指令说明

本手册中按照字母顺序列出了控制器全部的保留字。下面是针对每个保留字说明信息的描述。

保留字的操作码在左上角。操作码下面就是对指令的解释。解释的内容包括功能、说明、参数、应用限制、默认值、相关内容与竟用举例。

参数

轴参数

一些指令需要用户指定针对哪个或哪些轴起作用。这些指令本身就是以大写字母 A、B、C、D 这些轴名称作为参数的。这些参数之间不需要逗号，也不要插入空格，顺序也没有关系。正确的格式是在指令与轴参数之间用一个空格分开。如果不需要参数或不给出参数，则指令对所有轴起作用。下面是正确使用轴参数的格式：

MO A	电机关闭，A 轴
MO ABD	电机关闭，A 轴、B 轴、D 轴
MO ACD	电机关闭，A 轴、C 轴、D 轴
MO ABC	电机关闭，A 轴、B 轴、C 轴
MO BCAD	电机关闭，A 轴、B 轴、C 轴、D 轴
MO	电机关闭，所有轴（A 轴、B 轴、C 轴、D 轴）

在参数说明中，轴参数会以“x”来表示。

数值参数

一些指令后面需要数值参数。在参数说明中，通常会用字母 n、m、o、p、q、r 等表示数值参数，这些字母表示数值。对于这样的指令，多个参数之间用逗号分开，参数的次序非常重要。如省略了中间某个参数，相应的逗号不能省略，否则控制器会错误理解之后参数位置。下面是正确使用数值参数的格式，其中 n 在实际使用中要由实际的数值来代替。

AC n	为 A 轴指定参数
AC n, n	为 A 轴、B 轴指定参数
AC n, , n	为 A 轴、C 轴指定参数
AC n, n, n, n	为 A 轴、B 轴、C 轴、D 轴指定参数

也有一些指令的参数之间不完全是由逗号分开，部分参数由 < 或 > 将其与前面的参数分开。 < 与 > 既用于分开参数，本身也说明了参数的意义，当省略该参数时，则必须连同参数前的字符一起省略。例如：

LI n, n, n, n<o>p

如省略参数 o，则应为

LI n, n, n, n>p

还有一种特殊情况，是以其它轴作为参数。这种情况的应用格式与其它数值参数的格式是相同的，区别近在于参数不是数值。例如：

GA C, C, , C 设置 A、B、D 轴以 C 轴为电子齿轮同步主轴。

直接命令参数

对于直接设置各轴电机及运动状态的指令，还有另一个指定数据的方式：在轴名称后面跟等号的方式为单个轴设置数据。符号“*”可以用来代替轴名称。使用“*”表示为所有轴指定同一个数据。例如：

PRB=1000	设置 B 轴的数值为 1000
PR*=1000	设置所有轴（A 轴、B 轴、C 轴、D 轴）的数值为 1000
GAB=A	设置 B 轴以 A 轴为电子齿轮同步主轴

查询

大部分指令可以接受问号“?”作为数值参数。这个参数使控制器返回指令说明中列出的数值。指令后面为每个需要的轴跟一个“?”，格式与上面介绍的数值参数一样，只是用“?”代替了数值。

PR ?	控制器将返回 A 轴的 PR 值
PR , , , ?	控制器将返回 D 轴的 PR 值
PR ?, ?, ?, ?	控制器将返回 A 轴、B 轴、C 轴、D 轴的 PR 值
PR*=?	控制器将返回所有轴（A 轴、B 轴、C 轴、D 轴）的 PR 值

操作数应用

操作数可以理解为对应控制器状态、参数的内部变量。大部分指令有一个相应的操作数。操作数应用是描述使用操作数的格式与操作数的意义。操作数可以当作数值用于任何有效的 DMC 表达式。要输出操作数的值，用户可以使用指令

MG ‘操作数’

所有指令对应的操作数以符号“_”开始。例如，下面的指令可以将 A 轴当前的编码器回馈位置的值赋给用户变量 V。

V=_TPA

控制器本身还有一些专用的操作数，它们本身并没有相对应的指令，这些操作数通常并不以“_”开始。

应用说明

应用说明是指令应用中的限制。很多指令只能在一定的条件下才能使用。以下是对这些条件的解释：

“运动中”：该指令是否可以在相关轴正在运动时使用

“程序中”：该指令是否可以在控制器存储的 DMC 用户程序中使用

“指令行”：该指令是否可以由通讯接口输入使用

默认值说明

说明中的“默认值”指出控制器中该指令参数的默认值。这些值用户可以修改。新的值会被 BN 指令保存在控制器的非易失存储器中。如果新的参数值没有保存在控制器的非易失存储器中，在系统复位时原默认值会被恢复。以下情况系统会复位：控制器开机；按下控制器的复位（RESET）键；控制器收到 RS 指令。

对于一些动作性的指令、或者是函数等其它类型的保留字，是不存在相关默认值的。

默认格式是指查询指令或指令用于查询询时回馈数据的格式。格式通常以“整数字元数”+“小数点”+“小数字数”的方式说明。或者是“数值格式”，表示使用 VF 指令定义的格式；“位置格式”表示使用 PF 指令定义的格式。预设格式并不一定意味着该数值的范围与分辨率。

对于不可查询的指令或者是函数等其它类型的保留字，是是不存在相关预设格式的。

相关指令

这里会列举一些与该保留字作用有关系的其它保留字。对比、参照这些相关指令的内容，有助于该指令作用的理解。

举例

有时，几百字的说明不如一个简单的例子更有助于理解。对于大部分保留字，都提供了应用举例，大部分例子都有注释与说明。

在例子中，控制器对指令的回馈、输出内容，以斜体表示。

# #

功能：标号

说明：#符号表示程序标号。标号名称最长为 7 个字符，经常应用于子程序或循环中。标号可以分为使用者定义标号与自动子程序标号两种。控制器内存程序中的用户定义标号可以用

LL 指令列出，也可以用指令 MG_DL 输出出当前可定义的标号数量。自动子程序标号包括 #CMDERR、#LIMSWI、#ININT、#AUTO……

程序行中，标号只能出现在行首。

参数：#nnnnnnn 其中

nnnnnnn 是标号名，最多 7 个字符，首字符必须为字母，其中不得有空格

应用：

运动中	是
程序中	是
指令行	否

相关指令：

LL	列出标号
_DL	当前可定义标号数量
JP	跳转
JS	转子程序

举例：

```
#Loop;JP#Loop, x=10          在 x 值为 10 之前等待（循环）

#Move                        定义一个使 A 轴移动的子程序
PRA=1000
BG A
AM A
EN
```

$

功能：十六进制标记

说明：

$符号表示后面跟随的字符是十六进制数字。

参数：$nnnnnnnn.mmmm

n 是最多 8 个十六进制数字（表示 32 位整数）

m 是最多 4 个十六进制数字（表示 16 位小数）

应用：	运动中	是
	程序中	是
	指令行	是
默认值：	默认值	—
	预设格式	—

相关指令：

+ - * / %	加数学运算符号
MG {$8.4}	以十六进制输出数据

举例：

x=$7fffffff.0000	将 2147483647 存入变数x
y=x&$0000ffff.0000	将 x 的低 16 位存入 y
z= x&$0000ffff.0000/$10000	将 x 的高 16 位存入 z

& |

功能：与 或，逻辑/位运算符号

说明：

符号 & 与 | 经常应用于指令 IF、JP、JS 后面的条件表达式中。它们也可以用做位运算。

参数：n&m 或 n|m

n 与 m 是带符号的数，范围为-2147483648—2147483647。

在 IF、JP、JS 指令中，n 与 m 通常是逻辑表达式，例如（x>2）

应用：

运动中	是
程序中	是
指令行	是

默认值：

默认值	—
预设格式	—

相关指令：

@COM[n]	按位求补
IF	条件判断
JP	跳转
JS	转子程序

举例：

```
IF(x>2)&(Y=4)              x 要大于 2 而且 y 必须等于 4
MG “TRUE”                  输出信息
ENDIF

:MG 1|2                    按位运算：01 或 10=11
  3.0000
:
```

( )

功能：括号（运算顺序）

说明：

括号表示数学运算或逻辑运算的顺序。注意，控制器并不按照通常理解的顺序运算。例如，乘法不会先于加法被计算。控制器总是按照从左向右的顺序运算。为避免出现运算顺序上的错误，建议尽量使用括号

参数：(n)

n 是数学或逻辑表达式。

应用：

运动中	是
程序中	是
指令行	是

默认值：

默认值	—
预设格式	—

相关指令：

+ - * / %	数学运算符
& |	条件判断

举例：

```
:MG 1+2*3                    按位运算：01 或 10=11
 9.0000
: MG 1+(2*3)                 按位运算：01 或 10=11
 7.0000
```

;

功能：分号（指令分割符）

说明：

在同一行中使用多个 Galil 指令时，指令之间要用分号分开。在程序中使用分号，可能是由于以下三种情况：

1、 在同一行中为指令添加注释，用分号将指令与注释分开（BGA;'begin motion）

2、 在同一行中写入多个指令，以便将程序压缩在有限的行数内。

3、 在执行线程内给于更高的优先权。控制器在多线程分时执行程序，是将一个线程内一行执行完后，才处理下一个线程的。

参数：n;n;n;……n

n 是 GALIL 指令。

应用：

运动中	是
程序中	是
指令行	是

默认值：

默认值	—
预设格式	—

相关指令：

NO（'）	注释

举例：

```
BGA; 'comment                添加注释

PRA-1000;BGA;AMA             节省程序行

#High
a=a+1;b=b+1                  程序执行效率更高
JP#High

#Low
c=c+1                        程序执行效率正常
d=d+1
JP#Low
```

[ ]

功能：方括号（数组下标或函数变量）

说明：

方括号用于说明数组索引或函数中的自变量：

参数：mmmmmmmm[n]

mmmmmmmm 是数组名。

n 是数组索引，0~799 之间的整数。

@mmmm[n]

@mmmm 是函数名。

n 是函数的自变量。

应用：	运动中	是
	程序中	是
	指令行	是
默认值：	默认值	—
	预设格式	—

相关指令：

DM	定义数组
QU	列出/上传数组

举例：

DM A[100]	定义一个含有 100 元素的数组
A[0]=3	第一个元素设为 3
MG A[0]	输出数组元素
QU A[]	上载数组 A 中全部元素

+ - * / %

功能：数学运算符

说明：

加、减、乘、除、求余二元数学运算符，用于变量、常数、操作数的数学运算。

参数：(n+m) 或 (n-m) 或 (n*m) 或 (n/m) 或 (n%m)

n 与 m 是-2147483648~21473647 之间的带符号数值。

应用： 运动中 是

程序中 是

指令行 是

默认值： 默认值 —

预设格式 —

相关指令：

( ) 括号

举例：

:x=((1+(2*3))/7)-2 为 x 赋值 -1

:MG 40%6 输出 40 对 6 的余数

9.0000

:

< > = <= >= <>

功能：比较符

说明：

这些符号用于 IF、JP、JS 的逻辑表达式中。比较的结果也可以用 MG 指令输出或赋值给某个变数。

< 小于
> 大于
= 等于
<= 小于等于
>= 大于等于
<> 不等于

参数：(n<m) 或 (n>m) 或 (n=m) 或 (n<=m) 或 (n>=m) 或 (n<>m)

n 与 m 是-2147483648~21473647 之间的带符号数值。

应用：

运动中	是
程序中	是
指令行	是

默认值：

默认值	—
预设格式	—

相关指令：

( )	括号
IF	条件判断
JP	跳转
JS	转子程序

举例：

```
IF(x>2)&(y=4)          x 要大于 2 而且 y 必须等于 4
MG “true”              输出信息
ENDIF
```

=

功能：等于（赋值符号）

说明：

赋值符号有三种用途：

1、在使用变量之前对变量定义并初始化变量值（x=0）

2、为变量赋予新值（x=5）

3、输出变量或数组元素的值（x= 与MG x 作用相同）

参数：mmmmmmmm=n

mmmmmmmm 为变量名，n 是-2147483648~21473647 之间的带符号数值。

应用：

运动中	是
程序中	是
指令行	是

默认值：

默认值	—
预设格式	—

相关指令：

MG	输出信息

举例：

```
:x=5                        定义并初始化变量 x 为5
:x=                         以两种方式输出变量 x 的值
  5.0000
:MG x
  5.0000
:
```

~

功能：轴变量说明符号

说明：

~为轴变量说明符号

参数：~n=m

n 为小写字母，可以是 a、b、c、d、e、f、g、h

m 是以下数字或字母

0 或 A，表示第一轴

1 或 B，表示第二轴

2 或 C，表示第三轴

3 或 D，表示第四轴

8 或 S，表示插补轴

10 或 N，表示虚拟轴

应用：

运动中	是
程序中	是
指令行	是

默认值：

默认值	—
预设格式	1.0

操作数应用：

~n 对应数值 0、1、2、3、8 或 10

举例：

~a=2; ~b=3	设置~a 代表C 轴；设置~b 代表 D 轴
PR~a=1000	设置 C 轴位移 1000 个脉冲
JG~b=9000	设置 D 轴点动速度为 9000 脉冲/秒
BG~a~b	C、D 轴开始运动

`（Ascii96）

功能：续行符号

说明：

允许应用程序中的指令超过一行最大 40 字符的限制。这个符号适用于很长的 MG 指令或者条件复杂的 IF、JP、JS 指令。

应用：	运动中	是
	程序中	是
	指令行	否
默认值：	默认值	—
	预设格式	—

相关指令：

MG	输出信息

举例：

```
IF((var100=1000)&(var101=50));MG”GO”;EL`
SE;MG”STOP”;ENDIF
```

AB

功能：急停（Abort）

说明：

AB 停止所有电机的运动而没有减速过程。如果有程序在运行，AB 指令未明确参数为 1 的情况下，也会中止程序的运行。如果某个轴设置了出错关闭功能（参见 OE 指令），AB 指令会同时关闭此轴电机。

参数：ABn

n=0　控制器中止运动与程序。

n=1　控制器只中止运动

省略参数，则参数默认为 0

应用：

运动中	是
程序中	是
指令行	是

默认值：

默认值	—
预设格式	—

相关指令：

SH	打开电机
OE	设置电机出错时关闭

举例：

AB	停止运动
OE1, 1, 1, 1	设置电机出错关闭
AB	停止运动并关闭电机
#A	标号，程序开始
JG20000	设置 A 轴点动速度
BGA	A 轴点动开始
WT5000	等待 0.5 秒
AB1	中止运动，但不中止程序
WT5000	等待 0.5 秒
SH	打开电机
JP#A	跳转到标号 A
EN	程序结束

@ABS[n]

功能：绝对值（Absolute）

说明：

得到给定数据的绝对值。

参数：@ABS[n]

n 为带符号数，范围-2147483648~2147483647。

应用：

运动中	是
程序中	是
指令行	是

默认值：

默认值	—
预设格式	—

相关指令：

@SQR[n]	平方根

举例：

:MG @ABS[-2147483647]

2147483647.0000

AC

功能：加速度（Acceleration）

说明：

AC 指令设置电机独立运动中（PA、PR、JG）的线性加速度。加速度可以在运动中更改。DC 指令设置减速度

参数：AC n, n, n, n　　或　ACx=n

n 为无符号整数，范围 1024~1073740800，单位是脉冲每秒平方。输入的数据如不是 1024 的倍数，回自动变换为不大于输入数值的 1024 倍数。如输入的数据小于 1024，则自动变换为 1024。

n=?　返回该轴的加速度设置值。

应用：

运动中	是
程序中	是
指令行	是

默认值：

默认值	256000
预设格式	10.0

操作数应用：

_ACx	该轴的加速度数值.

相关指令：

DC	减速度
IT	S 曲线加减速平滑系数

举例：

AC 150000, 200000, 300000, 400000

AC ?, ?, ?, ?

149504，199668，299008，399360

V=_ACB　　B 轴加速度值赋给变数 V

提示：现实的加速度会受限制于物理系统的性能，如电机的扭矩、负载、驱动器的电流。如果设置的加速度超出实际能够达到的数值，电机会有很带的误差。对于步进进电机，这种误差这可能意味着丢步或堵转；对于伺服电机，这种误差可能意味着报警或震荡。

@ACOS[n]

功能：反余弦（Arc cosine）

说明：

得到给定数据的反余弦角度值。

参数：@ACOS[n]

n 为带符号数，范围-1~1。如 n 大于 1，则等效于 1，如 n 小于-1，则等效于-1。

应用：

运动中	是
程序中	是
指令行	是

默认值：

默认值	—
预设格式	—

相关指令：

@COS[n]	余弦
@ASIN[n]	反正弦
@SIN[n]	正弦
@ATAN[n]	反正切
@TAN[n]	正切

举例：

```
:MG @ACOS[-1]
 180.0000
:MG @ACOS[0]
 90.0000
:MG @ACOS[1]
 0.0000
```

AD

功能：超过距离（After Distance）

说明：

AD 指令是事件触发指令。这个指令会暂停程序的执行，直到以下三个条件之一被满足。

1、电机的位置从运动的起点开始超过了指定的距离。

2、该轴运动结束。

3、如果是点动（JG）模式，电机的运动方向变为使得距离变小的方向

一次只能设置一个轴，如果同时针对多个轴给出了参数，则只有第一个参数有效。

如果在点动（JG）或位置跟踪（PT）模式，则 AD 指令用于计算距离的原点是此前最后一次改变方向的位置。

AD 指令用于计算距离的位置，是控制器规划的位置，由于控制器对输出脉冲的平滑处理（参见 KS 指令），实际输出脉冲会略滞后于规划位置。

参数：ADn, n, n, n 或 ADx=n

n 为无符号整数，范围 0~2147483647。AD 指令一次只能为一个轴指定参数。

应用：

运动中	是
程序中	是
指令行	否

默认值：

默认值	—
预设格式	—

相关指令：

AV	插补运动中向量运动超过距离
AP	超过位置
AR	超过相对距离
MF	向前运动
MR	向后运动

举例：

#A；DP0, 0	程序开始
PR 10000, 20000	设置移动距离
BG	运动开始
AD 5000	等待 A 轴位置超过 5000
MG “Halfway toA”；RPA	发 送 消 息
AD, 10000	等待 B 轴位置超过 10000
MG “Halfway toA”；RPB	发送消息
EN	程序结束

提示：AD 指令在时间上的精度为控制器工作周期的两倍，因此运动速度乘以控制器工作周期的 2 倍，就是 AD 指令的最大滞后距离。

AI

功能：等待输入（After Input）

说明：

AI 指令是事件触发指令。这个指令会暂停程序的执行，直到控制器的通用输入满足指定的条件。

参数：AI+/-n

n 为 1~8 之间的整数，代表相应的通用输入信号。n 为正数，表示控制器的指定通用输入为截止或开路状态；n 为负数，表示控制器的指定通用输入为导通状态（第 8 路输入的情况可能与此相反，取决于控制器上 ABORT 跳线的设置情况，可参阅控制器手册 。

应用：

运动中	是
程序中	是
指令行	否

默认值：

默认值	—
预设格式	—

相关指令：

@IN[n]	读取通用输入信号
II	输入中断
#ININT	输入中断处理程序

举例：

```
#A                      程序开始
AI-1                    等待输入 1 导通
SP 10000
AC 20000
PR 400
BGA
EN                      程序结束
```

提示：AI 指令会中断程序的执行直到输入信号满足条件。如不需要中断程序的执行，也可以用条件跳转指令（JP）或输入中断指令（II）来实现需要的功能。

AI 指令的时间精度同样是控制器工作周期的 2 倍。

AL

功能：设置位置锁存（Arm Latch）

说明：

AL 指令设置控制器的位置锁存功能（控制器接收编码器位置或输出脉冲位置的高速捕捉）。此功能设置后，输入信号的变化就会触发位置锁存。每个轴都有触发位置锁存的输入信号：

A 轴锁存　通用输入 1
B 轴锁存　通用输入 2
C 轴锁存　通用输入 3
D 轴锁存 通用输入 4

指令 RL 可以返回指定轴锁存的位置值。锁存功能被触发后即被关闭。查询 AL 指令返回值为 1 时，说明锁存空能处于设置好的状态。CN 指令可以设置触发锁存的输入信号的变化极性。

参数：AL nnnn　或　AL n, n, n, n

n 为 A、B、C、D，由通用输入信号触发对编码器信号位置的锁存
n 为 SA、SB、SC、SD，由通用输入信号触发对输出脉冲位置的锁存
n 为 TA、TB、TC、TD，由编码器通用输入信号触发对编码器信号位置的锁存

应用：

运动中	是
程序中	是
指令行	是

默认值：

默认值	0
预设格式	1.0

操作数应用：

_Alx	表示锁存功能的状态。0 未设置或已经完成；1 等待被触发。

相关指令：

RL	查询位置锁存值

举例：

```
#A                  程序标号
ALB                 设置 B 轴锁存
JG, 50000           设置点动速度
BGB                 开始运动
#LOOP               用循环等待锁存被触发
JP#LOOP, _ALB=1
RLB                 发送锁存的值
EN                  程序结束
```

AM

功能：等待运动结束（After Move）

说明：

AM 指令是事件触发指令。这个指令会暂时停止程序的执行，直到指定的轴运动结束。AM 指令可以指定一个或多个轴。

AM 指令是以运动规划是否完成作为判断运动结束的标准，实际输出脉冲会比运动规划滞后几个工作周期。

参数：AM xxxxx

x 为 A、B、C、D、S、N 或这些轴的任意组合，省略所有参数则表示同时使用所有参数。

应用：

运动中	是
程序中	是
指令行	否

默认值：

默认值	0
预设格式	1.0

相关指令：

BG	运动结束时，操作数_BGx 为 0
MC	等待运动完成

举例：

```
#MOVE
PR5000, 5000, 5000, 5000    相对位移
BGA                         A 轴开始
AMA                         等待 A 轴运动结束
BGB                         B 轴开始
AMB                         等待 B 轴运动结束
BGC                         C 轴开始
AMC                         等待 C 轴运动结束
BGD                         D 轴开始
AMD                         等待 D 轴运动结束
EN
```

提示：在多个运动流程的控制中，AM 是一个非常重要的指令。例如，如果 A 轴相对位移中（PR），不能进行绝对位移（PAA;BGA）。必须等到前一个运动结束才可以。使用 AMA 指令可以暂停程序执行，等到第一个运动完成以后，再执行后面的指令。

AP

功能：等待绝对位置到达（After Absolute Position）

说明：

AP 指令是事件触发指令。这个指令会暂停程序的执行，直到下面的条件之一出现：

1、电机发送脉冲的位置到达指定的值。

2、电机运动结束

3、电机正在向远离指定位置的方向运动。

AP 指令只能针对正在运动的轴使用，而且每次只能指定一个轴的位置。

参数：AP n, n, n, n　　或　APx=n

n 为-2147483648 至 2147483647 之间的带符号整数。

应用：

运动中	是
程序中	是
指令行	否

默认值：

默认值	—
预设格式	—

相关指令：

AR	等待相对位置
MF	等待正向运动

举例：

```
#TEST
DP0                     定义当前位置为 0
JG 1000                 点动
BGA
AP2000                  等待位置到达 2000
V1=_TDA
MG"Position is", V1     输出信息
ST                      停止
EN                      程序结束
```

提示：AP 指令在时间上的精度为控制器工作周期的两倍，因此运动速度乘以控制器工作周期的 2 倍，就是指令的最大滞后距离。

AR

功能：等待相对距离完成（After Relative Distance）

说明：

AR 为事件触发指令。指令暂停程序的运行，直到下列条件之一被满足：

1、电机运动的距离超过指定的数值。

2、电机运动结束。

3、如果在点动（JG）模式或位置跟踪（PT）模式下，电机在 AR 指令后改变了运动方向。

用于判断电机运动距离的起点，为上一个 AR 或 AD 指令指定的位置，或者是运动的起点。对于点动（JG）模式或位置跟踪（PT）模式，则是最后一次改变方向的位置。AR 指令每次只能指定一个轴的距离。

参数：AP n, n, n, n　　或　APx=n

n 为 0 至 2147483647 之间的整数。

应用：

运动中	是
程序中	是
指令行	否

默认值：

默认值	—
预设格式	—

相关指令：

AV	等待向量位置到达
AP	等待绝对位置到达

举例：

```
#A
DP0, 0, 0, 0
JG50000, , , 70000          指定运动速度
BG AD                       开始运动
#B
AR25000                     等待 A 轴移动距离超过 25000
MG “Passed ”, _TDA          输出信息
JP#B                        跳转到标号#B
EN
```

提示：AR 指令用于指定上一个 AR 或 AD 指令后的增量距离。在一个运动中需要在不同位置多次触发的情况下，AR 指令非常适用。

AS

功能：等待速度到达（At Speed）

说明：

AS 指令为事件触发指令。AS 指令会暂停程序的运行，直到指定轴的规划运动速度达到设定的数值。AS 指令既可以在加速过成中也可在减速过程中使用。如果速度没有达到指定速度，那么在速度开始远离指定速度时，程序也会继续执行。

AS 指令一次只能指定一个轴。

参数：ASx

x 为 A、B、C、D 或 S

应用：

运动中	是
程序中	是
指令行	否

默认值：

默认值	—
预设格式	—

举例：

#SPEED	程序标号
PR100000	设置 A 轴相对移动距离
SP10000	设置 A 轴速度
BGA	A 轴开始运动
ASA	等待 A 轴速度达到设置速度
MG “At Speed”	发送信息
EN	程序结束

警告：

AS 指令只能应用于直线加/减速过程，对于 S 曲线加减速（IT 不等于 1），AS 指令会出错。

@ASIN[n]

功能：反正弦（Arc sine）

说明：

得到给定数值的反正弦角度数值。

参数：@ASIN[n]

n 为带符号数，范围-1~1。如 n 大于 1，则等效于 1，如 n 小于-1，则等效于-1。

应用：

运动中	是
程序中	是
指令行	是

默认值：

默认值	—
预设格式	—

相关指令：

@COS[n]	余弦
@ACOS[n]	反余弦
@SIN[n]	正弦
@ATAN[n]	反正切
@TAN[n]	正切

举例：

```
:MG @ASIN[-1]
 -90.0000
:MG @ASIN[0]
 0.0000
:MG @ASIN[1]
 90.0000
```

AT

功能：等待指定时刻（At Time）

说明：

AT 为事件触发指令。该指令暂停程序的运行，直到指定的时刻。时刻由特定的起点开始计算。AT0 定义计时的起点。ATn 表示从计时起点后 n 毫秒。AT-n 表示计时起点后 n 毫秒，并将该时刻作为新的计时起点。

参数：ATn

n 为带符号的偶数，在 0 至 2 亿之间。

n=0　定义当前时刻为计时起点。

n>0　等待到计时起点后 n 毫秒。

n<0　等待到计时起点后 n 毫秒，而后将该时刻作为新的计时起点。

应用：

运动中	是
程序中	是
指令行	否

默认值：

默认值	0
预设格式	—

举例：

以下指令顺序执行

AT 0	定义计时起点
AT 50	等待到计时起点之后 50ms
AT 100	等待到计时起点之后 100ms
AT –150	等待到计时起点之后 150ms，然后将该时刻定义为新计时起点
AT 80	等待到新计时起点后 80ms（原计时起点后 230ms）

@ATAN

功能：反正切（Arc tangent）

说明：

得到给定数据的反正切角度值。

参数：@ATAN[n]

n 为带符号数，范围-2147483648~2147483647。

应用：

运动中	是
程序中	是
指令行	是

默认值：

默认值	—
预设格式	—

相关指令：

@ACOS[n]	反余弦
@COS[n]	余弦
@ASIN[n]	反正弦
@SIN[n]	正弦
@TAN[n]	正切

举例：

```
:MG @ATAN[-10]
 -84.2894
:MG @ATAN[0]
 0.0000
:MG @ATAN[10]
 84.2894
```

#AUTO

功能：开机（复位）处理子程序

说明：

如果控制器中保存的用户程序中有#AUTO 这个标号，那么控制器开机（复位）后，就会自动从这个标号处开始执行程序。当整个控制系统中没有 PC 或其它上位控制器时，这个功能是非常有用的。

应用：

运动中	是
程序中	是
指令行	否

相关指令：

BP	保存程序
EN	程序结束

举例：

#AUTO	程序标号
PRA=1000	设置 A 轴相对移动距离
BGA	A 轴开始运动
AMA	等待 A 轴运动结束
EN	程序结束

注意：

使用 EN 指令结束程序。

#AUTOERR

功能：内存校验错误子程序

说明：

如果控制器中保存的用户程序中有#AUTOERR 这个标号，那么控制器在开机（复位）过程中出现内存检验错误时，就会自动从这个标号处开始执行程序。错误的类型可以由操作数_RS 得到。

应用：

运动中	是
程序中	是
指令行	否

相关指令：

_RS	检验错误代码
EN	程序结束

举例：

#AUROERR	程序标号
WT500	延时 0.5 秒
MG “CHECKSUM ERR “, _RS	发送错误信息
EN	程序结束

注意：

使用 EN 指令类结束程序流程

AV

功能：等待向量距离（After Vector Distance）

说明：

AV 为事件触发指令。AV 会暂停程序的运行，直到插补运动经过的向量路径达到指定距离。计算距离的起点是插补运动路径的起点，或上一个 AV 指令的终点。

参数：AV s

s 为无符号整数，范围是 0 至 2147483647。s 表示S 轴向量路径的长度。

应用：

运动中	是
程序中	是
指令行	否

默认值：

默认值	—
预设格式	—

操作数：

_AVS	表示 S 轴路径起点开始经过的总长度。

举例：

#MOVE	程序标号
DP0, 0	设置当前位置为 0
LMAB	设置 A\B 两轴进行直线插补
LI1000, 2000	设置插补向量路径
LI2000, 3000	设置插补向量路径
LE	结束直线插补向量路径
BG S	开始插补运动
AV 500	等待向量路径运动长度达到 500
MG”PATH>500”	输出信息
TDAB	输出 A、B 两轴发送脉冲数
EN	程序结束

AX

功能：急停输入设置

说明：

通用输入 8 可以选择只作为一个通用的数字输入还是用兼作控制器的急停信号输入。AX 指令就是用于设置通用输入 8 的作用。

参数：AX n

n=0	通用输入 8 只作为通用数字输入。
n=1	通用输入 8 兼作控制器的急停输入。
n=?	返回当前 AX 的设置值。

应用：

运动中	是
程序中	是
指令行	是

默认值：

默认值	1
预设格式	—

BG

功能：开始运动（Begin）

说明：

BG 指令使指定轴开始运动。

参数：BGxxxxxx

n 为 A、B、C、D、S 或 N 以及它们的任意组合，省略所有参数则表示同时使用 A、B、C、D。

应用：

运动中	否
程序中	是
指令行	是

默认值：

默认值	0
预设格式	—

操作数：

_BGx	为 0 表示 x 轴处于静止状态；否则为 1。

相关指令：

AM	等待运动结束
ST	停止运动

举例：

PR 2000, 3000, , 5000	设置 A、B、D 轴相对运动距离
BG ABD	A、B、D 轴开始运动
HM	设置所有轴原点返回动作
BGA	A 轴开始原点返回动作
JG1000, 4000	设置 A、B 轴点动速度
BGB	B 轴开始运动
BSTATE=_BGB	将 B 轴运动状态赋给变量 BSTATE
VMAB	设置 A、B 轴进行二维插补
VP 4000, -1000	设置二维插补用动向量
VE	结束插补运动设置
PR, , 8000, 5000	设置 C、D 轴相对运动距离
BGSCD	插补运动以及 C、D 两轴的相对运动开始
MG_BGS	输出插补运动状态信息

BK

功能：断点（Breakpoint）

说明：

BK 指令用于程序调试。控制器在指定线程运行程序到指定行号时，会暂停程序运行（被指定的行未运行）。其它线程不受影响。任何时刻都只能设置一个断点。只有在设置的断点起作用后，才能再设置新的断点，或者用 BK 指令继续执行程序。SL 指令可以使程序自中断处单步执行。断点可以在程序开始运行前或运行中设置。

参数：BK n, m

n 为 0 至 449 的整数，表示中断的行号，程序运行到这一行前暂停。

m 为 0 至 3 的整数，表示运行程序的线程号。

省略参数 n 则程序恢复正常运行。

应用：

运动中	是
程序中	否
指令行	是

默认值：

默认值	m=0
预设格式	—

操作数应用：

_BK 的值可以表示出是否设置了断点以及断点是否已经起作用。

= -2147483648	未设置断点
= 程序行号	断点已经起作用
= -程序行号	设置了断点，但还未起作用

相关指令：

SL	单步运行
TR	跟踪

举例：

BK3	0 线程运行的程序在行号 3 处暂停
BK5	0 线程运行的程序继续运行至行号 5 处暂停
SL	执行下一行
SL3	连续执行下三行
BK	恢复正常运行

BL

功能：反向软件极限（Backward Software Limit）

说明：

BL 指令用于设置反向软件极限位置。如果运动超过了这个极限位置，该轴会减速停止。控制器不允许超过这个极限位置的反向运动。

如果程序中含有标号#LIMSWI，在运动超过软件极限位置时，控制器会自动从标号#LIMSWI 开始执行程序。

参数：BL n, n, n, n　　或　BLx=n

n 为带符号整数，范围-2147483648~2147483647。反向极限位置为 n-1。

n=-2147483648，关闭软件极限

n=? 返回当前的软件极限设置值

应用：

运动中	是
程序中	是
指令行	是

默认值：

默认值	-2147483648
默认格式	位置格式

操作数应用：

_BLx	该轴反向软件极限的设置值。

相关指令：

FL	正向极限
PF	位置格式

举例：

#TEST	程序标号
AC 1000000	设置 A 轴加速度
DC 1000000	设置 A 轴减速度
BL –15000	设置 A 轴反向极限位置
JG –5000	反向点动
BGA	开始 A 轴运动
AMA	等待 A 轴运动结束（超过极限位置）
TDA	输出当前位置
EN	程序结束

提示：

GALIL 控制器也提供硬件位置极限。无论软极限或硬极限，都会引发#LIMSWI 的运行。

BN

功能：保存参数（Burn）

说明：

BN 将下列参数保存至控制器内部的闪存中。这个指令需要 1 秒左右的运行时间，执行时不可被中断。

BN 指令保存的参数：

AC	AX	BL	CB	CE	CN	CW
DC	DH	EO	FL	GA	GM	GR
IA	IK	IT	KS	LC	LD	^L^K
LZ	MO	MT	OE	OP	PF	PW
SB	SM	SP	TM	TR	VA	VD
VF	VS					

参数： 无

应用：

运动中	是
程序中	是
指令行	是

默认值：

默认值	—
预设格式	—

操作数应用:

_BN	表示控制器的序号。**相**

关指令：

BP	保存程序
BV	保存变量

举例：

AC 200000	设置 A 轴加速度
DC 150000	设置 A 轴加速度
SP 10000	设置 A 轴加速度
MT 2.5	设置 A 轴电机反向
MO	关闭所有电机
BN	保存设置

提示：

执行此指令时，通讯程序可能会出现通讯超时错误。

BP

功能：保存程序（Burn Program）

说明：

BP 指令将控制器中下载的用户程序保存至控制器内部的闪存中。这个指令需要 2 秒左右的运行时间，执行时不可被中断。

参数： 无

应用：

运动中	否
程序中	是
指令行	是

默认值：

默认值	—
预设格式	—

相关指令：

BN	保存参数
BV	保存变量

提示：

执行此指令时，通讯程序可能会出现通讯超时错误。

BV

功能：保存变量及数组（Burn Variables and Arrays）

说明：

BV 指令将用户变量及数组保存至控制器内部的闪存中。这个指令需要 2 秒左右的运行时间，执行时不可被中断。

参数： 无

应用：

	运动中	是
	程序中	是
	指令行	是

默认值：

默认值	—
预设格式	—

操作数应用:

_BV	表示控制器支持的轴数。

相关指令：

BN	保存程序
BP	保存变量

提示：

1、此指令将电子凸轮表同样保存至控制器内部闪存中。

2、执行此指令时，通讯程序可能会出现通讯超时错误。

CB

功能：清除输出位（Clear Bit）

说明：

CB 指令使指定的通用输出截止。

参数：CB n

n 为 1 至 4 之间的整数。

应用：

运动中	是
程序中	是
指令行	是

默认值：

默认值	—
预设格式	—

相关指令：

SB	设置输出位
OB	输出位
OP	输出

举例：

CB3	清除通用输出 3
CB1	清除通用输出 1

CD

功能：轮廓资料（Contour Data）

说明：

CD 指令指定轮廓运动各轴的位置增量。这个指令只应用于轮廓模式（样条插值）。位置增量对应时间（2~256 控制器工作周期）由指令 DT 给出，也可以在 CD 指令的等号后给出。

参数：CD n, n, n, n=m　　或　CDx=n

n 为+/-32762 之间的整数，表示位置增量。

m（可选）为 0 至 8 之间的整数，时间间隔为 2^m 个工作周期。

n=m=0 中断轮廓模式

n=0 m=-1 暂停轮廓模式

注意 1：指令 CD0, 0…=0 应跟随在最后一个轮廓数据的 CD 指令后。CD0, 0…=0 的作用类似 VE 和 LE。控制器执行到 CD0, 0…=0 指令，就会结束轮廓模式。

注意 2：指令 CD0=0 会被控制器理解成为变量 CD0 赋值 0，当使用这个指令结束轮廓模式时，要在 CD 后加一个空格，CD 0=0 不会被控制器理解错误。

应用：

运动中	是
程序中	是
指令行	是

默认值：

默认值	—
预设格式	—

相关指令：

CM	轮廓模式
DT	时间增量

举例：

```
#Cont0                      程序标号
CM ABCD                     设置轮廓模式
DT 4                        轮廓时间增量为 16 个控制器工作周期
CD 200, 350, -150, 500      设置 4 轴的位置增量数据
CD 100, 200, 300, 400       设置 4 轴的位置增量数据
CD 0, 0, 0, 0=0             位置数据结束
#Wait;JP#Wait, _CM<>32      等待运动结束（轮廓数据缓冲区全空）
EN                          程序结束

#Cont1                      程序标号
CM ABC                      设置轮廓模式
DT 8                        轮廓时间增量为 256 个控制器工作周期
CD 100, 100, 100            设置 3 轴的位置增量数据
CD 100, 100, 100            设置 3 轴的位置增量数据
CD 0, 0, 0=-1               暂停轮廓数据，设置 DT 指令恢复
CD 100, 100, 100            设置 3 轴的位置增量数据
CD 100, 100, 100            设置 3 轴的位置增量数据
```

```
CD 0, 0, 0=0                位置数据结束
#Wait;JP#Wait, _CM<>32      等待运动结束（轮廓数据缓冲区全空）
EN                          程序结束
```

（CD 0, 0, 0=-1 指令会使运动暂停，要继续完成运动并结束程序，需要在暂停后输入 DT8 指令。）

CE

功能：设置编码器（Configure Encoder）

说明：

CE 指令用设置接收的编码器信号类型和计数方向。控制器可以接收正交方波的信号，也可以接收脉冲/方向的信号。如编码器信号类型设置错误，控制器就不能正确处理接收到的编码器信号。

设置为脉冲/方向类型后，脉冲信号连接到 A+、A-；方向信号连到 B+、B-。

参数：CEn, n, n, n　　或　CEx=n

n 为 0~15 的整数。虽然 n 为 4bit 数，但只有低两位在设置时有效，高两位无效。

n=0　正交方波信号，A 相位超前时计数方向为正

n=1　脉冲/方向信号，B+为高电平、B-为低电平时计数方向为正。

n=2　正交方波信号，B 相位超前时计数方向为正

n=3　脉冲正交方波信号，B-为高电平、B+为低电平时计数方向为正。

n=?　返回当前的设置值，返回值的低两位对应编码器信号设置类型；高两位为 1 或 3（4 或 12），与 MT 指令设置的脉冲输出格式相关。

应用：

运动中	是
程序中	是
指令行	是

默认值：

默认值	4
预设格式	2.0

操作数应用：

_CEx	代表该轴编码器类型设置值。

相关指令：

MT	输出脉冲信号格式

举例：

CE 0, 3, 6, 2	设置编码器类型
CE?, ?, ?, ?	查询编码器类型设置值
4, 7, 6, 6	
MT2, 2.5, -2, -2.5	设置脉冲输出格式
CE?, ?, ?, ?	查询编码器类型设置值
4, 15, 6, 14	*由于脉冲输出格式的变化导致查询结果有变化*
V=_CEB	将 B 轴编码器的设置值赋给变数 V
V=	发送变数 V 的值
15	

注意：对于脉冲/方向信号，如果在脉冲信号持续过程中方向信号有变化，可能导致控制器对信号计数错误。

CF

功能：设置（Configure）

说明：

CF 指令设置控制器默认从哪个通讯端口发送主动信息。控制器出厂时预设从串口发送主动信息。使用者可以设置预设通讯端口为串口或是网口 4 个句柄中的一个。如果此设置错误，则上位控制器会接收不到 DMC-B140-M 控制器发出的主动信息。

参数：CF n

n 为 A、B、C、D、S 或 I。A~D 对应以太网的句柄 1~4；S 代表串口；I 表示使用接收到 CF 指令的通讯端口。无论使用者的上位控制器使用何种方式与 GALIL 运动控制器通讯，只要从上位控制器向 GALIL 运动控制器发送指令 CFI，都可以保证上位控制器可以收到 GALIL 运动控制器从预设端口发送的主动信息（MG 指令可以强制信息指定端口发送，而不通过默认埠）。

应用：

运动中	是
程序中	是
指令行	是

默认值：

默认值	S
预设格式	—

操作数应用：

_CF	表示当前设置字符 ASCII 代码值。

相关指令：

CW	设置主动信息的最高位
WH	查询当前句柄
TH	查询句柄

CI

功能：设置通讯中断（Configure Communication Interrupt）

说明：

CI 指令设置接收到什么信息时触发串口通讯中断。如用户程序中含有#COMINT 标号，当触发了串口通讯中断时，控制器会在 0 线程自动从#COMINT 处开始执行程序。其它线程不受影响。接收到的信息可以通过操作数 P1CH、P1NM、P1ST 得到。

如用户采用中断的方式处理串口通讯，控制器就不在主动处理从串口接受到的信息，也就是和说，控制器对从串口发来的指令不会主动识别、执行。

参数：CI n, m

n=0	任何条件下不触发中断
n=1	收到回车字符（ASCII 代码 13）触发中断
n=2	收到任何字符触发中断
n=-1	清空接收数据缓冲区
m=1	允许中断

应用：

运动中	是
程序中	是
指令行	是

默认值：

默认值	n=0，m=0
预设格式	—

相关指令：

IN	输入信息
MG	输出信息

举例：

CI 1, 1	从串口收到回车字符（ASCII 代码 13）触发中断
CI 2, 1	从串口收到一个字符触发中断

CM

功能：轮廓模式（Contour Mode）

说明：

CM（Contour Mode）指令用来初始化轮廓模式。这个模式支持任意轴任意运动轨迹的生成。CD 指令提供位置增量，DT 指令提供时间间隔。

CM?用于检查有效的轮廓片段数量。返回值为 1 表示轮廓缓冲区已满。返回值为 32 表示缓冲区是完全空的。

参数：CM xxxx

n 为 A、B、C、D 或几轴的组合

n=? 返回缓冲区状态

应用：

运动中	是
程序中	是
指令行	是

默认值：

默认值	0
预设格式	2.0

操作数应用：

_CM	缓冲区状态

相关指令：

CD	轮廓数据
DT	时间增量

举例：

MG _CM	输出缓冲区状态
CM?	输出缓冲区状态
CM AC	设置 A、C 轮廓模式运动

#CMDERR

功能：指令错误处理子程序

说明：

如果控制器中保存的用户程序中有#CMDERR 这个标号，那么控制器执行程序的中出现指令错误，就会自动在 0 线程从这个标号处开始执行程序。原 0 线程运行的程序暂停，在子程序完成后恢复，其它正常运行的线程不受影响。

应用：

运动中	是
程序中	是
指令行	否

相关指令：

TC	查询错误代码
_ED	上一次出错的程序行号
EN	程序结束

举例：

#BEGIN	主程序标号
IN”ENTER SPEED”, Speed	提示输入速度
JG Speed	
BGA	开始运动
EN	程序结束
#CMDERR	指令错误处理子程序
JP#DONE, _ED<>2	检查错误是否发生在 2 行
JP#DONE, _TC<>6	检查错误是否为参数超范围
MG”Speed Too High”	输出信息
MG”Try Again”	
ZS1	调整堆栈
JP#BEGIN	返回主程序重新执行
#DONE	如为其它错误则结束程序
ZS0	清空堆栈
EN1	程序结束

注意：只有在程序运行时，#CMDERR 才是有效的。使用 EN 结束程序。

CN

功能：设置输入信号（Configure）

说明：

CN 指令用于设置极限信号输入、原点开关输入、锁存信号的极性。也可以设置将通用输入作为每轴的单轴急停信号。

参数：CN m, n, o, p, q

m、n、o 为 1 或-1　　p、q 为 0 或 1

m=	1	极限信号截止状态有效
	-1	极限信号导通状态有效
n=	1	寻找原点开关时，原点信号为截止状态则向前寻找（参考 HM、FE 指令）
	-1	寻找原点开关时，原点信号为导通状态则向前寻找（参考 HM、FE 指令）
o=	1	设置输入信号截止时触发锁存
	-1	设置输入信号导通时触发锁存
p=	1	设置通用的 5、6、7、8 为 A、B、C、D 轴的专用单轴急停信号。当信号输入时，会自动执行#POSERR 子程序
	0	设置各轴无专用单轴急停信号
q=	1	急停输入信号不中止程序运行
	0	急停输入信号中止程序运行

应用：

运动中	是
程序中	是
指令行	是

默认值：

默认值	-1, -1, -1, 0, 0
预设格式	2.0

操作数应用：

_CN0	极限信号设置
_CN1	原点开关信号设置
_CN2	位置锁存信号设置
_CN3	单轴急停信号设置
_CN4	急停信号是否中断程序设置

相关指令：

AL	设置位置锁存

举例：

CN 1, 1	设置位置极限信号与原点开关信号均为截止有效
CN , , −1	设置锁存信号在导通时被触发

@COM[n]

功能：按位求补（Complement）

说明：

按位计算给定数据的补码（NOT）。

参数：@COM[n]

n 为-2147483648 至 2147483647 之间的带符号整数。这个整数作为 32bit 数据处理。

应用：

运动中	是
程序中	是
指令行	是

默认值：

默认值	—
预设格式	—

相关指令：

& |	逻辑与（AND）、或（OR）

举例：

:MG{$8.0} @COM[0]	以 8 位十六进制格式输出 0 的补码
$FFFFFFFF	
: MG{$8.0} @COM[$FFFFFFFF]	以 8 位十六进制格式输出$FFFFFFFF 的补码
$00000000	

#COMINT

功能：串口通讯中断处理子程序

说明：

如果控制器中保存的用户程序中有#COMINT 这个标号，那么控制器执行程序的中出现串口通讯中断，就会自动在 0 线程从这个标号处开始执行程序。原 0 线程运行的程序暂停，在子程序完成后恢复，其它正常运行的线程不受影响。

串口通讯中断产生的条件由指令 CI 设置。

应用：

运动中	是
程序中	是
指令行	否

相关指令：

CI	设置串口通讯中断
P1CD	串口接收到的字符代码
PICH	串口接收到的字符
P1NM	串口接收到的数位
P1ST	串口接受到的字符串
EN	程序结束

举例：

```
#A                  程序标号
CI2, 1              设置当串口接收到一个字符时产生通讯中断
#LOOP               循环
MG”LOOP”
WT1000
JP#LOOP
#COMINT
MG”COMINT”
EN1, 1
```

注意：只有在有其它应用程序执行时，中断程序才会运行。要使用 EN 指令结束子程序。

@COS[n]

功能：余弦（Cosine）

说明：

返回给定角度值的余弦。

参数：@COS[n]

n 为带符号的角度值，-32768~32767，小数部分分辨率为 16bit。

应用：

运动中	是
程序中	是
指令行	是

默认值：

默认值	—
预设格式	—

相关指令：

@ACOS[n]	反余弦
@ASIN[n]	反正弦
@SIN[n]	正弦
@ATAN[n]	反正切
@TAN[n]	正切

举例：

```
:MG @COS[0]
 1.0000
:MG @COS[90]
 0.0000
:MG @COS[180]
 -1.0000
:MG @COS[270]
 0.0000
:MG @COS[360]
 1.0000
```

CR

功能：圆弧（Circle）

说明：

CR 指令用于平面（二维）向量插补模式，向插补路径缓冲区中增加一段圆弧路径。CR 以半径 r、起始角度θ、旋转角度Δθ定义一段二维圆弧轨迹。θ、Δθ以角度为单位。在笛卡尔坐标内，以 X 轴正方向为 0 度位置，以逆时针方向为正，以顺时针方向为负在全部 VP、CR 指令之后，用 VE 结束二维插补运动描述。BG 指令用于开始运动。r、θ、Δθ这三个参数均不得省略。半径的单位是输出脉冲，角度的单位是度（一周为360度 。

参数：CR r, θ, Δθ<n>o

r 为无符号数，范围 10~2670227，表示圆弧半径。

θ为带符号数，范围0~+/–32760，以角度为单位，表示圆弧起点的角度。

Δθ为带符号数，范围+/–0.0001~+/–32760，以角度为单位，表示圆弧经过的角度。

n 为无符号偶数，2~3000000，表示运动到该段路径时的向量速度。

n=-1　表示使用 VV 指令设置的速度。

o 为无符号偶数，2~3000000，表示运动到该段路径末尾时的向量速度。

注意：一定要保证 r*Δθ 在+/-450000000 之间。

应用：

运动中	是
程序中	是
指令行	是

默认值：

默认值	—
预设格式	—

相关指令：

VP	向量位置
VS	向量速度
VD	向量加速度
VA	向量减速度
VM	平面（二维）向量模式
VE	向量路径结束
VV	向量速度变量

举例：

VM AB	定义 A、B 平面内的向量运动
VS 10000	指定向量速度
CR 1000, 0, 360	以 0 度角开始逆时针方向半径为 1000 的圆形路径
CR 1000, 0, 360<40000	以 40000 的速度重复一次
VE	结束向量路径
BGS	开始向量运动

CS

功能：清除插补路径缓冲区（Clear Sequence）

说明：

CS 指令用于清除插补缓冲区中未执行的路径（VP、CR、LI）。在正常情况下，每执行一段路径（一个 VP、CR 或 LI 指令），该路径定义描述就会自动从缓冲区消除，不需要使用 CS 指令。在插补路径描述有错，不能正常执行完成的情况下，才需要使用 CS 指令。

参数：CSS

S 为插补向量轴的名称。

应用：

运动中	否
程序中	是
指令行	是

默认值：

默认值	—
预设格式	—

操作数应用：

_CSS	表示插补路径中的段编号，可用于两种插补模式（LM、VM）。

相关指令：

CR	圆弧插补路径
LI	直线插补路径
LM	直线插补模式
VM	平面（二维）向量模式
VP	向量位置

举例：

#CLEAR	程序标号
VM AB	向量模式
VP 1000, 2000	向量位置
VP 4000, 8000	向量位置
CSS	清除上面两段向量路径
VP 1000, 5000	新的向量位置
VP 8000, 9000	新的向量位置
CSS	清除向量路径
EN	程序结束

CW

功能：版本信息与通讯设置

说明：

CW 指令有两个用途。参数 n 为 0 时，CW 指令可以返回控制器的版本信息。参数 n 不为 0 时，这个指令可以设置控制器串口发送的主动信息 ASCII 字符的最高位。CW 指令的另一个参数用于设置控制器信息发送失败时的处理方式。

每个 CW 指令只能使用一个参数。如果要同时设置主动信息的最高位与通讯失败的处理方式，使用指令 CW2;CW, 1，而不要使用指令 CW2, 1。

参数：CWn, m

n=0 使控制器返回版本信息

n=1 设置主动信息 ASCII 字符最高位为 1

n=2 设置主动信息 ASCII 字符最高位为 0

n=? 使控制器返回版本信息

m 为可选参数

m=0 控制器硬件握手失败时暂停内部用户程序

m=1 控制器硬件握手失败时继续执行内部用户程序，放弃需要发送的信息

应用：

运动中	是
程序中	是
指令行	是

默认值：

默认值	2, 0
预设格式	—

操作数应用：

_CW	表示主动信息最高位状态。2，为高位为 0；1，最高位为 1

注意：Galil 的通讯终端软件在运行中可能会自动修改 CW 指令的设置。如果用户使用 Galil 的通讯终端软件过程中执行了 BN 指令，可能会将修改过的 CW 指令设置保存在控制器上，使得控制器以后工作中产生用户不希望的结果。

DA

功能：释放变量与数组的存储空间（Deallocate）

说明：

DA 指令释放用户变量与数组的存储空间。一个 DA 指令中可以指定多个变量与数组。变量与数组之间用逗号分开。参数*会释放所有变量，参数*[0]会释放全部数组。

参数：DA c[0], 变量名

c[0]	定义过的数组名
变量名	定义过的变量名
*	释放全部变量
*[0]	释放全部数组
DA?	返回目前可定义的数组数量

应用：

运动中	是
程序中	是
指令行	是

默认值：

默认值	—
预设格式	—

操作数应用：

_DA	表示当前可定义的数组数量，最大为 6。

相关指令：

DM	定义数组

举例：

DM Cars[400], Sales[50]	定义两个数组
Total=70	定义变量 Total 并赋值 70
DA Cars[0], Sales[0], Total	取消这两个数组和一个变量
DA *[]	取消所有数组
DA *, *[]	取消所有变量与数组

注意：由于 DA 指令涉及到存储空间的重新分配，这个指令的运行时间可能会超过 2ms。

DC

功能：减速度（Deceleration）

说明：

DC 设置电机独立运动中（PA、PR、JG）的线性减速度。除非在点动（JG）模式下，电机独立运动过程中不能对该轴使用 DC 指令设置减速度。

参数：DC n, n, n, n　　或　DCx=n

n 为无符号整数，范围 1024~1073740800，单位是脉冲每秒平方。输入的数据如不是 1024 的倍数，回自动变换为不大于输入数值的 1024 整数倍数。如输入的数据小于 1024，则自动变换为 1024。

n=?　　返回该轴的减速度设置值。

应用：

运动中	是*
程序中	是
指令行	是

默认值：

默认值	256000
预设格式	10.0

操作数应用：

_DCx	该轴的减速度数值.

相关指令：

AC	加速度
PR	相对位置
PA	绝对位置
SP	速度
JG	点动
SD	位置极限信号有效时的减速度

举例：

PR 10000	设置位置
AC 2000000	设置加速度
DC 1000000	设置减速度
SP 5000	设置速度
BG	开始运动

注意：DC 指令只能在点动（JG）运动中修改，定位运动（PA、PR、IP）中不能修改。

DE

功能：定义编码器计数值

说明：

DE 指令定义当前编码器回馈位置（计数）值。

参数：DE n, n, n, n　　或　DEx=n

n 为带符号的整数，范围-2147483648~2147483647。

n=?　返回该轴输出脉冲位置（输出脉冲数）。

应用：

运动中	是
程序中	是
指令行	是

默认值：

默认值	0, 0, 0, 0
默认格式	位置格式

操作数应用：

_DEx	指定轴输出脉冲位置（输出脉冲数）。

相关指令：

DP	定义当前位置
TD	查询输出脉冲数

举例：

DE 0, 100, 200, 40	定义四轴当前编码器回馈位置
DE ?, ?, ?, ?	查询当前四轴输出脉冲数
PulseA=_DEA	将当前 A 轴输出脉冲数赋给变数

DH

功能：动态地址（DHCP）

说明：

DH 指令用于设置控制器的使用 DHCP 还是 BOOT-P 的 IP 地址分配。

参数：DH n

n=0	取消 DHCP，采用 BOOT-P
n=1	取消 BOOT-P，采用 DHCP
n=?	返回当前设置的状态

应用：

运动中	是
程序中	是
指令行	是

默认值：

默认值	1
预设格式	—

相关指令：

IA	IP 地址

举例：

DH1	使用 DHCP，IA 指令无效
DH0	控制器使用静态 IP 地址，地址可保存在控制器上

DL

功能：下载程序（DownLoad）

说明：

DL 指令使控制器准备接收一个档，并将该文件作为用户程序加载控制器 RAM 中。这个档必须是以<control>-Z、<control>-Q、<control>-D 或是 \ 结束。程序中指令之间不要插入空格。

如果没有指定参数，下载的档会清除控制器 RAM 中的全部程序，新的程序从 0 行开始。如果档中的行数或每行中的字符数超出控制器允许的范围，控制器会返回一个问号—
— ？。

如果下载的程序只要覆盖远 RAM 中程序某标号以后的内容，可以把标号作为 DL 指令的参数。单纯以 # 作为参数，则原 RAM 中的程序全部保留，下载的部分保存在原程序之后。

参数：DL n

n=#标号 在指定标号后一行开始下载新的程序

n=# 在原有程序之后下载新的程序

省略参数，则下载新程序时清除 RAM 中原有程序。

应用：

运动中	是
程序中	否
指令行	是

默认值：

默认值	—
预设格式	—

操作数应用：

_DL	表示目前程序中还可以增加多少标号，最多 62。

相关指令：

UL	上载

举例：

DL;	开始下载
#A;PR 4000;BGA	程序内容
AMA;MG DONE	程序内容
EN	程序内容
<control>-Z	下载结束

DM

功能：定义数组（Dimension）

说明：

DM 指令定义一个指定名称和元素数量的一维数组。数组中第一个元素下标为 0。

参数：DM c[n]

c 为数组名称，最长 8 个字符，首字符必须为字母。

n 为数组中元素数，第一个元素下标为 0，最后一个元素下标为 n-1。

应用：

运动中	是
程序中	是
指令行	是

默认值：

默认值	—
预设格式	—

操作数应用：

_DM	控制器中空闲的数组元素存储空间，最大为 800。

相关指令：

DA	释放变量与数组的存储空间

举例：

DM Pets[5], Dogs[2], Cats[3]	定义三个数组
DM Tests[800]	定义一个数组

DP

功能：定义位置（Define Position）

说明：

DP 指令设置控制器当前的规划位置，同时也设置了当前的输出脉冲位置。使用 DP0 以后，RP 与 TD 回馈的 A 轴数值均为 0。但是 DP 指令不会影响编码器回馈位置（计数值）。

参 数 ：DP n, n, n, n 或 DPx=n

n 为带符号的整数，范围-2147483648~2147483647。

n=? 返回该轴编码器回馈位置（计数值）。

应用：

运动中	否
程序中	是
指令行	是

默认值：

默认值	—
预设格式	—

操作数应用：

_DPx	指定轴编码器回馈位置（计数值）。

相关指令：

DE	定义编码器计数值
TD	查询输出脉冲数
TP	查询编码器回馈位置
PF	位置格式

举例：

DP 0, 100, 200, 400	定义四轴当前位置
DP , -50000	定义 B 轴当前位置

DR

功能：数据记录的更新速率（Data Record Update Rate）

说明：

控制器定时自动产生 QR 指令要求的数据记录，并通过使用 UDP 协议的以太网句柄向外发送。

参数：DR n, m

n 为更新间隔，范围 2~30000，单位为控制器工作周期，n=0 表示关闭此功能。

m 为用于发送的句柄，0 表示句柄 A；1 表示句柄 B；……；3 表示句柄 D。该句柄必须使用 UDP 协议。

应用：

运动中	是
程序中	是
指令行	是

默认值：

默认值	0
预设格式	—

操作数应用：

_DR	数据记录的更新间隔。

相关指令：

QZ	数据记录格式
QR	数据记录

举例：

```
DR 1000, 0
```

注意：数据记录为一组 2 进制格式数据。

DT

功能：时间增量（Delta Time）

说明：

DT 指令用于设置轮廓模式下每个位置增量对应的时间间隔。DT 指令设置一次后一直有效，直到新的 DT（或 CD 指令）改变时间间隔的设置值。

参数：DT n

n 为 0~8 之间的整数。

n=1~8　表示时间间隔为 2^n 个控制器的工作周期

n=-1　表示暂停轮廓模式运动，此时依然可以用 CD 指令向缓冲区写如数据，但控制器不输出脉冲，内部规划位置也不变化。再次使用任何参数为正数的 DT 指令可以继续轮廓模式的运动。

n=?　返回当前设置值。

应用：

运动中	是
程序中	是
指令行	是

默认值：

默认值	1
预设格式	1.0

操作数应用：

_DT	轮廓模式下时间间隔的设置值。

相关指令：

CM	轮廓模式
CD	轮廓数据

举例：

DT 4	设置时间间隔为 16 个工作周期
DT 7	设置时间间隔为 128 个工作周期
#Cont	程序标号
CM AB	设置轮廓模式
DT –1	暂停运动
CD 100, 200	数据写入缓冲区
CD 400, 200	数据写入缓冲区
CD 200, 100	数据写入缓冲区
CD 300, 50	数据写入缓冲区
AI –1	等待通用输入 1 导通
DT 8	设置时间间隔，恢复运动
CD 0, 0, 0, 0=0	数据结束
#Wait;JP#Wait, _CM<>32	等待运动结束（轮廓数据缓冲区全空）
EN	程序结束

EA

功能：电子凸轮主轴（ECAM Master）

说明：

EA 指令用于指定电子凸轮模式的主轴。可以指定 A、B、C、D 或虚拟轴 N 作为主轴。当指定实际的 A、B、C 或 D 轴为主轴时，从轴是以主轴的编码器回馈位置为基准的。当指定虚拟轴为主轴时，是以控制器的规划位置为基准的。

参数：EA x

x 为 A、B、C、D 或 N。

应用：

运动中	是
程序中	是
指令行	是

默认值：

默认值	—
预设格式	—

相关指令：

EB	电子凸轮模式使能
EC	电子凸轮表索引
EG	电子凸轮开始启动
EM	电子凸轮周期
EP	电子凸轮表定义
EQ	电子凸轮脱离
ET	电子凸轮表数据
EW	电子凸轮加宽数据间隔

举例：

EA B	以 B 轴作为电子凸轮模式的主轴

EB

功能：电子凸轮模式使能（Enable ECAM）

说明：

EB 指令打开或关闭电子凸轮模式。在电子凸轮模式下，在循环周期内指定主轴的开始位置。使用了 EB 指令后，主轴的位置就开始模式化。

参数：EB n

n=0	停止电子凸轮模式
n=1	开始电子凸轮模式
n=?	返回当前设置的值，0 或 1。

应用：

运动中	是
程序中	是
指令行	是

默认值：

默认值	0
预设格式	1.0

操作数应用：

_EB	当前电子凸轮模式的状态， 0 电子凸轮模式无效；1 电子凸轮模式有效。

相关指令：

EA	电子凸轮主轴
EC	电子凸轮表索引
EG	电子凸轮开始启动
EM	电子凸轮周期
EP	电子凸轮表定义
EQ	电子凸轮脱离
ET	电子凸轮表数据
EW	电子凸轮加宽数据间隔

举例：

```
EB 0
EB 1
B=_EB
```

EC

功能：电子凸轮数据索引（ECAM Counter）

说明：

EC 指令设置电子凸轮数据表的索引。这个指令只在输入凸轮数据表时有用。使用了 EC 指令后，输入数据时可以省略数据索引。参见 ET 指令。

参数：EC n

n 为 0~256 的整数

n=? 返回数据表中当前输入数据的索引。

应用：

运动中	是
程序中	是
指令行	是

默认值：

默认值	0
预设格式	1.0

操作数应用：

_EC	数据表中当前输入数据的索引

相关指令：

EA	电子凸轮主轴
EB	电子凸轮模式使能
EG	电子凸轮启动
EM	电子凸轮周期
EP	电子凸轮表定义
EQ	电子凸轮脱离
ET	电子凸轮表数据
EW	电子凸轮加宽数据间隔

举例：

EC 0	设置之后输入的数据从索引 0 开始
ET 200, 400	电子凸轮表的首个数据（索引 0）为 200, 400
ET 400, 800	电子凸轮表的下一个数据（索引 1）为 400, 800

ED

功能：编辑（Edit）

说明：

ED 指令可以在早期的 PC 操作系统上实现对控制器的编辑。

操作数应用：

_ED	最后一次错误所在行的行号
_ED1	出现错误的程序所运行的线程号

EG

功能：电子凸轮启动（ECAM Go）

说明：

EG 指令为电子凸轮中的每个从轴指定一个主轴位置，当主轴到达这个位置时，从轴开始按照数据表跟随主轴运动。如果 EG 指令的参数超出了主轴一个周期内的位置，则从轴立刻开始跟随主轴。

参数：EG n, n, n, n　　或　EGx=n

n 为凸轮模式下主轴的位置，当主轴到达这个位置时，从轴开始跟随。

n=?　　如果该从轴已经启动，返回 1；否则返回 0。

应用：

运动中	是
程序中	是
指令行	是

默认值：

默认值	0
预设格式	1.0

操作数应用：

_EGx	该轴状态。0，该轴未启动；1，该轴已经启动。

相关指令：

EA	电子凸轮主轴
EB	电子凸轮模式使能
EC	电子凸轮表索引
EM	电子凸轮周期
EP	电子凸轮表定义
EQ	电子凸轮脱离
ET	电子凸轮表数据
EW	电子凸轮加宽数据间隔

举例：

EG 700, 1300	主轴位置到达 700 时 A 轴启动，1300 时 B 轴启动
B=_EGB	返回 B 轴是否启动的状态

注意：EG 并不是一个条件触发指令。EG 指令出现在程序中，不会使程序暂停。如果需要在主轴位置到达前暂不执行后面的指令，请使用 MF 或 MR。

ELSE

功能：否则，用于条件分支结构

说明：

ELSE 指令是 IF 条件分支结构的可选部分。ELSE 指令必须出现在 IF 指令之后。当 IF 指令的逻辑表达式判断为假，则执行 ELSE 后的指令。如 IF 指令的逻辑表达式判断为真，控制器会执行 IF 与 ELSE 之间的指令；IF 指令的逻辑表达式判断为假，控制器直接跳过这些指令直到 ELSE 指令。

参数：无

应用：

运动中	是
程序中	是
指令行	否

默认值：

默认值	—
预设格式	—

相关指令：

ENDIF	条件分支结构结束
IF	条件判断

举例：

```
#A                                  程序标号
IF(@IN[1]=0)                        判断通用输入 1 状态
IF(@IN[2]=0)                        判断通用输入 2 状态
MG “INPUT 1 AND 2 ARE ACTIVE”       输出信息
ELSE                                如通用输入 2 不满足条件
MG “ONLY INPUT 1 IS ACTIVE”         输出信息
ENDIF                               结束对通用输入 2 的判断分支
ELSE                                如通用输入 1 不满足条件
MG “INPUT 1 IS NOT ACTIVE”          输出信息
ENDIF                               结束对通用输入 1 的判断分支
EN                                  程序结束
```

EM

功能：电子凸轮周期（ECAM Modulus）

说明：

EM 指令是电子凸轮模式的一部分，用来定义一个运动周期的位置变化。主轴的部分就是一个凸轮周期对应的主轴运动距离。对于从轴，则是指一个周期内的净移动距离。如果在一个周期中，从轴最终回到了开始时的位置，那么这个净移动距离就是 0。如果这个净移动距离是反向的，指令的参数使用绝对值。

参数：EM n, n, n, n　　或　EMx=n

n 为整数，对于主轴，n 的范围 1~8388607；对于从轴，n 的范围 0~2147483647。

应用：

运动中	是
程序中	是
指令行	是

默认值：

默认值	—
预设格式	—

操作数应用：

_EMx	该轴的设置值。

相关指令：

EA	电子凸轮主轴
EB	电子凸轮模式使能
EC	电子凸轮表索引
EG	电子凸轮启动
EP	电子凸轮表定义
EQ	电子凸轮脱离
ET	电子凸轮表数据
EW	电子凸轮加宽数据间隔

举例：

EA C	选择 C 轴为电子凸轮的主轴
EM 0, 3000, 2000	主轴以 2000 为一个周期，一个周期内 A 轴位置净变化为量 0，B 轴位置净变化量为 3000
V=_EMA	返回 A 轴在一个周期内的位置净变化量

EN

功能：结束（END）

说明：

EN 指令用于表示程序或子程序的结束。如果是用 JS 指令调用的子程序，EN 指令结束子程序并将程序流程转到 JS 后的指令。

EN 指令也可用于自动运行子程序#COMINT 和#CMDERR，此时 EN 指令可以有两个参数。第一个参数表示子程序完成后是，恢复 0 线程原运行的程序是，是否恢复事件触发状态，第二个参数决定是否继续支持中断。

参数：EN m, n

m=0	从子程序返回，取消原程序中对触发事件的等待，直接执行下一指令
m=1	从子程序返回，恢复原程序中对触发事件的等待
n=0	从子程序返回，不再响应新的中断，此参数仅用于#COMINT 子程序
n=1	从子程序返回，并会响应新的中断，此参数仅用于#COMINT 子程序

注意：参数默认值为 0。自动运行子程序#LIMSWI、#TCPERR 使用指令 RE 返回，自动运行子程序#ININT 使用指令 RI 返回。

应用：

运动中	是
程序中	是
指令行	否

默认值：

默认值	m=0
预设格式	—

相关指令：

RE	从错误处理子程序返回
RI	从输入中断返回

举例：

#A	程序标号
PR 500	指定运动距离
BGA	开始运动
AMA	等待 A 轴运动结束
EN	程序结束

ENDIF

功能：条件分支结构结束

说明：

ENDIF 指令表示 IF 条件分支结构结束。每执行过一个 IF 指令后，必须执行一个 ENDIF 指令。建议用户在 IF 条件分支结构内部不要使用跳转指令，这样可能会导致 ENDIF 没有被执行。

参数：无

应用：

运动中	是
程序中	是
指令行	否

相关指令：

IF	条件判断
ELSE	否则，用于条件分支结构
JP	跳转
JS	转子程序

举例：

#A	程序标号
IF(@IN[1]=0)	判断通用输入 1 状态
IF(@IN[2]=0)	判断通用输入 2 状态
MG “INPUT 1 AND 2 ARE ACTIVE”	输出信息
ELSE	如通用输入 2 不满足条件
MG “ONLY INPUT 1 IS ACTIVE”	输出信息
ENDIF	结束对通用输入 2 的判断分支
ELSE	如通用输入 1 不满足条件
MG “INPUT 1 IS NOT ACTIVE”	输出信息
ENDIF	结束对通用输入 1 的判断分支
EN	程序结束

EO

功能：响应（ECHO）

说明：

EO 指令用于设置串口通讯中响应功能开启或关闭。如开启响应功能，则控制器对串口接收到的所有通讯信息都会以原内容响应。

EO 指令不能通过以太网接口发送执行。

参数：

EO	n
n=0	关闭响应功能
n=1	开启响应功能

应用：

运动中	是
程序中	是
指令行	是

默认值：

默认值	0
预设格式	1.0

举例：

EO 0	关闭响应功能
EO 1	开启响应功能

提示：Galil 的通讯软件可能会自动调整 EO 的设置。

EP

功能：电子凸轮表定义（ECAM table master interval and phase shift）

说明：

EP 指令用于定义电子凸轮数据表中主轴位置数据。电子凸轮主、从轴位置关系是由位置数据表定义的。数据表中，以主轴位置为基准，等间距指定对应的从轴位置。EP 指令给出主轴位置基准的起点以及相邻两组数据对应的主轴位置间距。这个指令也就定义了电子凸轮数据表的大小。凸轮表最多可以有 257 组数据。

参数：EP m, n

m	1~32767 之间正整数，表示相邻两组数据对应的主轴位置间距。
m=?	返回当前的设置值。
n	-2147483648~2147483647 之间的整数，表示第一组数据对应的主轴位置。

应用：

运动中	是
程序中	是
指令行	是

默认值：

默认值	—
预设格式	—

操作数应用：

_EP	返回 m 参数的设置值。

相关指令：

EA	电子凸轮主轴
EB	电子凸轮模式使能
EC	电子凸轮表索引
EG	电子凸轮启动
EM	电子凸轮周期
EQ	电子凸轮脱离
ET	电子凸轮表数据
EW	电子凸轮加宽数据间隔

举例：

EP 20, 100	电子凸轮数据表起点为 100，间距为 20。
D=_EP	主轴变量 D 设置为当前凸轮表资料间距的值。

EQ

功能：电子凸轮脱离（ECAM Quit）

说明：

EQ 指令使电子凸轮中的从轴在主轴运动到指定位置时脱离与主轴的位置关系。每个从轴可以独立指定不同的脱离位置。如果指定的位置超出了主轴一个周期内的范围，则从轴立即与主轴脱离。

参数：EQ n, n, n, n　　或　EQx=n

n 为主轴位置，当主轴运动到这一位置时，从轴与主轴脱离电子凸轮的位置关系。

n=?　当从轴用 EG 指令设置了启动位置但尚未启动，返回 1

当从轴用 EQ 指令设置了脱离位置但尚未脱离，返回 2

当从轴已经完成了启动或脱离，返回 0

应用：		
	运动中	是
	程序中	是
	指令行	是
默认值：	默认值	—
	预设格式	—

操作数应用：

_EQx	从轴的跟随状态： 当从轴用 EG 指令设置了启动位置但尚未启动，返回 1 当从轴用 EQ 指令设置了脱离位置但尚未脱离，返回 2 当从轴已经完成了启动或脱离，返回 0

相关指令：

EA	电子凸轮主轴
EB	电子凸轮模式使能
EC	电子凸轮表索引
EG	电子凸轮启动
EM	电子凸轮周期
EP	电子凸轮表定义
ET	电子凸轮表数据
EW	电子凸轮加宽数据间隔

举例：

EQ 300, 700	在 A 轴与 B 轴分别在主轴位置为 300 与 700 时脱离电子凸轮运动

ES

功能：椭圆比例（Ellipes Scale）

说明：

ES 指令用于补偿二维插补时两轴分辨率的不同。如果参与二维插补的两个轴的位置分辨率不同，也就是说一个脉冲对应的长度不同，这样使用的正常的原弧插补指令得到的实际运动轨迹就会变成椭圆。用 ES 指令补偿两轴分辨率的比例后，就可以得到真正的圆形轨迹。

使用者也可以用这种方式得到所需比例的椭圆运动轨迹，但这种情况下，运动中的向量速度并不均匀。

参数：ES m, n

m 与 n 为 1~65535 的整数。用于表示两轴分辨率的比例关系。以 VM 指令的参数顺序决定对应的轴。

应用：	运动中	否
	程序中	是
	指令行	是
默认值：	默认值	1, 1
	预设格式	—

相关指令：

CR	圆弧
VM	平面（二维）向量模式
VP	向量位置

举例：

#A	程序标号
DP 0, , 0	将 A 轴、C 轴当前位置清零
VM CA	设置 C、A 两轴进行平面（二维）向量插捕
ES 2, 3	补偿 C、A 两轴的分辨率比例
VP 1000, 2000	插补运动向量
VE	插补曲线结束
BGS;AMS	开始运动；等待运动结束，
TD	输出位置：A 轴位置为 3000，C 轴为 1000
DP 0, , 0	将 A 轴、C 轴当前位置再次清零
VM CA	设置 C、A 两轴进行平面（二维）向量插捕
ES 3, 2	按照不同的比例补偿 C、A 两轴的分辨率
VP 1000, 2000	设置同样的插补运动向量
VE	插补曲线结束
BGS; AMS	开始运动；等待运动结束，
TD	输出位置：A 轴位置为 2000，C 轴为 1500
EN	

ET

功能：电子凸轮表数据（ECAM Table）

说明：

ET 指令设置电子凸轮表中的从轴位置数据。主轴的位置数据由 EP 指令定义，不需要逐一设置。从轴的数据项(n)就是当主轴在位置(n * i) + o 时，从轴的位置。其中的 i 与 o 就是 EP 指令设置的主轴位置间距和主轴位置起点。

参数：ET[m]=n, n, n, n

m	0~256 之间的整数。
n	-2147483648~2147483647 之间的整数。
n=?	返回设置值。

应用：

运动中	是
程序中	是
指令行	是

默认值：

默认值	—
预设格式	—

相关指令：

EA	电子凸轮主轴
EB	电子凸轮模式使能
EC	电子凸轮表索引
EG	电子凸轮启动
EM	电子凸轮周期
EP	电子凸轮表定义
EQ	电子凸轮脱离
EW	电子凸轮加宽数据间隔

举例：

ET[0]=0, , 0	输入 A 轴、C 轴在凸轮主轴位于周期起点时的位置
ET[1]=1200, , 400	输入 A 轴、C 轴在凸轮主轴位于第二点时的位置
EC0	设置数据索引为 0，后面 ET 指令不需要[m]参数
ET 0, , 0	输入 A 轴、C 轴在凸轮主轴位于周期起点时的位置
ET 1200, , 400	输入 A 轴、C 轴在凸轮主轴位于第二点时的位置

EW

功能：电子凸轮加宽数据间隔（ECAM Widen Segment）

说明：

EW 指令可以在电子凸轮数据中指定一个或两个数据项之间的间隔。通常，电子凸轮数据表中相邻两个数据项之间对应的主轴位置间距是由 EP 指令统一设置的。但是 EW 指令可以改变其中一或两个间距的大小。

参数：EW m1=n1, m2=n2

m1	为第一个需要调整的间距的索引。m1 为 1~255 之间的整数。
n1	为两个数据项之间对应主轴位置的间距。n1 为 1~2147483647 之间的整数。
m2	为第二个需要调整的间距的索引。m2 为 2~255 之间的整数。
n2	为两个数据项之间对应主轴位置的间距。n1 为 1~2147483647 之间的整数。

如果 m1 或 m2 设置为-1，则取消间距调整。m2 必须大于 m1，m2 不不能单独使用。

应用：

运动中	是
程序中	是
指令行	是

默认值：

默认值	-1, 0, -1, 0
预设格式	—

操作数应用：

_EW0	m1 的设置值
_EW1	n1 的设置值
_EW2	m2 的设置值
_EW3	n2 的设置值

相关指令：

EA	电子凸轮主轴
EB	电子凸轮模式使能
EC	电子凸轮表索引
EG	电子凸轮启动
EM	电子凸轮周期
EP	电子凸轮表定义
EQ	电子凸轮脱离
ET	电子凸轮表数据

举例：

EW 41=688	第 41 个间距调整为 688
EW 41=688, 124=688	第 41 和 124 个间距调整为 688

EY

功能：电子凸轮循环计数（ECAM Cycle Count）

说明：

EY 指令用于设置或查询电子凸轮运动的循环次数。EM 指令定义了电子凸轮一个运动周期对应主轴运动距离。在电子凸轮模式下，主轴每向前运动一个周期的距离，EY 指令对应的数值自动加 1；主轴每向后运动一个周期的距离，EY 指令对应的数值自动减 1。可以用下面的算式得到主轴的真实位置。

主轴绝对位置 = EY * EM + TP

参数：EM n

n 为带符号的整数，范围-2147483648~2147483647。

n=? 返回当前计数值

应用：

运动中	是
程序中	是
指令行	是

默认值：

默认值	—
预设格式	—

操作数应用：

_EY	当前计数值。

相关指令：

EM	电子凸轮周期

举例：

MG _EY*_EMB+_TPB	输出电子凸轮主轴（B 轴）的当前绝对位置

FE

功能：搜寻原点开关信号边缘（Find Edge）

说明：

FE 指令和 BG 指令使电机运动，直到该轴的原点输入信号产生变化。运动的方向取决于原点输入信号的初始状态。（CN 指令可以改变原点信号的极性定义。）检测到信号变化后，电机减速停止。

电机运动的速度、加速度、减速度，分别由指令 SP、AC、DC 设置。

参数：FE xxxx

x 为 A、B、C、D 或这些轴的任意组合，省略所有参数则表示使用全部参数。

应用：

运动中	否
程序中	是
指令行	是

默认值：

默认值	—
预设格式	—

相关指令：

FI	搜寻索引信号
HM	原点
BG	开始运动
AC	加速度
DC	减速度
SP	速度

举例：

FE	设置各轴搜寻原点信号变化边缘
BG	开始运动
FEA	设置 A 轴搜寻原点信号变化边缘
BGA	开始运动
FEB	设置 B 轴搜寻原点信号变化边缘
BGB	开始运动
FECD	设置 C、D 轴搜寻原点信号变化边缘
BGCD	开始运动

FI

功能：搜寻索引信号（Find Index）

说明：

FI 指令与 BG 指令使电机运动，直至检测到编码器的索引信号（标志脉冲、零位信号）由低变高。电机运动的速度与方向由 JG 指令设置。控制器检测到编码器的索引信号后，电机立即停止（无减速过程），并将当前位置设置为 0。

参数：FIxxxx

x 为 A、B、C、D 或这些轴的任意组合，省略所有参数则表示使用全部参数。

应用：

运动中	否
程序中	是
指令行	是HM

默认值：

默认值	—
预设格式	—

相关指令：

FE	搜寻原点开关信号边缘
HM	原点
BG	开始运动
AC	加速度
JG	点动（JOG）

举例：

#HOME	程序标号
JG 500	设置速度与方向
FIA	搜寻编码器索引信号
BGA	开始运动
AMA	等待运动结束
MG “FIND INDEX”	输出信息
EN	程序结束

FL

功能：正向软件极限（Forward Software Limit）

说明：

FL 指令用于设置正向软件极限位置。如果运动超过了这个极限位置，该轴会减速停止。控制器不允许超过这个极限位置的正向运动。

如果程序中含有标号#LIMSWI，在运动超过软件极限位置时，控制器会自动从标号#LIMSWI 开始执行程序。

参数：BL n, n, n, n　　或　BLx=n

n 为带符号整数，范围-2147483648~2147483647。正向极限位置为 n+1。

n=2147483647，关闭软件极限

n=?　反回当前的软件极限设置值

应用：

运动中	是
程序中	是
指令行	是

默认值：

默认值	2147483647
默认格式	位置格式

操作数应用：

_FLx	该轴正向软件极限的设置值。

相关指令：

BL	正向极限
PF	位置格式

举例：

#TEST	程序标号
AC 1000000	设置 A 轴加速度
DC 1000000	设置 A 轴减速度
BL 15000	设置 A 轴正向极限位置
JG 5000	点动
BGA	开始 A 轴运动
AMA	等待 A 轴运动结束（超过极限位置）
TDA	输出当前位置
EN	程序结束

提示：

GALIL 控制器也提供硬件极限。无论软极限或硬极限，都会引发#LIMSWI 的运行。

@FRAC[n]

功能：取小数部分（Fraction）

说明：

返回给定数据的小数部分。

参数：@FRAC[n]

n 带符号数，-2147483648~2147483647。

应用：

运动中	是
程序中	是
指令行	是

默认值：

默认值	—
预设格式	—

相关指令：

@INT[n]	取整数部分

举例：

:MG @FRAC[-2.4]

-0.4000

:MG @FRAC[1.2]

0.2000

GA

功能：电子齿轮主轴（Gear Master）

说明：

GA 指令用于定义电子齿轮的主轴。各轴的主轴可以相同也可以不同。Galil 控制器的电子齿轮功能支持一主多从以及多组主从关系的运动模式。主轴可以是任意轴的编码器回馈位置、输出脉冲位置或轨迹规划位置。也可以将虚拟轴或 LM 或 VM 插补的组合轴作为主轴（这两个轴没有编码器回馈位置与输出脉冲位置，而且组合轴的运动方向永远为正）。

电子齿轮的同步比例由指令 GR 设置，将同步比例设置为 0 可可以使该轴退出电子齿轮同步模式，不再跟踪主轴的运动。

参数：GA n, n, n, n　　或　GAx=n

n 可以是 A、B、C、D，这表示以相应轴的编码器回馈位置作为跟踪的主轴。

n 可以是 CA、CB、CC、CD，这表示以相应轴的轨迹规划位置作为跟踪的主轴。

n 可以是 DA、DB、DC、DD，这表示以相应轴的脉冲输出位置作为跟踪的主轴。

n 可以是 S 或 N，这表示以相应轴的轨迹规划位置作为跟踪的主轴。

应用：

运动中	否
程序中	是
指令行	是

默认值：

默认值	—
预设格式	—

相关指令：

GR	电子齿轮同步比例
GM	龙门同步模式

举例：

#GEAR	程序标号
GA , CA, S	设置 A 轴为 B 轴的主轴，S 轴为 C 轴的主轴。
GR , 0.5, -2.5	设置同步比例。
JG 5000	A 轴点动。
BGA	A 轴开始运动。B 轴也会跟踪 A 轴一起运动，方向与 A 轴相同，速度是 A 轴的一半。
WT 1000	等待 1 秒
STA	使 A 轴停止，B 轴将同步停止
AMA	等待 A 轴静止
EN	程序结束

注意：控制器不允许任何一个轴以自身作为同步的主轴，GA A、GA CA、GA DA 都是错误的用法。

GD

功能：电子齿轮同步过渡距离（Gear Distance）

说明：

GD 指令以主轴脉冲为基准，在齿轮模式打开/关闭或改变比例的时侯，为从轴设置一个逐渐实现同步的距离。设置的单位是主轴的输出脉冲或编码器回馈计数。设置的距离为绝对值，无论主轴向那个方向运动或 GR 设置为正或负。如距离设置为 0，则 GR 指令改变同步比例时，从动电机会立即回应。

参数：GD n, n, n, n　　或　GDx=n

n 为 0 至 32767 之间的整数。

n=0　从轴在改变痛痹比例时无过渡过程

n=?　返回当前设置值

应用：

运动中	是
程序中	是
指令行	是

默认值：

默认值	0
预设格式	5.0

操作数应用：

_GDx	该轴的设置值。

相关指令：

_GP	电子齿轮同步相位差
GR	电子齿轮同步比例
GA	电子齿轮同步主轴

举例：

#A	程序标号
GA, DA	设置 B 轴以 X 轴的输出脉冲为主轴进行电子齿轮同步跟踪
GD, 5000	设置过度距离为 5000 脉冲
JG5000;BGA	A 轴点动
AS A	等待 A 轴位置达到 5000 脉冲/秒
GR, 1	B 轴开始与A 轴同步
WT1000	等待 1 秒
GR, 3	改变 B 轴同步比例，可以看到 B 轴速度逐渐加快
WT1000	等待 1 秒
GR, 0	停止 B 轴与A 轴的同步关系，可以看到 B 轴减速到静止
EN	程序结束

GM

功能：龙门同步模式（Gantry Mode）

说明：

GM 指令使电子齿轮中的从轴使用龙门同步模式。在这个模式下，从轴不回因为 ST 指令以及位置极限信号停止。

参数：GM n, n, n, n　　或　GMx=n

n=0　龙门同步模式无效。

n=1　龙门同步模式有效。

n=?　返回当前设置值。

应用：

运动中	是
程序中	是
指令行	是

默认值：

默认值	0
预设格式	1.0

操作数应用：

_GMx	表示该轴的设置值。

相关指令：

GR	电子齿轮同步比例
GA	电子齿轮同步主轴

举例：

GM 1, 1, 1, 1	设置所有轴为龙门同步模式
GM 0	设置 A 轴龙门同步模式无效
GM , , 1, 1	设置 C、D 轴为龙门同步模式
GM 1, 0, 1, 0	设置 A、C 轴为龙门同步模式；B、D 轴龙门同步无效

提示：龙门同步模式适用于对大型负载进行双边驱动的模式。

_GP

功能：电子齿轮同步相位差（Gearing Phase Different）

说明：

操作数_GPx 表示电子齿轮同步中某从轴与主轴的累计的相位误差。这个相位误差是指电子齿轮中由于从轴的同步比例逐渐变化而造成位置相对于主轴的超前或滞后。如果 GD 指令设置为 0，则这个相位误差就不会再变化。

相关指令：

GR	电子齿轮同步比例
GA	电子齿轮同步主轴

举例：

#A	程序标号
JGB=1000	B 轴为点动模式
BG B	B 轴开始运动
GA DB	以 B 轴输出脉冲为 A 轴的主轴
GD 1000	设置 A 轴的同步过渡距离
AI=-1	等待通用输入 1 导通
GR 1	A 轴开始电子齿轮同步
P1=_TDB	读当前 B 轴位置
MF, (P1+1000)	等待同步过渡过程结束
IP_GPA	补偿相位误差，但此指令不影响_GPA 的值
EN	程序结束

GR

功能：电子齿轮同步比例（Gear Ratio）

说明：

GR 指令在电子齿轮模式下指定从轴的同步比例。主轴由 GA 指令设置。每个轴的同步比例独立设置。主轴可以双向运动。同步比例为正，表示从轴与主轴运动方向相同；同步比例为负表示从轴与主轴运动方向相反；同步比例为 0 表示该从轴退出电子齿轮模式。同步比例大于 1，表示从轴速度大于主轴速度。

除非使用龙门同步模式，否则从轴的位置极限信号，针对从轴的 ST 指令都会使从轴停止并退出电子齿轮模式。

参数：GR n, n, n, n　　或　GRx=n

n 为带符号的数，范围+/-127，分辨率为 1/65536。

n=0　退出电子齿轮模式

n=?　返回当前设置值

应用：

运动中	是
程序中	是
指令行	是

默认值：

默认值	0
预设格式	3.4

操作数应用：

_GRx	表示该轴的设置值。

相关指令：

GA	电子齿轮同步主轴
GM	龙门同步模式

举例：

#GEAR	程序标号
MO B	关闭 B 轴（B 轴由外部控制）
GA B, , B	A 轴、C 轴以 B 轴的编码器输出为主轴
GR .25, , -5	A 轴与 B 轴同向，速度为 B 轴的 1/4；C 轴与 B 轴反向，速度为 B 轴的 5 倍
EN	程序结束

HM

功能：原点（Home）

说明：

HM 指令的作用是运动到原点开关信号变化的位置，并将其作为系统的原点。这个过程分为两个部分：

1、电机按照 AC 指令设置的加速度以及 SP 指令设置的速度运动，直到该轴的原点输入信号产生变化。电机按照 DC 指令设置的减速度停止。运动的方向取决于原点输入信号的初始状态。（CN 指令可以改变原点信号的极性定义。）

2、电机改变运动方向，按照 256 的速度缓慢运动，直至检测到该轴的原点输入信号产生反向的变化。检测到信号变化后，电机立即停止，并将此位置设置为 0。

参数：HM xxxx

x 为 A、B、C、D 或这些轴的任意组合，省略所有参数则表示同时使用所有参数。

应用：

运动中	否
程序中	是
指令行	是

默认值：

默认值	—
预设格式	—

操作数应用：

_HMx	表示该轴原点输入信号状态。默认状态下，信号导通时，返回值为 0；信号截止机或悬空时，输入为 1。此值受 CN 指令设置的影响。

相关指令：

CN	设置
FI	搜寻索引信号
FE	搜寻原点开关信号边缘

举例：

HM	确认原点位置
BG	开始运动

提示：用户可以用 FI 与 FE 指令设计自己的原点搜索返回动作。

HS

功能：句柄交换（Handle Assignment Switch）

说明：

HS 用于在两个以太网句柄之间交换句柄的通讯对象。控制器的通讯句柄是由 IH 指令分配的，或是在其它设备与控制器建立通讯时自动分配的。如需要调整当前句柄分配，就需要用到 HS 指令。

参数：HSh=i

h 是需要交换的第一个句柄，为 A、B、C、D 或 S。

i 是需要交换的第二个句柄，为 A、B、C、D 或 S。

（S 代指发送本指令所使用的通讯句柄。）

应用：

运动中	是
程序中	是
指令行	是

默认值：

默认值	—
预设格式	—

相关指令：

IH	网络句柄

举例：

HSC=D	将远分配给句柄 C 的通讯与句柄 D 交换
HSS=B	将当前使用的通讯句柄与句柄 B 交换

HX

功能：中止运行（Halt Execution）

说明：

HX 指令用于中止控制器中运行的用户程序。

参数：HXn

n 为 0 至 3 之间的整数，表示运行程序的线程号。

如省略参数 n，则中止所有线程。

应用：

运动中	是
程序中	是
指令行	是

默认值：

默认值	—
预设格式	—

操作数应用：

_HXn	线程 n 的当前状态。
	0　线程未运行。
	1　线程在运行。
	2　线程由于 BK 指令设置的断点暂停。

相关指令：

XQ	运行程序

举例：

XQ#A	在 0 线程中，从#A 处开始运行程序
XQ#B, 3	在 3 线程中，从#B 处开始运行程序
HX0	中止 0 线程
HX3	中止 3 线程

IA

功能：IP 地址（IP Address）

说明：

IA 指令用于分配控制器的 IP 地址。在 TCP/IP 协议下，IA 也可以用来设置通讯超时的数值。IA 指令分配地址只能通过 RS232 接口发送执行。如通过网络接口发送 IA 指令来分配地址，那么在新地址生效后，通讯会立即被中断。

如控制器采用动态地址，则 IA 指令分配的地址无效。

参数：IA ip0, ip1, ip2, ip3　或　IA n　　或　IA<t

ip0、ip1、ip2、ip3 为逗号分开的四个单字节数，表示分配给控制器的 IP 地址。

n 为带符号 32bit 整数，表示分配给控制器的 IP 地址。

t 为 TCP/IP 协议中，重试的延迟时间，单位为毫秒，范围 1~2147483647。（最多重试 5 次）。

IA?　返回控制器的 IP 地址。

应用：

运动中	否
程序中	是
指令行	是

默认值：

默认值	n=0, t=250
预设格式	—

操作数应用：

_IA0	控制器的 IP 地址，一个 32bit 的带符号整数
_IA1	t 的值
_IA2	可用句柄数
_IA3	使用这个操作数的通讯句柄，0~3
_IA4	最近产生通讯错误的句柄号
_IA5	网络连接速度，10 或 100，如为连接网络，返回-1。

相关指令：

IH	网络句柄
DH	动态地址

举例：

IA 192, 168, 1, 9	设置控制器 IP 地址为 192.168.1.9
IA 3232235785	设置控制器 IP 地址为 192.168.1.9
IA<500	设置超时时间为 500ms

IF

功能：如果，用于条件分之结构

说明：

IF 指令与 ENDIF 指令一起组成条件分支结构。IF 指令的参数为一个逻辑条件。如果条件为真，IF 后面的指令会被执行。如果条件为假，控制器会忽略 IF 与 ELSE（或 ENDIF）之间的指令。

参数：IF(条件)

条件可以是由下符号组成的一个比较式：

<	小于
>	大于
=	等于于
<=	小于等于
>=	大于等于
<>	不等于

也可以多个比较式之间可以用逻辑运算符号 &（与） 及 |（或）组合，每个比较式必须在一个圆括号内。

（比较式也可以由单纯的数学表达式代替，计算结果为 0，等效于假，计算结果非零，等效于真。）

应用：

运动中	是
程序中	是
指令行	否

默认值：

默认值	—
预设格式	—

相关指令：

ELSE	否则，条件分支
ENDIF	条件分支结束

举例：

#A	程序标号
IF(_TDA<1000)	判断电机位置
MG”Motor is within 1000”	输出信息
ENDIF	结束 IF
EN	程序结束

IH

功能：网络句柄

说明：

当控制器作为客户端通过网络与服务器建立通讯时，使用 IH 指令。这个指令打开一个句柄并连接到一个服务器上。控制器只能支持 4 个句柄，分别用字母 A、B、C、D 表示。打开一个句柄，必须指定以下信息：

1、服务器的 IP 地址

2、使用的通讯协议：TCP/IP 或 UDP/IP

3、服务器的埠号。如果服务器不需要指定特别的埠号，也可以不指定，再此情况下，控制器预设使用埠 1000。

参数：IHh=ip0, ip1, ip2, ip3<p>q　　或　IHh=n<p>q　或　IHh=>-r

h 为句柄，A、B、C 或 D。

ip0、ip1、ip2、ip3 为 0~255 之间的整数，表示的 IP 地址。

n 为带符号 32bit 整数，表示服务器的 IP 地址。

p 为 0~65535 的整数，表示服务器上的埠号，此参数可选

q 为连接类型，省略时默认为 TCP

<0	无连接
1	UDP
2	TCP

r 为负数，表示中断并释放句柄

IHS=>r 中断并释放接收此指令所使用的句柄，r 为-1 或-2。

应用：

运动中	是
程序中	是
指令行	是

默认值：

默认值	—
预设格式	—

操作数应用：

_IHh0	句柄连接设备的 IP 地址。

相关指令：

IA	IP 地址

举例：

IHA=251, 29, 51, 1	打开句柄 A 与地址为 251, 29, 51, 1 的设备通讯
IHA=-2095238399	打开句柄 A 与地址为 251, 29, 51, 1 的设备通讯

提示：执行 IH 指令时，需要与服务器建立通讯，可能有一点时间延迟。

如端方设备的 IP 地址与 DMC-B140-M 控制器的 IP 地址不在同一子网范围内，或对方设备内对使用的埠、句柄有限制，都会导致无法建立通讯。

II

功能：输入中断（Input Interrupt）

说明：

II 指令设置由通用输入信号触发输入中断的功能。默认情况下，输入中断程序在通用输入信号导通状态被触发，也可以设置为信号截止状态触发中断程序。

如果在 0 线程有程序运行时，指定的输入信号状态有效，程序就回自动跳转到#ININT 开头的中断子程序中。结束输入中断子程序，使用指令 RI

参数：II m, n, o, p

m 为 0~8 的整数，0表示关闭中断。m 的值表示可以产生中断的最低的通用输入号。如果第二个参数 n 被省略，则只有 m 参数指定的一个通用输入可以触发中断程序。

n 为 2~8 的整数，这是一个可选参数，和参数 m 一起设置可触发中断程序的通用输入信号。如指令 II 2, 4 则只有通用输入 2、3 和 4 可以触发中断程序。

o 为 1~255 的整数，这个参数是以另外一种方式设置有效的通用输入信号。如果给定了 m 与 n，则这个参数被忽略。o 的值对应一个二进制数据。数据中某位为 1，则对应位的通用输入可以触发中断；为 0，则对应位的通用输入不能触发中断。例如 o=15，则对应 0001111。这个资料的低 4 位为 1，高 4 位为 0。这样通用输入 1、2、3、4 可以触发中断程序，通用输入 5、6、7、8 则不能触发中断程序。

p 为 1~255 的整数，这个参数用来设置前几个参数指定的通用输入信号在什么状态下触发中断程序。o 的值对应一个二进制数据。数据中某位为 1，则对应位的通用输入在截止状态下触发中断程序；为 0，则对应位的通用输入在导通状态下触发中断程序。例如 II1, 4, , 6，设置通用输入 1、2、3、4 可触发中断程序，其中通用输入 1、4 是在导通状态触发，通用输入 2、3 是在截止状态触发。

（控制器上的跳线 Abort 会改变通用输入 8 的导通/截止状态的作用。）

应用：

运动中	是
程序中	是
指令行	是

默认值：

默认值	—
预设格式	—

相关指令：

RI	从输入中断程序返回
#ININT	输入中断程序
AI	等待输入

举例：

```
#A                  程序标号
II 1                设置通用输入 1 导通时触发中断程序
JG 5000;BGA         A 轴点动
#LOOP;JP#LOOP       循环
EN                  程序结束
#ININT              输入中断程序
STA                 A 轴停止
MG"INTERRUPT"       输出信息
```

```
AMA                          等待 A 轴静止
#CLEAR
JP#CELAR, @IN[1]             等待通用输入 1 截止
BGA                          A 轴启动
RI 0                         自中断程序返回
```

IK

功能：阻塞以太端口

说明：

IK 指令可以阻止控制器从埠号低于 1000 的端口接受信息。其中 0、23、68、502 不受影响。

参数：IKn

n=0　允许控制器接受任何端口的信息

n=1　阻止控制器从埠号低于 1000 的端口接受信息。其中 0、23、68、502 不受影响。

n=?　返回当前设置值

应用：

运动中	是
程序中	是
指令行	是

默认值：

默认值	1
预设格式	1.0

相关指令：

TH	查询句柄状态
IH	网络句柄

举例：

IK1	阻止不需要的端口通讯
IK0	允许所有埠

IN

功能：输入变量（Input）

说明：

IN 指令允许用户从通讯口输入一个变量的值。当程序运行 IN 指令时，首先从通讯口输出提示信息，而到等待接收数值和回车符号，而后将接收到的数值赋给指定的变数。

IN 指令会暂停程序的运行，直至从通讯口接收到回车或分号。如果在回车或分号前没有收到数值，则变数保留其原来的值。在暂停期间输入中断、错误中断以及位置极限信号均可正常响应。

IN 指令只能用于 0 线程。

参数：IN “m”, n

m 为提示信息。

n 为变量名

m 与 n 的总字符数不能超过 40，不要在逗号前后插入空格。

应用：

运动中	是
程序中	是
指令行	否

默认值：

默认值	—
预设格式	—

举例：

#A	程序标号
IN “Enter Speed(mm/sec)”, V1	提示操作人员输入速度
IN “Enter Length(mm)”, V2	提示操作人员输入长度
V3=V1*4000	转换速度单位为脉冲/秒
V4=V2*4000	转换长度单位为脉冲
SP V3	设置运动速度
PR V4	设置运动距离（长度）
BGA	开始运动
AMA	等待运动结束
MG “MOVE DONE”	输出信息
EN	程序结束

@IN[n]

功能：读通用数字输入

说明：

返回指定通用数字输入信号的值。输入信号导通时，返回 0；输入信号截止时，返回 1。（控制器上的 Abort 跳线会影响通用输入 8 的返回值。）

参数：@IN[n]

n 为 1~8 的整数。

应用：

运动中	是
程序中	是
指令行	是

默认值：

默认值	—
预设格式	—

相关指令：

@OUT[n]	读数字输出

举例：

:MG @IN[1]	输出通用输入 1 的状态
1.0000	*控制器输出 1.0000*
:x=@IN[1]	将通用输入 1 的状态赋值给变量 x
:x=?	查询变数 x 的值
1.0000	*控制器输出 1.0000*

#ININT

功能：输入中断处理子程序

说明：

在控制器 0 线程运行程序时，如 II 指令设置的输入状态出现，则#ININT 开头的子程序会自动在 0 线程中运行。

如子程序运行完成时，通用输入状态未改变，仍然符合 II 指令的设置，则子程序将再次自动运行。

输入中断子程序由 RI 结束。

应用：

运动中	是
程序中	是
指令行	否

相关指令：

II	输入中断
@IN[n]	读数字输入
RI	自输入中断程序中返回

举例：

#A	程序标号
II1	通用输入 1 导通时触发中断程序
#MAIN	主程序
MG”MAIN”	输出信息
WT1000	等待 1 秒
JP#MAIN	循环
#ININT	输入中断程序
MG”ININT”	输出信息
AI1	等待通用输入 1 截止
RI	返回主程序

@INT[n]

功能：取整数部分

说明：

返回给定数据的整数部分。

参数：@INT[n]

n 为带符号数，范围-2147483648~2147483647。

应用：

运动中	是
程序中	是
指令行	是

相关指令：

@FRAC[n]	取小数部分

举例：

```
:MG @INT[1.2]
 1.0000
:MG @INT[1.9]
 1.0000
:MG @INT[-2.4]
 -2.0000
:MG @INT[-2.99]
 -2.0000
```

IP

功能：增量位置（Increment Position）

说明：

IP 指令可以在电机运动过程中改变电机的规划位置。IP 指令不需要用 BG 指令开始运动。当电机处于不同运动状态时，IP 指令可以有 4 种作用：

1、电机不处于运动状态：

IP 指令相当于 PR 与 BG 的组合，电机按照给定的速度（SP）、加速度（AC）和减速度（DC）运动。

2、电机处于独立定位运动（PA、PR、IP）过程中：

IP 指令使电机以之前的目标位置加上 IP 指令设置的增量为新的目标位置（电机的运动方向必须与 IP 指令给定的增量方向一致）。

3、电机作为电子齿轮的从轴正在跟踪主轴运动：

IP 指令使电机在同步的基础上迭加一个增量定位运动，迭加的速度、加速度、减速度、位移量分别由 SP、AC、DC、IP 指令设置。

4、电机处于点动（JG）过程中：

IP 指令使电机在点动的基础上瞬间运动指定的位置增量，这一瞬间的运动状态与 SP、AC、DC 设置无关。在此模式下，由于是瞬间运动，IP 的参数一定要比较小，否则可能导致电机无法跟上脉冲信号，甚至控制器来不及发出脉冲信号。

参数：IP n, n, n, n　　或　IPx=n

n 为带符号的整数，在-2147483648 至 2147483647 之间。

n=?　返回该轴的当前编码器回馈位置

应用：

运动中	是
程序中	是
指令行	是

默认值：

默认值	—
默认格式	位置格式

举例：

IP 50	向前运动 50 脉冲
#CORRECT	程序标号
AC 100000	设置加速度
JG 10000;BGA	点动
WT 1000	等待 1 秒
IP10	电机立即前进 10 个脉冲
STA;AMA	停止
EN	程序结束

IT

功能：独立时间常数——平滑功能

说明：

IT 指令在独立运动（JG、PA、PR）中对加减速过程进行滤波，产生平滑的速度曲线。得到的速度曲线加、减速是连续的变化，可以减小对机械的冲击。IT 指令设置滤波器的带宽，为 1 时无滤波作用，0.004 时最大滤波作用。要注意，这样滤波的会延长运动时间。

IT 指令对 AD、AR 的触发时间没有影响，这两个指令的触发是检测的滤波前的运动曲线。当 IT 不为 1 时，有可能在实际位置到达前，AD、AR 已经被触发。

参数：IT n, n, n, n　　或　ITx=n

n 为大于 0 小于等于 1 的数, 分辨率为 1/256。

n=?　　返回当前设置的值

应用：

运动中	是
程序中	是
指令行	是

默认值：

默认值	1
预设格式	1.4

操作数应用：

_ITx	x 轴的设置值。

相关指令：

PR	增量位置
PA	绝对位置
JG	点动
VM	平面向量模式
LM	直线插补模式

举例：

IT 0.8, 0.6, 0.9, 0.1	设置 A、B、C、D 轴的时间常数
IT ?	查询 A 轴时间常数
0.8	

JG

功能：点动（Jog）

说明：

JG 指令设置点动模式，并指定了点动的方向和速度。

参数：JG n, n, n, n　　或　JGx=n

n 为带符号的数，范围 0 至+/-3000000。（对于虚拟轴可使用 JGN=n）

n=?　返回该轴点动速度的绝对值

应用：

运动中	是
程序中	是
指令行	是

默认值：

默认值	25000
预设格式	8.0

操作数应用：

_JGx	x 轴点动速度的绝对值。

相关指令：

BG	开始运动
ST	停止
AC	加速度
DC	减速度
IP	增量位置

举例：

JG 100, 500, 2000, 5000	设置各轴点动方向与速度
BG	开始运动
JG, , -2000	改变 C 轴的运动方向

JP

功能：跳转（Jump）

说明：

JP 为程序运行的条件跳转指令。JP 指令的参数包括一个指定的程序位置和一个条件。这个指定位置可以是一个行号或标号。如条件为真，程序跳转到指定位置运行。如条件为假，程序顺序执行 JP 后面的指令。如省略条件，则默认条件为真。

参数：JP 位置, 条件

位置　程序中的行号或者标号

条件可以是由下符号组成的一个比较式：

<	小于
>	大于
=	等于于
<=	小于等于
>=	大于等于
<>	不等于

也可以多个比较式之间可以用逻辑运算符号 &（与） 及 |（或）组合，每个比较式必须在一个圆括号内。

（比较式也可以由单纯的数学表达式代替，计算结果为 0，等效于假，计算结果非零，等效于真。）

应用：

运动中	是
程序中	是
指令行	否

默认值：

默认值	—
预设格式	—

相关指令：

JS	转子程序
IF	如果，条件分支
ELSE	否则，条件分支
ENDIF	条件分支结束

举例：

JP #POS1, V1<5	如果变量 V1 的值小于 5，则程序跳转到标号#POS1 处运行，否则程序顺序执行
JP #A, V7*V8=0	如果变量 V7*V8 的值等于 0，则程序跳转到标号#A 处运行，否则程序顺序执行
JP #B, (V2<5)&(V2>3)	如果变量 V2 的值在 3 与 5 之间，则程序跳转到标号#B 处运行，否则程序顺序执行
JP #C	程序跳转到标号#C 处运行

JS

功能：跳转子程序（Jump to Subroutine）

说明：

JS 为调用子程序的指令。JS 指令的参数包括一个指定子程序入口和一个条件。这个指定子程序入口可以是一个行号或标号。如条件为假，程序顺序执行 JS 后的指令。如条件为真，程序调用该子程序，并在子程序运行结束后返回运行 JS 后的指令。如省略条件，则默认条件为真。

子程序允许递归调用。子程序最多嵌套 16 层。

参数：JS 位置, 条件

位置　程序中的行号或者标号

条件可以是由下符号组成的一个比较式：

<	小于
>	大于
=	等于于
<=	小于等于
>=	大于等于
<>	不等于

也可以多个比较式之间可以用逻辑运算符号 &（与） 及 |（或）组合，每个比较式必须在一个圆括号内。

（比较式也可以由单纯的数学表达式代替，计算结果为 0，等效于假，计算结果非零，等效于真。）

应用：

运动中	是
程序中	是
指令行	否

默认值：

默认值	—
预设格式	—

相关指令：

EN	程序（子程序）结束
JP	跳转

举例：

JS #SQUARE, V1<5	如果变量 V1 的值小于 5，则运行标号#SQUARE 开始的子程序
JS #LOOP, V1<>0	如果变量 V1 的值不等于 0，则运行标号#LOOP 开始的子程序
JS #A	运行标号#A 开始的子程序

KS

功能：脉冲输出平滑

说明：

KS 指令设置输出脉冲的平滑时间。在控制整步或半步驱动的步进电机时，这个功能非常有效。KS 对控制器内部的轨迹规划没有影响，KS 值设置得越大，对输出脉冲的速度变化越平滑。在定位运动中，实际运动时间比规划运动时间大约多出 3 倍于 KS 值的控制器内部工作周期时间。

注意：KS 使得在运动中输出脉冲位置滞后于控制器的规划位置。

参数：KS n, n, n, n　　或　KSx=n

n 为 0.5 到 64 之间的数，分辨率为 1/32。

n=?　　返回当前设置值

应用：

运动中	是
程序中	是
指令行	是

默认值：

默认值	2
预设格式	2.3

相关指令：

TM	工作周期

操作数应用：

_KSx	x 轴当前设置值。

举例：

KS 2, 4, 8	设置 A、B、C 轴
KS 5	设置 A 轴
KS, , 15	设置 C 轴

提示：

1、 KS 的最小值与控制器的工作周期（TM）设置有关。预设状态下，TM 为 1000，KS 最小值为 0.5。当 TM 为 500 时，KS 最小值为 1。当 TM 为 250 时，KS 最小值为 2。

2、 控制器允许在运动中改变 KS 设置，但是用户不要在电机运动中将该轴 KS 设置值改小，否则可能产生不可预期的错误。

LA

功能：列出数组（List Arrays）

说明：

LA 指令列出控制器中所有的用户数组。数组按照名称的字符顺序列出，数组名后是数组中的元素数量。

参数：无

应用：

	运动中	是
	程序中	是
	指令行	是

默认值：

默认值	—
预设格式	—

相关指令：

LL	列出标号
LS	列出程序
LV	列出变数

举例：

```
:LA
CA[10]
LA[5]
NY[25]
VA[17]
```

LC

功能：低电流模式（Low Current Mode）

说明：

LC 指令使控制器的使能输出信号在电机运动结束后自动关闭电机，在电机需要运动时，使能信号会自动提前打开电机。

参数：LC n, n, n, n　　或　LCx=n

n 为 1~32767 之间的整数，表示在脉冲信号输出完成后延迟多少个控制器工作周期后关闭电机。

n=0　取消低电流模式，运动结束后不自动关闭电机。

n=?　返回当前该轴设置值。

应用：

运动中	是
程序中	是
指令行	是

默认值：

默认值	0
预设格式	5.0

操作数应用：

_LCx	x 轴当前设置值。

相关指令：

SH	打开电机
MO	关闭电机

举例：

LCC=1

提示：在低电流模式下，控制器会在电机需要运动前打开电机。电机打开信号仅比脉冲信号提前一个工作周期。如电机驱动器的开关信号（使能、脱机信号）中间需要经过继电器处理， 可能由于信号处理造成的响应延迟造成驱动器没能正确接收最初的脉冲信号。

LD

功能：位置极限信号无效（Limit Disable）

说明：

LD 指令可以使位置极限输入信号（限位开关）无效。当硬件信号被设置为无效时，FL 与 BL 指令设置的软件极限依然有效。当由于干扰造成极限信号不稳定时，可以用 LD 使极限信号无效，以避免错误对电机运动的影响。如系统中确实不需要位置极限信号保护机构的行程范围，也可以用 LD 指令使极限信号无限。这样就可以将控制器上的极限输入信号当作通用输入信号使用。

参数：LD n, n, n, n　　或　LDx=n

n=0	极限信号有效（默认状态）
n=1	前极限信号无效
n=2	后极限信号无效
n=3	前、后极限信号均无效
n=?	返回该轴当前设置值

应用：

运动中	是
程序中	是
指令行	是

默认值：

默认值	0
预设格式	1.0

操作数应用：

_LDx	x 轴当前设置值。

相关指令：

_LF	前极限状态
_LR	后极限状态
BL	反向软件极限
FL	正向软件极限

举例：

LDA=1	使 A 轴的前极限信号无效

LE

功能：直线插补路径结束（Linear Interpolation End）

说明：

LE 指令用于表示直线插补路径结束。LE 与 VE 指令可以互换。

参数：LE　　或　　LE?

LE?	返回直线插补路径的总长度

应用：

运动中	是
程序中	是
指令行	是

默认值：

默认值	0
预设格式	PF

操作数应用：

_LE	直线插补路径的总长度，当插补运动结束后，返回值为 0。

相关指令：

LI	直线插补路径
BG	开始运动
LM	直线插补模式
VS	插补向量速度
VA	插补向量加速度
VD	插补向量减速度

举例：

LM CD	C、D 轴组成直线插补模式
LI, , 100, 200	插补路径
LE	插补路径结束
BGS	开始运动

_LF

功能：前极限信号状态（Forward Limit Switch）

说明：

操作数_LFx 表示 x 轴前极限信号是否有效，值为 0 或 1。

1 当前 x 轴前极限信号无效

0 当前 x 轴前极限信号有效

极限信号在什么状态有效，由 CN 指令设置。在使用 LD 指令设置极限信号无效后，操作数依然为可以反映出相应极限信号的状态。

相关指令：

CN	设置

举例：

MG _LFA	输出当前 A 轴前极限信号状态

LI

功能：直线插补路径（Linear Interpolation）

说明：

LI 指令用于直线插补模式，向插补路径缓冲区中增加一段插补路径。LI 指令的参数为参与直线插补的各轴在本段路径中运动的距离。缓冲区中最多可以存储 31 段路径，所以在开始运动前，最多可以 31 个 LI 指令。运动开始后，运动完成的路径段就会从缓冲区中释放，缓冲区中就有空间可以执行新的 LI 指令。运动过程中，只要缓冲区有空间，就可以不断执行 LI 指令，实现无限长度的插补路径。

插补运动按照给定的向量速度进行，控制器自动计算每段路径中各轴电机的速度。也可以在 LI 指令中指定该段路径的运动速度与末速度。

参数：LI n, n, n, n<o>p

n 为带符号整数，范围-8388607~8388607。表示在此段路径中，该轴的移动距离与方向。至少要有一个参数不为 0。

o 为无符号偶数，2~3000000，表示运动到该段路径时的向量速度。

o=-1　表示使用 VV 指令设置的速度。

p 为无符号偶数，2~3000000，表示运动到该段路径末尾时的向量速度。

应用：

运动中	是
程序中	是
指令行	是

默认值：

默认值	—
预设格式	—

相关指令：

LE	直线插补路径结束
LM	直线插补模式
VV	向量速度变量

举例：

LM ABC	A、B、C 三轴组成直线插补模式
LI 1000, 2000, 3000	定义插补路径
LE	路径结束
BGS	开始运动

#LIMSWI

功能：位置极限处理子程序

说明：

控制器运行用户程序时，如位置极限信号有效，而且用户程序中包含#LIMSWI 开头的子程序，则自动在 0 线程中运行该子程序。如 0 线程有程序在运行，会被打断。

如果是以下情况，位置极限信号不影响电机当前的运动，#LIMSWI 子程序也不会运行。

1、轴静止时，该轴的位置极限信号有效

2、电机从静止向位置极限信号有效的方向运动（但这样会产生指令错误）

3、轴运动时，与运动方向相反的位置极限信号有效

4、用 LD 指令屏蔽了硬件位置极限信号时，硬件极限信号处于有效的电气状态

（当某轴为电子齿轮龙门同步模式的从轴时，运动方向上的位置极限信号虽然不影响电机的运动，但#LIMSWI 子程序也会自动运行。）

无论 LD 指令对硬件位置极限信号的作用如何设置，当电机运动超出软极限位置时，也会使#LIMSWI 子程序自动运行。

如果#LIMSWI 子程序运行时间很短，电机减速停止的过程中#LIMSWI 子程序可能会自动运行多次。

#LIMSWI 子程序用 RE 指令返回。

应用：

运动中	是
程序中	是
指令行	否

相关指令：

_LF	前位置极限信号状态
_LR	后位置极限信号状态

举例：

```
#Main                    程序标号
FL 10000                 设置 A 轴正向软件极限
JG 1000;BGA              A 轴点动
#LOOP                    循环
MG "0";WT1000            每秒输出一次信息
JP#LOOP
#LIMSWI                  位置极限子程序
MG "1"                   输出信息
AM A                     等待 A 轴减速静止
RE 0                     从位置极限子程序返回
```

<control>L<control>K

功能：锁定程序（Lock）

说明：

这个指令用于锁定控制器上存储的用户程序，使其他用户不能查看程序的内容。程序被锁定后，依然可以运行，可以重新下载，但是当用户执行 ED、LS、TR、UL 指令时，控制器会拒绝执行。

程序锁定与解锁的状态会被 BN 指令保存。执行主复位后，控制器处于解锁状态。

参数：<control>L<control>K password, n

password	密码，由 PW 指令设置
n=1	锁定程序
n=0	解锁程序

应用：

运动中	是
程序中	是
指令行	是

默认值：

默认值	—
预设格式	—

相关指令：

PW	密码
ED	编辑
LS	列出程序
TR	跟踪运行
UL	上载程序

举例：

:PWtest, test	设置密码
:^L^Ktest, 1	锁定程序
:^L^Ktest, 0	解锁程序

LL

功能：列出标号（List Lables）

说明：

LL 指令会返回控制器中用户程序所使用的全部标号以及他们所在的行号。这些标号按照字母顺序排列。

参数：无

应用：

运动中	是
程序中	是
指令行	是

默认值：

默认值	—
预设格式	—

相关指令：

LA	列出数组
LS	列出程序
LV	列出变数

举例：

：LL

#FIVE=5

#FOUR=4

#ONE=1

#THREE=3

#TWO=2

LM

功能：直线插补模式（Linear Interpolation Mode）

说明：

LM 指令开始直线插补模式，并指定哪些轴参与直线插补。参与直线插补的轴可以有 1、2、3 或 4 个。用 LI 指令定义具体的插补路径，LE 指令结束插补路径。

在直线插补模式，电机按照给定的向量速度运动，控制器自动计算每段路径中各轴电机的速度。

参数：LM nnnn

n 为 A、B、C、D 或这些轴的任意组合。

n=? 返回插补路径缓冲区中可用空间，31 表示全空，0 表示已满。

应用：

运动中	否
程序中	是
指令行	是

默认值：

默认值	—
预设格式	—

操作数应用：

_LM 表示插补路径缓冲区中可用空间，31 表示全空，0 表示已满。

相关指令：

LE	插补路径结束
LI	插补路径段
VA	向量加速度
VD	向量减速度
CS	清除插补路径缓冲区

举例：

LM ABCD	A、B、C、D 四轴直线
VS10000	设置向量速度
VA100000;VD100000	设置向量加速度、向量减速度
LI 200, 300, 400, 500	直线插补路径
LE;BGS	结束路径，开始运动

_LR

功能：后极限信号状态（Reverse Limit Switch）

说明：

操作数_LRx 表示 x 轴后极限信号是否有效，值为 0 或 1。

1 当前 x 轴后极限信号无效

0 当前 x 轴后极限信号有效

极限信号在什么状态有效，由 CN 指令设置。在使用 LD 指令设置极限信号无效后，操作数依然为可以反映出相应极限信号的状态。

相关指令：

CN	设置

举例：

MG _LRA	输出当前 A 轴后极限信号状态

LS

功能：列出程序（List Program）

说明：

LS 指令返回控制器中的程序列表。

参数：LS m, n

m 与 n 是 0~449 直线的数字或程序中使用的标号。m 是需要列出的第一行，n是需要列出的最后一行。如 n 表示的程序行在 m 之前，则只列出一行。

应用：

运动中	是
程序中	否
指令行	是

默认值：

默认值	0, 最后一行
预设格式	—

相关指令：

LA	列出数组
LL	列出标号
LV	列出变数
<control>L<control>K	锁定程序

举例：

:LS #A, 6　　从标号#A 开始列出程序，至行号为 6 的程序行

2 #A
3 PR 500
4 BGA
5 AM
6 WT 200

LV

功能：列出变量（List Variables）

说明：

LV 指令返回控制器中所有用户变量，包括变量名和变量的值。变量按照字母顺序排列。

参数：无

应用：

运动中	是
程序中	是
指令行	是

默认值：

默认值	—
默认格式	变量格式

相关指令：

LA	列出数组
LS	列出程序
LL	列出标号

举例：

```
:LV
APPLE= 60.0000
BOY= 25.0000
ZEBRA= 37.0000
```

LZ

功能：前导 0（Leading Zeros）

说明：

LZ 指令用于设置控制器返回数据的格式。控制器返回的数字有自己的整数字元数与小数字数限制，当数值的实际位数少于设置的整数字元数时，LZ 指令确定控制器是否用 0 填充多出的整数字元。

参数：LZ n

n=1	取消前导 0
n=0	用前导 0 填充多出的整数位元
n=?	返回当前设置值

应用：

运动中	是
程序中	是
指令行	是

默认值：

默认值	1
预设格式	1.0

操作数应用：

_LZ	当前设置值。

举例：

```
:LZ 0
:TPA
 0000021645
:VAR1=
 0000000010.0000
:LZ 1
:TPA
 21645
:VAR1=
 10.0000
```

MB

功能：Modbus

说明：

MB 指令用于控制器主动与 IO 设备使用 MODBUS/TCP 协议通讯。

指令的格式取决于每个功能代码。如果功能代码为-1，表示使用第一层 MODBUS（生成原始数据报并接收原始数据）。二级通讯支持 10 个主要的功能代码。

功能代码	定义
01	读输出点状态（读多位）
02	读输入点状态（读多位）
03	读保持寄存器值（读多字）
04	读输入寄存器值（读多字）
05	置一个输出点状态（写单位）
06	预设一个寄存器值（写单字）
07	读错误代码
15	置多个输出点状态（写多位）
16	预设多个寄存器值（写多字）
17	报告设备 ID

参数：在所有的格式中都有 h 参数，h 表示连接的句柄（A、B、C 或 D）。参数 addr 为从设备地址，这个地址默认使用通讯句柄号（1~4）。

MBh=-1, len, array[]

len 为字节数

array[]为包含数据的数组名称

MBh=addr, 1, m, n, array[]

m 为起始位地址

n 为位数量

array[]数组中从第一个元素开始为返回值，每个元素包含 8 位数据

MBh=addr, 2, m, n, array[]

m 为起始位地址

n 为位数量

array[]数组中从第一个元素开始为返回值，每个元素包含 8 位数据

MBh=addr, 3, m, n, array[]

m 为起始寄存器地址

n 为寄存器数量

array[]数组用来保存返回值

MBh=addr, 4, m, n, array[]

m 为起始寄存器地址

n 为寄存器数量

array[]数组用来保存返回值

MBh=addr, 5, m, n

m 为位地址

n 为设置值，0 或 1

MBh=addr, 6, m, n

m 为寄存器地址

n 为 16bit 设置值

MBh=addr, 7, array[]

array[]数组用来保存返回值，每个元素一个字节

MBh=addr, 15, m, n, array[]

m 为起始位地址

n 为位数量

array[]数组中为设置值，每个元素包含 8 位数据

MBh=addr, 16, m, n, array[]

m 为起始寄存器地址

n 为寄存器数量

array[]数组中为设置值

MBh=addr, 17, array[]

array[]数组用来保存返回值

应用： 运动中 是

程序中 是

指令行 是

默认值： 默认值 —

预设格式 —

注意：使用 MODBUS/TCP 通讯句柄，必须使用端口 502。

MC

功能：运动完成（Motion Complete）

说明：

MC 指令是事件触发指令。这个指令会暂时停止程序的执行，直到指定轴的运动实际结束（脉冲发送完成）。MC 指令可以指定一个或多个轴。

参数：MC xxxx

x 为 A、B、C、D 或这些轴的任意组合，省略所有参数则表示同时使用所有参数。

应用：

运动中	是
程序中	是
指令行	否

默认值：

默认值	—
预设格式	—

相关指令：

BG	开始运动
AM	等待运动结束

举例：

#MOVE	程序标号
PR 2000, 4000	设置 A、B 轴运动距离
BG AB	还是运动
MC AB	等待运动完成
MG “DONE”; TD	输出信息
EN	程序结束

MF

功能：正向运动到位（Forward Motion to Position）

说明：

MF 指令是事件触发指令。这个指令会暂时停止程序的执行，电机位置（输出脉冲位置）大于或等于指定位置时，程序继续运行。MF 指令每次只能指定一个轴的位置。

参数：MF n, n, n, n　　或　MFx=n

n 为-2147483648 与 2147483647 之间的整数。

应用：

运动中	是
程序中	是
指令行	否

默认值：

默认值	—
预设格式	—

相关指令：

AP	绝对位置到达

举例：

#TEST	程序标号
DP0	当前位置清零
JG 1000	点动
BGA	开始运动
MF 2000	等待位置到达 2000
V1=_TDA	读当前位置
MG “Position is “, V1	输出信息
ST	停止运动
EN	程序结束

提示：MF 指令在时间上的精度为控制器工作周期的两倍，因此运动速度乘以控制器工作周期的 2 倍，就是指令的最大滞后距离。

MG

功能：输出信息（Message）

说明：

MG 指令用于从控制器的通讯接口输出信息。

参数：MG “m”, {^n}, V{Fm.n 或$m.n}{N}{Ex}{P1}

“m”为文本信息，可以包含字母、数字、符号或<control>G（最多 72 个字符）。

{^n}表示 ASCII 代码为 n 的字符。

V 为变量或数组元素，之后可以跟随如下格式说明：

{Fm.n}：十进制数值格式，整数部分输出 m 位，小数部分输出 n 位

{Zm.n}：与{Fm.n}相同，但不输出前导 0

{$m.n}：十六进制数值格式，整数部分输出 m 位，小数部分输出 n 位

{Sn}：字符串格式，长度为 n，n 的范围 1~6

{N}表示输出信息后不附加回车符号

{Ex}表示从以太网句柄 x 输出，x 为 A、B、C 或 D

{P1}表示串口输出

注意：有多文本、变量、字符输出时，彼此间用逗号分开。{N}、{Ex}、{P1}为可选项目，且没有次序要求。

应用：

运动中	是
程序中	是
指令行	是

默认值：

默认值	—
默认格式	变数格式

举例：

1、输出 ASCII 文本信息

MG“Good Morning”

2、输出变量或数组元素的数值

MG “The Answer is ”, Total {F4.2}

3、向端口发送 ASII 字符

MG {^13}, {^10}, {^48}, {^55}

MO

功能：电机关闭（Motor Off）

说明：

MO 指令使控制器关闭电机，此时控制器不能控制电机运动。打开电机使用指令 SH。

参数：MO xxxx

x 为 A、B、C、D 或这些轴的任意组合，省略所有参数则表示同时使用所有参数。

应用：

运动中	否
程序中	是
指令行	是

默认值：

默认值	0
预设格式	1.0

操作数应用：

_MOx	x 轴当前电机状态：0，电机打开；1：电机关闭。

相关指令：

SH	电机打开

举例：

MO	关闭所有电机
MOA	关闭 A 轴电机
MOB	关闭 B 轴电机
MOCA	关闭 A、C 轴电机
SH	打开所有电机
Bob=_MOB	将 B 轴状态赋值给变量 Bob
Bob=	输出变数 Bob 的值

MR

功能：反向运动到位（Reverse Motion to Position）

说明：

MR 指令是事件触发指令。这个指令会暂时停止程序的执行，电机位置（输出脉冲位置）小于或等于指定位置时，程序继续运行。MR 指令每次只能指定一个轴的位置。

参数：MR n, n, n, n　　或　MFx=n

n 为-2147483648 与 2147483647 之间的整数。

应用：

运动中	是
程序中	是
指令行	否

默认值：

默认值	—
预设格式	—

相关指令：

AP	绝对位置到达

举例：

#TEST	程序标号
DP0	当前位置清零
JG -1000	点动
BGA	开始运动
MR -2000	等待位置到达-2000
V1=_TDA	读当前位置
MG “Position is “, V1	输出信息
ST	停止运动
EN	程序结束

提示：MR 指令在时间上的精度为控制器工作周期的两倍，因此运动速度乘以控制器工作周期的 2 倍，就是指令的最大滞后距离。

MT

功能：电机信号类型（Motor Type）

说明：

MT 指令用于调整控制器输出的脉冲/方向信号，对于 DMC-B140（-M）控制器而言，这些调整可以通过 MT 指令实现，也可以通过调整控制器与驱动器之间的接线方式实现。

参数：MT n, n, n, n　　或　MTx=n

n 为+/-2 或+/-2.5。

n=?　　返回当前设置值

n 的值对脉冲方向信号的影响如下表（H 表示高电位；L 表示低电位

	脉冲信号（电机静止）		方向信号（正向运动）	
	STEP+	STEP-	DIR+	DIR-
2	H	L	H	L
-2	L	H	L	H
2.5	H	L	L	H
-2.5	L	H	H	L

在 2 与 2.5 之间或-2 与-2.5 之间改变MT 设置值时，方向信号的输出状态会在下一次运动开始时变化。

MT 指令的设置值对 CE 指令的返回值会产生影响。

应用：

运动中	否
程序中	是
指令行	是

默认值：

默认值	2, 2, 2, 2
预设格式	1.1

操作数应用：

_MTx	x 轴当前设置值。

相关指令：

CE	设置编码器

举例：

MT 2, 2, 2, 2	设置电机信号类型
MT ?, ?	返回电机信号类型
V=_MTA	将电机信号类型赋值给变量

MW

功能：Modbus 通讯等待（Modbus Wait）

说明：

打开 Modbus 通讯等待功能，控制器在执行了 Modbus 通讯指令后，会暂时停止程序的运行，直至接收到响应的信息。如长时间接收不到响应信息，会产生代码为 123 的错误。如程序中有#TCPERR 子程序也会自动运行。

参数：MW n

n=0	关闭 Modbus 通讯等待功能
n=1	打开 Modbus 通讯等待功能
n=?	返回当前设置的值

应用：

运动中	是
程序中	是
指令行	是

默认值：

默认值	1
预设格式	1.0

操作数应用：

_MW	表示返回的功能代码
_MW1	表示返回的错误代码

相关指令：

MB	Modbus

举例：

```
MW1
```

提示：执行MW0 指令后，通讯可能会变得不稳定，得不到预期的结果。

NO

功能：无动作（也可以使用单引号 ’）（No Operation）

说明：

NO 或单引号 ’ 指令没有任何作用，但可以用于注释程序。

参数：NO m 或 ‘m

m 为一组长度不超过 32 的字符。

应用：

运动中	是
程序中	是
指令行	是

默认值：

默认值	—
预设格式	—

举例：

```
#A
NO
NO This Program
NO Does Absolutely
NO Nothing
EN
```

OB

功能：输出位（Output Bit）

说明：

OB 指令用于设置通用输出状态。

参数：OB n, expression

n 为通用输出号，1、2、3、或 4。

Expression 可以是逻辑表达式或算术表达式。如表达式为假或计算结果为大于等于 0 而小于 1，输出截止，否则输出导通。

应用：

运动中	是
程序中	是
指令行	是

相关指令：

SB	设置输出位
CB	清除输出位
OP	输出

举例：

OB 1, POS=1	如变数 POS 值为 1，则通用输出 1 导通，否则通用输出 1 截止
OB 1, @IN[1]&@IN[2]	如通用输入 1、2 均导通，则通用输出 1 导通，否则通用输出 1 截止
OB 3, COUNT[1]	如数组元素 COUNT[1]值为小于 1 且大于等于 0，则通用输出 1 截止，否则通用输出 1 导通
OB N, COUNT[1]	如数组元素 COUNT[1]值为小于 1 且大于等于 0，则通用输出 N 截止，否则通用输出 N 导通

OP

功能：输出（Output Port）

说明：

OP 指令设置控制器的通用输出状态。

参数：OP m

m 为 0~15 的整数。m 的值为 4 位二进制数字，每一位的值依次对应一路通用输出的状态。最低位对应通用输出 1，最高为对应通用输出 4。数值为 1 表示该通用输出导通，数值为 0 表示该路通用输出截止。

应用：

运动中	是
程序中	是
指令行	是

默认值：

默认值	0
预设格式	7.0

操作数应用：

_OP 表示当前所有通用输出状态，0~15。

相关指令：

CB	清除输出位
SB	设置输出位

举例：

OP 11	设置通用输出 4、2、1 导通，通用输出 3 截止
MG _OP	输出当前通用输出状态

OE

功能：错误时自动关闭电机（Off on Error）

说明：

OE 指令设置在电机运动出现错误时自动关闭。这个指令的作用体现在当出现异常情况时降低系统的危险性。

参数：OE n, n, n, n　　或　OEx=n

n=0　取消自动关闭功能

n=1　当电机急停（AB）时自动关闭电机

n=2　当电机运动到位置极限时自动关闭电机

n=3　当电机运动到位置极限或急停（AB）时关闭电机

n=?　返回当前设置值

应用：

运动中	是
程序中	是
指令行	是

默认值：

默认值	0
预设格式	1.0

操作数应用：

_OEx	当前 x 轴设置值。

相关指令：

AB	急停
SH	电机打开
#POSERR	位置错误处理子程序
#LIMSWI	位置极限处理子程序

举例：

OE1, 1, 1, 1	设置四轴及 2 时关闭电机
OE?, ?	返回当前设置值

@OUT[n]

功能：读数字输出（Output）

说明：

返回指定通用数字输出信号的值。输出信号导通返回 1；输出信号截止返回 0。

参数：@OUT[n]

n 为 1、2、3、或 4。

应用：

运动中	是
程序中	是
指令行	是

默认值：

默认值	—
预设格式	—

相关指令：

@IN[n]	读数字输入
SB	设置输出位
CB	清除输出位

举例：

```
MG @OUT[1]
x=@OUT[2]
```

P1CD

功能：串口代码

说明：

P1CD 为表示当前串口接收状态的操作数，其值的含义如下：

0　未接收到新信息

1　接收到字符，但不是<enter>

2　接收到字符串，该字符串不能理解为数值

3　接收到数值

当接收到的信息被读出（P1CH、P1NM、P1ST）后，P1CD 的值恢复为 0。

应用：

运动中	是
程序中	是
指令行	是

默认值：

默认值	—
预设格式	—

相关指令：

P1CH	串口字符
P1NM	串口数字
P1ST	串口字符串
CI	设置通讯中断
#COMINT	通讯中断子程序

P1CH

功能：串口字符

说明：

P1CH 为串口操作数，表示最新接收到的字符。

应用：

运动中	是
程序中	是
指令行	是

默认值：

默认值	—
预设格式	—

相关指令：

P1CD	串口代码
P1NM	串口数字
P1ST	串口字符串
CI	设置通讯中断
#COMINT	通讯中断子程序

P1NM

功能：串口数值

说明：

P1NM 为串口操作数，表示最新接收到的数字的值。发送给串口的数字必须以 ASCII 代码方式发送，并以回车符结束。数值的范围必须在-2147483648~2147483647。

应用：	运动中	是
	程序中	是
	指令行	是
默认值：	默认值	—
	预设格式	—

相关指令：

P1CD	串口代码
P1CH	串口字符串
P1ST	串口字符串
CI	设置通讯中断
#COMINT	通讯中断子程序

P1ST

功能：串口字符串

说明：

P1NM 为串口操作数，表示最新接收到的字符串。发送给串口的字符串后面必须用回车符表示结束，**并且长度不能超过 6 个字符**（不含回车符）。

应用：	运动中	是
	程序中	是
	指令行	是
默认值：	默认值	—
	预设格式	—

相关指令：

P1CD	串口代码
P1CH	串口字符
P1NM	串口数字
CI	设置通讯中断
#COMINT	通讯中断子程序

PA

功能：绝对位置（Position Absolute）

说明：

PA 指令设置单轴运动的目标位置。这个位置值是相对于绝对 0 点的。

参数：PA n, n, n, n　　或　PAx=n

n 为带符号整数，范围-2147483648~2147483647，单位为脉冲。

n=?　返回该轴电机最后停止的位置。

应用：

运动中	否
程序中	是
指令行	是

默认值：

默认值	—
默认格式	位置格式

操作数应用：

_PAx	该轴电机最后停止的位置。

相关指令：

PR	相对位置
SP	速度
AC	加速度
DC	减速度
BG	开始运动

举例：

PA 400, -600, 500, 200	设置运动目标位置
BG; AM	开始运动并等待运动结束
PA ?, ?, ?, ?	运动结束后查询当前规划位置
400, -600, 500, 200	
PA700	设置 A 轴目标位置
BG	开始运动

PF

功能：位置格式（Position Format）

说明：

PF 指令用于设置控制器回馈位置数据时的格式。PF 指令可以设置这些数据以 10 进制或 16 进制输出，以及输出时包含的整数和小数字数。

如果实际整数字元数大于输出设置值，则输出指定位数中的最大值（如 999.99，$7FF.FF）。输出格式由 PF 指令设置的查询指令有：

BL? DE? DP? EM? FL? IP? LE? PA?

PR? VE? RP TD TP

参数：PF m.n

m 为-8~10 之间的整数，表示数值输出的整数字元数，如 m 为负，则表示以 16 进制形式输出。

n 为 0~4 之间的整数，表示数值输出的小数字数。

PF? 以 m.n 的形式返回当前设置值。

应用：

运动中	是
程序中	是
指令行	是

默认值：

默认值	10.0
预设格式	2.1

操作数应用：

_PF	当前位置格式设置值。

相关指令：

VF	变数格式
LZ	前导 0

PR

功能：相对位置（Position Relative）

说明：

PR 指令设置下一次单轴定位运动的方向和运动距离。运动的目标位置是相对于开始运动前的位置确定的。

参数：PR n, n, n, n　　或　PRx=n

n 为带符号整数，范围-2147483648~2147483647，单位为脉冲。

n=?　查询该轴电机当前（或下一次）运动的增量距离设置。

应用：

运动中	否
程序中	是
指令行	是

默认值：

默认值	0
默认格式	位置格式

操作数应用：

_PRx	n 轴电机的当前（或下一次）运动的增量距离设置。

相关指令：

PA	绝对位置
SP	速度
AC	加速度
DC	减速度
IP	增量位置
BG	开始运动

举例：

PR 100, 200, 300, 400	设置运动距离与方向
BG	开始运动并等待运动结束
PR ?, ?, ?	查询相对距离
100, 200, 300	
PR500	改变 A 轴移动距离设置
BG	A 轴开始运动 500，B、C、D 轴按照之前的设置分别运动 200、300、400。

PT

功能：位置跟随（Position Tracking）

说明：

PT 指令使控制器工作在位置跟随模式下。这个模式下，在电机运动中也可以执行 PA 指令。运动的速度、目标位置都可以在运动中任意调整。

当电机处于位置跟随模式，无论电机是运动还是静止，对控制器而言，电机始终处于运动状态。

对处于位置跟随模式下的电机使用 ST、PR 等指令，会使电机退出位置跟随模式。

参数：PT n, n, n, n　　或　PTx=n

n=1	为使电机处于位置跟随模式
n=0	为使电机退出位置跟随模式
n=?	查询该轴当前设置值

应用：

运动中	是
程序中	是
指令行	是

默认值：

默认值	—
预设格式	—

相关指令：

AC	加速度
DC	减速度
PA	绝对位置
SP	速度

举例：

```
#A                  程序标号
PT 1, 1, 1, 1       四轴电机均处于位置跟随模式
#LOOP               建立循环
PA V1, V2, V3, V4   又变量指定电机位置
WT10                等待 10 毫秒
JP#LOOP
EN
```

特别提示：由于电机处于位置跟随模式时，无论是运动还是静止，对控制器始终认为电机处于运动状态，所以 AM、MC 指令无效。

PW

功能：密码（Password）

说明：

PW 指令的作用是设置密码，这个密码用于锁定控制器中存储的用户程序。密码最多包含 8 个字母或数字。控制器执行主复位后，默认密码为空字符串（Null）。

密码会被 BN 指令保存，但无法查询。当控制器上存储的用户程序被锁定时，不允许重新设置密码。

参数：PW n, n

n 为最长 8 个字符的字符串。

应用：

运动中	是
程序中	否
指令行	是

默认值：

默认值	（NULL）
预设格式	—

相关指令：

<control>L<control>K	锁定程序

QD

功能：下载数组

说明：

控制器执行 QD 指令后，就准备接收一组数据。这组数据由 \ 结束，数据之间用逗号或者<CR><LF>分开。这些数据将依次赋值给指定数组中指定范围的元素。

参数：QD array[], start, end

array[]	有效的数组名称。
start	被赋值的第一个元素的下标，如省略则默认为 0
end	被赋值的最后一个元素的下标，如省略则默认为最后一个元素

应用：

运动中	是
程序中	否
指令行	是

默认值：

默认值	start=0, end=size-1
预设格式	—

相关指令：

QU	上载数组

QR

功能：数据记录

说明：

QR 指令会返回一个包含控制器状态信息的记录。这个记录中包括 4 字节的头信息和指定 I/O 区块或轴区块。

参数：QR nnnnnn

n 为 A、B、C、D、S、I 或这些字符的任意组合。

应用：

运动中	是
程序中	是
指令行	是

默认值：

默认值	—
预设格式	—

注意：由于控制器的状态信息是以二进制的形式输出，Galil 通讯软件中不能直接显示其内容。

QU

功能：上载数组

说明：

QU 指令使控制器依次输出指定数组中指定范围元素的值。输出的数据以<control>Z 作为结束标记。

参数：QU array[], start, end, delim

array[]	有效的数组名称。
start	被输出值的第一个元素的下标，如省略则默认为 0
end	被输出值的最后一个元素的下标，如省略则默认为最后一个元素
delim=1	各数值之间以逗号分开
delim=0	各数值之间以回车分开

应用：

运动中	是
程序中	是
指令行	是

默认值：

默认值	—
预设格式	—

相关指令：

QD	下载数组

QZ

功能：数据记录格式

说明：

QZ 指令返回控制器的状态数据记录的信息。这个指令返回四个数字，用逗号分割。第一个数字是控制器的轴数。第二个数字是控制器基本状态数据的字节数。第三个数字是插补运动状态数据的字节数。最后一个数字是每轴状态数据的字节数。

参数：QZ

应用：

运动中	是
程序中	是
指令行	是

默认值：

默认值	—
预设格式	—

相关指令：

DR	数据记录的更新速率

举例：

:QZ

4, 6, 16, 24

RA

功能：记录数组（Record Array）

说明：

RA 指令设置自动采集数据中记录数据的数组。被设置的数组必须是已经存在的。采集的数据内容与时间间隔分别由指令 RD 与 RC 设置。

参数：RA m[], n[], o[], p[]

m、n、o、p 为定义好的数组名，[]内为空。

应用：

运动中	是
程序中	是
指令行	是

默认值：

默认值	—
预设格式	—

相关指令：

DM	定义数组
RC	记录时间间隔
RD	记录数据内容

举例：

#Record	程序标号
DM POS[100]	定义用于记录数据的数组
RA POS[]	指定用 POS 数组记录采集到的数据
RD _TDA	自动采集 A 轴的发送脉冲数
RC 1	数据采集间隔为 2 个工作周期，开始采集数据
IP 1000	A 轴向前运动 1000
EN	程序结束

RC

功能：记录（Record）

说明：

RC 指令用于开始自动采集数据，同时设置采集数据的时间间隔以及采集次数。RC 0 指令停止数据采集

参数：RC n, m

n=0　　停止数据采集、记录

n=1~8　开始自动数据采集，时间间隔为 2^n 控制器工作周期。

n=?　　查询自动数据采集工作状态：

1：　目前在执行自动数据采集

0：　目前没有在执行自动数据采集

m 为可选参数，数据采集次数。如省略 m，则以记录数组中空间最小的数组可记录的数据量决定采集次数。如 m 为负，则使用数组中 0 至 m-1 个元素循环记录数据，无限次采集。

应用：

运动中	是
程序中	是
指令行	是

默认值：

默认值	—
预设格式	—

操作数应用：

_RC	表示自动数据采集工作状态：1，在工作；0，未工作。

相关指令：

DM	定义数组
RA	记录数组
RD	记录数据内容

举例：

```
#Record                 程序标号
DM POS[800]             定义用于记录数据的数组
RA POS[]                指定用 POS 数组记录采集到的数据
RD _TDA                 自动采集 A 轴的发送脉冲数
RC 2                    数据采集间隔为 4 个工作周期，开始采集数据
JG 1000;BG              A 轴点动向前
#A;JP#A, _RC=1          等待自动数据采集完成
MG “DONE REC”           输出信息
EN                      程序结束
```

RD

功能：记录数据（Record Data）

说明：

RD 指令用于说明自动数据记录功能记录哪些数据。可以同时记录 1 至 4 组数据（必须与 RA 指令一致）。可以被自动记录的数据有：

数据符号	说明
_SHx	规划位置
_RPx	规划位置
_DEx	发送脉冲位置
_TDx	发送脉冲位置
_TPx	编码器回馈位置
_TSx	开关点
_RLx	位置锁存的位置
_SCx	停止代码
_TI	通用输入状态
_OP	通用输出状态

x 为轴名称，A、B、C 或 D。

参数：RD m1, m2, m3, m4

m1、m2、m3、m4 上表中所列出的数据符号。

应用：

运动中	是
程序中	是
指令行	是

默认值：

默认值	—
预设格式	—

操作数应用：

_RD	下记录一次采集到的数据的数组元素下标。

相关指令：

DM	定义数组
RA	记录数组
RC	记录时间间隔

举例：

DM POS1[400], POS2[400]	定义数组 POS1[]与 POS2[]
RA POS1[], POS2[]	指定用于记录的数组 POS1[]与 POS2[]
RD _RPA, _TDA	自动采集 A 轴的规划位置与发送脉冲位置
RC1	开始以每 2 个工作周期一次的频率采集数据
JG10000;BGA	A 轴开始点动

RE

功能：自错误处理子程序返回（Return from Error Routine）

说明：

RE 指令用于结束#LIMSWI 或#TCPERR 子程序。如之前 0 线程中有程序被自动运行子程序暂停运行，则该程序在暂停处恢复运行。如该程序是暂停在条件触发指令处，RE 指令可设置该出发条件是否有效。

参数：RE n

n=0　取消原程序中对触发条件的等待，直接执行下一指令

n=1　恢复原程序中对触发条件的等待

省略参数，则默认参数为 0。

应用：

运动中	是
程序中	是
指令行	否

默认值：

默认值	0
预设格式	—

相关指令：

#LIMSWI	位置极限处理子程序
#TCPERR	以太网通讯错误处理子程序

举例：

```
#Main                    程序标号
FL 10000                 设置 A 轴正向软件极限
JG 1000;BGA              A 轴点动
#LOOP                    循环
MG "0";WT1000            每秒输出一次信息
JP#LOOP
#LIMSWI                  位置极限子程序
MG "1"                   输出信息
AM A                     等待 A 轴减速静止
RE 0                     从位置极限子程序返回
```

RI

功能：自输入中断程序返回（Return from Error Input Interrupt）

说明：

RI 指令用于结束#ININT 开头的输入中断子程序。子程序最后的 RI 指令使程序返回主程序。如果输入中断发生时，主程序正在执行某个条件触发指令，等待触发条件，那么 RI 指令的参数可以设置返回主程序后是继续等待该条件还是不再等待，直接执行后续指令。

如果不需要返回主程序继续运行，则可用 ZS 指令清除堆栈，然后用 JP 指令跳转到程序中的任意位置。

参数：RI n

n=0　取消原程序中对触发条件的等待，直接执行下一指令

n=1　恢复原程序中对触发条件的等待

省略参数，则默认参数为 0。

应用：

运动中	是
程序中	是
指令行	否

默认值：

默认值	0
预设格式	—

相关指令：

#ININT	输入中断处理子程序
II	输入中断

举例：

```
#TEST                程序标号
II 1                 设置输入中断
#LOOP                循环
AI 2;MG 0            等待通用输入 2 截止，输出 0
AI -2;MG 1           等待通用输入 2 导通，输出 1
JP#LOOP              循环
EN                   程序结束
#ININT               输入中断子程序
MG"IN"               输出信息
AI 1
RI 0
```

RL

功能：查询位置锁存值（Report Latched Position）

说明：

RL 指令查询位置锁存的位置值。位置锁存功能需要先由 AL 指令设置，而后由特定的外部输入信号触发。

参数：RL xxxx

x 为 A、B、C、D 或这些轴的任意组合。如省略参数，则表示同时使用所有参数。

应用：

运动中	是
程序中	是
指令行	是

默认值：

默认值	0
默认格式	位置格式

操作数应用：

_RLx	x 轴锁存的位置值。

相关指令：

AL	设置位置锁存

举例：

JG, 5000	B 轴点动
BG B	开始
AL TB	锁存 B 轴输入编码器 Z 所在位置
RLB	查询缩存结果

如 B 轴未连接编码器信号，则输出结果为 0

@RND[n]

功能：取整（Round）

说明：

返回与给定数据最接近的整数值。

参数：@RND[n]

n 为-2147483648~2147483647 之间的数。

应用：

运动中	是
程序中	是
指令行	是

默认值：

默认值	—
预设格式	—

相关指令：

@INT[n]	取整数部分

举例：

```
:MG @ RND [1.2]
 1.0000
:MG @ RND [5.7]
 6.0000
:MG @ RND [-1.2]
 -1.0000
:MG @ RND [-5.7]
 -6.0000
:MG @ RND [5.5]
 6.0000
:MG @ RND [-5.5]
 -5.0000
```

RP

功能：查询规划位置（Reference Position）

说明：

这个指令返回电机当前规划位置。

参数：RP xxxxx

x 为 A、B、C、D、N 或这些轴的组合。省略参数则表示同时使用 ABCD。

应用：

运动中	是
程序中	是
指令行	是

默认值：

默认值	0
默认格式	位置格式

操作数应用：

_RPx	n 轴当前的电机规划位置。

相关指令：

TD	查询输出脉冲位置
TP	查询编码器回馈位置

RS

功能：复位（Reset）

说明：

RS 指令使控制器恢复到上电状态，恢复之前保存在控制器闪存中的的参数、变量、数组、以及用户程序。

RS-1 使控制期实现“软主复位”，使当前的参数恢复为出厂值，但不影响保存在控制器闪存中的内容。

应用：

运动中	是
程序中	否
指令行	是

默认值：

默认值	0
预设格式	3.0

操作数应用：

_RS　　表示控制器自检的结果。如自检正常，值为 0，如自检出错，则相应位为 1：

Bit3	主复位错误
Bit2	程序校验错误
Bit1	参数校验错误
Bit0	变量校验错误

<control>R<control>S

功能：主复位

说明：

此指令将控制器内部恢复到出厂状态，内部闪存清空。

主复位也可以通过跳线与复位指令配合实现，当控制器上 MRST 跳线短接时，执行的复位会自动变成主复位动作。

参数：无

应用：	运动中	是
	程序中	否
	指令行	是
默认值：	默认值	—
	预设格式	—

<control>R<control>V

功能：固件信息

说明：

此指令使控制器输出内部固件的版本信息。同样的控制器硬件，不同的固件可以为用户提供不同的指令、参数和功能。

参数：无

应用：

运动中	是
程序中	否
指令行	是

SB

功能：设置输出位（Set Bit）

说明：

SB 指令使控制器的通用输出导通。

参数：SB n

n 为 1、2、3 或 4。

应用：

运动中	是
程序中	是
指令行	是

默认值：

默认值	—
预设格式	—

相关指令：

CB	清除输出位

举例：

SB4	使通用输出 4 导通
SB1	使通用输出 1 导通

SC

功能：停止代码（Stop Code

说明：

SC 是查询指令，返回电机的停止代码。电机的停止代码反应出电机停止的原因。不同的代码表示的含义如下表：（对于电子齿轮的从轴，在解除同步关系前，返回值不变）

代码	含义
0	独立运动中
1	完成独立运动，到达指定位置而停止
2	由于正向位置极限信号而停止
3	由于反向位置极限信号而停止
4	由于 ST 指令而停止
6	由于急停输入信号而停止
7	由于 AB 指令而停止
9	由于完成 FE 动作而停止
10	由于完成 HM 动作而停止
11	由于选定的单轴急停信号而停止
50	轮廓模式运动中
51	轮廓模式结束而停止
100	插补（LM 或 VM）运动中
101	插补运动结束而停止

参数：SC xxxx

x 为 A、B、C、D 或这些轴的任意组合，如省略参数，则表示使用全部参数。

应用：

运动中	是
程序中	是
指令行	是

默认值：

默认值	—
预设格式	3.0

操作数应用：

_SCx	x 轴当前的值。

举例：

TOM=_SCD　　将 D 轴的停止代码赋给变数 TOM

SD

功能：位置极限减速度（Switch Deceleration）

说明：

SD 指令设置在电机运动到位置极限时的减速度。

参数：SD n, n, n, n　　或　SDx=n

n 为无符号整数，范围 1024~1073740800，单位是脉冲每秒平方。输入的数据如不是 1024 的倍数，回自动变换为不大于输入数值的 1024 整数倍数。如输入的数据小于 1024，则自动变换为 1024。

n=?　返回该轴的设置值。

应用：

运动中	否
程序中	是
指令行	是

默认值：

默认值	256000
预设格式	10.0

操作数应用：

_SDx	x 轴当前设置值

相关指令：

AC	加速度
DC	减速度

举例：

PR 10000	设置运动距离
AC 2000000	设置加速度
DC 1000000	设置正常状态下的减速度，较小，使定位平稳
SD 5000000	设置极限位置的减速度，较大，使减速距离短
SP 5000	设置速度
BG A	开始运动

SH

功能：电机打开

说明：

SH 指令使控制器恢复对电机的控制器，同时改变使能信号的输出状态（如使用者设置了 LC 指令，即使执行了 SH 指令，使能信号输出依然保持在电机关闭的状态）。SH 指令可能会改变之前给予电机的运动状态设置（如运动模式、位置、速度等）。

参数：SH xxxx

x 为 A、B、C、D 或这些轴的任意组合，省略所有参数则表示同时使用全部参数。

应用：

运动中	否
程序中	是
指令行	是

默认值：

默认值	—
预设格式	—

相关指令：

MO	电机关闭

举例：

SH	打开全部轴的电机
SHA	打开 A 轴电机
SHAB	打开 A、B 两轴电机
SHCD	打开 C、D 两轴电机

@SIN[n]

功能：正弦（Sine）

说明：

返回给定角度数据的正弦值

参数：@SIN[n]

n 为带符号的数，单位为角度，范围-32768~32767，分辨率 1/65536。

应用：

运动中	是
程序中	是
指令行	是

默认值：

默认值	—
预设格式	—

相关指令：

@ACOS[n]	反余弦
@COS[n]	余弦
@ASIN[n]	反正弦
@ATAN[n]	反正切
@TAN[n]	正切

举例：

```
:MG @SIN[0]
 0.0000
:MG @SIN[90]
 1.0000
:MG @SIN[180]
 0.0000
:MG @SIN[270]
 -1.0000
:MG @SIN[360]
 0.0000
```

SL

功能：单步执行（Single）

说明：

SL 指令用于程序调试，在 BK 指令暂停程序运行后，只运行程序中的一行（或指定行数）的指令。用 BK 指令恢复程序的正常运行。

参数：SL n

n 为再次暂停前运行的程序行数，如省略，默认为 1。

应用：

运动中	是
程序中	否
指令行	是

默认值：

默认值	1
预设格式	—

相关指令：

BK	断点
TR	跟踪

举例：

BK3	运行到 3 行暂停（首行为 0 行）
BK5	运行到 5 行暂停（首行为 0 行）
SL	再运行一行暂停
SL3	再运行三行暂停
BK	运行至结束

SM

功能：子网掩码（Subnet Mask）

说明：

SM 指令用于设置控制器的子网掩码。控制器对来自其它子网 IP 地址的信息不予响应。关于子网掩码的更多知识，请阅读 TCP/IP 协议的相关资料。

参数：SM sm0, sm1, sm2, sm3　或　SM n

sm0, sm1, sm2, sm3 为逗号分开的单字节数（0~255），用来表示子网掩码。

n 为带符号 4 字节整数，用来表示子网掩码。

n=?　查询子网掩码设置值。

应用：

运动中	是
程序中	是
指令行	是

默认值：

默认值	0, 0, 0, 0
预设格式	—

操作数应用：

_SM0	子网掩码设置值。

相关指令：

IH	网络句柄
IA	IP 地址

举例：

SM 255, 255, 255, 255	忽略所有网络通讯信息
SM 0, 0, 0, 0	响应所有网络通讯信息

SP

功能：速度（Speed）

说明：

SP 指令由于设置各轴独立定位运动中的运动速度。在实际的运动中，如果运动距离过短，可能运动中的最大速度也没有达到 SP 指令设置的速度。

JG 指令会影响 SP 的设置值。

参数：SP n, n, n, n　　或　SPx=n

n 为 0~3000000 之间的无符号偶数，单位是脉冲/秒。

n=?　　返回当前设置值。

应用：

运动中	是
程序中	是
指令行	是

默认值：

默认值	25000
预设格式	5.0

操作数应用：

_SPx	x 轴当前设置值。

相关指令：

AC	加速度
DC	减速度
PA	绝对位置
PR	相对位置
IP	增量位置
BG	开始运动

举例：

PR 2000, 3000, 4000, 5000	设置四轴运动距离
SP 5000, 6000, 7000, 8000	设置四轴运动速度
BG	开始运动

@SQR[n]

功能：平方根（Square Root）

说明：

返回给定数值的平方根，如给定数值为负，则自动取其绝对值计算平方根。

参数：@SQR[n]

n 为-2147483648~2147483647 之间的带符号数。

应用：

运动中	是
程序中	是
指令行	是

默认值：

默认值	—
预设格式	—

相关指令：

@ABS[n]	绝对值

举例：

```
:MG @SQR[2]
 1.4142
:MG @SQR[-2]
 1.4142
```

ST

功能：停止（Stop）

说明：

ST 指令停止指定轴的运动.电机按照 DC 设置减速停止。如果 ST 指令是由指令行的方式执行而且没有指定任何轴，那么不但所有轴的运动会停止，控制器上的用户程序也会停止运行。

参数：ST xxxxxx

x 为 A、B、C、D、S、N 或这些轴的任意组合，省略全部参数则表示同时使用全部参数。

应用：	运动中	是
	程序中	是
	指令行	是
默认值：	默认值	—
	预设格式	—

相关指令：

BG	开始运动
DC	减速度

举例：

ST A	停止 A 轴运动
ST S	停止插补运动
ST ABCD	停止 A、B、C、D 轴运动
ST	停止 A、B、C、D 以及虚拟轴 N 的运动
ST SD	停止插补运动以及 D 轴运动

@TAN[n]

功能：正切（Tangent）

说明：

返回给定角度的正切值。

参数：@TAN[n]

n 为带符号的数，单位为角度，范围-32768~32767，分辨率 1/65536。

应用：

运动中	是
程序中	是
指令行	是

默认值：

默认值	—
预设格式	—

相关指令：

@ACOS[n]	反余弦
@COS[n]	余弦
@ASIN[n]	反正弦
@SIN[n]	正弦
@ATAN[n]	反正切

举例：

```
:MG @TAN[-90]
 -2147483647.0000
:MG @TAN[0]
 0.0000
:MG @TAN[90]
 2147483647.0000
```

TB

功能：查询状态字节（Tell Status Byte）

说明：

TB 指令用于查询控制器的状态字节。控制器处于以下状态时，对应位为 1，否则该位为 0。

BIT	状态
7	正在执行应用程序
6	
5	正在执行轮廓模式运动（包括暂停）
4	正在执行位置极限信号处理子程序
3	输入中断使能
2	正在执行输入中断处理子程序
1	
0	串口回应打开

参数：无

应用：

运动中	是
程序中	是
指令行	是

默认值：

默认值	—
预设格式	3.0

操作数应用：

_TB	当前状态字节的值。

举例：

M=_TB	将当前状态字节的值赋给变量
TB	查询当前值
0	输出 0

TC

功能：查询错误代码（Tell Error Code）

说明：

TC 指令会返回控制器的出错信息。当控制器中的程序异常中止或控制对输入的指令回馈 ？时，TC 指令的回馈信息是非常有价值的。错误信息输出后即被清除，所以对一次错误不能反复查询。（部分 Galil 的通讯软件会在指令行出现错误时自动发送一次 TC1 指令。）

以下是错误代码、错误指令清单的说明。

代码	信息	说明
0		无错误
1	Unrecognized command	不可识别的指令
2	Command only valid from program	指令只能在程序中使用
3	Command not valid in program	指令在程序中不可用
4	Operand error	参数错误
5	Input buffer full	输入缓冲区满
6	Number out of rang	参数超范围
7	Command not valid while running	指令在运动中不可执行
8	Command not valid when not running	指令在静止中不可执行
9	Variable error	变量错误
10	Undefined label or empty sequence line	不存在的标号或行号
11	Invalid label or line number	不存在的标号无或效的行号
12	Subroutine more than 16 deep	子程序嵌套超过 16 层
13	JG only valid when running in jog mode	只能在点动模式使用
14	EEPROM check sum error	内存检验错误
15	EEPROM write error	内存写入错误
16	IP wrong sign in pos move or IP during forced decel	IP 指令使用错误
17	ED and DL not valid when program running	程序运行中不能下载/编辑
18	Command not valid when contouring	轮廓模式下不可用
19	Application strand already executing	线程中有程序在运行
20	Begin not valid with motor off	电机关闭时不能开始运动
21	Begin not valid while running	电机运动中不能开始运动
22	Begin not possible due to limit switch	由于位置极限不能开始运动
24	Begin not valid because no sequence defined	未定义插补路径不能开始运动
25	Variable not given in IN command	IN 未使用变量
28	S operand not valid	S 轴不可用
29	Not valid during coordinated move	平面插补运动中无效
30	Sequence segments too short	路径段太短
31	Total move distance in sequence > 2billion	路径段太长
32	More than 31 segments in a sequence	缓冲区已满
33	VP or CR commands cannot be mixed with LI command	直线插补中不能使用 VP、CR 指令

41	Contouring record rang error	轮廓数据错误
42	Contour data being sent too slowly	轮廓数据提供太慢
46	Gear axis both master and follower	一个轴不能跟随自己
50	Not enough fields	没有足够的域
51	Question mark not valid	不支持查询
52	Missing “ or string too long	缺少引号
53	Error in {}	{}字符错误
54	Question mark part of string	
55	Missing [or []	缺少]
56	Array index invalid or out of range	数组下标超范围
57	Bad function or array	错误的函数或数组
58	Bad command rsponse	错误的指令响应
59	Mismatched parentheses	缺少括号
60	Download error – line too long or too many lines	下载程序错误
61	Duplicate or bad label	重复或错误的标号
62	Too many labels	标号太多
63	IF statement without ENDIF	IF 没有 ENDIF
65	IN command must have a comma	IN 必须有逗号
66	Array space full	数组空间已满
67	Too many arrays or variables	数组或变量太多
71	IN only valid in task #0	IN 只能使用在 0 线程
80	Record mode already running	记录已经在进行
81	No array or source specified	未指定数组
82	Undefined array	未定义数组
83	Not a valid number	无效数值
84	Too many elements	数组元素太多
90	Only A B C D valid operand	只有 A、B、C、D 有效
98	Binary Commands not valid in application program	程序中不能使用二进制指令
99	Bad binary command number	错误的二进制指令
100	Not valid when running ECAM	ECAM 运行中无效
101	Improper index into ET (must be 0-256)	ET 索引错误
102	No master axis defined for ECAM	未指定 ECAM 主轴
103	Master axis modulus greater than 256*EP value	主轴周期太长
104	Not valid when axis performing ECAM	轴工作在 ECAM 下无效
105	EB1 command must be given first	应首先执行 EB1
106	Privilege violation	
119	Not valid for axis configured as stepper	对于脉冲输出的轴无效
123	TCP lost sync or timeout	TCP 通讯错误
133	Command not valid when locked	锁定状态下指令无效
134	All motors must be in MO for this command	所有电机必须关闭
135	Motor must be in MO	电机必须关闭
136	Invalid Password	无效密码
137	Invalid lock setting	无效锁定设置

138	Password not identical	密码不一致

参数：TCn

n=0　返回错误代码

n=1　返回错误代码及说明

省略参数，则默认参数为 0。

应用：　运动中　是

程序中　是

指令行　是

默认值：默认值　—

预设格式　—

操作数应用：

_TC　错误代码

#TCPERR

功能：以太网通讯错误处理子程序

说明：

在运行控制器上的用户程序时，如出现以太网通讯错误（123 TCP lost sync or timeout），则自动在 0 线程运行以此标号开头的子程序。出现这样的错误，意味着通讯句柄没有响应 TCP 应答信号（例如，由于连接线断开）。这种情况下，出错的句柄被关闭。

#TCPERR 子程序使用户可以自己编写代码处理这种情况。

使用 RE 指令自子程序中返回。

应用： 运动中 是

程序中	是
指令行	否

相关指令：

TC	查询错误代码
_IA4	最近产生通讯错误的句柄号
MG	输出信息
SA	发送 ASCII 指令

举例：

```
#L
MG {EA} “L”
WT1000
JP#L
#TCPERR
MG {P1}”TCPERR”, _IA4
RE
```

TD

功能：查询输出脉冲位置

说明：

TD 指令用于查询控制器输出脉冲的计数值。

参数：TD xxxx

x 为 A、B、C、D 或这些轴的任意组合，如省略参数，则表示使用全部参数。

应用：

运动中	是
程序中	是
指令行	是

默认值：

默认值	—
默认格式	位置格式

操作数应用：

_TDx	x 轴当前输出脉冲计数值。

相关指令：

DP	定义位置

TH

功能：查询句柄（Tell Handle）

说明：

TH 指令用于查询控制器通讯句柄的状态。这个指令返回的信息包括控制器的 IP 地址、以太网物理地址，而后是每个句柄的连接类型、使用端口、对方 IP 地址、使用端口。

参数：无

应用：

运动中	是
程序中	是
指令行	是

默认值：

默认值	—
预设格式	—

相关指令：

IH	网络句柄
WH	当前句柄

举例：

```
TH
CONTROLLER IP ADDRESS 10, 51, 0, 87 ETHERNET ADDRESS 00-50-4C-08-01-1F
IHA TCP PORT 1050 TO IP ADDRESS 10, 51, 0, 89 PORT 1000
IHB TCP PORT 1061 TO IP ADDRESS 10, 51, 0, 89 PORT 1001
IHC TCP PORT 1012 TO IP ADDRESS 10, 51, 0, 93 PORT 1002
IHD TCP PORT 1023 TO IP ADDRESS 10, 51, 0, 93 PORT 1003
```

TI

功能：查询输入（Tell Input）

说明：

TI 指令用于查询通用数字输入的状态。返回值为 0~255 的整数。作为一个 8 位的二进制数值，每一位依次对应一路通用输入的状态，最低位对应通用输入 1，最高位对应同用输入 8。数值为 1，表示该通用输入为截止状态，数值为 0 表示该通用输入为导通状态。（控制器上的 ABORT 跳线可能改变通用输入 8 的状态对应关系）。

参数：无

应用：

运动中	是
程序中	是
指令行	是

默认值：

默认值	—
预设格式	3.0

操作数应用：

_TI	当前通用输入状态的值。

举例：

TI	查询状态
255	返回值

TIME

功能：时间操作数

说明：

TIME 操作数返回控制器复位（开机）后的工作周期数。

相关指令：

TM	工作周期

举例：

#A	标号
A= TIME	保存当前时刻
WT1000	等待
MG TIME-A	输出等待的时间（以控制器工作周期为单位）
EN	程序结束

TM

功能：工作周期

说明：

TM 指令用于设置控制器的工作周期。数值的时间单位为 1/1024 毫秒。如设置为 0 或负数，控制器将关闭内部时钟，这意味着大部分功能将不能工作。

参数：TM n

n 最小值为 250，最大为 20000，分辨为 125。

n=? 返回当前设置值。

应用：

运动中	是
程序中	是
指令行	是

默认值：

默认值	1000
预设格式	1.0

操作数应用：

_TM	当前设置值。

举例：

TM-1000	关闭内部时钟
TM250	将内部工作周期设置为最小值
TM1000	将内部工作周期设置为原默认值

注意：控制器虽然允许在运动中改变 TM 设置，但是建议用户不要在电机运动中将修改 TM 设置值，尤其不要在运动中将 TM 设置值改小，否则可能产生不可预期的错误。

TP

功能：查询编码器回馈位置（Tell Position）

说明：

TP 指令查询编码器回馈位置值。

参数：TP xxxx

x 为 A、B、C、D 或这些轴的任意组合，如省略参数，则表示使用全部参数。

应用：

运动中	是
程序中	是
指令行	是

默认值：

默认值	—
默认格式	位置格式

操作数应用：

_TPx	x 轴当前值。

相关指令：

TD	查询输出脉冲位置
TV	查询回馈速度

TR

功能：跟踪（Trace）

说明：

TR 指令用于程序的跟踪运行。在跟踪模式下，程序中每条指令在执行前都会从通讯接口发送出来。

参数：TR n, m

n=0　取消跟踪运行模式。

n=1　打开跟踪运行模式。

m 表示需要跟踪的线程，其值的范围是 0~15。该数值为 4 位二进制树，每一位依次代表一个线程，最低为代表 0 线程，最高位代表 3 线程。数值为 1 表示跟踪该线程。

应用：	运动中	是
	程序中	是
	指令行	是
默认值：	默认值	0, 15
	预设格式	—

TS

功能：查询轴状态（Tell Switch）

说明：

TS 指令返回表示轴状态的字节。字节中各位对应的状态如下：

位	状态
7	如果在运动为 1；否则为 0
6	未定义 0
5	电机关闭为 1；否则为 0
4	未定义
3	正向位置极限信号有效为 1；否则为 0
2	反向位置极限信号有效为 1；否则为 0
1	原点开关信号有效为 1；否则为 0
0	未设置位置锁存时为 1；否则为 0

（位置极限信号以及原嗲开关信号在什么电气状态下有效，由 CN 指令设置。）

参数：TS xxxx

x 为 A、B、C、D 或这些轴的任意组合，如省略参数，则表示使用全部参数。

应用：

运动中	是
程序中	是
指令行	是

默认值：

默认值	—
预设格式	3.0

操作数应用：

_TSx	x 轴当前状态值。

举例：

```
#A                          程序标号
V1=_TSA                     将 A 轴状态值赋给变量
IF V1>127                   判断最高位是否为 1
MG “Motor A is running”     输出信息
ENDIF                       判断结束
EN                          程序结束
```

TV

功能：查询回馈速度（Tell Velocity）

说明：

TV 指令返回编码器回馈信号的速度。单位是编码器信号计数每秒。返回的数值包含方向。这个资料是 0.25 秒内的平均值。

参数：TV xxxx

x 为 A、B、C、D 或这些轴的任意组合，如省略参数，则表示使用全部参数。

应用：

运动中	是
程序中	是
指令行	是

默认值：

默认值	—
预设格式	8.0

操作数应用：

_TVx	x 轴当前的编码器信号回馈速度。

相关指令：

TP	查询回馈位置

举例：

#A	程序标号
PT1	设置 A 轴为位置跟随模式
R=1000	设置跟随比例
#LOOP	循环标号
PAA= R*_TPA	依据 A 轴回馈位置设置 A 轴目标位置
SPA= R*_TVA	依据 A 轴回馈速度设置 A 轴速度
JP#LOOP	循环
EN	程序结束

UL

功能：上载程序（Upload）

说明：

UL 指令使控制器将存储的用户程序从通讯接口输出。输出的程序中不包括行号，以<control>Z 作为结束符号。

参数：无

应用：

运动中	是
程序中	否
指令行	是

默认值：

默认值	—
预设格式	—

操作数应用：

_UL	控制器中当前可定义变量数。用户最多可以定义 126 个变量。

相关指令：

DL	下载（程序）
<control>L<control>K	锁定程序

VA

功能：向量加速度（Vector Acceleration）

说明：

VA 指令设置插补运动中的向量加速度，适用与平面（二维）向量插补模式与直线插补模式。

参数：VA n

n 为无符号整数，范围 1024~1073740800。输入的数据如不是 1024 的倍数，回自动变换为不大于输入数值的 1024 整数倍数。如输入的数据小于 1024，则自动变换为 1024。

n=? 查询当前设置值。

应用：

运动中	是
程序中	是
指令行	是

默认值：

默认值	256000
预设格式	10.0

操作数应用：

_VA	向量加速度设置值。

相关指令：

VS	向量速度
VD	向量减速度
VP	向量位置
VE	向量路径结束
CR	圆弧
VM	平面（二维）向量插补模式
LM	直线插补模式
LI	直线插补路径
LE	直线插补路径结束
IT	平滑系数

举例：

VA 1024	设置向量加速度为 1024
VA ?	查询当前向量加速度
1024	返回 1024
VA 20000	设置向量加速度为 20000
VA ?	查询当前向量加速度
19456	返回 19456
ACCEL=_VA	将当前向量加速度值赋给变数 ACCEL

VD

功能：向量减速度（Vector Deceleration

说明：

VD 指令设置插补运动中的向量减速度。适用与平面（二维）向量插补模式与直线插补模式。

参数：VD n

n 为无符号整数，范围 1024~1073740800。输入的数据如不是 1024 的倍数，回自动变换为不大于输入数值的 1024 整数倍数。如输入的数据小于 1024，则自动变换为 1024。

n=? 查询当前设置值。

应用：

运动中	否
程序中	是
指令行	是

默认值：

默认值	256000
预设格式	10.0

操作数应用：

_VD	向量减速度设置值。

相关指令：

VS	向量速度
VA	向量加速度
VP	向量位置
VE	向量路径结束
CR	圆弧
VM	平面（二维）向量插补模式
LM	直线插补模式
LI	直线插补路径
LE	直线插补路径结束
IT	平滑系数

举例：

#VECTOR	程序标号
VM AB	定义向量平面
VA 1000000	设置向量加速度
VD 500000	设置向量减速度
VS 2000	设置向量速度
VP 10000, 20000	设置向量目标位置
VE	向量路径结束
BGS;AMS	开始运动，等到运动结束
EN	程序结束

VE

功能：向量路径结束（Vector Sequence End）

说明：

VE 指令表示平面（二维）向量路径结束。VE 与 LE 指令可以互换。

参数：VE n

n=?　　返回向量长度

无参数表示向量路径结束。

应用：

运动中	是
程序中	是
指令行	是

默认值：

默认值	—
默认格式	位置格式

操作数应用：

_VE	定义插补路径的长度，当插补运动结束后，值为 0。

相关指令：

VM	平面（二维）向量插补模式
VS	向量速度
VA	向量加速度
VD	向量减速度
CR	圆弧
VP	向量位置
BG	开始运动
CS	清除插补路径缓冲区

举例：

#A	程序标号
VM AB	定义向量平面
VP 1000, 2000	设置向量目标位置
CR 0, 90, 180	设置圆弧路径
VP 0, 0	设置向量目标位置
VE	向量路径结束
BGS	开始运动
AMS	等待运动结束
EN	程序结束

VF

功能：变量格式（Variable Format）

说明：

VF 指令用于设置控制器输出数值时的默认格式。VF 指令可以设置这些数据以 10 进制或 16 进制输出，以及输出时包含的整数和小数字数。

如果实际整数字元数大于输出设置值，则输出指定位数中的最大值。如数值的小数部分多于执行的小数字数，则进行四舍五入。

此指令只设置数值的输出格式，不影响数值的计算与比较。

参数：VF m.n

m 为-8~10 之间的整数，表示数值输出的整数字元数，如 m 为负，则表示以 16 进制形式输出。

n 为 0~4 之间的整数，表示数值输出的小数字数。

m=? 以 m.n 的形式返回当前设置值。

应用：

运动中	是
程序中	是
指令行	是

默认值：

默认值	10.4
预设格式	2.1

操作数应用：

_VF	当前设置值。

相关指令：

PF	变数格式
LZ	前导 0

举例：

VF 5.3	设置 10 进制输出，5 位整数，3 位小数
VF 8.0	设置 10 进制输出，8 位整数，无小数部分
VF –4.0	设置 16 进制输出，4 位整数，无小数部分

VM

功能：平面（二维）向量插补模式（Vector Mode）

说明：

VM 指令开始平面（二维）向量插补模式，并指定由哪两个轴构成向量平面。

插补向量路径由指令 VP 与 CR 设置。在开始运动前可以向插补路径缓冲区写入最多 31 段路径，其余的路径可以在运动开始后，缓冲区有空间后再陆续设置。为保证向量运动连贯平稳，用户要保证缓冲区中有足够长的向量路径。

在最后一段适量路径后，必须用 VE 声明向量路径的结束，这样控制器才会按照 VD 的设置减速停止。

参数：VM mn　　或　VM m, n

m、n 为构成向量平面的两轴，m 相当于 X-Y 平面中的 X 轴，n 相当于 X-Y 平面中的 Y 轴。m 可以是 A、B、C、D 轴，n 可以是 A、B、C、D 轴或虚拟轴 N。显然 m 与 n 不能相同。

应用：

运动中	否
程序中	是
指令行	是

默认值：

默认值	AB
预设格式	8.0

操作数应用：

_VM	当前的实时向量速度。

相关指令：

VP	向量位置
CR	圆弧
VE	向量路径结束

举例：

#A	程序标号
VM AB	定义向量平面
VP 1000, 2000	设置向量目标位置
VE	向量路径结束
BGS	开始运动
AMS	等待运动结束
EN	程序结束

VP

功能：向量位置（Vector Position）

说明：

VP 指令用于平面（二维）向量插补模式，向插补路径缓冲区中增加一段直线路径。VP 指令的参数为直线路径的终点位置。实际运动中的位置会受 ES 指令影响，ES 指令中参数较大的一轴，实际运动中输出脉冲数将大于 VP 指令设置的脉冲数。

参数：VP m, n<o>p

m 、 n 为带符号的整数，这两个数值表示这段路径的终点。数值范围 -2147483648~21474836487，但是一段路径的长度不能超过 8388607。这两个坐标是以插补运动开始（BGS）时的电机位置为相对零点。

o 为无符号偶数，2~3000000，表示运动到该段路径时的向量速度。

o=-1 表示使用 VV 指令设置的速度。

p 为无符号偶数，2~3000000，表示运动到该段路径末尾时的向量速度。

应用：

运动中	是
程序中	是
指令行	是

默认值：

默认值	—
预设格式	—

操作数应用：

_VPn 在插补运动过程中，表示 n 轴运动中的插补路径段起点的绝对坐标。可应用于直线插补或平面（二维）向量插补模式。n 为 A、B、C 或 D。

相关指令：

VM	平面（二维）向量模式
VE	向量路径结束
VV	向量速度变量
BG	开始向量运动

举例：

```
#A                  程序标号
VM AB               定义向量平面
VP 1000, 2000       设置向量目标位置
VP 2000, 4000       设置向量目标位置
CR 1000, 0, 360     设置圆弧路径
VE                  向量路径结束
BGS                 开始运动
AMS                 等待运动结束
EN                  程序结束
```

VR

功能：向量速度比例（Vector Speed Position）

说明：

VR 指令用于设定一个向量速度的倍数。VS 指令以及或每段路径中的参数都可以设置向量速度，而实际运动中的向量速度为这个设置值与 VR 设定倍数的乘积。VR 指令不影响实际加速度与减速度，但是速度变化所用的时间会受到 VR 的影响。

设置 VR 时，应确保实际速度不超过 3000000。

参数：VR n

n 为 0~10 之间的数，分辨率为 1/65536。

VR 指令不可以用 ？ 查询

应用：

运动中	是
程序中	是
指令行	是

默认值：

默认值	1
预设格式	2.4

操作数应用：

_VR	向量速度比例当前设置值。

相关指令：

VS	向量速度

举例：

```
#A                      程序标号
XQ#SPEED, 1
#LOOP
VM AB                   定义向量平面
VP 1000, 2000           设置向量目标位置
CR 1000, 0, 360         设置圆弧路径
VE                      向量路径结束
VS 2000                 设置向量速度
BGS                     开始运动
AMS                     等待运动结束
JP#LOOP                 循环
#SPEED                  程序标号
R=_TVD/100              根据 D 轴编码器输入速度调整实际向量速度
IF R>10;R=10;ENDIF      避免输入速度过大超出允许范围
VR R                    设置向量速度比率
JP#SPEED                循环
EN                      程序结束
```

VS

功能：向量速度（Vector Speed）

VS 指令设置插补运动中的向量速度。适用与平面（二维）向量插补模式与直线插补模式。向量速度是指参与插补各轴的速度按照向量方式迭加的结果，控制器回根据设置的向量速度与运动的路径自动计算各轴的速度。

参数：VS n

n 为偶数，范围 2~3000000。

n=? 查询当前设置值。

应用：

运动中	是
程序中	是
指令行	是

默认值：

默认值	25000
预设格式	8.0

操作数应用：

_VS	当前设置值。

相关指令：

VP	向量位置
CR	圆弧
LI	直线插补路径
BG	开始运动
VV	向量速度变量
VR	向量速度比率

举例：

VS 2000	设置向量速度
VS ?	查询向量速度设置值
2000	返回 2000

VV

功能：向量速度变量（Vector Speed Variable）

VV 指令设置 VM 或 LM 插补运动中的向量速度变量。用于在运动中调整某段路径的向量速度。如在定义某段插补路径时将向量速度（<后的参数）设置为-1，则该沿段路径运动时的速度为实际运动时 VV 指令设置的速度。

参数：VV n

n 为偶数，范围 2~3000000。

n=? 查询当前设置值。

应用：

运动中	是
程序中	是
指令行	是

默认值：

默认值	0
预设格式	8.0

操作数应用：

_VV	当前设置值。

相关指令：

VP	向量位置
CR	圆弧
LI	直线插补路径
BG	开始运动
VS	向量速度
VR	向量速度比率

举例：

VV 20000	设置向量速度变量
VP 1000, 2000<-1>100	设置路径的运动速度为变量值
VSPEED=_VV	将当前设置值赋给变数

WH

功能：当前句柄（Which Handle）

说明：

WH 指令返回控制器当前使用的通讯句柄。控制器会回馈 IHA、IHB、IHC 或 IHD 来表明当前通讯使用的以太网句柄。如果当前是使用串口通讯，控制器会返回 RS232。

参数：无

应用：

运动中	是
程序中	是
指令行	是

默认值：

默认值	—
预设格式	—

操作数应用：

_WH	数字表示当前句柄。0~3 表示以太网句柄 A~D，-1 表示串口。

相关指令：

TH	查询句柄

举例：

```
:WH
IHC

:WH
RS232
```

提示：虽然允许 WH 在程序中使用，控制器不会提示错误，程序也可以正常运行，但是这种用法本身得不到有意义的结果。

WT

功能：等待（Wait）

说明：

WT 为事件触发指令。该指令暂停程序的运行，直到等待了指定的时间。等待的时间从 WT 指令开始在执行算起，时间单位为毫秒。

参数：WT n

n 为 0 至 2 亿之间的整数，表示等待时间，单位为毫秒。

应用：

运动中	是
程序中	是
指令行	否

默认值：

默认值	—
预设格式	—

相关指令：

AT	等待指定时刻

举例：

#A	程序标号
PR 50000	相对位移
BG A	开始运动
AM A	等待运动结束
WT10000	等待 10 秒
SB 1	通用输出 1 导通
EN	程序结束

XQ

功能：运行程序

说明：

XQ 指令开始运行存储在控制器内的程序。程序从指定的行号或标号处开始运行。控制器最多可以同时运行 4 段程序。

参数：XQ #A, n　或　XQ m, n

A 为程序中存在的标号

m 为程序中存在的行号

n 为运行程序的线程号，0~3。

如省略#A 与 m，则程序从 0 行开始；如省略 n，则在 0 线程运行。

应用：

运动中	是
程序中	是
指令行	是

默认值：

默认值	0, 0
预设格式	—

操作数应用：

_XQn	n 线程当前运行的行号，-1 表示该线程当前未运行程序。

相关指令：

HX	中止运行

举例：

XQ #APPLE, 0	在 0 线程，从#APPLE 处开始运行程序
XQ #DATA, 2	在 2 线程，从#DATA 处开始运行程序
XQ	在 0 线程，从头开始运行程序

ZA

功能：数据记录用户变量

说明：

ZA 指令给每轴的数据记录中的用户变量赋值。这四个变量的值将作为控制器状态数据记录的一部分。

参数：ZA n, n, n, n　　或　ZAx=n

n 可以是表示数值的变量、操作数、数字或数学表达式。此时 n 的值为 4 字节整数，-2147483648~2147483647 之间。n 也可以是在双引号中的字符串，此时 n 的最大长度为 4 个字符。

n=?　查询当前设置值

应用：

运动中	是
程序中	是
指令行	是

默认值：

默认值	0
预设格式	10.0

操作数应用：

_Zax	x 轴当前设置值。

相关指令：

DR	数据记录的更新速率
QZ	数据记录格式

举例：

```
#Thread
ZAA=MyVar                    不断更新 ZA
JP# Thread
```

ZS

功能：清除堆栈（Zero Subroutine Stack）

说明：

每次调用子程序，都会将堆栈指针加一，并将程序入口保存至堆栈。从子程序返回时，回到指标指向的入口，并将指标减一。

ZS 指令用来调整堆栈指针。ZS1 将堆栈指针减一，最后一个保存的程序入口消失，最近执行的一次子程序调用变成了跳转。ZS0 则彻底清空堆栈。

如果在输入中断子程序中使用了 ZS 指令，则子程序的末尾不能使用 RI 指令，必须使用 EN 指令。如果需要继续响应输入中断信号，必须重新执行 II 指令。

参数：ZS n

n=0　清 空 堆 栈

n=1　清除堆栈中最后一条记录

省略参数，则清空堆栈

应用：

运动中	是
程序中	是
指令行	否

默认值：

默认值	0
预设格式	3.0

操作数应用：

_ZSn	n 线程中当前堆栈指针，0~16。未执行过子程序调用时为 0。

举例：

```
#A                  程序标号
II 1                设置输入中断
#B;JP#B;EN          空循环，确保回应中断
#ININT              输入中断处理程序
MG ”INTERRUPT”      输出信息
S=_ZS               查询堆栈指针
S=                  输出堆栈指针
ZS                  清空堆栈
S=_ZS               查询堆栈指针
S=                  输出堆栈指针
EN                  程序结束
```