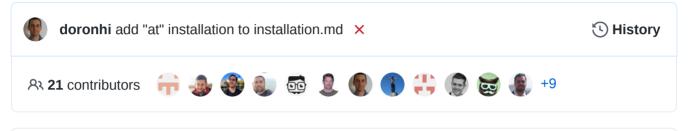
#### ☐ IntelRealSense / librealsense







## **Linux Ubuntu Installation**

**Note:** Due to the USB 3.0 translation layer between native hardware and virtual machine, the *librealsense* team does not support installation in a VM. If you do choose to try it, we recommend using VMware Workstation Player, and not Oracle VirtualBox for proper emulation of the USB3 controller.

Please ensure to work with the supported Kernel versions listed here and verify that the kernel is updated properly according to the instructions.

## **Ubuntu Build Dependencies**

The scripts and commands below invoke wget, git, add-apt-repository which may be blocked by router settings or a firewall. Infrequently, apt-get mirrors or repositories may also timeout. For *librealsense* users behind an enterprise firewall, configuring the system-wide Ubuntu proxy generally resolves most timeout issues.

## **Prerequisites**

1 di 7

**Important:** Running RealSense Depth Cameras on Linux requires patching and inserting modified kernel drivers. Some OEM/Vendors choose to lock the kernel for modifications. Unlocking this capability may require modification of BIOS settings

#### Make Ubuntu Up-to-date:

- Update Ubuntu distribution, including getting the latest stable kernel:
  - sudo apt-get update && sudo apt-get upgrade && sudo apt-get dist-upgrade

**Note:** On stock Ubuntu 14 LTS systems and kernels prior to 4.4.0-04 the standard apt-get upgrade command is not sufficient to bring the distribution to the latest recommended baseline.

It is recommended to upgrade the distribution with:

- sudo apt-get install --install-recommends linux-generic-lts-xenial xserver-xorg-core-lts-xenial xserver-xorg-lts-xenial xserver-xorg-video-all-lts-xenial xserver-xorg-input-all-lts-xenial libwayland-egl1-mesa-lts-xenial
- Update OS Boot and reboot to enforce the correct kernel selection with sudo update-grub && sudo reboot
- Interrupt the boot process at Grub2 Boot Menu -> "Advanced Options for Ubuntu" and select the kernel version installed in the previous step. Press and hold SHIFT if the Boot menu is not presented.
- Complete the boot, login and verify that a supported kernel version (4.[4,8,10,13,15,16]]) is in place with uname -r

#### Download/Clone librealsense github repository:

- Get *librealsense* sources in one of the following ways:
  - Download the complete source tree with git
     git clone https://github.com/IntelRealSense/librealsense.git
  - Download and unzip the latest stable version from master branch: https://github.com/IntelRealSense/librealsense/archive/master.zip

#### **Prepare Linux Backend and the Dev. Environment:**

1. Navigate to *librealsense* root directory to run the following scripts. Unplug any connected Intel RealSense camera.

2.

Install the core packages required to build *librealsense* binaries and the affected kernel modules:

```
sudo apt-get install git libssl-dev libusb-1.0-0-dev pkg-config libgtk-3-dev
```

Distribution-specific packages:

- $\circ\,$  Ubuntu 14 or when running Ubuntu 16.04 live-disk:
  - ./scripts/install\_glfw3.sh
- o Ubuntu 16:

```
sudo apt-get install libglfw3-dev
```

o Ubuntu 18:

```
sudo apt-get install libglfw3-dev libgl1-mesa-dev libglu1-mesa-dev at
```

**Cmake Note**: certain librealsense CMAKE flags (e.g. CUDA) require version 3.8+ which is currently not made available via apt manager for Ubuntu LTS. Go to the official CMake site to download and install the application

**Note** on graphic sub-system utilization:

*glfw3*, *mesa* and *gtk* packages are required if you plan to build the SDK's OpenGL-enabled examples. The *librealsense* core library and a range of demos/tools are designed for headless environment deployment.

3. Run Intel Realsense permissions script from librealsense root directory:

```
./scripts/setup_udev_rules.sh
```

Notice: One can always remove permissions by running: ./scripts /setup\_udev\_rules.sh --uninstall

- 4. Build and apply patched kernel modules for:
  - \* Ubuntu 14/16/18 with LTS kernel

```
./scripts/patch-realsense-ubuntu-lts.sh
```

\* Ubuntu with Kernel 4.16

```
./scripts/patch-ubuntu-kernel-4.16.sh
```

\* Intel® Joule™ with Ubuntu Based on the custom kernel provided by Canonical Ltd.

```
./scripts/patch-realsense-ubuntu-xenial-joule.sh
```

The script above will download, patch and build realsense-affected kernel modules (drivers).

Then it will attempt to insert the patched module instead of the active one. If failed the original uvc modules will be restored.

#### Arch-based distributions

- Install the base-devel package group.
- Install the matching linux-headers as well (i.e.: linux-lts-headers for the linux-lts kernel).
- Navigate to the scripts folder

```
cd ./scripts/
```

o Then run the following script to patch the uvc module:

```
./patch-arch.sh
```

 Odroid XU4 with Ubuntu 16.04 4.14 image Based on the custom kernel provided by Hardkernel

```
./scripts/patch-realsense-ubuntu-odroid.sh
```

Some additional details on the Odroid installation can also be found in installation\_odroid.md

Check the patched modules installation by examining the generated log as well as inspecting the latest entries in kernel log:

```
sudo dmesg | tail -n 50
```

The log should indicate that a new uvcvideo driver has been registered.

Refer to Troubleshooting in case of errors/warning reports.

#### 5. TM1-specific:

 Tracking Module requires hid\_sensor\_custom kernel module to operate properly. Due to TM1's power-up sequence constraints, this driver is required to be loaded during boot for the HW to be properly initialized.

In order to accomplish this, add the driver's name *hid\_sensor\_custom* to /etc/modules file, eg:

```
echo 'hid_sensor_custom' | sudo tee -a /etc/modules
```

## **Building librealsense2 SDK**

• On Ubuntu 14.04, update your build toolchain to gcc-5:

- o sudo add-apt-repository ppa:ubuntu-toolchain-r/test
- o sudo apt-get update
- o sudo apt-get install gcc-5 g++-5
- o sudo update-alternatives --install /usr/bin/gcc gcc /usr/bin
  /gcc-5 60 --slave /usr/bin/g++ g++ /usr/bin/g++-5
- o sudo update-alternatives --set gcc "/usr/bin/gcc-5" You can check the gcc version by typing: gcc -v If everything went fine you should see gcc 5.0.0.
- Navigate to librealsense root directory and run mkdir build && cd build

#### Run CMake:

- cmake ../ The default build is set to produce the core shared object and unit-tests binaries in Debug mode. Use -DCMAKE\_BUILD\_TYPE=Release to build with optimizations.
- cmake ../ -DBUILD\_EXAMPLES=true Builds librealsense along with the demos and tutorials
- o cmake ../ -DBUILD\_EXAMPLES=true -DBUILD\_GRAPHICAL\_EXAMPLES=false
  - For systems without OpenGL or X11 build only textual examples

•

5 di 7

Recompile and install librealsense binaries:

sudo make uninstall && make clean && make && sudo make install

The shared object will be installed in /usr/local/lib , header files in /usr/local/include .

The binary demos, tutorials and test files will be copied into /usr/local/bin**Tip:** Use make - jx for parallel compilation, where x stands for the number of CPU cores available:

sudo make uninstall && make clean && make \*\*-j8\*\* && sudo make install

This enhancement may significantly improve the build time. The side-effect, however, is that it may cause a low-end platform to hang randomly.

**Note:** Linux build configuration is presently configured to use the V4L2 backend by default.

**Note:** If you encounter the following error during compilation <code>gcc:internal</code> compiler error it might indicate that you do not have enough memory or swap space on your machine. Try closing memory consuming applications, and if you are running inside a VM, increase available RAM to at least 2 GB.

**Note:** You can find more information about the available configuration options on this wiki page.

2. Install IDE (Optional): We use QtCreator as an IDE for Linux development on Ubuntu. Follow the link for QtCreator5 installation

# Troubleshooting Installation and Patch-related Issues

Error	Cause	Correction Steps
git.launchpad access timeout	Behind Firewall	Configure Proxy Server
<pre>dmesg: uvcvideo: module verification failed: signature and/or required key missing - tainting kernel</pre>	A standard warning issued since Kernel 4.4-30+	Notification only - does not affect module's functionality
sudo modprobe uvcvideo produces dmesg: uvc kernel module is not loaded	The patched module kernel version is incompatible with	Verify the actual kernel version with uname -r. Revert and

https://github.com/IntelRealSense/librealsense/...

Error	Cause	<b>Correction Steps</b>
	the resident kernel	proceed from Make Ubuntu Up- to-date step
Execution of ./scripts/patch-video-formats-ubuntu-xenial.sh fails with fatal error: openssl/opensslv.h	Missing Dependency	Install <i>openssl</i> package from Video4Linux backend preparation step

7 di 7 20/12/20, 17:17