Allied Vision

Technical Note

Using Alvium Cameras with Third-Party Software

Document V1.0.0
March 2023

Scope of this document

This application note is intended for Alvium camera users who want to set up their first vision application with a GenICam-compliant third-party software. The following topics are included:

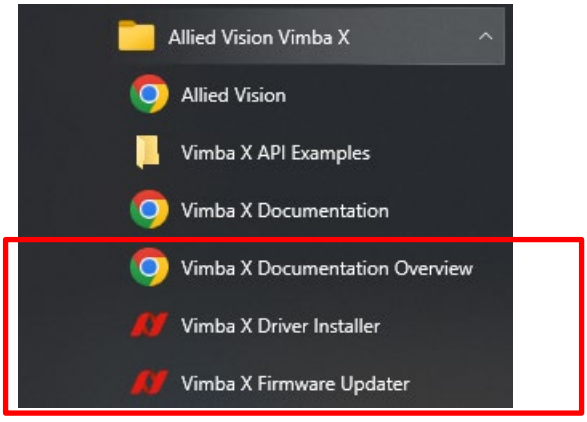
- Installing the required Allied Vision components
- Checking if your camera runs as expected
- Using the third-party software with the Vimba X transport layer
- HALCON on Linux and Linux ARM

Installing the required Allied Vision components

Windows

1. Download our Vimba X SDK – it contains the latest transport layers and supports the latest Alvium features (Vimba users: you don't have to deinstall Vimba, but read this document carefully to avoid complications).
2. Install Vimba X (connect your Alvium USB camera to install the driver automatically).
3. GigE (and USB if not connected during installation): Open *Vimba X Driver Installer* to use the GigE Filter driver or the USB camera driver.
4. Make sure your camera has the latest firmware. If not, update the firmware with *Vimba X Firmware Updater*.

If you need instructions, open *Vimba X Documenation Overview* and go to the *Developer Guide*.



If you have installed Vimba X and Vimba:

- GigE: If you have installed Vimba X and Vimba, read the information in the *Release Notes* regarding the GigE Filter Driver.
- All interfaces: The latest transport layers from Vimba X have the display name *AVT <interface> Transport Layer*. The outdated transport layers from Vimba have the display name *Vimba <interface> Transport Layer*. The file name hasn't changed. If your third-party software doesn't show the display name, remove the Vimba transport layers from Vimba's cti directory to make sure the latest transport layers from Vimba X are used.

Recommendation for best performance

GigE cameras: Follow the instructions in the camera's User Guide, especially chapter *Configuring the host computer.*
USB cameras: If you use multiple USB cameras or hubs or if the performance is lower than expected, read: Considerations for Setting Up USB Vision Systems

Linux and Linux ARM

MIPI CSI-2 cameras only:

As the first step, install the camera driver. You can find compatible boards, drivers, and instructions on GitHub.

All camera interfaces:

1. Download our Vimba X SDK – it contains the latest transport layers and supports the latest Alvium features (Vimba users: you don't have to deinstall Vimba).
2. Install Vimba X and the required transport layers according to the instructions in the Release Notes.
3. Make sure your camera has the latest firmware. If not, update the firmware with *Vimba X Firmware Updater* available in the Vimba X install directory /bin.

If you have installed Vimba X and Vimba:

- The latest transport layers from Vimba X have the display name *AVT <interface> Transport Layer*. The outdated transport layers from Vimba have the display name *Vimba <interface> Transport Layer*. If your third-party software doesn't show the display name, remove the Vimba transport layers from Vimba's cti directory to make sure the latest transport layers from Vimba X are used.

Recommendation for best performance

GigE cameras: Follow the instructions in the User Guide, especially chapter *Configuring the host computer.*
USB cameras: If you use multiple USB cameras or if the performance is lower than expected, read: Considerations for Setting Up USB Vision Systems

Checking if your camera runs as expected

To check if your camera runs as expected, use Vimba X Viewer to acquire images and set features. You can also use Vimba X Viewer to configure the IP address of your GigE camera. The viewer is available from the Windows Start menu or in the Vimba X install directory /bin.

If you experience issues, read chapter *Troubleshooting* in the *Developer Guide* available in the Vimba X install directory or on docs.alliedvision.com

Using the third-party software

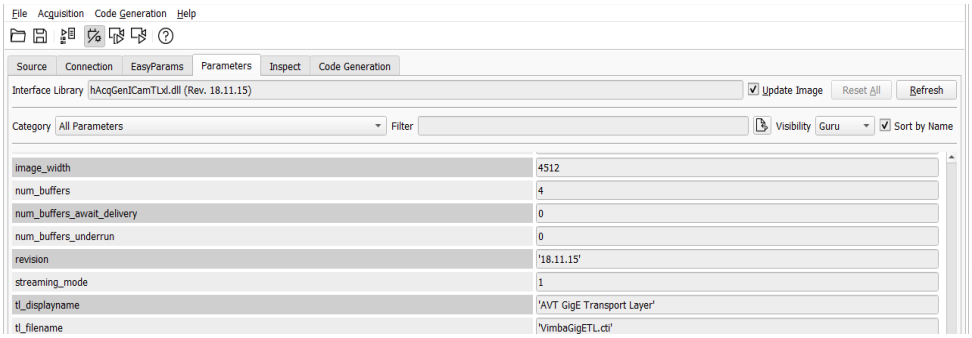
Preparations:

- Before starting the third-party software, make sure no other application uses the camera.
- MATLAB: *Vision Acquisition Toolbox* and *GenICam support package* must be installed.
- Note that camera recognition may take a while, especially with GigE cameras.

Using the camera:

- Windows only: We recommend using your Alvium GigE camera with the Vimba X Filter Driver.

- Follow the instructions of the third-party vendor to use your Alvium camera with the AVT transport layer.
    - CVB: Read Stemmer's instructions
    - HALCON:
        - Follow the instructions in HALCON's extensive documentation.
        - Select **Assistants** -> New Image Acquisition.
        - Select the **Source** tab -> Image Acquisition Interface -> GenICamTL.
        - Select **Connection** and select the desired device. Select the **Connect** tab.
        - Check if the AVT Transport Layer is used:



HALCON on Linux and Linux ARM

Here are some tips for using HALCON on Linux and Linux ARM:

- USB dongle and license:
    - The USB dongle needs a package to enable "raw" access to USB devices, which is called "libhidapi-hidraw0" under Ubuntu. The package can be installed with the command:

      ```
      sudo apt-get install libhidapi-hidraw0
      ```

    - Additionally, the USB dongle needs a udev rule so that any user can access it. You can create the file (as root or via sudo) /etc/udev/rules.d/99-hidraw-permissions.rules with the following line:

      ```
      KERNEL=="hidraw*", SUBSYSTEM=="hidraw", MODE="0664", GROUP="plugdev"
      ```

    - To activate the udev rule to for a user, the user has to be a member of the group "plugdev". With Ubuntu desktop, this is default. With some ARM boards or SOMs, you must add it explicitly with the command:

      ```
      sudo usermod -a -G plugdev MYUSERNAME
      ```

    - License file: Always check that the license file corresponds to the license number that is printed on the dongle itself.
    - The command line tool *hhostid* is handy for checking whether the dongle was recognized. It should print the license name (not just some number).
- Installation and development:
    - The HALCON GUI "hdevelop" is only available for Linux Desktop and not for Linux ARM64.
    - A Linux package for HALCON (at least the "full package") is basicallly a standalone directory packed as a tar.gz that doesn't need to be installed with the provided install script. It is sufficient to just unpack the tar.gz file and the binaries, e.g., for hdevelop, which are located

in the /bin subdirectory. The following shell script can be used to appropriately set the environment variables for HALCON to find its “install” directory (change the HALCONROOT variable in the shell script):

```
setup-env.sh # load with: source setup-env.sh
```

- On Linux desktop, use the command line program *hrun* that simply executes hdevelop scripts (pass the file name to *hrun*). On Linux for ARM64, this binary has to be compiled using *make*. It is located inside the directory: examples/arm-linux.
- Debugging HALCON and GenICam-compliant transport layers on ARM: Set the following shell environment variables to receive errors on stderr:

  HALCON_GENTL_LOGGING=1

  HALCON_LOG_LEVEL=99