LPCIe-8124-C

Advanced 4CH Encoder Card
with High-Speed Triggering

Function Library Reference

Manual Rev. 2.00

Revision Date: Jan. 8, 2016

Part No: 50-11261-1000

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

Advance Technologies; Automate the World.

Revision History

Revision	Release Date	Description of Change(s)
2.00	Jan.8, 2016	Initial Release

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

Preface

Copyright 2016 ADLINK Technology, Inc.

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.

Disclaimer

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.

Environmental Responsibility

ADLINK is committed to fulfill its social responsibility to global environmental preservation through compliance with the European Union's Restriction of Hazardous Substances (RoHS) directive and Waste Electrical and Electronic Equipment (WEEE) directive. Environmental protection is a top priority for ADLINK. We have enforced measures to ensure that our products, manufacturing processes, components, and raw materials have as little impact on the environment as possible. When products are at their end of life, our customers are encouraged to dispose of them in accordance with the product disposal and/or recovery programs prescribed by their nation or company.

Trademarks

Product names mentioned herein are used for identification purposes only and may be trademarks and/or registered trademarks of their respective companies.

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

Conventions

Take note of the following conventions used throughout this manual to make sure that users perform certain tasks and instructions properly.



Additional information, aids, and tips that help users perform tasks.



Information to prevent ***minor*** physical injury, component damage, data loss, and/or program corruption when trying to complete a task.



Information to prevent ***serious*** physical injury, component damage, data loss, and/or program corruption when trying to complete a specific task.

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

Table of Contents

Preface ..... **iii**

1 Introduction **1**
1.1 Data Types 1
Function Naming 2

2 Function Library **3**
2.1 List of Functions 3
2.2 System & Initialization 5
2.3 Encoder & Counter 7
2.4 PWM Configuration 10
2.5 TRG-OUT & TTL-OUT Configuration 12
2.6 Comparator 14
2.7 FIFO Comparing 15
2.8 Linear Comparing 19
2.9 Manual Trigger 21
2.10 Digital Input 22
2.11 Latch 23
2.12 Timer 25
2.13 Interrupt 26
2.14 Filtering & Monitoring 29
2.15 Return Errors 31

Important Safety Instructions **33**

Getting Service **35**

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

This page intentionally left blank.

Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

1 Introduction

The LPCIe-8124-C is an advanced quadrature encoder counter board featuring position comparison and high speed trigger pulse output. A state-of-the-art FPGA allows it to meet most machine maker requirements for high frequency encoder input and trigger pulse output.

The LPCIe-8124-C utilizes FPGA and FIFO to ensure that all comparison points automatically load and are compared with internal comparators, eliminating the need for software support. Trigger pulse frequency can be up to 5MHz and encoder input frequency can be up to 20MHz. Noise disturbance is prevented by a digital filter, and, pulse width ia adjustable by software. Comparator output mutually corresponds to trigger output. Trigger output pulses are counted and the result stored in the trigger counter.

The LPCIe-8124-C offers board index design supporting one unique index number in the PC system. For input and output channel, TTL (without isolation) and Open Collector (with isolation) are available, with the former meeting high frequency demands and the latter providing circuit protection.

ADLINK provides a Windows WDM driver and DLL function libraries aid application development.

The functions listed support development of programs in C, C++, and Visual Basic. If Delphi is used as the programming environment, it is necessary to transform the header files, pci_8124.h manually.

1.1 Data Types

The library uses these data types in **pci_8124.h**. We suggest you use these data types in your application programs. Data type names and ranges in C/C++ are as follows

Type	Description	Range
U8	8-bit ASCII character	0 to 255
I16	16-bit signed integer	-32768 to 32767

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

Type	Description	Range
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

Function Naming

The LPCIe-8124-C software drivers use full names to identify the functions. Naming conventions are as follows.

In a 'C' programming environment:

_{hardware_model}_{action_name}. e.g. **`_8124_initial()`**.

To differentiate between a C library and a VB library, a capital "B" is placed at the beginning of each function name e.g. **`B_8124_initial()`**.

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

2 Function Library

2.1 List of Functions

Category	Function
System & Initialization	_8124_initial
	_8124_close
	_8124_get_version
Encoder & Counter	_8124_set_encoder_input_mode
	_8124_set_encoder_value
	_8124_get_encoder_value
	_8124_set_encoder_up_down_count
	_8124_set_ez_clear
	_8124_set_counter_source
PWM Configuration	_8124_set_pwm_source
	_8124_set_pwm_mode
	_8124_set_pwm_pulse_width
	_8124_set_pwm_toggle_dir
	_8124_get_toggle_sts
	_8124_set_pwm_trigger_monitor_type
	_8124_get_pwm_trigger_monitor_type
TRG-OUT & TTL-OUT Configuration	_8124_set_trigger_source
	_8124_set_trgOut_logic
	_8124_set_ttlOut_source
	_8124_set_ttlOut
Comparator	_8124_set_comparator_data
	_8124_get_comparator_data

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

Category	Function
FIFO Comparison	_8124_reset_fifo
	_8124_get_fifo_sts
	_8124_set_fifo_data
	_8124_set_fifo_array
	_8124_set_fifo_shift
	_8124_set_fifo_level
	_8124_get_fifo_level
	_8124_get_fifo_empty_sts
Linear Comparison	_8124_set_linear_compare
	_8124_set_linear_source
	_8124_enable_linear_set
Manual Trigger	_8124_manual_trigger
Digital input	_8124_get_di_sts
Latch	_8124_set_latch
	_8124_get_latch_event_sts
	_8124_get_latch_value
Timer	_8124_set_Timer
	_8124_start_timer
Interrupt	_8124_set_int_factor
	_8124_get_int_factor
	_8124_wait_single_int
	_8124_reset_int
Filtering & Monitoring	_8124_set_input_filter_enable
	_8124_get_noise_detect
	_8124_get_pwm_trigger_count
	_8124_reset_pwm_trigger_count
	_8124_get_linear_next_compare
	_8124_get_linear_remain_trigger_count

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

2.2 System & Initialization

_8124_initialSw:

Initializes the LPCIe-8124-C cards in your system. All LPCIe-8124-C 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 TRG-OUT & TTL-OUT Configuration "_8124_set_trigger_source".

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

Syntax

C/C++ (Windows XP/7/8.1)

```
I16 _8124_initialSw (I32 *CardIdInBit, I16
     ManualId, I16 DefaultSw);
```

Visual Basic (Windows XP/7/8.1)

```
B_8124_initialSw (CardIdInBit As Integer, ByVal
     ManualId As Integer, ByVal DefaultSw As
     Integer) As Integer
```

_8124_close:

Closes the LPCIe-8124-C cards and releases resources, must be called at the end of your applications.

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

Syntax

C/C++ (Windows XP/7/8.1)

```
I16 _8124_close();
```

Visual Basic (Windows XP/7/8.1)

```
B_8124_close () As Integer
```

C/C++ (Windows XP/7/8.1)

_8124_get_version:

Retrieves hardware and software version information.

Syntax

C/C++ (Windows XP/7/8.1)

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

Visual Basic (Windows XP/7/8.1)

```
B_8124_get_version (ByVal CardId As Integer,
     Firmware_ver As Long, Driver_ver As Long,
     DLL_ver As Long) As Integer
```

Arguments

`I32 *CardIdInBit`: The information of LPCIe-8124-C cards' card ID in bit format.

- Ex. CardIdInBit = 0x0009; It means there are two LPCIe-8124-C 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)

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

I16 DefaultSw: Default output setting.

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

I16 CardId: The LPCIe-8124-Ccard's ID number. (0 ~ 15)

I32 *Firmware_ver: The firmware version.

I32 *Driver_ver: The device driver version.

I32 *DLL_ver: The DLL' version.

2.3 Encoder & Counter

_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 the appropriate encoder input mode before using counter.

Syntax

C/C++ (Windows XP/7/8.1)

```
I16 _8124_set_encoder_input_mode( I16 CardId, I16
     Channel, I16 IptMode );
```

Visual Basic (Windows XP/7/8.1)

```
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
```

_8124_set_encoder_value:

Sets the encoder (counter) value directly

Syntax

C/C++ (Windows XP/7/8.1)

```
I16 _8124_set_encoder_value( I16 CardId, I16
     Channel, I32 EncValue );
```

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

Visual Basic (Windows XP/7/8.1)

```
B_8124_set_encoder_value (ByVal CardId As
     Integer, ByVal Channel As Integer, ByVal
     IptMode As Integer) As Integer
```

_8124_get_encoder_value:

Reads the current encoder value.

Syntax

C/C++ (Windows XP/7/8.1)

```
I16 _8124_get_encoder_value( I16 CardId, I16
     Channel, I32 *EncValue );
```

Visual Basic (Windows XP/7/8.1)

```
B_8124_get_encoder_value (ByVal CardId As
     Integer, ByVal Channel As Integer, EncValue
     As Long) As Integer
```

_8124_set_encoder_up_down_count:

Sets the specified encoder to count up or count down.

Syntax

C/C++ (Windows XP/7/8.1)

```
I16 _8124_set_encoder_up_down_count(I16 CardId,
     I16 Channel, I16 Inverse );
```

Visual Basic (Windows XP/7/8.1)

```
B_8124_set_encoder_up_down_count (ByVal CardId As
     Integer, ByVal Channel As Integer, ByVal
     Inverse As Integer) As Integer
```

_8124_set_ez_clear:

Enables the counter clear function by sensing EZ input signal. If enabled, you should choose falling edge or rising edge to clear the counter. If disabled, the “ClrLogic” setting will be ignored.

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

Syntax

C/C++ (Windows XP/7/8.1)

```
I16 _8124_set_ez_clear( I16 CardId, I16 Channel,
     I16 Enable, I16 ClrLogic );
```

Visual Basic (Windows XP/7/8.1)

```
B_8124_set_ez_clear (ByVal CardId As Integer,
     ByVal Channel As Integer, ByVal Enable As
     Integer, ByVal ClrLogic As Integer) As
     Integer
```

_8124_set_counter_source:

Sets the counter source. This could be one of the EA/EB input channels or timers. The default counter source is EA/EB. Refer to the section 5.12 Timer for more information.

Syntax

C/C++ (Windows XP/7/8.1)

```
I16 _8124_set_counter_source( I16 CardId, I16
     Channel, I16 CtnSrc );
```

Visual Basic (Windows XP/7/8.1)

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

Arguments

`I16 CardId`: The LPCIe-8124-Ccard'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

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

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

2.4 PWM Configuration

_8124_set_pwm_source:

Sets the PWM source (trigger source); select from LIN, CMP (default), and timer. The sources are independent and can be enabled simultaneously.

Syntax

C/C++ (Windows XP/7/8.1)

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

Visual Basic (Windows XP/7/8.1)

```
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
```

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

_8124_set_pwm_mode:

Sets the TRG output; select from pulse type output and level-switch output.

Syntax

C/C++ (Windows XP/7/8.1)

```
I16 _8124_set_pwm_mode(  I16 CardId, I16 Channel,
     I16 PulseOrToggle );
```

Visual Basic (Windows XP/7/8.1)

```
B_8124_set_pwm_mode (ByVal CardId As Integer,
     ByVal Channel As Integer, ByVal
     PulseOrToggle As Integer) As Integer
```

_8124_set_pwm_pulse_width:

If pulse type output is PWM output, use this function to adjust pulse width.

Syntax

C/C++ (Windows XP/7/8.1)

```
I16 _8124_set_pwm_pulse_width( I16 CardId, I16
     Channel, I16 WidthPara );
```

Visual Basic (Windows XP/7/8.1)

```
B_8124_set_pwm_pulse_width (ByVal CardId As
     Integer, ByVal Channel As Integer, ByVal
     WidthPara As Integer) As Integer
```

_8124_set_pwm_toggle_dir:

If level-switch output is PWM output, use this function to toggle between conditions; select from 3 conditions (count directions).

Syntax

C/C++ (Windows XP/7/8.1)

```
I16 _8124_set_pwm_toggle_dir(  I16 CardId, I16
     Channel, I16 ToggleDir );
```

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

Visual Basic (Windows XP/7/8.1)

```
B_8124_set_pwm_toggle_dir (ByVal CardId As
     Integer, ByVal Channel As Integer, ByVal
     ToggleDir As Integer) As Integer
```

_8124_get_toggle_sts:

Retrieves the current logic of TRGx output signal.

Syntax

C/C++ (Windows XP/7/8.1)

```
I16 _8124_get_toggle_sts( I16 CardId, I16
     Channel, I16 *Sts );
```

Visual Basic (Windows XP/7/8.1)

```
B_8124_get_toggle_sts(ByVal CardId As Integer,
     ByVal Channel As Integer, Sts As Integer) As
     Integer
```

_8124_set_pwm_trigger_monitor_type:

Sets how PWM trigger counter counts and when PWM interrupt asserts. Set the counter counts and PWM interrupt asserts at rising edge, falling edge or both edges of trigger signal. Use _8124_get_pwm_trigger_count to get counter value.

Syntax

C/C++ (Windows XP/7/8.1)

```
I16 _8124_set_pwm_trigger_monitor_type(I16
     CardId, I16 Type);
```

Visual Basic (Windows XP/7/8.1)

```
B_8124_set_pwm_trigger_monitor_type(ByVal CardId
     As Integer, ByVal Type As Integer) As
     Integer
```

_8124_get_pwm_trigger_monitor_type:

Retrieves the current PWM trigger monitor type.

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

Syntax

C/C++ (Windows XP/7/8.1)

```
I16 _8124_get_pwm_trigger_monitor_type(I16
     CardId, I16 *Type);
```

Visual Basic (Windows XP/7/8.1)

```
B_8124_get_pwm_trigger_monitor_type(ByVal CardId
     As Integer, Type As Integer) As Integer
```

Arguments

I16 CardId: The LPCIe-8124-Ccard'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: Countdown

I16 Sts: logic of TRGx output signal

- > 0: low level
- > 1: high level

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

I16 Type: PWM trigger counter counts and when PWM interrupt asserts type

- > 0: rising edge
- > 1: falling edge
- > 2: both edge

2.5 TRG-OUT & TTL-OUT Configuration

_8124_set_trigger_source:

Select from two TGR sources; from SW2 setting and from PWM or DI signals. If SW2 setting is selected, the function will ignore the settings of “PwmInBit”, “TtlInInBit” and “EzInBit”. If PWM/TTL-IN/EZ is selected, other parameters should be set next; they can be set to “OR” output.

_8124_set_trgOut_logic:

Set the logic of TRG –OUT output signal.

Syntax

C/C++ (Windows XP/7/8.1)

```
I16 _8124_set_trigger_source( I16 CardId, I16
     Channel, I16 GroupSel, I16 PwmInBit, I16
     TtlInInBit, I16 EzInBit );
```

Visual Basic (Windows XP/7/8.1)

```
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
```

_8124_set_ttlOut_source:

Sets the source of TTL-OUT output. Select from two sources of TTL-OUT output; DO output, and the relative channel of TRG-OUT output.

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

Syntax

C/C++ (Windows XP/7/8.1)

```
I16 _8124_set_ttlOut_source( I16 CardId, I16
     Channel, I16 TtlOutSrc );
```

Visual Basic (Windows XP/7/8.1)

```
B_8124_set_ttlOut_source (ByVal CardId As
     Integer, ByVal Channel As Integer, ByVal
     TtlOutSrc As Integer) As Integer
```



Since TRG-OUT logic outputs can be inverted, the logic of TRG-OUT output may be opposite to TTL-OUT output.

_8124_set_ttlOut:

Controls TTL-OUT when TTL-OUT is set as digital output (DO).

Syntax

C/C++ (Windows XP/7/8.1)

```
I16 _8124_set_ttlOut( I16 CardId, I16 Channel,
     I16 TtlOutValue );
```

Visual Basic (Windows XP/7/8.1)

```
B_8124_set_ttlOut (ByVal CardId As Integer, ByVal
     Channel As Integer, ByVal TtlOutValue As
     Integer) As Integer
```

Arguments

`I16 CardId`: The LPCIe-8124-Ccard'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

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

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.

2.6 Comparator

_8124_set_comparator_data:

Overrides the current comparing data to the comparator (CMP).

Syntax

C/C++ (Windows XP/7/8.1)

```
I16 _8124_set_comparator_data( I16 CardId, I16
     Channel, I32 CmpData );
```

Visual Basic (Windows XP/7/8.1)

```
B_8124_set_comparator_data (ByVal CardId As Inte-
ger, ByVal Channel As Integer, ByVal CmpData As
Long) As Integer
```

_8124_get_comparator_data:

Retrieves the current comparing data from comparator (CMP).

Syntax

C/C++ (Windows XP/7/8.1)

```
I16 _8124_get_comparator_data( I16 CardId, I16
     Channel, I32 *CmpData );
```

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

Visual Basic (Windows XP/7/8.1)

```
B_8124_get_comparator_data (ByVal CardId As Inte-
ger, ByVal Channel As Integer, CmpData As Long) As
Integer
```

Arguments

I16 CardId: The LPCIe-8124-Ccard's ID number.

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

I32 CmpData: The current comparing data in comparator.

2.7 FIFO Comparison

_8124_reset_fifo:

Resets (clear) the FIFO; FIFO status will show "Empty" status.

Syntax

C/C++ (Windows XP/7/8.1)

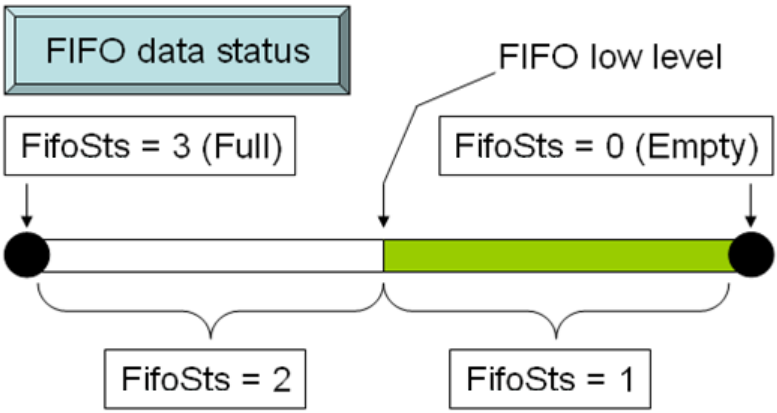
```
I16 _8124_reset_fifo( I16 CardId, I16 Channel );
```

Visual Basic (Windows XP/7/8.1)

```
B_8124_reset_fifo (ByVal CardId As Integer, ByVal
     Channel As Integer) As Integer
```

_8124_get_fifo_sts:

Retrieves the current FIFO status. This figure shows the four statuses which may be returned.



Syntax

C/C++ (Windows XP/7/8.1)

```
I16 _8124_get_fifo_sts( I16 CardId, I16 Channel,
     I16 *FifoSts );
```

Visual Basic (Windows XP/7/8.1)

```
B_8124_get_fifo_sts (ByVal CardId As Integer,
     ByVal Channel As Integer, FifoSts As
     Integer) As Integer
```

_8124_set_fifo_data:

Sets the specified comparing data to the FIFO. The FIFO capacity is 1023. When FIFO status is full, the data cannot be set into FIFO. This function does not check FIFO status.

Syntax

C/C++ (Windows XP/7/8.1)

```
I16 _8124_set_fifo_data( I16 CardId, I16 Channel,
     I32 FifoData );
```

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

Visual Basic (Windows XP/7/8.1)

```
B_8124_set_fifo_data (ByVal CardId As Integer,
     ByVal Channel As Integer, ByVal FifoData As
     Long) As Integer
```

_8124_set_fifo_array:

Sets comparing data array to the FIFO.

Syntax

C/C++ (Windows XP/7/8.1)

```
I16 _8124_set_fifo_array( I16 CardId, I16
     Channel, I32 *DataArr, I16 ArraySize );
```

Visual Basic (Windows XP/7/8.1)

```
B_8124_set_fifo_array (ByVal CardId As Integer,
     ByVal Channel As Integer, DataArr As Long,
     ByVal ArraySize As Integer) As Integer
```

_8124_set_fifo_shift:

Shifts the FIFO data to comparator (CMP) manually. When using the FIFO comparing, the first compared data must typically be shifted to the CMP manually via this function.

Syntax

C/C++ (Windows XP/7/8.1)

```
I16 _8124_set_fifo_shift( I16 CardId, I16 Channel
     );
```

Visual Basic (Windows XP/7/8.1)

```
B_8124_set_fifo_shift (ByVal CardId As Integer,
     ByVal Channel As Integer) As Integer
```

_8124_set_fifo_level:

Set the FIFO low level; select from 4 conditions. The default setting is 512 elements remaining in FIFO.

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

Syntax

C/C++ (Windows XP/7/8.1)

```
I16 _8124_set_fifo_level( I16 CardId, I16
     Channel, I16 Level );
```

Visual Basic (Windows XP/7/8.1)

```
B_8124_set_fifo_level (ByVal CardId As Integer,
     ByVal Channel As Integer, ByVal Level As
     Integer) As Integer
```

_8124_get_fifo_level:

Retrieves the FIFO low level setting.

Syntax

C/C++ (Windows XP/7/8.1)

```
I16 _8124_get_fifo_level( I16 CardId, I16
     Channel, I16 *Level );
```

Visual Basic (Windows XP/7/8.1)

```
B_8124_get_fifo_level (ByVal CardId As Integer,
     ByVal Channel As Integer, Level As Integer)
     As Integer
```

_8124_get_fifo_empty_sts:

Checks for compared data remaining in hardware buffer. Hardware buffer includes a FIFO and register. _8124_get_fifo_sts can check if FIFO is empty, but cannot check if compared data remains in the register. Use this to check if the FIFO and the register are all empty; all data are compared.

Syntax

C/C++ (Windows XP/7/8.1)

```
I16 _8124_get_fifo_empty_sts( I16 CardId, I16
     Channel, I16 *Sts);
```

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

Visual Basic (Windows XP/7/8.1)

```
B_8124_get_fifo_empty_sts(ByVal CardId As
     Integer, ByVal Channel As Integer, Sts As
     Integer) As Integer;
```

@ Argument

I16 CardId: The LPCIe-8124-Ccard'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)

I16 Sts: hardware buffer status

- 0: there are compared data remaining in hardware buffer
- 1: hardware buffer is empty (all data are compared)

2.8 Linear Comparison

_8124_set_linear_compare:

Configures the linear comparing trigger function. PWM from linear function must first be enabled, using "_8124_set_pwm_source". LPCIe-8124-Chas 10 linear sets can be used for linear, and can be compared with any counter

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

arbitrarily. Every set has 3 parameters; starting point (StartPoint), repeat times (RepeatTimes) and trigger interval (Interval). The total trigger pulse is (RepeatTimes + 1) times. For example, if RepeatTimes is 10, total output pulse will be 11. This formula calculates 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 dependent on the Interval signature. If RepeatTimes is set to 0, the linear comparing will not stop until counting stops or linear function is reset.

Syntax

C/C++ (Windows XP/7/8.1)

```
I16 _8124_set_linear_compare( I16 CardId, I16
     SetNum, I16 Channel, I32 StartPoint, F64
     RepeatTimes, I16 Interval );
```

Visual Basic (Windows XP/7/8.1)

```
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
```

_8124_set_linear_source:

Assigns which linear sets will be used in the specified trigger channel; select from 10 sets. One trigger channel can have more then one linear set.

Syntax

C/C++ (Windows XP/7/8.1)

```
I16 _8124_set_linear_source( I16 CardId, I16
     Channel, I16 SetNumInBit );
```

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

Visual Basic (Windows XP/7/8.1)

```
B_8124_set_linear_source( ByVal CardId As
     Integer, ByVal Channel As Integer, ByVal
     SetNumInBit As Integer ) As Integer
```

_8124_enable_linear_set:

Enables linear sets. Disable linear sets when no longer needed, to prevent unexpected triggering

Syntax

C/C++ (Windows XP/7/8.1)

```
I16 _8124_enable_linear_set( I16 CardId, I16
     SetNum, I16 Enable );
```

Visual Basic (Windows XP/7/8.1)

```
B_8124_enable_linear_set( ByVal CardId As
     Integer, ByVal SetNum As Integer, ByVal
     Enable As Integer )As Integer
```

@ Argument

I16 CardId: The LPCIe-8124-Ccard'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

2.9 Manual Trigger

_8124_manual_trigger:

Generates one pulse manually.

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

Syntax

C/C++ (Windows XP/7/8.1)

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

Visual Basic (Windows XP/7/8.1)

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

@ Argument

`I16 CardId`: The LPCIe-8124-Ccard's ID number.

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

2.10 Digital Input

_8124_get_di_sts:

Retrieves the status of digital inputs; select from TTL-IN, LTC and EZ. The parameters are showed in bit format. The former four bits of parameters represent 4 channels of each digital input.

Syntax

C/C++ (Windows XP/7/8.1)

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

Visual Basic (Windows XP/7/8.1)

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

@ Argument

`I16 CardId`: The LPCIe-8124-Ccard's ID number.

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

`I16 *LtcStsInBit`: The LTC status in bit format.

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

I16 *EzStsInBit: The EZ status in bit format.

2.11 Latch

_8124_set_latch:

Enables/disables the high-speed position capture (latch) function and sets latch condition. When enabled and latch condition is configured correctly (Rising or falling active), the current position will be latched when it receive the latch signal. Check latch events by polling "_8124_get_latch_event_sts" or using the interrupt function. See also section 5.13 Interrupt function.

Syntax

C/C++ (Windows XP/7/8.1)

```
I16 _8124_set_latch( I16 CardId, I16 Channel, I16
     Enable, I16 EdgeSel );
```

Visual Basic (Windows XP/7/8.1)

```
B_8124_set_latch (ByVal CardId As Integer, ByVal
     Channel As Integer, ByVal Enable As Integer,
     ByVal EdgeSel As Integer) As Integer
```

_8124_get_latch_event_sts:

Checks latch event status. If latch function is enabled and latch condition is met, the status will change from 0 from 1. The event status is read-clear.

Syntax

C/C++ (Windows XP/7/8.1)

```
I16 _8124_get_latch_event_sts( I16 CardId, I16
     *LatchEventInBit );
```

Visual Basic (Windows XP/7/8.1)

```
B_8124_get_latch_event_sts (ByVal CardId As
     Integer, LatchEventInBit As Integer) As
     Integer
```

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

_8124_get_latch_value:

Retrieves the latched counter value when latch event occurs.

Syntax

C/C++ (Windows XP/7/8.1)

```
I16 _8124_get_latch_value(I16 CardId, I16
     Channel, I32 *LatchValue );
```

Visual Basic (Windows XP/7/8.1)

```
B_8124_get_latch_value (ByVal CardId As Integer,
     ByVal Channel As Integer, LatchValue As
     Long) As Integer
```

@ Argument

`I16 CardId`: The LPCIe-8124-Ccard'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.

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

2.12 Timer

_8124_set_Timer:

Configures the timer. Start timer by external signal (TTL-IN signal), or by software function. Stop the timer with the function "_8124_start_timer".

Syntax

C/C++ (Windows XP/7/8.1)

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

Visual Basic (Windows XP/7/8.1)

```
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
```

_8124_start_timer:

Starts/stop timer manually.

Syntax

C/C++ (Windows XP/7/8.1)

```
I16 _8124_start_timer( I16 CardId, I16 Channel,
     I16 Start );
```

Visual Basic (Windows XP/7/8.1)

```
B_8124_start_timer (ByVal CardId As Integer,
     ByVal Channel As Integer, ByVal Start As
     Integer) As Integer
```

@ Argument

`I16 CardId`: The LPCIe-8124-Ccard's ID number.

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

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

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

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

2.13 Interrupt

_8124_set_int_factor:

Enables/disables interrupt source. Wait for the specified interrupt using "_8124_wait_single_int". This table shows the interrupt factor architecture of each channel.

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

(*): reserved bit (set to 0)

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

Syntax

C/C++ (Windows XP/7/8.1)

```
I16 _8124_set_int_factor( I16 CardId, I16
     Channel, I16 IntFactorsInBit );
```

Visual Basic (Windows XP/7/8.1)

```
B_8124_set_int_factor (ByVal CardId As Integer,
     ByVal Channel As Integer, ByVal
```

_8124_get_int_factor:

Retrieves current configuration of interrupt factors.

Syntax

C/C++ (Windows XP/7/8.1)

```
I16 _8124_get_int_factor( I16 CardId, I16
     Channel, I16 *IntFactorsInBit );
```

Visual Basic (Windows XP/7/8.1)

```
B_8124_get_int_factor (ByVal CardId As Integer,
     ByVal Channel As Integer, IntFactorsInBit As
     Integer) As Integer
```

_8124_wait_single_int:

Waits for a specific interrupt when interrupt function for specified factors using “_8124_set_int_factor” is enabled. When this function runs, the process will not stop until the event is triggered or the function times out. This function returns under one of these conditions:

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. No processor time is used while waiting for the INT state to become signaled or the time-out interval to elapse.

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

Syntax

C/C++ (Windows XP/7/8.1)

```
I16 _8124_wait_single_int( I16 CardId, I16
      Channel, I16 FactorBitNum, I32 TimeOutMs );
```

Visual Basic (Windows XP/7/8.1)

```
B_8124_wait_single_int (ByVal CardId As Integer,
      ByVal Channel As Integer, ByVal FactorBitNum
      As Integer, ByVal TimeOutMs As Long) As
      Integer
```

_8124_reset_int:

Sets the specified interrupt event to a non-signaled state. Typically put into program before the "_8124_wait_single_int" function.

Syntax

C/C++ (Windows XP/7/8.1)

```
I16 _8124_reset_int( I16 CardId, I16 Channel, I16
      FactorBitNum );
```

Visual Basic (Windows XP/7/8.1)

```
B_8124_reset_int( ByVal CardId As Integer, ByVal
      Channel As Integer, ByVal FactorBitNum As
      Integer) As Integer
```

Arguments

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

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

`I32 TimeOutMs`: 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)

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

2.14 Filtering & Monitoring

_8124_set_input_filter_enable:

Enables digital filter. Pulse width of input signals "EA, EB, EZ and LTC" must be greater than 100 ns, or the input signal will be ignored (treated as noise).

Available after Firmware version: 0x13.

Syntax

C/C++ (Windows XP/7/8.1)

```
I16 _8124_set_input_filter_enable( I16 CardId,
     I16 Channel, I16 Enable );
```

Visual Basic (Windows XP/7/8.1)

_8124_get_noise_detect:

Detects whether the signal is a noise or not. If input signal width is less than 100 ns, the status will turn ON. The status will be cleared after read.

Available after Firmware version: 0x13.

Syntax

C/C++ (Windows XP/7/8.1)

```
I16 _8124_get_noise_detect( I16 CardId, I16
     Channel, I16 *DetectStatus );
```

Visual Basic (Windows XP/7/8.1)

_8124_get_pwm_trigger_count:

Retrieves PWM trigger counter. Only for debug use. Since it lacks register latch protection (high word (16-31 bits) latch), of the counter when register is read, it may cause problems when the counter is counting up while you read the counter.

Available after Firmware version: 0x13.

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

Syntax

C/C++ (Windows XP/7/8.1)

```
I16 _8124_get_pwm_trigger_count( I16 CardId, I16
     Channel, F64 *PwmCount );
```

Visual Basic (Windows XP/7/8.1)

_8124_reset_pwm_trigger_count:

Resets PWM triggered counter.

Available after Firmware version: 0x13.

Syntax

C/C++ (Windows XP/7/8.1)

```
I16 _8124_reset_pwm_trigger_count( I16 CardId,
     I16 Channel );
```

Visual Basic (Windows XP/7/8.1)

_8124_get_linear_next_compare:

Retrieve the next linear compare data. Only for debug use. Since it lacks register latch protection (high word (16-31 bits) latch), of the counter when register is read, it may cause problems when the counter is count up while you read the counter.

Available after Firmware version: 0x13.

Syntax

C/C++ (Windows XP/7/8.1)

```
I16 _8124_get_linear_next_compare( I16 CardId,
     I16 SetNum, I32 *CmpData );
```

Visual Basic (Windows XP/7/8.1)

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

_8124_get_linear_remain_trigger_count:

Retrieves remaining triggered counter. Countdown counter which checks how many remaining triggers will be output.

Available after Firmware version: 0x13.

Syntax

C/C++ (Windows XP/7/8.1)

```
I16 _8124_get_linear_remain_trigger_count( I16
     CardId, I16 SetNum, F64 *Count );
```

Visual Basic (Windows XP/7/8.1)

Arguments

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

I16 Channel: Channel number: 0 ~ 4

I16 Enable:

- 0: Disable (Default)
- 1: Enable

I16 *DetectStatus: Status in bit format.

- bit 0: Noise occurred form EA, the signal is less than 100ns, this bit will turn ON
- bit 1: Noise occurred form EB, the signal is less than 100ns, this bit will turn ON
- bit 2: Noise occurred form EZ, the signal is less than 100ns, this bit will turn ON
- bit 3: Noise occurred form LTC, the signal is less than 100ns, this bit will turn ON
- bit 4: Pulse lost occurred, If the the signal is fast than decode speed. this bit will turned on.

F64 *PwmCount: 32 bit counter.

I32 *CmpData: Next linear compare data.

F64 *Count: 32 bit count remaining trigger counter.

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

2.15 Return Errors

A list of possible return values in the function library is as follows. If the return value is not zero, one or more errors may have occurred.

.

Code	Name	Description
0	ERR_NoError	No error, function successful
-1	ERR_OSVersion	OS mismatched
-2	ERR_OpenDriverFailed	Device driver opening failed/Driver interface creation failed
-3	ERR_InsufficientMemory	Insufficient system memory
-4	ERR_DeviceNotInitial	Cards not initialized
-5	ERR_NoDeviceFound	Cards not found in system
-6	ERR_CaridDuplicate	Card ID duplicated
-7	ERR_DeviceAlreadyInitialed	Cards already initialized
-8	ERR_InterruptNotEnable	Card interrupt events not enabled
-9	ERR_TimeOut	Function timed out
-10	ERR_ParametersInvalid	Function input parameters invalid
-11	ERR_SetEEPROM	Data setting to EEPROM failed
-12	ERR_GetEEPROM	Data retrieval from EEPROM failed
-1000	ERR_Win32Error	WIN32 API function error, contact ADLINK FAE help

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

Important Safety Instructions

For user safety, please read and follow all **instructions**, **WARNINGS**, **CAUTIONS,** and **NOTES** marked in this manual and on the associated equipment before handling/operating the equipment.

- Read these safety instructions carefully.
- Keep this user's manual for future reference.
- Read the specifications section of this manual for detailed information on the operating environment of this equipment.
- When installing/mounting or uninstalling/removing equipment:
  - Turn off power and unplug any power cords/cables.
- To avoid electrical shock and/or damage to equipment:
  - Keep equipment away from water or liquid sources;
  - Keep equipment away from high heat or high humidity;
  - Keep equipment properly ventilated (do not block or cover ventilation openings);
  - Make sure to use recommended voltage and power source settings;
  - Always install and operate equipment near an easily accessible electrical socket-outlet;
  - Secure the power cord (do not place any object on/over the power cord);
  - Only install/attach and operate equipment on stable surfaces and/or recommended mountings; and,
  - If the equipment will not be used for long periods of time, turn off and unplug the equipment from its power source.

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

- Never attempt to fix the equipment. Equipment should only be serviced by qualified personnel.
- A Lithium-type battery may be provided for uninterrupted, backup or emergency power.

WARNING:

Risk of explosion if battery is replaced with an incorrect type; please dispose of used batteries appropriately.

- Equipment must be serviced by authorized technicians when:
  - The power cord or plug is damaged;
  - Liquid has penetrated the equipment;
  - It has been exposed to high humidity/moisture;
  - It is not functioning or does not function according to the user's manual;
  - It has been dropped and/or damaged; and/or,
  - It has an obvious sign of breakage.

Please pay strict attention to all warnings and advisories appearing on the device, to avoid injury or damage.



LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

Getting Service

Ask an Expert: http://askanexpert.adlinktech.com

ADLINK Technology, Inc.
Address: 9F, No.166 Jian Yi Road, Zhonghe District
New Taipei City 235, Taiwan
新北市中和區建一路 166 號 9 樓
Tel: +886-2-8226-5877
Fax: +886-2-8226-5717
Email: service@adlinktech.com

Ampro ADLINK Technology, Inc.
Address: 5215 Hellyer Avenue, #110
San Jose, CA 95138, USA
Tel: +1-408-360-0200
Toll Free: +1-800-966-5200 (USA only)
Fax: +1-408-360-0222
Email: info@adlinktech.com

ADLINK Technology (China) Co., Ltd.
Address: 上海市浦东新区张江高科技园区芳春路 300 号 (201203)
300 Fang Chun Rd., Zhangjiang Hi-Tech Park
Pudong New Area, Shanghai, 201203 China
Tel: +86-21-5132-8988
Fax: +86-21-5132-3588
Email: market@adlinktech.com

ADLINK Technology Beijing
Address: 北京市海淀区上地东路 1 号盈创动力大厦 E 座 801 室(100085)
Rm. 801, Power Creative E, No. 1 Shang Di East Rd.
Beijing, 100085 China
Tel: +86-10-5885-8666
Fax: +86-10-5885-8626
Email: market@adlinktech.com

ADLINK Technology Shenzhen
Address: 深圳市南山区科技园南区高新南七道 数字技术园
A1 栋 2 楼 C 区 (518057)
2F, C Block, Bldg. A1, Cyber-Tech Zone, Gao Xin Ave. Sec. 7
High-Tech Industrial Park S., Shenzhen, 518054 China
Tel: +86-755-2643-4858
Fax: +86-755-2664-6353
Email: market@adlinktech.com

LiPPERT ADLINK Technology GmbH
Address: Hans-Thoma-Strasse 11
D-68163 Mannheim, Germany
Tel: +49-621-43214-0
Fax: +49-621 43214-30
Email: emea@adlinktech.com

PENTA ADLINK Technology GmbH
Ulrichsbergerstrasse 17
94469 Deggendorf, Germany
Tel: +49 (0) 991 290 94 – 10
Fax: +49 (0) 991 290 94 - 29
Email: emea@adlinktech.com

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

ADLINK Technology, Inc. (French Liaison Office)
Address: 6 allée de Londres, Immeuble Ceylan
91940 Les Ulis, France
Tel: +33 (0) 1 60 12 35 66
Fax: +33 (0) 1 60 12 35 66
Email: france@adlinktech.com

ADLINK Technology Japan Corporation
Address: 〒101-0045 東京都千代田区神田鍛冶町 3-7-4
神田 374 ビル 4F
KANDA374 Bldg. 4F, 3-7-4 Kanda Kajicho,
Chiyoda-ku, Tokyo 101-0045, Japan
Tel: +81-3-4455-3722
Fax: +81-3-5209-6013
Email: japan@adlinktech.com

ADLINK Technology, Inc. (Korean Liaison Office)
Address: 137-881 서울시 서초구 서초대로 326, 802 (서초동, 모인터빌딩)
802, Mointer B/D, 326 Seocho-daero, Seocho-Gu,
Seoul 137-881, Korea
Tel: +82-2-2057-0565
Fax: +82-2-2057-0563
Email: korea@adlinktech.com

ADLINK Technology Singapore Pte. Ltd.
Address: 84 Genting Lane #07-02A, Cityneon Design Centre
Singapore 349584
Tel: +65-6844-2261
Fax: +65-6844-2263
Email: singapore@adlinktech.com

ADLINK Technology Singapore Pte. Ltd. (Indian Liaison Office)
Address: #50-56, First Floor, Spearhead Towers
Margosa Main Road (between 16th/17th Cross)
Malleswaram, Bangalore - 560 055, India
Tel: +91-80-65605817, +91-80-42246107
Fax: +91-80-23464606
Email: india@adlinktech.com

ADLINK Technology, Inc. (Israeli Liaison Office)
Address: 27 Maskit St., Corex Building
PO Box 12777
Herzliya 4673300, Israel
Tel: +972-54-632-5251
Fax: +972-77-208-0230
Email: israel@adlinktech.com

ADLINK Technology, Inc. (UK Liaison Office)
Tel: +44 774 010 59 65
Email: UK@adlinktech.com

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000