ADLINK
TECHNOLOGY INC.

PCI-8124

4 Channel Encoder
Compare and Trigger Board
User's Manual

Manual Rev.	2.00
Revision Date:	February 12, 2007
Part No:	50-11138-1000

Recycled Paper

Advance Technologies; Automate the World.

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.

Trademarks

NuDAQ, NuIPC, DAQBench are registered trademarks of ADLINK TECHNOLOGY INC.

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 top priority for ADLINK Technology Inc. Please contact us should you require any service or assistance.

ADLINK TECHNOLOGY INC.

Web Site:	http://www.adlinktech.com
Sales & Service:	Service@adlinktech.com
TEL:	+886-2-82265877
FAX:	+886-2-82265717
Address:	9F, No. 166, Jian Yi Road, Chungho City, Taipei, 235 Taiwan

Please email or FAX this completed service form for prompt and satisfactory service.

Company Information		
Company/Organization		
Contact Person		
E-mail Address		
Address		
Country		
TEL	FAX:	
Web Site		
Product Information		
Product Model		
Environment	OS: M/B: Chipset:	 CPU: BIOS:

Please give a detailed description of the problem(s):

Table of Contents

List of Tables

List of Figures

1 Introduction

The PCI-8124 is an advanced quadrature encoder counter board with position compare and high speed trigger pulse output capability. With a state-of-the-art of FPGA, it can meet most machine maker requirements – high frequency encoder input and trigger pulse output.

Typically, most machine makers design motion control systems and vision inspection systems individually. Specifically, line scan application, the frame grabber needs high frequency of trigger pulse input to scan the whole image line-by-line. The more intensive lines, the more precise images users can get. This means users will have this demand for high frequency of trigger pulse input. With the advantage of integrating the trigger board and frame grabber together, the vision system completely retrieves the encoder information from servo motors or linear scale and the trigger pulse signal can be wired by flat cable internally. It is convenient for system integrators and also reduces the possibility of signal interference. With FPGA calculation, it can also save CPU loading of host PC.

The PCI-8124 utilizes a FPGA and FIFO to realize the purpose. All comparing points will automatically load into FIFO and be compared with the internal comparators. As a result, users do not have to handle this with software. Besides, the trigger pulse frequency can be up to 5MHz and encoder input frequency can be up to 10MHz. With adjustable pulse width, users can change this with software control as they wish. Furthermore, the relation between comparator and trigger output is mutually corresponding – One comparator to multi-trigger outputs or Multi-comparators to one trigger output.

The PCI-8124 offers the board index design. This can let users have one unique index number in the PC system. For input and output channel, this board provides two types – TTL (without isolation) and Open Collector (with isolation). The former one can meet high frequency demand; the latter one can let users have circuit protection.

ADLINK provides a Windows WDM driver for users. Users can use DLL function library to build up the application. The block diagram of PCI-8124 is as follows.

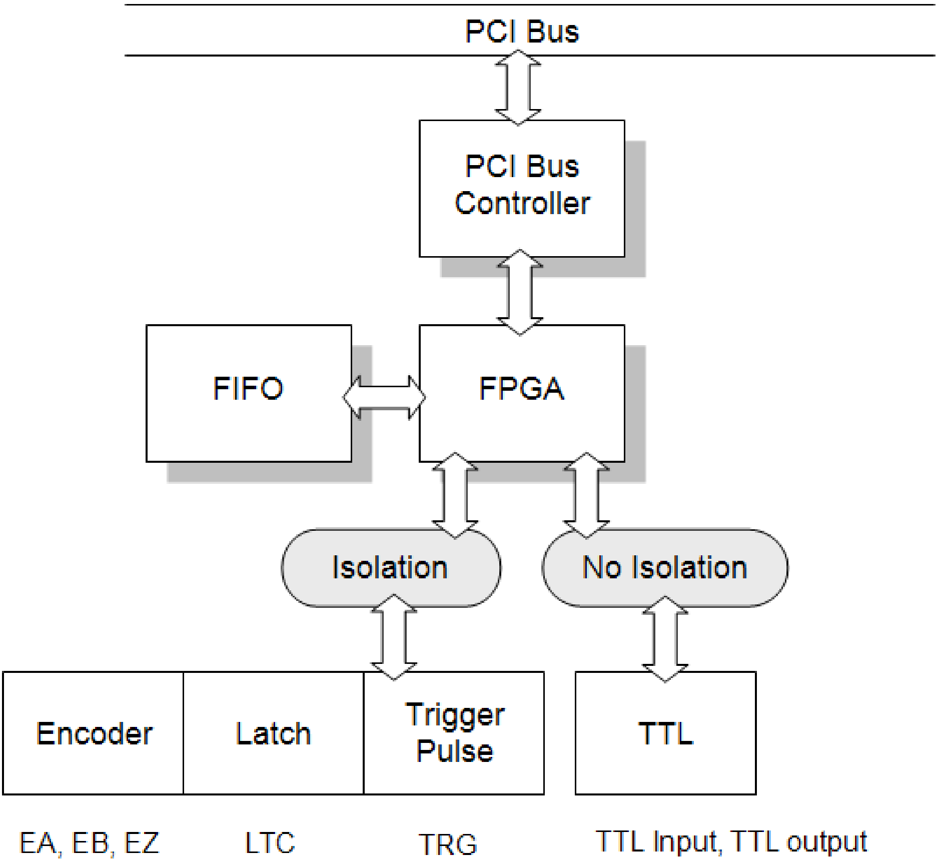


Figure 1-1: PCI-8124 Block Diagram

1.1 Features

- PCI Rev 2.2, 32-bit/33Mhz Compliment
- Support both PCI-X (3.3V) and PCI (5V) slots
- Support 4-bit board ID for multiple boards in one system
- 4 differential quardrature encoder counter input channels
- Each encoder counter input pair supports 1/2/4x A/B phase, dual pulse (CW/CCW) and signed pulse (OUT/DIR) modes
- 4 trigger output channels via both open collector and TTL pins
- Trigger pulse width is software programmable from 0.1µs to 3.2765ms
- Each trigger output channel is selected from all comparators, manual trigger commands, EZ signal and timers by demands
- 4 comparators for comparing encoder counter and FIFO data
- 10 comparators for comparing encoder counter and linear data
- Each encoder counter source is selected from encoder input pins or timers
- Comparator data is updated by next FIFO data or linear data when encoder counter is compared and trigger is outputted
- 14 comparators can select one of 4 trigger output channels individually
- 4 channel TTL output pins for general purpose output or trigger output
- 4 channel TTL input pins for general purpose or timer start signal
- 4 channel high speed latch input pins for counters
- EZ and Latch input pins can be used for general purpose input
- Latch input logic is selectable by rising or falling edge
- Encoder counter clear via EZ input pin as zero operation
- Programmable interrupt sources from linear data finished, triggered, FIFO empty/full/low, latched, TTL input on.
- Switch setting for trigger output default level while power on
- Trigger output pin logic programmable
- Provide DLL and driver to support Windows 2K/XP/Vista

1.2 Specifications

Main Chip

- Chip Vendor: ACTEL
- Model No: A3P1000
- Power usage: 3.3V and 1.5V
- Reference clock: 40Mhz

Encoder Counters:

- Number of channels: 4
- Type of wiring: dual differential pairs or single-ended pair
- Input Isolated photo coupler: PS9115
- Isolated Voltage: 2500 Vrms
- Maximum input frequency: 10 MHz
- Supported modes: AB phase x1/2/4, CW/CCW, OUT/DIR
- Data length: 32-bit
- Zero Operation: EZ input

Counter Comparators:

- Number of comparators: 14
- Data length: 32-bit
- Trigger pulse width: 0.1µs to 3.2765ms programmable

Internal Timers:

- Number of timers: 4
- Timer base: 400ns
- Resolution: 14-bit
- Synchronous start signal: TTL input or register setting
- Synchronous stop signal: register setting

Trigger Output:

- Number of channels: 4
- Maximum output frequency: 5Mhz
- Output types: Open collector or TTL
- Open collector Isolated photo-coupler: PS9115
- Open collector Isolated: 2500 Vrms
- Continuous triggering method: Timer, FIFO or Linear function data reloading
- Interrupt: Yes, when triggered

Comparator FIFO:

- Number of FIFO: 4
- FIFO depth: 1023x32bits
- Interrupt type: Empty/full/low (1/4,1/2,3/4)
- Auto reload: Yes, auto reload next FIFO data to comparator when compared

Comparator Linear Function

- Number of Linear Functions: 10
- Start Address length: 32-bits
- Repeat times length: 32-bit
- Interval length: 13-bit
- Direction length: 1-bit
- Interrupt: Yes, when repeat times count to 0
- Auto reload: Yes, auto reload next linear data to comparator when compared

Latch Input

- Number of Latch Input pins: 4
- Input isolated photo-coupler: PS9115
- Isolated Voltage: 2500 Vrms
- Band width: 10Mhz
- Interrupt: Yes, when latched

TTL Input

- Number of input pins: 4
- Photo-coupler Isolated: No
- TTL input mode: General purpose or timer synchronous start signal

TTL output

- Number of output pins: 4
- Photo-coupler Isolated: No
- TTL output mode: General purpose output or trigger output

General Specifications

- Connectors: 50-pin SCSI-type connector
- Operating Temperature: 0°C - 70°C
- Storage Temperature: -20°C - 80°C
- Humidity: 5 - 85%, non-condensing

Power Consumption

- Slot power supply (input): +5V DC ±5%, 450mA(max)
- Slot power supply (input): +12V DC ±5%, 100mA(max)
- Isolation power is from PCI bus (+12V to isolated +5V DC)

PCB Dimension: 151mm(L) X 107mm(W)

1.3 Supported Software

1.3.1 Programming Library

ADLINK provides Windows WDM driver and DLL function library for the PCI-8124. These function libraries are shipped with the board. It supports for Windows 2000/XP/Vista.

2 Installation

This chapter describes how to install the PCI-8124. Please follow these steps below:

- 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 I/O signal connections (Chapter 3) and their operation (Chapter 4)

2.1 Package Contents

Except this User's Guide, the package also includes the following items:

- PCI-8124: 4 channel encoder compare trigger board
- ADLINK All-in-one Compact Disc

If any of these items are missing or damaged, contact the dealer from whom you purchased the product. Save the shipping materials and carton to ship or store the product in the future.

2.2 PCI-8124 Outline Drawing

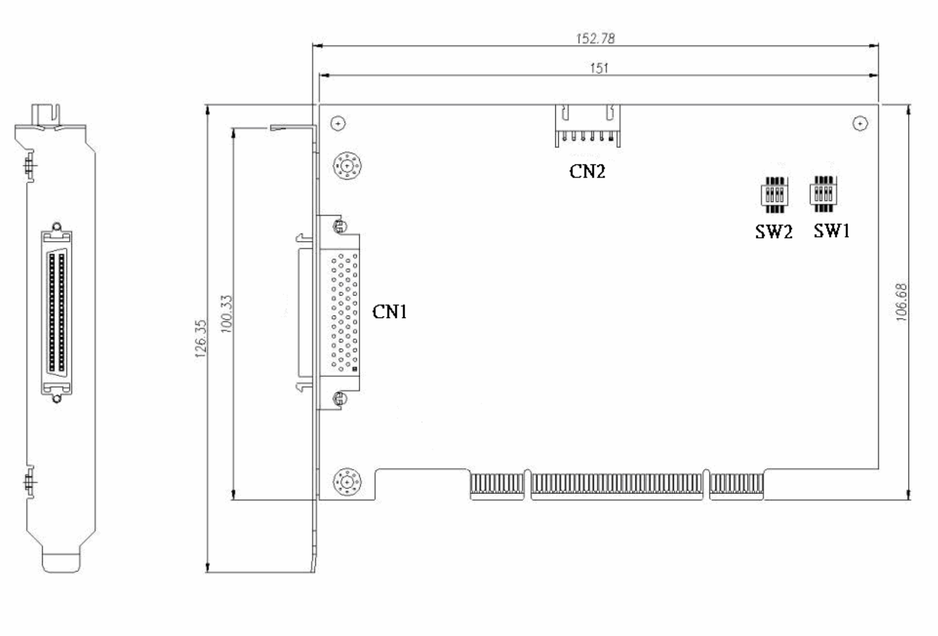


Figure 2-1: PCI-8124 PCB Layout

CN1: SCSI 50 pins Connector

CN2: TTL output

SW1: Board ID switch (ID: 0 to 15)

SW2: Default Trigger Level

2.3 PCI-8124 Hardware Installation

2.3.1 Hardware configuration

The PCI-8124 is fully Plug and Play compliant. Hence memory allocation (I/O port locations) and IRQ channel of the PCI card are assigned by the 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 system may have both PCI and ISA slots. Do not force the PCI card into a PC/AT slot. The PCI-8124 can be used in any PCI slot. It also supports dual voltage PCI bus.

2.3.3 Installation Procedures

1. Read through this manual and setup the board index with SMD switch. The ID range is from 0 to 15.
2. Turn off your computer. Turn off all accessories (printer, modem, monitor, etc.) connected to computer. Remove the cover from your computer.
3. Select a 32-bit PCI expansion slot. PCI slots are shorter than ISA or EISA slots and are usually white or ivory.
4. Before handling the PCI-8124, discharge any static buildup on your body by touching the metal case of the computer. Hold the edge of the card and do not touch the components.
5. Position the board into the PCI slot you have selected.
6. Secure the card in place at the rear panel of the system unit using screws removed from the slot.

2.3.4 Troubleshooting:

If your system doesn’t boot or if you experience erratic operation with your PCI board in place, it’s most likely caused by an interrupt conflict (possibly an incorrect ISA setup). In general, the solution, once determined it is not a simple oversight, is to consult the BIOS documentation that comes with your system.

Check the control panel of the Windows system if the card is listed by the system. If not, check the PCI settings in the BIOS or use another PCI slot.

2.4 Software Driver Installation

Using the All-In-One CD with PCI-8124 package and execute the following steps:

1. Autorun the ADLINK All-In-One CD.
2. Follow the procedures of the installer.
3. After setup installation is completed, restart windows.

2.5 CN1 Pin Assignments

CN1 is the 50-pin SCSI type connector for I/O signals.

No.	Name	I/O	Function	No.	Name	I/O	Function
1	INCOM1	-	Latch input common	26	INCOM3	-	Latch input common
2	LTC1	I	Latch input	27	LTC3	I	Latch input
3	INCOM2	-	Latch input	28	INCOM4	-	Latch input common
4	LTC2	I	Latch input	29	LTC4	I	Latch input
5	EGND	-	Trigger output common	30	EGND	-	Latch input common
6	TRG1	O	Trigger output 1	31	TRG3	O	Trigger output 3
7	EGND	-	Trigger output common	32	EGND	-	Trigger output common
8	TRG2	O	Trigger output 2	33	TRG4	O	Trigger output 4
9	EA1+	I	A-phase encoder (+)	34	EA3+	I	A-phase encoder (+)
10	EA1-	I	A-phase encoder (-)	35	EA3-	I	A-phase encoder (-)
11	EB1+	I	B-phase encoder (+)	36	EB3+	I	B-phase encoder (+)
12	EB1-	I	B-phase encoder (-)	37	EB3-	I	B-phase encoder (-)
13	EZ1+	I	Z-phase encoder (+)	38	EZ3+	I	Z-phase encoder (+)
14	EZ1-	I	Z-phase encoder (-)	39	EZ3-	I	Z-phase encoder (-)
15	EA2+	I	A-phase encoder (+)	40	EA4+	I	A-phase encoder (+)
16	EA2-	I	A-phase encoder (-)	41	EA4-	I	A-phase encoder (-)
17	EB2+	I	B-phase encoder (+)	42	EB4+	I	B-phase encoder (+)
18	EB2-	I	B-phase encoder (-)	43	EB4-	I	B-phase encoder (-)
19	EZ2+	I	Z-phase encoder (+)	44	EZ4+	I	Z-phase encoder (+)
20	EZ2-	I	Z-phase encoder (-)	45	EZ4-	I	Z-phase encoder (-)
21	TTL-IN1	I	TTL input signal	46	TTL-IN3	I	TTL input signal
22	TTL-IN2	I	TTL input signal	47	TTL-IN4	I	TTL input signal
23	TTL-OUT1	O	TTL output signal	48	TTL-OUT3	O	TTL output signal
24	TTL-OUT2	O	TTL output signal	49	TTL-OUT4	O	TTL output signal
25	DGND	-	TTL signal ground	50	DGND	-	TTL signal ground

Table 2-1: CN1 Pin Assignments

2.6 CN2 Pin Assignments: TTL Output

CN2 is a 6-pin header as follows:

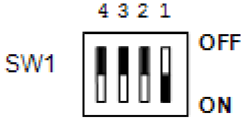
No.	Name	I/O	Function
1	+5V	-	+5V power supply
2	TTL-OUT1	O	TTL output signal
3	TTL-OUT2	O	TTL output signal
4	TTL-OUT3	O	TTL output signal
5	TTL-OUT4	O	TTL output signal
6	DGND	-	TTL signal ground

Table 2-2: CN2 Pin Assignments: TTL Output

Note: +5V and DGND pins are provided by the PCI-Bus. Therefore, these signals are not isolated.

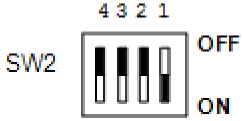
2.7 Board Index Switch Setting

The SW1 switch is used to set board unique index. Its range is from 0 to 15. For example, the board index is 1 if you switch on pin 1 and switch off pin 2, 3 and 4.



2.8 Default Trigger Level

The SW2 switch is used to set TRG pin's default level when power on. Each bit represents one TRG channel. "ON" position means default level of TRG channel is set to +5V and OFF position means it is set to 0V. Manufacturing setting is ON.



3 Signal Connections

Signal connections of all I/O’s are described in this chapter. Refer to the contents of this chapter before wiring any cables between the PCI-8124 and the end device.

This chapter contains the following sections:

Section 3.1	Connector (CN1) Pin Definition
Section 3.2	Isolated Input Ports (LTC, INCOM)
Section 3.3	Isolated Encoder Differential Input
Section 3.4	Isolated Output Ports (TRG, EGND)
Section 3.5	TTL Input Ports
Section 3.6	TTL Output Ports

3.1 Connector (CN1) Pin Definition

There are 4 Isolated Latch Inputs, 4 A-B-Z Encoder Inputs, 4 TTL level Inputs, 4 TTL level outputs and 4 Isolated Outputs on PCI-8124. The following table shows the pin assignment of CN1.

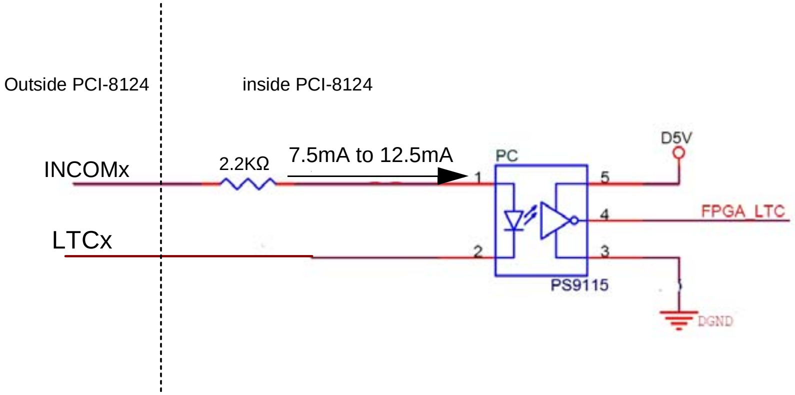
CN1No.	Name	I/O	Function
1	INCOM1	-	Latch input common
2	LTC1	I	Latch input
3	INCOM2	-	Latch input
4	LTC2	I	Latch input
5	EGND	-	Trigger output reference
6	TRG1	O	Trigger output
7	OUTCOM2	-	Trigger output reference
8	TRG2	O	Trigger output
9	EA1+	I	A-phase encoder (+)
10	EA1-	I	A-phase encoder (-)
11	EB1+	I	B-phase encoder (+)
12	EB1-	I	B-phase encoder (-)
13	EZ1+	I	Z-phase encoder (+)
14	EZ1-	I	Z-phase encoder (-)
15	EA2+	I	A-phase encoder (+)
16	EA2-	I	A-phase encoder (-)
17	EB2+	I	B-phase encoder (+)
18	EB2-	I	B-phase encoder (-)
19	EZ2+	I	Z-phase encoder (+)
20	EZ2-	I	Z-phase encoder (-)
21	TTL-IN1	I	TTL input signal
22	TTL-IN2	I	TTL input signal
23	TTL-OUT1	O	TTL output signal
24	TTL-OUT2	O	TTL output signal
25	DGND	-	TTL signal ground
26	INCOM3	-	Latch input common
27	LTC3	I	Latch input
28	INCOM4	-	Latch input common

CN1No.	Name	I/O	Function
29	LTC4	I	Latch input
30	EGND	-	Latch input reference
31	TRG3	O	Trigger output
32	EGND	-	Trigger output reference
33	TRG4	O	Trigger output
34	EA3+	I	A-phase encoder (+)
35	EA3-	I	A-phase encoder (-)
36	EB3+	I	B-phase encoder (+)
37	EB3-	I	B-phase encoder (-)
38	EZ3+	I	Z-phase encoder (+)
39	EZ3-	I	Z-phase encoder (-)
40	EA4+	I	A-phase encoder (+)
41	EA4-	I	A-phase encoder (-)
42	EB4+	I	B-phase encoder (+)
43	EB4-	I	B-phase encoder (-)
44	EZ4+	I	Z-phase encoder (+)
45	EZ4-	I	Z-phase encoder (-)
46	TTL-IN3	I	TTL input signal
47	TTL-IN4	I	TTL input signal
48	TTL-OUT3	O	TTL output signal
49	TTL-OUT4	O	TTL output signal
50	DGND	-	TTL signal ground

3.2 Isolated Input Ports (LTC, INCOM)

There are 4 pairs of isolated input ports on PCI-8124. Each pair has two terminals. One is INCOMx, and the other is LTCx. Because the input port of the photo coupler utilized on PCI-8124 is single-directional, LTC signal must be lower than INCOM signal. When there is a differential voltage between LTCx and corresponding INCOMx, the photo coupler will turn on immediately. There is an internal resistor in the PCI-8124 circuit as figure shows below. So it is not necessary to attach additional resistor connected outside the PCI-8124. The forward current of input signal is 7.5mA to 12.5mA. The typical forward voltage is 1.65V, and the maximum forward voltage is 1.9V. So users should calculate the appropriate voltage applied on LTCx and INCOMx.

CN1 Pin No.	Name	Description
1	INCOM1	In1 Com
3	INCOM2	In2 Com
30	INCOM3	In3 Com
32	INCOM4	In4 Com
2	LTC1	Isolated Latch 1
4	LTC2	Isolated Latch 2
27	LTC3	Isolated Latch 3
29	LTC4	Isolated Latch 4



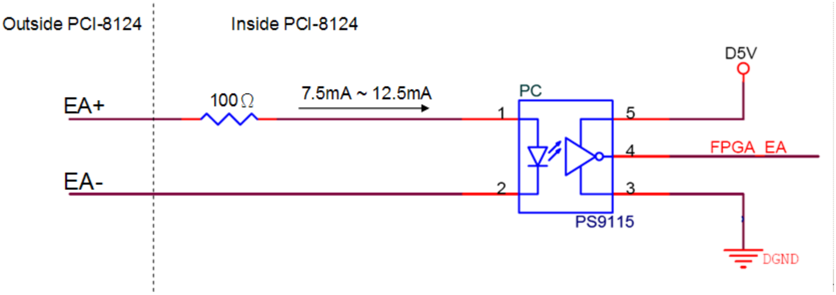
3.3 Isolated Encoder Differential Input

There are 4 pairs of A-B-Z encoder signal inputs on PCI-8124. The pin assignment of these encoder inputs are listed in the following table.

The typical forward voltage of the input diode is 1.65V. And the forward current of the input diode is 7.5mA to 12.5mA. There is an internal resistor in the PCI-8124 circuit as figure shows below. So if there is no additional resistor connected outside the PCI-8124, the voltage difference between Encoder+ and Encoder- should be lower than 5VDC.

CN1 Pin No.	Name	Description
9, 15, 34, 40	EAn+	Isolated Encoder A +
10, 16, 35, 41	EAn-	Isolated Encoder A -
11, 17, 36, 42	EBn+	Isolated Encoder B +
12, 18, 37, 43	EBn-	Isolated Encoder B -
13, 19, 38, 44	EZn+	Isolated Encoder Z +
14, 20, 39, 45	EZn-	Isolated Encoder Z -

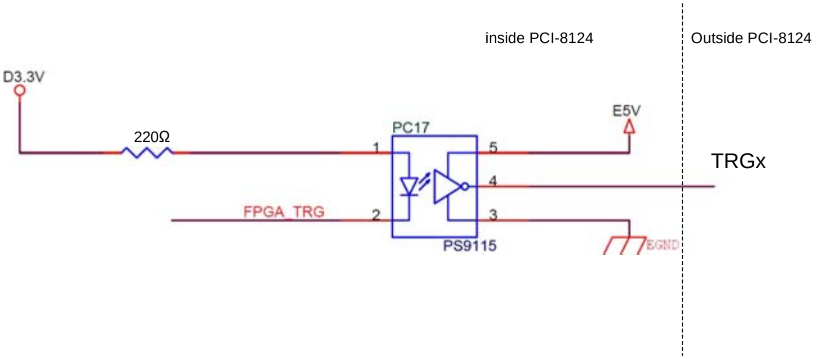
Note: n=1, 2, 3, 4

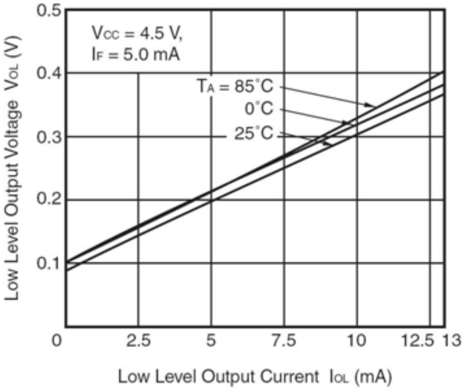


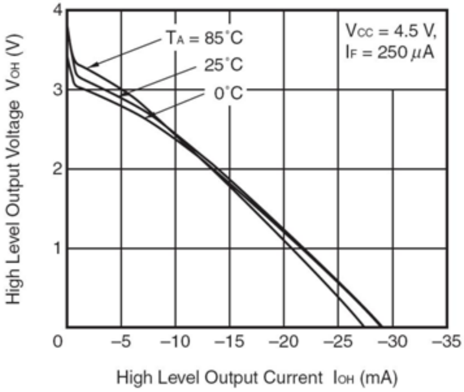
3.4 Isolated Output Ports (TRG, EGND)

There are 4 independent isolated output ports on PCI-8124. Because the use of photo coupler PS9115, the output pull high is not needed. The pin assignment of the isolated output ports on PCI-8124 is as follows, and PS9115 output characteristics are described in the following figure.

CN1 Pin No.	Name	Description
5	EGND	Output reference
7	EGND	Output reference
30	EGND	Output reference
32	EGMD	Output reference
6	TRG1	Trigger 1
8	TRG2	Trigger 2
31	TRG3	Trigger 3
33	TRG4	Trigger 4


inside PCI-8124
Outside PCI-8124
D3.3V
220Ω
PC17
E5V
1
5
4
2
3
TRGx
FPGA_TRG
PS9115
EGND


Vcc = 4.5 V,
IF = 5.0 mA
TA = 85˚C
0˚C
25˚C
0.5
0.4
0.3
0.2
0.1
0
2.5
5
7.5
10
12.5
13
Low Level Output Voltage VOL (V)
Low Level Output Current IOL (mA)


TA = 85˚C
25˚C
0˚C
Vcc = 4.5 V,
IF = 250 μA
4
3
2
1
0
−5
−10
−15
−20
−25
−30
−35
High Level Output Voltage VOH (V)
High Level Output Current IOH (mA)

3.5 TTL Input Ports

There are 4 single ended TTL-level input ports on PCI-8124. The following table shows the pin assignment of these TTL inputs.

CN1 Pin No.	Name	Description
21	TTL-IN1	TTL Level Input port 1
22	TTL-IN2	TTL Level Input port 2
46	TTL-IN3	TTL Level Input port 3
47	TTL-IN4	TTL Level Input port 4
25, 50	DGND	TTL Input reference GND

The maximum input voltage of TTL-level input ports is 4.6VDC. So user should not apply signals over this limitation.

3.6 TTL Output Ports

There are 4 single ended TTL-level output ports on PCI-8124. The following table shows the pin assignment of these TTL outputs.

CN1 Pin No.	Name	Description
23	TTL-OUT1	TTL Level Output port 1
24	TTL-OUT2	TTL Level Output port 2
48	TTL-OUT3	TTL Level Output port 3
49	TTL-OUT4	TTL Level Output port 4
25, 50	DGND	TTL Input reference GND

The maximum output or sink current of each channel is 25mA. And the maximum output voltage is D5V. User should take care of these limitations.

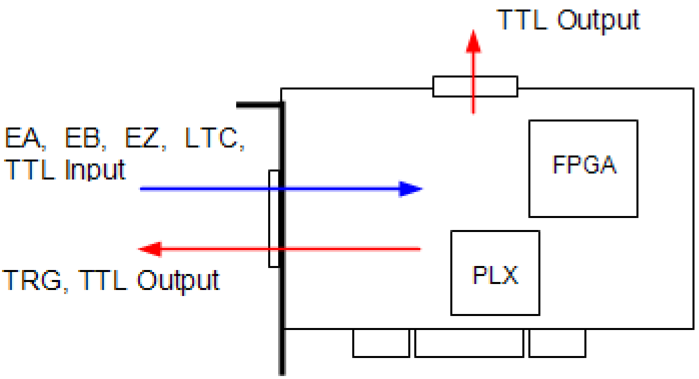
4 Operation Theory

This chapter describes the detail operation of the PCI-8124 board. Contents of the following sections are as follows:

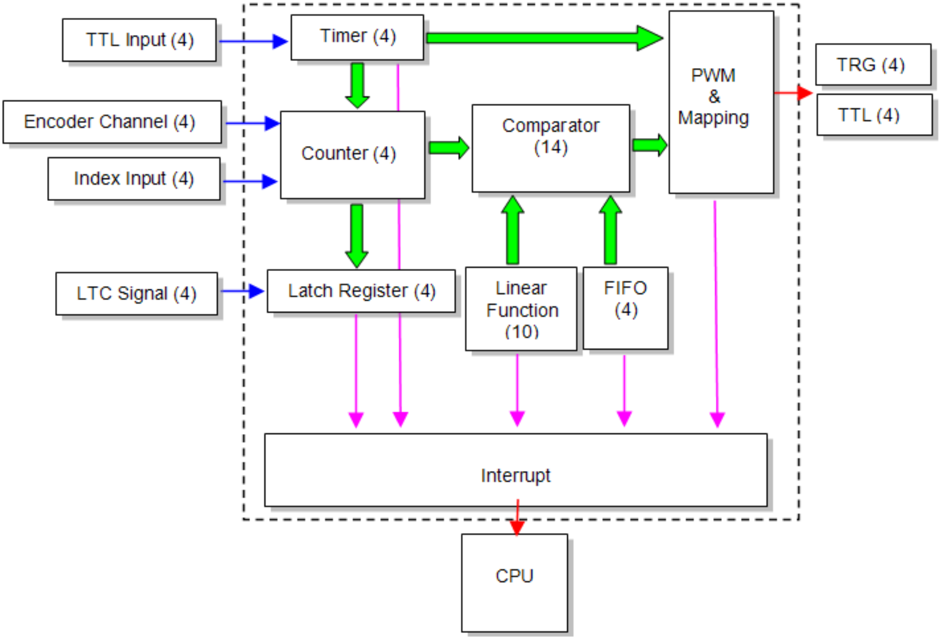
Section 4.1:	The trigger board overview
Section 4.2:	Trigger board operation
Section 4.3:	Multiple boards operation
Section 4.4:	Interrupt Control

4.1 Trigger Board Overview

The input and output relationships are shown as follows:



Inside the PCI-8124, all the position comparison and mapping tasks will be executed within the FPGA. The functional block is as follows:



The FIFO will automatically load the comparing points into 32-bit comparator. The comparator will compare the data from counter and FIFO individually. Due to the mapping mechanism, several combinations are provided for users to continuously send out the trigger pulse with isolation or non-isolation channels. Latch signal can let users retain and retrieve the position data at very short time.

Linear function will automatically load the next comparing points with a fixed incremental value into 32-bit comparator. Just like FIFO's purpose, the new comparing points will be loaded once the previous comparing point is compared and pulse is triggered.

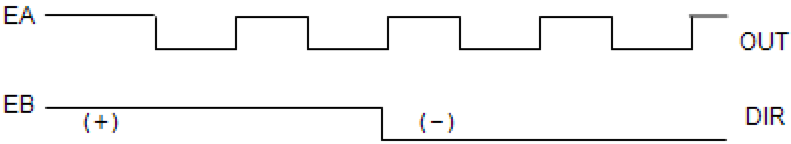
4.1.1 Encoder Channel

The PCI-8124 has four 32-bit binary up/down counters to keep the present feedback position from each encoder channel. The counter counts signal inputs from the EA and EB pins. The counter source could be assigned to internal timer.

It accepts 3 types of pulse inputs: (1). Single pulse mode (OUT/DIR) (2) Dual pulse mode (CW/CCW) (3) 90° phase shifted signals (AB phase mode). 90° phase shifted signals maybe multiplied by a factor of 1, 2 or 4. The 4xAB phase mode is the most commonly used in incremental encoder device. 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 or –8000 pulses per turn depending on its rotating direction. The three options will be explained as follows.

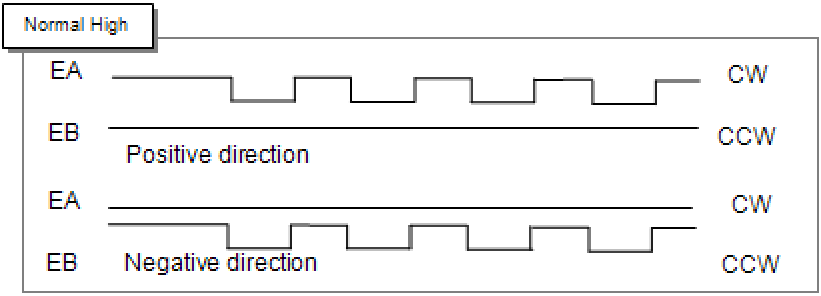
Single Pulse Mode (OUT/DIR Mode)

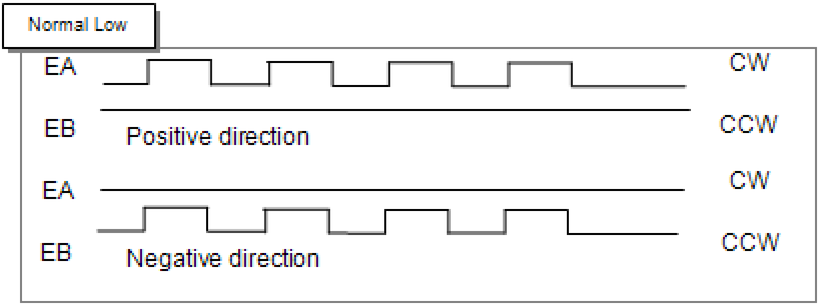
In this mode, EA is dedicated to the count the pulses and EB is to represent the direction. Simply put, we call it as PULSE/DIR mode. The DIR signal represents direction as positive (+) or negative (-). User can decide the normal high or normal low for those two channels according to users' devices. For example, we set these two input channel as normal high. The EA can count the pulse input when the signal level changes from high to low. EB will see the high level as positive direction unless the level changes to low level. The waveform is as follows.



Dual Pulse Mode (CW/CCW Mode)

In this mode, EA is dedicated to count the pulses from external source and view it as clockwise direction (CW). EB is dedicated to count the pulses from external source and view it as counterclockwise direction (CCW). Simply put, EA counts up and EB counts down. User can decide the normal high or normal low for those two channels according to users' devices. The following diagrams show the normal high and normal low cases individually.

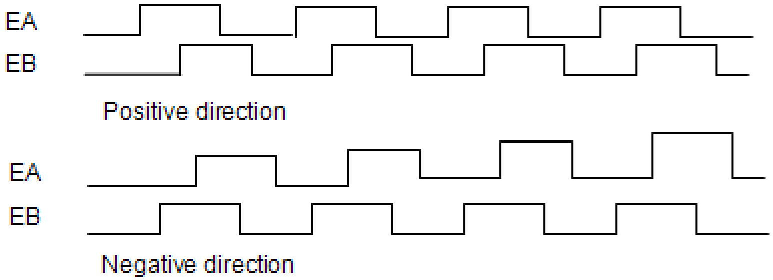




90° phase shifted signals (AB phase Mode)

In this mode, EA signal is a 90° phase leading or lagging to EB signal. “Lead” or “lag” of phase difference between two signals is caused by the turning direction of the motor. The up/down counters counts up when the phase of EA signal leads the phase of EB signal.

The following diagram shows the waveform.



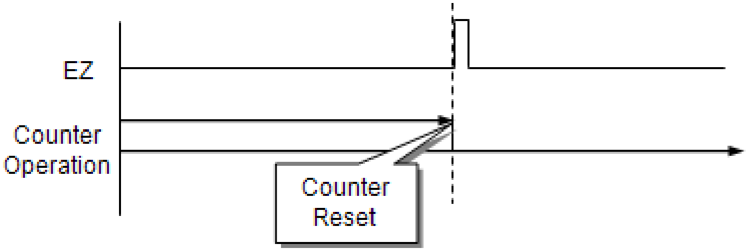
4.1.2 Index Input (EZ)

The index input (EZ) signals of the encoders are used as the “ZERO” reference. This signal is common on most rotational motors. EZ can be used to define the absolute position of the mechanism. The input logic polarity of the EZ signals is programmable with ADLINK library.

With EZ signal, users can realize the homing procedure and clear the counter as zero.

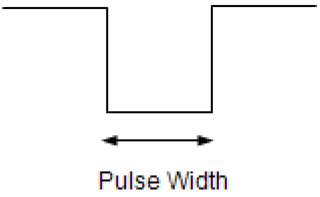
4.1.3 Homing with EZ Signal

The PCI-8124 can clear the counter value as zero according to the edge of EZ signal. Homing by edge can let users meet best homing positioning purpose. Rising or falling edge is supported. The following diagram shows the case about the homing by rising edge.



4.1.4 Trigger Pulse Width

For different applications, the trigger pulse width requirement is different. As for this reason, the trigger pulse width can be adjustable. The available values are from 0.1µs to 3.2765ms. The maximum frequency is up to 5 MHz.



4.1.5 Linear Function

Linear function is used to generate a new comparing position by a fixed incremental value linear function, Y=Y'+A. Y means a calculated comparing position, the linear data. Y' means a previous comparing position. Every time the position is compared, a new data is calculated by adding 'A', the fixed incremental value. This mechanism makes a the continuous triggering function.

4.1.6 FIFO

FIFO is first-in-first-out storage. It is used for storing some preset position data for comparing. Every time the position is compared, a new data is retrieve from FIFO into comparator. This mechanism makes a the continuous triggering function.

4.1.7 PWM & Mapping

PWM is used for adjusting pulse width of trigger. It could also be switched to a toggle mode. In this mode, the pulse level will change from low to high or high to low at every time when compared.

Mapping means four trigger signals are not one-to-one mapping to four comparators. For example, Comparator 1 could be linked to trigger channel 2. Comparator 2 could be linked to trigger channel 1 and 4. Comparator 2 and 4 could be linked to channel 3.

4.2 Trigger Board Operation

This section describes the operation theory for users to know more about PCI-8124.

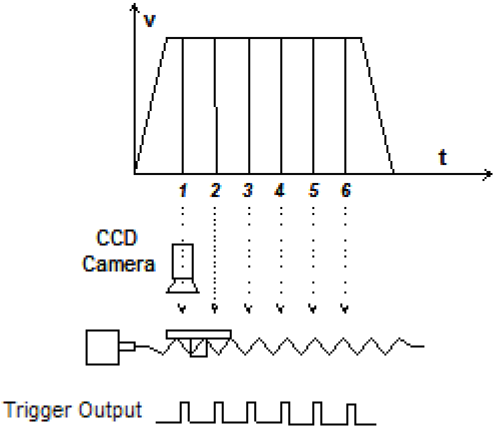
4.2.1 Position Comparison

The PCI-8124 provides position comparison functions for each channel. Once the counter reaches a preset value set by the user, the PCI-8124 will generate the trigger pulse. TRGx pins are for trigger pulse output channels. Users can also choose TTL Output pins for this purpose. TRG is an open collector pin.

The comparing method is “equal”. Consequently, when the counter value is exactly equal to the preset value by users, the trigger pulse will be generated. At the same time, the next comparing points saved in FIFO or linear function will automatically loaded into comparator. The following is an example for continuous trigger application.

Example: Using the continuous position comparison function.

In this application, the table is controlled by the motion command, and the CCD Camera is controlled by the position comparison output of the PCI-8124. An image of the moving object is easily obtained.

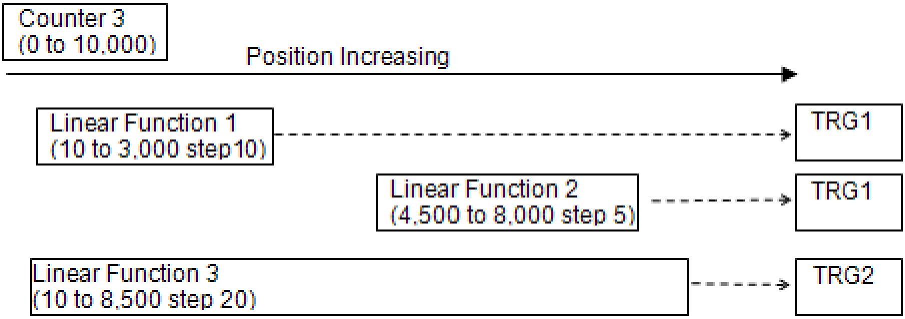


4.2.2 Position Latch

The position latch function is fulfilled by LTC signal. Once the LTC signal is active, the counter value of its latch channel will be saved to latched register at the same time. User's can read the latch register any time.

4.2.3 Linear Function

There are 10 linear functions in PCI-8124. Each function is independent. It means that linear function range overlapped is possible when operation. Each linear function has its own comparator and the comparator could be linked to any one of four counters. By this features, users can produce many kinds of trigger modes. Please see the following diagram:



Take three linear functions. Set the trigger interval and range as shown in the diagram. Set these three linear functions to counter3 and also comparator3. Set three linear functions to trigger output pin TRG1, TRG1, and TRG2 respectively. After these settings, when the counters start counting from 0 to 10,000, the trigger pins will output pulses respectively when the compare conditions are met inside linear function.

4.2.4 FIFO mode

Continuous triggering is fulfilled by linear function and FIFO. These two modes have their own comparators and can be used at the same time. The FIFO mode is usually used on random comparing data condition. Users can preset these data into FIFO and perform continuous triggering. Each FIFO has 1023 random comparing data space. If users have more random data than 1023, try to polling FIFO empty status or wait interrupt event and refill reset random data.

4.2.5 Timer function

There are 4 timers on PCI-8124. The timer is started by TTL input and the timer can be set to counters to simulate encoder inputs. It can also output to trigger pins directly. The timer is designed by a down-counter. Users must set a counter value into timer for down counting. Once the timer counter reaches zero, the timer will output a pulse to trigger pin or increase encoder counter by 1. The down counting speed is 400ns and the maximum counter value is 14-bit.

4.3 Multiple Boards Operation

The maximum installation in one PC system is up to 16 pieces. This means user can have 64 channels for triggering purpose. Since the PCI-8124 is Plug-and-Play compatible, the base address and IRQ settings for card are automatically assigned by the BIOS of the system when it is powered on.

When multiple cards are applied to a system, user can select the unique index number with S1. With this benefit, users can control each board accordingly without any confusion. For example, if three PCI-8124 cards are plugged in to PCI slots, then the corresponding channel number on each card will be:

	CH No.	**CH 1**	**CH 2**	**CH 3**	**CH 4**
Card ID					
1		0	1	2	3
2		0	1	2	3
3		0	1	2	3

Note: The index number is not accumulated if you have more than one board in PCI slot.

4.4 Interrupt Control

The PCI-8124 can generate an INT signal to the host PC. Users have to enable the interrupt service with this function call, _8124_set_int_factor. On the contrary, the parameter of intFactor of function is set to 0 to disable the interrupt service.

There are 7 kinds of factors could triggered the PCI-8124 interrupt. The factors include:

1. PWM:

   When PWM received a comparing trigger include manual trigger, it will make an interrupt to host PC.

2. FIFO level:

   When the status of FIFO is becoming FIFO level the you set by the function call “_8124_set_fifo_level”.

3. FIFO empty:

   When the FIFO is becoming empty.

4. FIFO full:

   When the FIFO is becoming full.

5. LTC:

   When the latch counter event is occurred, an interrupt will be triggered.

6. TTL-IN

   When timer is triggered by TTL-IN, an interrupt will be generated.

7. Linear function

   When linear comparing is finish, it will generate an interrupt.

   The following table shows the architecture of interrupt factor of each channel. Notice that the INT factors of last channel (ch 4) is not the same with the others. It is used to configure the interrupt of linear function.

	Bit	0	1	2	3	4	5	6	7	8	9
CH											
0		PWM0	FIFO0 Level	FIFO0 Empty	FIFO_0 Full	LTC0	TTL-IN0	-(*)	-	-	-
1		PWM1	FIFO1 Level	FIFO1 Empty	FIFO1 Full	LTC1	TTL-IN1	-	-	-	-
2		PWM2	FIFO2 Level	FIFO2 Empty	FIFO2 Full	LTC2	TTL-IN2	-	-	-	-
3		PWM3	FIFO3 Level	FIFO3 Empty	FIFO3 Full	LTC3	TTL-IN3	-	-	-	-
4		LIN0	LIN1	LIN2	LIN3	LIN4	LIN5	LIN6	LIN7	LIN8	LIN9

* (-) : reserved bit (set to 0)

Use wait function we provided to wait interrupts under Windows

To detect an interrupt signal from the PCI-8124 under Windows, we provided an easier way to deal with the interrupt process. The following steps are described that how you use those interrupt functions.

Steps:

1. Enable the interrupt factors that you want to be wait.

   `“_8124_get_int_factor”`

2. Using wait function to wait the specified interrupt.

   `“_8124_wait_single_int”`

3. In the end of interrupt process, you should disable the interrupt service.

   `“_8124_get_int_factor”`

Example:

```
_8124_set_int_factor( CardId, Channel, 0x4); //
     FIFO empty
…(FIFO operation)…

// wait FIFO empty interrupt
// FactorBitNum = 2
// 10 seconds time-out
_8124_wait_single_int( CardId, Channel, 2, 10000
     );
…
_8124_set_int_factor( CardId, Channel, 0 )//
     Disable the interrupt service.
```

5 Function Library

This chapter describes the supporting software for PCI-8124. User can use these functions to develop programs in C, C++, or Visual Basic.

5.1 List of Functions

This section details all the functions. The function prototypes and common data types are declared in pci_8124.h and type_def.h. We suggest you 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

Table 5-1: Data type definitions

The functions of the PCI-8124's software drivers use full-names to represent the functions real meaning. The naming convention rules are:

In a 'C' programming environment:

- _{8124}_{action_name (lower case)} e.g. _8124_initial ()

In order to recognize the difference between a C library and a VB library, a capital "B" is placed at he beginning of each function name e.g. B_8124_initial ().

System: Section 5.2

Function Name	Description
_8124_initialSw	PCI-8124 Cards Initialization
_8124_close	Release the resources of PCI-8124 Cards
_8124_get_version	Get the hardware and software versions

Encoder & Counter: Section 5.3

Function Name	Description
_8124_set_encoder_input_mode	Set the pulse input decoded mode
_8124_set_encoder_value	Set the encoder’s(counter’s) value
_8124_get_encoder_value	Get the encoder’s(counter’s) value
_8124_set_encoder_up_down_count	Set count up or count down for counter
_8124_set_ez_clear	Configure the EZ clear operation
_8124_set_counter_source	Set the source of counter

PWM Configuration: Section 5.4

Function Name	Description
_8124_set_pwm_source	Set the source of PWM
_8124_set_pwm_mode	Set the mode of PWM (Pulse or Toggle output)
_8124_set_pwm_pulse_width	Set the output pulse width
_8124_set_pwm_toggle_dir	Set the count direction to toggle the level output

TRG-OUT & TTL-OUT Configuration: Section 5.5

Function Name	Description
_8124_set_trigger_source	Set the source of TRG-OUT
_8124_set_trgOut_logic	Set the logic of TRG-OUT
_8124_set_ttlOut_source	Set the source of TTL-OUT
_8124_set_ttlOut	Set the TTL-OUT when it is as a digital output

Comparator: Section 5.6

Function Name	Description
_8124_set_comparator_data	Set the comparing data to the comparator
_8124_get_comparator_data	Get current comparing data from the comparator

FIFO Comparing: Section 5.7

Function Name	Description
_8124_reset_fifo	Reset FIFO data
_8124_get_fifo_sts	Get the current FIFO status
_8124_set_fifo_data	Set a comparing data to FIFO
_8124_set_fifo_array	Set comparing data array to FIFO
_8124_set_fifo_shift	Shift the FIFO data to comparator
_8124_set_fifo_level	Set the FIFO low level
_8124_get_fifo_level	Get the FIFO low level

Linear Comparing: Section 5.8

Function Name	Description
_8124_set_linear_compare	Set the linear comparing function
_8124_set_linear_source	Set the trigger output sources for linear functions
_8124_enable_linear_set	Enable/Disable the linear set

Manual Trigger: Section 5.9

Function Name	Description
_8124_manual_trigger	Set the trigger output manually

Digital input: Section 5.10

Function Name	Description
_8124_get_di_sts	Get the digital input signal's status

Latch: Section 5.11

Function Name	Description
_8124_set_latch	Configure the latch function
_8124_get_latch_event_sts	Get the latch event status
_8124_get_latch_value	Get the latched value

Timer: Section 5.12

Function Name	Description
_8124_set_Timer	Configure the Timer

Function Name	Description
_8124_start_timer	Start the timer manually

Interrupt Control: Section 5.13

Function Name	Description
_8124_set_int_factor	Set (Enable) the interrupt factors
_8124_get_int_factor	Get the setting of interrupt factors
_8124_wait_single_int	Wait the single interrupt event
_8124_reset_int	Set the interrupt event to a non-signaled state

5.2 System & Initialization

@ Name

_8124_initialSw	PCI-8124 Cards Initialization
_8124_close	Release the resources of PCI-8124 Cards
_8124_get_version	Get the hardware and software versions

@ Description

_8124_initialSw:

This function is used to initialize the PCI-8124 cards in your system. All PCI-8124 cards must be initialized by this function before calling other functions. The ID of cards can be assigned by system automatically or dip switch (SW1) manually.

The source of TRG-OUT and TTL-OUT output signals can be decided by SW2 or PWM output. See also section 5.5 “_8124_set_trigger_source”.

If DefaultSw = 1, the output signal of TRG-OUT and TTL-OUT will output as SW2. On the contrary, if DefaultSw = 0, the TRG-OUT will output as PWM default setting and TTL-OUT will output D5V (as a digital output).

_8124_close:

This function is to close the PCI-8124 cards and release its resources, which must be called at the end of your applications.

_8124_get_version:

Users can get the hardware and software version number by this function.

@ Syntax

C/C++ (Windows 2000/XP)

```
I16 _8124_initialSw( I16 *CardIdInBit, I16
     ManualId, I16 DefaultSw );
```

```
I16 _8124_close();
I16 _8124_get_version(I16 CardId, I32
     *Firmware_ver, I32 *Driver_ver, I32
     *DLL_ver);
```

Visual Basic (Windows 2000/XP)

```
B_8124_initialSw (CardIdInBit As Integer, ByVal
     ManualId As Integer, ByVal DefaultSw As
     Integer) As Integer
B_8124_close () As Integer
B_8124_get_version (ByVal CardId As Integer,
     Firmware_ver As Long, Driver_ver As Long,
     DLL_ver As Long) As Integer
```

@ Arguments

`I16 *CardIdInBit`: The information of PCI-8124 cards' card ID in bit format.

- Ex. CardIdInBit = 0x0009; It means there are two PCI-8124 cards in your system
- Card_0's ID = 0, Card_1's ID = 3

`I16 ManualId`: Cards' ID decided by SW1

- 0: Decided by system,
- 1:Decided by dip switch(SW1)

`I16 DefaultSw`: Default output setting.

- 0: Default output as PWM and Digital out
- 1: Default output as SW2

`I16 CardId`: The PCI-8124 card's ID number. (0 ~ 15)

`I32 *Firmware_ver`: The firmware version.

`I32 *Driver_ver`: The device driver version.

`I32 *DLL_ver`: The DLL' version.

5.3 Encoder & Counter

@ Name

_8124_set_encoder_input_mode	Set the pulse input decoded mode
_8124_set_encoder_value	Set the encoder's(counter's) value
_8124_get_encoder_value	Get the encoder's(counter's) value
_8124_set_encoder_up_down_count	Set count up or count down for counter
_8124_set_ez_clear	Configure the EZ clear operation
_8124_set_counter_source	Set the source of counter

@ Description

_8124_set_encoder_input_mode:

There are 5 types of pulse input mode(EA/EB) including OUT/DIR, CW/CCW, 1x , 2x, and 4x AB phase. Choose a correct encoder input mode before using counter.

_8124_set_encoder_value:

This function is used to set the encoder(counter) value directly.

_8124_get_encoder_value:

This function is used to read the current encoder value.

_8124_set_encoder_up_down_count:

This function is used to set the specified encoder to count up or count down.

_8124_set_ez_clear:

This function is used to enable the function of counter clear by sensing EZ input signal. If enabled, you should choose falling edge or rising edge to clear counter. On the contrary, if disabled, the setting of "ClrLogic" will be ignored.

_8124_set_counter_source:

This function is used to set the counter source. The source could be one of the EA/EB input channels or timers. The default counter source is EA/EB. About timer, please refer the section 5.12 Timer.

@ Syntax

C/C++ (Windows 2000/XP)

```
I16 _8124_set_encoder_input_mode( I16 CardId, I16
     Channel, I16 IptMode );
I16 _8124_set_encoder_value( I16 CardId, I16
     Channel, I32 EncValue );
I16 _8124_get_encoder_value( I16 CardId, I16
     Channel, I32 *EncValue );
I16 _8124_set_encoder_up_down_count(I16 CardId,
     I16 Channel, I16 Inverse );
I16 _8124_set_ez_clear( I16 CardId, I16 Channel,
     I16 Enable, I16 ClrLogic );
I16 _8124_set_counter_source( I16 CardId, I16
     Channel, I16 CtnSrc );
```

Visual Basic (Windows 2000/XP)

```
B_8124_set_pls_iptmode (ByVal CardId As Integer,
     ByVal Channel As Integer, ByVal IptMode As
     Integer) As Integer
B_8124_set_encoder_input_mode (ByVal CardId As
     Integer, ByVal Channel As Integer, ByVal
     IptMode As Integer) As Integer
B_8124_set_encoder_value (ByVal CardId As
     Integer, ByVal Channel As Integer, ByVal
     EncValue As Long) As Integer
B_8124_get_encoder_value (ByVal CardId As
     Integer, ByVal Channel As Integer, EncValue
     As Long) As Integer
B_8124_set_encoder_up_down_count (ByVal CardId As
     Integer, ByVal Channel As Integer, ByVal
     Inverse As Integer) As Integer
B_8124_set_ez_clear (ByVal CardId As Integer,
     ByVal Channel As Integer, ByVal Enable As
     Integer, ByVal ClrLogic As Integer) As
     Integer
```

```
B_8124_set_counter_source (ByVal CardId As
     Integer, ByVal Channel As Integer, ByVal
     CtnSrc As Integer) As Integer
```

@ Argument

`I16 CardId`: The PCI-8124 card's ID number.

`I16 Channel`: The specified channel number. (0 – 3)

`I16 IptMode`: The pulse input mode.

- ▶ 0: OUT/DIR (default)
- ▶ 1: CW/CCW
- ▶ 2: 1x AB-Phase
- ▶ 3: 2x AB-Phase
- ▶ 4: 4x AB-Phase

`I32 EncValue`: The encoder (counter) value.

`I16 Inverse`: Encoder count direction inverse or not

- ▶ 0: Not inverse (default)
- ▶ 1: Inverse

`I16 Enable`: Enable EZ-clear function

- ▶ 0: Disable (default)
- ▶ 1: Enable

`I16 ClrLogic`: Clear logic setting

- ▶ 0: Photo couple: OFF ⇒ ON (Falling edge) clear (default)
- ▶ 1: Photo couple: ON ⇒ OFF (Rising edge) clear

`I16 CtnSrc`: The counter's source (0 – 7)

- ▶ 0 – 3: EA/EB 0 – 3
- ▶ 4 – 7: Timer 0 – 3

5.4 PWM Configuration

@ Name

_8124_set_pwm_source	Set the source of PWM
_8124_set_pwm_mode	Set the mode of PWM (Pulse or Toggle output)
_8124_set_pwm_pulse_width	Set the output pulse width
_8124_set_pwm_toggle_dir	Set the count direction to toggle the level output

@ Description

_8124_set_pwm_source:

This function is used to set the PWM source (trigger source). A PWM has 3 kinds of source, LIN, CMP and timer respectively. Those three kind of sources are independently. Users can enable them at the same time. The default setting is CMP.

_8124_set_pwm_mode:

There are two types of TRG output. One is pulse type output, the other is level-switch output. Users can select output type by this function.

_8124_set_pwm_pulse_width:

If you selected the pulse type output as the PWM output. You can adjust the pulse width by this function.

_8124_set_pwm_toggle_dir:

If you selected the level-switch output as the PWM output, you can set the toggled condition by this function. There are 3 conditions (count directions) that the level output could be toggled.

@ Syntax

C/C++ (Windows 2000/XP)

```
I16 _8124_set_pwm_source( I16 CardId, I16
        Channel, I16 CmpEn, I16 LinearEn, I16
        TimerEn );
```

```
I16 _8124_set_pwm_mode(  I16 CardId, I16 Channel,
     I16 PulseOrToggle );
I16 _8124_set_pwm_pulse_width( I16 CardId, I16
     Channel, I16 WidthPara );
I16 _8124_set_pwm_toggle_dir(  I16 CardId, I16
     Channel, I16 ToggleDir );
```

Visual Basic (Windows 2000/XP)

```
B_8124_set_pwm_source (ByVal CardId As Integer,
     ByVal Channel As Integer, ByVal CmpEn As
     Integer, ByVal LinearEn As Integer, ByVal
     TimerEn As Integer) As Integer
B_8124_set_pwm_mode (ByVal CardId As Integer,
     ByVal Channel As Integer, ByVal
     PulseOrToggle As Integer) As Integer
B_8124_set_pwm_pulse_width (ByVal CardId As
     Integer, ByVal Channel As Integer, ByVal
     WidthPara As Integer) As Integer
B_8124_set_pwm_toggle_dir (ByVal CardId As
     Integer, ByVal Channel As Integer, ByVal
     ToggleDir As Integer) As Integer
```

@ Argument

I16 CardId: The PCI-8124 card's ID number.

I16 Channel: The channel's number of PWM

I16 CmpEn: Enable CMP(FIFO comparing) source (0: Disable, 1: Enable)

I16 LinearEn: Enable Linear comparing (0: Disable, 1: Enable)

I16 TimerEn: Enable timer source. (0: Disable, 1:Enable)

I16 PulseOrToggle: Select the pulse output or level switch output

- 0: Pulse output
- 1: Level switch output (toggle output)

I16 WidthPara: The parameter of pulse width. (0 – 8191)

- Pulse width(ns) = (WidthPara × 16 + 4) × 25

`I16 ToggleDir`: Toggled condition select

- 0: Directionless
- 1: Count up
- 2: Count down

5.5 TRG-OUT & TTL-OUT Configuration

@ Name

_8124_set_trigger_source	Set the source of TRG-OUT
_8124_set_trgOut_logic	Set the logic of TRG-OUT
_8124_set_ttlOut_source	Set the source of TTL-OUT
_8124_set_ttlOut	Set the TTL-OUT when it is as a digital output

@ Description

_8124_set_trigger_source:

There are two kinds of TGR sources. One is from SW2 setting, the other is from PWM or DI signals. If you select SW2 setting, the function will ignore the setting of parameters "PwmInBit", "TtlInInBit" and "EzInBit". On the contrary, If you select PWM/ TTL-IN/EZ as source, you should set the other parameters in next. They can be "OR" output.

_8124_set_trgOut_logic:

This function is used to set the logic of TRG –OUT output signal.

_8124_set_ttlOut_source:

This function is used to set the source of TTL-OUT output. There are two sources of TTL-OUT output. One is DO output. The other is output as the relative channel of TRG-OUT output.

Caution: Because of the logic of TRG-OUT outputs can be inversed, the logic of TRG-OUT output and TTL-OUT output may opposite.

_8124_set_ttlOut:

When TTL-OUT is set as digital output (DO), you can control it by this function.

@ Syntax

C/C++ (Windows 2000/XP)

```
I16 _8124_set_trigger_source( I16 CardId, I16
     Channel, I16 GroupSel, I16 PwmInBit, I16
     TtlInInBit, I16 EzInBit );
I16 _8124_set_ttlOut_source( I16 CardId, I16
     Channel, I16 TtlOutSrc );
I16 _8124_set_ttlOut( I16 CardId, I16 Channel,
     I16 TtlOutValue );
```

Visual Basic (Windows 2000/XP)

```
B_8124_set_trigger_source (ByVal CardId As
     Integer, ByVal Channel As Integer, ByVal
     GroupSel As Integer, ByVal PwmInBit As
     Integer, ByVal TtlInInBit As Integer, ByVal
     EzInBit As Integer) As Integer
B_8124_set_trgOut_logic (ByVal CardId As Integer,
     ByVal Channel As Integer, ByVal Logic As
     Integer) As Integer
B_8124_set_ttlOut_source (ByVal CardId As
     Integer, ByVal Channel As Integer, ByVal
     TtlOutSrc As Integer) As Integer
B_8124_set_ttlOut (ByVal CardId As Integer, ByVal
     Channel As Integer, ByVal TtlOutValue As
     Integer) As Integer
```

@ Argument

`I16 CardId`: The PCI-8124 card's ID number.

`I16 Channel`: The channel's number of TRG-OUT / TTL-OUT.

`I16 GroupSel`: The TRG-OUT source.

- 0: Output according to SW2 setting
- 1: PWM / TTL-IN / EZ signal

`I16 PwmInBit`: The source of PWM channel in bit format.

- Ex: PwmInBit = 0x2, means PWM_1

`I16 TtlInInBit`: The source of TTL-IN channel in bit format.

- Ex: TtlInInBit = 0x6, means TTL-IN1 and TTL-IN_2

`I16 EzInBit`: The source of EZ channel in bit format.

- Ex: EzInBit = 0x1, means EZ_0

`I16 TtlOutSrc`: The TTL-OUT output source

- 0: use as Digital output
- 1: Follow TRG-OUT

`I16 TtlOutValue`: TTL-OUT output control

- 0: DGND.
- 1: D5V.

5.6 Comparator

@ Name

_8124_set_comparator_data	Set the comparing data to the comparator
_8124_get_comparator_data	Get current comparing data from the comparator

@ Description

_8124_set_comparator_data:

This function is used to override the current comparing data to the comparator (CMP).

_8124_get_comparator_data:

This function is used to get the current comparing data from comparator (CMP).

@ Syntax

C/C++ (Windows 2000/XP)

```
I16 _8124_set_comparator_data( I16 CardId, I16
     Channel, I32 CmpData );
I16 _8124_get_comparator_data( I16 CardId, I16
     Channel, I32 *CmpData );
Visual Basic (Windows 2000/XP)
B_8124_set_comparator_data (ByVal CardId As
     Integer, ByVal Channel As Integer, ByVal
     CmpData As Long) As Integer
B_8124_get_comparator_data (ByVal CardId As
     Integer, ByVal Channel As Integer, CmpData
     As Long) As Integer
```

@ Argument

`I16 CardId`: The PCI-8124 card’s ID number.

`I16 Channel`: The channel’s number of comparator (CMP).

`I32 CmpData`: The current comparing data in comparator.

5.7 FIFO Comparing

@ Name

_8124_reset_fifo	Reset FIFO data
_8124_get_fifo_sts	Get the current FIFO status
_8124_set_fifo_data	Set a comparing data to FIFO
_8124_set_fifo_array	Set comparing data array to FIFO
_8124_set_fifo_shift	Shift the FIFO data to comparator
_8124_set_fifo_level	Set the FIFO low level
_8124_get_fifo_level	Get the FIFO low level

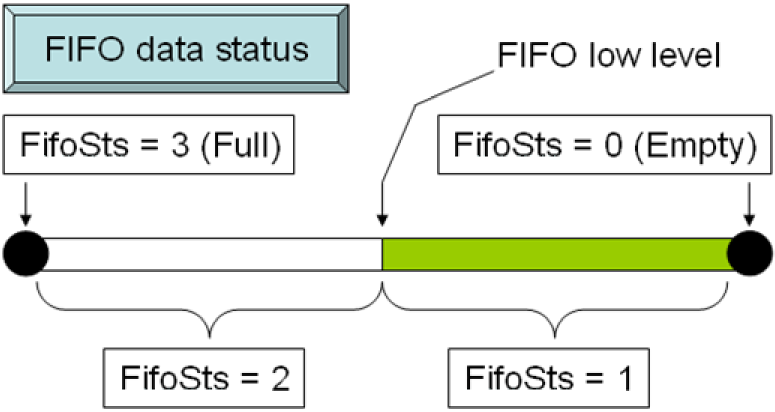
@ Description

_8124_reset_fifo:

This function is used to reset (clear) the FIFO. The FIFO status is return “Empty” status, when you issue this function.

_8124_get_fifo_sts:

This function is used to get the current FIFO status. There are four status this function would return. The value meaning is as below figure.



_8124_set_fifo_data:

This function is used to set the specified comparing data to the FIFO. The capacity of FIFO is 1023. When the status of FIFO is full, the data cannot be set into FIFO. This function won't check the FIFO status.

_8124_set_fifo_array:

This function is used to set comparing data array to the FIFO.

_8124_set_fifo_shift:

This function is used to shift the FIFO data to comparator (CMP) manually. Generally, when using the FIFO comparing, the first compared data must shift to the CMP manually via this function.

_8124_set_fifo_level:

This function is used to set the FIFO low level. There are four conditions in FIFO low level selection. The default setting is 512 elements remaining in FIFO.

_8124_get_fifo_level:

This function is used to get the setting of FIFO low level.

@ Syntax

C/C++ (Windows 2000/XP)

```
I16 _8124_reset_fifo( I16 CardId, I16 Channel );
I16 _8124_get_fifo_sts( I16 CardId, I16 Channel,
     I16 *FifoSts );
I16 _8124_set_fifo_data( I16 CardId, I16 Channel,
     I32 FifoData );
I16 _8124_set_fifo_array( I16 CardId, I16
     Channel, I32 *DataArr, I16 ArraySize );
I16 _8124_set_fifo_shift( I16 CardId, I16 Channel
     );
I16 _8124_set_fifo_level( I16 CardId, I16
     Channel, I16 Level );
I16 _8124_get_fifo_level( I16 CardId, I16
     Channel, I16 *Level );
```

Visual Basic (Windows 2000/XP)

```
B_8124_reset_fifo (ByVal CardId As Integer, ByVal
     Channel As Integer) As Integer
B_8124_get_fifo_sts (ByVal CardId As Integer,
     ByVal Channel As Integer, FifoSts As
     Integer) As Integer
B_8124_set_fifo_data (ByVal CardId As Integer,
     ByVal Channel As Integer, ByVal FifoData As
     Long) As Integer
B_8124_set_fifo_array (ByVal CardId As Integer,
     ByVal Channel As Integer, DataArr As Long,
     ByVal ArraySize As Integer) As Integer
B_8124_set_fifo_shift (ByVal CardId As Integer,
     ByVal Channel As Integer) As Integer
B_8124_set_fifo_level (ByVal CardId As Integer,
     ByVal Channel As Integer, ByVal Level As
     Integer) As Integer
B_8124_get_fifo_level (ByVal CardId As Integer,
     ByVal Channel As Integer, Level As Integer)
     As Integer
```

@ Argument

`I16 CardId`: The PCI-8124 card's ID number.

`I16 Channel`: The channel's number of FIFO (0 – 3)

`I16 *FifoSts`: The FIFO status.

- 0: FIFO datas = Empty
- 1: Low level ≥ FIFO data > Empty
- 2: Full > FIFO data > Low level
- 3: FIFO = Full

`I32 FifoData`: The data to FIFO.

`I32 *DataArr`: The data array to FIFO

`I16 ArraySize`: The size of data array to FIFO. (1 – 1023)

`I16 Level`: The FIFO low level setting.(remaining data numbers in FIFO)

- 0: Low level = 0 (Empty)
- 1: Low level = 256 (1/4 FIFO size)
- 2: Low level = 512 (1/2 FIFO size) (Default)
- 3: Low level = 768 (3/4 FIFO size)

5.8 Linear Comparing

@ Name

_8124_set_linear_compare	Set the linear comparing function
_8124_set_linear_source	Set the trigger output sources for linear functions
_8124_enable_linear_set	Enable/Disable the linear set

@ Description

_8124_set_linear_compare:

This function is used to configure the linear comparing trigger function. Before using linear comparing function, you should enable the source of PWM from linear function by "_8124_set_pwm_source". PCI-8124 has 10 linear sets can be used for linear and they can be compared with any counter arbitrarily. Every set has 3 parameters including starting point (StartPoint), repeat times (RepeatTimes) and trigger interval (Interval). The total trigger pulse would be (RepeatTimes + 1) times. For example, if you set the RepeatTimes as 10, the total output pulse will be 11. The follow formula is show that how do you calculate the next triggering position.

- Next triggering position = StartPoint + Interval x N
- N: triggered times, (N ≤ RepeatTimes)

The Interval can be set as a negative value. The comparing direction is depending on the signed of Interval. Moreover, if RepeatTimes is set to 0, the linear comparing will not stop until stopping counting or reset the linear function.

_8124_set_linear_source:

There are 10 linear sets in PCI-8124. Use this function to assign which sets will be used in the specified trigger channel. One trigger channel can have more then one linear set.

_8124_enable_linear_set:

The PCI-8124 has 10 linear sets. Before using these linear sets, enable the linear set by this function. When linear sets are no longer needed, it is recommended that the the linear set be disabled, or it could be unexpectedly triggered.

@ Syntax

C/C++ (Windows 2000/XP)

```
I16 _8124_set_linear_compare( I16 CardId, I16
     SetNum, I16 Channel, I32 StartPoint, F64
     RepeatTimes, I16 Interval );
I16 _8124_set_linear_source( I16 CardId, I16
     Channel, I16 SetNumInBit );
I16 _8124_enable_linear_set( I16 CardId, I16
     SetNum, I16 Enable );
```

Visual Basic (Windows 2000/XP)

```
B_8124_set_linear_compare (ByVal CardId As
     Integer, ByVal SetNum As Integer, ByVal
     Channel As Integer, ByVal StartPoint As
     Long, ByVal RepeatTimes As Double, ByVal
     Interval As Integer) As Integer
B_8124_set_linear_source( ByVal CardId As
     Integer, ByVal Channel As Integer, ByVal
     SetNumInBit As Integer ) As Integer
B_8124_enable_linear_set( ByVal CardId As
     Integer, ByVal SetNum As Integer, ByVal
     Enable As Integer )As Integer
```

@ Argument

`I16 CardId`: The PCI-8124 card’s ID number.

`I16 SetNum`: Linear Setting set’s number (0 - 9)

`I16 Channel`: The channel of linear output (0 - 3)

`I16 SetNumInBit`: Linear set number in bit format

`I32 StartPoint`: The first trigger point.

`F64 RepeatTimes`: The trigger repeat times. (32 bits integer)

`I16 Interval`: The linear interval (1 to 8191, -1 to -8191)

`I16 Enable`: Enable/Disable linear set

- 0: disable, 1: enable

5.9 Manual Trigger

@ Name

_8124_manual_trigger	Set the trigger output manually

@ Description

_8124_manual_trigger:

This function is used to generate one pulse manually.

@ Syntax

C/C++ (Windows 2000/XP)

```
I16 _8124_manual_trigger( I16 CardId, I16 Channel );
```

Visual Basic (Windows 2000/XP)

```
B_8124_manual_trigger (ByVal CardId As Integer,
     ByVal Channel As Integer) As Integer
```

@ Argument

`I16 CardId`: The PCI-8124 card's ID number.

`I16 Channel`: The channel's number of PWM(0 – 3)

5.10 Digital Input

@ Name

_8124_get_di_sts	Get the digital input signal's status

@ Description

_8124_get_di_sts:

This function is used to get the status of digital inputs. The digital inputs include TTL-IN, LTC and EZ. The parameters are showed as bit format. The former four bits of parameters represent 4 channels of each digital input.

@ Syntax

C/C++ (Windows 2000/XP)

```
I16 _8124_get_di_sts( I16 CardId, I16
     *TtlStsInBit, I16 *LtcStsInBit, I16
     *EzStsInBit );
```

Visual Basic (Windows 2000/XP)

```
B_8124_get_di_sts (ByVal CardId As Integer,
     TtlStsInBit As Integer, LtcStsInBit As
     Integer, EzStsInBit As Integer) As Integer
```

@ Argument

I16 CardId: The PCI-8124 card's ID number.

I16 *TtlStsInBit: The TTL-IN status in bit format.

I16 *LtcStsInBit: The LTC status in bit format.

I16 *EzStsInBit: The EZ status in bit format.

5.11 Latch

@ Name

_8124_set_latch	Configure the latch function
_8124_get_latch_event_sts	Get the latch event status
_8124_get_latch_value	Get the latched value

@ Description

_8124_set_latch:

This function is used to enable or disable the high-speed position capture (latch) function and set the latch condition. Whenever the users enable it and configure the latch condition (Rising or falling active) correctly, the current position will be latched when it receive the latch signal. Users can check the latch events by polling "_8124_get_latch_event_sts" or by using the interrupt function. See also section 5.13 Interrupt function.

_8124_get_latch_event_sts:

This function is used to check the latch event status. If the latch function is enabled and the latch condition is met, the status will become 1 from 0. The event status is read-clear.

_8124_get_latch_value:

When the latch event occurred, you can use this function get the latched counter value.

@ Syntax

C/C++ (Windows 2000/XP)

```
I16 _8124_set_latch( I16 CardId, I16 Channel, I16
     Enable, I16 EdgeSel );
I16 _8124_get_latch_event_sts( I16 CardId, I16
     *LatchEventInBit );
I16 _8124_get_latch_value(I16 CardId, I16
     Channel, I32 *LatchValue );
```

Visual Basic (Windows 2000/XP)

```
B_8124_set_latch (ByVal CardId As Integer, ByVal
     Channel As Integer, ByVal Enable As Integer,
     ByVal EdgeSel As Integer) As Integer
B_8124_get_latch_event_sts (ByVal CardId As
     Integer, LatchEventInBit As Integer) As
     Integer
B_8124_get_latch_value (ByVal CardId As Integer,
     ByVal Channel As Integer, LatchValue As
     Long) As Integer
```

@ Argument

`I16 CardId`: The PCI-8124 card's ID number.

`I16 Channel`: The channel's number of Latch (0 – 3)

`I16 Enable`: Enable the Latch function (0: Disable, 1: Enable)

`I16 EdgeSel`: Latching condition

- ▷ 0: Photo couple: OFF ⇒ ON (Falling edge) clear (default)
- ▷ 1: Photo couple: ON ⇒ OFF (Rising edge) clear

`I16 *LatchEventInBit`: Latch event status in bit format (Read clear)

- ▶ Bit0: Latch0 event status (1: Latched occurred , 0: No Latched)
- ▶ Bit1: Latch1 event status (1: Latched occurred , 0: No Latched)
- ▶ Bit2: Latch2 event status (1: Latched occurred , 0: No Latched)
- ▶ Bit3: Latch3 event status (1: Latched occurred , 0: No Latched)

`I32 *LatchValue`: The Latched counter value.

5.12 Timer

@ Name

_8124_set_Timer	Configure the Timer
_8124_start_timer	Start the timer manually

@ Description

_8124_set_Timer:

This function is used to configure the timer. The timer can be start by external signal (TTL-IN signal) or by software function. If the timer was triggered, users can stop the timer by issue the function ”_8124_start_timer”.

_8124_start_timer:

This function is used to start/stop timer manually.

@ Syntax

C/C++ (Windows 2000/XP)

```
I16 _8124_set_Timer( I16 CardId, I16 Channel, I16
     WaitTtlIn , I16 TtlTrigLogic, I16 Interval );
I16 _8124_start_timer( I16 CardId, I16 Channel,
     I16 Start );
```

Visual Basic (Windows 2000/XP)

```
B_8124_set_Timer (ByVal CardId As Integer, ByVal
     Channel As Integer, ByVal WaitTtlIn  As
     Integer, ByVal TtlTrigLogic As Integer,
     ByVal Interval As Integer) As Integer
B_8124_start_timer (ByVal CardId As Integer,
     ByVal Channel As Integer, ByVal Start As
     Integer) As Integer
```

@ Argument

`I16 CardId`: The PCI-8124 card’s ID number.

`I16 Channel`: The channel’s number of timer.

`I16 WaitTtlIn`: Enable TTL-IN start timer (0: Disable, 1: Enable)

`I16 TtlTrigLogic`: TTL-IN start timer condition

- 0: Falling edge of TTL-IN signal
- 1: Rising edge of TTL-IN signal

`I16 Interval`: Timer interval (0 – 16383)

- Timer tick interval = (Interval x 16 + 4) x 25 (ns)

`I16 Start`: Start timer

- 0: Stop timer
- 1: Start timer

5.13 Interrupt

@ Name

_8124_set_int_factor	Set (Enable) the interrupt factors
_8124_get_int_factor	Get the setting of interrupt factors
_8124_wait_single_int	Wait the single interrupt event
_8124_reset_int	Set the interrupt event to a non-signaled state

@ Description

_8124_set_int_factor:

This function is used to enable/disable the interrupt source. Users can use the function, "_8124_wait_single_int" to wait the specified interrupt. The following table shows the architecture of interrupt factor of each channel.

	Bit	0	1	2	3	4	5	6	7	8	9
CH											
0		PWM0	FIFO0 Level	FIFO0 Empty	FIFO_0 Full	LTC0	TTL-IN0	-(*)	-	-	-
1		PWM1	FIFO1 Level	FIFO1 Empty	FIFO1 Full	LTC1	TTL-IN1	-	-	-	-
2		PWM2	FIFO2 Level	FIFO2 Empty	FIFO2 Full	LTC2	TTL-IN2	-	-	-	-
3		PWM3	FIFO3 Level	FIFO3 Empty	FIFO3 Full	LTC3	TTL-IN3	-	-	-	-
4		LIN0	LIN1	LIN2	LIN3	LIN4	LIN5	LIN6	LIN7	LIN8	LIN9

(*): reserved bit (set to 0)

_8124_get_int_factor:

This function is used to get current configuration of interrupt factors in your card.

_8124_wait_single_int:

When the user enabled the interrupt function for specified factors by "_8124_set_int_factor", it could use this function to wait a specific interrupt. When this function was running, the process would never stop until the event was be triggered or the function was time out. This function returns when one of the following occurs:

1. The specified interrupt factor is in the signaled state.
2. The time-out interval elapses.

This function checks the current state of the specified interrupt factor. If the state is non-signaled, the calling thread enters the wait state. It uses no processor time while waiting for the INT state to become signaled or the time-out interval to elapse.

8124_reset_int:

This function sets the specified interrupt event to a non-signaled state. This function is usually put into program before the "_8124_wait_single_int" function.

@ Syntax

C/C++ (Windows 2000/XP)

```
I16 _8124_set_int_factor( I16 CardId, I16
     Channel, I16 IntFactorsInBit );
I16 _8124_get_int_factor( I16 CardId, I16
     Channel, I16 *IntFactorsInBit );
I16 _8124_wait_single_int( I16 CardId, I16
     Channel, I16 FactorBitNum, I32 TimeOutMs );
I16 _8124_reset_int( I16 CardId, I16 Channel, I16
     FactorBitNum );
```

Visual Basic (Windows 2000/XP)

```
B_8124_set_int_factor (ByVal CardId As Integer,
     ByVal Channel As Integer, ByVal
     IntFactorsInBit As Integer) As Integer
B_8124_get_int_factor (ByVal CardId As Integer,
     ByVal Channel As Integer, IntFactorsInBit As
     Integer) As Integer
B_8124_wait_single_int (ByVal CardId As Integer,
     ByVal Channel As Integer, ByVal FactorBitNum
     As Integer, ByVal TimeOutMs As Long) As
     Integer
B_8124_reset_int( ByVal CardId As Integer, ByVal
     Channel As Integer, ByVal FactorBitNum As
     Integer) As Integer
```

@ Argument

`I16 IntFactorsInBit`: Interrupt event factor setting in bit format.

`I16 FactorBitNum`: Specified the bit number of the IntFactorsInBit.

`I32 TimeOutM`s: Specify the time-out interval, in milliseconds.

If TimeOutMs is zero, the function tests the states of the specified event and returns immediately. If TimeOutMs is -1, the function's time-out interval never elapses (infinite)

5.14 Return Erros

The following table provides a list of possible return value in our library. If the return value is not zero, it means there are some errors occurred. This could help you to check the program easier and make it correctly.

Error Codes:

Code	Define	Description
0	ERR_NoError	No Error, function success
-1	ERR_OSVersion	Operation System type mismatched
-2	ERR_OpenDriverFailed	Open device driver failed - Create driver interface failed
-3	ERR_InsufficientMemory	System memory insufficiently
-4	ERR_DeviceNotInitial	Cards not be initialized
-5	ERR_NoDeviceFound	Cards not found(No card in your system)
-6	ERR_CardIdDuplicate	Cards' ID Number duplicate
-7	ERR_DeviceAlreadyInitialed	Cards have been initialed
-8	ERR_InterruptNotEnable	Cards' interrupt events not enable
-9	ERR_TimeOut	Function time-out
-10	ERR_ParametersInvalid	Function input parameters are invalid
-11	ERR_SetEEPROM	Set data to EEPROM failed
-12	ERR_GetEEPROM	Get data from EEPROM failed
-1000	ERR_Win32Error	WIN32 API function error, please contact with ADLINK's FAE staff.

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. When sending in damaged products for repair, please attach an RMA application form which can be downloaded from: http://rma.adlinktech.com/policy/.

2. All ADLINK products come with a limited two-year warranty, one year for products bought in China:

   - The warranty period starts on the day the product is shipped from ADLINK's factory.
   - Peripherals and third-party products not manufactured by ADLINK will be covered by the original manufacturers' warranty.
   - For products containing storage devices (hard drives, flash cards, etc.), please back up your data before sending them for repair. ADLINK is not responsible for any loss of data.
   - Please ensure the use of properly licensed software with our systems. ADLINK does not condone the use of pirated software and will not service systems using such software. ADLINK will not be held legally responsible for products shipped with unlicensed software installed by the user.
   - For general repairs, please do not include peripheral accessories. If peripherals need to be included, be certain to specify which items you sent on the RMA Request & Confirmation Form. ADLINK is not responsible for items not listed on the RMA Request & Confirmation Form.

3. Our repair service is not covered by ADLINK's guarantee in the following situations:

    - Damage caused by not following instructions in the User's Manual.
    - Damage caused by carelessness on the user's part during product transportation.
    - Damage caused by fire, earthquakes, floods, lightening, pollution, other acts of God, and/or incorrect usage of voltage transformers.
    - Damage caused by unsuitable storage environments (i.e. high temperatures, high humidity, or volatile chemicals).
    - Damage caused by leakage of battery fluid during or after change of batteries by customer/user.
    - Damage from improper repair by unauthorized ADLINK technicians.
    - Products with altered and/or damaged serial numbers are not entitled to our service.
    - This warranty is not transferable or extendible.
    - Other categories not protected under our warranty.

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

5. To ensure the speed and quality of product repair, please download an RMA application form from our company website: http://rma.adlinktech.com/policy. Damaged products with attached RMA forms receive priority.

If you have any further questions, please email our FAE staff: service@adlinktech.com.