

pylon 4 Camera Software Suite for Linux for Use with Basler
Gigabit Ethernet (GigE) and Basler USB 3.0 Cameras (U3V)

=====

System Requirements

=====

GigE

A GigE network adapter that supports jumbo frames is recommended. For best performance, we recommend using an adapter from the Intel PRO 1000 series. Although the pylon software will work with any GigE network adapter, we observed a significantly higher CPU load when grabbing image data with network adapters other than from the recommended Intel PRO 1000 series.

USB

For U3V devices a USB3-capable USB controller is necessary. For best performance and stability we highly recommend a kernel version $\geq 3.13.x$

Installation

=====

The installation of pylon for Linux is described in the INSTALL text document.

Performance Optimization

=====

To increase performance and to minimize CPU usage when grabbing images, the following settings should be considered:

GigE Devices

- * Enable Jumbo Frames.
Many GigE network adapters support so-called jumbo frames, i.e., network packets larger than the usual 1500 bytes. To enable jumbo frames, the maximum transfer unit (MTU) size of the PC's network adapter must be set to a high value (see the description in the INSTALL document). We recommend using a value of 8192.
- * Increase the packet size.
When jumbo frames are enabled, the camera's packet size must be increased to benefit from the larger packets. The 'Optimizing Packet Size' section of the INSTALL document describes how to set the packet size.
- * Real-time Priority
The GigE Vision implementation of Basler pylon software uses a thread for receiving image data. Basler pylon tries to set the thread priority for the receive thread to real-time thread priority. This requires certain permissions. The 'Permissions for Real-time Thread Priorities' section of the INSTALL document describes how to grant the required permissions.

U3V Devices

- * Increasing Packet Size
For faster USB transfers you should increase the packet size. You can do this by changing the "Stream Parameters" -> "Maximum Transfer Size" value from inside the pylon Viewer or by setting the corresponding value via the API. After increasing the package size you will likely run out of kernel space and see corresponding error messages on the console. The default value set by the kernel is 16 MB. To set the value (in this example to 1000 MB) you can execute as root:
 echo 1000 > /sys/module/usbcore/parameters/usbfs_memory_mb
This would assign a maximum of 1000 MB to the USB stack.
- * Zero-copy

The current linux kernel (3.14) requires one copy operation to transfer USB data from the device to user-memory. On slower devices this leads to significant cpu overhead. To overcome this issue Basler prepared a kernel patch, which allows direct transfer of USB data to user memory (ZeroCopy). This patch is currently in experimental state and we will provide patched kernel packages for selected distributions. Ask our support team (support.europe@baslerweb.com) for binaries and/or the patch if you would like to test zeroCopy functionality.

Documentation =====

The installation archive includes a "doc" sub-folder containing the pylon Programmer's Guide and API reference documentation. Open the index.html file with an internet browser.

Sample Programs =====

The installation archive includes a set of sample programs. These sample programs are simple command line programs showing the basic pylon use cases. They are located in the "Samples" folder. The folder contains a top-level Makefile that can be used to build the different sample programs.

Before calling "make" you must setup your environment for pylon to find the necessary dependencies:

```
cd Samples
source <path to pylon>/bin/pylon-setup-env.sh <path to pylon>
make
```

Camera Emulator =====

In addition to the GigE Vision transport layer, this release contains a transport layer able to create simple camera emulator devices that allow you to develop applications without having a physical camera device attached. The emulator has very limited functionality, but is able to create test images for different bit depths.

The number of available emulator devices can be controlled by exporting the PYLON_CAMEMU environment variable. For example,

```
export PYLON_CAMEMU=2
```

will provide two emulator devices. These devices are accessible both by using the pylon API and the pylon Viewer program.

When PYLON_CAMEMU is not set, no emulator devices are provided.

Troubleshooting =====

GigE Devices -----

* Problem: I can't see my camera in the pylon Viewer, even after waiting for more than one minute. (Note: Depending on the camera's and adapter's IP configuration it can take up to one minute until a valid IP address is assigned to the camera.)

Solution: Start the Basler IP Configurator (/opt/pylon4/bin/IpConfigurator). Is the camera shown by the IP Configurator?
If "yes", the camera's IP address is in a different subnet than the network adapter's IP address. Both, the camera and the network adapter, must use IP addresses within the same subnet. Refer to the INSTALL document for the IP address setup.

If "no", the most likely reason is an enabled firewall. As described in the INSTALL document, disable the firewall for those network adapters cameras will be connected to. If you don't have a firewall, enabled reverse-path filtering in the kernel may prevent detection of the camera. Refer to the next problem description for further details.

- * Problem: The IP Configurator can't see my camera. I'm not able to reconfigure the camera to make it visible again.

Solution: First make sure you don't have a firewall enabled on your network interface the camera is connected to.

If you still can't see the camera, reverse path filtering in the kernel may prevent the IP Configurator to detect the camera. On some Linux distributions reverse path filtering may prevent the discovery of GigE Vision cameras. This can happen if the camera's IP is not within the same subnet as the network adapter the camera is attached to. Normally the IP Configurator can handle this by using broadcasts to discover the camera on any subnet. Reverse-path filtering may prevent the IP Configurator to receive the answer from the broadcast which in turn prevents the IP Configurator from detecting the camera. To check whether filtering is turned on, run the following command:

```
sysctl -a 2>/dev/null | grep '\.rp_filter'
```

in the output look for the following lines:

```
net.ipv4.conf.all.rp_filter=1
net.ipv4.conf.eth1.rp_filter=1
```

where "eth1" is the network adapter the camera is connected to.

The "net.ipv4.conf.all.rp_filter" is a global switch which must be turned off. The "net.ipv4.conf.eth1.rp_filter" tells whether filtering for the specified network adapter is activated. To disable filtering, you must first turn off filtering for "all" and the specific network interface (in this sample "eth1"). Use the following commands to change the filtering behavior at runtime:

```
sudo sysctl net.ipv4.conf.all.rp_filter=0
sudo sysctl net.ipv4.conf.eth1.rp_filter=0
```

Restart the IP Configurator and check whether the camera(s) are detected. Reconfigure the camera(s) and use "Write Configuration" to make your changes persistent. You can re-enable filtering by executing the same commands but set a value of 1. If you want to turn off filtering permanently, you can edit the same values in /etc/sysctl.conf.

- * Problem: When grabbing images, the CPU load is higher than expected.

Solution: Ensure that jumbo frames are enabled and large network packets are used as described above in the 'Performance Optimization' section.

- * Problem: Grabbing images leads to errors with error code 0x81010014. This error code indicates that the PC received incomplete images, i.e., network packets have been dropped.

Solution 1: Ensure that jumbo frames are enabled as described above in the 'Performance Optimization' section. Also make sure that the camera's packet size is set to a high value, if possible 8192.

Solution 2: Increase the maximum UDP receive buffer size to a value that is large enough, e.g. by issuing the `sudo sysctl net.core.rmem_max = 2097152` command. This allows pylon to increase the socket buffer size to 2 MB to ensure a stable image acquisition. To make this setting persistent, you can add the `net.core.rmem_max` setting to the /etc/sysctl.conf file.

Solution 3: Ensure that the application has the required permissions to set the pylon receive thread's priority to real-time thread priority as described in the INSTALL document.

Solution 4: Check the cable. For GigE, Cat 5e cables are recommended. Poor cable quality or damaged cables can lead to unrecoverable transmission errors.

Solution 5: The amount of data produced by the camera(s) exceeds the amount of bandwidth the network adapter(s) or the PC can provide. Reduce the camera frame rate by increasing the camera's "Inter Packet Delay" parameter. When using the pylon API, the inter packet delay is controlled via the GevSCPD parameter.

- * Problem: The pylon Viewer seems to acquire images but no images are displayed.

Solution 1: Start the pylon Viewer from the command line to see if the viewer prints out any messages.

Solution 2: The viewer only receives incomplete frames. In that case, error messages such as "Failed to grab image: GX status 0x81010014" are displayed. Please refer to the previous problem for tips about how to solve this issue.

Solution 3: No error messages are printed out. All data packets transmitted from the camera seem to be discarded. Make sure that the camera's current packet size doesn't exceed the network adapter's MTU size.

If the packet size is less than or equal to the MTU size, but greater than 1500, stop image acquisition and set the camera's packet size to 1500. Restart image acquisition. If an image is displayed using this packet size, either jumbo frames are not enabled on the network adapter or the adapter doesn't support jumbo frames.

Solution 3: Grabbing images is failing for other reasons. Please report the error messages printed out by the viewer to Basler technical support.

U3V Devices

- * Problem: The camera is not found during enumeration.
Solution: Check the user permissions on the device. You must have read/write permissions. See the output of
ls -lR /dev/bus/usb
to check if you have read/write permissions. To setup the correct udev rules, execute:
./setup-usb.sh
from within the directory containing this README.

Known Issues

=====

- * Same IP address for camera and network interface.
If the camera has been assigned a permanent IP address that equals the address assigned to the network adapter, the camera will not be accessible, even when using the IP Configurator.

Either change the network adapter's address or use a different PC to assign a different IP address to the camera.
- * Error messages are printed by the pylon Viewer when image acquisition is stopped. When the pylon Viewer is started from a command line and image acquisition is stopped, the Viewer prints out error messages such as "Failed to grab image: GX status 0x81010017". This diagnostic output indicates that image buffers have been cancelled by the pylon GigE vision library on user request instead of being filled with image data. This is expected behavior.

Version Infos

=====

The pylon libraries have been built using the following tools.

Linux x86 32bit/64bit:

These binaries are build with the Linux standard base (lsb) SDK version 4.1
This is roughly the equivalent to:
libc-2.4.so
libstdc++.so.6.0.6

Linux armel (soft float):
Architecture: armv5t
libc-2.8.so
libstdc++.so.6.0.10

Linux armhf (hard float):
Architecture: armv6
libc-2.9.so
libstdc++.so.6.0.13