

As questões levantadas por Fausto exigem uma reflexão de precaução em relação a alguns conceitos que formam a base para o estudo que iremos enfrentar hoje!

** O eixo C deve ser ativado e inicializado para não ter um "comportamento indesejável". O eixo C é essencialmente o nosso fuso que, normalmente está em constante movimento, mas que, em condições particulares, pode ser controlado para ser usado como eixo rotativo. Escrevendo G96 S2000 pretendemos expressar uma velocidade de corte, em seguida, uma certa rotação do fuso que vai depender do diâmetro para ser maquinado. Quando se escreve (é um exemplo) M10, queremos utilizar o fuso sob forma de eixo controlado, então não mais uma certa rotação, mas um movimento controlado que depende do processamento. A ativação e desativação servem para distinguir entre as duas possibilidades: é, portanto, exigido um código de ativação que determina a condição de trabalho!

** Nossas CNC não são todas iguais! Geralmente fala-se sobre a série, acreditando que a única diferença está no número 00, 16, 21, 18, 32 ... mas não é assim! Há muitas diferenças, e entre essas eu quero listar apenas algumas! Nossas CNC diferem no sistema de código utilizado (A, B, C), o que determina algumas pequenas diferenças entre os códigos G e também de programação. Quando falamos de funções G, devemos sempre fazê-lo em relação ao sistema de códigos. *"Eu sei que G98/99 referem-se à possibilidade de retorno de uma função de perfuração" ... "Mas não! Não é G98/G99 mas G94/G95".* Parece estranho, mas essa discussão sobre quem está certo ou errado, não vai ter fim, se você não estabelece o tipo de sistema de codificação em uso! Ambos estão certos ambos estão errados! *"Mas o G98 é usado nas fresadoras, o G94 nos tornos... G92 é a velocidade máxima do fuso.. não, é G50 "*

Aqui estão outros exemplos de debates intermináveis que provavelmente vem do fato de que vocês está usando sistemas de códigos diferentes. O tipo de sistema de codificação também define aspectos concretos, tais como, as tabelas de ferramentas. Também encontramos diferenças devidas à ativação de certas opções que determinam a capacidade de executar determinadas ações ou não!

** Um aspecto interessante são as funções da máquina: G e M(ou B). Enquanto as primeiras (falamos de padrão e não-macro) são implementadas no controle, que as gerencia, as segundas (definidas auxiliares), além de poucas, são implementadas pelo fabricante e, em seguida, o CNC não faz diretamente o controle! As funções M controladas pelo CNC são poucas: M0, M1, M2, M30, M98, M99. As outras são implementadas pelo fabricante, que deve fornecer a lógica e operação. Isto significa que, em teoria, qualquer código pode corresponder a diferentes usos. É claro que há uma ISO que procurou unificar os diversos códigos (não só G, mas também M), mas esta legislação deixa muita liberdade que se traduz concretamente na possibilidade de ter códigos diferentes para mesmas ações. Então, isso é algo a considerar com muito cuidado!

Por que estas reflexões? Nos próximos exemplos, iremos utilizar uns códigos que não podem ser considerados universais (seja G como M); de fato implementados pelo fabricante ou códigos pertencentes a um sistema de código que não é seu. Não pare nesses pontos, substitua o código que você usa, é melhor tentar entender a lógica em vez da escrita.

O exemplo que eu vou fazer é tomado a partir de um arquivo enviado a mim por Fausto relativo á figura postada no fórum. Então vamos ver os códigos utilizados por ele e que usaremos neste tratado (tipo de código A)

G74 Ciclo de furação frontal (pica-pau rápido)
G83 Ciclo de furação frontal (pica-pau normal)
G84 Ciclo de rosqueamento
G86 Ciclo de furação frontal (mandrilagem)
G98 Avanço por minuto
G99 Avanço por rotação

e também

M10 ativação do eixo C
M11 desativação do eixo C
M13 ativação motorizado rotativo M3
M14 ativação motorizado rotativo M4
M15 desativação motorizado

Tudo bem, começamos!

Uma porção do programa escrito por Fausto (um programa genérico de 16 furos)

T0101	(chamada ferramenta)
M10	(ativação do eixo C)
G28 H0	(origem cnc eixo C)
G0 C0	(eixo C absoluto 0)
G97 M13 S1200	(velocidade fixa, ativação motorizado, sentido de rotação e número de rotações)
G0 X100 Z20 M8	(posicionamento)
G98 G83 Z-20 F100 R2 Q4	(avanço por minuto, dependendo da perfuração G)
H22.5 K15	(repetição incremental)
G0 G99 G96 X400 Z400 M9	(afastamento peça, velocidade de corte constante)
M15	(desativação motorizado)
M11	(desativação eixo C)

O problema de Fausto: repetições excessivas, automação.

Para automatizar o processo, isto é, torná-lo mais fácil para o operador temos que analisá-lo, conhecê-lo!

* Análise do problema *

Considere a operação de centragem. Parece uma operação simples, temos uma broca para centrar que funciona com o motorizado, sabendo o diâmetro do processamento, a operação é reduzida a uma rotação do eixo C.

Todas as outras operações são semelhantes: rotação do eixo C e função NC (G74, G83, G84, G86 são as funções analisadas).

A porção de programa escrito acima terá que ser repetido várias vezes: uma vez para centragem, duas vezes para perfuração (D10.5 e D10.2), outra para roscagem, e enfim duas vezes para o alargamento. O torneiro programador vai concordar que é escrever essencialmente o mesmo programa seis vezes. Fica chato e repetitivo e com alta probabilidade de erro (mesmo se você usou uma sub-rotina).

E' aqui que devemos nos perguntar: existe a possibilidade de criar um procedimento que simplifica a minha vida, que automatiza todo o discurso, para me libertar das ativações e desativações (que, afinal, deveriam ser implícitas), que calcule automaticamente

E' nessas ocasiões, que o operador deve-se perguntar:
Justifica-se criar uma macro ou não!? Mas como proceder?

O método mais simples é o de analisar todo o programa e ver quais funções são fixas e quais não. Como o programa apresentado fica uma série de repetições, vamos analisá-lo.

M10
G28 H0
G0 C0

Esta parte de programa vai sempre ser chamada e não vai mudar, então é uma constante.

G97 S1200 M13

G97 é um função constante (a ferramenta tem de trabalhar com regime fixo e não variável, pois estamos fora do centro). Pode variar a rotação da ferramenta S e, também, o sentido de rotação (M13/M14)

G0 X100 Z3 M8

Os valores de X e Z não são constantes, mas dependem da operação de maquinamento.

G(98/99) G(74,83,,,)

H() K()

As funções G e os movimentos (repetições) dependem do tratamento requerido

G0 G96 G99 X400 Z400 M9

M15

M11

As linhas finais podem ser consideradas constantes (os valores de X e Z representam um ponto longe da peça). Colocamos o CNC em condições de velocidade de corte constante e avanço por rotação, que são normalmente as condições de trabalho utilizadas no torneamento.

O discurso parece ser mais claro: o que muda é o ciclo de perfuração ou o posicionamento. Em contraste, as ativações e desativações são elementos constantes, isto é, você deve sempre escrever.

Tudo bem! Nossa macro deve automaticamente lidar com todas as funções constantes, enquanto vai receber através argumentos os valores das funções variáveis.

Pergunta: Quais funções temos que escrever sempre?

Ativação do eixo C e do motorizado, a função G97, o avanço por minuto (ou por rotação), o afastamento da peça, desativação do eixo C e motorizado, no caso restabelecer o avanço por rotação.

Pergunta: Quais são as funções variáveis?

Rotação do motorizado, a posição de perfuração, a função de perfuração, a rotação do eixo C (repetições).

Agora podemos dizer sem dúvida que todas as funções que são manipuladas pela macro não mudam, enquanto a mudança deve ser recebida por argumentos, através da chamada macro. Nossa macro irá então receber os valores de posicionamento, as rotações do motor, a função G a ser executada, e o número de repetições a serem executadas (o ângulo primário e secundário).

Bem, dada uma forma geral para a macro vamos escolher certas letras específicas para cada argumento. A escolha é livre, mas é melhor, se for possível, manter o mesmo significado de um programa ISO, o que permite uma interpretação mais fácil pelo programador.

O SIGNIFICADO DE UM ARGUMENTO NÃO TEM NADA A VER COM O SIGNIFICADO DOS CODIGOS DE UM PROGRAMA ISO.

G210 XYZABCD F as letras podem ter qualquer significado. Por exemplo, Z pode ser uma rotação. Você entende que, neste caso, estamos um pouco desorientados, pois teríamos preferido que Z representasse o valor do eixo Z já que este é o significado "normal" de um programa qualquer.

Isto nem sempre é possível ... mas faça o melhor para obtê-lo!

A macro que escreveremos foi concebida para lidar com incrementos angulares constante, isto é, o ângulo entre os furos é sempre o mesmo.

Chamamos de G210 a macro e usaremos os seguintes argumentos (lembrando o que dissemos antes)

D	diâmetro de perfuração
Z	profundidade de perfuração
R	aproximação rápida ferramenta
F	avanço ferramenta
Q	remoção na perfuração
I	primeiro ângulo referido X
J	ângulo entre os furos
K	número de furos
M	função de perfuração
S	rotação motorizado (ferramenta)

Em seguida, a macro será chamada desta maneira G210 D Z R F Q I J k M S

Talvez funcione! Logo mostramos o que podemos fazer.
Supondo que você tenha escrito a macro, escrevemos dois programas: o primeiro de modo clássico, o segundo utilizando a macro G210.
O que podemos deduzir?

Programa ISO	Programa macro G210
T0101	T0101
M10	G210 D100 Z-2 R3 F100 Q10 I0 J22.5 k0 M83 S1200
G28 H0	
G0 C0	
G97 S1200 M13	
G0 X100 Z3 M8	
G98 G83 Z-2 F100	
H22.5 K15	
G0 G96 G99 X300 Z300 M9	
M15	
M11	

É um grande passo! (Vamos ver o significado da escrita do G210 para frente)
Com a linha G210 DZRFQIJkMS quis executar a função G83 as quotas D100, Z-2, R3, com uma velocidade de F100 com remoção igual a Q, utilizando um número de rotações apropriado S1200, um certo número de repetições K15 e, por fim, a partir de um ângulo i=0° por incrementos angulares fixos de 22,5°.
Note-se que não há nenhuma menção em relação ao eixo C ou a motorizado: o cnc irá automaticamente manipular as funções escondidas.
Este exemplo destaca algumas características importantes: é curto, é fácil de interpretar, pode ser mudado muito rapidamente e, acima de tudo, tem baixa possibilidade de erro, **isto é automatizado**. O operador não se preocupa com a maioria dos aspectos "secundário", mas concentra-se na função de perfuração. O operador não levanta a questão de como a macro age, só precisa saber o que faz e ter a confiança de que ela vai trabalhar direitinho (assim como nós confiamos nas funções do CNC). Agora sabemos que é possível transformar o nosso programa clássico, em macro..... mas a macro onde está?
É claro que agora você tem que dar o segundo passo: a escrita!

Cada programa macro deve sempre começar a partir de uma análise do problema a fim de determinar a melhor solução. Nós analisamos o problema, chegamos à escolha dos argumentos. Agora temos que generalizar um programa adaptado-o para cada situação possível. A solução não é geralmente única, é bom escolher a mais simples, aplicável ao maior número de comandos. Isso significa usar funções simples que normalmente existem em todos os controles.
A escritura de um programa macro é livre (é só respeitar as regras de sintaxe), cada um tem sua própria técnica. Eu adoto um sistema pessoal que é o seguinte:

- Cabeçalho, tipo de chamada e específicas gerais
- Significado dos argumentos
- Análise das variáveis: valor e dependência
- Salvar dados modais
- Cálculo e corpo executivo da macro
- Restauração dos dados modais

Em seguida, realizaremos esse esquema, dando uma explicação passo a passo (Você continue usando o método pessoal se tiver).

Cabeçalho, tipo de chamada e específicas gerais

A macro deve ter um nome, deve explicar a chamada e você deve comentar sobre qualquer aspecto particular, apontando que a escrita do programa não vai acabar até que você fique 100% satisfeito. Isto deve ser feito por passos: escritura, teste, modificação, e de novo a ser repetido até que alcançamos o efeito desejado. Lembre-se sempre a comentar o seu programa!

O9009(MACRO PERFURAÇÃO)
(CHAMADA G210 DZRFQIJkMS)
(CHAMADA G65 P9009 DZRFQIJkMS)
(CUIDADO I, J, K)
(ESCREVER NA ORDEME EM QUE APARECEM)

Número do Programa e Título
chamada código G
chamada código G65
comentários importantes

Significado dos argumentos

Descrever o significado dos argumentos não é pelo operador, pois ele nunca entra no programa macro: então por que descrever? Você faz isso para você! É importante saber o significado exato de cada linha de cada sujeito e cada variável ... depois de um tempo....eu desafio você a lembrar se não há um comentário (breve, mas eficaz).

(SIGNIFICADO ARGUMENTOS)
(D DIÂMETRO DE PERFURAÇÃO)
(Z PROFUNDIDADE DE PERFURAÇÃO)
(R ABORDAGEM EM RAPIDO)
(F AVANÇO)
(Q REMOÇÃO)
(I PRIMEIRO ANGULO REFERIDO A X)
(J ANGULO ENTRE OS FUROS)
(K NUMERO FUROS)
(M FUNÇÃO DE PERFURAÇÃO)
(S ROTAÇÃO MOTORIZADO)

Análise dos argumentos: presença, valor, e dependência

Quando chamamos um macro, é preciso usar argumentos certos que são aqueles que a macro espera. Se escrevermos G210..B.. seria um erro! Igualmente esquecer um argumento pode levar a consequências inesperadas. E' portanto, importante verificar os argumentos, seja na presença que no valor. Uma variável pode ter um intervalo de valores dentro do qual for definida: fora deste campo não é permitido ir. Consideramos cada argumento e verificamos estes aspetos!

Presença

A presença/ausência dos argumentos deve ser previamente analisada.

IF[#7EQ#0]THEN#3000=100 (FALTA D)
IF[#26EQ#0]THEN#3000=100 (FALTA Z)
IF[#18EQ#0]THEN#3000=100 (FALTA R)
IF[#7EQ#0]THEN#3000=100 (FALTA F)
IF[#17EQ#0]THEN#3000=100 (FALTA Q)
IF[#4EQ#0]THEN#3000=100 (FALTA I)
IF[#5EQ#0]THEN#3000=100 (FALTA J)
IF[#6EQ#0]THEN#3000=100 (FALTA K)
IF[#13EQ#0]THEN#3000=100 (FALTA M)
IF[#19EQ#0]THEN#3000=100 (FALTA S)

As linhas escritas emitirão um aviso se um argumento for ausente.

Controle valores

(D #7)

O diâmetro não pode ter valor negativo, não pode ser 0 caso contrário, a macro não teria sentido! Assim, apenas os valores positivos. Se desejar, você pode inserir um diâmetro máximo definido como o valor máximo de torneamento.

IF[#7LE0]THEN#3000=1 (DIÂMETRO ERRADO)

(Z #26)

Z pode assumir qualquer valor: negativo, 0, positivo, dependendo da posição origem. Não podemos por limites! A menos que você trabalha sempre em determinadas condições, não podemos assumir valores certos ou errados para esta

variável.

(R #18)

O valor de R, tal como Z não está sujeito a restrições podendo ter qualquer valor.

(F #9)

O avanço é uma entidade real, física, e como tal não deve possuir valor negativo ou nulo.

IF[#9LE0]THEN#3000=2 (AVANÇO ERRADO)

(Q #17)

Pelas razões já dadas também Q deve ser positiva

IF[#17LE0]THEN#3000=3 (REMOÇÃO ERRADA)

(I #4)

O valor de I é o ângulo inicial (primeiro furo), em referência ao eixo X.

Tecnicamente, poderíamos assumir qualquer valor de I.

Este não é um discurso simples, porque muitos fatores entram em jogo.

Pode-se pensar na circunferência completa (360°), ou na sua metade (180°). Pode-se considerar qualquer ângulo positivo ou negativo, maior de 360° ou menor de -360° . Muito depende de quem escreve a macro! Eu vou considerar o ângulo de 180° , como valor máximo, deste modo um valor de $I=234^\circ$ deve ser expresso com um ângulo negativo (-126°).

IF[#4GT180]THEN#3000=4 (ÂNGULO I GRANDE)

IF[#4LT-180]THEN#3000=4 (ÂNGULO I GRANDE)

(J #5)

J representa o ângulo interposto entre os furos, o seu valor reflete o que dissemos anteriormente. O sinal de J será utilizado para determinar o movimento do eixo C.

IF[#5GT180]THEN#3000=5 (ÂNGULO J GRANDE)

IF[#5LT-180]THEN#3000=5 (ÂNGULO J GRANDE)

(Não é minha intenção dar a solução para todos os problemas mas propor algumas reflexões)

(K #6)

O valor de K representa o número de furos. Não pode ser negativo (que sentido teria?). Por conveniência, vamos supor que o valor de $K=0$ indica o colapso total do perímetro de acordo com o ângulo J. Suponha que você tem 6 furos em 60° . Neste exemplo, a circunferência é divisível completamente ($360/6=60$). Nestes casos, simplesmente escreveremos $K0$ e o programa irá saber que vai fazer uma série de perfuração dividindo a circunferência.

IF[#6LT0]THEN#3000=6 (VALOR K INCORRETO)

Na verdade pode-se usar o argumento K para escolher o sentido de rotação do eixo C ($K < 0$ movimento inverso do eixo C), e isso mostra como um programa macro não é único, mas pode ser escrito de várias formas: o mais importante é tomar uma decisão, avaliar e ter decisão em usá-la!

(M #13)

As funções que consideramos são G74, G83, G84, G86 (funções frontais de perfuração). M, portanto, deve ter um destes números e nada mais. Avaliaremos posteriormente o valor de M.

(Quando você estará com sua macro pronta, seria muito interessante introduzir as outras funções de perfuração, as laterais pense nisso).

(S #19)

A rotação do motorizado deve ser real, positiva. Se você deseja definir limites a este argumento, pode. Por exemplo, definir um valor máximo ou um valor mínimo.

IF[#19LE0]THEN#3000=7 (ROTAÇÃO ERRADA)

IF[#19GT5000]THEN#3000=8 (ROTAÇÃO EXCESSIVA)

Eis um exemplo definindo um valor máximo de 5000: não é uma necessidade, é apenas um controle a mais!

Dependência entre variáveis

Verificar os valores individuais não é suficiente, precisamos também analisar as relações entre as variáveis, o que nem sempre é fácil!

Por exemplo: R e Z são relacionadas!. O valor de R deve ser maior do que Z, dado que R é a abordagem rápida, enquanto Z a profundidade de perfuração.

IF[#26GT#18]THEN#3000=9 (VALOR Z/R ERRADO)

Salvar dados modais

Nesta seção você terá que salvar o estado da máquina quando você entra na macro. Esta condição, se for alterada pelo programa, vai ser retomada na saída, garantindo que as condições atuais não sejam alteradas. É muito importante não negligenciar este aspecto, pouco considerado em um programa paramétrico pois ele é executado junto ao código ISO. O conceito, portanto é: mudar tudo o que for necessário, enfim restaurar a condição encontrada ao entrar no programa macro. Desta forma, o operador não ficará louco para entender por que tive mudanças no estado da cnc sem que ele tinha escrito algo!

(SALVAR DADOS MODAIS)

#1=#5001 (POSIÇÃO ATUAL X)

#2=#5002 (POSIÇÃO ATUAL Z)

#3=#4010 (G98/G99)

#8=#4013 (G96/G97)

As variáveis guardam as funções ativas (G96/G97 e G98/G99) e a posição atual (X e Z). Salvar um dado (aqui dado significa qualquer coisa, função, posição...) significa colocá-lo em uma variável que nunca será usada para que você possa recuperá-lo quando necessário.

Esta seção do programa macro, não executa nenhum cálculo ou movimento.

A primeira parte é uma série de comentários que descrevem a macro, a segunda é uma análise da chamada (argumentos) e serve para não executar a macro em presença de erros. Agora entramos no verdadeiro programa, mas primeiro uma visão da nossa macro escrita até agora: uma leitura rápida para ver se nós perdemos algo.

O9009(MACRO PERFURAÇÃO)

(CHAMADA G210 DZRFQIJkMS)

(CHAMADA G65 P9009 DZRFQIJkMS)

(CUIDADO I, J, K)

(ESCREVER NA ORDEME EM QUE APARECEM)

(SIGNIFICADO ARGUMENTOS)

(D DIÂMETRO DE PERFURAÇÃO)

(Z PROFUNDIDADE DE PERFURAÇÃO)

(R ABORDAGEM EM RAPIDO)

(F AVANÇO)

(Q REMOÇÃO)

(I PRIMEIRO ANGULO FURO REFERIDO A)

(J ANGULO ENTRE OS FUROS)

(K NUMERO FUROS)

(M FUNÇÃO DE PERFURAÇÃO)

(S ROTAÇÃO MOTORIZADO)

(ANÁLISE ARGUMENTOS)

IF[#7EQ#0]THEN#3000=100 (FALTA D)

IF[#26EQ#0]THEN#3000=100 (FALTA Z)

IF[#18EQ#0]THEN#3000=100 (FALTA R)

IF[#7EQ#0]THEN#3000=100 (FALTA F)

IF[#17EQ#0]THEN#3000=100 (FALTA Q)

IF[#4EQ#0]THEN#3000=100 (FALTA I)

```

IF[#5EQ#0]THEN#3000=100 (FALTA J)
IF[#6EQ#0]THEN#3000=100 (FALTA K)
IF[#13EQ#0]THEN#3000=100 (FALTA M)
IF[#19EQ#0]THEN#3000=100 (FALTA S)
IF[#7LE0]THEN#3000=1 (DIÂMETRO ERRADO)
IF[#9LE0]THEN#3000=2 (AVANÇO ERRADO)
IF[#17LE0]THEN#3000=3 (REMOÇÃO ERRADA)
IF[#4GT180]THEN#3000=4 (ÂNGULO I GRANDE)
IF[#4LT-180]THEN#3000=4 (ÂNGULO I GRANDE)
IF[#5GT180]THEN#3000=5 (ÂNGULO J GRANDE)
IF[#5LT-180]THEN#3000=5 (ÂNGULO J GRANDE)
IF[#6LT0]THEN#3000=6 (VALOR K INCORRETO)
IF[#19LE0]THEN#3000=7 (ROTAÇÃO ERRADA)
IF[#19GT5000]THEN#3000=8 (ROTAÇÃO EXCESSIVA)
IF[#26GT#18]THEN#3000=9 (VALOR Z/R ERRADO)
(SALVAR DADOS MODAIS)
#1=#5001 (POSIÇÃO ATUAL X)
#2=#5002 (POSIÇÃO ATUAL Z)
#3=#4010 (G98/G99)
#8=#4013 (G96/G97)

```

Tens lido? Querem fazer mudanças? Algo errado? Lembrem-se, pode-se mudar o que quiser e quando quiser... por exemplo... eu li novamente o programa e percebi que é possível introduzir ligeiras modificações. Corrigir ou modificar um programa é comum, por isso é lógico tentar analisar este aspecto! As razões para as mudanças sempre surgem a partir de erros ou melhorias, por exemplo fazer a macro mais fácil para o operador introduzindo uns argumentos padrão.

(D diâmetro de perfuração)
 Não é um argumento estritamente necessário, pode-se assumir que, em caso de falta, o valor atual de X o representa. Então:

```

IF[#7EQ#0]THEN#3000=100 (FALTA D) tornará
IF[#7EQ#0]THEN#7=#1      (posição atual)

```

Esta mudança é incluída para mostrar algo a considerar com muito cuidado: **os argumentos de uma macro deverão ser o menor possível**, a fim de facilitar a sua utilização (uma chamada curta é mais fácil de lembrar!) Além disso, este aspecto mostra o uso de argumentos padrões: um argumento não presente na chamada macro pode ter sido esquecido ou pode ter um valor padrão.

O mesmo princípio o usamos para R! Vamos por R=3 por defeito dado que geralmente a origem Z é colocada sobre a face da frente. Se não fosse assim você terá que usar apenas o R.

```

IF[#18EQ#0]THEN#3000=100 (FALTA R) vai mudar para
IF[#18EQ#0]THEN#18=3

```

O parâmetro Q pode ser utilizado de uma maneira diferente lembrando um conceito de programação. A ausência de Q na função de perfuração G83 a transforma em uma função G81. Os tornos não tem esse código (G81), o tentaremos simular (esta consideração pode não se aplicar a todos os tornos.. cuidado).

```

IF[#17EQ#0]THEN#3000=100 (FALTA Q) esta linha vai ser tirada

```

O ângulo primário é importante se existem processos diferentes que devem ser colocados de uma maneira específica. Portanto é essencial parar o eixo C em condições adequadas. Neste caso, e na maior parte dos processos de ciclos, não importa o início, o que é importante é sempre partir do zero: este será o padrão!

```

IF[#4EQ#0]THEN#3000=100 (FALTA I) tornará

```



```
IF[#4EQ#0]THEN#4=0
```

Este modo de pensar (não abuse muito) permite escrever chamada macro genérica e outras específicas. Na prática, nós definimos uma condição geral, a mais utilizada, de modo a simplificar o trabalho do programador. A chamada macro pode, então, assumir um aspecto assim

```
G210 Z J F M S
```

que é simplificado (os argumentos ausentes vão assumir o valor padrão).

VOCÊ DEVE TOMAR MUITO CUIDADO EM RELAÇÃO AOS ARGUMENTOS PADRÕES, MAS ACIMA DE TUDO VOCÊ TEM QUE EXPLICAR BEM O USO.

É importante descrever estas condições nos comentários!

```
(ARGUMENTOS OBRIGATÓRIOS Z J F M S)
(ARGUMENTOS QUE PODEM FALTAR, Q)
(VALORES PADRÃO ARGUMENTOS)
(K=0 I=0 R=3 D=POSIÇÃO ATUAL)
```

A função do motorizado M13 é a condição padrão, pois uma broca geralmente roda em M3. Pode acontecer de usar ferramentas específicas para torno para executar uma perfuração, no caso o sentido de rotação muda. Desde que nós não queremos introduzir uma nova variável, vamos utilizar o valor de S para identificar o sentido correto de rotação (outra maneira de usar variáveis tendo dois valores)

```
S>0 -->> M13 (padrão)
S<0 -->> M14 >>
então vamos considerar este aspecto
```

```
IF[#19LE0]THEN#3000=7(ROTAÇÃO ERRADA)    se tornará
IF[#19EQ0]THEN#3000=7(ROTAÇÃO ERRADA)    e
#11=13
IF[#19LT0]THEN#11=14
```

também precisamos usar sempre o valor positivo de S

```
#19=ABS[#19]
```

Por ultimo devemos considerar a rotação máxima negativa

```
IF[#19LT-5000]THEN#3000=8(ROTAÇÃO EXCESSIVA)
```

Escreva sua macro, e se for possível, apresente uma condição geral de chamada para facilitar o uso!

Bem, a nossa macro alterada:

```
O9009(MACRO PERFURAÇÃO)
(CHAMADA G210 DZRFQIJkMS)
(CHAMADA G65 P9009 DZRFQIJkMS)
(SIGNIFICADO ARGUMENTOS)
(D DIÂMETRO DE PERFURAÇÃO)
(Z PROFUNDIDADE DE PERFURAÇÃO)
(R ABORDAGEM EM RAPIDO)
(F AVANÇO)
(Q REMOÇÃO)
(I PRIMEIRO ANGULO FURO REFERIDO A)
(J ANGULO ENTRE OS FUROS)
(K NUMERO FUROS)
(M FUNÇÃO DE PERFURAÇÃO)
(S ROTAÇÃO MOTORIZADO)
(CUIDADO I, J, K)
(ESCREVER NA ORDEME EM QUE APARECEM)
(ARGUMENTOS OBRIGATÓRIOS Z J F M S)
```

```

(ARGUMENTOS QUE PODEM FALTAR, Q)
(VALORES PADRÃO ARGUMENTOS)
(K=0 I=0 R=3 D=POSIÇÃO ATUAL)
(ANÁLISE ARGUMENTOS)
IF[#7EQ#0] THEN#7=#1
IF[#26EQ#0] THEN#3000=100 (FALTA Z)
IF[#18EQ#0] THEN#18=3
IF[#7EQ#0] THEN#3000=100 (FALTA F)
IF[#17EQ#0] THEN#3000=100 (FALTA Q)
IF[#4EQ#0] THEN#4=0
IF[#5EQ#0] THEN#3000=100 (FALTA J)
IF[#6EQ#0] THEN#3000=100 (FALTA K)
IF[#13EQ#0] THEN#3000=100 (FALTA M)
IF[#19EQ#0] THEN#3000=100 (FALTA S)
IF[#7LE0] THEN#3000=1 (DIÂMETRO ERRADO)
IF[#9LE0] THEN#3000=2 (AVANÇO ERRADO)
IF[#17LE0] THEN#3000=3 (REMOÇÃO ERRADA)
IF[#4GT180] THEN#3000=4 (ÂNGULO I GRANDE)
IF[#4LT-180] THEN#3000=4 (ÂNGULO I GRANDE)
IF[#5GT180] THEN#3000=5 (ÂNGULO J GRANDE)
IF[#5LT-180] THEN#3000=5 (ÂNGULO J GRANDE)
IF[#6LT0] THEN#3000=6 (VALOR K INCORRETO)
IF[#19EQ0] THEN#3000=7 (ROTAÇÃO ERRADA)
IF[#19GT5000] THEN#3000=8 (ROTAÇÃO EXCESSIVA)
IF[#19LT-5000] THEN#3000=8 (ROTAÇÃO EXCESSIVA)
IF[#26GT#18] THEN#3000=9 (VALOR Z/R ERRADO)
(SALVAR DADOS MODAIS)
#1=#5001 (POSIÇÃO ATUAL X)
#2=#5002 (POSIÇÃO ATUAL Z)
#3=#4010 (G98/G99)
#8=#4013 (G96/G97)
----
----
#11=13
IF[#19LT0] THEN#11=14
#19=ABS[#19]

```

(seria bom considerar outras dependências entre os argumentos).

Cálculos e corpo da macro

Agora temos que escrever a estratégia de usinagem. O programa tem sua base no que vimos antes: a macro não inventa nada de novo, mas automatiza o processo.

```

M10
G28H0
G0C0

```

Esta parte inicializa o eixo C

Ativamos o motorizado, e retomamos a escrita anterior em relação a M13/M14.

```

#11=13                (#11 representa o valor de M13/M14)
IF[#19LT0] THEN#11=14  (S<0? -->> M14, S>0? -->> M13)
#19=ABS[#19]          (S é sempre um valor positivo)
G97M#11S#19           (velocidade fixa, função e velocidade do motorizado)

```

```
G0X#7Z[#18+50]C#4
```

Posicionamento ao valor de X correspondente o diâmetro (por programa ou de acordo com a posição atual), enquanto que o eixo Z assume um valor igual ao valor de R mais uma porção a ser considerada como quota de segurança. O valor de 50 é escolhido como o padrão, você pode escolher o que quiser! Claro, também o eixo C tem que ser posicionado.

Agora que estamos posicionados, podemos analisar a função de perfuração a ser executada (G74, G83, G84, G86).

```
IF[#13EQ86]GOTO10 se for G86 desvia para N10
IF[#13EQ83]GOTO20 se for G83 desvia para N20
IF[#13EQ74]GOTO30 se for G74 desvia para N30
```

Agora, se não tiver chamado G83 ou G86 ou G74 significa que queremos executar o G84. Nos concentramos nesta função, referindo-se após as outras. O primeiro aspecto a considerar é a presença do G84; se isso não acontecer, nós entramos um valor incorreto da função de perfuração na chamada macro (se não for G83 ou G86 ou G74 ou G84 mesmo ... O que é?)

```
IF[ROUND[#13]NE84]THEN#3000=10 (FUNÇÃO ERRADA)
```

A função de roscagem G84, é especial, porque é um movimento sincronizado de rotação e avanço: é preciso, portanto, saber o passo da rosca para realizar o movimento correto. Não querendo introduzir outra variável, decidi usar a seguinte convenção:

M84 + passo, desta forma M84.passo (por exemplo G84.175)

E' por esta razão que precisamos usar a função ROUND para encontrar o valor 84. O modo encontrado para expressar o passo, fará com que esta macro não pode ser utilizada para roscas com um passo maior do que 9.999 ... que nos parece uma solução aceitável (se você tem exigências extremas me avise). Assim, o valor de M é formada por duas partes, que representam, a parte inteira a função 84, enquanto o decimal será o passo da rosca. O passo deve ser escrito sem vírgulas, por exemplo passo 1,25 será escrito 84.125, passo 2.5 será escrito 84.250, passo 0,5 será escrito 84.05

```
#10=ROUND[#13]      encontramos o valor inteiro (84)
#10=[#13-#10]*10,   dessa forma, encontramos o passo (84.175-84)*10=1,75
#13=ROUND[#13]      encontramos o valor inteiro (84)
```

O passo será utilizado para o cálculo da velocidade ferramenta.
Ha dois métodos: avanço por volta ou por minuto! (G98/G99)

Vamos agir da seguinte forma: para as funções G74/G83/G86 vamos usar G98, pela função G84 usaremos o G99.

Execução função de rosqueamento

```
G99G#13Z#26F#10R#18      função G84
GOTO40                    ciclo repetitivo
```

Bem, vamos concentrar nas outras funções (N10, N20, N30).

```
(G86)
N10G98G#13Z#26F#9R#18
GOTO40                    ciclo repetitivo
```

```
(G83)
N20IF[#17EQ#0]GOTO25      se Q está ausente desvia para função sem Q
G98G#13Z#26F#9R#18Q#17    função com presença de Q
GOTO40                    ciclo repetitivo
N25G98G#13Z#26F#9R#18    com nenhuma função Q
GOTO40                    ciclo repetitivo
```

```
(G74)
N30G98G#13Z#26F#9R#18Q#17
```

(O uso da #13 para encontrar o valor da função de perfuração é usada para manter as relativas funções relacionadas a uma variável. Isso garante que em caso de mudança do programa não devemos nos preocupar com o valor da função. E' claro que no contexto podemos escrever cada função com o próprio nome).

Agora temos que escrever o ciclo repetitivo

N40 (CICLO DE REPETIÇÃO)
H#5K[# 6-1]

K tem o valor diminuído de 1 pois o primeiro furo já foi realizado.

Nossa macro acabou! Faltas só restaurar os dados modais antes de sair

Restaurar a entrada significa colocar a CNC como o encontramos. Você se lembra dos valores armazenados na entrada?

1 = # 5001 (POS X)
2 = # 5002 (POS Z)
3 = # 4010 (G98/G99)
8 = # 4013 (G96/G97)

então

G#8G#3G0X#24Z#25M9 (por exemplo G99 G96 G0 X400 Z400)
M5
M15
M11
M99

Certamente as variáveis 3 e 8, são importantes, enquanto aos valores de X e Z podemos agir de acordo com as necessidades.

Bem, a nossa macro parece ter escrita: é bom corrigir eventuais erros ou modificá-la de acordo com os códigos que você usa. Deixe-me saber se você está satisfeito.

Aqui está a macro completa.

O9009(MACRO PERFURAÇÃO)
(CHAMADA G210 DZRFQIJkMS)
(CHAMADA G65 P9009 DZRFQIJkMS)
(SIGNIFICADO ARGUMENTOS)
(D DIÂMETRO DE PERFURAÇÃO)
(Z PROFUNDIDADE DE PERFURAÇÃO)
(R ABORDAGEM EM RAPIDO)
(F AVANÇO)
(Q REMOÇÃO)
(I PRIMEIRO ANGULO FURO REFERIDO A)
(J ANGULO ENTRE OS FUIROS)
(K NUMERO FUIROS)
(M FUNÇÃO DE PERFURAÇÃO)
(S ROTAÇÃO MOTORIZADO)
(CUIDADO I, J, K)
(ESCREVER NA ORDEME EM QUE APARECEM)
(ARGUMENTOS OBRIGATÓRIOS Z J F M S)
(ARGUMENTOS QUE PODEM FALTAR, Q)
(VALORES PADRÃO ARGUMENTOS)
(K=0 I=0 R=3 D=POSIÇÃO ATUAL)
(ANÁLISE ARGUMENTOS)
IF[#7EQ#0] THEN#7=#1
IF[#26EQ#0] THEN#3000=100 (FALTA Z)
IF[#18EQ#0] THEN#18=3
IF[#7EQ#0] THEN#3000=100 (FALTA F)
IF[#17EQ#0] THEN#3000=100 (FALTA Q)
IF[#4EQ#0] THEN#4=0
IF[#5EQ#0] THEN#3000=100 (FALTA J)
IF[#6EQ#0] THEN#3000=100 (FALTA K)
IF[#13EQ#0] THEN#3000=100 (FALTA M)

```

IF[#19EQ#0] THEN#3000=100 (FALTA S)
IF[#7LE0] THEN#3000=1 (DIÂMETRO ERRADO)
IF[#9LE0] THEN#3000=2 (AVANÇO ERRADO)
IF[#17LE0] THEN#3000=3 (REMOÇÃO ERRADA)
IF[#4GT180] THEN#3000=4 (ÂNGULO I GRANDE)
IF[#4LT-180] THEN#3000=4 (ÂNGULO I GRANDE)
IF[#5GT180] THEN#3000=5 (ÂNGULO J GRANDE)
IF[#5LT-180] THEN#3000=5 (ÂNGULO J GRANDE)
IF[#6LT0] THEN#3000=6 (VALOR K INCORRETO)
IF[#19EQ0] THEN#3000=7 (ROTAÇÃO ERRADA)
IF[#19GT5000] THEN#3000=8 (ROTAÇÃO EXCESSIVA)
IF[#19LT-5000] THEN#3000=8 (ROTAÇÃO EXCESSIVA)
IF[#26GT#18] THEN#3000=9 (VALOR Z/R ERRADO)
(SALVAR DADOS MODAIS)
M10 (ativação eixo C)
G28H0 (eixo C origem cnc)
G0C0 (eixo C absoluto)
#11=13 (#11 representa o valor de M13/M14)
IF[#19LT0] THEN#11=14 (S<0? -->> M14, S>0? -->> M13)
#19=ABS[#19] (S é sempre um valor positivo)
G97M#11S#19 (motorizado, rotação fixa, S)
G0X#7Z[#18+50]C#4 (posicionamento)
IF[#13EQ86] GOTO10 (se for G86 vai para N10)
IF[#13EQ83] GOTO20 (se for G83 vai para N20)
IF[#13EQ74] GOTO30 (se for G74 vai para N30)
IF[ROUND[#13]NE84] THEN#3000=10 (FUNÇÃO ERRADA) (controle existência funções G)
#10=ROUND[#13] (função G)
#10=[#13-#10]*10 (passo roscagem)
#13=ROUND[#13]
G99G#13Z#26F#10R#18 (função G84)
GOTO40 (desvia para rotina repetitiva)
N10G98G#13Z#26F#9R#18 (função G86)
GOTO40
N20IF[#17EQ#0] GOTO25 (se Q está ausente desvia para função sem Q)
G98G#13Z#26F#9R#18Q#17 (função G83 com presença de Q)
GOTO40 (ciclo repetitivo)
N25G98G#13Z#26F#9R#18 (função G83 sem Q)
GOTO40
N30G98G#13Z#26F#9R#18Q#17 (função G74)
(CICLO DE REPETIÇÃO)
N40H#5K[# 6-1]
G#8G#3G0X#24Z#25M9 (volta às condições na entrada)
M5
M15 (desativa motorizado)
M11 (desativa eixo C)
M99 (saída)

```

Qualquer macro deve sempre ser comentada! Escreva qualquer coisa, ... a memória não é tão forte quanto parece! Salve o seu programa em um PC para que você tenha sempre à mão uma cópia fiel do trabalho. O programa que ficará no CNC não precisa de comentários (como indicado abaixo).

```

O9009 (MACRO PERFURAÇÃO)
IF[#7EQ#0] THEN#7=#1
IF[#26EQ#0] THEN#3000=100 (FALTA Z)
IF[#18EQ#0] THEN#18=3
IF[#7EQ#0] THEN#3000=100 (FALTA F)
IF[#17EQ#0] THEN#3000=100 (FALTA Q)
IF[#4EQ#0] THEN#4=0
IF[#5EQ#0] THEN#3000=100 (FALTA J)
IF[#6EQ#0] THEN#3000=100 (FALTA K)
IF[#13EQ#0] THEN#3000=100 (FALTA M)
IF[#19EQ#0] THEN#3000=100 (FALTA S)
IF[#7LE0] THEN#3000=1 (DIÂMETRO ERRADO)

```

```

IF[#9LE0] THEN#3000=2 (AVANÇO ERRADO)
IF[#17LE0] THEN#3000=3 (REMOÇÃO ERRADA)
IF[#4GT180] THEN#3000=4 (ÂNGULO I GRANDE)
IF[#4LT-180] THEN#3000=4 (ÂNGULO I GRANDE)
IF[#5GT180] THEN#3000=5 (ÂNGULO J GRANDE)
IF[#5LT-180] THEN#3000=5 (ÂNGULO J GRANDE)
IF[#6LT0] THEN#3000=6 (VALOR K INCORRETO)
IF[#19EQ0] THEN#3000=7 (ROTAÇÃO ERRADA)
IF[#19GT5000] THEN#3000=8 (ROTAÇÃO EXCESSIVA)
IF[#19LT-5000] THEN#3000=8 (ROTAÇÃO EXCESSIVA)
IF[#26GT#18] THEN#3000=9 (VALOR Z/R ERRADO)
M10
G28H0
G0C0
#11=13
IF[#19LT0] THEN11=14
#19=ABS[#19]
G97M#11S#19
G0X#7Z[#18+50]C#4
IF[#13EQ86] GOTO10
IF[#13EQ83] GOTO20
IF[#13EQ74] GOTO30
IF[ROUND[#13]NE84] THEN#3000=10 (FUNÇÃO ERRADA)
#10=ROUND[#13]
#10=[#13-#10]*10
#13=ROUND[#13]
G99G#13Z#26F#10R#18
GOTO40
N10G98G#13Z#26F#9R#18
GOTO40
N20IF[#17EQ#0] GOTO25
G98G#13Z#26F#9R#18Q#17
GOTO40
N25G98G#13Z#26F#9R#18
GOTO40
N30G98G#13Z#26F#9R#18Q#17
N40H#5K[# 6-1]
G#8G#3G0X#24Z#25M9
M5
M15
M11
M99

```

A macro não foi otimizada... não quero mudá-la agora... vos deixo refletir sobre futuras alterações ou mudanças.

Escrever a macro, como você viu não foi complicado.

Vamos executar o programa de Fausto da maneira tradicional e em seguida, usando a macro como abaixo

```

T0101 (CENTRAGEM)
G210 D100 Z-6 F150 J22.5 k0 M83 S1500
T0202 (FURAÇÃO D=10,5)
G210 D100 Z-40 F100 Q3 I22.5 k0 J45 M83 S1200
T0202 (FURAÇÃO D=10,2)
G210 D100 Z-40 F120 Q3 J45 K0 M83 S1400
T0303 (ALARGAMENTO D16*10)
G210 D100 Z-10 F80 I22.5 J90 K0 M83 S800
T0404 (ALARGAMENTO 16/90°)
G210 D100 Z-8 I-22.5 F70 J90 K0 M83 S700
T0505 (ROSCAGEM)
G210 D100 Z-40 F300 J45 k0 M84.175 S300
M30

```

Sem dúvida..... o que você acha?

A macro funciona considerando como centro de rotação X0 (coisa normal geralmente). Não foi concebida para a execução de furos, que são dispostos em circunferência que não tem o seu centro no centro X0 (ver figura abaixo). Mas é possível executar a usinagem de figura? Não! A macro permite realizar a perfuração da figura, mas de uma forma indireta, isto é, por extrapolação das circunferência de cada buraco. Isso, no entanto, faz perder o sentido dessa macro mas é fácil escrever uma apropriada! As regras e o conhecimento necessário você tem... tente!

----- Fig. -----

Lembrando este último discurso gostaria de propor um macro para centro de usinagem.

```
:9008(G8.. EM CIRCUNFERÊNCIA)
(G210 Z R F Q D I J K M)
(FUNÇÕES HABILITADAS G81/G86/G73/G83/G84 )
(Z #26 PROFUNDIDADE FURO)
(R #18 ABORDAGEM EM RÁPIDO)
(F #9 AVANÇO)
(Q #17 REMOÇÃO)
(D #7 DIAMETRO)
(I #4 ANGULO PRIMÁRIO)
(J #5 ANGULO SEGUNDO)
(J<0 SENTIDO HORÁRIO)
(J>0 SENTIDO ANTI-HORÁRIO)
(K #6 REPETIÇÕES)
(NB,,, SE K=0 K=360/B)
(M #13 FUNÇÃO G)
(NB,,, M84.175)
(ROSCAGEM + PASSO)
(84=G84 0.175 PASSO 1.75)
(SALVA DADOS MODAIS)
#1=#4001(G0, G01, G02, G03, G33)
#2=#4003(G90/G91))
#3=#5003(Z)
#24=#5001(X)
#25=#5002(Y)
(CONTROLE REPETIÇÕES)
IF[#6EQ0]THEN#6=360/[ABS[#5]]
IF[[#6-[ROUND[#6]]]GT0.001]THEN#3000=1(DIVISÃO IMPOSSÍVEL)
(DESVIA PARA AS FUNÇÕES APROPRIADAS)
IF[#13EQ81]THENGOTO10
IF[#13EQ86]THENGOTO10
IF[#13EQ73]THENGOTO20
IF[#13EQ83]THENGOTO20
IF[[ROUND[#13]]NE84]THEN#3000=2(FUNÇÕES ERRADAS)
(CALCULO PASSO ROSCAGEM)
#8=ROUND[#13]
#8=[#13-#8]*10
(AVANÇO)
#9=#8*#4119
(FUNÇÃO)
#13=ROUND[#13]
N10G#13F#9Z#26R#18K0
GOTO30
N20G#13F#9Z#26R#18Q#17K0
(RAIO)
N30#7=#7/2
(CICLO)
WHILE[#6NE0]DO1
(ABSOLUTO X)
```

```

#19=#24+[#7*[COS[#4]]]
(ABSOLUTO Y)
#20=#25+[#7*[SIN[#4]]]
(POSICIONAMENTO)
G90X#19Y#20
(NOVO ANGULO)
#4=#4+#5
(CONTADOR)
#6=#6-1
(FIM DO CICLO)
END1 ,
(REPRISTINA VALORES MODAIS)
G90G0Z#3
X#24Y#25
G#1G#2
M99

```

Esta macro é idêntica à macro escrita para torno mas com alguma diferença pequena, mas muito importante. A macro pode ser invocada com as seguintes funções: G81/G86/G73/G83/G84, deixo para vocês a inclusão das outras (G82, G85, G88, G89 ..). A principal diferença é que esta macro executa as funções de perfuração em qualquer lugar no plano de trabalho e não apenas referida a origem. No torno a peça roda ao longo do eixo Z, o centro de rotação é tomado como o centro do círculo. Em um centro de usinagem não temos uma condição tão estreita, por isso temos de ser capazes de calcular e realizar qualquer perfuração em qualquer lugar do plano ... o importante é que os furos são dispostos numa circunferência hipotética. Você deve se posicionar no centro do círculo e o programa vai fazer o resto. Outra diferença que encontramos é no cálculo dos pontos; aqui são extrapolados com as funções trigonométricas.

Agora podemos voltar para nossos torneiros e perguntar, você pode aplicar o mesmo princípio para realizar furações circulares em qualquer ponto da face? Tentem!

Uma última observação! Um aspecto interessante na realização de programas macro fica na portabilidade, quer dizer a capacidade de executar o mesmo programa em cnc diferente. Uma vez que cada cnc possui a própria macro, podemos mudar de maquina em maquina sem nenhuma modifica (pouquíssimas) no programa principal. Isto vale também entre cnc diferentes como tornos e centro de usinagem.

Bom, aporte cada modifica que quiser e bom trabalho!
 À espera de suas ideias, suas sugestões e, possivelmente de seus programas!
 Até

badade10@libero.it