

Escrever um programa macro de uso geral, é independente da máquina?
Somos livres para usar qualquer frase macro, variável ou função?
A final, cada programa macro pode ser executado em qualquer cnc?
E' o que vamos descobrir!

Iremos escrever um programa macro (para um centro de usinagem) que permite a inserção automática da origem através de um procedimento manual (sonda mecânica). O operador deve realizar a sondagem manualmente enquanto o programa irá calcular e escrever os valores corretos na tabela origem escolhida entre G54->G59. (E' possível usar a macro com qualquer sistema de referência, tais como G54.1 P250, mas nos referimos apenas às seis principais origens).

Lembro que o procedimento manual referido aos controles Fanuc é testar a peça, abrir a tabela de pontos zero e escrever o valor correto.

Para aqueles que lêem pela primeira vez este post e não possui conhecimento adequado referem-se aos artigos sobre "programação Fanuc Macro B" (por isso não vou repetir os conceitos básicos a menos que fossem essenciais).

Escrever um programa macro leva tempo (ou melhor, várias páginas), mas este não será o nosso principal problema: queremos ver se o programa pode ser executado em qualquer cnc. Por isso não realizaremos um programa "perfeito", mas o usaremos para fazer algumas considerações.

PRIMEIRA PARTE: ESCREVER O PROGRAMA MACRO

Primeiro passo, vamos dar uma aparência ao nosso programa macro.

- O operador chama a sonda
- O operador deve posicionar-se no ponto onde deseja criar a origem (sondar)
- Pressionando o botão START o cnc irá introduzir o valor correto na tabela origem
- O operador irá executar outra sondagem ou encerrar o programa.

Agora passamos a escolher o tipo de sondagem que nós queremos fazer
Decidimos criar um procedimento que pode sondar

- um ou mais pontos (X Y Z)
- a média de dois pontos (X e/ou Y)
- o centro de um círculo

Claro que podemos fazer combinações de sondagem se for legítimo.

Desta forma, a macro vai nos permitir criar uma origem sondando um ponto, ou um retângulo, ou um círculo, externos e internos que sejam.

Vamos usar a notação G e chamar o programa macro de G120.
Estes os argumentos utilizados

- X Y Z definir pontos
- I J para definir o ponto médio
- W para definir a origem
- C para definir a circunferência

A macro pode aceitar os seguintes argumentos: G120 X Y Z I J C W

Os argumentos não são todos obrigatórios, mas a presença indica o tipo de sondagem a ser realizado.

G120 X W, significa criar a origem W sondando o ponto X.
G120 X J Z W, significa criar a origem W sondando o ponto X, o ponto médio do eixo Y e sondando o ponto Z.

Lista sondagem (W=origem)

G120 X W	sondagem ponto X
G120 Y W	sondagem ponto Y
G120 X Y W	sondagem pontos X e Y
G120 Z W	sondagem ponto Z
G120 X Z W	sondagem ponto X e sondagem Z
G120 Y Z W	sondagem ponto Y e sondagem Z
G120 X Y Z W	sondagem pontos X Y Z
G120 I W	sondagem ponto médio X
G120 J W	sondagem ponto médio Y
G120 I J W	sondagem pontos médios X Y
G120 I Z W	sondagem ponto médio X e sondagem Z
G120 J Z W	sondagem ponto médio Y e sondagem Z
G120 I J Z W	sondagem pontos médios X Y e sondagem Z
G120 X J W	sondagem ponto X e ponto médio Y
G120 X J Z W	sondagem ponto X e ponto médio Y e sondagem Z
G120 Y I W	sondagem ponto Y e ponto médio X
G120 Y I Z W	sondagem ponto Y e ponto médio X e sondagem Z
G120 C W	sondagem centro circunferência
G120 C Z W	sondagem centro circunferência e sondagem Z

É natural que devemos lidar com as inconsistências que podem ocorrer com uma escritura incorreta. G120 X I Y Z W apresenta os argumentos X e I que são incompatíveis: sondagem ponto X ou sondagem ponto médio X???

É claro que o argumento W, a origem, não pode faltar caso contrário, por que escrever o programa?

Vamos escrever algumas linhas!

```
O9018 (G54-G59 MACRO ORIGEM)
(USO G120 X Y Z I J C W)
(X PONTO X)
(Y PONTO Y)
(Z PONTO Z)
(I MÉDIA DOS PONTO EM X)
(J MÉDIA DOS PONTO EM Y)
(C CENTRO CÍRCULO)
(W ORIGEM 54->59)
```

As linhas definem o nome do programa, a chamada macro(G120) e quais são os argumentos utilizados.

O que deve estar sempre presente? W!!

Então verificamos imediatamente a presença de W sem o qual o programa não tem sentido.

```
IF[#23NE#0]THEN #3000=1(FALTA A ORIGEM W)
```

Além disso, os valores aceitos para a origem são: 54->(G54) até 59->(G59), que representam os seis sistemas básicos do Fanuc.

```
IF[#23LT54]THEN #3000=2(ORIGEM ERRADA)    o valor é inferior a 54
IF[#23GT59]THEN #3000=2(ORIGEM ERRADA)    o valor é maior que 59
```

Se o argumento W tem um valor inferior a 54 ou superior a 59 vai ter um alarme que nos avisa do erro.

Usamos o valor de #23 para obter um número como um lembrete para definir a origem

```
#23=#23-53
```

A partir desta diferença, o que origina

```
#23=1 G54          #23=2 G55          #23=3 G56
#23=4 G57          #23=5 G58          #23=6 G59
```

outros valores não são permitidos!

Este número vai nos ajudar mais tarde para identificar a origem correta

Como primeira ação da chamada macro G120, vamos colocar a sonda em um lugar adequado da CNC por exemplo a meio caminho do eixo X e Y e Z se for necessário. Não é essencial, mas desejável para trazer a sonda perto da peça.

```
T0M6 (SONDA)
G90G0G40G49G53X500Y-300
G90G0G53Z-200
```

Por segurança, nós excluimos a compensação de raio e comprimento. O movimento é feito em G53 (coordenadas máquina): é simples ajustar os valores de acordo com o tamanho da CNC.

Nossa macro toma forma

```
O9018 (G54-G59 MACRO ORIGEM)
(USO G120 X Y Z I J W C)
(X PONTO X)
(Y PONTO Y)
(Z PONTO Z)
(I MÉDIA DOS PONTO X)
(J MÉDIA DOS PONTO Y)
(C CENTRO CÍRCULO)
(W ORIGEM 54->59)
IF[#23NE#0]THEN #3000=1(FALTA A ORIGEM W)
IF[#23LT54]THEN #3000=2(ORIGEM ERRADA)
IF[#23GT59]THEN #3000=2(ORIGEM ERRADA)
#23=#23-53
T0M6 (PROBE)
G90G0G40G49G53X500Y-300
G90G0G53Z-200
```

E agora?

É claro que teremos que verificar os argumentos, seja na presença que no valor e naturalmente as inconsistências entre eles.

Este programa macro é particular porque os argumentos que devem estar sempre presentes são W e mais um, ou mais de um, o que determina o tipo de sondagem.

Precisamos também de gerenciar conflitos de escrita como:

G120 X I W: o que você quer sondar o ponto X ou o ponto médio em X?

Escrever assim não é aceitável pois o cn não saberia o que fazer.

Quais valores os argumentos podem assumir? Vamos resumi-los aqui!

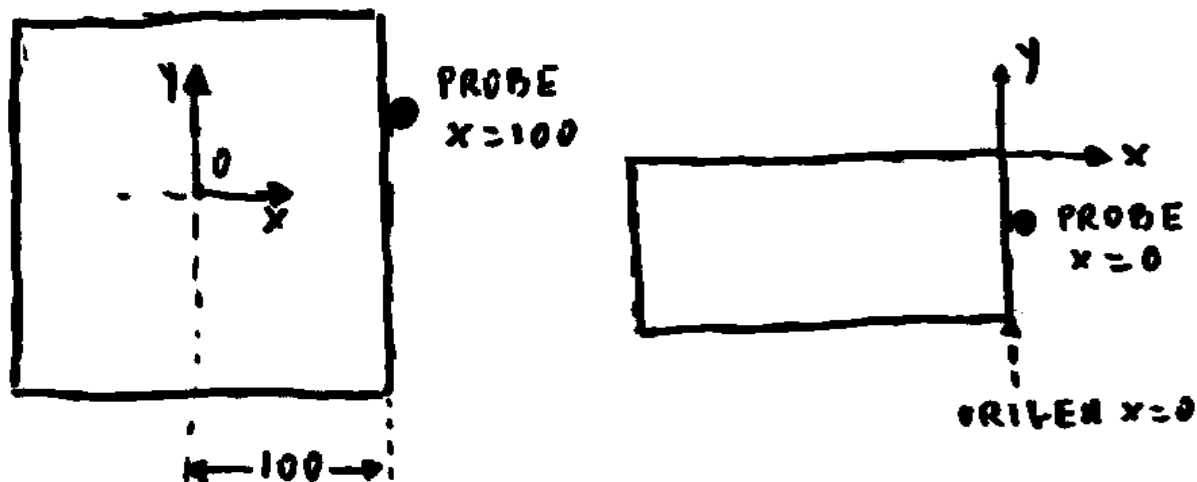
- X Y Z; qualquer valor.
- O valor dos argumentos I e J não significa muito, porque eles não representam um ponto de medida, mas a média de dois pontos sondados. Este valor não será verificado pela macro e assume o padrão igual a 0. Escrever I100, I-2, I23, não será, portanto, um erro pois o programa não o considera.
- O mesmo vale para C, o padrão assume o valor igual a 0.
- Os valores válido para W são 54 até 59.

O que quer dizer G120 X100?

G120 X100 significa que estamos sondando um ponto localizado a 100 mm a direita da origem de trabalho (origem que estamos a calcular), quer dizer a origem deve ser colocada 100 milímetros para a esquerda do ponto X.

O que quer dizer G120 X0?

G120 X0 quer dizer que o ponto sondado corresponde ao ponto $X=0$ da nossa origem.



Outra consideração é que, basicamente temos dois tipos de sondagem: plano, e eixo Z.

Para não ser muito complicado e por razões que explicarei mais tarde, escreveremos um programa em uma forma linear (ou seja, sem o uso de desvios ou ciclos repetitivos) a partir das considerações feitas.

Elenco dos procedimentos a ser escritos

```
IF[#24NE#0] GOTO10 (PROBE X)
IF[#25NE#0] GOTO20 (PROBE Y)
IF[#4NE#0] GOTO40 (PROBE I)
IF[#5NE#0] GOTO50 (PROBE J)
IF[#3NE#0] GOTO70 (PROBE C)
IF[#26NE#0] GOTO100 (PROBE Z)
#3000=3 (FALTA ARGUMENTO)
```

Monitoramos a presença dos argumentos, a partir de X até C, na ordem escrita. Se nenhum dos argumentos está presente, o programa irá gerar um alarme.

Escrever G120 W não tem sentido porque você quer criar uma origem sem fazer uma ação de sondagem.

A escolha desta ordem não é aleatória, mas gerado pelas seguintes considerações: a Z é colocada na parte inferior, porque é um tipo de sondagem diferente e, portanto, não necessita de comparação com as outras.

Verificar a X ou a Y como primeiro argumento é irrelevante, pelo contrário não é irrelevante X com I. Quero dizer o procedimento para calcular o ponto médio exige conhecer como calcular um ponto. Então, devemos primeiro considerar o ponto (X) e depois o valor médio. Então nós podemos escrever

```
IF[#25NE#0] GOTO20 (PROBE Y)
IF[#24NE#0] GOTO10 (PROBE X)
IF[#5NE#0] GOTO50 (PROBE J)
```

```

IF[#4NE#0] GOTO40 (PROBE I)
IF[#3NE#0] GOTO70 (PROBE C)
IF[#26NE#0] GOTO100 (PROBE Z)
#3000=3 (FALTA ARGUMENTO)

```

sem alterar o conceito expresso: X e Y em primeiro lugar e, em seguida, I e J.

Vamos prosseguir com a escritura dos procedimentos individuais listados aqui novamente.

```

IF[#24NE#0] GOTO10 (PROBE X)
IF[#25NE#0] GOTO20 (PROBE Y)
IF[#4NE#0] GOTO40 (PROBE I)
IF[#5NE#0] GOTO50 (PROBE J)
IF[#3NE#0] GOTO70 (PROBE C)
IF[#26NE#0] GOTO100 (PROBE Z)

```

Vamos vê-los um por um

N10 (PROBE X)

Antes de realizar o programa vamos fazer uma análise sobre a presença de outros argumentos.

- Y pode estar presente, o que indica uma dupla sondagem em X e Y. Neste caso, devemos desviar para o procedimento XY: sondagem de dois pontos (que escrevemos).
- Não é aceitável a presença de I ou C, pois seria uma escrita errada.
- J pode estar presente, temos que desviar para o procedimento XJ (que escrevemos).
- Z pode estar presente o que indica uma sondagem também em Z, desviar para sondagem Z

O programa deve ser escrito, considerando estes aspectos, pois

```

IF[#25NE#0] GOTO30 (PROBE XY)
IF[#4NE#0] THEN #3000=4 (DEMAIS ARGUMENTOS)
IF[#5NE#0] THEN GOTO80 (PROBE XJ)
IF[#3NE#0] THEN #3000=4 (DEMAIS ARGUMENTOS)
-----
aqui o procedimento "probe X"
-----
IF[#26NE#0] GOTO100 (PROBE Z)
GOTO110 (FINAL)

```

A última linha desvia para N110: podemos escrever directamente M99 (saída), mas não sabendo ainda qual ação tomaremos no final, é conveniente usar o comando GOTO, que nos permite variar a saída do programa, agendo uma só vez na linha N110.

IF[#25NE#0] GOTO30 (PROBE XY)
significa executar um procedimento especial para calcular simultaneamente X e Y

```

IF[#4NE#0] THEN #3000=4 (DEMAIS ARGUMENTOS)
IF[#3NE#0] THEN #3000=4 (DEMAIS ARGUMENTOS)

```

Se houver #4 e #3, que indica a presença de I ou C, vai ter um alarme que expressa o elevado número de argumentos pois o cnc não sabe o que sondar: X? I? C?

IF[#5NE#0] THEN GOTO80 (PROBE XJ) Se houver a #5 será necessário desviar.

Tendo analisado a presença de outros argumentos podemos prosseguir.

-Visualizar o tipo de sondagem
#3006=1 (PROBE PONTO X -> START)

A #3006 é uma variável mensagem, o programa fica em pausa.
É semelhante a #3000 enquanto visualiza um alarme mas não para o programa.
E' diferente de M0 pois chama a atenção do operador através da luz de alarme externa; por isso é claramente visível.

-Cálculo origem e inserção em tabela
#[5201+[#23*20]]=#5021+#24

As #5021(22/23) representam em coordenadas máquina a posição atual (X/Y/Z), enquanto as variáveis #5200->53.. representam os offset, quer dizer as origens dos sistemas de referência.

#5221 eixo X origem G54
#5222 eixo y origem G54
#5223 eixo Z origem G54

#5241 eixo X origem G55
#5242 eixo y origem G55
#5243 eixo Z origem G55

continuando assim por múltiplos de 20

#5261 eixo X origem G56

#5281 eixo X origem G57

#5301 eixo X origem G58

#5321 eixo X origem G59

Para calcular o valor correto de origem, usamos a #23 já analisada, que nos oferece uma forma eficaz de conhecê-la.

Anteriormente vimos que o valor de #23(1-6) de forma indireta indica como queremos definir a origem

#23=1 G54	#23=2 G55	#23=3 G56
#23=4 G57	#23=5 G58	#23=6 G59

Sabemos também que as variáveis relacionadas às origens estão dispostas em múltiplos de 20. Dito isto multiplicando o valor (#23*20) temos a referência correta para a origem (como um número) que iremos utilizar na definição do parâmetro.

Exemplo

queremos criar a origem G57.

G57 está relacionado com 4 (#23=57-53), #23=#23*20=80, que somado ao valor 5201 (o primeiro valor de referência para o X, para offsets) de 5281.

Esta é a variável a ser definida em relação ao eixo X.

O mesmo é verdadeiro para Y, ou Z, respectivamente, tomando como referência os números

Y->5202 Z->5203

A presença de #24 na linha #[5201+[#23*20]]=#5021+#24 indica que você pode sondar um ponto qualquer. Então, ao valor atual da posição da máquina será adicionado o valor de #24. Isto permite criar sondagem a X=0 ou qualquer valor

inserido no argumento.

-Verificar a presença da sonda Z

IF[#26NE#0] GOTO80 (PROBE Z), se for necessário desviamos para o probe Z....

-Ou terminar a sondagem e acabar o procedimento de sondagem.

GOTO110 (FINAL)

Aqui o probe X

```
N10 (PROBE X)
IF[#25NE#0] GOTO30 (PROBE XY)
IF[#4NE#0] THEN #3000=4 (DEMAIS ARGUMENTOS)
IF[#5NE#0] THEN GOTO80 (PROBE XJ)
IF[#3NE#0] THEN #3000=4 (DEMAIS ARGUMENTOS)
#3006=1 (PROBE PONTO X -> START)
#[5201+[#23*20]]=#5021+#24
IF[#26NE#0] GOTO100 (PROBE Z)
GOTO110 (FINAL)
```

N20 (PROBE Y)

Este procedimento segue os mesmos conceitos anteriormente vistos
(não vou repetir-me)

```
N20 (PROBE Y)
IF[#4NE#0]GOTO90 (PROBE YI)
IF[#5NE#0]THEN #3000=4 (DEMAIS ARGUMENTOS)
IF[#3NE#0]THEN #3000=4 (DEMAIS ARGUMENTOS)
#3006=2 (PROBE PONTO Y -> START)
#[5202+[#1*20]]=#5022+#25
IF[#26NE#0] GOTO100 (PROBE Z)
GOTO110 (FINAL)
```

Antes de considerar a sondagem I vamos examinar os probes XY e Z e também o encerramento do programa.

N30 (PROBE XY)

Este procedimento permite a sondagem de dois pontos, X e Y em conjunto.
O princípio é o mesmo dos procedimentos vistos aplicado aos dois eixos.

```
N30 (PROBE XY)
IF[#4NE#0]THEN #3000=4 (DEMAIS ARGUMENTOS)
IF[#5NE#0]THEN #3000=4 (DEMAIS ARGUMENTOS)
IF[#7NE#0]THEN #3000=4 (DEMAIS ARGUMENTOS)
#3006=1 (PROBE PONTO X ->START)
#[5201+[#23*20]]=#5021+#24
#3006=2 (PROBE PONTO Y ->START)
#[5202+[#1*20]]=#5022+#25
IF[#26NE#0]GOTO100 (PROBE Z)
GOTO110 (FINAL)
```

N100 (PROBE Z)

```
N100 (PROBE Z)
#3006=3 (PROBE PONTO Z ->START)
```

```
#[5203+[20*#1]]=#5023-160.25-#26  
GOTO110 (FINAL)
```

E' preciso esclarecer!

O probe Z é idêntico aos outros procedimentos com a adição de uma constante numérica (160,25) representando o comprimento da sonda. Você tem que pensar que quando você pressiona o botão START o valor do eixo Z é condicionado pelo comprimento da sonda que deve ser subtraído para obter o valor correto.

Eu adicionei um valor constante pois normalmente usa-se uma única sonda.

E' o uso mais comum, e certamente o mais fácil de implementar.

(Se você usa diferentes sondas deve variar o procedimento.)

Existem diferentes estratégias se usarmos diferentes sondas, ou para alterar o comprimento, ou devido a ruptura da mesma.

Você pode usar variáveis estáticas, usar valores de compensação de comprimento ou utilizar o G10.

Cada estratégia tem prós e contras, você deve avaliá-las cuidadosamente de acordo com sua própria realidade e finalidade!

N110 (FINAL)

O procedimento de fechamento é representado por M99.

No momento, não digo mais nada.

N110 M99 (FINAL)

Bem, um pequeno resumo, para não perder o fio do discurso.

```
O9018 (G54-G59 MACRO ORIGEM)  
(USO G120 X Y Z I J W C)  
(X PONTO X)  
(Y PONTO Y)  
(Z PONTO Z)  
(I MÉDIA DOS PONTO X)  
(J MÉDIA DOS PONTO Y)  
(C CENTRO CÍRCULO)  
(W ORIGEM 54->59)  
IF[#23NE#0]THEN #3000=1 (FALTA A ORIGEM W)  
IF[#23LT54]THEN #3000=2 (ORIGEM ERRADA)  
IF[#23GT59]THEN #3000=2 (ORIGEM ERRADA)  
#23=#23-53  
T0M6 (SONDA)  
G90G0G40G49G53X500Y-300  
G90G0G53Z-200  
IF[#24NE#0]GOTO10 (PROBE X)  
IF[#25NE#0]GOTO20 (PROBE Y)  
IF[#4NE#0]GOTO40 (PROBE I)  
IF[#5NE#0]GOTO50 (PROBE J)  
IF[#3NE#0]GOTO70 (PROBE C)  
IF[#26NE#0]GOTO100 (PROBE Z)  
#3000=3 (FALTA ARGUMENTO)  
N10 (PROBE X)  
IF[#25NE#0]GOTO30 (PROBE XY)  
IF[#4NE#0]THEN #3000=4 (DEMAIS ARGUMENTOS)  
IF[#5NE#0]THEN GOTO80 (PROBE XJ)  
IF[#3NE#0]THEN #3000=4 (DEMAIS ARGUMENTOS)  
#3006=1 (PROBE PONTO X -> START)  
#[5201+[#23*20]]=#5021+#24  
IF[#26NE#0] GOTO100 (PROBE Z)  
GOTO110 (FINAL)  
N20 (PROBE Y)  
IF[#4NE#0]GOTO90 (PROBE YI)
```



```

IF[#5NE#0] THEN #3000=4 (DEMAIS ARGUMENTOS)
IF[#3NE#0] THEN #3000=4 (DEMAIS ARGUMENTOS)
#3006=2 (PROBE PONTO Y -> START)
#[5202+[#1*20]]=#5022+#25
IF[#26NE#0] GOTO100 (PROBE Z)
GOTO110 (FINAL)
N30 (PROBE XY)
IF[#4NE#0] THEN #3000=4 (DEMAIS ARGUMENTOS)
IF[#5NE#0] THEN #3000=4 (DEMAIS ARGUMENTOS)
IF[#7NE#0] THEN #3000=4 (DEMAIS ARGUMENTOS)
#3006=1 (PROBE PONTO X -> START)
#[5201+[#23*20]]=#5021+#24
#3006=2 (PROBE PONTO Y -> START)
#[5202+[#1*20]]=#5022+#25
IF[#26NE#0] GOTO100 (PROBE Z)
GOTO110 (FINAL)
N40 (PROBE I)          a ser escrito
N50 (PROBE J)          a ser escrito
N60 (PROBE IJ)         a ser escrito
N70 (PROBE C)          a ser escrito
N80 (PROBE XJ)         a ser escrito
N90 (PROBE YI)         a ser escrito
N100 (PROBE Z)
#3006=3 (PROBE PONTO Z -> START)
#[5203+[20*#1]]=#5023-160.25-#26
N110 M99 (FINAL)

```

Os procedimentos descritos executam qualquer tipo de sondagem. Descrito pela primeira vez os outros são uma repetição do mesmo. O probe I, valor médio de X, como a sondagem J, representam o valor médio de duas sondagem (com uma precisão de 0,001 milímetros ou 0,0001 polegada). Praticamente se sonda um ponto, em seguida um segundo, e o programa irá calcular o valor médio e colocá-lo na posição correta.

Mas eu quero (para fins acadêmicos) introduzir o controle de movimento durante a sondagem do eixo que não deve mudar de posição. Se você quer sondar o ponto médio no eixo X é natural considerar o eixo Y parado, não deve mudar de posição. E' isto que iremos controlar.

N40 (PROBE I)

```

N40 (PROBE I)
IF[#5NE#0] GOTO60 (PROBE IJ)
IF[#24NE#0] THEN #3000=4 (DEMAIS ARGUMENTOS)
IF[#25NE#0] GOTO90 (PROBE YI)
IF[#3NE#0] THEN #3000=4 (DEMAIS ARGUMENTOS)
#3006=1 (PROBE PRIMEIRO PONTO X -> START)
#7=#5021 (1° POSICAO X)
#17=#5022 (1° POSICAO Y)
#3006=1 (PROBE SEGUNDO PONTO X -> START)
#18=#5022 (2° POSICAO Y)
IF[ABS[#18-#17]GT0.01] THEN #3000=10 (MOVIMENTO EM Y)
#8=#5021 (2° POSICAO X)
#7=[#7+#8]/2 (VALOR MEDIO X)
#[5201+[#1*20]]=#7 (INSERCAO)
IF[#26NE#0] GOTO100 (PROBE Z)
GOTO110 (FINAL)

```

#7 e #17 representam os valores de X e Y do primeiro ponto.

#8 e #18 representam os valores do segundo ponto.

É claro que #17 e #18 devem ficar idêntico, de outra forma, houve um movimento no eixo. O controle é realizado considerando-se 0,01 milímetros de tolerância.

IF [ABS[#18-#17]GT0.01] se o valor absoluto da diferença entre #18 e #17 é maior do que 0,01 (milímetros), em seguida, um alarme será gerado. Claro, o procedimento de controle foi realizado para fins acadêmicos, mas eu acho que é um bom procedimento e não deve ser negligenciado.

A primeira linha IF[#5NE#0]GOTO60 verifica a existência de #5 no caso significa que o programa foi chamado com os argumentos I e J, então, nós temos que realizar o procedimento IJ.

O raciocínio é repetido também para J

N50 (PROBE J)

```
N50 (PROBE J)
IF[#24NE#0]GOTO80 (PROBE XJ)
IF[#25NE#0]THEN #3000=4 (DEMAIS ARGUMENTOS)
IF[#3NE#0]THEN #3000=4 (DEMAIS ARGUMENTOS)
#3006=2 (PROBE PRIMEIRO PONTO Y ->START)
#9=#5022 (1° POSICAO Y)
#19=#5021 (1° POSICAO X)
#3006=2 (PROBE SEGUNDO PONTO Y ->START)
#20=#5021 (2° POSICAO X)
IF[ABS[#20-#19]GT0.01]THEN #3000=10 (MOVIMENTO EM X)
#10=#5021 (2° POSICAO Y)
#9=[#9+#10]/2 (VALOR MEDIO Y)
#[5202+[#1*20]]=#9 (INSERCAO)
IF[#26NE#0]GOTO100 (PROBE Z)
GOTO110 (FINAL)
```

OS procedimentos IJ, XJ, e YI, são nada mais do que repetições dos termos que já sabemos por isso não vou me repetir.

N60 (PROBE IJ)

```
N60 (PROBE IJ)
IF[#24NE#0]THEN #3000=4 (DEMAIS ARGUMENTOS)
IF[#25NE#0]THEN #3000=4 (DEMAIS ARGUMENTOS)
IF[#7NE#0]THEN #3000=4 (DEMAIS ARGUMENTOS)
#3006=1 (PROBE PRIMEIRO PONTO X ->START)
#7=#5021
#17=#5022
#3006=1 (PROBE SEGUNDO PONTO X ->START)
#18=#5022
IF[ABS[#18-#17]GT0.01]THEN #3000=10 (MOVIMENTO EM Y)
#8=#5021
#7=[#7+#8]/2
#[5201+[#1*20]]=#7
#3006=2 (PROBE PRIMEIRO PONTO Y ->START)
#9=#5022
#19=#5021
#3006=2 (PROBE SEGUNDO PONTO Y ->START)
#20=#5021
IF[ABS[#20-#19]GT0.01]THEN #3000=10 (MOVIMENTO EM X)
#10=#5022
#9=[#9+#10]/2
#[5202+[#1*20]]=#9
IF[#26NE#0]GOTO100
GOTO110
```

N80 (PROBE XJ)

```
N80 (PROBE XJ)
#3006=1 (PROBE PONTO X ->START)
```

```

# [5201+[#1*20]]=#5021+#24
#3006=2 (PROBE PRIMEIRO PONTO Y ->START)
#9=#5022
#19=#5021
#3006=2 (PROBE SEGUNDO PONTO Y ->START)
#20=#5021
IF[ABS[#20-#19]GT0.01]THEN #3000=10 (MOVIMENTO EM X)
#10=#5022
#9=[#9+#10]/2
#[5202+[#1*20]]=#9
IF[#26NE#0]GOTO100
GOTO110

```

N90 (PROBE YI)

```

N90 (PROBE YI)
#3006=2 (PROBE PONTO Y ->START)
#[5202+[#1*20]]=#5022+#25
#3006=1 (PROBE PRIMEIRO PONTO X ->START)
#7=#5021
#17=#5022
#3006=1 (PROBE SEGUNDO PONTO X ->START)
#18=#5022
IF[ABS[#18-#17]GT0.01]THEN #3000=10 (MOVIMENTO EM Y)
#8=#5021
#7=[#7+#8]/2
#[5201+[#1*20]]=#7
IF[#26NE#0]GOTO100
GOTO110

```

Agora temos que escrever o procedimento C

N70 (Procedimento C)

O procedimento requer um momento de explicação. Estamos executando um processo manual; não temos conhecimento de qualquer desvio da sonda ao longo do X e Y e, portanto, a nossa macro não pode considerar estes factores (a menos que você não os conhece ou calcula). Por esta razão consideramos que ocorre um erro no cálculo do valor do centro da circunferência. Este erro pode ser eliminado usando um comparador localizado no centro. Devemos, portanto, dizer que o procedimento nos dá um método bruto IJ (estamos falando de centésimos de erro) para o cálculo do centro de um círculo, enquanto o valor exato se deve fazer utilizando um comparador. Dito isto, o procedimento mostrou-se idêntico ao probe XY. Por conveniência vamos novamente escreve-lo.

N70 (PROBE C)

```

N70 (PROCEDURA C)
IF[#24NE#0]THEN #3000=4 (DEMAIS ARGUMENTOS)
IF[#25NE#0]THEN #3000=4 (DEMAIS ARGUMENTOS)
IF[#4NE#0]THEN #3000=4 (DEMAIS ARGUMENTOS)
IF[#5NE#0]THEN #3000=4 (DEMAIS ARGUMENTOS)
#3006=3 (COMPARADOR AO CENTRO CIRCULO ->START)
#[5201+[#1*20]]=#5021
#[5202+[#1*20]]=#5022
IF[#26NE#0]GOTO100
GOTO110

```

Bem, a nossa macro parece completa: aqui está , sem comentários!
 Certifique-se de guardar uma cópia da macro contendo todos os comentários possíveis ... depois de algum tempo será difícil lembrar cada passo ... estes serão de muita ajuda. O controle usará o file sem comentários.

```

09018 (G54-G59 MACRO ORIGEM)
(USO G120 X Y Z I J W C)
(X PROBE PONTO X)
(Y PROBE PONTO Y)
(Z PROBE PONTO Z)
(I PROBE MÉDIA DOS PONTO EM X)
(J PROBE MÉDIA DOS PONTO EM Y)
(C PROBE CENTRO DE CIRCUNFERÊNCIA)
(W ORIGEM 54->59)
IF[#23NE#0] THEN #3000=1 (FALTA A ORIGEM W)
IF[#23LT54] THEN #3000=2 (ORIGEM ERRADA)
IF[#23GT59] THEN #3000=2 (ORIGEM ERRADA)
#23=#23-53
T0M6 (PROBE)
G90G0G40G49G53X500Y-300
G90G0G53Z-200
IF[#24NE#0] GOTO10
IF[#25NE#0] GOTO20
IF[#4NE#0] GOTO40
IF[#5NE#0] GOTO50
IF[#3NE#0] GOTO70
IF[#26NE#0] GOTO100
#3000=3 (FALTA ARGUMENTO)
N10 (PROBE X)
IF[#25NE#0] GOTO30
IF[#4NE#0] THEN #3000=4 (DEMAIS ARGUMENTOS)
IF[#5NE#0] THEN GOTO80
IF[#3NE#0] THEN #3000=4 (DEMAIS ARGUMENTOS)
#3006=1 (PROBE PONTO X -> START)
#[5201+[#23*20]]=#5021+#24
IF[#26NE#0] GOTO100
GOTO110
N20 (PROBE Y)
IF[#4NE#0] GOTO90
IF[#5NE#0] THEN #3000=4 (DEMAIS ARGUMENTOS)
IF[#3NE#0] THEN #3000=4 (DEMAIS ARGUMENTOS)
#3006=2 (PROBE PONTO Y -> START)
#[5202+[#1*20]]=#5022+#25
IF[#26NE#0] GOTO100
GOTO110
N30 (PROBE XY)
IF[#4NE#0] THEN #3000=4 (DEMAIS ARGUMENTOS)
IF[#5NE#0] THEN #3000=4 (DEMAIS ARGUMENTOS)
IF[#7NE#0] THEN #3000=4 (DEMAIS ARGUMENTOS)
#3006=1 (PROBE PONTO X -> START)
#[5201+[#23*20]]=#5021+#24
#3006=2 (PROBE PONTO Y -> START)
#[5202+[#1*20]]=#5022+#25
IF[#26NE#0] GOTO100
GOTO110
N40 (PROBE I)
IF[#5NE#0] GOTO60 (PROBE IJ)
IF[#24NE#0] THEN #3000=4 (DEMAIS ARGUMENTOS)
IF[#25NE#0] GOTO90
IF[#3NE#0] THEN #3000=4 (DEMAIS ARGUMENTOS)
#3006=1 (PROBE PRIMEIRO PONTO X -> START)
#7=#5021
#17=#5022
#3006=1 (PROBE SEGUNDO PONTO X -> START)
#18=#5022
IF[ABS[#18-#17]GT0.01] THEN #3000=10 (MOVIMENTO EM Y)
#8=#5021
#7=[#7+#8]/2
#[5201+[#1*20]]=#7

```

```

IF[#26NE#0]GOTO100
GOTO110
N50 (PROBE J)
IF[#24NE#0]GOTO80
IF[#25NE#0]THEN #3000=4 (DEMAIS ARGUMENTOS)
IF[#3NE#0]THEN #3000=4 (DEMAIS ARGUMENTOS)
#3006=2 (PROBE PRIMEIRO PONTO Y ->START)
#9=#5022
#19=#5021
#3006=2 (PROBE SEGUNDO PONTO Y ->START)
#20=#5021
IF[ABS[#20-#19]GT0.01]THEN #3000=10 (MOVIMENTO EM X)
#10=#5021
#9=[#9+#10]/2
#[5202+[#1*20]]=#9
IF[#26NE#0]GOTO100
GOTO110
N60 (PROBE IJ)
IF[#24NE#0]THEN #3000=4 (DEMAIS ARGUMENTOS)
IF[#25NE#0]THEN #3000=4 (DEMAIS ARGUMENTOS)
IF[#7NE#0]THEN #3000=4 (DEMAIS ARGUMENTOS)
#3006=1 (PROBE PRIMEIRO PONTO X ->START)
#7=#5021
#17=#5022
#3006=1 (PROBE SEGUNDO PONTO X ->START)
#18=#5022
IF[ABS[#18-#17]GT0.01]THEN #3000=10 (MOVIMENTO EM Y)
#8=#5021
#7=[#7+#8]/2
#[5201+[#1*20]]=#7
#3006=2 (PROBE PRIMEIRO PONTO Y ->START)
#9=#5022
#19=#5021
#3006=2 (PROBE SEGUNDO PONTO Y ->START)
#20=#5021
IF[ABS[#20-#19]GT0.01]THEN #3000=10 (MOVIMENTO EM X)
#10=#5022
#9=[#9+#10]/2
#[5202+[#1*20]]=#9
IF[#26NE#0]GOTO100
GOTO110
N70 (PROBE C)
IF[#24NE#0]THEN #3000=4 (DEMAIS ARGUMENTOS)
IF[#25NE#0]THEN #3000=4 (DEMAIS ARGUMENTOS)
IF[#4NE#0]THEN #3000=4 (DEMAIS ARGUMENTOS)
IF[#5NE#0]THEN #3000=4 (DEMAIS ARGUMENTOS)
#3006=3 (COMPARATOR CENTRO CIRCULO ->START)
#[5201+[#1*20]]=#5021
#[5202+[#1*20]]=#5022
IF[#26NE#0]GOTO100
GOTO110
N80 (PROBE XJ)
#3006=1 (PROBE PONTO X ->START)
#[5201+[#1*20]]=#5021+#24
#3006=2 (PROBE PRIMEIRO PONTO Y ->START)
#9=#5022
#19=#5021
#3006=2 (PROBE SEGUNDO PONTO Y ->START)
#20=#5021
IF[ABS[#20-#19]GT0.01]THEN #3000=10 (MOVIMENTO EM X)
#10=#5022
#9=[#9+#10]/2
#[5202+[#1*20]]=#9
IF[#26NE#0]GOTO100

```

```

GOTO110
N90 (PROBE YI)
#3006=2 (PROBE PONTO Y ->START)
#[5202+[#1*20]]=#5022+#25
#3006=1 (PROBE PRIMEIRO PONTO X ->START)
#7=#5021
#17=#5022
#3006=1 (PROBE SEGUNDO PONTO X ->START)
#18=#5022
IF[ABS[#18-#17]GT0.01]THEN #3000=10 (MOVIMENTO EM Y)
#8=#5021
#7=[#7+#8]/2
#[5201+[#1*20]]=#7
IF[#26NE#0]GOTO100
GOTO110
N100 (PROBE Z)
#3006=3 (PROBE PONTO Z ->START)
#[5203+[20*#1]]=#5023-160.25-#26
N110 M99

```

SEGUNDA PARTE: REFLEXOES!

O que posso dizer?

A macro foi experimentada: funciona!

Repito-me! O programa macro não está otimizado, não ha ciclos repetitivos ou desvios. Porque? A razão veremos em um momento.

O problema que aflige muitas pessoas que querem usar uma macro é aquele que às vezes não funciona, independentemente de quem fez as declarações em contrário: mas por quê?

As razões são múltiplas; é disto que vamos falar.

Vamos tentar usar o programa em um contexto real, onde irão apresentar-se diversos problemas que tentaremos resolver. Isso deve nos ajudar a entender algumas atitudes para contemplar quando se criam programas macro.

Nós testamos a macro usando um controle Fanuc Series 21. Após as correções necessárias pode-se dizer que o programa executa a sua tarefa admiravelmente.

Para testar a macro è melhor modificar o número de programa (09018->06000) e escrever o seguinte programa

```
G65 P6000 X10 J0 Z6 W56
```

(alterando o número não precisa desabilitar a segurança dos programas 08000->09999. Quando testado podemos renomear novamente).

Claro que você tem que testar cada tipo de sondagem.

Suponha-se que tudo corra bem.

Gostaríamos de saber se a macro pode ser executada em um controle diferente Vamos tentar!

Testamos a macro em um controle Fanuc 16.

Aqui também não há problemas.

Nossa loja tem um controle de idade (OMD) nós queremos saber se o programa pode trabalhar. E' só tentar!

Aqui também temos que testar todas as variações possíveis de comandos começando com o mais simples

```
G65 P6000 X0 W55 até o mais complexo G65 I0 J0 Z12 W58
```

Resultado?

Eu vou inspirar-me numa situação real apresentada a mim por um de vocês!

Aparece um primeiro alarme: ALARME: FRASE MACRO NÃO RECONHECIDA!

O CNC não entende o que você escreveu!

Para entender o que está errado, você tem que executar o programa em bloco único.

O alarme se encontra início do programa

```
IF[#23NE#0]THEN #3000=1(FALTA ORIGEM)
```

é claro que o culpado é a declaração THEN, a qual não é reconhecida pelo cnc. Quando você não sabe o que fazer, ou quer experimentar, pode executar os comando em MDI e verificar o comportamento.

Por exemplo, se escrevermos #23=1 e esse valor é colocado na #23, podemos dizer que não há problemas e que o controle tem variáveis ativas: pois o erro está produzido pelo THEN. Se você não está convencido pode escrever (sempre MDI) #3000=1(FALTA ORIGEM) e verificar se o controle entra em estado de alarme exibindo a frase "FALTA ORIGEM".

O que fazer?

Como devemos nos comportar, se o CNC não reconhece a frase THEN?

Isto significa que a frase THEN não pode ser usada e por isso temos de encontrar alternativas. (O CNC é antigo, por isso carece de algumas frases macro).

A primeira coisa a fazer é armar-se de manual macro e ver quais os comandos que podemos usar.

As estradas que podemos tomar são diferente: assim avaliem muito bem!

Após uma primeira análise, decidimos usar no lugar da fase THEN, o GOTO na seguinte forma

```
IF[#23NE#0]THEN #3000=1(FALTA ORIGEM) -->> IF[#23NE#0] GOTO 950
```

Temos que substituir cada THEN com o GOTO que desvia para o fim do programa, onde teremos uma lista de alarmes.

Vamos ver!

As linhas azuis substituíram as antigas linhas vermelhas

```
O9018(MACRO ORIGEM G54-G59)
```

```
IF[#23NE#0] GOTO 950          THEN #3000=1(FALTA ORIGEM)
IF[#23LT54] GOTO 940          THEN #3000=2(ORIGEM ERRADA)
IF[#23GT59] GOTO 940          THEN #3000=2(ORIGEM ERRADA)
#23=#23-53
T0M6
G90G0G40G49G53X500Y-300
G90G0G53Z-200
IF[#24NE#0] GOTO10
IF[#25NE#0] GOTO20
IF[#4NE#0] GOTO40
IF[#5NE#0] GOTO50
IF[#3NE#0] GOTO70
IF[#26NE#0] GOTO100
#3000=3(FALTA ARGUMENTO)
N10IF[#25NE#0] GOTO30
IF[#4NE#0] GOTO 920          THEN #3000=4(DEMAIS ARGUMENTOS)
IF[#5NE#0] GOTO80
IF[#3NE#0] GOTO 920          THEN #3000=4(DEMAIS ARGUMENTOS)
#3006=1(PROBE PONTO X ->START)
#[5201+[#23*20]]=#5021+#24
IF[#26NE#0]GOTO100
GOTO110
N20IF[#4NE#0] GOTO90
IF[#5NE#0] GOTO 920          THEN #3000=4(DEMAIS ARGUMENTOS)
IF[#3NE#0] GOTO 920          THEN #3000=4(DEMAIS ARGUMENTOS)
```

```

#3006=2 (PROBE PONTO Y ->START)
#[5202+[#1*20]]=#5022+#25
IF[#26NE#0] GOTO100
GOTO110
N30IF[#4NE#0] GOTO 920 THEN #3000=4 (DEMAIS ARGUMENTOS)
IF[#5NE#0] GOTO 920 THEN #3000=4 (DEMAIS ARGUMENTOS)
IF[#7NE#0] GOTO 920 THEN #3000=4 (DEMAIS ARGUMENTOS)
#3006=1 (PROBE PONTO X ->START)
#[5201+[#23*20]]=#5021+#24
#3006=2 (PROBE PONTO Y ->START)
#[5202+[#1*20]]=#5022+#25
IF[#26NE#0] GOTO100
GOTO110
N40IF[#5NE#0] GOTO60
IF[#24NE#0] GOTO 920 THEN #3000=4 (DEMAIS ARGUMENTOS)
IF[#25NE#0] GOTO90
IF[#3NE#0] GOTO 920 THEN #3000=4 (DEMAIS ARGUMENTOS)
#3006=1 (PROBE PRIMEIRO PONTO X ->START)
#7=#5021
#17=#5022
#3006=1 (PROBE SEGUNDO PONTO X ->START)
#18=#5022
IF[ABS[#18-#17]GT0.01]GOTO 910 THEN #3000=10 (MOVIMENTO EM Y)
#8=#5021
#7=[#7+#8]/2
#[5201+[#1*20]]=#7
IF[#26NE#0] GOTO100
GOTO110
N50IF[#24NE#0] GOTO80
IF[#25NE#0] GOTO 920 THEN #3000=4 (DEMAIS ARGUMENTOS)
IF[#3NE#0] GOTO 920 THEN #3000=4 (DEMAIS ARGUMENTOS)
#3006=2 (PROBE PRIMEIRO PONTO Y ->START)
#9=#5022
#19=#5021
#3006=2 (PROBE SEGUNDO PONTO Y ->START)
#20=#5021
IF[ABS[#20-#19]GT0.01]GOTO 900 THEN #3000=10 (MOVIMENTO EM X)
#10=#5021
#9=[#9+#10]/2
#[5202+[#1*20]]=#9
IF[#26NE#0] GOTO100
GOTO110
N60IF[#24NE#0] GOTO 920 THEN #3000=4 (DEMAIS ARGUMENTOS)
IF[#25NE#0] GOTO 920 THEN #3000=4 (DEMAIS ARGUMENTOS)
IF[#7NE#0] GOTO 920 THEN #3000=4 (DEMAIS ARGUMENTOS)
#3006=1 (PROBE PRIMEIRO PONTO X ->START)
#7=#5021
#17=#5022
#3006=1 (PROBE SEGUNDO PONTO X ->START)
#18=#5022
IF[ABS[#18-#17]GT0.01] GOTO 910 THEN #3000=10 (MOVIMENTO EM Y)
#8=#5021
#7=[#7+#8]/2
#[5201+[#1*20]]=#7
#3006=2 (PROBE PRIMEIRO PONTO Y ->START)
#9=#5022
#19=#5021
#3006=2 (PROBE SEGUNDO PONTO Y ->START)
#20=#5021
IF[ABS[#20-#19]GT0.01] GOTO 900 THEN #3000=10 (MOVIMENTO EM X)
#10=#5022
#9=[#9+#10]/2
#[5202+[#1*20]]=#9
IF[#26NE#0] GOTO100

```



```

GOTO110
N70IF[#24NE#0] GOTO 920 THEN #3000=4(DEMAIS ARGUMENTOS)
IF[#25NE#0] GOTO 920 THEN #3000=4(DEMAIS ARGUMENTOS)
IF[#4NE#0] GOTO 920 THEN #3000=4(DEMAIS ARGUMENTOS)
IF[#5NE#0] GOTO 920 THEN #3000=4(DEMAIS ARGUMENTOS)
#3006=3(COMPADOR NO CENTRO ->START)
#[5201+[#1*20]]=#5021
#[5202+[#1*20]]=#5022
IF[#26NE#0] GOTO100
GOTO110
N80#3006=1(PROBE PONTO X ->START)
#[5201+[#1*20]]=#5021+#24
#3006=2(PROBE PRIMEIRO PONTO Y ->START)
#9=#5022
#19=#5021
#3006=2(PROBE SEGUNDO PONTO Y ->START)
#20=#5021
IF[ABS[#20-#19]GT0.01] GOTO 900 THEN #3000=10(MOVIMENTO EM X)
#10=#5022
#9=[#9+#10]/2
#[5202+[#1*20]]=#9
IF[#26NE#0] GOTO100
GOTO110
N90#3006=2(PROBE PONTO Y ->START)
#[5202+[#1*20]]=#5022+#25
#3006=1(PROBE PRIMEIRO PONTO X ->START)
#7=#5021
#17=#5022
#3006=1(PROBE SEGUNDO PONTO X ->START)
#18=#5022
IF[ABS[#18-#17]GT0.01] GOTO 910 THEN #3000=10(MOVIMENTO EM Y)
#8=#5021
#7=[#7+#8]/2
#[5201+[#1*20]]=#7
IF[#26NE#0] GOTO100
GOTO110
N100#3006=3(PROBE PONTO Z ->START)
#[5203+[20*#1]]=#5023-160.25-#26
N110 M99

```

nO final teremos

```

N900 #3000=6(MOVIMENTO EM X)
N910 #3000=5(MOVIMENTO EM Y)
N920 #3000=4(DEMAIS ARGUMENTOS)
N930 #3000=3(FALTA ARGUMENTOS)
N940 #3000=2(ORIGEM ERRADA)
N950 #3000=1(FALTA ORIGEM)

```

Alguns esclarecimentos.

Antes de escrever um programa macro é sempre bom perguntar-se para qual modelo de cnc pretendemos escrever. Sempre leia o manual para verificar o que você pode usar e o que não.

Agora percebemos que ter escrito uma macro em uma forma linear foi muito conveniente para mudar as linhas diferentes. Se tivéssemos usado loops e desvios condicionais provavelmente ficava mais complicado, não alterando algumas linhas, mas mudando todo o programa.

Corrigido o programa vamos prosseguir com a verificação.

G65 P6000 W55

Nossa!!! outro erro: VARIÁVEL NÃO RECONHECIDA!

A linha que dá problema é #3006=1(Probe X-> START)

Parece que este controle tem algo diferente dos controles que normalmente usamos, onde tudo era ativo só podemos remover #3006 e substituí-la por M0 (a #3000 não dá certo porque iria interromper o programa)

Prosseguimos com as mudanças

```
O9018(MACRO ORIGEM G54-G59)
IF[#23NE#0] GOTO 950
IF[#23LT54] GOTO 940
IF[#23GT59] GOTO 940
#23=#23-53
```

T0M6

G90G0G40G49G53X500Y-300

G90G0G53Z-200

IF[#24NE#0] GOTO10

IF[#25NE#0] GOTO20

IF[#4NE#0] GOTO40

IF[#5NE#0] GOTO50

IF[#3NE#0] GOTO70

IF[#26NE#0] GOTO100

#3000=3(FALTA ARGUMENTO)

N10IF[#25NE#0] GOTO30

IF[#4NE#0] GOTO 920

IF[#5NE#0] GOTO80

IF[#3NE#0] GOTO 920

M0(Probe Ponto X ->START)

#[5201+[#23*20]]=#5021+#24

IF[#26NE#0]GOTO100

GOTO110

N20IF[#4NE#0] GOTO90

IF[#5NE#0] GOTO 920

IF[#3NE#0] GOTO 920

M0(Probe Ponto Y ->START)

#[5202+[#1*20]]=#5022+#25

IF[#26NE#0] GOTO100

GOTO110

N30IF[#4NE#0] GOTO 920

IF[#5NE#0] GOTO 920

IF[#7NE#0] GOTO 920

M0(Probe Ponto X ->START)

#[5201+[#23*20]]=#5021+#24

M0(Probe Ponto Y ->START)

#[5202+[#1*20]]=#5022+#25

IF[#26NE#0] GOTO100

GOTO110

N40IF[#5NE#0] GOTO60

IF[#24NE#0] GOTO 920

IF[#25NE#0] GOTO90

IF[#3NE#0] GOTO 920

M0(Probe Primeiro Ponto X ->START)

#7=#5021

#17=#5022

M0(Probe Segundo Ponto X ->START)

#18=#5022

IF[ABS[#18-#17]GT0.01]GOTO 910

#8=#5021

#7=[#7+#8]/2

#[5201+[#1*20]]=#7

IF[#26NE#0] GOTO100

GOTO110

N50IF[#24NE#0] GOTO80

#3006=1(Probe Ponto X ->START)

#3006=2(Probe Ponto Y ->START)

#3006=1(Probe Ponto X ->START)

#3006=2(Probe Ponto Y ->START)

#3006=1(Probe Primeiro Ponto X ->START)

#3006=1(Probe Segundo Ponto X ->START)

IF[#25NE#0] GOTO 920	
IF[#3NE#0] GOTO 920	
M0 (PROBE PRIMEIRO PONTO Y ->START)	#3006=2 (PROBE PRIMEIRO PONTO Y ->START)
#9=#5022	
#19=#5021	
M0 (PROBE SEGUNDO PONTO Y ->START)	#3006=2 (PROBE SEGUNDO PONTO Y ->START)
#20=#5021	
IF[ABS[#20-#19]GT0.01]GOTO 900	
#10=#5021	
#9=[#9+#10]/2	
#[5202+[#1*20]]=#9	
IF[#26NE#0] GOTO100	
GOTO110	
N60IF[#24NE#0] GOTO 920	
IF[#25NE#0] GOTO 920	
IF[#7NE#0] GOTO 920	
M0 (PROBE PRIMEIRO PONTO X ->START)	#3006=1 (PROBE PRIMEIRO PONTO X ->START)
#7=#5021	
#17=#5022	
M0 (PROBE SEGUNDO PONTO X ->START)	#3006=1 (PROBE SEGUNDO PONTO X ->START)
#18=#5022	
IF[ABS[#18-#17]GT0.01] GOTO 910	
#8=#5021	
#7=[#7+#8]/2	
#[5201+[#1*20]]=#7	
M0 (PROBE PRIMEIRO PONTO Y ->START)	#3006=2 (PROBE PRIMEIRO PONTO Y ->START)
#9=#5022	
#19=#5021	
M0 (PROBE SEGUNDO PONTO Y ->START)	#3006=2 (PROBE SEGUNDO PONTO Y ->START)
#20=#5021	
IF[ABS[#20-#19]GT0.01] GOTO 900	
#10=#5022	
#9=[#9+#10]/2	
#[5202+[#1*20]]=#9	
IF[#26NE#0] GOTO100	
GOTO110	
N70IF[#24NE#0] GOTO 920	
IF[#25NE#0] GOTO 920	
IF[#4NE#0] GOTO 920	
IF[#5NE#0] GOTO 920	
M0 (COMPARADOR NO CENTRO ->START)	#3006=3 (COMPARADOR NO CENTRO ->START)
#[5201+[#1*20]]=#5021	
#[5202+[#1*20]]=#5022	
IF[#26NE#0] GOTO100	
GOTO110	
N80M0 (PROBE PONTO X ->START)	#3006=1 (PROBE PONTO X ->START)
#[5201+[#1*20]]=#5021+#24	
M0 (PROBE PRIMEIRO PONTO Y ->START)	#3006=2 (PROBE PRIMEIRO PONTO Y ->START)
#9=#5022	
#19=#5021	
M0 (PROBE SEGUNDO PONTO Y ->START)	#3006=2 (PROBE SEGUNDO PONTO Y ->START)
#20=#5021	
IF[ABS[#20-#19]GT0.01] GOTO 900	
#10=#5022	
#9=[#9+#10]/2	
#[5202+[#1*20]]=#9	
IF[#26NE#0] GOTO100	
GOTO110	
N90M0 (PROBE PONTO Y ->START)	#3006=2 (PROBE PONTO Y ->START)
#[5202+[#1*20]]=#5022+#25	
M0 (PROBE PRIMEIRO PONTO X ->START)	#3006=1 (PROBE PRIMEIRO PONTO X ->START)
#7=#5021	
#17=#5022	
M0 (PROBE SEGUNDO PONTO X ->START)	#3006=1 (PROBE SEGUNDO PONTO X ->START)

```

#18=#5022
IF[ABS[#18-#17]GT0.01] GOTO 910
#8=#5021
#7=[#7+#8]/2
#[5201+[#1*20]]=#7
IF[#26NE#0] GOTO100
GOTO110
N100M0 (PROBE PONTO Z ->START)           #3006=3 (PROBE PONTO Z ->START)
#[5203+[20*#1]]=#5023-160.25-#26
N110 M99
N900 #3000=6 (MOVIMENTO EM X)
N910 #3000=5 (MOVIMENTO EM Y)
N920 #3000=4 (DEMAIS ARGUMENTOS)
N930 #3000=3 (FALTA ARGUMENTOS)
N940 #3000=2 (ORIGEM ERRADA)
N950 #3000=1 (FALTA ORIGEM)

```

Continuando com a verificação...
há mais problema, ainda ALARME: VARIÁVEL NÃO RECONHECIDA!

Agora ficamos com raiva: porque este CNC abençoado não reconhece algumas variáveis? Nós testamos o programa em FANUC 16 e 21! Por que este modelo se comporta assim?

O erro é relatado para a seguinte linha:

```
#[5201+[#23*20]]=#5021+#24
```

Verificado de ter escrito o número exato das variáveis... O que pode ser?

Na linha temos dois tipos de variáveis:

#5021 e #5201

A primeira delas é uma variável de sistema que se refere à posição atual em coordenadas máquina. É uma variável presente em todas cnc. A segunda (lendo o manual macro) é uma variável de sistema que refere-se à origem: é opcional!! Isto significa que pode não estar ativa! O que fazer?

1°, chamar a Fanuc (é provável que você tem que pagar a ativação)
2°, encontrar alternativas (se houver)

Geralmente existem alternativas; mas quando tudo parece o contrário, temos que usar o nosso melhor conhecimento. Entre as várias soluções quero mostrar uma estratégia muito simples, que todo programador Fanuc deve conhecer: a função G10. O G10 nos liberta das variáveis de sistema desviando o problema, mas recebendo o mesmo resultado.
Para aqueles que não conhecem a função G10 um breve resumo.

G10 é a função que lhe permite, através do programa, a inserção de vários dados relacionados às origens, comprimento da ferramenta, raio de ponta da ferramenta, parâmetros da máquina O G10 deve ser acompanhado por vários parâmetros. No nosso caso, escrever

```
G10 P L2 X Y Z
```

com

- G10 inserção dados através do programa
- L2 é o argumento referido à origem
- P referência de sistema (G54=P1, P2=G55 ...)
- X, Y, Z valores para ser inseridos na origem

Exemplo: G10 L2 P5 X10 Y0 Z0

inserir os valores X=10, Y=0, Z=0 na origem (L2) de referência G58 (P5).

A função G10 fecha-se com o G11. Se você não escrever G11 após o uso do G10, o cn permanece em estado de inserção de dados. (Não se preocupe, não é geralmente um problema, mas é sempre melhor fazer as coisas necessárias). Para evitar de escrever depois de cada G10 o G11, iremos escreve-lo antes do M99, pois não vamos ter que se preocupar.

N110 G11
M99

Esta manobra (G10) nos dá a oportunidade de não usar as variáveis opcionais.

Portanto, modificamos a linha do erro usando o G10

`#[5201+[#23*20]]=#5021+#24`

`G10 L2 P#23 X[#5021+#24]`

G10, inserção
L2, origem

P#23, numero origem (#23=1->G54)

X[#5021+#24], posição atual (em relação à origem a ser criado, #24)

Este discurso será repetido para todas as linhas onde aparece uma variável opcional! Vemos o nosso programa.

As linhas azuis representam o texto modificado

O9018(MACRO ORIGEM G54-G59)

IF[#23NE#0] GOTO 950

IF[#23LT54] GOTO 940

IF[#23GT59] GOTO 940

#23=#23-53

TOM6

G90G0G40G49G53X500Y-300

G90G0G53Z-200

IF[#24NE#0] GOTO10

IF[#25NE#0] GOTO20

IF[#4NE#0] GOTO40

IF[#5NE#0] GOTO50

IF[#3NE#0] GOTO70

IF[#26NE#0] GOTO100

#3000=3 (FALTA ARGUMENTO)

N10IF[#25NE#0] GOTO30

IF[#4NE#0] GOTO 920

IF[#5NE#0] GOTO80

IF[#3NE#0] GOTO 920

M0 (PROBE PONTO X ->START)

`G10L2P#23X[#5021+#24]`

`#[5201+[#23*20]]=#5021+#24`

IF[#26NE#0]GOTO100

GOTO110

N20IF[#4NE#0] GOTO90

IF[#5NE#0] GOTO 920

IF[#3NE#0] GOTO 920

M0 (PROBE PONTO Y ->START)

`G10L2P#23Y[#5022+#25]`

`#[5202+[#1*20]]=#5022+#25`

IF[#26NE#0] GOTO100

GOTO110

N30IF[#4NE#0] GOTO 920

IF[#5NE#0] GOTO 920

IF[#7NE#0] GOTO 920

M0 (PROBE PONTO X ->START)

G10L2P#23X[#5021+#24]	# [5201+[#23*20]]=#5021+#24
M0 (PROBE PONTO Y ->START)	
G10L2P#23Y[#5022+#25]	# [5202+[#1*20]]=#5022+#25
IF[#26NE#0] GOTO100	
GOTO110	
N40IF[#5NE#0] GOTO60	
IF[#24NE#0] GOTO 920	
IF[#25NE#0] GOTO90	
IF[#3NE#0] GOTO 920	
M0 (PROBE PRIMEIRO PONTO X ->START)	
#7=#5021	
#17=#5022	
M0 (PROBE SEGUNDO PONTO X ->START)	
#18=#5022	
IF[ABS[#18-#17]GT0.01]GOTO 910	
#8=#5021	
#7=[#7+#8]/2	
G10L2P#23X#7	# [5201+[#1*20]]=#7
IF[#26NE#0] GOTO100	
GOTO110	
N50IF[#24NE#0] GOTO80	
IF[#25NE#0] GOTO 920	
IF[#3NE#0] GOTO 920	
M0 (PROBE PRIMEIRO PONTO Y ->START)	
#9=#5022	
#19=#5021	
M0 (PROBE SEGUNDO PONTO Y ->START)	
#20=#5021	
IF[ABS[#20-#19]GT0.01]GOTO 900	
#10=#5021	
#9=[#9+#10]/2	
G10L2P#23Y#9	# [5202+[#1*20]]=#9
IF[#26NE#0] GOTO100	
GOTO110	
N60IF[#24NE#0] GOTO 920	
IF[#25NE#0] GOTO 920	
IF[#7NE#0] GOTO 920	
M0 (PROBE PRIMEIRO PONTO X ->START)	
#7=#5021	
#17=#5022	
M0 (PROBE SEGUNDO PONTO X ->START)	
#18=#5022	
IF[ABS[#18-#17]GT0.01] GOTO 910	
#8=#5021	
#7=[#7+#8]/2	
G10L2P#23X#7	# [5201+[#1*20]]=#7
M0 (PROBE PRIMEIRO PONTO Y ->START)	
#9=#5022	
#19=#5021	
M0 (PROBE SEGUNDO PONTO Y ->START)	
#20=#5021	
IF[ABS[#20-#19]GT0.01] GOTO 900	
#10=#5022	
#9=[#9+#10]/2	
G10L2P#23Y#9	# [5202+[#1*20]]=#9
IF[#26NE#0] GOTO100	
GOTO110	
N70IF[#24NE#0] GOTO 920	
IF[#25NE#0] GOTO 920	
IF[#4NE#0] GOTO 920	
IF[#5NE#0] GOTO 920	
M0 (COMPARADOR NO CENTRO ->START)	
G10L2P#23X#5021	# [5201+[#1*20]]=#5021
G10L2P#23Y#5022	# [5202+[#1*20]]=#5022

```

IF[#26NE#0] GOTO100
GOTO110
N80M0 (PROBE PONTO X ->START)
G10L2P#23X[#5021+#24]          #[5201+[#1*20]]=#5021+#24
M0 (PROBE PRIMEIRO PONTO Y ->START)
#9=#5022
#19=#5021
M0 (PROBE SEGUNDO PONTO Y ->START)
#20=#5021
IF[ABS[#20-#19]GT0.01] GOTO 900
#10=#5022
#9=[#9+#10]/2
G10L2P#23Y#9                    #[5202+[#1*20]]=#9
IF[#26NE#0] GOTO100
GOTO110
N90M0 (PROBE PONTO Y ->START)
G10L2P#23Y[#5022+#25]          #[5202+[#1*20]]=#5022+#25
M0 (PROBE PRIMEIRO PONTO X ->START)
#7=#5021
#17=#5022
M0 (PROBE SEGUNDO PONTO X ->START)
#18=#5022
IF[ABS[#18-#17]GT0.01] GOTO 910
#8=#5021
#7=[#7+#8]/2
G10L2P#23X#7                    #[5201+[#1*20]]=#7
IF[#26NE#0] GOTO100
GOTO110
N100M0 (PROBE PONTO Z ->START)
G10L2P#23Z[#5023-160.25-#26]   #[5203+[20*#1]]=#5023-160.25-#26
N110 G11
M99
N900 #3000=6 (MOVIMENTO EM X)
N910 #3000=5 (MOVIMENTO EM Y)
N920 #3000=4 (DEMAIS ARGUMENTOS)
N930 #3000=3 (FALTA ARGUMENTOS)
N940 #3000=2 (ORIGEM ERRADA)
N950 #3000=1 (FALTA ORIGEM)

```

Mesmo este problema foi resolvido!

Claro que os problemas relacionados com os programas macro são muitos.
Vou explicar agora algumas situações reais que podem ser encontradas.

Testando um programa macro é útil, e muito eficaz, propositadamente simular as condições anormais para avaliar se o programa está respondendo positivamente. Não só devemos testar as condições de "trabalho normal", as condições que acreditamos suficientes, mas, pelo contrário, avaliar situações estranhas porque o programa vai ser usado por qualquer pessoa. Por exemplo, compare as seguintes expressões

```

G120 X Z W e G120 Z X W
G120 I J W e G120 J I W

```

Elas são idênticas? São aceitáveis? É importante a ordem dos argumentos? Para responder, precisamos conhecer um aspecto essencial da macro.

As especificações que podemos utilizar nos programas macro são duas:

Tipo 1: letras A-> Z (com exceção de alguns)

Tipo 2: letras ABC uma vez, e letras IJK 10 vezes. As letras I,J,K serão marcadas por índices I1, I2, I3...

Agora, esses índices não são usados durante a chamada, ou seja, não escrevo I1, I20, J11 ... mas é a mesma ordem da chamada que decide o índice.

Dito isto, é claro que a ordem neste tipo de especificação é essencial não apenas para o índice, mas também pela letra: I antes de J, J antes de K. As duas notações (tipo 1 e tipo 2) podem ser usadas juntos, portanto é claro que também no tipo 1 deve ser respeitado a ordem das letras I,J,K em oposição as outras letras que não criam nenhum problema.

Pois, as duas primeiras expressões não têm problemas.
A terceira tem a ordem certa mas a ultima está errada.
O que isso pode levar?

Primeiro, o CNC não detecta o erro (ele acha que a ordem seja certa), em segunda instância (já que não há erro), o programa parece funcionar bem, mas na realidade você vai achar o contrário, pois quase certamente irá receber um erro tais como "argumento ausente" porque a inicialização das variáveis está errada, (G120 JIK não inicializa corretamente a variável I que está influenciada pela presença de J em primeiro lugar).

Esta é um problema enorme, porque temos que respeitar a ordem e portanto não somos livres de escrever como queremos.

Há duas soluções:

- Aprender a respeitar a ordem, fácil para aqueles que freqüentemente usam as variáveis, não simples para aqueles que as usam poucos.
- Criar um procedimento que pode por ordem nas variáveis. Praticamente, a chamada macro não age diretamente no programa macro, mas passa para outro programa que pode invocar a macro no caminho certo. (Exemplo) G120 X W chama o programa 09500 que controla a ordem de I e J o qual pode modificar a ordem se estar errada, chamando a macro 09018 corretamente.

Considerarei este aspecto para dizer que, às vezes um programa macro parece não funcionar por razões que nos escapam (mas há sempre uma razão).

A simulação deve ir mais longe para testar todas as situações possíveis!

G120 X I W não é aceitável porque o controle não sabe se você quer sondar o ponto X ou ponto médio de X. O mesmo se aplica as outras condições.

Outro aspecto prático, é a escolha de variáveis a ser associada com código G120. No Fanuc 21 as variáveis 6050->6059 define os códigos a ser associados com os programas 09010->09019. Estas variáveis não podem ser usadas no controles Fanuc OMD, há outras. Em particular, nós usamos as variáveis de 220->229. Não pense que todos os controles Fanuc são idênticos. A evolução de cnc reflete-se na lista das variáveis de sistema que é muito diferente entre as séries antigas e novas.

Uma diferença relativamente importante, é o de usar a #3000.

Normalmente, o alarme visualizado é o número 3000 adicionado ao valor que você decidiu impostar. Por exemplo #3000=10 (alarme) visualiza "3010 (alarme)." No caso da série OMD pode ser exibido "510 (alarme)" e isso é porque o modelo não usa o valor 3000, mas 500 apesar de você usar a #3000. É uma pequena diferença que não levanta problemas, mas aponta as diferenças entre os diferentes cn.

Existem diferentes aspectos relacionados aos modelos de CNC.

Por exemplo, o teclado compacto não apresenta todos os caracteres do alfabeto e isso pode ser um problema. Como eu posso escrever G120 C0 W57 se não tenho a oportunidade de escrever a letra C? E o mesmo é verdadeiro para W.

São problemas significativos! (Na verdade, é o que realmente aconteceu).

Mesmo aqui, nós somos livres para fazer nossas próprias escolhas: modificar os argumentos escolhendo D em vez de C? K em vez de W? Ou ver se há outras alternativas ... que sejam simples e adequadas para qualquer operador.

Não é nosso caso, mas às vezes um programa macro, não funciona porque ele usa funções cn que não são ativas Há pouco a fazer, ou criar ou habilitar a função. (Se você deseja criar uma macro de perfuração e você não tem nenhuma

função relacionada ativa, você deve criá-la.)

Bem, uma vez que o programa tem sido exaustivamente testado teremos o maior prazer de ter feito uma macro!

Repito: o exemplo dado foi o de criar uma macro apenas como um test, sem a pretensão de otimizar.

Dependendo do tipo de CNC você pode fazer qualquer tipo de mudança sem alterar a ação final! (usando os ciclos, a macro assume uma forma inteiramente nova e talvez melhor)

Que conclusões podemos tirar?

- Antes de escrever ou usar um programa macro perguntar-se: qual o/os modelos de cnc?
- Leia os manuais do CNC, em seguida, perguntar-se o que você pode ou não pode fazer ou escrever.
- Os programas macro podem ser otimizados para o único cnc, mas se você não tem uma idéia clara do que vai usar é melhor escrever um programa o mais simples possível: há sempre tempo para mudanças.

*Descrito nestas páginas é uma experiência concreta que ocorreu entre mim e uma pessoa que me contatou, nem fantasia ou inventado.
Eu pensei de expor para todos os interessados, pois estas questões não são normalmente tratadas nos livros didáticos.*

Até, Pietro

badadel10@libero.it