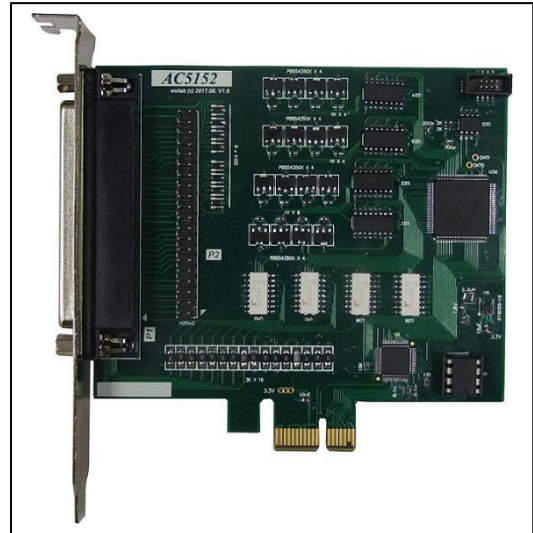

AC5152

使用手册

- PCI-E 总线隔离 I/O 卡
- 16 路隔离输入（支持共阴共阳模式）
- 16 路隔离 OC 输出
- 2 路 16 位计数器



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：

- ✧ AC5152控制卡一张
- ✧ 简介一份及光盘
- ✧ 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，超于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.....	1
一、AC5152 说明.....	3
1.1 AC5152 板卡简介.....	3
1.2 主要功能及性能.....	4
IO 部分.....	4
计数器部分.....	5
其他.....	5
二、安装与调试.....	6
2.1 驱动安装.....	6
2.2 连接器定义.....	8
2.3 配套端子板.....	9
2.4 典型接线图.....	10
输出.....	10
输入.....	10
计数器连接方式.....	12
三、软件开发.....	13
3.1 说明.....	13
3.2 接口函数说明.....	14
函数简介.....	14
约定.....	14
设备操作函数.....	14
系统函数.....	15
开关量操作函数.....	15
计数器操作函数.....	17
EERPOM 操作函数.....	18
3.3 VC 程序编程说明.....	18
3.4 LabView 程序编程说明.....	19
3.5 C#程序编程说明.....	20
四、附录.....	22
4.1 维修.....	22
4.2 板卡尺寸.....	22

一、AC5152 说明

1.1 AC5152 板卡简介

AC5152是一款低价格通用光电隔离I/O板，具有16路输入（支持共阳、共阴，默认为共阴输入）、16路输出和2路16位加法计数等功能。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。

AC5152的输入支持9-24伏输入，同时输出为集电极开路输出，输出驱动电流大于100毫安，可以方便地驱动小型继电器、LED等负载。

相关产品

PCI总线

AC6652: 隔离16DI/16DO, NPN输出。

AC6652B: 隔离8DI/8DO, NPN输出。

AC6652C: 隔离16DI/16DO, 4路计数器输入, NPN输出。

AC6652D: 隔离16DI (输入共阳) /16DO, PNP输出。

AC6652E: 隔离16DI/16DO, PNP输出。

AC6652R: 隔离16DI/16DO, 继电器输出 (1A / 30V AC/DC)。

AC6654: 隔离32DI/32DO, NPN输出。

USB总线

MP441: 隔离16DI/16DO, PNP输出。

MP452: 隔离16DI/16DO, NPN输出。

PCI-E总线

AC5152R: 隔离16DI/16DO, 继电器输出 (1A / 30V AC/DC)。

配套端子板

AC142A: DB37插座、螺丝端子连接板。

1.2 主要功能及性能

AC5152 原理框图主要由输入隔离模块、输出隔离模块、门阵列控制模块和 PCI-E 总线接口模块组成。其中，输入隔离模块中包含了 2 路隔离计数。

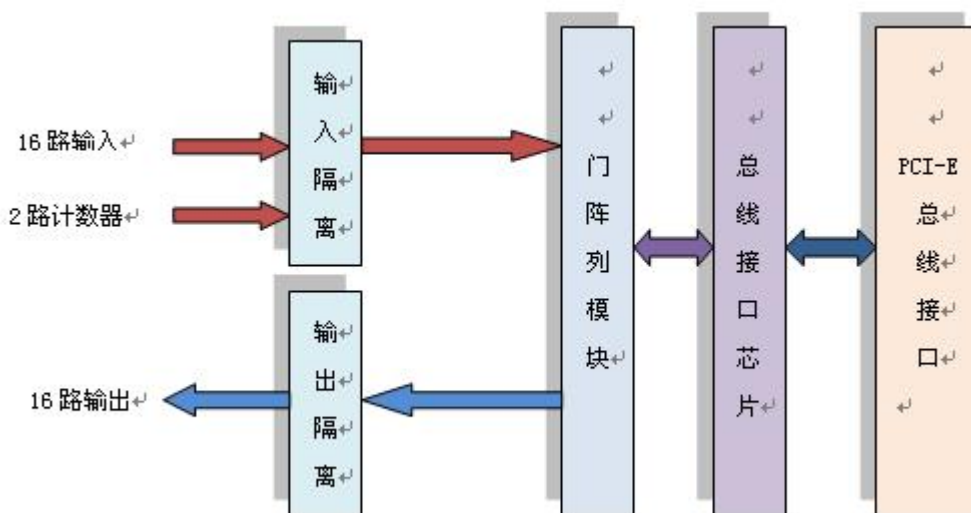


图 1.1 原理框图

IO 部分

■ 输入

1. 16路光耦隔离输入。输入为2组各8路独立隔离输入，每组有独立公共端。
2. 隔离电压大于1000V，绝缘电阻大于10G欧姆。
3. 输入电压范围：9~24V（电流约：3~8毫安）。
4. 小于1V，DI状态为低电平。
5. 9~24V，DI状态为高电平。
6. 悬空状态为断开。
7. 响应时间

电平变化	响应时间 (uS)
高->低	10
低->高	80

注：

1. 以上测试为典型数值。
2. 测试条件：输入电压为9伏。

■ 输出

1. 16路光耦隔离输出。分为2组，每8路为一组。两组输出地相互隔离。
2. 标准输出电流（下拉）：100mA/路，耐压：30V。
3. 输出压降：小于1V。
4. 最大输出频率：10KHz。
5. 晶体管输出模式：OC输出。
6. 最大输出频率：10KHz。

电平变化	响应时间（uS）
高->低	10
低->高	50

注：以上测试为典型数据值。

测试条件，输出上拉电阻1K欧姆，电压9V。

计数器部分

- 2路16位加法计数器。
- 与开关量输入DI0和DI1共用。
- 信号电压范围：9-24V。
- 触发模式：上升沿。
- 最大计数频率：5KHz。
- 计数器范围：0-65535。溢出后保持在溢出值，并且设置溢出标志位。

其他

- 总线：32位PCI-E总线。
- 软件支持：XP、win7、win8、win10。
- 供电：单3.3V / 12V。
- 板卡尺寸：10.2cm X 9.5cm（长X高）。

二、安装与调试

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 位系统驱动。
- ◇ driver x64: 64 位系统驱动。

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AC5152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AC5152插卡，请每一次只安装一个AC5152插卡。第一次安装的插卡的设备号为“0”，第二次安装的插卡的设备号为“1”，依此类推。
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”，寻找驱动存放的位置。最终能够看到 AC5152.inf 文件，点击选择，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，如图 2.1 所示。

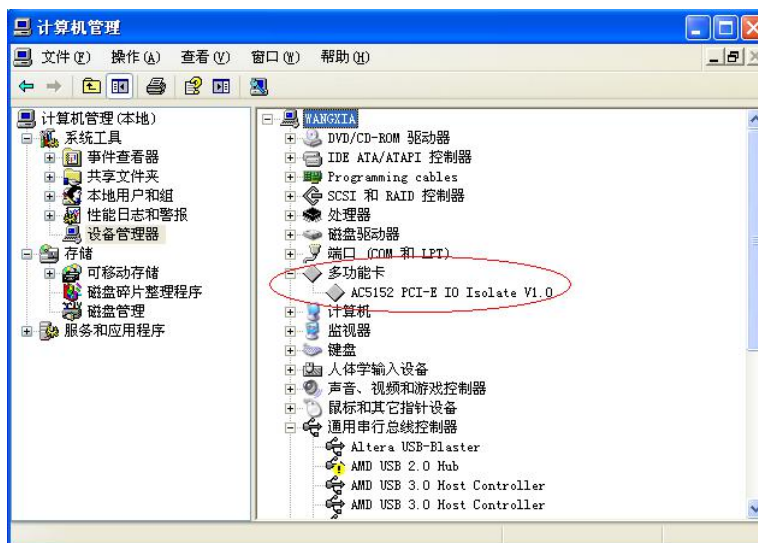


图 2.1 XP 系统下驱动完成效果图

Win7 系统

1. 关闭计算机，将AC5152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AC5152插卡，请每一次只安装一个AC5152插卡。第一次安装的插卡的设备号为“0”，第二次安装的插卡的设备号为“1”，依此类推。
2. 启动计算机。
3. win7 系统通常会自动为该设备寻找驱动程序，此时可以等待系统安装。通常情况下安装不成功。此时需要我们手动进行安装。
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”，寻找驱动存放的位置。最终能够看到 AC5152.inf 文件，点击选择，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。

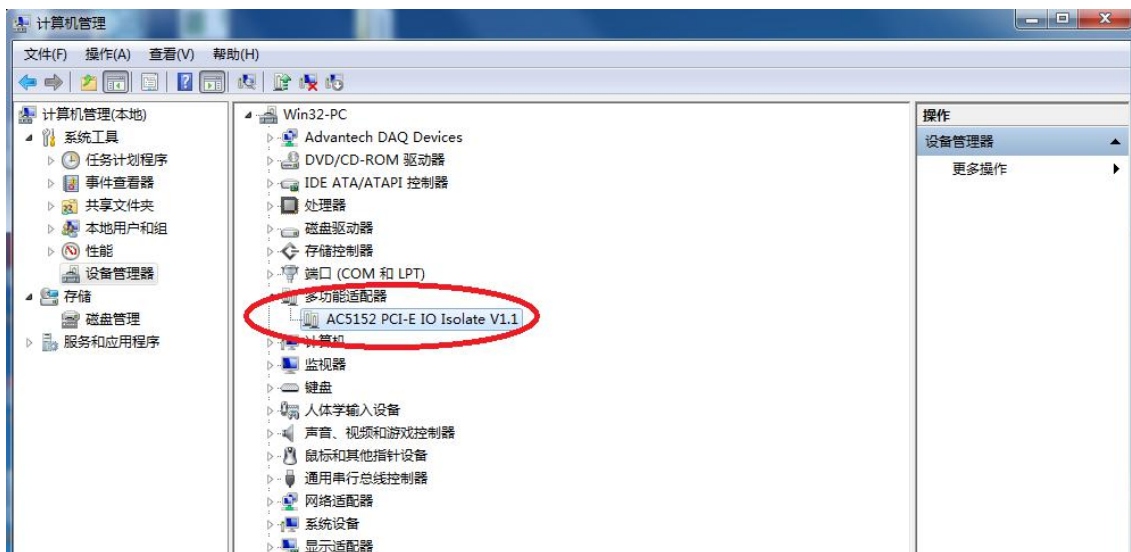
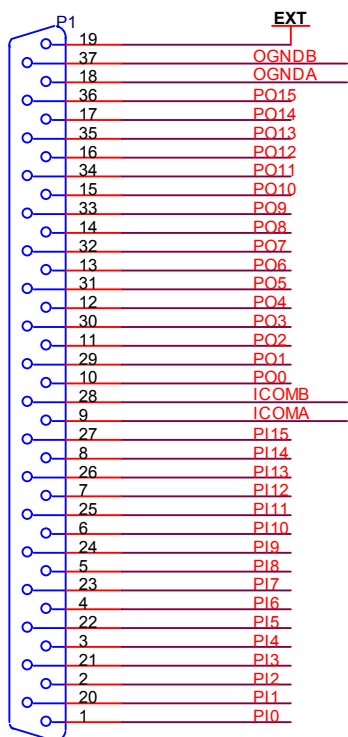


图 2.2 win7 系统下驱动完成效果图

关于驱动安装，更详细的请参看光盘中图文版〈板卡驱动安装.pdf〉。

2.2 连接器定义

■ 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 路输入公共端。输入若采用共阴模式，则该端为输入低 8 路地线；输入若采用共阳模式，则该端为外接供电电源（9-24V）。
4. 脚 28: 高 8 路输入公共端。输入若采用共阴模式，则该端为输入高 8 路地线；输入若采用共阳模式，则该端为外接供电电源（9-24V）。
5. 脚 18: 低 8 路输出地线。
6. 脚 37: 高 8 路输出地线。
7. 脚 19: 外部供电电源（9~24V），此电源的地线连接到输出地线（脚 18 或脚 37）。**您在使用开关量输出时，请使用该供电电源，以保证器件不被烧坏。**

■ P2: 40 芯双排针输出插座

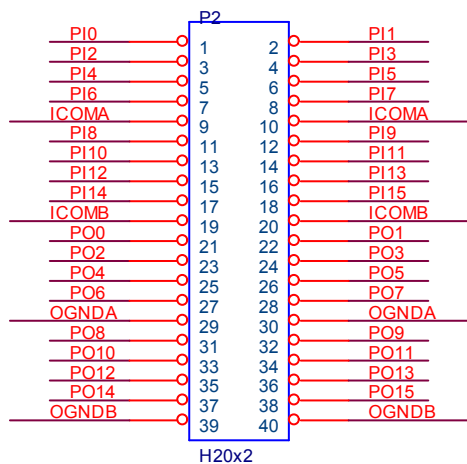


图 2.4 40 芯插座定义

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、10: 低 8 路输入公共端。
4. 脚 19、20: 高 8 路输入公共端。
5. 脚 29、30: 低 8 路输出地线。
6. 脚 39、40: 高 8 路输出地线。

复合 I/O 脚。

PI0、PI1: 同时作为计数器通道 0、1 输入。不用时为开关量输入的 0、1 通道。

2.3 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。

2.4 典型接线图

输出

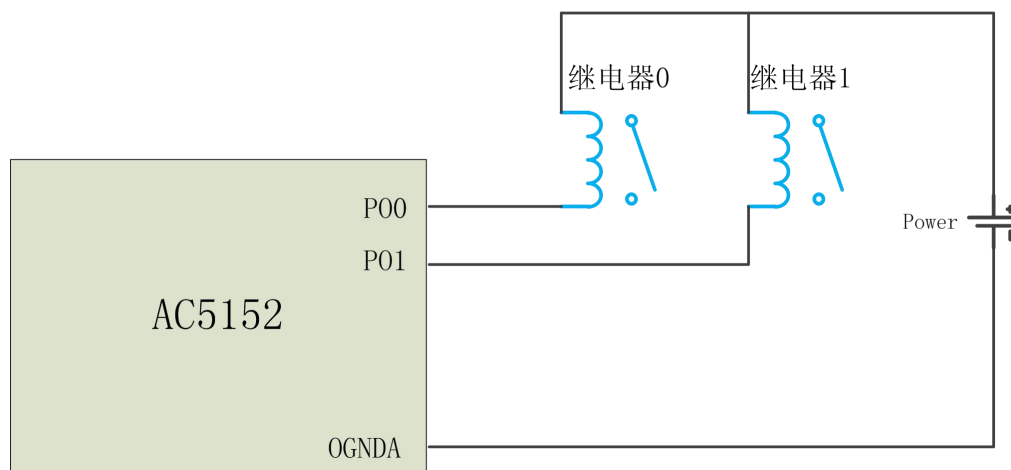


图 2.5 典型输出接线图

DB37 输出接口 驱动继电器图例中给出两路，其他输出通道的接法与此相同。

Power 的取值范围为 9~24V。

注：以上为输出驱动继电器的连接方式，输出驱动电磁阀，气缸等设备的接线方法也是如此。

输入

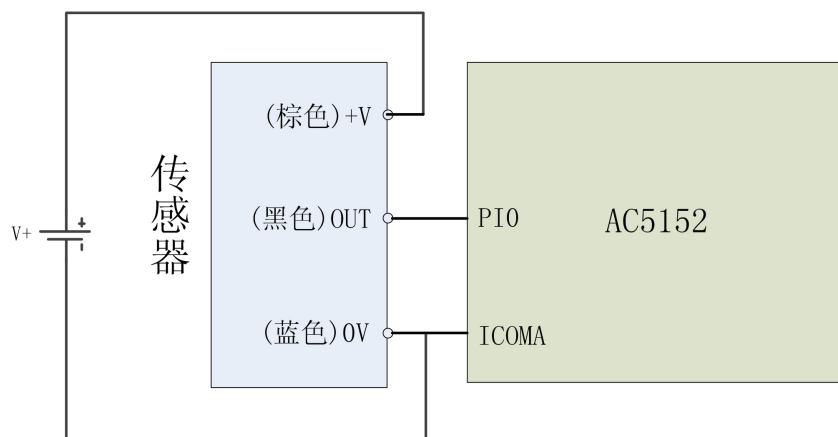


图 2.6 典型共阴输入接线图

图 2.6 为典型的**共阴**输入连接方式，以传感器为例。

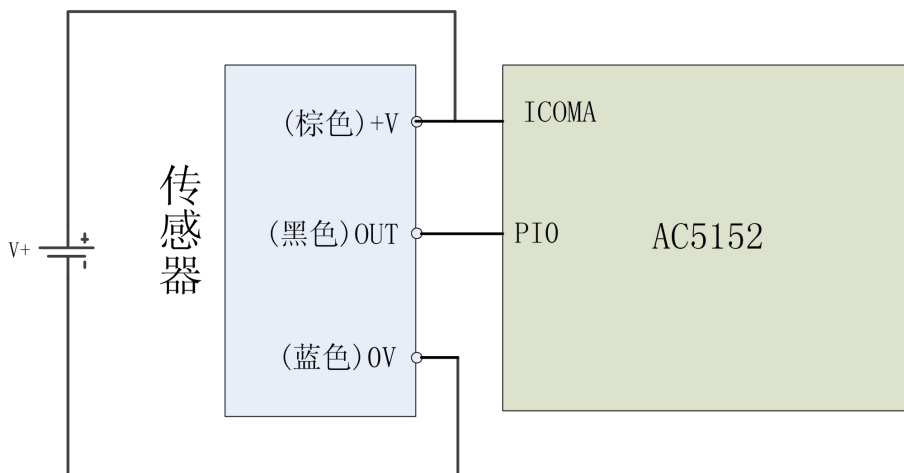


图 2.7 典型**共阳**输入接线图

图 2.7 为典型的**共阳**输入连接方式，以传感器为例。

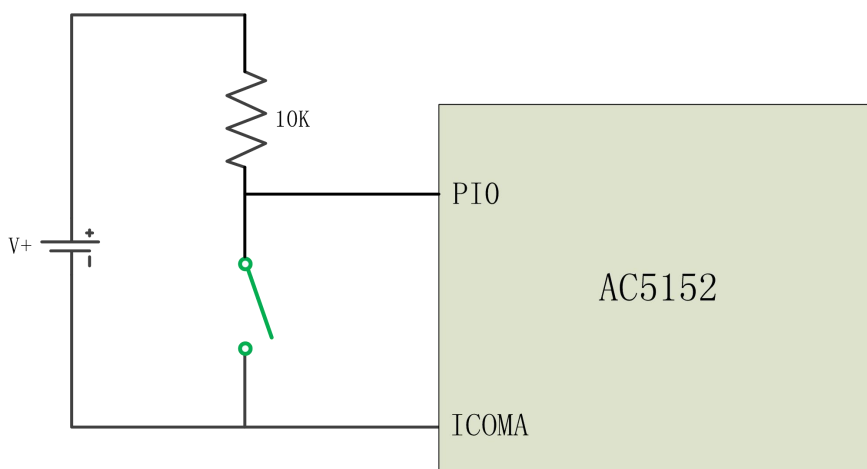


图 2.8 共阴输入接线图

图 2.8 为 AC5152 与开关连接的典型连接方式，采用共阴输入。

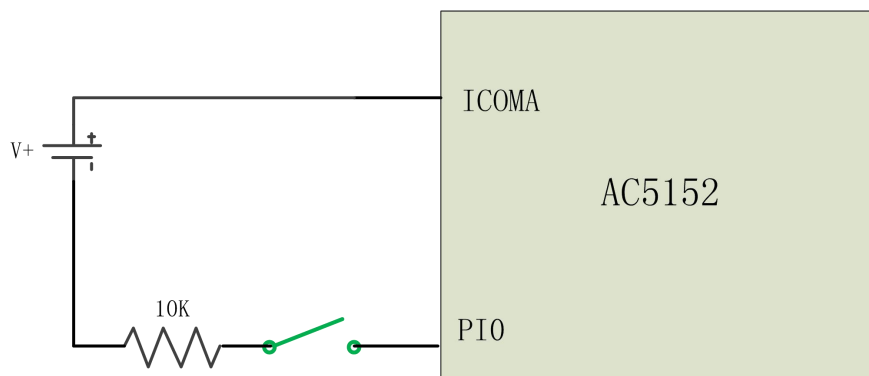


图 2.9 共阳输入接线图

AC5152 与开关的典型连接方式，共阳输入模式。

计数器连接方式

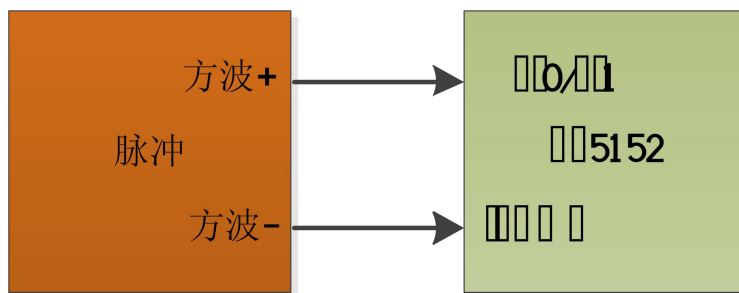


图 2.10 计数器连接方式

AC5152 计数器典型接线图。计数通道与输入通道的 PI0 和 PI1 复用。

方波每次上升沿到来时，计数器的计数值加一。计数值超过 65535，则保留 65535，并且置溢出标志为 1。

三、软件开发

3.1 说明

注意:板卡需要安装驱动后才能操作。

AC5152附带光盘中, 提供如下内容:

1. 说明书。
2. 驱动程序, 支持winXP/win7/win8/win10操作系统。
3. Visual C++、MFC、LabView、C#编程实例。
4. AC5152测试程序。

注: 由于win98、winNT微软已经不提供支持, 不建议使用。AC5152卡的驱动不支持win NT。

- 在光盘的\PCI-E\AC5152\DRIVER目录中包含: AC5152.inf、 AC5152.sys 、 AC5152.dll、 AC5152.lib、 AC5152.h等5个文件。
 - ◇ AC5152.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\AC5152\VC目录中包含:
 - ◇ VC的编程例子。
 - ◇ 编程需要的头文件。
- 在光盘的\PCI-E\AC5152\MFC目录中包含:
 - ◇ MFC的编程例子。
 - ◇ 编程需要的头文件。
- 在光盘的\PCI-E\AC5152\LabView目录中包含: LabView的编程例子。
- 在光盘的\PCI-E\AC5152\C#目录中包含:

- ◇ C#的编程例子。
- ◇ 编程所需要的dlm文件。

- Ac5152test.exe: 测试程序。

3.2 接口函数说明

函数简介

函数分为:

1. 设备初始化函数, 用于初始化串口, 并获得操作句柄。
2. IO 操作函数, 操作输入输出的状态。
3. 计数器函数, 操作计数器的模式, 读取计数器的值。
4. EEPROM 读写函数, 用户可以保存数据到板卡上的用户 EEPROM。

AC5152 通过不同的句柄来区分多个 AC5152 或其他设备, 应用 AC5152_OpenDevice 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

- 初始化串口, 获得操作句柄

函数: HANDLE AC5152_OpenDevice(__int32 Devicenum)

功能: 初始化串口, 获得 AC5152 板卡的操作句柄。

参数:

- ◇ Devicenum: 入口参数, AC5152 的设备号。=0, 1, 2...分别代表第一块, 第二块, 第三块... 板卡。

◇ 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。其他开发环境请参考例程。

□ 关闭设备，释放句柄

函数： __int32 AC5152_CloseDevice(HANDLE hDevice1)

功能： 释放以 hDevice 句柄打开的 AC5152 句柄。

参数：

◇ hDevice1：入口参数，卡的操作句柄。

◇ 函数返回值：0：成功 / -1：失败。

系统函数

□ 读入板卡信息

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

功能： 读入板卡的型号及版本号。

参数：

◇ hDevice：入口参数，卡的操作句柄。

◇ modle：char 指针，存放板卡信息。

◇ 函数返回：板卡信息。

开关量操作函数

AC5152 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 号口中的一个 8 位口输入数据。

注：AC5152 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 对应 PORT0，1 对应 PORT1。

◇ 函数返回：出口参数，返回读入的数据，低 8 位有效。对应输入端口的 7-0 号。

□ 开关量位输入

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: AC5152 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。

功能: 读取 DI 的位输入状态。

参数:

- ✧ `hDevice`: 入口参数, 卡的操作句柄。
- ✧ `ionum`: 入口参数。=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, 返回读入的数据。=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

函数: `AC5152_D0(HANDLE hDevice, __int32 ionum, __int32 iodata)`

功能: 设置 PORT0、PORT1 的数据。

注: AC5152 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`: 入口参数 0~1。0-PORT0; 1-PORT1。
- ✧ `dodata`: 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, 其它失败。

□ 开关量位输出

函数: `AC5152_D0B(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: AC5152 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。

□ 回读开关量输出状态

函数: `__int32 AC5152_RD0 (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: AC5152 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。
- ✧ 函数返回: 出口参数, 返回对应输入输出的数据 (低 8 位有效)。

计数器操作函数

AC5152 具有两路 16 位加法计数器。计数器有效数据范围为 0~65535。用户初始化后计数器数值为 0, 如果计数器数值超过 65535, 认为计数器溢出, 计数器数值保持在 65535 并置溢出标志为 1。溢出标志只有在重新初始化计数器后, 才会复位为 0。

□ 启动 AC5152 的计数器

函数: `__int32 AC5152_CReset (HANDLE hDevice, __int32 cntch)`

功能: 启动计数器。这个函数会复位计数器的值为 0, 并且启动计数器开始工作。

参数:

- ✧ hDevice: 入口参数, 卡的操作句柄。
- ✧ cntch: 计数器通道。范围 0-1, 代表计数器 0、1 通道。
- ✧ 函数返回: =0 操作成功; 其他失败。

□ 读出计数器数值及状态

函数: `__int32 AC5152_CRead (HANDLE hDevice, __int32 cntch)`

功能: 读出 16 位计数器数据、判断计数器是否溢出。

参数:

- ✧ hDevice: 入口参数, 卡的操作句柄。
- ✧ cntch: 计数器通道。范围 0-1, 代表计数器 0、1 通道。
- ✧ 函数返回: 出口参数, 小于 0 代表计数器溢出; >0, 表示 16 位计数器数据 (0~65535)。

注意:

1. 计数器通道 0、1 号的时钟输入与开关量输入的 0、1 号共用。
2. 计数器应用时, 应该先启动计数器, 然后查询计数器数值。
3. 用户测试计数器, 可以将开关量输出与输入连接, 通过发送高低电平来观察计数器的动作。
4. 计数器模式为加法计数模式。上升沿计数。

EEPROM 操作函数

用户可以利用 EEPROM 函数向 AC5152 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!

□ EEPROM 写入数据。

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: 出口参数: =0 操作成功/其他失败。

□ EEPROM 读取数据

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, 请客户参看样例。我们提供了 VC、MFC、LabView、C#等例程。

其他例程请客户搜索相关的资料或者向技术支持咨询。

双诺 研发部 010-62615449-212/210

3.3 VC 程序编程说明

编程前, 请将 AC5152.lib 及 AC5152.h 两个文件拷贝到用户当前目录中。

VC 编程的基本流程如下。

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。方法，程序的开始处加入如下语句：

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

详细可以参考 VC 目录中的程序，AC5152.h 文件包含了需要的函数的声明过程。

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 AC5152 的操作函数
#pragma comment(lib, "AC5152.lib")
#include "AC5152.h"

HANDLE hDevice=INVALID_HANDLE_VALUE; //硬件操作句柄

main()
{
    //获得 AC5152 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 AC5152 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。硬件句柄只要在程序启动时打开一次即可，不需要每次打开或关闭。

3.4 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 节点来调用我们所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 的函数转换为 labview 库的功能。用户在生成函数库时请注意相应的数值定义。

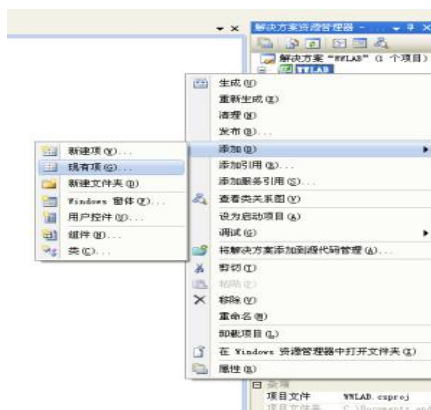
- `__int32`: 32 位有符号数。
- `HANDLE`: 操作句柄, 等效一个 32 位有符号数。
- `unsigned char`: 无符号 8 位整数。
- `double`: 双精度浮点数。
- `*变量`: 指向变量的指针或数组。

3.5 C#程序编程说明

编程前, 请将 `d1m` 文件夹拷贝到本地工程目录下以便添加到工程中, 动态链接库中的所有函数都封装在 `AC5152d1m.cs` 中, 用户只需要添加现有项 `AC5152d1m.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 `AC5152d1m.AC5152_OpenDevice()`、`AC5152d1m.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 d1m`。
3. 程序中声明全局变量 `IntPtr hDevice`。

注:

具体参照 `AC5152/C#`例程, `AC5152d1m.cs` 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;

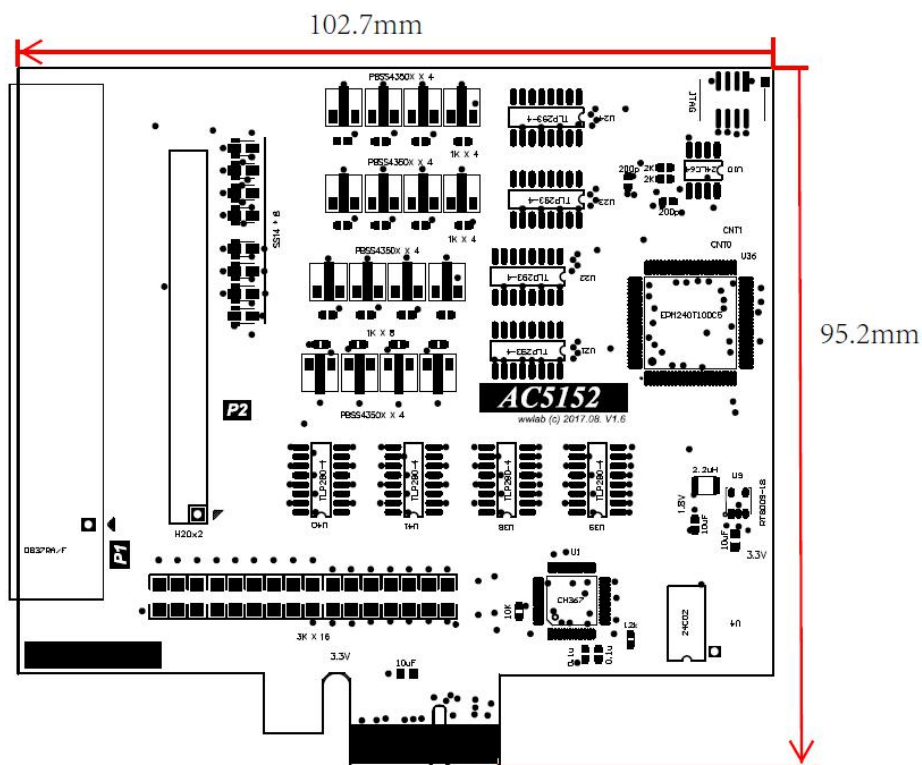
namespace WWLAB
{
    Public partial class AC5152:Form
    {
        IntPtr hDevice;
        其他
    }
}
```

四、附录

4.1 维修

1. 客户购买板卡后一周内，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，将收取客户一定的维修费用。
2. 客户维修前一定要联系厂家，不能直接寄回。
3. 客户寄回的板卡，只做维修，不能直接更换新的板卡。
4. 关于维修产生的邮费，需要客户和厂家各自负担一半。
5. 关于维修产生的器件费用，厂家会事先和客户联系。由厂家提供明确的器件清单和收费标准，经客户同意，方可维修。板卡维修好之后，需要客户支付维修费用，方可寄回。

4.2 板卡尺寸



手册结束 双诺测控