§ 6 PMAC 在线命令说明

<CONTROL-A>

功能: 中断所有程序和运动。

范围: 全局

语法: ASCII值 1D; \$01

备注: 这条命令中断所有的运动程序并停止所有卡上非编程产生的移动。它还可将任意被禁止的或者开环的电机转换到允许的零速度闭环状态。每个电机都将降低到某个度,该速度是由其自己的电机I变量Ix15定义的。然而,一个多轴系统在这种减速过程中有可能不会停止在程序预定的路径上。

一个**<CTRL-A>**式的停止并不意味着可以正常恢复,因为通常情况下各个轴不会 止在程序预定点上。除非给出一个**PMATCH**命令或者**I14**被置为 1 (这使**PMA** 使用中断时的位置作为移动起始点),下一个编程产生的移动将不会正常运行。另外, 当中断出现时,也可以对每个电机发出联机**J=**命令,使它移动到编程时设计的终点。 然后,程序可以用**R**(run)命令重新开始。

如果要以一种比较容易恢复的方式来停止一系列电机,可以采用Quit(Q或<CTRL-Q>)或者Hold(H或<CTRL-H>)来代替该命令。

当PMAC被设置成启动时所有电机均停止(Ix80=0),则本命令可以用来使能(enable)所有电机。(必须保证这些电机不是由PMAC来换向的----在那种情况下,每个电机由\$命令使能。)

对于在一个串联菊花链上的多个卡,本命令对所有此链上的卡都起作用,而不论当前软件中的地址是什么。

参考 Stop命令(使你的应用安全)

在线命令A,\$, J=, PMATCH, H, <CTRL-O>, Q, <CTRL-Q> |变量I14, lx15, lx80。

<CONTROL-B>

功能: 报告所有电机的状态字。

范围: 全局

语法: ASCII值2D: \$02

备注: 本命令使PMAC以十六进制ASCII码形式向主机报告所有电机的状态字。从**#1**电机开始,每个电机的状态字长为**12**个字符,电机之间用空格隔开。 状态字中的每个位的详细含义将在**?** 命令部分进行描述。

对于在一个串联菊花链上的多个卡,本命令只对地址在当前软件中的卡起作用

(@n) 。

范例 <CTRL-B>

参考 在线命令<CTRL-C>, <CTRL-G>, ?, @n

内存映象图寄存器X:\$003D, X:\$0079等, Y:\$0814, Y:\$08D4; 推荐的M变量定义Mx30-Mx45。

<CONTROL-C>

功能: 报告所有坐标系的状态字。

范围: 全部

语法: ASCII值3D; \$03

备注: 本命令使PMAC以十六进制ASCII码形式向主机报告所有坐标系的状态字。从坐标系#1开始,每个坐标系的状态字长为12个字符,坐标系之间用空格隔开。本命

令对各个坐标系所得到的报告字符与使用??命令是一样的。

状态字中的每个位的详细含义将在??命令部分进行描述。

对于在一个串联菊花链上的多个卡,本命令只对地址在当前软件中的卡起作用(利用@n命令)。

范例 <CTRL-C>

A80020020000 A80020020000 A80020020000 A80020020000

A80020000000 A80020000000 A80020000000

A80020000000<CR>

参考 在线命令<CTRL-B>, <CTRL-G>, ??;

内存映象图寄存器Y:\$0817, Y:\$08D7等, X:\$0818, X:\$08D8等;

推荐的M变量定义Mx80-Mx90。

<CONTROL-D>

功能: 禁止所有PLC(可编程控制器)程序。

范围: 全局

语法: ASCII值4D; \$04

备注: 本命令使所有的PLC程序被禁止(特别是停止那些正在运行的程序)。它与DISABLE

PLC 0..31和DISABLE PLCC 0..31的作用是相同的。当一个PLC中的CMD和

SEND语句已经运行异常时,本命令是非常有用的。

对于在一个串联菊花链上的多个卡,本命令对所有此链上的卡都有作用,而不论当

前软件中的地址是什么。

范例 TRIGGER FOUND

TRIGTRIGER FOTRIGGER FOUND

TRTRIGTRIGGER FOUND (PLC发出的SEND信息失去控制)

<CTRL-D> (禁止所有PLC)

(没有更多信息;现在可以编辑PLC)

参考 在线命令DISABLE PLC,ENABLE PLC,DISABLE PLCC,ENABLE PLCC,OPEN PLC

程序命令DISABLE PLC,ENABLE PLC,DISABLE PLCC,ENABLE PLCC,COMMAND,SEND

<CONTROL-E>

功能: 以二进制的形式报告设置地址的内容。

范围: 全局

语法: ASCII值5D: \$05

备注: 本命令使PMAC向主机报告某些地址的内容,这些地址是由数据收集I变量(I20-I44)指定的。这非常象一次数据收集和报告,只是回报的值是以二进制形式而不是以ASCII形式得到的。

每一短字长3个字节(LSByte(低字节)在前);每一长字有6个字节(LSByte 在前)。使用串口或者VME总线时,字节是按LSByte在前的方式依次报告的。而使用PC总线或者STD总线时,每次送出3个字节。这时,LSByte放在*base address* + 7(基地址+7,即通常的地址)中,中间的字节放在*base address* + 6中,MSByte 高字节)放在*base address* + 5中。其中没有前置或后置的握手字符;你必须知道将得到的字节的数目,然后按此编写自己的接收程序。

对于在一个串联菊花链上的多个卡,本命令只对地址在当前软件中的卡起作用(利用**@n**命令)。

注意:本命令不能在VME总线上使用。

范例 下面这个例子说明了你怎样试验本命令,先写打开的内存寄存器,然后用**<CTRL-E>** 读取数据内容。

I20=7<CR>
I21=\$0007F0<CR>
I22=\$4007F1<CR>
I23=\$8007F2<CR>
WY\$07F0,\$001122<CR>
WX\$07F1,\$334455<CR>
WL\$07F2,\$024680135791<CR>
<CTRL-E>
<\$22><\$11>\$<\$00><\$55><\$44><\$33><\$91><\$57><\$13>
<\$80><\$46><\$02>

在线命令GATHER, LIST GATHER

<CONTROL-F>

功能: 报告所有电机的跟随误差。

范围: 全局

语法: ASCII值6D; \$06

备注: 本命令使PMAC向主机报告所有电机的跟随误差。误差是用一个ASCII码串报告的,每个误差都以脉冲显示,精确到小数点后一位。每个电机的误差之间有一个空格符

有关电机误差计算方法的更多细节,可以查阅在线F命令

对于在一个串联菊花链上的多个卡,本命令只对地址在当前软件中的卡起作用(利用**@n**命令)。

范例 <CTRL-F>

0.5 7.2 -38.3 1.7 0 0 0 0 CR>

<CONTROL-G>

功能: 报告全局状态字。

范围: 全局

语法: ASCII值7D; \$07

备注: 本命令是PMAC向主机报告全局状态字,状态字是十六进制的ASCII格式,其长为12个字符。得到的字符和用???命令得到的是一样的,区别在于响应的最后没有命令确认字符(<ACK>或者<LF>,由I3确定)被送出。

状态字中的每个位的详细含义将在???命令部分进行描述。

对于在一个串联菊花链上的多个卡,本命令只对地址在当前软件中的卡起作用(用**@n**命令)。

范例 <CTRL-G>

003000400000<CR>

参考 在线命令**<CTRL-B>**, **<CTRL-C>**, **???** 内存映象图寄存器X:\$0003, Y:\$0003。

<CONTROL-H>

功能: 擦除上一个字符。

范围: 全局

语法: ASCII值8D; \$08(**<BACKSPACE>**)。

备注: 本字符通常是在用终端模式与PMAC对话时用**<BACKSPACE>**(退格键)输入的。它使最近被输入到PMAC的命令行接收缓冲区的字符被擦除。

参考 与PMAC对话

在线命令<CTRL-O>(所有进给保持)。

<CONTROL-I>

功能: 重复上一命令行

范围: 全局

语法: ASCII值9D; \$09 (**<TAB>**)

备注: 本字符有时候是用**<TAB>**(制表键)输入的,它使最近被送给PMAC的字母和数字命令行被重复执行。这样就给加快执行那些重复的任务提供了一个方便的办法,特别是当PMAC工作在交互式的终端状态时。其它的控制字符命令不能用本命令重复。

注意:大多数版本的PMAC执行程序由于它们自己的目的限制了**<CTRL-**l>或者**<TAB>**,并且即使在终端模式下也不把它们送到PMAC。

范例 下面的例子说明了怎样用TAB键寻找某些东西:

PC<CR> P1:10<CR> < **TAB>** P1:10<CR>

<TAB> P1:10<CR> <TAB> P1:11<CR>

在线命令<CONTROL-Y>。 参考

<CONTROL-K>

功能: 停止所有电机。

范围: 全局

语法: ASCII值11D; \$0B

通过对卡上电机打开伺服环,使输出为零以及将放大器使能信号(AENAn)断开(极 备注:

性由跳线**E17**确定),本命令能停止所有电机的输出。如果有运动程序正在执行, 它们都将被自动终止。(使用针对电机的K(kill)命令。如果电机位于一个坐标系 中,而该坐标系又正在执行一个运动程序,那么在PMAC接收K命令之前,就必 须用A(abort)或Q(quit)命令终止这个程序。)

对于在一个串联菊花链上的多个卡,本命令对所有此链上的卡都有作用,而不论当 前软件中的地址是什么。

在线命令K, A, <CONTROL-A> 参考

<CONTROL-L>

功能: 关闭打开的转动缓冲。

范围: 全局

语法: ASCII值12D; \$0C

本字符使PMAC关闭它的打开的程序转动缓冲区。这与CLOSE命令的效果是一样 备注:

的,但是发送起来却要快一些。**<CTRL-L>**主要是在需要反复打开和关闭转动缓冲 区时使用。关闭转动缓冲之后,就不会把一个在线命令错当成一个缓冲命令。

范例 <CTRL-U> : 打开转动缓冲

> X10 Y20 F5 M3 ; 将程序行放入缓冲 X30 Y40 F5 ; 将程序行放入缓冲 <CTRL-L> ; 关闭转动缓冲 М1 ; M1值的在线命令 : PMAC的响应

旋转运动的程序缓冲(编写一个运动程序) 参考

在线命令<CTRL-U>, OPEN ROT, CLOSE

<CONTROL-M>

功能: 输入命令行

范围: 全局

ASCII值13D: \$0D(<CR>) 语法:

备注: 一般本字符叫做<CR>(回车),它使PMAC命令接收缓冲中的字母和数字字符被

解释并执行。(控制字符命令不需要一个<CR>字符再执行。)

要注意的是多个PMAC以菊花链形式串联的情况,本字符将会同时影响所有的卡,

而不是仅仅在软件给出地址的卡。对于多个卡上同时进行的动作,最好在发出**<CR>**字符前载入所有卡上的命令行接收缓冲。

范例 #1J+<CR>

P1<CR>

@0&1B1R@1&1B7R<CR> (这使串联菊花链上的0号卡的CS 1执行PROG 1, 而与此同时, 1号卡的CS 1执行PROG 7)

参考 与PMAC对话。

<CONTROL-N>

功能: 报告命令行的校验和。

范围: 全局

语法: ASCII值14D: \$0E

备注: 本字符使PMAC计算并报告当前命令行中字母和数字字符的校验和(特别是上一

个回车符之后的字符)。

对于典型的应用,主计算机是把除回车符之外命令行整个送出的。而如果送出一个 **<CTRL-N>**字符,则PMAC将会返回校验和的值。如果这个值和主机自己计算得 到的校验和相符,主机就送出**<CR>**使PMAC执行这一命令行。如果两个校验和的 值不吻合,主机就送出一个**<CTRL-X>**命令擦除命令行,然后重新送出这一行,重 复上面的处理过程。

注意: PMAC执行程序终端模式将不会显示由**<CTRL-**N>命令得到的检验和的值。

范例 当I4=1和I3=2时:

主机送出: J+<CTRL-N>

PMAC送出: <117dec> (117=74[J]+43[+]; 正确)

主机送出: <CR>

PMAC送出: *<ACK><117dec>* (再次送握手和校验和)

主机送出: J/<CTRL-N>

PMAC送出: <122dec> (122!=74[J]+47[/]; 不正确)

主机送出: **<CTRL-X>** (擦除不正确的命令) **J/<CTRL-N>** (再次发送命令)

PMAC送出: <121dec> (121=74[J]+47[/]; 正确)

主机送出: <CR>

PMAC送出: <*ACK*><*121dec*> (再次送握手和校验和)

参考 通讯校验和(编写一个主机通讯程序)

I变量 I3, I4

在线命令<CTRL-M>(<CR>), <CTRL-X>

<CONTROL-O>

功能: 所有坐标系进给保持。

范围: 全局

语法: ASCII值15D; \$0F

备注: 本命令使PMAC中的所有坐标系实现进给保持。进给保持与%0命令非常相象,它

使坐标系停止运动但并不偏离所遵循的路径,即使是曲线路径也是如此。而当进给保持时,坐标系是以lx95确定的速度减速的,并可以用一个R(run)命令再次启动。之后系统会以lx95确定的速度加速,直到达到预期的%值(遵循内部或外部的时间基准)。从那以后,任何时基变化都按lx94中的速度产生。

对于在一个串联菊花链上的多个卡,本命令对所有此链上的卡都有作用,而不论当前软件中的地址是什么。

当一个闪速内存PMAC处于引导状态时(启动时E51为开),**<CTRL-O>**命令使PMAC

开始它的固件(firmware)重载命令。所有随后被送到PMAC的字符都被解释成PMA C

工作固件的机器码字节,并覆盖掉闪速内存中已有的工作固件。

参考 重置PMAC(与PMAC对话)

I变量I52,Ix94,Ix95

在线命令<CTRL-

H>(退格), **H**(进给保持), **R**(运行), **%**(进给率过载), ****(序停止)。

<CONTROL-P>

功能: 报告所有电机位置

范围: 全局

语法: ASCII值16D; \$10

备注: 本命令向主机报告所有电机的位置。这些位置以一串ASCII码表示,并按比例统计, 精确到小数点后一位。每两个电机的位置之间用空格符隔开。

不断地发出**<CTRL-P>**命令可使**PMAC**执行程序中的位置窗口工作从而在窗口中重排结果。

PMAC报告的值是这么得到的: *绝对位置寄存器*加*位置偏差寄存器*加*补偿校正寄存器*,如果**lx05**的16位为1(转向轮偏移模式),还要减去主位置寄存器。

对于在一个串联菊花链上的多个卡,本命令只对地址在当前软件中的卡起作用(利用@n命令)。

范例 <CTRL-P>

9999.5 10001.2 5.7 -2.1 0 0 0 0 CR>

参考 在线命令P, <CTRL-V>, <CTRL-F>。

<CONTROL-Q>

功能: 退出所有正在执行的运动程序。

范围: 全局

语法: ASCII值17D; \$11

备注: 本命令使任一坐标系中所有正在运行的动作程序在结束已经计算完的运动之后全部停止,程序可以用**R**(运行)或者**S**(单步)命令从停止点重新开始执行。

对于在一个串联菊花链上的多个卡,本命令对所有链上的卡都有作用,而不论当前软件中的地址是什么。

参考 在线命令**<**CTRL-A>, **<**CTRL-K>, **<**CTRL-O>, **<**CTRL-R>, **<**CTRL-S>, Q。 运动程序命令**STOP**。

<CONTROL-R>

功能: 开始执行所有坐标系中的运动程序。

范围: 全局

语法: ASCII值18D; \$12

备注: 本命令和向PMAC中所有坐标系发出**R**(运行)命令是一样的。每一个激活的坐标系(即最起码被分配了一个电机)在运行程序前必须先指定一个运动程序(起初这是用**B**{*prog num*}命令完成的)。

对于在一个串联菊花链上的多个卡,本命令对所有此链上的卡都有作用,而不论当前软件中的地址是什么。

当一个闪速内存PMAC处于引导状态时(启动时E51为开),**<CTRL-R>**命令使PMAC

进入它的正常工作模式,此时它的I变量,换算表和VME/DPRAM地址都是出厂默认值。

参考 执行一个运动程序(编写一个运动程序)

重置PMAC(与PMAC对话)

在线命令R,B。

跳线E51。

<CONTROL-S>

功能: 单步执行所有坐标系中的运动程序。

范围: 全局

语法: ASCII值19D; \$13

备注: 本命令与向PMAC中所有坐标系发送**S**(单步)命令是一样的。每一个激活的坐标系(即最起码被分配了一个电机)在运行程序前必须先指定一个运动程序(起初这是用**B**{*prog num*}命令完成的)。

一个没有运行的程序将执行所有的程序行直到完成下一个运动命令(移动或停止)。如果它先收到一个BLOCKSTART命令,那它会运行到完成下一个BLOCKSTART命令。

如果一个程序已经运行在连续执行模式(利用R(运行)命令),那么一个S命令将

使程序进入单步模式(在执行完下一条运动命令后停止)。在这种情况下,它与一个 Q(退出)命令是完全一样的。

对于在一个串联菊花链上的多个卡,本命令对所有此链上的卡都有作用,而不论当前软件中的地址是什么。

参考 在线命令<CTRL-A>, <CTRL-O>, <CTRL-Q>, <CTRL-R>, A, H, O, Q, R, S。

运动命令BLOCKSTART, BLOCKSTOP, STOP。

控制板端口(JPAN)输入STEP/。

<CONTROL-T>

功能: 改变串口半双向/全双向模式。

范围: 全局

语法: ASCII值20D; \$14

备注: 本命令使串口通讯在半双向(PMAC将不会向主机回送字符)和全双向(PMAC将

向主机回送字符)。开启默认的设置是半双向。

当多个PMAC以菊花链方式串连时,本命令是无效的。

参考 数据完整性检查(编写一个主机通讯程序)

在线命令**<CTRL-Z>**。

<CONTROL-U>

功能: 打开旋转程序缓冲区。

范围: 全局

语法: ASCII值21D; \$15

备注: 本字符使PMAC打开所有已有的旋转程序缓冲区以便进行输入。它和**OPEN**

ROTARY命令的效果是完全相同的,只是发送更快一些。利用<CTRL-L>命令,它允许快速打开和关闭旋转缓冲区。这样,就可以发送在线命令关闭缓冲区,以免这

些命令被错误地当作缓冲区命令。

范例 <CTRL-L> ; 关闭旋转缓冲

M1; 值为M1的在线命令

1 ; PMAC响应

<CTRL-U> ; 打开旋转缓冲

X10 Y20 F5 M3;将程序行放入缓冲X30 Y40 F5;将程序行放入缓冲

参考 旋转运动程序缓冲区(编写一个运动程序)

在线命令<CTRL-L>, OPEN ROT, CLOSE

<CONTROL-V>

功能: 报告所有电机的速度。

范围: 全局

语法: ASCII值22D: \$16

备注: 本命令使PMAC向主机报告所有电机的速度。速度单元是按每个伺服环来编码统计

的,并精确到小数点后一位。通过重复发送**<CTRL-V>**命令可以使PMAC执行程

序中的<F7>速度窗口工作,将响应的结果显示在屏幕上。

为了把这些值换算为计数脉冲/毫秒,可以将响应结果乘以8,388,608*(lx60+1)/110 (伺服周期/毫秒)。

注意:报告的速度值是通过减去串联的伺服环的位置得到的。所以,它们可能含有很多干扰。为了将它们显示出来,最好使用M变量访问寄存器

Y:\$082A: Y:\$08EA等中的平均速度值。

对于在一个串联菊花链上的多个卡,本命令只对地址在当前软件中的卡起作用(**@n**)。

> 在线命令**<CTRL-F>**, **<CTRL-P>**, **V** 内存寄存器X:\$0033, X:\$006F等 推荐的M变量定义Mx66

<CONTROL-W>

功能: 从双端口RAM(DPRAM)中取命令行

范围: 全局

语法: ASCII值23D; \$17

备注: 本命令使PMAC从双端口RAM(需要选件2)的寄存器中取一条ASCII命令行。

这允许通过双端口RAM发送命令,而只有**<CONTROL-W>**字符本身需要通过通讯

口发送。

注意: DPRAM通讯的这种用法很多已经被双向ASCII

DPRAM缓冲区取代了,双向ASCII DPRAM缓冲区可以用I58使能。

ASCII命令可以放在任何不用的内存寄存器中,但主要用在双端口RAM中。 <CTRL-W>可以通过任意端口传输,但主要通过总线端口(PC或VME)应用。 当接收到一个<CTRL-W>时,要通过两阶段的地址处理过程来通知PMAC到哪里获 取命令。I变量I47包含PMAC中一个寄存器的地址,而命令的起始内存地址就放 在这个寄存器里。也就是说,I47包含指向命令的指针地址,而不是命令地址本身。 指针寄存器通常就在DPRAM中。这种双重间接地址允许很多命令可以预载到 DPRAM中,这样对于一条新命令只需改变指针寄存器的内容。

当PMAC接收到一个<CTRL-W>命令时,它首先检查I47指定的Y寄存器地址。接下来,它会查看由这个寄存器确定的地址。它随后从内存中读入命令行,这行命令存放在由Y寄存器的LSByte(0-7位)指定的地址开始的位置。第二个字符将来自同一寄存器的MSByte(8-15位);第三个字符来自同一地址的X寄存器的LSByte;第四个字符来自X寄存器的MSByte;第五个字符来自更高一位地址的Y寄存器的LSByte;以此类推,直到出现一个<NULL>字符(ASCII值为零)。为了能够正常运行,一个命令行在<NULL>之前必须有一个<CR>字符(ASCII值为13)

PMAC按这种方法通过设定的主机端口而不是DPRAM对接收到的任意命令进行响应。

对于在一个串联菊花链上的多个卡,本命令对所有此链上的卡都起作用,而不论当前软件中的地址是什么。

范例 使用本命令,可以先从DPRAM的PMAC这一边开始工作。而这可以用PMAC执行程序的终端状态轻松实现。

首先向DPRAM中写入命令串。用下面的命令向DPRAM中由**\$D201**开始的地址写入**P1=2<CR>**,再向由**\$D203**开始的地址写入**P1=0<CR>**。

WY\$D201,\$3150 ; 50是P, 31是1 WX\$D201,\$323D ; 3D是=, 32是2

WY\$D202,\$000D : 0D是<CR>, 00是<NULL>

WY\$D203,\$3150; 50是P, 31是1WY\$D203,\$303D; 3D是=, 30是0

WY\$D204,\$000D ; 0D是**<CR>**, 00是**<NULL>**

接下来,将I47设置为\$D200(键入I47=\$D200),使它指向第一条命令的WY\$D200

\$D201。现在发出**<CRTL-W>**命令。变量P1现在应该被置为2,可以键入**P1**验证一下。

为了指向下面一条命令,键入**WY\$D200**,**\$D203**。发出**<CRTL-W>**命令,P1应该被改回**0**,再次键入**P1**进行验证。

当然,在实际应用中,你应该从主机一边写**DPRAM**,而不是**PMAC**一边。但是通过上面的例子,可以很好地理解和调试这些操作方法。

参考 双端口RAM说明(编写一个主机通讯程序) I变量|47, |58。

<CONTROL-X>

功能: 取消正在进行的通讯。

范围: 全局

语法: ASCII值24D; \$18

备注: 本命令使PMAC停止任一已经开始发送的消息,即使是多行的消息。它还使PMAC 清空来自主机的命令队列,所以它将擦除任一已被部分发送的命令。

当你发送一个希望得到精确响应格式的查询命令,而又不清楚之前PMAC做了些什么,那么在发出查询命令前发送本命令是非常有用的。因为它确保在期望的响应之前不会有其它的信息传来。同样,当试图建立最初的通讯联系时,本命令常常也是作为第一个字符被发送到PMAC。

注意:本命令清空PMAC

RAM中的命令队列,但是它不能擦除已经在响应端口中的1个或2个字符。 如果必需的话,一个完整的清除响应的算法应该再包括读入2个字符并能够 暂停的命令。

对于在一个串联菊花链上的多个卡,本命令对所有此链上的卡都有作用,而不论当前软件中的地址是什么。

参考 在线命令<CTRL-H>, <CTRL-Z>。

<CONTROL-Y>

功能: 报告上一个命令行。

范围: 全局

语法: ASCII值25D; \$19

备注: 本命令使PMAC报告主机的上一个命令行(不包括最后的<*CR*>)并将这一行重新输入到命令队列中,准备收到下一个<*CR*>后就执行。这允许以终端模式和PMAC通讯的用户可以取到上一个命令,并能用退格和键入来进行编辑从而完成需要的修改。当主机发送一个<*CR*>后,这个命令将被再次执行。

参考P123=5<CR>; 设置第一个值P124=7<CR>; 设置第二个值P123<CR>: 查询第一个值

5 ; PMAC响应的值

<CTRL-Y> : 让PMAC报告上一个命令

 P123
 ; PMAC报告上一个命令

<BACKSPACE>4<CR> ; 修改为P124并发送 7 ; PMAC反馈P124的值

参考 在线命令<CONTROL-I>。

<CONTROL-Z>

功能: 将PMAC设置为串口通讯状态。

范围: 全局

语法: ASCII值26D; \$1A

备注: 本命令使PMAC的串口成为工作的通讯输出端口。所有PMAC对主机的响应都将通过串口发送。这种模式将一直持续,直到通过总线(并行)端口接收到一条命令

使总线端口成为工作的通讯输出口。PMAC的开机/复位时都以串口为工作端口。

如果你试图通过串口和PMAC建立通讯联系,最好在所有查询命令之前发送本字符,

以确保PMAC将会通过串口进行响应。

无论哪一个端口工作在输出口,PMAC都能够通过任意口接收命令。用户必须保证不会也无同时从两个端口发送而选成会会的混淆

不会由于同时从两个端口发送而造成命令的混淆。

范例 主机串口发送: P1

PMAC从串口响应: 12

主机总线端口发送: P1=P1+1

主机串口发送: P1 PMAC从总线端口响应: 13

(主机串口不获得响应)

主机串口发送: <CTRL-Z>P1

PMAC从串口响应: *13*

参考 在线命令<CTRL-T>, <CTRL-X>;

跳线E44-E47。

#

功能: 报告当前被选址的电机。

范围: 全局

语法: #

备注: 本命令使PMAC返回当前被主机选址的电机----它执行主机的指定的电机命令。

注意:对于指定电机的控制板输入,可以从控制板端口用硬件选择一个电机

。该电机可以在PMAC中用程序里的COMMAND语句进行选址。

 #
 ; 询问PMAC哪一个电机被选址

 2
 ; PMAC报告电机2被选址

参考 控制板输入(将PMAC与机器连接)

在线命令#{常量}, &, &{常量}, @{常量}

程序命令ADDRESS, COMMAND

#{常量}

范例

功能: 将一个电机编址。

范围: 全局

语法: #{常量}

位置:

{常量}是一个1到**8**之间的整数,它表示将被编址的电机的号码。

备注 本命令使由**{常量}**指定的电机成为被编址的电机(即在线电机命令的操作对象)。编址是模态的,所以所有后续的指定电机的命令都将影响此电机,直到另外一个电机被编址。在开机/复位时,电机1是被编址的电机。

注意:对于指定电机的控制板输入,可以从控制板端口通过硬件选择一个电机。该电机可以在PMAC中用程序里的COMMAND语句进行选址。

范例 #1J+ ; 命令电机1正向手动进给

J- ; 命令电机1负向手动进给 #2J+ ; 命令电机2正向手动进给 J/ ; 命令电机2停止进给

参考 控制板输入(将PMAC与机器连接)

编址命令(与PMAC对话)

程序命令COMMAND, ADDRESS

在线命令#{常量}, &, &{常量}, @{常量}

#{常量}->

功能: 报告指定电机所在坐标系的坐标轴的定义。

范围: 指定的坐标系

语法: #{常量}

位置: {常量}是一个1到8之间的整数,它代表将被查询坐标中定义的电机的号。

备注 本命令使PMAC报告在*当前被选址的坐标系*中,被指定电机的坐标轴的定义。如果在当前选址系统中的电机并没有定义到一个坐标轴,PMAC将会返回一个**0**(即使电机在别的坐标系中被指定了一个坐标轴)。在一个坐标系中,一个电机一次只

能有一个坐标轴定义。

范例 &1 ; 选坐标系1

#1-> ; 查询坐标系1中的电机1的坐标轴定义

100000X ; PMAC返回坐标轴的定义

&2 : 选坐标系2

#1-> ; 查询坐标系1中的电机1的坐标轴定义 0 ; PMAC显示该坐标系中坐标轴的定义

参考 轴,坐标系(设置一个坐标系)

在线命令#**{常量}->0**,#**{常量}->{轴定义},UNDEFINE,UNDEFINE ALL**。

#{常量}->0

功能: 清除指定电机的坐标轴定义

范围: 指定的坐标系

语法: #{常量}->0

位置: **{常量}**是一个1到8之间的整数,它代表将被清除坐标轴定义的电机的号。

备注 如果当前选址的坐标系中,指定电机已经被定义了一个坐标轴,本命令可以清除 该电机的轴定义。如果电机在另一个坐标系中也被定义一个坐标轴,本命令将不

起作用。这允许指定电机被重新定义一个坐标轴,这个轴可以在当前坐标系中也可以在另一个坐标系中。

可以把本命令和另两个命令进行比较: UNDEFINE擦除当前选址坐标系中所有轴 定义, UNDEFINE ALL则擦除所有坐标系中的所有轴定义。

范例 下面的例子说明了怎样把一个电机从一个坐标系移到另一个坐标系中。

&1 ; 选坐标系1 #4-> ; 查询# 4的定义

5000A ; PMAC响应 #4->0 ; 清除定义 &2 ; 选坐标系2

#4->5000A ; 在坐标系2中进行新的定义

参考 轴,坐标系(设置一个坐标系)

在线命令UNDEFINE, UNDEFINE ALL, #{常量}->{轴定义}

#{常量}->{轴定义}

功能: 为指定电机分配一个坐标轴定义。

范围: 指定坐标系

语法: #{常量}->{轴定义}

位置: {常量}是一个1到8之间的整数,它代表将被定义坐标轴的电机的号。

{轴定义}包括:

1到3组[{比例因子}] {坐标轴},之间用字符+分开,其中:

{比例因子}是可选的,它是一个浮点常数,代表每坐标轴单位(工程单位)中的电机读数。如果没有给出此项值,PMAC认为该值为1.0:

{坐标轴}是一个字母(X, Y, Z, A, B, C, U, V, W)代表电机将要匹配的坐标轴。

[+{偏移}](可选)是一个浮点常数,它表示坐标轴零位置与电机零位置(起始位置) 之间相差的电机位置读数;如果没有给定,PMAC取该值为0.0。

注意: 在电机号和箭头符号之间不允许有容格。

备注 本命令给当前选址坐标系中的指定电机分配一组坐标轴。它也为坐标轴定义比例尺和起始偏移量。

在绝大多数的情况下,PMAC的电机和坐标轴之间是一一对应的,所以一条坐标轴 定义语句对一个电机只使用一个坐标轴名。

一个比例因子一般和坐标轴字符一起使用,以便使该轴的移动能够以标准单位(例如毫米、英寸、度)指定。这个数字指*该轴使用的是什么样的用户单位*。如果没有给出比例因子,一个单电机读数的用户单位将被用于这个轴。有时候,允许坐标轴零位置和电机起始位置不同,这时需要使用一个偏移量参数。(这是起始偏移量,

它在以后的过程中可以用多种方法来改变,包括**PSET**,{axis}=,ADIS和IDIS 命令)。

如果指定电机当时已被分配了一个其它坐标系中的坐标轴,PMAC将会拒绝执行本命令(返回一个ERR003如果I6=1或3。如果指定电机当时已被分配了一个选址坐标系中的坐标轴,那么旧的轴定义将被新值覆盖掉。

如果要撤消一个电机轴的定义,先选择它所在的坐标系,然后使用命令#{常量}->0。 要清除一个坐标系中的所有轴定义,选择坐标系,然后使用UNDEFINE命令。要清

所有坐标系中的所有轴定义,可使用UNDEFINE ALL。

对于更精密的系统,两或三个笛卡尔轴可以被定义为属于同样数目的电机的线性形

组合。这允许坐标系可以在其它事件间进行旋转和正交校正。可以指定一到三个轴(如果只指定了一个轴,它就是比上面所述更简单的一种定义)。在一个定义中指定 的所有轴都必须来自同一个笛卡尔轴组: XYZ或UVW。如果使用了这样的多轴定 义,一条移动某个轴的命令将会影响多个电机的移动。

范例 #1->X ; 用户单位=脉冲

> ; 2000 读数/用户单位 #4->2000 A #8->3333.333Z-666.667 ; 也可以使用非整数

; 可以给同一个轴分配两个电机, 此时若 #3->Y ;给出Y向移动,两个电机都会运动 #2->Y

;这使X和Y进行一个10000读数/单位 #1->8660X-5000Y ;的30°旋转。此旋转不涉及Z,但如果 #2->5000X+8660Y

#3->2000Z-6000 ;需要,也可以包括。

这个例子可以校正Y轴的1弧分垂直。

; 100000读数/英寸 #5->100000X #6->-29.1X+100000Y ;1/60的正弦和余弦

参考 轴,坐标系(设置一个坐标系)

在线命令#{常量}->, #{常量}->0, UNDEFINE, UNDEFINE ALL。

\$

功能: 电机复位。

范围: 指定电机

语法: \$

备注: 本命令使PMAC初始化被选址的电机,执行所需的任一交换定向,完整读出一个绝 对位置传感器,并将电机保持在一个闭环的零速状态。(对于一个带有增量编码器的 非换向电机,也可以使用**J/**命令。)

> 如果一个由PMAC换向电机的Ix80变量被设为0,本命令对于开机/复位后该电机 的初始化是必需的。如果Ix80为1,初始化将在开机/复位周期过程中自动执行。 在电机移动过程中,本命令将不被接受。

: 查询#1的开机模式变量值 范例 **I180**

> : PMAC返回0, 开机时不定向且停止 0

\$\$\$;将卡复位;电机保持停止状态 #1\$: 初始化电机,按需要定向和读取

绝对传感器(设置一个电机) 参考

开机定向(设置PMAC换向) I变量lx10, lx74, lx74, lx75, lx80, lx81

在线命令\$\$\$, J/

\$\$\$

功能: 将卡完全复位。

范围: 全局 语法: \$\$\$

备注: 本命令使PMAC执行卡的完全复位。**\$\$\$**的效果和PMAC开机的复位一样。此外,将INIT/脚置为低,然后再变高也可以起到同样的作用。

当跳线E51为其默认状态关(OFF)为PMAC-PC、-Lite、-VME, 开(ON)为PMAC-

STD)时,本命令就是执行一个PMAC的标准复位。在不带可选CPU部件(不带选件4A,5A或5B)的PMAC上,用上一条SAVE命令存储在永久性内存(EAROM)中,I 变量值、换算表设置和DPRAM与VME总线的地址都会被重新载入到工作内存(RAM)中。储存在电池供电的RAM中的所有信息,例如P变量和

入到工作内存(RAM)中。储存在电池供电的RAM中的所有信息,例如P变量和 Q变量值、M变量定义、运动和PCL程序,都不会被复位命令改变。

在带有可选CPU部件(选件4A,5A或5B)的PMAC上,PMAC在一个普通的复位周期中将闪烁内存中的内容复制到工作内存中,并且覆盖掉当前的旧内容。*这意味着PMAC工作内存中所有未存储到闪速内存的信息都会丢失*。前一次存储的P变量和Q变量值、M变量定义、运动和PLC程序在复位周期里都会被从闪速内存复制到RAM中。

如果跳线E51不是在默认状态(关为PMAC-PC、-Lite、-VME,开为PMAC-STD),

本命令对PMAC进行复位和重新初始化操作。在不带可选CPU部件(不带选件4A,5A或5B)的PMAC上,工厂默认的存储在固件(EPROM)中的I变量值、换算表设置和DPRAM与VME总线的地址都会被复制到工作内存(RAM)中。(存储在EAROM中的值只是不被使用,但并不会丢失。)

在带有可选CPU部件(选件4A,5A或5B)的PMAC上,PMAC会进入一个叫做"引导模式"的特殊的重新初始化模式,这种模式允许下载新的固件(该模式的详细

情况参见PMAC的PROM软件升级列表)。在这种引导模式中,只有很少的命令选项。可以向PMAC发送一个<CONTROL-R>字符来略过该模式中的下载操作。这使PMAC能够以现存的固件进入一般的操作模式中。存储在闪速内存的固件部分中的I变量、换算表设置和DPRAM与VME总线的地址的工厂默认值都会被复制到工作内存中。已经保存过的这些值不会被使用,但它们会保存在闪速内存的用户部分。

由于本命令使PMAC立即进入它的开机/复位周期,所以不会有认收字符(<ACK>或<LF>)返回给主机。

SAVE ; 将I变量保存到EAROM中

I130=80000 ; 再次改变增益

\$\$\$; 卡复位

!130 ; 查询参数的值

60000 ; PMAC报告当前值,此值是被保存的值

(使**E51**为开)

\$\$\$; 卡复位

!130 ; 查询参数的值

2000 ; PMAC报告当前值,此值是默认值

参考 复位PMAC(与PMAC对话)

PMAC的PROM软件升级列表

控制板端口INIT/输入(将PMAC与机器连接)

在线命令**\$\$\$***** I变量I5,Ix80 JPAN接口15脚

跳线 E51

功能: 全卡复位和重新初始化。

范围: 全局

语法: \$\$\$***

本命令进行卡的一个完全复位并重新初始化内存。所有的程序和其它缓冲区都被擦 备注: 除。所有的I变量、编码器转换表、VME和DPRAM选址参数都变为他们的工厂 默认值。(这些参数先前被保存的值仍然在EAROM中,而且可以用随后的一个\$\$\$ 命令载入到工作内存中。)它还会重新计算固件的校验和的参考值,并清除所有可 能已被键入的密码。

本命令不会影响M变量定义、P变量值、Q变量值和坐标轴定义。它清除缓冲和 缓冲指针使文件可以被重新发往PMAC。建议将参数和程序备份到主机磁盘上,以 便可以用它们进行恢复。PMAC执行程序具有"保存全部PMAC配置"和"恢复 全部PMAC配置"的功能,使得这一过程非常简便。

在跳线E51不在默认状态(关为PMAC-PC、-Lite、-VME,开为PMAC-STD)的 情况下,如果一个PMAC带有可选CPU部件(选件4A,5A或5B),它在接收到 本命令时会进入一个叫做"引导模式"的特殊的重新初始化模式。这种模式允许下 载新的固件(该模式的详细情况参见PMAC的PROM软件升级列表)。在这种引 导模式中,只有很少的命令选项。可以向PMAC发送一个<CONTROL-R>字符来 略过该模式中的下载操作。这使PMAC能够以现存的固件进入一般的操作模式中。

存储在闪速内存的固件部分中的I变量、换算表设置和DPRAM与VME总线的地址 的工厂默认值都会被复制到工作内存中。已经保存过的这些值不会被使用,但它们 会保存在闪速内存的用户部分。

范例 1130=60000 : 设置#1比例增益

> SAVE ; 保存到非易失的内存中 \$\$\$*** : 卡复位并重新初始化

I130 · 查询I130的值

2000 ; PMAC返回当前值, 它是默认值

\$\$\$: 卡的标准复位 **I130** ; 查询I130的值

60000 ; PMAC报告当前值,此值是被保存的值

参考 在线命令\$\$\$

PMAC的PROM软件升级列表

跳线E51

PMAC执行程序保存/恢复所有配置

%

功能: 报告被选址的坐标系的进给率修调值。

范围: 指定坐标系

语法: %

备注: 本命令使PMAC报告目前被选址的坐标系中的当前进给率修调值(时间基准)。值

100表示"实时";即移动速度和时间以指定值运行。

无论所得值的来源是什么(即使来源不是%{常量}命令),PMAC都会对本命令在作 出响应,并报告数值。

范例 % : 查询进给率修调值

> : PMAC响应: 100即实时 00

Н : 命令进给保持 % : 查询进给率修调值 0

参考 时基控制(使PMAC与外部事件同步)

I变量I10, lx93, lx94, lx95

在线命令%{常量},H

内存映象寄存器X:\$0808, X:\$08C8等

%{常量}

功能: 设置选址坐标系中的进给率修调值。

范围: 指定坐标系

语法: {常量}

位置: {常量}是一个非负的浮点数,它代表预期的进给率修调(时基)值(100表示实时)。

备注 本命令为当前选址的坐标系指定进给率修调值。 这个新指定的值变化速度由该坐标系的I变量Ix94决定。

指定坐标系的I变量Ix93必须设置成它的默认值(它使坐标系从%命令寄存器中获取时基值),这样才能让本命令发生作用。

PMAC能够实现的最大的%值为(2²³/l10)*100或(伺服更新速度(kHz))*100。如果你指定了一个大于上述值的量,PMAC将会用上述量大值代替。

如果你想用一个变量来控制时间基准,则应该为命令的时基寄存器(X:\$0806, X:\$08C6等)分配一个M变量(建议M197),然后给这个M变量赋一个变化值。

在这里赋的值应该等于预期的%值乘以(I10/100)。

范例 %0 ; 命令值为0, 停止运动

%33.333 ; 命令执行1/3的实时速度

%100 ; 命令执行实时速度 **%500** ; 命令执行一个过高的值

%500 ; 命令执行一个为 ; 查询当前值

225.88230574 ; PMAC响应, 这是允许的最大值

M197->X:\$0806, 24 : 给坐标系1的%命令寄存器分配变量

M197=P1*I100/100 ; 等于&1%(P1)

参考 时基控制(使PMAC与外部事件同步)

I变量I10, Ix93, Ix94, Ix95

在线命令%, H

内存映象寄存器X:\$0806, X:\$08C6等

&{常量}

功能: 对一个坐标系选址。

范围: 全局

语法: &{常量}

位置: {常量}是一个1到8之间的整数,它代表将被选址的坐标系号。

备注 本命令使由**{常量}**指定的坐标系成为被选址坐标系(即在线坐标系命令将在其上执行的坐标系)。选址是模态的,所以所有后续的指定坐标系的命令都会影响该坐标系,知道其它的坐标系被选址。在开机/复位后,坐标系1是被选址的坐标系。

注意:对于指定坐标系的控制板输入,可以从控制板端口用硬件同时选择另一个坐标系。该坐标系可以在PMAC中用程序里的COMMAND语句进行选 址。

即使控制板输入被I2=1禁止,主机选址的坐标系仍然能控制指示线路的到达位置、警告误差和致命误差功能。这些指示线路连接在控制板端口输出(所有版本的PMAC)和中断控制器上(PMAC-PC、PMAC-Lite、PMAC-STD)。(如果I2=0,硬件选择的坐标系控制这些线路。)

范例 &1B4R ; 坐标系1指向程序4的开始处并运行

Q ; 坐标系1推出运行的程序

&3B5R ; 坐标系3指向程序5的开始处并运行

A : 坐标系3终止程序

参考 | | | | | | | | | | |

在线命令#,#{电机号},& 程序命令ADDRESS,COMMAND

&

功能: 报告当前选址的坐标系

范围: 全局

语法: &

备注: 本命令使PMAC返回当前被主机选址坐标系的号。

注意:对于指定坐标系的控制板输入,可以从控制板端口用硬件选择另一个坐标系。该坐标系可以在PMAC中用程序里的COMMAND语句进行选址。

范例 & ;询问PMAC哪个坐标系被选址

4 ; PMAC报告坐标系4被选址

在线命令#,#{电机号},&{坐标系号} 程序命令ADDRESS,COMMAND

功能: 将程序停止在当前所执行运动的终点。

范围: 指定坐标系

语法: /

备注: 如果PMAC处于分段模式(I13>0),本命令让PMAC在当前运动的终点停止选址坐标系中的运动程序。如果PMAC不在分段模式(I13=0),/命令和Q命令的效果相同。它在最近计算出的运动的终点停止程序的运行,而该运动可能是当前运动后面的第1或2个运动。在运动的终点被终止以后,程序可以用R命令重新开始运行。其间,单个电机可以从终点开始手动进给。但在程序重新运行前,它们都必须用J=命令回到该点。如果试图从其它的点重新执行程序,将会产生一个错误(若I6=1或3,则报告为ERR017)。如果不希望从终止点继续程序的运行,在其它程序运行前应该发送一个A(中断)命令。

范例 &1B5R ; 指令坐标系开始程序5

/ ; 终止程序的运行 #1J+ ; 电机正向手动进给1 J/; 停止手动进给J=; 回到进给前的位置R; 继续执行程序5/; 终止程序的运行#2J-; 电机反向手动进给2

J/ : 进给停止

R; 试图继续执行程序5

<BELL>ERR017 ; PMAC报告错误,不在继续执行的位置上

 J=
 ; 回到进给前的位置

 R
 ; 继续执行程序5

参考 I变量I6, I13

在线命令R, J=, Q, A, /, H

?

功能: 报告电机状态。

范围: 指定电机

语法: ?

备注: 本命令使PMAC以ASCII码十六进制字(word)的形式报告电机状态。PMAC返回两个状态字12个字符。每个字符表示4个状态位(bit)。第一个字符表示第一个字的20-23位;第二个字符表示16-19位;以此类推,直到第六个字符表示0-3位。第七个字符表示第二个字的20-23位;第十二个字符表示0-3位。

某一位的值为1表示该状态为真,0表示为假。每一位的含义如下:

返回的第一个字(X:\$003D,X:\$0079等):

返回的第一个字符:

23位: *电机工作*: 当lx00为1且电机的计算被激活,该位为1。当lx00为0且电机的计算不工作,该位为0。

22位: *负向极限设置*: 当电机的实际位置小于软件的负向软限位(lx14),或者这个方向的硬件限制(+LIMn----注意!)超过时,该位为1;否则为0。如果电机未工作(第一个电机状态字的第23位为0)或被停止(第二个电机状态字的第14位为0),该位不会被修改。

21位 *正向极限设置*: 当电机的实际位置大于软件的正向软限位(lx13),或者这个方向的硬件限制(-LIMn----注意!)超过时,该位为1;否则为0。如果电机未工作(第一个电机状态字的第23位为0)或被停止(第二个电机状态字的第14位为0),该位不会被修改。

20位 *转向轮允许*: 若lx06为1且该轴上后续的位置是允许的,则此位为1。若lx06为0且该轴上后续的位置是禁止的,则此位为0。

返回的第二个字符:

19位 *电机定向*: 当lx01为1且该电机由PMAC换向,则此位为1; 当lx01为0 且该电机不是由PMAC换向,则此位为0。

18位 *开环模式*: 当电机的伺服环打开,无论输出被允许还是禁止,该位都为**1**。(参看放大允许状态位,以区分这两种状态。)当伺服环闭合时,此位为**0**(在位置控制时,输出总是允许的)。

17位 *执行定时运动*: 当电机在执行一个预先定义了终点和终止时间的运动时,此

位为1; 否则为0。这些运动包括: 所有运动程序的运动停止或延迟, 所有给定终点位置的手动进给运动, 和部分由信号触发的起始位置搜索运动。当要求的运动完成之后, 该位从1变为0。

16位 *积分模式*: 当lx34为1且伺服环积分增益只有在预期速度为0才工作时,此位为1。当lx34为0且伺服环积分增益总处于工作状态时,此位为0。

返回的第三个字符:__

15位 *停止状态*: 当电机的坐标系正在执行一个**DWELL**指令时,此位为1; 否则为0。

14位 数据块错误:如果下一个运动段的数据没有准备好,运动会被终止,此时该

位为1; 否则为0。这种错误是由于计算时间不够造成的。当另一个后续运动开始后, 此位由1变为0。该位与坐标系的运行时间错误位有关。

13位 *预期速度为零*: 当电机处于闭环控制状态且要求的速度为零时(即电机要保持位置),该位为**1**。当电机处于开环状态,或在闭环状态但要求一个非零的速度时,该位为**0**。

12位 *终止并减速*: 当电机由于一个终止命令或者由于达到硬件或软件的位置(超行程)极限而减速时,此位为1; 否则为0。当要求停止减速完成后,该位由1变为0。

返回的第四个字符:

11位

块请求: 当电机刚刚进入一个新的运动段,并正在请求计算即将进行的段时,该位为1; 否则为0。此位主要在内部使用。

10位 *起始位置搜索状态*: 当电机在处于一个等待触发的运动中时,该位为**1**。这些运动包括: 一个起始位置搜索运动,一个触发停止的手动进给运动,或者一个触发启动的运动。这些运动的计算一开始,此位就变为**1**; 而一旦接收到出发信号,

或者运动被用其它方法停止,此位就变为0。由于该位在触发以后的运动过程都是0,

所以用它并不能很好地观察是否所有的运动都完成了,而要用"返回起始位置"和" 预期速度为零"两位来代替。

8-9位 这两位储存一个指向数据块的指针,该数据块是为电机计算服务的。此位主要在内部使用。

返回的第五和第六个字符:

0-7位 这两位存储一个指向数据块的指针,该数据块是为电机计算服务的。此位主要在内部使用。

返回的第二个字(X:\$0814, X:\$08D4等):

返回的第七个字符:__

位23 *已分配坐标系*: 当电机已经通过一个轴定义语句在任意坐标系中分配了一个坐标轴时,该位为1。而当电机没有分配任意一个坐标系中的坐标轴时,该位为0。

位20-22 *(坐标系-1)号*: 这三位一起保存一个数值,它等于分配给电机的坐标系号减一。位22为高位(MSB),位20是低位(LSB)。例如,如果电机分配了一个坐标系6中的轴,这几位保存的值将为5: 位22=1,位21=0,位20=1。

返回的第八个字符:

位16-19 (为将来的应用保留)

返回的第九个字符:

位15 (为将来的应用保留)

位**14** *放大允许*:无论是开环状态还是闭环状态,当电机输出的放大被允许时,此位为**1**。(参看开环模式状态位,以区分这两种状态。)当输出被禁止(停止)时,此位为**0**。

位12-13 (为将来的应用保留)

返回的第十个字符:

位**11** *在位置极限停止*: 当电机由于一个软件或硬件(超行程)限制而停止,此位为

1, *即使造成停止的限制后来不存在了*。在其它时间里,即使存在一个限制但已经超过它,此位都为0。

位10 *返回起始位置*:此位在开机/复位时被置为0。当返回运动*成功地*定位于起始

位置的触发器后,该位变为1。这一点,电机通常及时减速到静止,或运动到lx26 确定的偏移位置。如果要进行第二次返回运动,此位在该运动开始时被置为0,直到返回运动成功地位于起始位置的触发器后,才会再次变为1。可以用预期速度为零状态位和/或已到位状态位来监测电机的运动终点。

位9 (为将来的应用保留)

位8 *定向搜索错误*:如果由PMAC换向电机的定向搜索运动由于放大错误、超行程限制或缺少检测到的运动而失败,则此位为1。如果定向搜索运动不因这些情况而失败(不作为一个成功的定向搜索的必要保证),此位为0。

返回的第十一个字符:

位7 *触发运动*:在触发手动进给运动或触发移动运动的开始,此位为1。如果在运动结束并接收到触发信号,该位变为0;如果未接收到触发信号,则该位保持为1。此位在确定运动是否成功接收到触发信号时是非常有用的。

位6 后续的整体严重误差:如果该电机由于一个被lx11和lx63设置的整体后续误差而被禁止,该位为1。后续严重误差位(位2)也会在这种情况下被设置。

位6在其它的所有情况下都为0。当电机被重新允许后该位也会重新变为0。

位5 f^2T 放大故障错误: 如果该电机由于一个整体电流错误而被禁止,该位为1。放大错误位(位3)也在此时被设置。位5在其它的所有情况下都为0。当电机被重新使能后该位也会重新变为0。

位4 后退方向标志:如果已经在负方向上进行后退,此位为1;否则为0。

返回的第十二个字符:

位3 放大错误:如果该电机由于一个放大错误信号而被禁止,该位为1,即使放大错误信号已经消失,或者电机的禁止是由I²T整体电流错误(位5也因此被设置)造成的。此位在其它的所有情况下都为0。当电机被重新使能后该位也会重新变为0。

位2 严重跟随误差:如果该电机由于超出其严重跟随误差限制(Ix11)或超出其整体错误限制(Ix63:位6也在此时被设置)而被禁止,该位为1。此位在其

它的所有情况下都为0。当电机被重新使能后该位也会重新变为0。

位1 警告跟随误差:如果电机的跟随误差超出其警告跟随误差限制(lx12),该位为1。如果电机由于严重跟随误差而被停止,此位将保持为1。在其它时间里,该位为0。当电机的跟随误差减少到低于限制,或者被重新使能(如果电机先前被停止了的话),该位会从1变为0。

位0 己到位: 当满足以下五种条件时,此位为1: 闭环状态; 预期速度为零

状态位1(要求闭环控制而且没有运动指令);程序时钟停止(当时不执行任何运动

停滞或延迟);跟随误差的量级小于lx28;经过(l7+1)次连续扫描,上四个条件均满足。

范例 #1?

: 查询电机1的状态

812000804401

- ; PMAC返回代表48位的12个十六进制数
- : 下面这些位为真(其它位为假)
- ; 字1位23: 电机工作
- ;位16:积分增益模式
- ;位13:预期速度为零
- ; 字2位23: 已分配坐标系
- ;(位20-22均为0----分配给坐标系1)
- ; 位14: 放大允许
- ;位10:返回起始位置
- ; 位0: 已到位

参考 在线命令**<CTRL-B>**, ??, ???

内存映象寄存器X:\$003D, X:\$0079等, Y:\$0814, Y:\$08D4等 推荐的M变量定义Mx30-Mx45

??

功能: 报告被选址坐标系的状态字。

范围: 指定坐标系

语法: ??

备注: 本命令使PMAC用一个ASCII码十六进制字报告选址坐标系的状态位。PMAC返回12个字符来表示两个状态字。每个字符代表4个状态位。第一个字符表示第一个字的20-23位;第二个字符表示16-19位;以此类推,直到第六个字符表示0-3位。第七个字符表示第二个字的20-23位;第十二个字符表示0-3位。

某一位的值为1表示该状态为真,0表示为假。每一位的含义如下:

返回的第一个字(X:\$0818, X:\$08D8等):

返回的第一个字符:

位23 *进给率计算中Z轴的使用*:如果该轴在基于F的运动的矢量进给率计算中被使用,则此位为1;否则为0。参见FRAX命令。

位22 Z轴增量模式:如果该轴为增量模式----运动由上一个程序点的距离确定,则此位为1。如果该轴为绝对模式----运动由终点位置而不是由距离确定,此位为0。 参见INC和ABS命令。

位21 *进给率计算中Y轴的使用*:(参见位23的说明。)

位20 Y轴增量模式: (参见位22的说明。)

返回的第二个字符:

- 位19 *进给率计算中X轴的使用*:(参见位23的说明。)
- 位18 X轴增量模式: (参见位22的说明。)
- 位17 *进给率计算中W轴的使用*: (参见位23的说明。)
- 位16 W轴增量模式: (参见位22的说明。)

返回的第三个字符:

- 位15 *进给率计算中V轴的使用*: (参见位23的说明。)
- 位14 V轴增量模式: (参见位22的说明。)
- 位13 *进给率计算中U轴的使用*: (参见位23的说明。)
- 位12 U轴增量模式: (参见位22的说明。)

返回的第四个字符:

- 位11 *进给率计算中C轴的使用*: (参见位23的说明。)
- 位10 C轴增量模式: (参见位22的说明。)
- 位9 *进给率计算中B轴的使用*:(参见位23的说明。)
- 位8 B轴增量模式: (参见位22的说明。)

返回的第五个字符:

- 位7 进给率计算中A轴的使用: (参见位23的说明。)
- 位6 A轴增量模式: (参见位22的说明。)
- 位5 半径矢量增量模式: 如果圆弧运动的矢量半径是按增量指定的(即从运动起
- 始点到圆弧中心),此位为1。如果圆运动的矢量半径是按绝对方式指定的(即从XY Z

的原点到圆弧中心),此位为0。参见INC(R)和ABS(R)命令。

位4 *连续运动请求*: 当坐标系被要求一系列的连续运动(比如**R**命令)时,此位为**1**。在其它情况下(如没有执行程序、lx92=1或者用一个**S**命令开始运行)此位为**0**。

返回的第六个字符:

- 位3 定时运动模式: 若某坐标系中的运动程序由时间指定(TM或TA),并且运
- 动速度也由此得到的,那么此位为1。而如果该坐标系中的运动由进给率(速度; F)确定的且速度由此决定,那么此位为0。
- 位2 *连续运动模式*:如果某坐标系中一系列运动在始末两点之间无停顿地连接,则此位为1。不论有什么原因,只要不是这样的运动,此位都为0。
- 位1 单步模式: 如果某坐标系中当前的程序被要求单步执行一个或一组运动,或者它接收到一个Q(退出)命令,此位为1。如果运动程序是用一个R(运行)命令开始的,或者根本就没有执行程序,此位为0。
- 位0 *正在运行程序*:如果坐标系正在执行一个运动程序,此位为1;否则为0。要注意的是:一旦计算出最后一个运动并运行到程序中的**RETURN**语句,即使电机

仍在执行计算的最后一个或两个运动,此位会由1变为0。参见电机的程序运行状态位。

返回的第二个字(X:\$0818, X:\$08D8等):

返回的第七个字符:

- 位23 *程序保持停止*: 当一个在目前选址坐标系中运行的运动程序被"\"命令从已分段运动(I13>0的LINEAR或CIRCLE模式)终止时,此位为1。
- 位22 运行时间错误: 当坐标系由于执行程序时的错误(如试图转移到一个不存在的标号,计算时间不足,等等)而终止一个运动程序时,此位为1。

位21

圆半径错误: 当一个运动程序试图完成一个距离大于两倍半径(一个大于**lx96** 值的数值)的圆弧被终止时,此位为**1**。

位20 放大错误: 当某坐标系中的任一电机由于接收到一个放大错误信号而被停止时,此位为1。当该电机被重新使能时,此位由1变为0。在其它时候该位为0。

返回的第八个字符:

- 位19 严重跟随误差:如果该坐标系中的任一电机由于超出其严重跟随误差限制(lx11)而被停止,该位为1。此位在其它的所有情况下都为0。当该电机被重新使能后此位也会重新变为0。
- 位18 警告性跟随误差:如果该坐标系中的任一电机的跟随误差超出其警告性跟随误差限制(lx12),该位为1。如果电机由于严重跟随误差而被停止,此位将保持为1.在其它时间里,此位为0。当该电机的跟随误差减少到限制以下,或者被重新使能(如果电机先前被停止了的话),该位会从1变为0。
- 位17 *已到位*: 当坐标系中的所有电机都已到位时,此位为1。这些电机必须满足五个条件: 闭环状态; 所有电机的预期速度为零; 坐标系不能有任何定时运动(即
- 使距离为0)或停滞;所有电机的跟随误差小于它们各自的lx28到位宽度;经过(I7+1)次连续扫描,上述四个条件均满足。
- 位16 转动缓冲请求:如果坐标系中的转动缓冲存在且其已接收到足够的程序行,并且在已经计算出的值之前包含至少I17行,则此位为1。一旦该位被置为1,它就会一直保持此值直到计算出的值之前的行数小于I16。"PR"命令可以用来查看当前已计算出值前的程序行数。

返回的第九个字符:

- 位15 延迟计算标志: (内部使用)。
- 位14 *块停止的终点*:指当前选址坐标系中运行的运动程序被"/"命令从一个已分段运动停止(I13>0的LINEAR或CIRCLE模式)。
- 位13 *同步M变量单次触发*: (内部使用)。
- 位12 缓冲的暂停运动: (内部使用)。

返回的第十个字符:

- 位11 *外部拐角切削补偿*: 当坐标系正在执行一个外部拐角的切削补偿时,此位为1: 否则为0。
- 位10 *切削补偿运动停止请求*:如果坐标系在切削补偿允许时执行一个运动而又接收到停止这一运动的请求,则此位为1。该位供内部使用。

位**9** *缓冲的切削补偿运动*:如果坐标系在切削补偿允许时执行一个运动而下一个运动又被计算并被存入缓冲,则此位为**1**。该位主要供内部使用。

位8 *提前手动进给运动标志*: 当坐标系中的任一电机在执行一个提前手动进给(J=命令)时,此位为1: 否则为0。

返回的第十一个字符:

位**7** *已分段运动正在进行*: 当坐标系正在以分段模式(**I13>0**)执行一个运动时,此位为**1**: 否则为**0**。该位主要供内部使用。

位6 *已分段运动加速*: 当坐标系在以分段模式(I13>0)执行一个运动并从一个停止点开始加速时,此位为1; 否则为0。该位主要供内部使用。

位5 *已分段运动停止请求*: 当坐标系以已分段模式(I13>0)执行一个运动并正向一个停止点开始减速运动时,此位为1: 否则为0。该位主要供内部使用。

位4 PVT/SPLINE运动模式: 如果坐标系正处于PVT模式或SPLINE模式,

此位为1。(如果本字的第0位为0,表示在PVT模式;如果第0位为1,表示在SPLIN E

模式。)如果坐标系处于其它运动状态(LINEAR(直线)、CIRCLE(圆)或RAPI D(快速))时,此位为0。参见后面的表。 返回的第十二个字符:

位3 左侧切削补偿:如果坐标系允许切削补偿,并且补偿在左侧(从运动方向上看),则此位为1。如果关闭补偿,或补偿在右侧,则此位为0。

位2 允许切削补偿:如果坐标系允许切削补偿,则此位为1;否则为0。

位1

逆时针圆/快速模式: 当位0为1且位4为0时,如果坐标系处于CIRCLE1(顺时针圆弧)运动状态模式,此位为0;如果坐标系处于CIRCLE2(逆时针圆弧)运动状态模式,此位为1。当位0和位4都为0时,如果坐标系处于RAPID运动模式,此位为1。其它情况此位都为0。参见后面的表。

下表总结了不同运动模式下, 位4和位1的状态:

模式	位4	位1	位0
LINEAR	0	0	0
RAPID	0	1	0
SPLINE	1	0	1
CIRCLE1	0	0	1
CIRCLE2	0	1	1
PVT	1	1	0

范例 ?? ; 查询坐标系状态字

A8002A020010

; PMAC响应,下列位为真:

; 字1位23: 进给率计算中使用Z轴

;位21:进给率计算中使用Y轴

: 位19: 进给率计算中使用X轴

: 位5: 半径矢量增量模式

;位3:定时运动模式

;位1:单步模式

;字2位17:已到位

; 位4: PVT/SPLINE模式

参考 在线命令**<CONTROL-C>**, ? , ???

???

功能: 报告全局状态字。

范围: 全局

语法: ???

备注: 本命令使PMAC用一个ASCII十六进制字报告选址坐标系的状态位。PMAC返回 12个字符来表示两个状态字。每个字符代表4个状态位。第一个字符表示第一个字的20-23位;第二个字符表示16-19位;以此类推,直到第六个字符表示0-3位。第七个字符表示第二个字的20-23位;第十二个字符表示第二个字的0-3位。

某一位的值为1表示该状态为真,0表示为假。每一位的含义如下:

返回的第一个字(X:\$0818, X:\$08D8等):

返回的第一个字符:

位23 *实时中断激活*:如果PMAC正在执行一个实时中断任务(PLC0或运动规划),此位为1。而如果PMAC在执行其它任务,则此位为0。注意:通讯不能在实时任务中进行,所以查询此位总会得到0值。该位主要供内部使用。

位22 *重新进入实时中断*:如果一个实时中断持续了太长的时间以至于当下一个实时中断到来(18+1个伺服周期之后)时,前一个中断仍在执行,此时此位为1。该位会一直保持为1,直到卡被复位或者此位被人工改为0。如果上述情况是由运动程序的计算引起的,那么并不是一个严重的问题;而如果是PLC0造成这种情况出现(没有运动程序在运行),那么有可能会很严重。

位21 *伺服激活*:如果PMAC正在执行伺服更新操作,此位为1;否则为0。注意:通讯不能在更新伺服时进行,所以查询此位总会得到0值。该位主要供内部使用。

位20 *伺服错误*:如果PMAC不能正常地完成它的伺服例程,此位为1。这是一个严重的错误情况。如果伺服例程正常完成,此位为0。

返回的第二个字符:

位19 数据采集功能允许: 当数据采集功能被激活时,此位为1: 否则为0。

位18 *伺服开始时进行数据采集*: 当设置下一个伺服周期开始数据采集时,此位为1; 否则为0。数据采集一开始此位就由1变为0。

位17 *用触发开始数据采集*: 当设置在机床输入2口(Machine Input 2)的上升沿开始数据采集时,此位为1; 否则为0。数据采集一开始此位就由1变为0。

位16 (为将来的应用保留)

返回的第三个字符:

位15 (为将来的应用保留)

位14 *丝杠螺距补偿允许*:如果PMAC中的丝杠螺距补偿被激活,则此位为1; 否则为0。

位13 *任一内存校验和错*:如果在PMAC固件或用户程序缓冲空间中检测到一个校验和错误,此位为1。注意和位12的情况区别。

位12 *PROM校验和错*:如果在PMAC内存中检测到一个固件校验和错误,此位为。如果没有检测到内存校验和错误,或者检测到一个用户程序校验和错误,此位都为0。注意和位13区别。

返回的第四个字符:

位11 *DPRAM错误*:如果PMAC在双端口RAM(DPRAM)通讯中检测到一个错误,此位为1;否则为0。

位10 *EAROM错误*:如果PMAC在从EAROM读取保存的数据时检测到一个错误(此时PMAC会用工厂默认值取代保存值),此位为1;否则为0。

位8-9 (为将来的应用保留)

返回的第五个字符:__

位7 (为将来的应用保留)

位6 *TWS变量奇偶错误*:如果最近从某设备(该设备支持奇偶校验)读取或写入的TWS格式M变量有一个奇偶校验错误,此位为1;否则为0。当该设备不支持奇偶校验时,此位的值不确定。

位5 *宏辅助通讯错误*:如果最近的宏辅助读或写命令失败,此位为1。该位在每个宏辅助读或写开始时被置为0。

位4 (为将来的应用保留)

返回的第六个字符:

位2-3 (为将来的应用保留)

位1 *选址所有卡*:如果在一个串接菊花链上的所有卡被用@@命令同时选址,此位为1;否则为0。

位0 *选址该卡*:如果该卡在一个串接菊花链上并且被用**@n**命令选址,此位为1;否则为0。

返回的第二个字(X:\$0818, X:\$08D8等):

返回的第七个字符:

位23 (内部使用)

位20-21 (内部使用)

返回的第八个字符:

位19 *运动缓冲区打开*:如果任一运动程序缓冲区(PROG或ROT)是打开的,则此位为1。若没有一个这缓冲区是打开的,则此位为0。

位18

转动缓冲区打开:如果旋转运动程序缓冲区(ROT)是打开的,则此位为1;否则为0。

位17 *PLC缓冲区打开*:如果PLC程序缓冲区是打开的,则此位为1:否则为0。

位16 *PLC命令*:如果PMAC正在处理的命令是由PLC或一个运动程序通过CMD""语句发送的,则此位为1;否则为0。该位主要供内部使用。

返回的第九个字符:

位15 VME通讯模式: 当PMAC准备通过VME总线"邮箱"端口进行通讯时,此位为1。当PMAC准备通过"主机端口"(PC总线或STD总线)或串口进行通讯时,此位为0。当PMAC通过VME总线邮箱端口接收到一个字母或数字命令时,此位从0变为1。当PMAC通过串口接收到一个<CTRL-Z>命令时,此位从1变为0。

位12-14(内部使用)

返回的第十个字符:

位11 *固定缓冲区满*: 当没有固定的运动(PROG)或PLC缓冲区打开,或者打开的缓冲区中可用空间小于I18个字时,此位为1。而当打开的缓冲区的可用空间大于I18个字时,此位为0。

位8-10 (内部使用)

返回的第十一和第十二个字符:

位0-7 (为将来的应用保留)

范例 ???

; 查询PMAC的全局状态字

A8002A020010

;PMAC返回全局状态字:

; 字1位13 (任一校验和错) 为真字1位12 (程序校验和错) 为真

;字1位12(程序校验和错)为真

; 字2位23(内部使用)为真

; 其余位为假

参考 在线命令?,??,<**CTRL-G>**

内存寄存器X:\$0003, Y:\$0003

@

功能: 报告菊花链中当前被选址的卡。

范围: 全局

语法: @

备注: 本命令使菊花链上被选址的PMAC向主机报告它的号码。如果所有的卡都被选址,

卡@0将会返回一个@字符。

要想使本命令被接受, 11必须被置为2或3。否则, 将会报告一个ERR003。

范例 。 ;询问PMAC链哪个卡被选中

4 ; PMAC@4报告它被选址

参考 选址命令(与PMAC对话)

多卡应用(使PMAC与外部事件同步)

I变量I1

在线命令#, #{常量}, &, &{常量}, @{常量}

跳线E40-E43 (PMAC-PC, -Lite, VME)

开关SW1-1到SW1-4(PMAC-STD)

功能: 执行程序进给保持(当为保持模式时,允许手动进给)。

范围: 指定坐标系

语法: \

备注: 当PMAC为分段模式(I13>0的LINEAR或CIRCLE模式)时,本命令使PMAC 在当前选址的坐标系中执行程序进给保持(当坐标系在保持模式时允许手动进给)。如果PMAC为分段模式(I13=0,或其它运动模式),\命令和H命令的效果相同----使电机按原路到达一个停止点(但当为进给保持模式时,不允许有其它的运动)。

进给保持模式中向停止点减速和随后的**R**命令的速度由I变量I52确定。该I变量是一个控制所有坐标系速度的变量。

一旦进给保持模式被停止,执行程序就可以用**R**命令重新开始。其间,每个电机可以从停止点各自进给,但在程序重新开始前必须全部用**J=**命令回到该点。任何想从

另一点开始重新运行程序的企图都会造成一个错误(若**I6=1或3**,则报告**ERR017**)。 如果不想从该点继续执行程序,应该在其它程序执行前发送一个A(终止)命令。

范例 &1B5R ; 命令坐标系1开始执行程序5

**** ; 命令程序进给保持 **#1J+** ; 电机1正向慢进给

J/ ; 停止进给(在此检查部件)

J=; 回到进给前位置R;继续执行程序5\;终止程序执行#2J-; 电机2反向慢进给

J/ ; 停止进给

R : 试图继续执行程序5

<BELL>ERR017 ; PMAC返回错误,不在继续执行的点

J= ; 回到进给前位置 **R** ; 继续执行程序**5**

参考 停止命令(使你的应用安全)

I变量I6,I13,I52,Ix95

在线命令R, J=, Q, A, /, H

Α

功能: 终止当前选址坐标系中的所有程序和运动。

范围: 指定坐标系

语法: A

备注: 本命令使当前坐标系中所有定义的轴立即开始减速直到停止,终止所有当前正在运行的运动程序(如果有的话)。它还使所有被禁止(停止)或开环的电机(在当前坐标系中定义的)进入一个允许的零速闭环状态。

如果在运动中,每个电机将以它们各自l变量lx15的速度进行减速。如果当A命令发出时有一个严重的跟随误差,那么运动停止的实际时间将会比较长。虽然要求的轨迹会按一个定义的速度停止到一点,但由于实际位置要"追上"指令位置而用一个较长时间。

注意:一个多轴系统在减速过程中可能不会停在程序预期的路径上。

中断命令不意味着可以很好地恢复,如果你希望比较容易地继续运行,可以用**H,Q**,/或\代替。

运动程序可以通过发送一个R或S命令继续运行(如果确实有一个运动程序被终止的话),但是必须注意两点。第一,除非发送一个PMATCH命令或I14=1,否则下一个运动将会从被终止运动的原始结束点作为起始点进行计算。第二,从终止位置到下一个运动终止位置的运动可能无法实现或预计。当I13为0(非分段模式)时,J=命令可以用来将坐标系中的每个电机进给到中断运动的原始终止位置。

范例 B1R ; 开始执行运动程序1

A ; 终止程序

#1J=#2J= ; 将电机进给到运动原始终止位置

R ;继续执行程序的下一个运动

参考 停止命令(使你的应用安全)

控制板端口STOP/输入(将PMAC与机器连接)

I变量I13, I14, Ix15

在线命令<CONTROL-A>, H, J/, K, Q

JPAN接口第10脚

ABS

功能: 为坐标系中的轴选择绝对位置模式。

范围: 指定坐标系

语法: ABS

ABS({轴}[, {轴}...])

位置: {轴}是一个字母(X, Y, Z, A, B, C, U, V, W), 它代表被指定的坐标轴, 也可以用字母R来指定半径矢量模式。

备注 本命令如果不带参数,则它使随后坐标系中各个轴的运动命令的所有位置都被当作绝对位置(这是默认状态)。一个带参数的**ABS**命令使被指定的轴成为绝对位置模式,而其它的轴的状态则不会改变。

如果R作为一个"轴"被指定,那么圆运动的半径矢量特性I,J和K条件就会被指

定为绝对形式(即作为一个从原点而不是从运动起始点开始的矢量)。一个不带任何参数的**ABS**命令将不会影响这一矢量特性。默认的半径矢量是增量式的。

如果本命令发送到PMAC时有一个运动程序缓冲区正打开,那么命令将会放到缓冲中并在以后执行。

范例 ABS(X,Y) ; X和Y设置为绝对模式----其它的轴

; 和半径矢量没有被改变。

ABS ; 所有轴都成为绝对模式----半径矢量

, 没有改变。

ABS(R); 半径矢量成为绝对模式----所有的轴

; 没有改变。

参考 圆弧运动(编写一个运动程序)

在线命令INC

程序命令ABS, INC

{轴}={常量}

功能: 重新定义指定的轴位置。

范围: 指定坐标系

语法: {轴}={常量}

位置: {轴}是一个字母(X, Y, Z, A, B, C, U, V, W), 它指定的是当前位置将被重新 定义的轴。

{常量}是一个浮点数,它代表坐标轴当前位置的新"名字"。

备注 本命令以用户单位(由轴定义中的比例因子确定)将当前轴位置重新定义为由**{常量}** 指定的值。它可以用来重新定位坐标系的原点。命令只是为位置分配一个新值,而不会引起指定轴的运动。

在内部,一个产生新位置偏移的位置偏置寄存器被写入。**PSET**是等效果的运动程序命令。

范例 X=0 ; 使X轴的当前位置为零

Z=5000 ; 将Z轴的位置重新定义为5000

参考 轴,坐标系(设置一个坐标系)

在线命令Z

程序命令PSET, ADIS, IDIS

B{常量}

功能: 将被选坐标系指向一个运动程序。

范围: 指定坐标系

语法: B{常量}

位置: {常量}是一个0.03到2767.99999间的浮点数,它代表坐标系要指向的程序和地址。 其中整数部分代表程序的号,小数部分乘以100,000的值代表行标号(若小数部分

为0,则代表程序的开头)。

备注 本命令使PMAC将被选坐标系的程序计数器指向指定的运动程序和地址。通常它用来将程序计数器定位到一个运动程序的开始处。下一个**R**或**S**命令将从这一点开始执行程序。

如果**{常量}**是一个整数,程序计数器将会指向号码和**{常量}**相符的程序的开始处。固定的运动程序缓冲(PROG)的号码可以从1到32,767。本命令用于旋转运动程序时程序号为0。

如果**{常量}**是一个非整数,则此数的小数部分代表指向的程序中的行标(N或O)。小数部分的值乘100,000后得到的值代表被指向的行标号(它采用补0的方法使该值成为5位十进制数)。

本命令被发送到PMAC时,如果有一个运动程序缓冲(包括旋转)是打开的 ,那么本命令就会被存入缓冲,并在将来作为一个B轴运动命令执行。

范例 B7 ; 指向程序7起始处

B0 ; 指向旋转缓冲的起始处

B12,6 : 指向程序12的标号N60000

B3, **025R** ; 指向程序3的标号N2500并执行

参考 在线命令DEFINE ROT, R, B

程序命令B{数据},N{常量},O{常量}

CLEAR

功能: 清除当前打开的缓冲。

范围: 全局

语法: CLEAR

CLR

备注: 本命令将清空当前打开的程序、PLC、旋转等缓冲区。一般来说,如果你在主机上创建了一个缓冲文件,那么你一开始就应该执行**OPEN{缓冲}**和**CLEAR**命令(即使这些行不是缓冲的技术部分),然后才是真正的内容。本命令将使你很容易地从主机上编辑缓冲区、反复下载缓冲和擦除正在使用的旧的缓冲内容。

范例 OPEN PROG 1 : 打开运动程序缓冲1

CLEAR; 清空这个缓冲F1000; 程序真正开始于此处X2500; …并在此行结束CLOSE; 关闭程序缓冲

OPEN PLC 3 CLEAR CLOSE; 清空PLC 3

参考 程序缓冲(与PMAC对话)

在线命令OPEN, CLOSE, DELETE

CLOSE

功能: 关闭当前打开的缓冲。

范围: 全局

语法: CLOSE

CLS

备注: 本命令关闭当前打开的缓冲。它应该在输入了一个运动、PLC、旋转等缓冲后立即

使用。如果缓冲仍然打开,那么后续的原本作为在线命令的语句(比如**P1=0**)也会进入缓冲。在被下载到PMAC文件的开头和结尾处使用**CLOSE**命令是一个比

较好的方法。

如果PMAC中的*任*一程序或PLC的结构不正确(例如:没有**ENDIF**或**ENDWHILE**

和一个IF或WHILE相匹配),则PMAC会对CLOSE命令为任意缓冲返回一个

ERR003,直到该问题被解决。

范例 CLOSE ; 这确保所有缓冲都被关闭

OPEN PROG 1 : 打开运动程序缓冲1

 CLEAR
 ; 清空这个缓冲

 F1000
 ; 程序在此真正开始...

 X2500
 ; …并在此结束

 CLOSE
 ; 关闭程序缓冲

LIST PROG 1 ; 查询被关闭的程序的清单

F1000 ; PMAC开始列写

X2500

RETURN ; 这是由CLOSE命令附加上去的

参考 程序缓冲(与PMAC对话)

在线命令OPEN, CLEAR, <CTRL-L>, <CTRL-U>

{常量}

功能: 为变量P0或表入口赋值。

范围: 全局

语法: {常量}

位置: {常量}是一个浮点数值。

备注 本命令的功能和**P0={常量}**相同,即将一个在命令行键入的值赋给P变量P0。它

可以通过一个哑终端设备建立数值的简单操作入口。PMAC的用户程序必须正确地使用P0。

注意:如果PMAC中的一个特殊表(比如补偿、激励)已经被定义但没有被填充,那么本命令使常量值填入此表而不是PO。

范例 在一个运动程序中:

P0=-1 ; 将P0设置为一个"非法"的值

SEND"Enter number of parts in run:"

; 提示哑终端上的操作者 ; 操作者只需键入数值

WHILE (P0<1) WAIT ; 等待直到获得合法的响应

 P1=0
 ; 初始化局部计数器

 WHILE (P0<P1)</td>
 ; 每个部分循环一次

P1=P1+1

-

参考 在线命令OPEN COMP, OPEN STIMULUS, P{常量}={表达式}

DATE

功能: 报告程序固件的修正日期。

范围: 全局

语法: DATE

DAT

备注 本命令使PMAC报告它所使用的程序固件修正的日期。该日期是按美国格式报告

的: mm/dd/yy(月/日/年)。

范例 DATE ; 向PMAC查询固件修正日期。

07/22/92 ; PMAC返回: 1992年7月22日

参考 在线命令VERSION, TYPE

DEFINE BLCOMP

功能: 定义间隙补偿表。

范围: 指定电机

语法: DEFINE BLCOMP {补偿点数}, {脉冲长度}

DEF BLCOMP, {脉冲长度}

位置: {记入项目}是一个正整数,它表示表中数值的个数。

(脉冲长度)是一个正整数,它表示表中电机的编码计数的空间。

备注 本命令为被选址电机建立一个间隙补偿表。下一个被发送到PMAC的{补偿点数}常

量将被放入该表。命令行中的最后一项**{计数长度}**代表电机编码计数的间隙补偿表的空间。在使用中,如果电机位置在0到计数长度的范围之外,在计算补偿前该位置会被"翻转"到范围之内。表中补偿点间的距离为**{脉冲长度}**被**{补偿点数}**除的

结果。

接下去的几行是输入表的真正补偿点数,它们是用空格或回车符隔开的常数。这些补偿点的单位是计数的1/16,而且这些项目必须是整数值。第一个点是离电机零位置第一间距的校正值(由最近的起始位置搜索运动或启动/复位确定),第二个点

是

第两间距的校正值,以此类推。表中定义的电机零位置的校正值为零。

当电机向负方向运动校正位置时,校正值是从命令的标称位置值中*减掉*的量(在负方向上加)。在调整电机位置前,从补偿表中得到的校正值被加到lx86的"恒定"的间隙参数中。对于任意丝杠补偿表中的校正值,如果此值把该电机作为目标电机

则它在两个方向上都总是激活的。

表中最后的一个补偿点表示的是离电机零位置{count length}距离的校正值。因为补偿表可以翻转,该补偿点也代表电机零位置上的校正值。因此,最近的补偿点在实际中将总被赋为零。

如果一个号码较低的电机有任一的BLCOMP缓冲,或者存在任一TBUF, ROTARY及GATHER缓冲,PMAC都会拒绝本命令,并报告一个ERR003 (若l6=1或3)。

I51必须置为1才能使本表被激活。

范例

参考 间隙补偿(设置一个电机)

丝杠补偿(设置一个电机)

I变量I99, Ix85, Ix86

在线命令DEFINE COMP, DELETE BLCOMP

DEFINE COMP(一维)

功能: 定义丝杠螺距补偿表。

范围: 指定电机

语法: DEFINE COMP {补偿点数}, [#{源电机}, [#{目标电机},]] {脉冲长度}

位置: {补偿点}是一个正整常数,它表示表中补偿点的个数。

{源电机}(可选)是一个1到8间的常数,它代表一个电机,在表中该电机的位置控制对应的记录将被用作激活的校正值。如果没有指定,PMAC将会把选址的电机作为源电机。

{目标电机}(可选)是一个1到8间的常数,它代表接收校正值的电机。如果没有指定目标,则PMAC会把选址电机作为目标电机。

{脉冲长度}是一个正整数,它代表表中的源电机编码统计的空间。

备注 本命令为选址电机建立并分配一个丝杠螺距补偿表。接下来发送到PMAC的**{补偿 点数}**常量被放到表中。定义以后,这个表可以用变量I51来禁止或激活。

该表"属于"当前被选址的电机。并且如果没有在命令行中指定,它将选址电机作为源位置数据和它的校正目标。每个电机只能拥有一个属于它的补偿表(在一个PMAC中总共有8个表),但是它可以作为多个电机的源或目标。

如果一个号码较低的电机有任一的BLCOMP缓冲,或者存在任一TBUF,ROTARY及GATHER缓冲,PMAC会拒绝本命令,并报告一个ERR003(若16=1或3)。任何一个这样的缓冲都必须先删除。COMP缓冲必须从大号码电机向小号码电机定义,而从小号码电机向大号码电机删除。

在本命令中,可以直接制订一个源电机(用**#{来源}**),或源电机和目标电机(用**# {来**

源},#{目标})。这些电机都可以不是补偿表所属的电机。(换句话说,一个表属于

某电机并不一定意味着它影响该电机的位置或被电机位置影响。)

命令行中最后一项是**{脉冲长度}**,它指定了补偿表的源电机编码统计的空间。在应用中,如果源电机的位置处于0到记录长度的范围之外,在计算校正值之前源电机位置会被"翻转"到该范围之内。表中补偿点间的距离为**{脉冲长度}**除以**{补偿点数**

, 的值。

在后面的行中给出的是表中的实际补偿点。这些补偿点的单位是脉冲的1/16,并且必须为整数值。第一个项目是离电机零位置一间距的校正值(由最近的起始位置搜索运动或启动/复位确定),第二个项目是两间距的校正值,以此类推。校正值是

了获得校正位置而向命令的标称位置中*加入*的量。表中定义的电机零位置的校正值为零。

表中最后一个补偿点表示的是离电机零位置{count length}距离的校正值。因为补偿表可以翻转,该补偿点也代表电机零位置上的校正值。由于这个原因,最近的补偿点在实际中将总被赋为零。

范例 #1 DEFINE COMP 4,2000 ; 为电机1创建表

ERR003 ; PMAC拒绝该命令

DELETE GATHER ; 清除其它缓冲以便能够载入

DELETE ROTARY

#8DEFINE COMP 500,20000 ; 将电机8作为源电机和目标电机 ,

;500补偿点,点间距为40脉冲

#7DEFINE COMP 256,#3,32768 ; 从属于电机7,将电机3作为

;源电机,电机7作为目标电机 ;256个补偿点,128个点间距

#6 DEFINE COMP 400.#5.#4.40000: 从属于电机6,将#5作为

;源电机,#4作为目标电机

;400个补偿点,100个点间距

#5 DEFINE COMP 200,#1,#1,30000 ; 从属于电机5,

;将#1作为来源和目标电机, ;200个补偿点,150个点间距

I51=1 ; 使能 (Enable) 补偿表

参考 丝杠螺距补偿(设置一个电机)

I变量I51

在线命令{常量}, LIST COMP, LIST COMP DEF, DELETE COMP, DELETE

GATHER, DELETE ROTARY, SIZE

DEFINE COMP(二维)

功能: 定义二维丝杠螺距补偿表。

范围: 指定电机

语法: DEFINE COMP {源1补偿点数} . {源2补偿点}, #{源电机1}, [#{源电机2}, [#{目

标}]], {长度1}, {长度2}

DEF COMP...

位置: {**补偿点1**}是一个正整常数,代表第一个来源电机的指定位置的补偿点数。

{补偿点2}是一个正整常数,代表第二个来源电机的指定位置的补偿点数。

{源电机1}是一个从1到8的整常数,代表第一个来源电机的号码。

{源电机2}是一个从1到8的整常数,代表第二个来源电机的号码。

{目标}是一个从1到8的整常数,代表目标电机的号码。它的默认值是选址电机。

{长度1}是表的长度,在计算中沿第一个电机的运动。

{长度2}是表的长度,在计算中沿第二个电机的运动。

备注 本命令为选址电机建立并分配一个二维的补偿表。接下去发送到PMAC的(源1补 *偿点数1+1)*(补偿点2+1)-1*个常数将被放到这个表里。这类表作为两电机组成的

平面位置(如X和Y轴)的一种功能,通常用来校正第三个电机的位置(如Z轴)。

在定义以后,这个表可以用变量I51来禁止或使能。

如果一个号码较低的电机有任一的COMP缓冲,或者存在任一TCOMP,B LCOMP, TBUF, ROTARY及GATHER缓冲, PMAC都会拒绝本命令,并 报告一个ERR003(若I6=1或3)。任何一个这样的缓冲都必须先删除。C OMP缓冲必须从大号码电机向小号码电机定义,而从小号码电机向大号码 申机删除。

第一个源电机必须在命令行中用#{源1}指定,第二个来源电机可以在命令行中用 #{源2}指定。如果未被指定,PMAC认为当前选址电机就是第二源电机。目标电 机可以在命令行中用#{目标}指定。如果未被指定,PMAC认为当前选址电机就是 目标电机。

换句话说,如果在命令行中只指定了一个电机,那么该电机就是第一源电机,而第 二源电机和目标电机都按默认值设为选址电机。如果在命令行中指定了二个电机, 那么其中的第一个就是第一源电机,第二个是第二源电机,而目标电机则为默认的 选址电机。如果指定了三个电机,那么其中的第一个是第一来源电机,第二个是第 二源电机,第三个是目标电机。三个电机之中不要求必须有选址电机。

命令行中最后的两项,{长度1}和{长度2},分别以编码统计的形式指定了第一和第 二源电机的补偿表间的距离。在应用中,如果源电机的位置处于0到脉冲长度的范 围之外,在计算校正值之前源位置会被"翻转"到该范围之内。表中补偿点间的距 离为**{脉冲长度}**除以**{补偿点数}**的值。

在后面的行中给出的是表中的实际补偿点。这些补偿点的单位是脉冲的1/16,并且 必须为整数值。第一个点是离第一源电机零位置一间距的校正值,也是第二源电机 零位置的校正值(由最近的起始位置搜索运动或启动/复位确定)。第二个点是第

源电机两间距和第二源电机零位置的校正值。以此类推,直到第{补偿点1}个项目 是离第一源电机{长度1}间距和第二来源电机零位置的校正值(如果想和初始位置 暗含的零校正值相对应, 你要沿边缘使用该表, 该补偿点应该为零)。

下一个补偿值在第一源电机零位置和离第二源电机零位置一个间距处。紧接着的补 偿点按如下位置顺序排列:从第一源电机到第2*{补偿点1}+1项(该项在距第一源 电机{长度1}, 距第二源电机零位置一间距)。这个序列在表中一行一行地重复,

后以第(**{补偿点1}+1**)*(**{补偿点2}+1**)-1项结束,该项位于(**{长度1}**,**{长度2}**)----如 果想和初始位置暗含的零校正值相对应、你要沿边缘使用该表、该补偿点应该为零

范例 #1 DEFINE COMP 30.40, #1, #2, #3, 300000, 400000

; 创建属于电机1的表

E RR007 ; PMAC拒绝该命令

DELETE GATHER ;清除其它缓冲以允许载入

&1 DELETE ROTARY

&2 DELETE ROTARY

#2 DELETE COMP

#3 DELETE COMP

DELETE TRACE

#4 DEFINE COMP 30.40, #1, #2, #3, 300000, 400000

: 创建相同的表, 现在属于

; 电机4----#1和#2是源, #3是

;目标:30x40,10,000个计数空间

(1270记录项) ; (30+1)*(40+1)-1个常数补偿点

#3 DEFINE COMP 20.25, #2, #3, #1, 200000, 250000

; 创建属于电机3的表----#2 ; 和#3是源, #1是目标; 20x25,

; 10,000个计数空间

(545记录项) ; (20+1)*(25+1)-1个常数补偿点

#2 DEFINE COMP 10.10, #1, #4, 10000, 20000

; 创建属于电机2的表----#1 ; 和#4是源, #4(默认)是

;目标;10x10,1000&2000个计数空间

(120记录项) ; (10+1)*(10+1)-1个常数补偿点

#1 DEFINE COMP 10.12, #4, 1280, 1200

; 创建属于电机1的表----#4 ; 和#1(默认)是源,#1

; (默认)是目标; 10x12, 128&100个计

; 数空间

(142记录项) ; (10+1)*(10+1)-1个常数补偿点

I51=1

参考 丝杠螺距补偿(设置一个电机)

|变量||51

再现命令{常量}, LIST COMP, LIST COMP DEF, DELETE COMP, DELETE

GATHER, DELETE ROTARY, DELETE TRACE, SIZE

DEFINE GATHER

功能: 创建一个数据采集缓冲区

范围: 全局

语法: DEFINE GATHER [{常量}]

DEF GAT [{常量}]

位置: {常量}是一个正整数,代表缓冲区保留的内存字数。

备注 本命令在PMAC的内存或DPRAM(取决于I45的设置)中为数据缓冲保留空间,

并准备好在其起始处采集数据。如果已经有一个数据采集缓冲存在,那么该缓冲的内容并不会被擦处。但"数据采集缓冲存储地址"(Y:\$0F20)将被重新初始化为

"数据采集缓冲开始地址" (X:\$0F20)。而且LIST

GATHER命令也将不再起作用。

当GATHER命令发出时,数据采集将从缓冲起始处再次开始。如果数据采集正在进行(一个ENDGATHER命令没有发出和采集缓冲没有填满),PMAC会拒绝本命令并报告一个错误。

可选项**{常量}**指定了为数据采集缓冲保留的长字的数目,而将剩余的PMAC内存留作其它缓冲(如运动程序和PLC程序)。如果没有指定**{常量}**的值,所有未使用的PMAC内存都为数据采集保留。在该缓冲被删除(用**DELETE GATHER**命令)之前,新的运动或PLC程序不能被输入到PMAC中。

注意:如果145=2或3,那么DEFINE GATHER {size}命令中所要求的{size}就指向一个DPRAM长字(32位)。如果{size}小于执行一个数据采集所需内存大小的倍数,那么实际的数据保存区将会超出所要求的{size}而变为稍大的放置数据所需储存字大小的倍数。举例来说,如果你要采集一个24位和一个48位的数据,那么就需要3个DPR AM长字内存。而你指定的{size}为4000个字(不是3的倍数),那实际的

存储区大小将会是4002个字(稍大的3的倍数)。

可以用SIZE命令来查看未用的缓冲内存中未用的长字的数目

范例 DEFINE GATHER
DEFINE GATHER 1000

在线命令GATHER, DELETE GATHER, <CTRL-G>, SIZE

DEFINE ROTARY

功能: 定义一个旋转运动程序缓冲区。

范围: 指定坐标系

语法: DEFINE ROTARY {常量}

DEF ROT {常量}

位置: {常量}是一个正整数,代表为缓冲保留的内存长字的数目。

备注 本命令使PMAC为当前选址的坐标系统创建一个旋转命令程序缓冲区,并为它分

配指定数目长字的内存。旋转缓冲允许在程序执行过程中下载程序行。

缓冲应该足够大,以便使它能够安全地存放一定数量的程序行(这些行是你在执行

前预先下载到PMAC中的)。每个内存中的长字能够存放一个运动程序的子块(例如:X1000 Y1000被认为是两个子块)。为坐标系分配的旋转缓冲区,用DELETE

ROT删除前会一直保留。

如果一个号码较低的坐标系有任一的ROTARY缓冲,或者存在一个GATHE R缓冲,PMAC都会拒绝本命令,并报告一个ERR003(若I6=1或3)。任何 一个这样的缓冲都必须先删除。ROTARY缓冲必须从大号码坐标系向小号

码坐标系定义,而从小号码坐标系向大号码坐标系删除。

范例 DELETE GATHER ; 确认打开内存

 &2DEFINE ROT 100
 ; 为坐标系2创建缓冲

 &1DEFINE ROT 100
 ; 为坐标系1创建缓冲

&1B0 &2B0 ; 指向这些缓冲

OPEN ROT ; 打开这些缓冲以便输入

: 在这里输入程序行

参考 旋转程序缓冲(编写一个运动程序)

在线命令OPEN ROTARY, DELETE ROTARY, DELETE GATHER

DEFINE TBUF

功能: 为笛卡儿轴转换创建一个缓冲。

范围: 全局

语法: DEFINE TBUF {常量}

DEF TBUF {常量}

位置: {常量}是一个正整数,代表创建的笛卡儿轴数目。

备注: 本命令在PMAC的内存中为一个或更多的轴转换保留空间。这些目标能够用于任

意坐标轴的X、Y和Z轴的实时移动、旋转、放大和镜像。一个坐标系统用TSELn 命令来选择使用哪一个母表,其中n是一个从1到创建的转换坐标数之间的整数。

如果存在任意的ROTARY或GATHER缓冲,PMAC都会拒绝本命令,并报告一个ERR003(若I6=1或3)。任何一个这样的缓冲都必须先删除。

范例 DELETE GATHER
DEF TBUF 1

DEFINE TBUF 8

参考 轴变换母表(编写一个运动程序)

在线命令DELETE TBUF, DELETE GATHER, SIZE 程序命令TSEL, ADIS, AROT, IDIS, IROT, TINIT

DEFINE TCOMP

功能: 定义一个扭矩补偿表。

范围: 指定电机

语法: DEFINE TCOMP {记录项}, {脉冲长度}

DEF TCOMP {记录项}, {脉冲长度}

位置: {补偿点}是一个正整常数,代表表中的值的数目。

{记录长度}是一个正整数,代表补偿表的电机编码统计的空间。。

备注 本命令为选址电机建立一个扭矩补偿表。接下来被发送到PMAC的{记入项目}常量

会被放到表中。命令行中的最后一项**{脉冲长度}**指定了补偿表的来源电机编码统计的空间。在应用中,如果来源电机的位置处于0到脉冲长度的范围之外,在计算校正值之前源位置会被"翻转"到该范围之内。表中补偿点间的距离为**{脉冲长度}**除

以{记录项目}的值。

接下来的行将给出补偿表的实际记录项,它们都是用空格或回车符分隔开的常数。 这些记录项的单位是一个16位DAC(即使实际的输出设备采用其它的分辨率), 而且补偿点必须是整数值。第一个项目是离第一源电机零位置一间距的校正值,也 是第二源电机零位置的校正值(由最近的起始位置搜索运动或启动/复位确定)。

第

二个项目是第一源电机两间距和第二源电机零位置的校正值,以此类推。PMAC用相邻记录项间的一个一级插值来计算表中补偿点间位置的校正值。表中定义的电机零位置的校正值为零。

校正值是该位置上加到PMAC伺服环输出的值。如果PMAC的命令是正的,那么表中的一个正值将会增加输出值。如果PMAC的命令是负的,一个表中的正值将会在负方向上增加输出值;一个负值将会减小输出的量值。

表中的最后一个补偿点代表距电机零位置距离为**{脉冲长度}**的校正值。因为补偿表可以翻转,该补偿点也代表电机零位置上的校正值。由于这个原因,最近的补偿点在实际中将总被赋为零。

如果一个号码较低的电机有任一的TCOMP缓冲,或者存在任一BLCOMP,TBUF,ROTARY及GATHER缓冲、PMAC都会拒绝本命令,并报告一个ERR003(若l6=1或3)。任何一个这样的缓冲都必须先删除。TCOMP缓冲必须从大号码电机向小号码电机定义,而从小号码电机向大号码电机删除。

151必须置为1才能允许本表。

范例

参考 扭矩补偿(设置一个电机)

I变量I51

在线命令DELETE TCOMP

DEFINE UBUFFER

功能: 为用户变量应用创建一个缓冲

范围: 全局

语法: DEFINE UBUFFER {常量}

DEF UBUF {常量}

位置: {**常量**}是一个正整数,它代表在PMAC内存中保留的48位字的数目。

备注 本命令在PMAC内存中为用户的任意应用保留空间。这部分内存空间将不会被其它任意的PMAC自动功能所使用。用户通过M变量或在线W(写)和R(读)命

令访问该缓冲。

本缓冲从PMAC内存地址\$9FFF处开始,向内存起始端(\$0000)增加,直到满足由**{常量}**指定的长(48位)字的数目。这一内存空间可以按用户认为合适的方式进行分割。在由电池备份的PMAC上,关机时缓冲中的值将会被保留到下次开机。在由闪存(flash)备份的PMAC上,前一次用SAVE命令保存的缓冲内的值将会在开机或复位时从闪存中拷贝到缓冲中。

要使PMAC接受本命令,必须先删除除固定运动程序(PROG)和PLC程序以外的所有缓冲。接受本命令,PMAC内存中不能有旋转运动程序、丝杠补偿表、变换母表、数据采集或激励缓冲(任一用DEFINE命令创建的缓冲)。最好在发送DEFINE

UBUFFER命令前用\$\$\$***命令将卡重新初始化。

未使用的内存的结尾端地址保存在寄存器X:\$0F3F中。该寄存器必须保存在地址\$A000中,表示没有定义的缓冲,从而使PMAC可以创建一个用户缓冲。一旦用户缓冲被成功定义,该寄存器将会保存缓冲的起始地址(用户缓冲的结束地址永远是\$9FFF)。然而如果其它的缓冲已被定义,未使用内存的结束地址将不会和用户缓冲的起始地址相同。

要释放该内存区域给其它用途,应该使用DEFINE

UBUFFER

0命令(没有DELETE

UBUFFER命令)。用\$\$\$***命令重新初始化板卡可能更简便一些。

范例 RHX:\$0F3F : 查找未使用的内存的结尾

 008A3D
 ; 返回值说明一些内存被保留

 \$\$\$****
 ; 重新初始化卡以清空内存

 RHX:\$0F3F
 ; 检查未使用的内存的结尾

 00A000
 ; 返回值说明没有内存被保留

 DEFINE UBUFFER 256
 ; 为缓冲保留内存

 RHX:\$0F3F
 ; 检查缓冲的起始端

009F00; 返回确认有256个字被保留M1000->D:\$9F00; 为第一个字定义M变量M1010->Y:\$9F80,12,1; 为第二个字定义M变量M1023->X:\$9FFF,24,S; 为第三个字定义M变量

参考 用户缓冲,M变量(计算的特点)

在线命令\$\$\$***, R[H]{地址}, W{地址}

DELETE BLCOMP

功能: 删除间隙补偿表。

范围: 指定电机

语法: DELETE BLCOMP

DEL BLCOMP

备注: 本命令使PMAC为选址电机删除补偿表,为其它应用释放该内存。

如果一个号码较低的电机有任一的BLCOMP缓冲,或者存在任一TBUF,ROTARY及GATHER缓冲,PMAC都会拒绝本命令,并报告一个ERR003(若16=1或3)。任何一个这样的缓冲都必须先删除。BLCOMP缓冲必须从大号码电机向小号码电机定义,而从小号码电机向大号码电机删除。

范例 #2 DEL BLCOMP ; 删除电机2的补偿表

ERR003 ; PMAC拒绝该命令

 #1 DEL BLCOMP
 ; 删除属于电机1的补偿表

 #2 DEL BLCOMP
 ; 删除属于电机2的补偿表

参考 间隙补偿(设置一个电机)

I变量I99,Ix85,Ix86 在线命令**DEFINE BLCOMP**

DELETE COMP

功能: 删除丝杠补偿表。

范围: 指定电机

语法: DELETE COMP

DEL COMP

备注: 本命令使PMAC为选址电机删除补偿表,为其它应用释放该内存。

如果一个号码较低的电机有任一的COMP缓冲,或者存在任一TCOMP、BL COMP、TBUF、ROTARY及GATHER缓冲、PMAC都会拒绝本命令、并报告一个ERR003(若16=1或3)。任何一个这样的缓冲都必须先删除。COM P缓冲必须从大号码电机向小号码电机定义,而从小号码电机向大号码电机 删除。

要注意一个*属于*电机的补偿表不一定影响电机或受电机影响。**LIST COMP DEF**命

令将会显示该表影响哪一个电机和被哪个电机影响。

范例 #2DEL COMP ; 擦除属于电机2的表

ERR003 : PMAC拒绝该命令

#1 DELETE COMP ; 擦除属于电机1的表

#2 DELETE COMP ; 擦除属于电机2的表

参考 丝杠补偿(设置一个电机)

I变量I51

在线命令{常量}, LIST COMP, LIST COMP DEF, DEFINE COMP

DELETE GATHER

功能: 删除数据采集缓冲。

范围: 全局

语法: DELETE GATHER

DEL GAT

备注: 本命令使数据采集缓冲被删除。被保留的内存现在被释放并可以被其它缓冲使用(

运

动程序、PLC程序、补偿表等)。如果数据采集正在进行(没有发送一个ENDGAT

HER

命令且采集缓冲没有被填满), PMAC在接收到本命令时会报告一个错误。

当PMAC的执行程序从PMAC上载一个缓冲到它自己的编辑区中时,它会自动把本命令插入到一个文件的头部。这样,接下去的下载将不会被已存在的采集缓冲所阻碍。在此强烈建议当你在编辑器中创建一个程序文件时使用本命令(参见下面的范例)

当执行程序的数据采集功能工作时,它自动将整个打开的缓冲为采集的数据保留。当这些完成之后,其它的附加程序或程序行都不能被输入到PMA C的缓冲空间,除非用DELETE GATHER命令释放这些内存。

范例 CLOSE ; 确认没有缓冲打开 DELETE GATHER ; 释放内存

OPEN PROG 50 ; 为输入打开新缓冲

 CLEAR
 ;擦除缓冲的内容

 ...
 ;在这里输入新的内容

参考 缓冲命令(与PMAC对话)

在线命令GATHER, DEFINE GATHER, SIZE

DELETE PLCC

功能: 删除指定的编译的PLC程序。

范围: 全局

语法: DELETE PLCC {常量}
DEL PLCC {常量}

位置: {常量}是一个从0到31的整数,代表程序的号码。

备注 本命令使PMAC删除编译的PLC程序。要记住的是,由于所有编译完的PLC程

序必须一起下载到PMAC,所以恢复这个PLC的唯一方法就是下载整套的PLC。 另外要注意的是,一个命令中只能删除一个PLCC程序。PLCC程序号码的排列和

列表清单在本命令中都不允许使用。

如果要对一个未编译的PLC程序进行相同的操作,则要按OPEN PLC n CLEAR

CLOSE的顺序进行。

参考 编译完的PLC(编写一个PLC程序)

I变量I5

在线命令DISABLE PLCC,ENABLE PLCC,CLEAR

DELETE ROTARY

功能: 删除选址坐标系的旋转运动程序缓冲区。

范围: 指定坐标系

语法: DELETE ROTARY

DEL ROT

备注: 本命令使PMAC删除当前选址坐标系的旋转缓冲区,并释放为其分配的内存。

如果该坐标系的ROTARY缓冲正打开或执行,或一个号码较低的坐标系有任一的COMP缓冲,或者存在任一TCOMP,BLCOMP,TBUF,ROTARY及GATHER缓冲,PMAC都会拒绝本命令,并报告一个ERR003(若16=1或3)。任何一个这样的缓冲都必须先删除。COMP缓冲必须从大号码坐标系向小号码坐标系定义,而从小号码坐标系向大号码坐标系删除。

范例 &2 DELETE ROTARY ; 试图删除坐标系2的旋转缓冲区

ERR003 : PMAC拒绝命令,坐标系1仍然有一个旋转缓冲

&1 DELETE ROTARY ; 删除坐标系1的旋转缓冲 **&2 DELETE ROTARY** ; 删除坐标系2的旋转缓冲

参考 旋转程序缓冲区(编写一个运动程序)

在线命令DEFINE ROTARY, OPEN ROTARY

DELETE TBUF

功能: 除轴传输矩阵的缓冲区。

范围: 全局

语法: DELETE TBUF DEL TBUF

备注: 本命令释放PMAC中被用于轴传输矩阵的空间。这些矩阵可以用于任意坐标系的X、

Y和Z轴的实时移动、旋转和镜像。

如果存在任一的ROTARY或GATHER缓冲,PMAC都会拒绝本命令,并报告一个ERR003(若I6=1或3)。任何一个这样的缓冲都必须先删除。

范例 DEL TBUF DELETE TBUF

参考 轴传输矩阵(编写一个运动程序)

在线命令DEFINE TBUF

程序命令TSEL, ADIS, AROT, IDIS, IROT, TINIT

DELETE TCOMP

功能: 删除扭矩补偿表。

范围: 指定电机

语法: DELETE TCOMP DEL TCOMP

备注: 本命令使PMAC删除选址电机的扭矩补偿表,为其它应用释放内存。

如果一个号码较低的电机有任一的TCOMP缓冲区,或者存在任一BLCOMP,TBUF,ROTARY及GATHER缓冲,PMAC都会拒绝本命令,并报告一个ERR003(若I6=1或3)。任何一个这样的缓冲都必须先删除。TCOMP缓冲必须从大号码电机向小号码电机定义,而从小号码电机向大号码电机删除。

范例 #2 DEL TCOMP ; 删除属于电机2的表 *ERR003* ; PMAC拒绝该命令

#1 DEL TCOMP ; 删除属于电机1的表

#2 DEL TCOMP ; 删除属于电机2的表

参考 扭矩补偿(设置一个电机)

I变量I51

在线命令DEFINE TCOMP

DELETE TRACE

功能: 原先用来擦除运动程序轨迹缓冲区。

范围: 全局

语法: DELETE TRACE DEL TRAC

备注 注意: PMAC中的轨迹(TRACE)缓冲区性能已经被去掉了。**DELETE TRACE**仍

然是一个合法的命令,并且当它被发送到PMAC时不会造成错误,但是它不会产生

任何的动作。

范例 CLOSE ; 确认没有缓冲被打开

DELETE GATHER DELETE TRACE ; 释放内存 OPEN PLC 17 ; 为输入打开新缓冲 CLEAR ; 擦除缓冲的内容 ... ; 在这里输入新内容

参考 在线命令DELETE GATHER

DISABLE PLC

功能: 禁止指定的PLC程序。

范围: 全局

语法: DISABLE PLC {常量} [, {常量}]

DIS PLC {常量} [, {常量}]
DISABLE PLC {常量} ..{常量}
DIS PLC {常量}..{常量}

这里

{常量}是一个从**0**到**31**的整数,它代表程序的号码。

备注 本命令使PMAC禁止(停止执行)指定的一个(或多个)PLC程序。PLC程序使用号码指定的。它们可以在命令中以单个、列表清单(用逗号分隔)或连续排列的

形式指定。PLC程序可以通过ENABLE PLC命令重新允许。

当本命令被发送到PMAC时,如果一个运动或PLC程序缓冲区正打开,则该命令将会被输入到缓冲区中以便稍后执行。

范例 DISABLE PLC 1

DIS PLC 5

DIS PLC 3, 4, 7 DISABLE PLC 0..31

参考 | | | | | | | | | | | | | |

在线命令ENABLE PLC,OPEN PLC,DISABLE PLCC,ENABLE PLCC,</br> **<CONTROL-D>**

程序命令DISABLE PLC, ENABLE PLC, DISABLE PLCC, ENABLE PLCC

DISABLE PLCC

功能: 禁止已编译的PLC程序。

范围: 全局

语法: DISABLE PLCC {常量} [, {常量}]

DIS PLCC {常量} [, {常量}]
DISABLE PLCC {常量}..{常量}
DIS PLCC {常量}..{常量}

这里

{常量}是一个从**0**到**31**的整数,它代表程序的号码。

备注 本命令使PMAC禁止(停止执行)指定的一个(或多个)已编译PLC程序。已编译的PLC程序使用号码指定的。它们可以在命令中以单个(不常见)、列表清单(用逗号分隔)或连续排列的形式指定。已编译的PLC程序可以通过ENABLE PLCC命令重新传能。

当本命令被发送到PMAC时,如果一个运动或PLC程序缓冲正打开,则该命令将会被输入到缓冲中以便稍后执行。

范例 DISABLE PLC C1

DIS PLC C5

DIS PLC C3, 4, 7

DISABLE PLC C0..31

参考 | | | | | | | | | | | | | |

在线命令ENABLE PLC,OPEN PLC,DISABLE PLCC,ENABLE PLCC <CONTROL-D>

程序命令DISABLE PLC,ENABLE PLC,DISABLE PLCC,ENABLE PLCC

ENABLE PLC

功能: 指定的PLC程序使能。

范围: 全局

语法: **ENABLE PLC** {常量} [, {常量}]

ENA PLC {常量} [, {常量}] ENABLE PLC {常量}..{常量} ENA PLC {常量}..{常量}

这里

{常量}是一个从**0**到31的整数,它代表程序的号码。

备注 本命令使PMAC(开始执行)使已编译的一个(或多个)PLC程序。PLC程序使用 号码指定的。它们可以在命令中以单个、列表清单(用逗号分隔)或连续排列的形 式指定。

当本命令被发送到PMAC时,如果一个运动或PLC程序缓冲区正打开,则该命令将会被输入到缓冲中以便稍后执行。

I变量I5必须处于正确的状态以允许本命令中指定的PLC命令执行。

∓OPEN

PLC命令自动禁止PLC程序,而且CLOSE并不重新让它使能。所以必须在一个PLC程序已经被输入或编辑之后,才能用本命令来开始执行程序的操作

范例 ENABLE PLC 1

ENA PLC 2, 7

ENABLE PLC 3, 21 ENABLE PLC 0..31

下面这个例子说明了下载一个非常简单的PLC程序并使其自动被使能的命令顺序。

OPEN PLC 7 CLEAR P1=P1+1 CLOSE ENABLE PLC 7

在线命令ENABLE PLC,OPEN PLC,<CONTROL-D>

程序命令DISABLE PLC, ENABLE PLC

ENABLE PLCC

功能: 已编译的PLC程序使能。

范围: 全局

语法: ENABLE PLCC {常量} [, {常量}]

ENA PLCC {常量} [, {常量}] ENABLE PLCC {常量}...{常量} ENA PLCC {常量}...{常量}

这里

{常量}是一个从**0**到31的整数,它代表程序的号码。

备注 本命令使PMAC(开始执行)使已编译的一个(或多个)PLC程序使能。已编译的 PLC程序使用号码指定的。它们可以在命令中以单个(不常见)、列表清单(用逗号 分隔)或连续排列的形式指定。

当本命令被发送到PMAC时,如果一个运动或PLC程序缓冲正打开,则该命令将 会被输入到缓冲中以便稍后执行。

I变量I5必须处于正确的状态以允许本命令中指定的PLC命令执行。

由于**OPEN PLC**命令自动禁止PLC程序,而且**CLOSE**并不重新让它使能。所以必须在一个PLC程序已经被输入或编辑之后,才能用本命令来开始执行程序的操作。

范例 ENABLE PLC 1

ENA PLC 2, 7

ENABLE PLC 3, 21 ENABLE PLC 0..31

在线命令DISABLE PLC,DISABLE PLC,ENABLE PLC,OPEN PLC,

<CONTROL-D>

程序命令DISABLE PLC,DISABLE PLCC,ENABLE PLC,ENABLE PLCC

ENDGATHER

功能: 停止数据采集。

范围: 全局

语法: ENDGATHER ENDG

备注: 本命令停止数据采集。数据采集可以用另一个GATHER命令再次开始(并不覆盖

旧的数据)。

本命令通常用于PMAC执行程序的数据采集和绘制功能的联合使用。

范例 GAT B1R ; 开始采集和运行程序1

ENDG : 停止采集----当关注的运动时

; 发出本命令

OPEN PROG2 CLEAR

X10

DWELL1000

CMD "GATHER" ; 在这里程序发出开始命令

X20 ; interest运动

DWELL50

CMD "ENDG" ; 在这里程序发出停止命令

CLOSE

参考 数据采集功能(特点分析)

I变量I19-I144

在线命令DEFINE GATHER, GATHER, LIST GATHER, DELETE GATHER

采集和绘制 (PMAC执行程序手册)

F

功能: 报告电机的跟随误差。

范围: 指定电机

语法: F

备注: 本命令使PMAC向主机报告当前电机的跟随误差(在统计中,精确到小数点后一位)。

跟随误差是指在某一时刻设计位置和测量到的位置之间的差值。当电机处于开环状

态(停止或允许)时,不存在跟随误差,所以PMAC将会报告一个0值。

范例 F ; 查询选址电机的跟随误差

12 ; PMAC响应

#3F ; 查询电机3的跟随误差

-6.7 ; PMAC响应

参考 跟随误差(伺服性能)

I变量lx11, lx12, lx67 在线命令**<CTRL-F>**, **P**, **V**

推荐的M变量定义Mx61,Mx62

内存映像寄存器D:\$0028, D:\$002C等; D:\$0840等

FRAX

功能: 指定坐标系的进给轴。

范围: 指定坐标系

语法: FRAX

FRAX ({轴}[, {轴}。。。])

这里

{轴}(可选)是(X, Y, Z, A, B, C, U, V, W)中的一个字母,它指定在矢量 讲给计算中使用哪一个轴。

备注: 本命令为即将进行的指定进给(**F**)的运动指定矢量进给率(速度)计算中所使用的轴。PMAC将所有进给率轴的矢量距离(轴距离的平方和的平方根)除以进给率的值作为这些运动的时间。而在同一行内得到指令的那些非进给率轴也将在相同的时间内完成运动,其速度必须保证在该时间内达到所指定的距离。

矢量进给率只有在直角坐标系中才有明显的几何意义:无论运动方向如何,它保证恒定的刀具速率。但是它也可以在一个非直角坐标系和多于三个轴的情况下指定。

如果在一个指定进给率的运动中只有一个非进给轴被命令运动,PMAC将矢量距离为0(也就是运动时间为0)进行计算,并进量在加速时间(TA或2*TS)中进行运动(这可能受电机的最大速度和/或加速参数限制)。这么做的将可能比原来要快得多。

如果本命令被发送到PMAC时一个运动程序缓冲正打开,命令将会被输入到缓冲中以便稍后执行。

例如,在一个直角坐标系XYZ中,如果你使用了FRAX(X,Y),那么所有你的指定进给率运动都将按XY平面中指定的矢量进给率进行,但是这在XYZ空间中却是不必要的。如果你使用FRAX(X,Y,Z)或FRAX,你的指定进给率运动都将按XYZ空间中指定的矢量进给率进行。坐标系的默认进给率轴是X,Y和Z。

范例 FRAX

; 使所有轴都作为进给率轴

FRAX (X,Y)

: 只使X和Y轴成为进给率轴

FRAX (X,Y,Z)

: 只把X、Y和Z轴作为进给率轴

参考 指定进给率的运动(编写一个运动程序)

程序命令F{数据},FRAX

GATHER

功能: 开始数据采集。

范围: 全局

语法: GATHER [TRIGGER]

GAT [TRIG]

备注: 本命令使数据采集按照I变量I19-I45中所定义的配置开始进行。如果命令中没有使用TRIGGER,采集将会在下一个伺服周期开始。如果使用了TRIGGER,采集将会在机器输入M12变为真后的第一个伺服周期开始。

采集将会以I19设置的频率(伺服中断周期数)进行。如果I19为0,每个GATHER 命令将只会采集一套数据。如果PMAC已经在采集数据,GATHER将会使采集周期和下一个伺服周期重新同步。

采集将会一直持续执行,直到PMAC接收到一个ENDGATHER命令,或由DEFINE GATHER命令创建的缓冲被填满。

本命令通常用于PMAC执行程序的绘制和数据采集功能和联合使用。

如果本命令被发送到PMAC时一个运动程序缓冲正打开,命令将会被输入到缓冲中以便稍后执行。

范例 GAT B1R ENDG : 开始采集和运行程序1

; 停止采集----当关注的运动结束时

;发出本命令

OPEN PROG2 CLEAR

X10

DWELL1000

CMD "GATHER" ; 在这里程序发出开始命令

X20 ; 关注的运动

DWELL50

CMD "ENDG" ; 在这里程序发出停止命令

CLOSE

参考 数据采集功能(分析功能)

|变量|19-|45

在线命令DEFINE GATHER,GATHER,LIST GATHER,DELETE GATHER

采集和绘制 (PMAC执行程序手册)

Н

功能: 执行进给保持。

范围: 指定坐标系

语法: H

备注: 本命令使当前选址坐标系立即暂停程序的执行(通过使它的时基值为零),并以I变量Ix95中定义的速率沿其路径减速。从技术角度讲,在一个H命令后程序仍然在执行,只不过其速率为0。这意味着在执行进给保持时坐标系中定义的电机不能运动。

如果要在当前选址坐标系处于进给保持模式时允许其中电机的慢进给,可以参见**/**("程序保持")命令

H命令和一个%0命令的效果是非常相似的,除了减速是由lx95而不是lx94控制和用R或S命令继续执行而不是%100命令。另外,H工作在外部时基下,而%0命令则不然。

沿路径的全速执行可以用一个R或S命令再次开始。向全速度的上升也是由lx95确定的(满时基值,内部或外部时基)。一旦达到了全速,就由lx94确定任意的时基变化。

参考 停止命令 (使你的应用安全)

控制板端口HOLD/输入(将PMAC与机器相连)

时基控制(使PMAC与外部事件同步)

I变量I51, Ix93, Ix94, Ix95

在线命令<CTRL-O>, %, %{常量}, A, K, /, Q

JPAN接口12脚

HOME

功能: 开始回零位置搜索运动。

范围: 指定电机

语法: HOME

НМ

备注: 本命令使选址电机执行一个回零位置搜索例程。回零位置搜索运动的特性由电机的

I变量lx03和lx19-lx26控制,并将编码I变量2和3用于电机位置编码。

在线HOME命令只是简单地开始回零位置搜索例程。PMAC并不保证自动指示该搜索已经完成(虽然"已到位"中断可以用于此目的)或是否该运动成功完成。通常使用查询或查询和中断的结合使用来确定运动是否成功和完成。

与此相对,如果一个回零位置搜索运动是在一个运动程序中给出的(例如 HOME1,2),那么该运动程序将自己保留完成的轨迹,并以此作为它的顺序算法规则的一部分。

如果电机的轴定义语句中有一个轴偏移量,并且/或者在电机伺服环中有后续误差,那么在回零位置搜索运动结束时报告的位置将等于轴偏移量减去跟随误差,而不是零。

范例 HOME ;选址电机开始回零位置搜索

#1HM ; 电机1开始返回回零位置

#3HM#4HM ; 电机3和4开始返回回零位置搜索

参考 控制板端口HOME/输入(将PMAC与机器相连)

返回回零位置运动(基本电机运动) I变量Ix03, Ix19-Ix26, 编码I变量2和3

在线命令HOMEZ

程序命令HOME{常量}, HOMEZ{常量}

JPAN接口11脚

HOMEZ

功能: 进行一个零移动的返回回零位置运动。

范围: 指定电机

语法: HOMEZ

HMZ

备注: 本命令使选址电机进行一个零移动的回零位置搜索运动。本命令将命令发出时电机 所在的位置(命令位置)设为回零位置,而不是进行慢进给直到发现一个预定义的

触发信号并将出发时的位置设为回零位置。

如果电机的轴定义语句中有一个轴偏移量,并且/或者在电机伺服环中有跟随误差,那么在回零位置搜索运动结束时报告的位置将等于轴偏移量减去跟随误差,而不是零。

范例: 在线命令例子

HOMEZ ;选址电机开始零移动的回零位置搜索

#1HM ; 电机1开始零移动的回零位置搜索

#3HM#4HM ; 电机3和4开始零移动的回零位置搜索

;缓冲区的运动程序例子

HOMEZ1 HOMEZ2,3

; 从PLC程序发送在线命令

IF (P1=1)

 CMD"#5HOMEZ"
 ;程序发送在线命令

 P1=0
 ;从而命令不会重复发送

ENDIF

参考 返回回零位置运动(基本电机运动)

在线命令HOME

程序命令HOME{常量}, HOMEZ{常量}

I{常量}

功能: 报告当前的I变量值。

范围: 全局

语法: 【常量】[..{常量}]

这里

{常量}是一个从0到1023的整数,它代表Ⅰ变量的号码。

第二个**{常量}是**可选的,它必须大于等于第一个**{常量}---**-它代表排列的最后一个的号码。

备注: 本命令使PMAC报告指定的I变量或I变量排列的当前值。

当I9为0或2时,只有I变量本身的值被返回(例如10000)。而当I9为1或3时,PMAC会返回整个变量赋值语句(例如*I130=10000*)。

当I9为0或1时,被选中的地址I变量的值是以十进制格式报告的。当I9为2或3时,这些变量的值是以十六进制格式报告的。

注意:如果一个运动程序缓冲区(包括一个旋转缓冲区)正打开,I(常量)将会被输入到缓冲区中以便稍后执行。它将认为是一个矢量沿X轴指向圆心的全圆周运动命令(参见"编写一个运动程序"部分的"圆周运动")。

2 ; PMAC响应

I130..135 ; 查询I130到I135的值

60000 ; PMAC响应

20000

下面的例子说明19对响应的格式的影响:

19=0 1125

49152 : 短格式, 十进制

I9=1 I125

1125=49152 ; 长格式,十进制

19=2 1125

\$C000 ; 短格式, 十六进制

I9=3 I125

1125=\$C000 ; 长格式,十六进制

参考 初始化(I)变量(计算的特点)

I变量的指定

I变量I9

在线命令I{常量}={表达式}, M{常量}, P{常量}, Q{常量}

程序命令{轴}{数据}{矢量}{数据}, I{数据}

I{常量}={表达式}

功能: 为一个I变量赋一个值。

范围: 全局

语法: | | (常量) | [..(常量)] | (表达式)

这里

{常量}是一个从0到1023的整数,它代表Ⅰ变量的号码。

第二个**{常量}是**可选的,它必须大于等于第一个**{常量}---**-它代表排列的最后一个的 号码。

{表达式}包含将被赋予指定I变量的值。

备注: 本命令将等号右侧的值赋给指定的I变量或I变量排列。

如果PMAC接收本命令时一个运动程序缓冲(包括一个旋转缓冲)正打开,命令将 会被输入到缓冲中以便稍后执行。

范例 I5=2

I130=1.25*I130 I22..44=0 I102=\$C003 I104=I103

参考 初始化(I)变量(计算的特点)

I变量的指定

在线命令I{常量}, M{常量}={表达式}, P{常量}={表达式}, Q{常量}={表达式}

I{常量}=*

功能: 为一个I变量赋以出厂时的缺省值。

范围: 全局

语法: | 【常量] [.. {常量 }]=*

这里

{常量}是一个从0到1023的整数,它代表Ⅰ变量的号码。

第二个**{常量}是**可选的,它必须大于等于第一个**{常量}---**-它代表排列的最后一个的 号码。

备注: 本命令将指定的I变量或I变量排列设置为出厂缺省值。每个I变量都有它自己的出厂缺省值,在"I变量的指定"部分有相关的说明。

范例 I13=*

I100..199=*

参考 初始化(I)变量(计算的特点)

I变量的指定----默认值

在线命令|{常量},|{常量}={表达式}

INC

功能: 指定增量运动模式。

范围: 指定坐标系

语法: INC

INC ({轴}[, {轴}...])

这里

{轴}是(X, Y, Z, A, B, C, U, V, W)中的一个字母, 代表被指定的轴;或用字母R指定半径矢量模式。

备注: 不带参数的INC命令使所有位置运动命令中的轴的全部后续位置都作按增量距离处理。一个带参数的INC语句时指定的轴变成增量模式,而所有的其它轴保持此前的状态。缺省的轴模式是绝对的。

如果 "R"被作为一个"轴"指定,那么旋转运动半径矢量性质的I、J和K部分将

被指定为增量形式(例如一个从运动起始点开始的矢量,而不是从原点开始)。一个不带任何参数的INC命令不会影响这些性质。缺省的矢量特性是增量的。

如果PMAC接收本命令时一个运动程序缓冲区(包括一个旋转缓冲区)正打开,命令将会作为一个程序语句输入到缓冲中。

范例 INC(A,B,C) ; A, B和C轴为增量模式----

; 其它轴和半径矢量不变

INC ; 所有轴都为增量模式----

; 半径矢量不变

INC(R) ; 半径矢量为增量模式

: 所有轴不变

参考 圆周运动(编写一个运动程序)

在线命令ABS 程序命令ABS,INC

J!

功能: 修正电机的命令位置为最接近的整数值。

范围: 指定电机

语法: J!

备注: 如果期望的速度为零,本命令使选址电机修正它的指定位置到最接近的整数值。当电机停止时,有可能由于指令位置为小数值产生高频振动,也有可能由于累积增量

而在指定位置前后产生振动。本命令可以用来消除这些不利的振动。

范例 OPEN PLC 7 CLEAR

IF (M50=1) ; 条件分支 **CMD"#1J/"** ; 命令电机停止

WHILE (M133=0) ; 等待期望的速度达到零

ENDWHILE

CMD"#1J!" ; 调整指令位置为整数值

M50=0;保持不被重复执行

ENDIF

参考 在线命令J/, J={常量}

J+

功能: 电机正向手动命令。

范围: 指定电机

语法: J+

备注: 本命令使选址电机在正方向上持续地进行手动运行。进给的加速度和速度由本命令

发出时lx19-lx22的值确定。

如果指定的电机所处的坐标系正在运行一个运动程序,PMAC将会拒绝本命令并报

告一个错误(如果I6为1或3,报告ERR001)。

范例 J+ ;选址电机正向手动进给

#7J+ ; 电机7正向手动进给 #2J+#3J+ ; 电机2和3正向手动进给

参考 控制板端口JOG+/输入(将PMAC与机器相连)

JPAN接口6脚

手动进给运动(电机基本运动)

I变量lx19-lx22

在线命令J-, J/, J=, J={常量}, J:{常量}, J^{常量}

J-

功能: 负向手动进给。

范围: 指定电机

语法: J-

备注: 本命令使选址电机在负方向上持续地进行手动运行。进给的加速度和速度由本命令

发出时lx19-lx22的值确定。

如果指定的电机所处的坐标系正在运行一个运动程序,PMAC将会拒绝本命令并报

告一个错误(如果I6为1或3,报告ERR001)。

范例 J- ; 选址电机负向手动进给

#5J+ ; 电机5负向手动进给 #3J-#4J- ; 电机3和4负向手动进给

参考 控制板端口JOG-/输入(将PMAC与机器相连)

JPAN接口4脚

手动进给运动(电机基本运动)

I变量lx19-lx22

在线命令J+, J/, J=, J={常量}, J:{常量}, J^{常量}

J/

功能: 手动进给停止。

范围: 指定电机

语法: J/

备注: 此命令使选址电机停止手动进给。如果电机伺服环是打开的(使能或禁止),它将

恢复位置控制,即用命令发出时的实际位置替换控制位置。手动进给负加速度由命

令发出时Ix19-Ix21的值确定。

范例 #1J+ ; 电机1正向手动进给

J/ ; 电机1停止手动进给

O5 ; 开环输出5%至电机1 O0 ; 开环输出0%

J/; 恢复闭环控制K; 停止输出J/; 恢复闭环控制

参考 控制面板端口JOG+/, JOG-/输入(将PMAC与机器连接)

JPAN管脚4,6

手动进给运动(电机基本运动)

I变量lx19-lx22

在线命令<CTRL-A>, A, J+, J-, J=, J= {常量}, J: {常量}, J^ {常量}, K, O{常量}

J:{常量}

功能: 相对指令位置手动进给。

范围: 指定电机

语法: J:{常量}

这里

{常量}为一浮点值,用数目指定手动进给距离

备注: 此命令控制电机手动进给由{常量}规定的相对当前命令位置的距离。手动进给加速度及速度由**lx19-lx21**在命令发出时的值决定。与**J^{常量}**比较,后者是相对当前实际位置的手动进给。

J:*命令可实现变加速慢进给。

如电机坐标系正在执行运动程序,则PMAC拒绝执行该命令。

范例 #1HM ; 电机1做回原点搜索

J:2000 ; 手动进给2000个计数距离(至2000) J:2000 ; 手动进给2000个计数距离(至4000)

参考 手动进给运动(电机基本运动)

I变量Ix19-Ix22

在线命令J+, J-, J/, J=, J= {常量}, J: {常量}, J^ {常量}

J:*

功能: 从当前位置手动进给至指定的可变距离。

范围: 指定电机

语法: J:*

备注: 此命令控制电机相对于当前命令位置手动进给一端由"变量手动进给位置/距离寄存器"确定的距离。手动进给加速度及速度由命令发出时**lx19-lx22**的值决定。与**J^*** 比较,后者是相对当前实际位置的手动进给。

变量手动进给位置/距离寄存器是单位计数的浮点寄存器。推荐存储浮点M变量。寄存器PMAC地址: 电机1 L: \$082B, 电机2 L: \$08EB。常规方法是将M变量赋予目标地址值后写入该寄存器,然后执行J:*命令。

如电机坐标系正在执行运动程序,则PMAC拒绝执行该命令。(如果**I6**为1或**3**,则显示错误代码**ERR001**)。

范例 M172->L:\$082B ; 定义1号变量"位置/距离寄存器"

#1HMZ ; 当前位置清零 M172=3000 ; 寄存器赋距离值

#1J:* ; 电机手动进给此段距离,命令结束位置值3000 #1J:* ; 电机手动进给此段距离,命令结束位置值6000

M172=P1*SIN (P2) ; 赋予寄存器新值 #1J:* ; 电机1手动进给此段距离 #1J= : 退回附近前目标位置

参考 手动进给运动(电机基本运动)

I变量lx19-lx22

内存地址寄存器 **L:\$082B**, **L:\$08EB** 等。 推荐的M变量定义**M172**, **M272**, 等等 在线命令**J**=, **J**= **{常量**}, **J**=*, **J**^*

J=

功能: 手动进给至上一次位置

范围: 指定电机

语法: J=

备注: 此命令使电机手动进给至前次预进给和手轮运动位置(最近一次的编程指定位置)。 手动进给加速度及速度由命令发出时**lx19-lx22**的值决定。

存储此位置信息的寄存器被称为目标位置寄存器(电机1D:\$080B,电机2D:\$08CB)。定义M163,M263等等M变量可在编程时读取这些寄存器。

如用/或\停止命令暂停程序执行,一个或多个电机偏离了停止时的位置,在程序继续执行前,必须J=命令使电机恢复停止时位置。J=命令也可用于电机运行中途程序退出的情况,因它可使电机退回至上次编程指定终点位置(设l13=0使PMAC不处于分段方式),程序因而可从此点正确恢复执行。

如电机坐标系正在执行运动程序,则PMAC拒绝执行该命令。(显示错误代码 **ERR001**,如果**I6**为1或**3**)。

范例 &1Q ; 动作结束时停止运动程序

#1J+ ; 电机1从该位置正向手动进给

J/ ; 停止手动进给

J= ; 退回程序退出时的位置

R ;继续运行程序

&1A ; 运动中途暂停程序执行

#1J=#2J=#3J : 将所有电机退回到初始终点位置

R ;继续运行程序

参考 控制面板端口 PREJ/输入 (将PMAC与机器连接)

JPAN管脚7

手动进给运动(电机基本运动)

|变量|x19-lx22

在线命令 J+, J-, J/, J= {常量}, J: {常量}, J^ {常量}

J={常量}

功能: 手动进给至指定位置

范围: 指定电机

语法: J={常量}

这里

{常量}为一浮点数值,用编码计数定义手动进给目标位置。

备注: 此命令使电机运行至指定{常量}位置,手动进给加速度及速度由命令发出时**lx19- lx22**的值决定。

J=* 命令适用于手动进给至可变目标地址。

如电机坐标系正在执行运动程序,则PMAC拒绝执行该命令。(显示错误代码

ERR001,如果I6为1或3)。

范例 J=0 ; 使电机手动进给至位置0

#4J=5000 ; 电机4手动进给5000个计数单位 #8J=-32000 ; 电机8手动进给-32000个计数单位

参考 手动进给运动(电机基本运动)

I变量lx19-lx22

在线命令 J+, J-, J/, J=, J: {常量}, J^ {常量}, J=*, J:*, J^*

J=*

功能: 手动进给至可变指定位置。

范围: 指定电机

语法: J=*

备注: 此命令使电机手动进给至"可变手动进给位置/距离"寄存器设定位置,手动进给加

速度及速度由命令发出时lx19-lx22的值决定。

"可变手动进给位置/距离"寄存器为单位计数的浮点寄存器。推荐由浮点M变量读取。寄存器位于PMAC地址 电机1 L:\$082B 电机2 L:\$08EB。通常赋予M变量值,设定寄存器目标位置,而后执行J=*命令。事实上可写入电机"目标位置"寄存器,J=命令获取相同结果。但调用J*命令允许你在其后返回实际目标位置,而不用重新写入目标位置寄存器。J=*命令使用的寄存器按计数单位整数比例计数,而非分数。

如电机坐标系正在执行运动程序,则PMAC拒绝执行该命令。(显示错误代码 ERR001,如果I6为1或3)。

范例 M172->I:\$082b ; 定义#1可变手动进给位置/距离寄存器。

M172=3000 ; 赋予寄存器位置值 #1J=* ; 电机1手动进给至此位置 M172=P1*SIN (P2) ; 赋予寄存器新位置值 #1J=* ; 电机1手动进给至此位置 #1J= ; 退至手动进给前目标位置

参考 手动讲给运动(电机基本运动)

I变量lx19-lx22 内存映像寄存器L:\$082B, L:\$08EB, M变量定义 M172, M272, 等等

在线命令 J=, J= {常量}, J:*, J^*

J=={常量}

功能: 手动进给至指定电机位置,并设定此位置为上一次进给位置。

范围: 指定电机

语法: J=={常量}

{常量}为一浮点数,设定手动进给位置,编码计数。

备注:

此命令用于使电机手动进给至{常量}设定的位置,并将此位置设为初始位置,即**J**=命令的目标位置。手动进给加速度及速度由命令发出时**Ix19-Ix22**的值决定。

如电机坐标系正在执行运动程序,则PMAC拒绝执行该命令。(显示错误 代码**ERR001**,如果**I6**为1或**3**)。

范例 #1J==10000 ; 电机1手动进给10000计数单位,并将此位置设为预进给位

置J+; 向正方向随机手动进给

J= ; 返回10000位置

参考 手动进给运动(电机基本运动)

I变量Ix19-Ix22

在线命令 J= , J= {常量}, J=*, J^*

J^{常量}

功能: 相对实际位置手动进给。

范围: 指定电机

语法: J^{常量}

这里

{常数}为一浮点值,以计数方式设定手动进给距离

备注: 此命令使电机手动进给{常量}设定的相对当时实际位置的一段距离。手动进给加度及速度由命令发出时**lx19-lx22**的值决定。与**J:{常量}**比较,后者为相对于当

命令位置的手动进给。

通常**J:{常量}**命令更有用,因其目标位置并不受命令发出,执行这段时间内产生误的影响,**J^0**命令用以消除存在的后续错误。**J^***命令可用于可变的手动进给

加。

如电机坐标系正在执行运动程序,则PMAC拒绝执行该命令。(显示错误

代码ERR001,如果I6为1或3)。

范例 #1HM ; 电机1归位

J^2000 ; 从实际位置手动进给2000计数单位距离

如当前为-5,新命令位置为1995

J^2000 ; 从实际位置手动进给2000计数单位距离

如当前为1992,新命令位置为3992

参考 手动进给运动(电机基本运动)

I变量lx19-lx22

在线命令 J+, J-, J/, J=, J= {常量}, J:{常量}, J=*, J:*, J^*

J^*

功能: 从当前实际位置手动进给至变量指定的距离。

范围: 指定电机

语法: J^*

备注:

此命令使电机相对于当前实际位置手动进给"变量手动进给位置/距离"寄存器设定

的一段距离。手动进给加速度及速度由lx19-

Ix22在命令发出时的值决定。和命令**J**:*比较,后者是相对于当前命令位置的手动进

给。

"可变手动进给位置/距离"寄存器是单位计数的浮点寄存器。由浮点变量M变量读取。该寄存器的PMAC地址: 电机1--L: \$082b, 电机2--L: \$08EB。

通常做法是赋予M变量目标位置值,写入该寄存器。调用J^*命令。 如电机坐标系正在执行运动程序,则PMAC拒绝执行该命令。(显示错误 代码ERR001,如果I6为1或3)。

范例 M172-^L:\$082B ; 定义#1变量读取手动进给位置距离寄存器

#1HMZ ; 当前位置定义为0 M172=3000 ; 赋予寄存器距离值

#1J^* ; 电机1手动进给此段距离; 若命令发出后

;的错误代码为3,则结束命令位置为2997

; 电机1手动进给此段距离; 若命令发出后的错

; 误代码为2,则结束命令位置为5995

M172=P1*SIN(P2) ; 赋予寄存器新距离值 #1J^* ; 电机1手动进给此段距离 #1J= ; 返回手动进给前目标位置

参考 手动讲给运动(电机基本运动)

I变量Ix19-Ix22

#1J^*

内存映像寄存器L:\$082B, L:\$08EB,

M变量定义 M172, M272等

在线命令 J= , J= {常量}, J=*, J^*

{手动进给命令}^{常量}

功能: 持续手动进给, 直到触发干预。

范围: 指定电机

语法: J=\{常量}

J={常量}^{常量} J:{常量}^{常量} J^(常量}^(常量} J=*^{常量} J: *^{常量} J^*^{常量}

这里

^后的{常量}为一浮点值,发现设定触发后,手动进给到用编码计数的该点的距离。

备注

此命令方式提供"手动进给直到触发"功能。当**^{常量}**结构加入任何确定的缓进命令,手动进给运动会被预先设定的触发条件中断,电机会停止在相对于最终命令设定的触发位置。不确定的手动进给命令**J+,J-**

不可变为手动进给直至触发方式。手动进给直至触发运动类似于归位运动,后者不需触发点就有一确定终止位置,且它不改变电机零位置。

手动进给直至触发功能可用于任何手动进给命令,无论基本的手动进给命令是确定的还是不确定的。如基本手动进给命令是确定的,(如: **J=10000**),没有触发点时,电机停止在予设位置。如基本手动进给命令是不确定的,(如: **J+**), 无触发点时,电机将持续运动直到被另一命令或出错信息终止。

手动进给直至触发运动的触发条件,可为一输入标志位或出错后的警告。若**lx03**的**17**位是**0**(默认值),触发即为一次输入标志与/或编码索引通道从**lx25**预定义的转换。编码/标志变量**2**和**3**(**l912**和**l913**)定义哪个输入信号的哪个沿产生触发。若**lx 03**的**17**位是**1**,触发源是电机出错位变真后的警告。**lx12**设定电机错误位。

触发位置可为硬件设置也可软件读取。若**Ix03的16**位为0(默认值),被PMAC的**D SPGATE**硬件触发而锁住的编码位置将作为触发位置使用。这是最准确的方式,它使用触发瞬间的位置值,但只能用在与触发标志位设定的通道号相同的递增编码反馈。此方式不可用于其他方式的反馈及后续错误触发。

如**lx03**的第**16**位是**1**,PMAC读取当前传感器测得触发源的位置。此方式可用于任意形式反馈及触发条件。但由于软件延迟,准确性相对于硬件低。慢进给加速度及速度由**lx19-lx22**在命令发出时的值决定。

如电机坐标系正在执行运动程序,则PMAC拒绝执行该命令。(显示错误 代码**ERR001**,如果**I6**为1或3)。

范例 #1J=^1000 ; 无触发源时手动进给到原始位置发现触发源

;后,手动进给距触发源+1000个计数距离

#2J:5000~100 ; 无触发源时,向正方向手动进给5000个计数距

; 离发现触发源后, 手动进给距触发源-100个计

; 数距离

#3J=20000⁰ ; 无触发源时手动进给20000, 发现触发源后回

; 复到触发位置。

参考 手动进给运动 (电机基本运动)

I 变量Ix19-Ix22, Ix03, Ix25, 编码/标志位 I变量2及3 在线命令J=, J= {常量}, J: {常量}, J^ {常量}, J=*, J:*, j^* 编程命令 {轴} {数据} ^{数据}

K

功能: 终止电机输出。

范围: 指定电机

语法: K

备注: 此命令使PMAC终止对电机的输出。司服环失效,**DAC**输出为0(**Ix29** 与 **/或Ix79** 设定有效),且**AENA**对电机输出为失效状态(**E17**设定极性)。

J/命令可恢复对电机的闭环控制。A命令可重建坐标系内所有电机的闭环控制, 〈CTRL-A〉命令可恢复所有PMAC电机的控制。

K命令的效果与PMAC在出现放大器错误或后续错误后自动采取的措施一致。

如电机坐标系正在执行运动程序,则PMAC拒绝执行该命令。(显示错误代码ERR 001,如果I6为1或3)。通常先用A命令终止程序。 全局<CTRL-K>命令在任意条件下立即终止所有电机运动。

范例 K ; 终止访问的电机 #1K : 终止电机1

J/ ; 重建对电机1的闭环控制

参考 放大器错误,跟随误差限制,停止命令

I- 变量lx29,lx79,

在线命令<CTRL-A>,<CTRL-K>,A,Q,H,J/ 跳线 E17, E17A-E17H

LEARN

功能: 获取当前命令位置。

范围: 指定坐标系

语法: LEARN[({轴}[,{轴}...]]

LRN[({轴}[,{轴}...])

备注: 此命令使PMAC开放程序缓存末尾加入一行,使一些/甚至全部坐标系中电机轴

置命令等于当前命令位置。用此种方式PMAC能获取一系列后续运动程序执行时

重复的点。

PMAC可有效的PMATCH功能,读取电机命令位置,将轴定义方程转换为计算位置。如果轴名称在LEARN命令中指定,该轴的命令位置被用于加入运动程序行中。如轴名称未在LEARN命令中指定,九个可能的轴名称的位置命令被用以加入程序的行中。对无电机名附及轴的位置命令将被置零。

注意:若电机做闭环运动,获取的位置会与实际的不同,因命令位置被用及,相差数量等同于后续位置错误数量。若电机为开环或死机,PMAC自动将电机命令位置设为电机实际位置,从而使LEARN功能在电机处于任意状态下有效。

范例 &1 ; 指定坐标系1

#1->1000X ; 在坐标系1中定义电机1 #2->1000Y ; 在坐标系1中定义电机2

OPEN PROG 1 CLEAR ; 清程序缓存

F10 TA200 TS50 ; 输入所需非运动命令 {将电机移至某一位置,如#1到13450, #2到29317} LEARN(X,Y) : PMAC获取这些位置点

X1.345 Y2.9317 ; 这是PMAC 加至程序1的序列

{将电机移动至新位置,如#1到16752,#2到34726 LEARN ; PMAC获取这些位置点

A0 B0 C0 U0 W0 X1.6752 Y3.4726 Z0

; PMAC 将所有轴位置加入程序1

参考 获取一个运动程序(编写一个运动程序)

在线命令PMATCH

LIST

功能: 列出当前打开的缓存的内容。

范围: 全局

语法: LIST

备注: 此命令使PMAC向主机报告当前打开缓存的内容(PLC, PROG, ROT)。若无

存打开,PMAC将报告出错(ERR003 如I6=1或3)。注意此命令不包括OPEN

CLEAR,CLOSE(因为它们不是程序命令)。

可在list命令中指定缓存名称,列出未打开的缓存。(LIST PROG 1)。见如下LIST

命令说明。

范例 OPEN PROG 1 : 打开缓存等待输入

LIST ; 列出打开缓存内容

LINEAR : PMAC报告打开的缓存内容

F10 X20 Y20 X0 Y0 RETURN

CLOSE ; 关闭缓存

LIST ; 列出打开缓存内容 <BELL>ERR003 ; PMAC报告出错

参考 在线命令OPEN,CLOSE,LIST PLC,LIST PROGRAM

LIST COMP

功能: 列出电机补偿表内容。

范围: 指定电机

语法: LIST COMP

备注: 此命令用于PMAC报告主机电机补偿列表内容。以十进制ASCII码形式成行显

空格分隔。

LIST COMP DEF命令用于显示表头内容

如不存在电机列表,PMAC拒绝执行此命令(显示ERR003如I6=1或3)。

此补偿列表不会影响电机位置或被其影响。

范例 LIST COMP ; 列出电机补偿列表内容

917 -3 6 35 87 65 24 18 -9 -16 -34

7 12 -3 -8 32 44 16 0 -20 -5 0PMAC连续响应

参考 丝杠补偿表

在线命令DEFINE COMP, DELETE COMP, LIST COP DEF

LIST COMP DEF

功能: 列出电机补偿列表定义。

范围: 指定电机

语法: LIST COMP DEF

备注: 此命令用于PMAC向主机报告电机补偿列表定义。定义包括DEFINE COMP命

建立的四项内容。(尽管有些项并未明确指定):

1.表中输入项数(二维列表为行数和列数)。

- 2.提供位置数据源的电机号。
- 3.被此列表改动位置的电机号。
- 4. 表中数据源电机跨距。

如电机无此表, PMAC拒绝此命令(显示ERR003 如 I6=1或3)

此补偿列表不会影响电机位置或被其影响。

范例 LIST COMP DEF ; 列出补偿表定义

100, #2, #2, 100000; PMAC 反应; 表中有100个输入项,

2号电机是目标及源,跨距100,000计

数单位。

#3 LIST COMP DEF ; 列出3号电机补偿表定义

10, 20, #4, #5, #6, 50000, 100000

; PMAC 反应A 2D 10x20列表, 跨距50Kx100K,

#4是源, #6是目标。

参考 丝杠补偿表

在线命令 DEFINE COMP, DELETE COMP, LIST COMP

LIST GATHER

功能: 报告数据采集缓存的内容。

范围: 全局

语法: LIST GATHER [{起点}] [,{长度}]

LIS GAT [{起点}] [,{长度}]

这里

可选项{起点}参数为一整数常量,设定列表开始的缓存地址(默认值为0)。 可选项{长度}参数为一整数常量,设定缓存送入主机的字节数(默认值至缓存尾地址)。

备注: 命令PMAC向主机报告数据采集缓存内容。数据形式为48位长字型,16进制,

每字12位,空格分隔。每行16个长字型

若**{起点},{长度}**均不设定,缓存内全部内容将被列出。设定**{起点}**,数据报告从**{起点}**个字数处开始。设定**{长度}**,报告从缓存开始共**{长度**}个字。

范例 LIST GATHER ; 列出整个缓存内容

LIST GATHER 256 ; 跳过前256个字 LIST GATHER 0, 32 ; 列出前32个字 LIST GATHER , 32 ; 与上效果相同

LIST GATHER 64, 128 ; 跳过前64字, 列出后续的128个字

参考 数据采集功能(分析性能)

I-变量 I19, I20, I21-44

在线命令GATHER, ENDGATHER, DEFINE GATHER

采集与绘图(PMAC 执行程序手册)

LIST LDS

功能: 列出梯形图功能的连接地址。

范围: 全局

语法: LIST LDS

备注: 此命令使PMAC列出PLC 交叉编辑器正确编辑连接程序所需的梯形逻辑内部例程

地址。此命令由交叉编辑器自动完成,用户只会在特殊调试时直接使用。

参考 在线命令 LIST LINK

LIST LINK

功能: 列出内部PMAC程序的连接地址。

范围: 全局

语法: LIST LINK

备注: 此命令使PMAC列出PLC交叉编辑器所需正确编辑连接程序的内部程序地址。

此令由PLC交叉编辑器在执行程序时自动完成。

对于独立的DOS交叉编辑器,PMAC反馈此命令的ASCII码必须包括与交叉编辑器在同一目录且同一子目录名为LISTLINK.TXT的文件。每个独立版本的固件分配不同的内部程序地址,因此即使是一次小的升级,如从V1.15A 到V1.15B,也必须创建一个新的LISTLINK.TXT。

范例 LIST LINK : 列出地址

004532 004A97 005619 005F21 0062FE 0063A4

; PMAC响应

参考 编译PLCs (编写一个PLC程序)

LIST PC

功能: 在程序计数器处列出程序内容。

范围: 指定坐标系

语法: LIST PC [, [{常量}]]

这里

{常量}代表要列出程序字数的正整数

备注: 此命令使PMAC列出将在坐标系计算的程序行数,第一行前标有程序号,后续行前标有初始地址。LIST PC列出下一个要计算的行。LIST PC,列出从下一行到程序末尾将被计算的行。LIST PC,{常量}列出从下一行开始的指定地址大小范围。用LIST PE 命令查询当前执行行。

因PMAC 在一序列动作之前提前计算, **LIST PC**(Program Calculation 程序计算)命令一般来讲返回的程序行数要比**LIST** 命令多。如坐标系未指向任何运动程序,PMAC会返回出错信息(**ERR003** 如**I6=1**或3)。必须用**B{常量**}命令初始指定。

PΕ

范例 LIST PC ; 列出要计算的下一行

P1:22:X10Y20 ; PMAC响应

LIST PC, 4; 列出要计算程序的下四个字

P1:22:X10Y20 ; PMAC响应

24:X15Y30

LIST PC, ; 列出程序其余部分 P1:22:X10Y20 ; PMAC响应

> 24:X15Y30 26:M1=0 28:RETURN

参考 在线命令B{常量}, LIST, PC, LIST PE, PE

LIST PE

功能: 列出程序处的程序内容执行。

范围: 指定坐标系

语法: LIST PE , [{常量}]]

这里

{常量}是代表列出程序字数的正整数

备注: 此命令使PMAC列出以正在坐标系内执行的运动为起始行的程序段,第一行前标有程序号,后续行前标有初始地址。

: PMAC响应

因PMAC 在一序列动作之前提前计算, LIST PC (Program Calculation 程序算) 命令一般来讲返回的程序行数要比LIST PE 命令多。

LIST PE 只返回当前执行行。LIST PE返回从当前行到结束行的情况。LIST PE, {量}返回从当前执行行开始到结束的字数。如坐标系未指向任何运动程序,PMAC返回出错信息(ERR003 如l6=1或3)。必须用B{常量}命令初始指定。

范例 LIST PE ; 列出当前执行行

P5:35:X5Y30 ; PMAC响应

LIST PE, 4; 从执行行开始,列出4行

P5:35:X5Y30

37:X12Y32 39:X0 Y10 41:RETURN

参考 在线命令B{常量}, LIST, LIST PC, PC, PE

LIST PLC

功能: 列出指定PLC程序内容。

范围: 全局

语法: LIST PLC {常量}

这里

{常量}为代表PLC程序数,从0到31的整数

备注: 此命令用于PMAC向主机报告指定PLC程序缓存内容。以ASCII文本显示。 如I 是0或2,内容以短格式列出(例如ENDW)。 如I9是1或3,内容以长格式列出。(例如ENDWHILE)。

PLCs0-15可加密保护。若PLCs加密,而密码未被正确给出,PMAC将拒绝执行此命令,(显示出错信息ERR002如I6=1或3)。

范例 LIST PLC 0 LIST PLC 5

参考 PLC程序特征

I-变量I3,I4,I9

在线命令LIST , LIST PROG , PASSWORD={字符串}

程序命令说明

LIST PROGRAM

功能: 列出指定运动程序内容。

范围: 全局

语法: LIST PROGRAM {常量} [{开始}] [, {长度}]

LIST PROG {常量} [{开始}] [, {长度}]

这里

{常量}为指定运动程序号从1到32767的整数。

可选参数**{起始点}**是一整数常量,指定列表开始位置距缓存起始处的距离默认值**0**)。

可选参数**{长度}**是一整数常量,指定送至主机的缓存字节数(默认值为 至缓存末尾)。

备注 此命令用于PMAC以**ASCII**文本格式向主机报告固定运动缓存内容。如 **I9**是0或2,内容以短格式列出(例如**LIN**)。如**I9**是1或3,内容以长格 式列出(例如**LINEAR**)。

若**{起始点}{长度}**均未指定,缓存全部内容将被报告。若**{起始点}**指定,报告开始于距缓存**{起始点}**个字节处。若**{长度}**指定,报告从起始点开始**{长度**}个字节。

若**{起始点}{长度}**任意或全部定义,列出的程序将包含每行的缓存地址偏移量。此偏移量将在后续命令**PC,LIST PC** 命令中用到。

若PMAC不存在由此命令调用的运动程序,则PMAC 拒绝执行此命令且显示错误信息(ERR002 若**I6**=1或3)。

PROGs1000-32767可加密。若PROG加密,而密码未被正确给出,PMAC将拒绝执行此命令,(显示出错信息ERR002如**I**6=1或3)。

范例 LIST PROG 9 : 列出运动程序9的全部内容

LINEAR ; PMAC响应

F10

X10Y10 X0Y0

RETURN

LIST PROG 9, ,列出附加地址偏移量的程序

0:LINEAR 1:F10

2:X10Y10 ; 注意两轴命令需两个地址

4:X0Y0

6:RETURN

LIST PROG 9, 4 ; 列表从地址4 起始

4:X0Y0

6:RETURN

LIST PROG 9, 2, 4; 列表从地址2起始,长度2个字节

2:X10Y10

4:X0Y0

LIST PROG 9, , 2 ; 列表从头开始, 长度**2**字节

0:LINEAR

1:F10

参考 编写一个运动程序

I-变量 I3, I4, I9

在线命令 LIST, PC, LIST PC., PASWORD={字符串}

程序命令说明

M{常量}

功能: 报告M变量值。

范围: 全局

语法: M{常量} [..{常量}]

这里

{常量}为一代表M变量号,从0到1023的整数。

第二个选项{常量}至少要与第一{常量}一样大,它代表最末一变量号。

备注: 此命令使PMAC报告当前指定或一段M变量值。PMAC 并不报告**M-**变量定义地址: 这是**M{常量}->**命令完成的。

注意:此命令送至PMAC时,若一运动程序缓存正打开(包括旋转缓存), 它将被送至缓存,等待执行,成为一M-代码子程序调用。

范例 MO ; 主机查询值

3548976 ; PMAC响应

M165 5. 75 M1..3

0 1

参考 M变量(计算的性能)

在线命令**M{常量}={表达式}**, **M{常量}->** 程序命令 **M{常量}**, **M{常量}={表达式**}

M{常量}={表达式}

功能: 向M变量赋值。

范围: 全局

语法: M{常量} [..{常量}]={表达式}

这里

{常量}为0到1023表示M变量号的整数。

第二个选项{常量}至少要与第一{常量}一样大,它代表最末一变量号。

{表达式}为即将赋予的指定M变量的值。

备注: 此命令将等号右边值赋予M变量。它并不向变量赋予定义(地址),这是由**M{常量}->{表达式**}命令完成。

此命令送至PMAC时,若一运动或PLC程序缓存正打开(包括旋转缓存),它将被送至缓存,等待依次执行。

范例 M1=1

M9=M9 & \$20 M102=-16384

M1..8=0

参考 M变量(计算的性能)

在线命令M**{常量}** , M**{常量}->{定义}** 程序命令 M**{常量**} , M**{常量}={表达式**}

M{常量}->

功能: 报告当前M变量定义。

范围: 全局

语法: M{常量} [..{常量}]->

这里

{常量}为0到1023表示M变量号的整数;

第二个选项{常量}至少要与第一{常量}一样大,它代表序列的最末一变量号。

备注: 此命令将等号右边值赋予M变量。它并不向变量赋予定义(地址),这是由**M{常量}->{表达式**}命令完成。

当I9为0或2, 只返回它自身的定义(Y:\$FFC2,0).当I9为1或3,返回全部定义句.(M11->Y:\$FFC2,0)

范例 M1-> ; 主机查询定义

Y:\$FFC2,8 ; PMAC响应

M101..103-> X: \$C001,24,S Y: \$C003,8,16,S X:\$C003,24,S

参考 M变量(计算的性能)

在线命令M{常量} ,M{常量}->{定义},M{常量}={表达式} 程序命令M{常量}={表达式}

M{常量}->*

功能: 自参考 M变量定义。

范围: 全局

语法: M{常量} [..{常量}]->*

这里

{常量}为0到1023表示M变量号的整数。

第二个选项{常量}至少要与第一{常量}一样大,它代表最末一变量号。

备注: 此命令使PMAC将指定M变量与自身定义字对照.若你只想把M变量用做标志位,状态位,计数器或其他简单变量,便无必要找一开放存储器,因为可以用定义空间来存储值.只须简单定义M变量类型,便可象P变量一样使用. 注意其范围为-1,048,576到+1,048,575.

当给出定义后,值自动置0。

此命令也可用于清除现存M变量定义。

范例 M100->*

M20..39->*

M0..1023->*;清除所有当前M变量定义

; 在调入新值之前用此命令是很适宜的

参考 M变量(计算的性能)

在线命令 M{常量}, M{常量}->, M{常量}->, M{常量}->{定义}, M{常量}={表达式} 程序命令 M{常量}={表达式}

M{常量}->D:{地址}

功能: 长定点M变量定义。

范围: 全局

语法: M{常量} [..{常量}]->D[:] {地址}

这里

{常量}为0到1023表示M变量号的整数。

第二个选项{常量}至少要与第一{常量}一样大,它代表最末一变量号。

{地址}为一从0到65,535的常整数(十六进制为\$FFFF)

备注: 此命令使PMAC在其地址空间的指定位置定义指定的**M**变量为一48位双字(包括X及Y存储器,X更重要)。该数据被称为定点整数.

定义包括字母D,可选项冒号(:), 及字地址.

在存储器分布图中,用此种方式的存储位置用**D:**标志

范例 M161->D:\$0028 ; 用十六进制指定电机1所需的位置寄存器

M161->D40 ; 用十进制指定电机1所需的位置寄存器 M162->D\$2C ; 用十六进制指定电机1的实际位置寄存器

参考 M变量(计算的特点)

在线命令 M{常量} , M{常量}->, M{常量}={表达式} 程序命令 M{常量}={表达式}

M{常量}->DP:{地址}

功能: 双端口RAM定点M变量定义。

范围: 全局

语法: M{常量} [..{常量}]->DP[:]{地址}

这里

{常量}为0到1023代表M变量号的整数。

第二个选项{常量}至少要与第一{常量}一样大,它代表最末一个变量号。

{地址}是一个从0到65,535的常整数(十六进制为\$FFFF)

备注: 此命令使PMAC定义一给定的**M**变量或者**M**变量的范围指向在PMAC的寻址空 指定位置的**X**和**Y**寄存器的低16位的32位数据位.

定义包括字母DP,可选项(:),及字地址。

此格式只适用于位于\$0到\$DFFF的双端RAM(须选项二)。从而主机可以此种格式的标准32位整数读取相应地址。位于X字节的数据更重要,

这意味着主机两个连续地址中的高位更重要。

范例 M 150—>DP:\$D200 M 250—>DP\$D201

参考 M变量(计算的特点)

双端口RAM(编写主机通信程序) 在线命令 M{常量}, M{常量}->, M{常量}->F: {地址}

M{常量}={表达式} 程序命令 M{常量}={表达式}

M {常量} ->F: {地址}

功能: 双端口RAM浮点M变量定义。

范围: 全局

语法: M{常量} [..{常量}]-> F[:]{地址}

这里

{常量}为0到1023代M变量号的整数;

第二个选项{常量}至少要与第一{常量}一样大,它代表最末一变量号。

{地址}为一从0到65,535的常整数(十六进制为\$FFFF)

备注: 此命令使PMAC定义指定M变量或M变量范围指向32位数据,

用PMAC地址空间指定位置的**X**和**Y**存储器的低16位表示。此数据被称为单精度**32**位格式浮点整数。

定义包括字母DP,可选项(:),及字地址。

此格式只适用于位于 \$ D000到 \$ DFFF的双端RAM(须有附件二)。从而主机可以标准IEEE 3 2位浮点数据格式读取相应地址。

IEEE32为浮点格式在位31有标志位(MSB);偏移指数在位30到位23(指数为该值减127);位22到0的小数(这里要向小数的尾数加1)。该字是以标准的Intel格式安排的。

范例 M155->F:\$D401 M255->F\$D402

参考 M变量(计算的特点)

双端 RAM(编写一个主机通信程序) 在线命令 M(常量}, M(常量}->,

M{常量}->DP: {地址}

M{常量}={表达式}

程序命令 M{常量}={表达式}

M {常量} ->L: {地址}

功能: 长字节浮点M变量定义。

范围: 全局

语法: M{常量}[..{常量}]->L[:] {地址}

这里

{常量}为一从0至1023的整数,代表M变量的个数。

第二选项{常量}必须至少与第一个{常量}相等,它代表系列末端变量号:

{地址}为一从0至65,535的整数常量(如定义为十六进制,范围从\$0至\$FFFF)

备注: 此命令使PMAC定义M变量或一系列M变量指向一位于PMAC地址空间指定位置的长字节(48位)数,包括X存储器及Y存储器。此数据为具有PMAC的48位浮点格式的浮点值。

此定义包括字母L,可选冒号(:),以及字节地址。

在存储器地址列表中用此种格式存储的用'L:'标志。

范例 M165->L:\$081F M265->L\$0820 M265->L2080

参考 M变量(计算的特点)

在线命令 M{常量}, M{常量}->, M{常量}->D: {地址} M{常量}={表达式} 程序命令 M{常量}={表达式}

M {常量} ->TWB: {地址}

功能: 二进制多路拨码开关定义。

范围: 全局

语法: M {常量} [..{常量}] ->TWB[:] {多路地址}, {偏移量}, {大小}

这里

{常量}为一从0至1023的整数,代表M变量的个数;

第二选项{常量}必须至少与第一个{常量}相等,它代表系列末端变量号:

{多路地址} 为一从 $0 \le 2 \le 5$ 的整型常量,代表**M**变量中用到的最低位的位地址:

偏移量} 为-0至7的整型常量,代表字节中哪一位是被**M**变量用到的最低位:

 $\{ 大小 \}$ 为一 1 到 3 2 的整型常量,代表M变量将用到的连续位个数。

{格式} (可选) U为无符号数, S代表符号数。如未指定, 默认值为 U

备注: 本命令使PMAC为一系列的连续输入位定义指定的M变量或M变量排列。这些输入在附件18或兼容的硬件的指轮端口上的进行选择。

范例 M0->TWB:0,0,1 M1-> TWB: 0, 1, 1 M10-> TWB: 3,4,4,U

M745->TWB: 4,0,16,S M872 ->TWB: 0,4,1

参考 M变量(计算的特点)

在线命令M{常量}, M{常量}->, M{常量}->, M{常量}->TWD: {地址}

指轮多路选择板(ACC-18)手册

M {常量} ->TWD: {地址}

功能: BCD多路拨码开关M变量定义。

范围: 全局

语法: M {常量} [..{常量}] ->TWD [:]

{多路地址}, {偏移量}, {大小}, [.{dp}],{格式}

这里

{常量}为一从0至1023的整数,代表**M**变量的个数;

第二选项**{常量}**必须至少与第一个**{常量}**相等,它代表系列末端变量号;

{多路地址} 为一从0至255的整型常量,代表M变量中用到的最低位的位地址;

{偏移量}为 0 到 4 ,表示最高位是低**4**位或高**4**位 **{大小}** 为一 1 到 1 2 的整型常量,代表**M**变量将用到的数字个数。

{dp} (可选) 为一从0到8的整型常量,代表小数点右侧的位数。

{格式} (可选) U为无符号数, S代表符号数。如为符号数,最重要数的最不重要位为符号位(其余位被忽略)。如未指定,默认值为U

备注: 本命令使PMAC定义指定的**M**变量或一定范围的**M**变量。这些变量指向一系列 二进制表示的十进制数,该数在附件**18**或兼容硬件的端口上的进行选择。

该M变量是只读的浮点变量。这些变量一旦被定义,就可以被查询和在表达式中使用。而每次这些变量在表达式中使用,多路选择板上相应的地址就被读取。

范例 M100->TWD:4,0,8.3,U意为多路地址4,低4位(左侧数);有8个数,其中3个小数位;永远为正。相对应于低地址指轮板的最下排的8位指轮数。小数点后有3位。

M99->TWD:0,0,1,U 意为只用了一个数,位于多重地址 0 ,低 4 位(左侧)。这对应于最低地址的指轮板的左侧上方的指轮。

参考 M变量(计算的特点)

在线命令**M{常量}**, **M{常量}->**, **M{常量}->**, **M**{常量}-> T W B: {地址}

多路选择板(ACC-18)手册

M {常量} ->TWR: {地址},{偏移值}

功能: 旋转变压器多路选择M变量定义。

范围: 全局

语法: M{常量} [..{常量}]->TWR[:] {多重地址}, {偏移量}

这里

{常量}是一个从0到1023的整数,它代表M变量的号码。

第二个{常量}时可选的,它不能比第一个{常量}小,代表排列的最后一个号码。

{多路地址}是一个常整数,被2除后应该在0到254之间。它代表多路选择(指轮 多路选择端口上的ACC-8D可选件7旋转变压器-数字转换器板中)中的地址,由该 板上的DIP开关设置确定。

备注: 本命令使PMAC定义指定的**M**变量或**M**变量的排列。这些变量指向一个来自旋转变压器-数字(**R/D**)转换器或相似设备(该设备有一个**ACC-8D可选件7**或兼容板的指轮端口选择)的**12**位字。

在这里指定的多路选择端口上的地址必须和**ACC-8D可选件7**的板上的DIP开关设置的地址相匹配。**ACC-8D可选件7**的手册中有一个所有可能情况的列表。

ACC-8D可选件7的一个DIP开关确定是否板上的R/D转换器带有0到3或4到7的偏移量。指定的{偏移量}必须和该开关的设置及板上R/D设备的号码相匹配。这是一个只读的M变量格式。如果在一个在线的查询命令或程序语句中应用该变量,将会使PMAC通过多路选择端口以12位的无符号数(从0到4095)对指定设备进行计数。

注意:对于访问一个用于实际的伺服或定向反馈目的的R/D转换器,并不一定要使用一个M变量。I变量(Ix10, Ix81, I8x, I9x)也可以完成这种任务。然而,即使你只将R/D转换器用于这种目的,通常也要为R/D转换器分配M变量以进行设置和诊断。

范例 M100->TWR:0,0 M99->TWR:4,5

参考 M变量(计算的特点)

在线命令**M{常量},M{常量}->,M{常量}->TWS:{地址}** 旋转变压器-到-数字转换器板(**附件7 ACC-8D**)手册

M{常量}->TWS:{地址}

功能: 串联多路选择器M变量定义。

范围: 全局

语法: M{常量} [..{常量}]->TWS[:] {多路地址}

这里

{常量}是一个从0到1023的整数,它代表M变量的号码。

第二个{常量}是可选的,不能小于第一个{常量},它代表变量序列的结束号码。

{多路地址}是一个常整数。它被4除后应在0到124之间,代表32位输入或输出字

中第一个字节的多路选择的地址。把**{多路地址}**加1可将其标为一个只读变量,将 其加2则成为一个只写变量。

备注: 本命令使PMAC定义指定的指向32位输入或输出字的一个**M**变量或变量序列。这些字可以用附件**34x**板上的"多路"端口逐个选择。

注意: 附件34x板上"多路"端口的每一位不能分别直接分配一个M变量。只有输入或输出的32位字(端口)才能被访问。

多路选择端口指定的地址必须与ACC-34x上DIP开关设置的地址一致。ACC-34x手册中列出了所有可能性的表。

整个字必须全部是输入或输出。在开机/复位时,所有ACC-34x的字都被软件设置 为输入(如果硬件被设置为输出,则所有的输出将为OFF----上拉为供电电压)。以 M变量对该端口上一个I/O字的任意一个后继写操作会自动将整个字变为输出字,此时由写入该字的值确定每一位为**ON**还是**OFF**。

如果一个自身设置或试图设置为输出,那么任一后继的读操作将使整个字变为一个输入字,并使每个硬件输出为**OFF**。所以,当用这些**M**变量进行工作时必须遵守一下规则:

不要用该M变量形式写一个设置为输入的字。 不要用该M变量形式写一个设置为输入的字。

由于当一个TWS型M变量被用来直接指向一个附件34x端口的基地址时(例如: M300->TWS:40)读和写都是允许的,所以坚决建议不要使用它们。当倒数第一和第二个有效地址位都为零时(例如:十进制的40=二进制的01000000),读和写是允许的。

在这种情况下,任一个对输出端口的偶然读(如通过执行程序的监视窗口)将会使所有的输出晶体管被置为off(输出被上拉为供电电压)!另一方面,对一个输入口的写操作将自动设置该口为输出端口!所以可用M变量定义的0位和1位有意地禁止读或写功能,这样比较安全和利于预控。设置这两位将赋予TWS M变量只读或只写形式。

把**{多路地址}**置为一个合法字节码值(见ACC34x手册表1的第二列)加1,可将一个指向输入端口的M变量定为只读。任何对这种状态下的M变量(其地址的0位为1)的写操作都会被PMAC的固件自动停止。对于一个输出口,{多路地址}应为一个合法字节码值(见ACC34x手册表1的第二列)加2。任何对这种状态下的M变量(其地址的0位为1)的读操作都会返回零,而真正的读则被PMAC的固件自动停止。

由于你不能直接访问附件**34x**板上"多路"端口的位,同时因为对数据出入PMAC进行即使要耗费相对较场的时间(对一个端口进行一个**32**位读或写操作大约会占用PMAC的后台时间片中的**64**毫秒),所以最好在内部存储器中保留一个"映像"。映像变量最好是一个**32**位或**48**位定点的**M**变量,但也可以是一个**48**位浮点**P**或**Q**变量。

下面给出在程序中使用**TWS M**变量的最佳过程。在一系列操作开始时,输入字(**TWS M**变量)应被复制为它的映像变量。之后,每个操作就可以用映像变量完成,而不用要求PMAC为每个操作真正读或写**I/O**端口。

输出字首先分配到它的映像变量,然后在一系列操作之后一次性复制到真正的输出字。这一过程允许**TWS M**变量最有效和灵活的应用。

这一类型的变量只能用于后台任务(PLC和PLCC 1-32),它们不能在前台任务中使用(运动程序、PLC和PLCC 0)。

范例 以下面的定义用M101将#1板的B端口定为输出。*由于端口的偶然读操作不被保护, 所以并不推荐这种格式*(可考虑使用只写形式)。

M100->TWS:4 ; ACC-34x的SW1开关均为ON, 其 ; B端口(BIO 0-31) **不**被置为只写。

要把上面的端口B编址为只写地址形式,可以使用下面的定义:

M100->TWS:6 ; ACC-34x的SW1开关均为ON, ; 其B端口(BIO 0-31)被置为只

: 写(6=4+2)。

用**M99**将板**#1**的端口A定为输入,可以使用下面的定义。*由于端口的偶然读操作不被保护,所以并不推荐这种格式*(可考虑使用只读形式)。

M99->TWS:0 ; ACC-34x的SW1开关均为ON, 其 ; A端口(AIO 0-31) **不**被置为只读。

要把上面的端口B编址为只读地址形式,可以使用下面的定义:

M100->TWS:1 ; ACC-34x的SW1开关均为ON,

: 其A端口(AIO 0-31)被置为只

;读(1=0+1)。

还有另外的一个例子: 用M300将板#6的端口A定为输入, 我们可以使用下面的定义:

M300->TWS:41 ; ACC-34x的SW1开关为ON, ON,

; OFF, ON, ON。其A端口(AIO 0-31)

;被置为只读(41=40+1)。

参考 M变量(计算的特点)

在线命令M{常量}, M{常量}->, M{常量}->TWR:{地址} 串行I/O多路选择器板(ACC-34)手册

M{常量}->X/Y:{地址}

功能: 短字M变量定义

范围: 全局

语法: M{常量} [..{常量}]->X[:] {地址}, {偏移量} [, {宽度} [, {格式}]]

M{常量} [..{常量}]->Y[:] {地址}, {偏移量} [, {宽度} [, {格式}]]

这里

{常量}是一个从0到1023之间的整数,代表M变量个号码。

第二个{常量}是可选的,不能小于第一个{常量},它代表变量序列的结束号码。

{地址}是一个从0到65535(十六进制的\$0到\$FFFF)之间的整数。

{偏移量}或者是一个从0到23的常整数,代表**M**变量中使用的字的起始(最小有效)位;或者是24,指定使用所有24位。

(宽度)(可选)是设置为**{1,4,8,12,16,20,24}**中的一个常整数,代表**M**变量中使用字的位的数目。如果没有指定**{宽度}**,则赋为**1**。

{格式}(可选)是集合[U, S, D, C]中的一个字母。他指定PMAC怎样解释数值: (U=无符号整数, S=带符号整数, D=二进制编码表示的十进制数, C=补码形式的二进制编码的十进制数)。如果**{格式}**没有被指定,则默认为U。

备注: 本命令使PMAC定义指定的M变量或M变量序列,这些M变量指向PMAC的一个数据存储器中的一个位置(PMAC的数据存储器对分为两部分,X和Y)。在这种格式中,变量的宽度可以从1位到24位,也可以用多种不同的方法进行解码,所以位偏移、位宽度和解码格式都必须被指定(位宽度和解码格式可以用默认值)。定义包括:一个字母(**X或Y**)、一个可选的冒号(:)、字地址、起始位(**偏移**)、一个可选的位宽度和一个可选的用来指定格式的字母。

位宽度和位偏移的合法值之间有一定的关系。下表列出了**{宽度}**的可能值和相对于每个**{宽度}**值的合法的**{偏移量}**的值。

{宽度} {偏移量} 1 0..23 4 0,4,8,12,16,20 8 0,4,8,12,16 12 0,4,8,12 16 0,4,8 20 0,4 24 0

{格式}与1位的M变量无关,并且不应包括它们。如果没有指定格式,默认为"U"。

范例 机器输出1

M1->Y:\$FFC2,8,1 ; 1位(完整表示) M1->Y\$FFC2,8 ; 1位(简短表示)

编码器1捕捉/比较寄存器

M103->X:\$C003,0,24,U ; 24位(完整表示) M103->X\$C003,24 ; 24位(简短表示)

DAC1输出寄存器

M102->Y:\$C003,8,16,S ; 16位值

M102->Y49155,8,16,S ; 同上,十进制地址

参考 M变量(计算的特点)

在线命令M{常量}, M{常量}->, M{常量}->D:{地址}, M{常量}=(表达式)程序命令M{常量}={表达式}

MACROAUX

功能: 报告或写MACRO辅助参数值。

范围: 全局

语法: MACROAUX {节点码} {参数码} [={常量}]

MX {节点码} {参数码} [={常量}]

这里

{节点码}是一个从0到15的常整数,指定了节点的从属码。

{参数码}是一个从**0**到**65535**的常整数,为节点指定辅助参数码(**2**到**254**对应写操作)

{常数}是一个从-32768到32767的常整数,代表将被写到指定参数中得值

备注: 本命令允许PMAC越过MACRO环由从属节点中读或写辅助寄存器的值。在命令中必须指定从属节点的节点码和辅助参数码。另外如果是一条写命令,还必须指定要写入寄存器的值。

作为一条读命令使用时(命令中没有"={常量}"),如果PMAC被要求查询它的某个参数的值,其将以ASCII形式向主机报告指定参数的值。

在一个命令行中只能进行一个单节点的辅助存取。

如果要访问MACRO节点n的辅助寄存器,I1000的位n必须被置为1。

如果从属节点返回一条错误信息,或者其在32个伺服循环中没有响应,PMAC将会报告ERR008。当报告一条这样的MACRO辅助通讯错误时,全局状态寄存器X:\$0003位5将被置为1。寄存器X:\$0798将保存错误值:如果是超时错误它被置为\$010000;如果是从属节点报告错误,它将被置为\$xxxxFE(xxxx是由从属节点报告的错误码)。

TANCE OF THE PROOF OF THE PRO

MACROAUX1,24 ; 查询节点1的参数24的值

2000 ; PMAC报告值

参考 在线命令MACROAUXREAD,MACROAUXWRITE

MACROAUXREAD

功能: 读取MACRO辅助参数值。

范围: 全局

语法: MACROAUXREAD {节点码} {参数码} {变量} MXR {节点码} {参数码} {变量}

这里

{节点码}是一个从**0**到15的常整数,指定了节点的从属码。

{参数码}是一个从0到65535的常整数,为节点指定辅助参数码。

{变量}是一个PMAC变量名(**Ⅰ,P,Q或M**),该变量用来存储复制参数的值。

备注: 本命令允许PMAC越过MACRO环由从属节点中读辅助寄存器的值。在命令中必须指定从属节点的节点码、辅助参数码和用来接受返回值的PMAC的变量的名称。在一个命令行中只能进行一个单节点的辅助存取。

如果要访问MACRO节点n的辅助寄存器,I1000的位n必须被置为1。

如果从属节点返回一条错误信息,或者其在32个伺服循环中没有响应,PMAC将会报告ERR008。当报告一条这样的MACRO辅助通讯错误时,全局状态寄存器X:\$0003位5将被置为1。寄存器X:\$0798将保存错误值:如果是超时错误它被置为\$010000;如果是从属节点报告错误,它将被置为\$xxxxFE(xxxx是由从属节点报告的错误码)。

芯例 MACROAUXREAD1,24,P1 ; 将节点1的参数24的值读到P1中 MXR5,128,M100 ; 将节点5的参数128的值读到M100中

参考在线命令MACROAUX,MACROAUXWRITE
PLC程序命令MACROAUXREAD,MACROAUXWRITE

MACROAUXWRITE

功能: 写MACRO辅助参数值。

范围: 全局

语法: MACROAUXWRITE {节点码} {参数码} {变量} MXW {节点码} {参数码} {变量}

这里

{节点码}是一个从0到15的常整数,指定了节点的从属码。

{参数码}是一个从2到253的常整数,为节点指定辅助参数码。

{变量}是一个PMAC变量名(**Ⅰ,P,Q或M**),该变量用来存储将写入参数的值。

备注: 本命令允许PMAC越过MACRO环向从属节点中写辅助寄存器的值。在命令中必须指定从属节点的节点码、辅助参数码和包含写入值的PMAC的变量的名称。 在一个命令行中只能进行一个单节点的辅助存取。

如果要访问MACRO节点*n*的辅助寄存器,**l1000的位***n*必须被置为1。

如果从属节点返回一条错误信息,或者其在32个伺服循环中没有响应,PMAC将会报告ERR008。当报告一条这样的MACRO辅助通讯错误时,全局状态寄存器X:\$0003位5将被置为1。寄存器X:\$0798将保存错误值:如果是超时错误它被置为\$010000;如果是从属节点报告错误,它将被置为\$xxxxFE(xxxx是由从属节点报告的错误码)。

MXW5,128,M100 ; 将**M100**的值写到节点5的参数128中

参考 在线命令MACROAUX,MACROAUXREAD PLC程序命令MACROAUXREAD,MACROAUXWRITE

MFLUSH

功能: 清除挂起的同步M变量赋值。

范围: 指定坐标系

语法: MFLUSH

备注: 本命令允许用户清除同步M变量赋值命令,这些赋值命令是放在堆栈中用于后续运

动的预定运行的(并没有执行)。作为一个在线命令,它对于确保在某个程序已经结

束后挂起的输出不被执行是非常有用的。

范例 / ; 停止程序的执行

MFLUSH ; 清除M变量堆栈

B1R ; 开始另一个程序, 先前挂起的M变量

; 不会被执行

参考 程序命令M(常量)=={表达式}

M{常量}&={表达式},

M{常量}|={表达式}, M{常量}^={表达式},

O{常量}

功能: 打开环输出。

范围: 指定电机

语法: O{常量}

这里

{常量}是一个范围为+/-100的浮点值,它代表输出的数量(用电机的**Ix69**的百分数表示)

备注: 本命令使PMAC将电机设置为开环模式,并强迫一个输出为指定的数量(用电机最大输出参数(**lx69**)的百分比表示)。本命令通常用来进行设置和诊断(例如: 一个正的**○**命令必须使位置在正方向上计算,或闭环控制不能被建立),但它也可以用于真正的应用。

如果该电机不是由PMAC换向的,本命令将会在一个DAC上为电机建立一个DC输出电压。如果该电机由PMAC换向,则换向算法继续工作,并且指定的输出量根据瞬时换向相位角被分配到电机对应的两个DAC输出上。

如果指定值超出了+/-100的范围,将以lx69的+/-100%输出。

该电机的闭环控制可以用J/命令重新建立。如果电机已经在开环模式中运动以后,最好首先用一个**OO**命令停止电机。

要进行一个可变的O命令,可以为过滤结果寄存器(X:\$003A等)定义一个M变量。向电机发送一个OO命令,使其进入开环模式,然后为M变量赋一个可变值。这种方法也可以用于PMAC换向的电机。

如果电机所在的坐标系正在运行一个运动程序,PMAC将会拒绝本命令(如果I6为1或3,报告ERR001)。

范例 O50 ; 为选址电机开环输出**lx69**的50%

#2033.333 ; 为电机2开环输出**lx69**的1/3

OO ; 开环输出0

J/ ; 重新建立闭环控制

参考 在线命令J/, K

内存映像寄存器X:\$003A, X:\$0076等

推荐的M变量定义Mx71

OPEN PLC

功能: 打开一个PLC程序缓冲。

范围: 全局

语法: OPEN PLC {常量}

这里

{常量}是一个从0到31的整数,代表要被打开的**PLC**程序。

备注: 本命令使PMAC打开指定的PLC程序缓冲区,以便进行输入和编辑。这样就使后续的合法的PLC程序行可以输入到该缓冲区中。当程序输入完成后,应该用CLOSE命令关闭缓冲以免后面的行进入该缓冲。

当本命令被发送时,其它的程序缓冲(PLC,固定或旋转运动)不能被打开,否则 PMAC会报告ERR007(如果I6=1或3)。最好在每个OPEN命令前都先执行一个 CLOSE命令,以确保没有其它的缓冲时打开。

PLC 0-15可以用密码进行保护。如果PLC被密码保护,而PMAC又没有获得正确的密码,它将会拒绝本命令(报告ERR002若I6=1或3)。

打开一个PLC程序缓冲会自动禁止该PLC程序,其它的PLC程序和运动程序则将会继续执行。输入后关闭PLC程序缓冲并不会重新允许该程序。要想重新允许该程序,必须使用ENABLE PLC命令或将PMAC复位(I5中的保存值将会使该PLC程序执行)。

范例 CLOSE

; 确保其它缓冲被关闭

DELETE GATHER

; 确保内存空

OPEN PLC 7

; 打开缓冲, 禁止程序

CLEAR IF (M11=1) ;擦除已有内容 ;输入新内容

• • •

CLOSE ENABLE PLC 7 ; 在程序完成后关闭缓冲 ; 重新允许程序

参考 PLC程序特点

I变量I5

在线命令CLOSE, DELETE GATHER, ENABLE PLC

OPEN PROGRAM

功能: 为输入打开一个固定运动程序缓冲。

范围: 全局

语法: OPEN PROGRAM {常量}
OPEN PROG {常量}

这里

{常量}是一个从**0**到32767的整数,代表要被打开的运动程序。

备注: 本命令使PMAC打开指定的固定(非旋转)运动程序缓冲,以便进行输入和编辑。 这样就使后续的合法的固定运动程序行可以输入到该缓冲区中。当程序输入完成后, 应该用**CLOSE**命令关闭缓冲以免后面的行进入该缓冲。

当本命令被发送时,其它的程序缓冲(PLC,固定或旋转运动)不能被打开,否则 PMAC会报告ERR007(如果I6=1或3)。最好在每个OPEN命令前都先执行一个 CLOSE命令,以确保没有其它的缓冲时打开的。

当本命令被发送时,所有坐标系中的程序都不能运行,否则PMAC会报告**ERR001** (如果**I6=1**或3)。而且只要固定运动程序缓冲被打开,所有坐标系中的程序就都不能运行,否则PMAC会报告**ERR015**(如果**I6=1**或3)。

PROG 1000-32767可以用密码进行保护。如果PROG被密码保护,而PMAC又没有获得正确的密码,它将会拒绝本命令(报告ERR002若I6=1或3)。

在任一个固定运动程序缓冲被打开后,每个坐标系在运行一个运动命令之前都必须用命令**B{常量}**将其指向一个运动程序(否则PMAC将报告E**RR015**若**I6=**1或3)。

范例 CLOSE ; 确保其它缓冲被关闭

DELETE GATHER DELETE TRACE ; 确保内存空 **OPEN PROG 255** ; 打开缓冲,禁止程序

 CLEAR
 ;擦除已有内容

 X10 Y20 F5
 ;输入新内容

...

 CLOSE
 ; 在程序完成后关闭缓冲

 &1B255R
 ; 指向该程序并运行

参考 编写一个运动程序

在线命令CLEAR,CLOSE,DELETE GATHER,DELETE TRACE 程序命令说明

OPEN ROTARY

功能: 为输入打开所有存在的旋转运动程序缓冲区。

范围: 全局

语法: OPEN ROTARY

OPEN ROT

备注: 本命令使PMAC打开所有现存的旋转运动程序缓冲区(用**DEFINE ROTARY**命令创建),以便进行输入。这样就使后续的合法的旋转运动程序行可以输入到缓冲区中(该缓冲由命令发出时选址的坐标系确定)。(分支和循环命令不要在旋转程序缓冲区中使用)。

当本命令被发送时,其它的程序缓冲区(PLC,固定或旋转运动)不能被打开,否

则PMAC会报告**ERR007**(如果**I6**=1或3)。最好在每个**OPEN**命令前都先执行一个**CLOSE**命令,以确保没有其它的缓冲时打开的。

<CTRL-U>命令的功能与OPEN ROTARY时一样的。

当旋转缓冲被打开时,使坐标系指向该旋转缓冲的B0命令不能使用,因为PMA C会将其解释为一个B轴运动命令。

范例 &2 DEFINE ROT 100

; 创建坐标系2的旋转缓冲

&1 DEFINE ROT 100

;创建坐标系1的旋转缓冲

&1 B0 &2 B0

;将两个坐标系都指向该旋转缓冲

OPEN ROT

; 打开缓冲

&1 X10 Y10 F5 &2 X30 Y30 F10 ;写入坐标系1的缓冲 ;写入坐标系2的缓冲

&1R &2R

: 开始执行两个缓冲

参考

旋转运动程序(编写一个运动程序)

在线命令<CTRL-L>,<CTRL-U>,CLOSE,DEFINE ROT,B{常量},R

P

功能: 报告电机位置。

范围: 指定电机

语法: P

备注:

本命令使PMAC向主机报告选址电机的当前实际位置,按数字测算,精确到小数点

后一位。

PMAC报告的值是这样计算的: *实际位置寄存器*加上*位置偏差寄存器*再加上*补偿校正寄存器*,如果**lx05**的位16是1(手轮偏移模式),还要减去*主位置寄存器*。

范例

Р

; 查询选址电机的位置

1995

; PMAC响应

#1P

; 查询电机1的位置

-0.5

: PMAC响应

#2P#4P

;查询电机2和4的位置

9998

: PMAC先返回电机2的位置

10002

: PMAC后返回电机4的位置

参考

在线命令**<CTRL-P>**,**F**,**V**

推荐的M变量定义Mx62, Mx64, Mx67, Mx69

内存映像寄存器D:\$002B, D:\$0813, D:\$002D, D\$0046等

P{常量}

功能: 报告当前的P变量值。

范围: 全局

语法:

P{常量} [..{常量}]

这里

{常量}是一个从0到1023的整数,代表P变量的号码。

可选的第二个{常量}不能小于第一个,它代表变量序列的结束号码。

备注: 本命令使PMAC报告指定P变量或P变量序列的当前值。

范例: P1 ; 主机查询变量的值

25 ; PMAC响应

P1005 3.44444444 **P100..102** 17.5

17.5 -373 0.005

参考 P变量(计算的特点)

在线命令I{常量}, M{常量}, Q{常量}, P{常量}={表达式}

P{常量}={表达式}

功能: 为P变量赋一个值。

范围: 全局

语法: P{常量} [..{常量}]={表达式}

这里

{常量}是一个从0到1023的整数,代表P变量的号码。

可选的第二个{常量}不能小于第一个,它代表变量序列的结束号码。

{表达式}包含要赋予P变量的值。

备注: 本命令使PAMC将等号右边的值赋予指定的P变量或P变量序列。

范例: P1=1

P75=P32+P10 P100..199=0 P10=\$2000

P832=SIN(3.14159*Q10)

参考 P变量(计算的特点)

在线命令 | {常量}={表达式},

M{常量}={表达式},

Q{常量}={表达式},

P{常量}

程序命令 P{常量}={表达式}

PASSWORD={字符串}

功能: 输入/设置程序密码。

范围: 全局

语法: PASSWORD={字符串}

这里

{字符串}是一系列非控制**ASCII**字符(其值从十进制**32**到十进制**255**)。密码字符 串是区分大小写的。

备注: 命令允许用户输入卡的密码,一旦输入正确后可以改变密码。没有输入正确的密码

前,PMAC将不会打开或列出号码大于等于1000的运动程序或PLC程序0-15。如果被要求这么做,PMAC将会返回一个错误(如果**I6**为1或3,报告ERR002)。

默认的密码是空密码(即列出程序不需要密码)。这是在卡运出工厂时的状态,而用一个**\$\$\$*****重新初始化命令后也可以使其变为这个状态。当时用空密码时,你会被自动认为已经在启动/复位时输入了正确的密码。

如果你已经正确地输入了密码(对于空密码来说总是如此)。PMAC会将 PASSWORD={字符串}解释为改变密码命令,你可以将密码该为希望的任意值。当 密码改变后,它已经被自动"匹配"并且主计算机能够访问被保护的程序。

注意:密码不需要引号。如果你在第一次输入密码字符串时使用了引号,你必须每次匹配密码字符串时都使用它们。

如果在前一次启动/复位后你还没有输入正确的密码,PMAC会将PASSWORD={字符串}解释为一个匹配已有密码的尝试。如果命令正确地匹配了现有密码,PMAC将其视为一个合法命令,之后主计算机能够访问被保护的程序直到PMAC被复位或启动。如果命令不能正确地匹配现有密码,PMAC返回一个错误(若I6=1或3则报告ERR002),主计算机也不能访问被保护程序。主计算机可以自由地进行匹配现有密码的尝试。

已有密码不能够被读去。如果密码被忘记了担忧希望访问被保护程序,必须用一个 \$\$\$***命令重新初始化卡,该命令会清除密码和所有的程序缓冲。之后,程序必须被重新载入,并输入一个新的密码。

范例 {从启动/复位开始,密码为空}

 LIST PLC 1
 ; 察看被保护程序的清单

 P1=P1+1
 ; 由于没有密码, PMAC响应

ETURN

PASSWORD=Bush; 将密码设置为 "Bush"LIST PLC 1; 察看被保护程序的清单

P1=P1+1; 由于密码在改变时已被匹配, 所以

RETURN ; PMAC响应

\$\$\$; 卡复位

LIST PLC 1 ; 察看被保护程序的清单

ERR002 ; 由于没有输入密码,PMAC拒绝命令

PASSWORD=Reagan; 试图输入密码ERR002; PMAC拒绝错误的密码PASSWORD=BUSH; 试图输入密码

ERR002 ; PMAC拒绝错误的密码(大小写不对)

PASSWORD=Bush ; 试图输入密码

; PMAC接受正确的密码

LIST PLC 1 ; 察看被保护程序的清单

P1=P1+1; 由于密码已经匹配,所以PMAC响应

RETURN

PASSWORD=Clinton; 像密码改为 "Clinton"LIST PLC 1; 察看被保护程序的清单

P1=P1+1; 由于密码在改变时已被匹配,所以

RETURN ; PMAC响应

\$\$\$; 卡复位

PASSWORD=Clinton ; 试图输入密码

; PMAC接受正确的密码

LIST PLC 1 : 察看被保护程序的清单

P1=P1+1; 由于密码已经匹配, 所以PMAC响应

RETURN

参考 在线命令LIST, LIST PC, LIST PE, OPEN

PC

功能: 报告程序计数器。

范围: 指定坐标系

语法: PC

备注: 本命令使PMAC报告即将被计算的运动程序号码和该程序中命令行的偏移地址(在选址坐标系中)。如果本命令位于一个**GOSUB**或**CALL**跳转(最多15层)中,

它也会报告必须跳回(用RETURN)的所有行的程序号码和地址偏移。

在返回的报告中冒号后面的数不是一个行号,作为一个地址偏移量,它是从程序顶端开始计算的内存的字数。当使用逗号分隔符时,LIST PROGRAM命令列出程序或程序某部分中每行的地址偏移。LIST PC命令能够列出从计算点开始的程序行的地址偏移。

因为PMAC在一系列连续的运动之前进行计算,所以PC(程序计算)命令返回的程序行通常比PE命令报告的要早一些。

如果坐标系没有指向任何运动程序,PMAC将返回一个错误(ERR003若I6=1或3)。最初的指向必须用B{常量}命令进行。

范例 PC ; 准备从程序1的顶端开始执行

P1:0

PC ; 准备从程序76的第22个字开始执行

P76:22

LIST PC ;程序行位于程序76的第22个字

P76:22:X10Y20

PC ; 执行将返回到程序3地址12

P1001:35>P3.12

参考 在线命令B{常量}, LIST, LIST PC, LIST PE, LIST PROGRAM, PE

PE

功能: 报告程序执行指针。

范围: 指定坐标系

语法: PE

备注: 本命令使PMAC报告在选址坐标系中正在执行的程序运动的运动程序号和地址偏移。这与报告将被计算的下一个运动的程序号和地址偏移的PC命令很相似。因为PMAC在一系列连续的运动之前进行计算,所以PC命令返回的程序行通常比PE

命令要早几步。

如果坐标系没有指向任何运动程序,PMAC将返回一个错误(ERR003若I6=1或3)。最初的指向必须用B{常量}命令进行。

范例 PE

P1:2 **PE** P1:5

参考 在线命令B{常量}, LIST, LIST PC, LIST PE, PC

PMATCH

功能: 重新匹配轴位置和电机位置。

范围: 指定坐标系

语法: PMATCH

备注: 本命令使PMAC为坐标系重新计算轴起始位置以匹配电机的当前指定位置(通过转换轴定义语句等式,并且解释轴位置)。

一般情况下不需要这么做。然而,如果在前一次轴运动或返回原点运动后,运行了一个电机运动函数(例如进给运动、开环运动、或由终止及限制命令产生的停止),PMAC将不会自动得知这个轴的位置已被改变。如果之后不使用PMATCH命令就试图进行一个轴的运动,PMAC将会在它的计算中使用错误的轴起始位置。

另外,在使用一个绝对传感器时,也应该在第一个程序运动前执行一个PMATCH 命令,以使起始轴位置和(非零)电机位置相匹配。

如果**PMATCH**函数没有被执行,**PMAC**将使用前一次的轴目标位置作为以后轴运动计算的起始位置。该位置不一定和电机的当前指定位置相同。

运动程序中,在**DWELL**前后用一个**CMD"PMATCH"**就可以执行**PMATCH**功能。 这对于在程序中改变坐标系配置的情况(如新的轴被引入或后继模式被改变)是十分有用的。

如果一个以上的电机被定义到一个给定轴(如在一个起重机系统中),PMAC在计算中将使用较低号码电机的指定位置。

如果I14被置为1,在每次程序开始运行时PMATCH功能都将会被自动执行。大多数的用户会希望使用I14=1,这样他们就不必为何时需要执行PMATCH而操心。

范例 #1J+ ; 电机1进给

#1J/ ; 停止进给

PMATCH ; 将轴位置与当前电机位置匹配

B200R : 执行程序200

OPEN PROG 10 CLEAR

...

CMD"&1#4->100C" ; 将C轴引入坐标系

DWELL100

CMD"PMATCH"

; 发出PMATCH使C轴拥有正确的起始位置

DWELL100

C90

...

参考 进一步的位置处理(设置一个电机)轴,坐标系(设置一个坐标系) I变量**I14,lx06**

PR

功能: 报告剩余的旋转程序。

范围: 指定坐标系

语法: PR

备注: 本命令使PMAC报告选址坐标系中已经被输入旋转缓冲但还没有被执行(剩余程序)的程序行的数目。本命令可以用来察看是否可以向缓冲发送新行。只有当旋转

程序缓冲打开时才会返回正确的值。

范例 B0 ; 指向旋转缓冲 **OPEN ROT** : 打开旋转缓冲

X10F10;输入第一行X20;输入第二行X30;输入第三行X40;输入第四行

 PR
 ; 查询 "剩余程序"

 4
 ; PMAC报告还剩4行

 R
 ; 开始运行程序

 PR
 ; 查询 "剩余程序"

 2
 ; PMAC报告还剩4行

参考 旋转程序缓冲(编写一个运动程序)

BREQ中断(使用中断----编写一个主机通讯程序)

I变量**I16**,**I17**

Q

功能: 运动结束后推出程序。

范围: 指定坐标系

语法: Q

备注: 本命令是当前选址坐标系在当前运行的程序或已被计算出的运动结束之后终止程

序。程序计数器被设置为下一程序行,所以可以从该除用R或S命令继续执行程序。

本命令和/命令相似,/命令是在当前运行程序的结束处停止执行。

范例 B10R ; 指向程序10的顶端并开始运行

 Q
 ; 终止执行

 R
 ;继续执行

 Q
 ;再次终止执行

S ;继续执行一个单运动

参考 停止命令(使你的应用安全)

控制板端口STEP/输入(将PMAC与机器连接)

JPAN接口9脚

在线命令<CTRL-Q>, A, H, K, /, \

Q{常量}

功能: 报告Q变量值。

范围: 指定坐标系

语法: Q{常量} [..{常量}]

这里

{常量}是一个从0到1023的整数,代表**Q**变量的号码。

可选的第二个{常量}不能小于第一个,它代表变量序列的结束号码。

备注: 本命令使PMAC报告选址坐标系中指定P变量或P变量序列的当前值。

范例 Q10

35

Q255

-3.4578

Q101..103

0 98.5 -0.3333333333

参考 Q变量(计算的特点)

在线命令I{常量}, M{常量}, P{常量}, Q{常量}={表达式}

Q{常量}={表达式}

功能: Q变量赋值。

范围: 指定坐标系

语法: Q{常量} [..{常量}]={表达式}

这里

{常量}是一个从0到1023的整数,代表**Q**变量的号码。

可选的第二个{常量}不能小于第一个,它代表变量序列的结束号码。

{表达式}包含要赋予Q变量的值。

备注: 本命令使PAMC将等号右边的值赋予选址坐标系中指定的Q变量或Q变量序列。

如果当本命令发送到PMAC时,一个运动程序缓冲是打开的,该命名将会被输入该

缓冲并在以后执行。

范例 Q100=2.5

Q1..10=0

参考 **Q**变量(计算的特点)

在线命令|{常量}={表达式},

M{常量}={表达式},

P{常量}={表达式}, Q{常量}

程序命令Q{常量}={表达式}

R

功能: 执行运动程序。 **范围**: 指定坐标系

语法: R

备注: 本命令使PMAC的选址坐标系开始连续执行运动程序,该程序是由坐标系的程序计数器指定的。另一方面,在发出"\"或"H"命令后,本命令也会恢复操作(即使没有程序在运行)。对程序计数器的指定是开始使用**B{常量**}命令进行的。

为了使PMAC接受本命令,坐标系必须处于正确的状态。否则PMAC会拒绝命令,如果**I6**为1或3则返回一个错误码。下列状态会使PMAC拒绝本命令(后面列出的是解决方法)

- -为坐标系的电机设置了限定(ERR012);清除限定
- -另一个运动正在进行(**ERR011**);停止运动(例如使用**J**/)
- -坐标系中有开环电机(ERR012);用J/或A闭合伺服环
- -坐标系中有不工作(Unactivated)的电机(ERR013),将lx00改为1或把电机从 坐标系中去掉
- -坐标系中没有电机(ERR014);至少在坐标系中放一个电机
- -固定运动程序缓冲打开(ERR015): 关闭缓冲并指向程序

-没有指向程序(ERR015);用B命令指向程序

-程序结构不正确(ERR016):修改程序结构

-程序位置与用/或\命令停止时的不一样(ERR017):用J=命令移动回停止位置

范例 &1B1R ; 坐标系1指向程序1并执行

&2B200.06 : 坐标系2指向程序200的N6000并执行

Q ; 退出这个程序

R ; 从停止点继续执行程序

H; 在程序中停止

R ; 从停止点继续执行程序

参考 控制板端口START/输入(将PMAC与机器连接)

运行一个运动程序(编写一个运动程序)

I变量I6

在线命令<CTRL-R>, A, H, Q, S

JPAN接口8脚

R[H]{地址}

功能: 报告指定地址的内容。

范围: 全局

语法: R[H] {地址} [, {常量}]

这里

{地址}包括一个字母X,Y或L;一个可选的冒号(:);以及一个从0到65535的整数值(十六进制的**\$0000到\$FFFF**)。它指定要读取得PMAC内存或**I/O**起始地址。

{常量}(可选)使一个从1到16的整数,它指定要读取的连续内存地址的数目。如果没有指定,PMAC默认其值为1。

备注: 本命令使PMAC向主机报告指定地址或地址序列的内存字的内容(它本质上是一个PEEK命令)。它可以指定PMAC X内存中的短(24位)字、Y内存中的短(24位)字、或包括X和Y内存的长(48位)字(X字更重要)。这是由命令中X、Y或L地址前缀分别指定的。

如果在命令中**R**后面有一个字母**H**, **PMAC**会以无符号的十六进制格式报告寄存器内容,用**6**位表示一个短字,**12**位表示一个长字。如果没有字母**H**, **PMAC**以带符号的十进制格式报告寄存器内容。

范例 RHX:49152 ; 16进制查询X寄存器49152(\$C000)

8F4017 ; PMAC以无符号16进制响应(注意没有"\$")

RHX:\$C000 : 16进制查询X寄存器\$C000(49152)

8F4017 : PMAC以无符号16进制响应

RX:49152 ; 10进制查询同一个寄存器的内容 -7389161 ; PMAC以带符号10进制响应

RX:\$C000; 10进制查询同一个寄存器的内容-7389161; PMAC以带符号10进制响应RX0; 10进制查询伺服环计数器的内容2953211; PMAC以带符号10进制响应

RL\$0028 : 10进制查询#1命令位置寄存器的内容

3072000 : PMAC响应

RHY1824,12 ; 查询转换表的配置字

00C000 00C004 00C008 00C00C 00C010 00C014 00C018

00C01C 400723 000295 000000 000000 ; PMAC以16进制响应

参考 PMAC内存映射(计算的特点)

在线命令**W{地址}** 内存和**I/O**映射说明

S

功能: 执行运动程序一个运动("单步")。

范围: 指定坐标系

语法: S

备注: 本命令使PMAC的选址坐标系开始单步执行运动程序,该程序是由坐标系的程序计数器指定的。对程序计数器的指定是开始使用**B{常量**}命令进行的。

当**I52**为默认的0值时,一个单步命令使程序执行到其中的下一个运动或**DWELL**命令,甚至可能由多个程序行组成的。

当**I52**被置为1时,即使一个程序行中没有运动或**DWELL**命令,一个单步命令仅仅执行一个程序行。而当一个程序行中有多个**DWELL**或**DELAY**命令时,一个单步命令将仅仅执行一个**DWELL**或**DELAY**命令。

而不论**I53**的值为多少,如果程序在单步运行时遇到一个**BLOCKSTART**语句,它都将一直运行直到遇到一个**BLOCKSTOP**语句。

如果在本命令发出时,坐标系已经在运行一个运动程序,命令将会把程序置为单步运行模式。因此执行在最近一个计算出的运动结束后会停止。在这种情况下,本命令的作用于**Q**命令时一样的。

为了使PMAC接受本命令,坐标系必须处于正确的状态。否则PMAC会拒绝命令,如果**I6**为1或3则返回一个错误码。导致拒绝R命令的那些状态也会使PMAC拒绝一个**S**命令,参考**R**命令说明中的那些状态。

范例 &3B20S; 坐标系3指向程序20的顶端并开始单步执行

P1 ; 查询P1的值 1 ; PMAC响应

 S
 ; 单步运行下一步程序

 P1
 ; 再次查询P1的值

-3472563 : PMAC响应----可能有错误

参考 控制板端口**STEP/**输入(将PMAC与机器连接)

运行一个运动程序(编写一个运动程序)

I变量I6, I52

在线命令<CTRL-S>, A, H, Q, R, /, \

程序命令{轴}{数据},BLOCKSTART,BLOCKSTOP,DWELL,DELAY

SAVE

功能: 将设置的参数拷贝到非易失内存中。

范围: 全局

语法: SAVE

备注: 本命令使PMAC从激活内存中将设置信息拷贝到非易失内存中,以使这些信息在 关机或复位时能被保留。它的正确操作取决于PMAC的类型。

对于带后备电池RAM的"标准"PMAC,SAVE命令只储存基本的配置信息:I变

量、编码转换表输入项和VME/DPRAM地址输入项。这些信息在标准启动/复位操 作是从闪存重新复制回活动内存中。用户程序、缓冲和定义只放在后备电池支持的 RAM中,没有必要保存它们。

对于带后备闪存RAM的"可选"PMAC, SAVE命令将所有的用户配置信息,包 括程序、缓冲和定义到闪存中。这些信息在标准启动/复位操作是从闪存重新复制 回活动内存中。这意味着所有在PMAC活动内存中没有保存到闪存中的改动信息 在 启动/复位时都会丢失。

可以通过改变跳线E50的默认状态来禁止SAVE操作。当E50没有处于它的默认 状态时,如果发送了SAVE命令,PMAC会报告一个错误。启动/复位时从非易失 内存恢复信息的操作也可以通过改变跳线E50的默认状态来禁止。

对SAVE命令、PMAC并不保证获取握手字符、直到它已经完成了保存操作之后 几分之一秒(带后备电池的PMAC)或约5到10秒(带后备闪存的PMAC)。主 机程序应该准备好为这个字符而等待比其它命令要长的时间。由于这个原因, 通常最好不要在"成块"下载一个大文件的操作过程中使用SAVE。 在SAVE的执行过程中,PMAC不执行其它的后台任务,包括用户的PLC和自动 安全检查(例如跟随误差和超程限定)。特别地,带后备闪存的板卡需要很多 秒进行保存,所以你必须确保当SAVE发送时系统没有以上这些安全任务。

范例 1130=60000 : 设置电机1的比例增益

; 保存到非易失内存中 SAVE

1130=80000 ; 设置新值

\$\$\$; 卡复位

I130 ; 查询I130的值

60000 : PMAC报告保存过的值

在线命令\$\$\$,\$\$\$*** 参考 跳线E50和E51。

SIZE

功能: 报告PMAC中未使用的缓冲的数量。

范围: 全局

语法: SIZE

本命令使PMAC向主机报告可作缓冲的未用内存的长字的数目。如果没有程序缓冲 备注:

(例如运动、PLC或旋转缓冲)被打开,报告的值是一个整数;而如果有一个缓冲

正打开,报告值则为一个负数。

范例: **DEFINE GATHER** : 为采集保留所有的剩余内存

> SIZE : 查询可打开的内存的数目

; PMAC报告无可用内存 0

; 从采集缓冲中释放内存 **DELETE GATHER** SIZE : 查询可打开的内存的数目 41301 ; PMAC报告可用字的数目 : 打开一个运动程序缓冲 **OPEN PROG 10** SIZE : 查询可打开的内存的数目

-41302 : 负号说明有一个缓冲被打开

|变量|18 参考:

在线命令DELETE GATHER, DELETE TRACE

TYPE

功能: 报告PMAC的类型。

范围: 全局

语法: TYPE

备注: 本命令使PMAC报告一个包含卡配置的字符串。它将用以下格式报告配置字符 串: {PMAC类型}, {总线类型}, {后备类型}, {伺服类型}, {梯形图类型}, {时钟倍频数}

位置

{PMAC类型}:

PMAC1 第一代PMAC(包括PMAC1.5)

PMAC2 第二代PMAC

PMACUL Ultra-lite (仅PMAC2有MACRO)

{总线类型}:

ISA IBM-PC ISA总线

VME VME总线 STD STD总线

ISA/VME PMAC1固件能够支持两种总线

{后备类型}:

BATTERY后备电池RAMFLASHAMD型后备闪存RAMI-FLASHIntel型后备闪存RAM

{伺服类型}:

PID标准PID伺服算法ESA选件6扩展伺服算法

{梯形图类型}:

{blank}不支持梯形图LDs支持梯形图

{时钟倍频数}

CLK Xn 晶体频率是CPU频率的n倍

范例 TYPE

PMAC1, ISA/VME, BATTERY, PID, CLK X1

TYPE

PMAC2, ISA, FLASH, ESA, CLK X3

TYPE

PMACUL, VME, FLASH, PID, LDs, CLK X2

参考 在线命令VERSION, DATE

UNDEFINE

功能: 擦除坐标系定义。

范围: 指定坐标系

语法: UNDEFINE UNDEF

备注: 本命令使PMAC擦除选址坐标系中所有的轴定义语句。它不会影响其他坐标系中的 轴定义语句。在定义新轴前使用本命令十分有用。

要想只擦除单个电机的轴定义语句,可使用#**{常量}->0**命令;要擦除每个坐标系中的所有轴定义语句,可使用**UNDEFINE ALL**命令。

范例 &1 ; 选坐标系1

#1-> ; 查询电机1的轴定义

10000X ; PMAC响应

#2-> ; 查询电机2的轴定义

10000X ; PMAC响应 UNDEFINE ; 擦除轴定义

&2 ; 选坐标系2

#1->10000X ; 在坐标系2重新定义电机1为X轴 #2->10000Y ; 在坐标系2重新定义电机2为Y轴

参考 轴,坐标系(设置一个坐标系)

在线命令#{常量}->, #{常量}->0, UNDEFINE ALL

UNDEFINE ALL

功能: 擦除所有坐标系中的坐标定义。

范围: 全局

语法: UNDEFINE ALL UNDEF ALL

备注: 本命令擦除所有坐标系中的全部轴定义语句。在重新载入PMAC坐标系定义前使用

本命令十分有用。

范例 &1#1-> ; 查询坐标系1中电机1的轴定义

1000X : PMAC响应

&2#5-> ; 查询坐标系2中电机5的轴定义

1000X ; PMAC响应

UNDEFINE ALL ;擦除所有轴定义

&1#1-> ; 查询坐标系1中电机1的轴定义

0 ; PMAC报告没有定义

&2#5-> ; 查询坐标系2中电机5的轴定义

0 ; PMAC报告没有定义

参考 轴, 坐标系(设置一个坐标系)

在线命令#{常量}->0, #{常量}->, UNDEFINE

V

功能: 报告电机速度。

范围: 指定电机

语法: V

备注: 本命令使PMAC向主机报告当前的真正电机速度,按数字/伺服循环测算,精确到

小数点后一位。他报告的是电机*真正速度寄存器*的内容(被[lx09*32]处)。

要想将报告的值转换为数字/毫秒,需将其乘以8,388.608*(lx60+1),再除以I10。 它也可以用其它的比例常数进一步转换为工程单位。

注意: 这里报告的速度值是通过减去连续伺服循环的位置得到的。因此他们可能 有很多干扰。如果要用于显示,最好采用保存在Y:\$082A、Y:\$08EA等寄存器中 的平均值。这些寄存器可以用M变量进行访问。

范例 V ; 查询所选电机的真实速度

> 21.9 ; PMAC报告21.9个数/循环

> > ;(*8,388.608/3,713.707=49.5个数/毫秒)

#6V ; 查询电机6的速度

-4.2 ; PMAC响应

#5V#2V ;查询电机5和2的速度 : PMAC先响应电机5的速度 0 7.6 ; PMAC后响应电机2的速度

|变量||10, ||x09, ||x60 参考

在线命令<CTRL-V>, F, P

内存映想寄存器X:\$0033, X:\$006F等, X:\$082A, X:\$08EA等

推荐的M变量定义Mx66

VERSION

功能: 报告PROM固件的版本号。

范围: 全局

语法: **VERSION**

VER

备注: 本命令使PMAC报告它使用的固件的版本号。当一个闪存PMAC处于"引导模式"

(启动时E51为ON)时,PMAC将报告引导固件的版本号而不是工作固件的版本 号。否则它将会报告工作固件的版本号。要从引导模式变为普通工作模式,可以用

<CTRL-R>命令。

范例 **VERSION** ; 查询PMAC的固件版本

> 1.12D : PMAC响应

参考 重置PMAC(与PMAC对话)

在线命令DATE, TYPE

W{地址}

功能: 向指定地址写值。

范围: 全局

语法: W{地址}, {常量} [,{常量}...]

位置

 $\{地址\}$ 包括一个字母X,Y或L;一个可选的冒号(;);以及一个从0到65535的 整数值(十六进制的\$0000到\$FFFF)。它指定要读取得PMAC内存或I/O起始地 址。

{常量}是一个十进制或十六进制整数,是要写到指定地址的值。

后一个{常量}是要向后续的高地址写入的值。

备注: 命令使PMAC将指定的**{常量}**值写入指定的内存字地址,或将指定一系列{常量}值写入后续的更高的地址(起始地址由{地址}指定,这实质上相当于POKE命令)。命令可以指定PMAC X内存中的短(24位)字、Y内存中的短(24位)字、或包括X和Y内存的长(48位)字(X字更重要)。这是由命令中X、Y或L地址前缀分别指定的。

范例 WY:\$C002,4194304 ; 这将使DAC2为5V

; (确保I200=0以免伺服被覆盖)

WY\$720,\$00C000,\$00C004,\$00C008,\$00C00C

; 这把前四个输入项写到

: 编码转换表中

参考 线命令R[H]{地址} 内存和I/O映像

 \mathbf{Z}

功能: 使指定轴位置为零。

范围: 指定坐标系

语法: Z

备注: 本命令使PMAC将坐标系中所有轴的当前轴位置重新标注为零。它仅仅是重新命名 当前位置,而不会产生任何运动。

本命令是对坐标系中所有轴执行**{轴}=0**的一个简便方法。**PSET X0 YP**等是具有相同功能的运动程序命令。

本命令并不会把电机位置寄存器置为零,它改变电机位置偏移寄存器从而影响电机零位置和轴零位置间的新偏移量。然而,报告的电机位置将会影响新偏移量,并报告零位置(+/-跟随误差)

范例 <CTRL-P> ; 查询报告的电机位置

2001 5002 3000 0 0 0 0 0 : PMAC报告位置

Z ; 坐标轴位置为零

<CTRL-P>; 查询报告的电机位置

12-100000 ; PMAC响应

参考 在线命令{轴}={常量}

内存映射寄存器**D:\$0813, D:\$08D3**等

推荐的M变量定义Mx64