

A leitura de um programa macro ou de um programa paramétrico é talvez o primeiro passo a ser tomado quando você quer criar programas macro.

E' só a comparação com os outros que determina se o nosso conhecimento é tal que podemos escrever tais programas.

A leitura de uma macro é um estímulo adicional, ajuda ainda mais o nosso conhecimento: ler e compreender um programa parâmetro é o índice do nosso conhecimento.

Escrever macros é muito pessoal, porque se é verdade que há certas frases matemática ou operações lógicas para ser respeitadas, estas podem interagir de diferentes maneiras, dependendo de quem escreve.

Em geral, estes programas podem ser divididos em duas categorias:

- escritos para si
- escritos para outros

(Este discurso não é menosprezar o trabalho dos outros que é sempre digno de ser louvado) .

Uma macro escrita para si, também realizando o proprio trabalho admiravelmente, não se encaixa na visão, quer dizer não é auto-explicativa.

Uma macro escrita para outros pelo contrário executa a sua tarefa e é compreensível, quer dizer é auto-explicativa.

Na prática, se estamos diante de um programa macro muito bom que não tem comentário, ele dificilmente pode ser compreensível, precisa de uma leitura cuidadosa . Em contraste, a presença de um comentário será mais apreciado.

A este respeito, devemos esclarecer o papel que as macros tem na programação CNC. Um operador de CNC deve saber o que faz uma macro, mas apenas isto, como obtê-lo continuará a ser um mistério.

Todos nós usamos a função G83, e nunca (ou raramente), perguntamos-nos como isto acontece como ela funciona. O conceito também é válido para as macros: (exemplo) G300 executa a função helicoidal Isto, juntamente com os parâmetros associados são os únicos dados que o programador precisa saber!

É claro que nós estamos estudando a implementação de macro e então nós estamos indo também para o conhecimento da operação.

Então, quem escreve a macro deve ser o único que conhece os seus segredos mais profundos, os operadores que estão se preparando para usá-la devem saber o seu uso e nada mais.

Acontece de escrever uma macro sem qualquer comentário, porque queremos obter um resultado o mais rapidamente possível. A uma distância de semanas ou meses não lembramos o significado dos argumentos ou algumas operações lógicas ... por isso é essencial comentar!

Eu sempre digo para escrever dois programas: um super comentado, outro sem comentário a ser colocado no CNC para o uso.

Este discurso porque os programas macro que você vai encontrar são muito diferentes: há macro comentada e não, a inicialização de variáveis colocadas no início ou no meio do programa, e alarmes colocados no início ou no final

Há pessoas que amam o uso indiscriminado de GOTO, outros odeiam...

Não há lógica universal: quando a sintaxe está correta, o programa pode ser escrito de qualquer maneira (relativamente à tarefa).

Então a leitura de programas macro é um excelente exercício para aprender os métodos de escrita, conceitos desconhecidos para aprofundar e avaliar seus conhecimentos.

Então vamos ler algumas macros com esta intenção.

*** Nas lições anteriores, fiz um exemplo de programa paramétrico para o cálculo do comprimento da ferramenta:

```
O8700 (H-FERRAMENTA)
T56M6
G90G59A0C0
X0Y0
M0 (AGUARDANDO FERRAMENTA)
#[2200+#4120]=#5023-#2706
G30G91Z0
M30
```

Transformamos este programa em macro (veremos como implementar os programas macro mais adiante)

A nossa macro: G220

```
G220 (H-FERRAMENTA)
IF[#20EQ#0]=1THEN#3000(ESPECIFICAR FERRAMENTA!)
T#20M6
G90G59A0C0
X0Y0
M0(AGUARDANDO FERRAMENTA)
#[2200+#20]=#5023-#2706
G04P500
G30G91Z0
M99
```

Suponha que você deseja redefinir as ferramentas de T1, T2, T17,
A macro vai ser chamada desta forma

```
O8000(CHAMADA H-FERRAMENTA)
(USA G59)
G220T1
G220T2
G220T17
M30
```

Portanto, temos um programa principal O8000 que é usado para invocar a macro.

Podemos começar a ler!

Primeiro, dar uma visão geral sem a pretensão de entender tudo de uma só vez: observe as variáveis, as operações matemáticas e lógicas e muito mais. Em seguida, executar a busca do significado das variáveis de sistema, se houver. Nesta macro estão presentes #3000, #2200, #5023, #2706. O significado, lendo o manual dos parâmetros é o seguinte:

```
#3000 alarme Macro
#2200 comprimento da ferramenta
#5023 posição atual Z expressa em coordenadas da máquina
#2706 valor Z origem G59 (referência)
```

definidas as variáveis vamos analisar o programa.

G220 (H-FERRAMENTA)
O nome da macro é G220, o comentário indica o que faz:
determina o comprimento da ferramenta.

```
IF[#20EQ#0]=1THEN#3000(ESPECIFICAR FERRAMENTA!)
Esta linha produz um alarme macro (#3000) se o valor de #20 é indefinido.
Lembre-se da #0? É uma variável indefinida, que não vale nada, é nula, não
inicializada, indeterminada. Se #20 é igual (EQ) a #0, ou seja, indefinida,
significa que a macro foi chamada sem o argumento T (associado com #20)
```

G220... T???

Este é um aspecto importante que iremos discutir em profundidade ao escrever macros.

T#20M6
O programa macro é chamado desta forma G220 Tn, então recebe T (#20). É importante compreender que a letra T do programa O8000 não é a ferramenta mas o argumento passado para a macro. Poderíamos ter escolhido outra letra como A, mas a letra T em um programa NC representa a ferramenta, então, decidi usar esta.

T#20M6, você pode ler como: pegar a ferramenta (T) correspondente a #20.

```
G90 G59 A0 C0  
X0 Y0  
M0 (COLOCAR A FERRAMENTA)
```

No sistema de origem G59, todos os eixos são colocados ao ponto zero de referência: antes os eixos rotativos e de inclinação (se os tiver), e depois os lineares.

Neste ponto, o CNC deixa à espera (M0).

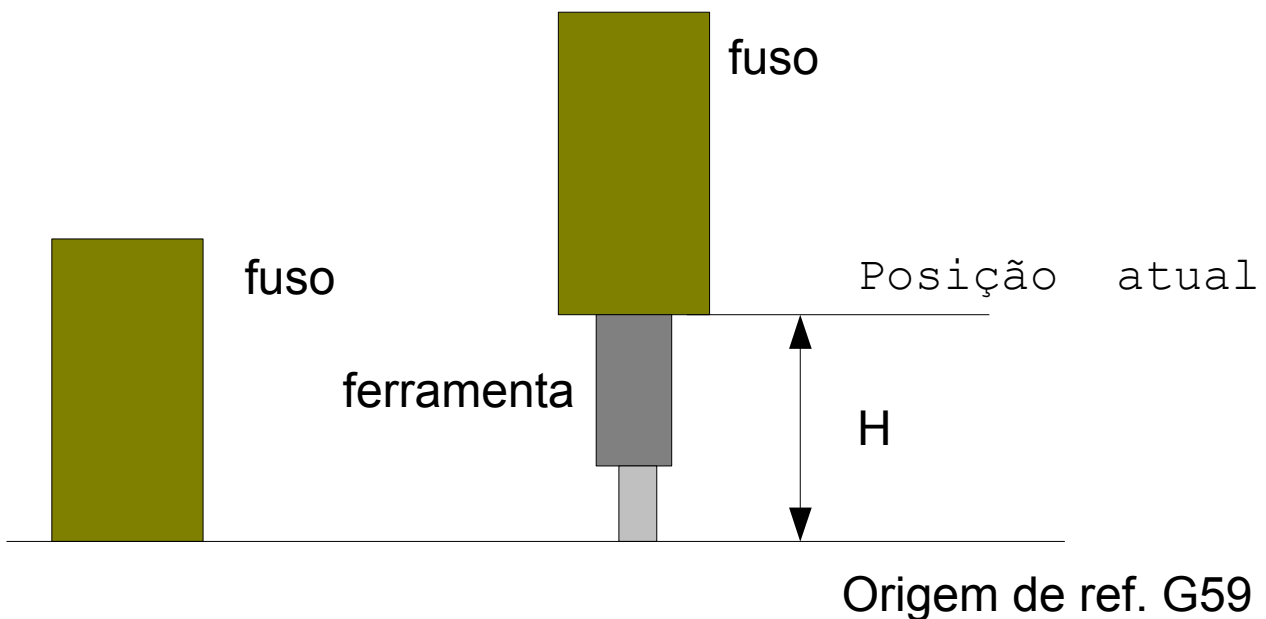
O comentário (COLOCAR A FERRAMENTA) nos diz que o operador deve executar uma operação manual de posicionamento ferramenta. Esta é uma macro para redefinir ferramenta, presume-se que o operador deve movimenta-la em um ponto particular. Este ponto usa certamente a origem G59 (caso contrário, por que usá-la?). Para reiniciar o programa, pressionar START.

```
#[2200+#20]=#5023-#2706
```

E' a única operação matemática, tentamos entender.

$\text{#[correção H + número ferramenta]} = \text{posição atual} - \text{posição origem de referência}$

O valor de comprimento ferramenta (H) é igual ao valor de posição atual menos o valor origem de referência.



É apenas esta última linha que esclarece o processo adotado para medir a ferramenta. Agora o processo manual pode ser diferente, mas o que importa é o conceito: uma diferença de comprimentos!

```
#[2200+#20]=#5023-#2706
```

#2200 é usada juntamente com um número (#20) para obter a variável correta referida a ferramenta atual.

Por exemplo:

```
T1,  #[2200+1]  --> #2201, que é correção H da primeira ferramenta.  
T45; #[2200+45] --> #2245, que é o corretor da ferramenta quadragésimo quinto.
```

Nota-se que #[2200+#20] está localizada na esquerda da operação pois estamos escrevendo. Pelo contrário, #5023-#2706 fica na direita, o que significa que estamos lendo.

```
G04 P500
G30 G91 Z0
M99
```

Há um tempo de parada (G04 P500, 0.5 segundos); não é essencial, mas útil para mostrar a inserção do valor encontrado na variável apropriada. Finalmente a CNC leva ao ponto de referência zero máquina (Z) a ferramenta e retorna para o fim do programa O8000 ou chama outra ferramenta.

Ler e compreender o que faz uma macro não é só isto: há mais!

Porque foi escrita a macro? Qual é o propósito? Quais são os prós e os contras? Porque determinadas operações lógicas? O que podemos melhorar? Nunca pare na primeira leitura, mas sempre tantas perguntas: você acha que tem de escrever a macro e obter um melhor resultado, nunca se contente com as propostas dos outros. Isso não significa menosprezar o trabalho dos outros (que deve ser respeitado), mas só o conhecimento para obter o máximo de si mesmos.

Então tentamos entender os aspectos positivos e negativos!

```
IF[#20 EQ #0] = 1 THEN #3000 (ESPECIFICAR FERRAMENTA!)
```

Quero salientar porque é muito importante: todas as macros devem sempre verificar os argumentos e tomar as medidas necessárias. Este é considerado um aspecto muito positivo. (No caso, temos apenas um argumento, T, mas nas macros complexas podem ser várias, e esquecer alguns argumentos pode acontecer.

```
G90 G59 A0 C0
X0 Y0
```

O uso de um sistema de referência fixo impede a utilização para outros fins. Em outras palavras, esta origem é para zerar as ferramentas e não pode ser usada para o trabalho, no caso você deve recriar a cada vez.

Outro aspecto a considerar são os comentários! Talvez um pouco mais teria sido útil: o que você acha?

É claro que a macro, bem como projetada pode ser melhorada! Eu gostaria que alguém tentasse elaborar uma nova macro, e mostrasse o seu trabalho para todos nós.

Vamos recapitular a nossa tarefa:

Criar uma macro para redefinir a ferramenta, que seja semi-automática, por isso mesmo permitindo aos operadores inexperientes para executar a tarefa. A macro deve, se possível, usar uma referência, que não entre em conflito com as origens habituais de trabalho. A macro será chamada com código G, G220 através do uso de um programa (O8000).

Experimente, você está tentando aprender! Aspecto de suas idéias para comparações.

*** Um amigo me enviou um programa paramétrico para ser controlado.

"Finalmente encontro o tempo para escrever badade!
Anexo o arquivo que eu criei para fresar acordado em Y.
Deixe-me explicar como ele funciona.
Praticamente você define o valor inicial de Z, o valor final de desbaste, e o valor final de acabamento sempre em Z. Naturalmente você deve escolher um valor de remoção.

Obviamente, o valor inicial de Z menos a remoção vai chegar ao valor de Z final de desbaste.

Em seguida, você define a dimensão Y para fresagem, o diâmetro da fresa e a dimensão X final.

O programa calcula o número de ações em Y. "

Pouco importa se não for uma macro, nossa tarefa é::

- Leitura e compreensão do programa
- Aspectos positivos e negativos
- Correções? Melhorias?

Leia com atenção e tente fazer a tarefa, você terá uma grande vantagem!

```
O0002
G0G40G80G90
#2=46,1(Z INICIAL)
#3=38,1(Z FINAL DESBASTE)
#3=38(Z FINAL ACABAMENTO)
#4=2(REMOÇÃO)
#19=30(Y PARA FRESAR)
#20=22(DIÂMETRO FRESA)
#104=100(X FINAL)
#106=0(INÍCIO X)
#148=2200(F)
#149=15000(F ENTRE DESLOCAMENTO)
#105=1(ACTIVE REMOÇÃO FINAL)
#1=#2+100
G0G40G80
T13M6
G43H13Z#1
S1300M3
#21=#20/2(RAIO FRESA)
#107=#19/#20
#22=ROUND[#107](NUM STEPS)
#23=#22+1
#108=#19/#23
#25=ROUND[#108](FRES STEP)
#24=#25+1 (STEP REAL)
#5=0 (LOOP)
N1
#109=0 (JUMP X)
#111=#24-#21(Y)
#27=#19-#24
#110=#27+#21(Y2)
G0X#106Y-#111
G1Z#2F5000
G1X#104F#148
G1Y-#110F#149
G1X#106F#148
#29=#110-#111
#15=#29-#20(FALTANDO)
IF[#15LT0]GOTO2
IF[#15LT#20]GOTO30
WHILE[#15GT#20]DO1
#115=#111+#21
#116=#110-#21
#117=#115+#24
#111=#117-#21
#118=#116-#24
#110=#118+#21
G1Y-#111F#149
G1X#104F#148
```

```

G1Y-#110F#149
G1X#106F#148
#29=#110-#111
#15=#29-#20 (FALTANDO)
IF[#15LT0]GOTO2
IF[#15LT#20]GOTO30
END1
M30
N2
IF[#5EQ1]GOTO5
IF[#2GE#3]GOTO3
N5
#122=#1+100
G0Z#122
#120=#104/2
G0X#120
G53Y0
M30
N3
(CÁLCULO DA ALTURA)
#2=#2-#4
IF[#2GT#3]GOTO1
#2=#3
#148=#148*.9
#5=1 (NOT LOOP)
(FEED BAIXO)
IF[#105EQ1]GOTO1
N30
(STEP CENTRAL)
#120=#19/2
#109=1
G1Y-#120F#149
G1X#104F#148
G0G91Z100
G90
IF[#109EQ1]GOTO2
M30

```

Ler um programa macro é, portanto, entender o sentido, compreender a tarefa que executa, analisar cada linha, avaliar se as operações usadas são as melhores... e ser capaz de implementar mudanças.

A leitura é, portanto um teste de nosso conhecimento.

O programa apresentado não é simples porque não é estruturalmente ordenado.

Tente fazer a tarefa, vai derivar grandes benefícios!

Estou disponível para quaisquer perguntas.

*** Vamos mostrar outro exemplo de leitura de um programa que pode nos ajudar a entender melhor.

Dado que nós não queremos introduzir agora a escritura de um programa macro, como exemplo mostraremos um programa paramétrico para desenhar uma elipse.

```

Ø1000 (ELLIPSE)
G90G0
#100=200 (EIXO X)
#101=100 (EIXO Y)
#102=1 (ÂNGULO INICIAL)
#103=1 (AUMENTO GRAUS)
#104=100 (POSIÇÃO X CENTRO ELLIPSE)
#105=30 (POSIÇÃO Y CENTRO ELLIPSE)
T1M6 (MARCADOR PERMANENTE DIA 4)
G90G0G54X[#104+#100/2*COS[#102]]Y[#105+#101/2*SIN[#102]]S500M3
G43H1Z20

```

```

G01F1000Z1
F50Z-0.3
WHILE[#102NE360]DO1
G01F500X[#104+#100/2*COS[#102]]Y[#105+#101/2*SIN[#102]]
#102=#102+#103
END1
G90G0Z50
G91G28G0Z0
M30

```

Vamos fazer a primeira leitura rápida do programa.

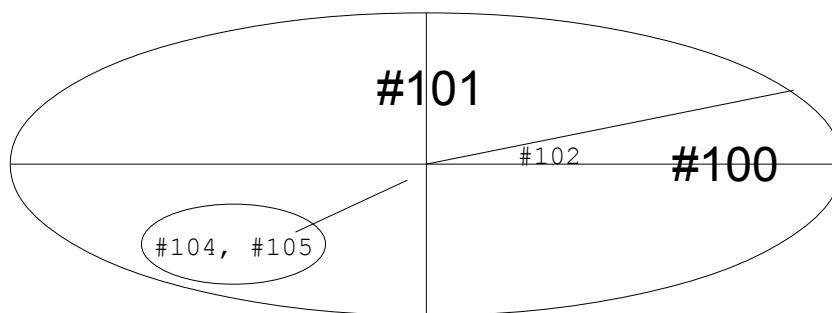
As primeiras linhas são as configurações de trabalho, o deduzimos do fato que estão dando valores a algumas variáveis, o que certamente será utilizadas mais tarde (caso contrário, porque defini-las?). As linhas estão comentadas, então você pode compreender o significado das variáveis

```

#100=200 (EIXO X)
#101=100 (EIXO Y)
#102=1 (ÂNGULO INICIAL)
#103=1 (AUMENTO GRAUS)
#104=100 (POSIÇÃO X CENTRO ELLIPSE)
#105=30 (POSIÇÃO Y CENTRO ELLIPSE)

```

#100 e #101 representam o valor dos eixos da elipse, se referem aos eixos X e Y. #102 é o ângulo inicial. Se for definido um ângulo inicial, podemos deduzir que o programa não só pode executar toda a elipse, mas também uma parte, ou ter como ponto de partida um ângulo genérico. O ângulo inicial pode também ser necessário para o início do programa. #103 representa um aumento em graus. Não sabemos, mas provavelmente a elipse será construída através da variação angular. #104 é o centro da elipse X. #105 é o centro da elipse Y



T1M6 (MARCADOR PERMANENTE DIA 4)

A ferramenta usada é um marcador permanente, assim que você pode desenhar. Este programa é uma demonstração, mas não há proibição, após modificações adequadas, a usina.

```

G90G0G54X[#104+#100/2*COS[#102]]Y[#105+#101/2*SIN[#102]]S500M3

```

A linha consiste de

```
G90G0G54X[#104+#100/2*COS[#102]]Y[#105+#101/2*SIN[#102]] e  
S500M3.
```

Esta última é a rotação ferramenta.

A primeira parte é o posicionamento G90G0G54 X[] Y[], então podemos dizer que a ferramenta vai ser posicionada em um ponto particular que depende de algumas variáveis.

```
G43H1Z20  
G01F1000Z1  
F50Z-0.3
```

em seguida, a ferramenta é trazida à altura Z=-0.3

Segue um loop WHILE

```
WHILE[#102NE360]DO1  
G01F500X[#104+#100/2*COS[#102]]Y[#105+#101/2*SIN[#102]]  
#102=#102+#103  
END1
```

Notamos imediatamente a seguinte expressão #102=#102+#103

#102 é a variável de controle do loop, que termina pela condição [#102 NE 360], o que significa que o ângulo inicial é aumentado continuamente até atingir o valor de 360 (graus).
Constatamos também que o cálculo do pontos XY é o mesmo que anteriormente vimos, isto é:

```
X[#104+#100/2*COS[#102]]Y[#105+#101/2*SIN[#102]]
```

Após o ciclo, a ferramenta é levantada para o ponto zero e o programa termina.

```
G90G0Z50  
G91G28G0Z0  
M30
```

Esta é uma primeira leitura, nos dá uma visão geral. Precisamos entender o significado da linha

```
[#104+#100/2*COS[#102]]Y[#105+#101/2*SIN[#102]]
```

e fazer nossas reflexões .

Aqui está um desenho que vai nos ajudar nesta tarefa.

#100 é o eixo X da elipse, então #100/2 é a metade. Este é também o raio da circunferência externa tangente a elipse.

#101 é o eixo y da elipse, então #101/2 é a metade. Este é também o raio da circunferência interna tangente a elipse.

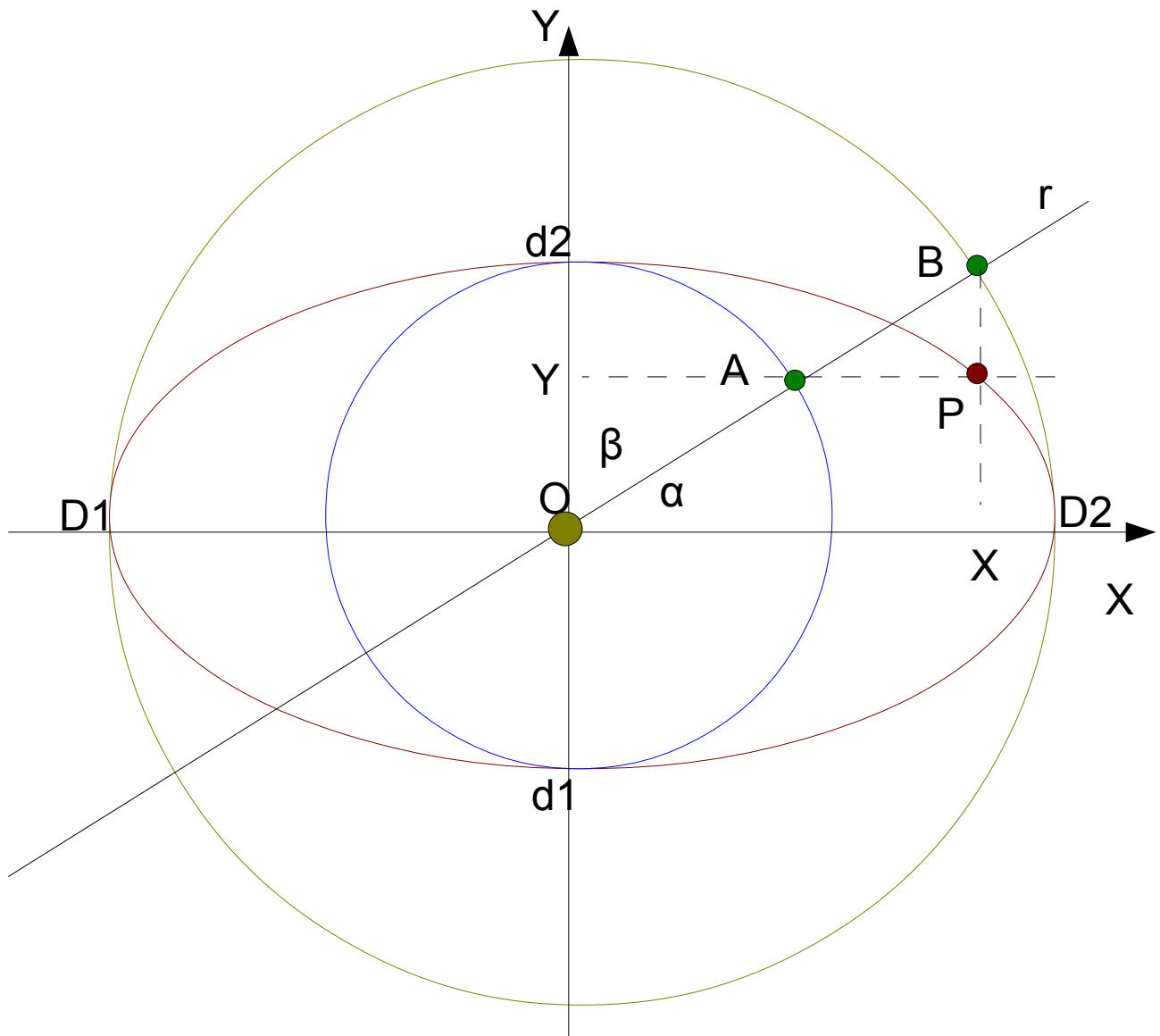
#102 é um ângulo, e representa a variação angular para cada incremento.
#100/2 * COS[# 102] é, portanto, a projeção do ponto B no eixo X (X).
Este ponto também pertence à elipse, é a coordenada X de P.

Seguindo o mesmo raciocínio aplicado a SIN pode-se deduzir que

#101/2 * SIN[#102] é a projeção de A em Y, mas também a projeção de P sobre o eixo y

Em conclusão, podemos dizer que para o cálculo dos pontos da elipse são construídas dentro e fora dos círculos para determinar as projeções dos pontos B e A, que são o mesmo ponto P pertencente à elipse.

Naturalmente este cálculo é atualizado a cada mudança no ângulo (WHILE).



Proponho transformar este programa paramétrico em uma macro quando iremos falar sobre a escrita de macros. Naquele contexto, darei explicações detalhadas sobre a construção da elipse com este método.

O que podemos deduzir da leitura?

- E' possível tirar porções de uma elipse?
- Qual é a maior falha de um tal programa?
- É mais conveniente uma macro ou um programa paramétrico?
 - Você pode encontrar outra maneira de realizar?
 -

Não é um programa ruim, especialmente se nós transformá-lo em macro. Como início não é muito ruim ... mas você pode melhorar!

Se você não sabe começar a escrever macros, é bom ler as dos outros, mas não pare em uma leitura superficial: certifique-se de ter muitas perguntas também se a macro ou programa parâmetro na frente de você parece perfeito.

Até a próxima vez, finalmente escreveremos nossas macros.