Macro Program Instruction

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: **CASET** +(-)X**+(-)X**+(-)X**+(-)X**+(-)X**+(-)X**+(-)X**

T** means tool number;
"+" means valid input;
"-" means invalid input;
"X" means input signal.

Example: CASET1-X16+X17-X18-X19+X20

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

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.

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 work piece 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 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: 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**

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.

2. Custom macro program instructions G101-G170，G65、G66、G67

1、G101-G170 Corresponds to macro program ProgramG101- ProgramG170; G101-G150 are modeless 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) Modeless 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—M889**）

Each custom G code corresponds to a macro program ProgramGxxx, M Code corresponding to a macro program ProgramUser1-ProgramUser9, 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
```

3. 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.

Example：

IniStatus=M means Boot into manual continuous status.
IniStatus=A means Boot into the automatic continuous status.

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** [1 is invalid, 4 is valid, 12 is all, +256: G54-G59 Display by USxx]"

//+2 means inner occupied;

//+4 means dialog will pop up when press Enter in main interface;

//+8 means available certain modification in both main interface and sub interface;

//+16 means dialog will pop up and display user-defined picture when press Enter in main interface;

//+32 means hide Z axis setting when set work piece coordinate.

//+64 means record all program running time automatically;

//+128 means block manual switch of 4th and 5th axis in 5 axis controller;

//+256 means display user coordinate with US0-US54 in robot controller;

//+512 means the 2nd channel shares spindle with 1st channel in bichannel controller(only works in the 2nd channel);

//+1024 means in tool compensation interface press F8 to back main interface. Otherwise, back to user coordinate setting interface;

//+2048 means if tool offset changed or workpiece coordinate changed after move feed axis manually, display moving value as modified value.

//+4096 means when press Enter in main interface, pop up dialog for page No. input, then pop up user-defined dialog.

//+8192 means system occupied;

//+16384 for standby;

//+32728 means system occupied;

Notice: Entering into workpiece coordinate from tool compensation interface, F1-F6 correspond to workpiece coordinate G54-G59, F7 corresponds to G54.1-G54.48, press F7 again to choose from G43.1-G54.8. Input E to clear offset value. The setting here is the same as that in the parameter interface, workpiece coordinate system can be delete from configuration to save storage.

//+64 means record all program running time automatically;

The data will be saved in file “runtimerecord.txt”, it can be backed up with parameter files. Pressing Q in program management interface, a dialog will pop up to show program running time, and input **RECORD** in Other parameter P12 inner parameter to clear file “runtimerecord.txt”.

//+2048 means if tool offset changed or workpiece coordinate changed after move feed axis manually, display moving value as modified value. When program running, press Pause, and move feed axis through handwheel or button, then modify current workpiece coordinate offset in tool compensation or parameter interface, controller will display moving coordinate as offset value, press Enter to set current coordinate system offset vale, and it is also editable.

The dialog box definition is as follows:

Start from **BeginMainFace**, end with **EndMainFace**[optional]

Sample:

BeginMainFace // the start configuration.

0,20,80,420,300, Please enter the parameter<F1-Clear, F8-Null> //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;

4,3,3,280,190,fillpict1.bmp //4 means picture, following 4 bites are l/t/w/h, fillpict1.bmp is picture name;

1,282,10,60,25,X: //First bite means control type: button, following 4 bites are l/t/w/h, the last bite corresponds to the macro variable(only for edit box) or prompt information;

3,330,10,85,25,345,I //editable box, corresponds to Macro variables #345, and integer type (I specified integer, otherwise floating point type)

1,282,40,60,25,Z:
3,330,40,85,25,288, //corresponds to Macro variables #345, integer 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, helical pitch K(E):
3,300,195,90,25,145,
1,200,223,100,25,time L:
3,300,223,90,25,225,
2,30,253,100,26,Enter
2,240,253,100,26, Cancel
EndMainFace //configuration end.

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;

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.

In the main interface, press Enter button will pop up the dialog box is also defined as following:

1. In the main interface, press the Enter button, it will pop up dialog box, in the graph input interface and in the automatic mode, if press the "Start" button directly, then input is effective, and start to run the current program.
2. Multipages parameter can be displayed by PageUp and PageDown button. BeginMainFace or BeginMainFace0 means page zero， BeginMainFace1 means the First pages, BeginMainFace2 means the second page, BeginMainFace9 means the ninth page, there are 0-9 pages totally 10 user-defined pages.

For example:

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

4) The system interface is divided into the following parts:

F: The coordinates display area in main interface (lower left part) [including fa/fb/fc totally three interfaces]

A: The status display of the current program area in the main interface(upper left part).

B: T/G/M code display in any interface(right part).

Other parameter P900 "user defined interface [4 table section; 8 table all]" works when its value is 8 or 12, at this time, these user-defined interface is open, then user can open these interface through the following commands:

```
//-----------------------------
MyFaceFa=1        Open Fa interface
MyFaceFb=1        Open Fb interface
```

MyFaceFc=1	Open Fc interface
MyFaceA=1	Open A interface
MyFaceB=1	Open B interface

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

MyFaceFa=0	Close Fa interface
MyFaceFb=0	Close Fb interface
MyFaceFc=0	Close Fb interface
MyFaceA=0	Close A interface
MyFaceB=0	Close A interface

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

SubFaceB0=1 means open name display of current program in B interface;
SubFaceB1=1means open current G code and tool No. With tool offset display in B interface;
SubFaceB2=1 means open the current M code display in B interface;
SubFaceB3=1 means open current feed speed, spindle speed and rate display in B interface;
SubFaceB4=1 means open the current machine coordinate display in B interface;
SubFaceB5=1 means open current single piece time display in B interface
SubFaceB6=1 means open current processing number in B interface;
SubFaceB7=1 means open the current spindle speed in B interface.
//-----------------------------

SubFaceB0=0 means close name display of current program in B interface;
SubFaceB1=0means close current G code and tool No. With tool offset display in B interface;
SubFaceB2=0 means close the current M code display in B interface;
SubFaceB3=0 means close current feed speed, spindle speed and rate display in B interface;
SubFaceB4=0 means close the current machine coordinate display in B interface;
SubFaceB5=0 means close current single piece time display in B interface
SubFaceB6=0 means close current processing number in B interface;
SubFaceB7=0 means close the current spindle speed in B interface.
//-----------------------------

The definition of three interface Fa/Fb/Fc correspond to the F button’s displays as follows:

Function1= Relative coordinate

Function2= Workpiece coordinate
Function3= Comprehensive coordinate

The character corresponds to the button display can be modified;

The definition of system F interface[including Fa/Fb/Fc totally 3 interface]

Fa interface:

BeginFaceFa

0,5,160,455,370; The first character defines the regional border style; following figure defines the scope of boarder;

......

EndFaceFa

Fb interface:

BeginFaceFb

0,5,160,455,370; The first character defines the regional border style; following figure defines the scope of boarder;

......

EndFaceFb

Fc interface:

BeginFaceFc

0,5,160,455,370; The first character defines the regional border style; following figure defines the scope of boarder;

......

EndFaceFc

Definition of system interface A:

BeginFaceA

0,5,35,455,120; The first character defines the regional border style; following figure defines the scope of boarder;

......

EndFaceA

Definition of system interface B:

BeginFaceB

0,470,35,325,525; The first character defines the regional border style; following figure defines the scope of boarder;

......

EndFaceB

Note:

1> border style includes:

'D' means concave border;

'U' means convex border;

'F' means flat table border;

'0 ' means with no borders.

2> if the interface borders range exceeds the actual scope, software will limit it automatically.

The range of system interface F:left=5, top=160, right=460, bottom=530;

The range of system interface A:left=5, top=35, right=460, bottom=155;
The range of system interface B: left=470;Top=35; right=795; bottom=560;
The scope of the current running program name in the B interface: left=465, top=45, right=795, bottom=75;
The scope of G code, Tool No. and Tool offset in the B interface: left=465, top=75, right=795, bottom=165;
The scope of the current M code in the B interface: left=465, top=165, right=795, bottom=245;
The scope of current feed and spindle speed and speed rate in the B interface:left=465, top=250, right=795, bottom=340;
The scope of the current machine coordinates in the B interface: left=465, top=350, right=795, bottom=460;
The scope of current single piece time in the B interface: left=465, top=465, right=795, bottom=493;
The scope of the current processing No. in the B interface: left=465, top=496, right=795, bottom=524;
The scope of current spindle speed in the B interface: left=465, top=527, right=795, bottom=555;

3> Control type:

0/D/U/F 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.

9 means user-defined edit box which refresh automatically and dynamically

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 minimize the image space.
For user-defined edit box:

when the corresponding macro variable number equal to or greater than 20000, it means show the current coordinates or other status information, the definitions are as follows:
20001/20002/20003/20004/20005 corresponds to XYZAB axis machine coordinate;
20011/20012/20013/20014/20015 corresponds to XYZAB axis workpiece coordinate or absolute coordinates;
20201 means current status of spindle, will show " CW", "CCW", "Stop" status information.

5) File interfacecn.cfg(Chinese) and interfaceeg.cfg(English) Must be edited with Windows Notepad or on the CNC system to install.

4. 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.

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 parameter type,code and code in F1 parameter menu, no. behind G means parameter password protection level;

1,X direction (0 positive, non-0 negative) //contents of prompt dialog;

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

2,G70G71G72 default allowance(default as 20) //contents of prompt dialog; Parameter contents will be added colon 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.

The file starts from **BeginDwSelfMcode**, ends with E**ndDwSelfMcode**.

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, r means
                  red light, following is display message;
    b,m49,g,Tool tighten,     //b means the 2nd case, m49 means
                          condition is relay M49 invalid, g means
                  green 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 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;
    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 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;
```

```
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 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
                  red light, following is display message。
b,m57,g,Practice off,   //b means the 2nd case, m57 means
                          condition is relay M57 invalid, g means
                          green light, following is display message;
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 efit 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.

5. Robot controller palletizing parameter configuration

There are 3 files used to save palletizing parameter

(1) **palletbase.txt** saves the basic parameter of palletzing(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.

ApprRtret: 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) **paltlayout.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-OrgX: //X of the original point ;

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

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

P-OrgC: //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.

6. 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, G91incremental 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 defineds 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, G91incremental 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, G91incremental 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, G91incremental 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 defineds 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, G91incremental 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, G91incremental 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.

7. 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 decrpyt 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 encry., 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.

8. 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
                                 red light, following is display message;
   b,m49,g,Tool tighten,    //b means the 2nd case, m49 means
                                 condition is relay M49 invalid, g means
                                 green 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 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;
   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 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;
   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 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
                       red light, following is display message。
b,m57,g,Practice off,   //b means the 2nd case, m57 means
                       condition is relay M57 invalid, g means
                       green light, following is display message;
EndDwSelfMcode    //End
```

成都鑫科瑞数控技术有限公司
CHENGDU NEWKer CNC-TECHNOLOGY CO., LTD

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