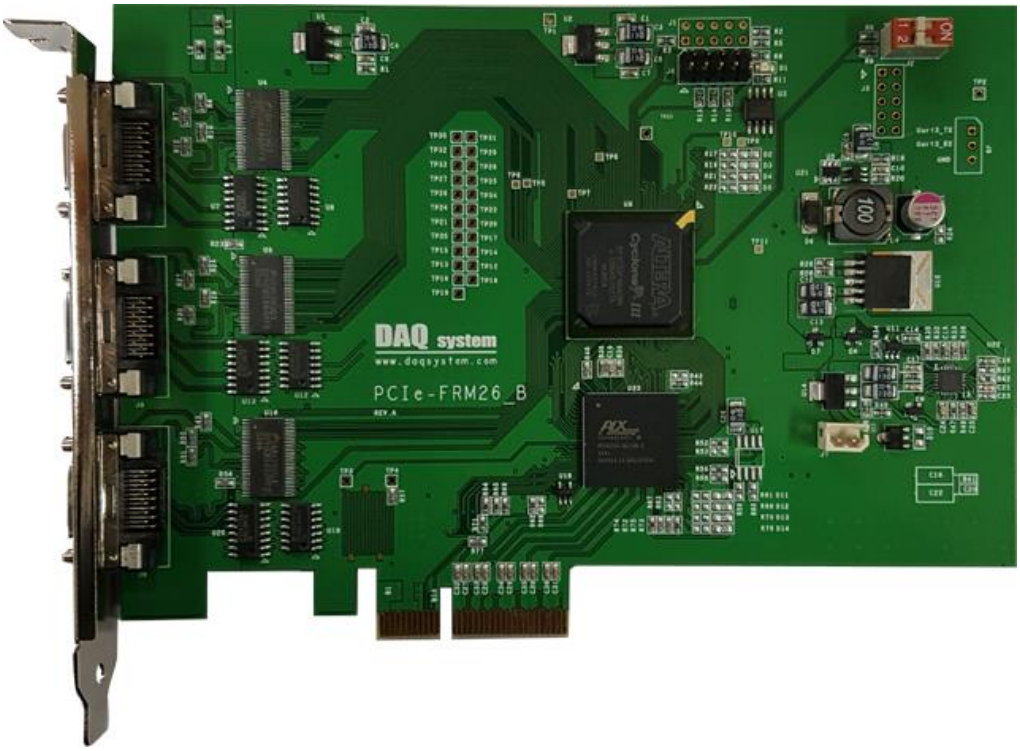
PCIe-FRM26_B

API 매뉴얼

버전 1.1



ⓒ **2005 DAQ SYSTEM Co., Ltd. All rights reserved.**

Microsoft® is a registered trademark; Windows®, Windows NT®, Windows XP®, Windows 7®, Windows 8®, Windows 10®
All other trademarks or intellectual property mentioned herein belongs to their respective owners.

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.

DAQ SYSTEM
Measurement & Automation

목 차

Board Level API Functions

OpenDAQDevice ---------- 4
CloseDAQDevice ---------- 4

LVDS(Camera Link) API Functions

LVDS_Init ---------- 5
LVDS_Start ---------- 6
LVDS_GetFrame ---------- 6
LVDS_Close ---------- 7
LVDS_Stop ---------- 7
LVDS_SetResolution ---------- 7
LVDS_GetResolution ---------- 8
LVDS_GetFrameSize ---------- 8
LVDS_SetDataMode ---------- 9
LVDS_GetVersion ---------- 9
LVDS_SetDeUse ---------- 10
LVDS_CC_Output ---------- 10
LVDS_SetTriggerOutput ---------- 11
LVDS_SetTrigger1Pulse ---------- 11
LVDS_SetTrigger2Pulse ---------- 12
LVDS_SetTrigger1Blank ---------- 12
LVDS_SetTrigger2Blank ---------- 13
LVDS_SetTrigger1Inv ---------- 13
LVDS_SetTrigger2Inv ---------- 14

UART API Functions

UART_Init ---------- 15
UART_GetData ---------- 15
UART_SendData ---------- 16
UART_Close ---------- 16
UART_SetBaud ---------- 17

DIO(Digital Input Output) API Functions

DIO_Read ---------- **18**
DIO_Write ---------- **18**

Board Level API Functions

Overview

int **OpenDAQDevice (int nBoard)**

BOOL **CloseDAQDevice (int nBoard)**

OpenDAQDevice

디바이스를 Open한다.

프로그램에서 초기에 반드시 한번 함수를 호출하여 디바이스를 Open하여야 한다.

Int **OpenDAQDevice (void)**

Parameters:

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

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.

CloseDAQDevice

Open된 디바이스를 Close한다.

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

BOOL **CloseDAQDevice (void)**

Parameters:

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

Return Value:

디바이스 Open에 성공할 경우 TRUE를 실패할 경우 FALSE를 리턴한다.

LVDS(Camera Link) API Functions

Overview

BOOL LVDS_Init (int nBoard, int nCh)
BOOL LVDS_Start (int nBoard, int nCh)
BOOL LVDS_GetFrame (int nBoard, int nCh, DWORD* nCnt, unsigned char* buf)
BOOL LVDS_Close (int nBoard, int nCh)
BOOL LVDS_Stop (int nBoard, int nCh)
BOOL LVDS_SetResolution (int nBoard, int nCh, DWORD xRes, DWORD yRes)
BOOL LVDS_GetResolution (int nBoard, int nCh, DWORD *xRes, DWORD *yRes)
BOOL LVDS_GetFrameSize (int nBoard, int nCh, DWORD *xRes, DWORD *yRes)
BOOL LVDS_SetDataMode (int nBoard, int nCh, int nMode)
BOOL LVDS_GetVersion (int nBoard, int nCh, int *nVersion)
BOOL LVDS_SetDeUse (int nBoard, int nCh, BOOL bUse)
BOOL LVDS_CC_Output (int nBoard, int nCh, DWORD dwVal)
BOOL LVDS_SetTriggerOutput (int nBoard, int nCh, DWORD dwValue)
BOOL LVDS_SetTrigger1Pulse (int nBoard, int nCh, int nPulseCount, int nPulseWidth)
BOOL LVDS_SetTrigger2Pulse (int nBoard, int nCh, int nPulseCount, int nPulseWidth)
BOOL LVDS_SetTrigger1Blank (int nBoard, int nCh, int nPulseBlank)
BOOL LVDS_SetTrigger2Blank (int nBoard, int nCh, int nPulseBlank)
BOOL LVDS_SetTrigger1Inv (int nBoard, int nCh, BOOL bUse)
BOOL LVDS_SetTrigger2Inv (int nBoard, int nCh, BOOL bUse)

LVDS_Init

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

BOOL LVDS_Init (int nBoard)

Parameters:

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

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 (int nBoard, int nCh)

Parameters:

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

nCh : 채널을 선택한다. 0 : 0번 채널, 1: 1번 채널, 2: 2번 채널

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

Parameters:

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

nCh : 채널을 선택한다. 0 : 0번 채널, 1: 1번 채널, 2: 2번 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 실패한 경우 'FALSE', 성공일 경우 'TRUE'를 return한다.

LVDS_Close

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

BOOL LVDS_Close (int nBoard, int nCh)

Parameters:

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

nCh : 채널을 선택한다. 0 : 0번 채널, 1: 1번 채널, 2: 2번 채널

Return Value:

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

LVDS_Stop

BOOL LVDS_Stop (int nBoard, int nCh)

프레임 데이터 Capture를 중지한다.

Parameters:

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

nCh : 채널을 선택한다. 0 : 0번 채널, 1: 1번 채널, 2: 2번 채널

Return Value :

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

LVDS_SetResolution

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

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

Parameters:

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

nCh : 채널을 선택한다. 0 : 0번 채널, 1: 1번 채널, 2: 2번 채널

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

Video 입력의 해상도를 가져온다.

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

Parameters:

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

nCh : 채널을 선택한다. 0 : 0번 채널, 1: 1번 채널, 2: 2번 채널

*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_GetFrameSize

Video 입력의 프레임 사이즈를 가져온다.

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

Parameters:

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

nCh : 채널을 선택한다. 0 : 0번 채널, 1: 1번 채널, 2: 2번 채널

*xRes : 수평해상도의 크기 값.

*yRes : 수직해상도의 크기 값.

Return Value:

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

LVDS_SetDataMode

이미지 픽셀 데이터 모드를 설정한다.

BOOL LVDS_SetDataMode (int nBoard, int nCh, int nMode)

Parameters:

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

nCh : 채널을 선택한다. 0 : 0번 채널, 1: 1번 채널, 2: 2번 채널

nMode : "2" 이면 24bit Mode이고, "Others" 이면 16bit Mode 이다.

Return Value:

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

LVDS_GetVersion

FPGA version을 가져온다.

BOOL LVDS_GetVersion (int nBoard, int nCh, int *nVersion)

Parameters:

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

nCh : 채널을 선택한다. 0 : 0번 채널, 1: 1번 채널, 2: 2번 채널

nVersion : FPGA 버전의 포인터.

Return Value:

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

LVDS_SetDeUse

데이터 유효(Data Valid) 신호선을 선택한다.

BOOL LVDS_SetDUse (int nBoard, int nCh, BOOL bUse)

Parameters:

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

nCh : 채널을 선택한다. 0 : 0번 채널, 1: 1번 채널, 2: 2번 채널

bUse : "True" 면 DVAL(Data Validation) 사용하고
"False" 면 HSYNC (Horizontal Synchronization)를 사용한다

Return Value :

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

LVDS_CC_Output

해당 비트의 CC 값을 출력한다.

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

Parameters:

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

nCh : 채널을 선택한다. 0 : 0번 채널, 1: 1번 채널, 2: 2번 채널

dwVal : 32bit 값으로 상위비트만 유효하다.
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_SetTriggerOutput

해당 비트의 CC 값을 출력한다.

BOOL LVDS_SetTriggerOutput (int nBoard, int nCh, DWORD dwValue)

Parameters:

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

nCh : 채널을 선택한다. 0 : 0번 채널, 1: 1번 채널, 2: 2번 채널

dwValue : bit0(CC1 out) = "0" : output "0" / "1": output "1"
bit1(CC2 out) = "0" : output "0" / "1": output "1"
bit2(CC3 out) = reserved
bit3(CC4 out) = reserved

Return Value :

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

LVDS_SetTrigger1Pulse

트리거1(CC1)의 출력 지연, 출력 폭을 선택한다.

BOOL LVDS_Set Trigger1Pulse (int nBoard, int nCh, DWORD nPulseCount, DWORD nPulseWidth)

Parameters:

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

nCh : 채널을 선택한다. 0 : 0번 채널, 1: 1번 채널, 2: 2번 채널

nPulseCount : 출력 Trigger1의 Delay 값

nPulseWidth : 출력 Trigger1의 Width 값 .

Return Value :

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

LVDS_SetTrigger2Pulse

트리거2(CC2)의 출력 지연, 출력 폭을 선택한다.

BOOL LVDS_Set Trigger2Pulse (int nBoard, int nCh, DWORD nPulseCount, DWORD nPulseWidth)

Parameters:

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

nCh : 채널을 선택한다. 0 : 0번 채널, 1: 1번 채널, 2: 2번 채널

nPulseCount : 출력 Trigger2의 Delay 값

nPulseWidth : 출력 Trigger2의 Width 값 .

Return Value :

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

LVDS_SetTrigger1Blank

트리거1(CC1)의 출력 간격을 선택한다.

BOOL LVDS_Set Trigger1Blank (int nBoard, int nCh, DWORD nPulseBlank)

Parameters:

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

nCh : 채널을 선택한다. 0 : 0번 채널, 1: 1번 채널, 2: 2번 채널

nPulseBlank : 출력 Trigger1의 Blank(간격) 값 .

Return Value :

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

LVDS_SetTrigger2Blank

트리거2(CC2)의 출력 간격을 선택한다.

BOOL LVDS_Set Trigger2Blank (int nBoard, int nCh, DWORD nPulseBlank)

Parameters:

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

nCh : 채널을 선택한다. 0 : 0번 채널, 1: 1번 채널, 2: 2번 채널

nPulseBlank : 출력 Trigger2의 Blank(간격) 값 .

Return Value :

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

LVDS_SetTrigger1Inv

Trigger1(CC1)의 pulse를 반전한다.

BOOL LVDS_SetTrigger1Inv (int nBoard, int nCh, BOOL bUse)

Parameters:

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

nCh : 채널을 선택한다. 0 : 0번 채널, 1: 1번 채널, 2: 2번 채널

bUse : "True" 면 Negative Pulse를 사용하고
"False" 면 Positive Pulse를 사용한다

Return Value :

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

LVDS_SetTrigger2Inv

Trigger2(CC2)의 pulse를 반전한다.

BOOL LVDS_SetTrigger2Inv (int nBoard, int nCh, BOOL bUse)

Parameters:

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

nCh : 채널을 선택한다. 0 : 0번 채널, 1: 1번 채널, 2: 2번 채널

bUse : "True" 면 Negative Pulse를 사용하고
"False" 면 Positive Pulse를 사용한다

Return Value :

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

UART API Functions

Overview

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

UART_Init

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

BOOL UART_Init (int nBoard, int nCh)

Parameters:

nBoard : 현재 시스템에 장착되어 있는 보드 번호를 알려준다.
보드 번호는 보드의 DIP스위치를 이용하여 설정한다.
nCh : 채널을 선택한다. 0 : 0번 채널, 1: 1번 채널, 2: 2번 채널

Return Value:

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

UART_GetData

차동 UART를 통하여 문자를 받는다.

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

Parameters:

nBoard : 현재 시스템에 장착되어 있는 보드 번호를 알려준다.
보드 번호는 보드의 DIP스위치를 이용하여 설정한다.
nCh : 채널을 선택한다. 0 : 0번 채널, 1: 1번 채널, 2: 2번 채널
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

차동 UART를 통하여 문자를 보낸다.

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

Parameters:

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

nCh : 채널을 선택한다. 0 : 0번 채널, 1: 1번 채널, 2: 2번 채널

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

UART 함수에서 사용한 모든 자원을 되돌려 준다.

BOOL UART_Close (int nBoard, int nCh)

Parameters:

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

nCh : 채널을 선택한다. 0 : 0번 채널, 1: 1번 채널, 2: 2번 채널

Return Value:

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

UART_SetBaud

UART Baud를 정한다.

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

Parameters:

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

nCh : 채널을 선택한다. 0 : 0번 채널, 1: 1번 채널, 2: 2번 채널

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

Return Value:

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

DIO(Digital Input Output) API Functions

Overview

BOOL DIO_Read (int nBoard, int nCh, DWORD* Val)

BOOL DIO_Write (int nBoard, int nCh, DWORD Val)

DIO_Read

32bit Digital 입력 값을 읽는다.

BOOL DIO_Read (int nBoard, int nCh, DWORD* Val)

Parameters:

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

nCh : 채널을 선택한다. 0 : 0번 채널, 1: 1번 채널, 2: 2번 채널

*dwVal : 입력 포트의 현재 값을 읽어올 변수이다.

Return Value:

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

DIO_Write

32bit Digital 값을 출력 포트에 출력한다.

BOOL DIO_Write (int nBoard, int nCh, DWORD Val)

Parameters:

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

nCh : 채널을 선택한다. 0 : 0번 채널, 1: 1번 채널, 2: 2번 채널

dwVal : 출력 포트에 기록할 값.

Return Value:

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

Memo

Contact Point

Web sit : https://www.daqsystem.com

Email : postmaster@daqsystem.com