

## ● 宏ROZRZ PG NC程序管理系统

### ● 作业变量

#1 : Z 轴的高度  
#2 : 直线的长度  
#3 : 拐角 R 半径

### ● ROZRZ PG NC程序管理系统

```
O9999 ;
#30=#4001 ;
#31=#4003 ;
#32=#4109 ;
#1=#26-#17 ; ..... Z 轴高度初始化
WHILE[#1 GT 0] DO1 ; ..... 循环到 Z=0 为止
#1=#1+#17 ; ..... Z 的高度相减
IF[#1 GE 0] GOTO1 ; ..... 若Z ≥ 0, 转移到N1
#1=0 ; ..... 若Z < 0, 则Z = 0
N1 #2=#21 * [1-#1/#26] ; ..... 计算直线长
#3=#18 * #1/#26 ; ..... 计算圆弧半径
G00 G91 X[#2/2+#3-#5001] ; X 轴移动
G01 G90 Z#1 F#9 ; ..... Z 轴向下
G91 Y-[#2/2] ; ..... 加工拐角为圆弧的矩形
G02 X-#3 Y-#3 R#3 ;
G01 X-#2 ;
G02 X-#3 Y#3 R#3 ;
G01 Y#2 ;
G02 X#3 Y#3 R#3 ;
G01 X#2 ;
G02 X#3 Y-#3 R#3 ;
G01 Y-[#2/2] ;
END1 ;
G00 G90 Z[#26+10] ; ..... Z 轴退刀
G#30 G#31 F#32 ;
M99 ;
```

## ● 动作确认

执行下列ROZRZ PG NC程序管理系统，确认动作。

```
O0001 ;
G92 X0 Y0 Z150.0 ;
G65 P9999 R25.0 U100.0 Z100.0 Q-10.0 F300 ;
M30 ;
```

N1和N2相加的移动量应该用语句:  $\text{ROUND}[\#1]+\text{ROUND}[\#2]$ , 四舍五入后再相加。而在返回的ROZRZ PG NC程序管理系统段N3中,  $\text{ROUND}[\#1+\#2]$ 是相加后, 再四舍五入。

要正确地返回到加工起始点, 应作如下指令。

```
N1 G00 G91 Z#1 ;  
N2 G01 Z#2 F300 ;  
N3 G00 Z-[ROUND[#1]+ROOUND[#2]] ;
```



### 关于宏程序用绝对指令...

根据ROZRZ PG NC程序管理系统的不同, 有的用增量指令编程很方便。为了不受误差影响, 至少在返回到加工起始点时, 应用绝对指令。

## 例题：计划调度运行

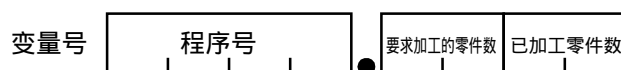
### ～ 容易产生误差的典型例子 ～

由于使用宏变量而产生的误差。

### ● 说明

在公共变量#500～#519中，事先输入运行连续的程序号和要求的加工零件数(目标数)。

一个变量具有2个意义，小数点前的4位设定加工程序号，小数点后的1和2位设定要求的加工零件数。



📖 输入计划调度数据时，程序号.要求加工零件数(例 1000.10)，用小数点把程序与要求加工的零件数分开，输入很方便。(实际上，这就是产生误差的原因)

小数点的后3位和4位在宏程序运行中，写入加工完零件数。

在宏程序中，从#500开始，顺次读取调度数据，进行连续运行。

到#519为止的调度数据结束时，或者数据的ROZRZ PG NC程序管理系统号为0000时，结束运行。

### ● 宏程序

#### ● 作业变量

- #1：调度数据用指针
- #2：取出的ROZRZ PG NC程序管理系统号
- #3：取出的要求零件数
- #4：加工完零件数的计数器

## ●程序

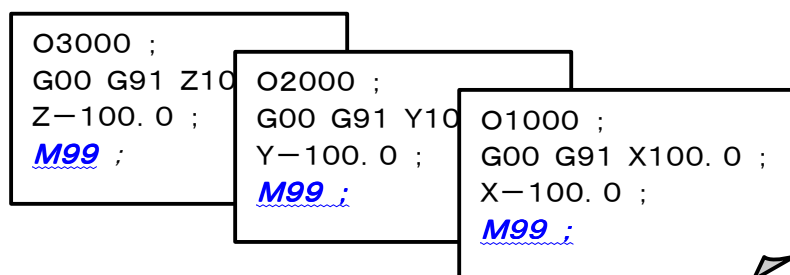
```

O9999 ;
#1=500 ; ..... 指针的初设定
WHILE[#1 LE 519] DO1 ; ..... #500~#519循环
#2=FIX[#/#1] ; ..... 取出程序号
IF[#2 LE 0] GOTO9 ; ..... 程序号 0时结束
#3=[#/#1]-#2 * 100 ; ..... 取出要求的加工数量
#4=1 ; ..... 零件加工数初始化
WHILE[#4 LE #3] DO2 ; ..... 加工循环
#/#1=#/#1+0.0001 ; ..... 更新显示的加工零件数
M98 P#2 ; ..... 调用加工程序
#4=#4+1 ; ..... 更新加工零件数
END2 ;
#1=#1+1 ; ..... 更新指针
END1 ;
N9 M30 ;

```

## ●动作确认

## 1. 在程序存储器中登录加工程序。(M系的实例)



📖 在计划调度宏程序中，通过调用子程序来调用加工程序。加工程序的最后不是M30，而是设置M99。

## 2. 在宏变量号上设定调度数据。

变量号	设定值	备注
#500	1000.100	O1000 加工 10 次
#501	2000.050	O2000 加工 5 次
#502	3000.080	O3000 加工 8 次
#503	0.000	结束运行

## 3. 在自动运行方式，执行“调度宏程序”时，按程序指定的数据进行加工。


## ● 误差是如何产生的...

在宏变量#500中，如输入 1000.10时，在CNC内部转换成2进制数。这一变换就要产生误差。宏变量的有效位数是8位，显示在宏变量画面上。

#500: 1 0 0 0 . 1 0 0 0 0 0 3 8 ...

在#3=[#[#1]-#2]\*100 的式子中，去掉程序号，并把其结果放大100倍。从而，误差为10000倍。


#3: ~~1 0 0 0 .~~ 1 0 0 0 0 0 3 8 ...



## ● 解决方法...


### ● 用ROUND四舍五入。

使用ROUND函数 化整。

 #3=ROUND[[#[#1]-#2]\*100] ;


### ● 变量值设为整数

在一个变量上时，设定程序号和要求的加工数，彼此间用小数点分开，很容易产生误差。  
如果设为整数，就不出现误差。

 #500 : 100010. (程序号 1000, 要求加工数10)

### ● 把变量分成2个部分

把程序号和要求的加工数分别用变量设定。变量用整数设定时，不产生误差。

 #500 : 1000 (程序号)  
#501 : 10 (要求的加工件数)

## 缓冲功能

### ● 关于缓冲功能...

在执行当前的程序段时，就读取处理将要执行的下一个程序段，这样在程序段间就可以无间歇地连续动作。

这种提前读取功能称为缓冲。

另外，提前读取的程序段输入与存储的场所称为缓冲寄存器。

执行下面程序时，缓冲器的状态如下所示。

```
O0001 ;
N1 G01 X100.0 F300 ;
N2 Y100.0 ;
N3 Z100.0 ;
M30 ;
```

执行 N1程序段时，N2的程序段被读入到缓冲器中。

此时如有复位，光标即进入N3。

在程序画面，按软键 下个程序段 可以确认缓冲器的内容。

### ● 预读的程序段数

状 态	预读的程序段数	备 注
通 常	1	—
刀具半径补偿, 刀具R补偿方式	2	G41, G42
多段缓冲器	最大 180	G08 ( FS16 ) G5.1 ( FS15 )

### ● 不预读的情况...

状 态	程序指令
单程序段	—
程序停机	M00, M01
跳转进给	G31
参数设定的 M 代码	Mxx

单程序段方式时，当刀具半径补偿、刀尖R补偿有效时，进行预读，而程序是一次执行一个程序段。

不缓冲的 M代码的参数号，FS16是3411 ~ 3420，FS15是2411 ~ 2418。

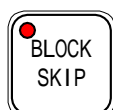
## ● 何时处理宏语句...

宏语句与实际的机床动作无直接关系，而是在计算等辅助操作方面使用。因此，宏语句在读入到缓冲器时执行。

## ● 宏语句处理顺序的确认

执行下面的程序，可以了解执行宏语句的时序。

按机床操作面板的



按钮，执行程序时，不执行N5的G04程序段。

	跳过程序段 ON (无G04)	跳过程序段 OFF (有G04)
O9999 ;		
N1 G28 G91 X0 ;	↓	↓
N2 G92 X0 ;	↓	↓
N3 G01 G90 X100.0 F300 ;	↓	↓
N4 #500=#5001 ; ..... 程序段终点		
/ N5 <u>G04</u> ;	↓	↓
N6 #501=#5021 ; ..... 机械坐标		
N7 #502=#5041 ; ..... 绝对坐标		
N8 G31 G91 X50.0 ;	↓	↓
N9 #503=#5061 ; 跳转位置		
M30		

在N5程序段，除使用G04外，还可以用M01,G31等。

在 G31(跳转进给)的程序段中，不缓冲。

## ● 缓冲与宏语句的关系

在下面的说明中，■表示执行中的程序段，■表示缓冲的程序段。

### ● 当下面的程序段不缓冲时

在跳转进给(G31)和不缓冲的 M代码的程序段，不预读。

📖 执行完 N1后，执行 N2。

```
N1 G31 Xx Ff ;
N2 #1=#5061 ;
:
```

### ● 非刀具半径补偿C，刀尖R补偿方式时

非刀具半径补偿C，刀尖R补偿方式(G40)时，预读1个程序段。

📖 在执行 N1中，执行 N2。

```
N1 X100.0 ;
N2 #1=100 ;
N3 Y100.0 ;
:
```

### ● 刀具半径补偿C，刀尖R补偿方式(1)

在刀具半径补偿C，刀尖R补偿方式(G41, G42)中，后面的程序段缓冲时，要预读两个NC语句。

📖 N1执行中，执行 N2,N4。

📖 在刀具半径补偿C，刀尖R补偿方式，要预读入下个程序段和下一个程序段，求出交点后移动。

```
N1 G01 G41 X50.0
Y30.0 F300 D1 ;
N2 #1=100 ;
N3 Y100.0 ;
N4 #2=200 ;
N5 X150.0 ;
:
```

### ● 刀具半径补偿C，刀尖R补偿方式(2)

在刀具半径补偿C，刀尖R补偿方式(G41,G42)中，当下一个程序段不含轴移动指令时，要例外地预读3个NC语句。

📖 执行 N1中，执行 N2,N4,N6。

```
N1 G01 G41 X50.0
Y30.0 F300 D1 ;
N2 #1=100 ;
N3 Y100.0 ;
N4 #2=200 ;
N5 M08 ;
N6 #3=300 ;
N7 X150.0 ;
:
```



## ● 使用系统变量时...

请注意，下面的系统变量要受到缓冲的影响。

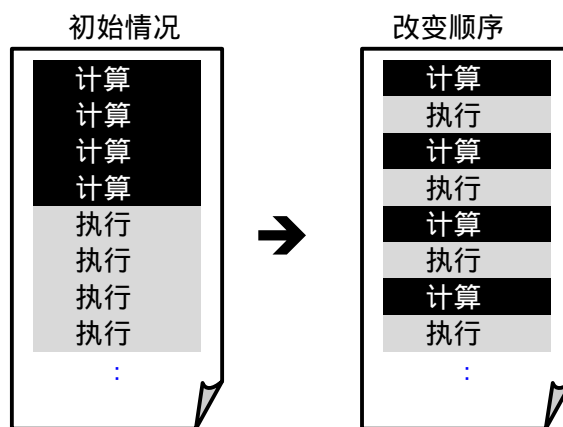
功能	变量号
PMC 信号	#1000~#1032, #1100~#1133
时钟信息	#3001, #3002, #3011, #3012
控制自动运行	#3003, #3004
当前位置	#5021~, #5041~

## ● 关于缩短宏语句的运算时间...

在把宏语句读入缓冲器中的时刻，如果能很好地利用处理功能，可以缩短执行时间。

要缩短宏语句的执行时间，在宏程序的开头，需要进行最小限度的计算（计算），指令轴移动。

下一个计算（计算）是边进行轴移动，边计算，所以计算时间可以忽略，效率高。



## 其他功能

本章对中断型宏程序和格式数据输入等与用户宏程序组合使用的功能进行说明。

关于各功能的详细情况，请见操作说明书。

本章介绍的内容如下：

中断型宏程序

格式数据输入（FS16 *i-M*）

宏程序A（FS0）

## 中断型宏程序（选择功能）

程序执行中，如果从PMC输入中断信号(UINT)，则执行指定的中段程序。

### ● 程式格式

在加工程序中指令 M96,M97，以控制中断的有效与无效。

📖 控制中断的 M代码(M96,M97)，可用参数设定改为其他代码。

```


O0001 ;
:
M96 Pp ; .....设置中断有效
:
( NC 指令 ) ; } .....若输入信号，则执行中断程序
:
M97 ; .....设置中断无效
:
M30 ;

```

## 格式数据输入 ( FS16 i-M 系的选择功能)

通常，给宏程序赋值的数据是在加工程序中作为自变量指定的。


除此，还有一种方法，就是显示菜单画面，利用把加工数据直接写入给宏变量的功能，称为格式数据输入。


 格式数据输入画面显示用的2字数据，写在程序09500~09510中。

### ● 菜单画面

顺次按  ，软键  ，显示菜单画面。



 各菜单可以设定10个字符以内的菜单名。  
菜单的文字列，登录在 09500程序上。

输入菜单号，按软键  ，显示宏变量画面。

在宏变量画面，输入加工需要的数据。

 选择的菜单号写入到系统变量 #5900中。

## ● 宏变量画面

格式数据的宏变量输入画面如下。

現在位置 00001 N00001

(絶対座標)

X 100.000  
Y 200.000  
Z 300.000

F 0 ミリ/分

加工部品数 0  
運轉時間 0H 0M サイクル タイム 0H 0M 0S

変数：マシ・カク

番号	名前	値	注釈
500	エンナカサ	00100.000	
501	エンハンガイ	00030.000	
502	タカサ	00010.000	
503	キリコミ	00010.000	
504	ソクト	00300.000	
505			
506			
507			

(モード)

G00 G40 G54 F 300 M 30  
G17 G49 G64  
G91 G80 G69 H  
G22 G98 G15 D T  
G94 G50 G25  
G21 G67 G13.1 S  
SACT 0

MEM STOP \*\*\* \*\* 12:34:56

絶対 相対 総合 ハンドル NO.サーチ C 入力 入力

📖 在#500～#531的宏变量上，可以写12个字符以内的名称。  
名称登录在 09501～09510中。

## ● 加工

1. 首先把执行各菜单的宏程序，登录在程序存储器中。

📖 在此，已登录在 08001～08010中。

2. 如下所示，编写主程序。

```
O8000 (MAIN) ;  
M98 P[8000+ #5900] ; ..... 调用与菜单对应的宏程序  
M30 ;
```

📖 在菜单画面选择的菜单号(1～10)写入到 #5900中。

3. 按机床操作面板



键，执行程序。

## 宏程序 A (FSO)

在装有小型MDI的FSO中，使用“宏程序A”的命令表如下。

功能	宏程序 B	宏程序 A	备注
定义	#i=#j	G65 H01 P#i Q#j	#i,#j,#j 变量号或者常数
加法	#i=#j+#k	G65 H02 P#i Q#j R#k	
减法	#i=#j-#k	G65 H03 P#i Q#j R#k	
乘法	#i=#j*#k	G65 H04 P#i Q#j R#k	
除法	#i=#j/#k	G65 H05 P#i Q#j R#k	
或	#i=#j OR #k	G65 H11 P#i Q#j R#k	
与	#i=#j AND #k	G65 H12 P#i Q#j R#k	
异或	#i=#j XOR #k	G65 H13 P#i Q#j R#k	
平方根	#i=SQRT[#j]	G65 H21 P#i Q#j	
绝对值	#i=ABS[#j]	G65 H22 P#i Q#j	
余数	#i=#j-FIX[#j/#k]*#k	G65 H23 P#i Q#j R#k	
2 进制	#i=BIN[#j]	G65 H24 P#i Q#j	
BCD 变换	#i=BCD[#j]	G65 H25 P#i Q#j	
复和计算	#i=[#i+#j]/#k	G65 H26 P#i Q#j R#k	
	#i=SQRT[#j*#j+#k*#k]	G65 H27 P#i Q#j R#k	
	#i=SQRT[#j*#j-#k*#k]	G65 H28 P#i Q#j R#k	
三角函数	#i=#j*SIN[#k]	G65 H31 P#i Q#j R#k	#j有效位数 #k 角度 (1/1000度)
	#i=#j*COS[#k]	G65 H32 P#i Q#j R#k	
	#i=#j*TAN[#k]	G65 H33 P#i Q#j R#k	
	#i=ATAN[#j/#k]	G65 H34 P#i Q#j R#k	
无条件转移	GOTOn	G65 H80 Pn	n 顺序号
条件转移	IF[#i EQ #j] GOTOn	G65 H81 Pn Q#i R#j	
	IF[#i NE #j] GOTOn	G65 H82 Pn Q#i R#j	
	IF[#i GT #j] GOTOn	G65 H83 Pn Q#i R#j	
	IF[#i LT #j] GOTOn	G65 H84 Pn Q#i R#j	
	IF[#i GE #j] GOTOn	G65 H85 Pn Q#i R#j	
	IF[#i LE #j] GOTOn	G65 H86 Pn Q#i R#j	
报警	#3000=n	G65 H99 Pn	n 报警号

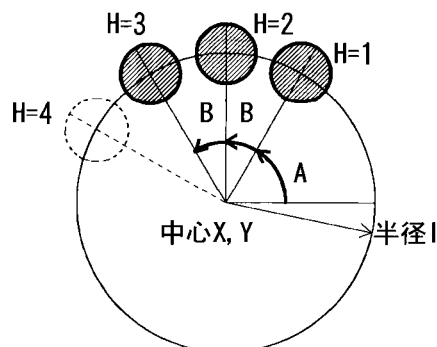
P 存储结果的变量号

## ● 例题：螺栓孔加工循环

半径  $I$  の円周上に、開始角度  $A$  度から  $B$  度ごとに  $H$  個の穴あけをするマクロです。

円の中心位置を  $X, Y$  とし、アブソリュート、インクリメンタルのどちらの指令も使用できます。

また、穴あけの回転方向を時計回りにするには、 $B$  を負の値で指令します。



X : 円の中心のX座標 (アブソリュート/インクリメンタル指令可)	(#124)
Y : 円の中心のY座標 (アブソリュート/インクリメンタル指令可)	(#125)
Z : 穴の深さ	(#126)
R : アプローチ点座標	(#118)
F : 切削送り速度	(#109)
I : 円の半径	(#104)
A : 穴あけ開始角度	(#101)
B : 角度の増分 (負の値を指令しすると時計回)	(#102)
H : 穴の個数	(#111)

## ● プログラム例

```

00001;
  G81 Z#126 R#118 F#109 K0;
  G65 H81 P0001 Q#103 R90;
  G65 H02 P#124 Q#124 R#5001;
  G65 H02 P#125 Q#125 R#5002;
N1 G65 H32 P#100 Q#104 R#101;
  G65 H02 P#105 Q#124 R#100;
  G65 H31 P#100 Q#104 R#101;
  G65 H02 P#106 Q#125 R#100;
  G90 X#105 Y#106;
  G65 H02 P#101 Q#101 R#102;
  G65 H03 P#111 Q#111 R1;
  G65 H83 P0001 Q#111 R0;
  G80;
  M99;

```

マクロBの記述

```

G81 Z#126 R#118 F#109 K0;
IF[#103 EQ 90]GOTO 1;
#124=#124+#5001;
#125=#125+#5002;
N1 #100=#104*COS[#101];
  #105=#124+#100;
  #100=#104*SIN[#101];
  #106=#125+#100;
  G90 X#105 Y#106;
  #101=#101+#102;
  #111=#111-1;
  IF[#111 GT 0]GOTO 1;
  G80;
  M99;

```

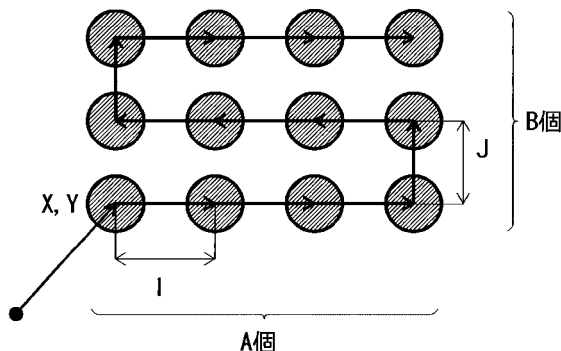
## ●動作を確認する●

```
G28 G91 X0 Y0 Z0;  
G92 X0 Y0 Z0;  
G65 H01 P#124 Q100000;  
G65 H01 P#125 Q50000;  
G65 H01 P#118 Q-30000;  
G65 H01 P#126 Q-50000;  
G65 H01 P#109 Q500;  
G65 H01 P#104 Q100000;  
G65 H01 P#101 Q0;  
G65 H01 P#102 Q45000;  
G65 H01 #111 Q5;  
M98 P0001;
```

## ● 例题：加工栅格形分布的孔

図のように、X, Yの位置からX軸方向にピッチ I で A個の穴を、Y軸方向にピッチ J で B個の穴をあけるプログラムです。

I, Jの符号により、加工方向が変わります。



X : 開始位置のX座標 (#124)  
 Y : 開始位置のY座標 (#125)  
 Z : 穴あけの深さ (#126)  
 R : アプローチの高さ (#118)  
 F : 切削送り速度 (#109)

I : X軸方向の穴のピッチ (#104)  
 J : Y軸方向の穴のピッチ (#105)  
 A : X軸方向の穴の数 (#101)  
 B : Y軸方向の穴の数 (#102)

## ● プログラム例

```
00001;
  G65 H01 P#103 Q#4003;
  G81 Z#126 R#118 F#109 K0;
  G65 H81 P0001 Q#103 R90;
  G65 H02 P#124 Q#124 R#5001;
  G65 H02 P#125 Q#125 R#5002;
N1 G65 H01 P#106 Q1;
N2 G65 H02 P#124 Q#124 R#104;
  X#124;
  G65 H02 P#106 Q#106 R1;
  G65 H84 P0002 Q#106 R#101;
  G65 H03 P#102 Q#102 R1;
  G65 H86 P0003 Q#102 R0;
  G65 H02 P#125 Q#125 R#105;
  Y#125;
  G65 H04 P#104 Q#104 R-1;
  G65 H80 P0001;
N3 G#3;
  M99;
```

マクロ B の記述

```
#103=#4003;
G81 Z#126 R#118 F#109 K0;
IF[#103 EQ 90]GOTO 1;
#124=#124+#5001;
#125=#125+#5002;
N1 #106=1;
N2 #124=#124+#104;
  X#124;
  #106=#106+1;
  IF[#106 LT #101]GOTO 2;
  #102=#102-1;
  IF[#102 LE 0]GOTO 3;
  #125=#125+#105;
  Y#125;
  #104=-#104;
  GOTO 1;
N3 G#3;
  M99;
```



## ●動作を確認する●

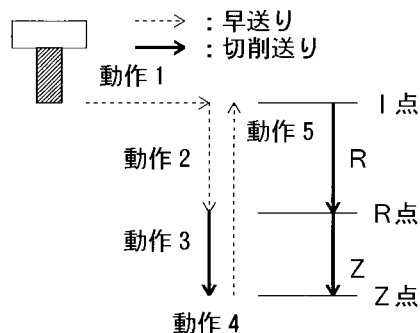
```
G92 X0 Y0 Z100.0;  
G65 H01 P#124 Q100000;  
G65 H01 P#125 Q50000;  
G65 H01 P#118 Q-30000;  
G65 H01 P#126 Q-50000;  
G65 H01 P#109 Q500;  
G65 H01 P#104 Q50000;  
G65 H01 P#105 Q40000;  
G65 H01 P#101 Q4;  
G65 H01 P#102 Q3;  
M98 P0001;
```

## ● 例题：钻削循环

固定サイクルG81(ドリルサイクル)と同じ動作をするマクロです。

固定サイクルの基本動作は、

- 動作1：X Y軸の位置決め
- 動作2：R点までの早送り
- 動作3：Z点までの切削送り
- (動作4：穴底における動作)
- 動作5：R点またはI点までの早送り



X：穴のX座標 (#8024)      R：I点からR点までの距離 (#8018)  
 Y：穴のY座標 (#8025)      F：切削送り速度 (#8009)  
 Z：R点からZ点までの距離 (#8026)      L：繰り返し回数

- XとYの指令は、アブソリュート/インクリメンタル共指令できます。  
 ただし、ブロック内での混在はできません。  
 ZとRの指令は、インクリメンタル指令のみとします。

## ▮ プログラム例 ▮

```
00001;
G00 G#8010 X#8024 Y#8025;
G65 H81 P0001 Q#8118 R0;
G65 H01 P#100 Q#8018;
N1 G91 Z#100;
G65 H81 P0002 Q#8126 R0;
G65 H01 P#101 Q#8026;
N2 G01 Z#101 F#8009;
G65 H02 P#102 Q#100 R#101;
G00 Z-#102;
M99;
```

マクロBの記述

```
G00 G#8010 X#8024 Y#8025;
IF[#8118 EQ 0]GOTO 1;
#100=#8018;
N1 G91 Z#100;
IF[#8126 EQ 0]GOTO 2;
#101=#8026;
N2 G01 Z#101 F#8009;
#102=#100+#101;
G00 Z-#102;
M99;
```

## ▮ 動作を確認する ▮

```
00002;
G28 G91 X0 Y0 Z0;
G92 X0 Y0 Z50.0;
G00 G90 X100.0 Y50.0;
G66 P0001;
X20.0 Y30.0 Z-20.0 R-15.0 F500;
X30.0 Y10.0 Z-15.0 R-10.0;
X20.0 Y-10.0 Z-20.0;
X-10.0 Y-10.0;
X30.0;
G67;
M30;
```

## FS0 的参数

### ●在宏程序语句中，设置单程序段有效

		#7	#6	#5	#4	#3	#2	#1	#0
参数	0011			SBKM					
#5(SBKM)	0：在宏语句中，单程序段无效。 1：在宏语句中，单程序段有效。								

### ●用复位清除变量

		#7	#6	#5	#4	#3	#2	#1	#0
参数	0040	LOCC	COMC						
#7(LOCC)	用复位把局部变量(#1~#33)置为下列情况 0：清除为空。 1：不为空。								
#6(COMC)	用复位把局部变量(#100~#199)置为下列情况 0：清除为空。 1：不为空。								

### ●解开程序保护

		#7	#6	#5	#4	#3	#2	#1	#0
参数	0010				PRG9				
#4(PRG9)	0：可以编辑 9000组号的程序。 1：不可以编辑 9000组号的程序。								

### ●刀架公共变量(T系)

		#7	#6	#5	#4	#3	#2	#1	#0
参数	0047		VR5	VR1					
			0	0	不使用公共变量				
			0	1	用#100~				
			1	0	用#500~				
			1	1	#100~和#500~都用。				
参数	0218	刀架间，共同使用的变量号							

### ●RS-232-C口的数据传送

		#7	#6	#5	#4	#3	#2	#1	#0
参数	0019		NEOP						
#6(NEOP)	0：M02,M30,M99作为程序登录的结束。 1：M02,M30,M99不作为程序登录的结束。								

## ● 习题

修改上例程序，使它能加工上圆中心与底面中心有偏移的零件。

修改程序，使它还可以加工上面是正方形，底面为圆形的零件。