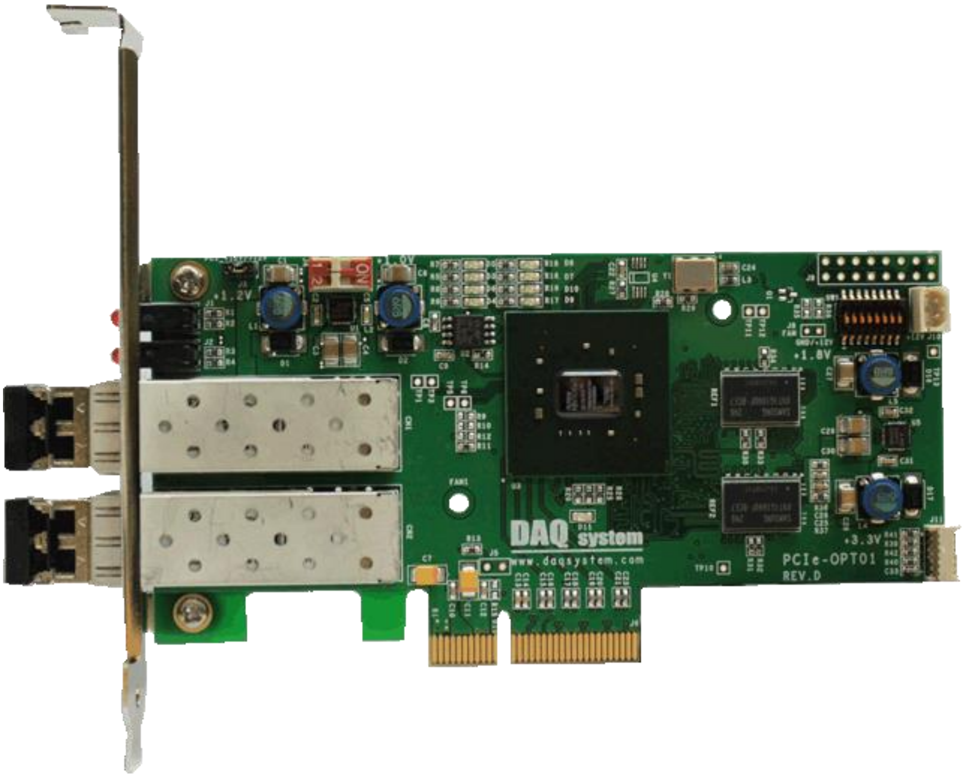
PCIe-OPT01

API Manual

Version 1.1



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.

DAQ SYSTEM
Measurement & Automation

Contents

Board Level API Functions

LVDS(Camera Link) API Functions

UART API Functions

Board Level API Functions

Overview

int	**OpenDAQDevice (void)**
BOOL	**ResetBoard (int nBoard)**
BOOL	**CloseDAQDevice (void)**
int	**GetBoardNum (void)**

OpenDAQDevice

This function initializes the device. You may call this function at the very first time you run the program.

BOOL OpenDAQDevice (void)

Parameters: None .

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)

ResetBoard

This function initializes a device at currently equipped system (PC).

BOOL ResetBoard (int nBoard)

Parameters:

nBoard : Numbers of discovered device. The board number is set up by DIP switch.

Return Value:

It returns TRUE in case of the success of reset and initialization.

If you get FALSE you should not call any API functions with the board and call the **CloseDAQDevice()** instead.

CloseDAQDevice

This function closes all opened devices (boards). If using of device is finished, you must certainly close a device for making it other programs so as usable.

BOOL CloseDAQDevice (void)

Parameters: None.

Return Value:

If the function fail to close, it returns "FALSE".

If the function succeed to close, it returns "TRUE".

GetBoardNum

This function returns currently detected board number in the system. If one board is installed, "1" is displayed. Up to 4 can be connected, and "4" is the maximum value.

int GetBoardNum (void)

Parameters: None

Return Value:

The number of boards, The Board number is set by dip switch.

LVDS(Camera Link) API Functions

Overview

BOOL	**PCI_LVDS_Init (int nBoard, int nCh)**
BOOL	**PCI_LVDS_Start (int nBoard, int nCh)**
BOOL	**PCI_LVDS_GetFrame (int nBoard, int nCh, DWORD* nCnt, unsigned char* buf)**
BOOL	**PCI_LVDS_Close (int nBoard, int nCh)**
BOOL	**PCI_LVDS_SetResolutuion (int nBoard, int nCh, DWORD *xRes, DWORD *yRes)**
BOOL	**PCI_LVDS_GetResolutuion (int nBoard, int nCh, DWORD *xRes, DWORD *yRes)**
BOOL	**PCI_LVDS_Stop (int nBoard, int nCh)**
DWORD	**PCI_LVDS_GetError (int nBoard, int nCh, DWORD *dwStatuse)**
BOOL	**PCI_LVDS_SetDataMode (int nBoard, int nCh, int nMode)**
BOOL	**PCI_LVDS_GetVersion (int nBoard, int nCh, int *nFpgaVer, int *nFirmVer)**
BOOL	**PCI_CC_Output (int nBoard, int nCh, DWORD dwVal)**
BOOL	**PCI_LVDS_DdrUse(int nBaord, int nCh, BOOL bUse)**

PCI_LVDS_Init

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

BOOL **PCI_LVDS_Init (int nBoard, int nCh)**

Parameters:

nBoard : Numbers of discovered device.
The board number is set up by DIP switch.

nCh : Selects the optical channel. "0" : Channel 0, "1" : Channel 1

Return Value:

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

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

PCI_LVDS_Start

This function starts receiving frame data. After calling this function, by calling PCI_LVDS_GetFrame function can be checked the complete data.

BOOL PCI_LVDS_Start (int nBoard, nCh)

Parameters:

nBoard : Numbers of discovered device.

The board number is set up by DIP switch.

nCh : Selects the optical channel. "0" : Channel, "1" : Channel 1

Return Value:

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

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

PCI_LVDS_GetFrame

This function acquires the image data from the frame buffer.

The size of the buffer to receive the data should be informed..

BOOL PCI_LVDS_GetFrame (int nBoard, int nCh, DWORD* nCnt, unsigned char* buf)

Parameters:

nBoard : Numbers of discovered device.

The board number is set up by DIP switch.

nCh : Selects the optical channel. "0" : Channel, "1" : Channel 1

*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.

*buf : Pointer of first pixel of image data.

Return Value:

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

If the function call succeeds, check the values of the size that you want to read nCnt.

PCI_LVDS_Close

This function releases all resources that used for LVDS function.

At the end of the program, the application program calls this function.

BOOL PCI_LVDS_Close (int nBoard, int nCh)

Parameters:

nBoard : Numbers of discovered device.

The board number is set up by DIP switch.

nCh : Selects the optical channel. "0" : Channel, "1" : Channel 1

Return Value:

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

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

PCI_LVDS_SetResolution

This function selects the resolution of the video input. The frame size is determined by this resolution.

BOOL PCI_LVDS_SetResolutuion (int nBoard, int nCh, DWORD xRes, DWORD yRes)

Parameters:

nBoard : Numbers of discovered device.

The board number is set up by DIP switch.

nCh : Selects the optical channel. "0" : Channel, "1" : Channel 1

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

yRes : Sets the vertical resolution value, 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".

PCI_LVDS_GetResolution

This function gets currently configured camera's frame resolution

BOOL PCI_LVDS_GetResolutuion (int nBoard, int nCh, DWORD *xRes, DWORD *yRes)

Parameters:

nBoard : Numbers of discovered device.

The board number is set up by DIP switch.

nCh : Selects the optical channel. "0" : Channel, "1" : Channel 1

*xRes : Width of image in pixels

*yRes : Height of Image in pixels

Return Value:

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

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

PCI_LVDS_Stop

This function stops the frame data capture.

BOOL PCI_LVDS_Stop (int nBoard, int nCh)

Parameters:

nBoard : Numbers of discovered device.

The board number is set up by DIP switch.

nCh : Selects the optical channel. "0" : Channel, "1" : Channel 1

Return Value:

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

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

PCI_LVDS_GetError

This function gets a Frame (Image) Error.

DWORD PCI_LVDS_GetError (int nBoard, int nCh, DWORD *dwStatus)

Parameters:

nBoard : Numbers of discovered device.
The board number is set up by DIP switch.

nCh : Selects the optical channel. "0" : Channel, "1" : Channel 1

*dwStatus : "1" : Overflow error
"2" : Read error
"4" : Size error

Return Value :

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

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

PCI_LVDS_SetDataMode

This function sets the image pixel data mode.

BOOL PCI_LVDS_SetDataMode (int nBoard, int nCh, int nMode)

Parameters:

nBoard : Numbers of discovered device.
The board number is set up by DIP switch.

nCh : Selects the optical channel. "0" : Channel, "1" : Channel 1

nMode : "0" : 8bit Mode, "1" : 16bit Mode
"2" : 24bit Mode, "3" : 32bit Mode
"4" : 16Bit YUV Mode.

Return Value :

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

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

PCI_LVDS_GetVersion

This function gets current FPGA and Firmware version.

BOOL PCI_LVDS_GetVersion (int nBoard, int nCh, int *nFpgaVer, int *nFirmVer)

Parameters:

nBoard : Numbers of discovered device.

The board number is set up by DIP switch.

nCh : Selects the optical channel. "0" : Channel, "1" : Channel 1

*nFpgaVer : FPGA version.

*nFirmVer : Firmware version.

Return Value :

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

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

PCI_CC_Output

This function outputs the CC value.

BOOL PCI_CC_Output (int nBoard, int nCh, DWORD dwVal)

Parameters:

nBoard : Numbers of discovered device.

The board number is set up by DIP switch.

nCh : Selects the optical channel. "0" : Channel, "1" : Channel 1

dwVal : 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".

PCI_LVDS_DdrUse

This function determines whether to use DDR memory.

BOOL PCI_LVDS_DdrUse (int nBoard, int nCh, BOOL bUse)

Parameters:

nBoard : Numbers of discovered device.
The board number is set up by DIP switch.

nCh : Selects the optical channel. "0" : Channel, "1" : Channel 1

bUse : If 'TRUE', use DDR memory
If 'FALSE', DDR memory is not used.

Return Value :

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

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

UART API Functions

Overview

BOOL PCI_UART_Init (int nBoard, int bCh)

BOOL PCI_UART_GetData (int nBoard, int nCh, DWORD* nCnt, unsigned char* buf)

BOOL PCI_UART_SetData (int nBoard, int nCh, DWORD * nCnt, unsigned char* buf)

BOOL PCI_UART_Close (int nBoard, int nCh)

BOOL PCI_UART_SetBaud (int nBoard, int nCh, DWORD nBaud)

PCI_UART_Init

This function initializes all resources that used for the UART sub-system, for example Interrupt and UART control register.

BOOL PCI_UART_Init (int nBoard, int nCh)

Parameters:

nBoard : Numbers of discovered device.
The board number is set up by DIP switch.

nCh : Selects the optical channel. "0" : Channel, "1" : Channel 1

Return Value:

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

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

PCI_UART_GetData

This function receives the characters through the UART.

BOOL PCI_UART_GetData (int nBoard, int nCh, DWORD* nCnt, unsigned char* buf)

Parameters:

nBoard : Numbers of discovered device.
The board number is set up by DIP switch.

nCh : Selects the optical channel. "0" : Channel, "1" : Channel 1

*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".

PCI_UART_SendData

This function sends the characters through the UART.

BOOL PCI_UART_SendData (int nBoard, int nCh, DWORD* nCnt, unsigned char* buf)

Parameters:

nBoard : Numbers of discovered device.
The board number is set up by DIP switch.

nCh : Selects the optical channel. "0" : Channel, "1" : Channel 1

*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".

PCI_UART_Close

This function releases all resource that used for UART function.

BOOL PCI_UART_Close (int nBoard, int nCh,)

Parameters:

nBoard : Numbers of discovered device.
The board number is set up by DIP switch.

nCh : Selects the optical channel. "0" : Channel, "1" : Channel 1

Return Value:

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

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

PCI_UART_SetBaud

This function sets an UART Baud.

BOOL PCI_UART_SetBaud (int nBoard, int nCh, DWORD nBaud)

Parameters:

nBoard : Numbers of discovered device.
The board number is set up by DIP switch.

nCh : Selects the optical channel. “0” : Channel, “1” : Channel 1

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

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

DAQ SYSTEM

Measurement & Automation