

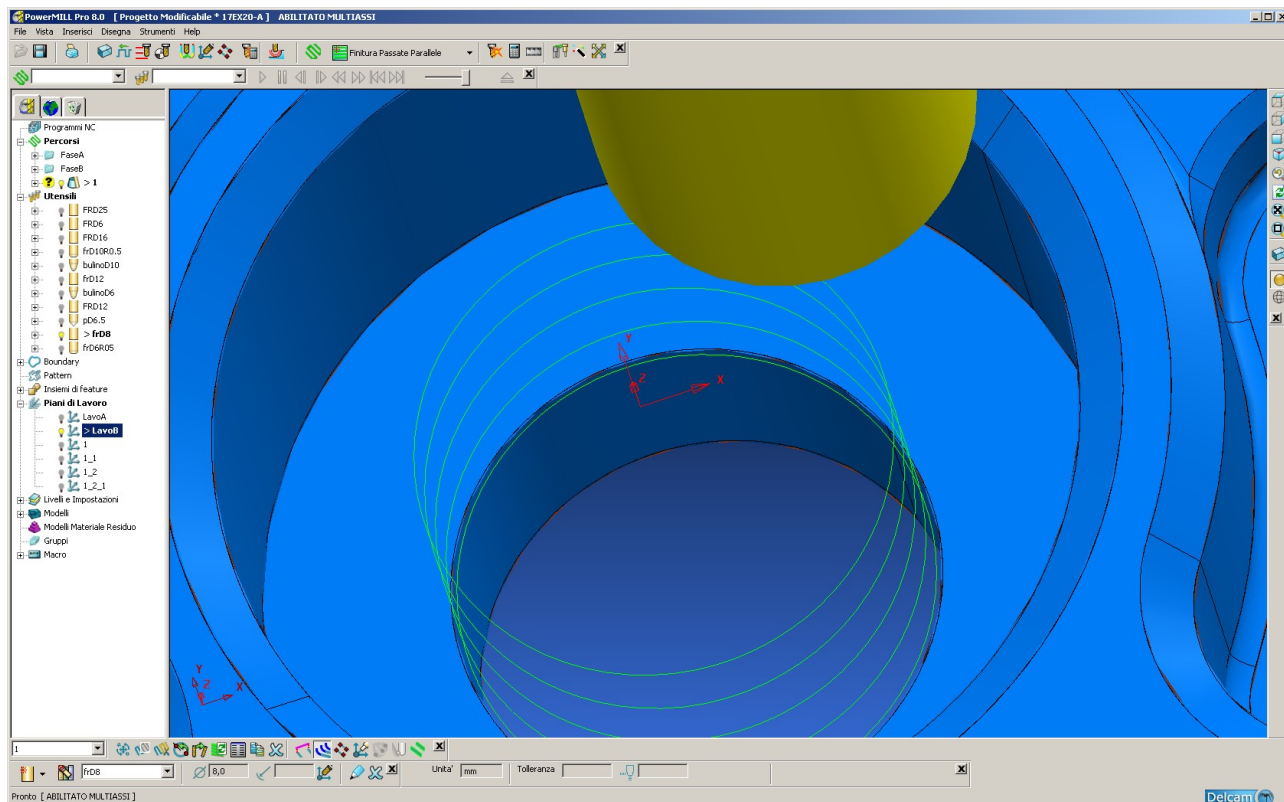
Olá,
hoje vamos ter alguns exemplos de programas paramétricos.
Não iremos parar muito tempo porque o argumento macro irá explicar em detalhe o que é um programa paramétrico e as normas que o rege.
Agora queremos mostrar o uso geral de variáveis em um programa G-código ISO. Os programas paramétricos são comuns programas de G-code que usam as variáveis. Tais programas não necessitam de controles especiais, porque as variáveis são definidas pelo operador. Eles são rápidos de fazer e se prestam a resolver problemas individuais. Podemos usar variáveis comuns ou variáveis de sistema (lembra-se que as variáveis locais são usadas nas macros). As variáveis de sistema serão discutidas juntamente com as macros, não se preocupem em saber agora.
Os exemplos propostos mostram como você pode usar as variáveis de sistema em um programa que não é uma macro. Se tivermos que escrever um programa solúvel com o uso de variáveis de sistema, ninguém proíbe; são variáveis como as outras, mas cuidado com o significado delas: você não pode mudá-lo.
A #5003 é o ponto final do bloco no sistema de coordenada usado atualmente, e terá sempre este significado. Dar um significado diferente, como é feito com as variáveis comuns, não é possível.
Use todas as variáveis que pretende, mas respeitando seu significado!

Iremos apresentar 4 exemplos querendo mostrar a adaptabilidade das variáveis em diferentes situações.

- looping
- looping controlado
- redefinição de ferramenta (H)
- aquecimento spindle

Looping

Nos vimos como repetir uma série de blocos em um programa cnc.
Queremos reiterar o argumento, mostrando a sua utilização.
O processo que vamos conseguir é a interpolação helicoidal.



```

Ø1000 (interpolação helicoidal)
G90G0
T1M6
G08P1
G90G0G54X0Y0A0C0S9000M3
G43H1Z100M8
G10L12P52R5
D52
Z5
G01F2000Z0
#100=20
G91G41X20
WHILE[#100NE0]DO1
G03I-20Z-1
#100=#100-1
END1
G03I-20
G01G40X-20
G90G0Z200M9
M5
G08P0
G91G28G0Z0
M30

```

A ferramenta (fresa de raio 5) é posicionada em X0 Y0 Z0

```

T1M6
G08P1
G90G0G54X0Y0A0C0S9000M3
G43H1Z100M8
G10L12P52R5
D52
Z5
G01F2000Z0

```

A #100 vai ser usada como contator, definida com o valor de 20 representando o número de repetições a serem executadas

```
#100=20
```

segue um ciclo repetitivo realizado com WHILE:
G03I-20Z-1 é uma interpolação helicoidal.

```

G91G41X20
WHILE[#100NE0]DO1
G03I-20Z-1
#100=#100-1
END1

```

o valor di #100 è controlado a cada repetição:
se #100 não chegou a 0 (WHILE[#100NE0]), o ciclo vai continuar, caso contrário irá executar uma interpolação circular e o programa acabará.

```

G03I-20
G01G40X-20
G90G0Z200M9
M5
G08P0
G91G28G0Z0
M30

```

O exemplo proposto mostra como você pode usar o loop WHILE para repetir uns blocos. Você pode usá-lo para repetir qualquer contorno ou blocos de comandos cnc.

```
-definir o contator          #100=n

-escrever o loop              WHILE[condição]Don
                               ENDn

-introduzir o contorno        WHILE[condição]Don
a ser usinado                 perfil
                               ENDn
```

Se temos que trabalhar mais perfis, você pode repetir o procedimento escrito. Não é a melhor solução, porque temos que escrever repetidamente os mesmos blocos. As macros muito bem resolvem o problema.

Looping controlado

Problema:

criar uma rotina para o controle das repetições, capaz de verificar o valor de Z. Dado um perfil, defina a profundidade de trabalho e o número de repetições, a rotina irá verificar que as repetições não ultrapassem a profundidade para qualquer valor de remoção.

E' sabido que durante um processo repetitivo, o valor a ser alcançado e o da remoção, estão relacionados: se for necessário remover 50 milímetros de um perfil fazendo uma remoção de 2mm, temos que realizar 25 repetições. Se em vez de 2 mm eu quero remover 1,75 milímetros, dependendo do tipo de usinagem, estamos obrigado a cálculos tediosos para atingir o tamanho desejado. Não seria útil um programa de controle que permite que você faça qualquer remoção capaz de ajustar automaticamente o valor final? Porque não posso fazer uma remoção de 1,456 milímetros ao invés de 2mm sem ter que verificar de superar o valor final? Seria economizar muito tempo incidindo apenas sobre o programa. Vamos tentar criar uma sub rotina de supervisão do valor alcançado do eixo Z e possivelmente que seja auto-limitado; a sub-rotina será invocada durante a execução do perfil para determinar a conclusão, a continuação ou modificação.

Neste exemplo useremos a #5003 cujo significado é o ponto final do bloco no sistema de coordenada usado atualmente para o eixo Z.

```
O8000 (Control Z)
(#120 profundidade de trabalho)
(#121 profundidade de corte)
(#122 cálculo do valor seguinte)
(#123 material a ser removido)
#122=#5003-#121
#123=ABS[#120-#5003]
IF[#122LT#120]THEN#121=#123
M99
```

As primeiras quatro linhas expressam o significado das variáveis usadas no programa.

#120	profundidade de trabalho,	(valor final Z para ser usinado)
#121	profundidade de corte,	(remoção particular)
#122	cálculo do valor seguinte,	(próximo valor de usinagem)
#123	material a ser removido,	(o que falta para chegar ao valor final)

As linhas restantes definem a rotina em si

```
#122=#5003-#121
```

A #122 calcula o próximo valor de Z, subtraindo o valor de corte (#121) ao valor atual de posição do eixo Z (#5003)

```
#123=ABS[#120-#5003]
```

A #123 calcula o valor absoluto da parte em falta, ou seja, a diferença entre o valor da posição atual e a profundidade de trabalho final.

Usamos o valor absoluto (ABS), pois estamos interessados a distância restante representado pela diferença dos valores.

Devemos sempre prestar atenção aos sinais das ações em questão, você vai achar que as duas coisas nem sempre coincidem!

```
IF[#122LT#120]THEN#121=#123
```

Se #122 é inferior a #120, então #121=123.

Se o próximo valor calculado (posição do eixo) é maior que a profundidade definida pelo programa (#120), então o próximo passo não será mais igual a #121 (o valor definido pelo programa), mas #123, que representa a distância restante.

Temos uma sub rotina muito pequena, capaz de controlar a profundidade de usinagem. É interessante que não há nenhuma menção do número de repetições!

Claro que isso pode ser a base para melhorias.... tente experimentar!

Como usar o programa O8000?

- definir #120 e #121,
- escrever um ciclo de repetição com WHILE
- chamar o programa O8000
- escrever o perfil dependente de #121

exemplo

Temos um perfil qualquer de 34,7 milímetros de profundidade, queremos fazer uma remoção de 1,5 mm (atingindo o valor de 34,7).

```
-----
-----
#120=-34.7 (profundidade final)      (definir #120 e #121)
#121=1.5 (profundidade de corte)
WHILE[#5003GT#120]DO1                (escrever um ciclo de repetição com WHILE)
M98 P8000                            (chamar o programa O8000)
G91 G01 F500 Z#-121                  (escrever o perfil dependente de #121)
(PERFIL)
----
END1
-----
-----
```

O programa O8000 é chamado para cada repetição do ciclo, se a distância restante é maior da #121, o programa não vai intervir e deixa que o ciclo continua. Se a distância restante é menor de #121 o programa reduz automaticamente a passagem final. A condição [#5003GT#120], é usado para sair do loop.

E' necessário que as variáveis associadas com a profundidade final e a de corte sejam respectivamente #120 e #121 (utilizados no programa 08000), pois a sub rotina não vai funcionar. Observe a presença de #121 no perfil. Analisar cuidadosamente o programa, analisar cada linha, procure soluções diferentes e.... muitas perguntas.

Você pode fazer o seguinte exercício: escrever um programa que define o número desejado de repetições que pode controlar o valor final de Z.

Redefinição da ferramenta (H)

A possibilidade de ter um programa paramétrico para o cálculo do comprimento de uma ferramentas pode ser útil.

Uma pessoa inexperiente, o procedimento de medição automática que não funciona, ou ter um único programa em todas as máquinas, são exemplos hipotéticos de uso.

Então vamos escrever um programa semi-automático que permite o cálculo do comprimento de uma ferramenta.

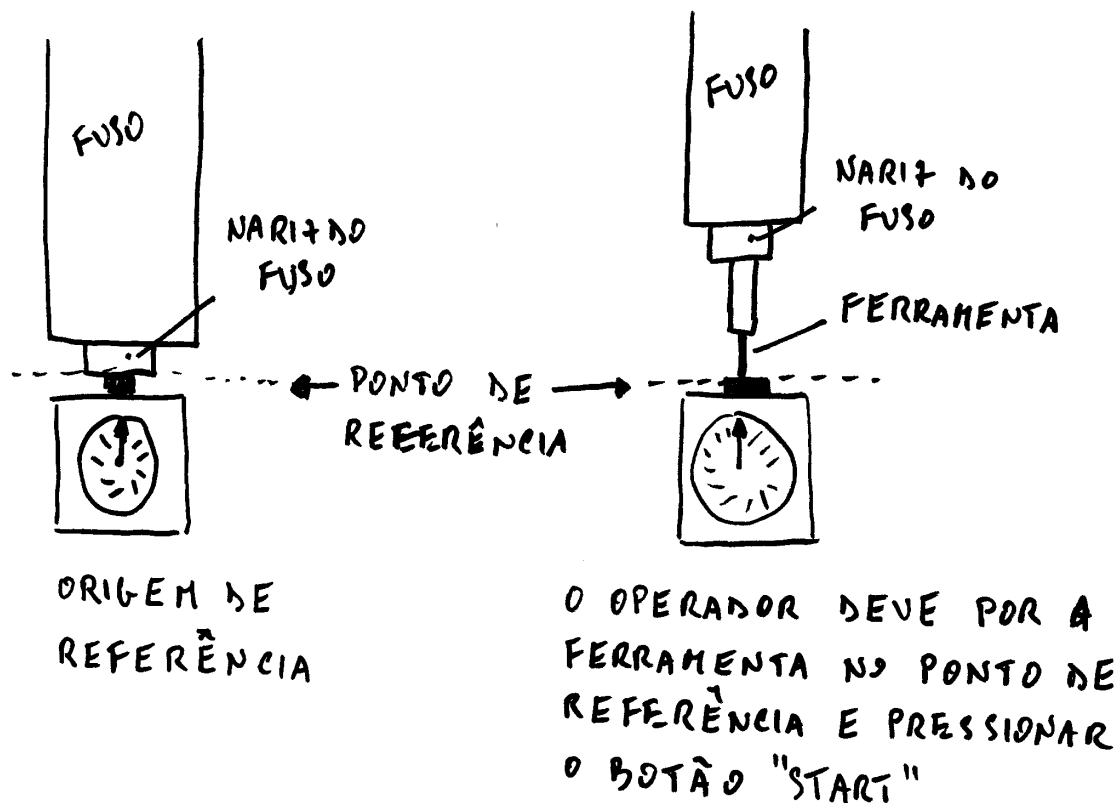
Vamos tomar como referência uma origem conhecida (por exemplo, G59).

O comprimento de uma ferramenta é calculado como a distância do nariz do fuso em relação a um ponto Z (G59) calculado anteriormente.

Comprimento 0, o nariz do fuso está posicionado no ponto de referência.

Comprimento 100, o nariz do fuso está posicionado 100 mm do ponto de referência.

O operador deve posicionar a ferramenta sobre a origem de referência, a máquina irá calcular e escrever na tabela de corretores H o valor encontrado.



```
O8700 (H-ferramenta)
T56M6
G90G59A0C0
X0Y0
M0 (aguardando ferramenta de colocação)
#[2200+#4120]=#5023-#2706
G30G91Z0
M99
```

-A ferramenta é colocada sobre a origem de referência (G59)

```
T56 M6
G90 G59 A0 C0
X0 Y0
```

-O CNC pára à espera do operador

```
M0
```

o operador deve colocar a ferramenta do fuso no ponto de referência e pressionar o botão START

-Cálculo de comprimento (o operador tem pressionado start)

```
#[2200+#4120]=5023#-#2706
```

#4120 é o número da ferramenta chamada (T1=1, T5=5, T20=20).

#2200 é o valor de correção da comprimento,

se a ferramenta é o T1, vai ser #[2200+#4120]=#2201 (H ferramenta 1)

se a ferramenta é o T2, vai ser #[2200+#4120]=#2202 (H ferramenta 2)

...

A #5023 é a posição atual do eixo Z na coordenada da máquina

A #2706 é o valor Z da origem de referência expressa em coordenada da máquina.

Podemos dizer:

$H = \text{valor Z atual} - \text{valor de origem de referência}$

-saída: após a introdução do valor o programa termina.

```
G30 G91 Z0
M99
```

Reflexões

O programa executa muito bem, mas é muito limitado.

O uso de G59 exclui a possibilidade de utilizar esta origem para qualquer outra finalidade. Você também pode ligar para uma única ferramenta de cada vez.

É melhor adotar outras medidas, a fim de encontrar um procedimento independente da origem de sistema da peça.

Claro, nada proíbe o uso de um sistema de referência que nunca usamos (G54.1 P250), mas não resolve o princípio subjacente: é preciso ser independente de qualquer sistema.

Veremos mais tarde como um programa macro é muito mais flexível e fácil de usar.



Aquecimento spindle

A Tecnologia High Speed Machining (H.S.M.) é realizada através da rotação do fuso muito alta bem como os avanços.
Para evitar pôr em perigo a vida do fuso é preferível o aquecimento antes da usinagem.

E possível escrever um programa paramétrico para esta finalidade?
Vamos tentar!

```

O9870 (AQUECIMENTO FUSO)
(VALORES DE DESLOCAMENTO EIXOS)
#101=-1000 (X)
#102=550 (Y)
#103=100 (Z)
#105=0 (COMPARAÇÃO DE TEMPO)
#106=20 (TEMPO DE AQUECIMENTO)
#107=2000 (VELOCIDADE INICIAL FUSO)
#3002=0 (CONTADOR TEMPO)
#3004=2 (CONTROLE DE VELOCIDADE DESATIVADO)
(ZERO CNC)
G91G28G0Z0
G91G28G0A0C0
G91G28G0X0Y0
  
```

```

S#107M3
(CICLO DE AQUECIMENTO)
WHILE[#105LT#106]DO1
(MOVIMENTO EIXOS)
G91 G01 F1500 X-[#101/2] Y-#102 Z-#103 A-90 C90
X-[#101/2] Y#102 Z#103 A90
X[#101/2] Y-#102 Z-#103 A-30 C-180
X[#101/2] Y#102 Z#103 A30 C90
(TEMPO DE PROCESSAMENTO)
#105=#3002*60
IF[#107GE9000]GOTO400
IF[#105GE12]GOTO300
IF[#105GE9]GOTO200
IF[#105GE6]GOTO100
IF[#105GE3]THEN#107=4500
GOTO400
N100#107=6500
GOTO400
N200#107=8000
GOTO400
N300#107=9000
N400END1
#3004=0 (CONTROLE DE VELOCIDADE ACTIVADO)
M30

```

```

#101=-1000 (X)
#102=550 (Y)
#103=100 (Z)

```

Estas variáveis estabelecem o valor do movimento dos eixos. Os valores cobrem a gama de ação da CNC, você pode mudá-lo. O aquecimento do fuso é levado junto com o movimento dos eixos (o valor de Z não deve ser muito elevado para evitar interferência com os objetos na área de trabalho).

```

#107=2000 (VELOCIDADE INICIAL FUSO)
#3004=2 (CONTROLE DE VELOCIDADE DESACTIVADO)

```

A #107 é a configuração da velocidade inicial do fuso. A #3004 é uma variável de sistema, quando tem o valor 2 o controle de velocidade manual está desligado (o operador não pode variar a velocidade dos eixos). Esta variável é útil para controlar com precisão o tempo, e em seguida, definir a variação da velocidade de rotação à vontade e corretamente.

```

#105=0 (COMPARAÇÃO DE TEMPO)
#106=20 (TEMPO DE AQUECIMENTO)
#3002=0 (CONTADOR TEMPO)

```

A #3002 é um contador de horas, é repostado no início do programa. A #106 é o tempo de aquecimento que acreditamos ser necessário (maior a velocidade do fuso, mais deve ser o tempo de aquecimento). A #105 vai ser usada para fazer uma comparação de tempo.

```

G91G28G0Z0
G91G28G0A0C0
G91G28G0X0Y0

```

Assumimos como ponto de partida do programa, o ponto zero da máquina.

```

S#107 M3 fuso ativado

```


O loop WHILE é dividido em duas partes:

1° movimento dos eixos

```
G91 G01 F1500 X-[#101/2] Y-#102 Z-#103 A-90 C90
X-[#101/2] Y#102 Z#103 A90
X[#101/2] Y-#102 Z-#103 A-30 C-180
X[#101/2] Y#102 Z#103 A30 C90
```

pode executar qualquer movimento você quer: atenção as interferências!

2° controle tempo e velocidade de rotação fuso

#105=#3002*60	contagem do tempo
IF[#107GE9000]GOTO400	estamos em 9000 rpm? Sim, não aumentar mais
IF[#105GE12]GOTO300	passamos os 12 minutos? Ir para 300 (S9000)
IF[#105GE9]GOTO200	passamos os 9 minutos? Ir para 200 (S8000)
IF[#105GE6]GOTO100	passamos os 6 minutos? Ir para 100 (S6500)
IF[#105GE3]THEN#107=4500	passamos os 3 minutos? s = 4500
GOTO400	
N100#107=6500	
GOTO400	
N200#107=8000	
GOTO400	
N300#107=9000	

O loop WHILE compara a #105 com #106 para determinar se for preciso continuar ou acabar. A #105 (tempo desde o início) está atualizada em cada ciclo e comparada com #106 (tempo de aquecimento definido pelo operador).

```
WHILE[#105LT#106]
```

Este programa paramétrico expressa o conceito básico do aquecimento do fuso, é um bom começo para o desenvolvimento futuro.

reflexões!

Os exemplos mostram como os programas paramétricos cobrem uma amplitude de ações. Podemos criar qualquer tipo de programa que pode nos ajudar! A maior limitação é que eles não são adequados para o repetitivo. Isso não significa que você não pode usá-los para repetir blocos, mas que há outras maneiras mais fáceis para realizá-las: as macros!

Atenção!!

Estes exemplos foram escritos sem considerar os controles apropriados, é preciso ter cuidado se você deseja usar. Estou aqui para qualquer dúvida ou explicação. Até.