AMPCI-9110 通用数据采集控制板

一、 概述

AMPCI-9110 板是 PCI 总线通用采集控制板,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及计数定时系统。

AMPCI-9110 板为用户提供了单端 32 路模拟量数据采集输入通道，模拟量输入通道具有程控放大功能,2 路 12Bit 模拟量电压信号输出,16Bit TTL 数字量输入和 16Bit TTL 数字量输出,配接 AMPCD821 光隔端子板实现光隔 I/O,可直接驱动继电器，3 路 16 位计数定时通道(一片 82C54),基准时钟 2M,它的 1、2 通道级连使用,用来产生定时触发 A/D 转换的时钟信号,其通道 0 保留给用户自行使用,构成脉冲计数、频率测量、脉冲信号发生器等电路。

对 AMPCI-9110 板的所有读写操作均为 16Bit 即 D00～D15,当对 82C54 进行读写时只有 D00～D07 有效,同样 A/D 转换数据一次读入的为 B00～B11。

二、 性能和技术指标

2.1 性能

- 模拟信号输入 A/D 分辨率 12Bit
- 32 路模拟信号通道
- 模拟信号输入具有程控放大
- 模拟信号输出 D/A 分辨率 12Bit (A 型不具备该功能)
- 模拟信号输出通道 2 路 (A 型不具备该功能)
- 16Bit DI/16Bit DO TTL 兼容数字量输入/输出
- 1 路计数定时通道(留给用户自行使用) (A 型不具备该功能)
- 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、等待方式

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±0.1%(10V 满量程)
- 模拟信号输入程控放大倍数: 1/2/4/8/16(AD526) (A 型不具备该功能)
- 输出电压范围: ±5V、0-5V、0-10V (A 型不具备该功能)
- 计数定时部分: 0 通道留给用户使用,1 通道和 2 通道级联用做板内定时触发,内部时钟基准 2MHz(A 型不具备该功能)

三、 使用

3.1 寄存器功能描述

①: 模拟输入通道选择、增益选择、D/A 和 82C54 地址选择寄存器（写操作）

Offset=00H，Offset:相对地址的偏移量,即该写操作的 I/O 地址(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	GT2	GT1	A1	A0	G2	G1	G0	C4	C3	C2	C1	C0

C4～C0: 模拟信号输入通道选择位，

C4	C3	C2	C1	C0		
0	0	0	0	0	CH00	通道 0
0	0	0	0	1	CH01	
0	0	0	1	0	CH02	
.	.	.	.	.	.	.
1	1	1	1	0	CH1E	
1	1	1	1	1	CH1F	通道 31

A1/A0: 82C54、D/A 转换器的地址选择位(A 型不具备该功能，此时无意义)

a:	A1	A0	8254 内部寄存器
	0	0	#0 通道数据寄存器地址
	0	1	#1 通道数据寄存器地址
	1	0	#2 通道数据寄存器地址
	1	1	控制寄存器地址

b:	A1	A0	D/A 转换器地址选择位(A 型不具备)
	0	0	D/A1 通道低 8 数据寄存器地址
	0	1	D/A1 通道高 4 数据寄存器地址
	1	0	D/A2 通道低 8 数据寄存器地址
	1	1	D/A2 通道高 4 数据寄存器地址

G2～G0:程控放大增益选择位(A 型不具备该功能，此时无意义)

G2	G1	G0	程控放大增益倍数
0	0	0	1
0	0	1	2
0	1	0	4
0	1	1	8
1	0	0	16

GT1/GT2:82C54 的 1 和 2 通道的 **GATE1/GATE2** 控制位(A 型不具备该功能，此时无意义)
GT1:计数定时器 1 的 **GATE1** 控制端
GT2:计数定时器 2 的 **GATE2** 控制端(详细操作见 **8254** 操作手册)

② 启动 A/D 转换 (写操作)
Offset=02H，**Offset** 是相对地址的偏移量，该写操作的 I/O 地址，(详细操作见软件说明书)

D15～D00 此时无意义

③ 计数定时器 8254 片选 (读操作+写操作)

Offset=04H，Offset 是相对地址的偏移量，该读写操作的 I/O 地址(详细操作见软件说明书)

此时只有 D7～D0 有意义

④ D/A 转换器片选 (写操作)

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

此时只有 D7～D0 有意义,具体写操作意义由 A1/A0 决定，D/A 转换器地址选择位见①:A1/A0

a: 写 D/A（D/A1 或 D/A2）低 8 数据寄存器数据格式

D7	D6	D5	D4	D3	D2	D1	D0
B07	B06	B05	B04	B03	B02	B01	B00

此时只有 D7～D0 有意义,注意 B07-B00 在 D15-D00 的 D07-D00 上，D15-D08 无意义

b: 写 D/A（D/A1 或 D/A2）高 4 位数据寄存器数据格式

D7	D6	D5	D4	D3	D2	D1	D0
X	X	X	X	B11	B10	B09	B08

此时只有 D3～D0 有意义, 注意 B11-B08 在 D15-D00 的 D03-D00 上，D15-D04 无意义

B11～B00:D/A 转换 12Bit 数据,X:未用位

⑤ 启动 D/A 转换器 (写操作)

Offset=08H Offset:相对地址的偏移量，该写操作的 I/O 地址(详细操作见软件说明书)

此时 D15～D00 无意义，12BIT 的 DA 转换数据写入 D/A 转换器后，必须执行启动 D/A 转换⑤ 输出电压才会变化

⑥ 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，该读操作的 I/O 地址(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
Z	X	X	X	B11	B10	B9	B8	B7	B6	B5	B4	B3	B2	B1	B0

B11～B0 : A/D 转换数据

Z: A/D 转换状态位 Z=1, A/D 转换器正在进行转换

Z=0, 转换结束,可以读取 A/D 转换数据

⑦: 16 位 TTL 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，该写操作的 **I/O** 地址**(**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
015	014	013	012	011	010	09	08	07	06	05	04	03	02	01	00

015～000 : 16Bit 数字量输出**,**对应 **J2** 的 **D015～D000**

⑧: 16 位 **TTL** 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，该读操作的 **I/O** 地址**(**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
I15	I14	I13	I12	I11	I10	I9	I8	I7	I6	I5	I4	I3	I2	I1	I0

I15～I0 : 16Bit 数字量输入**, I15～I0** 对应 **DI15～DI0**

AMPCI-9110 读写 **I/O** 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、命令字寄存器，见（1）
02H	写操作	启动 A/D 转换，见（2）
04H	读操作/ 写操作	计数定时器 8254 片选，见（3）
06H	写操作	D/A 转换器片选　，见（4）
08H	写操作	启动 D/A 转换器，见（5）
0AH	读操作	查询 A/D 转换状态位+A/D 转换数据，见（6）
0CH	写操作	16 位 TTL 数据量输出寄存器，见（7）
0EH	读操作	16 位 TTL 数据量输入寄存器，见（8）

3.2 工作方式

3.2.1 A/D 部分转换触发方式

①软件触发方式:
执行 Offset=02H 写命令,触发 A/D 转换

②定时触发方式:
初始化 8254 的#1#2 计数器,CLK2 输入为 2MHZ,CLK1 输入为 OUT2,OUT1 输出的下降沿触发 A/D 转换,需短接 JP5

3.2.2 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

3.2.2 A/D 转换工作过程

AMPCI-9110 板的 A/D 转换是通过软件编程来控制和启动 A/D 转换的

<1>使用时首先应程序设定要进行 A/D 转换的通道号，即要对哪一通道进行 A/D 转换，该写操作见前面《3.1 寄存器功能描述》的(1)节:“模拟输入通道选择、增益选择、D/A 和 82C54 地址选择寄存器”，该寄存器 I/O 地址 Offset=00H ，“Offset”是相对地址的偏移量，可理解为该寄存器的 I/O 地址(详细对“Offset”的读写操作见软件说明书部分)

<2>程序设定完要进行 A/D 转换的通道号和放大倍数后，需执行程序启动 A/D 转换操作，A/D 转换器才开始转换，即再执行《3.1 寄存器功能描述》的（2）节，“启动 A/D 转换”，该写操作的 I/O 地址 Offset=02H，“Offset”是相对地址的偏移量，(详细对“Offset”的读写操作见软件说明书部分)，该操作为软件触发启动一个周期的 A/D 转换，若 A/D 转换的放大倍数不是 1 倍，应在设定“模拟输入通道选择、增益选择、D/A 和 82C54 地址选择寄存器”操作后，适当加几个μS 的延时以给程控放大器足够的建立时间，再执行程序启动 A/D 转换操作，以保证精度

<3>启动 A/D 转换后，应通过软件查询 A/D 转换完成状态标志位，见前面《3.1 寄存器功能描述》的（6）节，“查询 A/D 转换状态位+A/D 转换数据 ”，“Z” 该位为 1，说明 A/D 转换器正在进行转换，不可读取 A/D 转换数据，继续查询该位，若该位为 0，说明 A/D 转换已完成，可读取 A/D 转换数据，该读操作的 I/O 地址 Offset=0AH，“Offset” 相对地址的偏移量(详细操作见软件说明书)，标志位 Z 对应该标志寄存器数据线 D15-D00 的 D15

<4>当执行上述“<3>查询 A/D 转换完成状态标志位”且“Z”位为 0 时，即可读取 A/D 转换数据， 该读操作的 I/O 地址同样为 Offset=0AH，“Offset” 相对地址的偏移量(详细操作见软件说明书)

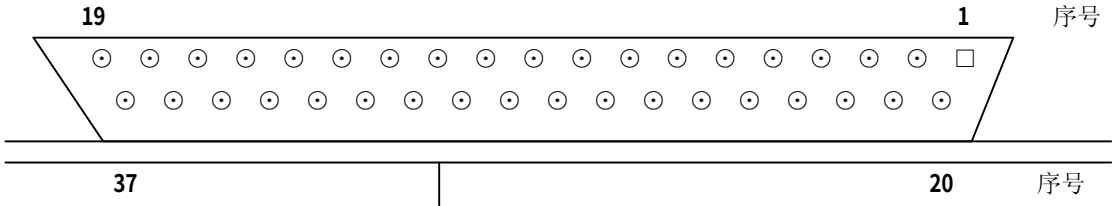
3.3 调整电位器定义

PR1: 程控放大器零偏移调整电位器
PR2: A/D 转换单极性零偏移调整电位器
PR3: A/D 转换双极性零偏移调整电位器

PR4: A/D 转换满量程调整电位器
PR5: D/A 基准源**+5V** 调整电位器
PR6: D/A2 零偏移调整电位器
PR7: D/A2 满量程调整电位器
PR8: D/A1 零偏移调整电位器
PR9: D/A1 满量程调整电位器

3.4 信号输入/输出插座定义**:**

a: J1 模拟输入/模拟输出定义**:**



37 芯孔式 **D** 型插头 **J1**

插头序号	1	2	3	4	5	6	7	8	9	10
引脚定义	CH00	CH01	CH02	CH03	CH04	CH05	CH06	CH07	CH08	CH09

插头序号	11	12	13	14	15	16	17	18	19	
引脚定义	CH0A	CH0B	CH0C	CH0D	CH0E	CH0F	GND	D/A1	D/A2	

插头序号	20	21	22	23	24	25	26	27	28	29
引脚定义	CH10	CH11	CH12	CH13	CH14	CH15	CH16	CH17	CH18	CH19

插头序号	30	31	32	33	34	35	36	37		
引脚定义	CH1A	CH1B	CH1C	CH1D	CH1E	CH1F	GND	GND		

CH00～CH1F: 对应模拟输入通道 **00～31** 共 **32** 个通道**,**对应通道代码 **00～1F**
D/A1,D/A2: 模拟输出通道 **1,2**

GND: 模拟输入/输出地

b:数字量和计数定时器#0 输出插座 J2 定义:

插座 **J2** 定义见下页图示其中:

DO0～DO15: TTL/COMS 兼容数字量输出

DI0～DI15: TTL/COMS 兼容数字量输入

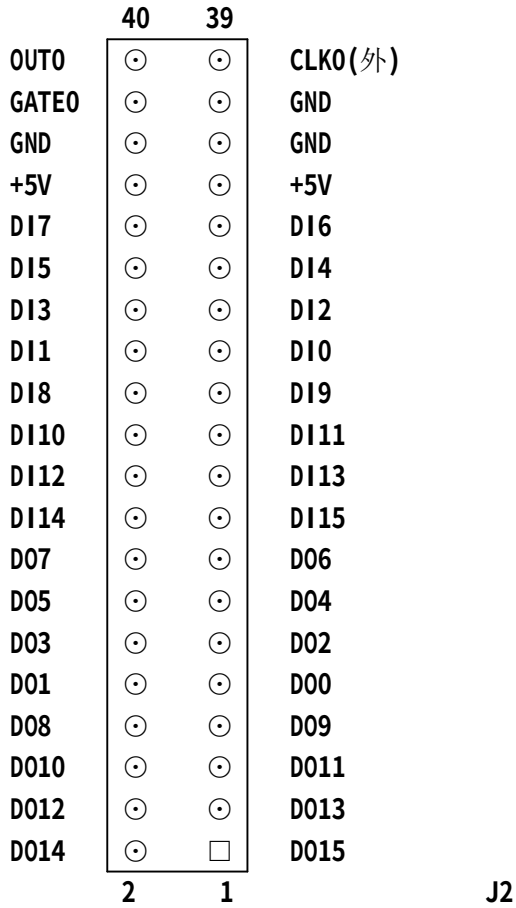
+5V: 机内 **PCI** 总线**+5V**

GND: 机内地

GATE0: 计数器 **0** 门控 **GATE**

OUT0: 计数器 **0** 输出

CLK0: 计数器 **0** 外部时钟输入



3.5 跨接线选择定义:

JP1: 1,2 短接,模拟输入选择单极性;**2,3** 短接,模拟输入选择双极性(出品状态)

JP2: 1,2 短接,**8254** 的 **CLK0** 接 **2MHZ** 时钟输入; **2,3** 短接,接外部输入(出品状态);

JP3: D/A2 输出选择: ①**1,2** 短接,**-5V**～**+5V**(出品状态)

②**2,3** 短接,**0**～**+10V**

③**3,4** 短接,**0**～**+5V**

JP4：D/A1 输出选择**：**跨接方式与 **D/A2** 相同
JP5：短接时定时器触发 **A/D** 转换允许；反之，禁止**(**出品状态**)**；

3．5 校准
① **A/D** 部分校准
a.程控放大器调整：软件设置放大倍数 **16** 倍，任一通道接地，调整 **PR1** 使
AD526 的 **8** 脚输出为 **0.0000V(**出品时已调整**)** （**AMPCI-9110A** 无此）
b.单极性输入 **0～+10V**：任一通道对地短接，调整 **PR2**，使转换值为 **001H**；
输入任一通道接 **7.5V**，调整 **PR4**，使转换值为 **C00H**；
c.双极性输入**-5V～+5V**：任一通道对地短接，调整 **PR3**，使转换值为 **800H**；
输入任一通道 **4.997V** 调整 **PR4** 使转换值为 **FFEH**；
② **D/A** 部分校准（**AMPCI-9110A** 无此）
基准源调整**：**调整 **PR5** 使 **336-5V** 输出**+5.000V,**出厂时已调整。
D/A1 校准：**1** 通道输出 **0V**，调整 **PR8**，测 **D/A1** 使 **D/A1** 输出为 **0.000V,**
1 通道输出 **5V**，调整 **PR9**，测 **D/A1** 使 **D/A1** 输出为正满度
D/A2 校准：**2** 通道输出 **0V**，调整 **PR6**，测 **D/A2** 使 **D/A2** 输出为 **0.000V,**
2 通道输出 **5V**，调整 **PR7**，测 **D/A2** 使 **D/A2** 输出为正满度

四、 产品成套性

① **AMPCI-9110** 成品板	壹块
② **37** 芯备件	壹套
③ **AMPCI-9110** 板说明书	壹本
④ **AMPCI-9110** 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，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，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

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**+**增益为 **1**
(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 **3456H,** 读数字量口输入
(1) 3456H 送 **Offset=0CH ;**写数字量输出口 **16** 位数据
(2) 读 **Offset=0EH ;**读数字量输入口 **16** 位数据

3.D/A1 输出 **0.000V(**设 **D/A1** 输出方式设置为**-5V - +5V)**

（**1**）**0000H** 送 **Offset=00H** ；设 **A1=0/A0=0**，**DA1** 的低 **8** 位数据 **B07-B00** 地址

（**2**）**0000H** 送 **Offset=06H** ；写入 **D/A1** 转换数据的低 **8** 位数据 **B07-B00**

（**3**）**0001H** 送 **Offset=00H** ；设 **A1=0/A0=1**，**DA1** 的高 **4** 位数据 **B11-B08** 地址

（**4**）**0008H** 送 **Offset=06H** ；写入 **D/A1** 转换数据的高 **4** 位数据 **B11-B08**

（**5**）**0000H** 送 **Offset=08H** ；启动 **D/A** 转换

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

一.WIN98 下测试程序说明

(1) 基本 **I/O** 口输入**/**输出测试**(VC** 调用静态库**,"PCI** 总线插卡**")**

在 **WIN98** 下按软件说明每一步安装完成后**,**运行程序**"PCI** 总线插卡**",**屏幕出现简单的读写演示画面**,**只插一块 **AMPCI** 插卡时板卡号选择不用设置**,**如进行写卡操作**,**先设置偏移地址**,**如要写 **AMPCI-9110** 卡的数字量口输出 **"ABCDH ",**则首先在右侧写地址栏填 **" OCH "(**即 **offset= OCH),**再在数值处填 **"ABCD",**当地址和数据均填写完成后**,**点击 **" 写 "**图标**,**到此完成一次写操作同样读卡操作与写操作类似**,**只需填写 **offset,** 如要读 **AMPCI-9110** 卡的数字量口输入状态**,**先在左侧读地址栏填 **" OEH "(**即 **offset=0EH),**再点击 **" 读 "**图标**,**到此完成一次读操作**,**读入数字量口输入状态，该程序只是对 **AMPCI** 卡进行读写的演示**,**用户请参照软件说明编写用户程序

(2) AMPCI-9110 卡测试程序**(VB** 调用 **DLL,"WIN_PCI_ALL)**

在 **WIN98** 下按软件说明每一步安装完成后**,**运行程序 **"WIN_PCI_ALL"**，点击“开始”图标每秒循环执行显示各通道转换值**,**该程序只是 **VB** 调用 **DLL** 进行读写的演示程序**,** 请参照该软件源程序说明编写用户程序

(3)光盘上 **WIN98** 目录内子目录 **"BC_PCI_DLL"**为 **C++ Builder** 调用 **DLL** 对 **AMPCI** 卡读写演示程序

二.WIN2000/XP 下测试程序说明

(1) WIN2000/XP 下子目录 **"sample"**

这个目录下的文件是为用户提供的演示源程序和读写测试可执行文件**,**它包括子目录 **VBTEST** 和 **VCTEST,** 用户可参照该软件源程序编写用户程序该演示程序是按插 **2** 块板考虑的**,**插多块板时请参考该程序**,**执行 **VBTEST** 或 **VCTEST** 写操作时先输入板号**(0,1),**如一块板板号处填 **"0",**再填写 **offset(**偏移地址**,00,02,04,….)**具体 **offset** 偏移地址操作意义见硬件部分

说明书，offset 处填写 2 位 16 进制数,还要填写要输出的数据(填写 4 位 16 进制数),如输出数据 ABCDH 则填 "ABCD",然后点击"确定"完成一次写操作

读操作与写操作类似,输入"板号"和" offset 然后点击"确定"完成一次读操作,右侧显示读到的数据,同样为 4 位 16 进制数

(2) WIN2000/XP 下子目录 "BC_PCI_DLL"

该目录 "BC_PCI_DLL"为 WIN2000/XP 下 C++ Builder 调用 DLL 对 AMPCI 卡读写演示程序

WIN98 系列 PCI 工业 I/O 卡软件使用说明书

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-9110 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 的 Windows 安装程序 VB 版；调用动态库,AMPCI-9110 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

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\vmm32 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\vmm32 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。

⑴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。

⑵运行光盘 win_pci_all 文件夹中的 setup.exe 文件，完成 VB 测试 9110 应用程序的安装。

⑶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。

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

⑴ 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。

⑵unsigned 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

⑶ 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)

⑷void 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

⑸int 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：

⑴ 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。

⑵ 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。

⑶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）。

⑷在源程序中包含头文件 plx9052_lib.h。其格式为：

```
#include “include\plx9052_lib.h”
```

⑸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。

⑹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

⑺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 pcidll.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("pcidll.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.Visal Basic 调用动态连接库

⑴ 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。

⑵ 对函数进行声明（这部分在 plx9052.bas 文件中）

```
Declare Function PLX9052_CountCards Lib "pcidll" Alias "#1" (ByVal dwVendorID As Long,
    ByVal dwDeviceID As Long) As Long
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
Declare Sub PLX9052_Close Lib "pcidll" Alias "#3" (ByVal hPLX9052 As Long)
Declare Function PLX9052_ReadWord Lib "pcidll" Alias "#5" (ByVal phPLX9052 As Long, ByVal
    addrSpace As Long, ByVal dwOffset As Long) As Integer
Declare Sub PLX9052_WriteWord Lib "pcidll" Alias "#9" (ByVal phPLX9052 As Long, ByVal
    addrSpace As Long, ByVal dwOffset As Long, ByVal data As Long)
Declare Function Hex_dec Lib "pcidll" 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&
```

⑶ 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 系列 PCI 工业 I/O 卡 VB 调用使用说明书

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

六、 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

⑴ **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

⑵ **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

⑶**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)

⑷**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

⑸**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：

⑴ 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。
⑵ 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

Visal Basic 调用动态连接库

⑴ 首先将 Module1 模块文件添加到你的工程文件中。

⑵ 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&
```

⑶ 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 系列 PCI 工业 I/O 卡 VC 调用使用说明书

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

⑴ 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。

⑵unsigned 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。

⑶void 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

⑷void 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

⑸int 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：

⑴ 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。

⑵ 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。

⑶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）。

⑷在源程序中包含头文件 ampci.h。其格式为：

#include “include\ampci.h”

⑸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。

七、 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。**