



**深圳市雷赛控制技术有限公司**

SHENZHEN LEADSHINE CONTROL TECHNOLOGY CO.,LTD

# **EMC1400 脉冲型独立式控制器**

## **用户使用手册**

**2024.06.20**

©Copyright 2023 Leadshine Control 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。

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。

# 目录

第 1 章	产品概述.....	3
1.1	雷赛控制 EMC1400 脉冲控制器的特点.....	3
1.2	EMC1400 脉冲控制器主要技术指标.....	3
1.3	EMC1400 脉冲控制器的典型应用.....	4
1.4	订货信息.....	5
1.5	产品图片.....	6
第 2 章	雷赛控制 EMC1400 脉冲控制器功能介绍.....	7
2.1	通讯功能.....	7
2.2	参数下载.....	9
2.3	运动控制功能.....	9
2.4	通用 IO 控制.....	13
2.5	EMC 控制器组网.....	13
第 3 章	硬件接口电路.....	14
3.1	接口布局.....	14
3.2	接口及引脚定义.....	15
3.3	通用 I/O 接口电路.....	17
第 4 章	Motion 调试软件的使用方法.....	20
4.1	Motion 调试软件的使用步骤.....	20
4.2	Motion 软件的功能与使用方法.....	21
第 5 章	应用软件开发方法.....	28
5.1	基于 WINDOWS 平台的应用软件结构.....	28
5.2	采用 VB 6.0 开发应用软件的方法.....	29
5.3	采用 VC 6.0 开发应用软件的方法.....	31
5.4	采用 C# 开发应用软件的方法.....	34
第 6 章	雷赛控制 EMC1400 脉冲控制器基本功能实现方法.....	37
6.1	控制器初始化.....	37
6.2	脉冲输出模式设置.....	37
6.3	停止命令的设置.....	38
6.4	回原点运动的实现.....	38
6.5	点位运动的实现.....	40
6.6	连续运动的实现.....	43

---

6.7	检测轴到位状态功能的实现.....	44
6.8	通用 I/O 控制的实现 .....	45
6.9	限位开关设置.....	46
第 7 章	基本功能函数说明.....	48
7.1	控制器设置函数.....	48
7.2	脉冲模式设置函数.....	48
7.3	回原点运动函数.....	49
7.4	位置计数器控制函数.....	52
7.5	运动状态检测及控制相关函数.....	52
7.6	单轴运动速度曲线设置函数.....	54
7.7	单轴运动函数.....	55
7.8	通用输入输出 IO 函数 .....	56
7.9	限位设置.....	60
7.10	打印输出函数.....	61
附 录	.....	62
附录 1	常见电路接线方法.....	62
附录 2	常见错误码说明.....	65
附录 4	运动控制函数索引.....	67
附录 5	常见问题解决方法.....	69

## 第 1 章 产品概述

### 1.1 雷赛控制 EMC1400 脉冲控制器的特点

EMC1400 运动控制器是深圳市雷赛控制技术有限公司开发的具有自主知识产权的新型脉冲运动控制器。该控制器拥有强大的点位运动控制功能。通讯接口方面，使用以太网和上位机进行通讯，通信速率可达到 100Mbps；支持背板总线，可直接在控制器背板上扩展模块。

EMC1400 其主要特点如下：

- 1) 接线方便，通过网线或者Type-c与电脑进行通讯。
- 2) 接口丰富，支持254个控制器组网控制，同时自带本体4入8出通用IO。
- 3) 支持背板扩展，最大可扩展至8个模块，支持类型有8入8出，16入16出，32出，32入等IO模块。
- 4) 轴控能力强，可控制4个脉冲轴。
- 5) 开放易用，提供易用的上位机调试软件，支持Windows/Linux/MacOS系统。
- 6) 算法优异，稳定可靠的点位运动算法，支持T形、S形速度规划。
- 7) 支持位置控制、速度控制、支持运动中变速功能。
- 8) 设置精良小巧，外形美观，安装方式兼容标准DIN导轨安装。

### 1.2 EMC1400 脉冲控制器主要技术指标

表 1.1 EMC1400 主要技术指标

技术指标 \ 控制器类型	EMC1400
轴数	4
通用数字输入口数量	4（可扩展）
通用数字输出口数量	8（可扩展）
通用数字输入口	光电隔离，RC滤波
通用数字输入口导通电流	≥4.2 mA（15V） 典型值6.9mA （24V）
通用数字输入口最高响应频率	4 kHz
通用数字输出口	光电隔离，集电极开路
通用数字输出口最大电流	500 mA（5~24Vdc，吸入）

技术指标 \ 控制器类型	EMC1400
工作温度	0~50 °C
贮存温度	-20~80 °C
湿度	5~85 %，非结露
外部电源（输入）	24VDC±5% ， 1A
运动控制函数库	支持VC、VB6.0、C#、VB.NET、LabVIEW、Delphi等多种语言
调试软件	Motion调试软件

表 1.2 EMC1400 主要功能

功能 \ 控制器类型	EMC1400
点位运动	√
在线变速	√
回零运动	√

### 1.3 EMC1400 脉冲控制器的典型应用

EMC1400脉冲控制器可应用于各行各业自动化设备中。主要面向3C、电子半导体、等行业客户群体。

主要设备有：

- ◆ 3C 组装段设备：测试类设备，上下料设备；
- ◆ 基础电子：锡膏印刷，锁附类设备，焊锡机；

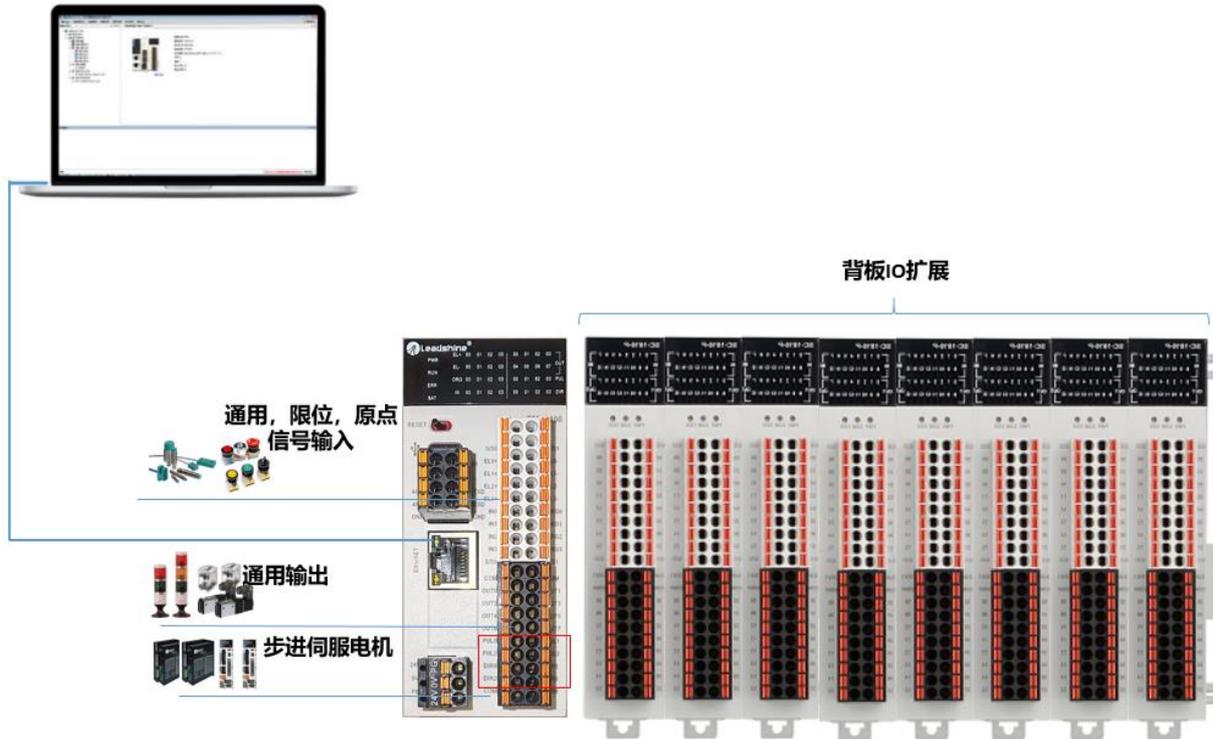


图1.1 EMC1400脉冲控制器典型系统架构

## 1.4 订货信息

表 1.3 EMC1400 脉冲控制器订货信息

产品系列	控制器型号	备注
EMC1000经济点位型	EMC1400	背板最大可扩展8个模块，EMC1400中：1表示经济型控制器，4代表4轴控制，以此类推；

表 1.4 EMC1400 脉冲控制器背板总线配件订货信息

类型	型号	点数	接线端子
	SC-1600	16入	弹压式
	SC-3200	32入	弹压式
	SC-3200-1	32入	MIL接头
	SC-0016-N	16出	弹压式

IO模块	SC-0016-P	16出	弹压式
	SC-0016-R	16出	弹压式（继电器输出模块）
	SC-0032-N	32出	弹压式
	SC-0032-N-1	32出	MIL接头
	SC-0032-N-2	32出	富士通接头
	SC-0808-N	8入8出	弹压式
	SC-1616-N	16入16出	弹压式
	SC-1616-P	16入16出	弹压式

## 1.5 产品图片

EMC1400脉冲控制器外观示意图如图1.5.1所示。

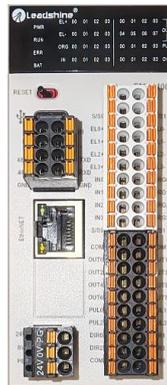


图1.5.1 EMC1400脉冲控制器外观示意图

EMC1400脉冲控制器尺寸示意图

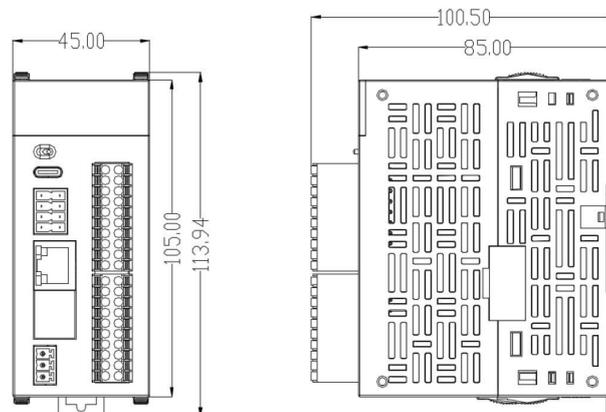


图 1.5.2 EMC1400 脉冲控制器结构尺寸图

## 第 2 章 雷赛控制 EMC1400 脉冲控制器功能介绍

雷赛控制 EMC1400 脉冲控制器是一款新型的脉冲运动外置卡，可以控制多个步进电机或伺服电机；适合多轴点位，IO 控制等功能应用。

EMC1400 脉冲控制器的运动控制函数库功能丰富、易学易用，用户开发应用软件十分方便。随卡免费提供的 Motion 调试软件，不但可以演示 EMC1400 脉冲控制器的控制功能，而且可用于控制器及运动控制系统的硬件测试。

### 2.1 通讯功能

#### 2.1.1 RS232 通信（保留）

EMC1400 运动控制器提供一路 RS232 串口，通讯参数可以根据实际情况配置，支持 MODBUS RTU 协议以及自定义协议。

##### 2.1.1.1 参数配置

EMC1400 运动控制器的 RS232 通信参数如表 2.1 所示

参数名称	出厂设置	数值范围
数据位	8	7, 8
停止位	1	1, 2
校验	2	0 无校验, 1 奇校验, 2 偶校验

表 2.1 RS232 通信参数

注意：使用串口通信时请注意通讯速率和线长相匹配，通讯速率和通讯线长成反比，波特率设置为 115200 时，线长应不超过 3 米，宜采用完整带屏蔽的通信线，不建议使用转接线。

##### 2.1.1.2 通讯协议

EMC1400 运动控制器的 RS232 串口支持 MODBUS RTU 协议，可以和任何支持标准 MODBUS RTU 协议的外设进行数据交互。

EMC1400 运动控制器的 RS232 串口还支持自由协议，用户可以自定义协议规则，使用对应的读写指令进行操作。

## 2.1.2 RS485 通讯（保留）

EMC1400 运动控制器提供一路 RS485 串口，通讯参数可以根据实际情况配置，支持 MODBUS RTU 协议以及自定义协议。

### 2.1.2.1 参数配置

EMC1400 运动控制器的 RS485 通信参数如表 3.2 所示

参数名称	出厂设置	数值范围
数据位	8	7, 8
停止位	1	1, 2
校验	2	0 无校验, 1 奇校验, 2 偶校验

表 2.2 RS485 通信参数

### 2.1.2.2 通信协议

EMC1400 运动控制器的 RS485 串口支持 MODBUS RTU 协议，可以和任何支持标准 MODBUS RTU 协议的外设进行数据交互。

EMC1400 运动控制器的 RS485 串口还支持自由协议，用户可以自定义协议规则，使用对应的读写指令进行操作。

## 2.1.3 以太网通讯

### 2.1.3.1 参数配置

EMC1400 运动控制器出厂默认 IP 地址为：

Ethernet: 192.168.5.11

用于以太网连接 PC 和控制器，控制器的 IP 地址第 4 网段可以任意设置。设置方式可在 Motion 中操作。用户也可根据自己的需要调用相关函数，为控制器设置不同的 IP 地址。在 PC 机与控制器连接之前，需要保证 PC 机的 IP 地址与控制器 IP 地址前 3 个处于同一网段，如 EMC 默认 IP 地址为 192.168.5.11，PC 机设置的 IP 与 EMC 的 IP 前 3 个字段要相同，第 4 个字段要不同。如图 2.1 所示，PC 机设置的 IP 设为 192.168.5.6 即可。

注意：网口通讯只用于控制器与上位机通讯。

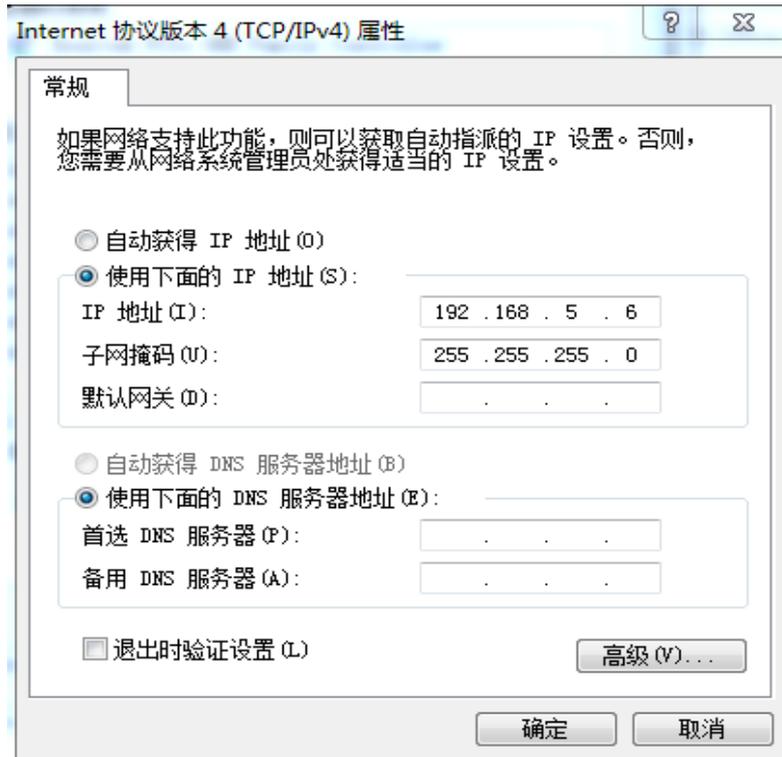


图 2.1 PC 网段设置

### 2.1.3.2 通讯协议

EMC1400 运动控制器 Ethernet 网口支持标准 TCP/IP 协议，可实现多台控制器组网。最大支持 254 台控制器进行组网。使用方便简洁，可极大的为用户节省工控机的成本。用户只需将每个控制器的地址 IP 设置不相同，链接号进行区分。即可精准的操作每个控制器。

## 2.2 参数下载

Motion 调试软件支持将 EMC1400 脉冲控制器的轴,IO 等相关参数，包括运动参数、回零参数等通过文件的形式导出保存，下次使用控制器时可以将参数文件通过 motion 导入至控制器本地。

## 2.3 运动控制功能

### 2.3.1 点位运动

点位运动是指：控制器控制运动平台从当前位置开始以设定的速度运动到指定位置后准确

地停止。

点位运动只关注终点坐标，对运动轨迹的精度没有要求。通过调用点位运动指令函数，使控制器按设定的速度发出指令，运动的距离由点位运动指令决定。位移与时间的关系如图2.2所示。

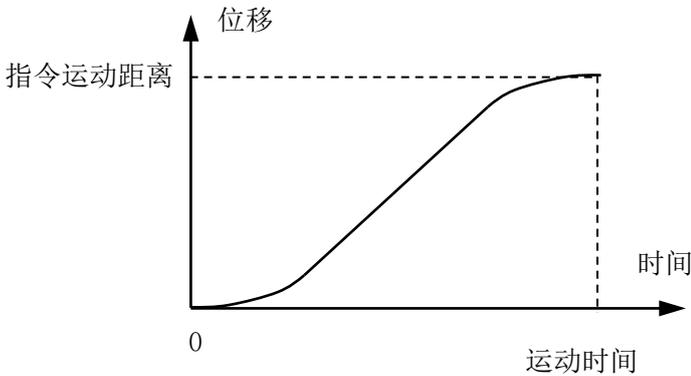


图2.2 定长运动位移曲线

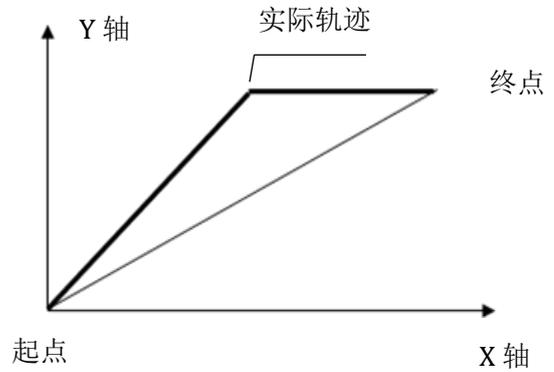


图2.3 两轴复合运动的轨迹

多轴同时做点位运动，称之为多轴联动。由于软件处理速度远比机械系统响应速度快，虽然多条运动指令连续发出，需几个微秒，可对机械系统而言，可认为是同时启动。如果每个轴的运动速度相同，则多轴运动的轨迹就可能是折线，如图2.3所示。

如果从起点到终点都需要按照规定的路径运动，就必须采用直线插补或圆弧插补功能。

### 2.3.1.1 梯形速度控制

为了让平台在运动过程中能平稳加速、准确停止，一般采用梯形速度曲线控制运动过程，如图2.4所示。即：电机以起始速度开始运动，加速至最大速度后保持速度不变，结束前减速至停止速度，并停止。运动的距离由点位运动指令决定。

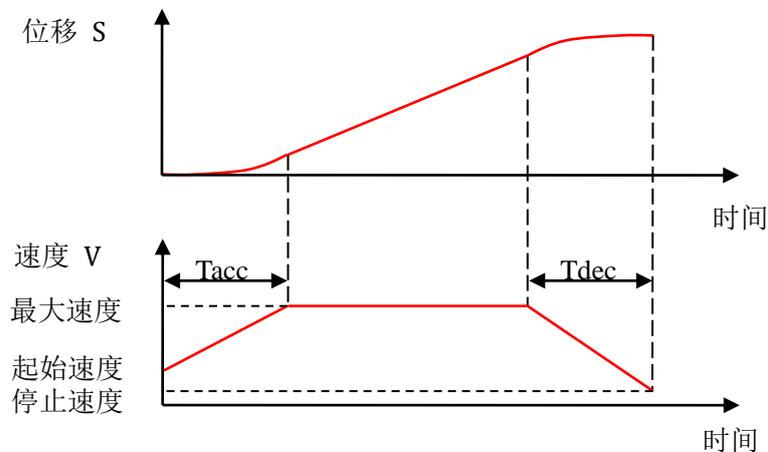


图2.4 梯形速度曲线及对应的位移曲线

Tacc: 总加速时间, Tdec: 总减速时间

### 2.3.1.2 S 形速度控制

为改善平台运动的平稳性，控制器还提供了S形速度控制曲线。

在S形速度控制过程中，指令速度按照内部设定的加速度作S形平滑加速到目标速度值，然后作匀速运动；运动结束前，指令速度作S形减速运动到停止速度，这时达到目标位置，运动停止，如图2.5所示。

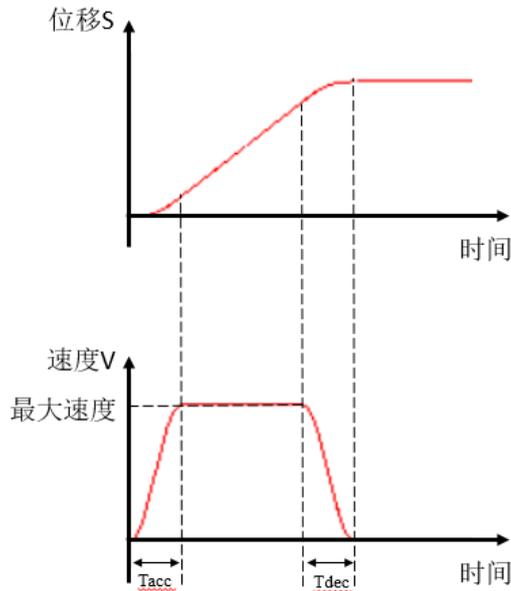


图2.5 S形速度曲线及对应的位移曲线

Tacc: 总加速时间, Tdec: 总减速时间

### 2.3.1.3 运动中改变当前速度

在点位运动（或连续运动）过程中，EMC1400 脉冲控制器可以改变运动的速度，且速度变化过程和预设的梯形速度曲线或 S 形速度曲线相同，如图 2.6 所示。

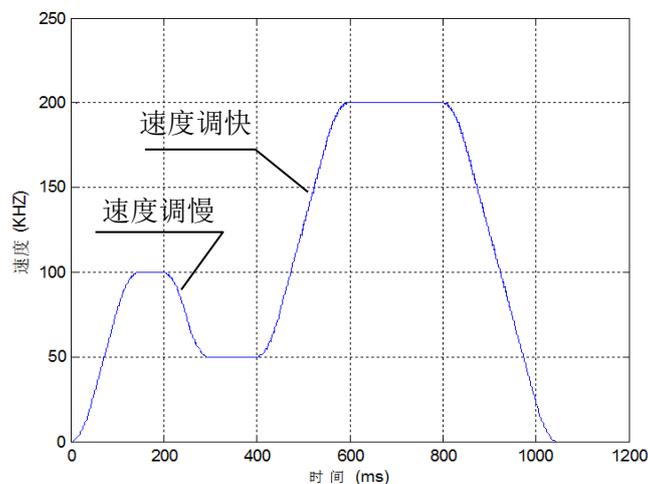


图2.6 调速过程

### 2.3.2 连续运动

连续运动是指：电机从起始速度开始运行，加速至最大速度后连续运动；只有当接收到停止指令或外部停止信号后，才减速停止。

连续运动指令其实就是速度控制指令，国外控制器将此指令称为 JOG 指令。

EMC1400 脉冲控制器可以控制电机以梯形或 S 形速度曲线在指定的加速时间内从起始速度加速至最大速度，然后以该速度连续运行，直至调用停止指令或者该轴遇到限位信号才会按设定的减速方式减速停止。连续运动指令的速度与时间曲线如图 2.7 所示。

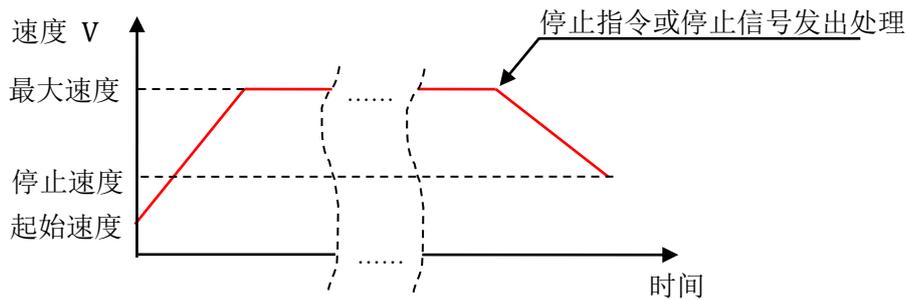


图2.7 连续运动速度曲线

该功能的主要用途是：速度控制，如：传送带的速度、包装机连续送料速度等。

### 2.3.3 回原点运动

在运动平台上，每个轴都有一个位置传感器用于设置一个位置参考点，即原点位置，以便进行位置控制。在正常运动之前，都需要用回零指令控制平台向原点方向运动，当控制器检测到原点信号 ORG 后，平台自动停止，并将停止位置作为该轴的原点。参见图 2.8。

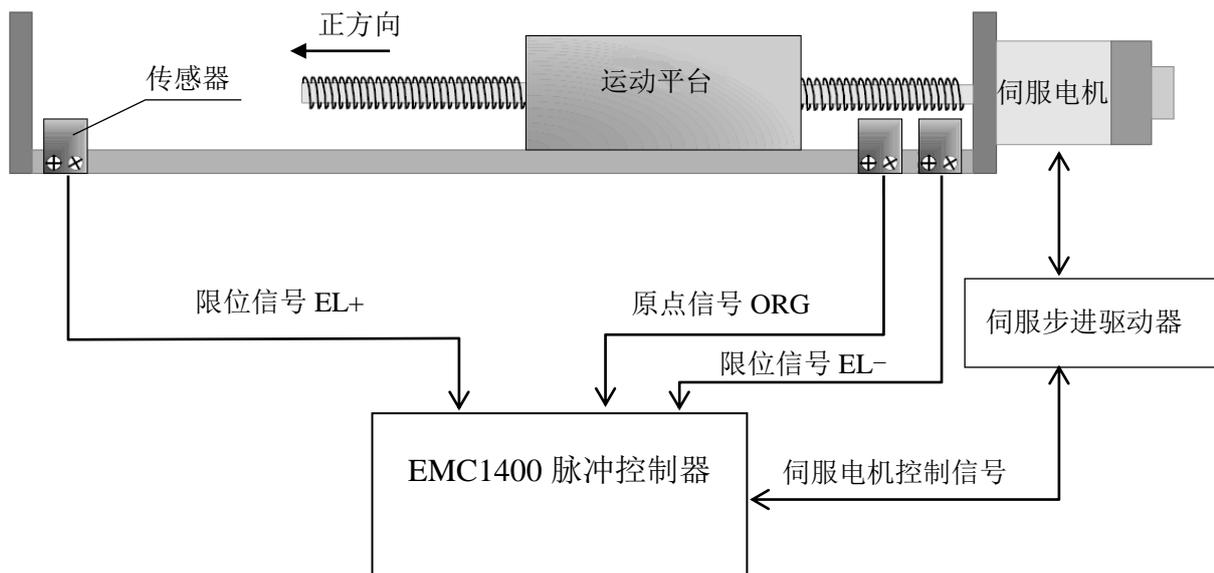


图 2.8 运动平台传感器信号及电机控制信号与伺服步进驱动器的关系

## 2.4 通用 IO 控制

EMC1400 脉冲控制器除了运动控制功能外，还提供了数字式输入信号（Input）和输出信号（Output），即 IO 信号的控制功能。

### 2.4.1 通用 IO 信号

在电机运动的同时，EMC1400 脉冲控制器还可接受按键、数字式传感器等信号，控制指示灯、继电器、电磁阀等器件。

### 2.4.2 背板 IO 扩展功能

EMC1400 脉冲运动控制器支持背板总线扩展，最大可支持扩展 8 个模块。

## 2.5 EMC 控制器组网

EMC1400 脉冲控制器通过 Ethernet 网口和 PC 进行通讯，可同时支持 254 控制器进行组网。使用方便简洁，可极大的为用户节省工控机的成本。用户只需将每个控制器的地址 IP 设置不相同，链接号进行区分。即可精准的操作每个控制器。

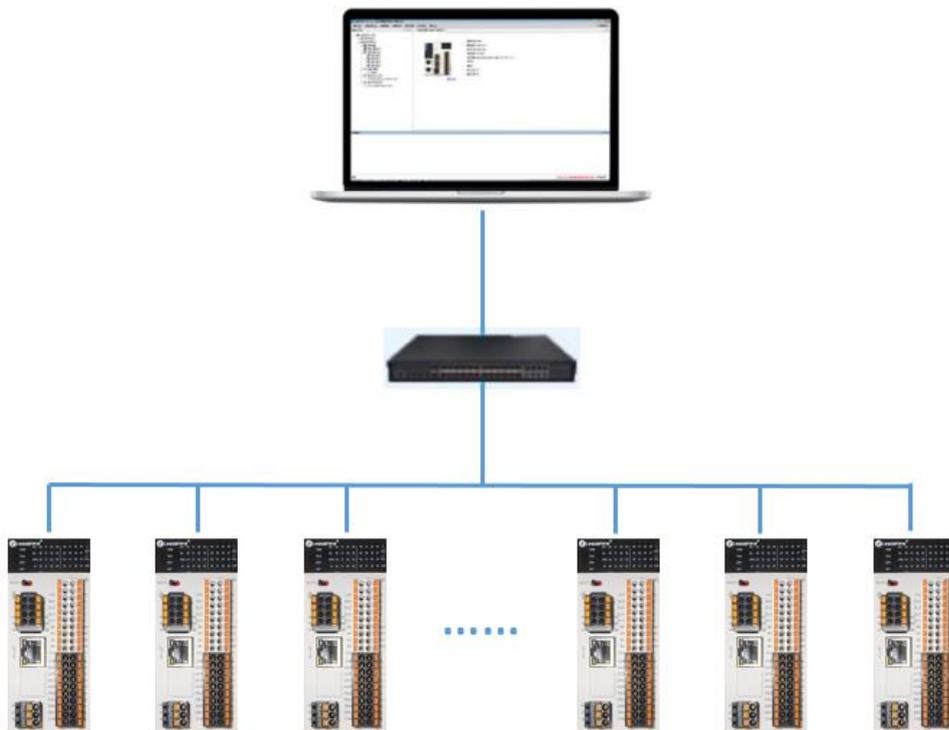


图 2.12 EMC 控制器组网示意图

## 第 3 章 硬件接口电路

### 3.1 接口布局

本章节主要介绍 EMC1400 运动控制器的外围接口以及相应的连接方式。用户可参照本章节内容，完成控制器与外部执行单元（电机驱动器、开关器件等）的电气连接部分工作

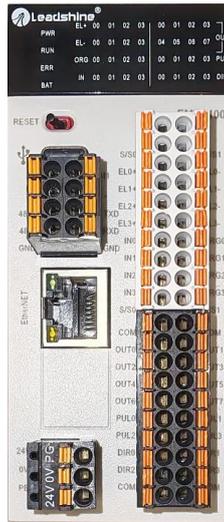


图3.1 EMC1400运动控制器实物图

EMC1400 脉冲控制器硬件布置及尺寸如图 3.2 所示。

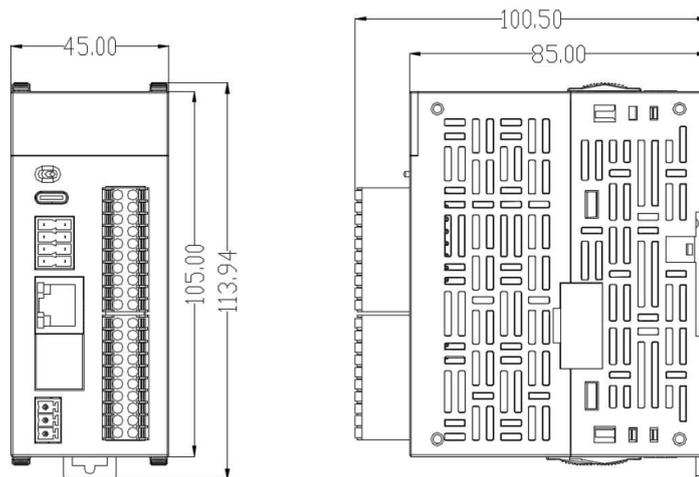


图 3.2 EMC1400 脉冲控制器结构尺寸图

### 3.2 接口及引脚定义

EMC1400 脉冲控制器的接口及脚位如表 3.1 所示。

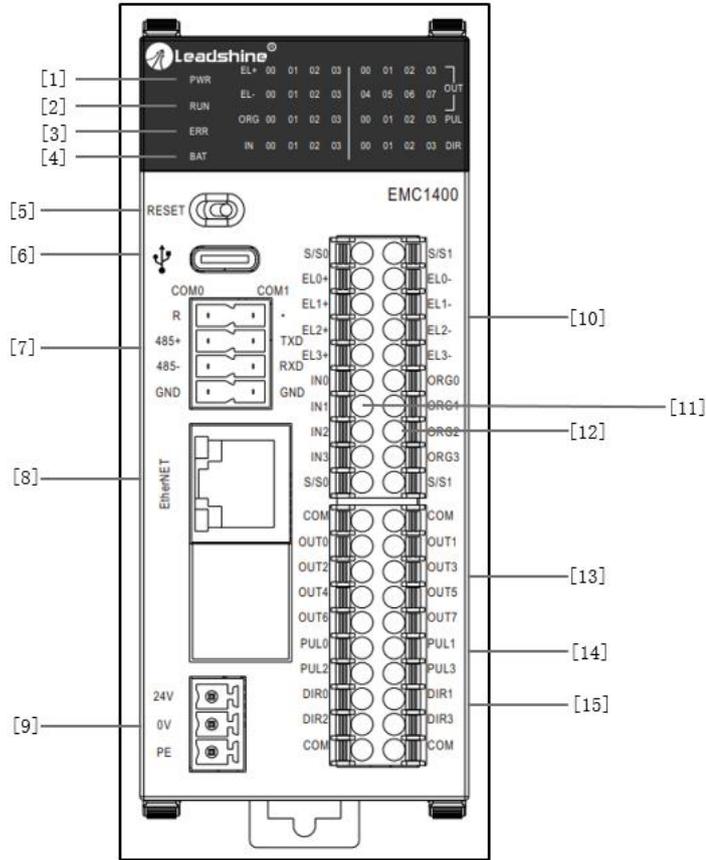


图 3.3 EMC1400 脉冲控制器引脚分布图

表 3.1 脚位定义

名称	功能介绍
1	PWR 指示灯
2	RUN 指示灯
3	ERR 指示灯
4	BAT 指示灯
5	RESET 拨码
6	type C 接口
7	RS485/232 接口
8	Ethernet 网口

9	电源接口
10	限位信号接口
11	通用输入 IO
12	原点信号接口
13	通用输出
14	脉冲信号输出接口
15	方向信号输出接口

### 引脚定义描述：

- 1) **POWER**：用于表示电源状态，电源正常：亮，异常：灭；
  - 2) **RUN**：用于系统是否正常运行指示；系统正常运行：**RUN** 灯闪烁；系统异常：**RUN** 灯常亮或者常灭
  - 3) **ERR**：表示当前报警状态；正常时：常灭；报警状态下：常亮；
  - 4) **BAT**：表示电池供电错误状态；供电不足，需要替换：常亮；电池供电正常：常灭
  - 5) **RESET 拨码**：用于实现恢复出厂设置；恢复出厂设置功能可通过多次切换 **RESET** 拨码状态，实现 5 秒内拨动次数超过 5 次则认为触发恢复出厂设置，触发后，需将控制器断电即能恢复到出厂设置；恢复出厂设置效果：恢复出厂设置参数和状态，配置文件可删除，成出厂默认配置；
  - 6) **type C 接口**：**Type C** 接口用于 **EMC1400** 运动控制器和上位机连接，主要用于查看控制器 **IP** 等相关状态的查询显示，不建议用于正常运行调试时的通讯。
  - 7) **RS485/232 接口**：**EMC1400** 多轴运动控制器支持 1 路 **RS485/232** 接口，支持 **modbus RTU** 及自由通讯协议；
  - 8) **Ethernet 网口**：用于和 **PC** 的通讯；
  - 9) 电源接口
  - 10) 限位信号接口：0-3 轴正负硬限位输入接口；输入有效：指示灯亮；输入无效：指示灯灭；
  - 11) 通用输入 **IO**：输入有效：指示灯亮；输入无效：指示灯灭；
  - 12) 原点信号接口：0-3 轴原点输入接口；输入有效：指示灯亮；输入无效：指示灯灭；
  - 13) 通用输出 **IO**：输入有效：指示灯亮；输入无效：指示灯灭；
  - 14) 脉冲信号输出接口：0-3 轴脉冲输出引脚；
  - 15) 方向信号输出接口：0-3 轴方向输出引脚；
- EMC1400** 脉冲输出为单端信号。

### 3.3 通用 I/O 接口电路

#### 3.3.1 通用数字输入信号接口

EMC1400 脉冲控制器本体有 4 路通用数字输入信号。所有输入接口均加有光电隔离元件，可以有效隔离外部电路的干扰，以提高系统的可靠性。通用数字输入信号接口原理图如图 3.4 所示。

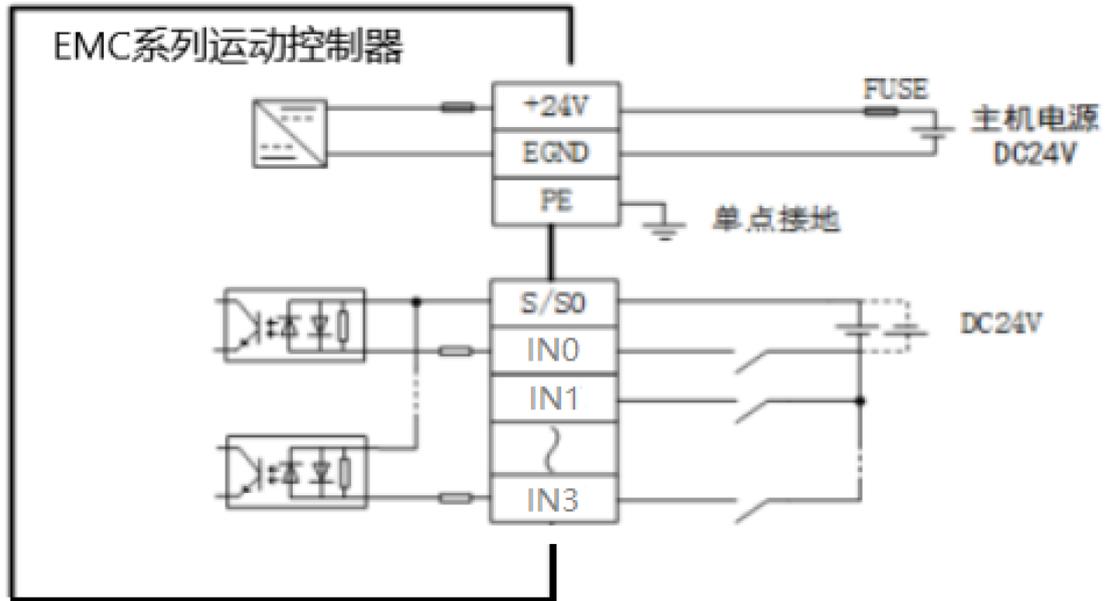


图 3.4 通用输入信号接口原理图

#### 3.3.2 通用数字输出信号接口

EMC1400 脉冲控制器有 8 路通用数字输出信号，由 MOS 管驱动，其最大工作电流为 500 mA（5~24Vdc，吸入），可用于控制继电器、电磁阀、信号灯或其它设备。

下面给出了通用数字输出信号接口控制几种常用元器件的接线图。

##### 1、发光二极管

通用数字输出端口控制发光二极管时，需要接一限流电阻  $R$ ，限制电流在 10mA 左右，电阻需根据使用的电源来选择，电压越高，使用的电阻值越大。接线图如图 3.5 所示。

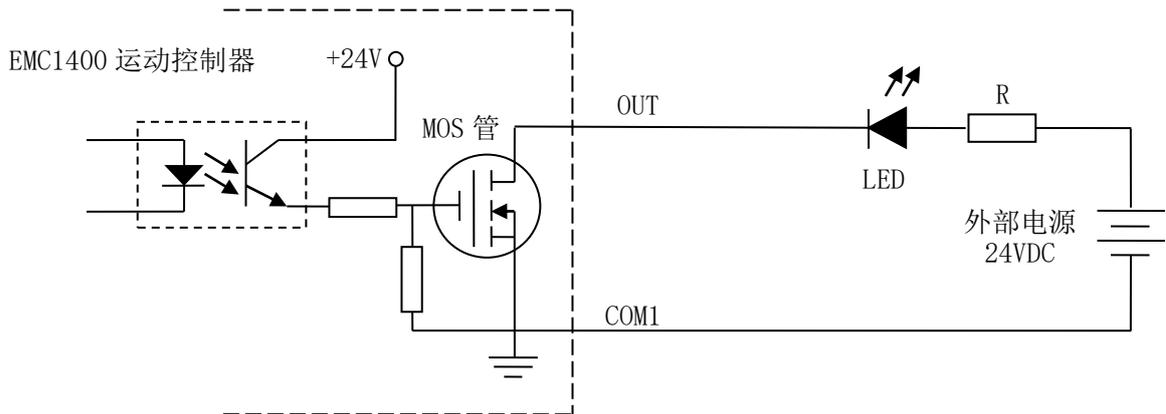


图 3.5 EMC1400 脉冲控制器输出口接发光二极管接线图

## 2、灯丝型指示灯

通用数字输出端口控制灯丝型指示灯时，为提高指示灯的寿命，需要接预热电阻  $R$ ，电阻值的大小，以电阻接上后，输出口为 1 时，灯不亮为原则。接线图如图 3.6 所示。

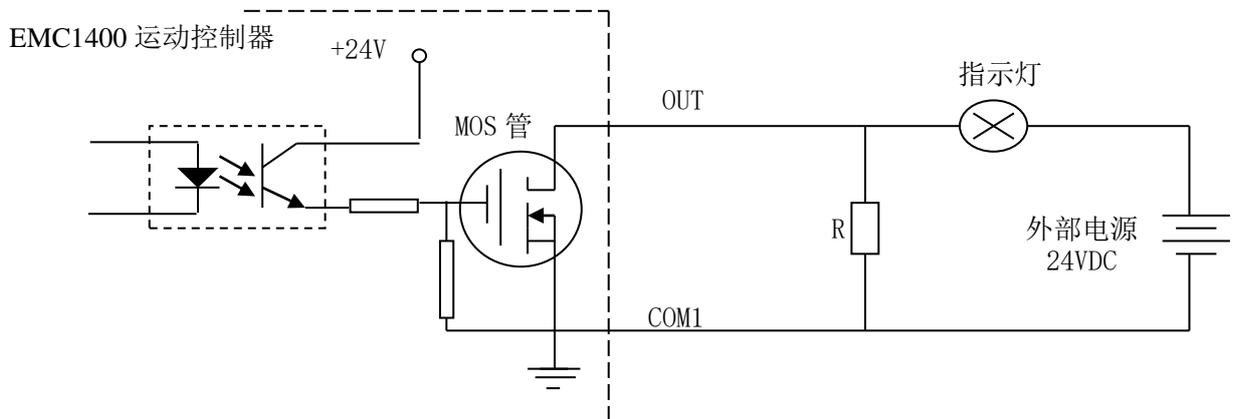


图 3.6 EMC1400 脉冲控制器灯丝型指示灯接线图

## 3、小型继电器

继电器为感性负载，必须并联一个续流二极管。当继电器突然关断时，继电器中的电感线圈产生的感应电动势可由续流二极管消耗，以免 ULN2803 或 MOS 管被感应电动势击穿。其接线图如图 3.7 所示。

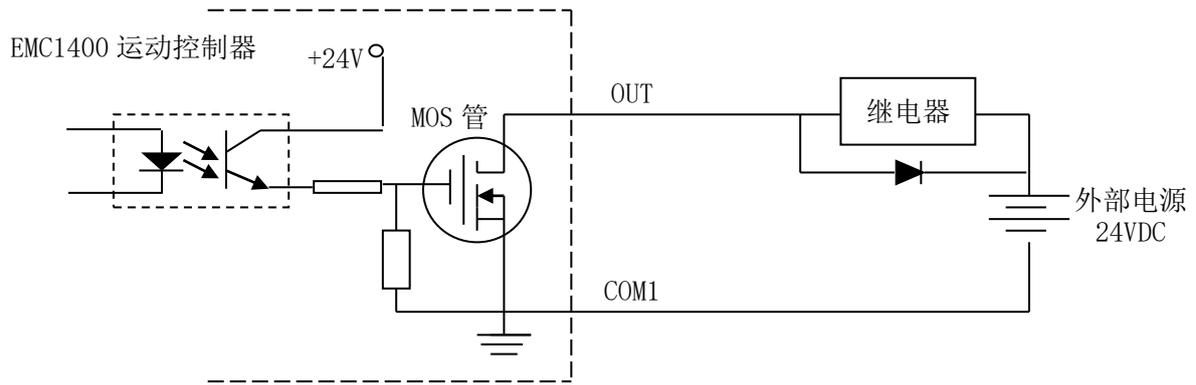


图 3.7 EMC1400 脉冲控制器接小型继电器接线图

注意：在使用通用数字输出端口时，切勿把外部电源直接连接至通用数字输出端口上，否则会损坏输出口。



◆ 驱动器、电机、控制器等运动控制系统均属于工业废弃物，请根据当地法律法规按照工业废弃物处理标准进行处理回收，避免污染环境。

◆ 请按当地法律法规处理废旧电池，请勿将电池作为生活垃圾/办公垃圾混合处理。电池处置不当可能会导致环境污染或爆炸。

## 第 4 章 Motion 调试软件的使用方法

用户在使用 VB、VC 或其它高级语言编写应用程序之前，可利用 Motion 调试软件快速熟悉 EMC1400 脉冲控制器的硬件、软件功能，还可以方便快捷地测试电机、传感器、开关元件、平台等在执行各种动作时的性能特点。

### 4.1 Motion 调试软件的使用步骤

安装 Motion 调试软件的步骤如下：

- 1) 使用 Motion 调试软件前，请确保已经通读本手册。
- 2) 启动 PC 机，进入 Windows 操作系统。
- 3) 将 EMC1400 脉冲控制器资料包中的“Motion 调试软件”，全部复制到硬盘中。
- 4) 在确保控制器 IP 地址与 PC 机 IP 地址处于同一网段的条件下，复制到硬盘中的 Motion 调试软件目录中找到 Leadshine.DMC.IDE.exe，双击运行。

在 Motion 首次解压时，打开 Motion 后若没有自动连接上控制器，可点击功能选项，选择连接设置，在弹框中输入控制器地址，点击确定。之后点击刷新或者重新打开 Motion 都会自动与控制器通讯上。如图 4.1 和 4.2。

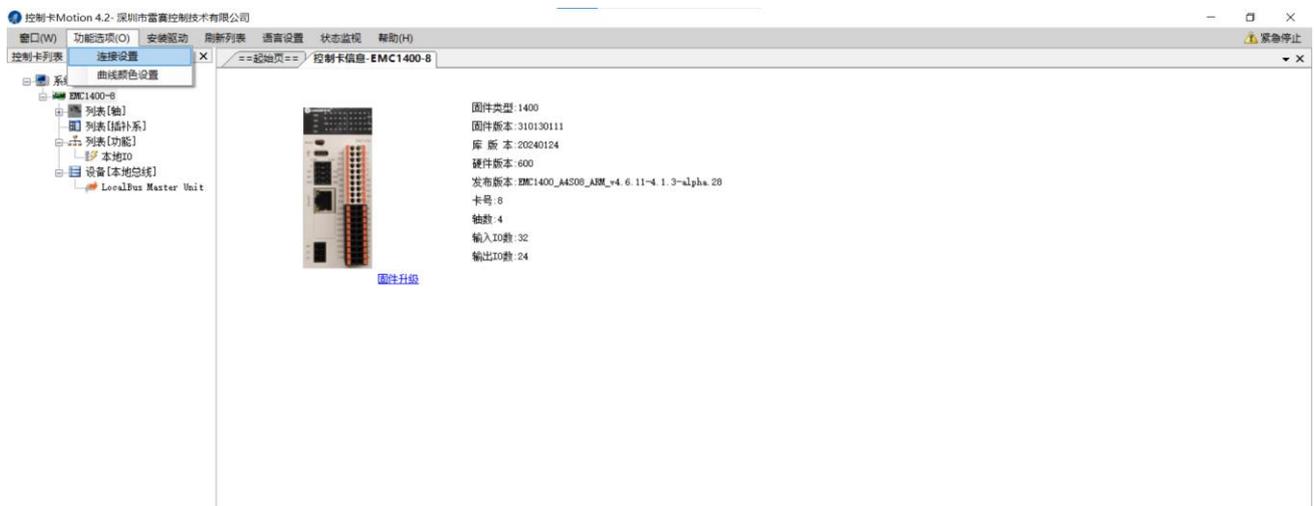


图 4.1 软件连接设置

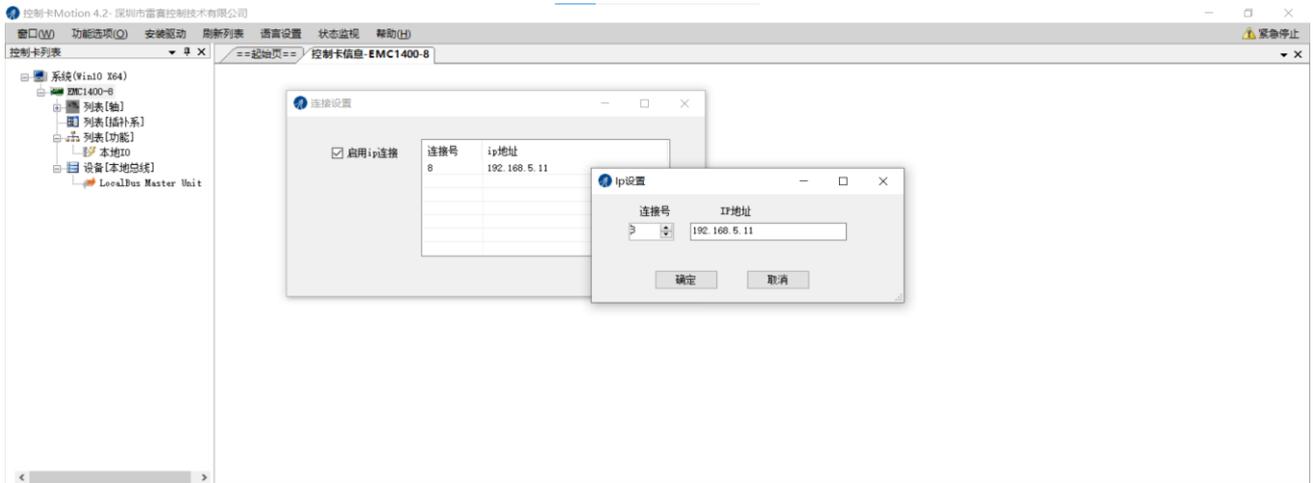


图 4.2 演示软件 IP 地址连接

通讯成功后便如图 4.3 所示的软件主界面。在此界面上可以通过点击“EMC1400-8”，查看控制器信息。

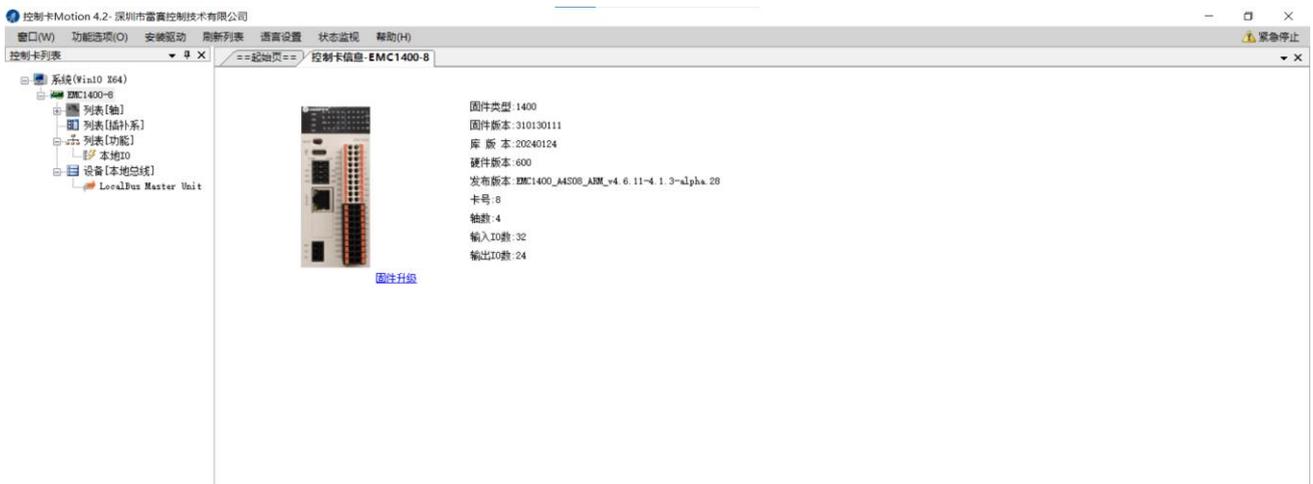


图 4.3 软件主界面

5) 选择所需的控制器，然后进行相应的操作。

6) 至此，即可使用 Motion 调试软件进行测试。

**注意：**本软件是基于 .NET 4.0 开发，请在使用软件前确保操作系统内已经安装了 .NET 4.0，若未安装请先安装。

## 4.2 Motion 软件的功能与使用方法

当使用 EMC1400 脉冲控制器时，Motion 调试软件提供了参数设置、IO 检测、运动测试、帮助这四个主要的操作界面。设置好界面上的参数后，就可以进行一些基本的控制操作，如：点位运动、I/O 信号检测等。

Motion 调试软件主要用于设备扫描配置以及功能测试使用，界面简单，操作方便。

软件主界面主要分为 5 个部分：

(1) 菜单栏，用于 Motion 软件的设置、刷新列表和紧急停止。

(2) 控制器列表窗口，控制器列表区的控制器列表是在软件启动时先扫描 PC 系统上已经连接的控制器并显示在列表上。轴列表会显示控制器上本体自带的脉冲轴，功能列表根据控制器的不同而有所不同，设备列表会显示已经加入模块类型。

(3) 功能页面窗口。

(4) 信息输出窗口，支持所有函数调用过程的信息输出。

(5) 状态栏，显示背板总线的状态，会根据控制器列表区选择的控制器类型不同而有所不同。主界面如下图：



图 4.4 主界面

各功能详细使用方法见 Motion 使用手册。

## 4.2.2 辅助功能

### 4.2.2.1 重启

当使用控制器出现异常情况时，可以通过复位来实现控制器重启。右键选择对应的控制器，即可出现复位控制器的菜单项。

复位分为两种：冷复位和热复位。

冷复位：重启控制器操作系统，控制器内部所有程序重启，复位速度较慢；

热复位：重启控制器应用程序，复位速度较快。

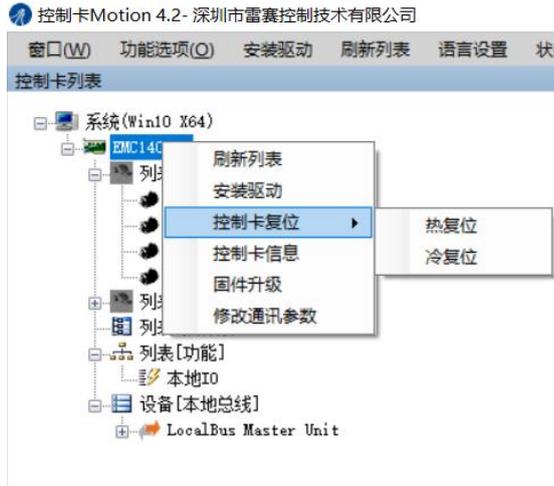


图 4.15 复位控制器菜单

**注意：**系统中增加或减少背板模块都要执行重新扫描配置并下载。

#### 4.2.2.2 控制器信息

当需要了解控制器相关信息时，可右键点击对应的控制器，选择“控制器信息”菜单项。



图 4.16 控制器信息菜单

#### 4.2.2.3 信息输出

Motion 调试软件支持所有函数调用过程的信息输出，在菜单栏“窗口”选择“函数输出窗口”即可显示输出窗口。

函数输出窗口中输出了函数的调用名称、参数以及返回值，当返回值为错误码时会以红色标记。

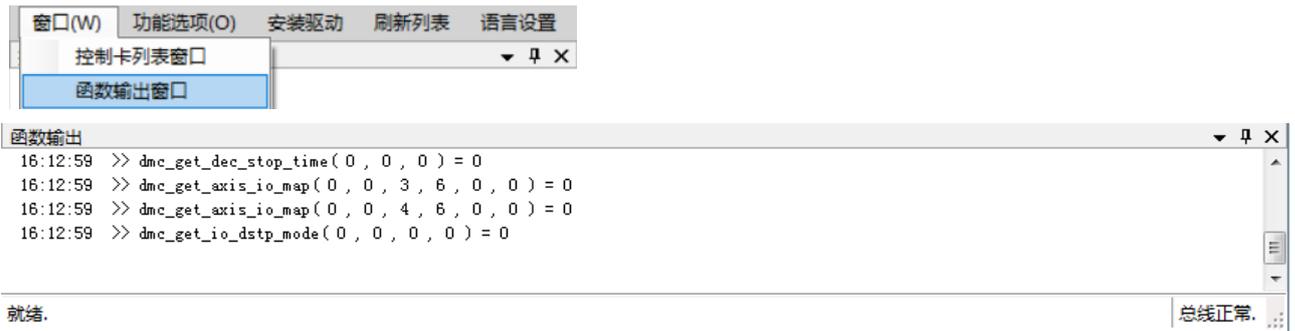


图 4.17 函数输出窗口

#### 4.2.2.4 错误码查询

在函数输出窗口右击，选择“错误码查询”菜单项，即可弹出错误码查询窗口，输入错误码，点击“查询”即可显示错误码对应的信息。

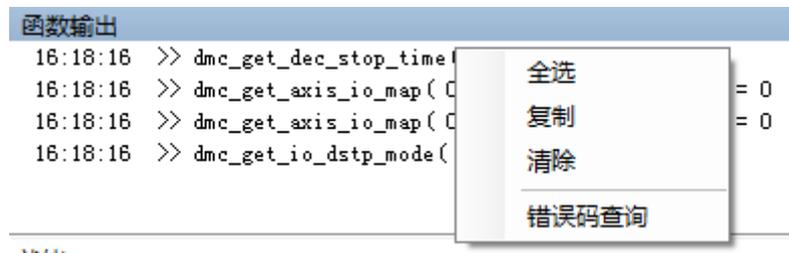


图 4.18 错误码查询菜单



图 4.19 错误码查询窗口

#### 4.2.3 参数配置

当使用控制器前，需要对控制器的轴运动、IO 信号以及伺服信号等相关参数进行合理配置。  
注意：不同型号控制器，请根据现场情况以及产品使用手册进行合理配置。

选择“控制器列表区”的控制器->展开“列表[轴]”列表->选择要操作的轴->双击“轴 0”按钮->单击“单轴参数”选项页，进入轴参数配置窗口。

在进入参数配置窗口后，可以通过点击“上传”按钮上传控制器内的参数到当前窗口显示，

点击“下载”按钮把当前界面的参数下载到控制器；当所有轴参数设置相同时，点击“应用到所有轴”按钮可以把当前轴设置的参数应用到所有轴。窗口打开时默认会上传控制内最新的参数。

注意：不同类型控制器参数项可能有所不同



图 4.20 参数配置窗口

#### 4.2.3.1 轴参数

##### 1) 基本设置

设置各个轴的基本运动参数，包括脉冲当量、初始速度、定位速度、终止速度、加减速时间、S 段时间等参数。

参数名	参数值	单位	说明
<b>基本设置</b>	<input checked="" type="checkbox"/> 导出		
脉冲模式	脉冲高+方向高		脉冲模式: 0-脉冲高+方向高; 1-脉冲低+方向高; 2-脉冲高+...
初始速度	0	unit/s	最大值为 2M
定位速度	0	unit/s	最大值为 2M
终止速度	0	unit/s	最大值为 2M
加速时间	0	s	最小值为 0.001s
减速时间	0	s	最小值为 0.001s
S段时间	0	s	范围: 0~0.5 s
减速停止时间	0	s	异常减速停止时间 范围: 0~∞

图4.21 基本参数设置

**初始速度：**设置单轴运动时，轴运动开始的起步速度，单位为 pluse/s。

**定位速度：**设置单轴运动时，轴运动的最大速度，单位为 pluse/s。

**终止速度：**设置单轴运动时，轴运动结束时的停止速度，单位为 pluse/s。

**加速时间：**设置单轴运动时，轴运动由初始速度加速到最大速度时所需要的时间，单位为 S。

**减速时间：**设置单轴运动时，轴运动快结束时由最大速度降速到终止速度时所需要的时间，单位为 s。

**S 段时间：**设置单轴运动时，轴 S 形曲线速度的时间，单位为 s。

减速停止时间：设置单轴运动时，调用减速停止指令或遇到异常信号减速后，减速停止的时间，单位 s。

## 2) 回零设置

设置各个轴的回零参数，回零参数包括回零低速、回零高速、加减速时间、回零模式。

回零设置			
回零低速 (unit/s)	10		
回零高速 (unit/s)	100		
回零偏移 (unit)	0		
加速时间 (s)	0.1		
减速时间 (s)	0.1		
回零模式	0		

图4.22 回零参数设置

回零低速：回零时低速运动时的速度。

回零高速：回零时高速运动时的速度。

加速时间：回零时由低速到高速的加速段时间。

减速时间：回零时由高速到低速的减速段时间。

回零模式：设置回零的回零方式。

## 4.2.4 功能测试

测试控制器各项功能，包括单轴测试，通用 IO 控制等。注意：不同类型的控制器可能测试项不同。

### 4.2.4.1 单轴测试

测试各个轴的定长、定速以及回零运动，支持在线变速运动，监视各个轴的运动状态以及轴状态信息。



图4.27 单轴测试界面

一般操作步骤:

- 1、设置模式，设置好运动模式及其他参数，注意：不同运动模式参数项不一样。
- 2、设置参数。
- 3、设置曲线显示，勾选需要采集的曲线进行曲线显示。
- 4、点击“启动”按钮，即开始运动。

#### 1) 模式设置

运动模式：选择测试运动的模式，包括定长运动、连续运动以及回零运动。

停止模式：设置点击“停止”按钮时，停止是减速停止还是立即停止。

位置类型：设置参数项中的目标位置是绝对定位还是相对定位。

#### 2) 参数项

运动轴选择：勾选需要参数运动的轴。

起始速度：定位运动起始的速度。

最大速度：定位运动达到的最大运动速度。

终止速度：定位运动结束运动时的速度。

加速时间：定位运动从起始速度到最大速度的加速段时间。

减速时间：定位运动从最大速度到停止速度的减速段时间。

S 段时间：S 曲线速度段的时间。

目标位置：设置定位运动的运动到的目标位置，若位置坐标类型设置为“相对坐标”则该值为相对于当前位置的相对距离。

运动方向：设置“相对坐标”模式下，相对定位方向。

回零低速：回零运动低速时的速度。

回零高速：回零运动高速时的速度。

回零方向：回零运动的方向，正方向/负方向。

回零模式：设置回零的模式，具体模式请参照对应型号的控制手册。

回零速度模式：设置回零时低速回零还是高速回零。

#### 3) 图形显示

位置-时间曲线：采集显示位置与时间关系曲线。

速度-时间曲线：采集显示速度与时间关系曲线。

#### 4) 操作

启动：启动选择的轴运动。

单轴停止：停止选择的轴。

紧急停止：紧急停止所有的轴运动。

计数器清零：设置所有轴指令位置和反馈位置为 0。

在线变速：在轴定位运动时改变选择的轴的运动速度

## 第 5 章 应用软件开发方法

EMC1400 脉冲控制器的应用软件可以在 Visual Basic 、 Visual C++、 C#等高级语言环境下开发。

如果您对 VB、 VC、 C#语言都不熟悉，建议您花两天时间阅读一本 VB 语言的培训教材，并且通过练习掌握该语言的基本技巧，如：编写简单的程序、创建窗体和调用函数。

如果您曾用 VB 或 VC 等程序语言开发过运动控制软件，并具有丰富的经验，则可直接阅读第 7 章“基本功能函数说明”。

### 5.1 基于 WINDOWS 平台的应用软件结构

使用雷赛控制器的自动化设备运动控制系统构架如图 5.1 所示：

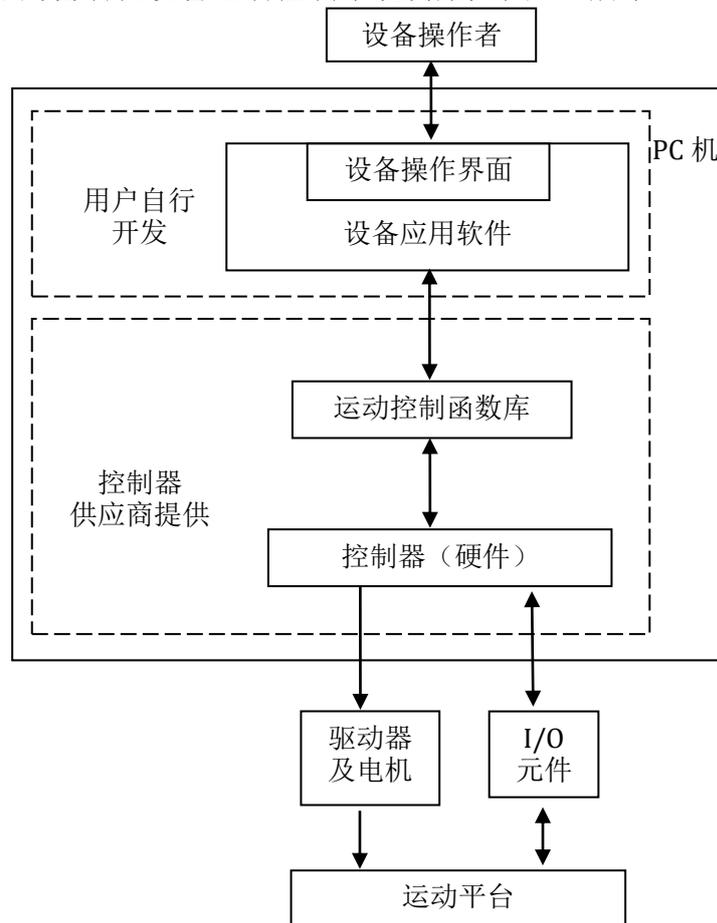


图 5.1 基于雷赛控制器的自动化设备运动控制系统构架

从图 5.1 中可看出，运动控制系统的工作原理可以简单描述为：

- 1) 操作员通过操作界面（包括显示屏和键盘）将指令信息传递给设备应用软件；
- 2) 设备应用软件将操作者的信息以及应用软件中已有的运动流程、运动轨迹等数据转化

为运动参数，并根据这些参数调用 DLL 库中运动函数；

3) 运动函数通过工控机向控制器发出控制指令；

4) 控制器根据控制指令发出相应的指令脉冲给驱动器及电机、读写通用输入输出数据。

用户根据设备的工艺流程、运动轨迹和友好的人机界面等要求开发设备应用软件。雷赛公司已提供支持 EMC1400 脉冲控制器的 DLL 运动函数库，包括控制器初始化、单轴控制、数字量输入/输出控制等多种函数。这些函数可以方便地完成与运动控制相关的功能，用户不需要更多了解硬件电路的细节以及运动控制细节，就能使用 VB、VC 等程序语言开发出自己的运动控制系统应用软件。

用户编写的设备应用软件的典型流程如图 5.2 所示。

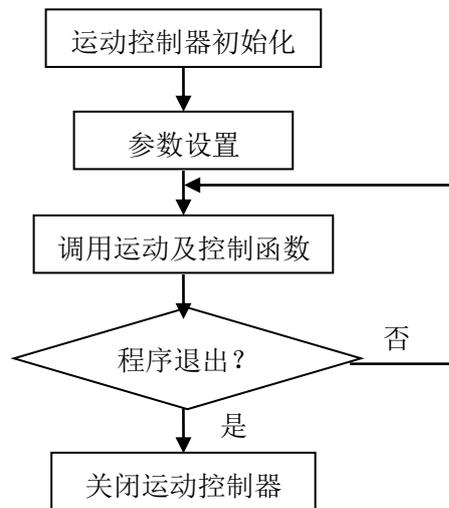


图 5.2 设备应用软件的典型流程

## 5.2 采用 VB 6.0 开发应用软件的方法

下面以 Visual Basic6.0 环境下编写一个点位运动的应用软件为例，讲解用 VB 开发应用软件的一般方法。

1) Motion 软件中，扫描驱动器，并将轴使能。在磁盘上新建一个目录，如 E:\test1

2) 打开 Visual Basic 6.0, 新建一个“标注 EXE”工程，在对话框上添加按钮“启动”和“停止”，并将其名称分别修改为“CB\_Start”和“CB\_Stop”，如图 5.3 所示。

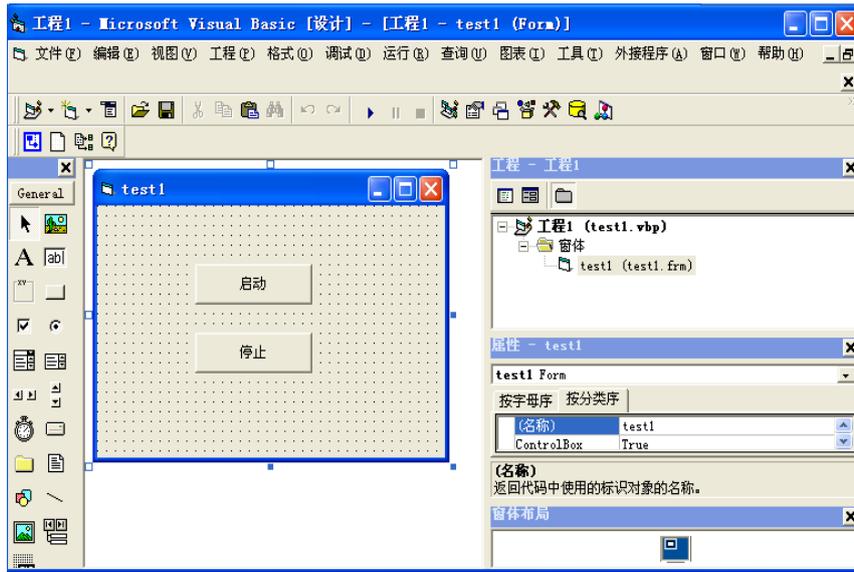


图 5.3 修改对话框（VB）

3) 工程保存在 E:\test1 目录下。

4) 在资料光盘相应目录下找到 LTDMC.bas、LTDMC.dll 和 PVT.dll 文件，拷贝到 test1 目录下。

5) 菜单中选择“工程”->“添加模块”->“现存”，找到 test1 目录下的 LTDMC.bas 文件，添加到工程中，如图 5.4 所示。

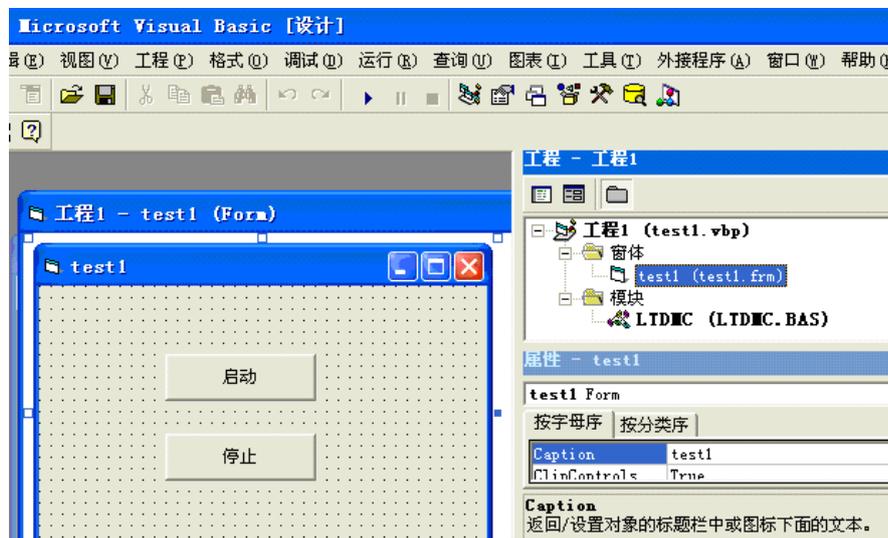


图 5.4 添加头文件

6) 双击窗口控件，在 Form\_Load 事件中添加代码 dmc\_board\_init\_eth。选择 UnLoad 事件，在 Form\_Unload 事件中添加代码 dmc\_board\_close 双击“启动”按钮，在 CB\_Start\_Click 事件中添加代码如下：

```
dmc_set_profile 0,0,500,5000, 0.01,0.01,500
dmc_pmove 0,0,200000,0
```

双击“停止”按钮，在 CB\_Stop\_Click()事件中添加代码如下：

```
dmc_stop 0,0,0
```

7) 程序编写完成。运行程序，显示界面如图 5.5 所示。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。



图 5.5 程序运行界面（VB）

### 5.3 采用 VC 6.0 开发应用软件的方法

下面以 Visual C++ 6.0 环境下编写一个点位运动的应用软件为例，讲解用 VC 开发应用软件的一般方法。

- 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 5.6。

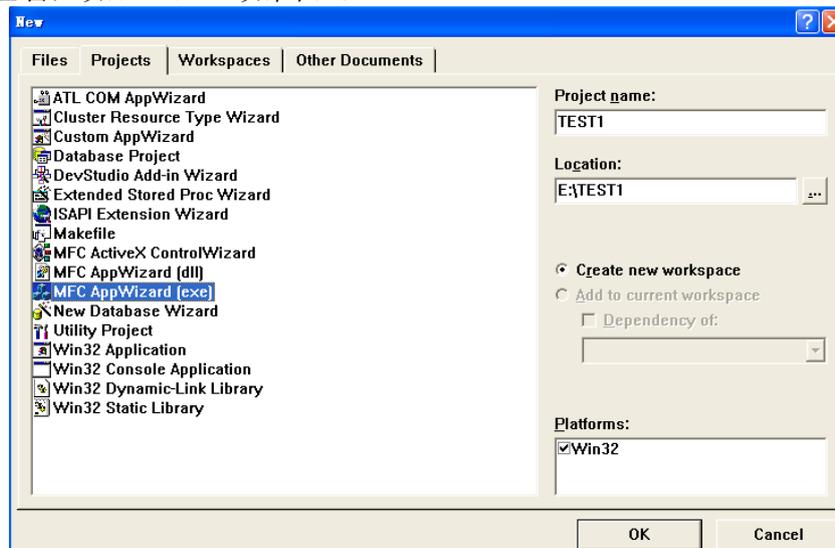


图 5.6 创建新工程

6) 在应用程序类型中选择“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”、“停止”和“使能”；并分别命名为“IDC\_BUTTON\_Start”、“IDC\_BUTTON\_Stop”和“IDC\_BUTTON\_enable”，如图 5.7 所示。

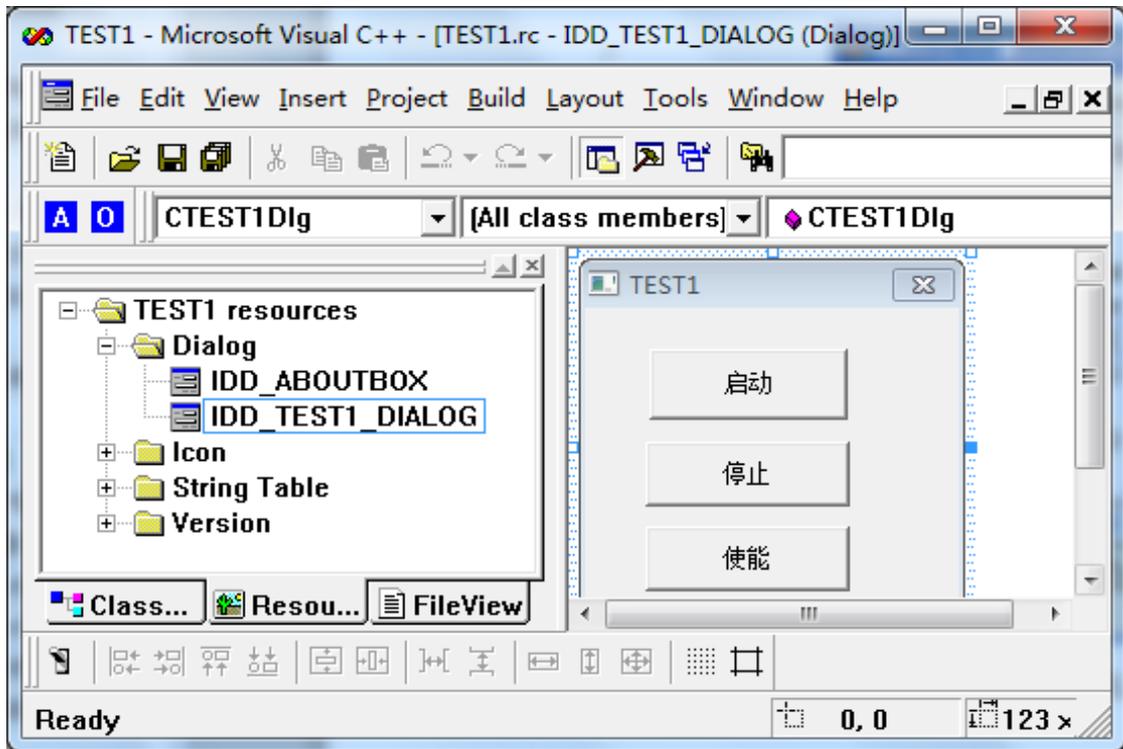


图 5.7 修改对话框

8) 在相应的目录下找到 LTDMC.h、LTDMC.lib、LTDMC.dll，拷贝到 E:\tes1 目录。

9) 在菜单中选择“工程”->“添加工程”->“文件”，选中 LTDMC.lib 文件加入到工程中。

10) 打开 test1.cpp 文件，在程序开始部分添加相应语句：`#include "LTDMC.h"`，如图 5.8 所示。

11) 在 `CTest1Dlg::OnInitDialog()` 函数中添加代码：`dmc_board_init_eth()`；如图 5.9。

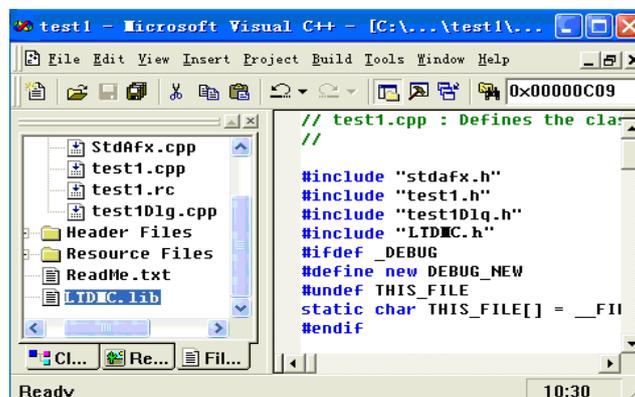


图 5.8 程序增加头文件

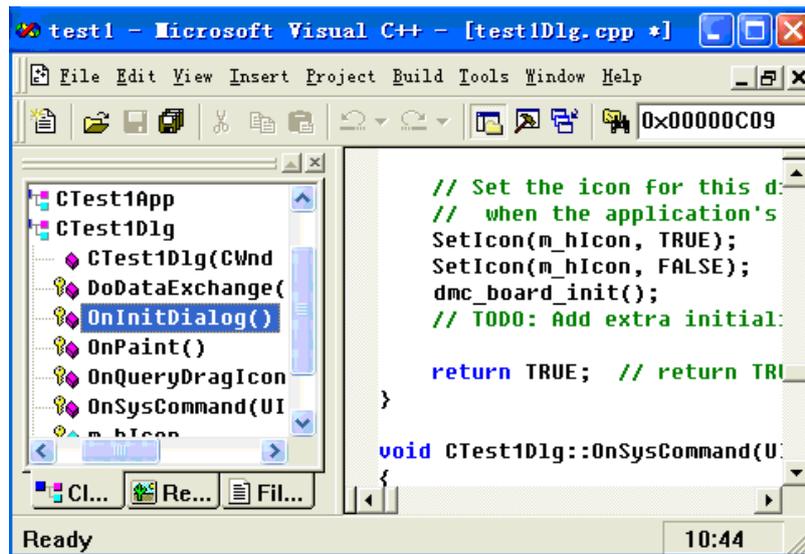


图 5.9 程序增加初始化函数

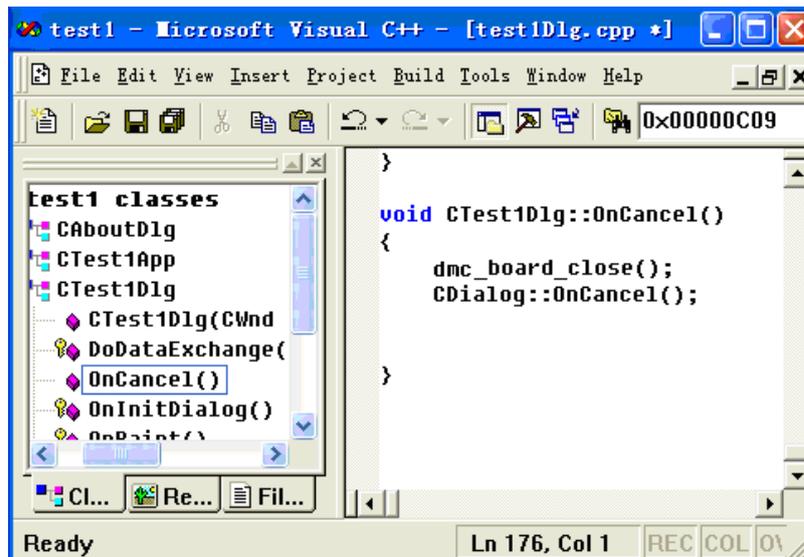


图 5.10 程序增加 OnCancel 函数

12) 如图 5.10 所示，在 Ctest1Dlg 中添加一个成员函数 OnCancel,在 OnCancel 函数中添加代码如下：

```

dmc_board_close();
CDialog::OnCancel();

```

13) 双击“启动”按钮在按钮点击事件中输入代码如下：

```

dmc_set_profile(0,0,500,5000, 0.01,0.01,500);
dmc_pmove(0,0,200000,0);

```

双击“停止”按钮在按钮点击事件中输入代码：

```
dmc_stop(0,0,0);
```

双击“使能”按钮在按钮点击事件中输入代码：

```
nmc_set_axis_enable(0,0);
```

14) 编译程序后，运行程序，显示图 5.11 所示的界面。按下“使能”，再按下“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。



图 5.11 程序运行界面

## 5.4 采用 C# 开发应用软件的方法

下面以 C# 环境下编写一个点位运动的应用软件为例，讲解用 C# 开发应用软件的一般方法。

1) 在磁盘上新建一个目录，如 E:\Test

2) 打开 C#，新建一个“Windows 窗体应用程序”，并将名称修改为“test1”，如图 5.12 所示。在对话框上添加按钮“启动”和“停止”，并将其名称分别修改为“Start”和“Stop”，如图 5.13 所示。

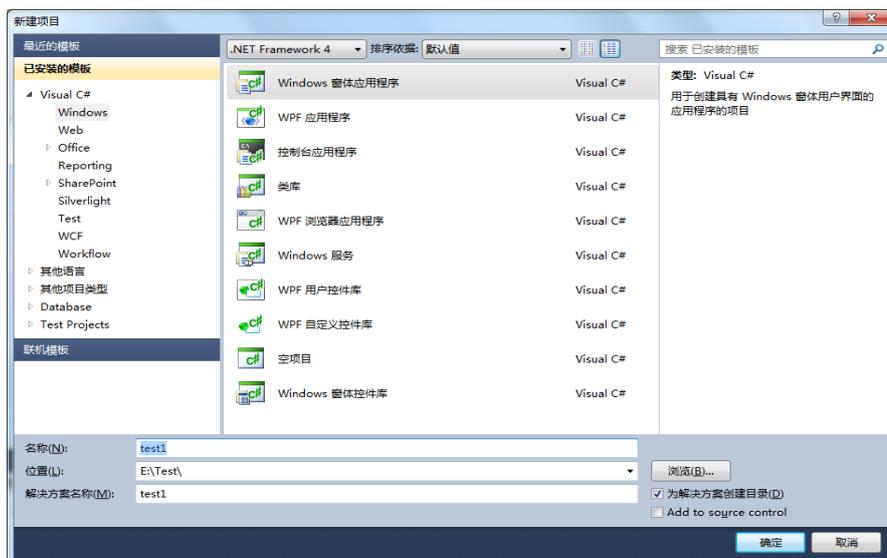


图 5.12 新建 Windows 窗体应用程序

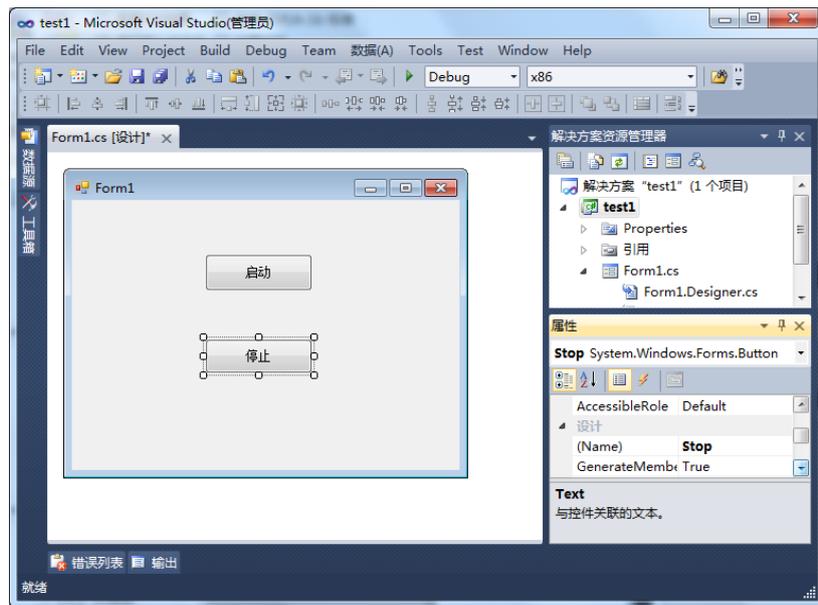


图 5.13 修改对话框

- 3) 点击“全部保存”，将应用程序保存在 E:\Test 目录下。
- 4) 在资料光盘相应目录下找到 LTDMC.dll 和 LTDMC.cs 文件，将 LTDMC.dll 文件拷贝至 E:\Test\test1\test1\bin\debug 目录下，LTDMC.cs 文件拷贝至 E:\Test\test1\test1 目录下。
- 5) 菜单中选择“项目”->“添加现有项”，找到 test1 目录下的 LTDMC.cs 文件，添加到应用程序中，如图 5.14 所示。

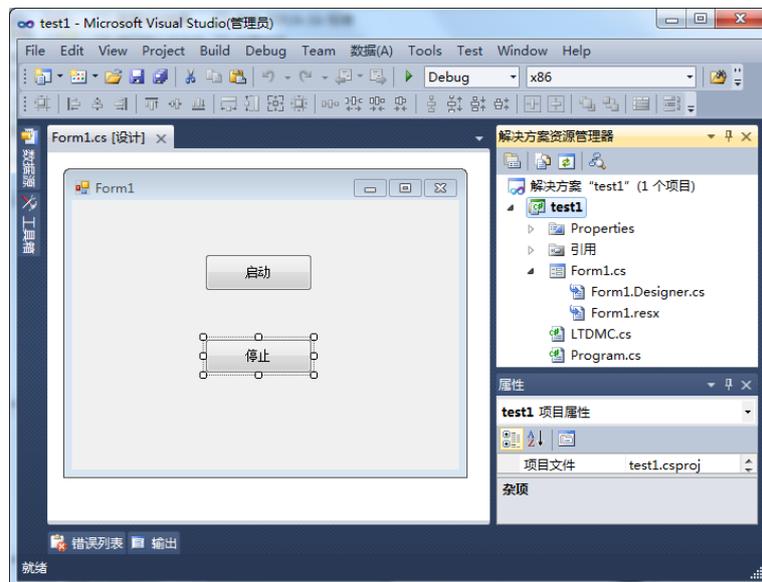


图 5.14 添加头文件

6) 在代码文件开头处添加控制器的命名空间：`using csLTDMC`；如图 5.15 所示。

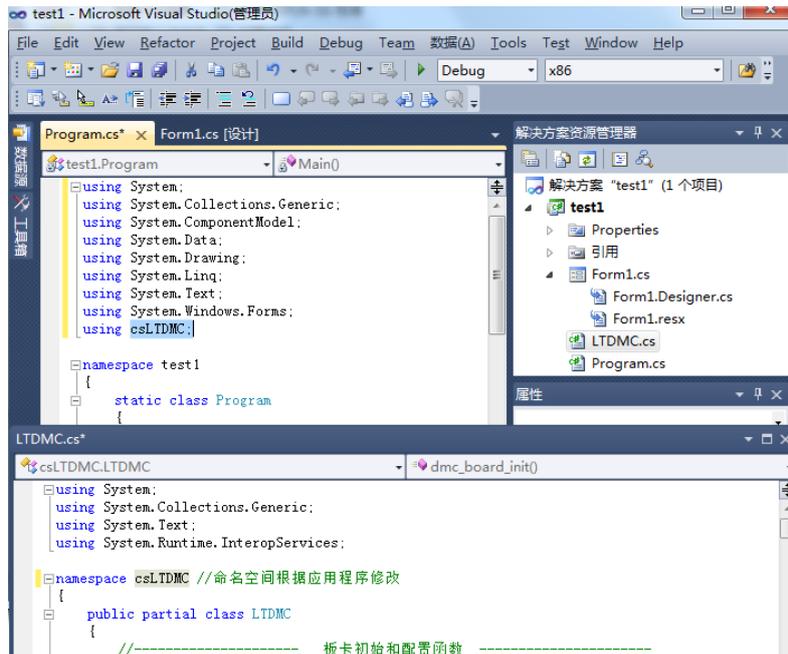


图 5.15 添加控制器的命名空间

7) 设定控制器链接号为 3，如图 5.16 所示，双击窗口控件，在 `Form1_Load` 事件中添加代码 `LTDMC.dmc_board_init_eth()`；双击属性窗口中 `FormClosed`，在 `Form1_FormClosed` 事件中添加代码 `LTDMC.dmc_board_close()`；双击“启动”按钮，在 `Start_Click` 事件中添加代码如下：

```
LTDMC.nmc_set_axis_enable(3, 0);
LTDMC.dmc_set_profile(3, 0, 500, 5000, 0.01, 0.01, 500);
LTDMC.dmc_pmove(3, 0, 200000, 0);
```

双击“停止”按钮，在 `Stop_Click` 事件中添加代码如下：

```
LTDMC.dmc_stop (3,0,0);
```

8) 程序编写完成。运行程序，显示界面如图 5.17 所示。按下“启动”按钮，第 0 轴就会走长度为 200000 的距离；运动中可以按下“停止”按钮，便会减速停止脉冲输出。

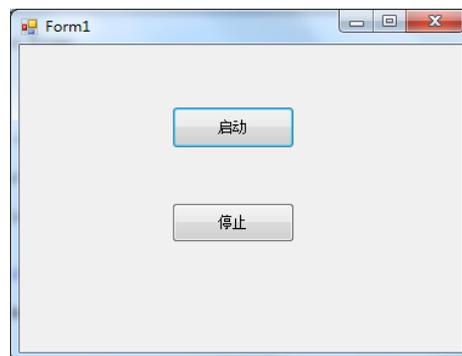


图 5.16 程序运行界面

## 第 6 章 雷赛控制 EMC1400 脉冲控制器基本功能实现方法

本章介绍采用 C# 语言通过调用相关函数实现 EMC1400 脉冲控制器的基本功能方法。

**注意：**在编程之前，一定要用 Motion 调试软件检测硬件系统，确保硬件接线正确。

### 6.1 控制器初始化

控制器初始化设置相关函数如下表 6.1 所示。

表 6.1 控制器初始化相关函数说明

名称	功能	参考
dmc_board_init_eth	控制器初始化函数	7.1 节
dmc_board_close	控制器关闭函数	

在操作控制器之前，必须调用函数 `dmc_board_init_eth` 为控制器分配资源。同样，当程序结束对控制器的操作时，必须调用函数 `dmc_board_close` 释放控制器所占用的 PC 系统资源，使得所占资源可被其它设备使用。

### 6.2 脉冲输出模式设置

EMC1400 脉冲控制器支持 7 种脉冲输出模式设置功能。相关函数如表 6.2 所示。

表 6.2 脉冲当量相关函数说明

名称	功能	参考
dmc_set_pluse_outmode	设置脉冲输出模式	7.2 节
dmc_get_pluse_outmode	读取脉冲输出模式	

脉冲输出模式共有 7 种可供选择，相关代码如下：

```

.....
ushort MyCardNo, MyAxis;
double MyEquiv;
MyCardNo=0; //控制器链接号
MyAxis=0; //轴号
MyEquiv=0; //设置脉冲模式为 0
LTDMC.dmc_set_pluse_outmode(MyCardNo, MyAxis, MyEquiv); //设置脉冲模式
.....
    
```

## 6.3 停止命令的设置

在设备进行运动功能调试之前，必须确保安全机制的有效。

停止开关在运动过程中出现意外的运动时，能起到紧急停止运动的功能，提高设备运行时的安全性能。在使用控制器进行运动控制之前，必须保证停止命令的有效性。

EMC1400 脉冲控制器提供了急停设置函数 `dmc_stop` 来设置急停功能。

相关函数如下表 6.3 所示。

表 6.3 急停开关设置相关函数说明

名称	功能	参考
<code>dmc_stop</code>	单轴急停或减速停	7.5 节

例：紧急停止前 4 轴

```

.....
ushort CardNo,Axis;
CardNo=0;           //控制器链接号
for(Axis=0;Axis<3;Axis++)//循环，依次对 0~3 号轴进行设置
{
    LTDMC.dmc_stop(CardNo, Axis, 1); //紧急停止轴
}
    
```

## 6.4 回原点运动的实现

### 6.4.1 回原点步骤

在进行精确的运动控制之前，需要设定运动坐标系的原点。运动平台上都设有原点传感器（也称为原点开关）。寻找原点开关的位置并将该位置设为平台的坐标原点的过程即为回原点运动。EMC1400 控制器共提供三种回原点方式。

回原点运动主要步骤如下：

- 1) 使用 `dmc_set_home_pin_logic` 函数设置原点开关的有效电平；
- 2) 使用 `dmc_set_homemode` 函数设置回原点方式；
- 3) 使用 `dmc_set_profile` 函数设置回原点运动的速度曲线；
- 4) 使用 `dmc_set_home_position` 函数设置回零偏移量、回零完成是否清零；
- 5) 使用 `dmc_home_move` 函数执行回原点运动；

## 6.4.2 回原点方式

EMC1400 控制器共提供三种回原点方式：

方式 0：一次原点回零

该方式以设定速度回原点；适合于行程短、安全性要求高的场合。动作过程为：电机从初始位置以恒定速度向原点方向运动，当到达原点开关位置，原点信号被触发，电机立即停止（过程 0）；将停止位置设为原点位置，如图 6.1 所示。



图 6.1 一次回零方式示意图

方式 1：一次原点回零加回找

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 6.2 所示。

方式 2：两次原点回零。



图 6.2 一次回零加回找方式示意图

方式 2：二次回零

如图 6.3 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 1 和方式 2 的说明。

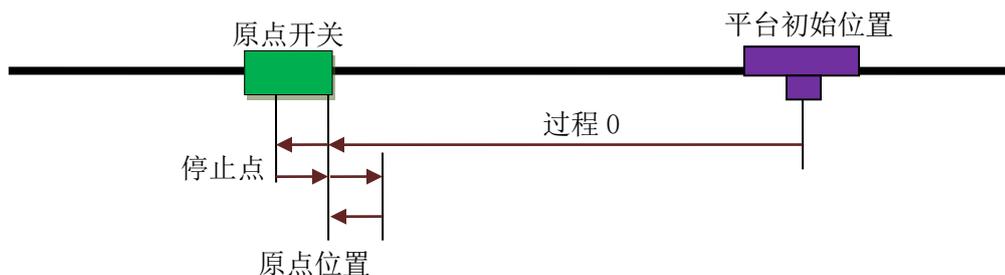


图 6.3 二次回零方式示意图

相关函数如下表 6.4 所示。

表 6.4 回原点相关函数说明

名称	功能	参考
dmc_set_home_pin_logic	设置原点信号的有效电平	7.3 节
dmc_set_homemode	选择回原点模式	
dmc_home_move	按指定的方向和速度方式开始回原点	
dmc_get_home_result	读取回零状态	

### 例程 7.3: 方式 1 低速回原点

```

.....
ushort MyCardNo, Myaxis, Myorg_logic, Myhome_dir, Mymode, MyEZ_count ;
double Myfilter, Myvel_mode;
MyCardNo = 0;    //链接号
Myaxis = 0;     //轴号
Myorg_logic = 0; //原点信号低电平有效
Myfilter = 0;   //保留参数
dmc_set_home_pin_logic MyCardNo, Myaxis, Myorg_logic, Myfilter; //设置 0 号轴原点信号
Myhome_dir = 0; //负方向回零
Myvel_mode = 0; //回零速度模式为低速
Mymode = 0;    //回零模式为方式 0, 一次回零
MyEZ_count = 0; //保留参数
LTDMC.dmc_set_homemode(MyCardNo, Myaxis, Myhome_dir, Myvel_mode, Mymode, MyEZ_count);
//设置 0 号轴回零模式
LTDMC.dmc_set_profile (MyCardNo, Myaxis, 500, 1000, 0.1, 0.1, 500 ); //设置 0 号轴梯形速度曲线
参数
LTDMC.dmc_home_move (MyCardNo, Myaxis); //0 号轴按照设置的模式进行回零运
动
    While (LTDMC.dmc_check_done (MyCardNo, Myaxis) = 0) //检测运动状态, 等待回原点动作完成
    {
        Application.DoEvents;
    }
LTDMC.dmc_set_position (MyCardNo, Myaxis, 0) ; //设置 0 号轴的指令脉冲计数器绝对位置
为 0
.....
    
```

## 6.5 点位运动的实现

EMC1400 脉冲控制器在描述运动轨迹时可以用绝对坐标也可以用相对坐标, 如图 6.4 所示。两种模式各有优点, 如: 在绝对坐标模式中用一系列坐标点定义一条曲线, 如果要修改中

间某点坐标时，不会影响后续点的坐标；在相对坐标模式中，用一系列坐标点定义一条曲线，用循环命令可以重复这条曲线轨迹多次。

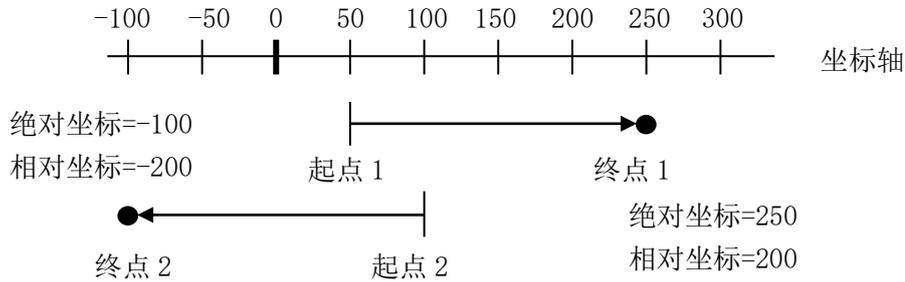


图 6.4 绝对坐标与相对坐标中轨迹终点的不同表达方式

EMC1400 脉冲控制器在执行点位运动控制指令时，可使电机按照梯形速度曲线或 S 形速度曲线进行点位运动。梯形速度曲线参见图 2.3，S 形速度曲线参见图 2.4。

相关函数如表 6.5 所示。

表 6.5 基于脉冲当量的点位运动相关函数说明

名称	功能	参考
dmc_set_profile	设置单轴运动速度曲线	7.6 节
dmc_set_s_profile	设置单轴速度曲线 S 段参数值	7.6 节
dmc_pmove	定长运动	7.7 节
dmc_check_done	检测指定轴的运动状态	7.5 节

### 例：执行点位运动

```

.....
ushort MyCardNo, Myaxis, Myposi_mode;
double MyMin_Vel, MyMax_Vel, MyTacc, MyTdec, MyStop_Vel, Mys_para;
int MyDist;
MyCardNo = 0; //控制器链接号
Myaxis = 0; //轴号
MyMin_Vel = 200; //起始速度 200pluse/s
MyMax_Vel = 5000; //最大速度 5000pluse/s
MyTacc = 0.01; //加速时间 0.01s
MyTdec = 0.01; //减速时间 0.01s
MyStop_Vel = 200; //停止速度 200pluse/s
MyDist = 60000; //位移为 6000pluse
Myposi_mode = 0; //保留参数 0
LTDMC.dmc_set_profile(MyCardNo, Myaxis, MyMin_Vel, MyMax_Vel, MyTacc, MyTdec,
MyStop_Vel); //设置单轴运动速度曲线
LTDMC.dmc_pmove(MyCardNo, Myaxis, MyDist, Myposi_mode); //执行点位运动
while (LTDMC.dmc_check_done(MyCardNo, Myaxis) == 0) //判断轴运动状态，等待运动完

```

```
成
{
    Application.DoEvents();
}
.....
```

在点位运行过程中，最大速度 Max\_Vel 可以实时改变。若在减速时改变目标速度，电机的速度变化曲线参见图 6.5。

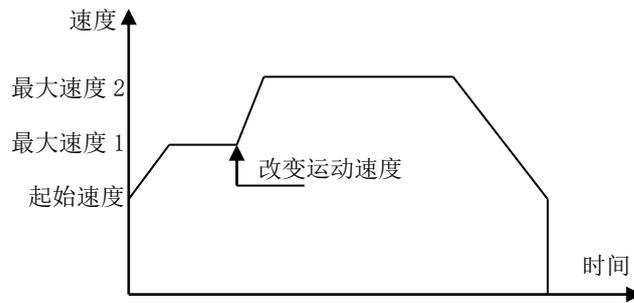


图 6.5 改变速度

实现这 2 个功能的函数如表 6.7 所示。

表 6.7 点位运动中改变速度、目标位置的相关函数说明

名称	功能	参考
dmc_change_speed	在线变速	7.7 节

**注意：** 1) 在线变速适用于点位及连续运动。

2) 在线变速可以设置变速时间，设置的变速时间是从当前速度变速到新速度的时间。此时控制器会重新计算起始速度加速到最高速度所需的时间以及最高速度减速到停止速度所需的时间，即加减速时间会被重新计算。变速一旦成立，该轴的默认运行速度将会被改写为 New\_Vel，加减速时间也会被控制器新计算的数值所覆盖，即当调用 dmc\_get\_profile 回读速度参数时会发生与 dmc\_set\_profile 所设置的值不一致的现象。

**例：** 点位运动中改变速度

```
.....
ushort MyCardNo, Myaxis, Myposi_mode, Mys_mode;
double MyMin_Vel, MyMax_Vel, MyTacc, MyTdec, MyStop_Vel, Mys_para;
int MyDist;
MyCardNo = 0; //控制器链接号
Myaxis = 0; //轴号
Myposi_mode=0; //相对运动模式
Mys_mode = 0; //参数保留
Mys_para = 0.02; //S 段时间为 0.02s
MyMin_Vel = 200; //起始速度 200pluse/s
MyMax_Vel = 5000; //最大速度 5000pluse/s
```

```

MyTacc = 0.05;           //加速时间 0.05s
MyTdec = 0.05;          //减速时间 0.05s
MyStop_Vel = 200;       //停止速度 200pluse/s
MyDist = 60000;         //位移为 60000pluse
    LTDMC.dmc_set_profile(MyCardNo, Myaxis, MyMin_Vel, MyMax_Vel, MyTacc, MyTdec,
MyStop_Vel); //设置单轴运动速度曲线
    LTDMC.dmc_set_s_profile(MyCardNo, Myaxis, Mys_mode, Mys_para); //设置 S 形速度曲线
    LTDMC.dmc_pmove(MyCardNo, Myaxis, MyDist, Myposi_mode); //执行点位运动
    if (“改变速度条件”) //如果在线变速条件满足
    {
        LTDMC.dmc_change_speed(MyCardNo, Myaxis, 10000, 0); //执行在线，速度变为 pluse/s
    }
    .....

```

## 6.6 连续运动的实现

连续运动模式中，EMC1400 脉冲控制器可以控制电机以梯形或 S 形速度曲线在指定的加速时间内从起始速度加速至最大速度，然后以该速度一直运行，直至调用停止指令或者该轴遇到限位信号才会按设定方式减速停止。

相关函数如表 6.8 所示。

表 6.8 连续运动相关函数说明

名称	功能	参考
dmc_vmove	指定轴连续运动	7.7 节
dmc_stop	指定轴停止运动	7.5 节

在执行连续运动过程中，可以调用 `dmc_change_speed` 实时改变速度。注意：在以 S 形速度曲线连续运动时，改变最大速度最好在加速过程已经完成的恒速段进行。图 6.6 和图 6.7 为梯形和 S 形速度曲线下连续运动中变速和减速停止过程的速度曲线。

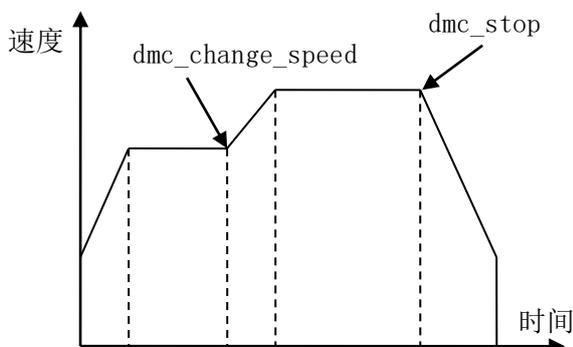


图 6.6 梯形运动中变速

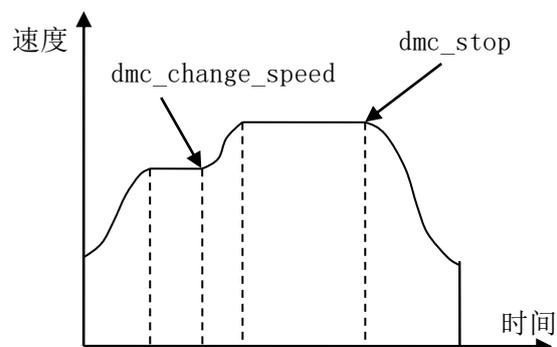


图 6.7 S 形运动中变速

**例：**以 S 形速度曲线加速的连续运动及变速、停止控制

```

.....
ushort MyCardNo, Myaxis, Mydir, Mystop_mode, statemachine;
double MyMin_Vel, MyMax_Vel, MyTacc, MyTdec, MyStop_Vel, Mys_para;
MyCardNo = 0; //控制器链接号
Myaxis = 0; //轴号
MyMin_Vel = 200; //起始速度 200pluse/s
MyMax_Vel = 5000; //最大速度 5000pluse/s
MyTacc = 0.05; //加速时间 0.05s
MyTdec = 0.05; //减速时间 0.05s
MyStop_Vel = 200; //停止速度 200pluse/s
Mys_para=0.01; //s 段时间
Mydir=1; //运动方向为正向
Mystop_mode = 0; //停止方式为减速停止
statemachine = 0;

LTDMC.dmc_set_profile(MyCardNo, Myaxis, MyMin_Vel, MyMax_Vel, MyTacc, MyTdec,
MyStop_Vel); //设置单轴运动速度曲线
LTDMC.dmc_set_s_profile(MyCardNo, Myaxis, 0, Mys_para); //速度曲线为 s 形
LTDMC.dmc_vmive(MyCardNo, Myaxis, Mydir); //执行连续运动
if (“在线变速条件”) //如果在线变速条件满足
{
    LTDMC.dmc_change_speed(MyCardNo, Myaxis, 10000, 0); //在线变速，速度变为 10000pluse/s
}
if (“减速停止条件”) //减速停止条件满足
{
    LTDMC.dmc_stop(MyCardNo, Myaxis, Mystop_mode); //执行减速停止
}
    
```

## 6.7 检测轴到位状态功能的实现

EMC1400 脉冲控制器提供了检测轴到位状态函数，当指令到达设定的目标位置时，dmc\_check\_done 指令返回值会置为 1。

**例：**检测轴到位状态

```

.....
ushort MyCardNo, Myaxis;

MyCardNo=0; //控制器链接号
Myaxis=0; //轴号
    
```

```

LTDMC.dmc_pmove (MyCardNo, Myaxis, 1000, 1);
    //定长运动, 位移为为 1000 pluse、绝对模式
while (LTDMC.dmc_check_done(MyCardNo, Myaxis) == 0)
    //判断轴运动状态, 等待运动完成

{
    Application.DoEvents();
}

.....
    
```

运行结果说明:

运动的目标位置脉冲数为 1000 pluse, 只有当 dmc\_check\_done 函数返回值为 1 时, 表示轴停止。程序才会往下执行。

## 6.8 通用 I/O 控制的实现

### 6.8.1 I/O 普通控制功能

用户可以使用 EMC1400 脉冲控制器上的数字 I/O 口用于检测开关信号、传感器信号等输入信号, 或者控制继电器、电磁阀等输出设备的信号, 通用输出口初始电平默认为高电平。相关函数如表 6.10 所示。

表 6.10 通用 IO 普通控制功能相关函数说明

名称	功能	参考
dmc_read_inbit_ex	读取指定控制器的某一位输入口的电平状态	7.8 节
dmc_write_outbit	对指定控制器的某一位输出口置位	
dmc_read_outbit / dmc_read_outbit_ex	读取指定控制器的某一位输出口的电平状态	
dmc_read_inport / dmc_read_inport_ex	读取指定控制器的全部输入口的电平状态	
dmc_read_outport / dmc_read_outport-ex	读取指定控制器的全部输出口的电平状态	
dmc_write_outport	设置指定控制器的全部输出口的电平状态	

**注意:** 在使用 dmc\_write\_outport 对控制器的全部输出口进行置位, 使用 dmc\_read\_inport、dmc\_read\_outport 进行 IO 电平读取显示时, 应该使用十六进制数进行赋值 (尽量避免使用十进制数, 特别是在不支持无符号变量的开发环境下)。在对 IO 电平进行控制与读取时, 使用十六进制数赋值远比使用十进制数赋值更加直观、方便。

**例:** 读取 0 号链接的通用输入口 1 的电平值, 并对通用输出口 3 置高电平

```

.....
ushort MyCardNo, MyInbitno, Mybitno, MyOutValue = 1 ;
    
```

```

short MyInValue;
MyCardNo=0;           //控制器链接号
MyInbitno=1;         //定义通用输入口 1
Mybitno=3;           //定义输出口 3
MyOutValue = 1;      //定义输出电平为高电平
MyInValue = LTDMC.dmc_read_inbit_ex();
                    //读取通用输入口 1 的电平值，并赋值给变量 MyInValue
LTDMC.dmc_write_outbit(MyCardNo, Mybitno, MyOutValue); //对通用输出口 3 置高电平
.....
    
```

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

```

.....
ushort MyCardNo,Myportno;
ulong a;
int n;
uint MyOutputValue;
MyCardNo=0;           //控制器链接号
Myportno=0;           //IO 口组号
MyOutputValue = 0xFFFFBFA;
                    //(0x 表示 16 进制数) 定义输出口电平值，输出口 0、2、10 为低电平，其余端口为高电平
a= LTDMC.dmc_read_inport(MyCardNo, Myportno);
                    //读取指定控制器的全部输入端口的电平
n = (int)a;           //将 ulong 型的 a 强制转换为 int 型的 n
string MyInportValue = Convert.ToString(n, 2);
                    //将信号返回值转换成二进制数，并赋值给 MyInportValue
LTDMC.dmc_write_outport(MyCardNo, 0, MyOutputValue); //对全部输出口进行电平赋值
.....
    
```

## 6.9 限位开关设置

### 6.9.1 限位开关的设置

EMC1400 脉冲控制器提供了限位开关设置函数 `dmc_set_el_mode` 来设置限位功能。用户必须根据设备的限位开关硬件接线，来设置限位开关工作的有效电平。

相关函数如下表 6.11 所示。

表 6.11 限位开关设置相关函数说明

名称	功能	参考
<code>dmc_set_el_mode</code>	设置限位开关信号	7.9 节

dmc_get_el_mode	读取限位开关信号设置	
-----------------	------------	--

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

#### 例程 6.1：设置限位开关为低电平

.....

```
ushort MyCardNo, Myaxis, Myel_enable, Myel_logic, Myel_mode;
```

```
MyCardNo=0;           //控制器链接号
```

```
Myaxis = 0           //轴号
```

```
Myel_enable = 1      //正负限位使能
```

```
Myel_logic = 0       //正负限位低电平有效
```

```
Myel_mode = 0        //正负限位停止方式为立即停止
```

```
LTDMC.dmc_set_el_mode (MyCardNo, Myaxis, Myel_enable, Myel_logic, Myel_mode) ' 设置 0 号轴限位信号
```

.....

## 第 7 章 基本功能函数说明

### 7.1 控制器设置函数

`short dmc_board_init_eth(WORD CardNo,char* ipaddr)`

功 能：控制器初始化函数，分配系统资源

参 数：CardNo                    控制器链接号  
          Ipaddr                    链接字符串，对应控制器的 IP 地址

返回值：错误代码

`short dmc_cool_reset(WORD CardNo)`

功 能：控制器冷复位函数

参 数：CardNo                    控制器链接号

返回值：错误代码

**注 意：**执行复位操作后，必须关闭控制器，等待 15 秒方可执行初始化控制器，否则会出错。  
出错后必须重新执行复位操作，再等待 15 秒后执行初始化控制器。

`short dmc_soft_reset(WORD CardNo)`

功 能：总线控制器热复位函数

参 数：CardNo                    控制器链接号

返回值：错误代码

**注 意：**执行总线热复位操作后，循环调用 `nmc_get_errcode` 函数判断总线状态，如果总线状态返回正常，则可退出循环，总线热复位完成。

`short dmc_board_close(void)`

功 能：控制器关闭函数，释放系统资源

参 数：无

返回值：错误代码

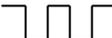
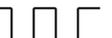
### 7.2 脉冲模式设置函数

`short dmc_set_pluse_outmode(WORD CardNo,WORD axis,WORD outmode)`

功 能：设置指定轴的脉冲输出模式

参 数: CardNo      控制器链接号  
          axis        指定轴号  
          outmode     脉冲方向输出方式选择, 如表 7.1 所示,  
 返回值: 错误代码

表 7.1 指令脉冲方向输出模式

输出脉冲类型 outmode	正方向脉冲		负方向脉冲	
	PLUSE 输出端	DIR 输出端	PLUSE 输出端	DIR 输出端
0		高电平		低电平
1		高电平		低电平
2		低电平		高电平
3		低电平		高电平
4		高电平	高电平	
5		低电平	低电平	
6	AB 相输出, A 超前 B 90°		AB 相输出, B 超前 A 90°	

注意: 在调用运动函数(如: dmc\_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 dmc\_set\_pluse\_outmode 设置控制器脉冲输出模式

```
short dmc_get_pluse_outmode(WORD CardNo,WORD axis,WORD* outmode)
```

功 能: 读取指定轴的脉冲输出模式设置

参 数: CardNo      控制器链接号  
          axis        指定轴号  
          outmode     返回脉冲输出方式

返回值: 错误代码

### 7.3 回原点运动函数

```
short dmc_set_home_pin_logic(WORD CardNo,WORD axis,WORD org_logic,double filter)
```

功 能: 设置 ORG 原点信号

参 数: CardNo      控制器链接号  
          axis        指定轴号  
          org\_logic    ORG 信号有效电平, 0: 低有效, 1: 高有效  
          filter        保留参数, 固定值为 0

返回值: 错误代码

short dmc\_get\_home\_pin\_logic(WORD CardNo,WORD axis,WORD \*org\_logic,double \*filter)

功 能：读取 ORG 原点信号设置

参 数： CardNo 控制器链接号  
axis 指定轴号  
org\_logic 返回设置的 ORG 信号有效电平  
filter 保留参数

返回值：错误代码

short dmc\_set\_homemode(WORD CardNo,WORD axis,WORD home\_dir,double vel\_mode,  
WORD mode,WORD EZ\_count)

功 能：设置回原点模式

参 数： CardNo 控制器链接号  
axis 指定轴号  
home\_dir 回零方向，0：负向，1：正向  
vel\_mode 回零速度模式：  
0：低速回零，即以本指令前面的 dmc\_set\_profile 函数设置的起始速度运行  
1：高速回零，即以本指令前面的 dmc\_set\_profile 函数设置的最大速度运行  
mode 回零模式：  
0：一次回零  
1：一次回零加回找  
2：二次回零  
EZ\_count 保留参数，固定值为 0

返回值：错误代码

short dmc\_get\_homemode(WORD CardNo,WORD axis,WORD\* home\_dir, double\* vel,  
WORD\* mode, WORD\* EZ\_count)

功 能：读取回原点模式

参 数： CardNo 控制器链接号  
axis 指定轴号  
home\_dir 返回回零方向  
vel 返回回零速度模式  
mode 返回回零模式

EZ\_count 保留参数

返回值：错误代码

short dmc\_set\_home\_position(WORD CardNo,WORD axis,WORD enable,double position)

功 能：设置回零偏移量及清零模式

参 数： CardNo 控制器链接号

axis 指定轴号

enable 清零模式，0：不清零，1：回零完成后清零，再偏移；2：回零以及偏移完成后清零

position 偏移量（正值正向偏移；负值负向偏移；0无偏移）。正限位回零只能设置负向偏移，负限位回零只能设置正向偏移

返回值：错误代码

short dmc\_get\_home\_position (WORD CardNo, WORD axis, WORD \* enable, double \* position)

功 能：读取回零偏移量及清零模式

参 数： CardNo 控制器链接号

axis 指定轴号

enable 读取清零模式

position 读取回零偏移量

返回值：错误代码

short dmc\_home\_move(WORD CardNo,WORD axis)

功 能：回原点运动

参 数： CardNo 控制器链接号

axis 指定轴号

返回值：错误代码

short dmc\_get\_home\_result(WORD CardNo,WORD axis,WORD\* state)

功 能：读取回零执行状态

参 数： CardNo 控制器链接号

axis 指定轴号

state 回零执行状态，1：回零完成，0：回零未完成

返回值：错误代码

## 7.4 位置计数器控制函数

short dmc\_set\_position(WORD CardNo,WORD axis,long current\_position)

功 能：设置指令脉冲位置

参 数： CardNo            控制器链接号  
         axis             指定轴号  
         current\_position    指令脉冲位置，单位：pluse

返回值：错误代码

long dmc\_get\_position(WORD CardNo,WORD axis)

功 能：读取指令脉冲位置

参 数： CardNo            控制器链接号  
         axis             指定轴号

返回值：指令脉冲位置，单位：pluse

## 7.5 运动状态检测及控制相关函数

short dmc\_get\_axis\_run\_mode(WORD CardNo, WORD axis, WORD\* run\_mode)

功 能：读取指定轴的运动模式

参 数： CardNo            控制器链接号  
         axis             指定轴号  
         run\_mode        返回运动模式：  
                            0,空闲模式  
                            1,定长模式  
                            2,定速模式  
                            3,回零模式

说 明：1) 该函数适用于所有运动，使用该函数可读取当前的运动模式

short dmc\_read\_current\_speed(WORD CardNo, WORD axis, double \*current\_speed)

功 能：读取指定轴的当前单轴速度

参 数： CardNo            控制器链接号  
         axis             指定轴号  
         current\_speed    返回速度值，单位：pluse/s

返回值：错误代码

short dmc\_get\_stop\_reason(WORD CardNo, WORD axis, long\* StopReason)

功 能：读取指定轴的停止原因

参 数：    CardNo        控制器链接号  
           axis            指定轴号  
           StopReason    停止原因：  
                           0：正常停止  
                           5：正硬限位立即停止  
                           6：负硬限位立即停止  
                           7：正硬限位减速停止  
                           8：负硬限位减速停止  
                           13：命令立即停止  
                           14：命令减速停止

返回值：错误代码

short dmc\_clear\_stop\_reason(WORD CardNo, WORD axis)

功 能：清除指定轴的停止原因

参 数：    CardNo        控制器链接号  
           axis            指定轴号

返回值：错误代码

short dmc\_axis\_io\_status(WORD CardNo, WORD axis)

功 能：读取指定轴有关运动信号的状态

参 数：    CardNo        控制器链接号  
           axis            指定轴号，

返回值：见表 7.2

表 7.2 轴的运动信号状态

位号	信号名称	描述
0	保留	
1	EL+	1：表示正硬限位信号 +EL 为 ON； 0：OFF
2	EL-	1：表示负硬限位信号-EL 为 ON； 0：OFF
4	ORG	1：表示原点信号 ORG 为 ON； 0：OFF
其他位	保留	

short dmc\_check\_done(WORD CardNo,WORD axis)

功 能：检测指定轴的运动状态

参 数： CardNo            控制器链接号  
         axis                指定轴号

short dmc\_stop(WORD CardNo, WORD axis, WORD stop\_mode)

功 能：指定轴停止运动

参 数： CardNo            控制器链接号  
         axis                指定轴号  
         stop\_mode        制动方式，0：减速停止，1：紧急停止

返回值：错误代码

注 意：此函数适用于单轴

## 7.6 单轴运动速度曲线设置函数

short dmc\_set\_profile (WORD CardNo, WORD axis, double Min\_Vel, double Max\_Vel, double Tacc, double Tdec, double Stop\_Vel)

功 能：设置单轴运动速度曲线

参 数： CardNo            控制器链接号  
         axis                指定轴号  
         Min\_Vel            起始速度，单位：pluse/s  
         Max\_Vel            最大速度，单位：pluse/s  
         Tacc                加速时间，单位：s  
         Tdec                减速时间，单位：s  
         Stop\_Vel            停止速度，单位：pluse/s

返回值：错误代码

short dmc\_get\_profile (WORD CardNo, WORD axis, double\* Min\_Vel, double\* Max\_Vel, double\* Tacc, double\* Tdec, double\* Stop\_Vel)

功 能：读取单轴运动速度曲线

参 数： CardNo            控制器链接号  
         axis                指定轴号  
         Min\_Vel            返回起始速度设置  
         Max\_Vel            返回最大速度设置

Tacc	返回加速时间设置
Tdec	返回减速时间设置
Stop_Vel	返回停止速度设置

返回值：错误代码

short dmc\_set\_s\_profile(WORD CardNo, WORD axis, WORD s\_mode, double s\_para)

功 能：设置单轴速度曲线 S 段参数值

参 数： CardNo 控制器链接号  
axis 指定轴号  
s\_mode 保留参数，固定值为 0  
s\_para S 段时间，单位：s；范围：0~1

返回值：错误代码

short dmc\_get\_s\_profile(WORD CardNo, WORD axis, WORD s\_mode, double \*s\_para)

功 能：读取单轴速度曲线 S 段参数值

参 数： CardNo 控制器链接号  
axis 指定轴号  
s\_mode 保留参数  
s\_para 返回设置的 S 段时间

返回值：错误代码

## 7.7 单轴运动函数

short dmc\_pmove (WORD CardNo, WORD axis, double Dist, WORD posi\_mode)

功 能：定长运动

参 数： CardNo 控制器链接号  
axis 指定轴号  
Dist 目标位置，单位：pluse  
posi\_mode 运动模式，0：相对坐标模式，1：绝对坐标模式

返回值：错误代码

short dmc\_vmove(WORD CardNo, WORD axis, WORD dir)

功 能：指定轴连续运动

参 数:    CardNo     控制器链接号  
          axis        指定轴号  
          dir         运动方向, 0: 负方向, 1: 正方向

返回值: 错误代码

short dmc\_change\_speed (WORD CardNo, WORD axis, double New\_Vel, double Taccdec)

功 能: 在线改变指定轴的当前运动速度

参 数:    CardNo     控制器链接号  
          axis        指定轴号  
          New\_Vel     新的运行速度, 单位: pluse/s  
          Taccdec     变速时间, 单位: s

返回值: 错误代码

注 意: 1) 该函数适用于单轴运动中的变速

2) 设置的变速时间是从当前速度变速到新速度的时间。此时控制器会重新计算起始速度加速到最高速度所需的时间以及最高速度减速到停止速度所需的时间, 即加减速时间会被重新计算

3) 变速一旦成立, 该轴的默认运行速度将会被改写为 New\_Vel, 加减速时间也会被控制器新计算的数值所覆盖, 也即当调用 dmc\_get\_profile\_pluse 回读速度参数时会发生与 dmc\_set\_profile 所设置的值不一致的现象

4) 在连续运动中 New\_Vel 负值表示往负向变速, 正值表示往正向变速。在点位运动中 New\_Vel 只允许正值

## 7.8 通用输入输出 IO 函数

short dmc\_read\_inbit(WORD CardNo, WORD bitno)

功 能: 读取指定控制器的某个输入端口的电平

参 数:    CardNo     控制器链接号  
          bitno     输入端口号, 取值范围: 0~3

返回值: 指定的输入端口电平: 0: 低电平, 1: 高电平

short dmc\_read\_inbit\_ex(WORD CardNo, WORD bitno, WORD \*status)

功 能: 读取指定控制器的某个输入端口的电平

参 数:    CardNo     控制器链接号  
          bitno     输入端口号 0-实际输入口数-1

status 指定的输入端口电平：0：低电平，1：高电平

返回值：错误代码

short dmc\_write\_outbit(WORD CardNo, WORD bitno, WORD on\_off)

功 能：设置指定控制器的某个输出端口的电平

参 数： CardNo 控制器链接号

bitno 输出端口号，取值范围： 0~7，如果扩展 IO 模块，依次往后累加

on\_off 输出电平，0：低电平，1：高电平

返回值：错误代码

short dmc\_read\_outbit(WORD CardNo, WORD bitno)

功 能：读取指定控制器的某个输出端口的电平

参 数： CardNo 控制器链接号

bitno 输入端口号，取值范围： 0~7，如果扩展 IO 模块，依次往后累加

返回值：指定输出端口的电平，0：低电平，1：高电平

short dmc\_read\_outbit\_ex(WORD CardNo, WORD bitno, WORD \*status)

功 能：读取指定控制器的某个输出端口的电平

参 数： CardNo 控制器链接号

bitno 输入端口号

status 指定输出端口的电平，0：低电平，1：高电平

返回值：错误代码

short dmc\_read\_inport(WORD CardNo, WORD portno)

功 能：读取指定控制器的全部输入端口的电平

参 数： CardNo 控制器链接号

portno IO 组号，取值范围：0~8；每组 32 位。

返回值：全部输入端口的电平；第 0 组：bit0~bit31 的定义见表 7.3

注意：当背板接入 8 个 32 位输入模块时，组号最大可设置 8。并且第一个模块的第一个输入口 bit 位为第 0 组的第 16 位，后面模块以此类推。

表 7.3 EMC1400 脉冲控制器 dmc\_read\_inport 函数第 0 组和 1 组返回值各 bit 位的定义表

第 0 组 (portno 参数)			第 1 组 (portno 参数)		
函数返回值 bit 位	输入口号	输入口名称	函数返回值 bit 位	输入口号	输入口名称

第 0 组 (portno 参数)			第 1 组 (portno 参数)		
函数返回值 bit 位	输入口号	输入口名称	函数返回值 bit 位	输入口号	输入口名称
0	0	IN0	0	32	IN20
1	1	IN1	1	33	IN21
2	2	IN2	2	34	IN22
3	3	IN3	3	35	IN23
4	4	EL0+	4	36	IN24
5	5	EL1+	5	37	IN25
6	6	EL2+	6	38	IN26
7	7	EL3+	7	39	IN27
8	8	EL0-	8	40	IN28
9	9	EL1-	9	41	IN29
10	10	EL2-	10	42	IN30
11	11	EL3-	11	43	IN31
12	12	ORG0	12	44	IN32
13	13	ORG1	13	45	IN33
14	14	ORG2	14	46	IN34
15	15	ORG3	15	47	IN35
16-31	16~31	IN4~19	16-31	48~63	IN36~51

short dmc\_read\_inport\_ex(WORD CardNo, WORD portno, DWORD \*status)

功 能：读取指定控制器的全部输入端口的电平

参 数：CardNo 控制器链接号

portno IO 组号

status 全部输入端口的电平

返回值：错误代码

short dmc\_read\_outport(WORD CardNo, WORD portno)

功 能：读取指定控制器的全部输出口的电平

参 数：CardNo 控制器链接号

portno IO 组号；取值范围：0~8；每组 32 位

返回值：bit0~bit31 的定义见表 7.4

short dmc\_read\_outport\_ex(WORD CardNo, WORD portno, DWORD \*status)

功 能：读取指定控制器的全部输出口的电平

参 数：CardNo 控制器链接号

portno IO 组号；取值范围：0~8；每组 32 位

status 对应输出 port 口号的电平

返回值：错误代码

short dmc\_write\_outport(WORD CardNo, WORD portno, DWORD port\_value)

功 能：设置指定控制器的全部输出口的电平

参 数：CardNo 控制器链接号

portno IO 组号；取值范围：0~8；每组 32 位

port\_value bit0~bit31 的定义见表 7.4

返回值：错误代码

注意：当背板接入 8 个 32 位输出模块时，组号最大可设置 8。并且第一个模块的第一个输出口 bit 位为第 0 组的第 8 位，后面模块以此类推。

表 7.4 EMC1400 脉冲控制器 dmc\_read\_outport、dmc\_write\_outport 函数返回值各位的定义表

第 0 组 (portno 参数)		
函数返回值的 bit	输出口号	输出口名称
0	0	OUT0
1	1	OUT1
2	2	OUT2
3	3	OUT3
4	4	OUT4
5	5	OUT5
6	6	OUT6
7	7	OUT7
8-31	8-31	背板：OUT8~OUT31

## 7.9 限位设置

short dmc\_set\_el\_mode(WORD CardNo,WORD axis,WORD el\_enable,WORD el\_logic,WORD el\_mode)

功 能：设置 EL 限位信号

参 数：    CardNo            控制器链接号  
          axis                指定轴号  
          el\_enable        EL 信号的使能状态：  
                              0: 正负限位禁止  
                              1: 正负限位允许  
                              2: 正限位禁止、负限位允许  
                              3: 正限位允许、负限位禁止  
          el\_logic        EL 信号的有效电平：  
                              0: 正负限位低电平有效  
                              1: 正负限位高电平有效  
                              2: 正限位低有效，负限位高有效  
                              3: 正限位高有效，负限位低有效  
          el\_mode        EL 制动方式：  
                              0: 正负限位立即停止  
                              1: 正负限位减速停止  
                              2: 正限位立即停止，负限位减速停止  
                              3: 正限位减速停止，负限位立即停止

返回值：错误代码

short dmc\_get\_el\_mode(WORD CardNo,WORD axis, WORD \*el\_enable,WORD \*el\_logic,WORD \*el\_mode)

功 能：读取 EL 限位信号设置

参 数：    CardNo            控制器链接号  
          axis                指定轴号  
          el\_enable        返回设置的 EL 信号使能状态  
          el\_logic        返回设置的 EL 信号有效电平  
          el\_mode        返回 EL 制动方式

返回值：错误代码

## 7.10 打印输出函数

`short dmc_set_debug_mode(WORD mode, const char *FileName)`

功 能：函数调用打印输出设置

参 数：     `mode`            打印输出模式，0：只打印报错函数，1：全部打印，2：全部不打印  
          `FileName`        文件保存路径：  
                              参数文件名+后缀：相对路径  
                              完整描述参数文件的路径+文件名后缀：绝对路径

返回值：错误代码

说 明：使能打印输出后，可监控运动函数库的调用情况。在用户调用函数时，将输出相关信息，并保存在指定文件路径中；函数设置模式 2 全部不打印需配合最新动态库（20181030 及以后动态库）使用。

`short dmc_get_debug_mode(WORD mode, char *FileName)`

功 能：读取函数调用打印输出设置

参 数：     `mode`            返回打印输出使能状态  
          `FileName`        返回文件保存路径

返回值：错误代码

## 附 录

### 附录 1 常见电路接线方法

#### 1.1 输入输出接口

##### 1.1.1 通用输出 IO 电路

EMC1400 运动控制器具有 8 路通用输出 OUT0-OUT7，支持 24V 负载。输出采用低损耗 MOS 开漏极输出，输出电流可达 500mA，可以直接驱动小型继电器和电磁阀等外设，内部的光耦、滤波器，可以隔离干扰信号，提高系统可靠性。

##### 1) 与电磁阀连接电路

当负载为感性器件时，在开关时感性元件的两端会产生感应电压尖峰，电感越大，开关频率越快则电压尖峰越大，有可能击穿开关 MOS 管，所以需要在感性负载上并联一个续流二极管，减小关断时电感电流的变化率降低电压尖峰，保护 MOS 管。其连接示意图如下：

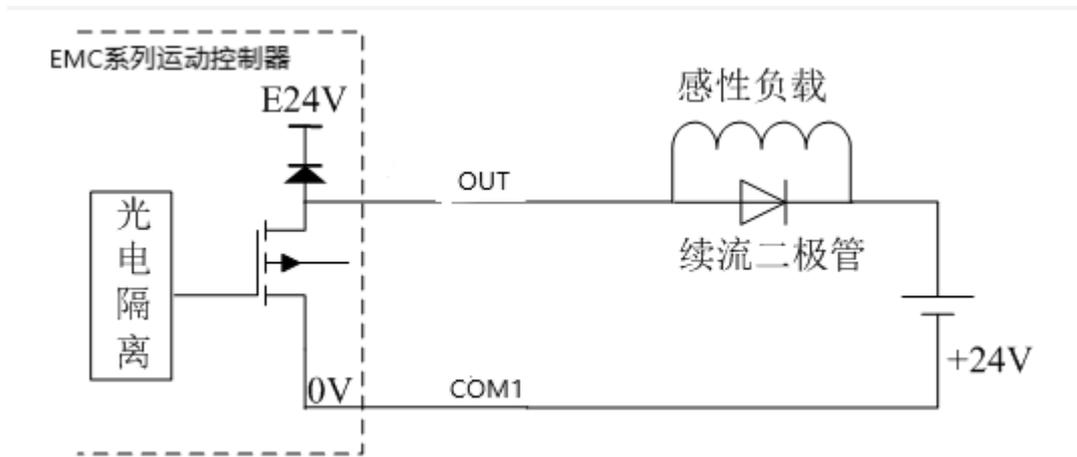


图 F1-1 控制器与电磁阀连接图

说明：+24V，0V 为外部电源和外部电源地。

注意：所有输出口使用时不能将电源端直接短接到输出口，这样会损坏内部电路结构引起控制器输出口失效。

##### 2) 与中间继电器连接电路

控制器连接 OMRON 中间继电器 LY1J 24VDC 的电路图如图 F1-2 所示。

继电器为感性负载，其线圈外必须并联一个续流二极管，以保护控制器输出端口驱动元件。在关闭继电器的瞬间，不被线圈产生的感应电动势击穿。（OMRON 中间继电器 LY1J 线圈电压 24VDC，最大开关电压 250VAC 或 125VDC，最大开关电流 15A。）

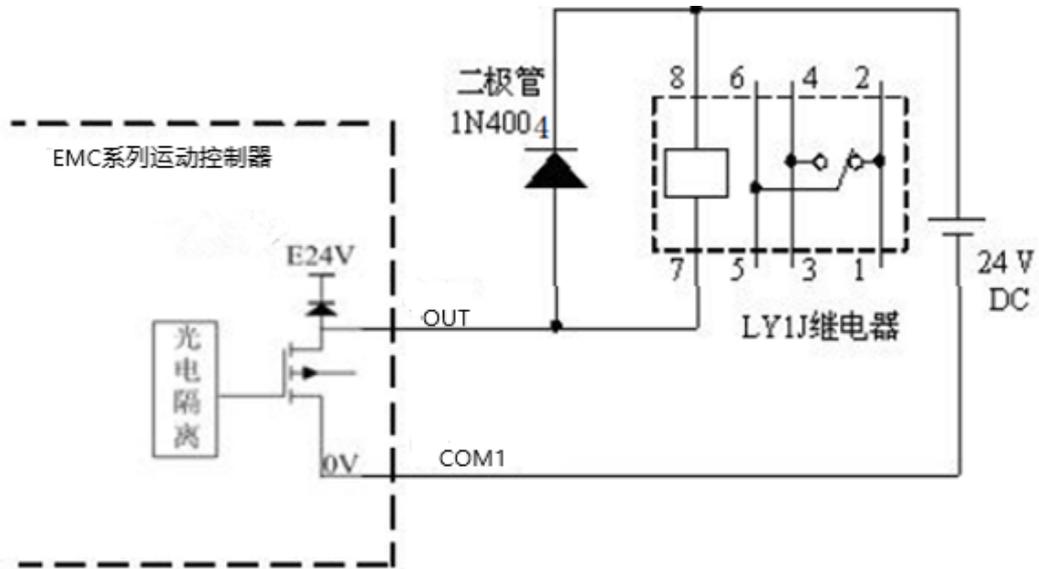


图 F1-2 控制器与中间继电器的接线图

### 3) 小型继电器:

继电器为感性负载，当继电器突然关断时，其电感会产生一个很大的反向电压，有可能击穿输出 MOS 管，模块内输出口有续流二极管，以保护输出口 MOS 管。继电器接线图如图 F1-3 所示：

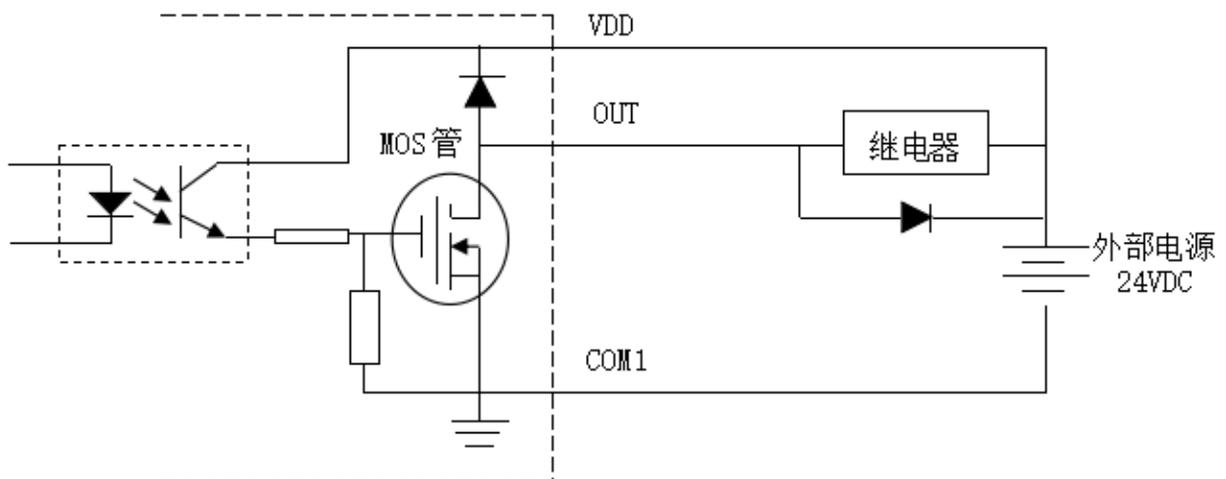


图 F1-3 通用输出接线图

**⚠ 注意：**在使用通用数字输出端口时，切勿把外部电源直接接至通用数字输出端口上，否则会造成 MOS 管损坏。

### 1.1.2 通用输入/轴信号电路

EMC1400 脉冲控制器具有 4 路通用输入 IN0-IN3，输入全部为光电隔离，支持 24V 电平。

通用输入的接线如图 F1-5 所示

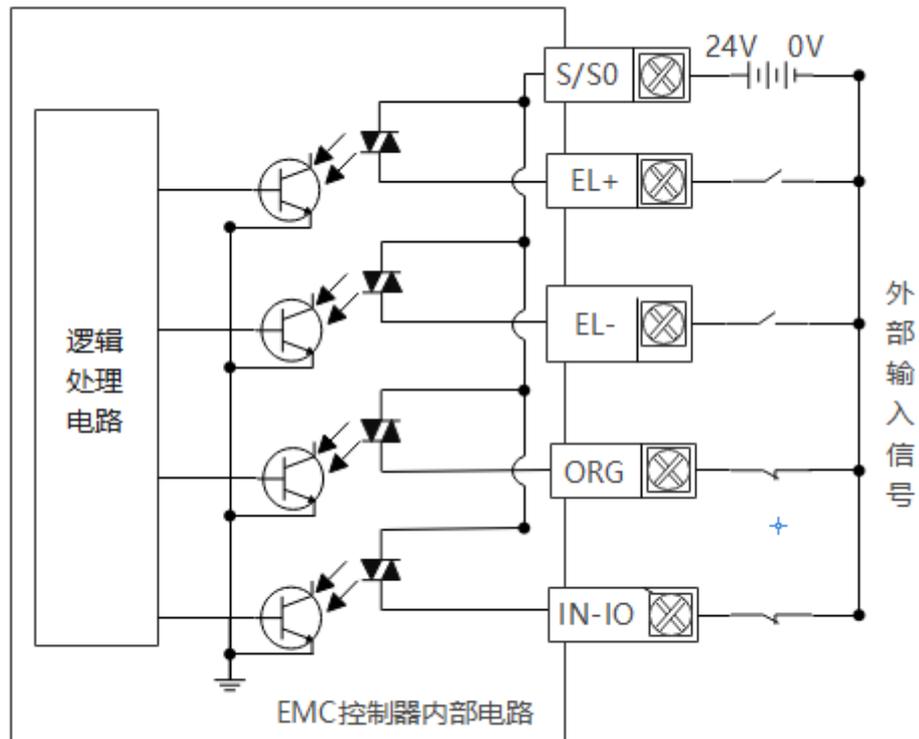


图 F1-5 通用输入接线图示

## 附录 2 常见错误码说明

EMC1400 脉冲控制器的主要函数返回值错误码见表 F2.0。

表 F2.0 函数返回值错误码

错误码	含义
0	无错误
-1	内部的一些空指针返回值
2	动态库层参数错误
4	动态库层检测到轴处于运动中
8	无法连接错误
9	动态库层控制器链接号错误
12	固件文件错误
14	动态库层下载固件与控制器/器型号不匹配
17	不支持的功能，预留，不能更改
19	链接对应型号类型不匹配
20	固件参数错误
22	固件当前状态不允许操作
24	固件不支持功能，保留原版本的错误码
1311	正负限位都有效
3000	目标速度小于等于零错误
3001	点位运动位移为零或规划时运动距离为零
3002	轴状态未处于完成或空闲状态，启动点位运动无效
3003	<b>PMOVE</b> 规划或设置的起跳速度小于 0
3004	<b>PMOVE</b> 规划或设置目标速度等于 0
3005	<b>PMOVE</b> 规划或设置结束速度小于 0
3006	正向硬件限位触发，不允许正向启动 <b>PMOVE</b>
3007	负向硬件限位触发，不允许负向启动 <b>PMOVE</b>
3012	在线变速失败，新的目标速度小于 0
3020	当前处于回零运动模式，不允许该操作
3032	<b>PMOVE</b> 平滑时间大于 1 秒报错
3033	设置的加速时间等于 0
3034	设置的减速时间等于 0
3035	设置的加速度小于等于 0
3036	设置的减速度小于等于 0
3043	设置的加速时间大于 100S
3044	设置的减速时间大于 100S
3045	设置的平滑时间大于加速或减速时间
3050	加/减加速度为零
3100	正向硬件限位触发，不允许正向启动 <b>VMOVE</b>
3101	负向硬件限位触发，不允许负向启动 <b>VMOVE</b>
3302	轴正在运动

---

---

3303	轴号超出范围
3308	正向硬件限位触发
3309	负向硬件限位触发
5128	设置的 S 段时间大于加速或减速时间
6001	不支持
6002	参数错误
6024	参数不在有效范围
6039	轴硬件限位信号有效

## 附录 4 运动控制函数索引

函数分类	函数名	描述	索引
控制器设置函数	dmc_board_init_eth	控制器初始化函数	7.1 节
	dmc_soft_reset	控制器热复位函数	
	dmc_cool_reset	控制器冷复位函数	
	dmc_board_close	控制器关闭函数	
脉冲模式设置函数	dmc_set_pluse_outmode	设置指定轴的脉冲输出模式	
	dmc_get_pluse_outmode	读取指定轴的脉冲输出模式设置	
回原点运动函数	dmc_set_home_pin_logic	设置 ORG 原点信号	7.3 节
	dmc_get_home_pin_logic	读取 ORG 原点信号设置	
	dmc_set_homemode	设置回原点模式	
	dmc_get_homemode	读取回原点模式	
	dmc_set_home_position	设置回零偏移量及清零模式	
	dmc_get_home_position	读取回零偏移量及清零模式	
	dmc_home_move	回原点运动	
限位开关设置函数	dmc_set_el_mode	设置 EL 限位信号	7.9 节
	dmc_get_el_mode	读取 EL 限位信号设置	
位置计数器控制函数	dmc_set_position	设置指令脉冲位置	7.4 节
	dmc_get_position	读取指令脉冲位置	
运动状态检测及控制函数	dmc_get_axis_run_mode	读取指定轴的运动模式	7.5 节
	dmc_read_current_speed	读取当前速度值	
	dmc_check_done	检测指定轴的运动状态	
	dmc_axis_io_status	读取指定轴有关运动信号的状态	
	dmc_get_stop_reason	读取轴停止原因	
	dmc_clear_stop_reason	清除轴停止原因	
	dmc_stop	指定轴停止运动	
单轴运动速度曲线设置函数	dmc_set_profile	设置单轴运动速度曲线	7.6 节
	dmc_get_profile	读取单轴运动速度曲线	
	dmc_set_s_profile	设置单轴速度曲线 S 段参数值	
	dmc_get_s_profile	读取单轴速度曲线 S 段参数值	
单轴运动函数	dmc_pmove	指定轴点位运动	7.7 节
	dmc_vmove	指定轴连续运动	
	dmc_change_speed	在线变速	
通用输入输出 IO 函数	dmc_read_inbit	读取指定控制器的某个输入口的电平状态	7.8 节
	dmc_read_inbit_ex	读取指定控制器的某个输入口的电平状态	
	dmc_write_outbit	设置指定控制器的某个输出口的电平状态	
	dmc_read_outbit	读取指定控制器的某个输出口的电平状态	
	dmc_read_outbit_ex	读取指定控制器的某个输出口的电平状态	

函数分类	函数名	描述	索引
	dmc_read_inport	读取指定控制器的全部输入口的电平状态	
	dmc_read_inport_ex	读取指定控制器的全部输入口的电平状态	
	dmc_read_outport	读取指定控制器的全部输出口的电平状态	
	dmc_read_outport_ex	读取指定控制器的全部输出口的电平状态	
	dmc_write_outport	设置指定控制器的全部输出口的电平状态	
打印输出函数	dmc_set_debug_mode	函数调用打印输出设置	7.10节
	dmc_get_debug_mode	读取函数调用打印输出设置	

## 附录 5 常见问题解决方法

序号	问题描述	可能原因及解决方法
1	控制器无法扫描到模块	a. 检查连接是否松动
		b. 检查模块是否上电、有无报错，若报错先断电重启，执行热复位再扫描
		c. 控制器是否正常工作，可复位系统完成后再扫描
2	PC 机不能和控制器通讯	检查网线是否有松动，控制器 RUN 灯是否正常； 参考软件手册检查应用软件是否编写正确。
3	电机可以转动，但工作不正常	检查控制器和驱动器是否正确接地，抗干扰措施是否做好。
4	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。
5	数字输入信号不能读取	接线是否正常；检查函数调用。
6	数字输出信号不正常	接线是否正常；检查函数调用。
7	电脑休眠后找不到控制器	刷新设备管理器或重启电脑



**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：深圳市南山区沙河西路南山智谷产业园 B 栋 15 楼

邮 编：518055

电 话：0755-26415968

传 真：0755-26417609

Email: [info@szleadtech.com.cn](mailto:info@szleadtech.com.cn)

网 址: <http://www.szleadtech.com.cn>