Journal of Material Science and Mechanical Engineering (JMSME)
p-ISSN: 2393-9095; e-ISSN: 2393-9109; Volume 3, Issue 3; April-June, 2016 pp. 250-253
© Krishi Sanskriti Publications
http://www.krishisanskriti.org/Publication.html

A Conceptual Framework of Common Variables in CNC Machines Programming for Fanuc Custom Macros

Mohd Asif Hasan

Department of Mechanical Engineering, University Polytechnic, Faculty of Engineering and Technology,
Aligarh Muslim University (AMU), Aligarh - 202002 (India)
E-mail: [1]hasan_in@hotmail.com, [2]asif.hasan.bp@amu.ac.in

Abstract—*Computer Numerical Control (CNC) machines are widely used in production industry. However, there fullest utilization is yet to be achieved. In order to make use of these machines to their full potential require an understanding of their hidden or not so well known capabilities. One such capability is Macros. Macros are simple part-programs which reside in the memory of the controller and are called using a specific code for the macro. All sort of canned cycles are basically Macros. Fanuc offers in its CNC controllers the optional feature to program CNC Machines using Macros which is also known as Parametric Programming. In order to develop Macros, Fanuc offers Custom Macro B which is similar to any computer programming language. Custom Macro B offers many CNC related features in addition to features of any computer programming language. Custom Macro B can be accessed at G and M code level, which means that it can be combined with manual programming techniques of CNC machines. Although, Subprograms are the first logical step into the macro development, the major difference between the two unique programming methods is the flexibility macros offer. To achieve this flexibility, Macros make extensive use of Variables. Local Variables and Common Variables are two such variables which are primarily used for the development of Macro programs. In order to address an inherent limitation of Local Variables in some situations, Common Variables are used for Macro development. This paper discusses use of Common Variables for the development of Macros in particular and a conceptual framework for the use of variables in Macros in general.*

Keywords*: Computer Numerical Control; CNC; CNC Programming; Macro; Macro Programming; Parametric Programming; CNC Variables; Common Variables; Custom Macro B.*

1. INTRODUCTION

Computer Numerical Control (CNC) machines are those machines which are controlled by numbers while making use of computers for processing the information fed to these machines. The numbers used for controlling these machines are actually alpha-numerals and are popularly known as Codes of CNC Machines e.g. G Codes, M Codes, etc. These codes are arranged in a logical sequence depending on the type and sequence of operations to be performed on a job. This logical sequence of CNC codes is known as Programme of Instructions or simply Part Program. These part programs are fed to the Controller which is the brain of the CNC machine, for processing and execution of these instructions and referred to as CNC Programming [1, 2, 9].

CNC programming using Macros is referred to as Parametric Programming or Macro Programming or simply Macros. Macros are simple part-programs which reside in the memory of the controller and are called using a specific code for the macro. All sort of canned cycles are basically Macros. CNC programming using Macros can be compared to any computer programming language like BASIC, C Language, and PASCAL. However, this programming language or feature is available right in the CNC controller and can be accessed at G code level, which means that it can be combined with manual programming techniques of CNC machines. As it is like any computer language and thus also possesses Computer-related features like variables, arithmetic, logic statements, and looping. Like computer programming languages, this feature of CNC programming using Macros also comes in several versions. The most popular is Fanuc's Custom Macro B (used by Fanuc and Fanuc-compatible controls) [1-7, 10].

Many control manufacturers (including Haas, Mistubishi, Mazak, Yasnac, and Seikos) use Custom Macro B as their version of CNC programming using Macros. Although for most CNC controls, Custom Macro B is an option for CNC programming using Macros but it doesn't come standard with the controller. However, many machine tool builders include Custom Macro B in the standard package of options they include with the machines they sell, especially if the machine has some special accessory like a probing system. In addition to computer related features, Custom Macro B could also be used for extensive CNC related features which are more commonly related to the utilities such as part counters, tool life managers, etc. and driving accessory devices such as probes, in-process and post process gauging systems, etc. [5, 6, 7]. A Part Program developed using Macros for one control model may not work with another control model. However, the logic and general approach discussed here can be adapted for various models of Fanuc controller and also for control systems other than Fanuc.

The optional feature of Macro Programming available with Fanuc Control Systems is called the Custom *Macro* or the *User Macro*. Typically, a letter *B* is added to the description, such as *Custom Macro B* or *User Macro B* which is just an indication of a level more advanced from the original version. Virtually all Fanuc controls now offer the optional *Macro B* version, even if it is not specified directly in the control description. In order to determine if a given control has Custom Macro B, one universal way is to perform a simple test in the Manual Data Input (MDI) mode of the CNC machine. In the MDI mode, simply enter and execute the command:

```
#101=1
```

One may expect this command to be invalid as most Fanuc controls require terminating MDI commands with a semicolon (;). But, if the control has Custom Macro B, it will execute this command without generating an alarm. If it does not, an alarm will be sounded something like “syntax error” or “unrecognizable address". Additionally, if the machine has Custom Macro B, one should be able to find a display screen that shows the Custom Macro variables. With Fanuc controls, it will either be in the OFFSET or SETTING display screen pages.

Some of the areas where CNC programming using Macros (or simply Macro Programming) is indispensable or at least offer an edge over other methods are Part Programming for Family of Parts, developing Special G and M codes, Canned Cycles and Complex tool motions, Offset Control, Probing and Gauging.

2. MACROS AND VARIABLES

An in-depth knowledge of the Preparatory Commands (G-codes), Miscellaneous Functions (M-codes) and other CNC codes and commands along-with their use in a part-program is indispensable for the development of a Macro Program or simply Macro. A macro program developed using Custom Macro B or else resembles a standard CNC program to a certain extent, but includes many features not found in regular programming. Essentially, a macro program is structured as a regular subprogram. It is stored under its own program number (**O-**), and it is called by the main program or by another macro, using a G-code (typically **G65**). However, in a very simple form, macro features can be used in a single program as well, with out the macro call command [3].

The most noticeable feature of Macros is the use of Variables for storing numerical values. A variable is a mathematical quantity that can assume any value within its allowed range and format. The word variable means change or changeable. As the data that may change is stored in the variables forms the basis of flexibility in Macros. In macros, variables can be used instead of real numerical values and they can be treated like algebraic variables for various mathematical operations [3, 10], for example, by adding two variables together, to get yet another value.

Variables are expressed in the program by a numerical value preceded by the pound sign for example #100. Values can be input manually into a variable register. Values can also be assigned via the part program, for example #100=5.5, #101= #100+1. Variables can only be Mathematical Quantities i.e. numbers. Macro Programming makes extensive use of Formulas, called Expressions, involving these variables. These formulas more or less work in the same fashion as algebraic formulas. While making use of these expressions, the following important procedures among others have to be followed: (a) Make use of square brackets ("[" and "]") instead of parenthesis, and (b) the standard operators are add ("+"), subtract ("-"), multiply ("*" asterisk and not "x"!), and divide ("/"). For Example: #6 = [#4 * #2/#1] + #3-#5.

In macro programming the flow of execution of part program can be done in different sequence by conditional instructions e.g. *IF..THEN..* , GOTO.., etc. In addition, a multiple repetition of the part program segment is possible using *REPEAT* loop, *WHILE* loop etc. [10]. A comprehensive detail of Fanuc’s Custom Macro B Variables including types of variables, arguments, arithmetic commands, control command, macro call, etc., can be accessed from [3, 8]. However, there is a list of other operators and functions that may be somewhat control-dependant. Make sure to check machine manual to confirm if the control supports them and if so just how they work.

Variables in Custom Macro B are fixed and designed meticulously to obtain maximum leverage. For a better understanding of variables, it is important to understand the various types of variables, their differences and respective applications. In Custom Macro B, there are the following four different categories of variables, called the variable types: (a) NULL variable (#0) (b) LOCAL variables (from #1 to #33) (c) COMMON or Global variables (from #100 to #149 and from #500 to #531) and (d) SYSTEM variables (#1000 and up). However, the thrust of this paper is on the Common Variables.

3. COMMON VARIABLES

The word “Common” means “Shared” which defines the function of the Common Variables. The use of Common or Global Variables is appreciated when a macro programmer is in need of sharing the values of a defined variable a number of times for different applications. In contrast to Local Variables, the values of Common Variables could be shared among several macros or a number of times for different applications. Although, this shortcoming of local variables could be easily compensated by making use of nesting but nesting in a macro presents its own merits and limitations. Typically, the local variables can be defined in the macro call G65 or G66, or in the main program itself, whereas the common variables can be defined in the body of the program, either in the main program, or in the macro and are never assigned as arguments in the G65 macro call.

Fanuc offered the common variables with the special objective that they remain active even when the macro they were defined in is completed and to avoid the problem when the local variables have been cleared and there still is the need to pass one or more variable values to another macro.

Before making any use of common variables, it is essential to understand how the common variables work (refer specific manuals for differences between control models), and also when and how the common variables are cleared [3]. They start with the first common variable - #100. There is another range of common variables, one that starts with the common variable #500. In the first range, the maximum number of common variables could be from #100 to #199, and in the second range, the maximum number of common variables could be from #500 to #699. However, the actual number of common variables in these two ranges varies on the various control systems of the Fanuc. There are the following four options offered by Fanuc for these two ranges of Common Variables on its various control systems: (i) #100 to #149 and #500 to #549 (Option A), (ii) #100 to #199 and #500 to #599 (Option B), (iii) #100 to #199 and #500 to #699 (Option C), and (iv) #100 to #199 and #500 to #999 (Option D).

The difference between the common variables range starting from #100 and the other range starting from #500 is very significant. Variables from #100 to #199 are cleared when the power of the control system is turned off and are therefore known as Volatile Group or Non-Holding group of Common Variables. Variables from #500 to #999 remain in effect even when the power to the control system has been turned off and are therefore called Non-Volatile Group or Holding Group of Common Variables. In contrast to local variables, common variables are not cleared by M99 or M30 functions. The higher level common variables (#500 to #999 range) may be cleared one by one in Manual Data Input Mode (MDI) mode by equating them to a NULL variable, for example, #505 = #0 will clear the variable #505. However, a better and easy way to clear a range of such variables is running a specially design macro program. Following is an example macro program (O8008) that could be used to clear higher level common variables in the range from #551 to # 999:

```
O8008
#33 = 551
WHILE [#33 LE 560] DO1
#[#33] = #0
#33 = #33+1
END1
M99
%
```

The macro O8008 could be suitably adapted to clear a different range of variables by appropriately assigning the initial value of the range of variables to be cleared to the local variable #33 which serves as a Counter, as well as resetting the final value of the range of variables to be cleared in the WHILE statement. However, care must be exercised in assigning the maximum value of the range which could be less than or equal to the maximum available variable with the control system. As O8008 is a macro, a call has to be given to it as “G65 P8008” and there is no need of any arguments with this G65 call. Moreover, Ingenuity may be applied to alter and adapt this macro to be applicable to clear two or more ranges of 500+ series of common variables as well as to clear common variables of 100+ series.

On some Fanuc’s control models, these higher level common variables can be given a meaningful English name, up to eight characters long, which could be an easy way to spot and avoid these variables from any undesirable handling or change. The function that can be used with single variable or a range of variables is “SETVN” an acronym for “Set Variable Name”. For Example, SETVN502[CENTER, TAPER, LENGTH, INNERDIA, OUTERDIA] results in variable #502 named as CENTER, Variable #503 named as TAPER, Variable #504 named as LENGTH, Variable #505 named as INNERDIA, and Variable #506 named as OUTERDIA.

Moreover, on some control models, common variables from #500 to #627 can be protected from any change in its data by setting certain system parameters.

4. DISCUSSION

This paper is primarily to understand the structure of Variables available in Fanuc’s “Custom Macro B” environment with a particular emphasis on Common Variables for the development of Macros. This paper also discussed the use of Common Variables to eliminate a serious limitation associated with the Local Variables. This paper also discussed the features, capabilities, limitations as well as handling of both the 100+ and 500+ series of Common Variables. This paper also briefly discussed the use of Fanuc’s “Custom Macro B” for other CNC related works. A check has also been provided to ascertain the availability of “Custom Macro B” which is an optional feature of Fanuc make CNC controllers. The future research direction which is also the limitation of this paper could be the practical implementation of this conceptual framework for the development of macros for required tool motions.

REFERENCES

[1] *Hasan, M.A. (2015),* "Computer Numerical Control Machines: An Account of Programming Methods and Techniques", *Journal of Material Science and Mechanical Engineering (JMSME),* Vol. 2, No. 12, pp. 14-17.

[2] *Hasan, M.A. (2015),* "Computer Numerical Control Machines: An Introduction to Parametric Programming using Custom-Macro B", Journal of Basic and Applied Engineering Research, Vol. 2, No. 18, pp. 1566-1569.

[3] *Smid, P. (2005), "Fanuc CNC Custom Macros"*, Industrial Press, Inc., 200 Madison Avenue, New York, NY, USA.

[4] Djassemi, M. (1998), "A Parametric Programming Technique For Efficient CNC Machining Operations" *Computers and Industrial Engineering,* Vol. 35, No. 1-2, pp. 33-36.

[5] *Lynch, M., "Modern Machine Shop Magazine"*, www.mmsonline.com

[6] CNC Concepts, Inc., *www.cncci.com*

[7] www.cnccookbook.com

[8] http://www.machinetoolhelp.com/Applications/macro/macro_variables.html

[9] Adithan, M. and Pabla, B.S. (2007), *"CNC Machines"*, New Age International (P) Limited, Publishers, New Delhi.

[10] Nikiel, G. (2007), "Computer-Aided CNC Programming for the Machining of Non-Typical Parts", *Advances in Manufacturing Science and Technology*, Vol. 31, No. 4, pp. 21-36.