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

雷赛运动控制卡

DMC1000S

用户手册

Version 1.0

2020.8.23

版 权 说 明

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，雷赛控制技术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，雷赛控制技术没有义务或责任负责。

关于本手册

本手册旨在帮助你学习 DMC1000S 控制卡的使用，包括软件函数的调用、参数的设置、硬件接线以及应用软件的编写等。

本手册总共分为 8 个部分：

1．引言：关于本产品的大概描述和关于本产品的相关申明；

2．硬件概述：关于本产品的硬件相关介绍，包括详细硬件结构尺寸和硬件功能描述，以及控制卡与步进电机驱动器接线示例等；

3．硬件配置与安装：包括硬件安装步骤以及板卡设置；

4．软件系统概述：关于本产品的软件相关介绍，包括本产品所支持的系统驱动程序、运动控制函数库详细说明、演示程序和例子程序的说明以及多卡运行部分；

5．驱动程序安装：控制卡在各种 Windows 操作系统下驱动程序的详细安装步骤；

6．演示软件及应用：包括演示软件的说明和使用方法；

7．用户系统开发：介绍了基于 windows 平台的应用软件结构，以及 VB 和 VC++开发环境下的开发方法，并提供 VB 和 VC++开发环境下的应用实例编程入门；

8．附录：提供详细硬件信号接口定义表、运动控制函数库、DMC1000S 一些说明、常见问题解决方法、抗干扰措施等。

第 1 章 引言

雷赛 DMC1000S 是一款基于 PCI 总线的高集成度、高可靠度的脉冲式运动控制卡，可控制多达 4 个步进电机或伺服电机。

DMC1000S 卡内含脉冲和方向接口，其位置指令可用单路脉冲（脉冲+方向）或双路脉冲（CW+CCW 脉冲）方式输出；可以是差分式输出电路也可以是单端式输出电路。

另外，除了通用输入输出信号接口外，DMC1000S 还包括原点、限位、减速等专用信号接口，具有即插即用、多轴同时启停功能，并可选择梯形或 S 形速度曲线，同时具有直线插补功能。

雷赛公司为 DMC1000S 设计了一套易学易用、功能丰富的运动函数库，大大缩短了用户应用软件开发和调试的时间。随卡免费提供的 Motion 软件，不但可以演示和测试 DMC1000S 的绝大多数控制功能，而且还可方便客户测试电机、IO 信号等硬件系统。

第 2 章 硬件概述

雷赛 DMC1000S 是一款兼容 PCI V2.3 标准的 32Bit PCI 标准半长卡。硬件方面分别提供了 4 轴的脉冲和方向控制信号，同时提供多种运动控制功能、机械位置控制信号、以及通用输入输出信号。

具体硬件系统框图如图 2-1 所示：

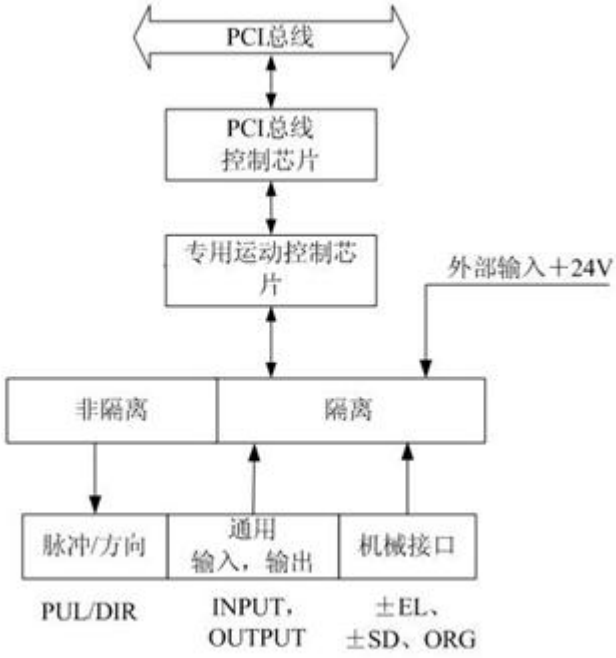


图 2-1 DMC1000S 运动控制卡系统框图

2.1 性能指标

2.1.1 电机控制指标

- 控制轴数：4 轴，最多可支持 8 张控制卡同时工作；
- 控制模式：位置控制、速度控制 2 种模式；
- 脉冲输出模式： 单脉冲（脉冲+方向）或双脉冲（CW +CCW）；
- 最大脉冲输出频率：4MHZ，可梯形速度曲线或 S 形速度曲线控制；
- 位置脉冲设置范围： －2147483648~2147483647 个脉冲(32 位)；

2.1.2 I/O 信号控制指标

- 通用 I/O 信号接口： 67 路。

通用输入信号接口：40 路，其中 24 路光电隔离；**其中 IN1~IN3 与轴 ALM 信号复用，**

默认为 4 个轴的 ALM 信号，作为通用输入需调用 d1000_set_ALM_PIN_Extern 函数关闭 ALM 功能。

通用输出信号接口：27 路，其中 12 路光电隔离；

- 专用 IO 信号接口： 12 路，包括正负限位信号±EL 以及原点信号 ORG

全部光电隔离；

- 通用数字输出口（光电隔离）最大驱动电流： 100mA；

2.1.3 通用指标：

- 工作温度： 0℃～50℃；
- 工作湿度： 5～85%， 非结露；
- 贮存温度： -20℃～80℃；
- 电源：

内部芯片电源(由 PCI 总线提供)： +5VDC±5%，最大 900mA；

外部接口电源(需用户提供)： +24VDC±5%，最大 500mA；

2.2 硬件结构尺寸

DMC1000S 数字运动控制卡外形结构尺寸分别如图 2-2 所示：

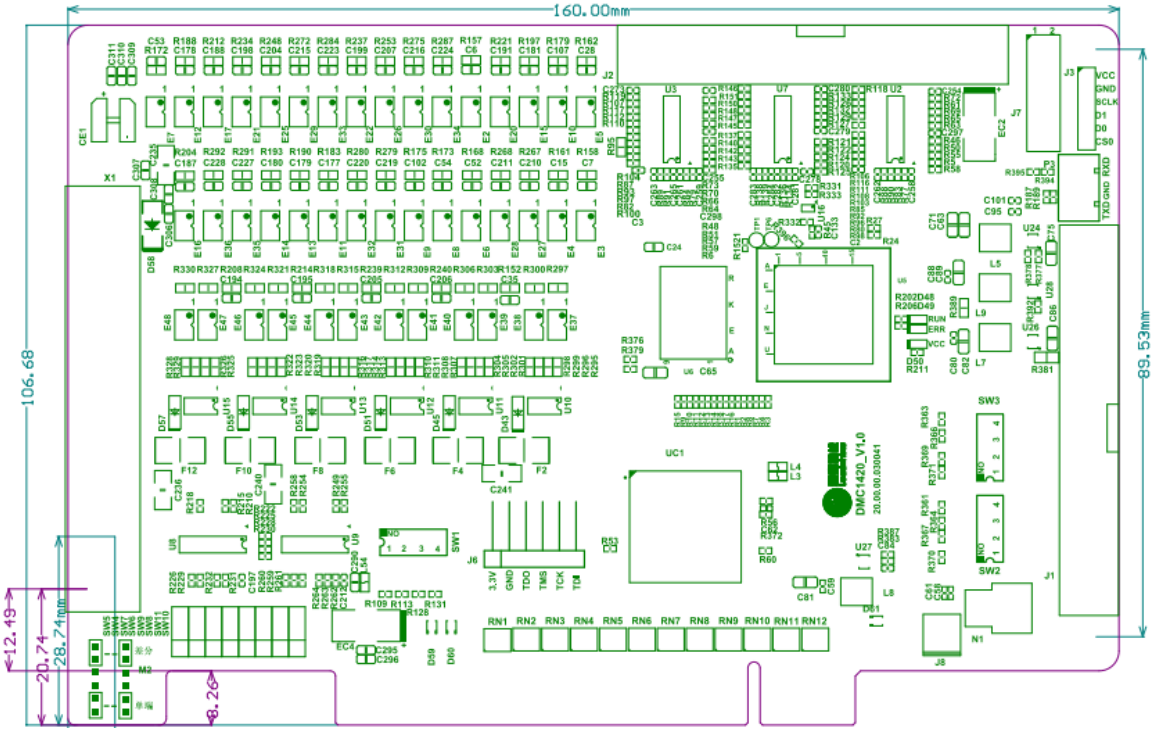


图 2-2 DMC1000S 硬件结构尺寸示意图

2.3 硬件功能描述

DMC1000S 运动控制卡硬件方面提供了多种运动控制功能，同时提供了 4 轴的脉冲和方向控制信号、机械位置控制信号、以及通用输入输出信号接口。

2.3.1 控制卡电源供给

DMC1000S 运动控制卡内部 IC 电源由 PC 机 PCI 总线提供（+5VDC），外部接口需用户提供+24VDC 电源。接线示意图如图 2-3 所示：

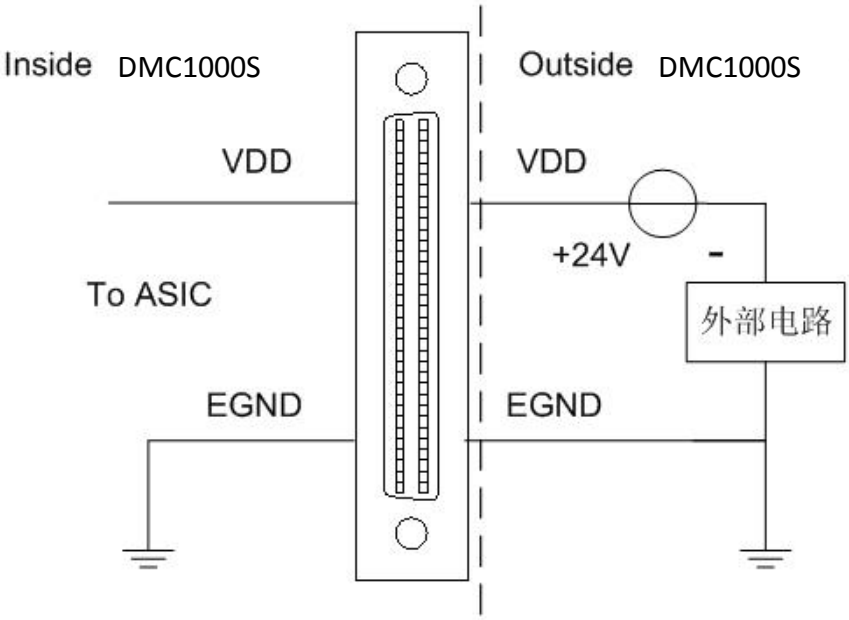


图 2-3 外部 I/O 供电接线示意图

2.3.2 运动控制功能

DMC1000S 运动控制卡为用户提供了丰富的运动控制功能，其中包括：位置控制、梯形或 S 型速度控制、直线插补运动控制等功能。

2.3.2.1 位置控制

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。设置加/减速度以及起始速度和最大速度等参数后执行位移控制指令，上位机将要执行的指令脉冲数写入 DMC1000S 运动控制卡， DMC1000S 卡即按设定的速度输出脉冲，当输出脉冲数等于指令脉冲数时，DMC1000S 卡将停止输出脉冲。位移与时间的关系如图 2-4 所示：

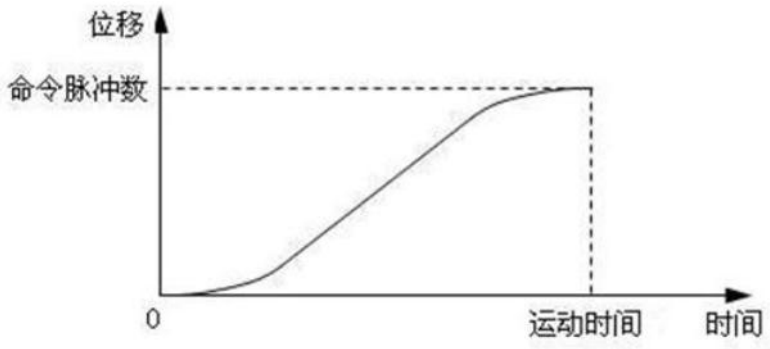


图 2-4 位移-时间曲线

2.3.2.2 速度控制

速度控制是指电机从起始速度开始运行，加速至指定速度连续运动。只有当接收到停止命令或外部停止信号后，才减速直至停止（也可设为立即停止）。其速度与时间的关系如图 2-5 所示。

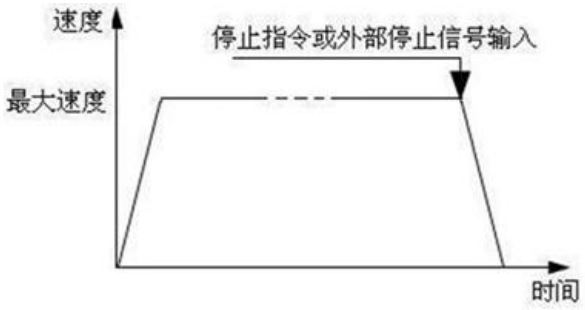


图 2-5 速度控制曲线

DMC1000S 运动控制卡为用户提供了梯形、S 型速度控制模式。

A）梯形曲线速度控制

梯形速度曲线控制指令是使 DMC1000S 卡按梯形速度曲线输出指令脉冲。即：电机从起始速度开始运动，加速至最大速度后保持速度不变，临结束前减速至起始速度，并停止。梯形速度曲线如图 2-6 所示。

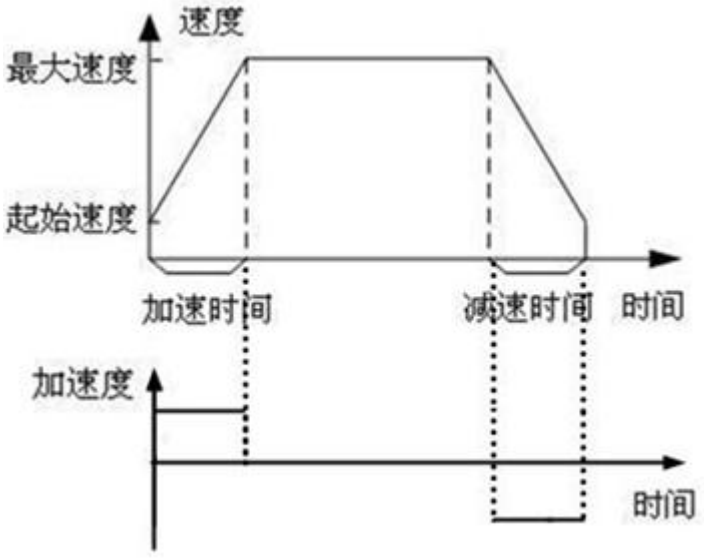


图 2-6 梯形速度曲线及其加速度曲线

B）S 形曲线速度控制

梯形速度曲线虽然实现起来简单，但它的加速度有突变，速度曲线不平滑，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 2-6），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。

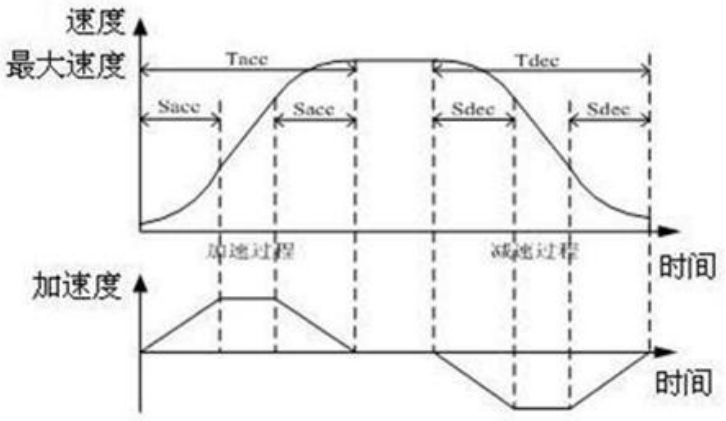


图 2-7 S 形速度曲线及其加速度曲线

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 2-8 所示。升速和减速阶段均变得像 S 的形状。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，同时还可以延长机械传动部分的寿命。

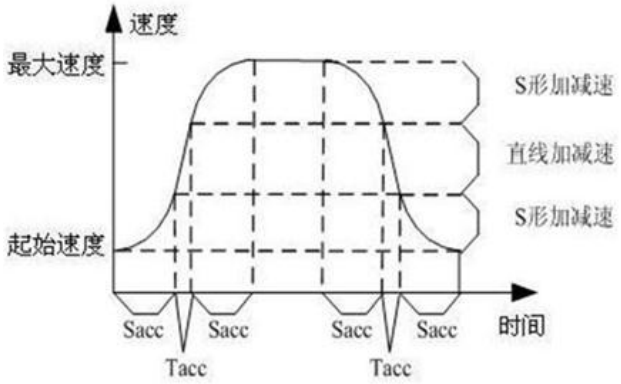


图 2-8 S 形速度曲线

2.3.2.3 插补运动

插补即根据给定的数学函数，在理想的轨迹式轮廓上的已知点之间，确定一些中间点的一种方法。插补方式有：直线插补、圆弧插补、抛物线插补、样条线插补等。

DMC1000S 卡具有直线插补功能，可以选择任意的二轴至四轴进行直线插补（详细介绍请参考 4.2.4.2 直线插补运动中相关内容）。

2.3.3 脉冲和方向控制接口

DMC1000S 硬件方面提供 4 轴的脉冲和方向控制信号接口。用户可参考 3.1.1 跳线配置，通过设定 SW4~SW11 跳线来设定指令脉冲为差分输出或单端输出两种电路（出厂默认均为差分输出方式）。两种典型接口电路输出分别如图 2-9 所示。对外接口 SCSI68 具体引脚分配见 8.1.1 接口 X1 引脚定义。

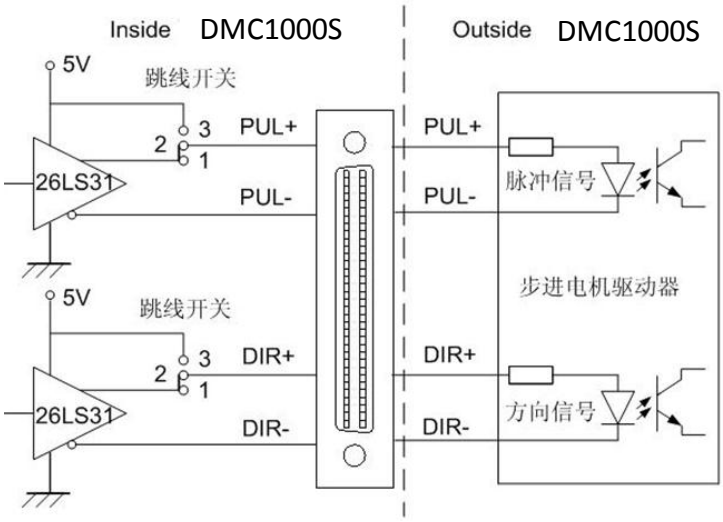


图 2-9 差分输出方式接线图

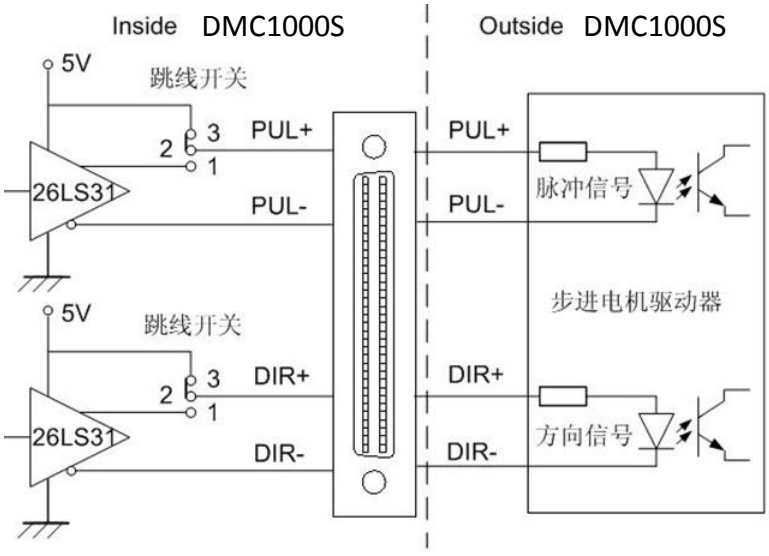


图 2-10 单端输出方式接线图

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

如图 2-9 所示，如果指令脉冲输出采用差分输出模式，则 PUL+、PUL-和 DIR+、DIR-端子组成互异的脉冲和方向信号。如图 2-10 所示，如果指令脉冲输出方式为单端输出，则 PUL+和 DIR+共用内部+5V 电压，PUL-和 DIR-输出端为脉冲和方向信号。

请注意流过 PUL-和 DIR-端的电流不能超过 20mA。此电流由 DMC1000S 卡通过 PUL+或 DIR+端子提供。

DMC1000S 运动控制卡可以输出两类脉冲信号：一种为脉冲+方向形式（单脉冲）；一种为正脉冲+负脉冲（双脉冲）形式。

单脉冲模式输出信号如图 2-11 所示：

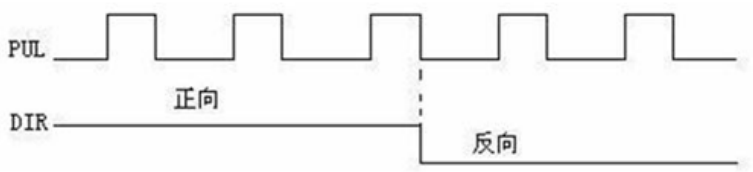


图 2-11 单脉冲模式

双脉冲模式输出信号如图 2-12 所示：

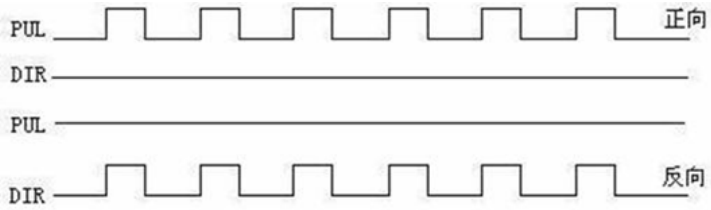


图 2-12 双脉冲模式

2.3.4 机械位置控制接口

DMC1000S 为每个轴提均供了 3 个专用信号输入接口，分别为 1 个原点信号（ORG）输入接口以及 2 个限位信号（EL+/EL-）输入接口。各信号相关功能的实现均由控制卡硬件来完成，用户只需按照正确的接线方式接线，并调用相应功能的函数执行运动指令即可。

2.3.4.1 ORG：原点位置信号输入接口

通常运动系统中都要用一个位置传感器设置一个位置参考点，即原点位置，以便于进行精确的位置控制。DMC1000S 为每个轴提供了 1 个原点位置传感器输入端口 ORG，对外接口 SCSI68 具体引脚分配见 8.1.1 接口 X1 引脚定义，典型接口电路如图 2-13 所示。原点开关请选用+24V、最小电流大于 6mA 的常开型开关。原点信号输入电路有低通滤波器，可以过滤高频噪声，提高可靠性。

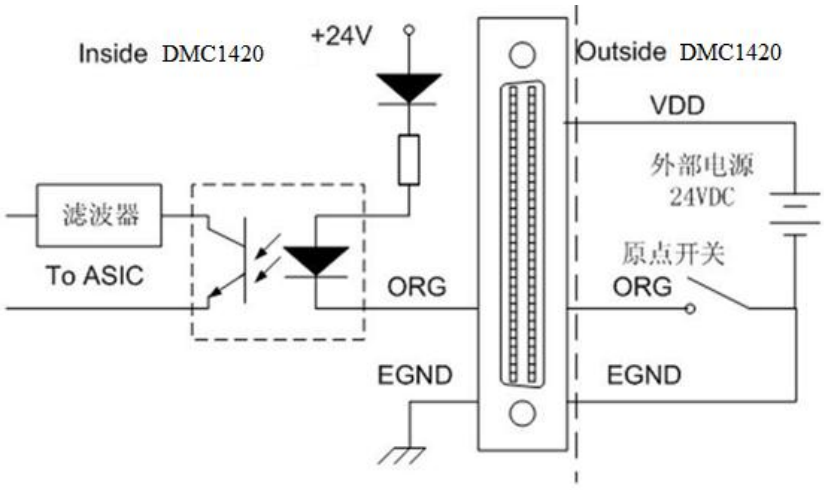


图 2-13 原点信号接口原理图

通常在进行运动控制之前，都需要用回原点运动控制平台向原点方向运动，当运动控制卡检测到原点传感器 ORG 信号后，运动自动停止，并将停止位置设置为该轴的原点。具体方式和操作请参考 4.2.5 回原点运动。

2.3.4.2 EL+/EL-：正负限位信号输入接口

运动系统中通常会用一个位置传感器设置一个机械限位点，以确定运动的边界位置，保护机械设备。DMC1000S 均为每个轴提供 2 个机械限位信号 EL+ 和 EL-，EL+为正向限位信号，EL-为反向限位信号。当运动部件接触到限位开关时，EL+/EL-即有效，DMC1000S 将立即停止向该方向输出脉冲。典型接口及接线原理如图 2-14 所示，对外接口 SCSI68 具体信号引脚分配见 8.1.1 接口 X1 引脚定义。

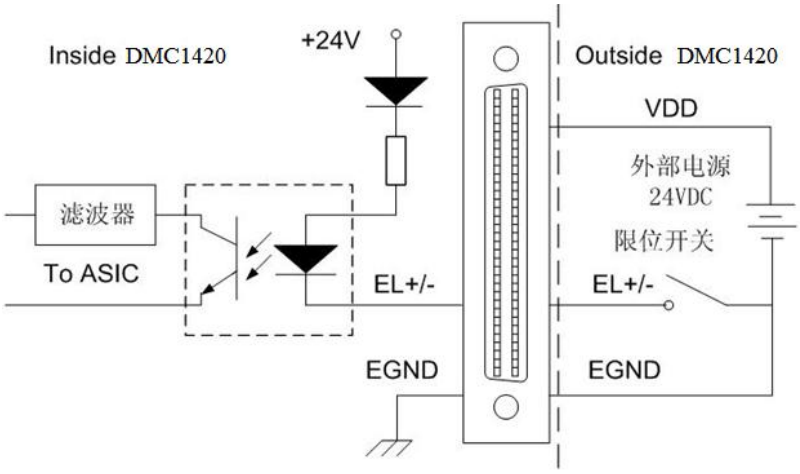


图 2-14 限位信号接口原理图

限位开关请选用+24V、最小电流大于 6mA 的常开或常闭型开关；限位开关的类型

可通过 SW2 开关设定，详见 3.1.2 开关配置。当使用常开型限位开关时，应选择 EL+/EL-信号为高电平有效（即为 OFF）；当使用常闭型限位开关时，应选择 EL+/EL-信号为低电平有效（即为 ON）。出厂默认全部为 OFF，即为常开型模式；

2.3.5 通用数字输入/输出信号接口

DMC1000S 运动控制卡提供了大量的通用数字 I/O 接口。DMC1000S 有 67 路通用 I/O 口，40 入 27 出，其中 IN1～IN16 、IN33～IN40 以及 OUT1～OUT12 为光电隔离，IN17～IN32 以及 OUT13～OUT27 为非光电隔离接口。

2.3.5.1 INPUT 通用数字输入信号接口

通用数字输入信号 INPUT 用于接近开关、光电开关、按键等传感器的信号输入。位于插座 X1 上的通用输入口都是光电隔离的，对外接口 SCSI68 具体信号引脚分配见 8.1.1 接口 X1 引脚定义。DMC1000S 中增加的通用输入口位于扩展接口 J1 上，均为非光电隔离，但采用专用的接线板后，可实现 J1 接口的通用输入口全部光电隔离。对外接口 DB37 具体信号引脚分配见 8.1.3 接口 J1 引脚定义。通用数字输入信号电路原理图如图 2-15 所示：

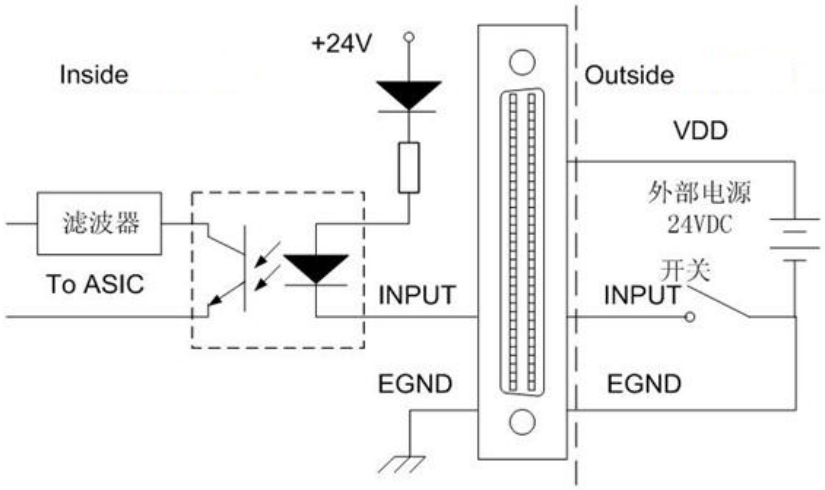


图 2-15 通用输入信号接口原理图

2.3.5.2 OUTPUT 通用数字输出信号接口

通用数字输出信号 OUT 用于控制继电器、指示灯等开关器件。位于插座 X1 上的通用输出口都是光电隔离的，对外接口 SCSI68 具体信号引脚分配见 8.1.1 接口 X1 引脚定义。DMC1000S 中增加的通用输出口位于扩展接口 J1 上，均为非光电隔离，但采

用专用的接线板后，可实现 J1 接口的通用输出口全部光电隔离。对外接口 DB37 具体信号引脚分配见 8.1.3 接口J1 引脚定义。通用数字输出信号电路原理图如图 2-16 所示：

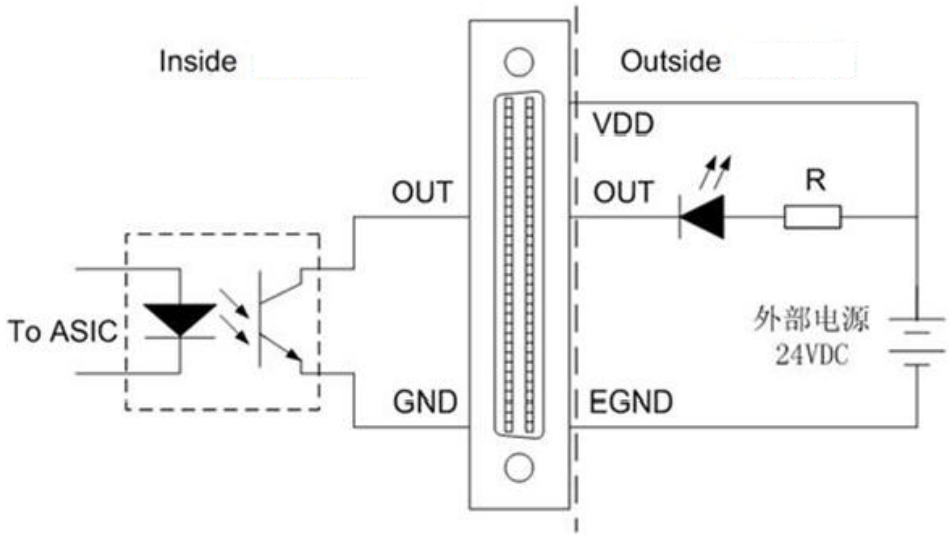
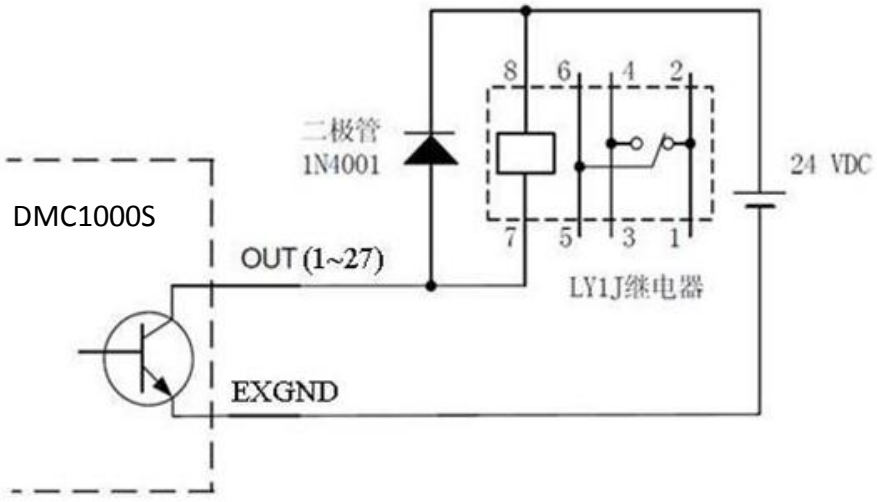


图 2-16 通用输出口电路原理图

注：通用数字输出端口的最大驱动电流为 100mA，严禁使用数字输出口直接驱动电磁阀等大电流、大功率元器件！当需要驱动小型继电器等感性负载时，线圈外必须并联一个续流二极管，以保护控制卡输出端口驱动元件，防止在关闭继电器的瞬间，被线圈产生的感应电动势击穿。如图 2-17 为 DMC1000S 卡与 OMRON 中间继电器（LY1J 24VDC）接线示意图：



2-17

DMC1000S 卡与中间继电器接线原理图

另外，当启动 PC 机时，DMC1000S 卡输出口的初始状态可以在拨码开关 SW1 上面进行设置，见 3.1.2 开关配置。

2.3.6 多卡运行

DMC1000S 运动控制卡最多支持 8 张卡同时工作。因此，一台 PC 机可以同时控制多达 32 轴步进/伺服电机同时工作。

DMC1000S 卡支持即插即用，用户可不必关心如何设置卡的基地址和 IRQ 中断值，当系统启动时这些全被系统 BIOS 自动设定。这里值得注意的是：在多卡运行时，卡的排列有可能从左往右依次为 1，2，3„，也有可能是从右往左依次为 1，2，3„。这与具体的 PC 机有关。表 2-1 为卡号与轴号的对照表：

表 2-1 多卡运行时卡号与轴号对照表

	第 1 轴	第 2 轴	第 3 轴	第 4 轴
卡 1	0	1	2	3
卡 2	4	5	6	7
卡 3	8	9	10	11
卡 n	（n-1）*4	（n-1）*4+1	（n-1）*+2	（n-1）*4+3

2.4 运动控制平台位置传感器及控制信号布局示例

DMC1000S 运动控制卡每轴都配有 2 个限位信号和 1 个原点信号输入接口。每路信号都进行了光电隔离，以减少外界对控制卡内部的干扰，保证运动控制的可靠性。

图 2-19 为一般运动平台位置传感器及控制信号的布置图。

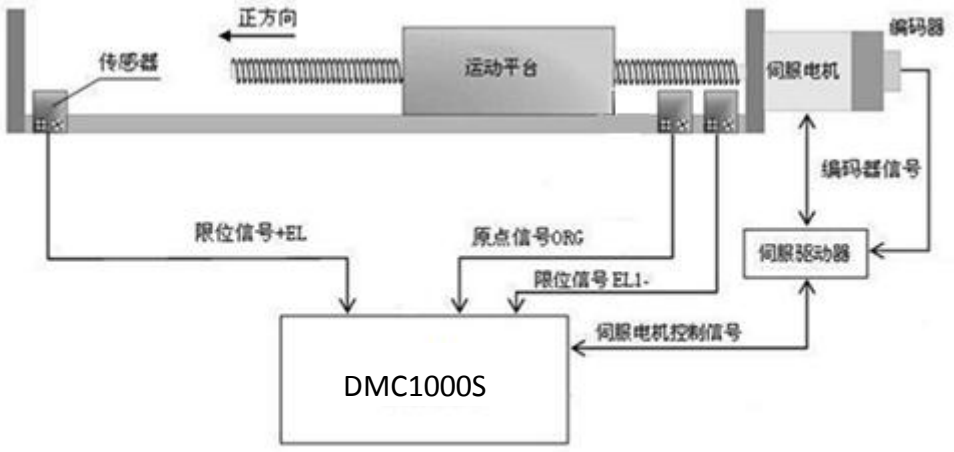


图 2-19 运动平台位置传感器及控制信号的布置图

DMC1000S 支持表 2-2 所示的回零模式，回零过程中支持遇限位自动反找功能（默认不使能），使用时需通过 d1000_set_home_el_return 函数使能自动反找功能。

回零模式	回零模式描述
0	一次回零
1	一次回零 + 反找原点
2	二次回零
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

表 2-2 DMC1000S 支持的回零模式

2.5 步进电机驱动器接线示例

2.5.1 单端输出接法

DMC1000S 卡第 1 轴与雷赛公司的 M415B 步进电机驱动器的接线方法如图 2-20 所示：

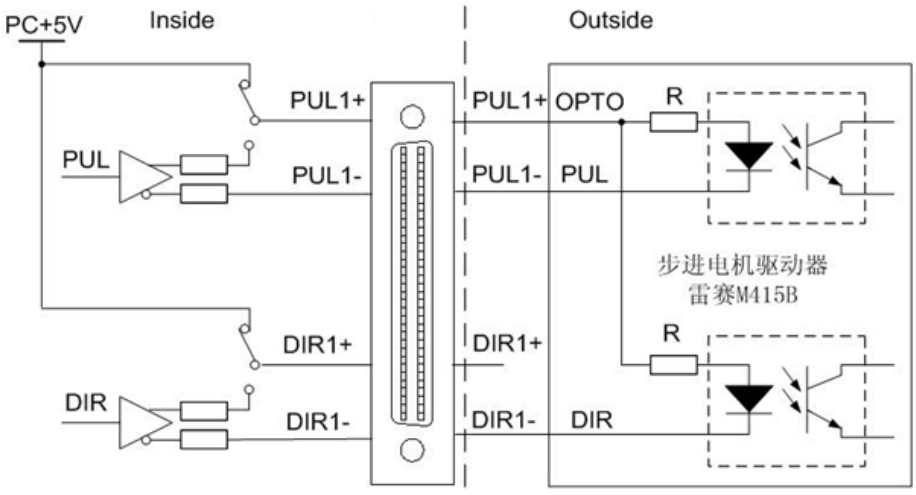


图 2-20 单端输出接线图

2.5.2 差分输出接法

DMC1000S 卡第 1 轴与雷赛公司 MD556 步进电机驱动器的接线方法如图 2-21 所示：

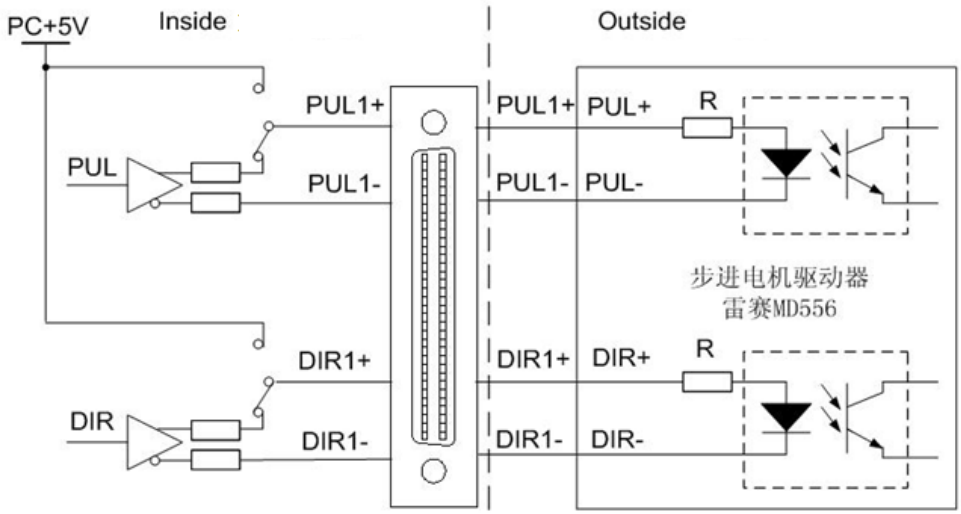


图 2-21 差分输出接线图

第 3 章 硬件配置与安装

3.1 硬件配置

DMC1000S 卡支持即插即用功能，其 I/O 地址的选择由系统 BIOS 自动指定，跟所有的支持即插即用的 PCI 卡（如声卡，Modem 卡，网卡）一样，系统 BIOS 均可为其自由分配一个工作地址，您也可在系统 BIOS 中手工设置。

DMC1000S 卡上有多组跳线开关 SW4~SW11 和拨码开关 SW1-SW2，分别用于设置 DMC1000S 卡的工作方式和参数。

3.1.1 跳线配置

跳线开关用于设置指令脉冲输出方式为差分输出或单端输出。DMC1000S 卡中表示为 SW4～SW11，设置方式如图 3-1 和图 3-2 所示；表 3-1 为脉冲和方向信号的跳线表。

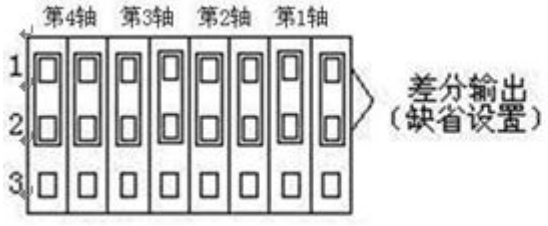


图 3-1 差分输出跳线设置

图 3-2 单端输出跳线设置

表 3-1 脉冲和方向信号跳线表

针脚号	针脚名	差分输出时 1 脚和 2 脚短路	单端输出时 2 脚和 3 脚短路
2	PUL0+	SW5	SW5
4	DIR0+	SW4	SW4
6	PUL0+	SW7	SW7
8	DIR0+	SW6	SW6
35	PUL0+	SW9	SW9
37	DIR0+	SW8	SW8
39	PUL0+	SW11	SW11
41	DIR0+	SW10	SW10
注：出厂时的缺省设置为 SW4～SW11 全为 1 脚和 2 脚短路，即差分输出方式。			

3.1.2 开关配置

DMC1000S 中配置了两个拨码开关 SW1 和 SW2，其中 SW2 用于设置 EL 限位模式，SW1 用来设定输出口的上电初始电平。

3.1.2.1 拨码开关 SW2

如图 3-3 所示，拨码开关 SW2.1~ SW2.4 用来设 EL 限位模式为常开型还是常闭型。EL 限位的缺省设定全部为 OFF，即各轴均为常开限位模式。表 3-2 为 SW2 拨码开关状态表，ON 代表常闭限位模式，OFF 代表常开限位模式。

图 3-3 拨码开关 SW2

表 3-2 SW2 拨码开关状态表

开关号	轴号	常开模式	常闭模式
SW2.1	0	OFF（出厂设置）	ON
SW2.2	1	OFF（出厂设置）	ON
SW2.3	2	OFF（出厂设置）	ON
SW2.4	3	OFF（出厂设置）	ON
注：SW2 开关的第 1 位为最左端一位。			

3.1.2.2 拨码开关 SW1

如图 3-4 所示，拨码开关 SW1.1~ SW1.4，其用来设定输出口的上电初始电平。对应该关系如表 3-3 如示：

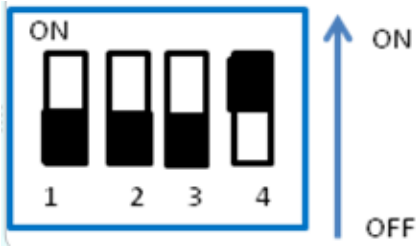


图 3-4 拨码开关 SW1

表 3-3 输出口拨码开关表

开关号	输出口号	常开模式	常闭模式
SW1.1	OUT21—OUT27	OFF（出厂设置）	ON
SW1.2	OUT13—OUT20	OFF（出厂设置）	ON
SW1.3	OUT5—OUT12	OFF（出厂设置）	ON
SW1.4	OUT1—OUT4	OFF	ON（出厂设置）
注：SW1 开关的第 1 位为最左端一位。			

3.2 硬件安装

DMC1000S 运动控制卡硬件结构遵从 32bit PCI 卡的结构标准，其安装方法类同于普通 32bit PCI 卡的安装，具体参考步骤如下：

1）打开 DMC1000S 的包装，参考 3.1 硬件配置 的说明，按照实际应用的需求，完成硬件配置；

2）使用辅助接口的用户，请将辅助接口与 DMC1000S 对应的插座连接，并确保连接牢固，可靠；

3）触摸地线，完全释放操作员身上的静电，带好防静电手套；

4）关闭 PC 机以及一切与 PC 相连的设备；

5）打开 PC 机的机箱；

6）选择一个靠近处理器的 32bit PCI 插槽，将 DMC1000S 垂直插入插槽中；

7）将 DMC1000S 用螺钉紧固在 PC 机机箱上，确保紧固、可靠 。

第 4 章 软件系统概述

DMC1000S 运动控制卡软件系统包括：驱动程序、运动控制函数库、演示程序、例子程序。

4.1 硬件驱动程序

DMC1000S 配套提供 Win7/Win10 等操作系统环境下的驱动程序。客户可以根据自己的需要选择相应的系统平台来开发适合自己的应用软件。硬件驱动程序具体安装方法请参考：5 驱动程序安装。

4.2 运动控制函数库

为了方便用户在 DMC1000S 卡上快速开发出功能强大的应用软件，雷赛公司采用科学的函数命名体系，为用户提供一套功能齐全，易学易用的函数库。用户只需根据自己的实际需要调用其中的一部分函数即可开发出各种用途的强大软件。

注：各函数具体功能介绍参考附录：8.2.2 函数说明。

4.2.1初始化、关闭运动控制卡

在操作 DMC1000S 运动控制卡之前，必须调用控制卡初始化函数为运动控制卡分配资源。同样，当结束对运动控制卡的操作时，必须调用控制卡关闭函数释放运动控制卡所占用的系统资源，使得所占资源可被其它设备使用。具体相关函数和功能如表 4-1 所示：

表 4-1 初始、关闭控制卡函数说明

	名称	功能	参考
1	d1000_board_init	初始化卡并分配系统资源	8.2.2.1
2	d1000_board_close	关闭卡并释放系统资源	8.2.2.1
注意：程序结束时，必须调用 d1000_board_close()函数释放系统资源。			

例程：初始化和关闭控制卡 (以标准 C 语言为例说明，下同)

```
CardCount = d1000_board_init();
if(CardCount == 0)
    {
        printf("\n 没有发现运动控制卡");
        getch();
        return();
    }

”
```

```
d1000_board_close();
””
```

4.2.2 设置脉冲输出模式

DMC1000S 采用脉冲指令控制步进/伺服电机。虽然脉冲指令是一个十分容易理解的概念，但由于市面上众多驱动器厂家对信号接口的要求各有不同，所以在使用控制卡控制不同型号的驱动器时，必须对脉冲信号输出方式进行正确设定，系统才能正常工作。相关函数和功能如表 4-2 所示：

表 4-2 脉冲设置函数说明

	名称	功能	参考
1	d1000_set_pls_outmode	设置指定轴的脉冲输出模式	8.2.2.2
注意：在调用运动控制函数之前应先调用该函数来设置指令脉冲模式。			

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

表 4-3 脉冲指令模式表

模式	PULn-脚输出	DIRn-脚输出
方向脉冲（pulse/dir）	脉冲信号	方向信号（电平）
双脉冲（CW/CCW）	正向（CW）脉冲	反向（CCW）脉冲
注：具体设置请参考 d1000_set_pls_outmode 函数具体说明 8.2.2.2 。		

4.2.2.1 脉冲/方向模式

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

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

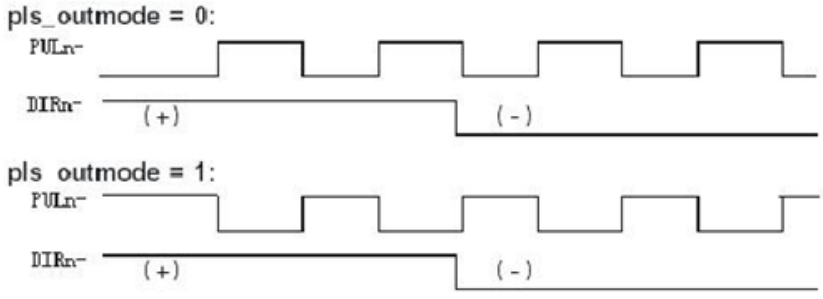


图 4-1 脉冲/方向信号图

4.2.2.2 双脉冲模式

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

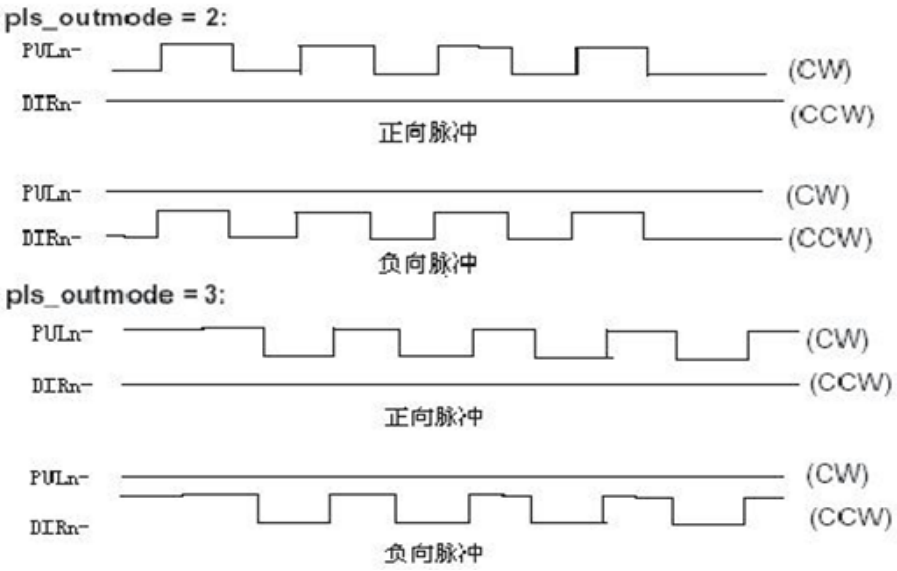


图 4-2　双脉冲信号图

例程（C 代码）：

```
""
d1000_set_pls_outmode(0,0);  // 表示设置第 0 轴脉冲输出模式为方向脉冲模式，PULn-
                             信号上升沿有效。
""
```

4.2.3 单轴位置和速度控制

DMC1000S 在表述运动轨迹时可以用绝对坐标和相对坐标这两种模式，如图 4-3 所示。

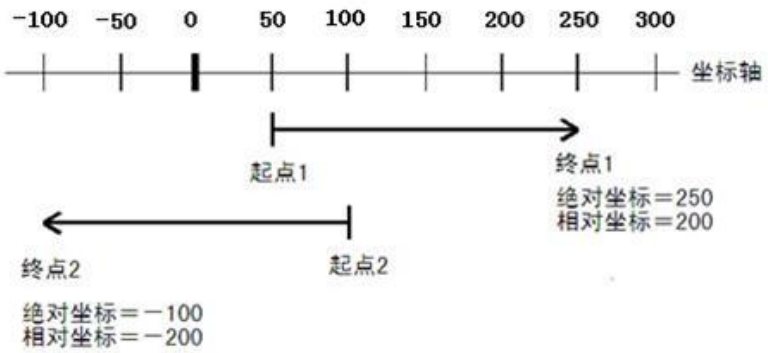


图 4-3　绝对坐标与相对坐标中轨迹终点的不同表达方式

这两种模式各有优点，如：在绝对坐标模式中用一系列坐标点定义一条曲线，如果要修改中间某点坐标时，不会影响后续点的坐标；在相对坐标模式中，用一系列坐标点定义一条曲线，用循环命令可以重复这条曲线轨迹多次。

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。

4.2.3.1 梯形速度曲线运动模式

通常位置控制采用这种速度控制模式。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 4-4 所示。

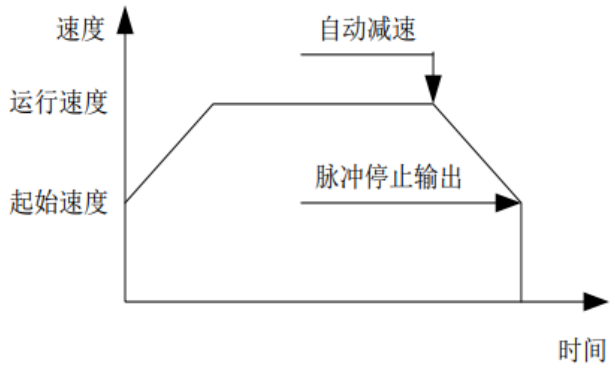


图 4-4 梯形速度曲线

运动速度之所以要按梯形曲线变化，是因为：电机转子和被拖动的物体具有惯性，不可能在瞬间达到指定速度，因此应该有一定的加速过程。减速时亦是类似，否则电机会因为瞬间力矩不足而出现丢步、过冲（步进电机系统）或振荡（伺服电机系统）现象。

实现梯形速度曲线控制的点位运动函数如表 4-4 所示：

表 4-4 梯形点位控制相关函数说明

	名称	功能	参考
1	d1000_start_t_move	以梯形速度曲线控制相对坐标的点位运动	8.2.2.5
2	d1000_start_ta_move	以梯形速度曲线控制绝对坐标的点位运动	8.2.2.5
注意：加减速是对称的，所以加速时间也即减速时间。			

例程：执行以梯形速度曲线作点位运动

```
""
d1000_stat_t_move(0,50000,500,6000,0.1)；//设置 0 号轴相对坐标下运动距离为 50000 个脉冲、起始速度为 500 脉冲/秒、运行速度为 6000 脉冲/秒、加减速时间为 0.1 秒、以梯形速度曲线开始执行运动
""
```

4.2.3.2 S 形速度曲线运动模式

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。若将加速度改为线性变化，则速度曲线相应将变得光滑，如 2.3.2.2 速度控制 中的 B 部分 S 形速度曲线说明所示。加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。

设置 S 形速度曲线及其点位运动的函数如表 4-5 所示：

表 4-5 S 形速度控制相关函数说明

	名称	功能	参考
1	d1000_start_s_move	以 S 形速度曲线控制相对坐标的点位运动	8.2.2.5
2	d1000_start_sa_move	以 S 形速度曲线控制绝对坐标的点位运动	8.2.2.5

例程：执行以 S 形速度曲线作点位运动

””

d1000_stat_s_move(0,50000,500,6000,0.1); //设置 0 号轴相对坐标下运动距离为 50000 个脉冲、起始速度为 500 脉冲/秒、运行速度为 6000 脉冲/秒、加减速时间为 0.1 秒、以 S 形速度曲线开始执行运动

””

4.2.3.3 连续运动模式

DMC1000S 控制卡可以控制电机以梯形或 S 形速度曲线在指定加速时间内从初速度加速至运行速度，然后以该速度连续运行，直至调用停止指令或该轴遇到限位信号才会停止。连续运动的函数如表 4-6 所示：

表 4-6 连续运动相关函数说明

	名称	功能	参考
1	d1000_start_tv_move	以梯形速度曲线控制连续运动	8.2.2.4
2	d1000_start_sv_move	以 S 形速度曲线控制连续运动	8.2.2.4

例程：连续运动

””

d1000_start_sv_move(0,500,6000,0.1)；//设置 0 号轴起始速度为 500 脉冲/秒、运行速度 6000 脉冲/秒、加减速时间为 0.1 秒、以 S 形速度曲线开始连续运动。

””

4.2.3.4 加减速过程的距离（脉冲数）计算

对于梯形加速曲线运动，加、减速段的运动距离（脉冲数）都很容易计算：

Dacc = (1/2)×(MaxVel + StrVel)×Tacc

其中，Dacc：加速段距离

MaxVel：运行速度

StrVel：起始速度

Tacc：加速时间

由于 DMC1000S 加、减速时对称的，所以减速段距离 Ddec = Dacc。

4.2.4 多轴运动控制

DMC1000S 运动控制卡单张卡可以同时控制 4 个轴以多种方式同时运动，包括多轴联动和直线插补。

4.2.4.1 多轴联动

多个轴同时运动，一般称为多轴联动。

DMC1000S 卡可以控制多个电机同时执行 d1000_start_t_move 或 d1000_start_ta_move 等单轴运动函数。所谓同时执行，是在程序中顺序调用这些函数，因为程序执行速度很快，在瞬间几个电机都开始运动，给人的感觉就是“同时开始运动”。

多轴联动在各轴速度设置不当时，各轴停止时间不同、在起点与终点之间运动的轨迹也不是直线。如图 4-5 所示。

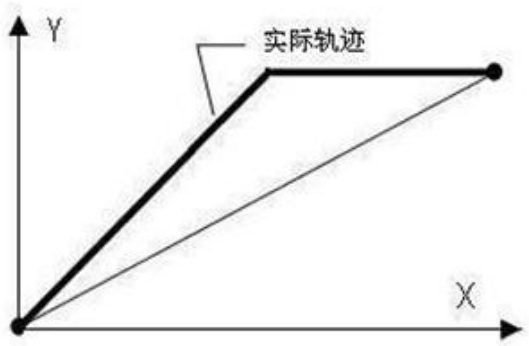


图 4-5 二轴联动示意图

4.2.4.2 直线插补运动

DMC1000S 控制卡支持 1 个插补系中设置任意 2 轴、3 轴以及 4 轴进行直线插

补，插补运动的计算由专用函数完成，用户只需调用相应的运动函数并设置运动速度、加速度、终点位置等参数，不需要介入插补过程的计算工作。

插补运动与多轴联动不同：插补运动不但能保证起点、终点位置准确，而且各轴的脉冲是按照直线斜率成比例发出的，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 4-6 所示。

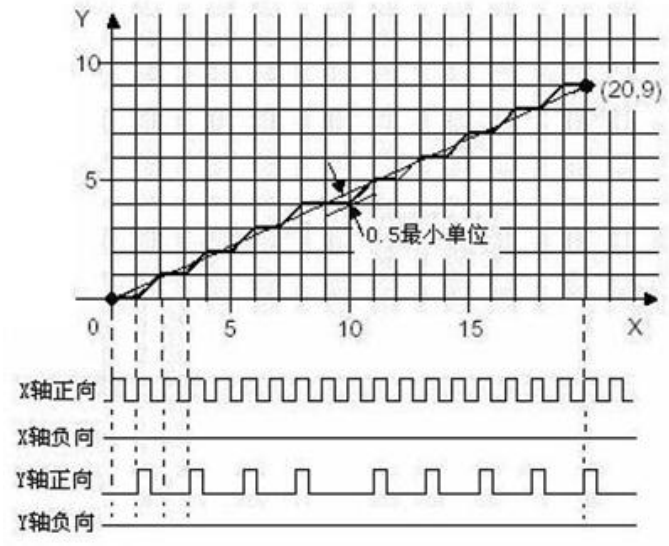


图 4-6 直线插补示意图

直线插补相关函数如表 4-7 所示：

表 4-7 直线插补运动相关函数说明

	名称	功能	参考
1	d1000_start_t_line	启动多轴相对坐标的直线插补	8.2.2.6
2	d1000_start_ta_line	启动多轴绝对坐标的直线插补	8.2.2.6

二轴直线插补：

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

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

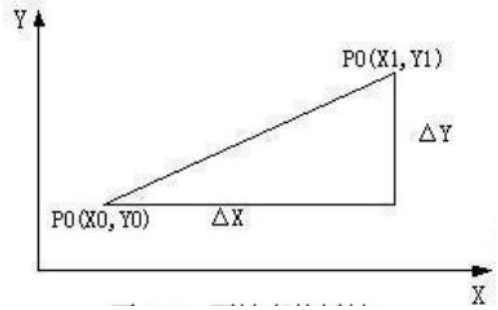


图 4-7 两轴直线插补

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

二轴直线插补例程（C 代码）：

```
""
Short AxisArray[2];
Short DistArray[2];
AxisArray[0]=0;// 轴 0、轴 1 直线差补
AxisArray[1]=1;
DistArray[0]=1000;      // ΔX＝1000 脉冲
DistArray[1]=2000;      // ΔY＝2000 脉冲
d1000_start_t_line (2, AxisArray, DistArray, 400,1000, 0.1); //起始速度为 400 pps，运行速度为 1000 pps，加速时间为 0.1s
""
```

三轴直线插补：

如图 4-8 示，XYZ 3 轴直线插补从 P0 点运动至 P1 点。插补过程中 3 轴的速度比为 X:Y:Z，与合成的矢量速度 P 的关系为：

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

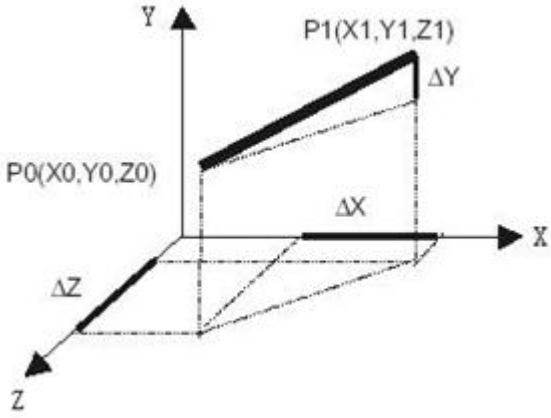


图 4-8 三轴直线插补

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

轴直线插补例程（C 代码）：

```
""
Short AxisArray[3];
Short DistArray[3];
AxisArray[0]=0;       //轴 0、轴 1、轴 2 直线差补
AxisArray[1]=1;
AxisArray[2]=2
DistArray[0]=1000;     //△X＝1000p
DistArray[1]=2000;     //△Y＝2000p
DistArray[2]=3000;     //△Z＝3000p
d1000_start_t_line (3, AxisArray, DistArray, 400,1000, 0.1); //起始速度为 400pps，运行速度为 1000pps，加速时间为 0.1s
"
```

4.2.5 回原点运动

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

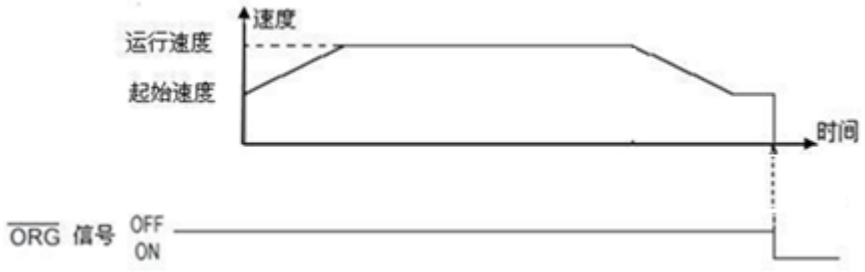


图 4-9 回原点过程

从图 4-9 可以看出，启动回原点运动后，设备向原点方向运动；当碰到原点信号有效便立即停止，该位置即为原点位置。当需要自动寻零时，即使回零中先遇到限位信号也能自动反相找原点信号，此时需要用户设置限位反找使能，这种方式方便用户使用，无需再编写反找程序。

具体相关函数如表 4-8 所示：

表 4-8 回原点运动相关函数说明

	名称	功能	参考
1	d1000_set_HOME_pin_logic	设置原点信号有效电平	8.2.2.7
2	d1000_config_home_mode	设置回零模式	8.2.2.7
3	d1000_set_home_el_return	设置限位反找(默认不使能反找)	8.2.2.7
4	d1000_home_move	启动指定轴进行回原点运动	8.2.2.7

回零支持的 6 种模式

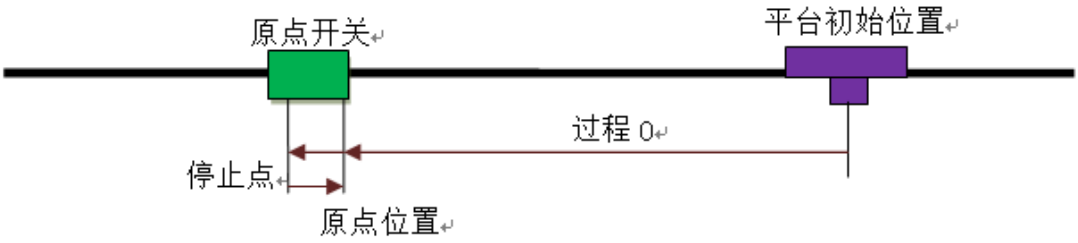
注意：回零完成，需要手动调用函数清除当前位置为零。回零反找的低速速度由起始速度指定，起始速度设置为 0 时，低速速度默认为 500pulse/s。

方式 0：一次 home 回零

该方式适合与行程短、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。

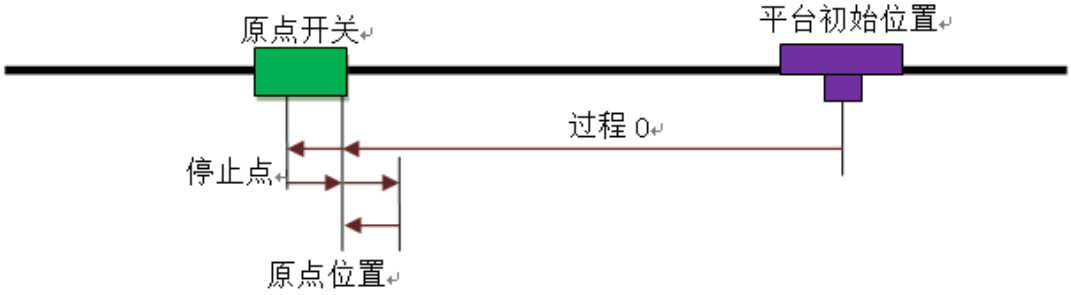
方式 1：一次 home+反找

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所示。



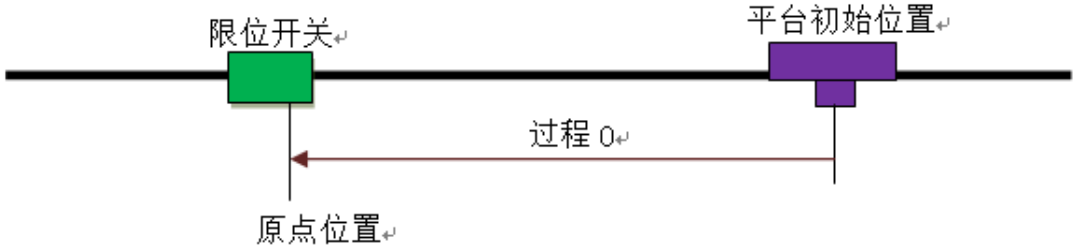
方式 2：两次 home 回零

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



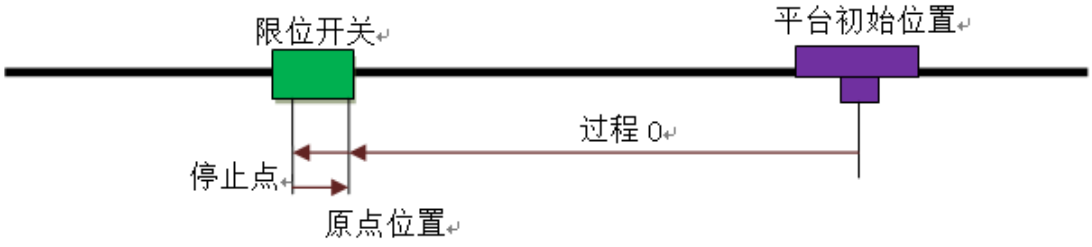
方式 10：一次限位回零

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，如图所示。



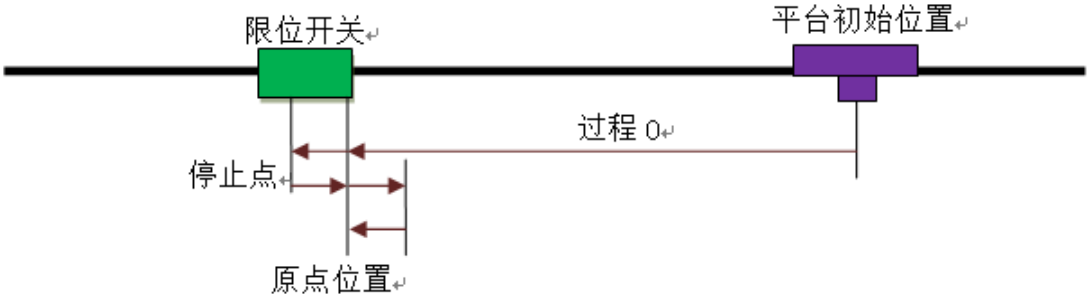
方式 11：一次限位回零加回找

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所示。



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。



例程：回原点运动

```
""
d1000_home_move (0,500,5000,0.1); //设置 0 号轴以起始速度为 500pps，运行速度为 5000pps，加速时间为 0.1s 正向回原点运动
while (d1000_check_done(0) = = 0)     //等待回原点动作完成
{
}
d1000_set_command_pos (0,0);    //将当前第 0 轴位置清零，即设为第 0 轴原点位置
""
```

4.2.6 指令脉冲计数

DMC1000S 每一轴都有一个 32 位的指令脉冲计数器，记录了当前对应轴的绝对指令脉冲位置值。可以通过调用 d1000_get_command_pos 函数来读取该计数器的值，也可以通过调用 d1000_set_command_pos 函数来设置该计数器的值。具体相关函数如表 4-9 所示：

表 4-9 指令脉冲计数相关函数说明

	名称	功能	参考
1	d1000_get_command_pos	读取指令位置计数器值	8.2.2.9
2	d1000_set_command_pos	设置指令位置计数器值	8.2.2.9

例程：位置操作

```
""
d1000_set_command_pos (0,100);          //设置轴 0 的脉冲位置为 100
position = d1000_get_command_pos (0);    //读取轴 0 的当前位置值至变量 position
""
```

4.2.7 通用 I/O 控制

DMC1000S 运动控制卡硬件上为用户提供了通用数字输入输出接口。用户可以使用这些数字 I/O 口用于检测开关信号、传感器信号等输入信号，或者输出继电器、指示灯等输出设备的控制信号。相关的函数如表 4-10 所示：

表 4-10 通用 IO 相关函数说明

	名称	功能	参考
1	d1000_out_bit	通用输出口输出	8.2.2.10
2	d1000_in_bit	读取通用输入口状态	8.2.2.10
3	d1000_get_outbit	读取通用输出口状态	8.2.2.10

例程：通用输入输出

```
””
If (d1000_in_bit (1)==0) //读取 INPUT1 输入口的状态，判断按键是否按下
{
    d1000_out_bit (1, 0); //如果按键按下，OUT1 口输出为低电平，LED 发光
}
else
{
    d1000_get_outbit(1, 1); //如果按键未按，OUT1 口输出为高电平，LED 不亮
}
””
```

4.3 演示程序

为协助用户更快掌握 DMC1000S 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 Motion。它具有多种运动控制功能和 I/O 测试功能。

将 DMC1000S 所配光盘放入光驱中，在相应的目录 motion 下，将其全部拷贝到计算机硬盘的任意指定位置后，运行 motion.exe，即可对控制卡的各项主要功能进行检测、学习。利用 Motion 软件，用户既可以快速地了解、熟悉 DMC1000S 运动控制卡的软硬件功能，又可以方便快捷地测试运动控制系统在执行各种动作时的性能及特性。详细资料可参考： 6. 演示软件及应用 。

4.4 例子程序

DMC1000S 运动控制卡为了方便用户利用 VC 或 VB 等编程工具对 DMC1000S 进

行应用开发，特针对典型的功能如：

单轴运动、回零运动、插补运动等，提供了示例源代码。

用户可以直接将配套的资料包中的代码作为程序模块，直接拷贝到您的程序工程中使用。

第 5 章 驱动程序安装

安装 DMC1000S 控制卡驱动程序的过程与安装其它即插即用卡（如 MODEM 卡，显卡等）的驱动程序十分类似，Win7 和 Win10 使用同一驱动程序，因此本手册只提供 Win7/Win10 操作系统环境下的安装示例。

5.1 在 Windows 7 /10 操作系统环境中安装步骤

1、关闭 PC 机并打开机箱，在 PCI 槽中插入 DMC1000S 运动控制卡，其过程和注意事项请认真参照 DMC1000S 硬件概述部分：3. 硬件配置与安装；

2、启动 PC 机，进入 Windows7 操作系统后，系统会提示发现新硬件，并在右下方的任务栏中出现如图 5-1 所示的信息提示，请先不要点击此信息提示，或直接将其关闭；

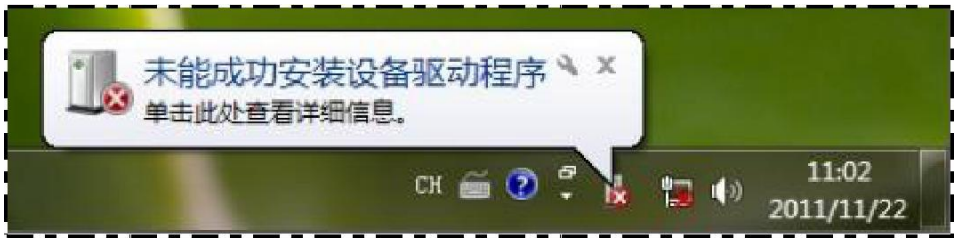
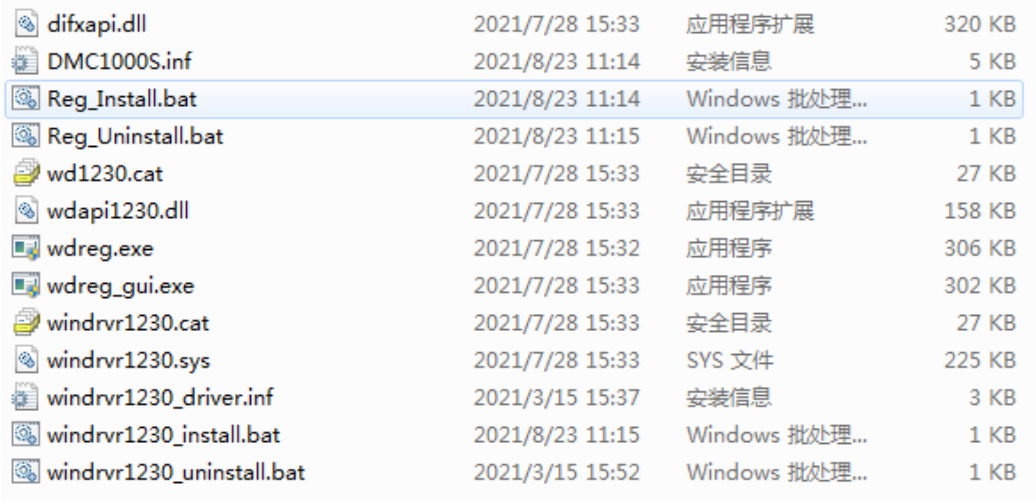


图 5-1 系统提示发现新硬件

3、打开 DMC1000S 驱动程序所在文件夹，在相应的目录下，打开“X64”资料夹，找到“Reg_Install.bat”批处理文件，如图 5-2 所示，双击鼠标左键，系统将弹出一个如图 5-3 所示的界面，开始自动注册 DMC1000S 卡的相关信息，并出现提示或警告信息，请选择“安装”，并等待安装结束。



difxapi.dll	2021/7/28 15:33	应用程序扩展	320 KB
DMC1000S.inf	2021/8/23 11:14	安装信息	5 KB
Reg_Install.bat	2021/8/23 11:14	Windows 批处理...	1 KB
Reg_Uninstall.bat	2021/8/23 11:15	Windows 批处理...	1 KB
wd1230.cat	2021/7/28 15:33	安全目录	27 KB
wdapi1230.dll	2021/7/28 15:33	应用程序扩展	158 KB
wdreg.exe	2021/7/28 15:32	应用程序	306 KB
wdreg_gui.exe	2021/7/28 15:33	应用程序	302 KB
windrvr1230.cat	2021/7/28 15:33	安全目录	27 KB
windrvr1230.sys	2021/7/28 15:33	SYS 文件	225 KB
windrvr1230_driver.inf	2021/3/15 15:37	安装信息	3 KB
windrvr1230_install.bat	2021/8/23 11:15	Windows 批处理...	1 KB
windrvr1230_uninstall.bat	2021/3/15 15:52	Windows 批处理...	1 KB

图 5-2 运行批处理文件

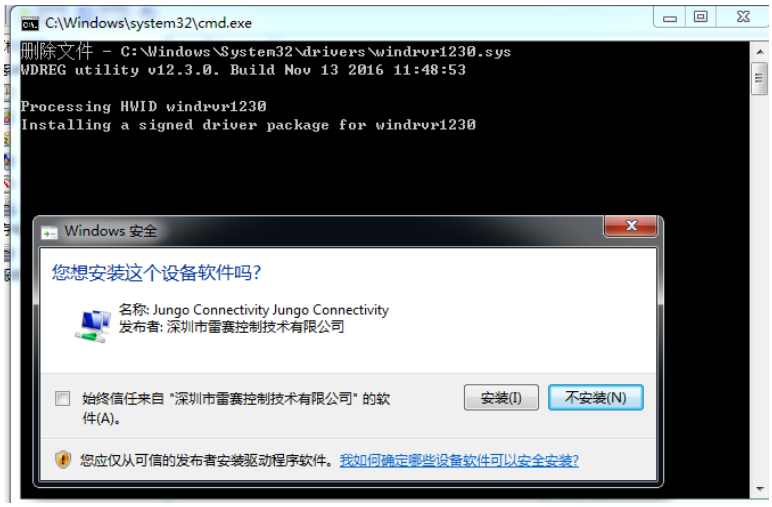


图 5-3　系统自动注册 DMC1000S 卡相关信息及安装提示

4、鼠标右键单击“计算机”—> 选择“管理(G)” —> 选择“设备管理器”，在“设备管理器”中看到如图 5-4 所示的驱动信息时，DMC1000S 控制卡驱动安装成功。

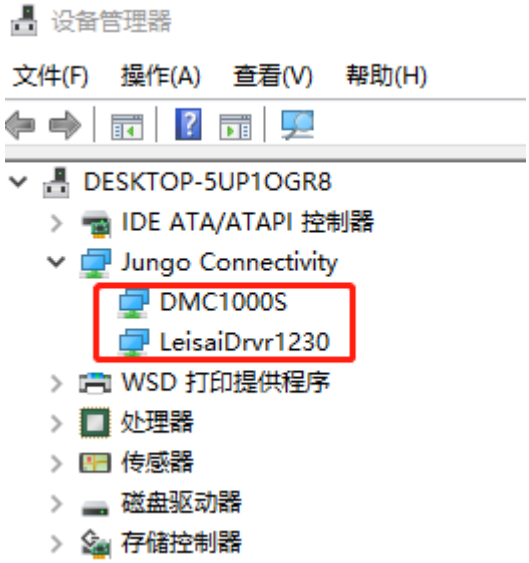


图 5-4 驱动安装成功

第 6 章 演示软件及应用

雷赛公司为了方便用户尽快熟悉 DMC1000S 控制卡运动功能和相关函数而配套提供的一个演示软件 Motion1000S。利用这个软件，用户既可以很快地熟悉 DMC 控制卡的软硬件功能，又可以方便快捷地测试电机驱动电路和系统的运动性能。

当您将 DMC1000S 运动控制卡的软硬件正确地安装到 PC 机上后，即可运行 Motion 软件，软件启动界面如图 6-1 所示。

图 6-1 软件启动界面

Motion 启动界面上有三个操作按钮，其功能如下：

运动操作：单击该按钮，进入运动控制操作界面，可以进行各种运动的演示，如：定长运动、回原点运动、直线插补运动等；

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

退出：停止当前所有对软件进行的操作，退出 Motion 演示软件。

6.1 I/O 检测演示

进入 IO 检测界面，如图 6-2 所示，可以查看运动控制卡各轴的专用输入信号状态、通用输入信号状态和通用输出信号状态，并可控制每个通用输出信号的电平。

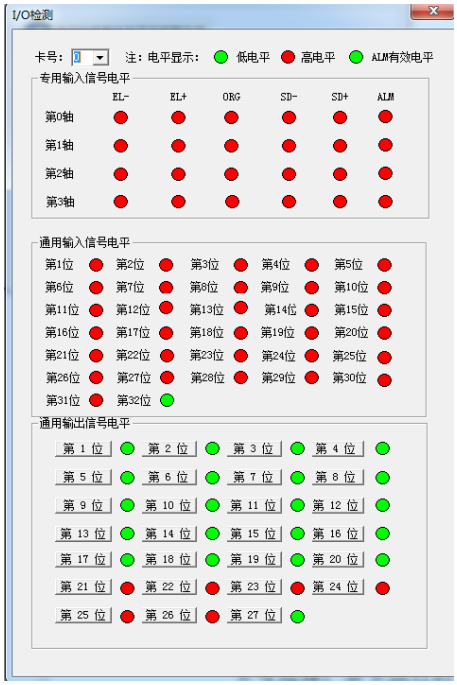


图 6-2　I/O 检测界面

卡号选择：多卡运行时，用于选择要测试的卡号；专用输入信号电平：检测每个轴的专用信号状态；通用输入信号电平：检测通用输入信号的电平状态；

通用输出信号控制和电平：控制通用输出信号，并显示当前各输出口的电平状态。通过单击显示为“第 X 位”的按钮，可以改变相应位的电平状态。

绿色表示该信号的电平为低电平，红色表示该信号的电平为高电平。

6.2 运动操作演示

进入运动控制操作界面，如图 6-3 所示，可以选择希望演示的运动功能，并对该运动的各种参数进行设置。

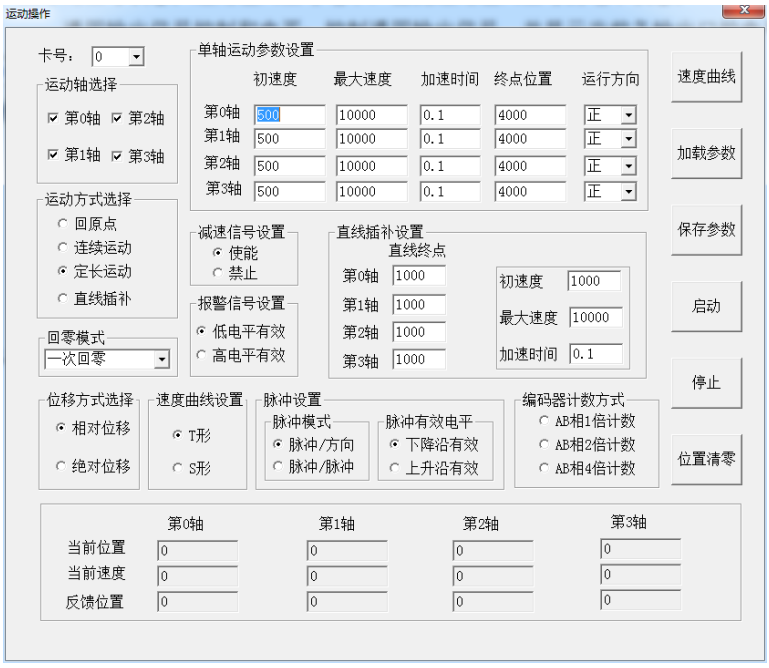


图 6-3 MOTION1000S 运动控制操作界面

界面各项介绍如下：

卡号选择：多卡运行时，选择控制卡卡号；运动轴选择：选择运动控制的轴号；

单轴运动参数设置：对回原点、连续运动、定长运动各轴的运动参数进行设置；

运动方式选择：选择要演示的运动类型；

直线插补参数设置：对插补运动的参数进行设置；位移方式选择：选择该次运动的位移方式；

加速曲线选择：选择该次运动的加速度的曲线；

脉冲设置：可以根据驱动器的具体要求设置脉冲输出类型；减速信号设置：设置减速信号是否有效。

各按钮功能如下：

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

保存参数：将当前界面中设置的参数信息保存到软件中；

启 动：按照当前的设置进行运动；

停 止：停止当前所有轴的运动；

位置清零：将“当前位置”中显示的各轴的位置设置为零。

第 7 章 用户系统开发

如果您对 C、C++、Visual Basic 等程序语言一点都不了解的话，我们建议您先花几天时间去阅读至少一本该语言的培训教材，并且通过练习掌握该语言的基本技巧，例如如何编写简单的程序，如何创建窗体和调用函数等。

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，那么可以浏览索引中的函数，并找到所需要的函数描述页码，跳转到“附录中运动函数库”查阅所需要的相关函数信息。

7.1 基于 windows 平台的应用软件结构

使用雷泰控制卡的机器控制系统构架如图 7-1 所示：

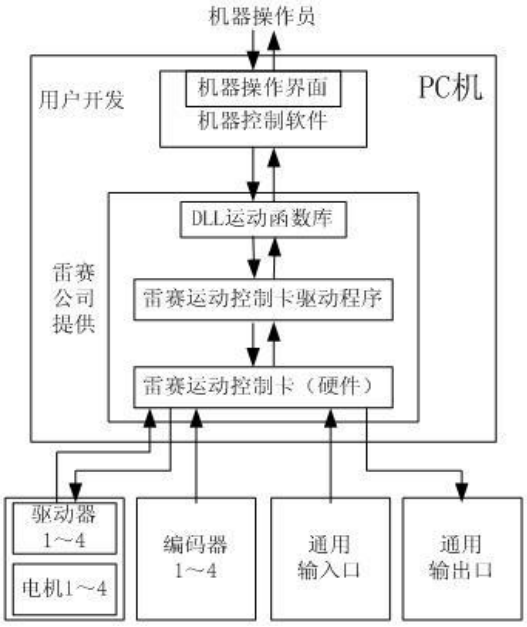


图 7-1 基于雷泰控制卡的机器控制系统构架

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）传递给机器控制软件；

2）机器控制软件将操作信息转化为运动参数并根据这些参数调用 DLL 库中运动函数；

3）运动函数调用雷泰运动控制卡驱动程序发出控制指令给控制卡；

4）雷泰运动控制卡再根据控制指令发出相应的控制信号(脉冲、方向信号)给电机驱动器；

5）电机驱动器根据控制信号来驱动电机运动；

6）电机的运动通过机械传动机构转化为机器的动作。

用户在开发应用软件（即机器控制软件）的过程中所需要做的就是针对上面

所说的第 1 ）步和第 2 ）步进行编程。雷赛公司已提供支持 DMC1000S 运动控制卡的硬件驱动程序和 DLL 运动函数库。这些函数提供了所有与运动控制相关的功能，使用极为方便。用户不需要更多了解硬件电路的细节以及运动和插补的计算细节，就能够使用 C、C++、Visual Basic 等程序语言调用这些函数来快速开发出自己的应用软件。

用户编写的机器控制软件典型流程如图 7-2 所示：

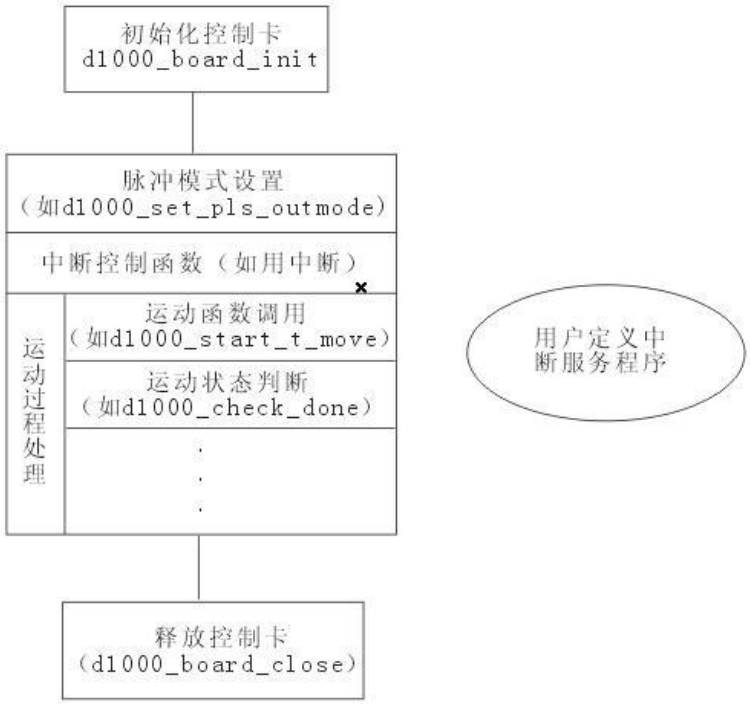


图 7-2 控制软件的典型流程

7.2Visual Basic6.0 环境下编程

确保 DMC1000S 运动控制卡已经插入到您的计算机插槽中，并安装好驱动程序、Motion1000S 测试软件和 VB，在调用 DMC1000S 运动函数之前，请做好下面几件事情：

1）启动 Motion 测试软件，进行所需要功能的简单测试。

如：单轴定长运动，以确定 DMC1000S 运动控制设备已经正常工作；

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 DMC1000.bas 和 DMC1000.dll 文件拷贝到该目录下；

4）运行 VB，并建立一个工程，然后保存此新建的工程到 vbMotion 目录中；

5）将运动函数库链接到您的工程项目中：

A、在 VB 编译器的“工程（P）”菜单中选择“添加模块”；

B、选择“现存”；

C、选择“DMC1000.bas”；

D、选择“确定”

6）运动函数的调用：

当您将运动函数链接到您的工程中后，就可以像调用其他 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 8.2.2 函数说明，当然还可以打开模块文件 DMC1000S.bas 了解每个函数的具体定义。

在编程过程中，您可以参阅我们提供的运动函数编程示例。我们提供的 VB 编程示例源代码，存放在对应的资料包中，只要您将控制卡及其驱动程序安装好，并安装了 VB 编译器，即可直接运行这些源代码。

7.3Visual C++6.0 环境下编程

确保 DMC1000S 运动控制卡已经插入到您的计算机插槽中，安装好驱动程序、Motion1000S 演示软件和 VC，在调用 DMC1000S 运动函数之前，请做好下面几件事情：

1）启动 Motion1000S 测试软件，进行所需要功能的简单测试。

如：单轴定长运动，以确定 DMC1000S 运动控制设备已经正常工作；

2）运行 VC，并建立一工程，将工程命名为 vcMotion（注：此工程名可以自己指定），路径为 d:\；

3）将 DMC1000.lib、DMC1000.h 和 DMC1000.dll 文件拷贝到该目录下(这两个文件在配套光盘 Module 目录下)；

4）将运动函数链接到您的工程中：将 DMC1000.lib 加入到工程中，然后在调用运动函数的文件头部代码加入#include“DMC1000.h”语句；

5）运动函数的调用：将运动函数链接到工程中后，您就可以像调用其它 Windows API 函数一样，调用运动函数。每个函数的具体功能，请参考 8.2.2 函数说明，当然，还可以打开头文件 DMC1000.h 了解每个函数的具体语法定义。

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

7.4 编程举例

本节详细介绍如何编写一个简单的程序：点位运动控制。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 DMC1000S 运动控制卡的各种运动函数来实现运动控制功能。

在动手编写控制程序之前，请确定：

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

2）已经按照前述章节中相关步骤安装了控制卡，用 Motion1000S 测试软件对硬件进行简单测试，以确认设备已经正常工作。

如果您希望更深入了解 DMC1000S 运动控制卡的编程方法，可以参阅资料包中的例程源代码。

7.4.1Visual 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：test，如图 7-3 所示。

图 7-3 建立工程

6）按“确定”按钮，在应用程序类型中选择“基本对话”，按“确定”，建立工程。

7）修改对话框内容，增加按钮“启动”（命名 IDC_BUTTON_Start）和“停止”（命名为 IDC_BUTTON_Stop），如图 7-4 所示。

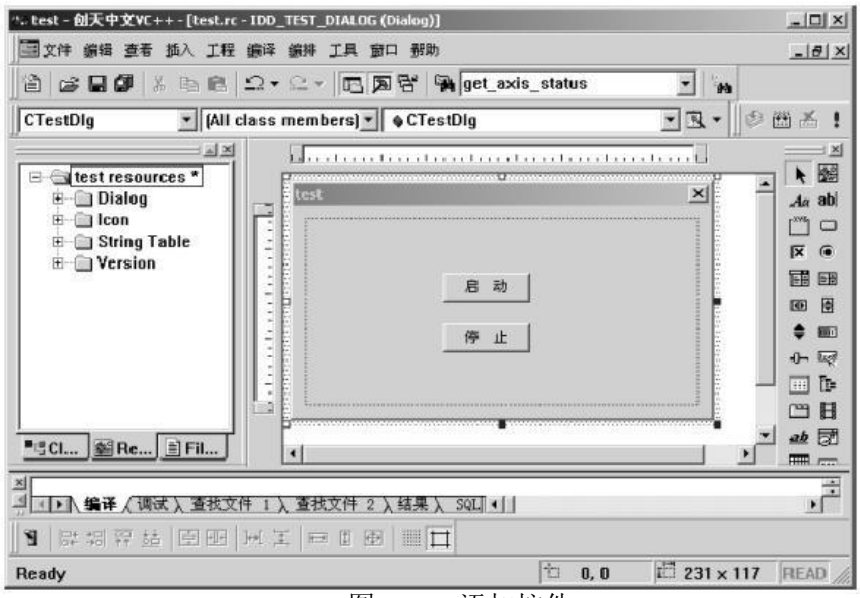


图 7-4 添加控件

8） 在 MotionDMC1000S 安装目录下找到 DMC1000.h 和 DMC1000.lib 文件拷贝到 E:\test 目录下。

9） 选择“工程”->“添加工程”->“文件”，选中 DMC1000.lib 文件加入到工程中。

10）打开 testDlg.cpp 文件，在头部添加语句：#include “DMC1000.h”，如 7-5 所示。

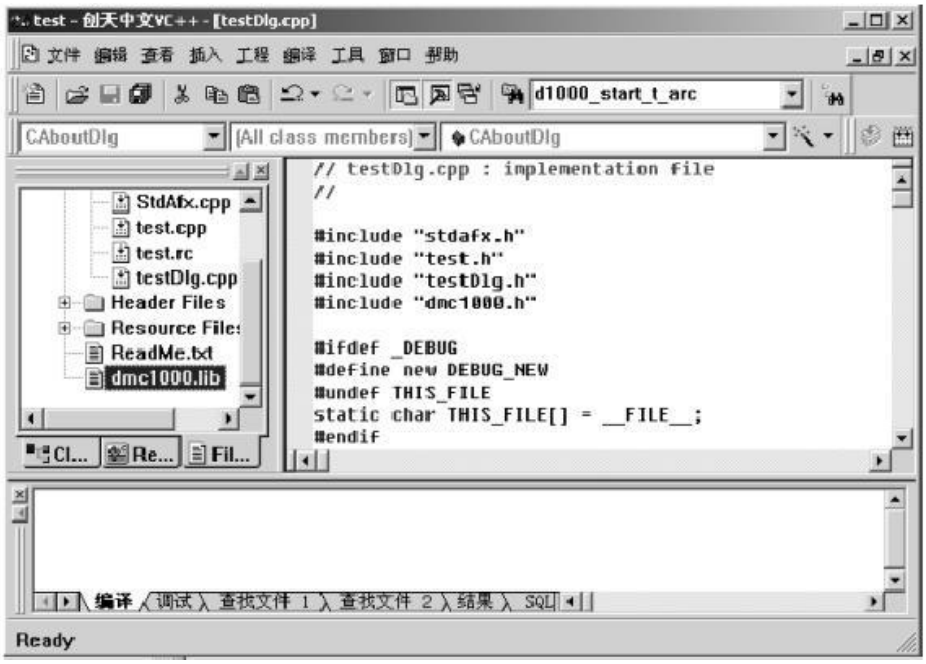


图 7-5 添加头文件

11）在 CTestDlg::OnInitDialog()函数中添加代码：d1000_board_init()；如图 7-6 所示。

图 7-6 添加初始化函数

12）在 CTestDlg 中添加一个成员函数 OnCacel,在该函数中添加代码：d1000_board_close()；CDialog::OnCancel()；如图 7-7 所示。

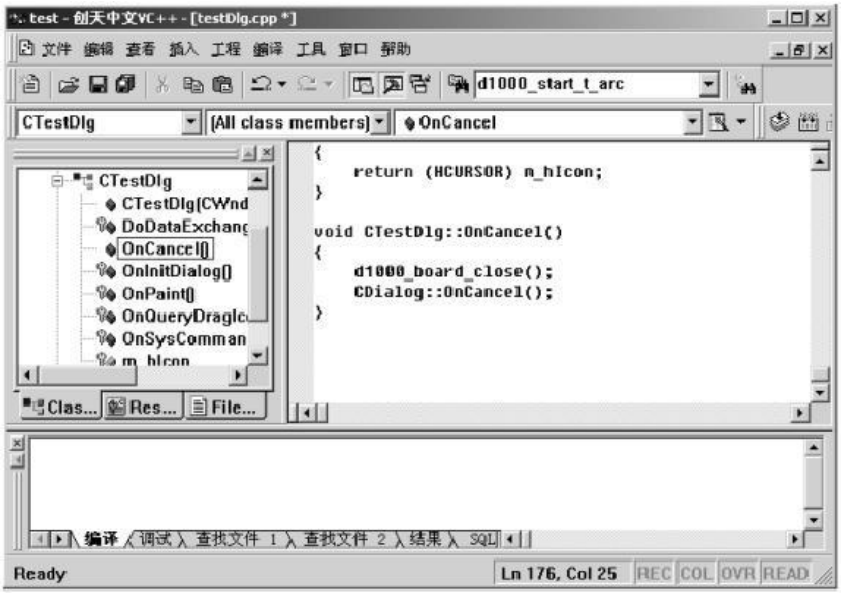


图 7-7 添加关闭控制卡函数

13）双击“启动”按钮在按钮点击事件中输入代码：

d1000_start_t_move(0,1000,400,1000,0.1)；

双击“停止”按钮在按钮点击时间中输入代码：d1000_decel_stop(0)；如图 7-8 所示。

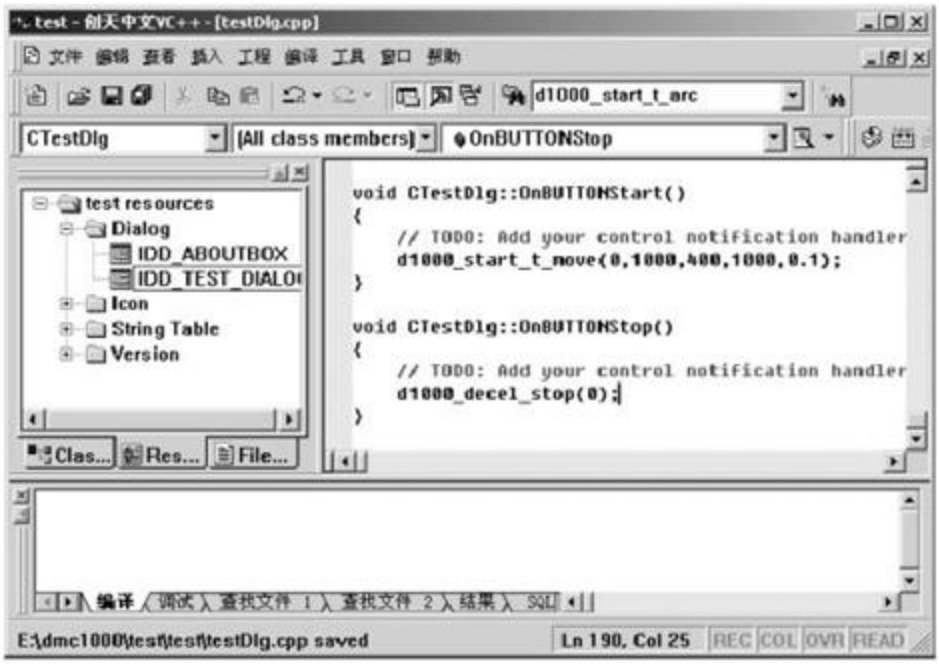


图 7-8 添加运动控制函数

14）编译，运行，按下如图 7-9 所示界面中的“启动”按钮，第 0 轴就会输出 1000 个脉冲。运动中可以按下“停止”按钮，即可减速停止脉冲输出。

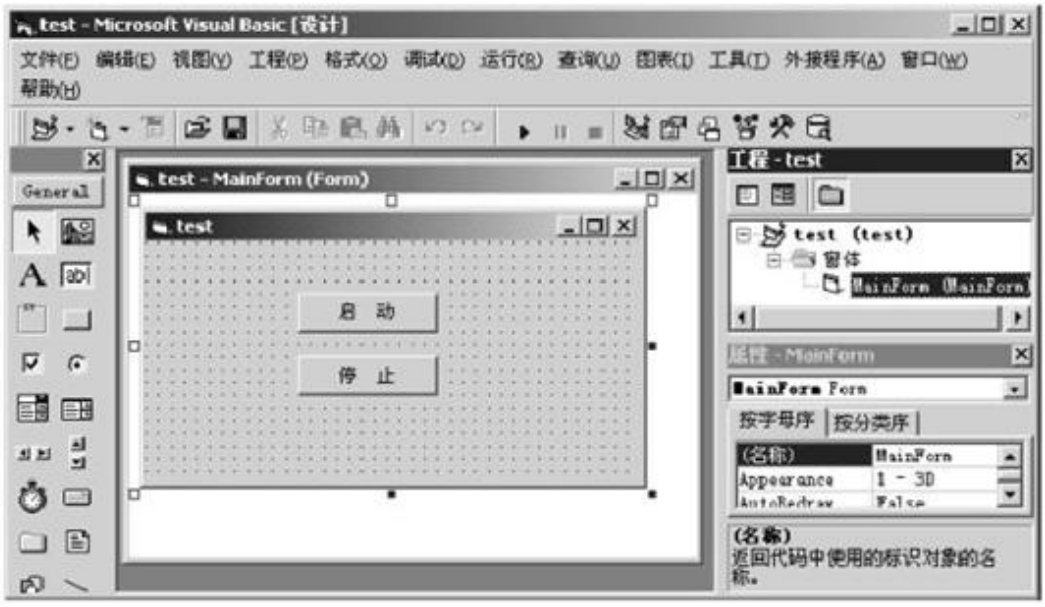
图 7-9 运行程序

7.4.2Visual Basic6.0 编程举例

1）在磁盘上新建一个目录，如 E:\test；

2）打开 Visual Basic 6.0；

3）新建一个工程，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 7-10 所示。

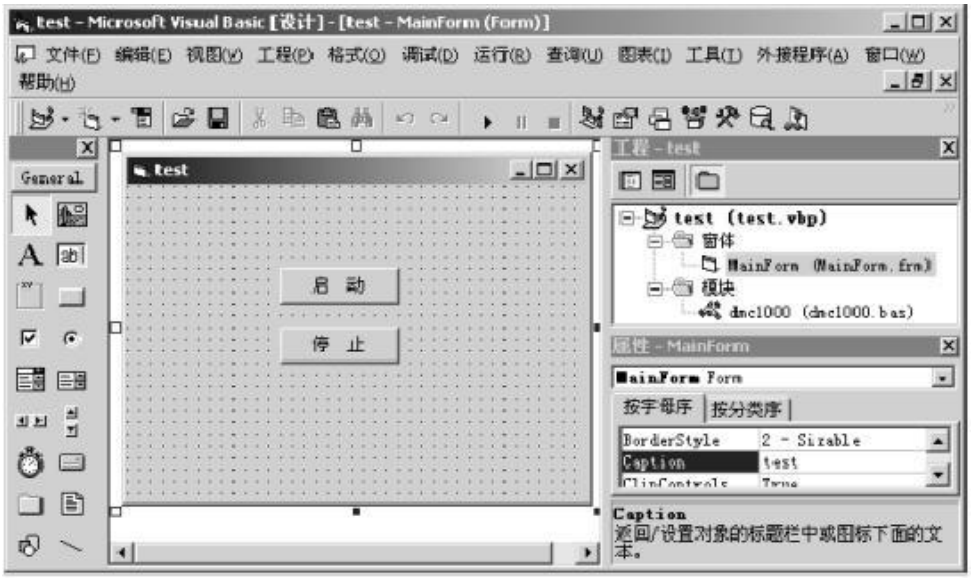


7-10 添加控件

4）把该工程保存在 E:\test 目录下。

5)在 Motion1000S 安装目录找到 DMC1000.bas 和 DMC1000.dll 文件,拷贝到 test 目录下。

6）选择“工程”->“添加模块”->“现存"，找到 test 目录下的 DMC1000.bas 文件，添加到工程中。如图 7-11 所示。



7-11 添加模块

7）双击窗口控件，在 Form_Load 事件中添加代码：d1000_board_init

选择 Unload 事件，在 Form_Unload 事件中添加代码：d1000_board_close

双击“启动”按钮，在 CB_Start_Click 事件中添加代码：

d1000_start_t_move 0,1000,400,1000,0.1

双击“停止”按钮，CB_Stop_Click 事件中添加代码：d1000_decel_stop 0，

如图 7-12 所示。

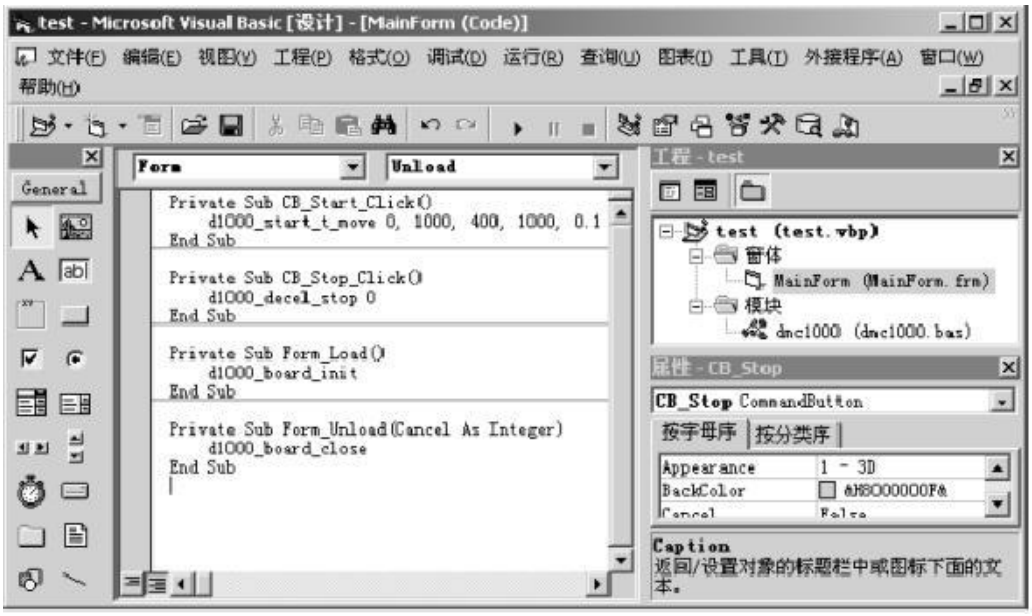


图 7-12 添加运动控制函数

8）运行，按下如图 7-13 所示界面中的“启动”按钮，第 0 轴就会输出 1000 个脉冲。运动中可以按下“停止”按钮，便会减速停止脉冲输出。

7-13 运行 VB 所写程序

第 8 章 附录

8.1 硬件信号接口表

8.1.1 接口 X1 引脚定义

如图 8-1 所示，X1 是 DMC1000S 电机控制、I/O 信号控制的主要接口，为 SCSI-Ⅱ型 68 针插座。

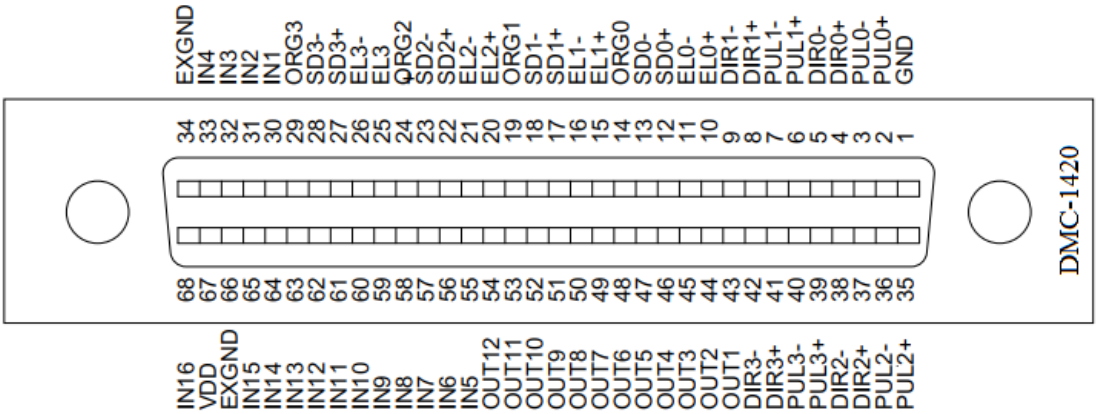


图 8-1 接口 X1 示例

其引脚号与信号的对应关系见表 8-1 所示：

表 8-1 接口 X1 引脚号与信号关系表

引脚	信号	说明	引脚	信号	说明
1	GND	BUS 电源地	35	PUL2+	第 2 轴脉冲正
2	PUL0+	第 0 轴脉冲正	36	PUL2-	第 2 轴脉冲负
3	PUL0-	第 0 轴脉冲负	37	DIR2+	第 2 轴方向正
4	DIR0+	第 0 轴方向正	38	DIR2-	第 2 轴方向负
5	DIR0-	第 0 轴方向负	39	PUL3+	第 3 轴脉冲正
6	PUL1+	第 1 轴脉冲正	40	PUL3-	第 3 轴脉冲负
7	PUL1-	第 1 轴脉冲负	41	DIR3+	第 3 轴方向正
8	DIR1+	第 1 轴方向正	42	DIR3-	第 3 轴方向负
9	DIR1-	第 1 轴方向负	43	OUT1	通用输出信号 1
10	EL0+	第 0 轴正向限位	44	OUT2	通用输出信号 2
11	EL0-	第 0 轴负向限位	45	OUT3	通用输出信号 3
12	IN33/SD0+	通用输入 33/第 0 轴正向减速信号	46	OUT4	通用输出信号 4
13	IN34/SD0-	通用输入 34/第 0 轴负向减速信号	47	OUT5	通用输出信号 5
14	ORG0	第 0 轴原点	48	OUT6	通用输出信号 6
15	EL1+	第 1 轴正向限位	49	OUT7	通用输出信号 7
16	EL1-	第 1 轴负向限位	50	OUT8	通用输出信号 8
17	IN35/SD1+	通用输入 35/第 1 轴正向减速信号	51	OUT9	通用输出信号 9

18	IN36/SD1-	通用输入 36/第 1 轴负向减速信号	52	OUT10	通用输出信号 10
19	ORG1	第 1 轴原点	53	OUT11	通用输出信号 11
20	EL2+	第 2 轴正向限位	54	OUT12	通用输出信号 12
21	EL2-	第 2 轴负向限位	55	IN5	通用输入信号 5
22	IN37/SD2+	通用输入 37/第 2 轴正向减速信号	56	IN6	通用输入信号 6
23	IN38/SD2-	通用输入 38/第 2 轴负向减速信号	57	IN7	通用输入信号 7
24	ORG2	第 2 轴原点	58	IN8	通用输入信号 8
25	EL3+	第 3 轴正向限位	59	IN9	通用输入信号 9
26	EL3-	第 3 轴负向限位	60	IN10	通用输入信号 10
27	IN39/SD3+	通用输入 39/第 3 轴正向减速信号	61	IN11	通用输入信号 11
28	IN40/SD3-	通用输入 40/第 3 轴负向减速信号	62	IN12	通用输入信号 12
29	ORG3	第 3 轴原点	63	IN13	通用输入信号 13
30	IN1/ALM0	通用输入 1/第 0 轴伺服报警（默认）	64	IN14	通用输入信号 14
31	IN2/ALM1	通用输入 2/第 1 轴伺服报警（默认）	65	IN15	通用输入信号 15
32	IN3/ALM2	通用输入 3/第 2 轴伺服报警（默认）	66	EXGND	外部电源地
33	IN4/ALM3	通用输入 4/第 3 轴伺服报警（默认）	67	VDD	外部电源 24V
34	EXGND	外部电源地	68	IN16	通用输入信号 16

注：当需要使用 IO 信号时，如限位、原点等专用信号或通用 IO 信号时，请务必在 67 脚和 66 脚输入+24VDC 电源（请不要将电源接反）。使用 ACC68_V3.2 接线板的用户请注意：其上标有“VDD”与“EXGND”的引脚号，这些引脚仅仅是为接线板的电源指示灯提供电源的，当要控制 IO 信号时，仍需在 67 脚与 66 脚输入外部+24VDC 电源。

IN1~IN4 默认为伺服报警信号，如需作为通用输入口，需调用函数 d1000_set_ALM_PIN_Extern 禁止伺服报警使能功能。

8.1.2 接口 J1 引脚定义

DMC1000S 卡中有扩展 IO 口的接口，其标志为 J1，且其上所有 IO 口都为非隔离的通用输入输出口。其示意图如图 8-3 所示：

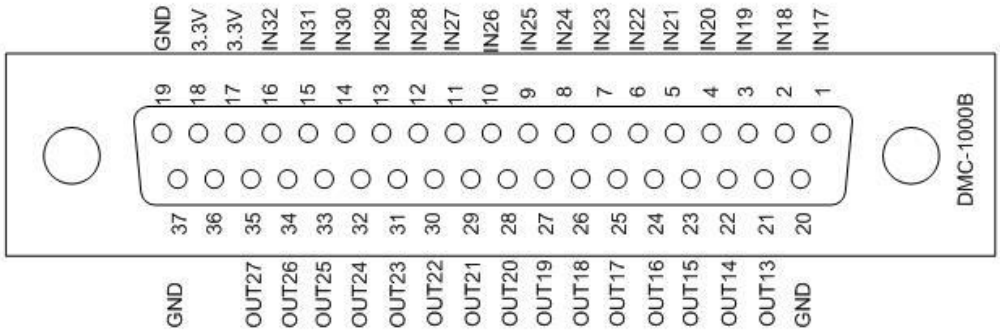


图 8-3 DMC1000S 接口 J1 示意图

其针脚号和针脚名定义见表 8-3 所示：

表 8-3 DMC1000S 接口 J1 引脚号和信号关系表

脚号	名称	I/O	功能	脚号	名称	I/O	功能
1	IN17	I	非隔离通用输入信号 17	20	GND	O	PC 电源地，输出
2	IN18	I	非隔离通用输入信号 18	21	OUT13	O	非隔离通用输出信号 13
3	IN19	I	非隔离通用输入信号 19	22	OUT14	O	非隔离通用输出信号 14
4	IN20	I	非隔离通用输入信号 20	23	OUT15	O	非隔离通用输出信号 15
5	IN21	I	非隔离通用输入信号 21	24	OUT16	O	非隔离通用输出信号 16
6	IN22	I	非隔离通用输入信号 22	25	OUT17	O	非隔离通用输出信号 17
7	IN23	I	非隔离通用输入信号 23	26	OUT18	O	非隔离通用输出信号 18
8	IN24	I	非隔离通用输入信号 24	27	OUT19	O	非隔离通用输出信号 19
9	IN25	I	非隔离通用输入信号 25	28	OUT20	O	非隔离通用输出信号 20
10	IN26	I	非隔离通用输入信号 26	29	OUT21	O	非隔离通用输出信号 21
11	IN27	I	非隔离通用输入信号 27	30	OUT22	O	非隔离通用输出信号 22
12	IN28	I	非隔离通用输入信号 28	31	OUT23	O	非隔离通用输出信号 23
13	IN29	I	非隔离通用输入信号 29	32	OUT24	O	非隔离通用输出信号 24
14	IN30	I	非隔离通用输入信号 30	33	OUT25	O	非隔离通用输出信号 25
15	IN31	I	非隔离通用输入信号 31	34	OUT26	O	非隔离通用输出信号 26
16	IN32	I	非隔离通用输入信号 32	35	OUT27	O	非隔离通用输出信号 27
17	3.3V	O	PC 电源，输出	36	—		未用
18	3.3V	O	PC 电源，输出	37	GND	O	PC 电源地，输出
19	GND	O	PC 电源地，输出				

为方便用户使用 DMC1000S 的扩展 IO 口，雷赛公司提供了专用的光电隔离接线板 ACC37_7480，使用 ACC37_7480 接线板后可实现 DMC1000S 扩展 IO 接口的输入输出信号全部光电隔离，其 ACC37_7480 接线板的引脚号与信号关系如表 8-4 所示：

表 8-4 ACC37_7480 引脚与 DMC1000S_J1 信号关系表

引脚号	信号名称	I/O	功 能	引脚号	信号名称	I/O	功 能
IN1	IN17	I	隔离后的通用输入 17	OUT1	OUT13	O	隔离后的通用输出 13
IN2	IN18	I	隔离后的通用输入 18	OUT2	OUT14	O	隔离后的通用输出 14
IN3	IN19	I	隔离后的通用输入 19	OUT3	OUT15	O	隔离后的通用输出 5
IN4	IN20	I	隔离后的通用输入 20	OUT4	OUT16	O	隔离后的通用输出 16
IN5	IN21	I	隔离后的通用输入 21	OUT5	OUT17	O	隔离后的通用输出 17
IN6	IN22	I	隔离后的通用输入 22	OUT6	OUT18	O	隔离后的通用输出 18
IN7	IN23	I	隔离后的通用输入 23	OUT7	OUT19	O	隔离后的通用输出 19
IN8	IN24	I	隔离后的通用输入 24	OUT8	OUT20	O	隔离后的通用输出 20
IN9	IN25	I	隔离后的通用输入 25	OUT9	OUT21	O	隔离后的通用输出 21
IN10	IN26	I	隔离后的通用输入 26	OUT10	OUT22	O	隔离后的通用输出 22
IN11	IN27	I	隔离后的通用输入 27	OUT11	OUT23	O	隔离后的通用输出 23

IN12	IN28	I	隔离后的通用输入 28	OUT12	OUT24	O	隔离后的通用输出 24
IN13	IN29	I	隔离后的通用输入 29	OUT13	OUT25	O	隔离后的通用输出 25
IN14	IN30	I	隔离后的通用输入 30	OUT14	OUT26	O	隔离后的通用输出 26
IN15	IN31	I	隔离后的通用输入 31	OUT15	OUT27	O	隔离后的通用输出 27
IN16	IN32	I	隔离后的通用输入 32	OUT16	未用	—	未用
EGND	EXGND	I	外部电源地，输入	EGND	EXGND	I	外部电源地，输入
+24V	VDD	I	外部电源+24V，输入	EGND	EXGND	I	外部电源地，输入

表 8-5　DMC1000S 接口 J2 引脚号和信号关系表

脚号	名称	I/O	功能	脚号	名称	I/O	功能
1	GND	O	PC 电源地，输出	20	GND	O	PC 电源地，输出
2	GND	O	PC 电源地，输出	21	GND	O	PC 电源地，输出
3	EA0+	I	第 1 轴编码器 A+相	22	EA2+	O	第 2 轴编码器 A+相
4	EA0-	I	第 1 轴编码器 A-相	23	EA2-	O	第 2 轴编码器 A-相
5	EB0+	I	第 1 轴编码器 B+相	24	EB2+	O	第 2 轴编码器 B+相
6	EB0-	I	第 1 轴编码器 B-相	25	EB2-	O	第 2 轴编码器 B-相
7	EZ0+	I	第 1 轴编码器 Z+相	26	EZ2+	O	第 2 轴编码器 Z+相
8	EZ0-	I	第 1 轴编码器 Z-相	27	EZ2-	O	第 2 轴编码器 Z-相
9	EA1+	I	第 2 轴编码器 A+相	28	EA3+	O	第 3 轴编码器 A+相
10	EA1-	I	第 2 轴编码器 A-相	29	EA3-	O	第 3 轴编码器 A-相
11	EB1+	I	第 2 轴编码器 B+相	30	EB3+	O	第 3 轴编码器 B+相
12	EB1-	I	第 2 轴编码器 B-相	31	EB3-	O	第 3 轴编码器 B-相
13	EZ1+	I	第 2 轴编码器 Z+相	32	EZ3+	O	第 3 轴编码器 Z+相
14	EZ1-	I	第 2 轴编码器 Z-相	33	EZ3-	O	第 3 轴编码器 Z-相
15	-	-	-	34	-	-	-
16	-	-	-	35	-	-	-
17	5V	O	PC 电源，输出	36	—	-	-
18	5V	O	PC 电源，输出	37	GND	O	PC 电源地，输出
19	-	-	-	38	GND	O	PC 电源地，输出
				39	GND	O	PC 电源地，输出
				40	GND	O	PC 电源地，输出

8.2 动控制函数库

DMC1000S 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很简单：您只要用 C/C++、Visual Basic 开发用户界面，并调用 DMC1000S 函数库中

的相关运动控制函数，您就可以随心所欲的对自己的多轴自动化设备进行精确、高速、协调的控制。因为 DMC1000S 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一样开发出自己的软件系统，从而大大缩短您的软件开发周期。

雷赛 DMC1000S 运动控制卡的运动控制函数库共有 11 类 ，分类列表如下：

8.2.1 函数列表

	函数名	描 述
初始化函数	d1000_board_init	初始化控制卡
	d1000_board_close	关闭控制卡
脉冲模式设置函数	d1000_set_pls_outmode	设定脉冲输出模式
S 形速度曲线设置	d1000_set_s_profile	设置 S 形速度曲线参数
	d1000_get_s_profile	读取 S 形速度曲线参数
连续运动函数	d1000_start_tv_move	以梯形速度曲线控制一个轴连续运动
	d1000_start_sv_move	以 S 形速度曲线控制一个轴连续运动
	d1000_get_speed	读取指定轴的脉冲输出速度
	d1000_change_speed	改变指定轴的脉冲输出速度
	d1000_immediate_stop	以梯形或 S 形急停一个轴
	d1000_decel_stop	以梯形或 S 形减速停止一个轴
单轴运动函数	d1000_start_t_move	以梯形速度曲线控制相对坐标的点位运动
	d1000_start_ta_move	以梯形速度曲线控制绝对坐标的点位运动
	d1000_start_s_move	以 S 形速度曲线控制相对坐标的点位运动
	d1000_start_sa_move	以 S 形速度曲线控制绝对坐标的点位运动
在运动中改变目标位置	d1000_reset_target_position	在运动中改变目标位置
读取指定轴的目标脉冲位置	d1000_get_target_position	读取指定轴的目标脉冲位置
直线插补函数	d1000_start_t_line	任意 2、 3、 4 轴相对坐标的直线插补运动
	d1000_start_ta_line	任意 2、 3、 4 轴绝对坐标的直线插补运动
	d1000_check_done_multicoor	读取插补的状态
	d1000_stop_multicoor	停止插补运动
回原点函数	d1000_set_HOME_pin_logic	设置原点信号的电平
	d1000_config_home_mode	设置回原点模式
	d1000_set_home_el_return	设置限位反找(默认不使能反找)
	d1000_home_move	回原点运动
运动状态检测函数	d1000_check_done	检测当前运动状态
停止原因读取和清除	d1000_get_stop_reason	读取指定轴的停止原因
	d1000_clear_stop_reason	清除停止原因

指令位置设定和读取函数	d1000_get_command_pos	读取指令位置计数器值
	d1000_set_command_pos	设置指令位置计数器值
通用 IO 接口函数	d1000_out_bit	通用输出口输出
	d1000_in_bit	读取通用输入口状态
	d1000_in_bit_ex	读取 SD 输入作为通用输入时的状态
	d1000_get_outbit	读取通用输出口状态
	d1000_in_port	读取输入端口的值
	d1000_get_outport	读取输出端口的值
	d1000_out_port	设置输出端口的值
专用 IO 接口函数	d1000_set_sd	设置减速开关是否有效
	d1000_get_axis_status	读取指定轴的所有输入状态
	d1000_set_ALM_PIN	设置 ALM 信号(停止所有轴配置)
	d1000_set_ALM_PIN_Extern	设置 ALM 信号(使能，停止所有轴或单轴配置)
	d1000_read_ALM_PIN	读取轴 ALM 信号电平状态
	d1000_config_EL_MODE	设置 EL 信号制动方式
	d1000_Enable_EL_PIN	设置 EL 信号的使能状态
编码器相关函数	d1000_counter_config	设置编码器的计数方式
	d1000_get_encoder	读取编码器计数值
	d1000_set_encoder	设置编码器计数值

8.2.2 函数说明

8.2.2.1 初始化函数

DWORD d1000_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

DWORD d1000_board_close (void)

功 能：关闭控制卡，释放系统资源

参 数：无

返回值：正确：返回 ERR_NoError

错误：返回相关错误码，具体含义参照 8.2.3 资源文件，下同。

8.2.2.2 脉冲输出设置函数

DWORD d1000_set_pls_outmode (WORD axis, WORD pls_outmode)

功 能：设置控制卡脉冲输出模式

参 数：axis：轴号。范围 0～(n×4-1)，n 为卡数。多卡运行时，轴号参考 2-1 多卡运行时轴号对照表；

pls_outmode：脉冲输出模式：

0：pulse/dir 模式，脉冲上升沿有效

1：pulse/dir 模式，脉冲下降沿有效

2：CW/CCW 模式，脉冲上升沿有效

3：CW/CCW 模式，脉冲下降沿有效

返回值：正确：返回 ERR_NoError

错误：返回相关错误码

8.2.2.3 S 形速度曲线设置函数

DWORD d1000_set_s_profile(WORD axis,double s_para)

功 能：设置 S 形速度曲线（S 参数为时间）

参 数：axis：轴号，范围 0～(n×4-1)，n 为卡数

s_para： S 形速度曲线参数，范围：0-1s

返回值：正确：返回 ERR_NoError
返回相关错误码

DWORD d1000_get_s_profile(WORD axis,double* s_para)

功 能：回读 S 时间
参 数：axis：轴号，范围 0～(n×4-1)，n 为卡数
s_para： S 形速度曲线参数，范围：0-1s
返回值：正确：返回 ERR_NoError
错误：返回相关错误码

8.2.2.4 连续运动函数

DWORD d1000_start_tv_move (WORD axis, long StrVel, Long MaxVel, double Tacc)

功 能：以梯形速度曲线控制指定轴至运行速度，并以运行速度连续运行
参 数：axis：轴号，范围 0～(n×4-1)，n 为卡数
StrVel： 初始速度，单位：pps
MaxVel： 运行速度，单位：pps，其值的正负表示运动方向
Tacc： 加速时间，单位：s
返回值：正确：返回 ERR_NoError
错误：返回相关错误码

DWORD d1000_start_sv_move (WORD axis, long StrVel, Long MaxVel, double Tacc)

功 能：以 S 形速度曲线控制指定轴至运行速度，并以运行速度连续运行
参 数：axis：轴号，范围 0～(n×4-1)，n 为卡数;
StrVel： 初始速度，单位：pps
MaxVel：运行速度，单位：pps，其值的正负表示运动方;
Tacc： 加速时间，单位：s
返回值：正确：返回 ERR_NoError
错误：返回相关错误码

DWORD d1000_get_speed (WORD axis)

功 能：读取指定轴当前脉冲输出速度;

参 数：axis：轴号，范围 0～(n×4-1) ， n 为卡数

返回值：指定轴当前的运动速度，单位 pps

DWORD d1000_change_speed (WORD axis，long NewVel)

功 能：改变指定轴当前脉冲输出速度

参 数：axis：轴号，范围 0～(n×4-1) ， n 为卡数

NewVel： 新设置的速度，单位：pps，取值范围：1~409550

返回值：正确：返回 ERR_NoError；

错误：返回相关错误码。

注意：

d1000_change_speed 函数变速后的速度与实际设定速度之间有一个允许的误差，误差值为 0~50 pps，具体的误差值与当前速度及速度改变量 V 有关，其绝对值| V|越小，则误差越小。

DWORD d1000_immediate_stop (WORD axis)

功 能：急停指定轴脉冲输出。

参 数：axis：轴号，范围 0～(n×4-1) ， n 为卡数

返回值：正确：返回 ERR_NoError；

错误：返回相关错误码。

DWORD d1000_decel_stop (WORD axis)

功 能：减速停止指定轴脉冲输出。

参 数：axis：轴号，范围 0～(n×4-1) ， n 为卡数

返回值：正确：返回 ERR_NoError；

错误：返回相关错误码。

8.2.2.5 点位运动函数

DWORD d1000_start_t_move (WORD axis, long Dist, long StrVel, long MaxVel, double Tacc)

功 能：以梯形速度曲线控制指定轴至运行速度，并以相对坐标运行一段指定距离

参 数：axis：轴号，范围 0～(n×4-1) ， n 为卡数

Dist： 相对运动距离，单位：pulse，其值的正负表示运动方向

StrVel： 初始速度，单位：pps；

MaxVel：运行速度，单位：pps；

Tacc：加速时间，单位：s。

返回值：正确：返回 ERR_NoError；

错误：返回相关错误码。

DWORD d1000_start_ta_move (WORD axis, long Pos, double StrVel, double MaxVel, double Tacc)

功 能：以梯形速度曲线控制指定轴至运行速度，并以绝对坐标运行一段指定距离

参 数：axis：轴号，范围 0～(n×4-1) ，n 为卡数

Pos： 绝对运动位置，单位：pulse

StrVel： 初始速度，单位：pps

MaxVel：运行速度，单位：pps

Tacc： 加速时间，单位：s

返回值：正确：返回 ERR_NoError

错误：返回相关错误码

DWORD d1000_start_s_move (WORD axis, long Dist, long StrVel, long MaxVel, double Tacc)

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

参 数：axis：轴号，范围 0～(n×4-1) ，n 为卡数

Dist： 相对运动距离，单位：pulse，其值的正负表示运动方向

StrVel： 初始速度，单位：pps

MaxVel：运行速度，单位：pps

Tacc： 加速时间，单位：s

返回值：正确：返回 ERR_NoError

错误：返回相关错误码

DWORD d1000_start_sa_move (WORD axis, long Pos, long StrVel, long MaxVel, double Tacc)

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

参 数：axis：轴号，范围 0～(n×4-1)，n 为卡数

Pos： 绝对运动位置，单位：pulse

StrVel： 初始速度，单位：pps

MaxVel：运行速度，单位：pps

Tacc： 加速时间，单位：s

返回值：正确：返回 ERR_NoError
错误：返回相关错误码

DWORD d1000_reset_target_position(WORD axis,long dist)

功 能：在运动中改变目标位置

参 数：axis：轴号，范围 0～(n×4-1)，n 为卡数
dist：目标位置，单位： pulse

返回值：正确：返回 ERR_NoError
错误：返回相关错误码

DWORD d1000_get_target_position(WORD axis)

功 能：读取指定轴的目标脉冲位置

参 数：axis： 轴号，范围 0～(n×4-1)，n 为卡数

返回值：指定轴的目标脉冲位置

8.2.2.6 直线插补函数

DWORD d1000_start_t_line (short TotalAxis, short *AxisArray, short *DistArray, long StrVel, long MaxVel, double Tacc）

功 能：启动多轴相对坐标的直线插补。

参 数： TotalAxis： 插补轴数，范围 2～4；
*AxisArray， AxisArray：轴号列表；
*DistArray， DistArray：对应轴号列表各轴的相对坐标的距离列表
StrVel： 初始速度，单位：pps
MaxVel： 运行速度，单位：pps
Tacc： 加速时间，单位：s

返回值：正确：返回 ERR_NoError
错误：返回相关错误码

DWORD d1000_start_ta_line (short TotalAxis, short *AxisArray, short *PosArray, long StrVel, long MaxVel, double Tacc)

功 能：启动多轴绝对坐标的直线插补

参 数：TotalAxis：插补轴数，范围 2～4

*AxisArray，AxisArray：轴号列表

*PosArray，PosArray：对应轴号列表各轴的绝对坐标的位置列表

StrVel： 初始速度，单位：pps；

MaxVel： 运行速度，单位：pps

Tacc： 加速时间，单位：s

返回值：正确：返回 ERR_NoError

错误：返回相关错误码

DWORD d1000_check_done_multicoor(WORD cardno,WORD crd)

功 能：读取插补的状态

参 数：cardno：卡号:0-7

crd：插补系号：0

返回值：正确：返回 ERR_NoError

错误： 返回相关错误码

DWORD d1000_stop_multicoor(WORD cardno,WORD crd,WORD stop_mode)

功 能：停止插补运动

参 数：cardno：卡号：0-7

crd：插补系号：0

stop_mode：停止模式，0：减速停止 1：立即停止

返回值：正确：返回 ERR_NoError

错误：返回相关错误码

8.2.2.7 回原点函数

DWORD d1000_set_HOME_pin_logic(WORD axis,WORD org_logic,WORD filter)

功 能：设置原点信号有效电平

参 数：axis：轴号，范围 0～(n×4-1)，n 为卡数

org_logic：ORG 有效信号，0：低电平 1：高电平

filter：保留参数，固定值为 0

返回值：正确：返回 ERR_NoError

错误：返回相关错误码

DWORD d1000_config_home_mode(WORD axis,WORD mode,WORD EZ_count)

功 能：设置回零模式

参 数：axis：轴号，范围 0～(n×4-1)，n 为卡数

mode： 回原点运动模式

0 一次回零

1 一次回零 + 反找原点

2 二次回零

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： 保留参数，固定值为 0

返回值：正确：返回 ERR_NoError

错误：返回相关错误码

DWORD d1000_set_home_el_return((WORD axis,WORD enable)

功 能：设置回零遇限位是否反找

参 数：axis：轴号，范围 0～(n×4-1)，n 为卡数

Enable 使能是否遇限位反找，0 不使能（默认），1 使能

返回值：正确：返回 ERR_NoError

错误：返回相关错误码

DWORD d1000_home_move (short axis, long StrVel, long MaxVel, double Tacc)

功 能：启动指定轴进行回原点运动

参 数：axis：轴号，范围 0～(n×4-1)，n 为卡数

StrVel： 回原点运动初始速度，单位：pps

MaxVel：回原点运动速度，负值表示往负方向找原点

正值表示往正方向找原点，单位：pps

Tacc： 加速时间，单位：s

返回值：正确：返回 ERR_NoError

错误：返回相关错误码

8.2.2.8 运动状态检测函数

DWORD d1000_check_done (WORD axis)

功 能：检测指定轴的运动状态

参 数：axis： 轴号，范围 0～(n×4-1)，n 为卡数

返回值：0：正在运行

1：脉冲输出完毕停止

2：指令停止（如调用了 d1000_decel_stop 函数）

3：遇限位停止

4：遇原点停止

注 意：1、当指定轴遇限位信号停止时，d1000_check_done 函数返回 3，若限位信号撤去，d1000_check_done 函数返回值变为 1

2、当指定轴遇原点信号停止时，d1000_check_done 函数返回 4，若原点信号撤去，d1000_check_done 函数返回值变为 1

DWORD d1000_get_stop_reason(WORD axis,long* stopreason)

功 能：读取指定轴的停止原因

参 数：axis： 轴号，范围 0～(n×4-1)，n 为卡数

stopreason：返回停止原因

返回值：正确：返回 ERR_NoError

错误：返回相关错误码。

DWORD d1000_clear_stop_reason(WORD axis)

功 能：清除停止原因

参 数：axis： 轴号，范围 0～(n×4-1)，n 为卡数

返回值：正确：返回 ERR_NoError

错误：返回相关错误码。

8.2.2.9 指令位置设定和读取函数

DWORD d1000_get_command_pos (WORD axis)

功 能：读取指令位置计数器计数值

参 数：axis：轴号，范围 0～(n×4-1)，n 为卡数

返回值：指定轴当前指令位置计数器值，单位：pulse

DWORD d1000_set_command_pos (WORD axis, double Pos)

功 能：设置指令位置计数器计数值

参 数：axis： 轴号，范围 0～(n×4-1)，n 为卡数
Pos：设置指令位置计数器值，单位：pulse
返回值：正确：返回 ERR_NoError
错误：返回相关错误码

8.2.2.10 通用 I/O 接口函数

DWORD d1000_out_bit (short BitNo, short BitData)

功 能：输出通用输出信号
参 数：BitNo：表示要输出的通用输出口的位号，范围参考表 8-5 及表 8-6
BitData：输出信号：0 - 表示低电平
1 - 表示高电平
返回值：正确：返回 ERR_NoError
错误：返回相关错误码
注：多卡 IO 位号分配说明，如表 8-5 所示：

8-5 多卡 IO 位号分配

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

DWORD d1000_in_bit (short BitNo)

功 能：读取通用输入信号状态
参 数：BitNo：表示要读取的通用输入口的位号，范围参考表 8-5
返回值：输入口状态： 0 - 表示低电平
1 - 表示高电平

DWORD d1000_in_bit_ex(WORD cardno,WORD BitNo)

功 能：读取 SD 输入作为通用输入时的状态
参 数：cardno：卡号
BitNo：表示要读取的通用输入口的位号
返回值：输入口状态： 0 - 表示低电平
1 - 表示高电平

注意：当 SD 输入作为通用输入时，（IN33~IN40）对应 4 个轴的 SD-/+,用 in_bit_ex 读取

DWORD d1000_get_outbit (short BitNo)

功 能：读取通用输出信号状态

参 数：BitNo：通用输出口位号，范围参考表 8-5

返回值：输出口状态： 0 - 表示低电平

1 - 表示高电平

DWORD d1000_in_port(WORD cardno)

功 能：读取输入端口的值

参 数：CardNo ：卡号

返回值： 全 部 输入口的电平

DWORD d1000_get_outport(WORD cardno)

功 能： 读取输出端口的值

参 数： CardNo ：卡号

返回值：全 部 输出口的电平

DWORD d1000_ out_port(WORD cardno, DWORD PortData)

功 能： 设置输出端口的值

参 数： CardNo ：卡号

PortData ： 设置全部输出口的电平

返回值：正确：返回 ERR_NoError

错误：返回相关错误码

8.2.2.11 专用 I/O 接口函数

DWORD d1000_set_sd (short axis, short SdMode)

功 能：设置减速信号是否使能

参 数：axis：轴号，范围 0～(n×4-1)， n 为卡数

SdMode： 减速使能模式

0：SD 信号无效

1：SD 信号有效，定长和定速运动有效

返回值：正确：返回 ERR_NoError

错误：返回相关错误码

DWORD d1000_get_axis_status (short axis)

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

参 数：axis：轴号，范围 0～(n×4-1)， n 为卡数

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

表 8-7 返回值的位号与信号对应表

位 号	信 号	电 平
0	-EL	0：高，1：低
1	+EL	0：高，1：低
2	ORG	0：高，1：低
3	--	
4	--	
5	-SD	0：高，1：低
6	+SD	0：高，1：低
7	保留	保留

DWORD d1000_set_ALM_PIN(WORD axis,WORD alm_logic,WORD alm_action)

功 能：设置 ALM 信号

参 数：axis：轴号，范围 0～(n×4-1)， n 为卡数

alm_logic：ALM 信号有效电平，0：低电平 1：高电平

alm_action：停止模式，0：立即停止

返回值：正确：返回 ERR_NoError

错误：返回相关错误码

DWORD d1000_set_ALM_PIN_Extern(WORD axis,WORD alm_enable,WORD alm_logic,WORD alm_all,WORD alm_action)(默认使能)

功 能：设置 ALM 信号(使能，停止所有轴或单轴配置)

参 数：axis：轴号，范围 0～(n×4-1)， n 为卡数

alm_enable：ALM 信号使能状态，0：禁止，1：允许（默认）

alm_logic：ALM 信号有效电平，0：低电平（默认），1：高电平

alm_all：ALM 信号控制方式，0：停止单轴（默认），1：停止所有轴

alm_action：停止模式，0：立即停止

返回值：正确：返回 ERR_NoError

错误：返回相关错误码

注意：IN1~IN4 默认为伺服报警信号，如需作为通用输入口，需调用函数 d1000_set_ALM_PIN_Extern 禁止伺服报警使能功能。

DWORD d1000_read_ALM_PIN(WORD axis)

功 能：读取轴 ALM 信号电平状态

参 数：axis：轴号，范围 0～(n×4-1)， n 为卡数

返回值：轴 ALM 信号电平状态

DWORD d1000_config_EL_MODE(WORD axis,WORD el_mode)

功 能：设置 EL 信号制动方式（默认立即停）

参 数：axis：轴号，范围 0～(n×4-1)， n 为卡数

el_mode：停止模式，0：立即停止 1：减速停止

返回值：正确：返回 ERR_NoError

错误：返回相关错误码

DWORD d1000_Enable_EL_PIN(WORD axis, WORD enable)

功 能：设置 EL 信号的使能状态（默认使能）

参 数：axis：轴号，范围 0～(n×4-1)， n 为卡数

enable：0：禁止 ， 1：使能

返回值：正确：返回 ERR_NoError

错误：返回相关错误码

DWORD d1000_config_softlimit(WORD axis,WORD ON_OFF, WORD source_sel,WORD SL_action, long N_limit,long P_limit)

功 能：设置软件限位

参 数：axis：轴号，范围 0～(n×4-1)， n 为卡数

ON_OFF 使能状态， 0：禁止， 1：允许

source_sel 计数器选择， 0：指令位置计数器， 1：编码器计数器

SL_action 限位停止方式， 0：立即停止 1：减速停止

N_limit 负限位位置，单位： pulse

P_limit 正限位位置，单位： pulse

返回值：正确：返回 ERR_NoError

错误：返回相关错误码

注 意： 正、负限位位置可为正数也可为负数，但正限位位置应大于负限位位置

DWORD d1000_get_config_softlimit(WORD axis,WORD *ON_OFF, WORD *source_sel,WORD *SL_action, long *N_limit,long *P_limit)

功 能：读取软件限位设置

参 数： axis：轴号，范围 0～(n×4-1)， n 为卡数

ON_OFF 返回使能状态

source_sel 返回计数器选择

SL_action 返回限位停止方式

N_limit 返回负限位 pulse

P_limit 返回正限位 pulse

返回值：正确：返回 ERR_NoError

错误：返回相关错误码

8.2.2.12 编码器相关函数

DWORD d1000_counter_config(WORD axis,WORD mode)

功 能：设定编码器的计数方式

参 数： axis：轴号，范围 0～(n×4-1)， n 为卡数

mode：编码器反馈输入模式

0 ： 保留

1 ： 1 倍频 AB 相脉冲信号

2 ： 2 倍频 AB 相脉冲信号

3 ： 4 倍频 AB 相脉冲信号

返回值：正确：返回 ERR_NoError

错误：返回相关错误码

DWORD d1000_get_encoder(WORD axis)

功能：读取编码器反馈的脉冲计数值

参数： axis：轴号，范围 0～(n×4-1)， n 为卡数

返回值：编码器的计数值

DWORD d1000_set_encoder(WORD axis,long encoder_value)

功能：设置编码器反馈脉冲计数值

参数： axis：轴号，范围 0～(n×4-1)， n 为卡数

返回值： 正确：返回 ERR_NoError

错误：返回相关错误码

8.2.3 资源文件

C/C++：DMC1000.h

Visual Basic：DMC1000.bas

注意：

（1）在所有运动函数参数设置中，MaxVel 的绝对值一定要大于 StrVel 的绝对值，否则实际速度为当前设置的最大速度。

（2）错误号的含义如下：

ERR_NoError 值为 0，代表参数设置正确；

ERR_BoardNumber 值为 1，代表卡号参数设置错误；

ERR_ParaAxis 值为 2，代表轴号参数设置错误；

ERR_ParaData 值为 3，除以上参数外的其他参数设置错误。

8.4 常见问题库

出现问题	解决建议
板卡插上后， PC 机系统还不能识别 DMC1000S	检查板卡驱动是否正确安装，在 WINDOWS 的设备管理器（可参看 WINDOWS 帮助文件）中查看驱动程序安装是否正常。如果发现有相关的黄色感叹号标志，说明安装不正确，需要按照软件部分安装指引，重新安装； 计算机主板兼容性差，请咨询主板供应商； PCI 插槽是否完好； PCI 金手指是否有异物，可用酒精清洗。
PC 机不能和 DMC1000S 通讯	PCI 金手指是否有异物，可用酒精清洗； 参考软件手册检查应用软件是否编写正确。
板卡和驱动器电机连接后，发出脉冲时，电机不转动	板卡上的设置脉冲发送方式和驱动器的输入脉冲方式是否匹配，跳线 J1~J8 是否正确； 可以用 motion 软件进行测试，观察脉冲计数等是否正常； 是否已经接上供给脉冲和方向的外部电源。
控制卡已经正常工作，正常发出脉冲，但电机不转动	检查驱动器和电机之间的连接是否正确。可以使用 motion 软件进行测试。 确保驱动器工作正常，没有出现报警。
电机可以转动，但工作不正常	检查控制卡和驱动器是否正确接地，抗干扰措施是否做好；脉冲和方向信号输出端光电隔离电路中使用的限流电阻过大，工作电流偏小。
能够控制电机，但电机出现振荡或是过冲	可能是驱动器参数设置不当，检查驱动器参数设置； 应用软件中加减速时间和运动速度设置不合理。
能够控制电机，但工作时，回原点定位不准	检查屏蔽线是否接地； 原点信号开关是否工作正常； 所有编码信号和原点信号是否受到干扰。
限位信号不起作用	限位传感器工作不正常； 限位传感器信号受干扰； 应用程序紊乱。
数字输入信号不能读取	接线是否正常； 检查函数调用是否正确。
数字输出信号不正常	接线是否正常； 检查函数调用是否正确。

8.5 抗干扰措施

DMC1000S 运动控制卡严格遵循抗干扰原则精心设计，具有较高的抗干扰能力，但工业环境往往比较恶劣和复杂，影响控制系统可靠和安全运行的主要因素主要来自系统内部和外部的各种电气干扰，以及系统结构设计、安装和外部环境条件等因素。对于干扰信号严重的应用场合，建议采取一定的措施增强控制系统的抗干扰能力。

可以在接地、滤波、屏蔽这三个方面，增加一些抗干扰措施：

DMC1000S 运动控制卡的 PC 电源必须和驱动器或其他干扰机器的电源分开，使用不同的电源；PC 机箱必须直接接地。

对于一些干扰信号较大的场合建议使用滤波器对电源滤波。

通信电缆建议使用带屏蔽层的电缆，对于方向脉冲信号和编码器信号，建议使用双绞屏蔽线连接。对于恶劣工作环境下的输入输出信号，建议使用单独的电源供电。

建议机箱内电机电源线与信号线不要并行走线。

控制卡和电机需要有一定的距离，计算机最好安装在金属控制柜中。使用变频器时，注意变频器和控制系统要有一定的距离。

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：深圳市深圳市南山区学苑大道 1001 号南山智园 A 3 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