

# 锐特运动控制卡

## RT1000 用户手册

## 目录

锐特控制运动控制卡.....	1
1 简介.....	4
2 硬件概述.....	4
2.1 硬件指标.....	4
2.1.1 电机控制指标.....	4
2.1.2 I/O 信号控制指标.....	4
2.1.3 通用指标：.....	4
2.2 硬件布局.....	5
2.3 硬件设置.....	5
2.3.1 控制卡电源供给.....	5
2.3.2 脉冲和方向控制接口.....	5
2.3.3 机械位置控制接口.....	6
2.3.4 通用数字输入/输出信号接口.....	8
2.3.5 多卡运行.....	9
2.4 运动控制平台位置传感器及控制信号布局示例.....	10
3 硬件配置与安装.....	10
3.1 硬件配置.....	10
3.1.1 跳线配置.....	10
3.1.2 开关配置.....	11
3.2 硬件安装.....	12
3.2.1 RT-CNC68 功能板.....	12
3.2.2 RT-CNC37 IO 扩展板.....	13
3.2.3 运动控制卡应用示意图.....	13
4 软件系统概述.....	14
4.1 运动控制函数库.....	14
4.2 运动控制函数库.....	14
4.2.1 初始化、关闭运动控制卡.....	14
4.2.2 设置脉冲输出模式.....	15
4.2.3 单轴位置和速度控制.....	16
4.2.4 多轴运动控制.....	17
4.2.5 回原点运动.....	20
4.2.6 指令脉冲计数.....	20
4.2.7 通用 I/O 控制.....	21
5 驱动程序安装.....	22
5.1 在 Windows 7 操作系统环境中安装步骤.....	22
6 演示软件及使用.....	26
6.1 I/O 检测演示.....	27
6.2 运动测试演示.....	28
7 用户系统开发.....	30
7.1 基于 windows 平台的应用软件结构.....	30
8 附录.....	31

8.1 硬件信号接口表.....	31
8.1.1 接口 X1 引脚定义.....	31
8.1.2 接口 X3 引脚定义.....	32
8.1.3 接口 J1 引脚定义.....	33
8.2 运动控制函数.....	34
8.2.1 函数列表.....	35
8.2.2 函数说明.....	36

# 1 简介

RT1000 是一款基于 PCI 总线的脉冲式运动控制卡，可控制多达 4 个步进电机或伺服电机。其位置指令可用单路脉冲（脉冲+方向）或双路脉冲（CW+CCW 脉冲）方式输出；可以是差分式输出电路也可以是单端式输出电路。另外，除了通用输入输出信号接口外，RT1000 还包括原点、限位、减速等专用信号接口，具有即插即用、多轴同时同步启停功能，并可选择梯形或 S 形速度曲线，同时具有软件直线插补功能。

## 2 硬件概述

### 2.1 硬件指标

#### 2.1.1 电机控制指标

- 控制轴数：4轴，最多可支持12块控制卡同时工作；
- 控制模式：位置控制、速度控制2种模式；
- 脉冲输出模式：单脉冲（脉冲+方向）或双脉冲（CW +CCW）；
- 最大脉冲输出频率：RT1000：1.2MHz，可梯形速度曲线或S形速度曲线控制；
- 位置脉冲设置范围：0~16,777,215个脉冲(24位)；

#### 2.1.2 I/O 信号控制指标

通用 I/O 信号接口：59 路。

通用输入信号接口：32 路，其中 16 路光电隔离；

通用输出信号接口：27 路，其中 12 路光电隔离；

- 专用 IO 信号接口：20 路，包括正负限位信号±EL、原点信号 ORG 以及正负减速信号±SD，全部光电隔离；
- 通用数字输出口最大驱动电流：45mA，最大承受电压：35V；
- 光电隔离耐压：2500V RMS；

#### 2.1.3 通用指标：

- 工作温度：0℃~50℃；
- 工作湿度：5~85%，非结露；
- 贮存温度：-20℃~80℃；
- 电源：内部芯片电源(由 PCI 总线提供)：+5VDC±5%，最大 900mA；外

部接口电源(需用户提供): +24VDC±5%, 最大 500mA;

## 2.2 硬件布局

## 2.3 硬件设置

### 2.3.1 控制卡电源供给

RT1000 运动控制卡内部 IC 电源由 PC 机的 PCI 总线提供 (+5VDC), 外部接口需用户提供+24VDC 电源。接线示意图如图 2-2 所示:

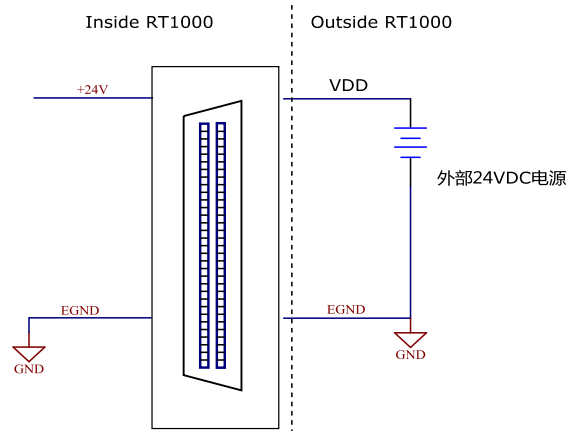


图 2-2 电源接线示意图

### 2.3.2 脉冲和方向控制接口

RT1000 硬件方面提供 4 轴的脉冲和方向控制信号接口。用户通过设定 SW1~SW8 跳线来设定指令脉冲为差分输出或单端输出两种电路 (出厂默认均为差分输出方式)。

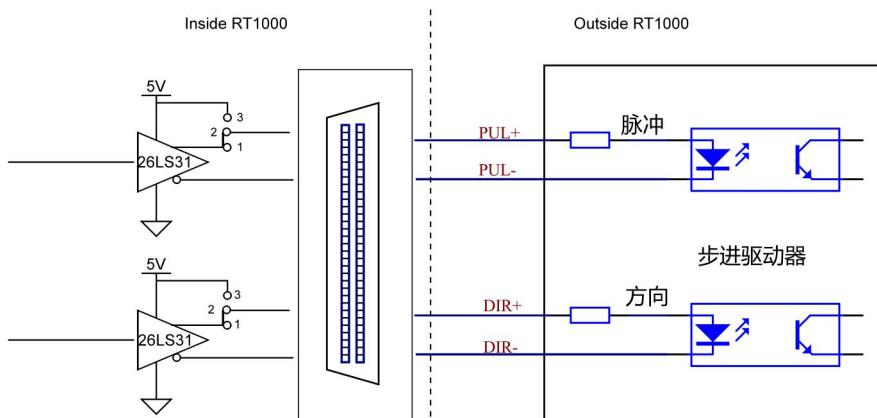


图 2-3 单端/差分输出设置

注意：使用差分输出方式可有效的减少传输中的干扰，建议线路较长时使用差分输出方式。

采用差分输出模式，则 PUL+、PUL-和 DIR+、DIR-端子组成互异的脉冲和方向信号。

采用单端输出模式，则 PUL+和 DIR+共用内部+5V 电压，PUL-和 DIR-输出端为脉冲和方向信号。

无论差分输出还是单端输出，每路 PUL、DIR 的输出电流不超过 20mA。

RT1000 运动控制卡可以输出两类脉冲信号：一种为脉冲+方向形式（单脉冲）；一种为正脉冲+负脉冲（双脉冲）形式。

单脉冲模式输出信号如图 2-4 所示：

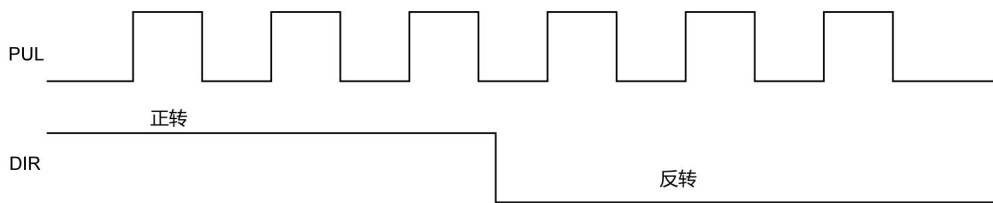


图 2-4 单脉冲模式

双脉冲模式输出信号如图 2-5 所示：

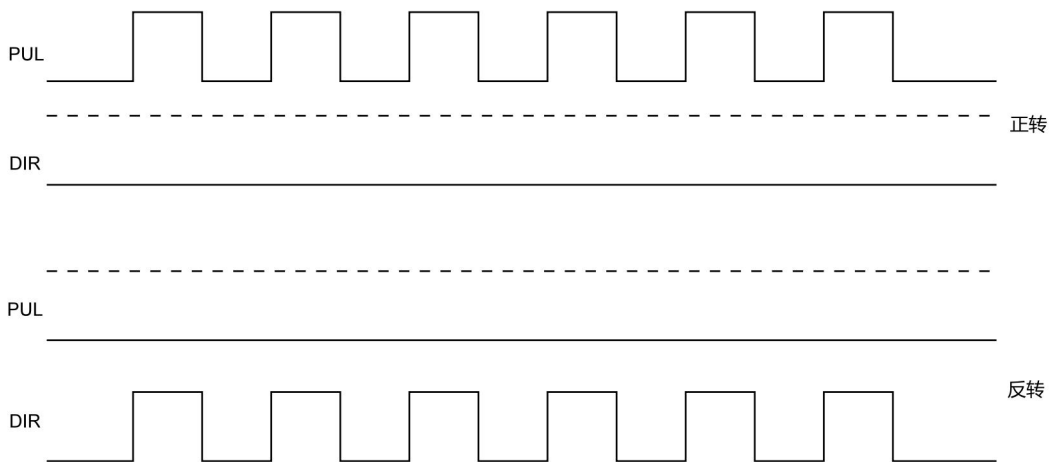


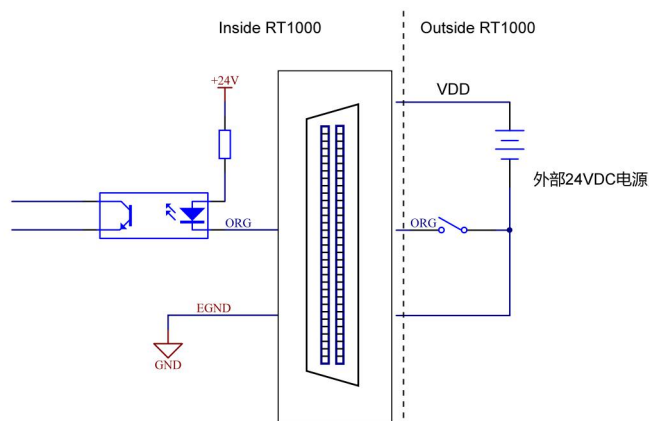
图 2-5 双脉冲模式

### 2.3.3 机械位置控制接口

RT1000 为每个轴提均供了 5 个专用信号输入接口，分别为 1 个原点信号（ORG）输入接口、2 个减速信号（SD+/SD-）输入接口以及 2 个限位信号（EL+/EL-）输入接口。各信号相关功能的实现均由控制卡硬件来完成，用户只需按照正确的接线方式接线，并调用相应功能的函数执行运动指令即可。

5 个专用信号请选用+24V、最小电流大于 6mA 的开关。

专用信号输入接口原理图如下：



### 2.3.3.1 ORG：原点位置信号输入接口

通常运动系统中都要用一个位置传感器设置一个位置参考点，即原点位置，以便于进行精确的位置控制。

RT1000 为每个轴提供了 1 个原点位置传感器输入端口 ORG，对外接口 SCSI68 具体引脚分配见 8.1.1 接口 X1 引脚定义。

原点开关请选用+24V、最小电流大于 6mA 的常开型开关。

RT1000 原点信号输入电路有低通滤波器，可以过滤高频噪声，提高可靠性。

通常在进行运动控制之前，都需要用回原点运动控制平台向原点方向运动，当运动控制卡检测到原点传感器 ORG 信号后，运动自动停止，并将停止位置设置为该轴的原点。

### 2.3.3.2 SD+/SD-：减速位置信号输入接口

运动系统中通常会用一个位置传感器设置一个减速起始点，以便于进行精确的位置控制。运动过程中当运动控制卡检测到减速信号 SD 后，相应轴的运行速度将减速至起始速度，若减速信号 SD 消失，则运行速度会重新加速至最大速度。RT1000 为每个轴都提供了 2 个减速信号：SD+和 SD-。对外接口 SCSI68 具体引脚分配见 8.1.1 接口 X1 引脚定义。

减速开关请选用+24V、最小电流大于 6mA 的常开型开关。RT1000 减速信号输入电路有低通滤波器，可以过滤高频噪声，提高可靠性。

### 2.3.3.3 EL+/EL-：正负限位信号输入接口

运动系统中通常会用一个位置传感器设置一个机械限位点，以确定运动的边界位置，保护机械设备。

RT1000 均为每个轴提供 2 个机械限位信号 EL+ 和 EL-，EL+为正向限位信

号,EL-为反向限位信号。当运动部件接触到限位开关时,EL+/EL-即有效,RT1000将立即停止向该方向输出脉冲。对外接口 SCSI68 具体信号引脚分配见 8.1.1 接口 X1 引脚定义。

限位开关请选用+24V、最小电流大于 6mA 的常开或常闭型开关；限位开关的类型可通过 SW9 开关设定，详见 3.1.2 开关配置。当使用常开型限位开关时，应选择 EL+/EL-信号为高电平有效（即为 OFF）；当使用常闭型限位开关时，应选择 EL+/EL-信号为低电平有效（即为 ON）。出厂默认全部为 OFF，即为常开型模式；

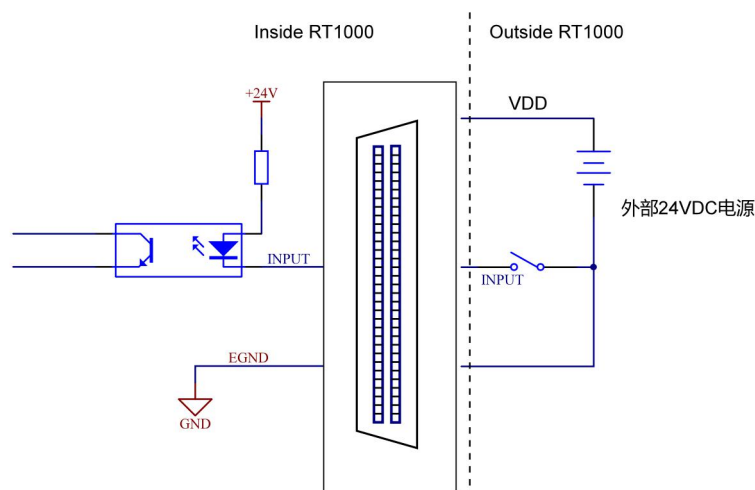
## 2.3.4 通用数字输入/输出信号接口

RT1000 运动控制卡提供了大量的通用数字 I/O 接口。RT1000 有 59 路通用 I/O 口,32 入 27 出,其中 IN1 ~ IN16 以及 OUT1 ~ OUT12 为光电隔离,IN17 ~ IN32 以及 OUT13 ~ OUT27 为非光电隔离接口；

### 2.3.4.1 INPUT 通用数字输入信号接口

通用数字输入信号 INPUT 用于接近开关、光电开关、按键等传感器的信号输入。RT1000 卡插座 X1 上的通用输入口都是光电隔离的，对外接口 SCSI68 具体信号引脚分配见 8.1.1 接口 X1 引脚定义。

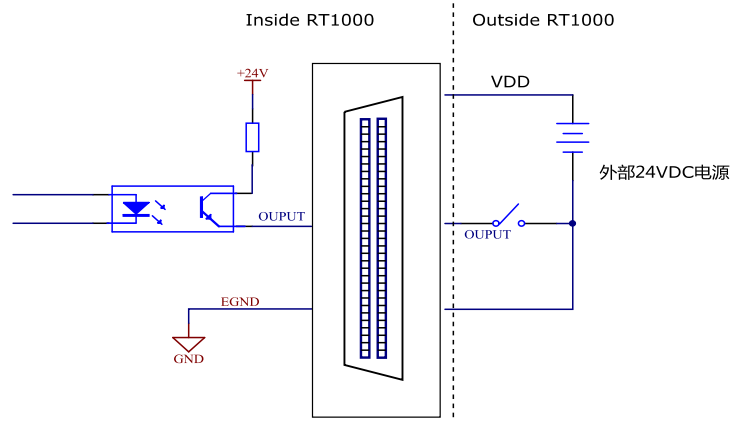
而 RT1000 中增加的通用输入口位于扩展接口 J1 上，均为非光电隔离，但采用专用的接线板后，可实现 J1 接口的通用输入口全部光电隔离。对外接口 DB37 具体信号引脚分配见 8.1.3 接口 J1 引脚定义。通用数字输入信号电路原理图如下图所示：





### 2.3.4.2 OUTPUT 通用数字输出信号接口

通用数字输出信号 OUT 用于控制继电器、指示灯等开关器件。RT1000 卡插座 X1 上的通用输出口都是光电隔离的，对外接口 SCSI68 具体信号引脚分配见 8.1.1 接口 X1 引脚定义。而 RT1000 中增加的通用输出口位于扩展接口 J1 上，均为非光电隔离，但采用专用的接线板后，可实现 J1 接口的通用输出口全部光电隔离。对外接口 DB37 具体信号引脚分配见 8.1.3 接口 J1 引脚定义。通用数字输出信号电路原理图如下图所示：



注：通用数字输出端口的最大驱动电流为 45mA，最大承受电压 35V，严禁使用数字输出口直接驱动电磁阀等大电流、大功率元器件！当需要驱动小型继电器等感性负载时，线圈外必须并联一个续流二极管。

另外，当启动 PC 机时，RT1000 卡输出端的初始状态可以在拨码开关 SW10 上面进行设置，见 3.1.2 开关配置。

### 2.3.5 多卡运行

锐特 RT1000 运动控制卡最多支持 12 张控制卡同时工作。因此，一台 PC 机可以同时控制多达 48 轴步进/伺服电机同时工作。

RT1000 控制卡中 X3 接口是同步启停信号 STA 和 STP 的控制输入口。多卡运行时，如果有两张以上的控制卡要求同时启停，相关控制卡的接口中对应的引脚应连接在一起。以 3 张运动控制卡运行为例，其接线方式如图 2-6 所示：

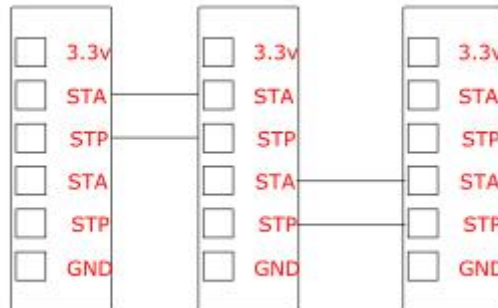


图 2-6 多卡同时启停的连接方式

## 2.4 运动控制平台位置传感器及控制信号布局示例

锐特 RT1000 运动控制卡每轴都配有 2 个限位信号、2 个减速信号和 1 个原点信号输入接口。每路信号都进行了光电隔离，以减少外界对控制卡内部的干扰，保证运动控制的可靠性。

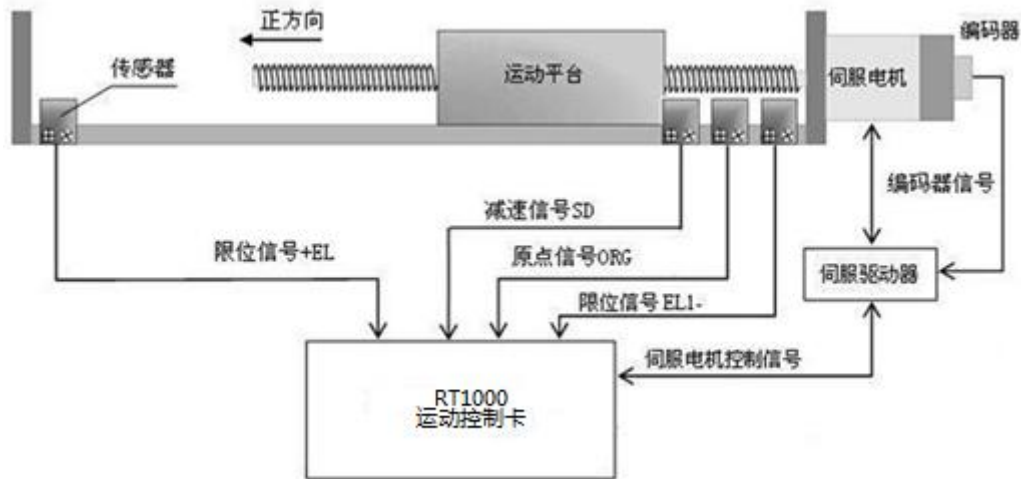


图 2-7 一般运动平台位置传感器及控制信号的布置图

## 3 硬件配置与安装

### 3.1 硬件配置

RT1000 卡支持即插即用功能，其 I/O 地址的选择由系统 BIOS 自动指定，跟所有的支持即插即用的 PCI 卡（如声卡，Modem 卡，网卡）一样，系统 BIOS 均可为其自由分配一个工作地址，您也可在系统 BIOS 中手工设置。RT1000 卡上有多组跳线开关 SW1~SW8 和拨码开关 SW9-SW10，分别用于设置 RT1000 卡的工作方式和参数。

#### 3.1.1 跳线配置

跳线开关用于设置指令脉冲输出方式为差分输出或单端输出。RT1000 卡中表示为 SW1~SW8，设置方式如图 3-1 和如图 3-2 所示。表 3-1 为脉冲和方向信号的跳线表。

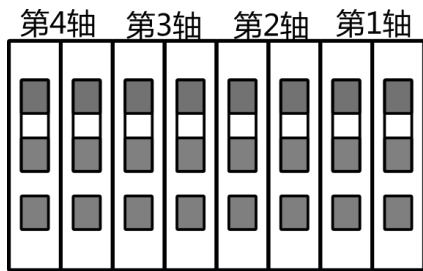


图 3-1 RT1000 差分输出跳线设置

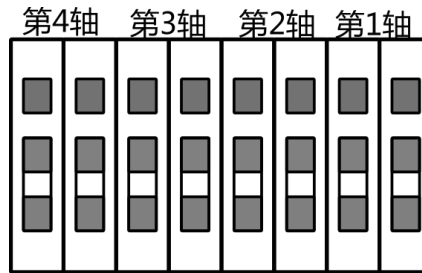


图 3-2 RT1000 单端输出跳线设置

表 3-1 脉冲和方向信号跳线表

针脚号	针脚名	差分输出时， 1脚和2脚短路	单端输出时， 2脚和3脚短路
2	PUL0+	SW2	SW2
4	DIR0+	SW1	SW1
6	PUL1+	SW4	SW4
8	DIR1+	SW3	SW3
35	PUL2+	SW6	SW6
37	DIR2+	SW5	SW5
39	PUL3+	SW8	SW8
41	DIR3+	SW7	SW7

注：出厂时的缺省设置为SW1~SW8全为1脚和2脚短路，即差分输出方式

### 3.1.2 开关配置

RT1000 卡中配置了两个拨码开关 SW9 和 SW10，其中 SW9 用于设置 EL 限位模式，SW10 用来设定输出口的上电初始电平。

#### 3.1.2.1 拨码开关 SW9

RT000 卡的拨码开关 SW9.1~SW9.4 用来设定 EL 限位模式为常开型还是常闭型。EL 限位的缺省设定全部为 OFF，即各轴均为常开限位模式。表 3-2 为 SW9 拨码开关状态表，ON 代表常闭限位模式，OFF 代表常开限位模式。

表 3-2 限位拨码开关表

开关号	轴号	常开模式	常闭模式
SW9.1 (S1.1)	0	OFF (出厂设置)	ON
SW9.2 (S1.2)	1	OFF (出厂设置)	ON
SW9.3 (S1.3)	2	OFF (出厂设置)	ON
SW9.4 (S1.4)	3	OFF (出厂设置)	ON

注：SW9开关的第1位为最左端一位。

### 3.1.2.2 拨码开关 SW10

RT1000 卡中有拨码开关 SW10.1~ SW10.4, 其用来设定输出口的上电初始电平。对应该关系如表 3-3 如示:

表 3-3 RT1000 输出口拨码开关表

开关号	输出口号	高电平	低电平
SW10.1	OUT21—OUT27	OFF (出厂设置)	ON
SW10.2	OUT13—OUT20	OFF (出厂设置)	ON
SW10.3	OUT5—OUT12	OFF (出厂设置)	ON
SW10.4	OUT1—OUT4	OFF (出厂设置)	ON

## 3.2 硬件安装

RT1000 运动控制卡与硬件安装具体参考步骤如下:

1)打开 RT1000 的安装包,参考 3.1 硬件配置的说明,按照实际应用的需求,完成硬件配置;

2)使用辅助接口的用户,请将辅助接口与 RT1000 对应的插座连接,并确保连接牢固,可靠;

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 32bit PCI 插槽,将 RT1000 垂直插入插槽中;

6)将 RT1000 用螺钉紧固在 PC 机机箱上,确保紧固、可靠。

### 3.2.1 RT-CNC68 功能板

若用户选择使用 RT-CNC68 功能板,应使用黑色连接线与位于板卡左侧的 SCSI-II 型 68 针插座接口 X1 (接口定义见附录 8.1.1 )连接起来,具体连接图见下图。



图 3-3 RT-CNC68 功能板接线图

### 3.2.2 RT-CNC37 IO 扩展板

若用户选择使用 RT-CNC38 隔离 IO 扩展板,先使用接口转接线与板卡 40Pin JTAG 接口（位于板卡右侧）连接转成接口 J1（接口定义见附录 8.1.3），再用连接线与扩展板连接，具体连接图见下图。

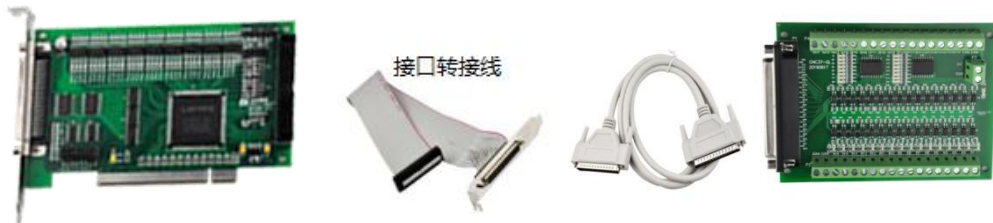
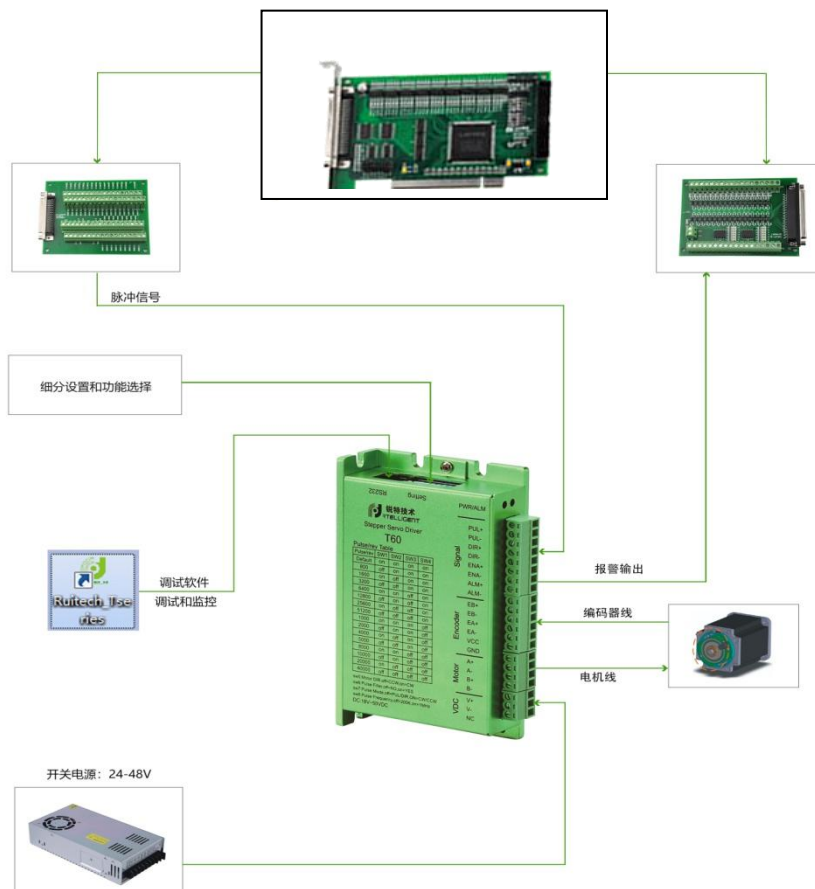


图 3-4 RT-CNC38 IO 扩展板连接图

### 3.2.3 运动控制卡应用示意图



## 4 软件系统概述

RT1000 运动控制卡软件系统包括：驱动程序、运动控制函数库、演示程序、例子程序。

### 4.1 运动控制函数库

RT1000 配套提供 Windows98/2000/XP/ Win7/Win10 等操作系统环境下的驱动程序。客户可以根据自己的需要选择相应的系统平台来开发适合自己的应用软件。硬件驱动程序具体安装方法请参考：5 驱动程序安装。

### 4.2 运动控制函数库

为了方便用户在 RT1000 卡上快速开发出功能强大的应用软件，锐特公司采用科学的函数命名体系，为用户提供一套功能齐全，易学易用的函数库。用户只需根据自己的实际需要调用其中的一部分函数即可开发出各种用途的强大软件。

注：各函数具体功能介绍参考附录：8.2.2 函数说明。

#### 4.2.1 初始化、关闭运动控制卡

在操作 RT1000 运动控制卡之前，必须调用控制卡初始化函数为运动控制卡分配资源。同样，当结束对运动控制卡的操作时，必须调用控制卡关闭函数释放运动控制卡所占用的系统资源，使得所占资源可被其它设备使用。具体相关函数和功能如表 4-1 所示：

表 4-1 初始、关闭控制卡函数说明

名称	功能	参考
Rt1000BoardInit	初始化RT1000卡并分配系统资源	8.2.2.1
Rt1000BoardClose	关闭RT1000卡并释放系统资源	8.2.2.1

注意：程序结束时，必须调用Rt1000BoardClose ()函数释放系统资源。

例程：初始化和关闭控制卡(以 C 语言为例说明，下同)

```
CardCount = Rt1000BoardInit ();
if(CardCount == 0)
{
printf(“\n 没有发现运动控制卡”);
getch();
return();
}
```

```
Rt1000BoardClose ();
```

```
....
```

## 4.2.2 设置脉冲输出模式

锐特 RT1000 卡采用脉冲指令控制步进/伺服电机。针对不同型号的驱动器，在使用控制卡时必须对脉冲信号输出方式进行正确设定，系统才能正常工作。相关函数和功能如表 4-2 所示：

表 4-2 脉冲设置函数说明

名称	功能	参考
Rt1000SetPlsOutmode	设置指定轴的脉冲输出模式	8.2.2.2

注意：在调用运动控制函数之前应先调用该函数来设置指令脉冲模式。

指令脉冲包括两项基本信息：电机运转距离即脉冲数和电机转动方向。有两种基本指令模式：两种基本模式如表 4-3 所示：

表 4-3 脉冲指令模式表

模式	PULn-脚输出	DIRn-脚输出
方向脉冲（pulse/dir）	脉冲信号	方向信号（电平）
双脉冲（CW/CCW）	正向（CW）脉冲	反向（CCW）脉冲

注意：具体设置请参考Rt1000SetPlsOutmode函数具体说明。

### 4.2.2.1 脉冲/方向模式

在此模式下，PULn-端子输出指令脉冲串，脉冲数对应电机运行的相应“距离”，而脉冲频率对应电机运行“速度”。DIRn-端子输出方向信号，该信号的输出电平对应电机的转动方向。此种模式在电机驱动器中应用最多。

脉冲信号可以设置为上升沿有效，即脉冲信号常态为低电平，当变为高电平时电机走一步；也可设置为下降沿有效，即脉冲信号常态为高电平，当变为低电平时电机走一步。所以实际上此种模式下有两种指令类型，如图 4-1 所示。

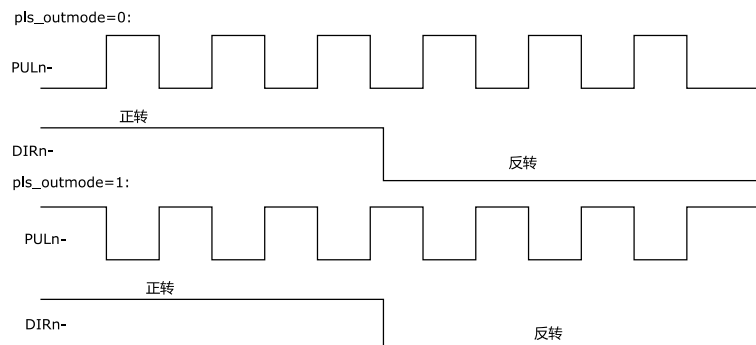


图 4-1 脉冲/方向信号图

## 4.2.2.2 双脉冲模式

在此模式下，PULn-和 DIRn-端子分别表示正向（CW）和反向（CCW）脉冲输出。从 PULn-端子输出的脉冲使电机朝正方向转动，而从 DIRn-端子输出的脉冲使电机朝负方向转动。脉冲信号有上升沿或下降沿有效的选择，所以该种模式下也有两种指令类型，如图 4-2 所示：

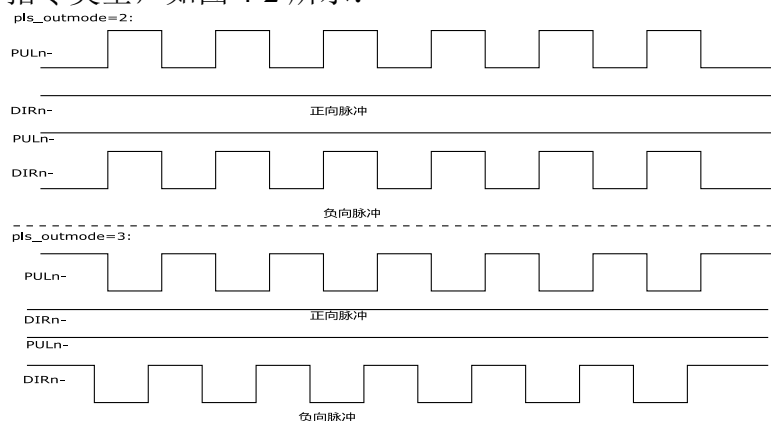


图 4-2 双脉冲信号图

例程：

```
....
Rt1000SetPlsOutmode(0,0); //表示设置第 0 轴脉冲输出模式为方向脉冲模式，
PULn-信号上升沿有效。
....
```

## 4.2.3 单轴位置和速度控制

### 4.2.3.1 梯形速度曲线运动模式

位置控制采用这种速度控制模式较为常见。电机在运动一段指定距离时，加速过程中加速度的大小恒定不变，其运动速度按梯形曲线变化。运动速度之所以要按梯形曲线变化，是因为：电机转子和被拖动的物体具有惯性，不可能在瞬间达到指定速度，因此应该有一定的加速过程。减速时亦是类似，否则电机会因为瞬间力矩不足而出现丢步、过冲（步进电机系统）或振荡（伺服电机系统）现象。

实现梯形速度曲线控制的点位运动函数如表 4-4 所示：

表 4-4 梯形点位控制相关函数说明

	名称	功能	参考
1	Rt1000StartTMove	以梯形速度曲线控制相对坐标的点位运动	8.2.2.4
2	Rt1000StartTaMove	以梯形速度曲线控制绝对坐标的点位运动	8.2.2.4

注意：在RT1000中加减速是对称的，所以加速时间等于减速时间



### 4.2.3.2 S形速度曲线运动模式

若加速度是线性变化，则加速和减速阶段的速度曲线将变为“S”形。这种速度曲线运动模式，运动过程更平稳，且有助于缩短加速过程、降低运动装置的振动和噪声以及延长机械传动部分的使用寿命。

设置 S 形速度曲线及其点位运动的函数如表 4-5 所示：

表 4-5 S 形速度控制相关函数说明

	名称	功能	参考
1	Rt1000StartSMove	以 S 形速度曲线控制相对坐标的点位运动	8.2.2.4
2	Rt1000StartSaMove	以 S 形速度曲线控制绝对坐标的点位运动	8.2.2.4

S 形速度曲线作点位运动例程：

....

```
Rt1000StartSMove(1,50000,500,6000,0.1); //设置 1 号轴相对坐标下运动距离为 5000 个脉冲、起始速度为 500 脉冲/秒、运行速度为 6000 脉冲/秒、加减速时间为 0.1 秒、以 S 形速度曲线开始执行运动
```

....

### 4.2.3.3 连续运动模式

锐特 RT1000 控制卡可以控制电机以梯形或 S 形速度曲线在指定加速时间内从初速度加速至运行速度，然后以该速度连续运行，直至调用停止指令或该轴遇到限位信号才会停止。连续运动的函数如表 4-6 所示：

表 4-6 连续运动相关函数说明

	名称	功能	参考
1	Rt1000StartTvMove	以梯形速度曲线控制连续运动	8.2.2.3
2	Rt1000StartSvMove	以S形速度曲线控制连续运动	8.2.2.3

例程：

....

```
Rt1000StartSvMove(1,500,6000,0.1); //设置 1 号轴起始速度为 500 脉冲/秒、运行速度为 6000 脉冲/秒、加减速时间为 0.1 秒、以 S 形速度曲线开始连续运动。
```

....

## 4.2.4 多轴运动控制

锐特 RT1000B 运动控制卡单张卡可以同时控制 4 个轴以多种方式同时运动，包括多轴联动和直线插补。

### 4.2.4.1 多轴联动

多个轴同时运动，称为多轴联动。RT1000 卡可以控制多个电机同时执行 Rt1000StartTMove 或 Rt1000StartTaMove 等单轴运动函数。所谓同时执行，是在程序中顺序调用这些函数，因为程序执行速度很快，在瞬间几个电机都开始运动，给人的感觉就是“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。如图 4-5 所示。

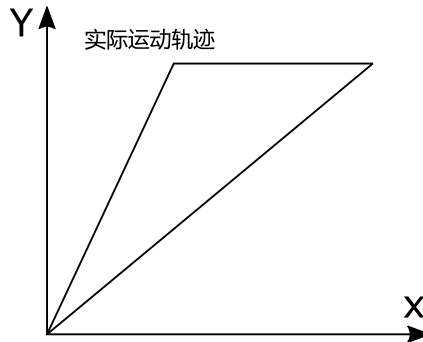


图 4-5 二轴联动示意图

### 4.2.4.2 直线插补运动

RT1000 控制卡可以指定同一卡上任意 2 轴、3 轴以及 4 轴进行直线插补，插补运动的计算是由专用函数完成的，用户只需调用相应的运动函数并设置运动速度、加速度、终点位置等参数，不需要介入插补过程的计算工作。

插补运动与多轴联动的区别：插补运动不但能保证起点、终点位置准确，而且各轴的脉冲是按照直线斜率成比例发出的。RT1000 控制卡直线插补相关函数如表 4-7 所示：

表 4-7 直线插补运动相关函数说明

	名称	功能	参考
1	Rt1000StartTLine	启动多轴相对坐标的直线插补	8.2.2.5
2	Rt1000StartTaLine	启动多轴绝对坐标的直线插补	8.2.2.5

二轴直线插补：

如图 4-7 所示，2 轴直线插补从 P0 点运动至 P1 点，X、Y 轴同时启动，并同时到达终点。X、Y 轴的运动速度之比为  $\Delta X : \Delta Y$ ，与合成的矢量速度  $\Delta P$  的关系为：

$$\frac{\Delta P}{\Delta t} = \sqrt{\left(\frac{\Delta X}{\Delta t}\right)^2 + \left(\frac{\Delta Y}{\Delta t}\right)^2}$$

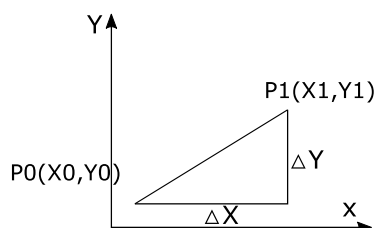


图 4-6 两轴直线插补示意图

调用二轴直线插补函数时,需要设定插补矢量速度,包括其初始速度 StrVel 和运行速度 MaxVel 等参数。

二轴直线插补例程:

```
Short AxisArray[2];
```

```
Short DistArray[2];
```

```
AxisArray[0]=0;// 轴 0、轴 1 直线差补
```

```
AxisArray[1]=1;
```

```
DistArray[0]=1000; // ΔX=1000 脉冲
```

```
DistArray[1]=2000; // ΔY=2000 脉冲
```

```
Rt1000StartTLine (2, AxisArray, DistArray, 400,1000, 0.1); //起始速度为 400 pps, 运行速度为 1000 pps, 加速时间为 0.1s。
```

三轴直线插补:

如图 4-7 示, XYZ 3 轴直线插补从 P0 点运动至 P1 点。插补过程中 3 轴的速度比为 ΔX: ΔY: ΔZ, 与合成的矢量速度 ΔP 的关系为:

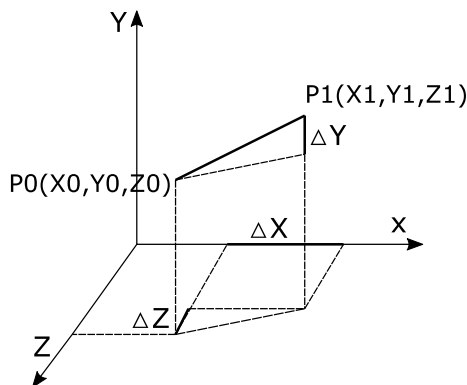


图 4-7 三轴直线插补

$$\frac{\Delta P}{\Delta t} = \sqrt{\left(\frac{\Delta X}{\Delta t}\right)^2 + \left(\frac{\Delta Y}{\Delta t}\right)^2 + \left(\frac{\Delta Z}{\Delta t}\right)^2}$$

调用 3 轴直线插补时,需要设定插补矢量速度,包括初速度 StrVel 和运行速度 MaxVel 等参数。

3 轴直线插补例程:

```
Short AxisArray[3];
```

```
Short DistArray[3];
```

```
AxisArray[0]=0;//轴 0、轴 1、轴 2 直线差补
```

```
AxisArray[1]=1;
```

```
AxisArray[2]=2
```

```
DistArray[0]=1000; //ΔX=1000 脉冲
```

```

DistArray[1]=2000; //△Y=2000 脉冲
DistArray[2]=3000; //△Z=3000 脉冲
Rt1000StartTLine (3, AxisArray, DistArray, 400,1000, 0.1); // 起始速度为
400pps, 运行速度为 1000pps, 加速时间为 0.1s。

```

## 4.2.5 回原点运动

在进行精确的位置控制之前，需要设定运动坐标系的原点。如图 4-9 所示为回原点运动的过程。

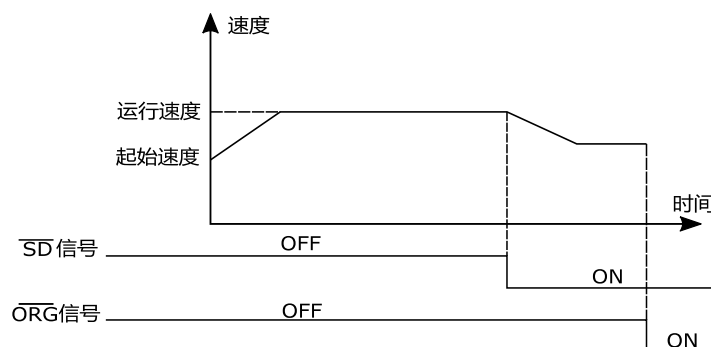


图 4-8 回原点过程示意图

从图 4-8 可以看出，启动回原点运动后，设备向原点方向运动；当碰到减速开关后，电机开始减速。当速度减至初速度后，继续以该速度运行，直到原点信号有效便立即停止，该位置即为原点位置。这种方式在高速回原点的运动中特别有用。具体回原点运动相关函数如表 4-8 所示：

表 4-8 回原点运动相关函数说明

	名称	功能	参考
1	Rt1000HomeMove	启动指定轴进行回原点运动	8.2.2.7
2	Rt1000SetSd	设置减速信号是否使能	8.2.2.11

回原点运动例程：

```

....
Rt1000HomeMove (1,450,5000,0.1); //设置 1 号轴以起始速度为 450pps, 运行
速度为 5000pps, 加速时间为 0.1s 正向回原点运动
while (Rt1000CheckDone(0)==0)//等待回原点动作完成
{
Rt1000SetCommandPos (1,0); //将当前第 1 轴位置清零，即设为第 1 轴原
点位置

```

## 4.2.6 指令脉冲计数

RT1000 控制卡每轴都有一个 24 位的指令脉冲计数器，记录了当前对应轴的绝对指令脉冲位置值。可以通过调用 Rt1000GetCommandPos 函数来读取该计数器的值，也可以通过调用 Rt1000SetCommandPos 函数来设置该计数器的值。具

体相关函数如表 4-9 所示：

表 4-9 指令脉冲计数相关函数说明

	名称	功能	参考
1	Rt1000GetCommandPos	读取指令位置计数器值	8.2.2.9
2	Rt1000SetCommandPos	设置指令位置计数器值	8.2.2.9

注：在点位运动中，每次可以输出的最大脉冲个数为16777215。

位置操作例程：

```
....
Rt1000SetCommandPos (1,100); //设置轴 1 的脉冲位置为 100
position = Rt1000GetCommandPos (1); //读取轴 1 的当前位置值至变量
position
....
```

## 4.2.7 通用 I/O 控制

RT1000 运动控制卡硬件上为用户提供了通用数字输入输出接口。用户可以使用这些数字 I/O 口用于检测开关信号、传感器信号等输入信号，或者输出继电器、指示灯等输出设备的控制信号。相关的函数如表 4-10 所示：

表 4-10 通用 IO 相关函数说明

	名称	功能	参考
1	Rt1000OutBit	通用输出口输出	8.2.2.10
2	Rt1000InBit	读取通用输入口状态	8.2.2.10
3	Rt1000GetOutbit	读取通用输出口状态	8.2.2.10

通用输入输出例程：

```
if (Rt1000InBit (1)==0) //读取 INPUT1 输入口的状态，判断按键是否按下
{
Rt1000OutBit (1, 0); //如果按键按下，OUT1 口输出为低电平，LED 发光
}
else
{
Rt1000GetOutbit(1, 1); //如果按键未按，OUT1 口输出为高电平，LED 不亮
}
....
```

## 5 驱动程序安装

首先打开“RT1000”整合的文件夹，在安装目录下，包含了开发 RT1000 所需的资料。Dll\_x64 文件夹里面的 RT1000.dll 适用于 64 位操作系统，Dll\_x86 文件夹里面的 RT1000.dll 适用于 32 位操作系统。

Motion\_x64、Motion\_x86 文件夹包含各平台下的测试程序。

名称	修改日期	类型	大小
DLL	2021/12/18 11:34	文件夹	
Motion	2021/12/18 11:30	文件夹	
RT1000 driver	2021/12/18 17:36	文件夹	
dotNetFx40_Full_x86_x64.exe	2021/7/26 12:02	应用程序	49,268 KB
RT1000 UM.pdf	2021/2/23 10:33	WPS PDF 文档	1,590 KB
RT1000_RunTimeFull.zip	2021/12/18 11:45	zip Archive	178,853 KB

图 5-3

在第一次使用 RT1000 卡时需要安装驱动所需的运行环境和动态库所需的运行环境，解压文件夹下的 RT1000\_RunTimeFull 文件，然后点击里面的安装选项，等待安装完成后再进行驱动程序安装。

名称	修改日期	类型	大小
bin	2021/12/18 11:41	文件夹	
license	2021/12/18 11:41	文件夹	
supportfiles	2021/12/18 11:43	文件夹	
dp.pmf	2021/12/18 11:43	PMF 文件	1 K
install.exe	2020/2/27 12:44	应用程序	5,327 K
install.ini	2021/12/18 11:43	配置设置	26 K
nidist.id	2021/12/18 11:43	ID 文件	1 K

### 5.1 在 Windows 7 操作系统环境中安装步骤

1. 关闭 PC 机并打开机箱，在 PCI 槽中插入 RT1000 运动控制卡。
2. 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。如图 5-5 所示：



图 5-5

3. 打开 RT1000 driver 文件夹，在“RT1000.inf”文件处，单击鼠标右键，选择“安装”，系统弹出如图 5-6 所示的安全提示对话框，请选择“始终安装此驱动程序软件(I)”，并继续安装；



图 5-6

5. 安装完成后，打开设备管理器，如图 5-7 所示，在“NI-VISA PXI Devices”选项下可以看到 RT1000 的驱动程序“RT1000”。至此，RT1000 控制卡就可以正常使用了。



图 5-7

## 5.2 在 Windows 10 操作系统环境中安装步骤

### 5.2.1 安装 RT1000.cer 认证文件

双击“RT1000.cer”文件，如图 5-8，点击“安装证书”

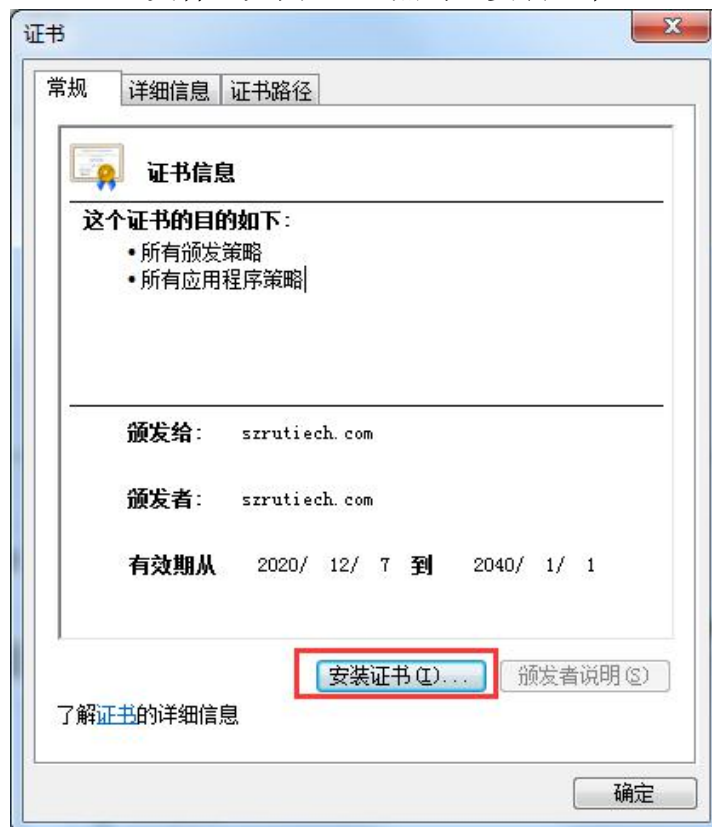


图 5-8



在弹出的对话框选择“下一步”



图 5-9

将证书安装到受信任的区域

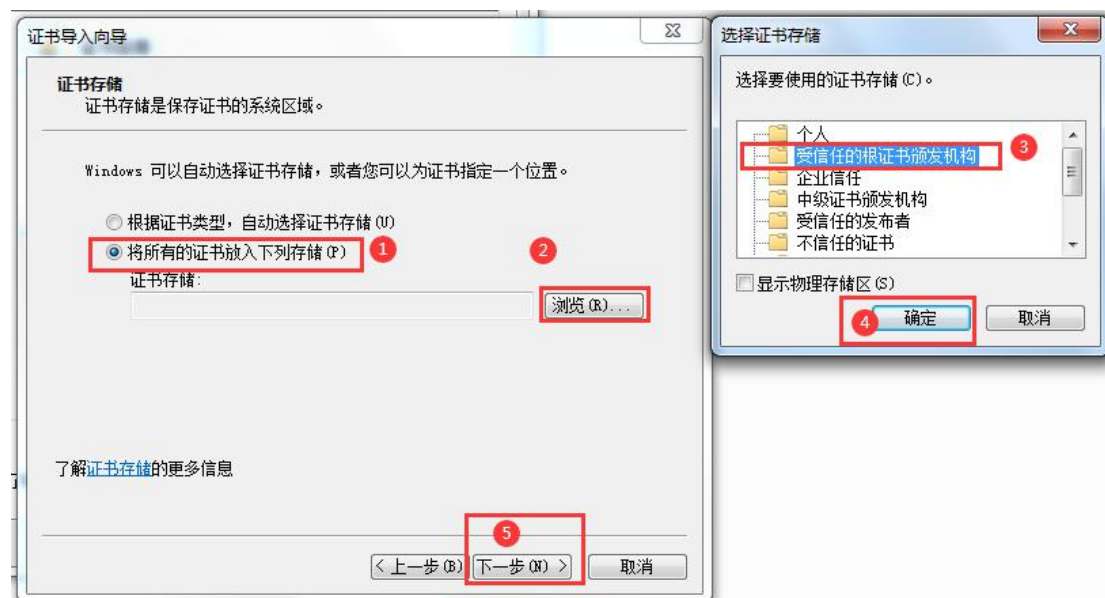


图 5-10

最后点击“完成”，如图 5-11

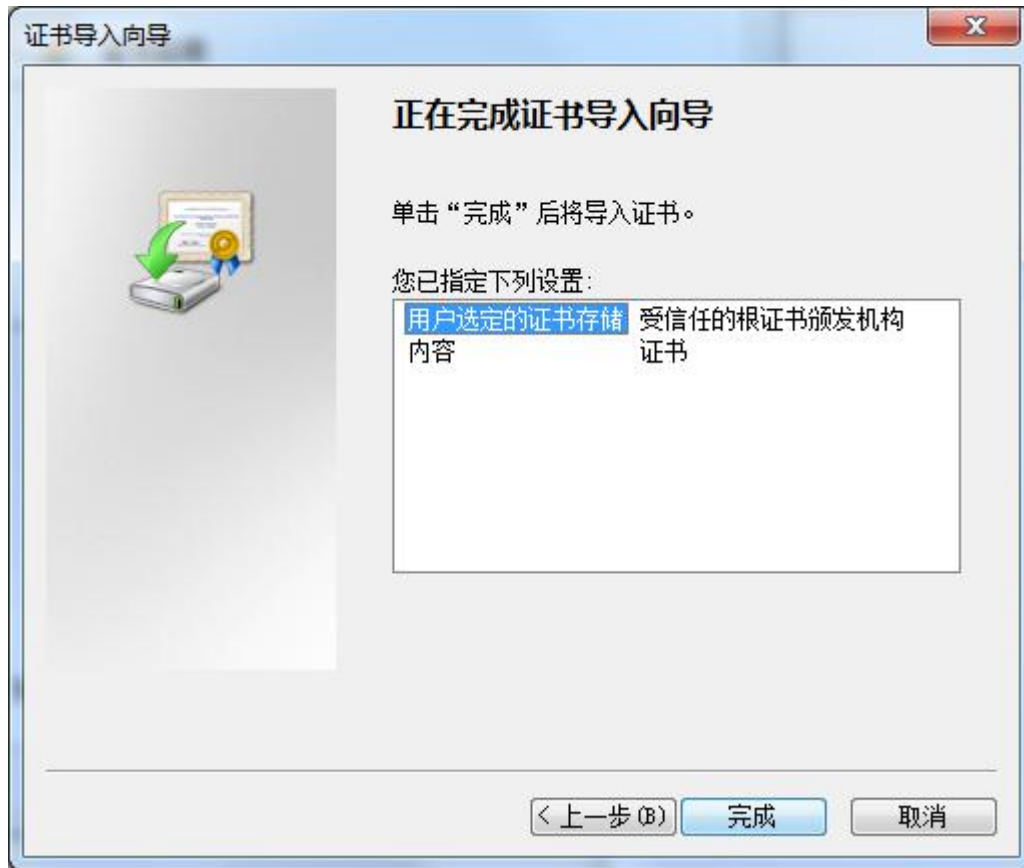


图 5-11

## 5.2.2 驱动安装

完成证书安装后，驱动安装方法同 5.1 中，windows7 系统的安装。

# 6 演示软件及使用

锐特公司为了方便用户尽快熟悉 RT1000 控制卡的运动功能和相关函数而配套提供的一个演示软件 Motion。利用这个软件，用户既可以很快地熟悉 RT1000 控制卡的软硬件功能，又可以方便快捷地测试电机驱动电路和系统的运动性能。

当 RT1000 运动控制卡的软硬件正确地安装到 PC 机上后，即可运行 Motion 软件，软件启动界面如图 6-1 所示。



图 6-1 Motion 软件启动界面

Motion 启动界面上有三个操作按钮，其功能如下：

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：回原点运动、连续运动、定长运动、直线插补运动等；

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

退出：停止当前所有对软件进行的操作，退出 Motion 演示软件。

## 6.1 I/O 检测演示

进入 IO 检测界面，如图 6-2 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。

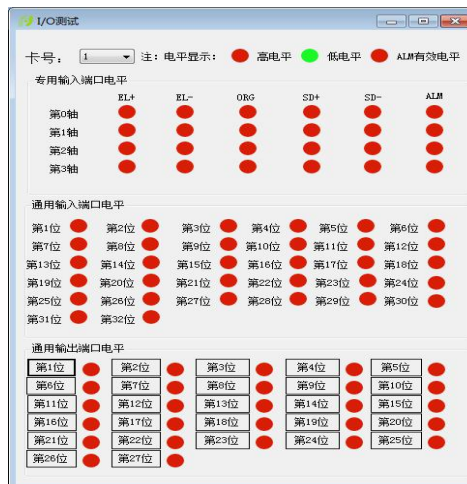


图 6-2 Motion I/O 测试界面

卡号选择：多卡运行时，用于选择要测试的卡号；

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。

## 6.2 运动测试演示

进入运动测试操作界面，如图 6-3 所示，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。



图 6-3 Motion 运动控制操作界面

操作界面介绍如下：

卡号选择：多卡运行时，选择控制卡卡号；

运动轴选择：选择运动控制的轴号；

单轴运动参数设置：对回原点、连续运动、定长运动各轴的运动参数进行设置；

运动方式选择：选择要演示的运动类型；

直线插补参数设置：对插补运动的参数进行设置；

位移方式选择：选择运动的位移方式；

速度曲线选择：选择运动的速度曲线；

脉冲设置：可以根据驱动器的具体要求设置脉冲输出类型以及脉冲有效电平；

减速信号设置：设置减速信号是否有效。

报警信号设置：设置报警信号有效电平。

各按钮功能如下：

加载参数：加载软件中上次保存的参数设置为当前设置，第一次运行会加载软件中默认保存的参数设置为当前设置；

保存参数：将当前界面中设置的参数信息保存到软件中；

启动：按照当前的设置进行运动；

停止：停止当前所有轴的运动；

位置清零：将“当前位置”中显示的各轴的位置设置为零。

## 6.3 函数测试演示



此功能便于用户快速查询并测试相关函数。

操作说明:

1. 添加待测函数: 选中左侧需要测试函数, 单击功能测试区 “-->”, 则将待测函数加入至测试区, 点击启动测试即可;
2. 删除测试函数: 选中功能测试文本框中的测试函数, 点击 “<---” 即可删除所选的测试函数。
3. 清空当前测试函数工作区: 点击 “清空”, 即可清除功能测试区全部测试函数。

# 7 用户系统开发

## 7.1 基于 windows 平台的应用软件结构

锐特 RT1000 控制卡的机器控制系统框架如图 7-1 所示：

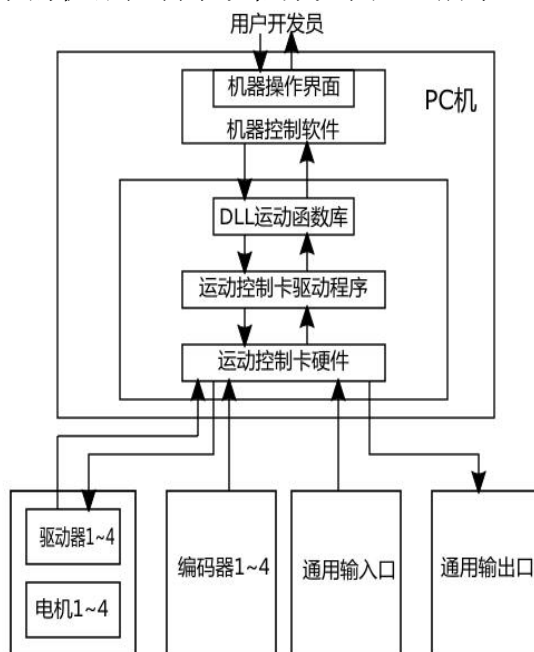


图 7-1 锐特控制卡的机器控制系统框架

从上面的示意图可以看出，控制系统的工作原理可以概述为：

1. 操作员的操作信息通过操作接口（包括显示屏和键盘）传递给机器控制软件；
2. 机器控制软件将操作信息转化为运动参数并根据这些参数调用 DLL 库中运动函数；
3. 运动函数调用运动控制卡驱动程序发出控制指令给控制卡；
4. 运动控制卡再根据控制指令发出相应的控制信号(脉冲、方向信号) 给电机驱动器；
5. 电机驱动器根据控制信号来驱动电机运动；
6. 电机的运动通过机械传动机构转化为机器的动作。

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。锐特公司已提供支持 RT1000 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 等程序语言调用这些函数来快速开发出自己的应用软件。

# 8 附录

## 8.1 硬件信号接口表

### 8.1.1 接口 X1 引脚定义

如图 8-1 所示，X1 是 RT1000 电机控制、I/O 信号控制的主要接口，为 SCSI-II 型 68 针插座。

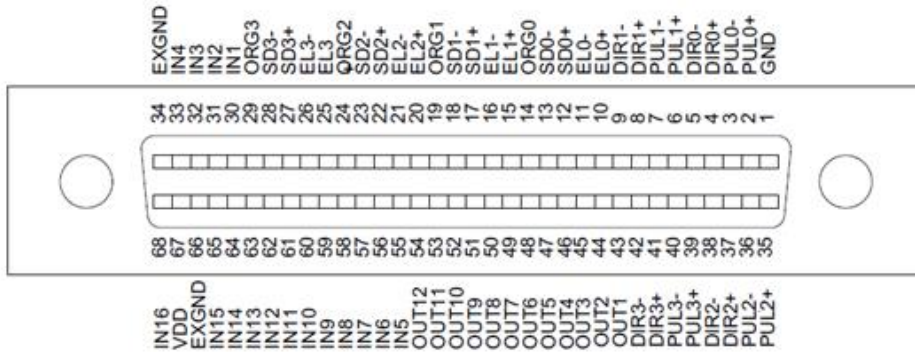


图 8-1 接口 X1 接口示例

其引脚号与信号的对应关系见表 8-1 所示：

表 8-1 接口 X1 引脚号与信号关系表

引脚	名称	说明	引脚	名称	说明
1	GND	脉冲和方向信号地	35	PUL2+	第 2 轴输出脉冲信号(+)
2	PUL0+	第 0 轴输出脉冲信号(+)	36	PUL2-	第 2 轴输出脉冲信号(-)
3	PUL0-	第 0 轴输出脉冲信号(-)	37	DIR2+	第 2 轴输出方向信号(+)
4	DIR0+	第 0 轴输出方向信号(+)	38	DIR2-	第 2 轴输出方向信号(-)
5	DIR0-	第 0 轴输出方向信号(-)	39	PUL3+	第 3 轴输出脉冲信号(+)
6	PUL1+	第 1 轴输出脉冲信号(+)	40	PUL3-	第 3 轴输出脉冲信号(-)
7	PUL1-	第 1 轴输出脉冲信号(-)	41	DIR3+	第 3 轴输出方向信号(+)
8	DIR1+	第 1 轴输出方向信号(+)	42	DIR3-	第 3 轴输出方向信号(-)
9	DIR1-	第 1 轴输出方向信号(-)	43	OUT1	隔离通用输出信号 1
10	EL0+	隔离第 0 轴正向限位信号(+)	44	OUT2	隔离通用输出信号 2
11	EL0-	隔离第 0 轴负向限位信号(-)	45	OUT3	隔离通用输出信号 3
12	SD0+	隔离第 0 轴正向减速信号(+)	46	OUT4	隔离通用输出信号 4
13	SD0-	隔离第 0 轴负向减速信号(-)	47	OUT5	隔离通用输出信号 5
14	ORG0	隔离第 0 轴原点信号	48	OUT6	隔离通用输出信号 6
15	EL1+	隔离第 1 轴正向限位信号(+)	49	OUT7	隔离通用输出信号 7

针脚	名称	说明	针脚	名称	说明
16	EL1-	隔离第1轴负向限位信号(-)	50	OUT8	隔离通用输出信号 8
17	SD1+	隔离第1轴正向减速信号(+)	51	OUT9	隔离通用输出信号 9
18	SD1-	隔离第1轴负向减速信号(-)	52	OUT10	隔离通用输出信号 10
19	ORG1	隔离第1轴原点信号	53	OUT11	隔离通用输出信号 11
20	EL2+	隔离第2轴正向限位信号(+)	54	OUT12	隔离通用输出信号 12
21	EL2-	隔离第2轴负向限位信号(-)	55	IN5	隔离通用输入信号 5
22	SD2+	隔离第2轴正向减速信号(+)	56	IN6	隔离通用输入信号 6
23	SD2-	隔离第2轴负向减速信号(-)	57	IN7	隔离通用输入信号 7
24	ORG2	隔离第2轴原点信号	58	IN8	隔离通用输入信号 8
25	EL3+	隔离第3轴正向限位信号(+)	59	IN9	隔离通用输入信号 9
26	EL3-	隔离第3轴负向限位信号(-)	60	IN10	隔离通用输入信号 10
27	SD3+	隔离第 轴正向减速信号(+)	61	IN11	隔离通用输入信号 11
28	SD3-	隔离第3轴负向减速信号(-)	62	IN12	隔离通用输入信号 12
29	ORG3	隔离第3轴原点信号	63	IN13	隔离通用输入信号 13
30	IN1	隔离通用输入信号1	64	IN14	隔离通用输入信号 14
31	IN2	隔离通用输入信号 2	65	IN15	隔离通用输入信号 15
32	IN3	隔离通用输入信号 3	66	EXGND	外部电源地, 输入
33	IN4	隔离通用输入信号 4	67	VDD	外部电源+24V, 输入
34	EXGND	外部电源地, 输入	68	IN16	隔离通用输入信号 16

注：当需要使用 IO 信号时，如限位、原点等专用信号或通用 IO 信号时，请务必在 67 脚和 66 脚输入+24VDC 电源（请不要将电源接反）。当要控制 IO 信号时，仍需 67 脚与 66 脚输入外部+24VDC 电源。

## 8.1.2 接口 X3 引脚定义

RT1000 运动控制卡中接口 X3 是多卡同时启动、同时停止、同时减速信号输入/输出口。其示意图如图 8-2 所示：

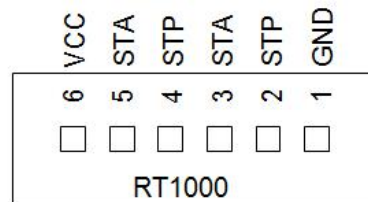


图 8-2 X3 接口示意图

其引脚号和引脚名定义见表 8-2 所示：

表 8-2 接口 X3 引脚号和信号关系表





脚号	名称	I/O	功 能	脚号	名称	I/O	功 能
14	IN30	I	非隔离通用输入信号30	33	OUT25	O	非隔离通用输出信号25
15	IN31	I	非隔离通用输入信号31	34	OUT26	O	非隔离通用输出信号26
16	IN32	I	非隔离通用输入信号32	35	OUT27	O	非隔离通用输出信号27
17	3.3V	O	PC电源, 输出	36	—		未用
18	3.3V	O	PC电源, 输出	37	GND	O	PC电源地, 输出
19	GND	O	PC电源地, 输出				

为方便用户使用 RT1000 的扩展 IO 口，锐特公司提供了专用的光电隔离接线板 RT-CNC37，使用 RT-CNC37 接线板后可实现 RT1000 扩展 IO 接口的输入输出信号全部光电隔离，其 RT-CNC37 接线板的引脚号与信号关系如表 8-4 所示：

表 8-4 RT-CNC37 引脚与 RT1000 转接口 J1 信号关系表

引脚号	信号名称	I/O	功 能	引脚号	信号名称	I/O	功 能
IN1	IN17	I	隔离后的通用输入17	OUT1	OUT13	O	隔离后的通用输出13
IN2	IN18	I	隔离后的通用输入18	OUT2	OUT14	O	隔离后的通用输出14
IN3	IN19	I	隔离后的通用输入19	OUT3	OUT15	O	隔离后的通用输出15
IN4	IN20	I	隔离后的通用输入20	OUT4	OUT16	O	隔离后的通用输出16
IN5	IN21	I	隔离后的通用输入21	OUT5	OUT17	O	隔离后的通用输出17
IN6	IN22	I	隔离后的通用输入22	OUT6	OUT18	O	隔离后的通用输出18
IN7	IN23	I	隔离后的通用输入23	OUT7	OUT19	O	隔离后的通用输出19
IN8	IN24	I	隔离后的通用输入24	OUT8	OUT20	O	隔离后的通用输出20
IN9	IN25	I	隔离后的通用输入25	OUT9	OUT21	O	隔离后的通用输出21
IN10	IN26	I	隔离后的通用输入26	OUT10	OUT22	O	隔离后的通用输出22
IN11	IN27	I	隔离后的通用输入27	OUT11	OUT23	O	隔离后的通用输出23
IN12	IN28	I	隔离后的通用输入28	OUT12	OUT24	O	隔离后的通用输出24
IN13	IN29	I	隔离后的通用输入29	OUT13	OUT25	O	隔离后的通用输出25
IN14	IN30	I	隔离后的通用输入30	OUT14	OUT26	O	隔离后的通用输出26
IN15	IN31	I	隔离后的通用输入31	OUT15	OUT27	O	隔离后的通用输出27
IN16	IN32	I	隔离后的通用输入32	OUT16	未用	—	未用
EGND	EXGND	I	外部电源地, 输入	EGND	EXGND	I	外部电源地, 输入
+24V	VDD	I	外部电源+24V, 输入	EGND	EXGND	I	外部电源地, 输入

## 8.2 运动控制函数

RT1000 运动控制函数库是一个运动控制 API 函数库，在其基础上进行应用软件开发的方法：使用 C/C++、Visual Basic 等开发用户界面，并调用 RT1000 函数库中的相关运动控制函数，就可以对多轴自动化设备进行精确、高速、协调

的控制。

RT100 运动控制卡的运动控制函数库共有 10 类 24 个函数，各函数分类列表如下：

## 8.2.1 函数列表

函数类别	函数名	描述
初始化函数	Rt1000BoardInit	初始化控制卡
	Rt1000BoardClose	关闭控制卡
脉冲模式设置函数	Rt1000SetPlsOutmode	设定脉冲输出模式
连续运动函数	Rt1000StartTvMove	以梯形速度曲线控制一个轴连续运动
	Rt1000StartSvMove	以 S 形速度曲线控制一个轴连续运动
	Rt1000GetSpeed	读取指定轴的脉冲输出速度
	Rt1000ChangeSpeed	改变指定轴的脉冲输出速度
	Rt1000ImmediateStop	以梯形或 S 形急停一个轴
	Rt1000DecelStop	以梯形或 S 形减速停止一个轴
单轴运动函数	Rt1000StartTMove	以梯形速度曲线控制相对坐标的点位运动
	Rt1000StartTaMove	以梯形速度曲线控制绝对坐标的点位运动
	Rt1000StartSMove	以S形速度曲线控制相对坐标的点位运动
	Rt1000StartSaMove	以S形速度曲线控制绝对坐标的点位运动
直线插补函数	Rt1000StartTLine	任意 2、3、4 轴相对坐标的直线插补运动
	Rt1000StartTaLine	任意 2、3、4 轴绝对坐标的直线插补运动
回原点函数	Rt1000HomeMove	回原点运动
运动状态检测函数	Rt1000CheckDone	检测当前运动状态
读取指令位置	Rt1000GetCommandPos	读取指令位置计数器值
取函数	Rt1000SetCommandPos	设置指令位置计数器值
通用IO接口函数	Rt1000OutBit	通用输出口输出
	Rt1000InBit	读取通用输入口状态
	Rt1000GetOutbit	读取通用输出口状态
专用 IO 接口函数	Rt1000SetSd	设置减速开关是否有效
	Rt1000GetAxisStatus	读取指定轴的所有输入状态

## 8.2.2 函数说明

### 8.2.2.1 初始化函数

DWORD Rt1000BoardInit (void)

功 能：为控制卡分配系统资源，并初始化控制卡。

参 数：无。

返回值：卡数：0~12，其中0表示没有卡。

DWORD Rt1000BoardClose (void)

功 能：关闭控制卡，释放系统资源。

参 数：无。

返回值：正确：返回 ERR\_NoError;

错误：返回相关错误码。

### 8.2.2.2 脉冲输出设置函数

DWORD Rt1000SetPlsOutmode (short axis, short pls\_outmode)

功 能：设置控制卡脉冲输出模式，用户可以根据驱动器具体接收脉冲的模式来选择控制卡的不同脉冲输出模式。

参 数：axis：轴号。范围0~(n×4-1)，n为卡数。

pls\_outmode：脉冲输出模式：

0：pulse/dir 模式，脉冲上升沿有效；

1：pulse/dir 模式，脉冲下降沿有效；

2：CW/CCW 模式，脉冲上升沿有效；

3：CW/CCW 模式，脉冲下降沿有效。

返回值：正确：返回 ERR\_NoError;

错误：返回相关错误码。

### 8.2.2.3 连续运动函数

DWORD Rt1000StartTvMove (short axis, long StrVel, Long MaxVel, double Tacc)

功 能：以梯形速度曲线控制指定轴至运行速度，并以运行速度连续运行。

参 数：axis：轴号，范围0~(n×4-1)，n为卡数。

StrVel：初始速度，单位：pps；

MaxVel：运行速度，单位：pps，其值的正负表示运动方向；

Tacc：加速时间，单位：s。

返回值：正确：返回 ERR\_NoError;

错误：返回相关错误码。

DWORD Rt1000StartSvMove (short axis, long StrVel, Long MaxVel, double

Tacc)

功 能：以 S 形速度曲线控制指定轴至运行速度，并以运行速度连续运行。

参 数：axis: 轴号，范围  $0\sim(n\times 4-1)$ ，n 为卡数；

StrVel: 初始速度，单位：pps；

MaxVel: 运行速度，单位：pps，其值的正负表示运动方向；

Tacc: 加速时间，单位：s。

返回值：正确：返回 ERR\_NoError；

错误：返回相关错误码。

DWORD Rt1000GetSpeed (short axis)

功 能：读取指定轴当前脉冲输出速度；

参 数：axis: 轴号，范围  $0\sim(n\times 4-1)$ ，n 为卡数。

返回值：指定轴当前的运动速度，单位 pps。

DWORD Rt1000ChangeSpeed (short axis, long NewVel)

参 数：axis: 轴号，范围  $0\sim(n\times 4-1)$ ，n 为卡数；

NewVel: 新设置的速度，单位：pps，取值范围：1~409550；

返回值：正确：返回 ERR\_NoError；

错 误：返回相关错误码。

**注意：**

Rt1000ChangeSpeed 函数变速后的速度与实际设定速度之间有一个允许的误差，误差值为  $0\sim 50$ pps，具体的误差值与当前速度及速度改变量  $\Delta V$  有关，其绝对值  $|\Delta V|$  越小，则误差越小。

DWORD Rt1000DecelStop (short axis)

功 能：减速停止指定轴脉冲输出。

参 数：axis: 轴号，范围  $0\sim(n\times 4-1)$ ，n 为卡数。

返回值：正确：返回 ERR\_NoError；

错误：返回相关错误码。

DWORD Rt1000ImmediateStop (short axis)

功 能：急停指定轴脉冲输出。

参 数：axis: 轴号：  $0\sim(n\times 4-1)$ ，n 为卡数。

返回值：正确：返回 ERR\_NoError；

错误：返回相关错误码。

## 8.2.2.4 点位运动函数

DWORD Rt1000StartTMove (short axis, long Dist, long StrVel, long MaxVel, double Tacc)

功 能：以梯形速度曲线控制指定轴至运行速度，并以相对坐标运行一定距离。

参 数: axis: 轴号, 范围  $0\sim(n\times 4-1)$ , n 为卡数。  
Dist: 相对运动距离, 单位: pulse, 其值的正负表示运动方向;  
StrVel: 初始速度, 单位: pps;  
MaxVel: 运行速度, 单位: pps;  
Tacc: 加速时间, 单位: s。  
返回值: 正确: 返回 ERR\_NoError;  
错误: 返回相关错误码。

DWORD Rt1000StartTaMove (short axis, long Pos, double StrVel, double MaxVel, double Tacc)

功 能: 以梯形速度曲线控制指定轴至运行速度, 并以绝对坐标运行一段指定距离。

参 数: axis: 轴号, 范围  $0\sim(n\times 4-1)$ , n 为卡数;  
Pos: 绝对运动位置, 单位: pulse;  
StrVel: 初始速度, 单位: pps;  
MaxVel: 运行速度, 单位: pps;  
Tacc: 加速时间, 单位: s。  
返回值: 正确: 返回 ERR\_NoError;  
错误: 返回相关错误码。

DWORD Rt1000StartSMove (short axis, long Dist, long StrVel, long MaxVel, double Tacc)

功 能: 以 S 形速度曲线控制指定轴至运行速度, 并以相对坐标运行一段指定距离;

参 数: axis: 轴号, 范围  $0\sim(n\times 4-1)$ , n 为卡数;  
Dist: 相对运动距离, 单位: pulse, 其值的正负表示运动方向;  
StrVel: 初始速度, 单位: pps;  
MaxVel: 运行速度, 单位: pps;  
Tacc: 加速时间, 单位: s。  
返回值: 正确: 返回 ERR\_NoError;  
错误: 返回相关错误码。

DWORD Rt1000StartSaMove (short axis, long Pos, long StrVel, long MaxVel, double Tacc)

功 能: 以 S 形速度曲线控制指定轴至运行速度, 并以绝对坐标运行一段指定距离。

参 数: axis: 轴号, 范围  $0\sim(n\times 4-1)$ , n 为卡数;  
Pos: 绝对运动位置, 单位: pulse;  
StrVel: 初始速度, 单位: pps;  
MaxVel: 运行速度, 单位: pps;  
Tacc: 加速时间, 单位: s。  
返回值: 正确: 返回 ERR\_NoError;  
错误: 返回相关错误码。

### 8.2.2.5 直线插补函数

DWORD Rt1000StartTLine (short TotalAxis, short \*AxisArray, short\*DistArray, long StrVel, long MaxVel, double Tacc)

功 能：启动多轴相对坐标的直线插补。

参 数：TotalAxis：插补轴数，范围 2~4；

\*AxisArray, AxisArray：轴号列表；

\*DistArray, DistArray：对应轴号列表各轴的相对坐标的距离列表

StrVel： 初始速度，单位：pps；

MaxVel： 运行速度，单位：pps；

Tacc： 加速时间，单位：s。

返回值：正确：返回 ERR\_NoError；

错误：返回相关错误码。

DWORD Rt1000StartTaLine(short TotalAxis, short \*AxisArray, short\*PosArray, long StrVel, long MaxVel, double Tacc)

功 能：启动多轴绝对坐标的直线插补。

参 数：TotalAxis：插补轴数，范围 2~4；

\*AxisArray, AxisArray：轴号列表；

\*PosArray, PosArray：对应轴号列表各轴的绝对坐标的位置列表；

StrVel： 初始速度，单位：pps；

MaxVel： 运行速度，单位：pps；

Tacc： 加速时间，单位：s。

返回值：正确：返回 ERR\_NoError；

错误：返回相关错误码。

### 8.2.2.6 回原点函数

DWORD Rt1000HomeMove (short axis, long StrVel, long MaxVel, double Tacc)

功 能：启动指定轴进行回原点运动。

参 数：axis：轴号，范围 0~(n×4-1)，n 为卡数。

StrVel： 回原点运动初始速度，单位：pps；

MaxVel：回原点运动速度，负值表示往负方向找原点，正值表示往正方向找原点，单位：pps；

Tacc：加速时间，单位：s。

返回值：正确：返回 ERR\_NoError；

错误：返回相关错误码。

### 8.2.2.7 运动状态检测函数

DWORD Rt1000CheckDone (short axis)

功 能：检测指定轴的运动状态。

参 数: axis: 轴号, 范围  $0 \sim (n \times 4 - 1)$ , n 为卡数。

返回值: 0: 正在运行;

1: 脉冲输出完毕停止;

2: 指令停止 (如调用了 Rt1000DecelStop 函数);

3: 遇限位停止;

4: 遇原点停止。

注 意: 1、当指定轴遇限位信号停止时, Rt1000CheckDone 函数返回 3, 若限位信号撤去, Rt1000CheckDone 函数返回值变为 1;

2、当指定轴遇原点信号停止时, Rt1000CheckDone 函数返回 4, 若原点信号撤去, Rt1000CheckDone 函数返回值变为 1。

### 8.2.2.8 指令位置设定和读取函数

DWORD Rt1000GetCommandPos (short axis)

功 能: 读取指令位置计数器计数值。

参 数: axis: 轴号, 范围  $0 \sim (n \times 4 - 1)$ , n 为卡数。

返回值: 指定轴当前指令位置计数器值, 单位: pulse。

DWORD Rt1000SetCommandPos (short axis, double Pos)

功 能: 设置指令位置计数器计数值。

参 数: axis: 轴号, 范围  $0 \sim (n \times 4 - 1)$ , n 为卡数。

Pos: 设置指令位置计数器值, 单位: pulse。

返回值: 正确: 返回 ERR\_NoError;

错误: 返回相关错误码。

### 8.2.2.9 通用 I/O 接口函数

DWORD Rt1000OutBit (short BitNo, short BitData)

功 能: 输出通用输出信号。

参 数: BitNo: 表示要输出的通用输出出口的位号, 范围参考表 8-5 及表 8-6;

BitData: 输出信号: 0 - 表示低电平;

1 - 表示高电平。

返回值: 正确: 返回 ERR\_NoError;

错误: 返回相关错误码。

注: 多卡 IO 位号分配说明, 如表 8-5 和 8-6 所示:

表 8-5 RT1000 多卡 IO 位号分配

	通用输出口	通用输入口
卡1	1~27	1~32
卡2	32+(1~27)	32+(1~32)
卡3	64+(1~27)	64+(1~32)
卡n	$(n-1)*32+(1~27)$	$(n-1)*32+(1~32)$



表 8-6 RT1000 多卡 IO 位号分配

	通用输出口	通用输入口
卡1	1~12	1~16
卡2	12+(1~12)	16+(1~16)
卡3	24+(1~12)	32+(1~16)
卡n	(n-1)*12+(1~12)	(n-1)*16+(1~16)

**DWORD Rt1000InBit (short BitNo)**

功 能：读取通用输入信号状态。

参 数：BitNo: 表示要读取的通用输入口的位号，范围参考表 8-5 及表 8-6;

返回值：输入口状态：0 -表示低电平;

1 -表示高电平。

**DWORD Rt1000GetOutbit (short BitNo)**

功 能：读取通用输出信号状态。

参 数：BitNo: 通用输出口位号，范围参考表 8-5 及表 8-6;

返回值：输出口状态：0 - 表示低电平;

1 - 表示高电平。

### 8.2.2.10 专用 I/O 接口函数

**DWORD Rt1000SetSd (short axis, short SdMode)**

功 能：设置减速信号是否使能。

参 数：axis: 轴号，范围 0~(n×4-1)，n 为卡数。

SdMode: 减速使能模式

0: SD 信号无效;

1: SD 信号有效。

返回值：正确：返回 ERR\_NoError;

错误：返回相关错误码。

**DWORD Rt1000GetAxisStatus (short axis)**

功 能：读取指定轴的专用接口信号状态，包括 EL+、EL-、STP、STA、SD+、SD-等信号状态。

参 数：axis: 轴号，范围 0~(n×4-1)，n 为卡数。

返回值：指定轴专用信号接口状态，取低字节，其二进制位号与信号的对应关系如表 8-7 所示：

表 8-7 返回值的位号与信号对应表

位 号	信 号	电 平
0	-EL	0: 高, 1: 低
1	+EL	0: 高, 1: 低
2	ORG	0: 高, 1: 低

3	STP	0: 高, 1: 低
4	STA	0: 高, 1: 低
5	-SD	0: 高, 1: 低
6	+SD	0: 高, 1: 低
7	保留	保留