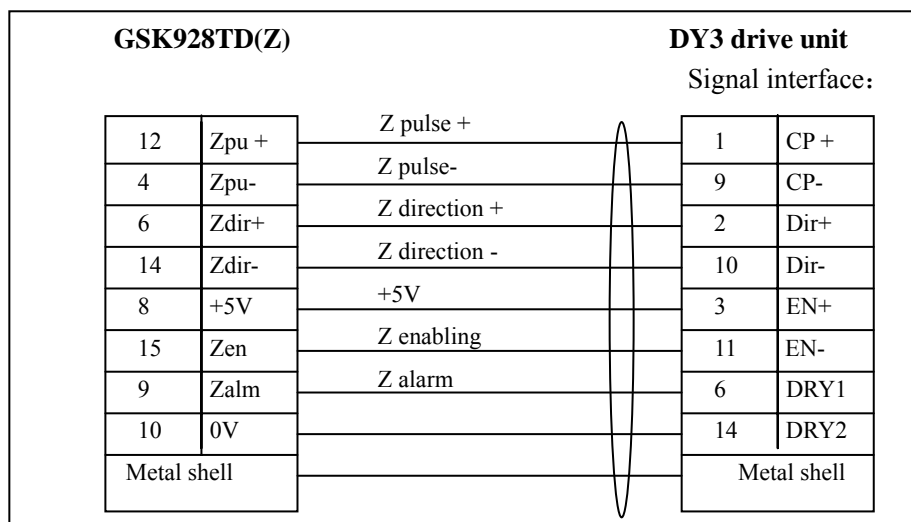
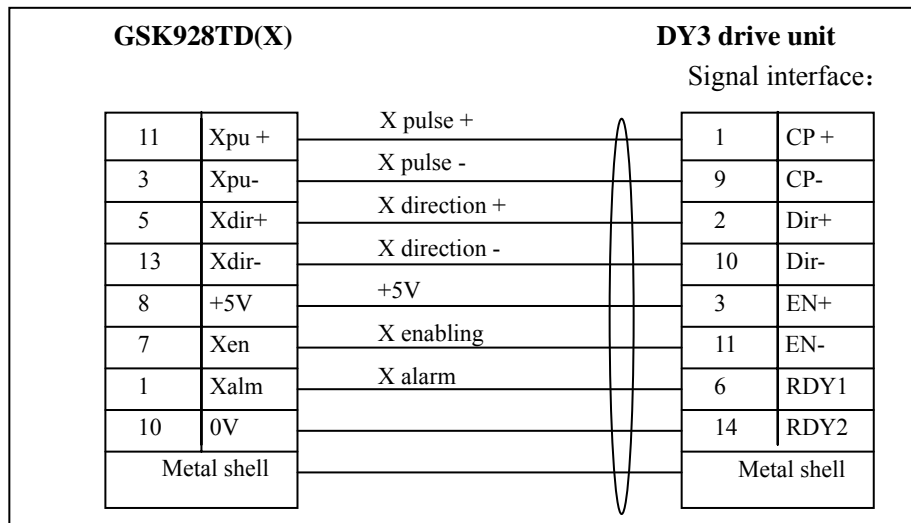
X axis connection diagram



Z axis connection diagram



Connection diagram between CNC and GSK DY3:



When other type of stepper drive unit is connected with the CNC system, the corresponding control switching value can be used. See the user manual of the corresponding drive device for the connection method.

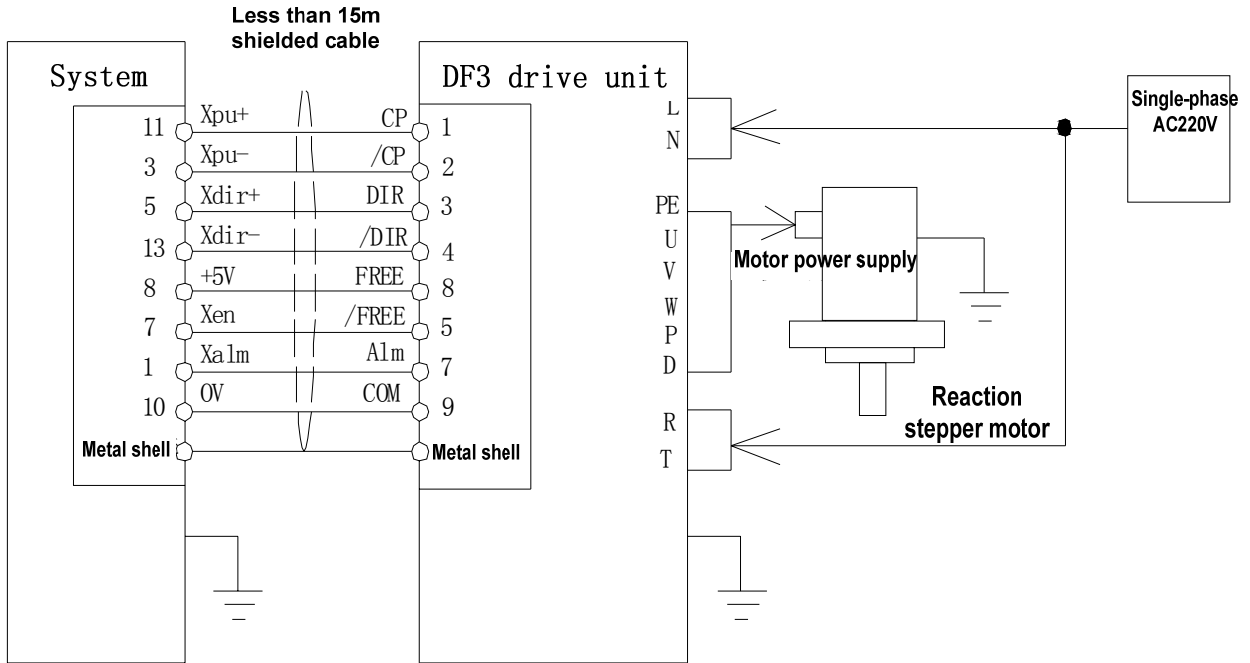
**【Note】**

- ❗ When the stepper motor is used, the user should adjust the bit parameter **P405(d 7~d 0)**, X-motion parameter(**P100~P116**) according to the actual conditions. See Parameter Operation Mode in this manual for details.
- ❗ The shield cable must be used to connect the stepper drive unit and CNC system. Otherwise the stepper motor may step out due to the external interference.
- ❗ The CNC system, stepper drive unit and stepper motor must be earthed reliably to prevent the motor stepping out because of the external interference.

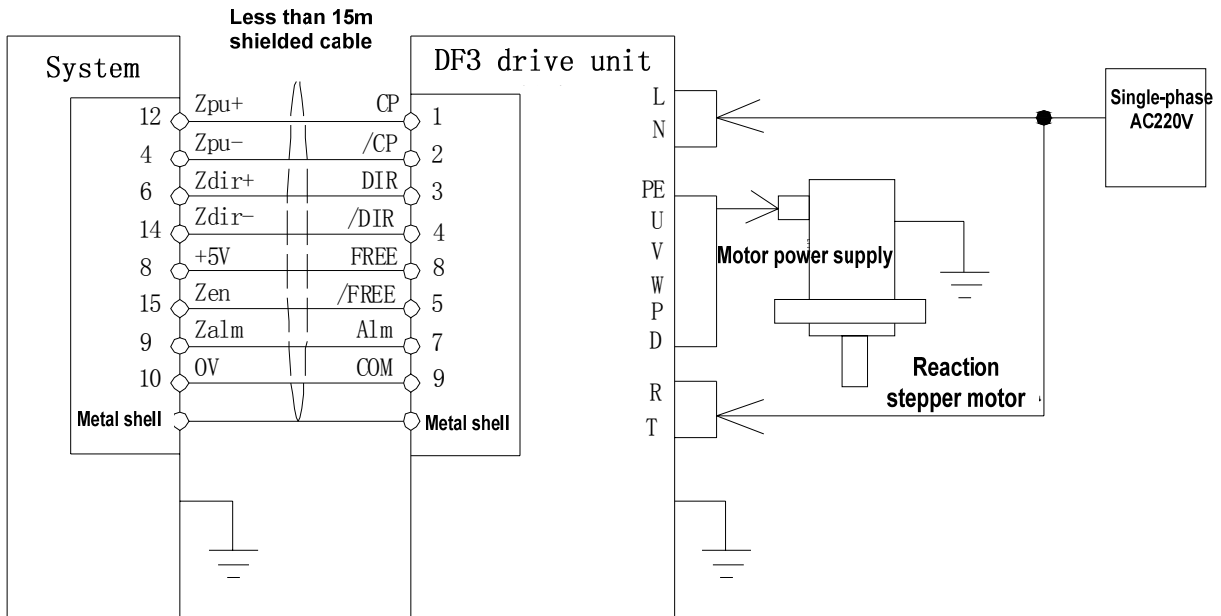
3.5.5 Connection between CNC and Drive Unit of Reaction Stepper Motor

Connection diagram between CNC and **GSK DF3**:

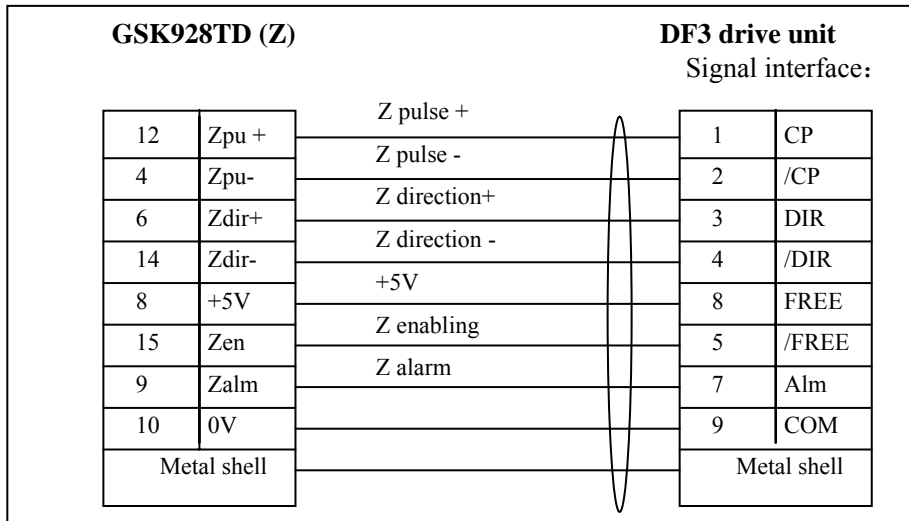
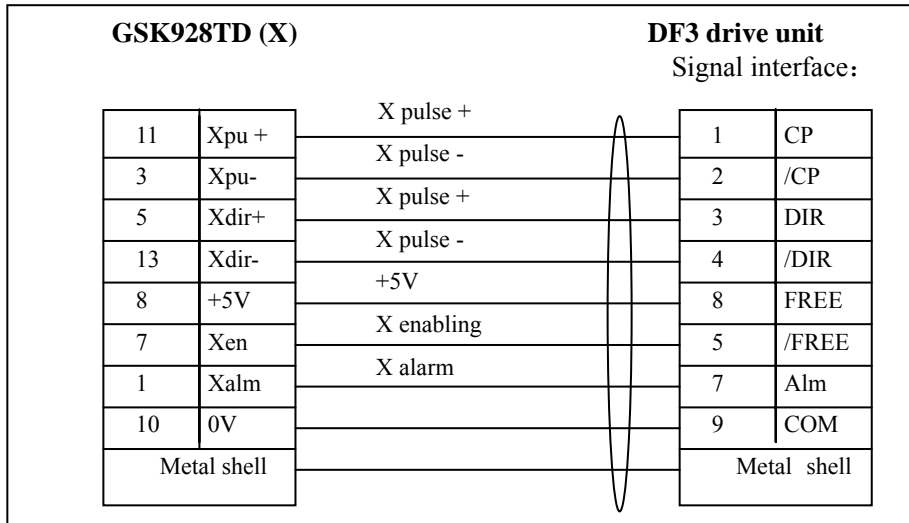
X connection diagram:



Z axis connection diagram



Connection diagram between CNC and **GSK DF3**:



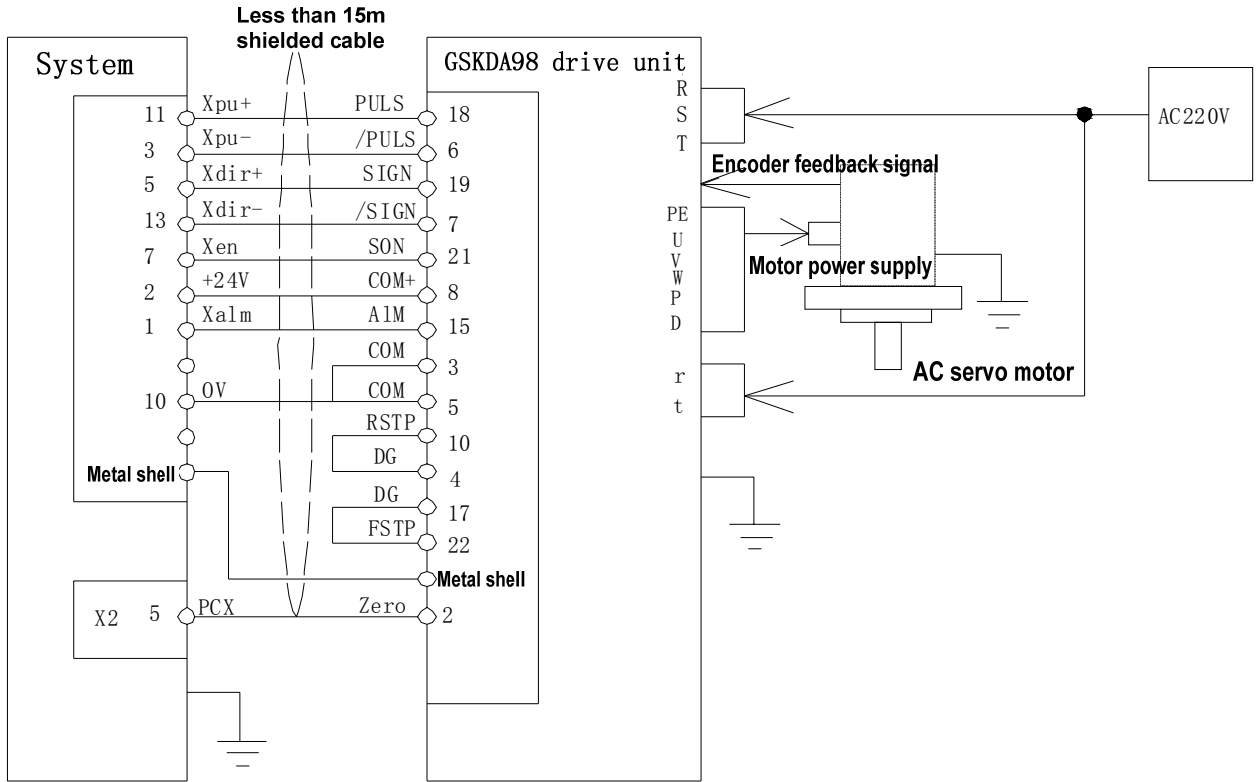
**【Note】**

- ❗ When the stepper motor is used, the user should adjust the bit parameter **P405(d 7~d 0)**, X-motion parameter(**P100~P116**) according to the actual conditions. See Parameter Operation Mode in this manual for details.
- ❗ The shield cable must be used to connect the stepper drive unit and CNC system. Otherwise the stepper motor may step out due to the external interference.
- ❗ The CNC system, stepper drive unit and stepper motor must be earthed reliably to prevent the motor stepping out because of the external interference.

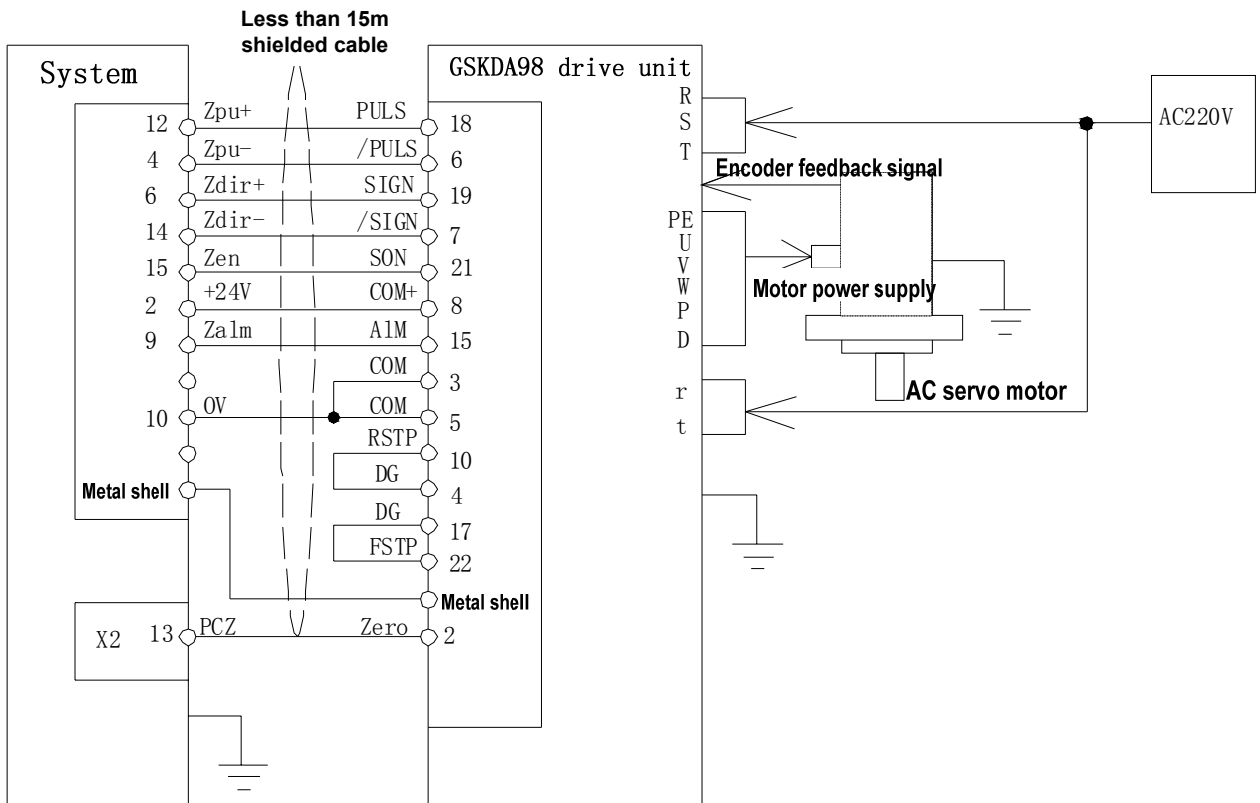
3.5.6 Connection between CNC and AC Servo Drive Unit

Connection diagram between CNC and **GSK DA98** AC servo drive unit:

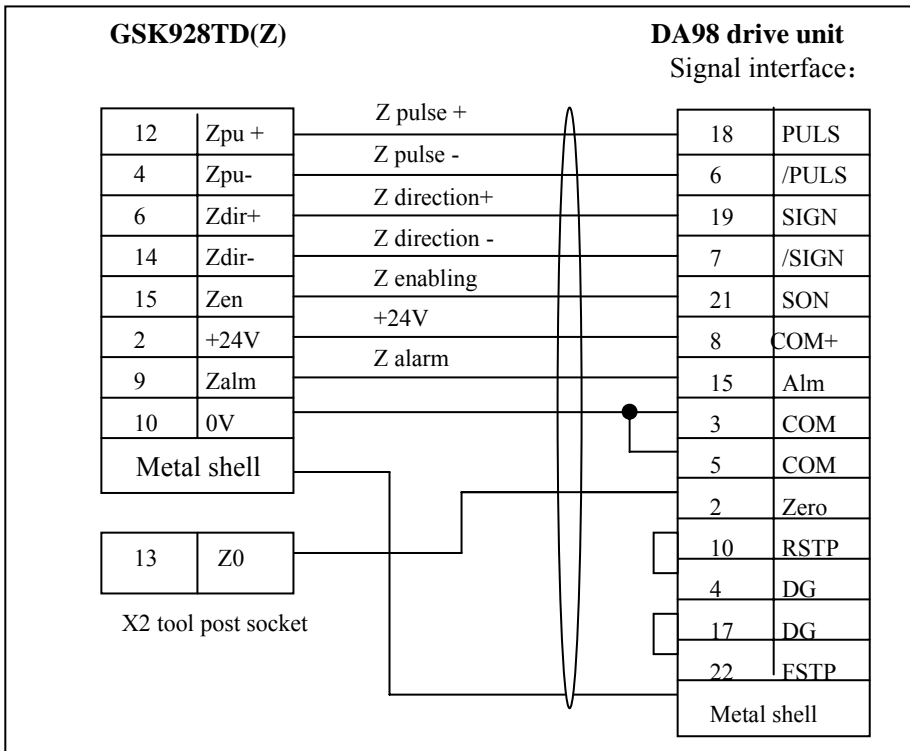
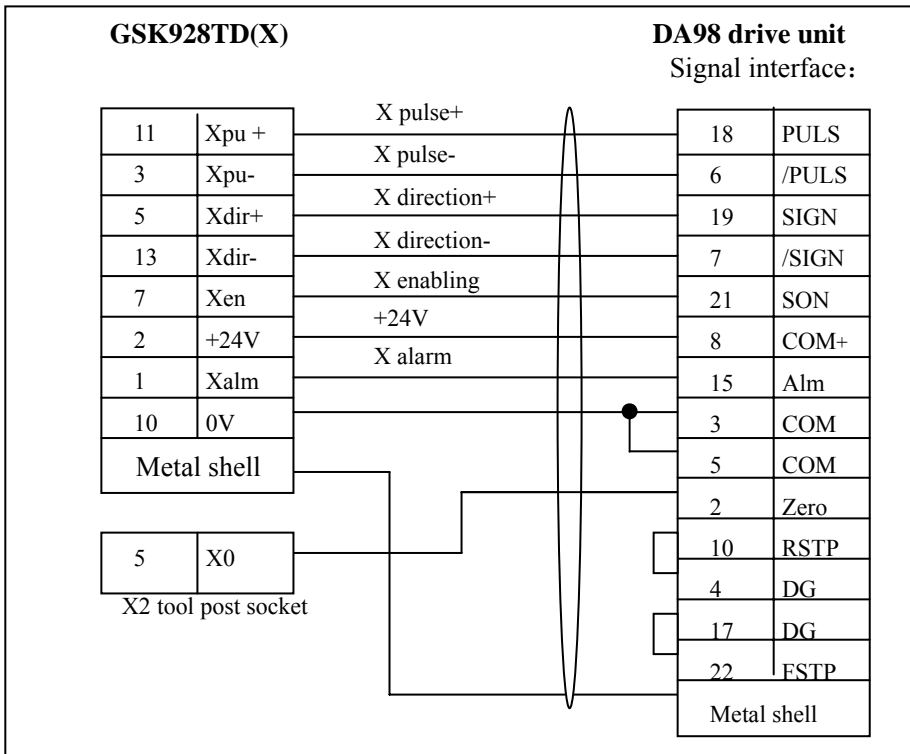
X axis connection diagram



Z axis connectin diagram



Connection diagram between CNC and GSK DA98 AC servo drive unit



【Note】

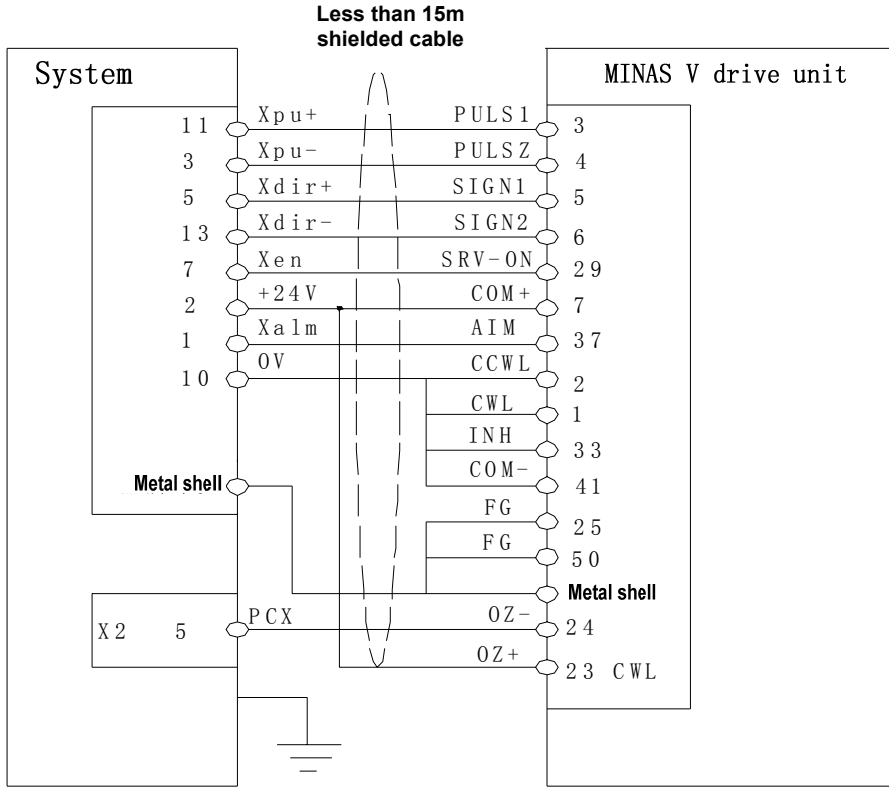
- ❗ When the servo motor is used, the user should adjust the bit parameter **P405(d 7~d 0)**, X-motion parameter(**P100~P116**) according to the actual conditions. See Parameter Operation Mode in this manual for details.
- ❗ The shield cable must be used to connect the servo driver unit and CNC system. Otherwise the motor may step out due to the external interference.

- ❗ The CNC system, servo drive unit and servo motor must be earthed reliably to prevent the motor stepping out because of the external interference.

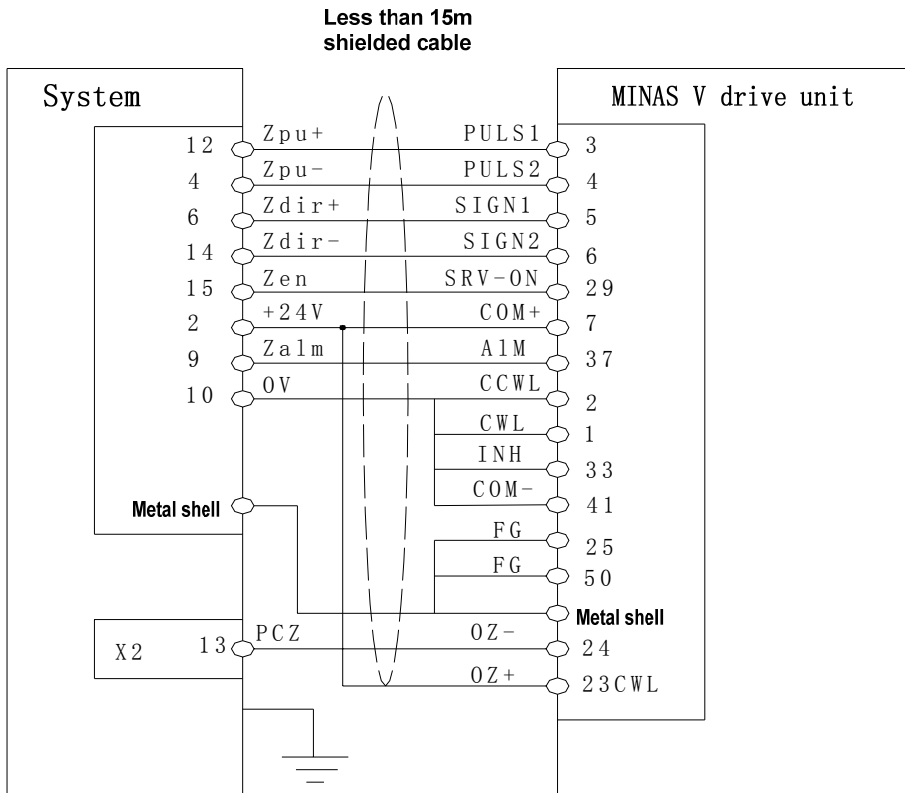
### 3.5.7 Connection Diagram between CNC and Panasonic Drive Unit

Connection diagram between CNC and Panasonic MINAS V serial drive unit:

X connection diagram (only for reference)



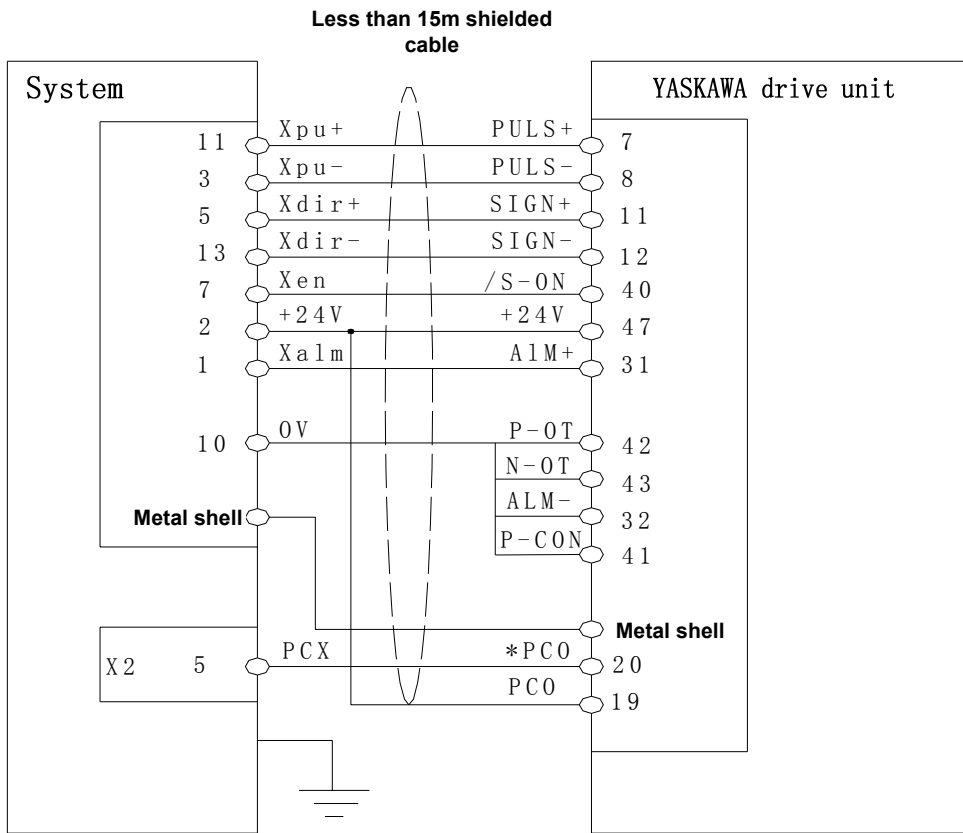
Z connection diagram (only for reference)



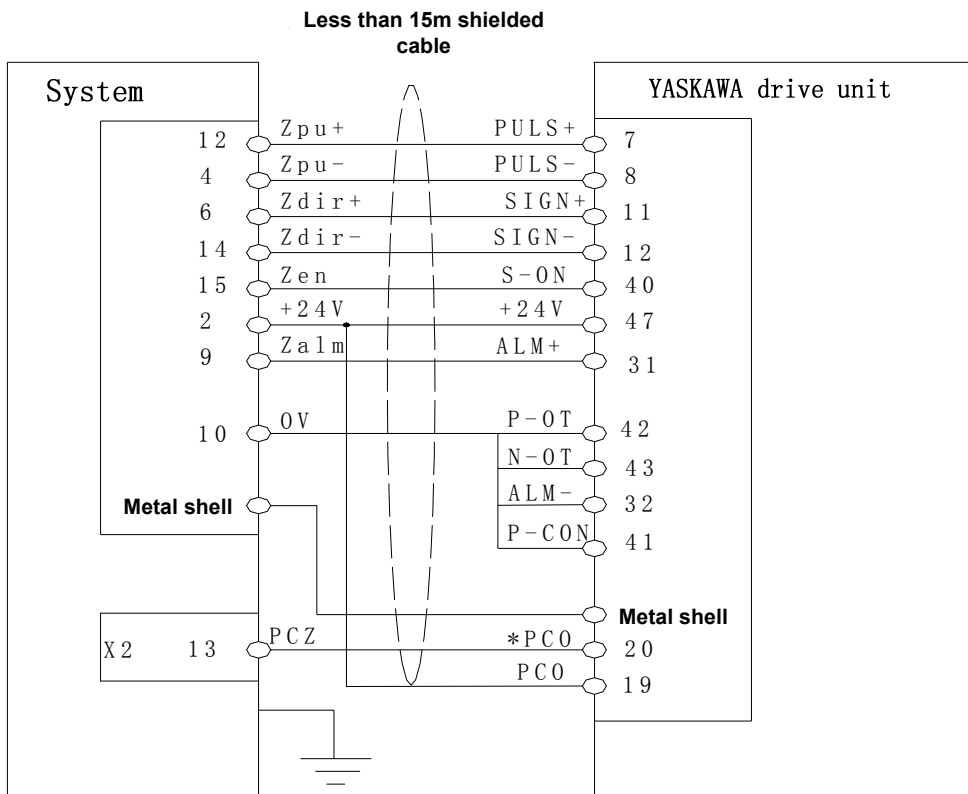


3.5.8 Connection Diagram between CNC and Yaskawa Drive Unit

X connection diagram (for reference only)



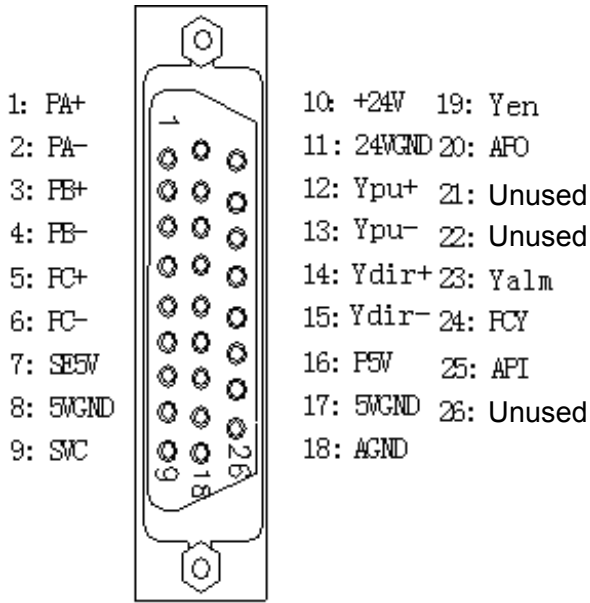
Z connection diagram (for reference only)



### 3.6 X4 Spindle Interface

The CNC system controls the connection between the voltage(SVC) output signal and the spindle converter by the spindle X4 converter, which realizes the stepless change-speed within limits.

#### 3.6.1 Signal Definitions



X4 spindle (DB26 female)

Pin name	Special signal name	Pin function
1	PA+	Encoder A pulse +
2	PA-	Encoder A pulse -
3	PB+	Encoder B pulse +
4	PB-	Encoder B pulse -
5	PC+	Encoder C pulse + (PZ+)
6	PC-	Encoder C pulse - (PZ-)
7	SE5V	Encoder 5V
8	0V (5VGND)	Encoder 5V earthing
9	SVC	Spindle analog voltage+
10	+24V	24V
11	0V (24VGND)	24V earthing
12	Ypu+	Y pulse + (Y function not supported)
13	Ypu-	Y pulse - (not supported)
14	Ydir+	Y + direction (not supported)
15	Ydir-	Y- direction (not supported)
16	P5V	Y drive unit 5V (not supported)
17	0V (5VGND)	Y drive unit 5V earthing (not supported)
18	AGND	Spindle analog voltage ground
19	Yen	Y enabling (not supported)
20	APO	Speed/position switch control (not supported)
21	Unused	Unused
22	Unused	Unused
23	Yalm	Y axis drive unit alarm (not supported)
24	PCY	Y axis zero point (not supported)
25	API	Speed/position state check (not supported)
26	Unused	Unused

#### 3.6.2 Converter Technical Specification

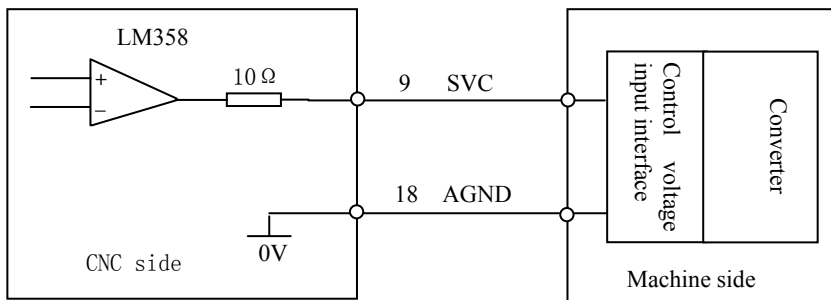
- System output analog voltage is 0 V~10 V.
- The connection cable between the system and the converter should be the shield cable, and the shield layer is connected with X4 socket metal shell.

### 3.6.3 Encoder Technical Specifications

- The system can connect an external spindle encoder via the spindle interface X4, which is used for the thread machining, and tapping.
- Encoders with different lines can be selected: set by parameter P209.
- Voltage +5V.

### 3.6.4 Connection Diagram of Converter Analog Voltage Interface

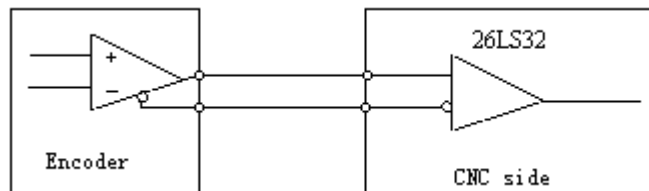
Analog spindle interface SVC can output 0~10V, and the circuit diagram is as follows:



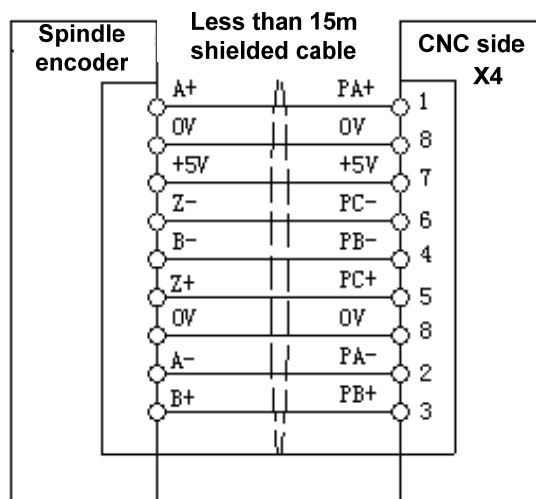
! The connection cable between the system and the converter should be the shield cable.

### 3.6.5 Encoder Interface Principle

Schematic circuit is as follows:



### 3.6.6 Encode Interface Connection Diagram

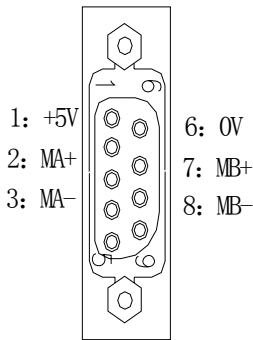


- ❗ The connection cable between the system and the spindle encoder must be the shield cable, and the shield layer must be connected the socket shell.
- ❗ When the spindle encoder output signal is not the difference output mode, PA-,PB-,PC- cannot be connected, and thus the anti-interference ability of the encoder output signal reduces. It is suggested that the system should use the spindle encoder with the difference output mode.

### 3.7 X5 MPG Interface

The CNC system can be connected to an external MPG via the MPG interface X5. MPG can control the movement of the coordinate axis. When the MPG connection line is less than 1m, the single-terminal connection can be used; when it is longer than 1.5m, it is advised to use the differential connection to improve the anti-interference ability.

#### 3.7.1 Signal Definition

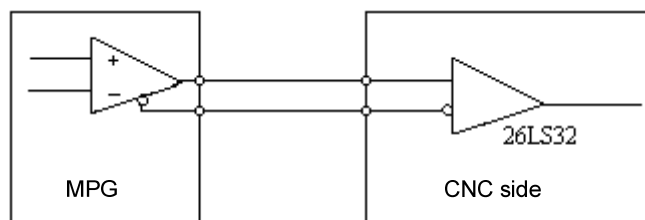


X5 MPG (DB9 female)

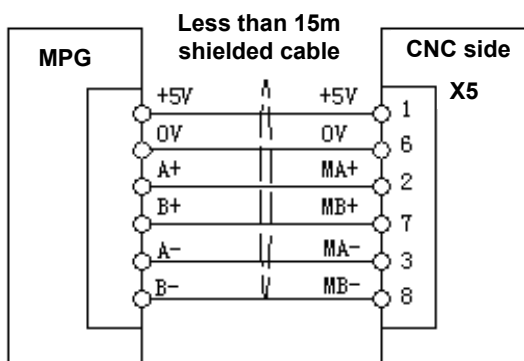
Pin	Special signal name	Pin function
1	+5V	MPG 5V
2	MA+	MPG A pulse+
3	MA-	MPG A pulse -
4	Empty	
5	Empty	
6	0V	MPG 5V ground
7	MB+	MPG B pulse +
8	MB-	MPG B pulse -
9	Empty	

#### 3.7.2 MPG Interface Principle

The schematic circuit is as follows:



#### 3.7.3 MPG Interface Connection Diagram



- ❗ When the axis movement is controlled by MPG, the moving direction of MPG cannot be changed quickly. Otherwise the moving distance does not coincide with the MPG dial;
- ❗ The shield cable should be used to connect the system and MPG;
- ❗ When the MPG output signal is not in the difference output mode, MA- can be disconnected with MB-;

## CHAPTER FOUR USE AND MAINTENANCE INFORMATION

### 4.1 Ambient Condition

System storage, transportation and working environmental condition are as follows:

Item	Working climate condition	Storage, transportation climate condition
Ambient temperature	0°C~45°C	-40°C~70°C
Relative humidity	≤90%RH(no dewing)	≤95%(40°C)

### 4.2 Earthing

The correct earthing in the electricity device is important and its aims are as follows:

- 1) Protect the operator from being injured because of the abnormal conditions;
- 2) Protect the electric devices from interference caused by the machine and other electric devices nearby. (the interference may cause abnormality of the control device).
- 3) The machine and system must be grounded firmly, and the neutral wiring in the electric wire net must not be used as the earthing line. Otherwise, it may cause injury to the operator or damage to the device.

### 4.3 Power Supply Requirements

The system can normally run in the following AC input power supply:

Voltage fluctuation: AC, (0.85~1.1) ×220V

Frequency fluctuation: 50Hz ± 2%

The requirements for power supply of machine tool is described in machine tool installation manual.

### 4.4 Protection

The protection grade is not less than IP20

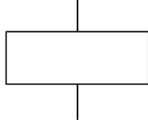
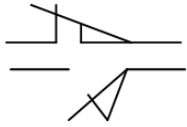

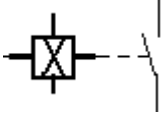


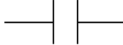
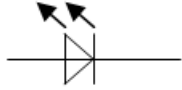



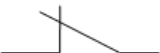
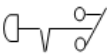
### 4.5 Use after Long-time Unuse

After the CNC system is not used for a long time, the user must firstly remove the dust, and dry up it, and then check the wiring and the earthing of the CNC system device, finally keep it power-on for a period of time to ensure that the system runs without any failure.

# **APPENDIX**

Appendix 1 CNC System Electrical Symbol Explanations

The following examples shown some electrical symbols of electrical components.

Name	Symbol	Graph	Name	Symbol	Graph
Relay	<b>K</b>		Stroke switch	<b>S</b>	
Motor	<b>M</b>		Hall proximity switch	<b>B</b>	
Semiconductor diode	<b>D</b>		Indicator	<b>E</b>	
Capacitor	<b>C</b>		LED	<b>P</b>	
Resistor	<b>R</b>		Normally-open contact		
Push-button switch with lock	<b>S</b>		Normally-closed contact		
Push-button switch without lock	<b>S</b>				



Shield layer



Protective earthing

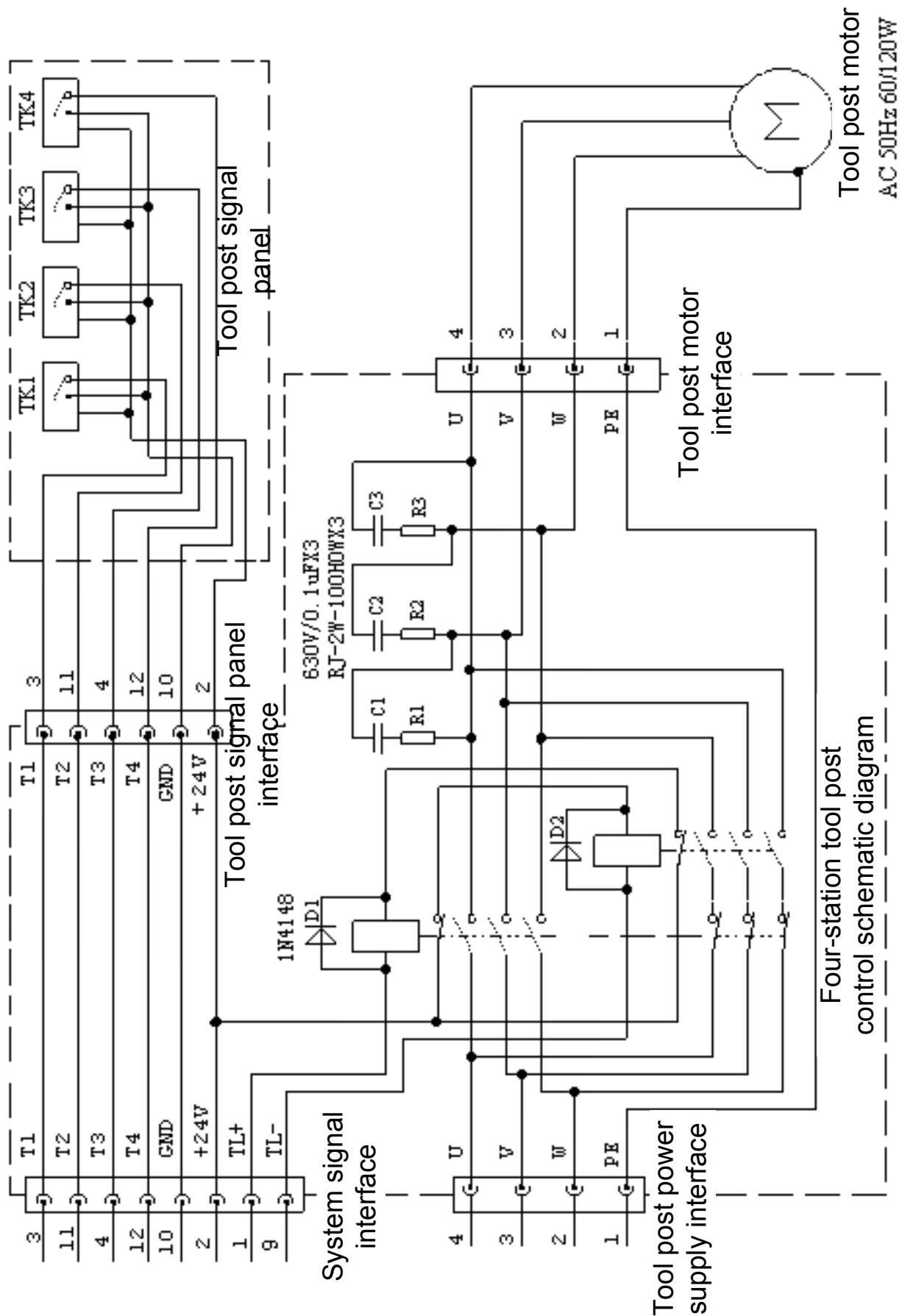


Exchange.



Wiring terminal

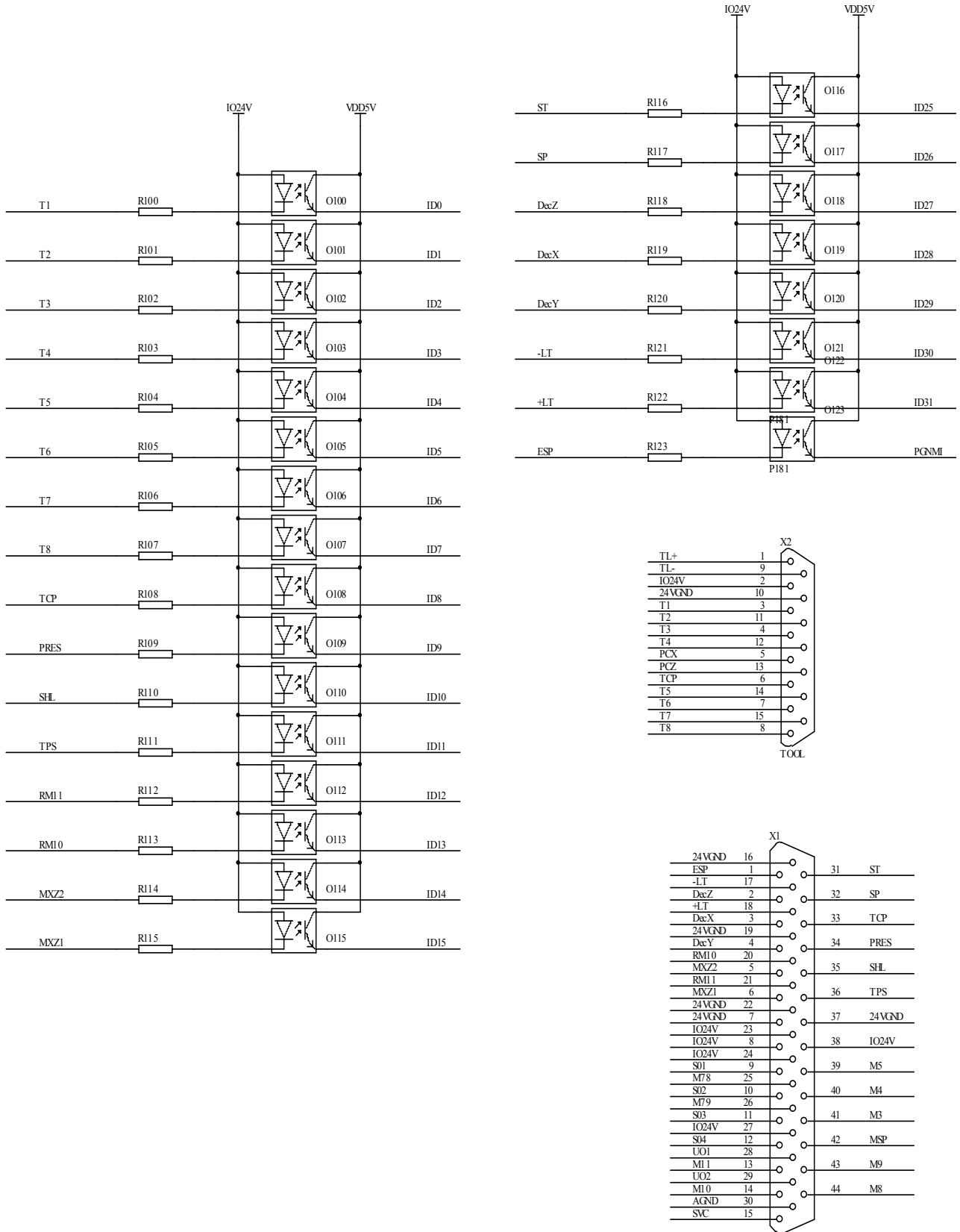
Appendix 2 CNC System Tool Post Controller Circuit Method Layout



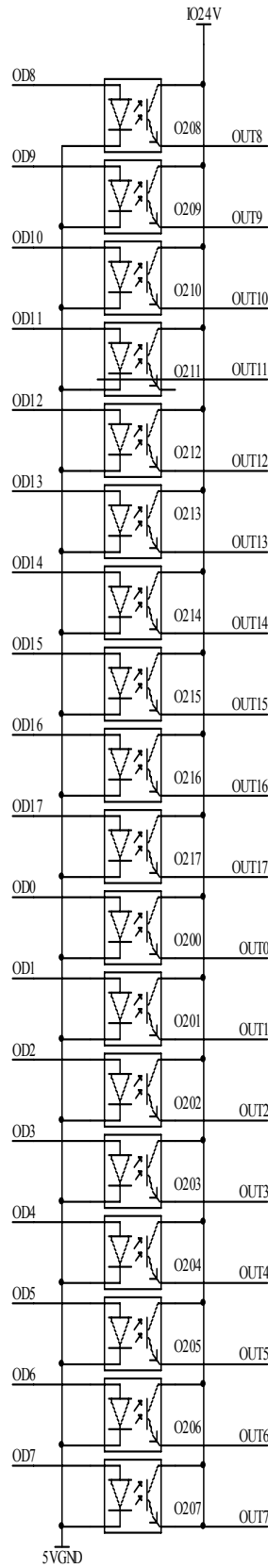
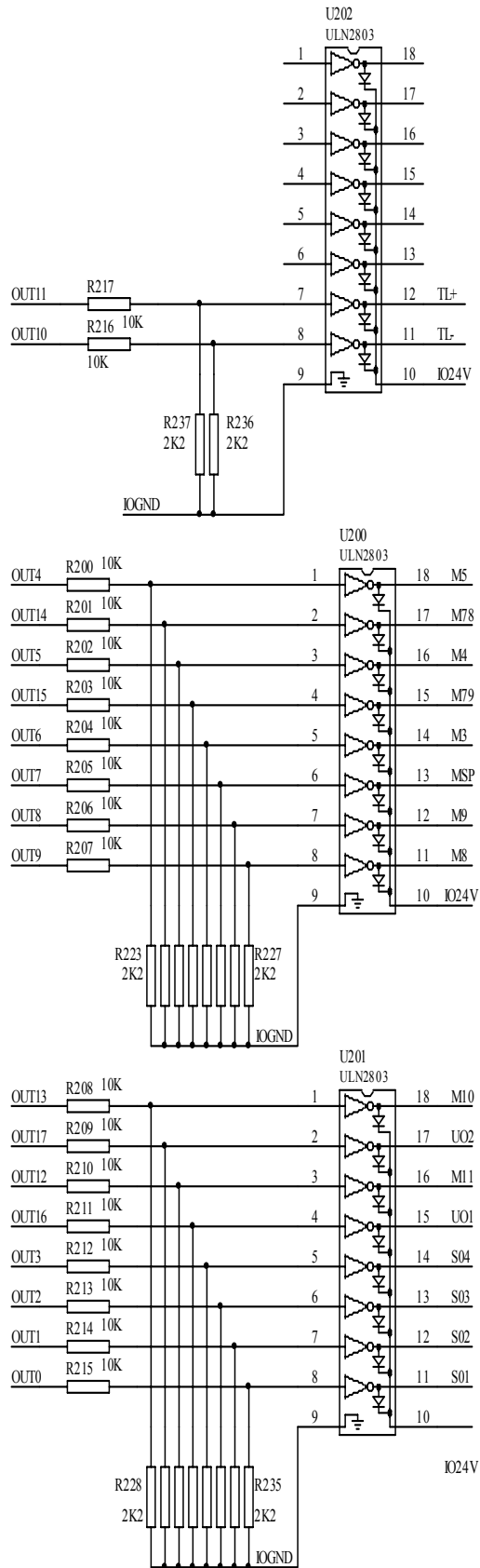


### Appendix 3 Interface Schematic Circuit

Interface schematic circuit 1 (input)

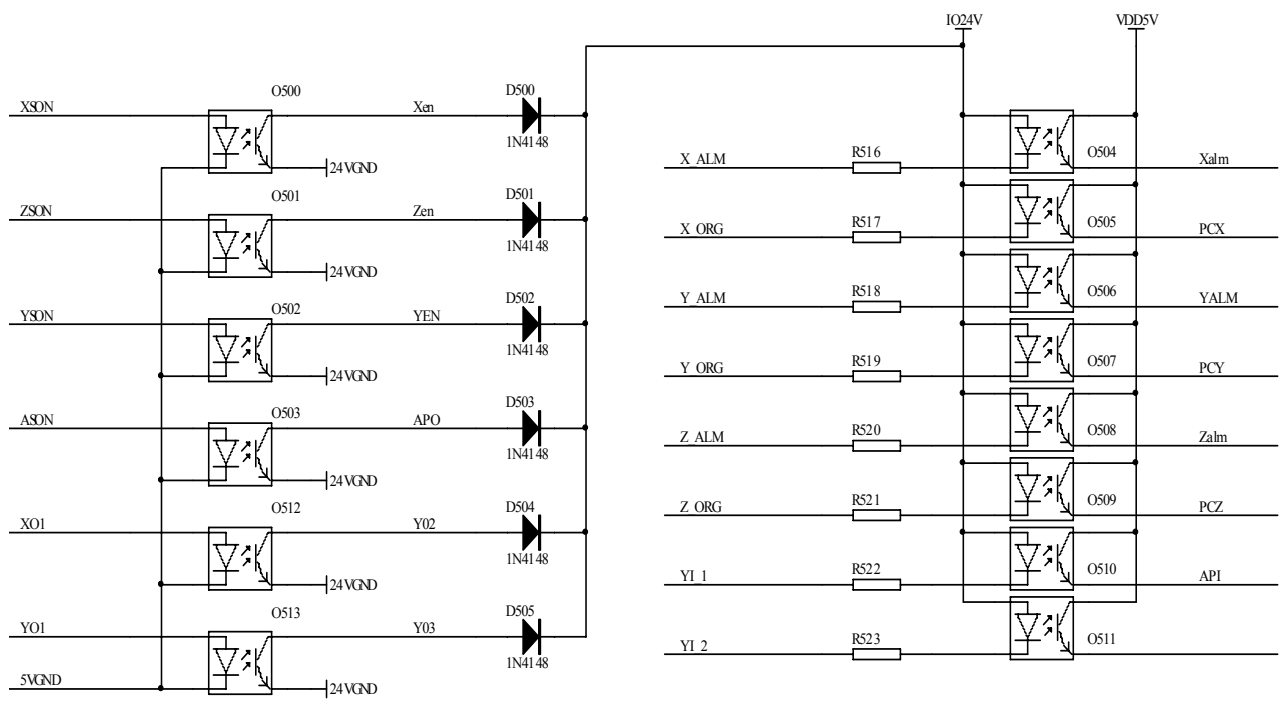
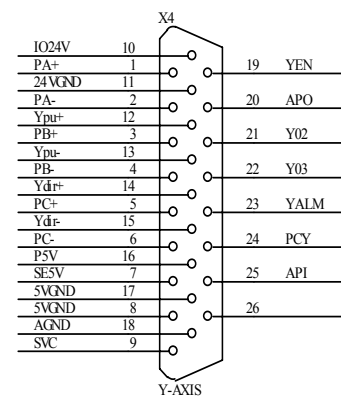
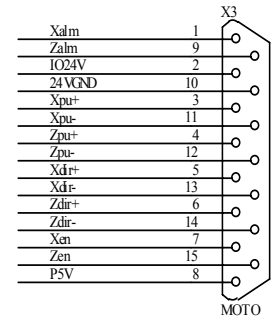
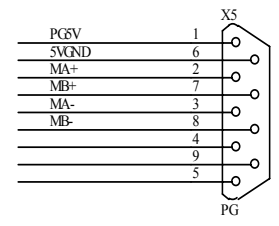
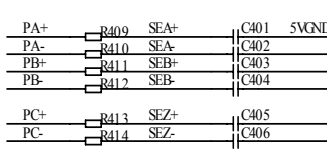
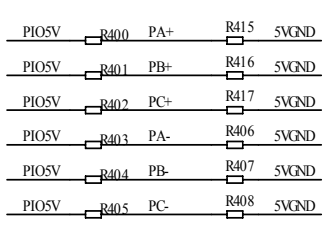
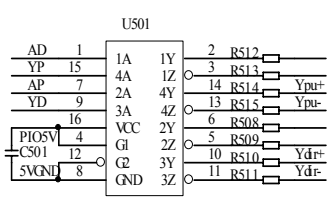
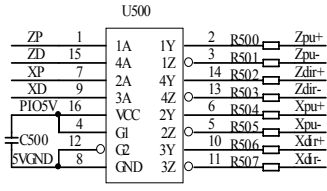
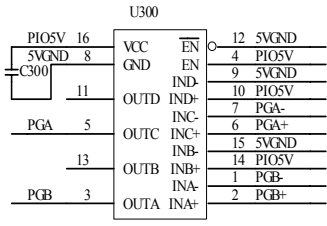
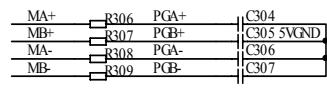
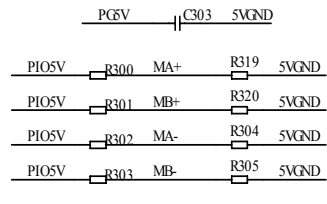


Interface schematic circuit 2 (output)

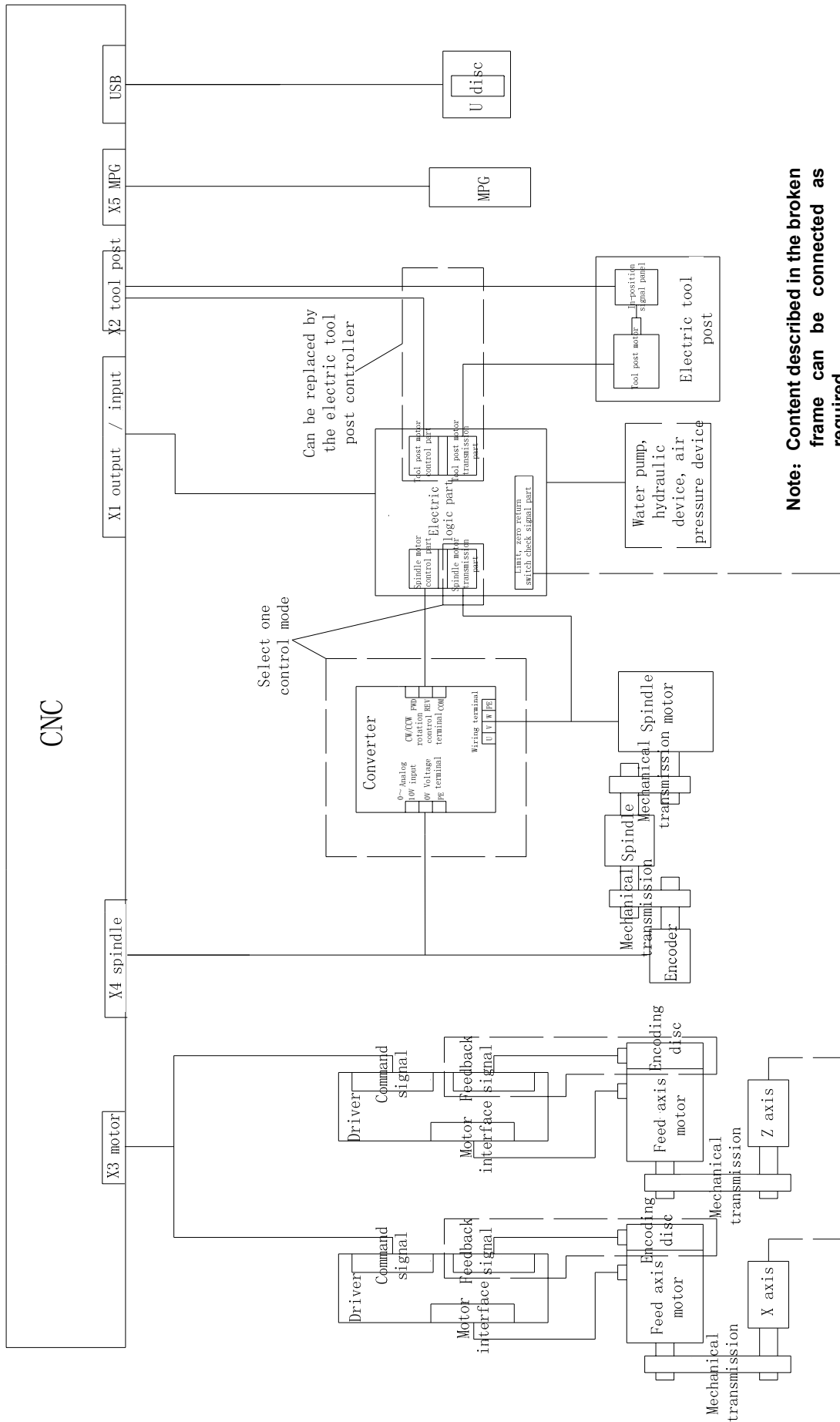


J1		JE	
ID0	A1	B1	ID1
5VGND	C1		
ID2	A2	B2	ID3
5VGND	C2		
ID4	A3	B3	ID5
5VGND	C3		
ID6	A4	B4	ID7
5VGND	C4		
ID8	A5	B5	ID9
5VGND	C5		
ID10	A6	B6	ID11
5VGND	C6		
ID12	A7	B7	ID13
5VGND	C7		
ID14	A8	B8	ID15
5VGND	C8		
OD0	A9	B9	OD1
5VGND	C9		
OD2	A10	B10	OD3
5VGND	C10		
OD4	A11	B11	OD5
5VGND	C11		
OD6	A12	B12	OD7
5VGND	C12		
OD8	A13	B13	OD9
5VGND	C13		
OD10	A14	B14	OD11
5VGND	C14		
OD12	A15	B15	OD13
5VGND	C15		
OD14	A16	B16	OD15
5VGND	C16		
ID16	A17	B17	ID17
5VGND	C17		
ID18	A18	B18	ID19
5VGND	C18		
ID20	A19	B19	ID21
5VGND	C19		
ID22	A20	B20	ID23
5VGND	C20		
ID24	A21	B21	ID25
5VGND	C21		
ID26	A22	B22	ID27
5VGND	C22		
ID28	A23	B23	ID29
5VGND	C23		
ID30	A24	B24	ID31
5VGND	C24		
OD16	A25	B25	OD17
5VGND	C25		
OD18	A26	B26	OD19
5VGND	C26		
OD20	A27	B27	OD21
5VGND	C27		
OD22	A28	B28	OD23
5VGND	C28		
OD24	A29	B29	OD25
5VGND	C29		
OD26	A30	B30	OD27
5VGND	C30		
OD28	A31	B31	OD29
5VGND	C31		
OD30	A32	B32	OD31
5VGND	C32		

Interface schematic circuit 3



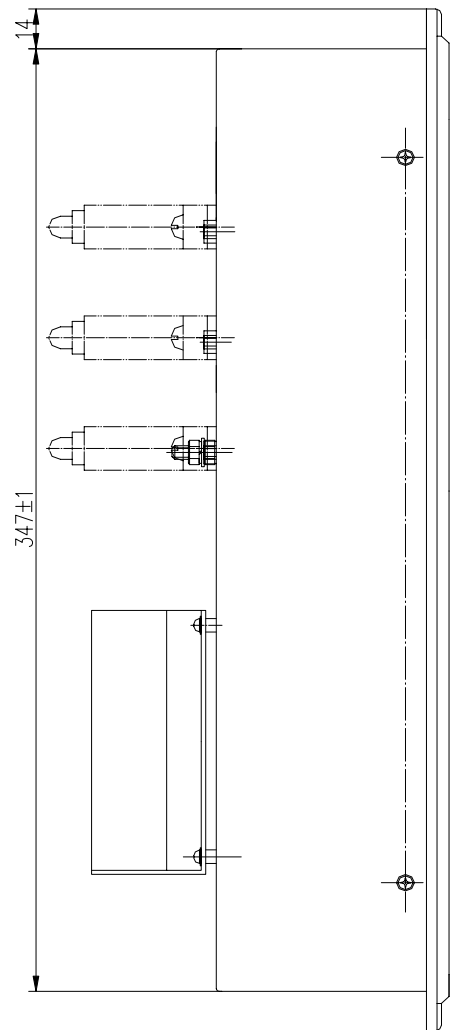
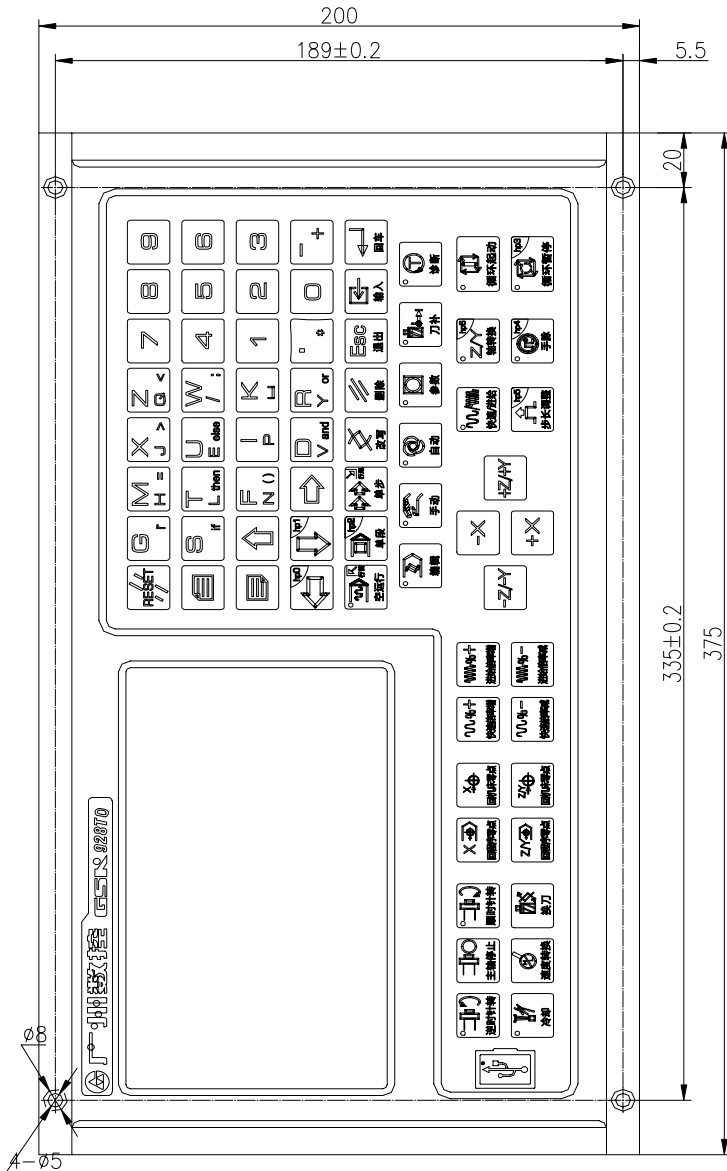
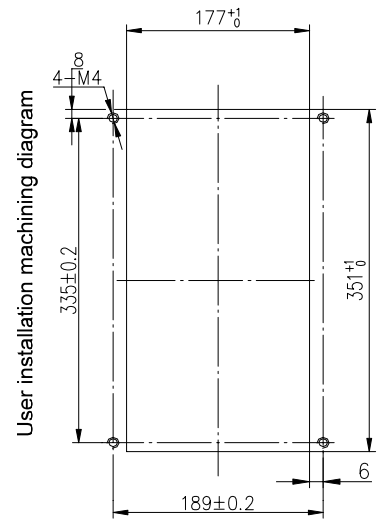
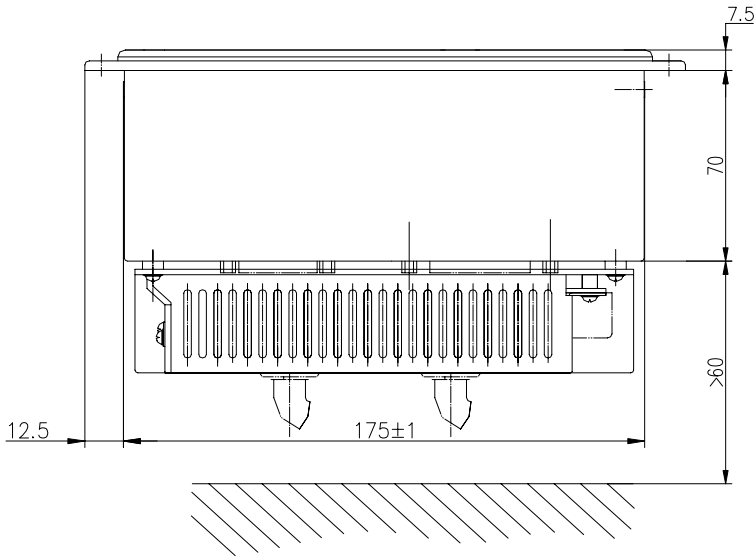
Appendix 4 External Control Connection Diagram



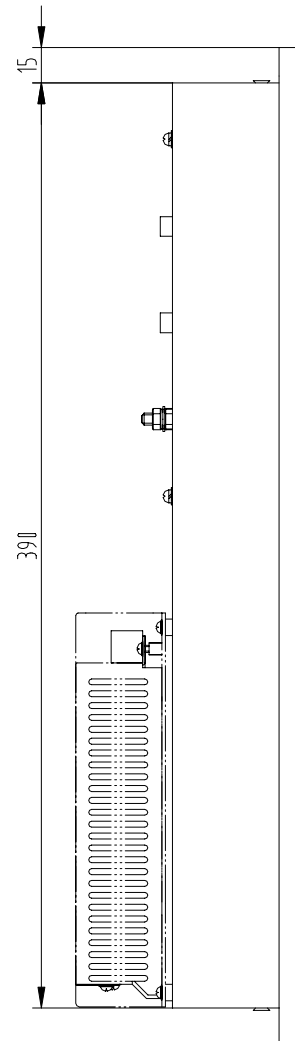
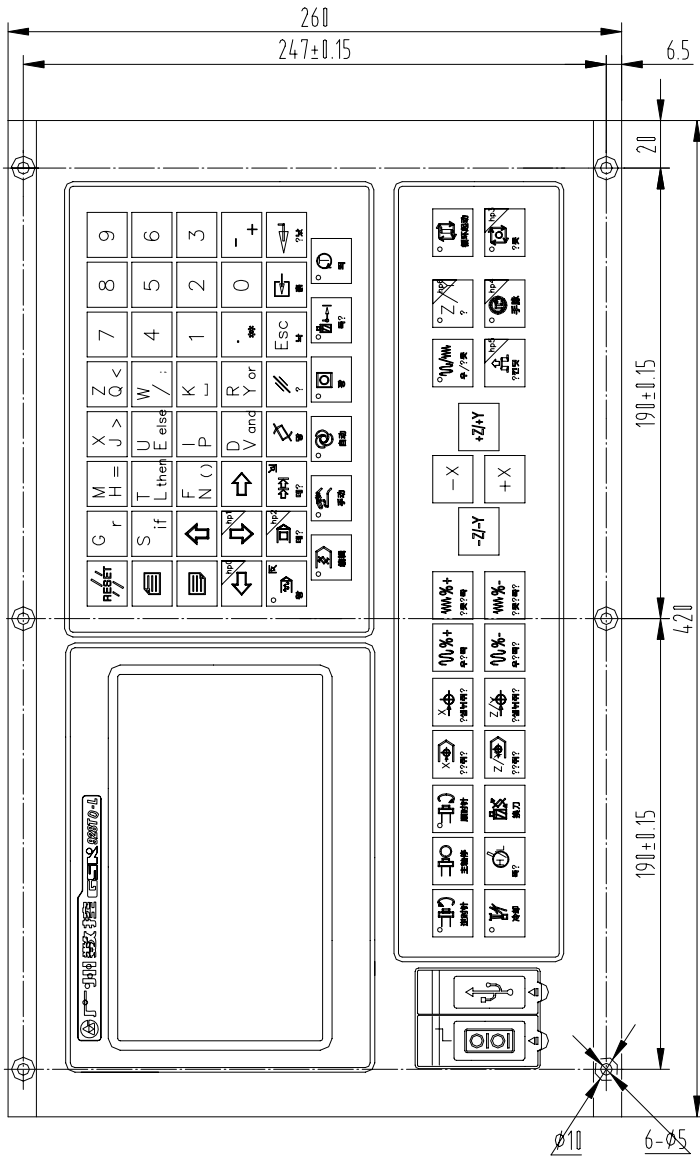
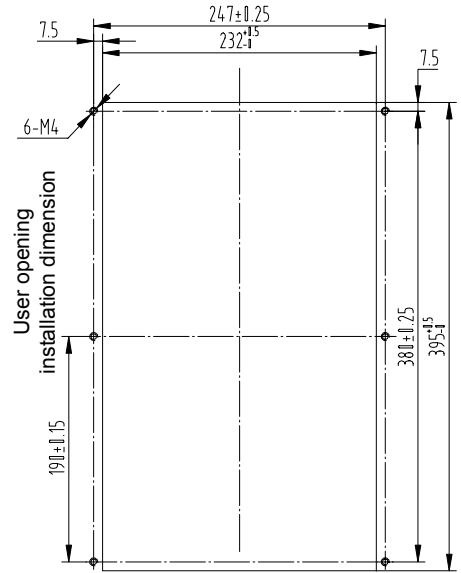
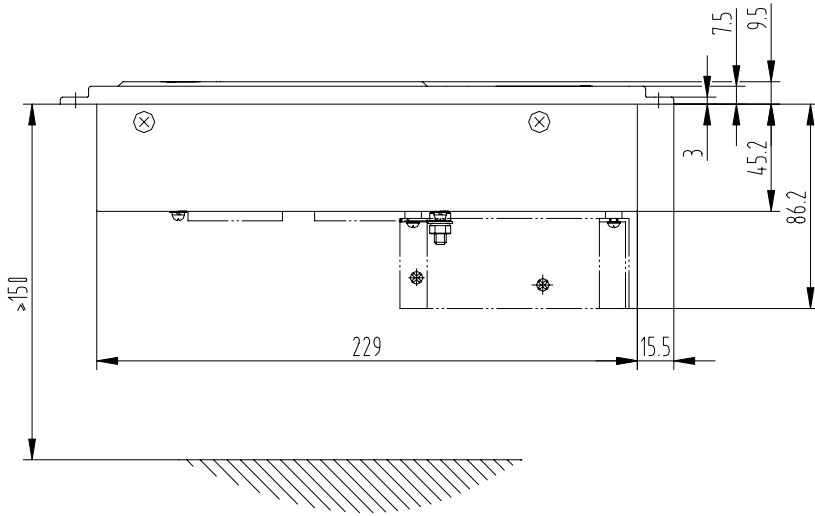
**Note:** Content described in the broken frame can be connected as required.

CNC external control connection

Appendix 5 GSK928TD CNC System Appearance Installation Dimension



Appendix 6 GSK928TD-L CNC System Appearance Installation Dimension



## Supplementary Explanation

for **GSK928TD** and **GSK928TD-L** (V3.21 software version)

(Modification and addition based on edition 1 of *USER MANUAL*)

### 1. Modified Functions and Commands

#### 1.1 Newly-added Interface Parameter P538, P539, P540—Z/X/Y Move Limit

Interface parameter P538, P539, P540—Z/X/Y move limit: (range: 0-32, password level: device administrator)

The setting of this parameter is the same as that of other interface parameters. When this parameter is set to be valid, its corresponding axis can move only when the signal is in the low level; otherwise, its corresponding feed axis is in the unmovable state.

In JOG or AUTO operation mode, if the system detects that the signal is OFF (i.e. “1”), it issues “P538 Z axis move limit”, and the axis cannot move.

#### 1.2 Newly-added Interface Parameter P351 — Alarm of Lubrication Check before Machining

Interface parameter P351— Alarm of Lubrication Check before Machining: (range: 0-32, password level: device administrator)

In AUTO operation mode, if the system detects the signal is ON (i.e. “0”) before executing the first command, it issues “Lubrication alarm detected”, and the machining cannot be performed.

#### 1.3 G76 Command Modification

The previous G76 command only has the function of the tool path cutting-in from the left, while the new G76 command has four cut-in modes, which are specified by H\_d3d2 in programming.

When H is:

Hxxxxx00xx: Cuts in from the left, (the same as the previous G76; Z offsets in the end point direction);

Hxxxxx01xx: Cuts in from the right (Z offsets in the direction opposite to the end point);

Hxxxxx10xx: Cuts in from left to right alternately;

Hxxxxx11xx: Cuts in from the middle (Z does not offset).

## 1.4 Diagnosis Operation Mode

The function of pressing “R” key to detect and diagnose the PA, PB and count signals of the encoder is newly added. This function is used for auxiliary diagnosis when the encoder breaks down. When the spindle is rotating, if PA, PB signal 0/1 appears alternately, and the counter counts recurrently, it means the PA and PB signals are normal.

## 1.5 AUTO Operation Mode

In AUTO operation mode, the “key word” search function is newly added in the block selection execution page. E.g. press the RIGHT direction key to enter the search state, then input “T11” and press ENTER, then press direction key UP and DOWN to search T11, finally press ENTER to confirm that the execution starts from this block.

## 2. Newly-added M Command: M50—M59, M84

### 2.1 Customized Commands Expanded to M50-M74 from M60-M74.

### 2.2 Newly-added Function of Calling M50-M72 before Machining

This system can output signals at four key points, i.e. switch from other pages to AUTO page, switch from AUTO page to other pages, machining start and machining end, for the external monitor.

The method is to solidify the corresponding M command in the No. 254 program, then modify parameter P344-P347, with the number corresponding to the M command; for example, when P334 is modified to 51, the M51 is called.

#### 【Example】

If P334 (call before machining) is set to 50, the system first executes M50 before executing the first machining program each time.

If P345 (call after machining) is set to 51, the systems executes M51 before executing end command M30/M02/M20. If the program is exited halfway, M51 is not executed.

If P346 (entering automatic call) is set to 52 , when there is no alarm occurring, the system will call and execute M52 each time the control is switched to AUTO page from other pages.



If P347 (exiting automatic call) is set to 53, when there is no alarm occurring, the system will call and execute M53 each time the control is switched to the other pages from the AUTO page.

### 【Note】

- 1) If the parameter is set with no corresponding customized command solidified in No. 254 program, this function is skipped.
- 2) After this function is used, the machining efficiency may be affected because of the additional call.

## 2.3 M84 — Input Signal Check within a Specified Time

### 【Command format. Example】

**M84 I8.0** ; If input U108 pin is the low level, the next is executed; otherwise, the system keeps waiting.

**M84 I8.1** ; If the input U108 pin is the high level, the next is executed; otherwise, the system keeps waiting.

**M84 I8.0 D3** ; The system constantly checks the signal within D3, if U108=0, the next is executed; otherwise, the system keeps checking the signal. If the time is up and the signal has not been checked, the system issues “E\*\*\* M84 check I8.0 signal overtime” and terminates the program.

## 3. Newly-added G Commands and Functions

### 3.1 G38 — Rigid Taping, Threading

#### 【Command format】

**G38 Z (W) \_ P (E) \_ H\_** ; Z taping

**G38 Y (V) \_ P (E) \_ H\_** ; Y taping, only for single-axis feed

#### 【Word】

Thereinto: Z (W)、Y (V): Command-specified position which the taping is performed to;

Z/Y is the absolute coordinate of Z axis/X axis respectively, and W/V is the relative coordinate of Z axis/Y axis respectively; the user can employ any one of relative coordinate and absolute coordinate; the relative coordinate is the displacement relative to the current position.

P: Metric thread lead;

E: Inch thread lead;

H: H\_d0=0 indicates M03 tapping; H\_d0=1 indicates M04 tapping.

**【Word range】**

Y, Z, V, W: -9999.999mm~9999.999mm

P: 0.001mm~500.000mm

E: (0.060~25400.000) teeth/inch

H: 00000000~11111111

**【Explanation】****G38 Z tapping cycle execution process:**

- ① If the spindle is rotating, the system executes M05 to exactly stop the spindle.
- ② After entering G38 mode, Z axis always rotates along with the spindle.
- ③ After starting the spindle, the speed increases gradually, and Z axis moves towards to the target position.
- ④ After Z axis reaches to the target position, the spindle speed decreases gradually till it stops.
- ⑤ Then the Z axis retraction is completed by the subsequent command.
- ⑥ Then the system executes the subsequent command, and the spindle rotates in the reverse direction and the speed increases gradually, and Z axis moves in the reverse direction with the spindle.
- ⑦ After Z axis reaches the target position, the spindle speed decreases gradually till it stops
- ⑧ Repeat the above steps until the subsequent command ends, then exit G38 command.

**【Note】**

- 1) For this command, the spindle has a period of deceleration time after it is turned off. At this moment, Z axis/X axis still rotates with the spindle till the spindle completely stops. Therefore, the hole bottom of the thread in the actual machining is deeper than the desired one, and the length depends on the spindle speed and speed change amount during tapping.
- 2) After executing G38 and the subsequent command, the position of the axes is not completely certain.
- 3) The subsequent command of G38 can only be an independent Z(W) word or independent S, M00 command; the next command of the subsequent command can also be the subsequent

command; if the next command following the subsequent command is other command, the system exits G38 mode.

4) The system forbids executing G38 command in Dry Run.

**【Example】**

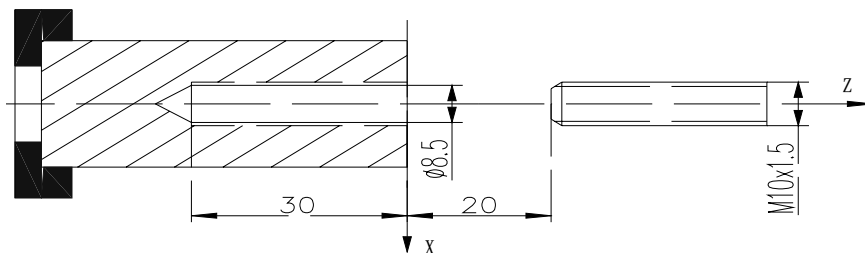


Fig. 4-15 Taping

Example: G38 taping, threading; thread lead 1.5mm.

```

N0000 G00 X0 Z20 ; Rapidly positions the start point of the workpiece
N0010 S300 ; Spindle CW rotation, sets the speed in taping
N0020 G01 Z2 F1000 ; Z approaches the workpiece, and a distance of 2mm is reserved
; at the front end
N0030 G38 Z-30 P1.5 H0 ; Z axis feeds for taping.
N0040 S500 ; Sets the speed at which Z axis retracts
N0050 Z2 ; Z axis retracts
N0060 S300 ;
N0070 Z-30 ; Performs taping again to remove burr
N0080 S500 ;
N0090 Z2 ; Z axis retracts
N0100 M02 ; End of program
    
```

### 3.2 G21, G20— Metric, Inch Input

**【Command format】**

**G20** ; Specifies the least input increment is the inch input.

**G21** ; Specifies the least input increment is the metric input.

G20/G21 command can only check the input unit of the current system and provide alarms rather than switching the input unit. If E285/E286 alarm occurs, please modify P413\_d3.

### 3.3 Detailed Explanation of Metric/Inch Switch

#### 3.3.1 Metric/Inch Switch Parameter

##### 【Relative parameter】

【Interface language】 \_\_bit parameter P413 (password level: 3, machine operator)

d3\_\_ Least input increment

**0: Metric** // The system input increment is the metric input, and the system initialization setting is 0.

**1: Inch** // The input increment of the system is the inch input.

##### 【Function explanation】

If you need to operate the system using the inch, set parameter P413\_d3 to 1, then the system automatically completes the conversion between metric and inch data.

- 1) After the system is switched to inch mode, there is **in** displaying on the upper left of the screen, indicating the current mode is inch. There is no such display for the metric mode.
- 2) System reference conversion formula: 1 inch = 25.4 mm; 1 foot = 12 inches
- 3) The system performs the below conversion at the moment of the metric conversion:
  - ① The system parameters relative to the coordinate position are converted to the inch system, and the unit is inch; e.g. program reference point, soft limit, zero coordinate, backlash, pitch error.
  - ② The system parameters relative to the traverse speed are converted to the inch system, and the unit is inch/min; e.g. max. traverse rate;
  - ③ The current coordinates and speed F are converted to the inch system;
  - ④ The tool offset and tool nose radius compensation in tool offset mode are converted into the inch system;
  - ⑤ If the Y axis is the rotation axis, the position coordinate of Y axis is not switched (When P27 is not 0, the system identifies Y axis as the rotation axis.)
  - ⑥ None of the system parameters relative to the rotation axis is converted.
- 4) In manual traverse or command feed, the actual traverse speed of the rotation axis is Fx10. In F20, the actual traverse speed is 200 degrees/min.
- 5) It is suggested that G20 be added in the first line of the program in the inch system, and G21 be added in the first line of the program in the metric system. If the program is different from the current system state, alarm “E285” occurs. To execute the program, you need to modify the relative parameter. If there is no G20/G21 in the program, the current mode is taken as the default.
- 6) Alarm message—E285/E286: Inch/Metric program, please modify bit d3 of P413. This alarm message indicates the least input increment in the program is inconsistent with that set by the system, and thus you need to modify parameter P413\_d3.

## Supplementary Explanation

### 7) System metric/inch comparison

	<b>Metric unit and range</b>	<b>Inch unit and range</b>	
Coordinate position	0-9999.999 mm	0-999.9999 inch	
Traverse speed	0-99999.999 mm/min	0-9999.9999 inch /min	
F in G99	0-99999.999 mm/rev	0-999.9999 inch /rev	
S in G96	0-9999 m/min	0-9999 foot /min	
Pitch error compensation (without decimal)	1 means 0.001 mm	1 is equivalent to 0.0001 inch	
P in thread commands such as G92, G33	mm/tooth	inch /tooth	
Position of rotation axis	Unit: degree	Unit: degree	
Speed of rotation axis	F1 indicates 1 degree/min	F1 indicates 10 degrees/min	

#### 【Note】

- 1) A position precision loss of 0.002mm may be generated in each metric/inch conversion. It is suggested that the conversion is performed when the coordinate position of the machine is the integer multiple of 2.54mm. When the system is in the inch mode, it is unavailable to switch the system software version.
- 2) In JOG mode, the step width of the rotation axis is 10 times as large as the displayed value.
- 3) If the parameters, tool offset data and program files sent to the PC or USB in the inch state need to be sent back to the system, the system should also receive them in the inch state. I.e. The data should be sent and received in the same mode state.
- 4) After the data are converted to the inch system, the coordinate and speed can be input with four decimals.

#### 【Program example】

The comparative program example of metric/inch system: supposing Y axis is the rotation axis.

<b>/Metric;</b>	<b>/Inch;</b>
%051	%052
G21	G20
N0000 G0 X100 Z100	N0000 G0 X3.937 Z3.937
N0010 T22 M3 S1500 M8	N0010 T22 M3 S1500 M8
N0020 G0 X45 Z3	N0020 G0 X1.7717 Z0.1181
N0030 G01 Z-5 100	N0030 G01 Z-0.1969 F3.9370
N0040 G1 Z1 F300	N0040 G1 Z0.0394 F11.8110
N0050 G01 Y90 <b>F200</b>	N0050 G01 Y90 <b>F20</b>
N0060 G1 Z-5 Y180 F100	N0060 G1 Z-0.1969 Y180 F3.9370

N0070 G1 Z1 F300	N0070 G1 Z0.0394 F11.8110
N0080 G0 Y0	N0080 G0 Y0
N0090 G1 Z-5 F100	N0090 G1 Z-0.1969 F3.9370
N0100 G1 Z1 F300	N0100 G1 Z0.0394 F11.8110
N0110 G0 X100 Z100	N0110 G0 X3.937 Z3.937
N0120 M30	N0120 M30

**CNCmakers**

---

[www.CNCmakers.com](http://www.CNCmakers.com)  
[info@CNCmakers.com](mailto:info@CNCmakers.com)