

## Resumo.

O uso de uma variável é necessário quando você não sabe ainda um valor ou uma função, ou seja, quando há uma situação de mudança que impede a determinação direta destes valores.

Uma variável pode ter qualquer significado: contador, valor de F, G, H, S, número, fator de controle.... o que você quer.

Sempre que você precisar escrever ou ler um valor em mudança, uma variável é útil!

As variáveis comuns têm o significado atribuído, em oposição às variáveis de sistema que são unívocas.

Com as variáveis podemos utilizar certas frases capazes de realizar desvios ou repetições (GOTO, IF, WHILE).

Podemos também usar expressões lógicas para fazer comparações.

Quando falamos de "variável" referemo-nos a uma expressão: uma ou mais variáveis e/ou constantes juntadas por expressões matemáticas ou lógicas.

E' o operador que usa e abusa das variáveis: significados, operações matemáticas e lógicas, desvios condicionais ou não.....tudo!

A variável é uma folha em branco: você pode escrever o que quiser!

Os conceitos aprendidos são fundamentais, essenciais para escrever um programa macro. Mas o que é um programa macro?

E' o que vamos descobrir hoje!

Um programa macro é um programa paramétrico, ou melhor, um sub-programa paramétrico. A diferença entre os dois (programa paramétrico e macro) é que uma macro aceita argumentos, isto é, pode receber valores.

Isto torna as macros muito flexíveis e poderosas.

Macros são programas paramétricos, então realizadas segundo os conceitos previamente vistos: variáveis aritméticas, instruções lógicas, desvios e ciclos repetitivos, ou seja, permitindo o desenvolvimento de propósito geral ou programas especializados. Então você já sabe escrever macros

O que devemos aprender é como se invoca uma macro, conhecer os parâmetros, tipo de variáveis e ainda mais.

Um programa macro pode ser invocado usando um dos seguintes métodos:

Chamada simples	G65	G65 P6054 L3 ...
Chamada modal	G66 G67	G66 P6325 L4 ...
Chamada de macro através do código G	G	G356 ...
Chamada de macro através do código M	M	M50 ...
Chamada de subprograma com código M	M	M100
Chamada de subprograma com código T	T	T44

os pontos substituem os argumentos (valores passados para a macro).

As diferenças fundamental entre uma subrotina (M98) e a chamada macro (G65) são as seguintes:

\* M98 não pode incluir argumentos, G65 sim

\* Quando um bloco M98 contém outros comandos NC (G01 X10 M98Pp) a sub-rotina é chamada após a execução dos outros comandos. G65 incondicionalmente chama a macro.

\* Com G65, o nível de variáveis locais varia, com M98 não muda.

Isto significa que cada nível tem suas próprias variáveis.

### \*\* Chamada simples G65

Quando for especificado G65, a macro especificada é chamada com o endereço P.

E' possível passar dados (argumentos) para a macro.

G65 Pp Ll <argumentos>

Pp número do programa para chamar  
Ll número de repetições (1=default)  
<argumentos> dados passados para a macro

E' possível o recurso a uma profundidade de quatro níveis, incluindo as chamadas simples e modais. Você não pode chamar M98.

#### \*\* Chamada modal G66

Depois da chamada G66, a macro especificada é invocada no final de todos os blocos que especificam um movimento eixo. Isso continua até que G67 está programado, o que elimina a chamada modal macro.

G66 Pp Ll <argumentos>

Pp número do programa para chamar  
Ll número de repetições (1=default)  
<argumentos> dados passados para a macro

Em um bloco G66 não pode ser chamada outra macro.

#### \*\* Chamada de macro através do código G

Através de parâmetros você pode especificar uma chamada de macro específica, usando códigos G não modal.

G233 X10 Y20 Z-20;

Os códigos G pode assumir valores entre 1 e 9999.

A correspondência entre o programa e o parâmetro é a seguinte:

programa	parâmetro
09010	6050
09011	6051
09012	6052
09013	6053
09014	6054
09015	6055
09016	6056
09017	6057
09018	6058
09019	6059

Se você deseja usar como macro o programa 09014, usando como número 555, você tem que colocar esse número no parâmetro 6054.  
A macro G555 ... vai invocar o programa 09014.

Em um programa macro chamado através do código G , você não pode chamar outras macros com códigos G. Em um subprograma chamado com códigos M ou T, você não pode chamar outras macros com códigos G.

A chamada com código G, é útil por várias razões:  
você não deve lembrar o número de programa,  
você pode atribuir um número a uma função, que seja fácil de lembrar:  
uma macro semelhante a função G83, por exemplo pode ser chamada  
G183, o que torna fácil de lembrar.

**\*\* Chamada de macro através do código M**

Através de parâmetros você pode especificar uma chamada de macro não modal específico com código M

M4567 C6 B4 A1

A correspondência entre o programa e o parâmetro é a seguinte:

programa	parâmetro
09020	6080
09021	6081
09022	6082
09023	6083
09024	6084
09025	6085
09026	6086
09027	6087
09028	6088
09029	6089

Se você deseja usar como macro o programa 09027, usando como número 555, você tem que colocar esse número no parâmetro 6087.  
A macro M555 ... vai invocar o programa 09027.

Em um programa macro chamado com código G ou uma sub-rotina chamada com códigos M ou T, você não pode chamar outras macros com código M. Este código é tratado como código M normal.

**\*\* Chamada de subprograma com código M**

Utilizando parâmetros você pode especificar uma sub-rotina (programa macro) com código M. O programa macro pode ser chamado com o mesmo modo de chamada de sub-rotina M98.

M37

A correspondência entre o programa e o parâmetro é a seguinte:

programa	parâmetro
09001	6071
09002	6072
09003	6073
09004	6074
09005	6075
09006	6076
09007	6077
09008	6078
09009	6079

Não é permitido especificar argumentos. Em um programa chamado com código G ou uma sub-rotina chamada com códigos M ou T, você não pode chamar outros subprogramas com código M.

**\*\* Chamada de subprograma com código T**

Usando um parâmetro (6001) você pode ativar as chamadas de sub-rotinas (programas de macro) através do código T.  
O programa macro é chamado cada vez que é especificado no programa um código T.

T55

O código T especificado pelo programa é definido na #149.  
Em um programa macro chamado com código G ou em uma sub-rotina chamada com códigos M ou T, você não pode chamar outras sub-rotinas com códigos T.  
Este código é tratado como código T normal.

Antes de prosseguir para a criação de uma macro é necessário abordar e esclarecer alguns aspectos relacionados com a macro:

- \* Argumentos e variáveis locais
- \* As variáveis de sistema

#### \*\* Argumentos

O termo argumento indica um dado (letra), ou seja, um valor que é passado para a macro. Suponha de ter a macro G200, que aceita uma variedade de argumentos, incluindo X, Y, Z. ..

G200 X100 Y100 Z-20 ...

Os argumentos são precisamente X, Y, Z a que nós passamos os valores

100, 100, -20

O número de argumentos que uma macro pode aceitar é limitada pela específica de chamada (discutido abaixo).

Na medida do possível a seleção dos argumentos devem refletir o sentido que normalmente usamos ( X, Y, Z valores para eixos, G função, M miscelânea..).

Suponha de inventar uma macro semelhante a G83, função NC.  
Ninguém proíbe de escrever a macro como

G83 A B C D ao invés de G83 Z F R Q

Claro que é mais familiar para o programador entender o segundo exemplo, porque é mais familiar.

**O significado dos argumentos é definido pelo programador (realizador macro). X, Y, Z, pode não ser relatado aos eixos, mas tomar um novo significado implementado na macro. Não confunda o significado das letras de um programa normal com as letras de um programa macro. É muito importante compreender esta diferença para não entrar em confusão!**

G83 Z-30 F100 Q2 R2, o significado de

Z é o valor da profundidade do ciclo G83  
F é a velocidade  
Q é o ciclo de descarga  
R é a descida rápida

G200 X199 Y45 Z45 F44 A22 B33 Q22

o significado dos argumentos X, Y, Z, F, Q, A, B é definido pelo realizador, Z pode não estar relacionada com a profundidade.

Portanto, é aconselhável manter o significado original das letras de um programa CNC

Nem sempre é possível manter o significado das letras de um programa macro iguais ao de um programa CNC! O truque é manter as variáveis mais importantes

com o mesmo significado e utilizar as outras para quaisquer fins secundários.

exemplo

Uma macro que usa o M83 (G83) ciclo hipotético

G230 Z-40 R0 F1000 M83 J0 Q0.4 B2 A1

Z é o valor da profundidade do ciclo M83-->G83

F é a velocidade do ciclo M83

Q é o ciclo de descarga de M83

R é a descida rápida

Outras variáveis podem ter um sentido geral, não alterando o significado original da função G83.

**\*\* Variáveis locais**

As variáveis usadas até agora, foram deliberadamente de tipo comum, ou seja, variáveis de base utilizáveis em qualquer contexto, sem significado, adaptável às nossas necessidades.

de #100 a #149 e de #500 a #531  
opcionalmente, de #150 a #199 e de #532 a #999

As variáveis locais começam a partir de #1 a #33 e só podem ser usadas dentro de um macro e estão diretamente relacionados com os argumentos: quando uma macro é chamada, os argumentos são atribuídos às variáveis locais.

Existem dois tipos de específica:

- \* Tipo de específica I
- \* Tipo de específica II

**\*\* Tipo específico I**

O tipo específico I usa as letras do alfabeto diferente de G, L, O, N, P, uma vez cada. A associação letra variável é a seguinte:

A #1	I #4	T #20
B #2	J #5	U #21
C #3	K #6	V #22
D #7	M #13	W #23
E #8	Q #17	X #24
F #9	R #18	Y #25
H #11	S #19	Z #26

Os endereços IJK devem ser especificados em ordem alfabética, os outros podem seguir a ordem desejada.

**\*\* Específicas do tipo II**

O tipo específico II usa as letras A, B e C, uma vez cada, e as letras I, J, K até 10 vezes. A associação letra variável é a seguinte:

A #1	K3 #12	J7 #23
B #2	I4 #13	K7 #24
C #3	J4 #14	I8 #25
I1 #4	K4 #15	J8 #26
J1 #5	I5 #16	K8 #27
K1 #6	J5 #17	I9 #28
I2 #7	K5 #18	J9 #29

J2	#8	I6	#19	K9	#30
K2	#9	J6	#20	I10	#31
I3	#10	K6	#21	J10	#32
J3	#11	I7	#22	K10	#33

O CNC identifica internamente a especificação adotada, e se utilizados em conjunto (o que é possível), a especificação do tipo de chamada para o último tem precedência.

Para esclarecer os conceitos vamos dar um exemplo.

G260 Z-30 R0 F1000 I20 D36 Q0.3

A macro aceita vários argumentos G260 Z, R, F, Q, I, D (específica I) que têm sido associados com certos valores:

Z=-30 R=0 F=1000 Q=0,3 I=20 D=36

Cada argumento está relacionado a uma variável, então os valores estão associados com as mesmas.

Z -> #26=-30	R -> #18=0	F -> #9=1000
Q -> #17=0.3	I -> #4=20	D -> #7=36

Então G260 Z-30 R0 F1000 I20 P36 Q0.3

Pode ser interpretado

G260 #26[-30] #18[0] #9[1000] #17[0.3] #4[20] #7[36]

***O aspecto importante para nunca esquecer é que as variáveis locais são sempre indefinidas, assumem um valor apenas quando elas estão associadas com um argumento, ou quando elas são inicializadas por alguma operação.***

**\*\* Variáveis de Sistema**

O operador deve consultar o manual de parâmetros de sua máquina. As variáveis descritas aqui foram tiradas a partir do manual dos parâmetros B-63090IT/01 Series FANUC 21

Há uma abundância de variáveis de sistema, não podemos enumerá-las todas. Vamos fazer uma lista de algumas variáveis que iremos usar em nossos exemplos.

As variáveis de sistema são usadas para ler e escrever alguns dados do CNC, como a posição atual, corretores de ferramenta, os dados modal, etc ...

Estas variáveis têm seu significado próprio e específico que não pode ser alterado. Há um número considerável, e nem todos diretamente útil para o programador: elas são essenciais para o desenvolvimento de programas de automação para uso geral.

Lista das variáveis mais comumente usadas pelo programador

* de #10001 a #13999	valores de correção da ferramenta
* #3000	alarme macro
* #3001 #3002 #3011 #3012	tempo
* #3003 #3004	controle de operação automática
* #3901 #3902	número de peças
* #4001 #4130	dados modal
* #5001 #5014	posição atual
* #5201 #5324	offset zero peça
de #2500 a #2806	

de #7001 a #7944  
de #14001 a #19984

\*\* valores de correção da ferramenta

O valor de compensação ferramenta pode ser lido e escrito. As variáveis utilizadas dependem do número de pares de corretores, a distinção entre as correções de geometria e correções de usura, a distinção entre compensação de comprimento e compensação de raio.

Se o número de pares de corretores não é superior a 200, também podemos usar as variáveis de #2001 a #2400

Variáveis de sistema para a memória tipo A de corretores ferramenta

Número corrector	Variável de sistema
1	#10001 (#2001)
2	#10002 (#2002)
.	.
.	.
200	#10200 (#2200)
201	#10201
.	.
.	.
999	#10999

Variáveis de sistema para a memória tipo B de corretores ferramenta

Número de correção	Correção geometria	Correção usura
1	#11001 (#2201)	#10001 (#2001)
2	#11002 (#2202)	#10002 (2002)
.	.	.
.	.	.
200	#11200 (#2400)	#10200 (#2200)
.	.	.
.	.	.
999	#11999	#10999

Variáveis de sistema para a memória tipo C de corretores ferramenta

Número corrector usura	comprimento de ferramenta		raio da ferramenta geometria
	Geometria	usura	
-----	-----	-----	-----
1	#11001 (#2201)	#10001 (#2001)	#13001 #12001
2	#11002 (#2202)	#10002 (#2002)	#13002 #12002
.	.	.	.
200	#11200 (#2400)	#10200 (#2200)	.
.	.	.	.
999	#11999	#10999	#13999 #12999

\*\* Alarmes macros # 3000

Quando é atribuído um valor de 0 a 200 à #3000, o CNC pára em alarme.  
Após o valor numérico, pode ser escrito uma mensagem de aviso não superior a 26 caracteres.  
A tela exibe o número de alarme, adicionando 3000 para o valor da #3000 e a mensagem de alarme.

#### Exemplo

#3000=10(interferência raio)

O alarme aparece assim na tela  
"3010 interferência raio"

#### \*\* Dados relatados ao tempo

#3001

É um temporizador que continuamente conta em incrementos de 1 milissegundo.  
Ao ligar o cnc tem valor 0, depois de 2147483648 retorna a 0.

#3002

É uma temporizador que conta em incrementos de 1 hora quando estamos no início do ciclo. Preserva o seu valor mesmo quando a máquina está desligada. Depois 9544.371767 de horas volta para 0.

#3011

É usada para ler a data atual. O dado ano/mês/dia é convertido em um número aparente decimal.  
Por exemplo, 15 de outubro de 2006 é representado como 20061015.

#3012

É usada para ler a hora atual. O dado horas/minutos/segundos é convertido em um número aparente decimal.  
Por exemplo, 20 horas e 28 minutos, 50 segundos é representado como 202850.

#### \*\* O controle automático

O controle do funcionamento automático pode ser alterado

##### - #3003

valor	bloco único	conclusão das funções auxiliares
0	habilitado	esperado
1	desativado	esperado
2	habilitado	não esperado
3	desativado	não esperado

##### - #3004

valor exato	Feed hold	Controle velocidade	Parada
0	Habilitado	habilitado	habilitado
1	desativado	ativado	ativado
2	ativado	desativado	ativado
3	deficientes	deficientes	habilitados
4	ativado	ativado	desativado
5	desabilitado	desabilitado	habilitado
6	desativado	ativado	desativado
7	deficientes	deficientes	deficientes



O valor de #3003 e #3004 é 0 quando depois de ter ligado o CNC.

\*\* Número de peças

- #3901 número de peças usinadas

- #3902 número de peças necessárias

Estas variáveis podem ser escritas e lidas, mas tome cuidado para não atribuir valores negativos!

\*\* Dados modal

você pode ler os valores dos dados modais especificados imediatamente antes do último bloco.

Variável	Função
#4001	G00, G01, G02, G03, G33
#4002	G17, G18, G19
#4003	G90, G91
#4004	
#4005	G94, G95
#4006	G20, G21
#4007	G40, G41, G42
#4008	G43, G44, G49
#4010	G98, G99
#4011	G50, G51
#4012	G65, G66, G67
#4013	G96, G97
#4014	G54-G59
#4015	G61-G64
#4016	G68, G69
.	.
#4022	grupo 22
#4102	Código B
#4107	Código D
#4109	Código F
#4111	Código H
#4113	Código M
#4114	Número de sequência
#4115	Número do programa
#4119	Código S
#4120	Código T
#4130	Código P( número do sistema de coordenadas da peça extra selecionado)

O resultado de #10=#4001, pode ser 0, 1, 2, 3 ou 33.

O resultado de #10=#4014 pode ser 54, 55, 56, 57, 58, 59

## \*\* Posição atual

Os dados da posição não podem ser escritos, mas apenas lidos

Variável -----	tipo -----	sistema de coordenadas -----	Correções ferramenta -----	leituradurante o movimento -----
#5001-#5004	ponto final bloco	peça	não incluídas	abilitada
#5021-#5024	posição atual	máquina	incluídas	desativada
#5041-#5041	posição atual	peça	incluídas	desativada
#5061-#5064	posição sinal de salto	peça	incluídas	desativada
#5081-#5084	entità compensação comprimento ferramenta			desativada
#5101-5104	erro de posição do servo			desativada

O primeiro dígito representa o número de eixo (5001>>X, 5002>>Y...).

As variáveis de #5061 a #5064 contêm as coordenadas da posição do salto em um bloco G31. Se não ha nenhum sinal de salto, elas contem as coordenadas do ponto final do bloco.

As variáveis de #5081 a #5084 contém o valor de correções ferramentas usadas atualmente.

## \*\* Zero offset peça

O zero offset peça pode ser lido e escrito (o offset representa o sistema de origem).

A lista dessas variáveis é muito longo para ser mostrado neste manual, vou dar uma dica, uma breve introdução: complete a leitura do seu manual máquina!

#5201 #5324	offset zero peça origem sistemas G54-G59
#2500 #2806	offset zero peça origem sistemas G54-G59
#7001 #7944	offset zero peça origem sistemas G54.1 P1, G54.1 P48
#14001 #19984	offset zero peça origem sistemas G54.1 P1, G54.1 P300

As variáveis de sistema permitem que você execute programas inteligentes, capazes de se adaptar a situações diferentes (uma ferramenta quebrada ou a mudança no ciclo de trabalho ou a criação de origens....).

Com as variáveis você tem pleno controle de CNC!