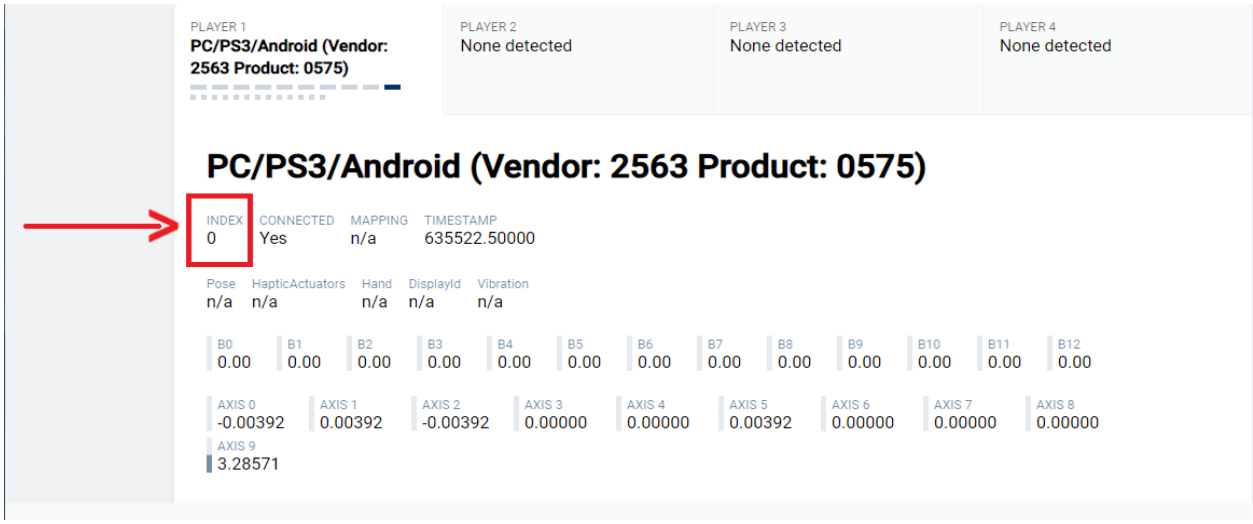
Teleoperations

Step 1: Learn how the gamepad work

- Connect USB adapter to PC
- Go to https://html5gamepad.com
- Check the INDEX of Gamepad. Here the INDEX is 0 (ZERO) in your case might be different so check the index



- The gamepad included supports two working modes. One is PC/PS3/Android mode and another is Xbox 360 mode
- **[Note]** To switch between PC/PS3/Android mode and the Xbox 360 mode, you can long-press the HOME button for about 7s. In Xbox mode, the left joystick is mapped to axes[0] and axes[1], right joystick is mapped to axes[2] and axes[3]. This mode is exactly what the NVIDIA examples use.

We recommend you to set your gamepad to this mode when you use it. Otherwise, you need to modify the codes.

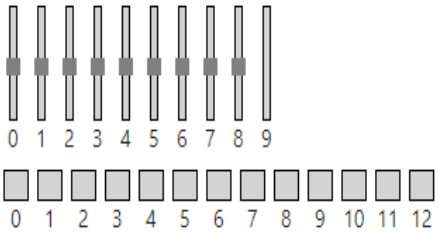
Step 2: Run the test gamepad

- Access JetBot by going to https://<jetbot_ip_address>:8888, navigate to ~/Notebooks/teleoperation
- Open teleoperation.ipynb file and following notebook.
- Modify the index as per the controller

Next, we'll create and display our controller using that index.

[1]:
```
import ipywidgets.widgets as widgets

controller = widgets.Controller(index=0)  # replace with index of your controller

display(controller)
```

Replace your INDEX as per the controller



Step 3: Connect gamepad controller to the robot motors

- Modify axes values if required, here we use axes[0] and axes[1] and try with axes[1] and axes[1] also. Modify the axes as per requirements

Connect gamepad controller to robot motors

Now, even though we've connected our gamepad, we haven't yet attached the controls to our robot! The first, and most simple control we want to attach is the motor control. We'll connect that to the left and right vertical axes using the `dlink` function. The `dlink` function, unlike the `link` function, allows us to attach a transform between the `source` and `target`. Because the controller axes are flipped from what we think is intuitive for the motor control, we'll use a small *lambda* function to negate the value.

> WARNING: This next cell will move the robot if you touch the gamepad controller axes!

[5]:
```
from jetbot import Robot
import traitlets

robot = Robot()

left_link = traitlets.dlink((controller.axes[1], 'value'), (robot.left_motor, 'value'), transform=lambda x: -x)
right_link = traitlets.dlink((controller.axes[1], 'value'), (robot.right_motor, 'value'), transform=lambda x: -x)
```

Modify axes if required

Awesome! Our robot should now respond to our gamepad controller movements. Now we want to view the live video feed from the camera!

Step 4: Create and display image widgets

Create and display Image widget

First, let's display an `Image` widget that we'll use to show our live camera feed. We'll set the `height` and `width` to just 300 pixels so it doesn't take up too much space.

> FYI: The height and width only effect the rendering on the browser side, not the native image resolution before network transport from robot to browser.

[6]:
```
image = widgets.Image(format='jpeg', width=300, height=300)

display(image)
```

Step 5: Create camera instance

[7]:
```
from jetbot import Camera

camera = Camera.instance()
```

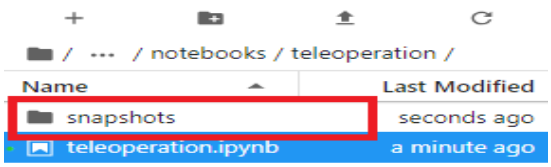
Step 6: Connect camera to image widgets

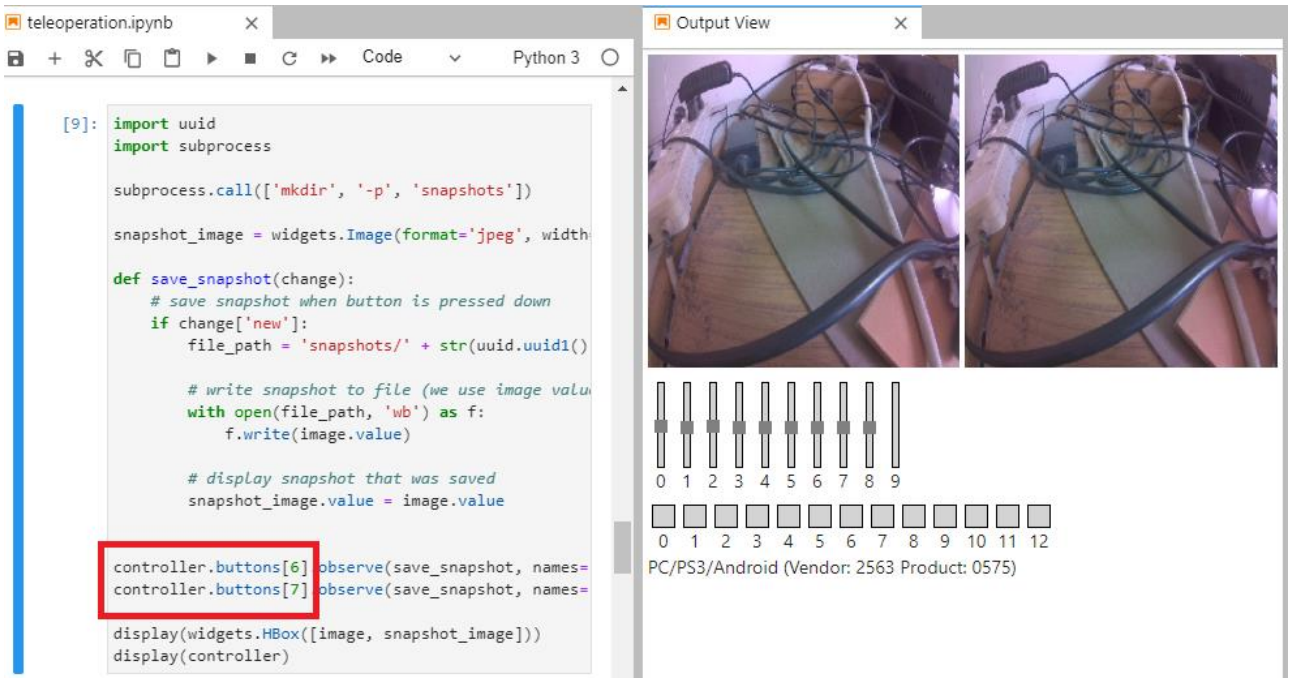
[8]:
```
from jetbot import bgr8_to_jpeg

camera_link = traitlets.dlink((camera, 'value'), (image, 'value'), transform=bgr8_to_jpeg)
```

You should now see the live video feed shown above!

Step 7: Snapshots with gamepad buttons

After running this cell you can see the snapshot folder is created





[Note] You can change the controllers to change the button for the snapshots.Here controller.buttons[6] and controller.buttons[7] both B6 and B7 used to click snapshots