

Macro Program Instruction

Robot controller M18xx, M28xx for input

M38xx, M48xx for output

For example:

M1800P10, when X00 is effective, then program jump into N10

M3801, output Y01

1. Customer macro instruction M881-M929

(1) M881-M929 correspond to macro program name ProgramUser-ProgramUser49 inside software;

(2) ProgramUser1- ProgramUser49 should be edited on the computer, and copy to disk, then transfer into the controller;

(3) ProgramUser1- ProgramUser49, ProgramTool, ProgramM6 commands:

1) Output instruction: **OUT**

Format: **OUT+(-/A/P/R)Y(M)**+(-/A/P/R)Y(M)**+(-/A/P/R)Y(M)****

Description: + means effective output;

- means ineffective output;

A mans reverse output status(exchange effective and ineffective)

P means judges turret nearby, if CW, then output is effective; if CCW, then output is ineffective;

R means judges turret nearby, if CW, then output is ineffective; if CCW, then output is effective.

Example: **OUT+Y5-Y7+Y9+Y11-Y15**

Output Y5, Y9, Y11 effective; output Y7, Y15 ineffective;

Example: **OUT+M12-M13+Y14+Y8-Y16**

Output M12, Y14, Y8 effective; output M13, Y16 ineffective;

2) Output instruction: **WHEN**OUT**

Used to output signal after judging tool number.

Format: **WHEN**OUT+(-/A/P/R)Y(M)**+(-/A/P/R)Y(M)**+(-/A/P/R)Y(M)****

Description: ** in **WHEN**OUT** refers to tool number.

+ means effective output;

- means ineffective output;

A mans reverse output status(exchange effective and ineffective)

P means judges turret nearby, if CW, then output is effective; if CCW, then output is ineffective;

R means judges turret nearby, if CW, then output is ineffective; if CCW, then output is effective.

Example: **WHEN7OUT +Y5-Y7+Y9+Y11-Y15**

When change into No.7 tool, Y5,Y9,Y11 output are effective, Y7,Y15 output are ineffective.

Example: **WHEN8OUT+M12-M13+Y14+Y8-Y16**

When change into No.8 tool, M12, M14, Y8 output are effective, and M13, Y16 output are ineffective.

3) Wait instruction: **WAT**

Description: waiting for effective or ineffective status of X,Y,M.

Format: **WAT+(-)X**+(-)X**+(-)X**+(-)X**+(-)X**+(-)X****

Description: + means waiting for effectiveness;

- means waiting for ineffectiveness;

Y or M can be zero or one at most, while X is unlimited;

3.1 the maximum waiting time instruction: **MAXWAT**

Used to limit the maximum waiting time of WAT instruction

Format: **MAXWAT******

Description:

the range of number behind **MAXWAT** is 0-99999, unit is ms;

System will alarm and quit when over time;

When assignment as 0 or sole **MAXWAT**, it means ineffective, namely unnecessary to limit maximum waiting time.

3.2 the holding time when meet WAT condition: **HOLDWAT******

Used to specify the holding time when meet condition of WAT

Format: **HOLDWAT******

Description:

the range of number behind **HOLDWAT****** is 0-99999, unit is ms;

System will alarm and quit when over time;

When assignment is 0 or sole HOLDWAT****, it means the function is ineffective, namely unnecessary to limit maximum holding time.

3.3 the maximum waiting time instruction mode: **MODWAT**

MODWAT used to couple with MAXWAT.

Format:**MODWAT1/MODWAT2/MODWAT3/MODWAT4**

Default: **MODWAT1**

MODWAT1: over WAT time, system will prompt alarm and quit;

MODWAT2: over WAT time, system will prompt alarm and
keep running program;

MODWAT3: over WAT time, system will not prompt alarm or quit;

MODWAT4: over WAT time, system will not prompt alarm, but
keeping running program.

Example: detect X0 signal for 5 seconds, if failed then alarm No.1.

MODWAT4

MAXWAT5000

WAT+X0

IF (-X0) THEN

OUT+M81

ERREXIT

ENDIF

4) delay command: **PAUS**

Instructions: for pausing during the program, the unit is MS (MS);

Format: PAUS***

Example: PAUS100

delay for 100ms;

5) suspend command: **M36**

Used for tool changing in segments, when tool change program reach M36, it will pause till press Start again

6) assignment instruction : =

Used to assign values to variables

Format: =

Example: #251=890.34

#450=#123

Available for mathematical expressions, for example:

#440=#234+#670

7) refresh tool status display: **DISP**

Used to refresh tool status display in main interface.

Format: DISP

8) save tool number: **SAVETOOL**

Used to save tool number when tool number does not depends on external switch, normally works in the end of program

Format: SAVETOOL

9) wait for tool No. changing ready: **CASE**

Waiting for certain tool No. changing ready, and judge the defined condition. When condition is met, it means change tool ready, program goes continually.

Format: CASE T** +(-)X**+(-)X**+(-)X**+(-)X**+(-)X**+(-)X**+(-)X**

T** means tool number;

“+” means valid input;

“-” means invalid input;

“X” means input signal.

Example: CASE T1-X16+X17-X18-X19+X20

When Tool 1 rotates ready, X16,X18,X19 are invalid,and X17,X20 are valid.

9a) Wait for tool No. Changing into right position: **CASS**

Waiting for the magazine rotate into the position of tool No. And judge the conditions, when conditions are right, then continue the next program

Format:

CASST**+(-)X**+(-)X**+(-)X**+(-)X**+(-)X**+(-)X**+(-)X**

Note:

T** means tool number;

“+” means valid input;

“-” means invalid input;

“X” means input signal.

After T could add #, means the tool number is a macro value.

After X could add #, means the input point is a macro value.

Example:

CASST1-X16+X17-X18-X19+X20

Wait the magazine rotate into T1 position, when into the position, X16、 X18、 X19 signal ineffective, X17、 X20 effective

Note: when the spindle tool number is the same as the current tool number in the tool table, then controller will ignore the CASS

10) Information prompt dialog

Format: **MESSAGEBOX**(Information string)

or **MSG**(Information string);

11) current tool changing status display

Format: **STATUSINFO**(Information string)

or **STAF**(Information string)

Notice: when just insert STATUSINFO or STAF or STATUSINFO() or STAF(), it means disable the status display.

12) choose tool on tool disk

Format: **CHOOSET(P1,P2)**; p1=1 means choose tool in umbrella magazine, P2 is tool number.

TS0 means tool number in spindle; **TAIM** means the aim tool

13) Condition statement: if the condition is met, then continue to execute, otherwise skip.

Format:

IF(auxiliary relay or input or output or macro variables) **THEN**

....

ENDIF

+Mxx: condition meet when auxiliary relay is effective;

-Mxx: condition meet when auxiliary relay is ineffective;

+Xxx: condition meet when input point is effective;

-Xxx: condition meet when input point is ineffective;

+Yxx: condition meet when output point is effective;

-Yxx: condition meet when output point is ineffective;

+#xx: condition meet when macro variable is 1(non-0);

-#xx: condition meet when macro variable is 0.

Example 1: Pop up dialog box and exit when the M13 is valid.

IF (+M13) **THEN**

MESSAGEBOX(Error: Tool in seat already or air pressure alarm, can not back tool!)

RETURN;// return;

ENDIF

Example 2: Pop up dialog box and exit when the X13 is valid.

IF (+X13) **THEN**

MESSAGEBOX(Error: Tool in seat already or air pressure alarm, can not back tool!)

RETURN;// return;

ENDIF

Example 3: Pop up dialog box and exit when the Y13 is valid.

IF (+Y13) **THEN**

MESSAGEBOX(Error: Tool in seat already or air pressure alarm, can not back tool!)

RETURN;// return;

ENDIF

Example 4: When the macro variables #313=1(not 0), Pop up dialog box and exit.

IF (+#313) **THEN**

MESSAGEBOX(Error: Tool in seat already or air pressure alarm, can not back tool!)

RETURN;// return;

ENDIF

Additional condition command:

1> **IF** [logic expression] **THEN**

.....

.....

ENDIF

- 2> IF [logic expression] THEN #345=235(assignment command)
 3> IF [logic expression] THEN OUT+Yxx-Yxx(output command)
 4> IF [logic expression] THEN GOTO xxx(skip command)

Notice: [logic expression] has the common format with macro program, available for GT/NE/LE logical judgement and mathematical expression.

14) Moving coordinate axis

Format: **MOVE(G,F,X,Y,Z,A,B,W);**

The first parameter **G** is G90 or G91, which is used to define relative or absolute.

F specifies speed, **XYZAB** specifies machine coordinates.

W specifies condition signal which is valid, stop running.

for example: W+5 means if X5 is valid, then stop running.

Example :

MOVE(G90,F8000,X-100);// The X axis moves to the machine coordinate X-100 mm

MOVE(G91,F8000,Z-100);// Z axis moves -100 mm.

MOVE(G91,F8000,Y-300,W-8);// Y axis moves -300 mm, but stops running when the input point X8 is invalid.

Parameters F and XYZAB can be defined value through macro variables.

Example: MOVE(G91,F#231,Y#240,W+2);// Y axis moves -#240 mm in the speed of #231, but when X2 is valid, then stop running.

Notice: if the axis is rotary, then run according to rotary rules, like proximity principle and coordinate reduction.

14a)MOVE and MOVL execute the error, then program will jump into this line, format: MOVJUMP

If program not have MOVJUMP, then will stop when program with error

14b)Robot controller, using MOVJ and speed to reach the position number

Format:MJ***

Example: MJ17 means using MOVJ and speed to reach to the position that set by position No.17.

15) Set the value of the current workpiece coordinate system and save it.

Format: **SETWK(X,Y,Z,A,B);**

Example: SETWK(X23.56,Z567.89);// Set the X of the current workpiece coordinate system is 23.56 mm, Z is 567.89 mm.

Parameter XYZAB can be defined value through macro variables.

Example: SETWK(X#238,Z#237);// Setting the X of the current workpiece coordinate system is #238, Z coordinate value is #237

16) Set the offset value of the current workpiece coordinate system and save it.

Format: **SETWF(X,Y,Z,A,B);**

Example: SETWF(X23.56,Z567.89);// Set X offset of the current workpiece coordinate system is 23.56 mm, Z offset is 567.89 mm.

Parameter XYZAB can be defined value through macro variables.

Example : SETWF(X#238,Z#237);// Setting the X offset of the current work piece coordinate system is #238, Z coordinate offset value is #237.

17) Set the value of the current machine coordinate system and save it.

Format: **SETMH(X,Y,Z,A,B);**

Example: SETMH(X23.56,Z567.89);// Set X value of the current machine coordinates as 23.56 mm, Z value as 567.89 mm.

Parameter XYZAB can be defined through macro variables

Example: SETMH(X#238,Z#237);// Set X value of the current machine coordinates as #238 mm, Z value as #237 mm.

18) Tool count command: **COUN**

The instruction used to wait tool changing ready through a tool counting signal, and judge the defined tool positioning signal, if condition is met, tool change is ready, then continue the program.

Format: COUNT**+(-)X**+(-)X**

Note: T** means tool number, TT means aim tool.

The first input signal X means tool counting signal, “+” rising-edge tool counting, “-” means falling-edge tool counting.

The second signal X defines position signal, “+” means valid position, “-” means invalid position.

The two signals can be the same. Or omission of the second signal is acceptable.

Increasing or decreasing of tool number depends on Tool Parameter “Tool selection”, if single direction, then tool number increase. If bidirectional, then judge tool number increasing or decreasing as the principle of proximity. Change Parameter “tool number depends on input point” into disable.

Example:

COUNTT+X11-X11 choose aim tool in T command, tool counting signal is X11, rising-edge tool counting, positioning signal is X11, invalid means position ready.

COUNTT+X10 choose aim tool in T command, tool counting signal is X10, rising-edge tool counting, not detect positioning signal.

COUNT6+X10-X10 choose Tool No.6, tool counting signal is X10, rising-edge tool counting, positioning signal is X10, invalid means position ready.

COUNT7-X6+X8 choose Tool No.7, tool counting signal is X6, falling-edge tool counting, positioning signal is X8, valid means position ready.

Notice: this tool counting command is just available to low-speed rotary magazine.

19) counting tool and output signal: **CUNOUT**

Output magazine rotating command and waiting for tool changing ready, then automatically close the magazine rotary output signal in parameter.

Format: CUNOUT+(-/A/P/R)Y**+(-/A/P/R)Y**+(-/A/P/R)Y**+(-/A/P/R)Y**

Note:

+ means effective output;

- means ineffective output;

A means reverse output status(exchange effective and ineffective)

P means judges turret nearby, if CW, then output is effective; if CCW, then output is ineffective;

R means judges turret nearby, if CW, then output is ineffective; if CCW, then output is effective.

Example: CUNOUT+Y9PY5RY7+Y11-Y15

The parameters need setting as follows:

32,Tool selection/number tool signal

25, Set up T08 signal [rise"1000+input number",drop"2000+input number"]

26, Set up M61 signal [1000+output number]

27, Set up M63 signal [1000+output number]

28, Set up M71 signal [1000+output number]

Notice: this tool counting command is available to high-speed rotary magazine.

20) Jump statement: **GOTO**

Format: GOTO xx

Example:

OUT-Y8

N12; mark needs to take up a line

OUT+Y4-Y6

PAUS5000

WAT+X23

GOTO 12; jump to N12

21) save macro variable into memorizer: **SAVEMACR**

Format: SAVEMACR

22) Error and exit,

Format:**ERREXIT**

23) Succeed and exit,

Format:**RETURN**

24) Special variable definition:

CURTS: current tool seat No.;

CURTH: current tool length compensation No.;

TAIM: aim tool No., input through T command;

TS(xx): Tool No. In tool seat xx;

TS(0) or **TS0**: tool No. on spindle.

25) Pause command: **M00** or **M0**

Used to pause normal program, program will pause at M00 line, only if press Start button again, can program continue again.

Format: M00 or M0

26) Set current tool length compensation value and save

Format: **SETTH(X,Y,Z,A,B)**

Example: SETTH(X23.56,Z567.89); //set current tool compensation X value as 23.56mm, Z value as 567.89mm.

Parameter XYZAB can be defined value through macro variable

Example: SETTH(X#238,Z#237);// set current tool compensation X value as #238mm, Z value as #237mm

27) Set current tool radius compensation value and save: **SETTD(Parameter)**

28) file operation command:

1> found and open a file:

FILEON(parameter) or **FILEON[parameter]**

Example:

FILEON(AABBCC) or **FILEON[AABBCC]**

means found a file named AABBCC and open it.

2> close file

FILECE means close the currently opened file. If without this, the opened file will be closed automatically after program finished.

3> write word or string into the opened

FILEWD(parameter) or **FILEWD[parameter]**

Example:

FILEWD(G54G0X0Z0) or **FILEWD[G54G0X0Z0]**

means write word string G54G0X0Z0 into the opened file.

4> Write current feed axis absolute coordinate into the opened file: **FILEWC**

5> Read power-off fully protected macro variables from a file:

FILEMR(parameter) or **FILEMR[parameter]**

Example:

FILEMR(AABB)

Read power-off fully protected macro variables from file AABB

6> Write the fully protected macro variables when power-off into a file

FILEMW(parameter) or **FILEMW[parameter]**

7> Save all the current macro variables with user program format into a file

FILEMS(parameter) or **FILEMS[parameter]**

29) Serial communication

1> configure serial check mode.

Format: COMP1 means odd check;

COMP2 means even check;

COMP3 means no check;

2> clear serial data receiving buffer.

Format: **COMC**

3> Send one or more character from serial port.

Format: **COMS(parameter)**; parameter is a character or more or macro variable(variable value is ASCII code corresponding to character).

Example: COMS(A); means send character A.

COMS(ABCF); means send characters ABCF.

COMS(#560); means send ASCII code that #560 value corresponds to. If #560=65, it will send character A.

COMS(##560); means translate #560 value into ASCII code and send it out. If #560=52.234, it will send string 52,234 out.

4> receive character from serial port

Format: **COMR(parameter)**;

parameter is a macro variable, the variable save the received character value(variable value is ASSCII code correspond to character).

Example: COMR(#561); means assign ASCII code value which corresponds to received character to variable #561. if received character is B, then variable #561 will be assigned as 66.

30) By communication to control

When welding machine start arc or finish arc, the progress could set as user-defined, the welding tech-group from 1-10 to 1-25 all will be ineffective, start arc is controlled by ProgramUser0, the finish arc is controlled by ProgramUser1, could use RS232 or EtherCAT to communicate to start or stop welding machine, the controlling code:

A)CMN or CMNN: configure and start communicate

Format:CMN or CMNN(mode, speed rate and port number, parity check)

Mode: 0 means port number is 0(RS232); 1 means port number is 1(RS485-1), 2 means port number 2 (RS485-2); 6 means Internet TCP; 7 means Internet UDP

Speed: 0=7200; 1=9600; 2=14400; 3=19200; 4=38400; 5=57600; 6=115200

Parity check: 1 or O means odd check; 2 or E even check; the other means no check

The difference between CMNN and CMN: if the communication is already opened, CMNN will be ignored, CMN will close the current communication, and open the communication.

Note:speed rate and parity check only work for serial port, internet no need

Example: CMN[0,6,1] means open serial port 0 communication, the baud rate is 115200, odd check

CMN[0,6,E] means open serial port 0 communication, the baud rate is 115200, even check

CMN[1] Serial port 1 can omit the last two items, the baud rate is 115200, no check

CMN[2,2502] open TCP internet communication, port number is 502

CMN[3,802] open UDP internet communication, port number is 802

B) CMC: clear receive buffer

Format: CMC means clear the buffer of communication that received

C) CMS: send data

Format: CMS[xxxxxxxx%] The data is in hexadecimal, % means sends the CRC check code (the check code is automatically generated by the system), and the data can be separated by commas or other characters (except %)

Example:CMS[03AC67D704%], send: 0x03, 0xAC, 0x67, 0xD7, 0x04, check code;

CMS[03AC67D704], send: 0x03, 0xAC, 0x67, 0xD7, 0x04;

CMS[03, AC, 67, D7, 04, %], send: 0x03, 0xAC, 0x67, 0xD7, 0x04, check code;

CMS[03HACH67HD7H04H], send: 0x03, 0xAC, 0x67, 0xD7, 0x04;

D) CMR:Receive data

Format:CMR[%N_L_M_] % means that the CRC code needs to be checked

N: Number of received characters (including CRC check code);

L: Start receiving data after waiting for a period of time [ms];

M: maximum waiting time [ms], if there is no L and M, it means that it is received directly without waiting time;

Receive data: macro variable #800 is the status, #801--#899 is the received data; #800 is 0 means there is no data;

If #800 is a negative number, it means that the check CRC code is wrong (the absolute value of #800 means the number of 8-bit data received);

#800 is a positive number, which means the number of 8-bit data received;

For example: CMR[%N9], receive 9 characters (including CRC check code);

CMR[%N8L50M300], wait for 50ms to receive 8 characters (including CRC check code), if the data is not received, wait up to 300ms;

If receiving data: 01 06 00 04 6F 00 E5 FB; then #800=8, #801=1, #802=6, #803=0,#804=4,#805=111,#806=0,#807=229,#808=251

E) CMM: ModBusRTU data exchange

Format:CMM[P_K_#_R_N_M_]

P: slave device address (hexadecimal);

K: Function code (hexadecimal): 03 to read register; 06 to write single register; 10 to write multiple registers;

#: Master macro variable number (decimal);

R: slave register number (decimal);

N: The number of data read or written (decimal);

M: Maximum waiting time [ms] (decimal), negative table infinite waiting (used during debugging);

If P_ is not programmed, the default station number is 0x1; if K_ is not programmed, the default is 03 function code;

Receive data: macro variable #800 is the status, #801--#899 is the received data; #800 is 0 means there is no data;

If #800 is a negative number, it means that the check CRC code is wrong (the absolute value of #800 means the number of 8-bit data received);

#800 is a positive number, which means the number of 8-bit data received;

For example: CMM[P2K3#500R5N3M-1],

If receive data: 02 03 03 12 13 14 15 15 15 3D 5F; then

#800=11,#801=2,#802=3,#803=3,#804=18,#805=19,#806=20,#807=21,#808=21,#809=21, #810=61,#811=95,#500=4627,#501=5141,#502=5397;

Note: read or write registers are 16-bit data, the system automatically generates CRC check code;

E.g:

<1>CMM[P2K3#500R5M-1], read the value of the No. 5 register of the No. 2 slave station (always waiting to receive data) save to the system #500 macro variable;

Received content: 02<station number>03<read>02<number>0612<data>7FE9<CRC>

<2>CMM[P2K3#500R5N3M-1], read the value of No. 5--7 register of No. 2 slave station and save it to the system

#500--#502 In macro variables;

Received content: 02<station number>03<read>06<number>1213 1415 1515<data>685F<CRC>

<3>CMM[P2K6#700R5], write the value in system #700 to the No. 5 register of No. 2 slave station;

Sending content: 02<station number>06<write>0005<address>1213<data>D495<CRC>

<4>CMM[P2K10#700R5N3], write #700--#702 to No. 5-7 registers of No. 2 slave station;

Sending content: 02<station number>10<write>0005<address>0003<array>06<number>1213 1415 0000 <data>61D6<CRC>;

F) CMF: close communication

Format:CMF means close the communication

Also in the diagnosis interface, press the "2" key to input the above communication command to run immediately, which is convenient for testing

31) It is necessary to power off to save the macro variable command

YESSAVE

When this command is not edited, the default is to save the macro variables when the power is turned off.

31a) Save macro variables without power off

NOTSAVE

After the macro variable is assigned in the control program, it is not necessary to power off and save, and the original value will be restored after the power is turned on again.

32), the command to return to the first reference point

Format: G281 G282 G283 G284 G285 G286 G287 G288 respectively correspond to 8 feed axes to return to zero.

Note: Only one command can be programmed in one line, and one command can only return the zero position of one axis.

33), the command to return the C axis of the lathe to zero

Format: M80 or M800 to realize the zero return of the C axis of the lathe

Note: Only one command can be programmed per line.

34), control the first analog output command

Format: ANAS****

Description: The numerical range after the command is 0-99999, and the unit is rpm (the maximum speed of the parameter corresponds to 10V voltage).

Note: # sign can be programmed, which means to call the corresponding macro variable value.

35), control the second analog output command

Format: ANBS****

Description: The numerical range after the command is 0-99999, and the unit is rpm (the maximum speed of the parameter corresponds to 10V voltage).

Note: # sign can be programmed, which means to call the corresponding macro variable value.

36), read the current coordinates of all absolute encoder feed axis motors

Format: M500

Adding L0 simultaneously clear pulse and feedback pulse:

M500 L0: Read the absolute motor positions of all feed axes and reset the current machine coordinates while clearing the command pulse and feedback pulse.

37), read the current coordinates of a feed axis absolute encoder motor

Format: M501 M502 M503 M504 M505 M506 M507 M508 Read the absolute encoder positions of eight feed axes respectively.

Note: Only one command can be programmed in one line, and one command can only read the absolute encoder position of one axis.

Add L0 to clear the command pulse and feedback pulse at the same time:

M500 L0: Read the absolute motor positions of all feed axes and reset the current machine coordinates while clearing the command pulse and feedback pulse.

M501 L0: Read the absolute motor position of the X axis and reset the current machine tool coordinates while clearing the command pulse and feedback pulse.

M502 L0: Read the absolute motor position of the Y axis and reset the current machine tool coordinates while clearing the command pulse and feedback pulse.

M503 L0: Read the Z-axis absolute motor position and reset the current machine tool coordinates while clearing the command pulse and feedback pulse.

M504 L0: Read the 4-axis absolute motor position and reset the current machine tool coordinates while clearing the command pulse and feedback pulse.

M505 L0: Read the 5-axis absolute motor position and reset the current machine tool coordinates while clearing the command pulse and feedback pulse.

M506 L0: Read the 6-axis absolute motor position and reset the current machine tool coordinates while clearing the command pulse and feedback pulse.

M507 L0: Read the absolute motor position of the 7-axis and reset the current machine tool coordinates while clearing the command pulse and feedback pulse.

M508 L0: Read the 8-axis absolute motor position and reset the current machine tool coordinates while clearing the command pulse and feedback pulse.

38), the machine coordinates are reset to zero

M417 resets the X-axis machine coordinates to zero.

M418 resets the Y-axis machine coordinates to zero.

M419 resets the Z-axis machine coordinates to zero.

M415 resets the A-axis machine coordinates to zero.

M416 resets the machine coordinates of the B axis to zero.

M412 resets the C-axis machine coordinates to zero.

M413 resets the machine coordinate of Xs axis to zero.

M414 resets the machine coordinate of Ys axis to zero.

M420 resets all machine coordinates of XYZABCXsYs axis to zero.

39), execute an absolute motor servo tool magazine axis coordinate reduction setting processing command: M411.

40), control the bus drive motor enable command

M135 X1 Y1 Z1 A1 B1 C1 XS1 YS1 Power-on enable

M135 X0 Y0 Z0 A0 B0 C0 XS0 YS0 disconnect enable

41), wait for the feed axis to stop completely

Wait for all feed axes to stop completely Command: M7-0

Wait for the X axis to stop completely Command: M7-1

Wait for the Y axis to stop completely Command: M7-2

Wait for the Z axis to stop completely Command: M7-3

Wait for the A-axis to stop completely Command: M7-4

Wait for the B axis to stop completely Command: M7-5

Wait for the C axis to stop completely Command: M7-6

Wait for the XS axis to stop completely Command: M7-7

Wait for the YS axis to stop completely Command: M7-8

42), switch the current workpiece coordinate system (mainly used for the automatic tool setting control program with tool setting gauge):

G54,G55,G56,G57,G58,G59,G54.1--48

43), add the PLx. instruction to the control program corresponding to the custom M code to specify the number of palletizing groups, such as PL2.

The following is the control code corresponding to the second group of palletizing functions until the next PLx. to finish.

For example: user-defined M881 corresponds to the control program mode as follows:

PL1.

OUT+....

WAT-....

....

....

PL2.

....

....

PL3.

....

....

PL5.

....

....

Executing M881 means only the part between PL1. and PL2. working, the default is L1.

Executing M881 L2 means only the part between PL2. and PL3. working.

Executing M881 L0 means ignore all PLx, and execute all program

PA3 means only work the part between PL3. and PL.5

PA2 means only work the part between PL2 and PL3

Also could use PLEND to end, example: user-defined M881:

OUT+....

WAT-....

....

....

PL2.

....

....

PLEND

....

....

PL4.

OUT+....

WAT-....

PLEND

....

....

OUT+....

WAT-....

....

....

M881 means except the parts between PL2.and PLEND, PL4 and PLEND, the rest parts all working, the default is L1

M881 L2 means except the part between PL4 and PLEND, the rest parts working

M881 L0 means ignore all PLx. All program working

M881 L4 means except the part between PL2 and PLEND, the rest parts are working

PA3 means except the parts between PL2 and PLEND, PL4 and PLEND, the rest parts working.

PA2 means except the part between PL4 and PLEND, the rest parts working

PA4 means except the part between PL2 and PLEND, the rest parts working

Note:

After format must add “.” like: PL2.

Can't input PL2. as PL02. otherwise controller can't understand

Custom macro program instructions G101-G170, G65, G66, G67

";MARO" The first section of the main program adds this command means the entire program is executed according to the macro program, and the macro variables are updated line by line;

1、 G101-G170 Corresponds to macro program ProgramG101- ProgramG170; G101-G150 are non-model command, G151-G170 are model command;

2、 ProgramG101- ProgramG170 only can be edited in computer, then copy to USB disk, and convert into the controller;

3、 Macro program instruction:

1) Input instruction : **WAT**

Wait for input point X valid or invalid.

Format: **WAT+(-)X**

Explain: “+” means input point is valid;

“-” means input point is invalid;

“X” means input point X00-X55, check it in I/O diagnose;

2) Output instruction: **OUT**

Set the output point be valid or invalid.

Format: **OUT+(-)Y**

Explain: “+” means input point is valid;

“-” means input point is invalid;

“Y” means input point Y00-Y31, check it in I/O diagnose;

3) Variables and assignment: =

Variable can be divided into following 5 types:

(1) #0--#20 Local variable: Local variables can only be used to store data in macro program, such as the result of the operation. when power off, the local variable is initialized to be null. when call macro program, independent variable assigns local variable.

(2) #21--#999 Global variable: global variable has the same meaning in different macro programs.

When power off, the variable #21--#100 will initialize to null, variable #101--#900 will be saved and will not lose even if power off.

Variable #900--#999 can be called as CMP to do comparison operation in PLC.

(3) #1000-- System variable: The system variable is used to read changes of various data during CNC operation, such as current position and the compensation value of tool.

Attention: macro variable #100--#204, #380--#390 have been occupied by the system, unavailable for user.

The global variable available for user: #31--#99, #205--#379 and #391--#999. the rest global variables are occupied or for standby.

Variable #161-#165 are used to record and display XYZAB machine coordinate when program jump upon G31/G311.

#201: repeat times of compound thread G176(Mode 1);

#202: tool offset distance in Z direction each time when repeat G176(unit: mm)(Mode 1);

#203: repeat times of compound thread G176(Mode 2);

#204: tool offset distance in Z direction each time when repeat G176(unit: mm)(Mode 2).

(4) Macro-variable #1001--#1099 correspond to the X axis offset value of T1--T99 in lathe(unit: micron).

Macro-variable #1401--#1499 correspond to the Z axis offset value of T1--T99 in lathe(unit: micron).

And the value can be read, Example: #200=#1003; means read the X axis offset value of Tool No.3 into the macro variable #200.

The value also can be edited, Example: #1003=23000; means change the X axis offset of Tool No.3 into 23000 microns.

#1003=#1003+50; Increase X axis offset of the Tool No.3 by 50 microns.

(5) Input and output variables:

#1800: X00-X07 (D0-D7)

#1801: X08-X15 (D0-D7)

#1802: X16-X23 (D0-D7)

#1802: X16-X23 (D0-D7)

- #1803: X24-X31 (D0-D7)
- #1804: X32-X39 (D0-D7)
- #1805: X40-X47 (D0-D7)
- #1806: X60-X67 (D0-D7)
- #1808: Y00-Y15 (D0-D15)
- #1809: Y16-Y31 (D0-D15)

Format: #i= expression;

4) Arithmetic and logic

Arithmetic and logical operation table:

Function	Format	Remarks
Definition	#i = #j	
Addition	#i = #j + #k ;	
Subtraction	#i = #j - #k ;	
Multiplication	#i = #j * #k ;	
Division	#i = #j / #k ;	
Sine	#i = SIN(#j) ;	Angle unit is degree, 90 degrees 30 minutes is 90.5 degrees.
Arc sine	#i = ASIN(#j);	
Cosine	#i = COS(#j) ;	
Arc cosine	#i = ACOS(#j);	
Tangent	#i = TAN(#j);	
Arc tangent	#i = ATAN(#j);	
Absolute Square	#i = SQRT(#j);	
Absolute value	#i = ABS(#j) ;	
Round off	#i= ROUND(#j);	
Round down	#i = FIX(#j);	
Round up	#i = FUP(#j);	
Natural logarithm	#i = LN(#j);	
Exponential	#i = EXP(#j);	
Or	#i = #j OR #k ;	Logical operation works one by one bit with binary number
Exclusive or	#i = #j XOR #k ;	
And	#i = #j AND #k ;	

5) Unconditional Jump: **GOTO n**

Transfer to the program segment marked with the order number n, when specified number except for 1 to 99999, it will alarm error, order also can be defined by expression.

Example: GOTO 5, GOTO #100

6) Conditional Jump:

- 1) **IF** (conditional expression) **GOTO OR THEN**
- 2) **IF** (conditional expression)
 - < operation >
 - ELSE**
 - < operation >
 - ENDIF**
- 3) **IF** (conditional expression)

< operation >

ELIF

< operation >

ELIF

< operation >

ENDIF

If the specified conditional expression is met, then execute following statement; if the specified condition expression is not met, then execute next section of the program.

Example: IF (#100 EQ 2) THEN #100=5
IF (#101 GT 2) GOTO 6

Operator character meaning:

EQ equal to =

NE not equal to

GT greater than >

GE greater than or equal to >=

LT less than <

LE less than or equal to <=

7) Loop statement: **WHILE (conditional expression) DO 1、 2、 3**

Specifying a conditional expression behind WHILE. When condition is met, execute program between DO and END. Otherwise, skip to segment after END.

The loop can be nested to 3 layers at most.

Example: WHILE (#100 LT 3) DO 1
.....
WHILE (#103 EQ 5) DO 2
.....
WHILE (#200 GE 20) DO 3
.....
END 3
.....
END 2
.....
END 1

8) Non-model call macro instruction: **G65**

Format: G65 P- L- <A-B-C-.....Independent variable passing data>

P is Macro program name, L is Call times, A、 B、 C are independent variable, name of independent variable are as follows: #0->A、 #1->B、 #2->C、 #3->D、 #4->E、 #5->F、 #6->H、 #7->I、 #8->J、 #9->K、 #10->M、 #11->Q、 #12->R、 #13->S、 #14->T、 #15->U、 #16->V、 #17->W、 #18->X、 #19->Y、 #20->Z.

Attention: Address G、 L、 N、 Q、 P Can not be used in the independent variable.

Example:

Main program: 9000

```
G00 X0 Z0
G65 P8000 L1 A5 B6
G0 X0 Z0
M30
```

Macro program: 8000

```
N1 #2=#0+#1
N2 IF (#2 EQ 10) GOTO 4
N3 GOO X#2
N4 G00 Z#1
N5 M99 ;return
```

9) Modal call macro program: **G66**、**G67**

G66 is used to call modal macro, G67 is used to cancel mode. the same format with G65.

Example:

```
Main program: 9000
G00 X0 Z0
G66 P8000 L2 A5 B6
A8 B1
A9 B10
G67
M30
Macro program: 8000
N1 #2=#0+#1
N2 IF (#2 EQ 10) GOTO 4
N3 GOO X#2
N4 G00 Z#1
N5 M99 ; return
```

10) Custom macro instruction (**G120—G160**, **M881—M929**)

Each custom G code corresponds to a macro program ProgramGxxx, M Code corresponding to a macro program ProgramUser1-ProgramUser49, user can not edit macro program in CNC system, it MUST be edited in PC, and install into controller through USB disc.

For example, the definition of G152 Function: circular hole drilling loop. (must install the macro program ProgramG152 into controller).

Format: G152 Xx Yy Zz Rr Ii Aa Bb Hh Ff;

X: The X coordinates of center specified with absolute or incremental value;

Y: The Y coordinates of the center specified with absolute or incremental value;

Z: Hole depth;

R: Rapid approaching point coordinates;

F: Feed rate;

I: Circle radius;

A: Angle of the first hole

B: Increment angle(negative is clockwise).

H: hole count ;

Macro program ProgramG152 contents are as follows:

```
#80=#0
#81=#1
#82=#2
#83=#3
#84=#4
#85=#5
#86=#6
#87=#7
#88=#8
#89=#9
#90=#10
#91=#11
#92=#12
#93=#13
#94=#14
#95=#15
#96=#16
#97=#17
#98=#18
#99=#19
#100=#20
#30=#4003
#31=#4014
G90
IF[#30 EQ 90] GOTO 1
G53
#98=#5001+#98
#99=#5002+#99
N1 WHILE[#86 GT 0] DO 1
#35=#98+#87*COS[#80]
#36=#99+#87*SIN[#80]
G81X#35Y#36Z#100R#92F#85
#80=#80+#81
#86=#86-1
END 1
G#30 G#31 G80
M99
```

2. Custom macro variable configuration dialog box interface

File **interfacecn.cfg**(Chinese) and **interfaceeg.cfg**(English) are used to configure the system display interface.

Its contents are as follows:

- 1) When the first character of a line is "/", means this is a comment line.
- 2) The first line specifies the system status at boot time: [the line is optional]
M means manual continuous;

I means manual increment;

H means Hand wheel;

A Automatic continuous;

a Automatic single segment.

The value 1 or 2 or 3 after the command means the controller main interface status when power on: 1/2/3 means Fa/Fb/Fc interface

Example:

IniStatus=M means Boot into manual continuous status.

IniStatus=A2 means Boot into the automatic continuous status and show the interface Fb

3) Pressing Enter button in main interface, it will pop up the dialog box which will be displayed in main interface (including the general coordinates and the current program display area)

Other parameter **P900" user-defined dialog box** [4 is part, 8 is all,] set as 4 means effective

The dialog is defined as follows:

Begin with BeginMainFace and end with EndMainFace. [this line is optional]

BeginMainFace means start configuration.

0, 20, 80, 420, 300, please input parameters <F1-clear, F8-clear> Four data represent dialog box l/t/w/h, followed by dialog box title prompt information.

4,3,3,280,190,fillpict1.bmp (fillpict1.bmp is the name of the image file)

1,282,10,60,25,X: The first data means control type, the next four data means l/t/w/h, and the rest means the macro variable number (only for edit box) or prompt information.

3,330,10,85,25,345,I corresponds to the #345 macro variable and is an integer (I specifies an integer, if not specified, it is a floating-point type)

1,282,40,60,25,Z:

3,330,40,85,25,288, corresponding to #288 macro variable, and is a floating point type

1,282,70,60,25,R:

3,330,70,85,25,500,I

1,282,100,60,25,P:

3,330,100,85,25,700,

1,282,130,60,25,S:

3,330,130,85,25,456,

1,200,195,100,25, Lead K(E):

3,300,195,90,25,145,

1,200,223,100,25, times L:

3,300,223,90,25,225,

2,30,253,100,26,Enter

2,240,253,100,26,Cancel

EndMainFace indicates the end of the configuration.

Control type:

0: Dialog box itself;

1: User-defined static text;

2: User-defined button;

3: User-defined edit box;

4: Picture ;

8: User-defined small font static text;

5 is the static text of MiniGUI; 6 is the button of MiniGUI; 7 is the edit box of MiniGUI; 9 is the custom edit box that can be automatically and dynamically refreshed.

[Note: void when the control type is 5,6,7].

Picture file name: fillpict1, fillpict2, fillpict3 And must be

16 color (4 bits) or 256 color (8 bit) or 16 or 24 bit BMP format picture.

However, attention should be paid to the use of 16 color (4 bits) or 256 color (8 bit) picture, in order to reduce the image space.

Better to use 16 color or 256 color picture to reduce the storage

In the main interface, press Enter button will pop up the dialog box is also defined the same.

For the custom edit box: when the corresponding macro variable number is greater than or equal to 20000, it means that the current coordinates or other status information is displayed. Currently, it is defined as follows:

20001/20002/20003/20004/20005 means XYZ45 axis machine coordinates.

20011/20012/20013/20014/20015 are the workpiece coordinates or absolute coordinates of XYZ45 axes respectively.

20201 means the current spindle status, displaying "CW", "CCW", "STOP" information.

For custom edit boxes that automatically refresh dynamically:

When the corresponding macro variable number is greater than or equal to 40000 and less than 40999, it means the corresponding input point X number plus the value of 40000, for example, 40013 corresponds to the X13 input point.

When the corresponding macro variable number is greater than or equal to 41000 and less than 41999, it means the corresponding output point Y number plus the value of 41000, for example, 41021 corresponds to the Y21 output point.

When the corresponding macro variable number is greater than or equal to 42000 and less than 43999, it means the corresponding auxiliary relay M number plus the value of 42000, for example, 42257 corresponds to the M257 auxiliary relay.

When you need to define the key functions of F1--F8, add the following statement after defining the size and position of the dialog box:

F1=Y17 means that the state of Y17 is exchanged when the F1 key is pressed.

F3=M78 means that the state of M78 is exchanged when the F3 key is pressed.

F4=P3 means that the ProgramUser3 user-defined program is executed when the F4 key is pressed (here, only ProgramUser0-ProgramUser49 can be used for a total of 50 programs).

N1=Blow switch means F1--F8 button display to configure

For example:

BeginMainFace means start configuration.

0, 20, 80, 420, 300, please input parameters <F1-clear, F8-clear> Four data represent dialog box l/t/w/h, followed by dialog box title prompt information.

F1=Y17 means that the state of Y17 is exchanged when the F1 key is pressed.

F3=M78 means that the state of M78 is exchanged when the F3 key is pressed.

F4=P3 means to execute ProgramUser3 user-defined program when pressing F4 key

F5=M74 means that the state of M74 is exchanged when the F5 key is pressed.

N2=Machine lighting means the button of F2 displays "Machine lighting".

4,3,3,280,190,fillpict1.bmp (the name of the fillpict1.bmp image file)

1,282,10,60,25,X: The first data means control type, the next 4 data means l/t/w/h, the rest means the macro variable number (only for edit box) or prompt information.

3,330,10,85,25,345,I corresponds to the #345 macro variable and is an integer (I specifies an integer, if not specified, it is a floating-point type)

1,282,40,60,25,Z:

3,330,40,85,25,288, corresponding to #288 macro variable, and is a floating point type

1,232,70,110,25,X13:

9,330,70,85,25,40013, corresponding to real-time display of X13 input point status (red light means invalid, green light means valid)

1,232,100,110,25,Y21:

9,330,100,85,25,41021, corresponding to real-time display of Y21 output point status (red light means invalid, green light means valid)

1,232,130,110,25,M257:

9,330,130,85,25,42257, corresponding to real-time display of M257 auxiliary relay status (red light means invalid, green light means valid)

1,200,195,100,25, Lead K(E):

3,300,195,90,25,145,

1,200,223,100,25, times L:

3,300,223,90,25,225,

2,30,253,100,26, OK

2,240,253,100,26,Cancel

EndMainFace indicates the end of the configuration.

Notice:

1>, in the main interface, press the Enter key to pop up a dialog box. When the chart input interface is in the automatic state, if you directly press the "Start" button, the input will be valid.

Do not exit the input dialog interface and start running the current program.

2>, can realize multi-page parameter display and setting, through PageUp and PageDown switching, BeginMainFace or BeginMainFace0 means page 0,

BeginMainFace1 represents page 1, BeginMainFace2 represents page 2, and BeginMainFace9 represents page 9. A total of 0--9 can achieve 10 pages of custom interfaces.

E.g:

```
BeginMainFace
0,20,80,420,300,....
.....
.....
EndMainFace
BeginMainFace1
0,20,80,420,300,....
.....
.....
EndMainFace
BeginMainFace2
0,20,80,420,300,....
.....
.....
EndMainFace
```

3. Robot controller, press F5 in main interface will pop up the dialog box

The dialog is defined as follows:

Begin with BeginManuFace and end with EndManuFace. [this line is optional]

The specific dialog and control definitions are exactly the same as the dialog BeginMainFace popped up by pressing the Enter key in the main interface.

It can also realize multi-page display, switch between PageUp and PageDown, BeginManuFace or BeginManuFace0 means page 0,

BeginManuFace1 represents page 1, BeginManuFace2 represents page 2, and BeginManuFace9 represents page 9, a total of 0--9 can realize 10 pages of custom interface.

4. controller interface got few parts:

F: The area where the coordinates are displayed under the main interface (upper left part) [a total of three interfaces Fa/Fb/Fc].

A: The area where the coordinates are displayed under the interface (the upper middle part).

B: The area where the T/G/M code is displayed in any interface (upper right part).

C: The display area of speed/piece time/number of pieces in any interface (lower left part).

D: The area where the program running progress bar and the current state of the system are displayed in any interface (lower right part).

The parameter "custom system interface [4 table part; 8 all]" is valid when it is set to 8. At this time, these custom interfaces are enabled by default.

At this time, you can open a certain interface with the following command:

//-----

MyFaceFa=1 means to open the Fa interface

MyFaceFb=1 means to open the Fb interface

MyFaceFc=1 means to open the Fc interface

MyFaceA=1 means to open the A interface

MyFaceB=1 means to open the B interface

MyFaceC=1 means to open the C interface

MyFaceD=1 means to open the D interface

//-----

MyFaceFa=0 means to close the Fa interface

MyFaceFb=0 means close the Fb interface

MyFaceFc=0 means close the Fc interface

MyFaceA=0 means to close the A interface

MyFaceB=0 means to close the B interface

MyFaceC=0 close the C interface

MyFaceD=0 close the D interface

//-----

SubFaceFa0=1 means to open the current running program display of the Fa interface

SubFaceFb0=1 means to open the current running program display of the Fb interface

SubFaceFc0=1 means to open the current running program display of the Fc interface

SubFaceFa1=1 means to open the current coordinate display of the Fa interface

SubFaceFb1=1 means to open the current coordinate display of the Fb interface

SubFaceFc1=1 means to open the current coordinate display of the Fc interface

SubFaceA0=1 means to open the current coordinate display of the A interface

SubFaceB0=1 means to open the display of current tool and tool compensation number in interface B

SubFaceB1=1 means to open the current G code display of the B interface

SubFaceB2=1 means to open the current M code display of the B interface

SubFaceB3=1 means to enable the display of current machine tool coordinates in interface B

SubFaceB4=1 means to enable display of other current information on interface B

SubFaceC0=1 means to enable the command rate display of the C interface

SubFaceC1=1 means to enable the actual speed display of the C interface

SubFaceC2=1 means to open the feed override display of C interface

SubFaceC3=1 means to enable the fast magnification display of the C interface

SubFaceC4=1 means to open the command speed display of C interface

SubFaceC5=1 means to enable the spindle override display on the C interface

SubFaceC6=1 means to open the actual speed display of the C interface

SubFaceC7=1 means to open the spindle SP display of the C interface

SubFaceC8=1 means to open the single time display of the C interface

SubFaceC9=1 means to enable the display of the number of processed pieces in the C interface

SubFaceD0=1 means that the program running progress bar of the D interface is opened

SubFaceD1=1 means to open the F warning bar display of the D interface

SubFaceD2=1 means to enable the display of the S warning bar on the D interface

SubFaceD3=1 means to enable the system status display of the D interface

//-----

SubFaceFa0=0 means to close the current running program display on the Fa interface

SubFaceFb0=0 means to close the current running program display on the Fb interface

SubFaceFc0=0 means to close the current running program display of the Fc interface

SubFaceFa1=0 means to close the current coordinate display of the Fa interface

SubFaceFb1=0 means to close the current coordinate display of the Fb interface

SubFaceFc1=0 means to close the current coordinate display of the Fc interface

SubFaceA0=0 means to close the current coordinate display of the A interface

SubFaceB0=0 means to close the display of the current tool and tool

compensation number in the B interface

SubFaceB1=0 means to close the current G code display of the B interface

SubFaceB2=0 means to close the current M code display of the B interface

SubFaceB3=0 means to close the display of current machine tool coordinates in interface B

SubFaceB4=0 means to close the display of other current information on the B interface (such as dry run/auxiliary lock/machine lock, etc.)

SubFaceC0=0 means to close the command rate display of the C interface

SubFaceC1=0 means to close the actual speed display of the C interface

SubFaceC2=0 means to close the display of feed override in C interface

SubFaceC3=0 means to close the fast magnification display of the C interface

SubFaceC4=0 means to close the command speed display of C interface

SubFaceC5=0 means to close the display of spindle override in C interface

SubFaceC6=0 means to close the actual speed display of the C interface

SubFaceC7=0 means to close the spindle SP display of the C interface

SubFaceC8=0 means to close the single time display of C interface

SubFaceC9=0 means to close the display of the number of workpieces in the C interface

SubFaceD0=0 means to close the program running progress bar display of the D interface

SubFaceD1=0 means to turn off the display of the F warning bar on the D interface

SubFaceD2=0 means to turn off the display of the S warning bar on the D interface

SubFaceD3=0 means to close the system status display of D interface

//-----

The three interfaces Fa/Fb/Fc correspond to the display definitions of the F soft key buttons:

Function1=relative coordinates

Function2=Workpiece coordinates

Function3=Comprehensive coordinates

You can modify the characters displayed by the button by yourself

The definition of system interface F [a total of three interfaces Fa/Fb/Fc]:

The Fa interface:

BeginFaceFa

0,10,30,450,400; the first character indicates the style of the area border; the following numbers indicate the range of drawing the area border

.....

EndFaceFa

Fb interface:

BeginFaceFb

0,10,30,450,400; the first character indicates the style of the area border; the following numbers indicate the range of drawing the area border

.....

EndFaceFb

Fc interface:

BeginFaceFc

0,10,30,450,400; the first character indicates the style of the area border; the following numbers indicate the range of drawing the area border

.....

EndFaceFc

Definition of system interface A:

BeginFaceA

0,470,30,200,400; The first character indicates the area border style; the following numbers indicate the range of drawing the area border

.....

EndFaceA

Definition of system interface B:

BeginFaceB

0,680,30,110,400; The first character indicates the area border style; the following numbers indicate the range of drawing the area border

.....

EndFaceB

Definition of system interface C:

BeginFaceC

0,10,435,450,100; the first character indicates the style of the area border; the following numbers indicate the range of drawing the area border

.....

EndFaceC

Definition of system interface D:

BeginFaceD

0,470,435,320,100; The first character indicates the border style of the area; the following numbers indicate the range of drawing the border of the area

.....

EndFaceD

Notice:

1>, The regional border style includes: 'D' for concave border; 'U' for convex border; 'F' for flat border; '0' for no border.

2>, point out that if the range of drawing the border of this area exceeds the actual range, the software will automatically limit it.

The maximum range of system interface F: left=10; top=30; right=460; bottom=430.

The maximum range of system interface A: left=470; top=30; right=670; bottom=430.

The maximum range of system interface B: left=680; top=30; right=790; bottom=430.

The maximum range of system interface C: left=10; top=435; right=460; bottom=535.

The maximum range of system interface D: left=470; top=435; right=790; bottom=535.

The scope of the current running program of the system interface Fa interface: left=; top=; right=; bottom=. //This is more troublesome, the number of axes is different, and the position of the displayed area is also different

The scope of the current running program of the system interface Fb interface: left=; top=; right=; bottom=. //This is more troublesome, the number of axes is different, and the position of the displayed area is also different

The scope of the current running program of the system interface Fc interface: left=; top=; right=; bottom=. //This is more troublesome, the number of axes is different, and the position of the displayed area is also different

The range of the current coordinates of the Fa interface of the system interface: left=; top=; right=; bottom=. //This is more troublesome, the number of axes is different, and the position of the displayed area is also different

The range of the current coordinates of the system interface Fb interface: left=; top=; right=; bottom=. //This is more troublesome, the number of axes is different, and the position of the displayed area is also different

The range of the current coordinates of the system interface Fc interface: left=; top=; right=; bottom=. //This is more troublesome, the number of axes is different, and the position of the displayed area is also different

The range of the current coordinates of the system interface A interface: left=470; top=30; right=670; bottom=430.

The range of the current tool and tool compensation number in the system interface B interface: left=675; top=63; right=792; bottom=113.

The range of the current G code of the system interface B interface: left=675; top=115; right=792; bottom=240.

The range of the current M code of the system interface B interface: left=675; top=270; right=792; bottom=369.

The range of the current machine tool coordinates in the system interface B interface: left=; top=; right=; bottom=. // Actually there is no such display area

The current range of other information on the system interface B interface: left=675; top=372; right=792; bottom=432.

The range of the command rate of the system interface C interface: left=5; top=435; right=220; bottom=460.

The range of the actual speed of the system interface C interface: left=5; top=460; right=220; bottom=485.

The range of the feed override of the system interface C interface: left=5; top=485; right=220; bottom=510.

The range of the quick magnification of the system interface C interface: left=5; top=510; right=220; bottom=535.

The range of the command speed of the system interface C interface: left=230; top=435; right=398; bottom=460.

The range of the spindle override of the system interface C interface: left=400; top=435; right=470; bottom=455.

The actual speed range of the system interface C interface: left=230; top=460; right=426; bottom=485.

The range of the main axis SP of the system interface C interface: left=428; top=460; right=470; bottom=480.

The range of the single piece time of the system interface C interface: left=230; top=485; right=450; bottom=510.

The range of the number of workpieces in the system interface C interface: left=230; top=510; right=450; bottom=535.

The range of the program running progress bar in the system interface D interface: left=479; top=470; right=635; bottom=486.

The range of the F warning bar in the system interface D interface: left=495; top=490; right=625; bottom=510.

The range of the S warning bar on the D interface of the system interface: left=495; top=512; right=625; bottom=532.

The range of the system state of the system interface D interface: left=630; top=488; right=796; bottom=539.

3>, control type: 0/D/U/F table dialog box itself; 1 means custom static text; 2 means custom button; 3 means custom edit box; 4 means picture; 8 means custom small font static text ;

9 means a custom edit box that can be automatically refreshed dynamically.

The image file names are: fillpict1, fillpict2, fillpict3 and must be 16-color (ie 4-bit) or 256-color (ie 8-bit) or 16-bit or 24-bit bmp format images.

But note: 16-color (ie 4-bit) or 256-color (ie 8-bit) pictures should be used as much as possible to reduce the space occupied by the pictures.

For the custom edit box: when the corresponding macro variable number is greater than or equal to 20000, it means that the current coordinates or other status information is displayed. Currently, it is defined as follows:

20001/20002/20003/20004/20005 respectively means XYZ45 axis machine coordinates.

20011/20012/20013/20014/20015 respectively means XYZ45 axis workpiece coordinates or absolute coordinates.

20201 indicates the current spindle status, and the information of "CW", "CCW" and "Stop" will be displayed respectively.

4.The files interfacecn.cfg (Chinese) and interfaceeg.cfg (English) must be edited with Windows Notepad or on the controller, then restore into controller, and cannot be edited with other editors.

3. User-defined parameter Configuration

1) System configuration file **cncsystemcn.cfg**(Chinese) and **cncsystemen.cfg**(English) include two kinds of contents

- a) System main interface configuration;
- b) Parameter No. and prompt message configuration.

Parameter file configuration PLC input point and auxiliary relay method:

(1), 'E' means to modify the normally open or normally closed state of the corresponding input point X in the PLC. (The parameter value is 0 means normally open, 1 means normally closed, no value means there is no such contact in the PLC). For example: Param#E12G1= //Indicates to set the normally open or normally closed state of X12.

(2), 'F' means to modify the normally open or normally closed state of the corresponding auxiliary relay M in the PLC. (The parameter value is 0 means normally open, 1 means normally closed, no value means there is no such contact in the PLC).

For example: Param#F250G1= //means to set the normally open or normally closed state of M250.

2) Parameter configuration:

ParameterConfig //means following is parameter configuration;

DefaultFunc=3 //means default entrance is F3 parameter after booting;

Function1=Process parameter //means F1 button parameter(F1 is
Process parameter)

Param#I67G1= //means the parameter type, code and content in the F1 key
parameter configuration, the first number after G is the parameter password
protection level

1,X direction (0 positive, non-0 negative) //contents of prompt dialog,this
line is not required if there is no additional information.;

Param#P61G0= //means parameter type,code and content in F1 parameter
menu, no. behind G means parameter password protection level;

2,G70G71G72 default allowance(default as 20) //contents of prompt dialog,
this line is not required if there is no additional information.;;

Parameter contents will be added “;” and “space” automatically.

Password protection level:

0 means user level, 1 means machine factory level, 2 means cnc level;
add 1 behind # means show the parameter but limited by axis quantity, add 2
behind # means show the parameter unlimited.

If there is second number behind G, it means axis quantity decides parameter
status: 3 means display when 3rd axis enabled, 4 means display when 4th axis
enabled, 5means display when 5th axis enabled.

//-----

As far now, the defined parameter types are as following:

- 1) 'X','Y','Z','A' corresponds to integer type reference point parameter
referX[],referY[],referZ[],referA[];
- 2) 'x','y','z','a','b' corresponds to integer type work piece coordinate
parameter CoX[],CoY[],CoZ[],CoA[],CoB[];
- 3) 'S' corresponds to floating point type speed parameter speed[];
- 4) 'I' corresponds to integer type axis parameter Axis[];
'i' corresponds to float point type axis parameter Axis1[];
- 5) 'O' corresponds to integer type other parameter Other[];
'o' corresponds to float point type other parameter Other1[];
- 6) 'P' corresponds to integer type additional parameter AddPara[];
- 7) 'M' corresponds to float point type macro variable parameter Macro[];
'm' corresponds to integer type macro variable parameter Macro[];
'c' corresponds to character type macro variable parameter Macro[];
- 8) 'N' corresponds to parameter note(comment).

Robot controller: V= is used to define feed speed. Unit is mm/s.

The solution for changing speed unit into mm/s in parameter configuration:
add character T and when character C following T: means data type keep
the same, the displayed value will be divided by 60 before showing. When
do setting, input value will be multiply by 60, then save into system. This is
normally used to switch mm/min into mm/s.

For example:

Param#S1G1TC=

1,J1 axis MOVJ speed(degree/s)

Param#S5G0TC=
8,MOVL/MOVC default speed(mm/s)
Param#S13G0TC=
10,Feed axis manual speed(mm/s)

//-----
//-----

New function: Txxx: capital T can be used to specify parameter data type.

a) The first number behind T will define the data type:

- 0: not define;
- 1: integer type;
- 2: real type;
- 3: bit parametric type;
- 4-9: boolean type

(4: unequal, 5: equal, 6: greater than, 7: less than, 8: greater than or equal to, 9: less than or equal to);

b) The second number behind T defines bit length when it is bit parameter type: 8 means 8 bits, otherwise 16 bits(default as 16 bits);

c) The second number behind T defines cut-off point when it is boolean type: condition met then true(1), condition not met then false(0).

//-----

The parameters need defining data type include:

- A) Integer type axis parameter Axis[]-'I'. maybe include boolean type and bit parameter type;
- B) Integer type other parameter Other[]-'O'. maybe include boolean type and bit parameter type;
- C) Integer type additional parameter AddPara[]-'P'. maybe include boolean type and bit parameter type;
- D) Float-point type showing macro variable parameter Macro[]-'M'. maybe include boolean type.

//-----

3) Parameter configuration file includes each parameter's type, No., contents and menu contents also.

When system booting, it will read parameter configuration file, and if the protection level does not open, it will not read.

When protection level is changed, system will read configuration file again.

Parameter configuration file includes 2 files: cncsystemcn.cfg(Chinese) and cncsystemeg.cfg(English). when configuration content is changed, system need to read configuration file again.

4) When the first character of any lines is “/”, it means this line is note.

5) M code display configuration(only available for bus type platform);

Auxiliary relay: M48-M79(only).

Letter and number display should be 3 character(only), and start with capital M.

If there is no editing, the original default display will remain unchanged.

The file starts from **BeginDwSelfMcode**, ends with **EndDwSelfMcode**.

Sample:

BeginDwSelfMcode //Start.

- 1, //1 means 1st display position, sequence is from left to right, from upper to lower.
 - a,M53,g,Spindle CW, //a means the 1st case, M53 means condition is relay M53 valid, g means green light, following is display message;
 - b,M54,g,Spindle CCW, //b means the 2nd case, M54 means condition is relay M54 valid, g means green light, following is display message;
 - c,M67,b,Spindle Stop, //c means the 3rd case, M67 means condition is relay M67 valid, b means blue light, following is display message;
 - d,NNN,r,Spindle Stop, //d means the 4th case, NNN means no above condition met, r means red light, following is display message;
- 2, //2 means 2nd display position, sequence is from left to right, from upper to lower.
 - a,M49,r,Tool loosen, //a means the 1st case, M49 means condition is relay M49 valid, g means green light, following is display message;
 - b,m49,g,Tool tighten, //b means the 2nd case, m49 means condition is relay M49 invalid, r means red light, following is display message;
- 3, //3 means 3rd display position, sequence is from left to right, from upper to lower.
 - a,M48,g,M08, //a means the 1st case, M48 means condition is relay M48 valid, g means green light, following is characters and numbers;
 - b,m48,r,M09, //b means the 2nd case, m48 means condition is relay M48 invalid, r means red light, following is characters and numbers;
- 4, //4 means 4th display position, sequence is from left to right, from upper to lower;
 - a,M51,g,Lubrication on, //a means the 1st case, M51 means condition is relay M51 valid, g means green light, following characters and numbers;
 - b,m51,r,M33, //b means the 2nd case, m51 means condition is relay M51 invalid, r means red light, following is characters and numbers;
- 5, //5 means 5th display position, sequence is from left to right, from upper to lower;
 - a,M56,g,Test on, //a means the 1st case, M56 means condition is relay M56 valid, g means green light, following is characters and numbers;

b,m56,r,Test off, //b means the 2nd case, m56 means condition is relay M56 invalid, r means red light, following is characters and numbers;
6, //6 means 6th display position, sequence is from left to right, from upper to lower;
a,M57,r,Practice on, //a means the 1st case, M57 means condition is relay M57 valid, g means green light, following is characters and numbers。
b,m57,g,Practice off, //b means the 2nd case, m57 means condition is relay M57 invalid, r means red light, following is characters and numbers;
EndDwSelfMcode //End

6) Graph configuration in configuration file **cncsystemeg.cfg**

```
GraphProgramEdit  
Gcode#19999=  
0,20,80,420,300,Input <F1-clear, F8-null>  
1,2,10,380,25,Input No. corresponding command  
1,2,90,380,25,3-SN; 4-SF  
8,2,110,380,18,5-AS; 6-AE; 7-WS; 8-WE  
1,20,215,60,25,Command  
3,70,215,180,25,  
2,30,250,130,30,Enter  
2,230,250,130,30,Esc
```

The 1st bit defines control type, following 4 bit data define l/t/w/h (start point, width, height), next is address word, the last is prompt message;

Control types:

- 0 refers to dialog box itself;
- 1 means user-defined static text;
- 2 means user-defined button;
- 3 means user-defined edit box;
- 4 means picture;
- 8 means user-defined small fonts static text.

Modify l/t/w/h to adjust the box, and also use abbreviation to suit box.

Additional Sample for graphic program:

```
GraphProgramEdit //Start  
Gcode#19999= //Gcode#19999= means general dialog box in the  
beginning;  
0,20,80,420,300,Please input G code number<F1-Clear, F8-Null>  
//the first bit refers to control type, four data from  
2nd bite to 5 bite, the dialog box l/t/w/h, l/t means  
start point coordinate of box, w/h means width and  
height, following is prompt information;
```

1,2,10,380,25,please input //the first bit refers to control type, four data from 2nd bite to 5 bite, the dialog box l/t/w/h, l/t means start point coordinate of box, w/h means width and height, following is address word(only edit box), the last bit is prompt information;

1,2,30,380,25,Please input//

1,10,200,60,25,Command G//

3,70,200,180,25,

2,30,250,130,30,Enter

2,230,250,130,30,Esc

Gcode#78= //Gcode#78=means G78 dialog box;

0,20,80,420,300,Please input G78 parameter <F1-Clear, F8-Null>

//the first bit refers to control type, four data from 2nd bite to 5 bite, the dialog box l/t/w/h, l/t means start point coordinate of box, w/h means width and height, the last bit is prompt information;

9,G78

1,275,10,60,25,X(U): //the first bit refers to control type, four data from 2nd bite to 5 bite, the dialog box l/t/w/h, l/t means start point coordinate of box, w/h means width and height, following is address word(only edit box), the last bit is prompt information;

3,330,10,85,25,X,

1,275,40,60,25,Z(W):

3,330,40,85,25,Z,

1,275,70,60,25,I:

3,330,70,85,25,I,

1,275,100,60,25,J:

3,330,100,85,25,J,

1,275,130,60,25,Q:

3,330,130,85,25,Q,

1,200,195,100,25,pitch K(E):

3,300,195,90,25,K,

1,200,223,100,25,leader quantity L(SP):

3,300,223,90,25,L,

9#Fixed circle command

2,30,250,130,30,Enter

2,230,250,130,30,Esc

4,2,2,270,190,gcode78.bmp //gcode78.bmp is picture file name;

EndGraphProgEdit //End

Control types:

- 0: Dialog box itself;
- 1: User-defined static text;
- 2: User-defined button;
- 3: User-defined edit box;
- 4: Picture;
- 5: MiniGUI static text;
- 6: MiniGUI edit box
- 7: User-defined small-font static text;
- 9: Generate command type fixed word string(if this control exists, it will not generate G code again, and this control will display in 1st line);
- 9#: generate fixed word string(used to generate note, and not add “,” behind #, displaying in the 1st line).

Attention: the sequence of word string generating from 9 and 9# accords to sequence of configuration file, and should be put after configuration, because 9# will generate comma(“,”) automatically.

As to MST code, they can be put into G code dialog box.

G code can correspond to pictures, name principle is: gcode+number.bmp. For example:gcode176.bmp, gcode8.bmp, gcode22.bmp. And must be 16 color (4 bits) or 256 color (8 bit) or 16 or 24 bit BMP format picture.

However, attention should be paid to the use of 16 color (4 bits) or 256 color (8 bit) picture, in order to minimize the image space.

In edit box, the address word adding number will define its displaying position, for example, 1 means in the 1st line, 3 means in the 3rd line, 23 means display in both 2nd and 3rd lines. The number except for 0-9 will be defined as the 1st line.

For picture type control: prompt message is picture file name.

The line beginning with “/” means note line.

If the string in edit box beginning with letter, then program will not generate address word string automatically.

For example, editor is 3,230,10,100,25,X, if input 23.456, it will generate X23.456, but if input U23.456, it will generate U23,456.

Input GCODEX in Other parameter P12 “Inner parameter”, and press Enter, system will delete all gcode+number.bmp picture file.

Picture file must be 16 color (4 bits) or 256 color (8 bit) or 16 or 24 bit BMP format. But, attention should be paid to the use of 16 color (4 bits) or 256 color (8 bit) picture, in order to minimize the image space.

4. Robot controller palletizing parameter configuration

There are 3 files used to save palletizing parameter

(1) **palletbase.txt** saves the basic parameter of palletizing(The range is 1--99)

The parameters including:

The number of palletizing;

The type is palletizing or depalletizing;

The operation sequence of line and column of palletizing(This parameter is only effective when the layout type set as 0);

The number of line, column and layer.

The distance between the line or column of palletizing; Pay attention: If not specify or 0, it will be useless, it will calculate according to the end point of teach and number of line and column.

Approach point(2 points at most);

The number of back points(2 points at most);

The approach speed;

The back speed;

Reach with low speed and the offset distance of leave the overlap points;

Reach with low speed and the offset speed of leave the overlap points;

The layout type:

0 means generate layout type of each layer automatically(The posture is the teach posture of basic point).

1 means every layer is the first layout mode of teach.

2 means the layout type according to the parity layer(Odd layer correspond to the first kind of layout mode, even layer correspond to the second layout mode).

3 means user-defined setting, every layer need specify the corresponding the layout mode.

The coordinate of 4 teach points;

The coordinate of approach point and back point;

The layout number of every layer, 0 means the default;

The special set for height of every layer, 0 means the default;

The logic control for loose and tight is user-defined instruction M880--M889,customer can set it through parameter.

Format of file include six parts:

BaseParam: //Basic parameter

RCLcharac: //Characteristic points of pile

ApprRtret: //Point of approach and back

NumLayout: //Special set for layout number of each layer

LayHeight: //Special set for height of every layer

ApRtSpecl: //Teach of back point and special approach in special layer

BaseParam: //Includes these parts:

The first data means the current number of palletizing (1--99)

B-Type: //Type

B-Sers: //Sequence

B-Rnum: //Number of line

B-Cnum: //Number of column

B-Lnum: //Number of layer

B-Rdst: //Distance of line
B-Cdst: //Distance of column
B-Ldst: //Distance of layer
B-ApPt: //Number of approach points
B-RtPt: //Number of back points
B-ApFd: //Approach speed
B-RtFd: //Back speed
B-LowD: //Distance of slow move
B-LowF: //Speed of slow move
B-Lout: //Layout format

RCLcharac: //Includes these parts:

The first data means the current number of palletizing (1--99)

R-OrgX: //X of original
R-OrgY: //Y of original
R-OrgZ: //Z of original
R-OrgC: //C of original
R-RowX: //X of line end point
R-RowY: //Y of line end point
R-RowZ: //Z of line end point
R-RowC: //C of line end point
R-ColX: //X of column end point
R-ColY: //Y of column end point
R-ColZ: //Z of column end point
R-ColC: //C of column end point
R-LayX: //X of layer end point
R-LayY: //Y of layer end point
R-LayZ: //Z of layer end point
R-LayC: //C of layer end point

Pay attention: Also need to set original point and layer end point, needn't set the line end point and column end point.

AprRtret: Includes these parts:

The first data means the current number of palletizing (1--99)

A-Ap2X: //X of approach point 2
A-Ap2Y: //Y of approach point 2
A-Ap2Z: //Z of approach point 2
A-Ap2C: //C of approach point 2
A-Ap1X: //X of approach point 1
A-Ap1Y: //Y of approach point 1
A-Ap1Z: //Z of approach point 1

A-Ap1C: //C of approach point 1

A-Rt1X: //X of back point 1

A-Rt1Y: //Y of back point 1

A-Rt1Z: //Z of back point 1

A-Rt1C: //C of back point 1

A-Rt2X: //X of back point 2

A-Rt2Y: //Y of back point 2

A-Rt2Z: //Z of back point 2

A-Rt2C: //C of back point 2

NumLayout: Includes these parts:

The first data means the current number of depalletizing (1--99)

One by one to arrange every layer which correspond to layout number.
Each layout number data on a line by itself.

LayHeight: Includes these parts:

The first data means the current number of depalletizing (1--99)

One by one to arrange every layer height which correspond to layout number. Each height data on a line by itself.

This height means from the bottom to this layer, unit:mm

ApRtSpecl: Includes these parts:

The first data is consist of 6 bis number:

The highest 2bits mean the current number of palletizing(1--99)

The middle 2bits mean the current layer number of palletizing(1--99)

The lowest 2bits mean the current layer stack number of palletizing(1--

99)

A-Ap2X: //X of approach point 2;

A-Ap2Y: //Y of approach point 2;

A-Ap2Z: //Z of approach point 2;

A-Ap2C: //C of approach point 2;

A-Ap1X: //X of approach point 1;

A-Ap1Y: //Y of approach point 1;

A-Ap1Z: //Z of approach point 1;

A-Ap1C: //C of approach point 1;

A-Rt1X: //X of back point 1;

A-Rt1Y: //Y of back point 1;

A-Rt1Z: //Z of back point 1;

A-Rt1C: //C of back point 1;

A-Rt2X: //X of back point 2;

A-Rt2Y: //Y of back point 2;

A-Rt2Z: //Z of back point 2;

A-Rt2C: //C of back point 2;

(2) **pallayout.txt** saves the data of layout(The range is 1--99)

Includes these parameters:

The number of line, column and 3 characteristic points in regular layout;

Coordinate of each point in irregular layout.

Pay attention: In the same number of layout, it only to be regular or irregular, can't be together in the same number.

The teach for both regular and irregular layout is from the bottom.

Format:

PalLayout: //Regular or irregular

The first data means the current number of depalletizing (Regular is marked with Y, irregular is marked with N) ;

Layout for regular:

P-OrigX: //X of the original point ;

P-OrigY: //Y of the original point ;

P-OrigZ: //Z of the original point ;

P-OrigC: //C of the original point ;

P-RowX: //X of the line end point ;

P-RowY: //Y of the line end point ;

P-RowZ: //Z of the line end point ;

P-RowC: //C of the line end point ;

P-ColX: //X of the column end point ;

P-ColY: //Y of the column end point ;

P-ColZ: //Z of the column end point ;

P-ColC: //C of the column end point ;

P-Rnum: //Number of line ;

P-Cnum: //Number of column ;

In this case, the fixed operation sequence of line and column are RCL.

Layout for irregular:

P-***N: //Specify the number of every stack, for example:P-002N: P-012N: P-006N:

Arrange every XYZC coordinate which correspond to every stack one by one.

If there is N in coordinate, it means there is no set for this stack.

(3) **palletcurr.txt** saves the current status of palletizing(The current stage will be saved when power off)

Format:

Palltcode:

The data behind is:

//The number of palletizing;
 //The total number of stack currently executed[Start from 0 to count];
 //The number of the current layer[Start from 0 to count] (If palletizing then count from bottom to top, if depalletizing then count from top to bottom);
 //The current line number [Start from 0 to count] (The meaning of palletizing is the same as depalletizing);
 //The current column number [Start from 0 to count] (The meaning of palletizing is the same as depalletizing);
 //The number of the stack in the current layer [Start from 0 to count](for depalletizing, it means the depalletized stack number).

(4) input **PALLET** in Other parameter P15 “Inner parameter”, system will delete palletbase.txt, paltlayout.txt and palletcurr.txt.

5. Intersection line command(for both cnc and robot)

MOVCI or **MCI** means intersecting line instruction(need to teach P1P2P3 points);

MOVC L93/L94 means intersecting line instruction(need to teach P1P2P3 points);

MC L93/L94 means intersecting line instruction(need to teach P1P2P3 points);

MOVCI L93/L94 means intersecting line instruction(need to teach P1P2P3 points);

MOCI L93/L94 means intersecting line instruction(need to teach P1P2P3 points);

G17 L93 G2/G3 I_ J_ X_ Y_ K_ (Q_) means intersecting line instruction;

I_ J_ means XY incremental coordinate value of small cylinder center based on P1 (unit:mm), or **I_ J_** can be omitted, and add **R_** to define radius of small cylinder;

X_ Y_ define coordinate of endpoint P3(G90 absolute coordinate, G91 incremental coordinate). If add **Q**, and when **Q** is not 0, it is radius of big cylinder(mm), **K1** or **K-1** will define Z offset direction of intersection P2 point based on P1. If not add **Q**, or **Q** is 0, then **K** defines Z axis incremental coordinate of middle point P2 based on P1(unit:mm); controller will calculate big cylinder’s radius automatically according to **K** value, and judge Z axis offset direction of P2 based on P1 according to sign of **K**;

G18 L93 G2/G3 K_ I_ Z_ X_ J_ (Q_)

K_ I_ means ZX incremental coordinate value of small cylinder center based on P1 (unit:mm), or **K_ I_** can be omitted, and add **R_** to define radius of small cylinder;

Z_ X_ define coordinate of endpoint P3(G90 absolute coordinate, G91 incremental coordinate). If add **Q**, and when **Q** is not 0, it is radius of big cylinder(mm), **J1** or **J-1** will define Y offset direction of intersection P2 point based on P1. If not add **J**, or **J** is 0, then **J** defines Y axis incremental coordinate of middle point P2 based on P1(unit:mm); controller will calculate big cylinder’s radius automatically according to **J** value, and judge Y axis offset direction of P2 based on P1 according to sign of **J**;

G19 L93 G2/G3 J_ K_ Y_ Z_ I_ (Q_)

J_ K_ means YZ incremental coordinate value of small cylinder center based on P1 (unit:mm), or **J_ K_** can be omitted, and add **R_** to define radius of small cylinder;

Y_ Z_ define coordinate of endpoint P3(G90 absolute coordinate, G91 incremental coordinate). If add **Q**, and when **Q** is not 0, it is radius of big cylinder(mm), **I1** or **I-1** will define X offset direction of intersection P2 point based on P1. If not add **J**, or **J** is 0, then **I** defines X axis incremental coordinate of middle point P2 based on P1(unit:mm); controller will calculate big cylinder’s radius

automatically according to I value, and judge X axis offset direction of P2 based on P1 according to sign of I;

G17 L93 G5 I_ J_ X_ Y_ K_ (Q_) means intersecting line instruction;

I_ J_ means XY incremental coordinate value of middle point P2 based on P1 (unit:mm);

X_ Y_ define coordinate of endpoint P3(G90 absolute coordinate, G91 incremental coordinate).

If add Q, and when Q is not 0, it is radius of big cylinder(mm), K1 or K-1 will define Z offset direction of intersection P2 point based on P1. If not add Q, or Q is 0, then K defines Z axis incremental coordinate of middle point P2 based on P1(unit:mm); controller will calculate big cylinder's radius automatically according to K value, and judge Z axis offset direction of P2 based on P1 according to sign of K;

G18 L93 G5 K_ I_ Z_ X_ J_ (Q_)

K_ I_ means ZX incremental coordinate value of middle point P2 based on P1 (unit:mm);

Z_ X_ define coordinate of endpoint P3(G90 absolute coordinate, G91 incremental coordinate).

If add Q, and when Q is not 0, it is radius of big cylinder(mm), J1 or J-1 will define Y offset direction of intersection P2 point based on P1. If not add J, or J is 0, then J defines Y axis incremental coordinate of middle point P2 based on P1(unit:mm); controller will calculate big cylinder's radius automatically according to J value, and judge Y axis offset direction of P2 based on P1 according to sign of J;

G19 L93 G2/G3 J_ K_ Y_ Z_ I_ (Q_)

J_ K_ means YZ incremental coordinate value of middle point P2 based on P1 (unit:mm);

Y_ Z_ define coordinate of endpoint P3(G90 absolute coordinate, G91 incremental coordinate).

If add Q, and when Q is not 0, it is radius of big cylinder(mm), I1 or I-1 will define X offset direction of intersection P2 point based on P1. If not add J, or J is 0, then I defines X axis incremental coordinate of middle point P2 based on P1(unit:mm); controller will calculate big cylinder's radius automatically according to I value, and judge X axis offset direction of P2 based on P1 according to sign of I;

Attention:

For MOVCI/MCI/G5/G6, only half of intersection can be programmed once, full intersection line needs two lines of program. With G2/G3 L93, full intersection can be programmed once. Till now, intersection line instruction only support one cylinder being perpendicular to another cylinder.

Definition rules of User(workpiece) coordinate system:

For MOVCI/MCI, axis of small cylinder is Z axis direction, axis of big cylinder is X axis direction.

For G6 L93, axis of small cylinder is Z axis direction, axis of big cylinder is X axis direction.

For G17 L93 G2/G3/G5, axis of small cylinder is Z axis direction, axis of big cylinder is X axis direction.

For G18 L93 G2/G3/G5, axis of small cylinder is Y axis direction, axis of big cylinder is Z axis direction.

For G19 L93 G2/G3/G5, axis of small cylinder is X axis direction, axis of big cylinder is Y axis direction.

For the intersection line specified by G2/G3/G5/G6, when change L93 into L94, definition rule will change;

For MOVCI L94/MCI L94, axis of small cylinder is Z axis direction, axis of big cylinder is Y axis direction.

For G6 L94, axis of small cylinder is Z axis direction, axis of big cylinder is Y axis direction.

For G17 L94 G2/G3/G5, axis of small cylinder is Z axis direction, axis of big cylinder is Y axis direction.

For G18 L94 G2/G3/G5, axis of small cylinder is Y axis direction, axis of big cylinder is X axis direction.

For G19 L94 G2/G3/G5, axis of small cylinder is X axis direction, axis of big cylinder is Z axis direction.

6. Program file encryption

Other parameter P904(password required) is used to enable encryption function;

Set P904=4: means enable program file decryption;

8: means enable program file encryption;

16: means enable special program file decryption;

32 means enable special program file encryption.

Special program includes:

Home program ProgramHome; M6 instruction file: ProgramM6;

User-defined M880-M929 instruction file;

User-defined G100-G170 instruction file.

When P904=4 or 8, in program content interface, move cursor to the file, and press “-” button, it will generate encryption or decryption file, whose name is original name+.ENCRY, and expanded name will be deleted when decrypt file.

When P904=16 or 32, pressing F1 in diagnosis interface, the dialog will pop up, the message is as follows:

Encryption program(H-home; T-T command; M-M6 command, 0-49: M880-M929; G100-G170);

Decryption program(H-home; T-T command; M-M6 command, 0-49: M880-M929; G100-G170);

Attention:

a. The expand name of main encryption file must be “.ENCRY” or “.ency”, otherwise, system can not execute the file normally.

b. User-defined tool change program and user-defined M code program file can be encryption file, but keep original file name; user-defined G code G100-G170 program file can be encryption file, but keep original file name.

7. G/M code status display configuration

BeginDwSelfMcode //Start.

1, //1 means 1st display position, sequence is from left to right, from upper to lower.

- a,M53,g,Spindle CW, //a means the 1st case, M53 means condition is relay M53 valid, g means green light, following is display message;
- b,M54,g,Spindle CCW, //b means the 2nd case, M54 means condition is relay M54 valid, g means green light, following is display message;
- c,M67,b,Spindle Stop, //c means the 3rd case, M67 means condition is relay M67 valid, b means blue light, following is display message;
- d,NNN,r,Spindle Stop, //d means the 4th case, NNN means no above condition met, r means red light, following is display message;
- 2, //2 means 2nd display position, sequence is from left to right, from upper to lower.
 - a,M49,r,Tool loosen, //a means the 1st case, M49 means condition is relay M49 valid, r means green light, following is display message;
 - b,m49,g,Tool tighten, //b means the 2nd case, m49 means condition is relay M49 invalid, g means red light, following is display message;
- 4, //4 means 4th display position, sequence is from left to right, from upper to lower.
 - a,M48,g,M08, //a means the 1st case, M48 means condition is relay M48 valid, g means green light, following is display message;
 - b,m48,r,M09, //b means the 2nd case, m48 means condition is relay M48 invalid, r means red light, following is display message;
- 5, //5 means 5th display position, sequence is from left to right, from upper to lower;
 - a,M51,g,Lubrication on, //a means the 1st case, M51 means condition is relay M51 valid, g means green light, following is display message;
 - b,m51,r,M33, //b means the 2nd case, m51 means condition is relay M51 invalid, r means red light, following is display message;
- 3, //3 means 3rd display position, sequence is from left to right, from upper to lower;
 - a,M56,g,Test on, //a means the 1st case, M56 means condition is relay M56 valid, g means green light, following is display message;
 - b,m56,r,Test off, //b means the 2nd case, m51 means condition is relay M51 invalid, r means red light, following is display message;

6, //6 means 6th display position, sequence is from left to right, from upper to lower;
a,M57,r,Practice on, //a means the 1st case, M57 means condition is relay M57 valid, r means green light, following is display message.
b,m57,g,Practice off, //b means the 2nd case, m57 means condition is relay M57 invalid, g means red light, following is display message;
EndDwSelfMcode //End

Address: Economic Headquarter E2, No.888 Chenglong Avenue, Longquanyi District, Chengdu City, Sichuan Province, People's Republic of China.

Website: WWW.NEWKER-CNC.COM

Email: NEWKer.CNC@aliyun.com

Phone: 028-84601148

Fax: 028-84609226

Skype: Newker-CNC James

Whatsapp: +8618990029555