



PCI-P16R16 系列

软件使用手册

[简体中文 1.0 版, 2011 年 7 月]

支援



PCI-P16R16 系列板卡包含 PCI-P8R8、PCI-P16R16、PCI-P8POR8、PCI-P16POR16、PCI-P16C16、PEX-P8POR8i 及 PEX-P16POR16i。以上板卡都支持 P16R16.DLL 驱动程序。

承诺

郑重承诺：凡泓格科技股份有限公司产品从购买即日起一年内无任何材料性缺损。

免责声明

凡使用本系列产品除产品质量所造成的损害，泓格科技股份有限公司不承担任何法律责任。泓格科技股份有限公司有义务提供本系列产品可靠而详尽资料，但保留修订权利，且不承担使用者非法利用资料对第三方所造成侵害构成的法律责任。

版权

版权所有 © 1999 泓格科技股份有限公司，保留所有权力。

商标

手册中所涉及所有公司商标，商标名称及产品名称分别属于该商标或名称的拥有者所有。

许可

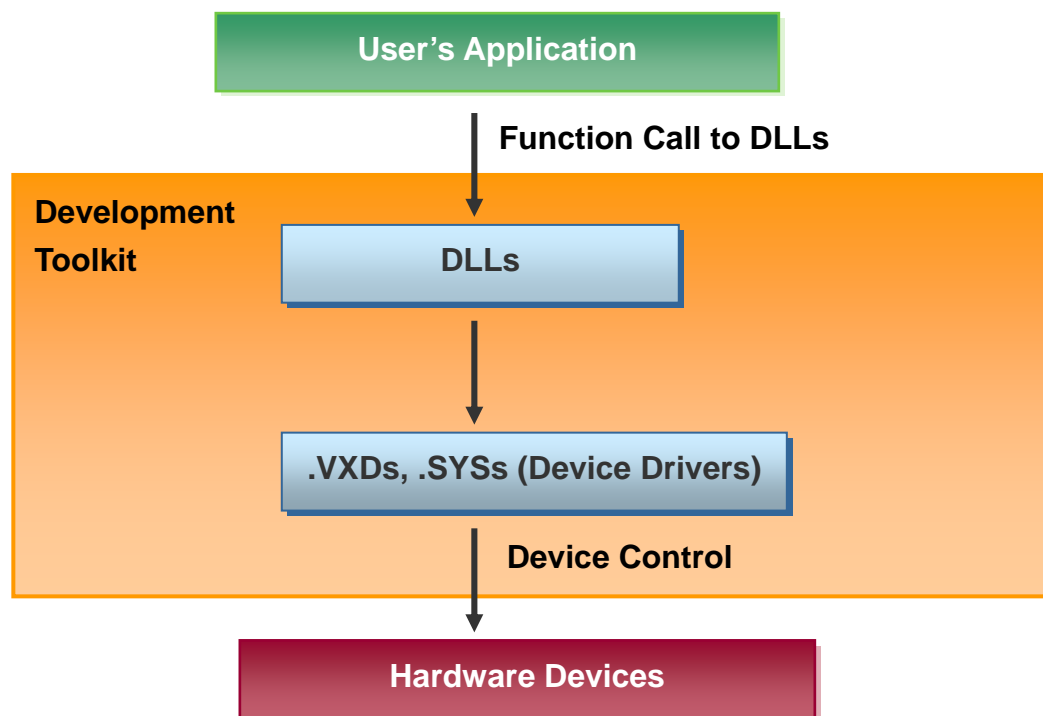
用户可以使用、修改和备份单片机上的软件部份，但无权复制，转移或发布该软件的全部或部份拷贝。

目录

1.	动态链接库功能说明	3
1.1	参考信息.....	4
1.2	档案文件说明	5
2.	函数明说.....	6
2.1	P16R16. H.....	8
2.2	测试函数.....	9
2.2.1	<i>PCI_FloatSub2</i>	9
2.2.2	<i>PCI_ShortSub2</i>	9
2.2.3	<i>PCI_GetDIIVersion</i>	10
2.3	DRIVER INITIALIZATION函数.....	10
2.3.1	<i>PCI_DriverInit</i>	10
2.3.2	<i>PCI_DriverClose</i>	12
2.3.3	<i>PCI_GetDriverVersion</i>	12
2.3.4	<i>PCI_GetCongfigAddressSpace</i>	13
2.4	PCI-P16R16 系列数字量输出输入函数	14
2.4.1	<i>P16R16_DO</i>	14
2.4.2	<i>P16R16_DI</i>	14
2.5	PCI-P8R8 系列数字量输出输入函数	17
2.5.1	<i>P8R8_DO</i>	17
2.5.2	<i>P8R8_DI</i>	18
2.6	问题回报.....	21

1. 动态链接库功能说明

在 Windows 95/98/ME/NT/2000 及 32-bit Windows XP/2003/Vista/7 系列中 P16R16.DLL 动态链接库包含了 PCI-P16R16 系列卡需要调用的函数。应用结构如下图。用户应用程序指定开发工具如 VB, VC, Delphi, Borland C++ Builder, VB.NET, VC.NET 及 C# 这些工具有能够去调用 P16R16.DLL。接下来 DLL 动态链接库将调用 P16R16.sys，然后 P16R16.sys 去访问硬件系统。



1.1 参考信息

可参考下列使用手册:

- **PCI-P16R16 PnP Driver Installation.pdf:**

描述如何在 Windows 95/98 安装即插即用 (Plug and Play) 驱动 PCI-P16R16 系列卡。

<http://ftp.icpdas.com/pub/cd/iocard/pci/napdos/pci/manual/>

- **Software Installation Guide.pdf:**

描述如何在 Windows 95/98/2000 及 32-bit Windows XP/2003/Vista/2008/7 安装软件包。

<http://ftp.icpdas.com/pub/cd/iocard/pci/napdos/pci/manual/>

- **Resource Checking .pdf:**

描述如何在 Windows 95/98/2000 及 32-bit Windows XP/2003/Vista/2008/7 检查板卡上的资源 I/O Port 地址, IRQ 号码及 DMA 号码。

<http://ftp.icpdas.com/pub/cd/iocard/pci/napdos/pci/manual/>

- **Calling DLL Functions.pdf:**

描述如何在 VC++6, VB6, Delphi4 及 Borland C++ Builder4 开发工具上使用 DLL 动态链接库。

<http://ftp.icpdas.com/pub/cd/iocard/pci/napdos/pci/manual/>

- **PCI-P8R8_P16R16_Series_Hardware_Manual.pdf:**

PCI-P16R16 系列卡硬件手册描述 PCI-P8R8/P16R16、PCI-P8POR8/P16POR16、PCI-P16C16 及 PEX-P8POR8i/P16POR16i 卡相关硬件使用说明。

<http://ftp.icpdas.com/pub/cd/iocard/pci/napdos/pci/pci-p16r16/manual/>

1.2 档案文件说明

安装驱动程序后，有关的演示程序，开发库，并声明头文件为不同的开发环境将可在下列地点。

如需详细的 PCI- P16R16 系列安装 Windows 驱动程序的信息，请参阅快速入门指南 (CD:\NAPDOS\PCI\PCI-P16R16\Manual\QuickStart\)

<http://ftp.icpdas.com/pub/cd/iocard/pci/napdos/pci/pci-p16r16/manual/quickstart/>

用于 Windows 的演示程序是包含在：

CD:\NAPDOS\PCI\PCI-P16R16\DLL_OCX\Demo\

http://ftp.icpdas.com/pub/cd/iocard/pci/napdos/pci/pci-p16r16/dll_ocx/demo/

- BCB 3 → for Borland C++ Builder 4
P16R16.H → Header files
P16R16.LIB → Linkage library for BCB only

- Delphi3 → for Delphi 3
P16R16.PAS → Declaration files

- VB6 → for Visual Basic 6
P16R16.BAS → Declaration files

- VC6 → for Visual C++ 6
P16R16.H → Header files
P16R16.LIB → Linkage library for VC6 only

- VB.NET2005 → for VB.NET2005
P16R16.vb → Visual Basic Source files

- CSharp2005 → for C#.NET2005
P16R16.cs → Visual C# Source files

2. 函数明说



下列函数定义表提供了 PCI-P16R16 系列函数更多的详细信息，下面部分将介绍每个函数。然而为了让语法描述简单清楚，属性为输入/输出的参数表现为 **[Input]/[Output]**，请明如下表：

关键词	在调用函数之前用户是否需要设定参数？	在调用这个函数之后是否可以得到来自这个参数的数据或值？
[Input]	Yes	No
[Output]	No	Yes
[Input, Output]	Yes	Yes

注：所有参数都需要由用户分配的空间。

功能 P16R16.DLL（Windows 95/98/NT 下动态链接库）的原形为 P16R16H.LIB（基于 DOS 系统的巨型模式）和 P16R16L.LIB（基于 DOS 系统的大型模式），这样可以方便的在不同的平台下应用。

■ DLL 返回的代码定义如下：

// 返回代码

Error Code	Error ID
0	NoError
1	DriverHandleError
2	DriverCallError
3	NotFoundBoard
4	FindBoardError
5	ExceedBoardNumber

■ DLL 函数如下:

章节	函数描述
2.2	Test 测试函数
2.2.1	float PCI_FloatSub2(float fA, float fB);
2.2.2	short PCI_ShortSub2(short nA, short nB);
2.2.3	DWORD PCI_GetDIIVersion(void);
2.3	Driver Initialization 函数
2.3.1	WORD PCI_DriverInit(WORD *wBoards);
2.3.2	WORD PCI_DriverClose(void);
2.3.3	WORD PCI_GetDriverVersion(WORD *wDriverVersion);
2.3.4	WORD PCI_GetConfigAddressSpace(WORD wBoardNo, WORD *TypeID, WORD *wAddress0, WORD *wAddress1, WORD *wAddress2, WORD *wAddress3, WORD *wAddress4, WORD *wAddress5);
2.4	PCI-P16R16 系列数字量输出输入函数
2.4.1	void CALLBACK P16R16_DO(WORD BoardAddr, WORD OutData);
2.4.2	WORD CALLBACK P16R16_DI(WORD BoardAddr);
2.5	PCI-P8R8 系列数字量输出输入 函数
2.5.1	void CALLBACK P8R8_DO(WORD BoardAddr, WORD OutData);
2.5.2	BYTE CALLBACK P8R8_DI(WORD BoardAddr);

2.1 P16R16.H

```
#define EXPORTS extern "C" __declspec (dllimport)

// return code
#define NoError 0
#define DriverHandleError 1
#define DriverCallError 2
#define NotFoundBoard 3
#define FindBoardError 4
#define ExceedBoardNumber 5

// define Type Name ID
#define TYPE_P16R16 0
#define TYPE_P8R8 1
#define TYPE_TMC12 2
#define TYPE_DA16 3
#define TYPE_DA8 4

EXPORTS float CALLBACK PCI_FloatSub2(float fA, float fB);
EXPORTS short CALLBACK PCI_ShortSub2(short nA, short nB);
EXPORTS WORD CALLBACK PCI_GetDIIVersion(void);

EXPORTS WORD CALLBACK PCI_DriverInit(WORD *wTotalBoards);
EXPORTS void CALLBACK PCI_DriverClose(void);
EXPORTS WORD CALLBACK PCI_GetDriverVersion(WORD *wVxdVersion);

EXPORTS WORD CALLBACK PCI_GetConfigAddressSpace
(WORD wBoardNo, WORD *TypeID,
WORD *wAddress0, WORD *wAddress1, WORD *wAddress2,
WORD *wAddress3, WORD *wAddress4, WORD *wAddress5);

EXPORTS WORD CALLBACK PCI_WhichBoardActive(void);
EXPORTS void CALLBACK P16R16_DO(WORD BaseAddr, WORD OutData);
EXPORTS WORD CALLBACK P16R16_DI(WORD BaseAddr);
EXPORTS void CALLBACK P8R8_DO(WORD BaseAddr, WORD OutData);
EXPORTS BYTE CALLBACK P8R8_DI(WORD BaseAddr);
```

2.2 测试函数

2.2.1 PCI_FloatSub2

- **说明:**
浮点数据类型的减法运算（诸如：A-B），并提供动 DLL 的连接测试。
- **语法:**
float **PCI_FloatSub2**(float **fA**, float **fB**)
- **参数:**

fA	[Input]	4 位浮点类型值
fB	[Input]	4 位浮点类型值
- **返回值:**
fA – fB 之值

2.2.2 PCI_ShortSub2

- **说明:**
短整数类型的减法运算（诸如：A-B），并提供动 DLL 的连接测试。
- **语法:**
short **PCI_ShortSub2**(short **nA**, short **nB**)
- **参数:**

nA	[Input]	2 位短整型数据值
nB	[Input]	2 位短整型数据值
- **返回值:**
nA – nB 之值

2.2.3 PCI_GetDllVersion

- **说明:**
获得 P16R16.DLL 版本号。
- **语法:**
WORD **PCI_GetDllVersion**(void)
- **参数:**
空
- **返回值:**
201(16 进制)即版本号为 2.01

2.3 Driver Initialization函数

2.3.1 PCI_DriverInit

- **说明:**
初始化设备驱动（napwnt.sys 适用于 Window Windows NT/2K/XP/Vista/7；nappci.vxd 适用于 Windows 95/98）。该函数为程序初始化过程中必要步骤。

- **语法:**
WORD **PCI_Driverinit** (WORD *wTotalBoard)

- **参数:**

wTotalBoard	[Input]	Address of wTotalBoard. When wTotalBoard = 1: 找到 1 张 P16R16 或 P8R8 板卡。 When wTotalBoard = 2: 有可能的组合为 → <ul style="list-style-type: none"> 找到 1 张 P16R16 板卡和 1 张 P8R8 板卡。* 找到 2 张 P16R16 板卡。* 找到 2 张 P8R8 板卡。
---------------------	---------	---

■ 返回值:

NoError	OK
NotFoundBoard	未发现板卡
DriverCallError	不能打开 nappci.vxd(Windows 95/98 系统); 不能打开 napwnt.sys(NT/2K/XP/Vista/7 系统)。
DriverHandleError	当设备驱动打开时, 返回错误句柄

■ 示例程序:

[VC example]

```
LRESULT CALLBACK WndProc(HWND hwnd, UINT iMsg, WPARAM wParam,
    LPARAM lParam, LPARAM lParam)
```

```
{
    static char cBuf[80];
    HDC        hdc;
    TEXTMETRIC tm;
    PAINTSTRUCT ps;
    int        i;
    switch (iMsg)
    {
    case WM_CREATE : // window initial

        /*****
        /* NOTICE: call PCI_DriverInit() to initialize the driver. */
        *****/

        /*****
        // Initial the device driver, and return the board number in the PC
        *****/

        wInitialCode=PCI_DriverInit(&wTotalBoard);

        if( wInitialCode!=NoError )
        {
            MessageBox(hwnd,"No PCI card in this system !!!","PCI Card
            Error",MB_OK);
        }
        :
        :
        :
    }
}
```

2.3.2 PCI_DriverClose

- **说明:**
终止设备驱动（napwnt.sys 适用于 Window NT/2K/XP/Vista/7；nappci.vxd 适用于 Windows 95/98）。在 DOS 中，该函数仅提供终止 Win32 程序并释放内存，且仅返回 NoError。
- **语法:**
void **PCI_DriverClose**(void)
- **参数:**
空
- **返回值:**
空

2.3.3 PCI_GetDriverVersion

- **说明:**
获得设备驱动版本号（napwnt.sys 适用于 Window NT/2K/XP/Vista/7；nappci.vxd 适用于 Windows 95/98）。
- **语法:**
WORD **PCI_GetDriverVersion**(WORD *wDriverVersion)

- **参数:**

*wDriverVersion	[Input]	Address of wDriverVersion. WDriverVersion = 200[hex] → Version 2.00
------------------------	---------	--

- **返回值:**

NoError	OK
NotFoundBoard	未发现板卡
DriverCallError	不能打开 nappci.vxd(Windows 95/98 系统); 不能打开 napwnt.sys(NT/2K/XP/Vista/7 系统)。
DriverHandleError	当设备驱动打开时，返回错误句柄

2.3.4 PCI_GetCongfigAddressSpace

- **说明:**

读取板卡 P8R8/P16R16 配置地址空间，并获得 Base Address0、Base Address1、Base Address2、Base Address3、Base Address4 和 Base Address5 容量。

- **语法:**

WORD **PCI_GetCongfigAddressSpace** (WORD **wBoardNo**, WORD ***wTypeID**, WORD ***wAddress0**, WORD ***wAddress1**, WORD ***wAddress2**, WORD ***wAddress3**, WORD ***wAddress4**, WORD ***wAddress5**)

- **参数:**

wBoardNo	[Input]	板卡 P16R16/P8R8 序号(从 0 开始)。
*wTypeID	[Input]	Address of wType: 0: 该板卡为 PCI_P16R16 1: 该板卡为 PCI_P8R8 2: 该板卡为 PCI_TMC12 3: 该板卡为 PIO_DA16 4: 该板卡为 PIO_DA8
*wAddress0	[Output]	六个 PCI 设备的基地址将储存在这些参数中。
*wAddress1		
*wAddress2		
*wAddress3		
*wAddress4		
*wAddress5		

- **返回值:**

NoError	OK
FindBoardError	未发现板卡
ExceedBoardError	输入的板卡序号超过 PC 现存的板卡数。合法的板卡序号不能超过(现存板卡数 -1)。

2.4 PCI-P16R16 系列数字量输出入函数

PCI-P16R16 系列卡包含 PCI-P16R16、PCI-P16C16、PCI-P16POR16 及 PEX-P16POR16i。

2.4.1 P16R16_DO

- **说明:**

从板卡的 P16R16D/O 端口发送 16 位数据。

- **语法:**

Void **P16R16_DO** (WORD **BaseAddr**, WORD **OutData**)

- **参数:**

BaseAddr	[Input]	D/O 端口基地址
OutData	[Input]	发送到 D/O 端口的 16 位数据

- **返回值:**

空

- **示例程序:**

请参考 16 页。

2.4.2 P16R16_DI

- **说明:**

从板卡 P16R16 D/I 端口读取 16 位数据。

- **语法:**

WORD **P16R16_DI** (WORD **BaseAddr**)

- **参数:**

BaseAddr	[Input]	D/O 端口基地址
-----------------	---------	-----------

- **返回值:**

从 D/I 端口读取的 16 位数据。

■ 示例程序:

```
/*
*****
/* This program is developed by Turbo C 2.0 */
*****
/* Demo 1: One P16R16 card demo. */
*****

#include "P16R16.H"
int main()
{
    int i,j;
    WORD nVal;
    float fVal;
    WORD wBoards,wRetVal,wVal;
    WORD wInData;
    WORD wTypeID;
    WORD wAddress0,wAddress1,wAddress2;
    WORD wAddress3,wAddress4,wAddress5;
    WORD P16R16_BaseAddress,P8R8_BaseAddress;
    WORD wP16R16No,wP8R8No;

    clrscr();

    /*initiating PCI-P16R16 card and detect how many P16R16/P8R8 card in PC */
    wRetVal=PCI_DriverInit(&wBoards);
    printf("Threr are %d PCI-P16R16/P8R8 Cards in this PC,
tally.\n",wBoards);

    if( wBoards==0 )
    {
        putchar(0x07); putchar(0x07); putchar(0x07);
        printf("There are no P16R16/P8R8 card in this PC !!!\n");
        exit(0);
    }

    /* dump every P16R16/P8R8 card's configuration address space */
    for(i=0; i<wBoards; i++)
    {
        wRetVal=PCI_GetConfigAddressSpace(i, &wTypeID,
&wAddress0,&wAddress1,&wAddress2,
&wAddress3,&wAddress4,&wAddress5);

        if( !wRetVal )
        {
            switch( wTypeID )
            {
                case 0: printf("==> %02d Board Name:PCI-P16R16\n",i);
P16R16_BaseAddress=wAddress2;
wP16R16No++;
break;
            }
        }
    }
}
```

```

case 1: printf("==> %02d Board Name:PCI-P8R8\n",i);
        P8R8_BaseAddress=wAddress2;
        wP8R8No++;
        break;
case 2: printf("==> %02d Board Name:PCI-TMC12\n",i);
        break;
case 3: printf("==> %02d Board Name:PCI-DA16\n",i);
        break;
case 4: printf("==> %02d Board Name:PCI-DA8\n",i);
        break;
    }
printf(" --> Addr0:%04x | Addr1:%04x | Addr2:%0x\n",
        wAddress0,wAddress1,wAddress2);
printf(" --> Addr3:%04x | Addr4:%04x | Addr5:%0x\n\n",
        wAddress3,wAddress4,wAddress5);
}
}

/* Getting the Driver version */
wRetVal=PCI_GetDriverVersion(&wVal);
printf("Driver Version=%0x\n",wVal);

/* call a function to test if exact calling LIB */
nVal=PCI_ShortSub2(1,2);
printf("PCI_ShortSub2(1,2) = %d\n",nVal);

/* call another function to test if exact calling LIB */
fVal=PCI_FloatSub2(1.0,2.0);
printf("PCI_FloatSub2(1.0,2.0) = %f\n",fVal);

if( wP16R16No<1 )
{
    putch(0x07);
    printf("Please plug one PCI-P16R16 in PC !!!\n");
    exit(0);
}

/*****
/***** PCI-P16R16 DO/DO demo *****/
*****/

printf("The PCI-P16R16 DO/DI testing !!!\n");
P16R16_DO(P16R16_BaseAddress,0x0000); /* Digital output */
delay(500); /* Delay a little time 500ms */
wInData=P16R16_DI(P16R16_BaseAddress); /* Digital input */
printf("Digital Output -> 0000H | Digital Input -> %04xH\n",wInData);

```

```

P16R16_DO(P16R16_BaseAddress,0xFFFF); /* Digital output */
delay(500); /* Delay a little time 500ms */

wInData=P16R16_DI(P16R16_BaseAddress); /* Digital input */
printf("Digital Output -> FFFFH | Digital Input -> %04xH\n",wInData);

P16R16_DO(P16R16_BaseAddress,0x5555); /* Digital output */
delay(500); /* Delay a little time 500ms */

wInData=P16R16_DI(P16R16_BaseAddress); /* Digital input */
printf("Digital Output -> 5555H | Digital Input -> %04xH\n",wInData);

P16R16_DO(P16R16_BaseAddress,0xAAAA); /* Digital output */
delay(500); /* Delay a little time 500ms */

wInData=P16R16_DI(P16R16_BaseAddress); /* Digital input */
printf("Digital Output -> AAAAH | Digital Input -> %04xH\n",wInData);

PCI_DriverClose();
return 0;
}

```

2.5 PCI-P8R8 系列数字量输出入函数

PCI-P8R8 系列卡包含 PCI-P8R8、PCI-P8POR8 及 PEX-P8POR8i。

2.5.1 P8R8_DO

- **说明:**
从板卡的 P8R8D/O 端口发送 8 位数据。
- **语法:**
Void **P8R8_DO** (WORD BaseAddr, WORD OutData)
- **参数:**

BaseAddr	[Input]	D/O 端口基地址
OutData	[Input]	发送到 D/O 端口的 8 位数据
- **返回值:**
空
- **示例程序:**
请参考 19 页。

2.5.2 P8R8_DI

- **说明:**

从板卡 P8R8 D/I 端口读取 8 位数。

- **语法:**

Void **P8R8_DO** (WORD **BaseAddr**, WORD **OutData**)

- **参数:**

BaseAddr	[Input]	D/O 端口基地址
-----------------	---------	-----------

- **返回值:**

空

- **示例程序:**

```
/*
*****
/* This program is developed by Turbo C 2.0 */
*****
/* Demo 2: One P8R8 card demo. */
*****
#include "P16R16.H"
int main()
{
    int i,j;
    WORD nVal;
    float fVal;
    WORD wBoards,wRetVal,wVal;
    WORD wInData;
    WORD wTypeID;
    WORD wAddress0,wAddress1,wAddress2;
    WORD wAddress3,wAddress4,wAddress5;
    WORD P16R16_BaseAddress,P8R8_BaseAddress;
    WORD wP16R16No,wP8R8No;

    clrscr();

    /* initiaing PCI-P16R16 card and detect how many P16R16/P8R8 card in PC */
    wRetVal=PCI_DriverInit(&wBoards);
    printf("Threr are %d PCI-P8R8/P16R16 Cards in this PC\n",wBoards);
}
```

```

if( wBoards==0 )
{
    putchar(0x07); putchar(0x07); putchar(0x07);
    printf("There are no P8R8/P16R16 card in this PC !!!\n");
    exit(0);
}

/* dump every P16R16/P8R8 card's configuration address space */
for(i=0; i<wBoards; i++)
{
    wRetVal=PCI_GetConfigAddressSpace(i,&wTypeID,
        &wAddress0,&wAddress1,&wAddress2,
        &wAddress3,&wAddress4,&wAddress5);
    if( !wRetVal )
    {
        switch( wTypeID )
        {
            case 0: printf("==> %02d Board Name:PCI-P16R16\n",i);
                P16R16_BaseAddress=wAddress2;
                wP16R16No++;
                break;
            case 1: printf("==> %02d Board Name:PCI-P8R8\n",i);
                P8R8_BaseAddress=wAddress2;
                wP8R8No++;
                break;
            case 2: printf("==> %02d Board Name:PCI-TMC12\n",i);
                break;
            case 3: printf("==> %02d Board Name:PCI-DA16\n",i);
                break;
            case 4: printf("==> %02d Board Name:PCI-DA8\n",i);
                break;
        }
        printf(" --> Addr0:%04x | Addr1:%04x | Addr2:%0x\n",
            wAddress0,wAddress1,wAddress2);
        printf(" --> Addr3:%04x | Addr4:%04x | Addr5:%0x\n\n",
            wAddress3,wAddress4,wAddress5);
    }
}

/* Getting the Driver version */
wRetVal=PCI_GetDriverVersion(&wVal);
printf("Driver Version=%x\n",wVal);

/* call a function to test if exact calling LIB */
nVal=PCI_ShortSub2(1,2);
printf("PCI_ShortSub2(1,2) = %d\n",nVal);

```

```

/* call another function to test if exact calling LIB */
fVal=PCI_FloatSub2(1.0,2.0);
printf("PCI_FloatSub2(1.0,2.0) = %f\n",fVal);

if( wP8R8No<1 )
{
    putchar(0x07);
    printf("Please plug one PCI-P8R8 in PC !!!\n");
    exit(0);
}

/*****
/*****  PCI-P8R8 DO/DO demo  *****/
/*****

printf("The PCI-P8R8 DO/DI testing !!!\n");
P8R8_DO(P8R8_BaseAddress,0x0000);    /* Digital output */
delay(500);                          /* Delay a little time */

wInData=P8R8_DI(P8R8_BaseAddress);    /* Digital input */
printf("Digital Output -> 0000H | Digital Input -> %04xH\n",wInData);

P8R8_DO(P8R8_BaseAddress,0xFFFF);    /* Digital output */
delay(500);                          /* Delay a little time */

wInData=P8R8_DI(P8R8_BaseAddress);    /* Digital input */
printf("Digital Output -> FFFFH | Digital Input -> %04xH\n",wInData);

P8R8_DO(P8R8_BaseAddress,0x5555);    /* Digital output */
delay(500);                          /* Delay a little time */

wInData=P8R8_DI(P8R8_BaseAddress);    /* Digital input */
printf("Digital Output -> 5555H | Digital Input -> %04xH\n",wInData);

P8R8_DO(P8R8_BaseAddress,0xAAAA);    /* Digital output */
delay(500);                          /* Delay a little time */

wInData=P8R8_DI(P8R8_BaseAddress);    /* Digital input */
printf("Digital Output -> AAAAH | Digital Input -> %04xH\n",wInData);

PCI_DriverClose();
return 0;
}

```

2.6 问题回报

当您所使用的程序发生问题或对程序有任何疑问，欢迎您来电或写信告知我们 (E-mail: Service.icpdas@gmail.com、Service@icpads.com)，我们将为您提供完善的咨询服务。

告知我们错误问题，包括以下信息：

1. 问题是可重现吗？如果是这样，怎么样？
2. 您使用什么样的平台和版本呢？例如，Windows98，Windows 2000 或 32 位 Windows XP/2003/Vista/2008/7。
3. 您使用我们什么产品种类？请参阅产品手册。
4. 如有错误信息对话框显示，请将剪下此画面包括完整的测试画面、标题栏文字。
5. 您使用的开发程序或使用的硬件设备或执行示例程序版本...等。
6. 欢迎提出您对于这个问题的建议的其它意见。

当我们收到您的问题及意见后，我们会约需两个工作日来测试您说的问题。然后尽快给您答复。请保持与我们联系。



E-mail: Service@icpads.com
Service.icpdas@gmail.com

Web Site: <http://www.icpdas.com>
<http://www.icpdas.com.tw>