# Camera Installation

Thomas Schenk

Contents

[Camera Installation 1](#_Toc129273487)

[Why a new camera? 3](#_Toc129273488)

[Process of connecting the new camera 4](#_Toc129273489)

[Step 1: Preinstall software with the new camera 4](#_Toc129273490)

[Step 2: Integrate the new software in a ROS environment 7](#_Toc129273491)

[Step 3: Rewriting the topic listener of the robot 9](#_Toc129273492)

## Why a new camera?

A research already has been done by another group member, diving deeper into why and which camera is the best. For comprehensive purposes I would like to reiterate why we would like a new camera to be present on the robot.

The current camera on the robot works and detects both depth and “normal” camera pictures. However the quality and most importantly, the speed of this camera leaves much to be desired. But new functionalities like infrared is also absent on this camera.

In hopes to increase both the speed and quality of the pictures taken, research determined that the best camera for this was the Intel Realsense L515. This camera was of a different brand than the camera present on the robot. The software which connects the camera to the ROS distro had to be rewritten.

## Process of connecting the new camera

### Step 1: Preinstall software with the new camera

In order to test if the camera is fully working and install the required SDK. It was the easiest and most efficient step to first link the camera with Linux. Our end goal here is to launch the official Realsense camera viewer software.

The prerequisites to this are that the Realsense SDK is fully installed.

Luckily, Realsense has amazing documentation showing us how to install their SDK.

“  
Installing the packages:

- Register the server's public key:

`sudo apt-key adv --keyserver keyserver.ubuntu.com --recv-key F6E65AC044F831AC80A06380C8B3A55A6F3EFCDE || sudo apt-key adv --keyserver hkp://keyserver.ubuntu.com:80 --recv-key F6E65AC044F831AC80A06380C8B3A55A6F3EFCDE`

In case the public key still cannot be retrieved, check and specify proxy settings: `export http\_proxy="http://<proxy>:<port>"`

, and rerun the command. See additional methods in the following [link](https://unix.stackexchange.com/questions/361213/unable-to-add-gpg-key-with-apt-key-behind-a-proxy).

- Add the server to the list of repositories:

`sudo add-apt-repository "deb https://librealsense.intel.com/Debian/apt-repo $(lsb\_release -cs) main" -u`

- Install the libraries (see section below if upgrading packages):

`sudo apt-get install librealsense2-dkms`

`sudo apt-get install librealsense2-utils`

The above two lines will deploy librealsense2 udev rules, build and activate kernel modules, runtime library and executable demos and tools.

- Optionally install the developer and debug packages:

`sudo apt-get install librealsense2-dev`

`sudo apt-get install librealsense2-dbg`

With `dev` package installed, you can compile an application with \*\*librealsense\*\* using `g++ -std=c++11 filename.cpp -lrealsense2` or an IDE of your choice.

Reconnect the Intel RealSense depth camera and run: `realsense-viewer` to verify the installation.  
”

By running the command mentioned in the latest sentence, we get an output! The realsense software opens up and displays some footage.

A screenshot of a video game

Description automatically generated

Please note that this picture is a 3D picture of amazing quality. These are clothes on my bed and it is able to detect all their contours and lines while im holding the camera with my hand. Basically the least efficient environment for this camera, while still performing quite well.

A computer screen capture

Description automatically generated with medium confidence

A screenshot of a computer

Description automatically generated with medium confidence

These are some of the 2D capabilities of the L515, as mentioned previously; an infrared module is also present (bottom left)

### Step 2: Integrate the new software in a ROS environment

Now that its established that the camera is working, we need to install the packages needed to incorporate this camera in ROS. These packages make use of the previously installed SDK and make sure all their data gets transmitted to camera topics. These topics are then used by the robot to gather camera data.

Luckily, there is a ROS library which incorporates these functionalities from the get-go.

Install Intel&reg; RealSense&trade; ROS from Sources

- Create a [catkin](http://wiki.ros.org/catkin#Installing\_catkin) workspace

\*Ubuntu\*

```bash

mkdir -p ~/catkin\_ws/src

cd ~/catkin\_ws/src/

```

\*Windows\*

```batch

mkdir c:\catkin\_ws\src

cd c:\catkin\_ws\src

```

- Clone the latest Intel&reg; RealSense&trade; ROS from [here](https://github.com/intel-ros/realsense/releases) into 'catkin\_ws/src/'

```bashrc

git clone https://github.com/IntelRealSense/realsense-ros.git

cd realsense-ros/

git checkout `git tag | sort -V | grep -P "^2.\d+\.\d+" | tail -1`

cd ..

```

- Make sure all dependent packages are installed. You can check .travis.yml file for reference.

- Specifically, make sure that the ros package \*ddynamic\_reconfigure\* is installed. If \*ddynamic\_reconfigure\* cannot be installed using APT or if you are using \*Windows\* you may clone it into your workspace 'catkin\_ws/src/' from [here](https://github.com/pal-robotics/ddynamic\_reconfigure/tree/kinetic-devel)

```bash

catkin\_init\_workspace

cd ..

catkin\_make clean

catkin\_make -DCATKIN\_ENABLE\_TESTING=False -DCMAKE\_BUILD\_TYPE=Release

catkin\_make install

```

\*Ubuntu\*

```bash

echo "source ~/catkin\_ws/devel/setup.bash" >> ~/.bashrc

source ~/.bashrc

```

\*Windows\*

```batch

devel\setup.bat

```

## Usage Instructions

### Start the camera node

To start the camera node in ROS:

```bash

roslaunch realsense2\_camera rs\_camera.launch

```

This will stream all camera sensors and publish on the appropriate ROS topics.

Please note that this is NOT a ROS distro, but a library. This installation will require a ROS distro to be installed on the target already. In our case, we should have done this already.

By running the command shown, we can visualize the same data we previously gathered; but in RVIZ. A visualization tool which makes use of ROS topics to gather its data, much like how the robot does.

A screenshot of a computer

Description automatically generated with medium confidence

This picture takes almost all topics from the camera and puts it in RVIZ, this is what the robot sees too.

### Step 3: Rewriting the topic listener of the robot

To make the robot use this camera, we have to configure its listener files to listen to these new camera topics. This is something which is still a work in progress, thus I cannot add it here already.