

第一章	PMAC 简介 .....	4
第一节	关于本手册 .....	5
第二节	PMAC 简单介绍 .....	6
第三节	PMAC 型号 .....	7
1.3.1	PMAC-PCI.....	9
1.3.2	PMAC-PCI-Lite .....	10
1.3.3	Mini PMAC PCI .....	11
1.3.4	PMAC2-PCI.....	12
1.3.5	PMAC2-PCI-Lite .....	13
1.3.6	PMAC2A-PC104 .....	14
第四节	PMAC (1) 连接端子描述.....	16
第五节	PMAC (2) 连接端子描述.....	18
第六节	PMAC 工作设定 .....	20
1.6.1	硬件设定.....	20
1.6.2	软件设定.....	20
1.6.3	PMAC 设定 .....	21
第七节	PMAC 工作应答 .....	22
1.7.1	单信号 I/O .....	22
1.7.2	换相更新.....	23
1.7.3	伺服环更新.....	23
1.7.4	-VME Mailbox 处理 .....	24
1.7.5	实时中断应答 .....	24
1.7.6	后台应答.....	25
1.7.7	观察与检测.....	25
第二章	软件工具.....	27
第一节	配置 PEWIN32 PRO 组件 .....	28
2.1.1	保存与重置 PMAC 参数.....	30
2.1.2	使用 POSITION、TERMINAL 等窗口 .....	31
2.1.3	察看电机、坐标系、全局状态 .....	33
2.1.4	上载、下载程序.....	33
2.1.5	使用备份功能.....	34
2.1.6	指导式 I、M、P、Q 变量用法 .....	35
第二节	快速使用 PMAC Plot PRO .....	36
第三节	快速使用 PMAC Tuning PRO 整定 PID .....	38
第三章	安装与配置 PMAC .....	42
第一节	跳线设定.....	43
第二节	串口连接.....	44
第三节	与上位机连接.....	44
3.3.1	安装驱动与上位机识别 .....	45
3.3.2	复位 PMAC .....	53
第四节	连接端子 8D、8P、8S、8E .....	54
第五节	电源指定.....	59
3.5.1	数字量支持.....	59
3.5.2	模拟量支持.....	59

3.5.3	标志开关支持 .....	59
第六节	行程限位、回零开关 .....	60
3.6.1	关闭行程限位功能 .....	60
3.6.2	行程限位开关形式 .....	60
3.6.3	回零开关 .....	61
3.6.4	检查标志输入 .....	62
第七节	电机控制信号连接 .....	63
3.7.1	编码器信号连接 .....	63
3.7.2	检查编码器输入 .....	63
3.7.3	检查 DAC 输出(1 型卡) .....	63
3.7.4	检查 PFM 输出 (2 型卡) .....	64
3.7.5	电机使能信号(AENAx/DIRx) .....	64
3.7.6	电机报警 (Faultx) .....	65
3.7.7	通用 I/O .....	65
3.7.8	常用设定 .....	70
第四章	PMAC 指令与应用 .....	75
第一节	在线指令 .....	76
第二节	缓冲区 (编程) 指令 .....	77
第三节	特色 .....	78
4.3.1	I 变量 .....	78
4.3.2	P 变量 .....	78
4.3.3	Q 变量 .....	78
4.3.4	M 变量 .....	79
4.3.5	队列处理 .....	79
4.3.6	运算方法 .....	80
4.3.7	功能简介 .....	80
4.3.8	比较功能 .....	80
4.3.9	用户自定义伺服算法 .....	80
第四节	内存地址表 .....	81
第五节	程序缓冲区 .....	81
第六节	编码器转换表 .....	81
第七节	PMAC 位置寄存器 .....	81
第八节	回零运动 .....	81
第九节	Command、Send 等增强指令 .....	82
第五章	电机编程 .....	83
第一节	PMAC 运动程序 .....	84
第二节	笛卡儿坐标系 .....	85
5.2.1	轴的定义 .....	85
5.2.2	轴定义的扩展与描述 .....	85
第三节	编写运动程序 .....	87
第四节	执行运动程序 .....	89
第五节	子程序或者辅助程序 .....	90
5.5.1	子程序/子例程变量交换 .....	90
5.5.2	G、M、T、D 代码 (标准机床代码) .....	90

第六节	混合运动/加速度模式: .....	92
第七节	圆弧查补 .....	94
第八节	Splined 运动 .....	95
第九节	PVT-Mode 运动 .....	96
第十节	其他扩展特色运动 .....	97
5.10.1	旋转缓冲区 .....	97
5.10.2	内部时基控制 .....	97
5.10.3	外部时基控制 (电子凸轮) .....	97
5.10.4	位置跟随 (电子齿轮) .....	98
5.10.5	刀具半径补偿 .....	98
5.10.6	同步 M 变量调用 .....	99
5.10.7	多块 PMAC 同步 .....	100
5.10.8	轴转置矩阵 .....	100
5.10.9	位置捕捉或者位置比较 .....	100
5.10.10	会学习的运动程序 .....	100
第六章	PLC 编程 .....	101
第一节	关于 PLC 程序 .....	102
第二节	PLC 程序结构 .....	102
第三节	计算功能 .....	102
第四节	有条件的计算 .....	103
第五节	While 循环 .....	103
第六节	Command、Send 等增强指令 .....	103
第七节	计时器 .....	104
第八节	编译 PLC 程序 .....	104
第七章	注意的问题 .....	105
附页 1:	PMAC 错误代码列表 .....	106
附页 2:	PMAC I 变量功能列表 .....	107
附页 3:	在线指令列表 .....	108
附页 4:	PMAC 编程指令列表 .....	109
附页 5:	电机常用 M 变量定义 .....	110
附页 7:	常见电机接线连接 .....	111
附页 8:	PMAC 附件与选项 .....	112

# 第一章      **PMAC 简介**

## 第一节 关于本手册

本手册是为第一次使用 PMAC 运动控制器的客户准备的，PMAC 卡具有易学易用、编程简单、程序集成化程度高等特点，是许多开发人员的首选开发工具。

在整个 PMAC 的系列产品中包括：PMAC(1)系列、PMAC2 系列、Turbo PMAC(1)系列、Turbo PMAC2 系列、MACRO 系列和 UMAC 系列等。

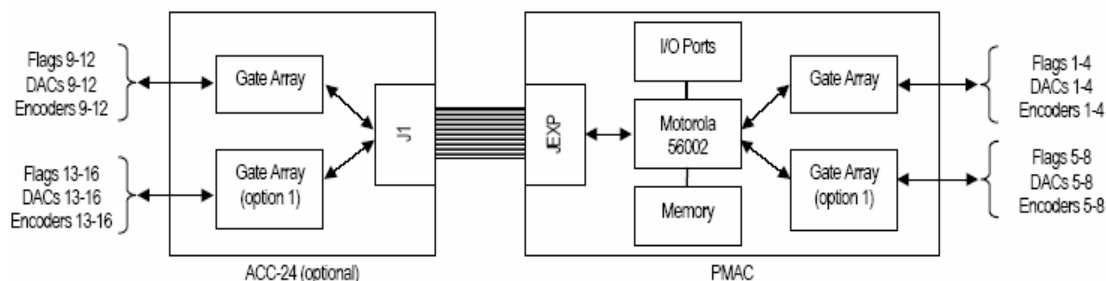
为了方便开发人员的编程工作，PMAC 卡提供了手动、编程、插补等大量，极大地提高了工程人员的开发效率。然而，在使用 PMAC 的过程中，我们发现，如果能对 PMAC 所提供的体系有一个全面的了解，将会大大缩短开发过程。

本书了

motion control. It is oriented to the PMAC(1) family of motion controls and it does not cover other PMAC families: PMAC2, Turbo PMAC(1), Turbo PMAC2, MACRO or UMAC. The subjects illustrated could be used as a quick informative features description or as a roadmap for a more advanced learning through the main documentation. It is strongly recommended to use this quick reference manual in conjunction with the following manuals

## 第二节 PMAC 简单介绍

PMAC 的全称是 Programmable Multi-Axis Controller。它是 Deltu Tau Data System 公司出产的一种系列广泛，支持众多的可编程多轴运动控制卡。



PMAC 作为一个高性能伺服运动控制器，通过数字信号处理器（DSP），以及灵活的高级语言最多可控制八轴同时运动。

- PMAC 给多轴控制器提供前所未有的性价比，采用 Motorola 的 DSP56 系列数字芯片作为 CPU，处理所有八轴的计算。
- PMAC 主要有两大类七种类型的板卡：
  - ✧ PMAC I 型卡：PMAC1-PCI, PMAC1-Lite-PCI, PMAC1-MINI-PCI, PMAC – VME;
  - ✧ PMAC II 型卡：PMAC2-PCI, PMAC2-Lite-PCI, PMAC2A-PC104;
  - ✧ 除以上还有 TURBO PMAC 系列, UMAC 系列, QMAC 系列, MACRO Station 系列，不在本书进行导论。
- PMAC 使用专用的 DSPGATE 门阵列 IC，每个 DSPGATE 可连接 4 路电机轴运动控制通道。每个 PMAC 支持 4 个 DSPGATE，即 PMAC 系统最多可拥有 16 个通道用于控制。
- PMAC DSPGATE 门阵列 IC 主要有两种接口形式：
  - ✧ 为 PMAC I 型卡提供的 $\pm 10V$  模拟量接口形式；
  - ✧ 为 PMAC I 型卡提供的数字量接口形式：包括 PWM 脉宽调制接口，脉冲方向组合接口，也可以使用双路 $\pm 10V$  模拟量接口。
- 每个通道轴接口电路包括：
  - ✧ 3 信号（A/B/C(Z)）积分编码器输入；
  - ✧ 4 个标示信号（用于行程限位和零点信号）输入，
  - ✧ 2 个标示信号输出；
- 任意一块 PMAC 卡都具有控制 8 个运动轴的能力。第一个 DSPGATE 提供 1 到 4 号通道（PMAC1-MINI-PCI 只能提供 1-2 通道），第二个 DSPGATE 可以通过 Option1 这个选项（PMAC1-Lite-PCI, PMAC1-MINI-PCI, PMAC2-Lite-PCI 没有这个选项）提供 5 到 8 号通道，附件 ACC-24 可以提供 9 到 12 号通道，附件 ACC-24 Option1 可以提供 13 到 16 号通道。
- PMAC 板卡拥有自己的内存和处理器，因此它完全可以独立进行各种控制，也可以使用提供平行处理的计算机，通过串口、总线、以太网来联合运行。

## 第三节 PMAC 型号

### 独立特性

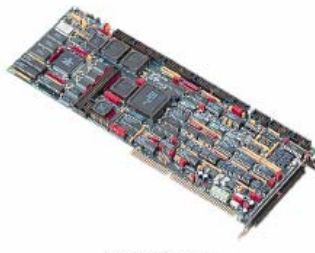
- Motorola DSP 56 系列数字处理芯片;
- 4 路输出 digital-to-analog (DAC) 、PWM、PFM (脉冲+方向);
- 4 路编码器通道;
- 16 点通用 I/O, 可与 OPTO-22 兼容 (PMAC I 型卡);
- 8 点通用 I/O, 可与 OPTO-22 兼容, 32 点自定义 I/O (PMAC 2 型卡);
- 可以扩展的千点以上的 I/O;
- 内置行程极限位, 零点, 电机报警, 电机使能;
- PCI 总线, ISA 总线或者 RS-232 通讯方式;
- 可独立工作;
- 支持标准 CNC G-代码;
- 线性和圆弧插补;
- 1024 个自定义的运动编程;
- 32 个自定义的 PLC 程序;
- 为巨大程序使用的旋转缓冲区;
- 36-位位置范围 (+/- 640 亿的计数)
- “S-曲线” 间速度和减速度
- Cubic Trajectory Calculations 计算, Splines 计算, PVT 计算
- 电子传动
- 高级 PID 伺服运动算法

### 可选择特性

- 最多 16 路输出 digital-to-analog (DAC) 、PWM、PFM (脉冲+方向);
- 最多 16 编码器通道;
- 8Kx16 双端口 RAM;
- 带电保存内存;
- 40, 60, 80 MHz 或更高的 CPU;
- 扩展的 (Pole-Placement) 多极点伺服算法;
- 更高的精确时钟 (<10 ppm);
- Voltage-to-frequency (V/F) 伏频转换电路;
- 12 位, 16 位可选模拟量到数字量转化输入;
- Sinusoidal 编码器反馈输入;
- Yaskawa 绝对编码器输入;
- 模拟量反馈输入;
- MLDTs 反馈输入;
- 并行二进制反馈或者格雷码反馈;
- RS-232 或 RS-422 串口通讯转化;
- 多达 2048 个串行 I/O ;
- 最远 100 米的远距离 I/O 选项;



PMAC PC



PMAC Lite



Mini PMAC



PMAC VME



PMAC STD



PMAC Pack



Turbo PMAC PC



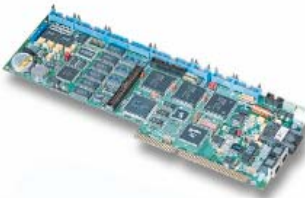
Turbo PMAC VME



Turbo PMAC2 3U



Turbo PMAC2 PC



Turbo PMAC2 PC Ultralite



UMAC Turbo System



### 1.3.1 PMAC-PCI

<b>PMAC-PCI</b>	<p>基本版本没有选项是一块 1 又 1/2 尺寸的板卡,包括:</p> <ul style="list-style-type: none"> <li>• 40 MHz DSP563xx CPU (相当于 120 MHz PMAC)</li> <li>• 128k x 24 位 SRAM 用于编译/存储程序</li> <li>• 快闪内存用于用户的备份和固件</li> <li>• 采用最新发布的固件版本</li> <li>• RS-232/RS-422 串口通讯,33 MHz 5V PCI 总线接口</li> <li>• 4 通道轴接口电路, 每一个包括: <ul style="list-style-type: none"> <li>• 16 位 +/- 10V 模拟量输出</li> <li>• 3 信号 (A/B/C(Z)) 微分编码器输入</li> <li>• 4 个标示信号输入, 2 个标示信号输出</li> <li>• 16 位 DAC 输出</li> </ul> </li> <li>• 显示, 控制面板, I/O 点接口</li> <li>• 缓冲区扩展点</li> <li>• 时钟频率在 +/- 100 ppm</li> <li>• 采用 PID/陷波/前馈 伺服算法</li> <li>• 出货日期后 1 年的保修期</li> </ul>
	<b>附加轴</b>
<b>OPT-1</b>	附加 4 路轴通道
	<b>通讯选项</b>
<b>OPT-2</b>	在板 8K x 16 位双端口 RAM,可通过 PCI 或者 USB 使用
	<b>处理器, 扩展内存及固件选项</b>
<b>OPT-5AF</b>	<p>默认的 CPU/内存配置: 40MHz DSP563xx CPU (相当于 80 MHz DSP56002) . 需要 V1.17 版本支持</p> <p>在使用卡的时候必须包含一个处理器选项</p>
<b>OPT-5CF</b>	<p>80MHz DSP563xx CPU (相当于 160 MHz DSP56002) . 需要 V1.17 版本支持</p>
<b>OPT-5EF</b>	<p>160MHz DSP563xx CPU (相当于 320 MHz DSP56002) 需要 V1.17 版本支持</p>
<b>OPT-6</b>	扩展的 "pole-placement" (多极点) 伺服算法固件
<b>OPT-6L</b>	Multi-block lookahead 功能固件
<b>OPT-10</b>	指定固件版本. 如果不指定, 将使用最新的固件版本.
	<b>基板选项</b>
<b>OPT-8A</b>	高精度时钟(+/-15ppm) 保证长时间的精度( +/-100ppm)
<b>OPT-12</b>	8 通道在板 12 位 A/D 转换
<b>OPT-12A</b>	在添加 8 通道在板 12 位 A/D 转换(必须选择 Option12)
<b>OPT-15</b>	变频转换用控制端子的模拟量输入
<b>OPT-16</b>	16Kx24 SRAM 电池保存参数内存

## 1.3.2 PMAC-PCI-Lite

<b>PMAC-PCI Lite</b>	<p>基本版本没有选项是一块 1 尺寸的板卡,包括:</p> <ul style="list-style-type: none"> <li>• 40 MHz DSP563xx CPU (相当于 120 MHz PMAC)</li> <li>• 128k x 24 位 SRAM 用于编译/存储程序</li> <li>• 快闪内存用于用户的备份和固件</li> <li>• 采用最新发布的固件版本</li> <li>• RS-232/RS-422 串口通讯,33 MHz PCI 总线接口</li> <li>• 4 通道轴接口电路, 每一个包括: <ul style="list-style-type: none"> <li>• 16 位 +/- 10V 模拟量输出</li> <li>• 3 信号 (A/B/C(Z)) 积分编码器输入</li> <li>• 4 个标示信号输入, 2 个标示信号输出</li> </ul> </li> <li>• 显示, 控制面板, I/O 点接口</li> <li>• 缓冲区扩展点</li> <li>• 时钟频率在 +/- 100 ppm</li> <li>• 采用 PID/陷波/前馈 伺服算法</li> <li>• 出货日期后 1 年的保修期</li> </ul>
	<b>通讯选项</b>
<b>OPT-2</b>	在板 8K x 16 位双端口 RAM,可通过 PCI 或者 USB 使用
	<b>处理器, 扩展内存及固件选项</b>
<b>OPT-5AF</b>	<p>默认的 CPU/内存配置: 40MHz DSP563xx CPU (相当于 80 MHz DSP56002) . 需要 V1.17 版本支持</p> <p>在使用卡的时候必须包含一个处理器选项</p>
<b>OPT-5CF</b>	<p>80MHz DSP563xx CPU (相当于 160 MHz DSP56002) . 需要 V1.17 版本支持</p>
<b>OPT-5EF</b>	<p>160MHz DSP563xx CPU (相当于 320 MHz DSP56002) 需要 V1.17 版本支持</p>
<b>OPT-6</b>	扩展的 "pole-placement" (多极点) 伺服算法固件
<b>OPT-6L</b>	Multi-block lookahead 功能固件
<b>OPT-10</b>	指定固件版本. 如果不指定, 将使用最新的固件版本.
	<b>基板选项</b>
<b>OPT-12</b>	8 通道在板 12 位 A/D 转换
<b>OPT-12A</b>	在添加 8 通道在板 12 位 A/D 转换(必须选择 Option12)
<b>OPT-15</b>	变频转换用控制端子的模拟量输入
<b>OPT-16</b>	16Kx24 SRAM 电池保存参数内存

### 1.3.3 Mini PMAC PCI

<b>Mini PMAC PCI</b>	<p>基本版本没有选项是一块 2/3 尺寸的板卡,包括:</p> <ul style="list-style-type: none"> <li>• 40 MHz DSP56303 CPU (相当于 120 MHz PMAC)</li> <li>• 512k x 24 位 SRAM 用于编译/存储程序</li> <li>• 快闪内存用于用户的备份和固件(5C0)</li> <li>• 采用最新发布的固件版本</li> <li>• RS-232/RS-422 串口通讯,33 MHz 5V PCI 总线接口</li> <li>• 2 通道轴接口电路, 每一个包括: <ul style="list-style-type: none"> <li>• 16 位 +/- 10V 模拟量输出</li> <li>• 3 信号 (A/B/C(Z)) 积分编码器输入</li> <li>• 4 个标示信号输入, 2 个标示信号输出</li> </ul> </li> <li>• 2 路附加的 3 信号 (A/B/C(Z)) 积分编码器输入</li> <li>• 显示, 控制面板, I/O 点接口</li> <li>• 缓冲区扩展点</li> <li>• 时钟频率在 +/- 100 ppm</li> <li>• 采用 PID/陷波/前馈 伺服算法</li> <li>• 出货日期后 1 年的保修期</li> </ul>
	<b>通讯选项</b>
<b>OPT-2</b>	在板 8K x 16 位双端口 RAM,可通过 PCI 或者 USB 使用
	<b>处理器, 扩展内存及固件选项</b>
<b>OPT-5AF</b>	<p>默认的 CPU/内存配置: <b>40MHz</b> DSP563xx CPU (相当于 80 MHz DSP56002) . 需要 V1.17 版本支持</p> <p>在使用卡的时候必须包含一个处理器选项</p>
<b>OPT-5CF</b>	<p><b>80MHz</b> DSP563xx CPU (相当于 160 MHz DSP56002) . 需要 V1.17 版本支持</p>
<b>OPT-5EF</b>	<p><b>160MHz</b> DSP563xx CPU (相当于 320 MHz DSP56002) 需要 V1.17 版本支持</p>
<b>OPT-6</b>	扩展的 "pole-placement" (多极点) 伺服算法固件
<b>OPT-6L</b>	Multi-block lookahead 功能固件
<b>OPT-10</b>	指定固件版本. 如果不指定, 将使用最新的固件版本.
	<b>基板选项</b>
<b>OPT-8A</b>	高精度时钟(+/-25ppm) 保证长时间的精度( +/-100ppm)
<b>OPT-15</b>	变频转换用控制端子的模拟量输入或者脉冲+方向的输出

### 1.3.4 PMAC2-PCI

<b>PMAC2-PCI</b>	<p>基本版本没有选项是一块 1 1/2 尺寸的板卡,包括:</p> <ul style="list-style-type: none"> <li>• 40 MHz DSP563xx CPU (相当于 120 MHz PMAC)</li> <li>• 128k x 24 位 零等待 SRAM</li> <li>• 采用最新发布的固件版本</li> <li>• RS-232 串口通讯,33 MHz 5V PCI 总线接口</li> <li>• 4 通道轴接口电路, 每一个包括: <ul style="list-style-type: none"> <li>• 3 种输出信号的设定</li> <li>• 双电位串行外部的模拟量(需要附件) 输出或者脉冲+方向信号</li> <li>• 三相 PWM 斩波输出</li> <li>• 3 信号 (A/B/C(Z)) 积分编码器输入</li> <li>• 9 个标示信号输入, 2 个标示信号输出</li> </ul> </li> <li>• 2 路辅助通道, 每一个包括: <ul style="list-style-type: none"> <li>• 2 信号 (A/B) 积分编码器输入</li> <li>• 1 个输出, 可设定为脉冲+方向信号或者 PWM 斩波输出</li> </ul> </li> <li>• I/O 点接口</li> <li>• 时钟频率 +/-100 ppm</li> <li>• 采用 PID/陷波/前馈 伺服算法</li> <li>• 出货日期后 1 年的保修期</li> </ul>
	<b>附加轴</b>
<b>OPT-1</b>	附加 4 路轴通道
	<b>通讯选项</b>
<b>OPT-2</b>	在板 8K x 16 位双端口 RAM
	<b>处理器, 扩展内存及固件选项</b>
<b>OPT-5AF</b>	默认的 CPU/内存配置: 40MHz DSP563xx CPU (相当于 80 MHz DSP56002) 需要 V1.17 支持 在使用卡的时候必须包含一个处理器选项
<b>OPT-5CF</b>	80MHz DSP56303 CPU (相当于 160 MHz DSP56002) 需要 V1.17 支持
<b>OPT-5EF</b>	160MHz DSP56303 CPU (相当于 320 MHz DSP56002) 需要 V1.17 支持
<b>OPT-6</b>	扩展的 "pole-placement" (多极点) 伺服算法
<b>OPT-6L</b>	lookahead 功能固件
<b>OPT-10</b>	指定固件版本. 如果不指定, 将使用最新的固件版本.
	<b>基板选项</b>
<b>OPT-8A</b>	高精度时钟(+/-15ppm) 保证长时间的精度( +/-100ppm)
<b>OPT-12</b>	8 通道在板 12 位 A/D 转换
<b>OPT-12A</b>	在添加 8 通道在板 12 位 A/D 转换(必须选择 Option12)
<b>OPT-15</b>	V-to-F 转换使用的模拟量输入
<b>OPT-16</b>	16Kx24 SRAM 电池保存参数内存

### 1.3.5 PMAC2-PCI-Lite

<b>PMAC2-PCI Lite</b>	<p>基本版本没有选项是一块 1 尺寸的板卡,包括:</p> <ul style="list-style-type: none"> <li>• 40 MHz DSP563xx CPU (相当于 120 MHz PMAC)</li> <li>• 128k x 24 位 零等待 SRAM</li> <li>• 采用最新发布的固件版本</li> <li>• 512K x 8 位 快闪内存用于用户的备份和固件(5C0)</li> <li>• RS-232 串口通讯,33 MHz 5V PCI 总线接口</li> <li>• 4 通道轴接口电路, 每一个包括: <ul style="list-style-type: none"> <li>• 3 种输出信号的设定</li> <li>• 双电位串行外部的模拟量(需要附件) 输出或者脉冲+方向信号</li> <li>• 三相 PWM 斩波输出</li> <li>• 3 信号 (A/B/C(Z)) 积分编码器输入</li> <li>• 9 个标示信号输入, 2 个标示信号输出</li> </ul> </li> <li>• 2 路辅助通道, 每一个包括: <ul style="list-style-type: none"> <li>• 2 信号 (A/B) 积分编码器输入</li> <li>• 1 个输出, 可设定为脉冲+方向信号或者 PWM 斩波输出</li> </ul> </li> <li>• I/O 点接口</li> <li>• 时钟频率 +/-100 ppm</li> <li>• 采用 PID/陷波/前馈 伺服算法</li> <li>• 出货日期后 1 年的保修期</li> </ul>
	<b>通讯选项</b>
<b>OPT-2</b>	在板 8K x 16 位双端口 RAM
	<b>处理器, 扩展内存及固件选项</b>
<b>OPT-5AF</b>	<p>默认的 CPU/内存配置: <b>40MHz</b> DSP56303 CPU (相当于 80 MHz DSP56002) 需要 V1.17 支持</p> <p>在使用卡的时候必须包含一个处理器选项</p>
<b>OPT-5CF</b>	<p><b>80MHz</b> DSP56303 CPU (相当于 160 MHz DSP56002)</p> <p>需要 V1.17 支持</p>
<b>OPT-5EF</b>	<p><b>160MHz</b> DSP56303 CPU (相当于 320 MHz DSP56002)</p> <p>需要 V1.17 支持</p>
<b>OPT-6</b>	扩展的 "pole-placement" (多极点) 伺服算法
<b>OPT-6L</b>	lookahead 功能固件
<b>OPT-8A</b>	高精度时钟(+/-25ppm) 保证长时间的精度( +/-100ppm)
<b>OPT-10</b>	指定固件版本. 如果不指定, 将使用最新的固件版本.
	<b>基板选项</b>
<b>OPT-12</b>	8 通道在板 12 位 A/D 转换
<b>OPT-12A</b>	在添加 8 通道在板 12 位 A/D 转换(必须选择 Option12)
<b>OPT-16</b>	16Kx24 SRAM <b>电池保存参数内存</b>

### 1.3.6 PMAC2A-PC104

<b>PMAC2A-PC/10</b> <b>4 4 or 8 Axis</b> <b>Stand-alone or</b> <b>PC104 Bus</b> <b>Controller</b>	<p>基本版本没有选项是一块 90mm X 95mm 尺寸的板卡,包括:</p> <ul style="list-style-type: none"> <li>• 40 MHz DSP563xx CPU (相当于 80 MHz DSP 56xxx)</li> <li>• 128k x 24 位 零等待 SRAM</li> <li>• 512k x 24 位 快闪内存用于用户的备份和固件</li> <li>• 采用最新发布的固件版本</li> <li>• RS-232 串口通讯</li> <li>• 4 通道轴接口电路, 每一个包括: <ul style="list-style-type: none"> <li>• 12 位 +/- 10V 模拟量输出</li> <li>• 脉冲+方向数字量输出</li> <li>• 3 信号 (A/B/C(Z)) 积分编码器输入</li> <li>• 4 个标示信号输入, 2 个标示信号输出</li> <li>• 3 相 PWM 斩波输出, 非默认的</li> </ul> </li> <li>• 50 芯的放大器和编码器接口</li> <li>• 34 芯的标示位接口</li> <li>• 采用 PID/陷波/前馈 伺服算法</li> <li>• 出货日期后 1 年的保修期</li> </ul>
处理器, 扩展内存及固件选项	
OPT-5AF	40MHz DSP56303 CPU (相当于 80 MHz DSP56002) 需要 V1.17 支持
OPT-5CF	80MHz DSP56303 CPU (相当于 160 MHz DSP56002)
OPT-5EF	160MHz DSP56303 CPU (相当于 320 MHz DSP56002)
OPT-6	扩展的 "pole-placement" (多极点) 伺服算法固件
OPT-6L	Multi-block lookahead 功能固件
OPT-10	指定固件版本. 如果不指定, 将使用最新的固件版本.
OPT-8A	高精度时钟(+/-25ppm) 保证长时间的精度( +/-100ppm)
基板选项	
OPT-2A	PC/104 总线栈式连接端子,方便使用 PC/104 计算机
OPT-12	2 通道在板 12 位 A/D 转换(不兼容 ACC 8FS 输出)
8 轴扩展选项	
ACC-1P	<p>ACC1P 提供 4 通道轴接口电路,总数达到 8 个伺服轴,包括:</p> <ul style="list-style-type: none"> <li>• 12 位 +/- 10V 模拟量输出</li> <li>• 脉冲+方向数字量输出</li> <li>• 3 信号 (A/B/C(Z)) 积分编码器输入</li> <li>• 4 个标示信号输入, 2 个标示信号输出</li> <li>• 16 位 DAC 输出</li> <li>• 3 相 PWM 斩波输出, 非默认的</li> </ul>
ACC-1P Opt 1	<p><b>Option 1 在 ACC-1P 轴扩展板上提供下列数字端子:</b></p> <ul style="list-style-type: none"> <li>• <u>Multiplexer Port</u>: 用来连接 PMAC ACC-34x 形式板. (推荐使用 ACC-21FH ). (34pin-&gt;50pin IDC)</li> <li>• <u>I/O Port</u>: 这个端子提供 8 inputs / 8outputs 在 5 到 24 VDC 范围. 这个 34-芯连接可以很轻松连接到 OPTO-22 输出模块.</li> </ul>

	<ul style="list-style-type: none"> <li>• <u>Handwheel port</u>: 这个端子提供 2 路扩展通道每个通告都可以编码器输入或者脉冲加方向输出之间选择.</li> </ul>
<b>ACC-1P Opt 2</b>	2 通道在板 12 位 A/D 转换(不兼容 ACC 8FS 输出)
	<b>通讯扩展选项</b>
<b>ACC-2P</b>	高速通讯及 I/O 扩展板(必须选择 Opt 1A, 1B, 2,或者 3 之一)
	<b>通讯选项</b>
<b>ACC-2P Opt 1A</b>	480 Mbit/sec USB 2.0 接口
<b>ACC-2P Opt 1B</b>	10 Mbit/sec 以太网接口和 480 Mbit/sec USB 2.0 接口
	<b>双端口 RAM 选项</b>
<b>ACC- 2P Opt 2</b>	在板 8K x 16 位双端口 RAM,可通过 PCI 或者阿以太网、USB 使用
<b>ACC-2P Opt 3</b>	<p><b>Option 3 在 ACC-2P 扩展板上提供下列数字端子:</b></p> <ul style="list-style-type: none"> <li>• <u>Multiplexer Port</u>: 用来连接 PMAC ACC-34x 形式板.</li> <li>• <u>TTL I/O Port</u>: 这个端子提供 16 个 在 5 VDC 范围.</li> <li>• <u>Handwheel port</u>: 这个端子提供 2 路扩展通道每个通告都可以编码器输入或者脉冲加方向输出之间选择.</li> </ul>

## 第四节 PMAC（1）连接端子描述

### ● JPAN:

操作控制面板接口，可连接的附件有 ACC18；I2=1 时，该接口的输入及输出可作为通用 I/O 点使用。

PMAC-PCI	PMAC-PCI-Lite	Mini PMAC PCI
J2	J2	N/A

### ● JTHW:

多路 I/O 点的扩展接口，该接口本身可提供 8 点输入及 8 点输出,还可与下属附件连接：

- i. ACC34 系列 I/O 扩展板，可提供 32IN 32OUT 输入及输出；
- ii. ACC34AE I/O 扩展板，可提供 8IN 8OUT 输入及输出；
- iii. ACC27 专用 8in/8out 接口板；
- iv. ACC8D opt 7 旋转变压器反馈接口板；
- v. ACC8D opt 9 安川绝对编码器反馈接口板；
- vi. ACC33 NC 操作面板接口板。

PMAC-PCI	PMAC-PCI-Lite	Mini PMAC PCI
J3	J3	J3

### ● JRS422 or JRS232:

PMAC 卡的串行通讯接口。

PMAC-PCI	PMAC-PCI-Lite	Mini PMAC PCI
J4 (RS422/232) J17 (RS232)	J4 (RS422/232)	J4 (RS232)

### ● JOPTO:

8IN/8OUT 通用 I/O 接口，可连接 ACC34AE I/O 扩展板，可提供 8IN 8OUT 输入及输出。

PMAC-PCI	PMAC-PCI-Lite	Mini PMAC PCI
J5	J5	J5

### ● JMACH:

60 芯接口，包括信号：

- i. 模拟量输出；
- ii. 编码器反馈；
- iii. 正负限位及回零信号；
- iv. 伺服使能和报警；
- v. 卡的看门狗报警；
- vi. 也可以为卡供给电源。

PMAC-PCI	PMAC-PCI-Lite	Mini PMAC PCI
J7 J8	J8	J11

### ● JANA(选项):

模拟量输入口，此接口可接受 16 路 0-5V 的模拟量信号。

PMAC-PCI	PMAC-PCI-Lite	Mini PMAC PCI
J30	J30	N/A

### ● JDISP:

显示器连接口，连接 LCD 显示模块，可用 Display 命令显示用户定义的字符串。



PMAC-PCI	PMAC-PCI-Lite	Mini PMAC PCI
J1	J1	J1

● **JEQU 或 JAUX:**

位置比较相等的输出信号接口。

PMAC-PCI	PMAC-PCI-Lite	Mini PMAC PCI
J9	J9	J8

● **JS:**

模拟量输入扩展，可与 ACC28A/B 连接。

PMAC-PCI	PMAC-PCI-Lite	Mini PMAC PCI
JS1 JS2	JS1	J7

● **JXIO:**

用于扩展功能。

PMAC-PCI	PMAC-PCI-Lite	Mini PMAC PCI
J6	J6	N/A

● **JUSB (选项):**

USB 通讯端子。

PMAC-PCI	PMAC-PCI-Lite	Mini PMAC PCI
J31	J31	? ?

● **JEXP:**

PMAC 卡的并行接口，可连接的附件有：ACC14 并行 I/O 或反馈输入板，ACC39 三洋绝对编码器输入口，DPRAM 双端口 RAM 等。

● **TB1:**

脱机使用时，GND，5V，+/-15V 的模拟量输入端子。

● **LED 灯:**

D20 和 D20A: 绿色 LED 指示灯，亮的时候表示卡上有+5V 工作电压输入；

D21 和 D21A: 红色 LED 指示灯，亮的时候表示卡看门狗报警；

D15: LED 指示灯，亮的时候表示卡上有+/-15V 工作电压输入。

## 第五节 PMAC（2）连接端子描述

### ● JTHW:

多路 I/O 点的扩展接口，该接口本身可提供 8 点输入及 8 点输出,还可与下属附件连接：

- i. ACC34 系列 I/O 扩展板，可提供 32IN 32OUT 输入及输出；
- ii. ACC34AE I/O 扩展板，可提供 8IN 8OUT 输入及输出；
- iii. ACC27 专用 8in/8out 接口板；
- iv. ACC8D opt 7 旋转变压器反馈接口板；
- v. ACC8D opt 9 安川绝对编码器反馈接口板；
- vi. ACC33 NC 操作面板接口板。

PMAC2-PCI	PMAC2-PCI-Lite
J2	J2

### ● JRS422 or JRS232:

PMAC 卡的串行通讯接口。

PMAC2-PCI	PMAC2-PCI-Lite
J5 (RS232) J5A (RS422)	J5 (RS232) J5A (RS422)

### ● JI/O:

通用 I/O 接口，此接口可提供 32 点用户自行定义是输入还是输出的 I/O，以及他们的极性。

PMAC2-PCI	PMAC2-PCI-Lite
J3	J3

### ● JMACH::

100 芯接口，包括信号：

- i. 模拟量、PWM、PFM（脉冲+方向）可选输出；
- ii. 编码器反馈；
- iii. 正负限位及回零信号；
- iv. 伺服使能和报警。

PMAC2-PCI	PMAC2-PCI-Lite
J9 J10 J11 J12	J9 J10

### ● JANA(选项):

模拟量输入口，此接口可接受 16 路 0-5V 的模拟量信号。

PMAC2-PCI	PMAC2-PCI-Lite
J1	J1

### ● JDISP:

显示器连接口，连接 LCD 显示模块，可用 Display 命令显示用户定义的字符串。

PMAC2-PCI	PMAC2-PCI-Lite
J6	J6

### ● JEQU:

位置比较相等的输出信号接口。

PMAC2-PCI	PMAC2-PCI-Lite
J8	J8

### ● JHW:

手轮编码器接口，可接受两路手轮或编码器信号，或作为脉冲+方向信号输出，并且不占用电机的编码器通道。

PMAC-PCI	PMAC-PCI-Lite
J7	J7

- **JEXP:**  
PMAC 卡的并行接口，可连接的附件有：ACC14 并行 I/O 或反馈输入板， ACC39 三洋绝对编码器输入口， DPRAM 双端口 RAM 等。
- **TB1:**  
脱机使用时，GND，5V，+/-12V 的模拟量输入端子。
- **TB3:**  
脱机使用时，卡的看门狗报警输出。

## 第六节 PMAC 工作设定

当第一次使用PMAC卡的时候，不需要DELTA TAU为您生产独特的硬件、软件结构。PMAC 已经拥用适合大部分生产研究所需要的特性了。使用 PMAC 完成您的工作是一件简单而又轻松的事：

- 灵活的中断驱动方案允许每件工作，每个运动规划和 每个PLC，都可以独立地运行，又可以相互合作。
- 指针式变量——M变量，可以监听 PMAC 中任意的记忆寄存器：运动过程寄存器，伺服算法规则，PLC的运行或主机的访问。
- 灵活强大的访问机制，保证任意时间、任何变量或状态指令可以被使用或者访问，从而灵活的改变电机运动。
- 同时最高8个轴可以同步运行和同时最高8个轴可以插补，可以灵活的根据控制或运动进行组合。
- 可以进行数据采集，灵活方便的分析运动轨迹，速度描绘或任何需要比较分析的数据。

### 1.6.1 硬件设定

在PMAC上，你将会见到许多跳线（金属制作的叉状物），叫做了电子点。一些已经被短路；其它的是默认为开路的。这些跳线用来定制PMAC的一些硬件特性。强烈推荐使用每个跳线的默认配置。对于跳线我们在以后的章节详细讨论。

### 1.6.2 软件设定

为了使用PMAC达到用户需要的特性，会使用一些能决定PMAC卡片特性的参数，这些参数被称作变量。通过设置变量，就可以适当地配置电机。这些变量被保存在EAROM里面。首次上电时，EAROM里面保存的是DELTA TAU在生产PMAC时写入的默认变量值。使用PMAC配套的软件PEWIN32 Pro和其它的相关的软件包：诸如：P1Setup、PMACPlot等，可以改变变量的值，从而构造我们需要的系统。PEWIN32 Pro系列软件包有下列的主要工具和特征：

- 终端窗体（Teminal Window）：这是使用者和PMAC之间沟通主要频道。
- 即时系统数据窗体（Watch Window）：提供即时系统数据窗体用来除错。
- 位置窗体（Position Window）：显示位置、速度、跟随误差。
- PMAC Tuning Pro：调节PMAC系统PID。
- PMAC Plot Pro：为绘制和计算采集到的数据。

在 PEWIN32 Pro 中可以随时更改变量，或向PMAC发送指令。举例来说，在终端窗体里键入I900<CR>，就可以得到I900值。改变I900值的方法是使用等号和新值，(举例来说 I900=3< CR>)。

**注意：**如果你改变了变量的值，你必须用 **save** 这个命令将它保存，否则，当失去电源时，你的改动将丢失。

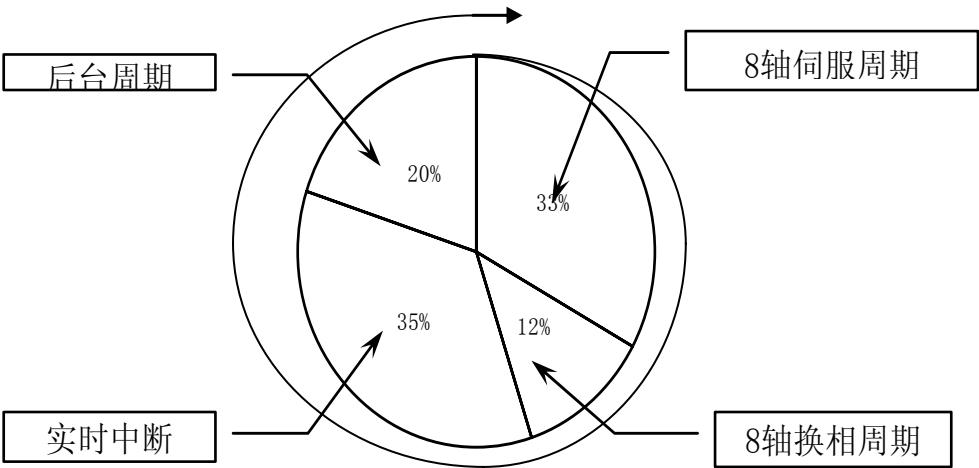
### 1.6.3 PMAC 设定

运动程序或者PLC程序可以在任何的文本处理软件中编辑，然后以PEWIN32 Pro进行下载。PEWIN32 Pro也提供一个内建的文本编辑软件，可以更加轻松的建立运动程序或者PLC程序。大多数的PMAC指令都能从与PMAC进行通讯的任何窗体发送。允许在线指令，比如说，手动控制电机，改变变量的值，报告变量的值，开始或者停止运动，根据状态数据的变化编写程序或PLC。

第七节 PMAC 工作应答

以40 MHz PMAC 再工作过程中的完成任务的应答能力为图示：

40MHz PMAC 440us的伺服周期



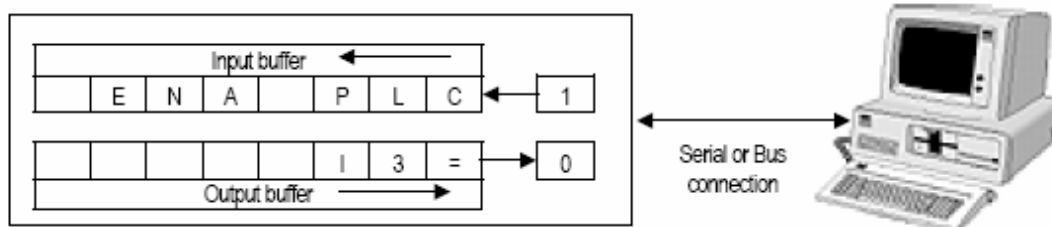
低优先级—————→高优先级																																													
后台周期	实时中断	伺服周期	换相周期																																										
其他应答结束	通过 I8 来设定	通过跳线和 Ix60 设定	通过跳线设定																																										
<table><tr><td>第 1 个使能的 PLC</td></tr><tr><td>第 1 个使能的 PLCC</td></tr><tr><td>.....</td></tr><tr><td>最后使能的 PLCC</td></tr><tr><td>在线命令请求</td></tr><tr><td>安全检查、看门狗计 时器设定</td></tr><tr><td>第 2 个使能的 PLC</td></tr><tr><td>第 2 个使能的 PLCC</td></tr><tr><td>.....</td></tr><tr><td>最后使能的 PLCC</td></tr><tr><td>在线命令请求</td></tr><tr><td>安全检查、看门狗计 时器设定</td></tr><tr><td>.....</td></tr><tr><td>最后使能的 PLC</td></tr></table>	第 1 个使能的 PLC	第 1 个使能的 PLCC	.....	最后使能的 PLCC	在线命令请求	安全检查、看门狗计 时器设定	第 2 个使能的 PLC	第 2 个使能的 PLCC	.....	最后使能的 PLCC	在线命令请求	安全检查、看门狗计 时器设定	.....	最后使能的 PLC	<table><tr><td>坐标系 1 的运动规划</td></tr><tr><td>坐标系 2 的运动规划</td></tr><tr><td>坐标系 3 的运动规划</td></tr><tr><td>坐标系 4 的运动规划</td></tr><tr><td>坐标系 5 的运动规划</td></tr><tr><td>坐标系 6 的运动规划</td></tr><tr><td>坐标系 7 的运动规划</td></tr><tr><td>坐标系 8 的运动规划</td></tr><tr><td>使能的 PLC0</td></tr><tr><td>使能的 PLCC0</td></tr><tr><td>看门狗计时器的消耗</td></tr></table>	坐标系 1 的运动规划	坐标系 2 的运动规划	坐标系 3 的运动规划	坐标系 4 的运动规划	坐标系 5 的运动规划	坐标系 6 的运动规划	坐标系 7 的运动规划	坐标系 8 的运动规划	使能的 PLC0	使能的 PLCC0	看门狗计时器的消耗	<table><tr><td>编码器转化表的计算</td></tr><tr><td>电机 1 伺服环更新</td></tr><tr><td>电机 2 伺服环更新</td></tr><tr><td>电机 3 伺服环更新</td></tr><tr><td>电机 4 伺服环更新</td></tr><tr><td>电机 5 伺服环更新</td></tr><tr><td>电机 6 伺服环更新</td></tr><tr><td>电机 7 伺服环更新</td></tr><tr><td>电机 8 伺服环更新</td></tr></table>	编码器转化表的计算	电机 1 伺服环更新	电机 2 伺服环更新	电机 3 伺服环更新	电机 4 伺服环更新	电机 5 伺服环更新	电机 6 伺服环更新	电机 7 伺服环更新	电机 8 伺服环更新	<table><tr><td>电机 1 换向更新</td></tr><tr><td>电机 2 换向更新</td></tr><tr><td>电机 3 换向更新</td></tr><tr><td>电机 4 换向更新</td></tr><tr><td>电机 5 换向更新</td></tr><tr><td>电机 6 换向更新</td></tr><tr><td>电机 7 换向更新</td></tr><tr><td>电机 8 换向更新</td></tr></table>	电机 1 换向更新	电机 2 换向更新	电机 3 换向更新	电机 4 换向更新	电机 5 换向更新	电机 6 换向更新	电机 7 换向更新	电机 8 换向更新
第 1 个使能的 PLC																																													
第 1 个使能的 PLCC																																													
.....																																													
最后使能的 PLCC																																													
在线命令请求																																													
安全检查、看门狗计 时器设定																																													
第 2 个使能的 PLC																																													
第 2 个使能的 PLCC																																													
.....																																													
最后使能的 PLCC																																													
在线命令请求																																													
安全检查、看门狗计 时器设定																																													
.....																																													
最后使能的 PLC																																													
坐标系 1 的运动规划																																													
坐标系 2 的运动规划																																													
坐标系 3 的运动规划																																													
坐标系 4 的运动规划																																													
坐标系 5 的运动规划																																													
坐标系 6 的运动规划																																													
坐标系 7 的运动规划																																													
坐标系 8 的运动规划																																													
使能的 PLC0																																													
使能的 PLCC0																																													
看门狗计时器的消耗																																													
编码器转化表的计算																																													
电机 1 伺服环更新																																													
电机 2 伺服环更新																																													
电机 3 伺服环更新																																													
电机 4 伺服环更新																																													
电机 5 伺服环更新																																													
电机 6 伺服环更新																																													
电机 7 伺服环更新																																													
电机 8 伺服环更新																																													
电机 1 换向更新																																													
电机 2 换向更新																																													
电机 3 换向更新																																													
电机 4 换向更新																																													
电机 5 换向更新																																													
电机 6 换向更新																																													
电机 7 换向更新																																													
电机 8 换向更新																																													

1.7.1 单信号 I/O

接收一个字符身，或发送一个字符到PMAC，这时串口或者总线在PMAC上拥用很高的

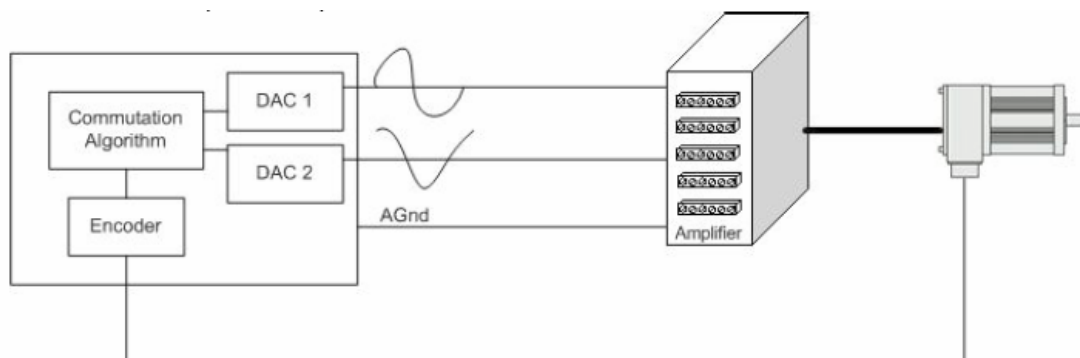
优先级。这个应答过程每个字符大约 200 nsec, 但是PMAC在处理时, 采用 character-by-character方式, 这样处理就会更快了。这一件工作是PMAC的总计算时间的重要部分。

注意这一件描述不包括全部命令, 只是指那些优先级较低的命令。(见到背景工作区段)



## 1.7.2 换相更新

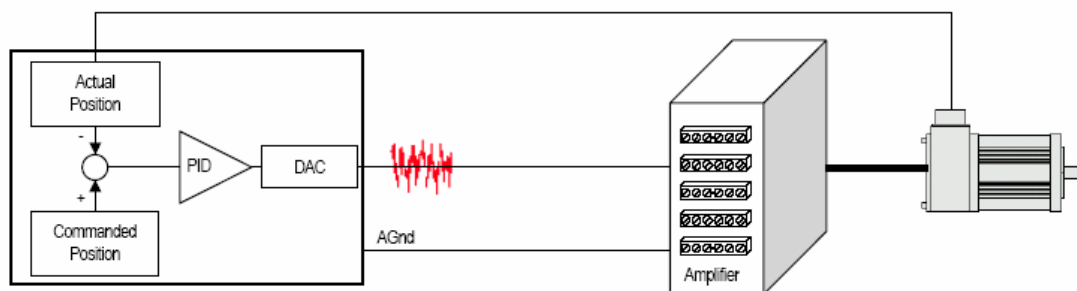
换相更新是在PMAC上的第二级优先处理。举例来说: 在20 MHz PMAC中, 当PMAC需要换相时(Ix01=1), 这个换相工作大约是每更新周期3  $\mu$  sec。这个工作的主时钟频率是根据跳线E98, E29-E33决定的。默认更新频率是9 kHz (110  $\mu$ sec cycle), 这时每个电机的换相计算大约占3%的PMAC计算处理能力。



**注意:** 这个应用是对于需要 PMAC 来处理换相的电机, 不适用于一般的伺服电机。

## 1.7.3 伺服环更新

伺服更新: 计算电机新的命令位置, 读取电机新的真实位置, 和计算在不同指令控制下的输出, 这是PMAC第三级优先处理。举例来说: 在20 MHz PMAC中, 当PMAC需要伺服环更新时(Ix00=1), 这个伺服数据更新的工作大约是每更新周期30  $\mu$  sec。这个工作的主时钟频率是根据跳线E98, E29-E33, E3-E6决定的。默认更新频率是2.26 kHz (442  $\mu$ sec cycle), 每个电机的伺服更新计算大约占7%的PMAC计算处理能力。



### 1.7.4 -VME Mailbox 处理

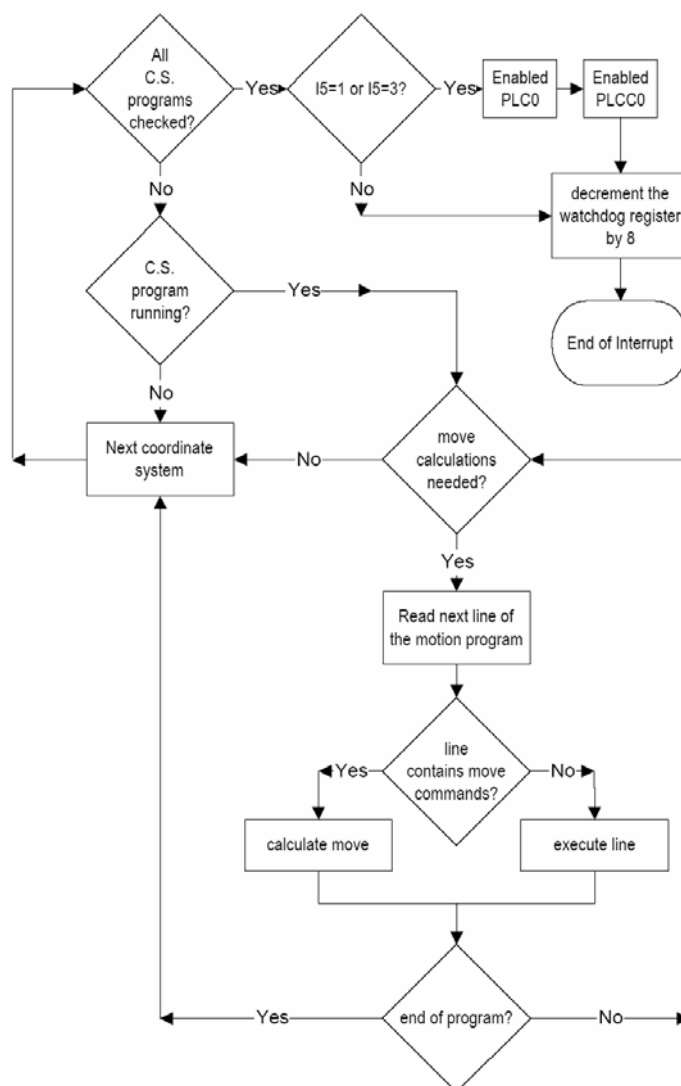
通过VME Mailbox 寄存器可以读取或改写最高十六个字的处理，这是PMAC第四级优先处理。主控制比率时刻发生，这是PMAC的重要计算。

### 1.7.5 实时中断应答

实时中断(RTI)是PMAC第五级优先处理。一旦发生，可以立即的在伺服环更新之后得到应答，应答关系可以由变量 I8 控制(每 I8+1个伺服更新周期更新)。这样的处理主要有二个：PLC 0/ PLCC0和运动轨迹规划。

PMAC会计算在不同坐标系下每个运动程序不同的运动指令。运动指令的计算根据不同的类型指令和模态指令而进行不同运算。当PMAC可靠收到非运动指令时，非运动指令可以在下列情况立即执行。当前的运动程序已经得到经过计算的运动轨迹规划，或者运动指令已经完成。

在运动轨迹规划的过程中，PMAC会有计划的向前寻找下一次移动指令,并尝试将运动轨迹混合，也可通过参数禁止PMAC混合运动。还有两种情况可以引起不同的跳转：**ENDWHILE**和**GOTO**。(RETURN将不计算)

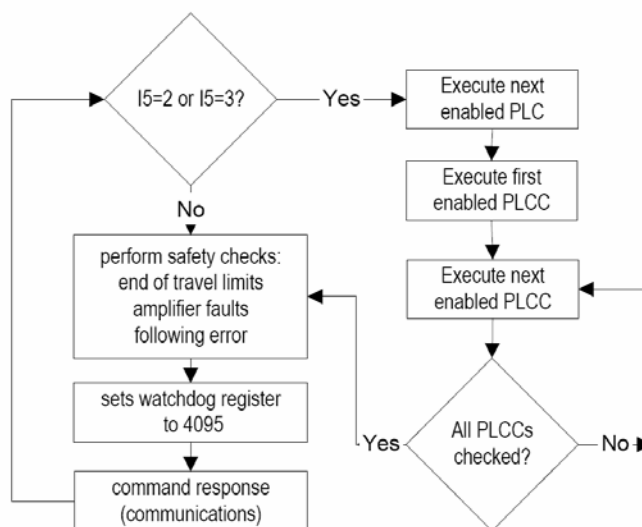




## 1.7.6 后台应答

在后台应答的工作期间，不会响应任何一个高优先级的任务，PMAC将会运行后台应答工作。后台应答主要有三件基本工作:指令处理,PLC1-31,和辅助处理程序。后台应答过程中:较多的高优先级工作被运行,那后台应答执行的越慢;后台应答任务越多,执行也越慢。

PLC执行过程扫描过程,不会被其他PLC中断(但它能被较高的优先级工作打断)。在PLC程序之间,PMAC 也会处理辅助处理程序和回应指令。



全部的PLCC能够在PLC运行之间全部扫描,不会被其他PLCC中断(但它能被较高的优先级工作打断)。但重新上电时,开始执行的PLCC 1-31是PLC1之后的。

任何指令的控制信号PMAC都会相应,并根据情况返回响应信号。通常情况下控制指令结束为回车符回返(<CR>),PMAC将回车符以前的所有字母与数字并用为一个控制指令。也有其他的控制指令具有自己的意义,这时PMAC也会及时响应。如果它是一个违法的指令,PMAC就会返回一个错误提示。

在对PLC程序的扫描之间,PMAC会运行辅助处理程序。最重要的辅助处理程序如:跟随误差报警,行程限位,电机出错,时钟监视等等。虽然这些处理优先级低,但PMAC确保了一个执行的最小的频率,如果低于最小频率,看门狗计时器就会触发,关闭PMAC。

## 1.7.7 观察与检测

PMAC有一个看门狗计时器(watchdog timer)电路,用来在危险故障发生时产生保护。在默认的设定下,如果实时中断应答频率降低至50Hz以下,或后台应答不能在512个实时中断应答循环周期里得到响应,这个计时器就会触发。这样做的目的是:保证PMAC卡的前台数据处理和后台数据处理都可以得到响应。如果这两个事件有一个不满足,看门狗计时器都会很快的产生动作。

PLC0或PLCC0必须在较高的频率下完成指定的工作(通常这一工作都比较少)。PLC0将会在指定数目的实时中断应答循环周期完成工作。PLC 0过于繁琐或者频繁就会干扰PMAC的正常工作安排。在最坏的情形下,看门狗计时器就会产生动作。

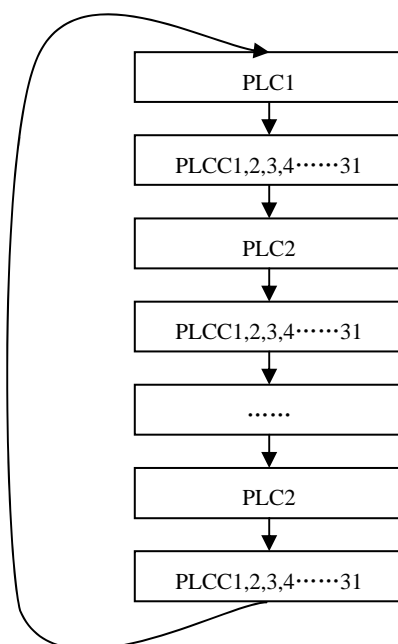
虽然由一个运动规划而引起一个看门狗计时器报警的情况是很罕见的,但是这确实有可能发生。如果有一个不运动的环,运动规划就会像PLC0一样产生。不正确的使用,即会导致产生大量重复的处理器时间,严重的看门狗计时器就会产生动作。

如果在运动之前有精密的轨迹规划计算,有可能出现,后台应答不能在512个实时中断应答循环周期里得到响应,看门狗计时器就会产生动作,这时就需要将大量的计算改为小指令

短运算。

使用编译过的PLCC程序可以得到更快的响应和执行速度。原因是：第一，除去了PLC程序指令的解释时间；第二，PLCC的计算能力比PLC强，浮点运算比PLC快2-3倍，整形计算快20-30倍。PMAC支持最多32个PLCC，可用最多15 K(15,360) 24位PMAC字节，或者是14K(14,336) 24位PMAC字节（用户自定义伺服算法使用的时候）。

在每个普通 PLC 扫描结束，都会穿插PLCC的扫描。如图所示：

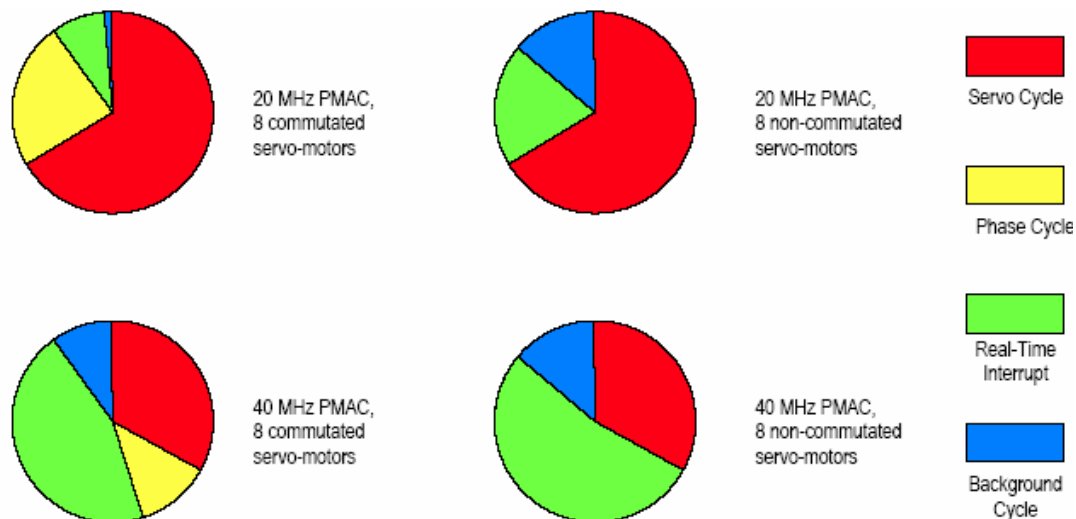


大部份辅助处理程序功能都用处理如：跟随误差报警，行程限位，电机出错，时钟监视等等。而额外的行程限位等功能需要处理，最好采用PLCC代替。

在重新上电或者重值（RESET）之后，PLC 将从1到31被运行。这样，可以使用PLC1在第一次运行时，进行初始化，禁止不用的PLC，设定密码等操作；在PLC1执行之后，PLC1还可译通过DISABLE指令禁止自己运行，这样作，有利于PMAC的上电操作。

**注意：** 本小节讨论的大部分是极限情况，建议初学者跳过本小节继续学习。

更好的合理分配资源，可以达到更好的效果。



## 第二章 软件工具

Pewin32 Pro是DELTA TAU公司开发的支持Microsoft Windows系列的软件工具包。它由以下组件组成：

- PEWIN32 Pro: 它是调试和设定PMAC的程序，是软件包的主程序；
- PMAC Plot Pro: 它可以在运动过程中访问各种内存寄存器，绘制和分析运动中电机的各种状态；
- PMAC Tuning Pro: 可以最优化电机的速度和加速度特性；
- P1 Setup32 Pro: 采用向导界面，帮助我们设置I型卡；
- P2 Setup32 Pro: 采用向导界面，帮助我们设置2型卡；
- Turbo Setup32 Pro: 采用向导界面，帮助我们设置Turbo PMAC卡、UMAC、QMAC、MACRO Station；
- UMAC Config Pro: 识别并配置UMAC的各种组件；



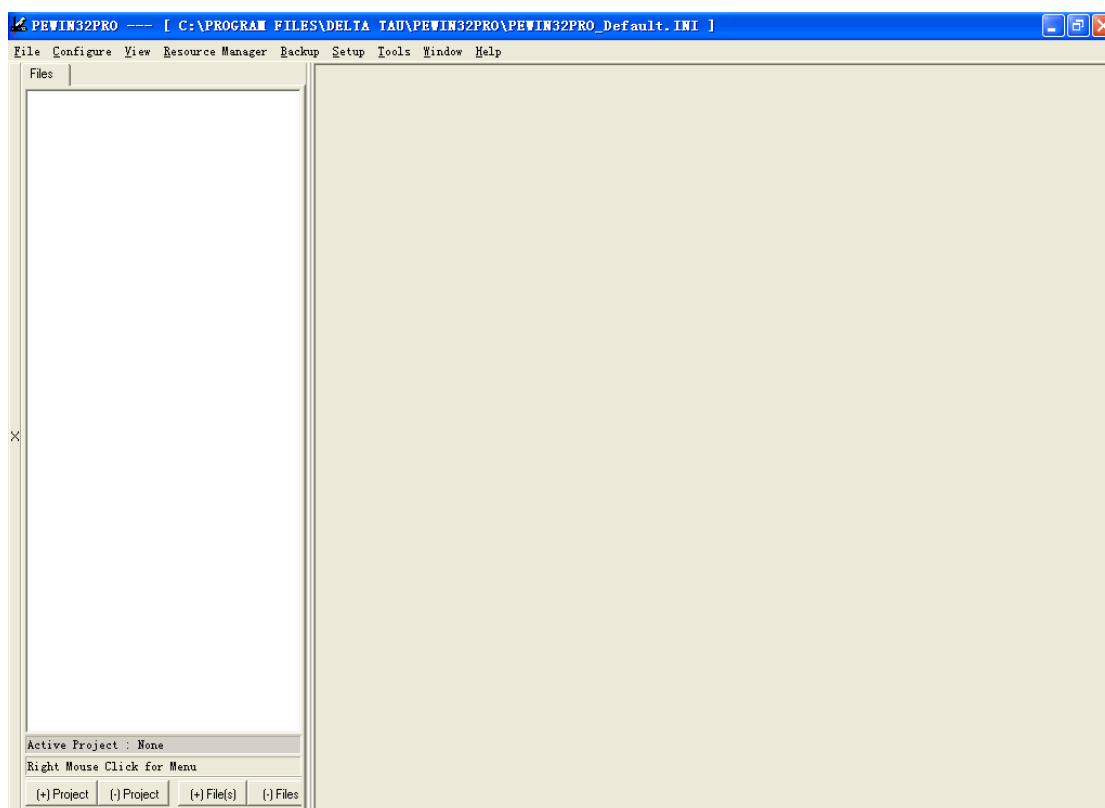
在本章中，我们只讨论 PEWIN32 Pro、PMAC Plot Pro、PMAC Tuning Pro三个主要软件

## 第一节 配置 PEWIN32 PRO 组件

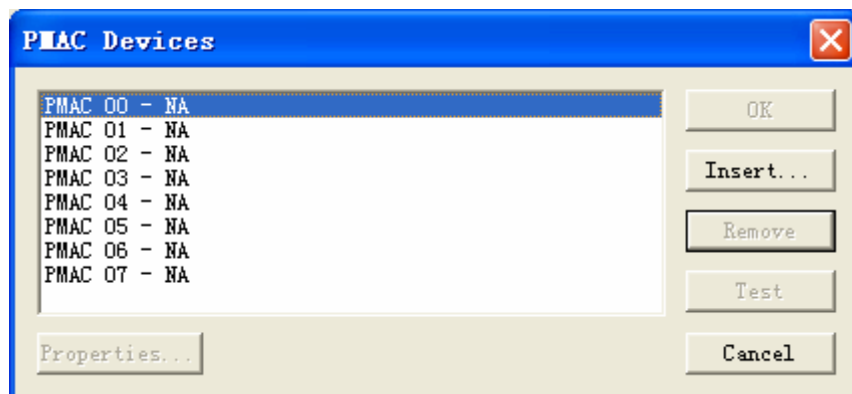
首先，我们必须保证 PEWIN32 PRO 软件正确安装和在计算机上已经被正确识别，如果你还没有做到，请参看第三章。当软件被正确安装值后，我们就可以得到如图：



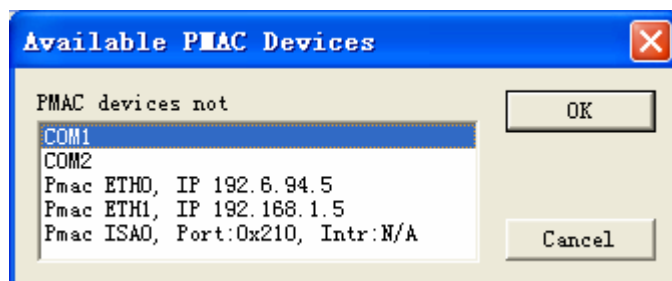
执行 PENWIN32PRO，如图所示：



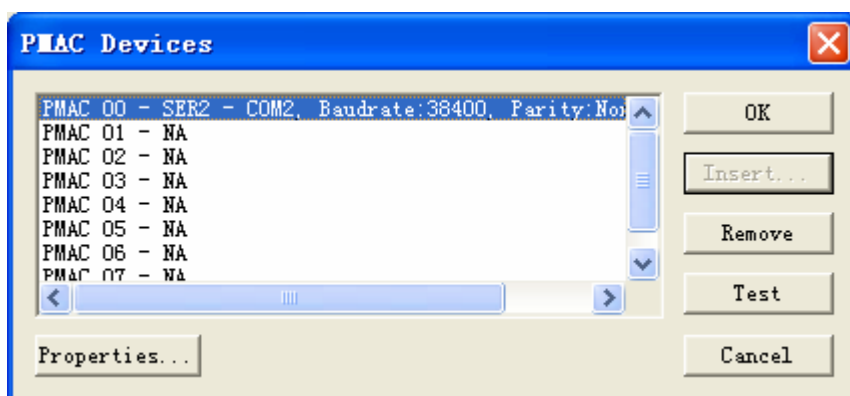
打开 setup 菜单，选择在 setup 菜单下选择 Force All Windows to Device Number



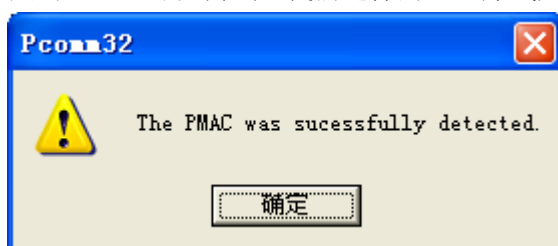
点击 Insert



点击，我们需要的通讯模式，点击 OK



点击 Test，可以测试，我们选择的通讯方式是否可用，一切正常时，就会显示



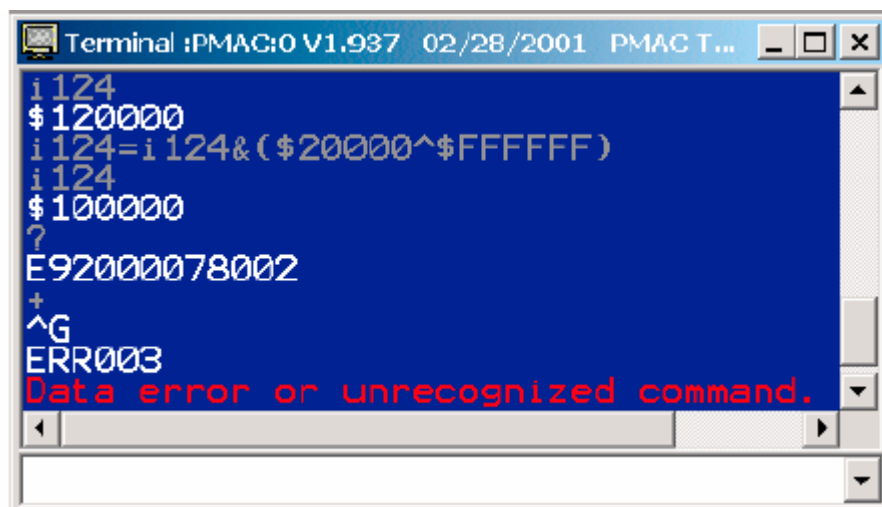
相反就是



如果出错，就要考虑是否存在错误，参看第三章。

## 2.1.1 保存与重置 PMAC 参数

当我们希望更改 PMAC 参数时，我们使用 View 菜单下 Terminal 选项，出现如下窗体：

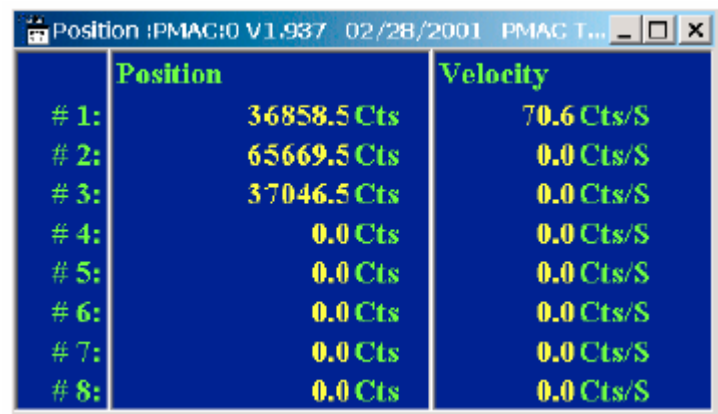


在 Terminal 窗体里面，可以键入指令，并观看返回值，（指令参考第四章）  
当我们需要保存改动时，使用的命令是 **Save**;

当我们需要将卡初始化，使用的命令是\$\$\$\*\*\*;  
当我们需要将卡重置或恢复 Save 之后的状态，使用的命令是\$\$\$。

### 2.1.2 使用 POSITION、TEMINAL 等窗口

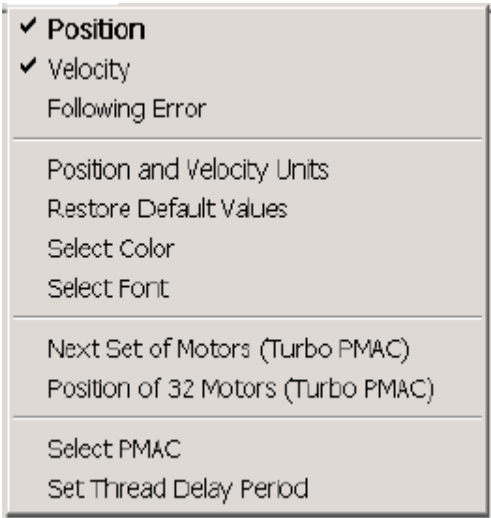
当我们希望查看电机运行的位置、速度等，我们使用 View 菜单下 Position 选项，出现如下窗体：



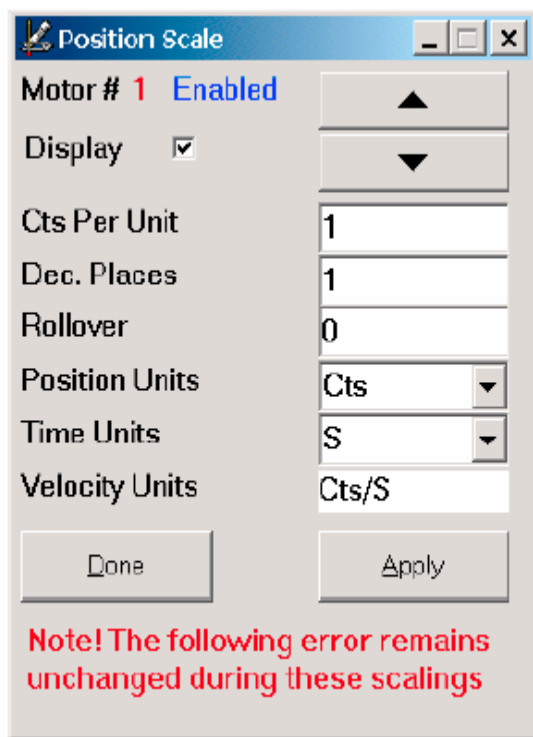
The screenshot shows a window titled "Position :PMAC:0 V1.937 02/28/2001 PMAC T...". It contains a table with two columns: "Position" and "Velocity". The table lists data for 8 motors, numbered #1 through #8. Motor #1 has a position of 36858.5 Cts and a velocity of 70.6 Cts/S. Motor #2 has a position of 65669.5 Cts and a velocity of 0.0 Cts/S. Motor #3 has a position of 37046.5 Cts and a velocity of 0.0 Cts/S. Motors #4 through #8 all have a position of 0.0 Cts and a velocity of 0.0 Cts/S.

	Position	Velocity
# 1:	36858.5 Cts	70.6 Cts/S
# 2:	65669.5 Cts	0.0 Cts/S
# 3:	37046.5 Cts	0.0 Cts/S
# 4:	0.0 Cts	0.0 Cts/S
# 5:	0.0 Cts	0.0 Cts/S
# 6:	0.0 Cts	0.0 Cts/S
# 7:	0.0 Cts	0.0 Cts/S
# 8:	0.0 Cts	0.0 Cts/S

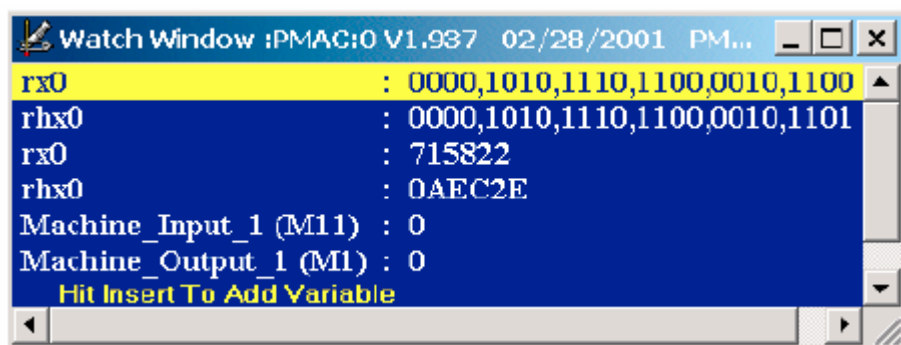
我们可以通过右键菜单，选择位置（Position）、速度（Velocity）和跟随误差（Following Error）的显示。



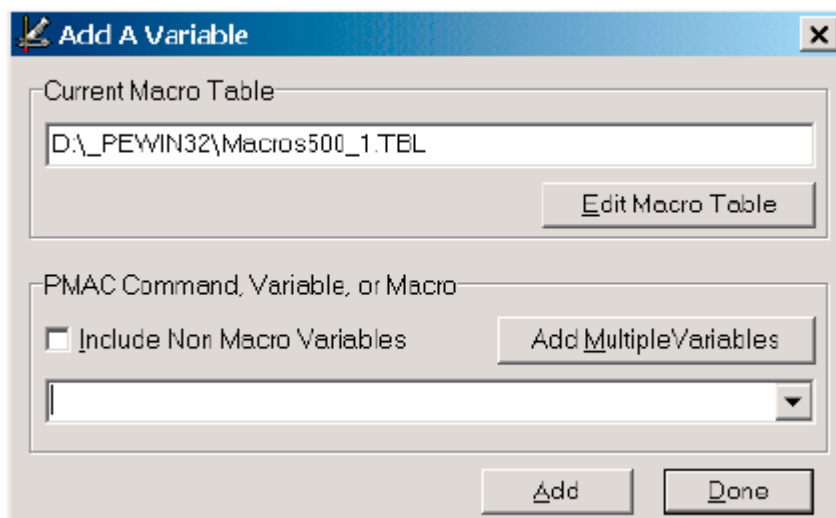
也可以通过右键菜单的，改变位置、速度的单位。



当我们希望查看内部寄存器、I/O 点、以及各种状态变化，我们使用 View 菜单下 Watch 选项，出现：



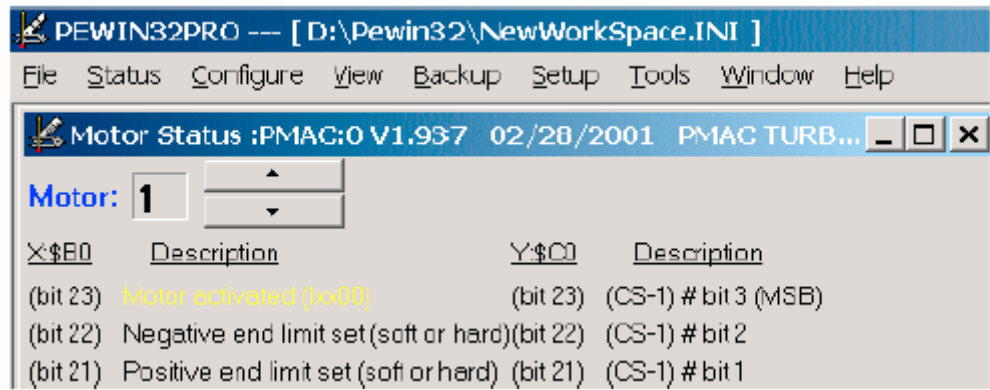
我们使用 INSERT 键，加入我们需要查看的变量，反之，用 Delete 键删除。



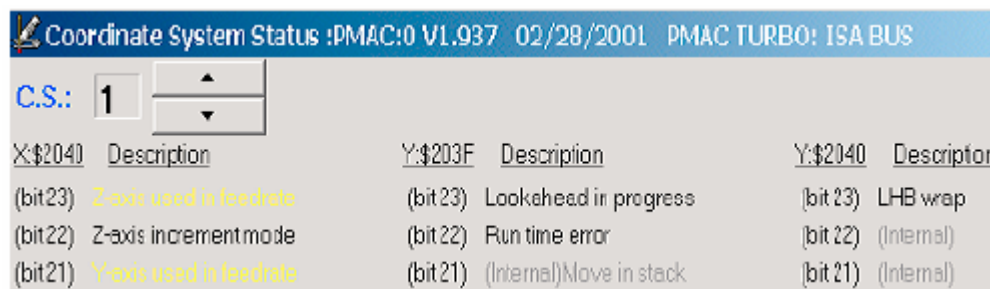


### 2.1.3 察看电机、坐标系、全局状态

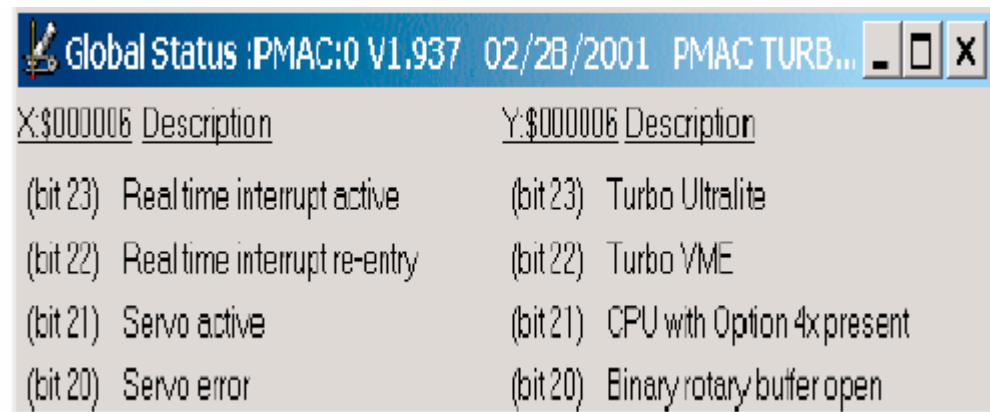
当我们希望查看电机的工作状态。我们使用 View 菜单下 Motor Status 选项，出现：



当我们希望查看坐标系的工作状态。我们使用 View 菜单下 Motor Status 选项，出现：

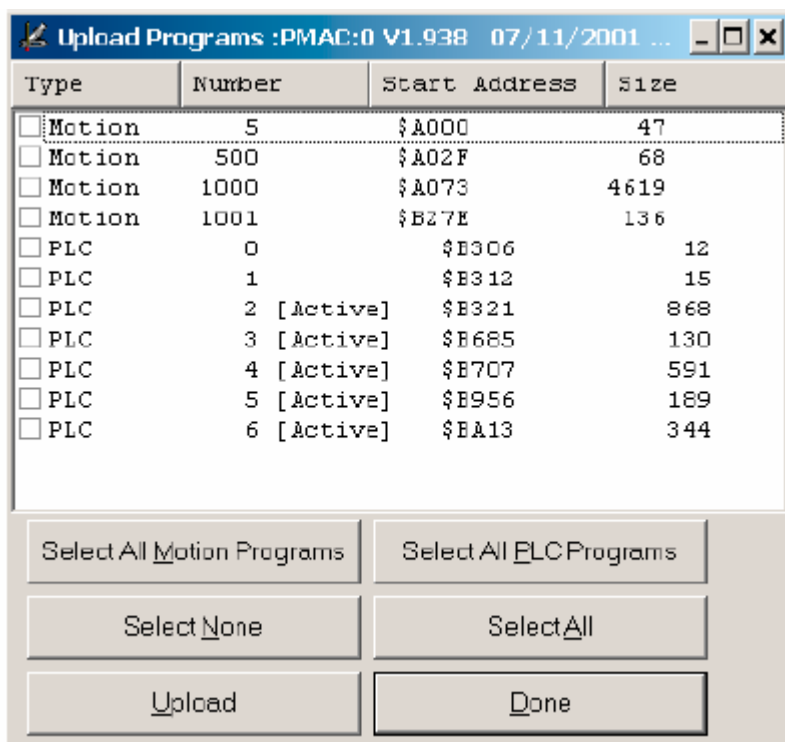


当我们希望查看电机的工作状态。我们使用 View 菜单下 Motor Status 选项，出现：

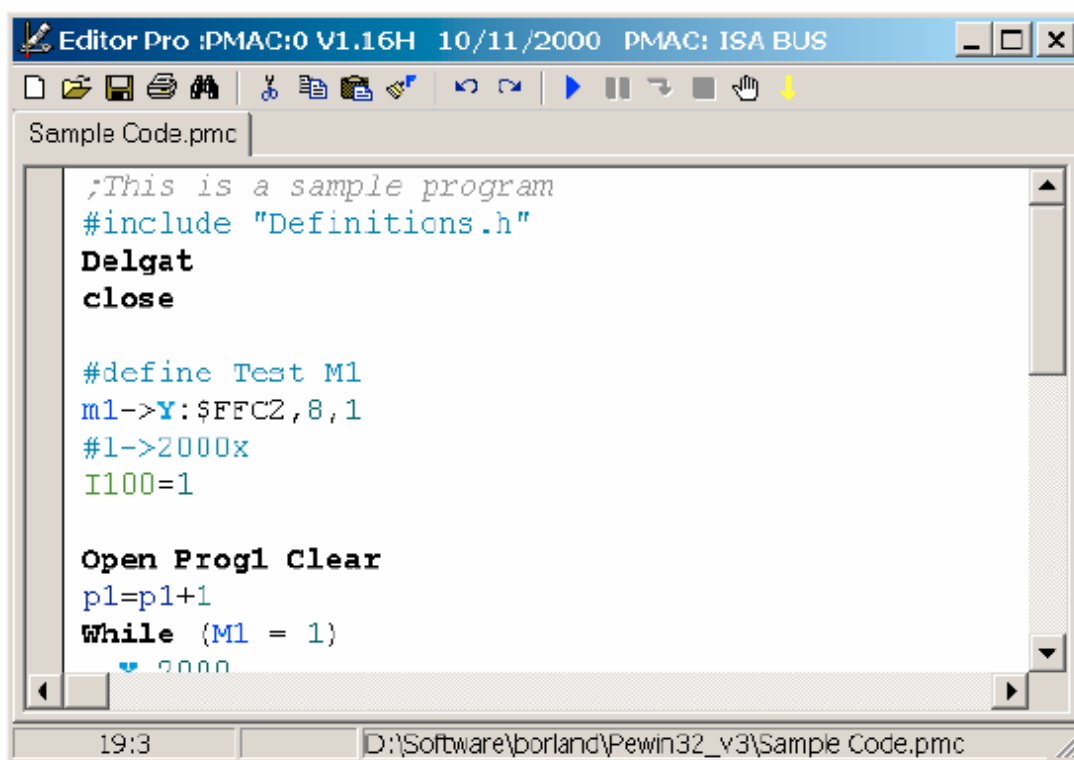


### 2.1.4 上载、下载程序

我们可以使用 File 菜单下的 Upload Programs 来上载卡中的程序。

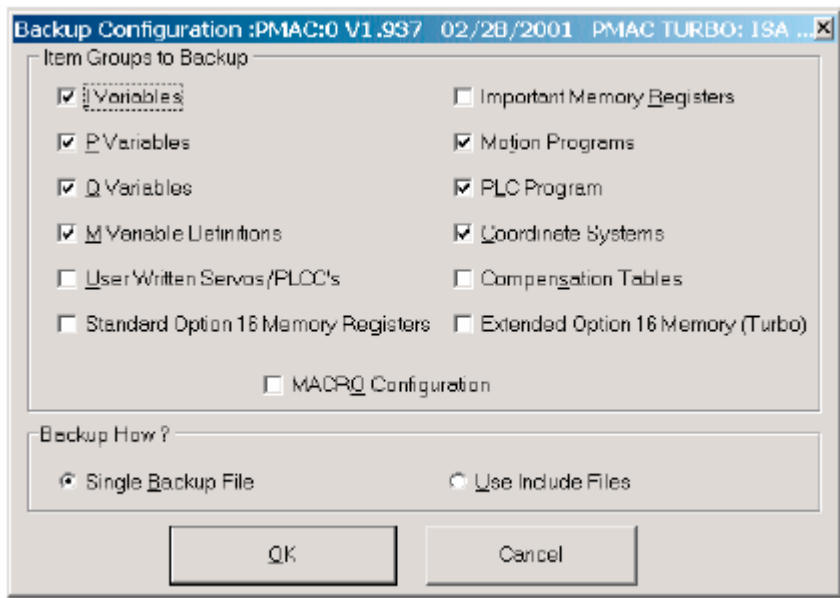


我们可以使用 File 菜单下的 New File 来编辑程序并下载，也可以使用 File 菜单下的 Download File 来下载已经编辑好的程序。



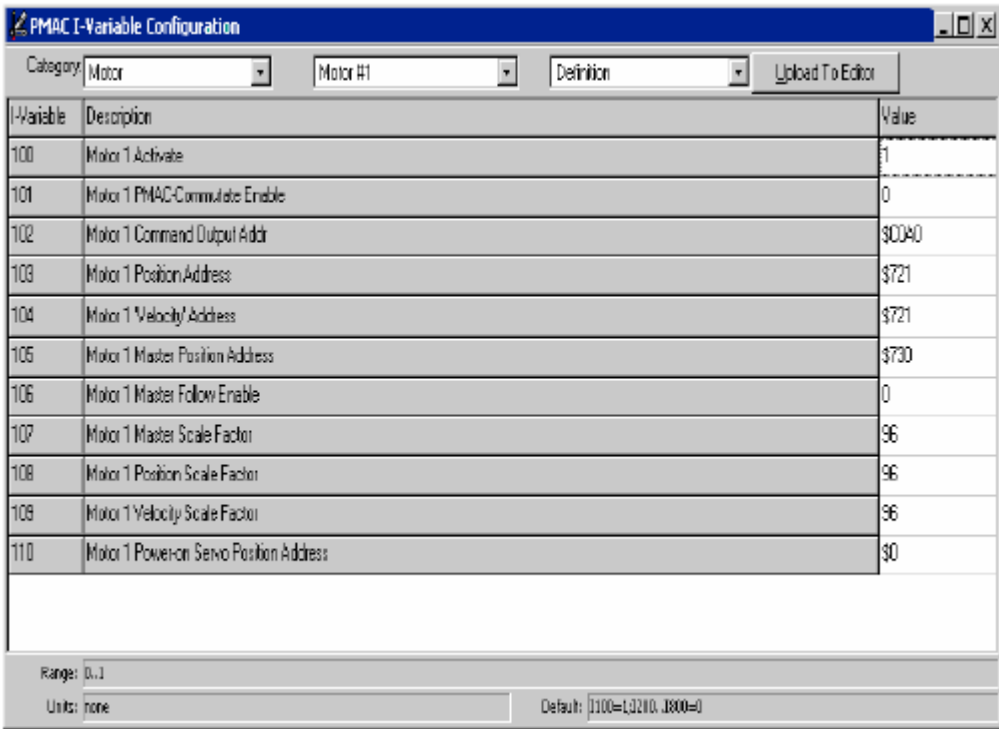
## 2.1.5 使用备份功能

我们可以将更改的参数备份为文件，也可以恢复它，使用 Backup 菜单的 Upload Configuration 和 Restore Configuration 选项



### 2.1.6 指导式 I、M、P、Q 变量用法

这是便捷的更改 I、M、P、Q 变量的用法。使用 Configure 菜单下的 I-Variables, M-Variables, P-Variables, Q-Variables。

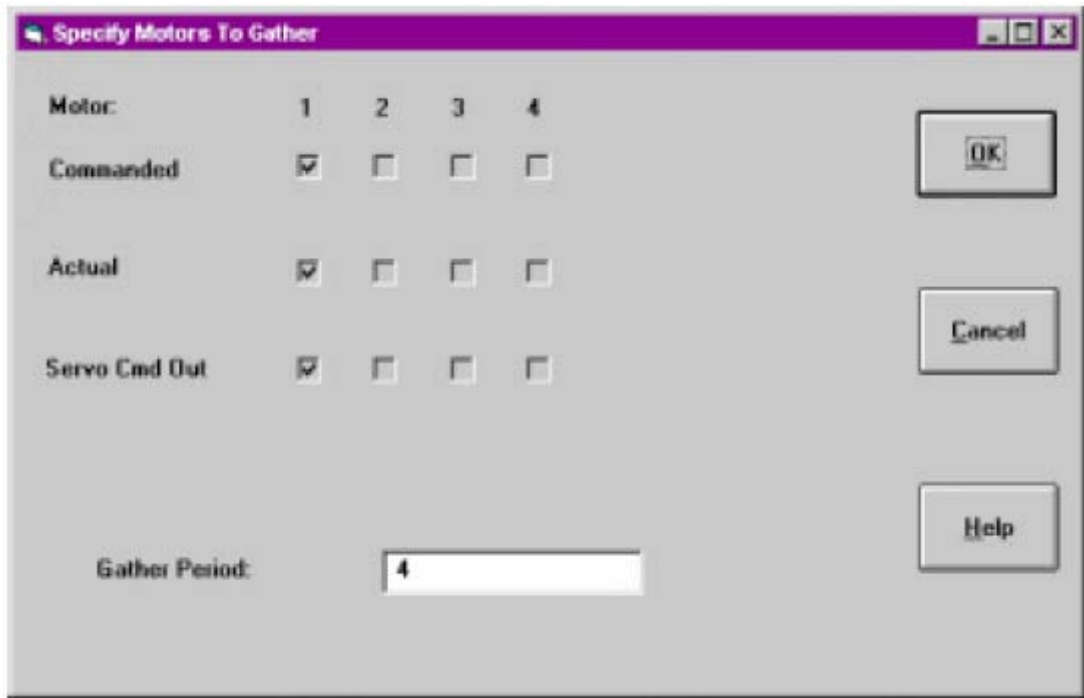


## 第二节 快速使用 PMAC Plot PRO

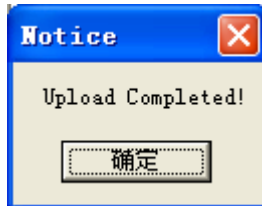
打开 PMAC Plot Pro 软件，如图：



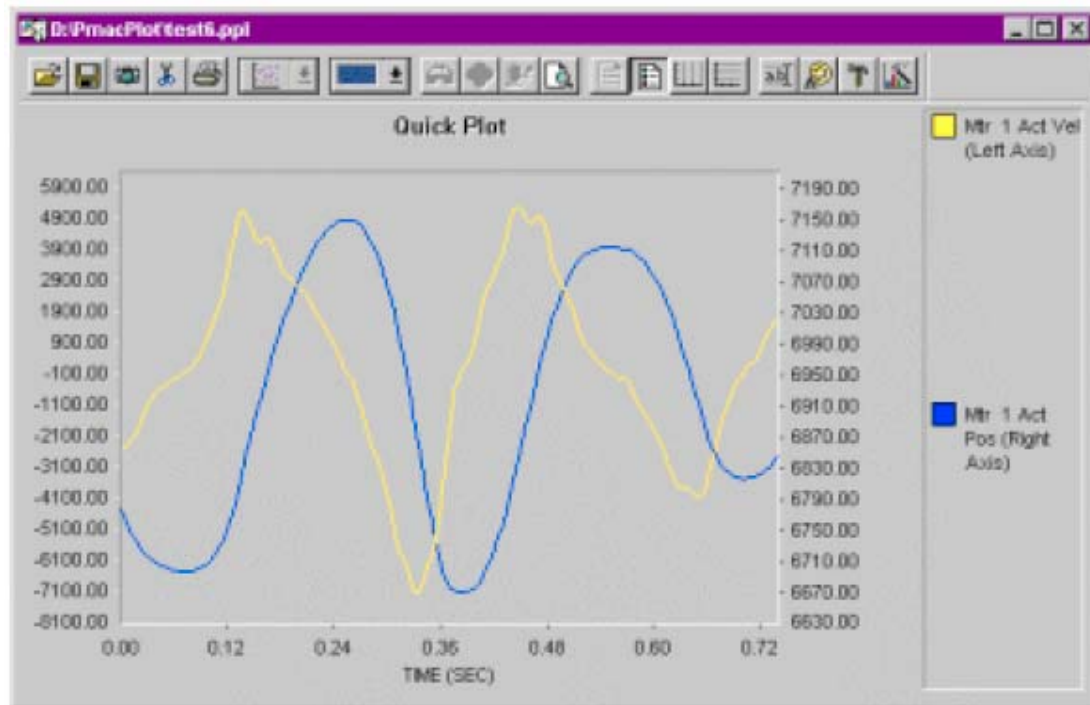
点击 Motors to Gather，设定需要采集的电机和采集周期。



然后选择我们要采集的数据，点击<< Add to Left 或者<<Add to Right；然后点击 Define Gather Buffer；需要开始采集时，点击 Begin Gathering；结束采集时，点击 End Gathering。最后点击 Upload Data；上载结束，出现提示：

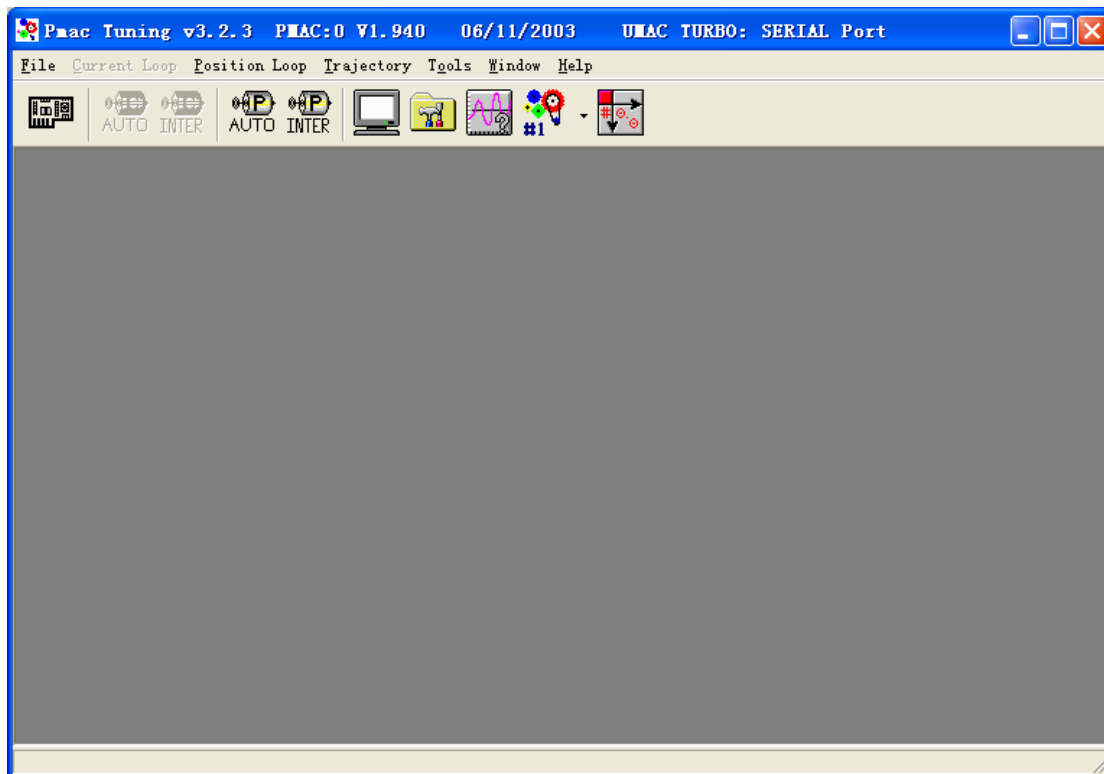


现在就可以使用 Plot Data，绘制我们需要的动态曲线。

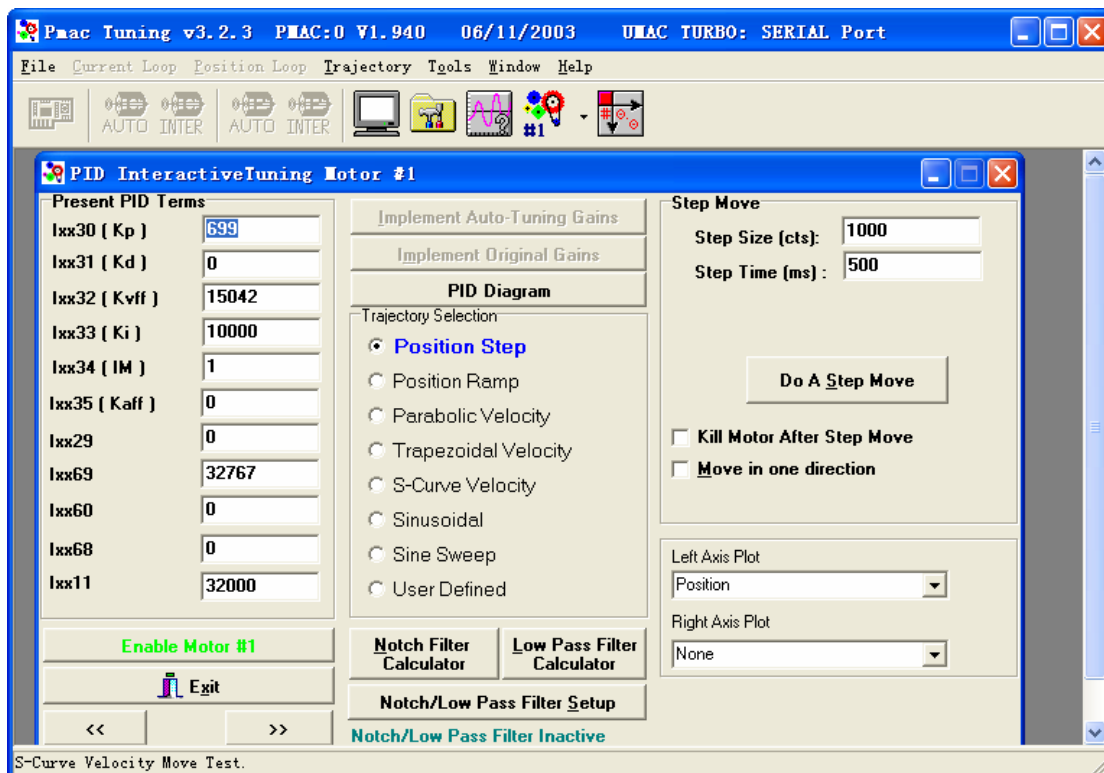


### 第三节 快速使用 PMAC Tuning PRO 整定 PID

打开 PMAC Tuning Pro 软件，如图：



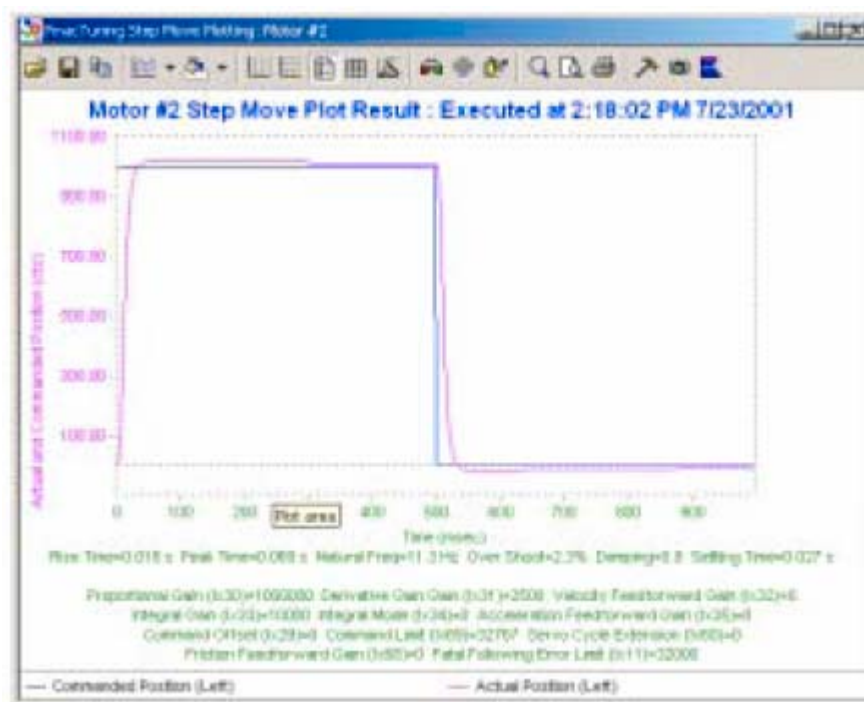
点击 INTER



选择 Position Step; 点击 Enable Motor #1, 然后点击 Do A Step Move 进行 PID 测试;



当出现上图的波形，表示超调，一般情况下，可以加大 I131，或者减小 I130；



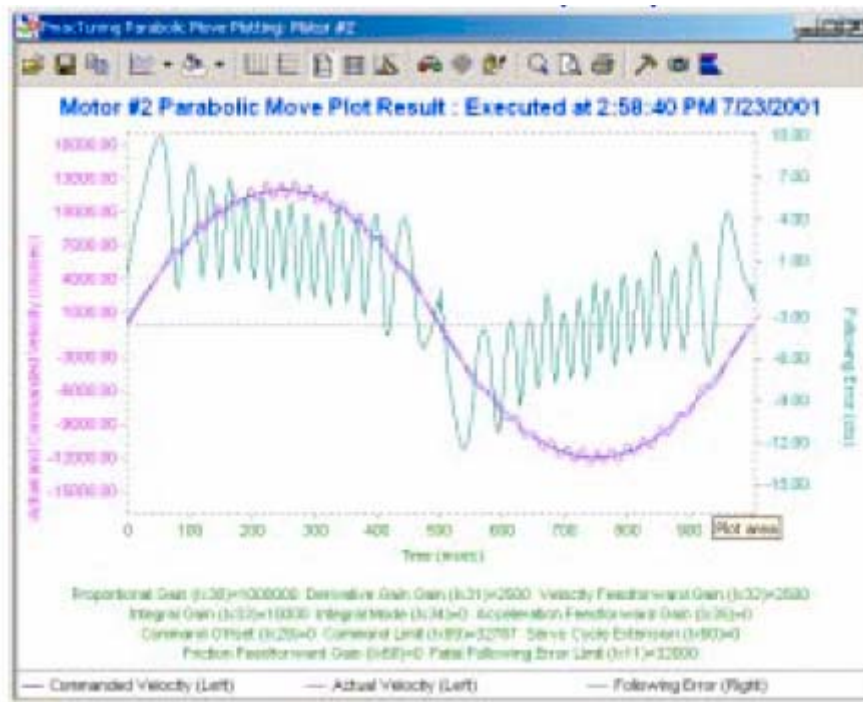
当出现上图的波形，表示整定时间过短，一般情况下，可以加大 I133；

当出现与蓝色理想波形很近的波形，表示当前 PID 很满意了。

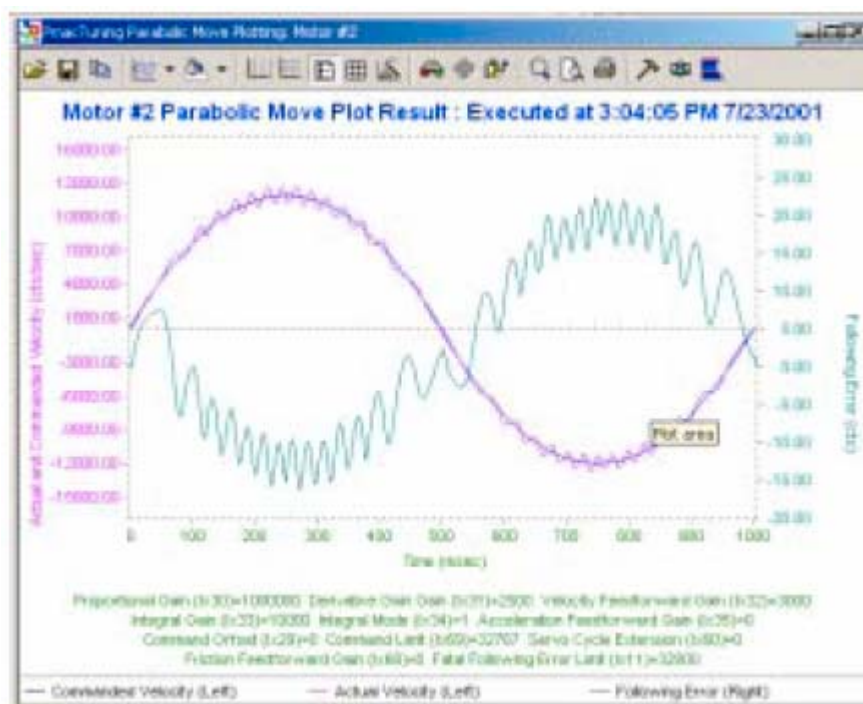
接下来就要调整跟随特性了

选择 Parabolic Velocity, 点击 Enable Motor #1, 然后点击 Do A Parabolic Move 进行 PID 测试；



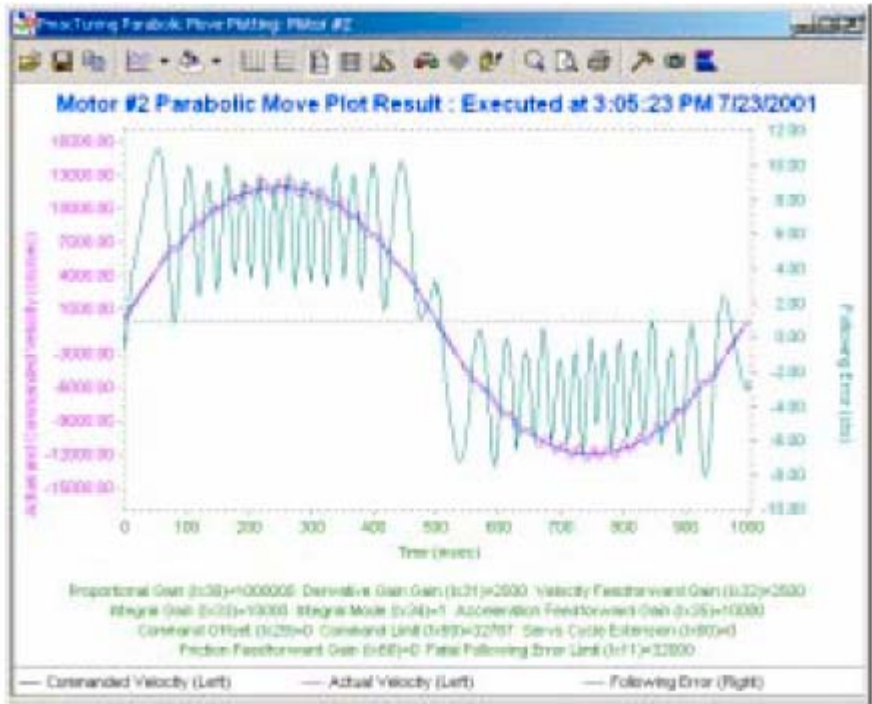


当出现上图的波形，跟随误差较大，一般情况下，可以加大 I132；



当出现上图的波形，跟随误差反向，一般情况下，表示 I132 过大，应该减小；





适当的调整 I135，可以使跟随误差均匀，并且很小，满足我们的要求。



利用  可以更改电机。

**注意：**如果你改变了电机，变量也需要相应改变，例如 2 号电机对应的 PID 参数为 I230，I231，等。

**注意：**如果你改变了变量的值，你必须用 **save** 这个命令将它保存，否则，当失去电源时，你的改动将丢失。

## 第三章      安装与配置 PMAC

## 第一节 跳线设定

在PMAC及其各种附件上，你将会见到许多跳线（金属制作的叉状物），叫做电子点。一些已经被短路；其它的是默认为开路的。这些跳线用来定制PMAC及其各种附件的一些硬件特性。强烈推荐使用每个跳线的默认配置。每个跳线的使用都应该详细查找硬件手册，确保不会引起不必要的麻烦。这里我们介绍一些常用的跳线，在使用过程中，它们的使用几率并不是很高。

### ● PMAC-PCI,PMAC-PCI-Lite的跳线设置：

E3 E4 E5 E6	伺服时钟频率，与I10变量有关，缺省为E5 E6为ON
E29 E30 E31 E32	相时钟频率，缺省为9.04KHZ,E31设为ON
E34 E35 E36 E37 E38	编码器采样时钟频率，缺省为：9.8304KHZ E34设为ON
E17A E17B E17C E17D E17E E17F E17G E17H	1-4电机的伺服使能的极性控制E24—E27
E40 E41 E42 E43	PMAC卡号的设置，缺省为E40—E43都设为ON,相当于@0号卡
E44 E45 E46 E47	设置串口通讯的波特率。缺省波特率为9600，E66设为ON
E51	PMAC卡初始化的设置，此跳线在PMAC卡正常工作时，应为OFF，卡由于故障或其他原因需要初始化时，才需将此跳线设为ON
E85 E87 E88	设置PMAC卡模拟量电源，缺省设置为都设为OFF状态，表示PMAC卡的模拟量电源+-12~-15V是通过外部供给（推荐），而如果都设为ON,表示模拟量+-12~-15电源是从总线得到
E89 E90	设置+/-LIMn和FLAG的电源供给，这两个跳线需要随着E85 E88 E89的设置变化而变化
E91 E92 E66 E67 E68 E69 E70 E71	PMAC卡占用计算机的总线地址设定。 缺省地址为：\$210(528),E66 E71设为OFF. 例：E66 E71设为OFF“OFF”=1 “ON”=0 总线地址 = (\$10 * E66) + (\$20 * E67) + (\$40 * E68) + (\$80 * E69) + (\$100 * E70) + (\$200 * E71) + (\$400 * E91) + (\$800 * E92) 对应\$210

### ● PMAC-PCI-MINI卡跳线设置

E110 E111 E112 E113 E114 E115	V/F转换的频率设置
E116 E117 E118 E119	V/F转换信号PULSE 和DIR 信号设置

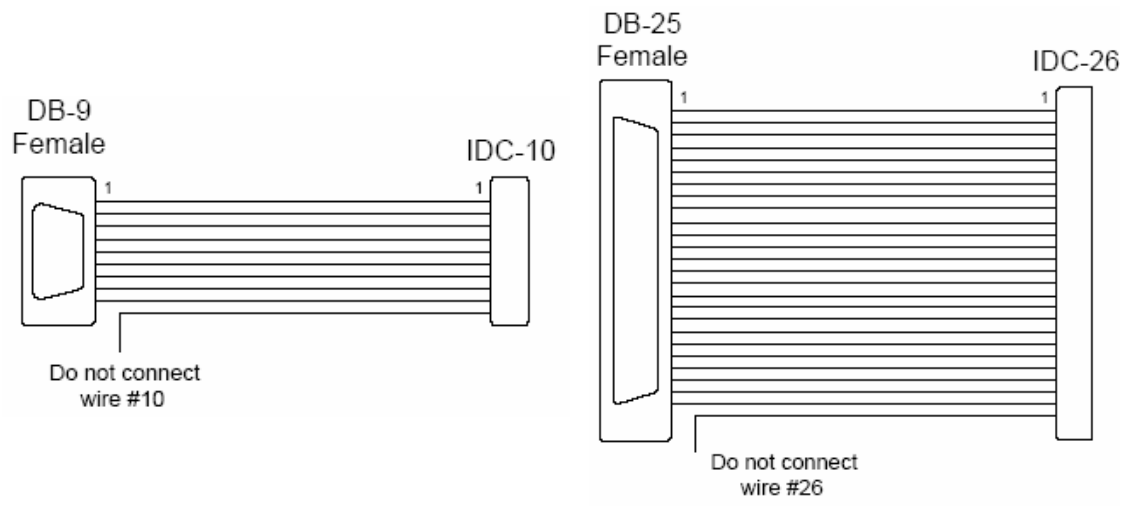
### ● PMAC2-PCI,PMAC2-PCI-Lite的跳线设置：

E1	PMAC2型卡的卡号设置，缺省设置为“0”号卡，E1为OFF
E3	PMAC2型卡从新初始化，此跳线用于2型卡在出现故障时，对卡进行初始化
E7 E8 E9 E10	PMAC2型卡中断PC机的选择设置
E14	PMAC2卡看门狗计时器控制

## 第二节 串口连接

当使用串口通讯时，我们可以采用DELTA TAU公司出产的附件，也可使用专用的附件ACC26，也可自己制作接线：

<b>ACC-3D</b>	3 米, RS-232 或者 RS-422 通信电缆, DB-25 转 IDC-26
<b>ACC-3E</b>	ACC-3D 的延长中继装置, 可使 ACC3D 最长 16 米, 15cm 大小
<b>ACC-3L</b>	3 米, RS-232 通信电缆, DB-9 转 IDC-10
<b>OPT-1</b>	DB-9 转 DB-25
<b>ACC-26A</b>	串行通信转换板, 必须 OPT 1, OPT 2 或者 OPT 3
<b>OPT-1</b>	上位机 RS-232 到 PMAC RS-422, 26 芯 60cm 电缆
<b>OPT-2</b>	RS-232 到 RS-232, 10 芯 60cm 电缆
<b>OPT-3</b>	RS-232 到 RS-422, (适用 GE Fanuc 系列 90-70 PLC)



## 第三节 与上位机连接

PMAC的通讯方式主要有PCI，ISA，USB，以太网以及串口。下面分别介绍他们的驱动程序如何安装。

**注意：**安装驱动之前，一定要确保我们的软件工具 Pewin32 Pro 或者开发工具 Pcomm32 Pro 已经安装。

**注意：**在最新版的用户光盘中，驱动程序是单独放在名字为“驱动程序”的文件夹下。

**注意：**在最新版的用户光盘中，所有软件、硬件手册都是由时间标注的，注意使用最新版说明手册。

### 3.3.1 安装驱动与上位机识别

- PCI、USB驱动程序

PCI、USB是属于**即插即用设备**（查看Microsoft Windows手册），当该设备与计算连接时，会自动被系统识别，我们只需要引导系统找到我们的驱动程序就可以了。不需用户做过多的设置。

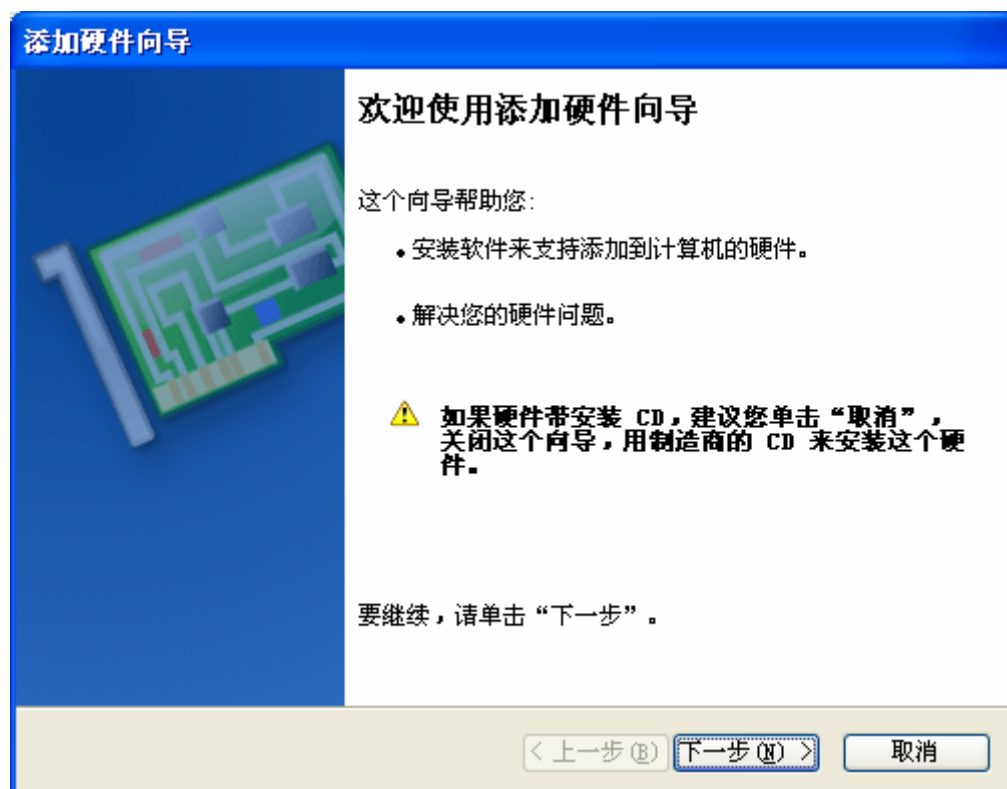
**注意：**USB 是支持热插拔的设备，Windows 系统在 USB 断电或者断开之后，会将立刻接上的 USB 设备分配一个新的 USB 设备号，这样我们原来构造的通讯就失败了，所以，在使用 USB 通讯时，当 USB 断电或者断开之后，建议用户等 5 分钟以上，让 Windows 系统将资源释放，我们就可以使用我们构造的通讯了。

- ISA，串口驱动程序

ISA、串口设备属于**非即插即用设备**（查看Microsoft Windows手册），当该设备与计算连接时，不会自动被系统识别，我们需要引导系统识别我们的设备，找到我们的驱动程序，需要用户做详细的设置。这里我们详细说一下：



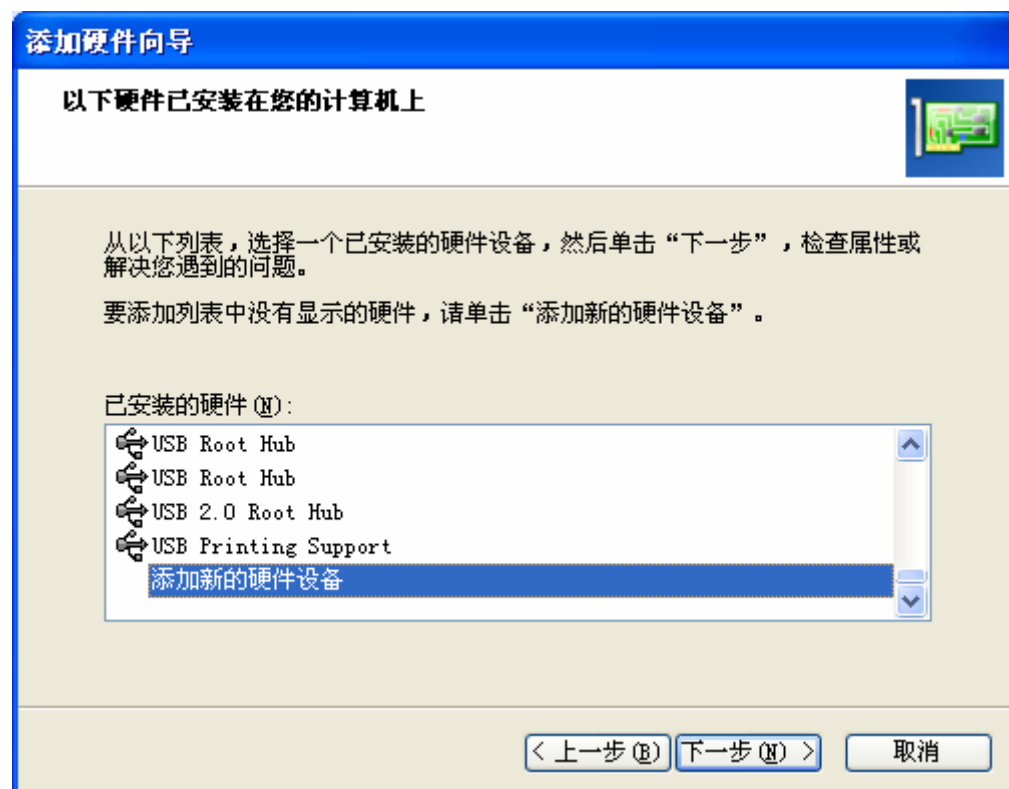
双击控制面板中的寻找新硬件



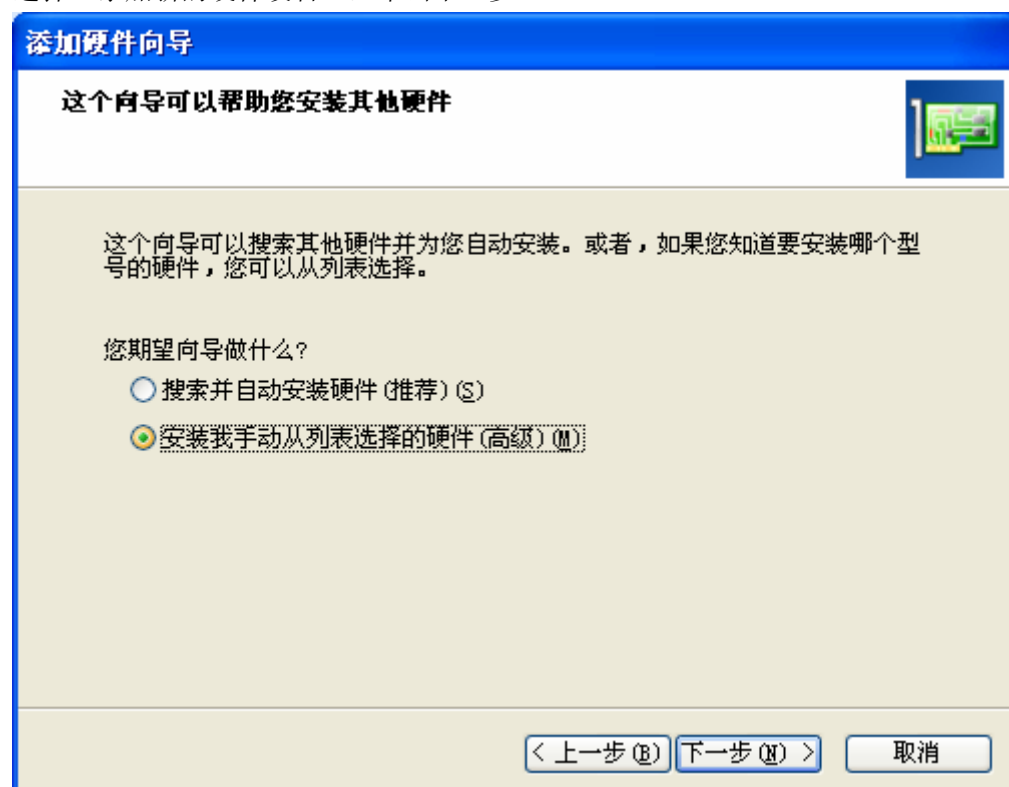
单击下一步



选择“是，我已经连接了此硬件 (Y)”，单击下一步



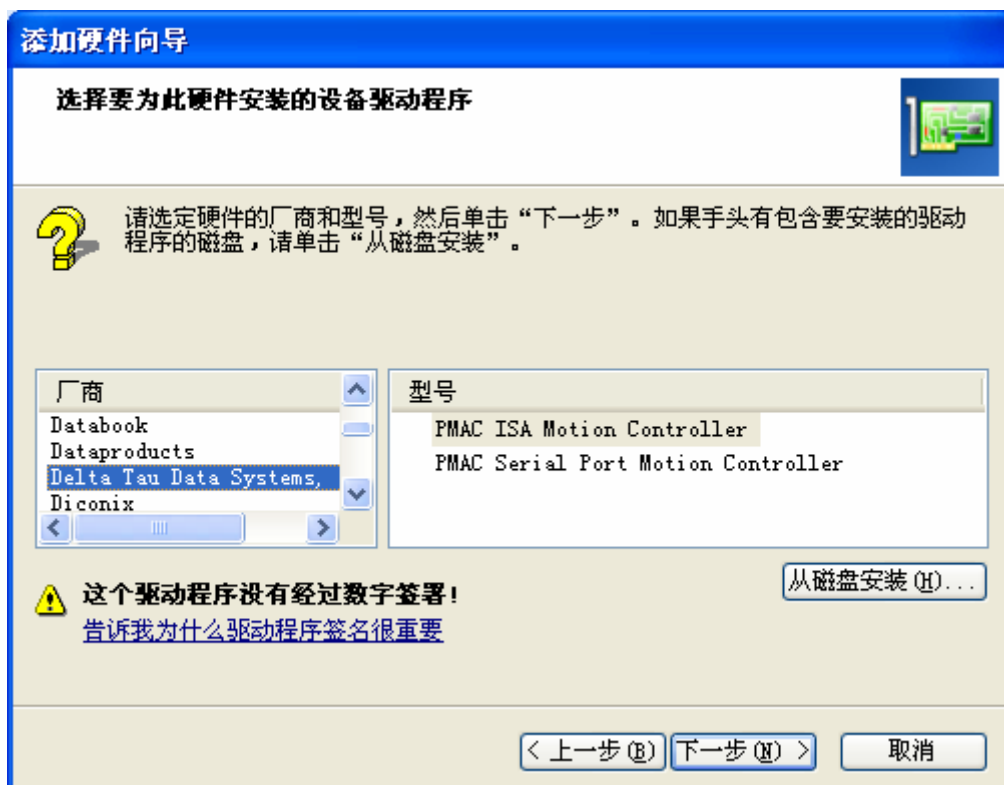
选择“添加新的硬件设备”，单击下一步



选择“安装握手东冲列表选择的硬件（高级）”，单击下一步



选择“显示所有设备”，单击下一步

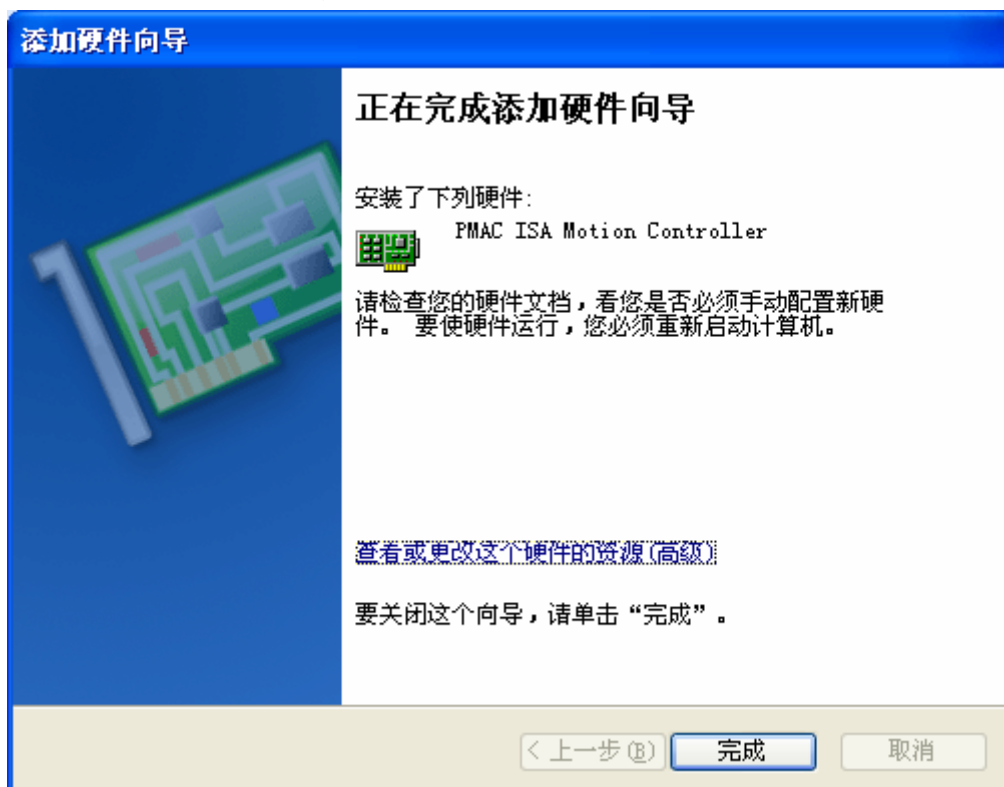


在厂商里面选择“Delta Tau Data Systems, Inc.”，如果是串口通讯，选择“PMAC Serial Port Motion Controller”，点击“下一步”，就安装完成了。如果是ISA通讯方式，选择“PMAC ISA Motion Controller” 点击“下一步”

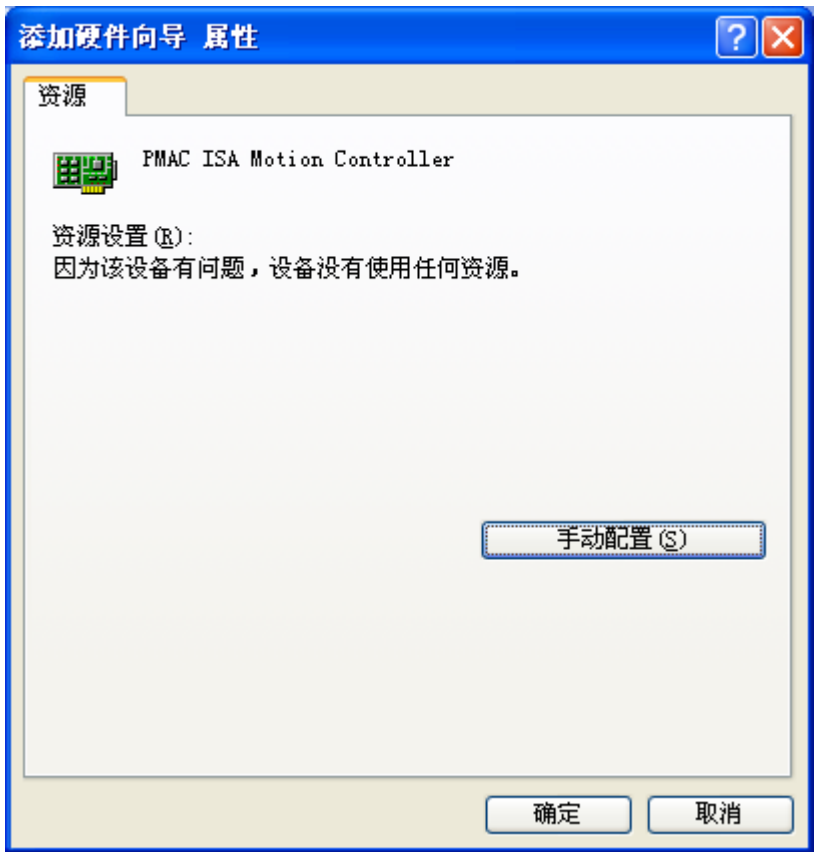




系统会查找驱动程序，出现



单击“查看或更改这个硬件的资源（高级）”

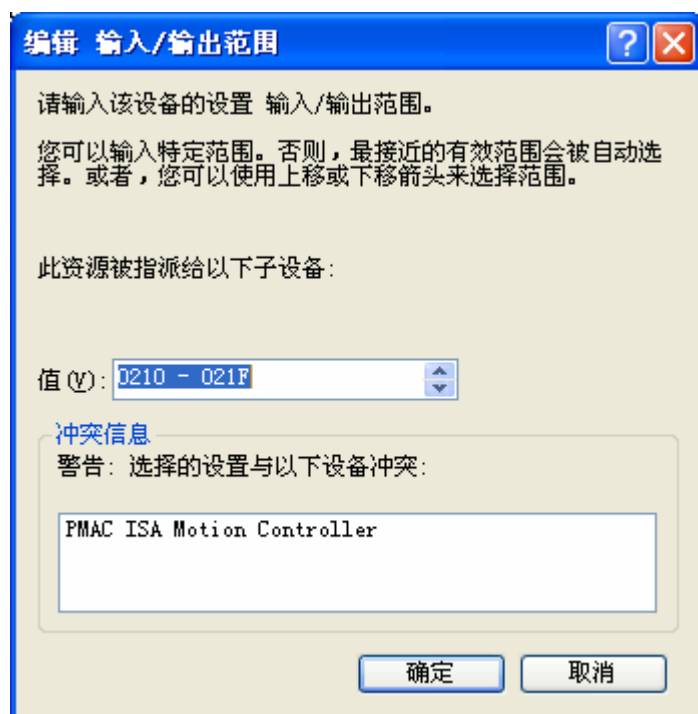


点击“手动配置 (S)”



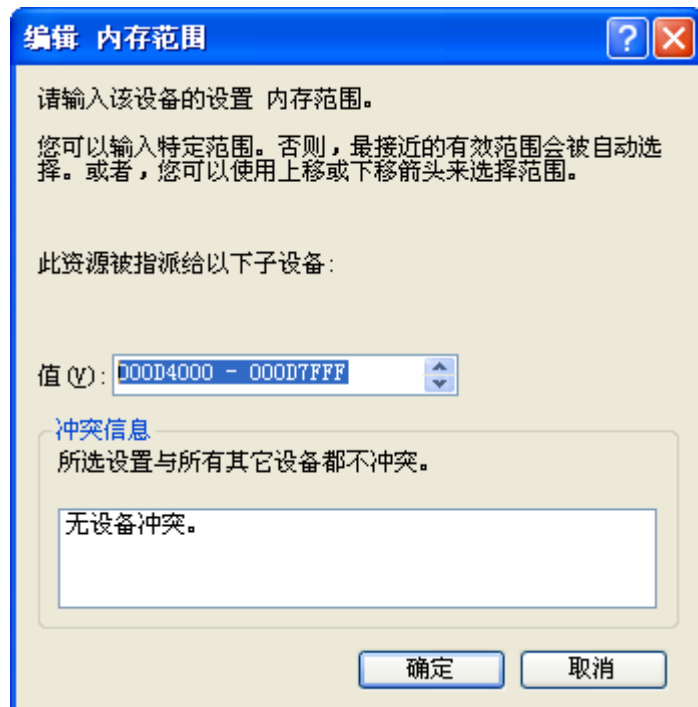
去掉“使用自动设置”的选择，就可以手动更改资源

如果是没有特殊的选择，使用基本配置0000，更改“输入/输出范围”的值为“0210 – 021F”



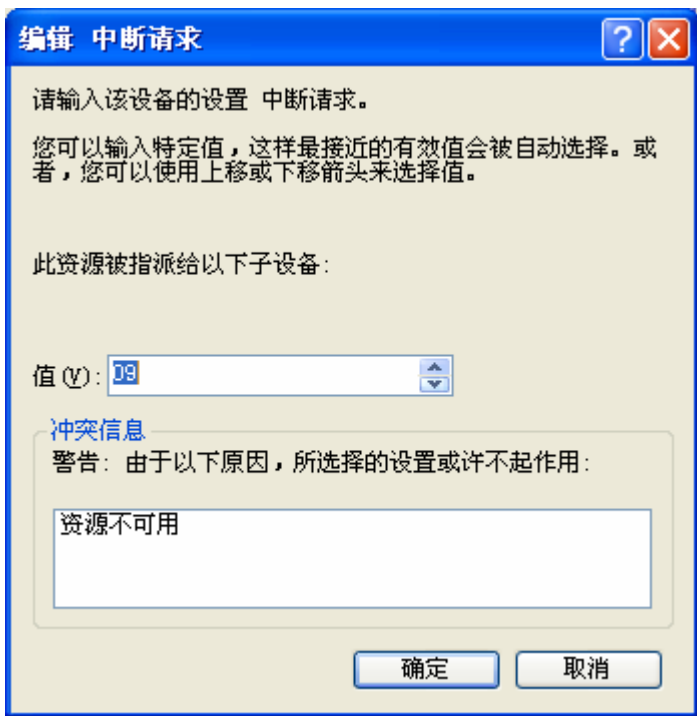
点击“确定”

如果是使用双端口RAM的选择，使用基本配置0001，更改“输入/输出范围”的值为“0210 – 021F”，更改“内存范围”的值为“000D4000 - 000D7FFF”



点击“确定”

如果是使用中断的选择，使用基本配置0002，更改“输入/输出范围”的值为“0210 – 021F”，更改“内存范围”的值为“000D4000 - 000D7FFF”，更改“中断请求”为适当值



当资源分配正确



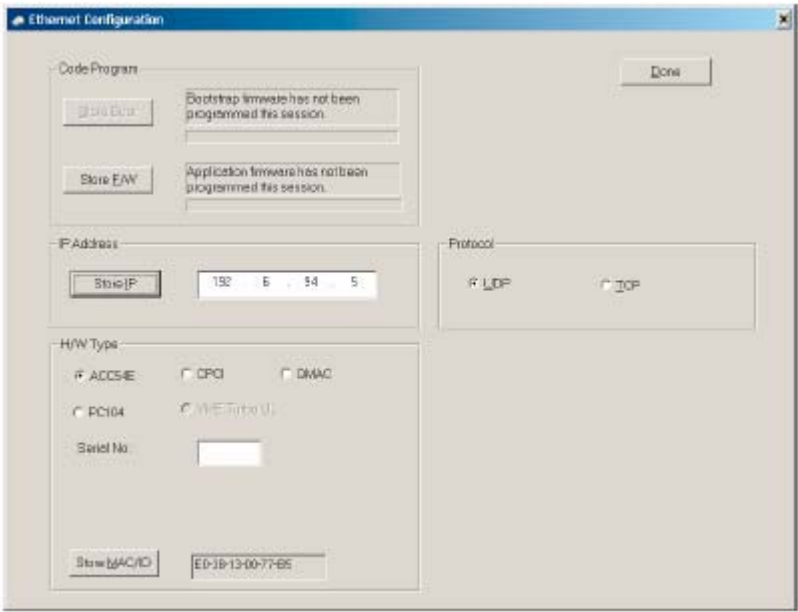
点击确定，就安装完毕。

- 以太网使用

以太网不需要安装自己的驱动程序，但需要先用USB为自己设置IP地址，使用Pewin32 Pro 自带的Configure Ethernet for PMAC或者Configure Ethernet 2.0 for PMAC（Pewin32 Pro 3.2.3.0版）



单击“OK”



在protocol选择“TCP”，在IP-Address里面填入“192.6.94.5”，点击“Store IP”.更改计算机的IP为“192.6.94.2”，就可以使用以太网通讯了。

### 3.3.2 复位 PMAC

标准的复位过程，如下

\$\$\$\*\*\*

P0..1023=0

Q0..1023=0

M0..1023->\* M0..1023=0

UNDEFINE ALL

SAVE

;全卡复位

;复位P变量

;复位Q变量

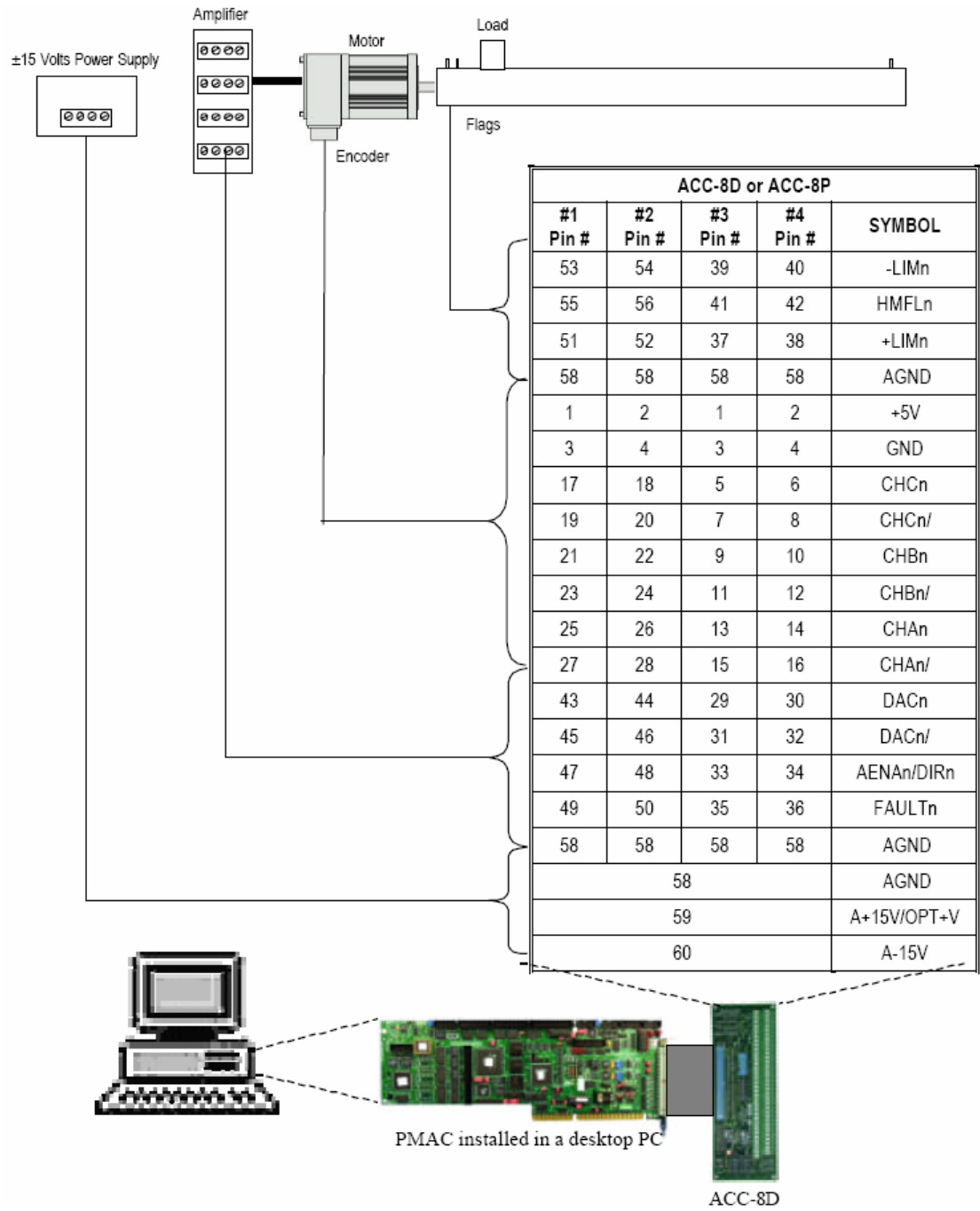
;复位M变量的定义和值

;复位坐标系定义

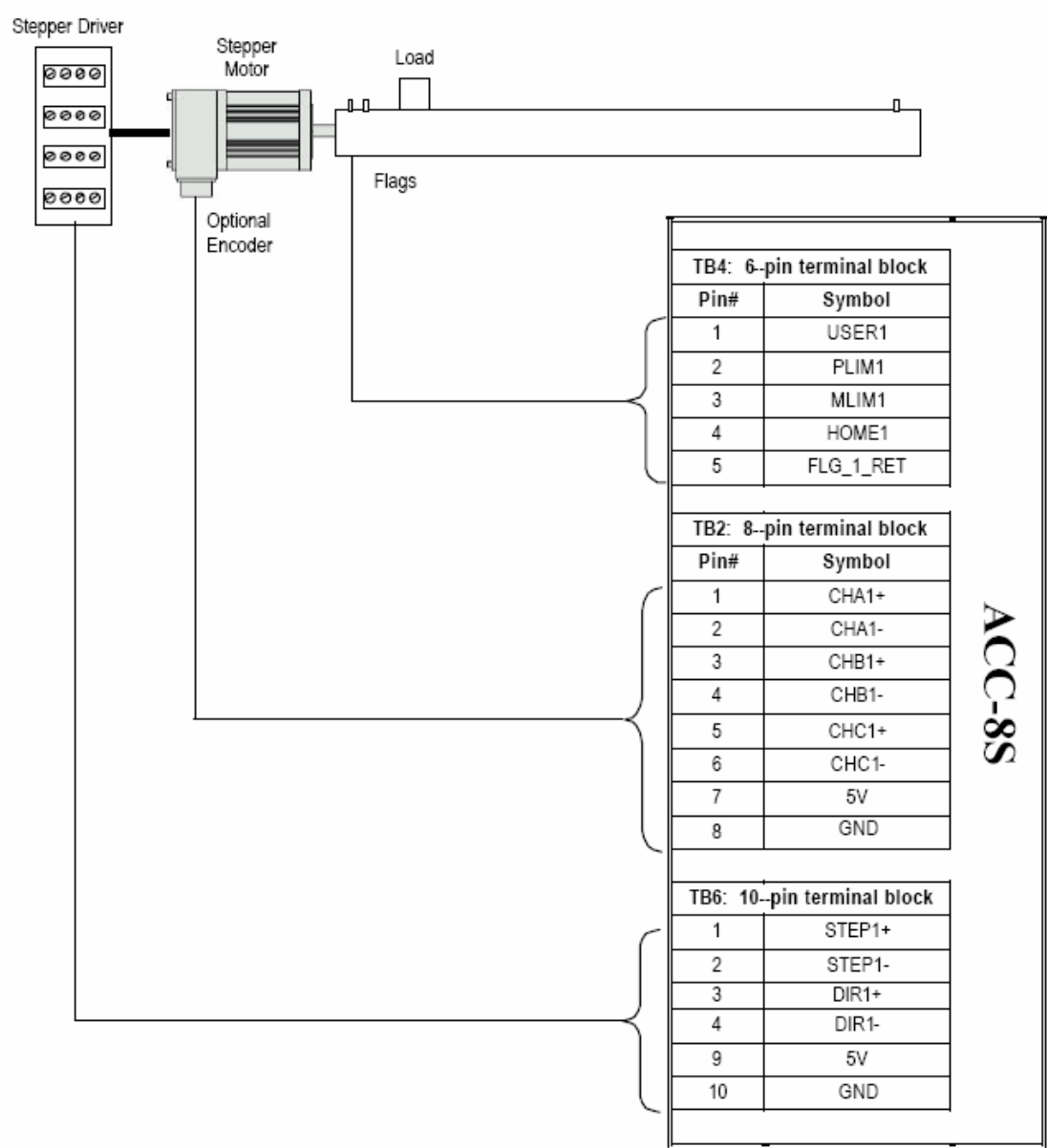
;保存清空的配置

## 第四节 连接端子 8D、8P、8S、8E

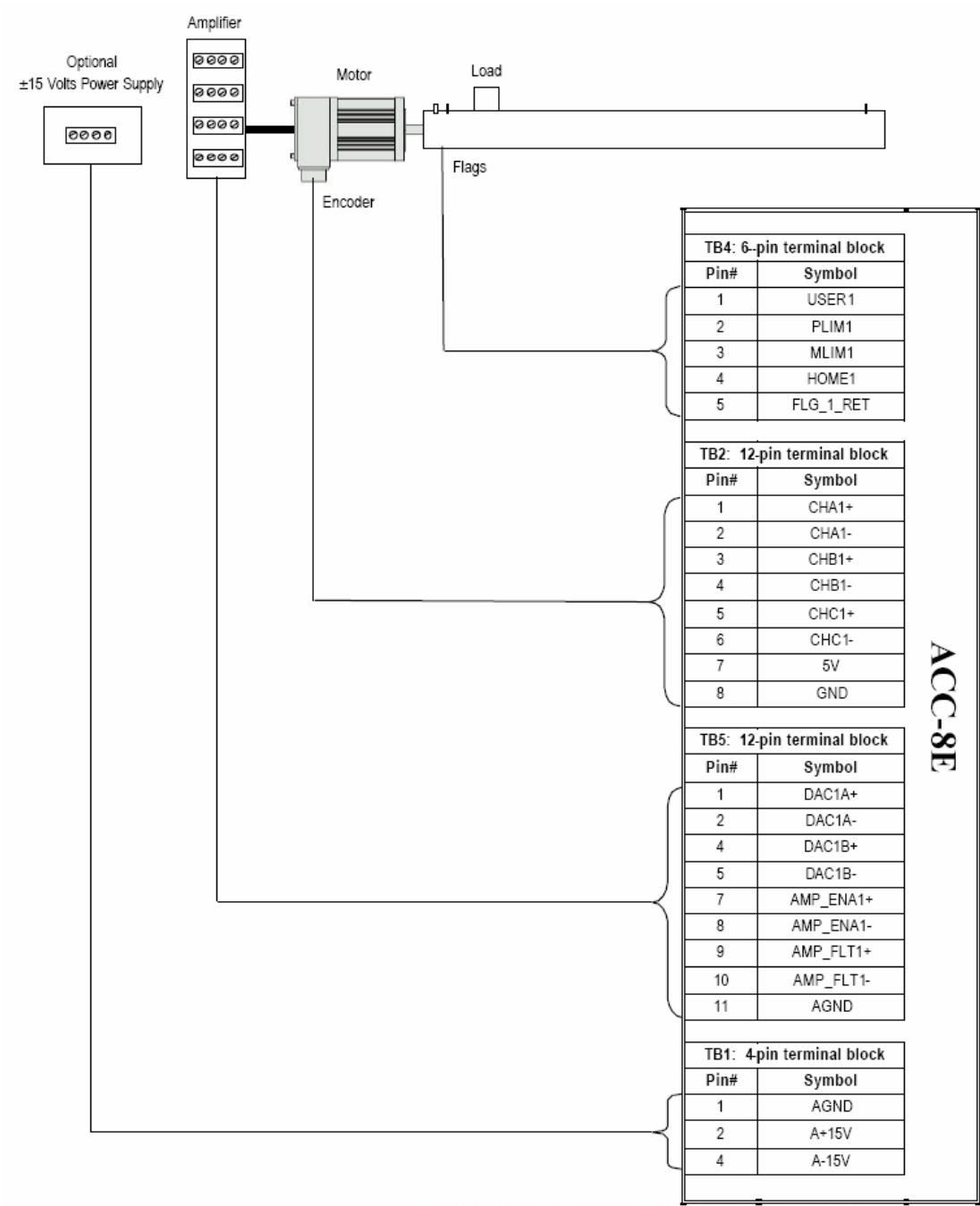
I型卡常见连接端子ACC-8P



2型卡常见连接端子ACC8S

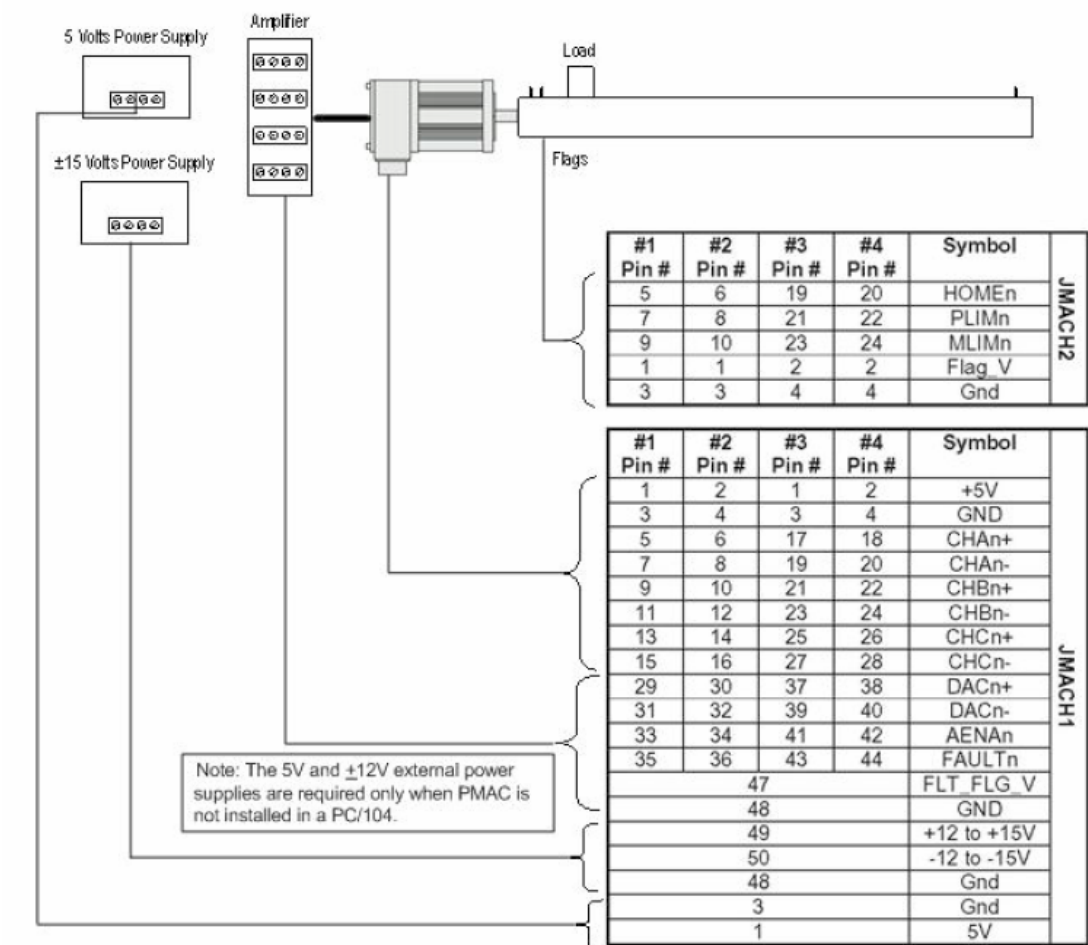


2型卡常见连接端子ACC8E

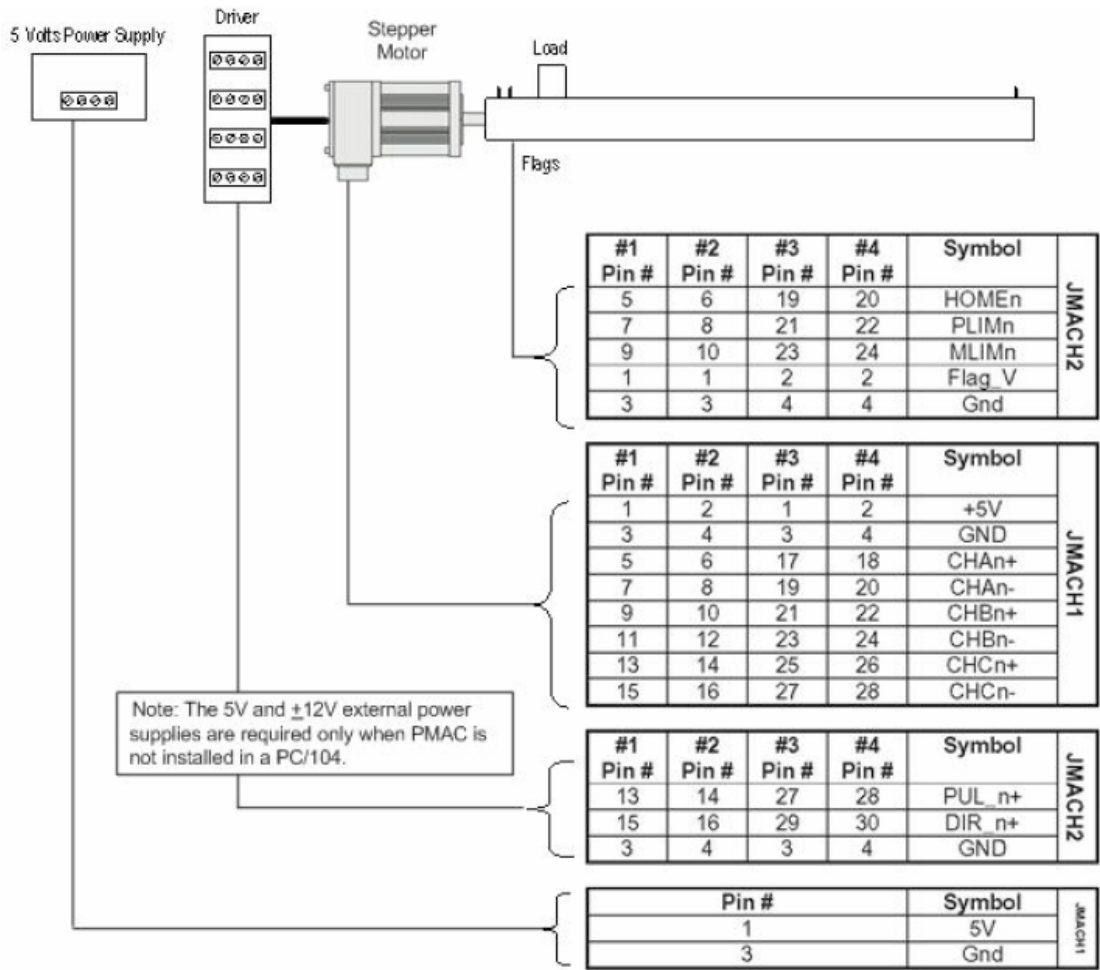




104型卡常见连接端子模拟量控制方式



104型卡常见连接端子脉冲+方向控制方式



## 第五节 电源指定

### 3.5.1 数字量支持

**1.5 A @ +5 V (+/-5%) (7.5 W)** (八个通道,每个通道编码器的典型负载)

如果PMAC被安装总线上,主机会提供5 V的电压供给。这个电压可以在能在连接端子ACC8P的1和3之间被测量,或者ACC8S TB6的9和10之间。独立使用时,PMAC需要使用外部的5 V供应,可使用TB1端子。

**注意:** 其他的数字量应用,还有自己的电流需求,所以,在使用PMAC的过程中,我们列举的仅是最低要求。

### 3.5.2 模拟量支持

**0.3A @ +12 to +15V (4.5W)**

**0.25A @ -12 to -15V (3.8W) (1型卡8通道输出典型负载)**

在PMAC上的DAC输出时,与计算芯片电路分开。这时在JMACH段子上使用58、59、60引进+12到 +15V和-12到 -15V的电压(推荐)。如果采用总线取电的方式,这时你需要将E85, E87, E88, 短路。

在PMAC2上的DAC输出时,与计算芯片电路分开。这时在ACC8E段子上使用TB1引进+12到 +15V和-12到 -15V的电压(推荐)。

**注意:** 其他的模拟量应用,还有自己的电流需求,所以,在使用PMAC的过程中,我们列举的仅是最低要求。

### 3.5.3 标志开关支持

无论是PMAC1还是PMAC2都提供+LIMn, -LIMn (行程限位), HMFLn (零点输入), FAULTn (电机报警), 一般情况下, 都需要12 to 24V的电压支持。我们在下一节, 将详细讨论。

## 第六节 行程限位、回零开关

PMAC 提供了一些固化功能的标准 I/O,最重要的是行程限位回零,这些信号为 PMAC 的安全和准确性提供了保证。

### 3.6.1 关闭行程限位功能

可以通过 Ix25 (X 代指电机,例: 1 号电机为 i125, 5 号电机为 i525) 关闭电机的限位功能,在 Pewin32 Pro 的 Terminal 窗体里键入如下命令即可,

I125=I125|\$20000 ;电机 #1

I225=I225|\$20000 ;电机 #2

I325=I325|\$20000 ;电机 #3

I425=I425|\$20000 ;电机 #4

I525=I525|\$20000 ;电机 #5

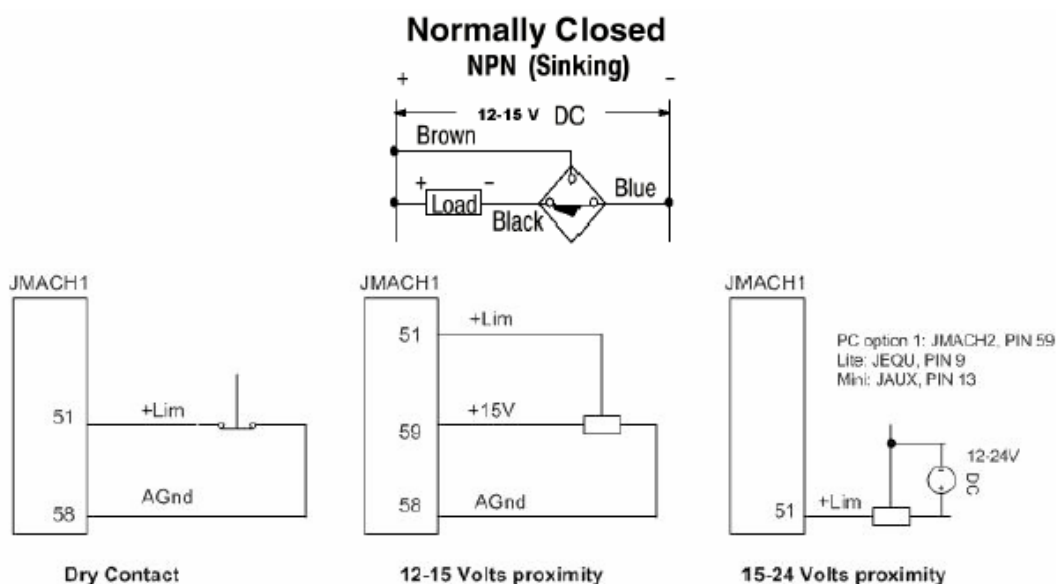
I625=I625|\$20000 ;电机 #6

I725=I725|\$20000 ;电机 #7

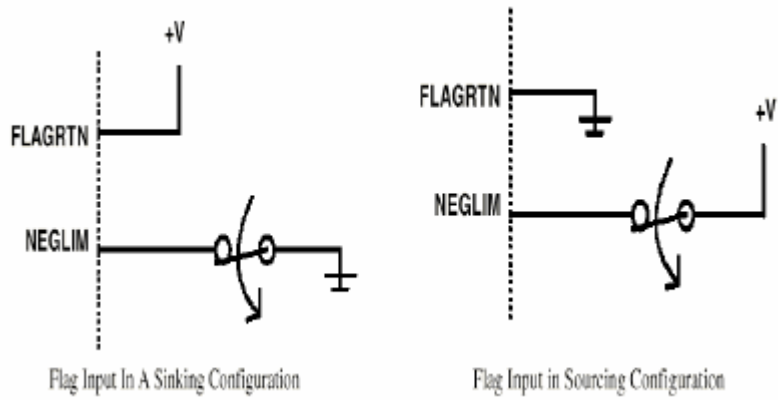
I825=I825|\$20000 ;电机 #8

### 3.6.2 行程限位开关形式

PMAC I 型的接口形式:



PMAC 2 型的接口形式:



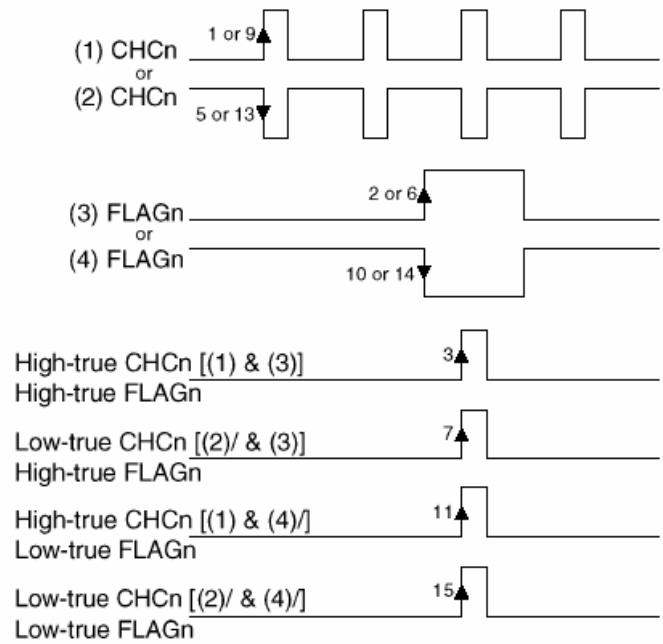
PMAC 2 型 104 的接口形式:



### 3.6.3 回零开关

PMAC应用中，行程限位开关需要常闭开关，但零点开关可以通过变量来更改到用户需要的形式：

- PMAC I 型卡使用 I902 ， I907,.,.。 I977。
- PMAC 2 型卡使用 I9n2 (n 代指电机，例：1 号电机为 i912，5 号电机为 i952)。



### 3.6.4 检查标志输入

可以在 Pwin32 Pro 的软件里面，使用 Watch 窗体检查各个标志信号的输入。

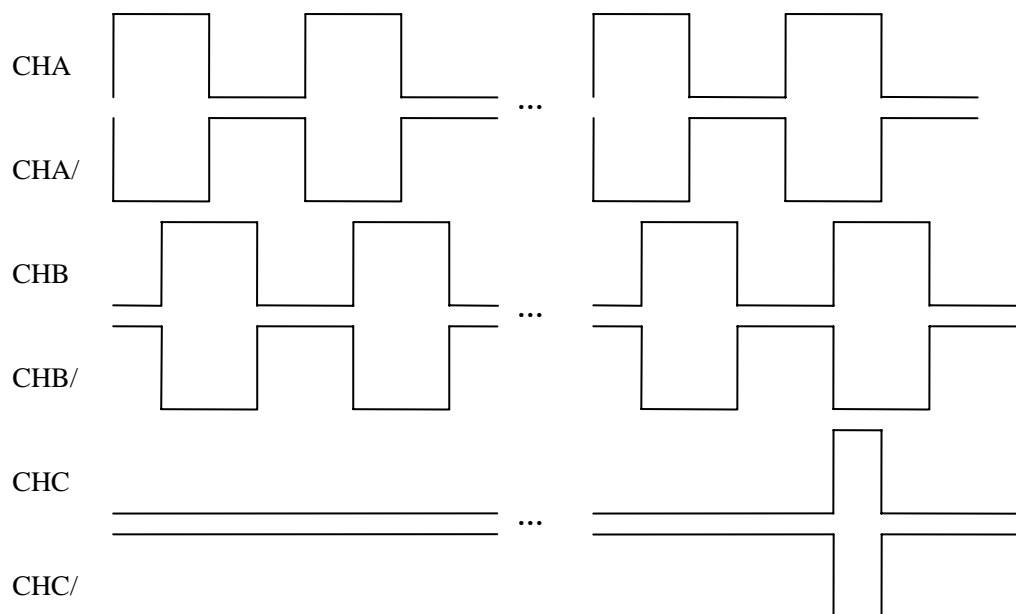
PMAC 1 型卡				
	电机#1	电机#2	电机#3	电机#4
回零信号	M120->X:\$C000,20	M220->X:\$C004,20	M320->X:\$C008,20	M420->X:\$C00C,20
-LIM 信号	M121->X:\$C000,21	M221->X:\$C004,21	M321->X:\$C008,21	M421->X:\$C00C,21
+LIM 信号	M122->X:\$C000,22	M222->X:\$C004,22	M322->X:\$C008,22	M422->X:\$C00C,22
	电机#5	电机#6	电机#7	电机#8
回零信号	M520->X:\$C010,20,1	M620->X:\$C014,20	M720->X:\$C018,20	M820->X:\$C01C,20
-LIM 信号	M521->X:\$C010,21	M621->X:\$C014,21	M721->X:\$C018,21	M821->X:\$C01C,21
+LIM 信号	M522->X:\$C010,22	M622->X:\$C014,22	M722->X:\$C018,22	M822->X:\$C01C,22
PMAC 2 型卡				
	电机#1	电机#2	电机#3	电机#4
回零信号	M120->X:\$C000,16	M220->X:\$C008,16	M320->X:\$C010,16	M420->X:\$C018,16
-LIM 信号	M121->X:\$C000,17	M221->X:\$C008,17	M321->X:\$C010,17	M421->X:\$C018,17
+LIM 信号	M122->X:\$C000,18	M222->X:\$C008,18	M322->X:\$C010,18	M422->X:\$C018,18
	电机#5	电机#6	电机#7	电机#8
回零信号	M520->X:\$C020,16	M620->X:\$C028,16	M720->X:\$C030,16	M820->X:\$C038,16
-LIM 信号	M521->X:\$C020,17	M621->X:\$C028,17	M721->X:\$C030,17	M821->X:\$C038,17
+LIM 信号	M522->X:\$C020,18	M622->X:\$C028,18	M722->X:\$C030,18	M822->X:\$C038,18

## 第七节 电机控制信号连接

### 3.7.1 编码器信号连接

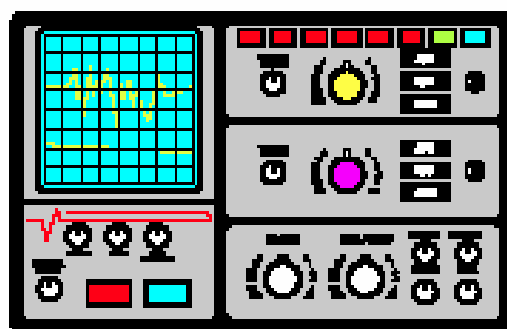
大部分电机反馈的编码器都是增量性长线驱动的形式，这样就可以直接将编码器的输出端与 JMACH 端子或者 ACC8S、ACC8E 之类的附件板连接。JMACH 端子或者 ACC8S、ACC8E 之类的附件板还提供 5V 电源，这样还可以为无源的反馈元件提供电源。

增量性编码器反馈的信号通常情况下是正交的信号，常用的信号是长线驱动形式，如图：



### 3.7.2 检查编码器输入

当正常情况下，当编码器输入时，我们可以在 PEwin32 Pro 的 Position 窗体里面看到编码器的反馈，也可以通过 P 或者<Ctrl + P>的命令查看编码器的反馈。如果你的编码器连接正确，但在软件窗体得不到正确的反馈信号，请查看相应通道的变量（Ix03,Ix04,I900, I905…(PMAC),I9n0(PMAC2)等），或者用示波器查看编码器的反馈。



### 3.7.3 检查 DAC 输出(1 型卡)

我们可以通过 M 变量监视 DAC 的输出，也可通过用万用表进行检测。

PMAC I 型卡的 DAC 输出地址：

电机#1	电机#2	电机#3	电机#4
M102->Y:\$C003,8,16,S	M202->Y:\$C002,8,16,S	M302->Y:\$C00B,8,16,S	M402->Y:\$C00A,8,16,S
电机#5	电机#6	电机#7	电机#8
M502->Y:\$C013,8,16,S	M602->Y:\$C012,8,16,S	M702->Y:\$C01B,8,16,S	M802->Y:\$C01A,8,16,S



例子:  
M102->Y:\$C003,8,16,S  
I100=0  
M102=16383  
<在JMACH1的43与58之间输出5V, (ACC-8D or ACC-8P)>  
M102=-16383  
<<在JMACH1的43与58之间输出-5V, (ACC-8D or ACC-8P)>

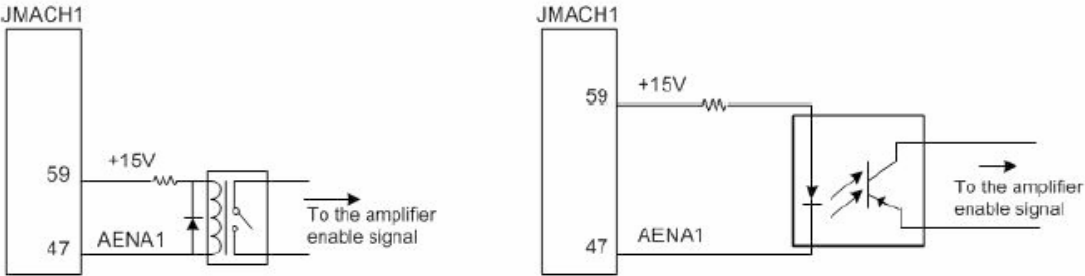
3.7.4 检查 PFM 输出（2 型卡）

输出的是脉冲和方向信号，可以借助示波器，或连接的伺服电机的技术功能。

3.7.5 电机使能信号(AENAx/DIRx)

大多数的电机都有使能/禁能的一个开关，来控制电机是否可以接受命令并响应。PMAC 的AENA端子就是完成这个目的的。

在 PMAC I 型卡中，JMACH 上的 47 和 59 端子就是#1 电机的使能信号，我们可以通过条线老改变输出的极性。



在 PMAC 2 型卡，提供的是开关形式。  
我们也可以在 ix00=0 的情况下，通过 M 变量来测试这个端子。

PMAC 1 型卡			
电机#1	电机#2	电机#3	电机#4
M114->X:\$C000,14	M214->X:\$C004,14	M314->X:\$C008,14	M414->X:\$C00C,14
电机#5	电机#6	电机#7	电机#8
M514->X:\$C010,14	M614->X:\$C014,14	M714->X:\$C018,14	M814->X:\$C01C,14
PMAC 2 型卡			
电机#1	电机#2	电机#3	电机#4
M114->X:\$C005,14	M214->X:\$C00D,14	M314->X:\$C015,14	M414->X:\$C01D,14
电机#5	电机#6	电机#7	电机#8



M514->X:\$C025,14	M614->X:\$C02D,14	M714->X:\$C035,14	M814->X:\$C03D,14
-------------------	-------------------	-------------------	-------------------

### 3.7.6 电机报警 (Faultx)

这个输入能使 PMAC 得知电机出错情况，及时终止计算转入错误处理。在 PMAC 上我们也可以通过 Ix25(例如：I125 对于#1 电机) 变量来禁止报警信号。使用这个信号，可以加强 PMAC 对电机的控制能力。

我们也可以在 ix00=0 的情况下，通过 M 变量来测试这个端子。

PMAC 1 型卡			
电机#1	电机#2	电机#3	电机#4
M123->X:\$C000,23	M223->X:\$C004,23	M323->X:\$C008,23	M423->X:\$C00C,23
电机#5	电机#6	电机#7	电机#8
M523->X:\$C010,23	M623->X:\$C014,23	M723->X:\$C018,23	M823->X:\$C01C,23
PMAC 2 型卡			
电机#1	电机#2	电机#3	电机#4
M123->X:\$C000,15	M223->X:\$C008,15	M323->X:\$C010,15	M423->X:\$C018,15
电机#5	电机#6	电机#7	电机#8
M523->X:\$C020,15	M623->X:\$C028,15	M723->X:\$C030,15	M823->X:\$C038,15

### 3.7.7 通用 I/O

PMAC 的通用 I/O 的连接提供 8in/8out，可以为 PMAC 提供更有效的功能。

；通用I/O-J5(PMAC-PC,-Lite,-VME)

M1->Y:\$FFC2,8,1 ; Machine Output 1

M2->Y:\$FFC2,9,1 ; Machine Output 2

M3->Y:\$FFC2,10,1 ; Machine Output 3

M4->Y:\$FFC2,11,1 ; Machine Output 4

M5->Y:\$FFC2,12,1 ; Machine Output 5

M6->Y:\$FFC2,13,1 ; Machine Output 6

M7->Y:\$FFC2,14,1 ; Machine Output 7

M8->Y:\$FFC2,15,1 ; Machine Output 8

M9->Y:\$FFC2,8,8,U ; Machine Outputs 1-8 treated as byte

M11->Y:\$FFC2,0,1 ; Machine Input 1

M12->Y:\$FFC2,1,1 ; Machine Input 2

M13->Y:\$FFC2,2,1 ; Machine Input 3

M14->Y:\$FFC2,3,1 ; Machine Input 4

M15->Y:\$FFC2,4,1 ; Machine Input 5

M16->Y:\$FFC2,5,1 ; Machine Input 6

M17->Y:\$FFC2,6,1 ; Machine Input 7

M18->Y:\$FFC2,7,1 ; Machine Input 8

M19->Y:\$FFC2,0,8,U ; Machine Inputs 1-8 treated as byte

例：

M1=1 ;J5 的 31 脚输出

；多路I/O-J3(PMAC-PC, -Lite, & -VME)

M40->Y:\$FFC1,8,1 ; SEL0 Output

M41->Y:\$FFC1,9,1 ; SEL1 Output

M42->Y:\$FFC1,10,1 ; SEL2 Output

M43->Y:\$FFC1,11,1 ; SEL3 Output

M44->Y:\$FFC1,12,1 ; SEL4 Output

M45->Y:\$FFC1,13,1 ; SEL5 Output

M46->Y:\$FFC1,14,1 ; SEL6 Output

M47->Y:\$FFC1,15,1 ; SEL7 Output

M48->Y:\$FFC1,8,8,U ; SEL0-7 Outputs treated as a byte

M50->Y:\$FFC1,0,1 ; DAT0 Input

M51->Y:\$FFC1,1,1 ; DAT1 Input

M52->Y:\$FFC1,2,1 ; DAT2 Input

M53->Y:\$FFC1,3,1 ; DAT3 Input

M54->Y:\$FFC1,4,1 ; DAT4 Input

M55->Y:\$FFC1,5,1 ; DAT5 Input

M56->Y:\$FFC1,6,1 ; DAT6 Input

M57->Y:\$FFC1,7,1 ; DAT7 Input

M58->Y:\$FFC1,0,8,U ; DAT0-7 Inputs treated as a byte

例：

m40=1 ; J3 的 4 脚输出

；JI/O-J3 (PMAC2-PC,-Lite)

M0->Y:\$C080,0 ; I/O00 Data Line; J3 Pin 1

M1->Y:\$C080,1 ; I/O01 Data Line; J3 Pin 2

M2->Y:\$C080,2 ; I/O02 Data Line; J3 Pin 3

M3->Y:\$C080,3 ; I/O03 Data Line; J3 Pin 4

M4->Y:\$C080,4 ; I/O04 Data Line; J3 Pin 5

M5->Y:\$C080,5 ; I/O05 Data Line; J3 Pin 6

M6->Y:\$C080,6 ; I/O06 Data Line; J3 Pin 7

M7->Y:\$C080,7 ; I/O07 Data Line; J3 Pin 8

M8->Y:\$C080,8 ; I/O08 Data Line; J3 Pin 9

M9->Y:\$C080,9 ; I/O09 Data Line; J3 Pin 10

M10->Y:\$C080,10 ; I/O10 Data Line; J3 Pin 11

M11->Y:\$C080,11 ; I/O11 Data Line; J3 Pin 12

M12->Y:\$C080,12 ; I/O12 Data Line; J3 Pin 13

M13->Y:\$C080,13 ; I/O13 Data Line; J3 Pin 14

M14->Y:\$C080,14 ; I/O14 Data Line; J3 Pin 15

M15->Y:\$C080,15 ; I/O15 Data Line; J3 Pin 16

M16->Y:\$C080,16 ; I/O16 Data Line; J3 Pin 17

M17->Y:\$C080,17 ; I/O17 Data Line; J3 Pin 18

M18->Y:\$C080,18 ; I/O18 Data Line; J3 Pin 19

M19->Y:\$C080,19 ; I/O19 Data Line; J3 Pin 20

M20->Y:\$C080,20 ; I/O20 Data Line; J3 Pin 21

M21->Y:\$C080,21 ; I/O21 Data Line; J3 Pin 22  
 M22->Y:\$C080,22 ; I/O22 Data Line; J3 Pin 23  
 M23->Y:\$C080,23 ; I/O23 Data Line; J3 Pin 24  
 M24->Y:\$C081,0 ; I/O24 Data Line; J3 Pin 25  
 M25->Y:\$C081,1 ; I/O25 Data Line; J3 Pin 26  
 M26->Y:\$C081,2 ; I/O26 Data Line; J3 Pin 27  
 M27->Y:\$C081,3 ; I/O27 Data Line; J3 Pin 28  
 M28->Y:\$C081,4 ; I/O28 Data Line; J3 Pin 29  
 M29->Y:\$C081,5 ; I/O29 Data Line; J3 Pin 30  
 M30->Y:\$C081,6 ; I/O30 Data Line; J3 Pin 31  
 M31->Y:\$C081,7 ; I/O31 Data Line; J3 Pin 32  
 M32->X:\$C080,0,8 ; Direction control for I/O00 to I/O07  
 M33->Y:\$E800,0 ; Buffer direction control for I/O00 to I/O07  
 M34->X:\$C080,8,8 ; Direction control for I/O08 to I/O15  
 M35->Y:\$E800,1 ; Buffer direction control for I/O08 to I/O15  
 M36->X:\$C080,16,8 ; Direction control for I/O16 to I/O23  
 M37->Y:\$E800,2 ; Buffer direction control for I/O16 to I/O23  
 M38->X:\$C081,0,8 ; Direction control for I/O24 to I/O31  
 M39->Y:\$E800,3 ; Buffer direction control for I/O24 to I/O31

例:

Open PLC1 Clear

M32=\$0

M34=\$0

M36=\$FF

M38=\$FF

M33=\$0

M35=\$0

M37=\$1

M38=\$1

Disable plc1

Close ; 定义 J3 的 I/O00-I/O15 是输入, I/O16-I/O31 是输出。

**; JTHW- J2(PMAC2-PC,-Lite)**

M40->Y:\$C082,8 ; SEL0 Line; J2 Pin 4

M41->Y:\$C082,9 ; SEL1 Line; J2 Pin 6

M42->Y:\$C082,10 ; SEL2 Line; J2 Pin 8

M43->Y:\$C082,11 ; SEL3 Line; J2 Pin 10

M44->Y:\$C082,12 ; SEL4 Line; J2 Pin 12

M45->Y:\$C082,13 ; SEL5 Line; J2 Pin 14

M46->Y:\$C082,14 ; SEL6 Line; J2 Pin 16

M47->Y:\$C082,15 ; SEL7 Line; J2 Pin 18

M48->Y:\$C082,8,8,U ; SEL0-7 Lines treated as a byte

M50->Y:\$C082,0 ; DAT0 Line; J2 Pin 3

M51->Y:\$C082,1 ; DAT1 Line; J2 Pin 5

M52->Y:\$C082,2 ; DAT2 Line; J2 Pin 7

M53->Y:\$C082,3 ; DAT3 Line; J2 Pin 9  
 M54->Y:\$C082,4 ; DAT4 Line; J2 Pin 11  
 M55->Y:\$C082,5 ; DAT5 Line; J2 Pin 13  
 M56->Y:\$C082,6 ; DAT6 Line; J2 Pin 15  
 M57->Y:\$C082,7 ; DAT7 Line; J2 Pin 17  
 M58->Y:\$C082,0,8,U ; DAT0-7 Lines treated as a byte  
 M60->X:\$C082,0,8 ; Direction control for DAT0 to DAT7  
 M61->Y:\$E800,4 ; Buffer direction control for DAT0 to DAT7, PCbus  
 ;M61->Y:\$E802,0 ; Buffer direction control for DAT0 to DAT7, VMEbus  
 M62->X:\$C080,8,8 ; Direction control for SEL0 to SEL7  
 M63->Y:\$E800,5 ; Buffer direction control for SEL0 to SEL7, PCbus  
 ;M63->Y:\$E802,1 ; Buffer direction control for SEL0 to SEL7, VMEbus

### **PMAC2A-PC/104 ACC1P- J7**

M0->Y:\$C080,0 ; Digital Output M00  
 M1->Y:\$C080,1 ; Digital Output M01  
 M2->Y:\$C080,2 ; Digital Output M02  
 M3->Y:\$C080,3 ; Digital Output M03  
 M4->Y:\$C080,4 ; Digital Output M04  
 M5->Y:\$C080,5 ; Digital Output M05  
 M6->Y:\$C080,6 ; Digital Output M06  
 M7->Y:\$C080,7 ; Digital Output M07  
 M8->Y:\$C080,8 ; Digital Input MI0  
 M9->Y:\$C080,9 ; Digital Input MI1  
 M10->Y:\$C080,10 ; Digital Input MI2  
 M11->Y:\$C080,11 ; Digital Input MI3  
 M12->Y:\$C080,12 ; Digital Input MI4  
 M13->Y:\$C080,13 ; Digital Input MI5  
 M14->Y:\$C080,14 ; Digital Input MI6  
 M15->Y:\$C080,15 ; Digital Input MI7  
 M32->X:\$C080,0,8 ; Direction Control (1=output, 0 = input)  
 M34->X:\$C080,8,8 ; Direction Control (1=output, 0 = input)  
 M40->X:\$C084,0,24 ; Inversion control (0 = 0V, 1 = 5V)  
 M42->Y:\$C084,0,24 ; J7 port data type control (1 = I/O)

例:

OPEN PLC1 CLEAR

M32=\$FF ;BITS 0-8 are assigned as output  
 M34=\$0 ;BITS 9-16 are assigned as input  
 M40=\$FF00 ;Define inputs and outputs voltages  
 M42=\$FFFF ;All lines are I/O type  
 DIS PLC1 ;Disable PLC1

CLOSE

### **PMAC2A—PC104 ACC-2P -J7**

M0->Y:\$C0C0,0 ; Digital Output M00  
 M1->Y:\$C0C0,1 ; Digital Output M01

M2->Y:\$C0C0,2 ; Digital Output M02  
 M3->Y:\$C0C0,3 ; Digital Output M03  
 M4->Y:\$C0C0,4 ; Digital Output M04  
 M5->Y:\$C0C0,5 ; Digital Output M05  
 M6->Y:\$C0C0,6 ; Digital Output M06  
 M7->Y:\$C0C0,7 ; Digital Output M07  
 M8->Y:\$C0C0,8 ; Digital Input MI0  
 M9->Y:\$C0C0,9 ; Digital Input MI1  
 M10->Y:\$C0C0,10 ; Digital Input MI2  
 M11->Y:\$C0C0,11 ; Digital Input MI3  
 M12->Y:\$C0C0,12 ; Digital Input MI4  
 M13->Y:\$C0C0,13 ; Digital Input MI5  
 M14->Y:\$C0C0,14 ; Digital Input MI6  
 M15->Y:\$C0C0,15 ; Digital Input MI7  
 M32->X:\$C0C0,0,8 ; Direction Control (1=output, 0 = input)  
 M34->X:\$C0C0,8,8 ; Direction Control (1=output, 0 = input)  
 M40->X:\$C0C4,0,24 ; Inversion control (0 = 0V, 1 = 5V)  
 M42->Y:\$C0C4,0,24 ; I/O port data type control (1 = I/O)

例:

OPEN PLC1 CLEAR

M32=\$FF	;BITS 0-8 are assigned as output
M34=\$0	;BITS 9-16 are assigned as input
M40=\$FFFF	;Define inputs and outputs voltages
M42=\$FFFF	;All lines are I/O type
DIS PLC1	;Disable PLC1

CLOSE

### **PMAC2A-PC104-ACC-1P-J2**

M40->Y:\$C082,8,1 ; SEL0 Output  
 M41->Y:\$C082,9,1 ; SEL1 Output  
 M42->Y:\$C082,10,1 ; SEL2 Output  
 M43->Y:\$C082,11,1 ; SEL3 Output  
 M44->Y:\$C082,12,1 ; SEL4 Output  
 M45->Y:\$C082,13,1 ; SEL5 Output  
 M46->Y:\$C082,14,1 ; SEL6 Output  
 M47->Y:\$C082,15,1 ; SEL7 Output  
 M48->Y:\$C082,8,8,U ; SEL0-7 Outputs treated as a byte  
 M50->Y:\$C082,0,1 ; DAT0 Input  
 M51->Y:\$C082,1,1 ; DAT1 Input  
 M52->Y:\$C082,2,1 ; DAT2 Input  
 M53->Y:\$C082,3,1 ; DAT3 Input  
 M54->Y:\$C082,4,1 ; DAT4 Input  
 M55->Y:\$C082,5,1 ; DAT5 Input  
 M56->Y:\$C082,6,1 ; DAT6 Input  
 M57->Y:\$C082,7,1 ; DAT7 Input

M58->Y:\$C0C2,0,8,U ; DAT0-7 Inputs treated as a byte

### PMAC2A-PC104-ACC-2P-J2

M40->Y:\$C0C2,8,1 ; SEL0 I/O Line

M41->Y:\$C0C2,9,1 ; SEL1 I/O Line

M42->Y:\$C0C2,10,1 ; SEL2 I/O Line

M43->Y:\$C0C2,11,1 ; SEL3 I/O Line

M44->Y:\$C0C2,12,1 ; SEL4 I/O Line

M45->Y:\$C0C2,13,1 ; SEL5 I/O Line

M46->Y:\$C0C2,14,1 ; SEL6 I/O Line

M47->Y:\$C0C2,15,1 ; SEL7 I/O Line

M48->Y:\$C0C2,8,8,U ; SEL0-7 I/O Lines treated as a byte

M50->Y:\$C0C2,0,1 ; DAT0 I/O Line

M51->Y:\$C0C2,1,1 ; DAT1 I/O Line

M52->Y:\$C0C2,2,1 ; DAT2 I/O Line

M53->Y:\$C0C2,3,1 ; DAT3 I/O Line

M54->Y:\$C0C2,4,1 ; DAT4 I/O Line

M55->Y:\$C0C2,5,1 ; DAT5 I/O Line

M56->Y:\$C0C2,6,1 ; DAT6 I/O Line

M57->Y:\$C0C2,7,1 ; DAT7 I/O Line

M58->Y:\$C0C2,0,8,U ; DAT0-7 I/O Lines treated as a byte

**注意：** 当使用PLC功能的时候，注意I5的值必须等于2或者3。

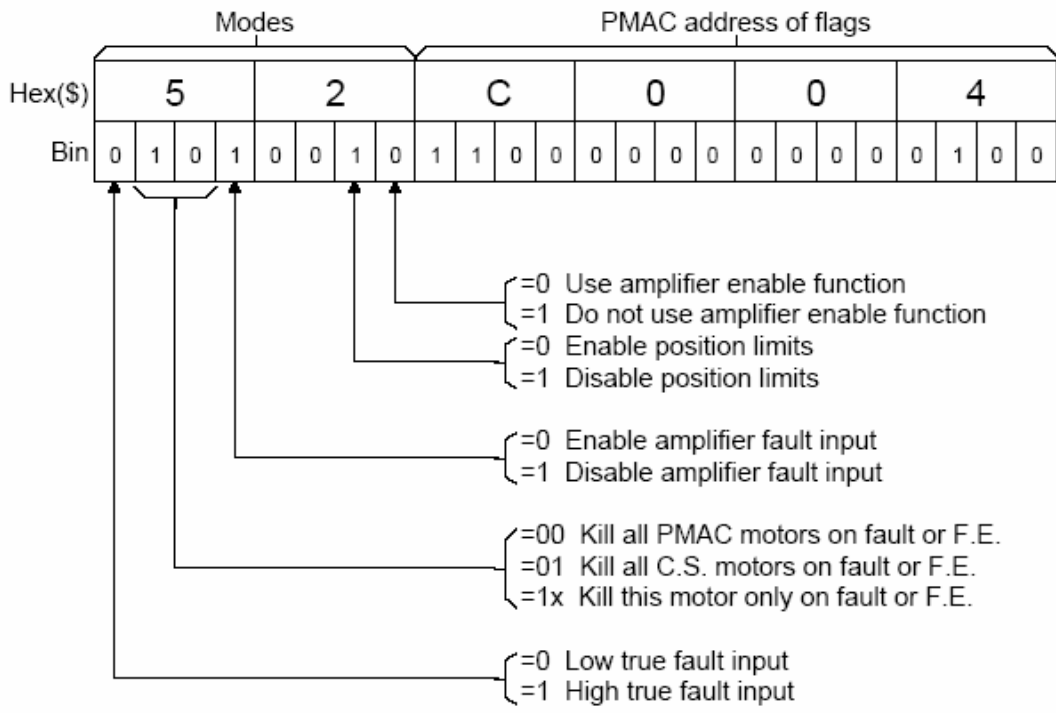
## 3.7.8 常用设定

当接线正确时，我们就可以设定电机，调整电机的特性了：

### 1. 常用参数

	常用 I 变量	范围	默认值	单位
Ix00	电机激活	0...1	0	
Ix11	电机跟随误差极限	0...8388607	32000	1/16 Count
Ix12	电机跟随误差报警	0...8388607	16000	1/16 Count
Ix13	电机正向软限位	$\pm 2^{47}$	0	编码器计数
Ix14	电机负向软限位	$\pm 2^{47}$	0	编码器计数
Ix15	电机异常停止减速	浮点数	0.25	Count/ms <sup>2</sup>
Ix16	电机最大编程速度	浮点数	32	Count/ms
Ix17	电机最大编程加速度	浮点数	0.015625	Count/ms <sup>2</sup>
Ix19	最大手动加速度	浮点数	0.015625	Count/ms <sup>2</sup>
Ix22	电机手动速度	浮点数	32	Count/ms
Ix25	标志信号地址	可用的地址	参看软件手册	

**注意：** x 代值电机，例：#1 电机，参数则为 i100



上面是 Ix25 得用法。

2. 设定参数：必须设定 Ix00=1，这样电机可控制，

**注意：**在初次调试时，如果没有限位开关，可以用IX25屏蔽限位功能，例  
I125=I125+\$520000。

**注意：**当PMAC 2型卡或者PMAC 2A-PC104采用脉冲加方向输出的时候，  
Ix02=Ix02+\$2  
I9n6=3  
也是必须更改的参数。

**注意：**当PMAC 2A-PC104型卡采用模拟量输出的时候，  
I900 = 1001                      I901 = 2  
I902 = 3                          I903 = 1746  
I906 = 1001                      I907 = 1746  
I9n6 = 0                          Ix69 = 1001  
I10 = 1710933  
也是必须更改的参数。

3. 调整硬件反馈：

验证反馈形式的方法，是采用开环指令，O。例：O10，表示输出最大电压或者频率的10%。通常我们在 I 型卡下使用 O10，而 2 型卡时使用 O1。这时，我们观察，如果电机反

馈计数向负方向增加，那么就表示，当前反馈方向不正确，需要把参数改动，例如，当 I900=7 时，我就可以把它改为 3。如果电机反馈计数向正方向增加，那么正确，不需要改动。

**注意：**使用O指令时，我们可以使用K指令，这时将没有输出。

通过下面硬件参数，对于反馈的编码器信号进行倍频

PMAC 1 型卡							
1 号反馈	2 号反馈	3 号反馈	4 号反馈	5 号反馈	6 号反馈	7 号反馈	8 号反馈
I900	I905	I910	I915	I920	I9250	I930	I935
9 号反馈	10 号反馈	11 号反馈	12 号反馈	13 号反馈	14 号反馈	15 号反馈	16 号反馈
I940	I945	I950	I955	I960	I965	I970	I975
PMAC 2 型卡							
电机#1		电机#2		电机#3		电机#4	
I910		I920		I930		I940	
电机#5		电机#6		电机#7		电机#8	
I950		I960		I970		I980	

取值：

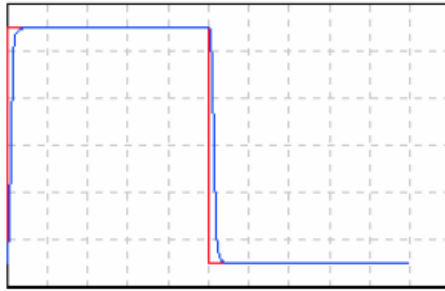
0	脉冲+方向反馈 CW
1	X1 正交编码器译码 CW
2	X2 正交编码器译码 CW
3	X4 正交编码器译码 CW
4	脉冲+方向反馈 CCW
5	X1 正交编码器译码 CCW
6	X2 正交编码器译码 CCW
7	X4 正交编码器译码 CCW

当使用模拟量指令的时候，有可能出现零漂得现象，我们可以使用 Ix29 来调整零漂。

4. 调整 PID

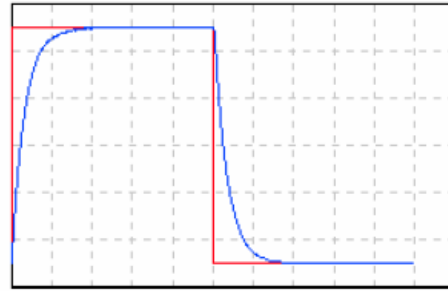
电机的特性互有不同，我们使用 PMAC Tuning Pro 就可以调整电机特性，为我们服务。





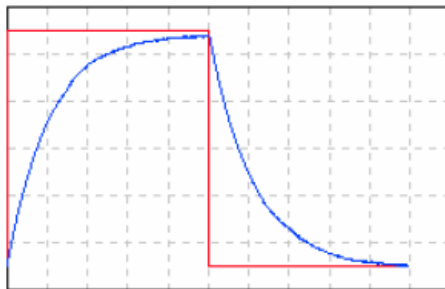
Ideal Case

The motor closely follows the commanded position



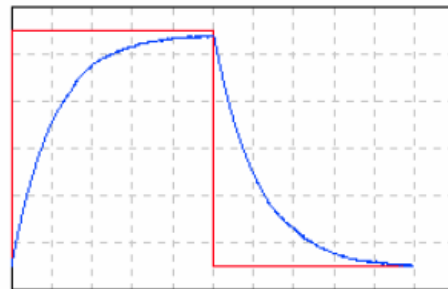
Position Offset

Cause: friction or constant force / system limitation  
Fix: Increase  $K_I$  (Ix33) and maybe use more  $K_F$  (Ix30)



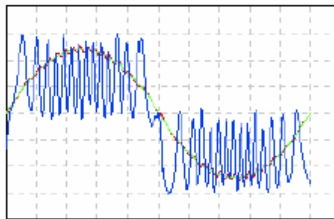
Sluggish Response

Cause: Too much damping or too little proportional gain  
Fix: Increase  $K_P$  (Ix30) or decrease  $K_D$  (Ix31)



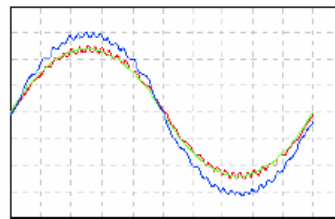
Overshoot and Oscillation

Cause: Too little damping or too much proportional gain  
Fix: Decrease  $K_P$  (Ix30) or increase  $K_D$  (Ix31)



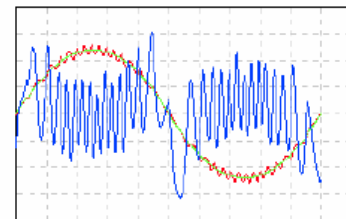
Ideal Case

The following error is reduced at minimum and is concentrated in the center, evenly along the move



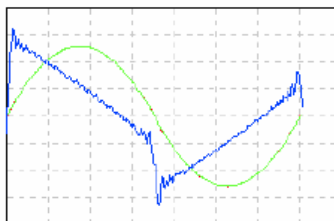
High vel \ FE correlation

Cause: damping  
Fix: Increase  $K_{vel}$  (Ix32)



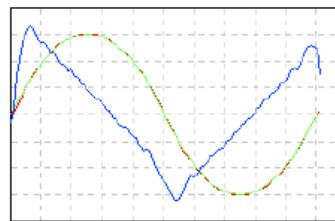
High vel \ FE correlation

Cause: friction  
Fix: Increase Integral gain (Ix33) or Friction Feedforward (Ix68)



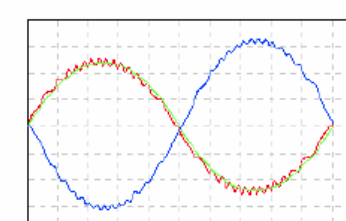
High acc \ FE correlation

Cause: Integral lag  
Fix: Increase  $K_{aff}$  (Ix35)



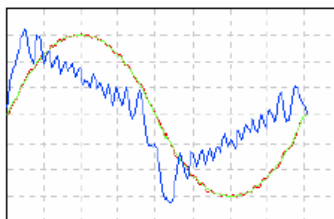
High acc \ FE correlation

Cause: Physical system limitations  
Fix: Use less sudden acceleration



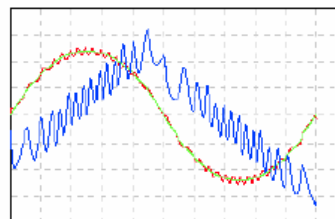
Negative vel \ FE correlation

Cause: Too much velocity FF  
Fix: Decrease  $K_{vel}$  (Ix32)



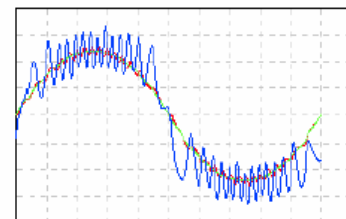
High vel \ FE correlation

Cause: damping and friction  
Fix: Increase  $K_{vel}$  (Ix32)



High acc \ FE correlation

Cause: Too much acc FF  
Fix: Decrease  $K_{aff}$  (Ix35)



High vel \ FE and acc \ FE correlation

Cause: Integral lag and friction  
Fix: Increase  $K_{aff}$  (Ix35)

## 5. 保存参数并手动

调整特性良好的电机，就可以响应我们的手动指令了，但不要忘记保存参数：

J+	正向连续运动
J-	反向连续运动
J/	运动停止(或者闭环)
J=	返回前一个 JOG 的(或上一个编程)位置
J={常数}	Jog 到指定的位置(单位是脉冲 counts)
J^	从当前实际位置运动指定的距离
J:{常数}	从当前指令位置运动指定的距离
J=*	Jog 运动一个变量(PMAC 的内存地址)

## 第四章      **PMAC 指令与应用**

使用PMAC非常简单；具有以下特征：

- 拥有一个机动灵活的的中断处理方案，允许每件工作，例如：运动程序和PLC，可以彼此独立运行。
- 使用指针型M变量，监视不同PMAC内存寄存器，例如：运动程序的状态, PLC的状态或主机状态。
- 查询可以实时得到响应。
- 八个轴可以同步、查补、单独运动，或组合运动。
- 数据采集，可以得到为了分析的数据的详细报告或图形，例如：运动轨迹，速度描绘和任何变量的组。

## 第一节 在线指令

许多送给PMAC的指令都是在线指令；在线指令可以立刻被PMAC运行，而引起一些改变，或把一些数据返回。像**P1=1**这样的指令，如果没有打开运动程序缓冲区，可以立刻被运行。如果打开运动程序缓冲区，就会被储存在缓冲区中。 像 **X1000 Y1000**这样的指令，不是在线指令。如果没有打开的缓冲区，这些指令将会被PMAC(如果I6被设定成1或3, PMAC返回ERR005) 拒绝执行。其他的在线指令，像是**J+**，不能放到程序缓冲区中去。(举例来说，除非以 **CMD "J+"**的形式)

在线指令有三个基本类型：

- 定义电机指令：只影响当前被选址的电机；例：**J+**、**J-**
- 定义坐标系指令：只影响当前被选址的坐标系；例：**R**、**A**
- 全局指令：不论选址如何，都影响卡的特性；例：**P1=1**

### 1. 电机指令

电机是通过**#n**的命令来选择的，其中**n**是电机的序号，范围是1到8。PMAC始终将指令送给当前的电机，直到在一个**#n**指令改变为止。

例如：**#1J-#2J+** 就是，电机#1反转，同时电机#2正转。

相同的命令主要还有：全部手动运动命令，回零命令，开环命令，报告电机位置、速度、跟随误差的命令。

### 2. 坐标系指令

坐标系是通过**&n**的命令来选择的，其中**n**是坐标系的序号，范围是1到8。PMAC始终将指令送给当前的电机，直到在一个**&n**指令改变为止。

例如：**&1B6R&2B8R** 就是，坐标系1执行6号程序，同时坐标系2执行8号程序。

相同的命令主要还有：全部程序运行命令，轴定义指令，一般缓冲区指令。

### 3. 全局指令

不论选址如何，都影响卡的特性的指令就是全局指令，例如 **i100=1** 等。还有**<Ctrl+A>**等这样的指令。

## 第二节 缓冲区（编程）指令

缓冲区指令不被立刻执行。PMAC有许多预计好的缓冲区：256个一般的运动程序缓冲区，八个旋转运动程序缓冲区（每一个对应一个坐标系），32个PLC缓冲区。在写入指令之前，那一个缓冲区一定要被打开。(举例来说**OPEN PROG 3, OPEN PLC 7**) 每个指令都写在缓冲区的结尾；如果为了代替现有的缓冲区内容,在写入**CLEAR**指令之后，之前的内容就会被清除。在结尾处,使用**CLOSE**指令关闭打开的缓冲区。

### 第三节 特色

#### 4.3.1 I 变量

I-变量(设定初值, 或装备变量值) 决定卡片的工作特性。 这些变量的值被保存在内存里的固定位置, 并有预定好的意义。共有1024个I变量,从I0到I1023,如下所示:

I00—I79	为 PMAC 卡的全局变量
I80—I99	连接旋转变压器的设置
I100—I186	为#1 电机的设置变量
I187—I199	为&1 坐标系的设置变量
I200—I286	为#2 电机的设置变量
I287—I299	为&2 坐标系的设置变量
.....	.....
I800—I886	为#8 电机的设置变量
I887—I899	为&8 坐标系的设置变量
I900—I979 (PMAC 1)	1—16 编码器的设置
I980—I1023 (PMAC 1)	保留
I900—I909 (PMAC 2)	门阵列 IC 设置
I910—I989 (PMAC 2)	第 1—8 通道硬件设定
I990—I1023 (PMAC 2)	辅助通道与 MACRO 设定

使用方法例子:

I120 = 45

I120 = (I120+P25\*3)

I 变量使用的时候, 会有范围限制, 如果超出范围, 不会导致出错。通过取模操作, 该值会自动转换至范围以内。

更改的 I 变量, 必须通过 save 的命令保存, 否则再失电后就会丢失。

I 变量都有自己的缺省值, 可以通过 I 变量=\*的指令, 使 I 变量恢复为出厂缺省值, 如果必要, 还可以通过\$\$\$\*\*\*或硬件跳线来恢复。

#### 4.3.2 P 变量

P 变量使全局用户变量, 用于 PMAC 编程中的计算, 它们在内存中有固定的位置, 为 48 位浮点变量, 没有预先定义的用途。共有 1024 个, 从 P0 到 P1023。

例: P342=100            P100=P101\*(sin(45))2            IF( M1!= 1 AND P10 = 0)

#### 4.3.3 Q 变量

Q-变量也是用户全局变量, 可用于 PMAC 编程中的计算。为 48 位浮点变量, 共有 1024 个, Q0~Q1023。每个 Q 变量与使用它的坐标系有关, 同一个 Q 变量在不同的坐标系中占用不同的地址, 而在 PMAC 的同一个地址, 在不同的坐标系对应不同的 Q 变量。Q 变量的特

殊用途：read 命令可以将读入的值前面加上字母 A 到 Z,分别放到 Q101 到 Q126 中。运动程序中的 S 表达式（主轴）将跟在后面的值放在 Q127 变量中。

### 4.3.4 M 变量

M-变量用于访问 PMAC 内存地址和 I/O 点地址。M-变量没有预先定义的含义。用户必须通过定义 M 变量的地址来访问 PMAC 的地址。M-变量一旦定义好后，可用于计算和判别触发。

M 变量称为 PMAC 卡的地址指针变量。变量范围：M0 到 M1023。

M 变量定义格式为：M\*\* 地址，偏置地址，宽度，方向

M 变量定义的地址前缀，可以是下列类型：

X: X 内存中的 1-24 位的固定地址位

Y: Y 内存中的 1-24 位的固定地址位

D: 同时占用 X 和 Y 内存的 48 位固定地址位

L: 同时占用 X 和 Y 内存的 48 位浮点地址位

DP: 32 位的固定地址位(双端口 RAM 使用)

F: 32 位的固定地址位(双端口 RAM 使用)

TWS: 多路 I/O 开关码

M 变量的功能为存取 PMAC 的内存和 I/O 点，M 变量一旦定义，通过卡的后备电池或闪烁存储器可以被保存下来。M 变量可以是 1 位的，也可以是一个字节的（8 位）或者是一个 24 位的字，48 位的浮点双字。

例：

M1->Y:\$FFC2,8,1

M102->Y:49155,8,16,S

M103->X:\$C003,0,24,S

M161->D:\$002B

M191->L:\$0822

M50->DP:\$D201

M51->F:\$D7FF

M100->TWD:4,0.8.3,U

### 4.3.5 队列处理

可以通过P变量的特性，与M变量的功能组合，形成高效的队列处理。

例：队列读取：

P1=10

P3=P(P1)

**队列写入：将31到40分配到P1至P10**

M34->L:\$1001；这是P1的地址

M35->Y:\$BC22,0,16；这是M34定义的地址

OPEN PLC 15 CLEAR

P100=31

WHILE (P100!>40)；

```

M34=P100 ;
P100=P100+1 ;
M35=M35+1 ;
ENDWHILE
DISABLEPLC15 ;
CLOSE
ena PLC15 ;

```

### 4.3.6 运算方法

PMAC 支持计算机语言里的很多运算符，用很强的计算能力：

- 算术运算符：+, -, \*, /;
- 取模运算：%;
- 逻辑运算：& (bit-by-bit AND), | (bit-by-bit OR), ^ (bit-by-bit EXCLUSIVE OR)。

### 4.3.7 功能简介

PMAC 支持计算机语言里的很多功能函数，用很强的计算能力：

**SIN, COS, TAN, ASIN, ACOS, ATAN, ATAN2, LN, EXP, SQRT, ABS, INT**

### 4.3.8 比较功能

= (equal to)  
 != (not equal to)  
 > (greater than)  
 !> (not greater than; less than or equal to)  
 < (less than)  
 !< (not less than; greater than or equal to)  
 ~ (approximately equal to -- within one)  
 !~ (not approximately equal to -- at least one apart)

### 4.3.9 用户自定义伺服算法

PMAC 允许客户针对特殊的（例：动态特性很差）的系统，编制自己的 PID 伺服算法。

**注意：**没有经验的用户不要使用这一功能，如果要这么做，用户必须熟知伺服理论，以及汇编知识。



## 第四节 内存地址表

Range	X-Memory	Y-Memory	Type
\$0000 - \$00FF	运算寄存器	运算寄存器	内置 DSP 内存
\$0100 - \$17FF	运算寄存器	运算寄存器	扩展 RAM
\$1800 - \$BBFF	用户缓冲区	用户缓冲区	扩展 RAM
\$BC00 - \$BFFF	用户自定义伺服算法	M 变量定义	扩展 RAM
\$C000 - \$C03F			DSP 门阵列寄存器
\$D000 - \$DFFF	Bits 0 to 15	Bits 0 to 15	双端口 RAM
\$E000 - \$F000	VME 设定寄存器	MailBox 寄存器	VME 总线
\$F000 - \$FFFF	N/A	N/A	I/O 寄存器

## 第五节 程序缓冲区

- 256 个运动程序存储能力。所有的程序在打开编辑时，必须停下来。
- 所有的程序在运行前必须停下来。
- 一个 PLC 可在在其他 PLC 运行的时候打开编辑。

## 第六节 编码器转换表

PMAC 利用编码器转换表处理输入的脉冲信号，这样我们输入的正交编码器信号就可以被 DSP 处理了。但是工业中反馈信号不是单一的编码器形式。当我们使用不同的反馈元件，我们就可以通过编码器转化表来处理。其他还支持得反馈元件形式主要有：

- 并行二进制编码器或格雷码反馈；
- 磁致伸缩传感器；
- 模拟量反馈；
- 旋转变压器反馈；
- 1Vpp 细分信号反馈。

## 第七节 PMAC 位置寄存器

举例说明：

**M162->D:\$002B ; #1 Actual position (1/[Ix08\*32] cts)**

**M164->D:\$0813 ; #1 Position bias (1/[Ix08\*32] cts)**

**M167->D:\$002D ; #1 Present master ((handwheel) pos (1/[Ix07\*32] cts)**

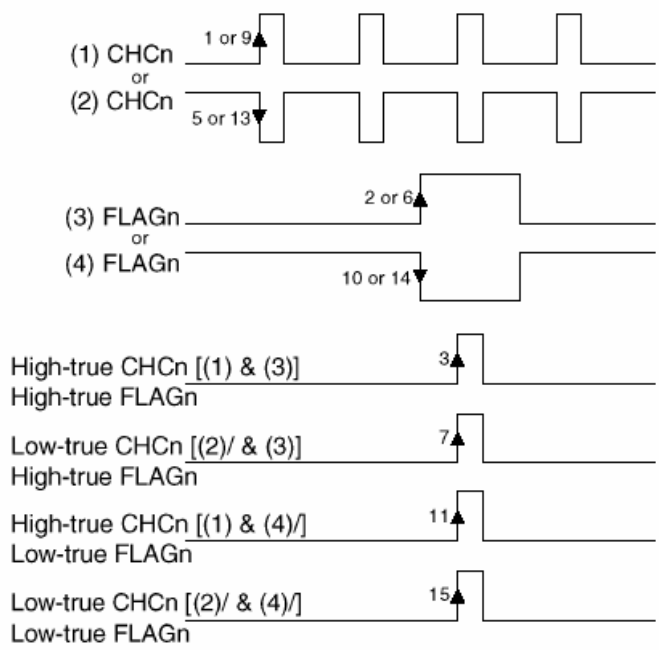
针对不同的板卡，地址略有不同，详见附页。

## 第八节 回零运动

如果PMAC不使用绝对型位置反馈，那么上电/重置之后一般都需要回零运动。常见的

回零运动需要的变量是：

Ix20	电机x手动/回零加速度时间
Ix21	电机x手动/回零S加速度时间
Ix23	电机x回零速度与方向
Ix26	电机回零偏置
I902, I907...	编码器回零捕捉控制(PMAC 1)
I903, I908...	编码器回零捕捉信号选择(PMAC 1)
I9n2...	编码器回零捕捉控制(PMAC 2)
I9n3...	编码器回零捕捉信号选择(PMAC 2)



回零运动的命令是HM，调用例子：

HOME1..8 ;在运动程序中  
#1HM ;在线命令  
CMD"#1HM" ;在PLC程序中

HMZ也是回零命令，但它不会产生运动，只是将当前位置计数清为零

## 第九节 Command、Send 等增强指令

使用指令**COMMCD**或**CMD**,可以在PLC或者运动程序里面使用在线指令。如果I62=0, PMAC将自动在消息的结尾加上传输的结束字符<CR>; 如果I62=1, PMAC将不在消息的结尾加上传输的结束字符<CR>, 这时通过SEND^M发送<CR>。

使用指令**SEND**,可以在PLC或者运动程序里面向主机或者终端发送字符串：如果I62=0, PMAC将自动在消息的结尾加上传输的结束字符<CR>; 如果I62=1, PMAC将不在消息的结尾加上传输的结束字符<CR>, 这时通过SEND^M发送<CR>。

## 第五章 电机编程

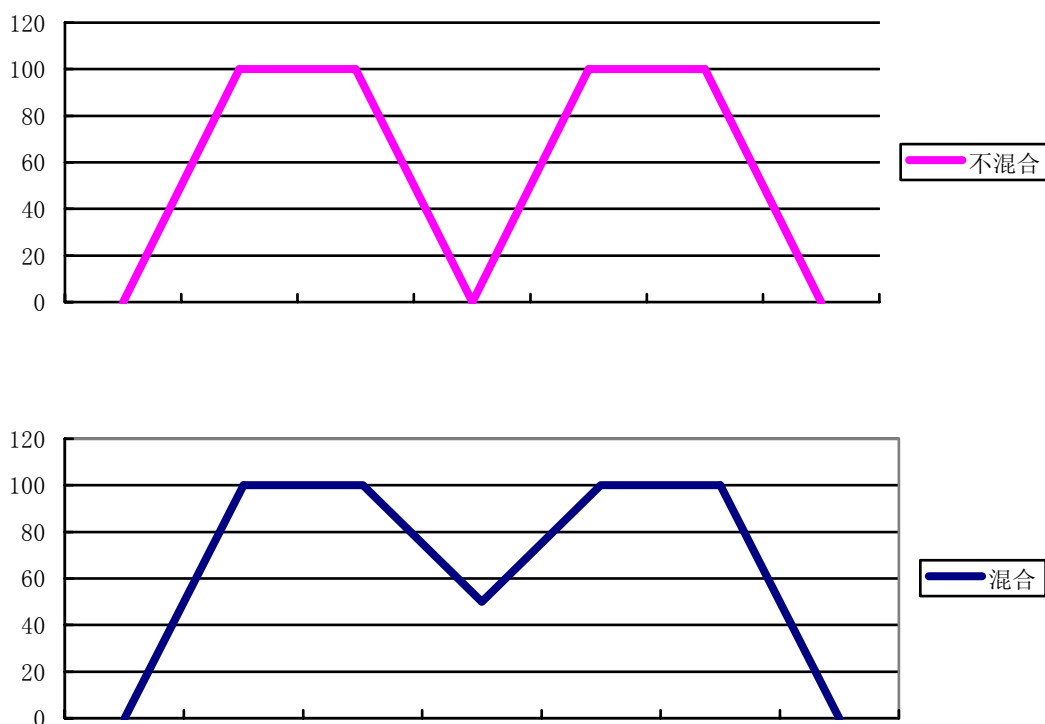
PMAC可以存储256个运动程序。

- PMAC在任意时刻都可以执行运动程序，并执行运动所需的所有计算；
- 运动程序必须在坐标系下运行；
- 一个程序可以在不同的坐标系下运行；
- 一个程序可以在不同的坐标系下同时运行；
- 一个坐标系一段时间只能运行一个运动程序；
- PMAC最多能够同时执行与卡上坐标系同样多的运动程序；
- 一个运动程序能够将其他的运动程序调用做子程序。

PMAC卡运动程序语言拥有与BASIC或者PASCAL的简明语法，同时PMAC卡还可以支持G代码（RS-274）这样的机器语言。PMAC还允许运动程序与PLC程序相互调用。

## 第一节 PMAC 运动程序

在 PMAC 卡的默认情况下，最容易完成的是直线混合运动。在这类运动中，一个轴以指定的速度运动到指定位置（还可以指定加速度和减速度）。如果这样的运动不止一个，那么，第一个运动将混合到第二个运动中去。



**注意：**可以通过**Ix92=1**可以关闭混合运动，但这样做会影响运动程序的执行效率。默认情况下，需要不混合定位时，可以使用**Dwell 0**来使电机在位准停。

线性混合运动模式是给运动程序的默认模式。如果需要别的模式，可以使用模态指令（例：**CIRCLE1**等）随时改变。在程序中使用**LINEAR**模态指令是一个好习惯。

## 第二节 笛卡儿坐标系

PMAC 的坐标系指的是一个协调在一起的电机组。一个坐标系（当最少有一个电机时）可以执行运动程序，而一个电机不行。PMAC 可拥有 8 个坐标系，分别是&1 至&8。在坐标系下电机的组织形式是任意的。

一般情况下，我们把需要联动、插补等协调运动的电机放在一个坐标系下。如果你希望它们都独立运动，最好它们分别放在不同的坐标系下，也可以通过这种灵活的机制，组合一些特殊的用法，完成一些高级应用（例如：时基控制…）。

### 5.2.1 轴的定义

轴是坐标系的元素。它类似于电机，但不是同一事物。一个坐标系可以同过下列字母定义轴：X, Y, Z, A, B, C, U, V, W。通过比例参数（浮点数）和偏置，周的定义就可以灵活的与电机组组合。

- X, Y, Z: 传统上的主要直线轴，可以实现：矩阵轴的定义、矩阵轴的转换、圆弧插补、切削半径补偿。
- U, V, W: 传统上的第二直线轴，可以实现：矩阵轴的定义。
- A, B, C: 传统的旋转轴（A 围绕 X, B 围绕 Y, C 围绕 Z），可以实现：位置翻转。

通常情况下，电机的轴定义是一对一的。但是这并非唯一用法，我们也可以在一个坐标系下定义不同的电机为相同的轴等。遵循原则如下：

&1 #1->X #2->Y	&2 #4->20X #6->25.4Y	标准定义法
&1 #1->X	&2 #2->X	该定义是允许的，两个电机在不同的坐标系下定义为 X 轴。
&1 #1->X #2->X		该定义是允许的，两个电机将同样作 X-轴轨迹，比如龙门机床。
&1 #1->X #1->Y		该定义不允许，一个电机不能在一个程序中执行不同的运动轨迹，第一次的轴定义将被第二次的轴定义代替。
&1 #1->X	&2 #1->X	该定义不允许，当运行两个程序时，一个电机将接收不一致的指令，第二个坐标系的定义将被拒绝。

### 5.2.2 轴定义的扩展与描述

- 每个坐标系可以定义速率轴“FEEDRATE”，这样任意轴可以是其它坐标系下的时间轴。例如：时基控制。
- 电机在坐标系下有“轴定义表达式”，使轴与电机匹配，并且具有放大和平移的功能：例：#1 -> 10000X + 5000

- 跟多的放大和平移功能，使 PMAC 卡具有定义轴定义矩阵的功能：

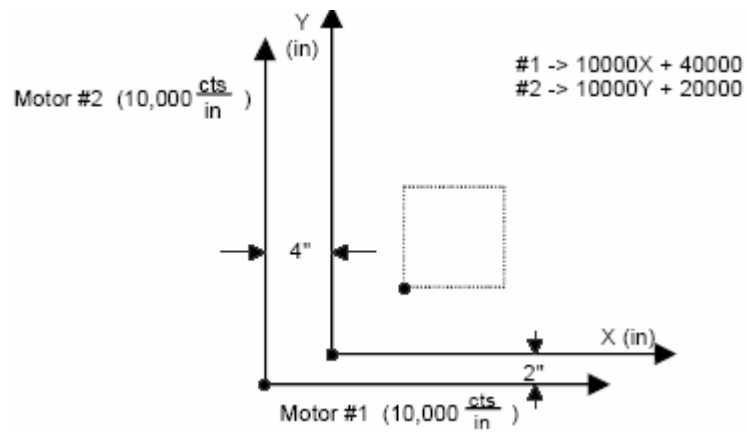
$$\#1 \rightarrow A11X + A12Y + A13Z + B1$$

$$\#2 \rightarrow A21X + A22Y + A23Z + B2$$

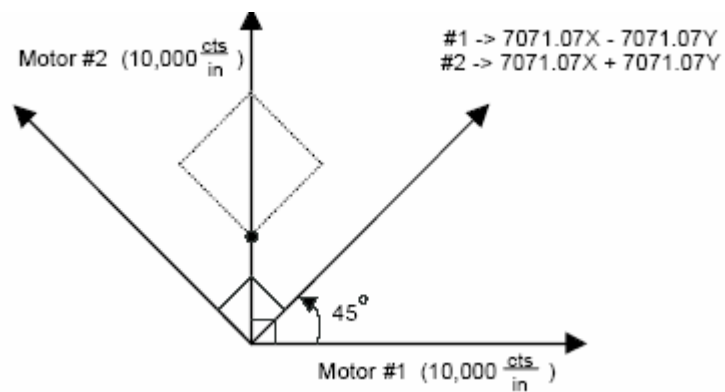
$$\#3 \rightarrow A31X + A32Y + A33Z + B3$$

这里 A11 到 A33 是比例因子，而 B1 到 B3 是偏移量，此定义形式可以实现对坐标系的：平移、放大、旋转、校正。

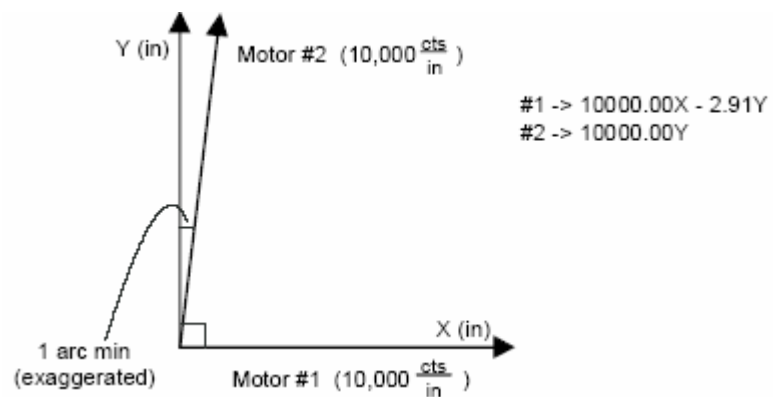
缩放和平移：



缩放和旋转：



校正：



### 第三节 编写运动程序

1. 打开运动程序缓冲区，使用指令**OPEN PROG {constant}**，其中{constant}是程序的编号，1到32767（其中1000号以后的程序有特殊用法和加密功能，建议使用1到999）。

2. 运动程序**Prog1000, Prog1001, Prog1002, Prog1003**用来做为标准G代码（RS274, 包括**G, M, T, D**等工具码）的解释，也可用作一般程序。

3. 在PMAC的内存空间允许的情况下，PMAC在板卡上可以保存256个运动程序。其中，特殊的运动程序是**Prog0**，它被保留给旋转缓冲区。旋转缓冲区的特点是，在程序运行的时候，可以向程序添加执行的程序行，PMAC自动将添加的内容放在程协的尾部。

4. 命令**CLEAR**可以清空打开的运动程序缓冲区，PLC程序缓冲区，旋转缓冲区。

5. PMAC运动程序的插补模式是由模态指令来改变的，模态指令包括：**LINEAR, RAPID, CIRCLE, PVT, SPLINE**。

6. 运动程序的运动方法可由指令**INC, ABS**改变，还可以指定运动时的时间量(**TA, TS, TM**)/速度(**F**)，这些指令与模态指令混合使用，可以在程序行开始，程序行结尾等任意时刻改变运动状态。

7. 运动程序行语句如下：

**X1000**

**Y100Z100**

写在同一行的指令，会被坐标系自动混合在一起处理。

8. 如果**TA, TS, TM, F**等指令在程序中没有指定，PMAC也会使用**Ix87, Ix88, Ix89**默认的值运行程序。来1000, 1001, 1002, 1003用来做为标准G代码（RS274, 包括**G, M, T, D**等工具码）的解释，也可用作一般程序。

**注意：**不要依靠这些参数，尽可能在程序里指定参数。使用默认参数，仅可以在一般调试情况下，加快我们的调试效率。

9. 在运动程序里，我们还可以使用循环语句（**WHILE**），条件语句（**IF..ELSE**），跳转语句（**GOTO**），子程序语句（**GOSUB**），互相调用语句（**CALL**）等强化功能。

**注意：**不**GOSUBs** 或者 **CALLs** 这样的调用，最多只可用15层。

10. 使用**CLOSE**命令就可以结束运动程序缓冲区，PLC程序缓冲区，旋转缓冲区的编辑，进入在线指令的环境。通过**RETURN**指令还作为程序的返回点。

例子：

```
close           ; 关闭任意的缓冲区
delete gather   ; 删除数据采集缓冲区
undefine all    ; 删除全部轴定义
#1->2000X       ; 电机#1的轴定义
OPEN PROG 1 CLEAR ; 打开缓冲区
LINEAR         ; 直线插补模式
INC            ; 增量模式
TA100          ; 直线加速度100 msec
TS0            ; 不使用S-curve曲线加速
F50            ; 速度是50个单位每Ix90 msec
```

**X1** ; 运行一个单位  
**CLOSE** ; 关闭缓冲区



## 第四节 执行运动程序

1. 必须先选择一个坐标系，使用指令**&n**，其中n代表1到8。例：**&1** 表示选择1号坐标系。

2. 选择一个运动程序使用指令**B{constant}**，其中{constant}代表运动程序的序号。不使用**B**指令更改的话，你运行将永远是上一次用**B**指令选择的程序。

3. 如果希望连续的执行运动程序，使用指令**R（或<CTRL+R>）**，这样运动程序将从头到尾执行，一直到结束程序的指令或者出错。

4. 如果希望单步的执行运动程序，使用指令**S（或<CTRL+S>）**，这样运动程序将局部运行程序。如果出现**BLOCKSTART**则一直运行到**BLOCKSTOP**。

5. 当执行运动程序时，PMAC会自动监测坐标系的状态。如果坐标系出现配制出错，PMAC会向主机发送**<BELL>**字符，如果**I6=1**或者**3**，它将报错：

- 坐标系下有一台电机在行程限位(ERR010)
- 坐标系下有一台电机正在执行运动(ERR011)
- 坐标系下有一台电机没有闭环(ERR012)
- 坐标系下有一台电机没有激活{Ix00=0} (ERR013)
- 坐标系没有电机定义(ERR014)
- 有一个（固定，非旋转的）缓冲区打开(ERR015)
- 程序结构问题或程序不存在(ERR016)
- 在/或\停止命令后，电机离开停止点(ERR017)

6. 在程序运行之前，可以使用命令**CTRL+A**结束PMAC上的其它运动。在程序运行之前，还需要保证电机是闭环的，我们可以使用**#1J^2000**这样的手动命令来验证电机是否可以运动。

7. 坐标系中的电机必须准备就绪，如果出现限位超程，或者电机不能闭环程序，那么程序都不能运行。

8. 任何一个坐标系都有自己的速度倍率（feedrate），可以使用如 **&1%** 的命令检验坐标系的速度倍率，如果速度为0，运动程序将不能正确运行，类似 **&1%100** 的指令，可以将坐标系的速度倍率设为100%。

9. 如果你第一次运行某一运动程序，建议使用 **%10** 命令，这样就可以避免不适当加速度、速度设置可能造成的不良后果。

10. 运动程序可以使用类似 **&1A** 指令停止，也可以使用指令**CTRL+A**停止全部坐标系的程序。

11. 如果运动程序的执行过程中，出现失控的现象，可以使用**CTRL+K**指令，这样就可以除去对电机的控制指令。（如果电机的使能接在PMAC上，还可以禁止电机的使能）。

12. 停止运动程序还有**CTRL+Q**，**CTRL+H**，**Q**，**H**。其中**CTRL+Q**和**Q**是停止程序，当前的程序会执行完当前语句，然后停止。其中**CTRL+H**和**H**是暂停程序，当前的程序会进给保持，当**CTRL+R**和**R**指令来到，程序继续运行。

## 第五节 子程序或者辅助程序

在 PMAC 运动程序中可以建立子程序和子例程。

使用 **GOSUB{constant}** 命令，可以跳转到程序中标志 **N{constant}** 的部分，遇到 **RETURN** 指令返回调用 **GOSUB** 指令的断点。

使用 **CALL{constant}** 命令，可以跳转到程序 **PROG {constant}** 的部分，遇到 **RETURN** 指令返回调用 **CALL** 指令的断点。如果 {constant} 是整数，就从子程序 **PROG** 的程序开始的地方执行。如果 {constant} 是浮点数，就从子程序 **PROG** 的程序行 {constant} 开始的地方执行。

### 5.5.1 子程序/子例程变量交换

通过使用 **READ** 语句，子例程可以调用语句中的变量。在使用过程中，字符 **A** 后的值放在 **Q101**，字符 **B** 后的值放在 **Q102**，一直到字符 **Z** 后的值放在 **Q126**，其中 **N** 和 **O** 不可以使用。这种结构在调用 **G**，**M**，**T**，**D** 等工具码时非常有用。

例子程序:

```
#1->2000X
open prog1 clear
LINEAR INC TA100 TS0 F50 ;Mode and timing parameters
gosub 100 H10             ;Subroutine call passing parameter H with
                           value 10
return                   ;End of the main program section (execution
                           ends)
n100                     ;Subroutines section. First subroutine
                           labeled 100
read(h)                  ;Read the H parameter value passed
IF (Q100 & $80 > 0)       ;If the H parameter has been passed Use the
X(Q108)                  H parameter value contained in Q108
endif
return                   ;End of the subroutine labeled 100
close                    ;End of the motion program code
```

### 5.5.2 G、M、T、D 代码（标准机床代码）

对 **G**，**M**，**T**，**D** 等工具码的解释是以子程序的形式调用的。

例子程序:

```
OPEN PROG 5 CLEAR        ; Prepare motion program 5 for entry
G17 G90                  ; XY plane, absolute move spec
G97 S1800                 ; Set spindle speed of 1800 rpm
F500                     ; Cutting speed 500 mm/min
G00 X10.00 Y5.00         ; Rapid move to (10, 5)
```

---

```

M03                ; Start spindle
G04 P2.0           ; Wait 2 seconds
G01 Z0             ; Lower cutter
X30.25 Y5.00       ; Linear XY move
G03 X35.25 Y10.00 J5 ; CCW arc move
G01 X35.25 Y50.10  ; Linear move
G03 X30.25 Y55.10 I-5 ; CCW arc move
G01 X10.00 Y55.10  ; Linear move
G03 X5.00 Y50.10 J-5 ; CCW arc move
G01 X5.00 Y10.00   ; Linear move
G03 X10.00 Y5.00 I5 ; CCW arc move
G01 Z5 M05         ; Cutter up, stop
G00 X0 Y0          ; Back to home
CLOSE
OPEN PROG 1000 CLEAR ; Prepare buffer 1000 for entry
RAPID RETURN        ; G00 Rapid mode (N0 is implied)
N01000 LINEAR RETURN ; G01 Linear interpolation mode
N02000 CIRCLE1 RETURN ; G02 Clockwise circle mode
N03000 CIRCLE2 RETURN ; G03 Counterclockwise circle mode
N04000 READ(P)       ; G04 Dwell for P seconds
IF (Q100 & 32768 > 0) ; P parameter specified?
DWELL (Q116*1000)    ; PMAC specifies dwell time in msec
ENDIF RETURN
N17000 NORMAL K-1 RET ; G17 Specify XY plane
N18000 NORMAL J-1 RET ; G18 Specify ZX plane
N19000 NORMAL I-1 RET ; G19 Specify YZ plane
N90000 ABS RET        ; G90 Absolute mode
N91000 INC RET        ; G91 Incremental mode
N97000 READ(S)        ; G97 Spindle speed set
IF (Q100 & 262144 > 0) ; S parameter specified?
I422=Q119/30         ; #4 jog speed in cts/msec
ENDIF RETURN
CLOSE
OPEN PROG 1001 CLEAR ; Prepare buffer 1001 for entry
N03000 CMD "#4J+" RET ; Start spindle clockwise (closed loop)
N04000 CMD "#4J-" RET ; Start spindle counterclockwise (ditto)
N05000 CMD "#4J/" RET ; Stop spindle CLOSE To run this program
&1 B5 R              ;Coordinate System 1, point to Beginning of ;Program 5, Ru n

```

第六节 混合运动/加速度模式：

- 下面两个运动效果一致

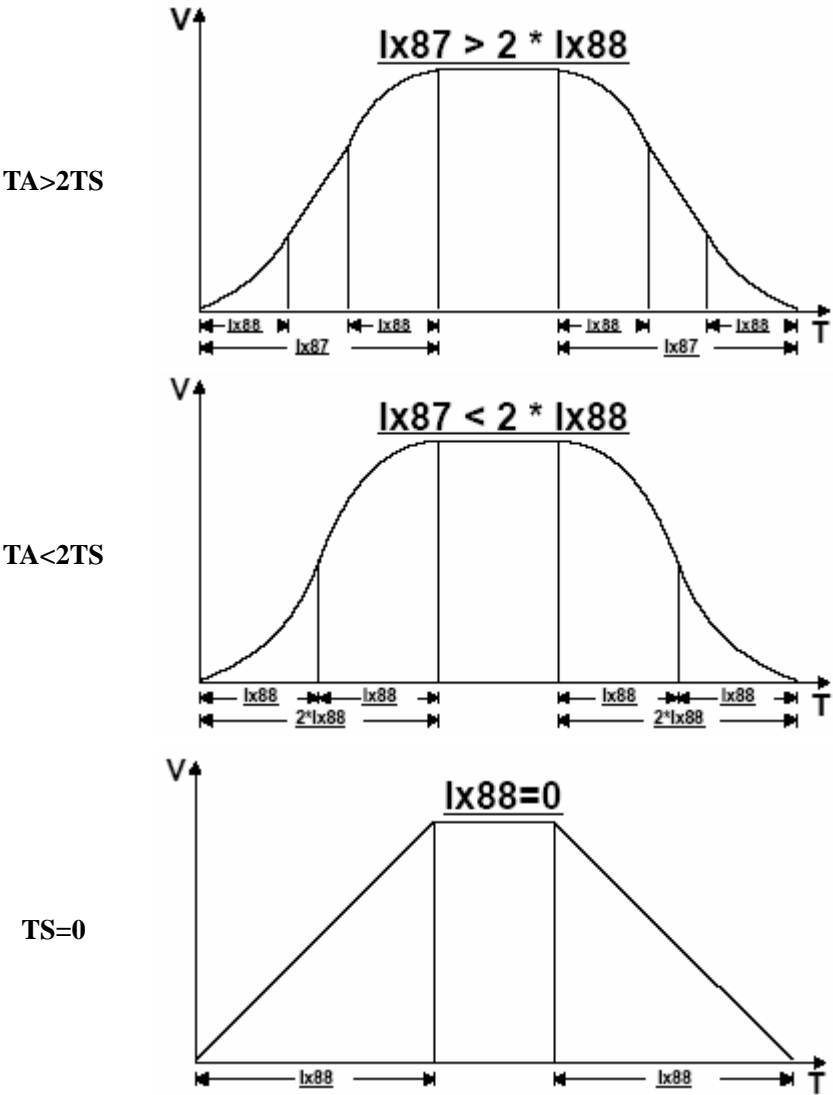
TM100  
X3 Y4

or

FRAX(X,Y)  
X3 Y4 F50

$$TM = \frac{1190 \cdot \sqrt{3^2 + 4^2}}{50} = \frac{5000}{50} = 100 \text{ msec}$$

- 用户在执行运动时，只需要指定全加速时间TA和S曲线加速时间TS。如果TA时间比2倍的TS时间要小，那么TA时间将是2倍的TS时间。
- 在混合运动下，TA 时间将不在满足我们的定义，这时，两个连着的运动，将不能出现准确地 TA。
- PMAC 通过 Ix16 和 Ix17 这两个参数来保证程序执行的安全，所有程序中指定的 TA，TS，F 都不可能控制电机大于这两个参数运行。



下面举例说明混合运动，例子里面默认 S 曲线加速度为 0，这是为了方便说明：

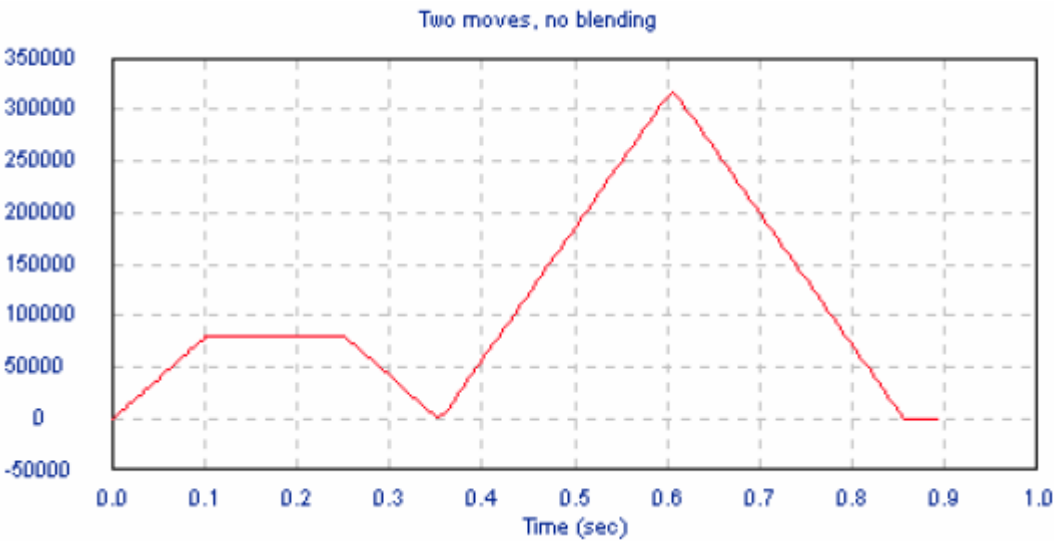
&1

#1->2000x

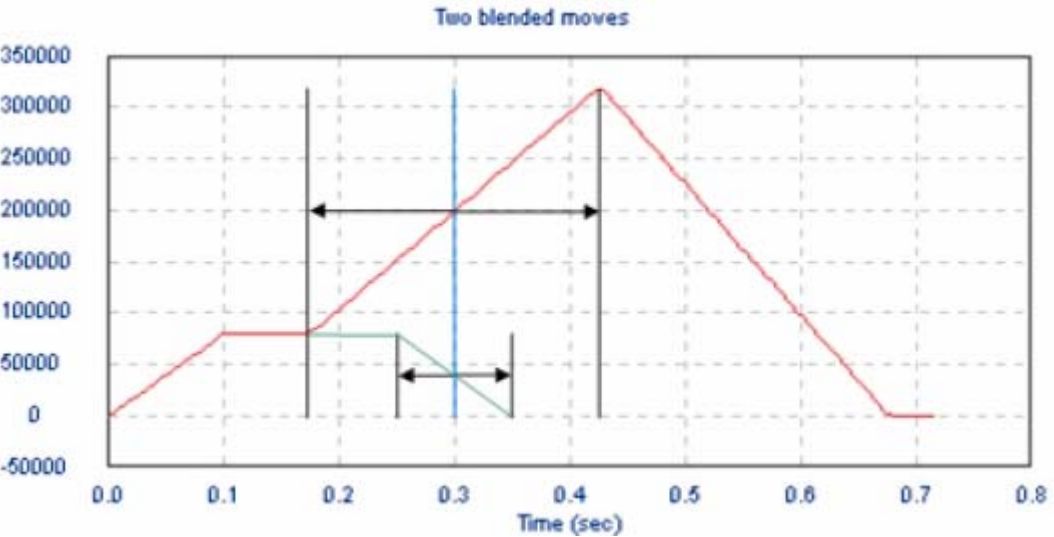
OPEN PROG 1 CLEAR

**LINEAR** ; Linear mode  
**INC** ; Incremental mode  
**TA100** ; The acceleration time is 100 msec, TA<sub>1</sub>  
**TS0** ; No S-curve component  
**TM250** ; Move time is 250 msec, TM<sub>1</sub>  
**X10** ; Move distance is 10 units, 20000 counts  
**TA250** ; Acceleration \ deceleration of the blended move is  
250 msec , TA<sub>2</sub>  
**X40** ; Move distance is 40 units, 80000 counts  
**CLOSE**

没有混合运动

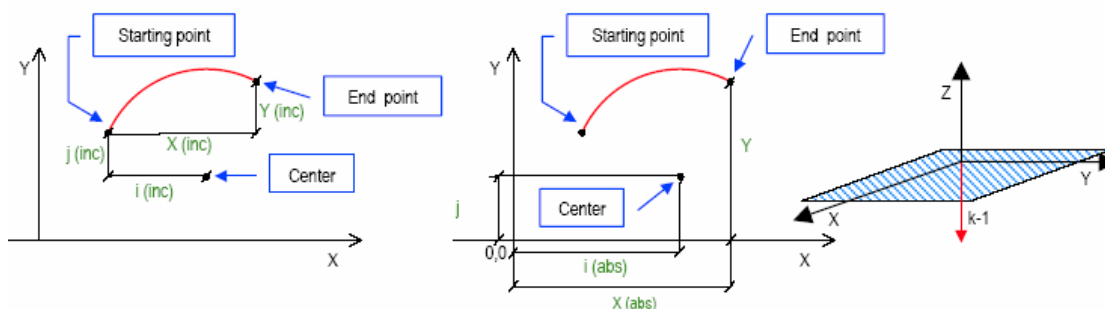


混合运动



## 第七节 圆弧查补

PMAC允许在坐标系里的X、Y、Z轴进行圆弧插补。圆弧插补时，也可以指定速率（F），也可以指定时间（TM）。



- 在PMAC上的参数**I13**是决定程序行步骤是否分段运行，如果**I13**等于0，将不再分段，圆弧插补将向直线插补一样。
- 如果**I13**大于0，这时圆弧插补就会过程中的速度和加速度就会受**Ix16**和**Ix17**节制。
- 在运行圆弧插补之前，必须使用**NORMAL**命令指定插补平面。

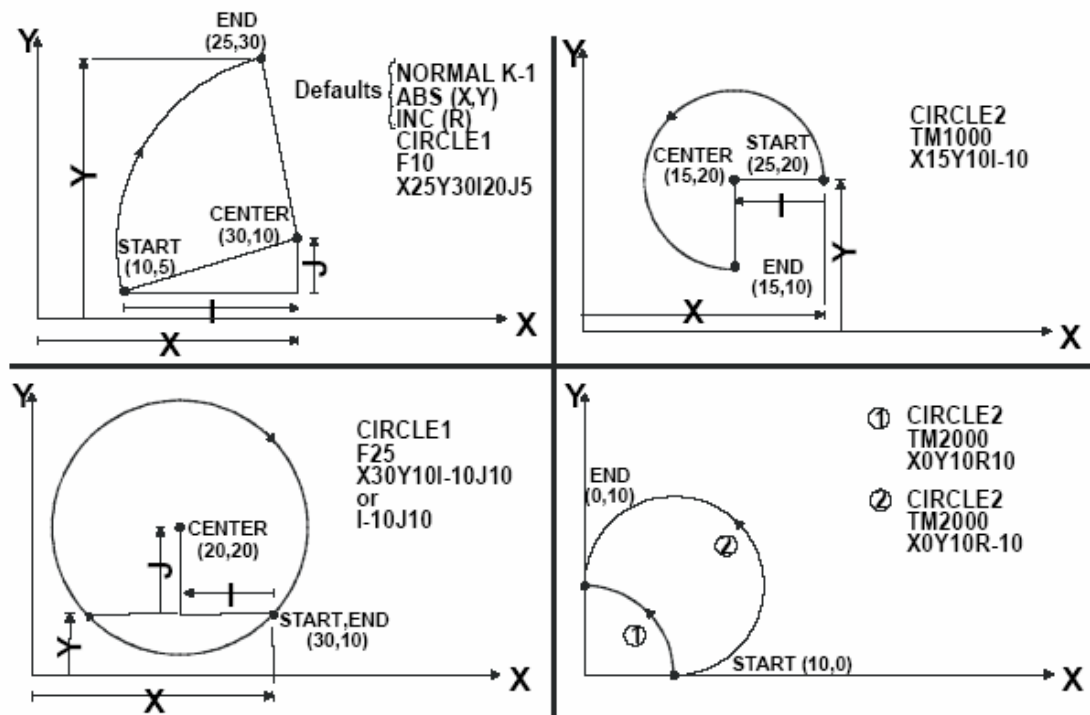
**NORMAL K-1** ; XY plane -- equivalent to G17

**NORMAL J-1** ; ZX plane -- equivalent to G18

**NORMAL I-1** ; YZ plane -- equivalent to G19

- 要使程序处于圆周模式下，使用编程命令**CIRCLE1**指定顺时针圆弧，命令**CIRCLE2**指定逆时针圆弧。指定起始点，半径（或用矢量表示）。注意在**ABS**和**INC**模式下，会有明显差别。

例子程序：



## 第八节 Splined 运动

PMAC能够执行样条运动。其中SPLINE1模式是利用了均匀非有理三次样条，SPLINE2模式是利用了不均匀非有理三次样条。

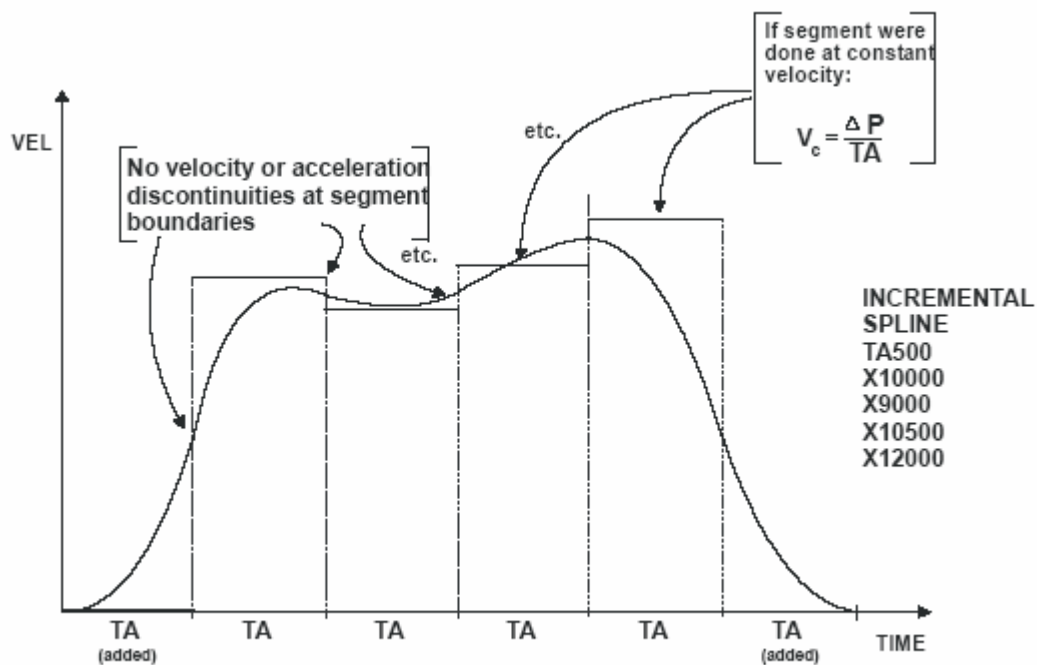
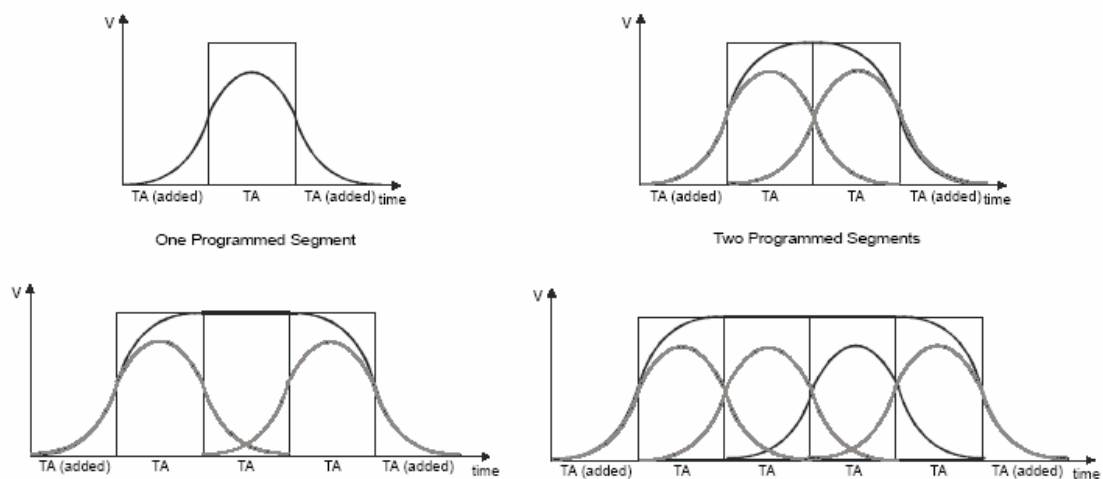
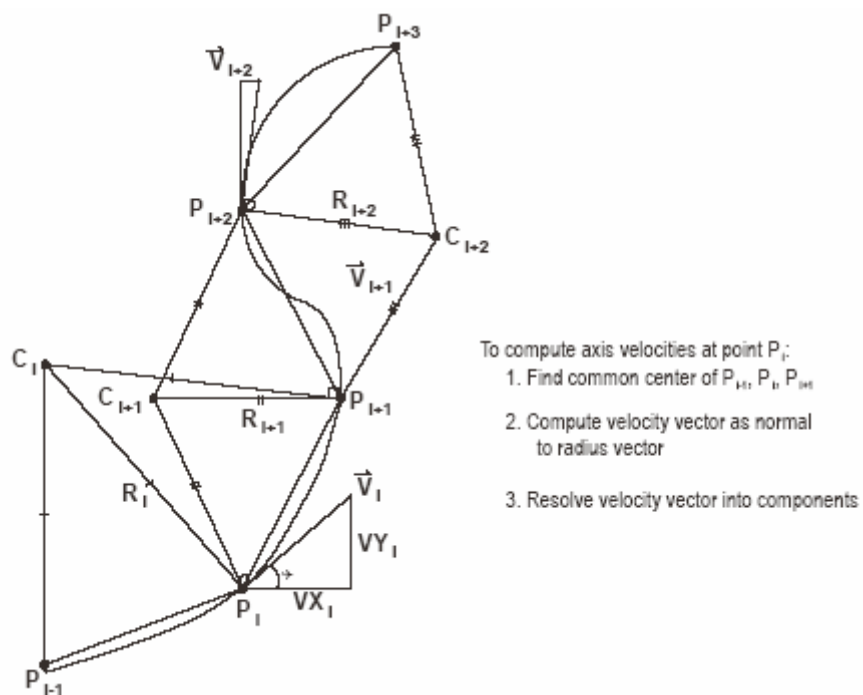


Figure 35 Splined Moves (All Segments at Same Time)



## 第九节 PVT-Mode 运动

对于要求直接地控制轨迹图形的用户，PMAC提供了位置-速度-时间（PVT）运动模式，在这些运动中，用户在运动时间的转换时直接指定轴的状态，这样要求上微机进行更多的运算，但是却允许对轨迹图形进行更紧凑的控制。

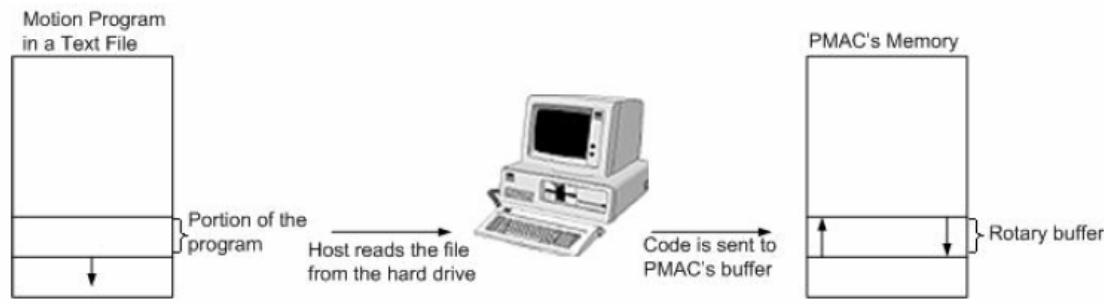




## 第十节 其他扩展特色运动

### 5.10.1 旋转缓冲区

PMAC 卡没有无限的内存空间存储我们的各种程序，但当有一个巨大的程序出现，我们可以使用旋转缓冲区的功能，这个功能看起来，既想拥有一个无限大的内存程序空间。



例子:

<b>&amp;1</b>	
<b>#1-&gt;2000X</b>	
<b>Define rot 2000</b>	;定义缓冲区
<b>Open rot</b>	;打开缓冲区
<b>R</b>	;开始运行
<b>X2000</b>	;电极响应指令
<b>A</b>	

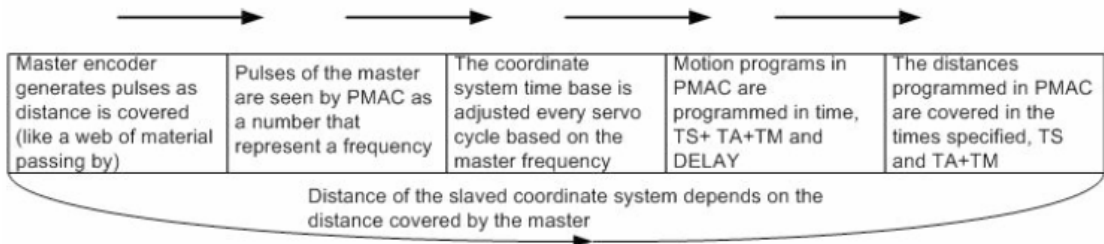
### 5.10.2 内部时基控制

每个坐标系都有自己的时间基础用来控制速度的倍率，这个参数是 **Ix93**，可以用不同方法调用：

- 在线指令：**%n**， $0 \leq n \leq 255$
- 地址访问：例如：**M197->X:\$0806,0,24,S**

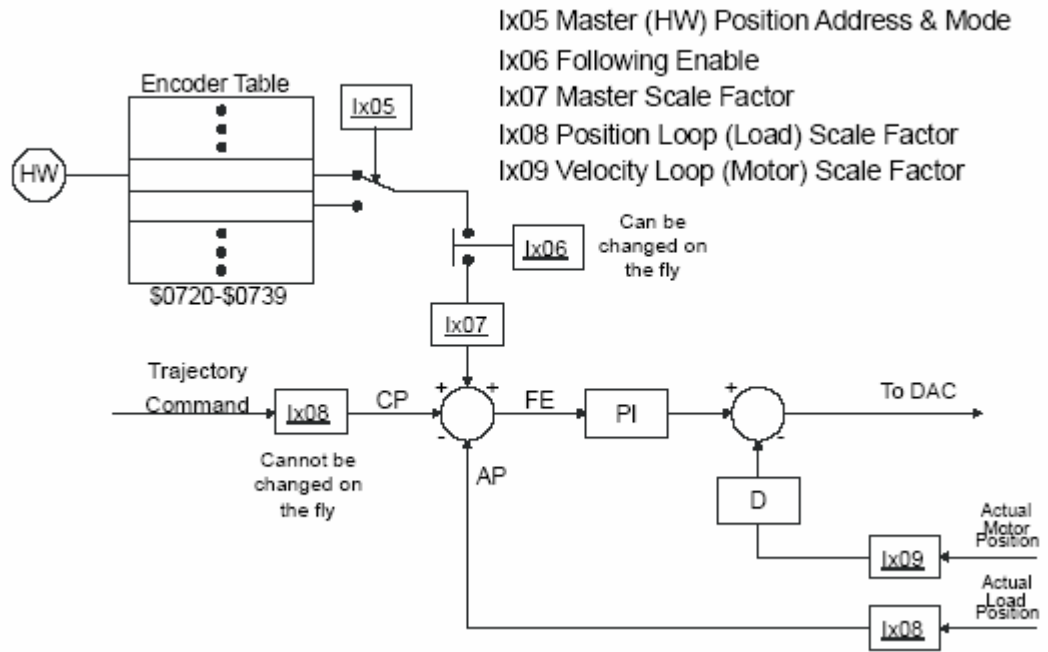
### 5.10.3 外部时基控制（电子凸轮）

利用外部编码器输入作为坐标系的时钟信号，这样该坐标系的运动，就可以与外部编码器具有严格的时间关系，保证了不同坐标系下的运动，具有一个严谨的同步性。



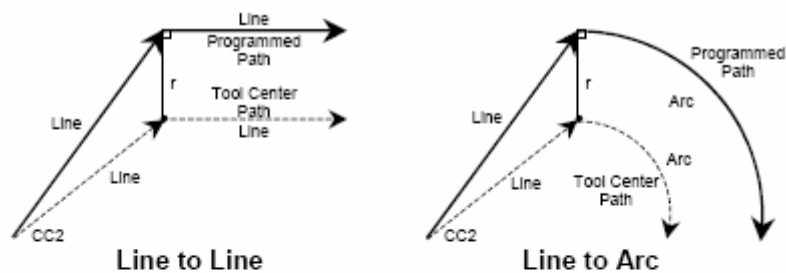
### 5.10.4 位置跟随（电子齿轮）

使用位置跟随功能，被控的电机就可以严格的按照指定的比例关系，与主控端同时运动，还可以使用手动指令进行调节。这个使用通过 Ix05，Ix06，Ix07 等参数实现。

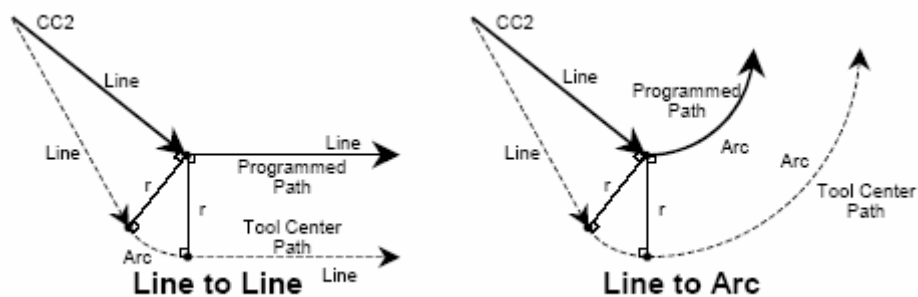
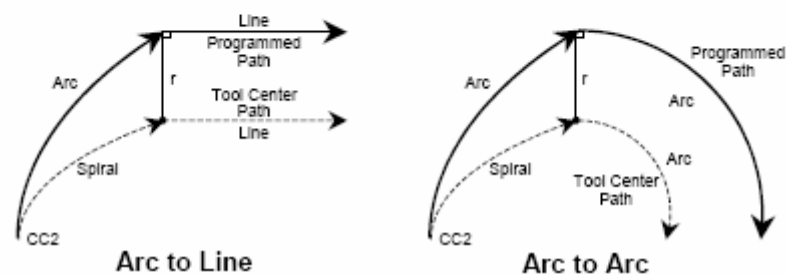


### 5.10.5 刀具半径补偿

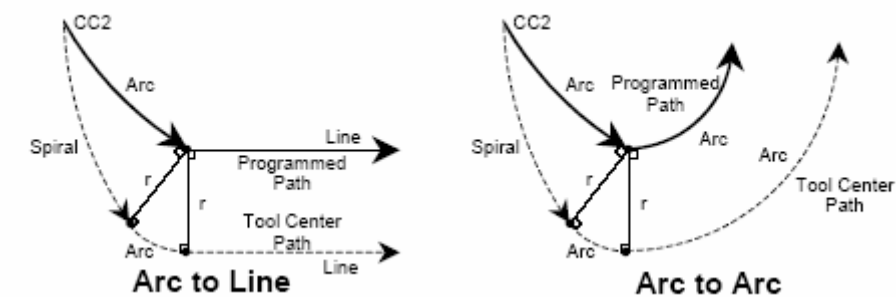
PMAC 提供了它在执行运动中实现刀具半径补偿的能力。该补偿可以在相互垂直的 X，Y，Z 轴中执行。该补偿通过一个编程的量给垂直于轨迹的运动增加偏置。刀具半径补偿只在线性和圆弧运动模式中才会有效。



内刀补



外刀补



### 5.10.6 同步 M 变量调用

当运动执行的时候，由于混合运动或者lookahead功能，PMAC会提前将程序中的M变量赋值语句执行，如果这时同步处理PLC，那么动作就会提前出现，影响效果，我们可以使用同步M变量调用方法，例如 $M1 == 1$ ；两个等号可以保证，M变量的复制在上一个语句的结束执行。这个指令只适用标准M变量，不适合P，Q，由TWB，TWD，TWR，TWS等定义的M变量。

## 5.10.7 多块 PMAC 同步

可以将多块PMAC同步于同一个时钟下，这样做需要跳线和连接的时钟线。

## 5.10.8 轴转置矩阵

使用轴的转置矩阵，可以跟灵活的调用X, Y, Z轴，或者建立不同于标准笛卡儿坐标系的坐标空间。为了支持这一功能，我们提供了灵活的Q变量调用方法，和指令**DEFINE TBUF, TSEL, TINIT, ADIS, IDIS, AROT, IROT**等。

## 5.10.9 位置捕捉或者位置比较

位置捕捉功能是当外部的一个信号输入时，将当前的位置信息保存起来。它是由硬件执行。不要软件干预，捕捉精度非常高。

位置比较功能是当位置信号满足我们的设定时，将通过硬件给出一个信号或中断。它是由硬件执行。不要软件干预，比较精度非常高。

## 5.10.10 会学习的运动程序

在打开的PMAC运动程序缓冲区里，可以使用在线指令，将当前指定的电机位置记录下来。

例子程序:

```
&1 ; Address coordinate system 1
#1->10000X#2->10000Y ;Define motor 1 motor 2 in C.S. 1
OPEN PROG 1 CLEAR ; Prepare program buffer for entry
F10 TA200 TS50 ; Enter required non-move commands
;{Move motors to a position, e.g. #1 to 13450 commanded, #2 to 29317 commanded}
LEARN(X,Y) ; Tell PMAC to learn these positions
X1.345 Y2.9317 ; This is the line that PMAC adds to PROG 1
;{move motors to new position, e.g. #1 to 16752 cmd., #2 to 34726 cmd}
LEARN ; Tell PMAC to learn positions
A0 B0 C0 U0 V0 W0 X1.6752 Y3.4726 Z0 ; PMAC adds positions for all axes to PROG 1
```

## 第六章      **PLC 编程**

## 第一节 关于 PLC 程序

PLC 程序可以不间断的在 PMAC 里面运行，通常用作处理与硬件有关的逻辑安排。PMAC 共有 64 个 PLC 程序（32 个编译的 PLCC，32 个未编译的 PLC）。PLC 的编译和未编译的程序都从 0 到 31 编号。PLC0 是在伺服中断周期结束时，以 I8 指定的频率运行一个很快的程序。该程序用于临界任务，必须足够的小。PLC1 到 31 在时间允许的范围内在后台不断地运行，实际是处于一个无限的循环中，他们会被电机定相、伺服环闭合、运动准备和 PLC0 等中断。

1. 打开PLC程序缓冲区，使用指令**OPEN PLC {constant}**，其中{constant}是程序的编号，0到32。
2. 命令**CLEAR**可以清空打开的运动程序缓冲区，PLC程序缓冲区，旋转缓冲区。
3. 使用**CLOSE**命令就可以结束运动程序缓冲区，PLC程序缓冲区，旋转缓冲区的编辑，进入在线指令的环境。通过**RETURN**指令还作为程序的返回点。
4. 使用命令**ENABLE PLC n**可以使能PLC程序。

**注意：**I5不同设置决定PLC是否能够使能，最常见的的设置是**I5=2**。

例子程序：

```
CLOSE
DELETE GATHER
DELETE TRACE
OPEN PLC n CLEAR
{PLC statements}
CLOSE
ENABLE PLC n
```

## 第二节 PLC 程序结构

在编写 PLC 程序的时候，必须记住一件重要的事，每个 PLC 程序都是无限循环的，它将不停的执行。

## 第三节 计算功能

PLC执行的大部分动作是通过变量的赋值语句来完成：**{variable}={expression}**。变量的类型可以是I,P, Q或M。这些动作能够影响到PMAC卡的内部和外部的很多事情。最简单的**P1=P1+1**

这样PLC程序执行，每次P1都将增加1。

这样的语句还可以更复杂一些：

**P2=M162/(I108\*32\*10000)\*COS (M262/(I208\*32\*100))**

## 第四节 有条件的计算

PLC 程序大部分动作是依靠 PMAC 的变量状态，如输入，输出，计算等语句。

比如：

**IF (M11=1)**

**P1=P1+1**

**ENDIF**

还可以使用边沿出发：

**IF (M11=1)**

**IF (P11=0)**

**P1=P1+1**

**P11=1**

**ENDIF**

**ELSE**

**P11=0**

**ENDIF**

## 第五节 While 循环

在 PLC 程序里面使用 While 循环时，遇到 Endwhile 就退出 PLC，下一次循环到来时，直接进入 While，而不是 PLC 程序头。

**WHILE (M11=1)**

**P1=P1+1**

**ENDWHILE**

## 第六节 Command、Send 等增强指令

使用 Commmad，Send 等增强指令，可以在缓冲区程序里面调用在线指令。

**M187->Y:\$0817,17,1** ; &1 In-position bit (AND of motors)

**OPEN PLC3 CLEAR**

**IF (M11=1)** ; input is ON

**IF (P11=0)** ; input was not ON last time

**P11=1** ; set latch

**COMMAND"&1A"** ; ABORT all motion

**WHILE (M187=0)** ; wait for motion to stop.

**ENDW**

**COMMAND"&1B10R"** ; start program 10

**ENDIF**

**ELSE**

**P11=0** ; reset latch

**ENDIF**

**CLOSE**

## 第七节 计时器

PMAC 拥有 4 个计时器，每个伺服周期 PMAC 给它们减值，用户可以在需要的时候，没他们赋值，读写通过 M 变量完成。

例子程序：

```

M90->X:$0700,0,24,S      ; Timer register 1 (8388608/I10 msec)
M91->Y:$0700,0,24,S      ; Timer register 2 (8388608/I10 msec)
M92->X:$0701,0,24,S      ; Timer register 3 (8388608/I10 msec)
M93->Y:$0701,0,24,S      ; Timer register 4 (8388608/I10 msec)
OPEN PLC3 CLEAR
M1=0                      ; Reset Output1 before start
M90=1000*8388608/I10      ; Set timer to 1000 msec, 1 second
WHILE (M90>0)             ; Loop until counts to zero
ENDWHILE
M1=1                      ; Set Output 1 after time elapsed
DIS PLC3                  ; disables PLC3 execution (needed in this example)
CLOSE

```

## 第八节 编译 PLC 程序

编译 PLC 的步骤如下：

1. 以解释的形式编写并调试 PLC 程序；
2. 将所有要被编译的 PLCn 程序的编号改为 PLCCn；
3. 对于整形便量，定义 L 变量，并用它们代替程序程序中老的变量名；
4. 将所有要编译的 PLC 程序在 PC 上组合成一个文件，用宏名代替 PMAC 中的代码；
5. 打开执行程序的下载时编译的特性，使用下载功能把文件下载到 PMAC 上；
6. 执行编译的 PLCC。



## 第七章      注意的问题

## 附页 1: PMAC 错误代码列表

错误	内容	解决方法
<b>ERR001</b>	执行程序中不可输入命令	终止程序
<b>ERR002</b>	口令错误	输入正确指令
<b>ERR003</b>	数据错误或不可知的命令	修改命令语法
<b>ERR004</b>	非法字符或奇偶校验出错	修改字符或检查线路
<b>ERR005</b>	需要缓冲区	打开缓冲区
<b>ERR006</b>	空间不足	释放无用空间
<b>ERR007</b>	缓冲区已经使用	关闭缓冲区
<b>ERR008</b>	MACRO 连接出错	
<b>ERR009</b>	程序结构错误	修改程序
<b>ERR010</b>	坐标系电机超程	修改行程或运动电机
<b>ERR011</b>	上次指令未完成	使用 Abort 结束指令
<b>ERR012</b>	坐标系电机开环	闭环
<b>ERR013</b>	坐标系电机没有激活	修改 Ix00
<b>ERR014</b>	坐标系无电机	定义电机
<b>ERR015</b>	非法缓冲区	清除缓冲区
<b>ERR016</b>	程序结构错误	修改程序
<b>ERR017</b>	使用/或\命令，电机没有在停止点	使用 j=指令

## 附页 2:     **PMAC I** 变量功能列表

## 附页 3: 在线指令列表

## 附页 4:     **PMAC** 编程指令列表

## 附页 5: 电机常用 **M** 变量定义

## 附页 7: 常见电机接线连接

## 附页 8: PMAC 附件与选项

C - 兼容

X - 不兼容

P - 局部兼容

O - 需要选项支持的兼容

A - 需要福建支持的兼容的

T - 兼容 Turbo 卡

N - 兼容非 Turbo 卡

附件	描述	PMAC MINI / PCI	PMAC LITE / PCI	PMAC PC / PCI	PMAC VME	PMAC2 MINI	PMAC2 LITE / PCI	PMAC2 PC / PCI	PMAC2 VME
<b>ACC-1L</b>	备份锂电池, 3V 1200 mAh, 2/3 A	X	X	X	X	X	X	X	X
<b>ACC-1LS</b>	备份锂电池, 3.6V 1000 mAh, 背板安装	C	C	C	C	O	O	O	O
<b>ACC-2A</b>	+/-15V (0.4 A) & +5 Volts HTAA-16WA 标准电源	C	C	P	P	A	O	O	P
<b>ACC-2B</b>	+/-15V (0.8 A) & +5 Volts HTAA-40WA 标准电源 (适用于 8 通道)	C	C	C	C	A	O	O	O
<b>ACC-3D</b>	3 米, RS-232 或者 RS-422 通信电缆, DB-25 转 IDC-26	X	O	C	C	X	C	O	C
<b>ACC-3D-O</b>									
<b>PT-1</b>	DB-9 转 DB-25	C	C	X	X	C	C	C	C
<b>ACC-3E</b>	ACC-3D 的延长中继装置, 可使 ACC3D 最长 16 米, 15cm 大小	X	O	C	C	X	C	A	C
<b>ACC-3L</b>	3 米, RS-232 通信电缆, DB-9 转 IDC-10	C	C	X	X	C	C	C	C
<b>ACC-3U1</b>	1.5m (5') terminated glass optical fiber cable	X	X	X	X	X	X	X	X
<b>ACC-3U2</b>	5m (15') terminated glass optical fiber cable	X	X	X	X	X	X	X	X
<b>ACC-3U3</b>	8m (28') terminated glass optical fiber cable.	X	X	X	X	X	X	X	X
<b>ACC-3U4</b>	Custom length terminated glass optical fiber cable <b>\$85.00 plus \$3.50 per meter.</b>	X	X	X	X	X	X	X	X
<b>ACC-8A</b>	2-通道模拟量接口板. 它可以提供脉冲加方向的输出, MLDTs 和其他的用法. 这个 Acc-8A 可以为每个通道提供一路编码器输出, 或者使用 Option-2, 为每通道提供两路模拟两输出(DAC)。在板提供 4 路分立的 DACs.	X	X	X	X	C	C	C	C
<b>ACC-8A-O</b>	256x 轴模拟量反馈插分	X	X	X	X	C	C	C	C



<b>PT-1</b>									
<b>ACC-8A</b>									
<b>OPT-2</b>	通讯	X	X	X	X	C	C	C	C
<b>ACC-8A-O</b>									
<b>PT-3</b>	安装支撑板	X	X	X	X	C	C	C	C
<b>ACC-8D</b>	PMAC(1) 4-通道终端连接板, 端子接头, IDC 接头形式 (必须选择 Opt-P or V)	C	C	C	C	X	X	X	X
<b>ACC-8D-O</b>									
<b>PT-P</b>	40 cm 电缆线 ,带 60-芯 IDC 转换端子	C	C	C	X	X	X	X	X
<b>ACC-8D-O</b>									
<b>PT-V</b>	40 cm 电缆线 ,带 96-芯 IDC 转换端子	X	X	X	X	X	X	X	X
<b>ACC-8D-O</b>									
<b>PTION V</b>									
<b>TO P</b>	40m 60 芯 IDC 转 96 芯	X	X	X	C	X	X	X	X
<b>ACC-8D-O</b>									
<b>PT-2</b>	V/F 转换输出-4 通道	C	X	X	C	X	X	X	X
<b>ACC-8D-O</b>									
<b>PT-A</b>	V/F 转换输出, 最大频率 10 KHz , 脉宽 20 u sec .	C	C	C	C	X	X	X	X
<b>ACC-8D-O</b>									
<b>PT-B</b>	V/F 转换输出, 最大频率 50 KHz , 脉宽 4 u sec .	C	C	C	C	X	X	X	X
<b>ACC-8D-O</b>									
<b>PT-C</b>	V/F 转换输出, 最大频率 100 KHz , 脉宽 2 u sec .	C	C	C	C	X	X	X	X
<b>ACC-8D-O</b>									
<b>PT-D</b>	V/F 转换输出, 最大频率 500 KHz , 脉宽 400 nsec .	C	C	C	C	X	X	X	X
<b>ACC-8D-O</b>									
<b>PT-E</b>	V/F 转换输出, 最大频率 1 MHz , 脉宽 200 nsec .	C	C	C	C	X	X	X	X
<b>ACC-8D-O</b>									
<b>PT-F</b>	V/F 转换输出, 最大频率 2 MHz , 脉宽 100 nsec .	C	C	C	C	X	X	X	X
<b>ACC-8D-O</b>									
<b>PT-G</b>	OPT 2 的安装支撑板	C	C	C	C	X	X	X	X
<b>ACC-8D-O</b>									
<b>PT-4</b>	模块化功率输出设备; 每通道 40W, 最高 30V , 电流环模式 平均电流 1.0A ,最大电流 2.0A , 需要 40-cm 电缆连接 ACC-8D 或者 ACC-8E	C	C	C	C	P	P	P	P
<b>ACC-8D-O</b>									
<b>PT-4A</b>	模块化功率输出设备; 每通道 150W, 最高 48V , 电流环模式 平均电流 3.0A ,最大电流 5.0A , 需要 40-cm 电缆连接 ACC-8D 或者 ACC-8E	C	C	C	C	C	C	C	C
<b>ACC-8D-O</b>									
<b>PT-B</b>	OPT 4 或者 OPT 4A 的安装支撑板	C	C	C	C	P	P	P	P
<b>ACC-8D-O</b>									
<b>PT-5</b>	OPT 8D 的安装支撑板	C	C	C	C	X	X	X	X
<b>ACC-8D-O</b>									
<b>PT-6</b>	带光电隔离的编码器输入, 需要 4 条 40-cm 电缆连接 ACC-8D 或者 ACC-8E 或者 ACC-8F	C	C	C	C	C	C	C	C
<b>ACC-8D-O</b>									
<b>PT-6</b>	OPT 6 的安装支撑板	C	C	C	C	C	C	C	C

<b>PT-6-OPT-A</b>									
<b>ACC-8D-OPT-7</b>	2-通道旋转变压器的输入; 12-位分辨率, 需要 2 根 10-芯, 40-cm 电缆, 1 根 26-芯 1 米电缆	C	C	C	C	C	C	C	C
<b>ACC-8D-OPT-7-OPT-A</b>	附加 2-通道旋转变压器的输入; 12-位分辨率, 需要 2 根 10-芯, 40-cm 电缆,	C	C	C	C	C	C	C	C
<b>ACC-8D-OPT-7-OPT-B</b>	OPT 7 的安装支撑板	C	C	C	C	C	C	C	C
<b>ACC-8DCE</b>	PMAC(1) 通道完整接口板, 通过 CE 认证的 (必须选择 Opt 1 or 2, 必须选择 Opt PCE 电缆或者替代物)	C	C	C	X	X	X	X	X
<b>ACC-8DCE-OPT-PCE</b>	PMAC-PC 连接 8DCE 电缆, 60-芯 45-cm	C	C	C	X	X	X	X	X
<b>ACC-8DCE-OPT-1</b>	端子输入型式	C	C	C	X	X	X	X	X
<b>ACC-8DCE-OPT-2</b>	D 型头输入型式	C	C	C	X	X	X	X	X
<b>ACC-8DCE-OPT-5</b>		C	C	C	X	X	X	X	X
<b>ACC-8DCE-OPT-6</b>	OPT 6 的安装支撑板	C	C	C	X	X	X	X	X
<b>ACC-8E</b>	PMAC2 2-通道模拟量终端接口板, 4 路模拟量输出, 2 路编码器输入	X	X	X	X	C	C	C	C
<b>ACC-8E-OP T-1</b>	安装支撑板	X	X	X	X	C	C	C	C
<b>ACC-8F</b>	PMAC2 2-通道 PWM 斩波终端接口板, 直接电流反馈, (必须选择 Option 1, 2, 或者 3)	X	X	X	X	C	C	C	C
<b>ACC-8F-OP T-1</b>	端子输入型式	X	X	X	X	C	C	C	C
<b>ACC-8F-OP T-2</b>	DB15 输入型式	X	X	X	X	C	C	C	C
<b>ACC-8F-OP T-4</b>	安装支撑板	X	X	X	X	C	C	C	C
<b>ACC-8F-OP T-5</b>	PWM 输入电缆的连接端子 (36 inch) (每个需要 2 个)	X	X	X	X	C	C	C	C
<b>ACC-8F-OP T-6</b>	增量编码器接口	X	X	X	X	C	C	C	C
<b>ACC-Cable</b>	100 pin flat cable, 24 inch long, used to interconnect PMAC2 family boards to breakout boards (ACC8xx)	X	X	X	X	C	C	C	C
<b>ACC-8M</b>	ACC-8M 利用附加轴通道做辅助输出. 必须 MINI PMAC 升级选项 OPTION LIM	C	X	X	X	X	X	X	X

<b>ACC-8P</b>	PMAC(1) 4-通道终端连接板, 端子接头, IDC 接头形式 (必须选择 Opt-P or V)	C	C	C	C	X	X	X	X
<b>ACC-8P OPT-P</b>	40 cm 电缆线 ,带 60-芯 IDC 转换端子	C	C	C	X	X	X	X	X
<b>ACC-8P OPT-V</b>	40 cm 电缆线 ,带 96-芯 IDC 转换端子	X	X	X	C	X	X	X	X
<b>ACC-8S</b>	PMAC2 两轴步进电机接口板	X	X	X	X	C	C	C	C
<b>ACC-8S-OP T-1</b>	增量编码器接口	X	X	X	X	C	C	C	C
<b>ACC-8S-OP T-2</b>	安装支撑板	X	X	X	X	C	C	C	C
<b>ACC-8T</b>	追加功能的扩展板	X	X	X	X	C	C	C	C
<b>ACC-8TOP T-1</b>	安装支撑板	X	X	X	X	C	C	C	C
<b>ACC-11P</b>	PCI 总线的数字 I/O 扩展板 . 24 光电隔离输入, 24 光电隔离输出, 24V 电压.	C	C	C	P	C	C	C	P
<b>ACC-11P OPT-1</b>	附加 24 光电隔离输入, 24 光电隔离输出 I/O 扩展, 24V 电压.	C	C	C	P	C	C	C	P
<b>ACC-12A</b>	显示面板, 24x2 字符 LCD 显示 高 5mm 需要 180-cm 电缆,	C	C	C	C	C	C	C	C
<b>ACC-12C1</b>	显示面板, 40x2 字符 LCD 显示 高 5mm 需要 180-cm 电缆,	C	C	C	C	C	C	C	C
<b>ACC-12D</b>	显示面板, 40x2 字符 真空荧光管显示 高 5mm 需要 180-cm 电缆,	C	C	C	C	C	C	C	C
<b>ACC-12D-O PT-1</b>	长距离扩展模块	C	C	C	C	C	C	C	C
<b>ACC-14P</b>	PCI 总线的数字 I/O 扩展板. 48 TTL 形式 I/O 点; 个别的需要方向选择. 没有发布	C	C	C	P	C	C	C	P
<b>ACC-14P OPT-1</b>	PCI 总线的数字 I/O 扩展板. 附加 48 TTL 形式 I/O 点; 个别的需要方向选择. 没有发布	C	C	C	P	C	C	C	P
<b>ACC-14V</b>	ISA 总线的数字 I/O 扩展板. 48 TTL 形式 I/O 点; 需要选择 Opt1-4.	P	P	P	C	P	P	P	C
<b>ACC-14V-O PT-1</b>	24 inputs 和 24 outputs; TTL 形式 5 volt	P	P	P	C	P	P	P	C
<b>ACC-14V-O PT-2</b>	24 inputs 和 24 outputs; 24 volt	P	P	P	C	P	P	P	C
<b>ACC-14V-O PT-3</b>	48 inputs; TTL 形式 5 volt, 用作并行反馈	P	P	P	C	P	P	P	C
<b>ACC-14V-O PT-4</b>	用户自定义形式	P	P	P	C	P	P	P	C

<b>ACC-14V-O PT-7</b>	20-cm 50-芯 3 个连接端子电缆, 用于 ACC-14D 连接 PMAC OPT-2, ACC-24P, ACC-29P 或着 ACC-36P	P	P	P	C	P	P	P	C
<b>ACC-21A</b>	Connector Board between JI/O port on PMAC 2 & Opto-22 Board	X	X	X	X	C	C	C	C
<b>ACC-21F</b>	Cable 50-pin card-edge to 34-pin IDC header for PMAC (1) JOPT connector, 180-cm (6') long, for connection to PB8/16/24 or equivalent boards	C	C	C	C	X	X	X	X
<b>ACC-21FH</b>	Cable 50-pin IDC header to 34-pin IDC header for PMAC (1) JOPT connector, 180-cm (6') long, for connection to PB8/16/24H or equivalent boards	C	C	C	C	X	X	X	X
<b>ACC-21G</b>	Cable 50-pin card-edge to 50-pin IDC header for ACC-14D/V & ACC-34B/D connector, 180-cm (6') long, for connection to PB8/16/24 or equivalent boards	A	A	A	A	A	A	A	A
<b>ACC-21GH</b>	Cable 50-pin IDC header to 50-pin IDC header for ACC-14D/V & ACC34B/D connector, 180-cm (6') long, for connection to PB8/16/24H or equivalent boards	A	A	A	A	A	A	A	A
<b>ACC-22</b>	Extended warranty, to 2 years from data of purchase (can be purchased for any Delta Tau product, please specify the product)	C	C	C	C	C	C	C	C
<b>ACC-24P</b>	4-通道 PMAC (1)-轴扩展卡, <b>ISA-总线</b> , 需要 2/8-cm 电缆. 不支持 PCI 系列	P	C	C	P	P	P	P	P
<b>ACC-24P-O PT-1</b>	附加 4-通道 PMAC (1)-轴扩展	P	C	C	P	P	P	P	P
<b>ACC-24P-O PT-2</b>	20-cm 50 芯 3-电缆插头连接 ACC-24P	P	C	C	P	P	P	P	P
<b>ACC-24P2</b>	4-通道 PMAC (2)-轴扩展卡, <b>ISA-总线</b> , 需要 1/8-cm 电缆. 不支持 PCI 系列	X	X	X	X	C	C	C	C
<b>ACC-24P2- OPT-1</b>	附加 4-通道 PMAC (2)-轴扩展	X	X	X	X	C	C	C	C
<b>ACC-24P2- OPT-2</b>	20-cm 50 芯 3-电缆插头连接 ACC-24P2	X	X	X	X	C	C	C	C
<b>ACC-24V</b>	4-通道 PMAC (1)-轴扩展卡, <b>VME-总线</b> , 需要 2/8-cm 电缆. 不支持 PCI 系列	X	X	X	C	X	X	X	P
<b>ACC-24V-O PT-1</b>	附加 4-通道 PMAC (1)-轴扩展	X	X	X	C	X	X	X	P
<b>ACC-26A</b>	串行通信转换板, 必须 OPT 1, OPT 2 或者 OPT 3	C	C	C	C	C	C	C	C
<b>OPT-1</b>	上位机 RS-232 到 PMAC RS-422, 26 芯 60cm 电缆	X	X	C	C	X	O	O	C
<b>OPT-2</b>	RS-232 到 RS-232, 10 芯 60cm 电缆	C	C	X	X	C	C	C	X
<b>OPT-3</b>	RS-232 到 RS-422, (适用 GE Fanuc 系列 90-70 PLC)	X	X	X	X	X	X	X	X

<b>ACC-28A</b>	4-通道 A/D 转换板, 15-位 绝对分辨率, 12-位重复精度, 需要 60-cm 电缆 (必须选择 Opt. 2A 或者 2B)	C	C	C	C	X	X	X	X
<b>ACC-28A OPT-2A</b>	DB-15 端子输入	C	C	C	C	X	X	X	X
<b>ACC-28A OPT-2B</b>	12-芯端子输入(默认)	C	C	C	C	X	X	X	X
<b>ACC-28A OPT-3</b>	安装支撑板	C	C	C	C	X	X	X	X
<b>ACC-28B</b>	2-通道 A/D 转换板, 16-位 绝对分辨率, 15-位重复精度, (必须选择 Opt. 2A 或者 2B)	C	C	C	C	A	A	A	A
<b>ACC-28B-O PT-1</b>	附加 2-通道 A/D 转换	C	C	C	C	A	A	A	A
<b>ACC-28B-O PT-2A</b>	DB-25 端子输入	C	C	C	C	A	A	A	A
<b>ACC-28B-O PT-2B</b>	端子输入	C	C	C	C	A	A	A	A
<b>ACC-28B-O PT-3</b>	连接 PMAC 2 适用	X	X	X	X	A	A	A	A
<b>ACC-28B-O PT-4</b>	安装支撑板	C	C	C	C	A	A	A	A
<b>ACC-31</b>	4-轴演示开发模块 (PMAC 必须单独购买) (没有控制面板)	C	C	C	C	X	X	X	X
<b>ACC-31-OP T-1</b>	附加 4-轴演示开发模块 (PMAC 必须单独购买) (没有控制面板)	C	C	C	C	X	X	X	X
<b>ACC-31-OP T-3</b>	独立放置 PMAC-Lite/-PC 的装置	X	C	C	X	X	X	X	X
<b>ACC-32 OPT-1</b>	固件升级服务, 允许用户购买额外的 EPROM	X	C	C	C	X	X	X	X
<b>ACC-34AA</b>	32 输入/32 输出光电隔离, 64 位分立式 I/O 板, 必须选择 Option 1 or 2	C	C	C	C	C	C	C	C
<b>ACC-34AA- OPT-1</b>	共阴极	C	C	C	C	C	C	C	C
<b>ACC-34AA- OPT-2</b>	共源极	C	C	C	C	C	C	C	C
<b>ACC-34AA- OPT-3</b>	安装支撑板	C	C	C	C	C	C	C	C
<b>ACC-34B</b>	32 输入/32 输出光电隔离, 64 位分立式 I/O 板, 可以方便的与 Opto22 连接	C	C	C	C	C	C	C	C
<b>ACC-34B-O PT-1</b>	安装支撑板	C	C	C	C	C	C	C	C
<b>ACC-36P</b>	16-通道, 12-位 A/D 转换板, 可使用 ISA Bus	C	C	C	P	C	C	C	P

<b>ACC-36P-O PT-2</b>	20-cm (8 inch) 50-芯 3-电缆插头 连接 ACC-36P 与 PMAC Option 2, ACC-14D 或者 ACC-24P	C	C	C	P	C	C	C	P
<b>ACC-36P-O PT-1</b>	安装支撑板	P	C	C	C	P	P	P	P
<b>ACC-40</b>	Delta Tau 公司的现场技术支持和培训, 最低是两天, 包括寄宿, 旅程时间 注意: <b>每小时\$100.00</b>	C	C	C	C	C	C	C	C
<b>ACC-42P2</b>	MACRO 接口板, 用于 PMAC2-PC (需要 OPT-A or C)	X	X	X	X	C	C	C	X
<b>ACC-42P2- OPTA</b>	Fiber optic 接口	X	X	X	X	C	C	C	X
<b>ACC-42P2- OPT-C</b>	RJ-45 接口	X	X	X	X	C	C	C	X
<b>ACC-42P2- OPT-3A</b>	OPT-3A 1.5m (5ft) terminated glass optical fiber cable	X	X	X	X	C	C	C	X
<b>ACC-42P2- OPT 3B</b>	OPT-3B 5m (15ft) terminated glass optical fiber cable	X	X	X	X	C	C	C	X
<b>ACC-42P2- OPT-3C</b>	OPT-3C 8m (25ft) terminated glass optical fiber cable	X	X	X	X	C	C	C	X
<b>ACC-42P2- OPT-3D</b>	OPT-3D Custom-length terminated glass optical fiber cable. <b>\$85.00 plus \$3.50 per meter.</b>	X	X	X	X	C	C	C	X
<b>ACC-42PB- OPT-3A</b>	OPT-3A 1.5m (5ft) terminated glass optical fiber cable	X	C	C	P	X	X	X	X
<b>ACC-42PB- OPT-3B</b>	OPT-3B 5m (15ft) terminated glass optical fiber cable	X	C	C	P	X	X	X	X
<b>ACC-42PB- OPT-3C</b>	OPT-3C 8m (25ft) terminated glass optical fiber cable	X	C	C	P	X	X	X	X
<b>ACC-42PB- OPT-3D</b>	OPT-3D Custom-length terminated glass optical fiber cable <b>\$85.00 plus \$3.50 per meter.</b>	X	C	C	P	X	X	X	X
<b>ACC-42PCI</b>	MACRO 接口板, 用于 PMAC2-PC (需要 OPT-A or C)	X	X	X	X	C	C	C	X
<b>ACC-42PCI -OPT-A</b>	Fiber optic 接口	X	X	X	X	C	C	C	X
<b>ACC-42PCI -OPT-C</b>	RJ-45 接口	X	X	X	X	C	C	C	X
<b>ACC-42PCI -OPT-3A</b>	OPT-3A 1.5m (5ft) terminated glass optical fiber cable	X	X	X	X	C	C	C	X
<b>ACC-42PCI -OPT-3B</b>	OPT-3B 5m (15ft) terminated glass optical fiber cable	X	X	X	X	C	C	C	X
<b>ACC-42PCI -OPT-3C</b>	OPT-3C 8m (25ft) terminated glass optical fiber cable.	X	X	X	X	C	C	C	X

<b>ACC-42PCI -OPT-3D</b>	OPT-3D Custom-lenth terminated glass optical fiber cable <b>\$85.00 plus \$3.50 per meter</b>	<b>X</b>	<b>X</b>	<b>X</b>	<b>X</b>	<b>C</b>	<b>C</b>	<b>C</b>	<b>X</b>
<b>ACC-51P</b>	2 通道高分辨率正弦波编码器插补, 4096 线细分, PCI 总线	<b>C</b>	<b>C</b>	<b>C</b>	<b>C</b>	<b>C</b>	<b>C</b>	<b>C</b>	<b>C</b>
<b>ACC-51P-O PT-1</b>	扩展 2 通道高分辨率正弦波编码器插补, 4096 线细分, PCI 总线	<b>C</b>	<b>C</b>	<b>C</b>	<b>C</b>	<b>C</b>	<b>C</b>	<b>C</b>	<b>C</b>
<b>ACC-51P-O PT2</b>	Hiperface 接口	<b>C</b>	<b>C</b>	<b>C</b>	<b>C</b>	<b>C</b>	<b>C</b>	<b>C</b>	<b>C</b>
<b>ACC-70P</b>	Tamagawa ABS 编码器输入. 2 通道	<b>C</b>	<b>C</b>	<b>C</b>	<b>P</b>	<b>C</b>	<b>C</b>	<b>C</b>	<b>P</b>
<b>ACC-70P - OPT-1</b>	附加 Tamagawa ABS 编码器输入. 2 通道	<b>C</b>	<b>C</b>	<b>C</b>	<b>P</b>	<b>C</b>	<b>C</b>	<b>C</b>	<b>P</b>
<b>ACC-76</b>	Uses shielded cable between PMAC and ACC-76 and "D" shell style connectors. A 26 pin IDC ribbon to DB25 connector adapter cable is included with ACC-76. This Accessory is Compatible with Opto22 SNAP-D64RS racks and all SNAP digital I/O modules.	<b>C</b>	<b>C</b>	<b>C</b>	<b>C</b>	<b>C</b>	<b>C</b>	<b>C</b>	<b>C</b>
<b>ACC-77</b>	Provides 64 input channels only. Compatible with Opto22 SNAP-D64RS backplanes and all SNAP digital I/O modules.	<b>C</b>	<b>C</b>	<b>C</b>	<b>C</b>	<b>C</b>	<b>C</b>	<b>C</b>	<b>C</b>
<b>Shielded Cable</b>	For ACC-76 or ACC-77 shielded cable - 30cm (12in) for interconnecting between multiple ACC-76 and ACC-77 modules								
<b>Shielded Cable</b>	For ACC-76 or ACC-77 interconnecting shielded cable - 60cm (24in) interconnecting between multiple ACC-76 and ACC-77 modules								
<b>Shielded Cable</b>	For ACC-76 or ACC-77 interconnecting shielded cable - 90cm (36in) interconnecting between multiple ACC-76 and ACC-77 modules								
<b>Shielded Cable</b>	For ACC-76 or ACC-77 interconnecting shielded cable - 180cm (72in) interconnecting between multiple ACC-76 and ACC-77 modules								
<b>Cable</b>	90cm (36in) flat ribbon cable for connecting ACC-76 or ACC-77 to PMAC								