PCI-8136

General Purpose Multi-Function I/O Card

User's Guide

Recycled

©Copyright 2000 ADLINK Technology Inc.

All Rights Reserved.

Manual Rev. 1.00: October 23, 2000

The information in this document is subject to change without prior notice in order to improve reliability, design and function and does not represent a commitment on the part of the manufacturer.

In no event will the manufacturer be liable for direct, indirect, special, incidental, or consequential damages arising out of the use or inability to use the product or documentation, even if advised of the possibility of such damages.

This document contains proprietary information protected by copyright. All rights are reserved. No part of this manual may be reproduced by any mechanical, electronic, or other means in any form without prior written permission of the manufacturer.

Trademarks

NuDAQ, PCI-8136 are registered trademarks of ADLINK Technology Inc, MS-DOS & Windows 95 are registered trademarks of Microsoft Corporation., Borland C++ is a registered trademark of Borland International, Inc. Other product names mentioned herein are used for identification purposes only and may be trademarks and/or registered trademarks of their respective companies.

Getting service from ADLINK

Customer Satisfaction is always the most important thing for ADLINK Tech Inc. If you need any help or service, please contact us and get it.

ADLINK Technology Inc.			
Web Site	http://www.adlink.com.tw http://www.adlinktechnology.com		
Sales & Service	service@adlink.com.tw		
Technical Support	NuDAQ	nudaq@adlink.com.tw	
	NuDAM	nudam@adlink.com.tw	
	NuIPC	nuipc@adlink.com.tw	
	NuPRO	nupro@adlink.com.tw	
	Software	sw@adlink.com.tw	
TEL	+886-2-82265877	FAX	+886-2-82265717
Address	9F, No. 166, Jian Yi Road, Chungho City, Taipei, 235 Taiwan, R.O.C.		

Please inform or FAX us of your detailed information for a prompt, satisfactory and constant service.

Detailed Company Information			
Company/Organization			
Contact Person			
E-mail Address			
Address			
Country			
TEL		FAX	
Web Site			

Questions	
Product Model	
Environment to Use	☐ OS_____ ☐ Computer Brand ☐ M/B: ☐ CPU: ☐ Chipset: ☐ Bios: ☐ Video Card: ☐ Network Interface Card: ☐ Other:
Challenge Description	
Suggestions to ADLINK	

Table of Contents

How to Use This Guide

This manual is designed to help you to use the PCI-8136. The manual describes how to modify various settings on the PCI-8136 card 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 PCI-8136.
- Chapter 3, "Signal Connection", describes the connectors' pin assignment and how to connect the outside signal and devices with the PCI-8136.
- Chapter 4, "Operation Theorem", describes detail operations of the PCI-8136.
- Chapter 5, "Function Library", describes high-level programming interface in C/C++ language. It helps programmer to control PCI-8136 in high-level language style..

1

Introduction

The PCI-8136 is a multi function DAQ card. It contains 6-CH 12-bit AD for signal measurement and 6-CH 16-bit DA for precise voltage output. It also has six-channel 32-bit differential encoder counters and six pulse train generators. Both of them supports three kinds of differential pulse types: Out/Dir, A/B phase, CW/CCW. There are still 17 NPN type digital input channels and 6 open collector type digital output. All of these digital I/O are photo coupler isolated. PCI-8136 also has two master modules of remote I/O which supports one slave module each. Each slave module has 64 points input and output. The slave module is optional. PCI-8136 also has a 24-bit programmable timer for users to generate a constant timer interval.

PCI-8136 provides Windows DLL for users to develop their own applications more easily. It supports Windows 95/98/NT/2000 platform. The development environment for Windows can be Visual Basic , Visual C++, C++ Builder, Delphi. Under DOS environment, it only supports Borland C++ 3.1 because PCI-8136 is a PCI 32-bit interface card.

1.1 Features

The PCI-8136 provides **ADPIO** functions for users. The ADPIO stands for Analog/Digital/Pulse Input/Output. The following lists summarize the main features of the PCI-8136.

- 32-bit PCI-bus, plug and play
- 6 channels 16-bit analog output
- 6 channels 12-bit analog input
- 19 channels isolated digital input
- 7 channels open collector digital output
- Programmable interrupt sources
- 6 differential types 32-bit encoder counters
- 6 differential type pulse generators
- One 24-bit programmable timer
- Software supports maximum up to 4 cards
- Function library for DOS and Windows 95/98/NT/2000
- Compact, half size PCB

1.2 Specifications

◆ ***Analog Input***

- 6 differential input channels
- Input range: Voltage: ±10V
  Current: ±20 mA(manually soldering 124ohms resistor)
- 12-bit ADC with 1-bit non-linearity
- Input impedance : 10^{12} Ohms || 100pF (voltage mode
  124Ohms (current mode)
- Sampling rate: 133 KHz multiplexing

◆ ***Analog Output***

- 6 output channels
- Output range: bipolar, ±10V
- 16-bit DAC resolution, 14-bit accuracy guaranteed
- Settling time: 2 μ second
- Voltage output drive: 5mA max.

◆ ***Digital Input***

- 19 input channels for NPN type sensor
- Input impedance: 4.7K Ohms
- Max. Current: 20mA
- Isolated voltage: 2500V RMS
- Throughput: 10KHz (0.1ms)

◆ ***Digital Output***

- 7 Output channels
- Output type: Darlington transistor with open collector type (ULN2003A)
- Sink current: 90mA/Ch (typical)
  150mA/Ch (max.)
  500mA/total
- Isolated voltage 2500V RMS
- Throughput 10KHz(0.1ms)

◆ ***Pulse Input (Encoder Counter)***

- 6 differential input channels
- 32-bit counter for AB-phase, CW/CCW, Pulse/Direction modes
- 2500V RMS optical isolation
- Maximum pulse input frequency: 2MHz
- 32-bit encoder counter comparison

◆ ***Pulse Output (Pulse Generator):***

- 6 output channels with differential line drivers
- Pulse command type: CW/CCW, Pulse/Direction, A/B Phase
- Maximum pulse rate: 500KHz with 1 μ second pulse width.

◆ ***Timer:***

- One 24-bit programmable timer
- Base clock: 33MHz from PCI bus

◆ ***General Specifications***

- Connectors:
  - 100-pin SCSI-type connector
  - DB25 female connector
  - DB9 male connector
- Operating Temperature: 0° C ~ 50° C
- Storage Temperature: -20° C ~ 80° C
- Humidity: 5 ~ 85%, non-condensing
- Power Consumption:
  - Slot power supply(input): +5V DC ±5%, 900mA max.
  - External power supply(input): +24V DC ±5%, 500mA max.
  - External power supply(output): +5V DC ±5%, 500mA, max.
- Dimension: 164mm(L) X 98.4mm(H)

1.3 Software Supporting

For the customers who are going to write their own programs, we provide programming library for Borland C/C++ 3.1 under DOS and DLL for Windows 95/98/NT/2000. These function libraries are shipped with the board.

2

Installation

This chapter describes how to install the PCI-8136 hardware and software correctly. Please follow the following steps.

- Check what you have (section 2.1)
- Check the PCB (section 2.2)
- Install the hardware (section 2.3)
- Install the software driver (section 2.4)
- Understanding the connectors' pin assignments (the rest of the sections) and wiring the connections

2.1 What You Have

In addition to this *User's Guide*, the package includes the following items:

- PCI-8136 General Purpose Multi-I/O Card
- DB9 and DB25 Bracket
- External power cable for CN1
- 124O resistors * 6
- ADLINK All-in-one Compact Disc

If any item is missing or damaged, contact the dealer from whom you purchased the product. Save the shipping materials and carton in case you want to ship or store the product in the future.

2.2 Outline Drawing

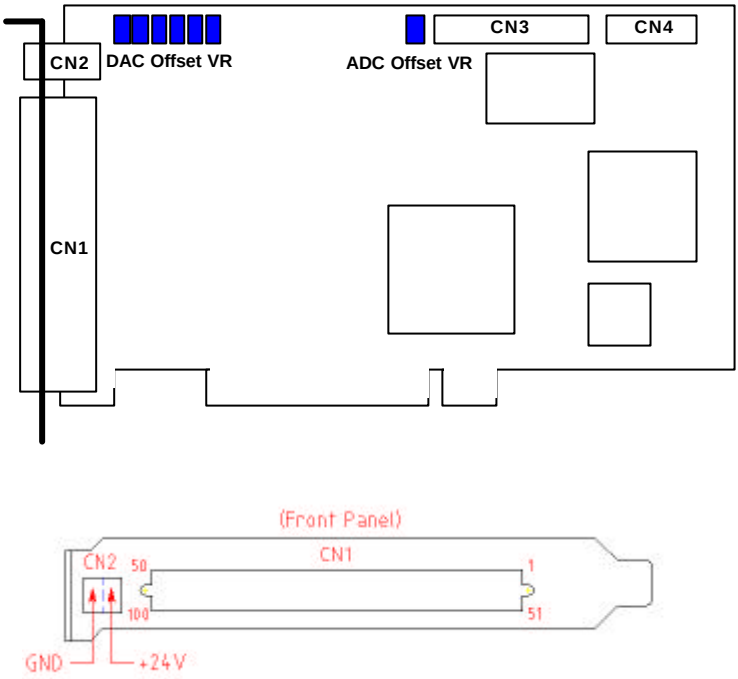


Figure 2.2.1 PCB Layout of the PCI-8136

CN1 Pin Assignments: Main Connector

CN1 Pin Assignments: External Power Input

CN3 Pin Assignments: DB25 Connector

CN4 Pin Assignments: DB9 Connector

DAC offset VR1~6

ADC offset VR

2.3 Hardware Installation

2.3.1 Hardware configuration

PCI-8136 has plug and play PCI controller on board. The memory usage (I/O port locations) of the PCI card is assigned by system BIOS. The address assignment is done on a board-by-board basis for all PCI cards in the system.

2.3.2 PCI slot selection

Your computer will probably have both PCI and ISA slots. Do not force the PCI card into a PC/AT slot. The PCI-8136 can be used in any PCI slot.

2.3.3 Installation Procedures

1. Read through this manual, and setup the jumper according to your application
2. Turn off your computer, Turn off all accessories (printer, modem, monitor, etc.) connected to computer.
3. Remove the cover from your computer.
4. Select a 32-bit PCI expansion slot. PCI slots are short than ISA or EISA slots and are usually white or ivory.
5. Before handling the PCI-8136, 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 PCI slot you selected.
7. Secure the card in place at the rear panel of the system unit using screw removed from the slot.

2.3.4 Trouble shooting:

If your system won't boot or if you experience erratic operation with your PCI board in place, it's likely caused by an interrupt conflict (perhaps because you incorrectly described the ISA setup). In general, the solution, once you determine it is not a simple oversight, is to consult the BIOS documentation that come with your system.

2.4 Software Installation

Please refer to the Software Installation Guide to install it.

2.5 CN1 Pin Assignments: Main Connector

The CN1 is the major connector for the Digital I/O, Pulse IO and Analog output signals.

No.	Name	I/O	Function	No	Name	I/O	Function
1	AGND	SG	Analog ground	51	AGND	SG	Analog ground
2	DAC1	O	Analog output, ①	52	DAC4	O	Analog output, ④
3	DAC2	O	Analog output, ②	53	DAC5	O	Analog output, ⑤
4	DAC3	O	Analog output, ③	54	DAC6	O	Analog output, ⑥
5	VCC+5V	O	+5V from PC	55	COM-	PG	+24V Ground
6	COM+	P	+24V Input for Digital I/O	56	COM-	PG	+24V Ground
7	COM+	P	+24V Input for Digital I/O	57	DI19	I	Digital Input 19
8	COM+	P	+24V Input for Digital I/O	58	DO7	O	Digital Output 07
9	DI01	I	Digital Input 01	59	DI02	I	Digital Input 02
10	DI07	I	Digital Input 07	60	DI09	I	Digital Input 09
11	DI08	I	Digital Input 08	61	DI10	I	Digital Input 10
12	DO1	O	Digital Output 01	62	DO2	O	Digital Output 02
13	DI03	I	Digital Input 03	63	DI04	I	Digital Input 04
14	DI11	I	Digital Input 11	64	DI13	I	Digital Input 13
15	DI12	I	Digital Input 12	65	DI14	I	Digital Input 14
16	DO3	O	Digital Output 03	66	DO4	O	Digital Output 04
17	DI05	I	Digital Input 05	67	DI06	I	Digital Input 06
18	DI15	I	Digital Input 15	68	DI17	I	Digital Input 17
19	DI16	I	Digital Input 16	69	DI18	I	Digital Input 18
20	DO5	O	Digital Output 05	70	DO6	O	Digital Output 06
21	EA1+	I	Encoder A-phase (+), ①	71	EA2+	I	Encoder A-phase (+), ②
22	EA1-	I	Encoder A-phase (-), ①	72	EA2-	I	Encoder A-phase (-), ②
23	EB1+	I	Encoder B-phase (+), ①	73	EB2+	I	Encoder B-phase (+), ②
24	EB1-	I	Encoder B-phase (-), ①	74	EB2-	I	Encoder B-phase (-), ②
25	EZ1+	I	Encoder Z-phase (+), ①	75	EZ2+	I	Encoder Z-phase (+), ②
26	EZ1-	I	Encoder Z-phase (-), ①	76	EZ2-	I	Encoder Z-phase (-), ②
27	EA3+	I	Encoder A-phase (+), ③	77	EA4+	I	Encoder A-phase (+), ④
28	EA3-	I	Encoder A-phase (-),③	78	EA4-	I	Encoder A-phase (-), ④
29	EB3+	I	Encoder B-phase (+),③	79	EB4+	I	Encoder B-phase (+), ④
30	EB3-	I	Encoder B-phase (-),③	80	EB4-	I	Encoder B-phase (-), ④
31	EZ3+	I	Encoder Z-phase (+),③	81	EZ4+	I	Encoder Z-phase (+), ④
32	EZ3-	I	Encoder Z-phase (-),③	82	EZ4-	I	Encoder Z-phase (-), ④
33	EA5+	I	Encoder A-phase (+),⑤	83	EA6+	I	Encoder A-phase (+),⑥
34	EA5-	I	Encoder A-phase (-),⑤	84	EA6-	I	Encoder A-phase (-),⑥
35	EB5+	I	Encoder B-phase (+),⑤	85	EB6+	I	Encoder B-phase (+),⑥
36	EB5-	I	Encoder B-phase (-),⑤	86	EB6-	I	Encoder B-phase (-),⑥
37	EZ5+	I	Encoder Z-phase (+),⑤	87	EZ6+	I	Encoder Z-phase (+),⑥
38	EZ5-	I	Encoder Z-phase (-),⑤	88	EZ6-	I	Encoder Z-phase (-),⑥
39	OUT1+	O	Pulse signal (+),①	89	OUT2+	O	Pulse signal (+), ②
40	OUT1-	O	Pulse signal (-),①	90	OUT2-	O	Pulse signal (-), ②
41	DIR1+	O	Dir. signal (+),①	91	DIR2+	O	Dir. Signal (+), ②
42	DIR1-	O	Dir. Signal (-),①	92	DIR2-	O	Dir. Signal (-), ②
43	OUT3+	O	Pulse signal (+), ③	93	OUT4+	O	Pulse signal (+),④
44	OUT3-	O	Pulse signal (-),③	94	OUT4-	O	Pulse signal (-),④
45	DIR3+	O	Dir. signal (+), ③	95	DIR4+	O	Dir. signal (+),④
46	DIR3-	O	Dir. signal (-), ③	96	DIR4-	O	Dir. signal (-),④
47	OUT5+	O	Pulse signal (+),⑤	97	OUT6+	O	Pulse signal (+),⑥
48	OUT5-	O	Pulse signal (-),⑤	98	OUT6-	O	Pulse signal (-),⑥
49	DIR5+	O	Dir. signal (+),⑤	99	DIR6+	O	Dir. signal (+),⑥
50	DIR5-	O	Dir. Signal (-),⑤	100	DIR6-	O	Dir. Signal (-),⑥

2.6 CN1 Pin Assignments: External Power Input

CN1 Pin No	Name	Description	Cable Color
1	EXGND	Grounds of the external power.	Black
2	EX+24V	External power supply of +24V DC ± 5%	Red

Notes:

1. CN2 is a plug-in terminal connector with no screw.

2. Be sure to use the external power supply. The +24V DC is used by external input/output signal circuit.

3. Wires for connection to CN2

Solid wire: φ 0.32mm to φ 0.65mm (AWG28 to AWG22)

Twisted wire:0.08mm^2 to 0.32mm^2 (AWG28 to AWG22)

Naked wire length:10mm standard

4. The EX+24V is shorted inside PCI-8136 with COM+ in CN1 (No. 6,7,8).

5. The EXGND is shorted inside PCI-8136 with COM- in CN1 (No. 55,56).

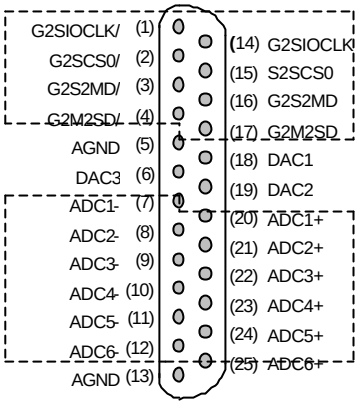
2.7 CN3 Pin Assignments: DB25 Connector

The signals on CN3 are for Analog input and remote serial IO.

Remote serial IO #2

Analog output

Analog input

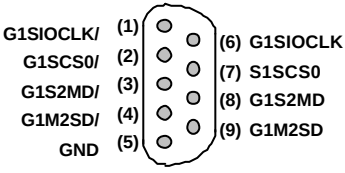


Note1: The DAC1~3 pins are the same with those on CN1

Note2: The Remote Serial IO #2 is reserved for future functions

2.8 CN4 Pin Assignments: DB9 Connector

The signals on CN4 are for remote serial IO#1.



Note1: The Remote Serial IO #1 is reserved for future functions

3

Signal Connections

The signal connections of all the I/O signals are described in this chapter. Please refer the contents of this chapter before wiring the cable between the PCI-8136 and IO Device.

This chapter contains the following sections:

- Section 3.1 Analog Inputs
- Section 3.2 Analog Outputs
- Section 3.3 Digital Inputs
- Section 3.4 Digital Outputs
- Section 3.5 Pulse Inputs
- Section 3.6 Pulse Outputs

3.1 Analog Input

The PCI-8136 provides 6 12-bit A/D converter channels. The analog source is selectable for each channel to be ±10V DC (Default) or ±20 mA by soldering a 124O resistor which is shipped with PCI-8136.

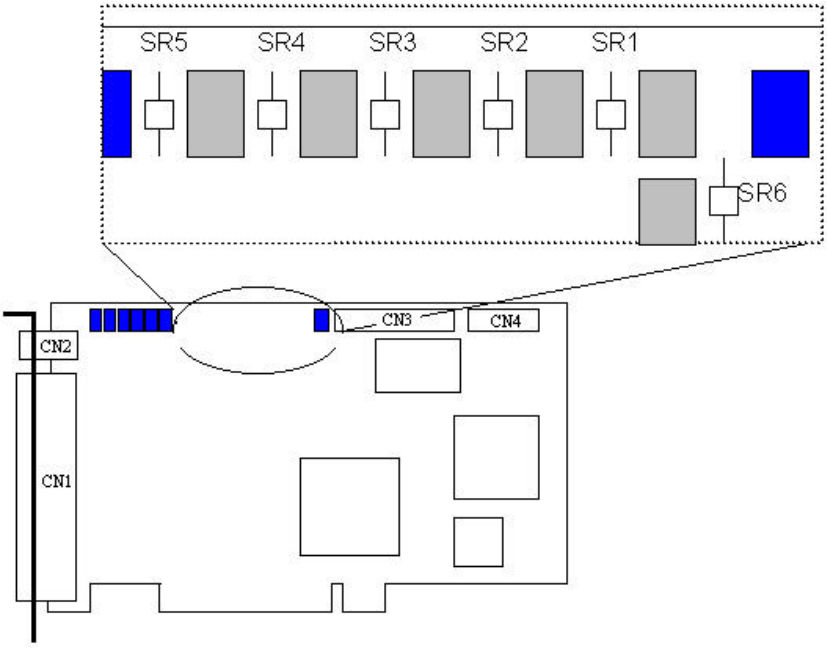


Figure 3.1.1: Current input mode – location of 124O resistor

To avoid ground loops and get more accuracy measurement of A/D conversion, it is quite important to understand the signal source type. The PCI-8136 provides differential input mode that consists of two inputs each channel.

Signal	PIN	Connector	Function
ADC1+	20	CN3 (DB25)	ADC channel 1 (+)
ADC1-	7	CN3 (DB25)	ADC channel 1 (-)
ADC2+	21	CN3 (DB25)	ADC channel 2 (+)
ADC2-	8	CN3 (DB25)	ADC channel 2 (-)
ADC3+	22	CN3 (DB25)	ADC channel 3 (+)
ADC3-	9	CN3 (DB25)	ADC channel 3 (-)
ADC4+	23	CN3 (DB25)	ADC channel 4 (+)
ADC4-	10	CN3 (DB25)	ADC channel 4 (-)
ADC5+	24	CN3 (DB25)	ADC channel 5 (+)
ADC5-	11	CN3 (DB25)	ADC channel 5 (-)
ADC6+	25	CN3 (DB25)	ADC channel 6 (+)
ADC6-	12	CN3 (DB25)	ADC channel 6 (-)

A differential source means the ends of the signal are not grounded. To avoid the danger of high voltage between the local ground of signal and the ground of the PC system, a shorted ground path must be connected. Figure 3.1.2 shows the connection of differential analog input source.

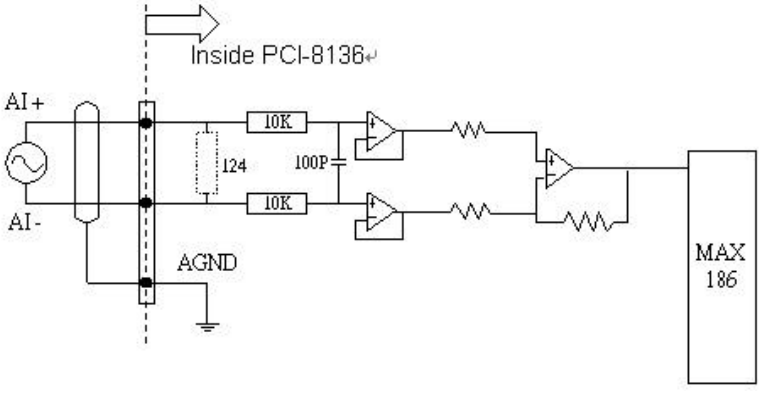


Figure 3.1.2 Analog input circuit

3.2 Analog Output

The PCI-8136 provides 6 16-bit Digital-to-Analog converter channels. The output voltage ranged from -10 V to +10V. To make correct connection, please refer to following figure:

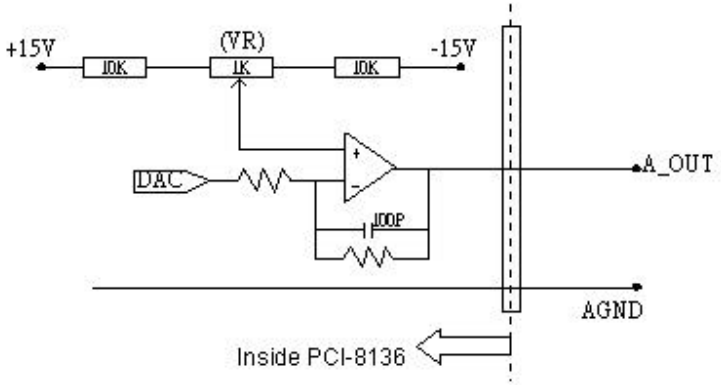


Figure 3.2.1 Analog output circuit

The Analog outputs are all single ended with common ground 'AGND'. The following is the pin assignment information for DAC.

Signal	PIN	Connector	Function
DAC1	2	CN1 (SCSI II – 100 Pin)	DAC Channel 1
DAC2	3	CN1 (SCSI II – 100 Pin)	DAC Channel 2
DAC3	4	CN1 (SCSI II – 100 Pin)	DAC Channel 3
DAC4	52	CN1 (SCSI II – 100 Pin)	DAC Channel 4
DAC5	53	CN1 (SCSI II – 100 Pin)	DAC Channel 5
DAC6	54	CN1 (SCSI II – 100 Pin)	DAC Channel 6
AGND	1	CN1 (SCSI II – 100 Pin)	Analog Ground
AGND	51	CN1 (SCSI II – 100 Pin)	Analog Ground
AGND	5	CN3 (DB25)	Analog Ground
AGND	13	CN3 (DB25)	Analog Ground
DAC1	18	CN3 (DB25)	DAC Channel 1
DAC2	19	CN3 (DB25)	DAC Channel 2
DAC3	6	CN3 (DB25)	DAC Channel 3

3.3 Digital Input

The PCI-8136 provides 19 digital inputs with 2500V rms isolation. The system recognizes a logical '1' when no current goes from COM+ to DIf, and, Logical '0' is returned when current goes from COM+ to DIf . The max current passing trough DIf must be less than 20mA.

Here is the input circuit of digital input channels.

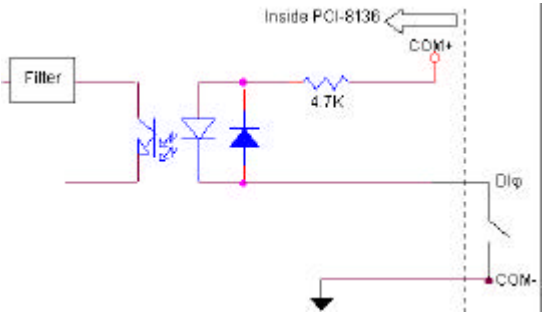
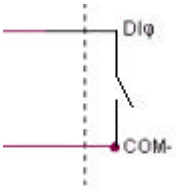


Figure 3.3.1 Digital input circuit

Example of input wiring:



When switch close → Logical “Low”

When switch open → Logical “High”

The following is the pin assignment information for digital input.

Signal	PIN	Connector	Function
DI01	9	CN1 (SCSI II – 100 Pin)	Digital Input Ch. 01
DI02	59	CN1 (SCSI II – 100 Pin)	Digital Input Ch. 02
DI03	13	CN1 (SCSI II – 100 Pin)	Digital Input Ch. 03
DI04	63	CN1 (SCSI II – 100 Pin)	Digital Input Ch. 04
DI05	17	CN1 (SCSI II – 100 Pin)	Digital Input Ch. 05
DI06	67	CN1 (SCSI II – 100 Pin)	Digital Input Ch. 06
DI07	10	CN1 (SCSI II – 100 Pin)	Digital Input Ch. 07
DI08	11	CN1 (SCSI II – 100 Pin)	Digital Input Ch. 08
DI09	60	CN1 (SCSI II – 100 Pin)	Digital Input Ch. 09
DI10	61	CN1 (SCSI II – 100 Pin)	Digital Input Ch. 10
DI11	14	CN1 (SCSI II – 100 Pin)	Digital Input Ch. 11
DI12	15	CN1 (SCSI II – 100 Pin)	Digital Input Ch. 12
DI13	64	CN1 (SCSI II – 100 Pin)	Digital Input Ch. 13
DI14	65	CN1 (SCSI II – 100 Pin)	Digital Input Ch. 14
DI15	18	CN1 (SCSI II – 100 Pin)	Digital Input Ch. 15
DI16	19	CN1 (SCSI II – 100 Pin)	Digital Input Ch. 16
DI17	68	CN1 (SCSI II – 100 Pin)	Digital Input Ch. 17
DI18	69	CN1 (SCSI II – 100 Pin)	Digital Input Ch. 18
DI19	57	CN1 (SCSI II – 100 Pin)	Digital Input Ch. 19
COM-	55	CN1 (SCSI II – 100 Pin)	DIO Common Ground
COM-	56	CN1 (SCSI II – 100 Pin)	DIO Common Ground

3.4 Digital Output

The PCI-8136 provides 7 open collector outputs with 2500 V rms isolation. The maximum output switching frequency is 10 kHz, when the continuous output supply current is subject to 500mA/total, 90mA/CH(typical), and 150mA/CH(max). In power-on state, the system issues a logical '0'.

Here is the output circuit of digital output channels.

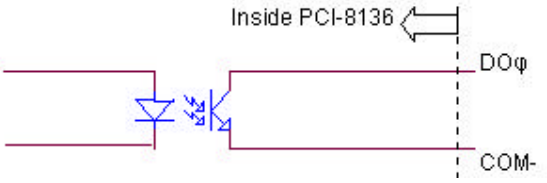
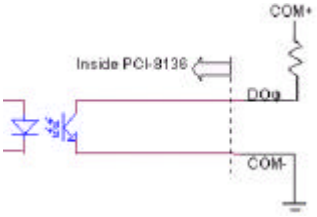


Figure 3.4.1 Digital output circuit

Example of output wiring:



When set output as “HIGH”, current goes from COM+ to COM-

When set output as “LOW”, the resistance value between DOf and COM- becomes extremely large, so that no current was drawn from COM+.

The following is the pin assignment information for Digital Output.

Signal	PIN	Connector	Function
COM+	6	CN1 (SCSI II – 100 Pin)	EXT +24V for Digital Output
COM+	7	CN1 (SCSI II – 100 Pin)	EXT +24V for Digital Output
COM+	8	CN1 (SCSI II – 100 Pin)	EXT +24V for Digital Output
DO1	12	CN1 (SCSI II – 100 Pin)	Digital Output CH.1
DO2	62	CN1 (SCSI II – 100 Pin)	Digital Output CH.2
DO3	16	CN1 (SCSI II – 100 Pin)	Digital Output CH.3
DO4	66	CN1 (SCSI II – 100 Pin)	Digital Output CH.4
DO5	20	CN1 (SCSI II – 100 Pin)	Digital Output CH.5
DO6	70	CN1 (SCSI II – 100 Pin)	Digital Output CH.6
DO7	58	CN1 (SCSI II – 100 Pin)	Digital Output CH.7
COM-	55	CN1 (SCSI II – 100 Pin)	DIO Common Ground
COM-	56	CN1 (SCSI II – 100 Pin)	DIO Common Ground

3.5 Pulse Input (Encoder Counter)

The PCI-8136 provides 6 differential pulse inputs with 2500V rms isolation. The pulse mode is software programmable to be AB-phase, CW/CCW, or Pulse/Direction, and the counter speed goes up to 2 MHz. The relative signal names, and pin numbers are shown in the following tables.

Signal	PIN	Connector	Function
EA1+	21	CN1 (SCSI II – 100 Pin)	Encoder CH.1 A-Phase (+)
EA1-	22	CN1 (SCSI II – 100 Pin)	Encoder CH.1 A-Phase (-)
EB1+	23	CN1 (SCSI II – 100 Pin)	Encoder CH.1 B-Phase (+)
EB1-	24	CN1 (SCSI II – 100 Pin)	Encoder CH.1 B-Phase (-)
EZ1+	25	CN1 (SCSI II – 100 Pin)	Encoder CH.1 Z-Phase (+)
EZ1-	26	CN1 (SCSI II – 100 Pin)	Encoder CH.1 Z-Phase (-)
EA2+	71	CN1 (SCSI II – 100 Pin)	Encoder CH.2 A-Phase (+)
EA2-	72	CN1 (SCSI II – 100 Pin)	Encoder CH.2 A-Phase (-)
EB2+	73	CN1 (SCSI II – 100 Pin)	Encoder CH.2 B-Phase (+)
EB2-	74	CN1 (SCSI II – 100 Pin)	Encoder CH.2 B-Phase (-)
EZ2+	75	CN1 (SCSI II – 100 Pin)	Encoder CH.2 Z-Phase (+)
EZ2-	76	CN1 (SCSI II – 100 Pin)	Encoder CH.2 Z-Phase (-)
EA3+	27	CN1 (SCSI II – 100 Pin)	Encoder CH.3 A-Phase (+)
EA3-	28	CN1 (SCSI II – 100 Pin)	Encoder CH.3 A-Phase (-)
EB3+	29	CN1 (SCSI II – 100 Pin)	Encoder CH.3 B-Phase (+)
EB3-	30	CN1 (SCSI II – 100 Pin)	Encoder CH.3 B-Phase (-)
EZ3+	31	CN1 (SCSI II – 100 Pin)	Encoder CH.3 Z-Phase (+)
EZ3-	32	CN1 (SCSI II – 100 Pin)	Encoder CH.3 Z-Phase (-)
EA4+	77	CN1 (SCSI II – 100 Pin)	Encoder CH.4 A-Phase (+)
EA4-	78	CN1 (SCSI II – 100 Pin)	Encoder CH.4 A-Phase (-)
EB4+	79	CN1 (SCSI II – 100 Pin)	Encoder CH.4 B-Phase (+)
EB4-	80	CN1 (SCSI II – 100 Pin)	Encoder CH.4 B-Phase (-)
EZ4+	81	CN1 (SCSI II – 100 Pin)	Encoder CH.4 Z-Phase (+)
EZ4-	82	CN1 (SCSI II – 100 Pin)	Encoder CH.4 Z-Phase (-)
EA5+	33	CN1 (SCSI II – 100 Pin)	Encoder CH.5 A-Phase (+)
EA5-	34	CN1 (SCSI II – 100 Pin)	Encoder CH.5 A-Phase (-)
EB5+	35	CN1 (SCSI II – 100 Pin)	Encoder CH.5 B-Phase (+)
EB5-	36	CN1 (SCSI II – 100 Pin)	Encoder CH.5 B-Phase (-)
EZ5+	37	CN1 (SCSI II – 100 Pin)	Encoder CH.5 Z-Phase (+)
EZ5-	38	CN1 (SCSI II – 100 Pin)	Encoder CH.5 Z-Phase (-)
EA6+	83	CN1 (SCSI II – 100 Pin)	Encoder CH.6 A-Phase (+)
EA6-	84	CN1 (SCSI II – 100 Pin)	Encoder CH.6 A-Phase (-)
EB6+	85	CN1 (SCSI II – 100 Pin)	Encoder CH.6 B-Phase (+)
EB6-	86	CN1 (SCSI II – 100 Pin)	Encoder CH.6 B-Phase (-)
EZ6+	87	CN1 (SCSI II – 100 Pin)	Encoder CH.6 Z-Phase (+)
EZ6-	88	CN1 (SCSI II – 100 Pin)	Encoder CH.6 Z-Phase (-)

The input circuits of the EA, EB, EZ signals are shown as follows.

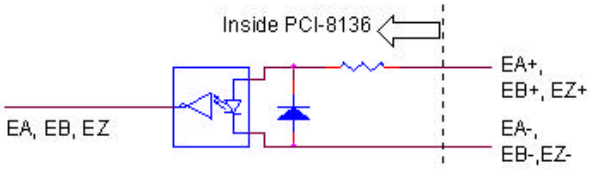


Figure 3.5.1 Pulse input (encoder counter) circuit

Note: The voltage across every differential pair of encoder input signals (EA+, EA-), (EB+, EB-) and (EZ+, EZ-) should be at least 3.5V or higher. Therefore, you have to take care of the driving capability when connecting with the encoder feedback or motor driver feedback.

Here are two examples of connecting the input signals with the external circuits. The input circuits can connect to the encoder or motor driver, which are equipped with: (1) differential line driver or (2) open collector output.

◆ ***Connection to Line Driver Output***

To drive the PCI-8136 encoder input, the driver output must provide at least 3.5V across the differential pairs with at least 6 mA driving capability. The ground level of the two sides must be tight together too.

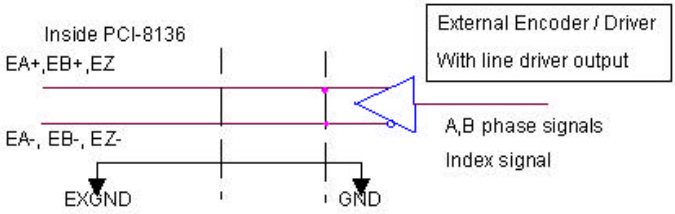


Figure 3.5.2 Connection to line driver output

◆ ***Connection to Open Collector Output***

To connect with open collector output, an external power supply is necessary. Some motor drivers also provide the power source. The connection between PCI-8136, encoder, and the power supply is shown in the following diagram. Please note that the external current limit resistor R is necessary to protect the PCI-8136 input circuit. The following table lists the suggested resistor value according to the encoder power supply.

Encoder Power(VDD)	External Resistor R
+5V	0 Ω (None)
+12V	1.8kΩ
+24V	4.3kΩ

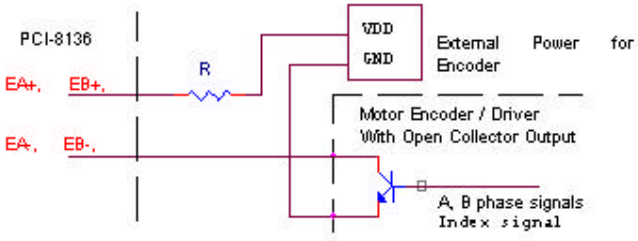


Figure 3.5.3 Connect to open collector output

3.6 Pulse Output (Pulse Generator)

The PCI-8136 provides 6 differential pulse output channels. The pulse mode is software programmable to be Pulse/Direction, CW/CCW, or AB-phase, and the output frequency goes up to 500 KHz. The operations of pulse output channels are quite straightforward. Call library functions to set pulse mode, send pulse at constant frequency, and stop pulse. Please refer to chapter 5 for detail function descriptions.

The relative signal names, and pin numbers are shown in the following tables.

Signal	PIN	Connector	Function
OUT1+	39	CN1 (SCSI II – 100 Pin)	CH.1 Pulse Signal (+)
OUT1-	40	CN1 (SCSI II – 100 Pin)	CH.1 Pulse Signal (-)
DIR1+	41	CN1 (SCSI II – 100 Pin)	CH.1 Direction Signal (+)
DIR1-	42	CN1 (SCSI II – 100 Pin)	CH.1 Direction Signal (-)
OUT2+	89	CN1 (SCSI II – 100 Pin)	CH.2 Pulse Signal (+)
OUT2-	90	CN1 (SCSI II – 100 Pin)	CH.2 Pulse Signal (-)
DIR2+	91	CN1 (SCSI II – 100 Pin)	CH.2 Direction Signal (+)
DIR2-	92	CN1 (SCSI II – 100 Pin)	CH.2 Direction Signal (-)
OUT3+	43	CN1 (SCSI II – 100 Pin)	CH.3 Pulse Signal (+)
OUT3-	44	CN1 (SCSI II – 100 Pin)	CH.3 Pulse Signal (-)
DIR3+	45	CN1 (SCSI II – 100 Pin)	CH.3 Direction Signal (+)
DIR3-	46	CN1 (SCSI II – 100 Pin)	CH.3 Direction Signal (-)
OUT4+	83	CN1 (SCSI II – 100 Pin)	CH.4 Pulse Signal (+)
OUT4-	84	CN1 (SCSI II – 100 Pin)	CH.4 Pulse Signal (-)
DIR4+	85	CN1 (SCSI II – 100 Pin)	CH.4 Direction Signal (+)
DIR4-	86	CN1 (SCSI II – 100 Pin)	CH.4 Direction Signal (-)
OUT5+	47	CN1 (SCSI II – 100 Pin)	CH.5 Pulse Signal (+)
OUT5-	48	CN1 (SCSI II – 100 Pin)	CH.5 Pulse Signal (-)
DIR5+	49	CN1 (SCSI II – 100 Pin)	CH.5 Direction Signal (+)
DIR5-	50	CN1 (SCSI II – 100 Pin)	CH.5 Direction Signal (-)
OUT6+	97	CN1 (SCSI II – 100 Pin)	CH.6 Pulse Signal (+)
OUT6-	98	CN1 (SCSI II – 100 Pin)	CH.6 Pulse Signal (-)
DIR6+	99	CN1 (SCSI II – 100 Pin)	CH.6 Direction Signal (+)
DIR6-	100	CN1 (SCSI II – 100 Pin)	CH.6 Direction Signal (-)

Here is the circuit of PCI-8136 pulse output (pulse generator) channels.

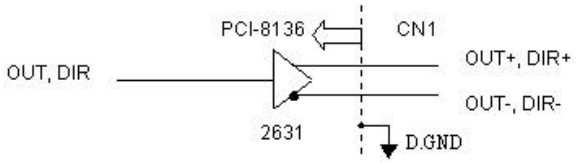


Figure 3.6.1 Pulse output (pulse generator) circuit

3.7 VCC Pin

There is one pin named VCC+5V on Pin5 of CN1. This voltage source is from computer directly. Please don't use this voltage source on any device which is connected to PCI-8136's isolation I/Os. If not, the grounds will be connected at both of the isolative sides and the noise will be introduced from this loop. So the isolation will be meaningless. The VCC's ground is the same with AGND in side the board.

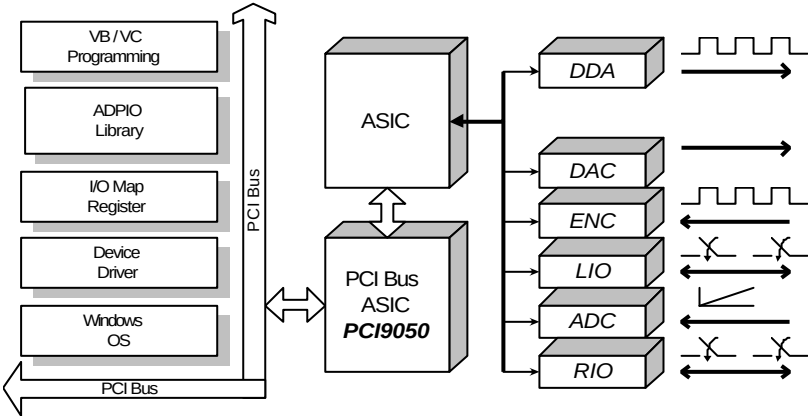
4

Operation Theorem

This chapter describes the detail operation of the PCI-8136 card. Contents of the following sections are as following.

- Section 4.1: AD Conversion and Preloaded Trigger
- Section 4.2: DA Conversion
- Section 4.3: Local DIO
- Section 4.4: Pulse Input and Position Compare
- Section 4.5: Pulse Output
- Section 4.6: Remote serial IO

Please refer to the following architecture diagram of PCI-8136

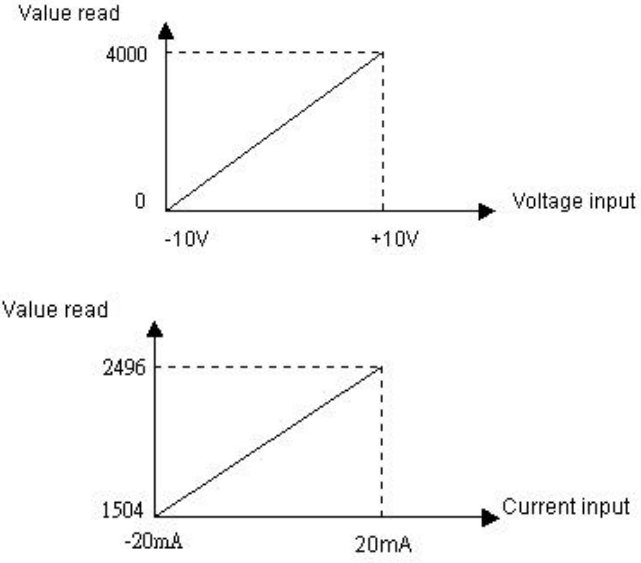


ADC: Please refer to Section 4.1: AD Conversion and Preloaded Trigger.
DAC: Please refer to Section 4.2: DA Conversion.
LIO: Please refer to Section 4.3: DIO.
ENC: Please refer to Section 4.4: Pulse Input and Position Compare.
RIO: Please refer to Section 4.6: Remote serial IO.

4.1 AD Conversion and Preloaded Trigger

4.1.1 ADC

The PCI-8136 provides 6 differential ADC channels. Each channel consists of two inputs. One is for (+) signal and the other is for (-) signal. The input signal may be voltage ranged from–10 ~ +10V or current ranged from–20mA ~ 20mA. The ADC resolution is 12-bit. The following figure shows the A (voltage or current) to D (value read) converting table. The zero voltage or current is at value 2000.



Related functions:

_8136_A_Initial(): please refer to section 5.2
_8136_A_Read_Value(), _8136_A_Read_Volt(): please refer to section 5.7

4.1.2 Voltage Compare

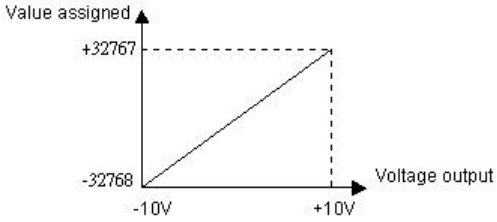
The voltage compare function of PCI-8136 is very useful. It allows user to set a compare value by software function. When one ADC signal reaches the pre-set value, an interrupt will be generated for corresponding channel.

Relative functions:

_8136_A_Set_Compare_Value(), _8136_A_Set_Compare_Volt (), _8136_S_Set_Int_Factor() : please refer to section 5,9

4.2 DA Conversion

The PCI-8136 provides 6 channel 16-bit, bipolar (±10V DC) digital to analog converter. The D (value assigned) to A (voltage output) converting chart is showed bellow.



4.2.1 DA Output by Trigger Source

PCI-8136 allows users to set a pre-load value for each DAC channel. The value will be sent once any trigger condition for this channel is happened. The trigger source could be from encoder counter comparators or ADCs by setting the trigger map in the software functions. Users can set every channel's trigger sources independently. The compare method could be set in Set_Int_Factor() function.

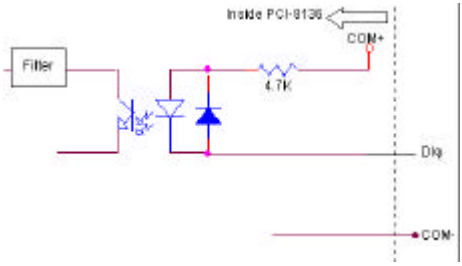
Related functions:

_8136_A_Initial() : please refer to section 5.2
_8136_A_Write_Value(), _8136_A_Write_Volt(),
_8136_A_Output_Control() : please refer to section 5.7
_8136_A_Set_Preload_Volt(), _8136_A_Set_Trigger(),
_8136_A_Set_Trigger_Map() : please refer to section 5.7
_8136_S_Set_Int_Factor() : please refer to section 5,9

4.3 Local DIO

4.3.1 Digital Input

The PCI-8136 provides 19 digital input channels with 2500Vrms isolation. The DI channel is logically “HIGH” when no current goes from COM+ to DIf , and, Logically “LOW” when current goes from COM+ to DIf . The max current passing through DIf must be less than 20mA..

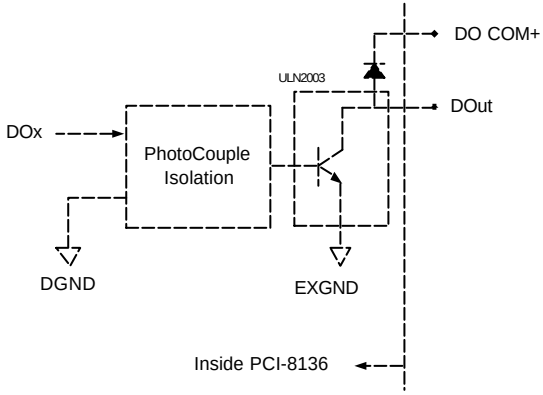


Related functions:

_8136_D_Input(), _8136_D_InputA() : please refer to section 5.5

4.3.2 Digital Output

The PCI-8136 provides 7 open collector output channels with 2500rms isolation. Please carefully refer to section 3.4 for the circuit wiring.



Related functions:

_8136_D_Output(), _8136_D_OutputA() : please refer to section 5.5

4.4 Pulse Input and Position Compare

4.4.1 Pulse Input

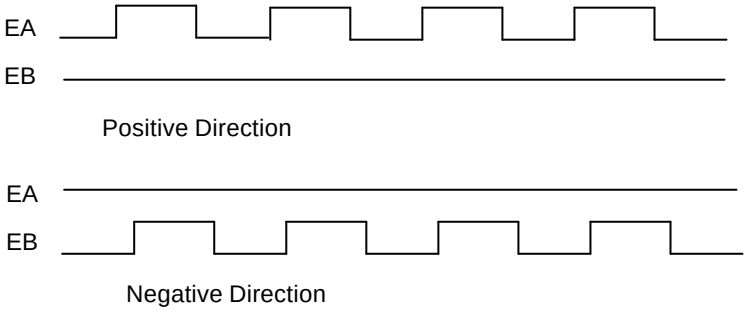
The PCI-8136 has 6 32-bit pulse input channels for encoder counter. It can accept 3 kinds of pulse signal:

1. Plus and minus pulses input (CW/CCW mode)
2. 90°phase difference signals(AB phase mode)
3. Pulse and direction input(Pulse/DIR).

90°phase difference signals may be selected to be multiplied by a factor of 1,2 or 4x AB phase mode is the most commonly used for incremental encoder input. For example, if a rotary encoder has 2000 pulses per phase (A or B phase), then the value read from the counter will be 8000 pulses per turn These input modes can be selected by software function call.

◆ ***Plus and minus pulses input mode(CW/CCW Mode)***

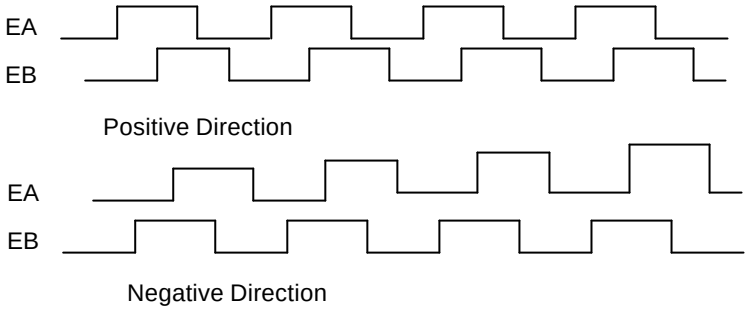
In this mode, pulse from EA causes the counter to count up, whereas EB caused the counter to count down.



◆ ***90°phase difference signals Input Mode(AB phase Mode)***

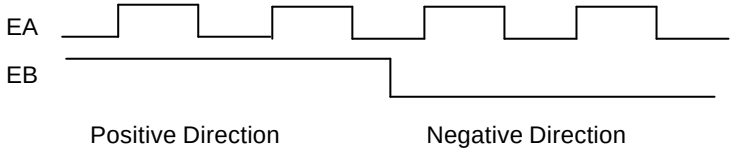
In this mode, the EA signal is 90° phase leading or lagging in comparison with EB signal. Where “lead” or “lag' of phase difference between two signals is caused by the turning direction of motors. The up/down counter counts up when the phase of EA signal leads the phase of EB signal.

The following diagram shows the waveform.



◆ ***Pulse and direction input (Pulse/DIR)***

In this mode, the high / low status of EB decides the counter value to increase or decrease (Direction), whereas EA decide the count number (Pulse).



4.4.2 Encoder Counter Value Capture (latch)

The EZ (index signal) of each pulse input channel doesn't affect counter value. It can capture (latch) current counter value by proper setting, and generates an interrupt signal when receiving a rising edge. The counter value capture function is very useful to sensing position of a moving object.

Related functions:

_8136_P_Initial() : please refer to section 5.2
_8136_P_Set_Input_Type(), _8136_P_Read(), _8136_P_Clear(),
_8136_P_Set_Index_Latch(), _8136_P_Read_Index(),
_8136_P_Read_Latch_Value() : please refer to section 5.8

4.4.3 Encoder Counter Value Compare

The PCI-8136 provides position compare function for all six pulse input channels. Once the counter value is reached the pre-set compare value, an interrupt signal will be generated immediately. This function can effectively reduce the overhead of CPU's polling for current position.

Related functions:

_8136_S_Set_Int_Factor() : please refer to section 5,9
_8136_P_Set_Compare_Value(): please refer to section 5.8

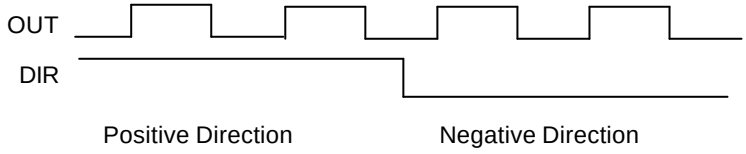
4.5 Pulse Output

The PCI-8136 provides 6 pulse output channels. They are used to send out constant-frequency pulse trains. When changing the output frequency of any channel, there is at most 265ms time delay.

There are also 3 kinds of pulse output: (1). plus and minus pulses input(CW/CCW mode); (2) pulse and direction input(Pulse/DIR); (3). 90° phase difference signals(AB phase mode);

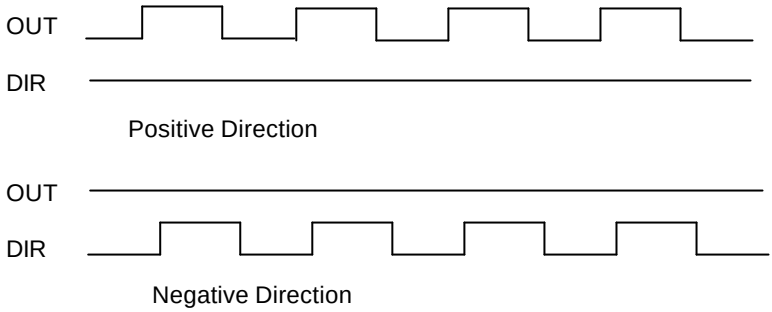
◆ ***Pulse and direction input(Pulse/DIR)***

In this mode, the high / low status of DIR defines the plus / negative direction, whereas OUT generates the pulse train.



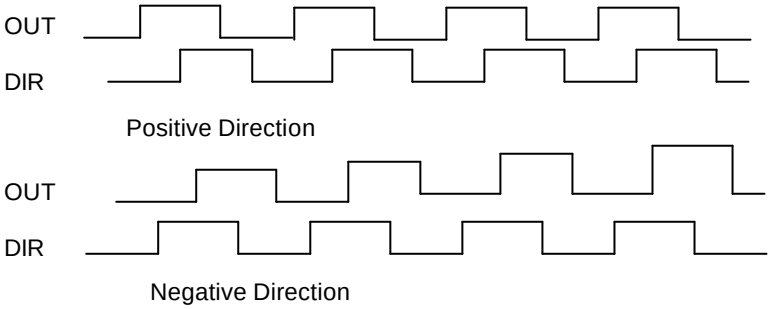
◆ ***Plus and minus pulses input mode(CW/CCW Mode)***

In this mode, plus frequency (plus direction) goes on OUT, whereas negative frequency (negative direction) is generated from DIR.



◆ ***90°phase difference signals input mode(AB phase Mode)***

In this mode, the OUT signal is 90° phase leading or lagging in comparison with DIR signal. Where “lead” or “lag' of phase difference between two signals is caused by the direction of pulse train.



Related functions:

_8136_P_Initial() : please refer to section 5.2
_8136_P_Set_Output_Type(), _8136_P_Send(), _8136_P_Stop(),
_8136_P_Change_Speed(), please refer to section 5.2

4.6 Remote Serial IO

The PCI-8136 support 2 sets of remote serial IO, each set may consist at most 64DI and 64DO. To use remote serial IO function, a slave module is needed.

Note: The remote I/O functions are reserved on current version

Related functions:

_8136_R_Initial() : please refer to section 5.2
_8136_R_Status(), _8136_R_Write(), _8136_R_Read() : please refer to section 5.6

4.7 Interrupt Control

The PCI-8136 can generate INT signal to host PC according to 7 types of factors, refer to **_8136_S_Set_Int_factor()** function for more details. Type 0 to type 2 are local digital Input interrupts. The following table represents the interrupt setting for local digital Input:

_8136_S_Set_Int_factor(Cardno, ChannelNo, Int_Factor)

Channel / Int. factor bit(type)	0	1	2	3	4	5
0	DI 6/7	DI 8/9	DI 10/11	DI 12/13	DI 14/15	DI 16/17
1	DI 18	-	-	-	-	-
2	DI 00	DI 01	DI 02	DI 03	DI 04	DI 05
3	EZ1	EZ2	EZ3	EZ4	EZ5	EZ6
4	CMP1	CMP2	CMP3	CMP4	CMP5	CMP6
5	Timer	-	-	-	-	-
6	ADC1	ADC2	ADC3	ADC4	ADC5	ADC6

Note: Once the first row of interrupt factor is set, both of the digit channels' interrupt are enabled. The interrupt comes when any one of the two digital channels' is triggered.

Note: EZ is index signal , CMP is position compare true, Timer is card's internal timer, ADC is analog compare true.

For each channel number, the interrupt can be set according to above table.

Because user can't deal with interrupt under Windows System. PCI-8136 has another way for users to receive interrupts. That is Windows message system. This card uses events to notice user's program if interrupt is coming. Users can create a thread to get the interrupt events and then use **_8136_S_Get_Int_Status()** to get the interrupt status. The status bits for each axis is in the following table:

_8136_S_Get_Int_Status(Cardno, AxisNo, Status)

Bits	Description
0	Digital Input pin 6/8/10/12/14/16 interrupt
1	Digital Input pin 7/9/11/13/15/17 interrupt
2	Digital Input pin 18 interrupt
3	Digital Input pin 0~5 interrupt
4	Index signal interrupt
5	Position counter compare interrupt
6	Internal timer interrupt
7	Analog compared interrupt

◆ ***Use Thread to deal with Interrupt under Windows NT/95***

In order to detect the interrupt signal from PCI-8136 under Windows NT/95, users must create a thread routine first. Then use APIs provided by PCI-8136 to get the interrupt signal.Each card has 7 events for these interrupts. Event 0 ~ 5 stands for channel 0~5 and event 6 stands for timer interrupt and alarm interrupt. The sample program is as follows:

Situatuins: Assume that we have one card and want to receive Timer interrupt.

Steps:

1. Define a Global Value to deal with interrupt event

```
HANDLE hEvent[7];
volatile bool ThreadOn;
```

2. In Initializing Section (you must Initialize PCI-8136 properly first), set interrupt types and enable an event for each axis.

```
_8136_S_Set_Int_Factor(0,0, 0x40);
_8136_S_INT_Control(0,1);
_8136_INT_Enable(0,&hEvent[0]);
```

Note: For each card, you must assign a 7-events-array in _8136_INT_Enable function.

---

3. Define a Global Function (Thread Body). Use WaitForSingleObject() or WaitForMultipleObjects() to wait events. Remenber to reset this event after you get the event.

```
UINT IntThreadProc(LPVOID pParam)
{
U32 IntSts;
while(ThreadOn==TRUE)
{
::WaitForSingleObject(hEvent[6],INFINITE);
_8136_S_Get_Int_status(0,0,&IntSts);
::ResetEvent(hEvent[6]);
}
return 0;
}
```

4. Start the thread(Use a boolean value to control the thread's life)

```
ThreadOn=TRUE;
AfxBeginThread(IntThreadProc,GetSafeHwnd(),THREAD_PRIORITY_
   NORMAL);
```

5. Before exit the program, remember to let the thread go to end naturally.

```
ThreadOn=FALSE;
```

Note: If users need to deal interrupt under VB6.0. Please refer to section 5.9 for details.

We suggest user to create a thread and use WaitForSingleObject() for each events in order to guarantee the performance.

5

Function Library

This chapter describes the supporting software for PCI-8136 cards. User can use these functions to develop application program in C or Visual Basic or C++ language.

The function prototypes and some common data types are decelerated in **PCI-8136.H**. These data types are used by PCI-8136 library. We suggest you to use these data types in your application programs. The following table shows the data type names and their range.

Type Name	Description	Range
U8	8-bit ASCII character	0 to 255
I16	16-bit signed integer	-32768 to 32767
U16	16-bit unsigned integer	0 to 65535
I32	32-bit signed long integer	-2147483648 to 2147483647
U32	32-bit unsigned long integer	0 to 4294967295
F32	32-bit single-precision floating-point	-3.402823E38 to 3.402823E38
F64	64-bit double-precision floating-point	-1.797683134862315E308 to 1.797683134862315E309
Boolean	Boolean logic value	TRUE, FALSE

The functions of PCI-8136's software drivers use full-names to represent the functions' real meaning. The naming convention rules are:

In C Environment :

_{hardware_model}_{action_name}. e.g. **_8136_Initial()**.

In order to recognize the difference between C/C++ library and Visual Basic library, A capital "B" is put on the head of each function name of the Visual Basic function. e.g. **B_8136_Initial()**

5.1 List of Functions

Initialization	***Section 5.2***
_8136_Initial(*cardno)	Interface card initialization
_8136_S_Close(cardno)	Interface card close
_8136_A_Initial(cardno)	Analog I/O initialization
_8136_P_Initial(cardno)	Pulse I/O initialization
_8136_R_Initial	Remote I/U Initialze

System Parameters	***Section 5.3***
_8136_R_Set_RIO_Clk(cardno, slaveno, clk)	Set Rio clock divider
_8136_A_Set_DAC_Clk(cardno,clk)	Set DAC clock divider
_8136_A_Set_ADC_Clk(cardno,clk)	Set ADC clock divider
_8136_S_Set_Timer_Value(cardno,timer)	Set Timer click count
_8136_P_Set_Enc_Filter(cardno,filter)	Set encoder filter clock

Card Information	***Section 5.4***
_8136_S_Get_IRQ_Channel(cardno, *irq)	Get I/O card's IRQ
_8136_S_Get_Base_Addr(cardno, *base)	Get I/O card's base address

Digital I/O	***Section 5.5***
_8136_D_Output(cardno, ch, value)	Digital output for one bit
_8136_D_Input(cardno, ch, *value)	Digital input for one bit
_8136_D_OutputA(cardno, value)	Digital Output for all bits once
_8136_D_InputA(cardno, *value)	Digital Input for all bits once

Remote I/O	***Section 5.6***
_8136_R_Status(cardno, slaveno)	Check Remote I/O Status
_8136_R_Write(cardno, slave, set, value)	Write a word to remote
_8136_R_Read(cardno, slave, set, *value)	Read a word from remote

Analog I/O	***Section 5.7***
_8136_A_Write_Value(cardno, ch, value)	Output DAC in value
_8136_A_Write_Volt(cardno, ch, value)	Output DAC in voltage
_8136_A_Read_Value(cardno, ch, *value)	Input from ADC in value
_8136_A_Read_Volt(cardno, ch, *volt)	Input from ADC in voltage
_8136_A_Output_Control(cardno, ch, ctrl)	Start or stop DAC output
_8136_A_Set_Trigger(cardno, ch, ctrl)	Set DAC output by trigger
_8136_A_Set_Trigger_Map(cardno, ch, src)	Select DAC trigger source
_8136_A_Set_Preload_Volt(cardno,ch ,volt)	Set DAC trigger output voltage
_8136_A_Set_Compare_Value(card,ch,value)	Set ADC compare value
_8136_A_Set_Compare_Volt(card,ch,value)	Set ADC compare voltage

Pulse I/O	***Section 5.8***
_8136_P_Set_Output_Type(card, enc, fmt)	Set pulse output mode
_8136_P_Set_Input_Type(card, enc, fmt, mul)	Set pulse input mode
_8136_P_Read(card, enc, *data)	Read encoder counter
_8136_P_Clear(card, enc)	Clear encoder counter
_8136_P_Send(card, enc, frequency)	Send a constant pulse train
_8136_P_Stop(card, enc);	Stop pulse train
_8136_P_Change_Speed(card, enc, freq)	Change pulse train frequency
_8136_P_Read_Index(card, enc, *index)	Read index value
_8136_P_Set_Index_Latch(card, enc, type)	Set index latch type
_8136_P_Read_Latch_Value(card, enc, *data)	Read a latched encoder data
_8136_P_Set_Compare_Value(card, enc, dta)	Set a encoder compare data

_8136_INT_Enable(cardno, *event)	Set interrupt event handler
_8136_INT_Disable(cardno)	Remove int. event handler
_8136_S_Set_Int_Factor(cardno, ax,factor,op)	Set interrupt factor
_8136_S_INT_Control(cardno, ctrl)	Enable/disable interrupt
_8136_S_Get_Int_Status(cardno, ch, *status)	Get Int. status
_8136_Callback_Function(cardno, *callbkfn)	Set a call back function for int.

5.2 Initialization

@ Name

```
_8136_Initial – Software Initialize for PCI-8136
_8136_S_Close – Software release for PCI-8136
_8136_A_Initial – Initialize ADC/DAC functions for PCI-8136
_8136_P_Initial – Initialize pulse output engine and encoder
   counter
_8136_R_Initial – Initialize Remote I/O Module
```

@ Description

_8136_Initial:

This function is used to initialize PCI-8136 card. User must use this function before any operation in the program. This function will return a number to notice user how many cards is found.

_8136_S_Close:

This function is used to close PCI-8136 card. It releases the resources which are declared by driver. User must use this function before the program ends.

_8136_A_Initial:

This function enables serial ADC and DAC functions and set the transmission clock divider. User must use this function before he wants to use Analog IO.

_8136_P_Initial:

This function is for setting the encoder counter's clockand enable pulse output functions. User must use this function to enable pulse input and output.

_8136_R_Initial:

This function is for initialization remote I/O module. User can initial each module from this function.

@ Syntax

C/C++ (DOS, Windows 95/98/NT/2000)

```
I16 _8136_Initial(I16 *existCards);
I16 _8136_S_Close(I16 CardNo);
I16 _8136_A_Initial(I16 CardNo);
I16 _8136_P_Initial(I16 CardNo);
U16 _8136_R_Initial(I16 CardNo, I16 SlaveControl);
```

Visual Basic 5.0 or higher

```
B_8136_Initial (existCards As Integer) As Integer
B_8136_S_Close (ByVal CardNo As Integer) As Integer
B_8136_A_Initial (ByVal CardNo As Integer) As Integer
B_8136_P_Initial (ByVal CardNo As Integer) As Integer
B_8136_R_Initial (ByVal CardNo As Integer, ByVal SlaveControl
   As Integer) As Integer
```

@ Arguments

```
CardNo: card number designated to set (Range 0 ~ 3)
AxisNo: axis number designated to set (Range 0 ~ 5)
*existCards: a return value to indicate how many cards are found
SlaveNo: assign slave number (Range 0~1)
SlaveControl: Enable/Disable Slave Module( 1 for enable, 0 for
   disable )
```

@ Return Code

```
ERR_RangeError
ERR_PCIBiosNotExist
ERR_NoError
```

5.3 System Parameters

@ Name

```
_8136_R_Set_RIO_Clk - Set Rio clock divider
_8136_A_Set_DAC_Clk - Set DAC clock divider
_8136_A_Set_ADC_Clk -Set ADC clock divider
_8136_S_Set_Timer_Value - Set Timer click count
_8136_P_Set_Enc_Filter - Set encoder filter clock
```

@ Description

_8136_R_Set_RIO_Clk :

PCI-8136 can connect two remote I/O slave module. This function is for setting module's transmission clock. Assign a clock divider number to change its transmission rate. The maximum transmission clock is about 16.7Mhz and the minimum is about 130Khz for each slave module.

_8136_A_Set_DAC_Clk

There are 6 serial type DA channels in PCI-8136. This function is for setting the DAC transmission clock. Assign a clock divider number to change the DAC transmission rate. The maximum transmission clock is about 8.33 Mhz and the minimum is about 65Khz.

_8136_A_Set_ADC_Clk

There are 6 serial type AD channels in PCI-8136. This function is for setting the ADC transmission clock. Assign a clock divider number to change the DAC transmission rate. The maximum transmission clock is about 8.33 Mhz and the minimum is about 65Khz.

_8136_S_Set_Timer_Value

There is a 24-bits counter in PCI-8136. This function is for setting the counter value and receiving a fixed interrupt interval from this timer when the counting is finished . The timer clock rate is 33.3Mhz. If user set the timer value to be 333000 and the interrupt interval will be 10ms.

_8136_P_Set_Enc_Filter

The encoder counter base clock is 33.3Mhz. This function is for setting the encoder counter filter to fit the pulse rate from users. The maximum value for this filter is 127 and it means user's input pulse rate is smaller than 260Khz.

@ Syntax

C/C++ (DOS, Windows 95/98/NT/2000)

```
I16 _8136_R_Set_RIO_Clk(I16 CardNo, I16 SlaveNo, I16
   Clk_Divider)
I16 _8136_A_Set_DAC_Clk(I16 CardNo, I16 Clk_Divider)
I16 _8136_A_Set_ADC_Clk(I16 CardNo, I16 Clk_Divider)
I16 _8136_S_Set_Timer_Value(I16 CardNo,U32 TimerValue)
I16 _8136_P_Set_Enc_Filter(I16 CardNo,I16 Filter)
```

Visual Basic 5.0 or higher

```
B_8136_R_Set_RIO_Clk(ByVal CardNo As Integer, ByVal SlaveNo As
   Integer, ByVal Clk_Divider As Integer) As Integer
B_8136_A_Set_DAC_Clk(ByVal CardNo As Integer, ByVal
   Clk_Divider As Integer) As Integer
B_8136_A_Set_ADC_Clk(ByVal CardNo As Integer, ByVal
   Clk_Divider As Integer ) As Integer
B_8136_S_Set_Timer_Value(ByVal CardNo As Integer , ByVal
   TimerValue As Long ) As Integer
B_8136_P_Set_Enc_Filter(ByVal CardNo As Integer, ByVal Filter
   As Integer) As Integer
```

@ Arguments

```
CardNo: card number designated to set (Range 0 ~ 3)
SlaveNo: Select which slave would be set ( Range 0~1)
Clk_Divider:  Set transmission clock divider ( Range 0~127 )
TimerValue: Set timer value (Range: 28-Bits)
Filter: Set Filter sample clock divider (Range 0~127)
```

@ Return Code

```
ERR_RangeError
ERR_NoError
```

5.4 Card Information

@ Name

```
_8136_S_Get_IRQ_Channel - Get I/O card's IRQ
_8136_S_Get_Base_Addr - Get I/O card's base address
```

@ Description

_8136_S_Get_IRQ_Channel:

Although PCI-8136 is a PCI interface card, user can use this function to get the IRQ channelwhich is assigned by PCI BIOS. This value is no meaning when operating this card. It is only a part of PCI-813's information

_8136_S_Get_Base_Addr

Although PCI-8136 is a PCI interface card, user can use this function to get the I/O Base Address which is assigned by PCI BIOS. This value is no meaning when operating this card. It is only a part of PCI-813's information

@ Syntax

C/C++ (DOS, Windows 95/98/NT/2000)

```
void _8136_S_Get_IRQ_Channel(I16 cardNo, U16 *irq_no )
void _8136_S_Get_Base_Addr(I16 cardNo, U16 *base_addr )
```

Visual Basic 5.0 or higher

```
B_8136_S_Get_IRQ_Channel(ByVal cardNo As Integer, irq_no As
   Integer)
B_8136_S_Get_Base_Addr(ByVal cardNo As Integer, base_addr As
   Integer )
```

@ Arguments

```
CardNo: card number designated to set (Range 0 ~ 3)
Irq_no: IRQ channel for this card, return value
Base_addr: Base Address for this card, return value
```

@ Return Code

None

5.5 Digital I/O

@ Name

```
_8136_D_Output - Digital output for one bit
_8136_D_Input - Digital input for one bit
_8136_D_OutputA - Digital Output for all bits once
_8136_D_InputA - Digital Input for all bits once
```

@ Description

_8136_D_Output:

There are 7 open collector output channels in PCI-8136. This function is for controlling these output bits by setting them on or off. This function can set each channel individually. Please refer to the pin assignment for bit numbers.

_8136_D_Input

There are 19 open isolated digital input channels in PCI-8136. This function is for getting these input bits by setting them on or off. This function can get each channel individually. Please refer to the pin assignment for bit numbers.

_8136_D_OutputA

There are 7 open collector output channels in PCI-8136. This function is for controlling these output bits by setting them on or off. This function can set all output channels on/off by a value once. Each bit of this value represents the actual output bit. Please refer to the pin assignment for bit numbers.

_8136_D_InputA

There are 19 open isolated digital input channels in PCI-8136. This function is for getting these input bits by setting them on or off. This function can get all input channels' status once. Each bit of this value represents the actual input channel. Please refer to the pin assignment for bit numbers.

@ Syntax

C/C++ (DOS, Windows 95/98/NT/2000)

```
U16 _8136_D_Output( I16 CardNo , I16 Channel , I16 Control );
U16 _8136_D_Input( I16 CardNo , I16 Channel , I16* Control );
U16 _8136_D_OutputA( I16 CardNo , I16 Value );
U16 _8136_D_InputA( I16 CardNo , I32 *Value );
```

Visual Basic 5.0 or higher

```
B_8136_D_Output(ByVal CardNo As Integer, ByVal Channe As
   Integer, ByVal Control As Integer) As Integer
B_8136_D_Input(ByVal CardNo As Integer, ByVal Channel As
   Integer, Control As Integer) As Integer
B_8136_D_OutputA (ByVal CardNo As Integer, ByVal Value As
   Integer) As Integer
B_8136_D_InputA(ByVal CardNo As Integer, Value As Long) As
   Integer
```

@ Argument

```
CardNo: card number designated to set (Range 0 ~ 3)
Channel: channel number designated to set (Range 0 ~ 6)
Control: control value for the designated channel (Range 0 ~
   1)
Value: control value for all digital output (Range 0 ~ 0x7f )
Control: return value for designated channel (Range 0 ~ 1)
Value: return value for all digital input (Range 0 ~ 0x7ffff)
```

@ Return Code

```
ERR_RangeError
ERR_NoError
```

5.6 Remote I/O (Not Released)

@ Name

```
_8136_R_Status - Check Remote I/O Status
_8136_R_Write - Write a word to remote
_8136_R_Read - Read a word from remote
```

@ Description

_8136_R_Status:

This function is for checking remote I/O stopped status. If the remote I/O is not running, user can use this function to check to status of remote I/O. It contains the following information: Remote I/O enable/disable, master/slave receive data status, and module fail.

_8136_R_Write

This function is for writing a word to one set. Each slave module has 4 sets. User must assign the slave number and set number and a value to use this function.

_8136_R_Read

This function is for reading a word from one set. Each slave module has 4 sets. User must assign the slave number and set number and a value to use this function.

@ Syntax

C/C++ (DOS, Windows 95/98/NT/2000)

```
I16 _8136_R_Status(I16 CardNo);
U16 _8136_R_Write(I16 CardNo,I16 SlaveNo,I16 SetNo,U16
   SetValue);
U16 _8136_R_Read(I16 CardNo,I16 SlaveNo,I16 SetNo,U16
   *SetValue);
```

Visual Basic 5.0 or higher

```
B_8136_R_Status (ByVal CardNo As Integer) As Integer
B_8136_R_Write (ByVal CardNo As Integer, ByVal SlaveNo As
   Integer, ByVal SetNo As Integer, ByVal SetValue As Long) As
   Integer
B_8136_R_Read (ByVal CardNo As Integer, ByVal SlaveNo As
   Integer, ByVal SetNo As Integer, SetValue As Long) As
   Integer
```

@ Argument

```
CardNo: card number designated to set (Range 0 ~ 3)
SlaveNo: assign slave number (Range 0~1)
SetNo: I/O range in slave module
   0: bit 0~15
   1: bit 16~31
   2: bit 32~47
   3: bit 48~63
SetValue: 16-bit value to write
*SetValue: 16-bit value to read
```

@ Return Code

```
ERR_RangeError
ERR_NoError
```

5.7 Analog I/O

@ Name

```
_8136_A_Write_Value - Output DAC in value
_8136_A_Write_Volt - Output DAC in voltage
_8136_A_Read_Value - Input from ADC in value
_8136_A_Read_Volt - Input from ADC in voltage
_8136_A_Output_Control - Start or stop DAC output
_8136_A_Set_Trigger - Set DAC output by trigger
_8136_A_Set_Trigger_Map - Select DAC trigger source
_8136_A_Set_Preload_Volt - Set DAC trigger output voltage
_8136_A_Set_Compare_Value - Set ADC compare value
_8136_A_Set_Compare_Volt - Set ADC compare voltage
```

@ Description

_8136_A_Write_Value

This function is for generating a DC value from DAC channel. The resolution of DA converter is 16-bits. The mapping value for 0 volt is 0x0000 , +10 volt is 0x7fff and –10 volt is 0x8000. There are six channels in one card. All channels are free to run individually.

_8136_A_Write_Volt

This function is for generating a DC value from DAC channel. The resolution of DA converter is 16-bits. User can assign a voltage value to this function directly. The range of voltage value is from –10.0 to +10.0. There are six channels in one card. All channels are free to run individually.

_8136_A_Read_Value

This function is for reading a digital value from ADC channel. The resolution of AD converter is 12-bits. User can read a word from this function. The mapping value for 0V is 2000, +10V is 4000 and –10V is 0. There are six channels in one card. All channels are free to run individually.

_8136_A_Read_Volt

This function is for reading a voltage value from ADC channel. The resolution of AD converter is 12-bits. User can read a voltage value from this function directly. The range of the read-back voltage value is for –10.0V to +10.0V. There are six channels in one card. All channels are free to run individually.

_8136_A_Output_Control

Once user wants to stop outputting any voltage from DA channel, he can use this function to close DA channel immediately. It can be open again by this function too.

_8136_A_Set_Trigger

This function is for enable/disable DA channel output by trigger source. The trigger source would be ADC comparator interrupt or encoder counter comparator interrupt.

_8136_A_Set_Trigger_Map

This function is for assigning each channel's trigger source. The trigger source is selected by one parameter. Each bit of this parameter represents one kind of trigger source. Bit0~5 means trigger source is from encoder counter channel 0~5 and bit8~13 means trigger source is from ADC channel 0~5. Set 1 is for enable and 0 is for disable.

_8136_A_Set_Preload_Volt

Once user set the DAC trigger function, the preload voltage must be set first. When the trigger is active, the DAC channel will output this preload value.

_8136_A_Set_Compare_Value

This function is for DAC trigger output or simply for generating ADC comparator interrupt. When user wants to output a preload voltage which is triggered by ADC level, the trigger level must be set by this function. The trigger level is set by a 12-bits ADC value with a range from 0 to 4000.

_8136_A_Set_Compare_Volt

This function is for DAC trigger output or simply for generating ADC comparator interrupt. When user wants to output a preload voltage which is triggered by ADC level, the trigger level must be set by this function. The trigger level is set by a 12-bits ADC voltage with a range from –10.0 to 10.0.

@ Syntax

C/C++ (DOS, Windows 95/98/NT/2000)

```
I16 _8136_A_Write_Value(I16 CardNo, I16 Channel, I16 Value)
I16 _8136_A_Write_Volt(I16 CardNo, I16 Channel, F64 Volt)
I16 _8136_A_Read_Value(I16 CardNo, I16 Channel, I16 *Value)
I16 _8136_A_Read_Volt(I16 CardNo, I16 Channel, F32 *Volt)
I16 _8136_A_Output_Control(I16 CardNo, I16 Channel, I16
   Control)
I16 _8136_A_Set_Compare_Value(I16 CardNo, I16 Channel, I16
   Value)
I16 _8136_A_Set_Compare_Volt(I16 CardNo, I16 Channel, F64
   Volt)
I16 _8136_A_Set_Trigger_Map(I16 CardNo, I16 Channel, I16
   Source)
I16 _8136_A_Set_Trigger(I16 CardNo, I16 Channel, I16 Control)
I16 _8136_A_Set_Preload_Volt(I16 CardNo, I16 Channel, F64
   Volt)
```

Visual Basic 5.0 or higher

```
B_8136_A_Write_Value (ByVal CardNo As Integer, ByVal Channel
   As Integer, ByVal Value As Integer) As Integer
B_8136_A_Write_Volt (ByVal CardNo As Integer, ByVal Channel As
   Integer, ByVal Volt As Single) As Integer
B_8136_A_Read_Value Lib (ByVal CardNo As Integer, ByVal
   Channel As Integer, Value As Integer) As Integer
B_8136_A_Read_Volt (ByVal CardNo As Integer, ByVal Channel As
   Integer, Volt As Double) As Integer
B_8136_A_Output_Control (ByVal CardNo As Integer, ByVal
   Channel As Integer, ByVal Control As Integer) As Integer
B_8136_A_Set_Compare_Value(ByVal CardNo As Integer, ByVal
   Channel As Integer, ByVal Value As Integer) As Integer
B_8136_A_Set_Compare_Volt (ByVal CardNo As Integer, ByVal
   Channel As Integer, ByVal Volt As Double) As Integer
B_8136_A_Output_Control (ByVal CardNo As Integer, ByVal
   Channel As Integer, ByVal Control As Integer) As Integer
B_8136_A_Set_Trigger_Map(ByVal CardNo As Integer,ByVal
   Channel As Integer, ByVal Source As Integer) As Integer
B_8136_A_Set_Trigger(ByVal CardNo As Integer, ByVal Channel As
   Integer, ByVal Control As Integer) As Integer
B_8136_A_Set_Preload_Volt(ByVal CardNo As Integer, ByVal
   Channel As Integer, ByVal Volt As Double) As Integer
```

@ Argument

CardNo: card number designated to set (Range 0 ~ 3)
Channel: channel number designated to set (Range 0 ~ 6)
Value: the output value for DAC channel (Range–32768 ~ +32767)
Volt: the output voltage for DAC channel (Range –10.0 ~ +10.0)
Value: the input value for ADC channel (Range 0 ~4000)
Volt: the input voltage for ADC channel (Range –10.0 ~ +10.0)
Control: enable or disable trigger (1 for enable/0 for disable)
Source: Set DAC trigger source
Value 0~5 is for encoder 0~5
Value 8~13 is for ADC channel 0~5

@ Return Code

ERR_RangeError
ERR_NoError

5.8 Pulse I/O

@ Name

```
_8136_P_Set_Output_Type - Set pulse output mode
_8136_P_Set_Input_Type - Set pulse input mode
_8136_P_Read - Read encoder counter
_8136_P_Clear - Clear encoder counter
_8136_P_Send - Send a constant pulse train
_8136_P_Stop - Stop pulse train
_8136_P_Change_Speed - Change pulse train frequency
_8136_P_Read_Index - Read index value
_8136_P_Set_Index_Latch - Set index latch type
_8136_P_Read_Latch_Value - Read a latched encoder data
_8136_P_Set_Compare_Value - Set a encoder compare data
```

@ Description

_8136_P_Set_Output_Type:

There are 3 pulse output types in PCI-8136. This function is for configuring pulse output type by a value. Write a value 0 is for pulse/direction type. Write a value 1 is for CW/CCW type. Write a value 2 is for A/B phase type.

_8136_P_Set_Input_Type

There are 3 encoder counter input types in PCI-8136. This function is for configuring encoder counter input type. Write a value 0 is for A/B phase type. Write a value 1 is for CW/CCW type. Write a value 2 is for pulse/direction type. When setting A/B phase type, user must assign the multiplier value by this function.

_8136_P_Read

This function is for reading the 32-bits encoder counter value immediately. There are six encoder counters in one card. Assign the channel value 0~5 to read the encoder counter individually. There are three internal counters for receiving pulse output command. When pulse output channel 0~2 is working, their output value will send to both CN1 and internal feedback counter. These three command feedback counters are at channel 6~8. If the channel parameter of this function is assigned as the above three channels, it will read the command value at the same time.

_8136_P_Clear

This function is for clearing the encoder counter value to zero immediately.

_8136_P_Send

This function is for sending a fixed frequency pulse train of each channel. It will output a pre-configured pulse format which is set by P_Set_Output_Type().

_8136_P_Stop

This function is for stopping the output pulse for each channel.

_8136_P_Change_Speed

This function is for changing the output pulse frequency on line.

_8136_P_Read_Index

There are 6 index signal input in one PCI-8136. This function is for checking the index status on or off. Each bit of this status value represents a index status.

_8136_P_Set_Index_Latch

The index signal is also a trigger source for latching the respective encoder counter value. There are two modes for this latch. Set 0 to be first trigger latch and set 1 to be last trigger latch. First trigger latch means only trigger once and last trigger latch means latch every time if index signal comes.

_8136_P_Read_Latch_Value:

Once the encoder counter is latched. Use this function can get the counter value at latched moment. It will not be clear until next index latched signal is coming.

_8136_P_Set_Compare_Value

There are 6 encoder counters in PCI-8136. Each encoder can set a compare value individually. This compare value is also a 32-bits value.

@ Syntax

C/C++ (DOS, Windows 95/98/NT/2000)

```
I16 _8136_P_Set_Output_Type(I16 CardNo, I16 ChannelNo, I16
   PulseFmt);
I16 _8136_P_Set_Input_Type(I16 CardNo, I16 EncNo, I16 EncFmt,
   I16 Mul);
I16 _8136_P_Read(I16 CardNo, I16 EncNo, I32 *EncData);
I16 _8136_P_Send(I16 CardNo,I16 ChannelNo,F64 FrqL);
I16 _8136_P_Stop(I16 CardNo,I16 ChannelNo);
I16 _8136_P_Clear(I16 CardNo, I16 EncNo);
I16 _8136_P_Set_Compare_Value(I16 CardNo,I16 EncNo, I32
   CompValue);
I16 _8136_P_Read_Latch_Value(I16 CardNo, I16 EncNo, I32
   *Value)
I16 _8136_P_Set_Index_Latch(I16 CardNo, I16 WhichIndex, I16
   Type);
I16 _8136_P_Read_Index(I16 CardNo, I16 AxisNo, I16 *Index);
I16 _8136_P_Change_Speed(I16 CardNo,I16 AxisNo,F32 Frq);
```

Visual Basic 5.0 or higher

```
B_8136_P_Initial (ByVal CardNo As Integer) As Integer
B_8136_P_Set_Output_Type (ByVal CardNo As Integer, ByVal
   AxisNo As Integer, ByVal PulseFmt As Integer) As Integer
B_8136_P_Set_Input_Type (ByVal CardNo As Integer, ByVal EncNo
   As Integer, ByVal EncFmt As Integer, ByVal Mul As Integer)
   As Integer
B_8136_P_Read (ByVal CardNo As Integer, ByVal EncNo As Integer,
   EncData As Long) As Integer
B_8136_P_Send (ByVal CardNo As Integer, ByVal AxisNo As Integer,
   ByVal FrqL As Double) As Integer
B_8136_P_Stop (ByVal CardNo As Integer, ByVal AxisNo As Integer)
   As Integer
B_8136_P_Clear (ByVal CardNo As Integer, ByVal EncNo As Integer)
   As Integer
B_8136_P_Set_Compare_Value (ByVal CardNo As Integer, ByVal
   EncNo As Integer, ByVal CompValue As Long) As Integer
B_8136_P_Read_Latch_Value(ByVal CardNo As Integer, ByVal
   EncNo As Integer, Value As Long) As Integer
B_8136_P_Set_Index_Latch (ByVal CardNo As Integer, ByVal
   WhichIndex As Integer, ByVal LatchType As Integer) As
   Integer
B_8136_P_Read_Index (ByVal CardNo As Integer, ByVal AxisNo As
   Integer, Index As Integer) As Integer
B_8136_P_Change_Speed (ByVal CardNo As Integer, ByVal AxisNo
   As Integer, ByVal Frq As Single) As Integer
```

@ Argument

```
CardNo: card number designated to set (Range 0 ~ 3)
ChannelNo: channel number designated to set (Range 0 ~ 6)
EncNo: encoder channel number designated to set (Range 0 ~ 6)
PulseFmt: Output pulse format
0 = Pulse/Direction
1 = CW/CCW
2 = A/B Phase
EncFmt: Input pulse format:
0 = A/B Phase
1 = CW/CCW
2 = Pulse/Direction
Mul: for A_B type's mutiplier
0 = 0X A_B Phase
1 = 1X A_B Phase
2 = 2X A_B Phase
3 = 4X A_B Phase
EncData: read back encoder data
FrqL: Pulse output frequency (Range 0~500k Hz)
CompValue: Encoder Compare Value
*Index:  Index Status (0 or 1)
Type: two trigger latch mode: 0 for first trigger, 1 for last
   trigger
WhichIndex: select index no. (0~5)
Frq: Pulse output frequency for change (Range 0~500k Hz)
```

@ Return Code

```
ERR_RangeError
ERR_PCIBiosNotExist
ERR_NoError
```

5.9 Interrupt

@ Name

```
_8136_INT_Enable - Set interrupt event handler
_8136_INT_Disable - Remove int. event handler
_8136_S_Set_Int_Factor - Set interrupt factor
_8136_S_INT_Control - Enable/disable interrupt
_8136_S_Get_Int_Status - Get Int. status
_8136_Callback_Function - Set a call back function for int
```

@ Description

_8136_INT_Enable

This function is only for Windows system. It allocates interrupt events in Windows system for passing message to application from kernel.

_8136_INT_Disable

This function is only for Windows system. It clear event resources which is allocated by INT_Enable() function.

_8136_S_Set_Int_Factor

This function is for setting interrupt source for each channel. Every channel has 6 types of interrupt can be set. It includes four digital input interrupt, index interrupt, encoder compare interrupt, timer interrupt and voltage compare interrupt.

_8136_S_INT_Control

This function controls the hardware interrupt pin. The interrupt won't come if this function hasn't used.

_8136_S_Get_Int_Status

Once the interrupt comes, user must use this function to check the interrupt types. There are 6 types of interrupt at one interrupt event. User must check what kinds of interrupt is coming this time. Two or more interrupt will comes at the same time. User must check each it to prevent lost any interrupt events.

_8136_Callback_Function

This function can set a user-defined function to be an ISR. When one interrupt comes, this user-defined function will wake up at the same time. Be careful to deal with this function. Don't take too much time inside this function.

@ Syntax

C/C++ (DOS, Windows 95/98/NT/2000)

```
U16_8136_INT_Enable(I16 CardNo, HANDLE *phEvent); (Windows
   Only)
U16 _8136_INT_Disable(I16 CardNo);  (Windows Only)
void _8136_S_INT_Control(I16 CardNo, U16 intFlag )
U16 _8136_S_Set_Int_Factor(I16 CardNo, I16 ChannelNo, U16
   IntFactor)
U16 _S_Get_Int_Status(I16 CardNo,I16 AxisNo,U16 *IntStatus)
void _8136_Callback_Function(I16 CardNo,void
   *callbackAddr(I16 AxisNo, U16 IntSts))
```

Visual Basic 5.0 or higher

```
B_8136_INT_Enable (ByVal card_number As Integer, phEvent As
   Long) As Integer
B_8136_INT_Disable (ByVal card_number As Integer) As Integer
B_8136_S_INT_Control (ByVal CardNo As Integer, ByVal intFlag
   As Integer)
B_8136_S_Set_Int_Factor (ByVal CardNo As Integer, ByVal AxisNo
   As Integer, ByVal IntFactor As Integer, ByVal OptionType As
   Integer) As Integer
B_8136_S_Get_Int_Status (ByVal CardNo As Integer, ByVal AxisNo
   As Integer, IntStatus As Long) As Integer
B_8136_Callback_Function (ByVal CardNo As Integer, ByVal
   lpCallBackProc As Long) As Integer
```

@ Arguments

```
CardNo: card number designated to set (Range 0 ~ 3)
ChannelNo: channel number designated to set (Range 0 ~ 6)
*phEvent: event handler array contains 7 handler for each card
   in Windows interrupt system
*existCards: a return value to indicate how many cards are found
intFlag: enable or disable interrupt signal (Range 0~1)
intFactor: enable or disable interrupt for each type in each
   bit:
bit0: Limit switch on
bit1: Emergency stop on
bit2: Home switch on
bit3: Index signal on
bit4: Encoder value compared
bit5: Timer interrupt
bit6: ADC pre-load value reached
```

*IntStatus: Read the interrupt status of one axis. To adjudge which interrupt is coming.
bit0: Positive Limit switch on
bit1: Minus Limit switch on
bit2: Emergency stop on
bit3: Home switch on
bit4: Index signal on
bit5: Encoder value compared
bit6: Timer interrupt
bit7: ADC pre-load value reached
OptionType: Set ADC compare direction
1 = Rising Direction
2 = Falling Driection
3 = Both Direction
*callbackAddr(I16 AxisNo, U16 IntSts) : function pointer type

@ Return Code

ERR_RangeError
ERR_NoError

Product Warranty/Service

Seller warrants that equipment furnished will be free form defects in material and workmanship for a period of one year from the confirmed date of purchase of the original buyer and that upon written notice of any such defect, Seller will, at its option, repair or replace the defective item under the terms of this warranty, subject to the provisions and specific exclusions listed herein.

This warranty shall not apply to equipment that has been previously repaired or altered outside our plant in any way as to, in the judgment of the manufacturer, affect its reliability. Nor will it apply if the equipment has been used in a manner exceeding its specifications or if the serial number has been removed.

Seller does not assume any liability for consequential damages as a result from our products uses, and in any event our liability shall not exceed the original selling price of the equipment.

The equipment warranty shall constitute the sole and exclusive remedy of any Buyer of Seller equipment and the sole and exclusive liability of the Seller, its successors or assigns, in connection with equipment purchased and in lieu of all other warranties expressed implied or statutory, including, but not limited to, any implied warranty of merchant ability or fitness and all other obligations or liabilities of seller, its successors or assigns.

The equipment must be returned postage-prepaid. Package it securely and insure it. You will be charged for parts and labor if you lack proof of date of purchase, or if the warranty period is expired.