ENGR 430 Computer Vision

Addendum to Assembly Instructions

# Changing network ssid and WPA Passphrase

Please follow these instructions, to change wifi network name and password: ssid and WPA\_passphrase must be between 8 and 32 characters.

After logging in with the username "duckiesky" and the password "bigbubba" you will be looking at a terminal with a command prompt

Change directory to /etc/hostapd/ by typing the command at the prompt followed by the enter key:  There is a space between the cd and the /

cd /etc/hostapd/

Edit the configuration file with the editor VIM.  However, the configuration file is a special file that only the computer administrator may access.  In ancient times, this administrator was called the "super user" so the sudo command gets its name from "super".  Enter the command:

sudo vim hostapd.conf

when asked for the password enter bigbubba

You will now be looking at the contents of the hostapd.conf file.  Use the arrow keys to navigate and find the ssid and WPA\_passphrase fields.  Delete the "defaultdrone" entry with the "x" key, then insert text by typing the "i" key.  I called mine "steindrone" but this could be anything.  Once in insert mode you may type and backspace as usual.  To exit INSERT MODE hit the escape (Esc) key. You may now use arrows again to navigate. Change the WPA\_passphrase in the same manner.  Make sure the new entries are at least 8 and not more than 32 characters.  Then type the keys ":wq" for "command-write-quit" to save the file and exit.  Do not type the quotes, just :wq

Double check by printing the file to the command window, making sure the changes you have made are reflected and that each entry is on a separate line.  Do this with this command

cat hostapd.conf

This is named because cats are much cooler pets than dogs.

Cycle power on the Pi to reboot.  You may need to have your computer "forget" the defaultdrone network.  You may have to wait awhile for your laptop to recognize the new network.  If this completely fails - (you have cycled power multiple times and waited long enough each time) don't panic, you can re-burn the image to the SD card and start again.

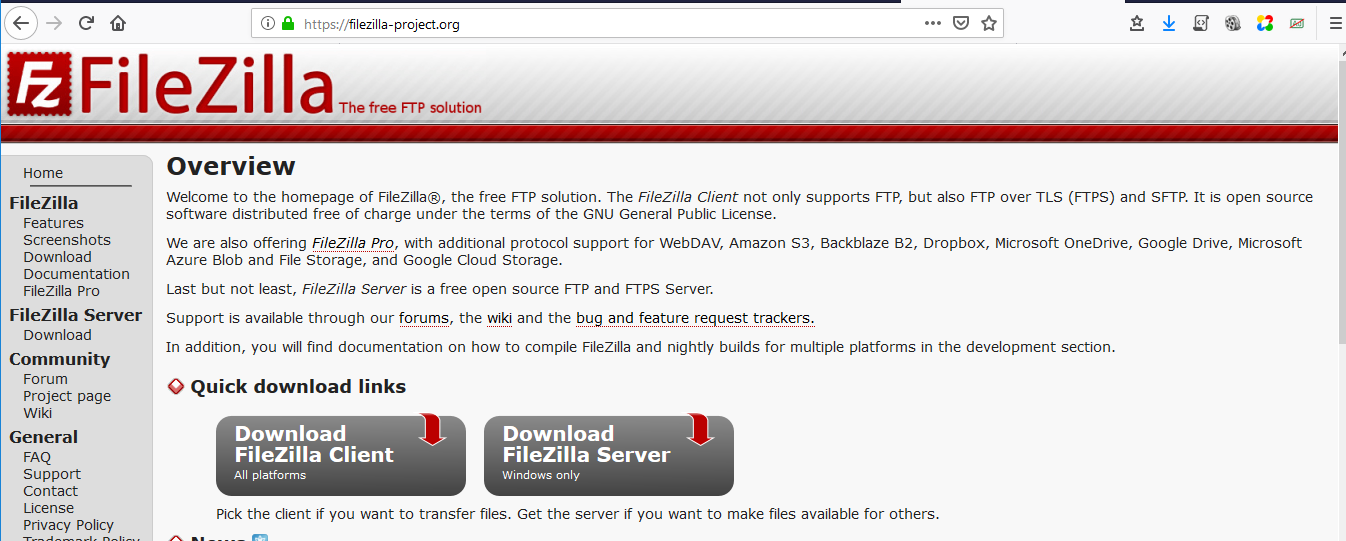
# Changing duckiesky user password

Everyone will be logging in to their Pi with the username “duckiesky” so to decrease the chances of unwanted logins, change the password from “bigbubba” to something else that only you know. Changing password is simple, just execute the “passwd” command at the command prompt. Note that the passwd command will reject passwords that are too short. See below.



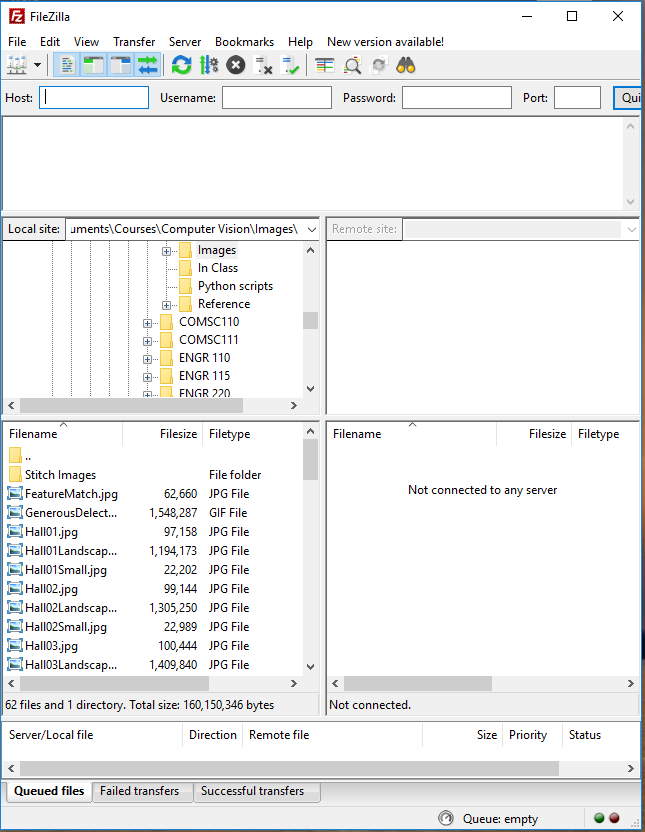
# Transferring files

You will need to routinely transfer files back and forth between your laptop and your Pi. On Windows and Mac a helpful app for doing this if FileZilla

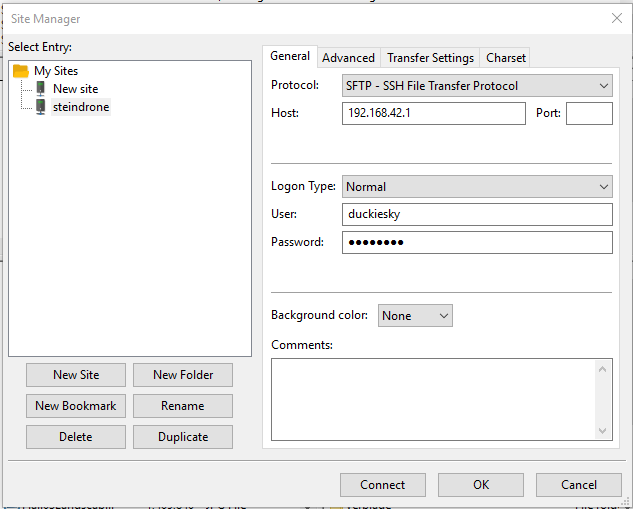


You will want the Client (left button) not the server

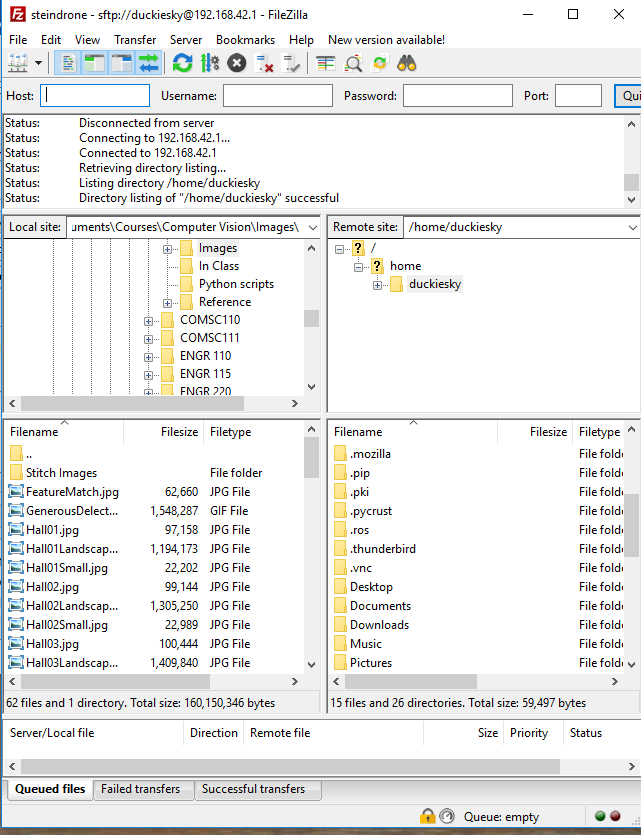
Once installed, launch the filezilla App, it looks like this:



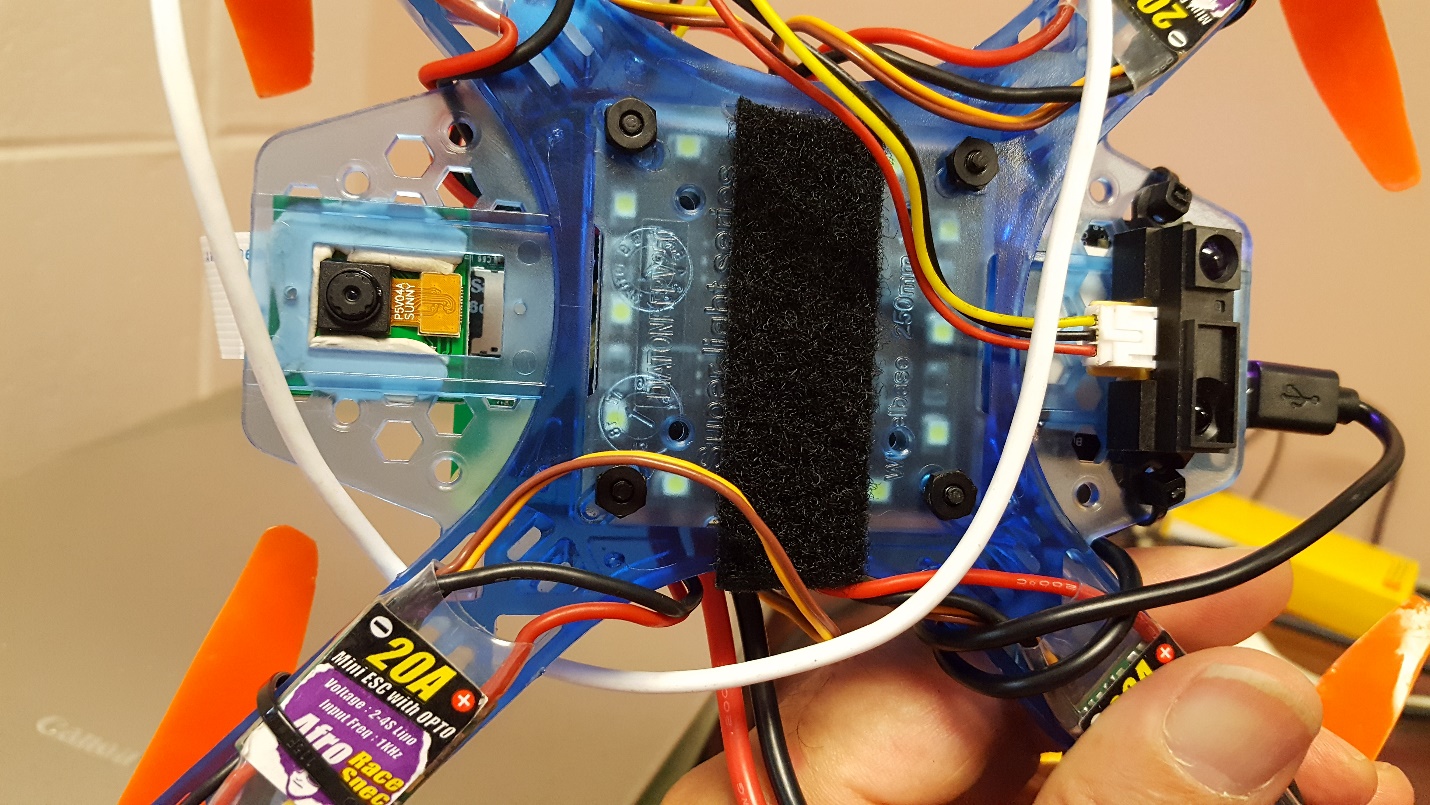
Create a new site and complete the fields as shown



After connecting you will see your laptop folders on the left pane and the Pi’s folders on the right.

You may use Right Mouse-> Download or Right Mouse->Upload to transfer files between computers. You will need to transfer your selfie.jpg image to the Pi to complete HW2. You will also likely need to transfer your files back to your laptop to submit them to Bridges. (Note that another option is to plug a live network wire directly into the network jack on the Pi. Install chromium and you may use that browser to interact with Bridges as usual.)

# Securing the PDB and Pi Mount

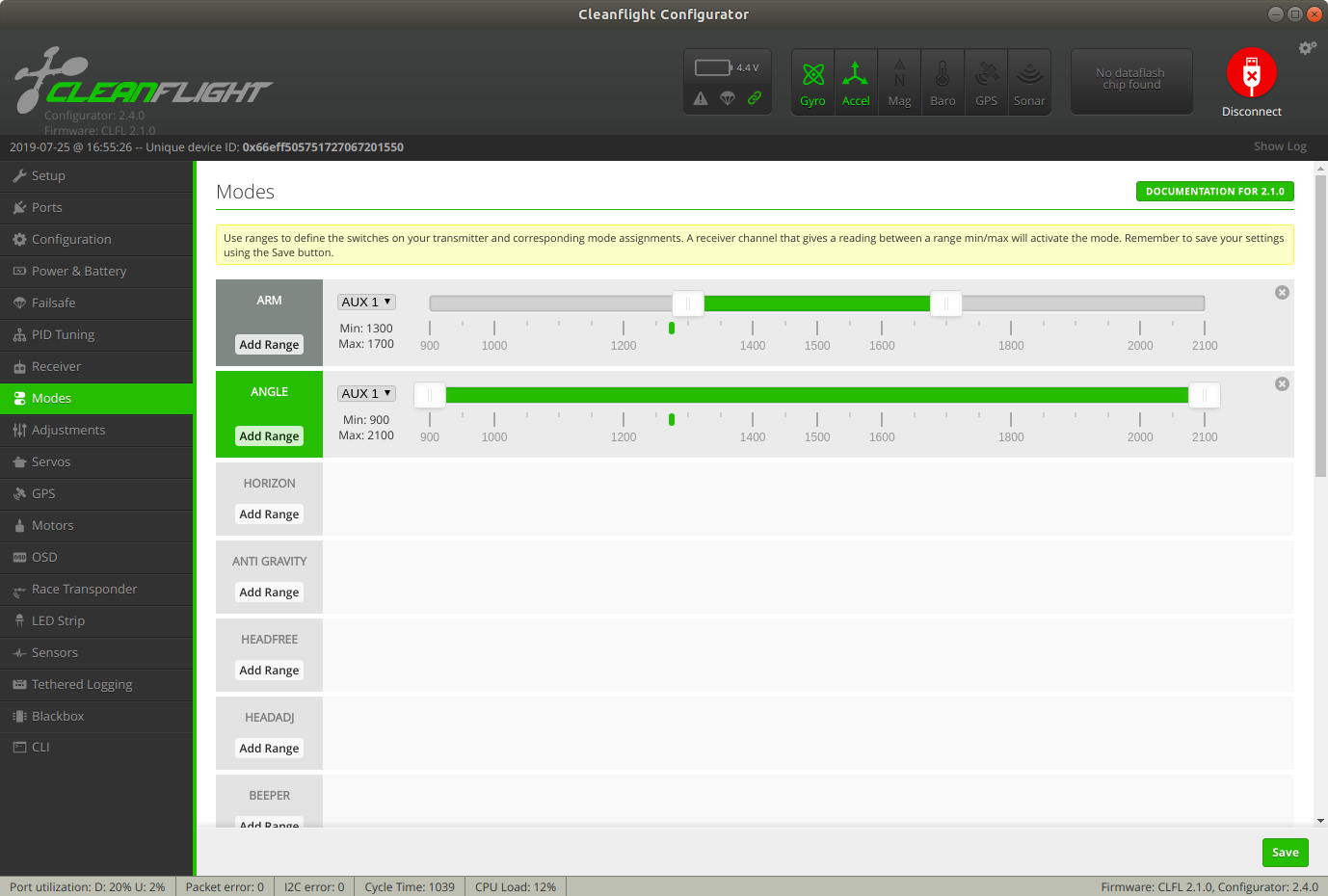
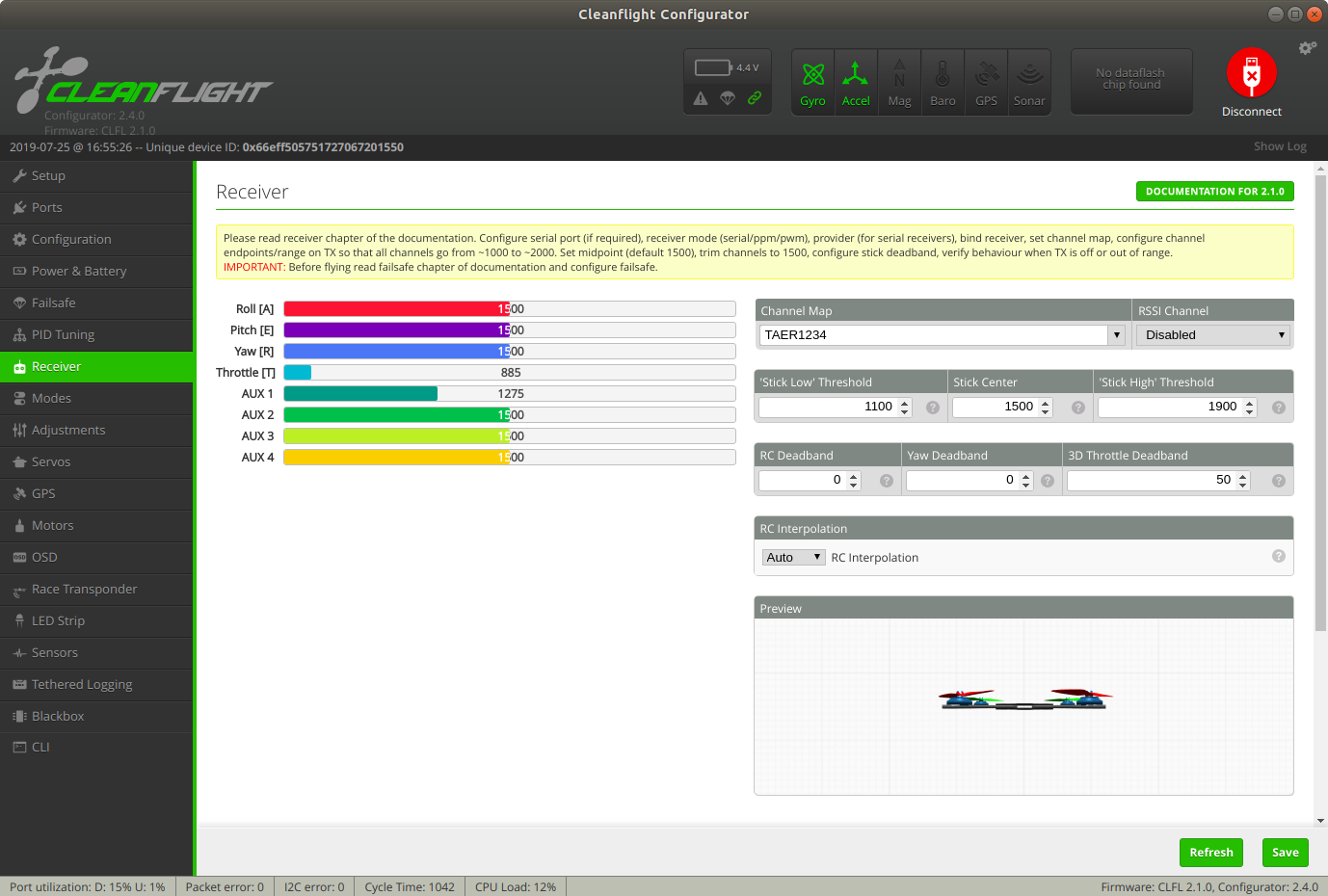
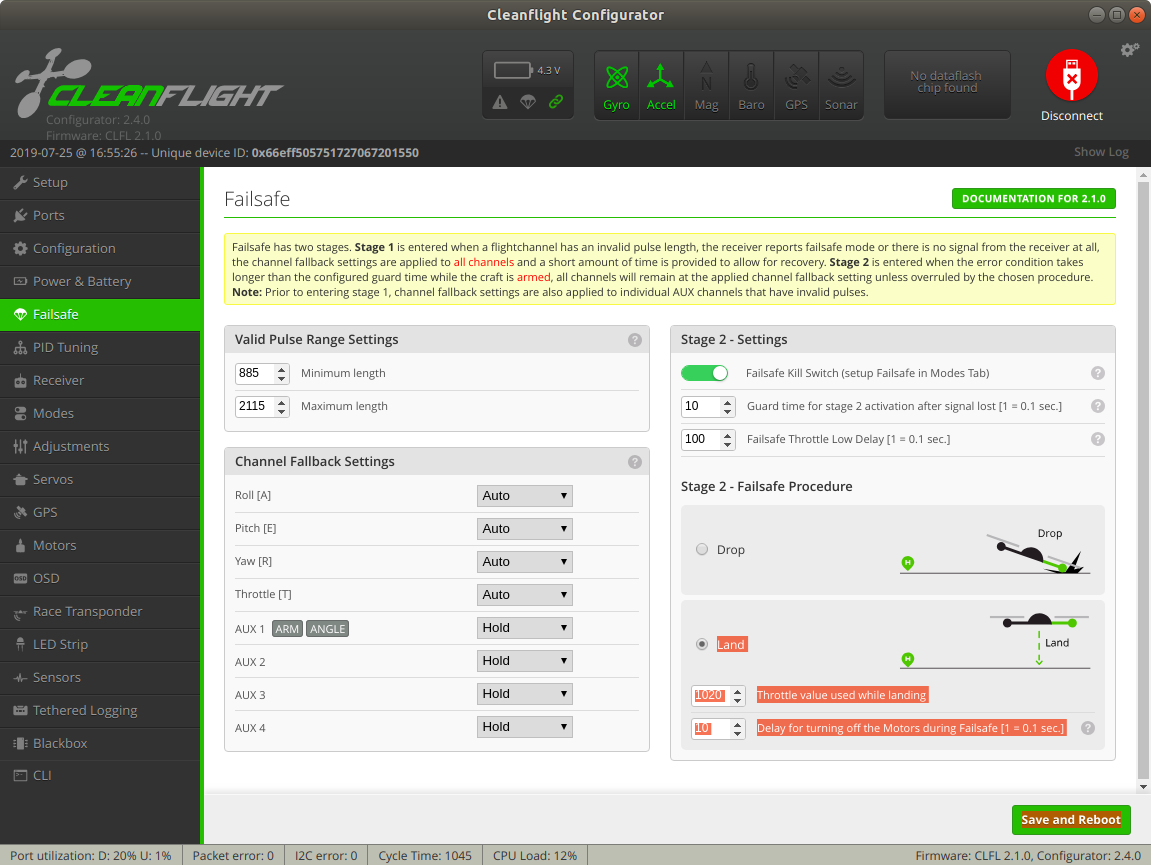
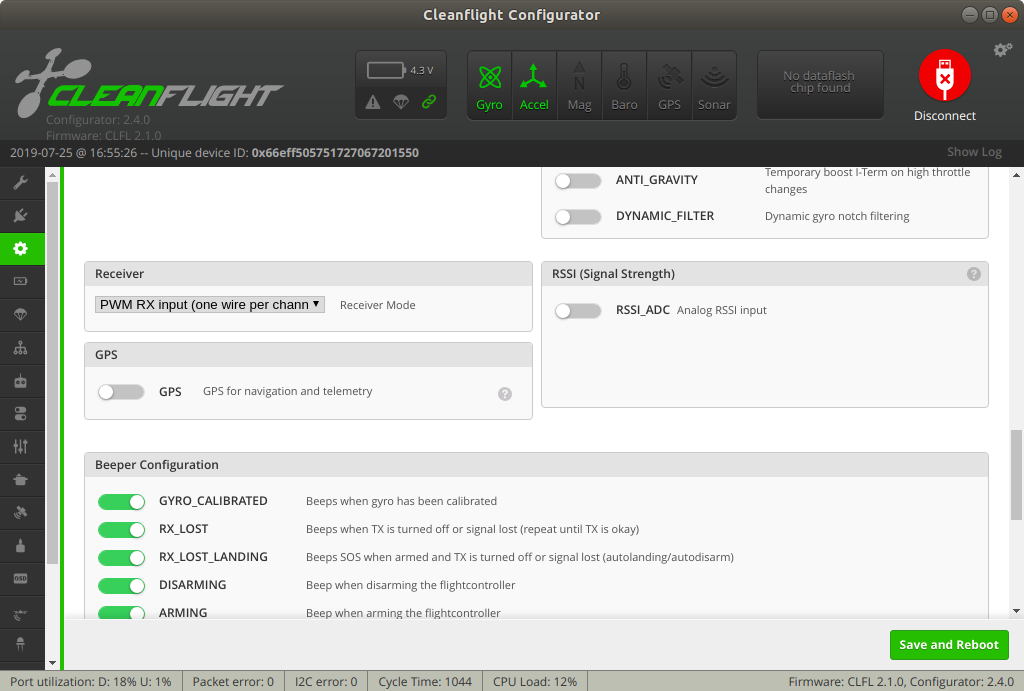


Modify procedure to secure Pi Mount and PDB to frame. Instead of driving standoffs into plastic frame where “bite” of thread is questionable, drill a 1/8” hole through the frame at each corner. From the plastic tube, cut standoffs to size using single sided razor blade. Insert ¾” nylon fastener through Pi mount, through standoff, through corner holes of PDB and through holes drilled in frame. Secure nylon nut to protruding end of fastener and tighten. Bottom view with nylon nuts visible shown here.

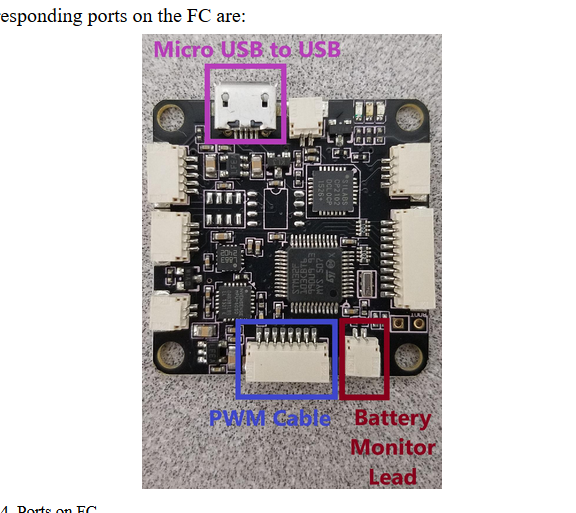
# Configuring Flight Controller for Remote Control

The flight controller must be configured differently for Raspberry Pi control and for radio control. The following screen shots show changes to the configuration. For all other settings follow the instructions on the Build website.

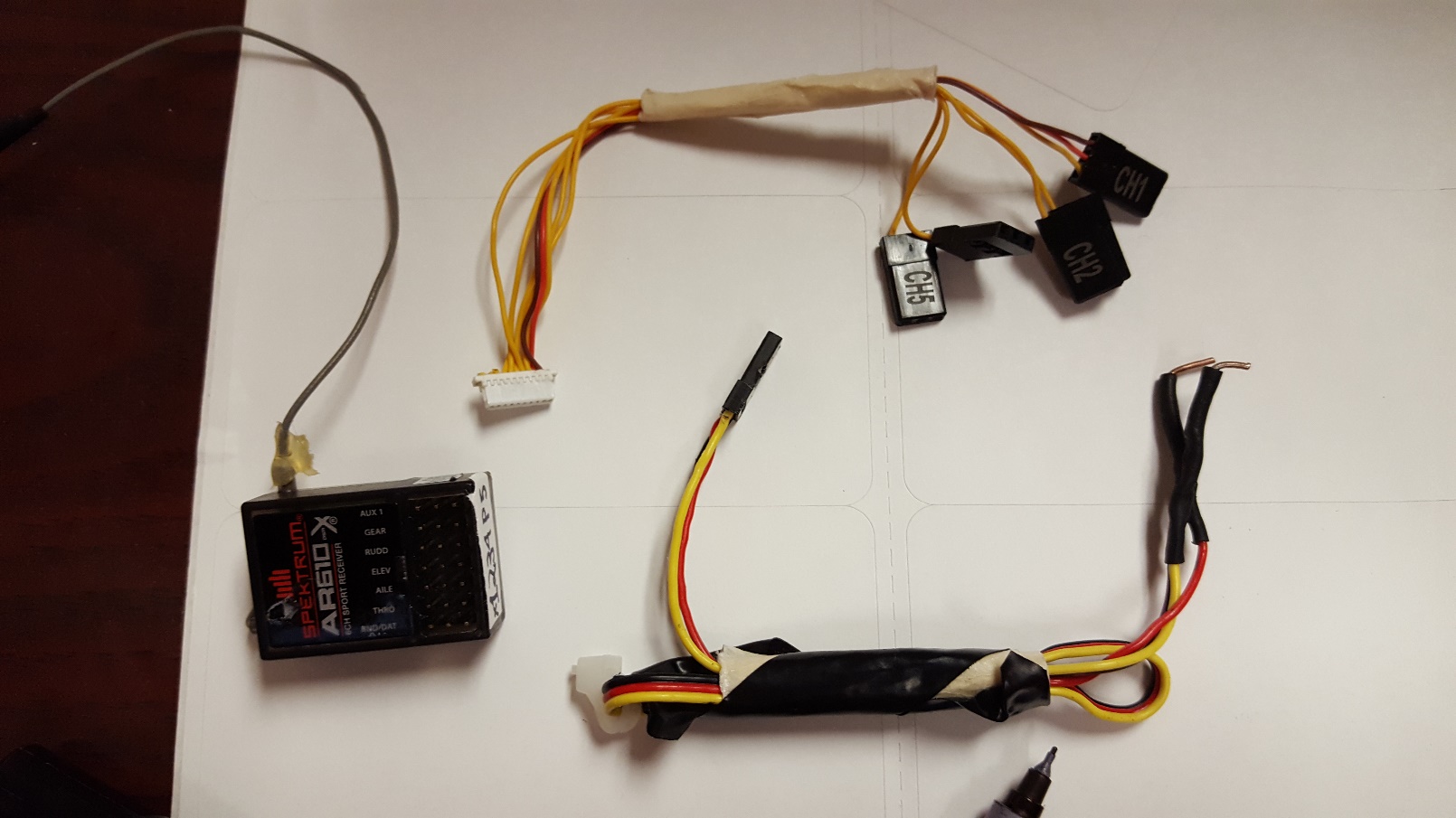
**Remove propellers before attempting**



Determine if the controller is active by watching the colored bars on the receiver panel. When connected, moving the remote levers will move the bars on the panel. Make sure this is working before activating motors, then arm the drone STILL WITH THE PROPELLORS OFF. Check that the motors spin in the correct directions. The motors should also change speed in response to controller commands. This can usually be perceived by the sound the spinning motors make when command levers are manipulated. If controller actions do not produce a noticeable change in motor pitch something is wrong, do not proceed.

Insert the receiver connector contained in each flight controller box into the receiver port of the flight controller. There is only one available port that it will fit.

Receiver port

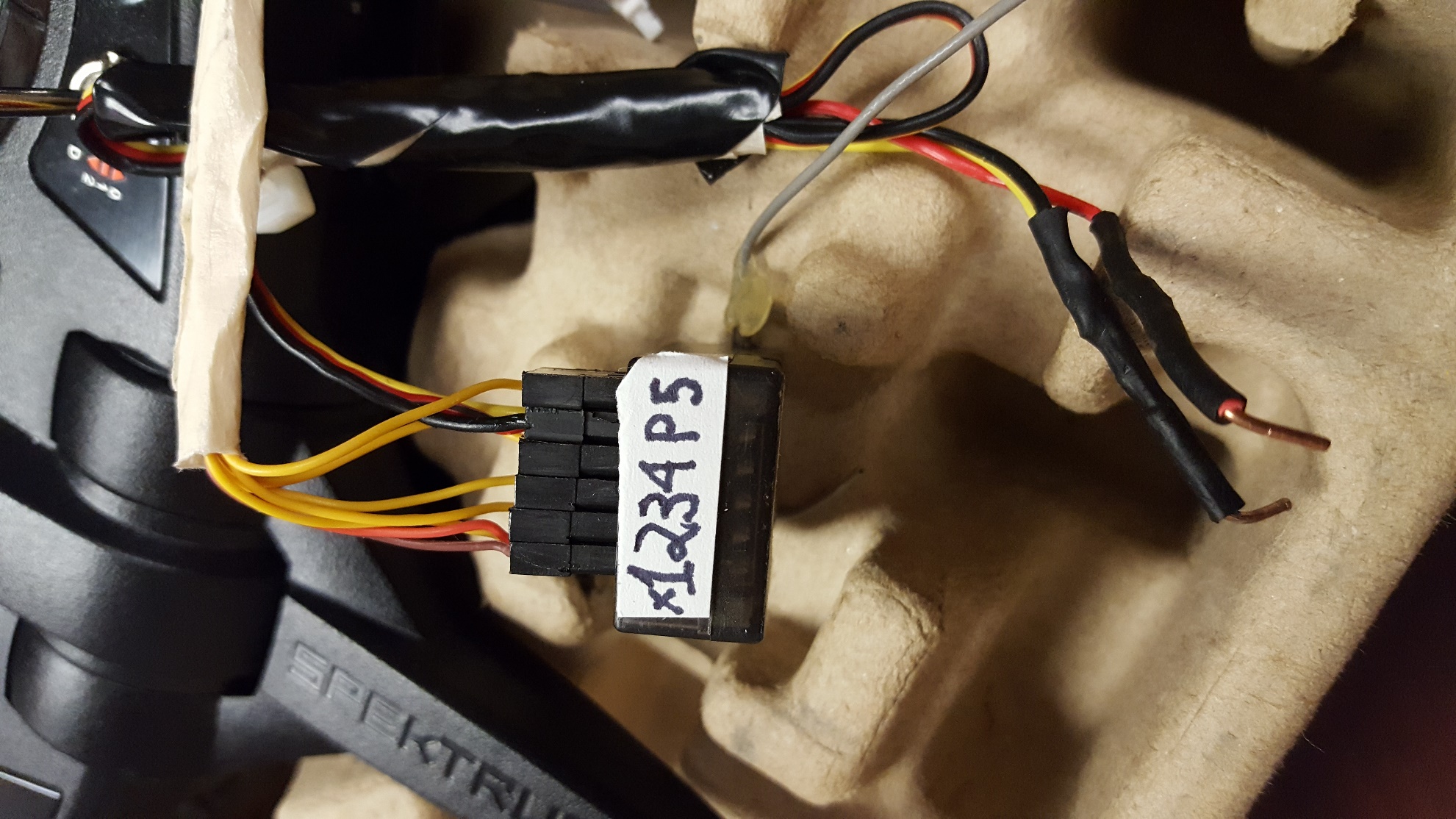


Power

Receiver connector

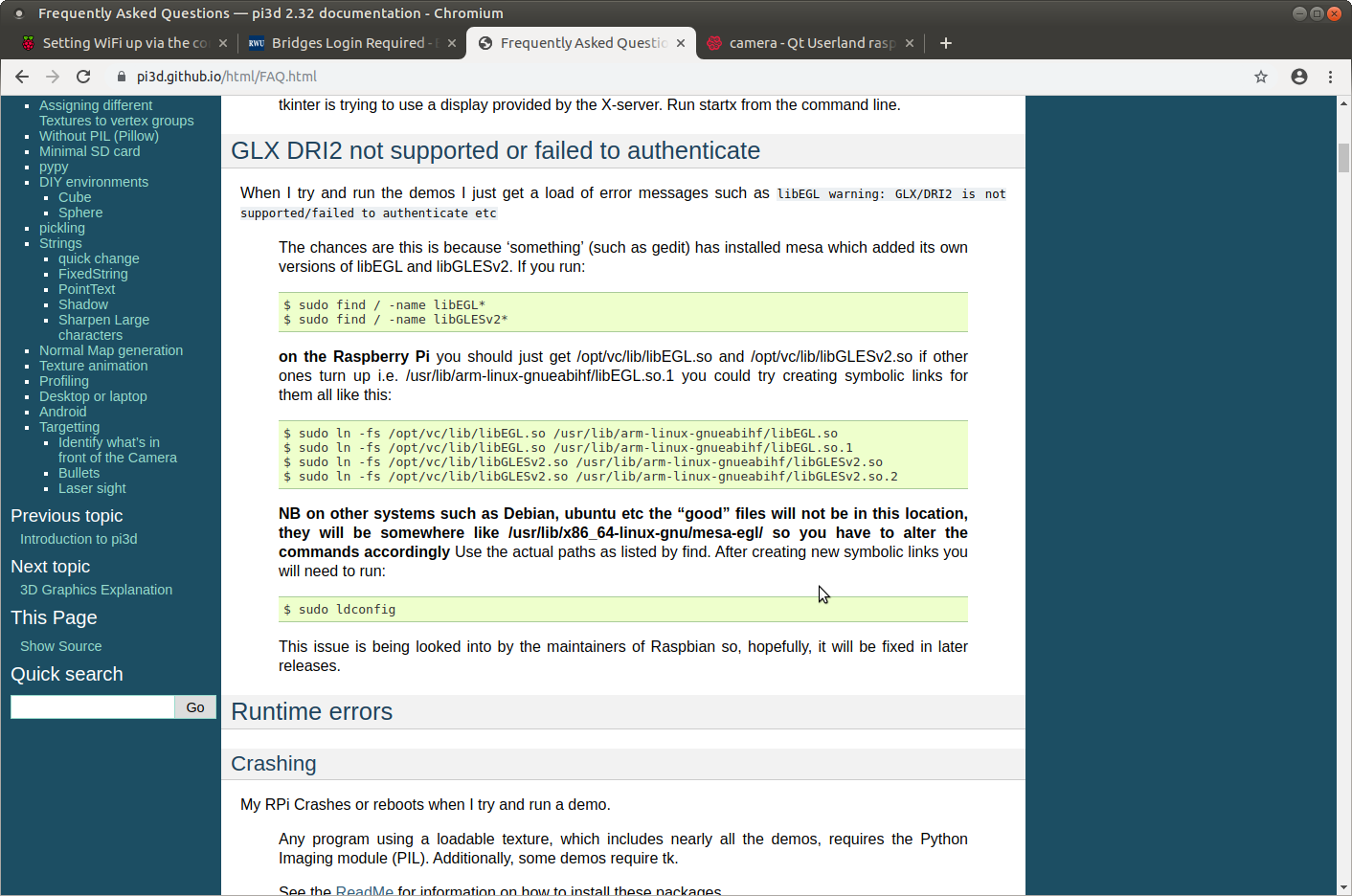
Receiver

Insert tabs labeled CH1-CH5 into the ports indicated. Insert the power connector to the port labeled “P”. Place the bare conductors on the other end of the power connector into available holes on the Pi Hat. Note the connectors are beveled so that they only fit one way. Do not force the connectors into the receiver, although the fit is pretty snug. Insert tabs into the slots as labeled below, note that 5 is closest to the antenna wire. Flight will be impossible if these are inserted in to the wrong tabs, as the pitch lever will control yaw, etc.



# CV2 installation issues

Because the pidrone\_pkg was configured explicitly for drone control under ROS, there are some issues with the OpenCV 2 installation we are encountering because we are doing general image processing in python scripts unrelated to the drone. Two issues are resolved here, the first is a known problem with CV2 on the raspberry pi that comes up when a python program tries to open a window using the cv2.imshow command. This is only relevant when an HDMI monitor is hooked up to the PI. The resolution of this error is described in this website.



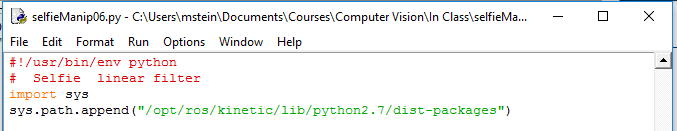
It is likely easier to copy-paste from the website directly: <http://pi3d.github.io/html/FAQ.html>

Mostly just do what it says, although in our installation we also have the directory mesa-egl to include

Execute all four commands again also including this directory, i.e.

sudo ln -fs /opt/vc/lib/libEGL.so /usr/lib/arm-linux-gnueabihf/mesa-egl/libEGL.so

When editing with IDLE directly on the PI with HDMI monitor, it is a minor annoyance that the “F5” command does not work. This is again because our image has CV2 installed as part of ROS and not for general programming. The following work-around explicitly adds our CV2 installation to the path. Place these lines on the top of your python programs and you will be able to Run Module from the IDLE editor



import sys

sys.path.append("/opt/ros/kinetic/lib/python2.7/dist-packages")

# Changing permissions on the pidrone\_pkg

Other students may never need to change the pidrone package to operate the drone, so it is reasonable that this folder is locked. In this course, however, we will be adding our own code to the pidrone package for computer vision. Execute the following command in a terminal to change ownership of the folder and all its subfolders to yourself. Note that the “-R” means “recursively” to change all subdirectories. Also note that you will need to type the superuser password in response to the sudo command.

cd ~

cd ws/src

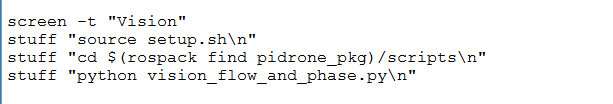
sudo chown –R duckiesky pidrone\_pkg/

# Move files to script folder and edit pi.screenrc to run single\_map\_localization.py

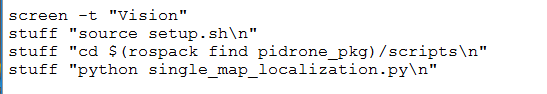
Copy single\_map\_localization.py to the scripts folder. You will also need single\_map\_helper.py and your map file renamed to map.jpg the scripts directory. Rename the current map.jpg to something line BrownMap.jpg and take a look at it. That map is about 10 feet by 5 feet wide, so their drones only see a portion of the map at a time. Our software is a simplified version of theirs that presumes that there is one map and they camera can see a pretty big chunk of the map at any time. In addition, single\_map\_localization\_monitor.py and single\_map\_helper\_monitor.py and drawMatches.py can be copied to the scripts folder and run directly from the command prompt. These are very useful if you have the Pi attached to a monitor, as it will drawMatches in real time between the map and the camera image it is currently seeing. Be sure to edit the MAP\_WIDTH and MAP\_HEIGHT values to match your values. You may also want to modify NUM\_FEATURES to get acceptable results.

The screen program opens up 9 windows that you have been switching between. It executes scripts in the scripts folder where the vision navigation program in window 4. Edit pi.screenrc to change the name of the program it executes to single\_map\_localization.py .

Before edit:



After edit



Save this file and run screen –c pi.screenrc. Note that if you cannot save the file because of permissions you have not completed the step above this one.