

Cada função tem sua finalidade, precisamente a sua funcionalidade: não pode ser diferente de como foi escrita, deve ser utilizada de acordo com o propósito para o qual foi criada. Às vezes, como no caso de Fausto, somos confrontados com situações específicas e as funções cnc que conhecemos parecem poucos úteis. No exemplo, podemos dizer que o G83 não dá resposta à necessidade que se procurou, quer dizer retratar a ferramenta fora da peça com um valor de R que se encontra por dentro. Eu queria oferecer alguns pensamentos sobre este problema para ver se é possível agir de uma maneira diferente e obter algo que nos satisfaz.

Primeiro vamos refletir sobre alguns conceitos

```
G0G0Z50
G98G83Z-10R2Q2F100
```

O comportamento deste programa, que todos nós sabemos: o valor Z50 indica o início da função de perfuração (e final da função), a ferramenta cai rapidamente para o valor de Z2 (R), haverá uma descida e subida repetidas (entre Z2 e Z-10) com base no valor de Q até que o valor final seja alcançado. Bem, e daí? Imagine definir um valor de Q=1000 (valor altamente improvável), o que acontece? O comportamento da função não muda, mas o que é interessante a considerar é o fato que o comando sempre mantém sob controle a função isto é de não ultrapassar o valor definido de profundidade.

```
G0G0Z50
G98G83Z-10R2Q1000F100
```

Neste exemplo, o valor de Q levaria a uma descarga de valor igual a 1000, que ultrapassa os limites da profundidade. Bem, a CNC vai parar a execução da função antes que isso aconteça. Na prática, o G83 torna-se G81, testemunhado uma descida da ferramenta continua até uma profundidade de Z-10. Deixamos por um momento o assunto para abrir outro!

```
G90G0Z50
G98G83Z-10R2Q2F100
X100Y100
X200Y200
X-23Y-34
-----
```

Este exemplo diz-nos que uma função de perfuração é modal e, como tal, uma sequência de pontos pode ser escrita após a função para executá-la novamente sem repeti-la. Na prática, quando você quiser fazer uma série de buracos não deve reescrever a função, mas apenas o ponto de perfuração com cuidado para não expressar o movimento usando uma função que exclui a função (G80, G00, G01, G02, G03). Agora vamos nos perguntar:

o que acontece, se em vez de X e Y nós escrevemos Z e R?

```
G0G0Z50
G98G83Z-10R2Q2F100
Z-20R-18
Z-30R-28
```

A função é repetida no mesmo ponto assumindo novos valores de Z e R para cada linha.

Bem, estas simples reflexões nos dá a ideia de que talvez você possa "dar a volta" ao nosso problema!

Vejam bem o caminho que as reflexões nos mostram:

o que é G01? Um movimento linear controlado

o que é G81? Um movimento linear controlado e repetido com retroação em rápido

o que é G83? Uma repetição di G81, quer dizer uma repetição di G01 com retração rápida e abordagem também e movimento controlado.

Introduzindo a "programação paramétrica" eu falei bem assim: "G00, G01, G02, G03, com estas funções você pode escrever qualquer programa para máquinas CNC. Querendo fazer um buraco, este seria escrito como uma sucessão de movimentos em G01. Exemplo

```
G01 Z-2 F100
G0 Z2
Z-1.9
G01 Z-4
G0 Z2
Z-3.9
G01 Z-6
...
```

Seria chato e um grande desperdício de tempo executar os programas desta forma. Engenheiros de software têm tomado medidas para introduzir frases para facilitar o trabalho do programador. O exemplo acima pode ser redefinido com a frase G83. G83 Z R F Q. G83 é chamado ciclo de perfuração ou função de perfuração". O que é bom saber é que também nós podemos realizar uma função adaptando-a às nossas necessidades. A primeira abordagem, portanto, pode ser endereçada desde modo

```
G90G0Z50
G98G81Z-35R-29F100
Z-40R-34
Z-45R-39
---
```

Como funciona o programa? A função G81 é chamada partindo do valor de Z50, desce rapidamente para o valor de Z-29, decida em G01 até Z-35, retorno em rápido para o valor R e logo em seguida Z (porque é onde termina a função). A função permanece ativa (não há a presença de G80, G0 ...) repetindo-se para os novos valores de Z e R escritos.

Temos o comportamento que o Fausto procurava!

A função G83 foi substituta por uma repetição de G81 mudando Z e R. Olhando para a usinagem, sem ler o programa, poderíamos pensar em estar na presença de um G83, mas na realidade estamos executando uma repetição cíclica (no mesmo ponto) do G81.

A abordagem encontrada (com todas suas limitações), abre o caminho para novas ideias e mostra a possibilidade de utilizar o processamento usando o G81.

É claro que se o buraco é profundo, a escrita se torna chata, a possibilidade de cometer erros provavelmente aumenta e os dedos começam a doer.

O assunto é sempre o mesmo: considerar com cuidado o contexto! Se nós raramente realizamos esse tratamento podemos levar alguns minutos

a elaboração de um programa desse tipo. Se, pelo contrário, a frequência é elevada deve-se encontrar soluções alternativas, capaz, ao mesmo tempo a execução do processamento mas contendo o comprimento do programa. Quando se trata de repetições é bom usar a paramétrica e o ciclo que o permite é o WHILE (mas também o GOTO). Continuamos o exemplo anterior, utilizando as variáveis em vez de valores constantes.

```
#100=-30 (profundidade inicial)
#101=-100 (profundidade final)
#103=5 (remoção)
#102=#100+1 (abordagem rápida)
G90G0Z50
G98G81Z#100R#102F100K0
D01
#100=#100-#103
#102=#100+1
IF[#100EQ#101]GOTO10
Z#100R#102
END1
N10G90G0Z200
```

Este programa que agora iremos explicar, sintaticamente correto não é o parâmetro final da escrita, tem várias falhas e não se presta a ser utilizado. No entanto, tem a vantagem de ser curto e fácil de interpretar, o que torna-o um excelente exemplo de estudo.

As linhas seguintes definem os valores de profundidade de perfuração de início e de fim, são os nossos limites.

```
#100=-30 (profundidade inicial)
#101=-100 (profundidade final)
```

A linha #103=5 define a remoção desejada

#102=#100+#103+1 (abordagem rápida), o movimento de aproximação rápido é calculado com base no valor da descarga (5), ao qual é adicionado um milímetro para a segurança.

G90G0Z50, posicionamento para o valor desejado de Z

G98G81Z#100R#102F100K0, ativamos a função da perfuração, mas não a executamos! Ao escrever K0, em uma função de perfuração, esta é somente lida; o CNC mantém a função ativa, isto é, lê os valores de Z, R, Q, F, P, mas não a executa. Esta abordagem serve apenas para ativar a função por cima dos pontos necessários que são o últimos da perfuração anterior. Agora escrevemos um loop de acordo com o que vimos anteriormente, calculando novos valores de Z e R em cada ciclo.

```
DO1 (Do sem WHILE = ciclo infinito)
#100=#100-#103 (novo valor de Z)
#102=#100+1 (novo valor de R)
IF[#100EQ#101]GOTO10 (controle do valor final da Z)
Z#100R#102 (perfuração)
END1
```

Eu não quero, neste momento, escrever um programa paramétrico preciso e funcional, quero mostrar o conceito de como você pode lidar com o problema proposto. O mesmo conceito, talvez mais familiar para alguns, pode ser expresso por meio de um subprograma.

```
#100=-30 (profundidade inicial)
#101=-100 (profundidade final)
#103=5 (remoção)
#102=#100+1
G90G0Z50
G98G81Z#100R#102F100K0
M98P1000
N10G90G0Z200
```

```
O1000
DO1
#100=#100-#103
#102=#100+1
IF[#100EQ#101]GOTO2
Z#100R#102
END1
N2M99
```

Supondo de usar o programa O1000 para executar este tipo de perfuração, o programa se reduz a escrever umas variáveis ...

Vamos recapitular o conceito: 1° ativação da função G81 (somente ativada), 2° criação de um ciclo de repetição da função  
O resultado é um movimento comparável ao G83 mas cuja retração é sempre fora da peça. Já é um bom ponto, vou parar por aqui.  
Os exemplos mostrados até agora (deliberadamente acadêmico) mostram o caminho

para a criação de uma função personalizada: uma macro.  
Tente executar o programa abaixo em seu cnc, é um exemplo simples de como a função G81 pode ser usada para implementar a forma 'G83' com retração fora da peça. Tentei usar as noções apreendidas até aqui, ciclo WHILE e contator

```
O1000(G83, furação profunda até -120mm)
G90G0
T1M6(BROCA D10)
G90G0G54X0Y0S1500M3
G43H1Z50M8
G83Z-60R2F100
G90G0Z200M9
M5
T2M6(BROCA D10)
#100=-60 (profundidade inicial)
#101=5 (remoção)
#102=#100+1 (abordagem rápida)
G90G0G54X0Y0S1500M3
G43H2Z50M8
G98G81Z#100R#102F100K0
#120=12 (12*5=60 para alcançar o valor di -120, #120 representa um contator)
WHILE[#120NE0]DO1
#100=#100-#101 (valor de furação)
#102=#100+1 (valor de abordagem)
z#100R#102 (furação)
#120=#120-1
END1
G90G0Z200M9
M5
G90G0G53Z0
M30
```

Observe que aos olhos está executando-se uma função de perfuração (G83) com recuo fora da peça (Z0 por cima da peça). Este poderia ser o início de uma macro, ou seja, uma função personalizada. Se assumirmos ter escrita (por exemplo, G183), com base no programa que você acabou de ver, temos de fato criado uma função cnc e como tal utilizável. Nossa perfuração seria realmente muito simples .. aqui está um exemplo

```
G90G0
T1M6 (BROCA D10)
G90G0G54X0Y0S1500M3
G43H1Z50M8
G98G183Z-100R-29Q5F100
G90G0Z200M9
M5
```

Uma ultima pergunta! Por que usar o G81 em vez de escrever uma série de movimentos em Z?  
Simplesmente porque a função de perfuração (G81) tem uma retração automática interna, exploramos este aspecto para evitar de escrever manualmente cada movimento. Isso permite que você escreva um loop com a consciência de que a cada iteração a ferramenta vai subir! Minha intenção, como eu disse, não era apresentar uma macro, mas refletir a nível geral o nosso conhecimento e como podemos moldá-los (com ajuda das variáveis). Estes exemplos, embora executáveis, exibem aspectos de discussão. Certamente servem para nos fazer entender certos conceitos, mas escrevê-los cada vez que realizar um buraco profundo não é a solução mais barata e certa! Talvez é melhor encontrar soluções mais simples: a macro!

Será que tem alguém no Brasil capaz de escrever a macro G183?  
Vou esperando!  
Até