NuDAQ®
cPCI-7248/7249R
PCI-7224/7248/7296
24/48/96-CH Digital I/O Card
Users’ Guide
# Getting service from ADLINK

Customer Satisfaction is top priority for ADLINK TECHNOLOGY INC. If you need any help or service, please contact us.

<table>
<thead>
<tr>
<th>ADLINK TECHNOLOGY INC.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Web Site</td>
</tr>
<tr>
<td>Sales &amp; Service</td>
</tr>
<tr>
<td>TEL</td>
</tr>
<tr>
<td>FAX</td>
</tr>
<tr>
<td>Address</td>
</tr>
</tbody>
</table>

Please email or FAX your detailed information for prompt, satisfactory, and consistent service.

## Detailed Company Information

<table>
<thead>
<tr>
<th>Company/Organization</th>
<th>Contact Person</th>
<th>E-mail Address</th>
<th>Address</th>
<th>Country</th>
<th>TEL</th>
<th>FAX</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

## Questions

### Environment

<table>
<thead>
<tr>
<th>OS:</th>
<th>Computer Brand:</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>M/B:</th>
<th>CPU:</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Chipset:</th>
<th>BIOS:</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Video Card:</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>NIC:</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Other:</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
</tr>
</tbody>
</table>

## Detail Description


## Suggestions for ADLINK
## Table of Contents

### Chapter 1 Introduction

1. Features ................................................................................. 2
1.2 Applications ............................................................................ 2
1.3 Specifications ......................................................................... 3
1.4 Software Support ................................................................. 5

### Chapter 2 Installation

2.1 What You Have ...................................................................... 9
2.2 Unpacking............................................................................. 10
2.3 PCB Layout .......................................................................... 11
2.4 Hardware Installation ............................................................ 13
2.5 Device Installation for Windows Systems ............................ 14
2.6 Connector Pin Assignment................................................... 15
2.7 Jumpers Description............................................................. 19
2.8 Termination Boards Connection ........................................... 20

### Chapter 3 Registers Format

3.1 PCI Plug and Play Registers ................................................ 23
3.2 I/O Address Map ................................................................... 24

### Chapter 4 Operation Theorem

4.1 Digital I/O Ports .................................................................... 25
4.2 Timer/Counter Operation ...................................................... 28
4.3 Interrupt Multiplexing ............................................................ 29
4.4 12V and 5V Power Supply ................................................... 31

### Chapter 5 C/C++ Libraries

5.1 Libraries Installation ............................................................. 33
5.2 Programming Guide ............................................................. 34
5.3 _7248/96_ Initial .................................................................... 35
5.4 Digital Input........................................................................... 36
5.5 Digital Output ........................................................................ 38
5.6 Configuration Port ............................................................... 39
5.7 Configuration Channel .......................................................... 40
5.8 Set Interrupt Control ............................................................. 42
5.9 Timer Start............................................................................ 43
<table>
<thead>
<tr>
<th>Section</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>5.10</td>
<td>Timer Read</td>
<td>44</td>
</tr>
<tr>
<td>5.11</td>
<td>Timer Stop</td>
<td>45</td>
</tr>
<tr>
<td>5.12</td>
<td>Cascaded Timer</td>
<td>46</td>
</tr>
<tr>
<td>5.13</td>
<td>Get IRQ Status</td>
<td>47</td>
</tr>
<tr>
<td>5.14</td>
<td>Clear IRQ</td>
<td>48</td>
</tr>
<tr>
<td>5.15</td>
<td>Software Reset</td>
<td>48</td>
</tr>
<tr>
<td>5.16</td>
<td>Interrupt Start under Windows</td>
<td>49</td>
</tr>
<tr>
<td>5.17</td>
<td>Interrupt Stop under Windows</td>
<td>50</td>
</tr>
</tbody>
</table>
How to Use This Guide

This manual is designed to help you use the 7248/96 series products. It describes how to modify and control various functions on the cards to meet your requirements. It is divided into five chapters:

- **Chapter 1, Introduction**, gives an overview of the product features, applications, and specifications.

- **Chapter 2, Installation**, describes how to install the 7248/96 series products. The layout of the card is shown; as well as the description for the jumper settings, connectors specifications, and installation.

- **Chapter 3, Registers Format**, describes the low-level register structure and format of the PCI-7224/7248/7296.

- **Chapter 4, Operation Theorem**, describes in further detail of the system’s versatile functions, including DIO, timer / counter, and interrupt systems.

- **Chapter 5, C/C++ Library** specifies the software library of C language under DOS environment allowing the ease of use of the functions on this card for the user.
Introduction

The 7248/7296 series products are general purpose digital I/O cards. This series includes four cards:

- PCI-7224: 24-CH DIO card
- PCI-7248: 48-CH DIO card
- PCI-7296: 96-CH DIO card
- cPCI-7248: 3U CompactPCI 48-CH DIO card
- cPCI-7249R: 3U CompactPCI 48-CH DIO card with Rear I/O

The 7248 series products are multi-function digital I/O boards used for industrial PC with PCI bus or CompactPCI bus. The cards are Plug and Play therefore it is not necessary to set any jumper for configuration of I/O address or interrupt resources.

PCI-7248 and PCI-7296 are 48-channel and 96-channel DIO cards respectively. PCI-7224 is a reduced version of PCI-7248, where all functions are exactly the same as PCI-7248 with the exception that there are only 24 channels. The cPCI-7248 is the 3U CompactPCI version of the PCI-7248. The software is fully compatible with the PCI-7248. In this manual, if the specifications for PCI-7224 and cPCI-7248’s are not specified, they are the same as that of PCI-7248.

The PCI-7248 emulates two industry standard 8255 Programmable Peripheral Interface (PPI) chips operated under mode zero configuration. The PCI-7296 emulates four PPI chips. These two cards are compatible not only on hardware connectors but also on software programming.

The cPCI-7248 is the CompactPCI version of PCI-7248. All the functions are exactly the same as PCI-7248. The cPCI-7249 is an extended version of cPCI-7248, with an additional latch register and rear I/O capability.

Every PPI connector has 3 ports: PA, PB, and PC. The PC can also be subdivided into 2 nibble-wide (4-bit) ports - PC Upper and PC Low. Each connector corresponds to one PPI chip with 24 DIO points. PCI-7224/7248/7296 are equipped with one, two, and four 50-pin male ribbon connectors respectively. The cPCI-7248 is equipped with one 100-pin SCSI-type connector.
1.1 Features

The 7248/96 series products provide the following advanced features:

1.1.1 Digital I/O Ports
- 24/48/96 TTL/DTL compatible with digital I/O lines
- Emulates industry standard mode 0 of 8255 PPI
- Buffered circuits for higher driving
- Direct interface with OPTO-22 compatible I/O module
- Output status read-back

1.1.2 Timer/Counter and Interrupt System
- A 32 bits timer to generate watchdog timer interrupt
- A 16 bits event counter to generate event interrupt
- Programmable interrupt source
- Dual interrupt system

1.1.3 Miscellaneous
- Provide 12V and 5V power supply on OPTO-22 connectors
- On board reset-able fuses to protect power supply for external devices

1.2 Applications
- Programmable mixed digital input & output
- Industrial monitoring and control
- Digital I/O control
- Contact closure, switch/keyboard monitoring
- Connects with OPTO-22 compatible modules
- Useful with A/D and D/A to implement a data acquisition and control system
## 1.3 Specifications

| I/O channels        | 24-bit for PCI-7224  
|                     | 48-bit for PCI-7248  
|                     | 96-bit for PCI-7296  
|                     | 48-bit for cPCI-7248 and cPCI-7249R |
| Digital Input Signal| Logic High Voltage: 2.0 V to 5.25V  
|                     | Logic Low Voltage: 0.0 V to 0.80V  
|                     | Logic High Current: 20.0 uA  
|                     | Logic Low Current: -0.2 mA  
| Digital Output Signal (Port A & B) | Logic High Voltage: Minimum 2.4 V  
|                     | Logic Low Voltage: Maximum 0.5V  
|                     | Logic High Current: -2.6 mA  
|                     | Logic Low Current: 24.0 mA  
| Digital Output Signal (Port C) | Logic High Voltage: Minimum 2.4 V  
|                     | Logic Low Voltage: Maximum 0.4V  
|                     | Logic High Current: -15.0 mA  
|                     | Logic Low Current: 24.0 mA  
| Operating Temperature | 0 °C - 60°C  
| Storage Temperature | -20°C - 80°C  
| Humidity | 5% - 95% non-condensing  
| I/O Connectors | 50-pin male ribbon cable connectors for PCI-7224/7248/7296 or  
|                     | 100-pin SCSI-type connectors for cPCI-7248  
| Bus | PCI bus for PCI-7224/7248/7296  
|                     | 3U 32-bit CompactPCI Bus for cPCI-7248  
| Power Consumption (without external devices) | PCI-7224:  
|                     | 330mA @ 5VDC (Typical)  
|                     | 350mA @ 5VDC (Maximum)  
|                     | PCI-7248:  
|                     | 500mA @ 5VDC (Typical)  
|                     | 540mA @ 5VDC (Maximum)  
|                     | PCI-7296:  
|                     | 860mA @ 5VDC (Typical)  
|                     | 940mA @ 5VDC (Maximum)  
|                     | cPCI-7248:  
|                     | 470mA @ 5VDC (Typical)  
|                     | 560mA @ 5VDC (Maximum)  
|                     | cPCI-7249R:  
|                     | 700mA @ 5VDC (Typical)  
| Transfer Rate | 500 K bytes/s (Maximum)  
| Transfer Rate | 500 K bytes/s (Maximum)  
<p>|</p>
<table>
<thead>
<tr>
<th>PCB Dimension</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PCI-7224:</td>
<td>148mm x 102mm</td>
</tr>
<tr>
<td>PCI-7248:</td>
<td>148mm x 102mm</td>
</tr>
<tr>
<td>PCI-7296:</td>
<td>16mm x 102mm</td>
</tr>
<tr>
<td>cPCI-7248:</td>
<td>3U CompactPCI form factor</td>
</tr>
<tr>
<td>cPCI-7249R:</td>
<td>3U CompactPCI form factor, J2 with Rear I/O option</td>
</tr>
</tbody>
</table>
1.4 Software Support

ADLINK provides versatile software drivers and packages for different approaches to building systems. We not only provide programming libraries such as DLL for many Windows systems, but also provide drivers for many software packages such as LabVIEW®, HP VEE™, DASYLab™, InTouch™, InControl™, ISaGRAF™, and many more.

All the software options are included in the ADLINK CD. The non-free software drivers are protected by serial licensed codes. Without the software serial number, the user can still install them and run the demo version for two hours for demonstration purpose. Please contact your dealer to purchase the formal license serial code.

1.4.1 Programming Library

For customers who are writing their own programs, we provide function libraries for many different operating systems, including:

- **DOS Library**: Borland C/C++ and Microsoft C++, the functions descriptions are included in this user’s guide.
- **Windows 95 DLL**: For VB, VC++, Delphi, BC5, the functions descriptions are included in this user’s guide.
- **PCIS-DASK**: Include device drivers and DLL for Windows 98, Windows NT and Windows 2000. DLL is binary compatible across Windows 98, Windows NT and Windows 2000. In other words, all applications developed with PCIS-DASK are compatible across Windows 98, Windows NT and Windows 2000. The developing environment can be VB, VC++, Delphi, BC5, or any Windows programming language that allows calls to a DLL. The user’s guide and function reference manual of PCIS-DASK are in the CD. Please refer the PDF manual files under the following directory: \Manual_PDF\Software\PCIS-DASK
- **PCIS-DASK/X**: Include device drivers and shared library for Linux. The developing environment can be Gnu C/C++ or any programming language that allows linking to a shared library. The user's guide and function reference manual of PCIS-DASK/X are in the CD. (\Manual_PDF\Software\PCIS-DASK-X.)
- **The above software drivers are shipped with the board. Please refer to the “Software Installation Guide” to install these drivers.**
1.4.2 PCIS-LVIEW: LabVIEW® Driver

PCIS-LVIEW contains the VIs, which are used to interface with NI’s LabVIEW® software package. The PCIS-LVIEW supports Windows 95/98/NT/2000. The LabVIEW® drivers are shipped free of charge with the board. You can install and use them without license. For more detailed information about PCIS-LVIEW, please refer to the user’s guide in the CD. (\Manual_PDF\Software\PCIS-LVIEW)

1.4.3 PCIS-VEE: HP-VEE Driver

The PCIS-VEE includes user objects, which are used to interface with HP VEE software package. PCIS-VEE supports Windows 95/98/NT. The HP-VEE drivers are shipped free of charge with the board. You can install and use them without license. For detail information about PCIS-VEE, please refer to the user’s guide in the CD. (\Manual_PDF\Software\PCIS-VEE)

1.4.4 DAQBench™: ActiveX Controls

We suggest the customers who are familiar with ActiveX controls and VB/VC++ programming to use the DAQBench™ ActiveX Control components library for developing applications. The DAQBench™ is designed under Windows NT/98. For more detailed information about DAQBench, please refer to the user’s guide in the CD. (\Manual_PDF\Software\DAQBench\DAQBench Manual.PDF)

1.4.5 DASYLab™ PRO

DASYLab is an easy-to-use software package, providing easy to setup instrument functions such as FFT analysis. Please contact ADLINK Technology to obtain DASYLab PRO, which includes DASYLab and ADLink hardware drivers.

1.4.6 PCIS-DDE: DDE Server and InTouch™

DDE stands for Dynamic Data Exchange specifications. The PCIS-DDE includes the PCI cards’ DDE server. The PCIS-DDE server is included in the ADLINK CD. A license is required. The DDE server can be used in conjunction with any DDE client under Windows NT.
1.4.7 PCIS-ISG: ISaGRAF™ driver
The ISaGRAF WorkBench is an IEC1131-3 SoftPLC control program development environment. The PCIS-ISG includes ADLink products’ target drivers for ISaGRAF under Windows NT environment. The PCIS-ISG is included in the ADLINK CD. A license is required.

1.4.8 PCIS-ICL: InControl™ Driver
PCIS-ICL is the InControl driver which supports the Windows NT. The PCIS-ICL is included in the ADLINK CD. A license is required.

1.4.9 PCIS-OPC: OPC Server
PCIS-OPC is an OPC Server, which can be linked with OPC clients. There are now many software packages on the market that can provide OPC clients. PCIS-OPC supports Windows NT and a license is required.
This chapter describes how to install the 7248/96 series products. Please take note of the packing and unpacking instructions as described below:

- Check what you have (section 2.1)
- Unpacking (section 2.2)
- Check the PCB (section 2.3)
- Hardware installation (section 2.4)
- Device Installation for Windows System (section 2.5)
- Connector pin assignment (section 2.6)
- Jumpers setup (section 2.7)
- Termination boards connection (section 2.8)

### 2.1 What You Have

In addition to this *User's Manual*, the package also includes:

- The PCI or CompactPCI board
- ADLINK CD
- Software Installation Guide
- In the cPCI-7249R box, a rear I/O transition board DB-100RU is included

If any of these items are missing or damaged, please contact the dealer from whom you purchased the product. Keep the shipping materials and box in case you want to ship or store the product in the future.
2.2 Unpacking

Your card contains sensitive electronic components that can be easily damaged by static electricity.

The card should be placed on a grounded anti-static mat. The operator should wear an anti-static wristband, grounded at the same point as the anti-static mat.

Inspect the card module carton for obvious damage. Shipping and handling may cause damage to your module. Ensure there is no shipping and handling damage on the module before processing.

After opening the card module carton, remove the system module and place only on a grounded anti-static surface with components side up.

Again inspect the module for damages. Press down on all the socketed IC’s to make sure they are properly seated. Only do this with the module placed on a firm flat surface.

**Note:** DO NOT APPLY POWER TO THE CARD IF IT HAS BEEN DAMAGED.

*You are now ready to install your 7248/96 series product.*
2.3 PCB Layout

2.3.1 PCI-7248/7224 PCB Layout

Figure 2.3.1 PCI-7248/7224 PCB Layout

2.3.2 PCI-7296 PCB Layout

Figure 2.3.2 PCI-7296 PCB Layout
2.3.3 cPCI-7248 PCB Layout

![Figure 2.3.3 cPCI-7248 PCB Layout](image)

2.3.4 cPCI-7249R PCB Layout

![Figure 2.3.4 cPCI-7249R Layout](image)
2.4 Hardware Installation

**PCI configuration**

The PCI cards (or CompactPCI cards) are equipped with Plug and Play PCI controller, it can request base addresses and interrupt according to PCI standard. The system BIOS will install the system resources based on the PCI cards' configuration registers and system parameters (which are set by system BIOS). Interrupt assignment and memory usage (I/O port locations) of the PCI cards can be assigned by system BIOS only. These system resource assignments are done on an individual board basis. It is not recommended to assign the system resource by any other methods.

**PCI slot selection**

The PCI card can be inserted into any PCI slot without any configuration for system resource. Please note that the PCI system board and slot must provide bus-mastering capability to operate this board well.

**PCI Installation Procedures**

1. Turn off your computer.
2. Turn off all peripherals (printer, modem, monitor, etc.) connected to your computer.
3. Remove the cover from your computer.
4. Set up jumpers on the PCI or CompactPCI card.
5. Select the 32-bit PCI slot. PCI slots are shorter than the ISA or EISA slots and are usually white or ivory.
6. Before handling the PCI cards, discharge any static buildup on your body by touching the metal case of the computer. Hold the edge and do not touch the components.
7. Position the board into the PCI slot you have selected.
8. Secure the card in place at the rear panel of the system.
CompactPCI Installation Procedures

1. Read through this manual, and set the jumpers according to your application.
2. Turn off your computer and turn off all accessories connected to computer.
3. Remove the slot cover from the CompactPCI.
4. Select a 32-bit CompactPCI slot on the back plane for cPCI-7248 board. Select a 32-bit CompactPCI slot with rear I/O extension for cPCI-7249R.
5. Before handling the boards, discharge any static buildup on your body by touching the metal case of the computer. Hold the edge and do not touch the components.
6. Position the board into the slot you selected.
7. Secure the card on the back plane by using the screw removed from the slot.
8. Install the rear I/O transition board for cPCI-7249R.

2.5 Device Installation for Windows Systems

Once Windows 95/98/2000 has started, the Plug and Play function of Windows system will find the new NuDAQ/NuIPC cards. If this is the first time you are installing NuDAQ/NuIPC cards on your Windows system, you will be informed to input the device information source. Please refer to the “Software Installation Guide” for the steps of installing the device.
2.6 Connector Pin Assignment

2.6.1 PCI-7224/7248/7296 Pin Assignment

The I/O ports of PCI-7224/7248/7296 emulate the mode 0 configuration of the 8255 general purpose programmable peripheral interface. The cards come equipped with 50-pin male IDC connectors that interface with OPTO-22. Figure 2.4 shows the circuits and pinout of PCI-7224/7248/7296's connectors (CN1-CN4).

Figure 2.6.1 Pin assignments and power signals of PCI-7224/7248/7296

The DIO pin names are specified as PnXb, where

- \( n \) : means the connector reference number \( n=1-4 \).
- \( X \) : means the port name, \( X=A, B \) or \( C \)
- \( b \) : means the bit number of a port, \( b=0-7 \)

For example, P1C4 means bit 4 of port C on connector CN1.

Note:

1. The pinout of the CN1-CN4 connectors are identical.
2. The power supply pins are protected by resettable fuses. Refer to section 4.4 for details on power supply.
2.6.2 cPCI-7248 Pin Assignment

The cPCI-7248 is equipped with a SCSI-type 100-pin connector. The pin assignment is described in Figure 2.6.2.

![Pinout Diagram]

**Figure 2.6.2 Pinout and power signals of cPCI-7248**

The DIO pin names are specified as PnXb, where

\[ \begin{align*}
    n & : \text{means the connector reference number } n=1-2. \\
    X & : \text{means the port name, } X=A, B \text{ or } C \\
    b & : \text{means the bit number of a port, } b=0-7
\end{align*} \]

For example, P1C4 means bit 4 of port C on connector CN1.

**Note:**

1. The pinout of the CN1-CN4 connectors are identical.
2. The power supply pins are protected by resettable fuses. Refer to section 4.4 for details on power supply.
2.6.3 cPCI-7249R Pin Assignment

The I/O ports of cPCI-7249R emulate the mode 0 configuration of the 8255 general purpose programmable peripheral interface. This card comes equipped with a SCSI-100 Pin connector. The cPCI-7249R supports a R7249 daughter board for rear I/O and includes two OPTO-22 connectors as well as a SCSI-100 connector.

The DIO pin names are specified as PnXb, where

- n : means the connector reference number n=1-2
- X : means the port name, X= A, B or C
- b : means the bit number of a port, b=0-7

For example, P1C4 means bit 4 of port C on connector CN1

- **EXTCLK**: External edge trigger signal for latch digital input data
- **EVENT**: External clock source for counter 0
- **V5V**: Onboard un-regulated 5V power supply output
- **+12V**: Onboard un-regulated +12V power supply output
2.6.4 R7249 OPTO-22 Connectors

The cPCI-7249R’s rear I/O transition board R7249 is equipped with two 50 pin male IDC connectors: XCN2, XCN3 that interface with OPTO-22.

![Figure 2.3 R7249 OPTO-22 Connectors Pin Assignment](image-url)

Note: The power supply pins are protected by resettable fuses.

Refer to section 3.7 for details of the power supply.
2.7 Jumpers Description

The 7248/96 DIO cards are Plug and Play, thus it is not necessary to set up the card configurations to fit the computer system. However, to fit different users’ versatile operating environment, there are still a few jumpers to set the power-on status of ports and the usage of the +12V output pins.

2.7.1 Power on Status of Ports

For every port on the 7248/96 cards, the power-on status is set as input therefore, the voltage could be pulled high, pulled low, or floating. It is dependent on the jumper setting. Table 2.1 lists the reference number of the jumpers and the corresponding port names.

<table>
<thead>
<tr>
<th>Jumper</th>
<th>Port Name</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>JA1</td>
<td>P1A (Port A of CN1)</td>
<td>for PCI-7224/7248/7296, cPCI-7248</td>
</tr>
<tr>
<td>JB1</td>
<td>P1B (Port B of CN1)</td>
<td>for PCI-7224/7248/7296, cPCI-7248</td>
</tr>
<tr>
<td>JC1</td>
<td>P1C (Port C of CN1)</td>
<td>for PCI-7224/7248/7296, cPCI-7248</td>
</tr>
<tr>
<td>JA2</td>
<td>P2A (Port A of CN2)</td>
<td>for PCI-7248/96, cPCI-7248</td>
</tr>
<tr>
<td>JB2</td>
<td>P2B (Port B of CN2)</td>
<td>for PCI-7248/96, cPCI-7248</td>
</tr>
<tr>
<td>JC2</td>
<td>P2C (Port C of CN2)</td>
<td>for PCI-7248/96, cPCI-7248</td>
</tr>
<tr>
<td>JA3</td>
<td>P3A (Port A of CN3)</td>
<td>for PCI-7296</td>
</tr>
<tr>
<td>JB3</td>
<td>P3B (Port B of CN3)</td>
<td>for PCI-7296</td>
</tr>
<tr>
<td>JC3</td>
<td>P3C (Port C of CN3)</td>
<td>for PCI-7296</td>
</tr>
<tr>
<td>JA4</td>
<td>P4A (Port A of CN4)</td>
<td>for PCI-7296</td>
</tr>
<tr>
<td>JB4</td>
<td>P4B (Port B of CN4)</td>
<td>for PCI-7296</td>
</tr>
<tr>
<td>JC4</td>
<td>P4C (Port C of CN4)</td>
<td>for PCI-7296</td>
</tr>
</tbody>
</table>

Table 2.1 Jumpers and Port names list

The physical meaning of all the jumpers are identical. The power-on status of each port can be set independently. The default is to pull all signals high. The following diagram use JA1 as an example to show the possible configurations.

1. Port A of CN1 are power-on pulled high. ‘H’

2. Port A of CN1 are power-on pulled low. ‘L’

3. Port A of CN1 are power-on floating. (The jumper is removed)
2.7.2 12V Power Supply Configuration
The pin 2 and pin 4 of the CN1-CN4 50-pin OPTO-22 connectors can be configured as 12V power supply or ground. Please refer to Figure 2.4 for the 12 volts power supply position. JP1-JP4 of 12V power are for CN1-CN4 respectively. Connections with ground are set as default. The following diagram shows the setting of JP2, connecting pin 2 and pin 4 of CN2 to ground.

![Diagram of 12V Power Supply Configuration](image)

2.8 Termination Boards Connection
There are many termination boards that can be connected with PCI-7248/7296 for isolated I/O expansion.

1. TB-24R, DIN-24R
The TB-24R and DIN-24R provide 24 Form C relays for digital output control.

![Diagram of TB-24R and DIN-24R](image)

2. TB-24 P, DIN-24P
The TB-24P and DIN-24P provide 24 opto-isolated digital input channels.

![Diagram of TB-24P and DIN-24P](image)
3. **TB-16P8R**
The TB-16P8R provides 16 opto-isolated digital input channels and 8 relay outputs.

![Diagram of TB-16P8R](image)

4. **TB-24, DIN-50S**
TB-24 and DIN-50S are termination boards with 50 pin ribbon connector. They are used for general-purpose applications.

5. **DIN-100S**
DIN-100S is equipped with 100-pin SCSI-type connector. It can be connected with cPCI-7248 and cPCI-7249R.
3

Registers Format

The detailed description of the registers format are specified in this chapter. This information is useful for the programmers who wish to handle the card by low-level programming. However, it is strongly recommended that the user has a more in depth understanding of the PCI interface before starting any low-level programming. In addition, the contents of this chapter can help the user understand how to use the software driver to manipulate this card.

3.1 PCI Plug and Play Registers

This PCI card functions as a 32-bit PCI target device to any master on the PCI bus. There are three types of registers: PCI Configuration Registers (PCR), Local Configuration Registers (LCR) and PCI-6308 registers.

The PCR, which is compliant to the PCI-bus specifications, is initialized and controlled by the Plug and Play PCI BIOS. Users can study the PCI BIOS specification to understand the operation of the PCR. Please contact PCISIG to acquire the specifications of the PCI interface.

The PCI bus controller PCI-9050 is provided by PLX Technology, Inc. (www.plxtech.com). For more detailed information on LCR, please visit the PLX technology website to download the relative information. Users are not required to understand the details of the LCR if they are using the software library. The PCI Plug and Play BIOS assigns the base address of the LCR. The assigned address is located at offset 14h of PCR.

The PCI-6308 registers are shown in the next section. The base address, which is also assigned by the PCI Plug and Play BIOS, is located at offset 18h of PCR. Therefore, users can read the 18h of PCR to find the base address by using the BIOS function call.

Please do not try to modify the base address and interrupt which are assigned by the PCI Plug and Play BIOS as it may cause resource conflict in your system.
### 3.2 I/O Address Map

All the 724X registers are 8 bits. The users can access these registers only by 8 bits I/O instructions. The following table shows the registers map, including descriptions and their offset addresses relative to the base address. Please refer to chapter 4 of this manual for a more detailed description on the operation of different registers.

<table>
<thead>
<tr>
<th>Offset</th>
<th>Write</th>
<th>Read</th>
<th>Boards</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x00</td>
<td>P1A</td>
<td>P1A</td>
<td>PCI-7224</td>
</tr>
<tr>
<td>0x01</td>
<td>P1B</td>
<td>P1B</td>
<td>PCI-7248</td>
</tr>
<tr>
<td>0x02</td>
<td>P1C</td>
<td>P1C</td>
<td>PCI-7296</td>
</tr>
<tr>
<td>0x03</td>
<td>P1Ctrl</td>
<td>Not used</td>
<td>cPCI-7248/49R</td>
</tr>
<tr>
<td>0x04</td>
<td>P2A</td>
<td>P2A</td>
<td>PCI-7248</td>
</tr>
<tr>
<td>0x05</td>
<td>P2B</td>
<td>P2B</td>
<td>PCI-7296</td>
</tr>
<tr>
<td>0x06</td>
<td>P2C</td>
<td>P2C</td>
<td>cPCI-7248/49R</td>
</tr>
<tr>
<td>0x07</td>
<td>P2Ctrl</td>
<td>Not used</td>
<td></td>
</tr>
<tr>
<td>0x08</td>
<td>P3A</td>
<td>P3A</td>
<td>PCI-7296 (1)</td>
</tr>
<tr>
<td>0x09</td>
<td>P3B</td>
<td>P3B</td>
<td>cPCI-7249R Only</td>
</tr>
<tr>
<td>0x0A</td>
<td>P3C</td>
<td>P3C</td>
<td>PCI-7248 (1)</td>
</tr>
<tr>
<td>0x0B</td>
<td>P3Ctrl</td>
<td>Not used</td>
<td>Not used</td>
</tr>
<tr>
<td>0x0C</td>
<td>P4A</td>
<td>P4A</td>
<td>PCI-7248</td>
</tr>
<tr>
<td>0x0D</td>
<td>P4B</td>
<td>P4B</td>
<td>PCI-7296</td>
</tr>
<tr>
<td>0x0E</td>
<td>P4C</td>
<td>P4C</td>
<td>cPCI-7248/49R</td>
</tr>
<tr>
<td>0x0F</td>
<td>P4Ctrl</td>
<td>Not used</td>
<td>Not used</td>
</tr>
<tr>
<td>0x10</td>
<td>Timer/Counter #0</td>
<td>Timer/Counter #0</td>
<td>PCI-7224</td>
</tr>
<tr>
<td>0x11</td>
<td>Timer/Counter #1</td>
<td>Timer/Counter #1</td>
<td>PCI-7248</td>
</tr>
<tr>
<td>0x12</td>
<td>Timer/Counter #2</td>
<td>Timer/Counter #2</td>
<td>PCI-7296</td>
</tr>
<tr>
<td>0x13</td>
<td>Timer/Counter Mode Control</td>
<td>Timer/Counter Mode Status</td>
<td>cPCI-7248/49R</td>
</tr>
<tr>
<td>0x20</td>
<td>ISC: Interrupt Source Control</td>
<td>Not used</td>
<td></td>
</tr>
<tr>
<td>0x30</td>
<td>Clear Interrupt</td>
<td>Not used</td>
<td></td>
</tr>
</tbody>
</table>
4.1 Digital I/O Ports

4.1.1 Introduction
The 7248/96 products can emulate one/two/four mode 0 configuration of 8255 programmable peripheral interface (PPI) chips. There are 24 DIO signals for every PPI.

4.1.2 8255 Mode 0
The basic functions of 8255 mode 0 are:
- Two 8-bit I/O ports-port A (PA) and port B (PB)
- Two nibble-wide (4-bit) ports C-PC upper and PC lower
- Each port can be used as either input or output
- Outputs are latched whereas inputs are buffered
- 16 different input/output configurations are available

4.1.3 Special Function of the DIO Signals
Two I/O signals (PC0 and PC3) of CN1 and CN2 can be used to generate hardware interrupt. Refer to section 4.3 for details on the interrupt control. In addition, the P1C4 signals can be used as input signal of event counter.
4.1.4 Digital I/O Port Programming

Users can write the digital output value to or read back the digital signal level from the PPI ports by using the software library. The port names are defined in Table 4.1. These port names are used throughout this manual as well as in the software library. The programming for PCI-7224/7248/7296 and cPCI-7248 are fully compatible.

<table>
<thead>
<tr>
<th>Connector Numbers</th>
<th>CN1</th>
<th>CN2</th>
<th>CN3</th>
<th>CN4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Port Names</td>
<td>P1A</td>
<td>P2A</td>
<td>P3A</td>
<td>P4A</td>
</tr>
<tr>
<td></td>
<td>P1B</td>
<td>P2B</td>
<td>P3B</td>
<td>P4B</td>
</tr>
<tr>
<td></td>
<td>P1C</td>
<td>P2C</td>
<td>P3C</td>
<td>P4C</td>
</tr>
<tr>
<td></td>
<td>P1CTRL</td>
<td>P2CTRL</td>
<td>P3CTRL</td>
<td>P4CTRL</td>
</tr>
</tbody>
</table>

Table 4.1 I/O Port Names

There are four ports on every 8255 PPI, including ports A, B, C, and the control port. PA, PB, and PC could be written or read but the control port is write only. Refer to chapter 5 for more details on the programming of DIO ports.

4.1.5 Control Word

The control word written in the control port is used to setup PA, PB, and PC as input or output port. Figure 4.1 shows the format of the control word. Table 4.2 shows the 16 possible control words and the respective I/O configurations.

D7 D6 D5 D4 D3 D2 D1 D0

\[
\begin{array}{cccc}
X & X & X & ? \quad ? \\
\end{array}
\]

1/0 : Input/Output of Port C low nibble
1/0 : Input/Output of Port B
1/0 : Input/Output of Port C high nibble
1/0 : Input/Output of Port A
X : don’t care

Figure 4.1 Control Word Format
Control Word | D4 | D3 | D2 | D1 | D0 | PORT A | PORT C UPPER | PORT B | PORT C LOWER
--- | --- | --- | --- | --- | --- | --- | --- | --- | ---
00H | 0 | 0 | 0 | 0 | 0 | O/P | O/P | O/P | O/P
01H | 0 | 0 | 0 | 1 | 0 | O/P | O/P | O/P | I/P
02H | 0 | 0 | 1 | 0 | 0 | O/P | O/P | I/P | O/P
03H | 0 | 0 | 1 | 1 | 0 | O/P | O/P | I/P | I/P
08H | 0 | 1 | 0 | 0 | 0 | O/P | I/P | O/P | O/P
09H | 0 | 1 | 0 | 1 | 0 | O/P | I/P | O/P | I/P
0AH | 0 | 1 | 1 | 0 | 0 | O/P | I/P | I/P | O/P
0BH | 0 | 1 | 1 | 1 | 0 | O/P | I/P | I/P | I/P
10H | 1 | 0 | 0 | 0 | 0 | I/P | O/P | O/P | O/P
11H | 1 | 0 | 0 | 1 | 0 | I/P | O/P | O/P | I/P
12H | 1 | 0 | 1 | 0 | 0 | I/P | O/P | I/P | O/P
13H | 1 | 0 | 1 | 1 | 0 | I/P | O/P | I/P | I/P
18H | 1 | 1 | 0 | 0 | 0 | I/P | I/P | O/P | O/P
19H | 1 | 1 | 0 | 1 | 0 | I/P | I/P | O/P | I/P
1AH | 1 | 1 | 1 | 0 | 0 | I/P | I/P | I/P | O/P
1BH* | 1 | 1 | 1 | 1 | 0 | I/P | I/P | I/P | I/P

Table 4.2 Summary of control word (D0-D4)
(* power on default configuration)

### 4.1.6 Power On Configuration

The default configuration after power on, hardware reset, or software reset sets all ports as input ports. Therefore, users do not need to worry about damaging external devices when system is on. In addition, the default signal level can be pulled high or pulled low by setting the jumpers. Refer to section 2.7 for setting the power on status of the DIO ports.

### 4.1.7 Note for Output Data

Be careful of the initial condition of digital output signals. If users set the control word as output port after power on, the previous uncertain output value will be put on the output pins immediately. Therefore, **BE SURE TO WRITE A SAFE VALUE TO THE PORTS BEFORE CONFIGURING THEM AS OUTPUT PORTS.**

### 4.1.8 Note for cPCI-7249R

The P1AE, P1BE, P1CE, P2AE, P2BE, and P2CE registers latch data from P1A, P1B, P1C, P2A, P2B, and P2C respectively. The latch signal is from Pin 99 of the 100-pin connector.
4.2 Timer/Counter Operation

4.2.1 Introduction
One 8254 programmable timer/counter chip is installed in the 7248/96 series. There are three counters in one 8254 chip and six possible operation modes for each counter. The block diagram of the timer/counter system is shown in Figure 4.2.

Timers #1 and #2 of the 8254 chip are cascaded as a 32-bit programmable timer. In the software library, Timers #1 and #2 are always set as mode 2 (rate generator).

In software library, counter #0 is used as an event counter that is, interrupt on terminal count of 8254 mode 0. Please refer to chapter 5 for programming the timer/counter functions.

4.2.2 General Purpose Timer/Counter
The counter 0 is a general purpose timer/counter for users applications. It can be used as an event counter, for measuring frequency, or other functions. The following Modes are provided by the 82C54 chip.

- Mode 0: Interrupt on Terminal Count
- Mode 1: Programmable One-Shot
- Mode 2: Rate Generator
- Mode 3: Square Wave Rate Generator
- Mode 4: Software Triggered Strobe
- Mode 5: Hardware Triggered Strobe

Figure 4.2 Timer/counter system of 7248/96 series.
The 8254 timer/counter IC occupies four I/O addresses. Users can refer to Tundra's or Intel®'s data sheet for a full description of the 8254 features. You can download the 8254 data sheet from the following web site:
http://support.intel.com/support/controllers/peripheral/231164.htm
or
http://www.tundra.com (for Tundra’s 82C54 datasheet)

4.2.2 Cascaded 32 Bits Timer
The input clock frequency of the cascaded timers is 2MHz. The output of the timer is sent to the interrupt circuit (refer to section 4.3). Therefore, the maximum and minimum watchdog timer interrupt frequency is (2MHz)/(2*2)=500KHz and (2MHz)/(65535*65535)= 0.000466Hz respectively.

4.2.3 Event Counter and Edge Control
The counter #0 of the 8254 chip can be used as an event counter. The input of counter #0 is PC4 of CN1 (P1C4). The counter clock trigger direction (H to L or L to H) is programmable. The gate control is always enabled. The output is sent to interrupt system which is named as event IRQ. If counter #0 is set as 8254 mode 0, the event counter IRQ will generate when the counter value is counting down to zero.

4.3 Interrupt Multiplexing

4.3.1 Architecture
The 7248/96 series products have a powerful and flexible interrupt multiplexing circuit which is suitable for many applications. The board is capable of accepting Dual Interrupts. The dual interrupt means that the hardware can generate two interrupt request signals at the same time and the software can service these two request signals by ISR. Note that the dual interrupts do not imply the card occupies two IRQ levels.

The two interrupt request signals (INT1 and INT2) come from digital input signals or the timer/counter output. An interrupt source multiplexer (MUX) is used to select the IRQ sources. Fig 4.3 shows the interrupt system.
4.3.2 IRQ Level Setting

There is only one IRQ level requested by this card, although it is a dual interrupt system. The motherboard circuits will transfer INTA# to one of the PC IRQ levels. The IRQ level is set by the PCI Plug and Play BIOS and saved in the PCI controller. Users can obtain the IRQ level setting in software library.

Fig 4.3 Dual Interrupt System of PCI-7224/7248/96

(*) Note: This interrupt is not available in PCI-7224

4.3.3 Note for Dual Interrupts

The PCI controller of PCI-7224/7248/96 can receive two hardware IRQ sources. However, a PCI controller can generate only one IRQ to PCI bus, the two IRQ sources must be distinguished by ISR of the application software if the two IRQ are all used.

The application software can use the “_72xx_Get_Irq_Status” function to distinguish which interrupt is inserted. After an ISR is completed, users must check if another IRQ is also asserted, then clear the current IRQ to allow room for the next IRQ.

The two IRQs are called INT1 and INT2. In PCI-7224/7248/7296, INT1 comes from P1C0, P1C3 or the event counter interrupt. INT2 comes from P2C0, P2C3 or the timer interrupt. However in PCI-7224, INT2 only comes from timer interrupt. The sources of INT1 and INT2 are selectable by the Interrupt Source Control (ISC) Register.
4.3.4 Interrupt Source Control

In ISC register (offset 0x20), there are four bits to control the IRQ sources of INT1 and INT2.

If the application requires only one IRQ, one of the IRQ sources can be disabled by software. If no IRQ source is required, both interrupts can be disabled. However, if the user only disables the IRQ source without changing the initial condition of the PCI controller, the PCI BIOS will still assign an IRQ level to the PCI card and occupy the PC resource.

It is not recommended to change the initial condition of the PCI card by the user’s own application software. If the user wishes to disable the IRQ level, they can use ADLINK’s utility ‘INIT7248.EXE’ or ‘INIT7296.EXE’ to change power on interrupt setting.

Table 4.3 shows the register format of the ISC (address offset 0x20). This register is write only. The four LSBs are used to control the source of INT1 and INT2.

<table>
<thead>
<tr>
<th>INT1</th>
<th>D3</th>
<th>D2</th>
<th>D1</th>
<th>D0</th>
<th>IRQ Source</th>
<th>IRQ Trigger Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Disable</td>
<td>X</td>
<td>X</td>
<td>0</td>
<td>0</td>
<td>INT1 disable</td>
<td>--</td>
</tr>
<tr>
<td>Mode 1</td>
<td>X</td>
<td>X</td>
<td>0</td>
<td>1</td>
<td>~P1C0</td>
<td>falling edge of P1C0</td>
</tr>
<tr>
<td>Mode 2</td>
<td>X</td>
<td>X</td>
<td>1</td>
<td>0</td>
<td>P1C0 OR ~P1C3</td>
<td>(see following)</td>
</tr>
<tr>
<td>Mode 3</td>
<td>X</td>
<td>X</td>
<td>1</td>
<td>1</td>
<td>Event Counter</td>
<td>Counter count down to 0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>INT2</th>
<th>D3</th>
<th>D2</th>
<th>D1</th>
<th>D0</th>
<th>IRQ Source</th>
<th>IRQ Trigger Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Disable</td>
<td>0</td>
<td>0</td>
<td>X</td>
<td>X</td>
<td>INT2 disable</td>
<td>--</td>
</tr>
<tr>
<td>Mode 1</td>
<td>0</td>
<td>1</td>
<td>X</td>
<td>X</td>
<td>~P2C0</td>
<td>falling edge of P2C0(*)</td>
</tr>
<tr>
<td>Mode 2</td>
<td>1</td>
<td>0</td>
<td>X</td>
<td>X</td>
<td>P2C0 OR ~P2C3</td>
<td>(see following) (*)</td>
</tr>
<tr>
<td>Mode 3</td>
<td>1</td>
<td>1</td>
<td>X</td>
<td>X</td>
<td>Timer Output</td>
<td>Timer count down to 0</td>
</tr>
</tbody>
</table>

Table 4.3 ISC register format

(*) Note: Not available on PCI-7224.

When the IRQ source is set as P1C0 OR P1C3, the IRQ trigger conditions are summarized in table 4.4.

<table>
<thead>
<tr>
<th>P1/2C0</th>
<th>P1/2C3</th>
<th>IRQ Trigger Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>High</td>
<td>X</td>
<td>PC0=H disable all IRQ</td>
</tr>
<tr>
<td>X</td>
<td>Low</td>
<td>PC3=L disable all IRQ</td>
</tr>
<tr>
<td>Low</td>
<td>1-&gt;0</td>
<td>PC3 falling edge trigger when PC0=L</td>
</tr>
<tr>
<td>0-&gt;1</td>
<td>High</td>
<td>PC0 rising edge trigger when PC3=H</td>
</tr>
</tbody>
</table>

Table 4.4 IRQ Trigger conditions

Because P1/P2C0 and P1/P2C3 are external signals, the user can utilize the combination of the four signals to generate a proper IRQ.

4.4 12V and 5V Power Supply
The OPTO-22 compatible connectors provide external devices for +12V and +5V power supply. To avoid shorting or overloading the power supply, the **resettable fuses** are added on all the output power. Refer to Figure 2.6.1

The maximum current for 5V on every connector is 0.5A. If the load current is larger than this, the resistance of the resettable fuse will increase due to the rise in temperature. This in turn will cause the power supply to drop and reduce its current. If the shorting or overloading problem is eliminated, the fuse will return to its normal condition. It is not necessary to repair or re-install the fuse.

The maximum current for 12V on all four connectors is also 0.5A. The action of the fuse is the same as that used for +5V power. The limitation is more restrictive than 5V power supply because the PCI bus can not provide large amounts of current.
This chapter describes the software library for operating this card. Only the functions in DOS library and Windows 95 DLL are described. A description of Windows 98/NT/2000 DLL functions can be found in the PCIS-DASK function reference manual located in the ADLINK CD. The functions of PCI-7248 can also be applied to PCI-7224 and cPCI-7248. Therefore in the following section, there are no special functions for PCI-7224 and cPCI-7248. The software driver of PCI-7224 is fully compatible with that of PCI-7248. In the sections below, all function calls referring to “7248” can also be applied to PCI-7224.

The function prototypes and some useful constants are defined in the header files LIB directory (DOS) and INCLUDE directory (Windows 95). For Windows 95 DLL, the developing environment can be Visual Basic 4.0 or above, Visual C/C++ 4.0 or above, Borland C++ 5.0 or above, Borland Delphi 2.x (32-bit) or above, or any Windows programming language that allows calls to a DLL. It provides the C/C++, VB, and Delphi include files.

### 5.1 Libraries Installation

Please refer to the “Software Installation Guide” for detailed installation information on the software libraries for DOS, Windows 95 DLL, or PCIS-DASK for Windows 98/NT/2000.

The device drivers and DLL functions of Windows 98/NT/2000 are included in the PCIS-DASK. Please refer to the PCIS-DASK user’s guide and function reference for detailed programming information in the ADLINK CD.
5.2 Programming Guide

5.2.1 Naming Convention
The functions of the NuDAQ PCI cards or NuIPC CompactPCI cards’ software driver use full-names to represent the functions' real meaning. The naming convention rules are:

In DOS Environment:
_{hardware_model}_{action_name}. e.g. _7248_Initial().
All functions in PCI-7248 driver are with 7248 as {hardware_model}. But they can be used by PCI-7248, PCI-7224 and cPCI-7248.
In order to recognize the difference between DOS library and Windows 95 library, a capital W is placed in the beginning of each function name of the Windows 95 DLL driver, e.g. W_7248_Initial().

5.2.2 Data Types
We defined some data type in Pci_7248.h (DOS) and Acl_pci.h (Windows 95). These data types are used by NuDAQ Cards’ library. ADLINK Technology Inc. suggests these data types to be used in the application programs. The following table gives the data type names and their range.

<table>
<thead>
<tr>
<th>Type Name</th>
<th>Description</th>
<th>Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>U8</td>
<td>8-bit ASCII character</td>
<td>0 to 255</td>
</tr>
<tr>
<td>I16</td>
<td>16-bit signed integer</td>
<td>-32768 to 32767</td>
</tr>
<tr>
<td>U16</td>
<td>16-bit unsigned integer</td>
<td>0 to 65535</td>
</tr>
<tr>
<td>I32</td>
<td>32-bit signed integer</td>
<td>-2147483648 to 2147483647</td>
</tr>
<tr>
<td>U32</td>
<td>32-bit single-precision floating-point</td>
<td>0 to 4294967295</td>
</tr>
<tr>
<td>F32</td>
<td>32-bit single-precision floating-point</td>
<td>-3.402823E38 to 3.402823E38</td>
</tr>
<tr>
<td>F64</td>
<td>64-bit double-precision floating-point</td>
<td>-1.797683134862315E308 to 1.797683134862315E309</td>
</tr>
<tr>
<td>Boolean</td>
<td>Boolean logic value</td>
<td>TRUE, FALSE</td>
</tr>
</tbody>
</table>
5.3 _7248/96_Initial

@ Description

The cards are initialized by this function. The software library could be used to control multiple cards.

@ Syntax

C/C++ (DOS)

U16 _7248_Initial (U16 *existCards, PCI_INFO *pciInfo)
U16 _7249_Initial (U16 *existCards, PCI_INFO *pciInfo)
U16 _7296_Initial (U16 *existCards, PCI_INFO *pciInfo)

C/C++ (Windows 95)

U16 W_7248_Initial (U16 *existCards, PCI_INFO *pciInfo)
U16 W_7249_Initial (U16 *existCards, PCI_INFO *pciInfo)
U16 W_7296_Initial (U16 *existCards, PCI_INFO *pciInfo)

Visual Basic (Windows 95)

W_7248_Initial (existCards As Integer, pciInfo As PCI_INFO) As Integer
W_7249_Initial (existCards As Integer, pciInfo As PCI_INFO) As Integer
W_7296_Initial (existCards As Integer, pciInfo As PCI_INFO) As Integer

@ Argument

existCards: The number of installed PCI-7224/7248/7296 cards. The returned value shows how many PCI-7224/7248/7296 cards are installed in your system.

pciinfo: It is a format to record the PCI bus Plug and Play initialization information which is decided by the Plug and Play BIOS. The PCI_INFO structure is defined in ACL_PCI.H. The base I/O address and the interrupt channel number are stored in pciinfo.

@ Return Code

ERR_NoError
ERR_PCIEBiosNot Exist
### 5.4 Digital Input

**@ Description**

This function is used to read 8-bit digital input data from digital input ports. You can get the 8-bit data from _7248_DI by using this function. The written data and read-in data is 8-bit data. Each data is mapped to a signal as shown below.

<table>
<thead>
<tr>
<th>D7</th>
<th>D6</th>
<th>D5</th>
<th>D4</th>
<th>D3</th>
<th>D2</th>
<th>D1</th>
<th>D0</th>
</tr>
</thead>
<tbody>
<tr>
<td>PA7</td>
<td>PA6</td>
<td>PA5</td>
<td>PA4</td>
<td>PA3</td>
<td>PA2</td>
<td>PA1</td>
<td>PA0</td>
</tr>
<tr>
<td>PB7</td>
<td>PB6</td>
<td>PB5</td>
<td>PB4</td>
<td>PB3</td>
<td>PB2</td>
<td>PB1</td>
<td>PB0</td>
</tr>
<tr>
<td>PC7</td>
<td>PC6</td>
<td>PC5</td>
<td>PC4</td>
<td>PC3</td>
<td>PC2</td>
<td>PC1</td>
<td>PC0</td>
</tr>
</tbody>
</table>

**@ Syntax**

**PCI-7224/7248, cPCI-7248, cPCI-7249R:**

**C/C++ (DOS)**

U16 _7248_DI (U16 cardNo, U16 channelPort, U8 *diData)

U16 _7249_DI (U16 cardNo, U16 channelPort, U8 *diData)

U16 _7296_DI (U16 cardNo, U16 channelPort, U8 *diData)

**C/C++ (Windows 95)**

U16 W_7248_DI (U16 cardNo, U16 channelPort, U16 *diData)

U16 W_7249_DI (U16 cardNo, U16 channelPort, U16 *diData)

U16 W_7296_DI (U16 cardNo, U16 channelPort, U16 *diData)

**Visual Basic (Windows 95)**

W_7248_DI (ByVal cardNo As Integer, ByVal channelPort As Integer, diData As Integer) As Integer

W_7249_DI (ByVal cardNo As Integer, ByVal channelPort As Integer, diData As Integer) As Integer

W_7296_DI (ByVal cardNo As Integer, ByVal channelPort As Integer, diData As Integer) As Integer

**@ Argument**

- **cardNo:** Card number to select board
- **diData:** Returns a 8-bit value from digital port.
- **channelPort:** Port of each channel

PCI_CH0_PA: CH1’s Port A
PCI_CH0_PB: CH1’s Port B
PCI_CH0_PC: CH1’s Port C
PCI_CH0_PCU: CH1’s Port C Upper Nibble
PCI_CH0_PCL: CH1’s Port C Low Nibble
PCI_CH1_PA: CH2’s Port A
PCI_CH1_PB: CH2’s Port B
PCI_CH1_PC: CH2’s Port C
PCI_CH1_PCU: CH2’s Port C Upper Nibble
PCI_CH1_PCL: CH2’s Port C Low Nibble
PCI_CH2_PA: CH2’s Port A
PCI_CH2_PPB: CH2’s Port B  
PCI_CH2_PPC: CH2’s Port C  
PCI_CH2PCODE: CH2’s Port C Upper Nibble  
PCI_CH2_PCL: CH2’s Port C Low Nibble  
PCI_CH3_PA: CH3’s Port A  
PCI_CH3_PPB: CH3’s Port B  
PCI_CH3_PPC: CH3’s Port C  
PCI_CH3PCODE: CH3’s Port C Upper Nibble  
PCI_CH3_PCL: CH3’s Port C Low Nibble  
PCI_CH0_PAE: CH1’s Port A uses External Latch  
PCI_CH0_PBE: CH1’s Port B uses External Latch  
PCI_CH0_PCE: CH1’s Port C uses External Latch  
PCI_CH1_PAE: CH2’s Port A uses External Latch  
PCI_CH1_PBE: CH2’s Port B uses External Latch  
PCI_CH1_PCE: CH2’s Port C uses External Latch

Note: 1. CH2 and CH3 are only available for PCI-7296.  
2. Only CH0 is available for PCI-7224.

@ Return Code
ERR_NoError
5.5 Digital Output

@ Description
This function is used to write data to digital output ports.

@ Syntax

C/C++ (DOS)
- U16 _7248_DO (U16 cardNo, U16 channelPort, U8 doData)
- U16 _7249_DO (U16 cardNo, U16 channelPort, U8 doData)
- U16 _7296_DO (U16 cardNo, U16 channelPort, U8 doData)

C/C++ (Windows 95)
- U16 W_7248_DO (U16 cardNo, U16 channelPort, U16 doData)
- U16 W_7249_DO (U16 cardNo, U16 channelPort, U16 doData)
- U16 W_7296_DO (U16 cardNo, U16 channelPort, U16 doData)

Visual Basic (Windows 95)
- W_7248_DO (ByVal cardNo As Integer, ByVal channelPort As Integer, ByVal doData As Integer) As Integer
- W_7249_DO (ByVal cardNo As Integer, ByVal channelPort As Integer, ByVal doData As Integer) As Integer
- W_7296_DO (ByVal cardNo As Integer, ByVal channelPort As Integer, ByVal doData As Integer) As Integer

@ Argument
- cardNo: Card number to select board
- channelPort: The same as channelPort in section 5.5
- doData: Value will be written to digital output port

@ Return Code
- ERR_NoError
5.6 Configuration Port

@ Description
This function is used to configure the Input or Output of each Port. Each I/O Port of PCI-7224/7248/7296 is either input or output, so it has to be configured as input or output before I/O operations are applied.

@ Syntax

C/C++ (DOS)
U16 _7248_Config_Port (U16 cardNo, U16 channelPort, U16 direction);
U16 _7249_Config_Port (U16 cardNo, U16 channelPort, U16 direction);
U16 _7296_Config_Port (U16 cardNo, U16 channelPort, U16 direction);

C/C++ (Windows 95)
U16 W_7248_Config_Port (U16 cardNo, U16 channelPort, U16 direction);
U16 W_7249_Config_Port (U16 cardNo, U16 channelPort, U16 direction);
U16 W_7296_Config_Port (U16 cardNo, U16 channelPort, U16 direction);

Visual Basic (Windows 95)
W_7248_Config_Port (ByVal cardNo As Integer, ByVal channelPort As Integer, ByVal direction As Integer) As Integer
W_7249_Config_Port (ByVal cardNo As Integer, ByVal channelPort As Integer, ByVal direction As Integer) As Integer
W_7296_Config_Port (ByVal cardNo As Integer, ByVal channelPort As Integer, ByVal direction As Integer) As Integer

@ Argument
- cardNo: Card number to select board
- channelPort: The same as channelPort in section 5.5
- direction: Port I/O direction
- INPUT_PORT: The port is configure as INPUT
- OUTPUT_PORT: The port is configure as OUTPUT

@ Return Code
- ERR_NoError
5.7 Configuration Channel

@ Description

This function is used to configure the Input or Output of each Channel. Each I/O Port of PCI-7224/7248/7296 is either input or output, so it has to be configured as input or output before I/O operations are applied.

@ Syntax

C/C++ (DOS)

U16 _7248_Config_Channel (U16 cardNo, U16 channelNo,
   U16 ctrlValue)
U16 _7249_Config_Channel (U16 cardNo, U16 channelNo,
   U16 ctrlValue)
U16 _7296_Config_Channel (U16 cardNo, U16 channelNo,
   U16 ctrlValue)

C/C++ (Windows 95)

U16 W_7248_Config_Channel (U16 cardNo, U16 channelNo,
   U16 ctrlValue)
U16 W_7249_Config_Channel (U16 cardNo, U16 channelNo,
   U16 ctrlValue)
U16 W_7296_Config_Channel (U16 cardNo, U16 channelNo,
   U16 ctrlValue)

Visual Basic (Windows 95)

W_7248_Config_Channel (ByVal cardNo As Integer, ByVal
   channelNo As Integer, ByVal ctrlValue As Integer)
   As Integer
W_7249_Config_Channel (ByVal cardNo As Integer, ByVal
   channelNo As Integer, ByVal ctrlValue As Integer)
   As Integer
W_7296_Config_Channel (ByVal cardNo As Integer, ByVal
   channelNo As Integer, ByVal ctrlValue As Integer)
   As Integer

@ Argument

cardNo: Card number to select board
channelNo: PCI_CH0, PCI_CH1, PCI_CH2, or PCI_CH3
ctrlValue: The control value to set up PA, PB, and PC as input or output port.
<table>
<thead>
<tr>
<th>ctrlValue</th>
<th>Port A</th>
<th>Port CU</th>
<th>Port B</th>
<th>Port CL</th>
</tr>
</thead>
<tbody>
<tr>
<td>PORT_OOO</td>
<td>OUT</td>
<td>OUT</td>
<td>OUT</td>
<td>OUT</td>
</tr>
<tr>
<td>PORT_OO0I</td>
<td>OUT</td>
<td>OUT</td>
<td>OUT</td>
<td>IN</td>
</tr>
<tr>
<td>PORT_O01O</td>
<td>OUT</td>
<td>OUT</td>
<td>IN</td>
<td>OUT</td>
</tr>
<tr>
<td>PORT_O011</td>
<td>OUT</td>
<td>OUT</td>
<td>IN</td>
<td>IN</td>
</tr>
<tr>
<td>PORT_O10O</td>
<td>OUT</td>
<td>IN</td>
<td>OUT</td>
<td>OUT</td>
</tr>
<tr>
<td>PORT_O101</td>
<td>OUT</td>
<td>IN</td>
<td>OUT</td>
<td>IN</td>
</tr>
<tr>
<td>PORT_O110</td>
<td>OUT</td>
<td>IN</td>
<td>IN</td>
<td>OUT</td>
</tr>
<tr>
<td>PORT_O111</td>
<td>OUT</td>
<td>IN</td>
<td>IN</td>
<td>IN</td>
</tr>
<tr>
<td>PORT_I000</td>
<td>IN</td>
<td>OUT</td>
<td>OUT</td>
<td>OUT</td>
</tr>
<tr>
<td>PORT_I001</td>
<td>IN</td>
<td>OUT</td>
<td>OUT</td>
<td>IN</td>
</tr>
<tr>
<td>PORT_I010</td>
<td>IN</td>
<td>OUT</td>
<td>IN</td>
<td>OUT</td>
</tr>
<tr>
<td>PORT_I011</td>
<td>IN</td>
<td>OUT</td>
<td>IN</td>
<td>IN</td>
</tr>
<tr>
<td>PORT_I100</td>
<td>IN</td>
<td>IN</td>
<td>OUT</td>
<td>OUT</td>
</tr>
<tr>
<td>PORT_I101</td>
<td>IN</td>
<td>IN</td>
<td>OUT</td>
<td>IN</td>
</tr>
<tr>
<td>PORT_I110</td>
<td>IN</td>
<td>IN</td>
<td>IN</td>
<td>OUT</td>
</tr>
<tr>
<td>PORT_I111</td>
<td>IN</td>
<td>IN</td>
<td>IN</td>
<td>IN</td>
</tr>
</tbody>
</table>

The ctrlValue constants are defined in acl_pci.h and acl_pci.bas.

@ Return Code

ERR_NoError
5.8 Set Interrupt Control

@ Description
This function is used to set the interrupt configuration. The interrupt should be configured before the function starts.

@ Syntax
C/C++ (DOS)
void _7248_Set_INT_Control(U16 cardNo, U16 ctrlValue);
void _7249_Set_INT_Control(U16 cardNo, U16 ctrlValue);
void _7296_Set_INT_Control(U16 cardNo, U16 ctrlValue);

C/C++ (Windows 95)
void W_7248_Set_INT_Control(U16 cardNo, U16 ctrlValue);
void W_7249_Set_INT_Control(U16 cardNo, U16 ctrlValue);
void W_7296_Set_INT_Control(U16 cardNo, U16 ctrlValue);

Visual Basic (Windows 95)
W_7248_Set_INT_Control (ByVal cardNo As Integer, ByVal ctrlValue As Integer)
W_7249_Set_INT_Control (ByVal cardNo As Integer, ByVal ctrlValue As Integer)
W_7296_Set_INT_Control (ByVal cardNo As Integer, ByVal ctrlValue As Integer)

@ Argument
- cardNo: Card number to select board
- ctrlValue: The valid values of configuring the interrupt are

<table>
<thead>
<tr>
<th>INT1</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>0x00</td>
<td>disable</td>
<td></td>
</tr>
<tr>
<td>0x01</td>
<td>mode1</td>
<td></td>
</tr>
<tr>
<td>0x02</td>
<td>mode2</td>
<td></td>
</tr>
<tr>
<td>0x03</td>
<td>mode3</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>INT2</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>0x00</td>
<td>disable</td>
<td></td>
</tr>
<tr>
<td>0x04</td>
<td>mode1</td>
<td></td>
</tr>
<tr>
<td>0x08</td>
<td>mode2</td>
<td></td>
</tr>
<tr>
<td>0x0c</td>
<td>mode3</td>
<td></td>
</tr>
</tbody>
</table>

Please refer to the section 4.3.4 for detailed description.
5.9 Timer Start

@ Description
This function is used to set and start timer 0 of on-board timer 8254.

@ Syntax
C/C++ (DOS)
void _7248_Timer_Start(U16 cardNo, U16 timer0Mode, U16 c0)
void _7249_Timer_Start(U16 cardNo, U16 timer0Mode, U16 c0)
void _7296_Timer_Start(U16 cardNo, U16 timer0Mode, U16 c0)

C/C++ (Windows 95)
void W_7248_Timer_Start(U16 cardNo, U16 timer0Mode, U16 c0)
void W_7249_Timer_Start(U16 cardNo, U16 timer0Mode, U16 c0)
void W_7296_Timer_Start(U16 cardNo, U16 timer0Mode, U16 c0)

Visual Basic (Windows 95)
W_7248_Timer_Start(ByVal cardNo As Integer, ByVal timer0Mode As Integer, ByVal c0 As Integer)
W_7249_Timer_Start(ByVal cardNo As Integer, ByVal timer0Mode As Integer, ByVal c0 As Integer)
W_7296_Timer_Start(ByVal cardNo As Integer, ByVal timer0Mode As Integer, ByVal c0 As Integer);

@ Argument
  cardNo: Card number to select board
  timer0Mode: The six modes of 8254, please refer to section 4.2.2 for detailed description
  c0: The counter/timer value of time0.
5.10 Timer Read

@ Description
This function is used to read the current count of timer 0 of the on-board timer 8254.

@ Syntax
PCI-7224/7248, CPCI-7248:

C/C++ (DOS)
void _7248_Timer_Read(U16 cardNo, U16 *counterValue);
void _7249_Timer_Read(U16 cardNo, U16 *counterValue);
void _7296_Timer_Read(U16 cardNo, U16 *counterValue);

C/C++ (Windows 95)
void W_7248_Timer_Read(U16 cardNo, U16 *counterValue);
void W_7249_Timer_Read(U16 cardNo, U16 *counterValue);
void W_7296_Timer_Read(U16 cardNo, U16 *counterValue);

Visual Basic (Windows 95)
W_7248_Timer_Read(ByVal cardNo As Integer, counterValue As Integer);
W_7249_Timer_Read(ByVal cardNo As Integer, counterValue As Integer);
W_7296_Timer_Read (ByVal cardNo As Integer,
                  counterValue As Integer);

@ Argument
cardNo: card number to select board
counterValue: the value of the current count
5.11 Timer Stop

@ Description
This function is used to stop timer 0 of the on-board timer 8254.

@ Syntax
C/C++ (DOS)
void _7248_Timer_Stop(U16 cardNo, U16 *counterValue);
void _7249_Timer_Stop(U16 cardNo, U16 *counterValue);
void _7296_Timer_Stop(U16 cardNo, U16 *counterValue);

C/C++ (Windows 95)
void W_7248_Timer_Stop(U16 cardNo, U16 *counterValue);
void W_7249_Timer_Stop(U16 cardNo, U16 *counterValue);
void W_7296_Timer_Stop(U16 cardNo, U16 *counterValue);

Visual Basic (Windows 95)
W_7248_Timer_Stop (ByVal cardNo As Integer, counterValue As Integer);
W_7249_Timer_Stop (ByVal cardNo As Integer, counterValue As Integer);
W_7296_Timer_Stop (ByVal cardNo As Integer, counterValue As Integer);

@ Argument
  cardNo: Card number to select board
  counterValue: The value of current count when the counter is stopped
5.12 Cascaded Timer

@ Description
This function is used to set and start the cascaded timer 1 and timer 2 of on-board timer 8254.

@ Syntax

C/C++ (DOS)
void _7248_Cascaded_Timer(U16 cardNo, U16 c1, U16 c2);
void _7249_Cascaded_Timer(U16 cardNo, U16 c1, U16 c2);
void _7296_Cascaded_Timer(U16 cardNo, U16 c1, U16 c2);

C/C++ (Windows 95)
void W_7248_Cascaded_Timer(U16 cardNo, U16 c1, U16 c2);
void W_7249_Cascaded_Timer(U16 cardNo, U16 c1, U16 c2);
void W_7296_Cascaded_Timer(U16 cardNo, U16 c1, U16 c2);

Visual Basic (Windows 95)
W_7248_Cascaded_Timer (ByVal cardNo As Integer, ByVal c1 As Integer, ByVal c2 As Integer)
W_7249_Cascaded_Timer (ByVal cardNo As Integer, ByVal c1 As Integer, ByVal c2 As Integer)
W_7296_Cascaded_Timer (ByVal cardNo As Integer, ByVal c1 As Integer, ByVal c2 As Integer)

@ Argument

cardNo: Card number to select board
c1: The value of divisor used in timer 1
c2: The value of divisor used in timer 2
5.13 Get IRQ Status

@ Description
This function is used to read back the status of interrupt when interrupt is inserted.

@ Syntax
C/C++ (DOS)
void _7248_Get_IRQ_Status(U16 cardNo, U16 *int1Status, U16 *int2Status);
void _7249_Get_IRQ_Status(U16 cardNo, U16 *int1Status, U16 *int2Status);
void _7296_Get_IRQ_Status(U16 cardNo, U16 *int1Status, U16 *int2Status);

C/C++ (Windows 95)
void W_7248_Get_IRQ_Status(U16 cardNo, U16 *int1Status, U16 *int2Status);
void W_7249_Get_IRQ_Status(U16 cardNo, U16 *int1Status, U16 *int2Status);
void W_7296_Get_IRQ_Status(U16 cardNo, U16 *int1Status, U16 *int2Status);

Visual Basic (Windows 95)
W_7248_Get_IRQ_Status (ByVal cardNo As Integer, int1Status As Integer, int2Status As Integer);
W_7249_Get_IRQ_Status (ByVal cardNo As Integer, int1Status As Integer, int2Status As Integer);
W_7296_Get_IRQ_Status (ByVal cardNo As Integer, int1Status As Integer, int2Status As Integer);

@ Argument

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>cardNo:</td>
<td>Card number to select board</td>
</tr>
<tr>
<td>int1Status:</td>
<td>The status of INT1,</td>
</tr>
<tr>
<td></td>
<td>int1Status = 0 → no interrupt</td>
</tr>
<tr>
<td></td>
<td>int1Status = 1 → interrupt 1 inserted</td>
</tr>
<tr>
<td>int2Status:</td>
<td>The status of INT2</td>
</tr>
<tr>
<td></td>
<td>int2Status = 0 → no interrupt</td>
</tr>
<tr>
<td></td>
<td>int2Status = 1 → interrupt 2 inserted</td>
</tr>
</tbody>
</table>

Please refer to section 4.3 for a more detailed description.
5.14 Clear IRQ

@ Description
This function is used to clear the interrupt generated from the 7248/96 series.

@ Syntax

C/C++ (DOS)
void _7248_CLR_IRQ(U16 cardNo);
void _7249_CLR_IRQ(U16 cardNo);
void _7296_CLR_IRQ(U16 cardNo);

C/C++ (Windows 95)
void W_7248_CLR_IRQ(U16 cardNo);
void W_7249_CLR_IRQ(U16 cardNo);
void W_7296_CLR_IRQ(U16 cardNo);

Visual Basic (Windows 95)
W_7248_CLR_IRQ (ByVal cardNo As Integer);
W_7249_CLR_IRQ (ByVal cardNo As Integer);
W_7296_CLR_IRQ (ByVal cardNo As Integer);

@ Argument
cardNo: card number to select board

5.15 Software Reset

@ Description
This function is used to reset the I/O port configuration. After resetting PCI-7224/7248/7296, all ports will be set as input ports. Note that this function cannot re-start the PCI bus nor will all hardware settings be changed.

@ Syntax

PCI-7224/7248, CPCI-7248:

C/C++ (DOS)
U16 _7248_Software_Reset (U16 cardNo)
U16 _7249_Software_Reset (U16 cardNo)
U16 _7296_Software_Reset (U16 cardNo)

C/C++ (Windows 95)
U16 W_7248_Software_Reset (U16 cardNo)
U16 W_7249_Software_Reset (U16 cardNo)
U16 W_7296_Software_Reset (U16 cardNo)
Visual Basic (Windows 95)

W_7248_Software_Reset (ByVal cardNo As Integer) As Integer
W_7249_Software_Reset (ByVal cardNo As Integer) As Integer
W_7296_Software_Reset (ByVal cardNo As Integer) As Integer

@ Argument
cardNo: Card number which the DIO will be reset.

@ Return Code
ERR_NoError

5.16 Interrupt Start under Windows

@ Description
This function is only available in the Windows 95/98 driver. This function is used to initialize and start up the interrupt control. Please refer to section 4.3 for a detailed description of the interrupt system. After calling this function, every time an interrupt requests a signal to be generated, a software event is signaled. This implies that in the program, the user can use wait the operation to wait for the event. When the event is signaled, it means an interrupt is generated. Please refer to sample program 7248int.c or 7296int.c.

@ Syntax

C/C++ (Windows 95)
void W_7248_INT_Start (U16 cardNo, U16 c1, U16 c2, U16 ctrlValue, HANDLE *hIntEvent)
void W_7249_INT_Start (U16 cardNo, U16 c1, U16 c2, U16 ctrlValue, HANDLE *hIntEvent)
void W_7248_INT_Start (U16 cardNo, U16 c1, U16 c2, U16 ctrlValue, HANDLE *hIntEvent)

Visual Basic (Windows 95)
W_7248_INT_Start (ByVal cardNo As Integer, ByVal c1 As Integer, ByVal c2 As Integer, ByVal ctrlValue As Integer, hIntEvent As Long)
W_7249_INT_Start (ByVal cardNo As Integer, ByVal c1 As Integer, ByVal c2 As Integer, ByVal ctrlValue As Integer, hIntEvent As Long)
W_7296_INT_Start (ByVal cardNo As Integer, ByVal c1 As Integer, ByVal c2 As Integer, ByVal ctrlValue As Integer, hIntEvent As Long)
@ Argument

cardNo: Card number which the DIO will be reset.
c1: If the interrupt source is set as internal timer source, this value is the frequency divider of Timer#1.
c2: If the interrupt source is set as internal timer source, this value is the frequency divider of Timer#2.
ctrlValue: The value for INT mode setting. The value can be set for INT1 is INT1_OFF, INT1_P1C0, INT1_P1C3C0, or INT1_EVENT_IRQ. The value can be set for INT2 is INT2_OFF, INT2_P2C0, INT2_P2C3C0, or INT2_TIMER_IRQ. Please refer to section 3.3.4 for detailed description.
hIntEvent: The handle of the event for interrupt signals.

@ Return Code

ERR_NoError

5.17 Interrupt Stop under Windows

@ Description
This function is only available in Windows 95/98 driver. This function is used to disable the interrupt signal generation.

@ Syntax

C/C++ (Windows 95)
void W_7248_INT_Stop (U16 cardNo)
void W_7249_INT_Stop (U16 cardNo)
void W_7296_INT_Stop (U16 cardNo)

Visual Basic (Windows 95)
W_7248_INT_Stop (ByVal cardNo As Integer)
W_7249_INT_Stop (ByVal cardNo As Integer)
W_7296_INT_Stop (ByVal cardNo As Integer)

@ Argument
cardNo: The card number which the DIO will reset.

@ Return Code

ERR_NoError
Warranty Policy

Thank you for choosing ADLINK. To understand your rights and enjoy all the after-sales services we offer, please read the following carefully:

1. Before using ADLINK’s products please read the user manual and follow the instructions exactly.

2. When sending in damaged products for repair, please attach an RMA application form.

3. All ADLINK products come with a two-year guarantee, repaired free of charge.
   - The warranty period starts from the product’s shipment date from ADLINK’s factory.
   - Peripherals and third-party products not manufactured by ADLINK will be covered by the original manufacturers’ warranty.
   - End users requiring maintenance services should contact their local dealers. Local warranty conditions will depend on local dealers.

4. This warranty will not cover repair costs due to:
   a. Damage caused by not following instructions.
   b. Damage caused by carelessness on the users’ part during product transportation.
   c. Damage caused by fire, earthquakes, floods, lightening, pollution, other acts of God, and/or incorrect usage of voltage transformers.
   d. Damage caused by unsuitable storage environments (i.e. high temperatures, high humidity, or volatile chemicals.
   e. Damage caused by leakage of battery fluid.
   f. Damage from improper repair by unauthorized technicians.
   g. Products with altered and/or damaged serial numbers.
   h. Other categories not protected under our guarantees.

5. Customers are responsible for shipping costs to transport damaged products to our company or sales office.

6. To ensure the speed and quality of product repair, please download a RMA application form from our company website: www.adlinktech.com. Damaged products with attached RMA forms receive priority.

For further questions, please contact our FAE staff.
ADLINK: service@adlinktech.com