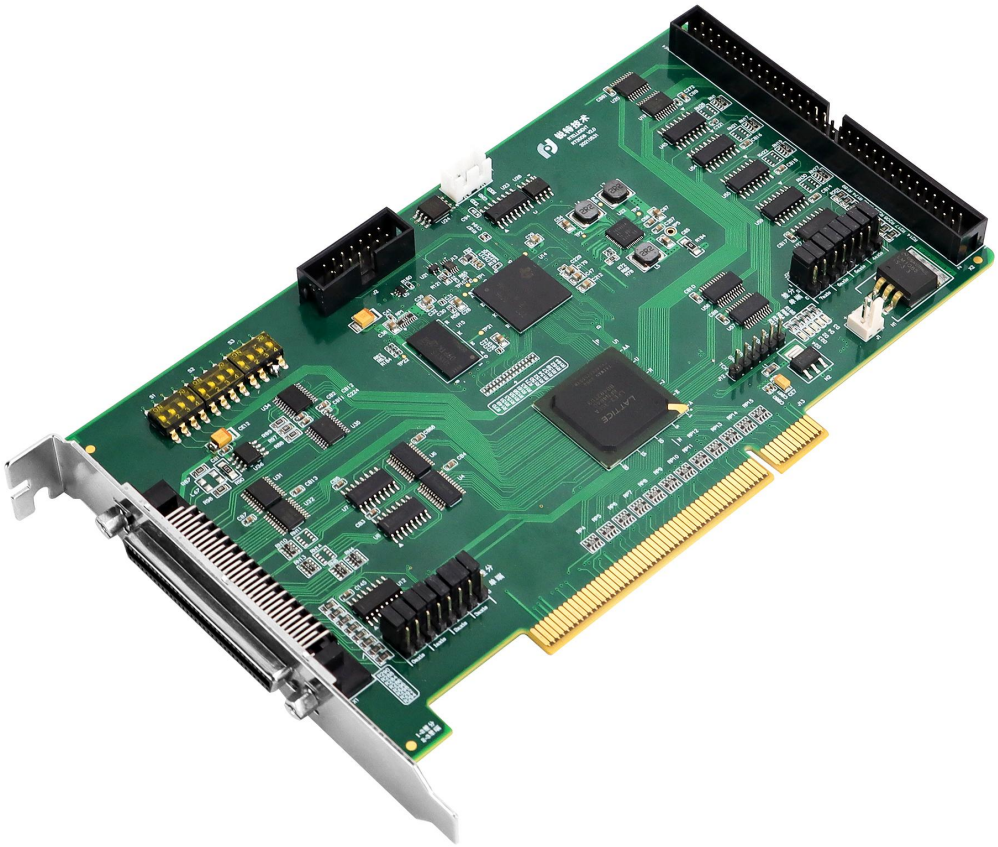
锐特 RT2008 运动控制卡使用手册



前言

感谢您购买锐特技术 RT2008 运动控制卡产品并阅读本使用手册。关于本手册有以下内容请知悉：

主要目的

通过阅读本手册，了解锐特技术 RT2008 运动控制卡的基本信息，完成运动控制卡的安装与运动控制系统的基本测试。

主要内容

使用手册主要包含锐特技术 RT2008 运动控制卡的产品信息，安装连接，软件调试与函数说明等内容。

服务对象

对运动控制卡的使用有一定了解的相关技术人员。

服务范围

购买本产品，锐特技术为您提供以下服务：

- 指导安装调试与试运行
- 指导保养维护
- 故障的维修与技术支持

安全注意事项

请熟读使用说明书以安全、正确地使用本产品。

运动中的设备有危险！使用者应确保在机器设计中有安全保护机制与有效的故障处理。如果没有执行必要的安全措施操作或者错误操作时，不仅会引起本产品的故障和损伤，有可能还会造成使用者（包括且不限于安装、操作、检查者等）受伤、死亡和重大事故。

保修期限

本产品保修期限为由本公司出厂后 18 个月。

保修范围

在保修期限内，正常使用下而发生的故障，由本公司负责免费维修，但不包括以下事项：

- 地震，火山，洪水，风暴，雷电，火灾或其他天灾与人祸造成的损害。
- 使用者不当的安装或使用。
- 未经本公司同意而进行的改装。
- 不完全或者错误的维护与检查。

另本公司负责自身产品的故障维修，但并不负责因产品故障引起的其他损失。

版权声明

锐特技术保留在不事先通知的情况下对本手册内容进行修改的权力。

技术支持

您可以通过以下方式获得我们的技术支持：

地址：深圳市宝安区固戍航空路庄边工业园厂房 B 栋 3 层

邮编：518100

电话：400-6822-996，0755-29503086

邮箱：support@szruitech.com

目录

前言.....1
第 1 章 产品概述.....6
1.1. 产品简介.....6
1.2. 型号说明.....6
1.3. 产品图片.....7
1.3.1. RT2008 主卡.....7
1.3.2. RT2008EX 接线盒.....8
1.3.3. 转接组件.....8
1.4. 功能说明.....9
第 2 章 硬件接口及电路.....11
2.1. 主卡接口说明.....11
2.2. 接线盒接口定义.....11
2.2.1. 轴信号的接口.....11
2.2.2. 通用数字输入输出信号、原点信号和限位信号接口.....13
2.2.3. 急停信号接口.....18
2.2.4. 辅助编码器接口.....19
第 3 章 快速启动指南.....21
3.1. 硬件安装步骤.....21
3.1.1. RT2008 硬件设置.....21
3.1.2. 跳线开关的设置.....21
3.1.3. 拨码开关设置.....22
3.2. 硬件安装.....24
3.3. 在 Windows 环境下，安装控制卡的驱动程序.....25
3.3.1. 驱动运行环境安装.....25
3.3.2. 安装证书.....25
3.3.3. 驱动安装.....28
3.4. 应用软件开发方法.....29
3.4.1. 基于 WINDOWS 平台的应用软件结构.....29

3.5. 实现基本功能的方法及相关函数.....31
3.5.1. 限位开关及急停开关的设置.....31
3.5.2. 回原点运动的实现.....32
3.5.3. 点位运动的实现.....38
3.5.4. 连续运动的实现.....42
3.5.5. 插补运动的实现.....43
3.5.6. PVT 运动功能的实现.....45
3.5.7. 异常减速停止时间设置功能的实现.....51
3.5.8. 手轮运动功能的实现.....51
3.5.9. 编码器检测的实现.....52
3.5.10. 检测轴到位状态功能的实现.....53
3.5.11. 通用 I/O 控制的实现.....54
3.5.12. 位置比较功能的实现.....56
3.5.13. 高速位置锁存功能的实现.....59
3.5.14. 软着陆的实现.....62
第 4 章 软件调试.....65
4.1. 概述.....65
4.2. 功能描述.....65
4.3. 启动与 I/O 检测.....66
4.3.1. 启动.....66
4.3.2. I/O 检测.....67
4.4. 运动测试.....68
4.4.1. 单轴运动.....68
4.4.2. 多轴运动.....70
4.4.3. 手轮测试.....71
4.4.4. PVT 测试.....72
4.4.5. 一维比较.....73
4.4.6. 二维比较.....75
4.4.7. 原点锁存.....76
4.4.8. 高速锁存.....78

4.5. 参数配置.....79
第 5 章 函数库详解.....80
5.1. 板卡设置函数.....80
5.2. 脉冲模式设置函数.....84
5.3. 回原点运动函数.....85
5.4. 限位开关设置函数.....88
5.5. 位置计数器控制函数.....91
5.6. 运动状态检测及控制相关函数.....92
5.7. 单轴运动速度曲线设置函数.....95
5.8. 单轴运动函数.....97
5.9. 插补速度曲线设置函数.....100
5.10. 插补运动函数.....102
5.11. PVT 运动函数.....103
5.12. 伺服驱动专用接口函数.....107
5.13. 通用输入输出 IO 函数.....111
5.14. 手轮功能函数.....113
5.15. 编码器函数.....115
5.16. 高速位置锁存函数.....118
5.17. 位置比较函数.....123
5.18. 异常信号接口函数.....129
5.19. 检测轴到位状态函数.....131
5.20. 软着陆函数.....133
5.21. 运动函数错误码说明.....136

第 1 章 产品概述

1.1. 产品简介

锐特公司生产的 RT2008 运动控制卡，可以实现多轴插补运动，高速的点位运动控制。其核心由 ARM 和 FPGA 组成，支持 8 轴位置比较和锁存，可实现 8 轴脉冲高性能轨迹控制，每轴最高脉冲频率 10MHZ。

RT2008 适合于多轴点位运动、插补运动、轨迹规划、手轮控制、编码器位置检测、IO 控制位置比较、位置锁存等功能的应用。运动函数库功能丰富，易学易用，用户开发应用软件十分方便。

RT2008 运动控制卡目前广泛应用于各行业自动化设备中。主要包括：

机器视觉及自动检测设备，如：影像测量仪，PCB 自动检测设备等；

激光加工设备，如：激光打标机，激光切割机等；

电子产品加工、装配设备，如：丝印机，贴片机，PCB 钻孔机等；

生物、医学自动采样、处理设备；

专用工业机器人等。

1.2. 型号说明

表 1-1 型号说明

8 轴高性能插补卡	运动控制卡	RT2008	主卡
	接线盒	RT2008EX	SCSI 接口配件
	转接线	SCSI68-2.0M-CN	
	转接组件	RT2008EX-M	黑色插座接口配件
	转接线	SCSI68-2.0M-CN	

1.3. 产品图片

1.3.1. RT2008 主卡

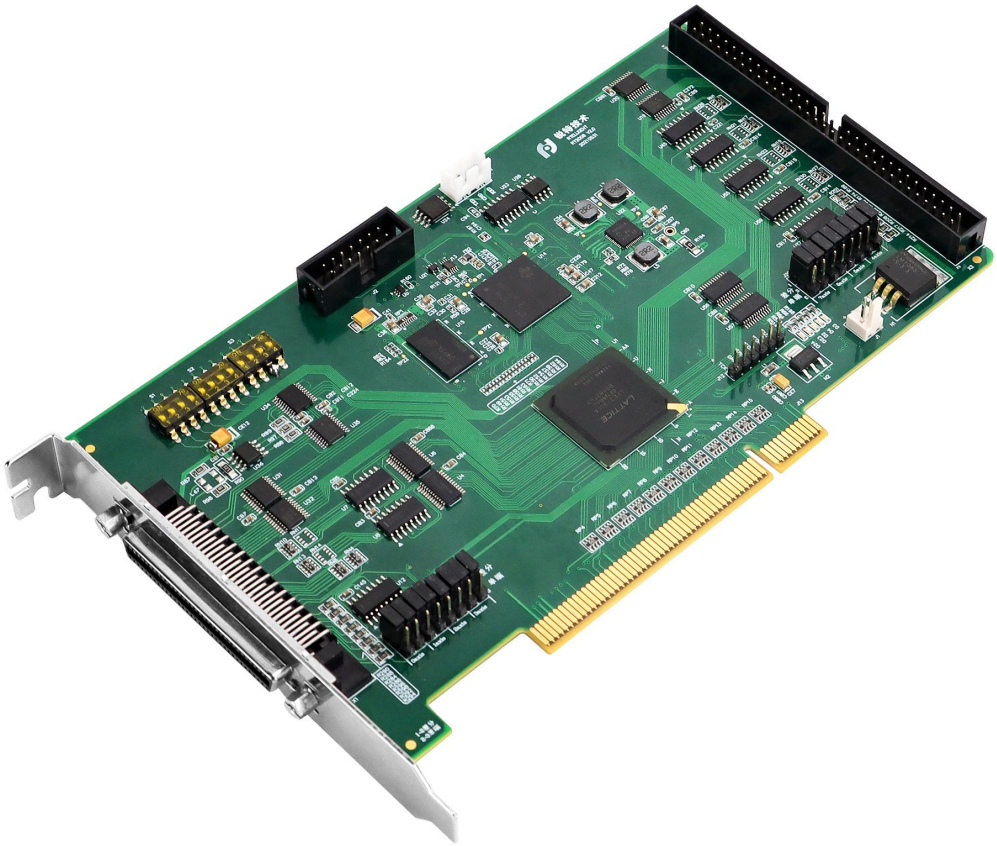


图 1-1 主卡外观

1.3.2. RT2008EX 接线盒

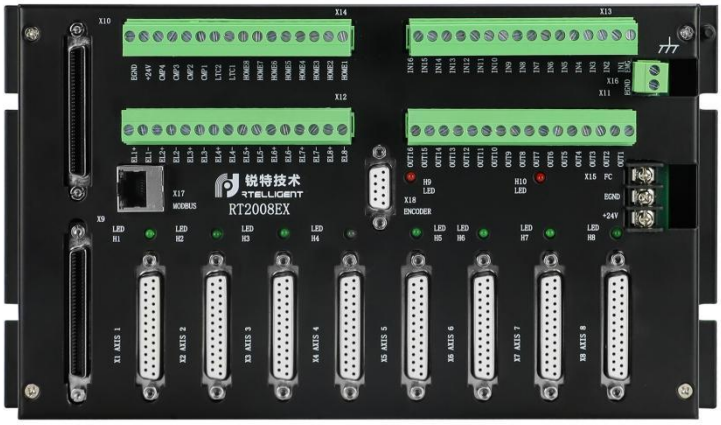


图 1-2 接线盒外观

1.3.3. 转接组件

RT2008EX-M　　SCSI68-2.0M-CN

图 1-3 配件

1.4. 功能说明

表 1-2 功能说明

名称	参数	说明
伺服控制周期	125us（不可调）	具备功能
控制周期	250us（不可调）	具备功能
脉冲量输出	8轴脉冲输出	具备功能
	4轴脉冲输出	可选功能
编码器输入	8路四倍频增量式 最高频率8MHz(四倍频后)	具备功能
	4路四倍频增量式 最高频率8MHz(四倍频后	可选功能
辅助编码器输入	一路四倍频增量式 最高频率8MHz(四倍频后)	具备功能
手轮信号输入	一路四倍频手轮输入 最高频率10MHz(四倍频后)	具备功能
限位信号输入	每轴正负极限光耦隔离	具备功能
原点信号输入	每轴一路光耦隔离	具备功能
驱动器报警信号输入	每轴一路光耦隔离	具备功能
驱动器使能信号输出	每轴一路光耦隔离	具备功能
驱动器复位信号输出	每轴一路光耦隔离	具备功能
驱动器位置信号输入	每轴一路光耦隔离	具备功能

通用数字信号输入	16路光耦隔离	具备功能
通用数字信号输出	16路光耦隔离	具备功能
位置比较输出	4路高速光耦隔离	具备功能
编码器位置锁存输入	两路高速光耦隔离	具备功能
点位运动	S-曲线、梯形曲线、Jog运动、电子齿轮运动	具备功能
同步运动	电子凸轮运动模式	具备功能
PT运动	位置时间运动模式	具备功能
PVT运动	位置、速度和时间运动模式	具备功能
插补运动	直线、圆弧、螺旋线等插补运动	具备功能
运动程序	在运动控置器上直接运行程序	具备功能
滤波器	PID+速度前馈+加速度前馈	具备功能
扩展模块	支持数字量扩展和模拟量扩展	具备功能
硬件捕获功能	编码器零位信号 原点信号 探针信号(请参考编程手册)	具备功能
安全措施	设置跟随误差极限 设置输出电压饱和极限	具备功能

第 2 章 硬件接口及电路

2.1. 主卡接口说明

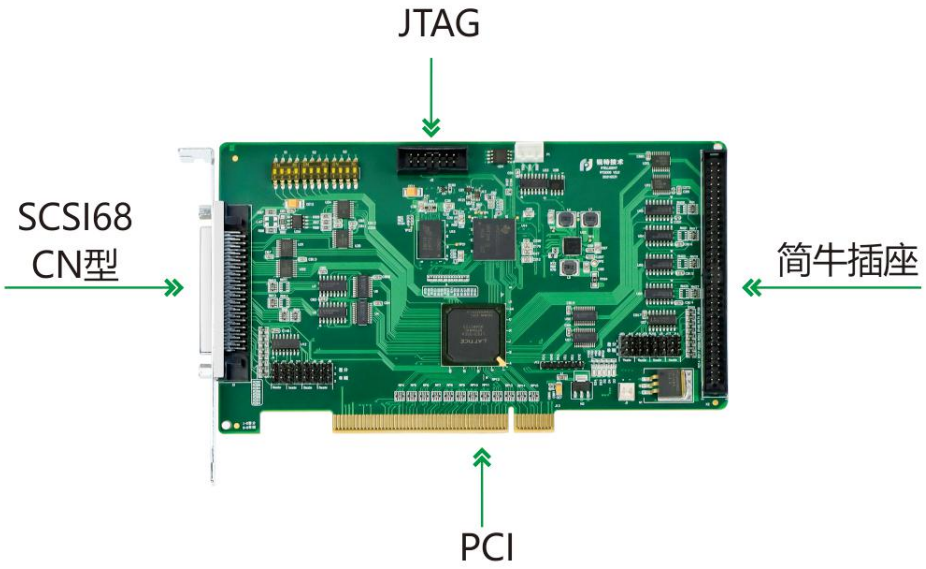


图 2-1 主卡接口示意图

2.2. 接线盒接口定义

2.2.1. 轴信号的接口

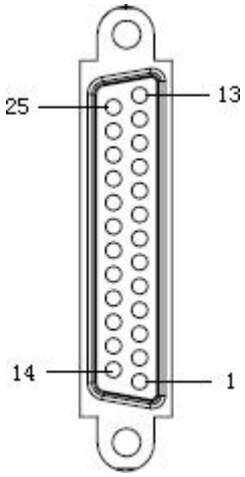


图 2-2 轴信号接口（X1-X8）引脚示意图

接线盒 X1~X8 接口是轴信号接口。其 25pin 引脚定义及接线电路如下：

表 2-1 轴信号接口（X1-X8）引脚定义

引脚	信号	说明	引脚	信号	说明
1	OGND	外部电源地	14	OVCC	+24V输出
2	ALM	驱动报警	15	RESET	驱动报警复位
3	ENABLE	驱动允许	16	INP	电机到位
4	A-	编码器输入	17	A+	编码器输入
5	B-	编码器输入	18	B+	编码器输入
6	Z-	编码器输入	19	Z+	编码器输入
7	+5V	电源输出	20	GND	数字地
8	NC		21	GND	数字地
9	DIR+	步进方向输出	22	DIR-	步进方向输出
10	GND	数字地	23	PULSE+	步进脉冲输出
11	PULSE-	步进脉冲输出	24	GND	数字地
12	备用	备用	25	NC	
13	GND	数字地			

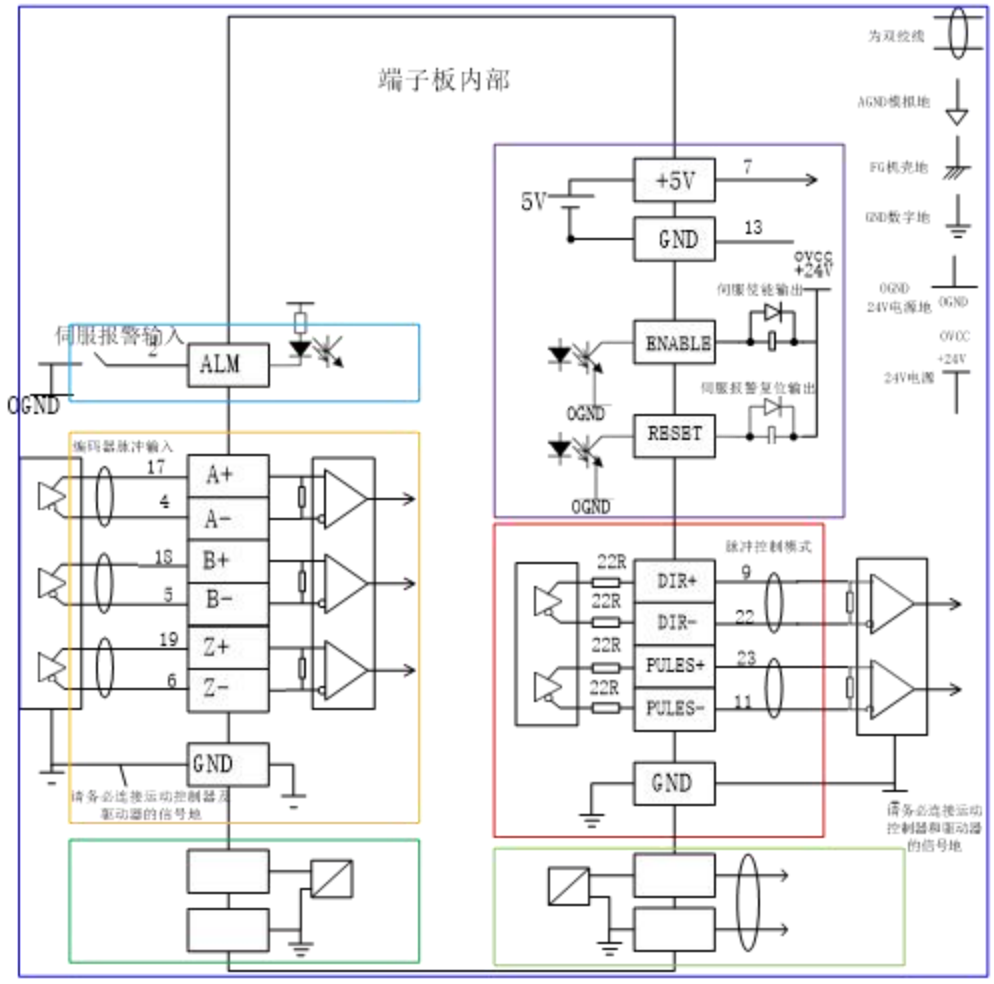


图 2-3 轴信号接口（X1-X8）接线电路

2.2.2. 通用数字输入输出信号、原点信号和限位信号接口

接线盒 X11、X12、X13 和 X14 接口是通用数字输入输出信号(简称通用 IO)、原点输入信号（简称 HOME）、限位输入信号接口（简称 EL）、位置锁存信号（简称 LTC）、位置比较输出信号（简称 CMP）。四个连接端子支持整体拆卸，在更换接线盒时，可以整体拆除后接入新的接线盒。

注意	在安装连接端子时，务必保证端子两头压实，否则高低不平的端子可造成引脚接触不良而引发信号不稳定。

表 2-2 通用数字输出接口（X11）引脚定义

引脚	信号	说明	引脚	信号	说明
1	OUT1	通用输出	9	OUT9	通用输出
2	OUT2	通用输出	10	OUT10	通用输出
3	OUT3	通用输出	11	OUT11	通用输出
4	OUT4	通用输出	12	OUT12	通用输出
5	OUT5	通用输出	13	OUT13	通用输出
6	OUT6	通用输出	14	OUT14	通用输出
7	OUT7	通用输出	15	OUT15	通用输出
8	OUT8	通用输出	16	OUT16	通用输出

表 2-3 限位输入信号接口（X12）引脚定义

引脚	信号	说明	引脚	信号	说明
1	EL1+	1轴正向限位	9	EL5+	5轴正向限位
2	EL1-	1轴负向限位	10	EL5-	5轴负向限位
3	EL2+	2轴正向限位	11	EL6+	6轴正向限位
4	EL2-	2轴负向限位	12	EL6-	6轴负向限位
5	EL3+	3轴正向限位	13	EL7+	7轴正向限位
6	EL3-	3轴负向限位	14	EL7-	7轴负向限位
7	EL4+	4轴正向限位	15	EL8+	8轴正向限位
8	EL4-	4轴负向限位	16	EL8-	8轴负向限位

表 2-4 通用数字输入接口（X13）引脚定义

引脚	信号	说明	引脚	信号	说明
1	IN1	通用输入	9	IN9	通用输入
2	IN2	通用输入	10	IN10	通用输入
3	IN3	通用输入	11	IN11	通用输入
4	IN4	通用输入	12	IN12	通用输入
5	IN5	通用输入	13	IN13	通用输入
6	IN6	通用输入	14	IN14	通用输入
7	IN7	通用输入	15	IN15	通用输入
8	IN8	通用输入	16	IN16	通用输入

表 2-5 原点输入信号接口（X14）引脚定义

引脚	信号	说明	引脚	信号	说明
1	HOME1	1轴原点输入	9	LTC1	位置1锁存输入
2	HOME2	2轴原点输入	10	LTC2	位置2锁存输入
3	HOME3	3轴原点输入	11	CMP1	比较输出1
4	HOME4	4轴原点输入	12	CMP2	比较输出2
5	HOME5	5轴原点输入	13	CMP3	比较输出3
6	HOME6	6轴原点输入	14	CMP4	比较输出4
7	HOME7	7轴原点输入	15	+24V	+24V供电输出
8	HOME8	8轴原点输入	16	EGND	+24V电源地

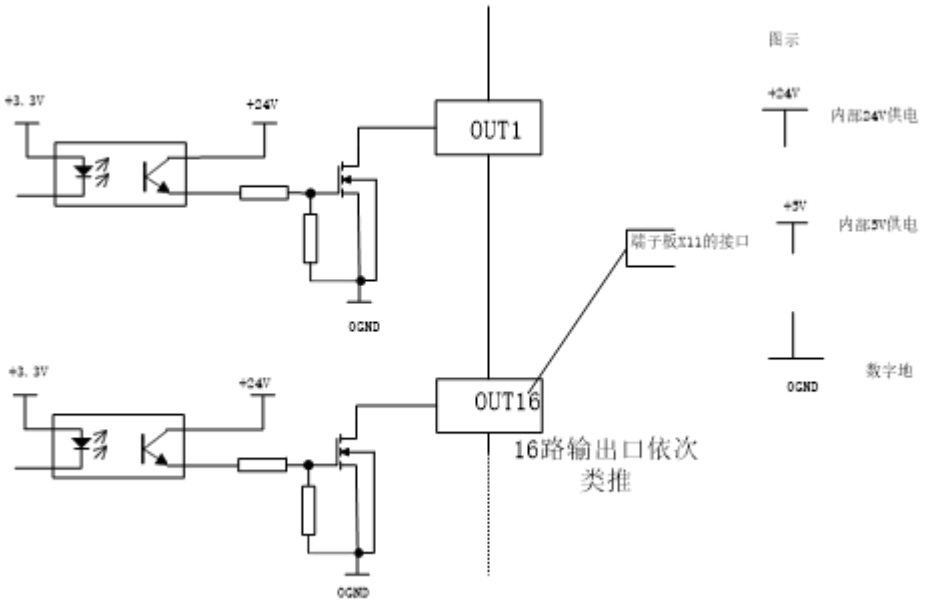


图 2-4 接线盒 X11 接口内部示意图

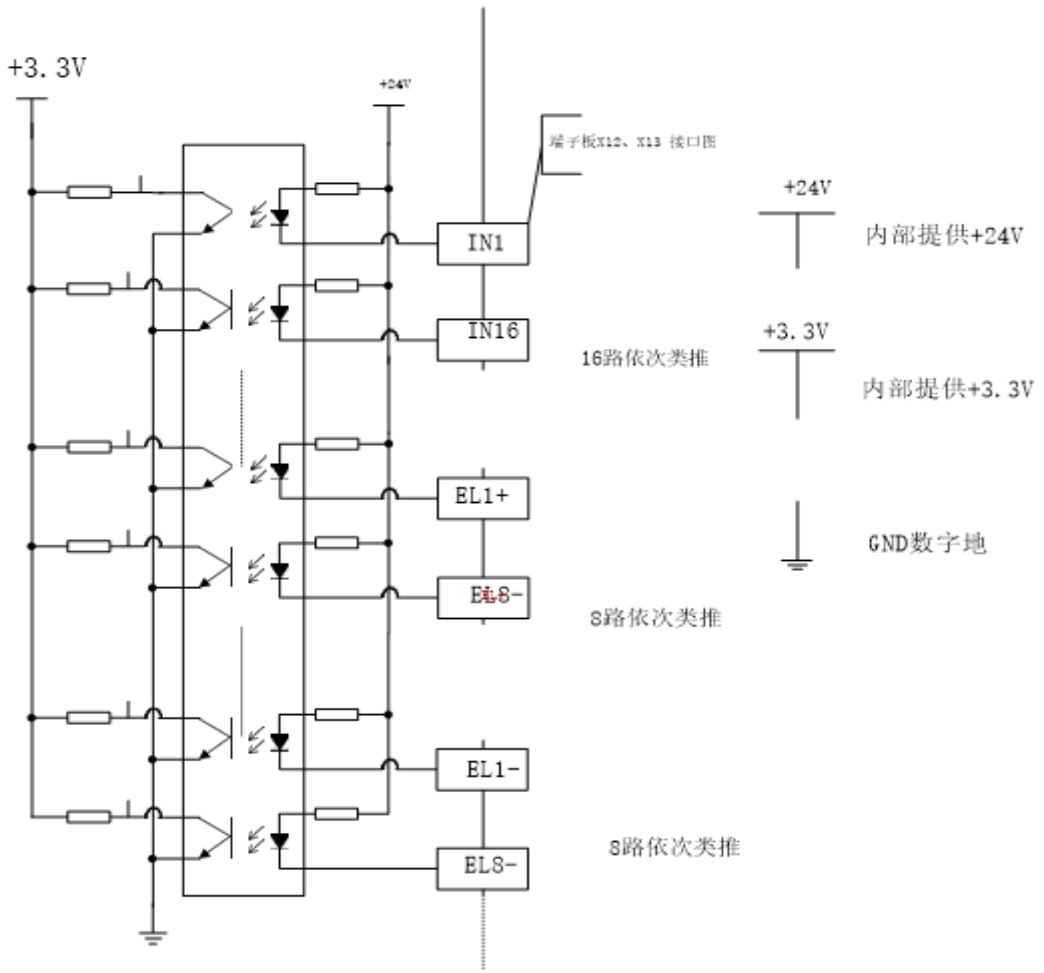


图 2-5 接线盒 X12,X13 接口内部示意图

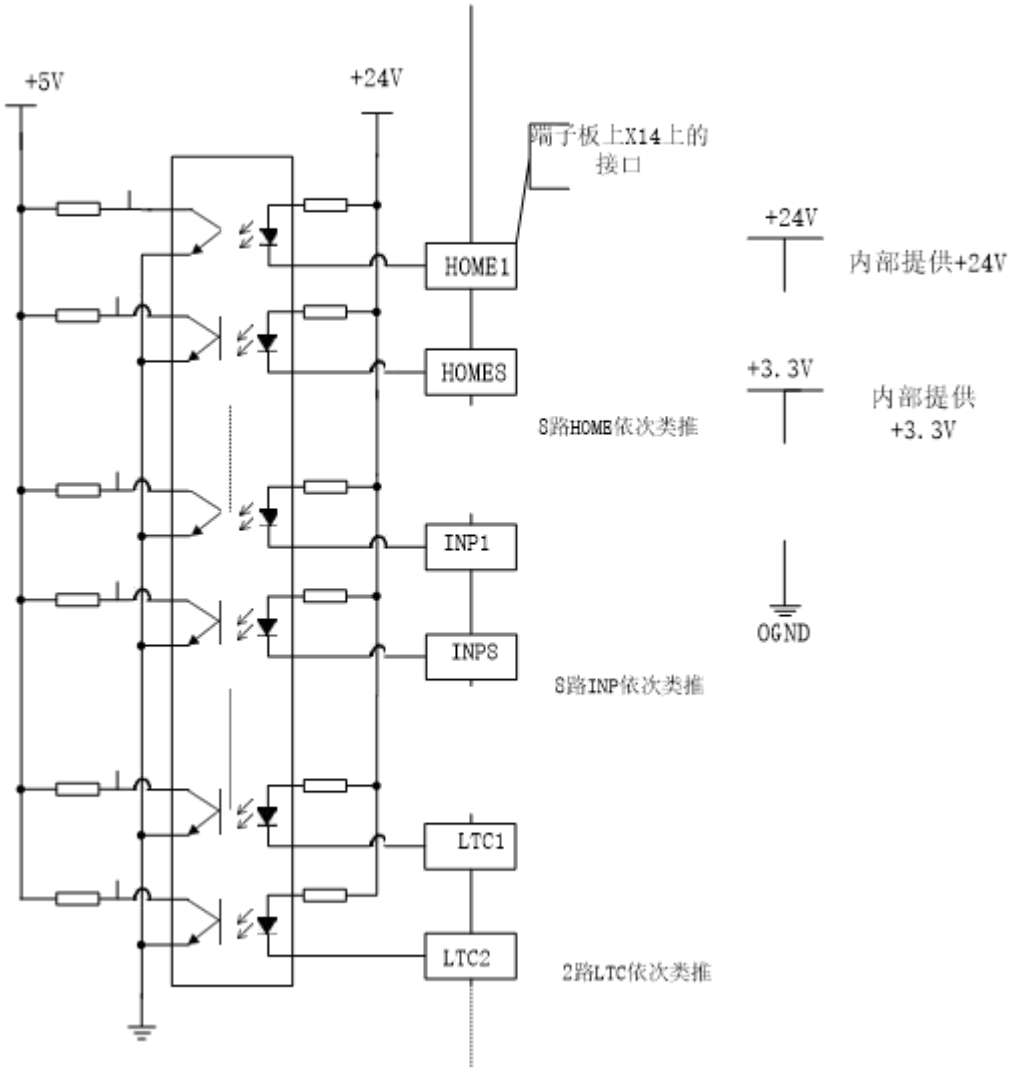


图 2-6 接线盒 X14 接口内部示意图

提示	当通用IO的输出接感性负载时，应考虑感性负载对IO的影响，尽量确保感性负载能量的泄放不经过通用数字输出。

2.2.3. 急停信号接口

接线盒 X16 接口是急停数字信号，连接端子支持整体拆卸，在更换接线盒时，可以整体拆除后接入新接线盒。

表 2-6 急停数字信号接口（X16）引脚定义

引脚	信号	说明
1	EMG	急停输入信号
2	EGND	数字地

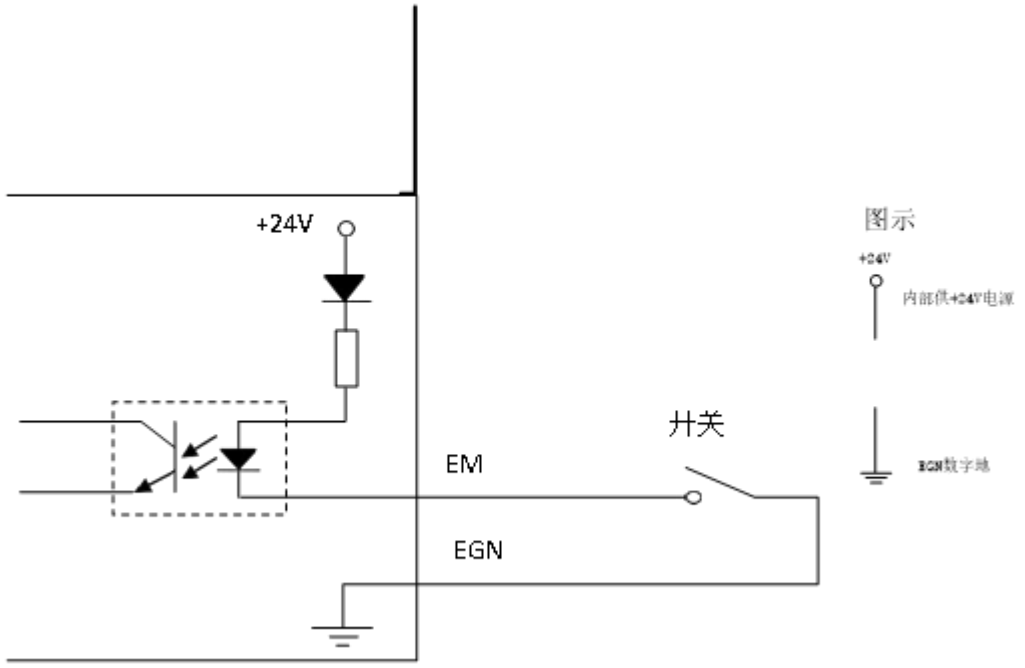


图 2-7 接线盒 X16 接口内部示意图

2.2.4. 辅助编码器接口

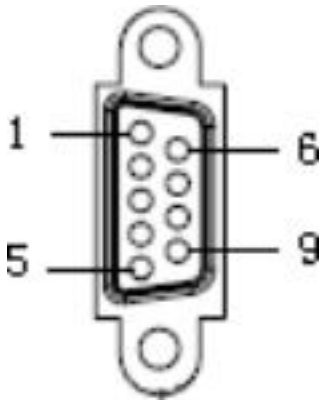


图 2-8 辅助编码器接口（X18）引脚示意图

接线盒 X18 接口是辅助编码器接口。辅助编码器接口接受 A 相、B 相和 Z 相

信号，（辅编不能用于捕获功能）。其 9pin 引脚定义如下：

表 2-7 辅助编码器接口（X18）引脚定义

引脚	信号	说明	引脚	信号	说明
1	A+	编码器输入	6	A-	编码器输入
2	B+	编码器输入	7	B-	编码器输入
3	Z+	编码器输入	8	Z-	编码器输入
4	备用	备用	9	GND	数字地
5	+5V	电源输出			

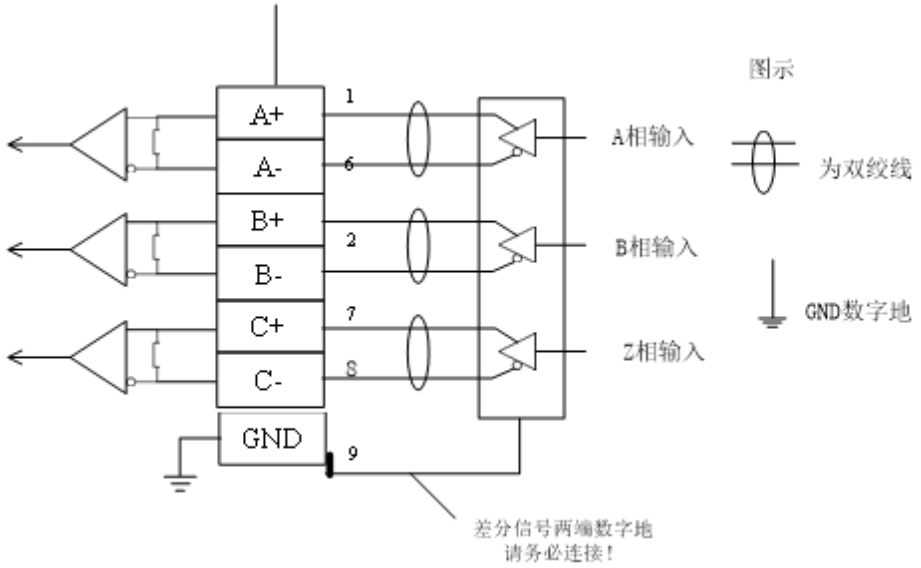


图 2-9 辅助编码器接口（X18）内部示意图

注意

X18提供的是差分接口，所以推荐用户以差分方式接线，且差分信号两端数字地务必连通。如果用户确实需要以单端方式来接线，请联系锐特公司。

第 3 章 快速启动指南

3.1. 硬件安装步骤

3.1.1. RT2008 硬件设置

如图 3-1 所示，RT2008 运动控制卡上有两组跳线开关和三组拨码开关 S1、S2 和 S3，用于设置 RT2008 卡的工作方式和参数。

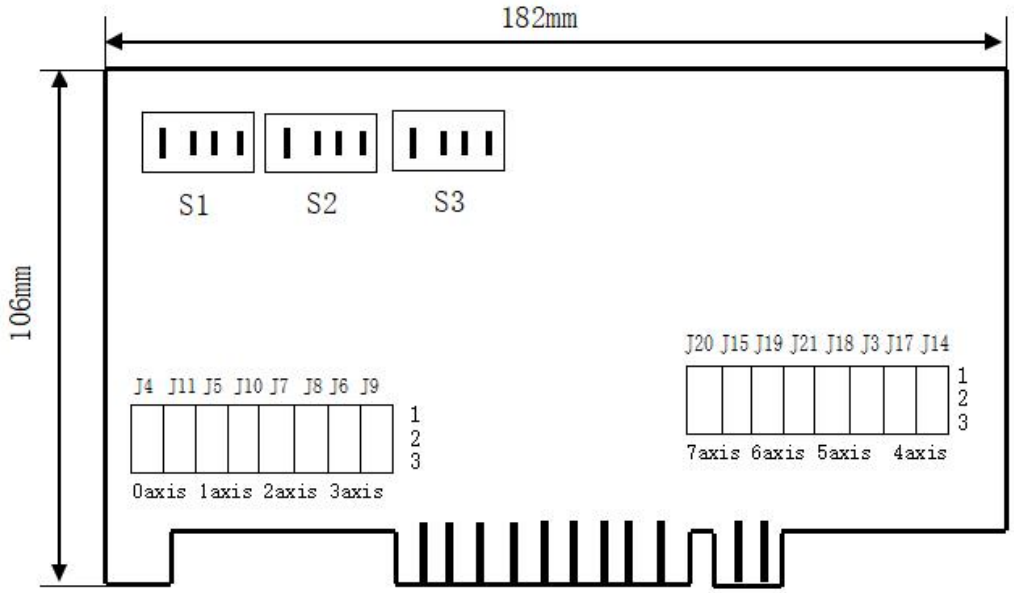


图 3-1 开关示意图

3.1.2. 跳线开关的设置

RT2008 上的跳线开关用于设置电机指令脉冲信号输出方式，可设为差分输出或单端输出方式。设置方式如图 3-2（左）和图 3-2（右）所示；各轴脉冲和方向信号及跳线开关的对应关系如表 3-1 所示。

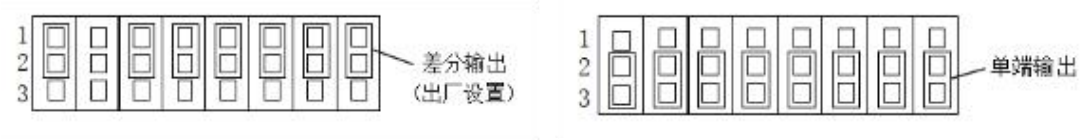


图 3-2 差分输出（左）与单端输出（右）跳线设置

表 3-1 各轴脉冲和方向信号及跳线开关的对应关系表

轴号	信号	跳线开关		轴号	信号	跳线开关	
第0轴	PUL0+	J4	上	第1轴	PUL1+	J5	上
	+5V		下		+5V		下
	DIR0+	J11	上		DIR1+	J10	上
	+5V		下		+5V		下
第2轴	PUL2+	J7	上	第3轴	PUL3+	J6	上
	+5V		下		+5V		下
	DIR2+	J8	上		DIR3+	J9	上
	+5V		下		+5V		下
第4轴	PUL4+	J17	上	第5轴	PUL5+	J3	上
	+5V		下		+5V		下
	DIR4+	J14	上		DIR5+	J18	上
	+5V		下		+5V		下
第6轴	PUL6+	J21	上	第7轴	PUL7+	J15	上
	+5V		下		+5V		下
	DIR6+	J19	上		DIR7+	J20	上
	+5V		下		+5V		下

3.1.3. 拨码开关设置

如图 3-3 所示，RT2008 卡上有三个 4 位拨码开关 S1、S2 和 S3；S1 用于卡号设置，S2 和 S3 为备用。具体设置方法如表 3-2 所示。

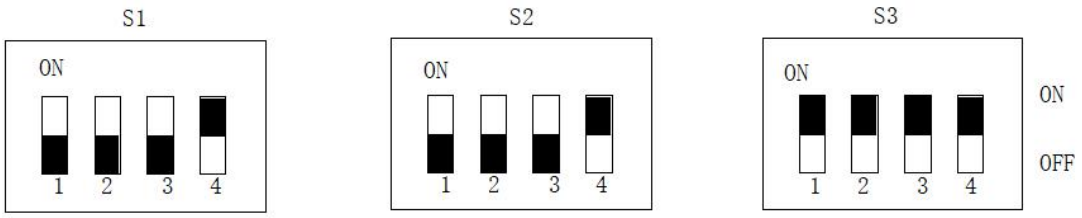


图 3-3 拨码开关示意图

表 3-2 拨码开关 S1 设置运动控制卡号

控制卡号	S1-1	S1-2	S1-3	S1-4
0	OFF	OFF	OFF	保留
1	ON	OFF	OFF	保留
2	OFF	ON	OFF	保留
3	ON	ON	OFF	保留
4	OFF	OFF	ON	保留
5	ON	OFF	ON	保留
6	OFF	ON	ON	保留
7	ON	ON	ON	保留

拨码开关S1-1、S1-2、S1-3出厂默认配置为OFF，即为0号卡。

当每张卡都设置为0号卡时，按靠近CPU的顺序自动排序。除此种情况外，如有两张卡以上设置为相同卡号，则初始化函数Rt200x_board_init会返回一个错误代码。

3.2. 硬件安装

RT2008 运动控制卡硬件遵从 32bit PCI 卡结构标准，其安装方法与其它 PCI 卡，如：声卡、网卡等相似。具体步骤如下：

1）打开控制卡的包装，参考相应运动控制卡硬件设置的说明，按照实际需求，完成跳线开关、拨码开关的设置；

2）操作员要带好防静电手套，并触摸一下地线，完全释放身上的静电；

3）关闭 PC 机以及一切与 PC 相连的设备；

4）打开 PC 机的机箱；

5）选择一个靠近处理器的 32bit PCI 插槽，将控制卡垂直插入插槽中；

6）将控制卡用螺钉固定在 PC 机机箱上，确保紧固可靠；

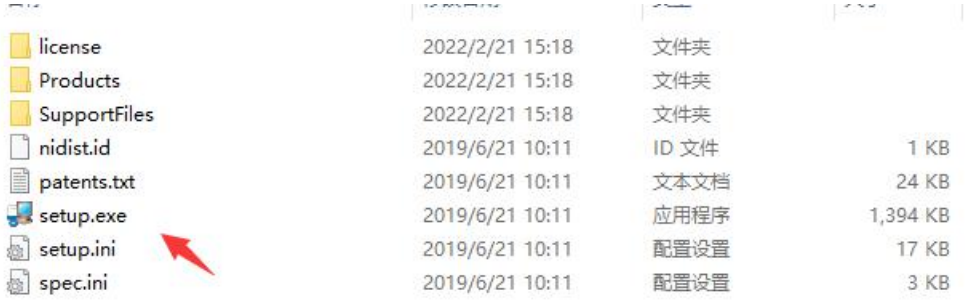
7）将接线板用电缆线与控制卡对应的插座连接，并确保连接牢固可靠。

3.3. 在 Windows 环境下，安装控制卡的驱动程序

3.3.1. 驱动运行环境安装

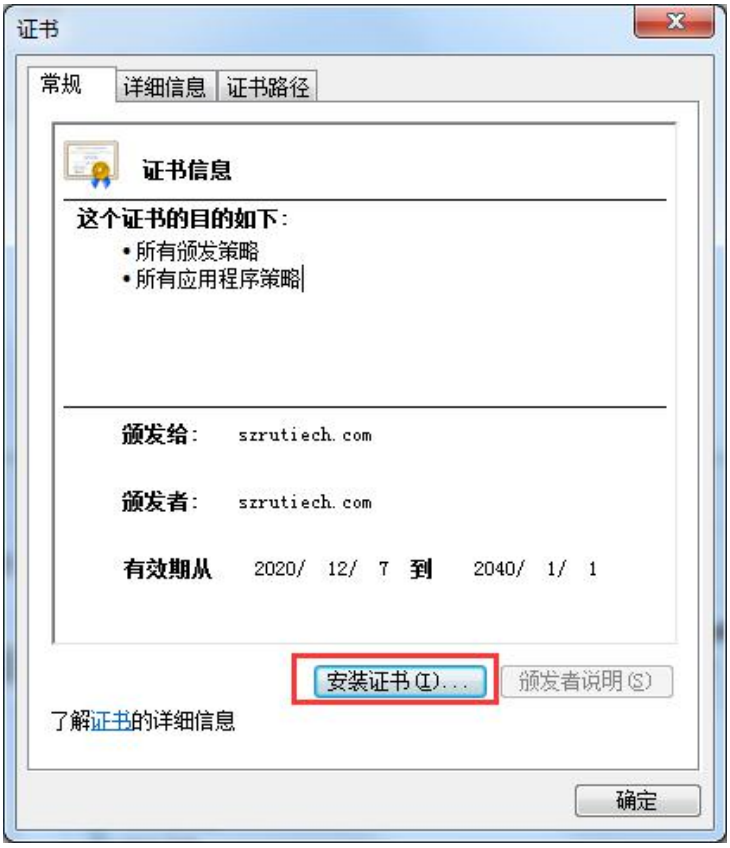
1　如果电脑未曾安装过锐特的运动控制卡驱动，则需要先安装驱动运行的环境。解压文件夹下的 NIVISA_runtime 压缩包，打开该文件夹。

2　点击 SetUp 安装驱动需要的环境，然后再重启电脑。



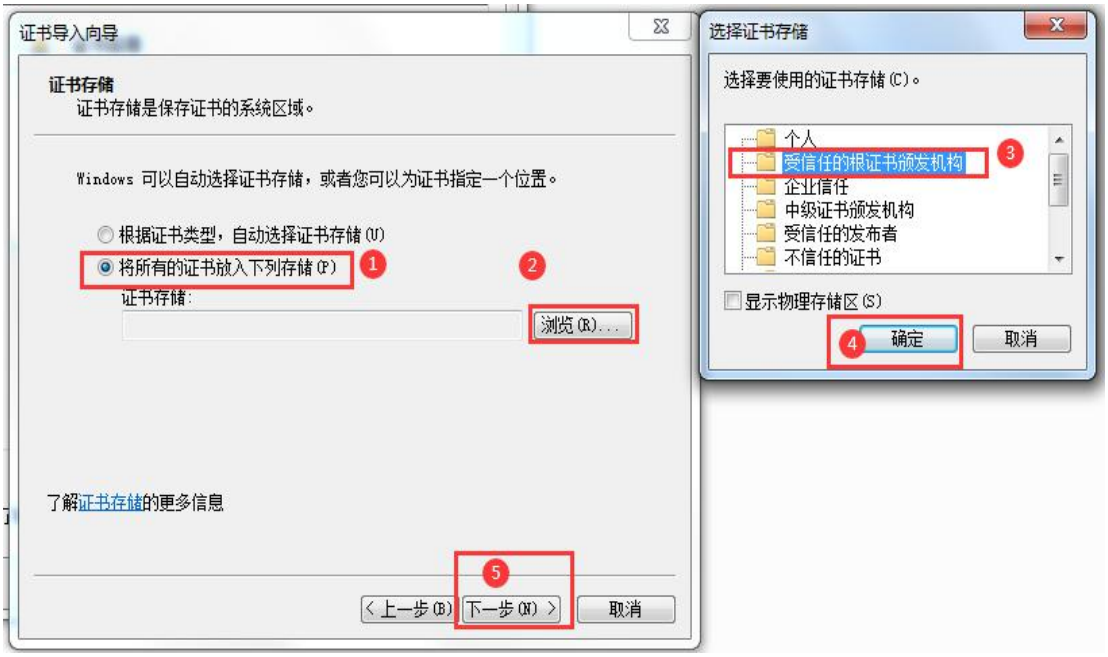
3.3.2. 安装证书

1　双击“RT2008.cer”文件，如图，点击“安装证书”。

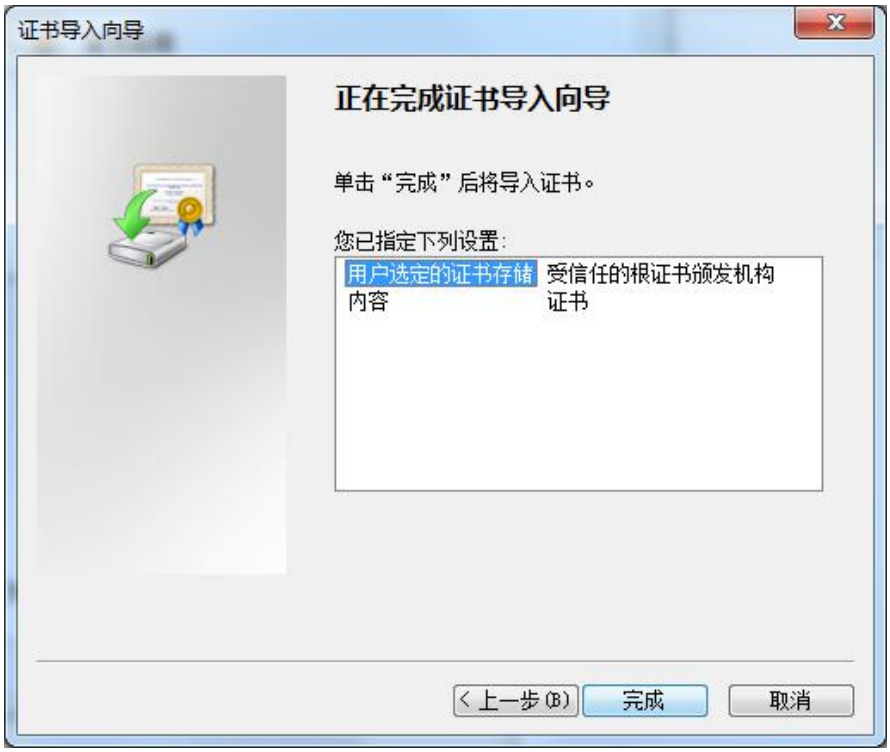


2 在弹出的对话框选择“下一步”。

3 将证书安装到受信任的区域。

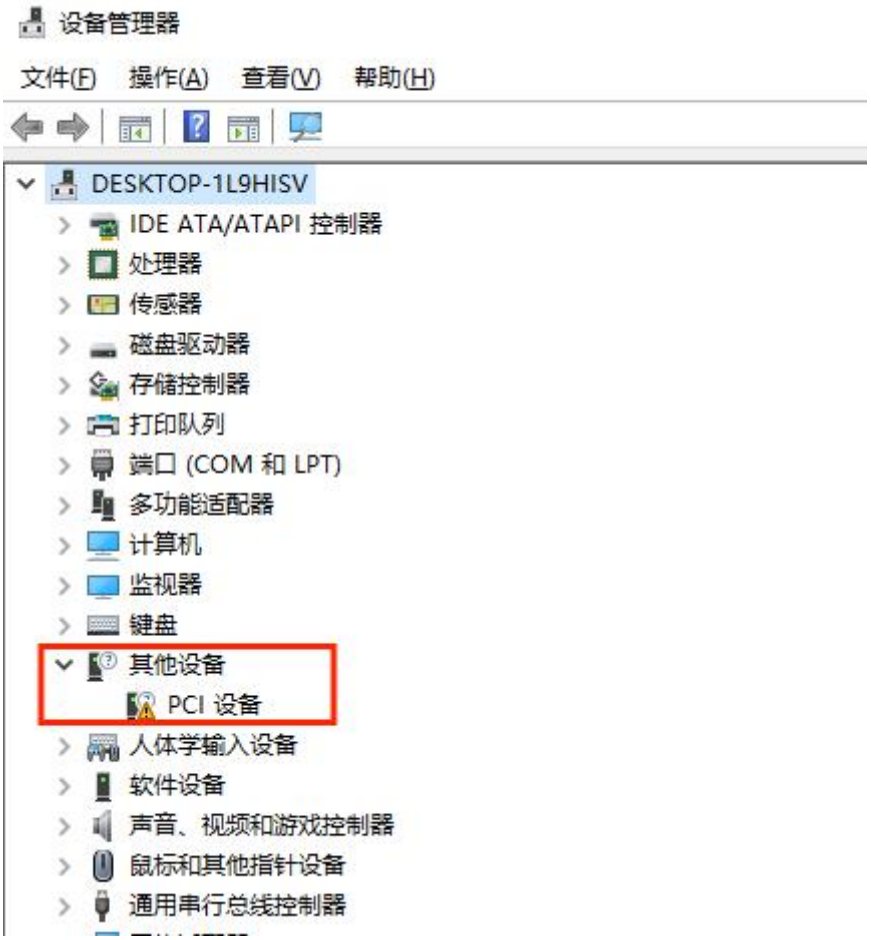


4 最后点击“完成”，如图所示即安装完成。



3.3.3. 驱动安装

1　鼠标右键单击“计算机”—选择“管理(G)”—选择“设备管理器”，在“设备管理器”中是否找得到带有黄色感叹号的“PCI 设备”选项。



2　在驱动文件夹下 rt2008.inf 文件上面，右键“安装”，按照提示即可以完成安装。

3.4. 应用软件开发方法

RT2008 运动控制卡的应用软件可以在 Visual Basic、Visual C++和 C#等高级语言环境下开发。

如果您对 VB、VC、C#语言都不熟悉，建议您花时间阅读一本编程语言的培训教材，并且通过练习掌握该语言的基本技巧，如：编写简单的程序、创建窗体和调用函数。

如果您曾用 VB、VC 或 C#等程序语言开发过运动控制软件，并具有丰富的经验，则可直接阅读第 5 章“函数库详解”。

3.4.1. 基于 WINDOWS 平台的应用软件结构

使用锐特运动控制卡的自动化设备运动控制系统构架如下图所示：

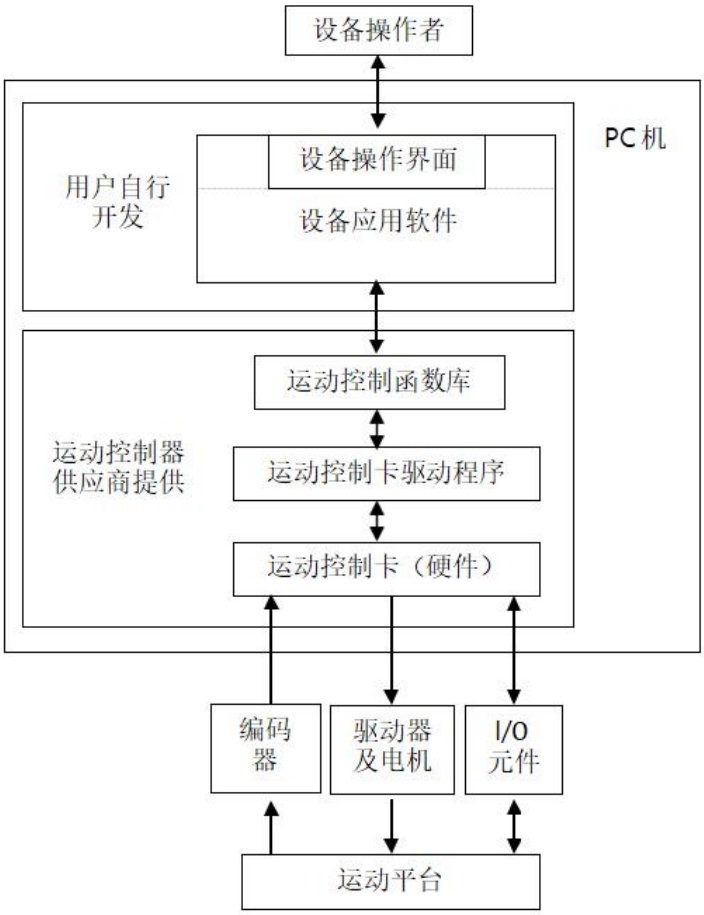


图 3-4 基于锐特运动控制卡的自动化设备运动控制系统构架

从上图中可看出，运动控制系统的工作原理可以简单描述为：

1）操作员通过操作界面（包括显示屏和键盘）将指令信息传递给设备应用软件；

2）设备应用软件将操作者的信息以及应用软件中已有的运动流程、运动轨迹等数据转化为运动参数，并根据这些参数调用 DLL 库中运动函数；

3）运动函数通过锐特运动控制卡驱动程序向运动控制卡发出控制指令；

4）运动控制卡根据控制指令发出相应的指令脉冲给驱动器及电机、读写通用输入输出口、读取编码器数据。

用户根据设备的工艺流程、运动轨迹和友好的人机界面等要求开发设备应用软件。锐特公司已提供支持 RT2008 运动控制卡的硬件驱动程序和 DLL 运动函数库，包括控制卡初始化、单轴及多轴控制、数字量输入/输出控制等多种函数。这些函数可以方便地完成与运动控制相关的功能，用户不需要更多了解硬件电路的细节以及运动控制和插补算法的细节，就能使用 VB、VC 等程序语言开发出自己的运动控制系统应用软件。

用户编写的设备应用软件的典型流程如图 3-5 所示。

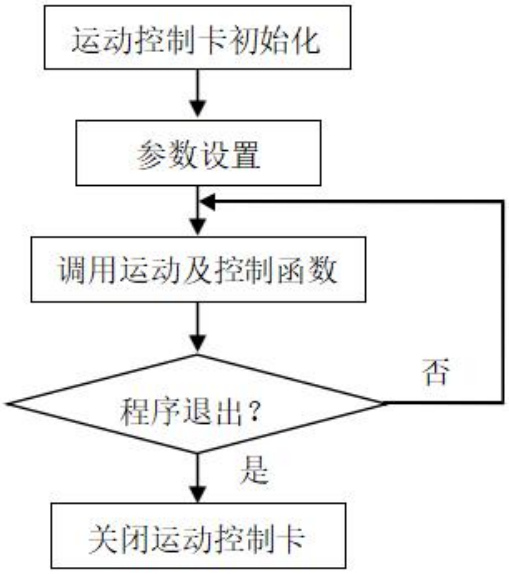


图 3-5 设备应用软件的典型流程

用户在开发应用软件（即机器控制软件）的过程中所需要做的就是针对上面所说的第 1 步和第 2 步进行编程。锐特控制公司已提供支持 RT2008 运动控制卡的硬件驱动程序和 DLL 运动函数库。这些函数提供了所有与运动控制相关的功能，使用极为方便。用户不需要更多了解硬件电路的细节以及运动和插补的计算细节，就能够使用 C、C++、Visual Basic、C#等程序语言调用这些函数来快速开发出自己的应用软件。

3.5. 实现基本功能的方法及相关函数

本部分介绍采用 C#语言通过调用相关函数实现 RT2008 运动控制卡基本功能的方法。

在编程之前，一定要用Motion软件检测硬件系统，确保硬件接线正确。

3.5.1. 限位开关及急停开关的设置

在设备进行运动功能调试之前，必须确保安全防护机制的有效运作。

限位开关在运动平台出现超出行程的运动时，起到限制作用，使电机减速或紧急停止，提高设备运行时的安全性能。在使用运动控制卡进行运动控制之前，必须保证限位开关的有效性。

急停开关在运动过程中出现意外的运动时，能起到紧急停止运动的功能，提高设备运行时的安全性能。在使用运动控制卡进行运动控制之前，必须保证急停开关的有效性。

3.5.1.1 限位开关的设置

RT2008 卡提供了限位开关设置函数 Rt200x_set_el_mode 来设置限位功能。用户必须根据设备的限位开关硬件接线，来设置限位开关工作的有效电平。

相关函数如下表所示:

名称	功能	参考
Rt200x_set_el_mode	设置限位开关信号	5.4限位开关设置函数
Rt200x_get_el_mode	读取限位开关信号设置	

假设设备正常运动时限位开关为高电平，运动平台碰到限位开关时为低电平，则此时应设置控制卡限位开关的有效电平为低电平。

例程：设置限位开关为低电平

```
……
Int MyCardNo,Myaxis,Myel_enable,Myel_logic,Myel_mode;
MyCardNo = 0;    ’卡号
Myaxis = 0; ’轴号
Myel_enable = 1;’正负限位使能
Myel_logic = 0; ’正负限位低电平有效
```

```
Myel_mode = 0;  ’正负限位停止方式为立即停止
Rt200x_set_el_mode (MyCardNo,Myaxis,Myel_enable,Myel_logic,Myel_mode);  设置 0 号轴限位信号
……
```

3.5.1.2 急停开关的设置

RT2008卡提供了急停开关设置函数Rt200x_set_emg_mode来设置限位功能。用户必须根据设备的急停开关硬件接线，来设置急停开关工作的有效电平。

相关函数如下表所示：

名称	功能	参考
Rt200x_set_emg_mode	设置急停开关信号	5.18异常信号接口函数
Rt200x_get_emg_mode	读取急停开关信号设置	

RT2008卡有专用于EMG急停开关的硬件接口，用户只需要根据自己的需求对应接口电路进行接线，然后调用急停开关设置函数进行设置即可。

假设设备正常运动时急停开关为高电平，当急停开关为低电平时紧急停止运动，则此时应设置控制卡急停开关的有效电平为低电平。

例程：设置急停开关为低电平

```
……
Int
MyCardNo,Myaxis,Mye
nable,Mylogic;
MyCardNo = 0;          ’卡号
Myaxis = 0;            ’轴号，保留参数，固定值为 0
Myenable = 1;          ’急停信号使能
Mylogic = 0;           ’急停信号低电平有效

 Rt200x_set_emg_mode (MyCardNo,Myaxis,Myenable,Mylogic); ’设置所有轴急停信号

 ……
```

3.5.2. 回原点运动的实现

3.5.2.1 回原点步骤

在进行精确的运动控制之前，需要设定运动坐标系的原点。运动平台上都设有原点传感器（也称为原点开关）。寻找原点开关的位置并将该位置设为平台的坐标原点的过程即为回原点运动。默认回原点时遇限位不会自动反找原点，如需

自动反找需初始化卡时调用一次 Rt200x_set_home_el_return 函数，RT2008 控制卡共提供了 10 种回原点方式。

1、采用回零方式 0~5 或 7~9 作回原点运动，主要步骤如下：

1）使用 Rt200x_set_home_pin_logic 函数设置原点开关的有效电平；

2）使用 Rt200x_set_ez_mode 函数设置 EZ 信号的有效电平；

3）使用 Rt200x_set_homemode 函数设置回原点方式；

4）设置回原点运动的速度曲线；

5）使用 Rt200x_home_move 函数进行回原点运动；

6）回到原点后，指令脉冲计数器清零。

2、采用原点捕获方式（方式 6）作回原点运动，主要步骤如下：

1）使用 Rt200x_set_homelatch_mode 函数设置原点信号锁存方式。

2）使用 Rt200x_set_homemode 函数设置回原点方式；

3）设置回原点运动的曲线速度形式；

4）使用 Rt200x_home_move 函数进行回原点运动；

5）回到原点后，指令脉冲计数器清零。

3.5.2.2 回原点方式

RT2008 系列控制卡提供了 10 种回原点运动的方式：

方式 0：一次回零

该方式以设定速度回原点；适合于行程短、安全性要求高的场合。动作过程为：电机从初始位置以恒定速度向原点方向运动，当到达原点开关位置，原点信号被触发，电机立即停止（过程 0）；将停止位置设为原点位置，如图 3-6 所示。

图 3-6 一次回零方式示意图

方式 1：一次回零加回找

该方式先进行方式 0 运动，完成后再反向回找原点开关的边缘位置，当原点信号第一次无效的时候，电机立即停止；将停止位置设为原点位置如图 3-7 所示。

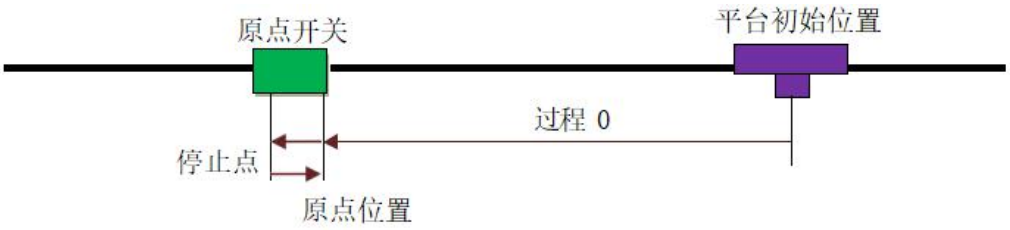


图 3-7 一次回零加回找方式示意图

方式 2：二次回零

如下图所示，该方式为方式 0 和方式 1 的组合。先进行方式 1 的回零加反找，完成后再进行方式 0 的一次回零。可参见方式 0 和方式 1 的说明。

图 3-8 二次回零方式示意图

方式 3：一次回零后再找 1 个 EZ 信号后回零

该方式在回原点运动过程中，当找到原点信号后，还要等待该轴的 EZ 信号出现，此时电机停止。回原点过程如图 3-9 所示。

图 3-9 一次回零后再找 1 个 EZ 信号回零方式示意图

方式 4：记 1 个 EZ 信号回零

该方式在回原点运动过程中，当检测到该轴的 EZ 信号出现一次后，此时电

机停止。回原点过程如图 3-10 所示。

图 3-10 记 1 个 EZ 信号回零方式示意图

方式 5：一次回零再反找 EZ 信号

该方式在回原点运动过程中，当找到原点信号后，减速停止，然后以反找速度反向找到 EZ 生效此时电机停止。回原点过程如图 3-11 所示。

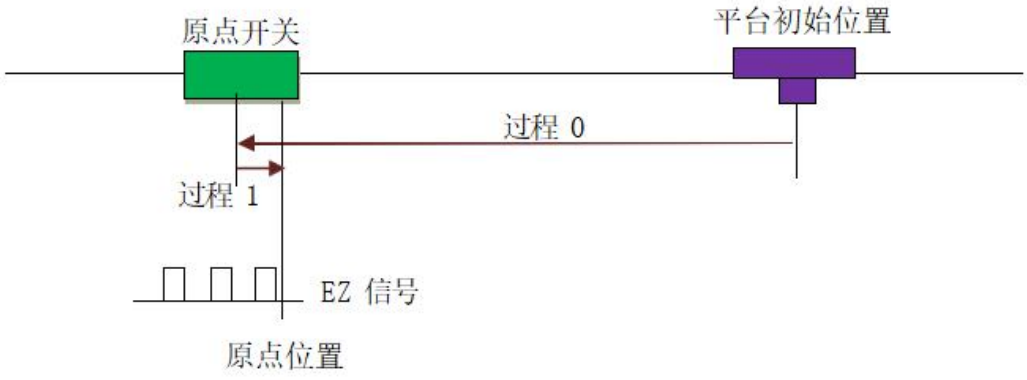


图 3-11 一次回零反找一个 EZ 进行回零方式示意图

方式 6：原点锁存

如图 3-12 所示，电机先以设定速度回原点，当原点开关边沿触发时，将当前位置锁存下来，同时电机减速停止。电机减速停止完成后再反向回找锁存位置，运动到锁存位置，电机停止。

图 3-12 原点锁存回零方式示意图

方式 7：原点锁存加同向 EZ 锁存

该方式先以方式 6 执行一次原点锁存回零，完成后继续沿设定回零方向运行到 EZ 信号产生，EZ 信号产生时锁存当前位置并执行减速停，电机减速停止之后再反向回找 EZ 的锁存位置，运动到锁存位置，电机停止。回原点过程如图 3-13 所示。

图 3-13 原点锁存加同向 EZ 锁存回零方式示意图

方式 8：单独记一个 EZ 锁存

在回零过程中检测到 EZ 有效边沿出现，锁存当前位置，执行减速停，电机减速停止之后再反向回找 EZ 的锁存位置，运动到锁存位置，电机停止。回原点过程如图 3-14 所示。

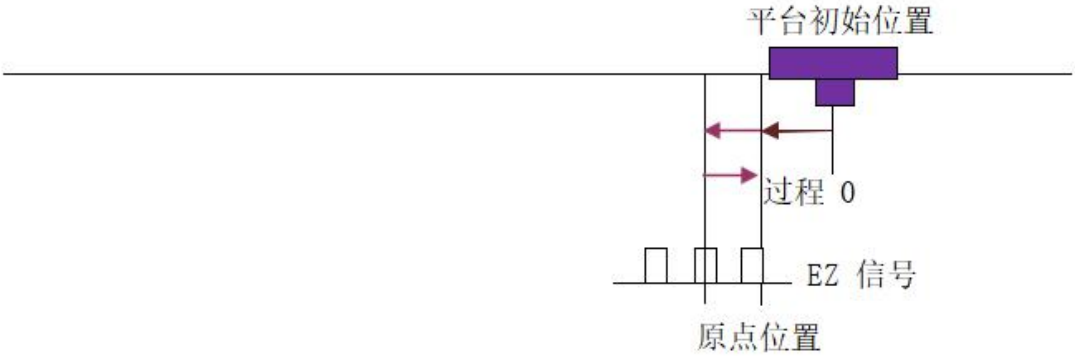


图 3-14 单独记一个 EZ 锁存回零方式示意图

方式 9：原点锁存加反向 EZ 锁存

该方式先以方式 6 执行一次原点锁存回零，完成后以与设定回零方向相反的方向运行到 EZ 信号产生，EZ 信号产生时锁存当前位置并执行减速停，电机减速停止之后再反向回找 EZ 的锁存位置，运动到锁存位置，电机停止。回原点过程如图 3-15 所示。

图 3-15 原点锁存加反向 EZ 锁存进行回零方式示意图

相关函数如下表所示：

名称	功能	参考
Rt200x_set_home_pin_logic	设置原点信号的有效电平	5.3回原点运动函数
Rt200x_set_homemode	选择回原点模式	
Rt200x_home_move	按指定的方向和速度方式开始回原点	5.5位置计数器控制函数
Rt200x_set_position	指令脉冲计数器清零	

注意

执行完Rt200x_home_move函数后，指令脉冲计数器不会自动清零；如需清零可以在回零运动完成后，调用Rt200x_set_position函数软件清零。

例程：方式0低速回原点

```
……
Int MyCardNo,Myaxis,Myorg_logic,Myhome_dir,Mymode,MyEZ_count;
Double Myfilter,Myvel_mode;
MyCardNo = 0;          ’卡号
Myaxis = 0;            ’轴号
Myorg_logic = 0;       ’原点信号低电平有效
Myfilter = 0;          ’保留参数
Rt200x_set_home_pin_logic (MyCardNo,Myaxis,Myorg_logic, Myfilter);  ’设置0轴原点信号
Myhome_dir = 0;         ’负方向回零
Myvel_mode = 0;         ’回零速度模式为低速
Mymode = 0;             ’回零模式为方式0，一次回零
MyEZ_count = 0;         ’保留参数
```

```
    Rt200x_set_homemode (MyCardNo,Myaxis, Myhome_dir, Myvel_mode, Mymode,
MyEZ_count);                    ’设置 0 号轴回零模式
    Rt200x_set_profile (MyCardNo,Myaxis,500,1000,0.1,0.1,500);’设置0号轴梯形
速度曲线参数
    Rt200x_home_move (MyCardNo,Myaxis);                    ’0号轴按照设置的模式进
行回零运动
    While (Rt200x_check_done (MyCardNo,Myaxis) == 0)    ’检测运动状态，等待回原
点动作完成
    {
        Application.DoEvents();
    }
    Rt200x_set_position (MyCardNo,Myaxis,0);        ’设置 0 号轴的指令脉冲计数器绝
对位置为0
    ……
```

3.5.3. 点位运动的实现

RT2008 卡在描述运动轨迹时可以用绝对坐标也可以用相对坐标，如图 3-16 所示。两种模式各有优点，如：在绝对坐标模式中用一系列坐标点定义一条曲线，如果要修改中间某点坐标时，不会影响后续点的坐标；在相对坐标模式中，用一系列坐标点定义一条曲线，用循环命令可以重复这条曲线轨迹多次。

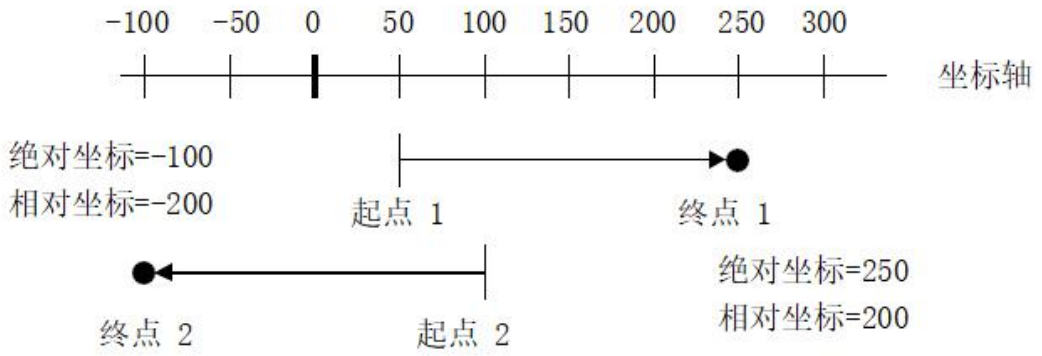


图 3-16 绝对坐标与相对坐标中轨迹终点的不同表达方式

在 RT2008 函数库中距离或位置的单位为 pulse；速度单位为 pulse/s。

RT2008 卡在执行点位运动控制指令时，可使电机按照梯形速度曲线或 S 形速度曲线进行点位运动。

3.5.3.1 梯形速度曲线下的点位运动

梯形速度曲线是位置控制中最基本的速度控制方式。

相关函数如表所示：

名称	功能	参考
Rt200x_set_profile	设置单轴运动速度曲线	5.7单轴运动速度曲线设置函数
Rt200x_pmove	指定轴点位运动	5.8单轴运动函数
Rt200x_check_done	检测指定轴的运动状态	5.6运动状态检测及控制相关函数

例程：执行以梯形速度曲线作点位运动

```
……
Int MyCardNo,Myaxis,Myposi_mode, Mys_mode;
double MyMin_Vel,MyMax_Vel,MyTEB,MyTdec,MyStop_Vel, Mys_para;
Long MyDist;
MyCardNo = 0;          ’卡号
Myaxis = 0;            ’轴号
MyMin_Vel = 500;       ’设置起始速度为500pulse/s
MyMax_Vel = 6000;      ’设置最大速度为6000pulse/s
MyTEB = 0.02;          ’设置加速时间为0.02s
MyTdec = 0.01;         ’设置减速时间为0.01s
MyStop_Vel = 500;      ’设置停止速度为500pulse/s
Rt200x_set_profile(MyCardNo,Myaxis,MyMin_Vel,MyMax_Vel,MyTEB,MyTdec,MyStop_
Vel; ’设置 0 号轴速度曲线参数
Mys_mode = 0;          ’保留参数
Mys_para = 0;          ’ S 段时间为0，即没有S 段运动
Rt200x_set_s_profile (MyCardNo,Myaxis, Mys_mode, Mys_para)  ’设置0号轴 S段
参数为0
MyDist = 50000;        ’设置运动距离为50000pulse
Myposi_mode = 0;       ’设置运动模式为相对坐标模式
Rt200x_pmove (MyCardNo,Myaxis, MyDist, Myposi_mode);’0号轴定长运动
While(Rt2004_check_done(MyCardNo,Myaxis) == 0)  ’判断0轴运动状态
{
Application.DoEvent();
}
……
```

在点位运行过程中，最大速度 Max_Vel 和目标位置 Dist 均可以实时改变，如图 3-17（左）所示。若在减速时改变目标位置，电机的速度将如图 3-17（右）所示发生变化。

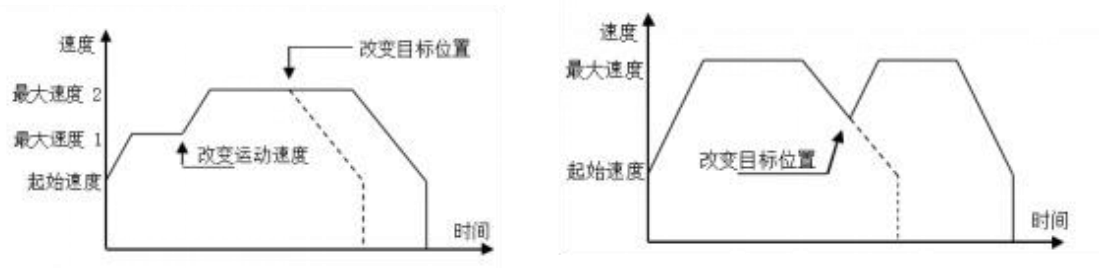


图 3-17 实时改变速度、终点示意图

实现这 2 个功能的函数如表所示：

名称	功能	参考
Rt200x_change_speed	在线改变指定轴的当前运动速度	5.8单轴运动函数
Rt200x_reset_target_position	在线改变指定轴的当前目标位置	

例程：改变速度、改变终点位置

```
……
int MyCardNo,Myaxis,Myposi_mode;
double MyCurr_Vel,MyTEBdec;
long Mydist;
MyCardNo = 0;    ’卡号
Myaxis = 0; ’轴号
Rt200x_set_profile (MyCardNo,Myaxis,500,6000,0.01,0.02,500);’设置梯形速度曲线参数
Rt200x_pmove( MyCardNo,Myaxis,50000,0); ’0号轴定长运动，运动距离50000pulse、相对坐标模式
If(“改变速度条件”)’如果改变速度条件满足，则执行改变速度命令
{
MyCurr_Vel = 9000;     设置新的速度为9000pulse/s
MyTEBdec = 0;    ’保留参数
Rt200x_change_speed (MyCardNo,Myaxis, MyCurr_Vel, MyTEBdec);’执行在线变速指令
}
If(“改变终点位置条件”)’如果改变终点位置条件满足，则执行改变终点位置命令
{
Mydist = 55000; ’改变终点位置为55000pulse
Myposi_mode = 0;’保留参数
Rt200x_reset_target_position( MyCardNo,Myaxis, Mydist, Myposi_mode) ’ 执行在线变位指令
```

```
}
……
```

3.5.3.2 S 形速度曲线运动模式

梯形速度曲线较简单；而 S 速度曲线运动更平稳。相关函数如表所示：

名称	功能	参考
Rt200x_set_profile	设置单轴运动速度曲线	5.7单轴运动速度曲线设置函数
Rt200x_set_s_profile	设置 S 段曲线参数值	
Rt200x_pmove	指定轴点位运动	5.8单轴运动函数
Rt200x_check_done	检测指定轴的运动状态	5.6运动状态检测及控制相关函数

例程：执行以 S 形速度曲线作点位运动

```
……
Int MyCardNo,Myaxis,Mys_mode;
double Mys_para;
MyCardNo = 0;   ’卡号
Myaxis = 0; ’轴号
Mys_mode = 0;   ’保留参数
Mys_para = 0.02;’  S段时间为0.02s
Rt200x_set_profile( MyCardNo,Myaxis,500,6000,0.05,0.05,500);’设置0号轴速度曲线参数
Rt200x_set_s_profile (MyCardNo,Myaxis, Mys_mode, Mys_para); ’设置0号轴S段参数
Rt200x_pmove( 0,0,50000,0); ’0号轴定长运动，运动距离为50000pulse、相对坐标模式
While(Rt200x_check_done(MyCardNo,Myaxis) == 0)  ’判断 0 轴运动状态
{
Application.DoEvent();
}
……
```

如果因为距离太短或加速太慢原因导致电机速度在加速段不能升至设定的最大值 Max_Vel 时，理论上加速段将突然切换至减速段，从而引起该轴出现较大震动。为了避免出现这种问题，RT2008 运动控制卡内置有自动调整功能，使得加减速段的过渡保持平滑，如图 3-18 所示。

在 S 形速度曲线下的点位运动过程中，也可以调用 Rt200x_change_speed 和 Rt200x_reset_target_position 函数实时改变运行速度和目标位置。但多轴插补运

行情况下不能实时改变运行速度和目标位置。

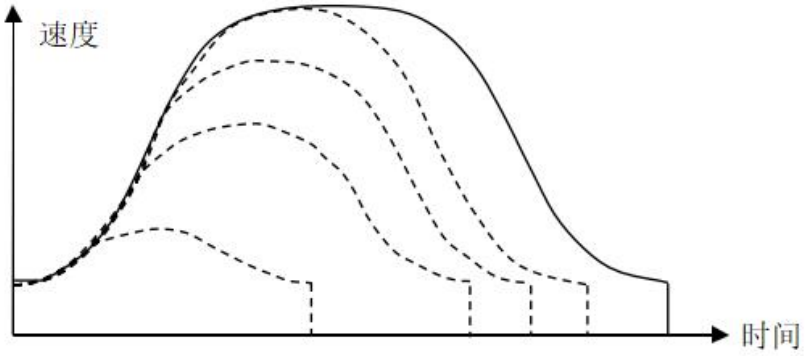


图 3-18 自动降速避免尖三角形

3.5.3.3 多轴联动

多轴同时做点位运动，称之为多轴联动。

RT2008 卡可以控制多个电机同时执行 Rt200x_pmove 这类单轴运动函数。所谓同时执行，是在程序中顺序调用 Rt200x_pmove 等函数，因为程序执行速度很快，在几微秒内电机都开始运动，感觉是同时开始运动。

多轴联动在各轴速度设置不当时，各轴停止时间不同、在起点与终点之间运动的轨迹也不是直线。

如果从起点到终点都需要按照规定的路径运动，就必须采用插补运动功能。

3.5.4. 连续运动的实现

连续运动中，RT2008 卡可以控制电机以梯形或 S 形速度曲线在指定的加速时间内从起始速度加速至最大速度，然后以该速度一直运行，直至调用停止指令或者该轴遇到限位信号才会按启动时的速度曲线减速停止。相关函数如表所示：

名称	功能	参考
Rt200x_vmove	指定轴连续运动	5.8单轴运动函数
Rt200x_stop	指定轴停止运动	5.6运动状态检测及控制相关函数

在执行连续运动过程中，可以调用 Rt200x_change_speed 实时改变速度。注意：在以 S 形速度曲线连续运动时，改变最大速度最好在加速过程已经完成的恒速段进行。图 3-19 为梯形和 S 形速度曲线下连续运动中变速和减速停止过程的速度曲线。

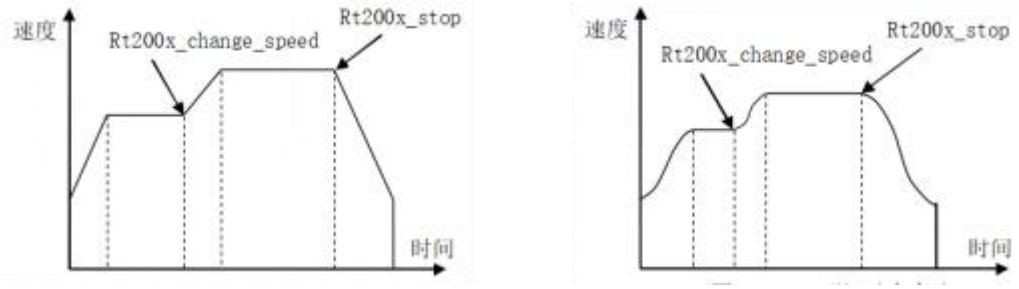


图 3-19 梯形运动（左）与 S 型运动（右）中变速

例程：以 S 形速度曲线加速的连续运动及变速、停止控制

```
……
Int MyCardNo,Myaxis,Mydir,Mystop_mode;
double MyCurr_Vel;
MyCardNo = 0;    ’卡号
Myaxis = 0; ’轴号
Mydir = 1;  ’设置连续运动方向为正方向
Rt200x_set_profile (MyCardNo,Myaxis,100,1000,0.1,0.1,100)    ’设置0号轴运动参数
Rt200x_set_s_profile( MyCardNo,Myaxis,0,0.002)  ’设置0号轴S段时间为0.002s
Rt200x_vmove (MyCardNo,Myaxis,Mydir)’0号轴执行连续运动
If(“改变速度条件”)’如果改变速度条件满足，则执行改变速度命令
{
    MyCurr_Vel = 1200;  ’设置新的速度
    Rt200x_change_speed (MyCardNo,Myaxis, MyCurr_Vel,0);’执行在线变速指令
}
If(“停止条件”)’如果运动停止条件满足，则执行减速停止命令
{
    Mystop_mode = 0;’设置停止模式为减速停止
    Rt200x_stop (MyCardNo,Myaxis, Mystop_mode); ’0 号轴减速停止
}
……
```

3.5.5. 插补运动的实现

插补运动是为了实现轨迹控制，运动控制卡按照一定的控制策略控制多轴联动，使运动平台用微小直线段精确地逼近轨迹的理论曲线，保证运动平台从起点到终点上的所有轨迹点都控制在允许误差范围内。

3.5.5.1 直线插补运动

RT2008 卡可以进行任意 2~3 轴直线插补，插补计算由控制卡的硬件执行，

用户只需将插补运动的速度、加速度、终点位置等参数写入相关函数即可。调用直线插补函数时，调用者需提供矢量速度，包括其最大矢量速度 Max_Vel 和加减速时间参数。相关函数如表所示：

名称	功能	参考
Rt200x_set_vector_profile_multicoor	设置插补速度	5.9插补速度曲线设置函数
Rt200x_line_multicoor	直线插补运动	5.10插补运动函数

例程：XY 轴直线插补

```
……
Int MyCardNo,MyCrd,MyaxisNum,Myposi_mode;
Int [2]AxisArray;
Double MyMin_Vel,MyMax_Vel,MyTEB,MyTdec,MyStop_Vel;
Long [2]Dist;
MyCardNo = 0; ’卡号
MyCrd = 0; ’参与插补运动的坐标系
MyMin_Vel = 0; ’保留参数
MyMax_Vel = 5000; ’插补运动最大矢量速度 5000pulse/s
MyTacc = 0.1; ’插补运动加减速时间 0.1s
MyTdec = 0; ’保留参数
MyStop_Vel = 0; ’保留参数
Rt200x_set_vector_profile_multicoor (MyCardNo, MyCrd, MyMin_Vel, MyMax_Vel,
MyTEB, MyTdec, MyStop_Vel); ’设置插补速度曲线参数
MyaxisNum = 2;  ’插补运动轴数为2
AxisArray(0) = 0;   ’定义插补0轴为X轴
AxisArray(1) = 1;   ’定义插补1轴为Y轴
Dist[0] = 20000; ’定义 X 轴运动距离为 20000pulse
Dist[1] = 40000; ’定义 Y 轴运动距离为 40000pulse
Myposi_mode = 0; ’插补运动模式为相对坐标模式
Rt200x_line_multicoor(MyCardNo,MyCrd,MyaxisNum,AxisArray,Dist,Myposi_mode);
’执行直线插补运动
While(Rt200x_check_done_multicoor(MyCardNo,MyCrd) == 0) ’判断坐标系0状态
{
    Application.DoEvent();
}
……
```

该例程使 X，Y 轴进行相对坐标模式直线插补运动，其相关参数为：ΔX=20000pulse,ΔY=40000pulse,最大矢量速度=5000pulse/s（0 轴,1 轴分速度为 2000，4000pulse/s）梯形加减速时间=0.1s。

3.5.5.2 圆弧插补运动

RT2008 卡的任意两轴之间可以进行圆弧插补，圆弧插补分为相对位置圆弧插补和绝对位置圆弧插补，运动的方向分为顺时针（CW）和逆时针（CCW）。相关函数如表所示：

名称	功能	参考
Rt200x_set_vector_profile_multicoor	设置插补速度	5.9插补速度曲线设置函数
Rt200x_arc_move_multicoor	圆弧插补运动	5.10插补运动函数

例程：XY 轴间的圆弧插补

```
……
Int MyCardNo,MyCrd,MyArc_Dir,Myposi_mode;
Int [2]AxisArray;
Long [2]Pos;
Long [2]Cen;
MyCardNo = 0;   ’定义卡号
MyCrd = 0;  ’定义参与插补运动的坐标系
AxisArray(0)=0; ’定义0轴为插补 X轴
AxisArray(1)=1; ’定义1轴为插补 Y轴
Pos[0] = 5000;
Pos[1] = -5000; ’设置终点坐标
Cen[0] = 5000;
Cen[1] = 0; ’设置圆心坐标
MyArc_Dir = 0;      ’设置圆弧方向为顺时针
Myposi_mode = 1;’设置圆弧插补模式为绝对坐标模式
Rt200x_set_vector_profile_multicoor (MyCardNo,MyCrd,0,2000,0.1,0,0);’设置矢量速度曲线
Rt200x_arc_move_multicoor(MyCardNo,MyCrd,AxisArray,Pos,Cen,MyArc_Dir,Myposi_mode); ’XY轴执行顺时针方向绝对圆弧插补运动，
While(Rt200x_check_done_multicoor(MyCardNo,MyCrd) == 0) ’判断坐标系0状态
{
    Application.DoEvent();
}
……
```

3.5.6. PVT 运动功能的实现

RT2008 卡共提供四种 PVT 模式，分别为 PTT、PTS、PVT、PVTS 模式。其中 PTT、PTS 运动模式用于单轴速度规划功能，PVT、PVTS 运动则用于多轴轨迹规划功能，用户可以根据实际需求选择适合的 PVT 模式。

3.5.6.1 单轴任意速度规划功能的实现

RT2008 卡中提供了两种 PVT 模式来实现单轴任意速度规划的功能，分别为 PTT 运动模式和 PTS 运动模式。PTT 运动模式是用于单轴梯形速度的规划，而 PTS 运动模式则用于单轴 S 形速度的规划。

PTT 运动模式

PTT 模式非常灵活，能够实现单轴任意速度规划。用户通过直接输入位置和时间参数描述运动规律。相关函数如表所示：

名称	功能	参考
Rt200x_PttTable	向指定数据表传送数据，采用 PTT 描述方式	5.11PVT运动函数
Rt200x_PvtMove	启动 PVT 运动	

例程：PTT 模式运动

```
……
Int MyCardNo,MyAxisNum;
Int [1]My_AxisList;
Double [7]MyPTime   ’定义 PTT 的时间数组
Long [7] MyPPos;    ’定义 PTT 的位置数组
Int MyCount;
MyCardNo = 0; ’卡号为 0
My_AxisList(0) = 0; ’0 号轴参与 PTT 运动
MyCount = 7; ’有 7 组数据
MyPPos[0] = 0;
MyPTime[0] = 0; ’定义 PVT 数组数据
MyPPos[1] = 1500;
MyPTime[1] = 1;
MyPPos[2] = 5500;
MyPTime[2] = 2;
MyPPos[3] = 14000;
MyPTime[3] = 3;
MyPPos[4] = 38000;
MyPTime[4] = 5;
MyPPos[5] = 65000;
MyPTime[5] = 8;
MyPPos[6] = 68000;
MyPTime[6] = 9;
Rt200x_PttTable( MyCardNo, My_AxisList, MyCount, MyPTime, MyPPos);  ’  以PTT 描述方式，向 0 号轴传送 PVT 数据
MyAxisNum = 1;  ’参与 PVT 运动的轴数为 1
Rt200x_PvtMove (MyCardNo,MyAxisNum,My_AxisList);’启动 PVT 运动
```

……

PTS 运动模式

PTS 运动模式是 PTT 的扩展功能模式，可以使各数据点的速度过渡更加平滑。用户通过输入位置、时间、百分比参数描述运动规律。数据点的百分比参数是指：相邻 2 个数据点之间加速度的变化时间占速度变化时间的百分比。相关函数如表所示：

名称	功能	参考
Rt200x_PtsTable	向指定数据表传送数据，采用PTS描述方式	5.11PVT运动函数
Rt200x_PvtMove	启动PVT运动	

例程：PTS 模式运动

```
……
Int MyCardNo, MyAxisNum;
Int [1] My_AxisList;
Double [7]MyPTime;  '定义 PTS 数据的时间数组
Long [7]MyPPos; '定义 PTS 数据的位置数组
Double [7] MyPPer;  '定义 PTS 数据的百分比数组
int MyCount;
MyCardNo = 0; ' 卡号为
y_AxisList[0] = 0;  ' 0 号轴参与PTS 运动
MyCount = 7;' 有7 组数据'
 定义 PTS 数据
MyPPos[0] = 0, MyPTime[0] = 0, MyPPer[0] = 0;
MyPPos[1] = 1500, MyPTime[1] = 1, MyPPer[1] = 20;
MyPPos[2] = 5500, MyPTime[2] = 2, MyPPer[2] = 0;
MyPPos[3] = 14000, MyPTime[3] = 3, MyPPer[3] = 60;
MyPPos[4] = 38000, MyPTime[4] = 5, MyPPer[4] = 0;
MyPPos[5] = 65000, MyPTime[5] = 8, MyPPer[5] = 20;
MyPPos[6] = 68000, MyPTime[6] = 9, MyPPer[6] = 80;
Rt200x_PtsTable( MyCardNo,My_AxisList, MyCount, MyPTime, MyPPos, MyPPer);
'以 PTS 描述方式向 0 号轴传送 PVT 数据
MyAxisNum = 1;  ' 参与PVT 运动的轴数为1
Rt200x_PvtMove( MyCardNo, MyAxisNum, My_AxisList);  '启动 PVT 运动
……
```

3.5.6.2 多轴高级轨迹规划功能的实现

RT2008 卡具有 PVT 高级运动曲线规划的功能，当用户需要规划一些特殊的运动轨迹而使用单轴运动及插补运动无法满足需求时，可以尝试使用 PVT 来规划自己的运动轨迹。

RT2008 卡共提供了两种 PVT 模式来实现多轴轨迹规划的功能，分别为 PVT、PVTS 运动模式。PVT 模式用于对各点的位置、时间、速度都有要求的轨迹规划，PVTS 模式用于只对各点的位置、时间有要求，而对各点的速度无太多要求的轨迹规划。

PVT 运动模式

PVT 模式使用一系列数据点的位置、速度、时间参数描述运动规律。相关函数如表所示：

名称	功能	参考
Rt200x_PvtTable	向指定数据表传送数据，采用PVT描述方式	5.11PVT运动函数
Rt200x_PvtMove	启动PVT运动	

当设置的各点P、V、T数据不合理时，很难得到理想的轨迹曲线。理想轨迹上取点越多，实际轨迹越接近理想轨迹。

例程：PVT 模式运动

```
……

Long [11] MyPPosX;  ’定义数组，用于存储 PVT 的位置数据（X 轴）

Long [11] MyPPosY   ’定义数组，用于存储 PVT 的位置数据（Y 轴）

int a, b, i;

a = 9000, b = 7000; ’ 定义椭圆长半轴、短半轴长

for(int j =0;j<11;j++) ’ 计算各点的  X、Y坐标

{

    MyPPosX[i] = a * Cos((10 - i) * 3.14159 / 10) + a;
    MyPPosY[i] = b * Sin((10 - i) * 3.14159 / 10);

}

Double [11]MyPVelX;      ’定义数组，用于存储  PVT中的速度数据（X 轴）
Double [11] MyPTimeX;    ’定义数组，用于存储  PVT中的时间数据（X 轴）
Double [11] MyPVelY;     ’定义数组，用于存储  PVT中的速度数据（Y 轴）
Double [11] MyPTimeY;    ’定义数组，用于存储  PVT中的时间数据（Y 轴）
Double MyWVel;                ’ 定义角速度
For(int j =0;j<11;j++)
{
MyPTimeX[j] = i;                        ’存储 X 轴各点时间数据
```

```
    MyPTimeY[j] = i;                          ’ 存储Y 轴各点时间数据
    }

    MyWVel = -3.14159 / 10;                   ’ 计算角速度W

    MyPVelX[0] = 0,MyPVelX[10] = 0; ’起始点与终止点X 轴速度设为 0
    MyPVelY[0] = 0, MyPVelY[10] = 0; ’起始点与终止点Y 轴速度设为 0
    For (j = 0;j<9;j++)
    {
    MyPVelX(j + 1) = -a * Sin((10 - j - 1) * 3.14159 / 10) * MyWVel ’计算其他点X
轴速度
    MyPVelY(j + 1) = b * Cos((10 - j - 1) * 3.14159 / 10) * MyWVel ’计算其他点Y
轴速度
    }
    int MyCardNo;
    int [2]My_AxisList;                          ’定义PVT运动的卡号、轴列表变量
    int MyCountX;                                ’定义X轴的PVT数据点编号变量
    int MyCountY;                                ’定义Y轴的PVT数据点编号变量
    MyCardNo = 0;                                 ’ 0号卡
    My_AxisList[0] = 0, My_AxisList[1] = 1;       ’ 0、1号轴(即X、Y轴)参与PVT运
动
    MyCountX = 11,MyCountY = 11;                   ’ 11组数据
    Rt200x_PvtTable(MyCardNo, My_AxisList, MyCountX, MyPTimeX, MyPPosX, MyPVelY);’
以PVT 描述方式向X 轴传送 PVT 数据
    Rt200x_PvtTable (MyCardNo, My_AxisList, MyCountY, MyPTimeY, MyPPosY, MyPVelY);’
以PVT 描述方式向Y 轴传送 PVT 数据
    int My_AxisNum;
    My_AxisNum = 2; ’ 参与 PVT 运动的轴数为 2
    Rt200x_PvtMove (MyCardNo, My_AxisNum, My_AxisList); ’ 启动两轴PVT 运动
    ……
```

PVTS 运动模式

PVTS 运动模式只需要定义理想轨迹上数据点的位置和时间，以及起点速度和终点速度。运动控制卡根据各数据点的位置、时间参数计算各点之间的轨迹位置和速度，确保各段轨迹的速度连续、加速度连续。相关函数如表所示：

名称	功能	参考
Rt200x_PvtsTable	向指定数据表传送数据，采用 PVTS 描述方式	5.11PVT运动函数
Rt200x_PvtMove	启动PVT运动	

例程：PVTS 模式运动

```
……
```

```
Double [11] MyPTimeX;
Double MyPVelBeginX,MyPVelEndX;
Double [11]MyPTimeY;
Double MyPVelBeginY,MyPVelEndY;
Double [11] MyPTimeZ;
Double MyPVelBeginZ,MyPVelEndZ;
Long [11] MyPPosX;
Long [11]MyPPosY;
Long [11]MyPPosZ;
Int MyCountX,MyCountY,MyCountZ;
Int  MyCardNo,My_AxisNum;
Int [2]My_AxisList;
Float pi;   '定义圆周率
Int  R, i;  '定义空间圆弧半径，循环变量
R = 15000;      '定义圆半径
pi = 3.141592654;'定义圆周率
MyCountX = 11, MyCountY = 11, MyCountZ = 11;'设置X、Y、Z 轴的数据点数
For (i = 0;i<11;i++)
{
    MyPPosX[i] = R * Cos(pi / 6) * Cos((10 - i) * pi / 10) + R * Cos(pi / 6);
    '计算X 轴各点位置坐标
    MyPPosY[i] = R * Sin(pi / 6) * Cos((10 - i) * pi / 10) + R * Sin(pi / 6);
    '计算Y轴各点位置坐标
    MyPPosZ[i] = R * Sin((10 - i) * pi / 10)'计算Z轴各点位置坐标
}
For (i = 0;i<11;i++)
{
    MyPTimeX[i] = i;'计算 X 轴各点时间
    MyPTimeY[i] = i;'计算 Y 轴各点时间
    MyPTimeZ[i] = i;'计算 Z 轴各点时间

}
MyPVelBeginX = 0, MyPVelEndX = 0;   'X 轴的起点速度及终点速度为 0
MyPVelBeginY = 0, MyPVelEndY = 0;   'Y 轴的起点速度及终点速度为 0
MyPVelBeginZ = 0, MyPVelEndZ = 0;   'Z 轴的起点速度及终点速度为 0
MyCardNo = 0;    ' 0 号卡运动
My_AxisList[0] = 0, My_AxisList[1] = 1, My_AxisList[2] = 2;  '0、1、2 号轴（即
X、Y、Z 轴）参加PVT 运动
Rt200x_PvtsTable (MyCardNo, My_AxisList, MyCountX, MyPTimeX, MyPPosX,
MyPVelBeginX,MyPVelEndX)    '以PVTS 描述方式向X 轴传送PVT 数据
Rt200x_PvtsTable (MyCardNo, My_AxisList, MyCountY, MyPTimeY, MyPPosY,
MyPVelBeginY,MyPVelEndY);   '以PVTS 描述方式向Y 轴传送PVT 数据
```

```
Rt200x_PvtsTable( MyCardNo, My_AxisList, MyCountZ, MyPTimeZ, MyPPosZ,
MyPVelBeginZ, MyPVelEndZ);  '以PVTS 描述方式向Z 轴传送PVT 数据
My_AxisNum = 3; '3 个轴参与PVT 运动
Rt200x_PvtMove( MyCardNo, My_AxisNum, My_AxisList); '启动 PVT 运动
……
```

3.5.7. 异常减速停止时间设置功能的实现

RT2008 卡支持异常减速停止时间设置功能，用户根据现场实际需求情况设定减速停止时间可达到理想的减速效果，相关函数如表所示：

名称	功能	参考
Rt200x_set_dec_stop_time	设置减速停止时间	5.18异常信号接口函数
Rt200x_get_dec_stop_time	读取减速停止时间设置	

当发生异常停止时，如：调用Rt200x_stop函数、限位信号（软硬件）被触发等进行减速停止时，减速停止时间都为Rt200x_set_dec_stop_time函数里设置的减速时间。

例程：设置异常减速停止时间为 0.5 秒

```
……
Int CardNo,Axis;
Double Stop_time;
CardNo = 0; ' 卡号
Axis = 0;   ' 指定轴号：第 0 轴
Stop_time = 0.5;' 异常减速停止时间为 0.5 秒
Rt200x_set_dec_stop_time (CardNo, Axis, Stop_time); ' 设置轴异常减速停止时间
……
```

3.5.8. 手轮运动功能的实现

RT2008 运动控制卡支持单轴手轮运动功能。该功能允许用户设置一个手轮通道对应一个运动轴进行运动。相关函数如表所示：

名称	功能	参考
Rt200x_set_handwheel_inmode	设置单轴手轮运动控制输入方式	5.14手轮功能函数

Rt200x_handwheel_move	启动手轮运动	

RT2008有八个手轮通道，与八个轴一一对应，可以同时控制八个轴运动。

当启动手轮运动后，只有发送Rt200x_stop或Rt200x_emg_stop命令才会退出手轮模式。

例程：单轴手轮运动

```
……
Int  MyCardNo,Myaxis,Myinmode;
Long Mymulti;
Double Myvh;
MyCardNo = 0;’ 0 号卡
Myaxis = 0; ’ 设置运动轴为0号轴
Myinmode = 0;   ’ 设置手轮输入方式为 AB相
Mymulti = 10;   ’ 设置手轮输入倍率为10
Myvh = 0;   ’ 保留参数
Rt200x_set_handwheel_inmode(MyCardNo,Myaxis,Myinmode,Mymulti, Myvh);’ 设置手轮运动
Rt200x_handwheel_move( MyCardNo,Myaxis);’ 启动手轮运动
Rt200x_stop (MyCardNo,Myaxis,1);’ 立即停止手轮运动
……
```

3.5.9. 编码器检测的实现

RT2008 卡的反馈位置计数器是一个 32 位正负计数器，对通过控制卡编码器接口 EA，EB 输入的脉冲（如编码器、光栅尺反馈脉冲等）进行计数。相关函数如表所示：

名称	功能	参考
Rt200x_set_counter_inmode	设置编码器输入口的计数方式	5.15编码器函数
Rt200x_get_encoder	读取编码器反馈的脉冲计数值	
Rt200x_set_encoder	设置编码器的脉冲计数值	

例程：编码器检测

```
……
Int MyCardNo,Myaxis,Mymode;
Long Myencoder_value, MyX_Position;
```

```
MyCardNo = 0;   ’卡号
Myaxis = 0; ’轴号
Mymode = 3; ’设置编码器的计数方式为 4 倍频，AB 相
Rt200x_set_counter_inmode (MyCardNo,Myaxis,Mymode); ’设置0号轴的编码器计数方式
Myencoder_value = 0;’设置0号轴的计数初始值为0
Rt200x_set_encoder (MyCardNo,Myaxis, Myencoder_value);’设置0号轴的计数初始值
MyX_Position = Rt200x_get_encoder (MyCardNo,Myaxis);’读轴 0 的计数器数值至变量MyX_Position
……
```

3.5.10. 检测轴到位状态功能的实现

RT2008 卡提供了检测轴到位状态函数，使用这些函数可以设置单轴运动中允许的误差范围，并检测单轴运动是否处于允许的误差范围内。相关函数如表所示：

名称	功能	参考
Rt200x_set_factor_error	设置位置误差带	5.19检测轴到位状态函数
Rt200x_check_success_pulse	检测指令到位	
Rt200x_check_success_encoder	检测编码器到位	

该功能检测只适用于单轴运动。

检测函数请在Rt200x_check_done检测到轴停止后调用，函数调用后会等待轴到位后返回，如果调用函数100ms内未到位，函数超时返回认为不到位。

例程：检测轴到位状态

```
……
Int  MyCardNo,Myaxis;
Double  MyFactor;
Long  MyError;
MyCardNo = 0;   ’0 号卡
Myaxis = 0; ’设置运动轴为 0 号轴
MyFactor = 5;   ’设置编码器系数为 5
MyError = 10;   ’设置位置误差带为 10 pulse
Rt200x_set_factor_error (MyCardNo, Myaxis, MyFactor, MyError);  ’设置位置误差带
```

```
Rt200x_pmove (MyCardNo,Myaxis,1000,1);  ’ 0 号轴定长运动，运动距离为 1000 pulse、绝对模式
While (Rt200x_check_done(MyCardNo,Myaxis) == 0) ’ 判断 0 号轴运动状态
    Application.DoEvent();
If (Rt200x_check_success_encoder(MyCardNo,Myaxis) == 0)’ 检测0号轴到位状态
    Console.WriteLine("编码器不到位!");
Else
    Console.WriteLine("编码器到位!");
……
```

运行结果说明：

运动的目标位置脉冲数为 1000pulse，假设当前编码器反馈脉冲数为 199pulse，由于误差带设为 10pulse，编码器系数设为 5 处理如下：

199*5=995(pulse)

1000-995=5(pulse)

在误差带范围[-10,10]之内，此时则认为编码器到位，否则认为编码器不到位。

3.5.11. 通用 I/O 控制的实现

3.5.11.1 I/O 普通控制功能

用户可以使用 RT2008 卡上的数字 I/O 口用于检测开关信号、传感器信号等输入信号，或者控制继电器、电磁阀等输出设备的信号。相关函数如表所示：

名称	功能	参考
Rt200x_read_inbit	读取指定控制卡的某一位输入口的电平状态	5.13通用输入输出IO函数
Rt200x_write_outbit	对指定控制卡的某一位输出口置位	
Rt200x_read_outbit	读取指定控制卡的某一位输出口的电平状态	
Rt200x_read_inport	读取指定控制卡的全部输入口的电平状态	
Rt200x_read_outport	读取指定控制卡的全部输出口的电平状态	
Rt200x_write_outport	设置指定控制卡的全部输出口的电平状态	

在使用Rt200x_write_outport对运动控制卡的全部输出口进行置位，使用Rt200x_read_inport、Rt200x_read_outport进行IO电平读取显示时，应该使用十六进制数进行赋值（尽量避免使用十进制数，特别是在不支持无符号变量的开发环境下）。在对IO电平进行控制与读取时，使用十六进制数赋值远比使用十进制数赋值更加直观、方便。

例程：读取第 0 号卡的通用输入口 1 的电平值，并对通用输出口 3 置高电平

```
……
Int  MyCardNo,MyInbitno,MyInValue,MyOutbitno,MyOutValue;
MyCardNo = 0;   ’卡号
MyInbitno = 1;  ’定义通用输入口 1
MyInValue = Rt200x_read_inbit(MyCardNo, MyInbitno); ’读取通用输入口1的电平值，并赋值给变量MyInValue
MyOutbitno = 3; ’定义通用输出口3
MyOutValue = 1; ’定义输出电平为高
Rt200x_write_outbit (MyCardNo, MyOutbitno, MyOutValue); ’对通用输出口 3 置高电平
……
```

例程：读取全部输入 IO 口的电平值并进行显示，对全部输出 IO 口的电平进行初始化

```
……
Int MyCardNo,MyInport,MyOutport;
Long MyInportValue,MyOutportValue;
String MyInportValueTemp;
MyCardNo = 0;   ’ 卡 号
MyInport = 0;   ’输入端口组号
MyInportValue = Rt200x_read_inport(MyCardNo, MyInport);’读取所有输入 IO口电平值，并赋值给变量 MyInportValue
MyInportValueTemp = Hex(MyInportValue); ’ 转 换 成 十 六 进 制
MyInTextShow = MyInportValueTemp;   ’显示在文本框MyInTextShow中
MyOutport = 0;  ’保留参数，固定为0
MyOutportValue = 0xFFFFFBFA;’0x表示十六进制，定义输出口电平值，输出口0、2、10为低电平，其余端口为高电平
Rt200x_write_outport(MyCardNo,MyOutport, MyOutportValue);   ’对全部输出口进行电平赋值
……
```

3.5.11.2 I/O 延时反转功能

RT2008 卡支持 I/O 延时翻转功能。该函数执行后，首先输出一个与当前电平相反的信号，延时设置的时间后，再自动翻转一次电平。相关函数如表所示：

名称	功能	参考
Rt200x_reverse_outbit	IO输出延时翻转	5.13通用输入输出IO函数

例程：对通用输出 IO 端口 0 进行延时翻转动作

```
……
Int MyCardNo,MyOutbitno;
Long MyDelayTime;
MyCardNo = 0;    ’卡号
MyOutbitno = 0; ’通用输出口0
MyDelayTime = 0.5; ’延时时间0.5s
Rt200x_reverse_outbit (MyCardNo, MyOutbitno, MyDelayTime);  ’启动 IO 延时翻转
……
```

运行结果：图 3-20 为该例程对应的输出端口 0 的电平状态时序图（假设输出端口 0 的初始电平状态为低电平）。

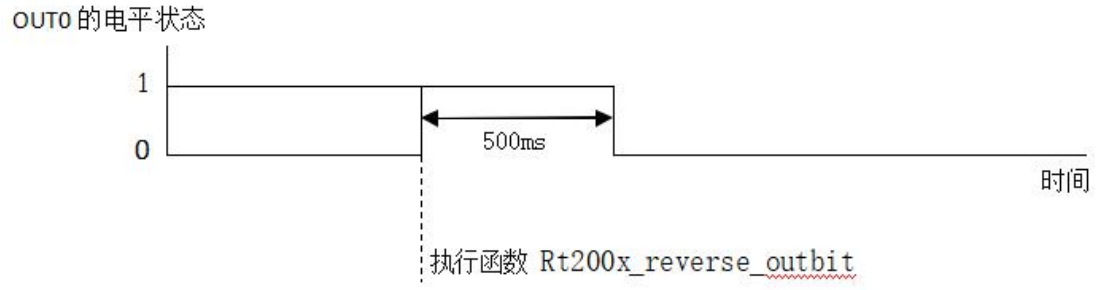


图 3-20 输出端口 0 的电平状态时序图

3.5.12. 位置比较功能的实现

RT2008 卡提供了位置比较功能，位置比较的一般步骤是：

1、配置比较器；

2、清除比较器；

3、添加/更新比较位置点；

4、开始运动并查看比较状态。

3.5.12.1 一维低速位置比较功能

RT2008 卡对每个轴都提供了一组一维低速位置比较，每轴最多都可以添加 256 个比较点。一维低速位置比较的触发延时时间小于 1ms。相关函数如表所示：

名称	功能	参考
Rt200x_compare_set_config	设置一维位置比较器	5.17位置比较函数
Rt200x_compare_clear_points	清除一维位置比较点	
Rt200x_compare_add_point	添加一维位置比较点	
Rt200x_compare_get_current_point	读取当前一维比较点位置	
Rt200x_compare_get_points_runned	查询已经比较过的一维比较点个数	
Rt200x_compare_get_points_remained	查询可以加入的一维比较点个数	

每轴的位置比较都是独立进行的。

执行位置比较时，每个比较点的触发是按照添加的比较点顺序执行的，即如果有一个比较点没有被触发比较动作，那么后面的比较点是不会起作用的。

例程：一维低速位置比较

```
……
Int MyCardNo,Myaxis,Myenable,Mycmp_source,Mydir,Myaction;
Long Mypos,Myactpara;
MyCardNo = 0;   ’卡号
Myaxis = 0; ’轴号
Myenable = 1;   ’设置比较器使能
Mycmp_source = 0;   ’设置比较源为指令位置
Rt200x_compare_set_config (MyCardNo,Myaxis,Myenable,Mycmp_source);’设置0号轴比较器
Rt200x_compare_clear_points (MyCardNo,Myaxis);  ’清除0号轴的比较点及比较点个数
Mypos = 10000;  ’设置比较位置为10000pulse
Mydir = 1;  ’设置比较模式为大于等于
Myaction = 3;   ’设置触发功能为 IO电平取反
Myactpara = 0;  ’设置输出IO端口0触发功能
Rt200x_set_position (MyCardNo,Myaxis,0);’设置 0 号轴的指令脉冲计数器绝对位置为0
Rt200x_compare_add_point (MyCardNo,Myaxis, Mypos, Mydir, Myaction, Myactpara)’添加比较点，位置10000pulse，模式大于等于，触发时动作为输出端口0电平取反
Mypos = 20000;  ’设置比较位置为30000pulse
Mydir = 1;  ’设置比较模式为大于等于
```

```
Myaction = 1;   ’设置触发功能为 IO端口置高电平
Myactpara = 3;  ’设置输出IO端口3触发功能
Rt2004x_compare_add_point (MyCardNo,Myaxis, Mypos, Mydir, Myaction,
Myactpara)’添加比较点，位置20000pulse，模式大于等于，触发时动作为输出端口3置高电平
Rt200x_set_profile (MyCardNo,Myaxis,2000,10000,0.1,0.1,2000);   ’设置梯形速度曲线参数
Rt200x_pmove (MyCardNo,Myaxis,50000,0); ’0号轴定长运动，运动距离为50000pulse、相对模式
……
```

运行结果：

当运动到 10000pulse 时，通用输出口 0 电平将翻转；当运动到 20000pulse 时，通用输出口 3 输出高电平。

3.5.12.2 二维低速位置比较功能

RT2008 卡提供了一组二维低速位置比较，最多都可以添加 256 个比较点。二维低速位置比较的触发延时时间小于 1ms。相关函数如表所示：

名称	功能	参考
Rt200x_compare_set_config_extern	设置二维位置比较器	5.17位置比较函数
Rt200x_compare_clear_points_extern	清除二维位置比较点	
Rt200x_compare_add_point_extern	添加二维位置比较点	
Rt200x_compare_get_current_point_extern	读取当前二维位置比较点位置	
Rt200x_compare_get_points_runned_extern	查询已经比较过的二维比较点个数	
Rt200x_compare_get_points_remained_extern	查询可以加入的二维比较点个数	

执行位置比较时，每个比较点的触发是按照添加的比较点顺序执行的，即如果有一个比较点没有被触发比较动作，那么后面的比较点是不会起作用的。

例程：二维低速位置比较

```
……
```

```
Int MyCardNo;
Int [2]My_axis; ’ 定义进行位置比较的卡号，轴号列表
Long [2] My_ComPos; ’ 定义进行位置比较的位置列表
Int [2] My_ComDir; ’ 定义进行位置比较的方向列表
Long [2] Pos;
Long [2] Cen; ’ 定义圆弧插补的终点位置及圆心坐标列表
MyCardNo = 0; ’ 卡号
Rt200x_compare_set_config_extern(MyCardNo,1,0);’ 设置比较器，0 号卡二维位置比较使能，比较源指令位置
Rt200x_compare_clear_points_extern(MyCardNo); ’ 清除位置比较点
My_axis[0]=0,My_axis[1]=1; ’ 设置比较轴号列表
My_ComPos[0]=-5000, My_ComPos[1]=5000; ’ 设置比较位置列表
My_ComDir[0]=0 ,My_ComDir[1]=1; ’ 设置比较模式列表
Rt200x_set_position (MyCardNo, My_axis,0);
Rt200x_set_position (MyCardNo, My_axis,0); ’ 设置轴的指令脉冲计数器绝对位置为 0
Rt200x_compare_add_point_extern (MyCardNo, My_axis, My_ComPos, My_ComDir,3,0); ’ 添加比较点，触发时动作为输出端口 0 电平取反
My_ComPos[0]=5000, My_ComPos[1]=5000; ’ 设置比较位置列表
My_ComDir[0]=1, My_ComDir[1]=0; ’ 设置比较模式列表
Rt200x_compare_add_point_extern(MyCardNo, My_axis, My_ComPos, My_ComDir,1,3); ’ 添加比较点，触发时动作为输出端口 3 置为高电平
Pos[0] = 0, Pos[1] = 0; ’ 设置终点坐标
Cen[0] = 0, Cen[1] = 5000; ’ 设置圆心坐标
Rt200x_set_vector_profile_multicoor (MyCardNo,0,0,2000,0.1,0,0);’ 设置矢量速度曲线
Rt200x_arc_move_multicoor( MyCardNo,0, My_axis, Pos, Cen,0,0); ’ 圆弧插补运动
……
```

3.5.13. 高速位置锁存功能的实现

3.5.13.1 高速单次位置锁存功能

RT2008 卡提供了高速单次位置锁存功能，位置锁存无触发延时时间，当捕获到位置锁存信号后立即锁存当前位置。相关函数如表所示：

名称	功能	参考
Rt200x_set_ltc_mode	设置指定轴的LTC信号	5.16高速位置锁存函数
Rt200x_set_latch_mode	设置锁存方式	

Rt200x_get_latch_value	从控制卡内读取编码器锁存器的值	
Rt200x_get_latch_flag	从控制卡内读取指定卡内锁存器的标志位	
Rt200x_reset_latch_flag	复位指定卡的锁存器的标志位	

在单次锁存中，多次触发高速锁存口只锁存第一次触发位置，只有调用函数清除锁存状态方可再次锁存。

例程：高速单次位置锁存

……

```
Int  MyCardNo,Myaxis,Myltc_logic,Myltc_mode;
Int  Myall_enable,Mylatch_source,Mytriger_chunnel;
Double  Myfilter;
Long  My_latch_Value;   ’定义锁存值
MyCardNo = 0;   ’卡号
Myaxis = 0; ’轴号
Myltc_logic = 0;’设置 LTC 触发方式为下降沿触发
Myltc_mode = 0; ’保留参数
Myfilter = 0;   ’保留参数
Rt200x_set_ltc_mode (MyCardNo,Myaxis,Myltc_logic,Myltc_mode,Myfilter);’设置 0 号轴的 LTC 信号，触发方式为下降沿触发
Myall_enable = 0;   ’设置锁存方式为单次锁存
Mylatch_source = 0; ’设置锁存源为指令位置
Mytriger_chunnel = 0;   ’保留参数
Rt200x_set_latch_mode(MyCardNo,Myaxis,Myall_enable,Mylatch_source,Mytriger_chunnel);’设置 0 号轴的锁存源是指令位置，单次锁存
Rt200x_reset_latch_flag (MyCardNo,Myaxis);  ’复位 0 号轴的锁存状态
Rt200x_set_profile (MyCardNo,Myaxis,1000,5000,0.1,0.1,2000);’设置梯形速度曲线参数
Rt200x_pmove (MyCardNo,Myaxis,50000,0); ’0号轴定长运动，运动距离为50000pulse、相对模式
While (Rt200x_get_latch_flag(MyCardNo,Myaxis) == 0) ’判断0号轴LTC锁存状态
    Application.DoEvent();
My_latch_Value = Rt200x_get_latch_value(MyCardNo,Myaxis);’从控制卡内读取编码器锁存器的值，并赋值给变量My_latch_Value
```

……

3.5.13.2 高速连续位置锁存功能

RT2008 卡提供了高速连续位置锁存功能，位置锁存无触发延时时间，当捕获到位置锁存信号后立即锁存当前位置；但相邻两个锁存点之间的间隔时间应大

于 10ms。相关函数如表所示：

名称	功能	参考
Rt200x_set_ltc_mode	设置指定轴的LTC信号	5.16高速位置锁存函数
Rt200x_set_latch_mode	设置锁存方式	
Rt200x_get_latch_flag_extern	从PC缓存中读取锁存器已锁存个数	
Rt200x_get_latch_value_extern	按索引号读取PC缓冲区中已保存的锁存值	
Rt200x_reset_latch_flag	复位指定卡的锁存器的标志位	

在连续锁存，多次触发高速锁存口锁存所有的触发位置，超过1000次剔除最先的触发位置保存最新的触发位置。在每次锁存后，不需要调用函数清除锁存状态。但是在启动高速连续位置锁存之前，必须调用函数清除锁存状态。

例程：高速连续位置锁存

```
……
Int MyCardNo,Myaxis;
Long [1000] My_latch_Value; ’ 定义锁存值
Int My_latch_flag;      ’ 定义锁存个数
Int  i;
MyCardNo = 0;   ’ 卡号
Myaxis = 0; ’ 轴号
Rt200x_set_ltc_mode (MyCardNo,Myaxis,0,0,0);’ 设置 0 号卡 0 号轴的LTC 信号，下降沿触发
Rt200x_set_latch_mode( MyCardNo,Myaxis,2,0,0);’ 设置 0 号轴的锁存源是指令位置，连续锁存
Rt200x_reset_latch_flag(MyCardNo,Myaxis);  ’ 复位 0 号轴的锁存状态
Rt200x_set_profile (MyCardNo,Myaxis,1000,5000,0.1,0.1,2000);’ 设置梯形速度曲线参数
Rt200x_pmove (MyCardNo,Myaxis,50000,0); ’ 0号轴定长运动，距离为50000pulse、相对模式
While(Rt200x_check_done(MyCardNo,Myaxis) == 0)  ’判断0号轴状态
    Application.DoEvent();
My_latch_flag = Rt200x_get_latch_flag_extern(MyCardNo,Myaxis);’从 PC 缓存中读取锁存器已锁存个数，并赋值给变量 My_latch_flag
For (i = 0 ;i< My_latch_flag;i++)
    My_latch_Value[i] = Rt200x_get_latch_value_extern(MyCardNo,Myaxis,i); ’ 按索引号读取PC 缓冲区中已保存的锁存值，并赋值给变量 My_latch_Value
```

……

3.5.14. 软着陆的实现

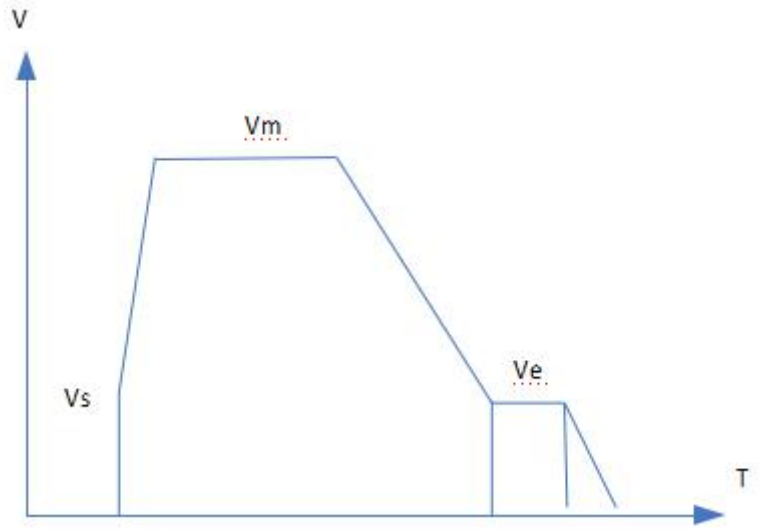


图 3-21 软着陆示意图

RT2008 运动控制卡支持软着陆功能。如上图所示，重点想实现的功能为 pmove 结束时能以一个很低速度靠近结束点。将 pmove 分为两段，第一段以 vs,vm，ve 进行规划，第二段以最大速度 ve 进行规划。同样，强制变位置 Rt200x_update_target_position_extern 同样有该项功能。相关函数如表所示：

名称	功能	参考
Rt200x_pmove_extern	实现profile，pmove整合，缩短指令时间	5.20软着陆函数
Rt200x_t_pmove_extern	实现profile，pmove整合，缩短指令时间，并且实现软着陆	
Rt200x_update_target_position_extern	强行改变指定轴的当前目标位置并且实现软着陆	

例程：软着陆运动

……

```
Double start,speed,stop, EB,dec,dis,softdist;
ushort cardID,axis ,posi_mode;
cardID = 0;
axis = 0;    ’轴号
```

```
dis =1000;  ’定长运动距离
softdist =40;   ’软着陆时低速运动距离
start = 10; ’启动速度
speed = 500;’运行速度
stop = 10;  ’停止速度，软着陆低速运动时速度
EB = 0.1;   ’加速时间
dec = 0.1;  ’减速时间
posi_mode = 1;  ’绝对运动模式
Rt200x_t_pmove_extern(cardID ,axis, dis, dis + softdist, start, speed, stop, EB, dec, posi_mode) ’执行运动
……
```

例程：强制变位软着陆运动

```
……
Double start,speed,stop, EB,dec,dis,softdist;
ushort cardID,axis ,posi_mode;
cardID = 0;
axis = 0;   ’轴号
dis =1000;  ’定长运动距离
softdist =40;   ’软着陆时低速运动距离
start = 10; ’启动速度
speed = 500;’运行速度
stop = 10;  ’停止速度，软着陆低速运动时速度
EB = 0.1;   ’加速时间
dec = 0.1;  ’减速时间
posi_mode = 1;  ’绝对运动模式
Rt200x_t_pmove_extern(cardID ,axis, dis, dis + softdist, start, speed, stop, EB, dec, posi_mode) ’执行运动
……
```

例 3.33：强制变位软着陆运动

```
……
Double start,speed,stop, EB,dec,dis,softdist;
Ushort cardID,axis ,posi_mode;
Int32 pos;
cardID = 0;
axis = 0;   ’轴号
dis =1000;  ’定长运动距离
softdist =40;   ’软着陆时低速运动距离
start = 10; ’启动速度
speed = 500;’运行速度
```

```
    stop = 10;  ’停止速度，软着陆低速运动时速度
    EB = 0.1;   ’加速时间
    dec = 0.1;  ’减速时间
    posi_mode = 1;  ’绝对运动模式
    s_para = 0; ’平滑时间
    RT200x_pmove_extern(cardID ,axis,dis,start,speed,stop,EB,dec,s_para,posi_mode); ’
执行运动
    While (true)
    {
        pos = Rt2004_get_position(cardID ,axis);
        If (pos > dis -50)
            Break;
        Application.DoEvent();
    }
    Rt200x_update_target_position_extern(cardID ,axis, dis + 500, dis + 500+ softdist,
0, 0)    ’强制变位并且实现软着陆

    ……
```

第 4 章 软件调试

4.1. 概述

RT2008 控制卡 Motion 软件是基于 Windows 平台开发的控制卡调试工具软件，其具有调试控制卡所有功能的能力，操作方便快捷。

用户在使用 VB、VC 或其它高级语言编写应用程序之前，可利用 Motion 软件快速熟悉 RT2008 卡的硬件、软件功能，还可以方便快捷地测试电机、传感器、开关元件、平台等在执行各种动作时的性能特点。

4.2. 功能描述

控制卡 Motion 软件是一款控制卡辅助调试软件，适用于初次使用锐特控制卡入门用户或控制卡使用调试查错。主要功能如下（注意：以下功能与具体型号控制卡有关，不同类型控制卡可能功能有所不同）：

*参数设置，支持各个轴所有参数的设置，包括脉冲模式、脉冲当量、加减速时间、回零模式、限位、伺服等参数设置。

*运动状态，支持控制卡的轴状态、IO 状态、轴专用信号监视。

*单轴测试，支持单个轴定长运动、定速运动、回零运动测试。

*多轴测试，支持直线插补、多类型圆弧/螺旋线单段插补运动测试。

*手轮测试，支持手轮倍率、通道、模式等相关参数设置，启动、停止等相关操作。

*PVT 测试，支持 PVT 模式、PVT 数据等设置，以及 PVT 理论曲线显示等，支持多轴 PVT 同时操作。

*单轴比较，支持比较点编辑、比较器状态监视等功能

*二维比较，支持二维比较点编辑、比较器状态监视等功能

*原点锁存，支持原点锁存模式、锁存源、触发方式等参数设置，配置、复位锁存器等操作。

*高速锁存，支持锁存方式、锁存源、触发方式等参数设置，配置、复位锁存器等相关操作。

4.3. 启动与 I/O 检测

4.3.1. 启动

Motion 启动之后会自动扫描电脑中的板卡，当选择板卡时界面上有三个操作按钮。如图 4-1 所示：

图 4-1motion 启动界面

其功能如下：

I/O 检测：单击该按钮，进入 I/O 检测界面，可以进行各专用输入口状态和通用输入口状态的检测，输出口测试；

运动测试：单击该按钮，进入运动控制操作界面，可以进行各种运动的演示，如：回单轴运动、多轴运动、手轮测试、PVT 测试、一维比较、二维比较、原点锁存、高速锁存等；

参数配置：单击改按钮，进入参数配置界面，该界面能够配置控制卡的一些参数，能够从控制卡中下载参数和上传参数至控制卡，同时还支持参数的一键导入和导出。

4.3.2. I/O 检测

进入 IO 检测界面，如图 4-2 所示，可以查看运动控制卡各轴的专用输入信号状态、通用输入信号状态和通用输出信号状态，并可控制每个通用输出信号的电平。

*专用输入信号电平：检测每个轴的专用信号状态；

*通用输入信号电平：检测通用输入信号的电平状态；

*通用输出信号控制和电平：控制通用输出信号，并显示当前各输出口的电平状态。通过单击显示为“第 X 位”的按钮，可以改变相应位的电平状态。绿色表示该信号的电平为低电平，红色表示该信号的电平为高电平。

*SEVON_and_ERC:SEVON 为控制卡控制私服启动器时提供的 SEVON 信号，通过点击显示为第 X 轴的按钮，能够控制 SEVON 的电平状态；ERC 为伺服误差清除信号，同样通过点击按钮，能够控制该信号的电平状态。

图 4-2 I/O 检测界面

4.4. 运动测试

进入运动测试操作界面，可以选择希望演示的运动功能，并对该运动的各种参数进行设置。

4.4.1. 单轴运动

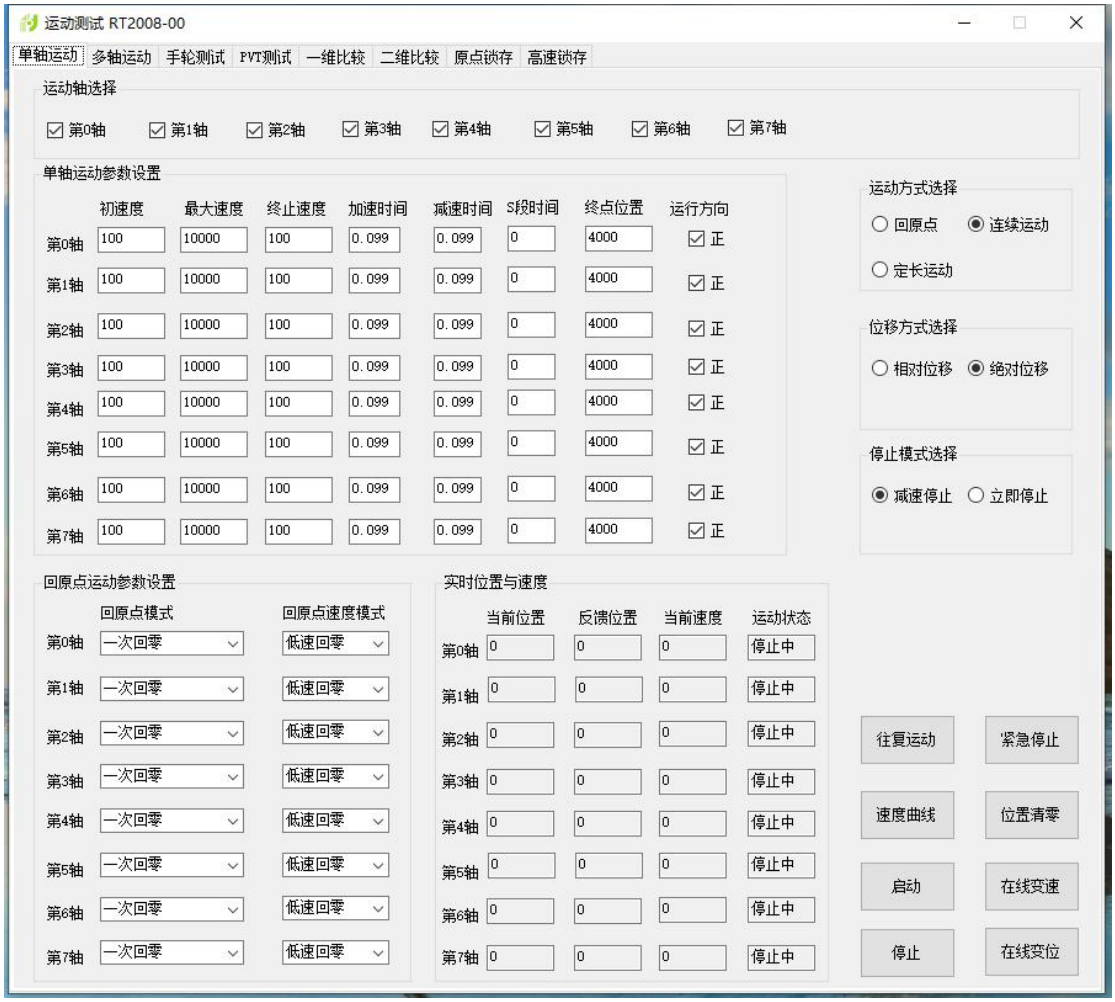


图 4-3 单轴运动测试界面

单轴运动操作界面如上图 4-3 所示，操作界面介绍如下：

*运动轴选择：选择需要运动控制的轴号；

*运动方式选择：选择需要演示的运动类型；

*位移方式选择：选择运动的位移方式；

*停止模式选择：选择运动停止时的停止模式；

*单轴运动参数设置：可以对演示运动的每个轴的参数进行设置，在运动启动时参数生效。

*回原点运动参数设置：可以针对回原点运动设置回原点模式和回原点速度模式，设置在回原点启动时生效。

*实时位置与速度：能够实时检测控制卡的八个轴的实时位置与速度还有运动状态等信息。

各按钮功能如下：

*速度曲线：单击该按钮能够弹出速度曲线计算界面，能够根据脉冲数还有运动速度计算出运动的速度曲线。

*启动：按照当前的设置进行运动。

*停止：根据设置的停止模式停止运动。

*紧急停止：紧急停止运动控制卡的所有运动。

*位置清零：清除选择轴的位置信息。

*在线变速：在运动过程中点击该按钮会根据单轴运动参数设置中的最大速度对当前运动进行变速。

*在线变位：在定长运动下点击该按钮能够根据单轴运动参数设置中的终点位置来对该运动的终点位置进行改变。

4.4.2. 多轴运动

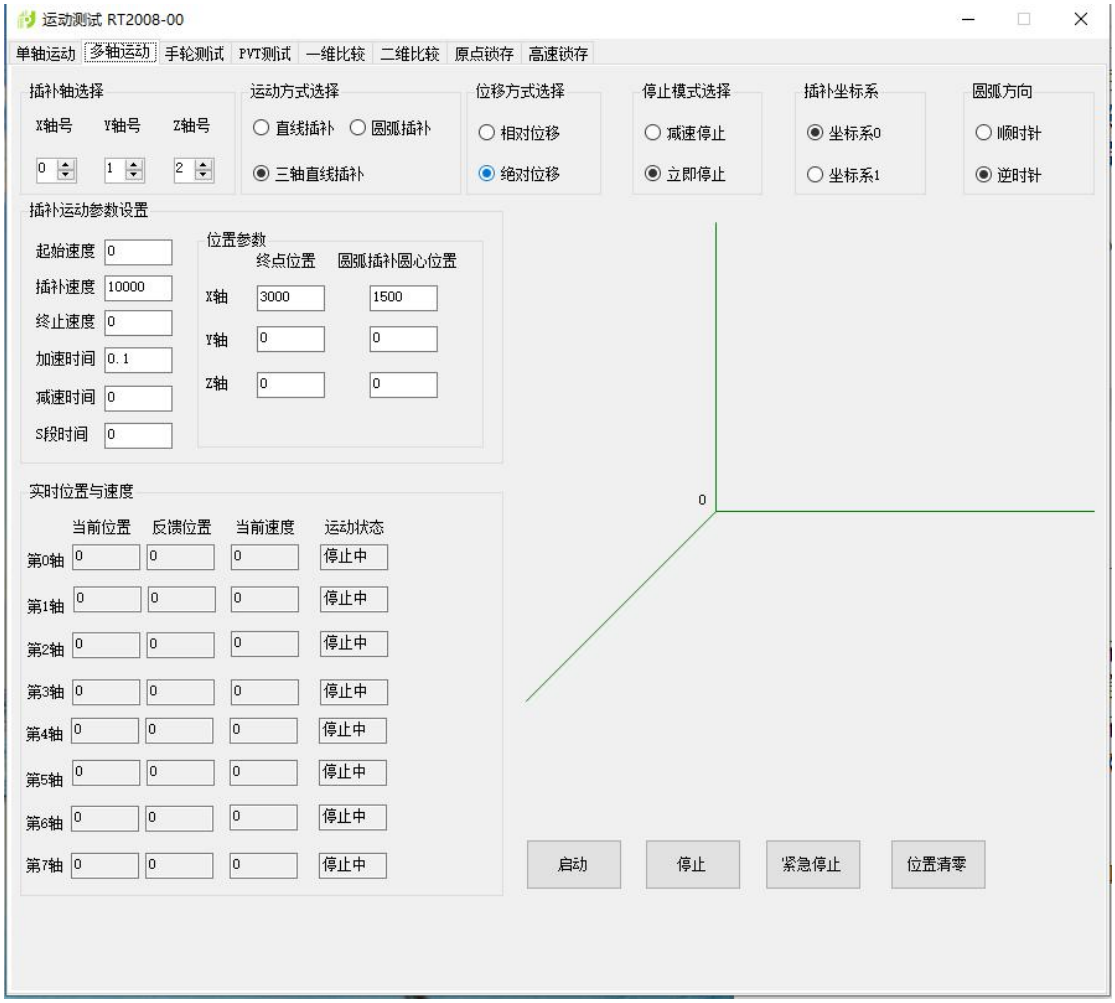


图 4-4 多轴运动测试界面

多轴运动操作界面如上图 4-4 所示。操作界面介绍如下：

*插补轴选择：选择需要插补的轴号，圆弧插补只支持 X 轴号和 Y 轴号两轴插补；

*运动方式选择：选择需要演示的运动类型；

*位移方式选择：选择运动的位移方式；

*停止模式选择：选择运动停止时的停止模式；

*插补坐标系选择：选择插补运动的坐标系；

*圆弧方向选择：选择在圆弧插补运动中的圆弧方向；

*插补运动参数设置：可以对插补运动的参数进行设置，在运动启动时参数生效。

*位置参数设置：可以对插补轴的终点位置进行设置，还可以设置圆弧插补

的圆心位置。

*实时位置与速度：能够实时检测控制卡的八个轴的实时位置与速度还有运动状态等信息。

各按钮功能如下：

*启动：按照当前的设置进行多轴插补运动。

*停止：根据设置的停止模式停止插补运动。

*紧急停止：紧急停止运动控制卡的所有运动。

*位置清零：清除选择轴的位置信息。

4.4.3. 手轮测试

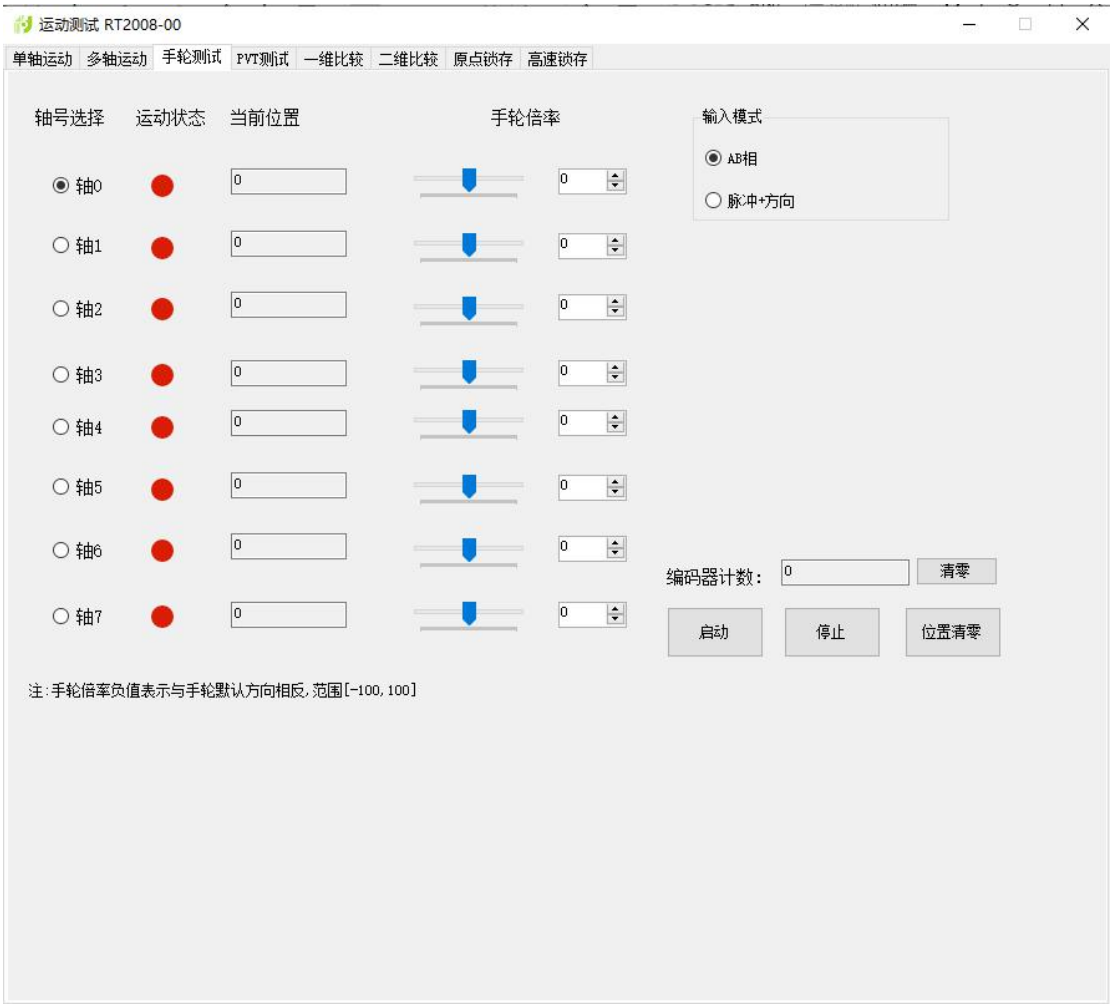


图 4-5 手轮测试界面

手轮测试操作界面如上图 4-5 所示。操作界面介绍如下：

*轴选择：选择需要运动的轴号，手轮只支持选择单轴运动；

*手轮倍率设置：可以设置单轴的手轮倍率。手轮倍率负值表示与手轮默认方向相反，范围[-100,100]。

*输入模式：可以选择手轮的输入模式。

*编码器计数：可以实时显示电机的编码器位置。

*运动状态：可以显示轴的运动状态，停止时为红色，当运动时显示为绿色。

各按钮功能如下：

*启动：按照当前的设置启动手轮运动。

*停止：停止手轮运动。

*位置清零：清除选择轴的位置信息。

4.4.4. PVT 测试

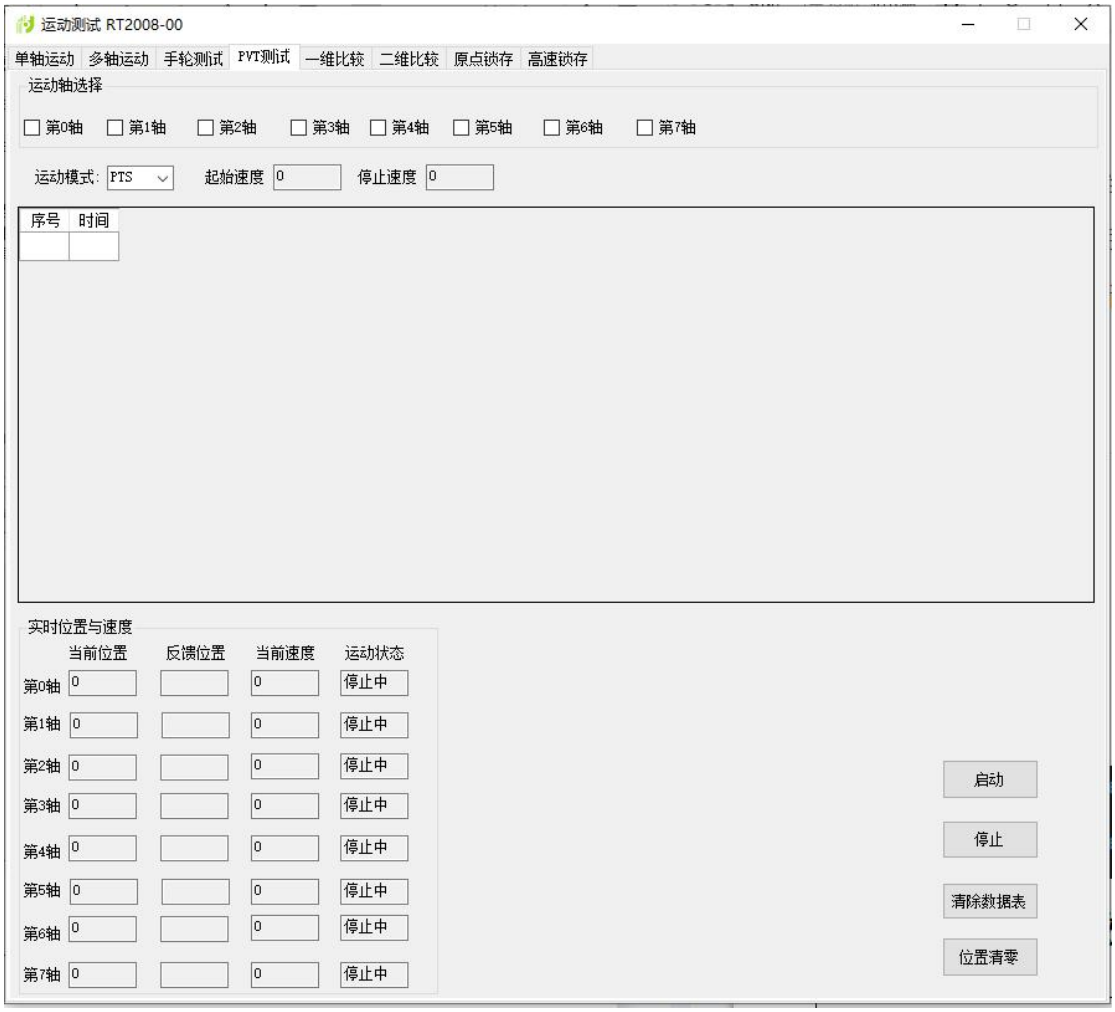


图 4-6 PVT 测试操作界面

PVT 测试操作界面如上图 4-6 所示。操作界面介绍如下：

*运动轴选择：选择需要运动的轴号；

*运动模式选择：可以选择 PVT 测试的运动模式，根据选择的不同轴号和不同运动模式，表格里的参数会随之改变。

*起始速度设置：在 PVTS 模式下需要设置。

*停止速度设置：在 PVTS 模式下需要设置。

*实时位置与速度：能够实时检测控制卡的八个轴的实时位置与速度还有运动状态等信息。

各按钮功能如下：

*启动：按照选择的 PVT 模式和数据表中输入的参数进行 PVT 运动。

*停止：停止当前的运动。

*清除数据表：点击此按钮能够清除 PVT 测试的数据表。

*位置清零：清除选择轴的位置信息。

4.4.5. 一维比较

图 4-7 一维比较操作界面

一维比较操作界面如上图 4-7 所示。操作界面介绍如下：

*参数设置：能够设置一维比较运动的参数；

*比较点的添加：通过在表格中右击鼠标能够弹出添加比较点的选项。如下图 4-8 所示。

*设置比较点的功能和参数添加到运动表格：如下图 4-9 所示。

各按钮功能如下：

*启动：启动根据表格中添加的比较点的运动，如果表格中没有比较点，将无事发生。

*停止：停止当前的运动。

*位置清零：清除选择轴的位置信息。

*比较器清零：清除运动控制卡中比较器的内容。

图 4-8 比较点添加界面

图 4-9 比较点设置界面

4.4.6. 二维比较

图 4-10 二维比较操作界面

二维比较同一维比较的区别是能够选择两个轴进行同时比较。二维比较操作界面如上图 4-10 所示。操作界面介绍如下：

*参数设置：能够设置一维比较运动的参数；

*比较点的添加：通过在表格中右击鼠标能够弹出添加比较点的选项。如上图 4-10 所示。

*设置比较点的功能和参数添加到运动表格：如下图 4-11 所示。

各按钮功能如下：

*启动：启动根据表格中添加的比较点的运动，如果表格中没有比较点，将无事发生。

*停止：停止当前的运动。

*位置清零：清除选择轴的位置信息。

*比较器清零：清除运动控制卡中比较器的内容。

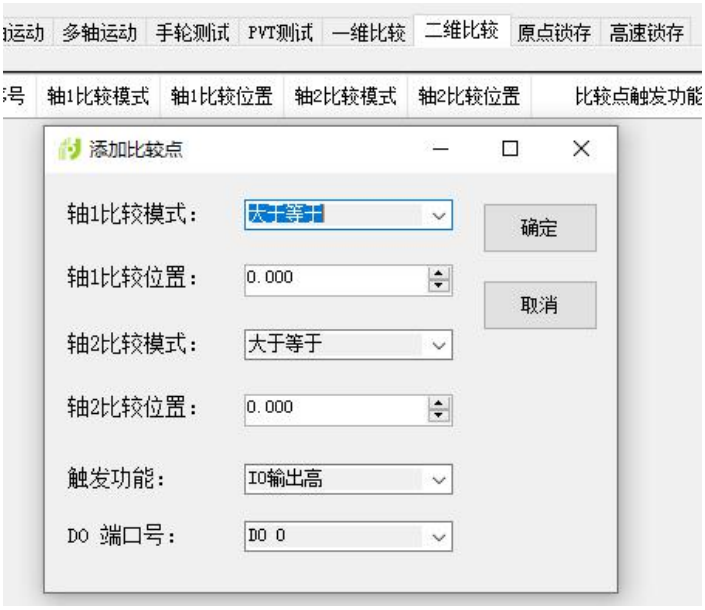


图 4-11 二维比较点添加界面

4.4.7. 原点锁存

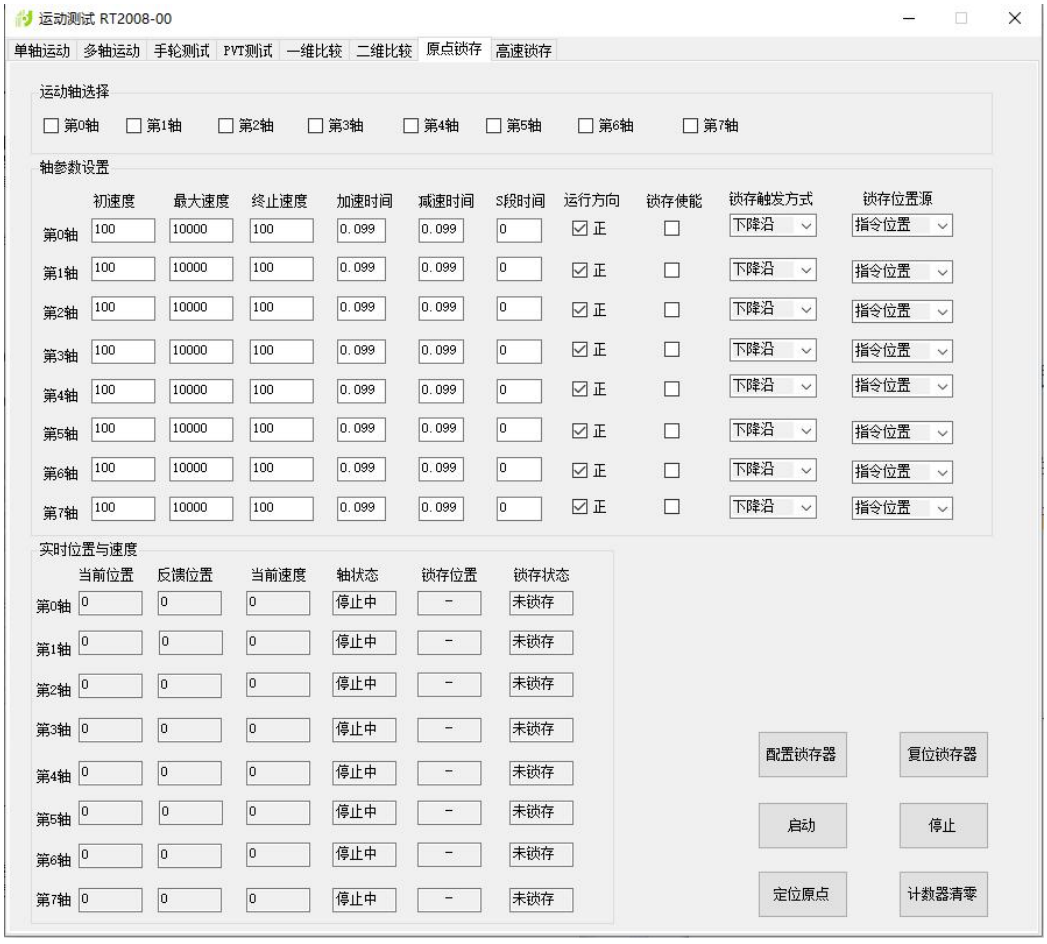


图 4-12 原点锁存测试界面

原点锁存测试界面如上图 4-12 所示。操作界面介绍如下：

*运动轴选择：选择需要原点锁存的轴号；

*轴参数设置：可以设置每个轴的运动参数，同时还可以配置原点锁存相关参数。

*实时位置与速度：能够实时检测控制卡的八个轴的实时位置与速度还有运动状态等信息，当原点锁存触发后，还将显示锁存位置和锁存状态。

各按钮功能如下：

*配置锁存器：按照轴参数设置的参数对控制卡原点锁存器进行配置，当锁存使能框打上勾时，该轴的原点锁存才有效。

*复位锁存器：对运动控制卡的锁存器进行复位，同时锁存位置和锁存状态也清除。

*启动：根据轴参数设置的参数启动运动，当遇到原点信号时，将会锁存原点位置。

*停止：停止当前选择轴的运动。

*定位原点：当选择轴成功锁存到原点位置时，单击此按钮能够将该轴的当前位置运动到锁存到的原点位置上。

*计数器清零：清除轴的位置信息。

4.4.8. 高速锁存

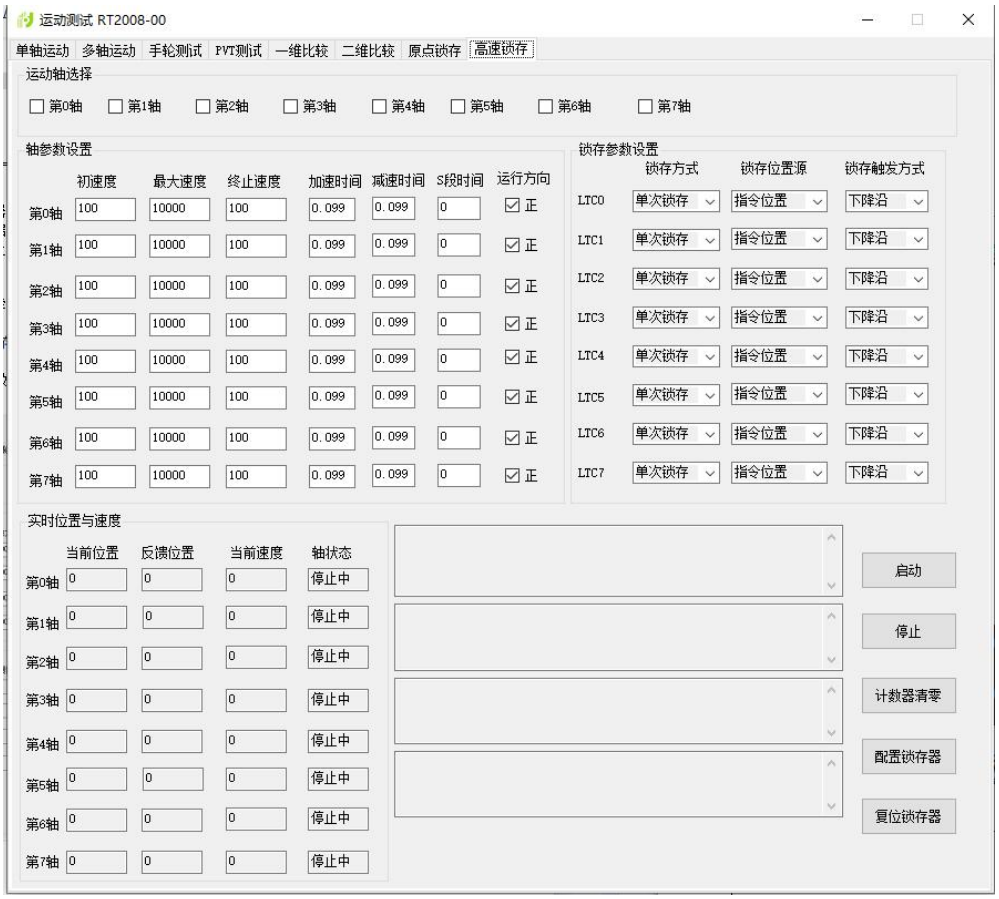


图 4-13 高速锁存测试界面

高速锁存测试界面如上图 4-13 所示。操作界面介绍如下：

*运动轴选择：选择需要高速锁存的轴号；

*轴参数设置：可以设置每个轴的运动参数。

*锁存参数设置：可以单独设置高速锁存八个锁存输入的锁存方式、锁存位置源与锁存触发方式等参数。

*实时位置与速度：能够实时检测控制卡的八个轴的实时位置与速度还有运动状态等信息。

各按钮功能如下：

*启动：根据轴参数设置的参数启动运动，当遇到锁存信号时，将会在界面输出锁存信息。

*停止：停止当前选择轴的运动。

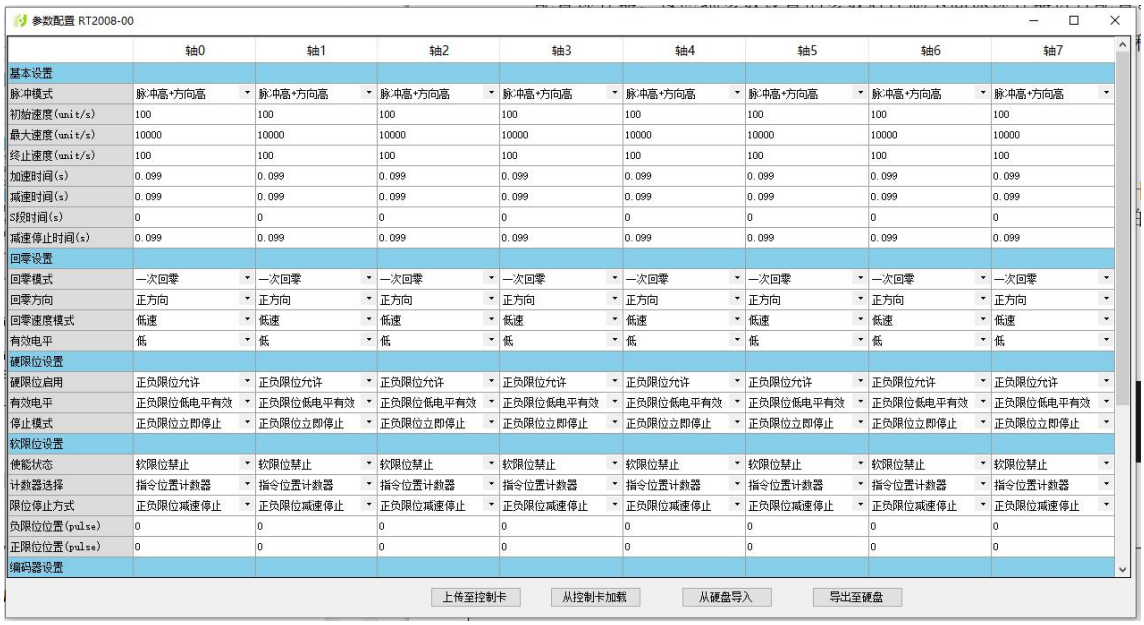
*计数器清零：清除轴的位置信息。

*配置锁存器：按照轴参数设置的参数对控制卡高速锁存器进行配置。

*复位锁存器：对运动控制卡的高速锁存器进行复位，同时锁存位置和锁存状态也清除。

4.5. 参数配置

进入参数配置操作界面，如下图 4-14 所示，可以在该界面配置控制卡的一些参数，能够从控制卡中下载参数和上传参数至控制卡，同时还支持参数的一键导入和导出。



	轴0	轴1	轴2	轴3	轴4	轴5	轴6	轴7
基本设置								
脉冲模式	脉冲高+方向高	脉冲高+方向高	脉冲高+方向高	脉冲高+方向高	脉冲高+方向高	脉冲高+方向高	脉冲高+方向高	脉冲高+方向高
初始速度(unit/s)	100	100	100	100	100	100	100	100
最大速度(unit/s)	10000	10000	10000	10000	10000	10000	10000	10000
终止速度(unit/s)	100	100	100	100	100	100	100	100
加速时间(s)	0.099	0.099	0.099	0.099	0.099	0.099	0.099	0.099
减速时间(s)	0.099	0.099	0.099	0.099	0.099	0.099	0.099	0.099
S段时间(s)	0	0	0	0	0	0	0	0
减速停止时间(s)	0.099	0.099	0.099	0.099	0.099	0.099	0.099	0.099
回零设置								
回零模式	一次回零	一次回零	一次回零	一次回零	一次回零	一次回零	一次回零	一次回零
回零方向	正方向	正方向	正方向	正方向	正方向	正方向	正方向	正方向
回零速度模式	低速	低速	低速	低速	低速	低速	低速	低速
有效电平	低	低	低	低	低	低	低	低
硬限位设置								
硬限位启用	正负限位允许	正负限位允许	正负限位允许	正负限位允许	正负限位允许	正负限位允许	正负限位允许	正负限位允许
有效电平	正负限位低电平有效	正负限位低电平有效	正负限位低电平有效	正负限位低电平有效	正负限位低电平有效	正负限位低电平有效	正负限位低电平有效	正负限位低电平有效
停止模式	正负限位立即停止	正负限位立即停止	正负限位立即停止	正负限位立即停止	正负限位立即停止	正负限位立即停止	正负限位立即停止	正负限位立即停止
软限位设置								
使能状态	软限位禁止	软限位禁止	软限位禁止	软限位禁止	软限位禁止	软限位禁止	软限位禁止	软限位禁止
计数器选择	指令位置计数器	指令位置计数器	指令位置计数器	指令位置计数器	指令位置计数器	指令位置计数器	指令位置计数器	指令位置计数器
限位停止方式	正负限位减速停止	正负限位减速停止	正负限位减速停止	正负限位减速停止	正负限位减速停止	正负限位减速停止	正负限位减速停止	正负限位减速停止
负限位位置(pulse)	0	0	0	0	0	0	0	0
正限位位置(pulse)	0	0	0	0	0	0	0	0
编码器设置								

图 4-14 参数配置界面

当多张卡同时工作时，可以通过初始时的板卡选择来选择需要操作的卡。在参数配置完成后，需要将配置上传至控制卡才能使配置的参数真正生效。

第 5 章 函数库详解

```
typedef unsigned long DWORD;
typedef unsigned short WORD;
```

5.1. 板卡设置函数

short Rt200x_board_init(void)	
控制卡初始化函数，分配系统资源	
参数	说明
无	无
返回值	0：没有找到控制卡，或者控制卡异常 1~8：控制卡数 负值：表明有2张或2张以上控制卡的硬件设置卡号相同；返回值取绝对值后减1即为该卡号

short Rt200x_board_reset(void)	
控制卡硬件复位函数	
参数	说明
无	无
返回值	错误代码
注意	执行复位操作后，必须等待5秒方可执行初始化控制卡，否则会出错。出错后必须重新执行复位操作，再等待5秒后执行初始化控制卡

short Rt200x_board_close(void)	
控制卡关闭函数，释放系统资源	
参数	说明
无	无
返回值	错误代码

short Rt200x_get_CardInfList (WORD* CardNun, DWORD* CardTypeList, WORD* CardIdList)	
获取控制卡硬件 ID 号	
参数	说明
CardNun	返回初始化成功的卡数
CardTypeList	返回控制卡固件类型数组
CardIdList	返回控制卡硬件ID号数组，卡号按从小到大顺序排列
返回值	错误代码
注意	参数CardTypeList类型为十六进制

short Rt200x_get_card_version(WORD CardNo,DWORD *CardVersion)	
获取控制卡硬件版本号	
参数	说明
CardNo	控制卡卡号
CardVersion	返回控制卡硬件版本号

返回值	错误代码

short Rt200x_get_card_soft_version(WORD CardNo,DWORD *FirmID,DWORD *SubFirmID)	
获取控制卡固件版本号	
参数	说明
CardNo	控制卡卡号
FirmID	返回控制卡固件类型
SubFirmID	返回控制卡固件版本号
返回值	错误代码
注　意	参数FirmID类型为十六进制

short Rt200x_get_card_lib_version(DWORD *LibVer)	
获取控制卡动态库文件版本号	
参数	说明
LibVer	返回库版本号
返回值	错误代码

short Rt200x_get_total_axes(WORD CardNo,DWORD *TotalAxis)	
获取当前卡的轴数	
参数	说明

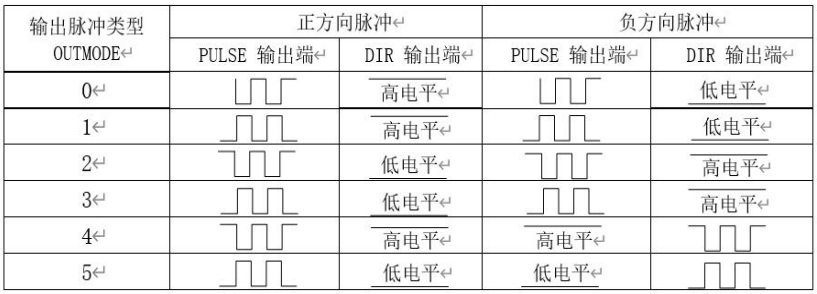
CardNo	控制卡卡号
TotalAxis	返回当前卡的轴数
返回值	错误代码

short Rt200x_download_configfile(WORD CardNo,const char *FileName)	
下载参数文件	
参数	说明
CardNo	控制卡卡号
FileName	文件路径: 参数文件名+后缀：相对路径 完整描述参数文件的路径+文件名后缀：绝对路径
返回值	错误代码
注　意	1)当使用相对路径时，参数文件与程序必须在同一目录下

short Rt200x_download_firmware(WORD CardNo,const char *FileName)	
下载固件文件	
参数	说明
CardNo	控制卡卡号
FileName	文件路径: 参数文件名+后缀：相对路径 完整描述参数文件的路径+文件名后缀：绝对路径

返回值	错误代码
注　意	1)当使用相对路径时，参数文件与程序必须在同一目录下

5.2. 脉冲模式设置函数

short Rt200x_set_pulse_outmode(WORD CardNo,WORD axis,WORD outmode)	
设置指定轴的脉冲输出模式	
参数	说明
CardNo	控制卡卡号
axis	指定轴号，取值范围：0~7
outmode	脉冲输出方式选择 
返回值	错误代码
注　意	在调用运动函数（如：Rt200x_vmove等）输出脉冲之前，一定要根据驱动器接收脉冲的模式调用Rt200x_set_pulse_outmode设置控制卡脉冲输出模式

short Rt200x_get_pulse_outmode(WORD CardNo,WORD axis,WORD* outmode)
读取指定轴的脉冲输出模式设置

参数	说明
CardNo	控制卡卡号
axis	指定轴号，取值范围：0~7
outmode	返回脉冲输出方式
返回值	错误代码

5.3. 回原点运动函数

short Rt200x_set_home_el_return(WORD CardNo,WORD axis,WORD enable)	
设置回原点时遇限位开关是否自动反找	
参数	说明
CardNo	控制卡卡号
axis	指定轴号，取值范围：0~7
enable	使能是否遇限位反找，0：不使能，1：使能
返回值	错误代码
注意	默认回原点时遇限位开关不会自动反找原点，每次初始化时配置一次即可

short Rt200x_set_home_pin_logic(WORD CardNo,WORD axis,WORD org_logic,double filter)	
设置ORG原点信号	
参数	说明
CardNo	控制卡卡号

axis	指定轴号，取值范围：0~7
org_logic	ORG 信号有效电平，0：低有效，1：高有效
Filter	保留参数，固定值为0
返回值	错误代码

short Rt200x_get_home_pin_logic(WORD CardNo,WORD axis,WORD *org_logic,double *filter)	
读取ORG原点信号设置	
参数	说明
CardNo	控制卡卡号
axis	指定轴号，取值范围：0~7
org_logic	返回设置的ORG信号有效电平保留参数
Filter	保留参数，固定值为0
返回值	错误代码

short Rt200x_set_homemode(WORD CardNo, WORD axis, WORD home_dir, double vel_mode, WORD mode, WORD EZ_count)	
设置回原点模式	
参数	说明
CardNo	控制卡卡号
axis	指定轴号，取值范围：0~7

home_dir	回零方向，0：负向，1：正向
vel_mode	回零速度模式： 0：低速回零，即以本指令前面的Rt200x_set_profile函数设置的起始速度运行 1：高速回零，即以本指令前面的Rt200x_set_profile函数设置的最大速度运行
mode	回零模式： 0：一次回零 1：一次回零加回找 2：二次回零 3：一次回零后再记一个EZ脉冲进行回零 4：记一个EZ脉冲进行回零 5：原点加反向EZ 6：原点捕获回零 7：原点锁存加同向EZ锁存 8：单独记一个EZ锁存 9：原点锁存加反向EZ锁存
EZ_count	保留参数，固定值为0
返回值	错误代码
注意	当回零模式mode=4时，回零速度模式将固定为低速回零

short Rt200x_get_homemode(WORD CardNo,WORD axis,WORD* home_dir, double* vel, WORD* mode, WORD* EZ_count)	
读取回原点模式	
参数	说明

CardNo	控制卡卡号
axis	指定轴号，取值范围：0~7
home_dir	返回回零方向
Vel	返回回零速度模式
Mode	返回回零模式
EZ_count	保留参数
返回值	错误代码

short Rt200x_home_move(WORD CardNo,WORD axis)	
回原点运动	
参数	说明
CardNo	控制卡卡号
axis	指定轴号，取值范围：0~7
返回值	错误代码

5.4. 限位开关设置函数

short Rt200x_set_el_mode(WORD CardNo,WORD axis,WORD el_enable,WORD el_logic,WORD el_mode)	
设置EL限位信号	
参数	说明

CardNo	控制卡卡号
axis	指定轴号，取值范围：0~7
el_enable	EL信号的使能状态： 0：正负限位禁止 1：正负限位允许
el_logic	EL信号的有效电平： 0：正负限位低电平有效 1：正负限位高电平有效
el_mode	EL制动方式： 0：正负限位立即停止 1：正负限位减速停止
返回值	错误代码

short Rt200x_get_el_mode(WORD CardNo,WORD axis, WORD *el_enable,WORD* el_logic,WORD* el_mode)	
读取EL限位信号设置	
参数	说明
CardNo	控制卡卡号
axis	指定轴号，取值范围：0~7
el_enable	返回设置的EL信号使能状态
el_logic	返回设置的EL信号有效电平
el_mode	返回EL制动方式

返回值	错误代码

short Rt200x_set_softlimit(WORD CardNo,WORD axis,WORD enable,WORD source_sel,WORD SL_action, long N_limit,long P_limit)	
设置软限位	
参数	说明
CardNo	控制卡卡号
axis	指定轴号，取值范围：0~7
enable	使能状态，0：禁止，1：允许
source_sel	计数器选择，0：指令位置计数器，1：编码器计数器
SL_action	限位停止方式，0：减速停止，1：立即停止
N_limit	负限位位置，单位：pulse
P_limit	正限位位置，单位：pulse
返回值	错误代码
注意	正、负限位位置可为正数也可为负数，但正限位位置应大于负限位位置

short Rt200x_get_softlimit(WORD CardNo,WORD axis,WORD* enable, WORD* source_sel,WORD* SL_action, long* N_limit,long* P_limit)	
读取软限位设置	
参数	说明
CardNo	控制卡卡号

axis	指定轴号，取值范围：0~7
enable	返回使能状态
source_sel	返回计数器选择
SL_action	返回限位停止方式
N_limit	返回负限位脉冲数
P_limit	返回正限位脉冲数
返回值	错误代码

5.5. 位置计数器控制函数

short Rt200x_set_position(WORD CardNo,WORD axis,long current_position)	
设置指令脉冲位置	
参数	说明
CardNo	控制卡卡号
axis	指定轴号，取值范围：0~7
current_position	指令脉冲位置，单位：pulse
返回值	错误代码

long Rt200x_get_position(WORD CardNo,WORD axis)	
读取指令脉冲位置	
参数	说明

CardNo	控制卡卡号
axis	指定轴号，取值范围：0~7
返回值	指令脉冲位置，单位：pulse

5.6. 运动状态检测及控制相关函数

Double Rt200x_read_current_speed(WORD CardNo,WORD axis)	
读取当前速度值	
参数	说明
CardNo	控制卡卡号
axis	指定轴号，取值范围：0~7
返回值	指定轴的速度，单位：pulse/s
注意	当执行直线插补运动时，该函数读取的速度为矢量速度；当执行圆弧插补运动时，该函数读取的速度为各轴分量速度

short Rt200x_check_done(WORD CardNo,WORD axis)	
检测指定轴的运动状态	
参数	说明
CardNo	控制卡卡号
axis	指定轴号，取值范围：0~7
返回值	0：指定轴正在运行，1：指定轴已停止

注意	此函数适用于单轴、PVT运动

short Rt200x_check_done_multicoor(WORD CardNo,WORD Crd)	
检测坐标系的运动状态	
参数	说明
CardNo	控制卡卡号
Crd	指定控制卡上的坐标系号（取值范围：0~1）
返回值	坐标系状态，0：正在使用中，1：正常停止
注意	此函数适用于插补运动

DWORD Rt200x_axis_io_status(WORD CardNo,WORD axis)	
读取指定轴有关运动信号的状态	
参数	说明
CardNo	控制卡卡号
axis	指定轴号，取值范围：0~7
返回值	表 8.2　轴的运动信号状态↵ 位号↵ 信号名称↵ 描述↵ 0↵ ALM↵ 1：表示伺服报警信号 ALM 为 ON； 0：OFF↵ 1↵ EL+↵ 1：表示正硬限位信号 +EL 为 ON； 0：OFF↵ 2↵ EL-↵ 1：表示负硬限位信号 - EL 为 ON； 0：OFF↵ 3↵ EMG↵ 1：表示急停信号 EMG 为 ON； 0：OFF↵ 4↵ ORG↵ 1：表示原点信号 ORG 为 ON； 0：OFF↵ 6↵ SL+↵ 1：表示正软限位信号+SL 为 ON； 0：OFF↵ 7↵ SL-↵ 1：表示负软件限位信号-SL 为 ON； 0：OFF↵ 8↵ INP↵ 1：表示伺服到位信号INP 为 ON； 0：OFF↵ 9↵ EZ↵ 1：表示EZ 信号为ON； 0：OFF↵ 其他位↵ 保留↵ ↵

short Rt200x_stop(WORD CardNo,WORD axis,WORD stop_mode)	
指定轴停止运动	
参数	说明
CardNo	控制卡卡号
axis	指定轴号，取值范围：0~7
stop_mode	制动方式，0：减速停止，1：紧急停止
返回值	错误代码
注意	此函数适用于单轴、PVT运动

short Rt200x_stop_multicoor(WORD CardNo,WORD Crd,WORD stop_mode)	
停止坐标系内所有轴的运动	
参数	说明
CardNo	控制卡卡号
Crd	指定控制卡上的坐标系号（取值范围：0~1）
stop_mode	制动方式，0：减速停止，1：立即停止
返回值	错误代码
注意	此函数适用于插补运动

short Rt200x_emg_stop(WORD CardNo)
紧急停止所有轴

参数	说明
CardNo	控制卡卡号
返回值	错误代码
注意	此函数适用于所有运动模式

long Rt200x_get_target_position(WORD CardNo,WORD axis)	
读取正在运动的目标位置（绝对坐标）	
参数	说明
CardNo	控制卡卡号
axis	指定轴号，取值范围：0~7
返回值	目标位置，单位：pulse

5.7. 单轴运动速度曲线设置函数

short Rt200x_set_profile(WORD CardNo,WORD axis,double Min_Vel,double Max_Vel,double TEB,double Tdec,double Stop_Vel)	
设置单轴运动速度曲线	
参数	说明
CardNo	控制卡卡号
axis	指定轴号，取值范围：0~7
Min_Vel	起始速度，单位：pulse/s(最大值为2M)

Max_Vel	最大速度，单位：pulse/s(最大值为2M)
TEB	加速时间，单位：s(最小值为0.001s)
Tdec	减速时间，单位：s(最小值为0.001s)
Stop_Vel	停止速度，单位：pulse/s(最大值为2M)
返回值	错误代码

short Rt200x_get_profile(WORD CardNo,WORD axis,double* Min_Vel,double* Max_Vel,double* TEB,double* Tdec,double* Stop_Vel)	
读取单轴运动速度曲线	
参数	说明
CardNo	控制卡卡号
axis	指定轴号，取值范围：0~7
Min_Vel	返回起始速度，单位：pulse/s
Max_Vel	返回最大速度，单位：pulse/s
TEB	返回加速时间，单位：s
Tdec	返回减速时间，单位：s
Stop_Vel	返回停止速度，单位：pulse/s
返回值	错误代码

short Rt200x_set_s_profile(WORD CardNo,WORD axis,WORD s_mode,double s_para)
设置单轴速度曲线S段参数值

参数	说明
CardNo	控制卡卡号
axis	指定轴号，取值范围：0~7
s_mode	保留参数，固定值为0
s_para	S段时间，单位：s；范围：0~0.5s
返回值	错误代码

short Rt200x_get_s_profile(WORD CardNo,WORD axis,WORD s_mode, double* s_para)	
读取单轴速度曲线S段参数值	
参数	说明
CardNo	控制卡卡号
axis	指定轴号，取值范围：0~7
s_mode	保留参数，固定值为0
s_para	返回设置的S段时间
返回值	错误代码

5.8. 单轴运动函数

short Rt200x_pmove(WORD CardNo,WORD axis,long Dist,WORD posi_mode)	
指定轴点位运动	
参数	说明

CardNo	控制卡卡号
axis	指定轴号，取值范围：0~7
Dist	目标位置，单位：pulse
posi_mode	运动模式，0：相对坐标模式，1：绝对坐标模式
返回值	错误代码
注意	当运动模式为相对坐标模式时，目标位置大于0时正向运动，小于0时反向运动

short Rt200x_vmove(WORD CardNo,WORD axis,WORD dir)	
指定轴连续运动	
参数	说明
CardNo	控制卡卡号
axis	指定轴号，取值范围：0~7
dir	运动方向，0：负方向，1：正方向
返回值	错误代码

short Rt200x_change_speed(WORD CardNo,WORD axis,double Curr_Vel,double TEBdec)	
在线改变指定轴的当前运动速度	
参数	说明
CardNo	控制卡卡号
axis	指定轴号，取值范围：0~7

Curr_Vel	改变后的运动速度，单位：pulse/s
TEBdec	保留参数，固定值为0
返回值	错误代码
注意	该函数适用于单轴运动中的变速 变速一旦成立，该轴的默认运行速度将会被改写为Curr_Vel，也即当调用Rt200x_get_profile回读速度参数时会发生与Rt200x_set_profile所设置的不一致的现象 在连续运动中Curr_Vel负值表示往负向变速，正值表示往正向变速。在点位运动中Curr_Vel只允许正值

short Rt200x_reset_target_position(WORD CardNo,WORD axis,long dist,WORD posi_mode)	
在线改变指定轴的当前目标位置	
参数	说明
CardNo	控制卡卡号
axis	指定轴号，取值范围：0~7
dist	目标位置，单位：pulse
posi_mode	保留参数，固定值为0
返回值	错误代码
注意	该函数只适用于点位运动中的变位 参数dist为绝对坐标位置值，无论当前的运动模式为绝对坐标还是相对坐标模式

short Rt200x_update_target_position(WORD CardNo,WORD axis,long dist,WORD

posi_mode)	
强行改变指定轴的当前目标位置（在线/非在线）	
参数	说明
CardNo	控制卡卡号
axis	指定轴号，取值范围：0~7
dist	目标位置，单位：pulse
posi_mode	保留参数，固定值为0
返回值	错误代码
注意	该函数适用于指定轴停止状态或点位运动中的变位 参数dist为绝对坐标位置值，无论当前的运动模式为绝对坐标还是相对坐标模式

5.9. 插补速度曲线设置函数

short Rt200x_set_vector_profile_multicoor(WORD CardNo,WORD Crd, Double Min_Vel, double Max_Vel, double TEB , double Tdec, double Stop_Vel)	
设置插补速度	
参数	说明
CardNo	控制卡卡号
Crd	指定控制卡上的坐标系号（取值范围：0~1）
Min_Vel	保留参数，固定值为0
Max_Vel	合成最大速度，单位：pulse/s

TEB	加减速时间，单位：s（最小值为0.001s）
Tdec	保留参数，固定值为0
Stop_Vel	保留参数，固定值为0
返回值	错误代码
说明	Rt200x卡支持两个插补系（参数Crd）。两个插补系的速度可独立设置，执行插补运动时两个插补系可独立进行插补运动（即可同时进行两组插补运动）

short Rt200x_get_vector_profile_multicoor(WORD CardNo, WORD Crd, Double *Min_Vel, double *Max_Vel, double *TEB , double *Tdec, double *Stop_Vel)	
读取插补速度设置	
参数	说明
CardNo	控制卡卡号
Crd	指定控制卡上的坐标系号（取值范围：0~1）
Min_Vel	保留参数，固定值为0
Max_Vel	返回合成最大速度，单位：pulse/s
TEB	返回加减速时间，单位：s
Tdec	保留参数，固定值为0
Stop_Vel	保留参数，固定值为0
返回值	错误代码

5.10. 插补运动函数

short Rt200x_line_multicoor(WORD CardNo, WORD Crd, WORD axisNum, WORD *axisList, long* DistList, WORD posi_mode)	
直线插补运动	
参数	说明
CardNo	控制卡卡号
Crd	指定控制卡上的坐标系号（取值范围：0~1）
axisNum	插补轴数，取值范围：2~4
axisList	插补轴列表
DistList	插补轴目标位置列表，单位：pulse
posi_mode	运动模式，0：相对坐标模式，1：绝对坐标模式
返回值	错误代码
注意	检测直线插补状态应使用坐标系状态检测函数 Rt200x_check_done_multicoor；停止正在执行的直线插补运动应使用坐标系停止函数Rt200x_stop_multicoor

short Rt200x_arc_move_multicoor (WORD CardNo, WORD Crd ,WORD *AxisList,long *Target_Pos, long *Cen_Pos, WORD Arc_Dir, WORD posi_mode)	
两轴圆弧插补运动，圆心位置+终点位置	
参数	说明
CardNo	控制卡卡号
Crd	指定控制卡上的坐标系号（取值范围：0~1）

AxisList	轴列表数组
Target_Pos	终点坐标，单位：pulse
Cen_Pos	圆心坐标，单位：pulse
Arc_Dir	圆弧方向，0：顺时针，1：逆时针
posi_mode	运动模式，0：相对坐标模式，1：绝对坐标模式
返回值	错误代码
注意	检测圆弧插补状态应使用坐标系状态检测函数 Rt200x_check_done_multicoor;停止正在执行的圆弧插补运动应使用坐标系停止函数Rt200x_stop_multicoor 圆弧插补设置的终点位置与理论终点位置的允许误差在+/-100个脉冲以内。以相对坐标模式为例，当圆心位置为（0,1000），终点理论位置为（0，2000）时，而终点位置被设置为（0,2100），该圆弧插补仍可正常运行

5.11. PVT 运动函数

short Rt200x_PttTable(WORD CardNo,WORD axis,DWORD count,double *pTime,long *pPos)	
向指定数据表传送数据，采用PTT模式	
参数	说明
CardNo	控制卡卡号
axis	指定轴号，取值范围：0~7
count	数据点个数，每个数据表具有1000个存储空间，每个数据点占用1个存储空间
pTime	数据点时间数组，单位：s（精度：ms）；数组长度：count

pPos	数据点位置数组，单位：pulse；数组长度：count
返回值	错误代码
注意	下载的第一组（即起始点）数据中位置、时间必须为0；数组中的数据都是以起始点的数据为参考点 调用该指令向数据表中传递数据时，会删除数据表中原先的数据，因此所有数据应当一次传送完毕。如果使用数据表的轴正在运动，禁止更新数据表

short Rt200x_PtsTable (WORD CardNo, WORD axis, DWORD count, double *pTime, long *pPos, double *pPercent)	
向指定数据表传送数据，采用PTS模式	
参数	说明
CardNo	控制卡卡号
axis	指定轴号，取值范围：0~7
count	数据点个数，每个数据表具有1000个存储空间，每个数据点占用1个存储空间
pTime	数据点时间数组，单位：s（精度：ms）；数组长度：count
pPos	数据点位置数组，单位：pulse；数组长度：count
pPercent	数据点百分比数组，百分比的取值范围：[0,100]；数组长度：count
返回值	错误代码
注意	下载的第一组（即起始点）数据中位置、时间必须为0；数组中的数据都是以起始点的数据为参考点 调用该指令向数据表中传递数据时，会删除数据表中原先的数据，因此所有数据应当一次传送完毕。如果使用数据表的轴正在运动，禁止更新数据表

short Rt200x_PvtTable(WORD CardNo,WORD axis,DWORD count,double *pTime,long *pPos,double* pVel)	
向指定数据表传送数据，采用PVT模式	
参数	说明
CardNo	控制卡卡号
axis	指定轴号，取值范围：0~7
count	数据点个数，每个数据表具有5000个存储空间，每个数据点占用1个存储空间
pTime	数据点时间数组，单位：s（精度：ms）；数组长度：count
pPos	数据点位置数组，单位：pulse；数组长度：count
pVel	数据点速度数组，单位：pulse/s；数组长度：count
返回值	错误代码
注意	下载的第一组（即起始点）数据中位置、时间、速度必须为0；数组中的数据都是以起始点的数据为参考点 调用该指令向数据表中传递数据时，会删除数据表中原先的数据，因此所有数据应当一次传送完毕。如果使用数据表的轴正在运动，禁止更新数据表

short Rt200x_PvtsTable(WORD CardNo,WORD axis,DWORD count,double *pTime,long *pPos,double velBegin,double velEnd)	
向指定数据表传送数据，采用PVTS模式	
参数	说明
CardNo	控制卡卡号

axis	指定轴号，取值范围：0~7
count	数据点个数，每个数据表具有5000个存储空间，每个数据点占用1个存储空间
pTime	数据点时间数组，单位：s（精度：ms）；数组长度：count
pPos	数据点位置数组，单位：pulse；数组长度：count
velBegin	设置的第一点的速度，单位：pulse/s
velEnd	设置的最后一点的速度，单位：pulse/s
返回值	错误代码
注意	下载的第一组（即起始点）数据中位置、时间、速度必须为0；数组中的数据都是以起始点的数据为参考点 调用该指令向数据表中传递数据时，会删除数据表中原先的数据，因此所有数据应当一次传送完毕。如果使用数据表的轴正在运动，禁止更新数据表

short Rt200x_PvtMove(WORD CardNo,WORD AxisNum,WORD* AxisList)	
启动PVT运动	
参数	说明
CardNo	控制卡卡号
AxisNum	轴数
AxisList	轴列表
返回值	错误代码

5.12. 伺服驱动专用接口函数

short Rt200x_write_sevon_pin(WORD CardNo,WORD axis,WORD on_off)	
控制指定轴的伺服使能端口的输出	
参数	说明
CardNo	控制卡卡号
axis	指定轴号，取值范围：0~7
on_off	设置伺服使能端口电平，0：低电平，1：高电平
返回值	错误代码

short Rt200x_read_sevon_pin(WORD CardNo,WORD axis)	
读取指定轴的伺服使能端口的电平	
参数	说明
CardNo	控制卡卡号
axis	指定轴号，取值范围：0~7
返回值	伺服使能端口电平，0：低电平，1：高电平

short Rt200x_read_rdy_pin(WORD CardNo,WORD axis)	
读取指定轴的RDY端口的电平	
参数	说明
CardNo	控制卡卡号

axis	指定轴号，取值范围：0~7
返回值	RDY端口电平，0：低电平，1：高电平

short Rt200x_set_inp_mode(WORD CardNo,WORD axis,WORD enable,WORD inp_logic)	
设置指定轴的INP信号	
参数	说明
CardNo	控制卡卡号
axis	指定轴号，取值范围：0~7
enable	INP 信号使能，0：禁止，1：允许
inp_logic	INP 信号的有效电平，0：低有效，1：高有效
返回值	错误代码
注意	当使能INP信号功能后，只有在INP信号为有效状态时，对应的轴才能进行运动，否则此时检测轴的状态是正在运行（即对轴运动作限制）

short Rt200x_get_inp_mode(WORD CardNo,WORD axis,WORD *enable,WORD *inp_logic)	
读取指定轴的INP信号设置	
参数	说明
CardNo	控制卡卡号
axis	指定轴号，取值范围：0~7
enable	返回INP信号使能状态

inp_logic	返回设置的INP信号有效电平
返回值	错误代码

short Rt200x_set_alm_mode(WORD CardNo,WORD axis,WORD enable,WORD alm_logic,WORD alm_action)	
设置指定轴的ALM信号	
参数	说明
CardNo	控制卡卡号
axis	指定轴号，取值范围：0~7
enable	ALM信号使能，0：禁止，1：允许
alm_logic	ALM信号的有效电平，0：低有效，1：高有效
alm_action	ALM信号的制动方式，0：立即停止（只支持该方式）
返回值	错误代码

short Rt200x_get_alm_mode(WORD CardNo,WORD axis,WORD *enable,WORD *alm_logic, WORD *alm_action)	
读取指定轴的ALM信号设置	
参数	说明
CardNo	控制卡卡号
axis	指定轴号，取值范围：0~7
enable	返回ALM信号使能状态

alm_logic	返回设置的ALM信号有效电平
alm_action	返回ALM信号的制动方式
返回值	错误代码

short Rt200x_write_erc_pin(WORD CardNo,WORD axis,WORD sel)	
控制指定轴的ERC信号输出	
参数	说明
CardNo	控制卡卡号
axis	指定轴号，取值范围：0~7
sel	输出电平，0：低电平，1：高电平
返回值	错误代码

short Rt200x_read_erc_pin(WORD CardNo,WORD axis)	
读取指定轴的ERC端口电平	
参数	说明
CardNo	控制卡卡号
axis	指定轴号，取值范围：0~7
返回值	ERC端口电平，0：低电平，1：高电平

5.13. 通用输入输出 IO 函数

short Rt200x_read_inbit(WORD CardNo, WORD bitno)	
读取指定控制卡的某个输入端口的电平	
参数	说明
CardNo	控制卡卡号
bitno	输入端口号，取值范围：0~31
返回值	指定的输入端口电平：0：低电平，1：高电平

short Rt200x_write_outbit(WORD CardNo, WORD bitno,WORD on_off)	
设置指定控制卡的某个输出端口的电平	
参数	说明
CardNo	控制卡卡号
bitno	输入端口号，取值范围：0~31
on_off	输出电平，0：低电平，1：高电平
返回值	错误代码

short Rt200x_read_outbit(WORD CardNo, WORD bitno)	
读取指定控制卡的某个输出端口的电平	
参数	说明
CardNo	控制卡卡号

bitno	输入端口号，取值范围：0~31
返回值	指定输出端口的电平，0：低电平，1：高电平

DWORD Rt200x_read_inport(WORD CardNo,WORD portno)	
读取指定控制卡的全部输入端口的电平	
参数	说明
CardNo	控制卡卡号
portno	IO组号，取值范围：0
返回值	bit0~bit31值分别代表第0~31号输入端口的电平

DWORD Rt200x_read_outport(WORD CardNo,WORD portno)	
读取指定控制卡的全部输出口的电平	
参数	说明
CardNo	控制卡卡号
portno	保留参数，固定值为0
返回值	bit0~bit31的值分别代表第0~31号输出端口的电平

short Rt200x_write_outport(WORD CardNo, WORD portno,DWORD poRt200x_value)	
设置指定控制卡的全部输出口的电平	
参数	说明

CardNo	控制卡卡号
portno	保留参数，固定值为0
poRt200x_value	bit0~bit31的值分别代表第0~31号输出端口的电平
返回值	错误代码

short Rt200x_reverse_outbit(WORD CardNo,WORD bitno,double reverse_time)	
IO输出延时翻转	
参数	说明
CardNo	控制卡卡号
Bitno	输出端口号,取值范围：0~15
reverse_time	延时翻转时间，单位：s
返回值	错误代码
注意	该函数只能对OUT0~15端口进行操作 当延时翻转时间参数设置为0时，此时延时翻转时间将为无限大

5.14. 手轮功能函数

short Rt200x_set_handwheel_inmode (WORD CardNo,WORD axis,WORD inmode,long multi, double vh)	
设置单轴手轮运动控制输入方式	
参数	说明
CardNo	控制卡卡号

axis	指定轴号，取值范围：0~7
inmode	手轮输入方式，0：A、B相位正交信号；1：脉冲+方向信号
multi	手轮倍率，正数表示默认方向，负数表示与默认方向反向
vh	保留参数，固定值为0
返回值	错误代码
注意	Rt200x有八个手轮通道，与八个轴一一对应，每次只能启动一个轴运动

short Rt200x_get_handwheel_inmode (WORD CardNo,WORD axis,WORD* inmode,long* multi, double* vh)	
读取单轴手轮运动控制输入方式	
参数	说明
CardNo	控制卡卡号
axis	指定轴号，取值范围：0~7
inmode	返回手轮输入方式
multi	返回手轮倍率
vh	保留参数，固定值为0
返回值	错误代码

short Rt200x_handwheel_move(WORD CardNo,WORD axis)	
启动手轮运动	
参数	说明

CardNo	控制卡卡号
axis	指定轴号，取值范围：0~7
返回值	错误代码
注意	当启动手轮运动后,只有发送Rt200x_stop或Rt200x_emg_stop命令后才会退出手轮模式

5.15. 编码器函数

short Rt200x_set_counter_inmode(WORD CardNo,WORD axis,WORD mode)	
设置编码器的计数方式	
参数	说明
CardNo	控制卡卡号
axis	指定轴号，取值范围：0~7
mode	编码器的计数方式： 0：非A/B相(脉冲/方向) 1：1×A/B 2：2×A/B 3：4×A/B
返回值	错误代码

short Rt200x_get_counter_inmode(WORD CardNo,WORD axis,WORD *mode)
读取编码器的计数方式

参数	说明
CardNo	控制卡卡号
axis	指定轴号，取值范围：0~7
mode	返回编码器的计数方式
返回值	错误代码

short Rt200x_set_encoder(WORD CardNo,WORD axis,long encoder_value)	
设置指定轴编码器脉冲计数值	
参数	说明
CardNo	控制卡卡号
axis	指定轴号，取值范围：0~7
encoder_value	编码器计数值，单位：pulse
返回值	错误代码

long Rt200x_get_encoder(WORD CardNo,WORD axis)	
读取指定轴编码器脉冲计数值	
参数	说明
CardNo	控制卡卡号
axis	指定轴号，取值范围：0~7
返回值	编码器计数值，单位：pulse

注意	此函数axis参数为8时可以读手轮编码器计数值

short Rt200x_set_ez_mode(WORD CardNo,WORD axis,WORD ez_logic,WORD ez_mode,double filter)	
设置指定轴的EZ信号	
参数	说明
CardNo	控制卡卡号
axis	指定轴号，取值范围：0~7
ez_logic	EZ信号有效电平，0：低有效，1：高有效
ez_mode	保留参数，固定值为0
filter	保留参数，固定值为0
返回值	错误代码

short Rt200x_get_ez_mode(WORD CardNo,WORD axis,WORD *ez_logic,WORD *ez_mode,double* filter)	
读取指定轴的EZ信号设置	
参数	说明
CardNo	控制卡卡号
axis	指定轴号，取值范围：0~7
ez_logic	返回设置的EZ信号有效电平
ez_mode	保留参数，固定值为0

filter	保留参数，固定值为0
返回值	错误代码

5.16. 高速位置锁存函数

short Rt200x_set_ltc_mode(WORD CardNo,WORD axis,WORD ltc_logic,WORD ltc_mode,double filter)	
设置指定轴的LTC信号	
参数	说明
CardNo	控制卡卡号
axis	指定轴号，取值范围：0~7
ltc_logic	LTC信号的触发方式，0：下降沿锁存，1：上升沿锁存
ltc_mode	保留参数，固定值为0
filter	保留参数，固定值为0
返回值	错误代码

short Rt200x_get_ltc_mode(WORD CardNo,WORD axis,WORD *ltc_logic,WORD *ltc_mode,double*filter)	
读取指定轴的LTC信号设置	
参数	说明
CardNo	控制卡卡号
axis	指定轴号，取值范围：0~7
ltc_logic	返回设置的LTC信号的触发方式

ltc_mode	保留参数，固定值为0
filter	保留参数，固定值为0
返回值	错误代码

short Rt200x_set_latch_mode(WORD CardNo,WORD axis,WORD all_enable,WORD latch_source, WORD triger_chunnel)	
设置锁存方式	
参数	说明
CardNo	控制卡卡号
axis	指定轴号，取值范围：0~7
all_enable	锁存方式： 0：单次锁存 1：四轴同时单次锁存,只支持反馈位置 2：连续锁存
latch_source	锁存源，0：指令位置计数器，1：编码器计数器
triger_chunnel	保留参数，固定值为0
返回值	错误代码
注意	当设置了四轴同时锁存后，所有通道会重新锁存，之前的值会被舍弃。四轴同时锁存的端口固定为LTC0。当设置单次锁存或连续锁存方式时，都是单轴独立锁存

short Rt200x_get_latch_mode(WORD CardNo, WORD axis, WORD* all_enable, WORD* latch_source, WORD* triger_chunnel)

读取锁存方式	
参数	说明
CardNo	控制卡卡号
axis	指定轴号，取值范围：0~7
all_enable	返回锁存方式设置
latch_source	返回锁存源设置
triger_chunnel	保留参数，固定值为0
返回值	错误代码

short Rt200x_set_latch_stop_time(WORD CardNo,WORD axis,long time)	
设置LTC端口触发延时急停时间	
参数	说明
CardNo	控制卡卡号
axis	指定轴号，取值范围：0~7
time	触发延时停止时间，单位：us，取值范围：1us~50ms
返回值	错误代码

short Rt200x_get_latch_stop_time(WORD CardNo,WORD axis,long* time)	
读取LTC端口触发延时急停时间	
参数	说明

CardNo	控制卡卡号
axis	指定轴号，取值范围：0~7
time	返回触发延时停止时间设置，单位：us
返回值	错误代码

long Rt200x_get_latch_value(WORD CardNo,WORD axis)	
从控制卡内读取编码器锁存器的值	
参数	说明
CardNo	控制卡卡号
axis	指定轴号，取值范围：0~7
返回值	锁存值
注意	当选择锁存方式为单次锁存时，用此函数读取锁存值

short Rt200x_get_latch_flag(WORD CardNo,WORD axis)	
从控制卡内读取指定卡内锁存器的标志位	
参数	说明
CardNo	控制卡卡号
axis	指定轴号，取值范围：0~7
返回值	0：未触发锁存，1：已触发锁存
注意	当选择锁存方式为单次锁存时，用此函数读取锁存器标志位

short Rt200x_reset_latch_flag(WORD CardNo,WORD axis)	
复位指定卡的锁存器的标志位	
参数	说明
CardNo	控制卡卡号
axis	指定轴号，取值范围：0~7
返回值	错误代码
注意	当使用锁存功能前，必须先调用此函数复位锁存器的标志位

short Rt200x_get_latch_flag_extern(WORD CardNo,WORD axis)	
从PC缓存中读取锁存器已锁存个数	
参数	说明
CardNo	控制卡卡号
axis	指定轴号，取值范围：0~7
返回值	当前已锁存个数，0表示此时无锁存值
注意	当选择锁存方式为连续锁存时，用此函数读取已锁存个数

long Rt200x_get_latch_value_extern(WORD CardNo,WORD axis,WORD index)	
按索引号读取PC缓冲区中已保存的锁存值	
参数	说明
CardNo	控制卡卡号

axis	指定轴号，取值范围：0~7
index	索引号
返回值	锁存值
注意	当选择锁存方式为连续锁存时，用此函数读取锁存值。索引号按锁存顺序从0开始排列（即第一次锁存的位置值存在索引号为0处，第二次锁存的位置值存在索引号为1处，以此类推）

5.17. 位置比较函数

short Rt200x_compare_set_config(WORD CardNo, WORD axis,WORD enable, WORD cmp_source)	
设置一维位置比较器	
参数	说明
CardNo	控制卡卡号
axis	指定轴号，取值范围：0~7
enable	比较功能状态，0：禁止，1：使能
cmp_source	比较源，0：指令位置计数器，1：编码器计数器
返回值	错误代码

short Rt200x_compare_get_config(WORD CardNo, WORD axis,WORD* enable,WORD* cmp_source)	
读取一维位置比较器设置	
参数	说明

CardNo	控制卡卡号
axis	指定轴号，取值范围：0~7
enable	返回比较功能状态
cmp_source	返回比较源
返回值	错误代码

short Rt200x_compare_clear_points(WORD CardNo,WORD axis)	
清除已添加的所有一维位置比较点	
参数	说明
CardNo	控制卡卡号
axis	指定轴号，取值范围：0~7
返回值	错误代码

short Rt200x_compare_add_point(WORD CardNo,WORD axis, long pos, WORD dir, WORD action, DWORD actpara)	
添加一维位置比较点	
参数	说明
CardNo	控制卡卡号
axis	指定轴号，取值范围：0~7
pos	比较位置，单位：pulse
dir	比较模式，0：小于等于，1：大于等于

action	比较点触发功能编号
actpara	比较点触发功能参数
	Rt2008_compare_add_point 函数 action, actpara 参数值 action / actpara / 功能 1 / IO 号 / IO 置为高电平 2 / IO 号 / IO 置为低电平 3 / IO 号 / 取反 IO 5 / IO 号 / 输出 500us 脉冲 6 / IO 号 / 输出 1ms 脉冲 7 / IO 号 / 输出 10ms 脉冲 8 / IO 号 / 输出 100ms 脉冲 13 / 轴号 / 停止指定轴
返回值	错误代码

short Rt200x_compare_get_current_point(WORD CardNo,WORD axis,long* pos)	
读取当前一维比较点位置	
参数	说明
CardNo	控制卡卡号
axis	指定轴号，取值范围：0~7
pos	返回当前比较点位置，单位：pulse
返回值	错误代码

short Rt200x_compare_get_points_runned(WORD CardNo,WORD axis,long* PointNum)	
查询已经比较过的一维比较点个数	
参数	说明
CardNo	控制卡卡号

axis	指定轴号，取值范围：0~7
PointNum	返回已经比较过的点数
返回值	错误代码

short Rt200x_compare_get_points_remained(WORD CardNo,WORD axis,long* PointNum)	
查询可以加入的一维比较点个数	
参数	说明
CardNo	控制卡卡号
axis	指定轴号，取值范围：0~7
PointNum	返回可以加入的比较点数
返回值	错误代码

short Rt200x_compare_set_config_extern (WORD CardNo, WORD enable, WORD cmp_source)	
设置二维位置比较器	
参数	说明
CardNo	控制卡卡号
enable	二维位置比较功能状态，0：禁止，1：使能
cmp_source	二维位置比较源，0：指令位置计数器，1：编码器计数器
返回值	错误代码

short Rt200x_compare_get_config_extern(WORD CardNo, WORD* enable,WORD* cmp_source)	
读取二维位置比较器设置	
参数	说明
CardNo	控制卡卡号
enable	返回比较功能状态
cmp_source	返回比较源
返回值	错误代码

short Rt200x_compare_clear_points_extern(WORD CardNo)	
清除已添加的所有二维位置比较点	
参数	说明
CardNo	控制卡卡号
返回值	错误代码

short Rt200x_compare_add_point_extern(WORD CardNo,WORD* axis, long* pos, WORD* dir, WORD action, DWORD actpara)	
添加二维位置比较点	
参数	说明
CardNo	控制卡卡号
axis	指定卡上的即将进行位置比较的轴列表(两个轴)

pos	二维位置比较位置列表，单位：pulse
dir	比较模式列表，0：小于等于，1：大于等于
action	二维位置比较点触发功能编号
actpara	二维位置比较点触发功能参数
	Rt2008_compare_add_point_ex 函数 action，actpara 参数值 action / actpara / 功能 1 / IO 号 / IO 置为高电平 2 / IO 号 / IO 置为低电平 3 / IO 号 / 取反 IO 5 / IO 号 / 输出 500us 脉冲 6 / IO 号 / 输出 1ms 脉冲 7 / IO 号 / 输出 10ms 脉冲 8 / IO 号 / 输出 100ms 脉冲 13 / 轴号 / 停止指定轴
返回值	错误代码

short Rt200x_compare_get_current_point_extern(WORD CardNo,long *pos)	
读取当前二维位置比较点位置	
参数	说明
CardNo	控制卡卡号
pos	返回当前二维位置比较点位置，单位：pulse
返回值	错误代码

short Rt200x_compare_get_points_runned_extern(WORD CardNo,long *PointNum)	
查询已经比较过的二维比较点个数	
参数	说明

CardNo	控制卡卡号
PointNum	返回已经比较过的二维位置比较点数
返回值	错误代码

short Rt200x_compare_get_points_remained_extern(WORD CardNo, long *PointNum)	
查询可以加入的二维比较点个数	
参数	说明
CardNo	控制卡卡号
PointNum	返回可以加入的二维位置比较点数
返回值	错误代码

5.18. 异常信号接口函数

short Rt200x_set_emg_mode(WORD CardNo,WORD axis,WORD enable,WORD emg_logic)	
设置EMG急停信号	
参数	说明
CardNo	控制卡卡号
axis	保留参数，固定值为0（8轴共用EMG信号）
enable	允许/禁止信号功能，0：禁止，1：允许
emg_logic	EMG信号有效电平，0：低有效，1：高有效

返回值	错误代码

short Rt200x_get_emg_mode (WORD CardNo, WORD axis,WORD *enbale,WORD* logic)	
读取EMG急停信号设置	
参数	说明
CardNo	控制卡卡号
axis	保留参数，固定值为0（8轴共用EMG信号）
enable	返回EMG信号功能状态
logic	返回设置的EMG信号有效电平
返回值	错误代码

short Rt200x_set_dec_stop_time(WORD CardNo,WORD axis,double stop_time)	
设置减速停止时间	
参数	说明
CardNo	控制卡卡号
axis	指定轴号，取值范围：0~7
stop_time	减速时间，单位：s
返回值	错误代码
注意	当发生异常停止时，如：调用Rt200x_stop函数、限位信号（软硬件）被触发等进行减速停止时，减速停止时间都为Rt200x_set_dec_stop_time函数里设置的减速时间

short Rt200x_get_dec_stop_time(WORD CardNo,WORD axis,double *stop_time)	
读取减速停止时间设置	
参数	说明
CardNo	控制卡卡号
axis	指定轴号，取值范围：0~7
stop_time	返回设置的减速时间，单位：s
返回值	错误代码

5.19. 检测轴到位状态函数

short Rt200x_set_factor_error(WORD CardNo,WORD axis,double factor,long error)	
设置位置误差带	
参数	说明
CardNo	控制卡卡号
axis	指定轴号，取值范围：0~7
factor	编码器系数
error	位置误差带，单位：pulse
返回值	错误代码
注意	当使用Rt200x_check_success_encoder 函数检测编码器是否到位时，其用于判断的编码器位置为：编码器计数值乘以编码器系数的值。

short Rt200x_get_factor_error(WORD CardNo,WORD axis,double* factor,long* error)	
读取位置误差带设置	
参数	说明
CardNo	控制卡卡号
axis	指定轴号，取值范围：0~7
factor	返回编码器系数设置
error	返回位置误差带设置
返回值	错误代码

short Rt200x_check_success_pulse(WORD CardNo,WORD axis)	
检测指令到位	
参数	说明
CardNo	控制卡卡号
axis	指定轴号，取值范围：0~7
返回值	0：表示指令位置在设定的目标位置的误差带之外 1：表示指令位置在设定的目标位置的误差带之内
注意	该函数只适用于单轴运动 检测函数请在Rt200x_check_done检测到轴停止后调用，函数调用后会等待轴到位后返回，如果调用函数100ms内未到位，函数超时返回认为不到位

short Rt200x_check_success_encoder(WORD CardNo,WORD axis)

检测编码器到位	
参数	说明
CardNo	控制卡卡号
axis	指定轴号，取值范围：0~7
返回值	0：表示编码器位置在设定的目标位置的误差带之外 1：表示编码器位置在设定的目标位置的误差带之内
注意	该函数只适用于单轴运动 检测函数请在Rt200x_check_done检测到轴停止后调用，函数调用后会等待轴到位后返回，如果调用函数100ms内未到位，函数超时返回认为不到位

5.20. 软着陆函数

WORD Rt200x_pmove_extern(WORD cardId,WORD axis, double dist, double Min_Vel,double Max_Vel, double TEB, double Tdec, double stop_Vel, double s_para, WORD posi_mode)	
实现profile，pmove整合，缩短指令时间，适用于高速场合	
参数	说明
cardId	控制卡卡号
axis	指定轴号，取值范围：0~7
dist	终点位置,单位：pulse
Min_Vel	起始速度
Max_Vel	最大速度

TEB	加速时间
Tdec	减速时间
stop_Vel	停止速度
s_para	平滑时间，单位：s，范围[0~0.5]
posi_mode	运动模式，0：相对模式，1：绝对模式
返回值	错误代码
注意	该函数只适用于单轴运动 检测函数请在Rt200x_check_done检测到轴停止后调用，函数调用后会等待轴到位后返回，如果调用函数100ms内未到位，函数超时返回认为不到位

WORD Rt200x_t_pmove_extern(WORD cardId,WORD axis, double mid_pos, double target_pos, double Min_Vel, double Max_Vel, double stop_Vel, double EB, double dec, WORD posi_mode)	
实现profile，pmove整合，缩短指令时间，并且实现软着陆	
参数	说明
cardId	控制卡卡号
axis	指定轴号，取值范围：0~7
mid_pos	第一段的终点位置,单位：pulse
aim_pos	第二段的终点位置,单位：pulse
Min_Vel	起始速度
Max_Vel	最大速度
stop_Vel	停止速度

EB	加速时间
dec	减速时间
posi_mode	运动模式，0：相对模式，1：绝对模式
返回值	错误代码

WORD Rt200x_update_target_position_extern(WORD cardId,WORD axis, double mid_pos, double aim_pos,double vel,WORD posi_mode)	
强行改变指定轴的当前目标位置并且实现软着陆	
参数	说明
cardId	控制卡卡号
axis	指定轴号，取值范围：0~7
mid_pos	第一段的终点位置,单位：pulse
aim_pos	第二段的终点位置,单位：pulse
vel	保留参数，固定值为0
posi_mode	保留参数，固定值为0
返回值	错误代码
注意	该函数适用于指定轴停止状态或点位运动中的变位，并且实现软着陆 参数mid_pos，aim_pos为绝对坐标位置值，无论当前的运动模式为绝对坐标还是相对坐标模式。mid_pos以速度规划指令中的最大速度运行，aim_pos以速度规划指令中的停止速度运行。

5.21. 运动函数错误码说明

错误码	名称	含义
0	ERR_NOERR	成功
1	ERR_UNKNOWN	未知错误
2	ERR_PARAERR	参数错误
3	ERR_TIMEOUT	操作超时
4	ERR_CONTROLLERBUSY	控制卡状态忙
6	ERR_CONTILINE	连续插补错误
8	ERR_CANNOT_CONNECTETH	无法连接错误
9	MOTION_ERR_HANDLEERR	卡号错误，可能由于函数参数中的卡号或轴号超出范围产生，比如有两张卡分别为0号和1号卡，当轴号为8时根据计算应该为2号卡，此时就会产生错误。
10	ERR_SENDERR	数据传输错误，请检查PCI槽位是否松动
12	ERR_FIRMWAREERR	固件文件错误
14	ERR_FIRMWAR_MISMATCH	固件不匹配
20	ERR_FIRMWARE_INVALID_PARA_ERR	固件参数错误
22	ERR_FIRMWARE_STATE_ERR	固件当前状态不允许操作
24	ERR_FIRMWARE_CARD_NOT_SUPPORT	控制器不支持的功能