OPERATING INSTRUCTIONS

SICK Vision Suite User Manual 1.1

SICK

Sensor Intelligence.

Contents

1 Preface 3

2 Notes on usage 4

3 Introduction 5

4 Basic terms 6

4.1 GenICam 6

4.2 GenAPI 6

4.3 SFNC 6

4.4 GenTL 7

4.5 GenTL modules 7

4.6 GenTL SFNC 8

5 Terms and abbreviations 9

6 SICK Vision Suite 10

7 Components 11

8 Applications in SICK Vision Suite 12

8.1 SICK Vision Cockpit 12

8.2 Command line tools 12

8.2.1 Configuring a camera 13

8.2.2 Configuring network settings 13

8.2.3 Updating the camera firmware 14

8.3 Executable samples 15

1 Preface

Introduction

SICK AG has taken every possible care in preparing this manual. We however assume no liability for the content, completeness or quality of the information contained therein. The content of this manual is regularly updated and adapted to reflect the current status of the software. We furthermore do not guarantee that this product will function without errors, even if the stated specifications are adhered to.

Under no circumstances can we guarantee that a particular objective can be achieved with the purchase of this product.

Insofar as permitted under statutory regulations, we assume no liability for direct damage, indirect damage or damages suffered by third parties resulting from the purchase of this product. In no event shall any liability exceed the purchase price of the product.

Please note that the content of this manual is neither part of any previous or existing agreement, promise, representation or legal relationship, nor an alteration or amendment thereof. All obligations of SICK AG result from the respective contract of sale, which also includes the complete and exclusively applicable warranty regulations. These contractual warranty regulations are neither extended nor limited by the information contained in this manual. Should you require further information on this product, or encounter specific problems that are not discussed in sufficient detail in the manual, please contact your local dealer or system installer.

Trademarks

Microsoft and Windows are trademarks or registered trademarks of Microsoft Corporation. All other products or company names mentioned in this manual are used solely for purposes of identification or description and may be trademarks or registered trademarks of the respective owners.

Copyright

© SICK AG. All rights reserved. This manual may not be reproduced, transmitted or translated to another language, either as a whole or in parts, without the prior written permission of SICK AG.

Status: April 2020

Contact

Visit our web site www.sick.com where you will find all the latest information about our software and hardware products.

Address	SICK AG Erwin-Sick-Str. 1 D-79183 Waldkirch, Germany
T	+49 7681 202-0
E	info@sick.com
W	www.sick.com

2 Notes on usage

ℹ This symbol indicates hints with useful information for better understanding and using features and functions.

⛔ This symbol indicates important warnings for product safety to prevent damage.

⚠ This symbol indicates important warnings for personal safety to prevent injury.

3 Introduction

SICK Vision Suite is a comprehensive software package from SICK AG that can be used with GenICam-compliant industrial cameras. SICK Vision Suite provides all necessary tools to open cameras in an application with graphical user interface, to parametrize them, to capture images, etc. or to program your own application.

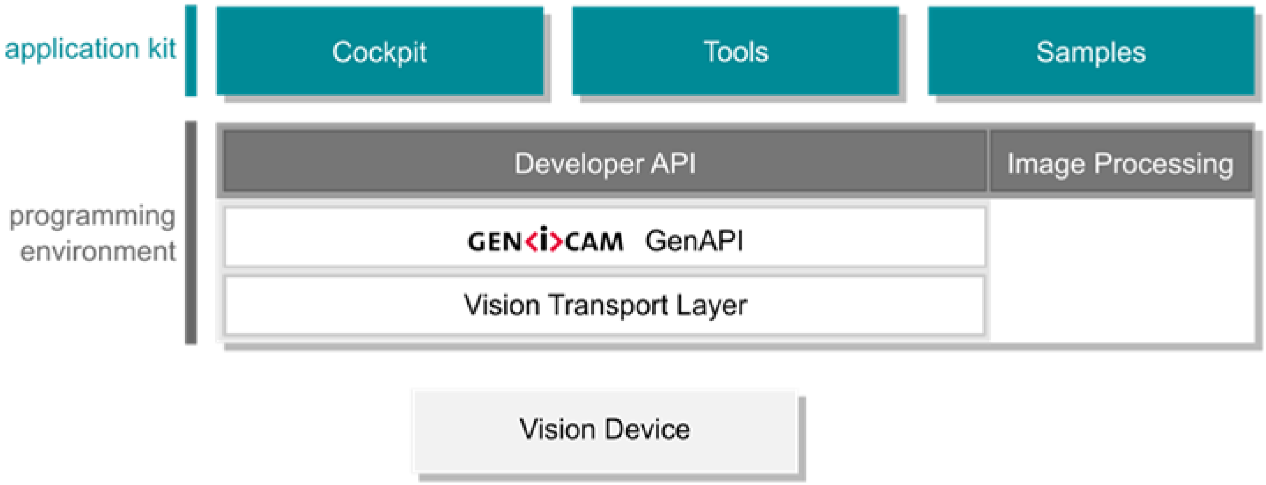


Fig. 1: SICK Vision Suite

4 Basic terms

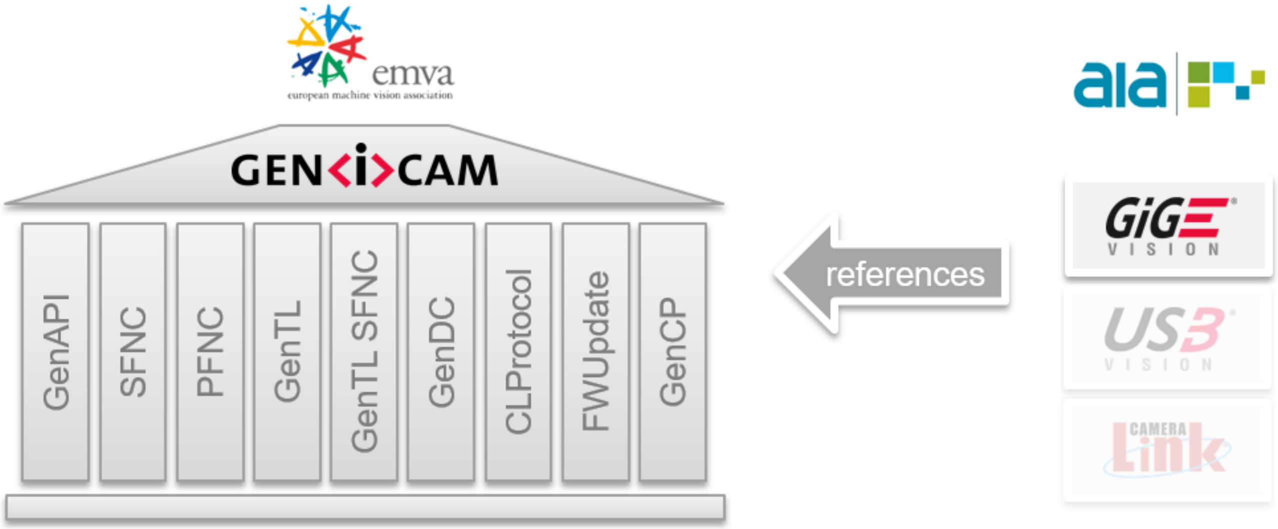


Fig. 2: GenICam and Vision standards

- GenICam
- GenAPI
- SFNC
- GenTL
- GenTL modules
- GenTL SFNC

4.1 GenICam

GenICam (Generic Interface for Cameras) is a generic programming interface for cameras in industrial image processing. The GenICam standard is supervised by EMVA (European Machine Vision Association). The current version of the GenICam definition and a reference implementation of the GenAPI are available on the web site www.genicam.org. There you will also find further information about the structure of GenICam standard.

4.2 GenAPI

Within the GenICam standard, the GenAPI (Generic Application Programming Interface) is used for camera configuration, camera access and camera control. The GenAPI reads the description files (camera XML) of the GenICam-compliant cameras describing which features the cameras support.

4.3 SFNC

The SFNC (Standard Features Naming Convention) defines camera features and properties and their names that can be included in the camera XML. The camera features are implemented in the firmware, i.e. the camera is not dependent on drivers installed on the PC. The features are grouped under different nodes in a tree and follow a well-defined access scheme. Each feature is clearly described:

- what type it is (integer, floating point, string, etc.)
- what the feature can do (feature description)
- which are the minimum and maximum values
- which discrete parameters exist for some features ("enumeration")
- which feature affects other features
  Example: adjusting the exposure time, for example, changes the maximum possible frame rate. In the

camera XML it is specified that the frame rate should be read out again in case of an exposure time change.

- etc.

This exact description allows to dynamically read and display the current state of the camera and its features at runtime. Normally, this is done in an application via a graphical tree view. Since each feature also specifies which value is below, the control elements can be chosen accordingly in the application and the tree can be built dynamically. This behavior is referred to as "generic".

4.4 GenTL

The GenTL (Generic Transport Layer) is a layer that is responsible for the transport of the camera data. The GenTL (or transport layer) converts the commands of the GenAPI and/or the application (consumer) into commands for the camera driver. Furthermore, the GenTL is responsible for submitting the camera XML to the GenAPI.

It is differentiated between a **GenTL Producer** and a **GenTL Consumer:**

- The GenTL Producer is the software that accesses the camera and provides the image data for an application. A GenTL Producer is provided as a platform-dependent, dynamically loadable library. The file extension of this library is *.cti (Common Transport Interface).
- The GenTL Consumer is a software e.g. an application that receives the data.

4.5 GenTL modules

The GenTL consists of different modules, which are represented in a hierarchical tree structure with the system module as root.

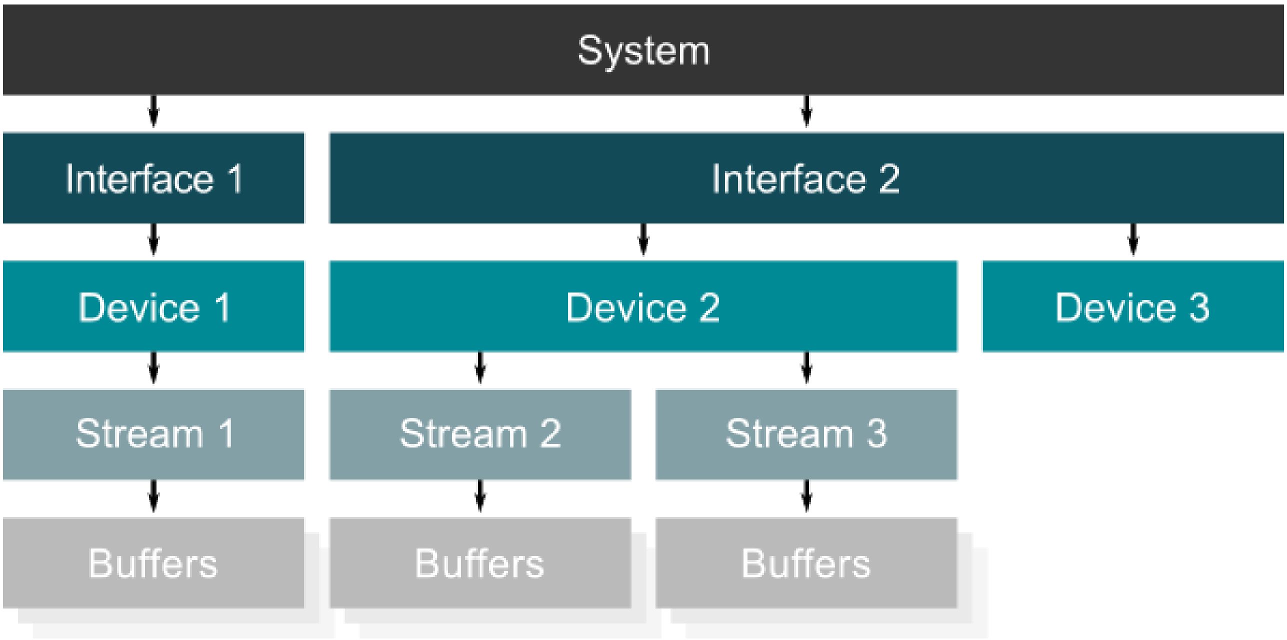


Fig. 3: GenTL modules

- System module
  The system module is called once for each GenTL. It is the entry point for a GenTL Producer from which the subsequent modules are called.
- Interface module
  An interface module represents a specific hardware interface, e.g. a network card for an Ethernet-based GenTL.
- Device module
  You can access multiple devices through an interface. These are the actually available devices e.g. cameras. A device can only be opened once with full access control.

© SICK AG • Subject to change without notice • 8025622/0000

- Stream module
  To capture images from a camera, for example, an image data stream has to be initialized in that device. The GenTL can capture multiple streams per device if the device supports it.

- Buffer module
  The buffers are the image memories into which the captured images are written. The image memories can be allocated either by the GenTL Consumer or GenTL Producer.

4.6 GenTL SFNC

The GenTL standard defines a set of feature names and their definitions. The GenAPI module is used to access these features. The GenTL SFNC (GenTL Standard Feature Naming Convention) standardizes the used features in order to separate the GenTL standard as far as possible from the definition of specific features.

The GenTL SFNC does not replace the features that are defined in the regular GenICam SFNC, but enhances them by explicitly covering only the features of the GenTL producer itself.

5 Terms and abbreviations

CTI	Common transport interface
Configuration	Module configuration via the GenTL port functions, a GenAPI-compliant XML description, and the GenTL Standard Features Naming Convention.
GenAPI	GenICam module that defines an XML Schema used to describe register tabs.
GigE	Gigabit Ethernet
GenICam	Generic interface for cameras
GenTL	Generic transport layer
GenTL Consumer	A library or application that uses an implementation of a transport layer interface.
GenTL Producer	Implementation of a transport layer interface
GenTL SFNC	GenICam module: GenTL Standard Features Naming Convention
SFNC	GenICam module: Standard Features Naming Convention
Signaling	Mechanism for notifying the requesting GenTL consumer of an asynchronous event.
TLI	Generic transport layer interface
USB	Universal serial bus

6 SICK Vision Suite

SICK Vision Suite components
SICK Vision Suite SDK • SICK Vision API (Application Programming Interface) • SICK LibIMG (Image Processing Library)
Samples (source code) • SICK Vision Suite samples • HALCON samples • MIL samples (Matrox Imaging Library)
SICK Vision TL • SICK GenICam Producer (GEVK) • SICK GenICam Producer (GEV)
SICK Vision Cockpit
SICK Vision Suite tools • sick_devicecommand • sick_deviceupdate • sick_ipconfig
Samples (binaries) • SICK Vision Suite samples

7 Components

SICK Vision Suite provides all necessary libraries and software interfaces for application developers and thus forms a complete SDK (software development kit). SICK Vision Suite is based on the module hierarchy defined by the standards (GenICam, GenTL). At the same time, the interface is considerably easier to operate without limiting the functionality of the standardized interface. SICK Vision Suite can be used with the C#, C++, and C programming languages. The use of C# or C++ is recommended.

In addition, SICK Vision Suite combines GUI-based applications and practical command line tools. The components in SICK Vision Suite are:

SICK Vision API

Application programming interface that provides convenient access to all associated libraries (GenAPI, GenTL, etc.). The core task of the SICK Vision API is the communication with the camera, the camera parametrization and the transfer of the image data to the computer.

The SICK Vision API is described in a separate documentation.

SICK LibIMG

This is a library for high-performance image processing on the computer (Image Processing Library). The SICK LibIMG can be used, for example, to convert camera image that were captured via the SICK Vision API from raw bayer format into color (debayering).

The SICK LibIMG is described in a separate documentation.

Samples (source code)

All samples are delivered open as source code in SICK Vision Suite. Thus, the samples can also be used as starting point for your own programming.

Under Windows, the samples are also included as executable files (binaries) and can be executed directly.

Transport layer

The transport layers are responsible for transferring the camera data to the user application. It is a low-level hardware interface for translating API commands into camera understandable commands.

- SICK GenICam Producer (GEVK)
  GenTL Producer (required for image processing programs based on the GenICam interface) including a kernel driver to improve Ethernet performance. The use of this GenTL Producer is recommended.
- SICK GenICam Producer (GEV)
  GenTL Producer (required for image processing programs based on the GenICam interface) This GenTL Producer can be installed as an alternative to "SICK GenICam Producer (GEVK)" if it is not possible to install a kernel driver on the system.

SICK Vision Cockpit

Graphical user interface for camera parametrization and live display of images. See SICK Vision Cockpit

Tools

Collection of useful command line programs, e.g. to update the camera firmware. See Command line tools

Samples (binaries)

Collection of sample programs that are offered both in source code and as executable files and can therefore be used directly. See Executable samples

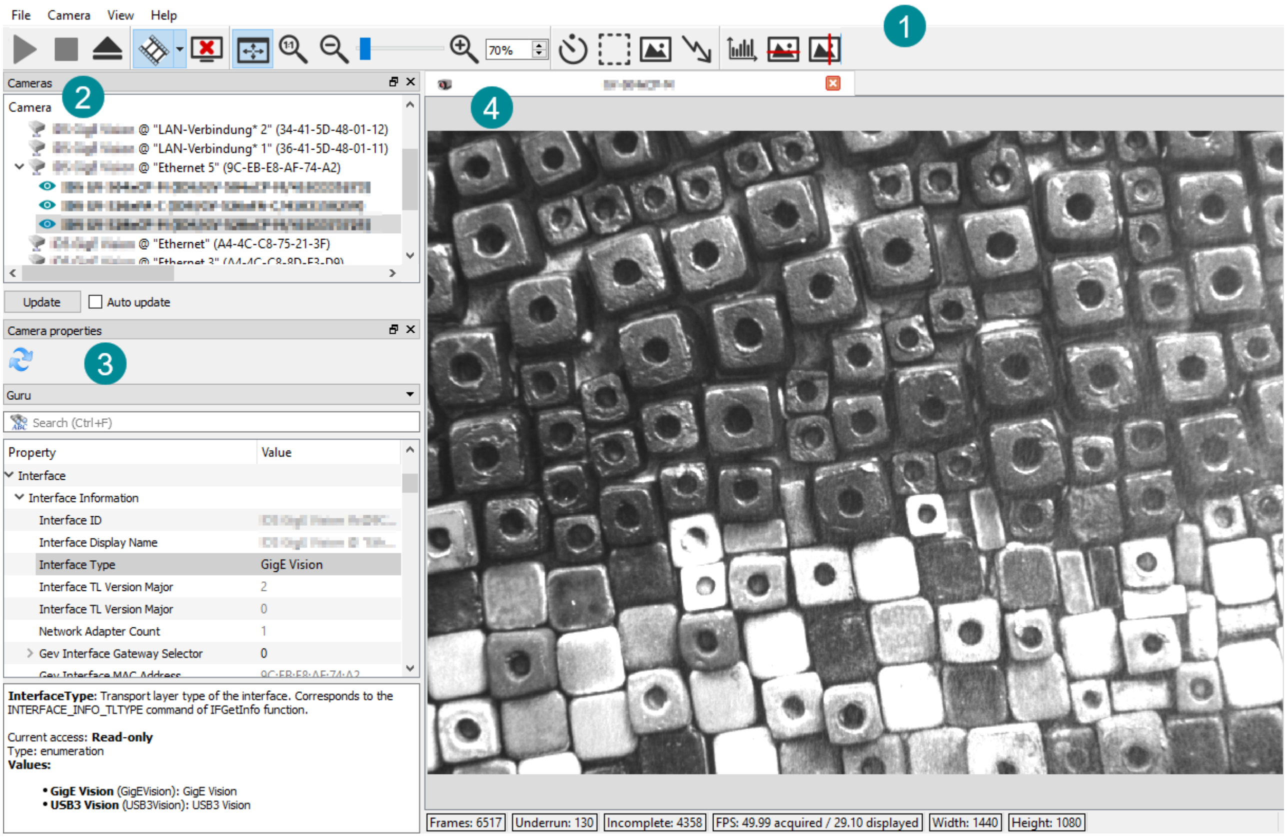
8 Applications in SICK Vision Suite

- SICK Vision Cockpit
- Command line tools
- Executable samples

8.1 SICK Vision Cockpit

The SICK Vision Cockpit provides you with a graphical interface to quickly and easily evaluate cameras without programming a line of code. In addition to display all camera parameters in a GenICam-typical tree view, the SICK Vision Cockpit provides you with a lot of practical image information and dialog-based settings.

1. Menu and symbol bar, e.g. open camera, start/stop image acquisition or change camera settings
2. Camera list with the available GenTL modules (transport layers, interfaces, cameras)
   In the camera list, for example, use the context menu to open the dialog for configuring the IP address of a GigE Vision camera.
3. Camera properties in the tree views with filtering by beginner, expert, guru and search term
4. Camera image with information display e.g. of the captured frames, frame rate (fps) and size (height and width in pixels)



Camera

- [redacted] @ "LAN-Verbindung* 2" (34-41-5D-48-01-12)
- [redacted] @ "LAN-Verbindung* 1" (36-41-5D-48-01-11)
- [redacted] @ "Ethernet 5" (9C-EB-E8-AF-74-A2)
  - [redacted]
  - [redacted]
  - [redacted]
- [redacted] @ "Ethernet" (A4-4C-C8-75-21-3F)
- [redacted] @ "Ethernet 3" (A4-4C-C8-8D-F3-D9)

Property	Value
Interface	
Interface Information	
Interface ID	[redacted]
Interface Display Name	[redacted]
Interface Type	GigE Vision
Interface TL Version Major	2
Interface TL Version Major	0
Network Adapter Count	1
Gev Interface Gateway Selector	0
Gev Interface MAC Address	9C:EB:E8:AF:74:A2

InterfaceType: Transport layer type of the interface. Corresponds to the INTERFACE_INFO_TLTYPE command of IFGetInfo function.

Current access: **Read-only**
Type: enumeration
Values:

- **GigE Vision** (GigEVision): GigE Vision
- **USB3 Vision** (USB3Vision): USB3 Vision

Frames: 6517 | Underrun: 130 | Incomplete: 4358 | FPS: 49.99 acquired / 29.10 displayed | Width: 1440 | Height: 1080

Fig. 4: SICK Vision Cockpit

8.2 Command line tools

Use the handy command line tools e.g. to automatically configure several cameras.

The **sick_cmdtools** shortcut gives you quick access to the command line tools and the SICK Vision Cockpit.

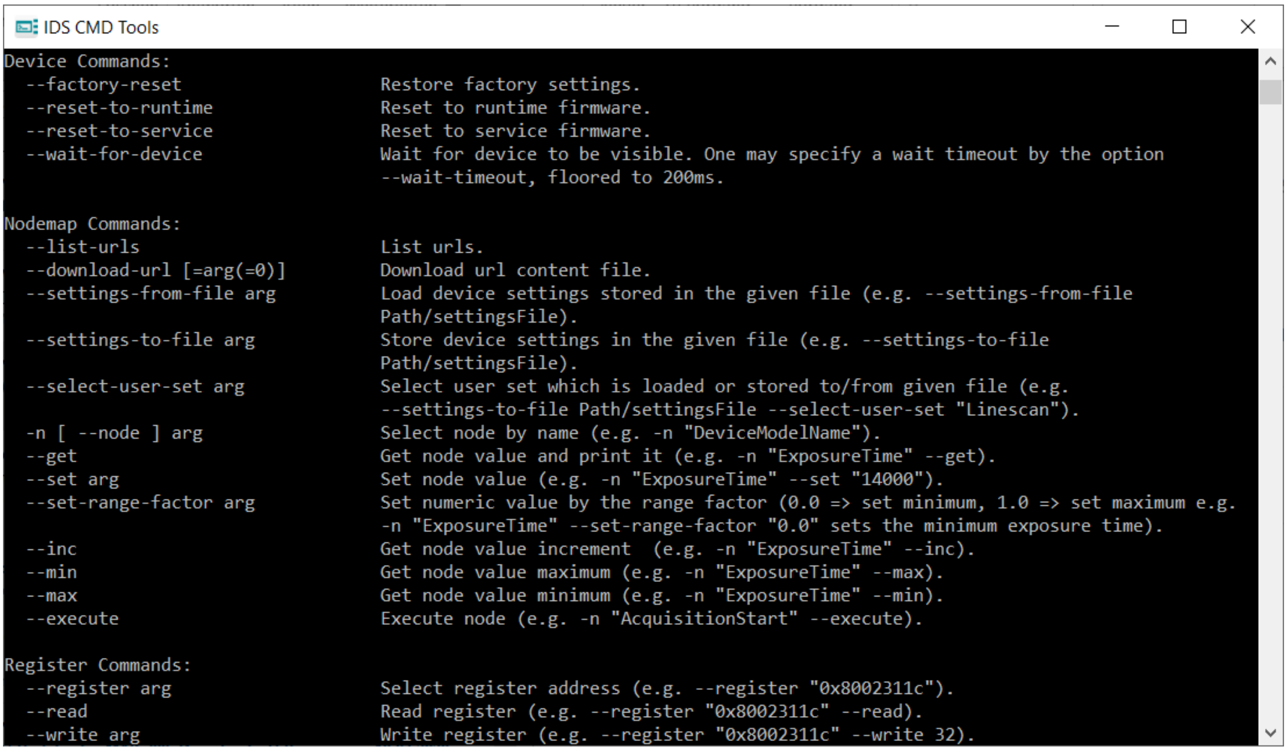
- Configuring a camera
- Configuring network settings
- Updating the camera firmware

8.2.1 Configuring a camera

sick_devicecommand provides you with information about all available camera settings via the GenICam interface and allows you to configure the camera directly. This tool is intended for expert users.

You can list further options with:

```
sick_devicecommand --help
```



```
Device Commands:
  --factory-reset                    Restore factory settings.
  --reset-to-runtime                 Reset to runtime firmware.
  --reset-to-service                 Reset to service firmware.
  --wait-for-device                  Wait for device to be visible. One may specify a wait timeout by the option
                                     --wait-timeout, floored to 200ms.

Nodemap Commands:
  --list-urls                        List urls.
  --download-url [=arg(=0)]          Download url content file.
  --settings-from-file arg           Load device settings stored in the given file (e.g. --settings-from-file
                                     Path/settingsFile).
  --settings-to-file arg             Store device settings in the given file (e.g. --settings-to-file
                                     Path/settingsFile).
  --select-user-set arg              Select user set which is loaded or stored to/from given file (e.g.
                                     --settings-to-file Path/settingsFile --select-user-set "Linescan").
  -n [ --node ] arg                  Select node by name (e.g. -n "DeviceModelName").
  --get                              Get node value and print it (e.g. -n "ExposureTime" --get).
  --set arg                          Set node value (e.g. -n "ExposureTime" --set "14000").
  --set-range-factor arg             Set numeric value by the range factor (0.0 => set minimum, 1.0 => set maximum e.g.
                                     -n "ExposureTime" --set-range-factor "0.0" sets the minimum exposure time).
  --inc                              Get node value increment  (e.g. -n "ExposureTime" --inc).
  --min                              Get node value maximum (e.g. -n "ExposureTime" --max).
  --max                              Get node value minimum (e.g. -n "ExposureTime" --min).
  --execute                          Execute node (e.g. -n "AcquisitionStart" --execute).

Register Commands:
  --register arg                     Select register address (e.g. --register "0x8002311c").
  --read                             Read register (e.g. --register "0x8002311c" --read).
  --write arg                        Write register (e.g. --register "0x8002311c" --write 32).
```

Fig. 5: sick_devicecommand

8.2.2 Configuring network settings

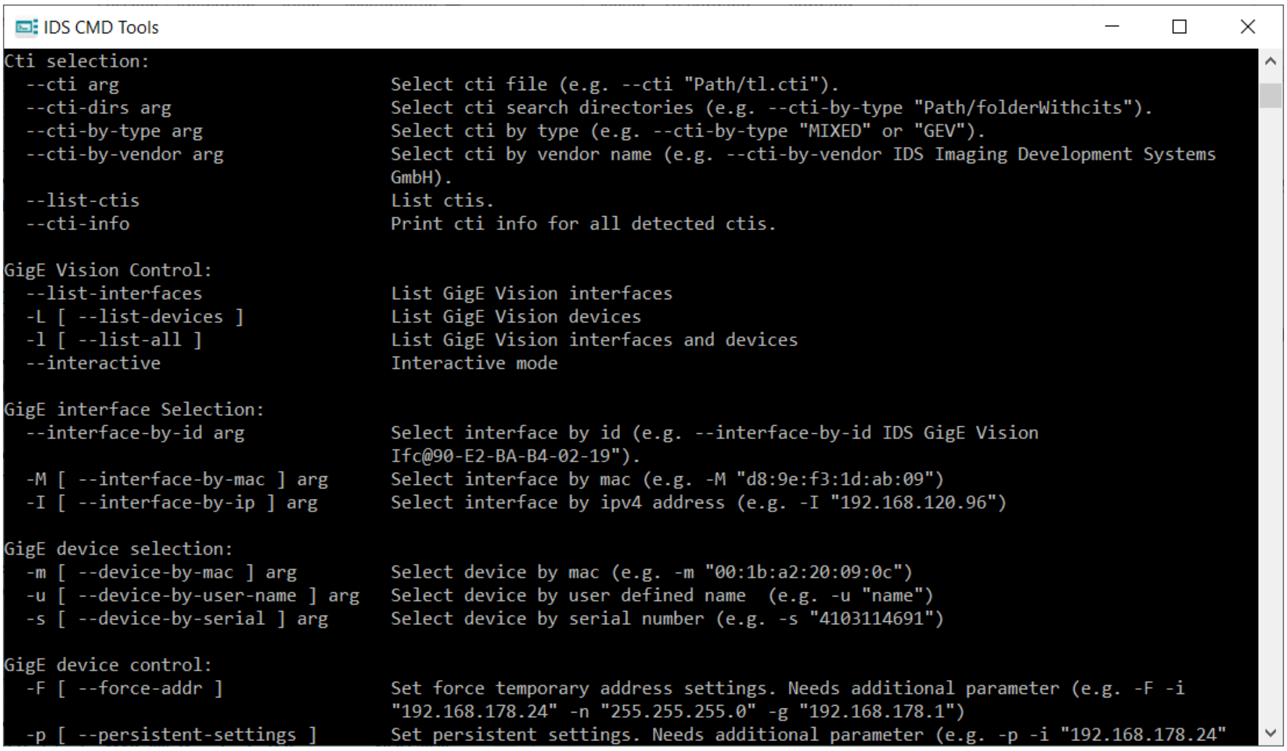
With **sick_ipconfig** you can query and configure the network settings of the cameras.

You can list further options with:

```
sick_ipconfig --help
```

With a special, interactive mode, you can configure the IP address easily with dialog guidance:

```
sick_ipconfig --interactive
```



```
Cti selection:
  --cti arg                           Select cti file (e.g. --cti "Path/tl.cti").
  --cti-dirs arg                      Select cti search directories (e.g. --cti-by-type "Path/folderWithcits").
  --cti-by-type arg                   Select cti by type (e.g. --cti-by-type "MIXED" or "GEV").
  --cti-by-vendor arg                 Select cti by vendor name (e.g. --cti-by-vendor IDS Imaging Development Systems
                                      GmbH).
  --list-ctis                         List ctis.
  --cti-info                          Print cti info for all detected ctis.

GigE Vision Control:
  --list-interfaces                   List GigE Vision interfaces
  -L [ --list-devices ]               List GigE Vision devices
  -l [ --list-all ]                   List GigE Vision interfaces and devices
  --interactive                       Interactive mode

GigE interface Selection:
  --interface-by-id arg               Select interface by id (e.g. --interface-by-id IDS GigE Vision
                                      Ifc@90-E2-BA-B4-02-19").
  -M [ --interface-by-mac ] arg       Select interface by mac (e.g. -M "d8:9e:f3:1d:ab:09")
  -I [ --interface-by-ip ] arg        Select interface by ipv4 address (e.g. -I "192.168.120.96")

GigE device selection:
  -m [ --device-by-mac ] arg          Select device by mac (e.g. -m "00:1b:a2:20:09:0c")
  -u [ --device-by-user-name ] arg    Select device by user defined name  (e.g. -u "name")
  -s [ --device-by-serial ] arg       Select device by serial number (e.g. -s "4103114691")

GigE device control:
  -F [ --force-addr ]                 Set force temporary address settings. Needs additional parameter (e.g. -F -i
                                      "192.168.178.24" -n "255.255.255.0" -g "192.168.178.1")
  -p [ --persistent-settings ]        Set persistent settings. Needs additional parameter (e.g. -p -i "192.168.178.24"
```

Fig. 6: sick_ipconfig

You can also configure the IP address using the graphical interface of SICK Vision Cockpit.

8.2.3 Updating the camera firmware

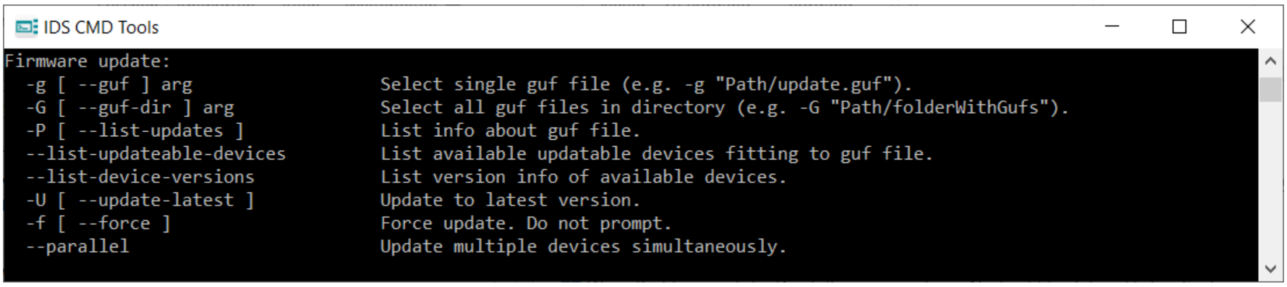
Use the firmware update tool **sick_deviceupdate** to transfer the firmware update to the camera using the GUF file. The command line tool allows you to choose the camera explicitly for the update.

The firmware update of Vision cameras can be started by double-clicking on the GUF file (GUF file version >= 1.5).

1. Make sure that GigE Vision cameras have a valid IP address.
2. Open the command line tool in the GUF directory.
3. List all available cameras with:
   `sick_deviceupdate --list-all`
4. You can also list all available cameras with their current firmware version:
   `sick_deviceupdate --list-device-versions`
5. Select a camera based on the last digits of its serial number and start the update, e.g.
   `sick_deviceupdate -s *7362 -U --guf c:\GUF\<model-name>-C.guf`
6. Confirm the update with "y".
   A firmware update can take several seconds. Wait until it is finished.

You can list further options with:

```
sick_deviceupdate --help
```



```
Firmware update:
  -g [ --guf ] arg              Select single guf file (e.g. -g "Path/update.guf").
  -G [ --guf-dir ] arg          Select all guf files in directory (e.g. -G "Path/folderWithGufs").
  -P [ --list-updates ]         List info about guf file.
  --list-updateable-devices     List available updatable devices fitting to guf file.
  --list-device-versions        List version info of available devices.
  -U [ --update-latest ]        Update to latest version.
  -f [ --force ]                Force update. Do not prompt.
  --parallel                    Update multiple devices simultaneously.
```

Fig. 7: sick_deviceupdate

8.3 Executable samples

C++

Sample	Description
ChunkLiveQmlCpp	Opens a camera and shows the use of chunk data via the SICK Vision API. The example uses QML (Qt Meta-object Language) for this.
ChunkLiveQtWidgetsCpp	Opens a camera and shows the use of chunk data via the SICK Vision API. The example uses QtWidgets for this.
DeviceTreeCpp	Builds a complete module tree.
GetFirstPixelCpp	Opens a camera and writes the first pixel for a defined number of images to the command line. • Test whether an open camera can capture images • Based on the pixel values you can perform a first validation (light/dark)
LegoTriggerCpp	Implements different trigger cases.
MultiCameraLiveQtWidgetsCpp	Opens several cameras and displays the live image. In addition, different information per camera is displayed, e.g. number of acquired images.
OpenCameraBySerNoCpp	Opens a camera with a specific serial number and reads some camera nodes.
OpenCameraCpp	Opens a camera and reads out some camera nodes. • Test whether a camera can be found and opened.
OpenCameraSelectCtiCpp	Opens a camera with a specific CTI and reads some camera nodes.
SaveImagesLiveQtWidgetsCpp	Opens a camera and displays the live image. The example uses QtWidgets for this. At the push of a button, a single image can be saved using SICK LibIMG.
SequencerLiveQtWidgetsCpp	This example requires a camera that supports the Sequencer feature. The example allows to parameterize 4 sequencer sets and to execute them in trigger mode. The following parameters can be used: • Exposure time • Gains (AnalogAll, DigitalAll, DigitalRed, DigitalGreen, DigitalBlue) • OffsetX and OffsetY
SimpleLiveQmlCpp	Opens a camera and displays the live image. The example uses QML (Qt Meta-object Language) for this.
SimpleLiveQtWidgets	Opens a camera and displays the live image. The example uses QtWidgets for this.

Sample	Description
WalkThroughCpp	Shows in detail how to handle cameras and features with the help of the SICK Vision API.

C#

Sample	Description
OpenCameraCSharp	Opens a camera and reads out some camera nodes. • Test whether a camera can be found and opened.
SimpleLiveWindowsFormsCSharp	Opens a camera and displays the live image. The example uses Windows Forms.
SimpleLiveWPFCSharp	Opens a camera and displays the live image. The example uses Windows Presentation Foundation (WPF).

- A -

Abbreviations 9

- C -

Camera
- configure 13
- parameter 13

Command line 12

- D -

DeviceTree 15

- F -

Firmware
- update 14

- G -

GenApi 6
GenICam 6
GenTL
- buffer 7
- consumer 7
- device 7
- GenTL Consumer 6
- GenTL Producer 6
- interface 7
- producer 7
- stream 7
- system 7

GenTL SFNC 6, 8
GetFirstPixel 15

- I -

IP address
- configure 13
- set 13

- L -

LegoTrigger 15

- N -

Network 13

- O -

OpenCamera 15
OpenCameraBySerNo 15
OpenCameraSelectCti 15

- S -

Sample 11, 15
SFNC 6
SICK LibIMG 11
SICK Vision API 11
SICK Vision Cockpit 11
- overview 12

SICK Vision Suite 5, 10
- application 12
- part 11

SimpleLiveQml 15
SimpleLiveQtWidgets 15

- T -

Tool
- command line 12

Transport Layer 11

- U -

Update
- firmware 14

- W -

WalkThrough 15

© SICK AG • Subject to change without notice • 8025622/0000

8025622/0000/2020-04/en

Australia
Phone +61 (3) 9457 0600
1800 33 48 02 – tollfree
E-Mail sales@sick.com.au

Austria
Phone +43 (0) 2236 62288-0
E-Mail office@sick.at

Belgium/Luxembourg
Phone +32 (0) 2 466 55 66
E-Mail info@sick.be

Brazil
Phone +55 11 3215-4900
E-Mail comercial@sick.com.br

Canada
Phone +1 905.771.1444
E-Mail cs.canada@sick.com

Czech Republic
Phone +420 234 719 500
E-Mail sick@sick.cz

Chile
Phone +56 (2) 2274 7430
E-Mail chile@sick.com

China
Phone +86 20 2882 3600
E-Mail info.china@sick.net.cn

Denmark
Phone +45 45 82 64 00
E-Mail sick@sick.dk

Finland
Phone +358-9-25 15 800
E-Mail sick@sick.fi

France
Phone +33 1 64 62 35 00
E-Mail info@sick.fr

Germany
Phone +49 (0) 2 11 53 010
E-Mail info@sick.de

Greece
Phone +30 210 6825100
E-Mail office@sick.com.gr

Hong Kong
Phone +852 2153 6300
E-Mail ghk@sick.com.hk

Hungary
Phone +36 1 371 2680
E-Mail ertekesites@sick.hu

India
Phone +91-22-6119 8900
E-Mail info@sick-india.com

Israel
Phone +972 97110 11
E-Mail info@sick-sensors.com

Italy
Phone +39 02 27 43 41
E-Mail info@sick.it

Japan
Phone +81 3 5309 2112
E-Mail support@sick.jp

Malaysia
Phone +603-8080 7425
E-Mail enquiry.my@sick.com

Mexico
Phone +52 (472) 748 9451
E-Mail mexico@sick.com

Netherlands
Phone +31 (0) 30 229 25 44
E-Mail info@sick.nl

New Zealand
Phone +64 9 415 0459
0800 222 278 – tollfree
E-Mail sales@sick.co.nz

Norway
Phone +47 67 81 50 00
E-Mail sick@sick.no

Poland
Phone +48 22 539 41 00
E-Mail info@sick.pl

Romania
Phone +40 356-17 11 20
E-Mail office@sick.ro

Russia
Phone +7 495 283 09 90
E-Mail info@sick.ru

Singapore
Phone +65 6744 3732
E-Mail sales.gsg@sick.com

Slovakia
Phone +421 482 901 201
E-Mail mail@sick-sk.sk

Slovenia
Phone +386 591 78849
E-Mail office@sick.si

South Africa
Phone +27 10 060 0550
E-Mail info@sickautomation.co.za

South Korea
Phone +82 2 786 6321/4
E-Mail infokorea@sick.com

Spain
Phone +34 93 480 31 00
E-Mail info@sick.es

Sweden
Phone +46 10 110 10 00
E-Mail info@sick.se

Switzerland
Phone +41 41 619 29 39
E-Mail contact@sick.ch

Taiwan
Phone +886-2-2375-6288
E-Mail sales@sick.com.tw

Thailand
Phone +66 2 645 0009
E-Mail marcom.th@sick.com

Turkey
Phone +90 (216) 528 50 00
E-Mail info@sick.com.tr

United Arab Emirates
Phone +971 (0) 4 88 65 878
E-Mail contact@sick.ae

United Kingdom
Phone +44 (0)17278 31121
E-Mail info@sick.co.uk

USA
Phone +1 800.325.7425
E-Mail info@sick.com

Vietnam
Phone +65 6744 3732
E-Mail sales.gsg@sick.com

Detailed addresses and further locations at **www.sick.com**

SICK AG | Waldkirch | Germany | www.sick.com

SICK
Sensor Intelligence.