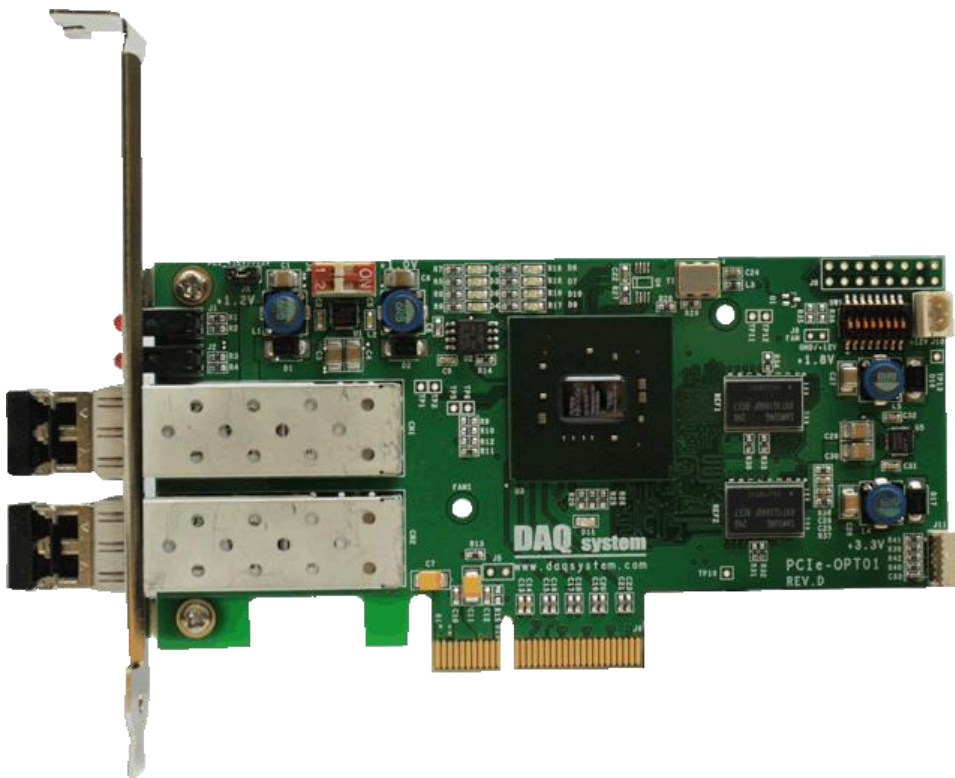


PCIe-OPT01 API Programming (Rev 1.1)



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 © 2019 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 Open 하여야 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 API 함수도 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 API Functions

Overview

BOOL	PCI_LVDS_Init (int nBoard, int nCh)
BOOL	PCI_LVDS_Start (int nBoard, int nCh)
BOOL	PCI_LVDS_GetFrame (int nBoard, int nCh, DWORD* nCnt, unsigned char* buf)
BOOL	PCI_LVDS_Close (int nBoard, int nCh)
BOOL	PCI_LVDS_SetResolutuion (int nBoard, int nCh, DWORD *xRes, DWORD *yRes)
BOOL	PCI_LVDS_GetResolutuion (int nBoard, int nCh, DWORD *xRes, DWORD *yRes)
BOOL	PCI_LVDS_Stop (int nBoard, int nCh)
DWORD	PCI_LVDS_GetError (int nBoard, int nCh, DWORD *dwStatuse)
BOOL	PCI_LVDS_SetDataMode (int nBoard, int nCh, int nMode)
BOOL	PCI_LVDS_GetVersion (int nBoard, int nCh, int *nFpgaVer, int *nFirmVer)
BOOL	PCI_CC_Output (int nBoard, int nCh, DWORD dwVal)
BOOL	PCI_LVDS_DdrUse(int nBaord, int nCh, BOOL bUse)

PCI_LVDS_Init

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

BOOL PCI_LVDS_Init (int nBoard, int nCh)

Parameters:

- nBoard : 현재 시스템에 장착되어 있는 보드 번호를 알려준다.
보드 번호는 보드의 DIP 스위치를 이용하여 설정한다.
- nCh : 광 채널을 선택한다. 0: 0번 채널, 1: 1번 채널

Return Value:

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

PCI_LVDS_Start

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

BOOL **PCI_LVDS_Start (int nBoard, int nCh)**

Parameters:

- nBoard : 현재 시스템에 장착되어 있는 보드 번호를 알려준다.
보드 번호는 보드의 DIP 스위치를 이용하여 설정한다.
- nCh : 광 채널을 선택한다. 0 : 0번 채널, 1: 1번 채널

Return Value:

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

PCI_LVDS_GetFrame

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

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

Parameters:

- nBoard : 현재 시스템에 장착되어 있는 보드 번호를 알려준다.
보드 번호는 보드의 DIP 스위치를 이용하여 설정한다.
- nCh : 광 채널을 선택한다. 0 : 0번 채널, 1: 1번 채널
- nCnt : 프레임 데이터를 받아 올 버퍼의 크기가 저장 되어있는 변수의 번지이다.
함수를 호출할 때 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"를 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.

PCI_LVDS_Close

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

BOOL PCI_LVDS_Close (int nBoard, int nCh)

Parameters:

nBoard : 현재 시스템에 장착되어 있는 보드 번호를 알려준다.
 보드 번호는 보드의 DIP 스위치를 이용하여 설정한다.
nCh : 광 채널을 선택한다. 0 : 0번 채널, 1: 1번 채널

Return Value:

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

PCI_LVDS_SetResolution

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

BOOL PCI_LVDS_SetResolutuion (int nBoard, int nCh, DWORD xRes, DWORD yRes)

Parameters:

nBoard : 현재 시스템에 장착되어 있는 보드 번호를 알려준다.
 보드 번호는 보드의 DIP 스위치를 이용하여 설정한다.
nCh : 광 채널을 선택한다. 0 : 0번 채널, 1: 1번 채널
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.

PCI_LVDS_GetResolution

Video 입력의 해상도를 가져온다.

BOOL PCI_LVDS_GetResolutuion (int nBoard, int nCh, DWORD *xRes, DWORD *yRes)

Parameters:

- nBoard : 현재 시스템에 장착되어 있는 보드 번호를 알려준다.
보드 번호는 보드의 DIP 스위치를 이용하여 설정한다.
- nCh : 광 채널을 선택한다. 0: 0번 채널, 1: 1번 채널
- *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.

PCI_LVDS_Stop

프레임 데이터 Capture를 중지한다.

BOOL PCI_LVDS_Stop (int nBoard, int nCh)

Parameters:

- nBoard : 현재 시스템에 장착되어 있는 보드 번호를 알려준다.
보드 번호는 보드의 DIP 스위치를 이용하여 설정한다.
- nCh : 광 채널을 선택한다. 0: 0번 채널, 1: 1번 채널

Return Value :

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

PCI_LVDS_GetError

프레임(이미지) 에러를 가져온다.

DWORD **PCI_LVDS_GetError (int nBoard, int nCh, DWORD *dwStatus)**

Parameters:

- nBoard : 현재 시스템에 장착되어 있는 보드 번호를 알려준다.
보드 번호는 보드의 DIP 스위치를 이용하여 설정한다.
- nCh : 광 채널을 선택한다. 0 : 0번 채널, 1: 1번 채널
- *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.

PCI_LVDS_SetDataMode

프레임(이미지) 데이터 모드를 선택한다.

BOOL **PCI_LVDS_SetDataMode (int nBoard, int nCh, int nMode)**

Parameters:

- nBoard : 현재 시스템에 장착되어 있는 보드 번호를 알려준다.
보드 번호는 보드의 DIP 스위치를 이용하여 설정한다.
- nCh : 광 채널을 선택한다. 0 : 0번 채널, 1: 1번 채널
- nMode : “0” 이면 8bit Mode이고, “1” 이면 16bit Mode
“2” 이면 24bit Mode이고, “3” 이면 32bit Mode이고
“4” 이면 16Bit YUV Mode이다.

Return Value :

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

PCI_LVDS_GetVersion

현재 프로그램의 FPG와 Firmware 버전을 가져온다.

BOOL **PCI_LVDS_GetVersion (int nBoard, int nCh, int *nFpgaVer, int *nFirmVer)**

Parameters:

- nBoard : 현재 시스템에 장착되어 있는 보드 번호를 알려준다.
보드 번호는 보드의 DIP 스위치를 이용하여 설정한다.
- nCh : 광 채널을 선택한다. 0 : 0번 채널, 1: 1번 채널
- *nFpgaVer : 현재 프로그램의 FPGA 버전 값.
- *nFirmVer : 현재 프로그램의 Firmware 버전 값.

Return Value :

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

PCI_CC_Output

해당 비트의 CC 값을 출력한다.

BOOL **PCI_CC_Output (int nBoard, int nCh, DWORD dwVal)**

Parameters:

- nBoard : 현재 시스템에 장착되어 있는 보드 번호이다.
보드 번호는 보드의 DIP 스위치 설정(0~ 3)과 일치한다.
- nCh : 광 채널을 선택한다. 0 : 0번 채널, 1: 1번 채널
- dwVal : 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.

PCI_LVDS_DdrUse

DDR 메모리의 사용 여부를 결정한다.

BOOL **PCI_LVDS_DdrUse (int nBoard, int nCh, BOOL bUse)**

Parameters:

nBoard : 현재 시스템에 장착되어 있는 보드 번호이다.

보드 번호는 보드의 DIP 스위치 설정(0~ 3)과 일치한다.

nCh : 광 채널을 선택한다. 0 : 0번 채널, 1: 1번 채널

bUse: 'TRUE' 이면 DDR 메모리를 사용하고

'FALSE' 이면 DDR 메모리를 사용하지 않는다.

Return Value :

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

UART Contril API Functions

Overview

BOOL	PCI_UART_Init (int nBoard, int bCh)
BOOL	PCI_UART_GetData (int nBoard, int nCh, DWORD* nCnt, unsigned char* buf)
BOOL	PCI_UART_SetData (int nBoard, int nCh, DWORD * nCnt, unsigned char* buf)
BOOL	PCI_UART_Close (int nBoard, int nCh)
BOOL	PCI_UART_SetBaud (int nBoard, int nCh, DWORD nBaud)

PCI_UART_Init

UART sub-system의 자원, 예를 들어 Interrupt와 UART control register를 초기화한다.

BOOL PCI_UART_Init (int nBoard, int nCh)

Parameters:

nBoard : 현재 시스템에 장착되어 있는 보드 번호이다.

보드 번호는 보드의 DIP 스위치 설정(0~ 3)과 일치한다.

nCh : 광 채널을 선택한다. 0 : 0번 채널, 1: 1번 채널

Return Value:

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

PCI_UART_GetData

차동 UART를 통하여 문자를 받는다.

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

Parameters:

nBoard : 현재 시스템에 장착되어 있는 보드 번호이다.
 보드 번호는 보드의 DIP 스위치 설정(0~ 3)과 일치한다.
nCh : 광 채널을 선택한다. 0 : 0번 채널, 1: 1번 채널
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"를 리턴함.

PCI_UART_SendData

차동 UART를 통하여 문자를 보낸다.

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

Parameters:

nBoard : 현재 시스템에 장착되어 있는 보드 번호이다.
 보드 번호는 보드의 DIP 스위치 설정(0~ 3)과 일치한다.
nCh : 광 채널을 선택한다. 0 : 0번 채널, 1: 1번 채널
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"를 리턴함.

PCI_UART_Close

UART 함수에서 사용한 모든 자원을 되돌려 준다.

BOOL PCI_UART_Close (int nBoard, int nCh,)

Parameters:

nBoard : 현재 시스템에 장착되어 있는 보드 번호이다.

보드 번호는 보드의 DIP 스위치 설정(0~ 3)과 일치한다.

nCh : 광 채널을 선택한다. 0 : 0번 채널, 1: 1번 채널

Return Value:

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

PCI_UART_SetBaud

UART Baud를 정한다.

BOOL PCI_UART_SetBaud (int nBoard, int nCh, DWORD nBaud)

Parameters:

nBoard : 현재 시스템에 장착되어 있는 보드 번호이다.

보드 번호는 보드의 DIP 스위치 설정(0~ 3)과 일치한다.

nCh : 광 채널을 선택한다. 0 : 0번 채널, 1: 1번 채널

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

Return Value:

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