

AMPCI-9101A 通用数据采集控制板

一、概述

AMPCI-9101 板是 PCI 总线通用数据采集控制板,该板可直接插入具备 PCI 插槽的工控机或
个人微机,构成模拟量电压信号、数字量电压信号采集、监视输入和模拟量电压信号输出、数字量电
压信号输出系统。

AMPCI-9101 板为用户提供了单端 16 路模拟量数据采集输入通道,1 路 12Bit 模拟量电压信号
输出,8Bit TTL 数字量输入和 8Bit TTL 数字量输出,对 AMPCI-9101 板的所有读写操作均为 16Bit
即 D00~D15。

二、性能和技术指标

2.1 性能

- 模拟信号输入 A/D 分辨率 12Bit
- 16 路单端输入模拟信号通道
- 模拟信号输出 D/A 分辨率 12Bit (A 型具备该功能)
- 模拟信号输出通道 1 路 (A 型具备该功能)
- 8Bit DI/8Bit DO TTL/COMS 兼容数字量输入/输出
- A/D 转换触发工作方式:软件触发
- A/D 转换数据传输方式:查询方式

2.2 技术指标

- 输入电压范围:±5V、0-10V
- 输入阻抗: > 100 MΩ
- A/D 转换时间: 8.5uS
- A/D 转换精度: 优于±0.2%(10V 满量程)
- 输出电压范围:±5V、0-5V、0-10V (A 型具备该功能)
- 数字量输入/输出电压:TTL/COMS 兼容

三、使用

3.1 寄存器功能描述

① 模拟信号输入通道选择寄存器(写操作)

Offset=00H, Offset:相对地址的偏移量(详细操作见软件说明书)

D15	D14	D13	D12	D11	D10	D09	D08	D07	D06	D05	D04	D03	D02	D01	D00
X	X	X	X	X	X	X	X	X	X	X	X	C3	C2	C1	C0

对 Offset=00H 写操作即设定 A/D 转换通道代码, X:无意义

C3~C0: 模拟信号输入通道选择位

C3	C2	C1	C0	通道代码	通道号
0	0	0	0	CH00	通道 0
0	0	0	1	CH01	通道 1
.

1	1	1	0	CHOE	通道 14
1	1	1	1	CHOF	通道 15

② 软件启动 A/D 转换(写操作)

Offset=02H, Offset: 相对地址的偏移量(详细操作见软件说明书)

D15~D0 此时无意义, 写 Offset=02H 即启动 A/D 转换

③ 查询 A/D 转换状态位+A/D 转换数据 (读操作)

Offset=0AH, Offset: 相对地址的偏移量(详细操作见软件说明书)

读数据格式如下:

D15	D14	D13	D12	D11	D10	D09	D08	D07	D06	D05	D04	D03	D02	D01	D00
Z	X	X	X	B11	B10	B9	B8	B7	B6	B5	B4	B3	B2	B1	B0

B11~B0 : 为 12 Bit A/D 转换数据

Z: A/D 转换状态位 Z=1, 正在进行转换

Z=0, 转换结束, 可以读取 A/D 转换数据

读 A/D 转换数据意义:

A/D 转换输出数据值 (B11~B0)	输入 ±5V	输入 0-10V
0 0 0 0 0 0 0 0 0 0 0 0	-5V	0V
1 0 0 0 0 0 0 0 0 0 0 0	0V	+5V
1 1 1 1 1 1 1 1 1 1 1 1	+5V	+10V

④ 写 D/A 转换数据寄存器 (写操作), (注: 只有 A 型具备 D/A)

Offset=06H, Offset: 相对地址的偏移量(详细操作见软件说明书)

写 12Bit D/A 数据寄存器数据格式如下:

D15	D14	D13	D12	D11	D10	D09	D08	D07	D06	D05	D04	D03	D02	D01	D00
X	X	X	X	B11	B10	B09	B08	B07	B06	B05	B04	B03	B02	B01	B00

B11~B00: D/A 转换的 12Bit 数据, X: 未用位

写 D/A 转换数据对应关系:

输出数据值(B11~B00)	输出 0-5V	输出 ±5V	输出 0-10V
0 0 0 0 0 0 0 0 0 0 0 0	0V	-5V	0V
1 0 0 0 0 0 0 0 0 0 0 0	+2.5V	0V	+5V
1 1 1 1 1 1 1 1 1 1 1 1	+5V	+5V	+10V

⑤ 8 位 TTL/COMS 兼容数据输出寄存器数据格式(写操作)

Offset=0CH, Offset:相对地址的偏移量(详细操作见软件说明书)

D15	D14	D13	D12	D11	D10	D09	D08	D07	D06	D05	D04	D03	D02	D01	D00
X	X	X	X	X	X	X	X	A07	A06	A05	A04	A03	A02	A01	A00

Ao7~Ao0 : 8Bit 数字量输出,对应 J1 的 Ao7~Ao0 ,X:无意义

⑥ 8 位 TTL/COMS 兼容数据输入寄存器数据格式 (读操作)

Offset=0EH, Offset:相对地址的偏移量(详细操作见软件说明书)

D15	D14	D13	D12	D11	D10	D09	D08	D07	D06	D05	D04	D03	D02	D01	D00
X	X	X	X	X	X	X	X	B17	B16	B15	B14	B13	B12	B11	B10

BI7~BI0 : 8Bit 数字量输入, BI7~BI0 对应 J1 的 BI7~BI0 ,X:无意义

操作命令表:

OFFSET =	操作	操作意义
00H	写操作	模拟信号输入通道选择寄存器,见(1)
02H	写操作	启动 A/D 转换,见(2)
06H	写操作	写 D/A 转换数据寄存器见(4)
0AH	读操作	查询 A/D 转换状态位+A/D 转换数据,见(3)
0CH	写操作	8 位 TTL 数据量输出寄存器,见(5)
0EH	读操作	8 位 TTL 数据量输入寄存器,见(6)

3.2 工作方式

A/D 部分转换触发方式

软件触发方式:

执行 Offset=02H 写命令,触发 A/D 转换

A/D 部分的数据传输: 查询方式

A/D 转换开始后,执行 Offset=0AH 读命令,Z=0 说明 A/D 转换完成可读数据

3.3 跨接线选择定义:

(1) JP1:用于设定 D/A 输出范围

① 1,2 短接, -5V~+5V

② 2,3 短接, 0~+10V

③ 3,4 短接, 0~+5V

(2) JP2 :用于设定 A/D 范围

① 2,3 短接, 模拟输入选择单极性 0V - 10V

② 1,2 短接, 模拟输入选择双极性, -5V - +5V, 出品状态

3.4 调整电位器定义

PR1: D/A 输出基准源调整电位器

PR2: D/A 输出零偏移调整电位器

PR3: D/A 输出满量程调整电位器

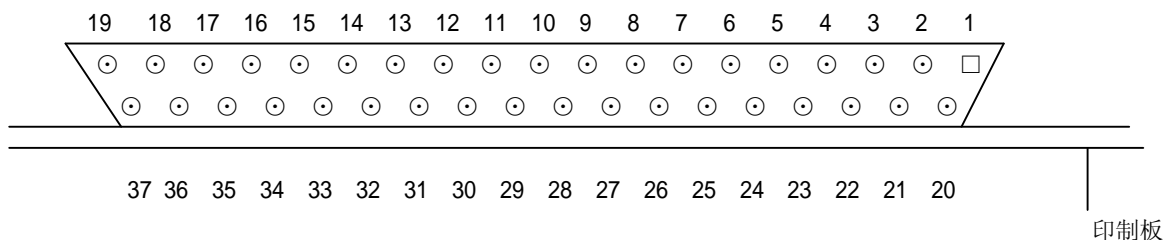
PR4: A/D 转换单极性输入零偏移调整电位器

PR5: A/D 转换双极性输入零偏移调整电位器

PR6: A/D 转换满量程调整电位器

3.5 37 芯孔式 D 型插头信号输入/输出插座定义:

J1 模拟输入/模拟输出数字量输入/数字量输出定义:



定义如下:

定义	GND	..	CH0E	CH0C	CH0A	CH08	CH06	CH04	CH02	CH00
J1 序号	19	..	8	7	6	5	4	3	2	1
J1 序号		..	27	26	25	24	23	22	21	20
定义		..	CH0F	CH0D	CH0B	CH09	CH07	CH05	CH03	CH01

定义	GND	+5V	D/A	B11	B13	B15	B17	A06	A04	A02	A00
序号	19	18	17	16	15	14	13	12	11	10	9
序号		37	36	35	34	33	32	31	30	29	28
定义		+12V	-12V	B10	B12	B14	B16	A07	A05	A03	A01

CH00~CH0F: 模拟输入通道 0~15

A00~A07 : 8BIT 数字量输出

T: 010-62615886, 62579956

BI0~BI7 : 8BIT 数字量输入
 D/A1: 模拟输出通道
 GND: 计算机系统的地
 +5V: 计算机系统的+5V
 +12V/-12V: 计算机系统的+12V/-12V

3.6 校准

A/D 部分校准

- a. 单极性输入 0~+10V: 任一通道对地短接, 调整 PR4, 使转换值为 001H;
 输入任一通道接 7.5V, 调整 PR6, 使转换值为 C00H;
- b. 双极性输入 -5V~+5V: 任一通道对地短接, 调整 PR5, 使转换值为 800H;
 输入任一通道 4.997V 调整 PR6 使转换值为 FFEH; 需反复调整

D/A 部分校准 (AMPCI-9101A)

基准源调整: 调整 PR1 使 336-5V 输出+5.000V, 出厂时已调整。

使 D/A 通道输出 0V, 调整 PR2, 测 D/A 使 D/A 输出为 0.000V

使 D/A 通道输出 10V, 调整 PR3, 测 D/A 使 D/A 输出为正满度

四、产品成套性

- | | |
|----------------------|----|
| ① AMPCI-9101 成品板 | 壹块 |
| ② 37 芯备件 | 壹套 |
| ③ AMPCI-9101 板说明书 | 壹本 |
| ④ AMPCI-9101 板软件安装光盘 | 壹张 |

五、产品保修

本产品自售出之日起 2 年内, 用户遵守储存和运输及使用要求的, 本公司负责免费修理。

操作过程举例(软件安装完毕后):

1. 0 通道 A/D 转换一次

- (1) 0000H 送 Offset=00H ; 设 00 通道
- (2) 0000H 送 Offset=02H ; 启动 A/D
- (4) 读 Offset=0AH ; 查询 Z 标志位若 Z=0 则已转换结束
- (5) 读 Offset=0AH ; 读 A/D 转换数据

2. 数字量口输出 56H, 读数字量口输入

- (1) 0056H 送 Offset=0CH ; 写数字量输出口 8 位数据
- (2) 读 Offset=0EH ; 读数字量输入口 8 位数据

注: 对 Offset=##H 的读写操作见后面 AMPCI 软件说明书

注意事项:

1. PCI 卡和驱动软件安装完毕后, 若将 PCI 卡取下, 再重新将卡插入 PCI 插槽时若插入的不是安装软件时的 PCI 插槽, 将无法操作, 此时需要在 WINDOWS 中卸掉系统中该 PCI 卡驱动,

再重新安装该卡驱动软件,当然如果重新插入原 PCI 插槽将不存在上述问题,因此使用时最好在一个 PCI 槽

WIN98 AMPCI 工业 I/O 卡软件使用说明

软件运行环境包括 Windows95 和 Windows98 (以下简称为 Windows)。推荐用户使用 Windows98。

一、软件列表

所有软件被包含在光盘内,常用软件的位置及主要功能如表 1 所示:

表 1 光盘内常用软件列表

位置与名称	主要功能描述
Win_app\wdreg_gui.exe	启动 Windows 驱动程序的可执行文件
Win_driver\PLX9050.inf	安装 Windows 驱动程序的配置文件
Win_driver\windrvr.vxd	AMPCI 的 Windows 驱动程序
AMPCI-9101 产品说明书	AMPCI 工业 I/O 卡软件使用说明
Win_app\setup.exe	AMPCI 的 Windows 安装程序 VC 版;调用静态库
Win_pci_all\setup.exe	AMPCI 测试 1 的 Windows 安装程序 VB 版;调用动态库,AMPCI-9101 测试
Win_pci_dodi\setup.exe	AMPCI 测试 2 的 Windows 安装程序 VB 版;调用动态库,AMPCI 卡读写测试

二、软件安装

Windows 环境中,软件安装步骤如下:

1、将 AMPCI 卡插入到主机的某一 PCI 插槽内。

2、启动 Windows98。

3、安装驱动程序。当出现“添加新硬件向导”对话框时,将带有驱动程序的光盘放入光驱,并选择“下一步”;在随后出现的对话框中,选择或输入光盘的 win_driver 文件夹。并依照相应提示,完成驱动程序的安装。

注意:如果同时插入两块 AMPCI 板卡,则这一步需安装两次,三块板卡时需安装三次,即插入几块板卡就需安装几次,这一过程主机均有提示。

4、检查驱动程序。启动 Windows 资源管理器,观察 Windows 文件夹内的 system\vm32 文件夹中是否包含文件 windrvr.vxd (在观察前,应通过资源管理器“查看”菜单的“文件夹选项”菜单项,使资源管理器能够显示所有文件)。若有,说明驱动程序安装正确;若无,可通过资源管理器将 win_driver 文件夹内的 windrvr.vxd 文件复制到 Windows 文件夹内的 system\vm32 文件夹中。

5、启动驱动程序。通过“开始”菜单的“运行...”,带参数运行光盘 win_app 文件夹中的 wdreg_gui.exe 文件,格式如下:

E:\win_app\wdreg_gui.exe -vxd install(假设 E:为光驱)。

6、安装应用程序。

(1)运行光盘 win_app 文件夹中的 setup.exe 文件，完成 VC 应用程序的安装。

(2)运行光盘 win_pci_dodi 文件夹中的 setup.exe 文件，完成 VB 应用程序的安装-pci 卡测试程序。

(3)运行光盘 win_pci_all 文件夹中的 setup.exe 文件，完成 VB 应用程序的安装-pci 卡测试程序。

7、启动应用程序。通过任务栏，选择开始→程序→PCI 总线插卡→PLX9050，即可启动应用程序。

三、软件开发

为便于用户开发自己的应用程序，提供了相应的开发文档、程序和函数。

在 Windows 环境下，用户层应用程序不能对端口直接进行读写，必须借助 Windows 驱动程序。但是，调用驱动程序的方式和通常函数调用的方式不同，编程人员需要具备相关知识。为方便广大用户，我们在编制好的 Windows 驱动程序之上，又开发了一组函数。用户可以用通常的函数调用方式，对这组函数进行调用，以完成对端口的读写。

1、函数介绍

(1) `BOOL PLX9052_Open(PLX9052_HANDLE *phPLX9052, unsigned long VendorID, unsigned long DeviceID, unsigned long nCardNum, unsigned long Options)`

功能：获得对 PLX9050 进行读写的操作句柄。在进行读写前，必须首先调用此函数。

参数：VendorID——生产商 ID，应调用宏 `PLX9052_DEFAULT_VENDOR_ID`

DeviceID——设备 ID，应调用宏 `PLX9052_DEFAULT_DEVICE_ID`

nCardNum——当同时使用多块 PLX9050 卡时，用以指示卡号。当仅使用一块时，应为 0；当使用二块卡时，应为 0、1；其余类推。计算机内所插入 PLX9050 卡的数量，可由函数 `PLX9052_CountCards()` 获得。

Options——获得操作句柄方式选择，应为 0

返回值：如果能够获得操作句柄，返回 TRUE，并得到操作句柄指针

*phPLX9052；如果不能获得操作句柄，返回 FALSE。

C 调用：

```
BOOL PLX9052_Open (PLX9052_HANDLE *phPLX9052, unsigned long VendorID, unsigned long DeviceID, unsigned long nCardNum, unsigned long Options)
```

VB 调用：

```
Declare Function PLX9052_Open Lib "pcidll" Alias "#2" (phPLX9052 As Long, ByVal dwVendorID As Long, ByVal dwDeviceID As Long, ByVal nCardNum As Long, ByVal dwOptions As Long) As Boolean
```

注意：此函数的卡号顺序是从 0 开始的，而演示程序中的卡号是从 1 开始的，这样做只是出于习惯。因此，编程时卡号必须从 0 开始。

(2) `unsigned short int PLX9052_ReadWord (PLX9052_HANDLE hPLX9052, PLX9052_ADDR addrSpace, unsigned long Offset)`

功能：从 PLX9050 的指定端口读入字。

参数：hPLX9052——通过调用 `PLX9052_Open ()` 函数获得的操作句柄

`addrSpace`——地址空间。由于 PLX9050 采取 I/O 映射方式，此处应调用宏 `PLX9052_AD_BAR3`

`Offset`——相对基址的偏移量。如读端口 1A，此处也应为 1A。

返回值：从 PLX9050 指定端口读入的字。

C 调用：

```
unsigned short int PLX9052_ReadWord (PLX9052_HANDLE hPLX9052, PLX9052_ADDR addrSpace, unsigned long Offset)
```

VB 调用：

```
Declare Function PLX9052_ReadWord Lib "pcidll" Alias "#5" (ByVal phPLX9052 As Long, ByVal addrSpace As Long, ByVal dwOffset As Long) As Integer
```

(3) `void PLX9052_WriteWord (PLX9052_HANDLE hPLX9052, PLX9052_ADDR addrSpace, unsigned long Offset, unsigned short data)`

功能：向 PLX9050 的指定端口写入字。

参数：`hPLX9052`——通过调用 `PLX9052_Open ()` 函数获得的操作句柄

`addrSpace`——地址空间。由于 PLX9050 采取 I/O 映射方式，此处应调用宏 `PLX9052_AD_BAR3`。

`Offset`——相对基址的偏移量。如读端口 1A，此处也应为 1A。

`data`——需向指定端口写入的字

返回值：无返回值

C 调用：

```
void PLX9052_WriteWord (PLX9052_HANDLE hPLX9052, PLX9052_ADDR addrSpace, unsigned long Offset, unsigned short data)
```

VB 调用：

```
Declare Sub PLX9052_WriteWord Lib "pcidll" Alias "#9" (ByVal phPLX9052 As Long, ByVal addrSpace As Long, ByVal dwOffset As Long, ByVal data As Long)
```

(4) `void PLX9052_Close(PLX9052_HANDLE hPLX9052)`

功能：关闭操作句柄。在程序结束前，必须调用此函数。

参数：`hPLX9052`——通过调用 `PLX9052_Open ()` 函数获得的操作句柄

返回值：无返回值

(5) `int PLX9052_CountCards(unsigned long VendorID, unsigned long DeviceID)`

功能：获得计算机内所插入的 PLX9050 板卡的数量。

参数：`VendorID`——生产商 ID，应调用宏 `PLX9052_DEFAULT_VENDOR_ID`

`DeviceID`——设备 ID，应调用宏 `PLX9052_DEFAULT_DEVICE_ID`

返回值：计算机内所插入的 PLX9050 板卡的数量。

C 调用：

```
void PLX9052_Close(PLX9052_HANDLE hPLX9052)
```

VB 调用:

```
Declare Sub PLX9052_Close Lib "pcidll" Alias "#3" (ByVal hPLX9052 As Long)
```

```
[5] long Hex_dec ( char Hex_str[] )
```

功能: 16 进制字符串转换成 10 进制数

参数: Hex_str——字符串

返回值: 长整数 ;< 0 表示转换失败

C 调用:

```
long Hex_dec( char Hex_str[] )
```

VB 调用:

```
Declare Function Hex_dec Lib "pcidll" Alias "#21" (ByVal str As String) As Long
```

2、函数调用前的准备

要正确调用函数，必须做好以下准备工作:

(1) 正确安装 PLX9050 的驱动程序。具体方法参见: 二、软件安装中的相应内容。若已进行过安装，此处不需要再次进行。

(2) 正确启动 PLX9050 的驱动程序。具体方法参见: 二、软件安装中的相应内容。若已进行过安装，此处不需要再次进行。

(3) 在运行 setup.exe 过程中建立的放有 Windows 软件的文件夹中 (缺省情况下为: c:\program files\PLX9050) 找到 include 和 lib 子文件夹, 并将该两文件夹及其内的所有文件复制到由 Visual C++ 创建的文件夹内 (也就是进行 Visual C++ 编程时生成的文件夹)。

(4) 在源程序中包含头文件 plx9052_lib.h。其格式为:

```
#include "include\plx9052_lib.h"
```

(5) 连接时, 在“工程”菜单中选择“设置”。在弹出的对话框中, 选择“Link”标签, 在 Project Options 框中输入: /DEFAULTLIB:lib\plx9052_lib.lib。注意: 在输入时不要输入空格。然后调用相应函数, 进行编译即可。

(6) 如果使用动态连接库 pcidll.dll 则需要将 pcidll.dll 拷入由 Visual C++ 创建的文件夹内, 并且不需要/DEFAULTLIB:lib\plx9052_lib.lib

(7) 如果使用 Vb 调用动态连接库 pcidll.dll 则需要将 bits.bas、pci_regs.bas、plx_9052.bas 添加到工程中。

四、函数调用举例

1. 使用静态连接库 plx9052_lib.lib

在 Visual C++ 6.0 中, 如若想读端口 04H, 将值 50 写入端口 18H, 函数调用方法如下:

```
PLX9052_HANDLE hPLX9052; //定义操作句柄
```

```
PLX9052_Open (&hPLX9052, PLX9052_DEFAULT_VENDOR_ID,
```

```
PLX9052_DEFAULT_DEVICE_ID, 0, 0); //获得操作句柄。
```

```
unsigned short int x; //定义字 x
```

```
x=PLX9052_ReadWord (hPLX9052, PLX9052_AD_BAR3, 0x04); //读端口 04H
```

```
PLX9052_WriteWord (hPLX9052, PLX9052_AD_BAR3, 0x18, 50); // 将值 50 写入端口 18H
```

```
PLX9052_Close(hPLX9052);//关闭操作句柄
```

2.使用动态连接库 pci.dll

在 Visual C++ 6.0 中，如若想读端口 04H，将值 50 写入端口 18H，动态连接库函数调用方法如下：

```
//函数定义
```

```
WORD (*PLX9052_ReadWord)(PLX9052_HANDLE hPLX9052, PLX9052_ADDR addrSpace,
    DWORD dwOffset);
```

```
char (*PLX9052_WriteWord)(PLX9052_HANDLE hPLX9052, PLX9052_ADDR addrSpace,
    DWORD dwOffset, WORD data);
```

```
BOOL (*PLX9052_Open)(PLX9052_HANDLE *, DWORD, DWORD, DWORD, DWORD);
```

```
char (*PLX9052_Close)(PLX9052_HANDLE hPLX9052);
```

```
DWORD (*PLX9052_CountCards)(DWORD dwVendorID, DWORD dwDeviceID);
```

```
//加载动态连接库
```

```
HMODULE hMyDll;
```

```
hMyDll=LoadLibrary("pci.dll");
```

```
PLX9052_HANDLE hPLX9052; //定义操作句柄
```

```
//定义函数
```

```
PLX9052_Open=(BOOL (*)(PLX9052_HANDLE *, DWORD, DWORD, DWORD,
    DWORD))::GetProcAddress(hMyDll,"PLX9052_Open");
```

```
PLX9052_Close=(char (*)(PLX9052_HANDLE ))::GetProcAddress(hMyDll,"PLX9052_Close");
```

```
PLX9052_CountCards=(DWORD (*)(DWORD,
```

```
    DWORD))::GetProcAddress(hMyDll,"PLX9052_CountCards");
```

```
PLX9052_ReadWord=(WORD (*)(PLX9052_HANDLE hPLX9052, PLX9052_ADDR addrSpace,
    DWORD dwOffset))::GetProcAddress(hMyDll,"PLX9052_ReadWord");
```

```
PLX9052_WriteWord=(char (*)(PLX9052_HANDLE, PLX9052_ADDR, DWORD,
    WORD))::GetProcAddress(hMyDll,"PLX9052_WriteWord");
```

```
//调用开始
```

```
nWritecard=1;
```

```
PLX9052_Open (&hPLX9052, PLX9052_DEFAULT_VENDOR_ID,
    PLX9052_DEFAULT_DEVICE_ID,m_nWritecard-1, PLX9052_OPEN_USE_INT); //获
    得操作句柄。
```

```
unsigned short int x; //定义字 x
```

```
x=PLX9052_ReadWord (hPLX9052, PLX9052_AD_BAR3, 0x04); //读端口 04H
```

```
PLX9052_WriteWord (hPLX9052, PLX9052_AD_BAR3, 0x18, 50); // 将值 50 写入端口 18H
```

```
PLX9052_Close(hPLX9052); //关闭操作句柄
```

注意：上述例子是对“0”号卡进行操作，若对“1”号卡进行操作，则为

```
PLX9052_Open (&hPLX9052, PLX9052_DEFAULT_VENDOR_ID,
```

PLX9052_DEFAULT_DEVICE_ID, 1, 0); 其余部分与上述例子相同。

说明:端口偏移地址##H 定义见硬件说明书

3. Visual Basic 调用动态连接库

(1) 首先将 bits.bas、pci_regs.bas、plx9052.bas、windrvr.bas 四个模块文件添加到你的工程文件中。

(2) 对函数进行声明 (这部分在 plx9052.bas 文件中)

```
Declare Function PLX9052_CountCards Lib "pci.dll" Alias "#1" (ByVal dwVendorID As Long,
    ByVal dwDeviceID As Long) As Long
```

```
Declare Function PLX9052_Open Lib "pci.dll" Alias "#2" (phPLX9052 As Long, ByVal
    dwVendorID As Long, ByVal dwDeviceID As Long, ByVal nCardNum As Long, ByVal
    dwOptions As Long) As Boolean
```

```
Declare Sub PLX9052_Close Lib "pci.dll" Alias "#3" (ByVal hPLX9052 As Long)
```

```
Declare Function PLX9052_ReadWord Lib "pci.dll" Alias "#5" (ByVal phPLX9052 As Long, ByVal
    addrSpace As Long, ByVal dwOffset As Long) As Integer
```

```
Declare Sub PLX9052_WriteWord Lib "pci.dll" Alias "#9" (ByVal phPLX9052 As Long, ByVal
    addrSpace As Long, ByVal dwOffset As Long, ByVal data As Long)
```

```
Declare Function Hex_dec Lib "pci.dll" Alias "#21" (ByVal str As String) As Long
```

```
Public Const PLX9052_AD_BAR3 = AD_PCI_BAR3
```

```
Public Const PLX9052_DEFAULT_VENDOR_ID As Long = &H10B5&
```

```
Public Const PLX9052_DEFAULT_DEVICE_ID As Long = &H9050&
```

```
Public Const PLX9052_OPEN_USE_INT As Long = &H1&
```

(3) 程序内容

```
Sub test_click()
    Dim hPLX9052 As Long
    Dim m_nCard As integer
    Dim m_nReadAdd As Long
    Dim m_nReadValue As Long
    Dim m_nWriteAdd As Long
    Dim m_nWriteValue As Long
    m_nCard = 1
    flg = PLX9052_Open(hPLX9052, PLX9052_DEFAULT_VENDOR_ID,
        PLX9052_DEFAULT_DEVICE_ID, m_nCard - 1, PLX9052_OPEN_USE_INT)
    m_nReadAdd = &h4;
    m_nReadValue = PLX9052_ReadWord(hPLX9052, PLX9052_AD_BAR3, m_nReadAdd)
    m_nWriteAdd = &h18
    m_nWriteValue = 50
    Call PLX9052_WriteWord(hPLX9052, PLX9052_AD_BAR3, m_nWriteAdd,
        m_nWriteValue)
    Call PLX9052_Close(hPLX9052)
```

End sub

WIN2K AMPCI 工业 I/O 卡 VB 调用使用说明

软件运行环境包括 Windows2000 和 WindowsNT（以下简称为 Windows）。推荐用户使用 Windows2000(Professional)。

一、软件列表

所有软件被包含在光盘内，常用软件的位置及主要功能如表 1 所示：

表 1 光盘内常用软件列表

位置与名称	主要功能描述
DRIVERS\DLL	启动 Windows 驱动程序的动态链接库
Pcisdk.inf	安装 Windows 驱动程序的配置文件
DRIVERS\SYS	AMPCI 的 Windows 驱动程序
DRIVERS\inc	H 文件
Samples\VBtest	VB 测试程序及源程序
Samples\VCtest	VC 测试程序及源程序

二、软件安装

Windows 环境中，软件安装步骤如下：

1、将 AMPCI 卡插入到主机的某一 PCI 插槽内。

2、启动 Windows2000。

3、安装驱动程序。当出现“添加新硬件向导”对话框时，将带有驱动程序的光盘放入光驱，并选择“下一步”；在随后出现的对话框中，选择或输入光盘的 Pcisdk.inf 文件。并依照相应提示，完成驱动程序的安装。

注意：如果同时插入两块 AMPCI 板卡，则这一步需安装两次，三块板卡时需安装三次，即插入几块板卡就需安装几次，这一过程主机均有提示。

4、安装完成后，AMPCI.SYS 自动拷贝到 WINNT\system32\driver 目录下；AMPCI.DLL 自动拷贝到 WINNT\system32 目录下。

5、应用程序。

将光盘里 Samples 文件夹下的 VBTEST 和 VCTEST 文件夹拷贝到自己喜欢的位置。打开后看到 Amtest.exe 文件，即是应用程序。

7、启动应用程序。双击 Amtest.exe，即可启动应用程序。

三、软件开发

为便于用户开发自己的应用程序，提供了相应的开发文档、程序和函数。

在 Windows 环境下，用户层应用程序不能对端口直接进行读写，必须借助 Windows 驱动程序。但是，调用驱动程序的方式和通常函数调用的方式不同，编程人员需要具备相关知识。为方便广

大用户，我们在编制好的 Windows 驱动程序之上，又开发了一组函数。用户可以用通常的函数调用方式，对这组函数进行调用，以完成对端口的读写。

函数介绍

(1) `BOOL PLX9052_Open(PLX9052_HANDLE *phPLX9052, unsigned long VendorID, unsigned long DeviceID, unsigned long nCardNum, unsigned long Options)`

功能：获得对 PLX9050 进行读写的操作句柄。在进行读写前，必须首先调用此函数。

参数：VendorID——生产商 ID，应调用宏 PLX9052_DEFAULT_VENDOR_ID

DeviceID——设备 ID，应调用宏 PLX9052_DEFAULT_DEVICE_ID

nCardNum——当同时使用多块 PLX9052 卡时，用以指示卡号。当仅使用一块时，应为 0；当使用二块卡时，应为 0、1。计算机内所插入 PLX9052 卡的数量，可由函数 PLX9052_CountCards() 获得。

Options——获得操作句柄方式选择，应为 0

返回值：如果能够获得操作句柄，返回 TRUE，并得到操作句柄指针 *phPLX9052；如果不能获得操作句柄，返回 FALSE。

VB 调用：

```
Declare Function PLX9052_Open Lib AMPCI.lib (phPLX9052 As Long, ByVal dwVendorID As Long, ByVal dwDeviceID As Long, ByVal nCardNum As Long, ByVal dwOptions As Long) As Boolean
```

(2) `unsigned short int PLX9052_ReadWord (PLX9052_HANDLE hPLX9052, PLX9052_ADDR addrSpace, unsigned long Offset)`

功能：从 PLX9052 的指定端口读入字。

参数：hPLX9052——通过调用 PLX9052_Open () 函数获得的操作句柄

addrSpace——地址空间。由于 PLX9052 采取 I/O 映射方式，此处应调用宏 PLX9052_AD_BAR3

Offset——相对基址的偏移量。如读端口 1A，此处也应为 1A。

返回值：从 PLX9052 指定端口读入的字。

VB 调用：

```
Declare Function PLX9052_ReadWord Lib AMPCI.lib (ByVal phPLX9052 As Long, ByVal addrSpace As Long, ByVal dwOffset As Long) As Integer
```

(3) `void PLX9052_WriteWord (PLX9052_HANDLE hPLX9052, PLX9052_ADDR addrSpace, unsigned long Offset, unsigned short data)`

功能：向 PLX9052 的指定端口写入字。

参数：hPLX9052——通过调用 PLX9052_Open () 函数获得的操作句柄

addrSpace——地址空间。由于 PLX9052 采取 I/O 映射方式，此处应调用宏 PLX9052_AD_BAR3。

Offset——相对基址的偏移量。如读端口 1A，此处也应为 1A。

`data`——需向指定端口写入的字

返回值：无返回值

VB 调用：

```
Declare Sub PLX9052_WriteWord Lib AMPCI.lib (ByVal phPLX9052 As Long, ByVal addrSpace As Long, ByVal dwOffset As Long, ByVal data As Long)
```

(4) `void PLX9052_Close(PLX9052_HANDLE hPLX9052)`

功能：关闭操作句柄。在程序结束前，必须调用此函数。

参数：`hPLX9052`——通过调用 `PLX9052_Open()` 函数获得的操作句柄

返回值：无返回值

(5) `int PLX9052_CountCards(unsigned long VendorID, unsigned long DeviceID)`

功能：获得计算机内所插入的 PLX9052 板卡的数量。

参数：`VendorID`——生产商 ID，应调用宏 `PLX9052_DEFAULT_VENDOR_ID`

`DeviceID`——设备 ID，应调用宏 `PLX9052_DEFAULT_DEVICE_ID`

返回值：计算机内所插入的 PLX9052 板卡的数量。

VB 调用：

```
Declare Sub PLX9052_Close Lib AMPCI.lib (ByVal hPLX9052 As Long)
```

[5] `long Hex_dec(char Hex_str[])`

功能：16 进制字符串转换成 10 进制数

参数：`Hex_str`——字符串

返回值：长整数 ;< 0 表示转换失败

VB 调用：

```
Declare Function Hex_dec Lib AMPCI.lib (ByVal str As String) As Long
```

3、函数调用前的准备

要正确调用函数，必须做好以下准备工作：

(1) 正确安装 PLX9052 的驱动程序。具体方法参见：二、软件安装中的相应内容。若已进行过安装，此处不需要再次进行。

(2) 正确启动 PLX9052 的驱动程序。具体方法参见：二、软件安装中的相应内容。若已进行过安装，此处不需要再次进行。

(3) 将 `Fang1` 添加到工程中。

四、函数调用举例

Visual Basic 调用动态连接库

(1) 首先将 `Module1` 模块文件添加到你的工程文件中。

(2) 对函数进行声明（这部分在 Module1.bas 文件中）

```

Declare Function PLX9052_CountCards Lib AMPCI.lib (ByVal dwVendorID As Long, ByVal
    dwDeviceID As Long) As Long
Declare Function PLX9052_Open Lib AMPCI.lib (phPLX9052 As Long, ByVal dwVendorID As
    Long, ByVal dwDeviceID As Long, ByVal nCardNum As Long, ByVal dwOptions As Long)
    As Boolean
Declare Sub PLX9052_Close Lib AMPCI.lib (ByVal hPLX9052 As Long)
Declare Function PLX9052_ReadWord Lib AMPCI.lib (ByVal phPLX9052 As Long, ByVal
    addrSpace As Long, ByVal dwOffset As Long) As Integer
Declare Sub PLX9052_WriteWord Lib AMPCI.lib (ByVal phPLX9052 As Long, ByVal addrSpace
    As Long, ByVal dwOffset As Long, ByVal data As Long)
Declare Function Hex_dec Lib AMPCI.lib (ByVal str As String) As Long
Public Const PLX9052_AD_BAR3 = AD_PCI_BAR3
Public Const PLX9052_DEFAULT_VENDOR_ID As Long = &H10B5&
Public Const PLX9052_DEFAULT_DEVICE_ID As Long = &H9050&
Public Const PLX9052_OPEN_USE_INT As Long = &H1&

```

(3) 程序内容

```

Sub test_click()
    Dim hPLX9052 As Long
    Dim m_nCard As integer
    Dim m_nReadAdd As Long
    Dim m_nReadValue As Long
    Dim m_nWriteAdd As Long
    Dim m_nWriteValue As Long
    m_nCard = 1
    flg = PLX9052_Open(hPLX9052, PLX9052_DEFAULT_VENDOR_ID,
        PLX9052_DEFAULT_DEVICE_ID, m_nCard - 1, PLX9052_OPEN_USE_INT)
    m_nReadAdd = &h4;
    m_nReadValue = PLX9052_ReadWord(hPLX9052, PLX9052_AD_BAR3, m_nReadAdd)
    m_nWriteAdd = &h18
    m_nWriteValue = 50
    Call PLX9052_WriteWord(hPLX9052, PLX9052_AD_BAR3, m_nWriteAdd,
        m_nWriteValue)
    Call PLX9052_Close(hPLX9052)

End sub

```

注意：上述例子是对“0”号卡进行操作，若对“1”号卡进行操作，则为
 PLX9052_Open (&hPLX9052, PLX9052_DEFAULT_VENDOR_ID,
 PLX9052_DEFAULT_DEVICE_ID, 1, 0); 其余部分与上述例子相同。
 说明:端口偏移地址##H 定义见硬件说明书

WIN2K AMPCI 工业 I/O 卡 VC 调用使用说明

软件运行环境包括 Windows2000 和 WindowsNT（以下简称为 Windows）。推荐用户使用 Windows2000(Professional)。

一、软件列表

所有软件被包含在光盘内，常用软件的位置及主要功能如表 1 所示：

表 1 光盘内常用软件列表

位置与名称	主要功能描述
DRIVERS\DLL	启动 Windows 驱动程序的动态链接库
Pcisdk.inf	安装 Windows 驱动程序的配置文件
DRIVERS\SYS	AMPCI 的 Windows 驱动程序
DRIVERS\inc	H 文件
Samples\VBtest	VB 测试程序及源程序
Samples\VCtest	VC 测试程序及源程序

二、软件安装

Windows 环境中，软件安装步骤如下：

- 1、将 AMPCI 卡插入到主机的某一 PCI 插槽内。
- 2、启动 Windows2000。

3、安装驱动程序。当出现“添加新硬件向导”对话框时，将带有驱动程序的光盘放入光驱，并选择“下一步”；在随后出现的对话框中，选择或输入光盘的 Pcisdk.inf 文件。并依照相应提示，完成驱动程序的安装。

注意：如果同时插入两块 AMPCI 板卡，则这一步需安装两次，三块板卡时需安装三次，即插入几块板卡就需安装几次，这一过程主机均有提示。

4、安装完成后，AMPCI.SYS 自动拷贝到 WINNT\system32\driver 目录下；AMPCI.DLL 自动拷贝到 WINNT\system32 目录下。

5、应用程序。

将光盘里 Samples 文件夹下的 VBTEST 和 VCTEST 文件夹拷贝到自己喜欢的位置。打开后看到 Amtest.exe 文件，即是应用程序。

7、启动应用程序。在 DOS 相应目录下键入 DLLTEST，即可启动应用程序。

三、软件开发

为便于用户开发自己的应用程序，提供了相应的开发文档、程序和函数。

在 Windows 环境下,用户层应用程序不能对端口直接进行读写,必须借助 Windows 驱动程序。但是,调用驱动程序的方式和通常函数调用的方式不同,编程人员需要具备相关知识。为方便广大用户,我们在编制好的 Windows 驱动程序之上,又开发了一组函数。用户可以用通常的函数调用方式,对这组函数进行调用,以完成对端口的读写。

4、函数介绍

(1) `BOOL PLX9052_Open (PLX9052_HANDLE *phPLX9052, unsigned long VendorID, unsigned long DeviceID, unsigned long nCardNum, unsigned long Options)`

功能: 获得对 PLX9052 进行读写的操作句柄。在进行读写前, 必须首先调用此函数。

参数: VendorID——生产商 ID, 应调用宏 `PLX9052_DEFAULT_VENDOR_ID`

DeviceID——设备 ID, 应调用宏 `PLX9052_DEFAULT_DEVICE_ID`

nCardNum——当同时使用多块 PLX9052 卡时, 用以指示卡号。当仅使用一块时, 应为 0; 当使用二块卡时, 应为 0, 1; 其余类推。计算机内所插入 PLX9052 卡的数量, 可由函数 `PLX9052_CountCards()` 获得。

Options——获得操作句柄方式选择, 应为 0

返回值: 如果能够获得操作句柄, 返回 `TRUE`, 并得到操作句柄指针

*phPLX9052; 如果不能获得操作句柄, 返回 `FALSE`。

(2) `unsigned short int PLX9052_ReadWord (PLX9052_HANDLE hPLX9052, PLX9052_ADDR addrSpace, unsigned long Offset)`

功能: 从 PLX9052 的指定端口读入字。

参数: hPLX9052——通过调用 `PLX9052_Open ()` 函数获得的操作句柄

addrSpace——地址空间。由于 PLX9050 采取 I/O 映射方式, 此处应调用宏 `PLX9052_AD_BAR3`

Offset——相对基址的偏移量。如读端口 1A, 此处也应为 1A。

返回值: 从 PLX9052 指定端口读入的字。

(3) `void PLX9052_WriteWord (PLX9052_HANDLE hPLX9052, PLX9052_ADDR addrSpace, unsigned long Offset, unsigned short data)`

功能: 向 PLX9052 的指定端口写入字。

参数: hPLX9052——通过调用 `PLX9052_Open ()` 函数获得的操作句柄

addrSpace——地址空间。由于 PLX9050 采取 I/O 映射方式, 此处应调用宏 `PLX9052_AD_BAR3`。

Offset——相对基址的偏移量。如读端口 1A, 此处也应为 1A。

data——需向指定端口写入的字

返回值: 无返回值

(4) `void PLX9052_Close(PLX9052_HANDLE hPLX9052)`

功能: 关闭操作句柄。在程序结束前, 必须调用此函数。

参数: hPLX9052——通过调用 `PLX9052_Open ()` 函数获得的操作句柄

返回值: 无返回值

(5) `int PLX9052_CountCards(unsigned long VendorID, unsigned long DeviceID)`

功能：获得计算机内所插入的 PLX9052 板卡的数量。

参数：VendorID——生产商 ID，应调用宏 PLX9052_DEFAULT_VENDOR_ID

DeviceID——设备 ID，应调用宏 PLX9052_DEFAULT_DEVICE_ID

返回值：计算机内所插入的 PLX9052 板卡的数量。

5、函数调用前的准备

要正确调用函数，必须做好以下准备工作：

(1) 正确安装 PLX9052 的驱动程序。具体方法参见：二、软件安装中的相应内容。若已进行过安装，此处不需要再次进行。

(2) 正确启动 PLX9052 的驱动程序。具体方法参见：二、软件安装中的相应内容。若已进行过安装，此处不需要再次进行。

(3) 找到 inc 文件夹和 lib 文件，并将该两文件夹及其内的所有文件复制到由 Visual C++ 创建的文件夹内（也就是进行 Visual C++ 编程时生成的文件夹）。

(4) 在源程序中包含头文件 `ampci.h`。其格式为：

```
#include "include\ampci.h"
```

(5) 连接时，在“工程”菜单中选择“设置”。在弹出的对话框中，选择“Link”标签，在 Project Options 框中输入：`/DEFAULTLIB:lib\ampci.lib`。注意：在输入时不要输入空格。然后调用相应函数，进行编译即可。

三、函数调用举例

在 Visual C++ 6.0 中，如若想读端口 04H，将值 50 写入端口 18H，函数调用方法如下：

```
PLX9052_HANDLE hPLX9052; //定义操作句柄
```

```
PLX9052_Open (&hPLX9052, PLX9052_DEFAULT_VENDOR_ID,
```

```
PLX9052_DEFAULT_DEVICE_ID, 0, 0); //获得操作句柄。
```

```
unsigned short int x; //定义字 x
```

```
x=PLX9052_ReadWord (hPLX9052, PLX9052_AD_BAR3, 0x04); //读端口 04H
```

```
PLX9052_WriteWord (hPLX9052, PLX9052_AD_BAR3, 0x18, 50); // 将值 50 写入端口 18H
```

```
PLX9052_Close(hPLX9052); //关闭操作句柄
```

注意：上述例子是对“0”号卡进行操作，若对“1”号卡进行操作，则为

```
PLX9052_Open (&hPLX9052, PLX9052_DEFAULT_VENDOR_ID,
```

```
PLX9052_DEFAULT_DEVICE_ID, 1, 0); 其余部分与上述例子相同。
```

注：C++ Builder 及 Delphi 环境下 WIN98/WIN2K 对 DLL 的调用请阅读随板附带光盘说明和例程举例