

PCI-1234

**光隔离 64 通道输入输出卡
使用说明书**

V1.11

2010-1

概述	3
管脚定义.....	4
开关选择.....	6
寄存器	7
输入输出说明.....	9
编程指南.....	10
程序demo.....	12

概述

PCI-1234 是一款基于 PCI 协议的光隔离 64 通道输入输出控制卡，用户可设置板 ID 号，兼容 5V/3.3V PCI 协议。

特点：

- 32 路光隔离输入通道（无极性），32 路光隔离输出
- 具有较高的输出驱动能力
- 可配置板 ID 号
- 兼容标准产品
- 兼容 5V 和 3.3V 2.2 版本 PCI 协议
- 光隔离输入通道输入电阻 $4.7K\Omega$ @1/4W
- 光隔离输入逻辑高：3V~24V，逻辑低为：0~2.4V
- 隔离输出通道采用达林顿管集电极开路输出，集电极开路电压 5V~35V(DC)
200mA 反向电流输出
- 隔离电压 3750Vrms

管脚定义

PCI1234 100-pin SCSI-II 接口如下

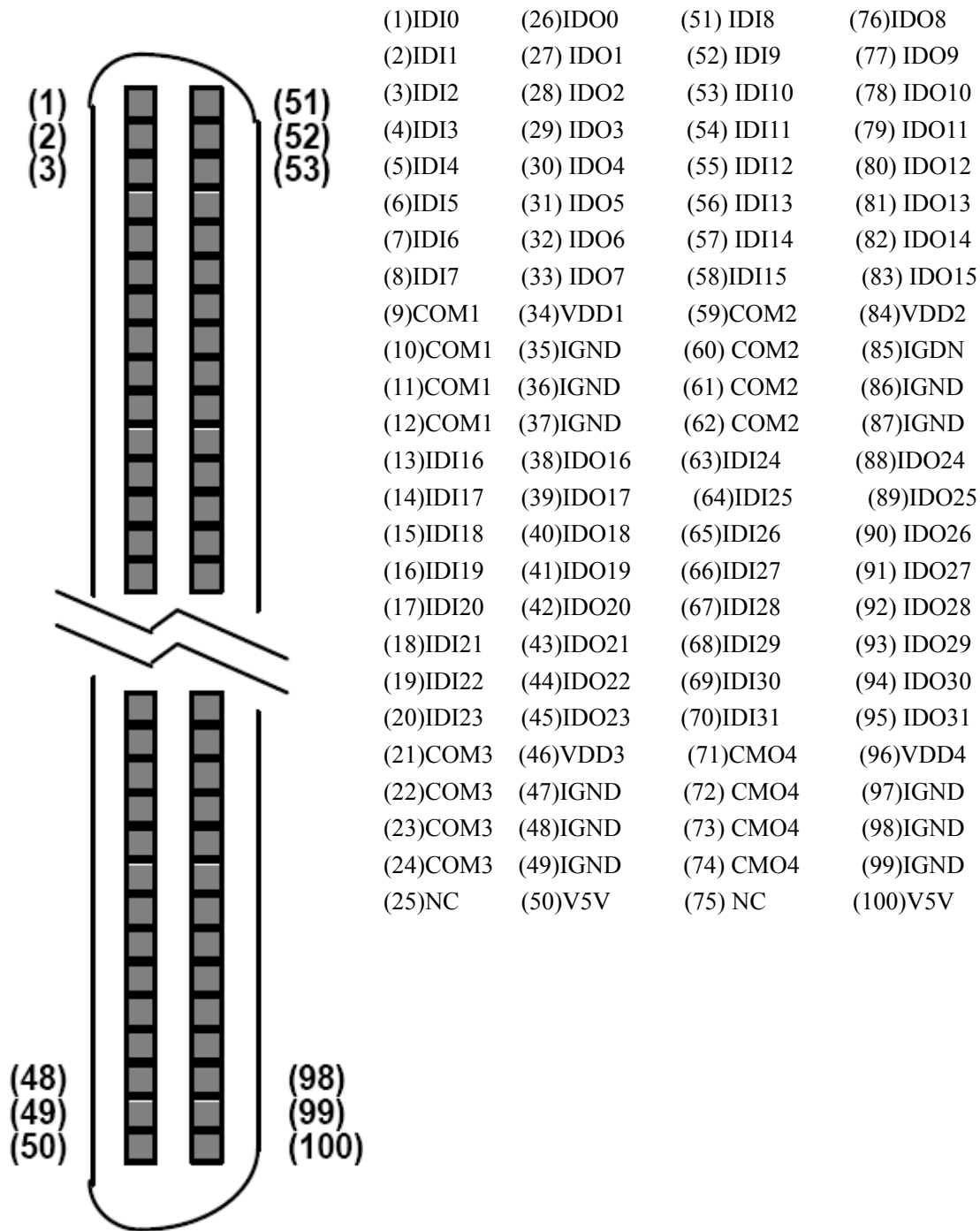


图 1 PCI1234 CN1 链接图

其中：

IDIn: 光隔离输入通道 n(n= 0~31)

IDOn: 光隔离输出通道 n(n= 0~31)

VDDm: 保护电源端

IGND: 光隔离输出通道地

COMm: 光隔离输入公共端。（COM1 是 IDI0~IDI7 的公共端，COM2 是 IDI8~IDI15 的公共端，COM3 是 IDI16~IDI23 的公共端，COM4 是 IDI24~IDI31 的公共端。）

V5V: 板载隔离 5V 电源输出。最大输出电流 600mA

NC: 空脚，没有连接

开关选择

SW2 ：板 ID 选择

开关选择 （开关箭头所指向的数字）	板ID
0	15
1	14
2	13
3	12
4	11
5	10
6	9
7	8
8	7
9	6
A	5
B	4
C	3
D	2
E	1
F	0

寄存器

PCI1234 寄存器为 32 位宽，用户可以通过 32 位 IO 指令来访问 PCI1234 寄存器。每一个寄存器的地址实际上是偏移地址加上控制卡的基地址。PCI1234 寄存器地址列表如下：

I/O 地址寄存器：

	地址	写操作	读操作
PCI-1234	Base+0x00	光隔离输出通道	光隔离输入通道

读操作对应 PCI1234 光隔离输入，写操作对应光隔离输出。PCI1234 有 32 路光隔离输入通道和 32 路光隔离输出通道，I/O 地址寄存器的每一位对应一个光隔离通道

读 I/O 地址寄存器

位数	7	6	5	4	3	2	1	0
对应的通道	IDI7	IDI6	IDI5	IDI4	IDI3	IDI2	IDI1	IDI0
位数	15	14	13	12	11	10	9	8
对应的通道	IDI15	IDI14	IDI13	IDI12	IDI11	IDI10	IDI9	IDI8
位数	23	22	21	20	19	18	17	16
对应的通道	IDI23	IDI22	IDI21	IDI20	IDI19	IDI18	IDI17	IDI16
位数	31	30	29	28	27	26	25	24
对应的通道	IDI31	IDI30	IDI29	IDI28	IDI27	IDI26	IDI25	IDI24

写 I/O 地址寄存器

位数	7	6	5	4	3	2	1	0
对应的通道	IDO7	IDO6	IDO5	IDO4	IDO3	IDO2	IDO1	IDO0
位数	15	14	13	12	11	10	9	8
对应的通道	IDO15	IDO14	IDO13	IDO12	IDO11	IDO10	IDO9	IDO8
位数	23	22	21	20	19	18	17	16
对应的通道	IDO23	IDO22	IDO21	IDO20	IDO19	IDO18	IDO17	IDO16
位数	31	30	29	28	27	26	25	24
对应的通道	IDO31	IDO30	IDO29	IDO28	IDO27	IDO26	IDO25	IDO24

板 ID 寄存器:

	地址	写操作	读操作
PCI-1234	Base+0x04	-----	板 ID

输入输出说明

光隔离输出采用的是达林顿管集电极开路输出形式，具有较高的驱动能力。光隔离输入无极性，逻辑高为3V~24V，逻辑低为：0~2.4V。输入电阻为4.7K Ω 。PCI1234光隔离输入与外部连接如下：

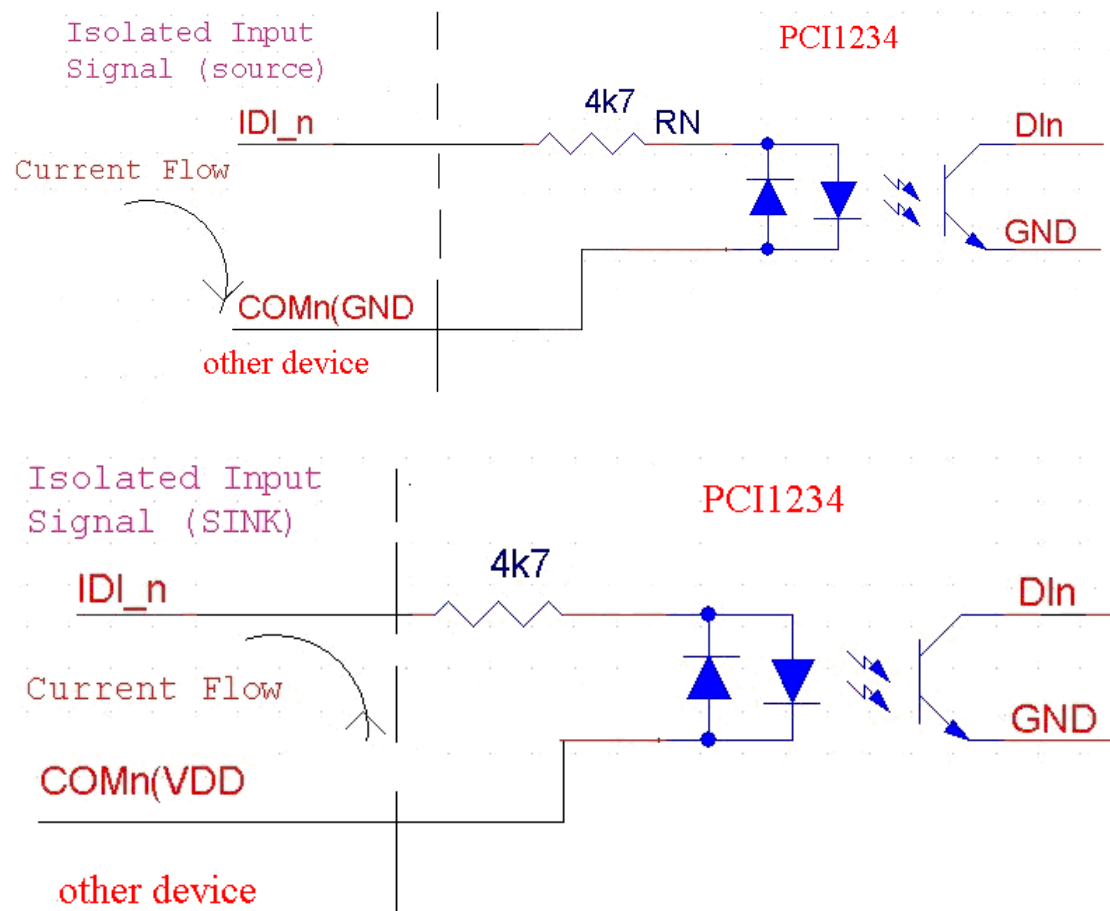
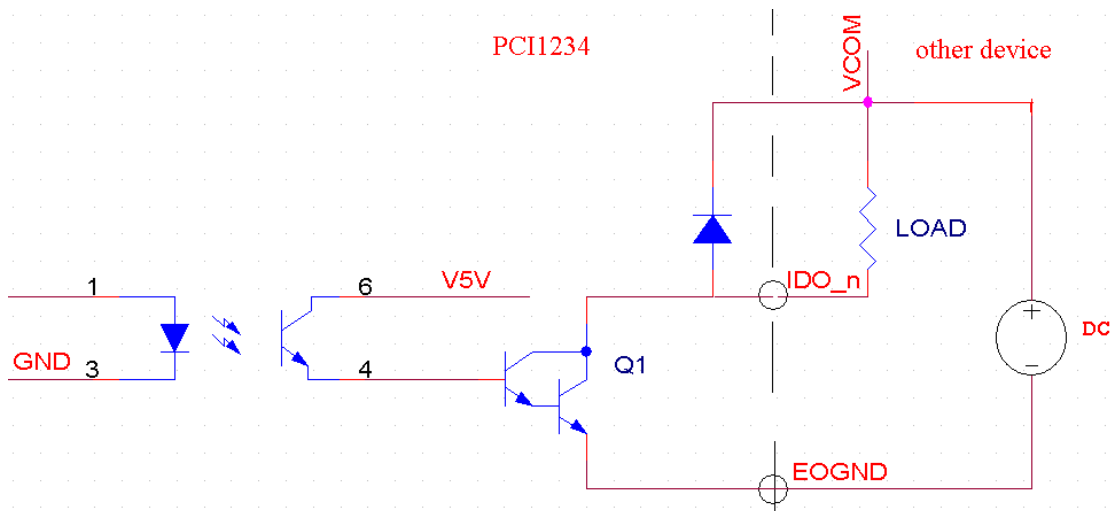


图2 光隔离输入通道连接图

光隔离输出通道与外部共地连接如下图所示：当光隔离输出通道为“ON”的时候，有电流从三极管流过，当光隔离输出通道为“OFF”的时候，没有电流从三极管流过，注意，当负载为感性负载时(如继电器，电机等等)，需要在VDD端外接一个电源，用来形成一个释放回路，保护三极管。当光隔离输出通道由“ON”变为“OFF”的时候，由于感性负载会产生一个很大瞬间电流。如果这个瞬间大电流没有回路释放，则很有可能烧毁三极管。



编程指南

数据类型 U32 : unsigned long

U32 Pci1234open (U32 boardid);

作用: 打开PCI1234设备, 获得资源

参数 boardid : PCI-1234 板ID. 必须在0~15之间

说明: 在使用设备前, 先确认该函数调用成功, 才能确保后面的函数调用成功

返回值: PCI1234DevNotFound
PCI1234Success
PCI1234InvalidBoardId

U32 Pci1234Read(U32 boardid, U32 *Data)

作用: 读PCI1234光隔离输入通道

参数 boardid : PCI-1234 板ID. 必须在0~15之间
Data 指向U32类型的指针, 用来存储读回来的数据

说明: 在使用这个函数前, 请确认Pci1234open函数调用成功

返回值: PCI1234InvalidBoardId.
PCI1234Success
PCI1234ApiFailed

U32 Pci1234Write(U32 boardid, U32 writedata);

作用: PCI1234光隔离输出

参数 boardid : PCI-1234 板ID. 必须在0~15之间
writedata: 需要输出的数据

说明: 在使用这个函数前, 请确认Pci1234open函数调用成功

返回值: PCI1234InvalidBoardId.
PCI1234Success
PCI1234ApiFailed

U32 Pci1234Close(U32 boardid);

作用： 关闭PCI1234设备，释放资源

参数 boardid : PCI-1234 板ID. 必须在0~15之间

说明： 最后不需要使用设备时，调用该函数，释放资源

返回值 : PCI1234InvalidBoardId.

PCI1234Success

PCI1234ApiFailed

程序demo

测试输入和输出，输入和输出直接相连。由于光藕响应速度慢，所以在输出之后添加了延时。

```
#include <time.h>
#include <stdio.h>
#include "PciDrvApi.h"
#include "PCI1234Error.h"
void Delay(void);
int main(void)
{
    U32 dwDataWriteToPCI;
    U32 dwDataReadFromPci;
    U32 dwTemp = 0;
    U32 dwSelect = 0;
    U32 dwERROR1 = 0;
    U32 dwsum = 0;
    U32 dwID = 15;
    printf("\n\n\t\t PCI1234 Test\n\t\t January 2007\n\n" );
    printf("\nPCI1234Test V0.1 \n\n" );
    printf("\t input the Board id to select: ");
    scanf(" %d", &dwID);
    dwERROR1=Pci1234Open(dwID);//获得资源;
    if(dwERROR1!=PCI1234Success)
    {
        printf("\t ERROR, can not find NO.%d card\n",dwID);
        return 0;
    }
    do{
        printf("\t input hex value to output to NO.%d board:",dwID);
        scanf(" %X", &dwDataWriteToPCI);
        dwERROR1=Pci1234Write(dwID,dwDataWriteToPCI);// 光隔离输出
        if(dwERROR1!=PCI1234Success)
        {
            printf("\t ERROR\n");
            return 0;
        }
        Delay();// 添加延时
        dwERROR1=Pci1234Read(dwID,&dwDataReadFromPci);
        printf("\t Data Read From the NO.%d board  %4x\n",dwID,dwDataReadFromPci);
        printf("\n");
        printf("\t input 0 to quite or 1 to continue :");
        scanf("%d",&dwSelect);
    }
```

```

    }while(dwSelect);
    dwERROR1 = Pci1234Close(dwID); //释 放资 源;
    if (dwERROR1 == PCI1234Success)
    {
        printf("\t close sucesed \n");
    }
    else
    {
        printf("\t   close fault \n");
    }
    printf("\n\n");
    return 0;
}

```

```

void Delay(void)//delay
{
    U32 n = 0x0ffff;
    do{
        n--;
    }while(n!=1);
}

```

修订记录

修改时间	说明
2007-7	创建
2009-9	修改输入示意图