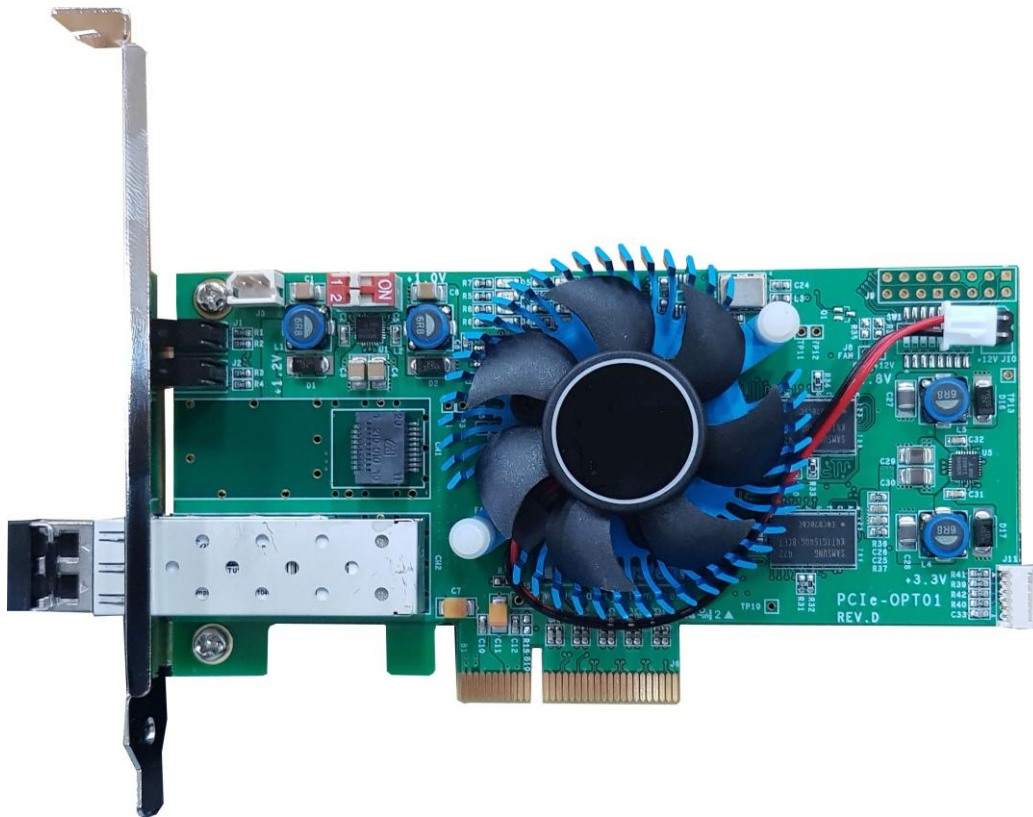


PCIe-OPT01 API Programming (Rev 1.0)



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 © 2005 DAQ system, All rights reserved.

Board Level APIs

Overview

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

GetDllVersion

DLL 버전을 알려준다.

Char* GetDllVersion (void)

Parameters: 없음.

Return Value:

설치되어 있는 DLL의 날짜가 넘어 온다.

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_GetResolutuion (int nBoard, int nCh, DWORD *xRes, DWORD *yRes)
BOOL	PCI_LVDS_Stop (int nBoard, int nCh)
BOOL	PCI_LVDS_SetDataMode (int nBoard, int nCh, int nMode)
BOOL	PCI_LVDS_GetVersion (int nBoard, int nCh, int *nFpgaVer, int *nFirmVer)
BOOL	PCI_LVDS_SelectInput (int nBoard, int nCh, int nInput)
BOOL	PCI_LVDS_SelectCable (int nBoard, int nCh, int nCablet)
BOOL	PCI_LVDS_BufferFlush (int nBoard, int nCh)
BOOL	PCI_LVDS_GetMipilD (int nBoard, int nCh, DWORD *dwID)
BOOL	PCI_LVDS_CheckSum(int nBoard, int nCh, BOOL bOn);
BOOL	PCI_LVDS_CheckSum (int nBoard, int nCh, BOOL bOn)
BOOL	PCI_LVDS_DdrUse (int nBoard, int nCh, BOOL bUse)
BOOL	PCI_LVDS_DdrReset(int nBoard, int nOption)
BOOL	PCI_LVDS_GetDmaBufferSize(DWORD *pdwBytes)
BOOL	PCI_LVDS_SetDmaBufferSize(DWORD dwBytes)

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"을 리턴함.

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_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 FPAG와 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_LVDS_SelectInput

LVDS 입력 모드를 설정한다.

BOOL **PCI_LVDS_SelectInput (int nBoard, int nCh, int nInput)**

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 MIPI 입력이고 "1" 이면 Parallel 입력이다.

Return Value :

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

PCI_LVDS_SelectCable

LVDS 케이블 모드를 설정한다.

BOOL PCI_LVDS_SelectInput (int nBoard, int nCh, int nCable)

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 직접 연결이고 “1” 이면 Cable 로 연결한다.

Return Value :

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

PCI_LVDS_BufferFlush

버퍼를 초기화 한다.

BOOL PCI_LVDS_BufferFlush (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_GetMipiID

MIPI 이미지 ID를 읽어온다.

BOOL PCI_LVDS_GetMipiID (int nBoard, int nCh, DWORD : *dwID)

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
*dwID: MIPI Image ID 번지.

Return Value :

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

LVDS_CheckSum

Check Sum 값을 설정한다.

BOOL LVDS_CheckSum (int nBoard, int nCh, BOOL bOn)

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

bOn : "1" 이면 2바이트 checksum, "0" 이면 check sum off

Return Value :

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

PCI_LVDS_DdrUse

이 함수는 DDR 메모리 사용 유무를 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 : "0" 이면 DDR 메모리를 사용하지 않고

"1" 이면 DDR 메모리를 사용한다

Return Value :

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

PCI_LVDS_DdrReset

이 함수는 PCIe-OPT01 보드의 DDR 메모리를 초기화한다.

BOOL PCI_LVDS_DdrReset (int nBoard, BOOL nOption)

Parameters:

nBoard : 현재 시스템에 장착되어 있는 보드 번호이다.

보드 번호는 보드의 DIP 스위치 설정(0~ 3)과 일치한다.

nOption : Reserved for future Use

Return Value :

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

PCI_LVDS_GetDmaBufferSize

이 함수는 DMA Buffer 크기를 가져온다.

BOOL **PCI_LVDS_GetDmaBufferSize (DWORD *pdwBytes)**

Parameters:

pdwBytes : DMA Buffer Size

Return Value :

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

PCI_LVDS_SetDmaBufferSize

이 함수는 DMA Buffer 크기를 설정한다.

BOOL **PCI_LVDS_GetDmaBufferSize (DWORD dwBytes)**

Parameters:

dwBytes : DMA Buffer Size

Return Value :

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

Clock API Functions

Overview

BOOL **PCI_CLK_Set (int nBoard, int nCh, DWORD val)**

BOOL **PCI_CLK_Off (int nBoard, int nCh, BOOL bOff)**

PCI_CLK_Set

Clock 주파수를 1039Hz ~ 68MHz 까지 설정할 수 있다.

BOOL **PCI_CLK_Set (int nBoard, int nCh, DWORD val)**

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

 val : 1039Hz ~ 68Mhz.

Return Value :

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

PCI_CLK_Off

Clock 동작을 중지시킨다.

BOOL **PCI_CLK_Off (int nBoard, int nCh, BOOL bOff)**

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

bOff : "True"면 Clock Off, "False"면 Clock On

Return Value :

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

Parallel GPIO Input/Output API Functions

Overview

BOOL	PCI_DIO_SetDirection (int nBoard, int nCh, DWORD dwVal)
BOOL	PCI_DIO_GetDirection (int nBoard, int nCh, DWORD *dwVal)
DWORD	PCI_DIO_Read (int nBoard, int nCh)
BOOL	PCI_DIO_Write (int nBoard, int nCh, DWORD dwVal)
BOOL	PCI_DIO_GetWrite (int nBoard, int nCh, DWORD *dwVal)

참조) User GPIO (General Purpose I/O) MIPI-OPT06/08 보드 매뉴얼 참조

PCI_DIO_SetDirection

각각의 포트를 입력으로 사용할지 출력으로 사용할지 설정한다.

BOOL PCI_DIO_SetDirection (int nBoard, int nCh, DWORD dwVal)

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
- dwVal : 입/출력 direction 설정 값.
각 포트에 '1'이면 출력 / '0'이면 입력

Return Value:

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

PCI_DIO_GetDirection

현재 설정된 direction 값을 읽어 온다.

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

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
- *dwVal : 입/출력 direction을 읽어올 변수.
각 포트에 '1'이면 출력 / '0'이면 입력

Return Value:

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

PCI_DIO_Read

입력 값을 읽어 온다.

DWORD PCI_DIO_Read (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_DIO_Write

원하는 값을 출력포트에 출력한다.

BOOL PCI_DIO_Write (int nBoard, int nCh, DWORD dwVal)

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
dwVal : 출력 포트에 기록할 값.

Return Value:

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

PCI_DIO_GetWrite

출력된 현재 값을 적는다.

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

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
*dwVal : 출력 포트의 현재 값을 쓸 변수이다.

Return Value:

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

Sensor GPIO Input/Output API Functions

Overview

BOOL	PCI_SDIO_SetDirection (int nBoard, int nCh, DWORD dwVal)
BOOL	PCI_SDIO_GetDirection (int nBoard, int nCh, DWORD *dwVal)
DWORD	PCI_SDIO_Read (int nBoard, int nCh)
BOOL	PCI_SDIO_Write (int nBoard, int nCh, DWORD dwVal)
BOOL	PCI_SDIO_GetWrite (int nBoard, int nCh, DWORD *dwVal)

참조) Sensor GPIO (General Purpose I/O) MIPI-OPT06/08 보드 매뉴얼 참조

PCI_SDIO_SetDirection

센서 각각의 포트를 입력으로 사용할지 출력으로 사용할지 설정한다.

BOOL PCI_SDIO_SetDirection (int nBoard, int nCh, DWORD dwVal)

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
- dwVal : 입/출력 direction 설정 값.
각 포트에 '1'이면 출력 / '0'이면 입력

Return Value:

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

PCI_SDIO_GetDirection

센서의 현재 설정된 direction 값을 읽어 온다.

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

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
- *dwVal : 입/출력 direction을 읽어올 변수.
각 포트에 '1'이면 출력 / '0'이면 입력

Return Value:

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

PCI_SDIO_Read

센서 입력 값을 읽어 온다.

DWORD PCI_SDIO_Read (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_SDIO_Write

센서에 원하는 값을 출력포트에 출력한다.

BOOL PCI_SDIO_Write (int nBoard, int nCh, DWORD dwVal)

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
dwVal : 출력 포트에 기록할 값.

Return Value:

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

PCI_SDIO_GetWrite

센서에 출력된 현재 값을 적는다.

BOOL PCI_SDIO_GetWrite (int nBoard, DWORD *dwVal)

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
*dwVal : 출력 포트의 현재 값을 쓸 변수이다.

Return Value:

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

User GPIO(3.3V Fix) Input/Output API Functions

Overview

BOOL	PCI_DIO33_SetDirection (int nBoard, int nCh, DWORD dwVal)
BOOL	PCI_DIO33_GetDirection (int nBoard, int nCh, DWORD *dwVal)
DWORD	PCI_DIO33_Read (int nBoard, int nCh)
BOOL	PCI_DIO33_Write (int nBoard, int nCh, DWORD dwVal)
BOOL	PCI_DIO33_GetWrite (int nBoard, int nCh, DWORD *dwVal)

참조) USER 3.3V GPIO(General Purpose I/O) MIPI-OPT06 보드 매뉴얼 참조

PCI_DIO33_SetDirection

사용자 GPIO 각각의 포트를 입력으로 사용할지 출력으로 사용할지 설정한다.

BOOL PCI_DIO33_SetDirection (int nBoard, int nCh, DWORD dwVal)

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
- dwVal : 입/출력 direction 설정 값.
각 포트에 '1'이면 출력 / '0'이면 입력

Return Value:

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

PCI_DIO33_GetDirection

사용자 GPIO 설정된 direction 값을 읽어 온다.

BOOL PCI_DIO33_GetDirection (int nBoard, int nCh, DWORD *dwVal)

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
- *dwVal : 입/출력 direction을 읽어올 변수.
각 포트에 '1'이면 출력 / '0'이면 입력

Return Value:

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

PCI_DIO33_Read

사용자 GPIO 입력 값을 읽어 온다.

DWORD PCI_DIO33_Read (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_DIO33_Write

사용자 GPIO 원하는 값을 출력포트에 출력한다.

BOOL PCI_DIO33_Write (int nBoard, int nCh, DWORD dwVal)

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
dwVal : 출력 포트에 기록할 값.

Return Value:

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

PCI_DIO33_GetWrite

사용자 GPIO 출력된 현재 값을 적는다.

BOOL PCI_DIO33_GetWrite (int nBoard, int nCh, DWORD *dwVal)

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
*dwVal : 출력 포트의 현재 값을 쓸 변수이다.

Return Value:

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

Power GPIO Input/Output API Functions

Overview

BOOL	PCI_PWR_DIO_SetDirection (int nBoard, int nCh, DWORD dwVal)
BOOL	PCI_PWR_DIO_GetDirection (int nBoard, int nCh, DWORD *dwVal)
DWORD	PCI_PWR_DIO_Read (int nBoard, int nCh)
BOOL	PCI_PWR_DIO_Write (int nBoard, int nCh, DWORD dwVal)
BOOL	PCI_PWR_DIO_GetWrite (int nBoard, int nCh, DWORD *dwVal)

참조) Power GPIO(General Purpose I/O) MIPI-OPT06 보드 매뉴얼 참조

PCI_PWR_DIO_SetDirection

전원 GPIO 각각의 포트를 입력으로 사용할지 출력으로 사용할지 설정한다.

BOOL PCI_PWR_DIO_SetDirection (int nBoard, int nCh, DWORD dwVal)

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
- dwVal : 입/출력 direction 설정 값.
각 포트에 '1'이면 출력 / '0'이면 입력

Return Value:

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

PCI_PWR_DIO_GetDirection

전원 GPIO의 설정된 direction 값을 읽어 온다.

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

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
- *dwVal : 입/출력 direction을 읽어올 변수.
각 포트에 '1'이면 출력 / '0'이면 입력

Return Value:

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

PCI_PWR_DIO_Read

전원 GPIO 입력 값을 읽어 온다.

DWORD PCI_PWR_DIO_Read (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_PWR_DIO_Write

전원 GPIO 원하는 값을 출력포트에 출력한다.

BOOL PCI_PWR_DIO_Write (int nBoard, int nCh, DWORD dwVal)

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
dwVal : 출력 포트에 기록할 값.

Return Value:

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

PCI_PWR_DIO_GetWrite

전원 GPIO 출력된 현재 값을 적는다.

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

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
*dwVal : 출력 포트의 현재 값을 쓸 변수이다.

Return Value:

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

UART Control API Functions

Overview

BOOL	PCI_UART_Init (int nBoard, int nCh)
BOOL	PCI_UART_Close (int nBoard, int nCh)
BOOL	PCI_UART_GetData (int nBoard, int nCh, DWORD* nCnt, unsigned char* buf)
BOOL	PCI_UART_SendData (int nBoard, int nCh, DWORD * nCnt, unsigned char* buf)
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_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.

I2C API Functions

Overview

BOOL	PCI_I2C_PWR_Init (int nBoard, int nCh, DWORD nKHz)
BOOL	PCI_I2C_PWR_Read (int nBoard, int nCh, BYTE sIAddr, DWORD nCnt, unsigned char* buf)
BOOL	PCI_I2C_PWR_Write (int nBoard, B int nCh, YTE sIAddr, DWORD nCnt, unsigned char* buf)
BOOL	PCI_I2C_PWR_Close (int nBoard, int nCh)
BOOL	PCI_I2C_ADC_Init (int nBoard, int nCh, DWORD nKHz)
BOOL	PCI_I2C_ADC_Read (int nBoard, int nCh, BYTE sIAddr, DWORD nCnt, unsigned char* buf)
BOOL	PCI_I2C_ADC_Write (int nBoard, int nCh, BYTE sIAddr, DWORD nCnt, unsigned char* buf)
BOOL	PCI_I2C_ADC_Close (int nBoard, int nCh)
BOOL	PCI_I2C_OS_Init (int nBoard, int nCh, DWORD nKHz)
BOOL	PCI_I2C_OS_Read (int nBoard, int nCh, BYTE sIAddr, DWORD nCnt, unsigned char* buf)
BOOL	PCI_I2C_OS_Write (int nBoard, int nCh, BYTE sIAddr, DWORD nCnt, unsigned char* buf)
BOOL	PCI_I2C_OS_Closet (int nBoard, int nCh)
BOOL	PCI_I2C_SYS_Reset (int nBoard, int nCh)
BOOL	PCI_I2C_SYS_Set_Clock (int nBoard, int nCh, int nClock)
BOOL	PCI_I2C_SYS_Read (int nBoard, int nCh, BYTE sIAddr, DWORD nAddrLen, DWORD nAddr, DWORD nCnt, unsigned char* buf)
BOOL	PCI_I2C_SYS_Write (int nBoard, int nCh, BYTE sIAddr, DWORD nAddrLen, DWORD nAddr, DWORD nCnt, unsigned char* buf)
BOOL	PCI_I2C_SYS_Read2 (int nBoard, int nCh, BYTE sIAddr, DWORD nAddrLen, unsigned char* Addrbuf, DWORD nCnt, unsigned char* buf)
BOOL	PCI_SEN_Reset (int nBoard, int nCh, BOOL bReset)
BOOL	PCI_SEN_Enable (int nBoard, int nCh, BOOL bEnable)
BOOL	PCI_SEN_Power (int nBoard, int nCh, BOOL bOn)

PCI_I2C_PWR_Init

Power 모듈의 I2C 기능을 정해진 클럭 속도에 따라 초기화한다. 프로그램 기동 시 한 번 호출한다. 다른 I2C PWR 함수는 이 함수를 호출한 이후에 사용한다.

BOOL PCI_I2C_PWR_Init (int nBoard, int nCh, DWORD nKHz)

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
- nKHz : I2C 통신 속도(Clock 속도를 설정한다.)
1: 100KHz, 2: 200KHz, 3: 300KHz, 4: 400KHz, Others: 100KHz

Return Value:

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

PCI_I2C_PWR_Read

I2C를 통하여 Power 모듈의 특정 어드레스 번지 값을 정해진 개수만큼 읽어 온다.

BOOL PCI_I2C_PWR_Read (int nBoard, int nCh, BYTE slAddr, 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
- slAddr : Slave address
- nCnt : 읽어 올 데이터의 바이트 개수.
- buf : 읽어 올 데이터의 버퍼 포인터.

Return Value:

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

PCI_I2C_PWR_Write

I2C를 통하여 Power 모듈의 특정 어드레스 번지 값을 정해진 개수만큼 기록한다.

BOOL **PCI_I2C_PWR_Write (int nBoard, int nCh, BYTE sIAddr, 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
sIAddr : Slave address
nCnt : 기록할 데이터의 바이트 개수.
buf : 기록할 데이터의 버퍼 포인터.

Return Value:

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

PCI_I2C_PWR_Close

I2C 기능 사용이 완전히 끝나고 리소스를 해제할 때 호출한다. 프로그램 종료 시 한번 호출하면 된다.

BOOL **PCI_I2C_PWR_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"을 리턴함.

PCI_I2C_ADC_Init

AD Converter 모듈의 I2C 기능을 정해진 클럭 속도에 따라 초기화한다. 프로그램 기동 시 한 번 호출한다. 다른 I2C ADC 함수는 이 함수를 호출한 이후에 사용한다.

BOOL PCI_I2C_ADC_Init (int nBoard, int nCh, DWORD nKHz)

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

nKHz : 1: 100KHz, 2: 200KHz, 3: 300KHz, 4: 400KHz, Others: 100KHz

Return Value:

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

PCI_I2C_ADC_Read

I2C를 통하여 AD Converter 모듈의 특정 어드레스 번지 값을 정해진 개수만큼 읽어 온다.

BOOL PCI_I2C_ADC_Read (int nBoard, int nCh, BYTE slAddr, 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

slAddr : Slave address

nCnt : 읽어 올 데이터의 바이트 개수.

buf : 읽어 올 데이터의 버퍼 포인터.

Return Value:

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

PCI_I2C_ADC_Write

I2C를 통하여 AD Converter 모듈의 특정 어드레스 번지 값을 정해진 개수만큼 기록 한다.

BOOL **PCI_I2C_ADC_Write (int nBoard, int nCh, BYTE sIAddr, 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
- sIAddr : Slave address
- nCnt : 기록할 데이터의 바이트 개수.
- buf : 기록할 데이터의 버퍼 포인터.

Return Value:

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

PCI_I2C_ADC_Close

I2C 기능 사용이 완전히 끝나고 리소스를 해제할 때 호출한다. 프로그램 종료 시 한번 호출하면 된다.

BOOL **PCI_I2C_ADC_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"을 리턴함.

PCI_I2C_OS_Init

Open/Short 모듈의 I2C 기능을 정해진 클럭 속도에 따라 초기화한다. 프로그램 기동 시 한 번 호출한다. 다른 I2C OS 함수는 이 함수를 호출한 이후에 사용한다.

BOOL PCI_I2C_OS_Init (int nBoard, int nCh, DWORD nKHz)

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
- nKHz : I2C 통신 속도(Clock 속도를 설정한다.)
1: 100KHz, 2: 200KHz, 3: 300KHz, 4: 400KHz, Others: 100KHz

Return Value:

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

PCI_I2C_OS_Read

I2C를 통하여 Open/Short 모듈의 특정 어드레스 번지 값을 정해진 개수만큼 읽어 온다.

BOOL PCI_I2C_OS_Read (int nBoard, int nCh, BYTE slAddr, 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
- slAddr : Slave address
- nCnt : 읽어 올 데이터의 바이트 개수.
- buf : 읽어 올 데이터의 버퍼 포인터.

Return Value:

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

PCI_I2C_OS_Write

I2C를 통하여 Open/Short 모듈의 특정 어드레스 번지 값을 정해진 개수만큼 기록한다.

BOOL **PCI_I2C_OS_Write (int nBoard, int nCh, BYTE sIAddr, 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
- sIAddr : Slave address
- nCnt : 기록할 데이터의 바이트 개수.
- buf : 기록할 데이터의 버퍼 포인터.

Return Value:

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

PCI_I2C_OS_Close

I2C 기능 사용이 완전히 끝나고 리소스를 해제할 때 호출한다. 프로그램 종료 시 한번 호출하면 된다.

BOOL **PCI_I2C_OS_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"을 리턴함.

PCI_I2C_SYS_Reset

시스템 모듈의 I2C system의 자원을 초기화 한다.

BOOL PCI_I2C_SYS_Reset (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_I2C_SYS_Set_Clock

시스템 모듈의 I2C system의 baud rate을 설정한다.

BOOL PCI_I2C_SYS_Set_Clock (int nBoard, int nCh, int nClock)

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

nClock : 100KHz, 200KHz, 300KHz (default :100000)

Return Value:

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

PCI_I2C_SYS_Read

시스템 모듈의 I2C를 통하여 특정 어드레스 번지 값을 정해진 개수만큼 읽어 온다.

BOOL **PCI_I2C_SYS_Read (int nBoard, int nCh, BYTE sIAddr, DWORD nAddrLen, DWORD nAddr, 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
- sIAddr : I2C address
- nAddrLen : Address Lengh
- nAddr : Register Address
- nCnt : 읽어 올 데이터의 바이트 개수.
- buf : 읽어 올 데이터의 버퍼 포인터.

Return Value:

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

PCI_I2C_SYS_Write

시스템 모듈의 I2C를 통하여 특정 어드레스 번지 값을 정해진 개수만큼 읽어 온다.

BOOL **PCI_I2C_SYS_Write (int nBoard, int nCh, BYTE sIAddr, DWORD nAddrLen, DWORD nAddr, 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
- sIAddr : I2C address
- nAddrLen : Address Lengh
- nAddr : Register Address
- nCnt : 기록할 데이터의 바이트 개수.
- buf : 기록할 데이터의 버퍼 포인터.

Return Value:

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

PCI_I2C_SYS_Read2

I2C를 통하여 데이터를 받는다. Repeated start 가능

BOOL **PCI_I2C_SYS_Read (int nBoard, int nCh, BYTE sIAddr, DWORD nAddrLen, DWORD nAddr, 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
- sIAddr : I2C address
- nAddrLen : Address Lengh
- nAddr : Register Address
- nCnt : 읽어 올 데이터의 바이트 개수.
- buf : 읽어 올 데이터의 버퍼 포인터.

Return Value:

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

PCI_SEN_Reset

센서를 초기화한다.

BOOL **PCI_SEN_Reset (int nBoard, int nCh, BOOL bReset)**

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
- bReset : "0" 이면 Disable
"1" 이면 Enable

Return Value:

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

PCI_SEN_Enable

센서 동작을 enable 시킨다.

BOOL PCI_SEN_Enable (int nBoard, int nCh, BOOL bEnable)

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
- bEnable : "0" 이면 Disable
 "1" 이면 Enable

Return Value:

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

PCI_SEN_Power

5V 전원을 On/Off 시킨다.

BOOL PCI_SEN_Power (int nBoard, int nCh, BOOL bOn)

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
- addr : "0" 이면 +5V Power Off
 "1" 이면 +5V Power On

Return Value:

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

Power Control API Functions

Overview

BOOL	PCI_PWR_IO_Voltage (int nBoard, int nCh, float fVoltage)
BOOL	PCI_PWR_IO_On (int nBoard, int nCh)
BOOL	PCI_PWR_IO_Off (int nBoard, int nCh)
BOOL	PCI_I2C_SEN_On (int nBoard, int nCh)
BOOL	PCI_I2C_SEN_Off (int nBoard, int nCh)

PCI_PWR_IO_Voltage

센서의 전압 값을 SendUserMsg 함수에 실어 보낸다.
(최소값은 1.25V).

BOOL **PCI_PWR_IO_Voltage (int nBoard, int nCh, float fVoltage)**

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
fVoltage : 전압 값.

Return Value:

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

PCI_PWR_IO_On

센서의 IO 전원을 enable 시킨다.

BOOL **PCI_PWR_IO_On (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_PWR_IO_Off

센서의 IO 전원을 disable 시킨다.

BOOL PCI_PWR_IO_Off (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_I2C_SEN_On

센서의 I2C를 enable 시킨다.

BOOL PCI_I2C_SEN_On (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_I2C_SEN_Off

센서의 I2C를 disable 시킨다.

BOOL PCI_I2C_SEN_Off (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.

