

Introduction

Installing Vimba on Torizon with Debian container requires several installation steps in the correct order. We have tested the following instructions on Toradex Apalis iMX8.

Installing Vimba on Torizon with Debian container

Step 1: Install Torizon with Debian container

To install and set up Torizon with Debian container, follow the instructions of the README on GitHub. If you don't use a camera with CSI-2 interface, skip the steps for kernel driver installation.

https://github.com/alliedvision/linux_imx8_apalis

Step 2: Download Vimba for ARM64

Download Vimba for ARM64 from <https://www.alliedvision.com/en/products/software.html>.

Step 3: Install the USB transport layer on the Torizon OS



Installation order

Install the USB transport layer first on the Torizon OS and then in the Debian container as described in the following instructions. The reason is that the udev rules to create a proper USB device must be applied on the base operating system.

The transport layers and their respective Install.sh scripts are located in Allied Vision/Vimba_x.x. Before starting the Debian container:

1. Extract the Vimba archive on the Torizon OS.
2. Run the Install.sh script for the USB transport layer. If error messages are displayed, ignore them.

Now the transport layers can be installed in the Debian container.

Step 4: Install the USB transport layer in the Debian container

Start the Debian container with the following command. If you start Debian on Torizon for the first time, the image will be downloaded automatically. If you already have an existing Debian image, please replace the image name "torizon/arm64v8-debian-weston-vivante:buster" with the name of your existing image. This container will run in privileged mode, which allows to access USB camera devices.

```
docker run -e ACCEPT_FSL_EULA=1 --rm --name=avt-container --net=host --cap-add CAP_SYS_TTY_CONFIG \
-v /dev:/dev -v /tmp:/tmp -v /run/udev:/run/udev/ --privileged \
--device-cgroup-rule='c 4:* rmw' --device-cgroup-rule='c 13:* rmw' \
--device-cgroup-rule='c 199:* rmw' --device-cgroup-rule='c 226:* rmw' \
torizon/arm64v8-debian-weston-vivante:buster \
weston-launch --tty=/dev/tty7 --user=torizon
```

Step 5: Start another session to copy the Vimba archive to the docker container

1. In another session (ssh or serial), log in to Torizon and copy the Vimba archive to the docker container:
`docker cp <path-to-Vimba_ARM64.tar.gz> avt-container:/home/torizon`
2. In the container, extract the Vimba archive and install the USBTL and GigETL with the Install.sh script.

Step 6: Commit the changes to the image

1. In the Torizon session (ssh or serial), get the id of the running container: `docker ps -a`
2. Save the container as new image or overwrite the existing one:
`docker commit <container_id> new_image_name:tag_name`

Step 7: Reboot the system

Reboot the system to apply the changes.

Step 8: Start the docker container

Run the command from Step 4 (`docker run...`) to start the docker container (adjust the image name if you created a new one).

Step 9: Install libpng12

Vimba Viewer requires libpng12. If you don't want to use Vimba Viewer, you can omit this step.

```
# Download libpng12 from https://packages.ubuntu.com/xenial/libpng12-0
tar xvf libpng_1.2.54.orig.tar.xz
cd libpng-1.2.54
./autogen.sh
./configure
make -j8 # Use the number of kernels of your system
sudo make install
sudo ldconfig
# Reboot the system
```

To compile from the sources, navigate to the examples directory, for example, Vimba Viewer. Compiling will create a binary linked to libpng16, which is already on the system.

```
# Navigate to your Vimba installation directory
cd VimbaCPP/Examples/VimbaViewer/Build/Make
make -j8 # Use the number of kernels of your system
sudo make install
```

Vimba and the Transport Layers can now be used inside the Debian container with USB and GigE cameras.

Disclaimer

For the latest version of this document, please visit our website. All trademarks are acknowledged as property of their respective owners. Copyright © 2020 Allied Vision Technologies