Testo SE & Co. KGaA

API Documentation testo IR cameras

(supported cameras: testo 87x testo 883)
PART NUMBER: 0501 8987

Table of Content

1 History

Version 2.0.0

- Add support for testo 883
- Use opencv 4.1.2

Version 1.1.0

- Added active (bActiveConnectMode) and passive connect mode
- Added connection test on open camera to check if it is still connected
- Fixed reconnection interval for active connection handling
- Added exception type access on a disconnected camera

Version 1.0.0

- First release no changes to 0.8.2

Version 0.8.2

- Fixing function setMeasAreaState()

Version 0.8.1

- Avoid disconnect after parameter error in functions getFileContent()

Version 0.8.0

- Extended interface for measurement points and areas (add get, set, remove, activation functions)

Version 0.7.1

- Added stand-alone example for bmt

Version 0.7.0

- Added simple infrared streaming function.
- Fixed issue with call function getAvailableColorPalettes() before call getIrImageBgr()
- Removed updatePreview and getIrImagePreview (non-static) since the function can be replaced by using getIrImageBgr() directly

Version 0.6.1

- Fixed reconnect if connection is interrupted.

Version 0.6.0

- Implemented simple connection lost case and reconnect.
- Switched to opencv 3.2 with reduced modules.

2 Introduction

2.1 IrApi

The Testo *IrApi (order number 0501 8987)* is a program library which makes it possible to access the content of the Testo picture format (.bmt) from your own application program. It includes also an interface to connect to the camera over WLAN or USB and load stored bmt images. There is also a simple method implemented to stream directly the camera live infrared temperature data and infrared palletized image.

Main Features of *IrApi*:

- Read information like device name, serial number or temperature of a point
- View thermal images
- View visual images
- Connect to the camera and load picture files

2.2 Supported operating systems

The *IrApi* supports following operation systems:

- Windows 10 (tested)
- Linux (like Ubuntu?) except for testo 87x USB interface

Over platforms (like Linux, Mac, iOs, Android) could be supported on request.
For testo 883 cameras Windows and Linux platforms are supported for all interfaces.
For older testo 887x cameras Linux is not support for USB communication.

2.3 Supported languages

IrApi supports following programming languages:

- C++
- any language that can handle C++ object libraries can be adapted.

2.4 Supported cameras

IrApi supports testo cameras with following type name testo 883, testo 865, testo 868, testo 871 and testo 872. If you need support for other cameras, namely testo 880, testo 875, testo 882, testo 870, testo 885 and testo 890, you need the previous IrApi version (part number 0501 8984)!

3 Installation

There is no install process, just unzip/untar the package.

3.1 Prerequisites (Windows only)

If not already installed, the *IrApi* needs the following prerequisites under Windows:

- Redistributable Package from Microsoft vc_redist.x64.exe or vc_redist.x86.exe for Visual Studio 2015, 2017 und 2019 (all in one)
- Or Microsoft Visual Studio 2019 in order to directly compile and debug the included example.
- The API does not include the USB Driver install that is needed in order to use the USB interface for older t87x (it is not needed for the t88x and newer t87x USB interface). The USB driver can be installed by using the IrSoft Installer!
- Note: the Wifi-Interface (IP: 192.168.100.1) of all cameras and the USB RNDIS Interface (IP: 169.254.42.42) uses port 50000-50005 for communication. Make sure your firewall settings allow this connection.

3.2 Package components

The package contains the interface "*IrApi* " libraries for debug and release build. The windows packages include all dlls used for execution in the "bin" folder.

3.2.1 Third parties

Since the opencv Mat container is used in the interface a copy of the used opencv thrid party binary is induced in the package. (current version opencv 4.1.2)

3.2.2 CMake

If you use CMake the package contains CMake config scripts for the *IrApi* and *opencv* and there are corresponding find scripts in the folder "*cmake*".

3.2.3 Examples

Besides the files needed by the *IrApi* and this documentation you find a folder *example* in the package folder of *IrApi.* It contains the source code file for two examples "*example_bmt.cpp*" and "*example_cam.cpp"*. The executable *IrApiExampleBmt.exe" and* "IrApiExampleCam.exe" are located in "*irapi/bin/"* and should run out of the box.

3.3 How to generate the Visual Studio example project with cmake

3.3.1 With Visual Studio under Windows 10

The packages include all needed cmake defines to generate make files or Visual Studio project files in order to compile and debug the example.

1. To generate a Visual Studio solution with CMake, first install CMake (https://cmake.org/) and start the CMake-Gui.

2. Set the “Where is the source code” field to the *IrApi* example subdirectory and set the “Where to build the binaries” field to the directory the Visual Studio solution and the binaries should be generated into (out-of-place build).
3. Click on “Configure” to configure the project and select the Visual Studio version of the *IrApi (currently VS* 2019).
4. When finished, click “Generate” to generate the Visual Studio Solution.
5. Both examples should work “out of the box” if all points from 3.1 are considered.
   On a typical windows platform the firewall may block the TCP/IP connection for WiFi or USB RNDIS and needs the be set for specific for this connection or port (see 3.1)

3.3.2 Other platform

A make file for the system compiler project can be generated with cmake-gui the same way like for Visual Studio under Windows by choosing make file generator during configure step.

4 Interface functions

All temperature values of the interface are represented in degree Celsius!

The interface supports two main functions: Access *BMT* files from the camera and read/write a *BMT-files* file. The camera connection functions are provided by the class *irapi::Image* und the *BMT* file functions are provided by the class irapi::Image

4.1 Function overview Irapi::Cam

- connect to camera over an existing WLAN Connection to a camera hotspot
- serial number ad the device type name of the camera
- lists all *BMT* image files from the camera together with the file modification timestamp
- import function to download a single image file from the camera
- function load only the ir image preview data from one *BMT* image

4.2 Function overview Irapi::Image

- extract the preview image from a BMT image from file or memory
- camera serial number, camera device Name, capture timestamp
- extract palletized Ir-Image in BGR format (opencv compatible)
- extract visual image if available (small size and full size)
- set and get functions for environmental parameter that includes:
  - emissivity (range: 0...1),
  - reflected temperature
  - ambient temperature
  - humidity (relative range: 0...100%)
- read and change the scaling modes and parameter, supported modes are:
  - auto scale
  - manual scale

    - scale assist
    - humidty mode
- read cursor information (middle cursor, hot and cold spot, deltaT if active)
- activate hot, cold spot
- read measurement area information if are is active
- activate measurement area
- get information if Superresolution was used
- save image to file or export data to memory
- get active measurement application and it's value (additional measurement values from clamp meter or manual insert values during image capture)

5 Getting started under Windows

There are components which need to load in your application target directory. Otherwise the application will fail to call any *IrApi* functions.

5.1 Way to get it run

Please ensure that your application finds the libraries files originally located in the package of the *IRApi*: Depending on you build type only debug or release version is needed (see chapter 2.2 for a file list). You can do that by editing the global path variable or by simply copying all the files to the folder your application is running in. Also it is advised to use the opencv version provided by the package.

5.2 Examples

Location: "example*example_cam.cpp*"and "example*example_bmt.cpp*".

The Example *example_cam* shows the basic usage of the *irapi::Image* and the *irapi::Cam* object. It list all bmt images from the camera and loads one of them into an *irapi::Image* object. For more detailed information please read the comments from the source code file.

To run the example successfully a WLAN connection to the camera must be established manual: Please activate WLAN on camera and connect your device to the camera Hotspot. Also make sure your firewall settings allow the example program build a connection with the camera.

The example *example_bmt* shows the general usage of the *irapi::Image* object. It does open the bmt file "*IR_EXAMPLE.BMT"* located in the source code directory (it was built from) or any other .bmt file passed as a command line argument when executed.