DVP-1412

1 Channel MPEG 1/2/4 Digital Video Encoder Module with 128-byte EEPROM, USB 2.0 interface

User Manual

Acknowledgements
IBM and PC are trademarks of International Business Machines Corporation. MS-DOS, Windows, Microsoft Visual C++ and Visual BASIC are trade-marks of Microsoft Corporation. Intel and Pentium are trademarks of Intel Corporation. DivX® is a registered trademark of DivX Networks, Inc. All other product names or trademarks are the properties of their respective owners.

On-line Technical Support
For technical support and service, please visit our support website at:
http://www.advantech.com/support

1nd. Edition
Printed in Taiwan January 2005

0.0.1 A Message to the Customer

Advantech customer services

Each and every Advantech product is built to the most exacting specifications to ensure reliable performance in the harsh and demanding conditions typical of industrial environments. Whether your new Advantech equipment is destined for the laboratory or the factory floor, you can be assured that your product will provide the reliability and ease of operation for which the name Advantech has come to be known. Your satisfaction is our primary concern. Here is a guide to Advantech's customer services. To ensure you get the full benefit of our services, please follow the instructions below carefully.

Technical support

We want you to get the maximum performance from your products. So if you run into technical difficulties, we are here to help. For the most frequently asked questions, you can easily find answers in your product documentation. These answers are normally a lot more detailed than the ones we can give over the phone.

So please consult this manual first. If you still cannot find the answer, gather all the information or questions that apply to your problem, and with the product close at hand, call your dealer. Our dealers are well trained and ready to give you the support you need to get the most from your Advantech products. In fact, most problems reported are minor and are able to be easily solved over the phone.

In addition, free technical support is available from Advantech engineers every business day. We are always ready to give advice on application requirements or specific information on the installation and operation of any of our products.

Table 0.1 DVP1412 Specification

Video / Audio	Input connectors	BNC or pin-header
	Video Standard	NTSC / PAL
	Resolution*	Full D1 / VGA / QVGA / CIF / QCIF
	Frame Rate	Up to 30fps (NTSC) / 25pfs (PAL)
	Video Encoding	MPEG-1, MPEG-2, MPEG-4
	Data Output format	MPEG-4 Simple Profile @ L3 plus B-picture support MPEG-2 MP @ ML MPEG-1
	Audio output	Stereo input, PCM format data stream output
Host Communication	Host interface	High-speed USB 2.0
	Max. module	Up to 8 modules in one PC system, set by DIP switch
	Driver and SDK	Supports Microsoft Windows® 2000 / XP
	Demo program	Complete demo program with C++ source code for reference
Hardware	EEPROM	128 bytes, includes SDK for programming
	Dimensions (WxL)	70mm x 38 mm
	Power consumption	+5V DC input by USB bus with standard 500mA supply
	Temperature	0 ~ 60 ºC (32 ~ 140 ºF; Operating) -20 ~ 70 ºC (-4 ~ 158 ºF; Non-Operating)

Table 0.2 Video Resolution Index

Abbreviate	**Technical Terms / Video System**	**NTSC**	**PAL**
D1 (DVD Quality)	Full resolution	720x480	720x576
VGA	**V**ideo **G**raphics **A**rray	640x480	
QVGA	**Q**uarter **VGA**	320x240	
CIF (VCD Quality)	**C**ommon **I**ntermediate **F**ormat	352x240	352x288
QCIF	**Q**uarter **CIF**	176x120	176x144

0.0.2 Product warranty

Advantech warrants to you, the original purchaser, that each of its products will be free from defects in materials and workmanship for two years from the date of purchase.

This warranty does not apply to any products which have been repaired or altered by persons other than repair personnel authorized by Advantech, or which have been subject to misuse, abuse, accident or improper installation. Advantech assumes no liability under the terms of this warranty as a consequence of such events.

If an Advantech product is defective, it will be repaired or replaced at no charge during the warranty period. For out-of-warranty repairs, you will be billed according to the cost of replacement materials, service time and freight. Please consult your dealer for more details.

If you think you have a defective product, follow these steps:

Step 1. Collect all the information about the problem encountered. (For example, type of PC, CPU speed, Advantech products used, other hardware and software used, etc.) Note anything abnormal and list any on-screen messages you get when the problem occurs.

Step 2. Call your dealer and describe the problem. Please have your manual, product, and any helpful information readily available.

Step 3. If your product is diagnosed as defective, obtain an RMA (return material authorization) number from your dealer. This allows us to process your return more quickly.

Step 4. Carefully pack the defective product, a fully-completed Repair and Replacement Order Card and a photocopy proof of purchase date (such as your sales receipt) in a shippable container. A product returned without proof of the purchase date is not eligible for warranty service.

Step 5. Write the RMA number visibly on the outside of the package and ship it prepaid to your dealer.

0.0.3 Initial Inspection

Before you begin installing your motherboard, please make sure that the following materials have been shipped:

- 1 * DVP-1412 video encoder card
- 1 * CD with driver utility , SDK and user manual (in PDF format)
- 1 * internal USB cable

If any of these items are missing or damaged, contact your distributor or sales representative immediately.

We have carefully inspected theDVP-1412 mechanically and electrically before shipment. It should be free of marks and scratches and in perfect working order upon receipt.

As you unpack the DVP-1412, check it for signs of shipping damage. (For example, damaged box, scratches, dents, etc.) If it is damaged or it fails to meet the specifications, notify our service department or your local sales representative immediately. Also notify the carrier. Retain the shipping carton and packing material for inspection by the carrier. After inspection, we will make arrangements to repair or replace the unit.

0.0.4 Release Note

Date	Revision	Change
January 2005	1st. Edition	Initial Release & support MPEG4 only

Table of Content

CHAPTER 1

General Information

Chapter 1 General Information

Designed for embedded application or upgrade with digital video function market, the DVP-1412 is a very compact-size, hardware MPEG4 encode embedded module which is very easy integrate with most of equipment such as Panel PC, Industrial Embedded controller. It can accept standard composite video input through BNC or pin-header. The DVP-1412 provides complete SDK and driver on MS-Windows 2000/XP/XP embedded environment.

By using a standard high-speed USB 2.0 interface, the DVP-1412 is very easy to work with most of Single Board Computer or PC system. Due to the hardware MPEG 1/2/4 video encode engine, the DVP-1412 can reduce the main CPU's loading and programmer's coding time. The DVP-1412 is also supporting up to 30/25 fps (NTSC/PAL) at D1 resolution. By the dynamically adjustable bit rate and frame rate to accommodate variable bandwidths, DVP-1412 can optimize the quality and bandwidth at the best. Through on board's 128 Bytes EEPROM space, programmer can place the protection code or system parameter on it via our SDK. The DVP-1412 is a ideal embedded digital video solution for most of applications such as DVR (Digital Video Recorder), PVR (Personal Video Recorder), Video phone or others embedded with video function equipments.. For extension issue, DVP-1412 can be up to 8 modules in one PC system by setting DIP switch.. Please reference the jumper / connector location for the location of module number.

1.1 Hardware Requirement

- Intel Pentium 1.0G Hz or above (The CPU speed is depends on the video frame rate, channels and resolution)
- 128MB RAM or above
- USB 2.0 host port
- CD-ROM
- Hard disk with 128MB free space

1.2 Software Requirement

- Microsoft Windows 2000/XP with DirectX 8.1 or above

1.3 Block Diagram

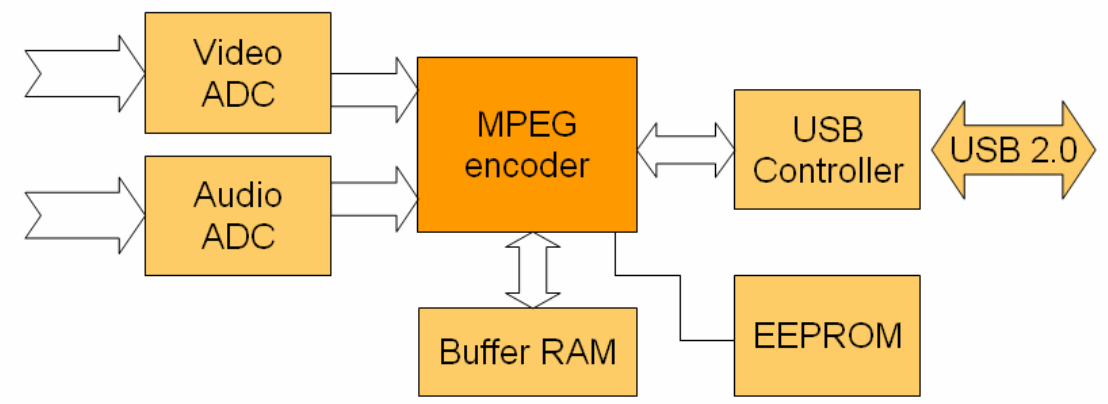


Figure 1.1 DVP 1412 block diagram

1.4 Dimension

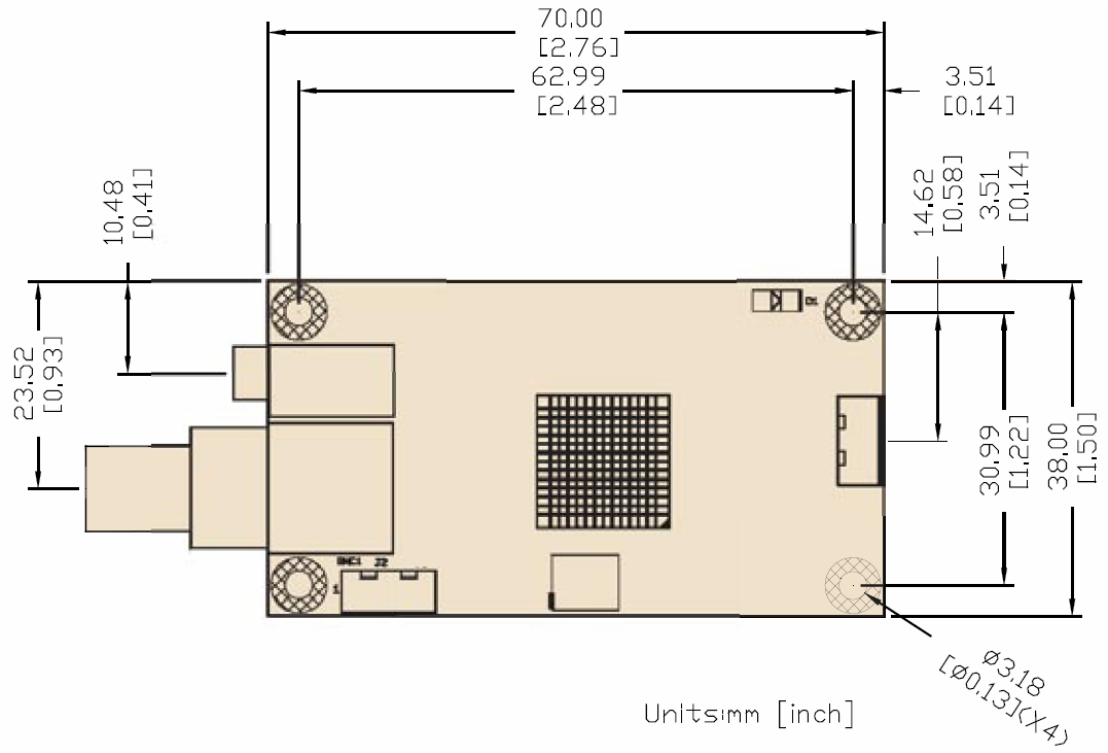


Figure 1.2 DVP-1412 Dimension Diagram

1.5 Jumper/Connector Location

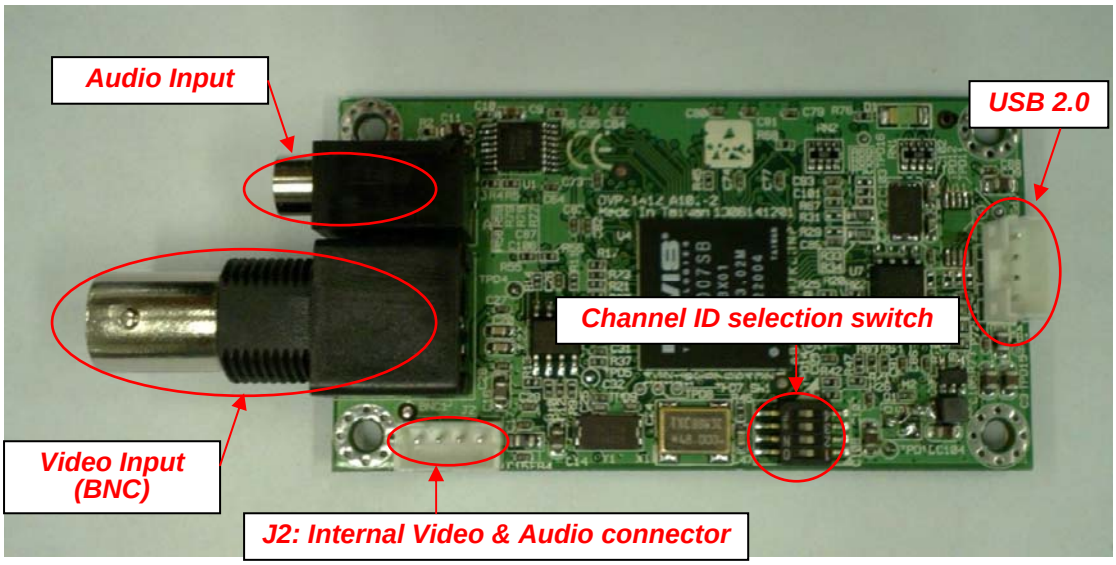


Figure 1.3 DVP-1412 Jumper & Connector location

1.6 Channel/Module ID selection switch

The channel switch is a DIP switch which can set the channel/module ID before installation.

Table 1.1 Channel/Module ID selection table

Channel	SW1	SW2	SW3	SW4 (reserved)
01	Off	Off	Off	Off (default)
02	On	Off	Off	Off (default)
03	Off	On	Off	Off (default)
04	On	On	Off	Off (default)
05	Off	Off	On	Off (default)
06	On	Off	On	Off (default)
07	Off	On	On	Off (default)
08	On	On	On	Off (default)

1.7 Pin-definition description

1.7.1 Internal video & audio input

The J2 video & audio input is a internal input which parallel with the BNC & audio connectors for board camera applications.

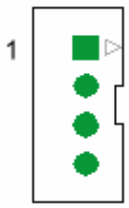


Table 1.2 Internal Video & Audio input			
Pin	**Signal**	**Pin**	**Signal**
1	Video in	3	Audio in - Right channel
2	Audio in - Left channel	4	GND

1.7.2 USB to host board interface

The internal USB connector is a USB 2.0 device interface which connected to host controller such as PCM series biscuit boards or PICMG CPU cards.

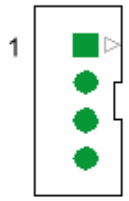


Table 1.3 USB 2.0 interface			
Pin	**Signal**	**Pin**	**Signal**
1	+5V	3	Data (+)
2	Data (-)	4	GND

1.8 Software / Driver Installation

Before you begin

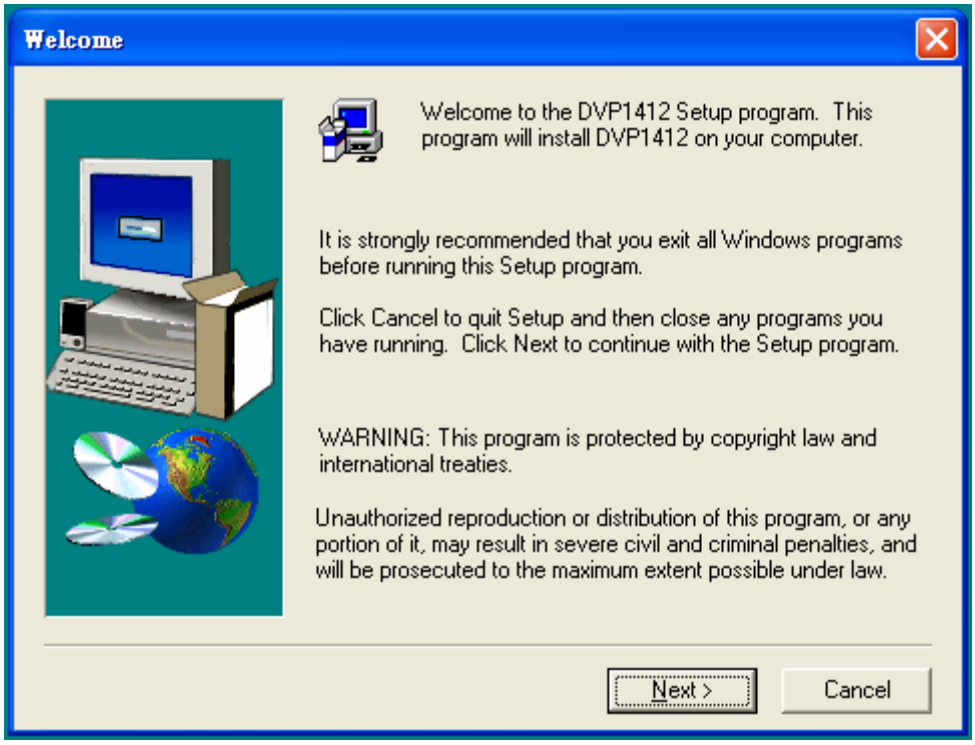
To facilitate the installation of the enhanced display device drivers and utility software, you should read the instructions in this chapter carefully before you attempt installation. The drivers for the DVP-1412 board are located on the software installation CD. Before you begin,

it is important to note that most display drivers need to have the relevant software application already installed in the system prior to installing the enhanced display drivers. In addition, many of the installation procedures assume that you are familiar with both the relevant software applications and operating system commands. Review the relevant operating system commands and the pertinent sections of your application software user manual before performing the installation.

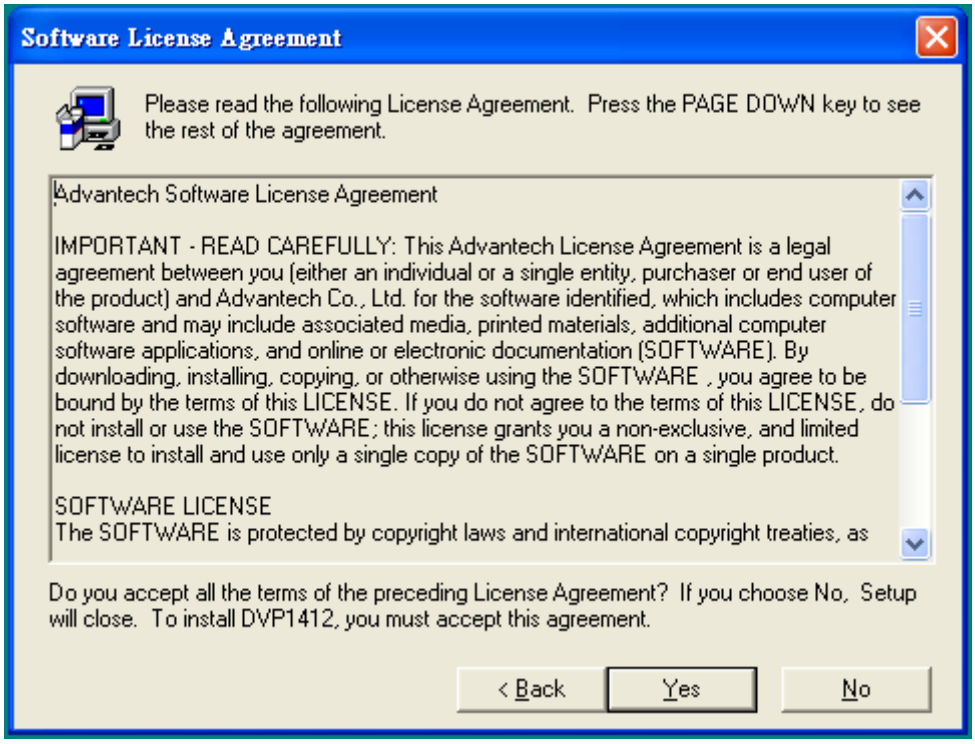
Installation Step

1. Insert the driver CD into your system's CD-ROM drive. Go to the SDK_Driver folder of DVP-1412 CD. Move the mouse cursor over the "Setup.exe" & double-click it. Then, a message pops up telling you to start the installation.

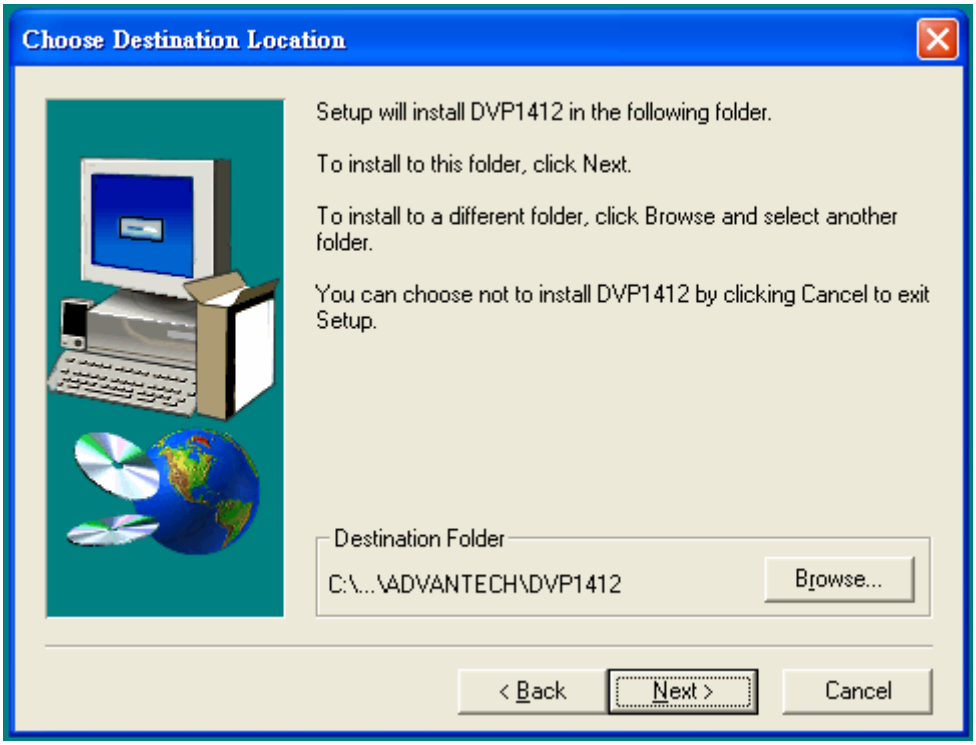
2. Click "Next" when you see the following message.



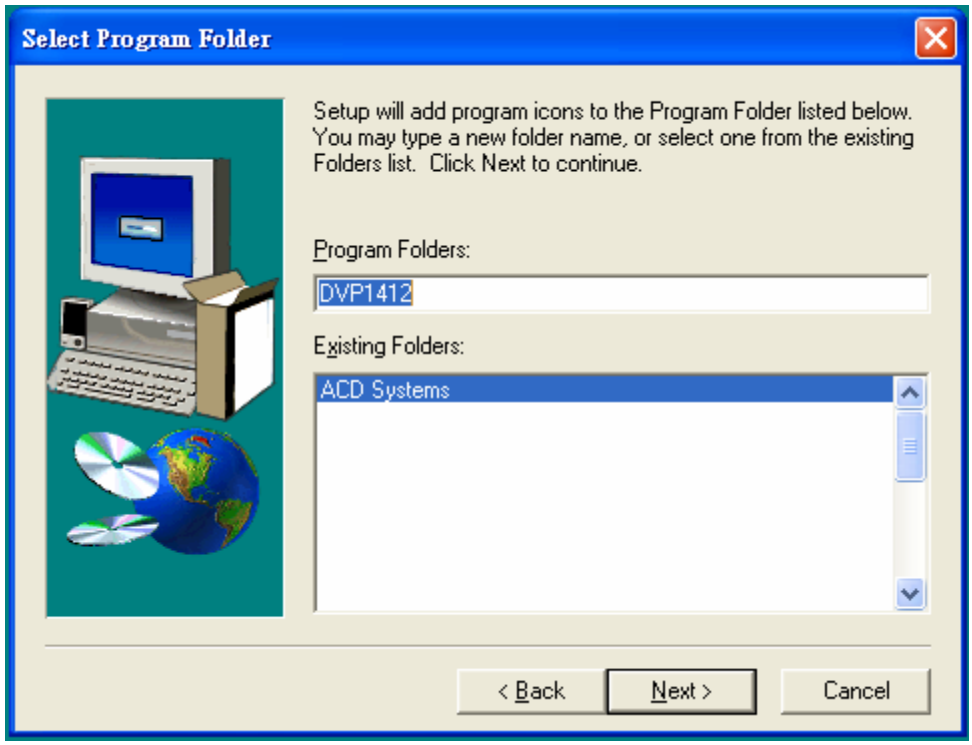
3. Please read the following license agreement and select "Yes" or "No" to next status.



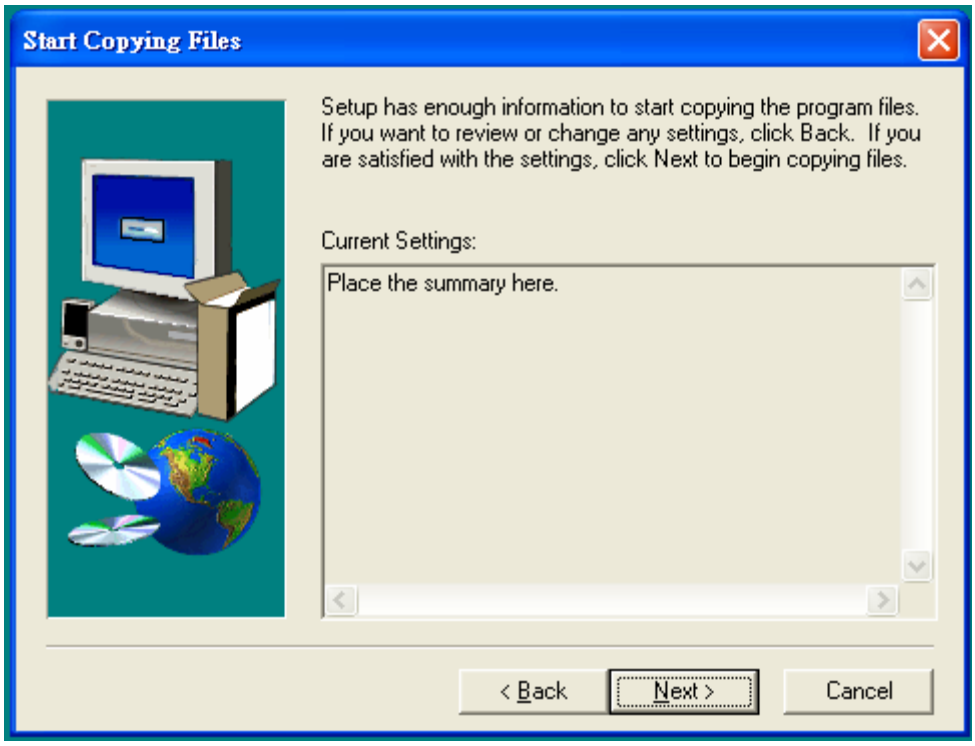
4. Please choose the destination folder and Click "Next".



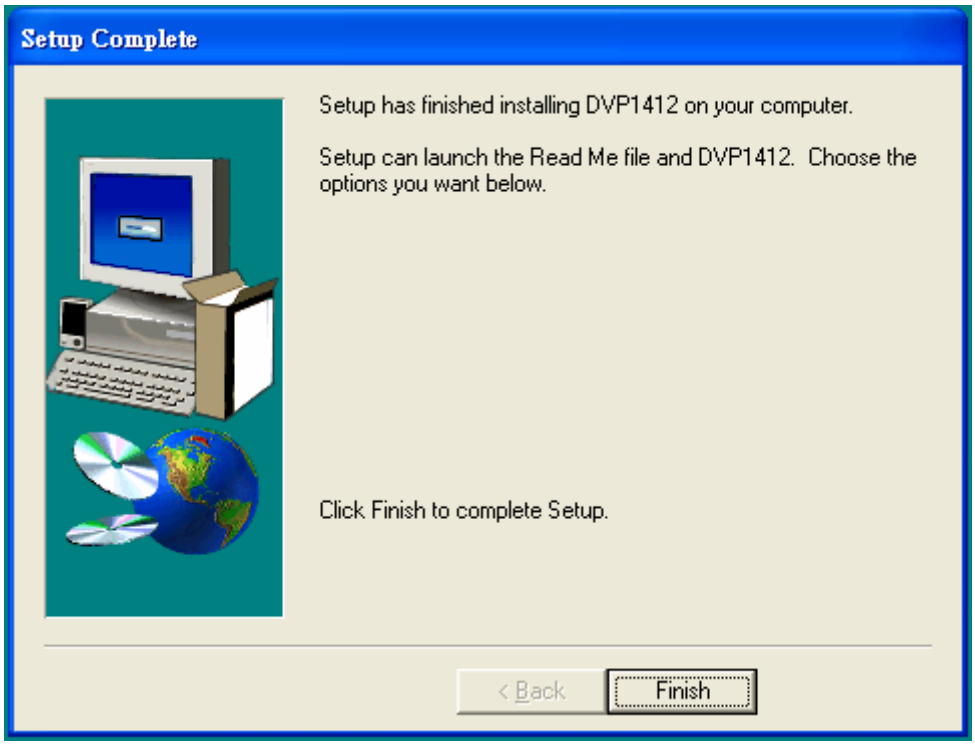
5. When the following message appears, give a name to the program folder, then click “Next” to install.



6. Click "Next" when you see the following message.



7. When the following message appears, click "Finish" to complete the installation and restart Windows.



1.9 Hardware Installation

1. Turn off your computer and unplug the power cord.
2. Remove the cover of your computer.
3. Touch the metal part on the surface of your computer to neutralize the static electricity that might be on your body.
4. Setting the channel/module ID switch (SW1) while you want.
5. Place the DVP-1412 into chassis and using screws to fix it.
6. Plug the USB cable into DVP-1412 USB connector and host port with 500mA capability.
7. Connect appropriate accessories (Video cable to camera. if necessary) to the DVP-1412 module.
8. Replace the cover of your computer chassis.
9. Plug in the power cord and turn on the computer.
10. Computer will detect the new hardware & manually set the driver destination folder to the same path of step 1.8 Software / Driver Installation.
    For example, “C:\Program Files\ADVANTECH\DVP1412”

Note:	Keep the anti-static bag for future use. You might need the original bag to store the card if you have to remove the card from the PC or transport it elsewhere.

1.10 MPEG 4 Codec Installation

For review the video encoded by DVP-1412, users need MPEG4 codex installation first. Free MPEG4 codex can be download at http://www.divx.com

Or users can install the DivX codex from the DVP-1412 utility disc. Please use the DivX codex of version 5.0 or above.

CHAPTER 2

Functions Library

Chapter 2 Functions Library

Summary

SDK Initialize and close
- DVP1412_CreateSDKInstence
- DVP1412_GetNoOfDevices
- DVP1412_InitSDK
- DVP1412_CloseSDK

Capture control
- DVP1412_Start
- DVP1412_Stop
- DVP1412_GetCapState
- DVP1412_SetRecord
- DVP1412_IsVideoPresent
- DVP1412_GetStatistics

Capture setting
- DVP1412_GetCompressMode
- DVP1412_SetCompressMode
- DVP1412_GetVideoStandard
- DVP1412_SetVideoStandard
- DVP1412_GetResolution
- DVP1412_SetResolution
- DVP1412_GetFrameRate
- DVP1412_SetFrameRate
- DVP1412_GetBitRate
- DVP1412_SetBitRate
- DVP1412_GetSequenceMode
- DVP1412_SetSequenceMode
- DVP1412_GetMpeg4GOPSize
- DVP1412_SetMpeg4GOPSize

Sensor Control
- DVP1412_GetBrightness
- DVP1412_SetBrightness
- DVP1412_GetContrast
- DVP1412_SetContrast
- DVP1412_GetHue
- DVP1412_SetHue
- DVP1412_GetSaturation
- DVP1412_SetSaturation

EE Controller
- DVP1412_ReadEE
- DVP1412_WriteEE

Note:	*[DVP-1412 Sample program] After installation, DVP-1412 sample program is in the destination folder in driver installation step as file name "dvp1412exe.zip". User can un-zip the file & take those sample program files as reference as programming or developing.*

Functions Reference

Struct

DeviceSettingStruct

```
typedef struct{
        VideoStandard videoStandard;
        CompressMode compressMode;
        MPEG4Mode mpeg4Mode;
        SequenceMode sequenceMode;
        Resolution resolution;
        FrameRate frameRate;
        int bitRate;
        int gopSize;
} DeviceSettingStruct;
```

Parameters

videoStandard:	Current video standard setting.
compressMode:	Current compress mode setting.
mpeg4Mode:	Current mpeg4 mode setting.
sequenceMode:	Current sequence mode setting.
resolution:	Current resolution setting.
frameRate:	Current frame rate setting.
bitRate:	Current bit rate setting.
gopSize:	Current gop size setting.

Description

A stucrt stores current capture setting.

DeviceInfoStruct

```
typedef struct{
        DeviceSettingStruct setting;
        CapState capState;
} DeviceInfoStruct;
```

Parameters

setting:	Struct of current capture setting..
capState:	Current capture state.

Description

A stucrt stores all current capture information.

StatisticInfo

```
typedef struct{
        unsigned __int64 ByteCount;
        unsigned FrameCount;
} StatisticInfo;
```

Parameters

ByteCount:	Current byte count.
FrameCount:	Current frame count.

Description

A stucrt stores current statistic setting.

Method

DVP1412_CreateSDKInstence

Syntax
int DVP1412_CreateSDKInstence(void **pp)

Parameters
pp: A pointer to the SDK.

Return Value

SUCCEEDED:	Function succeeded.
PARAMERROR:	Parameter error.
SDKINITFAILED:	Failed to initialize SDK.

Description
This function creates SDK instance.

DVP1412_GetNumberOfDevices

Syntax
int DVP1412_GetNoOfDevices(void)

Parameters
None

Return Value
Number of DVP1412 digital video encoder module

Description
This function gets number of DVP1412 module in the system. At most 8 channels are available in a DVP1412 integrated system.

DVP1412_InitSDK

Syntax
int DVP1412_InitSDK(int NoOfDevs, int* IDList)

Parameters
NoOfDevs: Number of devices.
IDs: An array pointer stores all board IDs. Negative value indentifys inactive channel.

Return Value

SUCCEEDED:	Function succeeded.
FAILED:	Function failed.
BOARDIDERROR:	Failed to get board ID or duplicate board ID.
NODEVICES:	No devices found.
CHECKKEYERROR:	Failed to check hardware key.

Description
This function initializes all DVP1412 modules in the system and gets all board IDs. After initializing each module, the capture status would be set as “STOPPED”.

See Also
DVP1412_GetNoOfDevices
DVP1412_GetCapState
DVP1412_CloseSDK

DVP1412_CloseSDK

Syntax
int DVP1412_CloseSDK(void)

Parameters
None

Return Value

SUCCEEDED:	Function succeeded.
PARAMERROR:	Parameter error.
SDKINITFAILED:	SDK not initialized.

Description
This function cleans all instances of capture devices and closes up the SDK.

See Also
DVP1412_InitSDK

DVP1412_Start

Syntax
int DVP1412_Start(int BoardID, BOOL toPreview, BOOL toRecord, BOOL doAudio, HWND hwndPreview)

Parameters
BoardID: Specifies the board ID number(0~7).
toPreview: Set current capture board to preview video.
toRecord: Set current capture board to record video.
doAudio: Set current capture board to do audio.
hwndPreview:A windows handle for display area.

Return Value

SUCCEEDED:	Function succeeded.
FAILED:	Function failed.
BOARDIDERROR:	Invalid board ID.
SDKINITFAILED:	SDK not initialized.
FILTERERROR:	Filter not initialized.

Description
This function starts video capturing on a specified capture board. The capture state would be set as “PREVIEW”, “RECORDING”, “PREVIEWandRECORDING” after a successful start. A DivX decoder filter must be installed to perform video previewing.

See Also
DVP1412_Stop
DVP1412_GetCapState

DVP1412_Stop

Syntax
int DVP1412_Stop(int BoardID)

Parameters
BoardID: Specifies the board ID number(0~7).

Return Value

SUCCEEDED:	Function succeeded.
FAILED:	Function failed.
BOARDIDERROR:	Invalid board ID.
SDKINITFAILED:	SDK not initialized.

Description
This function stops video capturing on a specified capture board. The capture state would be set as “STOPPED” after a successful stop.

See Also
DVP1412_Start
DVP1412_GetCapState

DVP1412_GetCapState

Syntax
int DVP1412_GetCapState(int BoardID)

Parameters
BoardID: Specifies the board ID number(0~7).

Return Value
BOARDIDERROR: Invalid board ID.
SDKINITFAILED: SDK not initialized.

Description
This function gets capture state of a specified capture board.

STOPPED	= 0,
PREVIEW	= 1,
RECORDING	= 2,
PREVIEWandRECORDING	= 3,
UNINITIALIZED	= -1,
UNKNOWNSTATE	= -2

See Also
DVP1412_InitSDK
DVP1412_Start
DVP1412_Stop

DVP1412_SetRecord

Syntax
int DVP1412_SetRecord(int BoardID, char Filename[256])

Parameters
BoardID: Specifies the board ID number(0~7).
Filename: Specifies recording file name.

Return Value

SUCCEEDED:	Function succeeded.
FAILED:	Function failed.
BOARDIDERROR:	Invalid board ID.
SDKINITFAILED:	SDK not initialized.

Description
This function sets recording file name of a specified capture board. File name must be assigned before start recording.

See Also
DVP1412_Start

DVP1412_IsVideoPresent

Syntax
Int DVP1412_IsVideoPresent (int BoardID, BOOL * VideoPresent)

Parameters

BoardID:	Specifies the board ID number(0~7).
VideoPresent:	A pointer to get if video signal is present.

Return Value

SUCCEEDED:	Function succeeded.
FAILED:	Function failed.
BOARDIDERROR:	Invalid board ID.
SDKINITFAILED:	SDK not initialized.

Description
This function check if video signal is present in the input jet of specified board.

DVP1412_GetStatistics

GetStatistics(int BoardID, StatisticInfo *statistic)

Syntax

int DVP1412_GetStatistics (int BoardID, StatisticInfo* statistic)

Parameters

BoardID: Specifies the board ID number(0~7).
statistic: A pointer to current video statistic.

Return Value

SUCCEEDED:	Function succeeded.
FAILED:	Function failed.
BOARDIDERROR:	Invalid board ID.
PARAMERROR:	Invalid parameter.
SDKINITFAILED:	SDK not initialized.

Description

This function gets video statistic information(current frame count, current bit count) of a specified capture board.

See Also

StatisticInfo

DVP1412_GetCompressMode

Syntax

int DVP1412_GetVideoFormat(int BoardID, CompressMode *compressmode, MPEG4Mode * mpeg4mode)

Parameters

BoardID: Specifies the board ID number(0~7).
Compressmode: This version is for Mpeg4 only.

```
typedef enum tagCompressMode
{
        Mpeg4           = 0,
        Mpeg2           = 1,
        Mpeg1           = 2,
} CompressMode;
```

mpeg4mode: A pointer to get mpeg4 encoding mode.

```
typedef enum tagMPEG4Mode
{
        None            = 0,
        Divx            = 1,
         Microsoft      = 2,
} MPEG4Mode;
```

Return Value

SUCCEEDED:	Function succeeded.
BOARDIDERROR:	Invalid board ID.
PARAMERROR:	Invalid parameter.
SDKINITFAILED:	SDK not initialized.

Description

This function gets mpeg4 encoding mode of a specified capture board.

See Also

DVP1412_SetCompressMode

DVP1412_SetCompressMode

Syntax

int DVP1412_SetVideoFormat(int BoardID, CompressMode *compressmode, MPEG4Mode mpeg4mode)

Parameters

BoardID: Specifies the board ID number(0~7).
Compressmode: This version is for Mpeg4 only.

```
typedef enum tagCompressMode
{
        Mpeg4           = 0,
        Mpeg2           = 1,
        Mpeg1           = 2,
} CompressMode;
```

mpeg4mode: Mpeg4 encoding mode.

```
typedef enum tagMPEG4Mode
{
        None            = 0,
        Divx            = 1,
         Microsoft      = 2,
} MPEG4Mode;
```

Return Value

SUCCEEDED:	Function succeeded.
FAILED:	Function failed.
BOARDIDERROR:	Invalid board ID.
PARAMERROR:	Invalid parameter.
SDKINITFAILED:	SDK not initialized.

Description

This function sets mpeg4 encoding mode of a specified capture board. This function should be called before “DVP1412_Start”.

See Also

DVP1412_GetCompressMode

DVP1412_GetVideoStandard

Syntax
int DVP1412_GetVideoStandard(int BoardID, VideoStandard* standard)

Parameters
BoardID: Specifies the board ID number(0~7).
standard: A pointer to get video standard.

```
typedef enum tagVideoStandard
{
        NTSC_M          = 0,
        NTSC_M_J        = 1,
        PAL_B           = 2,
        PAL_M           = 3,
        PAL_N           = 4,
} VideoStandard;
```

Return Value

SUCCEEDED:	Function succeeded.
BOARDIDERROR:	Invalid board ID.
PARAMERROR:	Invalid parameter.
SDKINITFAILED:	SDK not initialized.

Description
This function gets video standard of a specified capture board.

See Also
DVP1412_SetVideoStandard

DVP1412_SetVideoStandard

Syntax
int DVP1412_SetVideoStandard(int BoardID, VideoStandard standard)

Parameters
BoardID: Specifies the board ID number(0~7).
standard: Video standard.

```
typedef enum tagVideoStandard
{
        NTSC_M          = 0,
        NTSC_M_J        = 1,
        PAL_B           = 2,
        PAL_M           = 3,
        PAL_N           = 4,
} VideoStandard;
```

Return Value

SUCCEEDED:	Function succeeded.
BOARDIDERROR:	Invalid board ID.
PARAMERROR:	Invalid parameter.
SDKINITFAILED:	SDK not initialized.

Description
This function sets video standard of a specified capture board.

See Also
DVP1412_GetVideoStandard

DVP1412_GetResolution

Syntax
int DVP1412_GetResolution(int BoardID, Resolution *Size)

Parameters
BoardID: Specifies the board ID number(0~7).
Size: A pointer to get video resolution.
typedef enum tagResolution
{
NTSC720480 = 0,
NTSC640480 = 1,
NTSC320240 = 2,
NTSC176144 = 3,
PAL720576 = 4,
PAL640480 = 5,
PAL352288 = 6,
PAL176144 = 7,
} Resolution;

Return Value

SUCCEEDED:	Function succeeded.
BOARDIDERROR:	Invalid board ID.
PARAMERROR:	Invalid parameter.
SDKINITFAILED:	SDK not initialized.

Description
This function gets video resolution of a specified capture board.

See Also
DVP1412_SetResolution

DVP1412_SetResolution

Syntax
int DVP1412_SetResolution(int BoardID, Resolution Size)

Parameters
BoardID: Specifies the board ID number(0~7).
Size: Video resolution.
typedef enum tagResolution
{
NTSC720480 = 0,
NTSC640480 = 1,
NTSC320240 = 2,
NTSC176144 = 3,
PAL720576 = 4,
PAL640480 = 5,
PAL352288 = 6,
PAL176144 = 7,
} Resolution;

Return Value

SUCCEEDED:	Function succeeded.
BOARDIDERROR:	Invalid board ID.
PARAMERROR:	Invalid parameter.
SDKINITFAILED:	SDK not initialized.

Description
This function sets video resolution of a specified capture board. This function should be called before “DVP1412_Start”.

See Also
DVP1412_GetResolution

DVP1412_GetFrameRate

Syntax
int DVP1412_GetFrameRate(int BoardID , FrameRate *Framerate)

Parameters
BoardID: Specifies the board ID number(0~7).
FrameRate: A pointer to get video frame rate.

```
typedef enum tagFrameRate
{
    NTSC2997fps     = 0,
    NTSC15fps       = 1,
    NTSC10fps       = 2,
    NTSC5fps        = 3,
    PAL25fps        = 4,
    PAL12fps        = 5,
    PAL8fps         = 6,
    PAL5fps         = 7,
} FrameRate;
```

Return Value

SUCCEEDED:	Function succeeded.
BOARDIDERROR:	Invalid board ID.
PARAMERROR:	Invalid parameter.
SDKINITFAILED:	SDK not initialized.

Description
This function gets frame rate of a specified capture board.

See Also
DVP1412_SetFrameRate

DVP1412_SetFrameRate

Syntax
int DVP1412_SetFrameRate(int BoardID, double FrameRate)

Parameters
BoardID: Specifies the board ID number(0~7).
FrameRate: Frame rate.
typedef enum tagFrameRate
{
NTSC2997fps = 0,
NTSC15fps = 1,
NTSC10fps = 2,
NTSC5fps = 3,
PAL25fps = 4,
PAL12fps = 5,
PAL8fps = 6,
PAL5fps = 7,
} FrameRate;

Return Value

SUCCEEDED:	Function succeeded.
FAILED:	Function failed.
BOARDIDERROR:	Invalid board ID.
PARAMERROR:	Invalid parameter.
SDKINITFAILED:	SDK not initialized.

Description
This function sets frame rate of a specified capture board.
This function should be called before “DVP1412_Start”.

See Also
DVP1412_GetFrameRate

DVP1412_GetBitRate

Syntax
int DVP1412_GetBitRate(int BoardID, int *Bitrate)

Parameters
BoardID: Specifies the board ID number(0~7).
Bitrate: A pointer to get video bit rate.

Return Value

SUCCEEDED:	Function succeeded.
BOARDIDERROR:	Invalid board ID.
PARAMERROR:	Invalid parameter.
SDKINITFAILED:	SDK not initialized.

Description
This function gets bit rate of a specified capture board.

See Also
DVP1412_SetBitRate

DVP1412_SetBitRate

Syntax
int DVP1412_SetBitRate(int BoardID, int Bitrate)

Parameters
BoardID: Specifies the board ID number(0~7).
Bitrate: Bit rate(0~40000000).

Return Value

SUCCEEDED:	Function succeeded.
FAILED:	Function failed.
BOARDIDERROR:	Invalid board ID.
PARAMERROR:	Invalid parameter.
SDKINITFAILED:	SDK not initialized.

Description
This function sets bit rate of a specified capture board.
This function should be called before “DVP1412_Start”.

See Also
DVP1412_GetBitRate

DVP1412_GetSequenceMode

Syntax

int DVP1412_GetSequenceMode(int BoardID, SequenceMode *Sequence)
Parameters
BoardID: Specifies the board ID number(0~7).
Sequence: A pointer to get video sequence mode .

```
typedef enum tagSequenceMode
{
        IPB_frame       = 0,
        IP_frame        = 1,
        I_frame         = 2,
} SequenceMode;
```

Return Value

SUCCEEDED:	Function succeeded.
BOARDIDERROR:	Invalid board ID.
PARAMERROR:	Invalid parameter.
SDKINITFAILED:	SDK not initialized.

Description

This function gets sequence mode of a specified capture board.

See Also

DVP1412_SetSequenceMode

DVP1412_SetSequenceMode

Syntax
int DVP1412_SetSequenceMode(int BoardID, SequenceMode Sequence)

Parameters
BoardID: Specifies the board ID number(0~7).
Sequence: Sequence mode .

```
typedef enum tagSequenceMode
{
        IPB_frame       = 0,
        IP_frame        = 1,
         I_frame        = 2,
} SequenceMode;
```

Return Value

SUCCEEDED:	Function succeeded.
FAILED:	Function failed.
BOARDIDERROR:	Invalid board ID.
PARAMERROR:	Invalid parameter.
SDKINITFAILED:	SDK not initialized.

Description
This function sets sequence mode of a specified capture board.
This function should be called before “DVP1412_Start”.

See Also
DVP1412_GetSequenceMode

DVP1412_GetMpeg4GOPSize

Syntax
int DVP1412_GetMpeg4GOPSizeMode(int BoardID, int *GOPSize)

Parameters
BoardID: Specifies the board ID number(0~7).
GOPSize: A pointer to get mpeg4 GOP size.

Return Value

SUCCEEDED:	Function succeeded.
BOARDIDERROR:	Invalid board ID.
PARAMERROR:	Invalid parameter.
SDKINITFAILED:	SDK not initialized.

Description
This function gets mpeg4 GOP size of a specified capture board.

See Also
DVP1412_SetMpeg4GOPSizeMode

DVP1412_ SetMpeg4GOPSize

Syntax
int DVP1412_SetMpeg4GOPSizeMode(int BoardID, int GOPSize)

Parameters
BoardID: Specifies the board ID number(0~7).
GOPSize: Mpeg4 GOP size(<300).

Return Value

SUCCEEDED:	Function succeeded.
FAILED:	Function failed.
BOARDIDERROR:	Invalid board ID.
PARAMERROR:	Invalid parameter.
SDKINITFAILED:	SDK not initialized.

Description
This function sets mpeg4 GOP size of a specified capture board.
This function should be called before “DVP1412_Start”.

See Also
DVP1412_GetMpeg4GOPSizeMode

DVP1412_GetBrightness

Syntax
int DVP1412_GetBrightness(int BoardID, int *pnValue)

Parameters
BoardID: Specifies the board ID number(0~7).
pnValue: A long pointer to get brightness value.

Return Value

SUCCEEDED:	Function succeeded.
FAILED:	Function failed.
BOARDIDERROR:	Invalid board ID.
PARAMERROR:	Invalid parameter.
SDKINITFAILED:	SDK not initialized.

Description
This function gets brightness value of a specified capture board.

See Also
DVP1412_SetBrightness

DVP1412_SetBrightness

Syntax
int DVP1412_SetBrightness(int BoardID, int nValue)

Parameters
BoardID: Specifies the board ID number(0~7).
nValue: A value to set brightness(0~100).

Return Value

SUCCEEDED:	Function succeeded.
FAILED:	Function failed.
BOARDIDERROR:	Invalid board ID.
PARAMERROR:	Invalid parameter.
SDKINITFAILED:	SDK not initialized.

Description
This function sets brightness value of a specified capture board.

See Also
DVP1412_GetBrightness

DVP1412_GetContrast

Syntax
int DVP1412_GetContrast(int BoardID, int *pnValue)

Parameters
BoardID: Specifies the board ID number(0~7).
pnValue: A long pointer to get contrast value.

Return Value

SUCCEEDED:	Function succeeded.
FAILED:	Function failed.
BOARDIDERROR:	Invalid board ID.
PARAMERROR:	Invalid parameter.
SDKINITFAILED:	SDK not initialized.

Description
This function gets contrast value of a specified capture board.

See Also
DVP1412_SetContrast

DVP1412_SetContrast

Syntax
int DVP1412_SetContrast(int BoardID, int nValue)

Parameters
BoardID: Specifies the board ID number(0~7).
nValue: A value to set contrast(0~100).

Return Value

SUCCEEDED:	Function succeeded.
FAILED:	Function failed.
BOARDIDERROR:	Invalid board ID.
PARAMERROR:	Invalid parameter.
SDKINITFAILED:	SDK not initialized.

Description
This function sets contrast value of a specified capture board.

See Also
DVP1412_GetContrast

DVP1412_GetHue

Syntax
int DVP1412_GetHue(int BoardID, int *pnValue)

Parameters
BoardID: Specifies the board ID number(0~7).
pnValue: A long pointer to get hue value.

Return Value

SUCCEEDED:	Function succeeded.
FAILED:	Function failed.
BOARDIDERROR:	Invalid board ID.
PARAMERROR:	Invalid parameter.
SDKINITFAILED:	SDK not initialized.

Description
This function gets hue value of a specified capture board.

See Also
DVP1412_SetHue

DVP1412_SetHue

Syntax
int DVP1412_SetHue(int BoardID, int nValue)

Parameters
BoardID: Specifies the board ID number(0~7).
nValue: A value to set hue(0~100).

Return Value

SUCCEEDED:	Function succeeded.
FAILED:	Function failed.
BOARDIDERROR:	Invalid board ID.
PARAMERROR:	Invalid parameter.
SDKINITFAILED:	SDK not initialized.

Description
This function sets hue value of a specified capture board.

See Also
DVP1412_GetHue

DVP1412_GetSaturation

Syntax
int DVP1412_GetSaturation(int BoardID, int *pnValue)

Parameters
BoardID: Specifies the board ID number(0~7).
pnValue: A long pointer to get saturation value.

Return Value

SUCCEEDED:	Function succeeded.
FAILED:	Function failed.
BOARDIDERROR:	Invalid board ID.
PARAMERROR:	Invalid parameter.
SDKINITFAILED:	SDK not initialized.

Description
This function gets saturation value of a specified capture board.

See Also
DVP1412_SetSaturation

DVP1412_SetSaturation

Syntax
int DVP1412_SetSaturation(int BoardID, int nValue)

Parameters
BoardID: Specifies the board ID number(0~7).
nValue: A value to set saturation(0~100).

Return Value

SUCCEEDED:	Function succeeded.
FAILED:	Function failed.
BOARDIDERROR:	Invalid board ID.
PARAMERROR:	Invalid parameter.
SDKINITFAILED:	SDK not initialized.

Description
This function sets saturation value of a specified capture board.

See Also
DVP1412_GetSaturation

DVP1412_ReadEE

Syntax
int DVP1412_ReadEE(int BoardID, int addr, BYTE *pByte)

Parameters
BoardID:Specifies the board ID number(0~7).
addr: Address to read a value from EEPRom.
pByte: A BYTE pointer to the byte value stored in EEPRom of a specified capture board.

Return Value

SUCCEEDED:	Function succeeded.
FAILED:	Function failed.
BOARDIDERROR:	Invalid board ID.
PARAMERROR:	Invalid parameter.
SDKINITFAILED:	SDK not initialized.

Description
This function reads the value at specified address from EEPRom on a selected capture board.

See Also
DVP1412_WriteEE

DVP1412_WriteEE

Syntax
int DVP1412_WriteEE(int BoardID, int addr, BYTE pByte)

Parameters
BoardID:Specifies the board ID number(0~7).
addr: Address to write a value to EEPRom.
pByte: A BYTE value being written to EEPRom of a specified video channel.

Return Value

SUCCEEDED:	Function succeeded.
FAILED:	Function failed.
BOARDIDERROR:	Invalid board ID.
PARAMERROR:	Invalid parameter.
SDKINITFAILED:	SDK not initialized.

Description
This function reads the value at specified address from EEPRom on a selected capture board.

See Also
DVP1412_ReadEE