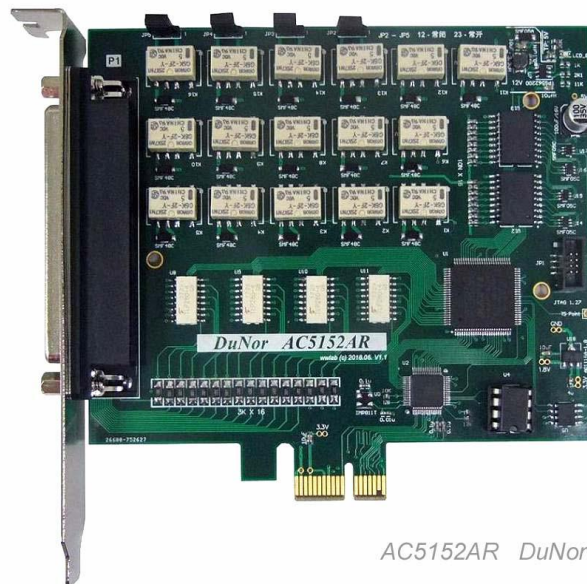


AC5152AR

使用手册

- ❑ PCI-E 总线继电器板
- ❑ 兼容 AC5152
- ❑ 16 路隔离输入（支持共阴、共阳模式）
- ❑ 独立 2 组共 16 路继电器输出
- ❑ 32byte 用户 EEPROM
- ❑ AC5152AR 函数完全兼容 AC5152。



在开始使用前请仔细阅读下面说明

检查

打开包装请查验如下：

- ✧ AC5152AR卡一个
- ✧ 用户指南及光盘
- ✧ DB37插头一套

安装

- ✧ 禁止带电插拔板卡。
- ✧ 详细安装请见第二章。

保修

用户需要遵守储存、运输和使用的要求。因违反操作规定和要求而造成损坏的，需缴纳器件费和维修费及相应的运输费用，如果板卡有明显烧毁、烧糊情况原则上不予维修。如果板卡开箱测试有问题，可以免费维修（限购买板卡 7 天内）。

具体的维修细则请参看附录。

本公司不承担由于运输过程产生的板卡损坏的问题。

版权信息

保留在不另行通知的情况下，更新手册的权利。

不承担由于使用产品不当，所造成的直接、间接、附带的或相应产生的损失或责任。

本产品及其软件具有专利权、版权及其他知识产权。未经授权，不得直接或间接复制、制造、加工本产品及其相关部分。

应用范围

本产品设计制造用于普通工业应用，不能用于易燃易爆等危险场合，超出预料的用途或对人的生命或财产造成重大影响的情况不在本产品应用和服务的范围内。

软件支持服务

自销售之日起提供 **3 个月** 的免费开发咨询。

目录

在开始使用前请仔细阅读下面说明.....	2
目录.....	3
版本信息.....	5
一、AC5152AR 说明.....	1
1.1 AC5152AR 板卡简介.....	1
相关产品.....	1
配套端子板.....	1
1.2 主要特点、性能.....	2
输入特性.....	2
输出特性.....	3
EEPROM 读写.....	3
其他.....	3
1.3 原理说明.....	4
输入.....	4
输出.....	5
二、软件安装.....	7
2.1 驱动安装.....	7
三、硬件安装与连接.....	1
3.1 安装.....	1
3.2 连接器插座定义.....	1
P1: DB37 输出插座.....	1
3.3 JP2-JP5: 继电器常开、常闭选择.....	2
3.4 配套端子板.....	2
3.5 常用信号的连接、处理.....	2
输出驱动继电器.....	3
输出与 PLC 相连.....	3
共阴输入接线图.....	4
共阳输入接线图.....	4
四、软件开发.....	5
4.1 说明.....	5
4.2 接口函数说明.....	6
函数简介.....	6
约定.....	6
设备操作函数.....	7
开关量操作函数.....	7

EEPROM 读写操作函数.....	9
4.3 VC 程序编程说明.....	10
4.4 Labview 程序编程说明.....	11
4.5 C#程序编程说明.....	11
五、附录.....	13
5.1 维修.....	13
5.2 板卡尺寸示意图.....	13
5.3 IO 译码地址表.....	14

版本信息

修改日期	版本号	修改内容
2018.06	1.0	原版

一、AC5152AR 说明

1.1 AC5152AR 板卡简介

AC5152AR是一款低价格通用光电隔离I/O板，具有16路隔离输入（支持共阳、共阴，**默认为共阴输入**）和16路继电器输出（分为2组独立，每组8路；16路继电器其中4路可选择常开或常闭）。板上32byte EEPROM可以供用户记录数据。采用PCI-E总线，支持即插即用，无需地址跳线。采用大规模可编程门阵列设计，提高可靠性。

AC5152AR的输入支持9~24V输入。输出为继电器输出，输出驱动电流大于500毫安，输出可以兼容NPN或PNP模式，同时支持交流控制。

AC5152AR函数完全软件兼容AC5152，AC5152程序可以直接兼容AC5152AR（需要重新安装AC5152AR驱动）。

相关产品

PCI-E 总线

AC5152 16路隔离输入；16路隔离输出（NPN输出）

PCI 总线

AC6650: PCI总线，8路隔离输入、8路继电器输出。

AC6652AR: PCI总线，16路隔离输入、16路继电器输出。

USB 总线

MP351AC: 16路隔离输入、16路NPN输出。

MP351AR: 16路隔离输入、16路继电器输出。

配套端子板

AC142A: DB37插座、螺丝端子连接板。

1.2 主要特点、性能

AC5152AR 函数完全软件兼容 AC5152，AC5152AR 原理框图主要由输入隔离模块、输出继电器模块、门阵列控制模块和 PCI-E 总线接口模块组成。

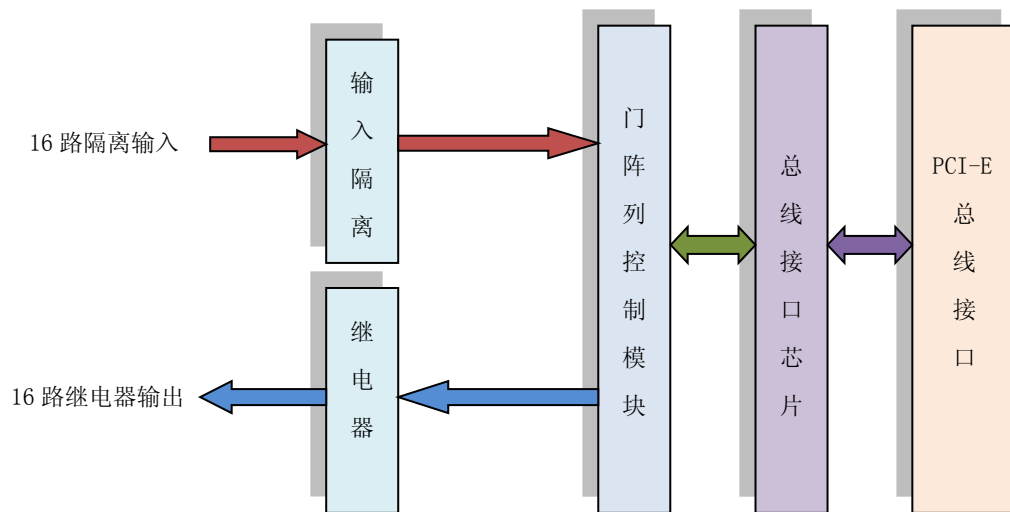


图 1.1 原理框图

输入特性

- 16路隔离输入，分为2组，每组8路独立隔离输入。每组有独立的公共端。
- 隔离电压大于：500伏（绝缘电阻大于500M Ω ）。
- 输入电压范围：9~24伏（电流约：3~8mA）。
- 输入支持共阴、共阳模式。
- 输入最大频率：5KHz。
- 悬空状态为断开。
- 响应时间

电平变化	响应时间（uS）
高->低	5
低->高	50

注：

1. 以上测试为典型数值。

2. 测试条件：输入电压9V。

输出特性

- 16 路继电器输出。继电器分为 2 组，每组 8 路。每组公共端相互隔离。
- 隔离电压：大于 500V。
- 输出继电器开关：默认为常开状态。其中，高 4 路可选择常开、常闭。
- 输出最大负载电流：标准 500 毫安，最大小于 1A/路。每组 8 路总电流小于 4A。
- 输出工作电压：36VDC/28VAC。
- 每个输出节点对公共端，并接有一个 48V 的双向 TVS 保护二极管，为感性负载提供保护。
- 动作时间： $\leq 5\text{ms}$ 。
- 继电器寿命：大于 1,000,000 次。

EEPROM 读写

AC5152AR支持EEPROM读写功能。用户可以利用EEPROM函数向AC5152AR板卡内部的flash rom记录1个32byte长的数据，做为软件序号、用户板卡使用记录或信号调整参数等应用。保存数据掉电后不消失，并且可以保存10年有效！

其他

- 总线：x1 PCI-E总线。
- 软件支持：windows XP以上的32/64位系统。
- 供电：单3.3V / 12V。
- 板卡尺寸：11.2cm X 10.4cm（长X高）。
- AC5152AR函数完全兼容AC5152，AC5152程序可以直接兼容AC5152AR（需要重新安装AC5152AR驱动）

1.3 原理说明

AC5152AR 采用 PCI-E 接口芯片 CH 系列及门阵列作为主控芯片。

输入

16 路隔离输入分为 2 组，每组 8 路。

第一组：输入通道 0-7 号。

第二组：输入通道 8-15 号。

各个组之间相互隔离，拥有独立的公共端（共阴或共阳）。**如果不要输入分组，必须把两组公共端都连接到一起。**

输入原理

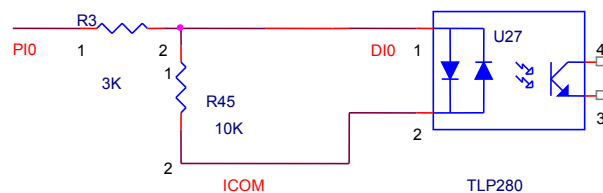


图 1.2 输入结构

输入部分：输入原理见图 1.2。输入电压经过电阻 R3 到光耦的输入，经隔离输出到门阵列芯片的输入。输入电流为：

$$I_{in} = (V_{in} - 1.2) / 3(\text{mA})。$$

共阴输入：如图 1.3。输入公共端（ICOMA 或 ICOMB）连接外部信号地线。当外部输入一个有效电平时，用户在对应位读入为“0”；当输入悬空时，用户在对应位读入为“1”。

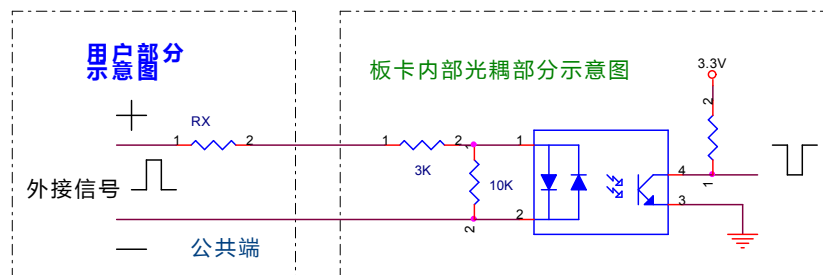


图 1.3 共阴输入原理及接线图

如果输入电流过大，可以在输入端额外串接一个限流电阻 RX，但必须保证输入电流大于 1 毫安。

共阳输入：以检测机械开关为例，如图 1.4。输入公共端连接外部信号的电源“+”端。当外部输入=0 时，用户在对应位读入为“0”。当输入悬空或等于外部电源时，读入的对应位为“1”。**注意：共阳输入的读出状态与共阴输入的读出状态相反。**

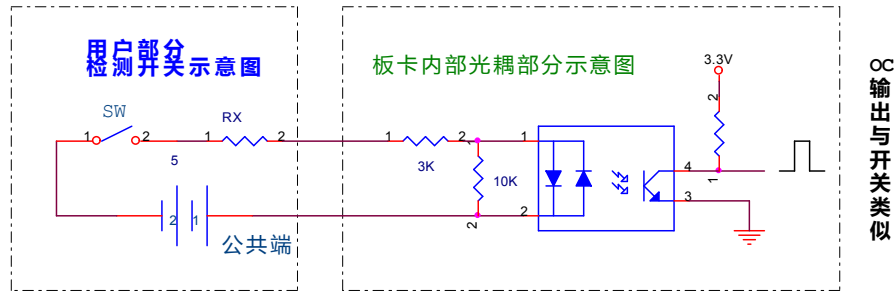


图 1.4 共阳输入原理及接线图

共阳输入主要应用于 OC 输出、机械开关检测方面。当 OC 输出有效 (=0) 时，输入被拉到低电平 (GND)，此时隔离光耦有效，对应读入位=0。

输出

16 路继电器输出分为 2 组，每组 8 路。

第一组：输出通道 0-7 号。

第二组：输出通道 8-15 号。其中，高 4 路（12-15）支持常开、常闭可选。具体方法请参考“2.2 连接器插座定义”。

各个组之间相互隔离，拥有独立的公共端。如果用户不要求分组，必须把所有的公共端都连接到一起。

输出原理

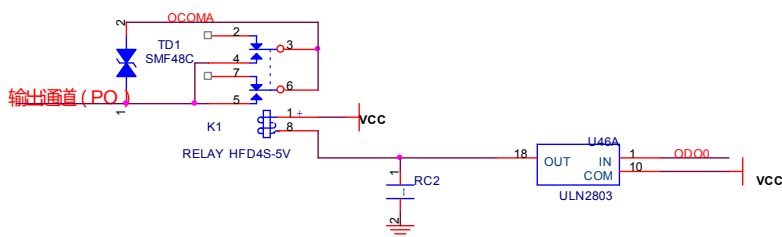


图 1.5 输出原理

注：

1. OCOMA：输出公共端。
2. PO：输出端。
3. TD1：双向 TVS 二极管，动作电压 48V

继电器输出默认为常开端。当设置对应通道输出为“0”时，继电器不动作，保持常开状态；当设置对应通道输出为“1”时，继电器触点与常开端吸合，构成回路。

输出可以驱动 30V 直流/交流负载，共阴或共阳方式均可；也可以作为继电器多路选择开关。

输出接线图请参考“2.4 常用信号的连接电路”。

二、软件安装

2.1 驱动安装

资料中提供的驱动程序，支持 winXP、win7、win8 和 win10 等系统。

- ✧ driver: 32 位系统的板卡驱动。
- ✧ driver x64: 64 位系统的板卡驱动。

以下是驱动的安装方法，请严格按照以下步骤安装驱动。

注意：禁止带电插拔板卡。

XP 系统

1. **关闭计算机。**将 AC5152AR 插入一个 PCI-E 插槽，并将外部输入、输出线连接好。如果有多个 AC5152AR 插卡，请每一次只安装一个 AC5152AR 插卡。PCI-E 插槽序号最低的卡的设备号为“0”，以此类推。
2. 启动计算机。
3. windows 将会显示驱动安装向导，选择“从列表或指定位置安装（高级）”，下一步。
4. 选择“不要搜索。我要自己选择要安装的驱动程序”，下一步。
5. 选择“显示所有设备”。注意：有的客户的计算机上可能没有这个界面，如果没有，跳过此步。
6. 选择“从磁盘安装”，在弹出来的对话框中选择“浏览”，找到驱动文件 AC5152AR.inf（光盘中或本地目录中）。
下一步。
7. 等待驱动安装完成，点击“完成”。驱动安装结束。安装完成后，如果 2.1 所示。

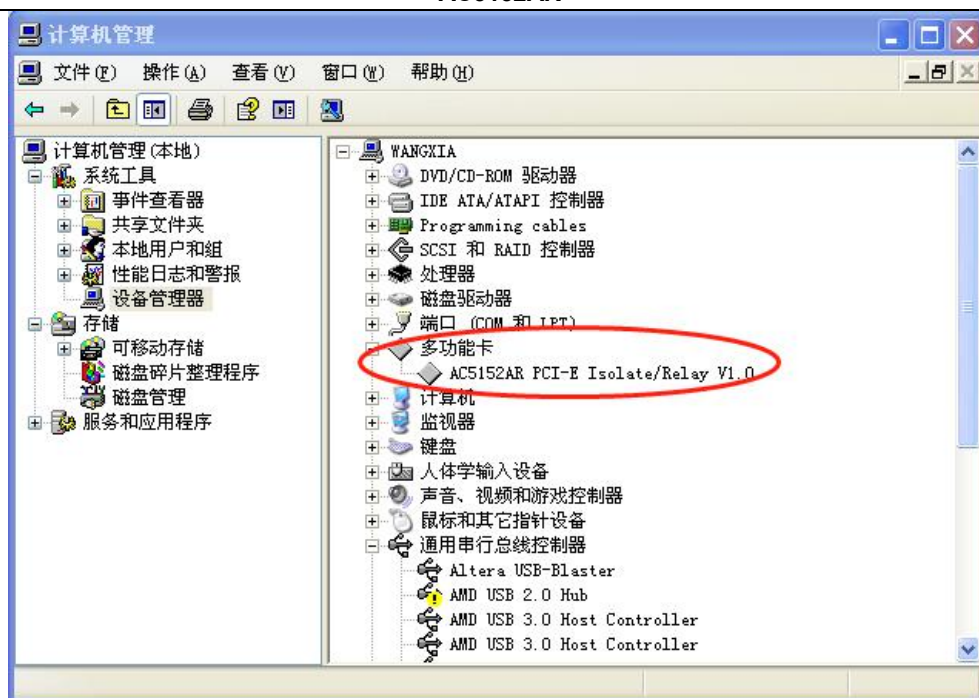


图 2.1 XP 系统下驱动完成效果图

win7 系统

1. 关闭计算机。将 AC5152AR 插入一个 PCI-E 插槽，并将外部输入、输出线连接好。如果有多个 AC5152AR 插卡，请每一次只安装一个 AC5152AR 插卡。PCI-E 插槽序号最低的卡的设备号为“0”，以此类推。
2. 启动计算机。
3. 通常情况下，win7 会自动为该设备寻找驱动程序，此时可以等待系统安装。通常情况下会提示安装不成功。此时需要进行手动安装。
4. 打开“设备管理器”，找到“PCI 数据捕获和信号处理控制器”。右键选择“更新驱动程序软件”。
5. 选择“浏览计算机以查找驱动程序软件（R）”。
6. 选择“从计算机的设备驱动程序列表中选择（L）”。**注意：这一步至关重要，很多用户在 win7 系统下安装不成功均是因为此处选择错误。**
7. 选择“显示所有设备”。**注意：有的客户的计算机上可能没有这个界面，如果没有，跳过此步。**
8. 选择“从磁盘安装”，在弹出来的对话框中选择“浏览”，找到驱动文件 AC5152AR.inf（光盘中或本地目录中）。下一步。
9. 等待驱动安装完成，点击“完成”。驱动安装结束。安装完成后，如图 2.2 所示。

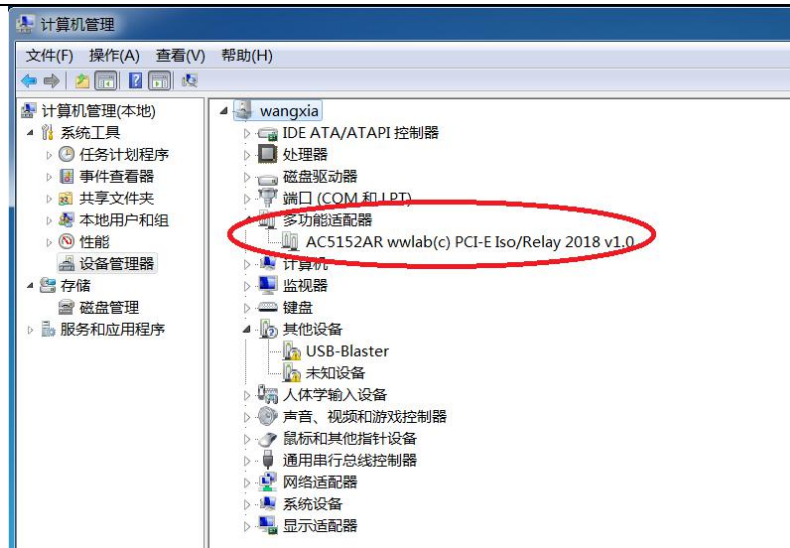


图 2.2 win7_64bit 系统下驱动完成效果图

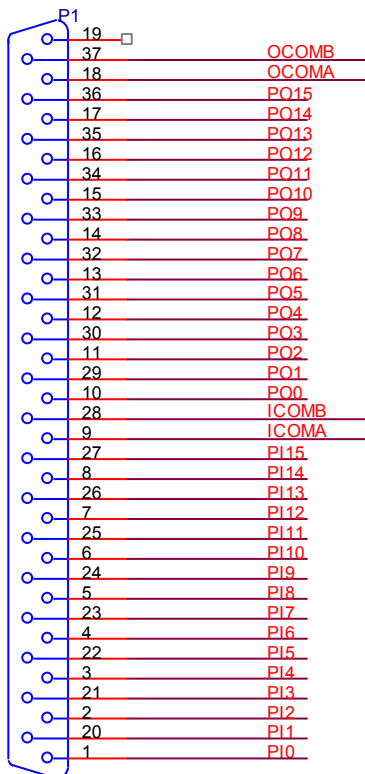
三、硬件安装与连接

3.1 安装

关掉 PC 机电源，将板卡插入主机的任何一个 PCI-E 插槽中并将外部的输入、输出线连好。如果主机有多套插卡，请逐个安装。

3.2 连接器插座定义

P1: DB37 输出插座



DB37RA/F

图 2.3 DB37 插座定义

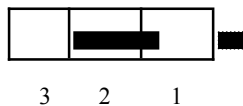
1. PO0~PO15: 对应输出通道 0~15 号。
2. PI0~PI15: 对应输入通道 0~15 号。
3. 脚 9: 低 8 路输入(PI0-PI7)公共端。
4. 脚 28: 高 8 路输入(PI8-PI15)公共端。
5. 脚 18: 低 8 路输出(PO0-PO7)公共端。
6. 脚 37: 高 8 路输出(PO8-PO15)公共端。

分组	公共端	管脚
PI0 – PI7 输入	ICOMA	P9
PI8 – PI15 输入	ICOMB	P28
PO0 – PO7 输出	OCOMA	P18
PO8 – PO15 输出	OCOMB	P37

注：以上端子的引脚定义在板卡背面已标明。

3.3 JP2-JP5：继电器常开、常闭选择

JP2 – JP5：对应选择输出继电器 P012、P013、P014、P015 选择。方法如下：



JP2-JP5 这几个跳线器位于板卡的最上方，可以选择继电器输出是常开还是常闭。

以 JP2 为例，从左到右依次为 3-2-1。

- 1-2：选择对应通道为**常闭触点**。
- 2-3：选择对应通道为**常开触点**（出厂时默认此状态）。

3.4 配套端子板

可以配接 AC142A 端子板。

AC142A

- ◇ 40 路螺丝端子，支持 32 路接线。
- ◇ DB37 或 40 脚扁平电缆插座。

3.5 常用信号的连接、处理

输出驱动继电器

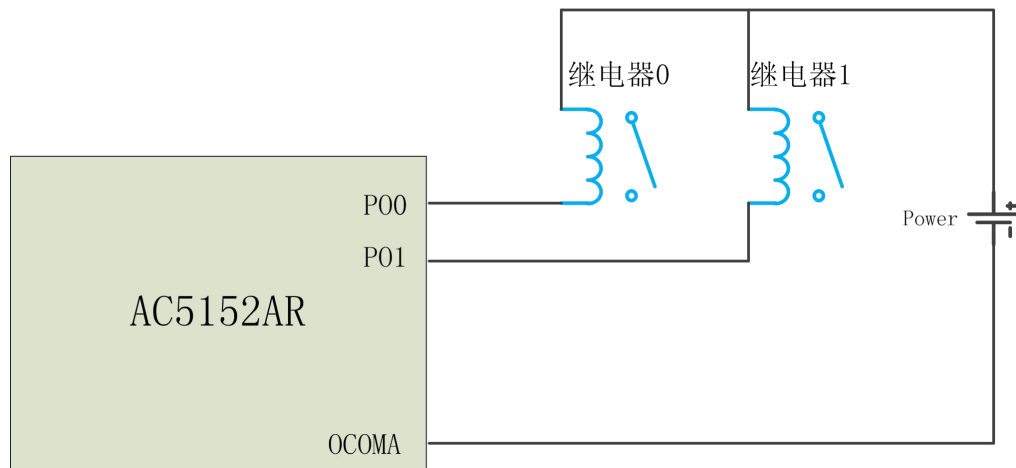


图 2.4 驱动继电器

图 2.4 为输出驱动继电器的原理，以低 8 路为例。Power 根据继电器的型号选择（如 24V），电源地线连接 AC5152AR 继电器输出的公共端。继电器的线圈两端反向并联一个二极管用来吸收线圈的反向电压。如果继电器触点驱动感性负载（如电磁阀），需要在开关触点两端并联一个 0.01uF 的高压电容，以吸收触点开关时产生的火花。

外部继电器或螺旋线圈可以选择共阳或共地。输出驱动电磁阀、气缸等设备的接线方法请参考上图。

输出与 PLC 相连

与共阳 PLC 连接，AC5152AR 的公共端（OCOMA、OCOMB）与 PLC 公共端相连，输出通道（PO0、PO1 • • •）与 PLC 的 X 点输入相连。

与共阴 PLC 连接，外接电源正端与 AC5152AR 的公共端（OCOMA、OCOMB）相连，输出通道（PO0、PO1 • • •）与 PLC 的 X 点相连，外接电源负端与 PLC 输入公共端相连。

共阴输入接线图

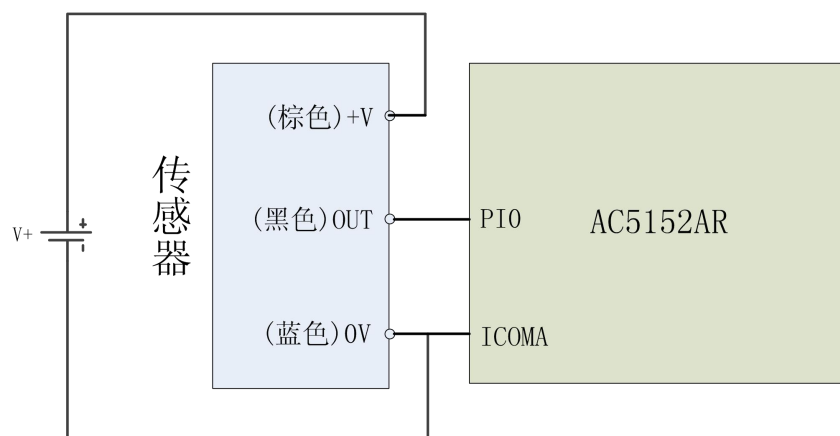


图 2.5 共阴输入接线图-传感器

共阳输入接线图

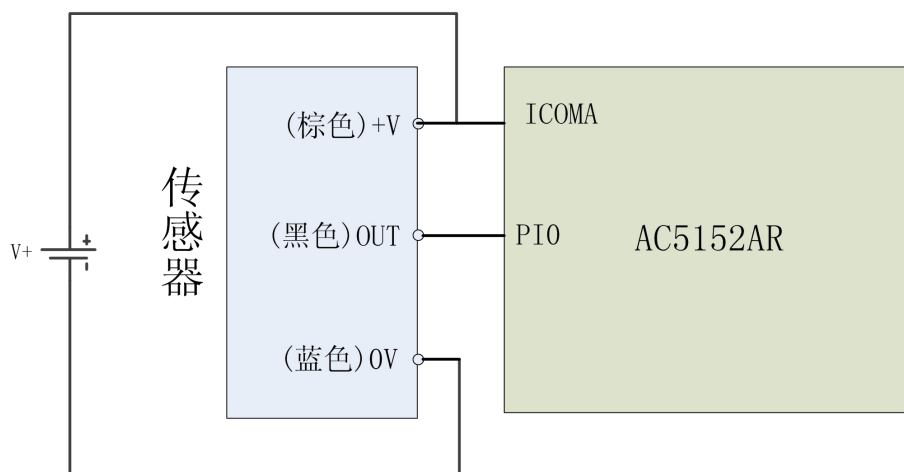


图 2.6 共阳输入接线图-传感器

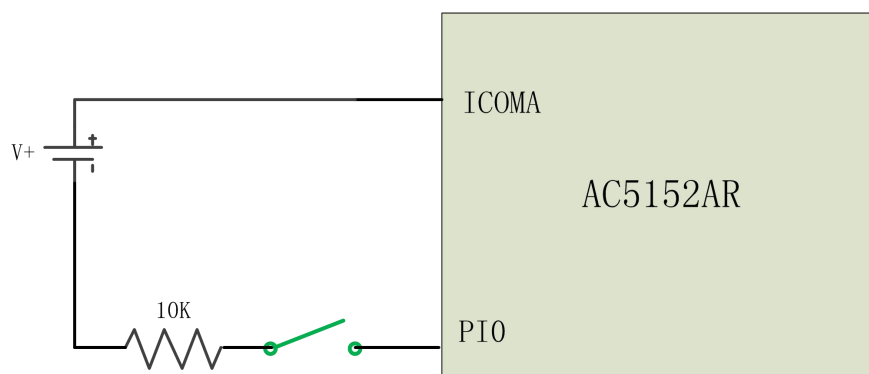


图 2.7 共阳输入接线图-开关

四、软件开发

本章介绍动态链接库函数使用方法以及针对 AC5152AR 的软件开发指导。请用户在编程前，仔细阅读本手册，了解相关信息。

AC5152AR 函数完全软件兼容 AC5152

4.1 说明

注意：板卡需要安装驱动后才能使用。

AC5152AR附带光盘中，提供如下内容：

1. 说明书。
2. 驱动程序，支持windows系统以上的32/64位操作系统。
3. MFC、VC编程实例。
4. LabView、C#编程实例。
5. AC5152AR测试程序。

注：AC5152AR卡的驱动不支持win NT。

- 在光盘的\PCI-E\AC5152AR\driver目录中包含：AC5152AR.inf、AC5152.sys、AC5152.dll、AC5152.lib、AC5152.h等文件。
 - ✧ AC5152 AR.inf：驱动安装文件。
 - ✧ AC5152.sys：驱动程序。
 - ✧ AC5152.dll：动态链接库。
 - ✧ AC5152.lib：VC的库文件。
 - ✧ AC5152.h：VC的头文件。
- 在光盘的\PCI-E\AC5152AR\driver x64文件夹中包含了64位系统下需要的板卡驱动程序。
 - ✧ AC5152AR.inf：驱动安装文件。
 - ✧ AC5152.sys：驱动程序。
 - ✧ AC5152.dll：32位模式动态链接库。
 - ✧ AC5152.lib：32位模式C的库文件。
 - ✧ AC5152DLL64.dll：64位模式动态链接库。
 - ✧ AC5152DLL64.lib：64位模式C的库文件。

- 在光盘的\PCI-E\AC5152AR\VC目录中包含以下文件。
 - ✧ VC的编程例子。
 - ✧ VC编程需要的头文件和lib文件。

- 在光盘的\PCI-E\AC5152AR\MFC目录中包含以下文件。
 - ✧ MFC的编程例子。
 - ✧ 编程需要的头文件和lib文件。

- 光盘的\PCI-E\AC5152AR\C#目录中包含以下文件。
 - ✧ C#的编程例子。
 - ✧ 编程需要的 dlm 文件（函数模块声明）。

- 光盘的\PCI-E\AC5152AR\Labview 目录中包含以下文件。
 - ✧ Labview 的编程例子。

- ac5152ar 用户测试.exe：测试程序。

4.2 接口函数说明

函数简介

AC5152AR 软件函数完全兼容 AC5152。

函数分为：

1. 设备初始化函数。该函数用于初始化串口，并获得操作句柄。
2. IO 操作函数。该函数操作输入输出状态。
3. EEPROM 读写函数。用户可以使用此函数保存数据到板卡上的用户 EEPROM 中。

AC5152AR 通过不同的句柄来区分多个 AC5152AR 或其他设备，应用 AC5152_OpenDevice 函数可以得到一个唯一的句柄。

约定

- HANDLE：操作句柄，等效 32 位有符号数。
- __int32：32 位有符号数。
- char、unsigned char：8 位有符号、无符号数。

- 以下函数以 C 语言方式给出。
- 数据格式为高位在前，低位在后。

设备操作函数

- 打开一个 AC5152AR 设备

函数：HANDLE AC5152_OpenDevice(__int32 DeviceNum)

功能：初始化串口，获得 AC5152AR 板卡的操作句柄。

参数：

- ✧ DeviceNum: 入口参数，AC5152AR 设备号，=0、1、....，表示第一个、第二个、....AC5152AR 插卡。设备号的定义参考驱动安装部分。
- ✧ 函数返回值：卡的操作句柄。

注：VC 中如果句柄不等于 INVALID_HANDLE_VALUE，表示正确。VB 中如果句柄不等于 &HFFFFFFF，正确。

- 关闭一个 AC5152AR 设备

函数：__int32 AC5152_CloseDevice(HANDLE hDevice1)

功能：关闭以 hDevice1 打开的 AC5152AR 卡。

参数：

- ✧ hDevice1: 入口参数，卡的操作句柄。
- ✧ 函数返回数值：0：成功 / -1：失败。

- 读入板卡信息

函数：__int32 AC5152_Info(HANDLE hDevice, char *modle)

功能：读入板卡型号及版本号。

参数：

- ✧ hDevice: 入口参数，卡的操作句柄。
- ✧ *modle: char 指针，存放着板卡信息。
- ✧ 函数返回：板卡信息。

开关量操作函数

AC5152AR 具有 16 路输入、16 路输出。

- 开关量输入

函数：__int32 AC5152_DI (HANDLE hDevice, __int32 ionum)

功能：读入 0-15 号口中一个 8 位口输入数据。

注：AC5152AR 的 16 路输入分为 PORT0、PORT1 等 2 个 8 位接口。

PORT0: 对应输入通道 0~7 号。

PORT1: 对应输入通道 8~15 号。

参数:

- ✧ hDevice: 入口参数, 卡的操作句柄。
- ✧ ionum: 入口参数, 0~1。0-PORT0; 1-PORT1。
- ✧ 函数返回: 出口参数, 返回读入的数据, 低 8 位数据 (D7~D0) 分别对应输入端口的 7~0 号。

□ 开关量输出

函数: __int32 AC5152_DO (HANDLE hDevice, __int32 ionum, __int32 iodata)

功能: 设置 PORT0、PORT1 的数据。

注: AC5152AR 的 16 路输出分为 PORT0、PORT1 等 2 个 8 位接口。

PORT0: 对应输出通道 0~7 号。

PORT1: 对应输出通道 8~15 号。

参数:

- ✧ hDevice: 入口参数, 卡的操作句柄。
- ✧ ionum: 入口参数 0~1。0-PORT0; 1-PORT1。
- ✧ iodata: 入口函数, 输出的数据。数据的低 8 位有效。8 位数据 (D7~D0) 分别对应端口的 8 个 IO 输出口 7~0 号。
- ✧ 函数返回: 出口参数, =0 操作成功, 其他失败。

□ 回读开关量输出

函数: __int32 AC5152_RDO(HANDLE hDevice, __int32 ionum)

功能: 回读 PORT0、PORT1 这两个 8 位口中的一个口的输出数据。

注: AC5152AR 的 16 路输出分为 PORT0、PORT1 等 2 个 8 位接口。

PORT0: 对应输出通道 0~7 号。

PORT1: 对应输出通道 8~15 号。

参数:

- ✧ hDevice: 入口参数, 卡的操作句柄。
- ✧ ionum: =0~1, 分别选择 PORT0、PORT1。
- ✧ 函数返回: 出口参数, 返回对应输出口的数据 (低 8 位有效)。

□ 开关量输入位操作

函数: __int32 AC5152_DIB(HANDLE hDevice, __int32 ionum, __int32 nbit)

功能: 读入 8 位开关量输入的一个端口的状态。

注: AC5152AR 的 16 路输出分为 PORT0、PORT1 等 2 个 8 位接口。

PORT0: 对应输入通道 0~7 号。

PORT1: 对应输入通道 8~15 号。

参数:

- ✧ hDevice: 入口参数, 卡的操作句柄。
- ✧ ionum: =0~1, 分别选择 PORT0、PORT1。
- ✧ nbit: =0~7, 对应选择读入的位。
- ✧ 函数返回: 出口参数, 返回读入的状态。=0: 表示对应输入端口为低电平; =1: 表示对应输入端口为高电平; <0: 表

示函数操作错误。

□ 开关量输出位操作

函数: `__int32 AC5152_DOB(HANDLE hDevice, __int32 ionum, __int32 nbit, __int32 bdata)`

功能: 设置 8 位端口的一个端口的输出状态。

注: AC5152AR 的 16 路输出分为 PORT0、PORT1 等 2 个 8 位接口。

PORT0: 对应输出通道 0~7 号。

PORT1: 对应输出通道 8~15 号。

参数:

- ✧ hDevice: 入口参数, 卡的操作句柄。
- ✧ ionum: =0~1, 分别选择 PORT0、PORT1。
- ✧ nbit: =0~7, 对应选择需要设置的位。
- ✧ bdata: =0~1。对应位的状态。=0: 表示设置成低电平; =1: 表示设置成高电平。
- ✧ 函数返回: 出口参数。=0, 表示函数操作正确; <0: 表示函数操作错误。

EEPROM 读写操作函数

用户可以利用 EEPROM 函数向 AC5152AR 板卡内部的 flash rom 记录 1 个 32byte 长的数据, 做为软件序号、用户板卡使用记录或信号调整参数等应用。保存数据掉电后不消失, 并且可以保存 10 年有效!

□ 写 EEPROM

函数: `__int32 AC5152_EEPROM_WR(HANDLE hDevice, unsigned char *wdata)`

功能: 向 EEPROM 中写入数据。

参数:

- ✧ hDevice: 入口参数, 卡的操作句柄。
- ✧ *wdata: char 指针, 指向一个 32byte 的数组, 数组中存放需要写入 EEPROM 的数据。
- ✧ 函数返回: -1, 失败; 0, 正常。

□ 读 EEPROM

函数: `__int32 AC5152_EEPROM_RD(HANDLE hDevice, unsigned char *rdata)`

功能: 从 EEPROM 中读出数据。

参数:

- ✧ hDevice: 入口参数, 卡的操作句柄。
- ✧ *rdata: char 指针, 指向一个 32byte 的数组, 数组中存放从 EEPROM 读出的数据。
- ✧ 函数返回: -1, 失败; 0, 正常。

注:

关于不同的编程平台具体的编程方法, 请客户参考样例。我们提供了 VC、MFC、Labview 和 C#等例程。

其他例程请用户自行搜索相关资料。

4.3 VC 程序编程说明

编程前，请将 **AC5152.lib** 及 **AC5152.h** 拷贝到用户当前目录中。

VC 编程的基本流程：

1. 利用显式调用加载函数。AC5152.lib、AC5152.h 文件必须在当前工作目录中。在程序的开始处加入以下语句。

```
#pragma comment(lib,"AC5152.lib")
#include"AC5152.h"
```

详细可以参考 VC 目录中的程序，AC5152.h 文件包含了所需函数的声明过程。

2. 利用 AC5152_OpenDevice 函数获得板卡的操作句柄。
3. 在退出程序时必须执行如下操作：利用 AC5152_CloseDevice 函数关闭句柄。

例：

```
//获得所有 AC5152AR 的操作函数
```

```
#pragma comment(lib,"AC5152.lib")
```

```
#include"AC5152.h"
```

```
HANDLE hDevice; //硬件操作句柄
```

```
main()
```

```
{
```

```
    //获得 AC5152AR 硬件操作句柄
```

```
    hDevice=AC5152_OpenDevice(0); //创建设备驱动句柄，设备号为 0
```

```
    ..... //用户程序
```

```
    //退出
```

```
    AC5152_CloseDevice(hDevice); //关闭操作句柄
```

```
}
```

详细可以参考光盘上的 AC5152AR 的 VC 目录下的例子。

在编程时必须注意，硬件操作句柄 HANDLE 必须为全局变量或必须传递给有相应硬件操作的函数。硬件句柄只需在程序启动时打开一次即可，不需要每次打开或关闭。

4.4 Labview 程序编程说明

本公司生产的所有采集卡的相关接口函数，均以动态链接库的形式提供给用户。在使用 Labview 对本公司采集卡进行开发时，只需通过 Labview 中的 Call Library Function Node 节点来调用我们提供的动态链接库函数即可对硬件进行相关操作。

Labview 中提供了将 DLL 的函数转换成 Labview 库的功能。用户在生成函数库时请注意相应的数值定义。

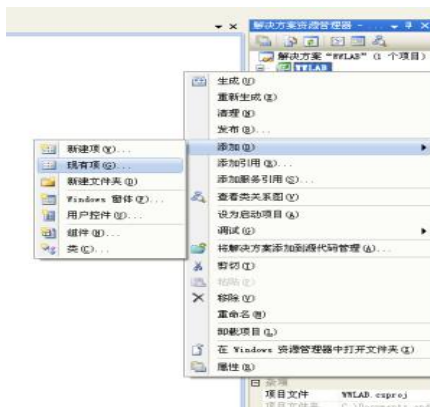
- `__int32`: 32 位有符号数。
- `HANDLE`: 操作句柄，等效一个 32 位有符号数。
- `unsigned char`: 无符号 8 位整数。
- `double`: 双精度浮点数。
- `*变量`: 指向变量的指针或数组。

4.5 C#程序编程说明

编程前将 `dll` 文件夹拷贝到本地工程目录下以便添加到工程中，动态库所有函数都封装在 `AC5152dml.cs` 中，只需添加现有项 `AC5152dml.cs`，软件会自动把所需文件都添加到当前项目中。在程序中直接调用的函数形式为 `AC5152dml.AC5152_OpenDevice()`，`AC5152dml.AC5152_DI()`.....等等。

C#编程基本流程如下。

1. 新建一个工程，然后在项目名称处添加现有项 `AC5152.cs`。



2. 程序中添加 `using dlm`。
3. 程序中声明全局变量 `IntPtr hDevice`。

注:

具体参照 AC5152AR 的 C#例程，`AC5152dml.cs` 已经包含了所有必要的函数的声明语句。

部分示例如下所示。

```
using System;
```

```
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;
using dlm;

namespace WWLAB
{
    public partial class AC5152 : Form
    {
        IntPtr hDevice;

        其他

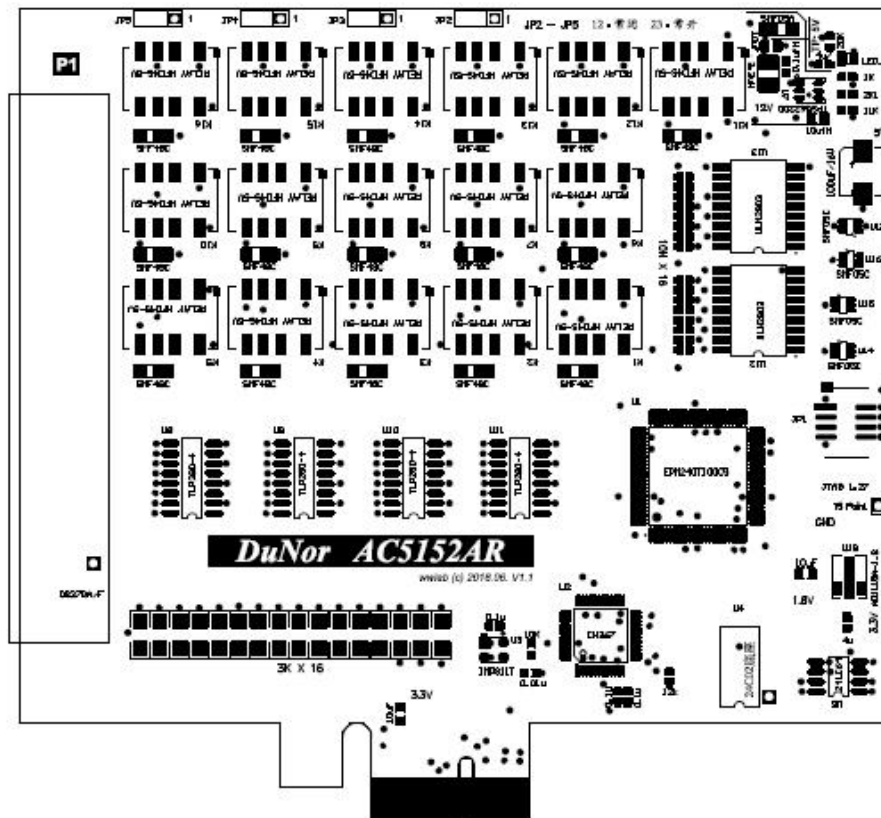
    }
}
```

五、附录

5.1 维修

1. 用户收到板卡后的一周内，如果板卡出现问题，可免费维修（不包括明显的烧毁和损坏等情况）。超过一周，视作维修，将收取用户一定的维修费用。
2. 用户维修前一定要联系厂家，不能直接寄回。
3. 用户寄回的板卡，只做维修，不能直接更换新的板卡。
4. 关于维修产生的邮费，需要用户和厂家各自负担一半。
5. 关于维修产生的器件费用，厂家会事先和用户联系。由厂家提供明确的器件清单和收费标准，经用户同意，方可维修。板卡维修好之后，需要用户支付维修费用，方可寄回。

5.2 板卡尺寸示意图



5.3 IO 译码地址表

板卡操作的基地址由主机自动分配。

PCI-E设备参数：

1. VID: 1C00H
2. DID: 5831H
3. SUBSYSID: 51520010H

偏移地址分配(IOBASE0)

偏移地址 A0-A3	读操作(RD)	写操作(WR)
0 H	读开关量低8位数据	写开关量低8位数据
1 H	读开关量高8位数据	写开关量高8位数据
10H	回读开关量输出低8位	
11H	回读开关量输出高8位	
22H	读EEPROM数据	设置数据寄存器
23H		设置EEPROM时钟
40H	读板卡型号52	
41H	读板卡型号51	
42H	读板卡版本号	
43H	读板卡日期月份	
44H	读板卡日期年份	

手册结束 双诺测控