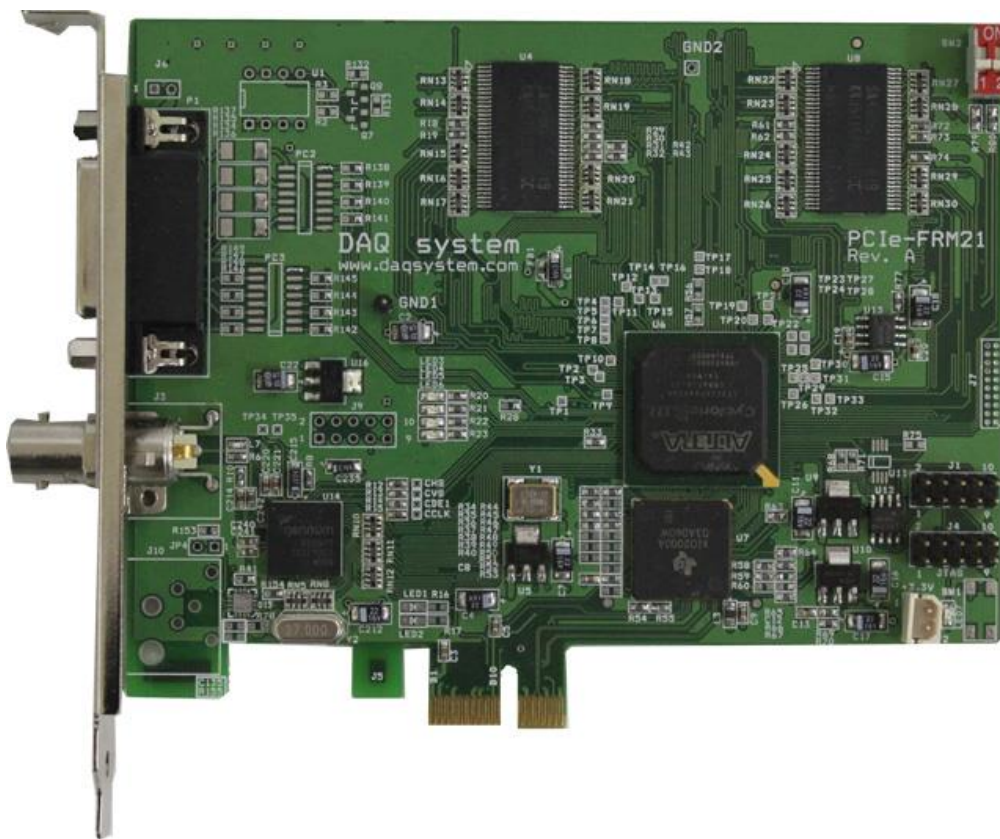


PCIe-FRM21 API Programming (Rev 1.1)



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 APIs

Overview

int	OpenDAQDevice (void)
BOOL	ResetBoard (int nBoard)
BOOL	CloseDAQDevice (void)
int	GetBoardNum (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, "-1"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. (0~3)

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 설치된 보드 개수를 알려준다. 보드 하나가 설치된 경우 "1"로 나타나고 최대 4개 까지 연결 가능함으로 "4" 가 최대 값이다.

int **GetBoardNum (void)**

Parameters:

없음.

Return Value:

설치된 보드의 개수를 리턴한다.

LVDS APIs

Overview

BOOL	LVDS_Init (void)
BOOL	LVDS_Start (void)
BOOL	LVDS_Check(void)
BOOL	LVDS_GetFrame (DWORD* nCnt, unsigned char* buf)
BOOL	LVDS_Close (void)
BOOL	LVDS_SetModel (int Model)
BOOL	LVDS_GetModel (int *Model)
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_SetFilter (DWORD dwValue)
BOOL	LVDS_SetHsFilter (DWORD dwValue)
BOOL	LVDS_GetFilter (DWORD *dwValue)
BOOL	LVDS_SetHsCount (int nCount)
BOOL	LVDS_GetHsCount (int *nCount)
BOOL	LVDS_GetVersion (int *nVersion)
BOOL	LVDS_CameraMode (int nMode)
BOOL	LVDS_SetTriggerOutput (BOOL bBypass, int nCCNumber)
BOOL	LVDS_SetTriggerPulse (int nPulseCount, int nPulseWidth)
BOOL	SDI_SetMode (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 실패할 경우 "FALSE" 성공일 경우 "TRUE"을 리턴함.

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 실패할 경우 "FALSE" 성공일 경우 "TRUE"을 리턴함.

LVDS_Check

프레임 데이터 획득이 완료 되었는가를 검사한다.

BOOL LVDS_Check (void)

Parameters: 없음.

Return Value:

프레임 데이터 획득이 완료 되었으면 "TRUE"를 완료 되지 않았으면 "FALSE"를 리턴한다.

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 실패할 경우 "FALSE" 성공일 경우 "TRUE"을 리턴함.

LVDS_SetModel

LVDS 인터페이스에 사용되는 카메라 모델을 설정한다.

BOOL LVDS_SetModel (int Model)

Parameters:

Model : Model number '0' selects 2048 x 1560 resolution Camera,
otherwise selects 3160 x 2560 resolution Camera.

Return Value:

함수 호출이 성공하면 "TRUE"를 실패하면 "FALSE"를 리턴한다.

LVDS_GetModel

LVDS 인터페이스에 사용되는 카메라 모델을 설정 값을 얻어온다.

BOOL LVDS_GetModel (int *Model)

Parameters:

Model: 카메라 모델 번호를 얻어올 변지 값

Return Value:

함수 호출이 성공하면 "TRUE"를 실패하면 "FALSE"를 리턴한다.

LVDS_SetResolution

Video 입력의 해상도를 설정한다. 이 해상도에 따라 Frame Size가 결정된다.

BOOL LVDS_SetResolution (DWORD xRes, DWORD yRes)

Parameters:

xRes : 수평해상도 값 즉, Frame의 Width를 설정한다.

yRes : 수직해상도 값 즉, Frame의 Height를 설정한다.

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 : “2” 이면 24bit Mode
“others” 이면 16bit Mode

Return Value :

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

LVDS_SetFilter

정상적인 경우에는 일어나지 않지만 만약 케이블을 연결할 경우나 혹은 파손된 카메라에서 정상적인 Vsync가 발생하지 않고 빠르게 들어오게 되면 시스템에 무리를 줄 수 있다. 이 경우에 Vsync에 하드웨어 필터를 설정하여 비 정상적인 신호를 걸러 낼 수 있다.

BOOL LVDS_SetFilter (DWORD dwValue)

Parameters:

dwValue : 값의 범위는 0에서 65535 이며 단위는 약 15nSEC 이며 디폴트 값은 160이다. 필터를 설정할 경우에는 Front porch 및 Back porch를 염두에 두어야 한다.

Return Value:

함수 호출이 성공하면 “TRUE”를 실패하면 “FALSE”를 리턴한다.

LVDS_HsSetFilter

정상적인 경우에는 일어나지 않지만 만약 케이블을 연결할 경우나 혹은 파손된 카메라에서 정상적인 Hsync가 발생하지 않고 빠르게 들어오게 되면 시스템에 무리를 줄 수 있다. 이 경우에 Hsync에 하드웨어 필터를 설정하여 비 정상적인 신호를 걸러 낼 수 있다.

BOOL LVDS_SetHsFilter (DWORD dwValue)

Parameters:

dwValue : 값의 범위는 0에서 65535 이며 단위는 약 15nSEC 이며 디폴트 값은 160이다. 필터를 설정할 경우에는 Front porch 및 Back porch를 염두에 두어야 한다.

Return Value:

함수 호출이 성공하면 "TRUE"를 실패하면 "FALSE"를 리턴한다.

LVDS_GetFilter

현재 하드웨어 적으로 Vsync에 설정된 필터 값을 읽어온다.

BOOL LVDS_GetFilter (DWORD *dwValue)

Parameters:

dwValue: 설정된 필터 값을 읽어올 변수의 번지.

Return Value:

함수 호출이 성공하면 "TRUE"를 실패하면 "FALSE"를 리턴한다.

LVDS_SetHsCount

실제 이미지를 체크하기 위해 Horizontal sync count 를 설정한다.

BOOL LVDS_SetHsCount (int nCount)

Parameters:

nCount: 프레임의 Horizontal sync의 최소 값. 프레임이 이 값보다 적다면 프레임은 쓸모없게 된다.

Return Value:

함수 호출이 성공하면 "TRUE"를 실패하면 "FALSE"를 리턴한다.

LVDS_GetHsCount

Horizontal sync count 값을 가져온다.

BOOL LVDS_GetHsCount (int *nCount)

Parameters:

*nCount: Horizontal count 값.

Return Value:

함수 호출이 성공하면 "TRUE"를 실패하면 "FALSE"를 리턴한다.

LVDS_GetVersion

현재 프로그램의 버전을 가져온다.

BOOL LVDS_GetVersion (int *nVersion)

Parameters:

*nVersion : 현재 프로그램 버전 값.

Return Value :

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

LVDS_CameraMode

카메라 촬영 모드를 선택한다.

BOOL LVDS_CameraMode (int nMode)

Parameters:

- nMode : "0" 이면 Area Mode이고(default),
- "1" 이면 Line Mode이고 (Free Run)
- "2" 이면 Line Mode이고 (외부 포트로부터 Trigger)
- "3" 이면 Line Mode이다. (내부(33MHz) 클럭으로 Trigger)

Return Value :

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

LVDS_SetTriggerOutput

Trigger 를 설정한다.

BOOL LVDS_SetTriggerOutput (BOOL bBypass, int nCCNumber)

Parameters:

bBypass : 외부 Trigger(DIO Board)의 pulse setting에 사용된 할당된 cc를
우회하면 "True", 아니면 "False"

nMode : "0" 이면 CC0, "1" 이면 CC1, "2" 이면 CC2, "3" 이면 CC3
"others" 이면 null 이다

Return Value :

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

LVDS_SetTriggerPulse

Trigger Pulse를 설정한다.

BOOL LVDS_SetTriggerOutput (int nPulseCount, int nPulseWidth)

Parameters:

nPulseCount : 얼마나 많은 pulse가 trigger out에 사용되는가 (default : 2200)

nPulseCount : 얼마나 많은 pulse가 trigger width에 사용되는가 (default : 4)

Return Value :

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

SDI_SetMode

프레임(이미지) 데이터 모드를 선택한다.

BOOL SDI_SetMode (int nMode)

Parameters:

nMode : "0" 이면 Progressive Mode이고, "1"이면 Interlace Mode이다.

Return Value :

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

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

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

BOOL **UART_Init (void)**

Parameters:

없음.

Return Value:

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

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 실패할 경우 "FALSE" 성공일 경우 "TRUE"을 리턴함.

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 실패할 경우 "FALSE" 성공일 경우 "TRUE"을 리턴함.

UART_Close

UART 함수에서 사용한 모든 자원을 되돌려 준다.

BOOL **UART_Close (void)**

Parameters:

없음.

Return Value:

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

UART_SetBaud

UART Baud를 정한다.

BOOL **UART_SetBaud (DWORD nBaud)**

Parameters:

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

Return Value:

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

UART_BufferFlush

UART Rx 버퍼를 clear 한다.

BOOL **UART_BufferFlush (void)**

Parameters:

없음.

Return Value:

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

DIO(Digital Input Output) API Functions

Overview

DWORD DIO_Read (void)
BOOL DIO_Write (DWORD val)

DIO_Read

입력 값을 읽는다.

DWORD DIO_Read (void)

Parameters: 없음.

Return Value:

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

DIO_Write

출력 포트의 값을 설정한다.

BOOL DIO_Write (DWORD val)

Parameters:

val : 포트에 쓰여지는 값.

Return Value:

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

Multi Board support APIs

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

Multi board API는 FPGA version #2 또는 그 이상의 보드에서만 사용할 수 있다.

LVDS APIs

Overview

BOOL	LVDS_Init_Mul (int nBoard)
BOOL	LVDS_Start_Mul (int nBoard)
BOOL	LVDS_Check_Mul (int nBoard)
BOOL	LVDS_GetFrame_Mul (int nBoard, DWORD* nCnt, unsigned char* buf)
BOOL	LVDS_Close_Mul (int nBoard)
BOOL	LVDS_SetModel_Mul (int nBoard, int Model)
BOOL	LVDS_GetModel_Mul (int nBoard, int *Model)
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_SetFilter_Mul (int nBoard, DWORD dwValue)
BOOL	LVDS_SetHsFilter_Mul (int nBoard, DWORD dwValue)
BOOL	LVDS_GetFilter_Mul (int nBoard, DWORD *dwValue)
BOOL	LVDS_SetHsCount_Mul (int nBoard, int nCount)
BOOL	LVDS_GetHsCount_Mul (int nBoard, int *nCount)
BOOL	LVDS_GetVersion_Mul (int nBoard, int *nVersion)
BOOL	LVDS_CameraMode_Mul (int nBoard, int nMode)
BOOL	LVDS_SetTriggerOutput_Mul (int nBoard, BOOL bBypass, int nCCNumber)
BOOL	LVDS_SetTriggerPulse_Mul (int nBoard, int nPulseCount, int nPulseWidth)
BOOL	SDI_SetMode_Mul (int nBoard, int nMode)

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 번호를 알려준다.
보드 번호는 보드의 DIP스위치를 이용하여 설정한다.

Return Value:

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

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 번호를 알려준다.
보드 번호는 보드의 DIP스위치를 이용하여 설정한다.

Return Value:

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

LVDS_Check_Mul

프레임 데이터 획득이 완료 되었는가를 검사한다.

BOOL LVDS_Check_Mul (int nBoard)

Parameters:

nBoard : 현재 시스템에 장착되어 있는 보드 번호를 알려준다.
보드 번호는 보드의 DIP스위치를 이용하여 설정한다.

Return Value:

프레임 데이터 획득이 완료 되었으면 "TRUE"를 완료 되지 않았으면 "FALSE"를 리턴한다.

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 번호를 알려준다.

보드 번호는 보드의 DIP스위치를 이용하여 설정한다.

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.

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 번호를 알려준다.

보드 번호는 보드의 DIP스위치를 이용하여 설정한다.

Return Value:

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

LVDS_SetModel_Mul

LVDS 인터페이스에 사용되는 카메라 모델을 설정한다.

BOOL LVDS_SetModel_Mul (int nBoard, int Model)

Parameters:

- nBoard : 현재 시스템에 장착되어 있는 보드 번호를 알려준다.
보드 번호는 보드의 DIP스위치를 이용하여 설정한다.
- Model : Model number '0' selects 2048 x 1560 resolution Camera,
otherwise selects 3160 x 2560 resolution Camera.

Return Value:

함수 호출이 성공하면 "TRUE"를 실패하면 "FALSE"를 리턴한다.

LVDS_GetModel_Mul

LVDS 인터페이스에 사용되는 카메라 모델을 설정 값을 얻어온다.

BOOL LVDS_GetModel_Mul (int nBoard, int *Model)

Parameters:

- nBoard : 현재 시스템에 장착되어 있는 보드 번호를 알려준다.
보드 번호는 보드의 DIP스위치를 이용하여 설정한다.
- Model: 카메라 모델 번호를 얻어올 번지값

Return Value:

함수 호출이 성공하면 "TRUE"를 실패하면 "FALSE"를 리턴한다.

LVDS_SetResolution_Mul

Video 입력의 해상도를 선택한다. 이 해상도에 따라 Frame Size가 결정된다.

BOOL LVDS_SetResolution_Mul (int nBoard, DWORD xRes, DWORD yRes)

Parameters:

- nBoard : 현재 시스템에 장착되어 있는 보드 번호를 알려준다.
보드 번호는 보드의 DIP스위치를 이용하여 설정한다.
- xRes : 수평해상도 즉, Frame의 Width를 설정한다.
- yRes : 수직해상도 즉, Frame의 Height를 설정한다.

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 번호를 알려준다.

보드 번호는 보드의 DIP스위치를 이용하여 설정한다.

*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 번호를 알려준다.

보드 번호는 보드의 DIP스위치를 이용하여 설정한다.

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 번호를 알려준다.

보드 번호는 보드의 DIP스위치를 이용하여 설정한다.

nMode : "0" 이면 8bit Mode이고, "1" 이면 16bit Mode

Return Value:

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

LVDS_SetFilter_Mul

정상적인 경우에는 일어나지 않지만 만약 케이블을 연결할 경우나 혹은 파손된 카메라에서 정상적인 Vsync가 발생하지 않고 빠르게 들어오게 되면 시스템에 무리를 줄 수 있다. 이 경우에 Vsync에 하드웨어 필터를 설정하여 비 정상적인 신호를 걸러 낼 수 있다.

BOOL LVDS_SetFilter_Mul (int nBoaed, DWORD dwValue)

Parameters:

nBoard : 현재 시스템에 장착되어 있는 보드 번호를 알려준다.

보드 번호는 보드의 DIP스위치를 이용하여 설정한다.

dwValue : 값의 범위는 0에서 65535 이며 단위는 약 15nSEC 이며 디폴트 값은 160이다. 필터를 설정할 경우에는 Front porch 및 Back porch를 염두에 두어야 한다.

Return Value:

함수 호출이 성공하면 "TRUE"를 실패하면 "FALSE"를 리턴한다.

LVDS_HsSetFilter_Mul

정상적인 경우에는 일어나지 않지만 만약 케이블을 연결할 경우나 혹은 파손된 카메라에서 정상적인 Hsync가 발생하지 않고 빠르게 들어오게 되면 시스템에 무리를 줄 수 있다. 이 경우에 Hsync에 하드웨어 필터를 설정하여 비 정상적인 신호를 걸러 낼 수 있다.

BOOL LVDS_SetHsFilter_Mul (int nBoard, DWORD dwValue)

Parameters:

nBoard : 현재 시스템에 장착되어 있는 보드 번호를 알려준다.

보드 번호는 보드의 DIP스위치를 이용하여 설정한다.

dwValue : 값의 범위는 0에서 65535 이며 단위는 약 15nSEC 이며 디폴트 값은 160이다. 필터를 설정할 경우에는 Front porch 및 Back porch를 염두에 두어야 한다.

Return Value:

함수 호출이 성공하면 "TRUE"를 실패하면 "FALSE"를 리턴한다.

LVDS_GetFilter_Mul

현재 하드웨어 적으로 Vsync에 설정된 필터 값을 읽어온다.

BOOL LVDS_GetFilter_Mul (int nBoard, DWORD *dwValue)

Parameters:

nBoard : 현재 시스템에 장착되어 있는 보드 번호를 알려준다.

보드 번호는 보드의 DIP스위치를 이용하여 설정한다.

dwValue: 설정된 필터 값을 읽어올 변수의 번지.

Return Value:

함수 호출이 성공하면 "TRUE"를 실패하면 "FALSE"를 리턴한다.

LVDS_SetHsCount_Mul

실제 이미지를 체크하기 위해 Horizontal sync count 를 설정한다.

BOOL LVDS_SetHsCount_Mul (int nBoard, int nCount)

Parameters:

nBoard : 현재 시스템에 장착되어 있는 보드 번호를 알려준다.

보드 번호는 보드의 DIP스위치를 이용하여 설정한다.

nCount: 프레임의 Horizontal sync의 최소 값. 프레임이 이 값보다 적다면 프레임은 쓸모없게 된다.

Return Value:

함수 호출이 성공하면 "TRUE"를 실패하면 "FALSE"를 리턴한다.

LVDS_GetHsCount_Mul

Horizontal sync count 값을 가져온다. (supported by version 2 or more)

BOOL LVDS_GetHsCount_Mul (int nBoard, int *nCount)

Parameters:

nBoard : 현재 시스템에 장착되어 있는 보드 번호를 알려준다.

보드 번호는 보드의 DIP스위치를 이용하여 설정한다.

*nCount: Horizontal count 값.

Return Value:

함수 호출이 성공하면 "TRUE"를 실패하면 "FALSE"를 리턴한다.

LVDS_GetVersion_Mul

FPGA version을 가져온다.

BOOL LVDS_GetVersion_Mul (int nBoard, int *nVersion)

Parameters:

nBoard : 현재 시스템에 장착되어 있는 보드 번호를 알려준다.

보드 번호는 보드의 DIP스위치를 이용하여 설정한다.

nVersion: FPGA 버전의 포인터.

어떤 API는 FPGA 버전이 2 또는 그 이상을 번호를 갖는 보드들에 의해 지원된다.

Return Value:

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

LVDS_CameraMode_Mul

카메라 촬영 모드를 선택한다.

BOOL LVDS_CameraMode_Mul (int nBoard, int nMode)

Parameters:

nBoard : 현재 시스템에 장착되어 있는 보드 번호를 알려준다.

보드 번호는 보드의 DIP스위치를 이용하여 설정한다.

nMode : "0" 이면 Area Mode이고(default),

"1" 이면 Line Mode이고 (Free Run)

"2" 이면 Line Mode이고 (외부 포트로부터 Trigger)

"3" 이면 Line Mode이다. (내부(33MHz) 클럭으로 Trigger)

Return Value :

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

LVDS_SetTriggerOutput_Mul

Trigger 를 설정한다.

BOOL LVDS_SetTriggerOutput_Mul (int nBoard, BOOL bBypass, int nCCNumber)

Parameters:

- nBoard : 현재 시스템에 장착되어 있는 보드 번호를 알려준다.
보드 번호는 보드의 DIP스위치를 이용하여 설정한다.
- bBypass : 외부 Trigger(DIO Board)의 pulse setting에 사용된 할당된 cc를
우회하면 "True", 아니면 "False"
- nMode : "0" 이면 CC0, "1" 이면 CC1, "2" 이면 CC2, "3" 이면 CC3
"others" 이면 null 이다

Return Value :

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

LVDS_SetTriggerPulse_Mul

Trigger Pulse를 설정한다.

BOOL LVDS_SetTriggerOutput_Mul (int nBoard, int nPulseCount,
int nPulseWidth)

Parameters:

- nBoard : 현재 시스템에 장착되어 있는 보드 번호를 알려준다.
보드 번호는 보드의 DIP스위치를 이용하여 설정한다.
- nPulseCount : 얼마나 많은 pulse가 trigger out에 사용되는가 (default : 2200)
- nPulseCount : 얼마나 많은 pulse가 trigger width에 사용되는가 (default : 4)

Return Value :

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

SDI_SetMode_Mul

프레임(이미지) 데이터 모드를 선택한다.

BOOL SDI_SetMode_Mul (int nBoard, int nMode)

Parameters:

nBoard : 현재 시스템에 장착되어 있는 보드 번호를 알려준다.

보드 번호는 보드의 DIP스위치를 이용하여 설정한다.

nMode : “0” 이면 Progressive Mode이고, “1”이면 Interlace Mode이다.

Return Value :

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

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

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 번호를 알려준다.
보드 번호는 보드의 DIP스위치를 이용하여 설정한다.

Return Value:

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

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 번호를 알려준다.
보드 번호는 보드의 DIP스위치를 이용하여 설정한다.
nCnt : 문자를 나타내는 어드레스를 바이트 사이즈로 받는다.
최대 받을 수 있는 문자는 4Kbyte(4096)로 제한되어 있다.
buf : 버퍼 어드레스.

Return Value:

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

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 번호를 알려준다.
 보드 번호는 보드의 DIP스위치를 이용하여 설정한다.
- nCnt : 문자를 나타내는 어드레스를 바이트 사이즈로 받는다.
 최대 전송할 수 있는 문자는 4Kbyte(4096)로 제한되어 있다.
- buf : 버퍼 어드레스.

Return Value:

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

UART_Close_Mul

UART 함수에서 사용한 모든 자원을 되돌려 준다.

BOOL **UART_Close_Mul (int nBoard)**

Parameters:

- nBoard : 현재 시스템에 장착되어 있는 보드 번호를 알려준다.
 보드 번호는 보드의 DIP스위치를 이용하여 설정한다.

Return Value:

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

UART_SetBaud_Mul

UART Baud를 정한다.

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

Parameters:

- nBoard : 현재 시스템에 장착되어 있는 보드 번호를 알려준다.
 보드 번호는 보드의 DIP스위치를 이용하여 설정한다.
- nBaud : 0: 9600, 1: 19200, 2: 38400, 3:57600, 4:115200bps

Return Value:

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

UART_BufferFlush_Mul

UART Rx 버퍼를 clear 한다.

BOOL UART_BufferFlush_Mul (int nBoard)

Parameters:

nBoard : 현재 시스템에 장착되어 있는 보드 번호를 알려준다.
 보드 번호는 보드의 DIP스위치를 이용하여 설정한다.

Return Value:

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

DIO(Digital Input Output) API Functions

Overview

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

DIO_Read_Mul

입력 값을 읽는다.

DWORD DIO_Read_Mul (int nBoard)

Parameters:

nBoard : 현재 시스템에 장착되어 있는 보드 번호를 알려준다.
보드 번호는 보드의 DIP스위치를 이용하여 설정한다.

Return Value:

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

DIO_Write_Mul

Digital Output 핀의 상태를 설정한다.

BOOL DIO_Write_Mul (int nBoard, DWORD val)

Parameters:

nBoard : 현재 시스템에 장착되어 있는 보드 번호를 알려준다.
보드 번호는 보드의 DIP스위치를 이용하여 설정한다.
val : 포트에 쓰여지는 값.

Return Value:

함수 호출에 실패할 경우 "FALSE" 성