

PCI-9014 编程手册

版权声明

本文档所有权归深圳市升立德科技有限公司(后面简称“升立德”)所有；升立德具有本产品及其软件的专利权、版权和其它知识产权。未经授权，任何单位和个人不得直接或者间接地复制、制造、加工、使用本产品及其相关部分。

升立德保留在不事先通知的情况下，修改本手册中的产品和产品规格等文件的权力

升立德全力维护本文档的正确性，但不承担由于本文档错误或使用本产品不当，所造成直接的、间接的、特殊的、附带的或相应产生的损失或责任。

目 录

1	操作原理.....	1
1.1	运动控制方式.....	1
1.1.1	脉冲输出方式.....	1
1.1.2	连续驱动.....	1
1.1.3	点位运动.....	2
1.1.4	原点查找.....	4
1.2	限位、原点状态.....	5
1.2.1	限位信号 (PEL,MEL)	6
1.2.2	原点信号 (ORG)	6
1.3	编码器输入.....	6
2	函数库使用.....	7
3	函数说明.....	7
3.1	初始化.....	8
3.2	输入/输出脉冲配置.....	9
3.3	速度/加速度设置参数.....	9
3.4	连续驱动.....	10
3.5	点位运动.....	11
3.6	原点查找.....	12
3.7	运动状态查询.....	12
3.8	I/O 状态控制及检测.....	13
3.9	信号有效电平的设定.....	14
3.10	位置计数器控制.....	14
3.11	通用 I/O 控制.....	15
3.12	版本信息查询.....	17
3.13	错误代码列表.....	17
4	修订记录.....	18

1 操作原理

1.1 运动控制方式

1.1.1 脉冲输出方式

PCI-9014 采用脉冲指令控制步进/伺服电机，输出两个信号：OUT, DIR；支持两种脉冲输出模式：

- (1) 脉冲/方向模式（即 PULSE/DIR 模式）
- (2) 双脉冲模式（即 CW/CCW 模式）

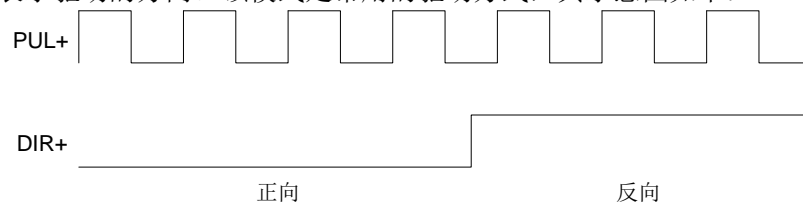
两种基本模式如下表所示：

模式	OUT 信号	DIR 信号
双脉冲模式	正向脉冲	反向脉冲
脉冲/方向模式	脉冲信号	方向信号

API 默认的输出方式为 PULSE/DIR 模式。

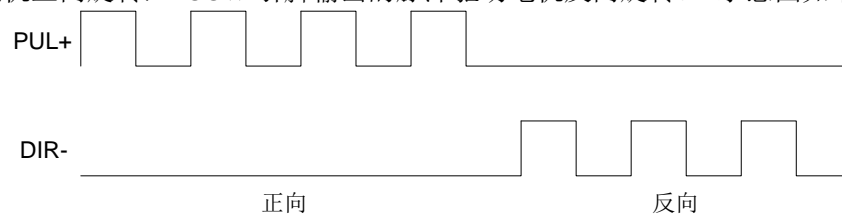
脉冲/方向模式

在该模式下，OUT 信号输出脉冲指令。脉冲的数目表示要移动的距离，脉冲的频率表示驱动速度。DIR 信号表示驱动的方向。该模式是常用的驱动方式，其示意图如下：



双脉冲模式

该模式下，OUT 和 DIR 引脚分别表示正向脉冲（CW）输出、反向脉冲(CCW)输出。CW 引脚输出的脉冲驱动电机正向旋转，CCW 引脚输出的脉冲驱动电机反向旋转。示意图如下：



相关的函数：p9014_set_pls_outmode.

注意：

- 1) 以上两个图中只画了PUL+与DIR+的信号，PUL-与PUL+，DIR-与DIR+的相位刚好相反（PUL+为0V时PUL-为5V，PUL+为5V时PUL-为0V）
- 2) 脉冲信号的占空比为 50%

1.1.2 连续驱动

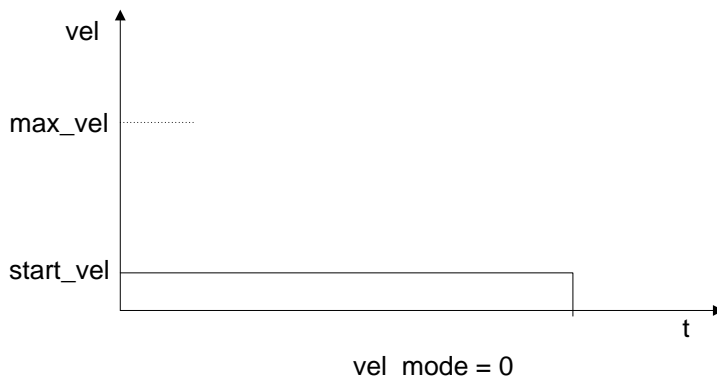
该模式用于启动控制轴以恒定速度持续运动；

当控制轴处于空闲状态时，发送连续驱动命令(p9014_vmove)将使控制轴以指定的速度持续运动，直至收到停止命令。

按照 vel_mode 参数值，可以分为三种模式：

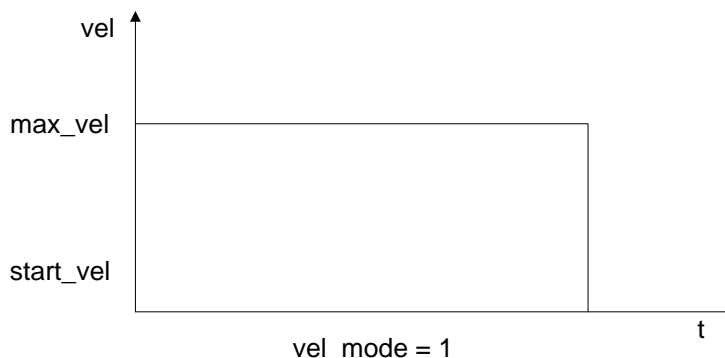
1) 模式 0

该模式下，控制轴以 `start_vel` 速度开始运动，收到停止命令（`p9014_stop`）后，停止运动；没有加速、减速过程；



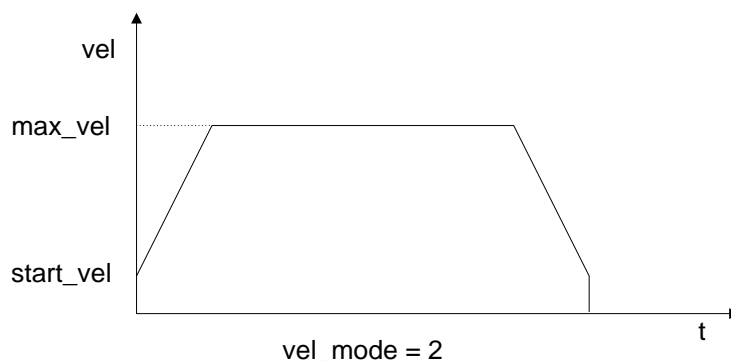
2) 模式 1

该模式下，控制轴以 `max_vel` 速度开始运动，收到停止命令（`p9014_stop`）后，停止运动；没有加速、减速过程；



3) 模式 2

在该模式下，控制轴从起始速度开始加速，直到最大速度；遇到减速停止命令后，减速至起始速度后停止运动。如果遇到紧急停止命令，没有减速过程，控制轴直接停止运动。



相关函数：

`p9014_set_t_profile`
`p9014_vmove`
`p9014_stop`

1.1.3 点位运动

该模式使用 T 型速度曲线将电机驱动指定距离，用于单个轴的点到点驱动。

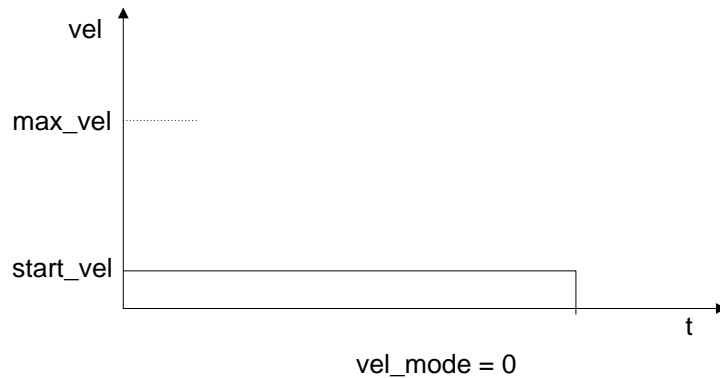
当控制轴处于空闲状态时，发送点位运动命令(`p9014_pmove`)将使控制轴以指定的速度运动一段指定的距离；在运动过程中，也可以使用 `p9014_stop` 函数使控制轴减速停止或者立即停止运动。

坐标方式可分为绝对坐标方式和相对坐标方式。在相对坐标方式下，使用位置偏移作为参数，位置偏移的符号决定了电机驱动的方向。在绝对坐标方式下，目标位置和当前坐标位置的差值符号决定了电机驱动的方向。加速度和减速度值不同时，构成非对称速度曲线。

点位运动按照 `vel_mode` 的值可以分为三种：

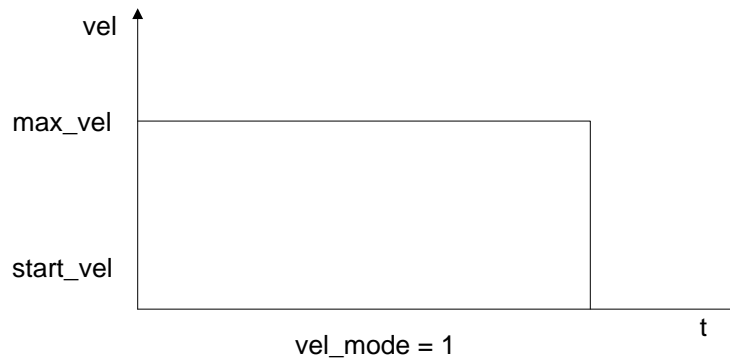
1) 模式 0

该模式下，控制轴以 `start_vel` 速度开始运动，运行指定距离后，停止运动；没有加速、减速过程；



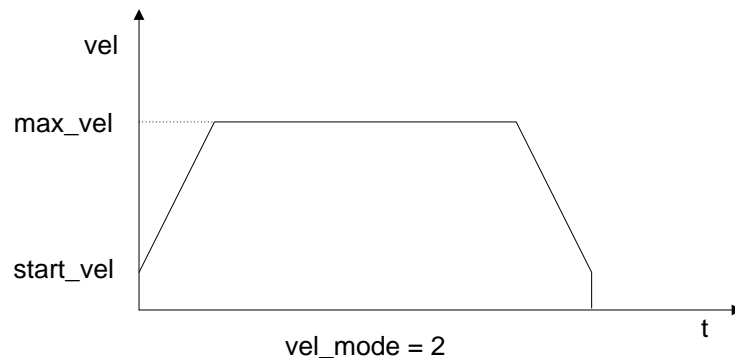
2) 模式 1

该模式下，控制轴以 `max_vel` 速度开始运动，运行指定距离后，停止运动；没有加速、减速过程；



3) 模式 2

在该模式下，控制轴从 `start_vel` 开始加速，直到最大速度，到减速点以后，自动开始减速。

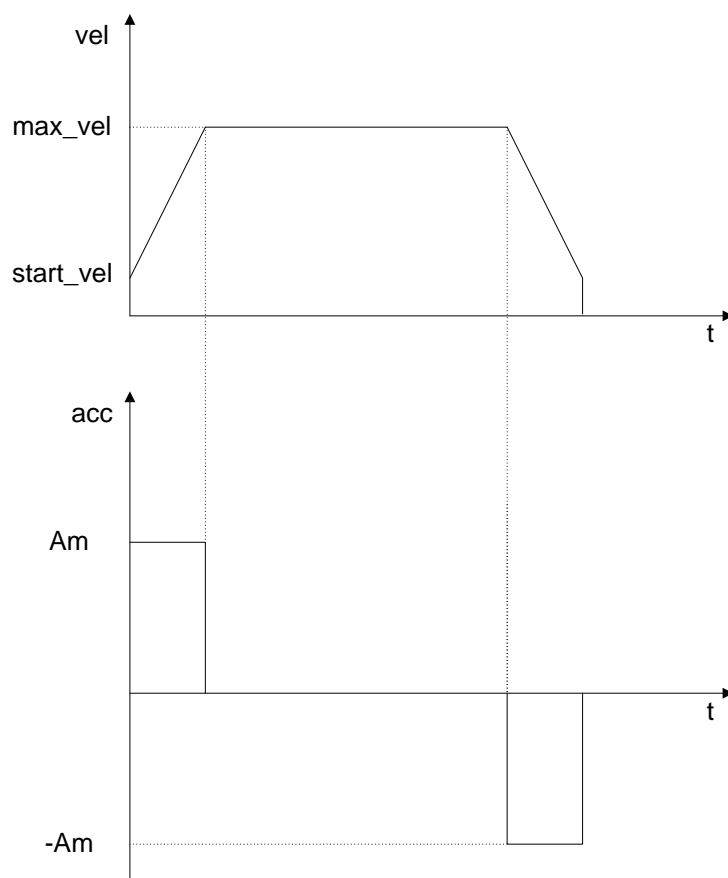


假设起始速度 `start_vel`，最大速度 `max_vel` 参数的单位是 pps，加速度 `acc`，减速度 `dec` 的单位为 pps/s 加速时间 `Tacc`，减速时间 `Tdec` 的计算公式为：

$$T_{acc} = (\max_vel - \text{start_vel}) / \text{acc}$$

$$T_{dec} = (\max_vel - \text{start_vel}) / \text{dec}$$

下图演示了 T 型速度曲线的速度-时间关系、加速度-时间关系。



T型速度曲线示意图

曲线下面的面积为移动距离。

1.1.4 原点查找

当控制轴处于空闲状态时，发送原点查找命令(p9014_home_move)将使控制轴启动原点查找，原点条件满足后，停止运动。原点查找过程是否结束，可以通过 p9014_get_motion_status 函数来查询。

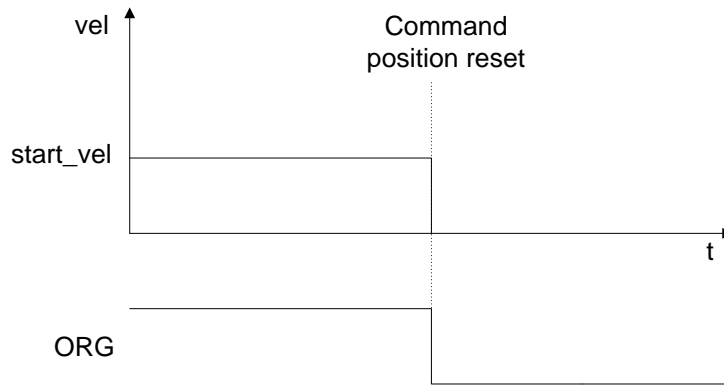
查找原点时的驱动方向，由 p9014_home_move 中 PlusDir 参数来决定。

P9014_home_config 可用来设置原点查找模式、ORG 信号有效电平、编码器 INDEX 信号有效边沿。

PCI-9014 支持两种原点查找模式：

(1) 模式 0, 低速模式

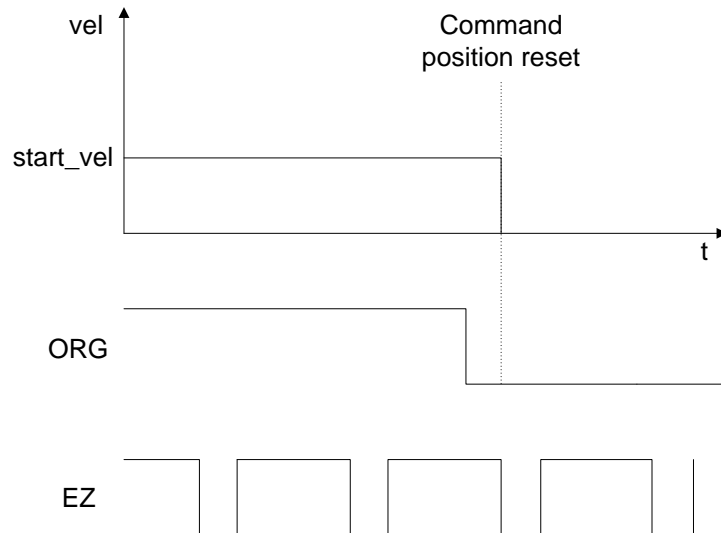
Home_mode = 0; 只有 ORG 有效，没有加速过程。有效的 ORG 信号立即使控制轴立即停止运动，停止过程没有减速；在 ORG 有效的边沿，位置计数器被清零。



回零模式0(ORG开关常开、低有效)

(2) 模式 2, 低速模式

Home_mode = 2; ORG 和 index 信号同时有效, 启动没有加速过程。ORG 信号有效后, 然后收到有效的 index 信号, 控制轴停止运动, 原点查找完成; 原点查找结束后, 位置计数器被清零;



回零模式2(ORG开关常开、低有效; EZ低电平有效)

相关函数:

p9014_set_t_profile
p9014_home_config
p9014_home_move

1.2 限位、原点状态

PCI-9014 对限位(PEL, MEL)、原点(ORG)进行了滤波处理; 这些信号的状态可以通过 p9014_get_io_status 函数进行查询。

1.2.1 限位信号（PEL,MEL）

当控制轴正向运动时，如果 PEL 有效，则立即停止发送脉冲；当控制轴反向驱动时，如果 MEL 有效，则立即停止发送脉冲；限位信号有效导致控制轴停止运动后，必须向控制轴发送反向驱动的命令，否则控制轴不能运动。这样的机制可以有效保护机械部分，提高了安全性。

限位开关的类型可以通过 `p9014_set_el_level` 来设置：常开类型、常闭类型。PCI-9014 默认设置为常开类型。常闭类型的开关在正常情况下闭合，只有机械部分撞到位限位开关后，开关才断开，控制轴停止运动；如果有其它异常情况，比如限位开关的信号线断开的时候，也会认为是限位开关有效，控制轴不能运动，这样可以避免设备在故障情况下继续工作，提高机械的安全性，所以建议限位开关使用常闭类型。

1.2.2 原点信号（ORG）

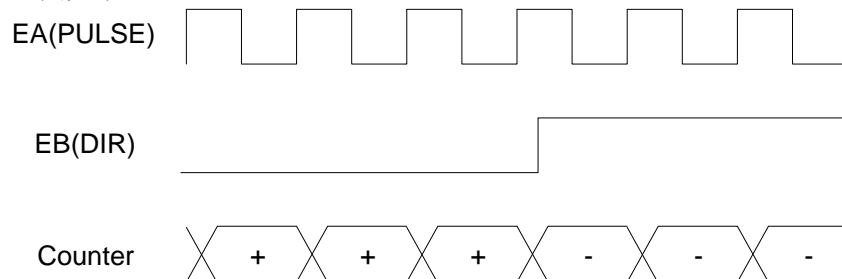
原点信号用于原点查找。在模式 0、1 下，原点查找只使用 ORG 信号。在模式 2 下，原点查找除了使用 ORG 信号外，还要使用编码器输入的 index(EZ)信号。

ORG、index 和回零模式可以通过 `p9014_set_home_config` 函数进行配置。

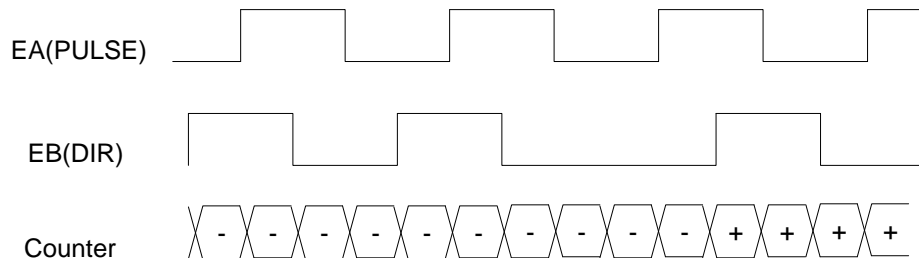
1.3 编码器输入

PCI-9014 运动控制卡为每个轴提供了 2 个 28 位的计数器，用来对输出脉冲（计数器 0）和编码器反馈脉冲（计数器 1）进行计数。其中计数器 1 支持两种格式输入：PULSE/DIR，4XAB，通过 `p9014_set_pls_iptmode` 函数来设置。

在 PULSE/DIR 输入模式下，EA 上升沿时 EB 为低电平，计数器值加 1；EA 上升沿时 EB 为高电平，计数器减 1；示意图如下：



在 4XAB 输入格式下，如果 EA 信号相位超前 EB 信号 90 度，则表示电机在正向运转，计数器值增加；如果 EB 信号相位超前 EA 信号 90 度，则表示电机在反向运转，计数器值递减；示意图如下：



计数器的初值可以通过 `p9014_set_pos` 函数来设置；计数器值可以通过 `p9014_get_pos` 函数来读出；

计数器溢出问题：由于计数器长度为 28 位，使用二进制补码，计数范围在 -134217728 ~ 134217727 ;如果计数器值已经计到 134217727 后又接收到正向脉冲，则产生向上溢出，计数器值变为-134217728; 如果计数器值已经计到-134217728 后又接收到反向脉冲，则产生向下溢出，计数器值变为 134217727; 用户的应用程序应该定时去判别该异常情况。

相关函数：

```
p9014_set_pls_iptmode
p9014_set_pos
p9014_get_pos
```

2 函数库使用

PCI-9014 提供了 Windows2000/xp 驱动程序、动态链接库供用户使用。用户使用前先要安装驱动程序，具体安装步骤请参见 9014 用户手册中的安装说明；

在 Windows 系统下，用户可以使用任何能够支持动态链接库的开发工具来开发应用程序。下面分别以 Visual C++为例讲解如何在这些开发工具中使用运动控制器的动态链接库。

使用 VC++ 6.0 来开发运动控制程序：

1. 启动 Visual C++ ， 新建一个工程；
2. 将安装目录的 Library 文件夹内动态链接库、头文件和 lib 文件复制到工程文件夹中；
3. 选择“Project” 菜单下的“Settings…” 菜单项；
4. 切换到“Link” 标签页， 在“Object/library modules” 栏中输入 lib 文件名 pci9014.lib；
5. 在应用程序文件中加入函数库头文件的声明， 例如：
#include “pci_9014.h”
6. 至此， 用户就可以在 Visual C++ 中调用函数库中的任何函数， 开始编写应用程序

3 函数说明

函数列表：

p9014_initial	板卡初始化
p9014_close	关闭板卡
p9014_set_pls_outmode	脉冲输出模式设定
p9014_set_pls_iptmode	脉冲输入模式设定
p9014_set_t_profile	运动 T 曲线参数设定
p9014_set_s_profile	运动 S 曲线参数设定
p9014_vmove	连续运动
p9014_stop	运动停止
p9014_pmove	点位运动
p9014_set_home_config	原点查找参数设定
p9014_home_move	原点查找动作
p9014_get_motion_status	获取当前运动状态
p9014_get_current_speed	获取当前运动速度
p9014_get_io_status	获取当前运动 I/O 状态
p9014_set_el_level	限位电平设定
p9014_set_alram	设置对应 DI 为 ALARM 输入端
p9014_get_pos	获取当前位置
p9014_set_pos	设定当前位置

p9014_set_do	设定 D0 端口输出状态
p9014_set_do_bit	设定 D0 端口中指定位输出状态
p9014_get_do	获取 D0 端口状态
p9014_get_di	获取 DI 端口状态
p9014_get_di_bit	获取 DI 端口中指定位的状态
p9014_get_version	获取逻辑版本等信息
p9014_get_vevision	获取运动功能版本信息

PCI-9014 的 API 依据功能，可以分为几大类：

- 初始化
- 连续运动
- 点位运动
- 原点查找
- I/O 控制
- 位置计数器控制

大部分 API 使用了 card_no, axis 参数。PCI-9014 里面使用一个拨码开关 S1 来设置板子的 ID 号(出厂默认设置为 0)，一旦 ID 号设定后，该卡上的 X, Y, Z, U 四个轴的编号(Axis)为

X	card_no*4 + 0
Y	card_no *4 + 1
Z	card_no *4 + 2
U	card_no *4 + 3

该章节详细描述 PCI-9014 API 函数参数及用法。为了提高程序的可移植性，我们统一使用了一些类型定义，建议用户在程序中尽量使用这些定义。

类型	描述	范围
I32	32 位有符号整数	-2147483648 ~ 2147483647
U32	32 位无符号整数	0 ~ 4294967295

3.1 初始化

I32 p9014_initial(I32* pCard_count, I32* pBoard_id)

功能描述：

查找并初始化系统里面所有的 PCI-9014，对板卡进行控制的首要条件。

调用该函数后，PCI-9014 默认的配置：脉冲输出为 PULSE/DIR 模式，两个位置计数器被清零，计数器 1 的输入格式设置为 4XAB 类型。

参数：

- *pCard_count 查到的 PCI-9014 数目
- *pBoard_id 有*existCards 个元素的数组，包含查找到的 PCI-9014 的 ID 号，建议使用含有 16 个元素的 I32 类型数组作为该参数

返回值：

正常返回 0；

出现错误时返回非 0 值；
具体含义请参考错误代码列表或者 ErrorCodeDef.h 里面的 Return code definition 部分定义。

I32 p9014_close(void)

功能描述：

关闭系统里面所有的 PCI-9014。

参数：

无

返回值：

正常返回 0；

出现错误时返回非 0 值；

具体含义请参考错误代码列表或者 ErrorCodeDef.h 里面的 Return code definition 部分定义。

3.2 输入/输出脉冲配置

I32 p9014_set_pls_outmode(I32 axis, I32 pls_outmode)

功能描述：

设置 OUT, DIR 引脚输出脉冲方式；初始化后，API 默认的脉冲输出为 PULSE/DIR 模式

参数：

Axis 轴号

Pls_outmode pls_outmode=0, PULSE/DIR 输出.
 pls_outmode=1, CW/CCW 输出

返回值：

正常返回 0；

出现错误时返回非 0 值；

具体含义请参考错误代码列表或者 ErrorCodeDef.h 里面的 Return code definition 部分定义。

I32 p9014_set_pls_iptmode(I32 axis, I32 pls_iptmode)

功能描述：

设置外部脉冲编码器输入信号类型； 共有两种类型可选： PULSE/DIR , 4X AB。初始化后，API 默认的输入信号类型为 4X AB。

参数：

Axis 轴号

Pls_iptmode 0 - . PULSE/DIR
 1 - 4X AB

返回值：

正常返回 0；

出现错误时返回非 0 值；

具体含义请参考错误代码列表或者 ErrorCodeDef.h 里面的 Return code definition 部分定义。

3.3 速度/加速度设置参数

I32 p9014_set_t_profile(I32 axis, double start_vel, double max_vel, double acc, double dec)

功能描述：

配置控制轴使用 T 型速度曲线加减速，并设置相应的起始速度、最大速度、加速度、减速度参数。

调用该函数后，该轴的点位运动、连续运动、回零运动、减速停止将使用这些速度、加速度参数进行驱动。

参数：

axis	轴号
start_vel	启动速度, 单位: pps, 最大值为 1000000pps
max_vel	最大速度, 单位: pps, 最大值为 1000000pps
acc	加速度, 单位: pps/s, 最大值为 100000000pps/s
dec	减速度, 单位: pps/s, 最大值为 100000000pps/s

返回值:

正常返回 0;

出现错误时返回非 0 值;

具体含义请参考错误代码列表或者 ErrorCodeDef.h 里面的 Return code definition 部分定义。

I32 p9014_set_s_profile(I32 axis, double start_vel, double max_vel, double acc, double dec, double jerk_percent)

功能描述:

配置控制轴使用 S 曲线加减速并设置 S 曲线相关参数

调用该函数后, 该轴的点位运动、连续运动、回零运动、减速停止将使用这些速度、加速度参数进行驱动。

注意: 只有逻辑版本在 0x0112 及以上版本的控制卡才支持 S 曲线。

参数:

axis	轴号
start_vel	启动速度, 单位: pps, 最大值为 1000000pps
max_vel	最大速度, 单位: pps, 最大值为 1000000pps
acc	加速度, 单位: pps/s, 最大值为 100000000pps/s
dec	减速度, 单位: pps/s, 最大值为 100000000pps/s
jerk_percent	加速/减速过程使用 S 曲线的比例, 范围为 0~1; 1 表示全部 S 曲线加减速; 0 表示没有 S 曲线部分(也就是 T 型曲线加减速)

返回值:

正常返回 0;

出现错误时返回非 0 值;

具体含义请参考错误代码列表或者 ErrorCodeDef.h 里面的 Return code definition 部分定义。

3.4 连续驱动

I32 p9014_vmove(I32 axis, I32 plus_dir, I32 vel_mode = 2)

功能描述:

启动控制轴进行连续驱动, 运动方向由 plus_dir 决定。

如果 vel_mode 为 0, 则该轴以 p9014_set_t_profile 中的 start_vel 为速度进行驱动;

如果 vel_mode 为 1, 则该轴以 p9014_set_t_profile 中的 max_vel 为速度进行驱动;

如果 vel_mode 为 2, 则该轴从 start_vel 加速到 max_vel 后, 以 max_vel 为速度进行驱动; 遇到 p9014_stop 命令或者限位有效后, 控制轴才停止运动。

在运动过程中, 限位信号从无效变为有效时, 控制轴停止运动, 控制卡硬件有锁存机制, 将状态锁存, 需要控制轴反向驱动后才能清除锁存的状态; 在无清除锁存状态前, 控制轴无法继续向同方向运动

参数:

axis	轴号
plus_dir	驱动方向, 1 为正向驱动, 0 为反向驱动;
vel_mode	速度模式; 0 表示以 start_vel 为速度进行驱动, 中间没有加速过程; 1 表示以 max_vel 为速度进行驱动, 中间没有加速过程;

2 表示从 start_vel 加速到 max_vel,中间有加速过程;

返回值:

正常返回 0;

出现错误时返回非 0 值;

具体含义请参考错误代码列表或者 ErrorCodeDef.h 里面的 Return code definition 部分定义。

I32 p9014_stop(I32 axis, I32 EmgStop);

功能描述:

使控制轴减速停止或立即停止。

如果控制轴减速停止(EmgStop = 0), 将使用 p9014_set_t_profile 中的 dec 参数进行减速, 减速到 start_vel 后, 停止运动。

如果控制轴立即停止(EmgStop = 1), 则没有减速过程, 立即停止运动。

参数:

axis 轴号

EmgStop 是否紧急停止; 0 表示减速停止; 1 表示紧急停止, 没有减速过程。

返回值:

正常返回 0;

出现错误时返回非 0 值;

具体含义请参考错误代码列表或者 ErrorCodeDef.h 里面的 Return code definition 部分定义。

3.5 点位运动

I32 p9014_pmove(I32 axis, I32 dist, I32 dist_mode = 0, I32 vel_mode = 2)

功能描述:

启动控制轴进行定脉冲驱动, 并立即返回; 运动是否完成, 可以通过 p9014_get_motion_status 函数进行你个查询。

Dist_mode 参数用来设置 dist 使用何种坐标模式 (相对坐标、绝对坐标); 在相对坐标方式下, 驱动的方向由 dist 决定; dist 大于 0 时, 正向驱动; dist 小于 0 时, 反向驱动; 在绝对坐标方式下, dist 是表示目标位置, 运动方向由目标位置和当前位置差值的符号决定。

在运动过程中, 限位信号从无效变为有效时, 控制轴停止运动, 控制卡硬件有锁存机制, 将状态锁存, 需要控制轴反向驱动后才能清除锁存的状态; 在无清除锁存状态前, 控制轴无法继续向同方向运动

Vel_mode 参数表示该命令使用的速度模式:

如果 vel_mode 为 0, 则该轴以 p9014_set_t_profile 中的 start_vel 为速度进行驱动;

如果 vel_mode 为 1, 则该轴以 p9014_set_t_profile 中的 max_vel 为速度进行驱动;

如果 vel_mode 为 2, 则该轴从 start_vel 加速到 max_vel 后, 以 max_vel 为速度进行驱动, 运行到减速点后, 控制轴开始减速。 在运行过程中, 可以使用 p9014_stop 函数使控制轴停止运动。

参数:

axis 轴号

dist 驱动的距离 (相对于当前位置);

dist_mode dist 参数的坐标模式, 0 表示相对坐标; 1 表示绝对坐标;

vel_mode 速度模式

0 表示以 start_vel 为速度进行驱动, 中间没有加速过程;

1 表示以 max_vel 为速度进行驱动, 中间没有加速过程;

2 表示从 start_vel 加速到 max_vel, 中间有加速、减速过程;

返回值:

正常返回 0;

出现错误时返回非 0 值;

具体含义请参考错误代码列表或者 ErrorCodeDef.h 里面的 Return code definition 部分定义。

3.6 原点查找

I32 p9014_set_home_config(I32 axis, I32 mode, I32 org_level, I32 ez_level)

功能描述:

配置原点开关有效电平、编码器 index 信号有效电平、回零模式; 这些参数在 p9014_home_move 函数中将使用到。

在模式 0 (home_mode = 0) 情况下, 只需要 ORG 信号有效即可, 对 ez_level 可以给任意值; 而在 2 模式下, 需要使用到编码器输入的 index 信号, 因而需要指定 index 信号的有效电平;

参数:

axis	轴号
mode	回零模式, 范围: 0~2; home_mode = 0 只有 ORG 有效, 没有加速过程。有效的 ORG 信号立即使控制轴立即停止运动, 停止过程没有减速; 在 ORG 有效的边沿, 位置计数器被清零; home_mode = 2 ORG 和 index 信号同时有效, 启动没有加速过程。ORG 信号有效后, 然后收到有效的 index 信号, 控制轴停止运动, 原点查找完成; 原点查找结束后, 位置计数器被清零;
org_level	原点信号的有效电平; 0 - 低有效; 1 - 高有效;
ez_level	编码器的 index 信号有效电平; 0 - 低有效; 1 - 高有效;

返回值:

正常返回 0;

出现错误时返回非 0 值;

具体含义请参考错误代码列表或者 ErrorCodeDef.h 里面的 Return code definition 部分定义。

I32 p9014_home_move (I32 axis, I32 PlusDir)

功能描述:

驱动指定的轴回零, 并立即返回。

回零是否完成, 可以通过 p9014_motion_done 查询状态来完成。

参数:

axis	轴号
plusDir	原点查找的运动方向; plusDir = 1 控制轴正向运动启动原点查找 plusDir = 0 控制轴反向运动启动原点查找

返回值:

正常返回 0;

出现错误时返回非 0 值;

具体含义请参考错误代码列表或者 ErrorCodeDef.h 里面的 Return code definition 部分定义。

3.7 运动状态查询

I32 p9014_get_motion_status(I32 axis, U32* pStatus)

功能描述:

查询指定轴的状态, 是正在运动, 或者运动停止, 处于空闲状态。

注意：停止运动的原因可能为到达目标位置，停止命令或者遇到错误，如 PEL, MEL 信号有效，因此用户程序需要查询 PEL, MEL 的状态才能获取控制轴运动停止的原因。

参数：

Axis 轴号
*pStatus 表示运动状态；
 0 表示控制轴运动完成，处于空闲状态；
 1 表示控制轴正在运动

返回值：

正常返回 0；
出现错误时返回非 0 值；
具体含义请参考错误代码列表或者 ErrorCodeDef.h 里面的 Return code definition 部分定义。

I32 p9014_get_current_speed(I32 axis, double* pSpeed)

功能描述：

获取指定轴的当前速度

参数：

axis 轴号
*pSpeed 读到的当前速度，单位： pps

返回值：

正常返回 0；
出现错误时返回非 0 值；
具体含义请参考错误代码列表或者 ErrorCodeDef.h 里面的 Return code definition 部分定义。

3.8 I/O 状态控制及检测

I32 p9014_get_io_status(I32 axis, U32* pStatus)

功能描述：

获取与驱动轴相关 I/O（如限位信号、原点信号）的状态

参数：

axis 轴号
* pStatus I/O 状态， 具体如下表：

各个 bit 定义如下：

位	信号	描述
0	PEL	正极限信号，1 表示正限位有效， 0 表示无效
1	MEL	负极限信号，1 表示负限位有效， 0 表示无效
2	ORG	原点信号，1 表示原点有效， 0 表示无效
3	EZ	Index 信号，1 表示 EZ 为高电平， 0 表示为低电平
4	EMG	EMG 信号，1 表示 EMG 输入为高， 0 表示输入为低；该信号为低电平有效。
5	DIR	当前运动方向，0 表示当前运动方向为正向， 1 表示当前运动方向为负向。

返回值:

正常返回 0;

出现错误时返回非 0 值;

具体含义请参考错误代码列表或者 ErrorCodeDef.h 里面的 Return code definition 部分定义。

3.9 信号有效电平的设置

I32 p9014_set_el_level(I32 axis, I32 active_level)

功能描述:

设定限位信号有效电平, 可以为高电平有效, 也可以为低电平有效, 默认为低电平有效。

参数:

axis	轴号
active_level	限位信号电平选择
	0 低电平有效
	1 高电平有效

返回值:

正常返回 0;

出现错误时返回非 0 值;

具体含义请参考错误代码列表或者 ErrorCodeDef.h 里面的 Return code definition 部分定义。

I32 p9014_set_alram(I32 axis, I32 enable, I32 active_level)

功能描述:

设定告警信号有效电平, 可以为高电平有效, 也可以为低电平有效, 默认为低电平有效。

DI0—3 可在硬件上连接 XYZU 轴的告警 ALM 信号。用此函数可查询轴告警状态电平

参数:

axis	轴号
enable	告警信号使能
active_level	限位信号电平选择
	0 低电平有效
	1 高电平有效

返回值:

正常返回 0;

出现错误时返回非 0 值;

具体含义请参考错误代码列表或者 ErrorCodeDef.h 里面的 Return code definition 部分定义。

3.10 位置计数器控制

I32 p9014_get_pos(I32 axis, I32 cntr_no, I32* pPos)

功能描述:

读取控制轴的位置计数器, 该计数器可以为输出脉冲计数器(cntr_no = 0)或者编码器反馈脉冲位置计数器(cntr_no = 1); 其对应的位置分别为指令脉冲位置 (逻辑位置), 或者编码器反馈脉冲位置 (实际位置)。

注意: 计数器使用 28 位补码进行计数, 因此可能会溢出, 请参考 “操作原理” 一节的 “编码器输入” 部分描述。

参数:

axis	轴号
------	----

cntr_no 输出脉冲计数器/编码器反馈脉冲位置计数器选择
 0 输出脉冲计数器
 1 编码器反馈脉冲位置计数器

***pPos** 位置计数器的值, 范围在-134217728 ~ 134217727;

返回值:
 正常返回 0;
 出现错误时返回非 0 值;
 具体含义请参考错误代码列表或者 ErrorCodeDef.h 里面的 Return code definition 部分定义。

I32 p9014_set_pos(I32 axis, I32 cntr_no, I32 pos)

功能描述:
 设置位置计数器

参数:
 axis 轴号
 cntr_no 输出脉冲计数器/编码器反馈脉冲位置计数器选择
 0 输出脉冲计数器
 1 编码器反馈脉冲位置计数器
 Pos 要设置的值, 范围在-134217728 ~ 134217727。

返回值:
 正常返回 0;
 出现错误时返回非 0 值;
 具体含义请参考错误代码列表或者 ErrorCodeDef.h 里面的 Return code definition 部分定义。

3.11 通用 I/O 控制

I32 p9014_set_do(I32 card_no, U32 data)

功能描述:
 设置 DO 端口输出状态

参数:
 card_no 卡号
 data 状态输出值, 其中的 bit 0 ~15 分别对应端子上的 DO0~DO15
 当 data 中的位为 1 时, 对应端子输出截止;
 当 data 中的位为 0 时, 对应端子输出导通;

返回值:
 正常返回 0;
 出现错误时返回非 0 值;
 具体含义请参考错误代码列表或者 ErrorCodeDef.h 里面的 Return code definition 部分定义。

I32 p9014_set_do_bit(I32 card_no, U32 bit_no, U32 data)

功能描述:
 设置 DO 端口中指定位的状态。

参数:
 card_no 卡号
 bit_no 对应的位(bit)号, 范围 0~15, 分别对应端子上的 DO0~DO15。
 data 对应 DO 端子输出值,
 1 对应端子输出截止;
 0 对应端子输出导通;

返回值:
 正常返回 0;

出现错误时返回非 0 值；
具体含义请参考错误代码列表或者 ErrorCodeDef.h 里面的 Return code definition 部分定义。

I32 p9014_get_do(I32 card_no, U32* pData)

功能描述：

读取 DO 端口的状态。

参数：

card_no	卡号
*pData	状态输出值，其中的 bit 0~15 分别对应端子上的 DO0~DO15 当 data 中的位为 1 时，对应端子输出截止； 当 data 中的位为 0 时，对应端子输出导通；

返回值：

正常返回 0；
出现错误时返回非 0 值；
具体含义请参考错误代码列表或者 ErrorCodeDef.h 里面的 Return code definition 部分定义。

I32 p9014_get_di(I32 card_no, U32* pData)

功能描述：

查询 DI 端口的状态。

DI0—3 可在硬件上连接 XYZU 轴的告警 ALM 信号。用此函数可查询轴告警状态

参数：

card_no	卡号
*pData	DI 端口状态，其中的 bit 0~15 分别对应端子上的 DI0~DI15 当*pData 中的位为 1 时，对应端子输入高电平； 当*pData 中的位为 0 时，对应端子输入低电平；

返回值：

正常返回 0；
出现错误时返回非 0 值；
具体含义请参考错误代码列表或者 ErrorCodeDef.h 里面的 Return code definition 部分定义。

I32 p9014_get_di_bit(I32 card_no, U32 bit_no, U32* pData)

功能描述：

查询 DI 端口中指定位的状态

DI0—3 可在硬件上连接 XYZU 轴的告警 ALM 信号。用此函数可查询轴告警状态

参数：

card_no	卡号
bit_no	对应的位(bit)号， 范围 0~15,分别对应端子上的 DI0~DI15。
*Pdata	对应 DI 端子输入值， 1 对应端子输入高电平； 0 对应端子输入低电平；

返回值：

正常返回 0；
出现错误时返回非 0 值；
具体含义请参考错误代码列表或者 ErrorCodeDef.h 里面的 Return code definition 部分定义。

3.12 版本信息查询

I32 p9014_get_version(I32 card_no, U32* pApi_ver, U32* pDriver_ver, U32 *pLogic_ver)

功能描述:

版本号信息查询，可以查询 API，驱动程序和逻辑的版本。

参数:

card_no	卡号
*pApi_ver	API 版本，其中的 bit0~7 为副版本号， bit8~15 为主版本号；
* pDriver_ver	驱动程序版本，其中的 bit0~7 为副版本号， bit8~15 为主版本号；
* pLogic_ver	逻辑软件版本，其中的 bit0~7 为副版本号， bit8~15 为主版本号；

返回值:

正常返回 0；

出现错误时返回非 0 值；

具体含义请参考错误代码列表或者 ErrorCodeDef.h 里面的 Return code definition 部分定义。

I32 p9014_get_vevision(I32 card_no, U32 *pLogic_revision)

功能描述:

逻辑版本信息查询。

参数:

card_no	卡号
* pLogic_revision	运动控制功能的版本号；

返回值:

正常返回 0；

出现错误时返回非 0 值；

具体含义请参考错误代码列表或者 ErrorCodeDef.h 里面的 Return code definition 部分定义。

3.13 错误代码列表

错误代码	定义
1	参数无效
2	分配地址出错
6	未知错误
7	通讯失败
12	设备已经打开
100	无法打开 PCI 设备
201	未知运动错误
202	轴处于运行状态
205	限位有效或告警有效

4 修订记录

日期	版本	修改说明
2008-10-25	1.00	创建
2009-11-4	1.01	删除插补、连续插补部分内容
2011-5-17	1.02	删除回零模式 1 的描述
2013-5-15	1.03	增加限位信号有效电平の設定 增加 p9014_set_s_profile 的描述 修改 p9014_get_io_status 的描述