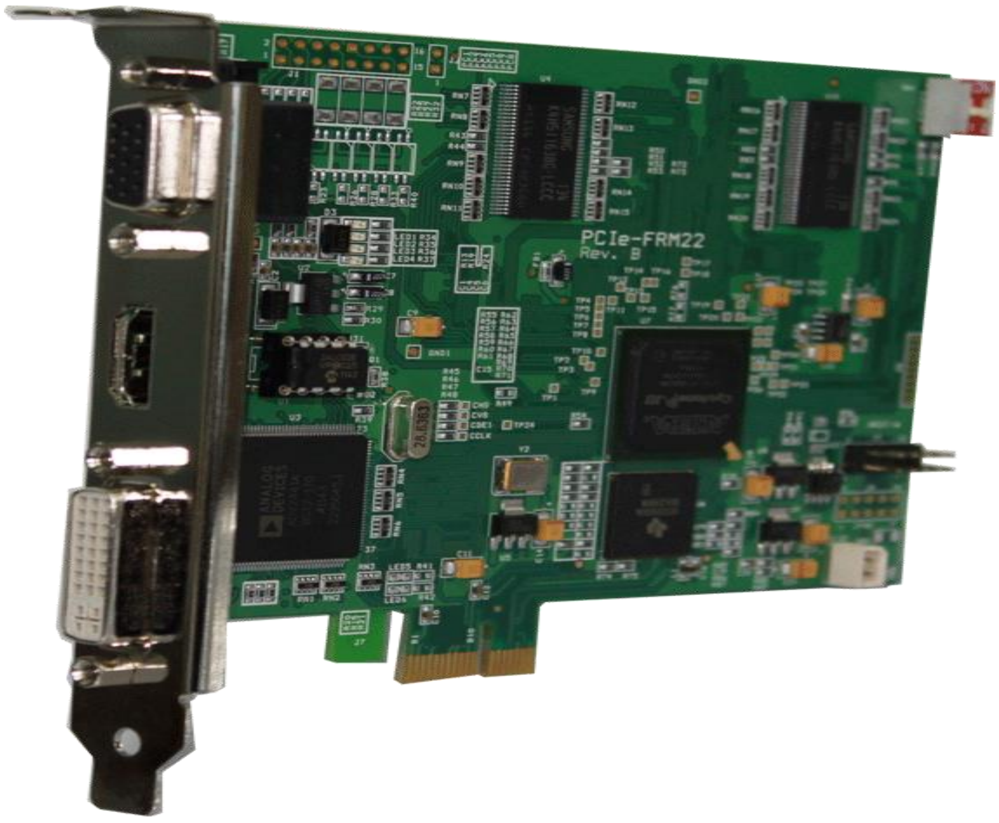
PCIe-FRM22

API Manual

Version 1.2



Contents

Multi-Board LVDS(Camera Link) API Functions

Multi-Board DIO(Digital Input Output)API Functions

Multi-Board I2C API Functions

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_SetDataMode (int nMode)**
BOOL	**LVDS_GetVersion (int *nVersion)**
BOOL	**LVDS_SelectVideo (int nVideo)**

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 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 (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 sets camera resolution for the specific camera Model. 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_SetDataMode

This function sets image pixel data mode.

BOOL LVDS_SetDataMode (int nMode)

Parameters:

nMode : If it is “2”, it is 24bit Mode, and if it is “Others”, it is 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 *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_SelectVideo

This function selects the video mode.

BOOL LVDS_SelectVideo (int nVideo)

Parameters:

nVideo :	“0” : HDMI,	“1” : DVI
	“2” : (RGB) 640x480,	“3” : (RGB) 800x600
	“4” : (RGB) 1024x768,	“5” : (RGB) 1024x720
	“6” : (RGB) 1280x720,	“7” : (RGB) 1280x768
	“8” : (RGB) 1280x960,	“9” : (RGB) 1280x1024
	“10” : (RGB) 1400x1050,	“11” : (RGB) 1600x1200
	“12” : (RGB) 1920x1200,	“13” : (RGB) SW_Reset

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 (void)**
BOOL	**DIO_Write (DWORD val)**

DIO_Read

This function reads the Digital Input state.

BOOL **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 outputs the desired camera control value to the output port. (refer to camera specifications)

BOOL **DIO_Write (DWORD val)**

Parameters:

val : 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".

I2C APIs

Overview

BOOL	**I2C_Init (DWORD nKHz)**
BOOL	**I2C_Read (BYTE byAddr, DWORD nCnt, unsigned char* buf)**
BOOL	**I2C_Write (BYTE byAddr, DWORD* nCnt, unsigned char* buf)**
BOOL	**I2C_Close (void)**

I2C_Init

This function determines the I2C communication speed.

BOOL I2C_Init (DWORD nKHz)

Parameters:

nKHz : "1" : 100KHz , "2" : 200KHz
"3" : 300KHz , "4" : 400KHz,
"Others" : 400KHz

Return Value:

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

I2C_Read

This function receives a character via I2C.

BOOL I2C_Read (BYTE byAddr, DWORD nCnt, unsigned char* buf)

Parameters:

byAddr : Device Address
nCnt : Number of bytes to read
*buf : Memory Buffer Address.

Return Value:

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

I2C_Write

This function sends a text via I2C.

BOOL I2C_Write (BYTE byAddr, DWORD nCnt, unsigned char* buf)

Parameters:

byAddr : Device Address

nCnt : Number of bytes to write

*buf : Memory Buffer Address.

Return Value:

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

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

I2C_Close

This function returns all the resources used by the I2C function.

BOOL UART_Close (void)

Parameters: None.

Return Value:

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

If the function call succeeds, 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_SetDataMode_Mul (int nBoard, int nMode)**
BOOL	**LVDS_GetVersion_Mul (int nBoard, int *nVersion)**
BOOL	**LVDS_SelectVideo_Mul (int nBoard, int nVideo)**

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 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_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, it returns "TRUE".

LVDS_Close_Mul

This function returns all the resources used by the 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 sets camera resolution for the specific camera Model. Frame size is determined according to this resolution.

BOOL LVDS_SetResolution_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_GetResolution_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_SetDataMode_Mul

This function sets image pixel 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 : 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 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.

Return Value:

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

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

LVDS_SelectVideo_Mul

This function selects the input image mode.

BOOL LVDS_SelectVideo_Mul (int nBoard, int nVideo)

Parameters:

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

nVideo :

"0" : HDMI,	"1" : DVI
"2" : (RGB) 640x480,	"3" : (RGB) 800x600
"4" : (RGB) 1024x768,	"5" : (RGB) 1024x720
"6" : (RGB) 1280x720,	"7" : (RGB) 1280x768
"8" : (RGB) 1280x960,	"9" : (RGB) 1280x1024
"10" : (RGB) 1400x1050,	"11" : (RGB) 1600x1200
"12" : (RGB) 1920x1200,	"13" : (RGB) SW_Reset

Return Value :

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

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

Multi-board DIO(Digital Input Output) API Functions

Overview

BOOL	**DIO_Read_Mul (int nBoard)**
BOOL	**DIO_Write_Mul (int nBoard, DWORD val)**

DIO_Read_Mul

This function reads from input port.

DWORD **DIO_Read_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".

DIO_Write_Mul

This function outputs the desired camera control value to the output port. (refer to camera specifications)

BOOL **DIO_Write (int nBoard, DWORD val)**

Parameters:

nBoard : It informs a board number at currently equipped system.
The board number set up by DIP switch.
val : 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 I2C APIs

Overview

BOOL I2C_Init_Mul (int nBoard, DWORD nKHz)

BOOL I2C_Read_Mul (int nBoard, BYTE byAddr, DWORD nCnt, unsigned char* buf)

BOOL I2C_Write_Mul (int nBoard, BYTE byAddr, DWORD* nCnt, unsigned char* buf)

BOOL I2C_Close_Mul (int nBoard)

I2C_Init_Mul

This function determines the I2C communication speed.

BOOL I2C_Init_Mul (int nBoard, DWORD nKHz)

Parameters:

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

nKHz : "1" : 100KHz , "2" : 200KHz
"3" : 300KHz , "4" : 400KHz,
"Others" : 400KHz

Return Value:

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

I2C_Read_Mul

This function receives a character via I2C.

BOOL I2C_Read_Mul (int nBoard, BYTE byAddr, DWORD nCnt, unsigned char* buf)

Parameters:

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

byAddr : Device Address

nCnt : Number of bytes to read

buf : Memory Buffer Address.

Return Value:

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

I2C_Write_Mul

This function sends a text via I2C.

BOOL I2C_Write_Mul (int nBoard, BYTE byAddr, DWORD nCnt, unsigned char* buf)

Parameters:

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

byAddr : Device Address

nCnt : Number of bytes to write

buf : Memory Buffer Address.

Return Value:

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

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

I2C_Close_Mul

This function returns all the resources used by the I2C function.

BOOL I2C_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".

Memo

Contact Point

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

Email : postmaster@daqsystem.com

DAQ SYSTEM
Measurement & Automation