

网口型 IO 卡使用手册

EIO-0808

手册版本：1.00

版权申明

珠海市睿控科技有限公司

保留所有权力

珠海市睿控科技有限公司（以下简称睿控科技）保留在不事先通知的情况下，修改本手册中的产品和产品规格等文件的权力。

睿控科技不承担由于使用本手册或本产品不当，所造成直接的、间接的、特殊的、附带的或相应产生的损失或责任。

睿控科技具有本产品及其软件的专利权、版权和其它知识产权。未经授权，不得直接或者间接地复制、制造、加工、使用本产品及其相关部分。



运动中的机器有危险！使用者有责任在机器中设计有效的出错处理和安全保护机制，睿控科技没有义务或责任对由此造成的附带的或相应产生的损失负责。

目录

第一章 硬件接口电路	4
1.1 硬件简介	4
1.2 IO 卡与配件的连接	5
1.3 通用 I/O 接口电路	5
1.4 CAN-IO 扩展模块接口电路	7
第二章 接口说明	8
2.1、外观尺寸布局说明	8
2.2 接口定义说明	9
第三章快速使用	12
3.1 开箱检查	12
3.2 安装步骤	12
第四章运动控制器函数库的使用	13
4.1 WINDOWS 系统下动态链接库的使用	13
第五章系统配置及使用	16
5.1 初始化	16
5.2 IO 基本操作	16
5.3 CAN 扩展模块的使用	18
第六章 同时使用多张卡	20
6.1 使用网卡连接多卡（以四口网卡为例）	20
6.2 使用交换机连接多卡	21
6.3 使用 DEMO 软件控制多卡（网卡或交换机连接）	21
附 A、CAN 总线 IO 扩展板说明	23
附 B、修改 IP 地址的方法	28
附 C、DEMO 调试 EIO-0808 的步骤	30
附 C.1、建立物理连接	30
附 C.2、设置 PC 机的 IP 地址	30
附 C.3、开启 DEMO 软件	30
附 C.4、选择需要打开卡的 IP 地址，建立连接。	30
附 C.5、IO 操作测试	31
附 C.6、进行从站 IO 操作（如果有接入 CAN 总线 IO 模块）	31

图1.2 EIO-0808系列卡硬件布置及尺寸图

1.2 IO 卡与配件的连接

1.2.1 EIO-0808 与配件的连接

EIO-0808 卡无须额外的必选配件，可单独与带网卡的电脑进行实时连接，连接示意图如图 1.3 所示。

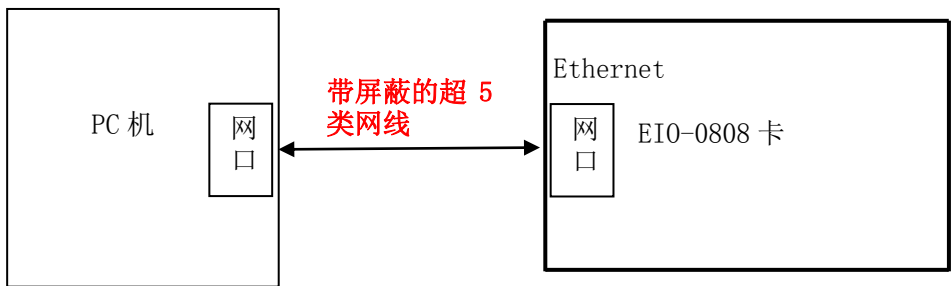


图1.3 EIO-0808与PC连接示意图

注意：建议使用带屏蔽的超5类网线（直通交叉自适应）进行连接通讯，并做好系统接地（屏蔽地）处理，以减少外部干扰对通讯的影响。

1.3 通用 I/O 接口电路

EIO-0808 卡总共提供了 16 路通用数字 I/O 接口。最多可扩展 256 个 IO 点。

1.3.1 通用数字输入信号接口

EIO-0808 系列卡有 8 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 1.4 所示。

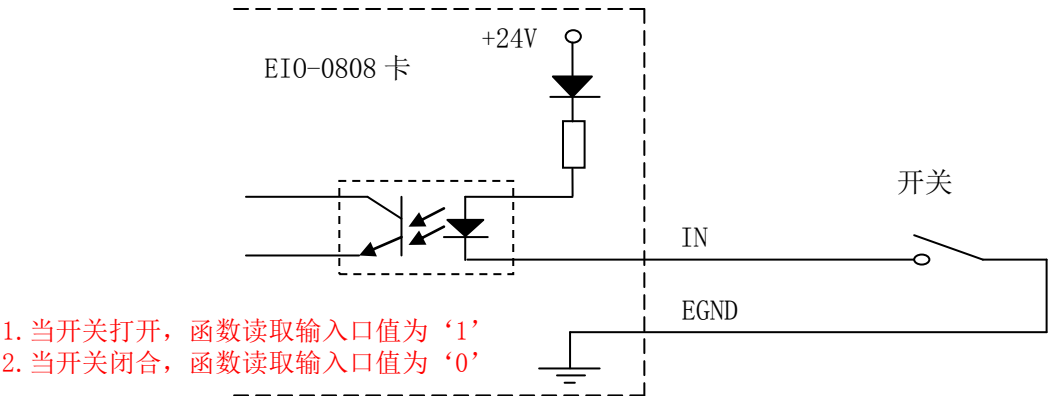


图 1.4 通用输入信号接口原理图

1.3.2 通用数字输出信号接口

EIO-0808 卡有 8 路通用数字输出信号，最大工作电流为 400 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 3 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 3.21 所示。

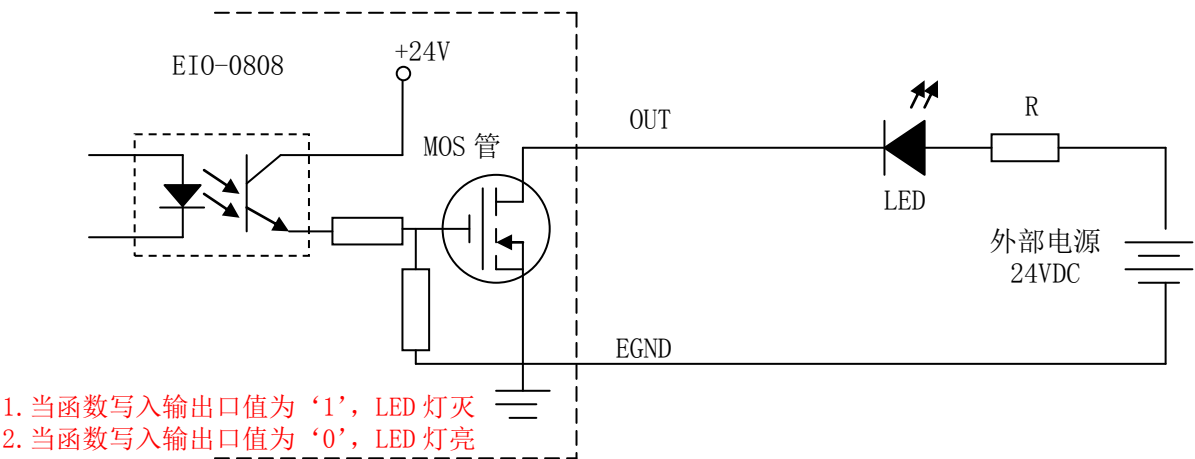


图 1.5 EIO-0808 卡输出口接发光二极管

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 1.6 所示。

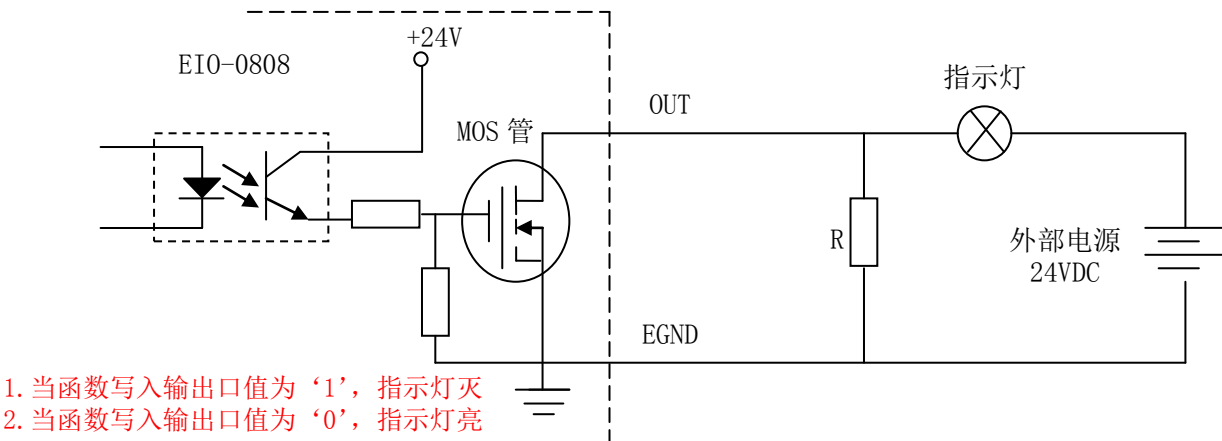


图 1.6 EIO-0808 灯丝型指示灯接线图

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 NPN 被感应电动势击穿。其接线图如图 1.7 所示。

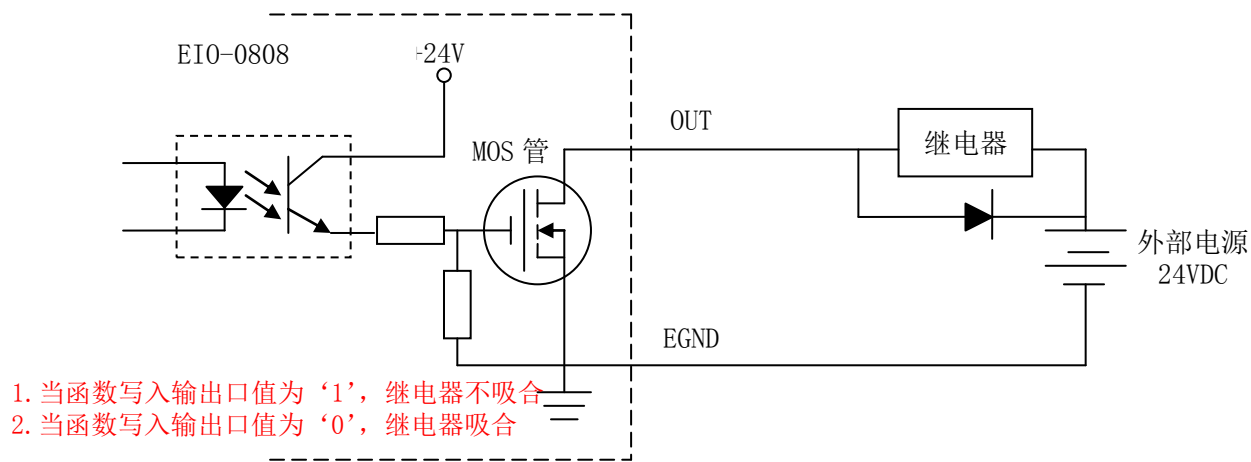


图 1.7 EIO-0808 接小型继电器的接线图

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。

1.4 CAN-IO 扩展模块接口电路

EIO-0808 卡有一路 CAN 总线接口，可外接 CAN 扩展模块，其连接电路如下：

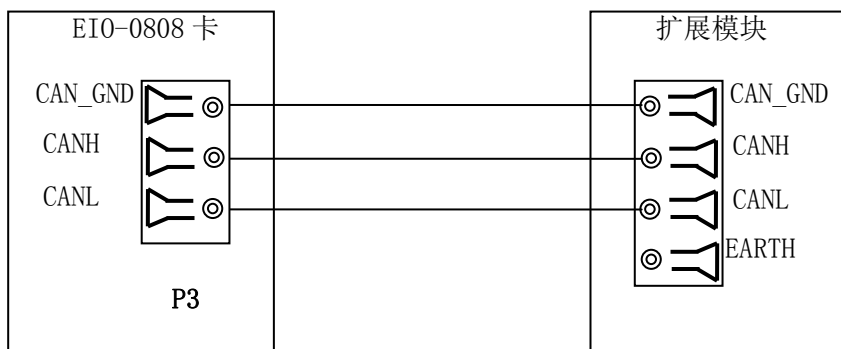


图 1.8 EIO-0808 与 CAN 模块接线图

表 1.1 CAN 总线信号说明

信号名称	信号说明	备注
CAN_GND	CAN 电源地信号	IO 卡和扩展模块的地必须相连
CANH	CAN 差分数据+	
CANL	CAN 差分数据-	
EARTH	CAN 屏蔽地/安规地	建议使用屏蔽线，并将屏蔽层接入到 EARTH

注意：连接多个 CAN 模块的时候需要在最后一个 CAN 模块上接入 120 欧的终端电阻，此终端电阻可通过 CAN 模块的跳线帽选择是否接入，具体详见 CAN 模块手册说明！

第二章 接口说明

2.1、外观尺寸布局说明

EIO-0808 卡外观如图 2.1 所示，接口位置示意图如图 2.2 所示。

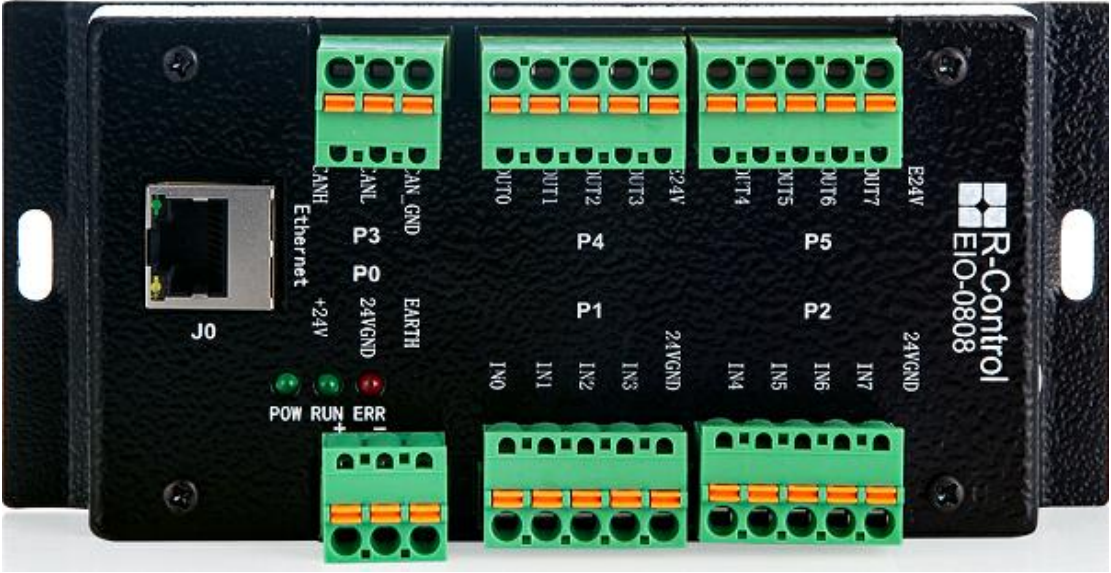


图 2.1 EIO-0808 卡外观图

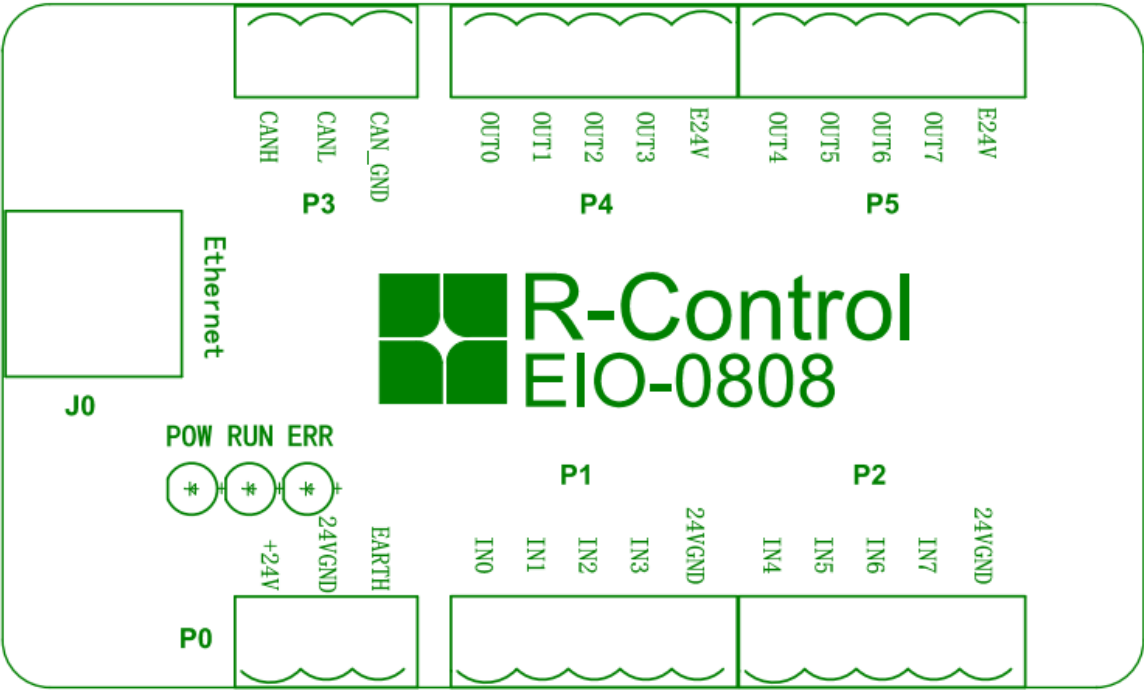


图 2.2 EIO-0808 卡接口位置示意图

2.2 接口定义说明

EIO-0808 卡总共有如下接口, 具体见表 2. 1:

表 2. 1 EIO-0808 卡接口功能简述

名称	功能介绍
J0	百兆网接口 (ETHERNET)
P0	DC24V 电源输入接口
P1-P2	通用输入端口 IN0-IN7
P4-P5	通用输出端口 OUT0-OUT7
P3	CAN 总线接口
注意: 当输出口接入感性负载时需要外接续流二极管	

2.2.1、P0 电源定义

接口 P0 是 IO 卡的电源输入接口, 板上及外壳标有 24V 的端子接+24V, 标有 EXGND 的端子接外部电源地。

表 2. 2 P0 接口引脚号和信号关系表

序	名称	I/O	说 明
1	+24V	I	DC24V 电源输入

序	名称	I/O	说 明
2	24VGND	I	DC24V 电源地
3	EARTH	I	安规地

2.2.2、P1-P2 通用输入信号（IN0-IN7）接口定义

P1-P2 为 8 路通用输入（IN0-IN7）接口，其引脚号和信号对应关系见表 2.3、表 2.4、所示。

表 2.3 P1 引脚号和信号关系表

序	名称	I/O	说 明
1	IN0	I	通用输入 0
2	IN1	I	通用输入 1
3	IN2	I	通用输入 2
4	IN3	I	通用输入 3
5	24VGND	I	24V 电源地

表 2.4 P2 引脚号和信号关系表

序	名称	I/O	说 明
1	IN4	I	通用输入 4
2	IN5	I	通用输入 5
3	IN6	I	通用输入 6
4	IN7	I	通用输入 7
5	24VGND	I	24V 电源地

2.2.3、P4-P5 通用输出信号（OUT0-OUT7）接口定义

P4-P5 为 8 路通用输出(OUT0-OUT7)接口，其引脚号和信号对应关系见表 2.5、表 F2.6、所示。

表 2.5 P4 引脚号和信号关系表

序	名称	I/O	说 明
1	OUT0	0	通用输出 0
2	OUT1	0	通用输出 1
3	OUT2	0	通用输出 2
4	OUT3	0	通用输出 3
5	E24V	0	DC24V 电源输出

表 2.8 P5 引脚号和信号关系表

序	名称	I/O	说 明
1	OUT4	0	通用输出 4
2	OUT5	0	通用输出 5
3	OUT6	0	通用输出 6
4	OUT7	0	通用输出 7
5	E24V	0	DC24V 电源输出

2.2.4、P3 CAN 总线接口定义

EIO-0808 卡有 1 路 CAN 总线接口 P3，可以连接睿控 CAN 总线从站扩展模块，实现更多的 I/O 扩展功能，具体端口定义如下表 2.9

表 2.9 P3 引脚号和信号关系表

序	信号名称	信号说明	备注
1	CANH	CAN 差分数据+	
2	CANL	CAN 差分数据-	
3	CAN_GND	CAN 电源地信号	IO 卡和扩展模块的地必须相连

2.2.5、指示灯定义

EIO-0808 模块表面有 3 个指示灯，分别为：

POW（绿色）：外部电源指示灯，系统正常上电运作状态下常亮；如果接入 DC24V 电源不亮灯，请仔细查看是否 DC24V 正常供电及接入顺序是否正确。

RUN（绿色）：IO 卡运行状态指示灯，正常连接状态下此绿色指示灯闪烁，其中运行状态根据闪烁频率来区分：


- a. 600ms 闪动频率，代表只有以太网上下位机通讯正常；
- b. 100ms 闪动频率，以太网上下位和 CAN 扩展通讯都正常
- c. 200ms 闪动频率，只有 can 扩展通讯正常。

ERR（红色）：IO 卡网络物理连接指示灯。IO 卡与 PC 网络物理连接状态下此灯常灭，如果连接失败此灯常亮。

第三章快速使用

3.1 开箱检查

打开包装后，请确认与所订型号是否一致，再检查IO卡的表面是否有损坏，然后按照装箱清单或订购合同仔细核对配件是否齐备。如果IO卡表面有损坏，或产品型号不符合，请不要使用，立即与睿控科技联系。

	<p>为了防止静电损害控制卡，请在接触IO卡电路之前触摸有效接地金属物体以释放身体所携带的静电荷。</p>
---	---

3.2 安装步骤

请按照以下安装步骤建立控制系统：

- 步骤1：将IO卡通过网线与PC连接
- 步骤2：为IO卡供电 (24V)
- 步骤3：连接输入输出实际负载
- 步骤4：使用demo软件或编写软件来控制IO卡

3.2.1 步骤 1：将 IO 卡通过网线与 PC 连接

使用光盘里附带的DEMO程序，测试主机是否和IO卡建立了联系。如果DEMO程序能工作开启，没有提示找不到卡，证明IO卡通讯正常。否则会提示错误信息“打开板卡失败”，证明IO卡通讯失败。在通讯成功的前提下，用户可进行相关功能的测试。

3.2.2 步骤 2：连接输入输出负载

具体接法请见第二章接线说明

3.2.3 步骤 3：接 CAN 扩展模块

如果需要更多的IO数量，请参照附A连接CAN扩展模块。如果不需要，则忽略此步骤。

第四章运动控制器函数库的使用

4.1 Windows 系统下动态链接库的使用

下面使用Visual Studio 2013 C#编写一个控制台应用程序来进行演示， 操作系统为64位系统。

1. 启动Visual Studio，新建一个工程；

点击菜单“文件”-->“新建”-->“项目”，如图

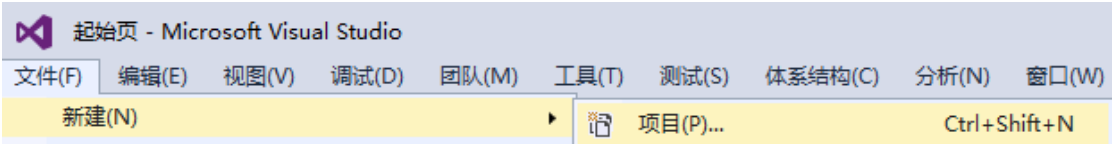


图4.1 建立工程的菜单

选择“Windows桌面”-->“控制台应用程序”，设置工程名称和存储位置，点击“确定”

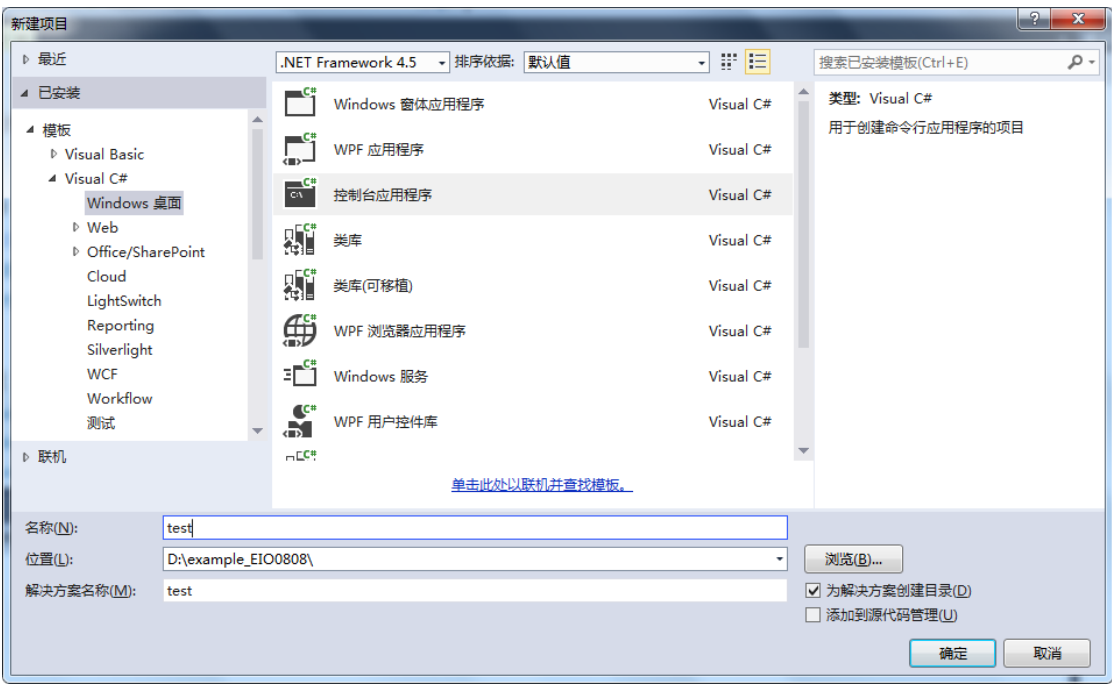


图4.2 建立控制台应用程序

2、复制动态链接库

复制“EIO.cs”、“EIO.dll”（注意选择64位版本的动态库）两个文件到工程中，如下图。

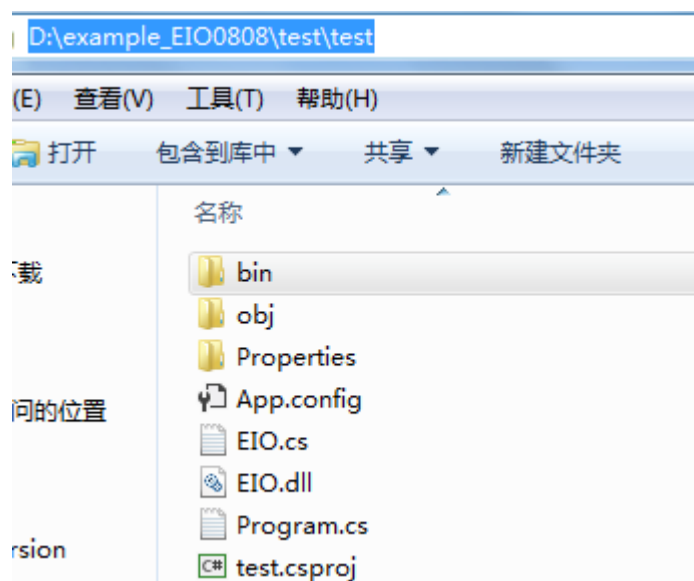


图4.3 工程中加入动态库

编译之后，还需要复制文件“EIO.dll”到 bin目录下，因为选择的解决方案平台是X64，解决方案是Debug，如图 所示，所以需要把“EIO.dll”拷贝到\bin\x64\Debug目录下。

备注：由于提供的动态库分32位和64位两个版本，需要根据选择的解决方案平台来选用。

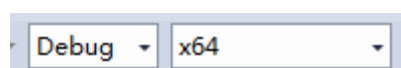


图4.4 确定解决方案及平台

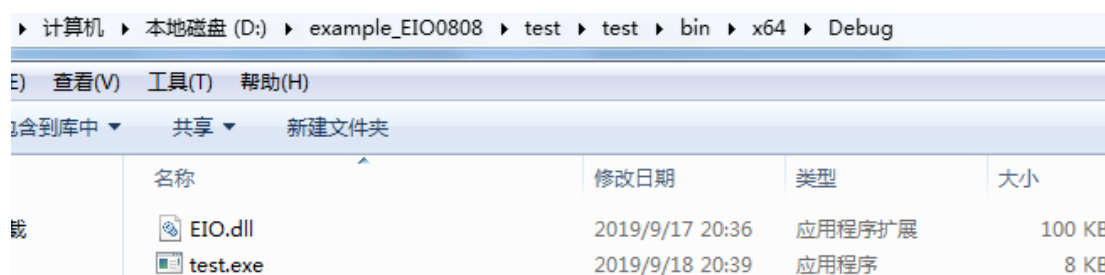


图4.5 在执行文件目录下添加动态库

3、编写代码

在文件Program.cs中添加

```
using csEIO;
```

在main函数中增加如下代码即可执行开卡的动作

```
using csEIO;
```

```
namespace test
```

```
{
```

```
    class Program
```

```
    {
```

```

static void Main(string[] args)
{
    UInt16 plt_CardNum = 1;
    UInt16[] plt_CardID = new UInt16[8];
    UInt16[] Section = new UInt16[8];
    UInt16[] Host_id = new UInt16[8];
    plt_CardID[0] = 0;
    Section[0] = 167;    //第三段固定为167
    Host_id[0] = 120;    //第四段设置为120（根据实际情况设置，卡出厂默认是120）
    int ret = csEIO.EIO.Plt_CardOpen(plt_CardNum, plt_CardID, Section, Host_id);
    if (ret != 0)//打开控制卡
    {
        Console.Write("打开EIO-0808失败");
    }
    else
        Console.Write("打开EIO-0808成功");

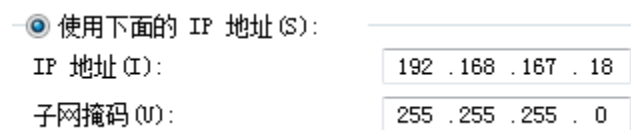
    //此处可以添加代码进行IO操作

    Console.ReadLine();
}
}
}

```

4、设置PC的IP地址

设置PC机的IP地址与控制卡在同一网段（192.168.167.*），第四段与控制卡不重复即可，建议使用小于120的值



☒ 使用下面的 IP 地址(S):

IP 地址(I): 192 . 168 . 167 . 18

子网掩码(U): 255 . 255 . 255 . 0

图4.6 PC机的IP设置

5、 执行上面程序的结果：

若正常连接控制卡，并正确设置卡号等，控制台上显示“打开EIO-0808成功”；

若未连接卡或设置错误，控制台程序会等待一段时间后显示“打开EIO-0808失败”。

第五章系统配置及使用

在使用I0卡进行各种操作之前，需要对I0卡进行配置，使I0卡的状态和各种工作模式能够满足客户的要求。

5.1 初始化

使用I0卡之前需要先初始化，使其为后续的指令做好相关准备；使用结束之后需要关闭I0卡。

表5.2 初始化指令列表

指令	说明
Plt_CardOpen	初始化控制卡
Plt_CardClose	关闭控制卡
Plt_CardGetVersion	获取版本号

表5.3 运动状态检测指令参数说明

Plt_CardOpen(uint16 TotalCards,uint16 *CardIdArray,uint16 *Section,uint16* Host_id)	
TotalCards	卡的数量
CardIdArray	卡的 ID 列表
Section	固定为 167
Host_id	卡的 IP 地址第四段（出厂为 120）
Plt_CardClose(void)	
Plt_CardGetVersion(uint16 cardid,DWORD *verson1,DWORD *verson2)	
cardid	卡号（0-3）
verson1	固件版本号
verson2	库版本号

重点说明：

初始化函数Plt_CardOpen的功能有两个：一是为I0卡分配资源，进行初始化；二是返回相关I0卡的信息，包括初始化成功的I0卡的数量，每个卡的卡号。

关闭I0卡函数Plt_CardClose的作用是在退出时释放资源，在应用程序关闭时应该先调用此函数

例子（C#）：参考第四章的例子

5.2 IO 基本操作

表5.4 IO基本操作指令列表

指令	说明
Plt_IoReadInputByBit	读取输入口的状态
Plt_IoWriteOutputByBit	设置输出口的状态
Plt_IoReadOutputByBit	读取输出口的状态
Plt_IoReadAllInput	读取所有输入口的值
Plt_IoReadAllOutput	读取所有输出端口的值
Plt_IoWriteAllOutput	设置所有输出端口的值
Plt_IoReverseOutputBit	翻转输出出口并保持设定时间

表5.5 运动状态检测指令参数说明

Plt_IoReadInputByBit (uint16 cardid,uint16 bitno,uint16 *active_level)	
cardid	卡号 (0-3)
bitno	IO 号 (0-7)
active_level	IO 状态 (1: 高电平, 0: 低电平)
Plt_IoWriteOutputByBit (uint16 cardid,uint16 bitno,uint16 active_level)	
cardid	卡号 (0-3)
bitno	IO 号 (0-7)
active_level	IO 状态 (1: 高电平, 0: 低电平)
Plt_IoReadOutputByBit (uint16 cardid,uint16 bitno,uint16 *active_level)	
cardid	卡号 (0-3)
bitno	IO 号 (0-7)
active_level	IO 状态 (1: 高电平, 0: 低电平)
Plt_IoReadAllInput (uint16 cardid,DWORD *active_level_1)	
cardid	卡号 (0-3)
active_level_1	所有输入口的状态 (bit0 对应 IN0, bit7 对应 IN7)
Plt_IoReadAllOutput (uint16 cardid,DWORD *active_level)	
cardid	卡号 (0-3)
active_level	所有输出口的状态 (bit0 对应 OUT0, bit7 对应 OUT7)
Plt_IoWriteAllOutput (uint16 cardid,DWORD active_level)	
cardid	卡号
active_level	所有输出口的状态 (bit0 对应 OUT0, bit7 对应 OUT7)
Plt_IoReverseOutputBit(uint16 cardid,uint16 bitno,double hold_time)	
cardid	卡号 (0-3)
bitno	IO 号 (0-7)
Hold_time	翻转保持时间 (单位: s)

重点说明:

输入输出可以逐个口进行操作。

Plt_I0_IoReadInputByBit: 逐个读取输入口状态

Plt_IO_IoWriteOutputByBit: 逐个写入输出口状态
 Plt_IO_IoReadOutputByBit: 逐个读取输出口状态
 输入输出口也可以同时操作所有的输入或输出口
 Plt_IO_IoReadAllInput: 同时读取所有输入口状态
 Plt_IO_IoWriteAllOutput: 同时写入所有输出口状态
 Plt_IO_IoReadAllOutput: 同时读取所有输出口状态

5.3 CAN 扩展模块的使用

表5.6 CAN扩展指令列表

指令	说明
Plt_IoCanReadInputByBit	读取输入口的状态
Plt_IoCanWriteOutputByBit	设置输出口的状态
Plt_IoCanReadOutputByBit	读取输出口的状态
Plt_IoCanReadAllInput	读取所有输入口的值
Plt_IoCanReadAllOutput	读取所有输出端口的值
Plt_IoCanWriteAllOutput	设置所有输出端口的值
Plt_IoCanGetLinkState	读取 CAN 模块连接状态

表5.7 CAN扩展指令参数说明

Plt_IoCanReadInputByBit(uint16 cardid,uint16 can_id,uint16 bitno,uint16 *active_level)	
cardid	卡号 (0-3)
Can_id	CAN 编号 (0-3)
bitno	IO 号 (0-15)
active_level	IO 状态 (1: 高电平, 0: 低电平)
Plt_IoCanWriteOutputByBit(uint16 cardid,uint16 can_id,uint16 bitno,uint16 active_level)	
cardid	卡号 (0-3)
Can_id	CAN 编号 (0-3)
bitno	IO 号 (0-15)
active_level	IO 状态 (1: 高电平, 0: 低电平)
Plt_IoCanReadOutputByBit(uint16 cardid,uint16 can_id,uint16 bitno,uint16 *active_level)	
cardid	卡号 (0-3)
Can_id	CAN 编号 (0-3)
bitno	IO 号 (0-15)
active_level	IO 状态 (1: 高电平, 0: 低电平)
Plt_IoCanReadAllInput(uint16 cardid,uint16 can_id,uint16 *active_level_1)	
cardid	卡号 (0-3)
Can_id	CAN 编号 (0-3)

active_level_1	所有输入口的状态（bit0 对应 IN0，bit15 对应 IN15）
Plt_IoCanReadAllOutput(uint16 cardid,uint16 can_id,uint16 *active_level)	
cardid	卡号（0-3）
Can_id	CAN 编号（0-3）
active_level	所有输出口的状态（bit0 对应 OUT0，bit15 对应 OUT15）
Plt_IoCanWriteAllOutput(uint16 cardid,uint16 can_id,uint16 active_level)	
cardid	卡号（0-3）
Can_id	CAN 编号（0-3）
active_level	所有输出口的状态（bit0 对应 OUT0，bit15 对应 OUT15）
Plt_IoCanGetLinkState(uint16 cardid,uint16 can_id,uint16 *link_state)	
cardid	卡号（0-3）
Can_id	CAN 编号（0-3）
Link_state	CAN 总线连接状态（1：连接 0：断开）

当EIO-3232提供的IO数量不够时，可以使用CAN模块扩展获得更多的IO口，具体使用与EIO-3232上本地IO使用方式相同，只是需要增加一个CAN编号参数，表示操作的是哪一个CAN模块中的IO。

EIO-3232最多可以连接4个CAN扩展模块，4个模块的CAN编号分别为：0、1、2、3。

CAN模块的具体介绍及接线等，请参考附A中

第六章 同时使用多张卡

6.1 使用网卡连接多卡（以四口网卡为例）

步骤1、各卡设置不同的IP地址。

例如：通过demo软件设置各卡的IP地址分别为192.168.167.120，
192.168.167.121，192.168.167.122

注：修改IP地址的方法请见附B

步骤2、各卡通过网线连接到网卡的各口，设置网卡各口的IP地址。

例如：网卡各口的IP地址分别设置为192.168.167.12，192.168.167.13，
192.168.167.15

步骤3、编写代码，参照4.1节内容

```
static void Main(string[] args)
{
    UInt16 plt_CardNum = 3; //开启3张卡

    UInt16[] plt_CardID = new UInt16[8];

    UInt16[] Section = new UInt16[8];

    UInt16[] Host_id = new UInt16[8];

    plt_CardID[0] = 0;

    //第一张卡的IP地址

    Section[0] = 167; //第三段设置为167(固定为167)

    Host_id[0] = 120; //第四段设置为120（根据实际情况设置，卡出厂默认是120）

    //第二张卡的IP地址

    Section[1] = 167;

    Host_id[1] = 121;

    //第三张卡的IP地址

    Section[2] = 167;

    Host_id[2] = 122;

    int ret = csEIO.EIO.Plt_CardOpen(plt_CardNum, plt_CardID, Section, Host_id);
```

```
if (ret != 0)//打开控制卡
{
    Console.Write("打开网口卡失败");
}
else
    Console.Write("打开网口卡成功");

Console.ReadLine();
}
```

控制台显示“打开网口卡成功”，表示3张卡都开启正常。

6.2 使用交换机连接多卡

步骤1、各卡设置不同的IP地址。

例如：通过demo软件设置各卡的IP地址分别为192.168.167.120，
192.168.167.121，192.168.167.122

步骤2、各卡通过网线连接到交换机上，交换机通过网线连接到PC。设置PC的IP地址。

例如：PC的IP地址设置为192.168.167.10

步骤3、编写代码，同6.1节代码一样

6.3 使用 demo 软件控制多卡（网卡或交换机连接）

步骤1、各卡设置不同的IP地址，参考6.1节

步骤2、设置PC端的IP地址，参考6.1节或6.2节

步骤3、

a、开启一个demo，控制卡1（假设前面设置卡1的IP地址为192.168.167.120）
卡0第四段IP设置为120，建立连接，返回连接成功



IP设置

卡数	1	卡0第四段IP	120
第一段IP	192	卡1第四段IP	121
第二段IP	168	卡2第四段IP	122
第三段IP	167	卡3第四段IP	123

建立连接

图6.1 建立连接

b、开启第二个demo，控制卡2（假设前面设置卡1的IP地址为192.168.167.121）

卡1第四段IP设置为121，建立连接，返回连接成功。

步骤4、可以在两个demo软件下操作各自的控制卡。

附 A、CAN 总线 IO 扩展板说明

睿控 CAN 扩展模块使用单电源供电，内部采用隔离电源减少外部干扰，提高系统稳定性。并且模块都已预留终端电阻选择跳线，客户可通过板上预留的跳线帽进行选择是否接入终端电阻。

附 A.1、RCAN-16I16O

RCAN_16I16O 是 16 入 16 出 IO 模块，主要有如下硬件接口：

序号	接口名称	个数
1	24V 电源输入	1 路
2	CAN 总线接口	1 路
3	通用输入	16 路
4	通用输出	16 路
5	栈号及 CAN 速度选择拨码	1 个 1*6 位拨码
6	终端电阻选择跳线	1 个 1*3 位插针

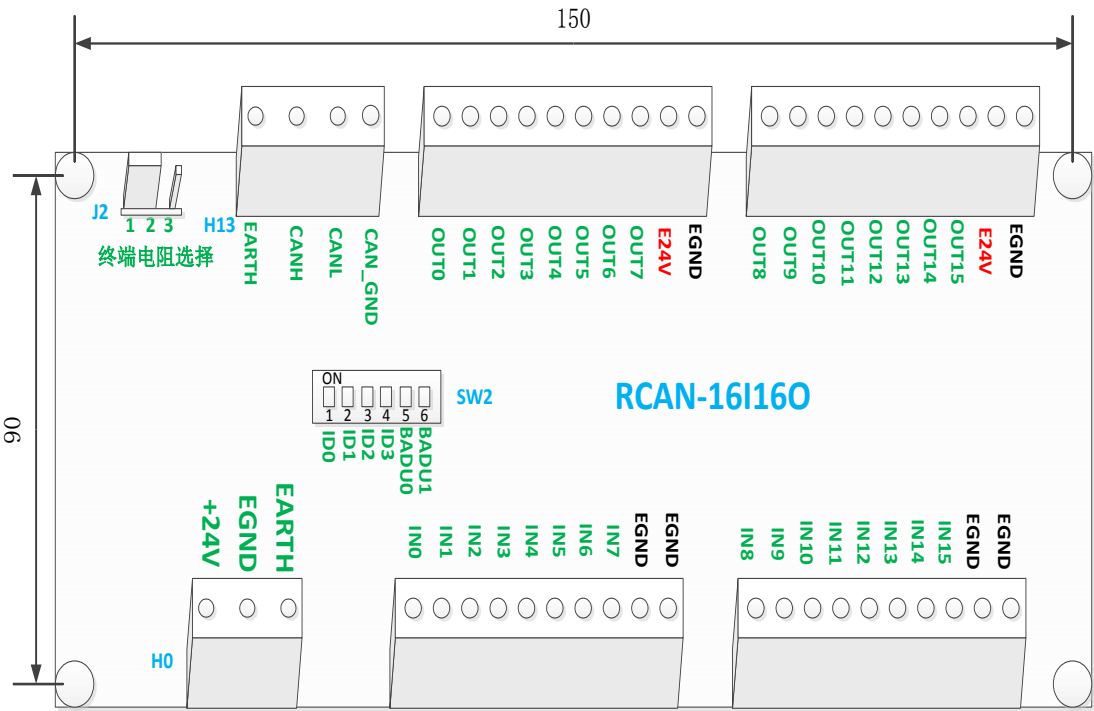


图 A.1 RCAN-16I16O 外形尺寸图

附 A.2、电源接口

表 A.1 电源接口信号说明

序号	信号名称	说明
1	+24V	外部+24V 电源输入
2	EGND	外部+24V 电源地
3	EARTH	安规地/屏蔽地

注意：建议将 EARTH 信号接入系统中，减少信号干扰。

附 A.3、CAN 总线接口

表 A.2 CAN 总线接口信号说明

序号	信号名称	说明
1	EARTH	安规地/屏蔽地
2	CANH	CAN 差分数据+
3	CANL	CAN 差分数据-
4	CAN_GND	CAN 信号地

注意：建议使用带屏蔽的线缆连接 CAN 总线，并将屏蔽线接入系统中。

附 A.4、CAN ID 号及 CAN 速度选择拨码开关

SW2 是 6 位拨码开关，其中前面 4 位为栈号选择（ID0-ID3），后面 2 位为 CAN 速度选择（BADU0-BADU1）。拨码拨到 ON 表示对应值为‘1’，OFF 表示对应值为‘0’，1-6 位出厂默认是 OFF 状态，具体信号说明如下：



图 A.2 拨码示意图

表 A.3 栈号拨码说明

拨码第 2 位	拨码第 1 位	组合值	说明
OFF	OFF	0	CAN ID 号 0
OFF	ON	1	CAN ID 号 1
ON	OFF	2	CAN ID 号 2
ON	ON	3	CAN ID 号 3

注意：多个模块连接时请勿设置相同的栈号。

表 A.4 CAN 速度拨码说明

拨码第 6 位	拨码第 5 位	组合值	说明
OFF	OFF	0	速度 500KBPS
OFF	ON	1	速度 250KBPS

ON	OFF	2	速度 125KBPS
ON	ON	3	速度 1MBPS

注意：CAN 速度根据总线传输距离合理的选择，距离越远其通信速度需要设置的越低，并且同一网络中 CAN 速度必须设定为相同的值。

附 A.5、终端电阻选择接口

RCAN-16I16O 预留了一个终端电阻选择接口 J2，当多个模块级联时最后一个模块需要接入 120 欧的终端电阻，此时通过跳线帽接到 J2 的 2、3 脚就可选择接入终端电阻，无须使用者额外接入电阻。其说明如下：



图 A.3 终端电阻默认未接入状态



图 A.4 终端电阻接入状态

表 A.5 终端电阻设置说明表

跳线帽位置	说明
J2 的 1、2 脚	出厂默认设置，未接入终端电阻
J2 的 2、3 脚	终端电阻接入状态

注意：请在 CAN 总线的最后一个模块通过跳线帽接入终端电阻。

附 A.6、通用输入接口

RCAN-16I16O 共有 16 个通用输入接口，并在 IO 端子上预留了 4 个 24V 地信号(EGND)，接线示意图及具体信号说明如下：

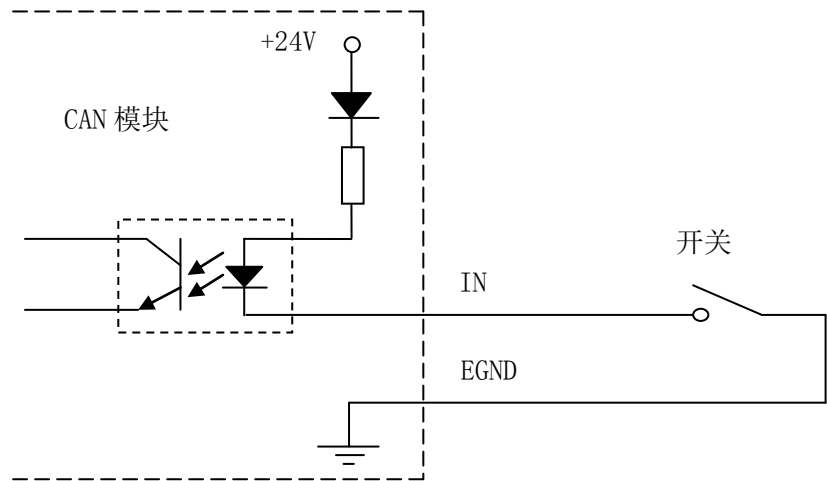


图 A.5 输入信号接线示意图

表 A.6 H7~H9 为低 8 位通用输入

端子序号	名称	信号说明
0	IN0	通用输入 0
1	IN1	通用输入 1
2	IN2	通用输入 2
3	IN3	通用输入 3
4	IN4	通用输入 4
5	IN5	通用输入 5
6	IN6	通用输入 6
7	IN7	通用输入 7
8	EGND	+24 电源地
9	EGND	+24 电源地

表 A.7 H10~H12 为高 8 位通用输入

端子序号	名称	信号说明
0	IN8	通用输入 8
1	IN9	通用输入 9
2	IN10	通用输入 10
3	IN11	通用输入 11
4	IN12	通用输入 12
5	IN13	通用输入 13
6	IN14	通用输入 14
7	IN15	通用输入 15
8	EGND	+24 电源地
9	EGND	+24 电源地

附 A.7、通用输出接口 H1~H6

RCAN-16I16O 共有 16 个通用输出接口,并在 IO 端子上预留了 E24 电源和 E24V 电源地(EGND)接口,接线示意图及具体信号说明如下:

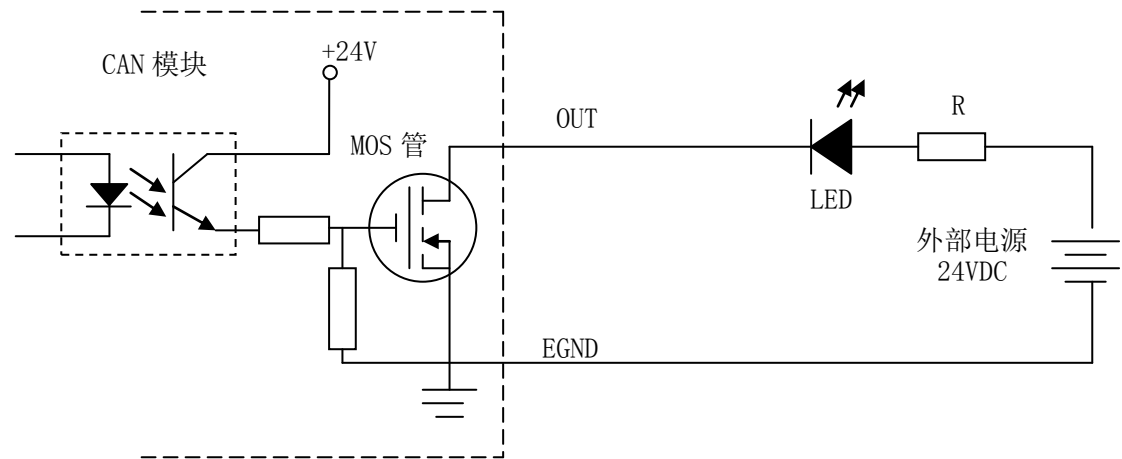


图 A.6 输出信号接线示意图

表 A.8 H1~H3 为低 8 位通用输出

端子序号	名称	信号说明
0	OUT0	通用输出 0
1	OUT 1	通用输出 1
2	OUT 2	通用输出 2
3	OUT 3	通用输出 3
4	OUT 4	通用输出 4
5	OUT 5	通用输出 5
6	OUT 6	通用输出 6
7	OUT 7	通用输出 7
8	E24V	+24 电源输出
9	EGND	+24 电源地

表 A.9 H4~H6 为高 8 位通用输出

端子序号	名称	信号说明
0	OUT 8	通用输出 8
1	OUT 9	通用输出 9
2	OUT 10	通用输出 10
3	OUT 11	通用输出 11
4	OUT 12	通用输出 12
5	OUT 13	通用输出 13
6	OUT 14	通用输出 14
7	OUT 15	通用输出 15
8	E24V	+24 电源输出
9	EGND	+24 电源地

附 B、修改 IP 地址的方法

步骤 1、开卡

打开 demo，选择“IO 卡连接”-->“手动连接”（如图附 B.1），设置卡的 IP 地址，如图附 B.2 所示，卡 0 第四段 IP 设置为 120（卡出厂设置是 120，如果修改过，请根据修改的 IP 进行设置），点击建立连接，返回图附 B.3 表示卡与 PC 建立连接成功。

注：PC 端的 IP 地址也需要进行修改，请参照第四章中的方法进行设置



图 B.1 建立连接



图 B.2 建立连接

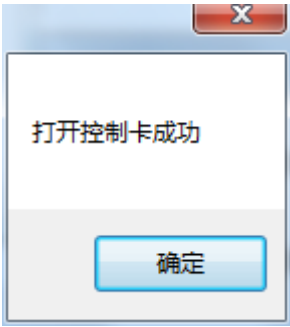


图 B.3 建立连接成功

步骤 2、修改 IP 地址

点击“IO 卡测试”-->“重置当前卡 IP 地址”，如图图附 B.4，

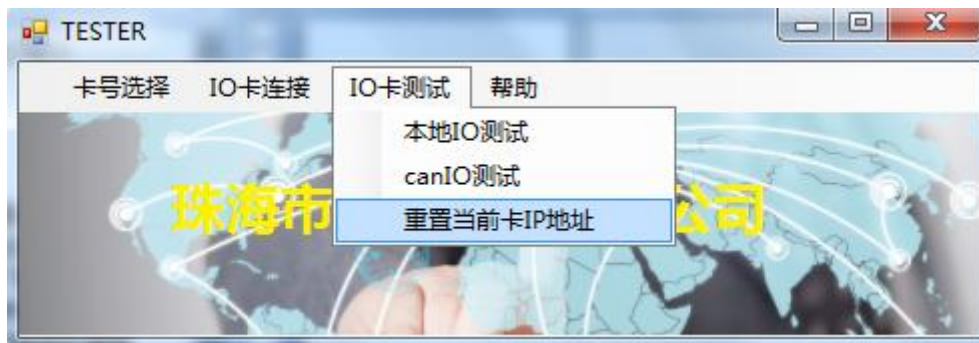


图 B.4 进入重置 IP 界面

“IP 读取”的第四段是当前的 IP 地址第四段，与前一步开卡的值相同

“IP 设置”的第四段是需要的新的 IP 地址。

点击“设置 IP”，即可完成设置。



图 B.5 重置 IP

步骤 3、EIO-0808 断电重启，此时需要用新的 IP 地址（此例是 192.168.167.122）进行开卡动作。

附 C、Demo 调试 EIO-0808 的步骤

附 C.1、建立物理连接

通过网线连接 PC 与 EIO-0808（默认 IP 地址：192.168.167.120），控制卡通电。

附 C.2、设置 PC 机的 IP 地址

设置 PC 机的 IP 地址与 EIO-0808 在同一网段，如下图所示，第四段在 1 到 119 之间选择。

☒ 使用下面的 IP 地址(S):

IP 地址(I):

192 . 168 . 167 . 18

子网掩码(U):

255 . 255 . 255 . 0

图 C.1 PC 端设置 IP 地址

附 C.3、开启 demo 软件

双击下面的图标，开启 demo 软件。

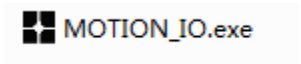


图 C.2 demo 软件图标

附 C.4、选择需要打开卡的 IP 地址，建立连接。

填入待操作卡的 IP 地址第四段（此例为 120），点击建立连接，返回图 C.5 提示即开卡成功。



图 C.3 建立连接的菜单

IP设置

卡数

1

卡0第四段IP

120

第一段IP

192

卡1第四段IP

121

第二段IP

168

卡2第四段IP

122

第三段IP

167

卡3第四段IP

123

建立连接

图 C.4 填写要打开卡的 IP 地址

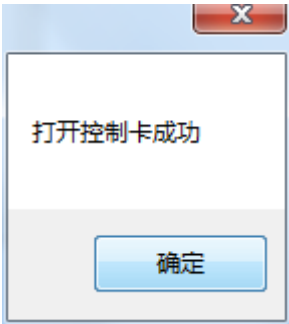


图 C.5 开卡成功的提示

附 C.5、IO 操作测试

进入"本地 IO"界面，可以进行输入与出口的操作和检测。



图 C.6 进行 IO 测试的菜单选择



图 C.7 IO 测试界面

附 C.6、进行从站 IO 操作（如果有接入 CAN 总线 IO 模块）

若连有 CAN 总线从站，则可以进入“canio”界面进行从站的 IO 操作。操作前需要选择从站的 ID 号（相关说明请见附 A.4）。界面左下角的“can 连接状态”显示为绿色图标时表示从站连接正常，可以正常操作。



图 C.8 进行 IO 测试的菜单选择



图 C.9 从站 IO 测试界面