

PCIe-FRM24

API Manual

Version 1.3



© 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
ResetBoard	4
CloseDAQDevice	5
GetBoardNum	5

LVDS(Camera Link) API Functions

LVDS_Init	6
LVDS_Start	7
LVDS_GetFrame	7
LVDS_Close	8
LVDS_SetResolution	8
LVDS_GetResolution	8
LVDS_Stop	9
LVDS_GetFrameSize	9
LVDS_SetDataMode	9
LVDS_GetVersion	10
LVDS_CameraMode	10
LVDS_TapControl	10
LVDS_CC_Output	11
LVDS_SetVsPol	11
LVDS_VsyncEn	11

UART API Functions

UART_Init	12
UART_GetData	12
UART_SendData	13
UART_Close	13
UART_SetBaud	13
UART_BufferFlush	14

Multi-Board LVDS(Camera Link) API Functions

LVDS_Init_Mul	-----	15
LVDS_Start_Mul	-----	16
LVDS_GetFrame_Mul	-----	16
LVDS_Close_Mul	-----	17
LVDS_SetResolution_Mul	-----	17
LVDS_GetResolution_Mul	-----	18
LVDS_Stop_Mul	-----	18
LVDS_GetFrameSize_Mul	-----	18
LVDS_SetDataMode_Mul	-----	19
LVDS_GetVersion_Mul	-----	19
LVDS_CameraMode_Mul	-----	20
LVDS_tapControl_Mul	-----	20
LVDS_CC_Output_Mul	-----	21
LVDS_SetVsPol_Mul	-----	21
LVDS_VsuncEn_Mul	-----	22

Multi-Board UART API Functions

UART_Init_Mul	-----	23
UART_GetData_Mul	-----	23
UART_SendData_Mul	-----	24
UART_Close_Mul	-----	24
UART_SetBaud_Mul	-----	25
UART_BufferFlush_Mul	-----	25

Board Level API Functions

Overview

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

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: 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

It initializes a device at currently equipped system (PC).

BOOL ResetBoard (int nBoard)

Parameters:

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

The board number 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

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: None.

Return Value:

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

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

GetBoardNum

Returns currently detected board number in the system.

int **GetBoardNum (void)**

Parameters: None

Return Value:

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

LVDS(Camera Link) API Functions

Overview

BOOL	LVDS_Init (void)
BOOL	LVDS_Start (void)
BOOL	LVDS_GetFrame (DWORD* nCnt, unsigned char* buf)
BOOL	LVDS_Close (void)
BOOL	LVDS_SetResolution (DWORD xRes, DWORD yRes)
BOOL	LVDS_GetResolution (DWORD *xRes, DWORD *yRes)
BOOL	LVDS_Stop (void)
BOOL	LVDS_GetFrameSize (DWORD *xRes, DWORD *yRes)
BOOL	LVDS_SetDataMode (int nMode)
BOOL	LVDS_GetVersion (int *nVersion)
BOOL	LVDS_CameraMode (int nMode)
BOOL	LVDS_TapControl (int nTap)
BOOL	LVDS_CC_Output (DWORD dwVal)
BOOL	LVDS_SetVsPol (BOOL bPol)
BOOL	LVDS_VsyncEn (BOOL bEn)

LVDS_Init

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

BOOL **LVDS_Init (void)**

Parameters: None.

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 (void)**

Parameters: None.

Return Value:

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

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

LVDS_GetFrame

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_GetFrame (DWORD* nCnt, unsigned char* buf)**

Parameters:

*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 : Frame buffer pointer.

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.

(Note) If the frame data is not completed, FALSE is returned immediately and the return occurs with the nCnt value set to 0.

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 (void)**

Parameters: None.

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 (DWORD xRes, DWORD yRes)**

Parameters:

xRes : Value of the horizontal Camera resolution

yRes : Value of the vertical Camera resolution

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 (DWORD *xRes, DWORD *yRes)**

Parameters:

*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_Stop

This function stops the frame data capture.

BOOL LVDS_Stop (**void**)

Parameters: None.

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 (**DWORD** ***xRes**, **DWORD** ***yRes**)

Parameters:

*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** nMode)

Parameters:

nMode : "0" : 8bit Mode, "1" : 16bit Mode

"2" : 24bit Mode, "3" : 64bit Mode

Return Value:

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

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

LVDS_GetVersion

This function gets a FPGA version.

BOOL LVDS_GetVersion (int *nVersion)

Parameters:

*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_CameraMode

This function sets a Camera Operation Mode.

BOOL LVDS_CameraMode (int nMode)

Parameters:

nMode : "0" : Area Scan Camera Mode (Default)

"1" : Line Scan Camera Mode (Free Run)

"2" : Line Scan Camera Mode (Trigger from external port)

"3" : Line Scan Camera Mode (Trigger from internal clock 33MHz)

Return Value :

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

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

LVDS_TapControl

This function selects the camera mode.

BOOL LVDS_TapControl (int nTap)

Parameters:

nTap : "0" : Normal Mode, "1" : Alternate Mode.

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, DWORD dwVal)

Parameters:

nBoard : The Board number is set by dip switch.

dwVal : bit0 → CC0 out

bit1 → CC1 out

bit2 → CC2 out

bit3 → CC3 out

Return Value :

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

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

LVDS_SetVsPol

This function selects the vertical signal (VSync: Vertical Synchronization) signal line.

BOOL LVDS_SetVsPol (BOOL bPol)

Parameters:

bPol : "0" : Use the Inverse VSync

"1" : Use the Normal Vsync

Return Value :

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

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

LVDS_VSyncEn

This function selects the vertical signal (VSync: Vertical Synchronization) signal line Enable/Disable. When disabled, no video is played. (requires initial On state)

BOOL LVDS_VSyncEn (BOOL bEn)

Parameters:

bEn : If it is "1", VSync On, and if it is "0", VSync Off is used.

Return Value :

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

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

UART API Functions

Overview

BOOL	UART_Init (void)
BOOL	UART_GetData (DWORD* nCnt, unsigned char* buf)
BOOL	UART_SendData (DWORD* nCnt, unsigned char* buf)
BOOL	UART_Close (void)
BOOL	UART_SetBaud (DWORD nBaud)
BOOL	UART_BufferFlush (void)

UART_Init

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

BOOL	UART_Init (void)
-------------	-------------------------

Parameters: None.

Return Value:

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

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

UART_GetData

This function receives characters through the differential UART.

BOOL	UART_GetData (DWORD* nCnt, unsigned char* buf)
-------------	---

Parameters:

*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 (DWORD* nCnt, unsigned char* buf)

Parameters:

*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 (void)

Parameters: None.

Return Value:

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

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

UART_SetBaud

This function sets UART Baud rates.

BOOL UART_SetBaud (DWORD nBaud)

Parameters:

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".

UART _BufferFlush

This function flushes UART RX Buffer

BOOL UART_BufferFlush (void)

Parameters: None.

Return Value:

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

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

Multi Board support APIs

In case of single board API, only one board is used in the installed system. However, in a system with two or more boards installed (up to 4 supported), multiple APIs must be used. Multi board API is only available for FPGA version #2 or higher boards.

Multi Board LVDS(Camera Link) APIs

Overview

BOOL	LVDS_Init_Mul (int nBoard)
BOOL	LVDS_Start_Mul (int nBoard)
BOOL	LVDS_GetFrame_Mul (int nBoard, DWORD* nCnt, unsigned char* buf)
BOOL	LVDS_Close_Mul (int nBoard)
BOOL	LVDS_SetResolution_Mul (int nBoard, DWORD xRes, DWORD yRes)
BOOL	LVDS_GetResolution_Mul (int nBoard, DWORD *xRes, DWORD *yRes)
BOOL	LVDS_Stop_Mul (int nBoard)
BOOL	LVDS_GetFrameSize_Mul (int nBoard, DWORD *xRes, DWORD *yRes)
BOOL	LVDS_SetDataMode_Mul (int nBoard, int nMode)
BOOL	LVDS_GetVersion_Mul (int nBoard, int *nVersion)
BOOL	LVDS_CameraMode_Mul (int nBoard, int nMode)
BOOL	LVDS_TapControl_Mul (int nBoard, int nTap)
BOOL	LVDS_CC_Output_Mul (int nBoard, DWORD dwVal)
BOOL	LVDS_SetVsPol_Mul (int nBoard, BOOL bPol)
BOOL	LVDS_VsyncEn_Mul (int nBoard, BOOL bEn)

LVDS_Init_Mul

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

BOOL LVDS_Init_Mul (int nBoard)

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_Start_Mul

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_Mul (int nBoard)

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_GetFrame_Mul

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

Parameters:

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

The board number 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. 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, check the values of the size that you want to read nCnt.

(Note) If the frame data is not completed, FALSE is returned immediately and the return occurs with the nCnt value set to 0.

LVDS_Close_Mul

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)**

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_SetResolution_Mul

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

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

Parameters:

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

The board number set up by DIP switch.

xRes : Value of the horizontal Camera resolution

yRes : Value of the vertical Camera resolution

Return Value:

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

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

LVDS_GetResolution_Mul

This function gets currently configured camera's frame resolution

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

Parameters:

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

The board number set up by DIP switch.

*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_Stop_Mul

This function stops the frame data capture.

BOOL LVDS_Stop_Mul (int nBoard)

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_GetFrameSize_Mul

This function gets the frame size of the Video input.

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

Parameters:

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

The board number set up by DIP switch.

*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_Mul

This function selects the frame (image) data mode.

BOOL LVDS_SetDataMode_Mul (int nBoard, int nMode)

Parameters:

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

The board number set up by DIP switch.

nMode : "0" : 8bit Mode, "1" : 16bit Mode

"2" : 24bit Mode, "3" : 64bit Mode

Return Value:

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

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

LVDS_GetVersion_Mul

This function gets a FPGA version.

BOOL LVDS_GetVersion_Mul (int nBoard, int *nVersion)

Parameters:

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

The board number set up by DIP switch.

*nVersion : The pointer of the FPGA version.

Some API are only supported by boards which have the FPGA version number 2 or more.

Return Value:

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

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

LVDS_CameraMode_Mul

This function sets a Camera Operation Mode.

BOOL LVDS_CameraMode_Mul (int nBoard, int nMode)

Parameters:

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

The board number set up by DIP switch.

nMode : "0" : Area Scan Camera Mode (Default)

"1" : Line Scan Camera Mode (Free Run)

"2" : Line Scan Camera Mode (Trigger from external port)

"3" : Line Scan Camera Mode (Trigger from internal clock 33MHz)

Return Value :

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

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

LVDS_TapControl_Mul

This function selects the camera mode.

BOOL LVDS_TapControl_Mul (int nBoard, int nTap)

Parameters:

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

The board number set up by DIP switch.

nTap : "0" : Normal Mode, "1" : Alternate Mode.

Return Value :

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

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

LVDS_CC_Output_Mul

This function outputs the CC value of the corresponding bit.

BOOL LVDS_CC_Output_Mul (int nBoard, DWORD dwVal)

Parameters:

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

The board number set up by DIP switch.

dwVal : bit0 → CC0 out

bit1 → CC1 out

bit2 → CC2 out

bit3 → CC3 out

Return Value :

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

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

LVDS_SetVsPol_Mul

This function selects the vertical signal (VSync: Vertical Synchronization) signal line.

BOOL LVDS_SetVsPol_Mul (int nBoarf, BOOL bPol)

Parameters:

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

The board number set up by DIP switch.

bPol : "0" : Use the Inverse VSync

"1" : Use the Normal Vsync

Return Value :

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

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

LVDS_VSyncEn_Mul

This function selects the vertical signal (VSync: Vertical Synchronization) signal line Enable/Disable. When disabled, no video is played. (requires initial On state)

BOOL LVDS_VSyncEn_Mul (int nBoard, BOOL bEn)

Parameters:

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

The board number set up by DIP switch.

bEn : If it is "1", VSync On,

and if it is "0", VSync Off is used.

Return Value :

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

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

Multi-Board UART API Functions

Overview

BOOL	UART_Init_Mul (int nBoard)
BOOL	UART_GetData_Mul (int nBoard, DWORD* nCnt, unsigned char* buf)
BOOL	UART_SendData_Mul (int nBoard, DWORD* nCnt, unsigned char* buf)
BOOL	UART_Close_Mul (int nBoard)
BOOL	UART_SetBaud_Mul (int nBoard, DWORD nBaud)
BOOL	UART_BufferFlush_Mul (int nBoard)

UART_Init_Mul

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

BOOL UART_Init_Mul (int nBoard)

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".

UART_GetData_Mul

This function receives characters through the differential UART.

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

Parameters:

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

The board number set up by DIP switch.

*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_Mul

This function sends characters through the differential UART.

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

Parameters:

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

The board number set up by DIP switch.

*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_Mul

This function releases all resource were used for UART function.

BOOL UART_Close_Mul (int nBoard)

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".

UART_SetBaud_Mul

This function sets UART Baud rates.

BOOL UART_SetBaud_Mul (int nBoard, DWORD nBaud)

Parameters:

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

The board number set up by DIP switch.

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".

UART_BufferFlush_Mul

This function flushes UART RX Buffer

BOOL UART_BufferFlush_Mul (int nBoard)

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".

Memo

Contact Point

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

Email : postmaster@daqsystem.com

