



Macro Programming Information

Scan code to get
the latest version
of this document



Translation
Available



Introduction

This document tells you how to use macros on the Haas control. It tells you about:

- The macro parameter
 - Macro definitions
 - The reasons to use macros
 - Macro statements
 - G and M codes
 - Settings
 - Variables
 - Program syntax
 - How to alias a G or M code
 - How to write a basic macro program
-

Macro Parameter

Parameter **57:22**, **ENABLE MACRO**.

The control needs an option code to activate this parameter.

To enable macros, set Parameter **57:22** to **1**.

When this parameter is set to **0**:

- programs that use macro statements can not be loaded from USB, RS-232 or other external devices
 - macro statements will be converted to comments with a question mark. Example: (#500=#500+1 ?). These macro statements will be ignored by the control.
 - macro statements can not be entered or edited from the keypad, the control will give an alarm.
-

Macro Definition

A macro is a routine or sub-program that includes non-G-code commands. It is typically used for common operations that will be called many times in a program.

Macro programming can be compared to any computer programming language like BASIC, C Language, and PASCAL. But the macro programming language is in the CNC control.

You can access macros at the G code level. You can combine manual programming and parametric programming

techniques.

Computer-related features like variables, arithmetic, logic statements, and looping are available.

Why Use Macros?

A G-code program is rigid in structure and can not be altered in mid- operation. Offsets are the only means to adjust the machine path from one run to the next.

Macros add flexibility to the control that are not possible with standard G-code programming.

With macros you can set up families of parts, create custom canned cycles, program complex motions and drive optional devices.

Macro statements

A macro statement can assign a value to a variable or read a value from a variable.

- Example: #500=21. will set variable 500 to a value of 21
- Example: #100=#101 will set the value of variable 100 to the value stored in variable 101

A macro statement can evaluate an expression.

- Example: #100=1+3 will set variable 100 to a value of 4.
- Example: #100=#100+1 will set variable 100 to a value of 1 greater then the value of variable 100 before this line of code is executed.
- Example: #100=SQRT[9] will set variable 100 to the square root of 9.

These types of macro statements can be used to perform trigonometry, round numbers, perform simple arithmetic, get the square root of a number, and for many other uses. Refer to the Haas Operators manual for a full list of the math functions that can be used with this type of macro statement.

A macro statement can unconditionally branch to another point within a program.

- Example: GOTO900 will cause the program to branch or jump to the next N900 in the program.
- Example: M99 P815 will cause the program to branch or jump to the next N815 in the program.

A macro statement can conditionally branch to another point within a program. This is sometimes called a Boolean function.

- Example: IF [#100 GT 6] GOTO103 will cause the program to branch or jump to the next N103 in the program, if the value in variable 100 is greater than 6.

Here is a list of the six Boolean operators:

- EQ -- Equal to
- NE – Not equal to
- GT – Greater than
- LT – Lees than
- GE – Greater that or equal to
- LE – Less that or equal to

A macro statement can conditionally repeat some section of program with a WHILE, DO, END statement.

- Example:

```

G103 P1 ( limit look ahead )
WH [ #100 LT 6 ] DO1 ( start WHILE command )
#100= #100+1 ( increment variable 100 by 1 )
END1 ( end WHILE command )
M30

```

This program will loop through the section of program from the WHILE to the END lines and increment variable 100 by a value of 1, as long as the value in variable 100 is less than 6. If the value in variable 100 is 6 or greater, the program will continue on the line after the END line.

Useful G and M Codes

- **M97 Pxx Local sub program call** . This code is used to call a subroutine, referenced by a line number (Nxxx) within the same program.
- **M98 Pxx Sub program call** . This code is used to call a subprogram, referenced by a program number (Oxxx).
 - Example G65 P9000 A3. B20. C1. This line of code will call sub program 9000 and set variable #1 to a value of 3, set variable #2 to a value of 20. and set variable #3 to a value of 1.
 - For more information on how these variables (arguments) are passed, please see Macro Arguments in the Macro section of a Haas Operators manual.
- **G103 P1 Block look ahead limit** . This code is used to limit the look ahead in a program. It is necessary to limit the look-ahead in most macro programs because the Haas control has 80 blocks of look ahead, and the macro expression will be performed at the look ahead time. This can cause problems in macro programs that have short logic sections within short machine motion sections, or short macro programs.
 - Without the G103 P1 command, macro expressions will be performed 80 blocks ahead of the program's motion commands.
 - With the G103 P1 command, macro expressions will be performed 1 block ahead of the program's motion commands.
 - Here is an example of how Look Ahead can cause problems with a short program.
 - If we set variable 100 to a value of 0 and run the following short program in single block: WH [#100 LT 6] DO1 #100= #100+1 END1 M30
 - When **[CYCLE START]** is pressed the first time, look ahead performs the macro expressions for the next 80 lines of program, in this case looping through the program several times. This will cause variable 100 to have been incremented to a value above 6 and the program will end.
 - If a G103 P1 with four End Of Blocks is added to the start of this program, it can be run in single block and the machine look ahead will not cause a problem.
 - G103 without a Px will cancel block look ahead.
 - Please see the operators manual for more information about this G code.
- **M109 Interactive user input** . This M code allows the program to place a short prompt or message on the screen. Then the machine operator can input an answer to the prompt or message and branch or alter the next part of the program based on this answer.
 - Example: this M code can be used in a program to stop the program and prompt the operator to measure a bore in the part, based on the operators answer to a simple question.
 - **Is bore #1 smaller then 1.0" (Y / N)** . The machine can either offset the tool that made bore #1 and re-run bore #1 (Y input) or continue on with the program (N input).
 - Refer to the operators manual for more information about this M code.

Macro Related Settings

- Setting **23, 9xxx PROGS EDIT LOCK** .
 - When this setting is **OFF**, the machine operates normally.
 - When this setting is **ON**, programs 9000 to 9999 (usually macro programs) are invisible to the operator and can not be uploaded or downloaded. They can not be listed, edited or deleted.
 - It is recommended to leave this setting **ON** if the machine has probing.
- Setting **74, 9xxx PROGS TRACE** .
 - When this setting is **OFF**, the control will not display blocks of code that are executed in a program 9000 or above.
 - When this setting is **ON**, the control will display all blocks of code that are executed in a program.
- Setting **75, 9xxx PROGS SINGLE BLK** .
 - When this setting is **ON** and the control is operating in **SINGLE BLOCK** mode, the control will stop at each block in 9000 series programs and wait for the operator to press **[CYCLE START]**.
 - When this setting is **OFF**, all blocks in 9000 series programs are executed in a continuous manner even with **SINGLE BLOCK ON**.

Variables

Local Variables

Local variables range between #1 and #33.

When a macro sub program G65 command is executed, any arguments on the G65 line are passed to the local variables.

- Example: G65 P9000 C16. will execute program 9000 and set variable 3 to a value of 16.

These variable are for temporary use and are cleared at the end of the program.

Variable/Address/Alternate	1/A/	2/B/	3/C/	4/I/	5/J/	6/K/	7/D/I	8/E/J	9/F/K	10/ /I	11/H/J
Variable/Address/Alternate	12/ /K	13/M/I	14/ /J	15/ /K	16/ /I	17/Q/J	18/R/K	19/S/I	20/T/J	21/U/K	22/V/I
Variable/Address/Alternate	23/W/J	24/X/K	25/Y/I	26/Z/J	27/ /K	28/ /I	29/ /J	30/ /K	31/ /I	32/ J	33/ /K

Global Variables

Global variables are accessible at all times.

Global variables occur in three ranges: 100-199, 500- 699 and 800- 999.

Variables 100- 199 and 500- 600 are reserved for Haas use and are typically used by probes or pallet changers.

Variables 600- 699 and 800- 999 are available for user macro programs.

Global variables remain in memory when power is turned off.

System Variables

System variables give the programmer the ability to interact with several control conditions.

Control functions can be modified by setting a system variable, .

The functions behavior can be modify based on the value in a system variable.

Some system variables have a Read Only status. The programmer cannot modify them.

System variables can be used to change offsets, turn on or off discrete outputs, check discrete inputs, check parameters and settings.

Refer to the operators manual for more information on system variables.

Program Syntax

Parentheses

Parentheses are used to enclose comments. These comments are ignored by the machine.

- Example: G01 X1.0 (COMMENT)

Brackets

- Brackets are used to control the order of execution of expressions within a macro program and to enclose MACRO expressions.
- Example : #100= [3* [9 + 3]] in this line of code the 9+3 is calculated before the x 3 so #100 has a value of 36.

Line Numbers

- Line numbers are a way of assigning a label to a block.
 - Line numbers can be used with sub-routine calls. Nnn indicates the target of a M99 Pnn, M97 Pnn, or GOTOnn statement.
-

Aliasing a G or M-Code

- Aliasing is the act of assigning a name (G-code) to a specific program.

Macros are typically a subprogram, not a stand-alone program.

They are called via G65 or M98.

This subprogram call can be replaced with a single M- or G-code.

The assignment of this new code to a program takes place through parameters 81-100.

Only programs O9000 to O9019 may be aliased.

Refert to the Operators manual for more information on aliasing g- codes.

Parts Counter Macro Example

O00001

G103 P1

;

;

;

;

#800= xxx (NUMBER OF PARTS)

N100

IF [#801 GE #800] GOTO200

#801= #801 + 1

(ADD MAIN PROGRAM)

IF [#801 LE #800] GOTO100

N200

#801= 0

M30