SDK4
for Zelos cameras

Documentation

Kappa opto-electronics GmbH
Kleines Feld 6
37130 Gleichen | Germany
Fon +49.5508.974.0
Fax +49.5508.974.100
info@kappa.de
www.kappa.de

realize visions .

Content

1 Introduction

Concept:
Kappa SDK4 is designed as a Programming Interface for GigEVision cameras of the Kappa Zelos family. SDK4 provides functionality for device access and image acquisition (SDK4Acquire) and for camera controlling (SDK4ZelosControl).

Programming interface:
SDK4 provides a C-Function Interface for Windows and Linux.
SDK4Acquire provides device access and data stream access. It has two different approaches for image buffer management, either by buffers being allocated by the application or by the engine of SDK4Acquire. Image buffers have to be polled, there is no event or callback mechanism.
For controlling functions, SDK4Acquire provides a control handle, which is used by SDK4ZelosControl in order to access its controlling functions.

2 Installation

System Requirements:

Operating System:	Windows XP 32Bit SP3, Windows Vista 32Bit, Windows 7 32/64 Bit, Linux 32/64 Bit (>Kernel 2.6)
CPU:	Dual Core 2 GHz
RAM:	2 GB
Graphics card:	256 MB
Network card:	"Jumbo-Frames" or "MTU-Size" of at least 4 kB.

2.1 Windows

Driver:
For the Windows 32 Bit operating systems, Kappa provides an additional filter driver on top of the driver of the network card, which is called `Kobra Driver`. `Kobra Driver` has to be installed **after** the driver of the network card!
For the installation of `Kobra Driver`, on the Setup CD in the folder `\bin\32bit\KobraManager` you find an installation tool called `KobraManager`.

Binaries:
The Setup CD contains a folder `\lib\` for each 32 bit and 64 bit operating systems. These folders contain static libraries `SDK4Acquire.lib` and `SDK4ZelosControl.lib`.
Furthermore, the Setup CD contains a folder `\bin\` for each 32 bit and 64 bit operating systems. These folders contain the following dynamic link libraries:
`SDK4Acquire.dll`
`SDK4ZelosControl.dll`
`GigEVisionTL.dll`
`Log4cxx.dll`
`Kobra.dll` (only in the 32bit folder)
The folder `\include\SDK4\` contains header files for SDK4Acquire, the folder `\include\SDK4\Control\` contains header files for SDK4ZelosControl.

Samples and Test applications:
The folder `\src\Acquire\VS2008\` contains a VisualStudio2008 solution with a console application sample, which demonstrates the use of SDK4Acquire and SDK4ZelosControl.

The folder `\src\Discovery\` contains `discover.cpp`, a console application sample, which demonstrates the device discovery.
The folder `\src\MultiAcquire\` contains `.cpp` and `.h` files for a console application sample, which demonstrates the use of threads for image acquiring.
The folder `\src\ZelosCtrl\` contains `zelosctrl.cpp`, a console application sample, which demonstrates camera controlling.

The folders `\bin\32bit\` and `\bin\64bit\` contain two test applications: `SDK4ApiTest.exe` in `\Console\` and `GigEVisionViewer.exe` in `\Viewer\`. Before running the viewer, `register.cmd` has to run once, in order to register `GenSourceFilter.ax`.

2.2 Logging

SDK4 provides a facility to log function calls. Therefore, a file named `log4j.xml` is needed in the execution folder. If the file does not exist, no logging is done. An example for this type of file is provided in the folders `\bin\32bit\` and `\bin\64bit\.`
This file contains the logging options, such as log level, log target, log folder and file name.

Log target, log folder and log file name:

<param name="file" value="${APPDATA}\SDK4.log" />

Log level:

<priority value="INFO" /> <!--(DEBUG, INFO or ERROR)-->

Log file properties:

The following listing results in the following: A rolling file appender is used, which writes its data to file named "SDK4.log" in the APPDATA folder, with a max. file size of 1024 kB. 5 backup files are created, older logging data are deleted. The log level can be switched between, DEBUG, INFO and ERROR.

```
<?xml version="1.0" encoding="UTF-8"?>
<log4j:configuration xmlns:log4j="http://jakarta.apache.org/log4j/">
    <appender name="SDK4FileAppender" class="org.apache.log4j.RollingFileAppender">
       <param name="file" value="${APPDATA}\SDK4.log" />
       <param name="append" value="true" />
       <param name="MaxFileSize" value="1024KB"/>
       <param name="MaxBackupIndex" value="5"/>
       <layout class="org.apache.log4j.PatternLayout">
       <param name="ConversionPattern" value="%-5p %d %5rms %-22.22c{1} %-30.30M -
       %m%n" />
       </layout>
    </appender>

    <root>
       <priority value="INFO" />
       <appender-ref ref="SDK4FileAppender"/>
    </root>
 </log4j:configuration>
```

2.3 SDK4Acquire

2.3.1 Device Access Functions

2.3.1.1 SDK4InitLib

This function initializes the library.

```
SDK4_ERROR SDK4InitLib ( void )
```

Return value

Error code:

`SDK4_ERR_SUCCESS (0)` in case of no error
`SDK4_ERR_ERROR (-1001)` in case of unspecified error
one of the other error codes defined in SDK4_ERROR

Example

```
SDK4InitLib();
...
SDK4CloseLib();
```

See also

`SDK4CloseLib`

2.3.1.2 SDK4CloseLib

This function closes the library.

```
SDK4_ERROR SDK4CloseLib ( void )
```

Return value

Error code:

`SDK4_ERR_SUCCESS (0)` in case of no error
`SDK4_ERR_ERROR (-1001)` in case of unspecified error
one of the other error codes defined in SDK4_ERROR

Example

```
SDK4InitLib();
...
SDK4CloseLib();
```

See also

`SDK4InitLib`

2.3.1.3 SDK4GetNumDevices

This function finds the number of GigEVision compatible devices found in the subnet.

```
SDK4_Error SDK4GetNumDevices ( int32_t* piNumDevices )
```

Parameters

`piNumDevices (int32_t*)`

Pointer to the number of devices

Return value

Error code:

`SDK4_ERR_SUCCESS (0)` in case of no error
`SDK4_ERR_ERROR (-1001)` in case of unspecified error
one of the other error codes defined in SDK4_ERROR

Example

```
SDK4InitLib();
int numDevices=0;
SDK4_ERROR err = SDK4GetNumDevices(&numDevices);
SDK4CloseLib();
```

See also

`SDK4InitLib`
`SDK4CloseLib`

2.3.1.4 SDK4GetDeviceID

This function gets the device ID (devicename) corresponing the specified device index.

```
SDK4_Error SDK4GetDeviceID(int32_t iIndex, char * sDeviceName,
          int32_t *piSize  )
```

Parameters

`piSize(int32_t )`

The index of the device

`sDeviceName(char *)`

The character array of the devicename

`piSize(int32_t *)`

The pointer to the size of array

Return value

Error code:

`SDK4_ERR_SUCCESS (0)` in case of no error
`SDK4_ERR_ERROR (-1001)` in case of unspecified error
one of the other error codes defined in SDK4_ERROR

Example

```
SDK4InitLib();
int32_t numDevices=0;
SDK4GetNumDevices(&numDevices);
for(uint32_t i = 0; i < numDevices; i++)
{
     char sName[256];
     int32_t sizeName=sizeof(sName);
     SDK4GetDeviceID(i,sName,&sizeName);
     cout << "[" << i+1 << "] " << sName << endl;
}
...
SDK4CloseLib();
```

See also

SDK4InitLib
SDK4CloseLib
SDK4GetNumDevices

2.3.1.5 SDK4OpenDevice

This function opens the device with the specified devicename and gives the device handle.

```
SDK4_Error SDK4OpenDevice (const char * psName, ENUM_DEVICE_ACCESS
           iOpenFlags, DEV_HANDLE *phDev )
```

Parameters

psName(const char *)
    the device name

iOpenFlags(ENUM_DEVICE_ACCESS)
    The flag of the access mode for opening the device

phDev(DEV_HANDLE *)
    The pointer to the handle of the device

Return value

Error code:

SDK4_ERR_SUCCESS (0) in case of no error

SDK4_ERR_ERROR (-1001) in case of unspecified error

one of the other error codes defined in SDK4_ERROR

Remarks

There are 3 different device access modes, specified in ENUM_DEVICE_ACCESS:

- The access can be “exclusive”, i. e. no other process has access to the camera.
- The access can be “read only”, i. e. the process can monitor the data stream from the camera, but has no access to controlling functions.
- The access can be a “control” access, i. e. the application has controlling access and allows other processes to have monitor access.

Example

```
SDK4InitLib();
int32_t numDevices=0;
SDK4GetNumDevices(&numDevices);
char sName[256];
int32_t sizeName=sizeof(sName);
DEV_HANDLE hDev;
if (numDevices>0)
{
	SDK4GetDeviceID(0,sName,&sizeName);
	SDK4OpenDevice(sName,DEVICEACCESS_CONTROL,&hDev);
	...
	SDK4CloseDevice(hDev);
}
...
SDK4CloseLib();
```

See also

`DEV_HANDLE`
`ENUM_DEVICE_ACCESS`
`SDK4CloseDevice`

2.3.1.6 SDK4CloseDevice

This function closes the device with the specified device handle.

```
SDK4_Error SDK4CloseDevice (DEV_HANDLE hDev )
```

Parameters

`hDev(DEV_HANDLE)`

the handle of the device

Return value

Error code:

`SDK4_ERR_SUCCESS (0)` in case of no error
`SDK4_ERR_ERROR (-1001)` in case of unspecified error
one of the other error codes defined in SDK4_ERROR

Example

```
SDK4InitLib();
int32_t numDevices=0;
SDK4GetNumDevices(&numDevices);
char sName[256];
int32_t sizeName=sizeof(sName);
DEV_HANDLE hDev;
if (numDevices>0)
{
	SDK4GetDeviceID(0,sName,&sizeName);
	SDK4OpenDevice(sName,DEVICEACCESS_CONTROL,&hDev);
	...
	SDK4CloseDevice(hDev);
}
...
SDK4CloseLib();
```

See also

`DEV_HANDLE`
`ENUM_DEVICE_ACCESS`
`SDK4OpenDevice`

2.3.1.7 SDK4DevStartAcquisition

This function starts the image acquisition of the device specified by its device handle.

```
SDK4_Error SDK4DevStartAcquisition(DEV_HANDLE hDev)
```

Parameters

`hDev(DEV_HANDLE)`
the handle of the device

Return value

Error code:
`SDK4_ERR_SUCCESS (0)` in case of no error
`SDK4_ERR_ERROR (-1001)` in case of unspecified error
one of the other error codes defined in SDK4_ERROR

Remarks

This function is only accessible by processes which have control or exclusive device access.

Example

```
SDK4InitLib();
DEV_HANDLE hDev;
...
SDK4DevStartAcquisition (hDev);
SDK4DevStopAcquisition (hDev);
...
SDK4CloseLib();
```

See also

`DEV_HANDLE`
`SDK4DevStopAcquisition`
`ENUM_DEVICE_ACCESS`

2.3.1.8 SDK4DevStopAcquisition

This function stops the image acquisition of the device specified by its device handle.

```
SDK4_Error SDK4DevStopAcquisition (DEV_HANDLE hDev)
```

Parameters

`hDev(DEV_HANDLE)`

  the handle of the device

Return value

Error code:

  `SDK4_ERR_SUCCESS (0)` in case of no error
  `SDK4_ERR_ERROR (-1001)` in case of unspecified error
  one of the other error codes defined in `SDK4_ERROR`

Remarks

This function is only accessible by processes which have control or exclusive device access.

Example

```
SDK4InitLib();
DEV_HANDLE hDev;
...
SDK4DevStartAcquisition (hDev);
SDK4DevStopAcquisition (hDev);
...
SDK4CloseLib();
```

See also

`DEV_HANDLE`
`SDK4DevStartAcquisition`
`ENUM_DEVICE_ACCESS`

2.3.1.9 SDK4DevGetControl

This function gives the control handle of the device, specified by its device handle.

```
SDK4_Error SDK4DevGetControl(DEV_HANDLE hDev, CTRL_HANDLE* phCtrl)
```

Parameters

`hDev(DEV_HANDLE)`

  the handle of the device

`phCtrl(CTRL_HANDLE*)`

  The pointer to the control handle of the device

Return value

Error code:

  `SDK4_ERR_SUCCESS (0)` in case of no error
  `SDK4_ERR_ERROR (-1001)` in case of unspecified error
  one of the other error codes defined in SDK4_ERROR

Remarks

This function is only accessible by processes which have control or exclusive device access.

Example

```
DEV_HANDLE hDev;
char sName[256];
int32_t sizeName=sizeof(sName);
SDK4GetDeviceID(0,sName,&sizeName);
DEV_HANDLE hDev;
SDK4OpenDevice(sName,DEVICEACCESS_CONTROL,&hDev);
CTRL_HANDLE hCtrl;
SDK4DevGetControl(hDev, &hCtrl);
SDK4CloseDevice(hDev);
```

See also

DEV_HANDLE
CTRL_HANDLE
SDK4OpenDevice
ENUM_DEVICE_ACCESS

2.3.1.10 SDK4DevGetDataStream

This function gives the handle to the data stream of the device, specified by its device handle.

```
SDK4_Error SDK4DevGetDataStream(DEV_HANDLE hDev, DS_HANDLE*
           phDataStream )
```

Parameters

hDev(DEV_HANDLE)

the handle of the device

phDataStream (DEV_HANDLE *)

pointer to the data stream handle of the device

Return value

Error code:

SDK4_ERR_SUCCESS (0) in case of no error

SDK4_ERR_ERROR (-1001) in case of unspecified error

one of the other error codes defined in SDK4_ERROR

Remarks

This function is accessible in all device access modes.

Example

```
DEV_HANDLE hDev;
char sName[256];
int32_t sizeName=sizeof(sName);
SDK4GetDeviceID(0,sName,&sizeName);
DEV_HANDLE hDev;
SDK4OpenDevice(sName,DEVICEACCESS_CONTROL,&hDev);
DS_HANDLE hDataStream;
SDK4DevGetDataStream(hDev, &hDataStream);
SDK4CloseDevice(hDev);
```

See also
`DEV_HANDLE`
`DS_HANDLE`
`SDK4OpenDevice`

2.3.1.11 SDK4DevGetClassID

This function gets the device class ID of the device (device class name).

```
SDK4_Error SDK4DevGetClassID(DEV_HANDLE hDev,char * sID, int32_t
           *piSize)
```

Parameters
`hDev(DEV_HANDLE)`
    the handle of the device
`sID(char * )`
    The character array of the class ID
`piSize(int32_t *)`
    The pointer to the maximal size of the array

Return value
Error code:
    SDK4_ERR_SUCCESS (0) in case of no error
    SDK4_ERR_ERROR (-1001) in case of unspecified error
    one of the other error codes defined in SDK4_ERROR

Remarks
The result of this function is "Zelos" in case of a Kappa Zelos camera.

Example
```
DEV_HANDLE hDev;
char sName[256];
int32_t sizeName=sizeof(sName);
SDK4GetDeviceID(0,sName,&sizeName);
DEV_HANDLE hDev;
SDK4OpenDevice(sName,DEVICEACCESS_CONTROL,&hDev);
char sClassId[256];
int32_t sizeClassId=sizeof(sClassId);
SDK4DevGetClassID(hDev, sClassId, &sizeClassId);
SDK4CloseDevice(hDev);
```

See also
`DEV_HANDLE`
`SDK4OpenDevice`

2.3.1.12 SDK4DevGetModel

This function gets the model of the device.

```
SDK4_Error SDK4DevGetModel(DEV_HANDLE hDev,char * sModel, int32_t
           *piSize)
```

Parameters

`hDev(DEV_HANDLE)`

the handle of the device

`sModel (char * )`

The character array of the model name

`piSize(int32_t *)`

The pointer to the maximal size of the array

Return value

Error code:

`SDK4_ERR_SUCCESS (0)` in case of no error

`SDK4_ERR_ERROR (-1001)` in case of unspecified error

one of the other error codes defined in SDK4_ERROR

Remarks

The result of this function is the specific model of Zelos camera, for example "Zelos 02150M".

Example

```
DEV_HANDLE hDev;
char sName[256];
int32_t sizeName=sizeof(sName);
SDK4GetDeviceID(0,sName,&sizeName);
DEV_HANDLE hDev;
SDK4OpenDevice(sName,DEVICEACCESS_CONTROL,&hDev);
static char sModel[256];
int32_t sizeModel=sizeof(sModel);
SDK4DevGetModel(hDev, sModel, &sizeModel);
SDK4CloseDevice(hDev);
```

See also

`DEV_HANDLE`

`SDK4OpenDevice`

2.3.2 Data Stream Functions

2.3.2.1 SDK4DSGetPayloadSize

This function gets the payload size of the image frames.

```
SDK4_Error SDK4DSGetPayloadSize (DS_HANDLE hDataStream, uint32_t*
          piSize)
```

Parameters

`hDataStream (DS_HANDLE)`

    The data stream handle

`piSize (uint32_t*)`

    The pointer to the payload size

Return value

Error code:

```
     SDK4_ERR_SUCCESS (0) in case of no error
     SDK4_ERR_ERROR (-1001) in case of unspecified error
     one of the other error codes defined in SDK4_ERROR
```

Remarks

This function is useful to determine the buffer size, which is needed for data stream buffer allocation and announcement.

Example

```
DEV_HANDLE hDev;
...
DS_HANDLE hDataStream;
SDK4DevGetDataStream(hDev, &hDataStream);
int32_t iSize=0;
SDK4DSGetPayloadSize(hDataStream, &iSize);
```

See also

`DEV_HANDLE`
`DS_HANDLE`
`SDK4DevGetDataStream`
`SDK4DSAllocAndAnnounceBuffer`
`SDK4DSAnnounceBuffer`

2.3.2.2 SDK4DSAnnounceBuffer

This function announces an image buffer to the stream engine, which was allocated by the application.

```
SDK4_Error SDK4DSAnnounceBuffer (DS_HANDLE hDataStream,
           void *pBuffer, size_t iSize, void *pPrivate,
           BUFFER_HANDLE *phBuffer)
```

Parameters

hDataStream (DS_HANDLE)

The data stream handle

pBuffer(void *)

The pointer to the image buffer

iSize(size_t)

The size of the private buffer

pPrivate(void *)

The pointer to a buffer, which contains private data, which is linked to the image buffer

phBuffer(BUFFER_HANDLE *)

The pointer to the buffer handle, which is returned

Return value

Error code:

SDK4_ERR_SUCCESS (0) in case of no error

SDK4_ERR_ERROR (-1001) in case of unspecified error

one of the other error codes defined in SDK4_ERROR

Example

```
DS_HANDLE hDataStream;
...
BUFFER_HANDLE * hBuffer;
nBuffers=3;

SDK4DSGetPayloadSize(hDataStream,&nPayloadSize);
hBuffer = (BUFFER_HANDLE*) malloc(nBuffers*sizeof(BUFFER_HANDLE));
for(int i=0;i<nBuffers;i++)
{
     SDK4DSAnnounceBuffer(hDataStream,
          new uint8_t[nPayloadSize],nPayloadSize,
          new uint8_t[4],&hBuffer[i]);
     SDK4DSQueueBuffer(hDataStream,hBuffer[i]);
}
```

See also

DEV_HANDLE
DS_HANDLE
SDK4DSAllocAndAnnounceBuffer
SDK4DSQueueBuffer
SDK4DSRevokeBuffer

2.3.2.3 SDK4DSAllocAndAnnounceBuffer

This function announces an image buffer to the stream engine, which is to be allocated by the engine.

```
SDK4_Error SDK4DSAllocAndAnnounceBuffer(DS_HANDLE hDataStream,
           size_t iSize, void *pPrivate, BUFFER_HANDLE *phBuffer )
```

Parameters

hDataStream (DS_HANDLE)
    The data stream handle
pBuffer(void *)
    The pointer to the image buffer
iSize(size_t)
    The size of the private buffer
pPrivate(void *)
    The pointer to a buffer, which contains private data, which is linked to the image buffer
phBuffer(BUFFER_HANDLE *)
    The pointer to the buffer handle, which is returned

Return value

Error code:

    SDK4_ERR_SUCCESS (0) in case of no error
    SDK4_ERR_ERROR (-1001) in case of unspecified error
    one of the other error codes defined in SDK4_ERROR

Example

```
DS_HANDLE hDataStream;
...
BUFFER_HANDLE * hBuffer;
nBuffers=3;

SDK4DSGetPayloadSize(hDataStream,&nPayloadSize);
hBuffer = (BUFFER_HANDLE*) malloc(nBuffers*sizeof(BUFFER_HANDLE));
for(int i=0;i<nBuffers;i++)
{
     SDK4DSAllocAndAnnounceBuffer(
          hDataStream,nPayloadSize,NULL,&hBuffer[i]);
     SDK4DSQueueBuffer(hDataStream,hBuffer[i]);
}
```

See also

DEV_HANDLE
DS_HANDLE
SDK4DSAnnounceBuffer
SDK4DSQueueBuffer
SDK4DSRevokeBuffer

2.3.2.4 SDK4DSQueueBuffer

This function gives an image buffer back to the queue, when it is not used anymore.

```
SDK4_Error SDK4DSQueueBuffer(DS_HANDLE hDataStream, BUFFER_HANDLE
           hBuffer)
```

Parameters

```
hDataStream (DS_HANDLE)
     The data stream handle
hBuffer(BUFFER_HANDLE)
     the buffer handle
```

Return value

Error code:

`SDK4_ERR_SUCCESS (0)` in case of no error
`SDK4_ERR_ERROR (-1001)` in case of unspecified error
one of the other error codes defined in `SDK4_ERROR`

Example

```
DS_HANDLE hDataStream;
...
SDK4DevStartAcquisition(hDev);
BUFFER_HANDLE hTempBuffer;
void* pBuffer;
uint32_t size;
int32_t bIsComplete=0;
SDK4DSWaitForBuffer(hDataStream,&hTempBuffer,nTimeout);
SDK4BufferGetPtr(hTempBuffer,&pBuffer);
SDK4BufferGetSize(hTempBuffer,&size);
SDK4BufferIsComplete(hTempBuffer,&bIsComplete);
ASSERT(1,bIsComplete);
if (SDK4BufferSizeFilled(hTempBuffer,&sizeFilled)==SDK4_ERR_SUCCESS)
{
     SDK4DSQueueBuffer(hDataStream,hTempBuffer);
}
SDK4DevStopAcquisition(hDev);
```

See also

```
DS_HANDLE
BUFFER_HANDLE
SDK4DSAllocAndAnnounceBuffer
SDK4DSAnnounceBuffer
SDK4DSRevokeBuffer
```

2.3.2.5 SDK4DSWaitForBuffer

This function waits for an image buffer to be filled. If the function call was successful, the pointer to a buffer handle is returned.

```
SDK4_Error SDK4DSWaitForBuffer(DS_HANDLE hDataStream, BUFFER_HANDLE*
          phBuffer, uint32_t timeout)
```

Parameters

`hDataStream (DS_HANDLE)`

The data stream handle

`phBuffer(BUFFER_HANDLE* )`

The pointer to the buffer handle, which is returned

`Timeout(uint32_t )`

The timeout in milliseconds

Return value

Error code:

`SDK4_ERR_SUCCESS (0)` in case of no error

`SDK4_ERR_ERROR (-1001)` in case of unspecified error

one of the other error codes defined in `SDK4_ERROR`

Example

```
DS_HANDLE hDataStream;
...
SDK4DevStartAcquisition(hDev);
BUFFER_HANDLE hTempBuffer;
void* pBuffer;
uint32_t size;
int32_t bIsComplete=0;
SDK4DSWaitForBuffer(hDataStream,&hTempBuffer,nTimeout);
...
SDK4DSQueueBuffer(hDataStream,hTempBuffer);
...
SDK4DevStopAcquisition(hDev);
```

See also

`DS_HANDLE`
`BUFFER_HANDLE`
`SDK4DSAllocAndAnnounceBuffer`
`SDK4DSAnnounceBuffer`
`SDK4DSQueueBuffer`

2.3.2.6 SDK4DSFlushQueue

This function removes all image buffers from the image queue.

```
SDK4_Error SDK4DSFlushQueue(DS_HANDLE hDataStream)
```

Parameters

`hDataStream (DS_HANDLE)`

The data stream handle

Return value

Error code:

`SDK4_ERR_SUCCESS (0)` in case of no error

`SDK4_ERR_ERROR (-1001)` in case of unspecified error

one of the other error codes defined in `SDK4_ERROR`

Example

```
for(int i=0;i<nBuffers;i++)
{
      SDK4DSAnnounceBuffer(hDataStream,
            new uint8_t[nPayloadSize],nPayloadSize,
            new uint8_t[4],&hBuffer[i]);
      SDK4DSQueueBuffer(hDataStream,hBuffer[i]);
}

SDK4DevStartAcquisition(hDev);
BUFFER_HANDLE hTempBuffer;
void* pBuffer;
uint32_t size;
int32_t bIsComplete=0;
SDK4DSWaitForBuffer(hDataStream,&hTempBuffer,nTimeout);
...
SDK4DSQueueBuffer(hDataStream,hTempBuffer);
SDK4DevStopAcquisition(hDev);
SDK4DSFlushQueue(hDataStream);

for(int i=0;i<nBuffers;i++)
{
      void* pBuffer=NULL;
      void* pPrivate=NULL;

      SDK4DSRevokeBuffer(hDataStream,hBuffer[i],&pBuffer,&pPrivate);
      if (pBuffer!=NULL)
            delete pBuffer;
      if (pPrivate!=NULL)
            delete pPrivate;
}
```

See also

`DS_HANDLE`
`SDK4DSAllocAndAnnounceBuffer`
`SDK4DSAnnounceBuffer`
`SDK4DSQueueBuffer`

2.3.2.7 SDK4DSRevokeBuffer

This function deletes an image buffer, which has been announced previously. When the image format has been changed, this has to be done, before new buffer(s) with a changed size is (are) going to be announced.

```
SDK4_Error SDK4DSRevokeBuffer(DS_HANDLE hDataStream, BUFFER_HANDLE
           hBuffer, void **ppBuffer, void **ppPrivate)
```

Parameters

`hDataStream (DS_HANDLE)`

The data stream handle

`hBuffer(BUFFER_HANDLE )`

The buffer handle

`ppBuffer(void **)`

The pointer to the image buffer

`ppPrivate(void **)`

The pointer to the private buffer

Return value

Error code:

`SDK4_ERR_SUCCESS (0)` in case of no error

`SDK4_ERR_ERROR (-1001)` in case of unspecified error

one of the other error codes defined in `SDK4_ERROR`

Example

```
for(int i=0;i<nBuffers;i++)
{
     SDK4DSAnnounceBuffer(hDataStream,
          new uint8_t[nPayloadSize],nPayloadSize,
          new uint8_t[4],&hBuffer[i]);
     SDK4DSQueueBuffer(hDataStream,hBuffer[i]);
}

SDK4DevStartAcquisition(hDev);
BUFFER_HANDLE hTempBuffer;
void* pBuffer;
uint32_t size;
int32_t bIsComplete=0;
SDK4DSWaitForBuffer(hDataStream,&hTempBuffer,nTimeout);
...
SDK4DSQueueBuffer(hDataStream,hTempBuffer);
SDK4DevStopAcquisition(hDev);
SDK4DSFlushQueue(hDataStream);

for(int i=0;i<nBuffers;i++)
{
     void* pBuffer=NULL;
     void* pPrivate=NULL;

     SDK4DSRevokeBuffer(hDataStream,hBuffer[i],&pBuffer,&pPrivate);
     if (pBuffer!=NULL)
          delete pBuffer;
     if (pPrivate!=NULL)
          delete pPrivate;
}
```

See also

```
DS_HANDLE
SDK4DSAllocAndAnnounceBuffer
SDK4DSAnnounceBuffer
SDK4DSQueueBuffer
```

2.3.3 Image Format Function

2.3.3.1 SDK4DSGetWidth

This function returns the image width.

```
SDK4_Error SDK4DSGetWidth(DS_HANDLE hDataStream, int32_t* pWidth)
```

Parameters

`hDataStream (DS_HANDLE)`

```
     The data stream handle
```

`pWidth(int32_t*)`

The pointer to the image width

Return value

Error code:

`SDK4_ERR_SUCCESS (0)` in case of no error

`SDK4_ERR_ERROR (-1001)` in case of unspecified error

one of the other error codes defined in `SDK4_ERROR`

Example

```
DEV_HANDLE hDev;
...
DS_HANDLE hDataStream;
SDK4DevGetDataStream(hDev, &hDataStream);
int32_t iWidth;
SDK4DSGetWidth(DS_HANDLE hDataStream, &iWidth);
```

See also

```
DS_HANDLE
SDK4DevGetDataStream
```

2.3.3.2 SDK4DSGetHeight

This function returns the image height.

```
SDK4_Error SDK4DSGetHeight(DS_HANDLE hDataStream, int32_t* pHeight)
```

Parameters

`hDataStream (DS_HANDLE)`

The data stream handle

`pHeight(int32_t*)`

The pointer to the image height

Return value

Error code:

`SDK4_ERR_SUCCESS (0)` in case of no error
`SDK4_ERR_ERROR (-1001)` in case of unspecified error
one of the other error codes defined in `SDK4_ERROR`

Example

```
DEV_HANDLE hDev;
...
DS_HANDLE hDataStream;
SDK4DevGetDataStream(hDev, &hDataStream);
int32_t iHeight;
SDK4DSGetHeight(DS_HANDLE hDataStream, &iHeight);
```

See also

`DS_HANDLE`
`SDK4DevGetDataStream`

2.3.3.3 SDK4DSGetPixelFormat

This function returns the pixelformat of the image.

```
SDK4_Error SDK4DSGetPixelFormat(DS_HANDLE hDataStream,
          ENUM_PIXELFORMAT* pFormat)
```

Parameters

`hDataStream (DS_HANDLE)`
The data stream handle
`pFormat (ENUM_PIXELFORMAT*)`
The pointer to the pixelformat

Return value

Error code:

`SDK4_ERR_SUCCESS (0)` in case of no error
`SDK4_ERR_ERROR (-1001)` in case of unspecified error
one of the other error codes defined in `SDK4_ERROR`

Example

```
DEV_HANDLE hDev;
...
DS_HANDLE hDataStream;
SDK4DevGetDataStream(hDev, &hDataStream);
ENUM_PIXELFORMAT iFormat;
SDK4DSGetPixelFormat(DS_HANDLE hDataStream, &iFormat);
```

See also

`DS_HANDLE`
`SDK4DevGetDataStream`
`ENUM_PIXELFORMAT`

2.3.3.4 SDK4DSGetTimePerFrame

This function returns the time per frame of the data stream.

```
SDK4_Error SDK4DSGetTimePerFrame(DS_HANDLE hDataStream, int32_t*
           pTimePerFrame)
```

Parameters

`hDataStream (DS_HANDLE)`

The data stream handle

`pTimePerFrame (int32_t*)`

The pointer to the "time per frame" variable

Return value

Error code:

`SDK4_ERR_SUCCESS (0)` in case of no error

`SDK4_ERR_ERROR (-1001)` in case of unspecified error

one of the other error codes defined in `SDK4_ERROR`

Remarks

The unit of time per frame is 100 ns.

Example

```
DEV_HANDLE hDev;
...
DS_HANDLE hDataStream;
SDK4DevGetDataStream(hDev, &hDataStream);
int32_t iTimePerFrame;
SDK4DSGetTimePerFrame (DS_HANDLE hDataStream, &iTimePerFrame);
```

See also

`DS_HANDLE`

`SDK4DevGetDataStream`

2.3.4 Buffer Functions

2.3.4.1 SDK4BufferGetPtr

This function returns the pointer to the image buffer related to the buffer handle.

```
SDK4_Error SDK4BufferGetPtr(BUFFER_HANDLE hBuffer ,void** ppPtr)
```

Parameters

`hBuffer(BUFFER_HANDLE )`

The buffer handle

`ppPtr(void** )`

The pointer to the image buffer

Return value

Error code:

`SDK4_ERR_SUCCESS (0)` in case of no error
`SDK4_ERR_ERROR (-1001)` in case of unspecified error
one of the other error codes defined in `SDK4_ERROR`

Example

```
...
BUFFER_HANDLE hBuffer;
void* pBuffer;
SDK4DSWaitForBuffer(hDataStream,&hBuffer,nTimeout);
SDK4BufferGetPtr(hBuffer,&pBuffer);
```

See also

`BUFFER_HANDLE`
`SDK4DSWaitForBuffer`

2.3.4.2 SDK4BufferGetPrivatePtr

This function returns the pointer to the private buffer related to the buffer handle.

```
SDK4_Error SDK4BufferGetPrivatePtr(BUFFER_HANDLE hBuffer ,void**
          ppPrivatePtr)
```

Parameters

`hBuffer(BUFFER_HANDLE )`

The buffer handle

`ppPrivatePtr (void** )`

The pointer to the private buffer

Return value

Error code:

`SDK4_ERR_SUCCESS (0)` in case of no error
`SDK4_ERR_ERROR (-1001)` in case of unspecified error
one of the other error codes defined in `SDK4_ERROR`

Example

```
...
BUFFER_HANDLE hBuffer;
void* pPrivatePtr;
SDK4DSWaitForBuffer(hDataStream,&hBuffer,nTimeout);
SDK4BufferGetPrivatePtr(hBuffer,&pPrivatePtr);
```

See also

`BUFFER_HANDLE`
`SDK4DSWaitForBuffer`

2.3.4.3 SDK4BufferGetSize

This function returns the size of the image buffer related to the buffer handle.

```
SDK4_Error SDK4BufferGetSize(BUFFER_HANDLE hBuffer ,uint32_t* pSize)
```

Parameters

`hBuffer(BUFFER_HANDLE )`
The buffer handle

`pSize (uint32_t * )`
The pointer to the buffer size

Return value

Error code:

`SDK4_ERR_SUCCESS (0)` in case of no error
`SDK4_ERR_ERROR (-1001)` in case of unspecified error
one of the other error codes defined in `SDK4_ERROR`

Example

```
...
BUFFER_HANDLE hBuffer;
uint32_t size;
SDK4DSWaitForBuffer(hDataStream,&hBuffer,nTimeout);
SDK4BufferGetSize(hBuffer,&size);
```

See also

`BUFFER_HANDLE`
`SDK4DSWaitForBuffer`

2.3.4.4 SDK4BufferIsComplete

This function returns 1, if the buffer was transmitted completely, and 0, if the transmission was incomplete, i. e. there were packets lost.

```
SDK4_Error SDK4BufferIsComplete(BUFFER_HANDLE hBuffer ,int32_t*
          pIsComplete)
```

Parameters

hBuffer(BUFFER_HANDLE)
The buffer handle
pIsComplete (uint32_t *)
The pointer to the "transmission complete" variable

Return value

Error code:
SDK4_ERR_SUCCESS (0) in case of no error
SDK4_ERR_ERROR (-1001) in case of unspecified error
one of the other error codes defined in SDK4_ERROR

Example

```
...
BUFFER_HANDLE hBuffer;
void* pBuffer;
int32_t bIsComplete=0;
SDK4DSWaitForBuffer(hDataStream,& hBuffer,nTimeout);
SDK4BufferIsComplete(hBuffer,&bIsComplete);
```

See also

BUFFER_HANDLE
SDK4DSWaitForBuffer

2.3.4.5 SDK4BufferSizeFilled

This function returns the number of bytes, with which the buffer was filled.
If SDK4BufferIsComplete returns 1, this number should be equal to the value, returned by SDK4DSGetPayloadSize.
If SDK4BufferIsComplete returns 0, this number, compared to the value of SDK4DSGetPayloadSize, indicates the number of lost bytes.

```
SDK4_Error SDK4BufferSizeFilled(BUFFER_HANDLE hBuffer ,int32_t*
          pSizeFilled)
```

Parameters

hBuffer(BUFFER_HANDLE)
The buffer handle
pSizeFilled (uint32_t *)
The pointer to the number of transmitted bytes

Return value

Error code:

`SDK4_ERR_SUCCESS (0)` in case of no error
`SDK4_ERR_ERROR (-1001)` in case of unspecified error
one of the other error codes defined in `SDK4_ERROR`

Example

```
DS_HANDLE hDataStream;
...
SDK4DevStartAcquisition(hDev);
BUFFER_HANDLE hBuffer;
int32_t bIsComplete=0;
int32_t sizeFilled=0;
SDK4DSWaitForBuffer(hDataStream,& hBuffer,nTimeout);
SDK4BufferIsComplete(hBuffer,&bIsComplete);
if (!bIsComplete)
      SDK4BufferSizeFilled(hBuffer,&sizeFilled);
```

See also

`BUFFER_HANDLE`
`SDK4DSWaitForBuffer`

2.4 SDK4ZelosControl

2.4.1 GigEVision Control Functions

2.4.1.1 SDK4GetGigEVersion

This function gets the current Version of GigEVision Standard, the camera is conform to.

```
SDK4_Error SDK4GetGigEVersion(CTRL_HANDLE hCtrl,uint32_t
           *puGigEVersion)
```

Parameters

`hCtrl(CTRL_HANDLE )`

The control handle of the device

`puGigEVersion (uint32_t * )`

The pointer to the version value

Return value

Error code:

`SDK4_ERR_SUCCESS (0)` in case of no error

`SDK4_ERR_ERROR (-1001)` in case of unspecified error

one of the other error codes defined in `SDK4_ERROR`

Remarks

The actual cameras are conform to GigEVision Version 1.1. (version value 0x00010001).

Example

```
...
DEV_HANDLE hDev;
CTRL_HANDLE hCtrl;
SDK4DevGetControl(hDev, &hCtrl);
uint32_t uGigEVersion = 0;
SDK4GetGigEVersion(hCtrl,&uGigEVersion);
```

See also

`SDK4DevGetControl`

`CTRL_HANDLE`

2.4.1.2 SDK4GetMacAddress

This function gets the MAC Address of the camera.

```
SDK4_Error SDK4GetMacAddress(CTRL_HANDLE hCtrl,char * psMacAddress,
           int32_t *piSize)
```

Parameters

`hCtrl(CTRL_HANDLE )`

The control handle of the device

`psMacAddress(char * )`

the character array of the MAC Address</param>

`piSize(int32_t *)`

the pointer to the maximal array size

Return value

Error code:

`SDK4_ERR_SUCCESS (0)` in case of no error
`SDK4_ERR_ERROR (-1001)` in case of unspecified error
one of the other error codes defined in `SDK4_ERROR`

Example

```
...
DEV_HANDLE hDev;
CTRL_HANDLE hCtrl;
SDK4DevGetControl(hDev, &hCtrl);
char sMacAddress [256];
int32_t size=sizeof(sMacAddress);
SDK4GetMacAddress(hCtrl,&sMacAddress, &size);
```

See also

`SDK4DevGetControl`
`CTRL_HANDLE`

2.4.1.3 SDK4GetCurrentIPAddress

This function gets the current IP Address of the camera.

```
SDK4_Error SDK4GetCurrentIPAddress(CTRL_HANDLE hCtrl,char*
          psIPAddress,int32_t *piSize)
```

Parameters

`hCtrl(CTRL_HANDLE )`
The control handle of the device

`psIPAddress (char * )`
the character array of the IP Address</param>

`piSize(int32_t *)`
the pointer to the maximal array size

Return value

Error code:

`SDK4_ERR_SUCCESS (0)` in case of no error
`SDK4_ERR_ERROR (-1001)` in case of unspecified error
one of the other error codes defined in `SDK4_ERROR`

Example

```
...
DEV_HANDLE hDev;
CTRL_HANDLE hCtrl;
SDK4DevGetControl(hDev, &hCtrl);
char sIPAddress [256];
int32_t size=sizeof(sIPAddress);
SDK4GetCurrentIPAddress (hCtrl,&sIPAddress, &size);
```

See also

`SDK4DevGetControl`
`CTRL_HANDLE`

2.4.1.4 SDK4GetCurrentSubnetmask

This function gets the current Subnet mask of the camera.

```
SDK4_Error SDK4GetCurrentSubnetmask(CTRL_HANDLE hCtrl,char*
          psSubnetmask,int32_t *piSize)
```

Parameters

`hCtrl(CTRL_HANDLE )`
The control handle of the device

`psSubnetmask (char * )`
the character array of the Subnet Mask</param>

`piSize(int32_t *)`
the pointer to the maximal array size

Return value

Error code:

`SDK4_ERR_SUCCESS (0)` in case of no error

`SDK4_ERR_ERROR (-1001)` in case of unspecified error

one of the other error codes defined in `SDK4_ERROR`

Example

```
...
DEV_HANDLE hDev;
CTRL_HANDLE hCtrl;
SDK4DevGetControl(hDev, &hCtrl);
char sSubnetmask [256];
int32_t size=sizeof(sSubnetmask);
SDK4GetCurrentSubnetmask (hCtrl,&sSubnetmask, &size);
```

See also

`SDK4DevGetControl`

`CTRL_HANDLE`

2.4.1.5 SDK4GetCurrentDefaultGateway

This function gets the Default Gateway of the camera.

```
SDK4_Error SDK4GetCurrentDefaultGateway(CTRL_HANDLE hCtrl,char*
          psDefaultGateway,int32_t *piSize)
```

Parameters

`hCtrl(CTRL_HANDLE )`
The control handle of the device

`psDefaultGateway (char * )`
the character array of the Default Gateway</param>

`piSize(int32_t *)`
the pointer to the maximal array size

Return value

Error code:

`SDK4_ERR_SUCCESS (0)` in case of no error
`SDK4_ERR_ERROR (-1001)` in case of unspecified error
one of the other error codes defined in `SDK4_ERROR`

Example

```
...
DEV_HANDLE hDev;
CTRL_HANDLE hCtrl;
SDK4DevGetControl(hDev, &hCtrl);
char sDefaultGateway [256];
int32_t size=sizeof(sDefaultGateway);
SDK4GetCurrentDefaultGateway (hCtrl,& sDefaultGateway, &size);
```

See also

`SDK4DevGetControl`
`CTRL_HANDLE`

2.4.1.6 SDK4SetPacketSize

This function sets the packet size of the UDP packets, by which the image frames are sent by the camera.

```
SDK4_Error SDK4SetPacketSize(CTRL_HANDLE hCtrl,uint32_t uPacketSize)
```

Parameters

`hCtrl (CTRL_HANDLE )`
The control handle of the device
`uPacketSize (uint32_t )`
the packet size

Return value

Error code:

`SDK4_ERR_SUCCESS (0)` in case of no error
`SDK4_ERR_ERROR (-1001)` in case of unspecified error
one of the other error codes defined in `SDK4_ERROR`

Remarks

The minimal packet size should be set to 4k Bytes, otherwise packet loss could occur. Packet size must not be greater than Jumbo frame size or MTU size of the network card. The packet size value should be increased in steps of 32.

Example

```
...
DEV_HANDLE hDev;
CTRL_HANDLE hCtrl;
SDK4DevGetControl(hDev, &hCtrl);
SDK4SetPacketSize (hCtrl, 8192);
```

See also

```
SDK4DevGetControl
CTRL_HANDLE
```

2.4.1.7 SDK4GetPacketSize

This function gets the current size of the UDP streaming packets.

```
SDK4_Error SDK4GetPacketSize(CTRL_HANDLE hCtrl,uint32_t*
          puPacketSize)
```

Parameters

`hCtrl (CTRL_HANDLE )`

The control handle of the device

`puPacketSize (uint32_t )`

the pointer to the packet size

Return value

Error code:

`SDK4_ERR_SUCCESS (0)` in case of no error
`SDK4_ERR_ERROR (-1001)` in case of unspecified error
one of the other error codes defined in `SDK4_ERROR`

Example

```
...
DEV_HANDLE hDev;
CTRL_HANDLE hCtrl;
SDK4DevGetControl(hDev, &hCtrl);
uint32_t size=0;
SDK4GetPacketSize (hCtrl, &size);
```

See also

```
SDK4DevGetControl
CTRL_HANDLE
```

2.4.2 General Device Functions

2.4.2.1 SDK4GetDeviceName

This function gets the name of the device.

```
SDK4_Error SDK4GetDeviceName(CTRL_HANDLE hCtrl,char * psDeviceName,
          int32_t *piSize)
```

Parameters

`hCtrl(CTRL_HANDLE )`

The control handle of the device

`psDeviceName (char * )`

the character array of the Device Name</param>

`piSize(int32_t *)`

the pointer to the maximal array size

Return value

Error code:

`SDK4_ERR_SUCCESS (0)` in case of no error
`SDK4_ERR_ERROR (-1001)` in case of unspecified error
one of the other error codes defined in `SDK4_ERROR`

Example

```
...
DEV_HANDLE hDev;
CTRL_HANDLE hCtrl;
SDK4DevGetControl(hDev, &hCtrl);
char sDeviceName [256];
int32_t size=sizeof(sDeviceName);
SDK4GetDeviceName (hCtrl,&sDeviceName, &size);
```

See also

`SDK4DevGetControl`
`CTRL_HANDLE`

2.4.2.2 SDK4GetSerialNumber

This function gets the serial number of the device.

```
SDK4_Error SDK4GetSerialNumber(CTRL_HANDLE hCtrl,char *
           psSerialNumber, int32_t *piSize)
```

Parameters

`hCtrl(CTRL_HANDLE )`
The control handle of the device

`psSerialNumber (char * )`
the character array of the serial number</param>

`piSize(int32_t *)`
the pointer to the maximal array size

Return value

Error code:

`SDK4_ERR_SUCCESS (0)` in case of no error
`SDK4_ERR_ERROR (-1001)` in case of unspecified error
one of the other error codes defined in `SDK4_ERROR`

Example

```
...
DEV_HANDLE hDev;
CTRL_HANDLE hCtrl;
SDK4DevGetControl(hDev, &hCtrl);
char sSerialNumber [256];
int32_t size=sizeof(sSerialNumber);
SDK4GetSerialNumber (hCtrl,& sSerialNumber, &size);
```

See also

`SDK4DevGetControl`
`CTRL_HANDLE`

2.4.2.3 SDK4GetFunctionModelID

This function gets the function model ID of the device.

```
SDK4_Error SDK4GetFunctionModelID(CTRL_HANDLE hCtrl,uint32_t *
           puModelID)
```

Parameters

`hCtrl(CTRL_HANDLE )`

The control handle of the device

`puModelID (uint32_t *)`

the pointer to the function model ID

Return value

Error code:

`SDK4_ERR_SUCCESS (0)` in case of no error
`SDK4_ERR_ERROR (-1001)` in case of unspecified error
one of the other error codes defined in `SDK4_ERROR`

Example

```
...
DEV_HANDLE hDev;
CTRL_HANDLE hCtrl;
SDK4DevGetControl(hDev, &hCtrl);
uint32_t id=0;
SDK4GetFunctionModelID (hCtrl,&id);
```

See also

`SDK4DevGetControl`
`CTRL_HANDLE`

2.4.2.4 SDK4GetFunctionModelVersion

This function gets the function model version of the device.

```
SDK4_Error SDK4GetFunctionModelVersion(CTRL_HANDLE
           hCtrl,SDK4_KVERSION * pkVersion)
```

Parameters

`hCtrl(CTRL_HANDLE )`

The control handle of the device

`pkVersion (SDK4_KVERSION *)`

the pointer to the function model version

Return value

Error code:

`SDK4_ERR_SUCCESS (0)` in case of no error
`SDK4_ERR_ERROR (-1001)` in case of unspecified error
one of the other error codes defined in `SDK4_ERROR`

Example

```
...
DEV_HANDLE hDev;
CTRL_HANDLE hCtrl;
SDK4DevGetControl(hDev, &hCtrl);
SDK4_KVERSION version;
SDK4GetFunctionModelVersion (hCtrl,& version);
```

See also

```
SDK4DevGetControl
CTRL_HANDLE
SDK4_KVERSION
```

2.4.2.5 SDK4GetRevision

This function gets the Revision of the device.

```
SDK4_Error SDK4GetRevision(CTRL_HANDLE hCtrl,SDK4_KREVISION *
           pkRevision)
```

Parameters

`hCtrl(CTRL_HANDLE )`
    The control handle of the device

`pkRevision (SDK4_ KREVISION *)`
    the pointer to the revision

Return value

Error code:

`SDK4_ERR_SUCCESS (0)` in case of no error
`SDK4_ERR_ERROR (-1001)` in case of unspecified error
one of the other error codes defined in `SDK4_ERROR`

Example

```
...
DEV_HANDLE hDev;
CTRL_HANDLE hCtrl;
SDK4DevGetControl(hDev, &hCtrl);
SDK4_ KREVISION revision;
SDK4GetRevision (hCtrl,&revision);
```

See also

```
SDK4DevGetControl
CTRL_HANDLE
SDK4_ KREVISION
```

2.4.2.6 SDK4GetSoftwareBuild

This function gets the Software Build of the device.

```
SDK4_Error SDK4GetSoftwareBuild(CTRL_HANDLE hCtrl,char *
          psSoftwareBuild, int32_t *piSize)
```

Parameters

`hCtrl(CTRL_HANDLE )`

The control handle of the device

`psSoftwareBuild (char * )`

the character array of the software build</param>

`piSize(int32_t *)`

the pointer to the maximal array size

Return value

Error code:

`SDK4_ERR_SUCCESS (0)` in case of no error

`SDK4_ERR_ERROR (-1001)` in case of unspecified error

one of the other error codes defined in `SDK4_ERROR`

Example

```
...
DEV_HANDLE hDev;
CTRL_HANDLE hCtrl;
SDK4DevGetControl(hDev, &hCtrl);
char sSoftwareBuild [256];
int32_t size=sizeof(sSoftwareBuild);
SDK4GetSoftwareBuild (hCtrl,& sSoftwareBuild, &size);
```

See also

`SDK4DevGetControl`

`CTRL_HANDLE`

2.4.2.7 SDK4GetSoftwareVersion

This function gets the Software Version of the device.

```
SDK4_Error SDK4GetSoftwareVersion(CTRL_HANDLE hCtrl,SDK4_KVERSION *
          pkVersion)
```

Parameters

`hCtrl(CTRL_HANDLE )`

The control handle of the device

`pkVersion (SDK4_KVERSION *)`

the pointer to the Software Version

Return value

Error code:

`SDK4_ERR_SUCCESS (0)` in case of no error

`SDK4_ERR_ERROR (-1001)` in case of unspecified error

one of the other error codes defined in `SDK4_ERROR`

Example

```
...
DEV_HANDLE hDev;
CTRL_HANDLE hCtrl;
SDK4DevGetControl(hDev, &hCtrl);
SDK4_KVERSION version;
SDK4GetSoftwareVersion (hCtrl,& version);
```

See also

SDK4DevGetControl
CTRL_HANDLE
SDK4_KVERSION

2.4.2.8 SDK4GetHardwareVersion

This function gets the Hardware Version of the device.

```
SDK4_Error SDK4GetHardwareVersion(CTRL_HANDLE hCtrl,SDK4_KVERSION *
          pkVersion)
```

Parameters

hCtrl(CTRL_HANDLE)

The control handle of the device

pkVersion (SDK4_KVERSION *)

the pointer to the Hardware Version

Return value

Error code:

SDK4_ERR_SUCCESS (0) in case of no error
SDK4_ERR_ERROR (-1001) in case of unspecified error
one of the other error codes defined in SDK4_ERROR

Example

```
...
DEV_HANDLE hDev;
CTRL_HANDLE hCtrl;
SDK4DevGetControl(hDev, &hCtrl);
SDK4_KVERSION version;
SDK4GetHardwareVersion (hCtrl,& version);
```

See also

SDK4DevGetControl
CTRL_HANDLE
SDK4_KVERSION

2.4.2.9 SDK4GetFirmwareVersion

This function gets the Firmware Version of the device.

```
SDK4_Error SDK4GetFirmwareVersion(CTRL_HANDLE hCtrl,SDK4_KVERSION *
           pkVersion)
```

Parameters

`hCtrl(CTRL_HANDLE )`

The control handle of the device

`pkVersion (SDK4_KVERSION *)`

the pointer to the Firmware Version

Return value

Error code:

`SDK4_ERR_SUCCESS (0)` in case of no error

`SDK4_ERR_ERROR (-1001)` in case of unspecified error

one of the other error codes defined in `SDK4_ERROR`

Example

```
...
DEV_HANDLE hDev;
CTRL_HANDLE hCtrl;
SDK4DevGetControl(hDev, &hCtrl);
SDK4_KVERSION version;
SDK4GetFirmwareVersion (hCtrl,& version);
```

See also

`SDK4DevGetControl`

`CTRL_HANDLE`

`SDK4_KVERSION`

2.4.3 Exposure Functions

2.4.3.1 SDK4SetExposureMode

This function sets the Exposure Mode to one of the values specified in SDK4_ENUM_EXPOSUREMODE.

```
SDK4_Error SDK4SetExposureMode(CTRL_HANDLE hCtrl,
           SDK4_ENUM_EXPOSUREMODE uMode)
```

Parameters

`hCtrl(CTRL_HANDLE )`

The control handle of the device

`uMode(SDK4_ENUM_EXPOSUREMODE )`

the exposure mode value

Return value

Error code:

`SDK4_ERR_SUCCESS (0)` in case of no error
`SDK4_ERR_ERROR (-1001)` in case of unspecified error
one of the other error codes defined in `SDK4_ERROR`

Remarks

In ZELOS_ENUM_EXPOSUREMODE_FREERUNNINGPARALLEL CCD exposure and CCD readout phases will be performed parallel. Due to this fact the maximum exposure time is the same as CCD readout time and depends on CCD, ROA settings and Binning mode.
In ZELOS_ENUM_EXPOSUREMODE_FREERUNNINGSEQUENTIAL CCD exposure and CCD readout phases will be performed sequentially.
In this mode there is theoretically no limit in exposure time. The frame rate depends on the adjusted exposure time.
In ZELOS_ENUM_EXPOSUREMODE_RESETRESTART CCD exposure starts at an external trigger signal or in case of an incoming Software Trigger message.
In ZELOS_ENUM_EXPOSUREMODE_FRAMEONDEMAND CCD exposure and CCD readout phases will be performed sequentially. The exposure time will be controlled by the length of trigger input signal. The period time of trigger signal controls the image frequency. Due to this fact the value set by the Exposure Message or Read by the Exposure Message is not valid.
In ZELOS_ENUM_EXPOSUREMODE_EXTERNALSYNC CCD exposure and CCD readout phases will be performed parallel. The beginning of the exposure will by synchronized to the external SYNC signal. In case the external SYNC signal is missing for a certain time, the camera performs its internal timing until the external SYNC signal establishes again. To keep a similar timing in that case it is advisable to adjust the frame duration to a slightly higher value than the duration of the extern SYNC Signal. Please refer to Frame Duration Message.

Example

```
CTRL_HANDLE hCtrl;
...
uint32_t uReadoutTime;
SDK4_KEXPOSURE kExposure;
SDK4SetExposureMode(
     hCtrl, ZELOS_ENUM_EXPOSUREMODE_FREERUNNINGPARALLEL);
SDK4GetExposure(hCtrl,&kExposure);
SDK4GetReadoutTime(hCtrl,&uReadoutTime);
```

See also

SDK4DevGetControl
CTRL_HANDLE
SDK4_ENUM_EXPOSUREMODE
SDK4SetExposure

2.4.3.2 SDK4GetExposureMode

This function gets the current value of the Exposure Mode.

```
SDK4_Error SDK4GetExposureMode(CTRL_HANDLE hCtrl,
          SDK4_ENUM_EXPOSUREMODE* puMode)
```

Parameters

hCtrl(CTRL_HANDLE)

The control handle of the device

puMode(SDK4_ENUM_EXPOSUREMODE *)

the pointer to the exposure mode value

Return value

Error code:

SDK4_ERR_SUCCESS (0) in case of no error
SDK4_ERR_ERROR (-1001) in case of unspecified error
one of the other error codes defined in SDK4_ERROR

Example

```
CTRL_HANDLE hCtrl;
...
uint32_t uReadoutTime;
SDK4_KEXPOSURE kExposure;
SDK4_ENUM_EXPOSUREMODE uMode;
SDK4GetExposureMode(hCtrl, &uMode);
if (uMode == ZELOS_ENUM_EXPOSUREMODE_FREERUNNINGPARALLEL)
{
      SDK4GetExposure(hCtrl,&kExposure);
      SDK4GetReadoutTime(hCtrl,&uReadoutTime);
}
```

See also

SDK4DevGetControl
CTRL_HANDLE
SDK4_ENUM_EXPOSUREMODE
SDK4SetExposureMode

2.4.3.3 SDK4SetExposure

This function sets the Exposure Time.

```
SDK4_Error SDK4SetExposure(CTRL_HANDLE hCtrl,SDK4_KEXPOSURE
           kExposure)
```

Parameters

`hCtrl(CTRL_HANDLE )`

The control handle of the device

`kExposure (SDK4_KEXPOSURE)`

the exposure time value

Return value

Error code:

`SDK4_ERR_SUCCESS (0)` in case of no error

`SDK4_ERR_ERROR (-1001)` in case of unspecified error

one of the other error codes defined in `SDK4_ERROR`

Remarks

Exposure Time can only be controlled manually, when AET is switched off. Its maximal value depends on the current Exposure Mode.

The Exposure value is calculated by SDK4_KEXPOSURE.Base x SDK4_KEXPOSURE.Counter.

In case SDK4_KEXPOSURE.Base is set to SDK4_ENUM_EXPOSUREBASE_PIXELCLOCK the Exposure value is dependent of the PixelClock value (see SDK4GetPixelClock). Thus it is calculated by SDK4_KEXPOSURE.Counter / pixelClock.

Example

```
CTRL_HANDLE hCtrl;
...
SDK4_ENUM_AET uAET;
SDK4GetAET(hCtrl, &uAET);
if (uAET == SDK4_ENUM_AET_OFF)
{
     SDK4SetExposureMode(
          hCtrl, ZELOS_ENUM_EXPOSUREMODE_FREERUNNINGSEQUENTIAL);
     // Set ExposureTime to 300 ms:
     SDK4_KEXPOSURE kExposure;
     kExposure.Base = SDK4_ENUM_EXPOSUREBASE_100ms;
     kExposure.Counter = 3;
     SDK4SetExposure(hCtrl,kExposure);
}
```

See also

`SDK4DevGetControl`
`CTRL_HANDLE`
`SDK4_KEXPOSURE`
`SDK4_ENUM_EXPOSUREBASE`
`SDK4SetExposureMode`
`SDK4GetAET`
`SDK4GetPixelClock`

2.4.3.4 SDK4GetExposure

This function gets the current value of the Exposure Time.

```
SDK4_Error SDK4GetExposure(CTRL_HANDLE hCtrl,SDK4_KEXPOSURE*
           pkExposure)
```

Parameters

`hCtrl(CTRL_HANDLE )`

The control handle of the device

`pkExposure (SDK4_KEXPOSURE*)`

the pointer to the exposure time value

Return value

Error code:

`SDK4_ERR_SUCCESS (0)` in case of no error

`SDK4_ERR_ERROR (-1001)` in case of unspecified error

one of the other error codes defined in `SDK4_ERROR`

Remarks

The Exposure value is calculated by SDK4_KEXPOSURE.Base x SDK4_KEXPOSURE.Counter. In case SDK4_KEXPOSURE.Base is set to SDK4_ENUM_EXPOSUREBASE_PIXELCLOCK the Exposure value is dependent of the PixelClock value (see SDK4GetPixelClock). Thus it is calculated by SDK4_KEXPOSURE.Counter / pixelClock.

Example

```
CTRL_HANDLE hCtrl;
...
SDK4_KEXPOSURE kExposure;
SDK4GetExposure(hCtrl,&kExposure);
```

See also

`SDK4DevGetControl`
`CTRL_HANDLE`
`SDK4_KEXPOSURE`
`SDK4_ENUM_EXPOSUREBASE`
`SDK4SetExposure`
`SDK4GetPixelClock`

2.4.3.5 SDK4SetAutoExposureLevel

This function sets the Automatic Exposure Level (i. e. the target level of AET and AGC).

```
SDK4_Error SDK4SetAutoExposureLevel(CTRL_HANDLE hCtrl,uint32_t
           uLevel)
```

Parameters

`hCtrl(CTRL_HANDLE )`

The control handle of the device

`uLevel(uint32_t )`

the auto exposure level value

Return value

Error code:

`SDK4_ERR_SUCCESS (0)` in case of no error

`SDK4_ERR_ERROR (-1001)` in case of unspecified error

one of the other error codes defined in `SDK4_ERROR`

Remarks

The possible values range from 0 to 255. AutoExposureLevel has only function, when either AET or AGC or both are active.

Example

```
CTRL_HANDLE hCtrl;
...
SDK4_ENUM_AET uAET;
SDK4GetAET(hCtrl, &uAET);
SDK4_ENUM_AGC uAGC;
SDK4GetAGC(hCtrl, &uAGC);
// if possible, set the level to 30:
if ((uAET == SDK4_ENUM_AET_ON) || (uAGC == ZELOS_ENUM_AGC_ON))
     SDK4SetAutoExposureLevel(hCtrl, 30);
```

See also

```
SDK4DevGetControl
CTRL_HANDLE
SDK4GetAutoExposureLevel
SDK4GetAET
SDK4GetAGC
```

2.4.3.6 SDK4GetAutoExposureLevel

This function gets the current value of Automatic Exposure Level.

```
SDK4_Error SDK4GetAutoExposureLevel(CTRL_HANDLE hCtrl,uint32_t*
          puLevel)
```

Parameters

`hCtrl(CTRL_HANDLE )`

The control handle of the device

`puLevel(uint32_t )`

the pointer to the auto exposure level value

Return value

Error code:

`SDK4_ERR_SUCCESS (0)` in case of no error

`SDK4_ERR_ERROR (-1001)` in case of unspecified error

one of the other error codes defined in `SDK4_ERROR`

Example

```
CTRL_HANDLE hCtrl;
...
uint32_t uLevel
SDK4GetAutoExposureLevel(hCtrl,&uLevel);
```

See also

`SDK4DevGetControl`

`CTRL_HANDLE`

`SDK4SetAutoExposureLevel`

2.4.3.7 SDK4SetAET

This function switches the Automatic Exposure Time on and off.

```
SDK4_Error SDK4SetAET(CTRL_HANDLE hCtrl,SDK4_ENUM_AET uAet)
```

Parameters

`hCtrl(CTRL_HANDLE )`

The control handle of the device

`uAet (SDK4_ENUM_AET )`

the aet value

Return value

Error code:

`SDK4_ERR_SUCCESS (0)` in case of no error

`SDK4_ERR_ERROR (-1001)` in case of unspecified error

one of the other error codes defined in `SDK4_ERROR`

Remarks

The camera automatically changes the exposure time in the range of 10µs to max. exposure time at exposure mode ZELOS_ENUM_EXPOSUREMODE_FREERUNNINGPARALLEL. However the automatic will run at both ZELOS_ENUM_EXPOSUREMODE_FREERUNNINGPARALLEL or ZELOS_ENUM_EXPOSUREMODE_FREERUNNINGSEQUENTIAL mode.
The exposure time can only be controlled with the exposure messages when the AET is switched off.
The SDK4_ENUM_AET_ON_EXT mode of the automatic allows the camera to switch the exposure mode from ZELOS_ENUM_EXPOSUREMODE_FREERUNNINGPARALLEL to ZELOS_ENUM_EXPOSUREMODE_FREERUNNINGSEQUENTIAL mode in case of short exposure times. This makes the automatic more stable for short exposure times.

Example

```
CTRL_HANDLE hCtrl;
...
SDK4SetAET(hCtrl, SDK4_ENUM_AET_OFF);
```

See also

```
SDK4DevGetControl
CTRL_HANDLE
SDK4GetAET
SDK4_ENUM_AET
```

2.4.3.8 SDK4GetAET

This function switches the Automatic Exposure Time on and off.

```
SDK4_Error SDK4GetAET(CTRL_HANDLE hCtrl,SDK4_ENUM_AET* puAet)
```

Parameters

`hCtrl(CTRL_HANDLE )`

The control handle of the device

`puAet (SDK4_ENUM_AET* )`

the pointer to the aet value

Return value

Error code:

`SDK4_ERR_SUCCESS (0)` in case of no error
`SDK4_ERR_ERROR (-1001)` in case of unspecified error
one of the other error codes defined in `SDK4_ERROR`

Example

```
CTRL_HANDLE hCtrl;
...
SDK4_ENUM_AET uAET;
SDK4GetAET(hCtrl, &uAET);
SDK4_ENUM_AGC uAGC;
SDK4GetAGC(hCtrl, &uAGC);
// if possible, set the level to 30:
if ((uAET == SDK4_ENUM_AET_ON) || (uAGC == ZELOS_ENUM_AGC_ON))
     SDK4SetAutoExposureLevel(hCtrl, 30);
```

See also

`SDK4DevGetControl`
`CTRL_HANDLE`
`SDK4SetAET`
`SDK4_ENUM_AET`

2.4.4 Gain Functions

2.4.4.1 SDK4SetGain

This function sets the manual gain of the camera.

```
SDK4_Error SDK4SetGain(CTRL_HANDLE hCtrl,uint32_t uGain)
```

Parameters

`hCtrl(CTRL_HANDLE )`
    The control handle of the device
`uGain(uint32_t )`
    the gain value

Return value

Error code:

- `SDK4_ERR_SUCCESS (0)` in case of no error
- `SDK4_ERR_ERROR (-1001)` in case of unspecified error
- one of the other error codes defined in `SDK4_ERROR`

Remarks

This function is only active, when AGC is switched off. The values range between 0 and 511 (0 to 18 dB, each step 0.035 dB).

Example

```
CTRL_HANDLE hCtrl;
...
SDK4_ENUM_AGC uAGC;
SDK4GetAGC(hCtrl, &uAGC);
// if possible, set gain to 10:
if (uAGC == ZELOS_ENUM_AGC_OFF)
      SDK4SetGain(hCtrl, 10);
```

See also

`SDK4DevGetControl`
`CTRL_HANDLE`
`SDK4GetGain`
`SDK4GetAGC`

2.4.4.2 SDK4GetGain

This function gets the current gain value of the camera.

```
SDK4_Error SDK4GetGain(CTRL_HANDLE hCtrl,uint32_t* puGain)
```

Parameters

`hCtrl(CTRL_HANDLE )`

The control handle of the device

`puGain(uint32_t )`

the pointer to the gain value

Return value

Error code:

`SDK4_ERR_SUCCESS (0)` in case of no error

`SDK4_ERR_ERROR (-1001)` in case of unspecified error

one of the other error codes defined in `SDK4_ERROR`

Example

```
CTRL_HANDLE hCtrl;
...
uint32_t uGain
SDK4GetGain(hCtrl,&uGain);
```

See also

`SDK4DevGetControl`

`CTRL_HANDLE`

`SDK4SetGain`

2.4.4.3 SDK4SetAGC

This function sets the Automatic Gain Control on and off.

```
SDK4_Error SDK4SetAGC(CTRL_HANDLE hCtrl,SDK4_ENUM_AGC uOnOff)
```

Parameters

`hCtrl(CTRL_HANDLE )`

The control handle of the device

`uOnOff (SDK4_ENUM_AGC)`

the agc switch

Return value

Error code:

`SDK4_ERR_SUCCESS (0)` in case of no error

`SDK4_ERR_ERROR (-1001)` in case of unspecified error

one of the other error codes defined in `SDK4_ERROR`

Remarks

This function changes gain in a range of 0 to 18 dB, when AGC is switched on.

Example

```
CTRL_HANDLE hCtrl;
...
SDK4SetAGC(hCtrl, SDK4_ENUM_AGC_OFF);
```

See also

`SDK4DevGetControl`
`CTRL_HANDLE`
`SDK4_ENUM_AGC`
`SDK4SetGain`
`SDK4GetAGC`

2.4.4.4 SDK4GetAGC

This function gets the current value of Automatic Gain Control.

```
SDK4_Error SDK4GetAGC(CTRL_HANDLE hCtrl,SDK4_ENUM_AGC* puOnOff)
```

Parameters

`hCtrl(CTRL_HANDLE )`

The control handle of the device

`puOnOff (SDK4_ENUM_AGC*)`

the pointer to the agc switch

Return value

Error code:

`SDK4_ERR_SUCCESS (0)` in case of no error
`SDK4_ERR_ERROR (-1001)` in case of unspecified error
one of the other error codes defined in `SDK4_ERROR`

Example

```
CTRL_HANDLE hCtrl;
...
SDK4_ENUM_AET uAET;
SDK4GetAET(hCtrl, &uAET);
SDK4_ENUM_AGC uAGC;
SDK4GetAGC(hCtrl, &uAGC);
// if possible, set the level to 30:
if ((uAET == SDK4_ENUM_AET_ON) || (uAGC == ZELOS_ENUM_AGC_ON))
     SDK4SetAutoExposureLevel(hCtrl, 30);
```

See also

`SDK4DevGetControl`
`CTRL_HANDLE`
`SDK4_ENUM_AGC`
`SDK4SetAGC`

2.4.5 Timing Functions

2.4.5.1 SDK4GetPixelClock

This function gets the current Pixel Clock of the camera (in Hz).

```
SDK4_Error SDK4GetPixelClock(CTRL_HANDLE hCtrl, uint32_t*
          puPixelClock)
```

Parameters

`hCtrl(CTRL_HANDLE )`

The control handle of the device

`puPixelClock (uint32_t*)`

the pointer to the pixel clock value

Return value

Error code:

`SDK4_ERR_SUCCESS (0)` in case of no error

`SDK4_ERR_ERROR (-1001)` in case of unspecified error

one of the other error codes defined in `SDK4_ERROR`

Remarks

Activating the slow scan feature reduces the pixel clock to increase the sensor performance.

Example

```
CTRL_HANDLE hCtrl;
...
uint32_t uPixelClock;
SDK4GetPixelClock (hCtrl, &uPixelClock);
```

See also

`SDK4DevGetControl`

`CTRL_HANDLE`

`SDK4SetSlowScan`

2.4.5.2 SDK4GetReadoutTime

This function gets the actual time, which is needed to read out the CCD (in units of 100 µs).

```
SDK4_Error SDK4GetReadoutTime(CTRL_HANDLE hCtrl,uint32_t*
          puReadoutTime)
```

Parameters

`hCtrl(CTRL_HANDLE )`

The control handle of the device

`puReadoutTime (uint32_t*)`

the pointer to the Readout Time value

Return value

Error code:

`SDK4_ERR_SUCCESS (0)` in case of no error
`SDK4_ERR_ERROR (-1001)` in case of unspecified error
one of the other error codes defined in `SDK4_ERROR`

Remarks

Readout time depends on exposure time and exposure mode.
In ZELOS_ENUM_EXPOSUREMODE_FREERUNNINGPARALLEL this value is the same as the maximum exposure time you can use.
In ZELOS_ENUM_EXPOSUREMODE_FREERUNNINGSEQUENTIAL this value is the sum of CCD exposure and CCD readout phase.

Example

```
CTRL_HANDLE hCtrl;
...
uint32_t uReadoutTime
SDK4GetReadoutTime(hCtrl, &uReadoutTime);
// calculate readout time in seconds:
double dReadoutTime = (double)uReadoutTime/(double)10000;
```

See also

`SDK4DevGetControl`
`CTRL_HANDLE`
`SDK4GetPixelClock`

2.4.5.3 SDK4SetReadoutTime

This function sets the Readout Time of the camera (in units of 100 µs).

```
SDK4_Error SDK4SetReadoutTime(CTRL_HANDLE hCtrl,uint32_t
          uReadoutTime)
```

Parameters

`hCtrl(CTRL_HANDLE )`
The control handle of the device
`uReadoutTime (uint32_t)`
the Readout Time value

Return value

Error code:

`SDK4_ERR_SUCCESS (0)` in case of no error
`SDK4_ERR_ERROR (-1001)` in case of unspecified error
one of the other error codes defined in `SDK4_ERROR`

Remarks

The readout time can be set to values, which are longer than the minimal readout time at the selected exposure mode and exposure time settings.

Example

```
CTRL_HANDLE hCtrl;
...
// set a readout time of 100 ms:
SDK4SetReadoutTime(hCtrl, 1000);
```

See also

```
SDK4DevGetControl
CTRL_HANDLE
SDK4GetReadoutTime
SDK4SetExposure
SDK4SetExposureMode
```

2.4.5.4 SDK4GetFrameDuration

This function gets the actual Frame Duration (in units of 100 µs).

```
SDK4_Error SDK4GetFrameDuration(CTRL_HANDLE hCtrl,uint32_t*
           puFrameDuration)
```

Parameters

`hCtrl(CTRL_HANDLE )`

    The control handle of the device

`puFrameDuration (uint32_t*)`

    the pointer to the Frame Duration value

Return value

Error code:

- `SDK4_ERR_SUCCESS (0)` in case of no error
- `SDK4_ERR_ERROR (-1001)` in case of unspecified error
- one of the other error codes defined in `SDK4_ERROR`

Remarks

Frame duration means the time difference between the beginning of two frames (in 100 µs units).

Example

```
CTRL_HANDLE hCtrl;
...
uint32_t uFrameDuration;
SDK4GetFrameDuration(hCtrl,&uFrameDuration);
```

See also

```
SDK4DevGetControl
CTRL_HANDLE
```

2.4.6 Trigger Functions

2.4.6.1 SDK4SetSoftTrigger

This function sends a software trigger to the camera.

```
SDK4_Error SetSoftTrigger(CTRL_HANDLE hCtrl,SDK4_ENUM_SWITCH uOnOff)
```

Parameters

`hCtrl(CTRL_HANDLE )`

The control handle of the device

`uOnOff (SDK4_ENUM_SWITCH)`

the trigger switch

Return value

Error code:

`SDK4_ERR_SUCCESS (0)` in case of no error
`SDK4_ERR_ERROR (-1001)` in case of unspecified error
one of the other error codes defined in `SDK4_ERROR`

Remarks

The trigger in send, if the switch is set to ON. The camera immediately starts an image exposure and readout sequence.
This function is only applicable in exposure mode ZELOS_ENUM_EXPOSUREMODE_RESETRESTART.

Example

```
CTRL_HANDLE hCtrl;
...
uint32_t uReadoutTime;
SDK4_KEXPOSURE kExposure;
SDK4SetExposureMode(hCtrl, ZELOS_ENUM_EXPOSUREMODE_RESETRESTART);
SetSoftTrigger(hCtrl, SDK4_ENUM_SWITCH_ON);
```

See also

`SDK4DevGetControl`
`CTRL_HANDLE`
`SDK4SetExposureMode`

2.4.6.2 SDK4SetTriggerTimer

This function sets the trigger timer of the camera.

```
SDK4_Error SetTriggerTimer(CTRL_HANDLE hCtrl,uint32_t uTime)
```

Parameters

`hCtrl(CTRL_HANDLE )`

The control handle of the device

`uTime (uint32_t)`

the time value in ms

Return value

Error code:

`SDK4_ERR_SUCCESS (0)` in case of no error
`SDK4_ERR_ERROR (-1001)` in case of unspecified error
one of the other error codes defined in `SDK4_ERROR`

Remarks

At exposure mode ZELOS_ENUM_EXPOSUREMODE_RESETRESTART this function activates a cycle "software" trigger. The camera is triggered in an adjustable time slice repeatedly. The accuracy of the timer is about ±10ms.

Example

```
...
```

See also

`SDK4DevGetControl`
`CTRL_HANDLE`
`SDK4GetExposureMode`
`SDK4_ENUM_EXPOSUREMODE`
`SDK4GetTriggerTimer`

2.4.6.3 SDK4GetTriggerTimer

This function gets the current value of trigger timer.

```
SDK4_Error GetTriggerTimer(CTRL_HANDLE hCtrl,uint32_t* puTime)
```

Parameters

`hCtrl(CTRL_HANDLE )`
The control handle of the device
`puTime (uint32_t*)`
the pointer to the time value in ms

Return value

Error code:

`SDK4_ERR_SUCCESS (0)` in case of no error
`SDK4_ERR_ERROR (-1001)` in case of unspecified error
one of the other error codes defined in `SDK4_ERROR`

Example

```
...
```

See also

`SDK4DevGetControl`
`CTRL_HANDLE`
`SDK4SetTriggerTimer`

2.4.6.4 SDK4SetTriggerDelay

This function sets the trigger delay of the camera.

```
SDK4_Error SDK4SetTriggerDelay(CTRL_HANDLE hCtrl,SDK4_KEXPOSURE
           kDelay)
```

Parameters

`hCtrl(CTRL_HANDLE )`
The control handle of the device

`kDelay (SDK4_KEXPOSURE)`
the trigger delay

Return value

Error code:

`SDK4_ERR_SUCCESS (0)` in case of no error
`SDK4_ERR_ERROR (-1001)` in case of unspecified error
one of the other error codes defined in `SDK4_ERROR`

Remarks

At exposure mode ZELOS_ENUM_EXPOSUREMODE_RESETRESTART this function sets the delay between the trigger event and the beginning of the exposure. The delay effects both the hardware trigger and the software trigger.

Example

```
...
```

See also

`SDK4DevGetControl`
`CTRL_HANDLE`
`SDK4_KEXPOSURE`
`SDK4GetExposureMode`
`SDK4_ENUM_EXPOSUREMODE`
`GetTriggerDelay`

2.4.6.5 SDK4GetTriggerDelay

This function gets the current trigger delay of the camera.

```
SDK4_Error SDK4GetTriggerDelay(CTRL_HANDLE hCtrl,SDK4_KEXPOSURE*
           pkDelay)
```

Parameters

`hCtrl(CTRL_HANDLE )`
The control handle of the device

`pkDelay (SDK4_KEXPOSURE*)`
the pointer to the trigger delay

Return value

Error code:

SDK4_ERR_SUCCESS (0) in case of no error
SDK4_ERR_ERROR (-1001) in case of unspecified error
one of the other error codes defined in SDK4_ERROR

Example

```
...
```

See also

SDK4DevGetControl
CTRL_HANDLE
SDK4_KEXPOSURE
SDK4GetExposureMode
SDK4_ENUM_EXPOSUREMODE
GetTriggerDelay

2.4.7 SlowScan Functions

2.4.7.1 SDK4SetSlowScan

This function switches the Slow Scan Mode of the camera on and off.

```
SDK4_Error SDK4SetSlowScan(CTRL_HANDLE hCtrl,SDK4_ENUM_SWITCH
           uOnOff)
```

Parameters

hCtrl(CTRL_HANDLE)
The control handle of the device

uOnOff (SDK4_ENUM_SWITCH)
the slow scan switch

Return value

Error code:

SDK4_ERR_SUCCESS (0) in case of no error
SDK4_ERR_ERROR (-1001) in case of unspecified error
one of the other error codes defined in SDK4_ERROR

Remarks

Activating the slow scan feature reduces the pixel clock to increase the sensor performance. This affects other features, such as exposure time and readout time.

Example

```
CTRL_HANDLE hCtrl;
...
uint32_t uPixelClock;
uint32_t uReadoutTime;
SDK4SetSlowScan(hCtrl, SDK4_ENUM_SWITCH_ON);
SDK4GetPixelClock (hCtrl, &uPixelClock);
SDK4GetReadoutTime(hCtrl,uint32_t &uReadoutTime);
```

See also

```
SDK4DevGetControl
CTRL_HANDLE
SDK4_ENUM_SWITCH
SDK4GetExposure
SDK4GetPixelClock
SDK4GetReadoutTime
SDK4GetSlowScan
```

2.4.7.2 SDK4GetSlowScan

This function gets the current Slow Scan Mode of the camera.

```
SDK4_Error SDK4GetSlowScan(CTRL_HANDLE hCtrl,SDK4_ENUM_SWITCH*
           puOnOff)
```

Parameters

`hCtrl(CTRL_HANDLE )`

The control handle of the device

`puOnOff (SDK4_ENUM_SWITCH*)`

the pointer to the slow scan switch

Return value

Error code:

`SDK4_ERR_SUCCESS (0)` in case of no error

`SDK4_ERR_ERROR (-1001)` in case of unspecified error

one of the other error codes defined in `SDK4_ERROR`

Example

```
CTRL_HANDLE hCtrl;
...
SDK4_ENUM_SWITCH uSlowScan;
SDK4GetSlowScan(hCtrl, &uSlowScan);
```

See also

```
SDK4DevGetControl
CTRL_HANDLE
SDK4_ENUM_SWITCH
SDK4SetSlowScan
```

2.4.8 Signal Format Functions

2.4.8.1 SDK4SetColorCoding

This function sets the color coding (pixel format) of the camera.

```
SDK4_Error SDK4SetColorCoding(CTRL_HANDLE hCtrl,
           SDK4_ENUM_COLORCODING uColorCoding)
```

Parameters

`hCtrl(CTRL_HANDLE )`

The control handle of the device

`uColorCoding (SDK4_ENUM_COLORCODING)`

the color coding, specified in SDK4_ENUM_COLORCODING

Return value

Error code:

`SDK4_ERR_SUCCESS (0)` in case of no error
`SDK4_ERR_ERROR (-1001)` in case of unspecified error
one of the other error codes defined in `SDK4_ERROR`

Remarks

Notice: As changes in color coding change the image format and might thus change the size of the image buffer, image aquisition has to be stopped, before changing the color coding.
For Color cameras its possible to switch between RGB24, YUV4:2:2 and D14.
For B/W cameras its possible to switch between Y8, Y12Packet and Y14.

Example

```
DEV_HANDLE hDev;
CTRL_HANDLE hCtrl;
SDK4DevGetControl(hDev, &hCtrl);
SDK4DevStartAcquisition(hDev);
...
SDK4DevStopAcquisition(hDev);
SDK4SetColorCoding(hCtrl, SDK4_ENUM_COLORCODING_YUV422);
SDK4DevStartAcquisition(hDev);
...
```

See also

`SDK4DevGetControl`
`CTRL_HANDLE`
`SDK4_ENUM_COLORCODING`
`SDK4DevStopAcquisition`
`SDK4DevStartAcquisition`
`SDK4GetColorCoding`

2.4.8.2 SDK4GetColorCoding

This function gets the current color coding (pixel format) of the camera.

```
SDK4_Error SDK4GetColorCoding(CTRL_HANDLE hCtrl,
           SDK4_ENUM_COLORCODING* puColorCoding)
```

Parameters

`hCtrl(CTRL_HANDLE )`

The control handle of the device

`puColorCoding (SDK4_ENUM_COLORCODING*)`

the pointer to the color coding

Return value

Error code:

`SDK4_ERR_SUCCESS (0)` in case of no error
`SDK4_ERR_ERROR (-1001)` in case of unspecified error
one of the other error codes defined in `SDK4_ERROR`

Example

```
CTRL_HANDLE hCtrl;
...
SDK4_ENUM_COLORCODING uColorCoding;
SDK4GetColorCoding (hCtrl, &uColorCoding);
```

See also

`SDK4DevGetControl`
`CTRL_HANDLE`
`SDK4_ENUM_COLORCODING`
`SDK4SetColorCoding`

2.4.9 Readout Area Functions

2.4.9.1 SDK4SetReadoutArea

This function sets the Readout Area of the CCD.

```
SDK4_Error SDK4SetReadoutArea(CTRL_HANDLE hCtrl,SDK4_KAREA kRoa)
```

Parameters

`hCtrl(CTRL_HANDLE )`

The control handle of the device

`kRoa (SDK4_KAREA)`

the readout area

Return value

Error code:

`SDK4_ERR_SUCCESS (0)` in case of no error
`SDK4_ERR_ERROR (-1001)` in case of unspecified error
one of the other error codes defined in `SDK4_ERROR`

Remarks

The readout area must not be greater than the maximal readout area and also not smaller than the minimal readout area (both specified in SDK4GetReadoutAreaLimits).
Due to the fact that only a group of lines is readout the CCD timing changes. For example the CCD readout times decreases.
The ROA messages automatically reset binning to 1x1 and the frame mask to the same as ROA.
Note:
As changes in readout area change the image format and thus change the size of the image buffer, image aquisition has to be stopped, before changing the readout area.

Example

```
CTRL_HANDLE hCtrl;
...
// Get the active readout area of the CCD and set the readout area
to this value:
SDK4_KAREA kActiveRoa;
SDK4GetReadoutAreaLimits(hCtrl, SDK4_ENUM_ROADOMAIN_ACTIVE,
       &kActiveRoa);
SDK4SetReadoutArea (hCtrl, kActiveRoa);
```

See also

SDK4DevGetControl
CTRL_HANDLE
SDK4_KAREA
SDK4GetReadoutAreaLimits
SDK4GetFrameMask
SDK4GetReadoutTime
SDK4GetReadoutArea
SDK4DevStopAcquisition

2.4.9.2 SDK4GetReadoutArea

This function gets the current Readout Area of the CCD.

```
SDK4_Error SDK4GetReadoutArea(CTRL_HANDLE hCtrl,SDK4_KAREA* pkRoa)
```

Parameters

hCtrl(CTRL_HANDLE)
    The control handle of the device
pkRoa (SDK4_KAREA*)
    the pointer to the readout area

Return value

Error code:

SDK4_ERR_SUCCESS (0) in case of no error
SDK4_ERR_ERROR (-1001) in case of unspecified error
one of the other error codes defined in SDK4_ERROR

Example

```
CTRL_HANDLE hCtrl;
...
SDK4_KAREA kRoa;
SDK4GetReadoutArea(hCtrl, &kRoa);
```

See also

```
SDK4DevGetControl
CTRL_HANDLE
SDK4_KAREA
SDK4SetReadoutArea
```

2.4.9.3 SDK4SetFrameMask

This function sets the Frame Mask of the camera.

```
SDK4_Error SDK4SetFrameMask(CTRL_HANDLE hCtrl,SDK4_KAREA kRoa)
```

Parameters

`hCtrl(CTRL_HANDLE )`

  The control handle of the device

`kRoa (SDK4_KAREA)`

  the frame mask

Return value

Error code:

  `SDK4_ERR_SUCCESS (0)` in case of no error
  `SDK4_ERR_ERROR (-1001)` in case of unspecified error
  one of the other error codes defined in `SDK4_ERROR`

Remarks

The Frame mask messages are used to select an image area from cameras output signal. It does not affect the CCD timing.

The ROA messages reset the frame mask setting to maximum. Position (0,0) of the frame mask is the first ROA pixel.

The Frame mask must not be greater than the maximal readout area and also not smaller than the minimal readout area (both specified in SDK4GetReadoutAreaLimits).

Note:

As changes in frame mask change the image format and thus change the size of the image buffer, image aquisition has to be stopped, before changing the frame mask.

Example

```
CTRL_HANDLE hCtrl;
...
// Get the active readout area of the CCD and set the frame mask to
this value:
SDK4_KAREA kActiveRoa;
SDK4GetReadoutAreaLimits(hCtrl, SDK4_ENUM_ROADOMAIN_ACTIVE,
       &kActiveRoa);
SDK4SetFrameMask (hCtrl, kActiveRoa);
```

See also

SDK4DevGetControl
CTRL_HANDLE
SDK4_KAREA
SDK4GetReadoutAreaLimits
SDK4SetReadoutArea
SDK4GetFrameMask

2.4.9.4 SDK4GetFrameMask

This function gets the current Frame Mask of the camera.

```
SDK4_Error SDK4GetFrameMask(CTRL_HANDLE hCtrl,SDK4_KAREA* pkRoa)
```

Parameters

hCtrl(CTRL_HANDLE)
    The control handle of the device
pkRoa (SDK4_KAREA*)
    the pointer to the frame mask

Return value

Error code:
    SDK4_ERR_SUCCESS (0) in case of no error
    SDK4_ERR_ERROR (-1001) in case of unspecified error
    one of the other error codes defined in SDK4_ERROR

Example

```
CTRL_HANDLE hCtrl;
...
SDK4_KAREA kMask;
SDK4GetFrameMask (hCtrl, &kMask);
```

See also

SDK4DevGetControl
CTRL_HANDLE
SDK4_KAREA
SDK4SetFrameMask

2.4.9.5 SDK4GetReadoutAreaLimits

This function gets the limits of the readout area settings of the CCD.

```
SDK4_Error SDK4GetReadoutAreaLimits(CTRL_HANDLE hCtrl,
          SDK4_ENUM_ROADOMAIN uSelect, SDK4_KAREA* pkRoa)
```

Parameters

hCtrl(CTRL_HANDLE)
    The control handle of the device
uSelect(SDK4_ENUM_ROADOMAIN)
    the selected readout area domain
pkRoa (SDK4_KAREA*)
    the pointer to the readout area

Return value

Error code:

SDK4_ERR_SUCCESS (0) in case of no error
SDK4_ERR_ERROR (-1001) in case of unspecified error
one of the other error codes defined in SDK4_ERROR

Remarks

This function gives the minimal, the maximal and the active readout area of the sensor, specified in SDK4_ENUM_ROADOMAIN. The ROA limit messages are useful to find cameras limits in ROA settings. These limits depend on the CCD. For example the ROA for all active pixels could be interesting if you don't want to see any black margins around your image.

Example

```
CTRL_HANDLE hCtrl;
...
// Get the active readout area of the CCD and set the readout area
to this value:
SDK4_KAREA kActiveRoa;
SDK4GetReadoutAreaLimits(hCtrl, SDK4_ENUM_ROADOMAIN_ACTIVE,
       &kActiveRoa);
SDK4SetReadoutArea (hCtrl, kActiveRoa);
```

See also

SDK4DevGetControl
CTRL_HANDLE
SDK4_KAREA
SDK4_ENUM_ROADOMAIN

2.4.10 Binning Functions

2.4.10.1 SDK4SetBinning

This function sets the Binning in one of the modes defined in SDK4_ENUM_BINNING, whereas asymmetrical binning is possible.

```
SDK4_Error SDK4SetBinning(CTRL_HANDLE hCtrl,SDK4_KBINNING kBinning)
```

Parameters

hCtrl(CTRL_HANDLE)
The control handle of the device

kBinning (SDK4_KBINNING)
the binning value

Return value

Error code:

```
SDK4_ERR_SUCCESS (0) in case of no error
SDK4_ERR_ERROR (-1001) in case of unspecified error
one of the other error codes defined in SDK4_ERROR
```

Remarks

This function is only implemented in monochrome cameras.
Changes in Binning affect CCD timing (see ReadoutTime).
Note:
As changes in binning change the image format and thus change the size of the image buffer, image aquisition has to be stopped, before changing binning.

Example

```
DEV_HANDLE hDev;
CTRL_HANDLE hCtrl;
SDK4DevGetControl(hDev, &hCtrl);
SDK4DevStartAcquisition(hDev);
...
// set an aysymmetrical binning of 4 x 2:
SDK4DevStopAcquisition(hDev);
SDK4_KBINNING kBinning;
kBinning.horizontal = SDK4_ENUM_BINNING_4x;
kBinning.vertical = SDK4_ENUM_BINNING_2x;
SDK4SetBinning (hCtrl, kBinning);
SDK4DevStartAcquisition(hDev);
...
```

See also

SDK4DevGetControl
CTRL_HANDLE
SDK4_KBINNING
SDK4_ENUM_BINNING
SDK4GetReadoutTime
SDK4GetBinning
SDK4DevStopAcquisition

2.4.10.2 SDK4GetBinning

This function gets the current Binning Mode.

```
SDK4_Error SDK4GetBinning(CTRL_HANDLE hCtrl,SDK4_KBINNING* pkBinning)
```

Parameters

hCtrl(CTRL_HANDLE)
    The control handle of the device
pkBinning (SDK4_KBINNING*)
    the pointer to the binning value

Return value

Error code:
    SDK4_ERR_SUCCESS (0) in case of no error
    SDK4_ERR_ERROR (-1001) in case of unspecified error
    one of the other error codes defined in SDK4_ERROR

Remarks

This function is only applicable in monochrome cameras.

Example

```
CTRL_HANDLE hCtrl;
...
SDK4_KBINNING kBinning;
SDK4GetBinning (hCtrl, &kBinning);
```

See also

SDK4DevGetControl
CTRL_HANDLE
SDK4_KBINNING
SDK4_ENUM_BINNING
SDK4SetBinning

2.4.11 Signal Analysis Functions

2.4.11.1 SDK4SetMeasureFrameShow

This function shows or hides the outline frame of the measure windows.

```
SDK4_Error SDK4SetMeasureFrameShow(CTRL_HANDLE hCtrl,
          SDK4_ENUM_SWITCH uShow)
```

Parameters

hCtrl(CTRL_HANDLE)

The control handle of the device

uShow (SDK4_ENUM_SWITCH)

the display switch

Return value

Error code:

SDK4_ERR_SUCCESS (0) in case of no error
SDK4_ERR_ERROR (-1001) in case of unspecified error
one of the other error codes defined in SDK4_ERROR

Remarks

The switch effects both implemented measure windows.

Example

```
CTRL_HANDLE hCtrl;
...
SDK4SetMeasureFrameShow (hCtrl, SDK4_ENUM_SWITCH_ON);
```

See also

SDK4DevGetControl
CTRL_HANDLE
SDK4_ENUM_SWITCH
SDK4GetMeasureFrameShow

2.4.11.2 SDK4GetMeasureFrameShow

This function gets the current display state of the measure windows.

```
SDK4_Error SDK4GetMeasureFrameShow(CTRL_HANDLE
hCtrl,SDK4_ENUM_SWITCH* puShow)
```

Parameters

`hCtrl(CTRL_HANDLE )`

  The control handle of the device

`puShow (SDK4_ENUM_SWITCH*)`

  the pointer fo the display switch

Return value

Error code:

  `SDK4_ERR_SUCCESS (0)` in case of no error

  `SDK4_ERR_ERROR (-1001)` in case of unspecified error

  one of the other error codes defined in `SDK4_ERROR`

Example

```
CTRL_HANDLE hCtrl;
...
SDK4_ENUM_SWITCH uShow;
SDK4GetMeasureFrameShow (hCtrl, &uShow);
```

See also

`SDK4DevGetControl`

`CTRL_HANDLE`

`SDK4_ENUM_SWITCH`

`SDK4SetMeasureFrameShow`

2.4.11.3 SDK4SetMeasureFrameColor

This function sets the color of the outline frame of the measure windows.

```
SDK4_Error SDK4SetMeasureFrameColor(CTRL_HANDLE hCtrl,
          SDK4_ENUM_MFCOLOR uColor)
```

Parameters

`hCtrl(CTRL_HANDLE )`

  The control handle of the device

`uColor (SDK4_ENUM_MFCOLOR)`

  the frame color

Return value

Error code:

  `SDK4_ERR_SUCCESS (0)` in case of no error

  `SDK4_ERR_ERROR (-1001)` in case of unspecified error

  one of the other error codes defined in `SDK4_ERROR`

Remarks

The switch effects both implemented measure windows.

Example

```
CTRL_HANDLE hCtrl;
...
SDK4SetMeasureFrameColor (hCtrl, ZELOS_ENUM_MFCOLOR_C_RED);
```

See also

SDK4DevGetControl
CTRL_HANDLE
SDK4_ENUM_MFCOLOR
SDK4SetMeasureFrameShow

2.4.11.4 SDK4GetMeasureFrameColor

This function gets the current color of the outline frame of the measure windows.

```
SDK4_Error SDK4GetMeasureFrameColor(CTRL_HANDLE hCtrl,
          SDK4_ENUM_MFCOLOR* puColor)
```

Parameters

hCtrl(CTRL_HANDLE)
The control handle of the device

puColor (SDK4_ENUM_MFCOLOR*)
the pointer to the frame color

Return value

Error code:

SDK4_ERR_SUCCESS (0) in case of no error
SDK4_ERR_ERROR (-1001) in case of unspecified error
one of the other error codes defined in SDK4_ERROR

Example

```
CTRL_HANDLE hCtrl;
...
SDK4_ENUM_MFCOLOR uColor;
SDK4GetMeasureFrameColor (hCtrl, &uColor);
```

See also

SDK4DevGetControl
CTRL_HANDLE
SDK4_ENUM_MFCOLOR
SDK4SetMeasureFrameShow

2.4.11.5 SDK4SetMeasureWindow

This function sets the position and size of the selected measure window for AET and AGC.

```
SDK4_Error SDK4SetMeasureWindow(CTRL_HANDLE hCtrl,
           SDK4_ENUM_MEASUREWINDOWSELECT uSelect, SDK4_KAREA kArea)
```

Parameters

`hCtrl(CTRL_HANDLE )`

The control handle of the device

`uSelect (SDK4_ENUM_MEASUREWINDOWSELECT)`

the selected measure window, specified in `SDK4_ENUM_MEASUREWINDOWSELECT`

`kArea (SDK4_KAREA)`

the measure window area

Return value

Error code:

`SDK4_ERR_SUCCESS (0)` in case of no error

`SDK4_ERR_ERROR (-1001)` in case of unspecified error

one of the other error codes defined in `SDK4_ERROR`

Remarks

This function sets start and end point of the measure window. Position 0/0 is the upper left corner. The settings will be limited to CCD size. Overlapping measure windows are allowed.

Example

```
CTRL_HANDLE hCtrl;
...
// Set measure window 2 to a size of 100 x 100, beginning at 10/10:
SDK4_KAREA kArea;
kArea.start.xPos = 10;
kArea.start.yPos = 10;
kArea.end.xPos = 110;
kArea.end.yPos = 110;
SDK4SetMeasureWindow (hCtrl, SDK4_ENUM_MEASUREWINDOW2, kArea);
```

See also

`SDK4DevGetControl`
`CTRL_HANDLE`
`SDK4_ENUM_MEASUREWINDOWSELECT`
`SDK4_KAREA`
`SDK4GetMeasureWindow`

2.4.11.6 SDK4GetMeasureWindow

This function gets the current position and size of the selected measure window.

```
SDK4_Error SDK4GetMeasureWindow(CTRL_HANDLE hCtrl,
           SDK4_ENUM_MEASUREWINDOWSELECT uSelect, SDK4_KAREA*
           pkArea)
```

Parameters

`hCtrl(CTRL_HANDLE )`

The control handle of the device

`uSelect (SDK4_ENUM_MEASUREWINDOWSELECT)`

the selected measure window, specified in `SDK4_ENUM_MEASUREWINDOWSELECT`

`pkArea (SDK4_KAREA*)`

`the pointer to the measure window area`

Return value

Error code:

`SDK4_ERR_SUCCESS (0)` in case of no error

`SDK4_ERR_ERROR (-1001)` in case of unspecified error

one of the other error codes defined in `SDK4_ERROR`

Example

```
CTRL_HANDLE hCtrl;
...
// get position and size of measure window 1:
SDK4_KAREA kArea;
SDK4GetMeasureWindow (hCtrl, SDK4_ENUM_MEASUREWINDOW1, &kArea);
```

See also

`SDK4DevGetControl`
`CTRL_HANDLE`
`SDK4_ENUM_MEASUREWINDOWSELECT`
`SDK4_KAREA`
`SDK4SetMeasureWindow`

2.4.11.7 SDK4GetMeasureValue

This function gets the current measure values for pixel saturation.

```
SDK4_Error SDK4GetMeasureValue(CTRL_HANDLE
           hCtrl,SDK4_ENUM_MEASUREVALUE uSelect,uint32_t* puValue)
```

Parameters

`hCtrl(CTRL_HANDLE )`

The control handle of the device

`uSelect (SDK4_ENUM_MEASUREVALUE)`

the selected measure attribute, specified in `SDK4_ENUM_MEASUREVALUE`

`puValue (uint32_t*)`

the pointer to the measure value

Return value

Error code:

SDK4_ERR_SUCCESS (0) in case of no error
SDK4_ERR_ERROR (-1001) in case of unspecified error
one of the other error codes defined in SDK4_ERROR

Remarks

This measure values are calculated within the two measure windows. Depending on the selection, the minimal, mean or maximal measure value is given (0-255, 255 means 100%).

Example

```
CTRL_HANDLE hCtrl;
...
// get the mean pixel saturation inside the measure windows:
uint32_t uValue;
SDK4GetMeasureValue(hCtrl, SDK4_ENUM_MEASUREVALUE_MEAN, &uValue);
```

See also

SDK4DevGetControl
CTRL_HANDLE
SDK4_ENUM_MEASUREVALUE

2.4.12 Overlay Functions

2.4.12.1 SDK4SetOverlayShow

This function is used to display or hide the selected line generator or circle generator.

```
SDK4_Error SDK4SetOverlayShow(CTRL_HANDLE hCtrl,
           SDK4_ENUM_OVERLAYSELECT uSelect, SDK4_ENUM_SWITCH uShow)
```

Parameters

hCtrl(CTRL_HANDLE)
The control handle of the device

uSelect (SDK4_ENUM_OVERLAYSELECT)
the selected overlay generator, specified in SDK4_ENUM_OVERLAYSELECT

uShow (SDK4_ENUM_SWITCH)
the switch value

Return value

Error code:

SDK4_ERR_SUCCESS (0) in case of no error
SDK4_ERR_ERROR (-1001) in case of unspecified error
one of the other error codes defined in SDK4_ERROR

Remarks

There are two line generators and two circle generators, which can be displayed.

Example

```
CTRL_HANDLE hCtrl;
...
// show overlay circle 1:
SDK4SetOverlayShow(hCtrl, SDK4_ENUM_OVERLAYSELECT_CIRCLEGENERATOR1,
        SDK4_ENUM_SWITCH_ON);
```

See also

SDK4DevGetControl
CTRL_HANDLE
SDK4_ENUM_OVERLAYSELECT
SDK4_ENUM_SWITCH
SDK4GetOverlayShow

2.4.12.2 SDK4GetOverlayShow

This function gets the display switch of the selected line generator or circle generator.

```
SDK4_Error SDK4GetOverlayShow(CTRL_HANDLE hCtrl,
           SDK4_ENUM_OVERLAYSELECT uSelect, SDK4_ENUM_SWITCH*
           puShow)
```

Parameters

hCtrl(CTRL_HANDLE)
    The control handle of the device

uSelect (SDK4_ENUM_OVERLAYSELECT)
    the selected overlay generator, specified in SDK4_ENUM_OVERLAYSELECT

puShow (SDK4_ENUM_SWITCH*)
    the pointer to the switch value

Return value

Error code:

SDK4_ERR_SUCCESS (0) in case of no error
SDK4_ERR_ERROR (-1001) in case of unspecified error
one of the other error codes defined in SDK4_ERROR

Example

```
CTRL_HANDLE hCtrl;
...
SDK4_ENUM_SWITCH uShowCircle1;
SDK4GetOverlayShow(hCtrl, SDK4_ENUM_OVERLAYSELECT_CIRCLEGENERATOR1,
        &uShowCircle1);
```

See also

SDK4DevGetControl
CTRL_HANDLE
SDK4_ENUM_OVERLAYSELECT
SDK4_ENUM_SWITCH
SDK4GetOverlayShow

2.4.12.3 SDK4SetOverlayPosition

This function sets the position of the selected line generator or circle generator.

```
SDK4_Error SDK4SetOverlayPosition(CTRL_HANDLE hCtrl,
           SDK4_ENUM_OVERLAYSELECT uSelect, SDK4_KPOSITION
           kPosition)
```

Parameters

`hCtrl(CTRL_HANDLE )`

The control handle of the device

`uSelect (SDK4_ENUM_OVERLAYSELECT)`

the selected overlay generator, specified in `SDK4_ENUM_OVERLAYSELECT`

`kPosition (SDK4_KPOSITION)`

the overlay position

Return value

Error code:

`SDK4_ERR_SUCCESS (0)` in case of no error

`SDK4_ERR_ERROR (-1001)` in case of unspecified error

one of the other error codes defined in `SDK4_ERROR`

Remarks

The position for each overlay generator is set with respect to the upper left corner of the image.

Example

```
CTRL_HANDLE hCtrl;
...
// set position of overlay circle 2 to 500/500:
SDK4_KPOSITION kPosition;
kPosition.xPos = 500;
kPosition.yPos = 500;
SDK4SetOverlayPosition (hCtrl,
        SDK4_ENUM_OVERLAYSELECT_CIRCLEGENERATOR2, kPosition);
```

See also

```
SDK4DevGetControl
CTRL_HANDLE
SDK4_ENUM_OVERLAYSELECT
SDK4_KPOSITION
SDK4SetOverlayShow
SDK4GetOverlayPosition
```

2.4.12.4 SDK4GetOverlayPosition

This function gets the current position of the selected line generator or circle generator.

```
SDK4_Error SDK4GetOverlayPosition(CTRL_HANDLE hCtrl,
           SDK4_ENUM_OVERLAYSELECT uSelect, SDK4_KPOSITION*
           pkPosition)
```

Parameters

`hCtrl(CTRL_HANDLE )`

The control handle of the device

`uSelect (SDK4_ENUM_OVERLAYSELECT)`

the selected overlay generator, specified in `SDK4_ENUM_OVERLAYSELECT`

`pkPosition (SDK4_KPOSITION*)`

the pointer to the overlay position

Return value

Error code:

`SDK4_ERR_SUCCESS (0)` in case of no error

`SDK4_ERR_ERROR (-1001)` in case of unspecified error

one of the other error codes defined in `SDK4_ERROR`

Example

```
CTRL_HANDLE hCtrl;
...
SDK4_KPOSITION kPosition;
SDK4GetOverlayPosition (hCtrl,
       SDK4_ENUM_OVERLAYSELECT_CIRCLEGENERATOR2, &kPosition);
```

See also

`SDK4DevGetControl`
`CTRL_HANDLE`
`SDK4_ENUM_OVERLAYSELECT`
`SDK4_KPOSITION`
`SDK4SetOverlayShow`
`SDK4SetOverlayPosition`

2.4.12.5 SDK4SetOverlayStyle

This function sets the style of the selected line generator or circle generator.

```
SDK4_Error SDK4SetOverlayStyle(CTRL_HANDLE hCtrl,
           SDK4_ENUM_OVERLAYSELECT uSelect, SDK4_KSTYLE kStyle)
```

Parameters

`hCtrl(CTRL_HANDLE )`

  The control handle of the device

`uSelect (SDK4_ENUM_OVERLAYSELECT)`

  the selected overlay generator, specified in `SDK4_ENUM_OVERLAYSELECT`

`kStyle (SDK4_KSTYLE)`

  the overlay style

Return value

Error code:

  `SDK4_ERR_SUCCESS (0)` in case of no error

  `SDK4_ERR_ERROR (-1001)` in case of unspecified error

  one of the other error codes defined in `SDK4_ERROR`

Remarks

The color of the overlay generators is specified in SDK4_ENUM_MFCOLOR, the line style is specified in SDK4_ENUM_LINESTYLE.

Example

```
CTRL_HANDLE hCtrl;
...
// set the style of overlay line 1 to magenta and dashed:
SDK4_KSTYLE kStyle;
kStyle.color = ZELOS_ENUM_MFCOLOR_C_MAGENTA;
kStyle.style = SDK4_ENUM_LINESTYLE_DASHED;
SDK4SetOverlayStyle (hCtrl,
       SDK4_ENUM_OVERLAYSELECT_LINEGENERATOR1_ALLLINES, kStyle);
```

See also

`SDK4DevGetControl`
`CTRL_HANDLE`
`SDK4_ENUM_OVERLAYSELECT`
`SDK4_KSTYLE`
`SDK4_ENUM_MFCOLOR`
`SDK4_ENUM_LINESTYLE`
`SDK4SetOverlayShow`
`SDK4GetOverlayStyle`

2.4.12.6 SDK4GetOverlayStyle

This function gets the current style of the selected line generator or circle generator.

```
SDK4_Error SDK4GetOverlayStyle(CTRL_HANDLE hCtrl,
           SDK4_ENUM_OVERLAYSELECT uSelect, SDK4_KSTYLE* pkStyle)
```

Parameters

`hCtrl(CTRL_HANDLE )`

The control handle of the device

`uSelect (SDK4_ENUM_OVERLAYSELECT)`

the selected overlay generator, specified in `SDK4_ENUM_OVERLAYSELECT`

`pkStyle (SDK4_KSTYLE*)`

the pointer to the overlay style

Return value

Error code:

`SDK4_ERR_SUCCESS (0)` in case of no error

`SDK4_ERR_ERROR (-1001)` in case of unspecified error

one of the other error codes defined in `SDK4_ERROR`

Remarks

The color of the overlay generators is specified in SDK4_ENUM_MFCOLOR, the line style is specified in SDK4_ENUM_LINESTYLE.

Example

```
CTRL_HANDLE hCtrl;
...
SDK4_KSTYLE kStyle;
SDK4GetOverlayStyle (hCtrl,
       SDK4_ENUM_OVERLAYSELECT_LINEGENERATOR1_ALLLINES, &kStyle);
```

See also

`SDK4DevGetControl`
`CTRL_HANDLE`
`SDK4_ENUM_OVERLAYSELECT`
`SDK4_KSTYLE`
`SDK4_ENUM_MFCOLOR`
`SDK4_ENUM_LINESTYLE`
`SDK4SetOverlayShow`
`SDK4GetOverlayStyle`

2.4.12.7 SDK4SetOverlaySize

This function sets the size of the selected generator(s).

```
SDK4_Error SDK4SetOverlaySize(CTRL_HANDLE hCtrl,
           SDK4_ENUM_OVERLAYSELECT uSelect, SDK4_KOVERLAYSIZE kSize)
```

Parameters

`hCtrl(CTRL_HANDLE )`

The control handle of the device

`uSelect (SDK4_ENUM_OVERLAYSELECT)`

the selected overlay generator, specified in `SDK4_ENUM_OVERLAYSELECT`

`kSize (SDK4_KOVERLAYSIZE)`

the overlay size, specified in SDK4_KOVERLAYSIZE

Return value

Error code:

`SDK4_ERR_SUCCESS (0)` in case of no error

`SDK4_ERR_ERROR (-1001)` in case of unspecified error

one of the other error codes defined in `SDK4_ERROR`

Remarks

Note:

For changes in overlay size only SDK4_ENUM_OVERLAYSELECT_LINEGENERATOR1_ALLLINES (changes in size apply to all lines), SDK4_ENUM_OVERLAYSELECT_CIRCLEGENERATOR1 or SDK4_ENUM_OVERLAYSELECT_CIRCLEGENERATOR2 can be selected.

The width attribute of SDK4_KOVERLAYSIZE is specified in SDK4_ENUM_OVERLAYWIDTH. The length attribute is defined as the length of a line, beginning in the center of the reticle, or as the radius of a circle.

Example

```
CTRL_HANDLE hCtrl;
// set the size of both overlay lines to a width of 5 pixels and a
length of 100 pixels:
// set the size of overlay circle 1 to a width of 3 pixels and a
radius of 30 pixels:
SDK4_KOVERLAYSIZE kSize;
kSize.width = SDK4_ENUM_OVERLAYWIDTH_PIXEL5;
kSize.length = 100;
SDK4SetOverlaySize (hCtrl,
       SDK4_ENUM_OVERLAYSELECT_LINEGENERATOR1_ALLLINES, kSize);
kSize.width = SDK4_ENUM_OVERLAYWIDTH_PIXEL3;
kSize.length = 30;
SDK4SetOverlaySize (hCtrl, SDK4_ENUM_OVERLAYSELECT_CIRCLEGENERATOR1,
       kSize);
```

See also

```
SDK4DevGetControl
CTRL_HANDLE
SDK4_ENUM_OVERLAYSELECT
SDK4_KOVERLAYSIZE
SDK4_ENUM_OVERLAYWIDTH
SDK4SetOverlayShow
SDK4GetOverlaySize
```

2.4.12.8 SDK4GetOverlaySize

This function gets the current size of the selected generator(s).

```
SDK4_Error SDK4GetOverlaySize(CTRL_HANDLE hCtrl,
           SDK4_ENUM_OVERLAYSELECT uSelect, SDK4_KOVERLAYSIZE*
           pkSize)
```

Parameters

`hCtrl(CTRL_HANDLE )`

The control handle of the device

`uSelect (SDK4_ENUM_OVERLAYSELECT)`

the selected overlay generator, specified in `SDK4_ENUM_OVERLAYSELECT`

`kSize (SDK4_KOVERLAYSIZE)`

the overlay size, specified in SDK4_KOVERLAYSIZE

Return value

```
Error code:
     SDK4_ERR_SUCCESS (0) in case of no error
     SDK4_ERR_ERROR (-1001) in case of unspecified error
     one of the other error codes defined in SDK4_ERROR
```

Remarks

Note:

For changes in overlay size only SDK4_ENUM_OVERLAYSELECT_LINEGENERATOR1_ALLLINES (changes in size apply to all lines), SDK4_ENUM_OVERLAYSELECT_CIRCLEGENERATOR1 or SDK4_ENUM_OVERLAYSELECT_CIRCLEGENERATOR2 can be selected.

Example

```
CTRL_HANDLE hCtrl;
...
// get the size of overlay circle 2:
SDK4_KOVERLAYSIZE kSize;
SDK4GetOverlaySize (hCtrl, SDK4_ENUM_OVERLAYSELECT_CIRCLEGENERATOR2,
       &kSize);
```

See also

```
SDK4DevGetControl
CTRL_HANDLE
SDK4_ENUM_OVERLAYSELECT
SDK4_KOVERLAYSIZE
SDK4_ENUM_OVERLAYWIDTH
SDK4SetOverlayShow
SDK4SetOverlaySize
```

2.4.13 Digital Enhancement Functions

2.4.13.1 SDK4SetBrightness

This function sets the Brightness (digital level offset).

```
SDK4_Error SDK4SetBrightness(CTRL_HANDLE hCtrl,uint32_t uBrightness)
```

Parameters

`hCtrl(CTRL_HANDLE )`

The control handle of the device

`uBrightness (uint32_t)`

the brightness value

Return value

Error code:

`SDK4_ERR_SUCCESS (0)` in case of no error
`SDK4_ERR_ERROR (-1001)` in case of unspecified error
one of the other error codes defined in `SDK4_ERROR`

Remarks

The values range from 0 to 4095.

Example

```
CTRL_HANDLE hCtrl;
...
SDK4SetBrightness (hCtrl, 2048);
```

See also

`SDK4DevGetControl`
`CTRL_HANDLE`
`SDK4GetBrightness`

2.4.13.2 SDK4GetBrightness

This function gets the current value of Brightness (digital level offset).

```
SDK4_Error SDK4GetBrightness(CTRL_HANDLE hCtrl, uint32_t*
          puBrightness)
```

Parameters

`hCtrl(CTRL_HANDLE )`

The control handle of the device

`puBrightness (uint32_t*)`

the pointer to the brightness value

Return value

Error code:

`SDK4_ERR_SUCCESS (0)` in case of no error
`SDK4_ERR_ERROR (-1001)` in case of unspecified error
one of the other error codes defined in `SDK4_ERROR`

Remarks

The values range from 0 to 4095.

Example

```
CTRL_HANDLE hCtrl;
...
uint32_t uBrightness
SDK4GetBrightness (hCtrl, &uBrightness);
```

See also

```
SDK4DevGetControl
CTRL_HANDLE
SDK4SetBrightness
```

2.4.13.3 SDK4SetContrast

This function sets the Contrast (digital level gain).

```
SDK4_Error SDK4SetContrast(CTRL_HANDLE hCtrl,uint32_t uContrast)
```

Parameters

`hCtrl(CTRL_HANDLE )`

The control handle of the device

`uContrast (uint32_t)`

the contrast value

Return value

Error code:

`SDK4_ERR_SUCCESS (0)` in case of no error

`SDK4_ERR_ERROR (-1001)` in case of unspecified error

one of the other error codes defined in `SDK4_ERROR`

Remarks

The values range from 0 to 255.

Example

```
CTRL_HANDLE hCtrl;
...
SDK4SetContrast (hCtrl, 127);
```

See also

```
SDK4DevGetControl
CTRL_HANDLE
SDK4GetContrast
```

2.4.13.4 SDK4GetContrast

This function gets the current value of Contrast (digital level gain).

```
SDK4_Error SDK4GetContrast(CTRL_HANDLE hCtrl,uint32_t* puContrast)
```

Parameters

`hCtrl(CTRL_HANDLE )`

The control handle of the device

`puContrast (uint32_t*)`

the pointer to the contrast value

Return value

Error code:

`SDK4_ERR_SUCCESS (0)` in case of no error

`SDK4_ERR_ERROR (-1001)` in case of unspecified error

one of the other error codes defined in `SDK4_ERROR`

Remarks

The values range from 0 to 255.

Example

```
CTRL_HANDLE hCtrl;
...
uint32_t uContrast
SDK4GetContrast (hCtrl, &uContrast);
```

See also

`SDK4DevGetControl`

`CTRL_HANDLE`

`SDK4SetContrast`

2.4.13.5 SDK4SetSharpness

This function sets the Sharpness (digital edge enhancement).

```
SDK4_Error SDK4SetSharpness(CTRL_HANDLE hCtrl,uint32_t uSharpness)
```

Parameters

`hCtrl(CTRL_HANDLE )`

The control handle of the device

`uSharpness (uint32_t)`

the sharpness value

Return value

Error code:

`SDK4_ERR_SUCCESS (0)` in case of no error

`SDK4_ERR_ERROR (-1001)` in case of unspecified error

one of the other error codes defined in `SDK4_ERROR`

Remarks

The values range from 0 to 255.

Example

```
CTRL_HANDLE hCtrl;
...
SDK4SetSharpness (hCtrl, 255);
```

See also

```
SDK4DevGetControl
CTRL_HANDLE
SDK4GetSharpness
```

2.4.13.6 SDK4GetSharpness

This function gets the current value of Sharpness (digital edge enhancement).

```
SDK4_Error SDK4GetSharpness(CTRL_HANDLE hCtrl,uint32_t* puSharpness)
```

Parameters

`hCtrl(CTRL_HANDLE )`

The control handle of the device

`puSharpness (uint32_t*)`

the pointer to the sharpness value

Return value

Error code:

`SDK4_ERR_SUCCESS (0)` in case of no error

`SDK4_ERR_ERROR (-1001)` in case of unspecified error

one of the other error codes defined in `SDK4_ERROR`

Remarks

The values range from 0 to 255.

Example

```
CTRL_HANDLE hCtrl;
...
uint32_t uSharpness
SDK4GetSharpness (hCtrl, &uSharpness);
```

See also

```
SDK4DevGetControl
CTRL_HANDLE
SDK4SetSharpness
```

2.4.13.7 SDK4SetACC

This function sets the mode of Automatic Contrast Control.

```
SDK4_Error SDK4SetACC(CTRL_HANDLE hCtrl,SDK4_ENUM_ACC uACC)
```

Parameters

`hCtrl(CTRL_HANDLE )`

The control handle of the device

`uACC (uint32_t)`

the ACC mode

Return value

Error code:

`SDK4_ERR_SUCCESS (0)` in case of no error
`SDK4_ERR_ERROR (-1001)` in case of unspecified error
one of the other error codes defined in `SDK4_ERROR`

Remarks

The camera contains two different auto contrast control processing algorithms:
Minimum/maximum histogram stretching and Histogram equalization.
In SDK4_ENUM_ACC_PUSHDRE mode the image is evaluated once and a histogram stretching is performed once, before it automatically changes back to SDK4_ENUM_ACC_PUSHDRE mode.
SDK4_ENUM_ACC_AUTODRE modes performs a continuous histogram stretching.
In SDK4_ENUM_ACC_HISTOGRAM mode the camera evaluates the image once and adjusts the histogram equalization. The camera will stay in this ACC mode to keep the adjusted histogram settings.
In SDK4_ENUM_ACC_AUTOHISTOGRAM mode a continuous adjustment of the histogram equalization is performed.

Example

```
CTRL_HANDLE hCtrl;
...
// start Auto DRE:
SDK4SetACC (hCtrl, SDK4_ENUM_ACC_AUTODRE);
```

See also

`SDK4DevGetControl`
`CTRL_HANDLE`
`SDK4_ENUM_ACC`
`SDK4GetACC`

2.4.13.8 SDK4GetACC

This function gets the actual mode of Automatic Contrast Control.

```
SDK4_Error SDK4GetACC(CTRL_HANDLE hCtrl,SDK4_ENUM_ACC* puACC)
```

Parameters

`hCtrl(CTRL_HANDLE )`
    The control handle of the device
`puACC (uint32_t)`
    the pointer to the ACC mode

Return value

Error code:
    `SDK4_ERR_SUCCESS (0)` in case of no error
    `SDK4_ERR_ERROR (-1001)` in case of unspecified error
    one of the other error codes defined in `SDK4_ERROR`

Example

```
CTRL_HANDLE hCtrl;
...
SDK4_ENUM_ACC uACC;
SDK4GetACC (hCtrl, &uACC);
```

See also

`SDK4DevGetControl`
`CTRL_HANDLE`
`SDK4_ENUM_ACC`
`SDK4GetACC`

2.4.14 Memory Functions

2.4.14.1 SDK4SetVerticalFlip

This function can flip the camera image vertically.

```
SDK4_Error SDK4SetVerticalFlip(CTRL_HANDLE hCtrl, SDK4_ENUM_SWITCH
          uOnOff)
```

Parameters

`hCtrl(CTRL_HANDLE )`
    The control handle of the device
`uOnOff (uint32_t)`
    the flip switch

Return value

Error code:
    `SDK4_ERR_SUCCESS (0)` in case of no error
    `SDK4_ERR_ERROR (-1001)` in case of unspecified error
    one of the other error codes defined in `SDK4_ERROR`

Remarks

The position of the measure windows and generated lines are independent from the vertical flip function.

Example

```
CTRL_HANDLE hCtrl;
...
SDK4SetVerticalFlip (hCtrl, SDK4_ENUM_SWITCH_ON);
```

See also

SDK4DevGetControl
CTRL_HANDLE
SDK4GetVerticalFlip
SDK4_ENUM_SWITCH

2.4.14.2 SDK4GetVerticalFlip

This function gets the current value of the vertical flip.

```
SDK4_Error SDK4GetVerticalFlip(CTRL_HANDLE hCtrl, SDK4_ENUM_SWITCH*
          puOnOff)
```

Parameters

hCtrl(CTRL_HANDLE)
    The control handle of the device

puOnOff (uint32_t*)
    the pointer to the flip switch

Return value

Error code:

SDK4_ERR_SUCCESS (0) in case of no error
SDK4_ERR_ERROR (-1001) in case of unspecified error
one of the other error codes defined in SDK4_ERROR

Example

```
CTRL_HANDLE hCtrl;
...
SDK4_ENUM_SWITCH uFlip;
SDK4GetVerticalFlip (hCtrl, &uFlip);
```

See also

SDK4DevGetControl
CTRL_HANDLE
SDK4SetVerticalFlip
SDK4_ENUM_SWITCH

2.4.14.3 SDK4SetHorizontalFlip

This function can flip the camera image horizontally.

```
SDK4_Error SDK4SetHorizontalFlip(CTRL_HANDLE hCtrl, SDK4_ENUM_SWITCH
          uOnOff)
```

Parameters

hCtrl(CTRL_HANDLE)

The control handle of the device

uOnOff (uint32_t)

the flip switch

Return value

Error code:

SDK4_ERR_SUCCESS (0) in case of no error

SDK4_ERR_ERROR (-1001) in case of unspecified error

one of the other error codes defined in SDK4_ERROR

Remarks

The position of the measure windows and generated lines are independent from the horizontal flip function.

Example

```
CTRL_HANDLE hCtrl;
...
SDK4SetHorizontalFlip (hCtrl, SDK4_ENUM_SWITCH_ON);
```

See also

SDK4DevGetControl
CTRL_HANDLE
SDK4GetHorizontalFlip
SDK4_ENUM_SWITCH

2.4.14.4 SDK4GetHorizontalFlip

This function gets the current value of the horizontal flip.

```
SDK4_Error SDK4GetHorizontalFlip(CTRL_HANDLE hCtrl,
          SDK4_ENUM_SWITCH* puOnOff)
```

Parameters

hCtrl(CTRL_HANDLE)

The control handle of the device

puOnOff (uint32_t*)

the pointer to the flip switch

Return value

Error code:

SDK4_ERR_SUCCESS (0) in case of no error

SDK4_ERR_ERROR (-1001) in case of unspecified error

one of the other error codes defined in SDK4_ERROR

Example

```
CTRL_HANDLE hCtrl;
...
SDK4_ENUM_SWITCH uFlip;
SDK4GetHorizontalFlip (hCtrl, &uFlip);
```

See also

`SDK4DevGetControl`
`CTRL_HANDLE`
`SDK4SetHorizontalFlip`
`SDK4_ENUM_SWITCH`

2.4.14.5 SDK4SetLastImageHold

This function can freeze the current camera image.

```
SDK4_Error SDK4SetLastImageHold(CTRL_HANDLE hCtrl,
          SDK4_ENUM_IMAGEHOLD uHold)
```

Parameters

`hCtrl(CTRL_HANDLE )`

The control handle of the device

`uHold (SDK4_ENUM_IMAGEHOLD)`

the image hold value

Return value

Error code:

`SDK4_ERR_SUCCESS (0)` in case of no error
`SDK4_ERR_ERROR (-1001)` in case of unspecified error
one of the other error codes defined in `SDK4_ERROR`

Remarks

The last image hold message can be used to freeze the current camera image (SDK4_ENUM_IMAGEHOLD_HOLD). The same image will be transmitted by the camera until switching back to live by setting SDK4_ENUM_IMAGEHOLD_LIVE. The hold image has the same timing as the live image.

Example

```
CTRL_HANDLE hCtrl;
...
// freeze the image:
SDK4SetLastImageHold (hCtrl, SDK4_ENUM_IMAGEHOLD_HOLD);
```

See also

`SDK4DevGetControl`
`CTRL_HANDLE`
`SDK4_ENUM_IMAGEHOLD`
`SDK4GetLastImageHold`

2.4.14.6 SDK4GetLastImageHold

This function gets the current value of last image hold.

```
SDK4_Error SDK4GetLastImageHold(CTRL_HANDLE hCtrl,
          SDK4_ENUM_IMAGEHOLD * puHold)
```

Parameters

`hCtrl(CTRL_HANDLE )`

The control handle of the device

`puHold (SDK4_ENUM_IMAGEHOLD *)`

the pointer to the image hold value

Return value

Error code:

`SDK4_ERR_SUCCESS (0)` in case of no error

`SDK4_ERR_ERROR (-1001)` in case of unspecified error

one of the other error codes defined in `SDK4_ERROR`

Example

```
CTRL_HANDLE hCtrl;
...
SDK4_ENUM_IMAGEHOLD uHold;
SDK4GetLastImageHold (hCtrl, &uHold);
```

See also

`SDK4DevGetControl`
`CTRL_HANDLE`
`SDK4_ENUM_SWITCH`
`SDK4SetLastImageHold`

2.4.14.7 SDK4StoreImage

This function can store the current camera image to a volatile memory, when the switch is set to 1.

```
SDK4_Error SDK4StoreImage(CTRL_HANDLE hCtrl,SDK4_ENUM_SWITCH uOnOff)
```

Parameters

`hCtrl(CTRL_HANDLE )`

The control handle of the device

`uOnOff (uint32_t)`

the store switch

Return value

Error code:

`SDK4_ERR_SUCCESS (0)` in case of no error

`SDK4_ERR_ERROR (-1001)` in case of unspecified error

one of the other error codes defined in `SDK4_ERROR`

Remarks

The stored image can be used to be subtracted from the live image. Please refer to SDK4SetImageSubtraction.

Example

```
CTRL_HANDLE hCtrl;
...
// store one image and subtract it from the live image:
SDK4StoreImage(hCtrl, SDK4_ENUM_SWITCH_ON);
SDK4StoreImage(hCtrl, SDK4_ENUM_SWITCH_OFF);
...
SDK4SetImageSubtraction(hCtrl, SDK4_ENUM_SWITCH_ON);
```

See also

`SDK4DevGetControl`
`CTRL_HANDLE`
`SDK4_ENUM_SWITCH`
`SDK4SetImageSubtraction`

2.4.14.8 SDK4SetImageSubtraction

This function controls the image subtraction.

```
SDK4_Error SDK4SetImageSubtraction(CTRL_HANDLE hCtrl,
          SDK4_ENUM_SWITCH uOnOff)
```

Parameters

`hCtrl(CTRL_HANDLE )`
    The control handle of the device
`uOnOff (uint32_t)`
    the image subtraction switch

Return value

Error code:

- `SDK4_ERR_SUCCESS (0)` in case of no error
- `SDK4_ERR_ERROR (-1001)` in case of unspecified error
- one of the other error codes defined in `SDK4_ERROR`

Remarks

The image previously stored to the camera can be subtracted from the live image.

Example

```
CTRL_HANDLE hCtrl;
...
// store one image and subtract it from the live image:
SDK4StoreImage(hCtrl, SDK4_ENUM_SWITCH_ON);
SDK4StoreImage(hCtrl, SDK4_ENUM_SWITCH_OFF);
...
SDK4SetImageSubtraction(hCtrl, SDK4_ENUM_SWITCH_ON);
```

See also

`SDK4DevGetControl`
`CTRL_HANDLE`
`SDK4_ENUM_SWITCH`
`SDK4StoreImage`
`SDK4GetImageSubtraction`

2.4.14.9 SDK4GetImageSubtraction

This function gets the current value of the image subtraction.

```
SDK4_Error SDK4GetImageSubtraction(CTRL_HANDLE hCtrl,
          SDK4_ENUM_SWITCH* puOnOff)
```

Parameters

`hCtrl(CTRL_HANDLE )`

The control handle of the device

`puOnOff (uint32_t)`

the pointer to the image subtraction switch

Return value

Error code:

`SDK4_ERR_SUCCESS (0)` in case of no error
`SDK4_ERR_ERROR (-1001)` in case of unspecified error
one of the other error codes defined in `SDK4_ERROR`

Example

```
CTRL_HANDLE hCtrl;
...
SDK4_ENUM_SWITCH uSubtract;
SDK4GetImageSubtraction(hCtrl, &uSubtract);
```

See also

`SDK4DevGetControl`
`CTRL_HANDLE`
`SDK4_ENUM_SWITCH`
`SDK4StoreImage`
`SDK4SetImageSubtraction`

2.4.14.10 SDK4SetRecursiveFilter

This function sets the factor of the recursive filter to one of the values specified in SDK4_ENUM_RECURSIVEFILTER.

```
SDK4_Error SDK4SetRecursiveFilter(CTRL_HANDLE hCtrl,
          SDK4_ENUM_RECURSIVEFILTER uFactor)
```

Parameters

`hCtrl(CTRL_HANDLE )`

The control handle of the device

`uFactor (SDK4_ENUM_RECURSIVEFILTER)`

the recursive filter factor

Return value

Error code:

`SDK4_ERR_SUCCESS (0)` in case of no error
`SDK4_ERR_ERROR (-1001)` in case of unspecified error
one of the other error codes defined in `SDK4_ERROR`

Remarks

The recursive filter accumulates up to 16 images to reduce noise in the image. Setting the recursive filter to SDK4_ENUM_RECURSIVEFILTER_FACTOR1 turns the feature off.

Example

```
CTRL_HANDLE hCtrl;
...
// accumulate 8 image:
SDK4SetRecursiveFilter (hCtrl, SDK4_ENUM_RECURSIVEFILTER_FACTOR8);
```

See also

SDK4DevGetControl
CTRL_HANDLE
SDK4_ENUM_RECURSIVEFILTER
SDK4GetRecursiveFilter

2.4.14.11 SDK4GetRecursiveFilter

This function gets the current factor of the recursive filter.

```
SDK4_Error SDK4GetRecursiveFilter(CTRL_HANDLE hCtrl,
           SDK4_ENUM_RECURSIVEFILTER* puFactor)
```

Parameters

`hCtrl(CTRL_HANDLE )`

The control handle of the device

`puFactor (SDK4_ENUM_RECURSIVEFILTER*)`

the pointer to the recursive filter factor

Return value

Error code:

`SDK4_ERR_SUCCESS (0)` in case of no error
`SDK4_ERR_ERROR (-1001)` in case of unspecified error
one of the other error codes defined in `SDK4_ERROR`

Example

```
CTRL_HANDLE hCtrl;
...
SDK4_ENUM_RECURSIVEFILTER uFactor;
SDK4GetRecursiveFilter (hCtrl, &uFactor);
```

See also

SDK4DevGetControl
CTRL_HANDLE
SDK4_ENUM_RECURSIVEFILTER
SDK4SetRecursiveFilter

2.4.15 Look Up Table Functions

2.4.15.1 SDK4SetLUTMode

This function sets the mode of the LUT mode function to one of the values specified in SDK4_ENUM_LUTMODE.

```
SDK4_Error SDK4SetLUTMode(CTRL_HANDLE hCtrl,SDK4_ENUM_LUTMODE uMode)
```

Parameters

`hCtrl(CTRL_HANDLE )`

The control handle of the device

`uMode (uint32_t)`

the LUT mode

Return value

Error code:

`SDK4_ERR_SUCCESS (0)` in case of no error

`SDK4_ERR_ERROR (-1001)` in case of unspecified error

one of the other error codes defined in `SDK4_ERROR`

Remarks

The camera provides a Look-Up-Table to transform the image data. This can either be a linear transformation (SDK4_ENUM_LUTMODE_LINEAR), or the Look-Up-Table can be loaded with non-linear trasformation table. This table can either be a Gamma function (SDK4_ENUM_LUTMODE_GAMMA) or a User defined function (SDK4_ENUM_ LUTMODE_USERTABLE), which can be loaded into the camera using the Update File Messages. The index of the table, which is to be used, is specified by SDK4SetLUTInterpolatorIndex

Example

```
CTRL_HANDLE hCtrl;
...
// switch LUT mode off (linear transformation):
SDK4SetLUTMode (hCtrl, SDK4_ENUM_LUTMODE_LINEAR);
```

See also

`SDK4DevGetControl`
`CTRL_HANDLE`
`SDK4_ENUM_LUTMODE`
`SDK4SetLUTInterpolatorIndex`
`SDK4GetLUTMode`

2.4.15.2 SDK4GetLUTMode

This function gets the current value of the LUT mode function.

```
SDK4_Error SDK4GetLUTMode(CTRL_HANDLE hCtrl,SDK4_ENUM_LUTMODE*
puMode)
```

Parameters

`hCtrl(CTRL_HANDLE )`

The control handle of the device

`puMode (uint32_t*)`

the pointer to the LUT mode

Return value

Error code:

`SDK4_ERR_SUCCESS (0)` in case of no error
`SDK4_ERR_ERROR (-1001)` in case of unspecified error
one of the other error codes defined in `SDK4_ERROR`

Example

```
CTRL_HANDLE hCtrl;
...
SDK4_ENUM_LUTMODE uMode;
SDK4GetLUTMode(hCtrl, &uMode);
```

See also

`SDK4DevGetControl`
`CTRL_HANDLE`
`SDK4_ENUM_LUTMODE`
`SDK4SetLUTInterpolatorIndex`
`SDK4SetLUTMode`

2.4.15.3 SDK4SetLUTInterpolatorIndex

This function sets the index of the Look-Up Table Interpolator function.

```
SDK4_Error SDK4SetLUTInterpolatorIndex(CTRL_HANDLE hCtrl,int32_t
iIndex)
```

Parameters

`hCtrl(CTRL_HANDLE )`
The control handle of the device
`iIndex (int32_t)`
the selected LUT interpolator index

Return value

Error code:

`SDK4_ERR_SUCCESS (0)` in case of no error
`SDK4_ERR_ERROR (-1001)` in case of unspecified error
one of the other error codes defined in `SDK4_ERROR`

Remarks

In combination with the LUTMode Messages this function selects a LUT interpolator index. The Look-Up Table, which is used, is selected by the function SDK4SetLUTMode.
If SDK4_ENUM_LUTMODE_GAMMA is selected, the LUT interpolator index can interpolate between a Gamma 0.45 function (-8) and a Gamma 2.2 function (+12). If SDK4_ENUM_ LUTMODE_USERTABLE is selected, the LUT interpolator index can interpolate between a user defined transformation function and a linear transformation (-8 to 0). 0 is a linear transformation.

Example

```
CTRL_HANDLE hCtrl;
...
// set gamma index to 0,45:
SDK4SetLUTMode(hCtrl, SDK4_ENUM_LUTMODE_GAMMA);
SDK4SetLUTInterpolatorIndex (hCtrl, -8);
```

See also

```
SDK4DevGetControl
CTRL_HANDLE
SDK4GetLUTInterpolatorIndex
SDK4SetLUTMode
```

2.4.15.4 SDK4GetLUTInterpolatorIndex

This function gets the current value of the index of the Look-Up Table interpolator function.

```
SDK4_Error SDK4GetLUTInterpolatorIndex(CTRL_HANDLE hCtrl, int32_t*
          piIndex)
```

Parameters

`hCtrl(CTRL_HANDLE )`

The control handle of the device

`piIndex (int32_t*)`

the pointer to the LUT interpolator index

Return value

Error code:

`SDK4_ERR_SUCCESS (0)` in case of no error

`SDK4_ERR_ERROR (-1001)` in case of unspecified error

one of the other error codes defined in `SDK4_ERROR`

Remarks

In combination with the LUT Mode Messages the LUT interpolator index can interpolate between a Gamma 0.45 function (-8) and a Gamma 2.2 function (+12) (SDK4_ENUM_LUTMODE_GAMMA) or between a user defined transformation function and a linear transformation (-8 to 0) (SDK4_ENUM_LUTMODE_USERTABLE). 0 is a linear transformation.

Example

```
CTRL_HANDLE hCtrl;
...
int32_t iIndex;
SDK4GetLUTInterpolatorIndex (hCtrl, &iIndex);
```

See also

```
SDK4DevGetControl
CTRL_HANDLE
SDK4SetLUTInterpolatorIndex
SDK4GetLUTMode
```

2.4.15.5 SDK4SetNegativeImage

This function can invert the camera image.

```
SDK4_Error SDK4SetNegativeImage(CTRL_HANDLE hCtrl, SDK4_ENUM_SWITCH
           uNegativeImage)
```

Parameters

`hCtrl(CTRL_HANDLE )`

The control handle of the device

`uNegativeImage (uint32_t)`

the inversion value

Return value

Error code:

`SDK4_ERR_SUCCESS (0)` in case of no error
`SDK4_ERR_ERROR (-1001)` in case of unspecified error
one of the other error codes defined in `SDK4_ERROR`

Remarks

The inverting is done to the raw image data of the CCD sensor.

Example

```
CTRL_HANDLE hCtrl;
...
// invert the image:
SDK4SetNegativeImage (hCtrl, SDK4_ENUM_SWITCH_ON);
```

See also

`SDK4DevGetControl`
`CTRL_HANDLE`
`SDK4_ENUM_SWITCH`
`SDK4GetNegativeImage`

2.4.15.6 SDK4GetNegativeImage

This function gets the current value of the image inversion.

```
SDK4_Error SDK4GetNegativeImage(CTRL_HANDLE hCtrl, SDK4_ENUM_SWITCH*
           puNegativeImage)
```

Parameters

`hCtrl(CTRL_HANDLE )`

The control handle of the device

`puNegativeImage (uint32_t)`

the pointer to the inversion value

Return value

Error code:

`SDK4_ERR_SUCCESS (0)` in case of no error
`SDK4_ERR_ERROR (-1001)` in case of unspecified error
one of the other error codes defined in `SDK4_ERROR`

Remarks

The inverting is done to the raw image data of the CCD sensor.

Example

```
CTRL_HANDLE hCtrl;
...
SDK4_ENUM_SWITCH uNegative;
SDK4GetNegativeImage (hCtrl, &uNegative);
```

See also

```
SDK4DevGetControl
CTRL_HANDLE
SDK4_ENUM_SWITCH
SDK4SetNegativeImage
```

2.4.16 Color Functions

2.4.16.1 SDK4SetColorBalance

This function sets the Color Balance of a Color Camera.

```
SDK4_Error SDK4SetColorBalance(CTRL_HANDLE hCtrl,SDK4_KRGB kRgbGain)
```

Parameters

`hCtrl(CTRL_HANDLE )`

The control handle of the device

`kRgbGain (SDK4_KRGB)`

the color balance

Return value

Error code:

`SDK4_ERR_SUCCESS (0)` in case of no error

`SDK4_ERR_ERROR (-1001)` in case of unspecified error

one of the other error codes defined in `SDK4_ERROR`

Remarks

This function is only implemented in color cameras.

It is enabled only when AutoWhiteset is switched off.

Example

```
CTRL_HANDLE hCtrl;
...
// set color balance to neutral values:
SDK4_KRGB kRgbGain;
kRgbGain.red = 127;
kRgbGain.green = 127;
kRgbGain.blue = 127;
SDK4SetColorBalance (hCtrl, kRgbGain);
```

See also

`SDK4DevGetControl`
`CTRL_HANDLE`
`SDK4_KRGB`
`SDK4SetAutoWhiteset`
`SDK4GetColorBalance`

2.4.16.2 SDK4GetColorBalance

This function gets the current Color Balance of a Color Camera.

```
SDK4_Error SDK4GetColorBalance(CTRL_HANDLE hCtrl, SDK4_KRGB*
          pkRgbGain)
```

Parameters

`hCtrl(CTRL_HANDLE )`
    The control handle of the device
`pkRgbGain (SDK4_KRGB*)`
    the pointer to the color balance

Return value

Error code:
    `SDK4_ERR_SUCCESS (0)` in case of no error
    `SDK4_ERR_ERROR (-1001)` in case of unspecified error
    one of the other error codes defined in `SDK4_ERROR`

Remarks

This function is only implemented in color cameras.

Example

```
CTRL_HANDLE hCtrl;
...
// set color balance to neutral values:
SDK4_KRGB kRgbGain;
SDK4GetColorBalance (hCtrl, &kRgbGain);
```

See also

`SDK4DevGetControl`
`CTRL_HANDLE`
`SDK4_KRGB`
`SDK4SetColorBalance`

2.4.16.3 SDK4SetLightSource

This function sets the Color Temperature of a Color Camera to a value specified in SDK4_ENUM_LIGHTSOURCE, in order to optimize the color processor for different light sources.

```
SDK4_Error SDK4SetLightSource(CTRL_HANDLE hCtrl,
           SDK4_ENUM_LIGHTSOURCE uSource)
```

Parameters

`hCtrl(CTRL_HANDLE )`

The control handle of the device

`uSource (SDK4_ENUM_LIGHTSOURCE)`

the selected light source

Return value

Error code:

`SDK4_ERR_SUCCESS (0)` in case of no error

`SDK4_ERR_ERROR (-1001)` in case of unspecified error

one of the other error codes defined in `SDK4_ERROR`

Remarks

This function is only implemented in color cameras.

Example

```
CTRL_HANDLE hCtrl;
// adapt the color temperature to day light:
SDK4SetLightSource (hCtrl, SDK4_ENUM_LIGHTSOURCE_DAYLIGHT);
```

See also

`SDK4DevGetControl`

`CTRL_HANDLE`

`SDK4_ENUM_LIGHTSOURCE`

`SDK4GetLightSource`

2.4.16.4 SDK4GetLightSource

This function gets the current Lightsource value of the Camera.

```
SDK4_Error SDK4GetLightSource(CTRL_HANDLE hCtrl,
           SDK4_ENUM_LIGHTSOURCE* puSource)
```

Parameters

`hCtrl(CTRL_HANDLE )`

The control handle of the device

`puSource (SDK4_ENUM_LIGHTSOURCE)`

the pointer to the light source

Return value

Error code:

`SDK4_ERR_SUCCESS (0)` in case of no error

`SDK4_ERR_ERROR (-1001)` in case of unspecified error

one of the other error codes defined in `SDK4_ERROR`

Remarks

This function is only implemented in color cameras.

Example

```
CTRL_HANDLE hCtrl;
...
SDK4_ENUM_LIGHTSOURCE uLightSource;
SDK4GetLightSource (hCtrl, &uLightSource);
```

See also

`SDK4DevGetControl`
`CTRL_HANDLE`
`SDK4_ENUM_LIGHTSOURCE`
`SDK4SetLightSource`

2.4.16.5 SDK4SetColorSaturation

This function sets the Color Saturation of a Color Camera.

```
SDK4_Error SDK4SetColorSaturation(CTRL_HANDLE hCtrl, uint32_t
          uSaturation)
```

Parameters

`hCtrl(CTRL_HANDLE )`
    The control handle of the device
`uSaturation (uint32_t)`
    the saturation

Return value

Error code:

`SDK4_ERR_SUCCESS (0)` in case of no error
`SDK4_ERR_ERROR (-1001)` in case of unspecified error
one of the other error codes defined in `SDK4_ERROR`

Remarks

This function is only implemented in color cameras.
The values range from 0 (0% = black and white) to 255 (200%).

Example

```
CTRL_HANDLE hCtrl;
...
// set color saturation to the default value:
SDK4SetColorSaturation (hCtrl, 127);
```

See also

`SDK4DevGetControl`
`CTRL_HANDLE`
`SDK4GetColorSaturation`

2.4.16.6 SDK4GetColorSaturation

This function gets the current Color Saturation of a Color Camera.

```
SDK4_Error SDK4GetColorSaturation(CTRL_HANDLE hCtrl, uint32_t*
          puSaturation)
```

Parameters

`hCtrl(CTRL_HANDLE )`
	The control handle of the device

`puSaturation (uint32_t*)`
	the pointer to the saturation

Return value

Error code:

`SDK4_ERR_SUCCESS (0)` in case of no error
`SDK4_ERR_ERROR (-1001)` in case of unspecified error
one of the other error codes defined in `SDK4_ERROR`

Remarks

This function is only implemented in color cameras.

Example

```
CTRL_HANDLE hCtrl;
...
uint32_t uSaturation
SDK4GetColorSaturation (hCtrl, &uSaturation);
```

See also

`SDK4DevGetControl`
`CTRL_HANDLE`
`SDK4SetColorSaturation`

2.4.16.7 SDK4SetAutoWhiteset

This function sets the Automatic Whiteset Control of a Color Camera to one of the values specified in SDK4_ENUM_WHITESET.

```
SDK4_Error SDK4SetAutoWhiteset(CTRL_HANDLE hCtrl, SDK4_ENUM_WHITESET
          uWhiteset)
```

Parameters

`hCtrl(CTRL_HANDLE )`
	The control handle of the device

`uWhiteset (SDK4_ENUM_WHITESET)`
	the whiteset value

Return value

Error code:

`SDK4_ERR_SUCCESS (0)` in case of no error
`SDK4_ERR_ERROR (-1001)` in case of unspecified error
one of the other error codes defined in `SDK4_ERROR`

Remarks

This function is only implemented in color cameras.
SDK4_ENUM_WHITESET_ON performs a permanent whiteset, SDK4_ENUM_WHITESET_PUSH once performs a whiteset and then stops.

Example

```
CTRL_HANDLE hCtrl;
...
// perform a whiteset once and then wait, until the whiteset has
// completed:
SDK4SetAutoWhiteset (hCtrl, SDK4_ENUM_WHITESET_PUSH);
while (true)
{
     SDK4_ENUM_WHITESET ws;
     SDK4GetAutoWhiteset (hCtrl, &ws);
     if (ws == SDK4_ENUM_WHITESET_OFF)
       break;
}
```

See also

SDK4DevGetControl
CTRL_HANDLE
SDK4_ENUM_WHITESET
SDK4GetAutoWhiteset

2.4.16.8 SDK4GetAutoWhiteset

This function gets the current value of the Automatic Whiteset Control of a Color Camera.

```
SDK4_Error SDK4GetAutoWhiteset(CTRL_HANDLE hCtrl,
           SDK4_ENUM_WHITESET* puWhiteset)
```

Parameters

`hCtrl(CTRL_HANDLE )`

The control handle of the device

`puWhiteset (SDK4_ENUM_WHITESET*)`

the pointer to the whiteset value

Return value

Error code:

`SDK4_ERR_SUCCESS (0)` in case of no error
`SDK4_ERR_ERROR (-1001)` in case of unspecified error
one of the other error codes defined in `SDK4_ERROR`

Remarks

This function is only implemented in color cameras.
After SDK4_ENUM_WHITESET.PUSH has been set, this function returns a whiteset value > 0, as long as the whiteset has not been completed.

Example

```
CTRL_HANDLE hCtrl;
...
// perform a whiteset once and then wait, until the whiteset has
// completed:
SDK4SetAutoWhiteset (hCtrl, SDK4_ENUM_WHITESET_PUSH);
while (true)
{
      SDK4_ENUM_WHITESET ws;
      SDK4GetAutoWhiteset (hCtrl, &ws);
      if (ws == SDK4_ENUM_WHITESET_OFF)
        break;
}
```

See also

```
SDK4DevGetControl
CTRL_HANDLE
SDK4_ENUM_WHITESET
SDK4SetAutoWhiteset
```

2.4.17 Internal Test Functions

2.4.17.1 SDK4SetPatternGenerator

This function sets one of Test patterns which are specified in SDK4_ENUM_PATTERN.

```
SDK4_Error SDK4SetPatternGenerator(CTRL_HANDLE hCtrl,
SDK4_ENUM_PATTERN uSelect)
```

Parameters

`hCtrl(CTRL_HANDLE )`

The control handle of the device

`uSelect (SDK4_ENUM_PATTERN)`

the selected test pattern

Return value

Error code:

`SDK4_ERR_SUCCESS (0)` in case of no error
`SDK4_ERR_ERROR (-1001)` in case of unspecified error
one of the other error codes defined in `SDK4_ERROR`

Remarks

The pattern SDK4_ENUM_PATTERN_BW_NORMAL resp. SDK4_ENUM_PATTERN_C_NORMAL switches to the live image of the CCD, all other options show test patterns.
Note:
The test patterns of monochrome and color cameras are different (see SDK4_ENUM_PATTERN_C and SDK4_ENUM_PATTERN_BW).

Example

```
CTRL_HANDLE hCtrl;
...
// switch from live image to a test pattern:
SDK4SetPatternGenerator (hCtrl, SDK4_ENUM_PATTERN_C_NORMAL);
...
SDK4SetPatternGenerator (hCtrl, SDK4_ENUM_PATTERN_C_COLORBARS);
```

See also

SDK4DevGetControl
CTRL_HANDLE
SDK4_ENUM_PATTERN
SDK4_ENUM_PATTERN_C
SDK4_ENUM_PATTERN_BW
SDK4GetPatternGenerator

2.4.17.2 SDK4GetPatternGenerator

This function gets the current Test pattern of the camera.

```
SDK4_Error SDK4GetPatternGenerator(CTRL_HANDLE hCtrl,SDK4_ENUM_PATTERN* puSelect)
```

Parameters

hCtrl(CTRL_HANDLE)

The control handle of the device

puSelect (SDK4_ENUM_PATTERN*)

the pointer to the test pattern

Return value

Error code:

SDK4_ERR_SUCCESS (0) in case of no error
SDK4_ERR_ERROR (-1001) in case of unspecified error
one of the other error codes defined in SDK4_ERROR

Remarks

Note:

The test patterns of monochrome and color cameras are different (see SDK4_ENUM_PATTERN_C and SDK4_ENUM_PATTERN_BW).

Example

```
CTRL_HANDLE hCtrl;
...
SDK4_ENUM_PATTERN uPattern:
SDK4GetPatternGenerator (hCtrl, &uPattern);
```

See also

SDK4DevGetControl
CTRL_HANDLE
SDK4_ENUM_PATTERN
SDK4_ENUM_PATTERN_C
SDK4_ENUM_PATTERN_BW
SDK4SetPatternGenerator

2.4.18 Basic Settings Functions

2.4.18.1 SDK4SaveSettings

This function stores the current settings to one of the user defined settings in the camera EEPROM.

```
SDK4_Error SDK4SaveSettings(CTRL_HANDLE hCtrl, SDK4_ENUM_SETTINGS
          uSetting)
```

Parameters

`hCtrl(CTRL_HANDLE )`

The control handle of the device

`uSetting (SDK4_ENUM_SETTINGS)`

the selected setting

Return value

Error code:

`SDK4_ERR_SUCCESS (0)` in case of no error
`SDK4_ERR_ERROR (-1001)` in case of unspecified error
one of the other error codes defined in `SDK4_ERROR`

Remarks

The default setting cannot be changed.

Example

```
CTRL_HANDLE hCtrl;
...
// save current settings to user settings 1 and then declare them as
// startup settings:
SDK4SaveSettings (hCtrl, SDK4_ENUM_SETTINGS_USERSETTINGS1);
SDK4SetStartupSettings (hCtrl, SDK4_ENUM_SETTINGS_USERSETTINGS1);
```

See also

`SDK4DevGetControl`
`CTRL_HANDLE`
`SDK4_ENUM_SETTINGS`
`SDK4LoadSettings`
`SDK4SetStartupSettings`

2.4.18.2 SDK4LoadSettings

This function loads either the default settings or one of the previously stored user defined settings from the camera EEPROM.

```
SDK4_Error SDK4LoadSettings(CTRL_HANDLE hCtrl, SDK4_ENUM_SETTINGS
          uSetting)
```

Parameters

`hCtrl(CTRL_HANDLE )`

The control handle of the device

`uSetting (SDK4_ENUM_ SETTINGS)`

the selected setting

Return value

Error code:

`SDK4_ERR_SUCCESS (0)` in case of no error

`SDK4_ERR_ERROR (-1001)` in case of unspecified error

one of the other error codes defined in `SDK4_ERROR`

Example

```
CTRL_HANDLE hCtrl;
...
// load the factory settings:
SDK4LoadSettings (hCtrl, SDK4_ENUM_SETTINGS_FACTORYSETTINGS);
```

See also

`SDK4DevGetControl`
`CTRL_HANDLE`
`SDK4_ENUM_SETTINGS`
`SDK4SaveSettings`
`SDK4SetStartupSettings`

2.4.18.3 SDK4SetStartupSettings

This function selects either the default settings or one of the previously stored user defined settings, for the camera to use at startup.

```
SDK4_Error SDK4SetStartupSettings(CTRL_HANDLE hCtrl,
          SDK4_ENUM_SETTINGS uSetting)
```

Parameters

`hCtrl(CTRL_HANDLE )`

The control handle of the device

`uSetting (SDK4_ENUM_ SETTINGS)`

the selected setting

Return value

Error code:

`SDK4_ERR_SUCCESS (0)` in case of no error

`SDK4_ERR_ERROR (-1001)` in case of unspecified error

one of the other error codes defined in `SDK4_ERROR`

Example

```
CTRL_HANDLE hCtrl;
...
// save current settings to user settings 1 and then declare them as
// startup settings:
SDK4SaveSettings (hCtrl, SDK4_ENUM_SETTINGS_USERSETTINGS1);
SDK4SetStartupSettings (hCtrl, SDK4_ENUM_SETTINGS_USERSETTINGS1);
```

See also

`SDK4DevGetControl`
`CTRL_HANDLE`
`SDK4_ENUM_SETTINGS`
`SDK4SaveSettings`
`SDK4GetStartupSettings`

2.4.18.4 SDK4GetStartupSettings

This function gets the currently selected startup setting.

```
SDK4_Error SDK4GetStartupSettings(CTRL_HANDLE hCtrl,
           SDK4_ENUM_SETTINGS* puSetting)
```

Parameters

`hCtrl(CTRL_HANDLE )`

    The control handle of the device

`puSetting (SDK4_ENUM_ SETTINGS*)`

    the pointer to the setting

Return value

Error code:

`SDK4_ERR_SUCCESS (0)` in case of no error
`SDK4_ERR_ERROR (-1001)` in case of unspecified error
one of the other error codes defined in `SDK4_ERROR`

Example

```
CTRL_HANDLE hCtrl;
...
SDK4_ENUM_SETTINGS uSettings;
SDK4GetStartupSettings (hCtrl, &uSettings);
```

See also

`SDK4DevGetControl`
`CTRL_HANDLE`
`SDK4_ENUM_SETTINGS`
`SDK4SaveSettings`
`SDK4SetStartupSettings`

2.5 Appendix

2.5.1 Enumerations SDK4Acquire

2.5.1.1 ENUM_DEVICE_ACCESS

This enumeration defines SDK4 types of device access.

```
enum ENUM_DEVICE_ACCESS
{
     Open the device read only. All Port functions can only
     read from the device:
     DEVICEACCESS_READONLY  =2,
     Open the device in a way that other hosts/processes can
     have read only access to the device. Device access level
     is read/write for this process:
     DEVICEACCESS_CONTROL   =3,
     Open the device in a way that only this host/process can
     have access to the device. Device access level is
     read/write for this process:
     DEVICEACCESS_EXCLUSIVE =4
};
```

2.5.1.2 ENUM_MEDIATYPE

This enumeration defines SDK4 types of device access.

```
enum ENUM_MEDIATYPE
{
     MEDIATYPE_VIDEOFRAME  =0,
     MEDIATYPE_VIDEOSTREAM =1,
     MEDIATYPE_AUDIOSTREAM =2
};
```

2.5.1.3 ENUM_PIXELFORMAT

This enumeration defines SDK4 pixel formats.

```
enum ENUM_PIXELFORMAT
{
        PIXELFORMAT_RGB24     = 0x01180001,
        PIXELFORMAT_RGB32     = 0x01200002,
        PIXELFORMAT_YUV       = 0x01100003,
        PIXELFORMAT_UYVY      = 0x0110000f,
        PIXELFORMAT_Y8        = 0x00080004,
        PIXELFORMAT_Y10       = 0x00100005,
        PIXELFORMAT_Y12       = 0x00100006,
        PIXELFORMAT_Y12P      = 0x00100010,
        PIXELFORMAT_Y12_MSB   = 0x00100012,
        PIXELFORMAT_Y14       = 0x00100007,
        PIXELFORMAT_Y14_MSB   = 0x00100013,
        PIXELFORMAT_Y16       = 0x00100008,
        PIXELFORMAT_Y12S      = 0x02100009,
        PIXELFORMAT_Y14S      = 0x0210000a,
        PIXELFORMAT_YUVS      = 0x0310000c,
        PIXELFORMAT_D8        = 0x00080014,
        PIXELFORMAT_D12       = 0x0010000d,
        PIXELFORMAT_D14       = 0x0010000e
};
```

2.5.2 Enumerations SDK4ZelosControl

2.5.2.1 SDK4_ENUM_SWITCH

This enumeration defines SDK4 switch for "on" and "off" state.

```
enum SDK4_ENUM_SWITCH
{
        SDK4_ENUM_SWITCH_OFF  = 0,
        SDK4_ENUM_SWITCH_ON   = 1
};
```

2.5.2.2 SDK4_ENUM_EXPOSUREBASE

This enumeration defines SDK4 types of exposure base.

```
enum SDK4_ENUM_EXPOSUREBASE
{
        SDK4_ENUM_EXPOSUREBASE_PIXELCLOCK     = 0,
        SDK4_ENUM_EXPOSUREBASE_1us            = 1,
        SDK4_ENUM_EXPOSUREBASE_10us           = 2,
        SDK4_ENUM_EXPOSUREBASE_100us          = 3,
        SDK4_ENUM_EXPOSUREBASE_1ms            = 4,
        SDK4_ENUM_EXPOSUREBASE_10ms           = 5,
        SDK4_ENUM_EXPOSUREBASE_100ms          = 6,
        SDK4_ENUM_EXPOSUREBASE_1s             = 7
};
```

2.5.2.3 SDK4_ENUM_EXPOSUREMODE

This enumeration defines SDK4 modes of exposure..

```
enum SDK4_ENUM_EXPOSUREMODE
{
     ZELOS_ENUM_EXPOSUREMODE_FREERUNNINGPARALLEL     = 1,
     ZELOS_ENUM_EXPOSUREMODE_FREERUNNINGSEQUENTIAL   = 2,
     ZELOS_ENUM_EXPOSUREMODE_RESETRESTART            = 3,
     ZELOS_ENUM_EXPOSUREMODE_FRAMEONDEMAND           = 4,
     ZELOS_ENUM_EXPOSUREMODE_EXTERNALSYNC            = 5
};
```

2.5.2.4 SDK4_ENUM_AET

This enumeration defines SDK4 modes of AET.

```
enum SDK4_ENUM_AET
{
     SDK4_ENUM_AET_OFF    = 0,
     SDK4_ENUM_AET_ON     = 1,
     SDK4_ENUM_AET_PUSH   = 2,
     SDK4_ENUM_AET_ON_EXT = 3
};
```

2.5.2.5 SDK4_ENUM_AGC

This enumeration defines SDK4 modes of AGC.

```
enum SDK4_ENUM_AGC
{
     ZELOS_ENUM_AGC_OFF   = 0,
     ZELOS_ENUM_AGC_ON    = 1,
     ZELOS_ENUM_AGC_PUSH  = 2
};
```

2.5.2.6 SDK4_ENUM_ROADOMAIN

This enumeration defines SDK4 types of readout area domains.

```
enum SDK4_ENUM_ROADOMAIN
{
     SDK4_ENUM_ROADOMAIN_MIN    = 0,
     SDK4_ENUM_ROADOMAIN_MAX    = 1,
     SDK4_ENUM_ROADOMAIN_ACTIVE = 2
};
```

2.5.2.7 SDK4_ENUM_BINNING

This enumeration defines SDK4 modes of binning.

```
enum SDK4_ENUM_BINNING
{
        SDK4_ENUM_BINNING_1x  = 1,
        SDK4_ENUM_BINNING_2x  = 2,
        SDK4_ENUM_BINNING_3x  = 3,
        SDK4_ENUM_BINNING_4x  = 4,
        SDK4_ENUM_BINNING_5x  = 5,
        SDK4_ENUM_BINNING_6x  = 6,
        SDK4_ENUM_BINNING_7x  = 7,
        SDK4_ENUM_BINNING_8x  = 8
};
```

2.5.2.8 SDK4_ENUM_ACC

This enumeration defines SDK4 modes of ACC.

```
enum SDK4_ENUM_ACC
{
        SDK4_ENUM_ACC_OFF                    = 0,
        SDK4_ENUM_ACC_PUSHDRE                = 1,
        SDK4_ENUM_ACC_AUTODRE                = 2,
        SDK4_ENUM_ACC_HISTOGRAM              = 3,
        SDK4_ENUM_ACC_AUTOHISTOGRAM          = 4
};
```

2.5.2.9 SDK4_ENUM_GAMMA

This enumeration defines SDK4 modes of gamma correction.

```
enum SDK4_ENUM_GAMMA
{
        SDK4_ENUM_GAMMA_OFF          = 0,
        SDK4_ENUM_GAMMA_045          = 1,
        SDK4_ENUM_GAMMA_USERTABLE    = 2
};
```

2.5.2.10 SDK4_ENUM_MEASUREVALUE

This enumeration defines SDK4 types of measure values.

```
enum SDK4_ENUM_MEASUREVALUE
{
        SDK4_ENUM_MEASUREVALUE_MIN           = 0,
        SDK4_ENUM_MEASUREVALUE_MEAN          = 1,
        SDK4_ENUM_MEASUREVALUE_MAX           = 2
};
```

2.5.2.11 SDK4_ENUM_COLORCODING

There are different enumerations of color coding types for monochrome and color cameras.
SDK4_ENUM_COLORCODING_BW defines SDK4 types of color coding in monochrome cameras.

```
enum SDK4_ENUM_COLORCODING_BW
{
     SDK4_ENUM_COLORCODING_Y8         = 0,
     SDK4_ENUM_COLORCODING_Y12        = 4,
     SDK4_ENUM_COLORCODING_Y14        = 6,
     SDK4_ENUM_COLORCODING_Y16        = 8,
     SDK4_ENUM_COLORCODING_Y12P       = 14,
     SDK4_ENUM_COLORCODING_Y12_MSB    = 24,
     SDK4_ENUM_COLORCODING_Y14_MSB    = 26,
};
```

SDK4_ENUM_COLORCODING_C defines SDK4 types of color coding in color cameras.

```
enum SDK4_ENUM_COLORCODING_C
{
     SDK4_ENUM_COLORCODING_YUV422     = 51,
     SDK4_ENUM_COLORCODING_UYV422     = 52,
     SDK4_ENUM_COLORCODING_RGB24      = 101,
     SDK4_ENUM_COLORCODING_RGB888     = 110,
     SDK4_ENUM_COLORCODING_BGR888     = 111,
     SDK4_ENUM_COLORCODING_D8         = 150,
     SDK4_ENUM_COLORCODING_D12        = 151,
     SDK4_ENUM_COLORCODING_D14        = 153,
     SDK4_ENUM_COLORCODING_D16        = 155,
};
```

2.5.2.12 SDK4_ENUM_IMAGEHOLD

This enumeration defines SDK4 modes of Image Hold.

```
enum SDK4_ENUM_IMAGEHOLD
{
     SDK4_ENUM_IMAGEHOLD_LIVE = 0,
     SDK4_ENUM_IMAGEHOLD_HOLD = 1
};
```

2.5.2.13 SDK4_ENUM_RECURSIVEFILTER

This enumeration defines SDK4 types of Recursive Filter Factors.

```
enum SDK4_ENUM_RECURSIVEFILTER
{
     SDK4_ENUM_RECURSIVEFILTER_FACTOR1     = 0,
     SDK4_ENUM_RECURSIVEFILTER_FACTOR2     = 1,
     SDK4_ENUM_RECURSIVEFILTER_FACTOR4     = 2,
     SDK4_ENUM_RECURSIVEFILTER_FACTOR8     = 3,
     SDK4_ENUM_RECURSIVEFILTER_FACTOR16    = 4
};
```

2.5.2.14 SDK4_ENUM_LIGHTSOURCE

This enumeration defines SDK4 types of light sources.

```
enum SDK4_ENUM_LIGHTSOURCE
{
        SDK4_ENUM_LIGHTSOURCE_DAYLIGHT  = 0,
        SDK4_ENUM_LIGHTSOURCE_HALOGEN   = 1,
        SDK4_ENUM_LIGHTSOURCE_WHITELED  = 2,
        SDK4_ENUM_LIGHTSOURCE_LINEAR    = 3
};
```

2.5.2.15 SDK4_ENUM_WHITESET

This enumeration defines SDK4 modes of whiteset.

```
enum SDK4_ENUM_WHITESET
{
        SDK4_ENUM_WHITESET_OFF     = 0,
        SDK4_ENUM_WHITESET_ON      = 1,
        SDK4_ENUM_WHITESET_PUSH    = 2
};
```

2.5.2.16 SDK4_ENUM_OVERLAYSELECT

This enumeration defines SDK4 types of overlays.

```
enum SDK4_ENUM_OVERLAYSELECT
{
        SDK4_ENUM_OVERLAYSELECT_LINEGENERATOR1_ALLLINES = 0,
        SDK4_ENUM_OVERLAYSELECT_LINRGENERATOR2          = 1,
        SDK4_ENUM_OVERLAYSELECT_CIRCLEGENERATOR1        = 2,
        SDK4_ENUM_OVERLAYSELECT_CIRCLEGENERATOR2        = 3
};
```

2.5.2.17 SDK4_ENUM_LINESTYLE

This enumeration defines SDK4 types of overlay line styles.

```
enum SDK4_ENUM_LINESTYLE
{
        SDK4_ENUM_LINESTYLE_SOLID       = 0,
        SDK4_ENUM_LINESTYLE_DASHED      = 1
};
```

2.5.2.18 SDK4_ENUM_OVERLAYWIDTH

This enumeration defines SDK4 types of overlay line width.

```
enum SDK4_ENUM_OVERLAYWIDTH
{
        SDK4_ENUM_OVERLAYWIDTH_PIXEL1 = 0,
        SDK4_ENUM_OVERLAYWIDTH_PIXEL3 = 1,
        SDK4_ENUM_OVERLAYWIDTH_PIXEL5 = 2,
        SDK4_ENUM_OVERLAYWIDTH_PIXEL7 = 3
};
```

2.5.2.19 SDK4_ENUM_MEASUREWINDOWSELECT

This enumeration defines SDK4 selectable measure windows.

```
enum SDK4_ENUM_MEASUREWINDOWSELECT
{
        SDK4_ENUM_MEASUREWINDOW1    = 0,
        SDK4_ENUM_MEASUREWINDOW2    = 1
};
```

2.5.2.20 SDK4_ENUM_MEASUREFRAMECOLOR

There are different enumerations of measure frame colors for monochrome and color cameras. `ZELOS_ENUM_MFCOLOR_BW` defines SDK4 measure frame colors in monochrome cameras.

```
enum ZELOS_ENUM_MFCOLOR_BW
{
        ZELOS_ENUM_COLOR_GREY0      = 0,
        ZELOS_ENUM_COLOR_GREY15     = 1,
        ZELOS_ENUM_COLOR_GREY30     = 2,
        ZELOS_ENUM_COLOR_GREY45     = 3,
        ZELOS_ENUM_COLOR_GREY60     = 4,
        ZELOS_ENUM_COLOR_GREY75     = 5,
        ZELOS_ENUM_COLOR_GREY90     = 6,
        ZELOS_ENUM_COLOR_GREY100    = 7
};
```

`ZELOS_ENUM_MFCOLOR_C` defines SDK4 measure frame colors in color cameras.

```
enum ZELOS_ENUM_MFCOLOR_C
{
        ZELOS_ENUM_MFCOLOR_C_BLACK       = 0,
        ZELOS_ENUM_MFCOLOR_C_BLUE        = 1,
        ZELOS_ENUM_MFCOLOR_C_RED         = 2,
        ZELOS_ENUM_MFCOLOR_C_MAGENTA     = 3,
        ZELOS_ENUM_MFCOLOR_C_GREEN       = 4,
        ZELOS_ENUM_MFCOLOR_C_CYAN        = 5,
        ZELOS_ENUM_MFCOLOR_C_YELLOW      = 6,
        ZELOS_ENUM_MFCOLOR_C_WHITE       = 7
};
```

2.5.2.21 SDK4_ENUM_PATTERN

There are different enumerations of color coding types for monochrome and color cameras. SDK4_ENUM_PATTERN_BW defines SDK4 types of test patterns in monochrome cameras.

```
enum SDK4_ENUM_PATTERN_BW
{
        SDK4_ENUM_PATTERN_BW_NORMAL      = 0,
        SDK4_ENUM_PATTERN_BW_BARS        = 1,
        SDK4_ENUM_PATTERN_BW_BURST       = 2,
        SDK4_ENUM_PATTERN_BW_BARSBURST   = 3
};
```

SDK4_ENUM_PATTERN_C defines SDK4 types of test patterns in color cameras.

```
enum SDK4_ENUM_PATTERN_C
{
        SDK4_ENUM_PATTERN_C_NORMAL       = 0,
        SDK4_ENUM_PATTERN_C_COLORBARS    = 1,
};
```

2.5.2.22 SDK4_ENUM_SETTINGS

This enumeration defines SDK4 types of settings, which are stored in the EEPROM of the camera.

```
enum SDK4_ENUM_SETTINGS
{
        SDK4_ENUM_SETTINGS_FACTORYSETTINGS    = 0,
        SDK4_ENUM_SETTINGS_USERSETTINGS1      = 1,
        SDK4_ENUM_SETTINGS_USERSETTINGS2      = 2
};
```

2.5.2.23 SDK4_ENUM_ACCESSMODE

This enumeration defines SDK4 access modes.

```
enum SDK4_ENUM_ACCESSMODE
{
        SDK4_ENUM_ACCESS_USERMODE             = 0,
        SDK4_ENUM_ACCESS_MAINTANANCEMODE      = 1,
        SDK4_ENUM_ACCESS_ADJUSTMODE           = 2
};
```

2.5.2.24 SDK4_ENUM_OPERATIONMODE

This enumeration defines SDK4 operation modes.

```
enum SDK4_ENUM_OPERATIONMODE
{
        SDK4_ENUM_OPERATION_BOOT         = 0,
        SDK4_ENUM_OPERATION_OPERATION    = 1,
        SDK4_ENUM_OPERATION_FAILURE      = 2
};
```

2.5.3 Structures SDK4ZelosControl

2.5.3.1 SDK4_KBINNING

This structure defines SDK4 datatype of Binning.

```
struct SDK4_KBINNING
{
     uint32_t horizontal;
     uint32_t vertical;
};
```

2.5.3.2 SDK4_KOVERLAYSIZE

This structure defines SDK4 datatype of Overlay Size.

```
struct SDK4_KOVERLAYSIZE
{
   uint32_t width;
   uint32_t length;
};
```

2.5.3.3 SDK4_KSTYLE

This structure defines SDK4 datatype of Overlay Style.

```
struct SDK4_KSTYLE
{
     uint32_t color;
     uint32_t style;
};
```

2.5.3.4 SDK4_KSIZE

This structure defines SDK4 datatype of Image Size.

```
struct SDK4_KSIZE
{
     uint32_t width;
     uint32_t height;
};
```

2.5.3.5 SDK4_KPOSITION

This structure defines SDK4 datatype of Position.

```
struct SDK4_KPOSITION
{
     uint32_t xPos;
     uint32_t yPos;
};
```

2.5.3.6 SDK4_KAREA

This structure defines SDK4 datatype of Area.

```
     struct SDK4_KAREA
     {
          SDK4_KPOSITION start;
          SDK4_KPOSITION end;
     };
```

2.5.3.7 SDK4_KEXPOSURE

This structure defines SDK4 datatype of Exposure Time.

```
     struct SDK4_KEXPOSURE
     {
          uint32_t base;
          uint32_t counter;
     };
```

2.5.3.8 SDK4_KRGB

This structure defines SDK4 datatype of Color Balance.

```
     struct SDK4_KRGB
     {
          uint32_t red;
          uint32_t green;
          uint32_t blue;
     };
```

2.5.3.9 SDK4_KVERSION

This structure defines SDK4 datatype of Version.

```
     struct SDK4_KVERSION
     {
          uint32_t major;
          uint32_t minor;
     };
```

2.5.3.10 SDK4_KREVISION

This structure defines SDK4 datatype of Revision.

```
     struct SDK4_KREVISION
     {
          uint32_t hardware;
          uint32_t software;
     };
```

2.5.3.11 SDK4_KACCESSMODE

This structure defines SDK4 datatype of Access Mode.

```
struct SDK4_KACCESSMODE
{
      uint32_t mode;
      uint32_t unlockCode;
};
```

2.5.4 Error Codes

2.5.4.1 SDK4_ERROR

This enumeration defines SDK4 error types.

```
enum SDK4_ERROR
{
     SDK4_ERR_SUCCESS                = 0,
     SDK4_ERR_ERROR                  = -1001,
     SDK4_ERR_NOT_INITIALIZED        = -1002,
     SDK4_ERR_NOT_IMPLEMENTED        = -1003,
     SDK4_ERR_RESOURCE_IN_USE        = -1004,
     SDK4_ERR_ACCESS_DENIED          = -1005,
     SDK4_ERR_INVALID_HANDLE         = -1006,
     SDK4_ERR_INVALID_ID             = -1007,
     SDK4_ERR_NO_DATA                = -1008,
     SDK4_ERR_INVALID_PARAMETER      = -1009,
     SDK4_ERR_IO                     = -1010,
     SDK4_ERR_TIMEOUT                = -1011,
     SDK4_ERR_CANTSETFEATURE         = -1012,
     SDK4_ERR_CANTGETFEATURE         = -1013,
     SDK4_ERR_OUTOFRANGE             = -1014,
     SDK4_ERR_DISABLED               = -1015,
     SDK4_ERR_ACQUISITIONRUNNING     = -1016,
     SDK4_ERR_INVALID_BUFFER_SIZE    = -1201,
     SDK4_ERR_INVALID_ADDRESS        = -1202,
     SDK4_ERR_INVALID_INTERFACE      = -1203,
};
```