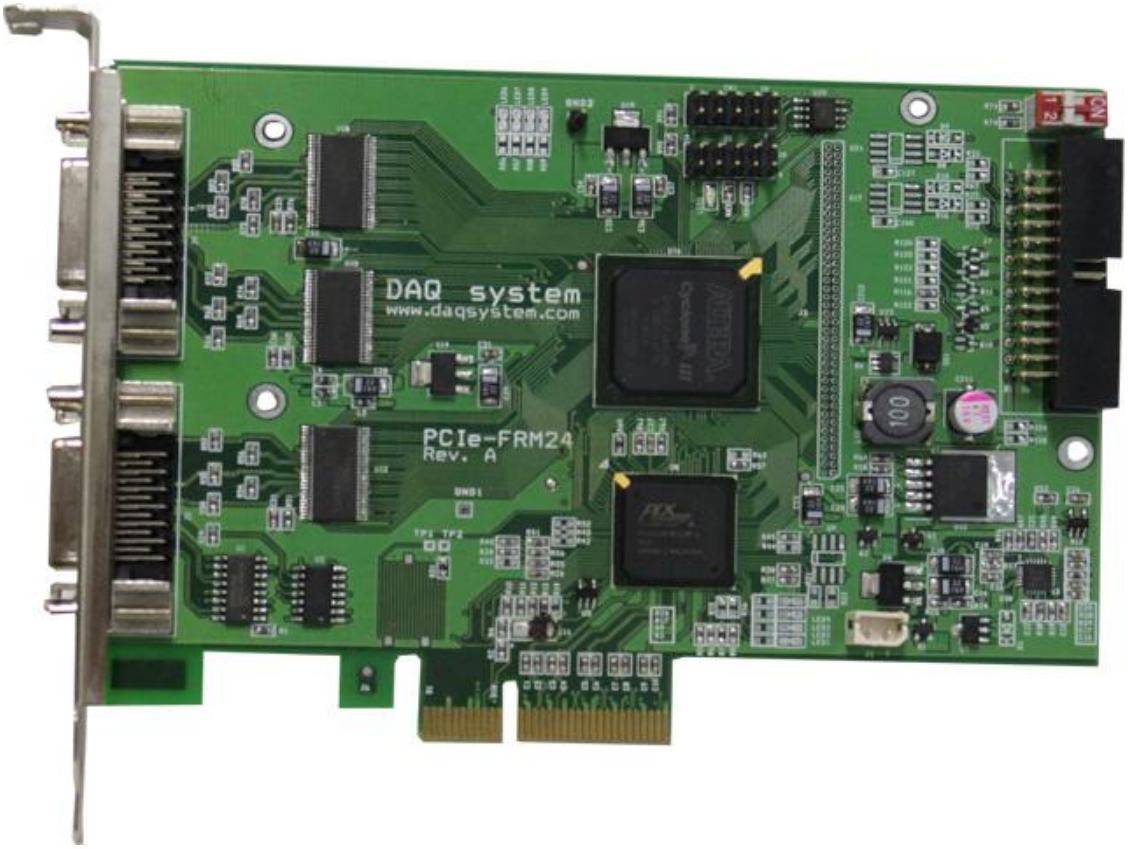
# PCIe-FRM24 API Programming (Rev 1.2)



# 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_SetResolutuion (DWORD xRes, DWORD yRes)**
**BOOL LVDS_GetResolutuion (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)**

## 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_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_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 size of video input.

**BOOL LVDS_GetFrameSize (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 “0”, the pixel data is expressed by 8bits.
If the value is “1”, the pixel data is expressed by 16bits.
If the value is “2”, the pixel data is expressed by 32bits.
If the value is “3”, the pixel data is expressed by 64bits.

**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”.

## LVDS_CameraMode

This function selects the Camera Mode.

**BOOL LVDS_CameraMode (int nMode)**

**Parameters**:

nMode : If the value is “0”, it is an Area Camera Mode.
If the value is “1”, it is a Line Camera Mode.
(Currently the 2 : Line (Ext) 3 : Line (Int) is not support)

**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 saved mode.

**BOOL LVDS_TapControl (int nTap)**

**Parameters**:

nTap : If the value is “0”, it is a Normal Mode.

If the value is “1”, it is an 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 CC value of corresponding bit.

**BOOL LVDS_CC_Output (DWORD dwVal)**

**Parameters**:

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”.

# UART APIs

## *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 call fails, it returns “FALSE”.

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

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

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

# Multi Board support APIs

Notes) 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.

## 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_SetResolutuion_Mul (int nBoard, DWORD xRes, DWORD yRes)**
**BOOL LVDS_GetResolutuion_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)**

### 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() 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.

**Caution)** If it is not complete Frame data and return FALSE, returns with nCnt value 0.

## 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_SetResolution_Mul

This function sets 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_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 size of video input.

**BOOL LVDS_GetFrameSize_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 “0”, the pixel data is expressed by 8bits.

If the value is “1”, the pixel data is expressed by 16bits.

If the value is “2”, the pixel data is expressed by 32bits.

If the value is “3”, the pixel data is expressed by 64bits.

**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”.

## LVDS_CameraMode_Mul

This function selects the Camera Mode.

**BOOL LVDS_CameraMode_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 “0”, it is an Area Camera Mode.

If the value is “1”, it is a Line Camera Mode.

(Currently the 2 : Line (Ext), 3 : Line (Int) is not support)

**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 save mode.

**BOOL LVDS_TapControl_Mul (int nBoard, int nTap)**

**Parameters**:

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

nTap : If the value is “0”, it is a Normal Mode.

If the value is “1”, it is an Alternate Mode.

**Return Value** :

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

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

# UART APIs

## *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 : The Board number is set 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 : The Board number is set 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 : The Board number is set 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 : The Board number is set 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 : The Board number is set by dip switch.

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_Mul

This function flushes UART RX Buffer

**BOOL UART_BufferFlush_Mul (int nBoard)**

**Parameters**:

nBoard : The Board number is set by dip switch.

**Return Value**:

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

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