





# PCI-P16R16 系列 DLL 软件使用手册

1.3版, 2015年10月

## <u>支援</u>

模块包含 PCI-P8R8, PCI-P8R8U, PCI-P16R16, PCI-P16R16U, PCI-P16C16, PCI-P16C16U, PCI-P16POR16, PCI-P16POR16U, PEX-P8POR8i 及 PEX-P16POR16i。

#### <u>承诺</u>

郑重承诺:凡泓格科技股份有限公司产品从购买后,开始享有一年保固,除人为使用 不当的因素除外。

## <u>责任声明</u>

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的法 律责任。泓格科技股份有限公司有义务提供本系列产品详细使用资料,本使用手册所 提及的产品规格或相关信息, 泓格科技保留所有修订之权利,本使用手册所提及之产 品规格或相关信息有任何修改或变更时, 恕不另行通知,本产品不承担用户非法利用 资料对第三方所造成侵害构成的法律责任,未事先经由泓格科技书面允许,不得以任 何形式复制、修改、转载、传送或出版使用手册内容。

#### <u>版权</u>

版权所有 © 2015 泓格科技股份有限公司, 保留所有权利。

#### <u>商标</u>

文件中所涉及所有公司的商标,商标名称及产品名称分别属于该商标或名称的拥有者 所持有。

# <u>联系我们</u>

如有任何问题欢迎联系我们,我们将会为您提供完善的咨询服务。 Email: <u>service@icpdas.com</u>; <u>service.icpdas@gmail.com</u>

1.	简介		2
	1.1	参考信息	3
2.	开始	安装使用	4
	2.1	取得 PCI−P16R16 驱动函式裤	4
	2.2	安装 PCI-P16R16 驱动程序函式裤	5
	23		7
	2.4	移除 PCI-P16R16 驱动承式库	
-			
3.	DLL	功态链接函数件	11
	3.1	错误码列表	
	3.2	测试函数	
	PCI_I	FloatSub2	
	PCI_S	ShortSub2	
	PCI_0	GetDIIVersion	
	3. 3	驱动函式集	
	PCI_I	DriverInit	
	PCI_I	DriverClose	
	PCI_0	GetDriverVersion	
	PCI_0	GetConfigAddressSpace	
	3.4	PCI-P16R16 系列数字量输出入函数	21
	P16R	R16_DO	
	P16R	R16_DI	
	3.5	PCⅠ-P8R8 系列数字量输出入函数	
	P8R8	3_DO	
	P8R8	3_DI	
۵	云列	程序	21
4.	1121	ייייייייייייייייייייייייייייייייייייי	
	4.1	WINDOWS 操作系统	
	4.2	DOS 操作系统	
5.	问题	回报	34

# 1. 简介

PCI-P16R16 系列卡提供了可调用的 P16R16.DLL 动态链接函数库,且能够在 Linux、Windows 98/NT/2000、32-Bit Windows XP/2003/Vista/2008/7/8 等操作系统环境下使用。

P16R16.DLL 动态链接函数库使开发更加容易及简单易懂的各种语言范例程序,如 Turbo C++、Borland C++、Microsoft C++、Visual C++、Borland Delphi、Borland C++ Builder、Visual Basic、Visual C#.NET、Visual Basic.NET...等,让用户能够快速的上手来使用。应用结构如下图。



# 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 & Play) 驱动 PCI-P16R16 系列卡。 http://ftp.icpdas.com/pub/cd/iocard/pci/napdos/pci/pci-p16r16/manual/
- PCI-P16R16\_DLL\_OCX\_Installation.pdf: 描述如何在 Windows 95/98 及 Windows NT 安装软件包。 <u>http://ftp.icpdas.com/pub/cd/iocard/pci/napdos/pci/pci-p16r16/manual/</u>
- PCI-P8R8\_P16R16\_Series\_Hardware\_Manual.pdf: PCI-P16R16 系列卡硬件手册描述 PCI-P8R8/P16R16、PCI-P16POR16、 PCI-P16C16 及 PEX-P8POR8i/P16POR16i 卡相关硬件使用说明。 <u>http://ftp.icpdas.com/pub/cd/iocard/pci/napdos/pci/pci-p16r16/manual/</u>

#### 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/

# 2. 开始安装使用

本章节将详细介绍如何取得驱动安装执行檔、驱动安装程序以及移除板卡…等。

# 2.1 取得 PCI-P16R16 驱动函式裤

P16R16.DLL 驱 动 函 式 库 能 够 在 Linux 、 Windows 98/NT/2000 、 32-Bit Windows XP/2003/Vista/2008/7/8 等操作系统环境下使用,且支持即插即用驱动安装,使安装过程便利又快速。

用户能够从随机出货的配件软件 CD 光盘或从泓格的软件下载网站中来取得 P16R16.DLL 驱动 函式库。详细取得/下载位置如下:



#### 请选择适合的操作系统来安装 PCI-P16R16 驱动函式库:

驱动程序名称	文件夹	操作系统
	Win98	支援 Windows 95 及 Windows 98
Pci-p16r16_Winxxxx_	WinNT	支援 Windows NT
xxx.exe	Win2K_XP_7	支援 Windows 2000, 32-bit Windows XP, 32-bit Windows 2003, 32-bit Windows Vista , 32-bit Windows 7及 32-bit Windows 8.

# 2.2 安装 PCI-P16R16 驱动程序函式裤

请先将 PCI-P16R16 系列卡安装至您的计算机,然后在执行 PCI-P16R16 驱动程序安装。详细 PCI-P16R16 系列卡硬件安装,可参考至硬件使用手册。

PCI-P16R16 系列卡硬件使用手册下载位置如下:

CD:\NAPDOS\PCI\PCI-P16R16 \Manual\

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

在 Windows XP 操作系统下执行安装,请依照下列步骤:

注意:如果您的操作系统为 Windows 95/98 及 Windows NT,请参考至 "PCI-P16R16\_DLL\_OCX\_Installation.pdf"说明文件来执行驱动程序安装。



步骤 2: 按 "Next>"

一个画面。

步骤 1: 双击 **"PCI-P16R16\_Win2k\_xxxx.exe"** 安装驱动函式库。



步骤 3: 选择安装目录, 默认为 C:\DAQPro\PCI-P16R16\_Win2000, 确认后按 "<u>N</u>ext>" 按钮 到下一个画面。

PCI-P16R16 DLL/OCX Win2000 Setup	×
Choose Destination Location Select folder where setup will install files.	
Setup will install PCI-P16R16_Win2000 in the following folder.	
To install to this folder, click Next. To install to a different folder, click Browse and select another folder.	
Destination Folder	
C:\DAQPro\PCI-P16R16_Win2000	
Instalishield Cancel	

步骤 5:选择 "Yes, restart computer now" 后,按下 "<u>F</u>inish" 按钮,系统会自动 重新启动, 在重新启动之后, 泓格 PCI-P16R16 驱动函式库安装完成。



# 2.3 即插即用驱动安装

注意:如果您的操作系统为 Windows 95 及 Windows 98,请参考至 "PCI-P16R16 PnP Driver Installation.pdf"说明文件来执行即插即用驱动安装。

步骤 1: 操作系统将找到新硬件, 然后将续继完成即插即用安装步骤。

注意: 有些作系统 (如, Windows Vista/7) 会找 到新硬件后,将自动完成即插即用驱动安装, 因此将会跳过步骤2 到步骤4。



步骤 2: 选择"自动安装软件 (建议选项)(I)"后,按下"下一步(N)>"按钮到下个页面。



步骤 3: 按下"完成"按钮,来完成安装。



步骤 4: 将显示"找到新硬件"讯息,板卡已成功完成安装。



使用手册, 1.3版本, 2015年 10月, 第8页

# 2.4 移除 PCI-P16R16 驱动函式库

泓格驱动函式库包括反安装工具来协助您从计算机上移除软件,如果您想要移除软件请完成下列 的流程来执行反安装工具。

步骤1: 至设定>>控制面板>>新增或移除程序下。

步骤2: 在选单列表上选择 PCI-P16R16 DLL/OCX Win2000 项目,并点击最右方的移除按键。

🖥 新碧或移除程式				
夏夏 夏夏 夏 夏 夏 夏 夏 日 夏 夏 夏 夏 日 夏 夏 日 夏	目前安裝的程式: 159 Network Film Monitor for Williagews 2000 看 OpenOffice.org 3.2	□ 顯示更新①)	排序方式②: 名稱 へ小 大小	0.40MB 338.00MB
新增程式图	PCI-P16R16 DLL/OCX ₩in2000 按這裡取得支援資訊。 要變更或從電腦移除這個程式,請按 [變更/移除]。		大小 已使用 上次使用在	<u>0.18MB</u> 很少 2014/9/23 變更 <i>雨</i> 8除
新増移除 Windows 元件( <u>A</u> )	<ul> <li>Realtek Ethernet Diagnostic Utility</li> <li>REAL TEK GbE &amp; FE Ethernet PCI-E NIC Driver</li> <li>Realtek High Definition Audio Driver</li> </ul>		大 <mark>大</mark> 大小 大小	2.73MB 3.43MB 70.20MB
会 設定程式 存取及預 設値(0)	<ul> <li>SketchUp 2014</li> <li>Sweet Home 3D version 4.3</li> <li>TcpClientServer 1.0.5</li> <li>Ulead PhotoImpact X3</li> </ul>		大小 大小 大小 大小	209.00MB 102.00MB 1.21MB 829.00MB
	<ul> <li>✓ VxComm Driver v2.12.09 (for 32/64-bit Windows 2K/XP/2k3/2k8/2012/Vista/7/8)</li> <li>✓ Windows Internet Explorer 8</li> <li>⊮ Windows XP Service Pack 3</li> <li>₩inRAR 壓縮工具</li> </ul>		大小 大小 大小	7.74MB 4.32MB 3.42MB

# **步骤 3:** 按下"是(Y)"按钮,来开始移除软件的动作。

PCI-P16R16 Win2000.XP Setup			
Preparing Setup Please wait while the InstallShield Wizard prepares the setup.			
PCI-P16R16 Win2000.XP Setup is preparing the InstallShield Wizard, which will guide you through the rest of the setup process. Please wait.			
PCI-P16R16 Win2000.XP			
This will remove PCI-P16R16 Win2000.XP from your computer. Do you wish to proceed?			
InstallShieldCancel			

步骤 4: 再按下"确定"按钮,确认已成功完成移除。

InstallShield Wizard	×
Setup Status	
PCI-P16R16 Win2000.XP Setup is performing the requested operations.	
PCI-P16R16 Win2000.XP In the second successfully uninstalled from your machine.	
Install5hield Cancel	

# 3. DLL 动态链接函数库明说

下列函数定义表提供了 PCI-P16R16 系列函数更详细信息,表 3-1 至表 3-5 部分将介绍每个函数。在使用 DLL 动态链接函数库前请注意下表关键词。以方便您的阅读:

▶ 表 3-1: 关键词

关键词	呼叫函式前需由使用者设定该参数	使用者呼叫函式后,会回传参数值	
[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。

#### ▶ 表 3-2: 测试函式集

<b>节</b> 章	功能函数
3.2	测试函式集
	<pre>float PCI_FloatSub2(float fA, float fB);</pre>
	<pre>short PCI_ShortSub2(short nA, short nB);</pre>
	DWORD PCI_GetD11Version(void);

#### ▶ 表 3-3: 驱动函式集

节章	功能函数
3.3	驱动函式集
	<pre>WORD PCI_DriverInit(WORD *wTolalBoard);</pre>
	WORD PCI_DriverClose(void);
	WORD <b>PCI_GetDriverVersion</b> (WORD <b>*</b> wDriverVersion);
	WORD PCI_GetConfigAddressSpace(WORD wBoardNo, WORD *TypeID, WORD
	<pre>*wAddress0, WORD *Waddress1, WORD *wAddress2, WORD *wAddress3, WORD</pre>
	<pre>*wAddress4, WORD *wAddress5);</pre>

## ▶ 表 3-4: 数字输出入函式集(PCI-P16R16 系列卡适用)

节章	功能函数	
3.4	> 数字输出入函式集 (PCI-P16R16 系列卡适用)	
	<pre>void CALLBACK P16R16_DO(WORD BoardAddr, WORD OutData);</pre>	
	WORD CALLBACK <pre>P16R16_DI(WORD BoardAddr);</pre>	

## ▶ 表 3-5: 数字输出入函式集(PCI-P8R8 系列卡适用)

节章	功能函数
3.5	数字输出入函式集(PCI-P8R8 系列卡适用)
	<pre>void CALLBACK P8R8_D0(WORD BoardAddr, WORD OutData);</pre>
	BYTE CALLBACK <a href="https://www.barneedim.exact.exact.com">P8R8_DI</a> (WORD BoardAddr);

# 3.1 错误码列表

当应用程序回传错误码时,建议参考下列几点来确认及检查:

- 1. 设备驱动程序是否安装成功?
- 2. 板卡是否有堵塞?
- 3. 板卡是否与其它设备冲突?
- 4. 关闭其它应用程序,以释放系统资源。
- 5. 赏试使用另一个插槽来插上板卡。
- 6. 再次重新启动系统。

#### ▶ DLL 返回的代码定义如下:

错误码	Error ID	Error String
0	NoError	正常
1	DriverHandleError	驱动程序开起错误
2	DriverCallError	呼叫驱动程序 NAPPCI.VXD 或 NAPWNT.SYS 错误
3	NotFoundBoard	在系统上没有发现板卡
4	FindBoardError	操作系统上找不到板卡
F	ExceedBoardNumber	板卡数量超过最大值。
Э		(有效板卡数范围 0~TotalBoard -1)

# 3.2 测试函数

# PCI\_FloatSub2

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

*<u>fB</u>* [Input] 4 位浮点类型值

# ▶ 回传值:

运算 fA – fB 之值

# PCI\_ShortSub2

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:

<u>nA</u> [Input] 2 位短整型数据值

<u>nB</u> [Input] 2 位短整型数据值

## ▶ 回传值:

运算 nA – nB 之值

# **PCI\_GetDIIVersion**

获得 P16R16.DLL 版本号。

▶ 语法:

WORD PCI\_GetDIlVersion(void);

#### ▶ 参数:

无

▶ 回传值:

P16R16.DLL 函式库的版本编号。如,传回值为 200 (hex),意旨版本为 2.00。

# 3.3 驱动函式集

# **PCI\_DriverInit**

呼叫此函式时会向系统要求分配资源,并且开始寻找所有 PCI-P16R16 有支持的板卡,而对每 一张板卡作初使化动作,最后取得板卡的数量。需在程序起始点,使用其他的函式之前呼叫。

#### ▶ 语法:

WORD PCI\_DriverInit(WORD \* wTolalBoard);

#### ▶ 参数:

#### <u>\*wTotalBoard</u>

[Output] 取得板卡的数量。

- 当 wTotalBoard = 1→ 一张 P16R16 或 P8R8 系列卡安装成功
- 当 wTotalBoard = 2 → 二张 P16R16 或 P8R8 系列卡安装成功 可能的组合如下
  - 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

#### ▶ 回传值:

请参考 "第 3.1 章错误码列表" 回传值定义

# ▶ 范例:

```
下列为 Visual C<sup>++</sup> 范例程序代码:
```

```
LRESULT CALLBACK WndProc(HWND hwnd, UINT iMsg, WPARAM wParam, LPARAM, IParam)
{
 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);
      }
      :
      :
      :
}
```

# PCI\_DriverClose

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,该函数仅仅提供终止 Win32 程序并释放内存,且仅返回 NoError。

▶ 语法:

vio PCI\_DriverClose(void);

▶ 参数:

无

▶ 回传值:

无

# PCI\_GetDriverVersion

取得 PCI-P16R16 驱动程序的版本编号。(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] 取得 PCI-P16R16 驱动程序的版本值。如,传回值为 200 (hex),意旨版本为 2.00。

▶ 回传值:

请参考 "第 3.1 章错误码列表" 回传值定义

# PCI\_GetConfigAddressSpace

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\_GetConfigAddressSpace (WORD wBoardNo,

DWORD **\*wTypeID**, WORD **\*wAddress0**, WORD **\*wAddress1**, WORD **\*wAddress2**, WORD **\*wAddress3**, WORD **\*wAddress4**, WORd **\*wAddress5**);

▶ 参数:

#### <u>wBoardNo</u>

[Input] 由用户设定的板卡模块识别号码

#### <u>wTypeID</u>

[Input] 用户设定板卡模块类型号码 (Type IDs), 此号码可参考下列表。

wTypeID	Model
0	为 PCI-P16R16 系列卡
1	为 PCI-P8R8 系列卡
2	为 PCI-TMC12 系列卡
3	为 PIO-DA16 系列卡
4	为 PIO-DA8 系列卡

wAddress0

<u>wAddress1</u> <u>wAddress2</u>

wAddress3

wAddress4

wAddress5

[Output] 六个 PCI 设备的基地址将储存在这些参数中。

## ▶ 回传值:

请参考 "第 3.1 章错误码列表" 回传值定义

# 3.4 PCI-P16R16 系列数字量输出入函数

注意: PCI-P16R16 系列卡包含 PCI-P16R16(U)、PCI-P16C16(U)、PCI-P16POR16(U) 及 PEX-P16POR16i。

# P16R16\_DO

从板卡的 P16R16 的 DO 端口发送 16 位数据。

▶ 语法:

void P16R16\_DO(DWORD BaseAddr, WORD OutData);

▶ 参数:

<u>BaseAddr</u> [Input] DO 端口基地址

<u>OutData</u> [Input] 发送到 DO 端口的 16 位数据

## ▶ 回传值:

无

## ▶ 范例:

请参考至第 24 到 27 页

# P16R16\_DI

从板卡 P16R16 DI 端口读取 16 位数据。

## ▶ 语法:

WORD P16R16\_DI(DWORD BaseAddr);

## ▶ 参数:

## <u>BaseAddr</u>

[Input] DO 端口基地址

## ▶ 回传值:

从 DI 端口读取的 16 位数据

#### ▶ 范例:

/*******	***************************************		
/* This prog /********	/* This program is developed by Turbo C 2.0 */ /**********************************		
/* Demo 1: ( /********	One P16R16 card demo. */ **********************************/		
#include "P1	.6R16.H"		
int main()			
{			
int	i,j;		
WORD	nVal;		
float	fVal;		
WORD	wBoards,wRetVal,wVal;		
WORD	wInData;		
WORD	wTypelD;		
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 P16R16 Cards in this PC\n",wBoards);
if( wBoards==0 )
{
   putch(0x07); putch(0x07); putch(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++)</pre>
{
   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( wP16R16No<1 )
 {
     putch(0x07);
     printf("Please plug one PCI-P16R16 in PC !!!\n");
     exit(0);
 }
 /********** PCI-P16R16 DI/DO demo **********/
 printf("The PCI-P16R16 DO/DI testing !!!\n");
 P16R16_DO(P16R16_BaseAddress,0x0000);
                                         /* Digital output */
              /* Delay a little time 500ms */
 delay(500);
 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;

# 3.5 PCI-P8R8 系列数字量输出入函数

注意: PCI-P8R8 系列卡包含 PCI-P8R8(U) 及 PEX-P8POR8i。

# P8R8\_DO

从板卡的 P8R8 的 DO 端口发送 8 位数据。

▶ 语法:

void P8R8\_DO(DWORD BaseAddr, WORD OutData);

▶ 参数:

<u>BaseAddr</u> [Input] DO 端口基地址

<u>OutData</u> [Input] 发送到 DO 端口的 8 位数据

## ▶ 回传值:

无

## ▶ 范例:

请参考至第 29 到 32 页

# P8R8\_DI

从板卡 P8R8 的 DI 端口读取 8 位数。

#### ▶ 语法:

WORD P8R8\_DI(DWORD BaseAddr);

#### ▶ 参数:

#### <u>BaseAddr</u> [Input] DO 端口基地址

#### ▶ 回传值:

从 DI 端口读取的 8 位数据

▶ 范例:

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();

```
/* initiating 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 )
{
   putch(0x07); putch(0x07); putch(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++)</pre>
{
   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)
 {
     putch(0x07);
     printf("Please plug one PCI-P8R8 in PC !!!\n");
     exit(0);
 }
   /********* PCI-P8R8 DI/DO demo **********/
   printf("The PCI-P8R8 DO/DI testing !!!\n");
 P8R8_DO(P8R8_BaseAddress,0x0000);
                                       /* Digital output */
                             /* Delay a little time */
 delay(500);
 wInData=P8R8_DI(P8R8_BaseAddress); /* Digital input */
 printf("Digital Output -> 0000H | Digital Input -> %04xH\n",wInData);
 P8R8_DO(P8R8_BaseAddress,0xFFFF);
                                       /* Digital output */
                             /* Delay a little time */
 delay(500);
 wInData=P8R8_DI(P8R8_BaseAddress); /* Digital input */
 printf("Digital Output -> FFFFH | Digital Input -> %04xH\n",wInData);
 P8R8_DO(P8R8_BaseAddress,0x5555);
                                       /* Digital output */
                             /* Delay a little time */
 delay(500);
```

}

```
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;
```

4. 示列程序

# 4.1 Windows 操作系统

PCI-P16R16 系列驱动函式库集成了各种函式,用户可以利用它们来开发各种应用程序在泓格的装置上。这些 API 函式支持各种开发环境及程序语言,包括了 Microsoft Visual C++, Visual Basic, Borland Delphi, Borland C Builder++, Microsoft Visual C++.NET, Microsoft Visual C#.NET, Microsoft Visual VB.NET。

用户能够从随机出货的配件软件 CD 光盘或从泓格的软件下载网站中来取范例程序。详细取得 下载位置如下::



如果 DLL 驱动没有正确安装那么所有 DEMO 程序将不能正常工作。在 DLL 驱动安装过程的时候, 安装程序将注册适当的内核驱动到操作系统中,并且拷贝 DLL 驱动和 DEMO 程序到适当的位置, 你可以选择(Win98/Me/NT/2000 and 32-bit Win XP/2003/Visa/7/8)驱动软件包。一次完整的 驱动安装,下列出现相关的 DEMO 程序、库文件、声明的头文件在不同的运行环境:

<ul> <li>♦ BCB3 → for Borland C<sup>++</sup> Builder 3</li> <li>P16R16.H → Header files</li> <li>P16R16.LIB → Linkage library for BCB only</li> </ul>	<ul> <li>Delphi3 → for Delphi 3</li> <li>P16R16.PAS → Declaration files</li> </ul>
<ul> <li>♦ VC6 → for Visual C<sup>++</sup> 6</li> <li>P16R16.H → Header files</li> <li>P16R16.LIB → Linkage library for VC only</li> </ul>	<ul> <li>◆ VB6 → for Visual Basic 6</li> <li>P16R16.BAS → Declaration files</li> </ul>
♦ VB.NET2005 → for VB.NET2005 P16R16.vb → Visual Basic Source files	<ul> <li>◆ CSharp2005 → for C#.NET2005</li> <li>P16R16.cs → Visual C# Source files</li> </ul>

<u>注意: 示例程序中所需的硬件配置、接线...等,请用户参考至各 PCI-P16R16 系列硬件手册来自</u> 行设定。

# 4.2 DOS 操作系统

示例程序,下载位置如下:

CD:\NAPDOS\PCI\PCI-P16R16\DOS\PP16R16\

http://ftp.icpdas.com/pub/cd/iocard/pci/napdos/pci/pci-p16r16/dos/pp16r16/

•	Demo code and Library for Borla \P16R16\BC\HUGE\DEMO \P16R16\BC\HUGE\LIB \P16R16\BC\LARGE\DEMO \P16R16\BC\LARGE\LIB	nd C++ → huge mode demo programs → huge mode library, P16R16H.LIB → large mode demo programs → large mode library, P16R16L.LIB
	Demo code and Library for MSC \P16R16\MSC\HUGE\DEMO \P16R16\MSC\HUGE\LIB \P16R16\MSC\LARGE\DEMO \P16R16\MSC\LARGE\LIB	<ul> <li>→ huge mode demo programs</li> <li>→ huge mode library, P16R16H.LIB</li> <li>→ large mode demo programs</li> <li>→ large mode library, P16R16L.LIB</li> </ul>
•	Demo code and Library for TC \P16R16\TC\HUGE\DEMO \P16R16\TC\HUGE\LIB \P16R16\TC\LARGE\DEMO \P16R16\TC\LARGE\LIB	<ul> <li>→ huge mode demo programs</li> <li>→ huge mode library, P16R16H.LIB</li> <li>→ large mode demo programs</li> <li>→ large mode library, P16R16H.LIB</li> </ul>

示例程序列表	
Demo1	数字量输出入示例程序适用于一张 PCI-P16R16 系列卡
Demo2	数字量输出入示例程序适用于一张 PCI-P8R8 系列卡
Demo3	数字量输出入示例程序适用于一张 PCI-P16R16 及一张 PCI-P8R8 系列卡
Demo4	数字量输出入示例程序适用于二张 PCI-P16R16 系列卡

<u>注意: 示例程序中所需的硬件配置、接线...等,请用户参考至各 PCI-P16R16 系列硬件手册来自</u> 行设定。

# 5. 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: <u>Service.icpdas@gmail.com</u>、<u>Service@icpads.com</u>),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/2008/7/8。

- 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。