How to connect a PS4 Dualshock controller to Jetson TX2 (L4T 32.2.0) with Ubuntu GNOME 18.04

Created by: David Rodriguez ([davidrm@dc.tohoku.ac.jp](mailto:davidrm@dc.tohoku.ac.jp))

Last update: Nov. 1, 2019

Updated by: David Rodriguez

# Preparation

Make sure that PIP3 (python 3 package manager) is installed. *Note: the same would apply if you need pip for python 2. Just substitute pip3 for pip.*

$ pip3 --version

If pip3 is not installed, start by updating the package list using the following:

$ sudo apt update

$ sudo apt upgrade

Use the following command to install pip3:

$ sudo apt install python3-pip

All the dependencies will be also installed.

*Note: When installing python modules globally, it is highly recommended to install distribution-provided python modules using the apt package manager since they are tested to work properly on Ubuntu systems. You should install python modules globally using pip only if there is no package available through the package manager.*

# Installation

Install **ds4drv**, a Sony DualShock 4 userspace driver for linux (<http://github.com/chrippa/ds4drv>)

$ sudo pip3 install ds4drv

# Connection

### Option 1: Connecting via bluetooth.

Run **ds4drv**.

$ sudo ds4drv

Hold **SHARE** and **PS** buttons in the PS4 controller until its indicator flashes. Ds4drv will automatically start searching for bluetooth devices and establish a connection.

Once **ds4drv** finds your controller, it will be assigned to */dev/input/js~*

### Option 2: Connecting via USB.

Run **ds4drv** in hidraw mode.

$ sudo ds4drv --hidraw

Connect the PS4 controller via usb to the Jetson. Ds4drv will automatically start searching for bluetooth devices and establish a connection.

Once **ds4drv** finds your controller, it will be assigned to */dev/input/js~*

# Testing

### Option 1: Without ROS

Install and run **jstest:**

$ sudo apt-get install jstest-gtk

$ jstest-gtk

*Note: the -gtk option is only required if you want to see the output displayed on an independent GUI.*

### Option 2: With ROS

First, make sure you have the joy package installed.

$ rospack find joy

If not installed, install the package. *Note: make sure you use your own ROS distribution*:

$ sudo apt install ros-*melodic*-joy

In a new terminal, launch ROS:

$ roscore

In a new terminal, tell the joy node which joystick device to use. Note: the default is js2.

$ rosparam set joy\_node/dev “/dev/input/js~”

Start the joy node

$ rosrun joy joy\_node

In a new terminal, run the following command to see the data published on the joy topic:

$ rostopic echo joy

# Application

### DualShock PS4 Controller buttons and axes mapping

Table of index number of */joy.buttons*:

Index Button name on the controller

0 Square

1 Cross

2 Circle

3 Triangle

4 L1

5 R1

6 L2

7 R2

8 Share

9 Options

10 Left joystick press

11 Right joystick press

12 PS

13 Touchpad press

Table of index number of */joy.axes*:

Index Axis name on the controller

0 Left/Right Axis (left stick)

1 Up/Down Axis (left stick)

2 Left/Right Axis (right stick)

3 L2

4 R2

5 Up/Down Axis (right stick)

9 Left/Right Arrow Btns (+ : left, - : right)

10 Up/Down Arrow Btns (+ : up, - : right)

Others Accelerometers/Gyroscope

# Troubleshooting

### Running ds4drv without sudo permissions.

If you want to use **ds4drv** as a normal user, you need to make sure ds4drv has permission to use certain features of your system. Ds4drv uses the kernel module uinput to create input devices in userland and the module *hidraw* to communicate with DualShock 4 controller (when using *--hidraw*), but this usually requires root permissions. You can change the permissions by copying the udev rules file to */etc/udev/rules.d* (called “50-ds4drv.rules”, check <https://github.com/chrippa/ds4drv/blob/master/udev/50-ds4drv.rules>):

File 50-ds4drv.rules should contain the following lines

|  |  |
| --- | --- |
|  | KERNEL=="uinput", MODE="0666" |
|  | KERNEL=="hidraw\*", SUBSYSTEM=="hidraw", ATTRS{idVendor}=="054c", ATTRS{idProduct}=="05c4", MODE="0666" |
|  | KERNEL=="hidraw\*", SUBSYSTEM=="hidraw", KERNELS=="0005:054C:05C4.\*", MODE="0666" |
|  | KERNEL=="hidraw\*", SUBSYSTEM=="hidraw", ATTRS{idVendor}=="054c", ATTRS{idProduct}=="09cc", MODE="0666" |
|  | KERNEL=="hidraw\*", SUBSYSTEM=="hidraw", KERNELS=="0005:054C:09CC.\*", MODE="0666" |

You may have to reload your udev rules after this with:

$ sudo udevadm control --reload-rules

$ sudo udevadm trigger

Check if linux recognize your PS4 controller.

Run the following command:

$ ls /dev/input

You will see a listing of all of your input devices. The joystick devices are referred to as js~.

PS4 controller not accessible for the ROS joy node.

First, list the permissions of the PS4 controller:

$ ls -l /dev/input/js0

You will see something similar to:

$ crw-rw-XX- 1 root dialout 188, 9 2019-11-01 12:04 /dev/input/js~

If XX is rw, the js device is configured properly. If XX is not rw, you need to run the following:

$ sudo chmod a+rw /dev/input/js~

Ds4drv hidraw mode not detecting your controller

This usually happens when the name associated to your controller is slightly different than the name defined in *hidraw.py*.

First, let’s check which names the controller has associated. Run the following command in a terminal. Make sure the controller is connected via USB.

$ dmesg

You will be able to read something like the following

[18798.634308] input: Sony Interactive Entertainment Wireless Controller as /devices/virtual/input/input21

*Note: if this line does not appear in your terminal, make sure your controller is listed as one of the usb inputs*. $ lsusb

In this case, the name associated with the device is “Sony Interactive Entertainment Wireless Controller”.

Then, go to */usr/local/lib/python3.6/dist-packages/ds4drv/backends* and open the file *hidraw.py*. Add the following line to the script (highlighted line only).

$ HID\_DEVICES = {

**“Sony Interactive Entertainment Wireless Controller”: HidrawUSBDS4Device**

“Sony Computer Entertainment Wireless Controller”: HidrawUSBDS4Device

“Wireless Controller”: HidrawBluetoothDS4Device

}