

PC-6405 光隔离开关量输入卡技术说明书

1. 概述:

PC-6405 开关量输入接口卡适用于具有 ISA 总线的 PC 系列微机, 具有很好的兼容性, CPU 从目前广泛使用的 64 位处理器直到早期的 16 位处理器均可适用, 操作系统可选用经典的 MS-DOS, 目前流行的 Windows 系列, 高稳定性的 Unix 等多种操作系统以及专业数据采集分析系统 LabVIEW 等软件环境。在硬件的安装上也非常简单, 使用时只需将接口卡插入机内任何一个 ISA 总线插槽中, 信号电缆从机箱外部直接接入。本卡也可插入我所研制的 PC 扩展箱内使用。

本卡适用于工业现场中各种开关信号的自动控制以及计算机同数字仪器的接口。考虑到在开关量的转换中“开/关”瞬态对计算机干扰十分强烈及现场强电的干扰, 本卡采用了光电隔离技术, 使计算机与现场信号之间全部隔离, 提高了计算机与本卡在工作中的抗干扰能力和抗损毁能力。

本卡上的开关量输入通道共 32 路, 分为 4 组, 每组 8 路。32 路输入信号均可通过 CPU 巡检方式工作, 且 32 路开入信号的低端共地。

2. 主要技术指标:

- 2.1 输入路数及电气连接方式: 32 路共地 (共阴) 方式。
- 2.2 输入信号电平范围: TTL~48V。
- 2.3 输入信号电流消耗: $\geq 5\text{mA}$ / 每路
- 2.4 每组信号之间、各组信号与接口卡之间隔离电平: $\geq 500\text{V}$ 。
- 2.5 电源功耗: $+5\text{V}(\pm 10\%) \leq 400\text{mA}$
- 2.6 环境要求: 工作温度: $10^\circ\text{C} \sim 40^\circ\text{C}$
相对湿度: $40\% \sim 80\%$
存贮温度: $-55^\circ\text{C} \sim +85^\circ\text{C}$
- 2.7 外型尺寸 (不含档板):
长 \times 高 = $152\text{mm} \times 106\text{mm}$ (6.0 英寸 \times 4.2 英寸)

3. 工作原理及操作说明:

3.1 开关量输入部分工作原理:

本卡上的所有 32 路开关量输入信号状态均可以由 CPU 通过巡检方式读出, 即 CPU 通过直接读取 I/O 口上的数据来判断输入信号的状态, 其工作原理如图 1 所示。

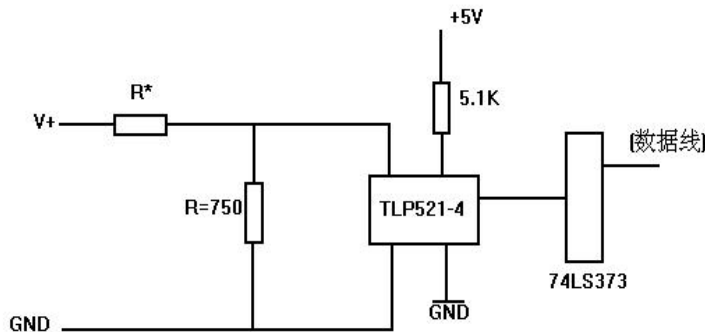


图 1 开关量输入部分工作原理

由图 1 所示, 当一个足够大 (TTL~48V) 的外部电压信号经过本卡上的电阻 R^* 与 R 分压后驱动光电耦合器的发光二极管发光, 使光电三极管导通, 输出一个低电平信号, 这个信号经过整形后即可通过三态门读入计算机数据总线。反之, 当外部电压信号为零或足够小时, 计算机读入的即为高电平信号。电阻 R^* 是一个限流保护电阻, 在卡上标为 $R_3 \sim R_{34}$, 第一组开入信号对应 $R_3 \sim R_{10}$, 第二组开入信号对应 $R_{11} \sim R_{18}$, 第三组开入信号对应 $R_{19} \sim R_{26}$, 第四组开入信号对应 $R_{27} \sim R_{34}$ 。插在焊接排上, 用户可以根据现场信号电压幅度而自行更换 (出厂时为 470Ω)。

R^* 的选用原则为: $R^* = (U_{IN} - U_R) / I$ (K Ω)

其中 U_{IN} 为现场信号高电平电压值, U_R 是加在光电耦合器上的电压值。一般 U_R 取值为 1V 左右, I 是流过发光二极管的电流, 一般取 5~20mA 左右。

根据上面的选用原则和使用经验, 我们推荐的输入信号和 R^* 的选择值见表 1:

表 1 R^* 的选择值

输入信号高电平	R^* 选择值
---------	-----------

3V~6V	470 Ω
6V~12V	2.4K Ω
12V~24V	4.7K Ω
24V~48V	10K Ω

上表中各档的阈值电压比较接近各档的下限值，阈值电压以下的电平将被认为是低电平，所以具有较高的抗噪声干扰的能力。

4. 安装及使用注意:

本卡的安装十分简便，只要将主机机壳打开，在关电情况下，将本卡插入主机的任何一个空余扩展槽中，再将档板固定螺丝压紧即可。37芯D型插头可从主机后面引出并与外设连接。

禁止带电插拔本接口卡。设置接口卡开关和安装接口带缆均应在关电状态下进行。

为保证人身及设备安全，应确保系统地线（计算机及外接设备接地点）接地良好。为防止外部设备中较大的电磁干扰，应注意对信号线进行屏蔽处理。

如果本卡连接的外部设备上加有较高的电压时，在安装或用手触摸本卡时，应先将外部设备的电源关闭并严禁触摸本卡。

5. 使用与操作:

5.1 I/O 基地址选择:

I/O 基地址的选择是通过开关 K₁ 进行的，开关拨至“ON”处为 0，反之为 1。拨码第 8 位表示地址线 A₉，拨码第 1 位表示地址线 A₂。初始地址的选择范围一般为 0100H~036FH 之间。用户应根据主机硬件手册给出的可用范围以及是否插入其它功能卡来决定本卡的 I/O 基地址。出厂时本卡的基地址设为 0100H，并从基地址开始占用连续 4 个地址。现举例说明见图 2。

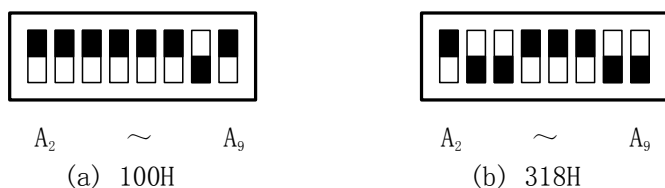


图 2 I/O 基地址选择

5.2 输入输出插座接口定义:

输入输出插座接口定义见表 2。

表 2 输出插座引线定义表

插座引脚号	信号定义	插座引脚号	信号定义
1	第一组 CH ₀	20	第一组 CH ₁
2	第一组 CH ₂	21	第一组 CH ₃
3	第一组 CH ₄	22	第一组 CH ₅
4	第一组 CH ₆	23	第一组 CH ₇
5	第二组 CH ₀	24	第二组 CH ₁
6	第二组 CH ₂	25	第二组 CH ₃
7	第二组 CH ₄	26	第二组 CH ₅
8	第二组 CH ₆	27	第二组 CH ₇
9	开入公共地	28	开入公共地
10	第三组 CH ₀	29	第三组 CH ₁
11	第三组 CH ₂	30	第三组 CH ₃
12	第三组 CH ₄	31	第三组 CH ₅
13	第三组 CH ₆	32	第三组 CH ₇
14	第四组 CH ₀	33	第四组 CH ₁
15	第四组 CH ₂	34	第四组 CH ₃
16	第四组 CH ₄	35	第四组 CH ₅
17	第四组 CH ₆	36	第四组 CH ₇
18	开入公共地	37	开入公共地

19	开入公共地		
----	-------	--	--

5.3 控制端口地址与有关数据格式:

5.3.1 各个控制端的操作地址与功能见表 3。

表 3 端口地址与功能表

端口操作地址	操作命令	功 能
基地址+0	读	读第一组 8 路开关量输入信号
基地址+1	读	读第二组 8 路开关量输入信号
基地址+2	读	读第三组 8 路开关量输入信号
基地址+3	读	读第四组 8 路开关量输入信号

5.3.2 开关量输入信号的数据格式:

开关量输入信号的数据格式采用的是位方式,即一个字节中的任意一位对应一路输入信号。其数据格式见表 4。

表 4 开关量输出信号数据格式

端口地址	组 别	D7	D6	D5	D4	D3	D2	D1	D0
基地址+0	第一组	CH7	CH6	CH5	CH4	CH3	CH2	CH1	CH0
基地址+1	第二组	CH7	CH6	CH5	CH4	CH3	CH2	CH1	CH0
基地址+2	第三组	CH7	CH6	CH5	CH4	CH3	CH2	CH1	CH0
基地址+3	第四组	CH7	CH6	CH5	CH4	CH3	CH2	CH1	CH0

6. 驱动程序简介:

PC-6000 系列演示程序及驱动程序是为 PC-6000 系列多功能工控采集板配制的工作在中西文 Windows 95/ 98/ NT 环境下的一组驱动程序以及使用该驱动程序组建的一个演示程序,可以方便地使用用户在中西文 Windows 环境下检测硬件的工作状态以及帮助软件开发人员在常用的 C\C++, Visual Basic, Delphi, Borland C++ Builder, Borland Pascal for windows 等开发环境中使用 PC-6000 系列工控采集板进行数据采集和过程控制等工作. 驱动程序是一个标准动态链接库 (DLL 文件)。它的输出函数可以被其它应用程序在运行时直接调用。用户的应用程序可以用任何一种可以使用 DLL 链接库的编程工具来编写。 每种板卡依据其自身功能的不同具有不同的输出函数和参数定义。

驱动程序输出函数定义:

所列函数的说明格式为 VC++6.0 环境下 PC6000.D11 库函数的原函数格式,无论使用哪一种开发工具,务必请注意数据格式的匹配及函数的返回类型,本说明中所使用的数据类型定义如下:

short ~ 16 位带符号数

unsigned long - 32 位无符号数

unsigned char - 8 位无符号数

* 函数: unsigned char APIENTRY DI6405Bit(short nAdd, short nBit)

功能: 采集某一位数字量输入信号的状态。

参数: nAdd 基地址
nBit 通道号: 0-31

* 函数: unsigned long APIENTRY DI6405All(short nAdd)

功能: 采集全部通道(32路)数字量输入信号的状态。

参数: nAdd 基地址
返回: 返回值为 32 个输入信号的状态。

如有需要使用 Windows 系列及 LabVIEW 驱动程序的用户可向本公司索取,请注明所使用的操作系统和开发软件。

7. 编程举例:

7.1 设本卡基地址为 0100H, 查询和显示各端口状态 (BASIC 语言):

```
10 ADD=&H100
20 D1=INP(ADD+0)      ; 读第一组 8 路状态
30 D2=INP(ADD+1)      ; 读第二组 8 路状态
40 D3=INP(ADD+2)      ; 读第三组 8 路状态
50 D4=INP(ADD+3)      ; 读第四组 8 路状态
60 PRINT D1           ; 显示第一组 8 路状态
70 PRINT D2           ; 显示第二组 8 路状态
80 PRINT D3           ; 显示第三组 8 路状态
90 PRINT D4           ; 显示第四组 8 路状态
100 END
```

7.2 C 语言程序举例:

```
/*
This demonstration program inllustrats how to carry out digital IO input functions use PC-6405. Program
receive 32 bits digital input from PC-6405 inport and display them on the screen.
*/
```

```
#include<stdio.h>
#include<conio.h>
#include<dos.h>

#define BaseAdd 0x300          /* Base address : 300 Hex */

main()
{
    unsigned char group1, group2, group3, group4;
    int i, g1[8], g2[8], g3[8], g4[8];

    clrscr();
    gotoxy(20, 2);
    printf(" PC-6405 test (Base Address 300 Hex)");
    gotoxy(5, 16);
    printf(" Press Esc to exit.");
    while(1)
    {
        group1 = inportb(BaseAdd+0);    /* Get input from Inport 1 */
        group2 = inportb(BaseAdd+1);    /* Get input from Inport 2 */
        group3 = inportb(BaseAdd+2);    /* Get input from Inport 3 */
        group4 = inportb(BaseAdd+3);    /* Get input from Inport 4 */
        for(i=0; i<8; i++)
        {
            g1[i]=(group1&1)?1:0;
            g2[i]=(group2&1)?1:0;
            g3[i]=(group3&1)?1:0;
            g4[i]=(group4&1)?1:0;
            group1>>=1;
            group2>>=1;
            group3>>=1;
            group4>>=1;
        }
        gotoxy(10, 5);
    }
}
```

```

printf("\nInport 1: bit7 - bit0 ==>");
for(i=7; i>=0; i--)
    printf(" %d", g1[i]);
printf("\nInport 2: bit7 - bit0 ==>");
for(i=7; i>=0; i--)
    printf(" %d", g2[i]);
printf("\nInport 3: bit7 - bit0 ==>");
for(i=7; i>=0; i--)
    printf(" %d", g3[i]);
printf("\nInport 4: bit7 - bit0 ==>");
for(i=7; i>=0; i--)
    printf(" %d", g4[i]);

if(bioskey(1))                /* Press Esc to exit */
    if((bioskey(0)&0xff)==0x1b)
        break;
    }
}

```

7.3 驱动程序使用举例：

在 Windows 95 / 98 环境下，使用 MicroSoft Visual Basic 6.0 开发环境，采用调用驱动程序的输出函数的方法对 PC6405 的 32 个通道进行操作。

注意：在 VB6 中，数据类型 Integer 为 16 位带符号整数，Long 为 32 位带符号整数，Byte 为 8 位无符号数。

首先创建一个窗口，名为 Form。设置一个定时器，名为 Timer1。

```
Private Declare Function DI6405Bit Lib "pc6000.dll" (ByVal nAdd As Integer, ByVal nBit As Integer) As Byte
```

```
Private Declare Function DI6405All Lib "pc6000.dll" (ByVal nAdd As Integer) As Long
```

```
Private Sub Timer1_Timer()
    ' 多通道输入
    Text1 = DI6405All(256)
    ' 单通道输入
    For i = 0 To 1
        Text2(i) = DI6405Bit(256, i)
    Next i
End Sub

```

8. 产品清单及保修：

产品清单：

1. 6405 光隔离开关量输入接口卡壹块。
2. 37 D 型接头壹套。

本产品自售出之日起一年内，凡用户遵守贮存、运输及使用要求，而产品质量低于技术指标的，凭保修单免费维修。因违反操作规定和要求而造成损坏的，需交纳器件和维修费。