ADLINK Vision Software

(AVS)

Manual Rev.:	1.0
Revision Date:	Oct. 12, 2021

Leading EDGE COMPUTING

Revision History

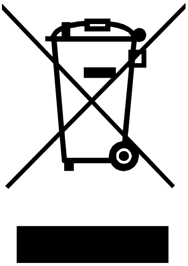
Revision	Release Date	Description of Change(s)
1.00	Oct. 12, 2021	Initial release

Preface

Disclaimer

The information in this document is subject to change without prior notice in order to improve reliability, design, and function and does not represent a commitment on the part of the manufacturer.

In no event will the manufacturer be liable for direct, indirect, special, incidental, or consequential damages arising out of the use or inability to use the product or documentation, even if advised of the possibility of such damages.



ADLINK is committed to fulfill its social responsibility to global environmental preservation through compliance with the European Union's Restriction of Hazardous Substances (RoHS) directive and Waste Electrical and Electronic Equipment (WEEE) directive. Environmental protection is a top priority for ADLINK. We have enforced measures to ensure that our products, manufacturing processes, components, and raw materials have as little impact on the environment as possible. When products are at their end of life, our customers are encour- aged to dispose of them in accordance with the product disposal and/or recovery programs prescribed by their nation or company.

Battery Labels (for products with battery)

Li-ion
RECYCLE
RBRC
Li·ion
1.800.822.8837

California Proposition 65 Warning

⚠ **WARNING:** This product can expose you to chemicals including acrylamide, arsenic, benzene, cadmium, Tris (1,3-dichloro-2-propyl)phosphate (TDCPP), 1,4-Dioxane, formaldehyde, lead, DEHP, styrene, DINP, BBP, PVC, and vinyl materials, which are known to the State of California to cause can- cer, and acrylamide, benzene, cadmium, lead, mercury, phthal- ates, toluene, DEHP, DIDP, DnHP, DBP, BBP, PVC, and vinyl materials, which are known to the State of California to cause birth defects or other reproductive harm. For more information go to www.P65Warnings.ca.gov.

Trademarks

Product names mentioned herein are used for identification pur- poses only and may be trademarks and/or registered trademarks of their respective companies.

Conventions

Take note of the following conventions used throughout this manual to make sure that users perform certain tasks and instructions properly.

✓ NOTE:	Additional information, aids, and tips that help users perform tasks.
⚠ CAUTION:	Information to prevent ***minor*** physical injury, component dam- age, data loss, and/or program corruption when trying to com- plete a task.
⚠ WARNING:	Information to prevent ***serious*** physical injury, component damage, data loss, and/or program corruption when trying to complete a specific task.

Table of Contents

This page intentionally left blank.

1. Introduction

The ADLINK Vision Software (AVS) is the application programming interface (API) for ADLINK vision products. AVS provides necessary details, where developers can develop applications in Linux and Windows OS, and support C/C++, C#/VB.NET languages. Along with the APIs, the AVS also includes product drivers and a set of sample programs and documentation.

1.1 Environment

Platform	x86, x86-x64
OS	Linux: Ubuntu 12.04 LTS (kernel 3.13) and above
	Windows: Windows 7/10
Programming language	Linux: C/C++
	Windows: C/C++, C#, VB.NET

1.2 AVS Installation

Linux	Debian, DKMS are must for AVS. $sudo apt-get install dkms $tar xvf **AVS-SDK_xxx.tar** $cd **AVS-SDK_xxx** $./install.sh
Windows	Perform "**AVS-SDK_x.x.x.xxxx.exe**" the installation

1.3 AVS Function & Parameter

AVS Function	Description	Installation Folder
General SDK functions	To get product handle and retrieve the device information	\AVS_SDK.h
Features function	Basic functions	\Feature*.h
AVS Parameters Reference	AVS parameters are grouped by C/C++ enumerations	\Common\I_common.h

1.4 AVS Sample program

Type	Description
GUI samples	An entry point for quickly performing samples with a GUI interface. This is for Windows only.
Console samples	The application that can be run in the command prompt in Windows and Linux.

1.5 AVS Installation Folder

1.5.1 Linux

Type	Installation Folder	File	Description
Include	/usr/local/include/ADLINK/AVS	AVS_SDK.h	Header file required for General SDK functions
	/usr/local/include/ADLINK/AVS/Common/	I_common.h	Header file required for AVS parameters
	/usr/local/include/ADLINK/AVS/Feature/	ExPowerSupply.h PowerController.h PoE.h ToE.h Fan.h	Header file required for basic functions
Library	/usr/local/lib/	libAVS_SDK.so libAVS_Product.so libAVS_CoreSDK.so	Exports API function definitions, required for Visual C/C++ applications
Sample Program	/usr/local/src/ADLINK/AVS/Sample	Build Sample $cd /usr/local/src/ADLINK/AVS/Sample $make Execute Sample $cd /usr/local/src/ ADLINK/AVS/Sample/bin $sudo ./AVS_GetSDKInfo	

1.5.2 Windows

Type	Installation Folder	File	Description
Include	\ADLINK\AVS\Include\C++	AVS_SDK.h I_common.h ExPowerSupply.h PowerController.h	Header file required for all C/C++ applications
	\ADLINK\AVS\Include\VB.NET	AVS_SDK.vb I_common.vb ExPowerSupply.vb PowerController.vb	Function definitions required for all VB.Net applications
	\ADLINK\AVS\Include\ C#	AVS_SDK.cs I_common.cs ExPowerSupply.cs PowerController.cs	Function definitions required for all C# applications
DLL	\ADLINK\AVS\Dll	AVS_SDK.dll AVS_Product.dll AVS_CoreSDK.dll	Dynamic link library files required for all applications
Library	\ADLINK\AVS\Dll	AVS_SDK.lib	Exports API function definitions, required for all Visual C/C++ applications
Sample Program	\ADLINK\AVS\Sample	Console: SampleGUI.exe GUI: VC++, C#/VB.ET	

1.6 Reference Code Snippet

1.6.1 Include head file

This code snippet shows how to include the head file.

```
#ifdef __linux__
#include "/usr/local/include/ADLINK/AVS/AVS_SDK.h"
#include "/usr/local/include/ADLINK/AVS/Common/I_common.h"
#elif _WIN32
#include "Include/C++/AVS_SDK.h"
#include "Include/C++/Common/I_common.h"
#endif
```

1.6.2 Get product handle

This code snippet shows how to get product handle and product Information.

```
using namespace std;
/*Show SDK Information*/
char buffer[256];
if (AVS_GetSDKInfo(buffer, sizeof(buffer)) == AVS_FUN_SUCCESS)
cout << buffer << endl;
...
/*Show Product Information*/
unsigned int nums = AVS_GetProductNums();
if (nums <= 0)
{
    cout << "Find no device!" << endl;
    return 0;
}

void* handle;
unsigned int devIndex = 0;
handle = AVS_GetProductHandle(devIndex);
if (!handle)
{
    cout << "AVS_GetProductHandle Error\n" << endl;
    return 0;
}

if (AVS_GetProductInfo(handle, buffer, sizeof(buffer)) == AVS_FUN_SUCCESS)
    cout << buffer << endl;
...
```

2. AVS Parameters Reference

The AVS parameters are grouped by enumerations and the enumerations can also be found in the header file “I_common”.

2.1. Product Name

ADLINK vision products supported by AVS SDK are grouped by **AVS_PRODUCT_TYPE** and listed.

String Identifier	Product Name
AVS_PRODUCT_UNKNOWN	Unknown product
AVS_PRODUCT_PCIEGIE72	PCIe-GIE72
AVS_PRODUCT_PCIEGIE74	PCIe-GIE74
AVS_PRODUCT_PCIEGIE72PRO	PCIe-GIE72 PRO
AVS_PRODUCT_PCIEGIE74PRO	PCIe-GIE74 PRO
AVS_PRODUCT_PCIEU304	PCIe-U304
AVS_PRODUCT_PCIEU308	PCIe-U308
AVS_PRODUCT_PCIEU312	PCIe-U312
AVS_PRODUCT_PCIE10GPOE	PCIe-10GPoE

2.2. Product Feature

The AVS parameters are grouped by **AVS_FEATURE_TYPE**.

String Identifier	Product Feature
AVS_FEATURE_POWERCONTROLLER	USB3 Power Management
AVS_FEATURE_POE	Power over Ethernet (PoE)
AVS_FEATURE_TOE	Trigger over Ethernet (ToE)
AVS_FEATURE_DIO	Digital Input / Output
AVS_FEATURE_EXTERNALPOWERSUPPLY	External power supply detection
AVS_FEATURE_NUMS	The number of the product feature

2.3. Edge Trigger

Parameters of the Edge Trigger, defined by AVS_ EDGETRIGGER_TYPE.

String Identifier	Parameter
AVS_EDGETRIGGER_RISINGEDGE	Signal rising edge trigger
AVS_EDGETRIGGER_FALLINGEDGE	Signal falling edge trigger
AVS_EDGETRIGGER_CHANGINGEDGE	Signal changing edge trigger
AVS_EDGETRIGGER_NUMS	The number of the edge trigger type

2.4. PoE

Parameters of PoE defined by AVS_POEPROPERTY_TYPE.

String Identifier	Parameter
AVS_POEPROPERTY_POEHWSETTING	PSE Hardware setting
AVS_POEPROPERTY_POEINITIALSTATE	PSE initial power
AVS_POEPROPERTY_OVERTEMPERATURE	Check if the device temperature is over high temperature (degree C)
AVS_POEPROPERTY_CURRENTTEMPERATURE	Get the current temperature
AVS_POEPROPERTY_CONSUMEPOWER	The consumer power(Watt)
AVS_POEPROPERTY_REMAINPOWER	The left power(Watt)
AVS_POEPROPERTY_POWERBUDGET	The total power budget(Watt)
AVS_POEPROPERTY_NUMS	The number of the PoE property

2.5. PoE Power Management

Parameters of PoE Power Management defined by AVS_POEPORTPROPERTY_TYPE.

String Identifier	Parameter
AVS_POEPORTPROPERTY_CURRENT	The current of a port (A)
AVS_POEPORTPROPERTY_VOLT	The volt of a port
AVS_POEPORTPROPERTY_POECLASS	The power class of a port
AVS_POEPORTPROPERTY_POWERGOOD	Check if the power is good of a port
AVS_POEPORTPROPERTY_LASTPOWERSTATE	Retrieve the last power state of a port
AVS_POEPORTPROPERTY_NUMS	The number of the PoE port properties

2.6. AVS Error Code

AVS error cod is defined by AVS_FUN_RETURNVAL. No error occurs if API return value >= 0, and if a negative value, please refer to below table.

Error Code	Definition	Description
0	AVS_FUN_SUCCESS	Execution successful
-1	AVS_FUN_FAILED	Execution failed
-5	AVS_FUN_NOTDEFINED	Unsupported function
-11	AVS_FUN_INVALIDPARA	The parameters are invalid
-13	AVS_FUN_NOTINIT	The object is not initialized
-100	AVS_FUN_BUSY	System is busy

3. Function Library

3.1 List of Functions

Category (Header File)	Function	Product
Common (AVS_SDK.h)	AVS_GetProductNums()	All Product
	AVS_GetSDKInfo()	
	AVS_GetProductHandle()	
	AVS_GetProductInfo()	
	AVS_GetProductType()	
	AVS_GetProductSN()	
	AVS_GetFeatureList()	
	AVS_GetCardID()	
USB Port Control (PowerController.h)	AVS_PowCtrlGetPortNums()	PCIe-U300 Series
	AVS_PowCtrlSetPortEnable()	
	AVS_PowCtrlGetPortEnable()	
External Power Supply (ExPowerSupply.h)	AVS_ExPowSupGetNums()	All Product
	AVS_ExPowSupGetState()	
Power over Ethernet (PoE.h)	AVS_PoEGetPortNums()	PCIe-GIE Series PCIe-10G Series
	AVS_PoESetPortEnable()	
	AVS_PoEGetPortEnable()	
	AVS_PoESetPowConsumCalcModel()	
	AVS_PoEGetPowConsumCalcModel()	
	AVS_PoESetProtectTempRange()	
	AVS_PoEGetProtectTempRange()	
	AVS_PoEGetProperty()	
	AVS_PoEGetPortProperty()	
Trigger over Ethernet (ToE.h)	AVS_ToEGetPortNums()	PCIe-GIE7x PRO, PCIe-10G Series
	AVS_ToESendActionCommand()	
	AVS_ToESetActionCommand()	
	AVS_ToEGetActionCommand()	
	AVS_ToESetTriggerSource()	
	AVS_ToEGetTriggerSource()	
	AVS_ToESetTriggerEnable()	
	AVS_ToEGetTriggerEnable()	
	AVS_ToESetTriggerActivation()	

	AVS_ToEGetTriggerActivation()	
	AVS_ToESetOneTriggerAllMode()	
	AVS_ToEGetOneTriggerAllMode()	
	AVS_ToESetExTriggerDebounce()	
	AVS_ToEGetExTriggerDebounce()	
	AVS_ToEGetExTriggerDebounceUnit()	
	AVS_ToEGetTriggerCount()	
	AVS_ToEResetTriggerCount()	

3.2 Function Library

AVS supports C/C++/C#/VB.NET programming language. The following is the example of C/C++ library.

3.2.1 Common

These functions are used to get product handle and retrieval device information.

3.2.1.1 AVS_GetProductNums

Function Name	AVS_GetProductNums
Purpose	This function returns the total number of products is installed in the system.
Category	Common
Header File	AVS_SDK.h
Syntax (C/C++)	unsigned int AVS_GetProductNums()
Parameter(s)	
Return Code	
Returns the total number of devices if return value >= 0; else please refer to Section 2 for more information about return codes.	

3.2.1.2 AVS_GetSDKInfo

Function Name	AVS_GetSDKInfo
Purpose	This function returns SDK version information, e.g., driver version, SDK version…etc.
Category	Common
Header File	AVS_SDK.h
Syntax (C/C++)	int AVS_GetSDKInfo(char* data, unsigned int bufsize)
Parameter(s)	
data: It is a string buffer indicates the parameters and information provided. Each parameter and information is split by a break line character '\n'. The format is **Parameter Name:Parameter Information '\n'** For example: **Driver version:0.2.0.0929** **SDK version:1.3.6.0121** **bufsize:** Indicates the size of "data" buffer. The suggested value is 256.	
Return Code	
Returns the total number of devices if return value >= 0; else please refer to Section 2 for more information about return codes.	

3.2.1.3 AVS_GetProductHandle

Function Name	AVS_GetProductHandle
Purpose	Initializes the specified device and get handle, should be called before other functions except those with no handle parameter.
Category	Common
Header File	AVS_SDK.h
Syntax (C/C++)	void* AVS_GetProductHandle(unsigned int index)
Parameter(s)	
index: Indicates the number of devices, beginning at 0 for the number of the first card, with the second card 1, and so on. This value should be less than (total numbers of the cards -1), the total number of the cards is acquired from AVS_GetProductNums.	
Return Code	
No error occurs if the return value >= 0, and if a negative value, please refer to Section 2 for more information about return codes.	

3.2.1.4 AVS_GetProductInfo

Function Name	AVS_GetProductInfo
Purpose	This function returns all of the product information, e.g., firmware version, product number…etc.
Category	Common
Header File	AVS_SDK.h
Syntax (C/C++)	int AVS_GetProductInfo(void* handle, char* data, unsigned int bufsize)
Parameter(s)	
handle: Acquired from the return value of AVS_GetProductHandle. **data:** It is a string buffer indicates the parameters and information provided. Each parameter and information is split by a break line character '\n'. The format is **Parameter Name:Parameter Information '\n'** For example: **Product:PCIe-10GPoE** **CPLDVersion:20210119_1610** **MCUVersion:2.3** **PN:91-64216-000E** **SN:0123456789** **PCBVersion:1** **MAC0:00:30:64:35:c4:26** **MAC1:00:30:64:35:c4:26** **Bufsize:** Indicates the size of "data" buffer. The suggested value is 256.	
Return Code	
No error occurs if the return value >= 0, and if a negative value, please refer to Section 2 for more information about return codes.	

3.2.1.5 AVS_GetProductType

Function Name	AVS_GetProductType
Purpose	Get the product type.
Category	Common
Header File	AVS_SDK.h
Syntax (C/C++)	int AVS_GetProductType(void* handle)
Parameter(s)	
handle: Acquired from the return value of AVS_GetProductHandle.	
Return Code	
Return the AVS_PRODUCT_TYPE, no error occurs if the return value >= 0, and if a negative value, please refer to Section 2 for more information about return codes.	

3.2.1.6 AVS_GetProductSN

Function Name	AVS_GetProductSN
Purpose	This function gets product serial number (SN).
Category	Common
Header File	AVS_SDK.h
Syntax (C/C++)	int AVS_GetProductSN(void* handle, char* data, unsigned int bufsize)
Parameter(s)	
handle: Acquired from the return value of AVS_GetProductHandle. **data:** Pointer of an allocated character buffer into the function copies the string. **Bufsize:** Indicates the size of “data” buffer. The suggested value is 32.	
Return Code	
No error occurs if the return value >= 0, and if a negative value, please refer to Section 2 for more information about return codes.	

3.2.1.7 AVS_GetFeatureList

Function Name	AVS_GetFeatureList
Purpose	This function gets product feature list.
Category	Common
Header File	AVS_SDK.h
Syntax (C/C++)	int AVS_GetFeatureList(void* handle, AVS_FEATURE_TYPE* list, unsigned int listsize)
Parameter(s)	
handle: Acquired from the return value of AVS_GetProductHandle. **list:** Pointer of an allocated buffer into the function copies the string. **listsize:** Indicates the size of “list” buffer. The suggested value is AVS_FEATURE_NUMS (AVS_FEATURE_NUMS is definded in I_common.h head file).	
Return Code	
No error occurs if the return value >= 0, and if a negative value, please refer to Section 2 for more information about return codes.	

3.2.1.8 AVS_GetCardID

Function Name	AVS_GetCardID
Purpose	This function gets a numeric card ID that corresponds to the handle.
Category	Common
Header File	AVS_SDK.h
Syntax (C/C++)	int AVS_GetCardID(void* handle, int* id)
Parameter(s)	
handle: Acquired from the return value of AVS_GetProductHandle. **id:** Indicates the card ID set by SW1 switch, a value from 0 to 15. Card ID and SW1 switch settings correlate as shown. Please refer to the table below.	
Return Code	
No error occurs if the return value >= 0, and if a negative value, please refer to Section 2 for more information about return codes.	

Card ID (id)	**Hardware Switch**			
	4	**3**	**2**	**1**
0	off	off	off	off
1	off	off	off	on
2	off	off	on	off
3	off	off	on	on
4	off	on	off	off
5	off	on	off	on
6	off	on	on	off
7	off	on	on	on
8	on	off	off	off
9	on	off	off	on
10	on	off	on	off
11	on	off	on	on
12	on	on	off	off
13	on	on	off	on
14	on	on	on	off
15	on	on	on	on

3.2.2 USB Port Control

This feature is use to control the power device.

3.2.2.1 AVS_PowCtrlGetPortNums

Function Name	AVS_PowCtrlGetPortNums
Purpose	This function gets a number of ports that corresponds to the handle.
Category	USB Port Control
Header File	PowerController.h
Syntax (C/C++)	int AVS_PowCtrlGetPortNums(void* handle, unsigned int* nums)
Parameter(s)	
handle: Acquired from the return value of AVS_GetProductHandle. **nums:** Number of ports.	
Return Code	
No error occurs if the return value >= 0, and if a negative value, please refer to Section 2 for more information about return codes.	

3.2.2.2 AVS_PowCtrlSetPortEnable

Function Name	AVS_PowCtrlSetPortEnable
Purpose	Controls power auto/off.
Category	USB Port Control
Header File	PowerController.h
Syntax (C/C++)	int AVS_PowCtrlSetPortEnable(void* handle, int* list, unsigned int listsize)
Parameter(s)	
handle Acquired from the return value of AVS_GetProductHandle. **list** An array to control power auto/off of each port. The index of the array corresponds to the index of port, the element of the array indicates to enable or disable the power. **listsize** The array must contain nums elements (nums is acquired from the return value of AVS_PowCtrlGetPortNums).	
Return Code	
No error occurs if the return value >= 0, and if a negative value, please refer to Section 2 for more information about return codes.	

3.2.2.3 AVS_PowCtrlGetPortEnable

Function Name	AVS_PowCtrlGetPortEnable
Purpose	Retrieves power auto/off.
Category	USB Port Control
Header File	PowerController.h
Syntax (C/C++)	int AVS_PowCtrlGetPortEnable(void* handle, int* list, unsigned int listsize)
Parameter(s)	
handle Acquired from the return value of AVS_GetProductHandle. **list** An array for receiving the power status from the ports. **listsize** The array must contain nums elements (nums is acquired from the return value of AVS_PowCtrlGetPortNums).	
Return Code	
No error occurs if the return value >= 0, and if a negative value, please refer to Section 2 for more information about return codes.	

3.2.3 External Power Supply

These functions are used for status of external power supply cables.

3.2.3.1 AVS_ExPowSupGetNums

Function Name	AVS_ExPowSupGetNums
Purpose	Get the number of the external power cable is connected.
Category	External Power Supply
Header File	ExPowerSupply.h
Syntax (C/C++)	int AVS_ExPowSupGetNums(void* handle, unsigned int* nums)
Parameter(s)	
handle Acquired from the return value of AVS_GetProductHandle. **nums** Number of the external power cable is connected.	
Return Code	
No error occurs if the return value >= 0, and if a negative value, please refer to Section 2 for more information about return codes.	

3.2.3.2 AVS_ExPowSupGetState

Function Name	AVS_ExPowSupGetState
Purpose	Get the detection results of external power supply cables.
Category	External Power Supply
Header File	ExPowerSupply.h
Syntax (C/C++)	int AVS_ExPowSupGetState(void* handle, int* statelist, unsigned int listsize)
Parameter(s)	
handle Acquired from the return value of AVS_GetProductHandle. **statelist** An array for detection results of external power cables. “0” indicates power is not supplied, the other values indicate power is supplied. **listsize** The array must contain nums elements (nums is acquired from the return value of AVS_ExPowSupGetNums).	
Return Code	
No error occurs if the return value >= 0, and if a negative value, please refer to Section 2 for more information about return codes.	

3.2.4 Power over Ethernet

This feature is for SmartPoE, the PoE power management functions.

3.2.4.1 AVS_PoEGetPortNums

Function Name	AVS_PoEGetPortNums
Purpose	Retrieve the number of the power over Ethernet.
Category	Power over Ethernet
Header File	PoE.h
Syntax (C/C++)	int AVS_PoEGetPortNums(void* handle, unsigned int* nums)
Parameter(s)	
handle Acquired from the return value of AVS_GetProductHandle. **nums** Return the number of the power over Ethernet.	
Return Code	
No error occurs if the return value >= 0, and if a negative value, please refer to Section 2 for more information about return codes.	

3.2.4.2 AVS_PoESetPortEnable

Function Name	AVS_PoESetPortEnable
Purpose	Controls power auto/off.
Category	Power over Ethernet
Header File	PoE.h
Syntax (C/C++)	int AVS_PoESetPortEnable(void* handle, int* list, unsigned int listsize)
Parameter(s)	
handle Acquired from the return value of AVS_GetProductHandle. **list** An array for setting the power status of the ports. Assign a list to enable or disable each port. * 0 to disable power or other to enable power. **listsize** The array must contain nums elements (nums is acquired from the return value of AVS_PowCtrlGetPortNums).	
Return Code	
No error occurs if the return value >= 0, and if a negative value, please refer to Section 2 for more information about return codes.	

3.2.4.3 AVS_PoEGetPortEnable

Function Name	AVS_PoEGetPortEnable
Purpose	Retrieves PoE state.
Category	Power over Ethernet
Header File	PoE.h
Syntax (C/C++)	int AVS_PoEGetPortEnable(void* handle, int* list, unsigned int listsize)
Parameter(s)	
handle Acquired from the return value of AVS_GetProductHandle. **list** An array for receiving the power status of the ports. **listsize** The array must contain nums elements (nums is acquired from the return value of AVS_PowCtrlGetPortNums).	
Return Code	
No error occurs if the return value >= 0, and if a negative value, please refer to Section 2 for more information about return codes.	

3.2.4.4 AVS_PoESetPowConsumCalcModel

Function Name	AVS_PoESetPowConsumCalcModel
Purpose	Controls power budget mode.
Category	Power over Ethernet
Header File	PoE.h
Syntax (C/C++)	int AVS_PoESetPowConsumCalcModel(void* handle, int usePowClassBudget)
Parameter(s)	
handle Acquired from the return value of AVS_GetProductHandle. **usePowClassBudget** 0: calculate the Consumed Power Budget by PoE power consumption. 1: calculate the Consumed Power Budget by PoE Class.	
Return Code	
No error occurs if the return value >= 0, and if a negative value, please refer to Section 2 for more information about return codes.	

3.2.4.5 AVS_PoEGetPowConsumCalcModel

Function Name	AVS_PoEGetPowConsumCalcModel
Purpose	Retrieves power budget mode.
Category	Power over Ethernet
Header File	PoE.h
Syntax (C/C++)	int AVS_PoEGetPowConsumCalcModel(void* handle, int* usePowClassBudget)
Parameter(s)	
handle Acquired from the return value of AVS_GetProductHandle. **usePowClassBudget** 0: calculate the Consumed Power Budget by PoE power consumption. 1: calculate the Consumed Power Budget by PoE Class.	
Return Code	
No error occurs if the return value >= 0, and if a negative value, please refer to Section 2 for more information about return codes.	

3.2.4.6 AVS_PoESetProtectTempRange

Function Name	AVS_PoESetProtectTempRange
Purpose	Controls of High Temperature (HT) and Recovery Temperature (RT). When HT is reached, automatically shuts down PoE power until the temperature sensor falls below RT.
Category	Power over Ethernet
Header File	PoE.h
Syntax (C/C++)	int AVS_PoESetProtectTempRange(void* handle, int high_temperature, int recovery_temperature)
Parameter(s)	
handle Acquired from the return value of AVS_GetProductHandle. **high_temperature** High temperature setting, more than or equal to 80°C, not exceeding 130°C, and at least 10°C higher than the Recovery Temperature setting. 1 byte data in 1°C unit. **recovery_temperature** Recovery temperature setting, equal to or exceeding 70°C and at least 10°C lower than the high_temperature setting.	
Return Code	
No error occurs if the return value >= 0, and if a negative value, please refer to Section 2 for more information about return codes.	

3.2.4.7 AVS_PoEGetProtectTempRange

Function Name	AVS_PoEGetProtectTempRange
Purpose	Retrieves of High Temperature and Recovery Temperature.
Category	Power over Ethernet
Header File	PoE.h
Syntax (C/C++)	int AVS_PoEGetProtectTempRange(void* handle, int high_temperature, int recovery_temperature)

Parameter(s)

handle

Acquired from the return value of AVS_GetProductHandle.

high_temperature

High Temperature setting.

recovery_temperature

Recovery Temperature setting.

Return Code

No error occurs if the return value >= 0, and if a negative value, please refer to Section 2 for more information about return codes.

3.2.4.8 AVS_PoEGetProperty

Function Name	AVS_PoEGetProperty
Purpose	Retrieves of PoE properties.
Category	Power over Ethernet
Header File	PoE.h
Syntax (C/C++)	int AVS_PoEGetProperty(void* handle, AVS_POEPROPERTY_TYPE property, float* value)

Parameter(s)

handle

Acquired from the return value of AVS_GetProductHandle.

property & value

Please refer to following table.

Return Code

No error occurs if the return value >= 0, and if a negative value, please refer to Section 2 for more information about return codes.

property	**value**
AVS_POEPROPERTY_POEHWSETTING	0: PoE-PSE HW setting disabled 1: PoE-PSE HW setting enabled
AVS_POEPROPERTY_POEINITIALSTATE	0: PoE-PSE should be turned off when powered on 1: PoE-PSE should be turned on

	when powered on.
AVS_POEPROPERTY_OVERTEMPERATURE	0: no interrupt 1: over temperature interrupt arise
AVS_POEPROPERTY_CURRENTTEMPERATURE	Thermal temperature in °C.
AVS_POEPROPERTY_CONSUMEPOWER	Consumed power budget in W.
AVS_POEPROPERTY_REMAINPOWER	Remaining available power in W.
AVS_POEPROPERTY_POWERBUDGET	Total Budget in W.
AVS_POEPROPERTY_NUMS	

3.2.4.9 AVS_PoEGetPortProperty

Function Name	AVS_PoEGetPortProperty
Purpose	Retrieves of PoE port properties.
Category	Power over Ethernet
Header File	PoE.h
Syntax (C/C++)	int AVS_PoEGetPortProperty(void* handle, unsigned int port, AVS_POEPORTPROPERTY_TYPE property, float* value)
Parameter(s)	
handle Acquired from the return value of AVS_GetProductHandle. **port** Indicates port number from 0 to N. (N is the number of port) **property & value** Please refer to following table.	
Return Code	
No error occurs if the return value >= 0, and if a negative value, please refer to Section 2 for more information about return codes.	

String Identifier	**Value**
AVS_POEPORTPROPERTY_CURRENT	Port PoE power current (unit: A)
AVS_POEPORTPROPERTY_VOLT	Port PoE power voltage (unit: V)
AVS_POEPORTPROPERTY_POECLASS	Port PoE Class
AVS_POEPORTPROPERTY_POWERGOOD	Port Power Good (unit: Watt)
AVS_POEPORTPROPERTY_LASTPOWERSTATE	0 : PoE power off 1 : PoE power on

3.2.5 Trigger over Ethernet

ToE is to send a preconfigured ToE action commands to GigE Vision camera when detecting a rising/falling edge of external trigger signal. User also can force the software ToE action commands by AVS SDK. The following shows the basic steps.

Procedure	AVS Function Name
1. Select Trigger Source	AVS_ToESetTriggerSource
2. Configure Action Command	AVS_ToESetActionCommand
3. Set Trigger Activation	AVS_ToESetTriggerActivation
4. Enable ToE function	AVS_ToESetTriggerEnable
5. Manually send ToE	AVS_ToESendActionCommand
6. ToE counters	AVS_ToEGetTriggerCount

3.2.5.1 AVS_ToEGetPortNums

Function Name	AVS_ToEGetPortNums
Purpose	Retrieves the number of ToE port.
Category	Trigger over Ethernet
Header File	ToE.h
Syntax (C/C++)	int AVS_ToEGetPortNums(void* handle, unsigned int* nums)
Parameter(s)	
handle Acquired from the return value of AVS_GetProductHandle. **nums** Number of ToE ports.	
Return Code	
No error occurs if the return value >= 0, and if a negative value, please refer to Section 2 for more information about return codes.	

3.2.5.2 AVS_ToESendActionCommand

<table>
<tr><td><b>Function Name</b></td><td>AVS_ToESendActionCommand</td></tr>
<tr><td><b>Purpose</b></td><td>Sends a software ToE action command over the selected port.</td></tr>
<tr><td><b>Category</b></td><td>Trigger over Ethernet</td></tr>
<tr><td><b>Header File</b></td><td>ToE.h</td></tr>
<tr><td><b>Syntax (C/C++)</b></td><td>int AVS_ToESendActionCommand(void* handle, unsigned int port)</td></tr>
<tr><td colspan="2"><b>Parameter(s)</b></td></tr>
<tr><td colspan="2"><b>handle</b>
Acquired from the return value of AVS_GetProductHandle.
<b>port</b>
Indicates port number from 0 to N. (N is the number of port)
-1 indicates to send ToE action command over all ports.</td></tr>
<tr><td colspan="2"><b>Return Code</b></td></tr>
<tr><td colspan="2">No error occurs if the return value >= 0, and if a negative value, please refer to Section 2 for more information about return codes.</td></tr>
</table>

3.2.5.3 AVS_ToESetActionCommand

<table>
<tr><td><b>Function Name</b></td><td>AVS_ToESetActionCommand</td></tr>
<tr><td><b>Purpose</b></td><td>Configures ToE action command for each port.</td></tr>
<tr><td><b>Category</b></td><td>Trigger over Ethernet</td></tr>
<tr><td><b>Header File</b></td><td>ToE.h</td></tr>
<tr><td><b>Syntax (C/C++)</b></td><td>int AVS_ToESetActionCommand(void* handle, unsigned int port, unsigned long devicekey, unsigned long groupkey, unsigned long groupmask)</td></tr>
<tr><td colspan="2"><b>Parameter(s)</b></td></tr>
<tr><td colspan="2"><b>handle</b>
Acquired from the return value of AVS_GetProductHandle.
<b>port</b>
Indicates port number from 0 to N. (N is the number of port)
<b>Devicekey</b>
32-bit number of your choice, the key must be unique among all cameras in the network segment. The Devicekey on the camera should be the same on the card.
<b>Groupkey</b>
A 32-bit number of your choice used to define a group of devices on which an action should be executed. If the action group key on the camera and the action group key in the protocol message are identical, the camera executes the corresponding action. The Groupkey on the camera should be the same on the card.
<b>Groupmask</b>
A 32-bit number of your choice used to filter out a sub-group of cameras belonging to a group of cameras. The cameras belonging to a sub-group execute an action at the same time. The Groupmask on the camera should be the same on the card.</td></tr>
</table>

Return Code
No error occurs if the return value >= 0, and if a negative value, please refer to Section 2 for more information about return codes.

3.2.5.4 AVS_ToEGetActionCommand

Function Name	AVS_ToEGetActionCommand
Purpose	Retrieves ToE action command for each port.
Category	Trigger over Ethernet
Header File	ToE.h
Syntax (C/C++)	int AVS_ToEGetActionCommand(void* handle, unsigned int port, unsigned long* devicekey, unsigned long* groupkey, unsigned long* groupmask)

Parameter(s)
handle Acquired from the return value of AVS_GetProductHandle. **port** Indicates port number from 0 to N. (N is the number of port) **Devicekey** Retrieves Devicekey on the card. **Groupkey** Retrieves Groupkey on the card. **Groupmask** Retrieves Groupmask on the card.

Return Code
No error occurs if the return value >= 0, and if a negative value, please refer to Section 2 for more information about return codes.

3.2.5.5 AVS_ToESetTriggerSource

Function Name	AVS_ToESetTriggerSource
Purpose	Configures ToE function source.
Category	Trigger over Ethernet
Header File	ToE.h
Syntax (C/C++)	int AVS_ToESetTriggerSource(void* handle, int source)
Parameter(s)	
handle Acquired from the return value of AVS_GetProductHandle. **source** 0: Software trigger source 1: External hardware trigger source	
Return Code	
No error occurs if the return value >= 0, and if a negative value, please refer to Section 2 for more information about return codes.	

3.2.5.6 AVS_ToEGetTriggerSource

Function Name	AVS_ToEGetTriggerSource
Purpose	Acquires ToE function source.
Category	Trigger over Ethernet
Header File	ToE.h
Syntax (C/C++)	int AVS_ToEGetTriggerSource(void* handle, int *source)
Parameter(s)	
handle Acquired from the return value of AVS_GetProductHandle. **source** 0: Software trigger source 1: External hardware trigger source	
Return Code	
No error occurs if the return value >= 0, and if a negative value, please refer to Section 2 for more information about return codes.	

3.2.5.7 AVS_ToESetTriggerEnable

Function Name	AVS_ToESetTriggerEnable
Purpose	Enable ToE function.
Category	Trigger over Ethernet
Header File	ToE.h
Syntax (C/C++)	int AVS_ToESetTriggerEnable(void* handle, int enable)
Parameter(s)	
handle Acquired from the return value of AVS_GetProductHandle. **enable** 0: Disable ToE function 1: Enable ToE function	
Return Code	
No error occurs if the return value >= 0, and if a negative value, please refer to Section 2 for more information about return codes.	

3.2.5.8 AVS_ToEGetTriggerEnable

Function Name	AVS_ToEGetTriggerEnable
Purpose	Retrieves ToE function.
Category	Trigger over Ethernet
Header File	ToE.h
Syntax (C/C++)	int AVS_ToEGetTriggerEnable(void* handle, int *enable)
Parameter(s)	
handle Acquired from the return value of AVS_GetProductHandle. **enable** 0: Disable ToE function 1: Enable ToE function	
Return Code	
No error occurs if the return value >= 0, and if a negative value, please refer to Section 2 for more information about return codes.	

3.2.5.9 AVS_ToESetTriggerActivation

Function Name	AVS_ToESetTriggerActivation
Purpose	Retrieves ToE function status.
Category	Trigger over Ethernet
Header File	ToE.h
Syntax (C/C++)	int AVS_ToESetTriggerActivation(void* handle, AVS_EDGETRIGGER_TYPE edgetrigger)

Parameter(s)
handle Acquired from the return value of AVS_GetProductHandle. **edgetrigger** Set the edge trigger mode AVS_EDGETRIGGER_TYPE, Please refer to following table.
Return Code
No error occurs if the return value >= 0, and if a negative value, please refer to Section 2 for more information about return codes.

String Identifier	Parameter
AVS_EDGETRIGGER_RISINGEDGE	Signal rising edge trigger
AVS_EDGETRIGGER_FALLINGEDGE	Signal falling edge trigger
AVS_EDGETRIGGER_CHANGINGEDGE	Signal changing edge trigger

3.2.5.10 AVS_ToEGetTriggerActivation

Function Name	AVS_ToEGetTriggerActivation
Purpose	Acquires ToE trigger activation mode of ToE, specifying that the source trigger is considered valid on the rising or falling edge.
Category	Trigger over Ethernet
Header File	ToE.h
Syntax (C/C++)	int AVS_ToEGetTriggerActivation (void* handle, AVS_EDGETRIGGER_TYPE* edgetrigger)
Parameter(s)	
handle Acquired from the return value of AVS_GetProductHandle. **edgetrigger** Please refer to following table.	
Return Code	
No error occurs if the return value >= 0, and if a negative value, please refer to Section 2 for more information about return codes.	

String Identifier	Parameter
AVS_EDGETRIGGER_RISINGEDGE	Signal rising edge trigger
AVS_EDGETRIGGER_FALLINGEDGE	Signal falling edge trigger
AVS_EDGETRIGGER_CHANGINGEDGE	Signal changing edge trigger

3.2.5.11 AVS_ToESetOneTriggerAllMode

Function Name	AVS_ToESetOneTriggerAllMode
Purpose	Sets ToE trigger mode, where 4 to 4 mode indicates that each pin of the DI performs corresponding port action command to active, and 1 to 4 mode indicates that DI_0 performs all port action commands to active.
Category	Trigger over Ethernet
Header File	ToE.h
Syntax (C/C++)	int AVS_ToESetOneTriggerAllMode(void* handle, int *enable)
Parameter(s)	
handle Acquired from the return value of AVS_GetProductHandle. **enable** 0: N to N mode (N is the number of port) 1: 1 to N mode (N is the number of port)	
Return Code	
No error occurs if the return value >= 0, and if a negative value, please refer to Section 2 for more information about return codes.	

3.2.5.12 AVS_ToEGetOneTriggerAllMode

Function Name	AVS_ToEGetOneTriggerAllMode
Purpose	Acquires ToE trigger mode, where 4 to 4 mode indicates that each pin of the DI performs corresponding port action command to active, and 1 to 4 mode indicates that DI_0 performs all port action commands to active.
Category	Trigger over Ethernet
Header File	ToE.h
Syntax (C/C++)	int AVS_ToESetOneTriggerAllMode(void* handle, int *enable)
Parameter(s)	
handle Acquired from the return value of AVS_GetProductHandle. **enable** 0: N to N mode (N is the number of port) 1: 1 to N mode (N is the number of port)	
Return Code	
No error occurs if the return value >= 0, and if a negative value, please refer to Section 2 for more information about return codes.	

3.2.5.13 AVS_ToESetExTriggerDebounce

Function Name	AVS_ToESetExTriggerDebounce
Purpose	Sets trigger de-bounce time for filtering the external trigger.
Category	Trigger over Ethernet
Header File	ToE.h
Syntax (C/C++)	int AVS_ToESetExTriggerDebounce(void* handle, unsigned long debounce)
Parameter(s)	
handle Acquired from the return value of AVS_GetProductHandle. **debounce** Set the debounce time, input range: 0x0 ~ 0xFFFFFF.	
Return Code	
No error occurs if the return value >= 0, and if a negative value, please refer to Section 2 for more information about return codes.	

3.2.5.14 AVS_ToEGetExTriggerDebounce

Function Name	AVS_ToEGetExTriggerDebounce
Purpose	Acquires trigger de-bounce time for filtering the external trigger.
Category	Trigger over Ethernet
Header File	ToE.h
Syntax (C/C++)	int AVS_ToEGetExTriggerDebounce(void* handle, unsigned long* debounce)
Parameter(s)	
handle Acquired from the return value of AVS_GetProductHandle. **debounce** The debounce time, range: 0x0 ~ 0xFFFFFF.	
Return Code	
No error occurs if the return value >= 0, and if a negative value, please refer to Section 2 for more information about return codes.	

3.2.5.15 AVS_ToEGetExTriggerDebounceUnit

Function Name	AVS_ToEGetExTriggerDebounceUnit
Purpose	Acquires trigger de-bounce unit in nano second.
Category	Trigger over Ethernet
Header File	ToE.h
Syntax (C/C++)	int AVS_ToEGetExTriggerDebounceUnit(void* handle, unsigned long* nanosec)

Parameter(s)

handle

Acquired from the return value of AVS_GetProductHandle.

nanosec

Time in nano second.

Return Code

No error occurs if the return value >= 0, and if a negative value, please refer to Section 2 for more information about return codes.

3.2.5.16 AVS_ToEGetTriggerCount

Function Name	AVS_ToEGetTriggerCount
Purpose	Acquires trigger count, directing 16-bit counter to count triggers originating from hardware or software and ToE commands sent from the card.
Category	Trigger over Ethernet
Header File	ToE.h
Syntax (C/C++)	int AVS_ToEGetTriggerCount(void* handle, unsigned int port, unsigned int* triggerincount, unsigned int* triggeroutcount)

Parameter(s)

handle

Acquired from the return value of AVS_GetProductHandle.

port

Indicates port number from 0 to N. (N is the number of port)

triggerincount

Number of triggers from hardware trigger or software trigger (Valid value: 0 to 65535)

triggeroutcount

Number of action commands from the card (Valid value: 0 to 65535)

Return Code

No error occurs if the return value >= 0, and if a negative value, please refer to Section 2 for more information about return codes.

3.2.5.17 AVS_ToEResetTriggerCount

Function Name	AVS_ToEResetTriggerCount
Purpose	Resets trigger count, with trigger counters reset to 0.
Category	Trigger over Ethernet
Header File	ToE.h
Syntax (C/C++)	int AVS_ToEResetTriggerCount (void* handle)
Parameter(s)	
handle Acquired from the return value of AVS_GetProductHandle.	
Return Code	
No error occurs if the return value >= 0, and if a negative value, please refer to Section 2 for more information about return codes.	