PCIe-FRM24_C

API Manual

Version 1.2

Contents

Board Level API Functions

Overview

int	**OpenDAQDevice (void)**
BOOL	**ResetBoard (int nBoard)**
BOOL	**CloseDAQDevice (void)**
int	**GetBoardNum (void)**
char*	**GetDllVersion(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.

GetDllVersion

This function tells the DLL version.

Char* GetDllVersion (void)

Parameters: None

Return Value:

The date of the installed DLL comes over.

LVDS(Camera Link) API Functions

Overview

BOOL	**LVDS_GetVersion (int nBoard, int *nVersion)**
BOOL	**LVDS_Init (int nBoard)**
BOOL	**LVDS_Close (int nBoard)**
BOOL	**LVDS_Start (int nBoard)**
BOOL	**LVDS_Stop (int nBoard)**
BOOL	**LVDS_SetDataMode (int nBoard, int nMode)**
BOOL	**LVDS_GetResolution (int nBoard, DWORD *xRes, DWORD *yRes)**
BOOL	**LVDS_GetFrame (int nBoard, DWORD* nCnt, unsigned char* buf)**
BOOL	**LVDS_GetFrameRate (int nBoard, int* nFrameRate)**
BOOL	**LVDS_GetDdrFrameRate (int nBoard, int* nFrameRate)**
BOOL	**LVDS_DdrEnable (int nBoard, BOOL bEn)**
BOOL	**LVDS_SetHsPol (int nBoard, BOOL bInv)**
BOOL	**LVDS_SetVsPol (int nBoard, BOOL bInv)**
BOOL	**LVDS_SetDeVal (int nBoard, BOOL bUse)**
BOOL	**LVDS_SetLeVal (int nBoard, BOOL bUse)**
BOOL	**LVDS_CC_Configure (int nBoard, int nCCcfg)**
BOOL	**LVDS_CC_Output (int nBoard, int nCCdata)**
BOOL	**LVDS_ConfigureTrig1 (int nBoard, DWORD dwEvent, DWORD dwDelay, DWORD dwWidth, DWORD dwBlank)**
BOOL	**LVDS_ConfigureTrig2 (int nBoard, DWORD dwEvent, DWORD dwDelay, DWORD dwWidth, DWORD dwBlank)**
BOOL	**LVDS_CameraMode (int nBoard, int nMode)**
BOOL	**LVDS_SetPageStart (int nBoard, DWORD dwEvent)**
BOOL	**LVDS_SetLineCount (int nBoard, DWORD dwCount)**
BOOL	**LVDS_SetPageDelay (int nBoard, DWORD dwCount)**
BOOL	**LVDS_SetReferrenceClock (int nBoard, int nClock)**
BOOL	**LVDS_ExtTrigEnable (int nBoard, BOOL bEn)**
BOOL	**LVDS_ExtTrigInv (int nBoard, BOOL bInv)**
BOOL	**LVDS_ExtTriConfigure (int nBoard, DWORD dwDelay, DWORD dwWidth, DWORD dwBlank)**
BOOL	**LVDS_EncAInv (int nBoard, BOOL bInv)**
BOOL	**LVDS_EncBInv (int nBoard, BOOL bInv)**
BOOL	**LVDS_EncZInv (int nBoard, BOOL bInv)**
BOOL	**LVDS_LineTrigInv (int nBoard, BOOL bInv)**
BOOL	**LVDS_PhaseTrigInv (int nBoard, BOOL bInv)**

BOOL LVDS_TrigOutInv (int nBoard, int nInv)

BOOL LVDS_VSyncBlankSet (int nBoard, int nBlank)

BOOL LVDS_TestPattern (int nBoard, BOOL bEnable int nData, int nClk,
DWORD nXres, DWORD nYres, DWORD nHsyncDelay,
DWORD nHsyncInterval, DWORD nVsyncInterval)

BOOL LVDS_FrameNumberEn (int nBoard, BOOL bEnable)

BOOL LVDS_FrameNumberClear (int nBoard)

BOOL LVDS_SetSyncCount (int nBoard, BOOL bEn, int nHsync, int nDE);

LVDS_GetVersion

This function gets the version of the current program.

BOOL LVDS_GetVersion (int nBoard, int *nVersion)

Parameters:

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

*nVersion : Current program version value.

Return Value :

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

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

LVDS_Init

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

BOOL LVDS_Init (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_Close

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_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 (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_Stop

This function stops the frame data capture.

BOOL LVDS_Stop (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

This function selects the frame (image) data mode.

BOOL LVDS_SetDataMode (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" : 32bit Mode, "3" : 64bit Mode,
"4" : 80bit Mode

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:

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_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 (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_GetFrameRate

This function gets the actual Frame Rate coming from the sensor or camera.

BOOL LVDS_GetFrameRate (int nBoard, DWORD *nFrameRate)

Parameters:

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

*nFrameRate : Actual Frame Rate value

Return Value:

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

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

LVDS_GetDdrFrameRate

This function gets the actual Frame Rate of the DDR memory.

BOOL LVDS_GetDdrFrameRate (int nBoard, DWORD *nFrameRate)

Parameters:

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

*nFrameRate : Actual Frame Rate value

Return Value:

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

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

LVDS_DdrEnable

This function selects DDR memory usage.

BOOL LVDS_DdrEnable (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 0”, DDR memory is not used,
If it is “1”, DDR memory is used.

Return Value :

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

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

LVDS_SetHsPol

This function selects a horizontal signal (Hsync: Horizontal Synchronization) signal line.

BOOL LVDS_SetHsPol (int nBoard, BOOL bInv)

Parameters:

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

bInv : "If it is 0", Normal HSync is used,
and if it is "1", Inverse HSync is used.

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 (int nBoard, BOOL bInv)

Parameters:

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

bInv : If it is "0", it uses Normal Vsync,
and if it is "1", it uses Inverse VSync.

Return Value :

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

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

LVDS_SetDeVal

This function uses the Data Valid signal on the Line Valid signal line.

BOOL LVDS_SetDeVal (int nBoard, BOOL bUse)

Parameters:

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

bUse : If it is "0", DVAL signal line is used for DVAL signal line.
If it is "1", DVAL signal line is used for LVAL signal line.

Return Value :

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

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

LVDS_SetLeVal

This function uses Line Valid signal for Data Valid signal line.

BOOL LVDS_SetLeVal (int nBoard, BOOL bUse)

Parameters:

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

bUse : If it is "0", LVAL signal line is used for LVAL signal line.
If it is "1", LVAL signal line is used for DVAL signal line.

Return Value :

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

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

LVDS_CC_Configure

This function outputs the CC signal if the data of the lower 4 bits is "0", and outputs the corresponding Trigger or External Clock if it is "1".

BOOL LVDS_CC_Configure (int nBoard, DWORD nCCcfg)

Parameters:

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

nCCcfg : bit0(CC1 configure) = "0" : CC1 out / "1" : Trigger1 output
bit1(CC2 configure) = "0" : CC2 out / "1" : Trigger2 output
bit2(CC3 configure) = "0" : CC3 out / "1" : Reserve
bit3(CC4 configure)= "0" : CC4 out / "1" : Reference Clock Out
others : Reserved

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 nCCdata)

Parameters:

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

nCCdata: "bit0" : CC1 out
"bit1" : CC2 out
"bit2" : CC3 out
"bit3" : CC4 out
"others" : Reserved

Return Value :

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

LVDS_ConfigureTrig1

This function selects trigger 1's input mode selection, output delay, output width, and output blank.

BOOL LVDS_ConfigureTrig1 (int nBoard, DWORD dwEvent,
DWORD dwDelay, DWORD dwWidth, DWORD dwBlank)

Parameters:

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

dwEvent : "0" : Continuous (Free running without any condition)
"1" : Rising edge on Line Trigger input
"2" : Rising edge on encoder A phase
"3" : Rising edge on encoder B phase
"4" : Rising edge encoder up clock using A and B phase
"5" : Rising edge encoder down clock A and B phase

dwDelay : number of reference clocks required before output Trig1 goes high

dwWidth : number of reference clocks required for output Trig1 to go high

dwBlank : number of reference clocks required before output Trig1 returns from delay

Return Value :

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

LVDS_ConfigureTrig2

This function selects trigger 2's input mode selection, output delay, output width, and output blank.

BOOL LVDS_ConfigureTrig2 (int nBoard, DWORD dwEvent,
DWORD dwDelay, DWORD dwWidth, DWORD dwBlank)

Parameters:

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

dwEvent : "0" : Continuous (Free running without any condition)
"1" : Rising edge on Line Trigger input
"2" : Rising edge on encoder A phase
"3" : Rising edge on encoder B phase
"4" : Rising edge encoder up clock using A and B phase
"5" : Rising edge encoder down clock A and B phase

dwDelay : number of reference clocks required before output Trig2 goes high

dwWidth : number of reference clocks required for output Trig2 to go high

dwBlank : number of reference clocks required before output Trig2 returns from delay

Return Value :

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

LVDS_CameraMode

This function selects whether the camera mode is Area Line Scan Camera or Line Scan Camera.

BOOL LVDS_CameraMode (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 (Default)
"Others" : Line Scan Camera.

Return Value :

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

LVDS_SetPageStart

This function selects the page initial event.

BOOL LVDS_SetPageStart (int nBoard, DWORD dwEvent)

Parameters:

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

dwEvent : "0" : Continuous (Free Running)
"1" : Rising edge in Page Trigger input
"2" : Rising edge on encoder z phase

Return Value :

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

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

LVDS_SetLineCount

This function selects the number of lines needed to get one image in one page.

BOOL LVDS_SetLineCount (int nBoard, DWORD dwCount)

Parameters:

Parameters:

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

dwCount : 1 ~ 65535 (Number of horizontal lines in Line Scan Camera)

Return Value :

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

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

LVDS_PageDelay

This function selects the number of clocks needed before acquiring one image as a page.

BOOL LVDS_SetLineCount (int nBoard, DWORD dwCount)

Parameters:

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

dwCount : 1 ~ 15

Return Value :

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

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

LVDS_SetReferenceClock

This function selects what CC4's reference clock is to be used.

BOOL LVDS_SetReferenceClock (int nBoard, int nClcok)

Parameters:

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

nClock : If it is “0”, the internal clock in the frame grabber is used (Default)
If it is “Others”, the external clock provided by the encoder or other board is used.

Return Value :

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

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

LVDS_ExtTrigEnable

This function selects an external trigger signal.

BOOL LVDS_ExtTrigEnable (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 "True", an external trigger signal is used and
If "False", internal trigger signal is used.

Return Value :

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

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

LVDS_ExtTrigInv

This function inverts the external trigger signal.

BOOL LVDS_ExtTrigInv (int nBoard, BOOL bInv)

Parameters:

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

bInv : If "True", the external trigger signal is inversely converted and used.
If "False", just use the external trigger signal.

Return Value :

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

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

LVDS_ExtTrigConfigure

This function configures the external trigger signal.

BOOL LVDS_ExtTrigConfigure (int nBoard, DWORD dwDelay, DWORD dwWidth, DWORD dwBlank)

Parameters:

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

dwDelay : the number of reference clocks required before the external trigger signal goes high.

dwWidth : the number of reference clocks required for the external trigger signal to be high.

dwBlank : The number of reference clocks required before the external trigger signal returns from the delay.

Return Value :

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

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

LVDS_EncAInv

This function inverts the encoder A signal.

BOOL LVDS_EncAInv (int nBoard, BOOL bInv)

Parameters:

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

bInv : If it is "True", the Encoder A signal is inversely transformed and used.
If "False", just use Encoder A signal.

Return Value :

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

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

LVDS_EncBInv

This function inverts the encoder B signal.

BOOL LVDS_EncBInv (int nBoard, BOOL bInv)

Parameters:

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

bInv : If it is "True", the Encoder B signal is inversely transformed and used.
If "False", just use Encoder B signal.

Return Value :

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

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

VDS_EncZInv

This function inverts the encoder Z signal.

BOOL LVDS_EncZInv (int nBoard, BOOL bInv)

Parameters:

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

bInv : If it is "True", the Encoder Z signal is inversely transformed and used.
If "False", just use Encoder Z signal.

Return Value :

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

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

LVDS_LineTrigInv

This function inverts the Line Trigger signal.

BOOL LVDS_LineTrigInv (int nBoard, BOOL bInv)

Parameters:

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

bInv : If "True", the Line Trigger signal is inversely converted and used.
If "False", just use the Line Trigger signal.

Return Value :

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

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

LVDS_PhaseTrigInv

This function inverts the phase trigger signal.

BOOL LVDS_PhaseTrigInv (int nBoard, BOOL bInv)

Parameters:

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

bInv : If "True", the phase Trigger signal is inversely converted and used.
If "False", just use the phase Trigger signal.

Return Value :

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

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

LVDS_TrigOutInv

This function inverts the CC output waveform.

BOOL LVDS_TrigOutInv (int nBoard, int nInv)

Parameters:

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

nInv : "0" : CC1, CC2 Normal Signal
"1" : CC1 Inverse Signal
"2" : CC2 Inverse Signal
"Others" : CC1, CC2 Inverse Signal

Return Value :

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

LVDS_VsyncBlankSet

This function determines what value to use as the blank of the Vsync signal.

BOOL LVDS_VsyncBlankSet (int nBoard, int nBlank)

Parameters:

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

nBlank : Blank interval (0 ~ 65535) of Vsync

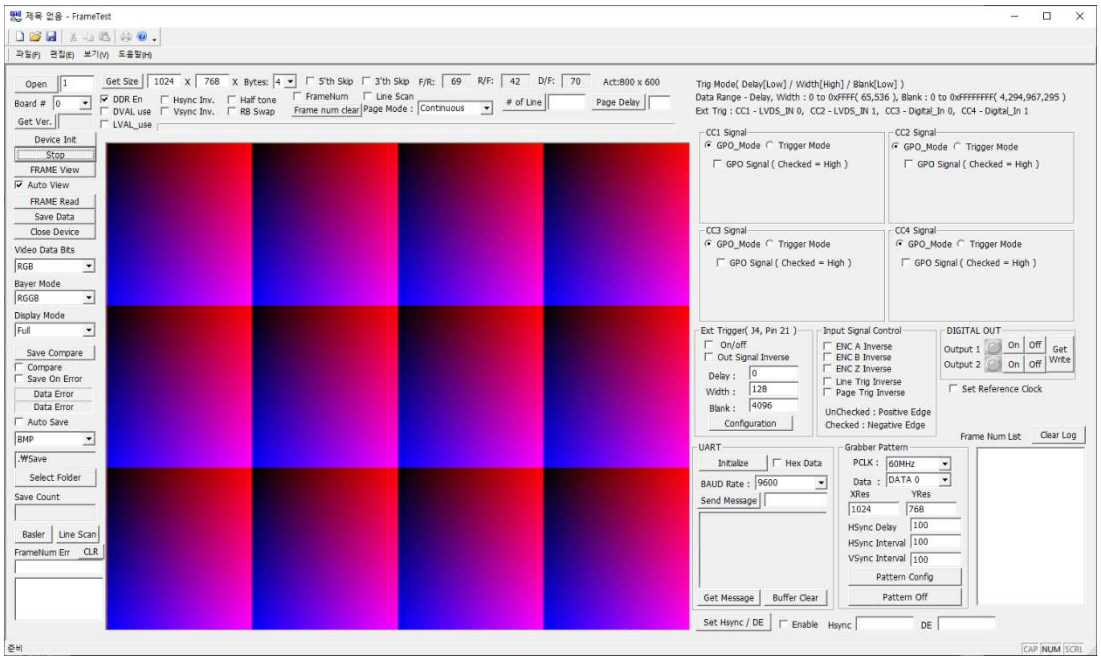
Return Value :

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

LVDS_TestPattern

This function determines which value to set the signal of the test pattern to.

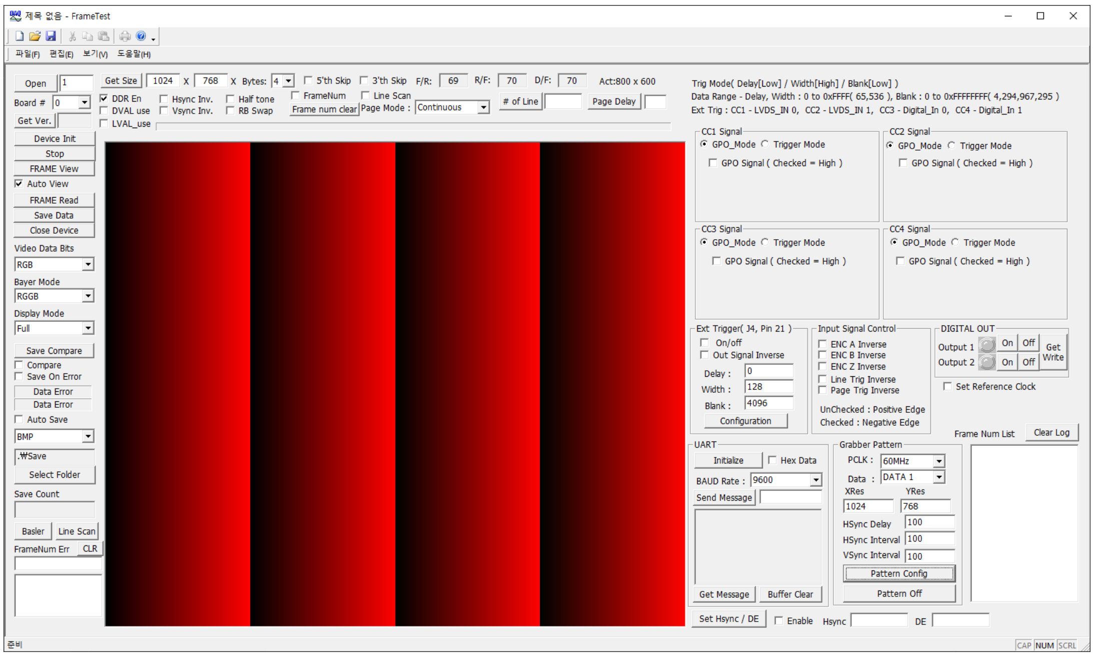
In the case of Data 0, in 4Bytes unit, upper 2bytes is Line Counter, lower 2bytes is Pixel Counter, Pixel Counter increases from 0 to horizontal resolution value, and Line Counter increases from 0 to vertical resolution.



Trig Mode(Delay[Low] / Width[High] / Blank[Low])
Data Range - Delay, Width : 0 to 0xFFFF(65,536), Blank : 0 to 0xFFFFFFFF(4,294,967,295)
Ext Trig : CC1 - LVDS_IN 0, CC2 - LVDS_IN 1, CC3 - Digital_In 0, CC4 - Digital_In 1

Get Size: 1024 X 768 X Bytes: 4, F/R: 69, R/F: 42, D/F: 70, Act:800 x 600
Page Mode : Continuous
Video Data Bits: RGB
Bayer Mode: RGGB
Display Mode: Full
Delay : 0
Width : 128
Blank : 4096
BAUD Rate : 9600
PCLK : 60MHz
Data : DATA 0
XRes: 1024
YRes: 768
HSync Delay: 100
HSync Interval: 100
VSync Interval: 100
UnChecked : Positive Edge
Checked : Negative Edge

In the case of Data 1, it is a pixel counter in units of 4 bytes, and the value ranges from 0 to horizontal resolution. When outputting in RGB, it is like the image below.



Trig Mode(Delay[Low] / Width[High] / Blank[Low])
Data Range - Delay, Width : 0 to 0xFFFF(65,536), Blank : 0 to 0xFFFFFFFF(4,294,967,295)
Ext Trig : CC1 - LVDS_IN 0, CC2 - LVDS_IN 1, CC3 - Digital_In 0, CC4 - Digital_In 1

Get Size: 1024 X 768 X Bytes: 4, F/R: 69, R/F: 70, D/F: 70, Act:800 x 600
Page Mode : Continuous
Video Data Bits: RGB
Bayer Mode: RGGB
Display Mode: Full
Delay : 0
Width : 128
Blank : 4096
BAUD Rate : 9600
PCLK : 60MHz
Data : DATA 1
XRes: 1024
YRes: 768
HSync Delay: 100
HSync Interval: 100
VSync Interval: 100
UnChecked : Positive Edge
Checked : Negative Edge

BOOL LVDS_Tesrpattern (int nBoard, BOOL bEnable, int nData, int nClk,
DWORD nXres, DWORD nYres, DWORD nHsyncDelay,
DWORD nHsyncInterval, DWORD nVsyncInterval)

Parameters:

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

bEnable : "1: If it is, the Test Pattern is executed.

nData : If "0" is Data 0, if "1" is Data 1 Pattern

nClk : "0", Camera Link PCLK0, "1", 60MHz,
"2" is 75 MHz, "3" is 85 MHz

nXres : Horizontal resolution value

nYres : Vertical resolution value

nHsyncDelay : Horizontal Sync Delay value

nHsyncInterrval: Horizontal Sync Interval value

nVsyncIntreval : Vertical Sync Interval value

Return Value :

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

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

LVDS_FrameNumberEn

This function adds Frame Count to the end of the frame line.

BOOL LVDS_FrameNumberEn (int nBoard, BOOL bEnanle)

Parameters:

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

bEnable : If "True", Frame Number On / Y Resolution + 1
ex) Resolution = 1024 x 768 => 1024 x 769
If "False", Frame Number Off

Return Value :

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

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

LVDS_FrameNumberClear

This function initializes the frame count value.

BOOL LVDS_FrameNumberClear (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_SetSyncCount

This function detects the rising edge of Hsync or DE (Data Valid) and forcibly determines the desired resolution in order to solve the problem if there is an error in Hsync or DE (Data Valid) during the input signal of the camera.

BOOL LVDS_SetSyncCount (int nBoard, BOOL bEn, int nHsync, int nDE)

Parameters:

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

bEn : If "True", the function is applied, if "False", the function is not applied.

Hsync : Horizontal Sync Value

nDe : Data Valid Value

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

UART_Init

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

BOOL **UART_Init (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

This function receives characters through the differential UART.

BOOL **UART_GetDatal (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

This function sends characters through the differential UART.

BOOL UART_SendData (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

This function releases all resource were used for UART function.

BOOL UART_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".

UART_SetBaud

This function sets UART Baud rates.

BOOL UART_SetBaud (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

This function flushes UART RX Buffer

BOOL UART_BufferFlush (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".

Digital Input/Output API Functions

Overview

BOOL	**DIO_Read (int nBoard, BOOL *bIn)**
BOOL	**DIO_Write (int nBoard, BOOL bOut)**
BOOL	**DIO_GetWrite (int nBoard, BOOL *bOut)**

DIO_Read

This function reads the value of Pin 17 DIGITAL_IN port of J4 connector.

DWORD **DIO_Read (int nBoard, BOOL *bIn)**

Parameters:

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

*bin : Digital register value.

Return Value:

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

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

DIO_Write

This function outputs the value to Pin 15 DIGITAL_OUT port of J4 connector.

BOOL **DIO_Write (int nBoard, BOOL bOut)**

Parameters:

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

bOut : The value to write to the digital buffer.

Return Value:

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

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

DIO_GetWrite

This function reads and checks the value of the output port.

BOOL DIO_GetWrite (int nBoard, BOOL* bOut)

Parameters:

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

*bOut : True if the value of the output port is normal, otherwise False.

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