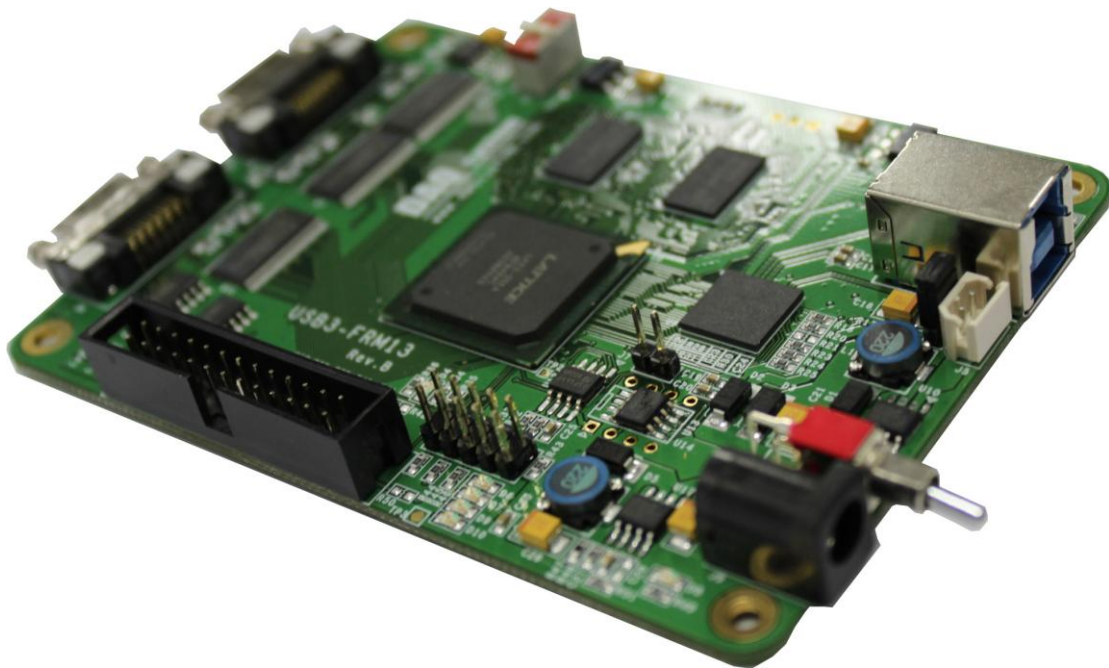


USB3-FRM13 API Programming (Rev 1.2)



Windows, Windows2000, Windows NT and Windows XP are trademarks of **Microsoft**. We acknowledge that the trademarks or service names of all other organizations mentioned in this document as their own property.

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.

Copyrights © 2010 DAQ system, All rights reserved.

Board Level API Functions

Overview

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

OpenDAQDevice

디바이스를 Open한다.

프로그램에서 초기에 반드시 한번 함수를 호출하여 디바이스를 Open하여야 한다.

Int OpenDAQDevice (void)

Parameters:

없음.

Return Value:

함수 호출에 성공한 경우, 설치된 보드의 개수를 리턴한다.

(멀티 보드인 경우 최대 4까지 가능)

함수 호출에 실패한 경우, "0"을 리턴한다. 이것의 의미는 시스템에 장치가 없다는 의미이다.

ResetBoard

현재 시스템(PC)에 장착된 디바이스를 초기화 한다.

BOOL ResetBoard (int nBoard)

Parameters:

nBoard :현재 시스템에 장착되어 있는 보드 번호를 알려준다.

보드 번호는 보드의 DIP 스위치를 이용하여 설정한다.

Return Value:

함수 호출에 성공일 경우 "TRUE"을 리턴함.

함수 호출에 실패한 경우, 어떤 API함수도 호출할 수 없고 CloseDAQDevice()를 부른다.

CloseDAQDevice

Open된 디바이스를 Close한다.

장치의 사용이 끝나게 되면, 반드시 장치를 Close하여 다른 프로그램에서 사용할 수 있도록 한다.

BOOL **CloseDAQDevice (void)**

Parameters:

없음.

Return Value:

디바이스 Open에 성공할 경우 TRUE를 실패할 경우 FALSE를 리턴한다.

GetBoardNum

시스템에서 오픈된 후 보드의 개수를 알려준다.

int **GetBoardNum (void)**

Parameters:

없음.

Return Value:

보드의 개수. (멀티 보드인 경우 최대 4까지 가능)

IsConnected

USB에 연결이 되어 있는지를 알려준다.

BOOL **IsConnected (void)**

Parameters:

없음.

Return Value:

함수 호출에 실패한 경우 'FALSE', 성공일 경우 'TRUE'를 return한다.

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_GetResolution (DWORD *xRes, DWORD *yRes)
BOOL	LVDS_Stop (void)
BOOL	LVDS_SetDataMode (int nMode)
BOOL	LVDS_GetVersion (int *nFpgaVer, int *nFirmVer)
DWORD	LVDS_GetError (DWORD *dwStatuse)
BOOL	LVDS_BufferFlush (void)
BOOL	LVDS_SetDeUse (BOOL bUse)
BOOL	LVDS_SetHsPol (BOOL bPol)
BOOL	LVDS_CameraMode (int nMode)
BOOL	LVDS_SetReferenceClock (int nClock)
BOOL	LVDS_ConfigureCc (DWORD dwCFG)
BOOL	LVDS_CcOutput (DWORD dwCC)
BOOL	LVDS_SetLineCount (DWORD dwCount)
BOOL	LVDS_SetPageStart (DWORD dwEvent)
BOOL	LVDS_SetPageDelay (DWORD dwCount)
BOOL	LVDS_ConfigureTrig1 (DWORD dwEvent, DWORD dwDelay, DWORD dwWidth, DWORD dwBlank)
BOOL	LVDS_ConfigureTrig2 (DWORD dwEvent, DWORD dwDelay, DWORD dwWidth, DWORD dwBlank)
BOOL	LVDS_SetTrigger (BOOL bUse)
BOOL	LVDS_SetTrigMode (BOOL bUse)
BOOL	LVDS_SetPoCLDelay (int nMode)

LVDS_Init

LVDS sub-system의 자원, 예를 들어 interrupt와 LVDS control register을 초기화한다.

BOOL LVDS_Init (void)

Parameters: 없음.

Return Value:

함수 호출에 실패한 경우 'FALSE', 성공일 경우 'TRUE'를 return한다.

LVDS_Start

프레임 데이터 수집을 시작한다. 이 함수를 호출한 후에 LVDS_GetFrame 함수를 호출하여 데이터가 완성이 되었는가를 확인할 수 있다.

BOOL LVDS_Start (void)

Parameters: 없음.

Return Value:

함수 호출에 실패한 경우 'FALSE', 성공일 경우 'TRUE'를 return한다.

LVDS_GetFrame

프레임 데이터가 완성이 되었는가를 검사하고 완성이 되었으면 프레임 데이터를 가져온다. 이때 데이터를 받아올 버퍼 크기를 반드시 알려주어야 한다.

BOOL LVDS_GetFrame (DWORD* nCnt, unsigned char* buf)

Parameters:

nCnt : 프레임 데이터를 받아 올 버퍼의 크기가 저장 되어있는 변수의 번지이다.

함수를 호출할때 버퍼크기를 지정하고 호출한 후에는 변수 값을 읽어서 실제로 읽어 온 개수를 확인한다. 데이터 크기는 바이트 단위이다.

buf : 프레임 버퍼 포인터.

Return Value:

함수 호출에 실패할 경우 "FALSE" 성공일 경우 "TRUE"을 리턴함.

호출이 성공하면 nCnt 값을 확인하여 원하는 크기 만큼 읽어 졌는가를 확인한다.

(주) 프레임 데이터가 완성이 되지 않았으면 바로 FALSE를 리턴하고 nCnt 값을 0으로 해서 리턴이 일어난다.

LVDS_Close

LVDS 함수에서 사용된 모든 자원을 되돌려준다. 어플리케이션 프로그램은 프로그램 종료 시 이 함수를 부른다.

BOOL LVDS_Close (void)

Parameters: 없음.

Return Value:

함수 호출에 실패한 경우 'FALSE', 성공일 경우 'TRUE'를 return한다.

LVDS_GetResolution

Video 입력의 해상도를 가져온다.

BOOL LVDS_GetResolution (DWORD *xRes, DWORD *yRes)

Parameters:

*xRes : 수평해상도 값 즉, Frame의 Width를 가져온다.

*yRes : 수직해상도 값 즉, Frame의 Height를 가져온다.

Return Value:

함수 호출에 실패한 경우 'FALSE', 성공일 경우 'TRUE'를 return한다.

LVDS_Stop

프레임 데이터 Capture를 중지한다.

BOOL LVDS_Stop (void)

Parameters: 없음.

Return Value :

함수 호출에 실패한 경우 'FALSE', 성공일 경우 'TRUE'를 return한다.

LVDS_SetDataMode

프레임(이미지) 데이터 모드를 선택한다.

BOOL LVDS_SetDataMode (int nMode)

Parameters:

nMode : “0” 이면 8bit Mode이고
“1” 이면 16bit Mode이고
“2” 이면 32bit Mode이고
“3” 이면 64bit Mode이다.

Return Value :

함수 호출에 실패한 경우 'FALSE', 성공일 경우 'TRUE'를 return한다.

LVDS_GetVersion

현재 프로그램의 버전을 가져온다.

BOOL LVDS_GetVersion (int *nFpgaVer, int *nFirmVer)

Parameters:

*nFpgaVer : 현재 FPGA 버전 값.
*nFirmVer : 현재 Firmware 버전 값.

Return Value :

함수 호출에 실패한 경우 'FALSE', 성공일 경우 'TRUE'를 return한다.

LVDS_GetError

프레임(이미지) 에러를 가져온다.

DWORD LVDS_GetError (DWORD *dwStatus)

Parameters:

*dwStatus : 에러 상태 값을 가져온다.
“1” 이면 Overflow error
“2” 이면 Read error
“4” 이면 Size error

Return Value :

함수 호출에 실패한 경우 'FALSE', 성공일 경우 'TRUE'를 return한다.

LVDS_BufferFlush

버퍼를 초기화 한다.

BOOL LVDS_BufferFlush (void)

Parameters: 없음.

Return Value :

함수 호출에 실패한 경우 'FALSE', 성공일 경우 'TRUE'를 return한다.

LVDS_SetDeUse

데이터 유효(Data Valid) 신호선을 선택한다.

BOOL LVDS_SetDUse (BOOL bUse)

Parameters:

bUse : "True" 이면 DVAL(Data Validation) 사용하고

"False" 이면 HSYNC (Horizontal Synchronization)를 사용한다

Return Value :

함수 호출에 실패한 경우 'FALSE', 성공일 경우 'TRUE'를 return한다.

LVDS_SetHsPol

수평 신호(HSYNC : Horizontal Synchronization) 신호선을 선택한다.

BOOL LVDS_SetHsPol (BOOL bPol)

Parameters:

bPol : "True" 이면 Normal HSYNC 사용하고

"False" 이면 Inverse HSYNC를 사용한다

Return Value :

함수 호출에 실패한 경우 'FALSE', 성공일 경우 'TRUE'를 return한다.

LVDS_CameraMode

카메라 모드가 Area Line Scan Camera인지 Line Scan Camera 인지를 선택한다.

BOOL LVDS_CameraMode (int nMode)

Parameters:

nMode : “0” 이면 Area Scan Camera 이고 (Default)
“Others” 이면 Line Scan Camera 이다

Return Value :

함수 호출에 실패한 경우 ‘FALSE’, 성공일 경우 ‘TRUE’를 return한다.

LVDS_SetReferenceClock

레퍼런스 클럭을 무엇으로 사용할 것인지를 선택한다.

BOOL LVDS_SetReferenceClock (int nClcok)

Parameters:

nClock : “0” 이면 프레임 그레버내의 Internal Clcok 을 사용하고 (Default)
“Others” 이면 엔코더나 다른 보드에서 제공되는 External Clock을 사용한다.

Return Value :

함수 호출에 실패한 경우 ‘FALSE’, 성공일 경우 ‘TRUE’를 return한다.

LVDS_ConfigureCc

해당 비트의 환경설정 값을 선택한다.

BOOL LVDS_ConfigureCc (DWORD dwCFG)

Parameters:

dwCFG : bit0(CC1 configure) = “0” : digital out1 / “1”: alternate (Trigger1 output)
bit1(CC2 configure) = “0” : digital out2 / “1”: alternate (Trigger2 output)
bit2(CC3 configure) = “0” : digital out3 / “1”: (Digital output)
bit3(CC4 configure)= “0” : digital out4 / “1”: alternate (Reference clock output)
others : Reserved

Return Value :

함수 호출에 실패한 경우 ‘FALSE’, 성공일 경우 ‘TRUE’를 return한다.

LVDS_CcOutput

해당 비트의 CC 값을 출력한다.

BOOL LVDS_CcOutput (DWORD dwCC)

Parameters:

dwCC : bit0(CC1 out) = "0" : output "0" / "1": output "1"
bit1(CC2 out) = "0" : output "0" / "1": output "1"
bit2(CC3 out) = "0" : output "0" / "1": output "1"
bit3(CC4 out) = "0" : output "0" / "1": output "1"
others : Reserved

Return Value :

함수 호출에 실패한 경우 'FALSE', 성공일 경우 'TRUE'를 return한다.

LVDS_SetLineCount

한 페이지에서 하나의 이미지를 얻기 위해 필요한 라인의 수를 선택한다.

BOOL LVDS_SetLineCount (DWORD dwCount)

Parameters:

dwCount : 1 ~65535

Return Value :

함수 호출에 실패한 경우 'FALSE', 성공일 경우 'TRUE'를 return한다.

LVDS_SetPageStart

페이지 초기 이벤트를 선택한다.

BOOL LVDS_SetPageStart (DWORD dwEvent)

Parameters:

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

Return Value :

함수 호출에 실패한 경우 'FALSE', 성공일 경우 'TRUE'를 return한다.

LVDS_SetPageDelay

한 페이지에서 하나의 이미지를 얻기 전에 필요한 클럭의 수를 선택한다.

BOOL LVDS_SetPageDelay (DWORD dwCount)

Parameters:

dwCount : 1 ~ 15

Return Value :

함수 호출에 실패한 경우 'FALSE', 성공일 경우 'TRUE'를 return한다.

LVDS_ConfigureTrig1

트리거1의 입력 모드 선택과 출력 지연, 출력 폭, 출력 공백을 선택한다.

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

Parameters:

dwEvent : "0" 이면 Continuous (Free running without any condition)

"1" 이면 Rising edge on Line Trigger input

"2" 이면 Rising edge on encoder A phase

"3" 이면 Rising edge on encoder B phase

"4" 이면 Rising edge encoder up clock using A and B phase

"5" 이면 Rising edge encoder down clock A and B phase

dwDelay : 출력 Trig1이 high가 되기 전에 필요한 레퍼런스 클럭 수

dwWidth : 출력 Trig1이 high에 필요한 레퍼런스 클럭 수

dwBlank : 출력 Trig1이 지연에서 복귀 전에 필요한 레퍼런스 클럭 수

Return Value :

함수 호출에 실패한 경우 'FALSE', 성공일 경우 'TRUE'를 return한다.

LVDS_SetTrigMode

Trigger 모드를 선택한다..

BOOL LVDS_SetTrigMode (BOOL bUse)

Parameters:

bUse : “True” 이면 Trigger Edge Mode
“False” 이면 Trigger Level Mode

Return Value :

함수 호출에 실패한 경우 ‘FALSE’, 성공일 경우 ‘TRUE’를 return한다.

LVDS_SetPoCLDelay

PoCL(Power over Camera Link) 전원을 사용할 때 지연시간(Delay time)을 선택할 수 있다. (DEFAULT 는 0.5sec이다.)

BOOL LVDS_SetPoCLDelay (int nMode)

Parameters:

nMode : “0” : 0.5sec, “1” : 1sec, “2” : 1.5sec, “3” : 2sec
“4” : 2.5sec, “5” : 3sec, “6” : 3.5sec, “7” : 4sec

Return Value :

함수 호출에 실패한 경우 ‘FALSE’, 성공일 경우 ‘TRUE’를 return한다.

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

UART sub-system의 자원, 예를 들어 interrupt와 UART control register를 초기화 한다.

BOOL **UART_Init (void)**

Parameters:

없음.

Return Value:

함수 호출에 실패한 경우 'FALSE', 성공일 경우 'TRUE'를 return한다.

UART_GetData

차동 UART를 통하여 문자를 받는다.

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

Parameters:

nCnt : 문자를 나타내는 어드레스를 바이트 사이즈로 받는다.

최대 받을 수 있는 문자는 1000 bytes 로 제한되어 있다.

buf : 버퍼 어드레스.

Return Value:

함수 호출에 실패한 경우 'FALSE', 성공일 경우 'TRUE'를 return한다.

UART_SendData

자동 UART를 통하여 문자를 보낸다.

BOOL **UART_SendData (DWORD* nCnt, unsigned char* buf)**

Parameters:

nCnt : 문자를 나타내는 어드레스를 바이트 사이즈로 받는다.

최대 전송할 수 있는 문자는 1000 bytes 로 제한되어 있다.

buf : 버퍼 어드레스.

Return Value:

함수 호출에 실패한 경우 'FALSE', 성공일 경우 'TRUE'를 return한다.

UART_Close

UART 함수에서 사용한 모든 자원을 되돌려 준다.

BOOL **UART_Close (void)**

Parameters:

없음.

Return Value:

함수 호출에 실패한 경우 'FALSE', 성공일 경우 'TRUE'를 return한다.

UART_SetBaud

UART Baud를 정한다.

BOOL **UART_SetBaud (DWORD nBaud)**

Parameters:

nBaud : 0: 9600, 1: 19200, 2: 38400, 3:57600, 4:115200bps

Return Value:

함수 호출에 실패한 경우 'FALSE', 성공일 경우 'TRUE'를 return한다.

UART_BufferFlush

UART Rx 버퍼를 clear 한다.

BOOL **UART_BufferFlush (void)**

Parameters:

없음.

Return Value:

함수 호출에 실패한 경우 'FALSE', 성공일 경우 'TRUE'를 return한다.

CC(Camera Control) API Functions

Overview

BOOL **DIO_Write (DWORD dwVal)**

DIO_Write

원하는 카메라 제어 값을 출력포트에 출력한다. (카메라 사양 참조)

BOOL **DIO_Write (DWORD dwVal)**

Parameters:

dwVal : 출력 포트에 기록할 값.

Return Value:

함수 호출에 실패한 경우 'FALSE', 성공일 경우 'TRUE'를 return한다.

Multi Board support APIs

싱글 보드의 API 인 경우 단지 하나의 보드가 인스톨된 시스템에서 사용된다. 하지만, 두 개의 보드이거나 그 이상(최대 4개 지원)의 보드가 인스톨 된 시스템에서는 다중 API 를 사용해야만 한다.

Multi-Board LVDS(Camera Link) API Functions

Overview

BOOL	IsConnected_Mul (int nBoard)
BOOL	LVDS_Init_Mul (int nBoard)
BOOL	LVDS_Check_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_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_GetError_Mul (int nBoard, DWORD *dwStatus)
BOOL	LVDS_BufferFlush_Mul (int nBoard)
BOOL	LVDS_SetDeUse_Mul (int nBoard, BOOL bUse)
BOOL	LVDS_SetHsPol_Mul (int nBoard, BOOL bPol)
BOOL	LVDS_CameraMode_Mul (int nBoard, int nMode)
BOOL	LVDS_SetReferenceClock_Mul (int nBoard, int nClock)
BOOL	LVDS_ConfigureCc_Mul (int nBoard, DWORD dwCFG)
BOOL	LVDS_CcOutput_Mul (int nBoard, DWORD dwCC)
BOOL	LVDS_SetLineCount_Mul (int nBoard, DWORD dwCount)
BOOL	LVDS_SetPageStart_Mul (int nBoard, DWORD dwEvent)
BOOL	LVDS_SetPageDelay_Mul (int nBoard, DWORD dwCount)
BOOL	LVDS_ConfigureTrig1_Mul (int nBoard, DWORD dwEvent, DWORD dwDelay, DWORD dwWidth, DWORD dwBlank)
BOOL	LVDS_ConfigureTrig2_Mul (int nBoard, DWORD dwEvent, DWORD dwDelay, DWORD dwWidth, DWORD dwBlank)
BOOL	LVDS_SetTrigger_Mul (int nBoard, BOOL bUse)
BOOL	LVDS_SetTrigMode_Mul (int nBoard, BOOL bUse)
BOOL	LVDS_SetPoCLDelay_Mul (int nBoard, int nMode)

IsConnected_Mul

USB에 연결이 되어 있는지를 알려준다.

BOOL IsConnected_Mul (int nBoard)

Parameters:

nBoard : 현재 시스템에 장착되어 있는 보드 번호이다.

보드 번호는 보드의 DIP 스위치 설정(0~ 3)과 일치한다.

Return Value:

함수 호출에 실패한 경우 'FALSE', 성공일 경우 'TRUE'를 return한다.

LVDS_Init_Mul

LVDS Sub-system의 자원, 예를 들어 Interrupt와 LVDS control register을 초기화한다.

BOOL LVDS_Init_Mul (int nBoard)

Parameters:

nBoard : 현재 시스템에 장착되어 있는 보드 번호이다.

보드 번호는 보드의 DIP 스위치 설정(0~ 3)과 일치한다.

Return Value:

함수 호출에 실패한 경우 'FALSE', 성공일 경우 'TRUE'를 return한다.

LVDS_Check_Mul

받은 프레임 데이터의 완료를 체크함.

BOOL LVDS_Check_Mul (int nBoard)

Parameters:

nBoard : 현재 시스템에 장착되어 있는 보드 번호이다.

보드 번호는 보드의 DIP 스위치 설정(0~ 3)과 일치한다.

Return Value:

함수 호출에 실패한 경우 'FALSE', 성공일 경우 'TRUE'를 return한다.

LVDS_Start_Mul

프레임 데이터 수집을 시작한다. 이 함수를 호출한 후에 LVDS_GetFrame 함수를 호출하여 데이터가 완성이 되었는가를 확인할 수 있다.

BOOL LVDS_Start_Mul (int nBoard)

Parameters:

nBoard : 현재 시스템에 장착되어 있는 보드 번호이다.
보드 번호는 보드의 DIP 스위치 설정(0~ 3)과 일치한다.

Return Value:

함수 호출에 실패한 경우 'FALSE', 성공일 경우 'TRUE'를 return한다.

LVDS_GetFrame_Mul

프레임 데이터가 완성이 되었는가를 검사하고 완성이 되었으면 프레임 데이터를 가져온다. 이때 데이터를 받아올 버퍼 크기를 반드시 알려주어야 한다.

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

Parameters:

nBoard : 현재 시스템에 장착되어 있는 보드 번호이다.
보드 번호는 보드의 DIP 스위치 설정(0~ 3)과 일치한다.
nCnt : 프레임 데이터를 받아 올 버퍼의 크기가 저장 되어있는 변수의 번지이다.
함수를 호출할때 버퍼크기를 지정하고 호출한 후에는 변수 값을 읽어서 실제로 읽어 온 개수를 확인한다. 데이터 크기는 바이트 단위이다.
buf : 프레임 버퍼 포인터.

Return Value:

함수 호출에 실패한 경우 'FALSE', 성공일 경우 'TRUE'를 return한다.

LVDS_Close_Mul

LVDS 함수에서 사용된 모든 자원을 되돌려준다. 어플리케이션 프로그램은 프로그램 종료 시 이 함수를 부른다.

BOOL LVDS_Close (int nBoard)

Parameters:

nBoard : 현재 시스템에 장착되어 있는 보드 번호이다.
보드 번호는 보드의 DIP 스위치 설정(0~ 3)과 일치한다.

Return Value:

함수 호출에 실패한 경우 'FALSE', 성공일 경우 'TRUE'를 return한다.

LVDS_GetResolution_Mul

Video 입력의 해상도를 가져온다.

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

Parameters:

nBoard : 현재 시스템에 장착되어 있는 보드 번호이다.
보드 번호는 보드의 DIP 스위치 설정(0~ 3)과 일치한다.
*xRes : 수평해상도 즉, Frame의 Width를 가져온다.
*yRes : 수직해상도 즉, Frame의 Height를 가져온다.

Return Value:

함수 호출에 실패한 경우 'FALSE', 성공일 경우 'TRUE'를 return한다.

LVDS_Stop_Mul

BOOL LVDS_Stop_Mul (int nBoard)

프레임 데이터 Capture를 중지한다.

Parameters:

nBoard : 현재 시스템에 장착되어 있는 보드 번호이다.
보드 번호는 보드의 DIP 스위치 설정(0~ 3)과 일치한다.

Return Value :

함수 호출에 실패한 경우 'FALSE', 성공일 경우 'TRUE'를 return한다.

LVDS_SetDataMode_Mul

이미지 픽셀 데이터 모드를 설정한다.

BOOL LVDS_SetDataMode_Mul (int nBoard, int nMode)

Parameters:

- nBoard : 현재 시스템에 장착되어 있는 보드 번호이다.
보드 번호는 보드의 DIP 스위치 설정(0~ 3)과 일치한다.
- nMode : “1” 이면 8bit Mode이고
“2” 이면 16bit Mode이고
“3” 이면 32bit Mode이고
“4” 이면 64bit Mode이다.

Return Value:

함수 호출에 실패한 경우 'FALSE', 성공일 경우 'TRUE'를 return한다.

LVDS_GetVersion_Mul

현재 FPGA version을 가져온다.

BOOL LVDS_GetVersion_Mul (int nBoard, int *nFpgaVer, int *nFirmVer)

Parameters:

- nBoard : 현재 시스템에 장착되어 있는 보드 번호이다.
보드 번호는 보드의 DIP 스위치 설정(0~ 3)과 일치한다.
- *nFpgaVer : 현재 FPGA 버전 값.
- *nFirmVer : 현재 Firmware 버전 값.

Return Value:

함수 호출에 실패한 경우 'FALSE', 성공일 경우 'TRUE'를 return한다.

LVDS_GetError_Mul

프레임(이미지) 에러를 가져온다.

DWORD LVDS_GetError_Mul (int nBoard, DWORD *dwStatus)

Parameters:

nBoard : 현재 시스템에 장착되어 있는 보드 번호이다.

보드 번호는 보드의 DIP 스위치 설정(0~ 3)과 일치한다.

*dwStatus : 에러 상태 값을 가져온다.

“1” 이면 Overflow error

“2” 이면 Read error

“4” 이면 Size error

Return Value :

함수 호출에 실패한 경우 'FALSE', 성공일 경우 'TRUE'를 return한다.

LVDS_BufferFlush_Mul

버퍼를 초기화 한다.

BOOL LVDS_BufferFlush_Mul (int nBoard)

Parameters:

nBoard : 현재 시스템에 장착되어 있는 보드 번호이다.

보드 번호는 보드의 DIP 스위치 설정(0~ 3)과 일치한다.

Return Value :

함수 호출에 실패한 경우 'FALSE', 성공일 경우 'TRUE'를 return한다.

LVDS_SetDeUse_Mul

데이터 유효(Data Valid) 신호선을 선택한다.

BOOL LVDS_SetDUse_Mul (int nBoard, BOOL bUse)

Parameters:

nBoard : 현재 시스템에 장착되어 있는 보드 번호이다.

보드 번호는 보드의 DIP 스위치 설정(0~ 3)과 일치한다.

bUse : “True” 이면 DVAL(Data Validation) 사용하고

“False” 이면 HSYNC (Horizontal Synchronization)를 사용한다

Return Value :

함수 호출에 실패한 경우 'FALSE', 성공일 경우 'TRUE'를 return한다.

LVDS_SetHsPol_Mul

수평 신호(HSYNC : Horizontal Synchronization) 신호선을 선택한다.

BOOL LVDS_SetHsPol_Mul (int nBoard, BOOL bPol)

Parameters:

nBoard : 현재 시스템에 장착되어 있는 보드 번호이다.

보드 번호는 보드의 DIP 스위치 설정(0~ 3)과 일치한다.

bPol : "True" 이면 Normal HSYNC 사용하고

"False" 이면 Inverse HSYNC를 사용한다

Return Value :

함수 호출에 실패한 경우 'FALSE', 성공일 경우 'TRUE'를 return한다.

LVDS_CameraMode_Mul

카메라 모드가 Area Line Scan Camera인지 Line Scan Camera 인지를 선택한다.

BOOL LVDS_CameraMode_Mul (int nBoard, int nMode)

Parameters:

nBoard : 현재 시스템에 장착되어 있는 보드 번호이다.

보드 번호는 보드의 DIP 스위치 설정(0~ 3)과 일치한다.

nMode : "0" 이면 Area Scan Camera 이고 (Default)

"Others" 이면 Line Scan Camera 이다

Return Value :

함수 호출에 실패한 경우 'FALSE', 성공일 경우 'TRUE'를 return한다.

LVDS_SetReferenceClock_Mul

레퍼런스 클럭을 무엇으로 사용할지를 선택한다.

BOOL LVDS_SetReferenceClock_Mul (int nBoard, int nClcok)

Parameters:

nBoard : 현재 시스템에 장착되어 있는 보드 번호이다.

보드 번호는 보드의 DIP 스위치 설정(0~ 3)과 일치한다.

nClock : "0" 이면 프레임 그레버내의 Internal Clcok 을 사용하고 (Default)

"Others" 이면 엔코더나 다른 보드에서 제공되는 External Clock을 사용한다.

Return Value :

함수 호출에 실패한 경우 'FALSE', 성공일 경우 'TRUE'를 return한다.

LVDS_ConfigureCc_Mul

해당 장치의 환경설정 값을 선택한다.

BOOL LVDS_ConfigureCc_Mul (int nBoard, DWORD dwCFG)

Parameters:

nBoard : 현재 시스템에 장착되어 있는 보드 번호이다.

보드 번호는 보드의 DIP 스위치 설정(0~ 3)과 일치한다.

dwCFG : bit0(CC1 configure) = "0" : digital out1 / "1": alternate (Trigger1 output)

bit1(CC2 configure) = "0" : digital out2 / "1": alternate (Trigger2 output)

bit2(CC3 configure) = "0" : digital out3 / "1": (Digital output)

bit3(CC4 configure)= "0" : digital out4 / "1": alternate (Reference clock
output)

others : Reserved

Return Value :

함수 호출에 실패한 경우 'FALSE', 성공일 경우 'TRUE'를 return한다.

LVDS_CcOutput_Mul

사용하려는 CC (Camera Control)를 선택한다.

CC1을 사용하려면 1을 쓰고, CC2는 2, CC3은 4, CC4는 8을 쓴다.

BOOL LVDS_CcOutput_Mul (int nBoard, DWORD dwCC)

Parameters:

nBoard : 현재 시스템에 장착되어 있는 보드 번호이다.

보드 번호는 보드의 DIP 스위치 설정(0~ 3)과 일치한다.

dwCC : bit0(CC1 out) = "0" : output "0" / "1": output "1"

bit1(CC2 out) = "0" : output "0" / "1": output "1"

bit2(CC3 out) = "0" : output "0" / "1": output "1"

bit3(CC4 out) = "0" : output "0" / "1": output "1"

others : Reserved

Return Value :

함수 호출에 실패한 경우 'FALSE', 성공일 경우 'TRUE'를 return한다.

LVDS_SetLineCount_Mul

한 페이지에서 하나의 이미지를 얻기 위해 필요한 라인의 수를 선택한다.

BOOL LVDS_SetLineCount_Mul (int nBoard, DWORD dwCount)

Parameters:

nBoard : 현재 시스템에 장착되어 있는 보드 번호이다.

보드 번호는 보드의 DIP 스위치 설정(0~ 3)과 일치한다.

dwCount : 1 ~65535

Return Value :

함수 호출에 실패한 경우 'FALSE', 성공일 경우 'TRUE'를 return한다.

LVDS_SetPageStart_Mul

페이지 초기 이벤트를 선택한다.

BOOL LVDS_SetPageStart_Mul (int nBoard, DWORD dwEvent)

Parameters:

nBoard : 현재 시스템에 장착되어 있는 보드 번호이다.

보드 번호는 보드의 DIP 스위치 설정(0~ 3)과 일치한다.

dwEvent : "0" 이면 Continuous (Free Running)

"1" 이면 Rising edge in Page Trigger input

"2" 이면 Rising edge on encoder z phase

Return Value :

함수 호출에 실패한 경우 'FALSE', 성공일 경우 'TRUE'를 return한다.

LVDS_SetPageDelay_Mul

한 페이지에서 하나의 이미지를 얻기 전에 필요한 클럭의 수를 선택한다.

BOOL LVDS_SetPageDelay_Mul (int nBoard, DWORD dwCount)

Parameters:

nBoard : 현재 시스템에 장착되어 있는 보드 번호이다.

보드 번호는 보드의 DIP 스위치 설정(0~ 3)과 일치한다.

dwCount : 1 ~ 15

Return Value :

함수 호출에 실패한 경우 'FALSE', 성공일 경우 'TRUE'를 return한다.

LVDS_SetTrigMode_Mul

Edge 또는 Level Trigger 모드를 선택한다.

BOOL LVDS_SetTrigMode (int nBoard, BOOL bUse)

Parameters:

nBoard : 현재 시스템에 장착되어 있는 보드 번호이다.

보드 번호는 보드의 DIP 스위치 설정(0~ 3)과 일치한다.

bUse : "True" 이면 Trigger Edge Mode (레벨이 상승하거나 하강할 때 Trigger 발생)

"False" 이면 Trigger Level Mode (레벨이 인가될 때 Trigger 발생)

Return Value :

함수 호출에 실패한 경우 'FALSE', 성공일 경우 'TRUE'를 return한다.

LVDS_SetPoCLDelay_Mul

PoCL(Power over Camera Link) 전원을 사용할 때 지연시간(Delay time)을 선택할 수 있다. (DEFAULT 는 0.5sec이다.)

BOOL LVDS_SetPoCLDelay_Mul (int nBoard, int nMode)

Parameters:

nBoard : 현재 시스템에 장착되어 있는 보드 번호이다.

보드 번호는 보드의 DIP 스위치 설정(0~ 3)과 일치한다.

nMode : "0" : 0.5sec, "1" : 1sec, "2" : 1.5sec, "3" : 2sec

"4" : 2.5sec, "5" : 3sec, "6" : 3.5sec, "7" : 4sec

Return Value :

함수 호출에 실패한 경우 'FALSE', 성공일 경우 'TRUE'를 return한다.

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

UART sub-system의 자원, 예를 들어 interrupt와 UART control register를 초기화 한다.

BOOL **UART_Init_Mul (int nBoard)**

Parameters:

nBoard : 현재 시스템에 장착되어 있는 보드 번호이다.

보드 번호는 보드의 DIP 스위치 설정(0~ 3)과 일치한다.

Return Value:

함수 호출에 실패한 경우 'FALSE', 성공일 경우 'TRUE'를 return한다.

UART_GetData_Mul

차동 UART를 통하여 문자를 받는다.

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

Parameters:

nBoard : 현재 시스템에 장착되어 있는 보드 번호이다.

보드 번호는 보드의 DIP 스위치 설정(0~ 3)과 일치한다.

nCnt : 문자를 나타내는 어드레스를 바이트 사이즈로 받는다.

최대 받을 수 있는 문자는 4Kbyte(4096)로 제한되어 있다.

buf : 버퍼 어드레스.

Return Value:

함수 호출에 실패한 경우 'FALSE', 성공일 경우 'TRUE'를 return한다.

UART_SendData_Mul

자동 UART를 통하여 문자를 보낸다.

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

Parameters:

nBoard : 현재 시스템에 장착되어 있는 보드 번호이다.

보드 번호는 보드의 DIP 스위치 설정(0~ 3)과 일치한다.

nCnt : 문자를 나타내는 어드레스를 바이트 사이즈로 받는다.

최대 전송할 수 있는 문자는 4Kbyte(4096)로 제한되어 있다.

buf : 버퍼 어드레스.

Return Value:

함수 호출에 실패한 경우 'FALSE', 성공일 경우 'TRUE'를 return한다.

UART_Close_Mul

UART 함수에서 사용한 모든 자원을 되돌려 준다.

BOOL **UART_Close_Mul (int nBoard)**

Parameters:

nBoard : 현재 시스템에 장착되어 있는 보드 번호이다.

보드 번호는 보드의 DIP 스위치 설정(0~ 3)과 일치한다.

Return Value:

함수 호출에 실패한 경우 'FALSE', 성공일 경우 'TRUE'를 return한다.

UART_SetBaud_Mul

UART Baud를 정한다.

BOOL **UART_SetBaud_Mul (int nBoard, DWORD nBaud)**

Parameters:

nBoard : 현재 시스템에 장착되어 있는 보드 번호이다.

보드 번호는 보드의 DIP 스위치 설정(0~ 3)과 일치한다.

nBaud : 0: 9600, 1: 19200, 2: 38400, 3:57600, 4:115200bps

Return Value:

함수 호출에 실패한 경우 'FALSE', 성공일 경우 'TRUE'를 return한다.

UART_BufferFlush_Mul

UART Rx 버퍼를 clear 한다.

BOOL UART_BufferFlush_Mul (int nBoard)

Parameters:

nBoard : 현재 시스템에 장착되어 있는 보드 번호이다.

보드 번호는 보드의 DIP 스위치 설정(0~ 3)과 일치한다.

Return Value:

함수 호출에 실패한 경우 'FALSE', 성공일 경우 'TRUE'를 return한다.

Multi-board CC(Camera Control) API Functions

Overview

BOOL DIO_Write (int nBoard, DWORD dwVal)

DIO_Write

원하는 카메라 제어 값을 출력포트에 출력한다. (카메라 사양 참조)

BOOL DIO_Write (int nBoard, DWORD dwVal)

Parameters:

nBoard : 현재 시스템에 장착되어 있는 보드 번호이다.

보드 번호는 보드의 DIP 스위치 설정(0~ 3)과 일치한다.

dwVal : 출력 포트에 기록할 값.

Return Value:

함수 호출에 실패한 경우 'FALSE', 성공일 경우 'TRUE'를 return한다.