

PCIe-HDMI01

API Manual

Ver. 1.0



© 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	-----	3
CloseDAQDevice	-----	3
GetDIIVersion	-----	3

LVDS API Functions

LVDS_GetVersion	-----	4
LVDS_Init	-----	4
LVDS_Close	-----	5
LVDS_Start	-----	5
LVDS_Stop	-----	5
LVDS_GetFrame	-----	6
LVDS_GetFrameRate	-----	6
LVDS_GetResolution	-----	7
LVDS_DdrUse	-----	7

HDMI API Functions

HDMI_SetDataMode	-----	8
HDMI_Reset	-----	8
HDMI_SetPort	-----	9

Board Level APIs

Overview

Int	OpenDAQDevice (void)
BOOL	CloseDAQDevice (void)
char	GetDllVersion (void)

OpenDAQDevice

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

Int **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.

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".

GetDllVersion

This function gives the DLL version.

Char* **GetDllVersion (void)**

Parameters: None.

Return Value:

The date of the installed DLL. Ex)Feb 15 2021

LVDS API Functions

Overview

BOOL	LVDS_GetVersion (int nBoard, int *nFpga)
BOOL	LVDS_Init (int nBoard)
BOOL	LVDS_Close (int nBoard)
BOOL	LVDS_Start (int nBoard)
BOOL	LVDS_Stop (int nBoard)
BOOL	LVDS_GetFrame (int nBoard, DWORD* nCnt, unsigned char* buf)
BOOL	LVDS_GetFrameRate (int nBoard, DWORD* nCnt, unsigned char* buf)
BOOL	LVDS_GetResolutuion (int nBoard, DWORD *xRes, DWORD *yRes)
BOOL	LVDS_DdrUse (int nBoard, int nCh, BOOL bUse)

LVDS_GetVersion

This function gives the FPGA version.

BOOL LVDS_GetVersion (int nBoard, int *nFpga)

Parameters:

nBoard : Numbers of discovered device.

The board number is set up by DIP switch.

*nFpga : Address to get the FPGA Version value.

Return Value:

"-1": Exceeding maximum (4 units) devices, "-2": Device not open

LVDS_Init

This function initializes all resources used in the LVDS function.

BOOL LVDS_Init (int nBoard)

Parameters:

nBoard : Numbers of discovered device.

The board number is set up by DIP switch.

Return Value:

"-1": Exceeding maximum (4 units) devices, "-2": Device not open

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 LVDS_Close (int nBoard)

Parameters:

nBoard : Numbers of discovered device.

The board number is set up by DIP switch.

Return Value:

"-1": Exceeding maximum (4 units) devices, "-2": Device not open

LVDS_Start

When the LVDS_Start function is called, data transfer starts from the board to the DLL. (Data is accumulated in the buffer of the DLL through the thread, and the LVDS_GetFrame function is used to copy it to DLL -> APP.)

BOOL LVDS_Start (int nBoard)

Parameters:

nBoard : Numbers of discovered device.

The board number is set up by DIP switch.

Return Value:

""-1": Exceeding maximum (4 units) devices, "-2": Device not open

LVDS_Stop

This function stops the frame data capture.

BOOL LVDS_Stop (int nBoard)

Parameters:

nBoard : Numbers of discovered device.

The board number is set up by DIP switch.

Return Value :

"-1": Exceeding maximum (4 units) devices, "-2": Device not open

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 **LVDS_GetFrame (int nBoard, DWORD* nCnt, unsigned char* buf)**

Parameters:

nBoard : Numbers of discovered device.

The board number is set up by DIP switch.

*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:

"-1": Exceeding maximum (4 units) devices, "-2": Device not open,

"-3" : Incorrect parameter

LVDS_GetFrameRate

This function shows the number of frames per second actually received in H/W.

BOOL **LVDS_GetFrameRate (int nBoard, DWORD *dwRate)**

Parameters:

nBoard : Numbers of discovered device.

The board number is set up by DIP switch.

*dwRate : Address of Number of frames

Return Value:

"-1": Exceeding maximum (4 units) devices, "-2": Device not open

LVDS_GetResolution

This function gets currently configured camera's frame resolution.

BOOL LVDS_GetResolutuion (int nBoard, DWORD *xRes, DWORD *yRes)

Parameters:

nBoard : Numbers of discovered device.

The board number is set up by DIP switch.

*xRes : Width of image in pixels

*yRes : Height of Image in pixels

Return Value:

"-1": Exceeding maximum (4 units) devices, "-2": Device not open,

"-3" : Incorrect parameter

LVDS_DdrUse

This function determines whether DDR memory is used or not.

BOOL PCI_LVDS_DdrUse (int nBoard, BOOL bUse)

Parameters:

nBoard : Numbers of discovered device.

The board number is set up by DIP switch.

bUse : "0" : Use not DDR Memory, "1" : Use DDR Memory

Return Value :

"-1": Exceeding maximum (4 units) devices, "-2": Device not open,

"-3" : Incorrect parameter

HDMI API Functions

Overview

BOOL HDMI_SetDataMode (int nBoard, int nRES_SET, int nRGB_YUV_SET,
 int nDATA_BIT_SET)
BOOL HDMI_Reset (int nBoard)
BOOL HDMI_SetPort (int nBoard, int nPort)

HDMI_SetDataMode

This function sets the resolution, video format (YUV or RGB), and video data bit of the HDMI chip.

BOOL CLK_Select (int nBoard, int nRES_SET, int nRGB_YUV_SET,
 int nDATA_BIT_SET)

Parameters:

nBoard : Numbers of discovered device.

The board number is set up by DIP switch.

[Table 1. Parameter Selection Table]

Parameter	Selection					
	"0" : 480_570		"1" : 720_1080		"2" : 4K_2K	
nRES_SET						
nRGB_YUV_SET	"0" : YCbCr422	"1" : RGB_444	"0" : YCbCr422	"1" : RGB_444	"0" : YCbCr422	"1" : RGB_444
nDATA_BIT_SET	"0" : 24bits RGB	"0" : 24bits RGB	"0" : 24bits RGB	"0" : 24bits RGB	"0" : YCbCr444_422	"0" : YCbCr444_422
	"1" : 30bits RGB	"1" : 30bits RGB	"1" : 30bits RGB	"1" : 30bits RGB		
	"2" : 16bits YUV,	"2" : 16bits YUV,	"2" : 16bits YUV,	"2" : 16bits YUV,		
	"3" : 20bits YUV	"3" : 20bits YUV	"3" : 20bits YUV	"3" : 20bits YUV		
	"4" : 24bits YUV	"4" : 24bits YUV	"4" : 24bits YUV	"4" : 24bits YUV		

Return Value:

"-1": Exceeding maximum (4 units) devices, "-2": Device not open,

"-3" : Incorrect parameter

HDMI_Reset

This function initializes the settings of the HDMI chip.

BOOL HDMI_Reset (int nBoard)

Parameters:

nBoard : Numbers of discovered device.

The board number is set up by DIP switch.

Return Value:

"-1": Exceeding maximum (4 units) devices, "-2": Device not open,

HDMI_SetPort

This function sets whether to use the HDMI port of the board as Port A (J1 connector) or Port B (J2 connector).

BOOL HDMI_SetPort (int nBoard, int nPort)

Parameters:

nBoard : Numbers of discovered device.

The board number is set up by DIP switch.

nPort : "0" : Port A , '1" Port B

Return Value:

"-1": Exceeding maximum (4 units) devices, "-2": Device not open,

Memo

Contact Point

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

Email : postmaster@daqsystem.com

