

# PCI1040 运动控制卡

## 软件使用说明书



北京阿尔泰科技发展有限公司

产品研发部修订



## 目 录

目 录 .....	1
第一章 概述.....	3
第一节、特点.....	3
第二节、规格.....	3
第二章 功能描述.....	6
第一节、脉冲输出功能.....	6
一、定长脉冲驱动输出.....	6
二、连续驱动输出.....	7
三、脉冲输出类型.....	7
第二节、速度曲线.....	7
一、定速驱动.....	7
二、直线加/减速驱动.....	8
三、S 曲线加/减速驱动 .....	8
第三节、插补.....	8
第四节、信号输入/输出.....	9
一、通用输入/输出.....	9
二、传感器输出.....	9
三、清除伺服驱动计数器输出.....	9
第五节、同步控制.....	9
第六节、外部启动和原点搜寻.....	9
一、外部启动.....	9
(一)、传感器位置驱动类型 I .....	9
(二)、传感器位置驱动类型 II .....	10
(三)、传感器位置驱动类型 III .....	10
二、返回原点.....	10
(一)、返回原点位置驱动类型 I .....	10
(二)、返回原点位置驱动类型 II .....	11
(三)、返回原点位置驱动类型 III .....	11
(四)、返回原点位置驱动类型 IV .....	11
第七节、中断功能.....	12
第八节、硬件限位功能.....	12
一、越限信号.....	12
二、伺服电机紧急停止报警信号.....	13
三、伺服电机到位信号.....	13
四、输入信号滤波器.....	13
五、减速停止信号.....	13
第九节、位置计数.....	13
一、逻辑位置计数器.....	13
二、实际位置计数器.....	13
三、比较寄存器.....	14
第三章 驱动函数库.....	15
第一节、驱动函数库函数列表.....	15
第二节、驱动函数库说明.....	17

一、设备对象管理函数.....	17
二、设置控制卡的基本参数函数.....	18
三、设置编码器输入信号类型.....	21
四、直线 S 曲线初始化、启动函数.....	22
五、任意 2 轴直线插补初始化、启动函数.....	23
六、任意 3 轴直线插补初始化、启动函数.....	25
七、任意 4 轴直线插补初始化、启动函数.....	26
八、插补相关函数.....	28
九、计数器相关函数.....	29
十、减速函数设置.....	34
十一、设置同步位.....	36
十二、设置原点搜寻外部信号停止电机驱动的输入信号.....	37
十三、外部信号启动电机定长驱动、连续驱动.....	38
十四、设置外部信号启动电机驱动的输入信号.....	38
十五、初始化外部输入信号.....	39
十六、设置常速有效和无效.....	41
十七、设置通用输入和通用输出.....	42
十八、读电机速度.....	43
十九、读状态寄存器的位状态.....	44
二十、中断位设置、插补中断状态清除.....	46
第四章 硬件参数结构.....	48
第一节、公用参数介绍 (PCI1040_PARA_DataList).....	48
第二节、直线和 S 曲线参数介绍 (PCI1040_PAPA_LCData).....	48
第三节、插补轴参数介绍 (PCI1040_PAPA_InterpolationAxis).....	49
第四节、直线插补和固定线速度直线插补参数介绍 (PCI1040_PAPA_LineData).....	49
第五节、设置中断位使能参数介绍 (PCI1040_PARA_Interrupt).....	50
第六节、IO 输出参数介绍 (PCI1040_PARA_DO).....	50
第七节、IO 输入参数介绍 (PCI1040_PARA_DI).....	51
第八节、运行状态寄存器参数介绍 (PCI1040_PARA_RUN).....	51
第九节、传感器状态寄存器参数介绍 (PCI1040_PARA_SENSOR).....	51
第十节、中断状态寄存器参数介绍 (PCI1040_FLAG_INT).....	51
第十一节、主要状态寄存器参数介绍 (PCI1040_MAIN_STATUS).....	52
第十二节、正常停止原因状态结构体介绍 (PCI1040_NORMAL_STOP_FACTOR).....	52
第十三节、错误停止原因状态结构体介绍 (PCI1040_ERROR_STOP_FACTOR).....	52
第十四节、Counter A/B 计数器参数设置结构体 (PCI1040_COUNTER_MODE).....	52



## 第一章 概述

PCI1040 是 PCI 总线 8 轴伺服/步进电机运动控制卡，它以高频率脉冲串形式输出，控制伺服/步进电机的运动。该卡能精确地控制所发出的脉冲频率（电机速度）、脉冲个数（电机转角）及脉冲频率变化率（电机加速度），它能满足步进电机的各种复杂的控制要求。可对电机进行位置控制、插补驱动、加速/减速等控制。具有圆弧、直线插补功能。它含有丰富的，功能齐全的软件库函数资源。在 Windows9X/2000/XP 环境下，用户可直接使用我们为您提供的设备驱动程序函数接口；以最大方便地使您在 Visual C++、Visual Basic 及各种其他软件环境中使用本设备。

### 第一节、特点

- ◇ 32 位 PCI 接口，即插即用
- ◇ 8 轴伺服/步进电机控制，每轴可独立控制，互不影响
- ◇ 脉冲输出的频率误差小于 0.1%
- ◇ 脉冲输出速度最高可达到 5MHz
- ◇ 可选择脉冲输出模式：PULSE/DIR，CW/CCW
- ◇ 直线加/减速驱动、S 曲线加/减速驱动
- ◇ 梯形与 S 曲线速度轮廓
- ◇ 任意 2 轴或 3 轴或 4 轴直线插补
- ◇ 固定线速度控制
- ◇ 多轴同步启动/停止
- ◇ 可编程控制加速与减速时间
- ◇ 在运动中改变输出脉冲数或驱动速度
- ◇ 运动中可以实时读出逻辑位置、实际位置、驱动速度、加速度、加/减速状态（加速中、定速中、减速中）
- ◇ 每轴都有 2 个 32 位比较寄存器用于逻辑位置计数器或者实际位置计数器的位置大小比较，可用于硬件限位
- ◇ 可接收伺服马达驱动器的各种信号，如硬件限位信号、到位信号、报警信号等
- ◇ 32 位递增/递减计数器用于附加编码器
- ◇ 可方便地与任意步进电机、AC 或 DC 伺服电机相连接
- ◇ 所有数字量输入/输出信号均有 2500Vrms 隔离

### 第二节、规格

- ◇ 控制轴 8 轴
- ◇ CPU 数据总线长度 64 位

#### 一、插补功能

- ◇ 任意 2 轴或 3 轴或 4 轴直线插补
  - 插补范围 各个轴 1 ~ 16,777,215
  - 插补速度 1~5MPPS
  - 插补位置精度  $\pm 0.5\text{LSB}$  以下（在全插补范围内）

#### 二、各轴通用规格

- ◇ 驱动输出脉冲（CLK=16.384MHz 时）
  - 输入时钟 最大 20.0MHz，推荐使用 16.384MHz 或 19.6608MHz
  - 输出速度精密度  $\pm 0.1\%$  以下（对设定数值）
  - 最大输出速率 1 ~ 5MPPS（直线加/减速）
  - 1 ~ 3.05MPPS（S 曲线加/减速）
  - 速度倍率 1~250

- 加/减速度 125 ~ 511,968,750 PPS/SEC
- 初始速度 1 ~ 4,095,750 PPS (倍率=1 的时候)
- 驱动速度 1 ~ 4,095,750 PPS (倍率=1 的时候)
- 输出脉冲数 0 ~ 268,435,455 (定量驱动)
- 速度曲线 定速/直线加减速/抛物线 S 曲线加减速驱动
- 定量驱动的减速模式 自动减速/手动减速/偏移减速
- 可以在驱动中改变输出脉冲数、驱动速度
- 可选择 CW/CCW、PULSE/DIR 输出模式
- 可以选择独立 2 脉冲/1 脉冲方向方式
- 输出脉冲计数范围 (R1) 1 ~ 16,777,215
- 减速开始点范围(R2) 0 ~ 16,777,215 (手动减速)  
-8388608 ~ 8388607 (自动计算偏移减速)
- 频率倍率范围(R0) 1 ~ 4095
- 步进频率计数(R3,R4) 1 ~ 16383 (直线加速/减速, S 曲线加速/减速, 减速手动设置模式)  
1 ~ 10000 (S 曲线加速/减速, 减速开始点自动设置模式)
- 加速/减速范围(R5,R6) 1 ~ 16383
- S 曲线加减速段范围(R7) 1 ~ 8191
- ◇ 驱动接口
  - 输出 时钟输出: 门控输出/2 时钟输出切换, 可逻辑切换  
单脉冲输出: 大约 1.9us, 逻辑切换  
伺服输出: 通用输出
  - 输入 驱动报警输入: 1 点/轴, 逻辑切换  
结束位置输入: 1 点/轴, 逻辑切换
- ◇ 编码器输入脉冲
  - A 相、B 相、Z 相输入, 每一个轴各 1 个
  - 可以选择 2 相脉冲/上下脉冲输入
- ◇ 传感器输入
  - 灵敏度范围: 0 ~ 255
  - 限位输入(ELP、ELM) 2 个点, +/-向, 可逻辑切换
  - 减速输入(SLP、SLM) 2 点, +/-向, 可逻辑切换, 也可逻辑切换减速/减速停止
  - 原点输入(ORG、EZ) 2 点/轴, 原点和 Z 相, 可逻辑切换
  - 定位传感器输入(INP) 1 点/轴, 可逻辑切换
- ◇ 位置计数器
  - 逻辑位置计数器 (输出脉冲用) 计数范围-8,388,608~+8,388,607
  - 实位计数器 (输入脉冲用) 计数范围-8,388,608~+8,388,607
  - 可以一直写入读出
- ◇ 比较寄存器
  - COMP P 寄存器位置比较范围 -8,388,608~+8,388,607
  - COMP Q 寄存器位置比较范围 -8,388,608~+8,388,607
  - 把和位置计数器的大小比较状态输出, 放在读寄存器中
  - 可以作为硬件限位
- ◇ 外部减速停止/立即停止信号
  - 每一个轴都有 1 个外部减速停止信号和 1 个外部立即停止信号
  - 任何信号都可以选择有效无效
- ◇ 伺服马达输入信号
  - ALM (警报), INP (定位完毕)



- 任何信号都可以选择有效/无效
- ◇ 超越限制信号输入
  - +方向 ELP, -方向 ELM, 每轴一个
- ◇ 报警紧急停止信号输入
  - 每轴都有一个 ALM
- ◇ 通用输入/输出信号
  - IN0~7 每一个轴 1 个
  - OUT0~7 每一个轴 1 个
- ◇ 其他输入/输出信号
  - SYNC 同步输入, 全轴只有一个
  - CLRA1~8 清计数器 A (逻辑计数器) 输入, 每一个轴 1 个
  - CLR0~7 清伺服驱动计数器输出, 每一个轴 1 个
  - SON0~7 传感器输出, 每一个轴 1 个
- ◇ 电气特性
  - 工作温度 0 ~ +50°C
  - 电源 24V (外接)
  - 时钟 16.384MHz

## 第二章 功能描述

### 第一节、脉冲输出功能

脉冲输出功能有两种模式：定长脉冲驱动输出和连续驱动脉冲输出。

#### 一、定长脉冲驱动输出

定长脉冲驱动的意思是以固定速度或加/减速度输出指定数量的脉冲需要移动到确定的位置或进行确定的动作时使用此功能加/减速的定量驱动如图 4-1 所示输出脉冲的剩余数比加速累计的脉冲数少时就开始减速输出指定的脉冲数后驱动也结束。

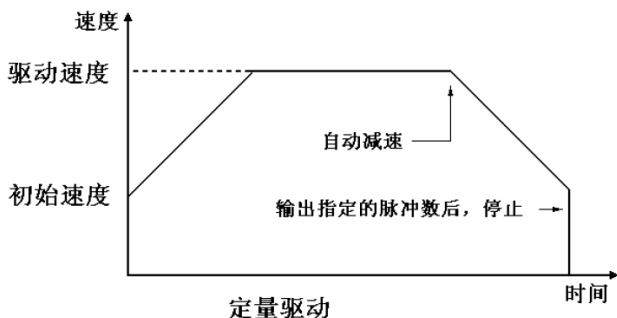


图 4-1

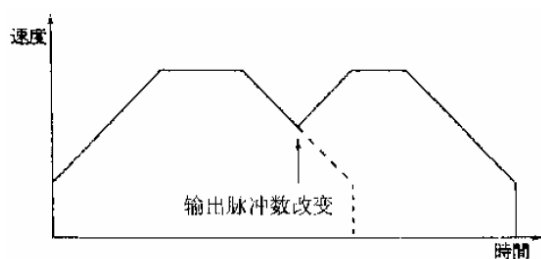
进行加/减速的定量驱动需要设定下列参数：

- 范围 R
- 加/减速 A/D
- 初始速度 SV
- 驱动速度 V
- 输出脉冲数 P

#### ■ 驱动中变更输出脉冲数

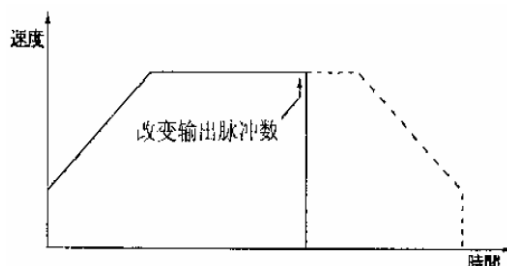
定量驱动中可以变更输出脉冲数。加/减速驱动中，开始减速时，如果输出脉冲数有变更的话，重新开始加速（图 4-2）。此外，如果变更的输出脉冲数比已经输出的脉冲数要少的话，立即停止。（图 4-3）

S 曲线减速时脉冲数有变更的话，不能正确地运行 S 曲线动作。



减速时的输入脉冲数变更

图 4-2



要改变的脉冲数比已输出的脉冲数少的情况

图 4-3

#### ■ 加/减速定量驱动的手动减速

加/减速定量驱动是一般如图 4-1 所示，从芯片计算的减速点开始自动减速。此外，也可以用手动减速。在直线加/减速定量驱动中，需要经常变更速度的情况下，芯片不能正确地计算自动减速点或无法算出此减速点，所以需要手动地计算减速点。可通过写寄存器 R2 来设定手动减速点。

先设定寄存器 R2 的值，在运行过程中当脉冲输出计数器（计数器 C）的值等于或小于寄存器 R2 的值，就开始减速。其它操作和一般的定量驱动一样。

■ 加/减速定量驱动的加速计数器偏移

对于加/减速定量驱动，加速时，用加速计数器计算加速中累计的脉冲数。当设定的脉冲输出剩余数少于加速计数器的值时，就开始减速。减速时，输出的脉冲数和加速中累计的脉冲数一致时，加速计数器偏移是在加速计数器上加上指定的偏移数值。如图 4-4 所示，设定偏移数值越大，自动减数点越往前移动。因此，减速完毕的初始速度要延迟。

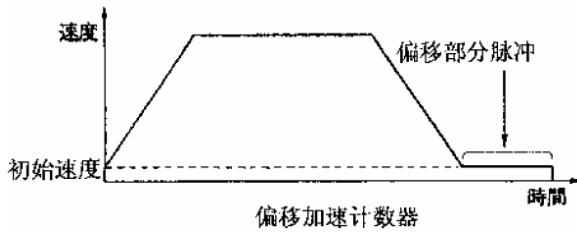


图 4-4

运行一般的直线加/减速驱动几乎不需要重新设定这个参数。S 曲线加/减速定量驱动中，如果驱动完毕速度降不到初始速度的话，要把加速计数器偏移数值设定为适当的数值，以修正它的速度。

## 二、连续驱动输出

在连续驱动中，连续输出驱动脉冲直至高位的停止命令或外部的停止信号有效。需要运行原点搜寻、扫描操作、控制马达旋转速度时，使用此功能。

有两种停止命令，一个是减速停止，另一个是立即停止。可以通过写入命令 30h 来设置立即停止或命令 31h 设置减速停止，也可以设定每个轴都有的用于减速/立即停止的外部信号（ $\overline{\text{SLDP}}$ 、 $\overline{\text{SLDM}}$ ）的逻辑方向（正/负逻辑）、信号模式（电平/边沿信号）来控制。

## 三、脉冲输出类型

通过软件程序的设置，每个轴的驱动信号可设为两种输出方式之一：

- 独立 2 脉冲方式(CW/CCW 方式)：正向脉冲(CW)通过  $\overline{\text{POUT}}$  输出，负方向脉冲(CCW)通过  $\overline{\text{PDIR}}$  输出。脉冲信号有上升沿有效或下降沿有效的选择
- 脉冲/方向方式(PULSE/DIR 方式)：正负 2 个方向的驱动脉冲都通过输出信号  $\overline{\text{POUT}}$  输出，脉冲方向通过  $\overline{\text{PDIR}}$  输出。

脉冲输出方向可以设置为正向脉冲，也可以设置为负向脉冲；方向信号可以设置为低电平正转高电平，也可以设置为高电平正转低电平。

脉冲输出类型表

脉冲输出方式	驱动方向	输出脉冲波形			
		CW (正)	CCW (负)	PULSE (脉冲)	DIR (方向)
CW/CCW 方式 (独立 2 脉冲方式)	正驱动方向	脉冲	高电平	高电平	脉冲
	负驱动方向	高电平	脉冲	脉冲	高电平
PULSE/DIR 方式 (1 脉冲方式)	正驱动方向	脉冲	脉冲	高电平	低电平
	负驱动方向	脉冲	脉冲	低电平	高电平

## 第二节、速度曲线

各轴的驱动脉冲输出一般使用正/负方向的定量驱动或连续驱动命令。此外，以设定模式或设定参数来产生定速、直线加/减速、S 曲线加/减速的速度曲线。

### 一、定速驱动

定速驱动就是以一成不变的速度输出驱动脉冲及初始速度等于驱动速度。如果设定驱动速度小于初始速度，



就没有加/减速驱动, 而是定速驱动。使用搜寻原点、编码器 Z 相等信号时, 找到信号后马上要立即停止的话, 不必进行加/减速驱动, 而是一开始就运行低速的定速驱动。如图 4-5。

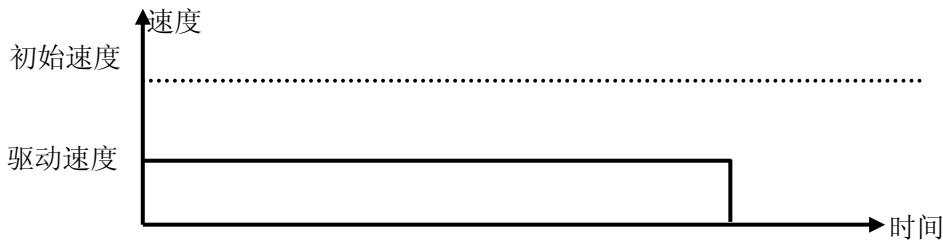


图 4-5 定速驱动

### 二、直线加/减速驱动

直性加/减速驱动是线性地从驱动开始的初始速度加速到指定的驱动速度。当加速度和减速度设置一样时, 速度时间图就是对称的梯形, 当加速度和减速度不一样时, 就不是对称的。

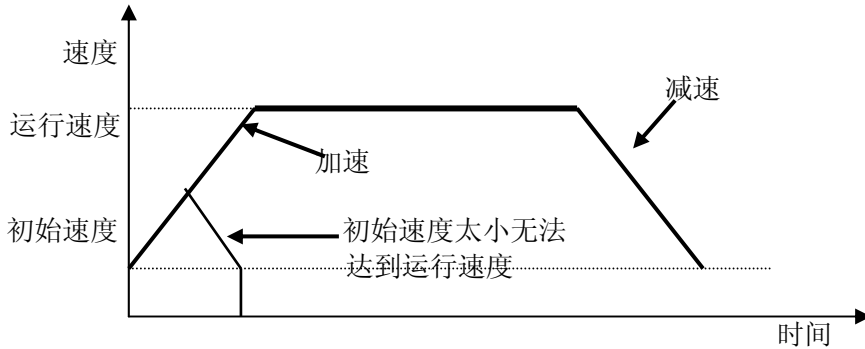


图 4-6 直线加/减速

### 三、S 曲线加/减速驱动

驱动速度加/减速时, PCI1040 线性地增加/减少加速度/减速度, 以产生 S 型速度曲线。S 曲线加/减速驱动如图 4-7 所示。

驱动开始加速时, 加速度以指定的加速度的增加率从 0 线性增加到指定的加速度值 (Acceleration), 因此, 这个速度曲线成为二次抛物线 (a 区间)。加速度达到指定数值 (Acceleration) 后保持此数值。这时速度曲线是直线型的, 速度在加速中 (b 区间)。目标速度和当前速度的差值比相应时间增加所增加的速度少时, 加速度开始变小趋向于 0。加速度的减速率和加速率一样是线性变化的, 因此, 速度曲线形成了二次抛物线 (c 区间)。

当减速时和加速时一样, 减速度以指定的增加率增大到减速度值, 然后减速度保持一段时间不变, 最后减速度减少直到 0 (d、e、f 区间)。这样具有部分固定加速度的加速为部分 S 曲线。

另一方面, 在 a 区间若在加速度达到指定数值 (Acceleration) 前, 目标速度 (DriveSpeed) 和当前速度的差值比相应时间增加所增加的速度少时, b 区间就消失, 只有 a 和 c 区间。这种没有固定加速度的加速称为完全 S 曲线加速。

同样, S 曲线加/减速运动也分为对称的和不对称的运动两种。

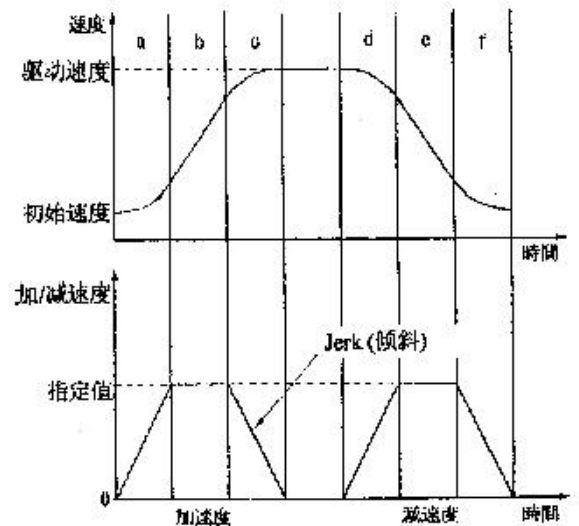


图 4-7 S 曲线加减速驱动

## 第三节、插补

PCI1040 能用轴 1~轴 4 实现 2 轴/3 轴/4 轴直线插补。在插补驱动过程中, 插补运算是在指定主轴 (ax1) 的基本脉冲时序下运行的。因此, 进行插补命令之前, 先要设定指定 ax1 的速度、加速度、终点位置 (脉冲数) 等参数。主轴由 ax1 指定, 在直线插补时并不一定是长轴。

设定相对于当前位置的终点坐标，写入 2 轴或 3 轴或 4 轴直线插补命令后，就开始进行线性插补。以 2 轴直线插补为例，如图 4-8 所示，从当前坐标至终点坐标执行直线插补，终点坐标由针对当前位置的相对数值设定。精确设定每个轴的输出脉冲数。在每个轴独立运行时，输出脉冲数设定为没有符号的数值。但是，在插补驱动时，用相对数值设定对当前位置的终点坐标。

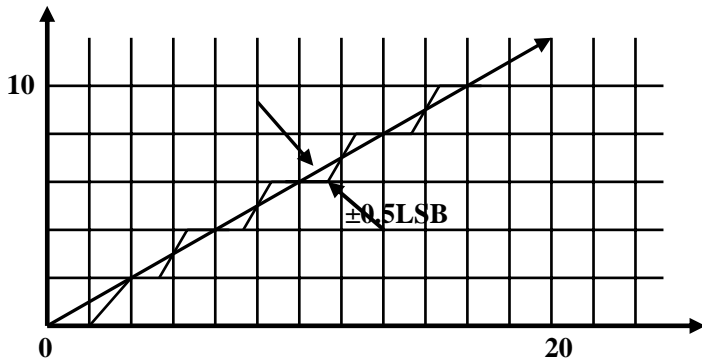


图 4-8

## 第四节、信号输入/输出

### 一、通用输入/输出

PCI1040 有 8 路通用信号输入和 8 路通用信号输出。 $\overline{IN0} \sim \overline{IN7}$  是 8 路输入管脚， $\overline{OUT0} \sim \overline{OUT7}$  是 8 路输出管脚，可输出+24V 电压。

### 二、传感器输出

PCI1040 有 8 路伺服传感器驱动信号输出  $\overline{SON1} \sim \overline{SON8}$ ，也能作为通用输出信号。

### 三、清除伺服驱动计数器输出

PCI1040 有 8 路清除伺服驱动计数器信号输出  $\overline{CLR1} \sim \overline{CLR8}$ ，也能作为通用输出信号。

清除伺服驱动计数器可以用单脉冲输出信号或通用输出信号来实现，这两种输出方式可以通过设置输出寄存器的 bit0 位来切换，当 bit0=0 时，是单脉冲输出信号，信号频率是参考时钟的 1/32；当 bit0=1 时，是通用输出信号，可输出高低电平。

## 第五节、同步控制

$\overline{SYNC}$  管脚是同步输入信号管脚，若要设置几轴同步，需要把这几轴的条件都设置一样，并选中“外部同步启动”，再启动几轴的电机运动，当  $\overline{SYNC}$  信号由高变低时，就输出脉冲。

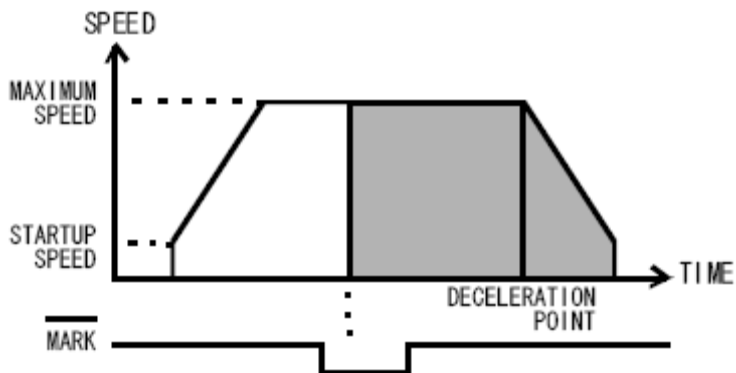
## 第六节、外部启动和原点搜寻

### 一、外部启动

设置外部启动后，当  $\overline{MARK}$  输入端有效时，传感器位置驱动就开始输出脉冲。输入灵敏度是参考时钟周期的 1 倍或者 16 倍。 $\overline{MARK}$  输入端可设置为正逻辑或负逻辑信号输入。

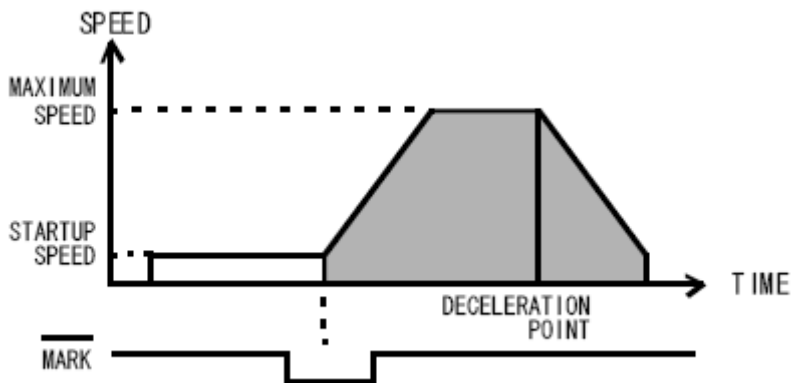
#### (一)、传感器位置驱动类型 I

当  $\overline{MARK}$ （传感器定位开始信号）输入端有效时，传感器位置驱动开始，即正在输出脉冲。命令写入后，加速过程就启动。如下图所示：



(二)、传感器位置驱动类型 II

当  $\overline{\text{MARK}}$  输入端有效时，传感器位置驱动开始，即正在输出脉冲。加速过程启动也是发生在  $\overline{\text{MARK}}$  输入端有效时。



(三)、传感器位置驱动类型 III

当  $\overline{\text{MARK}}$  输入端有效时，位置驱动开始。此种方式下，没有加减速运动过程。



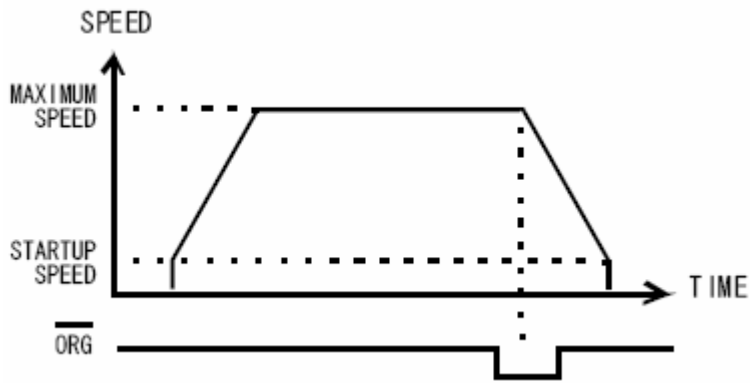
二、返回原点

设置返回原点模式时，操作是基于  $\overline{\text{ORG}}$  输入信号单独作用或者是  $\overline{\text{ORG}}$  与  $\overline{\text{EZ}}$  输入信号共同作用下实现的。输入灵敏度是参考时钟周期的 1 倍或者 16 倍。

有四种驱动类型的选择，可对原点逻辑方向和编码器逻辑方向进行选择。

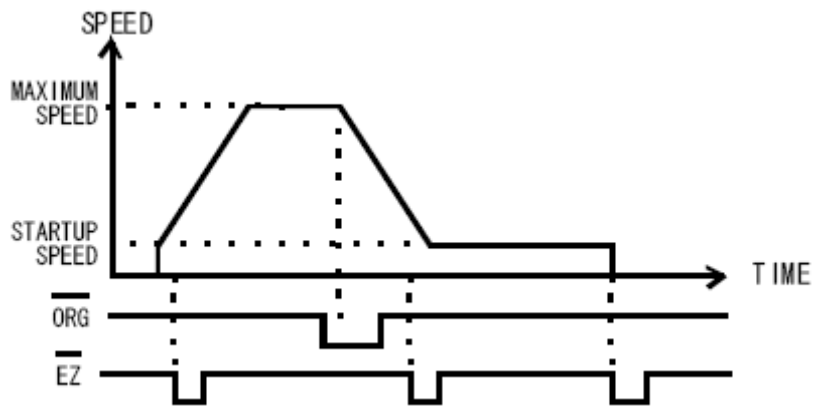
(一)、返回原点位置驱动类型 I

自动原点搜索带加减速过程。当  $\overline{\text{ORG}}$  有效时减速开始和运动结束。



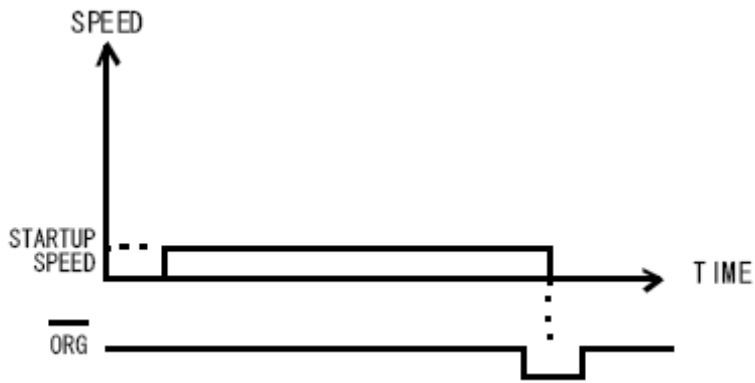
(二)、返回原点位置驱动类型 II

自动原点搜索带加减速过程。当  $\overline{\text{ORG}}$  有效时减速开始，在到达起始速度之后， $\overline{\text{EZ}}$  有效时运动结束。（没有加  $\overline{\text{ORG}}$  信号前的  $\overline{\text{EZ}}$  信号是不起作用的）



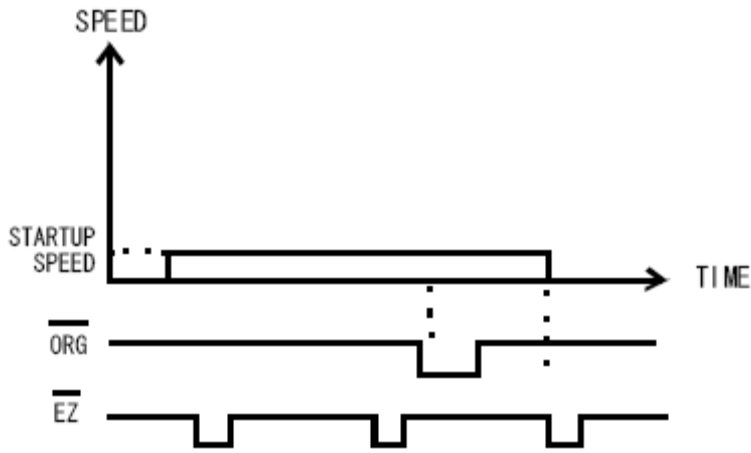
(三)、返回原点位置驱动类型 III

自动原点搜索以起始速度常速运行。当  $\overline{\text{ORG}}$  有效时减速开始，运动立即结束。



(四)、返回原点位置驱动类型 IV

自动原点搜索以起始常速运行。当  $\overline{\text{ORG}}$  有效之后， $\overline{\text{EZ}}$  变为有效时，运动立即结束。



## 第七节、中断功能

PCI1040 的中断位设置有 14 种情况，分别有 14 个指示灯对应表示中断位的选择情况：当选择了相应的中断位时，对应位的指示灯则亮，反之则不亮。

14 位中断标志位分别是：

- 1、脉冲输出完成中断标志
- 2、错误停止中断标志
- 3、减速开始点中断标志
- 4、最大加速度中断标志
- 5、计数器 A 进位中断标志
- 6、计数器 A 借位中断标志
- 7、计数器 B 进位中断标志
- 8、计数器 B 借位中断标志
- 9、 $\overline{\text{ORG}}$  中断标志
- 10、 $\overline{\text{EZ}}$  中断标志
- 11、 $\overline{\text{INO}}$  中断标志
- 12、 $\overline{\text{MARK}}$  中断标志
- 13、P = Q 中断标志
- 14、P > Q 中断标志

说明：

1、counterA 进位中断，counterB 进位中断测试，范围±2147483648，计数值达到 2147483648 时，会产生进位中断信号，并继续从-2147483648 计数，即在±2147483648 循环计数。

2、最大加速度中断：选择 S 曲线运动，定长脉冲方式，当所给的加速度达到最大时 counterA，counterB 借位中断，用电机反转即可测试，从-1 到 0 值时，会产生借位中断信号。

3、ORG 中断，EZ 中断，INO 中断，MARK 中断，即给该管脚一个低电平信号，即可产生中断。

4、P=Q 中断：把寄存器 P 和寄存器 Q 设置成一样；P>Q 把 Q 设置成 Comparator reg 并在 Comparator 上设置数值，当该轴电机运动时产生脉冲大于寄存器值的时候产生中断。

其中，P、Q 是寄存器，它们的值可以是计数器 A 或计数器 B 或者是另外单独设置的比较寄存器 (Comparator) 的值，比较寄存器 (Comparator) 的取值范围是-8,388,608 ~ +8,388,607。

## 第八节、硬件限位功能

### 一、越限信号

硬件限制信号  $\overline{\text{ELP}}$  和  $\overline{\text{ELM}}$  是正方向和负方向驱动脉冲的的限位信号输入。



可软件选择正逻辑/负逻辑越限信号。当限制信号的逻辑电平和限制信号有效时，可以由命令选择减速停止或立即停止。

电机正转，越限+正方向的越限信号，在驱动脉冲输出过程中此信号有效变化，就执行减速停止或立即停止。电机反转，越限-负方向的越限信号，在驱动脉冲输出过程中此信号有效变化，就执行减速停止或立即停止。

## 二、伺服电机紧急停止报警信号

$\overline{\text{ALM1}} \sim \overline{\text{ALM8}}$  输入信号接受从伺服马达驱动器的紧急停止警报信号。可软件选择输入逻辑状态为正逻辑或负逻辑。每轴都有一个  $\overline{\text{ALM}}$  信号管脚，当运动轴接收到  $\overline{\text{ALM}}$  信号时，将立即停止发出脉冲。

## 三、伺服电机到位信号

伺服马达定位完成输入信号。是伺服驱动器发出给控制卡的状态信号，当伺服电机定位完成时，伺服驱动器自动将该信号置为有效。可软件选择输入逻辑状态为正逻辑或负逻辑。

## 四、输入信号滤波器

输入信号滤波器的值决定了  $\overline{\text{ELP}}$ 、 $\overline{\text{ELM}}$ 、 $\overline{\text{ALM}}$ 、 $\overline{\text{SLDP}}$ 、 $\overline{\text{SLDM}}$  的灵敏度。灵敏度的取值范围为 1~256，可设置的值为 0~255，当灵敏度=0 时，表示 256。

计算公式： $16 * \text{灵敏度} \div 16384.0 = \text{信号延迟时间 (ms)}$

## 五、减速停止信号

$\overline{\text{SLDP}}$ 、 $\overline{\text{SLDM}}$  管脚分别为+方向、-方向的减速信号输入端，可软件选择输入信号的逻辑方向（正逻辑/负逻辑）、信号模式（电平信号/边沿信号）、运动方式（减速/减速停止）。

## 第九节、位置计数

PCI1040 的每一个轴都有一个逻辑位置计数器和实际位置计数器。

### 一、逻辑位置计数器

逻辑位置计数器（Counter A）是一个 32 位正负计数器，对控制卡输出的正/负方向脉冲进行计数。当输出一个正向脉冲后，计数器自动加 1；当输出一个负向脉冲后，计数器自动减 1。

$\overline{\text{CLRA1}} \sim \overline{\text{CLRA8}}$  信号低有效时，可以将对应轴的逻辑位置计数器清 0。

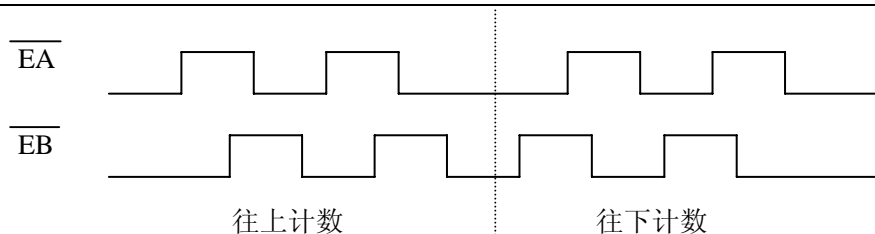
### 二、实际位置计数器

实际位置计数器也是一个 32 位正负计数器，计数来自外部编码器的输入脉冲  $\overline{\text{EA}}$ 、 $\overline{\text{EB}}$ ，输入信号可以设定为 2 相脉冲输入模式或下脉冲输入模式。

#### ■ 2 相脉冲输入模式

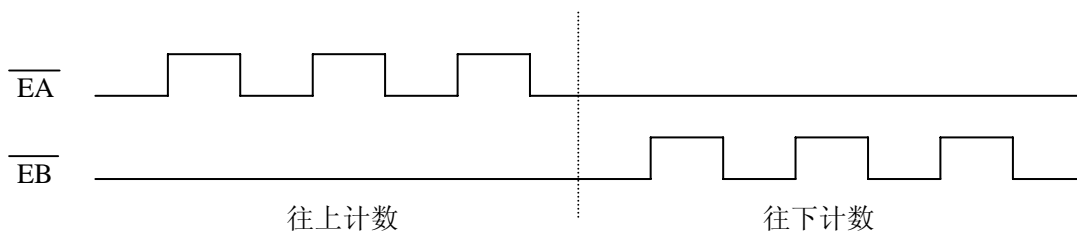
这种模式下，2 路相位差为 90 度的脉冲输入信号输入。按正逻辑当 A 相超前（ $\overline{\text{EA}}$  脉冲信号“超前” $\overline{\text{EB}}$  脉冲信号 90 度）时，加计数；当 B 相超前（ $\overline{\text{EA}}$  信号“滞后” $\overline{\text{EB}}$  信号）时，减计数。由 2 个信号的上升沿和下降沿进行向上/下计数。而且，用户可选用编码器 4、2、1 倍计数模式进行计数。

例如，4 倍计数时：若为正向计数，实际位置计数器的值为 EA 反馈脉冲数的 4 倍；若为负向计数，实际位置计数器的值为 EB 反馈脉冲数的 4 倍。其他倍数的情况类似。



■ 下脉冲输入模式

EA 作为向上计数输入，EB 作为向下计数输入，由脉冲的上升沿计数。



PCI1040 可以在任何时候写入或读出 2 个计数器的数据，计数范围在-2,147,483,648~+2,147,483,647 之间。

三、比较寄存器

PCI1040 有 2 个 24 位寄存器（寄存器 COMP P 和 COMP Q）用来与逻辑位置计数器或实际位置计数器进行比较。另有一个比较寄存器（Comparator），可以设置为-8,388,608~+8,388,607 之间的计数值。因此，COMP P 和 COMP Q 可以由程序选择设定为逻辑计数器（Counter A）或实位计数器（Counter B）或比较寄存器。当 P > Q 或者 P = Q 时，可以选择相应的中断位有效或无效。

## 第三章 驱动函数库

### 第一节、驱动函数库函数列表

(每个函数省略了前缀“PCI1040\_”)

函数名	函数功能
1、设备对象管理函	
<a href="#">CreateDevice</a>	创建句柄
<a href="#">GetDeviceCount</a>	获得设备总数
<a href="#">ReleaseDevice</a>	释放设备
<a href="#">Reset</a>	复位整个 PCI 设备
2、设置控制卡的基本参数函数	
<a href="#">PulseOutMode</a>	设置脉冲输出模式
<a href="#">SetR</a>	设置倍率(0.1-250)
<a href="#">SetA</a>	设置加速度(1-16383)
<a href="#">SetDec</a>	设置减速度(1-16383)
<a href="#">SetAccOrDecRate</a>	S 曲线加减速段设置
<a href="#">SetSV</a>	设置初始速度 (1-16383)
<a href="#">SetV</a>	设置驱动速度 (1-16383)
<a href="#">SetP</a>	设置定长脉冲数(0-16777215)
3、设置编码器输入信号类型	
<a href="#">SetEncoderDivRatio</a>	设置编码器输入信号类型
4、直线 S 曲线初始化、启动函	
<a href="#">InitLVDVCHV</a>	初始化连续, 定长脉冲驱动(带加减速)
<a href="#">StartLVDVCHV</a>	启动连续, 定长脉冲驱动(带加减速)
<a href="#">Start8DCHV</a>	启动 8 轴同时驱动(带加减速)
<a href="#">InitLVDVCNST</a>	初始化连续, 定长脉冲驱动(常速)
<a href="#">StartLVDVCNST</a>	启动连续,定长脉冲驱动(常速)
<a href="#">Start8DCNST</a>	8 轴同时启动(常速)
5、任意 2 轴直线插补初始化、启动函数	
<a href="#">InitLineInterpolation_2D</a>	初始化任意 2 轴直线插补驱动
<a href="#">StartLineInterpolation_2D</a>	启动任意 2 轴直线插补驱动
6、任意 3 轴直线插补初始化、启动函数	
<a href="#">InitLineInterpolation_3D</a>	初始化任意 3 轴直线插补驱动
<a href="#">StartLineInterpolation_3D</a>	启动任意 3 轴直线插补驱动
7、任意 4 轴直线插补初始化、启动函数	
<a href="#">InitLineInterpolation_4D</a>	初始化任意 4 轴直线插补运动
<a href="#">StartLineInterpolation_4D</a>	启动任意 4 轴直线插补运动
8、插补相关函	
<a href="#">InterpolationEnable</a>	线性插补基设置 (最大脉冲数设置)
<a href="#">SetMaxPulse</a>	将插补轴中的最大脉冲数设置到每一个轴的 R8 寄存器中
9、计数器相关函数	
<a href="#">SetCounterA</a>	设置 CounterA 计数器
<a href="#">SetValCounterA</a>	设置 CounterA 计数器值
<a href="#">ClearCounterA</a>	CounterA 计数器清零
<a href="#">ReadCounterA</a>	读取 CounterA 的计数值
<a href="#">SetCounterB</a>	设置 CounterB 计数器
<a href="#">SetValCounterB</a>	设置 CounterB 计数器值
<a href="#">ClearCounterB</a>	CounterB 计数器清零



<a href="#">ReadCounterB</a>	读取 CounterB 的计数值
<a href="#">SetInCLRAMode</a>	设置外部输入信号模式, 可以外部信号对 CounterA 计数器清零
<a href="#">SetCOMPP</a>	设置 COMP P 寄存器
<a href="#">SetCOMPQ</a>	设置 COMP Q 寄存器
<a href="#">ReadCOMPRegVal</a>	读取比较寄存器值
<a href="#">SetCOMPRegVal</a>	设置比较寄存器值
<a href="#">SetWithoutPulse</a>	设置没有脉冲输出
<a href="#">SetCSWithoutPulse</a>	设置固定速度并且没有脉冲输出
10、减速函数设	
<a href="#">DecMode</a>	减速模式
<a href="#">SetDecStartPoint</a>	减速点设置
<a href="#">DecValid</a>	减速有效
<a href="#">DecInvalid</a>	减速无效
<a href="#">DecStop</a>	减速停止
<a href="#">InstStop</a>	立即停止
<a href="#">InstStopAll</a>	立即全部停止
11、同步相关函数	
<a href="#">SetSynchronAction</a>	设置同步相关位
<a href="#">SynchronActionDisable</a>	设置同步位无效
12、设置原点搜寻外部信号停止电机驱动输入信号	
<a href="#">SetOutORG</a>	设置返回原点位置驱动类型的输入信号
13、外部信号启动电机定长驱动、连续驱动	
<a href="#">SetOutEnableDrv</a>	设置外部使能定量驱动
14、设置外部信号启动电机驱动的输入信号	
<a href="#">SetOutMARK</a>	设置传感器位置驱动类型的输入信号
15、初始化外部输入信号	
<a href="#">SetSLDPFun</a>	设置外部 SLD 信号
<a href="#">SetSLDMFun</a>	设置外部 SLD 信号
<a href="#">SetELPFun</a>	设置外部 ELP 信号
<a href="#">SetELMFun</a>	设置外部 ELM 信号
<a href="#">SetINPFun</a>	设置外部 INP 信号
<a href="#">SetALMFun</a>	设置外部 ALM 信号
<a href="#">SetSensitive</a>	设置输入滤波初始化(灵敏度)同时对 EL,ALM,SLD 有效
16、设置常速有效和无效	
<a href="#">CNSTValid</a>	常速有效
<a href="#">CNSTInvalid</a>	常速无效
17、设置通用输入和通用输出	
<a href="#">SetCLROutput</a>	设置 CLR 通用输出
<a href="#">SetSONOutput</a>	设置 SON 传感器通用输出
<a href="#">SetDeviceDO</a>	设置通用输出
<a href="#">GetDeviceDI</a>	获取通用输入状态
18、读当前速	
<a href="#">ReadCV</a>	读当前速度
19、读状态寄存器的位状态	
<a href="#">GetRunStatus</a>	获得运行状态寄存器的位状态
<a href="#">GetSensorStatus</a>	获得传感器状态寄存器的位状态
<a href="#">GetComparatorStatus</a>	获得比较器状态寄存器的位状态
<a href="#">GetINTFlag</a>	获得中断标志寄存器的位状态
<a href="#">GetMainStatus</a>	获得主要状态寄存器的状态
<a href="#">GetNSFactorStatus</a>	获得正常停止原因寄存器状态
<a href="#">GetESFactorStatus</a>	获得错误停止原因寄存器状态



20、中断位设置、插补中断状态清除	
<a href="#">SetInterruptBit</a>	设置中断位
<a href="#">GetInterruptAxis</a>	取得 8 个轴中断状态
<a href="#">INTFlagClear</a>	中断标志寄存器清零

## 第二节、驱动函数库说明

### 一、设备对象管理函数

#### 1. HANDLE CreateDevice( int DeviceID)

**功能：** 该函数负责创建 PCI 设备对象，并返回其设备对象句柄。

**参数：** DeviceID 设备 ID( Identifier )标识号。当向同一个 Windows 系统中加入若干相同类型的 PCI 设备时，我们的驱动程序将以该设备的“基本名称”与 DeviceID 标识值为名称后缀的标识符来确认和管理该设备。比如若用户往 Windows 系统中加入第一个 PCI1040 模板时，驱动程序则以“PCI1040”作为基本名称，再以 DeviceID 的初值组合成该设备的标识符“PCI1040-0”来确认和管理这第一个设备，若用户接着再添加第二个 PCI1040 模板时，则系统将以“PCI1040-1”来确认和管理第二个设备，若再添加，则以此类推。所以当用户要创建设备句柄管理和操作第一个 PCI 设备时，DeviceID 应置 0，第二个应置 1，也以此类推。但默认值为 0。

**返回值：** 如果执行成功，则返回设备对象句柄；如果没有成功，则返回错误码 INVALID\_HANDLE\_VALUE。由于此函数已带容错处理，即若出错，它会自动弹出一个对话框告诉您出错的原因。您只需要对此函数的返回值作一个条件处理即可，别的任何事情您都不必做。

**相关函数：** [ReleaseDevice](#)

#### Visual C++ & C++Builder 程序举例

```

:
HANDLE hDevice; // 定义设备对象句柄
hDevice=CreateDevice ( 0 );// 创建设备对象,并取得设备对象句柄
if(hDevice==INVALID_HANDLE_VALUE);// 判断设备对象句柄是否有效
{
    return; // 退出该函数
}
)

```

#### 2. int GetDeviceCount(HANDLE hDevice)

**功能：** 取得 PCI1040 设备的数量。

**参数：** hDevice 设备对象句柄，它应由 CreateDevice 创建。

**返回值：** 返回系统中 PCI1040 的数量。

#### 3. BOOL ReleaseDevice(HANDLE hDevice)

**功能：** 释放设备对象所占用的系统资源及设备对象自身。

**参数：** hDevice 设备对象句柄，它应由 CreateDevice 创建。

**返回值：** 若成功，则返回 TRUE， 否则返回 FALSE。

**相关函数：** [CreateDevice](#)

应注意的是，CreateDevice 必须和 ReleaseDevice 函数一一对应，即当您执行了一次 CreateDevice 后，再一次执行这些函数前，必须执行一次 ReleaseDevice 函数，以释放由 CreateDevice 占用的系统软硬件资源，如 DMA 控制器，系统内存等。只有这样，当您再次调用 CreateDevice 函数时，那些软硬件资源才可被再次使用。

#### 4. BOOL Reset(HANDLE hDevice, LONG AxisNum)

**功能：** 复位整个 PCI 设备。

**参数：**

<b>hDevice</b>	设备对象句柄, 它应由 CreateDevice 创建。
<b>AxisNum</b>	轴号。 PCI1040_AXIS1: 1 轴 PCI1040_AXIS2: 2 轴 PCI1040_AXIS3: 3 轴 PCI1040_AXIS4: 4 轴 PCI1040_AXIS5: 5 轴 PCI1040_AXIS6: 6 轴 PCI1040_AXIS7: 7 轴 PCI1040_AXIS8: 8 轴 PCI1040_AXISALL: 1、2、3、4、5、6、7、8 所有轴

**返回值:** 若成功, 则返回 TRUE, 否则返回 FALSE。

**相关函数:** [CreateDevice](#)

## 二、设置控制卡的基本参数函数

### 1. **BOOL PulseOutMode (HANDLE hDevice, LONG Mode, LONG PulseLogLevel, LONG DirLogLevel, LONG Direction, LONG AxisNum)**

**功能:** 设置脉冲输出模式。

**参数:**

<b>hDevice</b>	设备对象句柄, 它应由 CreateDevice 创建。
<b>Mode</b>	脉冲输出模式。PCI1040_CWCCW: CW/CCW 模式, PCI1040_CPDIR: CP/DIR 模式。
<b>PulseLogLevel</b>	脉冲信号电平(0:正向脉冲, 1:负向脉冲)。
<b>DirLogLevel</b>	方向信号电平(0:低电平正转高电平反转, 1:高电平正转低电平反转)。
<b>Direction</b>	运动方向。PCI1040_PDIRECTION: 正方向; PCI1040_MDIRECTION: 反方向。
<b>AxisNum</b>	轴号。 PCI1040_AXIS1: 1 轴 PCI1040_AXIS2: 2 轴 PCI1040_AXIS3: 3 轴 PCI1040_AXIS4: 4 轴 PCI1040_AXIS5: 5 轴 PCI1040_AXIS6: 6 轴 PCI1040_AXIS7: 7 轴 PCI1040_AXIS8: 8 轴 PCI1040_AXISALL: 1、2、3、4、5、6、7、8 所有轴

**返回值:** 若成功, 则返回 TRUE, 否则返回 FALSE。

### 2. **BOOL SetR (HANDLE hDevice, LONG Data, LONG AxisNum)**

**功能:** 设置倍率, 范围是(0.1-250)。

**参数:**

<b>hDevice</b>	设备对象句柄, 它应由 CreateDevice 创建。
<b>Data</b>	倍率值(0.1-250)。
<b>AxisNum</b>	轴号。 PCI1040_AXIS1: 1 轴



PCI1040\_AXIS2: 2 轴  
 PCI1040\_AXIS3: 3 轴  
 PCI1040\_AXIS4: 4 轴  
 PCI1040\_AXIS5: 5 轴  
 PCI1040\_AXIS6: 6 轴  
 PCI1040\_AXIS7: 7 轴  
 PCI1040\_AXIS8: 8 轴  
 PCI1040\_AXISALL: 1、2、3、4、5、6、7、8 所有轴

**返回值:** 若成功, 则返回 TRUE, 否则返回 FALSE。

**备注:** 电机运行时的实际加速度 (PPS/SEC) = 设定的加速度 × 倍率。

### 3. BOOL SetA( HANDLE hDevice, LONG Mode, LONG Data, LONG AxisNum)

**功能:** 设置指定轴加速度(Set acceleration), 范围(125~511,968,750)。

**参数:**

hDevice	设备对象句柄, 它应由 CreateDevice 创建。
Mode	线性运动模式和插补运动模式 (=0 为线性运动, =1 为插补运动)。
Data	加速度, 范围(1-16383)。
AxisNum	轴号。

PCI1040\_AXIS1: 1 轴  
 PCI1040\_AXIS2: 2 轴  
 PCI1040\_AXIS3: 3 轴  
 PCI1040\_AXIS4: 4 轴  
 PCI1040\_AXIS5: 5 轴  
 PCI1040\_AXIS6: 6 轴  
 PCI1040\_AXIS7: 7 轴  
 PCI1040\_AXIS8: 8 轴  
 PCI1040\_AXISALL: 1、2、3、4、5、6、7、8 所有轴

**返回值:** 若成功, 则返回 TRUE, 否则返回 FALSE。

**备注:** 电机运行时的实际加速度 (PPS/SEC) = 设定的加速度 × 倍率。

### 4. BOOL SetDec(HANDLE hDevice, LONG Mode, LONG Data, LONG AxisNum)

**功能:** 设置指定轴减速度(Set deceleration), 范围(125~ 511,968,750)。

**参数:**

hDevice	设备对象句柄, 它应由 CreateDevice 创建。
Mode	线性运动模式和插补运动模式 (=0 为线性运动, =1 为插补运动)。
Data	减速度值(1-16383)。
AxisNum	轴号。

PCI1040\_AXIS1: 1 轴  
 PCI1040\_AXIS2: 2 轴  
 PCI1040\_AXIS3: 3 轴  
 PCI1040\_AXIS4: 4 轴  
 PCI1040\_AXIS5: 5 轴  
 PCI1040\_AXIS6: 6 轴  
 PCI1040\_AXIS7: 7 轴  
 PCI1040\_AXIS8: 8 轴

PCI1040\_AXISALL: 1、2、3、4、5、6、7、8 所有轴

**返回值:** 若成功, 则返回 TRUE, 否则返回 FALSE。

**备注:** 电机运行时的实际减速度 (PPS/SEC) = 设定的减速度 × 倍率。

### 5. BOOL SetAccOrDecRate (HANDLE hDevice, LONG Data, LONG AxisNum)

**功能:** S 曲线加减速段设置。

**参数:** hDevice 设备对象句柄, 它应由 CreateDevice 创建。

Data 数据 (1-8191)。

AxisNum 轴号。

PCI1040\_AXIS1: 1 轴

PCI1040\_AXIS2: 2 轴

PCI1040\_AXIS3: 3 轴

PCI1040\_AXIS4: 4 轴

PCI1040\_AXIS5: 5 轴

PCI1040\_AXIS6: 6 轴

PCI1040\_AXIS7: 7 轴

PCI1040\_AXIS8: 8 轴

PCI1040\_AXISALL: 1、2、3、4、5、6、7、8 所有轴

**返回值:** 若成功, 则返回 TRUE, 否则返回 FALSE。

### 6. BOOL SetSV(HANDLE hDevice, LONG Mode, LONG Data, LONG AxisNum)

**功能:** 设置指定轴初始速度(Set start speed), 范围(1-16383)。

**参数:** hDevice 设备对象句柄, 它应由 CreateDevice 创建。

Mode 线性运动模式和插补运动模式 (=0 为线性运动, =1 为插补运动)。

Data 初始速度数据, 范围(1-16383)。

AxisNum 轴号。

PCI1040\_AXIS1: 1 轴

PCI1040\_AXIS2: 2 轴

PCI1040\_AXIS3: 3 轴

PCI1040\_AXIS4: 4 轴

PCI1040\_AXIS5: 5 轴

PCI1040\_AXIS6: 6 轴

PCI1040\_AXIS7: 7 轴

PCI1040\_AXIS8: 8 轴

PCI1040\_AXISALL: 1、2、3、4、5、6、7、8 所有轴

**返回值:** 若成功, 则返回 TRUE, 否则返回 FALSE。

**备注:** 电机运行时的实际初始速度 (PPS) = 设定的初始速度 × 倍率。

### 7. BOOL SetV( HANDLE hDevice, LONG Mode, LONG Data, LONG AxisNum)

**功能:** 设置指定轴驱动速度(Set drive speed), 范围(1-16383)。

**参数:** hDevice 设备对象句柄, 它应由 CreateDevice 创建。

Mode 线性运动模式和插补运动模式 (=0 为线性运动, =1 为插补运动)。



**Data** 驱动速度数据，范围(1-16383)。

**AxisNum** 轴号。

PCI1040\_AXIS1: 1 轴  
 PCI1040\_AXIS2: 2 轴  
 PCI1040\_AXIS3: 3 轴  
 PCI1040\_AXIS4: 4 轴  
 PCI1040\_AXIS5: 5 轴  
 PCI1040\_AXIS6: 6 轴  
 PCI1040\_AXIS7: 7 轴  
 PCI1040\_AXIS8: 8 轴  
 PCI1040\_AXISALL: 1、2、3、4、5、6、7、8 所有轴

**返回值:** 若成功，则返回 TRUE，否则返回 FALSE。

**备注:** 电机运行时的实际驱动速度 (PPS) = 设定的驱动速度 × 倍率。

### 8. BOOL SetP(HANDLE hDevice, LONG Data, LONG AxisNum)

**功能:** 设置指定轴定长脉冲数(Set pulse)，范围(0-16777215)。

**参数:** **hDevice** 设备对象句柄，它应由 CreateDevice 创建。

**Data** 定长脉冲数数据，范围(0-16777215)。

**AxisNum** 轴号。

PCI1040\_AXIS1: 1 轴  
 PCI1040\_AXIS2: 2 轴  
 PCI1040\_AXIS3: 3 轴  
 PCI1040\_AXIS4: 4 轴  
 PCI1040\_AXIS5: 5 轴  
 PCI1040\_AXIS6: 6 轴  
 PCI1040\_AXIS7: 7 轴  
 PCI1040\_AXIS8: 8 轴  
 PCI1040\_AXISALL: 1、2、3、4、5、6、7、8 所有轴

**返回值:** 若成功，则返回 TRUE，否则返回 FALSE。

## 三、设置编码器输入信号类型

### 1. BOOL SetEncoderDivRatio (HANDLE hDevice, LONG DivRatio, LONG AxisNum)

**功能:** 设置编码器输入信号类型。

**参数:** **hDevice** 设备对象句柄，它应由 CreateDevice 创建。

**DivRatio** 脉冲的倍频数。

PCI1040\_0\_DIVRATIO 0 倍  
 PCI1040\_1\_DIVRATIO 1 倍  
 PCI1040\_2\_DIVRATIO 2 倍  
 PCI1040\_4\_DIVRATIO 4 倍

**AxisNum** 轴号。

PCI1040\_AXIS1: 1 轴  
 PCI1040\_AXIS2: 2 轴  
 PCI1040\_AXIS3: 3 轴  
 PCI1040\_AXIS4: 4 轴  
 PCI1040\_AXIS5: 5 轴

PCI1040\_AXIS6: 6 轴

PCI1040\_AXIS7: 7 轴

PCI1040\_AXIS8: 8 轴

PCI1040\_AXISALL: 1、2、3、4、5、6、7、8 所有轴

返回值: 若成功, 则返回 TRUE, 否则返回 FALSE。

#### 四、直线 S 曲线初始化、启动函数

##### 1. BOOL InitLVDVCHV (HANDLE hDevice, PPCI1040\_PARA\_DataList pDL, PPCI1040\_PARA\_LCData pLC)

功能: 初始化指定轴定长或连续驱动(带加减速)。

参数: hDevice 设备对象句柄, 它应由 CreateDevice 创建。

pDL 公共参数结构体指针, 参数包括:

LONG Multiple 倍率 (0.1~250)

LONG StartSpeed 初始速度(1~4095750)

LONG DriveSpeed 驱动速度(1~4095750)

LONG Acceleration 加速度(125~511,968,750), 说明: 加速度设置为 125 的倍数

LONG Deceleration 减速度(125~511,968,750), 说明: 减速度设置为 125 的倍数

LONG AccOrDecRate S 曲线加减速段设置 (1~8191), 应当小于 (DriveSpeed - StartSpeed)/2

pLC 直线和 S 曲线参数结构体指针, 参数包括:

LONG AxisNum 轴号

LONG LV\_DV 驱动方式 (连续 | 定长)

LONG DecMode 减速方式 (自动减速 | 偏移减速 | 手动减速)

LONG PulseMode 脉冲方式 (CW/CCW 方式 | CP/DIR 方式)

LONG PulseLogLevel 脉冲信号电平(0:正向脉冲, 1:负向脉冲)

LONG DirLogLevel 方向信号电平(0:低电平正转高电平反转, 1:高电平正转低电平反转)

LONG Line\_Curve 运动方式 (直线 | 曲线)

LONG Direction 运动方向 (正方向 | 反方向)

LONG nPulseNum 定量输出脉冲数(0~268435455)

LONG DecStartPoint 减速点数

返回值: 若成功, 则返回 TRUE, 否则返回 FALSE。

备注: 在直线加减速驱动时不需要设定加速度变化率, 在 S 曲线驱动时必须设定, 如果 S 曲线中加速和减速曲线不一样, 就必须设置减速度变化率, 并且加速度和减速度都要设置成 8000。调用例子请参看使用 PCI1040\_StartLVDV 定长、连续脉冲驱动函数启动电机。

##### 2. BOOL StartLVDVCHV (HANDLE hDevice, LONG AxisNum)

功能: 启动单轴直线 S 曲线驱动(带加减速)。

参数: hDevice 设备对象句柄, 它应由 CreateDevice 创建。

AxisNum 轴号。

PCI1040\_AXIS1: 1 轴

PCI1040\_AXIS2: 2 轴

PCI1040\_AXIS3: 3 轴

PCI1040\_AXIS4: 4 轴

PCI1040\_AXIS5: 5 轴

PCI1040\_AXIS6: 6 轴

PCI1040\_AXIS7: 7 轴

PCI1040\_AXIS8: 8 轴





PCI1040\_AXISALL: 1、2、3、4、5、6、7、8 所有轴

返回值: 若成功, 则返回 TRUE, 否则返回 FALSE。

### 3. BOOL Start8DCHV (HANDLE hDevice)

功能: 启动 8 轴同时驱动(带加减速)。

参数: hDevice 设备对象句柄, 它应由 CreateDevice 创建。

返回值: 若成功, 则返回 TRUE, 否则返回 FALSE。

### 4. BOOL InitLVDVCNST (HANDLE hDevice, LONG StartSpeed, PPCI1040\_PARA\_LCData pLC)

功能: 初始化连续, 定长脉冲驱动(常速)。

参数: hDevice 设备对象句柄, 它应由 CreateDevice 创建。

StartSpeed 速度值即起始速度。

pLC 直线和 S 曲线参数结构体指针, 参数包括:

LONG AxisNum 轴号

LONG LV\_DV 驱动方式 (连续 | 定长 )

LONG DecMode 减速方式 (自动减速 | 偏移减速 | 手动减速)

LONG PulseMode 脉冲方式 (CW/CCW 方式 | CP/DIR 方式)

LONG PulseLogLevel 脉冲信号电平(0:正向脉冲, 1:负向脉冲)

LONG DirLogLevel 方向信号电平(0:低电平正转高电平反转, 1:高电平正转低电平反转)

LONG Line\_Curve 运动方式 (直线 | 曲线)

LONG Direction 运动方向 (正方向 | 反方向)

LONG nPulseNum 定量输出脉冲数(0~268435455)

LONG DecStartPoint 减速点数

返回值: 若成功, 则返回 TRUE, 否则返回 FALSE。

### 5. BOOL StartLVDVCNST(HANDLE hDevice, LONG AxisNum)

功能: 启动连续, 定长脉冲驱动(常速)。

参数: hDevice 设备对象句柄, 它应由 CreateDevice 创建。

AxisNum 轴号。

PCI1040\_AXIS1: 1 轴

PCI1040\_AXIS2: 2 轴

PCI1040\_AXIS3: 3 轴

PCI1040\_AXIS4: 4 轴

PCI1040\_AXIS5: 5 轴

PCI1040\_AXIS6: 6 轴

PCI1040\_AXIS7: 7 轴

PCI1040\_AXIS8: 8 轴

PCI1040\_AXISALL: 1、2、3、4、5、6、7、8 所有轴

返回值: 若成功, 则返回 TRUE, 否则返回 FALSE。

### 6. BOOL Start8DCNST(HANDLE hDevice)

功能: 8 轴同时启动(常速)。

参数: hDevice 设备对象句柄, 它应由 CreateDevice 创建。

返回值: 若成功, 则返回 TRUE, 否则返回 FALSE。

## 五、任意 2 轴直线插补初始化、启动函数

### 1. BOOL InitLineInterpolation\_2D (HANDLE hDevice, PPCI1040\_PARA\_DataList pDL,



### PPCI1040\_PARA\_InterpolationAxis pIA, PCI1040\_PARA\_LineData pLD)

**功能:** 初始化任意 2 轴直线插补。

**参数:**

<b>hDevice</b>	设备对象句柄, 它应由 CreateDevice 创建。
<b>pDL</b>	公共参数结构体指针, 参数包括:
LONG Multiple	倍率 (0.1~250)
LONG StartSpeed	初始速度(1~4095750)
LONG DriveSpeed	驱动速度(1~4095750)
LONG Acceleration	加速度(125~511,968,750), 说明: 加速度设置为 125 的倍数
LONG Deceleration	减速度(125~511,968,750), 说明: 减速度设置为 125 的倍数
LONG AccOrDecRate	S 曲线加减速段设置 (1~8191), 应当小于 (DriveSpeed-StartSpeed)/2
<b>pIA</b>	插补轴选择
Axis1;	主轴
Axis2;	第二轴
Axis3;	第三轴
Axis4;	第四轴
<b>pLD</b>	直线插补参数结构体指针, 参数包括:
Line_Curve	运动方式 PCI1040_LINE:直线 PCI1040_CURVE: 曲线
ConstantSpeed	固定线速度 PCI1040_NOCONSTAND 不固定线速度 PCI1040_CONSTAND: 固定线速度
n1AxisPulseNum	主轴输出脉冲数 (0~268435455)
n2AxisPulseNum	第二轴输出脉冲数 (0~268435455)
n3AxisPulseNum	第三轴输出脉冲数 (0~268435455)
n4AxisPulseNum	第四轴输出脉冲数 (0~268435455)

**返回值:** 若成功, 则返回 TRUE, 否则返回 FALSE。

**备注:** 在直线加减速驱动时不需要设定加速度变化率, 在 S 曲线驱动时必须设定。

### 2. BOOL StartLineInterpolation\_2D ( HANDLE hDevice, PCI1040\_PARA\_LCDData pLCAxis1, PCI1040\_PARA\_LCDData pLCAxis2)

**功能:** 启动任意 2 轴直线曲线插补。

**参数:**

<b>hDevice</b>	设备对象句柄, 它应由 CreateDevice 创建。
<b>pLCAxis1</b>	直线和 S 曲线参数结构体指针, 参数包括:
LONG AxisNum	轴号
LONG LV_DV	驱动方式 (连续   定长 )
LONG DecMode	减速方式 (自动减速   偏移减速   手动减速)
LONG PulseMode	脉冲方式 (CW/CCW 方式   CP/DIR 方式)
LONG PulseLogLevel	脉冲信号电平 (0:正向脉冲, 1:负向脉冲)
LONG DirLogLevel	方向信号电平 (0:低电平正转高电平反转, 1:高电平正转低电 平反转)
LONG Line_Curve	运动方式 (直线   曲线)
LONG Direction	运动方向 (正方向   反方向)
LONG nPulseNum	定量输出脉冲数 (0~268435455)
LONG DecStartPoint	减速点数
<b>pLCAxis2</b>	直线和 S 曲线参数结构体指针, 参数包括:
LONG AxisNum	轴号
LONG LV_DV	驱动方式 (连续   定长 )
LONG DecMode	减速方式 (自动减速   偏移减速   手动减速)
LONG PulseMode	脉冲方式 (CW/CCW 方式   CP/DIR 方式)
LONG PulseLogLevel	脉冲信号电平 (0:正向脉冲, 1:负向脉冲)



LONG DirLogLevel	方向信号电平 (0:低电平正转高电平反转, 1:高电平正转低电平反转)
LONG Line_Curve	运动方式 (直线   曲线)
LONG Direction	运动方向 (正方向   反方向)
LONG nPulseNum	定量输出脉冲数 (0~268435455)
LONG DecStartPoint	减速点数

返回值: 若成功, 则返回 TRUE, 否则返回 FALSE。

## 六、任意 3 轴直线插补初始化、启动函数

### 1. BOOL InitLineInterpolation\_3D (HANDLE hDevice, PCI1040\_PARA\_DataList pDL, PCI1040\_PARA\_InterpolationAxis pIA, PCI1040\_PARA\_LineData pLD)

功能: 初始化任意 3 轴直线插补。

参数:	hDevice	设备对象句柄, 它应由 CreateDevice 创建。
	pDL	公共参数结构体指针, 参数包括:
	LONG Multiple	倍率 (0.1~250)
	LONG StartSpeed	初始速度(1~4095750)
	LONG DriveSpeed	驱动速度(1~4095750)
	LONG Acceleration	加速度(125~511,968,750), 说明: 加速度设置为 125 的倍数
	LONG Deceleration	减速度(125~511,968,750), 说明: 减速度设置为 125 的倍数
	LONG AccOrDecRate	S 曲线加减速段设置 (1~8191), 应当小于 (DriveSpeed-StartSpeed)/2
	pIA	插补轴选择
	Axis1;	主轴
	Axis2;	第二轴
	Axis3;	第三轴
	Axis4;	第四轴
	pLD	直线插补参数结构体指针, 参数包括:
	Line_Curve	运动方式 PCI1040_LINE:直线 PCI1040_CURVE: 曲线)
	ConstantSpeed	固定线速度 PCI1040_NOCONSTAND 不固定线速度 PCI1040_CONSTAND: 固定线速度
	n1AxisPulseNum	主轴输出脉冲数 (0~268435455)
	n2AxisPulseNum	第二轴输出脉冲数 (0~268435455)
	n3AxisPulseNum	第三轴输出脉冲数 (0~268435455)
	n4AxisPulseNum	第四轴输出脉冲数 (0~268435455)

返回值: 若成功, 则返回 TRUE, 否则返回 FALSE。

备注: 在直线加减速驱动时不需要设定加速度变化率, 在 S 曲线驱动时必须设定。

### 2. BOOL StartLineInterpolation\_3D ( HANDLE hDevice, PCI1040\_PARA\_LCDData pLCAxis1, PCI1040\_PARA\_LCDData pLCAxis2, PCI1040\_PARA\_LCDData pLCAxis3)

功能: 启动任意 3 轴直线曲线插补。

参数:	hDevice	设备对象句柄, 它应由 CreateDevice 创建。
	pLCAxis1	直线和 S 曲线参数结构体指针, 参数包括:
	LONG AxisNum	轴号
	LONG LV_DV	驱动方式 (连续   定长 )
	LONG DecMode	减速方式 (自动减速   偏移减速   手动减速)
	LONG PulseMode	脉冲方式 (CW/CCW 方式   CP/DIR 方式)
	LONG PulseLogLevel	脉冲信号电平 (0:正向脉冲, 1:负向脉冲)

	LONG DirLogLevel	方向信号电平 (0:低电平正转高电平反转, 1:高电平正转低电平反转)
	LONG Line_Curve	运动方式 (直线   曲线)
	LONG Direction	运动方向 (正方向   反方向)
	LONG nPulseNum	定量输出脉冲数 (0~268435455)
	LONG DecStartPoint	减速点数
pLCAxis2	直线和 S 曲线参数结构体指针, 参数包括:	
	LONG AxisNum	轴号
	LONG LV_DV	驱动方式 (连续   定长 )
	LONG DecMode	减速方式 (自动减速   偏移减速   手动减速)
	LONG PulseMode	脉冲方式 (CW/CCW 方式   CP/DIR 方式)
	LONG PulseLogLevel	脉冲信号电平 (0:正向脉冲, 1:负向脉冲)
	LONG DirLogLevel	方向信号电平 (0:低电平正转高电平反转, 1:高电平正转低电平反转)
	LONG Line_Curve	运动方式 (直线   曲线)
	LONG Direction	运动方向 (正方向   反方向)
	LONG nPulseNum	定量输出脉冲数 (0~268435455)
	LONG DecStartPoint	减速点数
pLCAxis3	直线和 S 曲线参数结构体指针, 参数包括:	
	LONG AxisNum	轴号
	LONG LV_DV	驱动方式 (连续   定长 )
	LONG DecMode	减速方式 (自动减速   偏移减速   手动减速)
	LONG PulseMode	脉冲方式 (CW/CCW 方式   CP/DIR 方式)
	LONG PulseLogLevel	脉冲信号电平 (0:正向脉冲, 1:负向脉冲)
	LONG DirLogLevel	方向信号电平 (0:低电平正转高电平反转, 1:高电平正转低电平反转)
	LONG Line_Curve	运动方式 (直线   曲线)
	LONG Direction	运动方向 (正方向   反方向)
	LONG nPulseNum	定量输出脉冲数 (0~268435455)
	LONG DecStartPoint	减速点数

返回值: 若成功, 则返回 TRUE, 否则返回 FALSE。

## 七、任意 4 轴直线插补初始化、启动函数

### 1. BOOL InitLineInterpolation\_4D (HANDLE hDevice, PPCI1040\_PARA\_DataList pDL, PPCI1040\_PARA\_InterpolationAxis pIA, PPCI1040\_PARA\_LineData pLD)

功能: 初始化任意 4 轴直线插补。

参数: hDevice 设备对象句柄, 它应由 CreateDevice 创建。

pDL 公共参数结构体指针, 参数包括:

LONG Multiple 倍率 (0.1~250)

LONG StartSpeed 初始速度(1~4095750)

LONG DriveSpeed 驱动速度(1~4095750)

LONG Acceleration 加速度(125~511,968,750), 说明: 加速度设置为 125 的倍数

LONG Deceleration 减速度(125~511,968,750), 说明: 减速度设置为 125 的倍数

LONG AccOrDecRate S 曲线加减速段设置 (1~8191), 应当小于 (DriveSpeed-StartSpeed)/2

pIA 插补轴选择

Axis1; 主轴

Axis2; 第二轴

Axis3; 第三轴



	Axis4;	第四轴
pLD	直线插补参数结构体指针，参数包括：	
	Line_Curve	运动方式 PCI1040_LINE:直线 PCI1040_CURVE: 曲线)
	ConstantSpeed	固定线速度 PCI1040_NOCONSTAND 不固定线速度 PCI1040_CONSTAND: 固定线速度
	n1AxisPulseNum	主轴输出脉冲数 (0~268435455)
	n2AxisPulseNum	第二轴输出脉冲数 (0~268435455)
	n3AxisPulseNum	第三轴输出脉冲数 (0~268435455)
	n4AxisPulseNum	第四轴输出脉冲数 (0~268435455)

返回值：若成功，则返回 TRUE，否则返回 FALSE。

备注：在直线加减速驱动时不需要设定加速度变化率，在 S 曲线驱动时必须设定。

## 2. BOOL StartLineInterpolation\_4D ( HANDLE hDevice, PCI1040\_PARA\_LCDData pLCAxis1, PCI1040\_PARA\_LCDData pLCAxis2, PCI1040\_PARA\_LCDData pLCAxis3, PCI1040\_PARA\_LCDData pLCAxis4)

功能：启动任意 4 轴直线曲线插补。

参数：	hDevice	设备对象句柄，它应由 CreateDevice 创建。
	pLCAxis1	直线和 S 曲线参数结构体指针，参数包括：
	LONG AxisNum	轴号
	LONG LV_DV	驱动方式 (连续   定长 )
	LONG DecMode	减速方式 (自动减速   偏移减速   手动减速)
	LONG PulseMode	脉冲方式 (CW/CCW 方式   CP/DIR 方式)
	LONG PulseLogLevel	脉冲信号电平 (0:正向脉冲, 1:负向脉冲)
	LONG DirLogLevel	方向信号电平 (0:低电平正转高电平反转, 1:高电平正转低电 平反转)
	LONG Line_Curve	运动方式 (直线   曲线)
	LONG Direction	运动方向 (正方向   反方向)
	LONG nPulseNum	定量输出脉冲数 (0~268435455)
	LONG DecStartPoint	减速点数
	pLCAxis2	直线和 S 曲线参数结构体指针，参数包括：
	LONG AxisNum	轴号
	LONG LV_DV	驱动方式 (连续   定长 )
	LONG DecMode	减速方式 (自动减速   偏移减速   手动减速)
	LONG PulseMode	脉冲方式 (CW/CCW 方式   CP/DIR 方式)
	LONG PulseLogLevel	脉冲信号电平 (0:正向脉冲, 1:负向脉冲)
	LONG DirLogLevel	方向信号电平 (0:低电平正转高电平反转, 1:高电平正转低电 平反转)
	LONG Line_Curve	运动方式 (直线   曲线)
	LONG Direction	运动方向 (正方向   反方向)
	LONG nPulseNum	定量输出脉冲数 (0~268435455)
	LONG DecStartPoint	减速点数
	pLCAxis3	直线和 S 曲线参数结构体指针，参数包括：
	LONG AxisNum	轴号
	LONG LV_DV	驱动方式 (连续   定长 )
	LONG DecMode	减速方式 (自动减速   偏移减速   手动减速)
	LONG PulseMode	脉冲方式 (CW/CCW 方式   CP/DIR 方式)
	LONG PulseLogLevel	脉冲信号电平 (0:正向脉冲, 1:负向脉冲)
	LONG DirLogLevel	方向信号电平 (0:低电平正转高电平反转, 1:高电平正转低电 平反转)
	LONG Line_Curve	运动方式 (直线   曲线)
	LONG Direction	运动方向 (正方向   反方向)

	LONG nPulseNum	定量输出脉冲数 (0~268435455)
	LONG DecStartPoint	减速点数
pLCAxis4	直线和 S 曲线参数结构体指针, 参数包括:	
	LONG AxisNum	轴号
	LONG LV_DV	驱动方式 (连续   定长)
	LONG DecMode	减速方式 (自动减速   偏移减速   手动减速)
	LONG PulseMode	脉冲方式 (CW/CCW 方式   CP/DIR 方式)
	LONG PulseLogLevel	脉冲信号电平 (0:正向脉冲, 1:负向脉冲)
	LONG DirLogLevel	方向信号电平 (0:低电平正转高电平反转, 1:高电平正转低电平反转)
	LONG Line_Curve	运动方式 (直线   曲线)
	LONG Direction	运动方向 (正方向   反方向)
	LONG nPulseNum	定量输出脉冲数 (0~268435455)
	LONG DecStartPoint	减速点数

返回值: 若成功, 则返回 TRUE, 否则返回 FALSE。

## 八、插补相关函数

### 1. BOOL InterpolationEnable (HANDLE hDevice, LONG IEnable, LONG AxisNum)

功能: 线性插补基设置 (最大脉冲数设置)。

参数: hDevice 设备对象句柄, 它应由 CreateDevice 创建。

IEnable 0:非插补, 1:插补。

AxisNum 轴号。

PCI1040\_AXIS1: 1 轴

PCI1040\_AXIS2: 2 轴

PCI1040\_AXIS3: 3 轴

PCI1040\_AXIS4: 4 轴

PCI1040\_AXIS5: 5 轴

PCI1040\_AXIS6: 6 轴

PCI1040\_AXIS7: 7 轴

PCI1040\_AXIS8: 8 轴

PCI1040\_AXISALL: 1、2、3、4、5、6、7、8 所有轴

返回值: 若成功, 则返回 TRUE, 否则返回 FALSE。

### 2. BOOL SetMaxPulse (HANDLE hDevice, LONG IPulse, LONG AxisNum)

功能: 线性插补基设置 (最大脉冲数设置)。将插补轴中的最大脉冲数设置到每一个轴的 R8 寄存器中。

参数: hDevice 设备对象句柄, 它应由 CreateDevice 创建。

IPulse 脉冲数。

AxisNum 轴号。

PCI1040\_AXIS1: 1 轴

PCI1040\_AXIS2: 2 轴

PCI1040\_AXIS3: 3 轴

PCI1040\_AXIS4: 4 轴

PCI1040\_AXIS5: 5 轴

PCI1040\_AXIS6: 6 轴

PCI1040\_AXIS7: 7 轴

PCI1040\_AXIS8: 8 轴



PCI1040\_AXISALL: 1、2、3、4、5、6、7、8 所有轴

**返回值:** 若成功, 则返回 TRUE, 否则返回 FALSE。

**备注:** 将插补轴中的最大脉冲数设置到每一个轴的 R8 寄存器中。

## 九、计数器相关函数

### 1. BOOL SetCounterA (HANDLE hDevice, PCI1040\_COUNTER\_MODE sMode, LONG AxisNum)

**功能:** 设置 CounterA 计数器。

**参数:** hDevice 设备对象句柄, 它应由 CreateDevice 创建。

sMode 计数器参数结构体, 参数包括:

CountMode 计数器模式设置, 0:内部逻辑计数 1:外部实位计数

Direction 计数方向, 0:正方向 1:反方向

KeepCount 是否保存原来的计数值(=0 为保持, =1 为清零)

AxisNum 轴号。

PCI1040\_AXIS1: 1 轴

PCI1040\_AXIS2: 2 轴

PCI1040\_AXIS3: 3 轴

PCI1040\_AXIS4: 4 轴

PCI1040\_AXIS5: 5 轴

PCI1040\_AXIS6: 6 轴

PCI1040\_AXIS7: 7 轴

PCI1040\_AXIS8: 8 轴

PCI1040\_AXISALL: 1、2、3、4、5、6、7、8 所有轴

**返回值:** 若成功, 则返回 TRUE, 否则返回 FALSE。

### 2. BOOL SetValCounterA(HANDLE hDevice, LONG ICounterVal, LONG AxisNum)

**功能:** 设置 CounterA 计数器值。

**参数:** hDevice 设备对象句柄, 它应由 CreateDevice 创建。

ICounterVal 设置 CounterA 值。

AxisNum 轴号。

PCI1040\_AXIS1: 1 轴

PCI1040\_AXIS2: 2 轴

PCI1040\_AXIS3: 3 轴

PCI1040\_AXIS4: 4 轴

PCI1040\_AXIS5: 5 轴

PCI1040\_AXIS6: 6 轴

PCI1040\_AXIS7: 7 轴

PCI1040\_AXIS8: 8 轴

PCI1040\_AXISALL: 1、2、3、4、5、6、7、8 所有轴

**返回值:** 若成功, 则返回 TRUE, 否则返回 FALSE。

### 3. BOOL ClearCounterA(HANDLE hDevice, LONG AxisNum)

**功能:** CounterA 计数器清零。

**参数:** hDevice 设备对象句柄, 它应由 CreateDevice 创建。

AxisNum 轴号。



PCI1040\_AXIS1: 1 轴  
 PCI1040\_AXIS2: 2 轴  
 PCI1040\_AXIS3: 3 轴  
 PCI1040\_AXIS4: 4 轴  
 PCI1040\_AXIS5: 5 轴  
 PCI1040\_AXIS6: 6 轴  
 PCI1040\_AXIS7: 7 轴  
 PCI1040\_AXIS8: 8 轴  
 PCI1040\_AXISALL: 1、2、3、4、5、6、7、8 所有轴

返回值：若成功，则返回 TRUE，否则返回 FALSE。

**4. LONG ReadCounterA(HANDLE hDevice, LONG AxisNum)**

功能：读取 CounterA 的计数值。

参数：**hDevice** 设备对象句柄，它应由 CreateDevice 创建。  
**AxisNum** 轴号。

PCI1040\_AXIS1: 1 轴  
 PCI1040\_AXIS2: 2 轴  
 PCI1040\_AXIS3: 3 轴  
 PCI1040\_AXIS4: 4 轴  
 PCI1040\_AXIS5: 5 轴  
 PCI1040\_AXIS6: 6 轴  
 PCI1040\_AXIS7: 7 轴  
 PCI1040\_AXIS8: 8 轴  
 PCI1040\_AXISALL: 1、2、3、4、5、6、7、8 所有轴

返回值：若成功，则读取 CounterA 的计数值。

**5. BOOL SetCounterB (HANDLE hDevice, PCI1040\_COUNTER\_MODE sMode, LONG AxisNum)**

功能：设置 CounterB 计数器。

参数：**hDevice** 设备对象句柄，它应由 CreateDevice 创建。  
**sMode** 计数器参数结构体，参数包括：  
**CountMode** 计数器模式设置, 0:内部逻辑计数 1:外部实位计数  
**Direction** 计数方向, 0:正方向 1:反方向  
**KeepCount** 是否保存原来的计数值(=0 为保持, =1 为清零)  
**AxisNum** 轴号。

PCI1040\_AXIS1: 1 轴  
 PCI1040\_AXIS2: 2 轴  
 PCI1040\_AXIS3: 3 轴  
 PCI1040\_AXIS4: 4 轴  
 PCI1040\_AXIS5: 5 轴  
 PCI1040\_AXIS6: 6 轴  
 PCI1040\_AXIS7: 7 轴  
 PCI1040\_AXIS8: 8 轴  
 PCI1040\_AXISALL: 1、2、3、4、5、6、7、8 所有轴

返回值：若成功，则返回 TRUE，否则返回 FALSE。



## 6. BOOL SetValCounterB (HANDLE hDevice, LONG lCounterVal, LONG AxisNum)

**功能:** 设置 CounterB 计数器值。

**参数:** hDevice 设备对象句柄, 它应由 CreateDevice 创建。

lCounterVal 设置 CounterB 值。

AxisNum 轴号。

PCI1040\_AXIS1: 1 轴

PCI1040\_AXIS2: 2 轴

PCI1040\_AXIS3: 3 轴

PCI1040\_AXIS4: 4 轴

PCI1040\_AXIS5: 5 轴

PCI1040\_AXIS6: 6 轴

PCI1040\_AXIS7: 7 轴

PCI1040\_AXIS8: 8 轴

PCI1040\_AXISALL: 1、2、3、4、5、6、7、8 所有轴

**返回值:** 若成功, 则返回 TRUE, 否则返回 FALSE。

## 7. BOOL ClearCounterB (HANDLE hDevice, LONG AxisNum)

**功能:** CounterB 计数器清零。

**参数:** hDevice 设备对象句柄, 它应由 CreateDevice 创建。

AxisNum 轴号。

PCI1040\_AXIS1: 1 轴

PCI1040\_AXIS2: 2 轴

PCI1040\_AXIS3: 3 轴

PCI1040\_AXIS4: 4 轴

PCI1040\_AXIS5: 5 轴

PCI1040\_AXIS6: 6 轴

PCI1040\_AXIS7: 7 轴

PCI1040\_AXIS8: 8 轴

PCI1040\_AXISALL: 1、2、3、4、5、6、7、8 所有轴

**返回值:** 若成功, 则返回 TRUE, 否则返回 FALSE。

## 8. LONG ReadCounterB (HANDLE hDevice, LONG AxisNum)

**功能:** 读取 CounterB 的计数值。

**参数:** hDevice 设备对象句柄, 它应由 CreateDevice 创建。

AxisNum 轴号。

PCI1040\_AXIS1: 1 轴

PCI1040\_AXIS2: 2 轴

PCI1040\_AXIS3: 3 轴

PCI1040\_AXIS4: 4 轴

PCI1040\_AXIS5: 5 轴

PCI1040\_AXIS6: 6 轴

PCI1040\_AXIS7: 7 轴

PCI1040\_AXIS8: 8 轴

PCI1040\_AXISALL: 1、2、3、4、5、6、7、8 所有轴



**返回值:** 若成功, 则读取 CounterB 的计数值。

### 9. BOOL SetInCLRAMode(HANDLE hDevice, LONG LevelOrEdge, LONG AxisNum)

**功能:** 设置外部输入信号模式, 可以外部信号对 CounterA 计数器清零。

**参数:** hDevice 设备对象句柄, 它应由 CreateDevice 创建。

LevelOrEdge 允许电平清零有效还是边沿清零有效。

AxisNum 轴号。

PCI1040\_AXIS1: 1 轴

PCI1040\_AXIS2: 2 轴

PCI1040\_AXIS3: 3 轴

PCI1040\_AXIS4: 4 轴

PCI1040\_AXIS5: 5 轴

PCI1040\_AXIS6: 6 轴

PCI1040\_AXIS7: 7 轴

PCI1040\_AXIS8: 8 轴

PCI1040\_AXISALL: 1、2、3、4、5、6、7、8 所有轴

**返回值:** 若成功, 则返回 TRUE, 否则返回 FALSE。

### 10. BOOL SetCOMPP(HANDLE hDevice, USHORT ChCounter, LONG AxisNum)

**功能:** 设置 COMP P 寄存器。

**参数:** hDevice 设备对象句柄, 它应由 CreateDevice 创建。

ChCounter 选择计数器 0:CounterA; 1:CounterB; 2:CounterC; 3:Comparator register。

PCI1040\_PQ\_COUNTERA CounterA

PCI1040\_PQ\_COUNTERB CounterB

PCI1040\_PQ\_COUNTERC CounterC

PCI1040\_PQ\_COMPARATOR Comparator register

AxisNum 轴号。

PCI1040\_AXIS1: 1 轴

PCI1040\_AXIS2: 2 轴

PCI1040\_AXIS3: 3 轴

PCI1040\_AXIS4: 4 轴

PCI1040\_AXIS5: 5 轴

PCI1040\_AXIS6: 6 轴

PCI1040\_AXIS7: 7 轴

PCI1040\_AXIS8: 8 轴

PCI1040\_AXISALL: 1、2、3、4、5、6、7、8 所有轴

**返回值:** 若成功, 则返回 TRUE, 否则返回 FALSE。

### 11. BOOL SetCOMPQ(HANDLE hDevice, USHORT ChCounter, LONG AxisNum)

**功能:** 设置 COMP Q 寄存器。

**参数:** hDevice 设备对象句柄, 它应由 CreateDevice 创建。

ChCounter 选择计数器 0:CounterA; 1:CounterB; 2:CounterC; 3:Comparator register。

PCI1040\_PQ\_COUNTERA CounterA



	PCI1040_PQ_COUNTERB	CounterB
	PCI1040_PQ_COUNTERC	CounterC
	PCI1040_PQ_COMPARATOR	Comparator register
AxisNum	轴号。	
	PCI1040_AXIS1:	1 轴
	PCI1040_AXIS2:	2 轴
	PCI1040_AXIS3:	3 轴
	PCI1040_AXIS4:	4 轴
	PCI1040_AXIS5:	5 轴
	PCI1040_AXIS6:	6 轴
	PCI1040_AXIS7:	7 轴
	PCI1040_AXIS8:	8 轴
	PCI1040_AXISALL:	1、2、3、4、5、6、7、8 所有轴

**返回值:** 若成功，则返回 TRUE，否则返回 FALSE。

## 12. LONG ReadCOMPRegVal(HANDLE hDevice, LONG AxisNum)

**功能:** 读取比较寄存器值。

**参数:** hDevice 设备对象句柄，它应由 CreateDevice 创建。

AxisNum	轴号。	
	PCI1040_AXIS1:	1 轴
	PCI1040_AXIS2:	2 轴
	PCI1040_AXIS3:	3 轴
	PCI1040_AXIS4:	4 轴
	PCI1040_AXIS5:	5 轴
	PCI1040_AXIS6:	6 轴
	PCI1040_AXIS7:	7 轴
	PCI1040_AXIS8:	8 轴
	PCI1040_AXISALL:	1、2、3、4、5、6、7、8 所有轴

**返回值:** 若成功，则读取比较寄存器值。

## 13. LONG SetCOMPRegVal(HANDLE hDevice, LONG CompVal, LONG AxisNum)

**功能:** 设置比较寄存器值。

**参数:** hDevice 设备对象句柄，它应由 CreateDevice 创建。

CompVal	比较寄存器值。	
AxisNum	轴号。	
	PCI1040_AXIS1:	1 轴
	PCI1040_AXIS2:	2 轴
	PCI1040_AXIS3:	3 轴
	PCI1040_AXIS4:	4 轴
	PCI1040_AXIS5:	5 轴
	PCI1040_AXIS6:	6 轴
	PCI1040_AXIS7:	7 轴
	PCI1040_AXIS8:	8 轴
	PCI1040_AXISALL:	1、2、3、4、5、6、7、8 所有轴

**返回值:** 若成功，则设置比较寄存器值。

#### 14. LONG SetWithoutPulse(HANDLE hDevice, LONG AxisNum)

**功能:** 设置没有脉冲输出。

**参数:** hDevice 设备对象句柄, 它应由 CreateDevice 创建。

AxisNum 轴号。

PCI1040\_AXIS1: 1 轴

PCI1040\_AXIS2: 2 轴

PCI1040\_AXIS3: 3 轴

PCI1040\_AXIS4: 4 轴

PCI1040\_AXIS5: 5 轴

PCI1040\_AXIS6: 6 轴

PCI1040\_AXIS7: 7 轴

PCI1040\_AXIS8: 8 轴

PCI1040\_AXISALL: 1、2、3、4、5、6、7、8 所有轴

**返回值:** 若成功, 则没有脉冲输出。

#### 15. LONG SetCSWithoutPulse(HANDLE hDevice, LONG AxisNum)

**功能:** 设置固定速度并且没有脉冲输出。

**参数:** hDevice 设备对象句柄, 它应由 CreateDevice 创建。

AxisNum 轴号。

PCI1040\_AXIS1: 1 轴

PCI1040\_AXIS2: 2 轴

PCI1040\_AXIS3: 3 轴

PCI1040\_AXIS4: 4 轴

PCI1040\_AXIS5: 5 轴

PCI1040\_AXIS6: 6 轴

PCI1040\_AXIS7: 7 轴

PCI1040\_AXIS8: 8 轴

PCI1040\_AXISALL: 1、2、3、4、5、6、7、8 所有轴

**返回值:** 若成功, 则设置固定速度并且没有脉冲输出。

## 十、减速函数设置

### 1. BOOL DecMode (HANDLE hDevice, LONG DecStartMode, LONG LineOrCurve, LONG AxisNum)

**功能:** 减速模式。

**参数:** hDevice 设备对象句柄, 它应由 CreateDevice 创建。

DecStartMode 减速开始点模式。

LineOrCurve =0 线性加减速模式, =1 S 曲线加减速模式。

AxisNum 轴号。

PCI1040\_AXIS1: 1 轴

PCI1040\_AXIS2: 2 轴

PCI1040\_AXIS3: 3 轴

PCI1040\_AXIS4: 4 轴

PCI1040\_AXIS5: 5 轴



PCI1040\_AXIS6: 6 轴

PCI1040\_AXIS7: 7 轴

PCI1040\_AXIS8: 8 轴

PCI1040\_AXISALL: 1、2、3、4、5、6、7、8 所有轴

返回值: 若成功, 则返回 TRUE, 否则返回 FALSE。

## 2. **BOOL SetDecStartPoint (HANDLE hDevice, LONG DecStartPoint, LONG AxisNum)**

功能: 减速点设置。

参数: hDevice 设备对象句柄, 它应由 CreateDevice 创建。

DecStartPoint 减速点数。

AxisNum 轴号。

PCI1040\_AXIS1: 1 轴

PCI1040\_AXIS2: 2 轴

PCI1040\_AXIS3: 3 轴

PCI1040\_AXIS4: 4 轴

PCI1040\_AXIS5: 5 轴

PCI1040\_AXIS6: 6 轴

PCI1040\_AXIS7: 7 轴

PCI1040\_AXIS8: 8 轴

PCI1040\_AXISALL: 1、2、3、4、5、6、7、8 所有轴

返回值: 若成功, 则返回 TRUE, 否则返回 FALSE。

## 3. **BOOL DecValid (HANDLE hDevice, LONG AxisNum)**

功能: 减速有效。

参数: hDevice 设备对象句柄, 它应由 CreateDevice 创建。

AxisNum 轴号。

PCI1040\_AXIS1: 1 轴

PCI1040\_AXIS2: 2 轴

PCI1040\_AXIS3: 3 轴

PCI1040\_AXIS4: 4 轴

PCI1040\_AXIS5: 5 轴

PCI1040\_AXIS6: 6 轴

PCI1040\_AXIS7: 7 轴

PCI1040\_AXIS8: 8 轴

PCI1040\_AXISALL: 1、2、3、4、5、6、7、8 所有轴

返回值: 若成功, 则返回 TRUE, 否则返回 FALSE。

## 4. **BOOL DecInvalid (HANDLE hDevice, LONG AxisNum)**

功能: 减速无效。

参数: hDevice 设备对象句柄, 它应由 CreateDevice 创建。

AxisNum 轴号。

PCI1040\_AXIS1: 1 轴

PCI1040\_AXIS2: 2 轴

PCI1040\_AXIS3: 3 轴

PCI1040\_AXIS4: 4 轴  
 PCI1040\_AXIS5: 5 轴  
 PCI1040\_AXIS6: 6 轴  
 PCI1040\_AXIS7: 7 轴  
 PCI1040\_AXIS8: 8 轴  
 PCI1040\_AXISALL: 1、2、3、4、5、6、7、8 所有轴

**返回值:** 若成功, 则返回 TRUE, 否则返回 FALSE。

#### 5. BOOL DecStop (HANDLE hDevice, LONG AxisNum)

**功能:** 减速停止。

**参数:** hDevice 设备对象句柄, 它应由 CreateDevice 创建。  
AxisNum 轴号。

PCI1040\_AXIS1: 1 轴  
 PCI1040\_AXIS2: 2 轴  
 PCI1040\_AXIS3: 3 轴  
 PCI1040\_AXIS4: 4 轴  
 PCI1040\_AXIS5: 5 轴  
 PCI1040\_AXIS6: 6 轴  
 PCI1040\_AXIS7: 7 轴  
 PCI1040\_AXIS8: 8 轴  
 PCI1040\_AXISALL: 1、2、3、4、5、6、7、8 所有轴

**返回值:** 若成功, 则返回 TRUE, 否则返回 FALSE。

#### 6. BOOL InstStop (HANDLE hDevice, LONG AxisNum)

**功能:** 立即停止。

**参数:** hDevice 设备对象句柄, 它应由 CreateDevice 创建。  
AxisNum 轴号。

PCI1040\_AXIS1: 1 轴  
 PCI1040\_AXIS2: 2 轴  
 PCI1040\_AXIS3: 3 轴  
 PCI1040\_AXIS4: 4 轴  
 PCI1040\_AXIS5: 5 轴  
 PCI1040\_AXIS6: 6 轴  
 PCI1040\_AXIS7: 7 轴  
 PCI1040\_AXIS8: 8 轴  
 PCI1040\_AXISALL: 1、2、3、4、5、6、7、8 所有轴

**返回值:** 若成功, 则返回 TRUE, 否则返回 FALSE。

#### 7. BOOL InstStopAll (HANDLE hDevice)

**功能:** 立即全部停止。

**参数:** hDevice 设备对象句柄, 它应由 CreateDevice 创建。

**返回值:** 若成功, 则返回 TRUE, 否则返回 FALSE。

## 十一、设置同步位

#### 1. BOOL SetSynchronAction (HANDLE hDevice, LONG AxisNum)



**功能：** 设置同步。

**参数：** **hDevice** 设备对象句柄，它应由 CreateDevice 创建。

**AxisNum** 轴号。

PCI1040\_AXIS1: 1 轴

PCI1040\_AXIS2: 2 轴

PCI1040\_AXIS3: 3 轴

PCI1040\_AXIS4: 4 轴

PCI1040\_AXIS5: 5 轴

PCI1040\_AXIS6: 6 轴

PCI1040\_AXIS7: 7 轴

PCI1040\_AXIS8: 8 轴

PCI1040\_AXISALL: 1、2、3、4、5、6、7、8 所有轴

**返回值：** 若成功，则返回 TRUE，否则返回 FALSE。

## 2. BOOL SynchronActionDisable (HANDLE hDevice, LONG AxisNum)

**功能：** 设置同步位无效。

**参数：** **hDevice** 设备对象句柄，它应由 CreateDevice 创建。

**AxisNum** 轴号。

PCI1040\_AXIS1: 1 轴

PCI1040\_AXIS2: 2 轴

PCI1040\_AXIS3: 3 轴

PCI1040\_AXIS4: 4 轴

PCI1040\_AXIS5: 5 轴

PCI1040\_AXIS6: 6 轴

PCI1040\_AXIS7: 7 轴

PCI1040\_AXIS8: 8 轴

PCI1040\_AXISALL: 1、2、3、4、5、6、7、8 所有轴

**返回值：** 若成功，则返回 TRUE，否则返回 FALSE。

## 十二、设置原点搜寻外部信号停止电机驱动的输入信号

### 1. BOOL SetOutORG (HANDLE hDevice, LONG Sensitive, LONG ORGLogicDir, LONG EZLogicDir, LONG AxisNum)

**功能：** 设置返回原点位置驱动类型的输入信号。

**参数：** **hDevice** 设备对象句柄，它应由 CreateDevice 创建。

**Sensitive** 灵敏度 (=0 低灵敏度，=1 高灵敏度)。

**ORGLogicDir** ORG 信号逻辑方向 (=0 负逻辑，=1 正逻辑)。

**EZLogicDir** EZORG 信号逻辑方向 (=0 负逻辑，=1 正逻辑) 编码器。

**AxisNum** 轴号。

PCI1040\_AXIS1: 1 轴

PCI1040\_AXIS2: 2 轴

PCI1040\_AXIS3: 3 轴

PCI1040\_AXIS4: 4 轴

PCI1040\_AXIS5: 5 轴

PCI1040\_AXIS6: 6 轴

PCI1040\_AXIS7: 7 轴

PCI1040\_AXIS8: 8 轴

PCI1040\_AXISALL: 1、2、3、4、5、6、7、8 所有轴

**返回值:** 若成功, 则返回 TRUE, 否则返回 FALSE。

### 十三、外部信号启动电机定长驱动、连续驱动

#### 1. BOOL SetOutEnableDrv (HANDLE hDevice, LONG DrvMode, LONG Direction, LONG AxisNum)

**功能:** 设置外部使能定量驱动。

**参数:** hDevice 设备对象句柄, 它应由 CreateDevice 创建。

DrvMode 外部驱动模式选择, 选项:

PCI1040\_SENSOR\_POS1 传感器位置驱动类型 I

PCI1040\_SENSOR\_POS2 传感器位置驱动类型 II

PCI1040\_SENSOR\_POS3 传感器位置驱动类型 III

PCI1040\_REORIGIN\_POS1 返回原点位置驱动类型 I

PCI1040\_REORIGIN\_POS2 返回原点位置驱动类型 II

PCI1040\_REORIGIN\_POS3 返回原点位置驱动类型 III

PCI1040\_REORIGIN\_POS4 返回原点位置驱动类型 IV

Direction 正负方向选择 (=0 正方向, =1 负方向)。

AxisNum 轴号。

PCI1040\_AXIS1: 1 轴

PCI1040\_AXIS2: 2 轴

PCI1040\_AXIS3: 3 轴

PCI1040\_AXIS4: 4 轴

PCI1040\_AXIS5: 5 轴

PCI1040\_AXIS6: 6 轴

PCI1040\_AXIS7: 7 轴

PCI1040\_AXIS8: 8 轴

PCI1040\_AXISALL: 1、2、3、4、5、6、7、8 所有轴

**返回值:** 若成功, 则返回 TRUE, 否则返回 FALSE。

### 十四、设置外部信号启动电机驱动的输入信号

#### 1. BOOL SetOutMARK(HANDLE hDevice, LONG Sensitive, LONG LogicDir, LONG AxisNum)

**功能:** 设置传感器位置驱动类型的输入信号。

**参数:** hDevice 设备对象句柄, 它应由 CreateDevice 创建。

Sensitive 灵敏度 (=0 低灵敏度, =1 高灵敏度)。

LogicDir 逻辑方向 (=0 负逻辑, =1 正逻辑)。

AxisNum 轴号。

PCI1040\_AXIS1: 1 轴

PCI1040\_AXIS2: 2 轴

PCI1040\_AXIS3: 3 轴

PCI1040\_AXIS4: 4 轴

PCI1040\_AXIS5: 5 轴



PCI1040\_AXIS6: 6 轴

PCI1040\_AXIS7: 7 轴

PCI1040\_AXIS8: 8 轴

PCI1040\_AXISALL: 1、2、3、4、5、6、7、8 所有轴

返回值: 若成功, 则返回 TRUE, 否则返回 FALSE。

## 十五、初始化外部输入信号

### 1. BOOL SetSLDPFun (HANDLE hDevice, LONG LogicMode, LONG SignalMode, LONG FunMode, LONG AxisNum)

功能: 设置外部 SLDP (Slow down) 信号的功能。

参数: hDevice 设备对象句柄, 它应由 CreateDevice 创建。

LogicMode 逻辑方式 (=0 负逻辑, =1 正逻辑)。

SignalMode 信号模式(=0 电平信号, =1 边沿信号)。

FunMode =0: 减速, =1: 减速停止。

AxisNum 轴号。

PCI1040\_AXIS1: 1 轴

PCI1040\_AXIS2: 2 轴

PCI1040\_AXIS3: 3 轴

PCI1040\_AXIS4: 4 轴

PCI1040\_AXIS5: 5 轴

PCI1040\_AXIS6: 6 轴

PCI1040\_AXIS7: 7 轴

PCI1040\_AXIS8: 8 轴

PCI1040\_AXISALL: 1、2、3、4、5、6、7、8 所有轴

返回值: 若成功, 则返回 TRUE, 否则返回 FALSE。

### 2. BOOL SetSLDMFun (HANDLE hDevice, LONG LogicMode, LONG SignalMode, LONG FunMode, LONG AxisNum)

功能: 设置外部 SLDM (Slow down) 信号功能。

参数: hDevice 设备对象句柄, 它应由 CreateDevice 创建。

LogicMode 逻辑方式 (=0 负逻辑, =1 正逻辑)。

SignalMode 信号模式(=0 电平信号, =1 边沿信号)。

FunMode =0: 减速, =1: 减速停止。

AxisNum 轴号。

PCI1040\_AXIS1: 1 轴

PCI1040\_AXIS2: 2 轴

PCI1040\_AXIS3: 3 轴

PCI1040\_AXIS4: 4 轴

PCI1040\_AXIS5: 5 轴

PCI1040\_AXIS6: 6 轴

PCI1040\_AXIS7: 7 轴

PCI1040\_AXIS8: 8 轴

PCI1040\_AXISALL: 1、2、3、4、5、6、7、8 所有轴



**返回值：**若成功，则返回 TRUE，否则返回 FALSE。

**备注：**当软件限位有效时，就减速停止。

### 3. BOOL SetELPFun (HANDLE hDevice, LONG LogicMode, LONG AxisNum)

**功能：**设置外部 ELP（硬件限位）信号的功能。

**参数：**hDevice 设备对象句柄，它应由 CreateDevice 创建。

LogicMode 逻辑方式（=0 负逻辑，=1 正逻辑）。

AxisNum 轴号。

PCI1040\_AXIS1: 1 轴

PCI1040\_AXIS2: 2 轴

PCI1040\_AXIS3: 3 轴

PCI1040\_AXIS4: 4 轴

PCI1040\_AXIS5: 5 轴

PCI1040\_AXIS6: 6 轴

PCI1040\_AXIS7: 7 轴

PCI1040\_AXIS8: 8 轴

PCI1040\_AXISALL: 1、2、3、4、5、6、7、8 所有轴

**返回值：**若成功，则返回 TRUE，否则返回 FALSE。

### 4. BOOL SetELMFun (HANDLE hDevice, LONG LogicMode, LONG AxisNum)

**功能：**设置外部 ELM（硬件限位）信号的功能。

**参数：**hDevice 设备对象句柄，它应由 CreateDevice 创建。

LogicMode 逻辑方式（=0 负逻辑，=1 正逻辑）。

AxisNum 轴号。

PCI1040\_AXIS1: 1 轴

PCI1040\_AXIS2: 2 轴

PCI1040\_AXIS3: 3 轴

PCI1040\_AXIS4: 4 轴

PCI1040\_AXIS5: 5 轴

PCI1040\_AXIS6: 6 轴

PCI1040\_AXIS7: 7 轴

PCI1040\_AXIS8: 8 轴

PCI1040\_AXISALL: 1、2、3、4、5、6、7、8 所有轴

**返回值：**若成功，则返回 TRUE，否则返回 FALSE。

### 5. BOOL SetINPFun (HANDLE hDevice, LONG LogicMode, LONG AxisNum)

**功能：**设置外部 INP（外部到位）信号的功能。

**参数：**hDevice 设备对象句柄，它应由 CreateDevice 创建。

LogicMode 逻辑方式（=0 负逻辑，=1 正逻辑）。

AxisNum 轴号。

PCI1040\_AXIS1: 1 轴

PCI1040\_AXIS2: 2 轴

PCI1040\_AXIS3: 3 轴

PCI1040\_AXIS4: 4 轴



PCI1040\_AXIS5: 5 轴  
 PCI1040\_AXIS6: 6 轴  
 PCI1040\_AXIS7: 7 轴  
 PCI1040\_AXIS8: 8 轴  
 PCI1040\_AXISALL: 1、2、3、4、5、6、7、8 所有轴

返回值: 若成功, 则返回 TRUE, 否则返回 FALSE。

**6. BOOL SetALMFun (HANDLE hDevice,  
 LONG LogicMode,  
 LONG AxisNum)**

功能: 设置外部 ALM (报警) 信号的功能。

参数: hDevice 设备对象句柄, 它应由 CreateDevice 创建。  
 LogicMode 逻辑方式 (=0 负逻辑, =1 正逻辑)。  
 AxisNum 轴号。

PCI1040\_AXIS1: 1 轴  
 PCI1040\_AXIS2: 2 轴  
 PCI1040\_AXIS3: 3 轴  
 PCI1040\_AXIS4: 4 轴  
 PCI1040\_AXIS5: 5 轴  
 PCI1040\_AXIS6: 6 轴  
 PCI1040\_AXIS7: 7 轴  
 PCI1040\_AXIS8: 8 轴  
 PCI1040\_AXISALL: 1、2、3、4、5、6、7、8 所有轴

返回值: 若成功, 则返回 TRUE, 否则返回 FALSE。

**7. BOOL SetSensitive (HANDLE hDevice,  
 LONG ISensitive,  
 LONG AxisNum)**

功能: 设置输入滤波初始化 (灵敏度) 同时对 EL,ALM,SLD 有效。

参数: hDevice 设备对象句柄, 它应由 CreateDevice 创建。  
 ISensitive 灵敏度 (范围: 0—255, 0 代表 256)。  
 AxisNum 轴号。

PCI1040\_AXIS1: 1 轴  
 PCI1040\_AXIS2: 2 轴  
 PCI1040\_AXIS3: 3 轴  
 PCI1040\_AXIS4: 4 轴  
 PCI1040\_AXIS5: 5 轴  
 PCI1040\_AXIS6: 6 轴  
 PCI1040\_AXIS7: 7 轴  
 PCI1040\_AXIS8: 8 轴  
 PCI1040\_AXISALL: 1、2、3、4、5、6、7、8 所有轴

返回值: 若成功, 则返回 TRUE, 否则返回 FALSE。

**十六、设置常速有效和无效**

**1. BOOL CNSTValid (HANDLE hDevice,  
 LONG AxisNum)**

功能: 设置常速有效。

参数: hDevice 设备对象句柄, 它应由 CreateDevice 创建。  
 AxisNum 轴号。

PCI1040\_AXIS1: 1 轴  
 PCI1040\_AXIS2: 2 轴  
 PCI1040\_AXIS3: 3 轴  
 PCI1040\_AXIS4: 4 轴  
 PCI1040\_AXIS5: 5 轴  
 PCI1040\_AXIS6: 6 轴  
 PCI1040\_AXIS7: 7 轴  
 PCI1040\_AXIS8: 8 轴  
 PCI1040\_AXISALL: 1、2、3、4、5、6、7、8 所有轴

**返回值:** 若成功, 则返回 TRUE, 否则返回 FALSE。

## 2. BOOL CNSTInvalid (HANDLE hDevice, LONG AxisNum)

**功能:** 设置常速无效。

**参数:** hDevice 设备对象句柄, 它应由 CreateDevice 创建。  
AxisNum 轴号。

PCI1040\_AXIS1: 1 轴  
 PCI1040\_AXIS2: 2 轴  
 PCI1040\_AXIS3: 3 轴  
 PCI1040\_AXIS4: 4 轴  
 PCI1040\_AXIS5: 5 轴  
 PCI1040\_AXIS6: 6 轴  
 PCI1040\_AXIS7: 7 轴  
 PCI1040\_AXIS8: 8 轴  
 PCI1040\_AXISALL: 1、2、3、4、5、6、7、8 所有轴

**返回值:** 若成功, 则返回 TRUE, 否则返回 FALSE。

## 十七、设置通用输入和通用输出

### 1. BOOL SetCLROutput (HANDLE hDevice, LONG CLRMode, LONG CLRVal, LONG AxisNum)

**功能:** 设置 CLR 通用输出。

**参数:** hDevice 设备对象句柄, 它应由 CreateDevice 创建。  
CLRMode 输出模式 (0: 1 个脉冲输出, 宽度是时钟周期的 32 倍。1: 电平输出)。  
CLRVal 当 CLRMode = 1 时有效, 0: 低电平 1:高电平。  
AxisNum 轴号。

PCI1040\_AXIS1: 1 轴  
 PCI1040\_AXIS2: 2 轴  
 PCI1040\_AXIS3: 3 轴  
 PCI1040\_AXIS4: 4 轴  
 PCI1040\_AXIS5: 5 轴  
 PCI1040\_AXIS6: 6 轴  
 PCI1040\_AXIS7: 7 轴  
 PCI1040\_AXIS8: 8 轴  
 PCI1040\_AXISALL: 1、2、3、4、5、6、7、8 所有轴

**返回值:** 若成功, 则返回 TRUE, 否则返回 FALSE。



## 2. BOOL SetSONOutput (HANDLE hDevice, LONG SONVal, LONG AxisNum)

**功能:** 设置 SON 传感器通用输出。

**参数:** hDevice 设备对象句柄, 它应由 CreateDevice 创建。

SONVal =0: 低电平, =1:高电平。

AxisNum 轴号。

PCI1040\_AXIS1: 1 轴

PCI1040\_AXIS2: 2 轴

PCI1040\_AXIS3: 3 轴

PCI1040\_AXIS4: 4 轴

PCI1040\_AXIS5: 5 轴

PCI1040\_AXIS6: 6 轴

PCI1040\_AXIS7: 7 轴

PCI1040\_AXIS8: 8 轴

PCI1040\_AXISALL: 1、2、3、4、5、6、7、8 所有轴

**返回值:** 若成功, 则返回 TRUE, 否则返回 FALSE。

## 3. BOOL SetDeviceDO (HANDLE hDevice, PPCI1040\_PARA\_DO pPara)

**功能:** 设置通用输出。

**参数:** hDevice 设备对象句柄, 它应由 CreateDevice 创建。

pPara 通用输出结构体指针。

**返回值:** 若成功, 则返回 TRUE, 否则返回 FALSE。

## 4. BOOL GetDeviceDI (HANDLE hDevice, PPCI1040\_PARA\_DI pPara)

**功能:** 获取通用输入状态。

**参数:** hDevice 设备对象句柄, 它应由 CreateDevice 创建。

pPara 通用输入结构体指针。

**返回值:** 若成功, 则返回 TRUE, 否则返回 FALSE。

# 十八、读电机速度

## 1. LONG ReadCV(HANDLE hDevice, LONG AxisNum)

**功能:** 读当前速度。

**参数:** hDevice 设备对象句柄, 它应由 CreateDevice 创建。

AxisNum 轴号。

PCI1040\_AXIS1: 1 轴

PCI1040\_AXIS2: 2 轴

PCI1040\_AXIS3: 3 轴

PCI1040\_AXIS4: 4 轴

PCI1040\_AXIS5: 5 轴

PCI1040\_AXIS6: 6 轴

PCI1040\_AXIS7: 7 轴

PCI1040\_AXIS8: 8 轴

PCI1040\_AXISALL: 1、2、3、4、5、6、7、8 所有轴

**返回值:** 若成功, 则返回当前速度的值, 轴号错误返回 FALSE。

**备注:** 当前读出的速度, 范围在 (1~4095750), 实际速度 = 读出的速度值×倍率

## 十九、读状态寄存器的位状态

### 1. BOOL GetRunStatus (HANDLE hDevice, PPCI1040\_PARA\_RUN pPara, LONG AxisNum)

**功能:** 获得运行状态寄存器的位状态。

**参数:** hDevice 设备对象句柄, 它应由 CreateDevice 创建。

pPara RR0 寄存器状态。

AxisNum 轴号。

PCI1040\_AXIS1: 1 轴

PCI1040\_AXIS2: 2 轴

PCI1040\_AXIS3: 3 轴

PCI1040\_AXIS4: 4 轴

PCI1040\_AXIS5: 5 轴

PCI1040\_AXIS6: 6 轴

PCI1040\_AXIS7: 7 轴

PCI1040\_AXIS8: 8 轴

PCI1040\_AXISALL: 1、2、3、4、5、6、7、8 所有轴

**返回值:** 若成功, 则返回 TRUE, 否则返回 FALSE。

### 2. BOOL GetSensorStatus (HANDLE hDevice, PPCI1040\_PARA\_RUN pPara, LONG AxisNum)

**功能:** 获得传感器状态寄存器的位状态。

**参数:** hDevice 设备对象句柄, 它应由 CreateDevice 创建。

pPara 传感器结构体。

AxisNum 轴号。

PCI1040\_AXIS1: 1 轴

PCI1040\_AXIS2: 2 轴

PCI1040\_AXIS3: 3 轴

PCI1040\_AXIS4: 4 轴

PCI1040\_AXIS5: 5 轴

PCI1040\_AXIS6: 6 轴

PCI1040\_AXIS7: 7 轴

PCI1040\_AXIS8: 8 轴

PCI1040\_AXISALL: 1、2、3、4、5、6、7、8 所有轴

**返回值:** 若成功, 则返回 TRUE, 否则返回 FALSE。

### 3. BOOL GetComparatorStatus (HANDLE hDevice, PLONG PEqualQ, PLONG PLargerQ, LONG AxisNum)

**功能:** 获得比较器状态寄存器的位状态。

**参数:** hDevice 设备对象句柄, 它应由 CreateDevice 创建。

PEqualQ P 和 Q 是否相等(P is(not) equal to Q)。

PLargerQ P 是否大于 Q(P is(not) larger to Q)。

AxisNum 轴号。

PCI1040\_AXIS1: 1 轴

PCI1040\_AXIS2: 2 轴

PCI1040\_AXIS3: 3 轴



PCI1040\_AXIS4: 4 轴  
 PCI1040\_AXIS5: 5 轴  
 PCI1040\_AXIS6: 6 轴  
 PCI1040\_AXIS7: 7 轴  
 PCI1040\_AXIS8: 8 轴  
 PCI1040\_AXISALL: 1、2、3、4、5、6、7、8 所有轴

返回值: 若成功, 则返回 TRUE, 否则返回 FALSE。

**4. BOOL GetINTFlag(HANDLE hDevice,  
 PPCI1040\_FLAG\_INT pPara,  
 LONG AxisNum)**

功能: 获得中断标志寄存器的位状态。

参数: hDevice 设备对象句柄, 它应由 CreateDevice 创建。  
 pPara 中断状态结构体。  
 AxisNum 轴号。

PCI1040\_AXIS1: 1 轴  
 PCI1040\_AXIS2: 2 轴  
 PCI1040\_AXIS3: 3 轴  
 PCI1040\_AXIS4: 4 轴  
 PCI1040\_AXIS5: 5 轴  
 PCI1040\_AXIS6: 6 轴  
 PCI1040\_AXIS7: 7 轴  
 PCI1040\_AXIS8: 8 轴  
 PCI1040\_AXISALL: 1、2、3、4、5、6、7、8 所有轴

返回值: 若成功, 则返回 TRUE, 否则返回 FALSE。

**5. BOOL GetMainStatus (HANDLE hDevice,  
 PPCI1040\_MAIN\_STATUS pMain,  
 LONG AxisNum)**

功能: 获得主要状态寄存器的状态。

参数: hDevice 设备对象句柄, 它应由 CreateDevice 创建。  
 pMain 结构体。  
 AxisNum 轴号。

PCI1040\_AXIS1: 1 轴  
 PCI1040\_AXIS2: 2 轴  
 PCI1040\_AXIS3: 3 轴  
 PCI1040\_AXIS4: 4 轴  
 PCI1040\_AXIS5: 5 轴  
 PCI1040\_AXIS6: 6 轴  
 PCI1040\_AXIS7: 7 轴  
 PCI1040\_AXIS8: 8 轴  
 PCI1040\_AXISALL: 1、2、3、4、5、6、7、8 所有轴

返回值: 若成功, 则返回 TRUE, 否则返回 FALSE。

**6. BOOL GetNSFactorStatus (HANDLE hDevice,  
 PPCI1040\_NORMAL\_STOP\_FACTOR pNSFact,  
 LONG AxisNum)**

功能: 获得正常停止原因寄存器状态。

参数: hDevice 设备对象句柄, 它应由 CreateDevice 创建。

**pNSFact** 结构体。  
**AxisNum** 轴号。  
 PCI1040\_AXIS1: 1 轴  
 PCI1040\_AXIS2: 2 轴  
 PCI1040\_AXIS3: 3 轴  
 PCI1040\_AXIS4: 4 轴  
 PCI1040\_AXIS5: 5 轴  
 PCI1040\_AXIS6: 6 轴  
 PCI1040\_AXIS7: 7 轴  
 PCI1040\_AXIS8: 8 轴  
 PCI1040\_AXISALL: 1、2、3、4、5、6、7、8 所有轴

**返回值：**若成功，则返回 TRUE，否则返回 FALSE。

### 7. **BOOL GetESFactorStatus (HANDLE hDevice, PPCI1040\_ERROR\_STOP\_FACTOR pESFact, LONG AxisNum)**

**功能：**获得错误停止原因寄存器状态。

**参数：**  
**hDevice** 设备对象句柄，它应由 CreateDevice 创建。  
**pESFact** 结构体。  
**AxisNum** 轴号。

PCI1040\_AXIS1: 1 轴  
 PCI1040\_AXIS2: 2 轴  
 PCI1040\_AXIS3: 3 轴  
 PCI1040\_AXIS4: 4 轴  
 PCI1040\_AXIS5: 5 轴  
 PCI1040\_AXIS6: 6 轴  
 PCI1040\_AXIS7: 7 轴  
 PCI1040\_AXIS8: 8 轴  
 PCI1040\_AXISALL: 1、2、3、4、5、6、7、8 所有轴

**返回值：**若成功，则返回 TRUE，否则返回 FALSE。

## 二十、中断位设置、插补中断状态清除

### 1. **BOOL SetInterruptBit (HANDLE hDevice, PPCI1040\_PARA\_Interrupt pPara, LONG AxisNum)**

**功能：**设置中断位。

**参数：**  
**hDevice** 设备对象句柄，它应由 CreateDevice 创建。  
**pPara** 中断位结构体指针。  
**AxisNum** 轴号。

PCI1040\_AXIS1: 1 轴  
 PCI1040\_AXIS2: 2 轴  
 PCI1040\_AXIS3: 3 轴  
 PCI1040\_AXIS4: 4 轴  
 PCI1040\_AXIS5: 5 轴  
 PCI1040\_AXIS6: 6 轴  
 PCI1040\_AXIS7: 7 轴  
 PCI1040\_AXIS8: 8 轴  
 PCI1040\_AXISALL: 1、2、3、4、5、6、7、8 所有轴



**返回值:** 若成功, 则返回 TRUE, 否则返回 FALSE。

**2. BOOL GetInterruptAxis (HANDLE hDevice,  
                                  BOOL bAxisSts[8])**

**功能:** 取得 8 个轴中断状态。

**参数:** hDevice 设备对象句柄, 它应由 CreateDevice 创建。  
bAxisSts 数组至少 8 个, 存放每个轴的中断状态。

**返回值:** 若成功, 则返回 TRUE, 否则返回 FALSE。

**3. BOOL INTFlagClear(HANDLE hDevice,  
                          LONG INTFlagType,  
                          LONG AxisNum)**

**功能:** 中断标志寄存器清零。

**参数:** hDevice 设备对象句柄, 它应由 CreateDevice 创建。  
INTFlagType 中断标志类型。

PCI1040_INTFLAG_DEC	减速开始中断标志
PCI1040_INTFLAG_MAXA	最大加速度中断标志
PCI1040_INTFLAG_CNTAE	Counter A 进位中断标志
PCI1040_INTFLAG_CNTAB	Counter A 借位中断标志
PCI1040_INTFLAG_CNTBE	Counter B 进位中断标志
PCI1040_INTFLAG_CNTBB	Counter B 借位中断标志
PCI1040_INTFLAG_CNTCB	Counter C 借位中断标志
PCI1040_INTFLAG_ORG	ORG 传感器中断标志
PCI1040_INTFLAG_EZ	EZ 传感器中断标志
PCI1040_INTFLAG_IN0	IN0 传感器中断标志
PCI1040_INTFLAG_MARK	MARK 传感器中断标志
PCI1040_INTFLAG_PEQUQ	比较 P=Q 中断标志
PCI1040_INTFLAG_PBIGQ	比较 P>Q 中断标志
PCI1040_INTFLAG_ALL	清除所有中断标志

AxisNum

PCI1040_AXIS1:	1 轴
PCI1040_AXIS2:	2 轴
PCI1040_AXIS3:	3 轴
PCI1040_AXIS4:	4 轴
PCI1040_AXIS5:	5 轴
PCI1040_AXIS6:	6 轴
PCI1040_AXIS7:	7 轴
PCI1040_AXIS8:	8 轴
PCI1040_AXISALL:	1、2、3、4、5、6、7、8 所有轴

**返回值:** 若成功, 则返回 TRUE, 否则返回 FALSE。



## 第四章 硬件参数结构

### 第一节、公用参数介绍 (PCI1040\_PARA\_DataList)

```
typedef struct _PCI1040_PAPA_DataList
{
    LONG Multiple;           // 倍率 (0.1~250)
    LONG StartSpeed;        // 初始速度(1~4095750)
    LONG DriveSpeed;        // 驱动速度(1~4095750)
    LONG Acceleration;      // 加速度(125~511,968,750), 说明: 加速度设置为 125 的倍数
    LONG Deceleration;      // 减速度(125~511,968,750), 说明: 减速度设置为 125 的倍数
    LONG AccOrDecRate;      // S 曲线加减速段设置(1~8191), 应当小于(DriveSpeed - StartSpeed)/2
} PCI1040_PARA_DataList, *PPCI1040_PARA_DataList;
```

**Multiple** 倍率 (0.1~250)。

**StartSpeed** 初始速度(1~4095750)。

**DriveSpeed** 驱动速度(1~4095750)。

**Acceleration** 加速度(125~511,968,750), 说明: 加速度设置为 125 的倍数。

**Deceleration** 减速度(125~511,968,750), 说明: 减速度设置为 125 的倍数。

**AccOrDecRate** S 曲线加减速段设置(1~8191), 应当小于(DriveSpeed - StartSpeed)/2。

### 第二节、直线和 S 曲线参数介绍 (PCI1040\_PAPA\_LCData)

```
typedef struct _PCI1040_PAPA_LCData
{
    LONG AxisNum;           // 轴号
    LONG LV_DV;            // 驱动方式 (连续 | 定长 )
    LONG DecMode;          // 减速方式 (自动减速 | 偏移减速 | 手动减速)
    LONG PulseMode;        // 脉冲方式 (CW/CCW 方式 | CP/DIR 方式)
    LONG PulseLogLevel;    // 脉冲信号电平(0:正向脉冲, 1:负向脉冲)
    LONG DirLogLevel;      // 方向信号电平(0:低电平正转高电平反转, 1:高电平正转低电平反转)
    LONG Line_Curve;       // 运动方式 (直线 | 曲线)
    LONG Direction;        // 运动方向 (正方向 | 反方向)
    LONG nPulseNum;        // 定量输出脉冲数(0~268435455)
    LONG DecStartPoint;    // 减速点数
} PCI1040_PARA_LCData, *PPCI1040_PARA_LCData;
```

**AxisNum** 轴号选择, 取值如下表:

常量名	常量值	功能定义
PCI1040_AXIS1	0X00	1 轴
PCI1040_AXIS2	0x01	2 轴
PCI1040_AXIS3	0x02	3 轴
PCI1040_AXIS4	0x03	4 轴
PCI1040_AXIS5	0x04	5 轴
PCI1040_AXIS6	0x05	6 轴
PCI1040_AXIS7	0x06	7 轴
PCI1040_AXIS8	0x07	8 轴
PCI1040_AXISALL	0xFF	1、2、3、4、5、6、7、8 所有轴



LV\_DV 驱动方式选择，取值如下表：

常量名	常量值	功能定义
PCI1040_DV	0x0	定长驱动
PCI1040_LV	0x1	连续驱动

DecMode 减速方式，取值如下表：

常量名	常量值	功能定义
PCI1040_AUTO	0x0	自动减速
PCI1040_OFFSET	0x1	偏置设置减速
PCI1040_HAND	0x2	手动减速

PulseMode 脉冲输出方式选择，取值如下表：

常量名	常量值	功能定义
PCI1040_CWCCW	0x0	CW/CCW 方式
PCI1040_CPDIR	0x1	CP/DIR 方式

PulseLogLevel 脉冲信号电平(0:正向脉冲, 1:负向脉冲)。

DirLogLevel 方向信号电平(0:低电平正转高电平反转, 1:高电平正转低电平反转)。

Line\_Curve 运动方式选择，取值如下表：

常量名	常量值	功能定义
PCI1040_LINE	0x0	直线运动
PCI1040_CURVE	0x1	S 曲线运动

Direction 运动方向选择，取值如下表：

常量名	常量值	功能定义
PCI1040_PDIRECTION	0x0	正方向
PCI1040_MDIRECTION	0x1	反方向

nPulseNum 定量输出脉冲数(0~268435455)。

DecStartPoint 减速点数。

### 第三节、插补轴参数介绍 (PCI1040\_PAPA\_InterpolationAxis)

*Visual C++ & C++Builder:*

```
typedef struct _PCI1040_PAPA_InterpolationAxis
{
    LONG Axis1;           // 主轴
    LONG Axis2;           // 第二轴
    LONG Axis3;           // 第三轴
    LONG Axis4;           // 第四轴
} PCI1040_PARA_InterpolationAxis, *PPCI1040_PARA_InterpolationAxis;
```

### 第四节、直线插补和固定线速度直线插补参数介绍 (PCI1040\_PAPA\_LineData)

```
typedef struct _PCI1040_PAPA_LineData
{
```

```

    ULONG Line_Curve;           // 运动方式 (直线 | 曲线)
    ULONG ConstantSpeed;       // 固定速度 (不固定速度(带加减速) | 固定速度)
    ULONG n1AxisPulseNum;     // 主轴输出脉冲数 (0~268435455)
    ULONG n2AxisPulseNum;     // 第二轴输出脉冲数 (0~268435455)
    ULONG n3AxisPulseNum;     // 第三轴输出脉冲数 (0~268435455)
    ULONG n4AxisPulseNum;     // 第四轴输出脉冲数 (0~268435455)
} PCI1040_PARA_LineData, *PPCI1040_PARA_LineData;

```

Line\_Curve 运动方式选择, 取值如下表:

常量名	常量值	功能定义
PCI1040_LINE	0x0	直线运动
PCI1040_CURVE	0x1	S 曲线运动

ConstantSpeed 固定速度选择, 取值如下表:

常量名	常量值	功能定义
PCI1040_NOCONSTAND	0x0	不固定速度
PCI1040_CONSTAND	0x1	固定速度

- n1AxisPulseNum 主轴输出脉冲数 (0~268435455)。
- n2AxisPulseNum 第二轴输出脉冲数 (0~268435455)。
- n3AxisPulseNum 第三轴输出脉冲数 (0~268435455)。
- n4AxisPulseNum 第四轴输出脉冲数 (0~268435455)。

## 第五节、设置中断位使能参数介绍 (PCI1040\_PARA\_Interrupt)

```

typedef struct _PCI1040_PARA_Interrupt
{
    UINT PULSEINT;           // 1: 中断使能, 脉冲输出正常完成中断 0: 禁止中断
    UINT ERRORINT;          // 1: 中断使能, 错误停止时中断 0: 禁止中断
    UINT DECINT;             // 1: 中断使能, 减速开始点中断 0: 禁止中断
    UINT MAXAINT;           // 1: 中断使能, 最大加速度中断 0: 禁止中断
    UINT CNTAEINT;          // 1: 中断使能, Counter A 进位中断 0: 禁止中断
    UINT CNTABINT;          // 1: 中断使能, Counter A 借位 中断 0: 禁止中断
    UINT CNTBEINT;          // 1: 中断使能, Counter B 进位中断 0: 禁止中断
    UINT CNTBBINT;          // 1: 中断使能, Counter B 借位 中断 0: 禁止中断
    UINT CNTCBINT;          // 1: 中断使能, Counter C 借位 中断 0: 禁止中断
    UINT ORGINT;            // 1: 中断使能, ORG 中断 0: 禁止中断
    UINT EZINT;             // 1: 中断使能, EZ 中断 0: 禁止中断
    UINT IN0INT;            // 1: 中断使能, IN0 中断 0: 禁止中断
    UINT MARKINT;           // 1: 中断使能, MARK 中断 0: 禁止中断
    UINT PEQUQ;             // 1: 中断使能, P=Q 中断 0: 禁止中断
    UINT PBIGQ;             // 1: 中断使能, P>Q 中断 0: 禁止中断
} PCI1040_PARA_Interrupt, *PPCI1040_PARA_Interrupt;

```

## 第六节、IO 输出参数介绍 (PCI1040\_PARA\_DO)

```

typedef struct _PCI1040_PARA_DO
{
    UINT OUT0;              // 输出 0

```



```

    UINT OUT1;        // 输出 1
    UINT OUT2;        // 输出 2
    UINT OUT3;        // 输出 3
    UINT OUT4;        // 输出 4
    UINT OUT5;        // 输出 5
    UINT OUT6;        // 输出 6
    UINT OUT7;        // 输出 7
} PCI1040_PARA_DO, *PPCI1040_PARA_DO;

```

## 第七节、IO 输入参数介绍 (PCI1040\_PARA\_DI)

```

typedef struct _PCI1040_PARA_DI
{
    UINT IN0;         // 输入 0
    UINT IN1;         // 输入 1
    UINT IN2;         // 输入 2
    UINT IN3;         // 输入 3
    UINT IN4;         // 输入 4
    UINT IN5;         // 输入 5
    UINT IN6;         // 输入 6
    UINT IN7;         // 输入 7
} PCI1040_PARA_DI, *PPCI1040_PARA_DI;

```

## 第八节、运行状态寄存器参数介绍 (PCI1040\_PARA\_RUN)

```

typedef struct _PCI1040_PARA_RUN
{
    UINT RUN;         // 运行状态 1: 运行, 0: 停止
    UINT ASND;        // 在加/减速驱动中加速时, 为 1
    UINT DSND;        // 在加/减速驱动中减速时, 为 1
    UINT ERRFLAG;     // 错误标志为 ON 时, 为 1
    UINT STOP;        // 停止标志为 ON 时, 为 1
    UINT INT;         // 中断标志为 ON 时, 为 1
    UINT CLR;         // CLR 标志为 ON 时, 为 1
} PCI1040_PARA_RUN, *PPCI1040_PARA_RUN;

```

## 第九节、传感器状态寄存器参数介绍 (PCI1040\_PARA\_SENSOR)

```

typedef struct _PCI1040_PARA_SENSOR
{
    UINT PEL;         // 正向限位 P End Limit
    UINT MEL;         // 负向限位
    UINT ALM;         // 在加/减速驱动中减速时, 为 1
    UINT ORG;         // 原点标志为 ON 时, 为 1
    UINT EZ;          // 编码器 Z 向标志为 ON 时, 为 1
    UINT PSLD;        // 正向减速位置(Slow Down) 标志为 ON 时, 为 1
    UINT MSLD;        // 负向减速位置(Slow Down) 标志为 ON 时, 为 1
    UINT INP;         // 外部到位标志为 ON 时, 为 1
    UINT MARK;        // 外部启动标志为 ON 时, 为 1
} PCI1040_PARA_SENSOR, *PPCI1040_PARA_SENSOR;

```

## 第十节、中断状态寄存器参数介绍 (PCI1040\_FLAG\_INT)

```

typedef struct _PCI1040_FLAG_INT

```

```

{
  UINT PulseOC;           // 0: OFF, 1: ON   脉冲输出完成中断标志
  UINT ErrStop;          // 0: OFF, 1: ON   错误停止中断标志
  UINT DecStartP;        // 0: OFF, 1: ON   减速开始点中断标志
  UINT MaxAccRate;       // 0: OFF, 1: ON   最大加速度倍率中断标志
  UINT CNTAcarry;        // 0: OFF, 1: ON   CounterA carry 中断标志
  UINT CNTAborrow;       // 0: OFF, 1: ON   CounterA borrow 中断标志
  UINT CNTBcarry;        // 0: OFF, 1: ON   CounterB carry 中断标志
  UINT CNTBborrow;       // 0: OFF, 1: ON   CounterB borrow 中断标志
  UINT CNTCborrow;       // 0: OFF, 1: ON   CounterC borrow 中断标志
  UINT ORG;              // 0: OFF, 1: ON   原点搜寻中断标志
  UINT EZ;               // 0: OFF, 1: ON   编码器中断标志
  UINT IN0;              // 0: OFF, 1: ON   开关量输入 DI0 中断标志
  UINT MARK;            // 0: OFF, 1: ON   外部点动中断标志
  UINT PQequal;          // 0: OFF, 1: ON   P = Q 中断标志
  UINT PMoreQ;           // 0: OFF, 1: ON   P > Q 中断标志
} PCI1040_FLAG_INT, *PPCI1040_FLAG_INT;

```

## 第十一节、主要状态寄存器参数介绍 (PCI1040\_MAIN\_STATUS)

```

typedef struct _PCI1040_MAIN_STATUS
{
  PCI1040_PARA_RUN ParaRun; // 运行状态
  UINT INTPulseOC;          // 脉冲输出完成中断标志
  UINT INTCounter;          // 计数器中断标志
  UINT INTSensor;           // 传感器中断标志
  UINT INTComparator;       // 比较中断标志
  UINT INTAxis;             // 轴中断状态
}PCI1040_MAIN_STATUS, *PPCI1040_MAIN_STATUS;

```

## 第十二节、正常停止原因状态结构体介绍 (PCI1040\_NORMAL\_STOP\_FACTOR)

```

typedef struct _PCI1040_NORMAL_STOP_FACTOR
{
  UINT NSORG;               // 原点搜索, 0:没有停止 1: 正常停止
  UINT NSEZ;                // 编码器 Z 向标志, 0:没有停止 1: 正常停止
  UINT NSPSLD;              // 正向减速位置, 0:没有停止 1: 正常停止
  UINT NSMSLD;              // 负向减速位置, 0:没有停止 1: 正常停止
}PCI1040_NORMAL_STOP_FACTOR, *PPCI1040_NORMAL_STOP_FACTOR;

```

## 第十三节、错误停止原因状态结构体介绍 (PCI1040\_ERROR\_STOP\_FACTOR)

```

typedef struct _PCI1040_ERROR_STOP_FACTOR
{
  UINT ESPEL;               // 正向限位, 0:没有停止 1: 错误停止
  UINT ESMEL;               // 负向限位, 0:没有停止 1: 错误停止
  UINT ESALM;               // 在加/减速驱动中减速时, 0:没有停止 1: 错误停止
}PCI1040_ERROR_STOP_FACTOR, *PPCI1040_ERROR_STOP_FACTOR;

```

## 第十四节、Counter A/B 计数器参数设置结构体 (PCI1040\_COUNTER\_MODE)

```

typedef struct _PCI1040_COUNTER_MODE
{

```



```
UINT CountMode;      // 计数器模式设置, 0:内部逻辑计数 1:外部实位计数
UINT Direction;      // 计数方向, 0:正方向 1:反方向
UINT KeepCount;      // 是否保存原来的计数值(=0 为保持, =1 为清零)
}PCI1040_COUNTER_MODE, *PPCI1040_COUNTER_MODE;
```