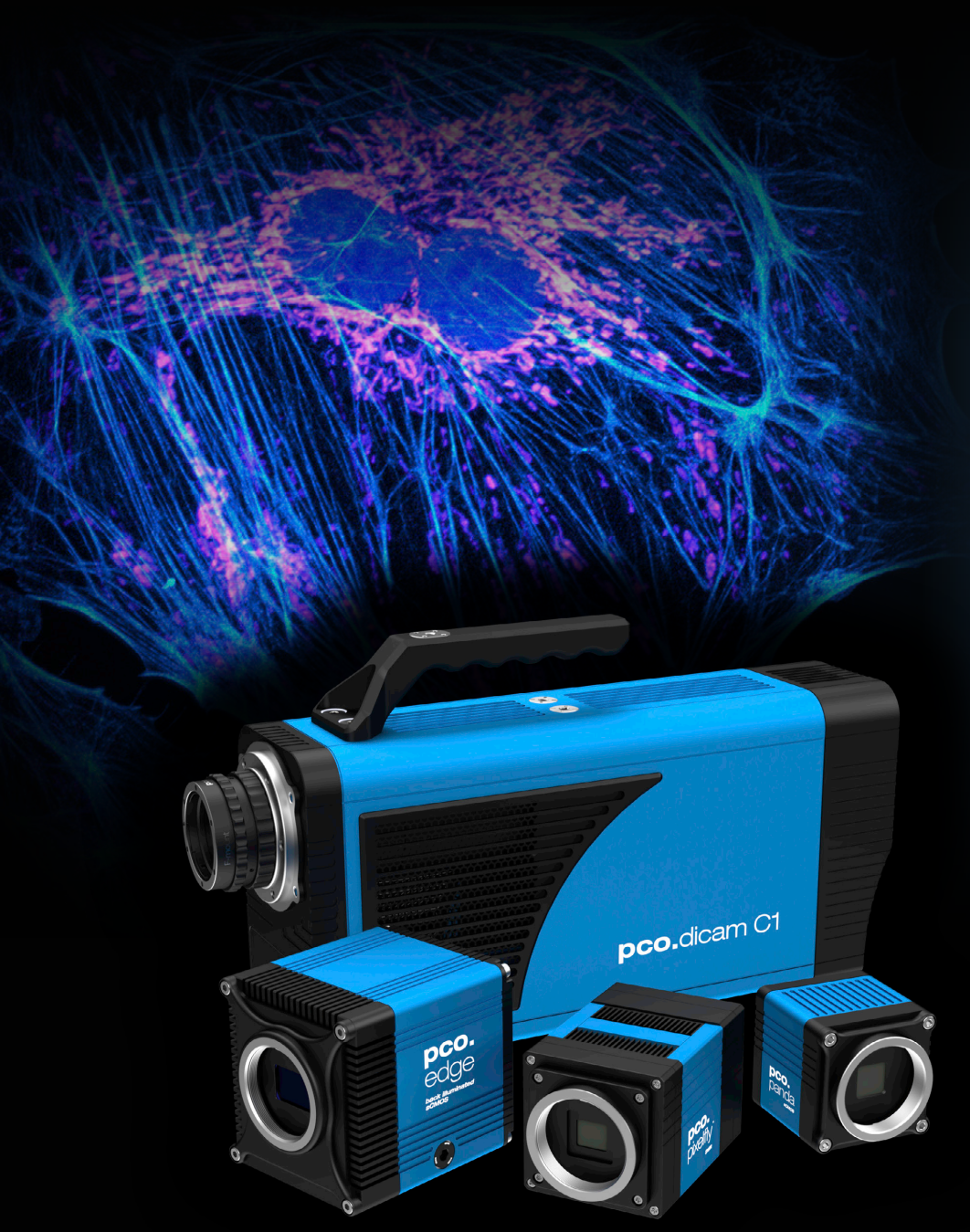
user manual

pco.convert



conv.

EXCELITAS TECHNOLOGIES®

pco.
An Excelitas Technologies Brand

Excelitas PCO GmbH asks you to carefully read and follow the instructions in this document.
For any questions or comments, please feel free to contact us at any time.

pco.
An Excelitas Technologies Brand

telephone:	+49 (0) 9441 2005 50
fax:	+49 (0) 9441 2005 20
postal address:	Excelitas PCO GmbH Donaupark 11 93309 Kelheim, Germany
email:	pco@excelitas.com
web:	www.pco.de

pco.convert user manual 1.51.0

Released November 2023

Contents

1 General

This convert SDK description can be used to implement the PCO convert routines in proprietary applications, which are used to control PCO cameras. It is prohibited to use the convert routines with third party cameras.

The pco.convert sdk consists of two parts: The LUT conversion functions **pco.conv.dll** and the dialog functions **pco_cdlg.dll**.

The conversion functions are used to convert data areas, b/w and color, with a resolution of more than 8 bit per pixel to either b/w data areas with a resolution of 8 bit per pixel or color data areas with a resolution of 24 (32) bit per pixel. The DLL also includes functions to create and fill the various convert objects.

The second part of the API contains the dialog functions. The dialogs are simple GUI dialogs which enable the user to set the parameters of the convert objects. The dialog functions are included in the **pco_cdlg.dll** and are based on some functions of the **pco.conv.dll**.

In the pco.sdk for pco cameras there exist two samples, which make use of the convert sdk. One is the **Test_cvDlg** sample and the other is the **sc2_demo**. Please take a look at those samples in order to 'see' the convert sdk functions in action.

1.1 B/W And Pseudo Color Conversion

The conversion algorithm used in the b/w function is based on the following simple routine:

```
dataout[pos] = lutbw[datain[pos]];
```

where:

- pos is the counter variable
- dataout is the output data area
- datain is the input data area
- lutbw is a data area of size 2^n containing the LUT, where n = resolution of the input area in bits per pixel

In the pseudocolor function the basic routine to convert to a RGB data area is:

```
val = lutbw[datain[pos]];
dataout[pout + 0] = lutred[val];
dataout[pout + 1] = lutgreen[val];
dataout[pout + 2] = lutblue[val];
```

where:

- pos is the input counter variable
- pout is the output counter variable
- dataout is the output data area
- datain is the input data area
- lutbw is a data area of size 2^n containing the LUT, where n = resolution of the input area in bits per pixel

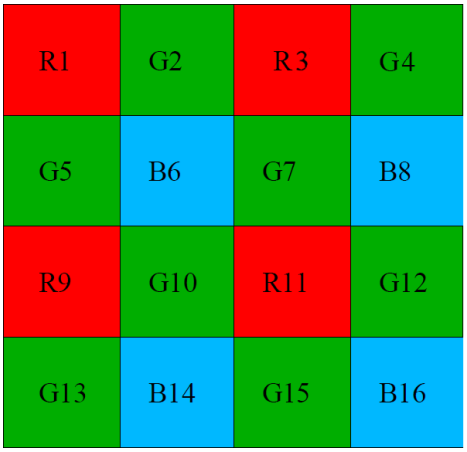
- lutred, lutgreen, lutblue are data areas of size 2^n containing the LUT, where n = resolution of the output area in bit per pixel.

1.2 Color Conversion

CCD color sensors used in PCO color cameras have filters for the colors red, green, and blue. Each pixel has one type of filter, thus originally you do not get full color information for each pixel. Rather each pixel delivers a value with a dynamic range of 12 bits for the color which passes the filter.

All color cameras at PCO work with the Bayer-filter demosaicking. The color filter pattern of those color image sensors can be reduced to a 2x2 matrix. The image sensor itself can be seen as a matrix of those 2x2 matrixes.

Suppose this color pattern:



The color itself is only an interpretation of the matrix. This interpretation will be done by a so called demosaicking algorithm. The pco_conv.dll works with a special proprietary method.

2 Convert API Function Description

2.1 PCO_ConvertCreate

Description Creates a new convert object based on the PCO_SensorInfo structure. The created convert handle will be used during the conversion. Please call PCO_ConvertDelete before the application exits and unloads the convert dll.

Prototype

```
int PCOCONVERT_API PCO_ConvertCreate (
    HANDLE* ph,                         //in out
    PCO_SensorInfo* strSensor,          //in
    int iConvertType                    //in
);
```

Parameter

Name	Type	Description
ph	HANDLE*	Pointer to a handle which will receive the created convert object
strSensor	PCO_SensorInfo*	Pointer to a sensor information structure. Please do not forget to set the wSize parameter.
iConvertType	int	Variable to determine the conversion type, either b/w, color, pseudo color or color 16

Return value

Name	Type	Description
ErrorMessage	int	0 in case of success, Errorcode otherwise.

2.2 PCO_ConvertDelete

Description Deletes a previously created convert object. It is mandatory to call this function before closing the application.

Prototype

```
int PCOCONVERT_API PCO_ConvertDelete (
    HANDLE ph                           //in
);
```

Parameter

Name	Type	Description
ph	HANDLE	Handle to a previously created convert object

Return value

Name	Type	Description
ErrorMessage	int	0 in case of success, Errorcode otherwise.

2.3 PCO_ConvertGet

Description Gets all the values of a previously created convert object.

Prototype

```
int PCOCONVERT_API PCO_ConvertGet (
    HANDLE ph,                      //in
    PCO_Convert* pstrConvert        //out
);
```

Parameter

Name	Type	Description
ph	HANDLE	Handle to a previously created convert object
pstrConvert	PCO_Convert*	Pointer to a pco convert structure

Return value

Name	Type	Description
ErrorMessage	int	0 in case of success, Errorcode otherwise.

2.4 PCO_ConvertSet

Description Sets necessary values for a previously created convert object.

Prototype

```
int PCOCONVERT_API PCO_ConvertSet (
    HANDLE ph,                      //in
    PCO_Convert* pstrConvert        //out
);
```

Parameter

Name	Type	Description
ph	HANDLE	Handle to a previously created convert object
pstrConvert	PCO_Convert*	Pointer to a pco convert structure

Return value

Name	Type	Description
ErrorMessage	int	0 in case of success, Errorcode otherwise.

2.5 PCO_ConvertGetDisplay

Description Gets the PCO_Display structure

Prototype

```
int PCOCONVERT_API PCO_ConvertGetDisplay (
    HANDLE ph,                          //in
    PCO_Display* pstrDisplay            //out
);
```

Parameter

Name	Type	Description
ph	HANDLE	Handle to a previously created convert object
pstrDisplay	PCO_Display*	Pointer to a pco display structure

Return value

Name	Type	Description
ErrorMessage	int	0 in case of success, Errorcode otherwise.

2.6 PCO_ConvertSetDisplay

Description Sets the PCO_Display structure

Prototype

```
int PCOCONVERT_API PCO_ConvertSetDisplay (
    HANDLE ph,                              //in
    PCO_Display* pstrDisplay                //in
);
```

Parameter

Name	Type	Description
ph	HANDLE	Handle to a previously created convert object
pstrDisplay	PCO_Display*	Pointer to a pco display structure

Return value

Name	Type	Description
ErrorMessage	int	0 in case of success, Errorcode otherwise.

2.7 PCO_ConvertSetBayer

Description Sets the bayer structure values of a previously created convert object. Use this functions to change the bayer pattern parameters.

Prototype

```
int PCOCONVERT_API PCO_ConvertSetBayer (
    HANDLE ph,                              //in
    PCO_Bayer* pstrBayer                    //in
);
```

Parameter

Name	Type	Description
ph	HANDLE	Handle to a previously created convert object
pstrBayer	PCO_Bayer*	Pointer to a PCO bayer structure

Return value

Name	Type	Description
ErrorMessage	int	0 in case of success, Errorcode otherwise.

2.8 PCO_ConvertSetFilter

Description Sets the filter structure values of a previously created convert object.

Prototype

```
int PCOCONVERT_API PCO_ConvertSetFilter (
    HANDLE ph,                              //in
    PCO_Filter* pstrFilter                  //in
);
```

Parameter

Name	Type	Description
ph	HANDLE	Handle to a previously created convert object
pstrFilter	PCO_Filter*	Pointer to a pco filter structure

Return value

Name	Type	Description
ErrorMessage	int	0 in case of success, Errorcode otherwise.

2.9 PCO_ConvertSetSensorInfo

Description Sets the PCO_SensorInfo structure for a previously created convert object

Prototype

```
int PCOCONVERT_API PCO_ConvertSetSensorInfo (
    HANDLE ph,                                  //in
    PCO_SensorInfo* pstrSensorInfo              //in
);
```

Parameter

Name	Type	Description
ph	HANDLE	Handle to a previously created convert object
pstrSensorInfo	PCO_SensorInfo*	Pointer to a sensor information structure. Please do not forget to set the wSize parameter

Return value

Name	Type	Description
ErrorMessage	int	0 in case of success, Errorcode otherwise.

2.10 PCO_SetPseudoLut

Description Load the three pseudolut color tables of plut

Prototype

```
int PCOCONVERT_API PCO_SetPseudoLut (
    HANDLE ph,                          //in
    unsigned char* pseudo_lut,          //in
    int inumcolors                      //in
);
```

Parameter

Name	Type	Description
ph	HANDLE	Handle to a previously created convert object
pseudo_lut	unsigned char*	Pointer to pseudo lut color values (R,G,B colors: 256 * 3 bytes, or 4 bytes)
inumcolors	int	Set to either 3 for R,G,B or 4 for R,G,B,A

Return value

Name	Type	Description
ErrorMessage	int	0 in case of success, Errorcode otherwise.

2.11 PCO_LoadPseudoLut

Description Loads a pseudo color lookup table to the convert object. This function can be used to load some of the predefined or self created pseudo lookup tables.

Prototype

```
int PCOCONVERT_API PCO_LoadPseudoLut (
    HANDLE ph,                          //in
    int format,                         //in
    char* filename                      //in
);
```

Parameter

Name	Type	Description
ph	HANDLE	Handle to a previously created convert object
format	int	0 lt1, 1 lt2, 2 lt3, 3 lt4
filename	char*	Name of the file to load

Return value

Name	Type	Description
ErrorMessage	int	0 in case of success, Errorcode otherwise.

2.12 PCO_Convert16TO8

Description Convert picture data in b16 to 8bit data in b8 (grayscale)

Prototype

```
int PCOCONVERT_API PCO_Convert16TO8 (
    HANDLE ph,                          //in
    int mode,                           //in
    int icolmode,                       //in
    int width,                          //in
    int height,                         //in
    word* b16,                          //in
    byte* b8                            //out
);
```

Parameter

Name	Type	Description
ph	HANDLE	Handle to a previously created convert object
mode	int	Mode parameter
icolmode	int	Color mode parameter
width	int	Width of the image to convert
height	int	Height of the image to convert
b16	word*	Pointer to the raw image
b8	byte*	Pointer to converted 8bit b/w image

Return value

Name	Type	Description
ErrorMessage	int	0 in case of success, Errorcode otherwise.

2.13 PCO_Convert16TO24

Description Convert picture data in b16 to 24bit data in b24 (grayscale)

Prototype

```
int PCOCONVERT_API PCO_Convert16TO24 (
    HANDLE ph,                          //in
    int mode,                           //in
    int icolmode,                       //in
    int width,                          //in
    int height,                         //in
    word* b16,                          //in
    byte* b24                           //out
);
```

Parameter

Name	Type	Description
ph	HANDLE	Handle to a previously created convert object
mode	int	Mode parameter

Continued on next page

Continued from previous page

Name	Type	Description
icolmode	int	Color mode parameter
width	int	Width of the image to convert
height	int	Height of the image to convert
b16	word*	Pointer to the raw image
b24	byte*	Pointer to converted 24bit color image

Return value

Name	Type	Description
ErrorMessage	int	0 in case of success, Errorcode otherwise.

2.14 PCO_Convert16TOCOL

Description Convert picture data in b16 to RGB data in b8 (color)

Prototype

```
int PCOCONVERT_API PCO_Convert16TOCOL (
    HANDLE ph,                          //in
    int mode,                           //in
    int icolmode,                       //in
    int width,                          //in
    int height,                         //in
    word* b16,                          //in
    byte* b8                            //out
);
```

Parameter

Name	Type	Description
ph	HANDLE	Handle to a previously created convert object
mode	int	Mode parameter
icolmode	int	Color mode parameter
width	int	Width of the image to convert
height	int	Height of the image to convert
b16	word*	Pointer to the raw image
b8	byte*	Pointer to converted 24bit color image

Return value

Name	Type	Description
ErrorMessage	int	0 in case of success, Errorcode otherwise.

2.15 PCO_Convert16TOPSEUDO

Description Convert picture data in b16 to pseudo color data in b8 (color)

Prototype

```
int PCOCONVERT_API PCO_Convert16TOPSEUDO (
    HANDLE ph,                          //in
    int mode,                           //in
    int icolmode,                       //in
    int width,                          //in
    int height,                         //in
    word* b16,                          //in
    byte* b8                            //out
);
```

Parameter

Name	Type	Description
ph	HANDLE	Handle to a previously created convert object
mode	int	Mode parameter
icolmode	int	Color mode parameter
width	int	Width of the image to convert
height	int	Height of the image to convert
b16	word*	Pointer to the raw image
b8	byte*	Pointer to converted 24bit pseudo color image

Return value

Name	Type	Description
ErrorMessage	int	0 in case of success, Errorcode otherwise.

2.16 PCO_Convert16TOCOL16

Description Convert picture data in b16 to RGB data in b16 (color)

Prototype

```
int PCOCONVERT_API PCO_Convert16TOCOL16 (
    HANDLE ph,                          //in
    int mode,                           //in
    int icolmode,                       //in
    int width,                          //in
    int height,                         //in
    word* b16in,                        //in
    word* b16out                        //out
);
```

Parameter

Name	Type	Description
ph	HANDLE	Handle to a previously created convert object
mode	int	Mode parameter

Continued on next page

Continued from previous page

Name	Type	Description
icolmode	int	Color mode parameter
width	int	Width of the image to convert
height	int	Height of the image to convert
b16in	word*	Pointer to the raw image
b16out	word*	Pointer to converted 48bit color image

Return value

Name	Type	Description
ErrorMessage	int	0 in case of success, Errorcode otherwise.

2.17 PCO_GetWhiteBalance

Description Gets white balanced values for `color_temp` and `tint`

Prototype

```
int PCOCONVERT_API PCO_GetWhiteBalance (
    HANDLE ph,                          //in
    int* color_temp,                    //out
    int* tint,                          //out
    int mode,                           //in
    int width,                          //in
    int height,                         //in
    WORD* gb12,                         //in
    int x_min,                          //in
    int y_min,                          //in
    int x_max,                          //in
    int y_max                           //in
);
```

Parameter

Name	Type	Description
ph	HANDLE	Handle to a previously created convert object
color_temp	int*	int pointer to get the calculated color temperature
tint	int*	int pointer to get the calculated tint value
mode	int	Mode parameter
width	int	Width of the image to convert
height	int	Height of the image to convert
gb12	WORD*	Pointer to raw picture data array
x_min	int	Rectangle to set the image region to be used for calculation
y_min	int	Rectangle to set the image region to be used for calculation
x_max	int	Rectangle to set the image region to be used for calculation
y_max	int	Rectangle to set the image region to be used for calculation

Return value

Name	Type	Description
ErrorMessage	int	0 in case of success, Errorcode otherwise.

2.18 PCO_GetMaxLimit

Description

GetMaxLimit gets the RGB values for a given temp and tint. The max value within the convert control dialog must not exceed the biggest value of the RGB values, e.g. in case R is the biggest value, the max value can increase till the R value hits the bit resolution (4095). Same condition must be met for decreasing the max value, e.g. in case B is the lowest value, the max value can decrease till the B value hits the min value.

Prototype

```
int PCOCONVERT_API PCO_GetMaxLimit (
    float* r_max,                    //out
    float* g_max,                    //out
    float* b_max,                    //out
    float temp,                      //in
    float tint,                      //in
    int output_bits                  //in
);
```

Parameter

Name	Type	Description
r_max	float*	Pointer to a float receiving the max red value
g_max	float*	Pointer to a float receiving the max green value
b_max	float*	Pointer to a float receiving the max blue value
temp	float	Color temperature
tint	float	Tint setting
output_bits	int	Bit resolution of the converted image (usually 8)

Return value

Name	Type	Description
ErrorMessage	int	0 in case of success, Errorcode otherwise.

2.19 PCO_GetColorValues

Description

Gets color temperature and tint for given R,G,B max values.

`GetColorValues` is used only in pco.camware. It calculates the color temperature and tint based on the Rmax,Gmax,Bmax values of the old color lut. The calculated values are used to convert old b16 and tif16 images with the new convert routines.

Prototype

```
int PCOCONVERT_API PCO_GetColorValues (
    float* pfColorTemp,                 //out
    float* pfColorTint,                 //out
    int iRedMax,                        //in
    int iGreenMax,                      //in
    int iBlueMax                        //in
);
```

Parameter

Name	Type	Description
pfColorTemp	float*	Pointer to a float for receiving the color temperature
pfColorTint	float*	Pointer to a float for receiving the color tint
iRedMax	int	Integer to set the current max value for red
iGreenMax	int	Integer to set the current max value for green.
iBlueMax	int	Integer to set the current max value for blue

Return value

Name	Type	Description
ErrorMessage	int	0 in case of success, Errorcode otherwise.

2.20 PCO_WhiteBalanceToDisplayStruct

Description Calculates the white balance and sets the values to the `strDisplay` struct while maintaining the limits. Gets the struct `strDisplay` from the convert `Handle` internally

Prototype

```
int PCOCONVERT_API PCO_WhiteBalanceToDisplayStruct (
    HANDLE ph,                                          //in
    PCO_Display* strDisplay,                            //out
    int mode,                                           //in
    int width,                                          //in
    int height,                                         //in
    WORD* gb12,                                         //in
    int x_min,                                          //in
    int y_min,                                          //in
    int x_max,                                          //in
    int y_max                                           //in
);
```

Parameter

Name	Type	Description
ph	HANDLE	Handle to a previously created convert object
strDisplay	PCO_Display*	Pointer to get the current values for display structure. This is necessary to get wScale_minmax and iScale_maxmax . Both values determine the range for iScale_min and iScale_max: 1...iScale_min...wScale_minmax; wScale_minmax...iScale_max...iScale_maxmax;
mode	int	Mode parameter
width	int	Width of the image to convert
height	int	Height of the image to convert
gb12	WORD*	Pointer to raw picture data array
x_min	int	Rectangle to set the image region to be used for calculation
y_min	int	Rectangle to set the image region to be used for calculation
x_max	int	Rectangle to set the image region to be used for calculation
y_max	int	Rectangle to set the image region to be used for calculation

Return value

Name	Type	Description
ErrorMessage	int	0 in case of success, Errorcode otherwise.

2.21 PCO_GetVersionInfoPCO_CONV

Description Returns version information about the dll.

Prototype

```
int PCOCONVERT_API PCO_GetVersionInfoPCO_CONV (
    char* pszName,                                  //out
    int iNameLength,                                //out
    char* pszPath,                                  //out
```

```
    int iPathLength,                                    //out
    int* piMajor,                                       //out
    int* piMinor,                                       //out
    int* piBuild                                        //out
);
```

Parameter

Name	Type	Description
pszName	char*	String to get the name of the module (can be NULL)
iNameLength	int	Length of the string in bytes (can be 0)
pszPath	char*	String to get the path of the module (can be NULL)
iPathLength	int	Length of the string in bytes (can be 0)
piMajor	int*	Pointer to int to get the major version number (can be NULL)
piMinor	int*	Pointer to int to get the minor version number (can be NULL)
piBuild	int*	Pointer to int to get the build version number (can be NULL)

Return value

Name	Type	Description
ErrorMessage	int	0 in case of success, Errorcode otherwise.

3 Typical Implementation

This typical step by step implementation shows the basic handling:

1. Declarations:

```
PCO_SensorInfo strsensorinf;
PCO_Display strDisplay;
```

2. Set all buffer 'size' parameters to the expected values:

```
strsensorinf.wSize = sizeof(PCO_SensorInfo);
strDisplay.wSize = sizeof(PCO_Display);
```

3. Set the sensor info parameters and create the convert object:

```
PCO_ConvertCreate(&hConvert,
    (PCO_SensorInfo*)&strsensorinf.wSize, ...);
```

4. Optionally open a convert dialog:

```
PCO_OpenConvertDialog(&m_hLutDialog, GetSafeHwnd(),
    "Convert Dialog", WM_APP+1011, m_hLut, 410, 252);
```

5. Set the min and max value to the desired range and set them to the convert object:

```
PCO_ConvertGetDisplay(hConvert,
    (PCO_Display*)&strDisplay.wSize, ...);
strDisplay.iScale_min = 200;
strDisplay.iScale_max = 2000;
PCO_ConvertSetDisplay(hConvert,
    (PCO_Display*)&strDisplay.wSize, ...);
```

6. Do the convert and set the data to the dialog if dialog is open:

```
PCO_Convert16TOCOL(hConvert, 0, 0, 1280, 1024, b16, b8rgb);
PCO_SetDataToDialog(hLutDialog, 1280, 1024, b16, b8rgb);
    // in realistic intervalls
```

7. Close the optionally opened convert dialog:

```
PCO_CloseConvertDialog(hConvertDialog);
```

8. Close the convert object:

```
PCO_ConvertDelete(hLut);
```

See the **Test_cvDlg** sample in the pco.sdk sample folder. Starting with v1.20, the range of the negative tint value has been doubled.

pco.

An Excelitas Technologies Brand

telephone:	+ 49 (0) 9441 2005 50
fax:	+ 49 (0) 9441 2005 20
postal address:	Excelitas PCO GmbH
	Donaupark 11
	93309 Kelheim, Germany
email:	pco@excelitas.com
web:	www.pco.de
	www.excelitas.com