DC100 TOF Sensor Guide

Release Version：V1.0.0

Release Date：2023－12－14

Security Level： □Top−Secret □Secret □Internal ■Public

Overview

This document aims to describe TOF Sensor related API interfaces.

Product Version

Chipset	Kernel Version
RV1126	Linux 4.19

Intended Audience

This document (this guide) is mainly applicable to the following engineers:

- Software development engineer
- Software Development Engineer

Revision History

Version	Author	Date	Revision History
V1.0.0	LEE	2023-12-14	Initial version

Contents

DC100 TOF Sensor Guide

1. OVERVIEW 6

1.1 OVERVIEW 6

1.2 API INVOKE PROCESS 6

1.2.1 Ps2_Initialize&Ps2_Shutdown........ 6

1.2.2 Ps2_GetDeviceCount &Ps2_GetDeviceListInfo/Ps2_GetDeviceInfo 6

1.2.3 Ps2_GetDeviceCount &Ps2_GetDeviceListInfo/Ps2_GetDeviceInfo 6

1.2.4 Ps2_StartStream&Ps2_StopStream........ 7

1.2.5 Ps2_ReadNextFrame&Ps2_GetFrame........ 7

1.2.6 Set&Get 7

2. API INTRODUCTION........ 7

2.1 API REFERENCE 7

2.1.1 Ps2_Initialize 7

2.1.2 Ps2_Shutdown 8

2.1.3 Ps2_GetDeviceCount 8

2.1.4 Ps2_GetDeviceListInfo 9

2.1.5 Ps2_GetDeviceInfo........ 10

2.1.6 Ps2_OpenDevice 10

2.1.7 Ps2_CloseDevice 11

2.1.8 Ps2_StartStream 12

2.1.9 Ps2_StopStream 12

2.1.10 Ps2_ReadNextFrame 13

2.1.11 Ps2_GetFrame........ 14

2.1.12 Ps2_SetDataMode........ 15

2.1.13 Ps2_GetDataMode........ 15

2.1.14 Ps2_GetDepthRange 16

2.1.15 Ps2_SetDepthRange 17

2.1.16 Ps2_GetThreshold........ 17

2.1.17 Ps2_SetThreshold........ 18

2.1.18 Ps2_GetPulseCount........ 19

2.1.19 Ps2_SetPulseCount........ 19

2.1.20 Ps2_GetGMMGain........ 20

2.1.21 Ps2_SetGMMGain........ 21

2.1.22 Ps2_GetProperty 22

2.1.23 Ps2_SetProperty 22

2.1.24 Ps2_GetCameraParameters 23
2.1.25 Ps2_GetCameraExtrinsicParameters 24
2.1.26 Ps2_SetWDROutputMode........ 25
2.1.27 Ps2_GetWDROutputMode........ 25
2.1.28 Ps2_SetWDRStyle 26
2.1.29 Ps2_GetMeasuringRange 27
2.1.30 Ps2_ConvertWorldToDepth 27
2.1.31 Ps2_ConvertDepthToWorld 28
2.1.32 Ps2_ConvertDepthFrameToWorldVector 29
2.1.33 Ps2_SetSynchronizeEnable 30
2.1.34 Ps2_GetSynchronizeEnable 31
2.1.35 Ps2_SetDepthDistortionCorrectionEnabled 31
2.1.36 Ps2_GetDepthDistortionCorrectionEnabled 32
2.1.37 Ps2_SetIrDistortionCorrectionEnabled 33
2.1.38 Ps2_GetIrDistortionCorrectionEnabled 34
2.1.39 Ps2_SetComputeRealDepthCorrectionEnabled........ 34
2.1.40 Ps2_GetComputeRealDepthCorrectionEnabled........ 35
2.1.41 Ps2_SetSpatialFilterEnabled 36
2.1.42 Ps2_GetSpatialFilterEnabled 36
2.1.43 Ps2_SetTimeFilterEnabled........ 37
2.1.44 Ps2_GetTimeFilterEnabled........ 38
2.1.45 Ps2_SetDepthFrameEnabled 38
2.1.46 Ps2_SetIrFrameEnabled 39
2.1.47 Ps2_SetImageMirror 40
2.1.48 Ps2_SetImageRotation 40
2.1.49 Ps2_SetHotPlugStatusCallback 41
2.1.50 Ps2_SetHotPlugStatusCallback_ 42
2.1.51 Ps2_GetWDRPulseCount 42
2.1.52 Ps2_SetWDRPulseCount 43
2.1.53 Ps2_GetSerialNumber........ 44
2.1.54 Ps2_GetFirmwareVersionNumber 45
2.1.55 Ps2_SetSlaveModeEnabled 45
2.1.56 Ps2_SetTofFrameRate........ 46
2.1.57 Ps2_GetTofFrameRate........ 47
2.1.58 Ps2_SetStandByEnabled........ 47
2.1.59 Ps2_OpenDeviceByAlias 48
2.1.60 Ps2_SetWaitTimeOfReadNextFrame........ 49
2.1.61 Ps2_SetSlaveTrigger 49
2.1.62 Ps2_RebootCamera 50

2.1.63 Ps2_SetLegacyAlgorithmicEnabled 51
2.1.64 Ps2_SetConfidenceFilterEnabled 51
2.1.65 Ps2_GetConfidenceFilterEnabled 52
2.1.66 Ps2_GetConfidenceFilterThreshold 53
2.1.67 Ps2_SeWDRConfidenceFilterThreshold 53
2.1.68 Ps2_GetWDRConfidenceFilterThreshold 54
2.2 DATA TYPE 55
2.2.1 PsDepthRange 55
2.2.2 PsDataMode 55
2.2.3 PsPropertyType 56
2.2.4 PsFrameType 56
2.2.5 PsSensorType 56
2.2.6 PsPixelFormat 57
2.2.7 PsReturnStatus 57
2.2.8 PsWDRTotalRange 58
2.2.9 PsWDRStyle 58
2.2.10 PsVector3f 59
2.2.11 PsDepthVector3 59
2.2.12 PsCameraParameters 60
2.2.13 PsCameraExtrinsicParameters 61
2.2.14 PsFrame 61
2.2.15 PsWDROutputMode 62
2.2.16 PsMeasuringRange 63
2.2.17 PsDeviceInfo 64
2.2.18 PsDataModeList 65
2.2.19 PsDepthRangeList 65
2.2.20 PsFrameReady 65

1. Overview

1.1 Overview

TOF camera is a 3D camera module using TOF (Time of Flight) technology. DC100 SDK currently provides an API applicable to Linux. Developers can obtain high-precision depth image data, gray image data, and point cloud data through the SDK.

1.2 API Invoke Process

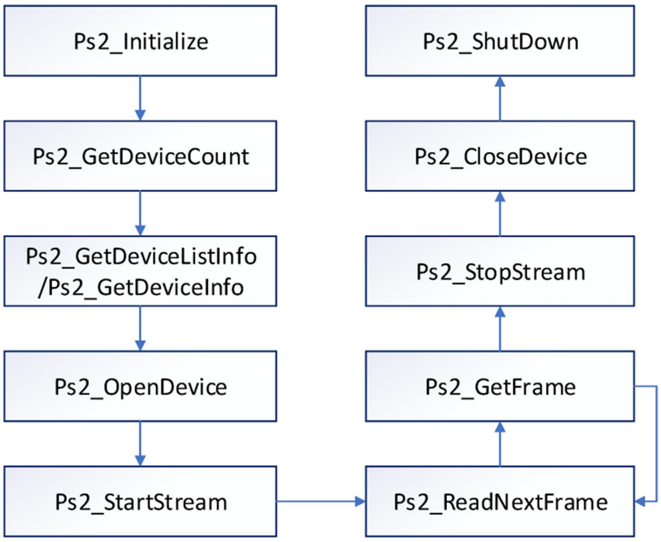

Figure 1-1 API Invoke Process

1.2.1 Ps2_Initialize&Ps2_Shutdown

Call the Ps2_Initializeinterface and initialize the API. Call the Ps2_Shutdown interface finally to log out the API and release all the resources created by the API.

1.2.2 Ps2_GetDeviceCount &Ps2_GetDeviceListInfo/Ps2_GetDeviceInfo

Call the Ps2_GetDeviceCount interface to get the number of devices currently connected. Call the Ps2_GetDeviceListInfo/Ps_GetDeviceInfo interface to get the info of devices currently connected.

1.2.3 Ps2_GetDeviceCount &Ps2_GetDeviceListInfo/Ps2_GetDeviceInfo

Call the Ps2_OpenDevice interface to open the specified depth camera device. Call the Ps2_CloseDevice interface to close the specified device.

1.2.4 Ps2_StartStream&Ps2_StopStream

Call the Ps2_StartStream interface to open the stream of the camera device. Call the Ps2_StopStream interface to close the stream of the camera device.

1.2.5 Ps2_ReadNextFrame&Ps2_GetFrame

In the main loop of image processing, each time Ps2_ReadNextFrame is called first to collect a frame image, and then call Ps2_GetFrame to obtain a frame image data of the specified image type, which is used for corresponding image processing.

1.2.6 Set&Get

The API provides a rich Set and Get type interface for setting and acquiring camera properties, parameters and data. If you need change the camera parameters before call the Ps2_ReadNextFrame, please invoking as below:

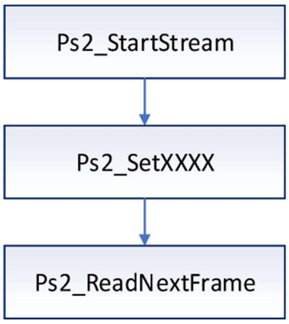

Figure 1-2 API Invoke Process

2. API Introduction

2.1 API Reference

2.1.1 Ps2_Initialize

【Description】

Initialize TOF camera API. It should be called first before calling any other API

【Grammar】

PsReturnStatus Ps2_Initialize();

【Parameter】

No.

【Return value】

Return value	Description
PsRetOK	Success.
Others	Fails. Refer to PsReturnStatus.

【Requirement】

Header file：Vzense_api_550.h
Library file：vzense_api.so

2.1.2 Ps2_Shutdown

【Description】

Shutdown the TOF camera API. It is forbidden to call any other API after the PsShutdown is called.

【Grammar】

PsReturnStatus Ps2_Shutdown();

【Parameter】

No.

【Return value】

Return value	Description
PsRetOK	Success.
Others	Fails. Refer to PsReturnStatus.

【Requirement】

Header file：Vzense_api_550.h
Library file：vzense_api.so

2.1.3 Ps2_GetDeviceCount

【Description】

Get the connected device count.

【Grammar】

PsReturnStatus Ps2_GetDeviceCount(int32_t* pDeviceCount);

【Parameter】

Parameter name	Description	Input/Output
pDeviceCount	The pointer to the variable that need to store the returned device count. It needs to create an int variable first and then pass its pointer to this function.	Output

【Return value】

Return value	Description
PsRetOK	Success.
Others	Fails. Refer to PsReturnStatus.

【Requirement】

Header file：Vzense_api_550.h
Library file：vzense_api.so

2.1.4 Ps2_GetDeviceListInfo

【Description】

Get the info list of devices currently connected.

【Grammar】

PsReturnStatus Ps2_GetDeviceListInfo (PsDeviceInfo* pDevicesList, uint32_t deviceCount);

【Parameter】

Parameter name	Description	Input/Output
deviceCount	The pointer to the variable that need to store the returned device count. It needs to create an int variable first and then pass its pointer to this function.	Output
pDevicesList	The pointer to the variable that need to store the returned devices info.	Input.

【Return value】

Return value	Description
PsRetOK	Success.
Others	Fails. Refer to PsReturnStatus.

【Requirement】

Header file：Vzense_api_550.h
Library file：vzense_api.so

2.1.5 Ps2_GetDeviceInfo

【Description】

Get the info of the device which index is deviceIndex.

【Grammar】

PsReturnStatus Ps2_GetDeviceInfo (PsDeviceInfo* pDevices, uint32_t deviceIndex);

【Parameter】

Parameter name	Description	Input/Output
pDevices	The index of device	Output
deviceIndex	The pointer to the variable that need to store the returned	Input.

【Return value】

Return value	Description
PsRetOK	Success.
Others	Fails. Refer to PsReturnStatus.

【Requirement】

Header file：Vzense_api_550.h
Library file：vzense_api.so

2.1.6 Ps2_OpenDevice

【Description】

Open the specific device indicated by uri and return the device handle.

【Grammar】

PsReturnStatus Ps2_OpenDevice (const char* uri, PsDeviceHandle* pDevice);

【Parameter】

Parameter name	Description	Input/Output
uri	The Identifier of device	Input
pDevice	Thehandle of the device	Output.

【Return value】

Return value	Description
PsRetOK	Success.
Others	Fails. Refer to PsReturnStatus.

【Requirement】

Header file：Vzense_api_550.h
Library file：vzense_api.so

2.1.7 Ps2_CloseDevice

【Description】

Close the specific device indicated by pDevice.

【Grammar】

PsReturnStatus Ps2_CloseDevice (PsDeviceHandle* device);

【Parameter】

Parameter name	Description	Input/Output
device	The device handle	Input

【Return value】

Return value	Description
PsRetOK	Success.
Others	Fails. Refer to PsReturnStatus.

【Requirement】

Header file：Vzense_api_550.h
Library file：vzense_api.so

2.1.8 Ps2_StartStream

【Description】

Start to capture the specific session stream indicated by device and sessionIndex.

【Grammar】

PsReturnStatus Ps2_StartStream (PsDeviceHandle device, uint32_t sessionIndex);

【Parameter】

Parameter name	Description	Input/Output
device	The device handle	Input
sessionIndex	The index of the session that include N TOF sensors and maximum N RGB sensors.range from 0 to SessionCount – 1. See PsDeviceInfo for more information. For example, the camera has 2 TOF sensor and 1 RGB sensor, the SessionCount is 2. If the sessionIndex is 0 mean that start 1 TOF stream and the RGB stream, and if the sessionIndex is 1 mean that start only 1 TOF stream.	Input

【Return value】

Return value	Description
PsRetOK	Success.
Others	Fails. Refer to PsReturnStatus.

【Requirement】

Header file：Vzense_api_550.h
Library file：vzense_api.so

2.1.9 Ps2_StopStream

【Description】

Stop to capture the specific session stream indicated by device and sessionIndex.

【Grammar】

PsReturnStatus Ps2_StopStream (PsDeviceHandle device, uint32_t sessionIndex);

【Parameter】

Parameter name	Description	Input/Output
device	The device handle	Input
sessionIndex	The index of the session that include N TOF sensors and maximum N RGB sensors.range from 0 to SessionCount – 1. See PsDeviceInfo for more information. For example, the camera has 2 TOF sensor and 1 RGB sensor, the SessionCount is 2. If the sessionIndex is 0 mean that start 1 TOF stream and the RGB stream, and if the sessionIndex is 1 mean that start only 1 TOF stream.	Input

【Return value】

Return value	Description
PsRetOK	Success.
Others	Fails. Refer to PsReturnStatus.

【Requirement】

Header file：Vzense_api_550.h
Library file：vzense_api.so

2.1.10 Ps2_ReadNextFrame

【Description】

Capture the next image frame of the specific device. This API should be called first before getting the frame data using Ps2_GetFrame.

【Grammar】

PsReturnStatus Ps2_ReadNextFrame (PsDeviceHandle device, uint32_t sessionIndex, PsFrameReady* pFrameReady);

【Parameter】

Parameter name	Description	Input/Output
device	The device handle	Input
sessionIndex	The session index. see Ps2_StartStream Ps2_StopStream for more info.	Input
pFrameReady	The flg of ready frame, see Ps2_FrameReady for more info.	Output

【Return value】

Return value	Description
PsRetOK	Success.
Others	Fails. Refer to PsReturnStatus.

【Requirement】

Header file：Vzense_api_550.h
Library file：vzense_api.so

2.1.11 Ps2_GetFrame

【Description】

Get the image data of current frame indicated by frame type. It needs to call Ps2_ReadNextFrame to capture one frame of image first before calling this API.

【Grammar】

PsReturnStatus Ps2_GetFrame (PsDeviceHandle device, uint32_t sessionIndex, PsFrameType frameType,
PsFrame* pPsFrame);

【Parameter】

Parameter name	Description	Input/Output
device	The device handle	Input
sessionIndex	The session index. see Ps2_StartStream Ps2_StopStream for more info.	Input
frameType	The frame type.see PsFrameType for more info.	Input
pPsFrame	The pointer of buffer to store the returned image data.	Output

【Return value】

Return value	Description
PsRetOK	Success.
Others	Fails. Refer to PsReturnStatus.

【Requirement】

Header file：Vzense_api_550.h
Library file：vzense_api.so

2.1.12 Ps2_SetDataMode

【Description】

Set the output data mode

【Grammar】

PsReturnStatus Ps2_SetDataMode (PsDeviceHandle device, uint32_t sessionIndex, PsDataMode dataMode);

【Parameter】

Parameter name	Description	Input/Output
device	The device handle	Input
sessionIndex	The session index. see Ps2_StartStream Ps2_StopStream for more info.	Input
dataMode	Output data mode, refer to PsDataMode	Input

【Return value】

Return value	Description
PsRetOK	Success.
Others	Fails. Refer to PsReturnStatus.

【Requirement】

Header file：Vzense_api_550.h
Library file：vzense_api.so

2.1.13 Ps2_GetDataMode

【Description】

Get the output data mode

【Grammar】

PsReturnStatus Ps2_GetDataMode (PsDeviceHandle device, uint32_t sessionIndex, PsDataMode dataMode);

【Parameter】

Parameter name	Description	Input/Output
device	The device handle	Input

sessionIndex	The session index. see Ps2_StartStream Ps2_StopStream for more info.	Input
dataMode	Output data mode, refer to PsDataMode	Input

【Return value】

Return value	Description
PsRetOK	Success.
Others	Fails. Refer to PsReturnStatus.

【Requirement】

Header file：Vzense_api_550.h
Library file：vzense_api.so

2.1.14 Ps2_GetDepthRange

【Description】

Get the depth range mode of the specific device.

【Grammar】

PsReturnStatus Ps2_GetDepthRange (PsDeviceHandle device, uint32_t sessionIndex, PsDepthRange* pDepthRange);

【Parameter】

Parameter name	Description	Input/Output
device	The device handle	Input
sessionIndex	The session index. see Ps2_StartStream Ps2_StopStream for more info.	Input
pDepthRange	The pointer of variable to store the returned depth range mode. Refer to PsDepthRange.	Output

【Return value】

Return value	Description
PsRetOK	Success.
Others	Fails. Refer to PsReturnStatus.

【Requirement】

Header file：Vzense_api_550.h
Library file：vzense_api.so

2.1.15 Ps2_SetDepthRange

【Description】

Set the depth range mode of the specific device.

【Grammar】

PsReturnStatus Ps2_SetDepthRange (PsDeviceHandle device, uint32_t sessionIndex, PsDepthRange pDepthRange);

【Parameter】

Parameter name	Description	Input/Output
device	The device handle	Input
sessionIndex	The session index. see Ps2_StartStream Ps2_StopStream for more info.	Input
pDepthRange	The depth range that needs to set. Refer to PsDepthRange.	Input

【Return value】

Return value	Description
PsRetOK	Success.
Others	Fails. Refer to PsReturnStatus.

【Requirement】

Header file：Vzense_api_550.h
Library file：vzense_api.so

2.1.16 Ps2_GetThreshold

【Description】

Get the threshold value of depth image

【Grammar】

PsReturnStatus Ps2_GetThreshold (PsDeviceHandle device, uint32_t sessionIndex, uint16_t* pThreshold);

【Parameter】

Parameter name	Description	Input/Output
device	The device handle	Input
sessionIndex	The session index. see Ps2_StartStream Ps2_StopStream for more info.	Input
pThreshold	The threshold value	Output

【Return value】

Return value	Description
PsRetOK	Success.
Others	Fails. Refer to PsReturnStatus.

【Requirement】

Header file：Vzense_api_550.h
Library file：vzense_api.so

2.1.17 Ps2_SetThreshold

【Description】

Set the threshold value of depth image

【Grammar】

PsReturnStatus Ps2_SetThreshold (PsDeviceHandle device, uint32_t sessionIndex, uint16_t pThreshold);

【Parameter】

Parameter name	Description	Input/Output
device	The device handle	Input
sessionIndex	The session index. see Ps2_StartStream Ps2_StopStream for more info.	Input
pThreshold	The threshold value	Input

【Return value】

Return value	Description

PsRetOK	Success.
Others	Fails. Refer to PsReturnStatus.

【Requirement】

Header file：Vzense_api_550.h
Library file：vzense_api.so

2.1.18 Ps2_GetPulseCount

【Description】

Get the pulse count of depth image, PulseCount is used for exposure.

【Grammar】

PsReturnStatus Ps2_GetPulseCount (PsDeviceHandle device, uint32_t sessionIndex, uint16_t* pPulseCount);

【Parameter】

Parameter name	Description	Input/Output
device	The device handle	Input
sessionIndex	The session index. see Ps2_StartStream Ps2_StopStream for more info.	Input
pPulseCount	Pointer to the variable that used to store returned pulse count	Output

【Return value】

Return value	Description
PsRetOK	Success.
Others	Fails. Refer to PsReturnStatus.

【Requirement】

Header file：Vzense_api_550.h
Library file：vzense_api.so

2.1.19 Ps2_SetPulseCount

【Description】

Set the pulse count of depth image, PulseCount is used for exposure.

【Grammar】

PsReturnStatus Ps2_SetPulseCount (PsDeviceHandle device, uint32_t sessionIndex, uint16_t pPulseCount);

【Parameter】

Parameter name	Description	Input/Output
device	The device handle	Input
sessionIndex	The session index. see Ps2_StartStream Ps2_StopStream for more info.	Input
pPulseCount	Pointer to the variable that used to store returned pulse count	Input

【Return value】

Return value	Description
PsRetOK	Success.
Others	Fails. Refer to PsReturnStatus.

【Requirement】

Header file：Vzense_api_550.h
Library file：vzense_api.so

2.1.20 Ps2_GetGMMGain

【Description】

Get Gamma gain of depth image.

【Grammar】

PsReturnStatus Ps2_GetGMMGain (PsDeviceHandle device, uint32_t sessionIndex, uint16_t* gmmgain);

【Parameter】

Parameter name	Description	Input/Output
device	The device handle	Input
sessionIndex	The session index. see Ps2_StartStream Ps2_StopStream for more info.	Input

gmmgain	To store the returned Gamma value variable pointer, you need to first create an unsigned short type variable and pass its pointer to the function	Output

【Return value】

Return value	Description
PsRetOK	Success.
Others	Fails. Refer to PsReturnStatus.

【Requirement】

Header file：Vzense_api_550.h
Library file：vzense_api.so

2.1.21 Ps2_SetGMMGain

【Description】

Set Gamma gain of depth image.

【Grammar】

PsReturnStatus Ps2_SetGMMGain (PsDeviceHandle device, uint32_t sessionIndex, uint16_t* gmmgain);

【Parameter】

Parameter name	Description	Input/Output
device	The device handle	Input
sessionIndex	The session index. see Ps2_StartStream Ps2_StopStream for more info.	Input
gmmgain	Gamma gain to be set.	Input

【Return value】

Return value	Description
PsRetOK	Success.
Others	Fails. Refer to PsReturnStatus.

【Requirement】

Header file：Vzense_api_550.h
Library file：vzense_api.so

2.1.22 Ps2_GetProperty

【Description】

Get the property value of the specific device indicated by deviceIndex.

【Grammar】

PsReturnStatus Ps2_GetProperty (PsDeviceHandle device, uint32_t sessionIndex, PsPropertyType propertyType, void* pData, int32_t* pDataSize);

【Parameter】

Parameter name	Description	Input/Output
device	The device handle	Input
sessionIndex	The session index. see Ps2_StartStream Ps2_StopStream for more info.	Input
propertyType	The property type. Refer to PsPropertyType.	Input
pData	The pointer of buffer to store the returned property value.	Output
pDataSize	Pass the buffer size of pData. Also return the actual size of returned property value in byte.	In/Output

【Return value】

Return value	Description
PsRetOK	Success.
Others	Fails. Refer to PsReturnStatus.

【Requirement】

Header file：Vzense_api_550.h
Library file：vzense_api.so

2.1.23 Ps2_SetProperty

【Description】

Set the property value of the specific device.

【Grammar】

PsReturnStatus Ps2_SetProperty (PsDeviceHandle device, uint32_t sessionIndex, PsPropertyType propertyType,
void* pData, int32_t dataSize);

【Parameter】

Parameter name	Description	Input/Output
device	The device handle	Input
sessionIndex	The session index. see Ps2_StartStream Ps2_StopStream for more info.	Input
propertyType	The property type. Refer to PsPropertyType.	Input
pData	The pointer of buffer to store the returned property value.	Input
dataSize	Pass the buffer size of pData. Also return the actual size of returned property value in byte.	Input

【Return value】

Return value	Description
PsRetOK	Success.
Others	Fails. Refer to PsReturnStatus.

【Requirement】

Header file：Vzense_api_550.h
Library file：vzense_api.so

2.1.24 Ps2_GetCameraParameters

【Description】

Get the camera internal parameters.

【Grammar】

PsReturnStatus Ps2_GetCameraParameters (PsDeviceHandle device, uint32_t sessionIndex, PssensorType sensorType,
PsCameraParameters* pCameraParameters);

【Parameter】

Parameter name	Description	Input/Output
device	The device handle	Input
sessionIndex	The session index. see Ps2_StartStream Ps2_StopStream for more info.	Input

sensorType	Type of sensor , 0 indicates the depth camera, 1 indicates the RGB camera	Input
pCameraParameters	Output the camera internal parameters, refer to PsCameraParameters	Output

【Return value】

Return value	Description
PsRetOK	Success.
Others	Fails. Refer to PsReturnStatus.

【Requirement】

Header file：Vzense_api_550.h
Library file：vzense_api.so

2.1.25 Ps2_GetCameraExtrinsicParameters

【Description】

Get camera rotation and transmission coefficient parameters.

【Grammar】

PsReturnStatus Ps2_GetCameraExtrinsicParameters (PsDeviceHandle device, uint32_t sessionIndex, PsCameraExtrinsicParameters* pCameraExtrinsicParameters);

【Parameter】

Parameter name	Description	Input/Output
device	The device handle	Input
sessionIndex	The session index. see Ps2_StartStream Ps2_StopStream for more info.	Input
pCameraExtrinsicParameters	Pointer to the structural variable used to store the returned camera parameters.	Output

【Return value】

Return value	Description
PsRetOK	Success.
Others	Fails. Refer to PsReturnStatus.

【Requirement】

Header file：Vzense_api_550.h
Library file：vzense_api.so

2.1.26 Ps2_SetWDROutputMode

【Description】

Set WDR output mode, refer to PsWDROutputMode.

【Grammar】

PsReturnStatus Ps2_SetWDROutputMode((PsDeviceHandle device, uint32_t sessionIndex,
PsWDROutputMode* pWDRMode);

【Parameter】

Parameter name	Description	Input/Output
device	The device handle	Input
sessionIndex	The session index. see Ps2_StartStream Ps2_StopStream for more info.	Input
pWDRMode	The WDR mode, refer to PsWDROutputMode	Input

【Return value】

Return value	Description
PsRetOK	Success.
Others	Fails. Refer to PsReturnStatus.

【Requirement】

Header file：Vzense_api_550.h
Library file：vzense_api.so

2.1.27 Ps2_GetWDROutputMode

【Description】

Get WDR mode.

【Grammar】

PsReturnStatus Ps2_GetWDROutputMode((PsDeviceHandle device, uint32_t sessionIndex,
PsWDROutputMode* pWDRMode);

【Parameter】

Parameter name	Description	Input/Output
device	The device handle	Input
sessionIndex	The session index. see Ps2_StartStream Ps2_StopStream for more info.	Input
pWDRMode	The WDR mode, refer to PsWDROutputMode	Output

【Return value】

Return value	Description
PsRetOK	Success.
Others	Fails. Refer to PsReturnStatus.

【Requirement】

Header file：Vzense_api_550.h
Library file：vzense_api.so

2.1.28 Ps2_SetWDRStyle

【Description】

Set output style of WDR mode.

【Grammar】

PsReturnStatus Ps2_SetWDRStyle (PsDeviceHandle device, uint32_t sessionIndex, PsWDRStyle wdrStyle)；

【Parameter】

Parameter name	Description	Input/Output
device	The device handle	Input
sessionIndex	The session index. see Ps2_StartStream Ps2_StopStream for more info.	Input
wdrStyle	The output style, in fusion or alternation, refer to PsWDRStyle	Output

【Return value】

Return value	Description
PsRetOK	Success.

Others	Fails. Refer to PsReturnStatus.

【Requirement】

Header file：Vzense_api_550.h
Library file：vzense_api.so

2.1.29 Ps2_GetMeasuringRange

【Description】

Get Measuring Range.

【Grammar】

PsReturnStatus Ps2_GetMeasuringRange (PsDeviceHandle device, uint32_t sessionIndex, PsDepthRange depthRange, PsMeasuringRange pMeasuringRange);

【Parameter】

Parameter name	Description	Input/Output
device	The device handle	Input
sessionIndex	The session index. see Ps2_StartStream Ps2_StopStream for more info.	Input
depthRange	The depth range.	Input
pMeasuringRange	The measuring range, refer to PsMeasuringRange	Output

【Return value】

Return value	Description
PsRetOK	Success.
Others	Fails. Refer to PsReturnStatus.

【Requirement】

Header file：Vzense_api_550.h
Library file：vzense_api.so

2.1.30 Ps2_ConvertWorldToDepth

【Description】

Convert the input points from the World coordinate system to the Depth coordinate system.

【Grammar】

```
PsReturnStatus Ps2_ConvertWorldToDepth (PsDeviceHandle device, uint32_t sessionIndex,
                                        PsVector3f* pWorldVector, PsDepthVector3* pDepthVector,
                                        int32_t pointCount, PsCameraParameters* pCameraParam);
```

【Parameter】

Parameter name	Description	Input/Output
device	The device handle	Input
sessionIndex	The session index. see Ps2_StartStream Ps2_StopStream for more info.	Input
pWorldVector	The pointer to the buffer which stored the x,y,z value of world coordinate of the input points to be converted, measured in millimeters.	Input
pDepthVect	The pointer to the buffer to store the output x,y,z value of depth coordinate. (x,y) is measured in pixels with (0,0) at the top left of the image. z is measured in millimeters, it is the depth value of the point to be converted.	Output
pointCount	The point count to be converted.	Input
pCameraParam	The camera parameters.	Input

【Return value】

Return value	Description
PsRetOK	Success.
Others	Fails. Refer to PsReturnStatus.

【Requirement】

Header file：Vzense_api_550.h
Library file：vzense_api.so

2.1.31 Ps2_ConvertDepthToWorld

【Description】

Convert the input points from the Depth coordinate system to the World coordinate system.

【Grammar】

```
PsReturnStatus Ps2_ ConvertDepthToWorld (PsDeviceHandle device, uint32_t sessionIndex,
                                         PsDepthVector3* pDepthVector, PsVector3f* pWorldVector,
                                         int32_t pointCount, PsCameraParameters* pCameraParam);
```

【Parameter】

Parameter name	Description	Input/Output
device	The device handle	Input
sessionIndex	The session index. see Ps2_StartStream Ps2_StopStream for more info.	Input
pDepthVect	The pointer to the buffer to store the output x,y,z value of depth coordinate. (x,y) is measured in pixels with (0,0) at the top left of the image. z is measured in millimeters, it is the depth value of the point to be converted.	Input
pWorldVector	The pointer to the buffer which stored the x,y,z value of world coordinate of the input points to be converted, measured in millimeters.	Output
pointCount	The point count to be converted.	Input
pCameraParam	The camera parameters.	Input

【Return value】

Return value	Description
PsRetOK	Success.
Others	Fails. Refer to PsReturnStatus.

【Requirement】

Header file：Vzense_api_550.h
Library file：vzense_api.so

2.1.32 Ps2_ConvertDepthFrameToWorldVector

【Description】

Convert all points in depthframe from the Depth coordinate system to the World coordinate system.

【Grammar】

```
PsReturnStatus Ps2_ConvertDepthFrameToWorldVector (PsDeviceHandle device, uint32_t sessionIndex,
                                                   const PsFrame& depthFrame, PsVector3f* pWorldVector);
```

【Parameter】

Parameter name	Description	Input/Output
device	The device handle	Input
sessionIndex	The session index. see Ps2_StartStream Ps2_StopStream for more info.	Input
depthFrame	The depth frame.	Input
pWorldVector	The pointer to the buffer which stored the x,y,z value of world coordinate of the input points to be converted, measured in millimeters.	Output

【Return value】

Return value	Description
PsRetOK	Success.
Others	Fails. Refer to PsReturnStatus.

【Requirement】

Header file：Vzense_api_550.h
Library file：vzense_api.so

2.1.33 Ps2_SetSynchronizeEnable

【Description】

Set whether the output RGB, Depth, IR and other images are synchronized in time

【Grammar】

PsReturnStatus Ps2_SetSynchronizeEnabled (PsDeviceHandle device, uint32_t sessionIndex, bool bEnabled)；

【Parameter】

Parameter name	Description	Input/Output
device	The device handle	Input
sessionIndex	The session index. see Ps2_StartStream Ps2_StopStream for more info.	Input
bEnabled	True is set to synchronize and false is set to asynchronize.	Input

【Return value】

Return value	Description
PsRetOK	Success.
Others	Fails. Refer to PsReturnStatus.

【Requirement】

Header file：Vzense_api_550.h
Library file：vzense_api.so

2.1.34 Ps2_GetSynchronizeEnable

【Description】

Get whether the output RGB, Depth, IR and other images are synchronized in time.

【Grammar】

PsReturnStatus Ps2_GetSynchronizeEnabled (PsDeviceHandle device, uint32_t sessionIndex, bool* bEnabled);

【Parameter】

Parameter name	Description	Input/Output
device	The device handle	Input
sessionIndex	The session index. see Ps2_StartStream Ps2_StopStream for more info.	Input
bEnabled	True is set to synchronize and false is set to asynchronize.	Output

【Return value】

Return value	Description
PsRetOK	Success.
Others	Fails. Refer to PsReturnStatus.

【Requirement】

Header file：Vzense_api_550.h
Library file：vzense_api.so

2.1.35 Ps2_SetDepthDistortionCorrectionEnabled

【Description】

Set to enable or disable the Depth distortion correction feature.

【Grammar】

```
PsReturnStatus Ps2_SetDepthDistortionCorrectionEnabled (PsDeviceHandle device, uint32_t sessionIndex,
                                                         bool bEnabled);
```

【Parameter】

Parameter name	Description	Input/Output
device	The device handle	Input
sessionIndex	The session index. see Ps2_StartStream Ps2_StopStream for more info.	Input
bEnabled	True to enable the feature, false to disable the feature.	Input

【Return value】

Return value	Description
PsRetOK	Success.
Others	Fails. Refer to PsReturnStatus.

【Requirement】

Header file：Vzense_api_550.h
Library file：vzense_api.so

2.1.36 Ps2_GetDepthDistortionCorrectionEnabled

【Description】

Get the Depth distortion correction feature, enable or disable.

【Grammar】

```
PsReturnStatus Ps2_GetDepthDistortionCorrectionEnabled (PsDeviceHandle device, uint32_t sessionIndex,
                                                         bool* bEnabled);
```

【Parameter】

Parameter name	Description	Input/Output
device	The device handle	Input
sessionIndex	The session index. see Ps2_StartStream Ps2_StopStream for more info.	Input

bEnabled	True to enable the feature, false to disable the feature.	Output

【Return value】

Return value	Description
PsRetOK	Success.
Others	Fails. Refer to PsReturnStatus.

【Requirement】

Header file：Vzense_api_550.h
Library file：vzense_api.so

2.1.37 Ps2_SetIrDistortionCorrectionEnabled

【Description】

Set to enable or disable the IR distortion correction feature.

【Grammar】

PsReturnStatus Ps2_SetIrDistortionCorrectionEnabled (PsDeviceHandle device, uint32_t sessionIndex,
bool bEnabled);

【Parameter】

Parameter name	Description	Input/Output
device	The device handle	Input
sessionIndex	The session index. see Ps2_StartStream Ps2_StopStream for more info.	Input
bEnabled	True to enable the feature, false to disable the feature.	Input

【Return value】

Return value	Description
PsRetOK	Success.
Others	Fails. Refer to PsReturnStatus.

【Requirement】

Header file：Vzense_api_550.h
Library file：vzense_api.so

2.1.38 Ps2_GetIrDistortionCorrectionEnabled

【Description】

Get to enable or disable the IR distortion correction feature.

【Grammar】

PsReturnStatus Ps2_SetIrDistortionCorrectionEnabled (PsDeviceHandle device, uint32_t sessionIndex, bool* bEnabled);

【Parameter】

Parameter name	Description	Input/Output
device	The device handle	Input
sessionIndex	The session index. see Ps2_StartStream Ps2_StopStream for more info.	Input
bEnabled	True to enable the feature, false to disable the feature.	Output

【Return value】

Return value	Description
PsRetOK	Success.
Others	Fails. Refer to PsReturnStatus.

【Requirement】

Header file：Vzense_api_550.h
Library file：vzense_api.so

2.1.39 Ps2_SetComputeRealDepthCorrectionEnabled

【Description】

Set to enable or disable the computer real depth correction feature.

【Grammar】

PsReturnStatus Ps2_SetComputeRealDepthCorrectionEnabled (PsDeviceHandle device, uint32_t sessionIndex, bool bEnabled);

【Parameter】

Parameter name	Description	Input/Output
device	The device handle	Input
sessionIndex	The session index. see Ps2_StartStream Ps2_StopStream for more info.	Input
bEnabled	True to enable the feature, false to disable the feature.	Input

【Return value】

Return value	Description
PsRetOK	Success.
Others	Fails. Refer to PsReturnStatus.

【Requirement】

Header file：Vzense_api_550.h
Library file：vzense_api.so

2.1.40 Ps2_GetComputeRealDepthCorrectionEnabled

【Description】

Get to enable or disable the computer real depth correction feature.

【Grammar】

PsReturnStatus Ps2_GetComputeRealDepthCorrectionEnabled (PsDeviceHandle device, uint32_t sessionIndex,
bool* bEnabled);

【Parameter】

Parameter name	Description	Input/Output
device	The device handle	Input
sessionIndex	The session index. see Ps2_StartStream Ps2_StopStream for more info.	Input
bEnabled	True to enable the feature, false to disable the feature.	Input

【Return value】

Return value	Description
PsRetOK	Success.
Others	Fails. Refer to PsReturnStatus.

【Requirement】

Header file：Vzense_api_550.h
Library file：vzense_api.so

2.1.41 Ps2_SetSpatialFilterEnabled

【Description】

Set to enable or disable the Spatial Filter feature.

【Grammar】

PsReturnStatus Ps2_SetSpatialFilterEnabled (PsDeviceHandle device, uint32_t sessionIndex, bool bEnabled);

【Parameter】

Parameter name	Description	Input/Output
device	The device handle	Input
sessionIndex	The session index. see Ps2_StartStream Ps2_StopStream for more info.	Input
bEnabled	True to enable the feature, false to disable the feature.	Input

【Return value】

Return value	Description
PsRetOK	Success.
Others	Fails. Refer to PsReturnStatus.

【Requirement】

Header file：Vzense_api_550.h
Library file：vzense_api.so

2.1.42 Ps2_GetSpatialFilterEnabled

【Description】

Get the Spatial Filter feature,enable or disable.

【Grammar】

PsReturnStatus Ps2_GetSpatialFilterEnabled (PsDeviceHandle device, uint32_t sessionIndex, bool* bEnabled);

【Parameter】

Parameter name	Description	Input/Output
device	The device handle	Input
sessionIndex	The session index. see Ps2_StartStream Ps2_StopStream for more info.	Input
bEnabled	True to enable the feature, false to disable the feature.	Output

【Return value】

Return value	Description
PsRetOK	Success.
Others	Fails. Refer to PsReturnStatus.

【Requirement】

Header file：Vzense_api_550.h
Library file：vzense_api.so

2.1.43 Ps2_SetTimeFilterEnabled

【Description】

Set to enable or disable the Time Filter feature.

【Grammar】

PsReturnStatus Ps2_SetTimeFilterEnabled (PsDeviceHandle device, uint32_t sessionIndex, bool bEnabled);

【Parameter】

Parameter name	Description	Input/Output
device	The device handle	Input
sessionIndex	The session index. see Ps2_StartStream Ps2_StopStream for more info.	Input
bEnabled	True to enable the feature, false to disable the feature.	Input

【Return value】

Return value	Description

PsRetOK	Success.
Others	Fails. Refer to PsReturnStatus.

【Requirement】

Header file：Vzense_api_550.h
Library file：vzense_api.so

2.1.44 Ps2_GetTimeFilterEnabled

【Description】

Get the Time Filter feature,enable or disable.

【Grammar】

PsReturnStatus Ps2_GetTimeFilterEnabled (PsDeviceHandle device, uint32_t sessionIndex, bool* bEnabled);

【Parameter】

Parameter name	Description	Input/Output
device	The device handle	Input
sessionIndex	The session index. see Ps2_StartStream Ps2_StopStream for more info.	Input
bEnabled	True to enable the feature, false to disable the feature.	Output

【Return value】

Return value	Description
PsRetOK	Success.
Others	Fails. Refer to PsReturnStatus.

【Requirement】

Header file：Vzense_api_550.h
Library file：vzense_api.so

2.1.45 Ps2_SetDepthFrameEnabled

【Description】

Enables or disables the Depth stream feature.

【Grammar】

PsReturnStatus Ps2_SetDepthFrameEnabled (PsDeviceHandle device, uint32_t sessionIndex, bool bEnabled)；

【Parameter】

Parameter name	Description	Input/Output
device	The device handle	Input
sessionIndex	The session index. see Ps2_StartStream Ps2_StopStream for more info.	Input
bEnabled	True to enable the feature, false to disable the feature.	Input

【Return value】

Return value	Description
PsRetOK	Success.
Others	Fails. Refer to PsReturnStatus.

【Requirement】

Header file：Vzense_api_550.h
Library file：vzense_api.so

2.1.46 Ps2_SetIrFrameEnabled

【Description】

Enables or disables the Ir stream feature.

【Grammar】

PsReturnStatus Ps2_SetIrFrameEnabled (PsDeviceHandle device, uint32_t sessionIndex, bool bEnabled)；

【Parameter】

Parameter name	Description	Input/Output
device	The device handle	Input
sessionIndex	The session index. see Ps2_StartStream Ps2_StopStream for more info.	Input
bEnabled	True to enable the feature, false to disable the feature.	Input

【Return value】

Return value	Description
PsRetOK	Success.
Others	Fails. Refer to PsReturnStatus.

【Requirement】

Header file：Vzense_api_550.h
Library file：vzense_api.so

2.1.47 Ps2_SetImageMirror

【Description】

Set image mirror.

【Grammar】

PsReturnStatus Ps2_SetImageMirror (PsDeviceHandle device, uint32_t sessionIndex, int32_t type);

【Parameter】

Parameter name	Description	Input/Output
device	The device handle	Input
sessionIndex	The session index. see Ps2_StartStream Ps2_StopStream for more info.	Input
type	1: left-right mirror, 2: up-down mirror, 3: both mirror (rotation 180)	Input

【Return value】

Return value	Description
PsRetOK	Success.
Others	Fails. Refer to PsReturnStatus.

【Requirement】

Header file：Vzense_api_550.h
Library file：vzense_api.so

2.1.48 Ps2_SetImageRotation

【Description】

Set image rotation.

【Grammar】

PsReturnStatus Ps2_SetImageRotation (PsDeviceHandle device, uint32_t sessionIndex, int32_t type);

【Parameter】

Parameter name	Description	Input/Output
device	The device handle	Input
sessionIndex	The session index. see Ps2_StartStream Ps2_StopStream for more info.	Input
type	0: counterclock 906y, 1: counterclock 1806 y , 2: counterclock 2706 y	Input

【Return value】

Return value	Description
PsRetOK	Success.
Others	Fails. Refer to PsReturnStatus.

【Requirement】

Header file：Vzense_api_550.h
Library file：vzense_api.so

2.1.49 Ps2_SetHotPlugStatusCallback

【Description】

Set the callbcak function

【Grammar】

PsReturnStatus Ps2_SetHotPlugStatusCallback (PtrHotPlugStatusCallback pCallback);

【Parameter】

Parameter name	Description	Input/Output
pCallback	The Callback function	Input

【Return value】

Return value	Description
PsRetOK	Success.
Others	Fails. Refer to PsReturnStatus.

【Requirement】

Header file：Vzense_api_550.h
Library file：vzense_api.so

2.1.50 Ps2_SetHotPlugStatusCallback_

【Description】

Set the callbcak function for c++

【Grammar】

PsReturnStatus Ps2_SetHotPlugStatusCallback_ (PtrHotPlugStatusCallback pCallback, void* contex);

【Parameter】

Parameter name	Description	Input/Output
pCallback	The Callback function	Input
contex	Pointer to the object of C++ class	Input

【Return value】

Return value	Description
PsRetOK	Success.
Others	Fails. Refer to PsReturnStatus.

【Requirement】

Header file：Vzense_api_550.h
Library file：vzense_api.so

2.1.51 Ps2_GetWDRPulseCount

【Description】

Get the pulsecount in WDR mode

【Grammar】

PsReturnStatus Ps2_GetWDRPulseCount (PsDeviceHandle device, uint32_t sessionIndex,
PsWDRPulseCount* pwdrPulseCount);

【Parameter】

Parameter name	Description	Input/Output
device	The device handle	Input
sessionIndex	The session index. see Ps2_StartStream Ps2_StopStream for more info.	Input
pwdrPulseCount	The pulsecount value	Output

【Return value】

Return value	Description
PsRetOK	Success.
Others	Fails. Refer to PsReturnStatus.

【Requirement】

Header file：Vzense_api_550.h
Library file：vzense_api.so

2.1.52 Ps2_SetWDRPulseCount

【Description】

Set the pulsecount in WDR mode

【Grammar】

PsReturnStatus Ps2_GetWDRPulseCount (PsDeviceHandle device, uint32_t sessionIndex,
PsWDRPulseCount pwdrPulseCount);

【Parameter】

Parameter name	Description	Input/Output
device	The device handle	Input
sessionIndex	The session index. see Ps2_StartStream Ps2_StopStream for more info.	Input

pwdrPulseCount	The pulsecount value	Output

【Return value】

Return value	Description
PsRetOK	Success.
Others	Fails. Refer to PsReturnStatus.

【Requirement】

Header file：Vzense_api_550.h
Library file：vzense_api.so

2.1.53 Ps2_GetSerialNumber

【Description】

Get the serial number.

【Grammar】

PsReturnStatus Ps2_GetSerialNumber (PsDeviceHandle device, uint32_t sessionIndex, char* sn, int length);

【Parameter】

Parameter name	Description	Input/Output
device	The device handle	Input
sessionIndex	The session index. see Ps2_StartStream Ps2_StopStream for more info.	Input
sn	The fw value	Output
length	The max length is 63.	Input

【Return value】

Return value	Description
PsRetOK	Success.
Others	Fails. Refer to PsReturnStatus.

【Requirement】

Header file：Vzense_api_550.h
Library file：vzense_api.so

2.1.54 Ps2_GetFirmwareVersionNumber

【Description】

Get the firmware version number.

【Grammar】

PsReturnStatus Ps2_GetSerialNumber (PsDeviceHandle device, uint32_t sessionIndex, char* fw, int length);

【Parameter】

Parameter name	Description	Input/Output
device	The device handle	Input
sessionIndex	The session index. see Ps2_StartStream Ps2_StopStream for more info.	Input
fw	The sn value	Output
length	The max length is 63.	Input

【Return value】

Return value	Description
PsRetOK	Success.
Others	Fails. Refer to PsReturnStatus.

【Requirement】

Header file：Vzense_api_550.h
Library file：vzense_api.so

2.1.55 Ps2_SetSlaveModeEnabled

【Description】

Enables or disables the SlaveMode feature.

【Grammar】

PsReturnStatus Ps2_SetSlaveModeEnabled (PsDeviceHandle device, uint32_t sessionIndex, bool bEnabled);

【Parameter】

Parameter name	Description	Input/Output

device	The device handle	Input
sessionIndex	The session index. see Ps2_StartStream Ps2_StopStream for more info.	Input
bEnabled	True to enable the feature, false to disable the feature	Input

【Return value】

Return value	Description
PsRetOK	Success.
Others	Fails. Refer to PsReturnStatus.

【Requirement】

Header file：Vzense_api_550.h
Library file：vzense_api.so

2.1.56 Ps2_SetTofFrameRate

【Description】

Sets the tof frame rate.

【Grammar】

PsReturnStatus Ps2_SetTofFrameRate (PsDeviceHandle device, uint32_t sessionIndex, uint8_t value);

【Parameter】

Parameter name	Description	Input/Output
device	The device handle	Input
sessionIndex	The session index. see Ps2_StartStream Ps2_StopStream for more info.	Input
value	The value of rate,in 3,5,6,10,15,30.	Input

【Return value】

Return value	Description
PsRetOK	Success.
Others	Fails. Refer to PsReturnStatus.

【Requirement】

Header file：Vzense_api_550.h
Library file：vzense_api.so

2.1.57 Ps2_GetTofFrameRate

【Description】

Get the tof frame rate.

【Grammar】

PsReturnStatus Ps2_SetTofFrameRate (PsDeviceHandle device, uint32_t sessionIndex, uint8_t* value);

【Parameter】

Parameter name	Description	Input/Output
device	The device handle	Input
sessionIndex	The session index. see Ps2_StartStream Ps2_StopStream for more info.	Input
value	The value of rate.	Output

【Return value】

Return value	Description
PsRetOK	Success.
Others	Fails. Refer to PsReturnStatus.

【Requirement】

Header file：Vzense_api_550.h
Library file：vzense_api.so

2.1.58 Ps2_SetStandByEnabled

【Description】

Enables or disables the StandBy feature.

【Grammar】

PsReturnStatus Ps2_SetStandByEnabled (PsDeviceHandle device, uint32_t sessionIndex, bool bEnabled);

【Parameter】

Parameter name	Description	Input/Output
device	The device handle	Input
sessionIndex	The session index. see Ps2_StartStream Ps2_StopStream for more info.	Input
bEnabled	True to enable the feature, false to disable the feature.	Input

【Return value】

Return value	Description
PsRetOK	Success.
Others	Fails. Refer to PsReturnStatus.

【Requirement】

Header file：Vzense_api_550.h
Library file：vzense_api.so

2.1.59 Ps2_OpenDeviceByAlias

【Description】

Open the specific device indicated by alias and return the device handle.

【Grammar】

PsReturnStatus Ps2_OpenDeviceByAlias (const char* alias, PsDeviceHandle* device);

【Parameter】

Parameter name	Description	Input/Output
alias	The alias of device.	Input
device	Thehandle of the device.	Output

【Return value】

Return value	Description
PsRetOK	Success.
Others	Fails. Refer to PsReturnStatus.

【Requirement】

Header file：Vzense_api_550.h
Library file：vzense_api.so

2.1.60 Ps2_SetWaitTimeOfReadNextFrame

【Description】

Set the waittime of read next frame.

【Grammar】

PsReturnStatus Ps2_SetWaitTimeOfReadNextFrame (PsDeviceHandle device, uint32_t sessionIndex, uint16_t time);

【Parameter】

Parameter name	Description	Input/Output
device	The device handle	Input
sessionIndex	The session index. see Ps2_StartStream Ps2_StopStream for more info.	Input
time	The waittime	Input

【Return value】

Return value	Description
PsRetOK	Success.
Others	Fails. Refer to PsReturnStatus.

【Requirement】

Header file：Vzense_api_550.h
Library file：vzense_api.so

2.1.61 Ps2_SetSlaveTrigger

【Description】

Trigger frame data once in slave mode.

【Grammar】

PsReturnStatus Ps2_SetSlaveTrigger (PsDeviceHandle device, uint32_t sessionIndex);

【Parameter】

Parameter name	Description	Input/Output
device	The device handle	Input
sessionIndex	The session index. see Ps2_StartStream Ps2_StopStream for more info.	Input

【Return value】

Return value	Description
PsRetOK	Success.
Others	Fails. Refer to PsReturnStatus.

【Requirement】

Header file：Vzense_api_550.h
Library file：vzense_api.so

2.1.62 Ps2_RebootCamera

【Description】

Reboot the Camera.

【Grammar】

PsReturnStatus Ps2_RebootCamera (PsDeviceHandle device, uint32_t sessionIndex);

【Parameter】

Parameter name	Description	Input/Output
device	The device handle	Input
sessionIndex	The session index. see Ps2_StartStream Ps2_StopStream for more info.	Input

【Return value】

Return value	Description
PsRetOK	Success.
Others	Fails. Refer to PsReturnStatus.

【Requirement】

Header file：Vzense_api_550.h

Library file：vzense_api.so

2.1.63 Ps2_SetLegacyAlgorithmicEnabled

【Description】

Set to enable or disable the Legacy Algorithmic feature , default disable.

【Grammar】

PsReturnStatus Ps2_SetLegacyAlgorithmicEnabled (PsDeviceHandle device, uint32_t sessionIndex, bool bEnabled);

【Parameter】

Parameter name	Description	Input/Output
device	The device handle	Input
sessionIndex	The session index. see Ps2_StartStream Ps2_StopStream for more info.	Input
bEnabled	True to enable the feature, false to disable the feature.	Input

【Return value】

Return value	Description
PsRetOK	Success.
Others	Fails. Refer to PsReturnStatus.

【Requirement】

Header file：Vzense_api_550.h
Library file：vzense_api.so

2.1.64 Ps2_SetConfidenceFilterEnabled

【Description】

Set to enable or disable the Confidence Filter feature.

【Grammar】

PsReturnStatus Ps2_SetConfidenceFilterEnabled (PsDeviceHandle device, uint32_t sessionIndex, bool bEnabled);

【Parameter】

Parameter name	Description	Input/Output
device	The device handle	Input
sessionIndex	The session index. see Ps2_StartStream Ps2_StopStream for more info.	Input
bEnabled	True to enable the feature, false to disable the feature.	Input

【Return value】

Return value	Description
PsRetOK	Success.
Others	Fails. Refer to PsReturnStatus.

【Requirement】

Header file：Vzense_api_550.h
Library file：vzense_api.so

2.1.65 Ps2_GetConfidenceFilterEnabled

【Description】

Get the Confidence Filter feature,enable or disable.

【Grammar】

PsReturnStatus Ps2_GetConfidenceFilterEnabled (PsDeviceHandle device, uint32_t sessionIndex, bool* bEnabled);

【Parameter】

Parameter name	Description	Input/Output
device	The device handle	Input
sessionIndex	The session index. see Ps2_StartStream Ps2_StopStream for more info.	Input
bEnabled	True to enable the feature, false to disable the feature.	Output

【Return value】

Return value	Description
PsRetOK	Success.

Others	Fails. Refer to PsReturnStatus.

【Requirement】

Header file：Vzense_api_550.h
Library file：vzense_api.so

2.1.66 Ps2_GetConfidenceFilterThreshold

【Description】

Get the Confidence Filter threshold value of depth image.

【Grammar】

PsReturnStatus Ps2_GetConfidenceFilterThreshold(PsDeviceHandle device, uint32_t sessionIndex, uint16_t* threshold)；

【Parameter】

Parameter name	Description	Input/Output
device	The device handle	Input
sessionIndex	The session index. see Ps2_StartStream Ps2_StopStream for more info.	Input
threshold	The threshold value.	Output

【Return value】

Return value	Description
PsRetOK	Success.
Others	Fails. Refer to PsReturnStatus.

【Requirement】

Header file：Vzense_api_550.h
Library file：vzense_api.so

2.1.67 Ps2_SeWDRConfidenceFilterThreshold

【Description】

Set the WDR Confidence Filter threshold value of depth image.

【Grammar】

PsReturnStatus Ps2_SetWDRConfidenceFilterThreshold (PsDeviceHandle device, uint32_t sessionIndex,

PsWDRConfidenceThreshold wdrconfidencethreshold);

【Parameter】

Parameter name	Description	Input/Output
device	The device handle	Input
sessionIndex	The session index. see Ps2_StartStream Ps2_StopStream for more info.	Input
wdrconfidencethreshold	The threshold value.	Output

【Return value】

Return value	Description
PsRetOK	Success.
Others	Fails. Refer to PsReturnStatus.

【Requirement】

Header file：Vzense_api_550.h
Library file：vzense_api.so

2.1.68 Ps2_GetWDRConfidenceFilterThreshold

【Description】

Get the WDR Confidence Filter threshold value of depth image.

【Grammar】

PsReturnStatus Ps2_GetWDRConfidenceFilterThreshold (PsDeviceHandle device, uint32_t sessionIndex,

PsWDRConfidenceThreshold* wdrconfidencethreshold);

【Parameter】

Parameter name	Description	Input/Output
device	The device handle	Input
sessionIndex	The session index. see Ps2_StartStream Ps2_StopStream for more info.	Input

wdrconfidencethreshold	The threshold value.	Output

【Return value】

Return value	Description
PsRetOK	Success.
Others	Fails. Refer to PsReturnStatus.

【Requirement】

Header file：Vzense_api_550.h
Library file：vzense_api.so

2.2 Data Type

2.2.1 PsDepthRange

【Description】

Depth Range mode

【Definiton】

```
typedef enum {
    PsUnknown = -1,
    PsNearRange = 0,          // Near Range mode, Range0
    PsMidRange,           // Middle Range mode, Range1
    PsFarRange,           // Far Range mode, Range2
    PsXNearRange,         // XNear range mode, Range3
    PsXMidRange,          // XMid range mode, Range4
    PsXFarRange,         // XFar range mode, Range5
    PsXXNearRange,       // XXNear range mode, Range6
    PsXXMidRange,        // XXMiddle range mode, Range7
    PsXXFarRange,        // XXFar range mode, Range8
} PsDepthRange;
```

2.2.2 PsDataMode

【Description】

Data mode setting, determine which frame output from device and frame fps.

【Definiton】

```
typedef struct {
      PsDepth_30 = 0,        //Output both depth frames at 30 fps.
      PsIR_30 = 1,           // Outputs both IR and RGB frames at 30 fps
      PsDepthAndIR_30 = 2,    // Outputs both depth and IR frames at 30 fps.
      PsNoCCD_30 = 4,       // Reserved for internal use.
      PsDepthAndIR_15 = 10,    // Outputs depth and IR frames at 15 fps, alternating between the two.
      PsWDR_Depth = 11,        // WDR (Wide Dynamic Range) depth mode.
} PsDataMode;

Note：Resolution is 640*480.
```

2.2.3 PsPropertyType

【Description】

Specific property type.

【Definiton】

```
typedef struct {
      PsPropertyDataModeList = 9,             // Gets the data mode lists that the device support
      PsPropertyDepthRangeList = 10,          // Gets the depth range lists that the device support
} PsPropertyType;
```

2.2.4 PsFrameType

【Description】

Specific image frame type.

【Definiton】

```
typedef enum {
      PsDepthFrame = 0,              // Depth frame with 16 bits per pixel in millimeters.
      PsIRFrame = 1,                  // IR frame with 16 bits per pixel.
      PsWDRDepthFrame = 9          // WDR depth frame with 16 bits per pixel in millimeters.
} PsFrameType;
```

2.2.5 PsSensorType

【Description】

The camera sensor type.

【Definiton】

```
typedef enum {
      PsDepthSensor = 0x01,           // Depth camera.
} PsSensorType;
```

2.2.6 PsPixelFormat

【Description】

Specific image pixel type.

【Definiton】

```
typedef enum {
      PsPixelFormatDepthMM16,        // Depth image pixel format, 16 bits per pixel in mm.
      PsPixelFormatGray16,           // IR image pixel format, 16 bits per pixel.
      PsPixelFormatGray8,            // Gray image pixel format, 8 bits per pixel.
} PsPixelFormat;
```

2.2.7 PsReturnStatus

【Description】

Return status of API.

【Definiton】

```
typedef enum {
      PsRetOK = 0,     // The function completed successfully.
      PsRetNoDeviceConnected   = -1,     // There is no depth camera connected or the camera has not
                                           been connected correctly. Check the hardware connection
                                           or try unplugging and re-plugging the USB cable.
      PsRetInvalidDeviceIndex     = -2,     // The input device index is invalid.
      PsRetDevicePointerIsNull   = -3,     // The device structure pointer is null.
      PsRetInvalidFrameType     = -4,     // The input frame type is invalid.
      PsRetFramePointerIsNull   = -5,      // The output frame buffer is null.
      PsRetNoPropertyValueGet   = -6,     // Cannot get the value for the specified property.
      PsRetNoPropertyValueSet     = -7,     // Cannot set the value for the specified property.
      PsRetPropertyPointerIsNull   = -8,      // The input property value buffer pointer is null.
      PsRetPropertySizeNotEnough    = -9,     // The input property value buffer size is too small to
                                                store the specified property value.
      PsRetInvalidDepthRange     = -10,      // The input depth range mode is invalid.
```

```
        PsRetReadNextFrameTimeOut   = -11,      // Capture the next image frame time out.
        PsRetInputPointerIsNull   = -12,        // An input pointer parameter is null.
        PsRetCameraNotOpened   = -13,       // The camera has not been opened.
        PsRetInvalidCameraType   = -14,        // The specified type of camera is invalid.
        PsRetInvalidParams   = -15,       // One or more of the parameter values provided are invalid.
        PsRetCurrentVersionNotSupport   = -16,   // This feature is not supported in the current version.
        PsRetUpgradeImgError   = -17,        // There is an error in the upgrade file.
        PsRetUpgradeImgPathTooLong   = -18,      // Upgrade file path length greater than 260.
        PsRetUpgradeCallbackNotSet   = -19,      // Ps2_SetUpgradeStatusCallback is not called.
        PsRetNoAdapterConnected   = -100,       // There is no adapter connected
        PsRetReInitialized   = -101,          // The SDK has been Initialized
        PsRetNoInitialized   = -102,           // The API has not been Initialized
        PsRetCameraOpened   = -103,       // The camera has been opened.
        PsRetCmdError   = -104,          // Set/Get cmd control error
        PsRetCmdSyncTimeOut   = -105,       // Set cmd ok.but time out for the sync return.

        PsRetOthers = -255,         // An unknown error occurred.
    } PsReturnStatus;
```

2.2.8 PsWDRTotalRange

【Description】

Count of ranges alternatively output in WDR mode.

【Definiton】

```
    typedef enum {
        PsWDRTotalRange_Two = 2,         // Two depth ranges.
        PsWDRTotalRange_Three = 3        // Three depth ranges.
    } PsWDRTotalRange;
```

2.2.9 PsWDRStyle

【Description】

WDR style setting used for API PsSetWDRStyle, which determine WDR image output is fusion from multi range (e.g. Near/Far) or output alternatively (e.g. Near/Far/Near/Far...).

【Definiton】

```
typedef enum {
    PsWDR_FUSION = 0,         // WDR image output is fused from multiple ranges.
    PsWDR_ALTERNATION = 1     // WDR image output alternates between depths
                                 (e.g. Near/Far/Near/Far ... ).
} PsWDRStyle;
```

2.2.10 PsVector3f

【Description】

Vector for float data, it is the point value of world axis and the unit is mm.

【Definiton】

```
typedef struct {
    float x, y, z;
} PsVector3f;
```

【Members】

Member name	Description
x	x components of the vector.
y	y components of the vector.
z	z components of the vector.

2.2.11 PsDepthVector3

【Description】

Depth Image Coordination Vector, it the point value of image axis.

【Definiton】

```
typedef struct {
    int   depthX;
    int   depthY;
    PsDepthPixel   depthZ;
} PsDepthVector3;
```

【Members】

Member name	Description

depthX	The x coordinate of the pixel.
depthY	The y coordinate of the pixel.
depthZ	The depth of the pixel, in millimeters.

2.2.12 PsCameraParameters

【Description】

Parameters of camera.

【Definiton】

```
typedef struct {
    double fx；
    double fy；
    double cx；
    double cy；
    double k1；
    double k2；
    double p1；
    double p2；
    double k3；
    double k4；
    double k5；
    double k6；
} PsCameraParameters；
```

【Members】

Member name	Description
fx	Focal length x (pixel)
fy	Focal length y (pixel)
cx	Principal point x (pixel)
cy	Principal point y (pixel)
k1	Radial distortion coefficient, 1st-order
k2	Radial distortion coefficient, 2nd-order
p1	Tangential distortion coefficient
p2	Tangential distortion coefficient
k3	Radial distortion coefficient, 3rd-order

k4	Radial distortion coefficient, 4st-order
k5	Radial distortion coefficient, 5nd-order
k6	Radial distortion coefficient, 6rd-order

2.2.13 PsCameraExtrinsicParameters

【Description】

Camera extrinsic parameters, it is used for mapping between the depth image and rgb image. The formula is:

$$\begin{bmatrix} x_{rgb} \\ y_{rgb} \\ z_{rgb} \end{bmatrix} = \begin{bmatrix} r0 & r1 & r2 \\ r3 & r4 & r5 \\ r6 & r7 & r8 \end{bmatrix} \times \begin{bmatrix} x_{depth} \\ y_{depth} \\ z_{depth} \end{bmatrix} + \begin{bmatrix} t0 \\ t1 \\ t2 \end{bmatrix}$$

【Definiton】

```
typedef struct {
      double rotation[9];
      double translation[3];
} PsDepthVector3;
```

【Members】

Member name	Description
rotation[9]	Orientation stored as an array of 9 double representing a 3x3 rotation matrix.
translation[3]	Location stored as an array of 3 double representing a 3-D translation vector.

2.2.14 PsFrame

【Description】

The image information.

【Definiton】

```
typedef struct {
      uint32_t   frameIndex；
      PsFrameType   frameType；
      PsPixelFormat   pixelFormat；
      uint8_t   imuFrameNo；
      uint8_t*   pFrameData；
      uint32_t   dataLen；
      float   exposureTime；
      PsDepthRange depthRange；
      uint16_t   width；
      uint16_t   height；
      PsTimeStamp   timestamp；
      uint64_t   hardwaretimestamp；
} PsFrame；
```

【Members】

Member name	Description
frameIndex	The index of the frame.
frameType	The type of frame. See ：：PsFrameType for more information.
pixelFormat	The pixel format used by a frame. See ：：PsPixelFormat for more information.
imuFrameNo	Used to synchronize with IMU, in the range of 0 to 255.
pFrameData	A buffer containing the frame’s image data.
dataLen	The length of pFrame, in bytes.
exposureTime	The exposure time, in milliseconds.
depthRange	The depth range mode of the current frame. Used only for depth frames.
width	The width of the frame, in pixels.
height	The height of the frame, in pixels.
timestamp	The timestamp of the frame that decoded.
hardwaretimestamp	The timestamp of the camera.

2.2.15 PsWDROutputMode

【Description】

Parameters of camera.

【Definiton】

```
typedef struct {
      PsWDRTotalRange   totalRange；
      PsDepthRange   range1；
      uint8_t   range1Count；
      PsDepthRange   range2；
      uint8_t   range2Count；
      PsDepthRange range3；
      uint8_t   range3Count；
} PsWDROutputMode；
```

【Members】

Member name	Description
totalRange	The number of ranges supported. Currently only two or three ranges are supported (e.g. Near/Far or Near/Mid/Far).
range1	The first range.
range1Count	The count of successive range1 frames.
range2	The second range.
range2Count	The count of successive range2 frames.
range3	Third range. This range only takes effect when totalRange is set to 3.
range3Count	The count of successive range3 frames. This only takes effect when totalRange is set to 3.

2.2.16 PsMeasuringRange

【Description】

Measuring range of camera.

【Definiton】

```
typedef struct {
      uint8_t   depthMode；
      uint16_t   depthMaxNear；
      uint16_t   depthMaxMid；
      uint16_t   depthMaxFar；
      uint16_t   effectDepthMaxNear；
      uint16_t   effectDepthMaxMid；
      uint16_t   effectDepthMaxFar；
      uint16_t   effectDepthMinNear；
      uint16_t   effectDepthMinMid；
      uint16_t   effectDepthMinFar；
} PsMeasuringRange；
```

【Members】

Member name	Description
depthMode	0(near/mid/far), 1(xnear/xmid/xfar), 2(xxnear/xxmid/xxfar)
depthMaxNear	The max depth value, in near range, in “depthMode”
depthMaxMid	The max depth value, in mid range, in “depthMode”
depthMaxFar	The max depth value, in far range, in “depthMode”
effectDepthMaxNear	The effect max depth value, in near range, in “depthMode”
effectDepthMaxMid	The effect max depth value, in mid range, in “depthMode”
effectDepthMaxFar	The effect max depth value, in far range, in “depthMode”
effectDepthMinNear	The effect min depth value, in near range, in “depthMode”
effectDepthMinMid	The effect min depth value, in mid range, in “depthMode”
effectDepthMinFar	The effect min depth value, in far range, in “depthMode”

2.2.17 PsDeviceInfo

【Description】

The information of device

【Definiton】

```
typedef struct {
      int SessionCount；
      PsDeviceType devicetype；
      char uri[256]；
      char fw[50]；
      PsConnectStatus status；
} PsDeviceInfo；
```

【Members】

Member name	Description
SessionCount	The count of session
devicetype	The type of device
uri	The identification of device
fw	The firmware version
status	The connect status

2.2.18 PsDataModeList

【Description】

The supportive datamode list of camera.

【Definiton】

```
typedef struct {
      uint8_t count;
      uint8_t datamodelist[32];
} PsDataModeList;
```

【Members】

Member name	Description
count	The count of datamode that supported
datamodelist	The list of datamode that supported

2.2.19 PsDepthRangeList

【Description】

The supportive depthrange list of camera.

【Definiton】

```
typedef struct {
      uint8_t count;
      uint8_t depthrangelist[9];
} PsDepthRangeList;
```

【Members】

Member name	Description
count	The count of depthrange that supported.
datamodelist	The list of depthrange that supported.

2.2.20 PsFrameReady

【Description】

The flg of the ready frame.1：available, 0： unavailable.

【Definiton】

```
typedef struct {
      uint32_t depth ： 1；
      uint32_t ir ： 1；
      uint32_t rgb ： 1；
      uint32_t mappedRGB ： 1；
      uint32_t mappedDepth ： 1；
      uint32_t mappedIR ： 1；
      uint32_t confidence ： 1；
      uint32_t wdrDepth ： 1；
      uint32_t reserved ： 24；
} PsFrameReady；
```

【Members】

Member name	Description
depth	flg of the ready Depth frame
ir	flg of the ready IR frame
rgb	flg of the ready RGB frame
mappedRGB	flg of the ready mappedRGB frame
mappedDepth	flg of the ready mappedDepth frame
mappedIR	flg of the ready mappedIR frame
confidence	flg of the ready confidence frame
wdrDepth	flg of the ready wdrdepth frame
reserved	not used