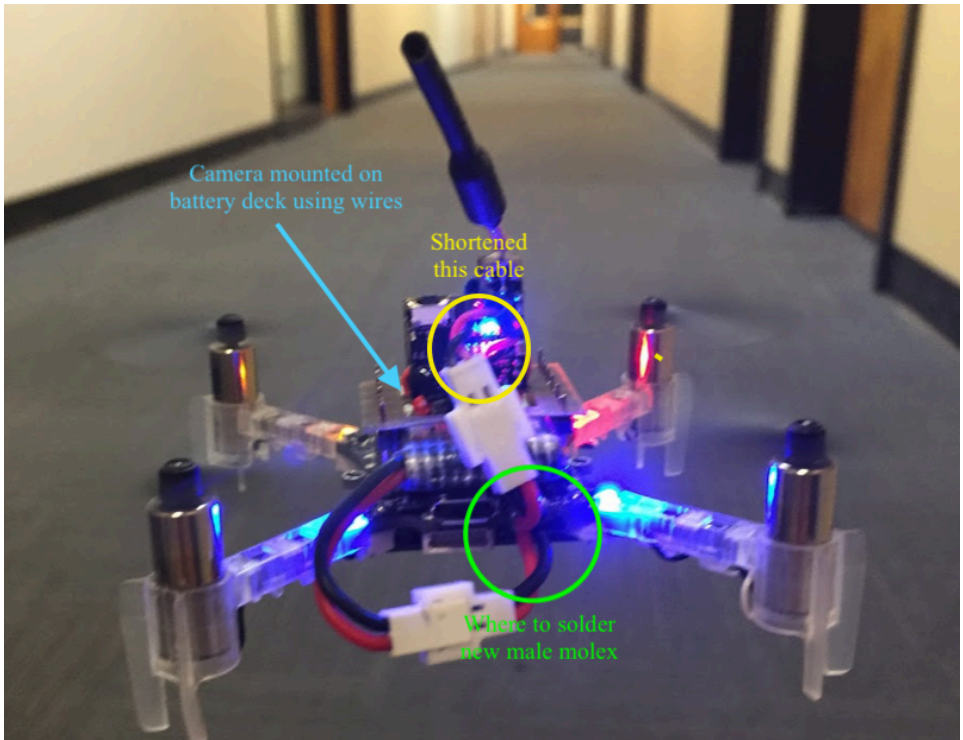
1. **Setting Up Crazyflies from the box:**
    a. Follow the full tutorial here:
    b. https://www.bitcraze.io/getting-started-with-the-crazyflie-2-0/
        i. Skip the Bitcraze virtual machine stuff, but a **python 3 Virtual Environment is highly recommended**
        ii. To install client on Mac or Linux, clone the following repositories
            1. https://github.com/bitcraze/crazyflie-lib-python
            2. Then “pip3 install -e .” in the top level of cflib
            3. Follow the “Setting udev permissions” for Linux section on the Readme of this repository (needed to use the usb radio without being root)
            4. https://github.com/bitcraze/crazyflie-clients-python/blob/master/README.md
            5. Then “ pip3 install -e . “ in the top level of crazyflie-clients-python
            6. Run the client now by running “cfclient” in terminal
2. **CFClient Configuration**
    a. Ensure that you can fly it by running the cfclient and controlling via joystick
    b. You need at least 2 USB ports for this, and 3 once we add the camera in
        i. Port 1: CF usb radio
        ii. Port 2: Gamepad joystick usb dongle
    c. To make sure the joystick is working, check that your gamepad device is showing up in Input Device -> Device. You can then configure the button mapping in Input Device -> Configure Device Mapping, and then select Input device, and click configure, and then at the bottom click load, and then select the profile name “PS3_Mode2” and click save.
    d. Confirm that you see the joystick axes responding to the inputs, and then exit the device mapping screen.
    e. Once the crazyflie is on, “Scan” and then select the interface (“Select an interface” dropdown) corresponding to the usb radio you are using. Then click “Connect”, and the flight data should start to be reflected in the right half of the screen.
    f. If you can control your crazyflie properly, move on to next step
3. **Setting Up Optical Flow**
    a. Attach the optical flow deck as shown in the video here:
        i. https://www.bitcraze.io/flow-deck/
    b. First update firmware if you haven’t already
        i. https://www.bitcraze.io/getting-started-with-the-crazyflie-2-0/#latest-fw
    c. Open up the crazyflie client, set “Assist mode” to “Hover”
    d. Open up the Input Device mapping again (load and set the PS3_Mode2 if necessary, see above), and now also enable the “Assisted control” button by pressing Detect and then pressing the desired button for takeoff (use RB, LB, or RR, LR; we used LR)
4. **Setting Up Camera**

a. First make sure the Tinywhoop camera can power on by connecting it to a 1S battery
b. Now connect the USB adaptor dongle to computer, and attach the yellow female plug to the yellow male plug of the Camera Receiver.
    i. Note: You might see overexposed camera images here -- the fix for this is to put a resistive divider before or after this connection.
    ii. This is what the potentiometer in the parts list is for (tunable resistor), but it is not essential while starting out.
c. Then power on the Camera Receiver by plugging the given power cable into the DC 12V jack. It should turn on and display a letter and a number, which specifies the current channel it is listening to.
d. You can do a frequency sweep to find the frequency that the Camera is on by holding the “CH” button for a few seconds, or you can manually just keep clicking the “CH” button until the image becomes clear.
    i. This requires you have an image of the camera, which you can do very easily by running an OpenCV script that uses VideoCapture
    ii. Or skip to the ROS section and run our “joycf.launch” script which should open up an image
        1. If it throws an scn == 3 || scn == 4 error, see the FAQ for a fix
e. Connecting it to the crazyflie
    i. We attached the camera to one of the mounting decks on the crazyflie (that holds the battery in place) using standard wires and adhesives.
    ii. To power on the camera, we soldered a new male molex (there should be extras, but if not, you can purchase one like this: http://graysonhobby.com/molex-2-0-cable-set.html?utm_source=google_shopping) onto the soldering pads where the battery molex is connected. We also shortened the length of the molex wire on the camera so that it wouldn’t collide with the propellers.
    iii. A picture can be seen below:



f. Connect the crazyflie to the battery and now both the crazyflie and the camera turn on!
    i. Note that the camera does decrease the battery life of the crazyflie non-trivially, both through weight and power usage.
g. Make sure the camera still works fine
    i. If the camera starts smoking or anything out of the ordinary happens, unplug the battery immediately to prevent damage to the crazyflie
h. Now the camera is setup to work with ROS!

5. ROS Setup (Kinetic)

a. http://wiki.ros.org/kinetic/Installation/Ubuntu
    i. Follow this tutorial for Ubuntu, you can install ros-kinetic-desktop-full
b. Once installation is complete, configure your ROS environment
    i. http://wiki.ros.org/ROS/Tutorials/InstallingandConfiguringROSEnvironment
c. In catkin_ws/src/ clone the following repository
    i. https://github.com/gkahn13/crazyflie
d. In the top level ("catkin_ws"), run catkin_make
    i. This builds the crazyflie package
e. Run "roslaunch crazyflie joycf.launch use_joy:=True"
    i. This should launch the crazyflie node, the joystick node, the control node, and the camera node. See the Camera section for troubleshooting if you see no image or you have an error.
    ii. Test that the crazyflie is responding to joystick commands
        1. Back & Rear Buttons
            a. LR (Left Rear) -> Takeoff
            b. RR (Right Rear) -> Land
        2. D pad buttons
            a. Up -> dz +
            b. Down -> dz -
        3. Right buttons
            a. B -> E-stop
        4. Left joystick
            a. left -> yaw +
            b. Right -> yaw -
        5. Right joystick
            a. Up -> dx + (velocity in x direction)
            b. Down -> dx - (velocity in x direction)
            c. Left -> dy + (velocity in y direction)
            d. Right -> dy + (velocity in y direction)
    iii. Note: these are not necessarily the same as what was configured in the CF Client joystick mapping
f. When you run the crazyflie with GCG side-by-side, you should set "use_joy:=False" so that the crazyflie doesn't get conflicting joystick messages

MISC FAQ:

- Is there a way to run joycf.launch such that the crazyflie doesn't respond to commands but still sends data?
    - Yes! roslaunch crazyflie joycf.launch **data_only:=True**
- The crazyflie is jittering back and forth while running GCG / other environments with the joystick?
    - This is usually due to getting joystick commands from two different sources. One of those sources is most likely going to be the crazyflie itself, so you can fix this by re-running joycf.launch with "use_joy:=False"
- What do all the lights mean on the crazyflie?
    - https://wiki.bitcraze.io/projects:crazyflie:userguide:index#led_indications
- Where can I read more about the crazyflie?
    - https://wiki.bitcraze.io/index
        - Bitcraze wiki documentation
- ImportError: no module named em?
    - pip uninstall em
    - pip install empy
    - pip install pyaml (unrelated to the error but for good measure)
- Error: cannot import ros messages?
    - source ~/catkin_ws/devel/setup.bash
    - This adds all the ros tools to your current bash environment
- OpenCV assertion failed scn == 3 || scn == 4 Error:
    - Ensure that camera usb adaptor is plugged in, and is connected, and the receiver is turned on
    - open up catkin_ws/src/crazyflie/src/Camera.py and change the number in the constructor of VideoCapture from 1 to 0 or 0 to 1, and try again.
    - Sometimes on linux, the usb port numbers get mixed up.
    - Rerun.
- Noisy image showing up instead of camera image?
    - Cause 1. Camera is not turned on, check that first
    - Cause 2. Camera is turned on but the receiver is at the wrong frequency
        - Fix by rescanning channels or manually changing channels until you see a good image
        - Press the "ch" button on the suggested receiver to do this
- USB permissions error?
    - See the "Setting udev permissions" for Linux section the crazyflie-lib-python repository
- Crazyflie is super wobbly in flight and usually crashes because of this?
    - Most likely this is due to the Optical Flow deck, which is not super reliable across all types of floors

    - One solution we used was to shine a flashlight under the crazyflie, especially in dark environments or where the floor pattern is not very distinctive, but this is certainly not guaranteed to work. Optical flow needs distinct features on the ground for it to be stable during flight.
- Crazyflie has a very laggy response?
    - Usually this is due to some unknown ROS lag on the computer you are using
    - Either restart the computer or try closing out of everything and unplugging and replugging the CF usb dongle.
- Crazyflie spins out of control immediately on takeoff?
    - First thing to check here is the CF battery. Usually once a crazyflie has gotten to a low enough battery, it will kind of ungracefully fall to the ground
    - There could also be too much weight on the crazyflie
- Camera doesn't power on or the antenna broke?
    - Get a new camera :)
- Overexposed image?
    - Put a resistive divider (for example, a potentiometer) in the middle of the yellow plug that comes out of the video receiver.