DVP-1020

4 Channel PC/104-Plus Video Capture Module

Copyright

Acknowledgments

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. Delphi and C++ Builder are trademarks of Inprise Corporation.

CE notification

The DVP-1020, developed by ADVANTECH CO., LTD., has passed the CE test for environmental specifications when shielded cables are used for external wiring. We recommend the use of shielded cables. This kind of cable is available from Advantech. Please contact your local supplier for ordering information

On-line Technical Support

For technical support and service, please visit our support website at:
http://www.advantech.com/support

Part No. 2062102000 1st Edition
Printed in Taiwan June. 2004
Rev. 0.1c

Contents

CHAPTER 1

General Information

Chapter 1 General Information

DVP-1020 is a 4-channel video input PC/104-Plus video capture module that supports up to D1 resolution at 30/25 fps frame rate. DVP-1020 allows installation of up to 4 modules in one PC system by utilizing an onboard DIP switch setting that identifies card ID by LED indicators. DVP-1020 supports NTSC/PAL signals and digitizes data to the PC through the PC/104-Plus interface.

DVP-1020 has an auto wake-up function that provides 10 sets of time-settings. From an "off" state, two digital trigger pins can also remotely trigger this function. DVP-1020 also has a built-in watchdog timer to reset the system and prevent crashes when an unknown error occurs. DVP-1020 is also designed with a programmable software protection key for anonymous user software copy protection.

1.1 Hardware Requirements

- Intel Pentium III 1GHz or above (The CPU speed depends on the video frame rate, channels and resolution)
- 256 MB RAM or above
- Free PC/104-Plus slot

1.2 Software Requirement

- Microsoft Windows 2000/XP with DirectX 8.1 or above

1.3 Block Diagram

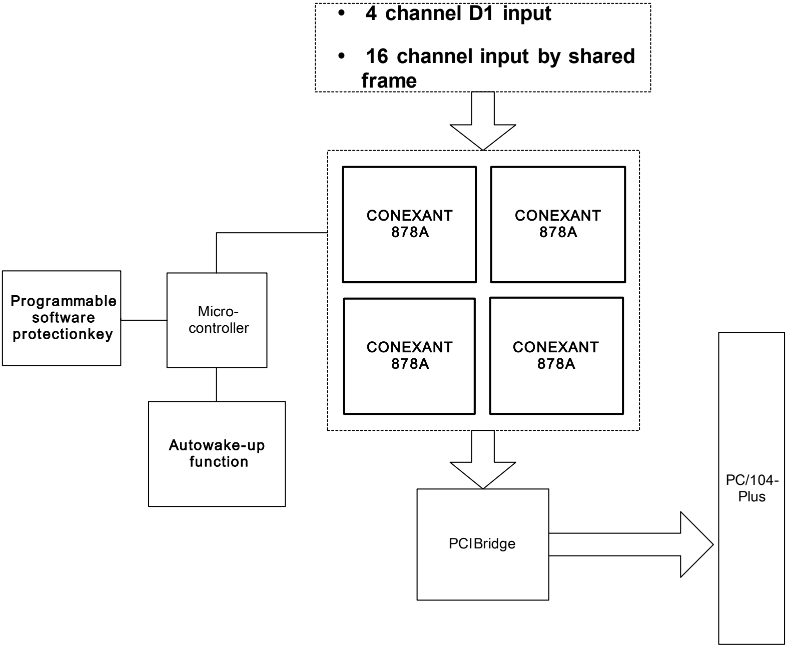


Figure 1.1 System diagram

1.4 Packing List

◆	DVP-1020 PC/104-Plus video capture module	X 1
◆	Utility CD	X 1
◆	Standby power wire (p/n: 1700000575)	X 1
◆	Connection wire for WDT (p/n: 1703020303)	X 1
◆	Connection wire for power switch (p/n: 1703020303)	X 1
◆	Wire for video input (p/n: 1700000860)	X 4
◆	BNC board (p/n: 9696NC2000)	X 4

1.5 Dimensions

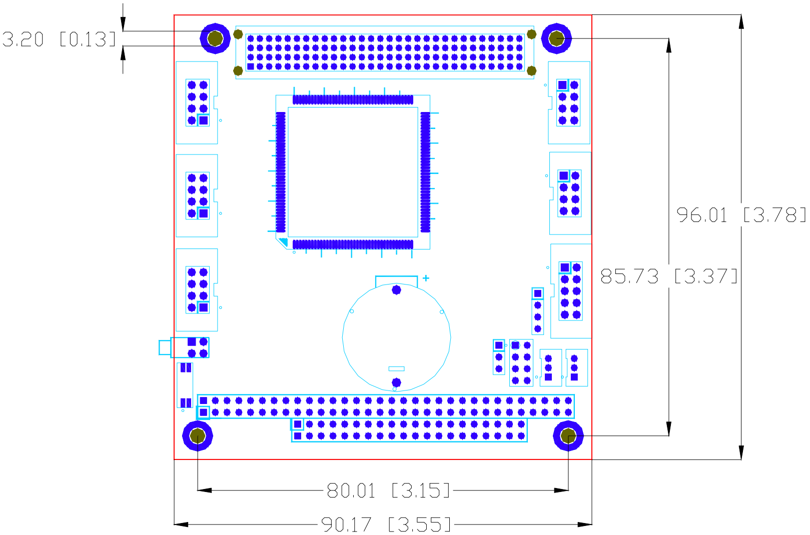


Figure 1.2 Dimensions

1.6 Jumpers & connectors location

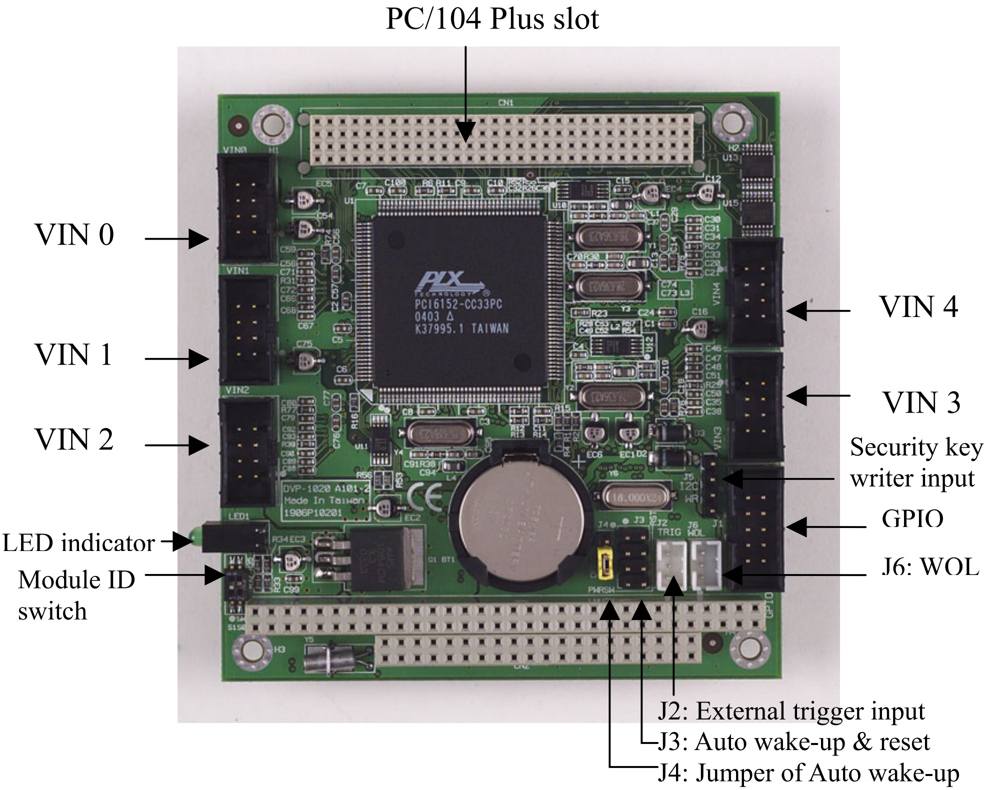


Figure 1.3 Jumpers & connectors location

1.7 Jumpers & connectors definition

1.7.0 Module ID switch : SW1

DVP-1020 can provide up to 3 cards on one board. **SW1** is the Module ID switch and for Card ID selection. Near SW1, the LED indicator is for the following

	SW1; LED indicator (OFF: 0, ON: 1)	
	1	2
Card 0	0	0
Card 1	0	1
Card 2	1	0
Not allowed	1	1

Table 1.1 Card ID

λ Warning: Please do not put Module ID Switch at (1,1) status. This would cause a H/Wconflict.

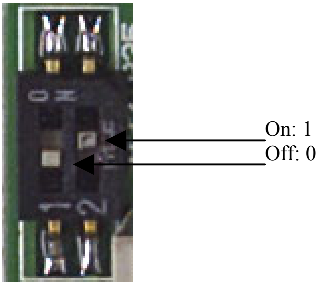


Figure 1.4 Module ID switch

1.7.1 GPIO:

- 10-bit TTL/CMOS level Digital I/O.

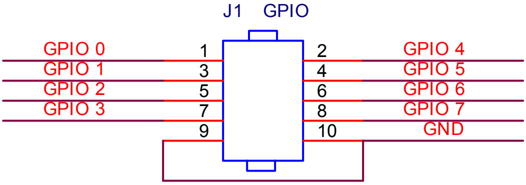


Figure 1.5 GPIO pin definition

1.7.2 External trigger input: J2

Two outer triggers through J2 can wake DVP-1020. These external triggers are set by J3 and the pin definitions are shown in Table 1.2 and Figure 1.6. The outer triggers can be sent through pin1 or pin2 and the signals must be low-active. (Figure 1.7)

External Trigger (J2) Pin definition	
Pin no.	**Description**
Pin 1	Trigger Pin 0
Pin 2	Trigger Pin 1
Pin 3	GND

Table 1.2 Pin definition of External trigger input (J2)

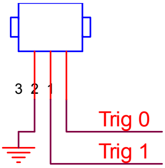


Figure 1.6 External trigger (J2) Pin definition

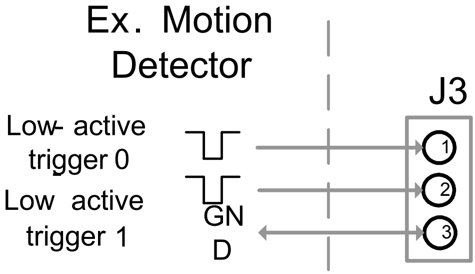


Figure 1.7 descriptions for active triggers of J2

1.7.3 Auto wake-up & reset function: J3

Auto wake-up and WDT reset function are set by J3 with 8 pins. The pins of 5,6,7,8 are for auto wake-up function (DVP-1020X) and 1,2,3,4 are for WDT function. Pin definitions are shown in Table 1.3, Figure 1.8 and 1.9. Auto wake-up function needs ATX motherboard and power.

Auto wake-up & reset (J3) Pin definition

Pin no.	Description
Pin 1	GND
Pin 2	GND
Pin 3	Reset Pin on Motherboard
Pin 4	Reset Pin on Panel
Pin 5	GND
Pin 6	GND
Pin 7	Power SW on Motherboard
Pin 8	Power SW on Panel

Table 1.3 Auto wake-up & reset (J3) Pin definition

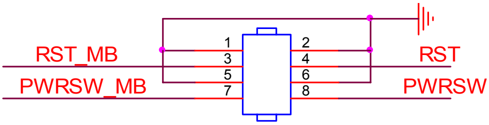


Figure 1.8 Auto wake-up & reset (J3) Pin definition (1)

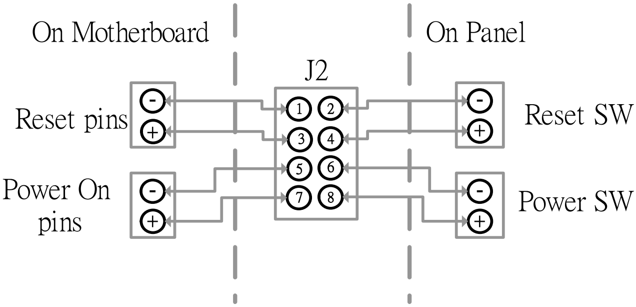


Figure 1.9 Auto wake-up & reset (J3) Pin definition (2)

λ The auto wake-up function needs to set up Jumper (J4) which is as the following table:

Jumper (J4) of Auto wake-up	
Pin status.	**Description**
Pin1-Pin2	Enable Auto wake-up
Pin2-Pin3	Disable Auto wake-up

Table 1.4 Jumper (J4) of Auto wake-up

1.7.4 Security key writer input: J5

To prevent any anonymous user from making illegal software copies, DVP-1020 has special programmable software protection key called Security Key Writer “DVA-1000”. Please see the ordering information in thedatasheet. J5 is the signal input for DVA-1000.

1.7.5 Wake on LAN (WOL): J6

J6 is the connector for power input of wake on LAN and it should be connected to the 5V standby power when users want to enable the auto wake-up function.

Pin definition of WOL (J6)

Pin no.	**Description**
Pin 1	5V/100 mA standby voltage
Pin 2	GND
Pin 3	Not connected

Table 1.5 Pin definition of WOL (J6)

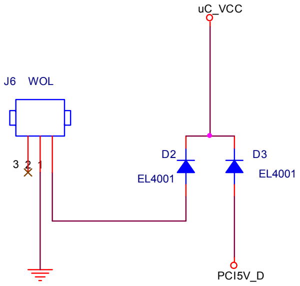


Figure 1.10 Pin definition of WOL (J6)

1.7.6 Video input connector: VIN0~VIN4

Standard D1 inputs: VIN0

VIN0 is the standard video input of DVP-1020. D1 resolution and real time frame rate can be available through this connector.

16 channels inputs by shared frame: VIN1~VIN4

With share frame technology, DVP-1020 can receive up to16 channel composite inputs through VIN1,VIN2, VIN3 and VIN4. It will however, decrease performance for each single channel but users can have more amounts of video input.

λ The resolution and captured frame rate is related to the performance of the system hardware and limited by PCI bandwidth. Please refer to: Chapter 1.1 Hardware requirements.

1.8 Battery

DVP-1020 has a lithium battery for the Real Time Clock (RTC) function. This battery is widely used and its specifications are as follows:

Battery Spec.	
Item	**Description**
Model name	CR2032
Normal Voltage	3.0V
Standard capacity	220 mAh (on continuous discharge at 20^0C under 15KΩ load to 2.5V end-voltage)
Standard weight	3.1g
Terminals	Positive electrode material: Nickel-plated stainless steel
	Negative electrode material: Nickel-plated stainless steel

Table 1.6 Battery Spec.

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 any static electricity that might be on your body.
4. Set the SW1 (Module ID).
5. Place the DVP-1020 into the Motherboard's PC/104-Plus slot and connect corresponding accessories to the DVP-1020.
   - Wires for video input.
   - BNC boards.
   - Video cables to camera. if necessary.
6. Replace the cover of your computer chassis.
7. Plug in the power cord and turn on the computer.

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

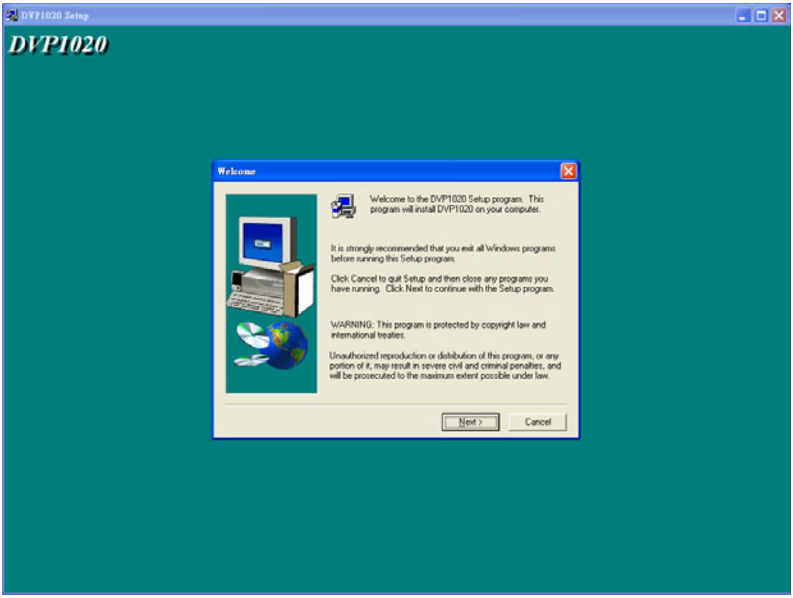
1.10 Software Installation

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 device drivers for the DVP-1020 are located on the utility CD. Review the relevant operating system commands and the pertinent sections of your application software user’s manual before performing the installation.

Installation

1. Insert the driver CD into your system's CD-ROM drive. In a few seconds, the software installation main menu appears. Move the mouse cursor over the "Manual" button under the "SETUP" heading, a message pops up telling you to start the installation.



Welcome to the DVP1020 Setup program. This program will install DVP1020 on your computer.

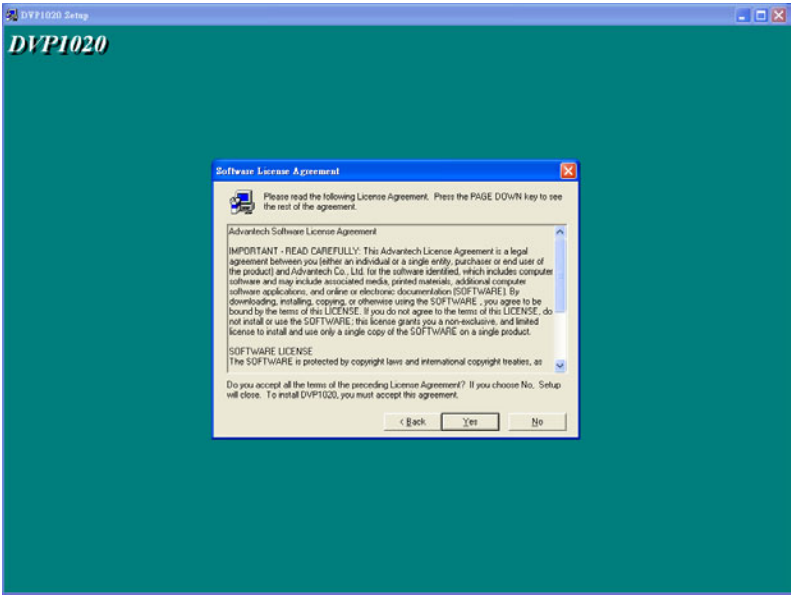
It is strongly recommended that you exit all Windows programs before running this Setup program.

Click Cancel to quit Setup and then close any programs you have running. Click Next to continue with the Setup program.

WARNING: This program is protected by copyright law and international treaties.

Unauthorized reproduction or distribution of this program, or any portion of it, may result in severe civil and criminal penalties, and will be prosecuted to the maximum extent possible under law.

2. Please read the following License Agreement. Press the PAGE DOWN key to see the rest of the agreement and Click "Yes" to continue the installation.



Please read the following License Agreement. Press the PAGE DOWN key to see the rest of the agreement.

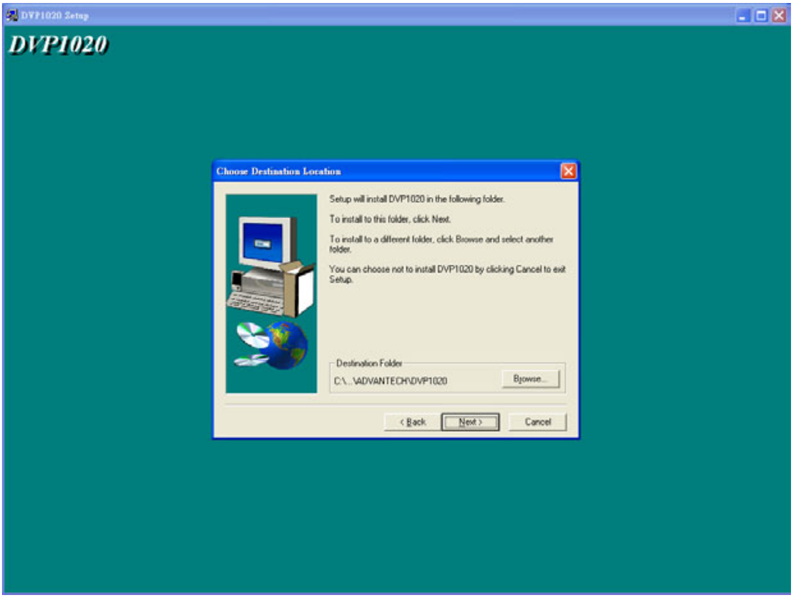
Advantech Software License Agreement

IMPORTANT - READ CAREFULLY: This Advantech License Agreement is a legal agreement between you (either an individual or a single entity, purchaser or end user of the product) and Advantech Co., Ltd. for the software identified, which includes computer software and may include associated media, printed materials, additional computer software applications, and online or electronic documentation (SOFTWARE). By downloading, installing, copying, or otherwise using the SOFTWARE , you agree to be bound by the terms of this LICENSE. If you do not agree to the terms of this LICENSE, do not install or use the SOFTWARE; this license grants you a non-exclusive, and limited license to install and use only a single copy of the SOFTWARE on a single product.

SOFTWARE LICENSE
The SOFTWARE is protected by copyright laws and international copyright treaties, as

Do you accept all the terms of the preceding License Agreement? If you choose No, Setup will close. To install DVP1020, you must accept this agreement.

3. Choose destination location on your system disc then click "Next" when you see the following message.



Setup will install DVP1020 in the following folder.

To install to this folder, click Next.

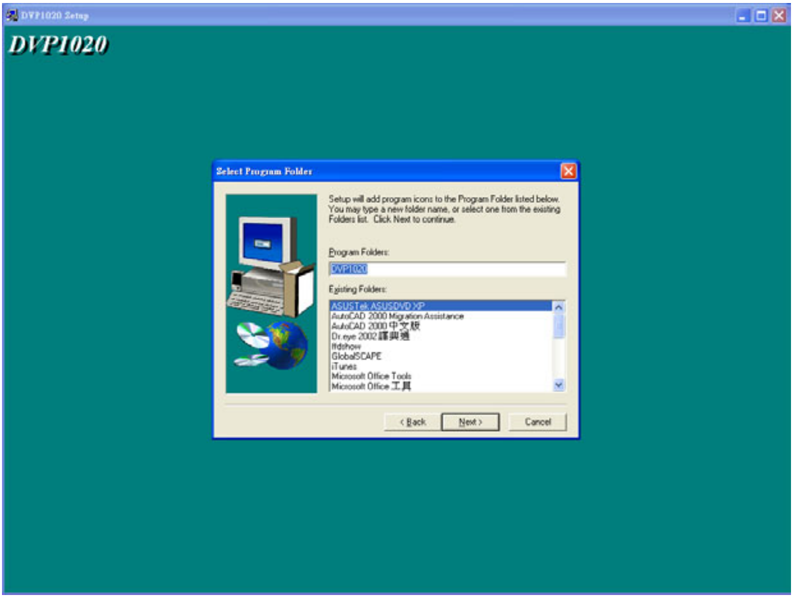
To install to a different folder, click Browse and select another folder.

You can choose not to install DVP1020 by clicking Cancel to exit Setup.

Destination Folder

C:\...\ADVANTECH\DVP1020

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

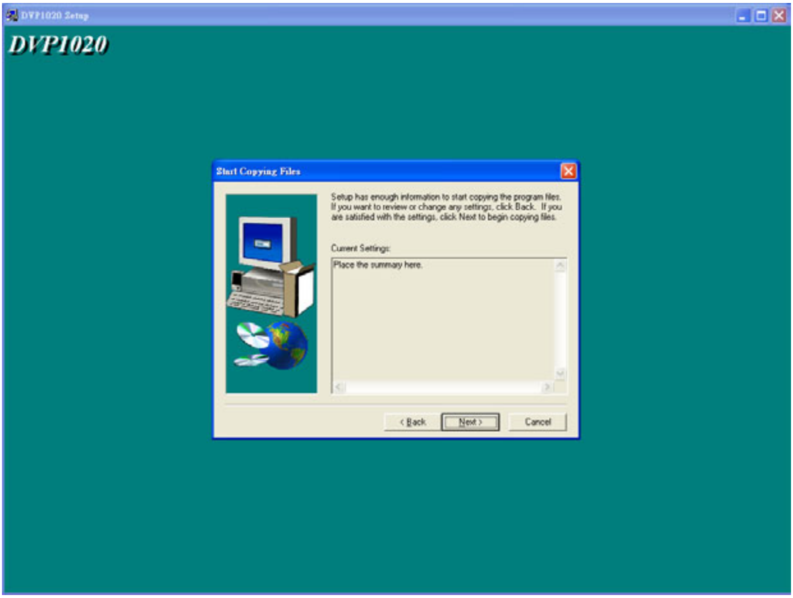


Setup will add program icons to the Program Folder listed below. You may type a new folder name, or select one from the existing Folders list. Click Next to continue.

Program Folders:
DVP1020

Existing Folders:
ASUSTek ASUSDVD XP
AutoCAD 2000 Migration Assistance
AutoCAD 2000 中文版
Dr.eye 2002 譯典通
ffdshow
GlobalSCAPE
iTunes
Microsoft Office Tools
Microsoft Office 工具

5. Please fill in the information and Click "Next".

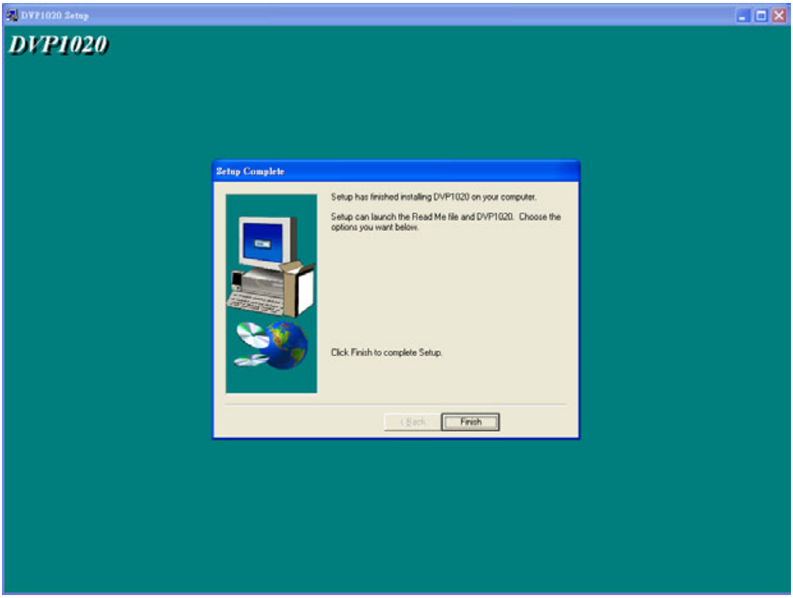


Setup has enough information to start copying the program files. If you want to review or change any settings, click Back. If you are satisfied with the settings, click Next to begin copying files.

Current Settings:

Place the summary here.

6. When the following message appears, click "Finish" to complete the installation and restart Windows or your computer.



Setup has finished installing DVP1020 on your computer.

Setup can launch the Read Me file and DVP1020. Choose the options you want below.

Click Finish to complete Setup.

7. After installation, all the drivers are automatically located at the following place:

   C:\Program Files\ADVANTECH\DVP1020\Driver

   Please update the drivers to the above location.

CHAPTER 2

General Information

Chapter 2   Software Information

2.1 DVP-1020 Functions Library Summary

Summary

SDK Initialize and close

Adv_DVPAPI_CreateSDKInstence
Adv_DVPAPI_GetNoOfDevices
Adv_DVPAPI_InitSDK
Adv_DVPAPI_CloseSDK

Capture control

Adv_DVPAPI_Start
Adv_DVPAPI_Stop
Adv_DVPAPI_GetCapState
Adv_DVPAPI_SetNewFrameCallback
Adv_DVPAPI_GetCurFrameBuffer

Capture setting

Adv_DVPAPI_GetVideoFormat
Adv_DVPAPI_SetVideoFormat
Adv_DVPAPI_GetFrameRate
Adv_DVPAPI_SetFrameRate
Adv_DVPAPI_GetResolution
Adv_DVPAPI_SetResolution
Adv_DVPAPI_GetVideoInput
Adv_DVPAPI_SetVideoInput

Sensor Control

Adv_DVPAPI_GetBrightness
Adv_DVPAPI_SetBrightness
Adv_DVPAPI_GetContrast
Adv_DVPAPI_SetContrast
Adv_DVPAPI_GetHue
Adv_DVPAPI_SetHue
Adv_DVPAPI_GetSaturation
Adv_DVPAPI_SetSaturation

GPIO

Adv_DVPAPI_GPIOGetData
Adv_DVPAPI_GPIOSetData

Micro Controller

Adv_DVPAPI_GetWDTTimeout
Adv_DVPAPI_SetWDTTimeout
Adv_DVPAPI_GetUCFlag
Adv_DVPAPI_SetUCFlag
Adv_DVPAPI_GetPoweronEvent
Adv_DVPAPI_GetAlarm
Adv_DVPAPI_SetAlarm
Adv_DVPAPI_GetChecksum
Adv_DVPAPI_GetEEData
Adv_DVPAPI_SetEEData
Adv_DVPAPI_GetRTCData
Adv_DVPAPI_SetRTCData

Functions Reference

Struct

IDStruct

```
typedef struct{
int Port[4];
int BoardID;
} IDStruct;
```

Parameters

Port[4]:		Four Port IDs on current board.
BoardID:	Board ID.

Description

A stucrt stores DVP1020 ID information.

TimeStruct

```
typedef struct{
BYTE second;
BYTE minute;
BYTE hour;
BYTE day;
BYTE date;
BYTE month;
BYTE year;
} TimeStruct;
```

Description

A stucrt stores time setting.

AlarmStruct

```
typedef struct{
BOOL enable;
BYTE type;
TimeStruct AlarmT;
} AlarmStruct;
```

Parameters

enable: Enable or disable alarm setting.

type: Type of alarm:
HOURLY_ALARM
DAILY_ALARM
WEEKLY_ALARM
MONTHLY_ALARM
YEARLY_ALARM
ONCE_ALARM

AlarmT: Time setting for this alarm.

Description

A stucrt stores alarm time setting.

UCFlag

```
typedef struct{
bool EnableWDT;
bool EnableAlarm;
bool EnableTrigger0;
bool EnableTrigger1;
} UCFlag;
```

Parameters

EnableWDT: Enable or disable watch dog timer.
EnableAlarm: Enable or disable alarm.
EnableTrigger0: Enable or disable trigger0 on board to boot the system.
EnableTrigger1: Enable or disable trigger1 on board to boot the system.

Description

A stucrt stores system boot setting.

2.2 Method

Adv_DVPAPI_CreateSDKInstence

Syntax

int Adv_DVPAPI_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.

Adv_DVPAPI_GetNumberOfDevices

Syntax

int Adv_DVPAPI_GetNoOfDevices(void)

Parameters

None

Return Value

Number of Capture Devices in a DVP1020 integrated system.

Description

This function gets number of DVP1020 Capture Devices in the system. At most 16 channels (four DVP1020 boards) are available in a DVP1020 integrated system.

Adv_DVPAPI_InitSDK

Syntax

int Adv_DVPAPI_InitSDK(int NoOfDevs, IDStruct* IDList)

Parameters

NoOfDevs:	Number of devices.
IDList:	An IDStruct array pointer stores all board IDs and Port IDs. Negative value indentifys inactive channel.

Return Value

SUCCEEDED:	Function succeeded.
FAILED:	Function failed.
BOARDIDERROR:	Failed to get board ID or duplicate board ID.
PORTIDERROR:	Failed to get port ID.
NODEVICES:	No devices found.

Description

This function initializes all DVP7010A capture devices in the system and gets all board IDs. After initializing each device, the capture status would be set as "STOPPED".

See Also

Adv_DVPAPI_GetNoOfDevices
Adv_DVPAPI_GetCapState
Adv_DVPAPI_CloseSDK
IDStruct

Adv_DVPAPI_CloseSDK

Syntax

int Adv_DVPAPI_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

Adv_DVPAPI_InitSDK

Adv_DVPAPI_Start

Syntax

int Adv_DVPAPI_Start(int BoardID, int PortID, int SwitchingChans, HWND Main, HWND hwndPreview)

Parameters

BoardID: Specifies the board ID number(0~3).
PortID: Specifies the port ID number(0~3).
SwitchingChans: Single video input or switching between video muxes.
0 single channel.
2 channels (mux0, mux1).
3 channels (mux0, mux1, mux2).
4 channels (mux0, mux1, mux2, mux3).
Main: A main window handle.
hwndPreview: A windows handle for display area.

Return Value

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

Description

This function starts video capturing on a specified capture port.
The capture state would be set as “RUNNING” after a successful start.

See Also

Adv_DVPAPI_Stop
Adv_DVPAPI_GetCapState

Adv_DVPAPI_Stop

Syntax

int Adv_DVPAPI_Stop(int BoardID, int PortID)

Parameters

BoardID:	Specifies the board ID number(0~3).
PortID:	Specifies the port ID number(0~3).

Return Value

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

Description

This function stops video capturing on a specified capture port. The capture state would be set as "STOPPED" after a successful stop.

See Also

Adv_DVPAPI_Start
Adv_DVPAPI_GetCapState

Adv_DVPAPI_GetCapState

Syntax

int Adv_DVPAPI_GetCapState(int BoardID, int PortID)

Parameters

BoardID:	Specifies the board ID number(0~3).
PortID:	Specifies the port ID number(0~3).

Return Value

BOARDIDERROR:	Invalid board ID.
PORTIDERROR:	Invalid port ID.
SDKINITFAILED:	SDK not initialized.

Description

This function gets capture state of a specified capture port.

STOPPED	= 1,
RUNNING	= 2,
UNINITIALIZED	= -1,
UNKNOWNSTATE	= -2

See Also

Adv_DVPAPI_InitSDK
Adv_DVPAPI_Start
Adv_DVPAPI_Stop

Adv_DVPAPI_GetCurFrameBuffer

Syntax

int Adv_DVPAPI_GetCurFrameBuffer(int BoardID, int PortID, long* bufSize, BYTE* buf, int VMux)

Parameters

BoardID:	Specifies the board ID number(0~3).
PortID:	Specifies the port ID number(0~3).
bufSize:	Frame buffer size.
buf:	Frame buffer.
VMux:	Video mux.

Return Value

SUCCEEDED:	Function succeeded.
FAILED:	Function failed.
BOARDIDERROR:	Invalid board ID.
PORTIDERROR:	Invalid port ID.
PARAMERROR:	Invalid parameter.
SDKINITFAILED:	SDK not initialized.
NOSAMPLE:	No buffer sample.

Description

This function gets current frame buffer of a specified capture port. Start capturing before the function is called.

See Also

Adv_DVPAPI_Start

Adv_DVPAPI_SetNewFrameCallback

Syntax

int Adv_DVPAPI_SetNewFrameCallback(int BoardID, int PortID, int callback)

Parameters

BoardID:	Specifies the board ID number(0~3).
PortID:	Specifies the port ID number(0~3).
callback:	Callback function.

Callback fumction type:

int (int lParam, int nID, int BoardID, int PortID, int VMux, int bufsize, BYTE* buf);

Return Value

SUCCEEDED:	Function succeeded.
BOARDIDERROR:	Invalid board ID.
PORTIDERROR:	Invalid port ID.
SDKINITFAILED:	SDK not initialized.

Description

This function sets a callback function to SDK. When new frame arrived, messages and frame information will be sent to callback function.

See Also

Adv_DVPAPI_GetVideoFormat

Syntax

int Adv_DVPAPI_GetVideoFormat(int BoardID, int PortID, AnalogVideoFormat* vFormat)

Parameters

BoardID:	Specifies the board ID number(0~3).
PortID:	Specifies the port ID number(0~3).
Vformat:	A pointer to get video format. Video_None, Video_NTSC_M, Video_NTSC_M_J, Video_PAL_B, Video_PAL_M, Video_PAL_N, Video_SECAM_B

Return Value

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

Description

This function gets video input format of a specified capture port.

See Also

Adv_DVPAPI_SetVideoFormat

Adv_DVPAPI_SetVideoFormat

Syntax

int Adv_DVPAPI_SetVideoFormat(int BoardID, int PortID, AnalogVideoFormat* vFormat)

Parameters

BoardID:	Specifies the board ID number(0~3).
PortID:	Specifies the port ID number(0~3).
Vformat:	video format: Video_None, Video_NTSC_M, Video_NTSC_M_J, Video_PAL_B, Video_PAL_M, Video_PAL_N, Video_SECAM_B

Return Value

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

Description

This function sets video input format a specified capture port. This function should be called before "Adv_DVPAPI_Start".

See Also

Adv_DVPAPI_GetVideoFormat

Adv_DVPAPI_GetFrameRate

Syntax

int Adv_DVPAPI_GetFrameRate(int BoardID, int PortID, double *FrameRate)

Parameters

BoardID:	Specifies the board ID number(0~3).
PortID:	Specifies the port ID number(0~3).
FrameRate:	A pointer to get video frame rate.

Return Value

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

Description

This function gets frame rate of a specified capture port.

See Also

Adv_DVPAPI_SetFrameRate

Adv_DVPAPI_SetFrameRate

Syntax

int Adv_DVPAPI_SetFrameRate(int BoardID, int PortID, double FrameRate)

Parameters

BoardID:	Specifies the board ID number(0~3).
PortID:	Specifies the port ID number(0~3).
FrameRate:	A value to set frame rate. (0.0<FrameRate<=30.0, Default value is 30.0)

Return Value

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

Description

This function sets frame rate of a specified capture port. This function should be called before "Adv_DVPAPI_Start".

See Also

Adv_DVPAPI_GetFrameRate

Adv_DVPAPI_GetVideoResolution

Syntax

int Adv_DVPAPI_GetResolution(int BoardID, int PortID, VideoSize *Size)

Parameters

BoardID:	Specifies the board ID number(0~3).
PortID:	Specifies the port ID number(0~3).
Size:	A pointer to get video resolution. SIZED1 (NTSC: 720x480, PAL: 720x576), SIZEVGA (640x480), SIZEQVGA (320x240), SIZESUBQVGA (160x120),

Return Value

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

Description

This function gets video resolution of a specified capture port.

See Also

Adv_DVPAPI_SetResolution

Adv_DVPAPI_SetVideoResolution

Syntax

int Adv_DVPAPI_SetResolution(int BoardID, int PortID, VideoSize Size)

Parameters

BoardID: Specifies the board ID number(0~3).
PortID: Specifies the port ID number(0~3).
Size: A value to set video resolution.
SIZED1 (NTSC: 720x480, PAL: 720x576),
SIZEVGA (640x480),
SIZEQVGA (320x240),
SIZESUBQVGA (160x120),

Return Value

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

Description

This function sets video resolution of a specified capture port. This function should be called before "Adv_DVPAPI_Start".

See Also

Adv_DVPAPI_GetResolution

Adv_DVPAPI_ GetVideoInput

Syntax

int Adv_DVPAPI_GetVideoInput(int BoardID, int PortID, int* input)

Parameters

BoardID:	Specifies the board ID number(0~3).
PortID:	Specifies the port ID number(0~3).
input:	A pointer to get video input mux.

Return Value

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

Description

This function gets video input mux of a specified capture port.
It returns “FAILED” when argument “SwitchingChans “ of Adv_DVPAPI_Start was not set to 0. (This function works for no video mux automatically switching.)

See Also

Adv_DVPAPI_Start
Adv_DVPAPI_SetVideoInput

Adv_DVPAPI_SetVideoVideoInput

Syntax

int Adv_DVPAPI_SetVideoInput(int BoardID, int PortID, int input)

Parameters

BoardID:	Specifies the board ID number(0~3).
PortID:	Specifies the port ID number(0~3).
input:	A value to set video input mux(0~3).

Return Value

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

Description

This function sets video input mux of a specified capture port.

It returns “FAILED” when argument “SwitchingChans” of Adv_DVPAPI_Start was not set to 0. (This function works for no video mux automatically switching.)

See Also

Adv_DVPAPI_Start
Adv_DVPAPI_GetVideoInput

Adv_DVPAPI_GetBrightness

Syntax

int Adv_DVPAPI_GetBrightness(int BoardID, int PortID, long *pnValue)

Parameters

BoardID:	Specifies the board ID number(0~3).
PortID:	Specifies the port ID number(0~3).
pnValue:	A long pointer to get brightness value.

Return Value

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

Description

This function gets brightness value of a specified capture port.

See Also

Adv_DVPAPI_SetBrightness

Adv_DVPAPI_SetBrightness

Syntax

int Adv_DVPAPI_SetBrightness(int BoardID, int PortID, long nValue)

Parameters

BoardID:	Specifies the board ID number(0~3).
PortID:	Specifies the port ID number(0~3).
nValue:	A value to set brightness(0~100).

Return Value

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

Description

This function sets brightness value of a specified capture port.

See Also

Adv_DVPAPI_GetBrightness

Adv_DVPAPI_GetContrast

Syntax

int Adv_DVPAPI_GetContrast(int BoardID, int PortID, long *pnValue)

Parameters

BoardID:	Specifies the board ID number(0~3).
PortID:	Specifies the port ID number(0~3).
pnValue:	A long pointer to get contrast value.

Return Value

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

Description

This function gets contrast value of a specified capture port.

See Also

Adv_DVPAPI_SetContrast

Adv_DVPAPI_SetContrast

Syntax

int Adv_DVPAPI_SetContrast(int BoardID, int PortID, long nValue)

Parameters

BoardID:	Specifies the board ID number(0~3).
PortID:	Specifies the port ID number(0~3).
nValue:	A value to set contrast(0~100).

Return Value

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

Description

This function sets contrast value of a specified capture port.

See Also

Adv_DVPAPI_GetContrast

Adv_DVPAPI_GetHue

Syntax

```
int Adv_DVPAPI_GetHue(int BoardID, int PortID, long
*pnValue)
```

Parameters

BoardID: Specifies the board ID number(0~3).
PortID: Specifies the port ID number(0~3).
pnValue: A long pointer to get hue value.

Return Value

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

Description

This function gets hue value of a specified capture port.

See Also

Adv_DVPAPI_SetHue

Adv_DVPAPI_SetHue

Syntax

int Adv_DVPAPI_SetHue(int BoardID, int PortID, long nValue)

Parameters

BoardID:	Specifies the board ID number(0~3).
PortID:	Specifies the port ID number(0~3).
nValue:	A value to set hue(0~100).

Return Value

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

Description

This function sets hue value of a specified capture port.

See Also

Adv_DVPAPI_GetHue

Adv_DVPAPI_GetSaturation

Syntax

int Adv_DVPAPI_GetSaturation(int BoardID, int PortID, long *pnValue)

Parameters

BoardID:	Specifies the board ID number(0~3).
PortID:	Specifies the port ID number(0~3).
pnValue:	A long pointer to get saturation value.

Return Value

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

Description

This function gets saturation value of a specified capture port.

See Also

Adv_DVPAPI_SetSaturation

Adv_DVPAPI_SetSaturation

Syntax

int Adv_DVPAPI_SetSaturation(int BoardID, int PortID, long nValue)

Parameters

BoardID:	Specifies the board ID number(0~3).
PortID:	Specifies the port ID number(0~3).
nValue:	A value to set saturation(0~100).

Return Value

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

Description

This function sets saturation value of a specified capture port.

See Also

Adv_DVPAPI_GetSaturation

Adv_DVPAPI_GPIOGetData

Syntax

int Adv_DVPAPI_GPIOGetData(int BoardID, int Pin, BOOL* value)

Parameters

BoardID:	Specifies the board ID number(0~3).
Pin:	GPIO pin.
value:	A pointer to get specified pin value.

Return Value

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

Description

This function gets value of specified GPIO pin on a selected board.

See Also

Adv_DVPAPI_GPIOSetData

Adv_DVPAPI_GPIOSetData

Syntax

int Adv_DVPAPI_GPIOSetData(int BoardID, int Pin, BOOL value)

Parameters

BoardID:	Specifies the board ID number(0~3).
Pin:	GPIO pin.
nValue:	A value to set specified pin value.

Return Value

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

Description

This function sets value of specified GPIO pin on a selected board .

See Also

Adv_DVPAPI_GPIOGetData

Adv_DVPAPI_GetWDTTimeout

Syntax

int Adv_DVPAPI_GetWDTTimeout(int BoardID, BOOL *EnableWDT, int *timeout)

Parameters

BoardID:	Specifies the board ID number(0~3).
EnableWDT:	A pointer to get watch dog timer state.
timeout:	A pointer to get watch dog timer.

Return Value

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

Description

This function gets state and value of watch dog timer on a selected board.

See Also

Adv_DVPAPI_SetWDTTimeout

Adv_DVPAPI_SetWDTTimeout

Syntax

int Adv_DVPAPI_SetWDTTimeout(int BoardID, BOOL EnableWDT, int timeout)

Parameters

BoardID:	Specifies the board ID number(0~3).
EnableWDT:	Enable or disable. Watch dog timer.
timeout:	A pointer to get watch dog timer.

Return Value

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

Description

This function sets state and value of watch dog timer on a selected board .

See Also

Adv_DVPAPI_GetWDTTimeout

Adv_DVPAPI_GetUCFlag

Syntax

int Adv_DVPAPI_GetUCFlag(int BoardID, BOOL *enableAlarm, BOOL *enableTrig0, BOOL *enableTrig1)

Parameters

BoardID:	Specifies the board ID number(0~3).
EnableWDT:	A pointer to get watch dog timer(enable or disable).
EnableAlarm:	A pointer to get alarm(enable or disable).
EnableTrigger0:	A pointer to get trigger0(enable or disable).
EnableTrigger1:	A pointer to get trigger1(enable or disable).

Return Value

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

Description

This function gets settings (enable or disable) of WDT, alarm, and triggers on a selected board.

See Also

Adv_DVPAPI_SetUCFlag

UCFlag

Adv_DVPAPI_SetUCFlag

Syntax

int Adv_DVPAPI_SetUCFlag(int BoardID, BOOL enableAlarm, BOOL enableTrig0, BOOL enableTrig1)

Parameters

BoardID: Specifies the board ID number(0~3).

EnableWDT: A value to enable or disable watch dog timer.

EnableAlarm: A value to enable or disable alarm.

EnableTrigger0: A value to enable or disable trigger0.

EnableTrigger1: A value to enable or disable trigger1.

Return Value

SUCCEEDED: Function succeeded.

FAILED: Function failed.

BOARDIDERROR: Invalid board ID.

PARAMERROR: Invalid parameter.

SDKINITFAILED: SDK not initialized.

Description

This function enables or disables WDT, alarm, and triggers on a selected board.

See Also

Adv_DVPAPI_GetUCFlag
UCFlag

Adv_DVPAPI_GetPoweronEvent

Syntax

int Adv_DVPAPI_GetPoweronEvent(int BoardID, POWERON_EVENT *powerEvent)

Parameters

BoardID: Specifies the board ID number(0~3).

powerEvent: A pointer to get system current boot type.
BY_USER,
BY_ALARM,
BY_TRIGGER0,
BY_TRIGGER1

Return Value

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

Description

This function gets the type of system boot after setting.

See Also

Adv_DVPAPI_GetAlarm

Syntax

int Adv_DVPAPI_GetAlarm(int BoardID, int index, AlarmStruct* alarm)

Parameters

BoardID:	Specifies the board ID number(0~3).
index:	Specifies the alarm number(0~9).
alarm:	A AlarmStruct pointer to get alarm setting.

Return Value

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

Description

This function gets setting of specified alarm on a selected board.

See Also

Adv_DVPAPI_SetAlarm
AlarmStruct

Adv_DVPAPI_SetAlarm

Syntax

int Adv_DVPAPI_SetAlarm(int BoardID, int index, AlarmStruct* alarm)

Parameters

BoardID:	Specifies the board ID number(0~3).
index:	Specifies the alarm number(0~9).
alarm:	Alarm setting.

Return Value

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

Description

This function sets setting of specified alarm on a selected board.

See Also

Adv_DVPAPI_GetAlarm
AlarmStruct

Adv_DVPAPI_GetChecksum

Syntax

int Adv_DVPAPI_GetChecksum(int BoardID, BYTE* key, BYTE *checksum)

Parameters

BoardID:	Specifies the board ID number(0~3).
key:	Input key for check sum value.
checksum:	A pointer to get check sum value.

Return Value

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

Description

This function gets check sum value with a “KEY” input on a selected board.

See Also

Adv_DVPAPI_GetEEData

Syntax

int Adv_DVPAPI_GetEEData(int BoardID, BYTE wordAddr, BYTE* pData)

Parameters

BoardID: Specifies the board ID number(0~3).

wordAddr: Specifies the word address(0~127).

pData: A pointer to get byte value stored in EE.

Return Value

SUCCEEDED: Function succeeded.

FAILED: Function failed.

BOARDIDERROR: Invalid board ID.

PARAMERROR: Invalid parameter.

SDKINITFAILED: SDK not initialized.

Description

This function read the value at specified EE word address on a selected board.

See Also

Adv_DVPAPI_SetEEData

Adv_DVPAPI_SetEEData

Syntax

int Adv_DVPAPI_SetEEData(int BoardID, BYTE wordAddr, BYTE* pData)

Parameters

BoardID:	Specifies the board ID number(0~3).
wordAddr:	Specifies the word address(0~127).
pData:	A value to set the byte value in EE.

Return Value

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

Description

This function writes the value at specified EE word address on a selected board.

See Also

Adv_DVPAPI_GetEEData

Adv_DVPAPI_GetRTCData

Syntax

int Adv_DVPAPI_GetRTCData(int BoardID, TimeStruct* time)

Parameters

BoardID:	Specifies the board ID number(0~3).
time:	A TimeStruct pointer to get RTC

Return Value

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

Description

This function gets RTC settings on a selected board.

See Also

Adv_DVPAPI_SetRTCData
TimeStruct

Adv_DVPAPI_SetRTCData

Syntax

int Adv_DVPAPI_SetRTCData(int BoardID, TimeStruct* time)

Parameters

BoardID:	Specifies the board ID number(0~3).
time:	A TimeStruct pointer to set RTC

Return Value

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

Description

This function sets RTC settings on a selected board.

See Also

Adv_DVPAPI_GetRTCData
TimeStruct