

Soft Servo Macro Specifications

Summary of Variable Types

- One undefined variable (null: #0)
- 99 local variables (#1 – #99)
- 400 numbered global variables (#100 – #499) of global scope (their values are lost when the control restarts)
- An unlimited number of symbolic global variables, with meaningful variable naming (such as “#position”). Symbolic global variables must be all lower case letters (#axxx – #zxxx)
- 500 numbered permanent variables (#500 – #999) of permanent scope (their values remain even when the control restarts)
- Up to 24,000 system variables (depending upon the no. of axes, no. of tool offsets, no. of workpiece coordinate offsets, etc.)

Flow of Control — Branching and Repetition

Control Command	Statement Format
Conditional branch	IF [<conditional expression>] GOTO n
Unconditional branch	GOTO n
Conditional execution	IF [<conditional expression>] THEN
Conditional execution with branching	IF [<conditional expression>] ... ELSE ... ENDIF
Conditional loop	WHILE [<conditional expression>] DO n ... END n

Notes:

- 1) Unlimited nesting of sub loops is allowed.
- 2) n is a direct number – NXXX (a line number) or variables can also be used.

Macro Calls

- Simple macro call: G65 P_<Macro Number>L_<Argument>
(The macro must be in the same file as the main program.)
- Sub program call:
M98 P_<Sub Program Name> R_<Repetition Number>
(The subprogram must named O<Sub Program Name>.dat and must be in a separate file located in the same folder as the main program.)
- Custom Macro Calls Using G, M, S or T Codes:
Examples: G34, M3.1, T20, S1000
(You must use the macro parameters to associate G, M, S and T codes with specific macro program files. The macro file can be anywhere, including on a network.)

Mathematical and Logical Operations

Operand	Format(s)
Assignment	#A = #B
Sum	#A = #B + #C
Product	#A = #B * #C
Difference	#A = #B - #C
Quotient	#A = #B / #C
Exponent	#A = #B ^ #C
Or	#A = #B or #C
And	#A = #B and #C
Xor	#A = #B xor #C
Not	#A = not #B, #A = !#B
Sine (degree)	#A = sin [#B]
Cosine (degree)	#A = cos [#B]
Tangent (degree)	#A = tan [#B]
Arcsine	#A = asin [#B]
Arccosine	#A = acos [#B]
Arctangent	#A = atan [#B]
Absolute value	#A = abs [#B]
Natural logarithm	#A = ln [#B]
Exponential function	#A = exp [#B]
Square root	#A = sqrt [#B]
Rounding off	#A = round [#B]
Rounding down	#A = fix [#B]
Rounding up	#A = fup [#B]

Comparison Operators

Operator	Meaning	Acceptable Formats
EQ (=)	Equal to	(#A EQ #B) , (#A = #B)
GT (>)	Greater than	(#A GT #B) , (#A > #B)
GE (≥)	Greater than or equal to	(#A GE #B) , (#A >= #B)
NE (≠)	Not equal to	(#A NE #B) , (#A != #B)
LT (<)	Less than	(#A LT #B) , (#A < #B)
LE (≤)	Less than or equal to	(#A LE #B) , (#A <= #B)

System Variables

Variable Type	Range	Notes
Input from PLC (by bit)	#1000 - #1015	Read only
Input from PLC (16 bit word)	#1032	Read only
Output to PLC (by bit)	#1100 - #1115	Read/Write ⁵
Output to PLC (16 bit word)	#1132	Read/Write ⁵
Output to PLC (32 bit dword)	#1133	Read/Write ⁵
Alarm ²	#3000	Read/Write ⁵
1 millisecond timer	#3001	Read only
G-Code modal groups	#4000 - #4031	Read only
B Code	#4102	Read only
F Code	#4109	Read only
H Code	#4111	Read only
M Code	#4113	Read only
Seq Number	#4114	Read only
S Code	#4119	Read only
T Code	#4120	Read only
Block end Point Position	#5001 - #5008	Read only
Current Position	#5021 - #5028	Read only
Current Position (Work)	#5041 - #5048	Read only
External work compensation ¹	#5201 - #5208	Read only
Work Coordinate 1	#5221 - #5228	Read/Write ⁵
Work Coordinate 2	#5241 - #5248	Read/Write ⁵
Work Coordinate 3	#5261 - #5268	Read/Write ⁵
Work Coordinate 4	#5281 - #5288	Read/Write ⁵
Work Coordinate 5	#5301 - #5308	Read/Write ⁵
Work Coordinate 6	#5321 - #5328	Read/Write ⁵
I/O Information ¹	#6000 - #6999	Read only
Servo Drive Control ^{2,3}	#7000 - #7999	Read only
Parameter Setting ¹	#8000 - #9999	Read only
Tool length wear compensation ⁴	#10001 - #10999	Read/Write ⁵
Tool length geometry compensation ⁴	#11001 - #11999	Read/Write ⁵
Tool radius wear compensation ⁴	#12001 - #12999	Read/Write ⁵
Tool radius geometry compensation ⁴	#13001 - #13999	Read/Write ⁵

1. ServoWorks S-100T only
2. Under development
3. Hardware dependent
4. ServoWorks MC-Quad and ServoWorks S-100M Series only
5. Limitation applies: contact Soft Servo Systems for details.

Referencing a Variable

Type of	Examples
Direct reference	#1=1. #position = 23.0 G01X#1Y#position → G01X1.Y23.
In a formula or conditional expression	#1=1. #2=2. G01X[#1+#2].

Processing a Null Variable

Any variable (local, global, permanent or symbolic) which has not been initialized with a value is undefined, or equal to the null variable (#0). A null variable is processed differently, depending upon whether it is used in a formula, in a movement command, or in a conditional expression.

When a null variable occurs...	Null variable is treated....	Examples
In a mathematical formula	As a zero (0.0)	#1=1. #2=#0. #3=#1+#2. → #3=1.0+0.0. #4=cos[#2]. → #4=cos[0.0]. #5=cos[#1+#2]. →
In a movement command	As if it weren't there (i.e., it is ignored)	#1=1. #2=#0. G01X#1Y#2 → G01X1
In a conditional expression	As a zero, except for EQ (equal to) or NE (not equal to)	#1=0. #2=#0. IF[#1 GT #2] → IF[0 GT 0] → FALSE IF[#1 EQ #2] → IF[0 EQ Null] → FALSE

NOTE: “→” indicates “is evaluated as.”

Argument Specifications

Argument specifications map argument addresses to variables. The argument assignment protocol for Soft Servo macros uses all letters once except the letters G, L, O, N and P.

Two examples follow:

G65 P9000 A1.0 B2.0 C3.0
G65 P9000 #1=1.0 #2=2.0 #3=3.0