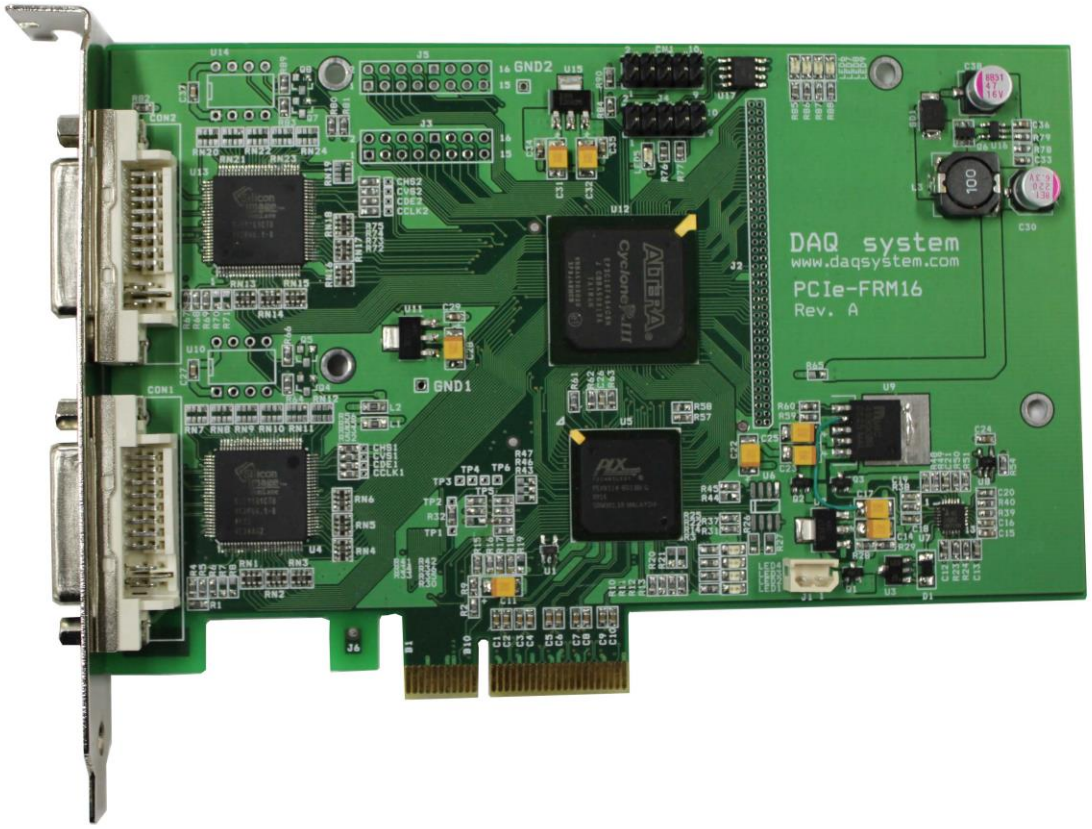
PCIe-FRM16 API Programming (Rev 1.2)



UPDATE HISTORY

2011-07-22 Add Function

BOOL LVDS_GetFrameSize (DWORD *xRes, DWORD *yRes)

BOOL LVDS_GetFrameSize_Mul (DWORD *xRes, DWORD *yRes)

Board Level APIs

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.

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.

int GetBoardNum (void)

Parameters: None

Return Value:

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

LVDS(Camera Link) APIs

Overview

BOOL LVDS_Init (void)
BOOL LVDS_Start (void)
BOOL LVDS_GetFrame (DWORD* nCnt, unsigned char* buf)
BOOL LVDS_Close (void)
BOOL LVDS_Stop (void)
BOOL LVDS_GetFrameSize (DWORD *xRes, DWORD *yRes)
BOOL LVDS_SetResolutuion (DWORD xRes, DWORD yRes)
BOOL LVDS_GetResolutuion (DWORD *xRes, DWORD *yRes)
BOOL LVDS_SetDataMode (int nMode)
BOOL LVDS_GetVersion (int *nVersion)

LVDS_Init

This function initialize resources 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, by calling LVDS_GetFrame() function can be checked the complete data.

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 acquires image data from the frame buffer.

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

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 : 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.

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 (void)

Parameters: None.

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 currently configured frame resolution.

BOOL LVDS_GetResolutuion (DWORD *xRes, DWORD *yRes)

Parameters:

*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”.

LVDS_SetResolution

This function sets the camera resolution for the specific camera Model.

BOOL LVDS_SetResolution (DWORD xRes, DWORD yRes)

Parameters:

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”.

LVDS_GetResolution

This function gets currently configured camera’s frame resolution

BOOL LVDS_GetResolution (DWORD *xRes, DWORD *yRes)

Parameters:

*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”.

LVDS_SetDataMode

This function sets the image pixel data mode.

BOOL LVDS_SetDataMode (int nMode)

Parameters:

nMode : If the value is 2, the pixel data is expressed by 24bits.
The others is expressed by 16bits.

Return Value:

If the function call fails, it returns “FALSE”.

If the function call succeeds, it returns “TRUE”.

LVDS_GetVersion

This function gets a current FPGA version.

BOOL LVDS_GetVersion (int *nVersion)

Parameters:

nVersion : FPGA version.

Return Value:

If the function call fails, it returns “FALSE”.

If the function call succeeds, it returns “TRUE”.

Digital Input/Output API Functions

Overview

DWORD **DIO_Read (void)**

BOOL **DIO_Write (DWORD Val)**

DIO_Read

This function reads the input value.

DWORD **DIO_Read (void)**

Parameters: None.

Return Value:

If the function call fails, it returns “FALSE”.

If the function call succeeds, it returns “TRUE”.

DIO_Write

This function writes the desired value to output port.

BOOL **DIO_Write (DWORD Val)**

Parameters:

dwVal : The value to be written to the port.

Return Value:

If the function call fails, it returns “FALSE”.

If the function call succeeds, it returns “TRUE”.

Multi Board support APIs

All single board API can be used with the system which have only one board installed, but multi board APIs must be used with the system which have more than two boards installed.

Note) Multi board API can be used with the board which have the FPGA version #2 or More(in the future). But, the current system does not support the use of two or more boards.

LVDS(DVI) 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_Stop_Mul (int nBoard)
BOOL LVDS_GetFrameSize_Mul (int nBoard, DWORD *xRes, DWORD *yRes)
BOOL LVDS_SetResolutuion_Mul (int nBoard, DWORD xRes, DWORD yRes)
BOOL LVDS_GetResolutuion_Mul (int nBoard, DWORD *xRes, DWORD *yRes)
BOOL LVDS_SetDataMode_Mul (int nBoard, int nMode)
BOOL LVDS_GetVersion_Mul (int nBoard, int *nVersion)

LVDS_Init_Mul

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

BOOL LVDS_Init_Mul (int nBoard)

Parameters:

nBoard : Numbers of discovered device. The board number is 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, by calling LVDS_GetFrame_Mul() function can be checked the complete data.

BOOL LVDS_Start_Mul (int nBoard)

Parameters:

nBoard : Numbers of discovered device. The board number is 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 acquires image data from the frame buffer.

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

BOOL LVDS_GetFrame_Mul (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. The data size is in bytes.

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.

LVDS_Close_Mul

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_Mul (int nBoard)

Parameters:

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

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 Numbers of discovered device. The board number is 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 currently configured frame resolution.

BOOL LVDS_GetResolutuion_Mul (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:

If the function call fails, it returns “FALSE”.

If the function call succeeds, it returns “TRUE”.

LVDS_SetResolution_Mul

This function sets the camera resolution for the specific camera Model.

BOOL LVDS_SetResolution_Mul (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:

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_GetResolution_Mul (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:

If the function call fails, it returns “FALSE”.

If the function call succeeds, it returns “TRUE”.

LVDS_SetDataMode_Mul

This function sets the image pixel data mode.

BOOL LVDS_SetDataMode_Mul (int nBoard, int nMode)

Parameters:

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

nMode : If the value is 2, the pixel data be expressed by 24bits, others be 16bits.

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 current FPGA version.

BOOL LVDS_GetVersion_Mul (int nBoard, int *nVersion)

Parameters:

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

*nVersion : FPGA version.

Return Value:

If the function call fails, it returns “FALSE”.

If the function call succeeds, it returns “TRUE”.

Multi-Board Digital Input/Output API Functions

Overview

DWORD **DIO_Read_Mul (int nBoard)**

BOOL **DIO_Write_Mul (int nBoard, DWORD dwVal)**

DIO_Read_Mul

This function reads input value.

DWORD **DIO_Read_Mul (int nBoard)**

Parameters:

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

Return Value:

If the function call fails, it returns “FALSE”.

If the function call succeeds, it returns “TRUE”.

DIO_Write_Mul

This function writes the desired value to output port.

BOOL **DIO_Write_Mul (int nBoard, DWORD dwVal)**

Parameters:

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

dwVal : The value to be written to the port.

Return Value:

If the function call fails, it returns “FALSE”.

If the function call succeeds, it returns “TRUE”.