# Application Notes

University of Alberta

ECE 492 Capstone Group #7

Brady Pomerleau, Conor McQuitty, Stefan Vidakovic, Tyler Santos

Application Notes Demo Date: April 10th, 2019

**Git Repository**

Clone our code using the following command

git clone https://github.com/uwdrone/Abzu.git

**Dependencies**

In order to run the code in 'ControllerSender', we have downloaded the following dependencies in advance.

- pygame
- pyUSB

To download and install pygame, run 'pip install pygame'. To download and install pyusb, run 'pip install pyusb'. The instructions can also be seen here https://www.pygame.org/wiki/GettingStarted for pygame and here https://github.com/pyusb/pyusb/blob/master/README.rst .

The rest of the dependencies are pre installed on python and are listed below

- sys
- time
- socket
- signal

**PlayStation Controller**

You need pyUSB installed to use the PlayStation controller. You then plug your controller in and the lights on the controller should flash. Press the PlayStation button in the center of the controller and the first light should turn on, and the rest should stop flashing.

If this doesn't work, try holding the PlayStation button for over 10 seconds and then releasing. This will do a soft reset and try the above steps again. If this still doesn't work, there is a small hole, large enough for a pin to enter on the back of the controller. Use a paperclip to press the button inside, this is the hard reset. After the hard reset, repeat the above steps.

**Static IP**

In order to run our code, the computers IP address must be changed to a static one. This is because the Pi and the computer communicate via an ethernet cable and the computer should

not need to have internet access, so it could potentially work in the middle of a lake or somewhere similar where there would be no wifi available.

First ensure that the ethernet cable from the Pi is plugged into the computers ethernet port.

In order to set the static IP, go to Network Connections (click on Network option at bottom right of screen). Click on Ethernet - Wired connection 1. Then click on the settings option (small ‘gear’ symbol at bottom left of Network Connections window). Go to the IPv4 Settings tab. Change Method to Manual. Add a line to Addresses. Make the Address: 192.168.1.103. Make the Netmask: 24. Make the Gateway: 192.168.1.1. This can be seen in Figure 1. Then press the save button in the bottom right of the window. Once these instructions are followed, your computer should be ready to execute the code.

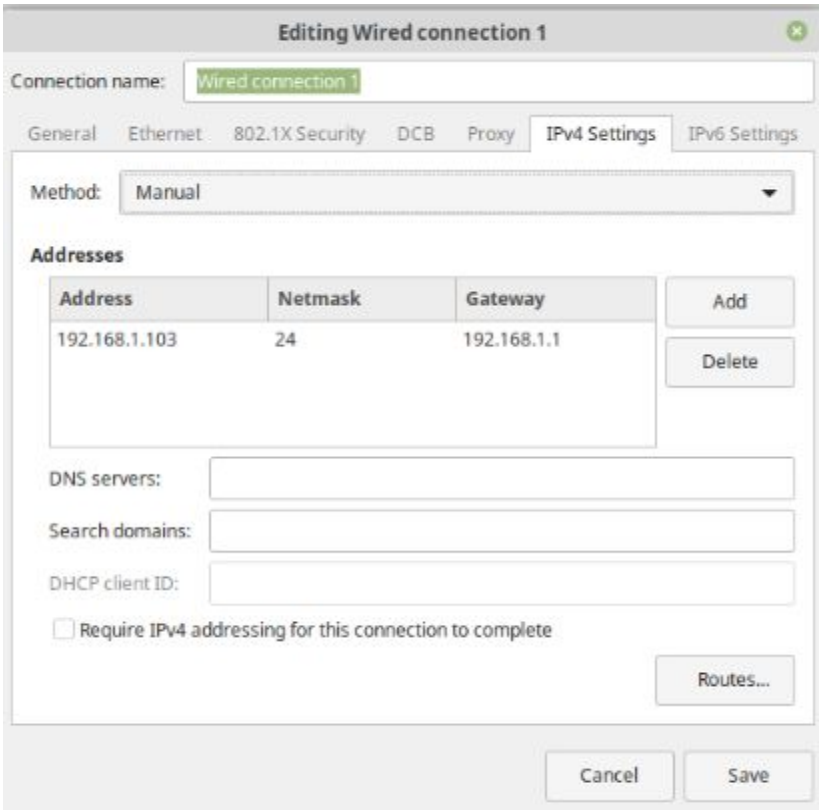


**Figure 1: Network Connection Setup**

**Drone Startup**

Once the static IP is done, the drone is ready to be powered up. Simply connect the drill battery to the power station we have set up. If the drill battery setup is not there, connect to a power supply at 18V.

**Execution Instructions**

You have to open two command prompts. The first one is to SSH into the Pi to control the Pi.

Command Window 1:

ssh pi@192.168.1.102

python3 Abzu/launcherProgram.py

The other command window will run the controller file. In order to execute the following code, you must change directories into the Abzu directory. In order to change into the Abzu, you must change into the directory where you cloned the git repository. i.e. if you cloned the repository into Desktop, you will need to:

Command Window 2:

cd Desktop

cd Abzu

python3 RemoteControl/ControllerSender.py

The code in Command Window 1 must be executed before Command Window 2. The drone can now be controlled with the PlayStation Controller. The controls can be seen in Figure 2.

**Control Guide**

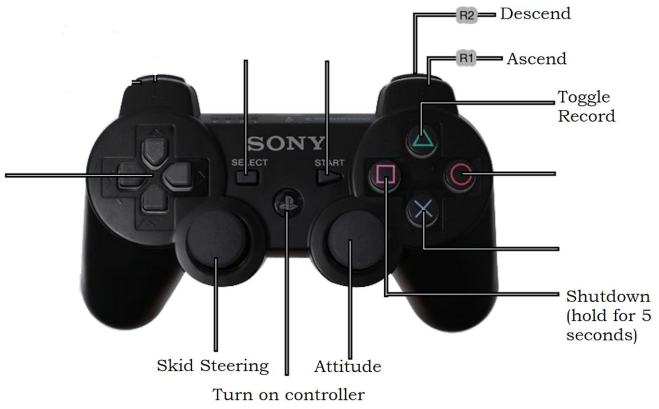


**Figure 2: Controller Layout**

**Video Stream**

To view the video stream, go to your browser and type in:

http://192.168.1.102:8100

**Shut Off**

On the Playstation controller, hold ‘Square’ for 5 seconds and release to kill the program.

To shutdown the pi, run ‘sudo shutdown now’ in Command Window 1.

**Accessing Videos**

From Command Window 2, after Shut Off:

./mv_vids.sh

If you get an execution permission error run this:

chmod -x mv_vids.sh