



深圳市雷赛控制技术有限公司
SHENZHEN LEADSHINE CONTROL TECHNOLOGY CO.,LTD

PCI 总线 2 轴运动控制卡

DMC1220 软件手册

Version 1.0

2020.11.13

©Copyright 2020Leadshine Technology Co., Ltd.

All Rights Reserved.

版权说明

本手册版权归深圳市雷赛控制技术有限公司所有，未经本公司书面许可，任何人不得翻印、翻译和抄袭本手册中的任何内容。

涉及 DMC1220 运动控制卡硬件的详尽资料，请参阅 DMC1220 硬件手册。

本手册中的信息资料仅供参考。由于改进设计和功能等原因，雷赛公司保留对本资料的最终解释权，内容如有更改，恕不另行通知。



调试机器要注意安全！用户必须在机器中设计有效的安全保护装置，在软件中加入出错处理程序。否则所造成的损失，雷赛公司没有义务或责任对此负责。

目录

第一章 概述	1
第二章 软件安装.....	3
2.1 安装 DMC1220 运动控制卡驱动程序	3
2.2 安装 Motion1220 软件和编程示例.....	8
第三章 应用软件开发方法.....	9
3.1 基于 windows 平台的应用软件结构	9
3.3 Visual C++ 6.0 环境下的软件开发介绍	11
第四章 Motion1220 演示软件.....	13
4.1 参数设置操作.....	14
4.1.1 脉冲参数设置.....	14
4.1.2 回原点参数设置.....	15
4.1.3 计数参数设置.....	15
4.1.4 限位参数设置.....	16
4.2 I/O 检测操作	17
4.3 计数功能操作.....	18
4.4 运动功能测试操作.....	19
第五章 运动控制功能和相关函数.....	21
5.1 初始化、关闭控制卡.....	21
5.2 脉冲输出模式的设置.....	22
5.2.1 脉冲/方向模式	23
5.2.2 双脉冲模式.....	24
5.3 单轴位置运动和速度控制.....	26
5.3.1 梯形速度曲线运动.....	27
5.3.2 S 形速度曲线运动	32
5.3.3 连续运动.....	34
5.3.4 加减速过程的距离（脉冲数）计算.....	37
5.4 回原点运动.....	37
5.5 多轴运动控制.....	42
5.5.1 多轴联动.....	42
5.5.2 直线插补运动.....	43

5.5.3 圆弧插补.....	45
5.5.4 连续缓冲插补运动.....	47
5.6 位置计数.....	50
5.6.1 命令位置计数器.....	50
5.6.2 反馈位置计数器.....	51
5.6.3 位置锁存.....	53
5.7 通用 I/O 控制	55
5.8 伺服驱动器接口.....	56
5.8.1 INP 定位完成信号	57
5.8.2 ALM 伺服报警信号	58
5.8.3 ERC 误差清除信号.....	58
5.8.4 SEVON 使能伺服驱动器信号	59
5.8.5 RDY 伺服准备好信号	59
第六章 编程举例.....	61
6.1 Visual C++ 6.0 编程举例	62
6.2 Visual Basic 6.0 编程举例	66
第七章 运动函数说明.....	69
7.1 初始化函数.....	72
7.2 脉冲模式设置函数设置函数.....	73
7.3 速度设置函数.....	75
7.4 单轴位置运动控制函数.....	77
7.5 多轴直线插补运动控制函数.....	79
7.6 两轴圆弧插补函数.....	79
7.7 回原点函数.....	80
7.8 手轮运动控制函数.....	81
7.9 运动状态及轴信号检测函数.....	81
7.10 驱动器专用接口信号的设定函数.....	83
7.11 位置计数器控制函数.....	87
7.12 通用 I/O 控制函数	87
7.13 编码器相关函数.....	89

第一章概述

DMC1220 运动控制卡是雷赛公司自主研发的高性能 PCI 两轴运动控制卡。雷赛公司配套提供 Windows XP/7/10 等操作系统环境下的设备驱动程序和运动控制函数动态链接库。DMC1220 运动控制函数库是一个运动控制 API 函数库，在其基础上开发应用软件不再是一件很困难的事情了：您只要用 C++、Visual Basic、C#或 LabView 编写一些用户界面，并调用 DMC1220 函数库中的相关运动控制函数，您就可以随心所欲的对您的多轴自动化设备进行精确、高速、协调的控制。因为 DMC1220 的函数库能帮您处理所有与运动控制有关的复杂问题，这样您不必了解底层硬件细节就可以根据特定的应用要求像搭积木似的开发出自己的软件系统，从而大大缩短您的软件开发周期。

为协助用户更快掌握 DMC1220 的应用技巧、编写出更适合于特定自动化设备的应用软件，雷赛公司还随卡提供一个演示软件 Motion1220，它具有多种运动控制功能和测试功能。利用 Motion1220 软件，用户既可以快速地熟悉 DMC1220 运动控制卡的软硬件功能，又可以方便快捷地测试电机驱动系统在执行各种运动时的性能及特性。

另外，软件光盘上还提供了许多在 VB、VC 编程环境下控制单轴和多轴运动的例程，您可以将这些例程作为程序模块，直接拷贝至用户的应用软件中使用。

本手册为 DMC1220 软件编程使用手册，共分七章进行论述：

第一章：概述；

第二章：软件安装：描述软件安装过程，包括安装设备驱动程序，Motion1220 演示软件，运动控制函数库和编程方法；

第三章：应用软件开发方法：概述使用 VB 和 VC 调用 DMC1220 库函数开发应用软件的

一般步骤；

第四章：Motion1220 演示软件的使用：介绍如何使用 Motion1220 演示函数库中部分函数的功能，帮助客户快速调试机械和硬件系统；

第五章：介绍运动控制功能和相关函数：对 DMC1220 卡常用的运动和控制功能进行描述，并介绍如何调用函数库中的相关函数来实现这些功能；

第六章：编程举例：提供几个典型运动的编程示例及详细解释，供刚刚接触运动控制卡编程的用户参考；

第七章：分类详细介绍 DMC1220 运动函数库中所有函数，包括功能描述、调用参数、返回参数、调用语法等。

（硬件方面的内容请参考 DMC1220 硬件使用手册）

第二章 软件安装

雷赛公司为 DMC1220 运动控制卡用户提供以下软件：

1. 驱动程序；
2. 运动控制函数库；
3. 演示软件 Motion1220；
4. 多个 VB、VC 典型运动控制例程源代码。

2.1 安装 DMC1220 运动控制卡驱动程序

安装 DMC1220 运动控制卡驱动程序的过程与安装其它 PC 机的板卡（如 MODEM 卡，显卡等）的驱动程序十分相似。DMC1220 卡提供如下安装方法：

- 1、将 DMC1220 卡插入电脑 PCI 槽中，启动电脑会出现如下提示：

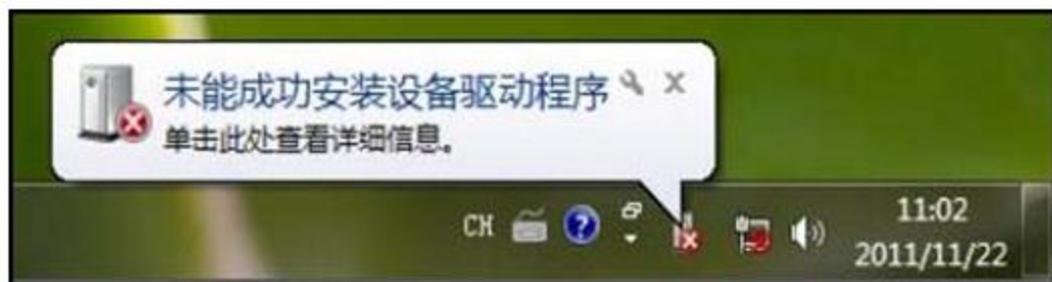


图 2-1 系统提示发现新硬件

- 2、找到驱动所在文件，根据电脑系统选择驱动文件



图 2-2

- 3、打开电脑系统对应的文件中“Reg_Install”，鼠标右键单击，“以管理员身份运行”，

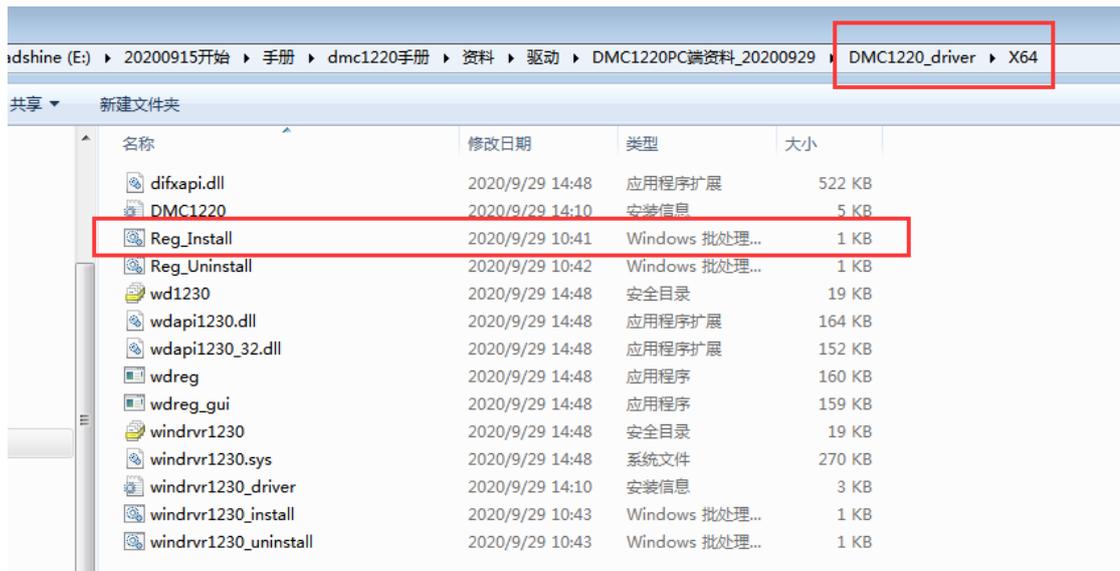


图 2-3

弹出如下窗口

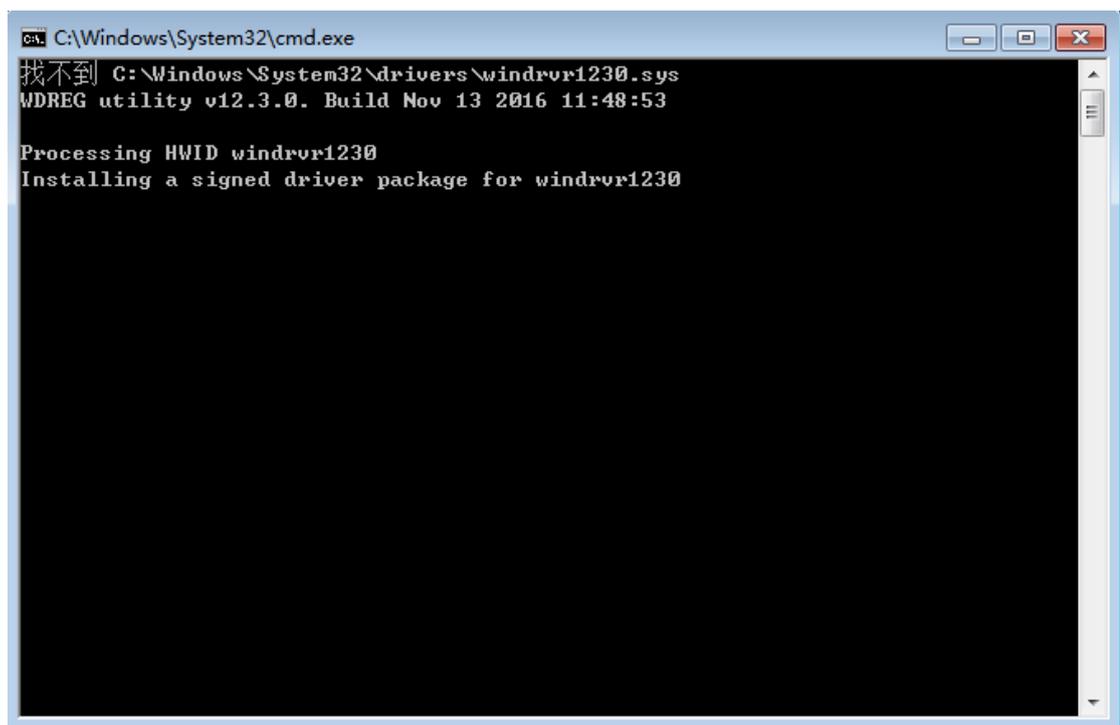


图 2-4



图 2-5

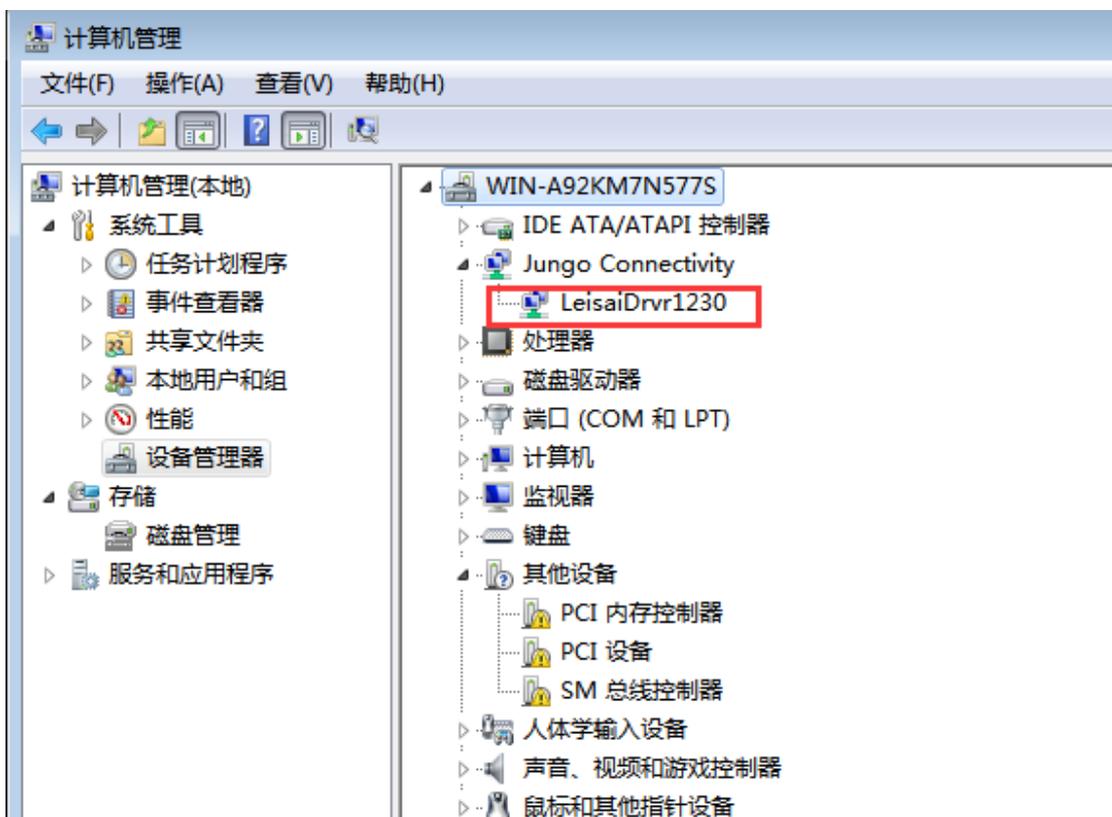


图 2-6

4、点击“安装”，cmd 刷新

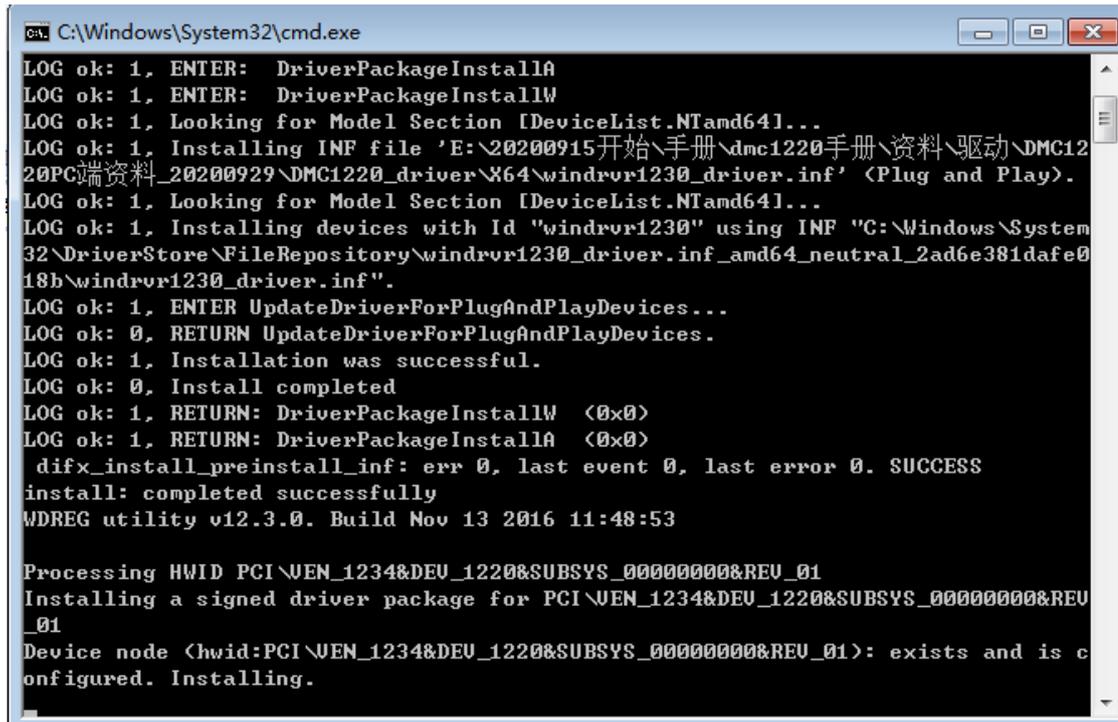


图 2-7

弹出如下窗口



图 2-8

点击“安装”，cmd 窗口刷新完成后自动关闭，此时驱动安装成功

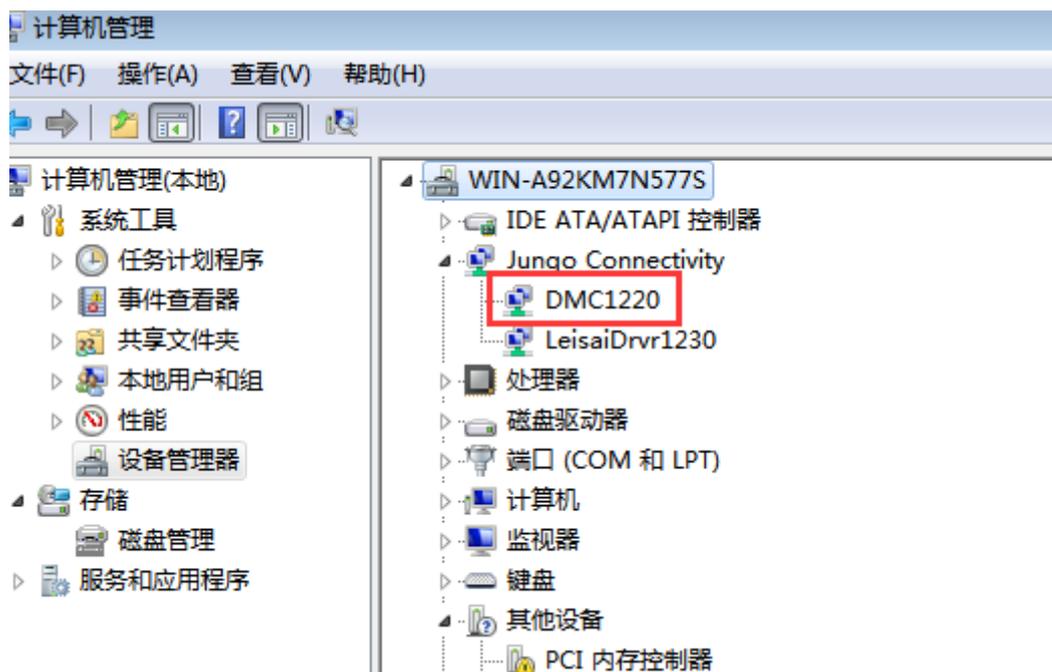


图 2-9

2.2 安装 Motion1220 软件和编程示例

将 DMC1220 的软件 CD 盘插进计算机光驱，在相应的目录 Motion1220 下，将其全部拷贝到计算机硬盘的任意指定位置后，运行 Motion1220.exe，即可对控制卡的各项主要功能进行检测、学习。

为了方便用户利用 VC 或 VB 等编程工具对 DMC1220 进行开发，针对典型的运动控制功能，如：单轴运动、回零动作、插补运动，编码器读取，数据锁存，位置比较等，雷赛公司提供了示例源代码，用户可以直接将软件 CD 中相应目录 sample 中的代码直接拷贝到您的程序工程中使用。

第三章应用软件开发方法

如果您对 C、C++、Visual Basic 等程序语言一点都不了解，建议您先花几天时间阅读一本该语言的培训教材，并且通过一些简单练习掌握该语言的基本技能，例如：如何编写简单的程序，如何创建窗体和调用函数。

如果您曾用 C、C++、Visual Basic 等程序语言进行过运动控制软件的开发，并具有丰富的经验，那么可以直接阅读第七章“运动函数说明”，查阅所需函数的详细信息。

3.1 基于 windows 平台的应用软件结构

使用雷赛控制卡的机器控制系统构架如图 3-1 所示：

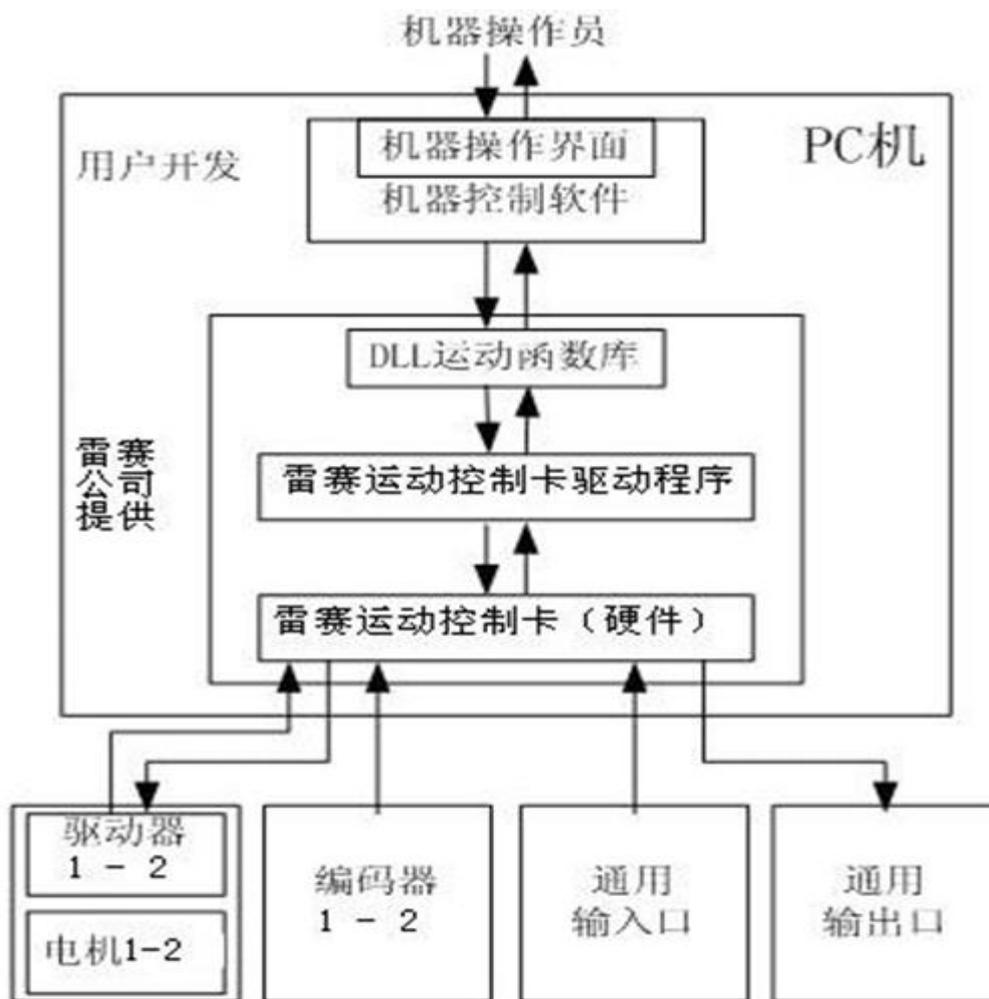


图 3-1 基于雷赛控制卡的机器控制系统构架

从上面的示意图可以看出，控制系统的工作原理可以简单描述为：

1. 操作员的操作信息通过操作接口（包括显示屏和键盘）传递给设备的控制软件；
2. 设备的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. 运动函数调用雷赛运动控制卡驱动程序发出控制指令给控制卡；
4. 雷赛运动控制卡再根据控制指令发出相应的驱动信号（如脉冲、方向信号）给驱动器及电机、读取编码器数据、读写通用输入输出。

用户在开发应用软件（即设备的控制软件）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。雷赛公司已提供支持 DMC1220 运动控制卡的硬件驱动程序和 DLL 运动函数库，包括控制卡初始化函数、单轴及多轴控制函数、输入/输出脉冲模式设置函数等许多函数。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、Visual C++、Visual Basic、LabView、Delphi 等程序语言调用这些函数来快速开发出自己的应用软件。

用户编写的机器控制软件的典型流程如图 3-2 所示：

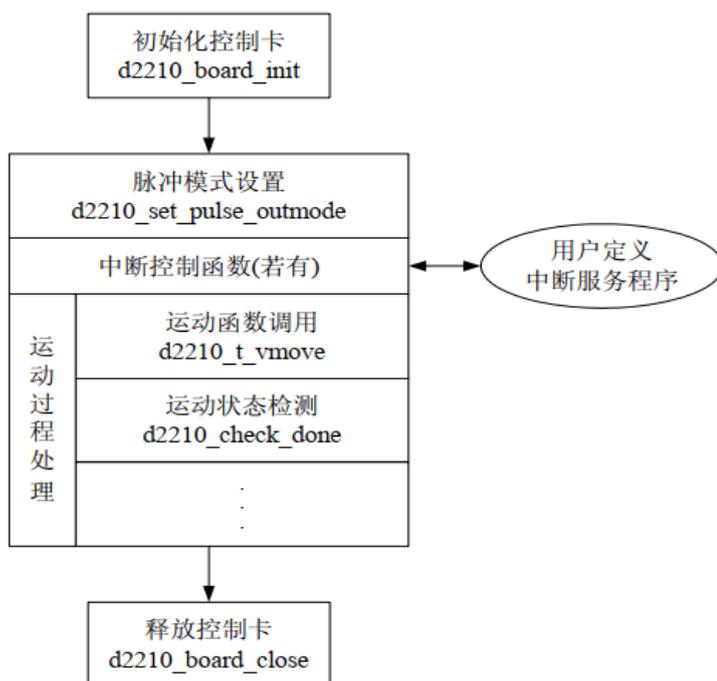


图 3-2 控制软件的典型流程

请确保 DMC1220 运动控制卡已经插入到你的计算机插槽中，安装好驱动程序、Motion1220 演示软件和 VB 编程环境，在调用 DMC1220 运动函数之前，需要做下面几项工作：

- 1 启动 Motion1220 演示软件，进行运动控制卡控制功能的简单测试，如：单轴定长运动等，以确定 DMC1220 运动控制卡软硬件安装正常。
- 2 建立自己的工作目录，如：d:\vbMotion（注：此目录名可以自己指定）。
- 3 将 DMC1220.bas 文件拷贝到该目录下（此文件在软件 CD 的 module 目录下可以找到）。
- 4 运行 VB，并建立一个工程，然后保存此新建的工程在 vbMotion 目录中。
- 5 按下述步骤，将运动函数库链接到你的工程项目中：
 - 5.1 在 VB 编译器的“工程（P）”菜单中选择“添加模块”；
 - 5.2 选择“现存”；
 - 5.3 选择“DMC1220.bas”；
 - 5.4 选择“确定”。

当您将运动函数链接到你的工程项目中后，就可以象调用其它 API 函数一样直接调用运动

函数，每个函数的具体功能，请参考第七章“运动函数说明”，当然还可以打开模块文件 DMC1220.bas 了解每个函数的具体定义。

在编程过程中，你可以参阅我们提供的运动函数编程实例：我们提供了 VB 的编程实例源代码，存放在光盘的 Samples 目录下。只要你将控制卡及其驱动程序安装好，并安装好 VB 编译器，即可直接运行这些源代码。

3.3 Visual C++ 6.0 环境下的软件开发介绍

请确保 DMC1220 运动控制卡已经插入到你的计算机插槽中，安装好驱动程序、Motion1220 测试软件和 VC，在调用 DMC1220 运动函数之前，需要做下面几项工作：

1. 启动 Motion1220 演示软件，进行运动控制卡控制功能的简单测试，如：单轴定长运动等，以确定 DMC1220 运动控制卡软硬件安装正常。

2. 运行 VC，并建立一工程，将工程命名为 `vcMotion`(可自定义)，路径为：`d:\vcMotion`；
3. 将 `DMC1220.lib` 和 `DMC1220.h` 文件拷贝到该目录下（此文件在 `module` 目录下）；
4. 将运动函数链接到你的工程项目中，将 `DMC1220.lib` 加入到工程中；
5. 在调用运动函数的文件头部代码中加入 `#include "DMC1220.h"` 语句。

当你将运动函数链接到你的项目中后，你就可以象调用其它 API 函数一样，调用运动函数，每个函数的具体功能，请参考第七章“运动函数说明”，当然，还可以打开头文件 `DMC1220.h` 了解每个函数的具体定义。

在编程过程中，你可以参阅我们提供的运动函数编程实例：我们提供了 VC 的编程实例源代码，存放在光盘的 `Samples` 目录下。只要你将控制卡及其驱动软件安装好，并安装好 VC 编译器，即可直接运行这些源代码。

第四章 Motion1220 演示软件

Motion1220 是雷赛公司为了便于用户熟悉 DMC1220 运动控制卡运动功能和相关函数开发的一个演示测试软件。利用这个软件，用户既可以很快地熟悉 DMC1220 卡的软硬件功能，又可以方便快捷地测试电机驱动系统执行各种运动时的性能特点。

当您将 DMC1220 运动控制卡的软硬件正确地安装到 PC 机上后，启动计算机，将软件 CD 上的演示软件拷贝到计算机上并运行该软件。Motion1220 软件提供了参数设置、I/O 检测、计数功能、运动测试，四个主要的操作界面。根据界面的信息，您可以进行一些基本的控制操作：比如简单的点位运动、I/O 信号检测等。

有关 Motion1220 的一些基本功能以及硬件接线的说明，请仔细阅读 DMC1220 硬件手册。

启动 Motion1220 软件后，首先进入演示软件主界面如图 4-1 所示，用户首先要选择适当的卡号，然后对相应的卡进行操作；只安装了一块 Motion1220 卡时卡号为 0 号。在此界面上可以通过单击其上按钮，选择进入下一级子界面。



图 4-1 演示软件主界面

4.1 参数设置操作

DMC1220 运动控制卡参数设置由五个子界面组成，分别是脉冲设置、回原点设置、计数设置、限位设置以及备用设置。按“加载”键，将计算机当前设置的数据加载到运动卡中；按“保存”键，当前控制卡中的数据被保存在计算机的 DMC1220 测试软件中。

4.1.1 脉冲参数设置

脉冲参数设置包括：脉冲输出类型设置、脉冲有效电平的设置和方向控制逻辑电平设置，如图 4-2 所示。



图 4-2 脉冲设置子界面

指令脉冲类型：也即脉冲输出类型，用户可以设置为正脉冲/负脉冲模式（CW/CCW 模式）或脉冲/方向模式（pulse/dir 模式）；该参数由用户使用的驱动器能够接收的脉冲类型来决定。

脉冲输出有效电平：选择下降沿有效时，脉冲停止时的状态为高电平；选择上升沿有效时，脉冲停止时状态为低电平。

方向控制逻辑电平：用户可以设置某一电平状态对应为电机的正转方向或反

转方向。该参数可使用户不改变硬件接线，就可以改变电机的运动方向。

4.1.2 回原点参数设置

回原点参数设置包括：原点信号有效电平、回原点速度方式和回原点运行方式，如图 4-3 所示。

原点信号有效电平：设置其有效电平根据传感器的实际电路设置；回原点速度方式：分为常速方式和快速方式两种方式；

回原点运行方式：有只找原点信号和原点+EZ 信号两种方式。只找原点方式是从一个默认的方向回原点，找到原点后立刻停止；原点+EZ 方式是从一个默认的方向回原点，找到原点后，再以慢速返回，直到 EZ 信号有效后，立刻停。

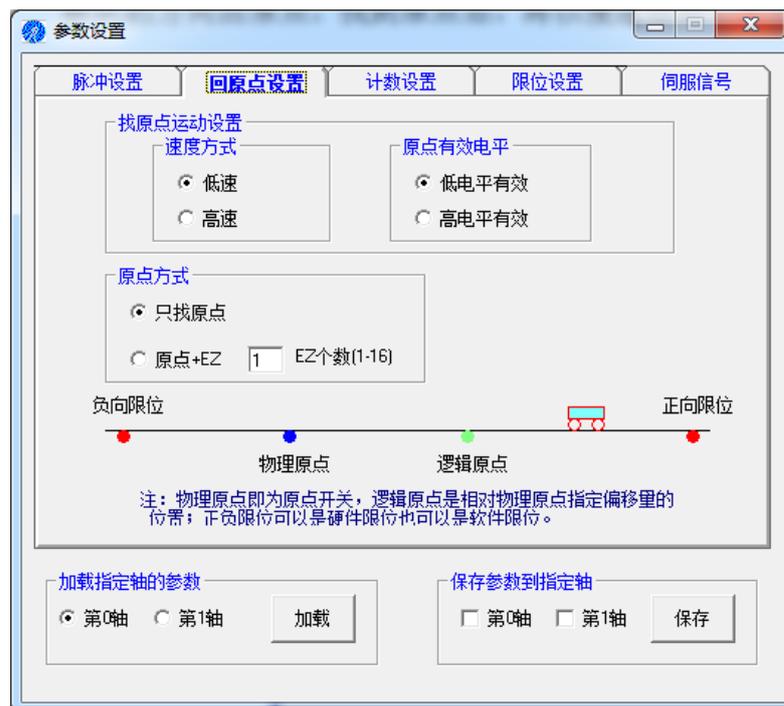


图 4-3 回原点设置子界面

4.1.3 计数参数设置

计数参数设置包括：脉冲输入模式、EZ 信号设置、触发方式，如图 4-4 所示。



图 4-4 计数设置子界面

脉冲输入模式的设置：指的是输入到编码器接口的信号类型，为 AB 相方式。

EZ 信号设置：分为有效电平的设置和是否自动清零编码器的计数器的设置。

触发方式设置：设置触发端口 LTC 的有效电平为高或低，单独锁存一个轴的数据还是全部轴。

4.1.4 限位参数设置

限位参数设置包括：响应方式的设置和有效电平的设置，如图 4-5 所示。



图 4-5 限位设置子界面

硬件限位制动方式：可以设置外部限位开关触发后，是立即停止或是减速后停止。

限位开关有效电平：是高有效还是低有效。

4.2 I/O 检测操作

I/O 信号检测操作界面用于观察各专用 I/O 的实时状态，每一个指示灯对应一个 I/O 信号的状态，绿色表示 ON，红色表示 OFF；还可以通过按钮设置每一个通用输出口的电平。

I/O 检测子界面如图 4-6 所示：

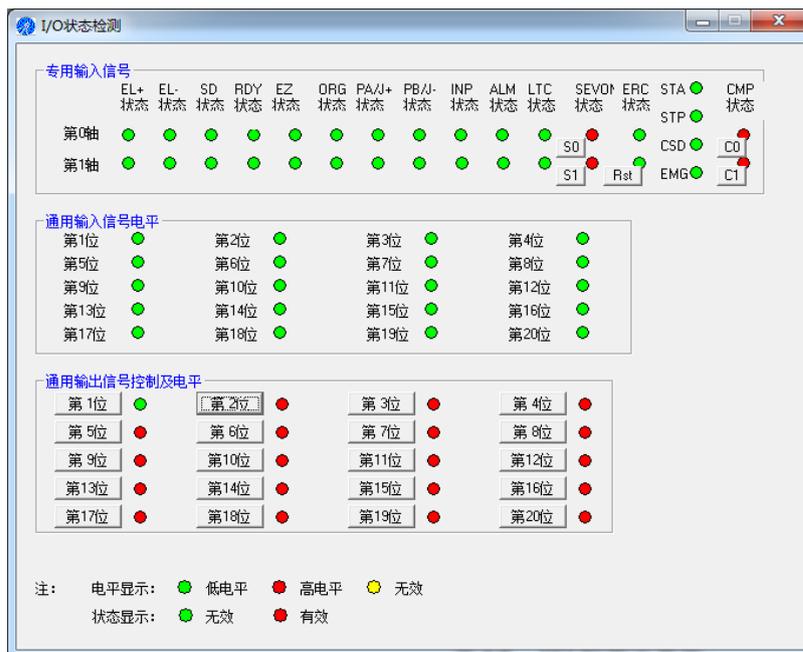


图 4-6 I/O 检测子界面

4.3 计数功能操作

计数功能测试界面如图 4-7 所示,用于 DMC1220 计数器和锁存器的各种功能的测试和验证。主要功能为:

锁存数据: 用来显示当触发信号有效时,锁存器内保存的命令计数器或编码器位置计数器内的脉冲值。

计数器状态: 用来显示第 0 轴到第 1 轴触发和清零信号的电平高低及状态。

读数方式: 分为自动计数和手动计数。自动计数大大方便了测试触发锁存功能,用户可以连续改变触发端口的电平,进行数据的锁存而不用考虑其他复位操作等。手动计数可帮助用户理解各种标识位对触发功能的影响,从而编制出自己的应用软件。

此界面的下方显示了第 0 轴到第 1 轴计数器和锁存器的数值。

界面的右边还有一些按钮用来复位和参数配置。



图 4-7 计数功能子界面

4.4 运动功能测试操作

如图 4-8 所示的编程操作界面可以用于运动控制函数调用与执行，使用十分简明方便。

运动轴选择：选择要测试的轴号，根据运动类型：单轴运动、2 轴直线插补或 2 轴圆弧插补来确定需选的轴数。

运动方式选择：运动模式有 5 种：回原点、连续运动、定长运动、直线插补、圆弧插补。

回原点运动时，指定运动轴按照设置好的回零参数进行回零动作。

连续运动和定长运动为单轴运动，指定轴可进行不会自动停止的运动或者运动一个指定的距离。

直线插补可完成 2 轴的直线插补运动。

圆弧插补是 2 轴的圆弧插补运动。

单轴运动参数的设置：设置第 0 轴到第 1 轴在进行回原点、连续运动、定长运动时的运动参数，包括：初速度（脉冲数/秒 PPS）、驱动速度(PPS)、总加速时间(秒 S)、S 段时间(S)、终点位置(脉冲数 P)、运行方向。

直线/圆弧插补参数及插补速度设置：用来设置第 0 轴到第 1 轴插补运动终点、圆心位置、插补运动的初速度、驱动速度、总加速时间和 S 段时间。

制动方式选择：确定运动轴在遇到停止指令时，是按照设置的运动曲线减速停止还是立即停止脉冲指令。

位移方式选择：确定位置坐标是相对还是绝对。

加速曲线选择：确定运动速度曲线是梯形还是 S 形。

圆弧方向选择：确定圆弧插补时的运动方向。

位置计数器选择：确定下方显示的 0~1 号轴当前位置是来自命令位置计数器或是反馈位置计数器。

右侧的按钮：

按“加载”键，将计算机当前设置的数据加载到运动卡中；

按“保存”键，当前控制卡中的数据被保存在计算机的 DMC1220 测试软件中；

按“启动”键，开始执行预置好参数的运动命令；按“停止”键，停止一切正在进行的运动过程；

按“位置清零”键，将所有位置寄存器的数值清零。



图 4-8 运动操作测试界面

第五章运动控制功能和相关函数

在本章中我们对 DMC1220 控制卡的一些主要控制功能进行简要介绍，并对实现这些功能所需调用的函数进行说明，以使用户更好更快地理解和使用这些功能及相关函数。

本节的主要内容如下：

5.1 初始化、关闭控制卡函数

5.2 指令脉冲输出

5.3 单轴位置和速度控制 用梯形和 S 形速度曲线控制单轴运动

5.4 回原点运动模式

5.5 多轴运动控制 直线插补、圆弧插补和多段连续缓冲插补运动

5.6 位置计数 命令位置、反馈位置和目标位置

5.7 通用 I/O 控制操作

5.8 伺服电机驱动器接口 伺服电机驱动器接口及相关操作函数

5.1 初始化、关闭控制卡

在操作运动控制卡之前，必须调用 `d2210_board_init` 函数为运动控制卡分配资源。同样，当程序结束对运动控制卡的操作时，必须调用 `d2210_board_colse` 释放运动控制卡所占用的 PC 系统资源，使得所占资源可被其它设备使用。

初始化函数：(以标准 C 语言为例说明，下同)

WORD `d2210_board_init(void)`

功能：为 DMC1220 运动控制卡分配系统资源并初始化控制卡

参数：无

返回值：卡数，(0~8)，其中 0 表示没有卡

关闭控制卡函数：

void `d2210_board_close(void)`

技术支持热线：0755-26417593

网址：www.szleadtech.com.cn21

功能：释放控制卡占用的系统资源。当程序结束时必须调用此函数，它与 d2210_board_init 函数是一个相反的过程

参数：无

返回值：无

例程：初始化和关闭控制卡 (以标准 C 语言为例说明，下同)

```
.....  
  
CardCount = d2210_board_init();  
  
if(CardCount == 0)  
{  
  
printf("\n 没有发现 DMC1220 运动控制卡");  
  
getch();  
  
return();  
  
}  
  
.....  
  
d2210_board_close();  
  
.....
```

注意：程序结束时，必须调用 d2210_board_close 函数释放系统资源。

5.2 脉冲输出模式的设置

DMC1220 卡采用指令脉冲控制步进/伺服电机。由于市面上的众多电机驱动器厂家信号接口要求各有不同（常用的有六种类型），所以在使用控制卡控制具体的电机驱动器时，必须对脉冲输出方式进行正确的设定，电机才能正常工作。

用户在调用运动控制函数之前应先调用 `d2210_set_pulse_outmode` 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：

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

两种基本模式如表 5-1 所示：

表 5-1 两种基本的指令脉冲输出方式

模式	PULn-脚输出	DIRn-脚输出
脉冲/方向 (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：指令脉冲信号的接口特征可能是差分的，也可能是单端（集电极开路）的。请参考硬件手册相关章节。

5.2.1 脉冲/方向模式

在此模式下，PULn-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-输出方向信号，该信号的不同电平对应电机不同的转动方向。此种模式在驱动器中最多。

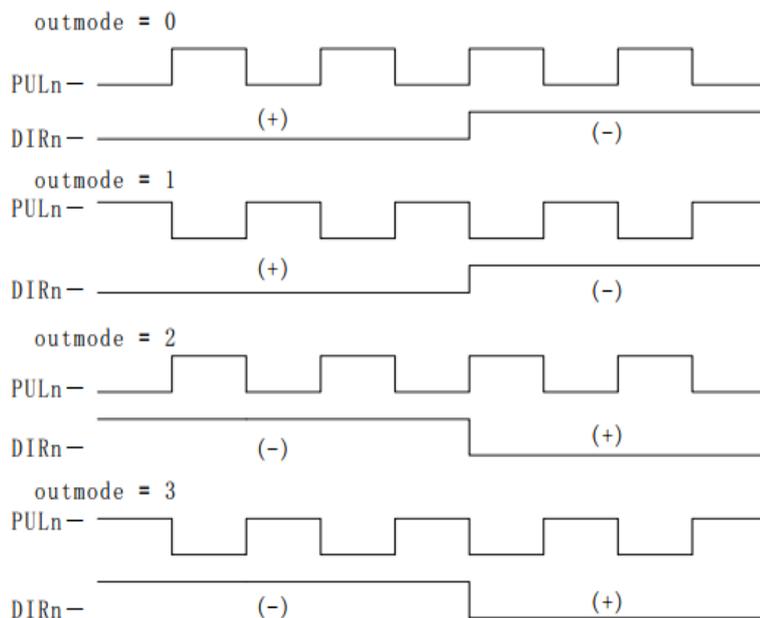


图 5-1 单脉冲输出模式

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，变化为高时电机走一步）；也可设置为下降沿有效（即脉冲信号常态为高电平，变化为低时电机走一步）。方向信号可设置为高电平对应正向或低电平对应正向两种选择。所以实际上此种模式下有四种指令类型，如图 5-1 所示：

5.2.2 双脉冲模式

在此模式下，PULn-和 DIRn-引脚分别表示正向（CW）和反向（CCW）脉冲输出。从 PULn-引脚输出的脉冲使电机正转，而 DIRn-引脚输出的脉冲使电机反转。脉冲信号有上升沿有效或下降沿有效的选择，所以此模式下共有两种指令类型，如图 5-2 所示：

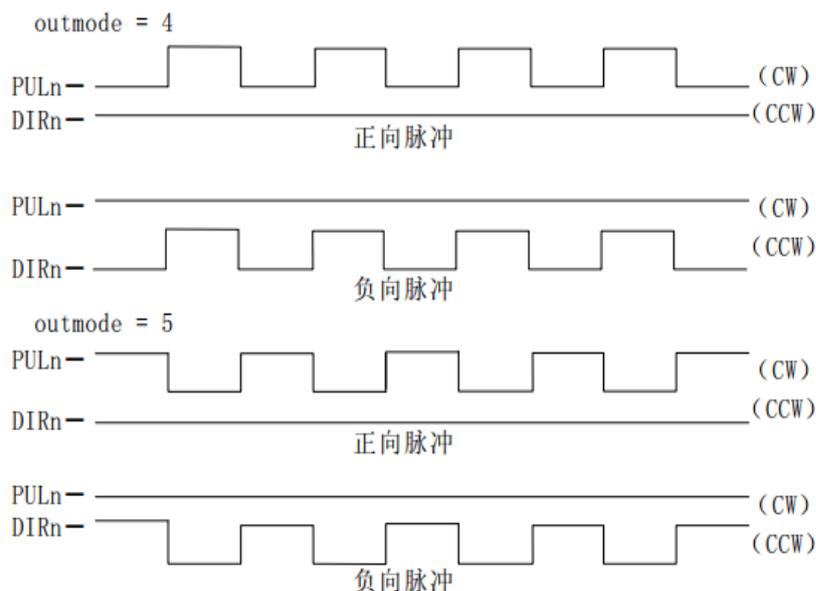


图 5-2 双脉冲输出模式

脉冲输出方式的设置函数:

void d2210_set_pulse_outmode(WORD axis, WORD outmode)

功能: 设置指定轴的脉冲输出方式

参数: axis: 轴号;

Outmode: 脉冲输出方式选择, 其值如表 5-2 所示:

TYPE	When feeding in a positive direction		When feeding in a negative direction	
	OUT output	DIR output	OUT output	DIR output
0		High		Low
1		High		Low
2		Low		High
3		Low		High
4		High	High	
5		Low	Low	
6	 OUT _____ DIR _____		 OUT _____ DIR _____	
7	 OUT _____ DIR _____		 OUT _____ DIR _____	

返回值: 无。

例程: 设置脉冲输出方式

.....

```
d2210_set_pulse_outmode (0,0); //设置第 0 轴脉冲输出模式为单脉冲模式,
PULn-信号上升沿有效, DIRn-正向为低电平。

d2210_set_pulse_outmode (1,4); //设置第 1 轴脉冲输出模式为双脉冲模式, 上升
沿有效。

.....
```

5.3 单轴位置运动和速度控制

DMC1220 在表述运动轨迹时可以用绝对坐标和相对坐标 2 种模式, 如图 5-3 所示。

2 种模式各有优点, 如: 在绝对坐标模式中用一系列坐标点定义一条曲线, 如果要修改中间某点坐标时, 不会影响后续点的坐标; 在相对坐标模式中, 用一系列坐标点定义一条曲线, 用循环命令可以重复这条曲线轨迹多次。

在 DMC1220 函数库中距离或位置的单位为脉冲; 速度单位为脉冲/秒; 时间单位为秒。

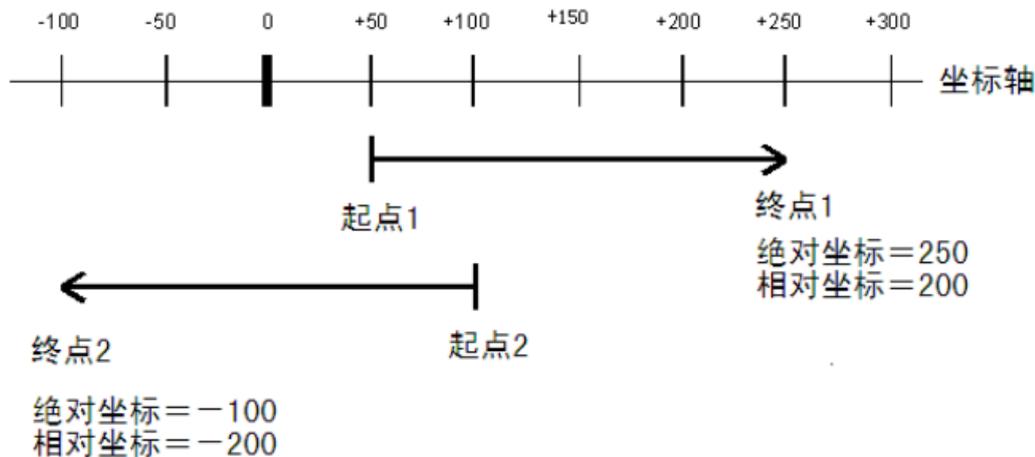


图 5-3 绝对坐标与相对坐标中轨迹终点的不同表达方式

最基本的位置控制是指从当前位置运动到另一个位置, 一般称为点位运动或定长运动。DMC1220 卡在执行单轴控制时, 可使电机按照梯形速度曲线或 S 形速度曲线进行点位运动或连续运动。

5.3.1 梯形速度曲线运动

这是位置控制中最基本的运动方式。电机在运动一段指定距离时，其运动速度按梯形曲线变化，如图 5-4 所示。

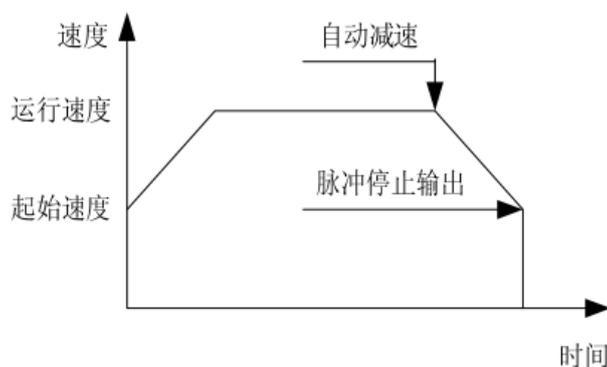


图 5-4 简单的梯形速度曲线

运动速度之所以要按梯形曲线变化，是因为：电机转轴和被拖动的物体具有惯性，不可能在瞬间内达到指定速度，因此应该给予一定的加速时间；减速时亦是类似，否则电机因为瞬间力矩不足而出现丢步、过冲（步进系统）或振荡（伺服系统）现象。

实现以梯形速度曲线运动的点位控制函数如下：

梯形速度曲线设置函数：

```
void d2210_set_profile(WORD axis, double Min_Vel, double Max_Vel, double Tacc, double Tdec)
```

功能：设定梯形速度曲线的起始速度、运行速度、加速时间、减速时间

参数：axis： 轴号

Min_Vel： 起始速度，或简称为低速

Max_Vel： 运行速度，或简称为高速

Tacc： 总加速时间(Time of Acceleration)

Tdec： 总减速时间(Time of Deceleration)

返回值： 无



提示：加速时间和减速时间可以相同，也可以不同。

以对称梯形曲线为速度的位移控制函数：

void d2210_t_pmove(WORD axis, long Dist, WORD posi_mode)

功能：让指定轴以对称梯形曲线速度作点位运动。

参数：axis：轴号

Dist：运动距离（Distance）

posi_mode：坐标模式，相对位移为 0，绝对位移为 1。

返回值：无

以非对称梯形曲线为速度的位移控制函数：

void d2210_ex_t_pmove(WORD axis, long Dist, WORD posi_mode)

功能：让指定轴以非对称梯形曲线速度作点位运动。

参数：axis：轴号

Dist：运动距离（Distance）

posi_mode：坐标模式，相对位移为 0，绝对位移为 1。

返回值：无

例程：执行以非对称梯形速度曲线作点位运动

.....

```
d2210_set_profile(0,500,6000,0.02,0.01) ;//设置 0 号轴起始速度为 500 脉冲/秒、运
```

行速度为 6000 脉冲/秒、加速时间为 0.02 秒、减速时间为 0.01 秒。

```
d2210_ex_t_pmove(0,50000,0) ; //设置 0 号轴、运动距离为 50000 个脉冲、相对
```

坐标，并开始执行运动

.....

在单轴运行过程中，运动速度 Max_Vel 和目标位置 Dist 均可以实时改变，如图 5-5 所示。若在减速时改变目标位置，电机的速度将如图 5-6 所示发生变化。实现这 2 个功能的函数如下：

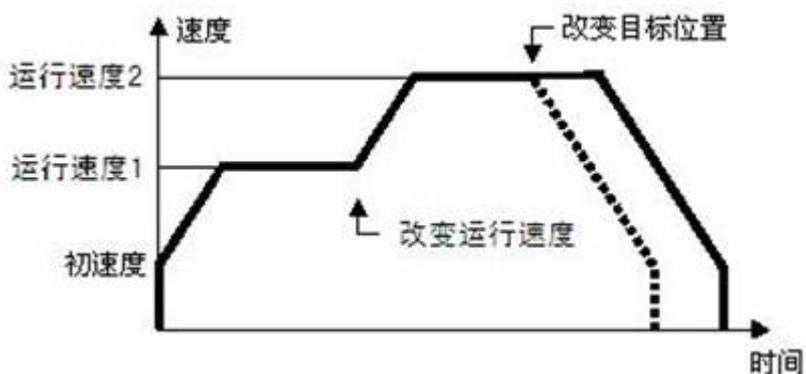


图 5-5 改变速度及改变目标位置

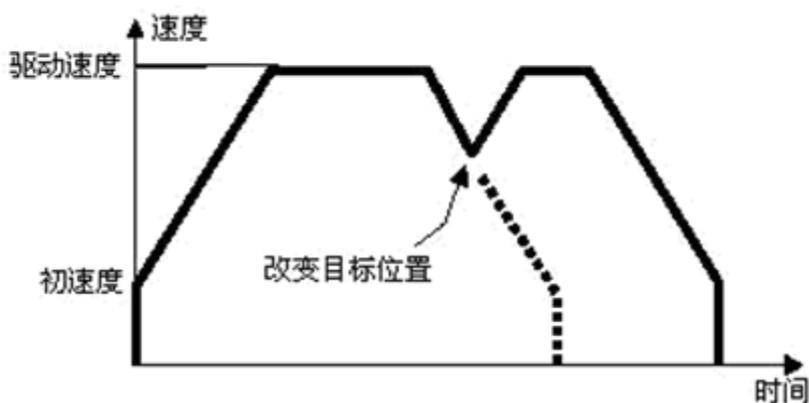


图 5-6 减速过程中改变目标位置时速度的变化情况

单轴运行中改变当前运行速度的函数：

void d2210_change_speed(WORD axis,double Curr_Vel)

功能：当指定轴在作连续运动时，调用此函数可以改变当前的运动速度，并立即按所指定的速度连续运行。

参数：Axis：轴号

Curr_Vel：新的运动速度

返回值：无

改变目标位置函数：

void d2210_reset_target_position(WORD axis,long dist)

功能：在运动中改变目标位置。若目标位置比当前位置远，则继续向前运动，到达新目标位置后，停止脉冲输出；若目标位置比当前位置近，控制卡将先停止当

前运动，然后向反方向运动至目标位置。注意：该函数仅能在相对位置模式下使用。

参数：axis：轴号

dist：新的目标位置值

返回值：无

例程：改变速度、改变终点位置

```
.....  
  
d2210_set_profile(0,500,6000,0.01,0.02); //设置梯形曲线速度、加、减速时间  
  
d2210_ex_t_pmove(0,50000,0); //设置轴号、运动距离 50000、相对坐标模式  
  
If("改变速度条件") //如果改变速度条件满足，则执行改变速度命令  
  
{  
  
Curr_Vel= 9000 ; //设置新的速度  
  
d2210_change_speed(0,Curr_Vel); //执行改变速度指令  
  
}  
  
If("改变终点位置条件") //如果改变终点位置条件满足，则执行改变终点位置命  
  
{  
  
d2210_reset_target_position(0,55000)//改变终点位置为 55000  
  
}  
  
.....
```

如果将运动中的运行速度设置得小于起始速度，整个运动过程中将会以起始速度作恒速运动，如图 5-7 所示。

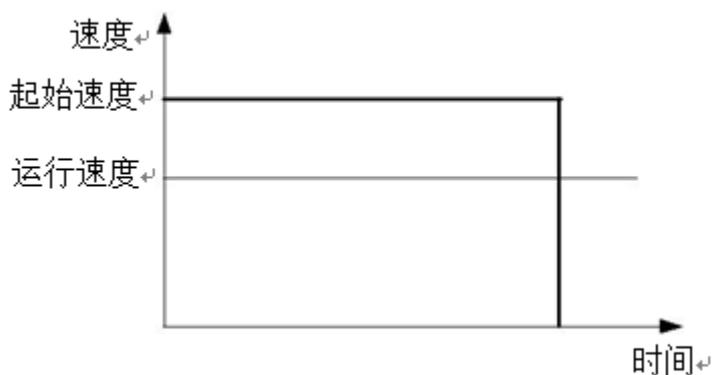


图 5-7 运行速度小于起始速度导致恒速

如果运动距离很短，当距离小于或等于 $(Max_Vel+Min_Vel) \times Tacc$ 时，理论上速度曲线将变为三角形；但 DMC1220 运动控制卡有自动调整功能，将三角形的尖峰去，以避免速度变化太大发生冲击现象，请参见图 5-8。

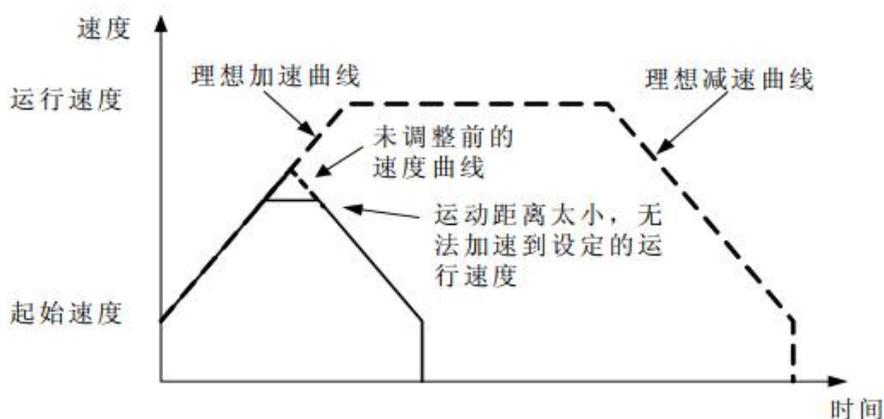


图 5-8 运动距离小于加速距离

控制轴运动的指令发出后，可以用 `d2210_check_done` 函数检测电机当前运动状态。该函数具体定义如下：

指定轴的运动状态监测函数：

WORD d2210_check_done(WORD axis);

功能：检测指定轴的运动状态，是运行还是停止。

参数：axis：轴号

返回值：0 表示指定轴正在运行

1 表示指定轴停止运行

5.3.2 S 形速度曲线运动

梯形速度曲线虽然简单，但它的速度曲线不平滑，其加速度有突变，因而运动中有冲击现象，容易引起机器噪声和传动机构的磨损。在梯形速度曲线上（参见图 5-4），运动的不平滑主要表现在四个瞬间的速度转折及相对应的加速度突变，这四个瞬间分别是：起始时、升至最高速度时、从最高速度下降时和最后停止时。

若将加速度改为线性变化，则速度曲线相应将变得光滑，如图 5-9 所示。升速和减速阶段均变得象“S”形状。采用此种速度曲线，运动更平稳，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寿命。

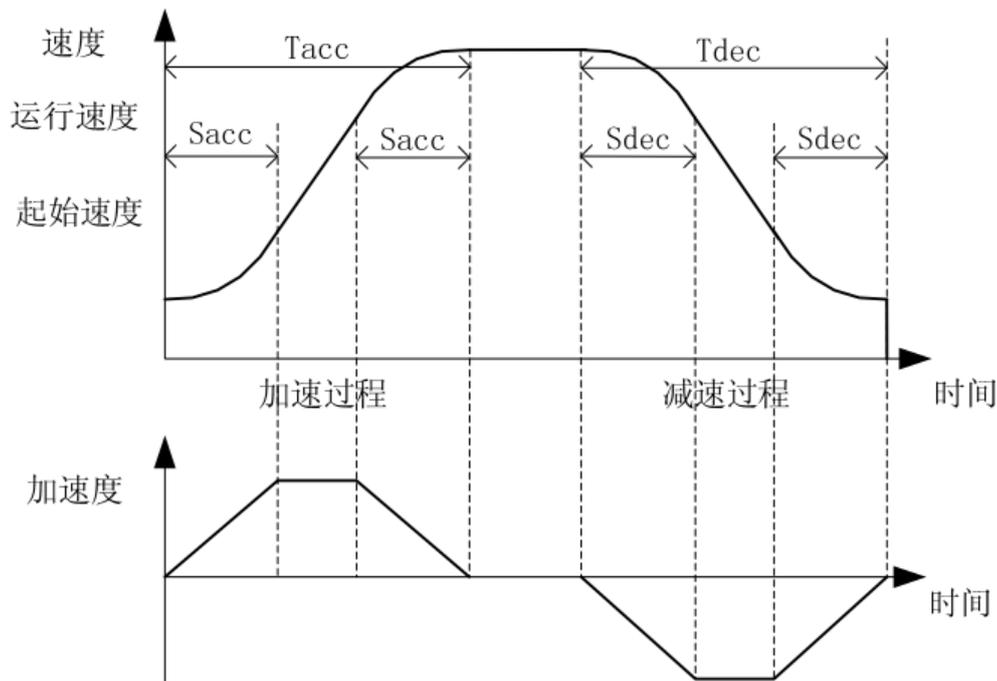


图 5-9 S 形速度曲线及其加速度曲线

设置 S 形速度曲线及其点位运动的函数如下：

设置 S 形速度曲线及其点位运动的函数如下：

非对称 S 形速度曲线设置函数：

```
void d2210_set_st_profile(WORD axis,double Min_Vel, double Max_Vel,double
Tacc, double Tdec, double Tsacc, double Tsdec)
```

功 能：设定 S 形曲线运动的起始速度、运行速度、总加减速时间、S 段加减速时间

参 数：axis 要设置的轴号

Min_Vel 起始速度，单位 pps

Max_Vel 运行速度，单位 pps

Tacc 总加速时间，单位 s

Tdec 总减速时间，单位 s

Tsacc S 加速段时间，其值应小于 Tacc 的一半。

Tsdec S 减速段时间，其值应小于 Tdec 的一半。

返回值：无

以对称 S 形曲线为速度的位移控制函数：

void d2210_s_pmove(WORD axis,long Dist,WORD posi_mode)

功能：让指定轴以对称 S 形速度曲线作点位运动。

参数：axis：轴号

Dist：运动距离 posi_mode：坐标模式，相对模式为 0，绝对模式为 1 返

回值：无

以非对称 S 形曲线为速度的位移控制函数：

void d2210_ex_s_pmove(WORD axis,long Dist,WORD posi_mode)

功能：让指定轴以非对称 S 形速度曲线作点位运动。

参数：axis：轴号

Dist：运动距离（Distance）

posi_mode：坐标模式，相对模式为 0，绝对模式为 1 返回值：无

注意：

(1).由于执行 S 形速度曲线运动时机器的振动较小，用户可以加大加速度，即提高速度曲线上线性升速区域的斜率，从而缩短加速或减速时间，从而缩短整个运动的时间。因此，S形曲线在运动速度要求十分高的设备中被广泛使用；

(2).使用 S 形曲线的目的是产生平滑运动，但是如果因为距离太短或加速太慢原因导致电机速度在加速段不能升至设定的最大值 Max_Vel 时，理论上加速段将突然切换至减速段，从而导致在速度曲线的中部出现尖三角，并因此引起该轴出现较大震动和相关问题。为了避免出现这种问题，DMC1220 内置有自动调整功能，使得加减速段的过渡保持平滑，如图 5-10 所示。

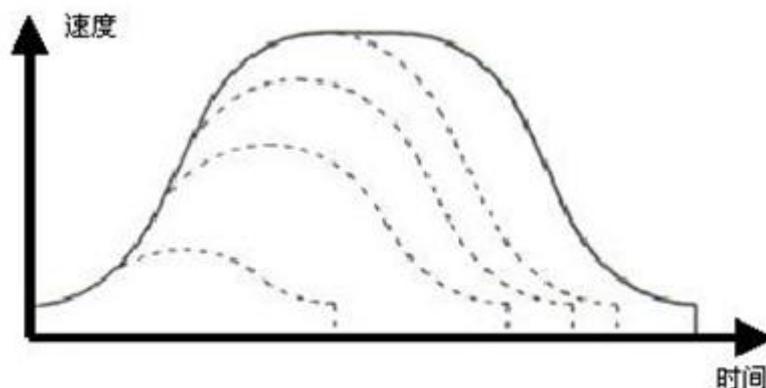


图 5-10 自动降速避免尖三角

单轴运行情况下，S 形速度曲线运动过程中也可以调用 `d2210_change_speed` 和 `d2210_reset_target_position` 函数实时改变运行速度和目标位置。注意多轴插补运行情况下不能实时改变运行速度和目标位置。

5.3.3 连续运动

DMC1220 运动控制卡可以控制电机以梯形或 S 形速度曲线在指定的加速时间内从起始速度加速至运行速度，然后以该速度连续运行，直至调用停止指令或者该轴遇到限位信号才会按启动时的速度曲线减速停止。连续运动模式时，减速停止如图 5-11 所示。

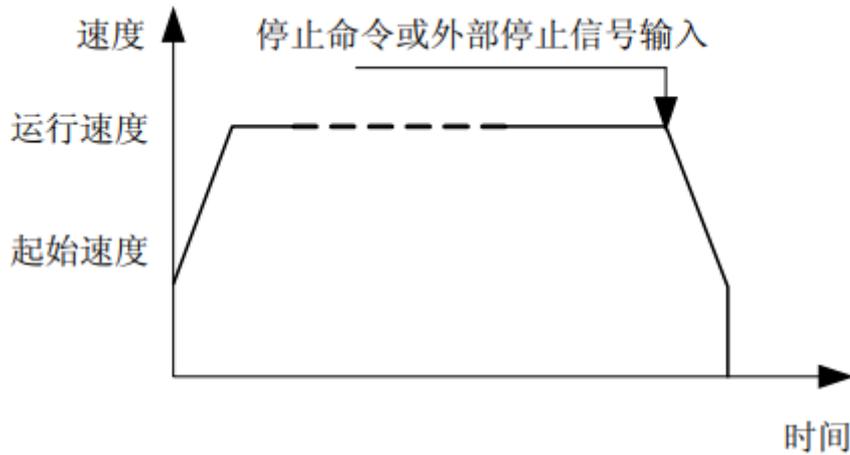


图 5-11 连续运动过程中启动及减速停止的速度曲线

梯形速度曲线单轴连续运动函数:

void d2210_t_vmove(WORD axis,WORD dir)

功能: 让指定轴以梯形速度曲线加速到指定的运行速度后, 连续运行。

参数: axis: 轴号

dir: 指定运动的方向, 其中 0 表示负方向, 1 表示正方向

返回值: 无

S 形速度曲线单轴连续运动函数:

void d2210_s_vmove(WORD axis,WORD dir)

功能: 让指定轴以 S 形速度曲线加速到指定的运行速度后, 连续运行

参数: axis: 轴号

dir: 指定运动的方向, 其中 0 表示负方向, 1 表示正方向

返回值: 无

减速停止函数:

void d2210_decel_stop(WORD axis,double Tdec)

功能: 指定轴减速停止。调用此函数后立即减速, 到达起始速度后停止

参数: axis: 轴号

Tdec: 总减速时间

返回值：无

在单轴执行连续运动过程中，可以调用 `d2210_change_speed` 实时改变速度。注意：S 形速度曲线连续运动中对运行速度的改变最好在加速已经完成的恒速段进行。图 5-12 和图 5-13 为梯形和 S 形速度曲线下连续运动中变速和减速停止过程的速度变化曲线。

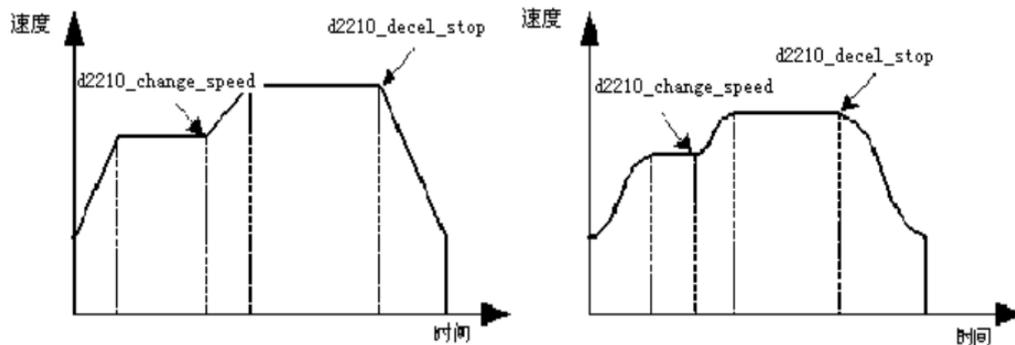


图 5-12 梯形速度控制中的变速 图 5-13 S 形速度控制中的变速

例程：以 S 形速度曲线加速的连续运动及变速、停止控制

.....

```
d2210_set_profile(0,500,1000,0.1,0.1); //设置梯形曲线速度，加、减速时间
```

```
d2210_t_vmove(0,1); //0 号轴连续运动，方向为正
```

```
if("改变速度条件")//如果改变速度条件满足，则执行改变速度命令
```

```
{
```

```
    Curr_Vel= 1200 ; //设置新的速度
```

```
    d2210_change_speed(0,Curr_Vel) ; //执行改变速度指令
```

```
}
```

```
if("停止条件")//如果运动停止条件满足，则执行减速停止命令
```

```
d2210_decel_stop(0,0.1); //减速停止，减速时间为 0.1 秒
```

.....

相关函数：

d2210_variety_speed_range

d2210_imd_stop

d2210_emg_stop

d2210_read_current_speed

请参阅本手册相关章节。

5.3.4 加减速过程的距离（脉冲数）计算

对于梯形速度曲线运动，加减速段的运动距离（脉冲数）可以按以下公式计算：

$$D_{acc} = (1/2) \times (Max_Vel + Min_Vel) \times T_{acc}$$

$$D_{dec} = (1/2) \times (Max_Vel + Min_Vel) \times T_{dec}$$

其中：D_{acc}，D_{dec} 分别为加速段距离和减速段距离；

Min_Vel，Max_Vel 为起始速度和运行速度；

T_{acc}，T_{dec} 为加速时间和减速时间。

以上公式也完全适合于 S 曲线的情况。

5.4 回原点运动

在进行精确的运动控制之前，需要设定运动坐标系的原点。DMC1220 提供多个与原点设置相关的函数：原点信号的有效逻辑电平由 d2210_set_HOME_pin_logic 函数设置，回原点模式由 d2210_config_home_mode 函数指定，回原点的动作由 d2210_home_move 函数控制，回原点的速度则由 d2210_set_profile 函数设置，到达原点后指令脉冲计数器由 d2210_set_position 函数清零。



提示：在运行回原点运动之前必须确保原点信号的正确接入。请参阅硬件手册相关章节，检查原点传感器的连接。

设置原点信号函数:

void d2210_set_HOME_pin_logic(WORD axis,WORD org_logic,WORD filter)

功能: 设置原点信号的电平和芯片内部滤波器使能。滤波器为硬件低通滤波器, 当输入信号脉宽小于 4us 时, 信号将被忽略。

参数: axis: 轴号

org_logic: 原点信号的有效电平, 0—低电平有效, 1—高电平有效

filter: 备用

返回值: 无

设定回原点模式函数:

void d2210_config_home_mode(WORD axis,WORD mode,WORD EZ_count)

功能: DMC1220 运动控制卡提供了多种不同的回原点模式, 实现精确定位到原点的方案, 通过调用此函数便可以选择其中一种模式。

参数: axis: 轴号;

mode: 回原点的模式, 0—只计 home, 1—计 home 和 EZ

EZ_count: 在回原点运动过程中, 当找到原点信号后, 该轴的 EZ 信号出现的了 EZ_count 次后, 电机停止。该参数的取值范围是: 1~16。

返回值: 无

回原点函数:

void d2210_home_move(WORD axis,WORD home_mode,WORD vel_mode)

功能: 按指定的方向和速度方式开始回原点。

参数: axis: 轴号

home_mode: 回原点的方法,

1—正方向回原点

2—负方向回原点

vel_mode: 选择回原点的速度方式,

0—低速回原点

1—高速回原点，遇原点信号，减速后停止

返回值：无



提示：执行完 `d2210_home_move` 函数后，指令脉冲计数器不会自动清零；如需清零可以在回零运动完成后，调用 `d2210_set_position` 函数软件清零。

回零模式包括：

- 0：一次回零
- 1：一次回零+同向 EZ（EZ 个数 1~16 个）
- 2：二次回零
- 3：一次回零+反找原点
- 10：一次限位回零
- 11：一次限位回零+反找限位
- 12：二次限位回零

注：回零遇限位默认不会自动反找，若需自动反找需使能反找功能，调用 `d2210_set_home_el_return` 设置。

下面介绍二种常用的回零方法的实现。

方式 0：只计 home 方式

该方式以低速回原点，适合与行程短、安全性要求高的场合。动作过程为：电机从初始位置以恒定低速度向原点方向运动，当到达原点开关位置，原点信号被触发，电机立即停止（过程 0）；将停止位置设为原点位置，如图 5-14 所示。



图 5-14 回原点方式 0

例程：低速回原点

```

    .....

    d2210_set_HOME_pin_logic(0,0,1); //设置 0 号轴的原点信号低电平有效，使能滤波

        功能

    d2210_config_home_mode(0,0,1); //设置 0 号轴模式为遇原点 after 停止，EZ 信号出现

        次数为 1

    d2210_set_profile(0,500,1000,0.1,0.1); //设置 0 号轴梯形曲线速度，加、减速时间

    d2210_home_move(0,2,0); //设置 0 号轴为负方向回原点，速度方式为低速回原点

    while (d2210_check_done(0) == 0) //等待回原点动作完成

    {

    }

    d5400_set_position(0,0); //设置 0 号轴的指令脉冲计数器绝对位置为 0

    .....
    
```

方式 1: 计 home 和 EZ 方式

若将上面的程序 `d2210_config_home_mode(0,0,1)` 改为 `d2210_config_home_mode(0,1,1)`，EZ 信号出现次数为 1 次，则回原点过程如图 5-15 所示。

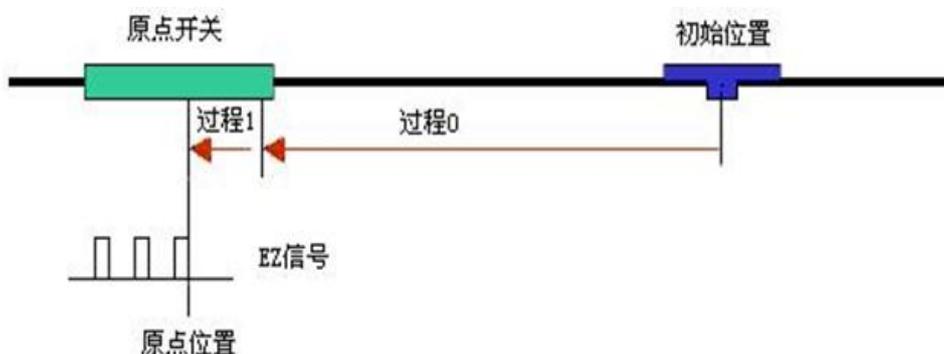


图 5-15 回原点方式 1

方式 2: 两次原点回零

如图 5.16 所示，该方式为方式 1 和方式 2 的组合。先进行方式 2 的回零加反找，完成后再进行方式 1 的一次回零。可参见方式 1 和方式 2 的说明。

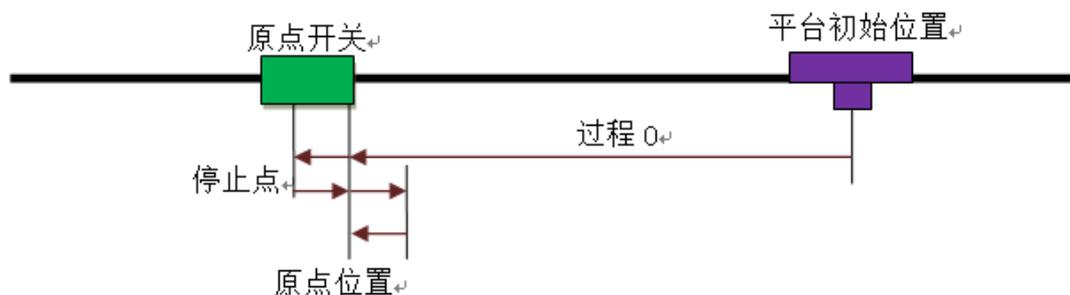


图 5-16 二次原点回零

方式 3：一次原点回零加回找

该方式先进行方式 1 运动，完成后再反向回找原点开关的边缘位置，当原点信号第一次无效的时候，电机立即停止；将停止位置设为原点位置如图 5.17 所示。



图 5-17 一次原点加回找

方式 10：一次限位回零

该方式以设定速度回原点；适合于行程短、安全性要求高的场合。动作过程为：电机从初始位置以恒定速度向限位方向运动，当到达限位开关位置，限位信号被触发，电机立即停止（过程 0）；将停止位置设为原点位置，如图 5.18 所示。



图 5-18 一次限位回零

方式 11：一次限位回零加回找

该方式先进行方式 10 运动，完成后再反向回找限位开关的边缘位置，当限

位信号第一次无效的时候，电机立即停止；将停止位置设为原点位置如图 5.19 所示。



图 5-19 一次限位加回找

方式 12: 两次限位回零

如图 5.20 所示，该方式为方式 10 和方式 11 的组合。先进行方式 11 的回零加反找，完成后再进行方式 10 的一次回零。可参见方式 10 和方式 11 的说明。

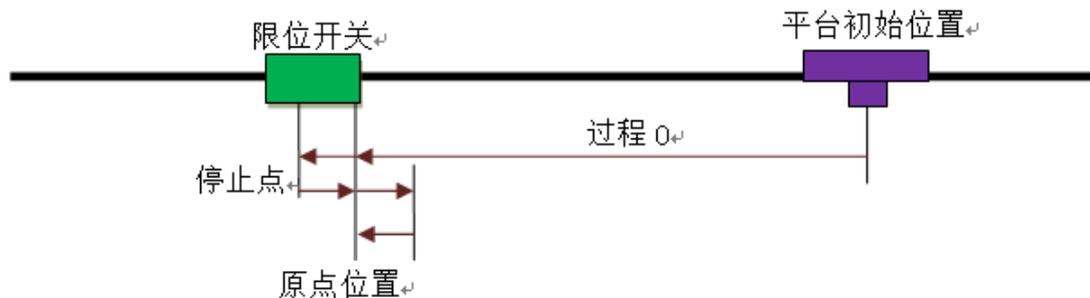


图 5-20 两次限位回零

相关函数：（请参阅相关章节）

d2210_set_position

d2210_variety_speed_range

d2210_imd_stop

d2210_emg_stop

d2210_read_current_speed

5.5 多轴运动控制

5.5.1 多轴联动

几个轴同时运动，一般称为多轴联动。

DMC1220 运动控制卡可以控制多个电机同时执行 `d2210_t_move`、

d2210_s_move 这类单轴运动函数。所谓同时执行，是在程序中顺序调用 d2210_t_move、d2210_s_move 等函数，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 5-19 所示。

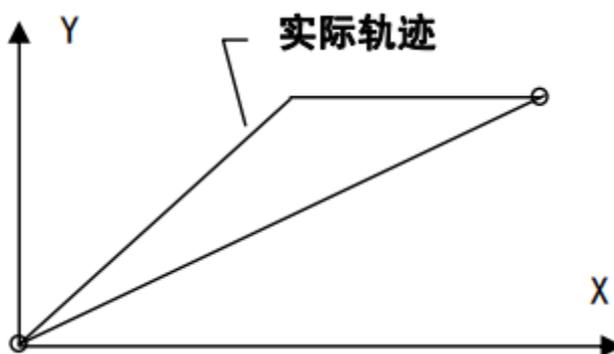


图 5-19 二轴联动示意图

5.5.2 直线插补运动

插补运动与多轴联动则不同：插补运动不但能保证起点、终点位置准确外，X 轴和 Y 轴的脉冲是按照直线斜率成比例发出的，所以在插补运动过程中的每一个时刻，其运动轨迹与理论曲线的误差总是小于一个脉冲当量，如图 5-20 所示。

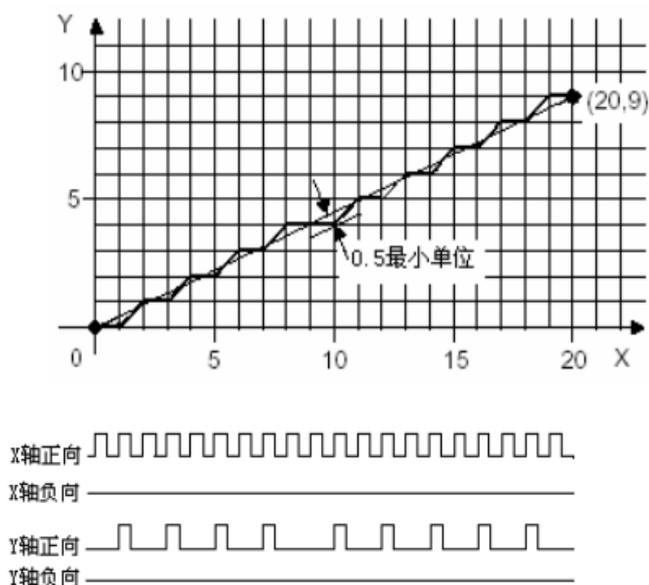


图 5-20 直线插补示意图

DMC1220 卡可以进行任意 2 轴直线插补，用户只需将插补运动的速度、加速度、终点位置等参数写入相关函数，而无需介入插补过程中的计算工作。

二轴直线插补：

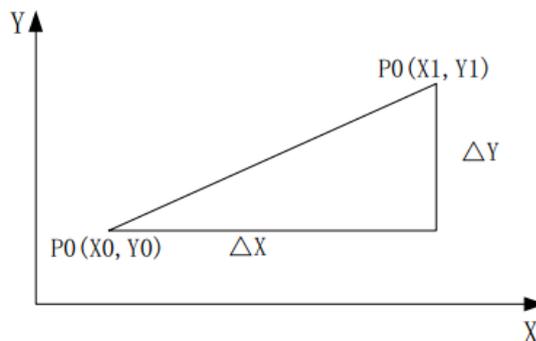


图 5-21 两轴直线插补

如图 5-21 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 $\Delta X : \Delta Y$ ，二轴合成的矢量速度为：

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

调用 2 轴直线插补函数时，调用者需提供矢量速度，包括其起始矢量速度 Min_Vel 和工作矢量速度 Max_Vel，梯形和 S 形速度曲线参数。

二轴直线插补函数：

void d2210_t_line2(WORD axis1,long Dist1,WORD axis2,long Dist2,WORD posi_mode)

功能：让指定的两轴作对称的梯形加减速插补运动。当 posi_mode 为 0 时，作相对位移运动，运动方向由 Dist 的正负值确定；为 1，作绝对位移运动，运动方向由 Dist 与当前位置的差值决定。

参数：axis1：第一、二轴轴号

Dist1, Dist2：第一、二轴距离

posi_mode：位置模式，0—相对位移, 1—绝对位移

返回值：无

例程：XY 轴直线插补

```
short AxisArray[2];  
  
AxisArray[0]=0; //定义插补 0 轴为 X 轴  
  
AxisArray[1]=1; //定义插补 1 轴为 Y 轴  
  
d2210_set_vector_profile(1000,5000,0.1,0.2);  
  
d2210_t_line2(AxisArray[0],30000,AxisArray[1],40000,0);
```

该例程使 x, y 轴进行相对模式直线插补运动，其相关参数为：

X=30000 pulse

Y=40000 pulse

起始矢量速度=1000pps (0 轴,1 轴分速度为 600, 800pps)工作矢量速度=5000pps (0 轴,1 轴分速度为 3000, 4000pps) 梯形加速时间=0.1s

梯形减速时间=0.2s

相关函数：

d2210_set_vector_profile

5.5.3 圆弧插补

DMC1220 卡的两轴之间可以进行圆弧插补，圆弧插补分为相对位置圆弧插补和绝对位置圆弧插补，运动的方向分为顺时针（CW）和逆时针（CCW），如图 5-23 所示。

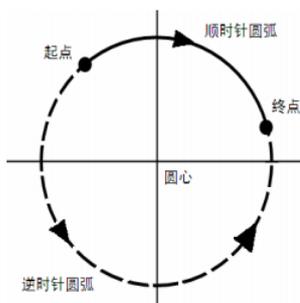


图 5—23

二轴绝对位置插补函数:

```
void d2210_arc_move(WORD *axis, long *target_pos,long *cen_pos, WORD  
arc_dir)
```

功能: 让指定的二轴作绝对位置圆弧插补运动。

参数: axis: 轴号列表

target_pos: 目标位置列表 (指定圆弧终点) cen_pos: 圆心位置列表

arc_dir:

圆弧方向, 0—顺时针 1—逆时针

返回值: 无

二轴相对位置插补函数:

```
void d2210_rel_arc_move(WORD *axis,long *rel_pos,long *rel_cen, WORD arc_dir)
```

功能: 让指定的二轴作相对位置圆弧插补运动。

参数: axis: 轴号列表

rel_pos: 目标位置列表 (指定圆弧终点)

rel_cen: 圆心位置列表

arc_dir: 圆弧方向, 0—顺时针, 1—逆时针

返回值: 无

例程: XY 轴圆弧插补

```
word AxisArray[2];  
  
AxisArray[0]=0;      //定义 0 轴为插补 X 轴  
  
AxisArray[1]=1;      //定义 1 轴为插补 Y 轴  
  
long Pos[2] = {5000, 0};  
  
long Cen[2] = {5000,-5000};  
  
d2210_set_vector_profile(1000,3000,0.1,0.2);  
  
d2210_arc_move (AxisArray, Pos, Cen, 0);      // XY 轴进行顺时针方向绝对圆弧插
```

补运动，终点 (5000, 0)，圆心 (5000, -5000)

5.5.4 连续缓冲插补运动

DMC1220 卡允许两轴电机进行连续缓冲插补运动，但是线段之间有一个小的加/减速过程，效果可以用图 5-24 来表示

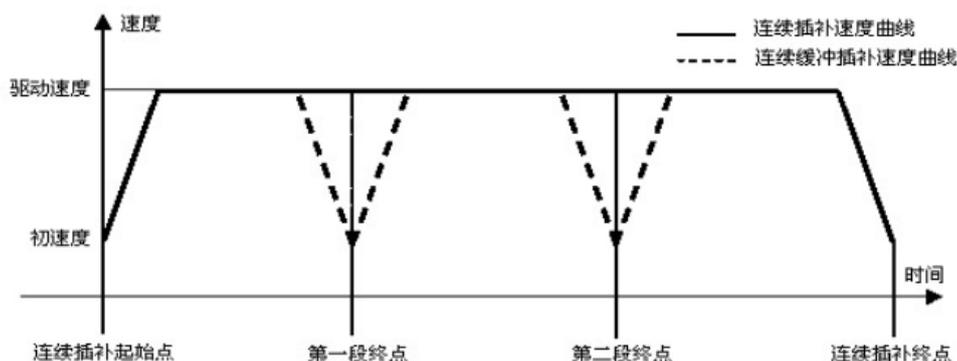


图 5-24 插补段连续运动

本控制卡上具有两级硬件缓冲功能，能预先存储两步运动的数据。控制卡当前运动的数据存储在工作寄存器 R (register) 中，下一个运动的数据存储在预置缓冲区 PR (pre-register) 中，如图 5-25 所示。当前运动完成，PR 中的数据自动移至 R 中开始执行，此时寄存器 PR1 变为空。PC 机查询到 PR 为空后，即可补充下一运动的数据。如此不断循环，直至完成所有运动。

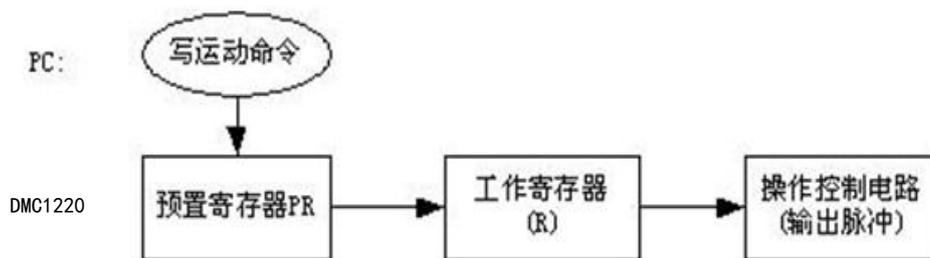


图 5-25 双缓冲寄存器连续缓冲插补的工作原理

读取指定轴预置缓冲区状态的函数：

WORD d2210_prebuff_status(WORD axis)

功能：读取指定轴预置缓冲区状态

参数：axis：轴号

返回值：预置缓冲区状态，0—预置缓冲区空，1—预置缓冲区满

连续缓冲插补注意事项:

1. 预置缓冲区不为空时, 用户不能写入新的运动命令, 否则将导致错误。
2. 如果在插补运动过程中, 出现触发限位而停止, 那么后续写入的资料和命令都是无效的。

下面通过一个例子说明连续缓冲插补的应用。

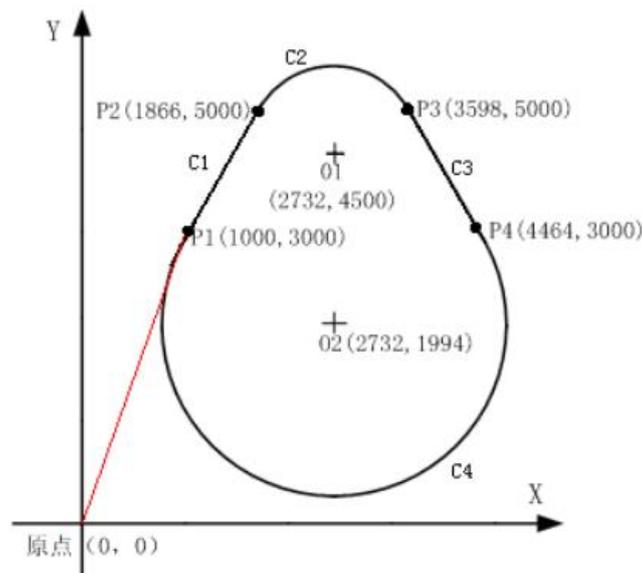


图 5-26 两轴连续缓冲插补示例

描述: 假定在一工作平面上, 刀具在系统 XY 轴物理原点的正向位置(X 轴对应 Axis[0], Y 轴对应 Axis[1]), 现刀具先回到系统的物理原点, 再执行插补运动 C1, 然后执行圆弧插补运动 C2, 接着执行插补运动 C3, 最后执行插补运动 C4 回到 P1 点, 如图 5-26 所示。

例程: XY 轴连续缓冲插补

```
WORD Axis[2]; //定义运动轴
```

```
Axis[0] = 0; Axis[1] = 1;
```

```
long Pos1[2] = {3598, 5000}; //定义 C2 圆弧插补的终点位置 P3
```

```
long Pos2[2] = {1000, 3000}; //定义 C4 圆弧插补的终点位置 P1
```

```
long Cen1[2] = {2732, 4500}; //定义 C2 圆弧插补的圆心位置 O1
```

```
long Cen2[2] = {2732 , 1994}; //定义 C2 圆弧插补的圆心位置 O2

d2210_set_vector_profile (50, 500, 0.01, 0.01); //设置插补速度、加减速时间

d2210_set_HOME_pin_logic(0, 0, 1); //设置轴 0 原点信号低电平有效，使能滤波功能

d2210_config_home_mode(0, 0, 1); //设置轴 0 模式为遇原点后停止，EZ 信号次数 1

d2210_set_HOME_pin_logic(1, 0, 1); //设置轴 1 原点信号低电平有效，使能滤波功能

d2210_config_home_mode(1, 0, 1); //设置轴 1 模式为遇原点后停止，EZ 信号次数 1

d2210_home_move(0, 2, 0); //设置轴 0 为负方向回原点，速度方式为低速回原点

while (d2210_check_done (0) == 0) //判停

{

d2210_home_move(1, 2, 0); //设置轴 1 为负方向回原点，速度方式为低速

while (d2210_check_done (0) == 0) //判停

{

d2210_t_line2(Axis[0], 1000, Axis[1], 3000, 1); //执行行程 C1,到达 P2 点

while (d2210_prebuff_status == 1) //等待缓冲寄存器为空

{

d2210_rel_arc_move(Axis, Pos1, Cen1, 0); //执行行程 C2,到达 P3 点

while (d2210_prebuff_status == 1) //等待缓冲寄存器为空

{

d2210_t_line2(Axis[0], 4464, Axis[1], 3000, 1); //执行行程 C3,到达 P4 点

while (d2210_prebuff_status == 1) //等待缓冲寄存器为空

{
```

```
d2210_rel_arc_move(Axis, Pos2, Cen2, 0); //执行行程 C3,到达 P4 点
```

```
while (d2210_check_done (0) == 0) //判停
```

```
{
```



注意：缓冲寄存器不为空时，不得写入新的运动命令，否则将导致错误。

5.6 位置计数

DMC1220 运动控制卡上每个轴均设置了命令位置计数器和反馈位置计数器，命令位置计数器用于监测指令位置，反馈位置计数器用于监测机械位置，下面详述其用法。

5.6.1 命令位置计数器

命令位置计数器是一个 32 位正负计数器，对控制卡输出脉冲进行计数。当输出一个正向脉冲后，计数器加 1；当输出一个负向脉冲后，计数器减 1。

读取指令脉冲位置函数：

```
long d2210_get_position( WORD axis )
```

功能：读取指定轴的指令脉冲计数器。

参数：axis：轴号

返回值：指令位置脉冲数

设置指令脉冲位置函数：

```
void d2210_set_position(WORD axis, long current_position)
```

功能：设置指定轴的指令脉冲计数器。

参数：axis：轴号

current_position：指令脉冲计数器的设定值

返回值：无

例程：位置操作

```
d2210_set_postion(0,100); //设置轴 0 的脉冲位置为 100  
position = d2210_get_position(0); //读轴 0 的当前位置值至变量 position
```

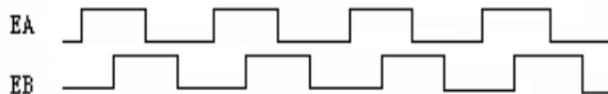
5.6.2 反馈位置计数器

反馈位置计数器是一个 32 位正负计数器，对通过控制卡编码器接口 EA，EB 输入的脉冲（如编码器、光栅尺反馈脉冲等）进行计数。下面对这个计数器的计数原理进行详细描述。

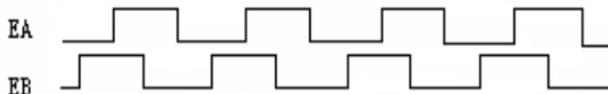
DMC1220 运动控制卡能接收 AB 相输入模式。

A/B 相输入模式：

正向计数



负向计数



这种模式下，EA 脉冲信号“超前”或“滞后”EB 脉冲信号 90 度，而这种“超前”或“滞后”就表示电机的运转方向。当 EA 信号“超前”EB 信号时，反馈位置加计数；EA 信号“滞后”EB 信号时，反馈位置减计数。而且，用户可选用 4、2、1 倍计数模式对 EA，EB 信号进行计数用 d2210_counter_config 进行设置，分别介绍如下：

4 倍计数：若为正向计数，反馈位置计数器的值为 EA 反馈脉冲数的 4 倍；若为负向计数，反馈位置计数器的值为 EB 反馈脉冲数的 4 倍。

2 倍计数：若为正向计数，反馈位置计数器的值为 EA 反馈脉冲数的 2 倍；若为负向计数，反馈位置计数器的值为 EB 反馈脉冲数的 2 倍。

1 倍计数：若为正向计数，反馈位置计数器的值为 EA 反馈脉冲数；若为负

向计数，反馈位置计数器值为 EB 反馈脉冲数。

例如：如果使用的编码器为 2500 线，即电机转一周反馈的 EA、EB 脉冲数都为 2500 个，让电机转一周，若设置反馈位置计数模式为 4 倍计数，反馈位置计数器的值为 10000；若设置为 2 倍计数，反馈位置计数器的值为 5000；若设置为 1 倍计数，反馈位置计数器的值为 2500。

设置编码器反馈输入模式函数：

void d2210_counter_config(WORD axis, WORD mode)

功能：设置编码器输入口的计数方式。

参数： axis：轴号

mode：编码器反馈输入模式

0 保留

1 1 倍 A/B 相脉冲信号

2 2 倍 A/B 相脉冲信号

3 4 倍 A/B 相脉冲信号

返回值：无

读取编码器位置函数：

long d2210_get_encoder(WORD axis)

功能：读取编码器反馈的脉冲计数值。范围： 28 位有符号数。

参数： axis：轴号

返回值：编码器的计数值

设置编码器位置函数：

void d2210_set_encoder (WORD axis, long encoder_value)

功能：设置编码器的脉冲计数值。范围： 28 位有符号数。

参数： axis：轴号

encoder_value：编码器脉冲计数的设定值。

返回值：无

例程： 编码器反馈计数的操作

```
d2210_counter_config(0,3); //设置轴 0 为 4 倍计数，默认的 EA、 EB 计数方向  
d2210_set_encoder(0,0); //设置轴 0 的计数初始值为 0  
X_Position = d2210_get_encoder(0); //读轴 0 的计数器的数值至变量 X_Position
```

5.6.3 位置锁存

DMC1220 卡提供编码器计数值锁存功能，该功能广泛应用于各种测量行业。位置锁存方式可以选择对每个编码器信号独立锁存，也可以通过任一个锁存端口对全部编码器计数值同时锁存，触发指令 LTC 接口一般接测量探头的触发信号。该功能用于位置测量十分准确、方便。在复位触发标志位后，当锁存信号被触发，当前编码器计数值立即被捕获至位置锁存器中，并将触发标志位置位，保护当前锁存器内的数值，直到触发标志位被再次复位。使用 `d2210_get_latch_value` 可读取锁存内的编码器计数值。

设置位置锁存方式函数:

void d2210_config_latch_mode(WORD cardno,WORD all_enable)

功能： 设置 LTC 端口的锁存方式

参数： cardno: 卡号 (0~N - 1, N 为卡数)

all_enable: 0: 单独锁存, 1: 两轴同时锁存

返回值：无

读取位置锁存值函数:

long d2210_get_latch_value(WORD axis)

功能： 读取被触发锁存到锁存器内的编码器计数值

参数： axis: 轴号

返回值： 返回编码器的脉冲数

例程： 位置锁存的操作

```
int key=0; //锁存状态标识

long LatchStatus; //触发状态标识

long xValue[100],yValue[100],zValue[100]; //存放锁存值的数组

int g_count=0; //计数器

while(g_count<100) //假设测 100 点

{

    LatchStatus = DMC1220_Read_Latch_Status(0); //得到锁存器状态

    If ( ( (LatchStatus & 0xF00) != 0) && key==0)

    {

        xValue[g_count] = DMC1220_Get_LatchValue(0); //得到 X 锁存值

        yValue[g_count] = DMC1220_Get_LatchValue(1);

        yValue[g_count] = DMC1220_Get_LatchValue(2);

        g_count++; //计数器加 1

        key = 1; //标识置位

    }

    else If ( ( (LatchStatus & 0xF) == 0) && key==1)

    {

        DMC1220_Reset_Latch_Flag (); //锁存器复位

        key = 0 ; //标识复位

    }

}
```

}

5.7 通用 I/O 控制

DMC1220 卡具有 20 个通用数字输入口，20 个通用数字输出口，通用输入采用了光耦隔离，输出口为集电极开路方式。用户可以使用这些 I/O 口用于输入开关信号、传感器信号等信号，或是输出继电器、电磁阀等输出设备的控制信号。

读输入口函数：

int d2210_read_inbit(WORD cardno,WORD bitno)

功能：读取指定卡的指定输入口的电平

参数： cardno：控制卡卡号，范围（0~N - 1， N 为卡数）

bitno：输入口位号（1~20）

返回值： 0 表示低电平； 1 表示高电平

写输出口函数：

void d2210_write_outbit (WORD cardno,WORD bitno,WORD on_off)

功能：设置指定卡的指定输出口的电平

参数： cardno：控制卡卡号，范围（0~N - 1， N 为卡数）

bitno：输出口位号（1~20）

on_off：输出电平， 0 表示输出低电平， 1 表示输出高电平

当拨码开关 S1 对应的位设置为 OFF 时，当对输出口 1~12 置位时， 0 表示高电平； 1 表示低电平。

返回值：无

例程：通用 I/O 的操作，如图 5-29

```
If(d2210_read_inbit(input1)==0) // 读取 INPUT1 口的状态，判断按键是否按下？
```

```

{

d2210_write_outbit(out1, 0); //如果按键按下， OUT1 口输出为 0， LED 发光

}

else

{

d2210_write_outbit(out1, 1); //如果按键未按， OUT1 口输出为 1， LED 不亮

}

```

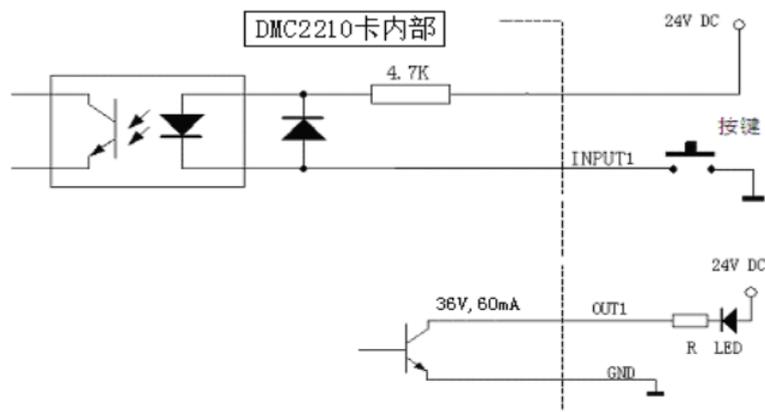


图 5-29 通用输入输出口的接口图

相关函数：

int d2210_read_outbit(WORD cardno,WORD bitno)

long d2210_read_inport(WORD cardno)

long d2210_read_outport(WORD cardno)

5.8 伺服驱动器接口

DMC1220 卡为伺服电机驱动器的 5 个信号： INP， ALM， ERC， SVON， RDY， 提供了专用接口。其中 INP 和 ALM 是伺服驱动器状态的输入； ERC 是控制卡输出到伺服驱动器的信号，用来清除伺服驱动器在运行过程命令位置与实际位置的误差； SVON 伺服电机使能信号是控制卡输出给伺服驱动器的控制信

号，用来控制是否打开伺服电机； RDY 是伺服驱动器发给控制卡的状态信号，表示伺服驱动器已经准备好接收控制指令。

5.8.1 INP 定位完成信号

INP 即伺服电机定位完成信号，是伺服电机驱动器发出给控制卡的状态信号，当伺服电机定位完成时，伺服驱动器自动将该信号置为有效。通常采用脉冲信号控制的伺服驱动器都有一个位置误差计数器，用于监测命令位置计数器和反馈位置计数器之间的误差，如图 5-30 所示。驱动器将控制电机运动，使得这个位置误差保持在较小范围。控制卡发出一个运动命令之后，到伺服电机执行完该命令总会有一定的延时；同样，当控制卡停止输出脉冲时，伺服电机也不会马上停下来，而是继续运动直到误差计数值减小到允许范围内，这时伺服驱动器将发出一个伺服电机到位（INP）信号，反馈给控制卡表明伺服电机已经停止运动。误差计数值可以在伺服驱动器上设置，一般设为几个脉冲，具体数值视该应用中要求的精度而定。

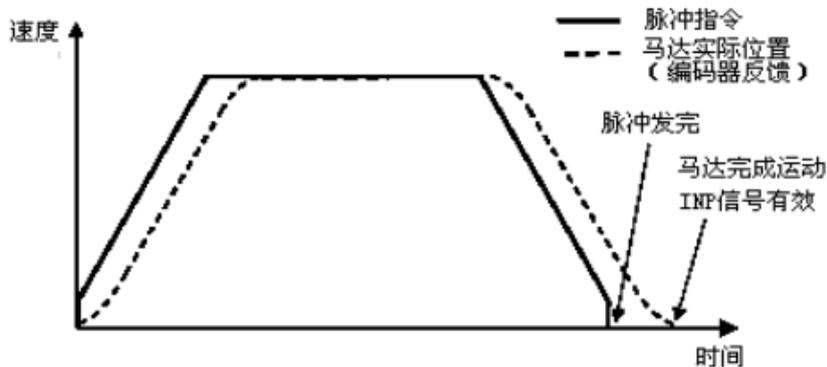


图 5-30 伺服定位完成时的 INP 信号

函数库中的 `d2210_config_INP_PIN` 函数可设置 INP 信号的有效电平值以及使能或者禁止对 INP 信号的响应。如果设成对 INP 信号不响应（即 `disable`），控制卡将在定长运动脉冲发完后，不管伺服定位是否完成，都立刻翻转卡上的运动完成标识，即 `d2210_check_done` 将返回“完成”状态。相反，如果使能 INP 信号（即 `enable`），那么控制卡在发完脉冲之后，还要等到 INP 信号变为有效之后，才翻转运动完成标识，`d2210_check_done` 函数返回“完成”状态。INP 信号电平可以通过 `d2210_get_rsts` 函数来读取。

相关函数:

d2210_config_INP_PIN

d2210_get_rsts

d2210_check_done

5.8.2 ALM 伺服报警信号

ALM 信号是从伺服电机驱动器发出给控制卡的状态信号，用来报告伺服驱动器或电机出错。DMC1220 控制卡接收到 ALM 信号时，将立即停止发出脉冲，该过程是一个硬件处理过程。对 ALM 信号的是否使能以及有效电平的设置可以通过调用 d2210_config_ALM_PIN 函数来完成，ALM 信号状态可以通过 d2210_axis_io_status 函数来读取。

相关函数:

d2210_config_ALM_PIN

d2210_axis_io_status

5.8.3 ERC 误差清除信号

伺服电机驱动器依赖电机目标位置(即要求电机达到的位置)和当前位置(即电机已经到达的位置)之间的误差来控制电机运动，若这个误差为零电机将停止运动。ERC 信号是控制卡输出给伺服驱动器的控制信号，当伺服驱动器接收到该信号时会立即清除误差并停止电机运转。在以下几种情况下可能需要用到 ERC 信号:

- (1) 回原点运动时触发原点信号时
- (2) 运动过程中，触发限位信号时
- (3) 伺服驱动器发出 ALM 信号时
- (4) 软件发出立即停止信号时

这四种情况出现时，控制卡将停止输出脉冲，但是由于伺服电机本身的特性，决定了在命令脉冲停止后到电机本身停止之间存在一个延迟时间，电机会继续运

行直到 INP 信号有效（即误差计数达到允许范围内），这时如果给伺服驱动器发出一个 ERC 信号就可将误差清零从而使电机停止。当指定轴在输出 ERC 信号后，开始启动定时器，定时器按设定使 ERC 保持有效，这个期间新的指令脉冲被忽略，直到定时结束（ERC 信号 OFF）后，才重新接收新的指令脉冲。用户可以利用函数 d2210_config_ERC_PIN 来设定 ERC 信号的工作方式。

相关函数：

d2210_config_ERC_PIN

5.8.4 SEVON 使能伺服驱动器信号

SEVON 是控制卡输出给伺服电机驱动器的控制信号，当 SEVON 信号为无效状态时，伺服驱动器不使能，电机处于自由状态；当 SEVON 信号有效时，伺服驱动器使能，电机锁紧。DMC1220 运动控制卡可使用函数 d2210_write_SEVON_PIN 输出使能信号，也可以通过一个通用数字输出口来输出 SEVON 信号。

相关函数：

d2210_write_SEVON_PIN

d2210_read_SEVON_PIN

5.8.5 RDY 伺服准备好信号

RDY 是伺服电机驱动器发给控制卡的状态信号，当 RDY 信号为有效时，表示伺服驱动器已经准备好，控制卡接收到该信号后就可以向伺服驱动器发出运动命令；如果 RDY 信号无效，表示伺服驱动器还未准备好，这时控制卡发出脉冲信号，伺服驱动器也不会按该命令运动。DMC1220 运动控制卡可使用函数 d2210_read_RDY_PIN 读取伺服准备好信号，也可以利用一个通用数字输入口来读取 RDY 信号。

相关函数：

d2210_read_RDY_PIN

d2210_read_inbit

第六章编程举例

前面已经对 DMC1220 卡的控制功能和库函数作了一些介绍,本章将详细介绍如何编写一个简单的程序:定长运动的控制,通过这个程序您可以了解到如何应用 VC、VB 等程序语言调用 DMC1220 运动控制卡的各种运动函数来实现运动控制功能。

如果您希望更深入地了解 DMC1220 运动控制功能的编程方法,可以参阅光盘中 Samples 目录下的例程源代码。我们在资料光盘中提供了以下几个示例:

示例一、单轴运动。卡的初始化,关闭操作,脉冲模式设置和单轴运动命令。

示例二、回原点运动,包括回原点的速度和方向的设置。

示例三、变速控制,目标位置改变。

示例四、多轴直线插补。

示例五、圆弧插补。

示例六、手轮功能,编码器反馈设置。

示例七、连续缓冲插补运动功能。

示例八、通用/专用输入输出。

示例九、编码器计数锁存。

示例十、位置比较输出。



注意: 在动手编写控制程序之前, 您需确认:

(1). PC 机上安装有所需要的开发工具, 如: Visual C++ 6.0 或 Visual Basic 6.0 等。

(2). 已经按照“第二章 软件安装”的步骤安装了控制卡(但不必接上驱动器和电机), 且用 Motion2210 测试软件对硬件进行过简单测试, 以确认硬件工作正常, 否则调用的运动函数不能正常工作。

6.1 Visual C++ 6.0 编程举例

- (1) 打开 Visual C++ 6.0;
- (2) 新建一个工程;
- (3) 选择 MFC APPWizard(exe);
- (4) 选择工程保存路径, 如: E:\;
- (5) 输入工程名, 如: test1。如图 6-1:

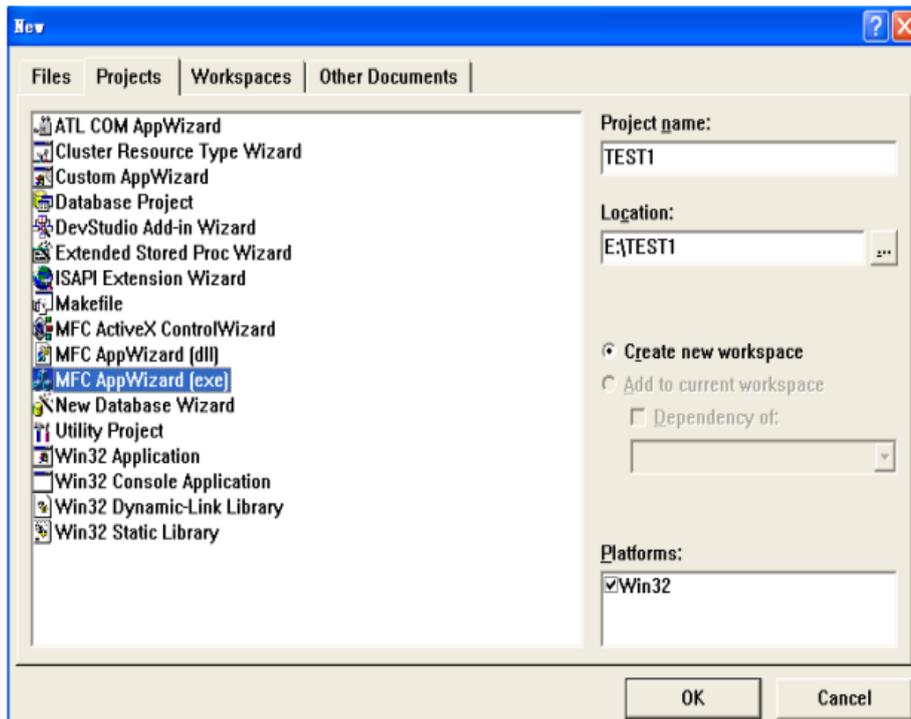


图 6-1 创建新工程

- (6) 在用程序类型中选择“基于对话框”，按“完成”键，建立工程。
- (7) 给对话框进行简单的修改，增加按钮“启动”（命名为 IDC_BUTTON_Start）和“停止”（命名为 IDC_BUTTON_Stop），如图 6-2 所示。

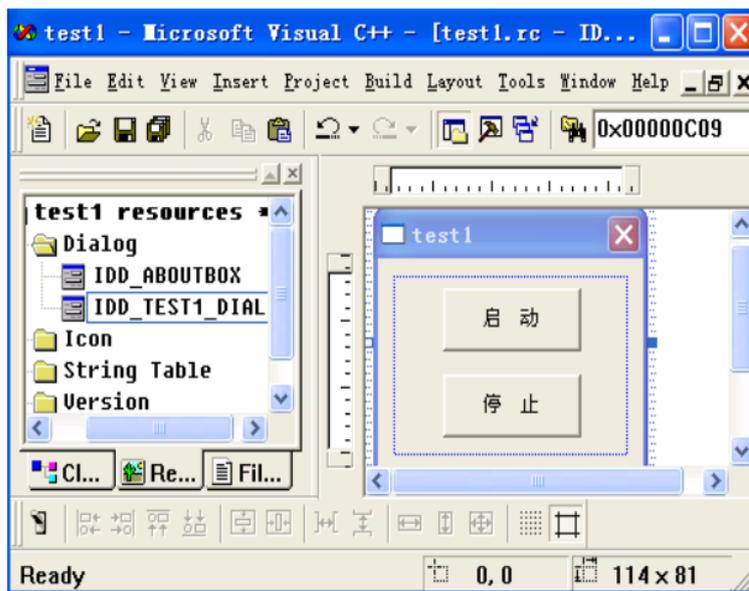


图 6-2 修改对话框

(8) 在应的目录下找到 DMC1220.h 和 DMC1220.lib 文件，拷贝到 E:\tes1 目录下。

(9) 选择“工程” -> “添加工程” -> “文件”，选中 DMC1220.lib 文件加入到工程中。

(10) 开 test1.cpp 文件，在程序开始部分添加语句：#include “DMC1220.h” 如图 6-3 所示：

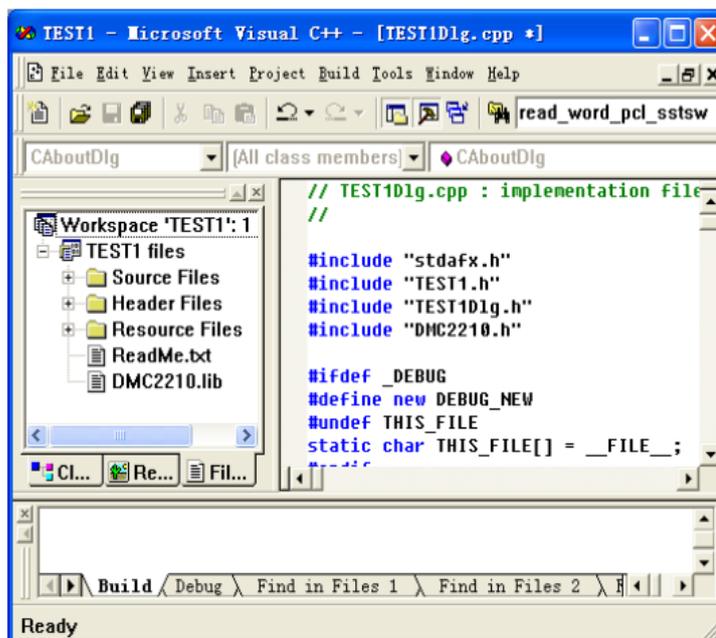


图 6-3 程序增加头文件

(11) 在 CTest1Dlg::OnInitDialog()函数中添加代码:

```
d2210_board_init();
```

如图 6-4:

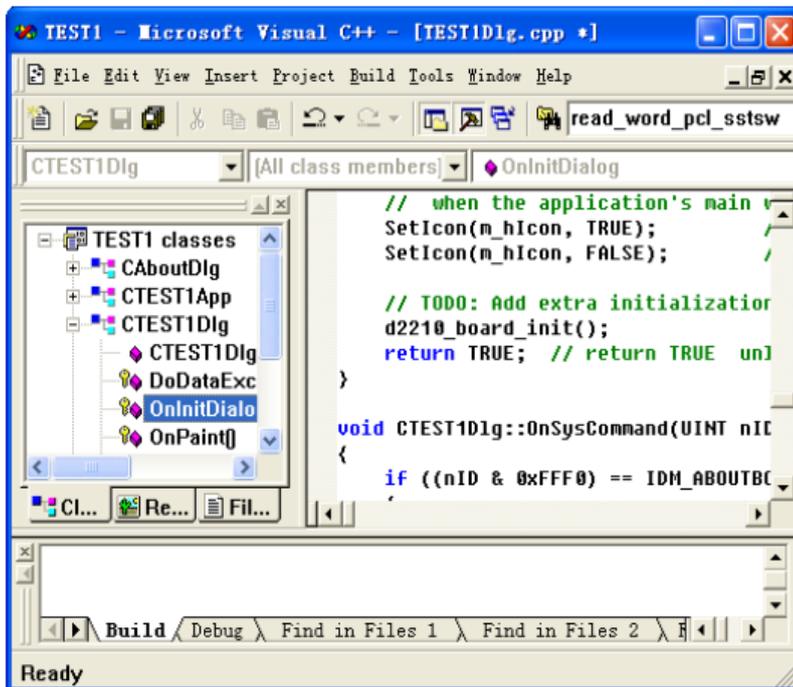


图 6-4 程序增加初始化函数

(12) 在 Ctest1Dlg 中添加一个成员函数 OnCancel,在 OnCancel 函数中添加代码:

```
d2210_board_close();
```

```
CDialog::OnCancel();
```

如图 6-5:

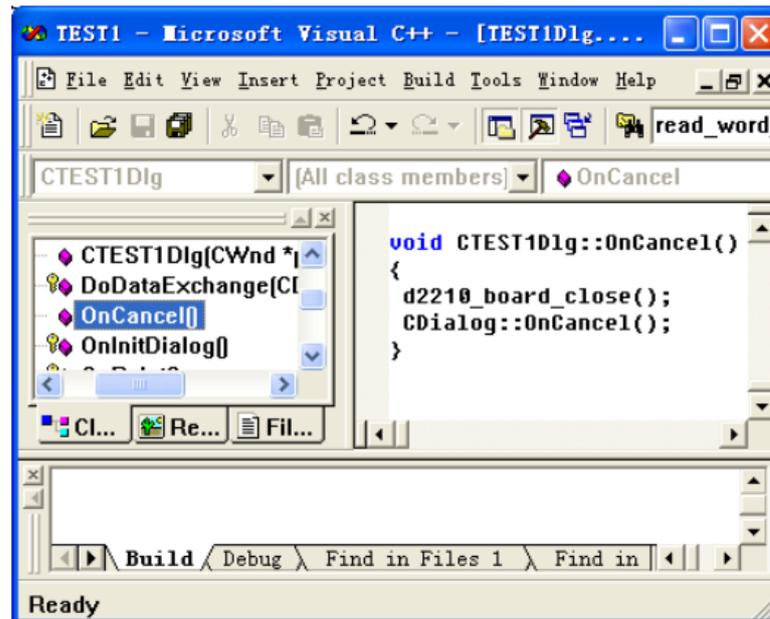


图 6-5 程序增加 OnCancel 函数

(13) 双击“启动”按钮在按钮点击事件中输入代码：

```
d2210_set_profile(0,500,5000, 0.01,0.01);
```

```
d2210_t_pmove(0,200000,0);
```

双击“停止”按钮在按钮点击事件中输入代码：

```
d2210_decel_stop(0,0.01);
```

如图 6-6 所示：

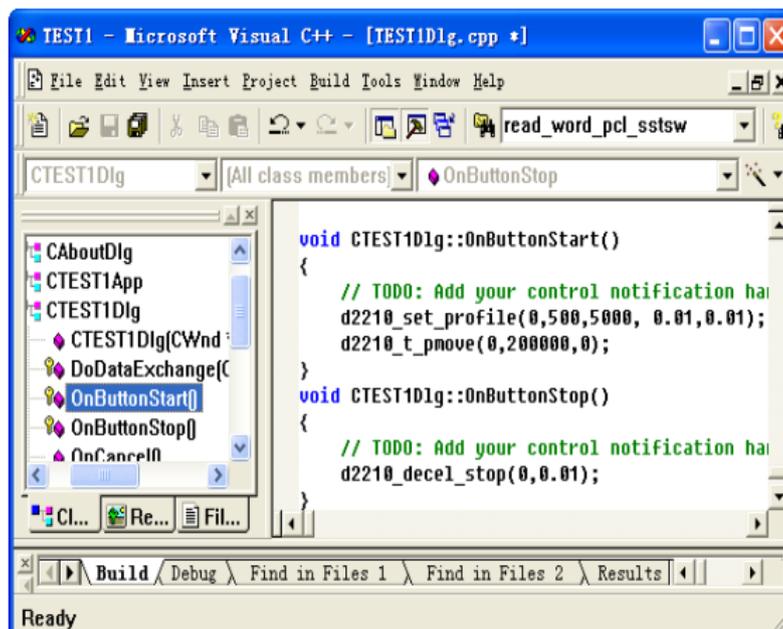


图 6-6 程序中调用运动控制卡库函数

(13) 编译，运行，按下“启动”按钮，第 0 轴就会输出长度为 200000 的脉冲，运动中可以按下“停止”按钮便会减速停止脉冲输出，如图 6-7 所示：

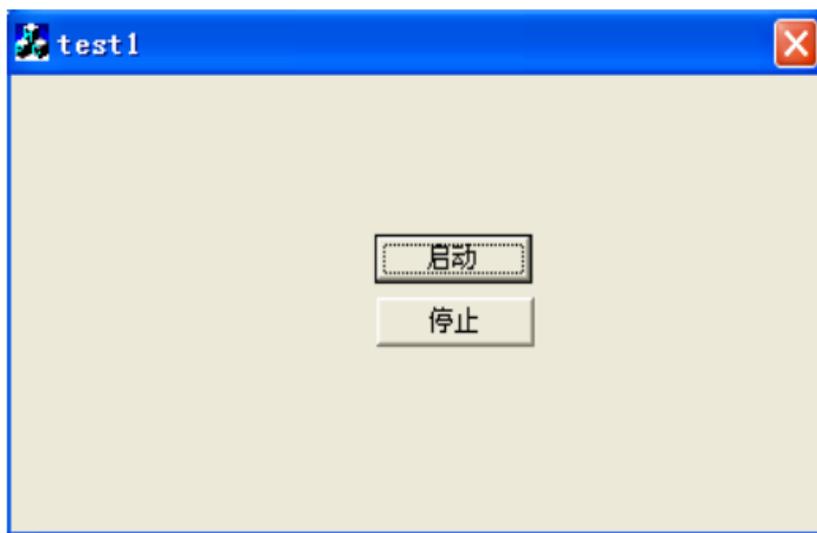


图 6-7 程序运行界面

6.2 Visual Basic 6.0 编程举例

- (1) 在磁盘上新建一个目录，如 E:\test1
- (2) 打开 Visual Basic 6.0
- (3) 新建一个“标注 EXE”工程，在对话框上添加按钮“启动”（名称修改为 CB_Start）和“停止”（名称修改为 CB_Stop），如图 6-8 所示：

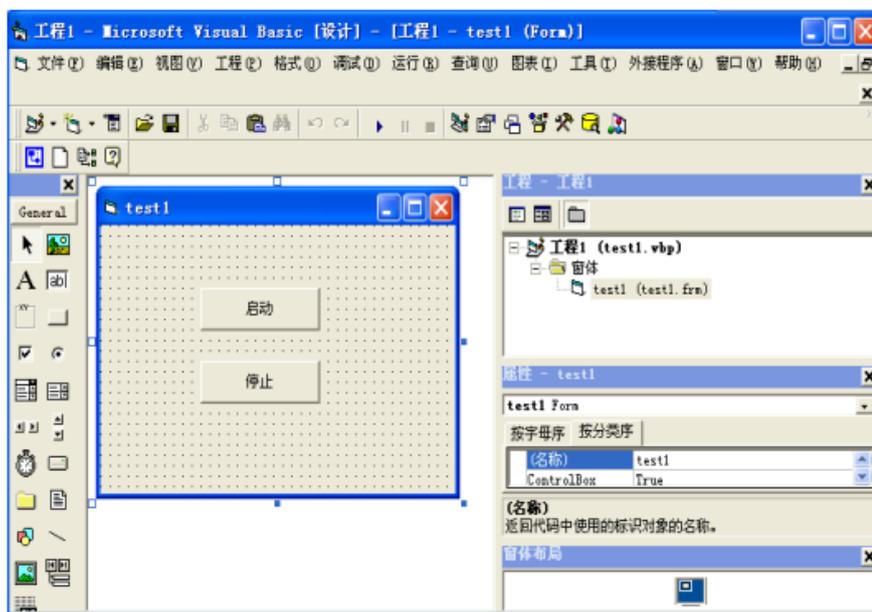


图 6-8 修改对话框 (VB)

- (4) 工程保存在 E:\test1 目录下。
- (5) 在资料光盘相应目录下找到 DMC1220.bas 文件, 拷贝到 test1 目录下。
- (6) 选择“工程” -> “添加模块” -> “现存”, 找到 test1 目录下的 DMC1220.bas 文件, 添加到工程中, 如图 6-9 所示:

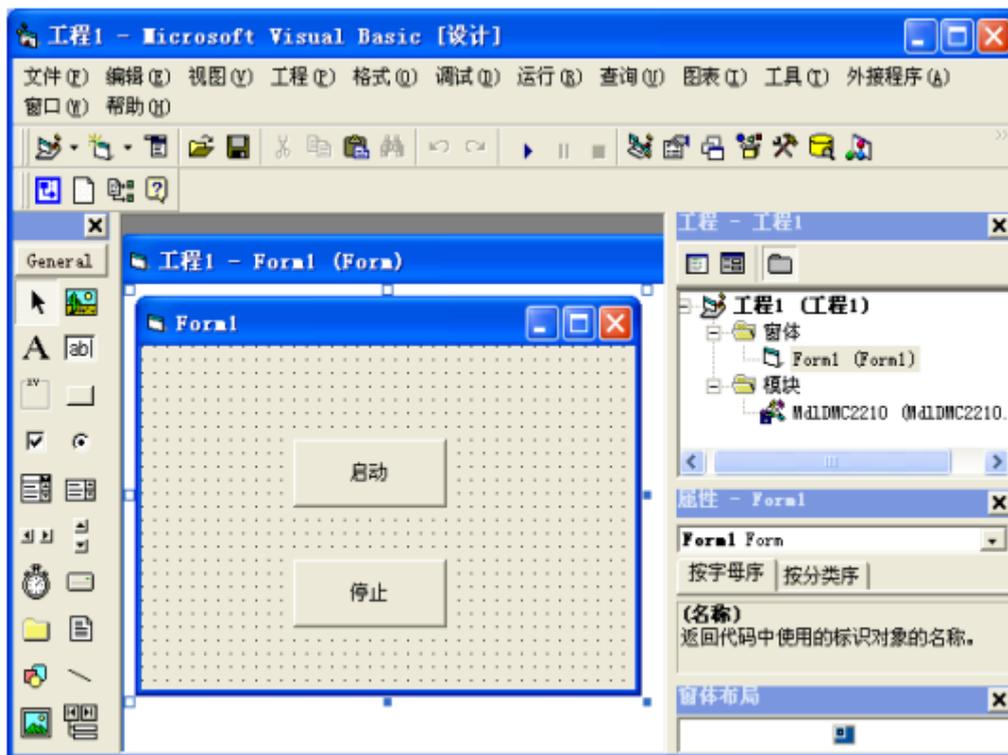


图 6-9 添加头文件

- (7) 双击窗口控件, 在 Form_Load 事件中添加代码:
d2210_board_init
- 选择 UnLoad 事件, 在 Form_Unload 事件中添加代码:
d2210_board_close
- 双击“启动”按钮, 在 CB_Start_Click 事件中添加代码:
d2210_set_profile 0,500,5000, 0.01,0.01
d2210_t_pmove 0,200000,0
- 双击“停止”按钮, 在 CB_Stop_Click()事件中添加代码:
d2210_decel_stop 0, 0.01

如图 6-10 所示:

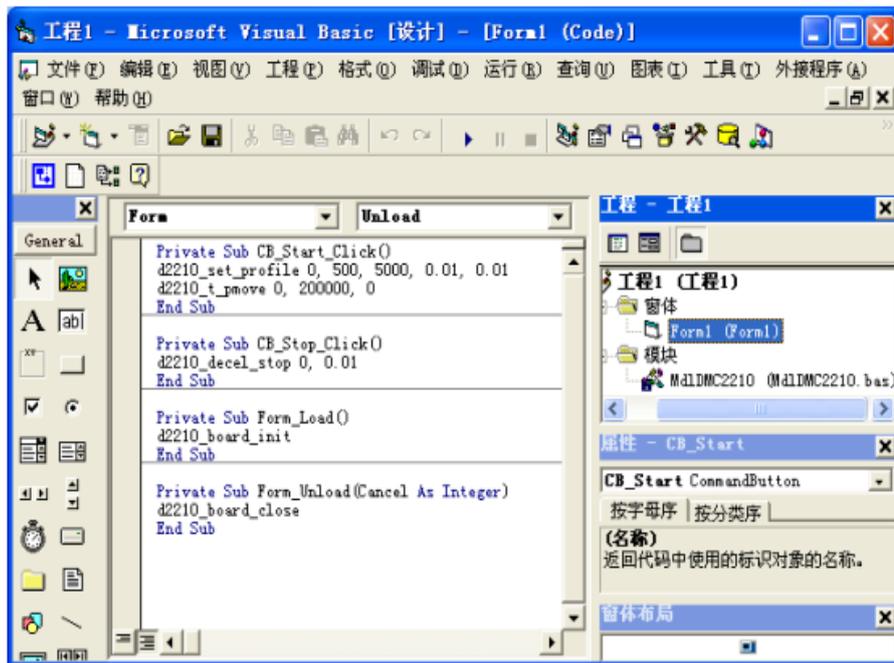


图 6-10 程序中调用运动控制卡库函数（VB）

（8）运行，按下“启动”按钮，第 0 轴就会输出长度为 20000 的脉冲，运动中可以按下“停止”按钮便会减速停止脉冲输出，如图 6-11 所示：



图 6-11 程序运行界面（VB）

第七章运动函数说明

DMC1220 运动控制卡的运动控制函数库包括 16 类 89 个库函数。在本章中，我们对这些函数分类进行详细的说明。如前所述，用户在具体应用中很可能只需调用函数库中的一小部分函数便可满足其实际需求。由于 DMC1220 用于无缝替换 DMC1220 卡，故 API 沿用 d2210_开头

库函数列表如下：

	函数名	描述
初始化函数	d2210_board_init	初始化控制卡
	d2210_board_close	关闭控制卡
	d2210_get_release_version	读取发布版本号
	d2210_get_lib_version	读取动态库版本
	d2210_get_card_version	读取控制卡硬件版本
	d2210_get_card_soft_version	读取控制卡硬件的固件版本
	d2210_board_reset	硬件复位
脉冲模式设置函数	d2210_set_pulse_outmode	设定脉冲输出模式
	d2210_counter_config	设定编码器输入脉冲的计数方式
运动速度设置函数	d2210_set_profile	设置梯形速度曲线
	d2210_set_st_profile	设置 S 形速度曲线（按 S 段时间）
	d2210_set_vector_profile	设置多轴插补运动的矢量速度曲线
	d2210_change_speed	在线改变指定轴的运动速度
	d2210_read_current_speed	读取指定轴的当前速度
	d2210_variety_speed_range	速度范围的设置
	d2210_decel_stop	单轴减速停止
	d2210_imd_stop	单轴立即停止

	d2210_simultaneous_stop	同步停止所有轴
	d2210_emg_stop	紧急停止所有轴
单轴位置运动控制函数	d2210_t_vmove	单轴梯形加速连续运动
	d2210_s_vmove	单轴 S 形加速连续运动
	d2210_t_pmove	单轴梯形定长运动（对称的 T 形加减速）
	d2210_ex_t_pmove	单轴梯形定长运动（非对称的 T 形加减速）
	d2210_s_pmove	单轴 S 形定长运动（对称的 S 形加减速）
	d2210_ex_s_pmove	单轴 S 形定长运动（非对称的 S 形加减速）
直线插补函数	d2210_t_line2	2 轴以对称的梯形加减速作直线插补
两轴圆弧插补函数	d2210_arc_move	两轴绝对位置圆弧插补
	d2210_rel_arc_move	两轴相对位置圆弧插补
回原点函数	d2210_config_home_mode	设置回原点模式
	d2210_home_move	启动回原点运动
	d2210_set_home_el_return	设置限位反找
手轮运动控制函数	d2210_set_handwheel_inmode_extern	设置多轴手轮脉冲计数方式
	d2210_handwheel_move	启动手轮运动
运动状态及相关 I/O 检测函数	d2210_check_done	读取指定轴的运动状态
	d2210_prebuff_status	读取缓存的状态
	d2210_get_rsts	读取指定轴外部信号的状态
	d2210_axis_io_status	读取指定轴有关运动信号的状态

驱动器专用接口 信号的设定函数	d2210_config_PCS_PIN	设置 PCS 信号
	d2210_set_pcs_position	触发 PCS 后执行在线变位
	d2210_config_INP_PIN	设置 INP 信号
	d2210_config_ERC_PIN	设置 ERC 信号
	d2210_config_ALM_PIN	设置 ALM 信号
	d2210_config_EZ_PIN	设置 EZ 信号
	d2210_config_LTC_PIN	设置 LTC 信号
	d2210_config_EL_MODE	设置 EL 信号
	d2210_config_HOME_pin _logic	设置 HOME 信号
	d2210_write_SEVON_PIN	输出 SEVON 信号
	d2210_read_SEVON_PIN	读取 SEVON 状态
	d2210_read_RDY_PIN	读取 RDY 状态
	d2210_write_ERC_PIN	控制 ERC 信号输出
位置计数器控制 函数	d2210_get_position	读取指定轴的当前位置
	d2210_set_position	设定指定轴的当前位置
	d2210_reset_target_positi on	在运动中改变目标位置
通用 I/O 接口函 数	d2210_read_inbit	读取输入口的状态
	d2210_write_outbit	设置输出口的状态
	d2210_read_outbit	读取输出口的状态
	d2210_read_inport	读取输入端口的值
	d2210_read_outport	读取输出端口的值
	d2210_write_outport	设置输出端口的值
编码器相关函数	d2210_get_encoder	读取编码器反馈脉冲计数值
	d2210_set_encoder	设置编码器反馈脉冲计数值

	d2210_config_latch_mode	设置锁存方式
	d2210_get_latch_value	读取编码器锁存器的值
	d2210_get_latch_flag	读取锁存器标志
	d2210_reset_latch_flag	复位锁存器标志

7.1 初始化函数

WORD d2210_board_init(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， (0~8)，其中 0 表示没有卡

void d2210_board_close(void)

功 能：关闭控制卡，释放系统资源

参 数：无

返回值：无

int32 d2210_board_reset(void)

功 能：控制卡硬件复位

参 数：无

返回值：错误码

void d2210_get_release_version(WORD CardNo,char *ReleaseVersion)

功 能：读取发布版本号

参 数： CardNo 控制卡卡号

ReleaseVersion 产品发布版本

返回值：无

void d2210_get_lib_version (long * libVer, long * subLibVer)

功 能： 读取动态库版本

参 数： CardNo 控制卡卡号

subLibVer 取动态库版本

返回值： 无

short d2210_get_card_version(WORD CardNo,DWORD *CardVersion)

功 能： 读取控制卡硬件版本

参 数： CardNo 控制卡卡号

CardVersion 硬件版本

返回值： 错误码

Short d2210_get_card_soft_version(WORD CardNo,DWORD *FirmID,DWORD *SubFirmID)

功 能： 获取控制卡固件版本号

参 数： CardNo 控制卡卡号

FirmID 返回控制卡固件类型

SubFirmID 返回控制卡固件版本号

返回值： 错误代码

注 意： 参数 FirmID 类型为十六进制

7.2 脉冲模式设置函数设置函数

void d2210_set_pulse_outmode(WORD axis, WORD outmode)

功 能： 设置指定轴的脉冲输出模式

参 数： axis 指定轴号

outmode 脉冲输出方式选择，其值如下表所示：

TYPE	When feeding in a positive direction		When feeding in a negative direction	
	OUT output	DIR output	OUT output	DIR output
0		High		Low
1		High		Low
2		Low		High
3		Low		High
4		High	High	
5		Low	Low	
6	OUT DIR		OUT DIR	
7	OUT DIR		OUT DIR	

返回值：无

注意事项：

在调用运动函数（如：`d2210_t_vmove` 等）输出脉冲之前，一定要根据驱动器接收脉冲的模式调用 `d2210_set_pulse_outmode` 设置控制卡脉冲输出模式。

void d2210_counter_config(WORD axis,WORD mode)

功 能：设置编码器的计数方式

参 数： axis 指定轴号

mode 编码器的计数方式：

0 保留

1 1X A/B

2 2X A/B

3 4X A/B

返回值：无

7.3 速度设置函数

void d2210_set_profile(WORD axis,double Min_Vel,double Max_Vel,double Tacc,double Tdec)

功 能： 设定梯形曲线的起始速度、 运行速度、 加速时间、 减速时间

参 数： axis 要设置的轴号

Min_Vel 起始速度， 单位 pps

Max_Vel 运行速度， 单位 pps

Tacc 总加速时间， 单位 s

Tdec 总减速时间， 单位 s

返回值： 无

void d2210_set_st_profile(WORD axis,double Min_Vel, double Max_Vel,double Tacc,double Tdec, double Tsacc, double Tsdec)

功 能： 设定 S 形曲线运动的起始速度、 运行速度、 总加减速时间、 S 段加减速时间

参 数： axis 要设置的轴号

Min_Vel 起始速度， 单位 pps

Max_Vel 运行速度， 单位 pps

Tacc 总加速时间， 单位 s

Tdec 总减速时间， 单位 s

Tsacc S 加速段时间， 其值应小于 Tacc 的一半。

Tsdec S 减速段时间， 其值应小于 Tdec 的一半。

返回值： 无

void d2210_set_vector_profile(double Min_Vel,double Max_Vel,double Tacc, doubleTdec)

功 能： 设定插补矢量运动曲线的起始速度、 运行速度、 加速时间、 减速时间

参 数： Min_Vel 起始速度，单位 pps

Max_Vel 运行速度，单位 pps

Tacc 总加速时间，单位 s

Tdec 总减速时间，单位 s

返回值：无

void d2210_change_speed(WORD axis,double Curr_Vel)

功 能：在线改变指定轴的当前运动速度。该函数只适用于单轴运动中的变速，且在调用前必须先调用 d2210_variety_speed_range 设置变速范围和使能

参 数： axis 要设置的轴号

Curr_Vel 新的运行速度，单位 pps

返回值：无

void d2210_variety_speed_range(WORD axis,WORD chg_enable ,double Max_Vel)

功 能：设定指定轴改变的速度上限，及变速使能

参 数： axis 要设置的轴号

chg_enable 禁止/使能连续运行中变速（禁止保留）

Max_Vel 运行速度的变速上限值，单位 pps

返回值：无

double d2210_read_current_speed(WORD axis)

功 能：读取当前速度值，单位 pps

参 数： axis 指定轴号

返回值：指定轴的速度脉冲数

void d2210_decel_stop(WORD axis,double Tdec)

功 能：指定轴减速停止，调用此函数时立即减速，减速到起始速度后停止脉冲输出

参 数： axis 指定轴号

Tdec 减速时间，单位 s

返回值：无

备注：当调用此指令时，减速停止时间按此函数设置的时间停止，其它异常情况（如 EL/ALM/EMG）导致的减速停止，其减速时间为速度规划函数的减速时间。

void d2210_imd_stop(WORD axis)

功 能：使指定轴立即停止，没有任何减速的过程

参 数： axis 指定轴号

返回值：无

void d2210_emg_stop(void)

功 能：使所有的运动轴紧急停止

参 数：无

返回值：无

7.4 单轴位置运动控制函数

void d2210_t_pmove(WORD axis,long Dist,WORD posi_mode)

功 能：使指定轴以对称梯形速度曲线做定长位移运动

参 数： axis 指定轴号

Dist （绝对/相对）位移值，单位：脉冲数

posi_mode 位移模式设定： 0 表示相对位移， 1 表示绝对位移

返回值：无

void d2210_ex_t_pmove(WORD axis,long Dist,WORD posi_mode)

功 能：使指定轴以非对称梯形速度曲线做定长位移运动

参 数： axis 指定轴号

Dist （绝对/相对）位移值，单位：脉冲数

posi_mode 位移模式设定： 0 表示相对位移， 1 表示绝对位移

返回值：无

void d2210_s_pmove(WORD axis,long Dist,WORD posi_mode)

功 能：使指定轴以对称 S 形速度曲线做定长位移运动

参 数： axis 指定轴号

Dist （绝对/相对）位移值，单位：脉冲数

posi_mode 位移模式设定： 0 表示相对位移， 1 表示绝对位移

返回值：无

void d2210_ex_s_pmove(WORD axis,long Dist,WORD posi_mode)

功 能：使指定轴以非对称 S 形速度曲线做定长位移运动

参 数： axis 指定轴号

Dist （绝对/相对）位移值，单位：脉冲数

posi_mode 位移模式设定： 0 表示相对位移， 1 表示绝对位移

返回值：无

void d2210_s_vmove(WORD axis,WORD dir)

功 能：使指定轴以 S 形速度曲线加速到高速，并持续运行下去

参 数： axis 指定轴号

dir 指定运动的方向，其中 0 表示负方向， 1 表示正方向

返回值：无

void d2210_t_vmove(WORD axis,WORD dir)

功 能：使指定轴以梯形速度曲线加速到高速，并持续运行下去

参 数： axis 指定轴号

dir 指定运动的方向，其中 0 表示负方向， 1 表示正方向

返回值：无

7.5 多轴直线插补运动控制函数

void d2210_t_line2(WORD axis1,long Dist1,WORD axis2,long Dist2,WORD posi_mode)

功 能：指定任意两轴以对称的梯形速度曲线做插补运动

参 数： axis1 指定两轴插补的第一轴

axis2 指定两轴插补的第二轴

Dist1 指定 axis1 的位移值，单位：脉冲数

Dist2 指定 axis2 的位移值，单位：脉冲数

posi_mode 位移模式设定： 0 表示相对位移， 1 表示绝对位移

返回值：无



注意：此函数可以对多条具有前后相连的线段实现连续缓冲插补的功能；当使用连续缓冲插补时，posi_mode 的值必须为 0，即设为相对位移模式。以下多轴插补函数均需同此设置。

7.6 两轴圆弧插补函数

void d2210_arc_move(WORD *axis,long *target_pos,long *cen_pos, WORD arc_dir)

功 能：指定任意的两轴以当前位置为起点，按指定的圆心、目标绝对位置和方向作圆弧插补运动

参 数： axis 轴号列表指针

target_pos 目标绝对位置列表指针，单位：脉冲数

cen_pos 圆心绝对位置列表指针，单位：脉冲数

arc_dir 圆弧方向： 0 表示顺时针， 1 表示逆时针

返回值：无

void d2210_rel_arc_move(WORD *axis,long *rel_pos,long *rel_cen, WORD arc_dir)

功 能：指定任意的两轴以当前位置为起点，按指定的圆心、目标相对位置和方向作圆弧插补运动

参 数： axis 轴号列表指针

rel_pos 目标相对位置列表指针，单位：脉冲数

cen_pos 圆心相对位置列表指针，单位：脉冲数

arc_dir 圆弧方向： 0 表示顺时针， 1 表示逆时针

返回值：无

7.7 回原点函数

void d2210_config_home_mode(WORD axis,WORD mode,WORD EZ_count)

功 能：设定指定轴的回原点模式

参 数： axis 指定轴号

mode 回原点的信号模式:0:只计 home, 1:计 home 和 EZ, 2:二次回零, 3:一次回零+反找原点, 10:一次限位回零, 11: 一次限位回零+反找限位, 12:二次限位回零

EZ_count 遇到原点信号后，EZ 信号出现 EZ_count 指定的次数后，轴运动停止。仅当 mode=1 时该设置有效，取值范围： 1—16

返回值：无

void d2210_home_move(WORD axis,WORD home_mode,WORD vel_mode)

功 能：单轴回原点运动

参 数： axis 指定轴号

home_mode 回原点方式： 1—正方向回原点， 2—负方向回原点

vel_mode 回原点速度： 0—低速回原点， 1—高速回原点

返回值：无

short d2210_set_home_el_return((WORD axis,WORD enable)

功 能：设置限位反找

参 数： axis 指定轴号

enable 限位翻找使能 0-不反找，1-限位反找(默认不使能反找)

返回值： 错误码

7.8 手轮运动控制函数

void d2210_set_handwheel_inmode_extern(WORD CardNo,WORD inmode,WORD AxisNum,WORD* AxisList,long* multi)

功能： 设置多轴手轮运动控制输入方式

参 数： CardNo 控制卡卡号

inmode 手轮输入方式： 0： A、B 相位正交信号(四倍频模式)； 1： 脉冲+方向信号

AxisNum 参与手轮运动的轴数

AxisList 参与手轮运动的轴号数组

multi 手轮倍率数组:正数表示默认方向， 负数表示与默认方向反向

返回值： 无

注 意： 通过该函数设置可以使一个手轮通道控制多个轴同时运动

void d2210_handwheel_move(WORD CardNo ,WORD axis)

功 能： 启动手轮运动

参 数： CardNo 控制卡卡号

axis 指定轴号， 取值范围：

返回值： 无

7.9 运动状态及轴信号检测函数

WORD d2210_check_done(WORD axis)

功 能： 检测指定轴的运动状态， 停止或是在运行中。

参 数： axis 指定轴号

返回值： 0 表示指定轴正在运行， 1 表示指定轴已停止。

WORD d2210_prebuff_status(WORD axis)

功 能： 读取指定轴的预置缓冲区的状态

参 数： axis 指定轴号

返回值： 0—缓冲区空， 1—缓冲区满

DWORD d2210_get_rsts(WORD axis)

功 能： 读取指定轴的外部信号状态

参 数： axis 指定轴号

返回值： 见表

位号	信号名称	描述
0~6	保留	
7	EMG	1: 表示紧急停止信号 (EMG) 为 ON
8	PCS	1: 表示 PCS 信号为 ON
9	ERC	1: 表示误差清除信号 (ERC) 为 ON
10	EZ	1: 表示索引信号 (EZ) 为 ON
11~13	保留	
14	SD	1: 表示 SD 信号为 ON
15	INP	1: 表示到位信号 INP 为 ON
16	DIR	脉冲输出方向(0: 表示正方向; 1: 表示负方向)
17~31	保留	

WORD d2210_axis_io_status(WORD axis)

功 能： 读取指定轴有关运动信号的状态， 包含指定轴的专用 I/O 状态。

参 数： axis 指定轴号

返回值： 见下表

位号	信号名称	描述
0~7	保留	
8	FU	1: 表示正在加速
9	FD	1: 表示正在减速
10	保留	
11	ALM	1: 表示伺服报警信号 ALM 为 ON
12	PEL	1: 表示正限位信号 +EL 为 ON
13	MEL	1: 表示负限位信号 -EL 为 ON
14	ORG	1: 表示原点信号 ORG 为 ON
15	保留	

7.10 驱动器专用接口信号的设定函数

void d2210_config_PCS_PIN(WORD axis,WORD enable,WORD pcs_logic)

功 能：设置允许/禁止 PCS 外部信号在运动中改变目标位置

参 数： axis 指定轴号

enable 允许/禁止信号功能： 0—无效， 1—有效

pcs_logic 设置 PCS 信号的有效电平： 0—低电平有效， 1—高电平有效

返回值：无

void d2210_set_pcs_position(WORD axis,double position)

功 能：触发 PCS 后执行在线变位

参 数： axis 指定轴号

position 新位置值

返回值：无

void d2210_config_INP_PIN(WORD axis,WORD enable,WORD inp_logic)

功 能：设置允许/禁止 INP 信号及其有效的逻辑电平

技术支持热线：0755-26417593

网址：www.szleadtech.com.cn 83

参 数： axis 指定轴号

enable 允许/禁止信号功能： 0—无效， 1—有效

inp_logic 设置 INP 信号的有效电平： 0—低电平有效， 1—高电平有效

返回值： 无

void d2210_config_ERC_PIN(WORD axis,WORD enable,WORD erc_logic,WORD erc_width,WORD erc_off_time)

功 能： 设置允许/禁止 ERC 信号及其有效电平和输出方式

参 数： axis 指定轴号

enable 范围： 0~3:

0—不自动输出 ERC 信号

1—接收 EL、 ALM、 CEMG 等信号停止时，自动输出 ERC 信号

2—接收 ORG 信号时，自动输出 ERC 信号

3—满足第 1 或 2 两项条件时，均会自动输出 ERC 信号

erc_logic 设置 ERC 信号的有效电平： 0—低电平有效， 1—高电平有效

erc_width 误差清除信号 ERC 有效输出宽度，范围 1~2000，单位 250us

erc_off_time ERC 保留

返回值： 无

void d2210_config_ALM_PIN(WORD axis,WORD alm_logic,WORD alm_action)

功 能： 设置 ALM 的逻辑电平及其工作方式

参 数： axis 指定轴号

alm_logic ALM 信号的输入电平： 0—低电平有效， 1—高电平有效

alm_action ALM 信号的制动方式： 0—立即停止， 1—减速停止

返回值： 无

void d2210_config_EZ_PIN(WORD axis,WORD ez_logic, WORD ez_mode)

功 能：设置指定轴的 EZ 信号的有效电平及其作用

参 数： axis 指定轴号

ez_logic EZ 信号逻辑电平： 0—低有效， 1—高有效

ez_mode 保留

返回值：无

void d2210_config_LTC_PIN(WORD axis,WORD ltc_logic, WORD ltc_mode)

功 能：设置指定轴锁存信号的有效电平。

参 数： axis 指定轴号

ltc_logic LTC 信号逻辑电平： 0—低有效， 1—高有效

ltc_mode 保留，可设为任意值

返回值：无

void d2210_config_EL_MODE(WORD axis,WORD el_mode)

功 能：设置 EL 信号的有效电平及制动方式

参 数： axis 指定轴号

el_mode EL 有效电平和制动方式：

0—立即停、低有效

1—减速停、低有效

2—立即停、高有效

3—减速停、高有效

返回值：无

void d2210_set_HOME_pin_logic(WORD axis,WORD org_logic,WORD filter)

功 能：设置 ORG 信号的有效电平，以及允许/禁止滤波功能

参 数： axis 指定轴号

org_logic ORG 信号的有效电平 0-低电平有效， 1—高电平有

效

filter 备用

返回值：无

void d2210_write_SEVON_PIN(WORD axis, WORD on_off);

功 能：输出对指定轴的伺服使能端子的控制

参 数： axis 指定轴号

on_off 设定管脚电平状态： 0—低， 1—高。 SEVON 输出口初始状态可选。

返回值：无

int d2210_read_SEVON_PIN(WORD axis)

功 能：读取指定轴的“伺服使能”端子的电平状态

参 数： axis 指定轴号

返回值： 0—低电平， 1—高电平。 SEVON 输出口初始状态可选。

int d2210_read_RDY_PIN(WORD axis)

功 能：读取指定运动轴的“伺服准备好”端子的电平状态

参 数： axis 指定轴号

返回值： 0—低电平， 1—高电平

void d2210_write_ERC_PIN(WORD axis, WORD sel)

功 能：控制指定轴“误差清除”端子信号的输出

参 数： axis 指定轴号

sel 0—复位 ERC 信号， 1—输出 ERC 信号

返回值：无

void d2210_config_EMG_PIN(WORD cardno, WORD enable,WORD emg_logic)

功能描述： EMG 信号设置，急停信号有效后会立即停止所有轴。

输 入： cardno 卡号

Enable 0： 无效； 1： 有效

emg_logic 0: 低有效; 1: 高有效

返回值: 无

7.11 位置计数器控制函数

long d2210_get_position(WORD axis)

功 能: 读取指定轴的指令脉冲位置

参 数: axis 指定轴号

返回值: 指定运动轴的命令脉冲数, 单位: 脉冲

void d2210_set_position(WORD axis, long current_position)

功 能: 设置指定轴的指令脉冲位置

参 数: axis 指定轴号

current_position 绝对位置值

返回值: 无

void d2210_reset_target_position(WORD axis, long dist)

功 能: 在相对模式的单轴定长运动中改变目标位置。

参 数: axis 指定轴号

dist 相对位置值

返回值: 无

7.12 通用 I/O 控制函数

int d2210_read_inbit(WORD cardno, WORD bitno)

功 能: 读取指定控制卡的某一位输入口的电平状态

参 数: cardno 指定控制卡号, 范围 (0~N - 1, N 为卡数)

bitno 指定输入口位号（取值范围： 1~20）

返回值： 0 表示低电平； 1 表示高电平

void d2210_write_outbit (WORD cardno, WORD bitno,WORD on_off)

功 能：对指定控制卡的某一位输出口置位

参 数： cardno 指定控制卡号，范围（0~N - 1， N 为卡数）

bitno 指定输出口位号（取值范围： 1~20）

on_off 输出电平： 0—表示输出低电平， 1—表示输出高电平

拨码 ON 时： OUT（1-12）初始电平为低； OUT(13-18)固定高电平；

拨码 OFF 时： OUT（1-12）初始电平为高； OUT(13-18)固定高电平；

返回值： 无

int d2210_read_outbit(WORD cardno, WORD bitno)

功 能：读取指定控制卡的某一位输出口的的电平状态

参 数： cardno 指定控制卡号，范围（0 - N - 1,N 为卡数）

bitno 指定输入口位号（取值范围： 1—20）

返回值： 0 表示低电平； 1 表示高电平。

当拨码开关 S1 对应的位设置为 ON 时，读取输出口 1 - 12 的状态时， 0 表示高电平； 1 表示低电平。

long d2210_read_inport(WORD cardno)

功 能：读取指定控制卡的全部通用输入口的电平状态

参 数： cardno 指定控制卡号，范围（0~N - 1， N 为卡数）

返回值： bit0~bit19 位值分别代表第 1~20 号输入端口值

long d2210_read_outport(WORD cardno)

功 能：读取指定控制卡的全部通用输出口的电平状态

参 数： cardno 指定控制卡号，范围（0~N - 1， N 为卡数）

返回值： bit0~bit19 位值分别代表第 1~20 号输出端口值

当拨码开关 S1 对应的位设置为 ON 时，读取输出口 1 - 12 的状态时， 0 表示高电平； 1 表示低电平。

void d2210_write_outport(WORD cardno, DWORD port_value)

功 能：指定控制卡的全部通用输出端口的电平状态

参 数： cardno 指定控制卡号，范围（0~N - 1， N 为卡数）

port_value bit0~bit19 位值分别代表第 1~20 号输出端口值。

当拨码开关 S1 对应的位设置为 ON 时，对输出口 1 - 12 置位时， 0 表示高电平； 1 表示低电平。

返回值：无

7.13 编码器相关函数

long d2210_get_encoder(WORD axis)

功 能：读取指定轴编码器反馈位置脉冲计数值，范围： 28 位有符号数

参 数： axis 指定轴号

返回值：位置反馈脉冲值，单位：脉冲数

void d2210_set_encoder(WORD axis, long encoder_value)

功 能：设置指定轴编码器反馈脉冲计数值，范围： 28 位有符号数

参 数： axis 指定轴号

encoder_value 编码器的设定值。

返回值：无

void d2210_config_latch_mode(WORD cardno, WORD all_enable)

功 能：设置锁存方式为单轴锁存或是两轴同时锁存

参 数： cardno 指定控制卡号

all_enable 锁存方式： 0—单独锁存， 1—两轴同时锁存

返回值：无

long d2210_get_latch_value(WORD axis)

功 能：读取编码器锁存器的值

参 数： axis 指定轴号

返回值：锁存器内的编码器脉冲数，单位：脉冲

long d2210_get_latch_flag(WORD cardno)

功 能：读取指定控制卡的锁存器的标志位

参 数： cardno 指定控制卡号，范围（0~N - 1， N 为卡数）

返回值：见下表

返回值位号	指定卡的状态位意义描述
0	1: 0 轴有触发信号.
1	1: 1 轴有触发信号.
8	1: 0 轴位置已锁存
9	1: 1 轴位置已锁存

void d2210_reset_latch_flag(WORD cardno)

功 能：复位指定控制卡的锁存器的标志位

参 数： cardno 指定控制卡号

返回值：无



深圳市雷赛控制技术有限公司
SHENZHEN LEADSHINE CONTROL TECHNOLOGY CO.,LTD

深圳市雷赛控制技术有限公司

地 址：深圳市南山区学苑大道 1001 号南山智园 A3 栋 9 楼

邮 编：518052

电 话：0755-26415968

传 真：0755-26417609

Email: info@szleadtech.com.cn

网 址: <http://www.szleadtech.com.cn>