

APPLICATION NOTE

Installing Vimba under Linux

Document V2.3.1
Feb 2023

Scope of this document

This document describes:

- Installing Vimba under Linux
- Changing the IP configuration if your GigE camera is in a foreign subnet
- Finding and running Vimba's code examples
- Compiling the C++ API

Prerequisites

To install Vimba, you need tar and the C runtime library glibc6 (PC: version 2.11 or higher, ARM: version 2.15 or higher).

Installing Vimba

Vimba comes as a tarball. To install it:

1. Uncompress the archive to a directory you have writing privileges for, such as /opt:

```
tar -xzf ./Vimba.tgz -C /opt
```

In this directory, Vimba will be installed in its own folder. In this document, we refer to this path as [InstallDir].
2. **GigE camera users:** Go to [InstallDir]/Vimba_x_x/VimbaGigETL.
USB camera users: Go to [InstallDir]/Vimba_x_x/VimbaUSBTL.
CSI-2 camera users: Go to [InstallDir]/Vimba_x_x/VimbaCSITL.
3. Execute the shell script Install.sh with root privileges (for example, `sudo ./Install.sh` or `su -c Install.sh`). If you use GigE and USB cameras, perform this step for both TLs (transport layers).

Troubleshooting

Ubuntu 20.04 LTS

Vimba uses qt4, which is no longer contained in Ubuntu 20.04 LTS.

```
# Install qt4 on Ubuntu 20.04 LTS
sudo add-apt-repository ppa:rock-core/qt4
sudo apt-get update
sudo apt-get install libqtcore4
sudo apt-get install libqt4-network --fix-missing
sudo apt-get install libqt4-qt3support
```

Transport layers not found

In most cases, Install.sh automatically registers the GENICAM_GENTL32_PATH and GENICAM_GENTL64_PATH environment variables in /etc/profile.d, so that every GenICam GenTL consumer can access the Vimba transport layers.

If the transport layers are not found:

If multiple users work with the system, make sure all users can access /etc/profile.d

If your display manager doesn't support the install script (for example, lightdm and wdm):

Please add the required environment variables to the /etc/environment file.

If login shell support is not supported, Install.sh in /etc/profile.d will not be loaded for X-Session. In this case, please copy the following line into the ~/.bashrc file and reboot.

```
# 32-bit
export GENICAM_GENTL32_PATH=$GENICAM_GENTL32_PATH:"/PATH_TO_VIMBAFOLDER/VimbaGigETL/CTI/x86_32bit/"
# 64-bit
export GENICAM_GENTL64_PATH=$GENICAM_GENTL64_PATH:"/PATH_TO_VIMBAFOLDER/VimbaGigETL/CTI/x86_64bit/"
```

To apply the changes, log off and log in again.

Vimba Viewer

With Vimba Viewer, you can control Allied Vision cameras and capture images.

Vimba Viewer can be found in, for example, [InstallDir]/Vimba_x_x/Tools/Viewer/Bin/x86_64bit/.

Prerequisites

To build Vimba viewer, you need (if not already installed):

- pkg-config:
`sudo apt-get install pkg-config`
- libqt4-dev:
`sudo apt-get install libqt4-dev`
- Ubuntu 18.04 LTS requires the libcanberra-gtk-module:
`sudo apt-get install libcanberra-gtk-module`

- In some cases, libpng12 is missing. Please find detailed instructions in the document:
<https://cdn.alliedvision.com/fileadmin/content/documents/products/software/software/Vimba/apnote/libpng-installation.pdf>

Changing the IP configuration in a foreign subnet

To change the IP configuration of a GigE camera in a foreign subnet, run Vimba Viewer with root privileges (for example, `sudo -E [InstallDir]/Vimba_2_1/Tools/Viewer/Bin/x86_32bit/VimbaViewer`). Note that running it as root instead of using `sudo -E` requires that `GENICAM_GENTL32_PATH` and/or `GENICAM_GENTL64_PATH` are set for the root as well.

Compiling the code examples and the C++ API

Vimba includes many code examples that can be found in, e.g.,

`[InstallDir]/Vimba_x_x/VimbaC/Examples/Bin/x86_32bit` and

`[InstallDir]/Vimba_x_x/VimbaCPP/Examples/Bin/x86_32bit`.

Vimba for ARM comes with compiled code examples. To compile the precompiled code examples on a PC or to compile the open source Vimba C++ API, you additionally need the packages listed below. Very likely, most of them are already part of your system:

- `make`
- `ffmpeg`
- `g++` (PC: Version 4.4.5 or higher / ARM: Version 4.7.3 or higher)
- `Qt` (PC: Version 4.8.4 / ARM: 4.8.5)
- `TinyXML` (Version 2.5.3 or higher)

Vimba provides all necessary runtime libraries for executing the examples including the Vimba Viewer example.

Exception

The Vimba C++ code example `AsynchronousOpenCVRecorder` requires OpenCV 3.0. The example includes a script for compiling and installing OpenCV on Debian-based distributions.



Download OpenCV

<http://opencv.org/>

Compiling the examples

To compile the examples (not required on ARM systems), go to `Build/Make` in the `VimbaC` and `VimbaCPP` example folders and type **`make`** in your shell.

Disclaimer

For the latest version of this document, please visit our website. All trademarks are acknowledged as property of their respective owners.

Copyright © 2023 Allied Vision Technologies.