

# PCIe-HDMI01

## API 매뉴얼

버전 1.0



© 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.

# 목 차

## Board Level API Functions

OpenDAQDevice	-----	3
CloseDAQDevice	-----	3
GetDIIVersion	-----	3

## LVDS API Functions

LVDS_GetVersion	-----	4
LVDS_Init	-----	4
LVDS_Close	-----	5
LVDS_Start	-----	5
LVDS_Stop	-----	5
LVDS_GetFrame	-----	6
LVDS_GetFrameRate	-----	6
LVDS_GetResolution	-----	7
LVDS_DdrUse	-----	7

## HDMI API Functions

HDMI_SetDataMode	-----	8
HDMI_Reset	-----	8
HDMI_SetPort	-----	9

## Board Level APIs

### Overview

Int	OpenDAQDevice (void)
BOOL	CloseDAQDevice (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 번호를 리턴한다.

함수 호출에 실패한 경우, "-1"을 리턴한다. 이것의 의미는 시스템에 장치가 없다는 의미이다.

### CloseDAQDevice

Open된 디바이스를 Close한다. 장치의 사용이 끝나게 되면, 반드시 장치를 Close 하여 다른 프로그램에서 사용할 수 있도록 한다.

**BOOL**                    **CloseDAQDevice (void)**

**Parameters:** 없음.

**Return Value:**

디바이스 Open에 성공할 경우 TRUE를 실패할 경우 FALSE를 리턴한다.

### GetDllVersion

이 함수는 설치되어 있는 현재 DLL 버전을 알려준다.

**Char\***                    **GetDllVersion (void)**

**Parameters:** 없음.

**Return Value:**

설치되어 있는 DLL의 날짜가 넘어 온다. 예)Feb 15 2021

## LVDS API Functions

### Overview

BOOL	LVDS_GetVersion (int nBoard, int *nFpga)
BOOL	LVDS_Init (int nBoard)
BOOL	LVDS_Close (int nBoard)
BOOL	LVDS_Start (int nBoard)
BOOL	LVDS_Stop (int nBoard)
BOOL	LVDS_GetFrame (int nBoard, DWORD* nCnt, unsigned char* buf)
BOOL	LVDS_GetFrameRate (int nBoard, DWORD* nCnt, unsigned char* buf)
BOOL	LVDS_GetResolutuion (int nBoard, DWORD *xRes, DWORD *yRes)
BOOL	LVDS_DdrUse (int nBoard, int nCh, BOOL bUse)

### LVDS\_GetVersion

이 함수는 FPGA 버전을 알려준다.

**BOOL** LVDS\_GetVersion (int nBoard, int \*nFpga)

#### Parameters:

- nBoard : 현재 시스템에 장착되어 있는 보드 번호를 알려준다.  
보드 번호는 보드의 DIP 스위치를 이용하여 설정한다.
- \*nFpga : FPGA Version 값을 얻어올 번지.

#### Return Value:

"-1" : 최대(4대) 장치 초과, "-2" : 장치가 열리지 않음,

### LVDS\_Init

LVDS 함수에서 사용된 모든 자원을 초기화한다.

**BOOL** LVDS\_Init (int nBoard)

#### Parameters:

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

#### Return Value:

"-1" : 최대(4대) 장치 초과, "-2" : 장치가 열리지 않음,

## LVDS\_Close

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 DIP 스위치를 이용하여 설정한다.

#### Return Value:

"-1" : 최대(4대) 장치 초과, "-2" : 장치가 열리지 않음,

## LVDS\_Start

LVDS\_Start 함수 호출이 되면 Board에서 DLL로 Data를 전송 시작한다.  
(Thread를 통하여 DLL의 Buffer에 Data를 쌓고, LVDS\_GetFrame 함수로 DLL -> APP 로 copy 한다.)

### BOOL LVDS\_Start (int nBoard)

#### Parameters:

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

#### Return Value:

"-1" : 최대(4대) 장치 초과, "-2" : 장치가 열리지 않음,

## LVDS\_Stop

프레임 데이터 Capture를 중지한다.

### BOOL LVDS\_Stop (int nBoard)

#### Parameters:

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

#### Return Value :

"-1" : 최대(4대) 장치 초과, "-2" : 장치가 열리지 않음,

## 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, DWORD\* nCnt, unsigned char\* buf)

**Parameters:**

nBoard : 현재 시스템에 장착되어 있는 보드 번호를 알려준다.

보드 번호는 보드의 DIP 스위치를 이용하여 설정한다.

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

함수를 호출할 때 버퍼 크기를 지정하고 호출한 후에는 변수 값을 읽어서 실제로 읽어 온 개수를 확인한다. 데이터 크기는 바이트 단위이다.

buf : 프레임 버퍼 포인터.

**Return Value:**

"-1" : 최대(4대) 장치 초과, "-2" : 장치가 열리지 않음,

"-3" : 잘못된 파라미터

## LVDS\_GetFrameRate

실제 H/W 적으로 받은 초당 프레임 개수를 보여준다.

**BOOL** LVDS\_GetFrameRate (int nBoard, DWORD \*dwRate)

**Parameters:**

nBoard : 현재 시스템에 장착되어 있는 보드 번호를 알려준다.

보드 번호는 보드의 DIP 스위치를 이용하여 설정한다.

dwRate : 프레임 개수 번지

**Return Value:**

"-1" : 최대(4대) 장치 초과, "-2" : 장치가 열리지 않음,

## LVDS\_GetResolution

Video 입력의 해상도를 가져온다.

**BOOL** LVDS\_GetResolutuion (int nBoard, DWORD \*xRes, DWORD \*yRes)

**Parameters:**

nBoard : 현재 시스템에 장착되어 있는 보드 번호를 알려준다.

보드 번호는 보드의 DIP스위치를 이용하여 설정한다.

\*nXres : 수평해상도 값 번지.

\*nYres : 수직해상도 값 번지.

**Return Value:**

"-1" : 최대(4대) 장치 초과, "-2" : 장치가 열리지 않음,

"-3" : 잘못된 파라미터

## LVDS\_DdrUse

이 함수는 DDR 메모리 사용 유무를 결정한다.

**BOOL** PCI\_LVDS\_DdrUse (int nBoard, BOOL bUse)

**Parameters:**

nBoard : 현재 시스템에 장착되어 있는 보드 번호이다.

보드 번호는 보드의 DIP 스위치 설정(0~ 3)과 일치한다.

bUse : "0" 이면 DDR 메모리를 사용하지 않고

"1" 이면 DDR 메모리를 사용한다

**Return Value :**

"-1" : 최대(4대) 장치 초과, "-2" : 장치가 열리지 않음,

"-3" : 잘못된 파라미터

## HDMI API Functions

### Overview

- BOOL HDMI\_SetDataMode (int nBoard, int nRES\_SET, int nRGB\_YUV\_SET, int nDATA\_BIT\_SET)
- BOOL HDMI\_Reset (int nBoard)
- BOOL HDMI\_SetPort (int nBoard, int nPort)

### HDMI\_SetDataMode

HDMI 칩의 해상도(Resolution), 영상 포맷(YUV 또는 RGB), 영상 Data Bit를 설정한다.

- BOOL CLK\_Select (int nBoard, int nRES\_SET, int nRGB\_YUV\_SET, int nDATA\_BIT\_SET)

**Parameters:**

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

[표 1. Parameter Selection Table]

Parameter	Selection					
nRES_SET	"0" : 480_570		"1" : 720_1080		"2" : 4K_2K	
nRGB_YUV_SET	"0" : YCbCr422	"1" : RGB_444	"0" : YCbCr422	"1" : RGB_444	"0" : YCbCr422	"1" : RGB_444
nDATA_BIT_SET	"0" : 24bits RGB	"0" : 24bits RGB	"0" : 24bits RGB	"0" : 24bits RGB	"0" : YCbCr444_422	"0" : YCbCr444_422
	"1" : 30bits RGB	"1" : 30bits RGB	"1" : 30bits RGB	"1" : 30bits RGB		
	"2" : 16bits YUV,	"2" : 16bits YUV,	"2" : 16bits YUV,	"2" : 16bits YUV,		
	"3" : 20bits YUV	"3" : 20bits YUV	"3" : 20bits YUV	"3" : 20bits YUV		
	"4" : 24bits YUV	"4" : 24bits YUV	"4" : 24bits YUV	"4" : 24bits YUV		

**Return Value:**

- "-1" : 최대(4대) 장치 초과, "-2" : 장치가 열리지 않음,
- "-3" : 잘못된 파라미터



## HDMI\_Reset

HDMI 칩의 설정을 초기화 한다.

### BOOL HDMI\_Reset (int nBoard)

**Parameters:**

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

**Return Value:**

"-1" : 최대(4대) 장치 초과, "-2" : 장치가 열리지 않음,

## HDMI\_SetPort

보드의 HDMI 포트를 Port A(J1 커넥터) 또는 Port B(J2 커넥터)로 할지를 설정한다.

### BOOL HDMI\_SetPort (int nBoard, int nPort)

**Parameters:**

nBoard : 현재 시스템에 장착되어 있는 보드 번호를 알려준다.  
보드 번호는 보드의 DIP 스위치를 이용하여 설정한다.  
nPort : "0" 이면 : Port A , '1" 이면 Port B

**Return Value:**

"-1" : 최대(4대) 장치 초과, "-2" : 장치가 열리지 않음,

---

# Memo

## Contact Point

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

Email : [postmaster@daqsystem.com](mailto:postmaster@daqsystem.com)

