**Work with Camera**

Before plugging in the camera to the USB let’s check the output of:

ls /dev/video\*

Knowing our camera device node path (/dev/video2) we will now check the USB attributes of the camera using the command:

udevadm info --name=/dev/video2 --attribute-walk

Camera parameters

v4l2-ctl --list-formats-ext -d /dev/video2

Install uvc\_camera package

$ sudo apt-get install ros-kinetic-uvc-camera

Also install image related packages

$ sudo apt-get install ros-kinetic-image-\*

$ sudo apt-get install ros-kinetic-image-view

From here the actual maneuver confirmation As always,

$ roscore

Launch another terminal,

$ rosrun uvc\_camera uvc\_camera\_node

Then, various messages come out.

[ WARN] [1522822252.725651583]: [camera] does not match name narrow\_stereo in file /home/kondo/.ros/camera\_info/camera.yaml

Although the warning appears, of course, I have not set the calibration file of the internal parameters, so ignore it now. If there is no problem with the connection, the USB camera light should be on.

If you check [rostopic list] etc., you can see that there is a topic message.

(/ Camera\_info, / image\_raw etc.)

Image check

With this node up (now, roscore and uvc\_camera\_node), start up a new terminal,

$ rosrun image\_view image\_view image:=/image\_raw

$ rosrun uvc\_camera uvc\_camera\_node \_device:=/dev/video0 \_camera\_info\_url:=file:///home/{user\_name}/.../camera.yaml

Then, the viewer stands up and you can see the camera image.

You can also view the video from rqt\_image\_view and RViz.

$ rosrun rqt\_image\_view rqt\_image\_view

Call camera node from launch file

catkin\_create\_pkg runcamera catkin roscpp std\_msgs

catkin\_create\_pkg runcamera catkin rospy std\_msgs

catkin\_create\_pkg beginner\_tutorials std\_msgs rospy roscpp

<launch>

<node pkg="uvc\_camera" name="camera1" type="uvc\_camera\_node" output="screen">

<param name="device" type="string" value="/dev/video0"/>

<param name="width" type="int" value="640"/>

<param name="height" type="int" value="480"/>

<param name="pixel\_format" value="mjpeg"/>

<param name="camera\_frame\_id" value="usb\_cam"/>

<param name="io\_method" value="mmap"/>

<param name="frame\_id" value="camera"/>

<param name="camera\_info\_url" type="string" value="file:///home/{user\_name}/.../camera.yaml"/>

</node>

<node name="image\_view" pkg="image\_view" type="image\_view" output="screen">

<remap from="image" to="image\_raw">

</remap>

</node>

</launch>

$ rospack list-names

usb\_cam

~video\_device (string, default: "/dev/video0")

The device the camera is on.

~image\_width (integer, default: 640)

Image width

~image\_height (integer, default: 480)

Image height

~pixel\_format (string, default: "mjpeg")

Possible values are mjpeg, yuyv, uyvy

~io\_method (string, default: "mmap")

Possible values are mmap, read, userptr

~camera\_frame\_id (string, default: "head\_camera")

The camera's tf frame

~framerate (integer, default: 30)

The required framerate

~contrast (integer, default: 32)

Contrast of video image (0-255)

~brightness (integer, default: 32)

Brightness of video image (0-255)

~saturation (integer, default: 32)

Saturation of video image (0-255)

~sharpness (integer, default: 22)

Sharpness of video image (0-255)

~autofocus (boolean, default: false)

Enable camera's autofocus

~focus (integer, default: 51)

If autofocus is disabled, the focus of the camera (0=at infinity)

~camera\_info\_url (string, default: )

An url to the camera calibration file that will be read by the CameraInfoManager class

~camera\_name (string, default: head\_camera)

The camera name. This must match the name in the camera calibration

Web video server

Install web-video-server ROS node

$ sudo apt install ros-kinetic-web-video-server

To make it right, create catkin workspace for our custom launch file:

$ mkdir -p ~/rosvid\_ws/src

$ cd ~/rosvid\_ws

$ catkin\_make

$ source devel/setup.bash

Then create ROS package:

$ cd src

$ catkin\_create\_pkg vidsrv std\_msgs rospy roscpp

And create launch file using nano, vim, etc.:

$ mkdir -p vidsrv/launch

$ nano vidsrv/launch/vidsrv.launch <https://gist.github.com/WinKILLER/3dea1ab7cf5645c99f1271496cca2e21>

<launch>

<!-- This node description you can take from usb\_cam-test.launch -->

<node name="usb\_cam" pkg="usb\_cam" type="usb\_cam\_node" output="screen" >

<param name="video\_device" value="/dev/video0" />

<param name="image\_width" value="640" />

<param name="image\_height" value="480" />

<param name="pixel\_format" value="yuyv" />

<param name="camera\_frame\_id" value="usb\_cam" />

<param name="io\_method" value="mmap"/>

</node>

<!-- This node will launch web video server -->

<node name="web\_video\_server" pkg="web\_video\_server" type="web\_video\_server" />

</launch>

Build package:

$ cd ..

$ catkin\_make

Again, start ROS core on master

$ roscore

And run created launch file:

$ roslaunch vidsrv vidsrv.launch

$ rospack list-names