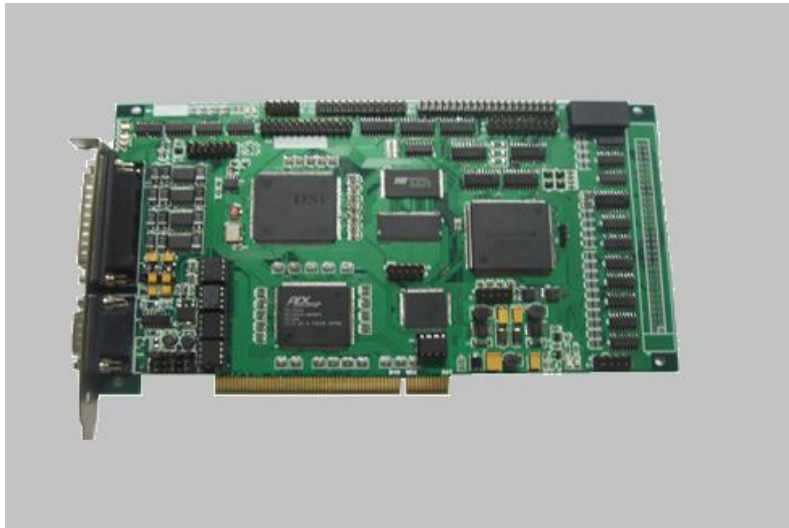




用户在使用控制器之前请务必阅读该手册

- 本手册内容是电气操作手册和编程手册
- 先仔细的阅读电气手册,保证正确电气连接

RDS3040G 通用运动控制器 软件开发手册



深圳市睿达科技有限公司

地 址: 深圳市南山区登良中路天安南油工业区 2
座 501、502

电 话: 0755--26066687

传 真: 0755--26982287

电子邮件: sales@rd-ac.com

网 址: www.rd-ac.com

RuiDa Technology Co., Ltd

Addr: 7TH FLOOR, 5TH BUILDING, NANYOU
TIANAN INDUSTRIAL ZONE, SHENZHEN

Tel: 0755--26066687

Fax: 0755--26982287

E-mail: sales@rd-ac.com

Web: www.rd-ac.com

睿达声明

声明一：

睿达科技具有本产品的专利版权和知识产权。

未经睿达科技授权，不得直接或者间接地复制、制造、加工、使用本产品及其相关部分。
否则睿达科技将依法追究相关的法律责任。

声明二：

深圳市睿达科技有限公司保留在不事先通知的情况下，修改本手册中的产品和产品规格等文件的权力，但睿达科技定期的发布有关更新内容供用户查阅。

声明三：

睿达科技不承担以下直接的或间接的责任：

- 用户使用本手册或本产品不当
- 用户没有遵循相关的安全操作规程
- 自然力造成的损失



注意

危险！

本控制器的应用中牵涉到激光、旋转或者直线运动机械机构，虽然控制器中已经具有一些保护机制以及状态反馈机制供用户使用，但仍然存在一定的危险性，由于用户使用不当造成的损失，睿达科技不承担相关责任！

前言

技术支持

您可以通过以下途径获得我们的技术支持和售后服务：

电子邮件：support@rd-acis.com

电话：(0755)26982287

发函至：深圳市南山蛇口南海大道北科创业大厦 310

手册用途

本手册为 **RDS3040G-PCI** 通用运动控制器软件开发包使用说明，指导用户开发应用系统。

适用对象

本手册适用于对运动控制系统有一定的基础，能熟练使用编程语言进行系统开发的软件系统开发工程师。

手册内容

本手册的内容分两部分，第一部分是本控制器的电气定义和连接指南，第二部分是开发包函数说明和编程指南。

使用说明

本手册的函数动态连接库采用 VC++ 编写。如果用户需要其它编程语言下的函数动态连接库，请与睿达科技联系。

相关文件

《RDS3040G-PCI 通用运动控制器应用手册》

目 录

睿达申明	1
前言	2
目 录	3
第一篇 电气手册	4
第一章 概述	7
1.1 简介	7
1.2 通用运动控制器型号及含义	7
1.3 通用运动控制器功能列表	8
第二章 使用	10
2.1 产品清单	10
2.2 通用控制器外形结构	10
2.3 系统安装与应用步骤	12
2.3.1 步骤1 将控制卡插入计算机	12
2.3.2 步骤2 驱动程序安装	12
2.3.3 步骤3 控制卡运行测试	13
2.3.4 步骤4 连接运动控制卡和扩展电机接口板	13
第二篇 软件开发手册	22
第一章 运动控制其函数库的使用	22
1.1 快速应用动态连接库	22
第二章 函数执行监测	23
2.1 函数返回值	23
2.1 指令错误寄存器	24
第三章 初始化通用运动控制卡	25
3.1 板卡初始化	25
3.1.1 初始化函数	25
3.1.2 重点说明	25
第四章 通用运动控制卡状态检测	27
4.1 函数列表	27
4.2 函数说明	27

第五章 轴参数控制	29
5.1 限位开关参数设置	29
5.2 编码器参数设置	31
5.3 报警输入设置	36
5.4 脉冲输出参数设置	39
5.5 轴控制模式设置	39
5.6 轴控制指令	39
第六章 坐标系参数设置	44
6.1 坐标系定义	44
6.2 设置坐标系参数	44
6.3 例程	44
第七章 运动模式	45
7.1 连续轨迹运动	45
7.2 立即模式下的直线插补	45
7.3 停止坐标系运动	45
7.4 连续轨迹插补	45
7.5 位置捕获	45
7.6 限位状态处理	45
7.7 控制轴驱动报警处理	45
第八章 输入输出	47
8.1 函数列表	47
8.2 函数说明	47
8.3 辅助编码器接口	49
第九章 辅助功能	51
第九章 激光控制	51
第十章 函数列表	52
第十一章 函数详细说明	53

第一篇

电气手册

第一章 概述

1.1 简介

基于 PCI 接口的 RDS3040G-PCI 是一款通用的 4 轴运动控制器。该卡以高性能的浮点的 DSP 和高密度的 FPGA 为硬件平台，结合优化的软件算法，适用于运动控制控制的多种场合。

RDS3040G-PCI 具有完善的运动控制功能，支持 4 轴直线插补、空间圆弧插补、样条插补功能。具有开环控制和闭环控制功能。同时还支持龙门轴控制功能。

1.2 通用运动控制器型号及含义

RDS3040G-PCI

编号	状态
RD	睿达科技标志
S	S: 多轴协调运动控制 P: 点位控制 T: 特殊控制器
3	硬件平台版本
0	硬件平台扩展版本
4	轴数
0	0: 通用功能 X: 特殊定制功能
PCI	PCI 接口卡

1.3 通用运动控制器功能列表

功能	RDS3040G-PCI
模拟量输出	4 路
限位信号输入	4 轴，每轴正负限位，光隔输入
原点信号输入	4 轴，每轴一路原点信号输入，光隔输入
驱动报警信号输入	4 轴，每轴 1 路，光隔输入
驱动使能信号输出	4 轴，每轴 1 路，光隔输入
驱动复位信号输出	4 轴，每轴 1 路，光隔输入
通用数字信号输入	16 路，光隔输入
通用数字信号输出	16 路，光隔输出
A/D	8 路，为可选功能
高速 IO 输出	2 路，差分输出，光电隔离
脉冲量输出	4 路，差分输出
方向信号输出	4 路，差分输出
2 路高速 IO 输出	光电隔离，正负极性可选

第二章 使用

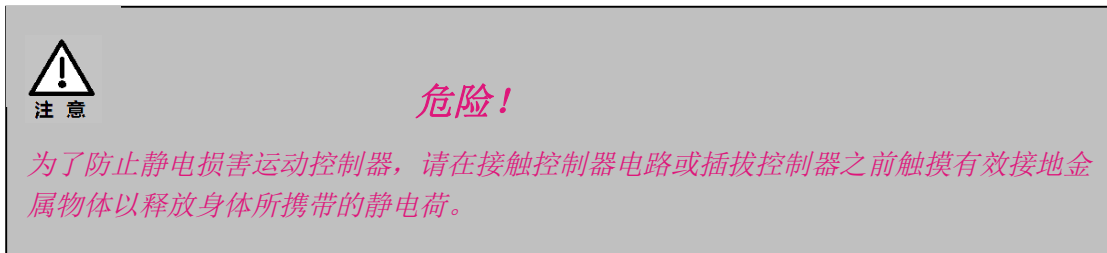
2.1 产品清单

收到产品后，用户应仔细的核对产品包装的信号是否和订购的产品型号一致，如果不一致，请不要打开包装，并与睿达科技销售部门联系。如果型号相符，用户可以打开包装，打开包装后，由于控制器的电路板上有很多静电敏感的器件，为了防止静电造成的危害，请佩戴好静电手套，然后仔细的检查板卡有无损坏，如果有损坏，请与睿达科技联系。

请妥善的保管好通用运动控制卡的包装，以便在返修的时候用原包装返回睿达科技，避免损坏。

通用运动控制器产品清单：

- RDS3040G-PCI 控制卡 1 块；
- 扩展端子板 1 块；
- 68pin 连接电缆 1 条；
- 配套说明书及光盘 1 张。
- 防静电手套 1 只



2.2 通用运动控制器的外形结构

PCI 系列通用运动控制器的外形结构如图 2-1 所示：

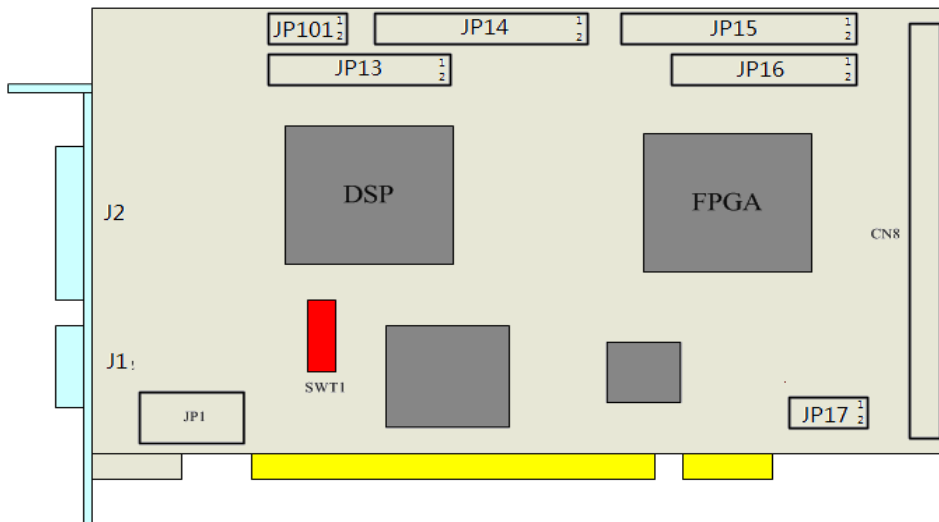


图 2-1 RDS3040G-PCI 外形图

表 2-1 拨码开关 SWT1 默认状态

编号	状态
1	OFF
2	OFF
3	OFF
4	OFF
5	OFF
6	OFF

J2 为外部扩展控制接口。

J1 为高速 IO 输出接口，

JP101 为扩展功能

JP13 为扩展功能

JP14 为扩展功能

JP15 为扩展功能

JJP16 为扩展功能

JP17 为扩展编码器输入接口

JP11,JP12,JP20 为高速 IO 输出选择端子

和 RDS3040G-PCI 控制器配套的扩展端子板外形结构如图 2-2 所示：

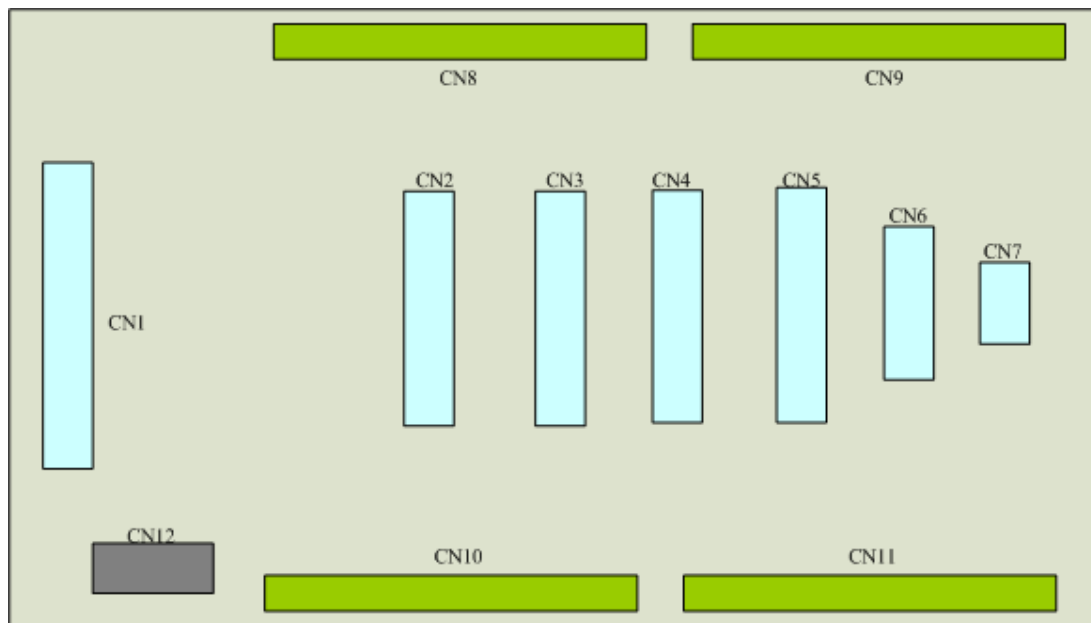


图 2-2: RDS3040G-PCI 扩展板结构图

各个端子的功能定义如下表 2-2 所示。

表 2-2 端子定义表

端子引脚	信号定义
CN1	和主卡的连接端子
CN2	电机轴接口
CN3	电机轴接口
CN4	电机轴接口
CN5	电机轴接口
CN6	模拟量输入接口
CN7	扩展高速 I/O 输出
CN8	专用 I/O 输入
CN9	通用 I/O 输入
CN10	通用 I/O 输出
CN11	通用 I/O 输出
CN12	电源接口

2.3 系统安装与应用步骤

请按照以下安装步骤建立控制系统。

2.3.1 步骤 1: 将运动控制器插入计算机



危险!

为了防止静电损害运动控制器，请在接触控制器电路或插拔控制器之前触摸有效接地金属物体以释放身体所携带的静电荷。

1. 关断计算机电源，确保已经拔掉交流输入的插头。
2. 佩戴好防静电手套
3. 打开计算机机箱，将通用运动控制器可靠地插入该槽。
4. 拧紧其上的固定螺丝。
5. 打开 PC 电源，启动计算机，此时如果板卡左上角指示灯闪烁，表示通用运动控制器工作正常
6. 此时关闭计算机，断开电源
7. 连接其它控制设备到通用运动控制卡，如激光器，电机等
8. 再次打开 PC 电源，系统可正常工作

2.3.2 步骤 2: 驱动程序安装

本控制器适用于 Windows XP 系统,如果是其它的操作系统,请与睿达科技技术支持联系。

1. WINDOW 操作系统启动后,系统会自动地检测到 PCI 的设备,并提示安装驱动选择“从列表或指定位置安装”选项,点击“下一步”。
2. 在该页面下,选择“在这些位置上搜索最佳驱动程序”,并选择“在搜索中包含这个位置”,点击“浏览”,进入文件选择界面。
3. 将包含有产品驱动程序的光盘放入光驱。
4. 利用“浏览”选择“光驱:\driver\win xp”下相应操作系统的目录。
5. 跟随“添加硬件向导”点击“下一步”,直到完成。
6. 此时查看计算机系统的设备属性里,会有一个“RDDriver”的新设备。点击左侧“+”,显示“RuiDaTech RD400SCAN Ver 1.0”。

2.3.3 步骤 3: 通用运动控制卡运行测试

首先观测控制卡的指示灯是否正常闪动,如果不闪,则说明出现了故障,请与睿达科技联系。其次确认已经正确安装了 PCI 卡的驱动程序,则接下来要进行板卡和 PCI 的通讯测试。

用户可以调用软件开发包的函数进行测试板卡的通讯是否正常。请参考第 2 篇的“软件开发手册”的指导进行。

在**通讯成功**的前提下,用户可进行通用控制程序的开发,否则参考《附录 故障处理》,确定问题所在,排除故障后重新测试。如果需要,请按照封面的公司信息与我们联系。

2.3.4 步骤 4: 联接运动控制卡和电机控制扩展板

如果用户不需要扩展端子板,则直接把控制卡插入计算机 PCI 插槽中即可。如果需要电机扩展端子板,则需要通过 68 引脚的连接电缆把控制卡和扩展端子板连接起来。

电机控制接口板的各个接口的定义如下:

● 电机控制接口

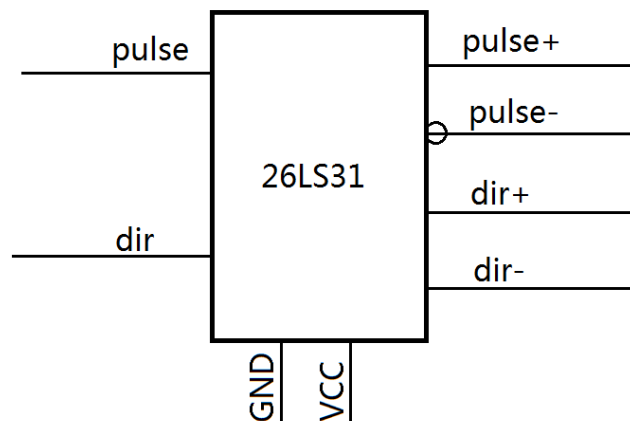


图 2-3 脉冲差分输出信号示意图

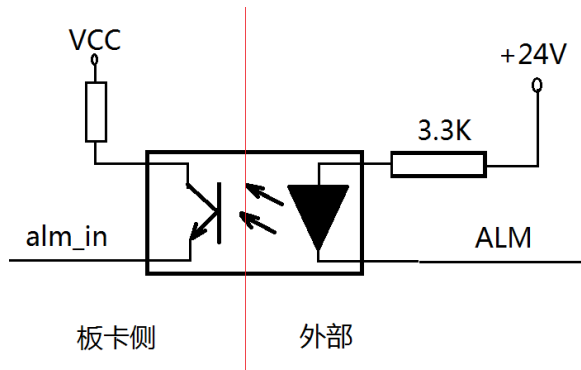


图 2-4 伺服报警信号输入示意图

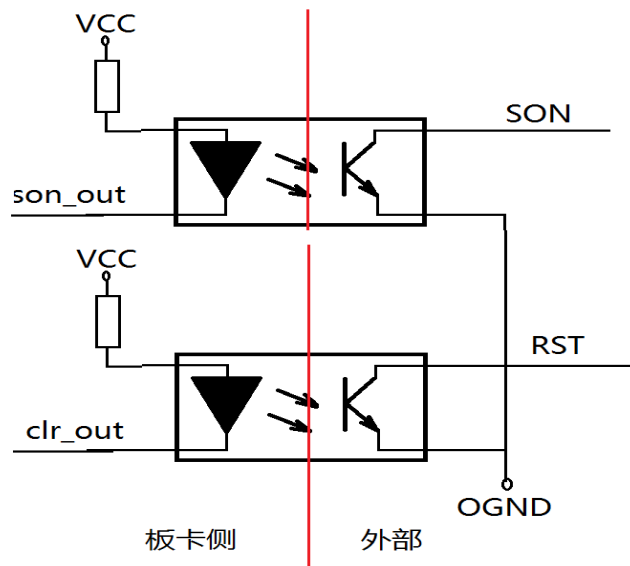


图 2-5 伺服使能和伺服复位输出信号示意图

表 2-3 CN2~CN5 信号定义

引脚	信号	说明	引脚	信号	说明
PIN1	OGND	外部电源地	PIN14	0VCC	外部电源
PIN2	ALM	伺服报警输入	PIN15	CLR	伺服复位信号输出
PIN3	SON	伺服使能输出	PIN16	NC	悬空
PIN4	A0-	编码器 A 负相输入	PIN17	A0+	编码器 A 正相输入
PIN5	B0-	编码器 B 负相输入	PIN18	B0+	编码器 B 正相输入
PIN6	C0-	编码器 C 负相输入	PIN19	C0+	编码器 C 正相输入
PIN7	+5V	内部+5V 输出	PIN20	GND	内部参考地
PIN8	Vout	伺服速度指令输出	PIN21	GND	内部参考地
PIN9	DIR+	方向信号正相输出	PIN22	DIR-	方向信号反相输出
PIN10	AGND	伺服速度输出参考地	PIN23	PULSE+	脉冲信号正相输出
PIN11	PULSE-	脉冲信号反相输出	PIN24	GND	内部参考地
PIN12	NC	悬空	PIN25	NC	悬空
PIN13	GND	内部参考地			

● 高速隔离 IO 输出接口

高速隔离的 IO 输出采用高速光耦 6N137 隔离。

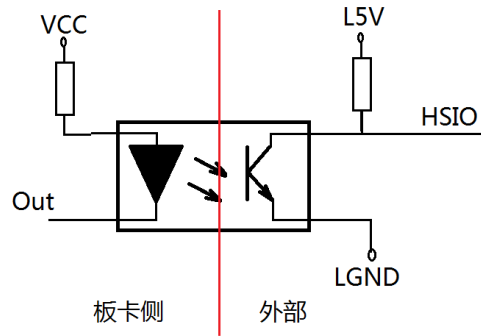


图 2-6 高速 io 输出示意图

表 2-4 CN7 信号定义

引脚	信号	说明
PIN1	0GND	外部电源地
PIN2	HSI01+	高速 I01 正相输出
PIN3	HSI02+	高速 I02 正相输出
PIN4	NC	编码器 A 负相输入
PIN5	+5V	编码器 B 负相输入
PIN6	05V	编码器 C 负相输入
PIN7	HSI01-	高速 I01 反相输出
PIN8	HSI02-	高速 I02 反相输出
PIN9	GND	内部地

● 通用 IO 输出

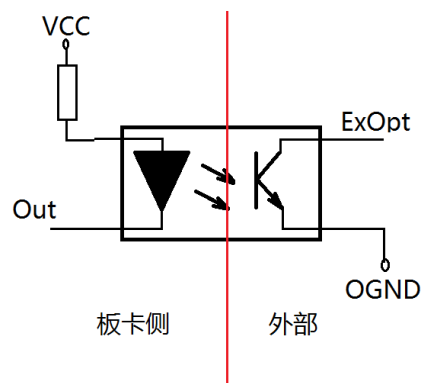


图 2-7 通用 IO 光电隔离输出示意图

表 2-5 CN10 通用 IO 输出定义

引脚	信号	说明	引脚	信号	说明
PIN1	ExOut0	通用 IO 输出 0	PIN9	ExOut8	通用 IO 输出 8
PIN2	ExOut1	通用 IO 输出 1	PIN10	ExOut9	通用 IO 输出 9
PIN3	ExOut2	通用 IO 输出 2	PIN11	ExOut10	通用 IO 输出 10
PIN4	ExOut3	通用 IO 输出 3	PIN12	ExOut11	通用 IO 输出 11
PIN5	ExOut4	通用 IO 输出 4	PIN13	ExOut12	通用 IO 输出 12
PIN6	ExOut5	通用 IO 输出 5	PIN14	ExOut13	通用 IO 输出 13
PIN7	ExOut6	通用 IO 输出 6	PIN15	ExOut14	通用 IO 输出 14
PIN8	ExOut7	通用 IO 输出 7	PIN16	ExOut15	通用 IO 输出 15

表 2-6 CN11 通用 IO 输出定义

引脚	信号	说明	引脚	信号	说明
PIN1	ExOut16	通用 IO 输出 16	PIN9	ExOut24	通用 IO 输出 24
PIN2	ExOut17	通用 IO 输出 17	PIN10	ExOut25	通用 IO 输出 25
PIN3	ExOut18	通用 IO 输出 18	PIN11	ExOut26	通用 IO 输出 26
PIN4	ExOut19	通用 IO 输出 19	PIN12	ExOut27	通用 IO 输出 27
PIN5	ExOut20	通用 IO 输出 20	PIN13	0VCC	外部电源输出
PIN6	ExOut21	通用 IO 输出 21	PIN14	0VCC	外部电源输出
PIN7	ExOut22	通用 IO 输出 22	PIN15	0GND	外部电源地
PIN8	ExOut23	通用 IO 输出 23	PIN16	0GND	外部电源地

● 通用 IO 输入

通用 IO 输入采用光电隔离输入，输入侧的电源标准为+24V。

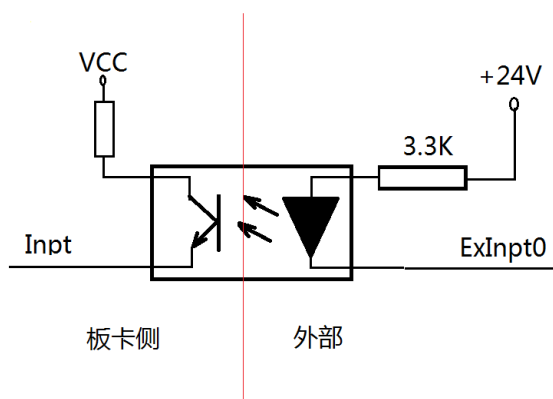


图 2-8 通用 IO 光电隔离输入示意图

表 2-7 CN9 通用 IO 输入定义

引脚	信号	说明	引脚	信号	说明
PIN1	ExIn0	通用 IO 输入 0	PIN9	ExIn8	通用 IO 输入 8
PIN2	ExIn1	通用 IO 输入 1	PIN10	ExIn9	通用 IO 输入 9
PIN3	ExIn2	通用 IO 输入 2	PIN11	ExIn10	通用 IO 输入 10
PIN4	ExIn3	通用 IO 输入 3	PIN12	ExIn11	通用 IO 输入 11
PIN5	ExIn4	通用 IO 输入 4	PIN13	ExIn12	通用 IO 输入 12
PIN6	ExIn5	通用 IO 输入 5	PIN14	ExIn13	通用 IO 输入 13
PIN7	ExIn6	通用 IO 输入 6	PIN15	ExIn14	通用 IO 输入 14
PIN8	ExIn7	通用 IO 输入 7	PIN16	ExIn15	通用 IO 输入 15

● 专用 IO 输入

专用输入采用光电隔离输入，示意图如下。

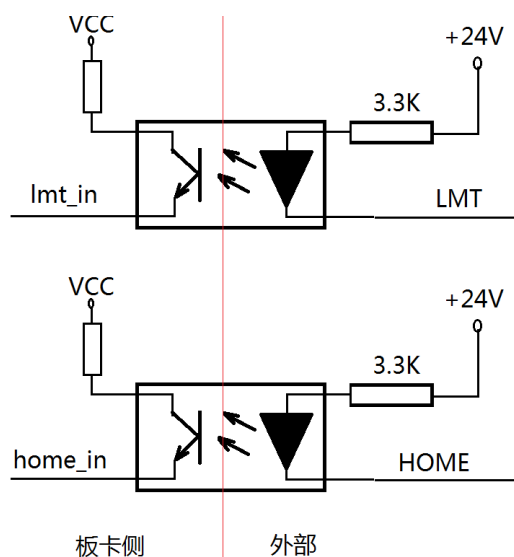


图 2-8 专用信号输入示意图

表 2-8 CN8 专用 IO 输入定义

引脚	信号	说明	引脚	信号	说明
PIN1	HOME0	第 1 轴原点开关信号输入	PIN9	LMT2+	第 3 轴正限位开关信号输入
PIN2	HOME1	第 2 轴原点开关信号输入	PIN10	LMT2-	第 3 轴负限位开关信号输入
PIN3	HOME2	第 3 轴原点开关信号输入	PIN11	LMT3+	第 4 轴正限位开关信号输入
PIN4	HOME3	第 4 轴原点开关信号输入	PIN12	LMT3-	第 4 轴负限位开关信号输入
PIN5	LMT0+	第 1 轴正限位开关	PIN13	0VCC	外部电源输出

		信号输入			
PIN6	LMT0-	第 1 轴负限位开关 信号输入	PIN14	0VCC	外部电源输出
PIN7	LMT1+	第 2 轴正限位开关 信号输入	PIN15	0GND	外部电源地
PIN8	LMT1-	第 2 轴负限位开关 信号输入	PIN16	0GND	外部电源地

第二篇

软件开发手册

第一章 运动控制器函数库的使用

RDS3040G-PCI 通用运动控制器需要一个具有 PCI 插槽的 PC 兼容机或工业控制计算机，操作系统为 windows XP 系统。

RDS3040G-PCI 通用运动控制器配套附件包括下面的内容：

- 动态链接库： rds.lib, rds .dll , rds.h。
- RDS3040G-PCI 控制器的 Windows 驱动程序。
- 《软件开发手册》

想了解 RDS3040G-PCI 通用运动控制器的详细的版本信息，可以调用相关的函数获得控制器的版本号。

`RD_ScanGetDSPVer ()` 获得 DSP 软件的版本

`RD_ScanGetHdVer ()` 获取硬件版本号

RDS3040G-PCI 通用运动控制器提供 Windows 下的动态链接库。用户只要调用函数库中的指令，就可以实现运动控制器的各种功能。下面讲述 Windows 系统下函数库的使用方法。

1.1 调用动态链接库

在 WindowsXP 系统下使用 RDS3040G-PCI 通用运动控制器，首先要安装驱动程序，RDS3040G-PCI 通用运动控制器的驱动程序存放在产品配套光盘的 Drive\win xp 文件夹下。必须按照正确的步骤安装驱动程序，并保证驱动程序的正确性，驱动程序的安装在本手册第一篇里有详细的描述。

通用运动控制器指令函数动态链接库存放在产品配套光盘的 Windows\DLL\文件夹下。RDS3040G-PCI 通用运动控制器的动态链接库文件名为 rds .dll, rds .lib, rds.h。

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

1. 启动 Visual C++，新建一个工程；
2. 将产品配套光盘 Windows\VC 文件夹中的 rds .dll, rds .lib, rds.h 复制到工程文件夹中；
3. 把 rds .lib 添加到工程中。
4. 在应用程序文件中加入函数库头文件的声明， `#include "rdsc.h"`
5. 用户就可以在 Visual C++ 中调用函数库中的任何函数，开始编写应用程序。

第二章 函数执行监测

2.1 函数返回值

RDS3040G-PCI 通用运动控制器按照主机发送的指令工作。RDS3040G-PCI 通用运动控制器指令封装在动态链接库中。用户在编写应用程序时，通过调用动态库里的控制函数，从而把相关的控制指令发送到 PCI 卡中。为了监测每个指令的执行情况，必须监测每个函数的返回值，如果返回值为 0，则表示该函数被正确执行，如果返回值不为 0，则表示函数执行出错，一旦出错，主机都要判断出错原因，并成功发送该指令。

RDS3040G-PCI 通用运动控制器在接收到主机发送的指令时，每一条运动控制指令都具有返回值，将执行结果反馈到主机，指示当前指令是否正确执行。指令返回值的定义如表 2-1 所示：

表 2-1 RDS3040G-PCI 通用运动控制器指令返回值定义

返回值	意义	处理方法
-3	指令重入	不要在多个线程当中同时调用运动控制器指令。
-1	通讯出错	检查运动控制器是否工作正常。
0	指令执行成功	继续。
1	指令错误	调用 RD_GetCmdSts 指令，确定出错原因，予以改正。 一般为缓冲区满
2	保留	
3	保留	直线位移位置点相同，应过滤掉该点。
4	保留	
8	不支持的控制指令	和睿达科技联系
7	指令参数错误	检查该指令的参数是否正确，改正后重发此指令。
10	打开控制器出错	可能和操作系统有关



注意

注意

建议在用户程序中，检测每条指令的返回值，以判断指令的执行状态。
并建立必要的错误处理机制，保证程序安全可靠地运行。

如果指令返回值为-1，而且重复调用该指令仍返回-1，说明运动控制器的通讯出现故障，运动控制器没有正确地接收主机的指令；或是运动控制器工作不正常，不能正确处理主机指令。此时请按照如下步骤进行检查：

1. 在 Windows 系统下使用运动控制器时，请检查驱动程序是否正确安装；
2. 检查运动控制器、接插件和连接线是否插牢，清洗板卡金手指。
3. 检查 PC 机的插槽是否能够正常工作，并尝试更换插槽或 PC 机；

2.2 指令错误寄存器

如果函数返回值为 1，则可以调用 RD_ScanGetCmdSts 获得指令寄存器的状态，获取出错的原因，则寄存器的状态为 0，有错误发生，该位置 1。定义如表 2-2 所示：

表 2-2 RDS3040G-PCI 通用运动控制器指令错误寄存器定义

位	定义
0	保留
1	轴参数非法
2	保留
3	控制轴状态错误， （1）控制轴处于“伺服使能”状态，调用函数 RD_RsrDrv()复位对应轴的驱动器， （2）控制轴处于“伺服使能”状态，调用函数 RD_SetAlmOn(),RD_SetAlmOff(), RD_EncOn(),RD_EncOff() （3）伺服驱动器产生报警时，调用函数 RD_SrvON()
4	保留
5	设置捕获状态出错 用户重复调用捕获设置函数， （1）控制轴已经设置了捕获，但还没有捕获到信息，用户再次调用了捕获设置函数 （2）控制轴已经设置了捕获，已经捕获到信息，但用户没有清除捕获状态，再次调用了捕获设置函数
6	控制轴状态出错 （1）当控制轴处于开环控制模式时，而调用了闭环控制的相关函数 （2）当控制轴处于闭环控制模式时，调用了开环控制的相关函数 （3）当调用 RD_MapCnt()函数时，相关轴处于运动状态
7	保留
8	（1）当前立即指令运动没有停止，就调用另外一条立即指令 （2）设置最大速度，起跳速度，加速度和急停加速度时运动没有停止
9	缓冲区状态管理指令出错

位	定义
	(1) 在系统没有停止的情况下，调用了 RD_StrtMtn() (2) 在缓冲区已经打开或者系统正在运动 时，调用了 RD_StrtList() (3) 在运动没有停止的情况下，调用了 RD_RestoreMtn()
10	保留
11	系统运动的情况下，调用了 RD_ZeroPos()或者 RD_SetAtlPos()
12	保留
13	保留
14	保留
15	运动控制器指令缓冲区满，刚才被调用的缓冲区指令没有被压入缓冲区

第三章 初始化运动控制卡

RDS3040G-PCI 可以实现 4 轴的电机的连续轨迹的运动控制，为了保证系统正常的控制，必须根据实际的系统配置正确的初始化运动控制卡。

运动控制卡的初始化包括以下的几个方面：

- 电机轴参数初始化，包括限位开关、报警输入、原点开关、限位开关有效电平、捕获源选择，捕获源有效电平、编码器使能、编码器计数方向
- 伺服控制参数初始化，包括轴控制模式、脉冲输出方式、PID 参数、零飘电压补偿、输出电压饱和极限、积分误差饱和极限、跟随误差极限
- 坐标系参数初始化，起跳速度、最大速度

3.1 板卡初始化

3.1.1 初始化函数

表 3-1 通用运动控制器初始化函数

指令	说明
RD_Open	打开运动控制器，获得设备的访问权
RD_Reset	复位运动控制器，建立起和 DSP 计算机的通讯。外围复位端口的输出状态。
RD_SetLmt	设置限位开关输入有效或者无效 如果用户要禁止掉指定的限位开关输入，把其参数设置为 0 即可，设置后，无论实际的硬件限位开关是什么状态，都不影响系统的运动。
RD_SetAlarm	使能或者禁止报警信号输入
RD_SetLmtSns	设置限位触发输入信号的有效电平，可以设置高电平或者低电平。如果用户联结的硬件限位开关为常开开关，当有限位触发时，则输入电平从高变低，此时如果设置的限位触发信号有效电平为低电平，则板卡执行限位保护；如果此时设置的限位触发信号有效电平为高电平，则系统永远位于限位保护状态，直到限位开关变为闭合。
RD_SetEncoder	伺服闭环控制时，打开或者关闭编码器输入，控制器可以对编码器输入信号进行计数
RD_SetEncSns	伺服闭环控制时，编码器计数器的计数方向，该指令用于调整伺服闭环负反馈的极性。
RD_StepOutMode	开环控制模式下，输出脉冲信号的类型
RD_SetStrtVel	设置坐标系运动的起跳速度
RD_SetMaxVel	设置坐标系运动的最大速度

3.1.2 重点说明

3.1.2.1 打开通用运动控制器

必须首先调用指令 `RD_Open()` 打开 RDS3040G-PCI 通用运动控制器，获得设备的访问权。

在退出应用程序时应当调用指令 `RD_Close()` 关闭运动控制器，释放 PCI 通讯的句柄。

3.1.2.2 复位运动控制器

调用指令 `RD_Reset` 将 RDS3040G-PCI 运动控制器复位，建立起和 DSP 计算机的通讯，在调用 `RD_Open()` 打开控制器后，必须调用 `RD_Reset()` 复位运动控制器，保证正常的通讯。复位后各个输出端的输出电平如表 3-2 所示。

3.1.2.3 限位开关信号设置

- 使能和禁止限位开关信号输入

调用函数 `RD_SetLmt()` 可以使能或者禁止限位开关信号输入，如果禁止了限位开关信号输入，则板卡就不再检查外部限位开关的状态。

- 限位开关有效电平

调用函数 `RD_SetLmtSns()` 函数可以设置限位开关信号输入的有效电平。

表 3-2 各轴正限位开关和 `RD_SetLmtSns()` 的参数对应关系

状态位	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
限位 开关	轴 8	轴 7	轴 6	轴 5	轴 4	轴 3	轴 2	轴 1
	保留	保留	保留	保留	Lmt3+	Lmt2+	Lmt1+	Lmt0+

状态位	BIT15	BIT14	BIT13	BIT12	BIT11	BIT10	BIT9	BIT8
限位 开关	轴 8	轴 7	轴 6	轴 5	轴 4	轴 3	轴 2	轴 1
	保留	保留	保留	保留	Lmt3-	Lmt2-	Lmt1-	Lmt0-

运动控制卡为每个轴都提供了正负限位开关的接口，各个轴的限位开关应该和限位输入接口和外部电源地（OGND）连接。如果正确的设置了限位开关输入，则当某个轴运动触发了限位开关，则运动控制卡将自动停止整个系统的运动，以保证系统的安全。当相关轴离开限位开关后，轴状态字中的限位触发位仍然存在，必须调用 `RD_ClrSts()` 才可清除。

运动控制卡默认的限位开关为“常闭”开关，即默认情况下输入到限位开关接口的为“低”电平。当限位开关的输入电平转高时，限位触发！如果用户使用的是常开开关，则必须调用 `RD_SetLmtSns()` 函数，重新设置其触发电平。否则，默认时，各轴都处于限位触发状态！

3.1.2.4 编码器控制参数设置

每个轴都支持一路编码器信号的输入，编码器输入的信号差分的 A, B 相脉冲和 Z 脉冲。

运动控制卡在默认情况下是禁止外部的编码器输入的，所以，为了得到外部编码器的实际计数值，必须调用函数 `RD_SetEncoder()` 打开编码器输入，通过板卡函数 `RD_GetAtlPos()` 才能获得编码器的实际计数值。

有时编码器的计数方向和实际坐标系的方向相反，此时希望编码器能反向计数，那么就要调用函数 `RD_SetEncSns()` 函数来改变编码器的计数方向。运动控制卡有如下约定：

当编码器输入信号 A 相超前 B 相 90 度时，板卡内部计数为增计数，否则为反计数。所

以用户必须在系统初始化之前必须清除编码器输入和运动方向的对应关系，即当 X 往正向运动时，编码器反馈的计数值为递增计数，否则为递减计数。

注意：这些初始化的函数都是在系统没有上伺服之前以及运动完成的情况下才能调用！



注意

注意

在没有正常的初始化板卡之前，系统的输出状态可能为不定的状态，会导致出光，有可能对人身造成伤害，在和系统联机的情况下，一定要先给安装有板卡的工控机供电，保证板卡的正常初始化，再开启设备的电源。同样的，关闭系统时，一定要先关闭设备电源，在关闭工控机的电源。

第四章 通用运动控制卡状态检测

为了让用户可以方便的获取板卡内部的工作状态，板卡提供了多个状态获取函数。用户可以在任何时候通过调用这些接口函数获取板卡当前的状态。这些状态包括：

- 当前规划位置的 X, Y 坐标信息
- 当前的指令缓冲区的状态
- 当前指令的执行状态

4.1 函数列表

表 4-1 状态检测指令列表

指令函数	说明
RD_GetCrdSts	获取坐标系运动的状态
RD_GetCmdSts	获取指令错误状态
RD_GetPrfPnt	获取当前的规划位置
RD_GetListSpace	获取当前缓冲区的剩余空间的大小
RD_GetSts	获取指定轴的状态

4.2 函数说明

用户可以调用不同的函数获得必需的状态反馈信息。如坐标系运动状态，控制轴状态等。

4.2.1 坐标系运动状态寄存器

RDS3040G-PCI 通用运动控制器提供一个寄存器记录多轴运动控制的状态。控制轴运动状态标志位表示实际表示的是控制器内部运动输出状态，而不是控制轴的实际运动状态，用户调用指令 `RD_GetCrdSts()` 读取该寄存器，用户程序不能直接清除或设置该寄存器的状态。通用运动控制卡的运动状态寄存器定义如表 4-3 所示：

表 4-3 通用运动控制器状态寄存器

位	定义
0	坐标系运动标志位，0：坐标系运动正在进行，1：运动完成（默认）
1	缓冲区是否打开，0：打开； 1：关闭（默认）
2	轨迹运动预处理是否正常，0：正常（默认），1：预处理时间不够
3	保留
4	当前段运动状态，0：当前段正在运动； 1：当前段运动完成（默认）
5~6	保留
7	指令输入类型，0：缓冲区指令输入； 1：立即指令输入（默认）

位	定义
8	保留
9	坐标系运动中，轴运动是否出现异常，如相关轴的限位触发，伺服报警等，0：相关轴正常（默认）；1：相关轴异常
10	脉冲输出是否异常，0：正常（默认）；1：异常
11~12	保留
13	缓冲区是否空，0：不空（默认）；1：空
14	缓冲区是否满，0：不满（默认）；1：满
15	保留

说明：**BIT0：**

坐标系运动标志位，表示当前的坐标系运动是否完成。该位表示用户所有的加工信息已经执行完毕，坐标系运动完成且处于静止状态，用户已经关闭了控制器的缓冲区。满足这些条件才表示整个坐标系的运动已经完成。

BIT1：表示处于缓冲区运动状态还是立即运动状态。1表示是缓冲区运动模式，0表示立即运动模式

BIT2：运动轨迹预处理是否正常。

BIT4：指示当前段的执行状态，是已经运动完成还是正在运动。

BIT7：当前指令时缓冲区指令还是立即指令

4.2.2 轴状态寄存器

系统给每个轴都提供了一个轴状态寄存器供用户调用察看。

表 4-4 轴状态寄存器

位	定义
0	当前段运动状态，0：正在运动；1：运动完成（默认）
1	伺服报警标志，0：无报警（默认）；1：有报警
2	系统暂停标志位，0：正常运行（默认）；1：暂停
3	HOME+INDEX 捕获标志，0：没有捕获（默认）；1：捕获发生
4	运动错误标志，0：无错误（默认）；1：错误
5	正限位标志，0：没有触发（默认）；1：触发 <i>注：已经考虑了限位输入极性的因素</i>
6	负限位标志，0：没有触发（默认）；1：触发 <i>注：已经考虑了限位输入极性的因素</i>
7	轴参数设置错误，0：正确（默认）；1：错诉
8	闭环控制标志位，0：开环（默认）；1：闭环
9	电机伺服使能标志，0：伺服禁止（默认）；1：伺服使能
10	轴运动状态标志位，0：轴正在运动；1：轴运动完成

位	定义
11	限位开关使能标志位, 0: 表示无效; 1: 表示有效(默认)
12~13	轴号 BIT13 BIT12 0 0 轴 1 0 1 轴 2 1 0 轴 3 1 1 轴 4
14	HOME 捕获使能, 0: 禁止 (默认); 1: 使能
15	INDEX 捕获使能, 0: 禁止 (默认); 1: 使能

第五章 轴参数控制

轴作为坐标系运动中的基本单元，必须合理正确的设置其参数。才能保证坐标系运动的正常。

常见的轴参数包括：

- 1 限位开关参数设置
- 2 编码器参数设置
- 3 报警输入参数设置
- 4 脉冲输出模式设置
- 5 轴控制模式设置
- 6 轴使能控制
- 7 轴状态清除

5.1 限位开关参数设置

限位开关设置的函数如表 5-1 所示。

表 5-1 限位信号输入参数设置

指令函数	说明																				
RD_SetLmt(unsigned short axis,unsigned short lmt);																					
参数	Axis: 轴号，从 0~3，分别对应轴 1~轴 3																				
说明	使能或禁止指定轴的限位信号输入 如果设置为禁止，则无论外部限位输入是什么状态，都不影响系统的运动。																				
RD_SetLmtSns(unsigned short sns);																					
参数	Sns: 位和轴号的对应关系如下 <table style="width: 100%; border-collapse: collapse;"> <tr> <td style="text-align: center;">BIT0</td> <td style="text-align: center;">BIT1</td> <td style="text-align: center;">BIT2</td> <td style="text-align: center;">BIT3</td> </tr> <tr> <td style="text-align: center;">LMT0+</td> <td style="text-align: center;">LMT1+</td> <td style="text-align: center;">LMT2+</td> <td style="text-align: center;">LMT3+</td> </tr> <tr> <td colspan="4" style="text-align: center;">-----</td> </tr> <tr> <td style="text-align: center;">BIT8</td> <td style="text-align: center;">BIT9</td> <td style="text-align: center;">BIT10</td> <td style="text-align: center;">BIT11</td> </tr> <tr> <td style="text-align: center;">LMT0-</td> <td style="text-align: center;">LMT1-</td> <td style="text-align: center;">LMT2-</td> <td style="text-align: center;">LMT3-</td> </tr> </table>	BIT0	BIT1	BIT2	BIT3	LMT0+	LMT1+	LMT2+	LMT3+	-----				BIT8	BIT9	BIT10	BIT11	LMT0-	LMT1-	LMT2-	LMT3-
BIT0	BIT1	BIT2	BIT3																		
LMT0+	LMT1+	LMT2+	LMT3+																		

BIT8	BIT9	BIT10	BIT11																		
LMT0-	LMT1-	LMT2-	LMT3-																		
说明	设置相关轴的限位开关的输入有效电平 如果指定位为 1，表示该限位输入信号是高电平有效，即当外部的限位开关输入由低电平变为高电平时，限位触发，板卡自动停止运动！																				

5.2 编码器参数设置

编码器参数设置包括使能和禁止编码器输入以及编码器计数方向控制。这些参数必须在系统没有运动或者轴没有使能。

表 5-2 编码器参数设置

指令函数		说明
RD_SetEncoder(unsigned short axis,unsigned short enc);		
参数	Axis: 轴号, 从 0~3, 分别对应轴 1~轴 3	
说明	使能或禁止指定轴的编码器信号输入 如果设置为禁止, 则通过函数 RD_GetAtIPos()获取的计数值为内部脉冲计数值, 如果设置为使能, 则该函数返回的计数值为实际编码器的计数值	
RD_SetEncSns(unsigned short sns);		
参数	Sns: 位和编码器的轴号的对应关系如下 BIT0 BIT1 BIT2 BIT3 ENC0 ENC1 ENC2 ENC3	
说明	设置相关轴的编码器计数方向 如果指定位为 1, 则编码器正向计数, 否则为反向计数	

5.3 报警输入设置

报警信号输入用于作为伺服驱动器或者其他保护信号输入时, 紧急制动系统运动。用户可以使能或者禁止报警信号输入, 如表 5-3 所示。

表 5-3 报警信号输入设置

指令函数		说明
RD_RD_SetAlarm (unsigned short axis,unsigned short alm);		
参数	Axis: 轴号, 从 0~3, 分别对应轴 1~轴 3	
说明	使能或禁止指定轴的报警信号输入 如果设置为禁止, 则外部有报警信号输入时, 系统将忽略该信号。 如果设置为使能, 则 外部有报警信号输入时, 系统将紧急停止。	

5.4 脉冲输出参数设置

当设置轴的控制方式为脉冲控制方式时, 可以调用函数 RD_SetStepOutMode () 设置脉冲输出的方式。如表 5-4 所示。

表 5-4 脉冲输方式选择

指令函数		说明
RD_SetStepOutMode (unsigned short axis,unsigned short mode);		
参数	Axis: 轴号, 从 0~3, 分别对应轴 1~轴 4	
说明	在脉冲输出模式下, 设置轴的输出是 PULSE+DIR 或 +/-PULSE 模式。 0: PULSE+DIR	

指令函数	说明
1: +/-PULSE	

5.5 轴控制模式设置

对于一工作轴，可以工作在步进模式，即开环控制模式，也可以工作在伺服模式，即闭环控制模式。系统默认工作在开环模式。

如果用户连接的伺服系统，且要工作在闭环控制模式，则需调用函数 RD_SetCtrlMode()来设置轴的工作模式。如表 5-5 所示。

表 5-5 轴控制模式选择

指令函数	说明
RD_SetCtrlMode (unsigned short axis,unsigned short mode);	
参数	Axis: 轴号, 从 0~3, 分别对应轴 1~轴 4 Mode : 0: 开环模式 1: 闭环模式
说明	设置为闭环工作模式, 此时控制器输出模拟电压速度指令, 实现对伺服系统的闭环控制。

一旦设置了轴控制模式为闭环控制模式, 必须正确的设置伺服控制参数。伺服控制参数如表 5-6 所示。

表 5-6 伺服参数设置

指令函数	说明
RD_SetServoFilterPara (unsigned short axis, TPID *pPID);	
参数	Axis: 轴号, 从 0~3, 分别对应轴 1~轴 4 Para : 伺服闭环控制滤波器参数 typedef struct { double Kp; double Ki; double Kd; double Kvff; double Kaff; }TPID;
说明	闭环反馈控制的滤波器参数

调用函数 RD_SetServoFilterPara () 设置滤波器参数后, 必须调用 RD_Updt()同步函数更新数据, 这些设置的参数才生效。

为了保证闭环反馈控制能够安全可靠的工作, 还必须设置好系统的极限参数。如 表 5-8 所示。

表 5-8 轴极限参数设置

指令函数	说明
RD_SetAxisSysPara(unsigned short axis, TSysPara *pSys);	
参数	Axis: 轴号, 从 0~3, 分别对应轴 1~轴 4 Para : 伺服闭环控制滤波器参数 <pre>typedef struct { short mtrBias; short mtrLmt; short mtrIlmt; short posError; }TSysPara;</pre> mtrBias: 零飘电压补偿 mtrLmt: 输出电压饱和极限 mtrIlmt: 误差积分饱和极限 posError: 跟随误差极限
说明	闭环反馈控制的极限参数, 用于系统控制的保护。

当轴工作在闭环模式时, 在运动控制器的速度指令输出为 0 时, 理论上伺服系统电机应该处于静止状态, 但是实际上电机会转动, 造成电机转动的原因就是零飘电压引起的。如果不消除零飘电压, 将会影响系统控制的精度, 造成控制性能的下降。

有两种方式可以消除零飘电压, 一种就是调用函数 RD_SetAxisSysPara () 设置反零飘电压的值。一旦设置了该值, 板卡输出一个模拟电压的偏移值, 以抵消伺服驱动器零飘电压的影响。用户设置时, 需要逐步的测试, 直到伺服电机静止为止。另外一个方法就是, 伺服控制滤波器除了比例系数外, 还必须增加积分系数。这样才能保证运动精度

5.6 轴控制指令

轴控制指令包括轴的使能控制和轴的状态清除。轴的运动必须在轴使能的情况下才有效。

表 5-7 轴控制

指令函数	说明
RD_AxisOn (unsigned short axis);	
参数	Axis: 轴号, 从 0~3, 分别对应轴 1~轴 4
说明	使能指定轴的控制, 必须使能指定轴, 才能实现对各个轴的运动控制 如果有报警或者限位触发, 将不能使能轴。
RD_ClrSts(unsigned short axis)	
参数	Axis: 轴号, 从 0~3, 分别对应轴 1~轴 4
说明	清除指定轴的状态。 轴的报警输入状态或者限位触发状态, 一旦触发, 就一直有效, 板卡本身不能自动清除, 在触发源消除后, 用户必须 调用该函数清除状态。

第六章 坐标系参数设置

启动一个坐标系的运动，在正确的设置了轴参数后，必须合理的设置坐标系参数。才能保证一个正确的坐标系运动。

坐标系参数包括如下的参数：

- (1) RD_SetMapCnt()
- (2) RD_SetStrtVel()
- (3) RD_SetOverride()
- (4) RD_SetMaxVel()

6.1 坐标系定义

- 机床坐标系：

机床坐标系是以机床原点位置为坐标原点的坐标系，它是在系统执行回零操作后，接触到原点开关后确立的坐标系。

- 加工坐标系

加工坐标系是指当前工件的参考的相对的坐标系。这个坐标系相对于机床坐标系是有一个偏移量。

在系统默认情况下，机床坐标系和加工坐标系是重合的。调用 RD_SetMapCnt()可以设置这两个坐标系的偏移量。

表 6-1 坐标系偏移

指令函数		说明
RD_SetMapCnt(unsigned short axis, long cnt);		
参数	Axis:轴号 Cnt: 坐标系偏移量	
说明	设置指定控制轴坐标偏移量	

6.1 设置坐标系参数

坐标系参数设置函数如下表：

表 6-2 坐标系参数

指令函数		说明
RD_SetStrtVel(double strtvel);		
参数	strtvel: 起跳频率	
说明	设置坐标系运动的起跳频率	
RD_SetOverride(double override);		
参数	Override: 速度倍率	
说明	设置坐标系运动的速度倍率	
RD_SetMaxVel(double maxvel);		

指令函数		说明
参数	Maxvel: 最大速度	
说明	设置坐标系运动的最大速度	

6.1.1 RD_SetStrtVel

该指令设置轨迹运动的起跳速度，单位为 **pulse/ms**。一般在对运动控制器进行参数初始化时调用该指令，该指令调用后立即生效。在运动过程中可以改变启动速度，但会影响运动控制器的速度规划，**建议用户只在初始化时调用该指令**。

对于步进电机，存在一个起跳频率，步进电机可以直接以起跳频率为启动速度（不必是零速）开始加速到进给速度，可减少加速时间，提高加工效率，对于伺服电机，起跳速度可以设置为 0。

6.1.2 RD_SetMaxVel

该指令设置轨迹运动的最大速度，单位为 **pulse/ms**。一般在对运动控制器进行参数初始化时调用该指令，该指令在调用后立即生效。在运动过程中可以改变最大速度，但会影响运动控制器的速度规划，**建议用户只在初始化时调用该指令**。

根据步进电机的矩频特性，运行频率高于某个频率时，输出力矩下降非常快，可能出现丢步或电机堵转的情况。建议用户在初始化时根据机床负载和电机工作技术参数情况调用指令 RD_SetMaxVel 设置最大速度。运动控制器的默认脉冲输出最大频率为：1MHZ。

6.1.3 RD_SetStpAcc

该指令设置轨迹运动的急停加速度，单位为 **pulse/ms²**。在运动过程中，系统出现限位触发、伺服报警等异常情况时，RDS3040G-PCI 将按照该加速度实现减速停止，防止从高速立即停止下来造成对电机和系统的损害。一般在对运动控制器进行参数初始化时调用该指令。

用户在设置急停加速度时应注意使急停加速度大于进给加速度，并保证当系统运行在最高速度时急停，不会对系统和电机产生过度损伤。

6.1.4 RD_Override

用户可调用 **RD_Override** 指令修改当前的速度倍率，达到在线修改当前轨迹运动的进给速度的目的，同时也改变了以后所有轨迹运动的进给速度。一旦成功修改速度倍率，该倍率值将一直有效，直到再次调用 RD_Override 指令修改速度倍率。

运动控制器默认的速度倍率均为 100%。

一般来说速度倍率的可调整范围在 0%~200%之间，如果用户调整速度倍率使进给速度大于系统初始化设置的最大速度，运动控制器取最大速度为进给速度。

6.3 例程

例程 6-1 设置轨迹运动的进给速度、加速度

该程序示例一个轨迹运动的进给速度、加速度的设置以及单位说明：

丝杆导程 5mm，电机每转脉冲数为 10000pulse（4 倍频后）。

进给速度（WorkVel）为 3m/min，进给加速度（WorkAcc）为 0.9m/s²，启动速度（StartVel）为 300pulse/s，最大速度（MaxVel）为 15m/min，急停加速度（StopAcc）为 3m/s²。

$$WorkVel = \frac{3000mm}{60 * 10^3 ms} \times \frac{10000 pulse}{5mm} = 100 pulse / ms$$

$$WorkAcc = \frac{900mm}{(10^3 ms)^2} \times \frac{10000 pulse}{5mm} = 1.8 pulse / ms^2$$

$$StartVel = \frac{300 pulse}{10^3 ms} = 0.3 pulse / ms$$

$$MaxVel = \frac{15000mm}{60 * 10^3 ms} \times \frac{10000 pulse}{5mm} = 500 pulse / ms$$

$$StopAcc = \frac{3000mm}{(10^3 ms)^2} \times \frac{10000 pulse}{5mm} = 6 pulse / ms^2$$

第七章 运动模式

RDS3040G-PCI 的运动模式可以分为坐标系插补模式。坐标系插补模式基于正交的笛卡尔坐标系，最多完成 4 轴的连续轨迹直线插补和 3 轴的空间圆弧曲线插补。

7.1 连续轨迹运动

RDS3040G-PCI 可以实现 4 轴联动，提供了直线插补和笛卡尔坐标系里各个平面的圆弧插补功能，螺旋线插补以及样条插补功能。结合速度预测功能，可以实现多段连续轨迹的光顺的插补运动。

7.2 立即模式下的直线插补

立即运动指令本身的参数包含了要运动的目标位置、速度和加速度，不需要另外设置速度和加速度。一旦调用了该指令，只要系统状态正常，则会启动一段立即运动，立即指令下的直线插补指令运动时，速度从 0 开始加速，然后降到 0。

7.2.1 指令列表

表 7-1 轨迹运动的轨迹描述指令列表

指令函数		说明
RD_MoveToXYZ(long x,long y,long z,double vel,double acc);		
参数	X,Y,Z 为终点坐标 Vel: 合成速度 Acc: 合成加速度	
说明	启动一个空间直线的移动	
RD_MoveToXY(long x,long y,double vel,double acc);		
参数	X,Y: 终点坐标 Vel: 合成速度 Acc: 合成加速度	
说明	启动一个 XY 平面的直线移动	
RD_MoveToXYZA(long x,long y,long z,long a,double vel,double acc);		
参数	X,Y,Z,A 为终点坐标 Vel: 合成速度 Acc: 合成加速度	
说明	启动 4 轴直线的移动	

7.2.2 重点说明

RDS3040G-PCI 运动控制器提供的轨迹描述指令都是在正交坐标系中描述的。轨迹描述指令的长度单位为 pulse，要求用户在系统机械设计方面确保各控制轴的脉冲当量一致，以保证轨迹描述的正确和速度的一致性。

7.3 停止坐标系运动

7.3.1 指令列表

表 7-2 停止控制轴运动指令列表

指令函数	说明
RD_StpMtn(void);	
参数	
说明	平滑停止坐标系运动
RD_EStpMtn(void);	
参数	
说明	紧急停止坐标系运动

7.3.2 重点说明

如果需要在运动过程当中停止坐标系运动，可以调用指令 RD_StpMtn 或 RD_EStpMtn。

RD_StpMtn 是立即指令，能够平滑停止坐标系运动。调用指令 RD_StpMtn 时，系统根据用户所设定的工作加速度平滑减速到 0。当控制轴平滑停止运动完成之后(必须调用函数 RD_GetCrdSts() 函数判断运动是否完成)，用户才能重新设置坐标系参数继续运动。指令 RD_StpMtn 的效果如下所示：

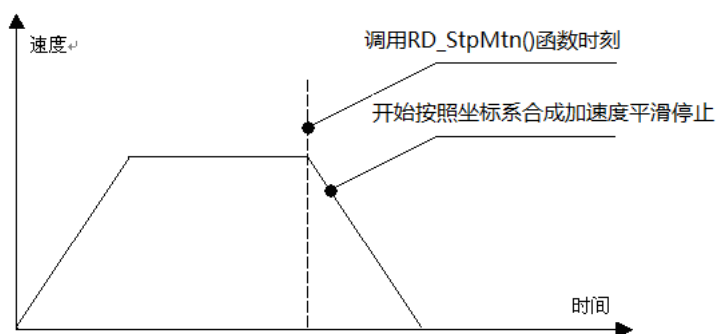


图 7-1 坐标系运动平滑停止

RD_EStpMtn 是立即指令，能够紧急停止控制轴的运动。调用 RD_EStpMtn 指令时，系统将按照函数 Rd_SetStpAcc () 设定的急停加速度立即减速到 0。

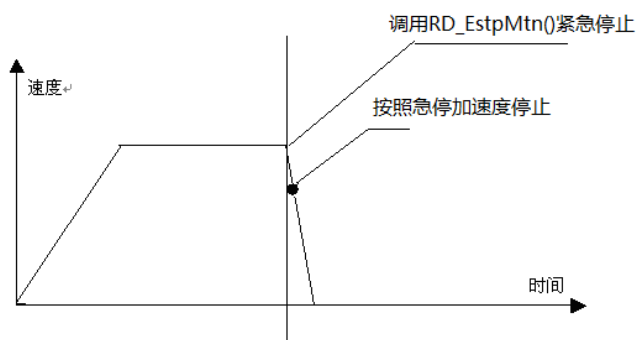


图 7-2 紧急停止过程

7.4 连续轨迹插补

实现多段连续轨迹运动，必须启动缓冲区指令输入和缓冲区指令执行方式。在该方式下，系统将指令一系列的轨迹段的平滑的连续轨迹运动。

为了实现多段连续轨迹运动，运动控制器提供一个大小为 8000 条指令的**缓冲区**。用户可先将部分轨迹描述或参数指令存放在该缓冲区中（以缓冲区满为限），然后发出执行指令。在运动控制器执行缓冲区内轨迹描述指令的同时，用户可继续向这个缓冲区内下载轨迹描述或参数指令。

运动控制器通过对缓冲区内多段轨迹描述指令的速度预处理，能够获得良好的速度规划特性。

下面分两部分讲述利用缓冲区实现连续轨迹运动的方法：

- 将轨迹描述指令和参数指令输入缓冲区；
- 执行缓冲区连续轨迹运动；
- 循环压指令到缓冲区；

7.4.1 将轨迹指令和参数指令输入缓冲区

7.4.1.1 缓冲区控制指令列表

表 7-3 缓冲区输入管理指令列表

指令函数	说明
RD_StrtList(void);	
参数	
说明	打开运动控制卡缓冲区，允许缓冲区指令输入
RD_EndList(void);	
参数	

指令函数		说明
说明	关闭运动控制卡缓冲区，禁止缓冲区指令输入	
RD_AddList(void);		
参数		
说明	暂停后，再启动时，重新打开缓冲区	

7.4.1.2 缓冲区指令列表

表 7-4 可输入缓冲区的指令列表

指令	说明
RD_SetSynVel	设置进给速度
RD_LnXYG0	两维直线插补（空走）
Rd_LnXYZG0	三维直线插补（空走）
RD_LnXYZAG0	四轴直线插补（空走）
RD_LnXY	两维直线插补
RD_LnXYZ	三维直线插补
RD_LnXYZA	四轴直线插补
RD_ArcXY	XY 平面圆弧插补（以圆心位置和角度为输入参数）
RD_ArcXYP	XY 平面圆弧插补（以终点位置和半径为输入参数）
RD_ArcYZ	YZ 平面圆弧插补（以圆心位置和角度为输入参数）
RD_ArcYZP	YZ 平面圆弧插补（以终点位置和半径为输入参数）
RD_ArcZX	ZX 平面圆弧插补（以圆心位置和角度为输入参数）
RD_ArcZXP	ZX 平面圆弧插补（以终点位置和半径为输入参数）
RD_SetDccVel	设置轨迹段的终点速度
Rd_BufIO	在缓冲区里开关输出点,对整个 IO 进行更新
Rd_BufIOBit	在缓冲区里开关输出点,对某一个输出点进行更新
RD_BufDA	在缓冲区运动中,对一路 DA 进行输出
RD_Delay	在缓冲区运动中执行延时,包括 IO 间和运动间的延时(单位是 ms)

7.4.1.3 重点说明

为正确描述圆弧插补运动，将圆弧插补指令描述的运动轨迹放在正交坐标系平面内，圆弧插补的旋转正方向按照右手螺旋定则定义为：从坐标平面的“上方”（即垂直于坐标平面的第三个轴的正方向）看，逆时针方向为正（图 11-3）。可以这样简单记忆：将右手拇指前伸，其余四指握拳，拇指指向第三个轴的正方向，其余四指的方向即为旋转的正方向。

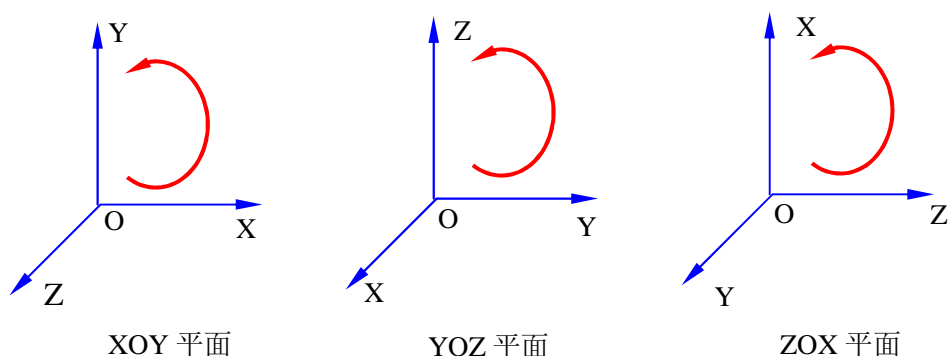


图 7-3 圆弧插补正方向

(1) 打开并清空缓冲区

RD_StrtList: 打开并清空缓冲区。

启动缓冲区连续轨迹运动之前，必须先调用指令 `RD_StrtList` 进入缓冲区指令输入状态。该指令在缓冲区关闭且无缓冲区连续轨迹运动时有效。

运动控制器规定，打开并清空缓冲区 (`RD_StrtList`) 后，当用户调用指令 `RD_StrtMtn` 启动缓冲区连续轨迹运动后，运动控制器将从当前位置按照直线插补的方式运动到缓冲区的第一段终点，紧接着缓冲区的下面段运动。

(2) 关闭缓冲区

RD_EndList: 关闭缓冲区。该指令在缓冲区处于打开状态时有效。在连续轨迹的缓冲区运动中，如果缓冲区指令全部执行完毕后，用户没有调用该函数关闭缓冲区，运动完成标志位将不会置起，只有关闭了缓冲区，调用函数 `RD_GetCrdSts()` 获取的运动完成标志位才能置起来。

(3) 连续轨迹运动指令存入缓冲区的注意事项

用户能够在调用指令 `RD_StrtList` 之后的任意时刻，调用指令 `RD_EndList` 结束缓冲区指令输入状态；

控制器对缓冲区中轨迹运动指令的处理机制

用户可以不断向缓冲区中添加轨迹描述或参数指令，直到缓冲区满。

缓冲区满时，运动控制器拒绝接收用户输入的轨迹描述指令和参数指令，并返回缓冲区满的信息。这时运动控制器不再接受新的轨迹指令，用户需要检查每个指令的返回值，如果是缓冲区满造成得到返回值为非 0，则在启动运动后，需要反复的压该指令，直到返回值为 0，否则该轨迹指令会被运动控制器忽略。

启动缓冲区连续轨迹运动后，随着轨迹描述指令的执行完成，缓冲区会有新的空间，用户可以继续发送轨迹描述或参数指令。返回值不对时，该指令发送不成功，保持加工文件的连续性，此段必须重发到成功为止。详细的指令压入流程如下：

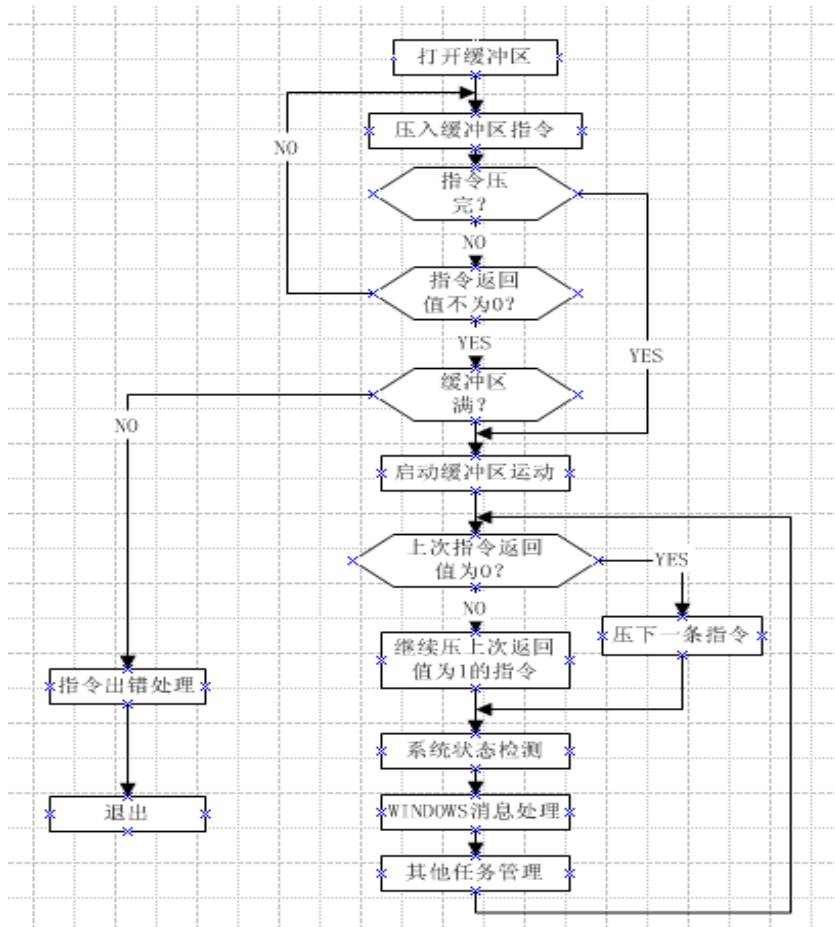


图 7-4 缓冲区指令管理流程

7.4.2 启动、停止缓冲区连续轨迹运动

7.4.2.1 指令列表

表 7-5 启动、停止指令列表

指令	说明
RD_StrtMtn	启动缓冲区连续轨迹运动
RD_StpMtn	平滑停止轨迹运动
RD_EstpMtn	紧急停止轨迹运动
RD_RestoreMtn	暂停后恢复缓冲区连续轨迹运动

7.4.2.2 重点说明

(1) 启动缓冲区连续轨迹运动

启动缓冲区连续轨迹运动前，用户应确保运动控制器的缓冲区内有轨迹描述指令。

建议用户在压满缓冲区后，再启动缓冲区的运动，此后一边压指令一边运动。

用户调用指令 **RD_StrtMtn** 成功后，缓冲区中的指令被顺序执行。

启动缓冲区连续轨迹运动后，运动控制器将用户传送的轨迹描述或参数指令放入缓冲区的同时，执行缓冲区连续轨迹运动，直到用户调用指令 `RD_EndList` 关闭缓冲区指令输入方式。

(2) 中断缓冲区中指令的执行

调用指令 `RD_StpMtn()` 后，运动控制器按照设定的加速度平滑停止轨迹运动；调用指令 `RD_EStpMtn()` 后，运动控制器按照急停加速度停止轨迹运动。

调用指令 `RD_StpMtn()` (`RD_EStpMtn()`) 成功后，缓冲区被立即关闭，相当于调用了指令 `RD_EndList`。当缓冲区连续轨迹运动规划停止后，运动控制器记录当前位置信息和当前段段号（又称断点信息），以确保恢复缓冲区轨迹运动时，能够准确地回到断点位置继续缓冲区连续轨迹运动。

当轨迹运动停止且缓冲区没有被再次打开，用户可以调用立即运动轨迹描述指令，运动控制器立即执行。

`RD_StpMtn()` 和 `RD_EStpMtn()` 指令也可中断正在进行的立即轨迹运动 (`RD_MoveToXY()`)，但不保留立即轨迹运动的断点信息，即立即轨迹运动被中断后无法恢复，必须重新调用轨迹描述指令启动立即轨迹运动。

(3) 恢复缓冲区中指令的执行

当调用 `RD_StpMtn()` (`RD_EStpMtn()`) 使缓冲区连续轨迹运动停止，且已经运动完成（调用 `RD_GetCrdSts` 获取的状态信息中的 `BIT0` 为 1），用户可以调用立即指令做轴的运动到任何位置。如果用户还要继续缓冲区内的运动，则**必须**调用立即指令运动到上次缓冲区运动的断点处，断点信息可以通过相关的函数获取。然后调用 `RD_RestoreMtn()` 指令，继续缓冲区连续轨迹运动。

恢复缓冲区的运动，不需要再调用 `RD_StrtList()` 重新打开缓冲区。一旦调用了函数 `RD_RestoreMtn()`，缓冲区会自动打开，用户调用的缓冲区指令会被自动的压入缓冲区。

调用 `RD_RestoreMtn()` 指令前，用户必须确保运动到了上次缓冲区的断点处，并确定曾经调用 `RD_StrtMtn()` 启动了缓冲区的运行，否则可能引起错误的运动。

暂停后恢复缓冲区连续轨迹运动**必须**调用 `RD_RestoreMtn()` 指令，若调用 `RD_StrtMtn()` 来恢复运动，会引起错误的运动。

7.4.3 速度预处理

为解决速度和精度的矛盾，RDS3040G-PCI 运动控制器提供基于前瞻预处理的速度规划策略。用户通过设定本身机床的工艺特征参数（脉冲当量、进给速度、最大加速度、允许拐弯时间等），结合 RDS3040G-PCI 运动控制器的前瞻预处理功能模块，可实现小线段连续轨迹加工。

指令 `RD_SetSynVel` 可改变进给速度。如果设定的进给速度超过了用户设定的最大速度，取最大速度为进给速度。

7.4.3.1 速度预处理指令列表

表 7-6 前瞻预处理指令列表

指令	说明
RD_InitLookAhead	初始化前瞻预处理参数
RD_AddLookData	将轨迹段信息添加到预处理缓冲区
RD_CalVel	计算轨迹段终点速度
RD_SetDccVel	设置轨迹段终点速度

7.4.3.2 重点说明

前瞻预处理模块的任务就是根据曲线的特征以及机械的特征参数、运动参数，通过一定的速度规划策略，规划出每一个轨迹段的合理的终点速度。

7.4.3.2.1 初始化前瞻预处理参数

RD_InitLookAhead: 设置前瞻预处理功能模块的初始化参数

表 7-7 前瞻预处理初始化参数列表

参数名称	参数意义	说明
T	拐弯时间（单位：s）	T 越大，计算出来的终点速度越大，但会降低加工精度；反之，提高了加工的精度，但计算出的终点速度偏低，一般取 0.003~0.006
acc_max	系统能承受的最大加速度（单位： mm/s^2 ）	根据不同的机械系统和电机驱动器取值不同，经验范围：100~10000 mm/s^2
acc_run	系统加速度（单位： mm/s^2 ）	设定加速度一定小于最大加速度
vel_run	系统进给速度（单位： mm/s ）	系统加工速度

参数 **T** 和 **acc_max** 会影响拐点处的位置误差：

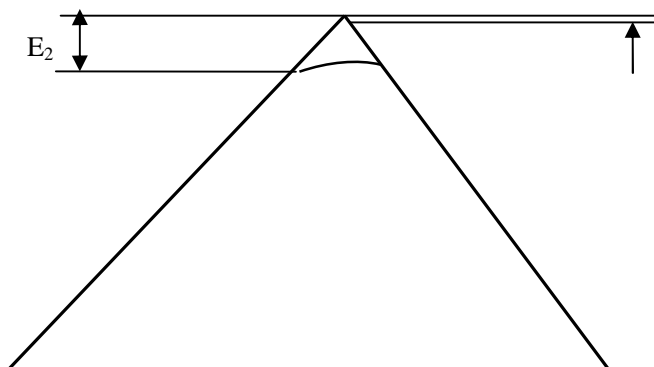


图 7-5 不同 T 和 Acc_max 参数的位置误差比较

观察图 7-5，当 T 或 acc_max 参数设置较大时，降速较少，但位置误差 E_2 较大；若 T 和 acc_max 参数设置合理，可以保证位置误差 E 进入系统能接受的精度范围。

在运动过程中，由于 X、Y 轴不同的惯量，实际应用中选用电机驱动时，一般会选择不功率的电机，如下图，X 轴的惯量要明显小于 Y 轴，因此，从加速度上分析，X 方向由于惯量比较小，可以有比较高的加速度，而 Y 轴则不能有比较高的加速度和速度，否则会造成系统的抖动。

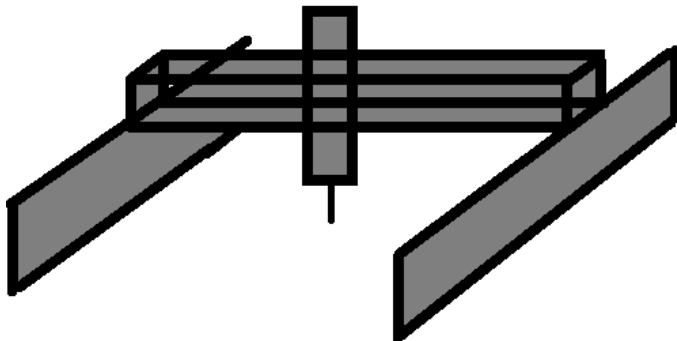


图 7-6 XY 轴不同惯量示意图

基于此，在我们的速度前瞻模块中，为了实现高速高精度的控制，还考虑了 XY 不同惯量的因素。用户必须设置 XY 的转动惯量比，同时 XY 轴的加速度分开设置。系统自动的根据实际的位置坐标修改系统运行的加速度和速度，达到全幅面范围内的最优的运动控制。

7.4.3.2.2 传送轨迹特征数据进入预处理缓冲区

RD_AddLookData: 将轨迹特征数据加入预处理缓冲区

表 7-8 传递轨迹特征数据列表

参数名称	参数意义	说明	
Code	曲线类型	0	G00 直线
		1	G01 直线
		2	G02 顺圆弧
		3	G03 逆圆弧
PlaneGroup	平面代码	17	XY 平面
		18	ZX 平面
		19	YZ 平面
R	圆弧半径	若当前段不是圆弧，设为 0	
x, y, z	坐标终点位置	单位：mm	
F	当前段进给速度	允许每段有不同的进给速度（单位：mm/s）	
cx, cy	圆心坐标	若当前段不是圆弧，设为 0	
i	压入到预处理缓冲区的位置		
n	当前轨迹段的序号		
flag	降速标志	表示当前段的终点速度是否减速到 0	

7.4.3.2.3 计算终点速度

RD_Calvel: 计算预处理当前轨迹段的终点速度；调用该指令的返回值若为 0，表示当前轨迹段的终点速度计算完毕，正确返回，若为 1 则表示预处理缓冲区里的数据已经处理完，所有的轨迹段的终点速度已经计算完毕。

7.4.4 坐标系信息反馈

7.4.4.1 指令列表

表 7-9 连续轨迹运动信息反馈指令列表

指令	说明
RD_GetAtlPos()	读取指定控制轴的实际位置
RD_GetPrfPos()	读取指定控制轴的规划位置
RD_GetPrfPnt()	获得全部四个轴的规划位置
RD_GetPrfVel()	读取规划速度
RD_GetCrdSts()	读取连续轨迹运动状态
RD_GetSts()	读取指定控制轴状态
RD_GetCurSegNm()	读取当前轨迹段编号
Rd_GetBrkPnt()	读取所有控制轴的断点位置

7.4.4.2 重点说明

7.4.4.2.1 读取规划位置 and 实际位置

RD_GetPrfPos(): 调用该指令读取连续轨迹运动中指定控制轴的规划位置。

RD_GetAtlPos(): 调用该指令读取指定控制轴的实际位置。该位置是从电机编码器反馈回来的值。如果电机在定位时有抖动，则读回来的值和规划位置也有些不同。

RDS3040G-PCI 运动控制器处于脉冲输出方式时，系统处于开环控制，默认关闭编码器。

用户可以调用 **RD_SetEncoder()** 指令打开指定控制轴的编码器，调用 **RD_GetAtlPos()** 读取编码器的实际位置值。

若关闭编码器，则无法读取编码器的实际位置，调用 **Rd_GetAtlPos** 读取的位置信息实际为规划位置值。

7.4.4.2.2 轨迹段号

RD_GetCurSegNm(): 该指令返回当前轨迹段编号。

轨迹段编号规则

1. 启动缓冲区中指令执行后，轨迹段编号随输入缓冲区段数递增，段号从 1 开始；
2. 段号由 1 开始累加，32 位表示范围，溢出后从 0 开始计数；
3. 当缓冲区内的轨迹段已经执行完毕，但用户又没有执行 **RD_EndList()** 指令关闭缓冲区，连

续轨迹运动状态字的 bit0 和 bit1 为 0，轨迹段号保持在当前段；

4. 用户若调用 RD_StpMtn()或 RD_EStpMtn()指令中断缓冲区连续轨迹运动，连续轨迹运动状态字的 Bit1 和 Bit0 为 1，段号保持当前值，直到恢复缓冲区连续轨迹运动；
5. 当前段运动结束时，若没有连续轨迹运动段，段号保持当前值；
6. 调用 RD_StrtList()指令，段号清零。

7.4.4.2.3 连续轨迹运动状态和其它状态

RD_GetCrdSts(): 调用该指令读取连续轨迹运动状态，其标志位定义见表 7-10。

表 7-10 连续轨迹运动状态定义

位	定义
0	坐标系运动标志位，0：坐标系运动正在进行，1：运动完成（默认）
1	缓冲区是否打开，0：打开； 1：关闭（默认）
2	轨迹运动预处理是否正常，0：正常（默认），1：预处理时间不够
3	保留
4	当前段运动状态，0：当前段正在运动； 1：当前段运动完成（默认）
5~6	保留
7	指令输入类型，0：缓冲区指令输入； 1：立即指令输入（默认）
8	保留
9	坐标系运动中，轴运动是否出现异常，如相关轴的限位触发，伺服报警等，0：相关轴正常（默认）； 1：相关轴异常
10	脉冲输出是否异常，0：正常（默认）； 1：异常
11~12	保留
13	缓冲区是否空，0：不空（默认）； 1：空
14	缓冲区是否满，0：不满（默认）； 1：满
15	保留

7.4.4.2.4 轴状态寄存器状态位定义

RD_GetSts(): 调用该指令获取指定控制轴的外部信号状态。正在进行连续轨迹运动的控制轴出现限位、伺服报警等异常状态时，RDS3040G-PCI 运动控制器自动停止轨迹运动，并将控制轴状态相应标志位置位，便于用户检测后进行相应处理。

表 7-11 轴状态寄存器定义

位	定义
0	当前段运动状态，0：正在运动； 1：运动完成（默认）

位	定义															
1	伺服报警标志, 0: 无报警 (默认); 1: 有报警															
2	系统暂停标志位, 0: 正常运行 (默认); 1: 暂停															
3	HOME+INDEX 捕获标志, 0: 捕获禁止 (默认); 1: 捕获使能															
4	运动错误标志, 0: 无错误 (默认); 1: 错误															
5	正限位标志, 0: 没有触发 (默认); 1: 触发 <i>注: 已经考虑了限位输入极性的因素</i>															
6	负限位标志, 0: 没有触发 (默认); 1: 触发 <i>注: 已经考虑了限位输入极性的因素</i>															
7	轴参数设置错误, 0: 正确 (默认); 1: 错诉															
8	闭环控制标志位, 0: 开环 (默认); 1: 闭环															
9	电机伺服使能标志, 0: 伺服禁止 (默认); 1: 伺服使能															
10	轴运动状态标志位, 0: 轴正在运动; 1: 轴运动完成															
11	限位开关使能标志位, 0: 表示无效; 1: 表示有效(默认)															
12~13	轴号 <table border="1" style="margin-left: 40px;"> <thead> <tr> <th>BIT13</th> <th>BIT12</th> <th></th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>轴 1</td> </tr> <tr> <td>0</td> <td>1</td> <td>轴 1</td> </tr> <tr> <td>1</td> <td>0</td> <td>轴 1</td> </tr> <tr> <td>1</td> <td>1</td> <td>轴 1</td> </tr> </tbody> </table>	BIT13	BIT12		0	0	轴 1	0	1	轴 1	1	0	轴 1	1	1	轴 1
BIT13	BIT12															
0	0	轴 1														
0	1	轴 1														
1	0	轴 1														
1	1	轴 1														
14	HOME 捕获使能, 0: 禁止 (默认); 1: 使能															
15	INDEX 捕获使能, 0: 禁止 (默认); 1: 使能															

7.5 位置捕获

RDS3040G-PCI 提供了高速位置捕获功能，供用户应用于回零，外部触发等。

7.5.1 指令列表

表 7-12 Home/Index 捕获指令列表

指令	说明
<i>RD_SetCapt</i>	将指定控制轴的捕获触发源
<i>RD_GetCapt</i>	读取指定控制轴的捕获位置
<i>RD_SetCaptSns</i>	设置触发源的有效电平

7.5.2 函数说明

`RD_SetCapt(unsigned short axis, unsigned short homeindex)` 函数用于设置指定轴的捕获源，每个轴可以有多个触发源选择，如表 7-13。

表 7-13 捕获源选择

参数	触发源
0	INDEX 捕获
1	HOME 开关捕获
2	HOME+INDEX 捕获
3	外部 IO 捕获

`RD_SetCaptSns(unsigned short sns)` 函数用于设置捕获源输入的有效触发电平。如表 7-14 所示。

表 7-14 捕获源选择

参数		触发源
BIT0~BIT3	INDEX0~INDEX3 的有效触发电平	0: 下降沿触发 1: 上升沿触发
BIT4~BIT7	HOME0~HOME3 的有效触发电平	
BIT8	外部 IO 捕获的有效触发电平	

`RD_GetCapt(unsigned short axis, long *pos)` 函数返回各个轴的捕获位置，该位置为相对于机械原点的 32 位的绝对坐标。

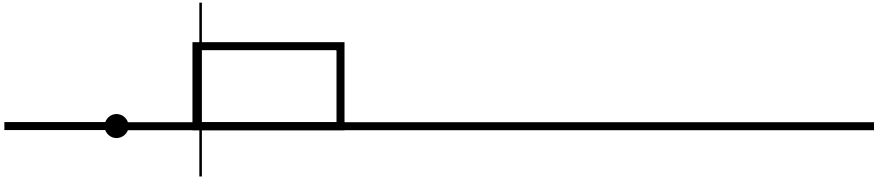
7.5.3 典型应用

- HOME 回原点：

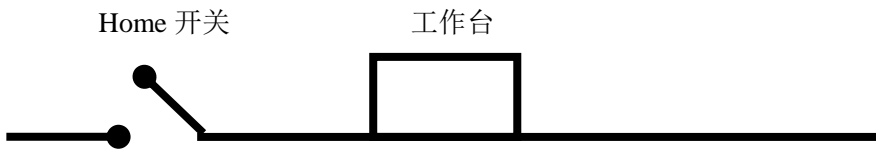
(1) 启动 Home 捕获，工作台向原点 (Home) 开关方向运动，



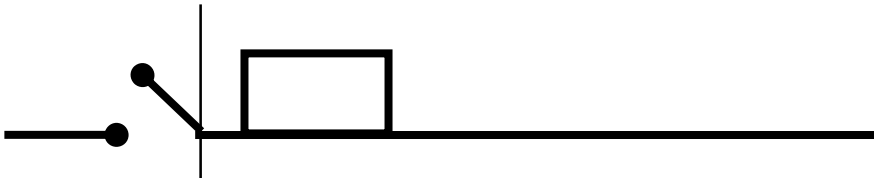
(2) 当 Home 信号产生时，平滑停止工作台；



(3) 调用指令 *RD_GetCapt()* 读取 Home 信号触发时工作台的实际位置，然后反向运动到回到该位置；



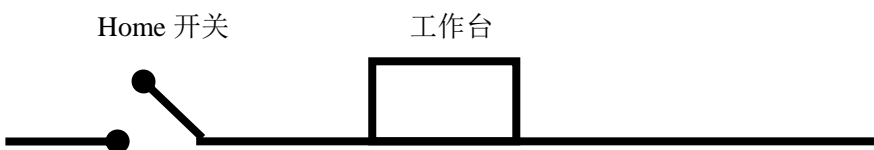
(4) 继续向前运动一段指定距离，离开 Home 开关，等工作台停稳以后调用指令 *RD_ZeroPos()* 将工



作台位置清零，建立机床坐标系。

● Home+Index 回原点

(1) 工作台向原点 (Home) 开关方向运动，启动 Home 捕获；



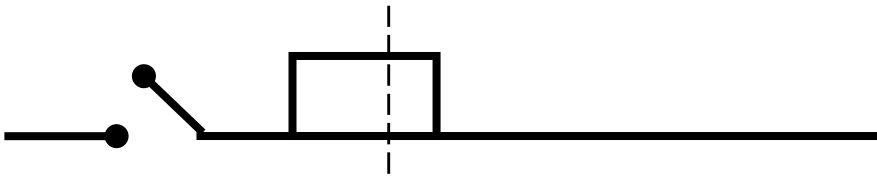
(2) 当 Home 信号产生时，平滑停止工作台；



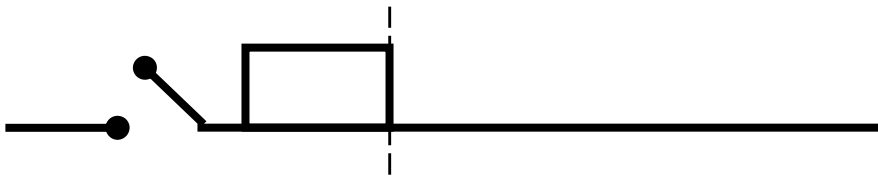
- (3) 调用指令 *RD_GetCapt()* 读取 Home 信号触发时工作台的实际位置，然后反向运动回到该位置；



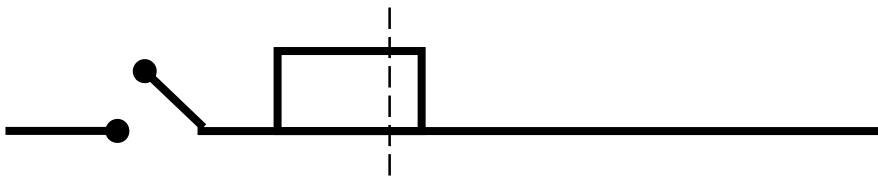
- (4) 启动 Index 捕获，继续运动一圈多一点，当 Index 信号产生时，平滑停止工作台；



- (5) 调用指令 *RD_GetCapt()* 读取 Index 信号触发时工作台的实际位置，然后反向运动到该位置；



- (6) 反向运动一段指定距离，消除丝杠的反向间隙，等工作台停稳以后调用指令 *RD_ZeroPos()* 将工作台位置清零，建立机床坐标系。



7.6 限位状态处理

运动控制器能够通过限位开关限制各控制轴的运动范围。装有限位开关的控制轴的“安全运动范围”如图所示：

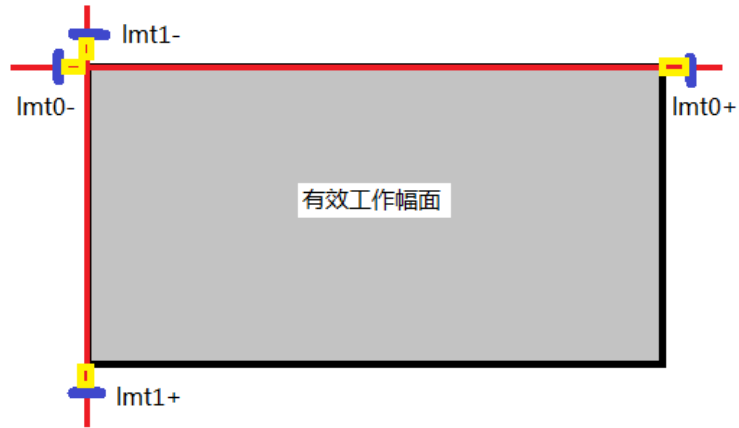


图 7-7 限位示意图

工作台碰到限位开关时，运动控制器紧急停止工作台的运动，此时会产生较大的冲击。建议用户不要使用限位开关回原点，而应当使用原点开关完成回原点的操作。

限位开关触发以后，运动控制器禁止向触发限位方向上继续运动，同时将控制轴状态寄存器的相应状态位置 1。离开限位开关回到安全运动范围以后，需要调用指令 `RD_ClrSts()` 将控制轴状态寄存器的限位触发状态位清 0，才能使控制轴从超限状态回到正常状态，否则控制轴不能正常运动。

7.7 控制轴驱动报警处理

运动控制器提供专用的驱动报警信号输入接口，若已经打开监视驱动报警信号功能，且检测到驱动器报警信号，运动控制器将控制轴状态寄存器的相应标志位置 1，紧急停止发生报警轴的运动，并关闭驱动器的伺服使能信号。

驱动器报警信号产生以后，应当执行以下操作：

1. 确定引起驱动器报警的原因，并加以改正；
2. 复位驱动器；
3. 重新回机床原点。

第八章 输入输出

8.1 函数列表

表 8-1 通用数字 I/O 指令

函数指令	说明
<i>RD_ExInpt</i>	立即指令，读取外部的 I/O 输入状态，隔离
<i>RD_ExOpt</i>	立即指令，直接 I/O 输出，隔离
<i>RD_ExOptEx</i>	立即指令，直接 I/O 输出，隔离
<i>RD_ExOptList</i>	缓冲区内输出通用 IO
<i>RD_ExOptExList</i>	缓冲区内输出通用 IO
<i>RD_ExDAList</i>	缓冲区内向直接向指定通道输出 DA 值
<i>RD_GetAD</i>	获取指定通道的 AD 转换值

8.2 函数说明

运动控制器提供非隔离的 16 路通用数字量输入和 16 路通用数字量输出接口。

调用指令 *RD_ExInpt()* 可以读取 16 路通用数字量输入。指令参数的状态位与主卡上外部输入接口 JP15 的对应关系如表 6-2 所示：

表 6-2 通用 I/O 输入接口定义

状态位	15	14	13	12	11	10	9	8
CN6	EXI15	EXI14	EXI13	EXI12	EXI11	EXI10	EXI9	EXI8
状态位	7	6	5	4	3	2	1	0
CN6	EXI7	EXI6	EXI5	EXI4	EXI3	EXI2	EXI1	EXI0

调用指令 *RD_ExOpt()* 可以设置 16 路通用数字量输出。指令参数的状态位与主卡上外部输出接口 CN9 的对应关系如表 6-3 所示：

表 6-3 通用 I/O 输出接口定义

状态位	15	14	13	12	11	10	9	8
CN6	EXO15	EXO14	EXO13	EXO12	EXO11	EXO10	EXO9	EXO8
状态位	7	6	5	4	3	2	1	0
CN6	EXO7	EXO6	EXI5	EXO4	EXO3	EXO2	EXO1	EXO0

调用指令 *RD_ExOptEx()* 可以设置 12 路通用数字量输出。指令参数的状态位与主卡上外部输出

接口 CN10 的对应关系如表 6-3 所示:

表 6-3 通用 I/O 输出接口定义

状态位	15	14	13	12	11	10	9	8
CN6	EXO15	EXO14	EXO13	EXO12	EXO11	EXO10	EXO9	EXO8
状态位	7	6	5	4	3	2	1	0
CN6	EXO7	EXO6	EXO5	EXO4	EXO3	EXO2	EXO1	EXO0

8.3 辅助编码器接口

板上 JP17 端子提供了一路辅助编码器接口供用户使用，示意图如下:

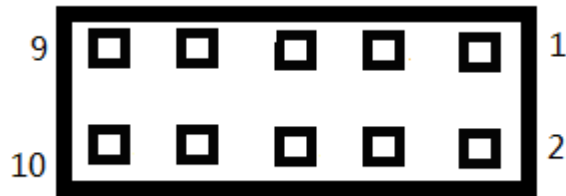


图 6.1 JP17 端子示意图

表 6.5 编码器输入接口信号表

引脚	1	2	3	4	5	6	7	8	9	10
定义	A+	A-	B+	B-	C+	C-	GND	VCC	NC	NC

引脚中的 VCC 为电源输出，最大驱动能力 200 毫安。用户正确的连接编码器后，可以通过函数

RD_GetEncEx() 获取辅助编码器的计数值，该计数值的编码位：16 位循环计数，计数范围为

0000~FFFF~0000~FFFF。

第九章 辅助功能

为了方便用户在系统开发的时候进行系统参数的管理以及其他信息的管理，本控制器板上提供了可永久保存的存储器供用户使用，用户可以利用它保存参数或者进行系统产品的加密管理等。

另外，本控制器还提供了实时时钟的功能，用户可以获取当前的实时时间，该时间不依赖于计算机的系统时间。用户利用该功能可以进行系统产品的管理。

参数操作函数有两个：

- *Rd_ReadPara(unsigned short num)*
- *Rd_WritePara(unsigned short num, unsigned long para);*

*Rd_ReadPara(unsigned short num)*参数为参数号，用户可以根据自己的需要定义为不同的含义。函数返回值为 32 位的无符号整数。为该参数号对应下的参数值。

同样的，*Rd_WritePara(unsigned short num, unsigned long para)*为写参数函数，函数第一个参数为参数号，第 2 个参数为需要写入的参数值。

RDS3040G-PCI 允许写入的参数空间为 0~0x400 (1024) 个参数。大于此参数范围，数据写入无效。

时钟操作函数有两个：

- *Void RD_WriteRTTime(SDDATATIME time)*
- *Void RD_ReadRTTime(SDDATATIME *time)*

参数的数据结构如下：

```
typedef struct _DATATIME_
{
    Uint8 year;
    Uint8 month;
    Uint8 day;
    Uint8 week;
    Uint8 hour;
    Uint8 minute;
    Uint8 second;
}SDDATATIME;
```

说明：其中 SDDATATIME 的数据格式如下：

如：2012 年 5 月 25 日周 3,12 点 30 分 55 秒，则直接把 SDDATATIME 的参数按照十进制赋值即可。

则按照以上的数据格式，则数据参数表示为：

```
Year = 12
Month=5
Day=25
Week=3
Hour=12
Minute=30
Second=55
```

同样的，从板卡读回的数据也符合十进制的数据格式。

第十章 激光函数列表（仅适用于 RDC3020G-PCI）

RDS3040G-PCI 控制器不具备专用的激光控制功能。

RDC3020G-PCI 控制器除了基本的运动控制功能外，还具有高速 IO 输出控制以及 PWM 输出功能，这个功能常用于激光控制。

10.1 函数列表

表 10-1 激光控制函数列表

函数指令	说明
RD_SetPwmTimeBase	设置 PWM 输出的时基
RD_SetLaserMode	设置激光控制模式，CO2 还是 YAG 模式
RD_SetLaserPwm	设置输出 PWM 参数
RD_SetLaserFps	设置 YAG 模式下的首脉冲抑制信号的参数
RD_SetStandBy	设置 CO2 模式下的待机 PWM 输出
RD_SetLaserDelay	设置激光开关延时
RD_LaserEnable	使能激光控制输出
RD_LaserDisable	禁止激光控制输出
RD_SetLaserOn	开激光
RD_SetLaserOff	关激光
RD_SetLaserFollow	设置激光控制 PWM 的速度跟随
RD_SetLaserFollowList	缓冲区内设置激光控制 PWM 的速度跟随
RD_SetLaserDA	设置激光功率电流模拟电压输出
RD_SetLaserDAEx	设置激光频率模拟电压输出

10.2 激光控制原理

对于激光控制，必须有一个 IO 信号来控制激光器的开关，另外还需要一个控制激光器输出功率的信号，这个信号可能是 PWM 信号，也可能是模拟电压信号。

激光器类型一般有 CO2 和 YAG 以及光纤激光器类型。调用函数 `RD_SetLaserMode()` 可以设置激光控制信号的类型。

`RD_LaserEnable()` 和 `RD_LaserDisable()` 用于控制激光控制总使能。只有使能的情况下，调用其它的函数设置的激光控制才有效，激光控制信号才有输出，如果禁止输出，则所有的激光控制信号输出为低电平。

`RD_SetPwmTimeBase()` 用于设置 PWM 输出的时基，这个参数和 PWM 的分辨率有关，用户要求 PWM 的变化量就是时基，举例如下：

假设用户需要的 PWM 输出的频率为 1KHz，这个参数确定了 PWM 的周期为 1000us；

如果用户选择的时基为 1MHz，则时基的最小单位为 1us，则需要往周期寄存器中填写 1000。此时 PWM 输出的最小脉冲宽度为 1us，即往脉宽寄存器中写 1。

假设用户需要的 PWM 输出频率为 1KHz，脉冲宽度最小为 1/8us；则此时用户必须把时基选择为 8MHz，由于时基为 8MHz，则要实现 1KHz 的频率，则需要往周期寄存器中写的的数据为 8000

10.2.1 CO₂ 激光器

CO₂ 激光器大致分为玻璃管和金属管两种，虽然类型不同，但控制信号大致相同。对于玻璃管激光器，需要的控制信号有：

- 开关信号
- PWM 或者模拟电压信号
- 待机脉冲

对于射频管，需要控制的信号有：

- 开关信号
- PWM
- 待机脉冲

`RD_SetLaserPwm()`用来设置 PWM 输出的参数，`RD_SetLaserDA()`用来设置模拟电压输出的参数。待机脉冲是在激光开关信号关闭的情况下输出的 PWM 信号。如图 10.1 所示。

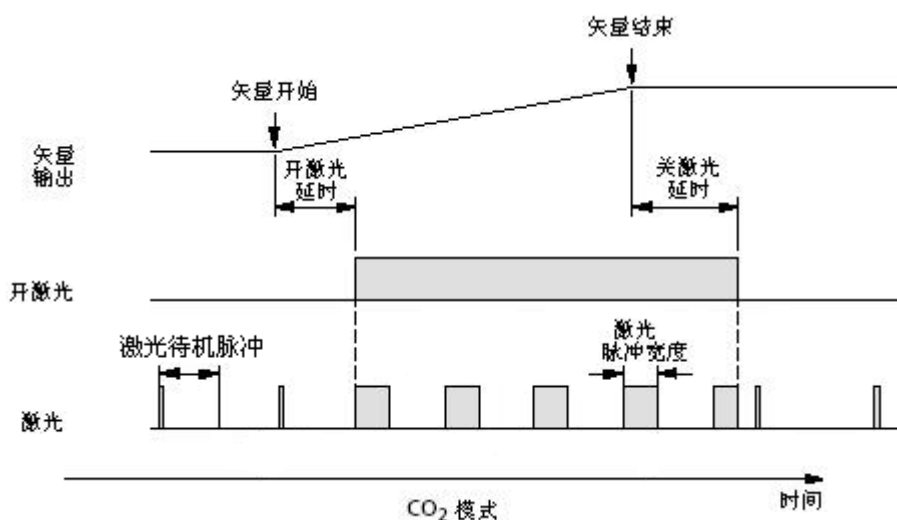


图 10.1 待机脉冲输出图

10.2.2 YAG 激光器

YAG 类型的激光器需要的控制信号如下：

- 激光开关信号
- Q 脉冲
- 首脉冲抑制信号
- 功率控制信号

要实现 YAG 激光器的控制，按照以下的步骤设置参数：

- (1) 调用 `RD_SetPwmTimeBase()` 设置时基
- (2) 调用函数 `RD_SetLaserMode()` 设置 YAG 类型
- (3) 调用函数 `RD_SetLaserPwm()` 设置 Q 脉冲的参数

- (4) 调用函数 `RD_SetLaserFps()` 设置 FPS 的参数
- (5) 调用函数 `RD_SetLaserDA()` 设置 YAG 激光器的电流

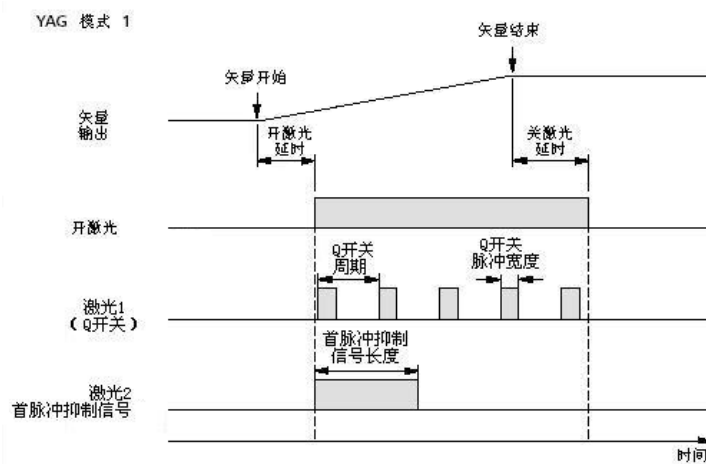


图 10.2 YAG 激光器的控制信号

FPS 信号的说明:

FPS 信号的输出通过调用 `RD_SetLaserFps()` 函数实现，该函数中有两个参数，一个是 FPS 信号的持续宽度 `width`，一个是 FPS 信号向对于激光开关信号的延时时间 `dly`，对于 `dly` 参数，可以为正也可以为负，当为正时，表示 FPS 信号滞后激光开关信号输出，当为负时，表示 FPS 信号超前激光开关信号输出。如下图所示：

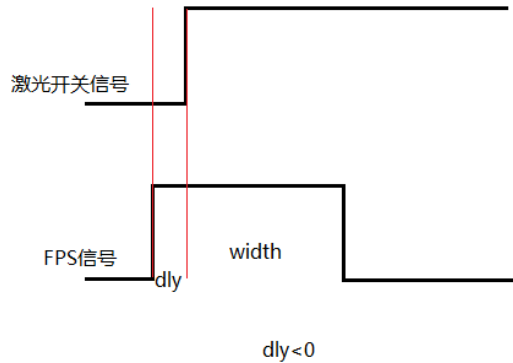


图 10.3 FPS 信号超前激光开关输出

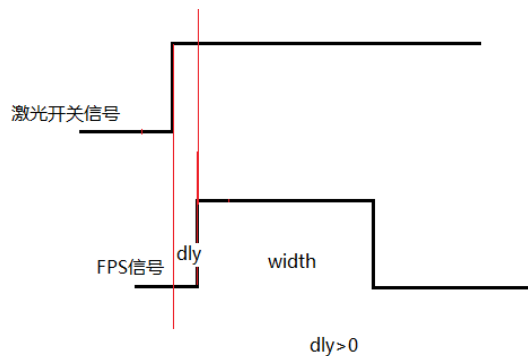


图 10.4 FPS 信号滞后激光开关输出

10.2.3 激光功率控制

根据系统运动的速度合理的控制激光输出的功率，是实现激光高质量切割的一个必要手段，RDS3040G-PCI 卡提供了相应的函数来实现该功能。

RD_SetLaserFollow()是立即指令，**RD_SetLaserFollowList()**是缓冲区指令，在该函数中有 3 个参数，一个是跟随系数，另外两个是最小功率和最大功率。合理的设置这三个参数，就可以实现功率输出密度均匀的切割效果。

第十一章 函数列表

指令	说明
初始化部分	
RD_Open	打开运动控制器
RD_Close	关闭运动控制器
RD_Reset	运动控制器复位
RD_HrdRst	硬件复位运动控制器
轴参数设置	
RD_SetAlarm	禁止或使能报警信号输入
RD_SetLmt	禁止或使能限位信号输入
RD_SetLmtSns	设置限位信号触发的有效电平
RD_SetEncoder	禁止或使能外部编码器输入
RD_SetEncSns	设置外部编码器的计数方向
RD_SetFilterPara	设置闭环伺服控制的轴滤波器参数
RD_SetAxisSysPara	设置伺服滤波器的极限参数
RD_StepOutMode	设置开环输出下的脉冲输出方式
RD_Update	同步的更新轴参数
RD_ZeroPos	指定轴的规划位置清零
RD_ClrSts	清除指定轴的轴状态寄存器
RD_AxisOn	指定轴使能
RD_AxisOff	指定轴禁止
RD_CtrlMode	设置指定轴是开环或闭环控制方式
RD_GetSts	获取轴的状态
硬件捕获功能	
RD_SetCapt	设置指定轴的捕获源
RD_SetCaptSns	设置指定轴捕获信号的有效电平
RD_GetCapt	获取捕获坐标值
硬件通道函数	
RD_GetLmtSwT	读取外部限位开关的电气状态

指令	说明
RD_GetLmtSts	获取是否有限位触发
RD_GetHomeSwT	获取外部 HOME 开关的电气状态
RD_GetAlmSts	获取是否有报警输入
RD_ExOpt	通用 16 位输出
RD_ExOptEx	通用扩展的 12 位输出
RD_ExInpt	通用 16 位输入
坐标系参数设置	
RD_CalVel	速度处理终点速度计算
RD_InitLookAhead	速度处理初始化函数
RD_AddLookData	速度处理数据压栈函数
RD_SetStrtVel	设置坐标系运动的起跳频率
RD_SetOverride	设置坐标系运动的速度倍率
RD_SetMaxVel	设置坐标系运动的最大速度
RD_SetSynVel	设置坐标系运动的速度
RD_SetDccVel	设置轨迹段的终点速度
RD_SetMapCnt	坐标系偏移指令
RD_SetAtlPos	设置指定轴的规划位置
立即运动指令	
RD_MoveToXYZA	4 轴直线立即运动指令
RD_MoveToXY	直线 2 轴立即运动指令
RD_MoveToXYZ	3 轴直线立即运动指令
缓冲区插补指令	
RD_LnXY	缓冲区直线插补指令
RD_LnXYZ	缓冲区直线插补指令
RD_LnXYZA	缓冲区直线插补指令
RD_LnXYZAG0	缓冲区直线插补空走指令
RD_LnXYG0	缓冲区直线插补空走指令
RD_LnXYZG0	缓冲区直线插补空走指令
RD_ArcXY	XY 平面缓冲区圆弧插补指令
RD_ArcZX	ZX 平面缓冲区圆弧插补指令
RD_ArcYZ	YZ 平面缓冲区圆弧插补指令

指令	说明
RD_ArcXYP	XY 平面缓冲区圆弧插补指令
RD_ArcYZP	YZ 平面缓冲区圆弧插补指令
RD_ArcZXP	ZX 平面缓冲区圆弧插补指令
缓冲区 IO 函数	
RD_BufIO	缓冲区 IO 输出函数
RD_BufIOBit	缓冲区 IO 指定位输出
RD_BufDA	缓冲区 DA 输出
RD_Delay	缓冲区延时
RD_LaserPreHeat	缓冲区内预热指令
缓冲区控制函数	
RD_StrtList	打开缓冲区
RD_EndList	关闭缓冲区
RD_AddList	重新打开缓冲区,追加缓冲区指令
RD_StrtMtn	启动缓冲区运动
RD_RestoreMtn	恢复缓冲区运动
RD_StpMtn	平滑停止缓冲区运动
RD_EStpMtn	紧急停止缓冲区运动
信息状态获取	
RD_GetPrfPnt	获取坐标系各个轴的规划位置
RD_GetPrfVel	获取坐标系运动的规划速度
RD_GetAdditionEnc	获取辅助编码器的 计数值
RD_GetBufCapacity	获取控制内部的指令缓冲区的空间
RD_GetBrkPnt	获取断点位置
RD_GetCurSegNum	获取当前正在执行的轨迹段号
RD_GetCrdSts	获取坐标系运动的状态
RD_GetCmdSts	获取指令执行的状态
辅助指令	
Rd_WritePara	向板卡内部存储器写参数
Rd_ReadPara	从板卡内部存储器读参数
RD_WriteRTTime	校准板卡上的日期和时间
RD_ReadRTTime	从板卡读取日期和时间

指令	说明
激光控制功能（仅 RDC3020G-PCI 有激光控制功能）	
RD_SetLaserDelay	设置激光开关延时
RD_SetLaserMode	设置激光控制模式
RD_SetStandBy	设置 CO2 模式下的待机脉冲
RD_SetPwmTimeBase	设置 PWM 输出的时基
RD_SetLaserPwm	设置 PWM 输出的参数
RD_SetLaserFps	设置 YAG 模式下的 FPS 信号的参数
RD_SetLaserFollow	设置激光功率跟随的参数
RD_SetLaserOn	激光开
RD_SetLaserOff	激光关
RD_LaserEnable	激光输出使能
RD_LaserDisable	激光输出禁止
RD_SetLaserFollowList	缓冲区内设置激光功率跟随系数

第十二章 函数详细说明

函数名称	RD_Open
指令类别	立即指令
适用控制器	RDS3040G-PCI
函数原型	short RD_Open(void)
功能描述	打开运动控制器，获得设备的访问权
参数	无
应用说明	<ul style="list-style-type: none"> ● 一定要打开控制器，否则控制器不能工作。 ● 只在初始化中调用一次。
参考指令	
参考例程	

函数名称	RD_Close
指令类别	立即指令
适用控制器	RDS3040G-PCI
函数原型	short RD_Close(void)
功能描述	关闭 PCI 卡在系统中的访问句柄
参数	无
应用说明	关闭应用程序时，调用该函数关闭通用运动控制卡。
参考指令	
参考例程	

函数名称	RD_Reset
指令类别	立即指令
适用控制器	RDS3040G-PCI
函数原型	short RD_Reset()
功能描述	复位运动控制卡，把系统的的变量和参数恢复初始状态，但并不复位 DSP 同时建立 PC 和控制卡的通讯。
参数	无
应用说明	<ul style="list-style-type: none"> ● 只在初始化中调用一次。 ● 如果返回值不为 0，则说明主机和板卡的通讯不正常。
参考指令	RD_HrdRst()
参考例程	

函数名称	RD_HrdRst
指令类别	立即指令
适用控制器	RDS3040G-PCI
函数原型	short RD_HrdRst()
功能描述	硬件复位 DSP，板卡全部复位
参数	无
应用说明	该指令要慎用，一旦调用该指令，板卡将硬件复位，DSP 程序重新执行
参考指令	
参考例程	

函数名称	RD_SetAlarm
指令类别	立即指令
适用控制器	RDS3040G-PCI
函数原型	short RD_SetAlarm(unsigned short axis,unsigned short alarm)
功能描述	设置报警信号输入有效或无效
参数	Axis: 轴号 Alarm: 使能或禁止, 0: 禁止; 1: 使能
应用说明	<ul style="list-style-type: none"> 调用该指令后，指定轴的报警信号将被禁止或使能，一旦被禁止，则无论 ALARM 引脚是什么电平，控制器将忽略。不影响系统的运动 该指令只能在初始化的时候调用，必须在运动完成，且没有调用 RD_AxisOn()函数使能轴之前
参考指令	
参考例程	

函数名称	RD_SetLmt
指令类别	立即指令
适用控制器	RDS3040G-PCI
函数原型	Short RD_SetLmt(unsigned short axis,unsigned short lmt)
功能描述	立即指令，设置指定轴的限位开关输入有效或无效
的参数	Axis: 轴号 lmt: 使能或禁止, 0: 禁止; 1: 使能
应用说明	<ul style="list-style-type: none"> 调用该指令后，指定轴的报警信号将被禁止或使能，一旦被禁止，则无论 lmt 引脚是什么电平，控制器将忽略。不影响系统的运动 该指令只能在初始化的时候调用，必须在运动完成，且没有调用 RD_AxisOn()函数使能轴之前
参考指令	

参考例程	
函数名称	RD_SetLmtSns
指令类别	立即指令
适用控制器	RDS3040G-PCI
函数原型	short RD_SetLmtSns(unsigned short sns)
功能描述	设置各个轴限位开关输入信号的触发有效电平
参数	Sns: 有效电平 Bit0~Bit3: 轴0~轴3的正限位输入触发, 1: 高电平有效, 0: 低电平有效 Bit8~Bit11: 轴0~轴3的负限位输入触发, 1: 高电平有效, 0: 低电平有效
应用说明	<ul style="list-style-type: none"> 该指令只能在初始化的时候调用, 必须在运动完成, 且没有调用 RD_AxisOn()函数使能轴之前 设置有效电平后, 轴状态中的限位触发标志不自动清除, 用户需要手动清除。
参考指令	
参考例程	

函数名称	RD_SetEncoder
指令类别	立即指令
适用控制器	RDS3040G-PCI
函数原型	short RD_SetEncoder(unsigned short axis,unsigned short onoff)
功能描述	禁止或使能外部编码器输入
参数	Axis:轴号 Onoff: 1: 禁止; 0: 使能
应用说明	<ul style="list-style-type: none"> 打开指定轴的编码器, 则表示运动控制器内部的实际位置计数器的计数源是外部编码器的脉冲, 如果禁止了编码器输入, 则表示控制器内部的实际位置计数器的计数源是控制器内部发出的电机脉冲 如果函数返回值为 1,则表示用户已经调用 RD_AxisOn()函数使能了轴控制。一旦使能了轴控制, 则不允许用户再执行轴通道的初始化函数 如果系统处于运动状态, 用户调用该函数, 系统也将报错。
参考指令	RD_GetAtlPos()
参考例程	

函数名称	RD_SetEncSns
指令类别	立即指令
适用控制器	RDS3040G-PCI
函数原型	short RD_SetEncSns(unsigned short axis, unsigned short mode)
功能描述	立即指令, 设置编码器的计数方向

参数	Axis:轴号 mode: 1: 反向计数; 0: 正向计数
应用说明	<ul style="list-style-type: none"> ● 正向: 编码器信号 A 超前 B, 反向: 编码器信号 B 超前 A ● 在做伺服闭环控制时, 必须谨慎的设置该参数, 以避免造成系统伺服控制的正反馈, 导致“飞车”现象 ● 一般在初始化时调用。必须在系统运动完成且轴没有被使能之前掉噢能够
参考指令	
参考例程	

函数名称	RD_CtrlMode
指令类别	立即指令
适用控制器	RDS3040G-PCI
函数原型	short RD_CtrlMode(unsigned short axis, unsigned short mode)
功能描述	设置指定轴是闭环伺服控制还是开环脉冲控制
参数	Axis:轴号 mode: 1: 闭环伺服控制; 0: 开环脉冲控制
应用说明	<ul style="list-style-type: none"> ● 该指令只能在初始化的时候调用, 必须在运动完成, 且没有调用 RD_AxisOn()函数使能轴之前 ● 调用该函数后, 其他可以直接输出电压的函数无效。
参考指令	
参考例程	

函数名称	RD_StepOutMode
指令类别	立即指令
适用控制器	RDS3040G-PCI
函数原型	short RD_StepOutMode(unsigned short axis, unsigned short mode)
功能描述	设置开环脉冲输出模式下, 轴脉冲的输出模式
参数	Axis: 轴号 Mode: 0: pulse + dir 1: +/-pulse
应用说明	该指令只能在初始化的时候调用, 必须在运动完成, 且没有调用 RD_AxisOn()函数使能轴之前
参考指令	
参考例程	

函数名称	RD_SetAxisSysPara
指令类别	立即指令
适用控制器	RDS3040G-PCI
函数原型	short RD_SetAxisSysPara(unsigned short axis, TSysPara *pSys)
功能描述	设置轴的闭环伺服控制极限参数
参数	Axis: 轴号 pSys: typedef struct { short mtrBias; //零飘电压补偿 short mtrLmt; //输出电压饱和极限 short mtrIlimt; //误差积分饱和极限 short posError; //跟随误差极限 }TSysPara;
应用说明	
参考指令	
参考例程	

函数名称	RD_SetFilterPara
指令类别	立即指令
适用控制器	RDS3040G-PCI
函数原型	short RD_SetFilterPara(unsigned short axis, TPID *pPID)
功能描述	设置闭环伺服滤波器的参数
参数	Axis: 轴号 pPID: typedef struct { double Kp; //比例系数 double Ki; //积分系数 double Kd; //微分系数 double Kvff; //速度前馈 double Kaff; //加速度前馈 }TPID;
应用说明	
参考指令	
参考例程	

函数名称	RD_Updt
指令类别	立即指令
适用控制器	RDS3040G-PCI
函数原型	Short RD_Update(unsigned short axis)
功能描述	轴闭环伺服控制参数同步刷新到运动控制器
参数	Axis: 轴号
应用说明	该函数为立即指令，调用 RD_SetFilterPara()函数设置参数后，必须调用该函数同步刷新到运动控制卡中。
参考指令	RD_SetFilterPara()
参考例程	

函数名称	RD_AxisOn
指令类别	立即指令
适用控制器	RDS3040G-PCI
函数原型	short RD_AxisOn(unsigned short axis)
功能描述	使能指定轴输出，并同步输出伺服使能信号
参数	Axis: 指定轴
应用说明	<ul style="list-style-type: none"> ● 当系统有报警或者限位触发时，RD_AxisOn()将无效，必须消除这些报警，才能使能轴的控制 ● 伺服使能信号输出为 24V 电平标准，开路输出，有效时三极管导通
参考指令	无
参考例程	

函数名称	RD_AxisOff
指令类别	立即指令
适用控制器	RDS3040G-PCI
函数原型	short RD_AxisOff(unsigned short axis)
功能描述	禁止指定轴输出，并同步输出伺服使能信号
参数	<ul style="list-style-type: none"> ● 伺服使能信号输出为24V电平标准，开路输出，有效时三极管截止
应用说明	。
参考指令	无
参考例程	

函数名称	RD_GetSts
指令类别	立即指令
适用控制器	RDS3040G-PCI
函数原型	short RD_GetSts(unsigned short axis,unsigned short *status)
功能描述	获取指定轴的轴状态
参数	Axis : 指定轴 Status: 轴状态, 轴状态的详细定义请参见表4-4
应用说明	
参考指令	无
参考例程	

函数名称	RD_ClrSts
指令类别	立即指令
适用控制器	RDS3040G-PCI
函数原型	short RD_ClrSts(unsigned short axis)
功能描述	清除轴状态寄存器的状态
参数	Axis : 指定轴
应用说明	报警信号或者限位信号一旦触发, 轴状态寄存器中的相关位将被置位。如果报警信号或者限位信号清除, 轴状态寄存器的相关位仍处于置位状态, 必须调用 RD_ClrSts()才能清除轴状态寄存器的相关位
参考指令	
参考例程	

函数名称	RD_ZeroPos
指令类别	立即指令
适用控制器	RDS3040G-PCI
函数原型	short RD_ZeroPos(unsigned short axis)
功能描述	规划位置、实际位置清零。
参数	Axis: 轴号
应用说明	
参考指令	无
参考例程	

函数名称	RD_SetCapt
指令类别	立即指令
适用控制器	RDS3040G-PCI
函数原型	short RD_SetCapt(unsigned short axis,unsigned short homeindex)
功能描述	设置指定轴的捕获源
参数	Axis: 指定轴号 Homeindex: 捕获源 0: INDEX 捕获 1: HOME 捕获 2: HOME+INDEX 捕获 3: 外部输入捕获
应用说明	
参考指令	
参考例程	

函数名称	RD_SetCaptSns
指令类别	立即指令
适用控制器	RDS3040G-PCI
函数原型	short RD_SetCaptSns(unsigned short sns)
功能描述	设置捕获源的有效电平
参数	Sns :捕获源的有效电平 详细请参见表7-14
应用说明	为了能正确地启动硬件捕获功能，必须根据外部接的信号输入的极性正确的设置捕获源的触发电平。
参考指令	
参考例程	

函数名称	RD_GetCapt
指令类别	立即指令
适用控制器	RDS3040G-PCI
函数原型	short RD_GetCapt(unsigned short axis, long *pos)
功能描述	获取指定轴的捕获位置
参数	Axis: 指定轴 Pos: 捕获位置，32位有符号数。
应用说明	获取的是外部触发信号触发的时刻，控制器捕获的控制轴的规划位置
参考指令	

参考例程	
------	--

函数名称	RD_ExOpt
指令类别	立即指令
适用控制器	RDS3040G-PCI
函数原型	Short RD_ExOpt(unsigned short output)
功能描述	立即指令，直接 16 路 IO 输出
参数	unsigned short output: 输出的16为数字量
应用说明	通用 IO 的输出操作，通用 IO 为集电极开路输出方式。
参考指令	
参考例程	

函数名称	RD_ExOptEx
指令类别	立即指令
适用控制器	RDS3040G-PCI
函数原型	Short RD_ExOptEx(unsigned short output)
功能描述	立即指令，直接 12 路扩展 IO 输出
参数	unsigned short output: 输出的12 位数字量
应用说明	通用 IO 的输出操作，通用 IO 为集电极开路输出方式。
参考指令	
参考例程	

函数名称	RD_ExInpt
指令类别	立即指令
适用控制器	RDS3040G-PCI
函数原型	short RD_ExInpt(unsigned short *input)
功能描述	立即指令，读取外部 IO 的输入状态
参数	unsigned short *input 返回外部IO的状态
应用说明	
参考指令	
参考例程	

函数名称	RD_GetLmtSwt
指令类别	立即指令
适用控制器	RDS3040G-PCI
函数原型	short RD_GetLmtSwt(unsigned short *lmtsts)
功能描述	立即指令，读取外部限位开关信号输入的状态
参数	unsigned short * lmtsts 返回外部限位的状态
应用说明	
参考指令	
参考例程	

函数名称	RD_GetLmtSts
指令类别	立即指令
适用控制器	RDS3040G-PCI
函数原型	short RD_GetLmtSts(unsigned short *lmtsts)
功能描述	立即指令，读取外部限位开关的触发状态
参数	unsigned short * lmtsts 返回限位开关触发的状态
应用说明	本函数和 RD_GetLmtSwt 不同，RD_GetLmtSwt 仅仅返回外部限位开关的物理状态，而 RD_GetLmtSts 返回的是实际是否有限位开关触发的状态，即这个函数返回的状态已经结合了 RD_SetLmtSns 设置的触发有效电平。
参考指令	
参考例程	

函数名称	RD_GetHomrSts
指令类别	立即指令
适用控制器	RDS3040G-PCI
函数原型	short RD_GetHomeSwt(unsigned short *homests)
功能描述	立即指令，读取外部 HOME 开关的输入状态
参数	unsigned short * homests 返回外部HOME开关的状态
应用说明	
参考指令	
参考例程	

函数名称	RD_GetAlmSts
指令类别	立即指令
适用控制器	RDS3040G-PCI
函数原型	short RD_GetAlmSts(unsigned short *almsts)
功能描述	立即指令，读取外部报警的输入状态
参数	unsigned short * almsts 返回外部报警的状态
应用说明	
参考指令	
参考例程	

函数名称	RD_InitLookAhead
指令类别	立即指令
适用控制器	RDS3040G-PCI
函数原型	Short RD_InitLookAhead(double T,double amax,double am,double vmax,short n,double con)
功能描述	初始化速度预处理模块的参数
参数	<p>T 拐弯时间。单位：S，用户在调用预处理功能指令时注意其时间单位的统一，即速度和加速度的时间单位均为 S，求出的终点速度也以秒为时间单位。 T 的经验范围：0.001~0.01S。 T 越大，计算出来的终点速度越大，但却降低了加工精度。反之，提高了加工的精度，但计算出的终点速度的偏低。因此要合理选择 T 值。</p> <p>Amax 系统能承受的最大加速度，单位：mm/s^2。根据不同的机械系统和电机驱动器取值不同，经验范围：$0.1\sim 10 \text{ mm/s}^2$。</p> <p>Am 系统设定加速度，单位：$\text{mm/s}^2$。 Vmax 用户加工的目标速度，单位：$\text{mm/s}$。 N 前瞻段数 Con 当量</p>
应用说明	<ul style="list-style-type: none"> ● 该函数属于速度预处理模块。 ● 设置时要特别注意参数的单位和控制器接收的指令的单位的对应关系。 ● 预处理缓冲区的大小最大为 100 段，缓冲区越大，运算量越大，则计算出来的终点速度也就越合理。用户应该合理的选取预处理缓冲区的大小。
参考指令	
参考例程	

函数名称	RD_AddLookData
指令类别	立即指令
适用控制器	RDS3040G-PCI
函数原型	short RD_AddLookData(char code, char plane_group, double r, double x, double y, double z, double vel, double cx, double cy, int i, long n, short flag)
功能描述	压入数据到预处理缓冲区
参数	<p>Code: 曲线类型。0: G00 直线 1: G01 直线 2: 顺圆弧 3: 逆圆弧</p> <p>plane_group: 平面代码。 17: XY 平面。</p> <p>r: 圆弧半径。如果当前轨迹指令为直线, 则该参数设置成为 0。</p> <p>x, y, z: 终点位置, 单位为 mm。如果只是 XY 平面的运动, 则 z 设置为 0。</p> <p>vel: 当前段的目标速度。单位 mm/s。</p> <p>cx, cy: 圆弧圆心位置, 若当前段不是圆弧, cx, cy 设为 0。</p> <p>i: 压入到预处理缓冲区的具体位置。</p> <p>n: 当前段的标识, 即用户加工代码的段号。</p> <p>flag: 当前段是否减速到 0, 一般情况下设为 0。当工艺特征需要系统减速到零停止时, flag 为 1。</p>
应用说明	<ul style="list-style-type: none"> ● 指令的返回值若为“0”表示预处理缓冲区接收成功, 为“1”表示该指令传送的位置数据与前次相同, 此时需要发下一条。 ● 整圆的起点和终点重合, 所以请将整圆分解为两个半圆进行处理。 ● 该指令与 RD_InitLookAhead(), RD_CalVelEx() 指令配合使用。 ● 该指令属于速度预处理模块, 其中预处理缓冲区为用户在计算机内存中开辟的存储单元。
参考指令	
参考例程	

函数名称	RD_CalVelEx
指令类别	立即指令
适用控制器	RDS3040G-PCI
函数原型	short RD_CalVelEx(double *vel, long *number, double *f)
功能描述	计算预处理缓冲区中当前轨迹段的终点速度和目标速度。指令返回值若为“0”表示预处理缓冲区中还存在需要处理的数据, 为“1”表示预处理结束。

参数	<p>vel: 返回终点速度。</p> <p>number: 对应的用户加工代码段号, 对应 RD_AddLookData()参数中的 n 参数。</p> <p>f: 当前轨迹的目标速度, 即在运动的过程中为了获得更好的系统特性需要对每一段轨迹都设置一个目标速度。</p>
应用说明	<ul style="list-style-type: none"> ● 计算每一轨迹段的终点速度和目标速度。 ● 预处理缓冲区的大小最大为 100 段, 缓冲区越大, 运算量越大, 则计算出来的终点速度也就越合理。用户应该合理的选取预处理缓冲区的大小。
参考指令	
参考例程	

函数名称	RD_StrtVel
指令类别	立即指令
适用控制器	RDS3040G-PCI
函数原型	short RD_SetStrtVel(double strtvel)
功能描述	设置坐标系运动的起跳速度
参数	Double strtvel: 起跳速度, 单位为pulse/ms
应用说明	<ul style="list-style-type: none"> ● 该指令用来设定在连续轨迹运动的启动速度, 启动速度与电机特性和机械系统特性有直接关系, 请用户根据实际系统进行设置。系统上电后, 运动控制器默认的起跳频率为 0Hz。 ● 对于一般的步进电机控制来说, 需要合理的设置其起跳频率
参考指令	
参考例程	

函数名称	RD_SetMaxVel
指令类别	立即指令
适用控制器	RDS3040G-PCI
函数原型	Short RD_SetMaxVel(double maxvel)
功能描述	该指令设定系统最高运行速度。单位是 pulse/ms
参数	double maxkvel 矢量运动的最大速度值
应用说明	<ul style="list-style-type: none"> ● 当用户设定速度大于该函数设定的最大速度时, 系统按此速度运行。 ● 一般在初始化时调用。
参考指令	
参考例程	

函数名称	RD_SetOverride
指令类别	立即指令
适用控制器	RDS3040G-PCI
函数原型	Short RD_SetOverride(double override)
功能描述	设置 RD_LnXY、RD_LnXYZ、RD_LnXYZA 的速度倍率。
参数	override: 速度倍率，一般是 100%~200%
应用说明	<ul style="list-style-type: none"> ● 调用该指令可以在线改变后续轨迹加工的进给速度，使设定速度改变为原设定速度的 override 倍，且一直有效，直到再次调用该指令。 ● 用户一旦调用该指令，立即生效，运动的速度会根据用户设定的倍率立即改变。
参考指令	
参考例程	

函数名称	RD_SetOverrideG0
指令类别	立即指令
适用控制器	RDS3040G-PCI
函数原型	Short RD_SetOverrideG0(double override)
功能描述	设置 RD_LnXYG0、RD_LnXYZG0、RD_LnXYZAG0 的速度倍率。
参数	override: 速度倍率，一般是 100%~200%
应用说明	<ul style="list-style-type: none"> ● 调用该指令可以在线改变后续空走运动的进给速度，使设定速度改变为原设定速度的 override 倍，且一直有效，直到再次调用该指令。 ● 用户一旦调用该指令，立即生效，运动的速度会根据用户设定的倍率立即改变。
参考指令	
参考例程	

函数名称	RD_StpAcc
指令类别	立即指令
适用控制器	RDS3040G-PCI
函数原型	short RD_SetStpAcc(double stpacc)
功能描述	设置系统的紧急停止加速度
参数	axis: 轴号 stpacc: 停止加速度

应用说明	<ul style="list-style-type: none"> ● 该指令为立即指令，一旦执行立即生效。 ● 当限位开关触发或者伺服报警时，系统将按照这个停止加速度停止坐标系的运动
参考指令	
参考例程	

函数名称	RD_SetSynVel
指令类别	缓冲区指令
适用控制器	RDS3040G-PCI
函数原型	short RD_SetSynVel(double synvel)
功能描述	设置坐标系运动的合成速度
参数	synvel合成速度 单位 pulse/ms
应用说明	
参考指令	
参考例程	

函数名称	RD_SetDccVel
指令类别	缓冲区指令
适用控制器	RDS3040G-PCI
函数原型	short RD_SetDccVel(double dccvel)
功能描述	设置轨迹段的终点速度
参数	dccvel: 终点速度
应用说明	该函数的终点速度值来自速度预处理模块
参考指令	
参考例程	

函数名称	RD_SetMapCnt
指令类别	立即指令
适用控制器	RDS3040G-PCI
函数原型	short RD_SetMapCnt(unsigned short axis, long cnt)
功能描述	坐标系偏移指令
参数	Axis: 轴号 cnt: 偏移的脉冲数
应用说明	
参考指令	
参考例程	

函数名称	RD_SetAtIPos
指令类别	立即指令
适用控制器	RDS3040G-PCI
函数原型	short RD_SetAtIPos(unsigned short axis,long pos)
功能描述	设置各个轴的实际位置
参数	Axis: 轴号 Pos: 位置值, 单位为脉冲
应用说明	
参考指令	无
参考例程	无

函数名称	RD_MoveToXY
指令类别	立即指令
适用控制器	RDS3040G-PCI
函数原型	short RD_MoveToXY(long x,long y,double vel,double acc)
功能描述	立即运动指令, 直线插补运动到目标位置
参数	X, Y 目标位置 Vel: 目标速度 Acc: 加速度
应用说明	
参考指令	无
参考例程	

函数名称	RD_MoveToXYZ
指令类别	立即指令
适用控制器	RDS3040G-PCI
函数原型	short RD_MoveToXYZ(long x,long y,long z,double vel,double acc)
功能描述	立即运动, 3 轴插补运动到目标位置
参数	X, Y, Z 目标位置 Vel: 目标速度 Acc: 加速度
应用说明	
参考指令	无

参考例程	
------	--

函数名称	RD_MoveToXYZA
指令类别	立即指令
适用控制器	RDS3040G-PCI
函数原型	short RD_MoveToXYZA(long x,long y,long z,long a,double vel,double acc)
功能描述	立即运动，4轴插补运动到目标位置
参数	X, Y, Z, A 目标位置 Vel: 目标速度 Acc: 加速度
应用说明	
参考指令	无
参考例程	

函数名称	RD_LnXY
指令类别	缓冲区指令
适用控制器	RDS3040G-PCI
函数原型	short RD_LnXY(long x,long y)
功能描述	缓冲区里的两轴连续轨迹插补指令
参数	X: 轨迹段x轴的目标位置 单位 pulse Y: 轨迹段y轴的目标位置 单位 pulse
应用说明	<ul style="list-style-type: none"> ● 直线轨迹加工指令 ● 参数 x、y 分别为相应控制轴的终点位置，单位是 pulse。直线插补运动的起点位置是前一段轨迹描述的终点位置（缓冲区命令）或者是当前位置（立即命令）。
参考指令	无
参考例程	

函数名称	RD_LnXYZ
指令类别	缓冲区指令
适用控制器	RDS3040G-PCI
函数原型	short RD_LnXYZ(long x,long y,long z)
功能描述	空间直线插补指令
参数	X 轨迹段x轴的目标位置 单位 pulse Y 轨迹段y轴的目标位置 单位 pulse Z 轨迹段Z轴的目标位置 单位 pulse

应用说明	<ul style="list-style-type: none"> ● 直线轨迹加工指令 ● 参数 x、y、z 分别为相应控制轴的终点位置，单位是 pulse。直线插补运动的起点位置是前一段轨迹描述的终点位置（缓冲区命令）或者是当前位置（立即命令）。
参考指令	
参考例程	

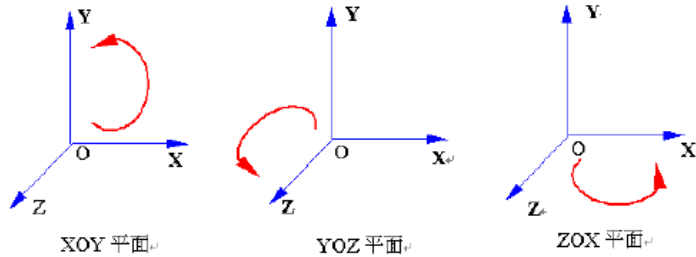
函数名称	RD_LnXYZA
指令类别	缓冲区指令
适用控制器	RDS3040G-PCI
函数原型	Short RD_LnXYZA(long x,long y,long z,long a)
功能描述	空间 4 轴直线插补指令
参数	X 轨迹段x轴的目标位置 单位 pulse Y 轨迹段y轴的目标位置 单位 pulse Z 轨迹段Z轴的目标位置 单位 pulse A 轨迹段A轴的目标位置 单位 pulse
应用说明	<ul style="list-style-type: none"> ● 直线轨迹加工指令 ● 参数 x、y、z、a 分别为相应控制轴的终点位置，单位是 pulse。直线插补运动的起点位置是前一段轨迹描述的终点位置（缓冲区命令）或者是当前位置（立即命令）。
参考指令	
参考例程	

函数名称	RD_LnXYG0
指令类别	缓冲区指令
适用控制器	RDS3040G-PCI
函数原型	short RD_LnXYG0(long x,long y)
功能描述	缓冲区里的两轴连续轨迹空走插补指令
参数	X 轨迹段x轴的目标位置 单位 pulse Y 轨迹段y轴的目标位置 单位 pulse
应用说明	<ul style="list-style-type: none"> ● 直线空走轨迹指令 ● 参数 x、y 分别为相应控制轴的终点位置，单位是 pulse。直线插补运动的起点位置是前一段轨迹描述的终点位置（缓冲区命令）或者是当前位置（立即命令）。
参考指令	无

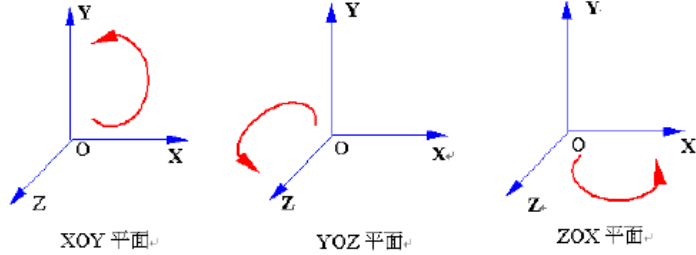
参考例程	
------	--

函数名称	RD_LnXYZG0
指令类别	缓冲区指令
适用控制器	RDS3040G-PCI
函数原型	short RD_LnXYZG0(long x,long y,long z)
功能描述	空间直线空走插补指令
参数	X 轨迹段x轴的目标位置 单位 pulse Y 轨迹段y轴的目标位置 单位 pulse Z 轨迹段Z轴的目标位置 单位 pulse
应用说明	<ul style="list-style-type: none"> ● 直线轨迹空走指令 ● 参数 x、y、z 分别为相应控制轴的终点位置，单位是 pulse。直线插补运动的起点位置是前一段轨迹描述的终点位置（缓冲区命令）或者是当前位置（立即命令）。
参考指令	
参考例程	

函数名称	RD_LnXYZAG0
指令类别	缓冲区指令
适用控制器	RDS3040G-PCI
函数原型	Short RD_LnXYZAG0(long x,long y,long z,long a)
功能描述	空间 4 轴直线插补指令
参数	X 轨迹段x轴的目标位置 单位 pulse Y 轨迹段y轴的目标位置 单位 pulse Z 轨迹段Z轴的目标位置 单位 pulse A 轨迹段A轴的目标位置 单位 pulse
应用说明	<ul style="list-style-type: none"> ● 直线轨迹空走指令 ● 参数 x、y、z、a 分别为相应控制轴的终点位置，单位是 pulse。直线插补运动的起点位置是前一段轨迹描述的终点位置（缓冲区命令）或者是当前位置（立即命令）。
参考指令	
参考例程	

函数名称	RD_ArcXY, RD_ArcYZ, RD_ArcZX
指令类别	缓冲区指令
适用控制器	RDS3040G-PCI
函数原型	short RD_ArcXY(long x_center, long y_center, double angle) short RD_ArcYZ(long y_center, long z_center, double angle) Short RD_ArcZX(long z_center, long x_center, double angle)
功能描述	该指令实现 XOY, YOZ, ZOZ 平面内的圆弧插补, 圆弧插补的起点位置时上一段轨迹的终点位置, 如果是立即指令, 则是当前位置。
参数	<p>x_center: 圆心坐标, 单位是PULSE y_center: 圆心坐标, 单位是PULSE z_center: 圆心坐标, 单位是PULSE Angle: 圆弧角度, 单位为度, 正负代表旋转方向, 旋转角度的取值范围为[-360, +360], 插补正方向的定义如下所示:</p>  <p>为了正确地描述圆弧运动, 将圆弧指令描述的轨迹放在正交的坐标系内, 则圆弧插补的正方向按照右手螺旋的定则定义为: “从坐标平面的上方 (即垂直于坐标平面的第3个轴方向)” 看, 逆时针方向为插补的正方向。</p>
应用说明	RDS3040G-PCI 目前版本仅支持二维的圆弧插补, 空间圆弧插补在后续版本推出
参考指令	
参考例程	

函数名称	RD_ArcXYP, RD_ArcYZP, RD_ArcZXP
指令类别	缓冲区指令
适用控制器	RDS3040G-PCI
函数原型	short RD_ArcXYP(long x_end, long y_end, double r, short dir) short RD_ArcYZP(long y_end, long z_end, double r, short dir) Short RD_ArcZXP(long z_end, long x_end, double r, short dir)
功能描述	该指令实现 XOY, YOZ, ZOZ 平面内的圆弧插补, 圆弧插补的起点位置时上一段轨迹的终点位置, 如果是立即指令, 则是当前位置。

参数	<p>x_end: 圆心坐标, 单位是PULSE y_end: 圆心坐标, 单位是PULSE z_end: 圆心坐标, 单位是PULSE Dir: 圆弧旋转方向, 1为正方向, -1为负方向 r: 半径, 有符号数, 正表示劣弧, 负表示优弧 正负代表旋转方向, 旋转角度的取值范围为[-360, +360], 插补正方向的定义如下所示:</p>  <p>为了正确地描述圆弧运动, 将圆弧指令描述的轨迹放在正交的坐标系内, 则圆弧插补的正方向按照右手螺旋的定则定义为: “从坐标平面的上方(即垂直于坐标平面的第3个轴方向)”看, 逆时针方向为插补的正方向。</p>
应用说明	RDS3040G-PCI 目前版本仅支持二维的圆弧插补, 空间圆弧插补在后续版本推出
参考指令	
参考例程	

函数名称	RD_BufIO
指令类别	缓冲区指令
适用控制器	RDS3040G-PCI
函数原型	Short RD_BufIO(unsigned short iostatus)
功能描述	缓冲区内直接 16 位的 IO 输出
参数	Iostatus: IO输出的状态
应用说明	
参考指令	
参考例程	

函数名称	RD_BufIOBit
指令类别	缓冲区指令
适用控制器	RDS3040G-PCI
函数原型	Short RD_BufIOBit(unsigned short bit,unsigned short bitstatus)
功能描述	缓冲区内设置某一个 IO 口输出

参数	Bit: 指定位 Bitstatus: 指定位的输出状态
应用说明	
参考指令	无
参考例程	

函数名称	RD_BufDA
指令类别	缓冲区指令
适用控制器	RDS3040G-PCI
函数原型	Short RD_BufDA(unsigned short davalue)
功能描述	缓冲区内输出指定通道 DA 值
参数	davalue : 16位的DA输出值
应用说明	通道仅限于第四通道
参考指令	
参考例程	

函数名称	RD_Delay
指令类别	缓冲区指令
适用控制器	RDS3040G-PCI
函数原型	short RD_Delay(unsigned short delay)
功能描述	缓冲区延时
参数	Delay: 运动延时, 单位为ms
应用说明	最大延时时间是 100000ms
参考指令	无
参考例程	

函数名称	RD_LaserPreHeat
指令类别	缓冲区指令
适用控制器	RDS3040G-PCI
函数原型	short RD_LaserPreHeat(unsigned short time)
功能描述	缓冲区内激光开启一段时间
参数	time: 单位为ms
应用说明	最大预热时间为 10000ms
参考指令	无
参考例程	

函数名称	RD_StrtList
指令类别	立即指令
适用控制器	RDS3040G-PCI
函数原型	short RD_StrtList (void)
功能描述	使能缓冲区输入
参数	
应用说明	该指令为立即指令，一旦执行立即生效。缓冲区指令才允许输入
参考指令	无
参考例程	

函数名称	RD_AddList
指令类别	立即指令
适用控制器	RDS3040G-PCI
函数原型	short RD_AddList (void)
功能描述	重新打开缓冲区,允许追加缓冲区指令输入
参数	
应用说明	<ul style="list-style-type: none"> ● 该指令为立即指令，一旦执行立即生效。缓冲区指令才允许输入 ● 该指令用于用户停止缓冲区运动后，运动停止后，板卡自动关闭缓冲区 ● 如果要继续恢复缓冲区的运动必须要调用该指令重新打开缓冲区，继续输入加工文件数据
参考指令	无
参考例程	

函数名称	RD_EndList
指令类别	立即指令
适用控制器	RDS3040G-PCI
函数原型	Short RD_EndList(void)
功能描述	禁止缓冲区输入
参数	
应用说明	一旦执行该指令，则后面的缓冲区指令都为非法指令，除非重新打开缓冲区输入
参考指令	无
参考例程	

函数名称	RD_StrtMtn
指令类别	立即指令
适用控制器	RDS3040G-PCI
函数原型	short RD_StrtMtn(void)
功能描述	启动缓冲区的运动
参数	
应用说明	该指令为立即指令，一旦执行立即生效。
参考指令	无
参考例程	

函数名称	RD_RestorMtn
指令类别	立即指令
适用控制器	RDS3040G-PCI
函数原型	short RD_RestoreMtn(void)
功能描述	继续恢复缓冲区的运动
参数	
应用说明	该指令为立即指令，一旦执行立即生效。
参考指令	无
参考例程	

函数名称	RD_StpMtn
指令类别	立即指令
适用控制器	RDS3040G-PCI
函数原型	short RD_StpMtn(void)
功能描述	平滑停止缓冲区运动
参数	
应用说明	该指令为立即指令，一旦执行立即生效。
参考指令	
参考例程	

函数名称	RD_EStpMtn
指令类别	立即指令
适用控制器	RDS3040G-PCI
函数原型	short RD_EStpMtn(void)
功能描述	紧急停止缓冲区运动

参数	
应用说明	该指令为立即指令，一旦执行立即生效。
参考指令	无
参考例程	

函数名称	RD_GetPrfPnt
指令类别	立即指令
适用控制器	RDS3040G-PCI
函数原型	short RD_GetPrfPnt(long *pnt)
功能描述	获取坐标系运动各个轴的规划位置
参数	pnt: 各个轴的规划位置
应用说明	该指令为立即指令，一旦执行立即生效。
参考指令	无
参考例程	

函数名称	RD_GetPrfVel
指令类别	立即指令
适用控制器	RDS3040G-PCI
函数原型	short RD_GetPrfVel(double *prfvel)
功能描述	获取当前坐标系运动的规划合成速度
参数	Prfvel: 规划速度，单位为pulse/ms
应用说明	
参考指令	无
参考例程	

函数名称	RD_GetAdditionEnc
指令类别	立即指令
适用控制器	RDS3040G-PCI
函数原型	short RD_GetAdditionEnc(unsigned short encNum, long *pos)
功能描述	读取辅助编码器的计数值
参数	encNum: 编码器编号 Pos: 位置值
应用说明	

参考指令	无
参考例程	

函数名称	RD_GetBufCapacity
指令类别	立即指令
适用控制器	RDS3040G-PCI
函数原型	short RD_GetBufCapacity(unsigned short *Capacity)
功能描述	获取指令缓冲区剩余空间
参数	Capacity: 指令缓冲区剩余指令空间
应用说明	
参考指令	无
参考例程	

函数名称	RD_GetBrkPnt
指令类别	立即指令
适用控制器	RDS3040G-PCI
函数原型	shor RD_GetBrkPnt(unsigned short axis,double *pnt)
功能描述	获取轴的断点位置
参数	Axis:轴号 Pnt: 位置
应用说明	运动暂停后, 控制器自动记忆当前的断点位置供用户查询
参考指令	无
参考例程	

函数名称	RD_GetCurSegNum
指令类别	立即指令
适用控制器	RDS3040G-PCI
函数原型	shor RD_GetCurSegNum(long *segNum)
功能描述	获取当前正在执行的段号
参数	segNum: 返回段号
应用说明	
参考指令	无
参考例程	

函数名称	RD_GetCrdSts
指令类别	立即指令
适用控制器	RDS3040G-PCI
函数原型	short RD_GetCrdSts(unsigned short *status)
功能描述	获取当前坐标系的运动状态
参数	status: 坐标系的运动状态
应用说明	坐标系运动的状态请参见表4-3
参考指令	无
参考例程	

函数名称	RD_GetCmdSts
指令类别	立即指令
适用控制器	RDS3040G-PCI
函数原型	short RD_GetCmdSts(unsigned short *status)
功能描述	获取当前指令执行的状态
参数	status: 指令出错原因
应用说明	详细定义情参见表 2-2
参考指令	无
参考例程	

函数名称	RD_WritePara
指令类别	立即指令
适用控制器	RDS3040G-PCI
函数原型	Short RD_WritePara(unsigned short num, unsigned long para)
功能描述	往板卡上的永久存储器中写参数
参数	unsigned short num: 参数号 参数号最大有1024个，超过1024个无效 unsigned long para: 参数值 参数值为无符号的32位整数。
应用说明	
参考指令	无
参考例程	

函数名称	RD_ReadPara
指令类别	立即指令
适用控制器	RDS3040G-PCI

函数原型	short RD_ReadPara(unsigned short num, unsigned long *para)
功能描述	读取板上永久存储器上对应参数号的参数值
参数	unsigned short num: 参数号, 最大1024, 超过1024无效 unsigned long *para: 返回的参数值
应用说明	
参考指令	无
参考例程	

函数名称	RD_ReadRTTime
指令类别	立即指令
适用控制器	RDS3040G-PCI
函数原型	short RD_ReadRTTime(SDDATATIME *time)
功能描述	读取板卡的当前的实时时间
参数	
应用说明	返回的数据格式如 SDDATATIME 结构所示。返回的数据按照 10 进制的方式存在。如 year=12 表示, 当前的年份为 2012 年。
参考指令	无
参考例程	

函数名称	RD_WriteRTTime
指令类别	立即指令
适用控制器	RDS3040G-PCI
函数原型	short RD_WriteRTTime(SDDATATIME time)
功能描述	设定实时时间
参数	SDDATATIME time: 设定时间, 结构如下 <pre> typedef struct _DATATIME_ { Uint8 year; //00~99; Uint8 month; // 1~12 Uint8 day; //1~31 Uint8 week; //0~6 Uint8 hour; //0~23 Uint8 minute; //0~59 Uint8 second; //0~59 }SDDATATIME; </pre>
应用说明	返回的数据按照 16 进制表示
参考指令	无

参考例程	
------	--

函数名称	RD_GetDSPVer
指令类别	立即指令
适用控制器	RDS3040G-PCI
函数原型	short RD_GetDSPVer (char ver[7])
功能描述	读取板卡 DSP 软件的版本
参数	char ver[7]: 版本号字符，格式为：ver[6].ver[5]ver[4].ver[3]ver[2].ver[1]ver[0]
应用说明	
参考指令	无
参考例程	

函数名称	RD_GetHdVer
指令类别	立即指令
适用控制器	RDS3040G-PCI
函数原型	short RD_GetHdVer (char ver[7])
功能描述	读取板卡硬件的版本
参数	char ver[7]: 版本号字符，格式为：ver[6].ver[5]ver[4].ver[3]ver[2].ver[1]ver[0]
应用说明	
参考指令	无
参考例程	

函数名称	RD_SetLaserDelay
指令类别	立即指令
适用控制器	RDS3040G-PCI
函数原型	short RD_SetLaserDelay(unsigned short ondly,unsigned short offdly);
功能描述	设置激光开关延时
参数	Ondly: 激光开延时，单位为us Offdly: 激光关延时，单位为us
应用说明	激光开关延时的参考点都是激光开关信号触发时的时刻。
参考指令	无
参考例程	

函数名称	RD_SetLaserMode
指令类别	立即指令
适用控制器	RDS3040G-PCI
函数原型	short RD_SetLaserMode(unsigned short mode)
功能描述	设置激光控制模式
参数	mode: 0: CO2 模式 1: YAG模式 2: 光纤模式
应用说明	对于 CO2 模式, 输出信号有开关信号, PWM 信号或者模拟电压信号, 待机脉冲。 对于 YAG 模式, 输出信号有开关信号, Q 脉冲信号, FPS 信号, 模拟电压信号
参考指令	无
参考例程	

函数名称	RD_SetStandBy
指令类别	立即指令
适用控制器	RDS3040G-PCI
函数原型	short RD_SetStandBy(unsigned short period,unsigned short width)
功能描述	设置激光控制模式
参数	Period: PWM周期 Width: PWM脉冲宽度
应用说明	设置 CO2 模式下待机脉冲的信号参数, 本信号仅在 CO2 模式下才有效。
参考指令	无
参考例程	

函数名称	RD_SetPwmTimeBase
指令类别	立即指令
适用控制器	RDS3040G-PCI
函数原型	short RD_SetPwmTimeBase(unsigned short timebase)
功能描述	设置 PWM 输出的时基

参数	0: 128KHZ 1: 256KHZ 2: 512KHZ 3: 1MHZ 4: 2MHZ 5: 4MHZ 6: 8MHZ 7: 16MHZ
应用说明	
参考指令	无
参考例程	

函数名称	RD_SetLaserPwm
指令类别	立即指令
适用控制器	RDS3040G-PCI
函数原型	short RD_SetLaserPwm(unsigned short period,unsigned short width)
功能描述	设置 PWM 输出的参数
参数	Period: PWM周期 Width: PWM脉冲宽度
应用说明	
参考指令	无
参考例程	

函数名称	RD_SetLaserFps
指令类别	立即指令
适用控制器	RDS3040G-PCI
函数原型	short RD_SetLaserFps(unsigned short width,unsigned short dly)
功能描述	设置 YAG 模式下 FPS 的输出参数
参数	width: FPS的有效宽度 dly: 延时时间
应用说明	延时时间是相对于激光开关信号有效的时刻
参考指令	无
参考例程	

函数名称	RD_SetLaserFollow
指令类别	立即指令
适用控制器	RDS3040G-PCI
函数原型	short RD_SetLaserFollow(double follow_ratio, double minpower, double maxpower)
功能描述	设置立即指令下的激光功率跟随参数
参数	follow_ratio: 跟随系数, 速度越高, PWM输出的脉冲宽度约越大 minpower: 最小功率占空比 Maxpower: 最大功率占空比
应用说明	
参考指令	无
参考例程	

函数名称	RD_SetLaserOn
指令类别	立即指令
适用控制器	RDS3040G-PCI
函数原型	short RD_SetLaserOn(void)
功能描述	开激光
参数	
应用说明	
参考指令	无
参考例程	

函数名称	RD_SetLaserOff
指令类别	立即指令
适用控制器	RDS3040G-PCI
函数原型	short RD_SetLaserOff(void)
功能描述	关激光
参数	
应用说明	
参考指令	无
参考例程	

函数名称	RD_LaserEnable
指令类别	立即指令
适用控制器	RDS3040G-PCI
函数原型	short RD_RD_LaserEnable (void)
功能描述	激光总使能
参数	
应用说明	
参考指令	无
参考例程	

函数名称	RD_LaserDisable
指令类别	立即指令
适用控制器	RDS3040G-PCI
函数原型	short RD_RD_LaserDisable (void)
功能描述	激光总禁止
参数	
应用说明	
参考指令	无
参考例程	

函数名称	RD_SetLaserFollowList
指令类别	缓冲区指令
适用控制器	RDS3040G-PCI
函数原型	short RD_SetLaserFollowList(double follow_ratio, double minpower, double maxpower)
功能描述	缓冲区内设置 PWM 输出的速度跟随系数
参数	
应用说明	
参考指令	无
参考例程	

深圳睿达科技有有限公司

地 址：深圳市南山区登良路天安南油工业区 2 座
501-502

电 话：0755-26066687

传 真：0755-26982287

电子邮件：support@rd-ac.com

网 址：<http://www.rd-ac.com>