User Manual

Macro of CNC System

V2.2

Shenzhen Guanhong Automation CO.,LTD
Website: www.szghauto.com
Add:Room 503 Anxin Building, No 536 Shenhui Road, Liuyue community, Henggang Street ,Longgang District, Shenzhen City,Guangdong Province, ChinaProvince, China
Post code: 518115

-Catalog-

Chapter 1 MACRO

1.1 Non-Mode Macro Command G65

Format: G65 P_ L_ A_ B_ C_ ......

Non-mode macro command G65 only work at current line , which is different to mode macro command(G66),which always work until macro cancel command(G67)

P_ : Specify name of macro program, E.g: P6000 , name of specified macro program is 6000 .

L_: Set times of call macro program

<A_B_C_... ...> : Argument , which is used for transfer data to macro varible(#**) , Transfering table is as following

Argument	Variable	Argument	Variable	Argument	Variable
A	#0	I	#7	T	#14
B	#1	J	#8	U	#15
C	#2	K	#9	V	#16
D	#3	M	#10	W	#17
E	#4	Q	#11	K	#18
F	#5	R	#12	Y	#19
H	#6	S	#13	Z	#20

Special Note: Address G , L , N , Q , P cannot used as arguments.

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 G00 X#2
N4 G00 Z#1
N5 M99      ; Return
```

1.2 Mode Macro Command G66/G67

G66 is mode macro command , G67 is cancel mode macro command

Format: G66 P_ L_ A_ B_ C_ ......

G67

G66 Mode macro command,which always call macro program until macro cancel command(G67)

P_ : Specify name of macro program, E.g: P7000 , name of specified macro program is 7000 .

L_: Set times of call macro program

<A_B_C_... ...> : Argument , which is used for transfer data to macro varible(#**) , the transfering table is same as above table.

Example:

```
Main Program : 4000
G00 X0 Z0
G66 P6000 L2 A5 B6
A8 B1
A9 B10
G67
M30
Macro Program: 6000
N1 #2=#0+#1
N2 IF (#2 EQ 10) GOTO 4
N3 G00 X#2
N4 G00 Z#1
N5 M99          ; Return
```

Chapter 2 M MACRO

2.1 User-defined M Macro Program(M881-M889)

M881-M889 , user-defined M macro instruction ,whose corresponding macro program is ProgramUser1 - ProgramUser9. And related M codes can call corresponding macro program. When we use M codes in processing program, CNC system will excutive inner codes of M macro program , which is similar to subprogram.

M codes	Macro Program	M codes	Macro Program
M881	ProgramUser1	M886	ProgramUser6
M882	ProgramUser2	M887	ProgramUser7
M883	ProgramUser3	M888	ProgramUser8
M884	ProgramUser4	M889	ProgramUser9
M885	ProgramUser5		

M Macro program(ProgramUser1 - ProgramUser9) must be edited on computer, and then copy to U-disk , and restore into cnc system.

2.2 Instructions of M Macro Program(M881-M889)

2.2.1 Output Instruction: OUT +/- Y/M

Explanation: Set output Y, M is valid or invalid

Attention: "+" means the output is valid ; "-" means the output is invalid;

Eg. : OUT+Y3:set output of Y3 is valid ; OUT-M28:set output of M28 is invalid.

OUT+Y5-Y7+Y9+Y11-Y15+M300 : set output of Y5, Y9, Y11 & M300 is valid, output of Y7 & Y15 is invalid.

2.2.2 Wait Instruction: WAT+/- X/Y/M

Explanation:Waiting for X/Y/M is valid or invalid; when the conditions are met, macro program can excutive following instructions.

Attention: "+" to means wait input is effective; "-" means wait input is invalid;

Eg.: WAT+X01-X02 : Wait X01 is valid,X02 is invalid,and then run following codes.

2.2.3 Delay Instruction: PAUS

Explanation: Instruction that used for delaying(0-99999ms), unit: ms ;

Eg.: PAUS100 : delay 100 ms .

PAUS1000 : delay 1 second

2.2.4 Assignment Instruction: =

Explanation: used for assignment of a variable

Eg.: #251=890.34 #450=#123

And also it could be mathematical expression , eg.: #440=#234+#470

2.2.5 Information Prompt Dialog

Format: MESSAGEBOX(InfoPrt1) , " InfoPrt1" is the prompt information.

2.2.6 Conditional Statement

Conditional statement: when condition is met, excutive codes after THEN, otherwise jump conditonal codes.

Format:
```
IF(**) THEN
       ENDIF
```

" ** " : auxilary relays(M), Input points(X) , Output points(Y) or macro variable(#*) .

+M** : When auxilary relay is valid, condition is met .

-M** : When auxilary relay is invalid, condition is met .

+X** : When input point X is valid, condition is met .

-X** : When input point X is invalid, condition is met

+Y** : When output point Y is valid, condition is met .

-Y** : When output point Y is invalid, condition is met .

+#** : When macro variable is 1(Non 0), condition is met .

-#** : When macro variable is 0, condition is met .

Eg.1 : When input X13 is valid , prompt dialog and exit.

```
IF(+X13) THEN
MESSAGEBOX(ERROR:CAN'T EXCHANGE TOOL!)
RETURN
ENDIF
```

E.g.2: When macro variable #313 is 1(non-0) , pop up dialog and exit.

```
IF(+#313)THEN
MESSAGEBOX(ERROR: COOLANT PUMP IS OVERLOAD!)
RETURN
ENDIF
```

2.2.7 Move Coordinate Axes

Format: MOVE(G_ , F_ , X_ ,Y_, Z_, A_, B_, W_) ;

1st Parameter: G_ , which could be G90 or G91 , G90: absolute programming ; G91: Incremental programming.Value of axes is absolute or incremental ;

2nd Parmaeter: F_ , which is set for feeding speed ;

3rd Parameter: X_,Y_,Z_,A_,B_, specify coodinate value of each axes ;

4th Parameter: W_ , which is set for input point ,which is met condition, to stop moving. Eg.: W+5, when input point X5 is valid,stop moving .

E.g.: MOVE(G90,F8000,X-100) ; //X axis moves to X-100 of machine coordinate

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

MOVE(G91,F8000,Y-300,W-8); //Y axis moves -300mm,when input point 8 is invalid,stop moving.

2nd & 3rd parameter can be specified by macro variable(#**)

Eg.: MOVE(G91,F#231,Y#240,W+2);//Y axis move #240 with #231 speed ; when input point X2 is valid, stop moving;

Suggestion: This codes could be used to tool setting and sets of tool compenstion with tool setting gauge.

2.2.8 Set & Save Current Workpiece Coordinate

Format: SETWK(X_ , Y_ , Z_ , A_ , B_) ;

Eg.: SETWK(X23.45,Z567.89); // set X-axis to 23.45 , set Y-axis to 567.89 at current coordinate system.

Value of XYZAB axis also can be specified by macro variable(#**) ;

Eg.: SETWK(X#238,Z#237); // at current coordiante system ,set value of X-axis to #238 , set value of Z-axis to #237.

2.2.9 Set & Save Current Machine Coordinate(G53)

Format : SETMH(X_ , Y_ , Z_ , A_, B_)

Eg.: SETMH(X23.56,Z567.89);// Set X-axis value to 23.56 , set Z-axis value to 567.89 at current machine tool coordinate ;

Value of parameters X_/Y_/Z_/A_/B_ can be specified by macro varible(#**).

E.g.: SETMH(X#238,Z#239) ; // Set X-axis value to #238 , set Z-axis value to #239 at current machine tool coordinate ;

2.2.10 Error & Exit

Format: ERREXIT

When macro program have errors, it will exit macro program

2.2.11 Return

Format: RETURN

When macro program is processed succefully,and use this code to exit macro program and return back.

Chapter 3　G MACRO

3.1 User-defined G Macro Program

Program Names of G120-G160 , user-defined G codes, are corresponding to ProgramG120 - ProgramG160 .

G Macro program(ProgramG120 - ProgramG160) must be edited on computer, and then copy to U-disk , and restore into cnc system.

3.2 Macro program instruction

3.2.1 Input Instruction: WAT

Waiting for the input port X valid or invalid instruction

Format: WAT+ (-) X

Attention: "+" to means the input is effective;

"-" means the input is invalid;

"X" means the input port X00-X55; see the I/O diagnosis;

3.2.2 Output Instruction: OUT

Set the output port Y is valid or invalid instruction

Format: OUT +(-)Y

Attention: "+" means the output is effective;

"-" means the output is invalid;

"Y" means the output port Y00-Y31; see the I/O diagnosis;

3.2.3 Assignment Instruction: =

Explanation: used for assignment of a variable

Eg.: #251=890.34　　#450=#123

And also it could be mathematical expression , eg.: #440=#234+#470

3.2.4 Unconditional Jump: GOTO n

“GOTO n” is the command that for jump to the program line that is specified by sequence number (N**) unconditionally. n is the sequence number.

E.g.: GOTO 5 ; // Jump to N5 program line.

Note: when specified program line , n , is beyond sequence number of N1-N99999, cnc system will hint error.

n , program line,could be macro variable (#**)

E.g.: GOTO #100

3.2.5 Conditional Jump

1) IF (Conditional express) GOTO n

If condition is met, execute GOTO n ,jump to N** program line; if the condition isnot met, execute the next segment.

```
Example:   N1 IF(#200 EQ 1) GOTO 20
           N10 G00 X0
           N20 G00 Z0
```

Explanation: If #200 is equal to 1, system will execute GOTO 20 , jump to N20 , and execute “G00 Z0”, if #200 isn’t equal to 1, system don’t execute operation of “GOTO 20” ,and will execute next segments , “G00 X0”,and then execute “G00 Z0”.

2) IF (Conditional express) THEN <A Expression>

<B opertional segment>

If condition is met, system execute A expression , and then execute B operational segment ; if condition is not meet, execute the next segment , B operation.

```
Example:   #101=0
           N1 IF(#100 EQ 1) THEN #101=1
           N2 IF(#101 EQ 1) GOTO 4
           N3 G00 X100
           N4 G00 Z100
```

Explanation: If #100 is equal to 1, system will execute “#100=1”, and then judge #101 is equal to 1 , jump to N4 & “execute G00 Z100” ; if #100 isn’t equal to 1, system will judge #101 also isn’t equal to 1 directly , and execute “G00 X100” & “G00 Z100”.

NOTE: 1.<A expression> normally is assignment statement.

2. <A expression> after THEN must exist, otherwise system will hint grammatical errors.

Prolongation:

```
3) IF(conditional express)
    <A operational command>
    ELSE
    <B operational command>
    ENDIF
    4) IF(conditional express)
        <A operational command>
        ELIF
        <B operational command>
        ENDIF
```

3.2.6 Loop Command

```
Format:   (Conditions Initialization)
            WHILE (conditional expression) DO n
            <A operational segments>
            [Alter condition of loop]
            END n
            <B operational segments>
```

When conditions are met during WHILE cycle command, execute the operational segments between DO n and END n . Otherwise,when condition isnot met, jump to the program line after END n ,also execute B operational segments.

We can nest for loops by placing one loop within another.

Note: 1.There must have operational codes that are for change condition at operational segments ,which is between Do n & END n. Otherwise system will enter endless loop.

2.Nesting of macro program loop statements of SZGH CNC system is 3 pcs of loops at most . Also n only could be 1 , 2 , 3 .

3.n of "DO n" & "END n" must keep same.

```
Example:   #100=2 #150=5 #200=25
             WHILE （#100 LT 3）DO 1
```

```
G00 X100
WHILE （#150 EQ 5）DO 2
G00 Y100
WHILE （#200 GE 20）DO 3
G00 Z100
#200=#200-2
END 3
#150=#150-1
END 2
#100=#100-1
END 1
```

3.3 Operators' meaning

Operator	Sign	Ex.	Operator	Sign	Ex.	Operator	Sign	Ex.
EQ	=	equal	GT	>	greater	LT	<	Less
NE	≠	unequal	GE	≥	G&E	LE	≤	L&E

3.4 Arithmetic & Logic Operation

Table:

Function	Format	Attention
Definition	#i = #j	
Addition Subtraction Multiplication Division	#i = #j + #k ; #i = #j - #k ; #i = #j * #k ; #i = #j / #k ;	

Sin
Asin
Cos
Acos
Tan
Atan
	#i = SIN(#j) ;
	#i = ASIN(#j);
	#i = COS(#j) ;
	#i = ACOS(#j);
	#i = TAN(#j);
	#i = ATAN(#j);
		90.5 degrees means 90 degrees & 30 points

Square root
Absolute value
Rounding off
Round down
Round up
Natural logarithm
Exponential function
	#i = SQRT(#j);
	#i = ABS(#j) ;
	#i= ROUND(#j);
	#i = FIX(#j);
	#i = FUP(#j);
	#i = LN(#j);
	#i = EXP(#j);	

Or Exclusive or And	#i = #j OR #k ; #i = #j XOR #k ; #i = #j AND #k ;	Executing with binary system

3.5 Message Hinting Box

Format: MSG(Parameter) OR MSG[Parameter]

Parameter is hinting message. After run this code, CNC will shift to Pause.

Format: STAF(Parameter) OR STAF[Parameter]

Parameter is hinting message. After run this code, CNC just hints messages , won’t shift to Pause, and keep on running.

Chapter 4 Variable

4.1 Local Variable

#0--#20 : local variables only can be used to store data in macro program, such as a result of operation, when power is off, the local variables are initialized to the empty. The argument assignment to the local variable when calling the macro program.

4.2 Global Variable

#21--#600 : Their meanings are the same in different macro program.

When power is off, the variable #21--#100 is initialized to zero, the variable #101--#600 data is saved not to loss even if the power is off.

4.3 System Variable

#1000-- : the system variables are used to change various data when reading the running CNC. For example, the current position and the compensation of tool.

Special Attention: macro variables #100--#155 and #190--#202 have been used by the system, users can not use.

4.3.1 System Parameter Variable

#1001--#1099 : Value of X-axis length compensation for T1--T99(Unit: um)

#1101--#1199 : Value of D1 radius compensation for T1--T99(Unit: um)

#1201--#1299 : Value of Y(C)-axis length compensation for T1--T99(Unit: um)

#1301--#1399 : Value of D2 radius compensation for T1--T99(Unit: um)

#1401--#1499 : Value of Z-axis length compensation for T1--T99(Unit: um)

#1501--#1599 : Value of D3 radius compensation for T1--T99(Unit: um)

#1601--#1699 : Value of A-axis length compensation for T1--T99(Unit: um)

#1701--#1799 : Value of D4 radius compensation for T1--T99(Unit: um)

4.3.2 I/O variables

#1800: X00-X07 (D0-D7) ; input resistor

#1801: X08-X15 (D0-D7) ; input resistor

#1802: X16-X23 (D0-D7) ; input resistor

#1802: X16-X23 (D0-D7) ; input resistor

#1803: X24-X31 (D0-D70 ; input resistor

#1804: X32-X39 (D0-D7) ; input resistor

#1805: X40-X47 (D0-D7) ; input resistor

#1806: X60-X67 (D0-D7) ; input resistor

#1807: X74-X81 (D0-D7) ; Alarm of driver/Spindle

#1808: Y00-Y15 (D0-D15) ; output resistor

#1809: Y16-Y31 (D0-D15) ; output resistor

#1810: Y32-Y47 (D0-D15) ; output resistor

#1811: M00-M15(D0-D15) ; auxiliary relay

#1812: X101-X112(D0-D11) ; input of operational panel

#1813: Y50-Y57(D0-D7) ; servo enable

#1814: M16-M31(D0-D15) ; auxiliary relay

#1815: M32-M47(D0-D15) ; auxiliary relay

#1816: M48-M63(D0-D15) ; auxiliary relay

#1817: M64-M79(D0-D15) ; auxiliary relay

#1818: M100-M115(D0-D15) ; auxiliary relay

#1819: M80-M95(D0-D15) ; auxiliary relay

#1820: M120-M135(D0-D15) ; auxiliary relay

#1850: X-axis Offset value of G54 workpeice coordinate system(unit:mm)

#1851: X-axis Offset value of G55 workpeice coordinate system(unit:mm)

#1852: X-axis Offset value of G56 workpeice coordinate system(unit:mm)

#1853: X-axis Offset value of G57 workpeice coordinate system(unit:mm)

#1854: X-axis Offset value of G58 workpeice coordinate system(unit:mm)

#1855: X-axis Offset value of G59 workpeice coordinate system(unit:mm)

#1856: Y-axis Offset value of G54 workpeice coordinate system(unit:mm)

#1857: Y-axis Offset value of G55 workpeice coordinate system(unit:mm)

#1858: Y-axis Offset value of G56 workpeice coordinate system(unit:mm)

#1859: Y-axis Offset value of G57 workpeice coordinate system(unit:mm)

#1860: Y-axis Offset value of G58 workpeice coordinate system(unit:mm)

#1861: Y-axis Offset value of G59 workpeice coordinate system(unit:mm)

#1862: Z-axis Offset value of G54 workpeice coordinate system(unit:mm)

#1863: Z-axis Offset value of G55 workpeice coordinate system(unit:mm)

#1864: Z-axis Offset value of G56 workpeice coordinate system(unit:mm)

#1865: Z-axis Offset value of G57 workpeice coordinate system(unit:mm)

#1866: Z-axis Offset value of G58 workpeice coordinate system(unit:mm)

#1867: Z-axis Offset value of G59 workpeice coordinate system(unit:mm)

#1868: A-axis Offset value of G54 workpeice coordinate system(unit:mm)

#1869: A-axis Offset value of G55 workpeice coordinate system(unit:mm)

#1870: A-axis Offset value of G56 workpeice coordinate system(unit:mm)

#1871: A-axis Offset value of G57 workpeice coordinate system(unit:mm)

#1872: A-axis Offset value of G58 workpeice coordinate system(unit:mm)

#1873: A-axis Offset value of G59 workpeice coordinate system(unit:mm)

#1874: X-axis Coordinate Value of 1st refenerce point (unit:mm)

#1875: X-axis Coordinate Value of 2nd refenerce point (unit:mm)

#1876: X-axis Coordinate Value of 3rd refenerce point (unit:mm)

#1877: X-axis Coordinate Value of 4th refenerce point (unit:mm)

#1878: Y-axis Coordinate Value of 1st refenerce point (unit:mm)

#1879: Y-axis Coordinate Value of 2nd refenerce point (unit:mm)

#1880: Y-axis Coordinate Value of 3rd refenerce point (unit:mm)

#1881: Y-axis Coordinate Value of 4th refenerce point (unit:mm)

#1882: Z-axis Coordinate Value of 1st refenerce point (unit:mm)

#1883: Z-axis Coordinate Value of 2nd refenerce point (unit:mm)

#1884: Z-axis Coordinate Value of 3rd refenerce point (unit:mm)

#1885: Z-axis Coordinate Value of 4th refenerce point (unit:mm)

#1886: A-axis Coordinate Value of 1st refenerce point (unit:mm)

#1887: A-axis Coordinate Value of 2nd refenerce point (unit:mm)

#1888: A-axis Coordinate Value of 3rd refenerce point (unit:mm)

#1889: A-axis Coordinate Value of 4th refenerce point (unit:mm)

#1890: Set of Metric/Inch in system

#1891: Name of 4th axis

#1892: Function Set of 4th axis

#1893: Rotary mode when 4th axis is as rotary axis

#1894: Display of relative coordinate when 4th axis is as rotary axis

#1895: Display of workpiece coordinate when 4th axis is as rotary axis

#1896: Reverse Backlash Compensatoin of X-axis (unit: um)

#1897: Reverse Backlash Compensatoin of Y-axis (unit: um)

#1898: Reverse Backlash Compensatoin of Z-axis (unit: um)

#1899: Reverse Backlash Compensatoin of A(4th)-axis (unit: um)

#1900: Feeding axis use electric gear(0: use , non-0: no)

#1901: Numerator of X-axis's electron gear (1—999999999)

#1902: Denominator of X-axis's electron gear (1—999999999)

#1903: Numerator of Y-axis's electron gear (1—999999999)

#1904: Denominator of Y-axis's electron gear (1—999999999)

#1905: Numerator of Z-axis's electron gear (1—999999999)

#1906: Denominator of Z-axis's electron gear (1—999999999)

#1907: Numerator of A-axis's electron gear (1—999999999)

#1908: Denominator of A-axis's electron gear (1—999999999)

#1909: Spindle use electric gear (0: use , non-0: no)

#1910: Numerator of electron gear ratio of spindle in Low grade (1—999999999)

#1911: Denominator of electron gear ratio of spindle in Low grade (1—999999999)

#1912: Numerator of electron gear ratio of spindle in Low grade (1—999999999)

#1913: Denominator of electron gear ratio of spindle in High grade (1—999999999)

#1914: Direction signal of X-axis(0: reverse , non-0: normal)

#1915: Direction signal of Y-axis(0: reverse , non-0: normal)

#1916: Direction signal of Z-axis(0: reverse , non-0: normal)

#1917: Direction signal of A-axis(0: reverse , non-0: normal)

#1918: Direction signal of Spindle servo(0: reverse , non-0: normal)

#1923: Max scope of X-axis in negative direction (mm / inch)

#1924: Max scope of Y-axis in negative direction (mm / inch)

#1925: Max scope of Z-axis in negative direction (mm / inch)

#1926: Max scope of A-axis in negative direction (mm / inch)

#1927: Max scope of X-axis in positive direction (mm / inch)

#1928: Max scope of Y-axis in positive direction (mm / inch)

#1929: Max scope of Z-axis in positive direction (mm / inch)

#1930: Max scope of A-axis in positive direction (mm / inch)

#1950: G00 Speed of X-axis (mm/min)

#1951: G00 Speed of Y-axis (mm/min)

#1952: G00 Speed of Z-axis (mm/min)

#1953: G00 Speed of A-axis (mm/min or degree/min)

#1954: Default speed of G01(mm/min)

#195: Simulate Speed(mm/min)

#1956: Max feeding speed(mm/min)

#1957: Acceleration of X-axis ((mm/min)/s)

#1958: Acceleration of Y-axis ((mm/min)/s)

#1959: Acceleration of Z-axis ((mm/min)/s)

#1960: Acceleration of A-axis((mm/min)/s)

#1961: Manual Speed of Spindle (rpm)

#1962: Manual Speed of Feedinga axis (mm/min)

#1963: Speed of X-axis return to reference point in positive direction (mm/min)

#1964 : Speed of Y-axis return to reference point in positive direction (mm/min)

#1965: Speed of Z-axis return to reference point in positive direction (mm/min)

#1966: Speed of A-axis return to reference point in positive direction (mm/min)

#1967: Speed of X-axis return to reference point in negative direction (mm/min)

#1968: Speed of Y-axis return to reference point in negative direction (mm/min)

#1969: Speed of Z-axis return to reference point in negative direction (mm/min)

#1970: Speed of A-axis return to reference point in negative direction (mm/min)

#1971: Max speed of spindle (rpm)

#1972: Transmission Ratio of spindle in low grade(Motor_Speed/SP_Speed)

#1973: Transmission Ratio of spindle in high grade(Motor_Speed/SP_Speed)

#1974: Starting running speed in manual (mm/min)

#1975: Allow jump speed when continuous track interpolation(mm/min)

#1990: Number of processing workpieces

#1991: Times of cycle auto when using M20 code

#1992: Delay time when loose/tighten tool (s)

#1993: Tight tool auto after loose tool(0: no , non-0: yes)

#1994: Selection of Lanuage

#1995: Direction of Q offset of G76 canned cycle

#1996: Direction of Q offset of G87 canned cycle

#1997: Width of track line in Diagram simulation (unit:pixel)

#1998: Mode of length compensation of tool

#1999: Type of spindle(0: servo spindle , Non-0: analog spindle)

#2000: Spindle is with Low/high grade (0: no , Non-0: yes)

#2001: Acceleration of spindle running normally

#2002: Turn on spindle when spindle change gear (0: no , Non-0: yes)

#2003: Time from stop to swing when spindle change gear (0.01s)

#2004: Time from swing to change gear when spindle change gear (0.01s)

#2005: Swing speed when spindle change gear (0.01 rpm)

#2006: Direction of starting swing when spindle change gear (0: CW, Non-0: CCW)

#2007: Time of swing CW when spindle change gear (0.01s)

#2008: Time of swing CCW when spindle change gear (0.01s)

#2009: Holding output singal that spindle change gear (0: no , Non-0: yes)

#2010: Hint user-defined alarm (0: no , Non-0: yes)

#2011: Hint alarm of feeding axis & spindle (0: no , Non-0: yes)

#2012: Hint alarm of coolant overload (0: no , Non-0: yes)

#2013: Hint alarm of coolant under-water (0: no , Non-0: yes)

#2014: Hint alarm of lubricate overload (0: no , Non-0: yes)

#2015: Hint alarm of lubricate under-oil (0: no , Non-0: yes)

#2016: Close function of 5 axis linkage interpolation (0: no , Non-0: yes)

#2017: Detect position feedback of spindle (0: no , Non-0: yes)

#2018: Use electric gear of spindle position feedback (0: no , Non-0: yes)

#2019: Control mode of spindle (G74/G84/G33 thread & spindle orientation)

#2020: Managment way of tool(0: use M06, Non-0: use T code directly)

#2021: Way of selecting tool(0: single direction, Non-0: both direction random)

#2022: Way of feeding axes return to reference point

#2023: Max length of detecting zero position when homing

#2024: Max no. of tool case in fixed tool area

#2025: Reference point of Z-axis return when changing tool

#2026: Stoping degreen when spindle orientation during tool change

#2027: Control mode when spindle homing

#2028: Power condition of each axis when booting

#2029: Max speed when Z-axis using handwheel (mm/min)

#2030: Max speed when X/Y/A-axis using handwheel (mm/min)

#2031: Detect SP_orientation & Point of changing point before change tool(Non-0: yes , 0: No)

#2046: stoping degree of spindle orientation when boring canned cycle

#2047: Homing Speed of spindle

#2048: Speed of spindle orientation

#2049: Allow Error Range of spindle orientation

#2050: Way of tool radius C compensation set up

#2051: Way of tool radius C compensation cancel

#2052: Position direction when SP-orientation adopt pulse interpolation control way

#2053: Delay time when spindle change direction suddenly (unit: 0.1s)

#2054: Acceleration of spindle servo ((mm/min)/s)

#2055: Active function of tool change

#2056: Transmission bits when communicate with RS232 serial port

#2057: Delay time between driver off of power & reboot (unit: 0.1s)

#2058: Interlock between motion of Z-axis and spindle take tool（0: No , Non-0:Yes)

#2059: Shifting distance after X-axis homing (unit:0.1mm)

#2060: Shifting distance after Y-axis homing (unit:0.1mm)

#2061: Shifting distance after Z-axis homing (unit:0.1mm)

#2062: Shifting distance after A-axis homing (unit:0.1mm)

#2070: Value of d quit tool in canned cycle G73 (mm)

#2071: Value of d quit tool in canned cycle G83 (mm)

#2100——#2199: Tool number of 0-99 tool case

#2032~#2045: Spare #2063~#2069: Spare

Format:#i=Expression

Could read the value, for example: #200=#1003; To read the X axis offset value of the third tool into macro variables #200.

Could modify the value, for example: #1003=23000; To modify the X axis offset value of the third tool to 23000 micron.

#1003=#1003+50; To increase the X axis offset value of the third tool 50 micron.

Appendix I: Example of User-Defined M Macro Code

(1) Automatically Tool Setting

Macro variable of M880

X25: Input point , for detecting position of tool setting .

#380: Machine coordinate Value of X-axis Original Position Point for setting tool.

#381: Machine coordinate Value of Y-axis Original Position Point for setting tool.

#382: Machine coordinate Value of Z-axis Original Position & Return Point.

#383: Speed in negative direction of setting tool; (unit: mm/min).

#384: Speed in positive direction of setting tool; (unit: mm/min).

#385: Coordinate Value of Z-axis of surface of workpiece at current coordinate system after setting tool.

#386: Speed of moving to position point (unit: mm/min).

#387: Mode of Automatially Tool Setting (1: Fixed point for tool setting 0:Float point for tool setting) ; Fixed point for tool setting means that tool setting gauge is put at fixed position , XYZ axis will move to original position point before tool setting firstly at each time; Float point for tool setting means that Z-axis will move in negative direction to search for signal of tool setting signal.

#388: Min coordinate value in negative direction of Z-axis .

#389: Drop value of Z-axis,also height between surface of tool setting gauge & surface of workpiece .

#1925:Max scope of Z-axis in negative direction.

#1929:Max scope of Z-axis in positive direction.

#5023: Curren machine coordinate value of Z-axis.

M Macro Program of " ProgramUser0 " as following:

```
STATUSINFO(Operating Tool Setting!);
IF (+#387) THEN;
MOVE(G90,F#386,Z#382);
MOVE(G90,F#386,X#380,Y#381);
```

```
ENDIF;
#50=#5023;
MOVE(G90,F#383,Z#388,W+25);//#1925
PAUS160;
IF (-X25) THEN
MESSAGEBOX(Error: Cannot detect signal of tool setting gauge!)
ERREXIT;
ENDIF
MOVE(G90,F#384,Z#1929,W-25);
IF (+X25) THEN
MESSAGEBOX(Error:Cannot off of signal of tool setting gauge!)
ERREXIT;
ENDIF
#51=#385+#389;
SETWK(Z#51);
MOVE(G90,F#386,Z#50);
MESSAGEBOX(Tool Setting Automatically Succefully!!)
RETURN;
```

(2) M882

```
STATUSINFO(Auto Set Drop Value of Z-axis of Tool Setting );
IF (+#387) THEN;
MOVE(G90,F#386,Z#382);
MOVE(G90,F#386,X#380,Y#381);
ENDIF;
MOVE(G90,F#383,Z#388,W+25);//#1925
PAUS160;
IF (-X25) THEN
MESSAGEBOX(Error:Cannot detect signal of tool setting gauge!)
ERREXIT;
```

```
ENDIF
MOVE(G90,F#384,Z#1929,W-25);
IF (+X25) THEN
MESSAGEBOX(Error:Cannot off of signal of tool setting gauge!)
ERREXIT;
ENDIF
#50=#5023;
MESSAGEBOX(Move Tool Tip to Surface of Workpiece & Run M883 code!)
RETURN;
```

(3) M883

```
#389=#50-#5023;
MESSAGEBOX(Drop of Z-axis Set Well for setting tool!)
RETURN;
```

(4) Steps of Automatically Tool Setting

① Set P380~P388 in Other Parameter

② Set Drop of Z-axis of tool setting automatically

a. Run M882 command in MDI , Prepare set drop

b. Move tool tip of Z-axis to surface of workpiece manually

c. Run M883 command in MDI, set drop of Z-axis automatically(P389)

③ Select workpiece coordinate system(G54~G59)

④ Automaticaly Tool Setting: Run M880 in MDI to set offset of Z-axis at current workpiece coordinate system.

Appendix II: Example of User-Defined G Macro Code

For example, defines the G152 function: the arc model porous drilling cycle. (must copy the macro program ProgramG152 into system).

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

X: The X coordinate with absolute value or incremental value of center to specify.

Y: The Y coordinate with absolute value or incremental value of center to specify.

Z: Hole depth

R: Approaching fast to the point coordinate

F: Cutting feed speed

I: Radius

A: The angle of the first hole

B: Incremental angle specify(CW when negative)

Macro program ProgramG152 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
```