

# PCI/PEX-1002 H/L

---

## Linux Software User Manual

### **Warranty**

All products manufactured by ICP DAS are warranted against defective materials for a period of one year from the date of delivery to the original purchaser.

### **Warning**

ICP DAS assume no liability for damages consequent to the use of this product. ICP DAS reserves the right to change this manual at any time without notice. The information furnished by ICP DAS is believed to be accurate and reliable. However, no responsibility is assumed by ICP DAS for its use, nor for any infringements of patents or other rights of third parties resulting from its use.

### **Copyright**

Copyright 2019 by ICP DAS. All rights are reserved.

### **Trademark**

The names used for identification only may be registered trademarks of their respective companies.

# Tables of Contents

|   |           |
|---|-----------|
| <b>Linux Software Installation .....</b>                | <b>3</b>  |
| 1.1 LINUX DRIVER INSTALLING PROCEDURE.....              | 3         |
| 1.2 LINUX DRIVER UNINSTALLING PROCEDURE.....            | 3         |
| <b>2. Static Library Function Description.....</b>      | <b>4</b>  |
| 2.1 TABLE OF ERRORCODE AND ERRORSTRING.....             | 5         |
| 2.2 FUNCTION DESCRIPTIONS.....                          | 5         |
| 2.3 COMMON FUNCTION DEFINITION .....                    | 6         |
| 2.3.1 PCIDA_GetDriverVersion .....                      | 6         |
| 2.3.2 PCIDA_GetLibraryVersion .....                     | 6         |
| 2.3.3 PCIDA_Open .....                                  | 6         |
| 2.3.4 PCIDA_Close.....                                  | 7         |
| 2.3.5 PCIDA_DriverInit.....                             | 7         |
| 2.3.6 PCIDA_SetGain.....                                | 7         |
| 2.3.7 PCIDA_SetTriggerType .....                        | 8         |
| 2.3.8 PCIDA_SetAIChannel.....                           | 9         |
| 2.3.9 PCIDA_ReadAI.....                                 | 9         |
| 2.3.10 PCIDA_IntInstall .....                           | 9         |
| 2.3.11 PCIDA_IRQEnable.....                             | 10        |
| 2.3.12 PCIDA_IntRemove.....                             | 10        |
| 2.3.13 PCIDA_8254Control .....                          | 11        |
| 2.3.14 PCIDA_8254C0 .....                               | 11        |
| 2.3.15 PCIDA_8254C1 .....                               | 11        |
| 2.3.16 PCIDA_8254C2 .....                               | 12        |
| 2.3.17 PCIDA_WriteDigitalOutput.....                    | 12        |
| 2.3.18 PCIDA_ReadDigitalInput.....                      | 13        |
| 2.3.19 PCIDA_ResetDevice .....                          | 13        |
| <b>3. PCI/PEX-1002 H/L Demo Programs For Linux.....</b> | <b>14</b> |
| 3.1 DEMO CODE "DIO.C" 、"DIO2.C" 、"DIOA.C" .....         | 15        |
| 3.2 DEMO CODE "RST.C" 、"RST_A.C" .....                  | 15        |
| 3.3 DEMO CODE "TIME_SPAN.C" .....                       | 15        |
| 3.4 DEMO CODE "AI_SOFT.C" 、"AI_SOFT_A.C" .....          | 15        |
| 3.5 DEMO CODE "AI_PACER.C" 、"AI_PACER_A.C" .....        | 15        |
| 3.6 DEMO CODE "AI_TRIGGER.C" 、"AI_TRIGGER_A.C" .....    | 15        |
| 3.7 DEMO CODE "INT.C" 、"INT_A.C" .....                  | 15        |

---

## Linux Software Installation

The PCI/PEX-1002 H/L can be used in linux kernel 2.4.X and 2.6.X to 4.15.X. For Linux O.S, the recommended installation and uninstall steps are given in Sec 1.1 ~ 1.2

### 1.1 Linux Driver Installing Procedure

Step 1: Copy the linux driver “ixpci-0.8.16.tar.gz”(or the later driver version) in the directory “NAPDOS\Linux” of the companion CD to the linux host that you want to install driver.

Step 2: Decompress the tarball “ixpci-0.8.16.tar.gz”.

Step 3: Type ``cd'` to the directory containing the package's source code and type `./configure'` to configure the package for your system.

Step 4: Type ``make'` to compile the package.

Step 5: Type ``./ixpci.inst'` to install the PCI driver module and build the device file “ixpciX” in the device directory “/dev” automatically.

---

### 1.2 Linux Driver Uninstalling Procedure

Step 1: Type ``cd'` to the directory containing the package's source code.

Step 2: Type ``./ixpci.remove'` to remove the PCI driver module.

## 2. Static Library Function Description

The static library is the collection of function calls of the PCI cards for linux kernel 2.4.x and 2.6.x system. The application structure is presented as following figure. The user application program developed by C(C++) language can call library “libpci.a/libpci\_64.a” in user mode. And then static library will call the module ixpci to access the hardware system.

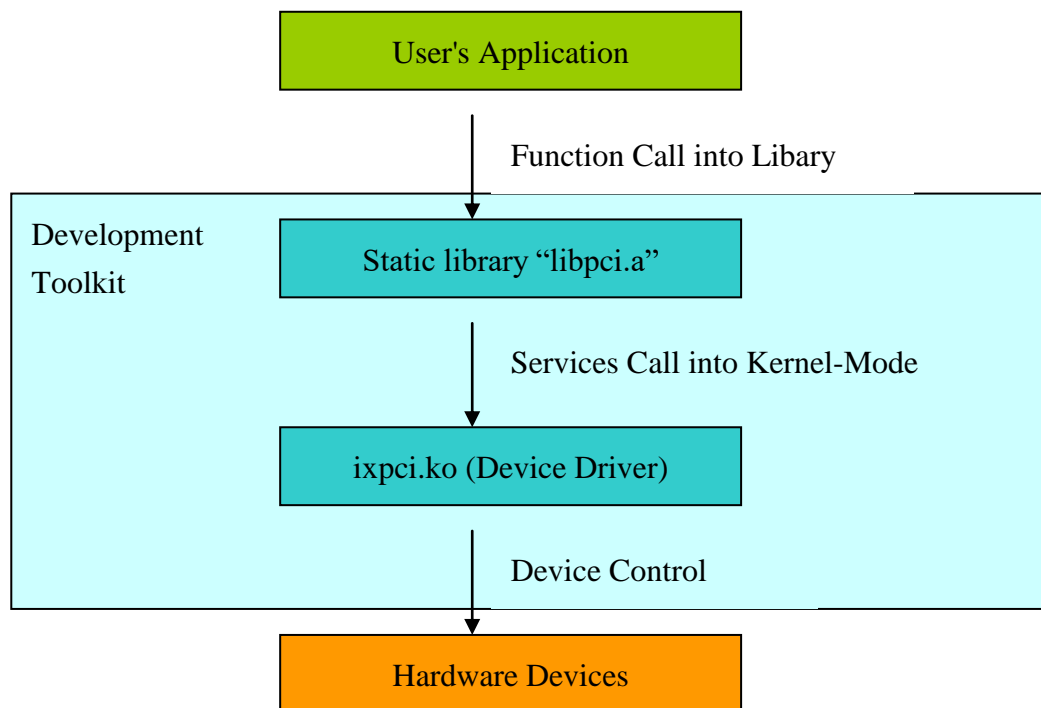


Figure 2.1

## 2.1 Table of ErrorCode and ErrorString

Table 2.1

| Error Code | Error ID                    | Error String                                      |
|------------|-----------------------------|---|
| 0          | PCIDA_NOERROR               | OK (No error )                                    |
| 1          | PCIDA_MODULE_NAME_GET_ERROR | Module name can't get from file /proc/ixpci/ixpci |
| 2          | PCIDA_PARAMETER_ERROR       | Parameter error                                   |
| 3          | PCIDA_IOTCL_ERROR           | IOCTL error                                       |
| 4          | PCIDA_SIGACTION_ERROR       | Set sigaction error                               |

---

## 2.2 Function Descriptions

Table 2.2

| Common Function Definition                  |
|---|
| char* PCIDA_GetDriverVersion(void);         |
| char* PCIDA_GetLibraryVersion(void);        |
| int PCIDA_Open(char *);                     |
| WORD PCIDA_Close(WORD);                     |
| WORD PCIDA_DriverInit(WORD);                |
| WORD PCIDA_SetGain(WORD, BYTE);             |
| WORD PCIDA_SetTriggerType(WORD, WORD);      |
| WORD PCIDA_SetAIChannel(WORD ,BYTE);        |
| WORD PCIDA_ReadAI(WORD, BYTE, WORD*);       |
| WORD PCIDA_IntlInstall(WORD, HANDLE, WORD); |
| WORD PCIDA_IRQEnable(WORD);                 |
| WORD PCIDA_IntRemove(WORD);                 |
| WORD PCIDA_8254Control(WORD, WORD);         |
| WORD PCIDA_8254C0(WORD, BYTE, BYTE);        |
| WORD PCIDA_8254C1(WORD, BYTE, BYTE);        |
| WORD PCIDA_8254C2(WORD, BYTE, BYTE);        |
| WORD PCIDA_WriteDigitalOutput(WORD, WORD);  |
| WORD PCIDA_ReadDigitalInput(WORD, WORD*);   |
| WORD PCIDA_ResetDevice(WORD);               |

---

## 2.3 Common Function Definition

---

### 2.3.1 PCIDA\_GetDriverVersion

- **Description:**  
Show the version number of PCI linux driver.
- **Syntax:**  
`char* PCIDA_GetDriverVersion(Void)`
- **Parameter:**  
None
- **Return:**  
Ixpci driver version.

---

### 2.3.2 PCIDA\_GetLibraryVersion

- **Description:**  
Show the version number of PCI linux static library.
- **Syntax:**  
`char* PCIDA_GetLibraryVersion(void)`
- **Parameter:**  
None
- **Return:**  
Ixpci static lib version.

---

### 2.3.3 PCIDA\_Open

- **Description:**  
Open device file.
- **Syntax:**  
`int PCIDA_Open(char *dev_file)`
- **Parameter:**  
`dev_file` : The path of device file
- **Return:**  
File descriptor of device file. If the file descriptor < 0, it means that open device file failure.

---

#### 2.3.4 PCIDA\_Close

- **Description :**  
Close device file.
- **Syntax :**  
WORD PCIDA\_Close(WORD fd)
- **Parameter :**  
fd : The file descriptor of device file that get from function PCIDA\_Open.
- **Return:**  
The code "PCIDA\_NOERROR"(Please refer to "Section 2.1 Error Code").

---

#### 2.3.5 PCIDA\_DriverInit

- **Description :**  
To allocates the computer resource for the device.  
This function must be called once before applying other PCIDA functions.
- **Syntax :**  
WORD PCIDA\_DriverInit(WORD fd)
- **Parameter :**  
fd: The file descriptor of device file that get from function PCIDA\_Open
- **Return:**  
"PCIDA\_NOERROR"  
"PCIDA\_MODULE\_NAME\_GET\_ERROR" (Please refer to "Section 2.1 Error Code").

---

#### 2.3.6 PCIDA\_SetGain

- **Description :**  
Set Analog Input gain control
- **Syntax :**  
WORD PCIDA\_SetGain (WORD fd, BYTE gain)
- **Parameter :**  
fd: The file descriptor of device file that get from function PCIDA\_Open.  
gain: The analog input gain control code, and the corresponding gain is:

For PCI-1002L/LU and PEX-1002L:

| [Bit1, Bit0] | [0 0] | [0 1] | [1 0] | [1 1] |
|--------------|-------|-------|-------|-------|
| Gain         | 1     | 2     | 4     | 8     |

For PCI-1002H/HU and PEX-1002H:

| [Bit1, Bit0] | [0 0] | [0 1] | [1 0] | [1 1] |
|--------------|-------|-------|-------|-------|
| Gain         | 1     | 10    | 100   | 1000  |

| Model              | PCI-1002L/LU and PEX-1002L (Low-Gain)  |           |             |              |
|--------------------|--|-----------|-------------|--------------|
| Gain               | 1                                      | 2         | 4           | 8            |
| Bipolar            | $\pm 10$ V                             | $\pm 5$ V | $\pm 2.5$ V | $\pm 1.25$ V |
| Sampling Rate Max. | 100 kS/s.                              |           |             |              |
| Model              | PCI-1002H/HU and PEX-1002H (High-Gain) |           |             |              |
| Gain               | 1                                      | 10        | 100         | 1000         |
| Bipolar            | $\pm 10$ V                             | $\pm 1$ V | $\pm 0.1$ V | $\pm 0.01$ V |
| Sampling Rate Max. | 44 kS/s.                               | 36 kS/s.  | 7 kS/s.     | 0.8 kS/s.    |

- **Return:**  
 "PCIDA\_PARAMETER\_ERROR"  
 "PCIDA\_NOERROR"  
 (Please refer to "Section 2.1 Error Code")

---

### 2.3.7 PCIDA\_SetTriggerType

- **Description :**  
Set AD trigger method.
- **Syntax :**  
WORD PCIDA\_SetTriggerType (WORD fd, WORD trigger)
- **Parameter :**  
 fd: The file descriptor of device file that get from function PCIDA\_Open.  
 trigger: set trigger type  
     1 means software trigger  
     2 means pacer trigger  
     3 means external trigger
- **Return:**  
 "PCIDA\_NOERROR"  
 "PCIDA\_PARAMETER\_ERROR"  
 "PCIDA\_IOTCL\_ERROR"



(Please refer to "Section 2.1 Error Code")

---

### 2.3.8 PCIDA\_SetAIChannel

- **Description :**  
Set channel to read Analog Input value.
- **Syntax :**  
WORD PCIDA\_SetAIChannel (WORD fd, BYTE ch)
- **Parameter :**  
fd : The file descriptor of device file that get from function PCIDA\_Open.  
ch: Analog Input channel.
- **Return:**  
"PCIDA\_NOERROR"  
"PCIDA\_IOTCL\_ERROR"  
(Please refer to "Section 2.1 Error Code")

---

### 2.3.9 PCIDA\_ReadAI

- **Description :**  
Read back Analog Input value.
- **Syntax :**  
WORD PCIDA\_ReadAI(WORD fd, BYTE trigger, WORD\* value)
- **Parameter :**  
fd : The file descriptor of device file that get from function PCIDA\_Open.  
trigger: trigger type, which you set in PCIDA\_SetTriggerType().  
value: Analog Input value.
- **Return:**  
"PCIDA\_NOERROR"  
"PCIDA\_PARAMETER\_ERROR"  
"PCIDA\_IOTCL\_ERROR"  
(Please refer to "Section 2.1 Error Code")

---

### 2.3.10 PCIDA\_IntlInstall

- **Description :**  
Set user signal for interrupt.
- **Syntax :**

WORD PCIDA\_IntInstall(WORD fd, HANDLE hisr, WORD sig\_id);

- **Parameter :**

fd :The file descriptor of device file that get from function PCIDA\_Open.

hisr : interrupt handler.

sig\_id: signal number.

- **Return:**

“PCIDA\_SIGACTION\_ERROR”

“PCIDA\_IOTCL\_ERROR”

“PCIDA\_NOERROR”

(Please refer to "Section 2.1 Error Code")

---

### 2.3.11 PCIDA\_IRQEnable

- **Description :**

Enable board interrupt.

- **Syntax :**

WORD PCIDA\_IRQEnable(WORD fd);

- **Parameter :**

fd :The file descriptor of device file that get from function PCIDA\_Open.

- **Return:**

“PCIDA\_NOERROR”

“PCIDA\_IOTCL\_ERROR”

(Please refer to "Section 2.1 Error Code")

---

### 2.3.12 PCIDA\_IntRemove

- **Description :**

Remove user signal for interrupt.

- **Syntax :**

WORD PCIDA\_IntRemove(WORD fd);

- **Parameter :**

fd :The file descriptor of device file that get from function PCIDA\_Open.

- **Return:**

“PCIDA\_NOERROR”

(Please refer to "Section 2.1 Error Code")

---

### 2.3.13 PCIDA\_8254Control

- **Description :**  
Control 8254 timer.
- **Syntax :**  
WORD PCIDA\_8254Control(WORD fd, WORD value);
- **Parameter :**  
fd :The file descriptor of device file that get from function PCIDA\_Open.  
value : Used to control 8254.
- **Return:**  
"PCIDA\_NOERROR"  
"PCIDA\_IOTCL\_ERROR"  
(Please refer to "Section 2.1 Error Code")

---

### 2.3.14 PCIDA\_8254C0

- **Description :**  
8254 timer1.
- **Syntax :**  
WORD PCIDA\_8254C0(WORD fd, BYTE high\_byte, BYTE low\_byte);
- **Parameter :**  
fd :The file descriptor of device file that get from function PCIDA\_Open.  
high\_byte : 8254 timer1 high byte  
low\_byte: 8254 timer1 low byte
- **Return:**  
"PCIDA\_NOERROR"  
"PCIDA\_IOTCL\_ERROR"  
(Please refer to "Section 2.1 Error Code")

---

### 2.3.15 PCIDA\_8254C1

- **Description :**  
8254 timer2.
- **Syntax :**  
WORD PCIDA\_8254C1(WORD fd, BYTE high\_byte, BYTE low\_byte);
- **Parameter :**  
fd :The file descriptor of device file that get from function PCIDA\_Open.

high\_byte : 8254 timer2 high byte

low\_byte: 8254 timer2 low byte

- **Return:**  
"PCIDA\_NOERROR"  
"PCIDA\_IOTCL\_ERROR"  
(Please refer to "Section 2.1 Error Code")

---

### 2.3.16 PCIDA\_8254C2

- **Description :**  
8254 timer3.
- **Syntax :**  
WORD PCIDA\_8254C2(WORD fd, BYTE high\_byte, BYTE low\_byte);
- **Parameter :**  
fd :The file descriptor of device file that get from function PCIDA\_Open.  
high\_byte : 8254 timer3 high byte  
low\_byte: 8254 timer3 low byte
- **Return:**  
"PCIDA\_NOERROR"  
"PCIDA\_IOTCL\_ERROR"  
(Please refer to "Section 2.1 Error Code")

---

### 2.3.17 PCIDA\_WriteDigitalOutput

- **Description :**  
Set digital output value.
- **Syntax :**  
WORD PCIDA\_WriteDigitalOutput(WORD fd, WORD value);
- **Parameter :**  
fd :The file descriptor of device file that get from function PCIDA\_Open.  
value: Digital output value.
- **Return:**  
"PCIDA\_NOERROR"  
"PCIDA\_IOTCL\_ERROR"  
(Please refer to "Section 2.1 Error Code")

---

### 2.3.18 PCIDA\_ReadDigitalInput

- **Description :**  
Read back digital input value.
- **Syntax :**  
WORD PCIDA\_ReadDigitalInput(WORD fd, WORD\* value);
- **Parameter :**  
fd :The file descriptor of device file that get from function PCIDA\_Open.  
value: Digital input value.
- **Return:**  
"PCIDA\_NOERROR"  
"PCIDA\_IOTCL\_ERROR"  
(Please refer to "Section 2.1 Error Code")

---

### 2.3.19 PCIDA\_ResetDevice

- **Description :**  
Reset device module.
- **Syntax :**  
WORD PCIDA\_ResetDevice(WORD);
- **Parameter :**  
fd :The file descriptor of device file that get from function PCIDA\_Open.
- **Return:**  
"PCIDA\_NOERROR"  
"PCIDA\_IOTCL\_ERROR"  
(Please refer to "Section 2.1 Error Code")

---

### 3. PCI/PEX-1002 H/L Demo Programs For Linux

All of demo programs will not work normally if PCI linux driver would not be installed correctly. During the installation process of PCI linux driver, the install-scripts “ixpci.inst” will setup the correct kernel driver. After driver (version 0.8.16 or the later driver version) compiled and installation, the related demo programs, development library and declaration header files for different development environments are presented as follows.

Table 3.1

| Driver Name  | Directory Path               | File Name            | Description                              |
|--------------|------------------------------|----------------------|--|
| ixpci-0.8.16 | Include                      | pcidio.h             | PCI library header                       |
|              | lib                          | libpci.a/libpci_64.a | PCI static library                       |
|              | examples/<br>pci1002_pex1002 | dio.c                | DI/O demo                                |
|              |                              | dio2.c               | DI/O demo                                |
|              |                              | rst.c                | Reset control register                   |
|              |                              | time_span.c          | Delay setting time                       |
|              |                              | ai_soft.c            | Read AI by software trigger              |
|              |                              | ai_pacer.c           | Read AI by pacer trigger                 |
|              |                              | ai_trigger.c         | Read AI by external trigger              |
|              |                              | Int.c                | Example of interrupt handler             |
|              |                              | dio_a.c              | DI/O demo with library                   |
|              |                              | rst_a.c              | Reset control register with library      |
|              |                              | ai_soft_a.c          | Read AI by software trigger with library |
|              |                              | ai_pacer_a.c         | Read AI by pacer trigger with library    |
|              |                              | ai_trigger_a.c       | Read AI by external                      |

|  |  |         |   |
|--|--|---------|---|
|  |  |         | trigger with library                      |
|  |  | int_a.c | Example of interrupt handler with library |

---

### 3.1 Demo code “dio.c” 、 ”dio2.c” 、 ”dioa.c”

These demo programs are used to output data from CON1 and read data from CON2.

---

### 3.2 Demo code “rst.c” 、 ”rst\_a.c”

These demo programs are used to reset control register.

---

### 3.3 Demo code “time\_span.c”

This demo program will use 8254 timer2 to delay 6000 us.

---

### 3.4 Demo code “ai\_soft.c” 、 ”ai\_soft\_a.c”

These demo programs are used to read analog input value by software trigger.

---

### 3.5 Demo code “ai\_pacer.c” 、 ”ai\_pacer\_a.c”

These demo programs are used to read analog input value by pacer trigger.

---

### 3.6 Demo code “ai\_trigger.c” 、 ”ai\_trigger\_a.c”

These demo programs are used to read analog input value by external trigger.

---

### 3.7 Demo code “int.c” 、 ”int\_a.c”

These demo programs are used to read analog input value by external trigger.