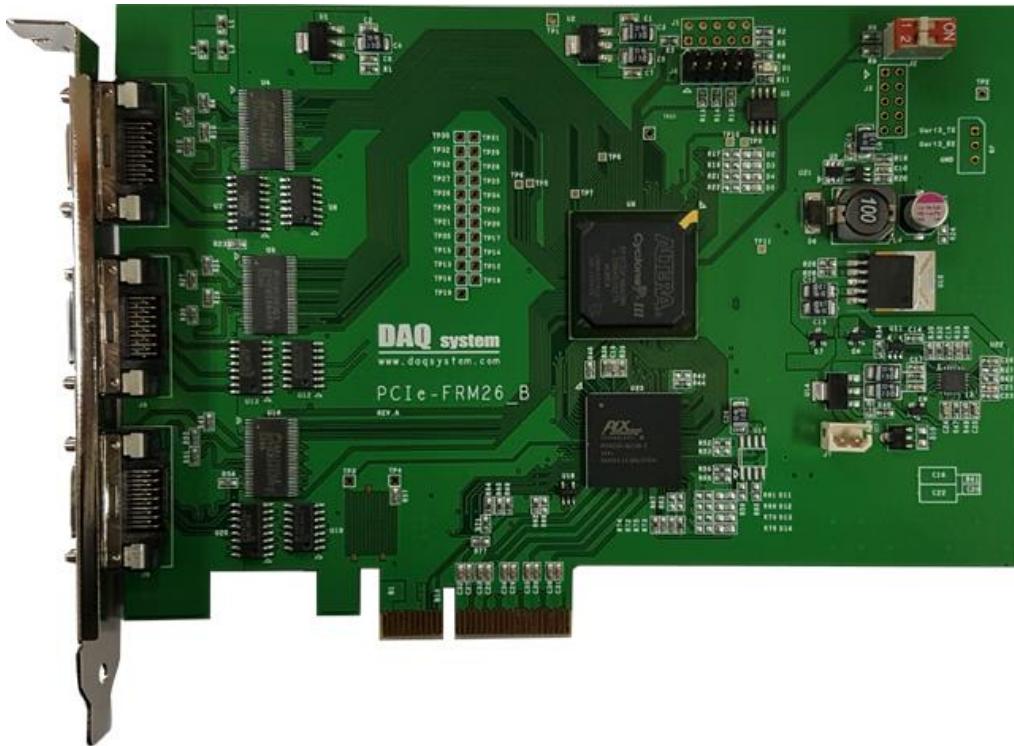


# PCIe-FRM26\_B

## API Manual

Version 1.1



© 2005 DAQ SYSTEM Co., Ltd. All rights reserved.

Microsoft® is a registered trademark; Windows®, Windows NT®, Windows XP®, Windows 7®, Windows 8®, Windows 10®  
All other trademarks or intellectual property mentioned herein belongs to their respective owners.

Information furnished by DAQ SYSTEM is believed to be accurate and reliable. However, no responsibility is assumed by DAQ SYSTEM for its use, nor for any infringements of patents or other rights of third parties which may result from its use. No license is granted by implication or otherwise under any patent or copyrights of DAQ SYSTEM.

The information in this document is subject to change without notice and no part of this document may be copied or reproduced without the prior written consent.

# Contents

## Board Level API Functions

OpenDAQDevice	-----	4
CloseDAQDevice	-----	4

## LVDS(Camera Link) API Functions

LVDS_Init	-----	5
LVDS_Start	-----	6
LVDS_GetFrame	-----	6
LVDS_Close	-----	7
LVDS_Stop	-----	7
LVDS_SetResolution	-----	8
LVDS_GetResolution	-----	8
LVDS_GetFrameSize	-----	9
LVDS_SetDataMode	-----	9
LVDS_GetVersion	-----	10
LVDS_SetDeUse	-----	10
LVDS_CC_Output	-----	11
LVDS_SetTriggerOutput	-----	11
LVDS_SetTrigger1Pulse	-----	12
LVDS_SetTrigger2Pulse	-----	12
LVDS_SetTrigger1Blank	-----	13
LVDS_SetTrigger2Blank	-----	13
LVDS_SetTrigger1Inv	-----	14
LVDS_SetTrigger2Inv	-----	14

## UART API Functions

UART_Init	-----	15
UART_GetData	-----	15
UART_SendData	-----	16
UART_Close	-----	16
UART_SetBaud	-----	17

## DIO(Digital Input Output) API Functions

<b>DIO_Read</b>	-----	<b>18</b>
<b>DIO_Write</b>	-----	<b>18</b>

## Board Level API Functions

### **Overview**

**int            OpenDAQDevice (int nBoard)**  
**BOOL        CloseDAQDevice (int nBoard)**

### **OpenDAQDevice**

It opens a device. You may call this function at the very first time you run the program and some suspicious operation.

**Int            OpenDAQDevice (void)**

**Parameters:**

nBoard : It informs a board number at currently equipped system.  
The board number set up by DIP switch.

**Return Value:**

If the function succeeds, it returns the number of boards which were detected.  
If the function fails, the return value is -1, it means there is no device in the system.  
(In case of multi-board, up to 4 is possible)

### **CloseDAQDevice**

The CloseDAQDevice function closes all opened devices (boards). If use of device is finished, it can certainly close a device for making it other programs so as usable.

**BOOL        CloseDAQDevice (void)**

**Parameters:**

nBoard : It informs a board number at currently equipped system.  
The board number set up by DIP switch.

**Return Value :**

If the function call fails, it returns "FALSE".  
If the function call succeeds, it returns "TRUE".

## LVDS(Camera Link) API Functions

### *Overview*

BOOL	LVDS_Init (int nBoard, int nCh)
BOOL	LVDS_Start (int nBoard, int nCh)
BOOL	LVDS_GetFrame (int nBoard, int nCh, DWORD* nCnt, unsigned char* buf)
BOOL	LVDS_Close (int nBoard, int nCh)
BOOL	LVDS_Stop (int nBoard, int nCh)
BOOL	LVDS_SetResolution (int nBoard, int nCh, DWORD xRes, DWORD yRes)
BOOL	LVDS_GetResolution (int nBoard, int nCh, DWORD *xRes, DWORD *yRes)
BOOL	LVDS_GetFrameSize (int nBoard, int nCh, DWORD *xRes, DWORD *yRes)
BOOL	LVDS_SetDataMode (int nBoard, int nCh, int nMode)
BOOL	LVDS_GetVersion (int nBoard, int nCh, int *nVersion)
BOOL	LVDS_SetDeUse (int nBoard, int nCh, BOOL bUse)
BOOL	LVDS_CC_Output (int nBoard, int nCh, DWORD dwVal)
BOOL	LVDS_SetTriggerOutput (int nBoard, int nCh, DWORD dwValue)
BOOL	LVDS_SetTrigger1Pulse (int nBoard, int nCh, int nPulseCount, int nPulseWidth)
BOOL	LVDS_SetTrigger2Pulse (int nBoard, int nCh, int nPulseCount, int nPulseWidth)
BOOL	LVDS_SetTrigger1Blank (int nBoard, int nCh, int nPulseBlank)
BOOL	LVDS_SetTrigger2Blank (int nBoard, int nCh, int nPulseBlank)
BOOL	LVDS_SetTrigger1Inv (int nBoard, int nCh, BOOL bUse)
BOOL	LVDS_SetTrigger2Inv (int nBoard, int nCh, BOOL bUse)

### **LVDS\_Init**

This function initializes resources used for the LVDS sub-system, for example interrupt and LVDS control register.

**BOOL           LVDS\_Init (int nBoard)**

#### **Parameters:**

nBoard : It informs a board number at currently equipped system.

The board number set up by DIP switch.

nCh : Select a channel. 0 : Channel 0, 1: Channel 1, 2: Channel 2

#### **Return Value:**

If the function call fails, it returns "FALSE".

If the function call succeeds, it returns "TRUE".

## LVDS\_Start

This function starts receiving frame data. After calling this function, you can check whether the data is complete by calling the LVDS\_GetFrame function.

**BOOL            LVDS\_Start (int nBoard, int nCh)**

**Parameters:**

nBoard : It informs a board number at currently equipped system.

The board number set up by DIP switch.

nCh : Select a channel. 0 : Channel 0, 1: Channel 1, 2: Channel 2

**Return Value:**

If the function call fails, it returns "FALSE".

If the function call succeeds, it returns "TRUE".

## LVDS\_GetFrame

This function checks whether the frame data is complete, and if it is, retrieves the frame data. At this time, the size of the buffer to receive data must be informed.

**BOOL            LVDS\_GetFrame (int nBoard, int nCh, DWORD\* nCnt, unsigned char\* buf)**

**Parameters:**

nBoard : It informs a board number at currently equipped system.

The board number set up by DIP switch.

nCh : Select a channel. 0 : Channel 0, 1: Channel 1, 2: Channel 2

\*nCnt : It is the address which contains the number of data to be received in byte size. Specifies the size buffer when the function is called, and read the values of the variables after a call to find out how many actually read. The data size is in bytes.

\*buf : The buffer address.

**Return Value:**

If the function call fails, it returns "FALSE".

If the function call succeeds, it returns "TRUE".

## LVDS\_Close

This function releases all resource were used for LVDS function. The application program calls this function when the program ends.

**BOOL**      **LVDS\_Close (int nBoard, int nCh)**

**Parameters:**

nBoard : It informs a board number at currently equipped system.

The board number set up by DIP switch.

nCh : Select a channel. 0 : Channel 0, 1: Channel 1, 2: Channel 2

**Return Value:**

If the function call fails, it returns "FALSE".

If the function call succeeds, it returns "TRUE".

## LVDS\_Stop

This function stops the frame data capture.

**BOOL**      **LVDS\_Stop (int nBoard, int nCh)**

**Parameters:**

nBoard : It informs a board number at currently equipped system.

The board number set up by DIP switch.

nCh : Select a channel. 0 : Channel 0, 1: Channel 1 , 2: Channel 2

**Return Value:**

If the function call fails, it returns "FALSE".

If the function call succeeds, it returns "TRUE".

## LVDS\_SetResolution

This function selects the resolution of the Video input. Frame size is determined according to this resolution.

**BOOL LVDS\_SetResolution (int nBoard, int nCh, DWORD xRes, DWORD yRes)**

**Parameters:**

nBoard : It informs a board number at currently equipped system.

The board number set up by DIP switch.

nCh : Select a channel. 0 : Channel 0, 1: Channel 1, 2: Channel 2

xRes : Sets the horizontal resolution, that is, the width of the frame.

yRes : Sets the vertical resolution, that is, the height of the frame.

**Return Value:**

If the function call fails, it returns "FALSE".

If the function call succeeds, it returns "TRUE".

## LVDS\_GetResolution

This function gets currently configured camera's frame resolution

**BOOL LVDS\_GetResolution (int nBoard, int nCh, DWORD \*xRes, DWORD \*yRes)**

**Parameters:**

nBoard : It informs a board number at currently equipped system.

The board number set up by DIP switch.

nCh : Select a channel. 0 : Channel 0, 1: Channel 1, 2: Channel 2

\*xRes : Address pointer to receive horizontal Camera resolution

\*yRes : Address pointer to receive vertical Camera resolution

**Return Value:**

If the function call fails, it returns "FALSE".

If the function call succeeds, it returns "TRUE".

## LVDS\_GetFrameSize

This function gets the frame size of the Video input.

**BOOL LVDS\_GetFrameSize (int nBoard, int nCh, DWORD \*xRes, DWORD \*yRes)**

**Parameters:**

nBoard : It informs a board number at currently equipped system.

The board number set up by DIP switch.

nCh : Select a channel. 0 : Channel 0, 1: Channel 1, 2: Channel 2

\*xRes : Horizontal resolution size value.

\*yRes : Size value of vertical resolution.

**Return Value:**

If the function call fails, it returns "FALSE".

If the function call succeeds, it returns "TRUE".

## LVDS\_SetDataMode

This function selects the frame (image) data mode.

**BOOL LVDS\_SetDataMode (int nBoard, int nCh, int nMode)**

**Parameters:**

nBoard : It informs a board number at currently equipped system.

The board number set up by DIP switch.

nCh : Select a channel. 0 : Channel 0, 1: Channel 1, 2: Channel 2

nMode : "2 : 24bit Mode, "Others" : 16bit Mode,

**Return Value:**

If the function call fails, it returns "FALSE".

If the function call succeeds, it returns "TRUE".

## LVDS\_GetVersion

This function gets FPGA version.

**BOOL            LVDS\_GetVersion (int nBoard, int nCh, int \*nVersion)**

**Parameters:**

nBoard : It informs a board number at currently equipped system.

The board number set up by DIP switch.

nCh : Select a channel. 0 : Channel 0, 1: Channel 1, 2: Channel 2

nVersion : The pointer of the FPGA version.

**Return Value:**

If the function call fails, it returns "FALSE".

If the function call succeeds, it returns "TRUE".

## LVDS\_SetDUse

This function sets a Data Valid or Horizontal Synchronization signal.

**BOOL            LVDS\_SetDUse (int nBoard, int nCh, BOOL bUse)**

**Parameters:**

nBoard : It informs a board number at currently equipped system.

The board number set up by DIP switch.

nCh : Select a channel. 0 : Channel 0, 1: Channel 1, 2: Channel 2

bUse : If the value is "True", use the DVAL(Data Validation).

If the value is "False", use the HSYNC (Horizontal Synchronization).

**Return Value:**

If the function call fails, it returns "FALSE".

If the function call succeeds, it returns "TRUE".

## LVDS\_CC\_Output

This function outputs the CC value of the corresponding bit.

**BOOL** **LVDS\_CC\_Output (int nBoard, int nCh, DWORD dwVal)**

**Parameters:**

nBoard : It informs a board number at currently equipped system.

The board number set up by DIP switch.

nCh : Select a channel. 0 : Channel 0, 1: Channel 1, 2: Channel 2

dwVal : As a 32-bit value, only the upper bit is valid.

bit0(CC1 out) = "0" : output "0" / "1": output "1"

bit1(CC2 out) = "0" : output "0" / "1": output "1"

bit2(CC3 out) = "0" : output "0" / "1": output "1"

bit3(CC4 out) = "0" : output "0" / "1": output "1"

others : Reserved

**Return Value:**

If the function call fails, it returns "FALSE".

If the function call succeeds, it returns "TRUE".

## LVDS\_SetTriggerOutput

This function outputs the CC value of the corresponding bit.

**BOOL** **LVDS\_SetTriggerOutput (int nBoard, int nCh, DWORD dwValue)**

**Parameters:**

nBoard : It informs a board number at currently equipped system.

The board number set up by DIP switch.

nCh : Select a channel. 0 : Channel 0, 1: Channel 1, 2: Channel 2

dwValue : bit0(CC1 out) = "0" : output "0" / "1": output "1"

bit1(CC2 out) = "0" : output "0" / "1": output "1"

bit2(CC3 out) = reserved

bit3(CC4 out) = reserved

**Return Value:**

If the function call fails, it returns "FALSE".

If the function call succeeds, it returns "TRUE".

## LVDS\_SetTrigger1Pulse

This function selects the output delay and output width of trigger 1 (CC1).

**BOOL**      **LVDS\_Set Trigger1Pulse (int nBoard, int nCh, DWORD nPulseCount,  
                  DWORD nPulseWidth)**

**Parameters:**

nBoard : It informs a board number at currently equipped system.

The board number set up by DIP switch.

nCh : Select a channel. 0 : Channel 0, 1: Channel 1, 2: Channel 2

nPulseCount : Delay value of output Trigger1

nPulseWidth : Width value of output Trigger1.

**Return Value:**

If the function call fails, it returns "FALSE".

If the function call succeeds, it returns "TRUE".

## LVDS\_SetTrigger2Pulse

This function selects the output delay and output width of trigger 2 (CC2).

**BOOL**      **LVDS\_Set Trigger2Pulse (int nBoard, int nCh, DWORD nPulseCount,  
                  DWORD nPulseWidth)**

**Parameters:**

nBoard : It informs a board number at currently equipped system.

The board number set up by DIP switch.

nCh : Select a channel. 0 : Channel 0, 1: Channel 1, 2: Channel 2

nPulseCount : Delay value of output Trigger2

nPulseWidth : Width value of output Trigger2.

**Return Value:**

If the function call fails, it returns "FALSE".

If the function call succeeds, it returns "TRUE".

## LVDS\_SetTrigger1Blank

This function selects the output interval of trigger 1 (CC1).

**BOOL            LVDS\_Set Trigger1Blank (int nBoard, int nCh, DWORD nPulseBlank)**

**Parameters:**

nBoard : It informs a board number at currently equipped system.

The board number set up by DIP switch.

nCh : Select a channel. 0 : Channel 0, 1: Channel 1, 2: Channel 2

nPulseBlank : Blank value of output Trigger1 .

**Return Value:**

If the function call fails, it returns "FALSE".

If the function call succeeds, it returns "TRUE".

## LVDS\_SetTrigger2Blank

This function selects the output interval of trigger 2 (CC2).

**BOOL            LVDS\_Set Trigger2Blank (int nBoard, int nCh, DWORD nPulseBlank)**

**Parameters:**

nBoard : It informs a board number at currently equipped system.

The board number set up by DIP switch.

nCh : Select a channel. 0 : Channel 0, 1: Channel 1, 2: Channel 2

nPulseBlank : Blank value of output Trigger2.

**Return Value:**

If the function call fails, it returns "FALSE".

If the function call succeeds, it returns "TRUE".

## LVDS\_SetTrigger1Inv

This function inverts the pulse of Trigger1 (CC1).

**BOOL LVDS\_SetTrigger1Inv (int nBoard, int nCh, BOOL bUse)**

**Parameters:**

nBoard : It informs a board number at currently equipped system.

The board number set up by DIP switch.

nCh : Select a channel. 0 : Channel 0, 1: Channel 1, 2: Channel 2

bUse : If "True", use Negative Pulse and

If "False", use Positive Pulse

**Return Value:**

If the function call fails, it returns "FALSE".

If the function call succeeds, it returns "TRUE".

## LVDS\_SetTrigger2Inv

This function inverts the pulse of Trigger2 (CC2).

**BOOL LVDS\_SetTrigger2Inv (int nBoard, int nCh, BOOL bUse)**

**Parameters:**

nBoard : It informs a board number at currently equipped system.

The board number set up by DIP switch.

nCh : Select a channel. 0 : Channel 0, 1: Channel 1, 2: Channel 2

bUse : If "True", use Negative Pulse and

If "False", use Positive Pulse

**Return Value:**

If the function call fails, it returns "FALSE".

If the function call succeeds, it returns "TRUE".

## UART API Functions

### *Overview*

<b>BOOL</b>	<b>UART_Init (int nBoard, int nCh)</b>
<b>BOOL</b>	<b>UART_GetData (int nBoard, int nCh, DWORD* nCnt, unsigned char* buf)</b>
<b>BOOL</b>	<b>UART_SendData (int nBoard, int nCh, DWORD* nCnt, unsigned char* buf)</b>
<b>BOOL</b>	<b>UART_Close (int nBoard, int nCh)</b>
<b>BOOL</b>	<b>UART_SetBaud (int nBoard, int nCh, DWORD nBaud)</b>

### **UART\_Init**

This function initialize resources used for the UART sub-system, for example interrupt and UART control register.

<b>BOOL</b>	<b>UART_Init (int nBoard, int nCh)</b>
-------------	--

#### **Parameters:**

nBoard : It informs a board number at currently equipped system.

The board number set up by DIP switch.

nCh : Select a channel. 0 : Channel 0, 1: Channel 1, 2: Channel 2

#### **Return Value:**

If the function call fails, it returns "FALSE".

If the function call succeeds, it returns "TRUE".

### **UART\_GetData**

This function receives characters through the differential UART.

<b>BOOL</b>	<b>UART_GetData (int nBoard, int nCh, DWORD* nCnt, unsigned char* buf)</b>
-------------	--

#### **Parameters:**

nBoard : It informs a board number at currently equipped system.

The board number set up by DIP switch.

nCh : Select a channel. 0 : Channel 0, 1: Channel 1, 2: Channel 2

nCnt : The address which contains the number of characters to be received.

The maximum number of characters to be received is limited to 4Kbyte(4096).

buf : The buffer address.

#### **Return Value:**

If the function call fails, it returns "FALSE".

If the function call succeeds, it returns "TRUE".

## UART\_SendData

This function sends characters through the differential UART.

**BOOL           UART\_SendData (int nBoard, int nCh, DWORD\* nCnt, unsigned char\* buf)**

**Parameters:**

nBoard : It informs a board number at currently equipped system.

The board number set up by DIP switch.

nCh : Select a channel. 0 : Channel 0, 1: Channel 1, 2: Channel 2

nCnt : The address which contains the number of characters to be sent.

The maximum number of characters to be sent is limited to 4K byte(4096).

buf : The buffer address

**Return Value:**

If the function call fails, it returns "FALSE".

If the function call succeeds, it returns "TRUE".

## UART\_Close

This function releases all resource were used for UART function.

**BOOL           UART\_Close (int nBoard, int nCh)**

**Parameters:**

nBoard : It informs a board number at currently equipped system.

The board number set up by DIP switch.

nCh : Select a channel. 0 : Channel 0, 1: Channel 1, 2: Channel 2

**Return Value:**

If the function call fails, it returns "FALSE".

If the function call succeeds, it returns "TRUE".

## UART\_SetBaud

This function sets UART Baud rates.

**BOOL            UART\_SetBaud (int nBoard, int nCh, DWORD nBaud)**

**Parameters:**

nBoard : It informs a board number at currently equipped system.

The board number set up by DIP switch.

nCh : Select a channel. 0 : Channel 0, 1: Channel 1, 2: Channel 2

nBaud : 0: 9600, 1: 19200, 2: 38400, 3:57600, 4:115200bps

**Return Value:**

If the function call fails, it returns "FALSE".

If the function call succeeds, it returns "TRUE".

## DIO(Digital Input Output) API Functions

### *Overview*

**BOOL**      **DIO\_Read (int nBoard, int nCh, DWORD\* Val)**  
**BOOL**      **DIO\_Write (int nBoard, int nCh, DWORD Val)**

### **DIO\_Read**

This function reads 32bit digital input value.

**BOOL**      **DIO\_Read (int nBoard, int nCh, DWORD\* Val)**

#### **Parameters:**

nBoard : It informs a board number at currently equipped system.  
The board number set up by DIP switch.  
nCh : Select a channel. 0 : Channel 0, 1: Channel 1, 2: Channel 2  
\*Val : It is a variable from which to read the current value of the input port.

#### **Return Value:**

If the function call fails, it returns "FALSE".  
If the function call succeeds, it returns "TRUE".

### **DIO\_Write**

This function outputs a 32-bit digital value to the output port.

**BOOL**      **DIO\_Write (int nBoard, int nCh, DWORD Val)**

#### **Parameters:**

nBoard : It informs a board number at currently equipped system.  
The board number set up by DIP switch.  
nCh : Select a channel. 0 : Channel 0, 1: Channel 1, 2: Channel 2  
Val : Value to write to the output port.

#### **Return Value:**

If the function call fails, it returns "FALSE".  
If the function call succeeds, it returns "TRUE".

# Memo

## Contact Point

Web sit : <https://www.daqsystem.com>

Email : [postmaster@daqsystem.com](mailto:postmaster@daqsystem.com)

