DB-8150

Modulized Function Board -
High Speed Trigger

User's Manual

Manual Rev. 2.00

Revision Date: May 20, 2008

Part No: 50-14012-1000

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: CPU: Chipset: BIOS:

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

Table of Contents

List of Tables ..... **iii**

List of Figures ..... **iv**

1 Introduction **1**
- 1.1 Specifications 4
- 1.2 Supported Software 5

2 Installation **7**
- 2.1 Package Contents 7
- 2.2 DB-8150 and External I/F Outline Drawing 8
- 2.3 DB-8150 Hardware Installation 10
  - Hardware configuration 10
  - Installation Procedures 10
  - Troubleshooting: 11
- 2.4 Software Driver Installation 12

3 Signal Connections **13**
- 3.1 CN3 and CN4 Connectors Pin Definition 13
  - CN3 Pin Description: 14
  - CN4 Pin Description: 15
- 3.2 Encoder Signals EA, EB 16
- 3.3 Trigger Out Signals TRG 0-7 18
- 3.4 Digital Input and Digital Output 20

4 Operation Theory **23**
- 4.1 The Trigger Modular Board Overview 23
  - Encoder Channel 25
  - Index Input (EZ) 28
  - Homing with EZ Signal 28
  - Trigger Pulse Width 29
  - Linear Function 29
  - FIFO function 29
  - PWM 30
  - Mapping 30
- 4.2 Trigger Modular Board Operation 31
  - Position Comparison 31
  - Position Latch 32

Linear Function 32
FIFO mode 33
Timer function 33
4.3 Multiple Boards Operation 34
4.4 Interrupt Control 35

5 DB-8150 Utility 39
5.1 Execute the DB-8150 Utility 39
5.2 Interface Overview 40
Main Menu 40
Operation Steps 41
Show Channel Status Item 44
Setting Parameter Menu 46
Setting Mode and Logic Menu 48
Operation Menu 49

6 Function Library 51
6.1 List of Functions 51
6.2 System & Initialization 55
6.3 Encoder & Counter 57
6.4 PWM Configuration 60
6.5 TRG-OUT Configuration 62
6.6 Comparator 64
6.7 FIFO Comparing 66
6.8 Linear Comparing 70
6.9 Manual Trigger 72
6.10 Digital Input 73
6.11 Latch 75
6.12 Timer 77
6.13 Interrupt 79
6.14 Return Error 83

List of Tables

Table 1-1: DB-815x Series 1
Table 3-1: CN3 Pin Description 14
Table 3-2: CN4 Pin Description 15
Table 3-3: Encoder Signals EA, EB 16
Table 3-4: Trigger Out Signals TRG 0-7 18
Table 3-5: Digital Input and Digital Output 20
Table 6-1: Data Type Definitions 51

List of Figures

Figure 1-1: Block Diagram of the DB-8150 3
Figure 2-1: DB-8150 PCB Layout 8
Figure 2-2: DB-8150 Extension Bracket and Cable Layout 9
Figure 2-3: Board Configuration 10
Figure 3-1: CN3 and CN4 Connectors 13
Figure 5-1: Interface Overview 40
Figure 5-2: Parameter Settings 41
Figure 5-3: Setting Mode and Logic 42
Figure 5-4: Operation 43
Figure 5-5: Example of Interrupt Operation 43
Figure 5-6: Show Channel Status Item 45
Figure 5-7: Setting Parameter Menu 47
Figure 5-8: Setting Mode and Logic Menu 48
Figure 5-9: Operation Menu 50

1 Introduction

In order to increase the available functions on the PCI-8154/58 4-axis/8-axis pulse train output control channel carrier board, ADLINK offers four daughter boards providing a variety of functions. These DB-815x daughter boards can only be used with the PCI-8154/58 carriers only. Additional functionality provided by the DB-815x series includes high-speed triggering and distributed I/O control. Daughter boards can be added based on application requirements. The four daughter boards in the DB-815x series are:

Model Name	Primary Function	Description
DB-8150	High Speed Trigger Out	▶ High speed trigger pulse out up to 1 MHz ▶ Simultaneous 8 differential trigger output ▶ 2 channels encoder input (from external I/F or carrier. FIFO/Linear function compared trigger output
DB-8151	HSL Master Controller	▶ High Speed Link for distributed I/O
DB-8152	ECAM Controller	▶ 2 channel encoder input (master & slave) ▶ Master/Slave controller ▶ Easy to implement electronic cam behavior
DB-8153	Motionnet Master Controller	▶ Serial single axis motion control bus ▶ Does not support I/O function

Table 1-1: DB-815x Series

The DB-8150 offers a high speed trigger pulse output function with six separate compared sources. It is an advanced dual encoder counter board with position compare and high speed trigger pulse output capability. The DB-8150 incorporates a state-of-the-art of

FPGA to provide high frequency encoder input and trigger pulse output.

Motion control system and vision inspection system are typically designed individually. For line scan application, the frame grabber needs a high frequency trigger pulse input to scan the whole image line-by-line. The more intensive the lines, the more precise the images obtained. This means high frequency trigger pulse input is a necessity. Moreover, with the advantage of integrating the trigger board and frame grabber together, a vision system purely retrieves the encoder information from servo motors or linear scale. 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 DB-8150 utilizes an FPGA and FIFO to realize this purpose. All comparing points will automatically load into FIFO and be compared with the internal comparators for the first axis. As a result, users do not have to handle this with software. The trigger pulse frequency can be up to 1 MHz and encoder input frequency can be up to 6.5 MHz. With adjustable pulse width, users can change this with software control. Furthermore, the relation between comparator and trigger output is mutually corresponding. One of all external encoder inputs can be selected as the compare source which from external connector, from carrier, or internal timer. Additionally, there is also a FIFO and linear function purposed as compare method.

For input and output channel, this board provides differential types and support simultaneous trigger pulse output up to 8 channels maximum. DB-8150 also offers several coupling for trigger output channel selection, for example, user can link one comparator to any trigger output cannel and link reminder to other trigger out cannel, which is more flexible for applications.

High speed trigger output solution is for those who want:

- Easy to integrate high speed image inspection application.
- Reduce CPU calculating loading in complex motion-vision system.
- Easy to implement multiple simultaneous triggered devices application.

The block diagram of DB-8150 is as follows.

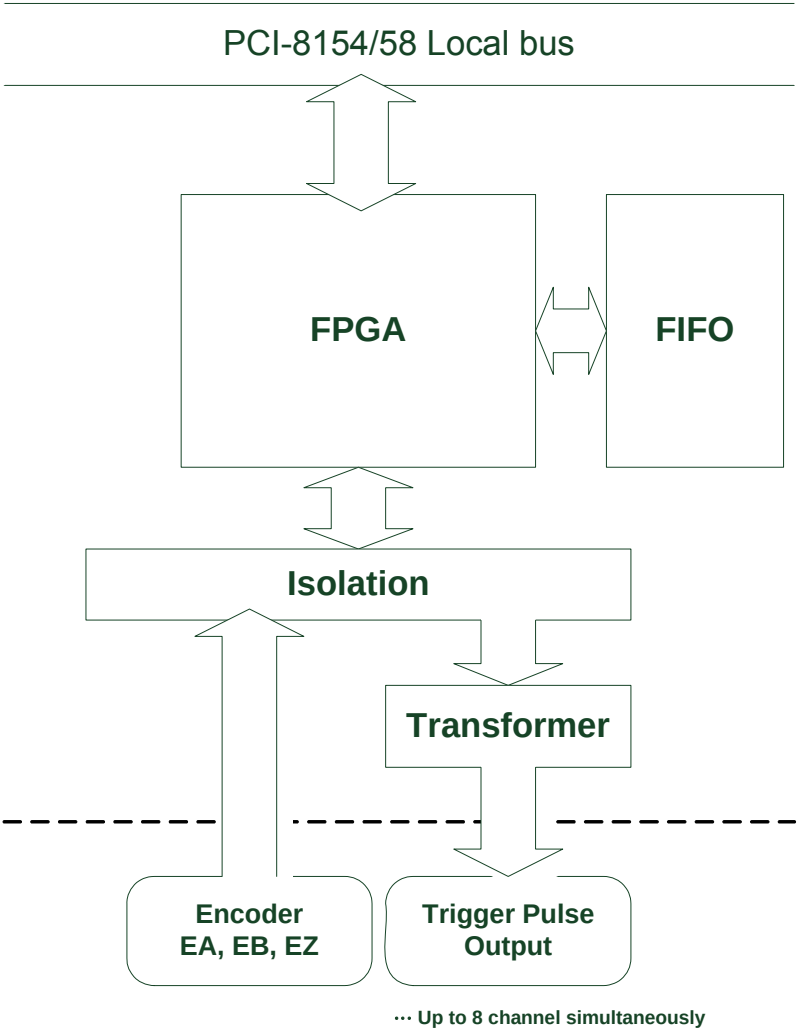


Figure 1-1: Block Diagram of the DB-8150

1.1 Specifications

Interface

- For use with the PCI-8154 and PCI-8158 PCI cards only
- D-Sub 9 and 25 bracket required when using the DB-8150

Encoder Counter and Compare Trigger

- An FPGA is used

Interface

- D-Sub 25: For 8 channel isolated differential trigger output and 2 GPI and 1 GPO.
- D-Sub 9: For 2 channel isolated encoder source (EA2, EB2, EA3, EB3)

Connector

- D-SUB 26P (Standard LPT Port type)
- D-SUB 9P (Standard COM port type)

Interrupt

- PWM, Linear Finish,Counter Latch, Counter Clear, FIFO status event are available

Dimension

- 96.42 (L) x 62 (W) mm

Operating Temperature

- 0 to 60°C

Storage Temperature

- -20 to 80°C

Power Consumption

- +3.3V @ 200 mA (typical)
- +5V @100 mA (typical)

1.2 Supported Software

Program Library

ADLINK provides a Windows WDM driver and DLL function library for the DB-8150. These function libraries are shipped with the board and supports Windows 2000/XP.

2 Installation

This chapter describes how to install the DB-8150. Please follow these steps below:

- Check what you have (Section 2.1, page 7)
- Check the PCB (Section 2.2, page 8)
- Install the hardware (Section 2.3, page 10)
- Install the software driver (Section 2.4, page 12)
- Understanding the I/O signal connections (Chapter 3, page 13) and their operations (Chapter 4, page 23)

2.1 Package Contents

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

- DB-8150: Salve motion control board x1
- Copper Pillar x 4
- Fixed Screws x8
- Extension I/F bracket with 25P and 9P D-SUB connectors x1
- 25 Pins & 10 Pins Flat Cable x1

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 DB-8150 and External I/F Outline Drawing

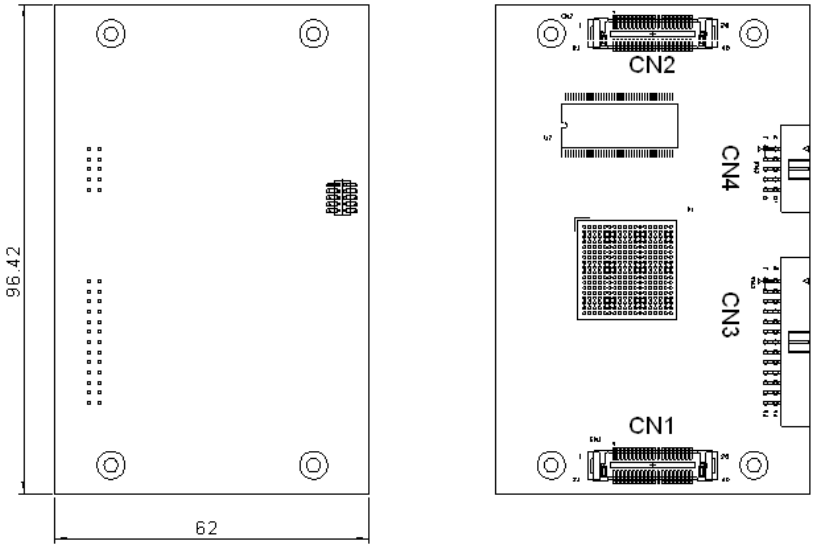


Figure 2-1: DB-8150 PCB Layout

DB-8150:

- CN1, CN2: Daughter Board connect with PCI-8154/58
- CN3: Trigger Pulse Signals (Connecting to Extension I/F)
- CN4: Encoder Feedback Signals (Connecting to Extension I/F)

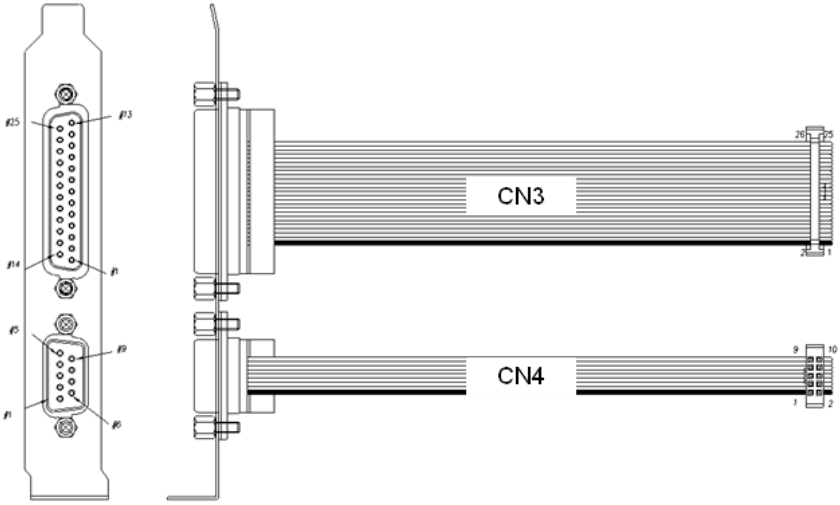


Figure 2-2: DB-8150 Extension Bracket and Cable Layout

DB-8152 Extension I/F:

- CN3: Trigger Pulse Signals
- CN4: Encoder Feedback Signals

2.3 DB-8150 Hardware Installation

2.3.1 Hardware configuration

DB-8150 must be installed on to a PCI-8154 or PCI-8158 only. Please ensure correct orientation of the DB-8150 before it attaching it to the PCI-8154 or PCI-8158.

2.3.2 Installation Procedures

Please follow installation procedure as follows.

Daughter Board Installation

1. Attach the DB-8150 on to the PCI-8154 or PCI-8158. Please ensure correct orientation of the DB-8150 daughter board.
2. Screw the eight screws into the corresponding copper pillars.
3. Connect the DB-8150 and its extension bracket via the 26P and 10P flat cables.

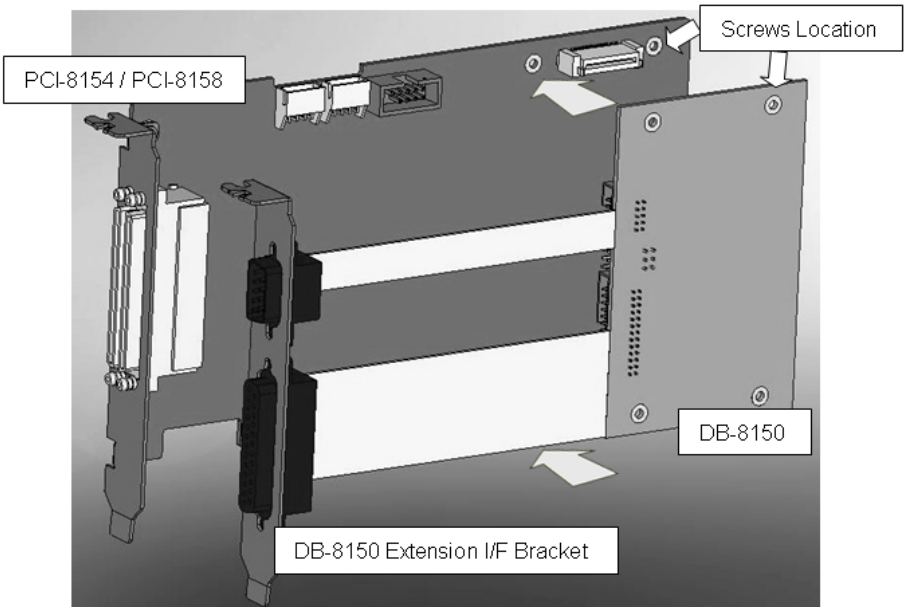


Figure 2-3: Board Configuration

Carrier Board Installation

4. Turn off the computer. Turn off all accessories (printer, modem, monitor, etc.) connected to the computer. Remove the cover from the computer.
5. Select one available 32-bit PCI expansion slot. PCI slots are shorter than ISA or EISA slots and are usually white or ivory.
6. Before handling the PCI-8154 or PCI-8158 with DB-8150 and its extension bracket, 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.
7. Plug the PCI-8154/58 with DB series vertically down into the PCI slot, and then secure the PCI-8154/58 bracket and DB-8150 extension bracket onto rear panel.

2.3.3 Troubleshooting:

If the system doesn't boot or if any erratic behavior of the PCI board is experience, it is most likely caused by an interrupt conflict (possibly an incorrect ISA setup). 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

Execute the following steps:

1. Auto-run the ADLINK All-In-One CD.
2. Follow the procedures of the installation wizard.
3. After setup installation has completed, reboot the system.

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 cable between the DB-8150 and any device.

3.1 CN3 and CN4 Connectors Pin Definition

CN3 offers 8 channels simultaneous trigger pulse output interface. These trigger output type adopts TTL compatible differential signals. In addition to the encoder signals which from carrier, CN4 offers two channels external encodr input interface. For more detailed signal description, please refer to the following section. CN3 and CN4 outlines are illustrated as following figure.

CN3

1	TRG0+	14	TRG0-
2	TRG1+	15	TRG1-
3	TRG2+	16	TRG2-
4	TRG3+	17	TRG3-
5	TRG4+	18	TRG4-
6	TRG5+	19	TRG5-
7	TRG6+	20	TRG6-
8	TRG7+	21	TRG7-
9	EGND	22	EGND
10	DO	23	VDD
11	EGND	24	EGND
12	DI0	25	DI1
13	EX+24V		

CN4

6	DEA2-	1	DEA2+
7	DEB2-	2	DEB2+
8	DEA3-	3	DEA3+
9	DEB3-	4	DEB3+
		5	EGND

Figure 3-1: CN3 and CN4 Connectors

3.1.1 CN3 Pin Description:

CN3 Pin No.	I/O	Signal Name	Description
1	O	TRG0+	Trigger Output 0 Postive
2	O	TRG1+	Trigger Output 1 Postive
3	O	TRG2+	Trigger Output 2 Postive
4	O	TRG3+	Trigger Output 3 Postive
5	O	TRG4+	Trigger Output 4 Postive
6	O	TRG5+	Trigger Output 5 Postive
7	O	TRG6+	Trigger Output 6 Postive
8	O	TRG7+	Trigger Output 7 Postive
9	-	EGND	External Ground
10	O	DO	Digital Output
11	-	EGND	External Ground
12	I	DI0	Digital Input 0
13	I	EX+24V	External +24V Power
14	O	TRG0-	Trigger Output 0 Negative
15	O	TRG1-	Trigger Output 1 Negative
16	O	TRG2-	Trigger Output 2 Negative
17	O	TRG3-	Trigger Output 3 Negative
18	O	TRG4-	Trigger Output 4 Negative
19	O	TRG5-	Trigger Output 5 Negative
20	O	TRG6-	Trigger Output 6 Negative
21	O	TRG7-	Trigger Output 7 Negative
22	-	EGND	External Ground
23	O	VDD	+5V Power
24	-	EGND	External Ground
25	I	DI1	Digital Input 1

Table 3-1: CN3 Pin Description

3.1.2 CN4 Pin Description:

CN4 Pin No.	I/O	Signal Name	Description
1	I	DEA2+	Daughter Board Encoder A2(+)
2	I	DEB2+	Daughter Board Encoder B2(+)
3	I	DEA3+	Daughter Board Encoder A3(+)
4	I	DEB3+	Daughter Board Encoder B3(+)
5	-	EGND	External Ground
6	I	DEA2-	Daughter Board Encoder A2(-)
7	I	DEB2-	Daughter Board Encoder B2(-)
8	I	DEA3-	Daughter Board Encoder A3(-)
9	I	DEB3-	Daughter Board Encoder B3(-)

Table 3-2: CN4 Pin Description

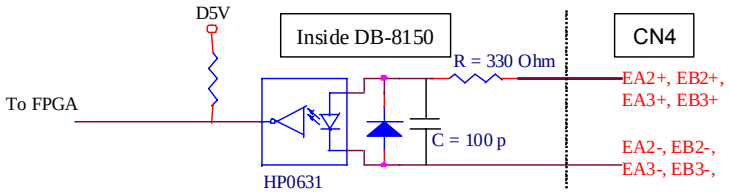
3.2 Encoder Signals EA, EB

There are 2 channel encoder counters with 6 encoder sources could be selected on the DB-8150. Two encoder signals from CN4 and two encoder signals from carrier board and two embedded timer sources can be selected as encoder souece. The following table shows encoder signals on CN4.

CN4 Pin No.	Signal Name	Description
1	DEA2+	Daughter Board Encoder A2(+)
2	DEB2+	Daughter Board Encoder B2(+)
3	DEA3+	Daughter Board Encoder A3(+)
4	DEB3+	Daughter Board Encoder B3(+)
6	DEA2-	Daughter Board Encoder A2(-)
7	DEB2-	Daughter Board Encoder B2(-)
8	DEA3-	Daughter Board Encoder A3(-)
9	DEB3-	Daughter Board Encoder B3(-)

Table 3-3: Encoder Signals EA, EB

The input circuit of the encoder signals is shown as follows:

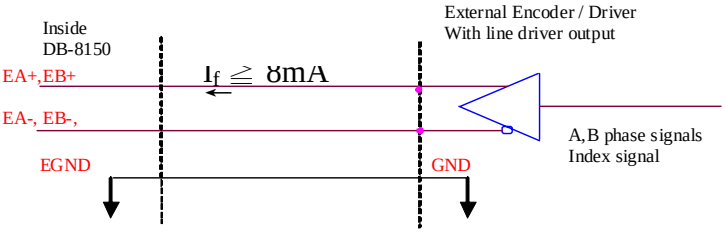


Please note that the voltage across each differential pair of encoder input signals should be at least 3.5 V. Therefore, the output current must be observed when connecting to the encoder feedback or motor driver feedback as not to over drive the source. The differential signal pairs are converted to digital signals EA and EB; then feed to the FPGA.

Below are examples of connecting the input signals with an external circuit. The input circuit can be connected to an encoder or motor driver if it is equipped with: (1) a differential line driver or (2) an open collector output.

Connection to Line Driver Output

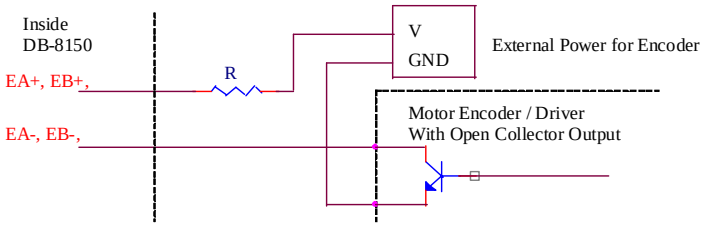
To drive the DB-8150 encoder input, the driver output must provide at least 3.5 V across the differential pairs with at least 8mA driving capacity. The grounds of both sides must be tied together. The maximum frequency of master encoder is 5 Mhz.



Connection to Open Collector Output

To connect with an open collector output, an external power supply is necessary. Some motor drivers can provide the power source. The connection between the DB-8150, encoder, and the power supply is shown in the diagram below. Note that an external current limiting resistor R is necessary to protect the DB-8150 input circuit. The following table lists the suggested resistor values according to the encoder power supply.

Encoder Power (V)	External Resistor R
+5 V	0 Ω (None)
+12 V	1.5 kΩ
+24 V	3.0 kΩ



For more operation information on the encoder feedback signals, refer to Chapter 4

3.3 Trigger Out Signals TRG 0-7

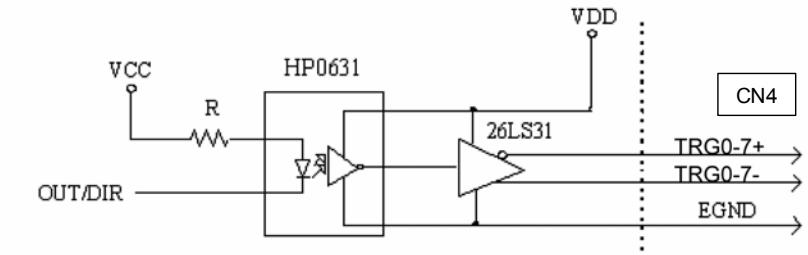
The DB-8150 provides 8 channel high speed diffential trigger output: TRG0 to TRG7 which provide up to 1 MHZ trigger Speed.

The trigger output is located on CN3. Names and pin numbers of these trigger signals are shown below:

CN3 Pin No.	Signal Name	CN3 Pin No.	Signal Name
1	TRG0+	14	TRG0-
2	TRG1+	15	TRG1-
3	TRG2+	16	TRG2-
4	TRG3+	17	TRG3-
5	TRG4+	18	TRG4-
6	TRG5+	19	TRG5-
7	TRG6+	20	TRG6-
8	TRG7+	21	TRG7-

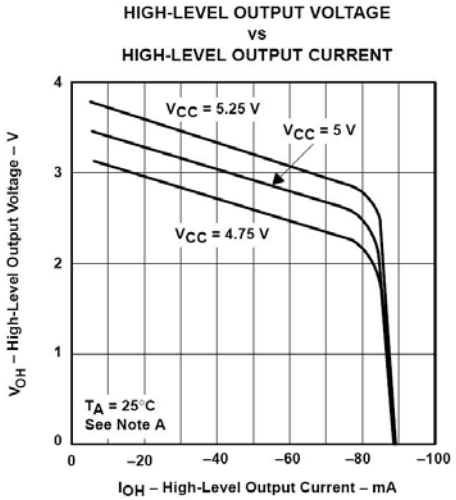
Table 3-4: Trigger Out Signals TRG 0-7

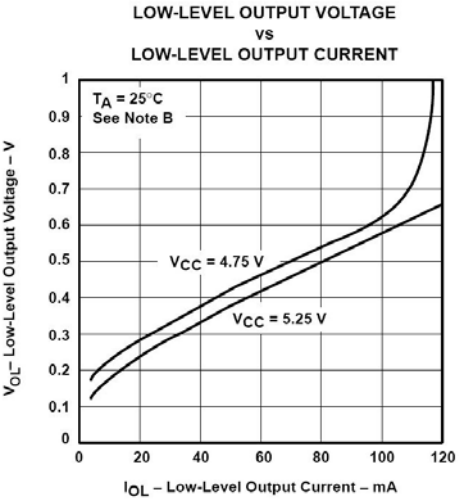
The following wiring diagram is for trigger signals on the 8 channel:



Warning: The sink and driving current must not exceed 20mA or the 2631 will be damaged!

More detail characteristics of 26LS31 are shown as follow:





3.4 Digital Input and Digital Output

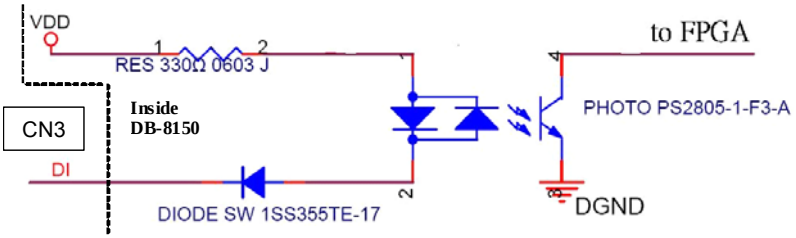
DB-8150 has one opto-isolated digital output for general purpose use and two open collector digital inputs for counter latch use. Pin assignments are illustrated in the table below:

CN3 Pin No	Signal Name
12	DI0
25	DI1
10	DO

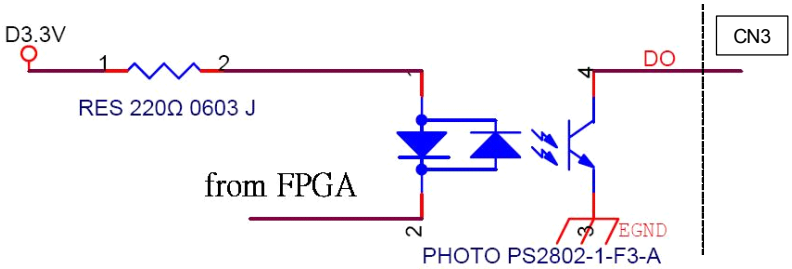
Table 3-5: Digital Input and Digital Output

When DIN is given “LOW”, the voltage should be lower than 0.5V to ensure PS2805 in turn on status.

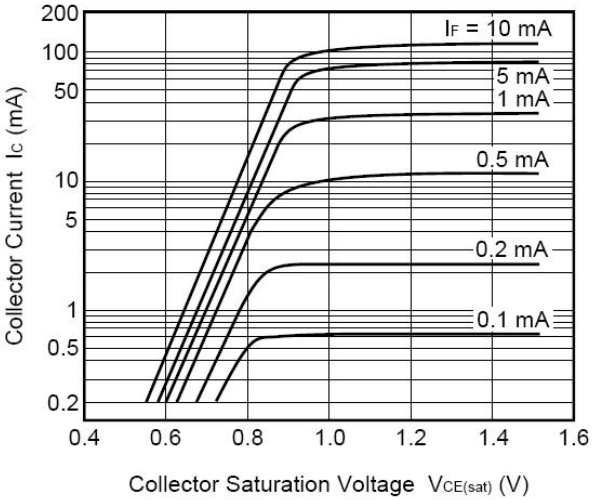
Isolated Input channels diagram



Isolated Output channels diagram



More detail characteristics of PS2802 are shown as follow:

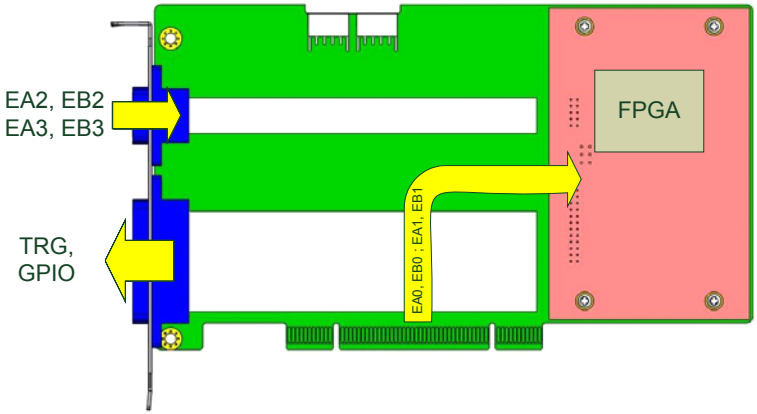


4 Operation Theory

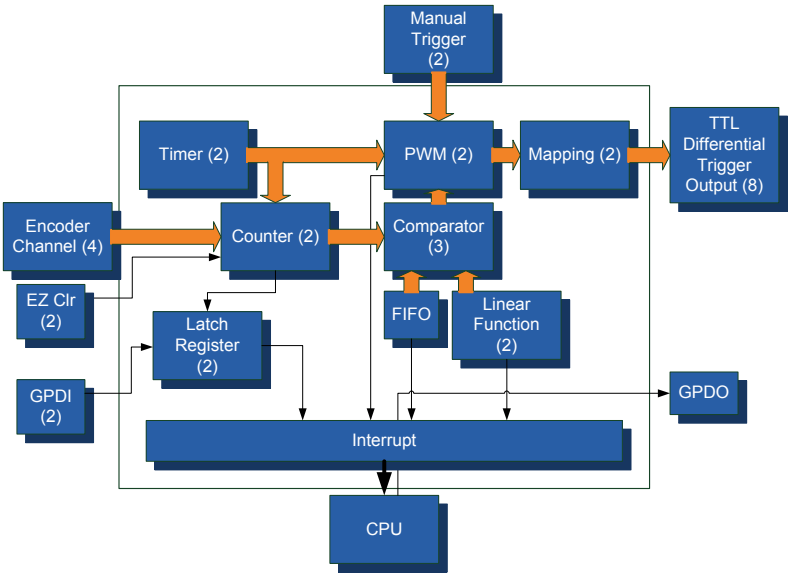
This chapter describes the detail operation of the DB-8150 board.

4.1 The Trigger Modular Board Overview

The DB-8150 was assembled with PCI-8154/58, hence there is overview of motion and high speed trigger board as follows:



Inside the DB-8150, all the position comparison and mapping tasks will be executed within the FPGA. The function block figures as follows:



The FIFO will automatically load the comparing points into 32-bit comparator herein. Users are able to set the comparing source as four encoder inputs and two timer channels. Due to mapping logic setting, the several combinations are provides for user to perform continuously trigger output function. The comparator will compare the data from counter and FIFO individually. Besides, the comparator is also able to compare the counter and comparing data which was produced by embedded Linear function. At very short time, user can retain and retrieve the current counter data by general purposed digital input. Linear function will automatically load the ext comparing points with fixed incremental value into 32-bit comparator. Like FIFO usage purpose, the new comparing points will be loaded once the previous comparing point is compared and pulse is triggered.

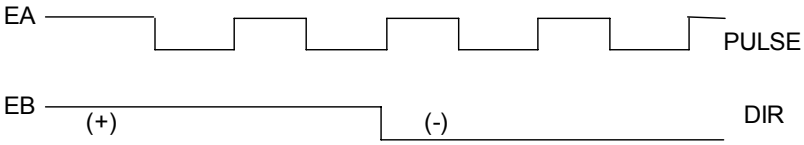
4.1.1 Encoder Channel

The DB-8150 has a 32-bit binary up/down counter managing the present position feedback for whole dual channel. The counter counts signal inputs from the EA and EB pins of CN4 or from the EA and EB pins of PCI-8154/58 individually.

It accepts 3 types of pulse inputs: (1). Single pulse mode (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. 4x AB phase mode is the most commonly used in incremental encoder inputs. 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.

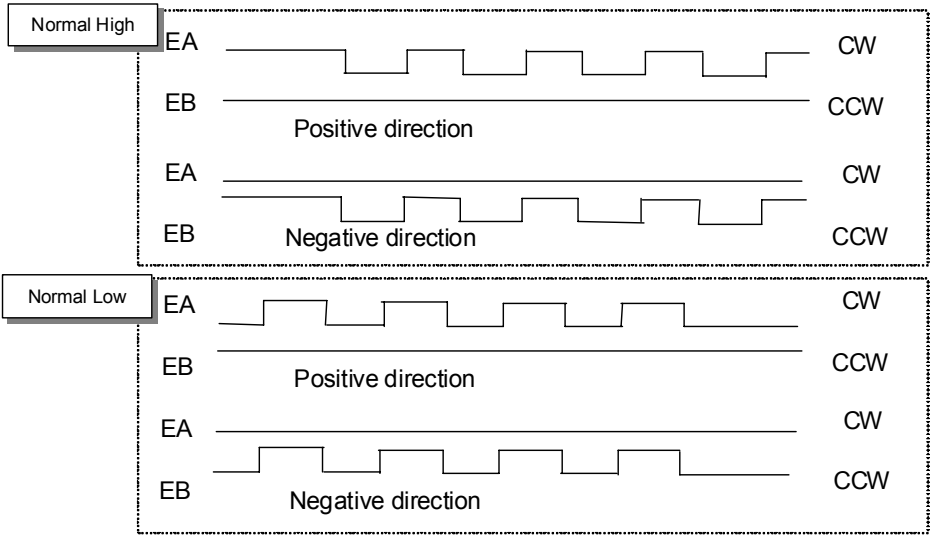
Single Pulse Mode (PULSE/DIR Mode)

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)

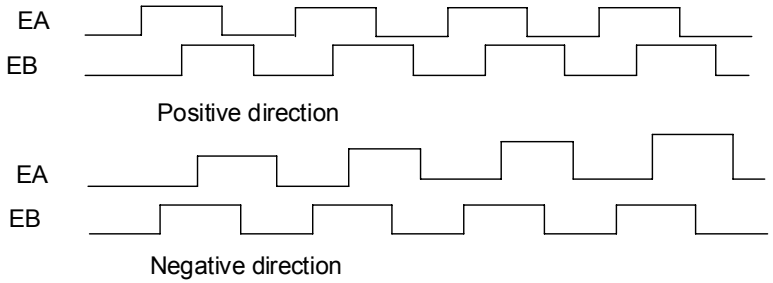
In this mode, EA is dedicated to count the pulses from external source and view it as clockwise direction (CW). EB is dedicated to count the pulses from external source and view it as counterclockwise direction (CCW). Simply put, EA counts up and EB counts down. User can decide the normal high or normal 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)

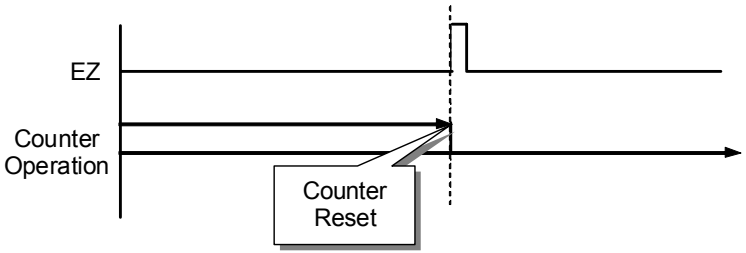
The index input (EZ) signals of the encoders are used as the "ZERO" reference. This signal is common on most rotational motors. EZ can be used to define the absolute position of the mechanism. The input logic polarity of the EZ signals is programmable with ADLINK library.

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

In DB-8150, there are dual EZ signals were received from PCI-8154/58 ONLY.

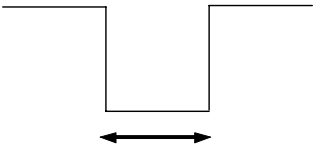
4.1.3 Homing with EZ Signal

The DB-8150 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 0.2 us to 6.55 ms. The maximum frequency is up to 1 MHz.



Pulse Width

4.1.5 Linear Function

Linear function is used to generate a new comparing position point by a fixed incremental value linear function, D=D' + A. D means a calculated comparing position, the linear data. D' means a previous comparing position. Every time the position was compared, hence a new data is calculated by adding 'A', the fixed incremental value. User can implement a continuous triggering function promptly by using linear function trigger.

4.1.6 FIFO function

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

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.

4.1.8 Mapping

Mapping means eight trigger signals are not one-to-one mapping to three comparators. Trigger output channels are able to , 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 Modular Board Operation

The DB-8150 provides eight trigger pulse – TRGx (0-7) which be calculated by two major comparing methods – FIFO and Linear function. This section describes the operation theory for users to know more about DB-8150.

4.2.1 Position Comparison

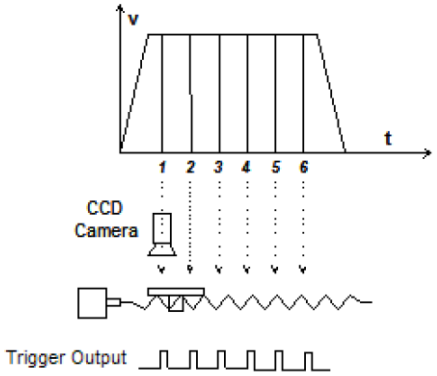
The DB-8150 provides position comparison functions for each channel. Once the counter reaches a preset value set by the user, the DB-8150 will generate the trigger pulse. TRGx pins are for trigger pulse output channels which were designed by TTL differential output type. Therefore, the differential signal is able to reduce noise-effect efficiently while trigger pulse signal transmitting.

The comparing method is “equal”. Consequently, when the counter value is exactly equal to the pre-set 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 DB-8150. An image of the moving object is easily obtained.

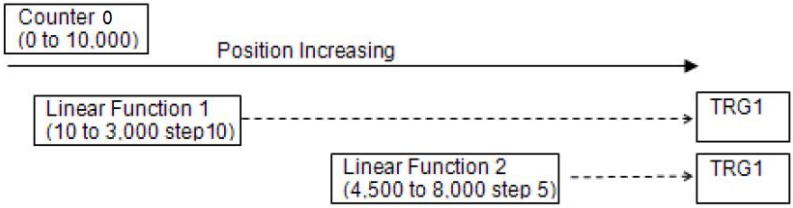


4.2.2 Position Latch

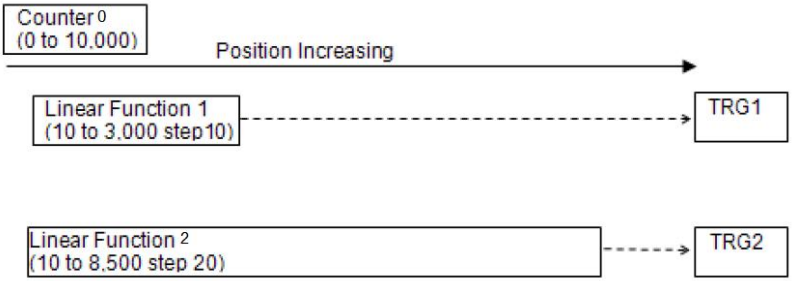
The position latch function is fulfilled by DI signal. Once the DI signal is active, the counter value of its latch channel will be saved to latched register at the same time. User can read the latch register any time.

4.2.3 Linear Function

There are 2 linear functions in DB-8150 which supports to each PWM channel individually and each function is independent. It allows that linear function range overlapped is possible when system operating. Each linear function has its own comparator and the comparator could be linked to any one of two counters. By this features, users can produce many kinds of trigger modes. Please see the following diagrams:



Take two linear functions. Set the trigger interval and range as shown in the diagram. Set these two linear functions to counter0 and also comparator0. Set two linear functions to same trigger output pin TRG1. 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. From TRG1 pins, the trigger pulse output will be observed which was generated by linear function 1 and linear function 2 sequentially.



For other case, 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. This diagram is also able to present the overlap status of TRG1 output if makes TRG 2 instead of TRG1.

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. In DB-8150, there has only one FIFO and it has 2 million random comparing data space. The FIFO just supports to PWM0 ONLY. If users have more random data than 2 million, try to polling FIFO empty status or wait interrupt event and refill reset random data.

4.2.5 Timer function

There are 2 timers on DB-8150. 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 100ns and the maximum counter value is 30-bit. User can set the timer interval from 100ns to 26ms.

4.3 Multiple Boards Operation

The maximum installation in one PC system is up to 16 pieces.

This means user can have 32 channels for triggering purpose. Since the DB-8150 is integrated with PCI-81454/58 which 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 on PCI-8154/58. By software setting, the DB-8150 card ID is same with carrier-"PCI-8154/58". With this benefit, users can control each board accordingly without any confusion. For example, if three DB-8150 cards are used, then the corresponding channel number on each card will be:

	CH No.	CH1	CH2
Card ID			
1		0	1
2		0	1
3		0	1

4.4 Interrupt Control

The DB-8150 can generate an INT signal to the host PC. Users have to enable the interrupt service with this function call, **`_8154_db50_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 DB-8150 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 empty:

   When the FIFO is becoming empty.

3. FIFO low:

   When the FIFO status is becoming low level. The low level could be set by function, _8154_db50_set_low_fifo_level().

4. FIFO full:

   When the FIFO is becoming full.

5. DI:

   The factor means is able to trigger two different events. One of all events is purely DI signal was occurred. Other event is combining with latch counter function usage, in the first instance user MUST enable the counter latch function, _8154_db50_set_di_latch(). Therefore, when DI interrupt occurred, the counter value was latched simultaneously. When the latch counter event is occurred, an interrupt will be triggered. User can check the latched counter values in related ISR while DI interrupt event coming.

6. EZ clear

The factor means is able to trigger two different events. One of all events is purely EZ signal was occurred. Other event is combining with counter clear function usage, in the first instance user MUST enable the counter clear function, _8154_db50_set_ez_clear(). Therefore, when EZ interrupt occurred, the counter value was cleared simultaneously. When the clear counter event is occurred, an interrupt will be triggered.

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.

Bit	0	1	2	3	4	5	6	7
Factor	PWM0	PWM1	Linear function 0 finished	Linear function 1 finished	DI0	DI1	EZ0	EZ1
Bit	8	9	10	11	12	13	14	15
Factor	FIFO_empty	FIFO_low	FIFO_full	-	-	-	-	-

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

Use wait function we provided to wait interrupts under Windows

To detect an interrupt signal from the DB-8150 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 function

   ```
   "_8154_db50_set_int_control( I16 CardId, I16
        intFlag )"
        intFlag:
        1: Enable
        0: Disable
   ```

2. Enable int_factor you want to wait

   ```
   "_8154_db50_set_int_factor( I16 CardId, I16
        IntFactorsInBit )"
   ```

3. Using wait function to wait the specified interrupt.

   ```
   "_8154_db50_wait_single_int( I16 CardId, I16
        FactorBitNum, I32 TimeOutMs )"
   ```

4. In the end of interrupt process, you should reset and disable the int_factor and interrupt function.

   ```
   "_8154_db50_reset_int( I16 CardId, I16
        FactorBitNum )"
   "_8154_db50_set_int_control( I16 CardId, I16
        intFlag )"
   ```

Example:

```
_8154_db50_int_control(CardId,1) //Enable the
     interrupt service
_8154_db50_set_int_factor(CardId, 0x01) // PWM0

<-----(PWM0 operation)----->

//Wait PWM0 interrupt signal
//FactorBitNum=0
//10 seconds time-out

_8154_db50_wait_single_int(CardId,0,10000)
          …

_8154_db50_reset_int(CardId,0x01) //reset the
     int_factor
_8154_db50_int_control(CardId,0) //Disable the
     interrupt service
```

5 DB-8150 Utility

After installing the hardware, it is necessary to correctly configure all cards and double check the system before running. This chapter gives guidelines for establishing a control system and manually testing the DB-8150 card to verify correct operation. The Utility for DB-8150 software provides a simple yet powerful means to setup, configure, test and debug a trigger control system.

Note that DB-8150 Utility is only available for Windows 2000/XP with a screen resolution higher than 1024x768. Recommended screen resolution is 1024x768. It cannot be executed under the DOS environment.

5.1 Execute the DB-8150 Utility

After installing the software drivers for the DB-8150, the Utility program can be located at <chosen path>**ADLINK\PCI-8154\Utility**. To execute the program, double click on the executable file or use **Start>ProgramFiles>ADLINK>PCI-8154>Utility**.

5.2 Interface Overview

5.2.1 Main Menu

The overview of Utility for DB-8150 is showed below.

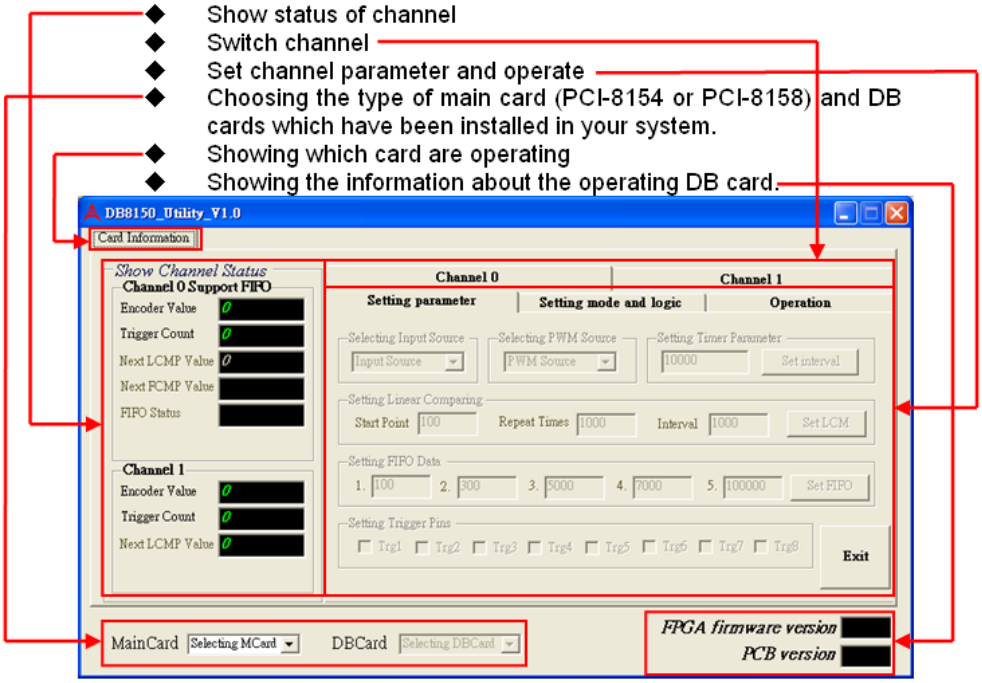


Figure 5-1: Interface Overview

Note: In this Utility, when the color of the label is gray, it means disable. The next page describes operation steps approximately.

5.2.2 Operation Steps

1. Choose Main Card and DB Card. If no Main Card and DB Card be chosen, all others item wouldn't operation. After this, the select input source item would be enabled. Note: if the Timer source is selected, the setting timer parameter would be enabled.
2. Select input source. After this, the selecting PWM source item would be enabled .In this, the Timer0 source is chosen.
3. Select PWM source. There are three (if Timer source is chosen, or only two items in selecting PWM source comboBox) items in this comboBox. When Linear Comparing is chosen, the Setting Linear Comparing item would be enabled or FIFO Comparing is chosen, the Setting FIFO Data item would be enabled. In this case, the Linear Comparing is selected.
4. Press SetLCM button and choose Trigger Pins. Next, click the Setting mode and logic menu to switch to the Setting mode and logic mode.

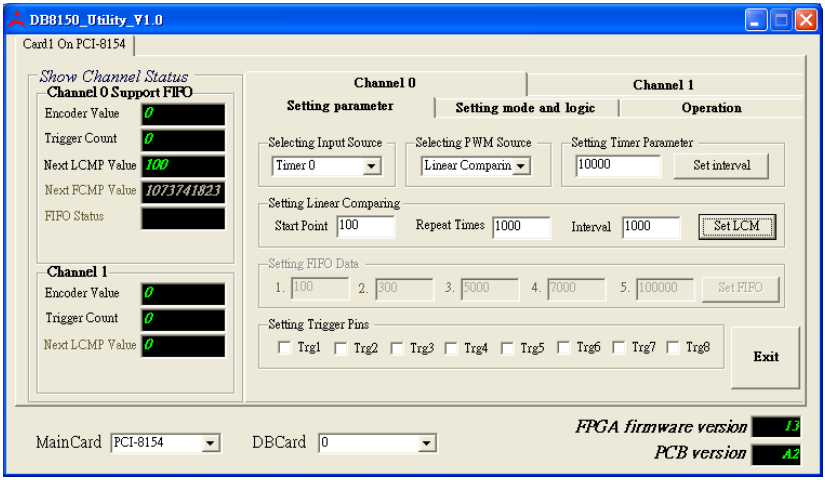


Figure 5-2: Parameter Settings

5. Set the parameter for operation. Then, click the operation menu to switch to the Operation mode. See Figure 5-3 and Figure 5-4.
6. Click the StartTimer button to start. If any interrupt source was chosen, the items of Time out for Interrupt and Interrupt Status would be enabled. If the PWM0 interrupt source is chosen, the color of PWM0 in Interrupt Status item would change (color red: time out, color green: receive the interrupt signal within time out, color yellow: still waiting). See Figure 5-5.

Note: Change the Main Card, DB Card, Channel anytime, and the status can be recorded automatically for recovery when switching back.

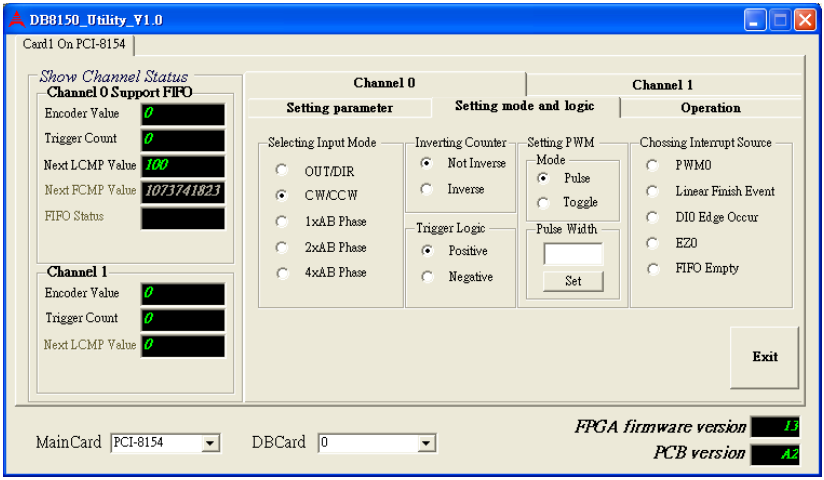


Figure 5-3: Setting Mode and Logic

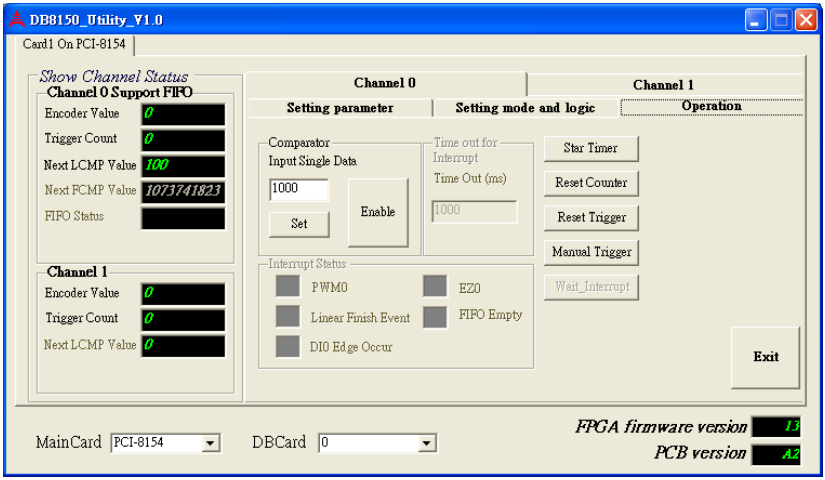


Figure 5-4: Operation

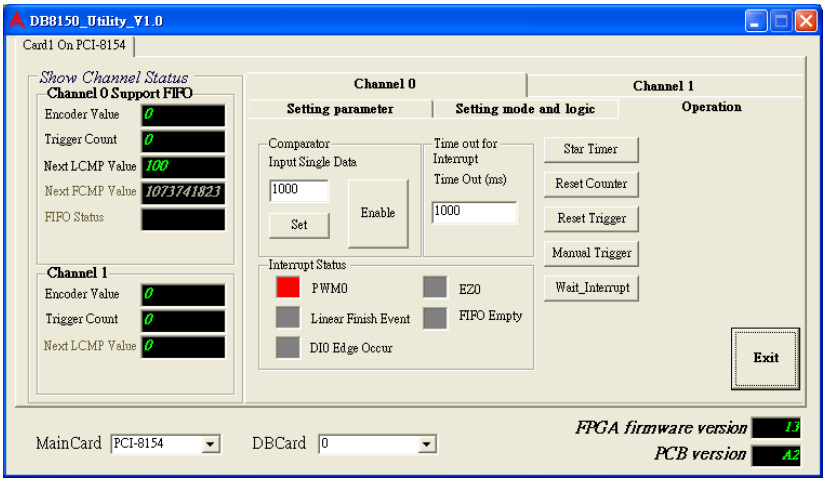


Figure 5-5: Example of Interrupt Operation

5.2.3 Show Channel Status Item

The show channel status item can observe the encoder value, trigger count, next LCMP value, next FCMP value, and FIFO status. Notes: Only channel 0 supports the FIFO function. The detail for each part as below:

1. Encoder Value: Get the encoder's (counter's) value. The related function is **`_8154_db50_get_encoder_value`**
2. Trigger Count: Get trigger number which describes the trigger pulse number generated by all trigger sources. The related function is **`_8154_db50_get_trigger_num`**.
3. Next LCMP Value: Get next compared value in linear comparator. The related function is **`_8154_db50_get_linear_compare_data`**.
4. Next CMP Value: Get next compared value in the comparator. The related function is **`_8154_db50_get_comparator_data`**. Notes: The default value is 1073741823.
5. FIFO Status: Get the current FIFO status. The related function is **`_8154_db50_get_fifo_sts`**.The meaning of each value shows below:
   - ▷ Number 2: FIFO array is equal or small present level.
   - ▷ Number 3: FIFO array is Empty or equal or smaller present level

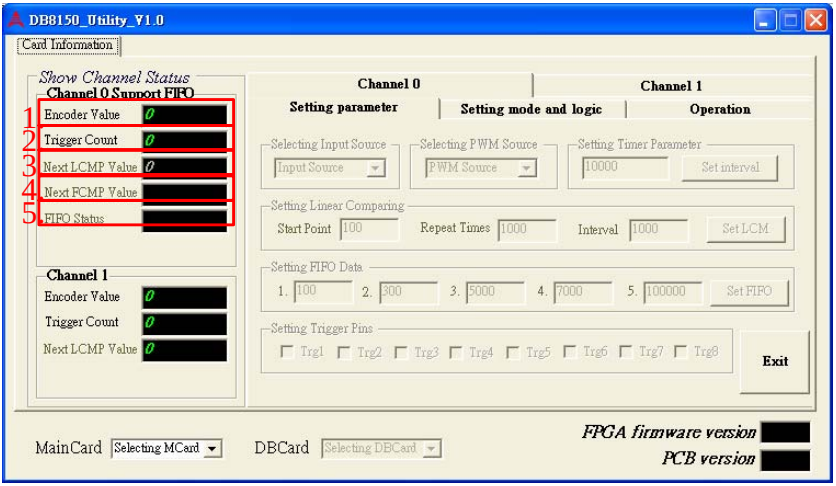


Figure 5-6: Show Channel Status Item

5.2.4 Setting Parameter Menu

In this menu, users can configure input source, PWM source, timer interval, linear comparing, FIFO data, and trigger out.

1. Selecting Input Source: The input source include EA/EB0 - 3 and Timer. The related function call is **_8154_db50_set_counter_source**.
2. Selecting PWM Source. The PWM source includes FIFO comparing, linear comparing and Timer. Notes: The FIFO comparing only supports channel 0. The related function call is **_8154_db50_set_pwm_source**.
3. Setting Timer Parameter: Set the timer interval and enable the timer. The actual time interval can be computed with bellow equation. Timer tick interval = Interval x 16 + 4) x 40 (ns). The related function call is **_8154_db50_set_Timer**.
4. Setting Linear Comparing: Set and enabled the linear comparing function including starting point (Start Point), repeat times (Repeat Times) and trigger interval (Interval). The related function call is **_8154_db50_set_linear_compare**.
5. Setting FIFO Data: Set five data into the FIFO register. The related function call is **_8154_db50_set_fifo_array**.
6. Setting Trigger Pins: Choose “TrgOut” pins to trigger. The related function call is **_8154_db50_set_trigger_source**.

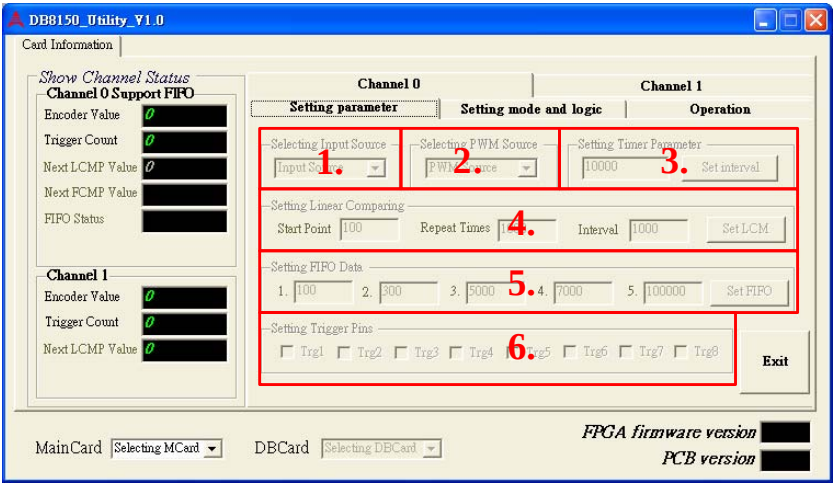


Figure 5-7: Setting Parameter Menu

5.2.5 Setting Mode and Logic Menu

In this menu, users can configure input mode, counter, trigger logic, PWM, interrupt source.

1. Selecting Input Mode: Set the pulse input decoded mode including OUT/DIR, CW/CCW, 1xAB phase, 2xAB phase and 4xAb phase. The related function is **_8154_db50_set_encoder_input_mode**.
2. Invert Counter: Set count up or count down for counter. The related function is **_8154_db50_set_encoder_up_down_count**.
3. Trigger logic: Set the logic of TRG-OUT. The related function is **_8154_db50_set_trgOut_logic**.
4. Setting PWM: Set the PWM mode and its width. The related function is **_8154_db50_set_pwm_mode** and **_8154_db50_set_pwm_pulse_width**.
5. Choosing Interrupt Source: Set the interrupt source. In this Utility, only one interrupt source can be chosen, actually all the interrupt sources can be enabled at a time.

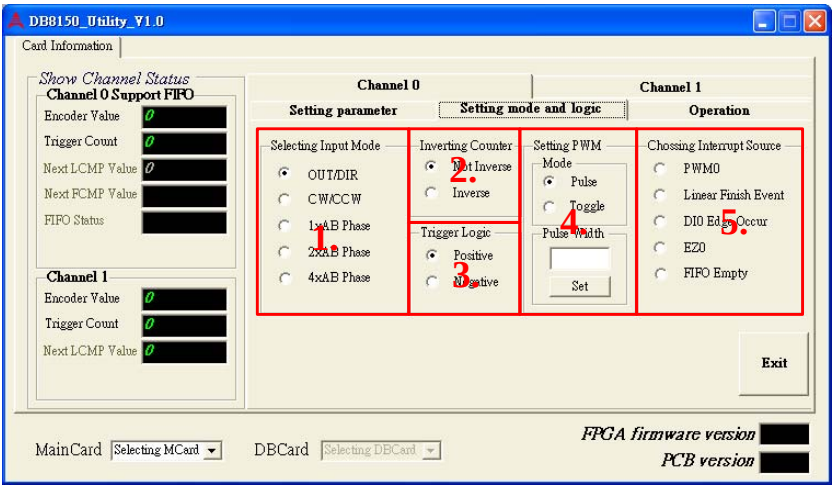


Figure 5-8: Setting Mode and Logic Menu

5.2.6 Operation Menu

In this menu, users can operate the trigger system including start timer, reset counter, manual trigger, set one data into the comparator and set time out for interrupt. In addition, users can observe the interrupt status in this menu.

1. Comparator: Set one data into the comparator and enabled it. The related function calls are **_8154_db50_set_comparator_data** and **_8154_db50_enable_comparator**.
2. Time out for interrupt: Set time out for waiting the interrupt signal which you chosen. The related function call is **_8154_db50_wait_single_int**.
3. Interrupt status: The color of square can be changed to show the status of interrupt which you chosen. The green color, red color describe receiving the interrupt and not receiving the interrupt respectively within time out. And, the yellow color means waiting.
4. Buttons:
   - ▷ Start Timer: Start/Stop the timer. The related function call is **_8154_db50_start_timer**. Note: if the timer wasn't be chosen, this button is disble.
   - ▷ Reset Counter: Reset the encoder value. The related function call is **_8154_db50_set_encoder_value**.
   - ▷ Reset Trigger: Reset the trigger value. The related function call is **_8154_db50_reset_trigger_num**.
   - ▷ Manual Trigger: Trigger the PWM pulse by manual operation. The related function call is **_8154_db50_manual_trigger**.
   - ▷ Wait_Interrupt: Start to wait the interrupt single. The related function call is **_8154_db50_wait_single_int**. Note: if no one interrupt was chosen, this button is disabled.

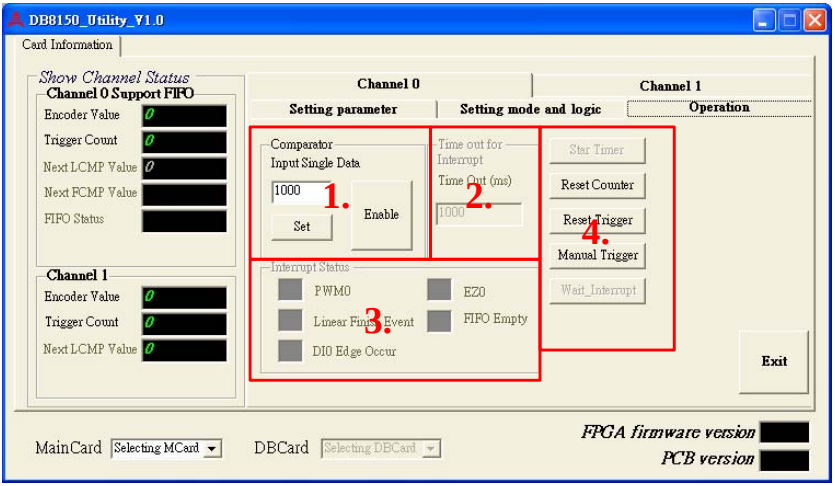


Figure 5-9: Operation Menu

6 Function Library

This chapter describes the supporting software for DB-8150. Users can use these functions to develop programs in C, C++, or Visual Basic.

6.1 List of Functions

This section details all the functions. The function prototypes and common data types are declared in **8154_db8150.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 6-1: Data Type Definitions

The functions of the DB-8150's software drivers use full-names to represent the functions real meaning. All function calls have the same **_8154_db50_** prefix when using the PCI-8154 as the carrier. If the PCI-8158 is selected as the primary motion controller, all functions will carry the **_8158_db50_** prefix. The **_8154_db50_** prefix is used below for use with the PCI-8154. If functions belong to a system level purpose, the convention will be as follows.

In a 'C' programming environment:

```
_{ 8154_db50}_{action_name (lower case)} e.g.
     _8154_db50_initial ()
```

In order to recognize the difference between a C library and a VB library, a capital "B" is placed at he beginnig of each function name e.g. **B_8154_db50_initial** ().

System: Section 6.2

Function Name	Description
_8154_db50_initial	DB-8150 Cards Initialization
_8154_db50_close	Release the resources of DB-8150 Cards
_8154_db50_get_DBfpga_version	Get the hardware's fpga firmware version
_8154_db50_get_PCB_version	Get the hardware's PCB version

Encoder & Counter: Section 6.3

Function Name	Description
_8154_db50_set_encoder_input_mode	Set the pulse input decoded mode
_8154_db50_set_encoder_value	Set the encoder's(counter's) value
_8154_db50_get_encoder_value	Get the encoder's(counter's) value
_8154_db50_set_encoder_up_down_count	Set count up or count down for counter
_8154_db50_set_ez_clear	Configure the EZ clear operation
_8154_db50_set_counter_source	Set the source of counter

PWM Configuration: Section 6.4

Function Name	Description
_8154_db50_set_pwm_source	Set the source of PWM
_8154_db50_set_pwm_mode	Set the mode of PWM (Pulse or Toggle output)
_8154_db50_set_pwm_pulse_width	Set the output pulse width

TRG-OUT Configuration: Section 6.5

Function Name	Description
_8154_db50_set_trigger_source	Set the source of TRG-OUT
_8154_db50_set_trgOut_logic	Set the logic of TRG-OUT
_8154_db50_reset_trigger_num	Clear the trigger number
_8154_db50_get_trigger_num	Get trigger number

Comparator: Section 6.6

Function Name	Description
_8154_db50_enable_comparator	Enable/Disable Comparator
_8154_db50_set_comparator_data	Set the comparing data to the comparator
_8154_db50_get_comparator_data	Get current comparing data from the comparator

FIFO Comparing: Section 6.7

Function Name	Description
_8154_db50_reset_fifo	Reset FIFO data
_8154_db50_get_fifo_sts	Get the current FIFO status
_8154_db50_set_fifo_array	Set comparing data array to FIFO
_8154_db50_set_fifo_shift	Shift the FIFO data to comparator
_8154_db50_set_fifo_low_level	Set the FIFO low level
_8154_db50_get_fifo_low_level	Get the FIFO low level

Linear Comparing: Section 6.8

Function Name	Description
_8154_db50_enable_linear_compare	Enable/Disable linear comparing function
_8154_db50_set_linear_compare	Set the linear comparing function
_8154_db50_get_linear_compare_data	Get current data from linear comparator

Manual Trigger: Section 6.9

Function Name	Description
_8154_db50_manual_trigger	Set the trigger output manually

Digital input: Section 6.10

Function Name	Description
_8154_db50_get_di_sts	Get the digital input signal's status
_8154_db50_set_gpio_output	Set output value
_8154_db50_get_gpio_output	Get output value

Latch: Section 6.11

Function Name	Description
_8154_db50_set_di_latch	Configure the di latch function
_8154_db50_get_di_latch_value	Get the di latched value

Timer: Section 6.12

Function Name	Description
_8154_db50_set_Timer	Configure the Timer
_8154_db50_start_timer	Start the timer manually

Interrupt Control : Section 6.13

Function Name	Description
_8154_db50_set_int_control	Enable/Disable INT service
_8154_db50_set_int_factor	Set (Enable) the interrupt factors
_8154_db50_get_int_factor	Get the setting of interrupt factors
_8154_db50_wait_single_int	Wait the single interrupt event
_8154_db50_reset_int	Reset INT event

6.2 System & Initialization

@ Name

_8154_db50_initial	DB-8150 Cards Initialization
_8154_db50_close	Release the resources of DB-8150 Cards
_8154_db50_get_DBfpga_version	Get the hardware's fpga firmware version
_8154_db50_get_PCB_version	Get the hardware's PCB version

@ Description

_8154_db50_initial:

Initialize the hardware and software states of the 8154_DB-8150 card. Before calling this function, users must call _8154_initial first to get resource of DB-8150. Users can check the return code of this function to know if the initialization is successful or not..

_8154_db50_close:

This function is to release the resource occupied by the 8154_DB-8150 card. When terminating the program, do not forget to call this function to release all the resource occupied by 8154_DB-8150 card.

_8154_db50_get_DBfpga_version:

Users can get the hardware's fpga firmware version by this function.

_8154_db50_get_PCB_version:

Users can get the hardware's PCB version by this function.

@ Syntax

C/C++ (Windows 2000/XP)

```
I16 _8154_db50_initial(I16 Card_ID);
I16 _8154_db50_close(I16 Card_ID);
I16 _8154_db50_get_DBfpga_version(I16 CardId, I16
     *Version);
```

```
I16 _8154_db50_get_PCB_version(I16 CardId, I16
     *Version);
```

Visual Basic (Windows 2000/XP)

```
B_8154_db50_initial (ByVal CardId As Integer) As
     Integer
B_8154_db50_close (ByVal CardId As Integer) As
     Integer
B_8154_db50_get_DBfpga_version (ByVal CardId As
     Integer, Version As Integer) As Integer
B_8154_db50_get_PCB_version (ByVal CardId As
     Integer, Version As Integer) As Integer
```

@ Argument

`I16 CardId`: Specify the DB-8150 card index. Normally, the board index sequence would be decided by the system. The index is from 0.

`I16 * Version`: The hardware's fpga firmware/PCB version.

6.3 Encoder & Counter

@ Name

_8154_db50_set_encoder_input_mode	Set the pulse input decoded mode
_8154_db50_set_encoder_value	Set the encoder's(counter's) value
_8154_db50_get_encoder_value	Get the encoder's(counter's) value
_8154_db50_set_encoder_up_down_count	Set count up or count down for counter
_8154_db50_set_ez_clear	Configure the EZ clear operation
_8154_db50_set_counter_source	Set the source of counter

@ Description

_8154_db50_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.

_8154_db50_set_encoder_value:

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

_8154_db50_get_encoder_value:

This function is used to read the current encoder value.

_8154_db50_set_encoder_up_down_count:

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

_8154_db50_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.

_8154_db50_set_counter_source:

This function is used to set the counter source. The source could be on of 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 _8154_db50_set_encoder_input_mode( I16
     CardId, I16 Channel, I16 IptMode );
I16 _8154_db50_set_encoder_value( I16 CardId, I16
     Channel, I32 EncValue );
I16 _8154_db50_get_encoder_value( I16 CardId, I16
     Channel, I32 *EncValue );
I16 _8154_db50_set_encoder_up_down_count(I16
     CardId, I16 Channel, I16 Inverse );
I16 _8154_db50_set_ez_clear( I16 CardId, I16
     Channel, I16 Enable, I16 ClrLogic );
I16 _8154_db50_set_counter_source( I16 CardId,
     I16 Channel, I16 CtnSrc );
```

Visual Basic (Windows 2000/XP)

```
B_8154_db50_set_encoder_input_mode (ByVal CardId
     As Integer, ByVal Channel As Integer, ByVal
     IptMode As Integer) As Integer
B_8154_db50_set_encoder_value (ByVal CardId As
     Integer, ByVal Channel As Integer, ByVal
     EncValue As Long) As Integer
B_8154_db50_get_encoder_value (ByVal CardId As
     Integer, ByVal Channel As Integer, EncValue
     As Long) As Integer
B_8154_db50_set_encoder_up_down_count (ByVal
     CardId As Integer, ByVal Channel As Integer,
     ByVal Inverse As Integer) As Integer
B_8154_db50_set_ez_clear (ByVal CardId As
     Integer, ByVal Channel As Integer, ByVal
     Enable As Integer, ByVal ClrLogic As
     Integer) As Integer
B_8154_db50_set_counter_source (ByVal CardId As
     Integer, ByVal Channel As Integer, ByVal
     CtnSrc As Integer) As Integer
```

@ Argument

I16 CardId: Specify the DB-8150 card index. Normally, the board index sequence would be decided by the system. The index is from 0.

I16 Channel: The specified channel number. (0 - 1)

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 – 5)

- 0:Encoder0(Carrier Board EA/B 0)
- 1:Encoder1(Carrier Board EA/B 1)
- 2:Encoder2(Daughter Board DEA/B 2)
- 3:Encoder3(Daughter Board DEA/B 3)
- 4:Timer0
- 5:Timer1

6.4 PWM Configuration

@ Name

_8154_db50_set_pwm_source	Set the source of PWM
_8154_db50_set_pwm_mode	Set the mode of PWM (Pulse or Toggle output)
_8154_db50_set_pwm_pulse_width	Set the output pulse width

@ Description

_8154_db50_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.

_8154_db50_set_pwm_mode:

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

_8154_db50_set_pwm_pulse_width:

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

@ Syntax

C/C++ (Windows 2000/XP)

```
I16 _8154_db50_set_pwm_source( I16 CardId, I16
     Channel, I16 CmpEn, I16 LinearEn, I16
     TimerEn );
I16 _8154_db50_set_pwm_mode(  I16 CardId, I16
     Channel, I16 PulseOrToggle );
I16 _8154_db50_set_pwm_pulse_width( I16 CardId,
     I16 Channel, I32 WidthPara );
```

Visual Basic (Windows 2000/XP)

```
B_8154_db50_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_8154_db50_set_pwm_mode (ByVal CardId As
      Integer, ByVal Channel As Integer, ByVal
      PulseOrToggle As Integer) As Integer
B_8154_db50_set_pwm_pulse_width (ByVal CardId As
      Integer, ByVal Channel As Integer, ByVal
      WidthPara As Long) As Integer
```

@ Argument

I16 CardId: Specify the DB-8150 card index. Normally, the board index sequence would be decided by the system. The index is from 0.

I16 Channel: The channel's number of PWM. (0 - 1).

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

The default setting of CmpEn is 1.

Note: Cmp only support one channel (channel 0)

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

The default setting of LinearEn is 0.

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

The default setting of TimerEn is 0.

I16 PulseOrToggle: Select the pulse output or level switch output

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

I32 WidthPara: The parameter of pulse width. (1 - 65535)

The default setting of WidthPara is 999. => 100 µs

Pulse width(ns) = (WidthPara × 100) + 85

6.5 TRG-OUT Configuration

@ Name

_8154_db50_set_trigger_source	Set the source of TRG-OUT
_8154_db50_set_trigger_logic	Set the logic of TRG-OUT
_8154_db50_reset_trigger_num	Clear the trigger number
_8154_db50_get_trigger_num	Get trigger number

@ Description

_8154_db50_set_trigger_source:

The DB-8150 has eight “TrgOut” pins to trigger. And DB-8150 has two channel of PWM. By this function, TrgOut maps to PWM channel. They can be “OR” output.

_8154_db50_set_trigger_logic:

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

_8154_db50_reset_trigger_num:

Reset trigger number which output from PWM channel.

_8154_db50_get_trigger_num:

Get trigger number which output from PWM channel.

@ Syntax

C/C++ (Windows 2000/XP)

```
I16 _8154_db50_set_trigger_source( I16 CardId,
     I16 TriggerLine, I16 Channel, I16 Enable);
I16 _8154_db50_set_trigger_logic( I16 CardId, I16
     Channel, I16 Logic );
I16 _8154_db50_reset_trigger_num( I16 CardId, I16
     Channel );
I16 _8154_db50_get_trigger_num( I16 CardId, I16
     Channel, F64 *TriggerNum );
```

Visual Basic (Windows 2000/XP)

```
B_8154_db50_set_trigger_source (ByVal CardId As
      Integer, ByVal TriggerLine As Integer, ByVal
      Channel As Integer, ByVal Enable As Integer)
      As Integer
B_8154_db50_set_trigger_logic (ByVal CardId As
      Integer, ByVal Channel As Integer, ByVal
      Logic As Integer) As Integer
B_8154_db50_reset_trigger_num (ByVal CardId As
      Integer, ByVal Channel As Integer) As
      Integer
B_8154_db50_get_trigger_num (ByVal CardId As
      Integer, ByVal Channel As Integer,
      TriggerNum As Long) As Integer
```

@ Argument

I16 CardId: Specify the DB-8150 card index. Normally, the board index sequence would be decided by the system. The index is from 0.

I16 Channel: The channel's number of PWM. (0 ~ 1).

I16 TriggerLine: The TRG-OUT pins. (0 ~ 7)

I16 Enable: Map the TRG-OUT to the PWM channel

- 0: Disable mapping between TRG-OUT and PWM channel.
- 1: Enable mapping between TRG-OUT and PWM channel.

I16 Logic: The logic of TrgOut pin.

- 0: Not inverse
- 1: Inverse

F64 *TriggerNum: Trigger numbers from PWM channel.

6.6 Comparator

@ Name

_8154_db50_enable_comparator	Enable/Disable Comparator
_8154_db50_set_comparator_data	Set the comparing data to the comparator
_8154_db50_get_comparator_data	Get current comparing data from the comparator

@ Description

_8154_db50_enable_comparator:

This function is used to enable/disable comparator..

_8154_db50_set_comparator_data:

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

_8154_db50_get_comparator_data:

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

@ Syntax

C/C++ (Windows 2000/XP)

```
I16 _8154_db50_enable_comparator( I16 CardId, I16
     Enable );
I16 _8154_db50_set_comparator_data( I16 CardId,
     I32 CmpData );
I16 _8154_db50_get_comparator_data( I16 CardId,
     I32 *CmpData );
```

Visual Basic (Windows 2000/XP)

```
B_8154_db50_enable_comparator(ByVal CardId As
     Integer, ByVal Enable As Integer) As Integer
B_8154_db50_set_comparator_data (ByVal CardId As
     Integer, ByVal CmpData As Long) As Integer
B_8154_db50_get_comparator_data (ByVal CardId As
     Integer, CmpData As Long) As Integer
```

@ Argument

I16 CardId: Specify the DB-8150 card index. Normally, the board index sequence would be decided by the system. The index is from 0.

I16 Enable: Enable/Disable comparator

- 0: Disable (default)
- 1: Enable

I32 CmpData: The current comparing data in comparator.

The default setting of CmpData is 0x3fffffff.

Note: CMP only support one channel (channel 0).

6.7 FIFO Comparing

@ Name

_8154_db50_reset_fifo	Reset FIFO data
_8154_db50_get_fifo_status	Get the current FIFO status
_8154_db50_set_fifo_array	Set comparing data array to FIFO
_8154_db50_set_fifo_shift	Shift the FIFO data to comparator
_8154_db50_set_low_fifo_level	Set the FIFO low level
_8154_db50_get_low_fifo_level	Get the FIFO low level

@ Description

`_8154_db50_reset_fifo`:

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

`_8154_db50_get_fifo_status`:

This function is used to get the current FIFO status. There are three bits to check FIFO status on bit 0~2. The capacity of FIFO totally is 2097151.

Bit 0 is FIFO Empty Status.

- 1: indicate the FIFO is empty.
- 0: indicate the FIFO is not empty.

Bit 1 is FIFO Low Status. (0, 1/4, 2/4, 3/4)

- 1: indicate the data number in the FIFO is equal or smaller than the preset level.
- 0: indicate the data number in the FIFO is greater than the preset level.

Bit 2 is FIFO Full Status.

- 1: indicate the FIFO is full.
- 0: indicate the FIFO is not full.

_8154_db50_set_fifo_array:

This function is used to set comparing data array to the FIFO. The capacity of FIFO is 2097151. When the status of FIFO is full, the data cannot be set into FIFO. This function won't check the FIFO status. When using this function, you should also enable comparator by "**_8154_db50_enable_comparator**".

_8154_db50_set_fifo_shift:

This function is used to shift the FIFO data to comparator (CMP) manually.

_8154_db50_set_low_fifo_level:

This function is used to set the FIFO low level. There are four conditions in FIFO low level selection.

_8154_db50_get_low_fifo_level:

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

@ Syntax

C/C++ (Windows 2000/XP)

```
I16 _8154_db50_reset_fifo( I16 CardId );
I16 _8154_db50_get_fifo_status( I16 CardId, I16
     *FifoSts );
I16 _8154_db50_set_fifo_array( I16 CardId, I16
     ArraySize ,I32 *DataArr, I16 ShiftFlag );
I16 _8154_db50_set_fifo_shift( I16 CardId );
I16 _8154_db50_set_fifo_low_level( I16 CardId,
     I16 Level );
I16 _8154_db50_get_fifo_low_level( I16 CardId,
     I16 *Level );
```

Visual Basic (Windows 2000/XP)

```
B_8154_db50_reset_fifo (ByVal CardId As Integer)
     As Integer
B_8154_db50_get_fifo_status (ByVal CardId As
     Integer, FifoSts As Integer) As Integer
B_8154_db50_set_fifo_array (ByVal CardId As
     Integer, ByVal ArraySize As Integer, DataArr
     As Long, ByVal ShiftFlag As Integer ) As
     Integer
B_8154_db50_set_fifo_shift (ByVal CardId As
     Integer ) As Integer
B_8154_db50_set_fifo_low_level (ByVal CardId As
     Integer, ByVal Level As Integer) As Integer
B_8154_db50_get_fifo_low_level (ByVal CardId As
     Integer, Level As Integer) As Integer
```

@ Argument

`I16 CardId`: Specify the DB-8150 card index. Normally, the board index sequence would be decided by the system. The index is from 0.

Note: The FIFO CMP only supports one channel. (Channel 0)

`I16 *FifoSts`: The FIFO status.

Bit 0 is FIFO Empty Status.

- 1: indicate the FIFO is empty.
- 0: indicate the FIFO is not empty.

Bit 1 is FIFO Low Status. (0, 1/4, 2/4, 3/4)

- 1: indicate the data number in the FIFO is equal or smaller than the preset level.
- 0: indicate the data number in the FIFO is greater than the preset level.

Bit 2 is FIFO Full Status.

- 1: indicate the FIFO is full.
- 0: indicate the FIFO is not full.

`I16 ArraySize`: The size of data array to FIFO. (1 – 2097151)

`I32 *DataArr`: The data array to FIFO.

I16 ShiftFlag: Determine if shift the FIFO data to comparator.

- 0: Don't auto shift. User wants to manually call **_8154_db50_set_fifo_shift** to shift data.
- 1: Auto shfit a FIFO data to comparator

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

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

6.8 Linear Comparing

@ Name

_8154_db50_enable_linear_compare	Enable/Disable Linear compare
_8154_db50_set_linear_compare	Set the linear comparing function
_8154_db50_get_linear_compare_data	Get current data form linear comparator

@ Description

_8154_db50_enable_linear_compare:

This function is used to enable/disable linear compare.

_8154_db50_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 "**_8154_db50_set_pwm_source**". DB-8150 has 2 linear sets(channel) can be used for linear compare. Every set(channel) 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 index, (N= 1, 2…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 only be compared with StartPoint.

_8154_db50_get_linear_compare_data:

This function is used to get current data form linear comparator.

@ Syntax

C/C++ (Windows 2000/XP)

```
I16 _8154_db50_enable_linear_compare( I16 CardId,
      I16 Channel, I16 Enable);
I16 _8154_db50_set_linear_compare( I16 CardId,
      I16 Channel, I32 StartPoint, F64
      RepeatTimes, I16 Interval );
I16 _8154_db50_get_linear_compare_data( I16
      CardId, I16 Channel, F64 *CurrentData);
```

Visual Basic (Windows 2000/XP)

```
B_8154_db50_enable_linear_compare (ByVal CardId
      As Integer, ByVal Channel As Integer, ByVal
      Enable As Integer) As Integer
B_8154_db50_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_8154_db50_ get_linear_compare_data (ByVal
      CardId As Integer, ByVal Channel As Integer,
      CurrentData As Double) As Integer
```

@ Argument

I16 CardId: Specify the DB-8150 card index. Normally, the board index sequence would be decided by the system. The index is from 0.

I16 Channel: The channel of linear set. (0-1)

I16 Enable: Enable/Disable linear compare

- 0: Disable (default)
- 1: Enable

I32 StartPoint: The first trigger point.

F64 RepeatTimes: The trigger repeat times. (32 bits integer) (1 - 4294967295)

I16 Interval: The linear interval. (1 to 32767, -1 to -32767)

F64 * CurrentData: The current data form linear comparator.

6.9 Manual Trigger

@ Name

_8154_db50_manual_trigger	Set the trigger output manually

@ Description

_8154_db50_manual_trigger:

This function is used to generate one pulse manually.

@ Syntax

C/C++ (Windows 2000/XP)

```
I16 _8154_db50_manual_trigger( I16 CardId, I16
     Channel );
```

Visual Basic (Windows 2000/XP)

```
B_8154_db50_manual_trigger (ByVal CardId As
     Integer, ByVal Channel As Integer) As
     Integer
```

@ Argument

I16 CardId: Specify the DB-8150 card index. Normally, the board index sequence would be decided by the system. The index is from 0.

I16 Channel: The channel's number of PWM. (0 - 1).

6.10 Digital Input

@ Name

_8154_db50_get_di_sts	Get the digital input signal's status
_8154_db50_set_gpio_output	Set the digital otput signal's status
_8154_db50_get_piio_output	Get the digital otput signal's status

@ Description

`_8154_db50_get_di_sts`:

This function is used to get the status of digital inputs. The digital inputs include Di and EZ signal. The parameters are showed as bit format.

`_8154_db50_set_gpio_output`:

This function is used to set the status of digital output.

`_8154_db50_get_gpio_output`:

This function is used to get the status of digital output.

@ Syntax

C/C++ (Windows 2000/XP)

```
I16 _8154_db50_get_di_sts( I16 CardId, I16
     *DiStsInBit, I16 *EzStsInBit );
I16 _8154_db50_set_gpio_output( I16 CardId, I16
     DoValue );
I16 _8154_db50_get_gpio_output( I16 CardId, I16 *
     DoValue );
```

Visual Basic (Windows 2000/XP)

```
B_8154_db50_get_di_sts (ByVal CardId As Integer,
     DiStsInBit As Integer, EzStsInBit As
     Integer) As Integer
B_8154_db50_set_gpio_output (ByVal CardId As
     Integer, ByVal DoValue As Integer) As
     Integer
B_8154_db50_get_gpio_output (ByVal CardId As
     Integer, DoValue As Intege) As Integer
```

@ Argument

I16 CardId: Specify the DB-8150 card index. Normally, the board index sequence would be decided by the system. The index is from 0.

I16 * DiStsInBit: The Di status in bit format.

- bit 0: DI0 Stauts
- bit 1: DI1 Status

I16 *EzStsInBit: The EZ status in bit format.

- bit 0: Ez0 Status
- bit 1: Ez1 Status

I16 DoValue: Set the status of digital output.

- 0: DO OFF
- 1: DO ON

I16 *DoValue: Get the status of digital output.

6.11 Latch

@ Name

_8154_db50_set_di_latch	Configure the di latch function
_8154_db50_get_dI_latch_value	Get the di latched value

@ Description

`_8154_db50_set_di_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.

`_8154_db50_get_di_latch_value`:

Use this function get the latched counter value.

@ Syntax

C/C++ (Windows 2000/XP)

```
I16 _8154_db50_set_di_latch( I16 CardId, I16
     Channel, I16 Enable, I16 EdgeSel );
I16 _8154_db50_get_di_latch_value(I16 CardId, I16
     Channel, I32 *LatchValue );
```

Visual Basic (Windows 2000/XP)

```
B_8154_db50_set_di_latch (ByVal CardId As
     Integer, ByVal Channel As Integer, ByVal
     Enable As Integer, ByVal EdgeSel As Integer)
     As Integer
B_8154_db50_get_di_latch_value (ByVal CardId As
     Integer, ByVal Channel As Integer,
     LatchValue As Long) As Integer
```

@ Argument

I16 CardId: Specify the DB-8150 card index. Normally, the board index sequence would be decided by the system. The index is from 0.

I16 Channel: The channel's number. (0 – 1)

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

I16 EdgeSel: Latching condition

- 0: DI falling edge to latch (default)
- 1: DI Rising edge to latch

I32 *LatchValue: The Latched counter value.

6.12 Timer

@ Name

_8154_db50_set_Timer	Configure the Timer
_8154_db50_start_timer	Start the timer manually

@ Description

_8154_db50_set_Timer:

This function is used to configure the timer. The timer can be start by software function. If the timer was triggered, users can stop the timer by issue the function "**_8154_db50_start_timer**". *Note: Set "CW/CCW" as encoder input mode by calling function "**_8154_db50_set_encoder_input_mode**" when using timer.

_8154_db50_start_timer:

This function is used to start/stop timer manually.

@ Syntax

C/C++ (Windows 2000/XP)

```
I16 _8154_db50_set_Timer( I16 CardId, I16
     Channel, I16 Interval );
I16 _8154_db50_start_timer( I16 CardId, I16
     Channel, I16 Start );
```

Visual Basic (Windows 2000/XP)

```
B_8154_db50_set_Timer (ByVal CardId As Integer,
     ByVal Channel As Integer, ByVal Interval As
     Integer) As Integer
B_8154_db50_start_timer (ByVal CardId As Integer,
     ByVal Channel As Integer, ByVal Start As
     Integer) As Integer
```

@ Argument

`I16 CardId`: Specify the DB-8150 card index. Normally, the board index sequence would be decided by the system. The index is from 0.

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

`I16 Interval`: Timer interval (0 –1073741823) (default is 0)

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

`I16 Start`: Start timer

- 0: Stop timer
- 1: Start timer

6.13 Interrupt

@ Name

_8154_db50_set_int_control	Enable/Disable INT service
_8154_db50_set_int_factor	Set (Enable) the interrupt factors
_8154_db50_get_int_factor	Get the setting of interrupt factors
_8154_db50_wait_single_int	Wait the single interrupt event
_8154_db50_reset_int	Reset INT event

@ Description

`_8154_db50_set_int_control`:

This function is used to enable the windows interrupt event

`_8154_db50_set_int_factor`:

This function is used to enable/disable the interrupt source. Users can use the function, "**`_8154_db50_wait_single_int`**" to wait the specified interrupt. The following table shows the architecture of interrupt factor.

Bit	0	1	2	3	4	5	6	7
Factor	PWM0	PWM1	L0fin	L1fin	DI0	DI1	EZ0	EZ1
Bit	8	9	10	11	12	13	14	15
Factor	FIFO_empty	FIFO_low	FIFO_full	-	-	-	-	-

(-) : reserved bit (set to 0)
PWM0: PWM0 Trigger Out Event
PWM1: PWM1 Trigger Out Event
L0fin: LinearFunction0 Finish Event
L1fin: LinearFunction1 Finish Event
DI0: DI0 Edge Occur
DI0: DI1 Edge Occur
EZ0: EZ0 Edge Occur
EZ1: EZ1 Edge Occur
FIFO_empty: FIFO Empty event
FIFO_low: FIFO Low event
FIFO_full: FIFO Full event

_8154_db50_get_int_factor:

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

_8154_db50_wait_single_int:

When the user enabled the interrupt function for specified factors by "**_8154_db50_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.

_8154_db50_reset_int:

Reset an interrupt event for specified factors.

@ Syntax

C/C++ (Windows 2000/XP)

```
I16 _8154_db50_set_int_control( I16 CardId, I16
      IntFlag );
I16 _8154_db50_set_int_factor( I16 CardId, I16
      IntFactorsInBit );
I16 _8154_db50_get_int_factor( I16 CardId, I16
      *IntFactorsInBit );
I16 _8154_db50_wait_single_int( I16 CardId, I16
      FactorBitNum, I32 TimeOutMs );
I16 _8154_db50_reset_int( I16 CardId, I16
      FactorBitNum );
```

Visual Basic (Windows 2000/XP)

```
B_8154_db50_set_int_control (ByVal CardId As
      Integer, ByVal IntFlag As Integer) As
      Integer
B_8154_db50_set_int_factor (ByVal CardId As
      Integer, ByVal IntFactorsInBit As Integer)
      As Integer
B_8154_db50_get_int_factor (ByVal CardId As
      Integer, IntFactorsInBit As Integer) As
      Integer
B_8154_db50_wait_single_int (ByVal CardId As
      Integer, ByVal FactorBitNum As Integer,
      ByVal TimeOutMs As Long) As Integer
B_8154_db50_reset_int (ByVal CardId As Integer,
      ByVal FactorBitNum As Integer) As Integer
```

@ Argument

I16 CardId: Specify the DB-8150 card index. Normally, the board index sequence would be decided by the system. The index is from 0.

I16 IntFlag: Enable/Disable Interrupt function

I16 IntFactorsInBit: Interrupt event factor setting in bit format.

Bit	0	1	2	3	4	5	6	7
Factor	PWM0	PWM1	L0fin	L1fin	DI0	DI1	EZ0	EZ1
Bit	8	9	10	11	12	13	14	15
Factor	FIFO_empty	FIFO_low	FIFO_full	-	-	-	-	-

I16 FactorBitNum: Specified the bit number of the IntFactorsInBit.

- 0: PWM0
- 1: PWM1
- 2: L0fin
- 3: L1fin
- 4: DI0
- 5: DI1
- 6: EZ0
- 7: EZ1
- 8: FIFO_empty
- 9: FIFO_low
- 10: FIFO_full

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)

6.14 Return Error

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	Describtion
0	No Error, function success
- 10400	Error! Card Number
- 10401	Error! DB-8150 already Initilized
- 10402	Error! DB-8150 not Initialized yet
-10403	Error! Function input parameters are invalid
- 10404	Error! Event already enabled
- 10405	Error! Event not enabled yet
- 10406	Error! INT wait failed