



Testo LabVIEW™ Bibliothek

für Testo IR-Kamera

testo 885

testo 890

Version 1.0

11.11.2015

LabVIEW™ ist eine Software der Firma National Instrument

Inhalt

VERSION 1.0 (11.11.2015)	3
FUNKTIONSUMFANG DER TESTO LABVIEW BIBLIOTHEK	4
UNTERSTÜTZTE INFRAROT KAMERAS	4
UNTERSTÜTZTE BETRIEBSSYSTEME	5
UNTERSTÜTZTE LABVIEW VERSIONEN	5
BENÖTIGTE SOFTWARE	5
INSTALLATION DER TESTO LABVIEW BIBLIOTHEK	5
DOKUMENTATION DER FUNKTIONSAUFRUFE	5
FUNKTIONSUMFANG DER TESTO LABVIEW BIBLIOTHEK	6
SUBVI_GETCAMDEVICES.VI	6
SUBVI_OPENCAM.VI	6
SUBVI_GETEMISSIVITY.VI	6
SUBVI_SETEMISSIVITY.VI	6
SUBVI_GETHUMIDITY.VI	7
SUBVI_SETHUMIDITY.VI	7
SUBVI_GETNUMMEASRANGES.VI	7
SUBVI_GETMEASUREMENTRANGE.VI	7
SUBVI_SETMEASUREMENTRANGE.VI	7
SUBVI_GETREFLECTEDTEMPERATURE.VI	9
SUBVI_SETREFLECTEDTEMEPERATURE.VI	9
SUBVI_GETATMOSPHERECORRECTION.VI	9
SUBVI_SETATMOSPHERECORRECTION.VI	9
SUBVI_CAPTUREIR.VI	10
SUBVI_CAPTUREVIS.VI	12
SUBVI_GET2DTEMPERATUREARRAY.VI	12
SUBVI_RGBCONVERTTO1DARRAY.VI	13
UBVI_STOPSTREAMIR.VI	13
SUBVI_STOPSTREAMVIS.VI	13
BEISPIELE	14

1. BEISPIEL MAIN IRVISION.....	14
<i>Gruppe Start</i>	14
<i>Gruppe Statusanzeige</i>	14
<i>Gruppe Parameter</i>	15
<i>Gruppe IR Image</i>	15
2. BEISPIEL VISIMAGE.....	16
<i>Verbindungsstatus</i>	16
<i>Gruppe Start</i>	16
<i>Gruppe VIS Image</i>	16
3. BEISPIEL 3DSURFACE	17
<i>Verbindungsstatus</i>	17
<i>Gruppe Start</i>	17
<i>Temperature Array</i>	17
4. BEISPIEL GETSETPARAMETER	18
<i>Verbindungsstatus</i>	18
<i>Gruppe Start</i>	18
WICHTIGE HINWEISE	19
KOMMUNIKATION MIT MEHREREN KAMERAS	19



Version 1.0 (11.11.2015)

ERSTE VERSION

Funktionsumfang der Testo LabVIEW Bibliothek

Die Bibliothek ermöglicht LabVIEW Programmen mit den unterstützten Testo Infrarot Kameras t890 und t885 zu kommunizieren. Sie erlaubt das Auslesen und Setzen von Parametern. Weiter ermöglicht sie das Auslesen des VIS- sowie des IR- Bildes und den dazugehörigen Temperaturwerte. Zur Verwendung benötigen Sie auf dem PC eine installierte LabVIEW Entwicklungsumgebung.

Im Lieferumfang enthalten sind:

- Die Testo Toolbox für die unterstützten Geräte.
- VI Dateien, die die bereitgestellten Funktionen für die direkte Verwendung in LabVIEW einbinden.
- Eine VI Datei mit GUI, die den gesamten Funktionsumfang, der durch die Bibliothek bereitgestellten Funktionen verwendet, um die Kamera anzusteuern und das IR Bild auszulesen bzw. darzustellen.
- Drei VI Demo Dateien, in welcher die Verwendung der einzelnen VI-Funktionen beispielhaft implementiert sind.
- Diese Dokumentation, welche den Funktionsumfang der LabVIEW Bibliothek beschreibt.

Unterstützte Infrarot Kameras

Folgende IR Kameras werden unterstützt:

- t885
- t890

Die IR Kameras werden über USB an den PC angeschlossen. Für die Ansteuerung ist die Installation der korrekten USB Treiber zwingend notwendig.



Unterstützte Betriebssysteme

Es werden folgende Betriebssysteme unterstützt:

- Microsoft Windows 7 (mindestens SP1), 32 und 64 Bit
- Microsoft Windows 8, 8.1, 32 und 64 Bit

Unterstützte LabVIEW Versionen

Es werden folgende LabVIEW 32-Bit Versionen unterstützt:

- LabVIEW 2011 SP1
- LabVIEW 2012 SP1
- LabVIEW 2013 SP1
- LabVIEW 2014 SP1

Benötigte Software

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

testo IIRSoft 3.8 oder höher

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

Installation der Testo LabVIEW Bibliothek

Für die Installation ist das mitgelieferte Setup auszuführen.


Dokumentation der Funktionsaufrufe

Die Aufrufe der Testo LabVIEW Bibliothek sind in der Datei LabVIEW_cwrapper_1_0_0.dll enthalten und als C-Schnittstelle implementiert. Für die relevanten Funktionen sind jeweils .vi Dateien vorhanden, welche die Aufrufe innerhalb von LabVIEW bereitstellen.

Funktionsumfang der Testo LabVIEW Bibliothek


SubVI_GetCamDevices.vi

Diese Funktion prüft, welche Kameras an den PC angeschlossen sind. Es wird die Anzahl, ein Array mit den Seriennummern und ein Array mit den Bezeichnungen der Kameras zurückgegeben. Des Weiteren kann ein Fehlercode ausgelesen werden. Diese Werte können, wie in den Beispielen demonstriert, verwendet werden, um eine Verbindung zu einer bestimmten Kamera aufzubauen.

Symbol	Parameter	Typ	Beschreibung
	u32NumberOfCameras [out]	U32	Anzahl der angeschlossenen Kameras
	au32CameraSerials [out]	U32 1D-Array	Array mit den Seriennummern der angeschlossenen Kameras
	strCameraDeviceType [out]	String 1D-Array	Array mit den Kamera Bezeichnungen
	U8Error [out]	U8	Fehlercode der Funktion 0: Erfolg 1: Fehler 2: keine Kamera angeschlossen


SubVI_OpenCam.vi

Diese Funktion öffnet eine Verbindung zu einer Kamera, welche durch die übergebene Seriennummer definiert ist. Die Verbindung wird automatisch durch das Verbinden mit einer anderen Kamera oder durch das Beenden des Programms getrennt.

Symbol	Parameter	Typ	Beschreibung
	u32SerialNumber [in]	U32	Seriennummer der Kamera zu der eine Verbindung geöffnet werden soll
	U8Error [out]	U8	Fehlercode der Funktion 0: Erfolg 1: Fehler


SubVI_GetEmissivity.vi

Diese Funktion liest den in der Kamera aktuell eingestellten Wert des Emissionsgrades aus.

Symbol	Parameter	Typ	Beschreibung
	fEmissivity [out]	SGL	Aktuell eingestellter Emissionsgrad
	U8Error [out]	U8	Fehlercode der Funktion 0: Erfolg 1: Fehler


SubVI_SetEmissivity.vi

Diese Funktion setzt den Emissionsgrad der Kamera auf den durch die Funktion übergebenen Wert.

Symbol	Parameter	Typ	Beschreibung
	fEmissivity [in]	SGL	Wert des Emissionsgrades, der eingestellt werden soll
	U8Error [out]	U8	Fehlercode der Funktion 0: Erfolg 1: Fehler


SubVI_GetHumidity.vi

Diese Funktion liest den in der Kamera aktuell eingestellten Wert der Luftfeuchtigkeit aus.

Symbol	Parameter	Typ	Beschreibung
	fHumidity [out]	SGL	Aktuell eingestellte Luftfeuchtigkeit
	U8Error [out]	U8	Fehlercode der Funktion 0: Erfolg 1: Fehler


SubVI_SetHumidity.vi

Diese Funktion setzt den Wert der Luftfeuchtigkeit auf den durch die Funktion übergebenen Wert.

Symbol	Parameter	Typ	Beschreibung
	fHumidity [in]	SGL	Wert der Luftfeuchtigkeit, der eingestellt werden soll
	U8Error [out]	U8	Fehlercode der Funktion 0: Erfolg 1: Fehler


SubVI_GetNumMeasRanges.vi

Diese Funktion ermittelt die Anzahl an auswählbaren Messbereichen. Für die Auswahl eines bestimmten Messbereichs ist die Funktion **SetMeasurementRange.vi** zu verwenden, die als Eingangswert einen Messbereichsindex < U32NumOfMeasRanges-1 erwartet.

Symbol	Parameter	Typ	Beschreibung
	U32NumMeasRanges [out]	U32	Anzahl wählbarer Messbereiche
	U8Error [out]	U8	Fehlercode der Funktion 0: Erfolg 1: Fehler


SubVI_GetMeasurementRange.vi

Diese Funktion liest den Index des aktuell eingestellten Messbereichs aus.

Symbol	Parameter	Typ	Beschreibung
	u32MeasRangeIndex [out]	U32	Index des aktuell Messbereichs
	U8Error [out]	U8	Fehlercode der Funktion 0: Erfolg 1: Fehler

SubVI_SetMeasurementRange.vi


Diese Funktion wählt den Messbereich aus, der verwendet werden soll. Dies erfolgt durch die Übergabe des Index des Messbereichs. Wie viele und welche Messbereiche ausgewählt werden können, geht aus der Funktion **SubVI_GetNrOfMeasRanges.vi** hervor.

Symbol	Parameter	Typ	Beschreibung
	U32MeasRangeIndex [in]	SGL	Index des Messbereichs, der verwendet werden soll
	U8Error [out]	U8	Fehlercode der Funktion 0: Erfolg 1: Fehler




SubVI_GetReflectedTemperature.vi

Diese Funktion liest den in der Kamera aktuell eingestellten Wert der reflektierten Temperatur aus.

Symbol	Parameter	Typ	Beschreibung
	fReflectedTemp [out]	SGL	Aktuell eingestellte reflektierte Temperatur
	U8Error [out]	U8	Fehlercode der Funktion 0: Erfolg 1: Fehler


SubVI_SetReflectedTemperature.vi

Diese Funktion setzt den Wert der reflektierten Temperatur auf den durch die Funktion übergebenen Wert.

Symbol	Parameter	Typ	Beschreibung
	fReflectedTemp [in]	SGL	Wert der reflektierten Temperatur, der eingestellt werden soll
	U8Error [out]	U8	Fehlercode der Funktion 0: Erfolg 1: Fehler


SubVI_GetAtmosphereCorrection.vi

Diese Funktion liest den eingestellten Status des Atmosphärenkorrekturwertes aus.

Symbol	Parameter	Typ	Beschreibung
	bAtmosCorrEnable [out]	Boolean	Status der Atmosphärenkorrektur: TRUE, falls ein- FALSE, falls ausgeschaltet
	U8Error [out]	U8	Fehlercode der Funktion 0: Erfolg 1: Fehler

SubVI_SetAtmosphereCorrection.vi

Diese Funktion setzt den Status des Atmosphärenkorrekturwertes.

Symbol	Parameter	Typ	Beschreibung
	u8AtmosCorrEnable [in]	U8	Status der Atmosphärenkorrektur: 1 falls eingeschaltet werden soll, 0 falls ausgeschaltet werden soll
	U8Error [out]	U8	Fehlercode der Funktion 0: Erfolg 1: Fehler

SubVI_CaptureIr.vi

Diese Funktion liest die durch die Kamera aufgenommenen IR-Bilder aus. Hierzu werden der Schnittstelle drei Pointer auf allokierte *U8 1D-Arrays* übergeben, in welche die RGB-Werte des IR-Bildes gespeichert werden. Unter Verwendung des RGB-Konverters kann hieraus das IR-Bild dargestellt werden (siehe Beispiel **ExampleVI_3DSurface.vi**). Die Wahl der Farbpalette erfolgt durch den Parameter *u8PaletteType*, der dem Index der gewählten Farbpalette entspricht:


Index	Bezeichnung Farbpalette
--------------	--------------------------------

- | | |
|----------|-----------|
| 1 | Iron |
| 2 | Rainbow |
| 3 | BlueRed |
| 4 | Testo |
| 5 | Sepia |
| 6 | Dewpoint |
| 7 | RainbowHC |

Die Wahl der Farbpalette bezieht sich nur auf die Darstellung des Bildes in LabVIEW und hat somit keinen Einfluss auf die Darstellung des Bildes in der Kamera.


Des Weiteren wird der Funktion ein *SGL 1D-Array* übergeben, in das die für das ausgelesene Bild gemessenen Temperaturwerte abgelegt werden. Auslesen lassen sich die Temperaturwerte am Ausgang über ein *SGL 2D-Array*.

ACHTUNG: Dieses VI darf **nicht** parallel zu **SubVI_Get2DTemperatureArray.vi** ausgeführt werden.

Symbol	Parameter	Typ	Beschreibung
	au8ChannelB [in]	U8 1D-Array	Allokiertes 1D-Array für den B-Kanal des IR-Bildes
	au8ChannelG [in]	U8 1D-Array	Allokiertes 1D-Array für den G-Kanal des IR-Bildes
	au8ChannelR [in]	U8 1D-Array	Allokiertes 1D-Array für den R-Kanal des IR-Bildes
	af1DTempMat [in]	SGL 1D-Array	Allokiertes Buffer-1D-Array für die Temperaturwerte des IR-Bildes
	u8PaletteType [in]	U8	Index der ausgewählten Farbpalette, die zur Berechnung des IR-Bildes verwendet werden soll. Dies hat keinen Einfluss auf die Darstellung des Bildes in der Kamera, sondern beeinflusst lediglich die Darstellung in LabVIEW.
	au8ChannelB [out]	U8 1D-Array	1D-Array mit den Werten des B-Kanals des IR-Bildes
	au8ChannelG [out]	U8 1D-Array	1D-Array mit den Werten des G-Kanals des IR-Bildes
	au8ChannelR [out]	U8 1D-Array	1D-Array mit den Werten des R-Kanals des IR-Bildes
	af2DTempMat [out]	SGL 2D-Array	2D-Array mit den Temperaturwerten für jedes Pixel des IR-Bildes
	i32Height [out]	I32	Höhe des IR Bildes
	i32Width [out]	I32	Breite des IR Bildes
	U8Error [out]	U8	Fehlercode der Funktion 0: Erfolg 1: Fehler

SubVI_CaptureVIS.vi


Diese Funktion liest die von der Kamera aufgenommenen VIS-Bilder aus. Hierzu wird der Schnittstelle jeweils ein Pointer auf drei allokierte *U8 1D-Arrays* übergeben, in welche die RGB-Werte des VIS-Bildes gespeichert werden. Unter Verwendung des RGB-Konverters kann hieraus das VIS-Bild dargestellt werden (siehe Beispiel *ExampleVI_VISImage.vi*).

Symbol	Parameter	Typ	Beschreibung
	au8ChannelB [in]	U8 1D-Array	Allokiertes 1D-Array für den B-Kanal des VIS-Bildes
	au8ChannelG [in]	U8 1D-Array	Allokiertes 1D-Array für den G-Kanal des VIS-Bildes
	au8ChannelR [in]	U8 1D-Array	Allokiertes 1D-Array für den R-Kanal des VIS-Bildes
	au8ChannelB [out]	U8 1D-Array	1D-Array mit den Werten des B-Kanals des VIS-Bildes
	au8ChannelG [out]	U8 1D-Array	1D-Array mit den Werten des G-Kanals des VIS-Bildes
	au8ChannelR [out]	U8 1D-Array	1D-Array mit den Werten des R-Kanals des VIS-Bildes
	i32VisHeight [out]	I32	Höhe des VIS Bildes
	i32VisWidth [out]	I32	Breite des VIS Bildes
	u8Error [out]	U8	Fehlercode der Funktion 0: Erfolg 1: Fehler

SubVI_Get2DTemperatureArray.vi


Diese Funktion liest die durch die Kamera aufgenommenen 2D Temperatur Matrix aus. Hierzu wird der Schnittstelle ein allokiertes *SGL 1D-Array* übergeben, in das die Werte gespeichert werden. Am Ausgang lassen sich die ermittelten Temperaturwerte als 2D Array auslesen. Weiter wird die aktuelle Zeilen- sowie Spaltenanzahl des 2D Temperaturarrays zurückgegeben.

ACHTUNG: Dieses VI darf **nicht** parallel zu *SubVI_CaptureIRImage.vi* ausgeführt werden. Falls sowohl das IR-Bild als auch die Temperaturmatrix ausgelesen werden soll, muss das *SubVI_CaptureIRImage.vi* verwendet werden.

Symbol	Parameter	Typ	Beschreibung
	af1DTempMat [in]	SGL 1D-Array	Allokiertes 1D-Buffer-Array für die Temperatur Werte.
	af2DTempMat [out]	SGL 2D-Array	2D-Array mit den Temperaturwerten des aktuellen Bildes
	i32Height [out]	I32	Zeilenanzahl der 2D Temperaturmatrix
	i32Width [out]	I32	Spaltenanzahl der 2D Temperaturmatrix
	u8Error [out]	U8	Fehlercode der Funktion 0: Erfolg 1: Fehler

SubVI_RGBConvertTo1DArray.vi


Diese Funktion erstellt aus einzelnen R-, G-, B- 1D Arrays ein einzelnes RGB 1D Array, das für die Darstellung der aufgenommenen Bilder verwendet wird. Die Verwendung dieser Funktion ist in den Beispielen ersichtlich.

Symbol	Parameter	Typ	Beschreibung
	au8ChannelB [in]	U8 1D-Array	1D-Array das den B-Kanal Bildes enthält
	au8ChannelG [in]	U8 1D-Array	1D-Array das den G-Kanal Bildes enthält
	au8ChannelR [in]	U8 1D-Array	1D-Array das den R-Kanal Bildes enthält
	au8SerializedImage [in]	U8 1D-Array	Allokiertes 1D Array für das zu konvertierende RGB 1D Bild
	i32PixelAmount [in]	I32	Gesamtanzahl der Pixel des Bildes, Breite mal Höhe
	au8SerializedImage [out]	U8 1D-Array	1D Array mit konvertiertem RGB Bild

SubVI_StopStreamIR.vi

Diese Funktion stoppt das Streaming von IR-Bildern. In Kombination mit dem VI **CaptureIRImage.vi** ermöglicht es die effektive Aufnahme von Einzelbildern ohne Verzögerungen, die durch die Zwischenspeicherung vorangegangener Bilder hervorgerufen werden. Die Verwendung dieses VI ist im Beispiel **Example_3DSurface.vi** ersichtlich.


ACHTUNG: Diese Funktion **nur** bei Aufnahme von Einzelbildern verwenden.

Symbol	Parameter	Typ	Beschreibung
	U8Error [out]	U8	Fehlercode der Funktion 0: Erfolg 1: Fehler

SubVI_StopStreamVIS.vi

Diese Funktion stoppt das Streaming von VIS-Bildern. In Kombination mit **CaptureVISImage.vi** ermöglicht es die effektive Aufnahme von Einzelbildern ohne Verzögerungen, die durch die Zwischenspeicherung vorangegangener Bilder hervorgerufen werden. Die Verwendung dieses VI ist im Beispiel **Example_VISImage.vi** ersichtlich.

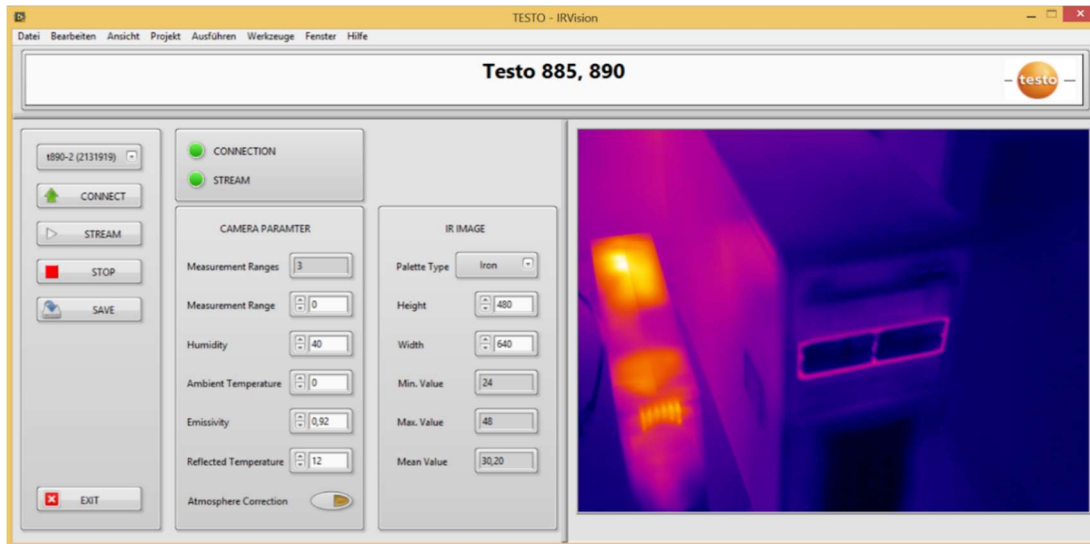
ACHTUNG: Diese Funktion **nur** bei Aufnahme von Einzelbildern verwenden.

Symbol	Parameter	Typ	Beschreibung
	U8Error [out]	U8	Fehlercode der Funktion 0: Erfolg 1: Fehler

Beispiele

1. Beispiel Main IRVision

In diesem Beispiel werden alle bereitgestellten Funktionen zur Kamerasteuerung verwendet. Es lassen sich Parameter setzen, abfragen und das IR-Bild auslesen. Die Oberfläche ist in drei Bereiche aufgeteilt, auf die im Folgenden eingegangen wird.



Gruppe Start

Kamera Auswahl

Im Kombinationsfeld werden alle angeschlossenen Kameras mit Ihren Seriennummern und Bezeichnungen aufgelistet. Hier wird festgelegt, welche Kamera verwendet werden soll. Bei Programmstart ist die erste Kamera der Liste dargestellt.

CONNECT – Verbindungsaufbau

Öffnet eine Verbindung zur gewählten Kamera.

STREAM – Übertragung/Wiedergabe

Startet den Echtzeitstream der IR-Bilder.

STOP - Stream beenden

Mit Drücken der Stop-Schaltfläche wird der Stream beendet. Das zuletzt übertragene Bild bleibt in der Anzeige erhalten.

SAVE – Ansicht speichern

Speichert das aktuelle IR-Bild als JPG-Datei.

EXIT – Modus beenden

Beendet die Anwendung.

Gruppe Statusanzeige

- CONNECTION
Zeigt den Status der Verbindung an. Besteht eine Verbindung, leuchtet die Anzeige grün.
- STREAM
Zeigt den Status des Streams an. Ist das Streaming aktiv wird leuchtet die Anzeige grün.

Gruppe Parameter

Es werden die Parameter der Kamera angezeigt, welche geändert werden können. Bei Programmstart werden die aktuellen eingestellten Parameterwerte der Kamera ausgelesen. Nähere Information zu den einzelnen Parametern und deren Einstellmöglichkeiten finden Sie im Handbuch Ihrer Kamera.

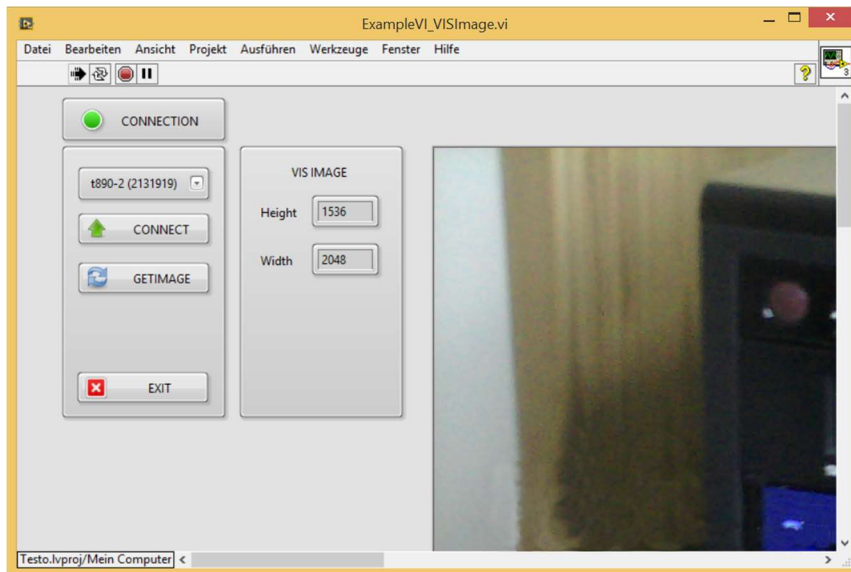
Hinweis: Einige Parameter benötigen Zeit, um sich den Änderungsvorgaben anzupassen. Währenddessen werden keine weiteren IR-Bilder ausgelesen. Auch die Kamerabedienung ist dann nur eingeschränkt möglich.

Gruppe IR Image

Es enthält die Parameter des IR-Bildes. Die Farbpalette kann über das entsprechende Kombinationsfeld ausgewählt werden. Nähere Hinweise zu den einzelnen Farbpaletten finden Sie im Handbuch Ihrer Kamera.

2. Beispiel VISImage

Dieses Beispiel zeigt, wie sich das VIS-Bild auslesen und darstellen lässt.



Verbindungsstatus

Die Anzeige leuchtet bei bestehender Verbindung in hellem Grün auf.

Gruppe Start

Kamera Auswahl

Im Kombinationsfeld werden alle angeschlossenen Kameras mit Ihren Seriennummern und Bezeichnungen aufgelistet. Hier wird festgelegt, welche Kamera verwendet werden soll. Bei Programmstart wird die erste Kamera der Liste ausgewählt.

CONNECT – Verbindungsaufbau

Öffnet eine Verbindung zur gewählten Kamera.

GETIMAGE

Liest Daten des aktuellen VIS Bildes aus und stellt dieses im Anzeigebereich dar.

EXIT – Modus beenden

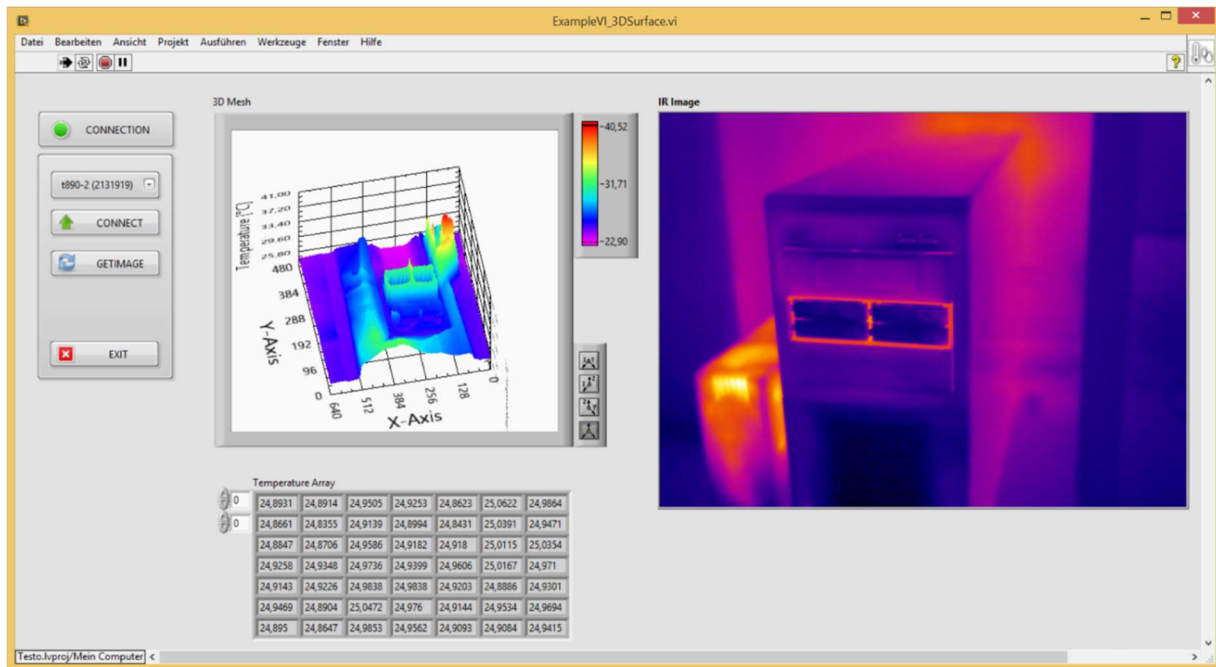
Beendet die Anwendung.

Gruppe VIS Image

Angabe der Abmessungen des VIS-Bildes in Pixeln.

3. Beispiel 3DSurface

In dem Beispiel wird exemplarisch gezeigt, wie sich die über die Kamera ausgelesenen Temperaturwerte weiterverarbeiten lassen, um diese mittels LabVIEWs 3DMesh als 3D-Plot darzustellen.



Verbindungsstatus

Die Anzeige leuchtet bei bestehender Verbindung in hellem Grün auf.

Gruppe Start

Kamera Auswahl

Im Kombinationsfeld werden alle angeschlossenen Kameras mit Ihren Seriennummern und Bezeichnungen aufgelistet. Hier wird festgelegt, welche Kamera verwendet werden soll. Bei Programmstart wird die erste Kamera der Liste ausgewählt.

CONNECT – Verbindungsaufbau

Öffnet eine Verbindung zur gewählten Kamera.

GETIMAGE

Liest Daten des aktuellen IR-Bildes aus und stellt dieses im Anzeigebereich dar.

EXIT – Modus beenden

Beendet die Anwendung.

3D Mesh-Anzeige

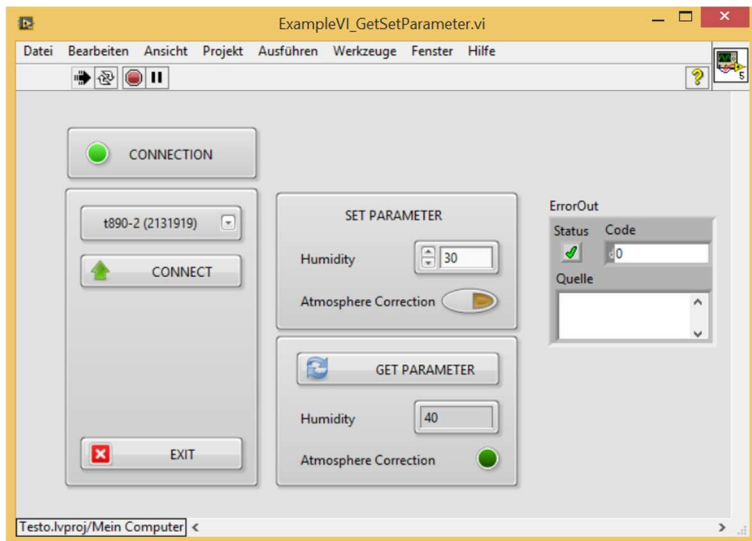
Die Temperaturwerte werden mittels LabVIEWs 3DMesh als 3D-Plot dargestellt. Die Temperaturwerte sind auf der z-Achse aufgetragen.

Temperature Array

Das Temperatur-Array zeigt einen Teil der aus der Kamera ausgelesenen Temperaturwerte in einer Matrix an. Diese Werte dienen als Grundlage für den 3D-Plot. Mit den linken Reitern lässt sich an die gewünschte Stelle (x, y) der Matrix springen. Der zugehörige Temperaturwert wird in der linken oberen Ecke der Matrix dargestellt.

4. Beispiel GetSetParameter

Dieses einfache Beispiel zeigt, wie die Parameter der Kamera abgerufen und gesetzt werden können.



Verbindungsstatus

Die Anzeige leuchtet bei bestehender Verbindung in hellem Grün auf.

Gruppe Start

Kamera Auswahl

Im Kombinationsfeld werden alle angeschlossenen Kameras mit Ihren Seriennummern und Bezeichnungen aufgelistet. Hier wird festgelegt, welche Kamera verwendet werden soll. Bei Programmstart wird die erste Kamera der Liste ausgewählt.

CONNECT – Verbindungsaufbau

Öffnet eine Verbindung zur gewählten Kamera.

EXIT – Modus beenden

Beendet die Anwendung.

Wichtige Hinweise

Kommunikation mit mehreren Kameras

Bei der Verwendung der mitgelieferten VI sollte zu jedem Zeitpunkt sichergestellt werden, dass nur mit einer Kamera kommuniziert wird. Die parallele Kommunikation mit mehreren Kameras wird nicht unterstützt.