APPLICATION NOTE

USB Camera Handling - Linux

Scope of this document

This document helps you optimize the performance of USB3 Vision cameras on Linux systems.

Additional information

More detailed information and USB troubleshooting is available in the technical manual or user guide of your Allied Vision USB camera:
https://www.alliedvision.com/en/support/technical-documentation.html

Optimizing camera performance

For optimal camera performance, we strongly recommend using the latest Vimba version.

For smooth data transfer of USB3 Vision cameras, the host computer must be equipped with a high-performance USB controller.

Suitable USB 3.0 accessories

For suitable USB 3.0 host controller cards, hubs, and cables, visit:
https://www.alliedvision.com/en/products/accessories

Adjusting DeviceLinkThrougputLimit

During freerun, Alvium cameras do not automatically adapt the frame rate to the USB controller's limits. If the data rate is too high for your USB controller, it receives corrupted frames. The image transfer status in Vimba Viewer is signaled as Running. However, the corrupted frames are not displayed. To avoid corrupted frames, adjust the `DeviceLinkThroughputLimit` value. Note that Vimba Viewer does not gray out values that the USB controller cannot receive.

Calculating DeviceLinkThroughputLimit

Numbers in this section

For best readability, numbers are shown as 1,000 instead of 1000. In program code and in Vimba Viewer, values must be entered as 1000.

Values required for `DeviceLinkThroughputLimit` (D) for different fame rates:

RGB8: D = H × V × 3 × frame rate [Byte/s]
Mono8: D = H × V × 1 × frame rate [Byte/s]

Example calculation for RGB8:
D = 1944 pixels × 2592 pixels × 3 × 28 fps = 423,263,232 Byte/s

Adding 5% (estimation) for the overhead of the USB3 Vision protocol:
423,263,232 Byte/s × 1.05 = 444,426,393.6 Byte/s

You can find examples of performance on reference systems in the Alvium USB camera User Guide:
https://www.alliedvision.com/en/support/technical-documentation/alvium/documentation.html

Increasing the USBFS buffer size

By default, the USBFS buffer size is 16 MB. This value is too low for image sizes > 2 MB or high frame rates.

Check the USBFS buffer size:
`cat /sys/module/usbcore/parameters/usbfs_memory_mb`

Increase the USBFS buffer size until the next reboot (here: example value 1000):
`sudo sh -c 'echo 1000 > /sys/module/usbcore/parameters/usbfs_memory_mb'`

Increase the USBFS buffer size permanently:
To increase the buffer size permanently, add the kernel parameter `usbcore.usbfs_memory_mb=1000` the bootloader configuration. How to do this depends on the bootloader on your system.

*** Before changing the bootloader configuration, create a backup of your system.***
A faulty bootloader may cause that your system doesn't start up. A fix might require an external boot media or reinstallation of the system. Note that changing the bootloader configuration is at your own risk.

GRUB 2:
1. Open /etc/default/grub. Replace: GRUB_CMDLINE_LINUX_DEFAULT="quiet splash" (or other contents within the quotation marks depending on your system) with:
   GRUB_CMDLINE_LINUX_DEFAULT="quiet splash usbcore.usbfs_memory_mb=1000"
2. Update grub: `sudo update-grub` and reboot the system.

Syslinux:
Open /boot/extlinux/extlinux.conf, add `usbcore.usbfs_memory_mb=1000` to the APPEND line, and reboot the system.

Other bootloaders:
To configure additional kernel parameters of other bootloaders, please see the manual of your bootloader.

Adapting the Vimba USB Transport Layer settings

Depending on your use case, changing the default value of MaxTransferSize of the Vimba USBTL (USB Transport Layer) may be beneficial.

To adjust the value of the VimbaUSBTL.xml file:

1. In the Vimba program folder, open VimbaUSBTL.

2. Depending on your system, the XML file is located in, for example, Bin/x86_64bit/VimbaUSBTL.xml.

3. Open the XML file and find MaxTransferSize.

4. Per default, the value is commented out. Delete the XML comments to activate the value.

5. Replace the default value (262144) with the value of your choice.