

Vimba for ARM64 Release Notes

Version: 6.1

Summary

Vimba 6.1 is a release version for ARM64 only. This release contains mainly changes of the CSI TL and support of JetPack 4.6.1.

Components and version reference

Transport Layer	Version
CSI Transport Layer (VimbaCSITL.cti)	1.1.0*
GigE Transport Layer (VimbaGigETL.cti)	1.9.0
USB Transport Layer (VimbaUSBTL.cti)	1.4.1

APIs	Version
Vimba C API (libVimbaC.so)	1.9.2*
Vimba C++ API (libVimbaCPP.so)	1.9.0
Vimba Python API	1.3.0*
Image Transform Library (libVimbalImageTransform.so)	1.6.0

Tools	Version
Vimba Viewer (VimbaViewer)	2.5.1*
Vimba Class Generator (VimbaClassGenerator)	1.0.6
Vimba Firmware Updater (VimbaFirmwareUpdater.exe)	1.2.1*
Vimba Firmware Updater Console (FWUpdaterConsole.exe)	1.2.1*

Third Party Libraries	Version
TinyXML (not shipped with Vimba)	2.6.2
Qt	4.8.5

Third Party Libraries	Version
Qwt	6.1.3
libTIFF	4.0.7
OpenCV	3.0.0

* Changed in this release of Vimba

Hardware requirements

- ARM board with ARMv8-compatible 64-bit processor
- Allied Vision GigE cameras require a built-in Gigabit Ethernet network interface or at least one Gigabit Ethernet network card.

Note for IP configuration: By default, IP Configuration Mode is set to DHCP. You can also select LLA (link-local address) if you connect just one camera. If multiple cameras are connected at the same time, we recommend using DHCP or Persistent IP Address to enable discovery of all cameras.

- Allied Vision USB cameras require a built-in USB 3.0 controller for PCI Express bus.

Tested platforms and cameras

GigE and USB cameras

Vimba was tested on NVIDIA Xavier AGX with JetPack 4.6.1 (L4T 32.7.1).

Alvium MIPI cameras

The following NVIDIA Jetson developer kit platforms with JetPack 4.6.1 (L4T 32.7.1) support the Alvium MIPI driver and the Vimba CSI TL:

- AGX Xavier
- Xavier NX
- TX2 NX
- Nano 4GB
- Nano 2GB

For supported camera models, the latest known issues, and other useful information, see: [Getting Started with GenICam for CSI-2](#)

Installation

To install Vimba, see: [Vimba installation under Linux](#)

Short instruction:

- Uncompress the archive with the command `tar -xf ./Vimba.tgz` to a directory you have writing privileges for like `/opt`. Under this directory Vimba will be installed in its own folder. In the following, we will refer to this path as `[InstallDir]`.
- Go to `[InstallDir]/VimbaGigETL` and/or `[InstallDir]/VimbaUSBTL` and execute the shell script `Install.sh` with super user privileges (e.g., `sudo ./Install.sh`). This registers the `GENICAM_GENTL32_PATH` and/or the `GENICAM_GENTL64_PATH` environment variable through a startup script in `/etc/profile.d` so that every GenICam GenTL consumer (such as the examples that ship with Allied Vision Vimba) can access the Allied Vision Gigabit Ethernet and USB Transport Layers. Please note that this is a per-user setting. If transport layers are not found, follow the instructions in this document: [Vimba installation under Linux](#)
- Reboot computer

Vimba Viewer: Vimba Viewer can be found in, for example, `[InstallDir]/Tools/Viewer/Bin/x86_64bit/`. This program allows you to configure your Allied Vision cameras and capture images.

Ubuntu 18.04 LTS:

- To build Vimba Viewer, install the `libcanberra-gtk-module`: `apt-get install libcanberra-gtk-module`

Vimba includes many precompiled examples that can be found in `Vimba/VimbaC/Examples/Bin` and `Vimba/VimbaCPP/Examples/Bin`. If you want to compile these examples yourself, please refer to the user manual for instructions and required packages.

For uninstalling Vimba, remove the startup scripts by running the shell scripts `Uninstall.sh` as super user. This prevents any GenTL consumer from loading the Vimba and USB Transport Layers. Then simply remove the installation directory.

Known issues

- When experiencing trouble with image acquisition, try to increase the priority of your application with `“sudo -E nice -n -20 ”`.

GigE Transport Layer:

- A change of the available camera access modes cannot be reported for cameras in different subnets or with malconfigured IP addresses. These cameras can be accessed in configuration mode from different hosts simultaneously.
- Not all ARM boards come with a GigE network interface. When acquiring images with only 100 Mbit/s, make sure to set the “StreamBytesPerSecond” feature to a reasonable small value (12 400 000). This adjustment is done automatically when the camera is connected directly to a 100 Mbit/s interface.

Transport layers not found (Error -16):

- Please follow the instructions in [Vimba Installation under Linux](#):
- Vimba Firmware Updater and FWUpdaterConsole.exe: Mako-U cameras are not detected after updating the firmware. To detect the camera, please reboot the board.

Camera detection:

- If multiple IP addresses are configured on one physical Ethernet adapter, then Vimba sends GVCP discovery requests only to the last added IP address. Therefore, the camera is detected only if its IP address was added last.

Vimba Viewer:

- If a GigE camera was opened and then closed with Vimba Viewer, it is displayed as locked and Vimba Viewer does not open it again. Workaround: Close and open Vimba Viewer.
- When multiple opened GigE cameras are disconnected at the same time while no other camera stays connected to the very same NIC, the Vimba Viewer’s camera windows stay open.
- qt error: please [install libpng12](#) if it is missing.
- After streaming several hours, the displayed image is not updated. Changing camera settings results in a black image. Stopping and starting image acquisition updates the displayed image. The issue affects only the displayed image, not the image itself.

Vimba code examples

- qt error: please [install libpng12](#) if it is missing.

Vimba APIs:

- When API startup and shutdown is performed excessively within the same process, camera initialization may fail sporadically, which may cause exceptions. We recommend to have only one API session during one process.

CSI-2:

- For the latest known issues, see [Getting Started with GenICam for CSI-2](#)
- If your application causes dropped frames, try to increase the number of frame buffers to approximately 7.
- Switching the camera from GenICam for CSI-2 to V4L2 or vice versa requires rebooting the board.
- Exposure times greater than approx. 1 second: Stopping acquisition may cause an error. Please close and open the camera to start streaming again.

Changes and release history

Changes in Vimba 6.1.0

VimbaPython

- Fixed: Missing frames from some cameras, especially with short exposure times.

VimbaC

- Fixed: IP configuration

CSITL (supported camera models and ARM boards and known issues: see above)

- Supports new streaming format (better performance in Announce mode), latest camera firmware required

Vimba Viewer

- Improved performance with MIPI cameras

Firmware Updater

- Main window now displays more information about an occurred error

Firmware Updater Console

- Fixed: ShowFailureMessage in case exception is raised

Changes in Vimba 6.0.0

VimbaC

- Fixed: Issue with null terminator for zipped XML files

CSITL

- First release (supported camera models and ARM boards and known issues: see above)

USBTL

- Bug fix (relevant for Windows only)

VimbaPython:

- Less restrictive C API version check

Changes in Vimba 5.1.0

VimbaC

- New feature: Support for Alloc And Announce frame allocation mode
- Feature endianness related fixes (corresponding to GigE TL)

VimbaCPP

- New feature: Supports Alloc And Announce frame allocation mode

VimbaPython

- New feature: Supports Alloc And Announce frame allocation mode

Vimba Viewer

- Supports Alloc And Announce frame allocation mode
- Chinese localization
- Supports changes of GigE TL and VimbaC
- Fixed: IP labels not updated after changing the IP via `GevCameraForceAddressSend`
- Fixed: Auto ROI tab always resets ROI at startup

Firmware Updater

- Supports Git hash representation

Firmware Updater Console

- Supports Git hash representation

GigETL

- Changed behaviour of endianness handling (values must match host endianness in little-endian), for better compatibility with third-party software. Affected features that now have a changed byte order:
 - Config Mode
 - IP-related features including MulticastIPAddress
 - Action Commands
- New feature DeviceUpdateTimeout (only applicable if GigE discovery is switched to Broadcast)
- New GVSPMaxLookBack default value (0 instead of 1) for improved packet resend handling
- Fixed: High memory usage during performance bottleneck (Nonpaged pool growing with multiple GigE cameras)
- Fixed: Action commands caused an error message in the console log
- Fixed: Multicast IP address range, better compatibility with third-party software

USB TL

- Updated 3rd-party libraries

Changes in Vimba 5.0.0

GigETL

- New feature: GVSPHostReceiveBufferSize (SO_RCVBUF) replaces GVSPHostReceiveBuffers (which is still usable in existing applications).
- Fixed for 5 GigE cameras:
 - Fixed limitations of certain GVSPPacketSize values

VimbaC

- Fixed: Callbacks after stopping image acquisition caused exceptions in the Python API.
- Fixed: ListAncillaryDataFeatures example (error “Could not queue frame”)

Vimba Firmware Updater

- Prepared for CSI-2 support

Vimba Firmware Updater Console

- Prepared for CSI-2 support

Vimba Viewer

- Prepared for CSI-2 support

Changes in Vimba 4.3.0

GigETL

- Prepared for use with 5 GigE Vision cameras

VimbaC

- Prepared for use with 5 GigE Vision cameras

VimbaCPP

- Preparations for use with 5 GigE Vision cameras in the underlying VimbaC API

Vimba Python

- Updated for use with new VimbaC version
- Fixed: read_memory now returns all 0_bytes

Vimba Firmware Updater

- Preparations for use with 5 GigE Vision cameras

Vimba Firmware Updater Console

- Preparations for use with 5 GigE Vision cameras

Vimba Viewer

- Preparations for use with 5 GigE Vision cameras

Changes in Vimba 4.2.0

Vimba C

- Standard-compliant ForcelP features instead of Allied Vision custom features, Updated VimbaC ForcelP example to use the standard-compliant ForcelP features

Vimba Python

- Redirected the used Vimba version to the one delivered in this package.
- Install.sh suggests compatible Python interpreter versions only
- Install.sh and Uninstall.sh: Added support of virtual environments, Added debug flag

GigETL

- Standard-compliant ForcelP features instead of Allied Vision custom features (see also Vimba C)
- Prepared for use with extended IDs

USBTL

- Increased default MaxTransferSize value for better performance with current Linux versions
- Improved camera detection on Linux systems (plugin event handling)

Vimba Viewer

- Added a dialog for enforcing IP settings to a GigE Vision camera

Changes in Vimba 4.1.0

USBTL

- Solved conflicts between customers' boost version and the internally used version
- Other internal bug fixes

Changes in Vimba 4.0.0

Vimba C

- Bug fixes

Vimba C++

- Bug fixes

Vimba Python

- New API

GigETL

- Improved resend handling
- By default, packet resends are enabled
- Other bug fixes

USBTL

- Improved visibility of USB camera status
- Other bug fixes

Vimba GigE Filter Driver

- Bug fixes

Vimba Viewer

- Bug fixes

Changes in Vimba 3.1.0

Vimba C

- Updated for use with GenTL 1.5

Vimba C++

- Updated for use with GenTL 1.5

GigETL

- GenTL 1.5 support

USBTL

- GenTL 1.5 support

Vimba Viewer

- Bug fixes

Changes in Vimba 3.0.0

Vimba C

- Bug fixes

Vimba C++

- Bug fixes

Vimba Viewer

- Supports Alvium USB cameras
- Fixed handling of float increments
- Fixed occasional crashes with high frame rates
- Several minor bug fixes

GigETL

- Fixed: GVSPPacketSize now updates as expected

USBTL

- Bug fixes

Changes in Vimba 2.1.3

Vimba C

- Added ActionCommands programming example
- Vimba C Manual: Updated documentation of Action Commands
- Prepared the API for use with GenTL 1.5
- Bug fixes

Vimba C++

- Added ActionCommands programming example
- Vimba CPP Manual: Updated documentation of Action Commands
- Prepared the underlying C API for use with GenTL 1.5
- Vimba CPP Manual: Added chapter Asynchronous image acquisition - overview
- Bug fixes

Vimba Viewer

- New plugin architecture

Vimba Setup

- Removed Vimba Class Generator desktop symbol

Vimba Manual

- Integrated Vimba Features Manual

Changes in Vimba 2.1.0

Vimba C

- Added functionality for Action Commands.
- Bug fixes

Vimba C++

- Added functionality for Action Commands.
- Bug fixes

Vimba Image Transform Library

- Support for conversion from 10-bit raw source formats to 16-bit color target formats.
- Support for conversion from 12-bit raw source formats to 16-bit color target formats.
- Support for conversion from 14-bit raw source formats to 16-bit color target formats.
- Support for conversion from 10-bit mono source formats to 16-bit mono target formats.
- Support for conversion from 12-bit mono source formats to 16-bit mono target formats.
- Support for conversion from 14-bit mono source formats to 16-bit mono target formats.

Vimba Viewer

- Allows now saving of 12-bit and 16-bit images (12-bit images are saved in 16-bit containers).
- Added functionality for Action Commands.
- Camera list shows the IP address of each device.
- Bug fixes

GigE Transport Layer

- The parameter BUFFER_INFO_DELIVERED_IMAGEHEIGHT is filled correctly.
- Added functionality of Action Commands.
- Bug fixes

Changes in Vimba 2.0.0

Vimba Setup

- Added Vimba Viewer Guide document
- Improved documents, new layout

Vimba C

- Added functionality of loading and saving camera settings from / to XML
- Bug fixes

Vimba C++

- Added functionality of loading and saving camera settings from / to XML
- Bug fixes

Vimba Image Transform Library

- Support for conversion from BayerXX10,12 formats to 2 byte output formats (RGB[A], BGR[A])
- Added interface functions VmbSetImageInfoFromInputParameters and VmbSetImageInfoFromInputImage

Vimba Viewer

- Enabled making use of load / save settings functionality
- Added tabsheet widgets for Brightness, ROI, Autofunction ROI, Trigger IO, Color
- Bug fixes

USB Transport Layer

- Minor bug fixes

GigE Transport Layer

- Reworked and improved filter driver

Vimba Class Generator

- Changes for Camera Link support (affects Windows only)

Changes in Vimba 1.4.1

Package for ARMv7

- Support of USB Transport Layer

Package for other Linux systems

- Support of USB Transport Layer
- Improved documentation

Vimba C

- Extended camera support (e.g. float increments)
- Stability improvements
- Improved some examples
- Bug fixes

Vimba C++

- Extended camera support (e.g. float increments, needs Vimba C 1.4)
- Improved error handling
- Improved some examples
- Added and improved examples
- Bug fixes

Vimba Image Transform Library

- Added support for PFNC 10p and 12p formats
- Improved the documentation

GigE Transport Layer

- Improved stability on high traffic networks
- Bug fixes for multi-camera usage and incomplete cleanup
- Fixed range of `GevHeartbeatInterval` and `GevHeartbeatTimeout`

Vimba Viewer

- Added open camera by ID
- Added endianness switch in direct register dialog
- Many bug fixes

Vimba Class Generator

- Small improvements

Changes in Vimba 1.3.0

Package for ARMv7 (new)

- Support of Vimba C API
- Support of Vimba C++ API
- Support of GigE Transport Layer

Package for other Linux systems

- Added a Vimba Tour document and a Vimba Quickstart Guide
- Improved the other documents
- Added examples for Vimba C and C++
- Added an OpenMP version of the Vimba Image Transform Library

Vimba C 1.3.0

- Fixed feature access for ancillary data
- Made device discovery blocking
- Added system feature GeVDiscoveryAllDuration for adjustable waiting time
- Fixed issues with mixed feature names
- Bug-fixes for improving stability

Vimba C++ 1.3.0

- Fixed feature access for ancillary data
- Changes caused by bug fixes in Vimba C

Vimba Image Transform Library 1.2.0

- Allowed more input for VmbSetImageInfoFromString
- Corrected return codes
- Removed transformations to Mono10/12/14 and big-endian data

Vimba Viewer 1.1.3

- Improved handling for GigE camera list changes
- Fixed issues around saving images
- GUI usage improvements

GigE Transport Layer 1.3.1

- Fixed GenTL compliance: Changed the Device ID from the serial number to a MAC-based ID. This changes the way all APIs (C, C++ and .Net) identify the cameras
- Added a document for GigE camera features
- Changed the supported GenTL version to 1.3 (e.g. new feature categories BufferHandlingMode and StreamInformation)
- Separate features for heartbeat timeout and heartbeat interval
- Adjustable device discovery waiting time
- Fixed issues with similar serial numbers
- Fixed issues with many network interfaces

Vimba Class Generator

- Minor changes to appearance and documentation

Changes in Vimba 1.2.0

Vimba C 1.2.0

- Version 1.2.0 is only compatible to AVT GigE TL Version 1.2.0 onwards
- Name independent search for AVT transport layers
- Made EnumEntries' "plsImplemented" condition equivalent to "plsAvailable" to mimic the behavior of GenICam reference implementation

Vimba C++ 1.2.0

- Fix for opening camera by IP address

Vimba .NET 1.2.0

- Fix for 64bit version attempting to load 32bit transport layers
- Fix for opening camera by IP address

Vimba Viewer 1.1.1

- SFNC Features directly beneath the root category get listed
- Many new pixel formats available for histogram support

GigE Transport Layer 1.2.0

- Improved performance in case of many events
- Restriction to eight cameras removed
- Corrected Heartbeat feature
- Fixed support for small packets