Testo LabVIEW™ Library

**for Testo IR cameras**

**Testo 885**

**Testo 890**

**Version 1.0**

**11.11.2015**

**LabVIEW**™ **software was developed by National Instruments**

Contents

[Version 1.0 (11.11.2015) 3](#_Toc460574906)

[Range of functions in the Testo LabVIEW library 4](#_Toc460574907)

[Supported infrared cameras 4](#_Toc460574908)

[Supported operating systems 5](#_Toc460574909)

[Supported LabVIEW versions 5](#_Toc460574910)

[Required software 5](#_Toc460574911)

[Installing the Testo LabVIEW library 5](#_Toc460574912)

[Documentation of function calls 5](#_Toc460574913)

[Functions contained in the Testo LabVIEW library 6](#_Toc460574914)

[SubVI\_GetCamDevices.vi 6](#_Toc460574915)

[SubVI\_OpenCam.vi 6](#_Toc460574916)

[SubVI\_GetEmissivity.vi 6](#_Toc460574917)

[SubVI\_SetEmissivity.vi 6](#_Toc460574918)

[SubVI\_GetHumidity.vi 7](#_Toc460574919)

[SubVI\_SetHumidity.vi 7](#_Toc460574920)

[SubVI\_GetNumMeasRanges.vi 7](#_Toc460574921)

[SubVI\_GetMeasurementRange.vi 7](#_Toc460574922)

[SubVI\_SetMeasurementRange.vi 8](#_Toc460574923)

[SubVI\_GetReflectedTemperature.vi 8](#_Toc460574924)

[SubVI\_SetReflectedTemeperature.vi 8](#_Toc460574925)

[SubVI\_GetAtmosphereCorrection.vi 8](#_Toc460574926)

[SubVI\_ SetAtmosphereCorrection.vi 8](#_Toc460574927)

[SubVI\_CaptureIr.vi 9](#_Toc460574928)

[SubVI\_CaptureVIS.vi 11](#_Toc460574929)

[SubVI\_Get2DTemperatureArray.vi 11](#_Toc460574930)

[SubVI\_RGBConvertTo1DArray.vi 12](#_Toc460574931)

[SubVI\_StopStreamIR.vi 12](#_Toc460574932)

[SubVI\_StopStreamVIS.vi 12](#_Toc460574933)

[Demos 13](#_Toc460574934)

[1. Main IRVision 13](#_Toc460574935)

[Start group 13](#_Toc460574936)

[Status Indicator group 13](#_Toc460574937)

[Parameter group 14](#_Toc460574938)

[IR Image group 14](#_Toc460574939)

[2. VISImage 15](#_Toc460574940)

[Connection status 15](#_Toc460574941)

[Start group 15](#_Toc460574942)

[VIS Image group 15](#_Toc460574943)

[3. 3DSurface 16](#_Toc460574944)

[Connection status 16](#_Toc460574945)

[Start group 16](#_Toc460574946)

[3D Mesh 16](#_Toc460574947)

[Temperature Array 16](#_Toc460574948)

[4. GetSetParameter 17](#_Toc460574949)

[Connection status 17](#_Toc460574950)

[Start group 17](#_Toc460574951)

[Important information 18](#_Toc460574952)

[Communication with multiple cameras 18](#_Toc460574953)

# **Version 1.0 (11.11.2015**)

First version

# Range of functions in the Testo LabVIEW library

The library lets the LabVIEW programs communicate with the supported Testo infrared cameras t890 and t885. It allows for the reading and setting of parameters. It also enables reading of the VIS or IR image and the corresponding temperature values. To use the library, you must have a LabVIEW developer environment installed on your computer.

The package includes the following:

* Testo Toolbox for supported devices.
* VI files containing functions for working directly in LabVIEW.
* A VI file with a GUI that works with the entire range of functions contained in the library in order to access the camera and to read or display an IR image.
* Three demo VI files showing examples of using the various VI functions.
* This manual describing the range of functions contained in the LabVIEW library.

# Supported infrared cameras

The following IR cameras are supported:

* t885
* t890

The IR cameras are connected to the computer via USB port. In order to control the cameras, you must have the proper USB drivers installed.

# Supported operating systems

The following operating systems are supported:

* Microsoft Windows 7 (SP1 and higher), 32- and 64-bit
* Microsoft Windows 8, 8.1, 32- and 64-bit

## Supported LabVIEW versions

The following 32-bit LabVIEW versions are supported:

* LabVIEW 2011 SP1
* LabVIEW 2012 SP1
* LabVIEW 2013 SP1
* LabVIEW 2014 SP1

## Required software

Microsoft Visual Studio 2013 or Microsoft Visual C++ Redistributable Packages for Visual Studio 2013 (<https://www.microsoft.com/de-de/download/details.aspx?id=40784>)

testo IRSoft 3.8 or higher

(<http://www.testo.com/irsoft> )

## Installing the Testo LabVIEW library

Installation is performed using the included Setup program.

## Documentation of function calls

The calls in the Testo LabVIEW library are found in “LabVIEW\_cwrapper\_1\_0\_0.dll” and implemented as C interfaces. For each function, there exists a .vi file that calls the relevant function.

# Functions contained in the Testo LabVIEW library

## SubVI\_GetCamDevices.vi

This function checks what kind of cameras are connected to the computer and returns the number of cameras, an array with their serial numbers, and an array with a description of the cameras. Or it returns an error code. As shown in the demos, these values can be used to establish a connection to a particular camera.

|  |  |  |  |
| --- | --- | --- | --- |
| **Symbol** | **Parameter** | **Type** | **Description** |
| C:\Users\david.CADI\AppData\Local\Microsoft\Windows\INetCache\Content.Word\SerialS.PNG | u32NumberOfCameras **[out]** | U32 | Number of connected cameras |
| au32CameraSerials **[out]** | U32 1D array | Array containing serial numbers of the connected cameras |
| strCameraDeviceType **[out]** | string 1D Array | Array with the description of the cameras |
| U8Error **[out]** | U8 | Error code  0: Success  1: Error  2: no cameras connected |

## SubVI\_OpenCam.vi

This function opens a connection to the camera defined by the relevant serial number. The connection is automatically interrupted when another camera is connected or when the program is terminated.

|  |  |  |  |
| --- | --- | --- | --- |
| **Symbol** | **Parameter** | **Type** | **Description** |
|  | u32SerialNumber **[in]** | U32 | Serial number of the camera with which to establish a connection |
| U8Error **[out]** | U8 | Error code  0: Success  1: Error |

## SubVI\_GetEmissivity.vi

This function reads the emissivity value currently set in the camera.

|  |  |  |  |
| --- | --- | --- | --- |
| **Symbol** | **Parameter** | **Type** | **Description** |
| C:\Users\david.CADI\AppData\Local\Microsoft\Windows\INetCache\Content.Word\EmissionS.PNG | fEmissivity **[out]** | SGL | Currently set emissivity value |
| U8Error **[out]** | U8 | Error code  0: Success  1: Error |

## SubVI\_SetEmissivity.vi

This function sets the camera’s emissivity value to the value passed by the function.

|  |  |  |  |
| --- | --- | --- | --- |
| **Symbol** | **Parameter** | **Type** | **Description** |
| C:\Users\david.CADI\AppData\Local\Microsoft\Windows\INetCache\Content.Word\EmissionS.PNG | fEmissivity **[in]** | SGL | Desired emissivity value |
| U8Error **[out]** | U8 | Error code  0: Success  1: Error |

## SubVI\_GetHumidity.vi

This function reads the humidity value currently set in the camera.

|  |  |  |  |
| --- | --- | --- | --- |
| **Symbol** | **Parameter** | **Type** | **Description** |
| C:\Users\david.CADI\AppData\Local\Microsoft\Windows\INetCache\Content.Word\HumidityS.PNG | fHumidiy **[out]** | SGL | Current humidity value |
| U8Error **[out]** | U8 | Error code  0: Success  1: Error |

## SubVI\_SetHumidity.vi

This function sets the humidity value to the value passed by the function.

|  |  |  |  |
| --- | --- | --- | --- |
| **Symbol** | **Parameter** | **Type** | **Description** |
|  | fHumidity **[in]** | SGL | Desired humidity value |
| U8Error **[out]** | U8 | Error code  0: Success  1: Error |

## SubVI\_GetNumMeasRanges.vi

This function determines the number of possible measurement ranges. In order to select a particular measurement range, use the function ***SetMeasurementRange.vi***, which expects an input value of measurement range index < *U32NumOfMeasRanges*-1.

|  |  |  |  |
| --- | --- | --- | --- |
| **Symbol** | **Parameter** | **Type** | **Description** |
|  | U32NumMeasRanges **[out]** | U32 | Number of possible measurement ranges |
| U8Error **[out]** | U8 | Error code  0: Success  1: Error |

## SubVI\_GetMeasurementRange.vi

This function reads the index of the currently used measurement range.

|  |  |  |  |
| --- | --- | --- | --- |
| **Symbol** | **Parameter** | **Type** | **Description** |
| **C:\Users\david.CADI\AppData\Local\Microsoft\Windows\INetCache\Content.Word\MeasureRangeS.PNG** | u32MeasRangeIndex **[out]** | U32 | Index of the currently used measurement range |
| U8Error **[out]** | U8 | Error code  0: Success  1: Error |

## SubVI\_SetMeasurementRange.vi

This function selects the measurement range to be used. This is done by passing the index of the measurement range. The number of measurement ranges and which can be selected is determined by the function ***SubVI\_GetNrOfMeasRanges.vi***.

This function expects an input value of measurement range index < *U32NumOfMeasRanges*-1.

|  |  |  |  |
| --- | --- | --- | --- |
| **Symbol** | **Parameter** | **Type** | **Description** |
|  | U32MeasRangeIndex **[in]** | SGL | Index of measurement range to be used. |
| U8Error **[out]** | U8 | Error code  0: Success  1: Error |

## SubVI\_GetReflectedTemperature.vi

This function reads the reflected temperature value currently set in the camera.

|  |  |  |  |
| --- | --- | --- | --- |
| **Symbol** | **Parameter** | **Type** | **Description** |
|  | fReflectedTemp **[out]** | SGL | Currently set reflected temperature value |
| U8Error **[out]** | U8 | Error code  0: Success  1: Error |

## SubVI\_SetReflectedTemeperature.vi

This function sets the reflected temperature value to the value passed by the function.

|  |  |  |  |
| --- | --- | --- | --- |
| **Symbol** | **Parameter** | **Type** | **Description** |
| **C:\Users\david.CADI\AppData\Local\Microsoft\Windows\INetCache\Content.Word\ReflectTempS.PNG** | fReflectedTemp **[in]** | SGL | Desired reflected temperature value |
| U8Error **[out]** | U8 | Error code  0: Success  1: Error |

## SubVI\_GetAtmosphereCorrection.vi

This function reads the currently set atmospheric correction status.

|  |  |  |  |
| --- | --- | --- | --- |
| **Symbol** | **Parameter** | **Type** | **Description** |
| **C:\Users\hodel\AppData\Local\Microsoft\Windows\INetCache\Content.Word\atmoscorr.png** | bAtmosCorrEnable **[out]** | Boolean | Currently used status of the atmospheric correction value: TRUE – on; FALSE – off |
| U8Error **[out]** | U8 | Error code  0: Success  1: Error |

## SubVI\_ SetAtmosphereCorrection.vi

This function sets the status of the atmospheric correction.

|  |  |  |  |
| --- | --- | --- | --- |
| **Symbol** | **Parameter** | **Type** | **Description** |
| **C:\Users\hodel\AppData\Local\Microsoft\Windows\INetCache\Content.Word\atmoscorr.png** | u8AtmosCorrEnable **[in]** | U8 | Status of the atmospheric correction:  1 to turn on, 0 to turn off |
| U8Error **[out]** | U8 | Error code  0: Success  1: Error |

## SubVI\_CaptureIr.vi

This function reads the IR images recorded by the camera. To this end, three pointers to allocated *U8 1D arrays* are passed to the interface. These arrays are used to store the IR image’s RGB values. Using the RGB converter, this information can be used to display the IR image (see ***ExampleVI\_3DSurface.vi***). The color palette is chosen via the parameter *u8PaletteType*, which corresponds to the index of the chosen color palette:

|  |  |
| --- | --- |
| **Index** | **Description of color palette** |
| **1** | Iron |
| **2** | Rainbow |
| **3** | BlueRed |
| **4** | Testo |
| **5** | Sepia |
| **6** | Dewpoint |
| **7** | RainbowHC |

The choice of color palette relates only to the image’s display in LabVIEW and has no influence on how it is displayed in the camera.   
In addition, the function is passed an *SGL 1D array* in which the temperature values measured in the image are stored. The temperature values may be read via an *SGL 2D array* at the output.

**NOTE:** This VI **cannot** be executed at the same time as ***SubVI\_Get2DTemperatureArray.vi***.

|  |  |  |  |
| --- | --- | --- | --- |
| **Symbol** | **Parameter** | **Type** | **Description** |
|  | au8ChannelB **[in]** | U8 1D array | Allocated 1D array for the IR image’s B-channel |
| au 8ChannelG **[in]** | U8 1D array | Allocated 1D array for the IR image’s G-channel |
| au8ChannelR **[in]** | U8 1D array | Allocated 1D array for the IR image’s R-channel |
| af1DTempMat **[in]** | SGL 1D array | Allocated 1D buffer array for the IR image’s temperature values |
| u8PaletteType **[in]** | U8 | Index of the chosen color palette that is to be used to calculate the IR image. This influences only how the image is displayed in LabVIEW, not how it is displayed in the camera. |
| au8ChannelB **[out]** | U8 1D array | 1D array with the values for the IR image’s B-channel |
| au8ChannelG **[out]** | U8 1D array | 1D array with the values for the IR image’s G-channel |
| au8ChannelR **[out]** | U8 1D array | 1D array with the values for the IR image’s R-channel |
| af2DTempMat **[out]** | SGL 2D array | 2D array with the temperature values for each pixel of the IR image |
| i32Heigth **[out]** | I32 | Height of IR image |
| i32Width **[out]** | I32 | Width of IR image |
| U8Error **[out]** | U8 | Error code  0: Success  1: Error |

## SubVI\_CaptureVIS.vi

This function reads the VIS images recorded by the camera. To this end, pointers to three allocated *U8 1D arrays* are passed to the interface. These arrays are used to store the VIS image’s RGB values. Using the RGB converter, this information can be used to display the VIS image (see ***ExampleVI\_VISImage.vi***).

|  |  |  |  |
| --- | --- | --- | --- |
| **Symbol** | **Parameter** | **Type** | **Description** |
|  | au8ChannelB **[in]** | U8 1D array | Allocated 1D array for the VIS image’s B-channel |
| au8ChannelG **[in]** | U8 1D array | Allocated 1D array for the VIS image’s G-channel |
| au8ChannelR **[in]** | U8 1D array | Allocated 1D array for the VIS image’s R-channel |
| au8ChannelB **[out]** | U8 1D array | 1D array for the VIS image’s B-channel |
| au8ChannelG **[out]** | U8 1D array | 1D array for the VIS image’s G-channel |
| au8ChannelR **[out]** | U8 1D array | 1D array for the VIS image’s R-channel |
| i32VisHeigth **[out]** | I32 | Height of VIS image |
| i32VisWidth **[out]** | I32 | Width of VIS image |
| u8Error **[out]** | U8 | Error code  0: Success  1: Error |

## SubVI\_Get2DTemperatureArray.vi

This function reads the 2D temperature array recorded by the camera. To this end, an *SGL 1D array* for storing the values is passed to the interface. The temperature values may be read as a 2D array at the output. In addition, the function returns the current number of rows and columns in the 2D temperature array.

**NOTE:** This VI **cannot** executed at the same time as ***SubVI\_CaptureIrImage.vi***. If you wish to read the IR image as well as the temperature array, you must use ***SubVI\_CaptureIrImage.vi***.

|  |  |  |  |
| --- | --- | --- | --- |
| **Symbol** | **Parameter** | **Type** | **Description** |
|  | af1DTempMat **[in]** | SGL 1D array | Allocated 1D buffer array for the temperature values. |
| af2DTempMat **[out]** | SGL 2D array | 2D array containing the current image’s temperature values |
| i32Heigth **[out]** | I32 | Number of rows in the 2D temperature array |
| i32Width **[out]** | I32 | Number of columns in the 2D temperature array |
| u8Error **[out]** | U8 | Error code  0: Success  1: Error |

## SubVI\_RGBConvertTo1DArray.vi

This function uses the individual 1D arrays (R, G, B) to create a single 1D RGB array that is used to display the recorded images. The use of this function can be seen in the demos.

|  |  |  |  |
| --- | --- | --- | --- |
| **Symbol** | **Parameter** | **Type** | **Description** |
| **VISModeS** | au8ChannelB **[in]** | U8 1D array | 1D array containing the image’s B-channel |
| au8ChannelG **[in]** | U8 1D array | 1D array containing the image’s G-channel |
| au8ChannelR **[in]** | U8 1D array | 1D array containing the image’s R-channel |
| au8SerializedImage **[in]** | U8 1D array | Allocated 1D array for 1D RGB image to be converted |
| i32PixelAmount **[in]** | I32 | Total number of pixels in the image, width times height |
| au8SerializedImage **[out]** | U8 1D array | 1D Array with converted RGB image |

## SubVI\_StopStreamIR.vi

This function stops the streaming of IR images. In combination with ***CaptureIRImage.vi***, it makes it possible to take individual pictures without the delays caused by the caching of previous images. The use of this VI can be seen in ***Example\_3DSurface.vi***.

**NOTE:** Use this function **only** when recording individual images.

|  |  |  |  |
| --- | --- | --- | --- |
| **Symbol** | **Parameter** | **Type** | **Description** |
|  | U8Error **[out]** | U8 | Error code  0: Success  1: Error |

## SubVI\_StopStreamVIS.vi

This function stops the streaming of IR images. In combination with ***CaptureVISImage.vi***, it makes it possible to take individual pictures without the delays caused by the caching of previous images. The use of this VI can be seen in ***Example\_VISImage.vi***.

**NOTE:** Use this function **only** when recording individual images.

|  |  |  |  |
| --- | --- | --- | --- |
| **Symbol** | **Parameter** | **Type** | **Description** |
|  | U8Error **[out]** | U8 | Error code  0: Success  1: Error |

# Demos

The demos on the following pages are intended more for developing your own VIs and to facilitate the learning process.

## 1. Main IRVision

This demo shows almost all possibilities provided by the LabVIEW library. In this way, you can test how the camera can be controlled in LabVIEW.

****

### Start group

#### Camera selection

The combo box shows all connected cameras, including their serial numbers and descriptions. This field determines which camera is to be used. At startup, the program selects the first camera on the list.

#### CONNECT – establish connection

Opens a connection to the selected camera.

#### STREAM – transfer/playback

Starts the IR images’ real-time stream.

#### STOP – end stream

Click this button to end the stream. The display area shows the last image to be transferred.

#### SAVE – save image

Saves the current IR image as a JPG file.

#### EXIT – exit mode

Ends the application.

### Status Indicator group

* CONNECTION

Shows the connection status. The indicator is illuminated green if a connection exists.

* STREAM

Shows the stream status. The indicator is illuminated green if streaming is active.

### Parameter group

Shows those camera parameters that can be changed. At startup, the program reads the camera’s current parameter values. For more information on the individual parameters and their possible settings, see your camera manual.

**Note:** Some parameters require time for the changes to be enacted. No further IR images can be read during this time, and the camera’s operation is limited as well.

### IR Image group

This group contains the parameters of the IR image. The combo box is used to select the color palette. For more information on the individual color palettes, see your camera manual.

## 2. VISImage

This demo shows how to retrieve and display VIS images.



### Connection status

The indicator is illuminated light green if a connection exists.

### Start group

#### Camera selection

The combo box shows all connected cameras, including their serial numbers and descriptions. This field determines which camera is to be used. At startup, the program selects the first camera on the list.

#### CONNECT – establish connection

Opens a connection to the selected camera.

#### GETIMAGE

Reads the data of the current VIS image and displays it in the display area.

#### EXIT – exit mode

Ends the application.

### VIS Image group

Shows the VIS image’s dimensions in pixels.

## 3. 3DSurface

This demo shows how LabVIEW functions like 3DMesh are used to work with the temperature matrix and its corresponding IR image retrieved from the camera.



### Connection status

The indicator is illuminated light green if a connection exists.

### Start group

#### Camera selection

The combo box shows all connected cameras, including their serial numbers and descriptions. This field determines which camera is to be used. At startup, the program selects the first camera on the list.

#### CONNECT – establish connection

Opens a connection to the selected camera.

#### GETIMAGE

Reads the data of the current IR image and displays it in the display area.

#### EXIT – exit mode

Ends the application.

### 3D Mesh

LabVIEW graphs are used to show the measured temperature values as 3D plot. The Z axis shows the temperature values. The higher a measured value, the higher the amplitude displayed in the graph.

### Temperature Array

The matrix displays a part of the retrieved temperature matrix. In the upper left corner, (x, y) indices can be chosen to get the IR image’s corresponding temperature values at a certain position.

## 4. GetSetParameter

This simple demo shows how to retrieve and set the camera’s parameters.

****

### Connection status

The indicator is illuminated light green if a connection exists.

### Start group

#### Camera selection

The combo box shows all connected cameras, including their serial numbers and descriptions. This field determines which camera is to be used. At startup, the program selects the first camera on the list.

#### CONNECT – establish connection

Opens a connection to the selected camera.

#### EXIT – exit mode

Ends the application.

# Important information

## Communication with multiple cameras

When using the included VIs, always make sure that you are communicating with only one camera. Parallel communication with multiple cameras is not supported.