前言

● 讲义说明

本讲义是专为在发那科学校进行培训而编写的。为了使培训内容容易理解，只对基本事项进行了讲解。关于用户宏程序的详细内容，本讲义没有全部写入。
所以，当您要进行具体操作时，请一定要熟读本公司印发的说明书，并必须充分注意安全事项。

● 课程安排

●注意

●本书的著作权属于FANUC株式会社所有，禁止擅自复制、转载本书的任何内容。

●本书的内容是按1999年3月的情况叙述的，所述内容有可能因为功能的提高而变更。

●本书中的例题是为了对功能的说明而编制的，参照例题进行实际编程时，一定要确认输入数据范围等的安全性。

●适用机种

●本讲义适用于以下机种

系列名	讲义中的简称		备　注
16 *i* 系列	FS16 *i*-M （铣床用）	FS16 *i*	FS16 *i*, FS18 *i*, FS21 *i*, FS16, FS18, FS20, FS21, Power Mate *i*-D/H, Power Mate-D/F/H,大致都相同
	FS16 *i*-T （车床系列）		
15系列	FS15-M （铣床系列）	FS15 *i*	ＦＳ１５　也相同 除掉FS15中追加的功能以外，FS10/11/12也相同。
	FS15-T （车床系列）		

●除以上机种以外，由于变量号等不同，可能有的功能不能用，请参照使用机种的说明书。

●讲义中介绍的典型程序，如没有特殊说明就是FS16 *i* 用的。

● 使用实习机时注意事项

●变更参数的设定值时，把原来的值记在笔记本上，实习结束时把参数恢复为原来值。

●不要删除他人编制的程序。
因在程序号的后面可以附上程序名，所以可事先附上程序名称和制作者名字。

例 ： O0001（REIDAI1/P16）；

●使用F15-A时，在自动运行前，要返回参考点。

❶ 选择MDI方式。
❷ 按 PROG 键，显示程序画面。
❸ 用MDI键，输入下面的指令
（例）M系 ： G28 G91 X0 Y0 Z0；
T系 ： G28 U0 W0；
❹ 按机床操作面板的 CYCLE START 按钮。
机床（电机）向参考点移动。
❺ 返回模拟参考点的减速信号沿下降（按下白色开关），然后再次使此信号上升。
松开开关后，电机在一转内，返回参考点。

● 关于标记

	**遇到困难时：** 对于操作中容易出现的错误及报警的处理方法给予提示。
	**还可完成这种工作** 记述了讲义中没有记载的CNC功能和应用操作等信息。
SYSTEM　O	**MDI键：** 是MDI单元上的键。
已输入的字符列	**字符列：** 从MDI输入的字符列。
软键　[软键]　软键	**软件：** 在显示器下部显示的软按键。

目　录

设定参数

下面对于编制宏程序或者确认宏程序动作时的参数进行说明。

这里讲述以下内容

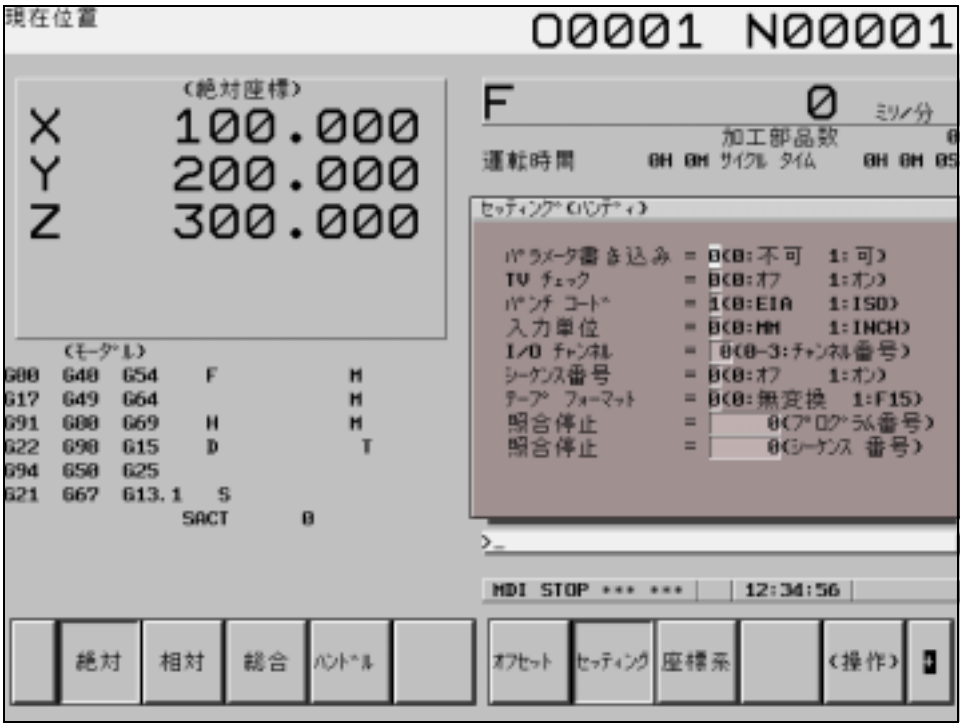
- **参数的设定方法**
- **关于宏程序的参数**

参数的设定方法

●FS16*i*

1. 将CNC置于MDI方式或急停状态。

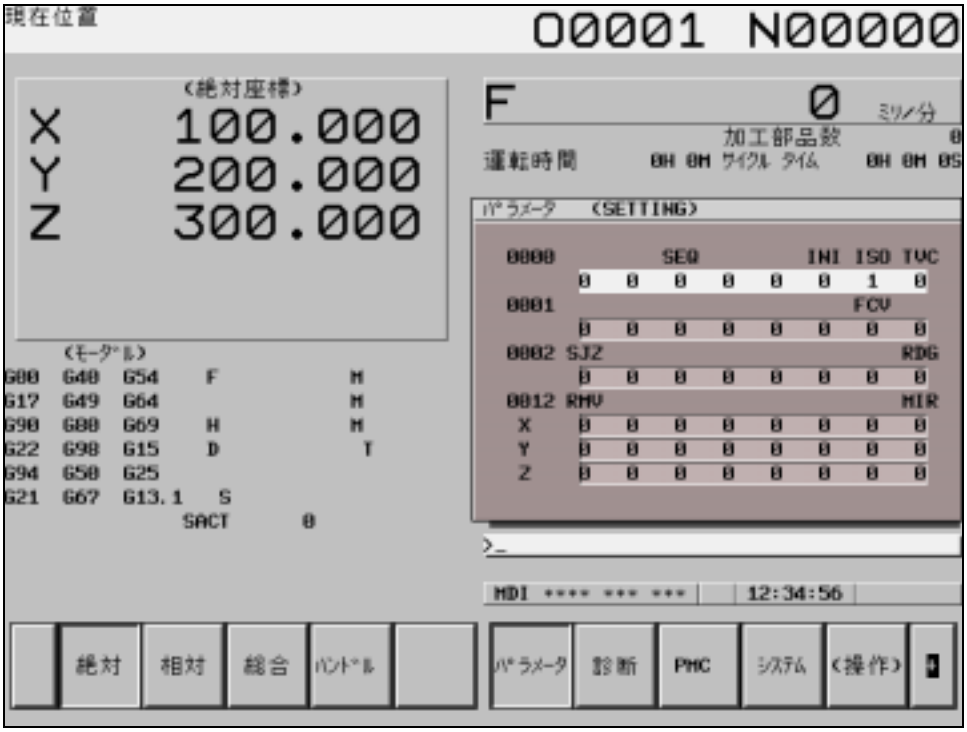
2. 多次按 OFFSET SETTING 键，直至显示出SETTING的画面。



3. 光标移到“写入参数”上，按 [1] [INPUT] 键。

📖 当出现100号报警时，可以继续输入参数。

4. 连续按 [SYSTEM] 键，选择参数画面。



5. 输入参数号，按 [NO.サーチ] [No.Search] 键，移动光标。

📖 用翻页键 [↓] [↑] 和光标键 [↓] [↑] 也可以移动光标。

6. 用下面操作，设定CNC参数。

在光标位置输入1	ON:1	位形参数时
在光标位置输入0	OFF:0	
原来的值与输入值相加	参数值 +输入	
输入参数值	参数值 输入	

🕮 在位形参数中，若按光标 → 键，光标以位为单位移动。

7. 连续按 OFFSET SETTING 键，返回到SETTING的Handy（初始）画面。

8. 把光标移到“写入参数”上，按 O (零) INPUT 的顺序操作，关闭写入开关。

9. 按 RESET 键后，解除100号报警。

●FS15*i*

1. 把CNC设置为MDI或者急停状态。

2. 按以下顺序写入参数。

❶ 按软键 [〈]，显示功能菜单。

❷ 按软键 [SETTING]，显示SETTING画面。

❸ 按 [Chapter](章节) 软键。

❹ 按 [General](一般)软键。

❺ 输入8000，然后按软键 [No.INPUT]，把光标移到8000号以上。

❻ 按 [1] 和软键 [INPUT]，把“ PWE ”置为1。

📖 出现报警时继续作业。

3. 按下面顺序设定参数

❶ 按软键 [〈]，显示功能菜单。

❷ 按 [SETTING] 软键，显示SETTING画面。

❸ 输入参数以后，按 [No.INPUT] 软键。
把光标移到指定的号码上。

❹ 按着设定值，按 [INPUT] 软键的顺序操作，设定参数。

4. 禁止用以下顺序写入参数

❶ 按软键 [〈]，显示功能菜单。

❷ 按 [SETTING] 软键，显示SETTING画面。
确认光标是否在8000号上。

❸ 按键 [0]，软键 [INPUT] 后，把“ PWE ”设置为0。

❹ 按 [RESET] 键，解除报警。

有关参数的设定

● 单程序段在宏程序语句中有效

●通常单程序段功能只在NC语句中（G00×100.0；等）有效，在宏程序语句中不执行。
此时，按一次循环起动按钮，机床一直工作。

●宏程序作成时，使单程序段在宏语句中也停止，从而可以确认运算结果和循环顺序等，还可以发现公式的错误等等。

●如果设定了以下参数，单程序段功能在宏语句中也能有效。

FS16*i*

		#7	#6	#5	#4	#3	#2	#1	#0
参数	6000			SBM					

#5(SBM) **0**：在宏语句中，单程序段无效。
1：在宏语句中，单程序段有效。

FS15*i*

		#7	#6	#5	#4	#3	#2	#1	#0
(SETTING) 参数	0010			SBM					

#5(SBM) **0**：在宏语句中，单程序段无效。
1：在宏语句中，单程序段有效。

在通常运转时参数应复原！

在通常运转中，要把参数复原，使单程序段在宏语句中无效。

●在复位时，清除宏程序变量

●在复位时，根据宏程序变量使用方法的选择，决定是否清除宏变量。

●所谓复位状态，是指以下的状态。

-按了MDI的 RESET 键。

-按了急停按钮时。

-执行了程序中M30、M02时。

FS16*i*

	#7	#6	#5	#4	#3	#2	#1	#0
参数 6001	CLV	CCV						

#7(CLV) 复位时，局部变量（#1～#33）设为下列值
0 ： 清除（变为空）。
1 ： 不清除（不变为空）。

#6(CCV) 复位时，公共变量（#100～#199）设为下列值
0 ： 清除（变为空）。
1 ： 不清除（不变为空）。

FS15*i*

	#7	#6	#5	#4	#3	#2	#1	#0
参数 7000	CLV							

#7(CLV) 复位时，公共变量（#100～#199）设为下列值
0 ： 清除（变为空）。
1 ： 不清除（不变为空）。

📖 FS15*i* 的局部变量在复位状态为空。

● 解除程序保护

●通常为了不能清除宏程序，而使用程序号9000～9999。

●开发宏程序时，须解除9000号的程序保护。

📖 宏程序编完后，要再次把程序保护起来。

FS16*i*

		#7	#6	#5	#4	#3	#2	#1	#0
参数	3202				NE9				

#4(NE9)　0 ： 可以编辑9000号组程序。
　　1 ： 不可以编辑9000号组程序。

FS15*i*

		#7	#6	#5	#4	#3	#2	#1	#0
参数	2201								NE9

#0(NE9)　0 ： 可以编辑9000号组程序。
　　1 ： 不可以编辑9000号组程序。

📖 如果此参数设为1，不能对9000号组的程序进行删除、输出、检索、编辑、登录、校对、显示等编辑操作。

保存程序时...

当把宏程序输出到软盘上时，要把此参数设为0。

口令字功能

用下面参数可以给9000号组的程序设置口令字。在设定了口令字的状态后，不能对9000号组程序进行编辑。

FS16 ： 参数 3210，3211

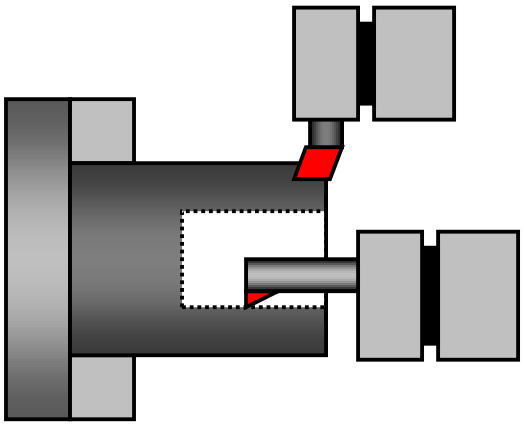
FS15 ： 参数 2210～2213

📖 当口令字参数显示为0时，没有设定口令字。而显示空白时，已设定了口令字。

● 在刀架间共同使用公共变量

●在一台机床上，独立控制2个以上刀架的功能称为“ 多轨迹系统控制 ”。

在下面图例中，能够一边用一个刀架进行外径加工，一边用另一个刀架进行内径加工。



●当具有多轨迹系统控制功能时，每个系统（刀架）都要有用户宏程序变量。

在刀具使用次数管理等功能时，系统间可以共同使用公共宏变量。（包括回转控制）

FS16*i*

参数	6036	在刀架间使用公共宏变量的数量（100号组）
参数	6037	在刀架间使用公共宏变量的数量（500号组）

📖 在设定刀架上设定。

FS15*i*

		#7	#6	#5	#4	#3	#2	#1	#0
参数	7001	VR5	VR1						

#7(VR5) 在刀架间是否共同使用公共变量（#500～#999）
0 ： 不使用。
1 ： 使用。

#6(VR1) 在刀架间是否共同使用公共变量（#100～#199）
0 ： 不使用。
1 ： 使用。

● 从RS-232C接口读取程序时

从RS-232接口等外部装置读取程序时，要确认以下参数。

FS16*i*

		#7	#6	#5	#4	#3	#2	#1	#0
参数	3201		NPE						

#6(NPE)　0 ： 把M02，M30，M99作为程序登录结束。
　　1 ： 把M02，M30，M99不作为程序登录结束。

FS15*i*

		#7	#6	#5	#4	#3	#2	#1	#0
参数	2200					NPE			

#3(NPE)　0 ： 把M02，M30，M99作为程序登录结束。
　　1 ： 把M02，M30，M99不作为程序登录结束。

📖 如果不设定此参数，当读取程序时，若程序中途指令了M99等指令，则程序被分割成两部分。

使用变量

下面就变量的基本使用方法进行说明。

内容如下:

- 变量的种类
- 变量值的显示和设定
- 变量使用方法
- 未定义变量
- 决定变量的方法

变量的种类

● 变量的表示

●变量是以“＃”之后写上变量号的形式表示的。

📖 CNC程序是把G00和X100.0这样的地址与数值组合起来构成指令单位字。

📖 若只有变量号，调试较难。
使用控制出/入后，对于带有注释的程序就容易读了。

（例） ＃1＝＃2＋3（TOOL NUMBER）；

● 变量的种类

●根据变量号，宏变量可分为以下三大类。

变量的种类	用　途
局部变量	在各宏程序中独自使用的变量。 用于存储运算结果，代入调用宏程序的自变量等。
公共变量	在多个宏程序中，共同使用的变量。
系统变量	读、写当前位置的信息，刀具补偿量等CNC系统信息的变量。

🕮 使用公共变量时，为了与其他宏程序的变量号不重复，制成了变量表进行管理。

🕮 局部变量是宏程序固有的变量，可以自由使用。

● 变量号的范围

●各变量号的范围如下：

🕮 下表中是变量号范围的最大值，各装置中可以使用的变量号的范围随选择功能不同而不同。

变 量	变量号	备 注
局部变量	#1～#33	非保持型
公共变量	#100～#199	
	#500～#999	保持型
系统变量	#1000～	—

●非保持型变量，当切断电源时，其值被清除为空。
保持型变量，当切断电源时，保持变量值。

🕮 系统变量根据变量号分为非保持型变量和保持型变量。

●变量#146，#147，#149在用参数设定调用子程序的功能中可以使用。

● 复位处理

控制装置被复位时，各宏程序的变量状态如下。

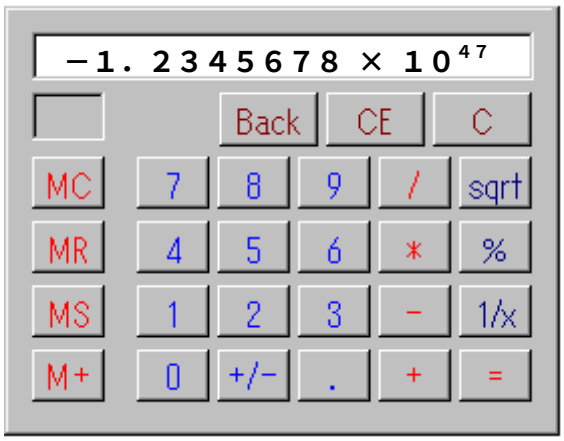
变 量		FS16*i*	FS15*i*
局部变量		参数6001#7=0时，初始化为空	必须初始化为空
公共变量	#100～	参数6001#6=0时，初始化为空	参数7000#6=1时，初始化为空
	#500～	被保持	
系统变量		（根据变量种类而不同）	

🕮 可以用MDI的 RESET 键，急停，程序中的M30和M02等操作复位。

●宏程序使用数值的范围

●有效位数

宏变量在CNC内部，用32位浮动小数点形式处理。
数值的有效位数换成10进制数为8位。



●变量值的范围

宏程序中可以使用的变量值范围如下：

正值	$+10^{-29}$～$+10^{47}$
0	0
负值	-10^{-29}～-10^{47}

🕮 如果超出上述范围，则出现111报警，报警灯亮。

●在表达式中可以使用的范围

在表达式中可以使用8位数的数值。

🕮 FS15*i*-MA 等机种，可以使用9位数的数值。

正值	0. 0000001～99999999.
0	0
负值	－0. 0000001～－99999999.

🕮 当 #1=123456780000000；时出现003报警。

当 #1=12345678*10000000；时可以执行。

变量值的显示和设定

●按以下操作顺序，在画面上可以显示宏程序变量的变量值。

●用MDI可以输入公共变量的变量值。

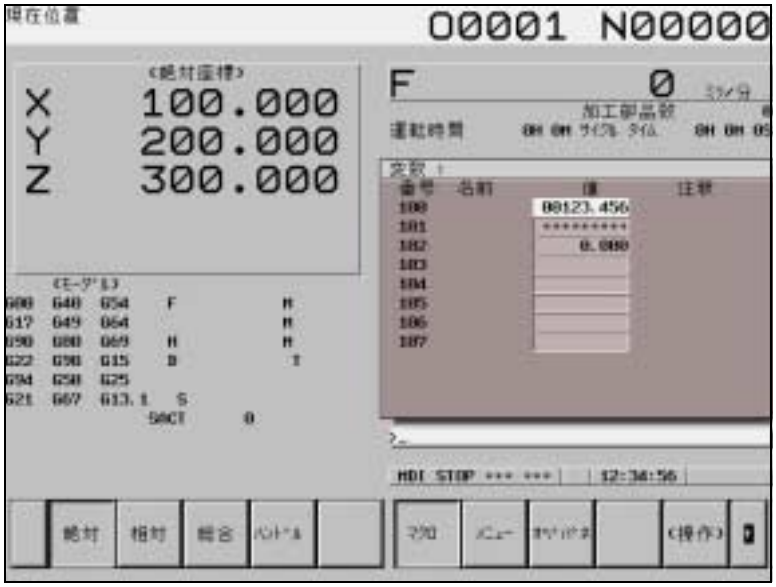
●FS16*i*

●显示宏变量画面

1. 按 OFFSET SETTING 键。

2. 按菜单键（Next）

3. 按软键 Marcro ，显示宏变量画面。



●变量值为空白状态时，意味空（未定义）。

●“ ******* ”表示在“ 画面上不能显示变量值 ”
变量值的绝对值超出99999999或者小于0.0000001时，可以显示。

🕮 宏变量值只是不显示，但可以正确地存储。

●宏变量的设定

●按以下操作顺序，用MDI可以设定宏变量值。

📖 用MDI键可以设定的只是公共变量。设定局部变量时，在MDI方式，输入类似“ #1=1； ”这样的程序后执行。

1. 输入要显示的变量号后，按软键 [No.search] 移动光标。

2. 用下列方法，输入宏变量值。

设定项目	操作方法
用MDI输入变量值	变量值 [INPUT]，（或者用软键 [INPUT] ）
输入相对坐标值	轴名称，软键 [C INPUT]
变量值为空	（什么都不输入），软键 [输入]

●在下列参数中，可以对宏变量的设定方法附加条件。

	#7	#6	#5	#4	#3	#2	#1	#0
参数 3290		MCM				MCY		

#6(MCM)　0 ： 任何运行方式都可输入宏变量值。
1 ： 只在MDI方式，可以输入宏变量值。

#2(MCY)　0 ： 用MDI可以输入宏变量值。
1 ： 用MDI不可以输入宏变量值。

📖 #6（MCM）在FS16-B以后的机种使用。

●FS15*i*

1. 按以下操作顺序，显示宏变量画面。

顺次按 〈 ，软键 SETTING ，软键 Chapter 软键 宏变量参数

> 变量值的绝对值超过99999999时，显示“上溢出”，不足0.0000001时显示“下溢出”。

2. 用下列操作，从MDI输入公共变量的变量值。

❶ 按机床操作面板的 MDI 键，运转方式置于MDI。

❷ 把机床操作面板的存储保护键  置成OFF。

❸ 输入显示变量号后，若按软键 输入变量号 则光标移到被指定的变量号上。

❹ 用下列方法输入公共变量的变量值。

功 能	操 作
输入变量值	变量值，软键 INPUT
原来值与输入的值相加	变量值，软键 +INPUT
把变量值置为空	软键 空INPUT
把公共变量全置为0	软键 全部清除 ，全部
把100组号的公共变量置0	软键 全部清除 ，公共变量 1
把500组号的公共变量置0	软键 全部清除 ，公共变量 2

●用MDI键的输入保护

如果设定了下面参数，从MDI不能输入500组号的公共变量。

参数		
	7031	要保护的宏变量的开头号-500

要保护510以后变量时，设定10。

参数		
	7032	要保护的宏变量个数（0～127）

变量的使用

● 表达式的书写方法

●在式子左侧是代入运算结果的变量号。

●式子右侧是运算式。

#1=#2+#3表达式是命令#2和#3相加的结果写入#1中。

🕮 此处的“ = ”意思不是数学上的等号，而是表示赋值的意思。

● 变量值的定义

●省略小数点时，为计算器型。

#1＝123 ； ..... #1

🕮 在NC语句中，省略小数点时（例X100）是否是计算器型，要根据参数设定来决定。

●式子中，使用数值的最大位数是8位。（有的机种是9位）

#2＝123. 45678 ； ..... #2

● 变量值的引用

●在程序中，引用变量值时，在地址字后面要指定变量号。

G01 X10. 0 F#1 ； ..... F

🕮 执行后，给变量号代入变量值。

●变量值按地址字的单位四舍五入后再引用。

G00 G90 X#2 ； ..... X的坐标值

🕮 此时地址字的单位是设定单位1/1000㎜。

T#2 ； ..... 刀具号

🕮 刀具号（T）和程序循环次数（L）等，使用变量时，经四舍五入为整数后再引用。

●引用的变量要进行运算时，用[ ]括起来。

G00 G90 X[#2/10] ; ..... X 的坐标值

当变更变量值的符号后使用时，不需要[ ]。

（例）G00 G90 X－#2 ;

● 间接指定

●用变量指定变量号时，称为“ 间接指定 ”。

●请参考以下例子。

刀具号为1～5的刀具使用次数存储在宏变量#501～#505中:

#501	刀具 1 的使用次数
#502	刀具 2 的使用次数
#503	刀具 3 的使用次数
#504	刀具 4 的使用次数
#505	刀具 5 的使用次数

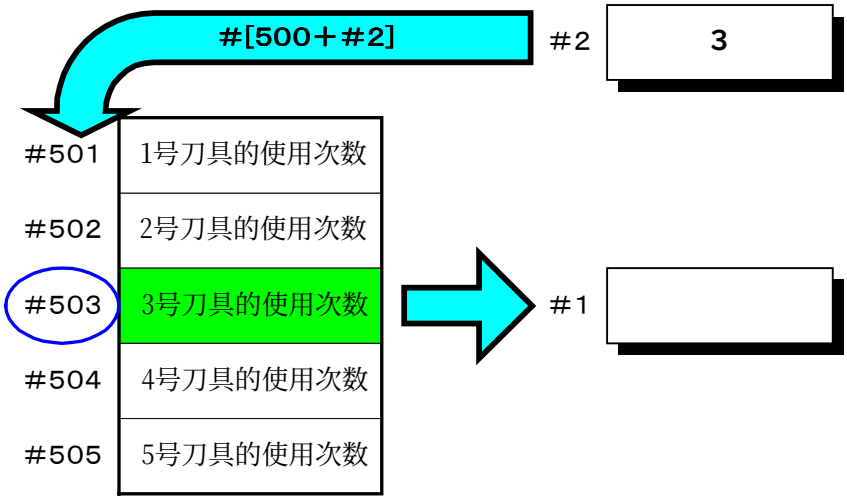
－ 当把 1 号刀具的使用次数读取到 #1 变量时 ..... #1＝#501 ;
－ 当把 2 号刀具的使用次数读取到 #1 变量时 ..... #1＝#502 ;
－ 当把 3 号刀具的使用次数读取到 #1 变量时 ..... #1＝#503 ;
－ 当把 4 号刀具的使用次数读取到 #1 变量时 ..... #1＝#504 ;
－ 当把 5 号刀具的使用次数读取到 #1 变量时 ..... #1＝#505 ;

读取n号刀具的使用次数时，#1变量可以写为：#1=#[500+n]。
（n=1,2...5）

当n用宏变量（例如 #2）置换时，则上式可以写成

#1=#[500+#2]。

如果#2的值为3，则3号刀具的使用次数（#503）的值存在#1中。



●未定义

●未定义的含义...

变量值未设定的状态称为“空”。

除了少数例子外，空和0是不同的。

●#0

通常#0是空的变量，不能代入值，它用于变量值的比较和置换上。

#500置为空时，#500=#0；

●使用注意事项

引用变量值时，如果变量值为空，则地址被忽视。

G00 G91 X#1 Y#2 ；时的各轴移动量如下表。

#1	#2	动 作
50. 0	100. 0	X轴，Y轴都移动
空	100. 0	只Y轴移动
50. 0	空	只X轴移动

●运算式的使用

●用空置换时，空和0不同。

#2＝#1 ；.....#1 空 → #2 空

#1 0 → #2 0

●运算式中，空与0相同。

#2＝#1＊5 ；.....#1 空 → #2 0

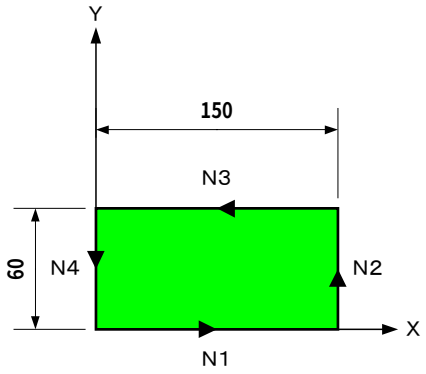
#1 0 → #2 0

● 最简单的宏程序

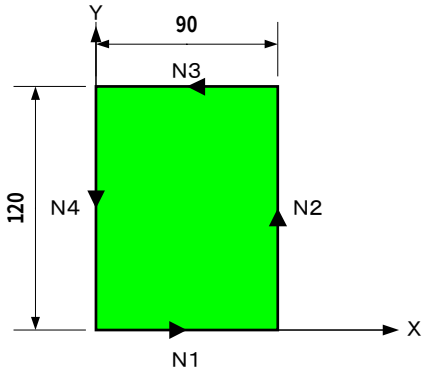
●把加工程序宏程序化的最简方法。
首先在手工编制中编制尺寸不同的2个程序。然后，在这两个程序中把不同的部分设成变量。

●在下面2个程序中，把不同的部分作上标记。

🕮 G91是增量（移动量）方式的指令。

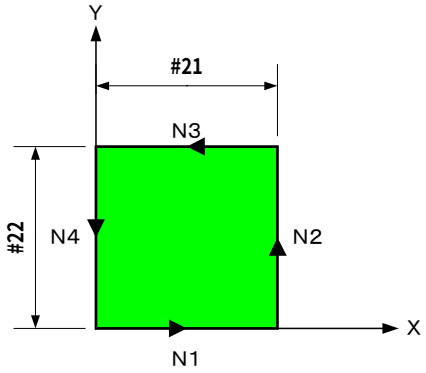



```
O9999 ;
N1 G01 G91 X150. 0 F300 ;
N2 Y60. 0 ;
N3 X－150. 0 ;
N4 Y－60. 0 ;
M30 ;
```




```
O9999 ;
N1 G01 G91 X90. 0 F300 ;
N2 Y120. 0 ;
N3 X－90. 0 ;
N4 Y－120. 0 ;
M30 ;
```

●如果2边的长度用宏变量#21,#22置换，则程序如下所示。




```
O9999 ;
#21＝90. 0 ;
#22＝120. 0 ;
N1 G01 G91 _____ F300 ;
N2 _____ ;
N3 _____ ;
N4 _____ ;
M30 ;
```

🕮 在实际运用中，#21和#22的值作为自变量使用，在主程序中，作为地址U和V的自变量。

使用运算命令

本章就运算命令的种类和使用方法进行说明。如果使用上取整、下取整，以及三角函数，则将扩大程序的应用范围。

内容如下：

- 下面说明运算式的种类和使用方法。

运算式的种类和使用方法

● 加减乘除

运算种类	运算符	运算式
加法	＋	＃1＝＃2＋＃3 ；
减法	－	＃1＝＃2－＃3 ；
乘法	＊	＃1＝＃2＊＃3 ；
除法	／	＃1＝＃2/＃3 ；

● 数值处理

运算种类	函数名	运算式
上取整	FIX	＃1＝FIX[＃2] ；
下取整	FUP	＃1＝FUP[＃2] ；
四舍五入	ROUND	＃1＝ROUND[＃2] ；
绝对值	ABS	＃1＝ABS[＃2] ；

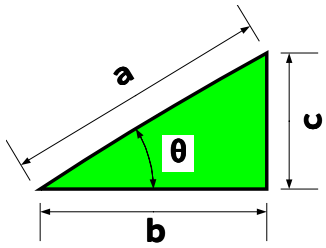
📖 经过舍取处理，把数值的绝对值变小或变大，称为下取整或上取整。

＃2：1．234时 、 ＃1＝FIX[＃2] ； …… ＃1

＃2：－1．234 时、 ＃1＝FUP[＃2] ； …. ＃1

● 三角函数

直角三角形（边长为a,b,c）的边长和角度的关系用下面公式可以求得。



运算种类	表达式	答案
正弦	#1＝SIN[θ]；	c/a
余弦	#1＝COS[θ]；	b/a
正切	#1＝TAN[θ]；	c/b
反正切	#1＝ATAN[c]／[b]；	θ
反正弦	#1＝ASIN[c/a]	
反余弦	#1＝ACOS[b/a]；	

🕮 角度θ的单位是度。（90度30分，表示为90.5度）

🕮 ASIN和ACOS在FS15和FS16-B以后的机种可以使用。

🕮 ATAN,ASIN,ACOS的结果在下面范围内。

函 数	FS16*i*		FS15*i*
	参 数 6004#0＝0	参 数 6004#0＝1	
ATAN	0～360	－180～＋180	－180～＋180
ASIN	270～90	－90～90	270～90
ACOS	180～0		180～0

#1＝SIN[90]； ..... #1

#1＝COS[90]； ..... #1

#1＝TAN[45]； ..... #1

#1＝ATAN[1]／[1]； ..... #1

#1＝ATAN[-1]／[-1]； ..... #1

● 其他函数

运算的种类	函数名	表达式	FS15 *i*	FS16 *i*
平方根	SQRT	#1＝SQRT[#2] ;	○	○
自然对数	LN	#1＝LN[#2] ;	○	△
以e为底的指数	EXP	#1＝EXP[#2] ;	○	△
余数	MOD	#1＝#2 MOD #3 ;	○	×

○:可以用　　×:不可以用　　△:对于可以使用的机种有限制。

📖 **FS16*i***
自然对数（LN）和指数（EXP）函数在FS16-B以后的机种可以使用。

📖 **FS16*i***
用下面的式子可以求出余数。

```
#1＝#2－FIX[#2/#3]＊#3 ;
```

当#2：2.0　时、#1＝SQRT[#2] ; ..... #1

● 逻辑运算

●在CNC内部使用2进制的数值。以2进制的为单位进行逻辑运算。

📖 10进制的21用8位2进制数表示为0010101。

●逻辑运算的输入值和运算结果（真值表）如下所示。

输　入		或 OR	异或 XOR	与 AND
0	0	0	0	0
0	1	1	1	0
1	0	1	1	0
1	1	1	0	1

📖 逻辑和(或)OR在把某位置于“1”时使用。

📖 按位加(“异或”)XOR在把某位反相时使用。

📖 逻辑积(与)AND在把某位置于“0”时使用。

●#2为21（2进制数为00010101），#3为12（2进制数是00001100）时各运算结果如下表。

值/表达式	10进制数	2进制数
#2	21	00010101
#3	12	00001100
#1＝#2 OR #3	29	00011101
#1＝#2 XOR #3	25	00011001
#1＝#2 AND #3	4	00000100

●与PMC（内装可编程控制器）执行数据传送时，使用变换数据形式命令。

功 能	函数名	使用例
从BCD到2进制的变换	BIN	#1＝BIN[#2] ;
从2进制到BCD的变换	BCD	#1＝BCD[#2] ;

🕮 BCD代码是二—十进制码。

● 省略函数名的输入

●在宏程序中使用函数时，只用函数名前头的2个字符，后面部分可以省略。

（例）ROUND ➔ RO
FIX ➔ FI 等等...

● 运算的顺序

●运算的优先顺序与通常的计算式相同。

① 函数

② 乘除运算（*，/，AND，MOD）

③ 加减运算（+，-，OR，XOR）

●改变运算顺序时，使用括号[ ]。
包括函数在内，括号可以使用到5重。

改变程序流程
（转移和循环）

本章讲述转移和循环的方法。在加工和操作作业中，有很多重复的动作。如果使用转移和循环，可以缩短宏程序。

现就以下三种语句进行说明：

- IF语句
- GOTO语句
- WHILE语句

条件转移 （IF 语句）

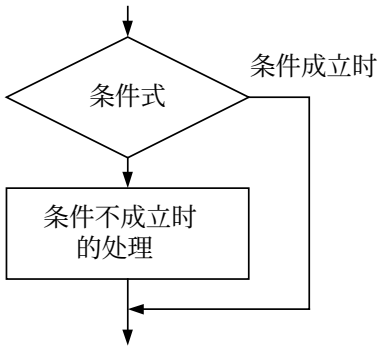
●功能

●根据条件，跳过程序的一部分后执行时，使用IF语句。

●当IF语句的条件成立时，转移到被指定的顺序号上。

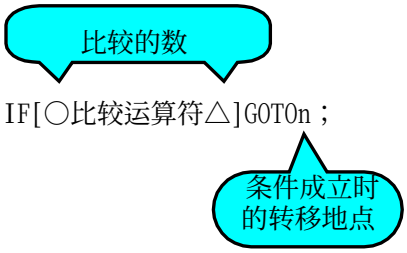
IF语句的条件式不成立时，进入下一个程序段。

●框图如下所示。



●程序的书写方法

●在IF后面的[ ]中，要书写条件式。



●在[ ]中，比较运算符（GE和LT等等）的两侧是进行比较的2个数值或者是式子，常数。

●在[ ]的右侧，GOTO的后面写着当条件成立时的转移目的地顺序号。

●比较运算符有以下几种。

比较运算符	意　义	英语拼音
EQ	＝	Equal(等于)
NE	≠	Not Equal(不等)
GT	＞	Greater Than(大于)
GE	≧	Greater or Equal(大于等于)
LT	＜	Less Than(小于)
LE	≦	Less or Equal(小于等于)

IF[#1 GT 10] GOTO1 ；

如果#1比10大，就转移到顺序号1，如果不大于10（包括等于10）时，进入下一个程序段。

GOTO N1，是错误的。

● 也可以使用下面的方法

●用变量也可以指定转移目的地顺序号。

```
IF[ ..... ] GOTO#1 ;
N1      ..............
          （#1 为 1 时的处理）
        GOTO9 ;
N2      ..............
          （#1 为 2 时的处理）
        GOTO9 ;
N3      ..............
          （#1 为 3 时的处理）
        GOTO9 ;
N9
```

●在条件式的后面，可以书写宏语句。

FS15　FS16-B以后的机种

```
IF[#1 EQ #2] THEN #3=1 ;
```

无条件转移 （GOTO语句）

●如果执行此程序，则无条件地转移到被指定的顺序号上。

●在程序中GOTO之后书写转移目的地顺序号。

```
GOTO1 ;
```

📖 GOTO N1；错误。

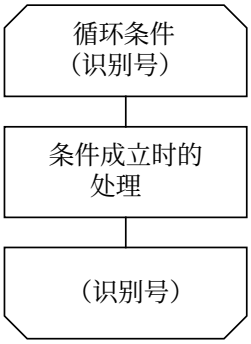
循环 （WHILE语句）

●功能

●反复执行处理的某一部分时，使用WHILE语句。

●条件成立时，执行从DO到END之间的程序。
条件不成立时，进入到END语句。

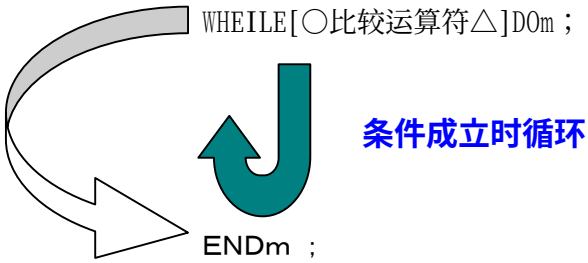
●框图如下。



●程序的书写方法

●在WHILE之后书写条件式。

📖 条件式的书写方法与IF语句相同。



条件不成立时，执行END语句

●条件式的后面，书写DO和识别号（1,2,3）。

●在循环的最后，书写END和识别号（和DO指定的识别号相同）。

📖 识别号（1,2,3）是和指定循环范围的。
可以使用1,2,3任意一个。

● 循环语句的使用

下面编制一个重复某一动作3次的程序。

1. 对动作重复的部分，进行编程。

```
G00 G91 X100.0 ;
Y100.0 ;
```

2. 决定记忆循环次数的变量（计数器）。

📖 在此 1#作为计数器使用。

3. 决定计数器的初始值。重要的是要明确使用计数器的目的。

变量的用途	初始值	备 注
加工完的工件数	0	
要求加工的工件数	1	
剩余的工件数	3	当要求3时

📖 在下面程序中，计数器为加工完的工件数的计数。

```
#1=0 ;

G00 G91 X100.0 ;
Y100.0 ;
```

4. 循环范围的开头要书写DO，最后写END。

```
#1=0 ;
DO1 ;
G00 G91 X100.0 ;
Y100.0 ;

END1 ;
```

📖 在此，识别号为1。

5. 在END前，写入计数器更新计数值的程序。

变量的用途	计数器的增减	表达式
加工完的工件器	＋1	＃1＝＃1＋1 ；
要求加工的工件数		
剩余的工件数	－1	＃1＝＃1－1 ；

```
＃1＝0 ；
DO1 ；
G00 G91 X100. 0 ；
Y100. 0 ；
＃1＝＃1＋1 ；
END1 ；
```

📖 此式子的意思是#1的值与1相加的和再代入#1中，也就是#1+1的意思。

6. 在DO之前编写WHILE语句，并写入重复条件。在条件式中，作为计数器应写上使用的变量号（本例中是#1）和目标循环次数。

```
＃1＝0 ；
WHILE[＃1    3] DO1 ；
G00 G91 X100. 0 ；
Y100. 0 ；
＃1＝＃1＋1 ；
END1 ；
```

7. 最后，决定比较运算符。
本例中，作为加工完了的工件数使用#1，通过WHILE语句后，#1的值如下所示。

次 数	#1的值	目标循环次数
第 1 次	0	3
第 2 次	1	
第 3 次	2	

此表中，循环的条件是“ #1比3小时 ”，所以比较运算符写为LT(Less Than)。

```
#1＝0 ；
WHILE[#1 LT 3] DO1 ；
G00 G91 X100. 0 ；
Y100. 0 ；
#1＝#1＋1 ；
END1 ；
```

●习题1

#1作为“ 要求加工的工件数 ”，修改程序。

●习题2

#1作为“ 剩余工件数 ”，修改程序。

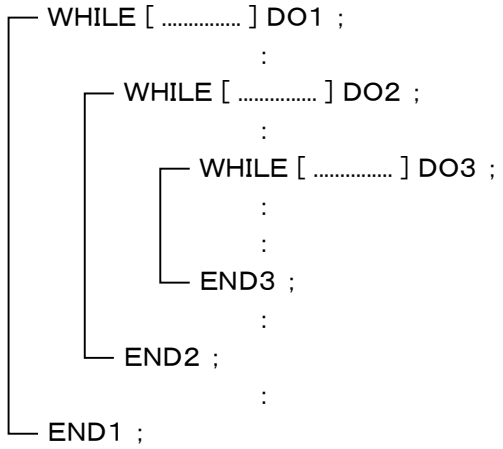
●总结

重复动作程序的基本形式如下所示，请牢记。

```
( 计数器 )＝( 初始值 )
while[( 计数器 ) 比较运算符 ( 目标值 )] DOm；
( 循环程序 )
( 计数器 )＝( 计数器 ) ± 1 ；
ENDm；
```

●识别号和循环语句的嵌套

●在使用了WHILE语句的循环体中，还可使用WHILE语句即可以使循环多重嵌套。
WHILE语句中，嵌套最多为3重。

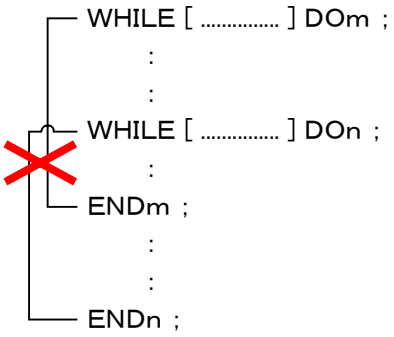


📖 识别号为1，2，3中任意一个。

●如果循环的范围不重叠的话，识别号使用几次都可以。

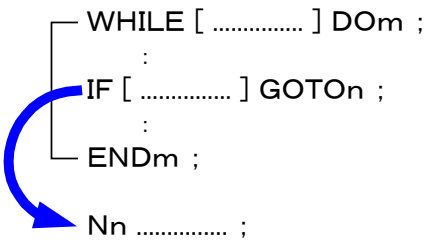
○:可以使用

×:不可使用

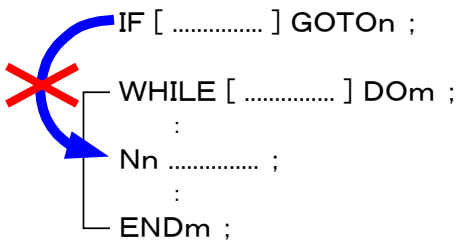


●使用IF语句，可以中止循环。

○:可以使用



×:不可使用



📖 在右侧的例中，虽然循环体中可以有转移，但执行END命令时，
出现报警。

●如果省略WHILE和条件式，则出现无限循环。

```
DOm ;
    :
    :
    :
ENDm ;
```

🕮 上述情况用于和IF语句组合，在多个条件中，结束循环时使用。

使用WHILE语句循环

●使用IF语句和WHILE语句都可以处理循环。但是使用IF语句的循环处理，在内部要检索顺序号，需要处理时间。

●使用IF语句的顺序号检索，如下所示。

① 从当前执行程序段向后检索顺序号。

② 至程序的尾部没有发现检索的顺序号时，继续从程序的开头检索。

③ 在程序内，没有发现要求的顺序号时，报警。

●原则上，循环用WHILE语句，当把程序的一部分转到下方时，使用IF语句。

● 未定义变量的使用

关于在条件式中未定义变量的使用如下所示。

比较运算符	处　理
EQ	空和0不同
NE	
GT	空和0相同
GE	
LT	
LE	

🕮 GT,GE,LT,LE时，在2个值相减的结果中，要判断大小，所以空和0同样用。

编 程

使用前面学习的宏程序的基本功能，编制深孔加工循环的宏程序。现在编制一个只用宏程序动作的程序。在学习了后面讲述的“ 调用宏程序 ”后，改成从加工程序中调用的形式。

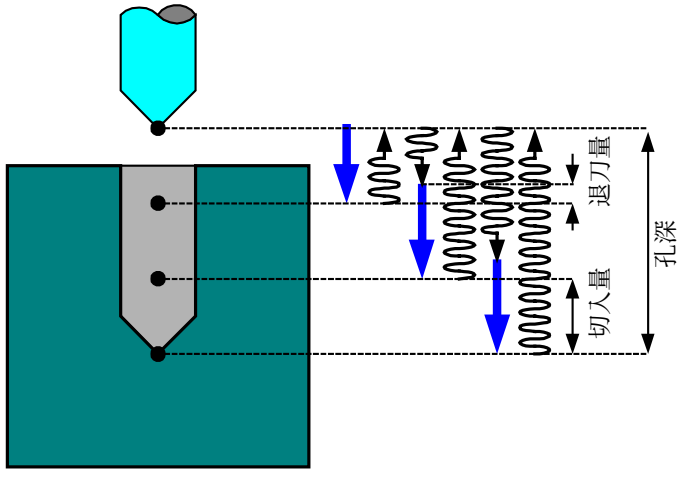
内容如下：

- 宏程序的简单编制方法（深孔循环）
- 宏程序的确认方法
- 典型程序（孔加工问题）

深孔加工循环

● 程序说明

●根据下面动作，编制加工孔的宏程序。



🕮 在车床（T）系的机床中，把图横过来考虑。

● 决定算法

●为了编制宏程序，需要决定称为“ 算法 ”的计算顺序。

●决定宏变量

此宏变量用于以下2方面:

— 定义加工尺寸时用

— 在宏程序内，用于计算的变量。

●定义加工尺寸时使用的变量决定如下：

基准点坐标（绝对）.....#18(R)

孔底坐标（绝对）.....#26(Z)

每次的切入量 < 0 .....#17(Q)

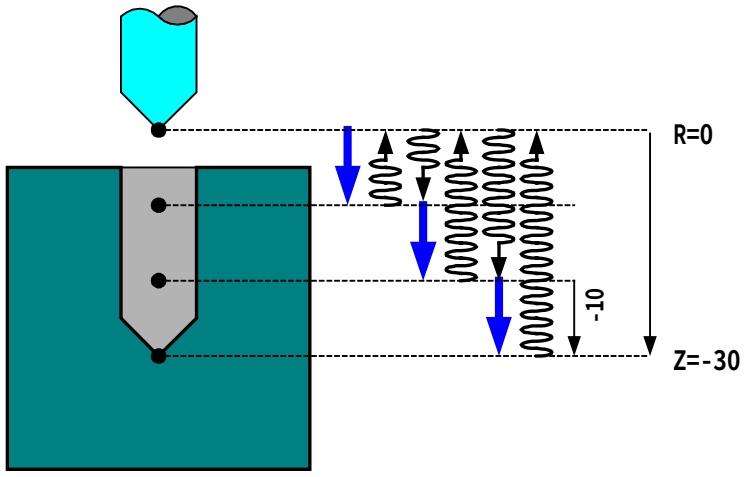
切入速度..... #9(F)

📖 定义加工尺寸的变量号，参考后面将学习的“ 调用宏程序 ”，使用从加工程序中很容易调用的变量号。（）中的字符是主程序调用时的自变量地址。

● 从手工编制到宏程序...

关于把手编的深孔加工循环变成宏程序的方法，说明如下。

1. 根据具体的孔加工数值，手编深孔加工程序。



为了编程简单，不使用“退刀量”，另外把加工起始点作为程序原点，用绝对指令（坐标值）编程序。

```
G01 G90 Z_____ F300 ; ..... 第 1 次孔加工
G00 Z_____ ;
G00 G90 Z_____ ; ..... 第 2 次孔加工
G01 Z_____ F300 ;
G00 Z_____ ;
G00 G90 Z_____ ; ..... 第 3 次孔加工
G01 Z_____ F300 ;
G00 Z_____ ;
```

📖 在车床系中，省略G90。

2. 从上面编好的程序中，找出重复的路线。此时，为了用2个程序段写出第一次孔加工的程序。要清楚不规则（不重复）的部分。从而设定了移动量为0的快速进给程序段，这样做可以使上述三次孔加工程序全部都可以用3个程序段构成。

```
G00 G90 Z0 ; ..... 追加
G01 G90 Z－10. 0 F300 ; ..... 第 1 次孔加工
G00 Z0 ;
G00 G90 Z－10. 0 ; ..... 第 2 次孔加工
G01 Z－20. 0 F300 ;
G00 Z0 ;
G00 G90 Z－20. 0 ; ..... 第 3 次孔加工
G01 Z－30. 0 F300 ;
G00 Z0 ;
```

3. 下面考虑移动量的规律性。

— 快速进给程序段的移动量，与在其前面的加工中孔的深度相同。

— 本次孔底坐标是前一次孔底坐标加上切入量（-10㎜）后的值。

次数	程序	本次孔底坐标	上次孔底坐标
第 1 次	G00 G90 Z0 ; G01 Z−10. 0 F300 ; G00 Z0	−10. 0	0
第 2 次	G00 G90 Z−10. 0 ; G01 Z−20. 0 F300 ; G00 Z0 ;	−20. 0	−10. 0
第 3 次	G00 G90 Z−20. 0 ; G01 Z−30. 0 F300 ; G00 Z0 ;	−30. 0	−20. 0

4. 现在，作为作业用的变量，使用本次孔底的坐标(#1)和前一次的孔底坐标(#2)。

孔加工时使用3次变量的情况归纳如下：

```
G00 G90 Z#2 ; ..... 用快速进给趋近
G01 Z#1 F300 ; ..... 用切削进给切入
G00 Z0 ; ..... 用快速进给返回
```

5. 用WHILE语句，把此程序改为循环。

```
#26=−30. 0 ; ..... 最后的孔底坐标
#17=−10. 0 ; ..... 1次切入量
#1=0 ; ..... 本次孔底坐标初始化
#2=0 ; ..... 前次孔底坐标初始化
WHILE[#1 GT #26] DO1 ; ..... 循环条件
#1=#1+#17 ; ..... 更新孔底坐标
G00 G90 Z#2 ; ..... 用快速进给趋近
G01 Z#1 F300 ; ..... 用切削进给切入
G00 Z0 ; ..... 用快速进给返回
#2=#1 ; ..... 存储本次孔底坐标
END1 ; ..... 循环结束
```

6. 当从基准点到孔底的距离不是切入量的倍数时，要考虑零数的情况。

例如，孔底坐标为-25㎜时，要变更手编程序中的哪一部分？

次数	程序	本次孔底坐标	上次孔底坐标
第 1 次	G00 G90 X0 ; G01 Z−10.0 F300 ; G00 Z0	−10.0	0
第 2 次	G00 G90 Z−10.0 ; G01 Z−20.0 F300 ; G00 Z0 ;	−20.0	−10.0
第 3 次	G00 G90 Z−20.0 ; G01 Z~~−30.0~~ −25.0 F300 ; G00 Z0 ;	~~−30.0~~ −25.0	−20.0

要判断本次孔底坐标(#1)是否超过最终孔底坐标(#26)，如果超过了，要置换成最终孔底坐标。

```
WHILE[#1 GT #26] DO1 ;
#1=#1+#17 ;
IF[#1 GE #26] GOTO1 ; ..... 确认孔底坐标
#1=#26 ; ..... 若太深了，置换为最终值
N1 G00 … ..... 附加顺序号
    :
END1 ;
```

7. 现在要追加退刀量。从快速进给的坐标中减去退刀量。但是，在第1次切入中，前一次的孔底坐标(#2)为0，因此只向正方向移动1个退刀量。所以第一次先把退刀量置于0，第二次以后，退刀量为有效。实际的退刀量为#3，如果指定退刀量为1㎜时，#3的初始值设为0，第2次以后的#3为1。

```
    :
#3=0 ; ..... 实际的退刀量
    :
WHILE[ … ] ;
    :
N1 G00 G90 Z[#2+#3] ; ..... 快速进给趋近工件，移动1个退刀量
    :
#2=#1 ;
#3=1.0 ; ..... 第2次以后，退刀量有效
END1 ;
```

8. 最后，把加工开始点作为程序原点编制宏程序。为了能指定基准点的坐标值，应进行修改。
现在，用#18(R)指定基准点坐标。

```
#18=3.0 ;  ..... 基准点坐标值
     :
#1=#18 ;  ..... 把最初的位置设在基准点上
#2=#18 ;  ..... （ 同上 ）
     :
WHILE[ ... ] ;
     :
N1 G00 G90 Z[#2+#3] ;
   G01 Z-#1 F300 ;
   G00 Z#18 ;  ..... 返回到基准点
     :
END1 ;
```

●宏程序

```
O9999 ;
     #26=-25.0 ;  ..... 孔底坐标
     #17=-10.0 ;  ..... 1次的切入量
     #18=3.0 ;  ..... 基准点坐标值
     #9=300 ;  ..... 切削速度
     #1=#18 ;  ..... 本次孔底坐标
     #2=#18 ;  ..... 前次孔底坐标
     #3=0 ;  ..... 实际退刀量
     WHILE[#1 GT #26] DO1 ;  ..... 循环到目标坐标
     #1=#1+#17 ;  ..... 计算本次孔底坐标
     IF[#1 GE #26] GOTO1 ;  ..... 确认切入是否过大
     #1=#26 ;  ..... 如果切入过大返回
N1   G00 G90 Z[#2+#3] ;  ..... 用快速进给接近
     G01 Z#1 F#9 ;  ..... 切孔
     G00 Z#18 ;  ..... 快速退回
     #2=#1 ;  ..... 存储孔深
     #3=1.0 ;  ..... 第2次以后退刀有效
     END1 ;  ..... 循环范围结束
     M30 ;  ..... 程序结束
```

编程要点...

把基本动作的程序准确地作出后，如果增加功能，程序是简单的。

确认宏程序的动作

现在对宏程序输入到CNC中以后确认动作的步骤进行说明。

主要流程如下:

顺序	作　业	补充说明
1	输入宏程序	在EDIT(编辑)方式，把宏程序输入到程序存储器中。
2	用单程序段确认	把机床操作面板的单程序段按钮置于ON，一边交替显示程序确认画面和宏变量画面，一边执行程序。
3	用连续运行确认	把机床操作面板的单程序段按钮置于OFF，然后确认宏程序的动作。

1. 在程序存储器上登录宏程序。

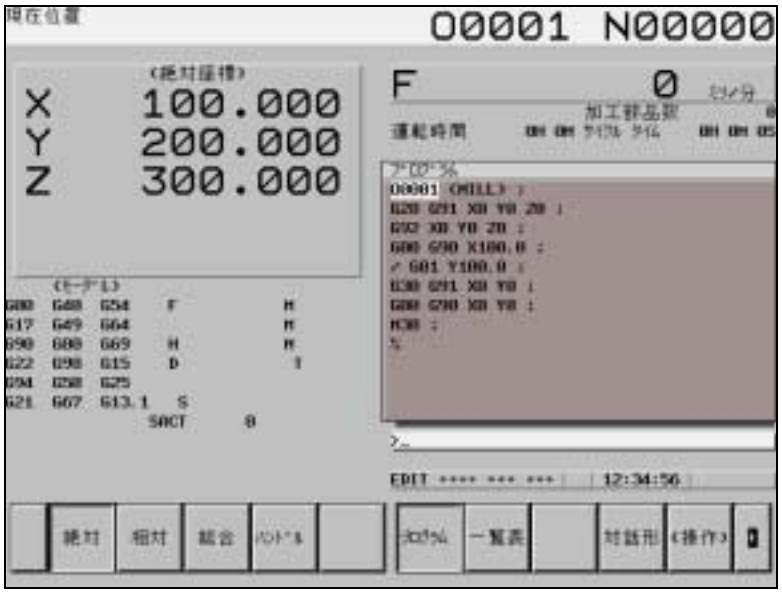
❶ 按机床操作面板的 

按钮，运转方式设置成EDIT(编辑)方式。

❷ 机床操作面板的存储器保护键  设置为OFF。

❸ 按 

键，显示程序画面。



❹ 顺次按 O 程序号 INSERT，登录程序。

📖 此时若在程序号后输入 EOB，则出现格式错。

📖 指定了已经登录了的程序号时，出现073报警。

❺ 输入1个程序段的程序，EOB，INSERT 依次输入各段程序。

📖 删除字时，把光标移到要删除的位置，然后按 DELETE。

📖 要变更字时，把光标移到要变更的字上，输入置换字后，按 ALTER。

❻ 输入程序到最后，然后按 RESET 键，把光标返到开头。

2. 把单程序段设置为ON，然后确认宏程序的动作。

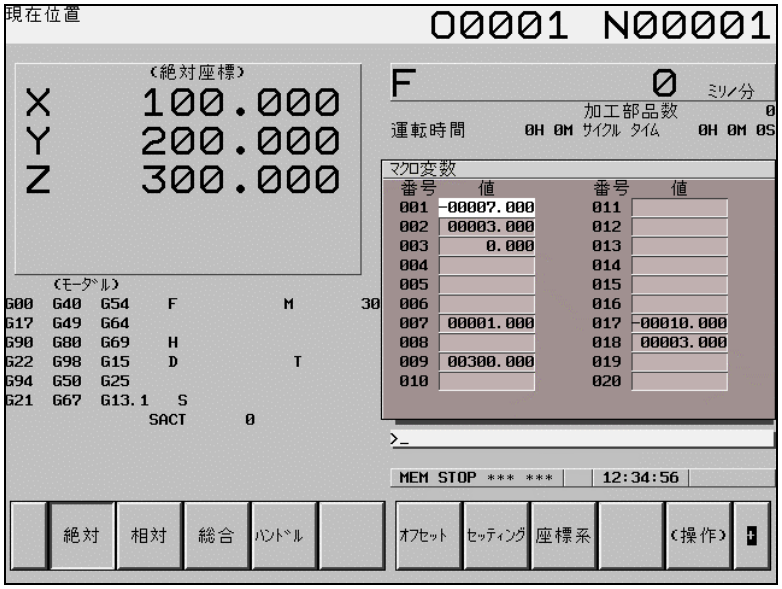
❶ 按机床操作面板的 MEMORY，把运行方式设置成存储器方式。

❷ 把机床操作面板的 SINGLE BLOCK 设置为ON。

❸ 输入程序号，软键 OSearch，选择欲确认的程序。

❹ 按机床操作面板的 CYCLE START，执行一个程序段的程序。

❺ 显示宏变量画面后，确认宏变量值。



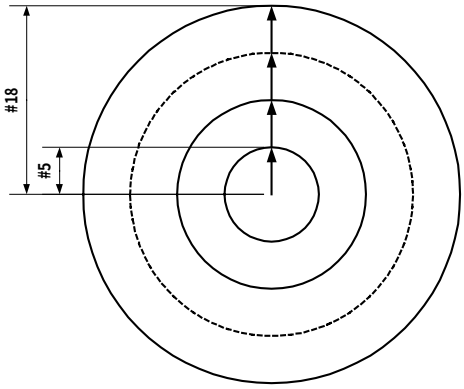
❻ 对于每个程序段，一边确认程序检查画面的光标位置，当前位置显示，宏变量的值，一边执行程序。

3. 把单程序段置于OFF，执行宏程序，确认一连串的动作。

圆形凹槽 （M系）

● 说明

●如图所示，从圆心开始加工，一面增加圆弧半径，一面进行圆加工，形成圆形凹槽，请编写圆凹槽的加工程序。



●使用的变量如下：

#5 ：1 次的切入量 .....（10. 0）

#18 ：最终圆的半径 .....（45. 0）

- 程序用增量指令编写。
- 无Z轴移动。
- 圆的中心作为X,Y轴的加工原点。
- 不考虑刀具半径补偿。

● 宏程序 （凹槽加工）

```
O9999 ;
      #5＝10. 0 ; ..... 圆半径增量
      #18＝45. 0 ; ..... 最终半径
      #1＝0 ; ..... 本次圆弧半径
      #2＝0 ; ..... 前次圆弧半径
      WHILE[#1 LE #18] DO1 ; ..... 循环到最终半径
      #1＝#1＋#5 ; ..... 计算本次圆弧半径
      IF[#1 LE #18] GOTO1 ; ..... 确认切入是否过量
      #1＝#18 ; ..... 用最终半径限制
N1    G01 G91 Y[          ] F300 ; ..... 切入
      G02 J－#1 ; ..... 加工圆
      #2＝#1 ; ..... 为了下一次存储圆半径
      END1; ..... 循环范围结束
      M30 ;
```

● 习题

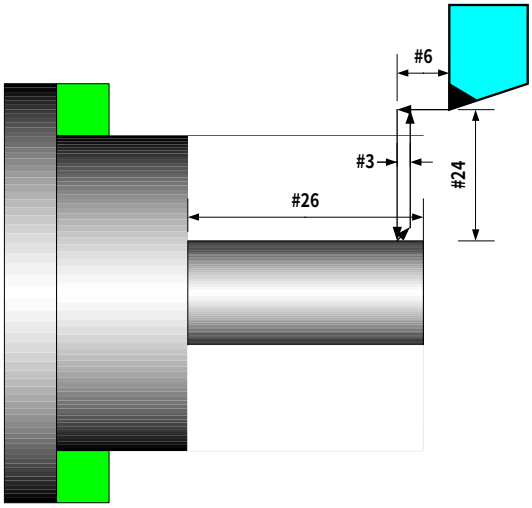
与深孔加工循环组合，请附加Z轴动作。

粗车削循环 （T系）

● 说明

●如图所示，请编制切削棒料的程序。

切削方向为X方向。用增量方式编写宏程序。



●使用的变量如下：

#3 ：退刀量
#6 ：切入量
#9 ：切削速度
#24 ：X方向切削余量
#26 ：Z方向切削余量

● 宏程序

```
O9999 ;
      #3=1.0 ; ..... 退刀量
      #4=0 ; ..... 实际退刀量
      #6=10.0 ; ..... 1次切入量
      #9=300.0 ; ..... 切削速度
      #24=20.0 ; ..... X方向切削余量
      #26=48.0 ; ..... Z方向切削余量
      WHILE[#26 GT 0] DO1 ; ..... 有剩余切削余量的循环
      #26=#26-#6 ; ..... 计算剩余切削余量
      IF[#26 GE 0] GOTO1 ; ..... 检查是否过切
      #6=#6+#26 ; ..... 若过切修正到最终值
N1    G00 W-[          ] ; ..... Z轴方向的移动
      G01 U-#24 F#9 ; ..... X轴的切入动作
      #4=#3 ; ..... 第2次以后退刀量有效
      G00 U[#4*2] W#4 ; ..... 退刀动作
      U[#24-#4*2] ; ..... X方向返回动作
      END1 ;
      M30 ;
```

清除变量

● 说明

●编一程序，把宏变量#500～#507清为空。

● 要点

●编一个不使用循环语句的程序

```
#500＝#0 ；
#501＝#0 ；
      :
#506＝#0 ；
#507＝#0 ；
```

●简化上面程序后，可以写成如下形式

```
#[n]＝#0 ；( n＝500,501,,,506,507 )
```

●把n换成#1，变成WHILE语句的循环形式。

● 宏程序

```
O9999 ；
    #1＝500 ；  ..... 变量初始化为空
    WHILE[#1 LE 507] DO1 ；  ..... 循环到变量507为止
    ＿＿＿＿＿＿＝#0 ；  ..... 清除#1变量为空
    #1＝#1＋1 ；  ..... 变量号+1
    END1 ；  ..... 循环范围结束
    M30 ；
```

求SIN 0～SIN 90

●说明

●求SIN0～SIN90，以10度为单位变化的值，写入变量#500～#509中。

●要点

●不使用循环的程序如下：

```
#500＝SIN[0] ;
#501＝SIN[10] ;
      :
#508＝SIN[80] ;
#509＝SIN[90] ;
```

●运算公式：

#[500＋n]＝SIN[n＊10] ；（ n＝0,1,,,8,9 ）

●把n换成#1，用循环语句控制，则程序如下。

●宏程序

```
O9999 ;
    #1＝0 ; ..... 把n初始化
    WHILE[#1 LE 9] DO1 ; ..... n从0到9循环
    #[          ]＝SIN[          ] ; ..... 求SIN值写入变量中
    #1＝#1＋1 ; ..... n+1
    END1 ; ..... 循环范围结束
    M30 ;
```

找出最大值

●说明

●在宏变量#500～#505中，事先设定常数值，如下所示。

#500 ： 30
#501 ： 60
#502 ： 40
#503 ： 80
#504 ： 20
#505 ： 50

●要求编制程序，从这些值中找出最大值，写入#506中。

●要点

❶ 开始假定#500的值为最大值，写入#506中
❷ 把#501与#506相比，若#501大，写入#506中
❸ 把#502与#506相比，若#502大，写入#506中
❹ 把#503与#506相比，若#503大，写入#506中
❺ 把#504与#506相比，若#504大，写入#506中
❻ 把#505与#506相比，若#505大，写入#506中

●宏程序

```
O9999 ;
      #1＝501 ; ..... 初始化要比较的变量号
      #506＝#500 ; ..... 假定#500的值最大
      WHILE[#1 LE 505] DO1 ; ..... 进行比较到#505为止
      IF[#506 GT #1] GOTO1; ..... 如果小，进入N1
      #506＝#1 ; ..... 若大时，变更最大值
N1    #1＝#1＋1 ; ..... 要比较的变量号+1
      END1 ;
      M30 ;
```

●习题

改成按数值大小顺序排列的程序。

例题答案

● 高速深孔加工循环

```
O9999 ;
      #26＝－25. 0 ;
      #17＝－10. 0 ;
      #18＝3. 0 ;
      #9＝300 ;
      #1＝#18 ;
      #2＝#18 ;
      #3＝0 ;
      WHILE[#1 GT #26] DO1 ;
      #1＝#1＋#17 ;
      IF[#1 GE #26] GOTO1 ;
      #1＝#26 ;
N1    G00 G90 Z[#2＋#3] ;
      G01 Z#1 F#9 ;
      G00 Z#18 ;
      #2＝#1 ;
      #3＝1. 0 ;
      END1 ;
      M30 ;
```

🕮 在车床系中，不用G90。

● 圆形凹槽（M）

```
O9999 ;
      #5＝10. 0 ;
      #18＝45. 0 ;
      #1＝0 ;
      #2＝0 ;
      WHILE[#1 LT #18] DO1 ;
      #1＝#1＋#5 ;
      IF[#1 LE #18] GOTO1 ;
      #1＝#18 ;
N1    G01 G91 Y[#1－#2] F300 ;
      G02 J－#1 ;
      #2＝#1 ;
      END1 ;
      M30 ;
```

●粗车循环（T）

```
O9999 ;
      #3＝1. 0 ;
      #4＝0 ;
      #6＝10. 0 ;
      #9＝300. 0 ;
      #24＝20. 0 ;
      #26＝48. 0 ;
      WHILE[#26 GT 0] DO1 ;
      #26＝#26－#6 ;
      IF[#26 GE 0] GOTO1 ;
      #6＝#6＋#26 ;
N1    G00 W－[#6＋#4] ;
      G01 U－#24 F#9 ;
      #4＝#3 ;
      G00 U[#4＊2] W#4 ;
      U[#24－#4＊2] ;
      END1 ;
      M30 ;
```

●清除变量

```
O9999 ;
      #1＝500 ;
      WHILE[#1 LE 507] DO1 ;
      #[#1]＝#0 ;
      #1＝#1＋1 ;
      END1 ;
      M30 ;
```

●求SIN0～SIN90

```
O9999 ;
      #1＝0 ;
      WHILE[#1 LE 9] DO1 ;
      #[500＋#1]＝SIN[#1＊10] ;
      #1＝#1＋1 ;
      END1 ;
      M30 ;
```

●找出最大值

```
O9999 ;
      #1＝501 ;
      #506＝#500 ;
      WHILE[#1 LE 505] DO1 ;
      IF[#506 GT #[#1]] GOTO1 ;
      #506＝#[#1] ;
N1    #1＝#1＋1 ;
      END1 ;
      M30 ;
```

●按数值大小顺序排列

```
O9999 ;
      #2＝501 ;
      WHILE[#2 LE 505] DO2 ;
      #1＝#2 ;
      #506＝#[#2－1] ;
      WHILE[#1 LE 505] DO1 ;
      IF[#506 GT #[#1]] GOTO1 ;
      #506＝#[#1] ;
      #[#1]＝#[#2－1] ;
      #[#2－1]＝#506
N1    #1＝#1＋1 ;
      END1 ;
      #2＝#2＋1 ;
      END2 ;
      M30 ;
```

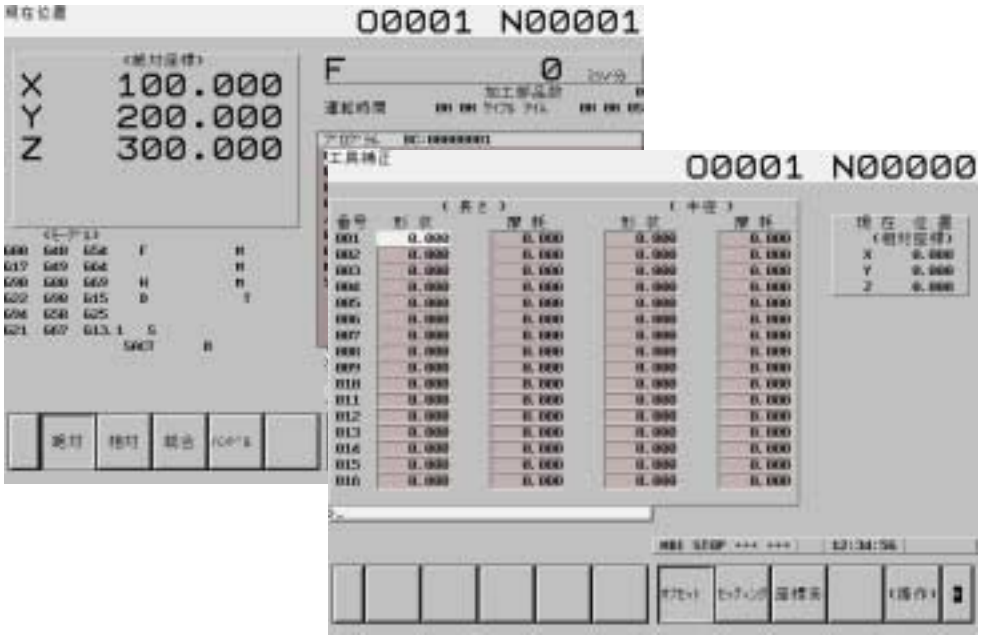
系统变量

使用系统变量可以读取机床当前位置和变更刀具补偿量。

现在就下列内容进行说明:

- **FS16 *i*-M 的系统变量**
- **各机种的系统变量表**

●在CNC上可以显示程序检查画面，SETTING画面，刀具补偿画面等各种画面，并可以读取加工时需要的信息。



●为了从宏程序中读、写当前位置信息、模态信息、刀具补偿等信息，CNC为用户设计了系统变量。

在此就 FS16 *i*-M 进行说明

本节关于FS16 ***i***-M的系统变量进行说明，可用于其它系统。其他机种与此没有太大的不同。关于其他机种的系统变量请见下一节的“系统变量表”。

●刀具补偿量

●使用系统变量，可以读取和写入刀具补偿量。

●刀具补偿存储器的形式不同，使用的系统变量号也不同。

●刀具补偿存储器 A

补偿号	系统变量
1 : (200) : 999	#10001 (#2001) : (#2200) : #10999

●刀具补偿存储器 B

补偿号	形状补偿	磨损补偿
1 : (200) : 999	#11001 (#2201) : (#2400) : #11999	#10001 (#2001) : (#2200) : #10999

●刀具补偿存储器 C

补偿号	刀具长度补偿(H)		刀具半径补偿(D)	
	形状补偿	磨损补偿	形状补偿	磨损补偿
1 : (200) : 999	#11001 (#2201) : (#2400) : #11999	#10001 (#2001) : (#2200) : #10999	#13001 : : : #13999	#12001 : : : #12999

📖 最大刀具补偿号小于200时，可以使用#2001～#2400，但是刀具补偿C不能读取和写入半径补偿量。

🕮 若使用参数6004#5(D15)，则可以使用#2401～#2600和
#2601～#2800进行刀具半径补偿(D)。

FS16*i*

如果设定下列参数，可以把刀具补偿的系统变量号与FS15 ***i*** 设置同样的号。

🕮 此功能对FS16-B以后的机种可以使用。

		#7	#6	#5	#4	#3	#2	#1	#0
参数	6000					V15			

#3(V15)　0 ： 刀具补偿用系统变量是FS16用的号码。
1 ： 刀具补偿用系统变量是FS15用的号码。

当参数设定为1时，刀具补偿用的系统变量如下所示。

● 刀具补偿存储器 B

补偿号	形状补偿	磨损补偿
1 : 999	#10001 : #10999	#11001 : #11999

● 刀具补偿存储器 C

补偿号	刀具长度补偿(H)		刀具半径补偿(D)	
	形状补偿	磨损补偿	形状补偿	磨损补偿
1 : 999	#10001 : #10999	#11001 : #11999	#12001 : #12999	#13001 : #13999

🕮 #2001～#2400的系统变量不能使用。

● 给CNC设置报警

系统变量	设定范围	功 能
#3000	0～200	给CNC设置报警

●把0～200的值代入#3000，CNC变成报警状态，自动运行停止。

●在表达式后面（）中，可以写上26个字符以内的报警信息。

●在报警画面上，显示报警号和报警信息。报警号为3000与#3000中的值相加的和，报警信息是（）中的内容。

(例) #3000＝1（ERROR）；

➔ 在CNC画面上显示“报警 3001 ERROR”。

● 时钟信息

使用下面系统变量，可以读、写日期和时刻。

系统变量	单位	功 能	写 入
#3001	毫秒	是计数的计时器。 电源接通时，设为0。 计数至2147483647毫秒后返回到0。	可以
#3002	小时	对循环起动灯亮的时间进行计数。 即使切断电源，其值也保持。 计数至9544.371767小时后返回到0。	可以
#3011	年.月.日	可以读取日期。 (例)1999年12月31日→19991231	不可以
#3012	时.分.秒	可以读取当前的时刻。 (例) 12时34分56秒→123456	不可以

🕮 计时器精度16毫秒。

● 控制自动运行状态

#3003	单程序段	辅助功能结束	备 注
0	有效	等待	复位时
1	无效	等待	
2	有效	不等待	
3	无效	不等待	

🕮 如果单程序段停止无效，则即使机床操作面板按钮为ON,单程序段功能也无效。

🕮 若辅助功能结束信号为“不等待”，则不等待辅助功能结束信号就进入下个程序段。
另外，从CNC给PMC的分配完了信号(DEN)也不输出。

#3004	进给暂停	倍 率	准确定位	备注
0	有效	有效	有效	复位时
1	无效	有效	有效	
2	有效	无效	有效	
3	无效	无效	有效	
4	有效	有效	无效	
5	无效	有效	无效	
6	有效	无效	无效	
7	无效	无效	无效	

🕮 进给暂停无效时，动作如下:

— 若连续按进给暂停按钮时，则在单程序段状态，机床停止。
但#3003中，如果单程序段无效时，不停止。

— 若按了进给暂停按钮就松开时，进给暂停指示灯亮，但机床不停止。继续执行程序，在进给暂停有效的第一个程序段停止。

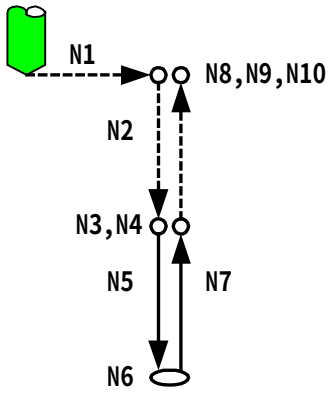
🕮 若倍率设置无效，则机床操作面板上的切削进给倍率开关无效。
切削进给倍率固定在100%。

🕮 准确定位无效是指:即使在切削程序段以外，也不进行准确停。
（到位确认是指:确认定位完了后，再进入下个程序段的功能。）

●用于攻丝加工的例子

●执行CNC的攻丝循环(G84)时，攻丝中的单程序段方式和进给暂停操作无效，切削进给倍率固定在100%上。

●在宏程序中，编制相当于攻丝循环的动作时；可以很方便地编制与CNC功能同样的单程序段无效的程序。



📖 参数6000#5(SBM)置于0后，确认动作。

```
O9999 ;
N1    G00 G91 X#24 Y#25 ; ..... X,Y轴在孔心位置上定位
N2    Z#18 ; ..... 趋近Z轴
      G04 ; ..... 用G04延迟读下个程序段
N3    #3003=3 ; ..... 单程序段等无效
N4    #3004=7 ; ..... 进给暂停等无效
N5    G01 Z#26 F#9 ; ..... Z轴下降
N6    M04 ; ..... 主轴反转
N7    G01 Z－[ROUND[#18]＋ROUND[#26]] ;
                                ..... Z轴返回
      G04 ; ..... 用G04延迟读下个程序段
N8    #3004=0 ; ..... 进给暂停等有效
N9    #3003=0 ; ..... 单程序段等有效
N10   M03 ;
          :
```

● SETTING信息

系统变量	功 能	写 入
#3005	可以写入输入单位(公、英制)设定状态等SETTING信息。	可以

							512	256
SETTING	–	–	–	–	–	–	FCV	–

	128	64	32	16	8	4	2	1
SETTING	–	–	SEQ	–	–	INI	ISO	TVC

FCV ： 使用FS15纸带格式变换
SEQ ： 自动插入顺序号
INI ： 设置英制输入
ISO ： 设置数据输出代码为ISO码/EIA码
TVC ： 进行TV校验

🕮 设定值用10进制数。

● 镜像

系统变量	功 能	写入
#3007	可以知道各轴镜像的状态。 若设定了SETTING画面的镜像和镜像信号中任意一个时，该轴的值就为1。	不可以

镜像 #3007	128	64	32	16	8	4	2	1
	第8轴	第7轴	第6轴	第5轴	第4轴	第3轴	第2轴	第1轴

0 ： 镜像无效。
1 ： 镜像有效。

🕮 设定值用10进制数。
FS16-B以后的机种可以使用。

● 加工零件数 (选择功能)

系统变量	功 能	写入
#3901	加工零件数	可以
#3902	要求的加工零件数	可以

● 模态信息

系统变量	模态信息
#4000	主程序号
#4001	G00,G01,G02,G03,G33, G60,G75,G77～G79
#4002	G17,G18,G19
#4003	G90,G91
#4004	G22,G23
#4005	G94,G95
#4006	G20,G21
#4007	G40,G41,G42
#4008	G43,G44,G49
#4009	G73,G74,G76,G80～G89
#4010	G98,G99
#4011	G50,G51
#4012	G66,G67
#4013	G96,G97
#4014	G54～G59
#4015	G61～G64
#4016	G68,G69
#4017	G15,G16
#4018	G40.1,G41.1,G42.1
#4019	G25,G26
#4020	G160,G161
#4021	G12.1,G13.1
#4022	G50.1,G51.1

系统变量	模态信息
#4102	B 代码
#4107	D 代码
#4109	F 代码
#4111	H 代码
#4113	M 代码
#4114	顺序号
#4115	程序号
#4119	S 代码
#4120	T 代码
#4130	选择的附加工件 坐标系

📖 #4000在FS16-B以后的机种中可以使用。

● 位置信息

在下面的系统变量中，可以读取位置信息。

🕮 不能写入。

系统变量	位置信息	坐标系	刀具补偿
#5001～8	程序段终点位置	绝对坐标	不含
#5021～8	当前位置	机械坐标	含
#5041～8		绝对坐标	
#5061～8	跳转信号位置		
#5081～8	刀具长度补偿量	—	—
#5101～8	伺服位置偏差量	—	—

🕮 变量号的第1位数字是轴号。

🕮 跳转进给(G31)的程序段终点位置如下:

— 输入了跳转信号时　　：是跳转信号ON的位置。

— 没有输入跳转信号时　：是被指令的程序段终点位置。

🕮 跳转信号位置(#5601～)只在相应的移动指令轴上设定。
例: G31 ×100.0 Ff，只有#5061被设定，#5062～#5068保持原来值不变。

🕮 刀具长度补偿量是当前使用中的值。

● 工件原点偏移 (选择功能)

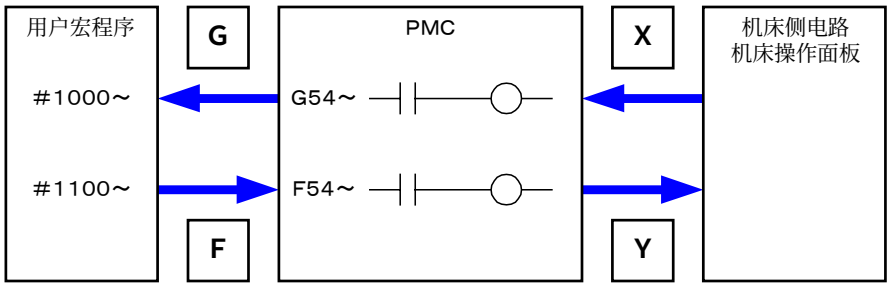
系统变量	工件坐标系	写入
#5201～8	第 1 轴～第 8 轴 外部工件原点偏移	可以
#5221～8	第 1 轴～第 8 轴 工件原点 G54	可以
#5241～8	第 1 轴～第 8 轴 工件原点 G55	可以
#5261～8	第 1 轴～第 8 轴 工件原点 G56	可以
#5281～8	第 1 轴～第 8 轴 工件原点 G57	可以
#5301～8	第 1 轴～第 8 轴 工件原点 G58	可以
#5321～8	第 1 轴～第 8 轴 工件原点 G59	可以

🕮 变量号的第1位数字是轴号。

🕮 使用附加工件坐标系时，系统变量为#7001～（追加48组时），#14001～（追加300组时）。

●宏程序与PMC间的信号

在宏程序和PMC(内装可编程控制器)之间，可以相互进行信号的读写。



📖 为了使用数量#1000～和#1100～，需要编制PMC的顺序程序。

●PMC ➔ 宏程序的信号

		#7	#6	#5	#4	#3	#2	#1	#0
地址	G054	UI7	UI6	UI5	UI4	UI3	UI2	UI1	UI0
	变量号	#1007	#1006	#1005	#1004	#1003	#1002	#1001	#1000
地址	G055	UI15	UI14	UI13	UI12	UI11	UI10	UI9	UI8
	变量号	#1015	#1014	#1013	#1012	#1011	#1010	#1009	#1008

📖 #1000～#1015的变量值是0或1。

📖 #1032上的16位信号一起读取。
变量范围是0～65535。

●宏程序 ➔ PMC 的信号

		#7	#6	#5	#4	#3	#2	#1	#0
地址	F054	UO7	UO6	UO5	UO4	UO3	UO2	UO1	UO0
	变量号	#1107	#1106	#1105	#1104	#1103	#1102	#1101	#1100
地址	F055	UO15	UO14	UO13	UO12	UO11	UO10	UO9	UO8
	变量号	#1115	#1114	#1113	#1112	#1111	#1110	#1109	#1108

📖 #1100～#1115的变量值为0或1。

📖 #1032上的16位信号一起读取。
变量范围是0～65535。

●使用#1133可以把数值用带符号的32位2进制数一次写入PMC中。

		#7	#6	#5	#4	#3	#2	#1	#0
地址	F056	UO107	UO106	UO105	UO104	UO103	UO102	UO101	UO100
	F057	UO115	UO114	UO113	UO112	UO111	UO110	UO109	UO108
	F058	UO123	UO122	UO121	UO120	UO119	UO118	UO117	UO116
	F059	UO131	UO130	UO129	UO128	UO127	UO126	UO125	UO124

🕮 变量值的范围为 -99999999～99999999。

系统变量表 （FS16 *i*-M）

●接口信号

变量	输入信号
#1000 : #1015	UI000 信号 : UI015 信号
#1032	UI000～UI015 （ 16 位一起处理）

变量	输出信号
#1100 : #1115	UO000 : UO015
#1132	UO000～UO015 （ 16 位一起处理）
#1133	UO100～IO131 （ 32 位一起处理）

●刀具补偿量

刀具补偿存储器 A		刀具补偿存储器 B		
补偿号	变量号	补偿号	形状补偿	磨损补偿
1 : （200） : 999	#10001 （#2001） （#2200） : #10999	1 : （200） : 999	#11001 （#2201） （#2400） : #11999	#10001 （#2001） （#2200） : #10999

刀具补偿存储器 C				
补偿号	刀具长度补偿（H）		刀具半径补偿（D）	
	形状补偿	磨损补偿	形状补偿	磨损补偿
1 : （200） : 999	#11001 （#2201） （#2400） : #11999	#10001 （#2001） （#2200） : #10999	#13001 : : : #13999	#12001 : : : #12999

●报警、SETTING、计时器

变 量	功 能	变 量	功 能
#3000	宏程序报警	#3011	时钟（年月日）
#3001	计时器[毫秒]	#3012	时钟（时分秒）
#3002	计时器[小时]	#3901	加工零件数
#3005	SETTING	#3902	所需零件数
#3007	镜像	—	—

●自动运行控制

#3003	单程序段	辅助功能结束
	有效	等待
1	无效	等待
2	有效	不等待
3	无效	不等待

#3004	进给暂停	倍率	准确停
0	有效	有效	有效
1	无效	有效	有效
2	有效	无效	有效
3	无效	无效	有效
4	有效	有效	无效
5	无效	有效	无效
6	有效	无效	无效
7	无效	无效	无效

●模态数据

变 量	模态数据
#4000	主程序号
#4001	G00,G01,G02,G03, G33,G60,G75, G77～G79
#4002	G17,G18,G19
#4003	G90,G91
#4004	G22,G23
#4005	G94,G95
#4006	G20,G21
#4007	G40,G41,G42
#4008	G43,G44,G49
#4009	G73,G74,G76, G80～G89
#4010	G98,G99
#4011	G50,G51
#4012	G66,G67
#4013	G96,G97
#4014	G54～G59
#4015	G61～G64

变 量	模态数据
#4016	G68,G69
#4017	G15,G16
#4018	G40.1,G41.1, G42.1
#4019	G25,G26
#4020	G160,G161
#4021	G12.1,G13.1
#4022	G50.1,G51.1
#4102	B 代码
#4107	D 代码
#4109	F 代码
#4111	H 代码
#4113	M 代码
#4114	顺序号
#4115	程序号
#4119	S 代码
#4120	T 代码
#4130	附加零件坐标系

●位置数据

变 量	意 义	坐标系	刀具补偿
#5001～8	程序段终点位置	工件坐标系	不考虑
#5021～8	当前位置	机械坐标系	考虑
#5041～8		工件坐标系	
#5061～8	跳转信号位置		
#5081～8	刀具长度补偿	—	—
#5101～8	伺服位置偏差	—	—

●工件坐标系原点

基本 6 组		附加 54 组		附加 300 组	
#5201～8	外部	#7001～8	G54.1 P1	#14001～8	G54.1 P1
#5221～8	G54	#7021～8	G54.1 P2	#14021～8	G54.1 P2
#5241～8	G55	#7041～8	G54.1 P3	#14041～8	G54.1 P3
#5261～8	G56	:	:	:	:
#5281～8	G57	#7901～8	G54.1 P46	#19941～8	G54.1 P298
#5301～8	G58	#7921～8	G54.1 P47	#19961～8	G54.1 P299
#5321～8	G59	#7941～8	G54.1 P48	#19981～8	G54.1 P300

系统变量表 （FS16 *i*-T）

●接口信号

变 量	输入信号
#1000 : #1015	UI000 信号 : UI015 信号
#1032	UI000～UI015 （ 16 位一起处理）

变量	输出信号
#1100 : #1115	UO000 : UO015
#1132	UO000～UO015 （ 16 位一起处理）
#1133	UO100～IO131 （ 32 位一起处理）

●刀具补偿量

补偿量	X轴补偿量		Z轴补偿量		刀尖R补偿量		假想刀尖	Y轴补偿量	
	磨损	形状	磨损	形状	磨损	形状		磨损	形状
1 : 49 : 64	#2001 : #2049 : #2064	#2701 : #2749 － －	#2101 : #2149 : #2164	#2801 : #2849 － －	#2201 : #2249 : #2264	#2901 : #2949 － －	#2301 : #2349 : #2364	#2401 : #2449 － －	#2451 : #2499 － －

补偿量	X轴补偿量		Z轴补偿量		刀尖R补偿量		假想刀尖	Y轴补偿量	
	磨损	形状	磨损	形状	磨损	形状		磨损	形状
1 : 99	#10001 : #10099	#15001 : #15099	#11001 : #11099	#16001 : #16099	#12001 : #12099	#17001 : #17099	#13001 : #13099	#18001 : #18099	#19001 : #19099

●报警、SETTING、计时器

变 量	功 能	变 量	功 能
#3000	宏程序报警	#3011	时钟（年月日）
#3001	计时器[毫秒]	#3012	时钟（时分秒）
#3002	计时器[小时]	#3901	加工零件数
#3005	SETTING	#3902	所需零件数
#3007	镜像	—	—

●自动运行控制

#3003	单程序段	辅助功能结束
0	有效	等待
1	无效	等待
2	有效	不等待
3	无效	不等待

#3004	进给暂停	倍率	准确停
0	有效	有效	有效
1	无效	有效	有效
2	有效	无效	有效
3	无效	无效	有效
4	有效	有效	无效
5	无效	有效	无效
6	有效	无效	无效
7	无效	无效	无效

●模态数据（G代码体系A）

变 量	模态数据
#4000	主程序号
#4001	G00,G01,G02,G03, G32～G36, G71～G74, G90,G92,G94
#4002	G96,G97
#4003	
#4004	G68,G69
#4005	G98,G99
#4006	G20,G21
#4007	G40,G41,G42
#4008	G25,G26
#4009	G22,G23
#4010	G80～G89
#4011	
#4012	
#4013	

变 量	模态数据
#4014	G54～G59
#4015	
#4016	G17,G18,G19
#4017	
#4018	
#4019	
#4020	G50.2,G51.2
#4021	G12.1,G13.1
#4022	
#4109	F 代码
#4113	M 代码
#4114	顺序号
#4115	程序号
#4119	S 代码
#4120	T 代码

●位置数据

变 量	意 义	坐标系	刀具补偿
#5001～8	程序段终点位置	工件坐标系	不考虑
#5021～8	当前位置	机械坐标系	考虑
#5041～8		工件坐标系	
#5061～8	跳转信号位置		
#5081～8	刀具位置补偿	—	—
#5101～8	伺服位置偏差	—	—

●工件坐标系原点

基本 6 组	
#5201～8	外部
#5221～8	G54
#5241～8	G55
#5261～8	G56
#5281～8	G57
#5301～8	G58
#5321～8	G59

系统变量 （FS15 *i*-M ）

●接口信号

变 量	输入信号
#1000 : #1015	UI000 信号 : UI015 信号
#1032	UI000～UI015

变 量	输出信号
#1100 : #1115	UO000 : UO015
#1132	UO000～UO015
#1133	UO100～IO131

●刀具补偿量

200组以下				
补偿号	刀具长度补偿（H）		刀具半径补偿（D）	
	形状补偿	磨损补偿	形状补偿	磨损补偿
1 : 200	#2001 : #2200	#2201 : #2400	#2401 : #2600	#2601 : #2800

刀具补偿存储器 A		刀具补偿存储器 B		
补偿号	变量号	补偿号	形状补偿	磨损补偿
1 : 999	#10001 : #10999	1 : 999	#10001 : #10999	#11001 : #11999

刀具补偿存储器 C				
补偿号	刀具长度补偿（H）		刀具半径补偿（D）	
	形状补偿	磨损补偿	形状补偿	磨损补偿
1 : 999	#10001 : #10999	#11001 : #11999	#12001 : #12999	#13001 : #13999

●报警、SETTING、计时器

变 量	功 能	变 量	功 能
#3000	宏程序报警	#3011	时钟（年月日）
#3001	计时器[毫秒]	#3012	时钟（时分秒）
#3002	计时器[小时]	#3901	加工零件数
#3005	SETTING	#3902	所需零件数
#3007	镜像	—	—

●自动运行控制

#3003	单程序段	辅助功能结束
0	有效	等待
1	无效	等待
2	有效	不等待
3	无效	不等待

#3004	进给暂停	倍率	准确停
0	有效	有效	有效
1	无效	有效	有效
2	有效	无效	有效
3	无效	无效	有效
4	有效	有效	无效
5	无效	有效	无效
6	有效	无效	无效
7	无效	无效	无效

●模态数据

模态信息	前一个程序段	执行中
G 代码(00组) : G 代码(25组)	#4001 : #4025	#4201 : #4225
B 代码	#4102	#4302
D 代码	#4107	#4307
E 代码	#4108	#4308
F 代码	#4109	#4309
H 代码	#4111	#4311
M 代码	#4113	#4313
顺序号	#4114	#4314
程序号	#4115	#4315
S 代码	#4119	#4319
T 代码	#4120	#4320
附加工件坐标系	#4130	#4330

●位置数据

变 量	意 义	坐标系	刀具补偿
#5001～8	程序段终点位置	工件坐标系	不考虑
#5021～8	当前位置	机械坐标系	考虑
#5041～8		工件坐标系	
#5061～8	跳转信号位置		
#5081～8	刀具长度偏置	—	—
#5101～8	伺服位置偏差	—	—

●工件坐标系原点

基本 6 组		附加 54 组	
#5201～8	外部	#7001～8	G54.1 P1
#5221～8	G54	#7021～8	G54.1 P2
#5241～8	G55	#7041～8	G54.1 P3
#5261～8	G56	:	:
#5281～8	G57	#7901～8	G54.1 P46
#5301～8	G58	#7921～8	G54.1 P47
#5321～8	G59	#7941～8	G54.1 P48

系统变量 （FS15-T ）

●接口信号

变 量	输入信号
#1000 : #1015	UI000 信号 : UI015 信号
#1032	UI000～UI015 （ 16 位一起处理）

变 量	输出信号
#1100 : #1115	UO000 : UO015
#1132	UO000～UO015 （ 16 位一起处理）
#1133	UO100～IO131 （ 32 位一起处理）

●刀具补偿量

补偿号	X轴补偿量		Z轴补偿量		刀尖R补偿量		假想刀尖
	磨损	形状	磨损	形状	磨损	形状	
1 : 64	#2001 : #2064	#2701 : #2764	#2101 : #2164	#2801 : #2864	#2201 : #2264	#2901 : #2964	#2301 : #2364

补偿号	X轴补偿量		Z轴补偿量		刀尖R补偿量		假想刀尖
	磨损	形状	磨损	形状	磨损	形状	
1 : 160	#10001 : #10160	#15001 : #15160	#11001 : #11160	#16001 : #16160	#12001 : #12160	#17001 : #17160	#13001 : #13160

●报警、SETTING、计时器

变 量	功 能	变 量	功 能
#3000	宏程序报警	#3011	时钟（年月日）
#3001	计时器[毫秒]	#3012	时钟（时分秒）
#3002	计时器[小时]	#3901	加工零件数
#3005	SETTING	#3902	所需零件数
#3007	镜像	—	—

●自动运行控制

#3003	单程序段	辅助功能结束
0	有效	等待
1	无效	等待
2	有效	不等待
3	无效	不等待

#3004	进给暂停	倍率	准确停
0	有效	有效	有效
1	无效	有效	有效
2	有效	无效	有效
3	无效	无效	有效
4	有效	有效	无效
5	无效	有效	无效
6	有效	无效	无效
7	无效	无效	无效

●模态数据（G代码体系A）

模态信息	前一个程序段
G 代码（00组） : G 代码（25组）	#4001 : #4025
B 代码	#4102
D 代码	#4107
E 代码	#4108
F 代码	#4109
H 代码	#4111
M 代码	#4113
顺序号	#4114
程序号	#4115
S 代码	#4119
T 代码	#4120

●位置数据

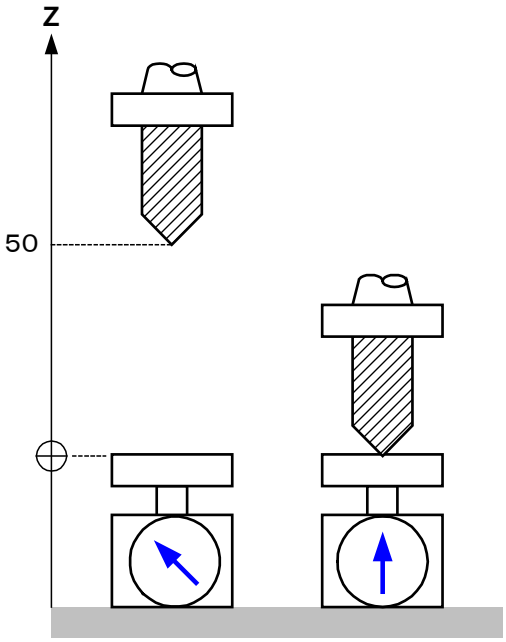
变 量	意 义	坐标系	刀具补偿
#5001～8	程序段终点位置	工件坐标系	不考虑
#5021～8	当前位置	机械坐标系	考虑
#5041～8		工件坐标系	
#5061～8	跳转信号位置		
#5081～8	刀具位置补偿	—	—
#5101～8	伺服位置偏差	—	—

●工件坐标系原点

基本 6 组	
#5201～8	外部
#5221～8	G54
#5241～8	G55
#5261～8	G56
#5281～8	G57
#5301～8	G58
#5321～8	G59

例题 ： 手动测量刀具长度

●操作方法



❶ 选择MDI方式。

❷ 指令Hh；（h是补偿号）然后指定测量刀具的补偿号。

❸ 如果从任意位置起动程序，则将刀具定位在传感器上方50mm的位置。

然后，程序停，用M00停止自动运行。

❹ 切换到手动或手轮进给方式。

❺ 手动使轴移动到基准点。

❻ 返回到自动运行方式。

❼ 若起动自动运行，则补偿量自动设定在被选择的补偿号上，并且Z轴返回到开始测量的高度。

🕮 在此程序中，传感器的基准点是绝对坐标的Z0。

🕮 在次程序中，可以补偿的刀具长度补偿量设为±50mm。

●宏程序

```
O9999 ;
     IF[#4111 EQ 0] GOTO9 ; ..... 不指定补偿号时，轴移到N9
     #1＝#5043 ; ..... 存储绝对坐标的当前位置
     G00 G90 G43 Z50. 0 ; ..... 定位在传感器上方50mm处
     M00 ; ..... 用手动移到基准点
     #[11000＋#4111]＝#5043 ; ..... 把当前位置写入刀具补偿号中
     G00 G49 Z#1 ; ..... 把Z轴返回
     M30 ;
N9   #3000＝9（OFFSET# ERROR）;
```

简单调用（G65）

学习从加工程序中调用用户宏程序的方法。本章中将介绍基本的调用“ 简单调用 ”方法，并且对宏程序中变量的赋值方法进行说明。

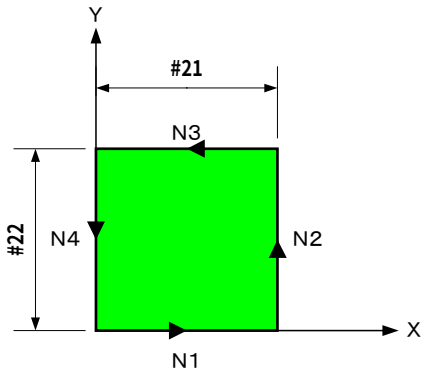
本章介绍的内容如下：

- **宏程序调用指令G65**
- **指定自变量**
- **典型程序**

宏程序调用

● 加工程序调用宏程序

编制一个加工矩形的用户宏程序，由加工程序简单地调用这个程序。




```
O9999 ;
#21=90. 0 ;
#22=120. 0 ;
N1 G01 G91 X#21 F300 ;
N2 Y#22 ;
N3 X-#21 ;
N4 Y-#22 ;
M30 ;
```

🕮 加工程序给宏程序赋值，称为给自变量赋值。

1. 用M98调用用户宏程序。设加工程序的程序号为00001，用户宏程序的程序号为09999。

O0001 ；	O9999 ；
	#21＝90. 0 ；
	#22＝120. 0 ；
M98 P9999 ；	N1 G01 G91 X#21 F300 ；
M30 ；	N2 Y#22 ；
	N3 X－#21 ；
	N4 Y－#22 ；
	~~M30~~ M99 ；

2. 每当加工边长不同的矩形时，09999宏程序中的#21，#22都需要编辑。

为简便起见，把定义#21和#22的程序段移到加工程序中去。

O0001 ；	O9999 ；
#21＝90. 0 ；	~~#21＝90. 0 ；~~
#22＝120. 0 ；	~~#22＝120. 0 ；~~
M98 P9999 ；	N1 G01 G91 X#21 F300 ；
M30 ；	N2 Y#22 ；
	N3 X－#21 ；
	N4 Y－#22 ；
	M99 ；

3. 但是，若数据种类很多时，程序会变得较长，很麻烦。为此改为后面书写变量的定义，这样只用1个语句行就完成了，很方便。

```
#21＝90. 0 ；
#22＝120. 0 ；
M98 P9999 ；
```

↓

M98 P9999 #21＝90. 0 #22＝120. 0 ；

4. 但是该程序中，M98的NC语句和#21=的宏语句混在一起。而且，#21、#22意义也不太清楚。所以改用局部变量的地址A～Z进行指令。

（#21为U，#22为V。详细情况后述。）

```
M98 P9999 #21＝90.0 #22＝120.0 ;
          ↓
M98 P9999 U90.0 V120.0 ;
```

5. 现在，如果把X,Y等轴的名称写在M98的后面，这样轴的移动指令与自变量的赋值就不便于区分，所以为了给宏程序赋值，把M98改为宏程序调用指令G65。

```
M98 P9999 U90.0 V120.0 ;
  ↓
G65 P9999 U90.0 V120.0 ;
```

于是变成宏程序形式的程序如下：

O0001 ;	O9999 ;
G65 P9999 U90.0 V120.0 ;	N1 G01 G91 X#21 F300 ;
M30 ;	N2 Y#22 ;
	N3 X－#21 ;
	N4 Y－#22 ;
	M99 ;

●G65的书写格式

●调用用户宏程序。

简单调用	G65 Pp Ll < 自变量 > ;
	P　　：用户宏程序的程序号 L　　：循环次数 自变量：给用户宏程序赋值的数据

📖 G65在程序段的开头指令。

📖 循环次数为1时，可以省略L。

自变量的指定

●调用宏程序时，从调用的源程序中，给宏程序赋值的参量称为自变量。

●自变量指定方法有两种。自变量指定Ⅰ是使用地址A～Z。自变量指定Ⅱ是使用地址A,B,C 1次和10组I,J,K。

🕮 自变量指定方法，根据使用地址字种类的不同，自动判别。

●把自变量的值代入与地址相对应的局部变量(#1～#33)中。

🕮 调用宏程序时，未指定自变量的地址所对应的局部变量的值为空。

● 自变量指定Ⅰ

● 使用地址A～Z。

● 地址与局部变量的对应关系如下所示:

地址	变量号	地址	变量号	地址	变量号
A	#1	J	#5	S	#19
B	#2	K	#6	T	#20
C	#3	(L)	(#12)	U	#21
D	#7	M	#13	V	#22
E	#8	(N)	(#14)	W	#23
F	#9	(O)	—	X	#24
(G)	(#10)	(P)	(#16)	Y	#25
H	#11	Q	#17	Z	#26
I	#4	R	#18	—	—

🕮 地址G(#10),L(#12),N(#14),P(#16)在通常的宏程序调用中，不可以作为自变量使用。
但是，用参数设定的G代码调用和程序段调用（G66.1）时，可以把它们作为自变量使用。

● 自变量指定 II

●地址A,B,C使用1次，地址I,J,K重复使用到10次。

用于把三维作标值作为自变量使用。

地址	变量号	地址	变量号	地址	变量号	地址	变量号
A	#1	I_3	#10	I_6	#19	I_9	#28
B	#2	J_3	#11	J_6	#20	J_9	#29
C	#3	K_3	#12	K_6	#21	K_9	#30
I_1	#4	I_4	#13	I_7	#22	I_{10}	#31
J_1	#5	J_4	#14	J_7	#23	J_{10}	#32
K_1	#6	K_4	#15	K_7	#24	K_{10}	#33
I_2	#7	I_5	#16	I_8	#25	—	—
J_2	#8	J_5	#17	J_8	#26	—	—
K_2	#9	K_5	#18	K_8	#27	—	—

🕮 地址下标的数字表示地址的顺序，并不存储在程序中。

(例) G65 P9999 A1.0 B1.3 C4.1 I100.0 J50.0 K30.0
I50.0 J30.0 K10.0 I40.0 J60.0 K30.0 ;

🕮 I,J,K的顺序不能改变。

(例)如果指令了 G65 P9999 J1.0 I2.0 K3.0 J4.0 I5.0;
则代入局部变量的数值如下:

#4(I1) : —
#5(J1) : 1.0
#6(K1) : —
#7(I2) : 2.0
#8(J2) : —
#9(K2) : 3.0
#10(I3) : —
#11(J3) : 4.0
#12(K3) : —
#13(I4) : 5.0

● 自变量的小数点

●调用宏程序变量赋值时，不带小数点所赋的值自动地按各地址的单位换算。

●如下所示，不带小数点的数值，局部变量的值换算如下：

```
G65 Pp A1 B2 D3 T4 X5 ;
```

..... A : #1 ：0. 001
..... B : #2 ：0. 002
..... D : #7 ：3. 000
..... T : #20 ：4. 000
..... X : #24 ：0. 005

🕮 把地址B作为第2辅助功能使用时，#2的值为2.0。

🕮 公制车床机床轴构成不同，其地址单位也不同。

带小数点的自变量指定

为了保持加工程序的互换性，自变量指定时必须带小数点。

● 宏程序调用的嵌套

●宏程序可以调用其他的宏程序。

●宏程序调用的嵌套层数、简单调用(G65)和模态调用(66)都是4重。

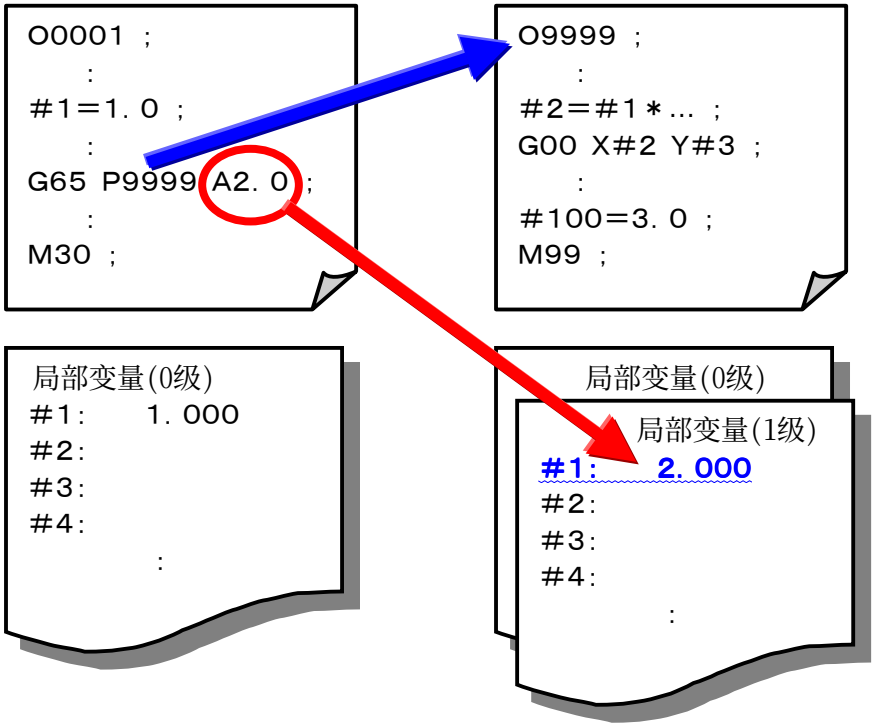
🕮 M98子程序调用的嵌套数是4重。

● 局部变量的级别

●对应于调用宏程序的嵌套，有0级～4级的局部变量。

🕮 在主程序中，也可使用宏变量。
主程序局部变量是0级。

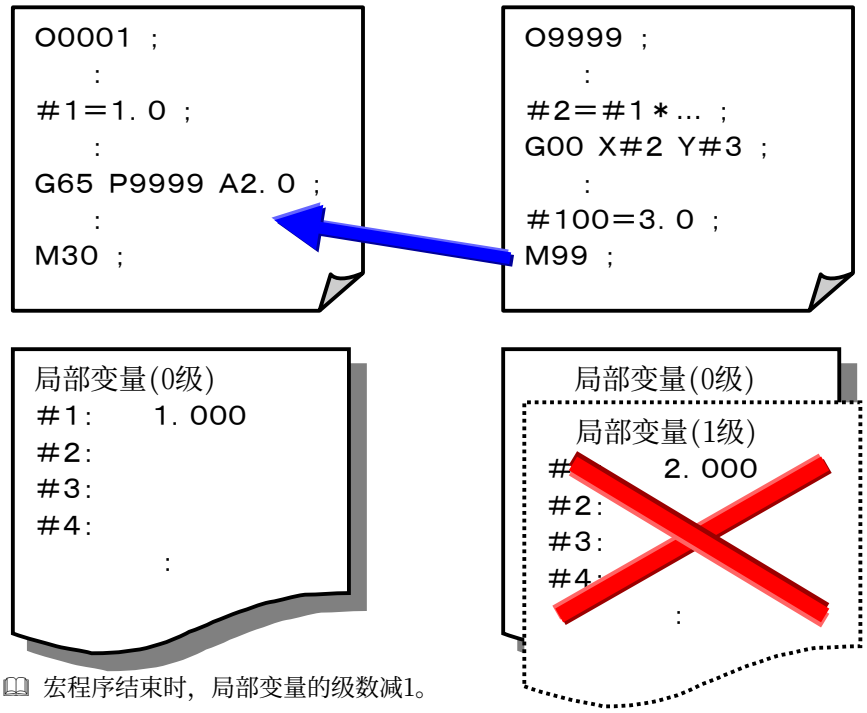
●执行调用宏程序嵌套(G65,G66)时，已使用过的局部变量保存在CNC内部，新调用的宏程序成为新的局部变量。
即调用时，如果指定了自变量，则把值代入与地址相对应的新的局部变量中。



🕮 调用宏程序时，局部变量的级数加1。

●在宏程序中执行M99时，返回到调用的程序中。

已使用的局部变量被删除，把调用宏程序时已保存的局部变量复原。



🕮 宏程序结束时，局部变量的级数减1。

🕮 执行M30时，局部变量是否被删除，由参数设定来决定。

●公共变量与调用的嵌套级无关，各级别的程序使用相同的公共变量。

给宏程序赋值的是自变量
从宏程序传送出数据公共变量...

在加工程序和宏程序之间，相互读取数据时，使用下面的变量。

●从加工程序给宏程序赋值时使用自变量，把值赋给局部变量。

●从宏程序中把数据传送到加工程序(或其他程序)时，使用公共变量。

● 使用模态数据的注意事项！

●首先执行下面的程序。

●加工程序

```
O0001 ;
N10 G00 G90 X100. 0 Y50. 0; ..... 以绝对值移动
N11 G65 P9999 U90. 0 V120. 0 ; ..... 调宏程序
N12 X200. 0 Y100. 0 ; ..... 按增量值移动
M30 ;
```

●宏程序

```
O9999 ;
N1 G01 G91 X#21 F300 ; ..... 宏程序按增量值移动
N2 Y#22 ;
N3 X－#21 ;
N4 Y－#22 ;
M99 ;
```

❶ 在N10中，因指定了快速进给方式(G00)和绝对坐标方式(G90)，所以定位在X100.0,Y100.0的位置上。

❷ 在N11中，调用用户宏程序，加工矩形。
在宏程序N1中，要切换到切削进给方式(G01)和增量方式(G91)。

❸ G00和G90是模态G代码，所以如果不调宏程序，仍用G00,G90的话，N12的程序段是可以省略指令的。但由于执行了宏程序O9999，实际上是增量指令。

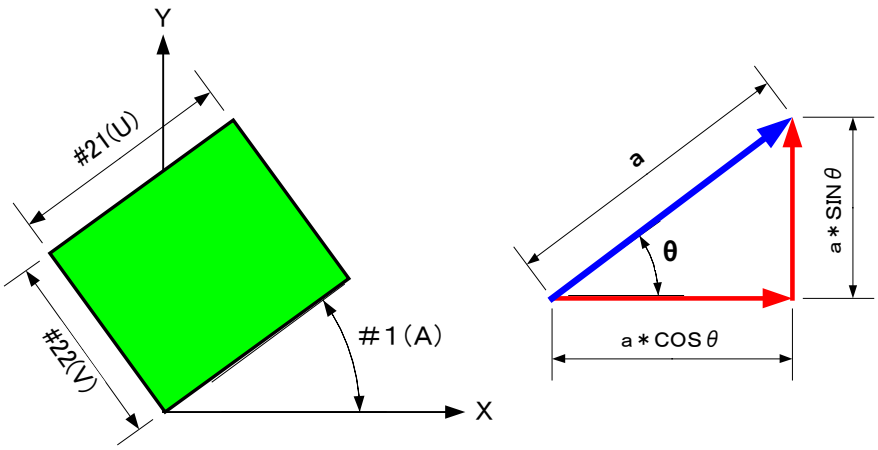
模态信息的复原

为了减轻编程负担，在宏程序的开头，读取模态信息，
到宏程序结束时再复原。

```
O9999 ;
#30＝#4001 ; ..... 存储01组的G代码
#31＝#4003 ; ..... 存储03组的G代码
#32＝#4109 ; ..... 存储 F代码
N1 G01 G91 X#21 F300 ;
N2 Y#22 ;
N3 X－#21 ;
N4 Y－#22 ;
G#30 G#31 F#32 ; ..... 恢复模态信息
M99 ;
```

● 习题 1

加工矩形，如图所示，指定倾角A，加工出倾斜的矩形。



●调用形式

```
G65 P9999 Uu Vv Aa ;
```

●程序

```
O9999 ;
#30=#4001 ;
#31=#4003 ;
#32=#4109 ;
#10=#21*COS[#1] ; ..... 自变量U的X轴方向分量
#11=#21*SIN[#1] ; ..... 自变量U的Y轴方向分量
#12=#22*COS[90+#1] ; ..... 自变量V的X轴方向分量
#13=#22*SIN[90+#1] ; ..... 自变量V的Y轴方向分量
N1 G01 G91 X#10 Y#11 F300 ;
N2 X#12 Y#13 ;
N3 X-#10 Y-#11 ;
N4 X-#12 Y-#13 ;
G#30 G#31 F#32 ;
M99 ;
```

●确认动作

执行下列程序，确认动作。

```
O0001 ;
G92 X0 Y0 ;
G00 G90 X100.0 Y50.0 ;
G65 P9999 U100.0 V50.0 A30.0 ;
M30 ;
```

● 习题 2

●加工程序调用宏程序，变更加工孔深的循环。

调用形式

```
G65 P9999 Zz Rr Qq Ff ;
```

R：基准点坐标（绝对） .....（#18）
Z：孔底的坐标（绝对） .....（#26）
Q：切入量 〈 0 .....（#17）
F：切削速度 .....（#9）

●程序

```
O9999 ;
     #1＝#18 ;  ..... 本次孔深
     #2＝#18 ;  ..... 上次孔深
     #3＝0 ;  ..... 实际的回退量
     WHILE[#1 GT #26] DO1 ;  ..... 重复加工到孔深
     #1＝#1＋#17 ;  ..... 计算本次孔深
     IF[#1 GE #26] GOTO1 ;  ..... 过切检查
     #1＝#26 ;  ..... 如果过切退回
N1   G00 G90 Z[#2＋#3] ;  ..... 快速逼近
     G01 Z#1 F#9 ;  ..... 钻孔
     G00 Z#18 ;  ..... 快速退回
     #2＝#1 ;  ..... 孔深存储
     #3＝1.0 ;  ..... 第2次以后的回退量
     END1 ;
     M99 ;
```

●确认动作

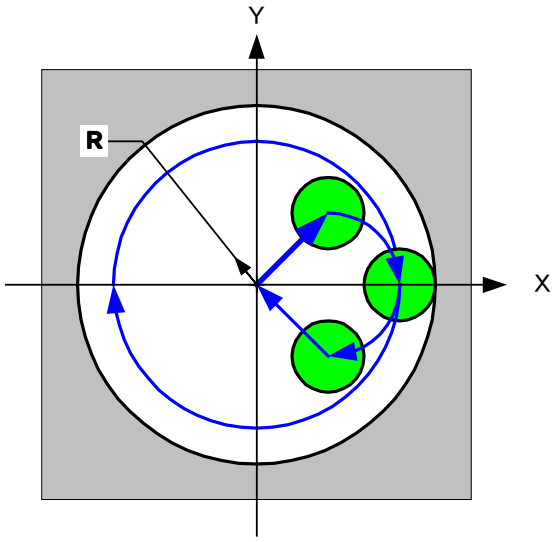
执行下面程序，确认动作。

```
O0001 ;
G00 G90 Z50.0 ;
G65 P9999 Z－25.0 R3.0 Q－10.0 F300 ;
M30 ;
```

例题 ： 切削圆 （M）

●说明

●这是切削内圆的程序。如右图，切削圆时，如果用圆弧进刀，无法找到加工开始/结束点，为此用以下方法。



●用自变量指定刀具补偿号。使用系统变量读取刀具半径值。用已给出的圆的半径与刀具半径的差值的1/2为半径作圆，刀具趋近此圆，来实现进刀。

●刀具半径大于圆半径时，出现宏程序报警。

●调用形式

```
G65 P9999 Rr Dd Ff ;
```

R：圆的半径 .....（#18）
D：刀具半径补偿号 .....（#7）
F：切削进给速度 .....（#9）

●宏程序

●作业用变量

#1 ：刀具半径补偿量
#2 ：减去刀具半径后的圆的半径
#3 ：圆半径的1/2
#30：1组的G代码
#31：3组的G代码
#32：F 代码

●程序

```
O9999 ;
      #30＝#4001 ; ..... 存储 01组 G代码
      #31＝#4003 ; ..... 存储 03组 G代码
      #32＝#4109 ; ..... 存储切削进给速度
      #1＝#[13000＋#7] ; ..... 读取刀具补偿量(注)
      #2＝#18－#1 ; ..... 圆半径减去刀具半径
      IF[#2 LE 0] GOTO9 ; ..... 刀具半径过大？
      #3＝#2/2 ; ..... 加工半径的1/2
N1    G01 G91 X#3 Y#3 F#9 ; ..... 直线移动
N2    G02 G17 X#3 Y－#3 R#3 ; ..... 圆弧趋近
N3    I－#2 ; ..... 加工圆
N4    X－#3 Y－#3 R#3 ; ..... 用圆弧退刀
N5    G01 X－#3 Y#3 ; ..... 返回到中心
      G#30 G#31 F#32 ; ..... 恢复模态代码
      M99 ;
N9    #3000＝1（OFFSET ERROR）; ..... 显示报警并停止
```

🕮 刀具补偿号的系统变量应符合使用的装置。

●确认动作

1. 把下面程序登录在程序的存储器中。

```
O0001 ;
G92 X0 Y0 Z0 ;
G65 P9999 R50.0 D1 F300 ;
M30 ;
```

2. 把刀具半径补偿量(10.0)输入到1号刀具半径补偿量(形状)上。

3. 起动自动运行。

●习题

●编制外径加工的程序。

●上述程序用逆铣加工。精加工时要用顺铣的程序。

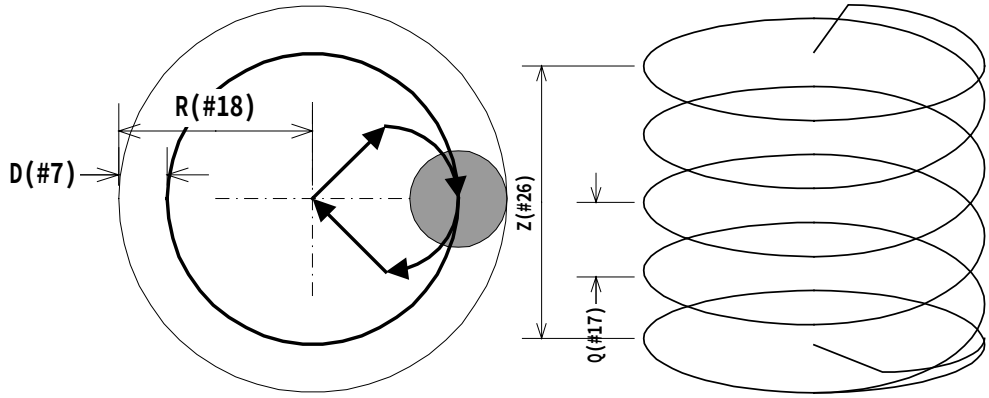
例题 ： 螺旋镗 （M）

●调用形式

```
G65 P9999 Rr Zz Qq Dd Ff ;
```

R：圆的半径 .....（#18）
Z：孔的深度(增量) 〈 0 .....（#26）
Q：Z轴的一次切入量(增量) 〈 0 .....（#17）
D：刀具补偿量 .....（#7）
F ：切削速度 .....（#9）

🕮 孔深Z是每次切入量Q的整数倍。



●宏程序

●作业变量

#1 ：刀具半径补偿量
#2 ：减去刀具补偿量后的圆半径
#3 ：圆半径的1/2
#4 ：切入深度
#30：1组的G代码
#31：3组的G代码
#32：F 代码

●程序

```
O9999 ;
      #30＝#4001 ;
      #31＝#4003 ;
      #32＝#4109 ;
      #1＝#[13000＋#7] ; ..... 读取刀具补偿量(注)
      #2＝#18－#1 ; ..... 圆半径-刀具补偿量
      #3＝#2/2 ; ..... 加工半径的1/2
      IF[#2 LE 0] GOTO 9 ; ..... 补偿量过大
      G01 G91 X#3 Y#3 F#9 ; ..... 移动到趋近圆中心
      G02 G17 X#3 Y－#3 R#3 ; ..... 趋近
      #4＝0 ; ..... 切深初始化
      WHILE[#4 GT #26] DO1 ; ..... 循环到指定深度
      G02 I－#2 Z#17 ; ..... 螺旋补偿
      #4＝#4＋#17 ; ..... 计算切入量
      END1 ;
      G02 I－#2 ; ..... 孔底的圆
      X－#3 Y－#3 R#3 ; ..... 退刀
      G01 X－#3 Y#3 ; ..... 返回到中心
      G#30 G#31 F#32 ;
      M99 ;
N9    #3000＝1(OFFSET ERROR) ; ..... 报警信息
```

📖 刀具半径补偿的系统变量，根据使用装置相应地变更。

●确认动作

1. 1号刀具补偿存储器中，设定刀具半径10.0。

2. 输入下面程序。

```
O0001 ;
G00 G90 X0 Y0 Z10. 0 ;
G01 Z0 F100 ;
G65 P9999 R50. 0 Q－2. 5 Z－25. 0 D1 F300 ;
G00 Z10. 0 ;
M30 ;
```

3. 进行自动运行，确认动作。

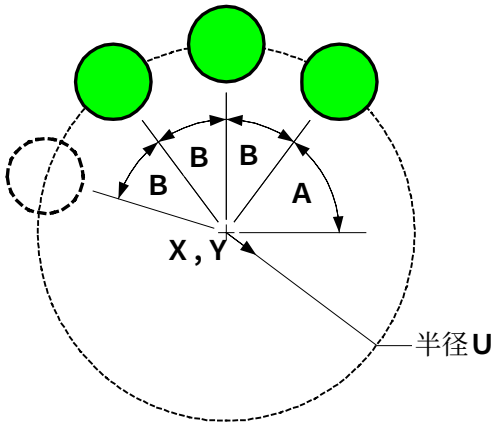
●习题

上述程序是用逆铣的。请把它变更为顺铣的精加工程序。

例题 ： 钻孔循环 （M）

● 说明

在半径为U的圆周上，从A度开始钻H个孔，孔间隔为B度。编制其加工程序。



🕮 圆的中心X,Y和孔的深度Z,R可以用绝对坐标或相对坐标指定。

🕮 如果孔间隔角度B为负值，则按顺时针方向钻孔。

● 调用形式

```
G65 P9999 Xx Yy Zz Rr Uu Aa Bb Hh Ff ;
```

X：圆中心X坐标（绝对指令/增量指令均可用） .....（#24）
Y：圆中心Y坐标（绝对指令/增量指令均可用） .....（#25）
Z：孔的深度（绝对指令/增量指令均可用） .....（#26）
R：趋近点的坐标（绝对指令/增量指令均可用） .....（#18）
U：圆的半径 .....（#21）
A：开始钻孔的角度 .....（#1）
B：孔间隔角度值 .....（#2）
H：孔的个数 .....（#11）
F：切削进给速度 .....（#9）

● 宏程序

●用于作业的变量

#5 ：孔位置的X坐标
#6 ：孔位置的Y坐标
#30：1 组 G代码
#31：3 组 G代码
#32：F 代码

●程序

```
O9999 ;
      #30＝#4001 ;
      #31＝#4003 ;
      #32＝#4109 ;
      IF[#31 EQ 90] GOTO1 ; ..... 转移到绝对方式N1
      #24＝#24＋#5001 ; ..... 把X自变量变成绝对方式
      #25＝#25＋#5002 ; ..... 把Y自变量变成绝对方式
      #26＝#26＋#5003 ; ..... 把Z自变量变成绝对方式
      #18＝#18＋#5003 ; ..... 把R自变量变成绝对方式
N1    WHILE[#11 GT 0] DO1 ; ..... 孔数大于0时循环
      #5＝#24＋#21＊COS[#1] ; ..... 求孔的X坐标
      #6＝#25＋#21＊SIN[#1] ; ..... 求孔的Y坐标
      G81 G90 X#5 Y#6 Z#26 R#18 F#9 ;
                                ..... 钻孔
      #1＝#1＋#2 ; ..... 更新孔的角度
      #11＝#11－1 ; ..... 孔数相减
      END1 ;
      G80 G#30 G#31 F#32 ;
      M99 ;
```

●确认动作

●执行下面程序，确认动作。

```
O0001 ;
G90 G92 X0 Y0 Z100. 0 ;
G65 P9999 X100. 0 Y50. 0 R3. 0 Z－50. 0 F300
                              U100. 0 A0 B45. 0 H5. 0 ;
M30 ;
```

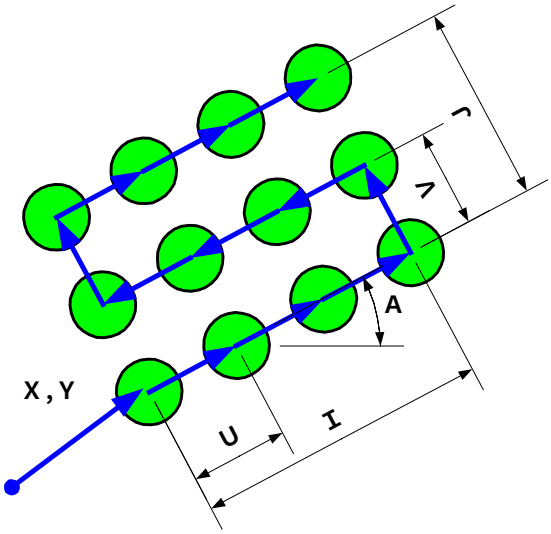
●习题

不指定角度，请把程序改为钻n个间距相等的孔。

例题 ： 加工栅格状分布的孔（M）

编制右图所示的孔加工程序。从基准位置X,Y开始，在X方向钻间距为U的I个孔，在Y轴方向钻间距为V的J个孔。

此例子中指令了角度A，当栅格与X轴平行时，A可以省略。



● 调用形式

G65 P9999 Xx Yy Zz Rr Ff Uu Vv Ii Jj Aa ;

X：基准位置的X坐标（绝对指令/增量指令均可使用）.....(#24)
Y：基准位置的Y坐标（绝对指令/增量指令均可使用）.....(#25)
Z：孔的深度（绝对指令/增量指令均可使用）.....(#26)
R：趋近点的高度（绝对指令/增量指令均可使用）.....(#18)
F：切削进给速度.....(#9)
U：X轴方向孔的间隔.....(#21)
V：Y轴方向孔的间隔.....(#22)
I：X轴方向孔数间隔.....(#4)
J：Y轴方向孔数间隔.....(#5)
A：回转角度.....(#1)

● 宏程序

●用于作业的变量

#6 ：孔数的计数器
#7 ：基准点为0的X方向的孔号
#8 ：X轴的移动方向（正向:1）
#10：U 的 X轴方向分量
#11：U 的 Y轴方向分量
#12：V 的 X轴方向分量
#13：V 的 Y轴方向分量
#14：基准点为0的Y方向的孔号
#30：1 组的 G代码
#31：3 组的 G代码
#32：F 代码

●程序

```
O9999 ;
      #30＝#4001 ; ..... 存 01组的 G代码
      #31＝#4003 ; ..... 存 03组的 G代码
      #32＝#4109 ; ..... 存储 F代码
      IF[#31 EQ 90] GOTO1 ; ..... 若绝对方式转到N1
      #24＝#24＋#5001 ; ..... 把X自变量变成绝对方式
      #25＝#25＋#5002 ; ..... 把Y自变量变成绝对方式
      #26＝#26＋#5003 ; ..... 把Z自变量变成绝对方式
      #18＝#18＋#5003 ; ..... 把R自变量变成绝对方式
N1    #10＝#21＊COS[#1] ; ..... U 的 X轴方向分量
      #11＝#21＊SIN[#1] ; ..... U 的 Y轴方向分量
      #12＝#22＊COS[90＋#1] ; ..... V 的 X轴方向分量
      #13＝#22＊SIN[90＋#1] ; ..... V 的 Y轴方向分量
      G81 G90 Z#26 R#18 F#9 K0 ; ... 设定钻孔条件
      #7＝0 ; ..... X方向孔的号数
      #8＝1 ; ..... X轴正向=1, 负向=-1
      #14＝0 ; ..... Y轴方向孔号-1
      WHILE[#14 LT #5] DO1 ; ..... 循环开始
      #6＝1 ; ..... 一列内的孔的计数器初始化
      WHILE[#6 LE #4] DO2 ; ..... 在X方向的I个循环
      X[#24＋#10＊#7＋#12＊#14]
       Y[#25＋#11＊#7＋#13＊#14] ; 钻孔(G81是模态)
      #6＝#6＋1 ; ..... 一列内的孔数更新
      #7＝#7＋#8 ; ..... X轴方向孔号
      END2 ;
      #14＝#14＋1 ; ..... Y方向的孔号更新
      #8＝－#8 ; ..... X轴移动方向变号
      #7＝#7＋#8 ; ..... 修正X轴的孔号
      END1 ;
      G80 G#30 G#31 F#32 ; ..... 恢复模态信号
      M99 ;
```

●动作确认

执行下列程序，确认动作。

```
O0001 ;
G90 G92 X0 Y0 Z100.0 ;
G65 P9999 X100.0 Y50.0 R3.0 Z－50.0 F300
                          U50.0 V40.0 I4.0 J3.0 A30.0 ;
M30 ;
```

●习题

为了提高定位精度，请用单向定位修改程序。

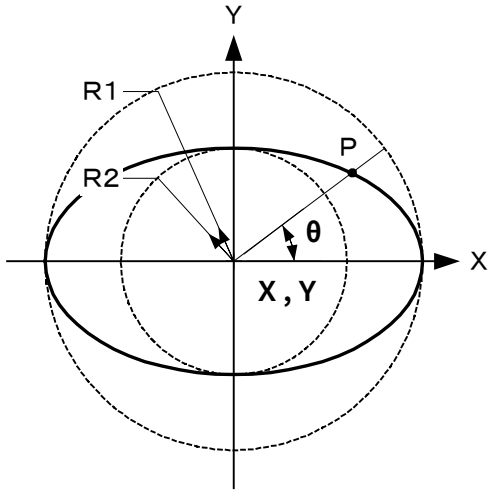
例题 ： 椭圆插补
～ 用直线逼近 ～

●CNC只有圆弧插补，螺旋插补等多种插补功能。对没有插补功能的椭圆等轮廓轨迹，可以用微小距离的直线来近似。

●指定椭圆长短轴半径，分割角度，把各角度的椭圆上的点用直线连接起来。

●椭圆长轴半径设为R_1
短轴半径为R_2，分割角度为θ，则椭圆上的点P的的坐标P_X,P_Y可以用下式求出

$$P_X = R_1 * COS\ \theta$$
$$P_Y = R_2 * SIN\ \theta$$



● 调用格式

```
G65 P9999 Xx Yy Rr Uu Aa Ff ;
```

X：椭圆中心X坐标　（绝对坐标）　.....(#24)
Y：椭圆中心Y坐标　（绝对坐标）　.....(#25)
R：椭圆短轴半径　.....(#18)
U：椭圆长轴半径　.....(#21)
A：分割角度的增量　.....(#1)
F：切削进给速度 .....(#9)

● 宏程序

●作业变量

#2　：椭圆上 P点X坐标
#3　：椭圆上 P点Y坐标
#5　：分割角度
#30：1 组的 G代码
#31：3 组的 G代码
#32：F 代码

●程序

```
O9999 ;
    #30＝#4001 ; ..... 存 01组的 G代码
    #31＝#4003 ; ..... 存 03组的 G代码
    #32＝#4109 ; ..... 存储 F代码
    #5＝#1 ; ..... 椭圆分割角度初始化
    WHILE[#5 LE 360] DO1 ; ..... 循环到360度
    #2＝#24＋#21＊COS[#5] ; ..... 椭圆的 X坐标
    #3＝#25＋#18＊SIN[#5] ; ..... 椭圆的 Y坐标
    G01 G90 X#2 Y#3 F#9 ; ..... 用直线连结椭圆上的点
    #5＝#5＋#1 ; ..... 更新分割角度
    END1 ;
    G#30 G#31 F#32 ;
    M99 ;
```

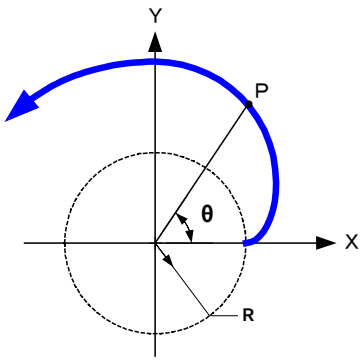
●确认动作

执行程序，确认动作。

```
O0001 ;
G92 X0 Y0 ;
G01 G90 X80. 0 F500 ;
G03 I－80. 0 ;
G01 X100. 0 ;
G03 I－100. 0 ;
G65 P9999 U100. 0 R80. 0 A1. 0 F300 ;
M30 ;
```

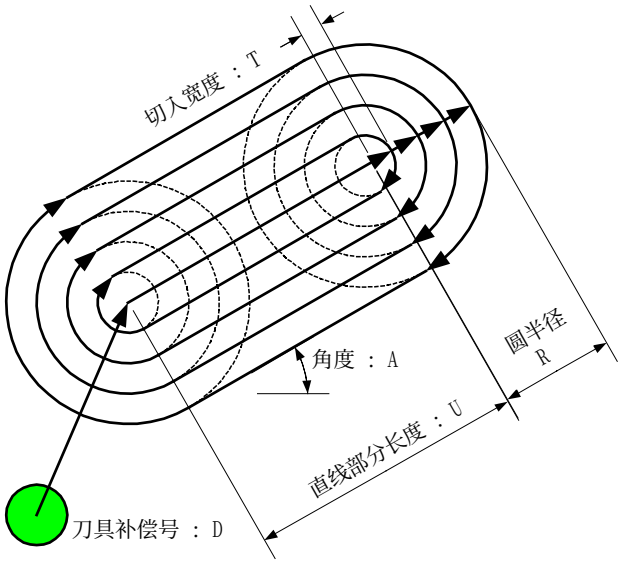
●习题

请用直线近似编制渐开线插补程序。



$$Px=R*[\ \cos\ \theta+\theta*\pi/180*\sin\ \theta\ ]$$
$$Py=R*[\ \sin\ \theta-\theta*\pi/180*\cos\ \theta\ ]$$

例题 ： 环带形凹槽（M）



●调用格式

```
G65 P9999 Xx Yy Rr Uu Tt Dd Ff ;
```

X：基准位置的X坐标（绝对坐标/相对坐标） ..... （#24）
Y：基准位置的Y坐标（绝对坐标/相对坐标） ..... （#25）
R：圆半径 ..... （#18）
U：直线部分的长度..... （#21）
A：倾角 ..... （#1）
T：切入宽度（对于刀具直径的%） ..... （#20）
D：刀具半径的补偿号 ..... （#7）
F：切削进给速度 ..... （#9）

●宏程序

●作业变量

#2 ：由刀具半径和切入率求得的切入量
#3 ：圆的半径
#17：设定的刀具补偿量
#10：直线部分的 X轴分量
#11：直线部分的 Y轴分量
#12：圆弧的终点位置偏移量的 X轴的分量
#13：圆弧的终点位置偏移量的 Y轴的分量
#14：切入位置偏移量的 X轴的分量
#15：切入位置偏移量的 Y轴的分量

●程序

```
O9999 ;
      #30＝#4001 ;
      #31＝#4003 ;
      #32＝#4109 ;
      IF[#31 EQ 90] GOTO1 ; ..... 绝对坐标时转移到N1
      #24＝#24＋#5001 ; ..... 自变量X变成绝对坐标方式
      #25＝#25＋#5002 ; ..... 自变量Y变成绝对坐标方式
N1    #17＝#[13000＋#7] ; ..... 读取刀具半径补偿量
      #18＝#18－#17 ; ..... 从圆半径减去刀具半径
      #2＝#17＊2＊#20/100 ; ..... 计算切入量
      #3＝0 ; ..... 圆弧半径初始化
      #10＝#21＊COS[#1] ; ..... 切入量的 X分量
      #11＝#21＊SIN[#1] ; ..... 切入量的 Y分量
      G00 G90 X#24 Y#25 ; ..... 定位在基准点
      G01 X[#24＋#10] Y[#25＋#11] F#9 ;
      WHILE[#3 LT #18] DO1 ; ..... 半径未达目标时循环
      #3＝#3＋#2 ; ..... 更新圆弧半径
      IF[#3 LT #18] GOTO2 ; ..... 不过切时转到N2
      #3＝#18 ; ..... 过切的目标值
N2    #12＝#3＊COS[#1－90] ; ..... 圆弧终点位置偏移量的 X分量
      #13＝#3＊SIN[#1－90] ; ..... 圆弧终点位置偏移量的 Y分量
      #14＝#3＊COS[#1] ; ..... 切入位置偏移量的 X分量
      #15＝#3＊SIN[#1] ; ..... 切入位置偏移量的 Y分量
      G01 X[#24＋#10＋#14] Y[#25＋#11＋#15] ;
      G02 X[#24＋#10＋#12] Y[#25＋#11＋#13] R#3 ;
      G01 X[#24＋#12] Y[#25＋#13] ;
      G02 X[#24－#12] Y[#25－#13] R#3 ;
      G01 X[#24＋#10－#12] Y[#25＋#11－#13] ;
      G02 X[#24＋#10＋#14] Y[#25＋#11＋#15] R#3 ;
      END1 ;
      G#30 G#31 F#32 ;
      M99 ;
```

🕮 刀具半径补偿的系统变量号，要符合使用装置。

●动作确认

1. 1号刀具半径补偿量(形状)设定10.0。

2. 输入下面程序，确认动作。

```
O0001 ;
G92 G90 X0 Y0 ;
G65 P9999 X50. 0 Y30. 0 D1 R45. 0 U60. 0 A30. 0 T80 F300 ;
M30 ;
```

例题 ： 之字形进给 （M）

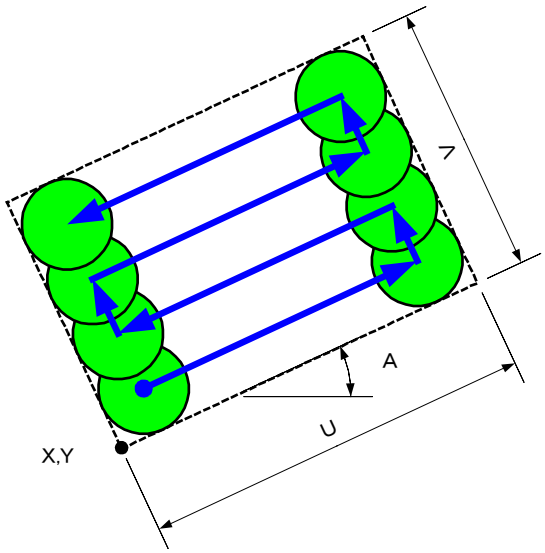
●编制一个以之字形进给进行X-Y平面表面加工的程序。

●从自变量给予的刀具补偿号中读取刀具半径。

刀具直径乘以切入率的积作为切入宽。

📖 使用固定切入量的补偿。

●可以指定角度



● 调用格式

```
G65 P9999 Xx Yy Uu Vv Aa Dd Tt Ff ;
```

X：基准点的 X坐标（绝对） .....（#24）
Y：基准点的 Y坐标（绝对） .....（#25）
U：X轴方向的长度 .....（#21）
V：Y轴方向的长度 .....（#22）
A：回转角度 .....（#1）
D：刀具补偿号 .....（#7）
T：切入率(%) .....（#20）
F：切削进给速度 .....（#9）

● 宏程序

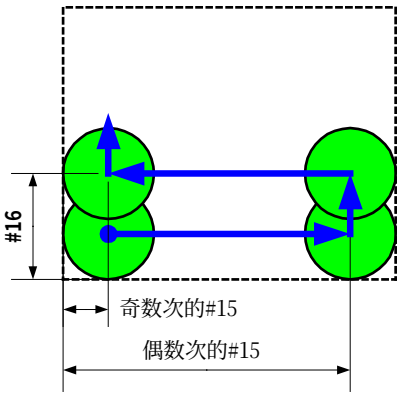
●作业变量

#10：刀具半径
#11：由刀具直径求得的切入宽度
#12：Y方向切入次数
#13：用切入次数修正的切入宽度
#14：循环次数
#15：X 轴方向的基准点到加工点的距离
#16：Y 轴方向的基准点到加工点的距离
#19：坐标回转后的 X 方向距离
#20：坐标回转后的 Y 方向距离
#21：X 方向边长减去刀具半径
#22：Y 方向边长减去刀具半径

●程序

```
O9999 ;
      #30＝#4001 ;
      #31＝#4003 ;
      #32＝#4109 ;
      #10＝#[13000＋#7] ; ..... 读取刀具半径
      #21＝#21－#10 ; ..... 减去刀具半径的X方向距离
      #22＝#22－#10＊2 ; ..... 减去刀具半径的Y方向距离
      #11＝#10＊2＊#20/100 ; ..... 每一次切入宽度
      #12＝FUP[#22/#11] ; ..... 求返回次数
      #13＝#22/#12 ; ..... 平均的切入宽度
      #14＝0 ; ..... 循环次数
      #15＝#10 ; ..... 基准点到加工点的X距离
      WHILE[#14 LE #12] DO1 ;
      #16＝#10＋#13＊#14 ; ..... 更新Y偏移量
      #19＝#15＊COS[#1]＋#16＊COS[90＋#1] ;
                                ..... Y偏移量的X轴方向分量
      #20＝#15＊SIN[#1]＋#16＊SIN[90＋#1] ;
                                ..... Y偏移量的Y轴方向分量
      G01 G90 X[#24＋#19] Y[#25＋#20] F#9 ;
                                ..... 在Y轴方向切入
      #15＝#21 ; ..... 偶数次时 X方向距离
      IF[[#14 AND 1] EQ 0] GOTO1 ; ..... 偶数次时转移到N1
      #15＝#10 ; ..... 奇数次时 X方向距离
N1    #19＝#15＊COS[#1]＋#16＊COS[90＋#1] ;
                                ..... X偏移量的X轴方向分量
      #20＝#15＊SIN[#1]＋#16＊SIN[90＋#1] ;
                                ..... X偏移量的Y轴方向分量
      X[#24＋#19] Y[#25＋#20] ; ..... 在X轴方向切入
      #14＝#14＋1 ; ..... 更新循环次数
      END1
      G#30 G#31 F#32 ;
      M99 ;
```

●程序中的#15和#16表示下面的距离。



🕮 实际上，X,Y轴移动#15和#16的距离。

● 动作确认

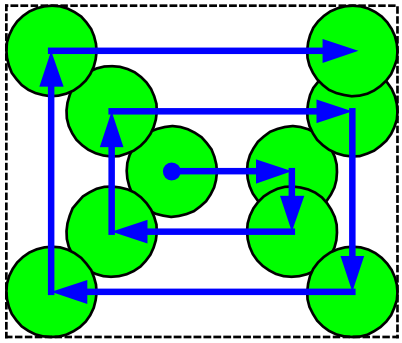
1. 给1号刀具半径补偿(形状)量输入补偿值(10.0)。

2. 执行下列程序，确认动作。

```
O0001 ;
G92 X0 Y0 ;
G65 P9999 X100. 0 Y50. 0 U100. 0 V60. 0 A30. 0 D1 T80 F300;
M30 ;
```

● 习题

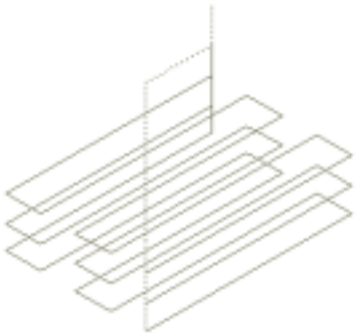
请编制如图所示的从内侧向外侧切削的程序。



例题 ： 加工凹槽 （M）

● 调用方法

●在环带形凹槽和之字形进给的程序上，
组合深孔循环的Z轴动作，进行凹槽加工。



🕮 把之字形进给和深孔循环组合起来。

● 调用格式

```
G65 P9999 Zz Rr Qq Ee Xx Yy Uu Vv Aa Dd Tt Ff ;
```

●用于深孔循环的自变量。

Z：孔深（绝对坐标） .....（#26）
R：趋近点的高度（绝对坐标） .....（#18）
Q：切入量 〈 0 .....（#17）
E：钻孔的切削速度 .....（#8） （注）

🕮 从R点到切入开始点的动作速度是从快速进给变成了切削进给（进给速度是钻孔速度的5倍）。

🕮 因钻孔和之字形进给的切削进给速度不同，把深孔循环用的切削进给速度的自变量由 F 变更为 E 。

●之字形进给用的自变量

X：基准点的 X作标（绝对） .....（#24）
Y：基准点的 Y作标（绝对） .....（#25）
U：X 轴方向长度 .....（#21）
V：Y 轴方向长度 .....（#22）
A：回转角度 .....（#1）
D：刀具补偿号 .....（#7）
T：切入率（%） .....（#20）
F：之字形进给用的切削进给速度.....（#9）

●宏程序

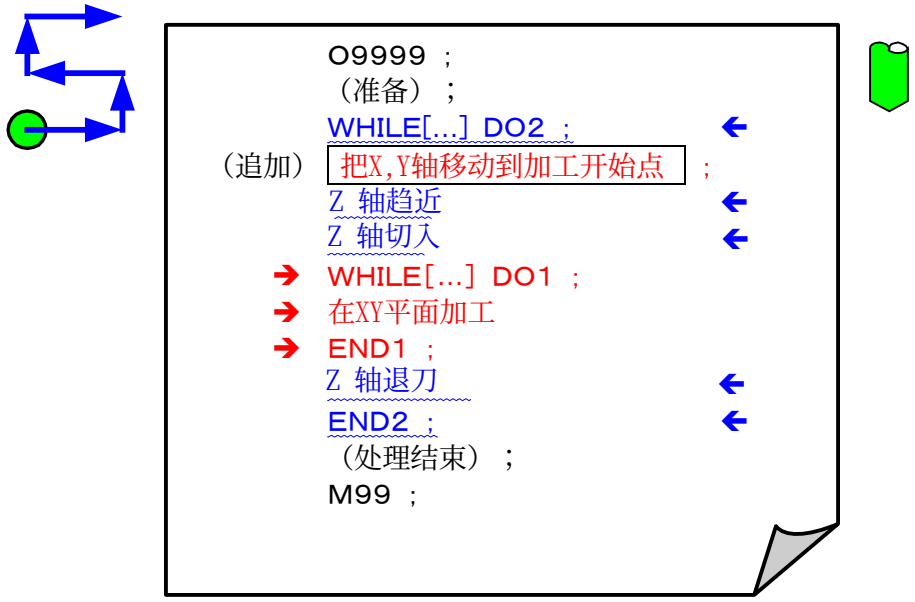
```
O9999 ;
      #30＝#4001 ;
      #31＝#4003 ;
      #32＝#4109 ;
      #10＝#[13000＋#7] ;
      #21＝#21－#10 ; ..... 减去刀具半径的X轴方向距离
      #22＝#22－#10＊2 ; ..... 减去刀具半径的Y轴方向距离
      #11＝#10＊2＊#20/100 ; ..... 每次切入宽度
      #12＝FUP[#22/#11] ; ..... 求返回次数
      #13＝#22/#12 ; ..... 已平均过的切入宽度
      #4＝#18 ; ..... 本次孔深
      #5＝#18 ; ..... 前次孔深
      #3＝0 ; ..... 退刀量
      #27＝#24＋#10＊COS[#1]＋#10＊COS[90＋#1] ;
                                  ..... Z 轴切入位置的 X坐标
      #28＝#25＋#10＊SIN[#1]＋#10＊SIN[90＋#1] ;
                                  ..... Z 轴切入位置的 Y坐标
      WHILE[#4 GT #26] DO2 ; ..... Z 轴方向的循环开始
      #4＝#5＋#17 ; ..... 计算 Z轴孔位置
      IF[#4 GE #26] GOTO2 ; ..... 检查过切
      #4＝#26 ; ..... 过切时的限制
N2    G00 G90 X#27 Y#28 ;
      Z[#5＋#3] F[#8＊5] ; ..... Z 轴趋近
      G01 Z#4 F#8 ; ..... Z 轴切入
      #14＝0 ; ..... 循环次数
      #15＝#10 ; ..... 基准点到加工点的X距离
      WHILE[#14 LE #12] DO1 ;
      #16＝#10＋#13＊#14 ; ..... 更新 Y 偏移量
      #19＝#15＊COS[#1]＋#16＊COS[90＋#1] ;
                                  ..... Y 偏移量的 X轴方向分量
      #20＝#15＊SIN[#1]＋#16＊SIN[90＋#1] ;
                                  ..... Y 偏移量的 Y轴方向分量
      G01 G90 X[#24＋#19] Y[#25＋#20] F#9 ;
                                  ..... 在 Y 轴方向的切入
      #15＝#21 ; ..... 偶数次时 X方向的距离
      IF[[#14 AND 1] EQ 0] GOTO1 ; ..... 偶数次时转移到N1
      #15＝#10 ; ..... 奇数次时 X方向的距离
N1    #19＝#15＊COS[#1]＋#16＊COS[90＋#1] ;
                                  ..... X 偏移量的 X轴方向分量
      #20＝#15＊SIN[#1]＋#16＊SIN[90＋#1] ;
                                  ..... X 偏移量的 Y轴方向分量
      X[#24＋#19] Y[#25＋#20] ; ..... 在 X 轴方向的切入
      #14＝#14＋1 ; ..... 更新循环次数
      END1
      G00 Z#18 ; ..... Z 轴退回
      #5＝#4 ; ..... 存储本次孔深
      #3＝1.0 ; ..... 第2次以后的退刀量
      END2 ;
```

```
G#30 G#31 F#32 ;
M99 ;
```

- 以前，孔循环的深度是用#1和#2控制的，为了防止变量的重复，现在把#1改为#4，#2改为#5。
- 刀具补偿的系统变量号，根据使用的装置，应相应地变更。

● 总结

编制大程序时，首先作出基本的动作（此例中是XY平面的动作）的程序，然后再追加其后附加的动作，这样就简单了。



● 动作确认

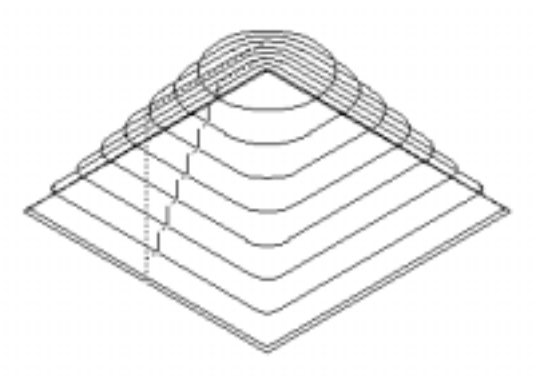
1. 给1号刀具半径补偿量(形状)输入刀具半径补偿值(10.0)。

2. 执行下列程序，确认动作。

```
O0001 ;
G92 X0 Y0 Z100.0 ;
G65 P9999 Z-50.0 R3.0 Q-10.0 E200
      X100.0 Y50.0 U100.0 V60.0 A30.0
D1 T80 F300 ;
M30 ;
```

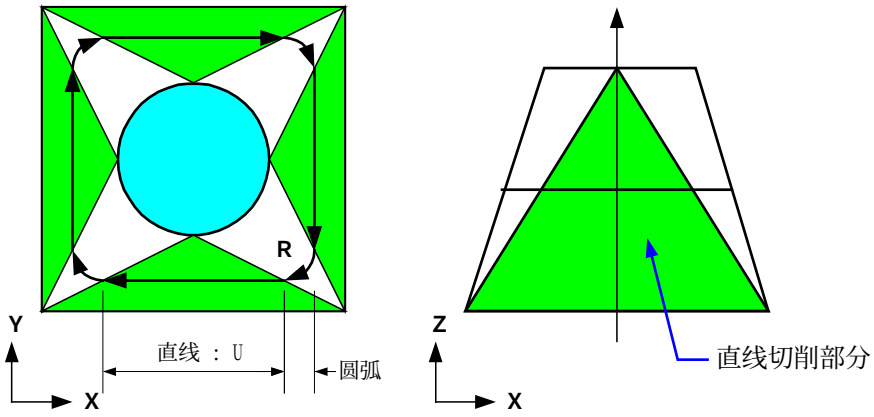
例题 ： 上圆下方体零件的加工（M）

如图所示，用等高线加工的方法，编制上面是圆形，底部为方形的零件加工宏程序。



● 要点

若找到刀具轨迹的规律，则可以知道正方形四个角的拐角R，直线长度和圆半径与Z轴的高度是成比例的。



🕮 最上面直线长度为0，底面圆的半径为0。

● 调用格式

```
G65 P9999 Rr Uu Zz Qq Ff ;
```

R：圆半径 .....(＃18)
U：正方形的边长 .....(＃21)
Q：一次切入量〈 0 .....(＃17)
Z：高度 〉0（底面为0） .....(＃26)
F：切削进给速度 .....(＃9)

🕮 在开始加工之前，事先把X,Y轴定位在圆心上。

●宏程序

●作业变量

#1 ：Z 轴的高度
#2 ：直线的长度
#3 ：拐角 R 半径

●程序

```
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

执行下列程序，确认动作。

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

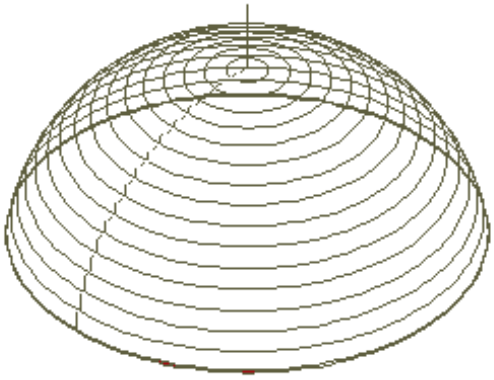
● 习题

●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。

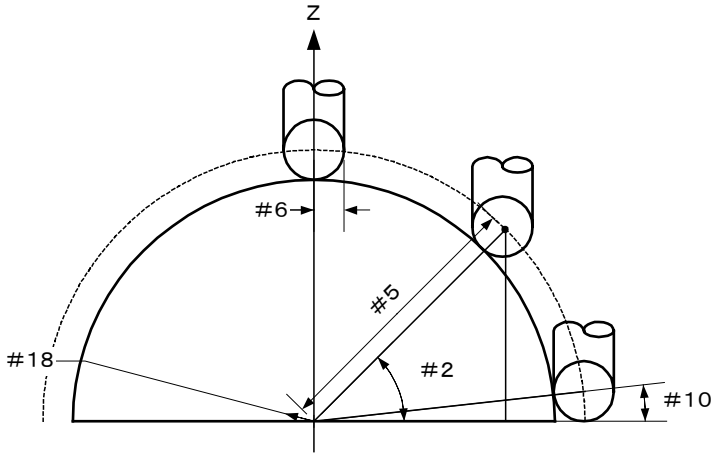
●修改程序，使它还可以加工上面是正方形，底面为圆形的零件。

例题 ： 加工球（M）

●编制使用球铣刀加工球的宏程序。



●从90度到0度方向，按一定角度进行角度分割，求出每个角度的Z轴高度和X-Y平面圆的半径，进行圆弧插补。



🕮 在此程序中，球的底面坐标为Z0。

●如果重复圆弧插补到0度，过切部分只是球刀半径。由球半径和球刀半径求出最终角度(#10)。

●读取使用的刀具半径补偿量，加工放大的球，放大量为刀具半径补偿量。

● 调用格式

```
G65 P9999 Rr Dd Aa Ff ;
```

R：球半径 .....（#18）
D：刀具半径补偿号 .....（#7）
A：分割角度 .....（#1）
F：切削进给速度 .....（#9）

●程序

●作业变量

#2 ：切削位置的角度
#3 ：切削位置的 X坐标
#4 ：切削位置的 Z坐标
#5 ：加上球刀半径后的半径
#6 ：球刀半径
#7 ：最后一次球半径
#10：最终加工的角度
#24：中心的 X 坐标
#25：中心的 Y 坐标

●程序

```
O9999 ;
      #30＝#4001 ;
      #31＝#4003 ;
      #32＝#4109 ;
      #24＝#5001 ; ..... 加工开始点的 X 坐标
      #25＝#5002 ; ..... 加工开始点的 Y 坐标
      #6＝#[13000＋#7] ; ..... 读取刀具半径补偿量(注)
      #5＝#18＋#6 ; ..... 加上球刀半径后的球半径
      #7＝SQRT[#5＊#5－#6＊#6] ; .... 最后一次球半径
      #2＝90＋#1 ; ..... 角度初始化
      #10＝ATAN[#6]/[#7] ; ..... 求最终角度
      G01 G90 Z#5 F#9 ; ..... Z 轴趋近
      WHILE[#2 GT #10] DO1 ; ..... 循环到最终角度
      #2＝#2－#1 ; ..... 角度更新
      IF[#2 GT #10] GOTO1 ; ..... 若过切转移到N1
      #2＝#10 ; ..... 过切时的最终角度
N1    #3＝#24＋#5＊COS[#2] ; ..... 切削点的 X 坐标
      #4＝#25＋#5＊SIN[#2] ; ..... 切削点的 Y 坐标
      G00 X[#24＋#3] ; ..... 移动到 X 轴的切削位置
      G01 Z#4 ; ..... Z 轴切入
      G02 I－[#3] ; ..... 加工圆
      END1 ;
      G#30 G#31 F#32 ;
      M99 ;
```

🕮 刀具补偿的系统变量号，根据使用装置作相应变更。

● 动作确认

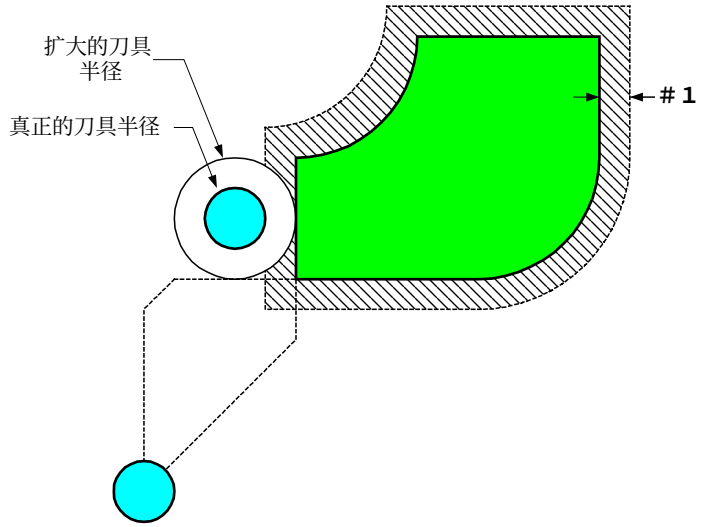
1. 给1号刀具半径补偿量(形状)输入球刀的半径(10.0)。

2. 执行下列程序，确认动作。

```
O0001 ;
G92 X0 Y0 Z100. 0 ;
G65 P9999 R50. 0 D1 A5. 0 F300 ;
M30 ;
```

例题 ： 切入（进刀）宏程序
～ 把操作作业宏程序化 ～

编一个能在加工程序中，用宏程序变更所用的刀具半径补偿量，并自动地进行的切入的程序。



📖 在操作作业中，加大刀具半径补偿量，进行粗加工，然后渐渐地把刀具半径补偿量接近真正的值，循环加工进行切入。此操作就宏程序化了。

●调用格式

```
G65 P9999 Aa Bb Qq Dd Tt ;
```

A：加工余量 ..... （＃1）
B：精加工余量 ..... （＃2）
D：刀具半径补偿号 ..... （＃7）
Q：切入量 ..... （＃17）
M：加工程序 ..... （＃13）

●宏程序

●作业变量

＃4：精加工时的刀具半径补偿量
＃5：实际的刀具半径补偿。

●程序

```
O9999 ;
      #5＝#[13000＋#7] ; ..... 读取实际刀具半径补偿量(注)
      #4＝#5＋#2 ; ..... 精加工时刀具半径补偿量
      #13099＝#5＋#1 ; ..... 刀具半径补偿量的初始值(注)
      WHILE[#13099 GT #4] DO1 ; .....
      #13099＝#13099－#17 ; ..... 刀具半径补偿量相减
      IF[#13099 GE #4] GOTO1 ; ..... 若没过切转移到N1
      #13099＝#4 ; ..... 恢复精加工时刀具补偿量
N1    M98 P#13 ; ..... 调用加工程序
      END1 ;
      M99 ;
```

🕮 在此程序中，使用刀具补偿号99。按照可以使用的号码，变更刀具半径补偿用的系统变量号和加工程序中的刀具半径补偿号D(﹏ 线部分)。

●动作确认

1. 给2号刀具半径补偿(形状)设定实际的刀具半径(10.0)。

   🕮 在宏程序中，改变刀具半径补偿(形状)99号的值，重复加工。

2. 执行下列程序，确认动作。

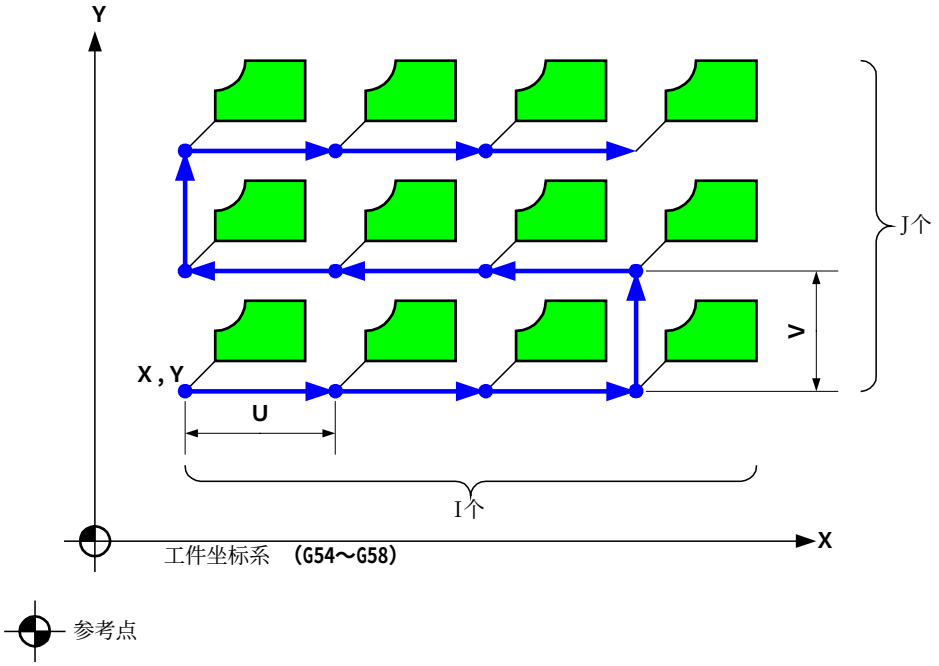
```
O0001 (MAIN) ;
G92 X－50. 0 Y－50. 0 ;
G65 P9999 A15. 0 B0. 2 D2 Q5. 0 M2 ;
M30 ;

O0002 (PARTS) ;
G01 G90 G41 X0 Y－20. 0 F200 D99 ;
G91 Y30. 0 ;
G03 X20. 0 Y20. 0 R20. 0 ;
G01 X30. 0 ;
Y－40. 0 ;
G02 X－10. 0 Y－10. 0 R10. 0 ;
G01 G90 X－20. 0 ;
G40 X－50. 0 Y－50. 0 ;
M99 ;
```

   🕮 加工程序编成子程序。

例题 ： 用一个宏程序加工多个工件
～ 应用CNC功能 ～

编制一个在宏程序中变化工件坐标系，使用交换工作台，加工多个工件的程序。在车床上，可用于沟道加工。



●在宏程序中，一面改变工件坐标系6(G59)的偏移量，一面进行循环加工。

🕮 加工程序中，不能使用G59。

● 调用格式

```
G65 P9999 Xx Yy Uu Vv Ii Jj Dd ;
```

X：基准点的 X 作标（绝对） ..... (#24)
Y：基准点的 Y 作标（绝对） ..... (#25)
U：X 轴方向的间隔 ..... (#21)
V：Y 轴方向的间隔 ..... (#22)
I：X 轴方向的个数 ..... (#4)
J：Y 轴方向的个数 ..... (#5)
M：加工程序号 ..... (#13)

● 宏程序

●作业变量

#3 ： 工件坐标系的编号
#6 ： X方向的循环次数

●程序

```
O9999 ;
      #30＝#4001 ;
      #31＝#4003 ;
      #32＝#4109 ;
      #33＝#4014－53 ; ..... 取使用的工件坐标系号
      #5321＝#24＋#[5201＋20＊#33]－#21 ;
                                ..... 把G59的 X 初始化
      #5322＝#25＋#[5202＋20＊#33] ;
                                ..... 把G59的 Y 初始化
      DO1 ;
      #6＝1 ; ..... X 方向循环次数初始化
      WHILE[#6 LE #4] DO2 ; ..... 开始 X 方向的循环
      #5321＝#5321＋#21 ; ..... 更新 G59 的 X
      G00 G90 G59 X0 Y0 ; ..... 移动到 G59 的参考点
      M98 P#13 ; ..... 调出加工条件
      #6＝#6＋1 ; ..... 更新 X 方向循环次数
      END2 ;
      #5＝#5－1 ; ..... Y 方向循环次数减1
      IF[#5 LE 0] GOTO1 ; ..... 若结束转到N1
      #5321＝#5321＋#21 ; ..... 更新 G59的 X
      #5322＝#5322＋#22 ; ..... 更新 G59的 Y
      #21＝－#21 ; ..... 改变 X 的移动方向
      END1 ;
N1    G[53＋#33] G#30 G#31 F#32 ;
      M99
```

●动作确认

执行下列程序，确认动作。

```
O0001（MAIN）;
G28 G91 X0 Y0 Z0 ;
G00 G54 X0 Y0 Z0 ;
G65 P9999 X150. 0 Y100. 0 U100. 0 V80. 0 I4. 0 J3. 0 M2 ;
M30 ;

O0002（PART）;
G00 G90 X50. 0 Y50. 0 ;
G01 X100. 0 F300 ;
Y90. 0 ;
X70. 0 ;
G02 X50. 0 Y70. 0 R20. 0 ;
G01 Y50. 0 ;
M99 ;
```

📖 将加工程序编为子程序。

例题 ： 自动测量刀具长度

●编制使用接触式传感器自动测量刀具长度的程序。

●动作如下

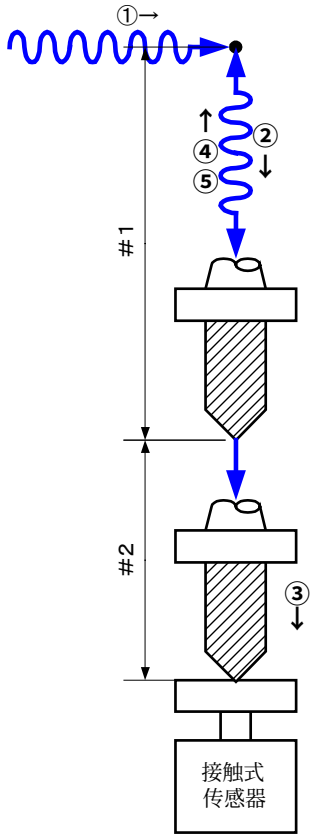
❶ 把X,Y轴移动到测量位置上。

❷ 使用当前设定的刀具长度补偿量，Z 轴下降#1的距离。

❸ 用跳转进给功能，把 Z轴移动2倍#2的距离。
（可以变更的补偿量最大值是#2）。

❹ 测量后，Z轴返回到测量开始点。

❺ 计算刀具长度补偿量，写入相应的刀具补偿存储器中。



●系统变量的跳转信号位置(#5061～)是读取跳转信号时的指令位置。

从读取的跳步位置减去伺服位置偏差量，可以求出正确的跳转信号位置。

🕮 通常跳转进给功能在空运行，进给速度倍率，自动加减速中不用。

●伺服位置偏差量可以用下述方法求得。

— 计算公式如下

$$\text{伺服位置偏差量} = \frac{\text{进给速度〔mm/min〕}}{60} \times \frac{1}{\text{伺服回路增益}}$$

🕮 伺服回路增益由参数1825中得到。

— 进给跳转时，读取诊断画面显示的伺服位置偏差量。

🕮 伺服位置偏差的诊断号分别为300号（FS16*i*），3000号（FS15*i*）。

🕮 改变跳转进给功能的进给速度时，伺服位置偏差量也变化。

● 调用形式

```
G65 P9999 Hh ;
```

H：刀具长度补偿号（#11）

● 宏程序

```
O9999 ;
      #30＝#4001 ;
      #31＝#4003 ;
      #32＝#4109 ;
      #1＝300 ; ..... 原点与趋近点的距离
      #2＝100 ; ..... 传感器与趋近点的距离
      G28 G91 Z0 ;
      #4＝#5003 ; ..... 存原点的绝对坐标值
      G00 G90 G53 X200. 0 Y150. 0 ; ... X,Y轴移到传感器上方
      G91 G43 Z－#1 H#11 ; ..... Z 轴下降到趋近点
      #5＝#5003－#2 ; ..... 计算传感器上面绝对坐标值
      G31 Z－[#2＊2] F300 ; ..... 测量
      G00 G90 G49 Z#4 ; ..... Z 轴退回
      #6＝#5063－#[11000＋#11] ; ..... 接触到传感器时的绝对坐标
      IF[#6 LE [#5－#2]] GOTO9 ; ..... 若没接触到传感器转到N9
      #[11000＋#11]＝#5063－#5 ; ..... 求刀具长度补偿
      G#30 G#31 F#32 ;
      M99 ;
N9    #3000＝1（CAN NOT MEASURE）;
```

📖 刀具长度补偿的系统变量号，根据使用的装置不同而相应的变化。

● 动作确认

1. 在1号刀具长度补偿(形状)上，输入测量前的刀具长度补偿量(100.0)。

2. 执行下列程序，确认动作。

```
O0001 ;
G28 G91 X0 Y0 Z0 ;
G92 X0 Y0 Z500. 0 ;
G65 P9999 H1 ;
M30 ;
```

3. 用G31指令 Z轴下降时，输入跳转信号。

4. 程序结束后，确认是否正确地设定了刀具长度补偿量。

用图形画面确认动作

本章将介绍，使用图形功能，确认程序轨迹的方法。

●FS16 *i*-M

1. 选择描绘图形的程序。

❶ 按操作面板上的 MEMORY 键，选择存储器运行方式。

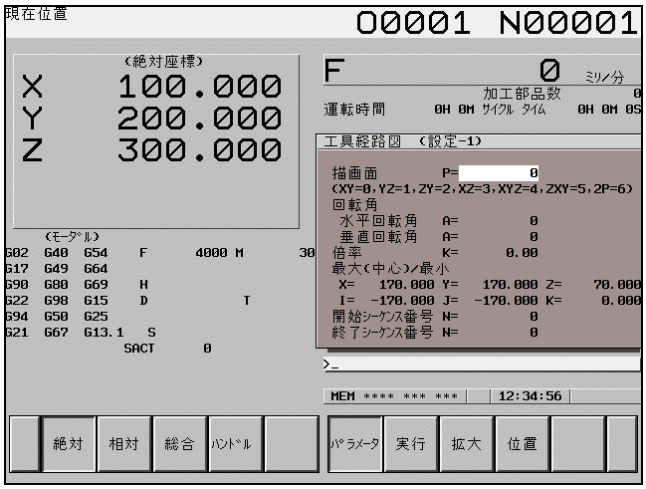
❷ 按 PROG 键，显示程序画面。

❸ 顺次按程序号，软键 0检索 ，选择要描绘的程序。

2. 设定描绘刀具轨迹图的参数。

❶ 按 GRAPH 键。

显示描绘刀具轨迹图的参数设定画面。



❷ 选择描绘画面。

输入 0～6，按 INPUT 键。

❸ 在描绘平面中，选择4(X Y Z)或5(Z X Y)时，须设定回转角。

3. 设定要显示的颜色等。

❶ 按 [页↓] 键。

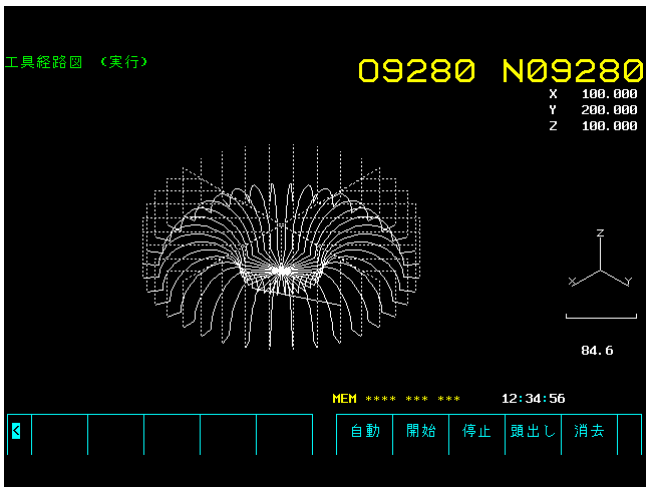
显示设定-2画面。

❷ 关于是否使用刀具半径补偿的设定。
使用时设定 0，不使用时设定 1。

❸ 设定描绘的颜色。用刀具交换命令M06，改变颜色。

❹ 按 [〈] 键，结束描绘参数的设定。

4. 顺次按软键 [(操作)] ， [自动] ，开始绘图。



在描绘中，机床操作面板的循环起动指示灯亮。

5. 如果需要，可将描绘的图形放大。

❶ 按软键 [放大] 。

❷ 按软键 [(操作)] ，显示 ← → ↑ ↓等软键。

❸ 按软键 ← → ↑ ↓，把 ■ 移到放大中心。

❹ 按 [P] (放大)，或 [M] (缩小)键，变更描绘倍率。

❺ 若按软键 [执行] ，则用指定的倍率，描绘图形。

6. 按以下步骤显示加工中的机械位置。

❶ 按软键 [位置] 。

在画面上，显示表示机械位置 ■ 。

❷ 按机床操作面板的 

键，开始自动运行。

● FS15 *i*-M

1. 选择描绘画面。

按软键 [描绘] ，即使是处于描绘画面，也要按一次软键 [描绘] 。

2. 设定描绘参数。

❶ 按软键 [设定] 。
显示描绘参数画面。

❷ 把光标移到描绘平面上，从软键中选择平面。

❸ 刀具轨迹和快速进给描绘有效。

❹ 把光标移到颜色指定，设定每把刀具的颜色。

❺ 按软键 [设定结束] 后，结束设定。

3. 描绘加工轨迹图。

❶ 按软键 [程序号] 。

❷ 输入程序号。

❸ 按软键 [自动开始] 。

模态调用
（G66，G66 1）

学习连续调用宏程序的模态调用方法。
当加工一连串孔时，使用这种方法很方便。

本章介绍下列内容：

- **位置移动后调用（G66）**
- **多重模态调用**
- **每个程序段调用（G66.1）**

模态调用有以下2种。

名 称	代 码	功 能
移动后调用	G66	每执行一个移动指令的程序段时，即调用宏程序。
FS15 *i* 每个程序段调用	G66. 1	每当执行一个CNC语句时，即调用宏程序。

移动后调用（G66）

● 功能

●从指令模态调用指令G66的程序段开始，到用G67取消为止，其间所执行的每个轴移动指令的程序段都调用宏程序。

●无移动指令只有辅助功能和暂停(G04)等的程序段，不调用宏程序。

🕮 即使移动量为零，但只要有轴移动指令，就调用宏程序。

● 书写格式

●在移动的程序段后，调用范围的开头指令G66，在调用范围的最后指令G67。

●在G66之后，指令调用的宏程序程序号(P)和循环次数(L)并给自变量赋值。

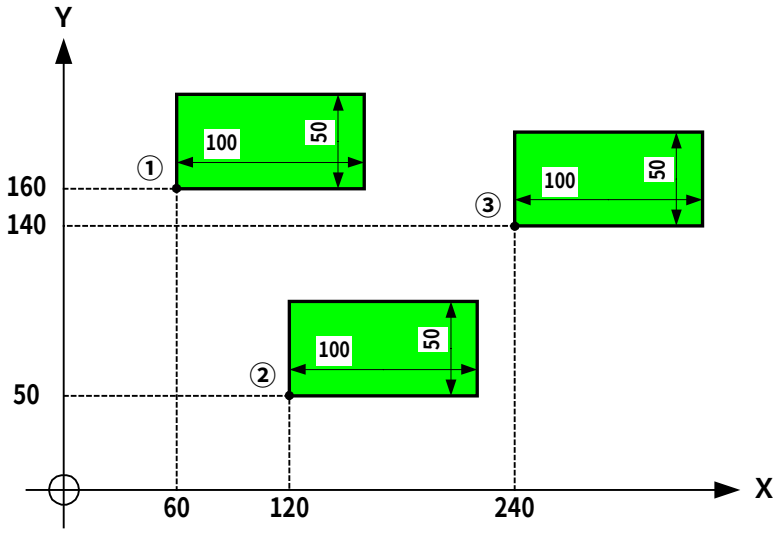
```
O0001 ;
    :
G66 Pp <自变量> ; ..... 移动后开始调用
    :
G00 Xx Yy ; ..... 每执行一次移动指令即调用一次宏程序
    :
G67 ; ..... 取消移动后调用
    :
M30 ;
```

移动后 调用	G66 Pp Ll <自变量>; **P**：调用用户宏程序的程序号 **L**：循环次数 自变量：给用户宏程序内的变量赋值。

- G66的书写格式与简单调用(G65)相同。
- G66写在程序段的开头。
- 在G66的程序段中，不调用宏程序。
- 只在执行G66程序段时，才把自变量的值代入到局部变量中。调用宏程序时并不设定。
  自变量是在宏程序内代入局部变量值时使用。

● 使用例

加工多个相同边长的矩形的程序。



O0001(主程序)；	O9999(宏程序)；
N11 G66 P9999 U100.0 V50.0 ；	#30＝#4001 ；
N12 G00 G90 X60.0 Y160.0 ；	#31＝#4003 ；
N13 X120.0 Y50.0 ；	#32＝#4109 ；
N14 X240.0 Y140.0 ；	N1 G01 G91 X#21 F300 ；
N15 G67 ；	N2 Y#22 ；
M30 ；	N3 X－#21 ；
	N4 Y－#22 ；
	G#30 G#31 F#32 ；
	M99 ；

📖 在N11的G66，N15的G67，M99程序段中，单程序段工作时不停止。

例题 ：
模态调用深孔加工循环

●编制“ 深孔钻削循环 ”的程序，然后用G66指令连续调用。

📖 与深孔加工循环的程序相同。

●执行下列程序。

```
O0001 ;
G92 X0 Y0 Z200.0 ;
G66 P9999 R3.0 Z－25.0 Q－10.0 F300 ;
G00 G90 X100.0 Y50.0 ;
Y150.0 ;
X0 ;
G67 ;
M30 ;
```

例题 ： 切削距离（G66）

●使用移动后调用(G66)，编写一个测量切削距离的程序。

●在此程序中，用直线插补移动的切削距离的累积值写入宏变量#501中。

●程序实例

```
O9999 ;
      IF[#3 EQ #0] GOTO9 ; ..... 若前面无终点位置 即转移到N9
      IF[#4001 NE 1] GOTO9 ; ..... 若不是G01 即转移到N9
      #1＝#5001－#3 ; ..... 求 X 轴移动量
      #2＝#5002－#4 ; ..... 求 Y 轴移动量
      #501＝#501＋SQRT[#1＊#1＋#2＊#2] ;
                                  ..... 累积移动量
N9    #3＝#5001 ; ..... 存 X 轴程序段终点位置
      #4＝#5002 ; ..... 存 Y 轴程序段终点位置
      M99 ;
```

●动作确认

执行下面的程序，确认动作。

```
O0001 ;
/ G28 G91 X0 Y0 ;
G66 P9999 ;
G00 G90 X100. 0 Y50. 0 ;
G01 X150. 0 Y100. 0 F300 ;
G91 Y100. 0 ;
G67 ;
G00 G90 X100. 0 Y50. 0 ;
M30 ;
```

●习题

●修改程序，使它能累积圆弧插补的距离。

●改编程序，当切削距离的累积值超过设定值(#500)时，
用(M00)停止程序。

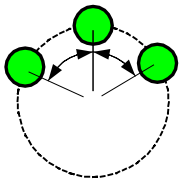
多重模态调用

在G66模态调用中，若再一次指令模态调用时，称为多重模态调用。

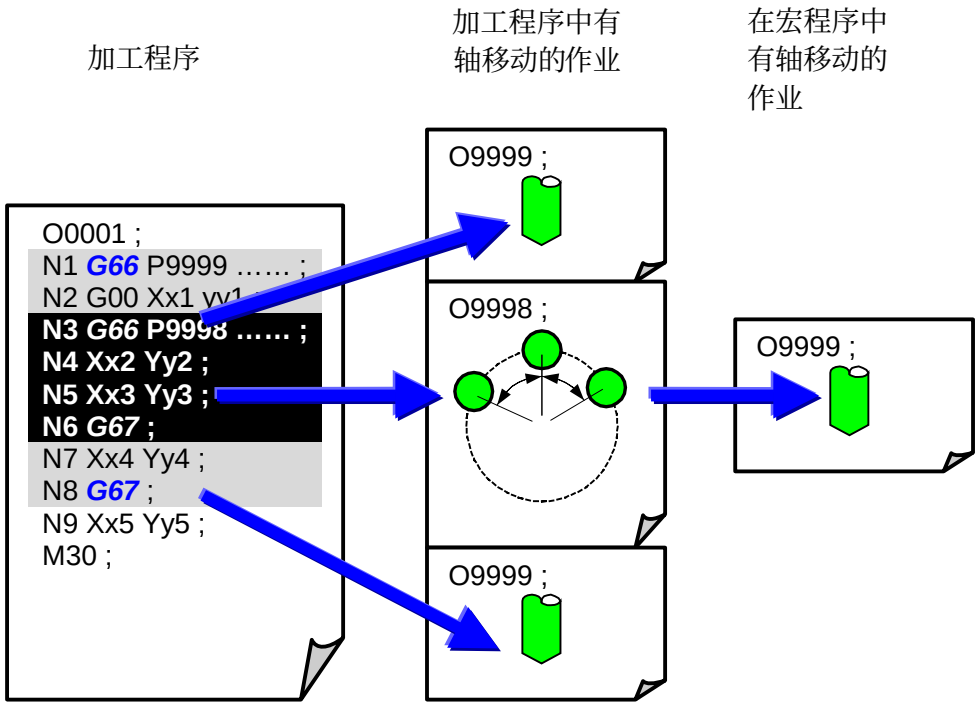
📖 宏程序调用的嵌套级数G65和G66(G66.1)合在一起共4重。

● 使用例子

●这是多重调用钻孔加工循环和螺栓孔循环的例子。

首先编制好钻孔循环  和螺栓孔循环  两个宏程序。

●从加工程序中，多重调用孔加工循环和螺栓孔循环的动作如下：



📖 在加工程序的 部分，若指令了轴移动，则调用深孔加工循环。

在加工程序的 部分，若指令了轴移动，则调用螺栓孔循环。

在螺栓孔循环的宏程序中，若指令了轴移动，则调用深孔循环。

●使用例子

●加工程序

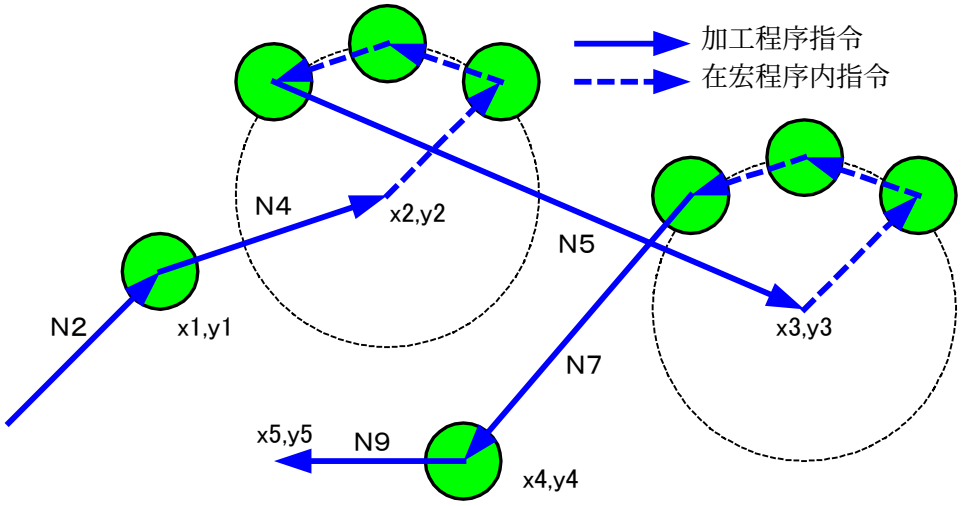
```
O0001 ;
     G92 X0 Y0 Z50. 0 ;
N1   G66 P9999 Z－20. 0 R5. 0 F300 ;.... 用模态调用孔加工循环
N2   G90 G00 X50. 0 Y30. 0 ; ..... 移动后加工孔
N3   G66 P9998 U100. 0  A45. 0 B45. 0 H3 ;
                                ..... 用模态调用螺栓循环
N4   X100. 0 Y120. 0 ; ..... 加工第1个螺栓孔
N5   X300. 0 Y100. 0 ; ..... 加工第2个螺栓孔
N6   G67 ; ..... 取消调用螺栓孔循环
N7   X120. 0 Y－10. 0 ; ..... 移动后加工孔
N8   G67 ; ..... 取消调用孔加工循环
N9   X50. 0 Y－10. 0 ; ..... 只是轴移动
     M30 ;
```

●孔加工循环

```
O9999 ;
     #30＝#4001 ;
     #31＝#4003 ;
     #32＝#4109 ;
     G00 G90 Z#18 ;
     G01 Z#26 F#9 ;
     G00 Z#18 ;
     G#30 G#31 F#32 ;
     M99 ;
```

●螺栓孔循环

```
O9998 ;
     #30＝#4001 ;
     #31＝#4003;
     #15＝#5001 ; ..... 圆心的 X 坐标
     #16＝#5002 ; ..... 圆心的 Y 坐标
     #23＝#1 ; ..... 复写自变量 A
     #22＝#11 ; ..... 复写自变量 H
     WHILE[#22 GT 0] DO1 ;
     #5＝#15＋#21＊COS[#23] ; ..... 求孔的 X 坐标
     #6＝#16＋#21＊SIN[#23] ; ..... 求孔的 Y 坐标
     G90 X#5 Y#6 ; ..... 若X,Y移动 即调用孔循环
     #23＝#23＋#2 ; ..... 更新角度
     #22＝#22－1 ; ..... 孔数相减
     END1 ;
     G#30 G#31 ;
     M99 ;
```



自变量的值复制后使用

执行G66的程序段后，把自变量代入局部变量中。调用宏程序时，不设定。
在宏程序内部用于作业的自变量，将其值赋给局部变量。

每个程序段调用 （G66.1） FS15 *i*

● 功能

●从指令每个程序段调用的G66.1开始到用G67取消为止，其间每执行NC语句都调用一次宏程序。

●从G66.1到G67的所有NC指令都由自变量赋值给宏程序。

📖 NC指令在执行前，由自变量给宏程序赋值。

📖 M98(调用子程序)，M99只调子程序，不调宏程序。

● 书写格式

●在进行每个程序段调用范围的开头指令G66.1，调用范围的最后的指令G67。

●在G66.1后面，指令调用的宏程序程序号(P)，循环次数并给自变量赋值。

```
O0001 ;
    :
G66. 1 Pp <自变量> ;..... 每个程序段的调用指令
    :
( NC语句) ..... 在NC语句中调用宏程序。自变量赋值。
    :
G67 ; ..... 取消每个程序段调用功能。
    :
M30 ;
```

每个程序段调用	G66. 1 Pp Ll <自变量> ;
	P：调用用户宏程序的程序号 **L**：循环次数 自变量：给用户宏程序中局部变量赋予的数值

📖 G66.1的书写格式与简单调用(G65)相同。

📖 在G66.1程序中，也可调用宏程序。

🕮 每个程序段调用的范围（从G66.1的下一个程序段到G67的前一个程序段）中，G(#10)，L(#12)，P(#16)也为自变量。
但是，当指令了多个 G 代码时，最后的G代码成为自变量。并且，模态G代码作为模态信息处理。

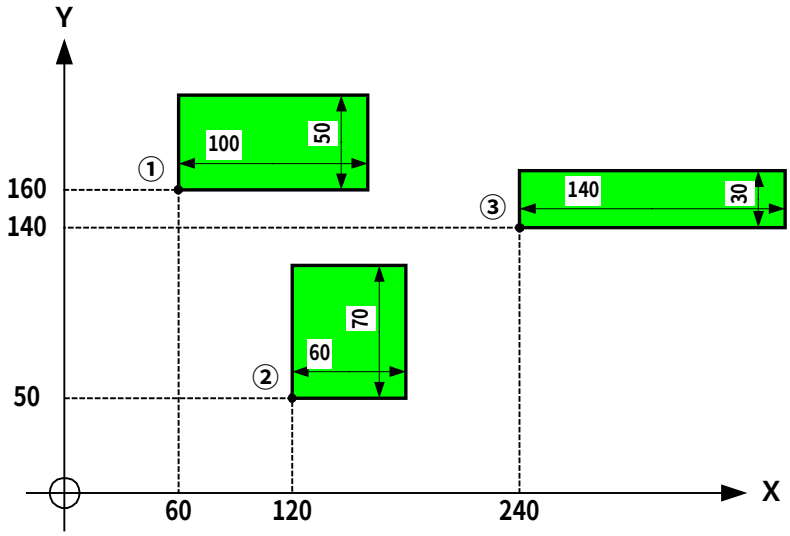
(例)G00 G90...被指令时，G90被代入#10。并且G00和G90作为模态信息处理，是快速进给，绝对坐标方式。

🕮 在O,N以外的地址后面指令了N时，当作自变量#14。

(例) N100 G00 X10.0 N50；在这个程序中，
N100是顺序号，N50是自变量。

● 使用例

加工多个边长不同的矩形的程序。



●加工程序

```
O0001 ;
N11   G66.1 P9999 ;
N12   G00 G90 X60.0 Y160.0 U100.0 V50.0 ;
N13   X120.0 Y50.0 U60.0 V70.0 ;
N14   X240.0 Y140.0 U140.0 V30.0 ;
N15   G67 ;
      M30 ;
```

●宏程序

```
O9999 ;
      #30=#4001 ;
      #31=#4003 ;
      #32=#4109 ;
N1    G00 X#24 Y#25 ;  ..... 定位在基准点
N2    G01 G91 X#21 F300 ;
N3    Y#22 ;
N4    X-#21 ;
N5    Y-#22 ;
      G#30 G#31 F#32 ;
      M99 ;
```

- 在N11的G66.1程序段中也调用程序。但因自变量全部为空，所以轴不移动。

- 从G66.1到G67范围的X,Y值也作为自变量赋值给宏程序。在宏程序N1中，定位在矩形的基准点上。

- 宏程序执行前可以修改加工程序的指令值，所以可以有各种各样的应用。

  例如:把N1 写成 G00 X#25 Y#24；时可以替换X,Y坐标。

例题 ： 统计孔的个数（G66. 1）

●这是使用每个程序段调用(G66.1)的程序实例。对在固定循环(G80～G89)方式中加工孔的数量进行计数。

●变量的意义如下:

#500 ..... 刀具寿命值

#501 ..... 孔数的累积值

●程序例子

```
O9999 ;
      X#24 Y#25 R#18 Z#26 F#9 L#12 ;
                                  ..... 给自变量赋值
      IF[#4009 EQ 80] GOTO 2 ; ..... 若不是固定循环 转到N2
      IF[#12 EQ 0] GOTO2 ; ..... 若是L0 转到N2
      IF[#12 NE #0] GOTO1 ; ..... 指定L时 转到N1
      #12=1 ; ..... 没指定 L时 置为L1
N1    #501=#501+#12 ; ..... 更新累积值
      IF[#501 LE #500] GOTO2 ; ..... 不到寿命 转到N2
      M00 (LIFE OVER) ; ..... 呼叫操作者
      #501=0 ; ..... 清除累积值
N2    M99 ;
```

●动作确认

执行下列程序，确认动作。

```
O0001 ;
G92 X0 Y0 Z100. 0 ;
G66. 1 P9999 ;
G81 G91 X10. 0 Y20. 0 R-90. 0 Z-150. 0 F300 L3 ;
X40. 0 Y30. 0 ;
G90 X100. 0 Y80. 0 ;
G80 ;
G67 ;
G00 G90 X0 Y0 Z100. 0 ;
M30 ;
```

●习题

修改程序，使每把刀具号都可以统计加工的孔数。

用G,M,T代码调用宏程序

本章学习用参数设定的G,M,T代码调用宏程序的方法。在用G65和G66确认动作后设定参数。该方法使调用变得简单，可用于加工程序的互换性和自动化。

本章介绍下列内容:

- **用G,M代码调用宏程序**
- **用M,T,S,B 代码调用子程序**

用G,M代码调用宏程序

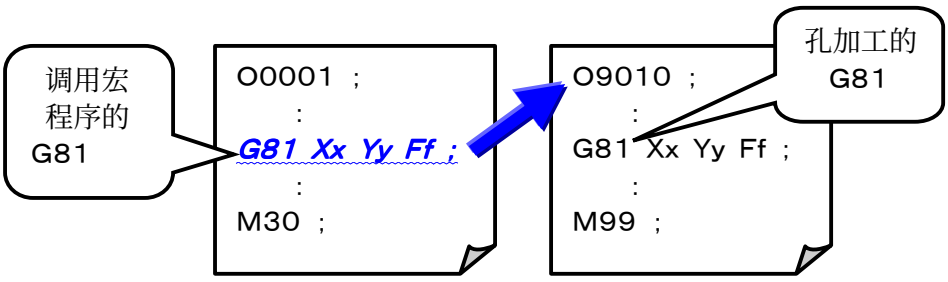
● 用G,M代码调用宏程序

●把调用宏程序的G代码，M代码事先设定在参数上，可以调用(G65)宏程序09010～09029。

📖 例如，用G81调用宏程序09010的参数事先设好后，下面的两个程序段进行同样的动作。

```
G65 P9010 X100. 0 Y50. 0 R3. 0 Z－50. 0 F300 ;
        ↓
G81 X100. 0 Y50. 0 R3. 0 Z－50. 0 F300 ;
```

●调用宏程序的 G代码设定了 G81时，加工程序的 G81就是调用宏程序的命令。被调出的宏程序中的 G81就是孔加工的 G81。



●调用宏程序的代码种类和调用的程序号与设定的参数号之间关系如下所示。

调用代码	程序号	参数号	
		FS16*i*	FS15*i*
用 G 代码调用宏程序	O9010	6050	7050
	O9011	6051	7051
	O9012	6052	7052
	O9013	6053	7053
	O9014	6054	7054
	O9015	6055	7055
	O9016	6056	7056
	O9017	6057	7057
	O9018	6058	7058
	O9019	6059	7059

📖设定值的范围是 1～999。

📖 **FS15*i***
若给 G代码调用的参数设定负值，则为模态调用。

模态调用的方法根据参数 7000#3的设定可以选择 G66.1任意一种。

调用代码	程序号	参数号	
		FS16*i*	FS15*i*
用 M 代码调用宏程序	O9020	6080	7080
	O9021	6081	7081
	O9022	6082	7082
	O9023	6083	7083
	O9024	6084	7084
	O9025	6085	7085
	O9026	6086	7086
	O9027	6087	7087
	O9028	6088	7088
	O9029	6089	7089

📖 设定值的范围是 1～99999999。

●自变量的指定，与通常宏程序调用相同，有自变量指定Ⅰ和自变量指定Ⅱ两种。根据所使用的地址，可以自动地判别。

●地址 P 也可以作为自变量使用。

●在用 G 代码，M代码调用的宏程序中，不能再次用G代码、M代码调用，须用G65或G66调用其他的宏程序。

FS16*i* 中所用的 G代码模态调用

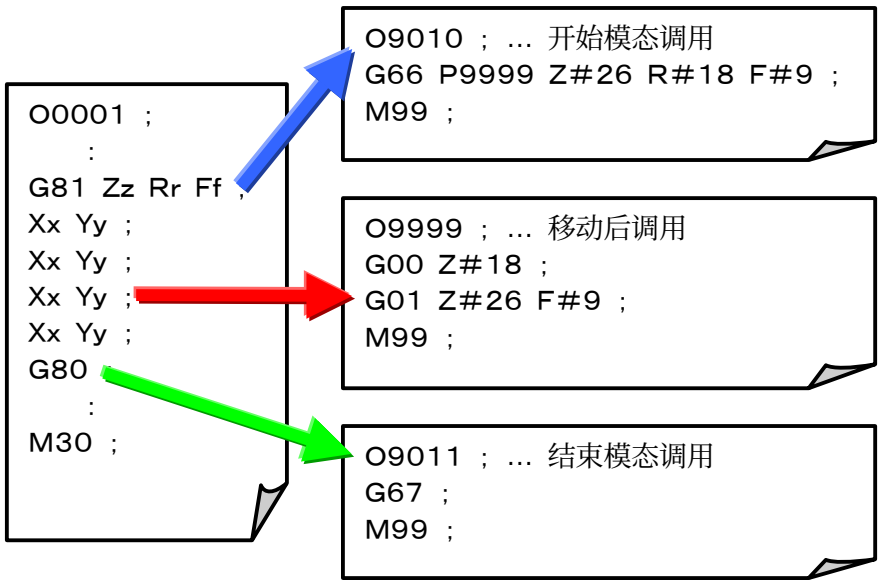
在FS16*i*中，也可以使用G代码模态调用。

●当用G81为模态调用开始，G80为取消模态调用时...

❶ 设定用G81进行G代码调用(例O9010)的参数。

❷ 设定用G80进行G代码调用(例O9011)的参数。

❸ 编程如下:



● 带小数点的G代码调用 FS15*i*

●若设定下面参数，可以用带小数点的G代码调用宏程序。

		#7	#6	#5	#4	#3	#2	#1	#0
参数	7002			DPG					

#5(DPG)　0 ： 不使用带小数点的G代码调用宏程序。
1 ： 使用带小数点的G代码调用宏程序。

调用代码	程序号	参数	
		FS16*i*	FS15*i*
用小数点的 G 代码调用	O9040	—	7060
	O9041	—	7061
	O9042	—	7062
	O9043	—	7063
	O9044	—	7064
	O9045	—	7065
	O9046	—	7066
	O9047	—	7067
	O9048	—	7068
	O9049	—	7069

📖 把去掉小数点的值（例 G12.3时为123）设定在参数上。

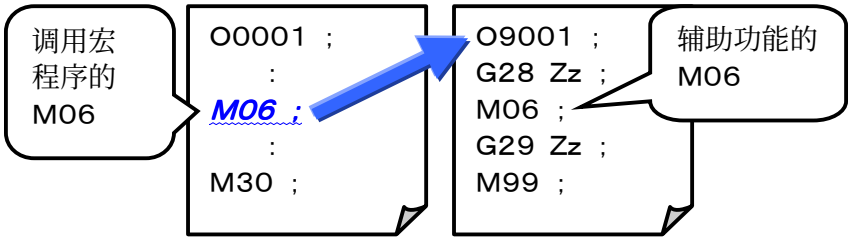
用M,T,B,S代码调用子程序

● 用M代码调用子程序

●事先在参数上设定调用宏程序的 M代码，程序中可用该 M代码和子程序调用指令(M98)一样调用宏程序。

📖 例如：若在M06中，事先设定了调用09001程序的参数，则下面两个程序的工作是一样的。

```
M98 P9001 ;
   ↓
M06 ;
```



●因为是调用子程序，所以不能指定自变量。

●宏程序号与设定的参数号之间关系如下表。

调用代码	程序号	参数号	
		FS16 *i*	FS15 *i*
M 代码 子程序	O9001	6071	7071
	O9002	6072	7072
	O9003	6073	7073
	O9004	6074	7074
	O9005	6075	7075
	O9006	6076	7076
	O9007	6077	7077
	O9008	6078	7078
	O9009	6079	7079

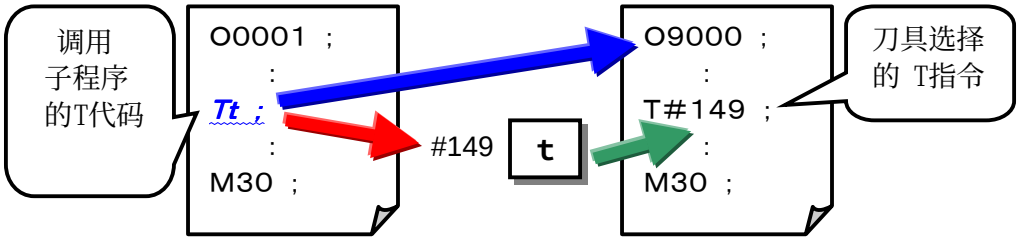
📖 设定值的范围是 1～99999999。

● 用T,S,B代码调用子程序

●若事先设定下面的参数，则在加工中执行T,S,B指令时，可调用各代码对应的宏程序。

📖 例如：若使用 T代码调用子程序，则加工程序中执行刀具选择指令 Tt时就调用 O9000。

●因为是子程序调用，不能指定自变量。则加工程序中指定的T,S,B代码由公共变量(#146,#147,#149)输入。



●宏程序和需要设定的参数号之间关系如下表。

调用方法	程序号	参数		相关的公共变量
		FS16*i*	FS15*i*	
T 代码子程序	O9000	6001#5	7000#0	#149
B 代码子程序	O9028	—	7000#2	#146
S 代码子程序	O9029	—	7000#1	#147

📖 用 B 代码调用子程序时，O9028 不能再用于 M 代码调用子程序。

📖 用 S 代码调用子程序时，O9029 不能再用于 M 代码调用子程序。

●在同一程序段中，不能同时指令用 M,T,S,B 代码调用子程序。

●用 G,M 代码调用的宏程序中或用 M,T,S,B 代码的子程序调出的宏程序中，不能再使用 M,T,S,B 代码调用子程序。

例题 ： 刀具使用时间
～ M 代码的子程序调用 ～

●计算每把刀具使用时间的程序。

●把 1～5 号刀具的使用时间代入宏变量中。

1 号刀具..... #501

2 号刀具..... #502

3 号刀具..... #503

4 号刀具..... #504

5 号刀具..... #505

●刀具的使用时间用加工程序的指令M03开始计数。用M05结束计数。

另外，使用时间用系统变量#3002(自动运行时间)计时。

●程序实例

●用M03调用宏程序

```
O9001(M03) ;
      M01 ;
      IF[#4120 LE 0] GOTO 9 ; ..... 没指定刀具号时
      IF[#4120 GT 5] GOTO 9 ; ..... 大于5的刀具号无效
      #3002=0 ; ..... 清除自动运行时间
N9    M03 ; ..... 指令主轴正转
      M99 ;
```

●用M05调用宏程序

```
O9002(M05) ;
      M01 ;
      IF[#4120 LE 0] GOTO 9 ; ..... 没指定刀具号时
      IF[#4120 GT 5] GOTO 9 ; ..... 大于5的刀具号无效
      #[500+#4120]=#[500+#4120]+#3002 ;
                                ..... 更新刀具使用时间（注）
N9    M05 ; ..... 指令主轴停止
      M99 ;
```

🕮 以分为单位管理时，变更为 #[500+#4120]+#3002*60 。

● 动作确认

1. 与用 M03,M05调用子程序的方法调用宏程序一样，设定参数。

FS16*i*

参数	6071	用子程序调用O9001 的 M 代码

设定 3

参数	6072	用子程序调用O9002 的 M 代码

设定 5

FS15*i*

参数	7071	用子程序调用O9001 的 M 代码

设定 3

参数	7072	用子程序调用O9002 的 M 代码

设定 5

2. 显示宏程序画面把刀具使用时间清零。

3. 因实习机上没有刀具交换装置(ATC)，所以把辅助功能忽略信号

(信号名 ： AFL)置于 ON。

> 若把辅助功能忽略信号置于 1时，虽然指令了 M,S,T功能，机床也不动作。

4. 执行下列程序，确认动作。

```
O0001 ;
T1 M06 ; ..... 把刀具号变更为 1～5后，确认动作。
M03 ;
G04 X10.0 ;
M05 ;
M30 ;
```

● 习题

修改程序，使它能满足下述要求：当使用刀具时间的累积值超过了刀具寿命值时，设置程序停(M00)，并呼叫操作者。

例题 ： 加工零件的计数
～ 用 M 代码调用子程序 ～

●这是通过每执行一次M30，来给加工零件计数的程序。

📖 此程序中，加工零件用的计数器，使用宏变量#501。

●当加工零件数大于所要求的零件数时，给PMC发信号，通知已达到了加工的零件数。

●宏程序

```
O9003 ;
      M01 ;
      #1100=0 ; ..... 达到加工零件数的信号清零
      #501=#501+1 ; ..... 更新加工件数
      IF[#501 LT #500] GOTO 9 ; ..... 若没达到所要求数时 转N9
      #1100=1 ; ..... 送出达到加工零件数信号
N9    M30 ; ..... 程序真正结束
```

📖 送出达到加工零件数的方法，根据各机床实际情况相应变更。

●动作确认

1. 如同用M30通过子程序调用来调出用户宏程序 O9003一样，设定参数。

FS16*i*

参数	6073	用子程序调用 O9003 的 M 代码

设定 30

FS15*i*

参数	7073	用子程序调用 O9003 的 M 代码

设定 30

2. 用下列程序，确认动作。

```
O0001 ;
G04 X5.0 ;
M30 ; ..... 为加工零件计数
```

例题 ： 简单的刀具管理
～ 用 T 代码调用子程序 ～

● 说明

●把用途相似的刀具作为一组，事先登录，当刀具的使用次数达到寿命时，作为同一组内的备用刀具使用，编制简单地刀具管理程序。

●此程序中，刀具组有8个，把每组刀具号和刀具寿命值（使用次数）登录在宏变量中。
组号与变量号的对应如下表所示。

组 号	刀具号使用的变量	刀具寿命使用的变量
1	#500	#508
2	#501	#509
3	#502	#510
4	#503	#511
5	#504	#512
6	#505	#513
7	#506	#514
8	#507	#515

●一组中能够登录的刀具号最多为4把。
对于宏变量，在小数点以前，每2位数设置1把刀具号。
达到刀具寿命的刀具，在宏程序中，刀具号移到小数点以后。

（例）1组的刀具号为1、2、3时，则在#500的变量上设定030201。

📖 按距小数点近的刀具号顺次使用。

（例）例如 1组的1号刀具已达到寿命时，则变量#500变为0302.01。

●各组的刀具寿命值最大为999。小数点以后表示刀具的使用次数。

●加工程序中选择刀具时，若指令了T101～T108，则进行刀具组管理，选择组内尚未达到寿命的刀具。但是，当指令了T1～T99时，则不进行刀具组管理，直接选择它的刀具。

●宏程序

```
O9000 ;
      IF[#149 LE 100] GOTO8 ; ..... T1～T100直接指定号码
      #1=500+#149-101 ; ..... 算出刀具组用的变量号
      #2=508+#149-101 ; ..... 算出刀具受命的变量号
      #3=FIX[#[#2]] ; ..... 指定的组的寿命值
      #4=ROUND[[#[#2]-#3]*1000] ; 算出现在为止的累计寿命值
      IF[#3 GT #4] GOTO2 ; ..... 若没达到寿命 转到N2
      #[#1]=#[#1]/100 ; ..... 更新刀具数据
N1    #[#2]=FIX[#[#2]] ; ..... 清除累积刀具寿命
N2    #5=FIX[#[#1]] ; ..... 算出下一把刀具号 (1)
      #6=#5-FIX[#5/100]*100 ; ..... 算出下一把刀具号 (2)
      IF[#6 GT 0] GOTO3 ; ..... 若有下一把刀 转到N3
      M00 (LIFE OVER) ; ..... 使用寿命结束 要求换刀
      GOTO1 ; ..... 再找下一把刀
N3    #[#2]=#[#2]+0.001 ; ..... 更新刀具寿命值
      T#6 ; ..... 选择找到的刀具
      M99 ;
N8    T#149 ; ..... 直接选择刀具(T1～T100)
      M99 ;
```

●动作确认

1. 用 T 代码调用子程序 09000，事先设定参数。

FS16*i*

	#7	#6	#5	#4	#3	#2	#1	#0
参数 6001			TCS					

#5(TSC) 0 : 用 T 指令选择刀具。
1 : 用 T 指令调用子程序 09000。

FS15*i*

	#7	#6	#5	#4	#3	#2	#1	#0
参数 7000								TCS

#0(TCS) 0 : 用 T 指令选择刀具。
1 : 用 T 指令调用子程序 09000。

2. 在宏变量上设定每组刀具的刀具号和刀具寿命值。

刀具号		刀具寿命	
#500	030201. 000	#508	2. 000
#501	060504. 000	#509	3. 000
#502	090807. 000	#510	4. 000
#503	131211. 000	#511	5. 000
#504	1514. 000	#512	8. 000
#505	16. 000	#513	10. 000
#506	1191817. 000	#514	2. 000
#507	20. 000	#515	25. 000

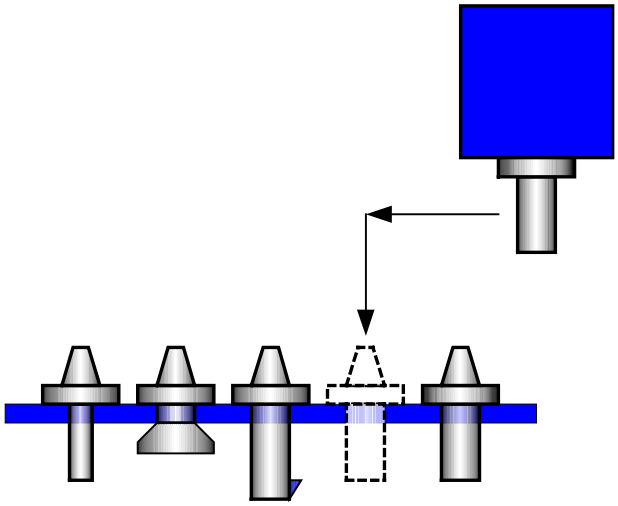
📖 为了确认动作，先设置刀具寿命小的值。

3. 执行下面程序，确认动作。

```
O0001 ;
T101 ; ..... 选择 01组的刀具
G04 X5. 0 ;
T1 ; ..... 直接选择 T1
G04 X5. 0 ;
M30 ;
```

例题 ： 移动式刀具交换装置
～ 用 T 代码调用子程序 ～

●编制宏程序，使之能够实现如图所示的刀套排列的换刀。



宏变量	用 途	设定
#500	装在主轴上的刀具号	—
#501	刀套 1 的 X 坐标	100. 000
#502	刀套 2 的 X 坐标	151. 200
#503	刀套 3 的 X 坐标	200. 100
#504	刀套 4 的 X 坐标	249. 500
#505	刀套 5 的 X 坐标	300. 500
#506	刀套的 Y 坐标	100. 000
#507	刀套的 Z 坐标	0. 000

🕮 刀套的坐标用机械坐标设定。

🕮 当刀套不是直线时，应修改程序，使每个刀套都有确定的Y,Z坐标。

●宏程序

●刀具交换宏程序

```
O9000 ;
      IF[#149 GT 5] GOTO9 ; ..... 刀具号超出范围 转到N9
      #30=#4001 ;
      #31=#4003 ;
      #32=#4109 ;
      #3=#5003 ; ..... 存储 Z 轴绝对坐标
      #4=#4008 ; ..... 存刀具长度补偿用 G 代码
      IF[[#500+0] EQ 0] GOTO 1 ; ..... 无返回刀具时 转到N1
      #1=#[500+#500] ; ..... 返回刀具的刀套 X 坐标
      #2=40 ; ..... 松开指令(M代码)
      M98 P9998 ; ..... 把主轴上的刀具返回到刀套
      #500=0 ; ..... 清除主轴的刀具号
N1    IF[#149 EQ 0] GOTO 2 ; ..... 指令T0时 转移到N2
      #1=#[500+#149] ; ..... 指令刀具刀套的 X 坐标
      #2=41 ; ..... 夹紧指令(M代码)
      M98 P9998 ; ..... 取刀具指令
      #500=#149 ; ..... 存主轴的刀具号
N2    G#30 G#31 G#4 F#32 ;
      M99 ;
N9    #3000=1 (ILLEGAL TOOL NUMBER) ;
```

●坐标轴移动的子程序

```
O9998 ;
      G00 G90 G53 X#1 Y#506 ; ..... X,Y轴移到刀套上方
      G49 G53 Z[#507+50] ; ..... Z 轴趋近
      G01 G53 Z#507 F100 ; ..... Z 轴下降
      M#2 ; ..... 松开/夹紧刀具
      G00 G#4 Z#3 ; ..... Z 轴返回
      M99 ;
```

误差与缓冲

关于宏程序的编制方法已学习很多了。现在学习宏程序中最需要注意的运算误差和缓冲(先读)功能的处理。

本章介绍内容如下:

- **误差的处理方法**
- **缓冲的处理方法**

运算误差

● 运算误差

●运算时都有运算误差。运算的种类和运算误差大致如下表。

运算的种类	平均误差	误差的种类
a=b*c	1.55×10^{-10}	相对误差
a=b/c	4.66×10^{-10}	
a=SQRT[b]	1.24×10^{-9}	
a=b+c a=b−c	2.33×10^{-10}	
a=SIN[b] a=COS[b]	5.0×10^{-9}	绝对误差（度）
a=ATAN[b]/[c]	1.8×10^{-6}	

🕮 关于最大误差，请见说明书。

🕮 相对误差是指运算结果的误差比率。

🕮 宏变量有效位数是8位。

● 不使用 EQ,NE

●进行运算时，有时运算结果有误差。
在条件表达式中，尽可能不用EQ,NE，而使用GE或LE。

> 🕮 #1从10到9,8.....计数下去，直到为0结束时，表达式不写成 IF[#1 EQ 0]而写成 IF[#1 LF 0]。

●通常要小心表示误差的条件式。

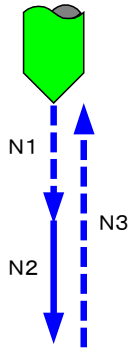
下例中，#1和#2 认为大致相同的。

(例) IF[ABS[#1－#2] LE 0.001] GOTO n ；

> 🕮 下划线部分，根据允许值来决定。

● 用绝对坐标编程

●执行下列程序时，因为是增量指令，所以不能返回到加工起始点。




```
O0001 ;
     #1=2.3456 ; ..... 作为运算结果(1)
     #2=3.4567 ; ..... 作为运算结果(2)
N1   G00 G91 Z#1 ; ..... 用快速进给趋近
N2   G01 Z#2 F300 ; ..... 切入
N3   G00 Z-[#1+#2] ; ..... 返回
```

> 🕮 在车床系中，把 G91 Zz部分变成 Wz。

●执行程序，测量 N1～N3的移动量。

```
N1 G00 G91 Z#1 ; ..... 移动量
N2 G01 Z#2 F300 ; ..... 移动量
N3 G00 Z-[#1+#2] ; ..... 移动量
```

●引用变量值时，变量值根据各地址的单位，自动地进行四舍五入。上面的程序与下面的程序是相同的。

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

●N1和N2相加的移动量应该用语句：ROUND[#1]+ROUND[#2]，四舍五入后再相加。而在返回的程序段N3中，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...

根据程序的不同，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，或者数据的程序号为0000时，结束运行。

● 宏程序

●作业变量

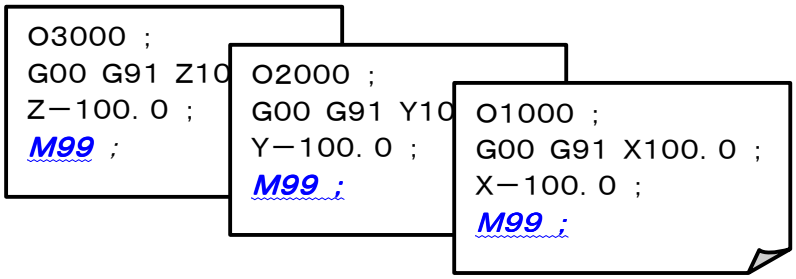
#1 ：调度数据用指针
#2 ：取出的程序号
#3 ：取出的要求零件数
#4 ：加工完零件数的计数器

●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内部变换成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: 1000.10000038...

●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: ~~1000~~.10000038...

●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程序端时，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 G04 ;	↓	↓
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下个程序段，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...

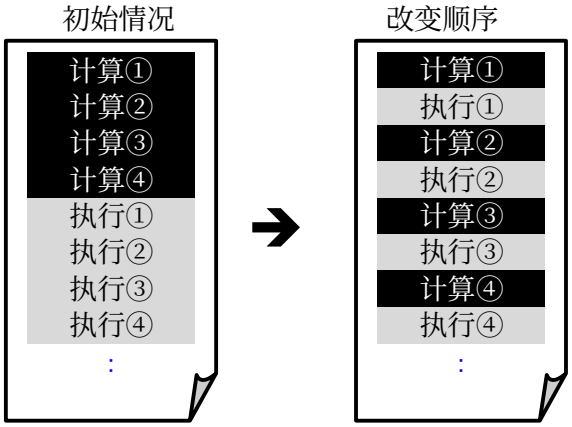
请注意，下面的系统变量要受到缓冲的影响。

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①)，指令轴移动。
下一个计算(计算②)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）

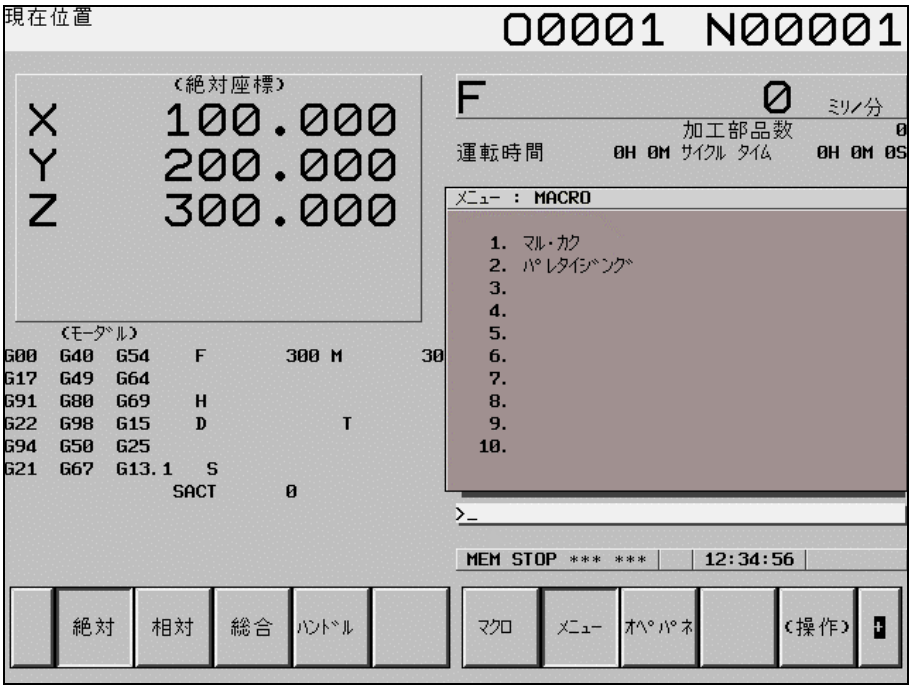
通常，给宏程序赋值的数据是在加工程序中作为自变量指定的。

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 OFFSET SETTING，软键 Menu，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。

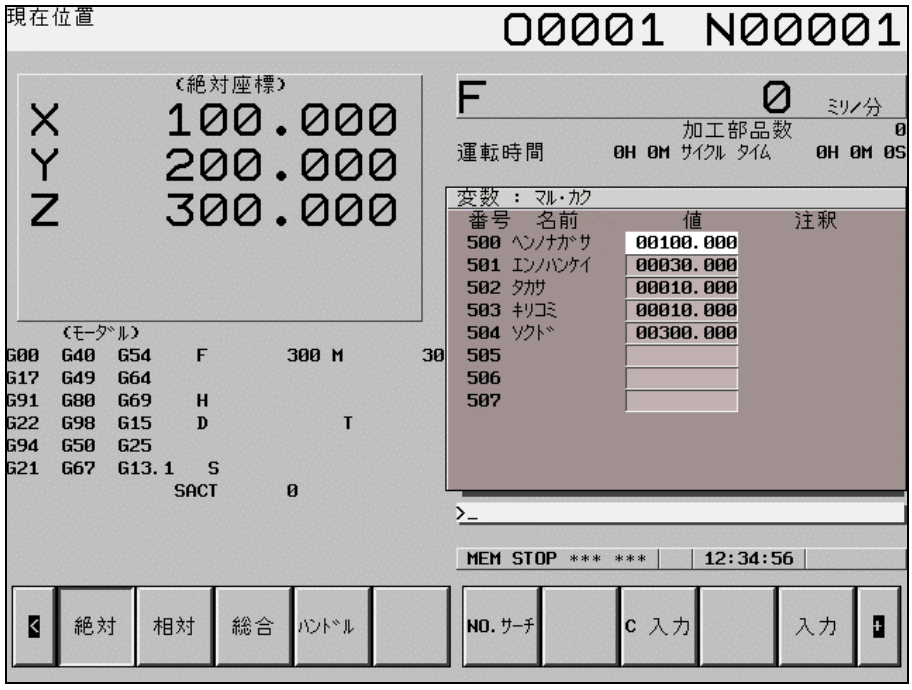
●输入菜单号，按软键 选择，显示宏变量画面。

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。



🕮 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 CYCLE START 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

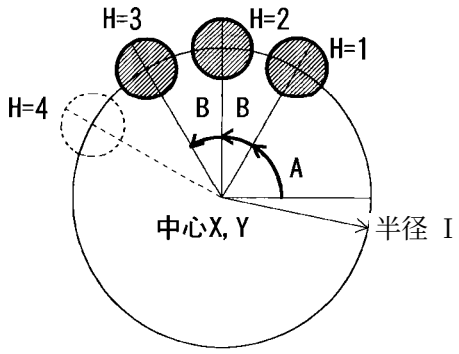
P 存储结果的变量号

● 例题 ： 螺栓孔加工循环

在半径为I的圆周上，从初始角A度开始，加工H个孔，各孔之间间隔为B度。

圆的中心位置为X,Y，绝对指令，增量指令都可以使用。

另外，孔加工的方向为顺时针时，B角度为负值。



X ：	圆中心的X坐标（绝对/增量指令均可）	(#124)
Y ：	圆中心的Y坐标（绝对/增量指令均可）	(#125)
Z ：	孔深	(#126)
R ：	趋近点坐标	(#118)
F ：	切削进给速度	(#109)
I ：	圆的半径	(#104)
A ：	钻孔的起始角度	(#101)
B ：	孔间隔角度（指令负值时为顺时针转）	(#102)
H ：	孔的个数	(#111)

程序实例

宏程序 A

```
O0001;
   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;
```

宏程序 B

```
   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;
```

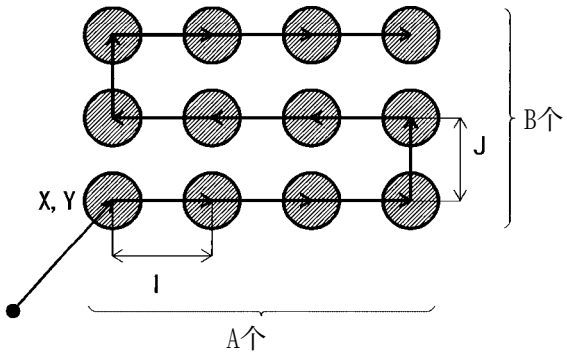
动作确认

```
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

如图所示，从X,Y的位置开始，在X轴方向上加工间隔为I的A个孔，在Y轴方向上加工间隔为J的B个孔。

加工方向根据I,J符号变化。



X ： 开始位置的X坐标	(#124)	I ： X轴方向孔间距	(#104)
Y ： 开始位置的Y坐标	(#125)	J ： Y轴方向孔间距	(#105)
Z ： 孔深	(#126)	A ： X轴方向孔数	(#101)
R ： 趋近点高度	(#118)	B ： Y轴方向孔数	(#102)
F ： 切削进给速度	(#109)		

程序实例

宏程序A

```
O0001;
   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;
```

宏程序B

```
   #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;
```

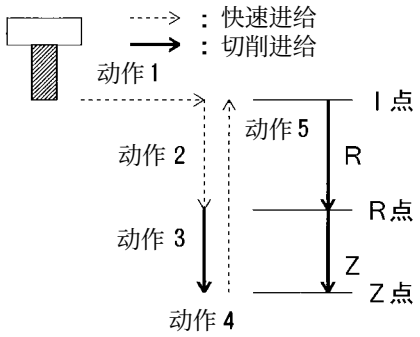
确认动作

```
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

与固定循环 G81(钻削循环)同样动作的宏程序。

固定循环的基本动作如下:
动作1 ： X,Y轴定位
动作2 ： 快速进给到R点
动作3 ： 切削进给到Z点
（动作4 ： 在孔底的动作）
动作5 ： 快速进给到R点或到I点



X ： 孔的X坐标	(#8024)	R ： I点到R点的距离	(#8018)
Y ： 孔的Y坐标	(#8025)	F ： 切削进给速度	(#8009)
Z ： 从R点到Z点距离	(#8026)	L ： 循环次数	

X和Y指令用绝对/增量 指令都可以。
但在同一个程序段内, 不能混合使用。
Z和R只能用增量指令。

程序实例

宏程序A

```
O0001;
   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;
```

宏程序B

```
   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;
```

确认动作

```
O0002;
   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					

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。