Setting up DANNCE Rig

Introduction
Room Requirements
Assembling Rig
Building Computer
Creating Splitter PCB
Installing Software
    Permission denied on /dev/ttyACM0:
    Name cameras:
STEPS FOR SETTING UP RIG
    Updated 2024-06-05
FAQ
Errors
    Launching Pylon Viewer: VCruntime140_1.dll not found.
    "Error while writing node: 'BslSerialBaudRate' not found" –
    Arduino not triggering:

GOAL: write a tutorial for building a new DANNCE rig from scratch

ESTIMATED TIME TO BUILD: 1 week (excl. shipping)

ESTIMATED COST: bill of materials spreadsheet

TODO for this documentation:

- **Pictures (inspiration from instructables)**
- **Keep in mind the audience**
- **Make sure to document versions of things**
- **This procedure works with XXX camera and YYY version of the Operating system.**
- **Documentation is:**
    - **About process**
    - **About component**
    - **Certifying baseline working configuration**

Introduction

This manual explains how to build the 6-camera setup used for recording rats in the Olveczky Lab. Other labs have different systems for setting up cameras including Kyle Serverson's setup for mice at the Wang Lab, which is much smaller and can fit on a desk. Or a the setup in the T. Dunn Lab which uses a single camera and four mirrors.

If you don't know what you're doing and you have a room you can use to set up the rig, we recommend you follow the instructions in this guide to build the **6-camera rig**. Other rigs will work, but may not be officially supported.

Room Requirements

These are approximate dimensions to build the 6-camera rig in this tutorial. Other rig setups may be more appropriate for smaller spaces:

- Approx 10' x 10' for arena; more space required for a operator desk
- Each camera is ~40" from the center of the arena (and from the neighboring two cameras)
- Table is 28" tall
- Cameras are mounted 48" off the ground
- Light fixtures about 50" from the center of the arena
- Light figure about 70" off the ground (on tripod); angled towards the center of the arena

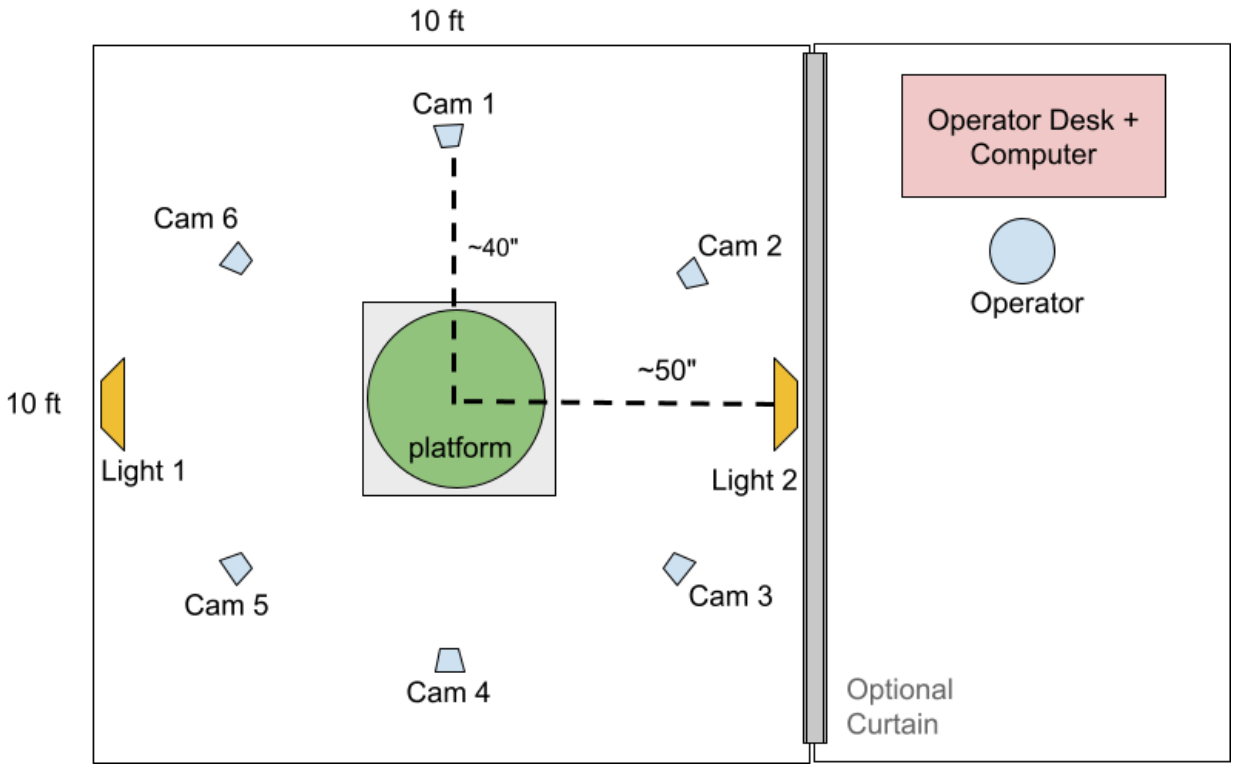


Figure: Example layout of room

Building Computer

See a computer-building tutorial. There are many good ones in text or video form. E.g. [this one from Intel](#).

You can alternatively use a pre-assembled computer.

Assembling Rig

Using colored tape/heat shrink tubing to mark cameras and cables is **optional, but highly recommended** in order debug potential cable or USB port issues.

Preparing power IO cables

1. The basler Power/IO cables have one end with 6 pins and another end with loose wires; we want to do the following steps on the loose wires end.

2. Cut away shielding from the tips of the wires until the start of the rubber cable, using wire cutters. Be careful not to clip any of the wires

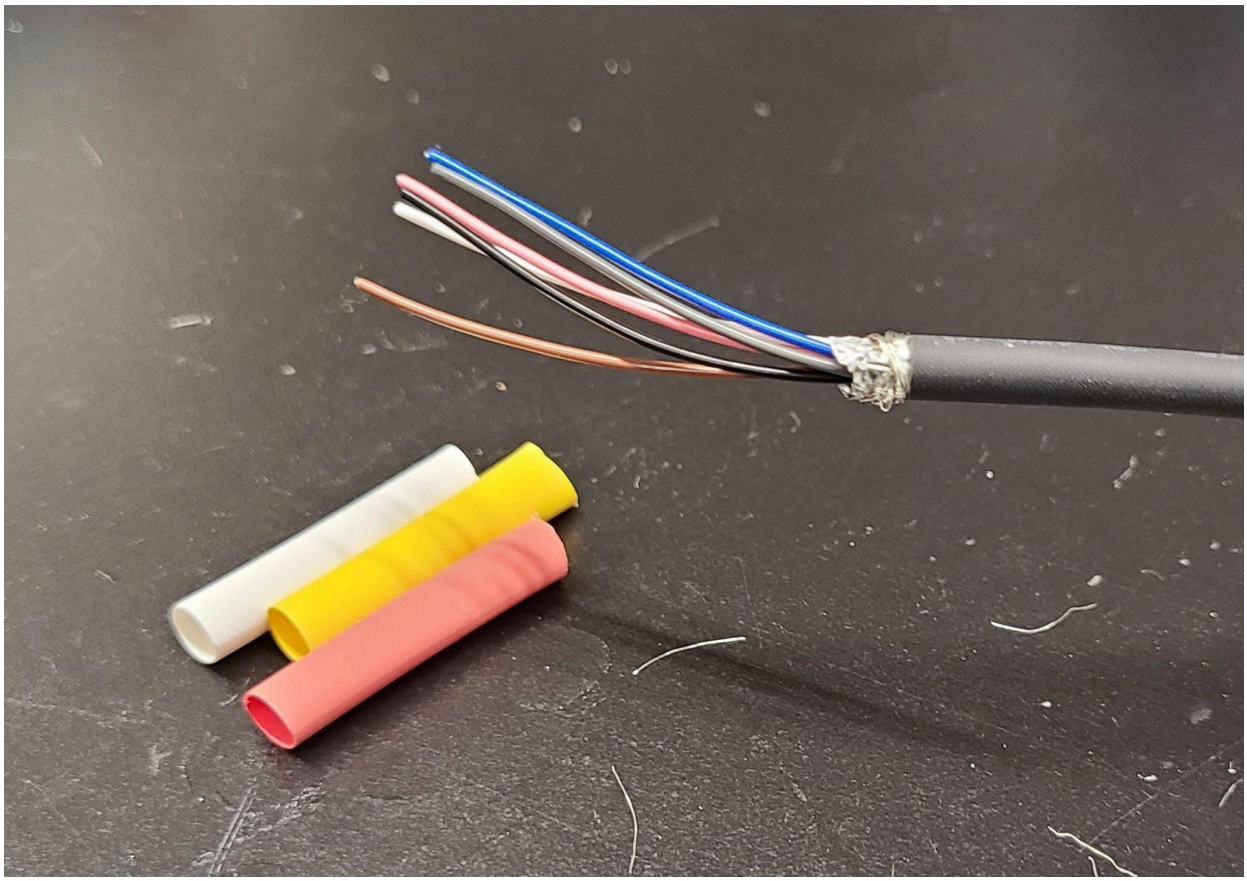


Figure: Power/IO cable with shielding cut away

3. Fold back 4 of the 6 wires: specifically the **brown, white, blue, and gray wires**. This should leave the **pink** and **black** wires unfolded
4. Cut a 1.5-inch length of heat shrink tubing (¼ inch diameter). Do this for each of the 6 colors of heat-shrink tubing (e.g. we used white, yellow, red, green, blue, and black).
5. Slide one piece of heat shrink-tubing over the lose wires so that the only the pink and black wires are sticking out, and the other 4 wires are folded back.

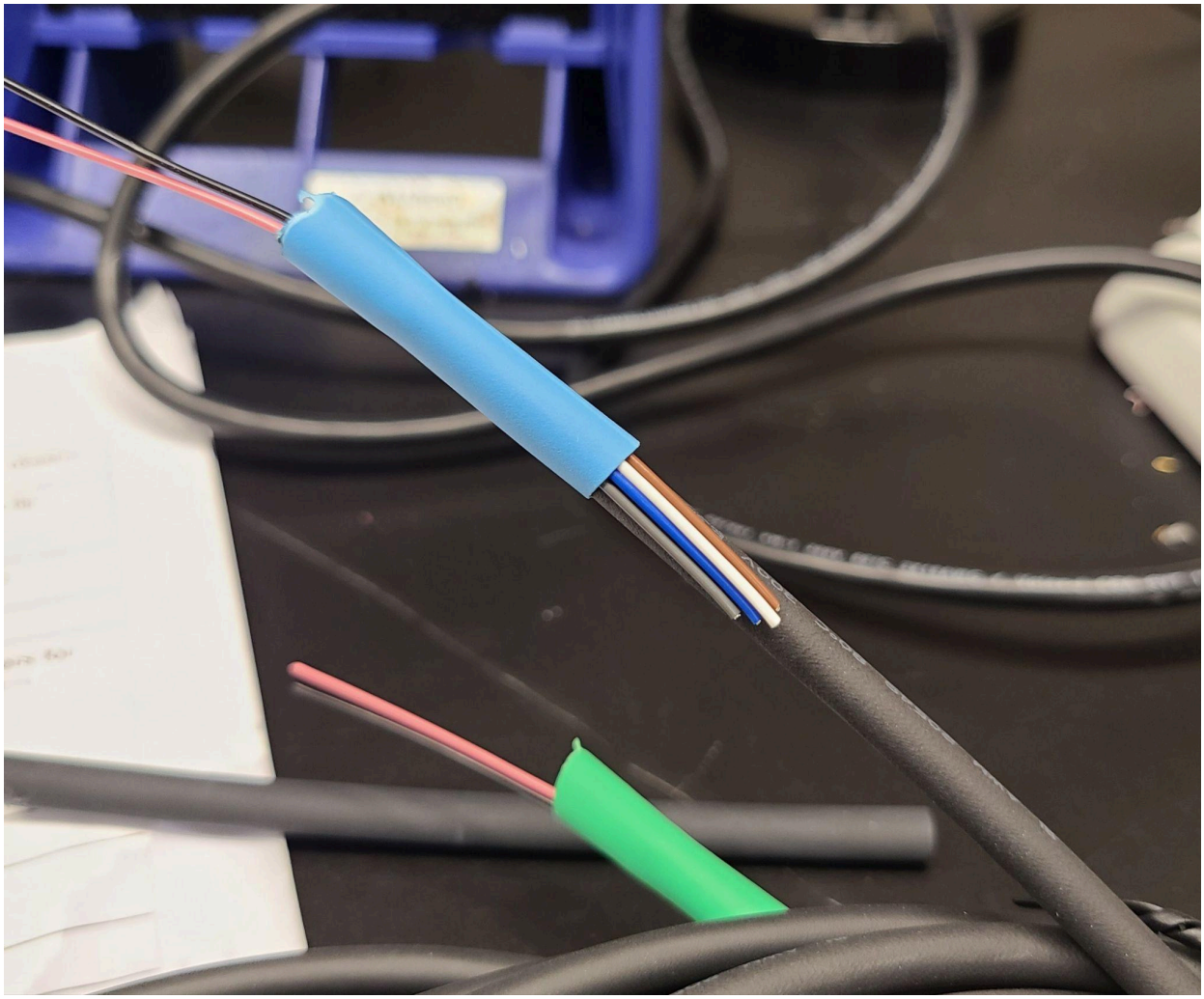


Figure: Heat-shrink tubing on one cable (blue) showing 4 wires folded back.

6. Using a hot-air station, **carefully** blow hot air onto the heat-shrink tubing. This will cause it to shrink and hold fast on the end of the cable. This should take about 5-15 seconds per piece of tubing. **Do not blow hot air at your hand directly as this can cause burns. Do not blow hot air at the wire for too long as this might start a fire.**
7. Each of the 6 Power/IO cables should now have heat-shrink tubing on the loose end, and exactly 2 wires should be sticking out of each cable: pink and black wires.
8. For each Power/IO cable, use a strip of colored tape matching the color of the heat-shrink tubing. Wrap a small strip of tape around the 6-pin connector end of the cable. Alternatively you can use larger diameter colored heat shrink tubing
9. Each cable should now have a colored piece of tape on one end and a colored heat-shrink tubing on the other end.
10. Using a wire stripper, strip about 0.25 inches of coating from the pink and black wires for each cable. These cables should be AWG 24.

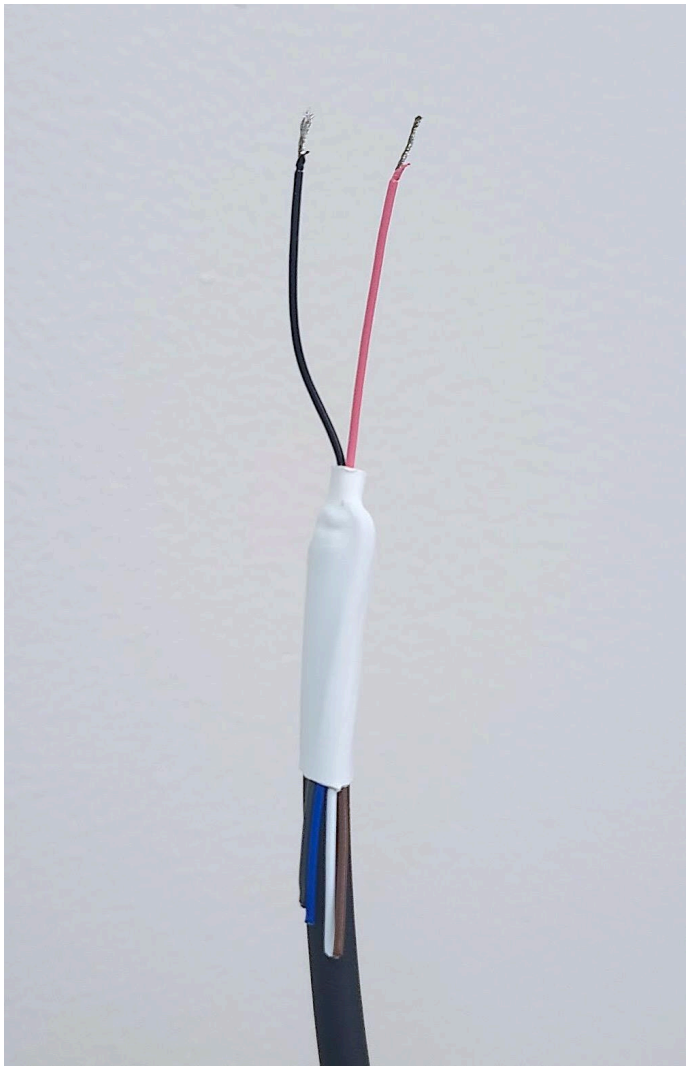
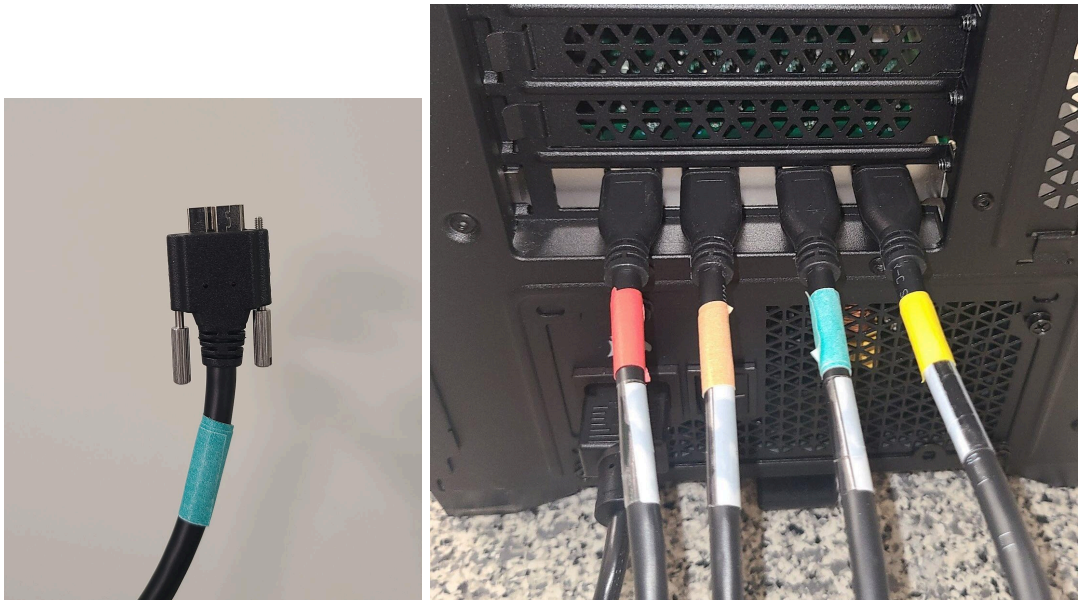


Figure: Power/IO cable with heat-shrink tubing & stripped wires

Preparing USB cables

1. There should be 6 identical USB cables. One end will have a USB-A connector (typical USB shape), and the other end should have USB 3.0 Micro-B connector (long and skinny connector with two segments) and may have 2x screws.
2. Use the same set of colors used above for the Power IO cables. For each USB cable, cut 2x piece of tape of a single color, and wrap one around the cable near each end. You will end up with 6x cables with each end marked with a different color.



Figures: Both ends of camera USB cables with colored tape

Preparing Cameras

1. Using the same set of 6 colors, cut a small piece of tape (about 0.5 inch long) and stick it to the camera's body, making sure not to obscure the serial number. Use a different color for each camera. You can alternatively put the tape on the tripod attached to each camera.
2. Attach the lens to each camera, and screw on snugly.
3. Attach the 4-hole tripod mounting plate to each camera and connect to each camera with 3 screws.
4. For each camera, loosen the aperture/F-stop screw and spin the aperture ring to the widest aperture; denoted by the lowest number of the ring. E.g. going from *f/4.2* to *f/2*. Tighten the aperture screw in this position.

❓ **Why use a large aperture? See FAQ Section at bottom of tutorial**

You should now have 6x USB cables marked with colors on both end, 6x Power IO cables marked with colors on both ends, and 6x cameras marked with colors. When it is time to attach the cables to the cameras, use the same color for each set of camera, power IO cable, and USB cable.

Setting up the room:

1. Place table in the center of the room
2. Mount each camera on a tripod (6x)

3. Attach lenses to cameras and screw in with your hands. It should be snug, but do not use excessive force. (6x)
4. Arrange tripods carrying cameras around the center table in a hexagon
5. Connect the powerIO cable to each camera, taking care to match the cable color with the camera color. Screw in the power IO cable until it is snug.
6. Route the cables cleanly and create a bundle of cables with the loose wire ends of the Power IO cables and zip/cable tie them together. We will attach them to the splitter board later.
7. Connect the USB cable to each camera, taking care to match the cable color with the camera color. Screw in the USB cable until it is snug. You may use a small screwdriver for this process as it can be difficult to grip the cable screw, but be careful not to over-tighten as you can deform the plastic around the USB micro-B connector. Rule of thumb is stop tightening when the screws are visibly in all the way.
8. Route the cables cleanly and create a bundle of cables with the USB-A end of the USB cables and use a zip/cable tie to connect them together.
9. Plug the USB cables into the computer. First use the 4x slots on the USB3 PCIe card, then use 2x USB 3 slots on the motherboard. **You must use USB 3 ports for each of these cables** (usually ports with blue plastic instead of white/black).
10. Plug Arduino cable (ether USB-A to micro-USB for newer Arduinos/Teensy boards or USB-A to USB-B 2.0 for the Arduino Uno/Mega) into the motherboard on the PC. You can use USB 2.0 or 3.0 ports for this.
11. Plug the arduino cable into the arduino.
12. Using two strands of hookup wire, connect the following pins on the arduino to the following pins on the splitter board:
    - Black cable: GND pin on the Arduino to a black pin on the splitter board
    - Red cable: Digital pin #9 on the Arduino to a red pin on the splitter board.
13. Connect each of the power IO loose wires to the splitter board, such that the pink wires are connected to the red pin on the splitter board and the black wire is connected to the black pin on the splitter board.
14. Set up the studio lights and plug into a power source. Each light tripod should be positioned between two camera tripods; and the two light tripods should be on opposite sides of the arena (e.g. between cameras 1 & 2 and cameras 4 & 5). The lights should be raised to about 6 ft high. The lights should be angled towards the center of the platform.

Focus cameras:

1. Place a couple of objects with sharp lines in the at various points of the arena (e.g. center, edges, etc.). These will be used to focus the cameras. Good objects include: calibration target(s), pens, screw drivers, legos, etc.

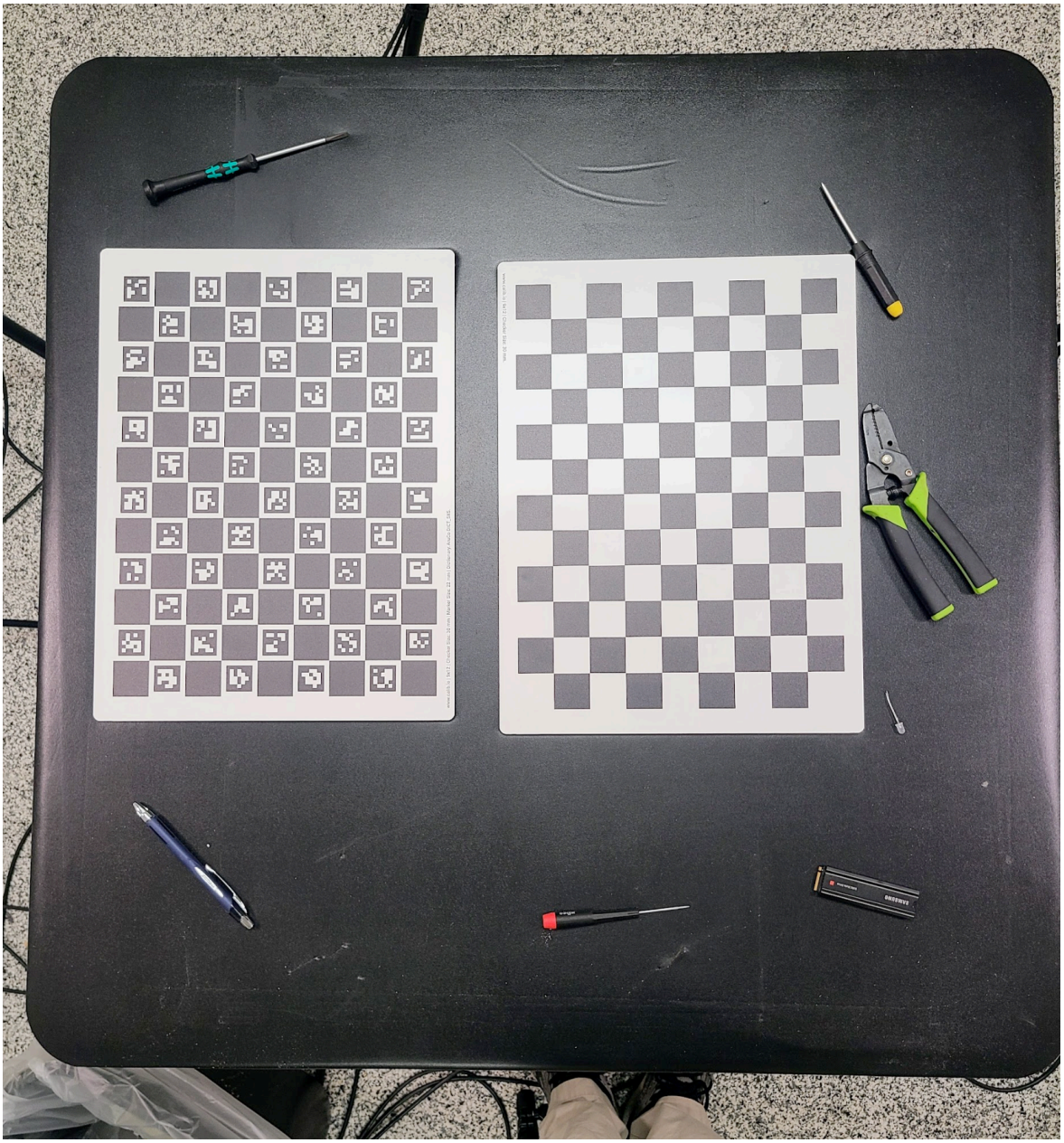


Figure: objects on the table for focusing cameras

2. Using pylon viewer go through the following process for each camera:
3. Adjust the camera's position so that the picture is not tilted, and the arena is centered horizontally in the view.
    a. The bottom/closest edge of the arena platform should be just above the bottom of camera view and the farthest edge of the arena should be about ½ to ⅔ of the way up the camera view. This is so you will be able to record a subject of about 8 inches tall (e.g. fully reaching rat), in the far edge of the arena.
    b. You may have to move the cameras towards/away from the arena or adjust the camera tripod heights to achieve this.

4. Loosen the lens focus screw (you may need a small flat-head screwdriver)
5. Adjust the focus ring until objects in the far region of the arena are in focus. Now, adjust focus until objects in the near region of the arena are in focus. Finally, move the focus ring about half-way between those two points. At this location, objects in the center of the arena should be completely in focus and objects at the near- and far-edges should be mostly in focus.
6. Re-tighten the focus screw with your fingers. Do not tighten excessively with a screw driver.

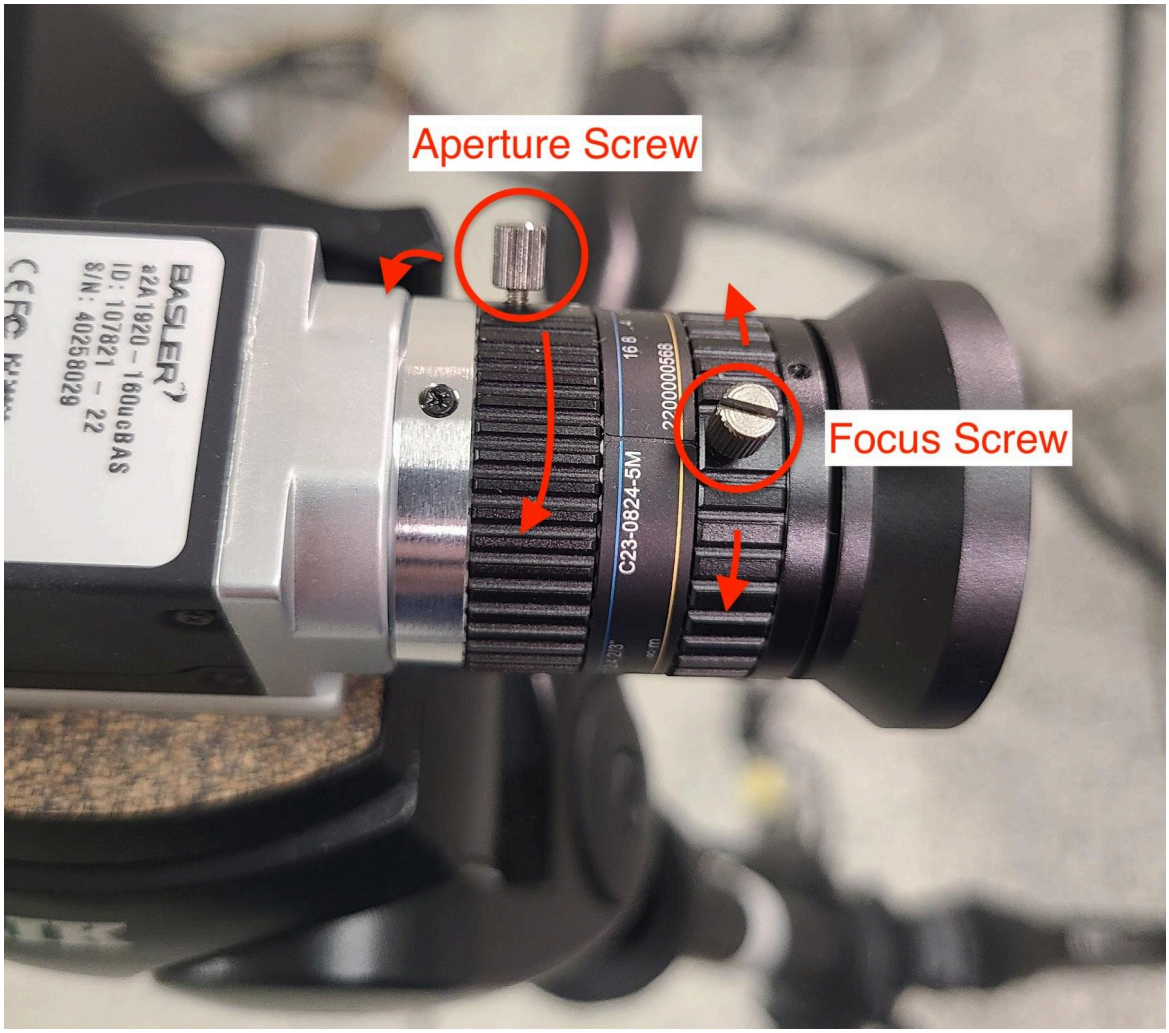


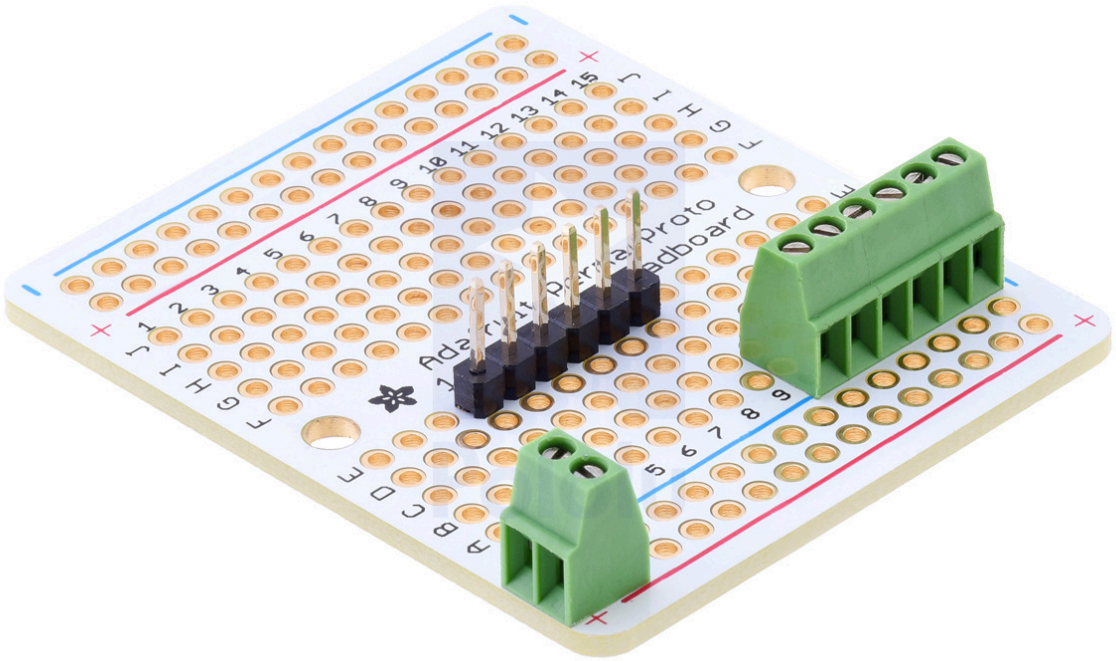
Figure: Aperture and focus screws on camera lens.

Creating Splitter PCB

Create a PCB for connecting camera trigger pins:

- Solder a prototype breadboard according to the following diagrams

    - You can also use a solderless broadband but it will be less durable
- You can use screw or quick-release terminal blocks (see below) to attach the camera GPIO wires without solder:



www.pololu.com

Figure: example 6-terminal block with flat-head screws. (I use 2x 6-terminal blocks)

⚠️**WARNING: Note the unfortunate color scheme!** On the basler cables, the PINK wire is ground, while the BLACK wire is General Purpose I/O (GPIO). On our circuit; the black wire is ground, and the red wire is data (as is convention). This means you're connecting the pink wire on the Power/IO cable to the black wire on the arduino, and black wire on the Power/IO cable to the red wire on from arduino.

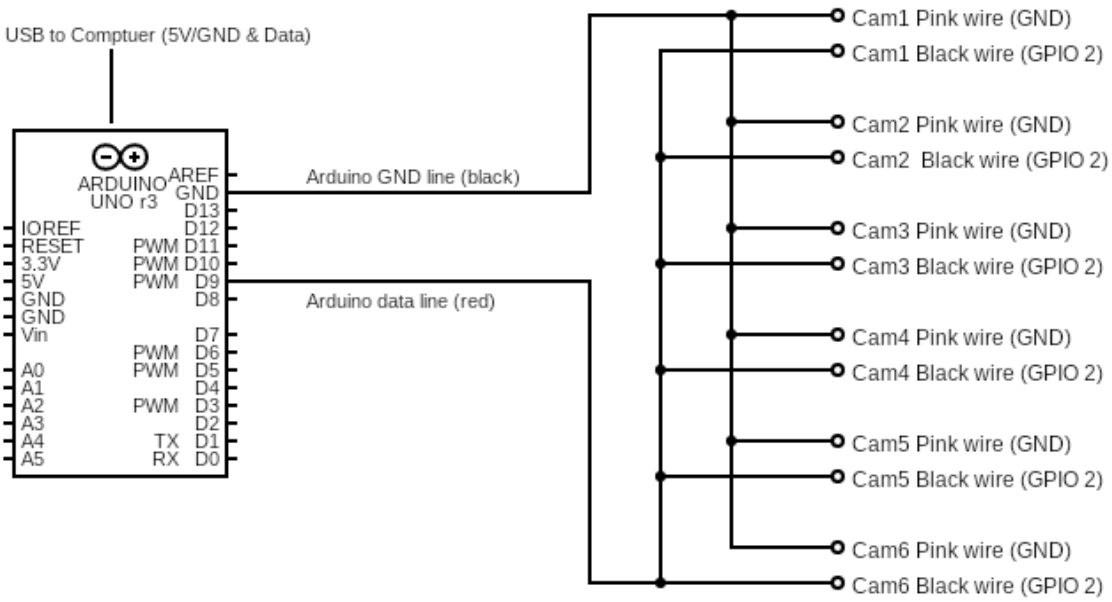
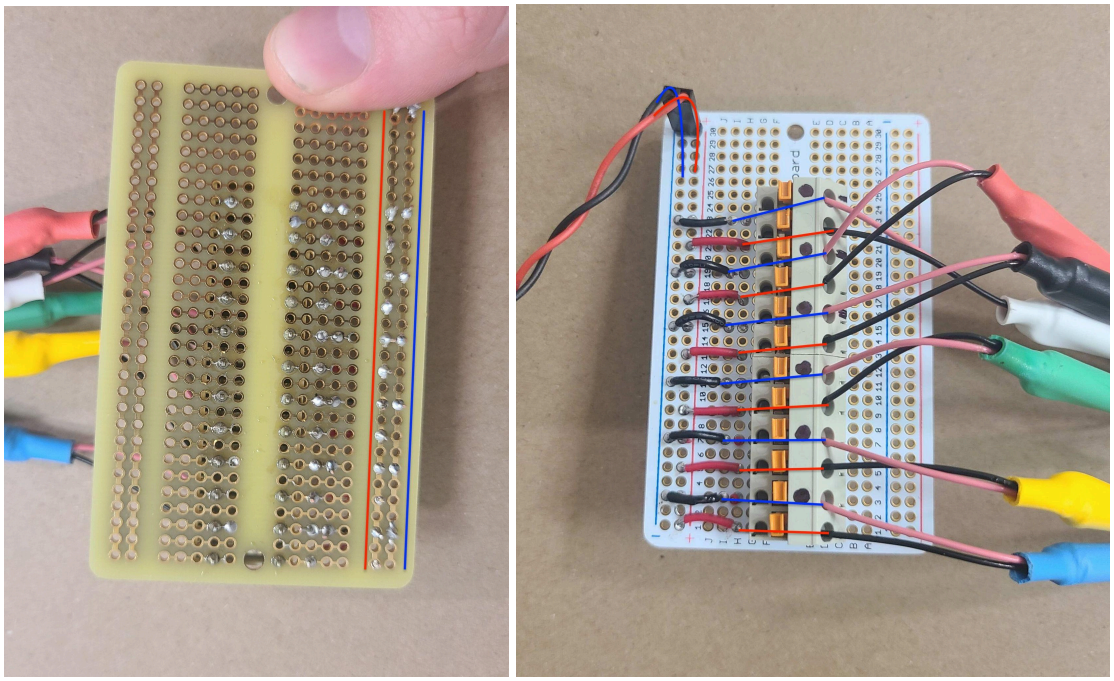


Figure: Circuit diagram of splitter board



Figures: top and bottom of splitter board (red & blue lines show connectivity)

Installing Software

Instructions for Windows:

1. Build computer from parts. E.g. tutorial on pcmag.com.
2. Install Windows 10/11 from a USB stick. See this article if you need to create one.
3. Install the following software:
    - Basler Pylon Software Suite (link)
    - NVIDIA Drivers (link)
    - Anydesk (link) - optional but useful for remote desktop
    - Drivers for your USB PCI-e card
    - Git for Windows (link)
    - Arduino IDE (link)
    - Miniconda (link)

Set up Campy

- Clone campy from github (repo link)
    - `git clone https://github.com/ksseverson57/campy.git`
    - Follow the instructions on the campy repo to install campy

Check USB bandwidth with Pylon Viewer

- Open Pylon Viewer
- You should see all six cameras listed in the "devices" pane. The name for each camera will be the serial number. You may see an additional device called "Camera Emulation", but you can ignore this.

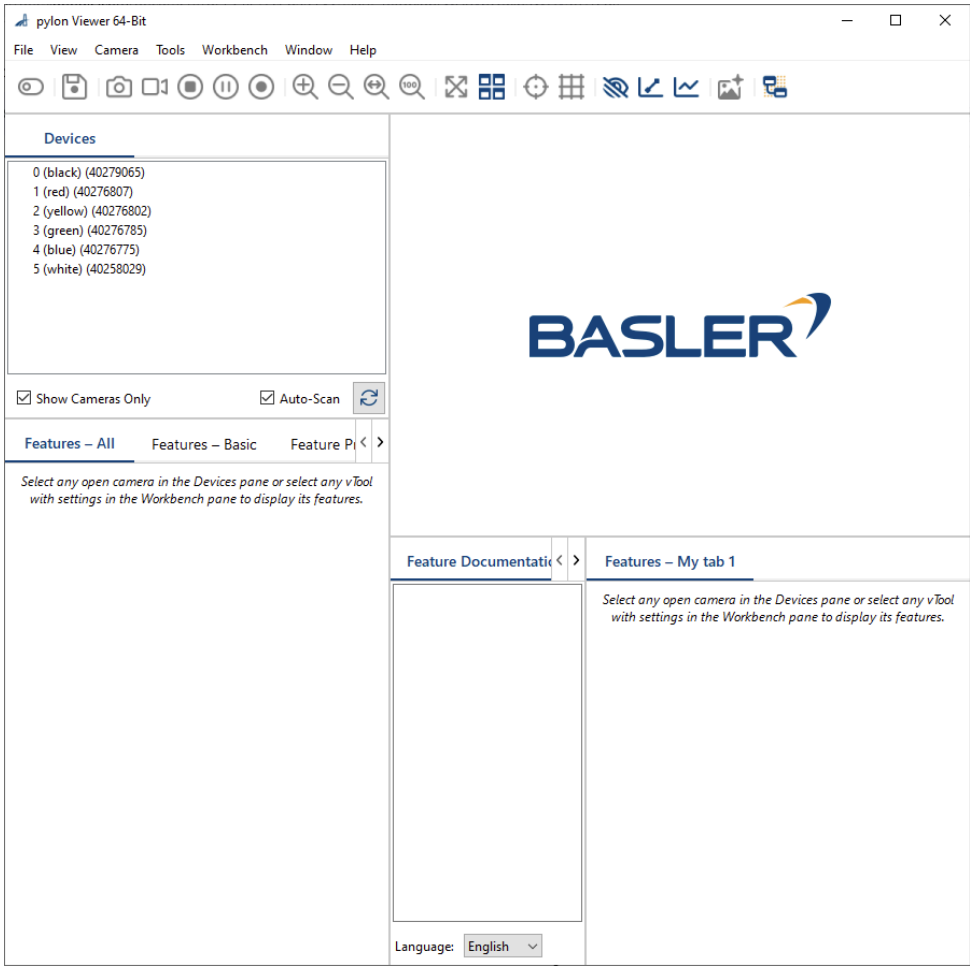


Figure: Pylon Viewer. Note the Devices tab on the upper-left of the window.

- Double-click on each camera in the Devices window. This will open each camera for reading into Pylon Viewer; but images should not yet appear on the screen. The camera name will turn bold when it is open.
- Now, for each camera in the Devices menu, click on the camera's name to select that camera. Next, click on the "Camera Icon" in the top bar to take a picture. Ensure that each camera creates a picture.
    - If you don't see a picture, it is possible you have previously enabled **Hardware Triggering** on that camera. This means the camera will not produce an image until it receives a signal along one of the GPIO lines.
    - To disable hardware triggering: Go to the "Features - All" Tab (bottom left panel). Click on the category "Acquisition Control" and scroll down to the feature "Trigger Mode". Select "Off" from the dropdown menu.
    - It is also possible the camera is not connected properly, or connected to a USB 2.0 port instead of a USB 3.0 port. Check your connections, make sure all camera USB ports are USB 3.0. Check if the PCI-e USB card you're using is properly connected and any drivers are installed.

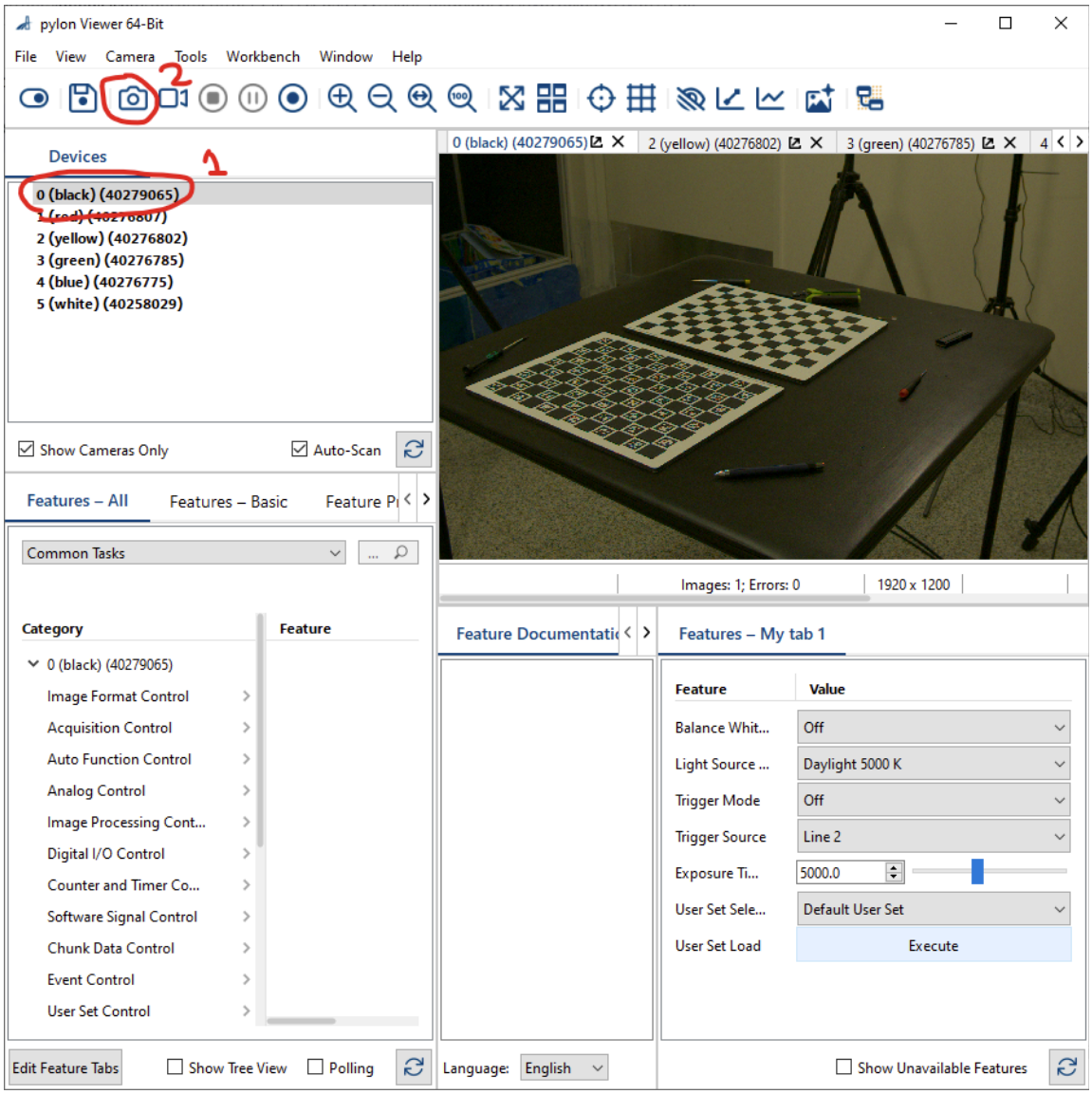


Figure: Opening cameras in pylon viewer and taking single shots.

- On the Pylon Viewer top bar, click on "Tools" -> "Bandwidth Manager". That should open the following window, where you should see all 6 cameras.

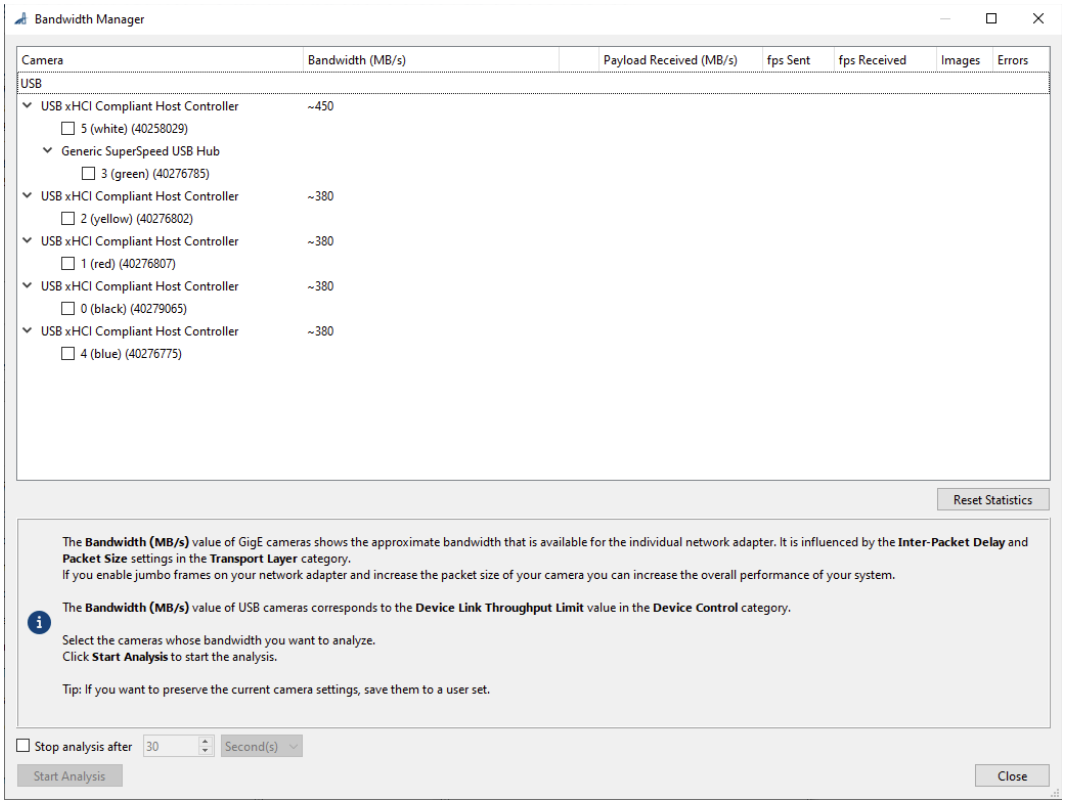


Figure: Pylon Bandwidth Manager

- Select all 6 cameras in the Bandwidth manager, and click "Start Analysis" . Click "Stop Analysis" after a few seconds. Make sure the FPS Received is at least as high as the target FPS for your recordings, and there are no errors received. Ideally FPS sent and FPS received will be the same (within 1 FPS).
    - If this is not working, try plugging in the USB cables to different USB ports on the computer. Some ports share a USB controller and thus share bandwidth. Try filling out the PCI-e extension card's USB ports before using any on the motherboard. Ideally you should spread out the cameras across different USB Host controllers to maximize bandwidth available.
    - If the software gives you an error about Hardware Triggering enabled, this means the cameras are set to use an external source for frame triggering. You can disable the this for each camera in the "Features pane" of the main Pylon Viewer window.

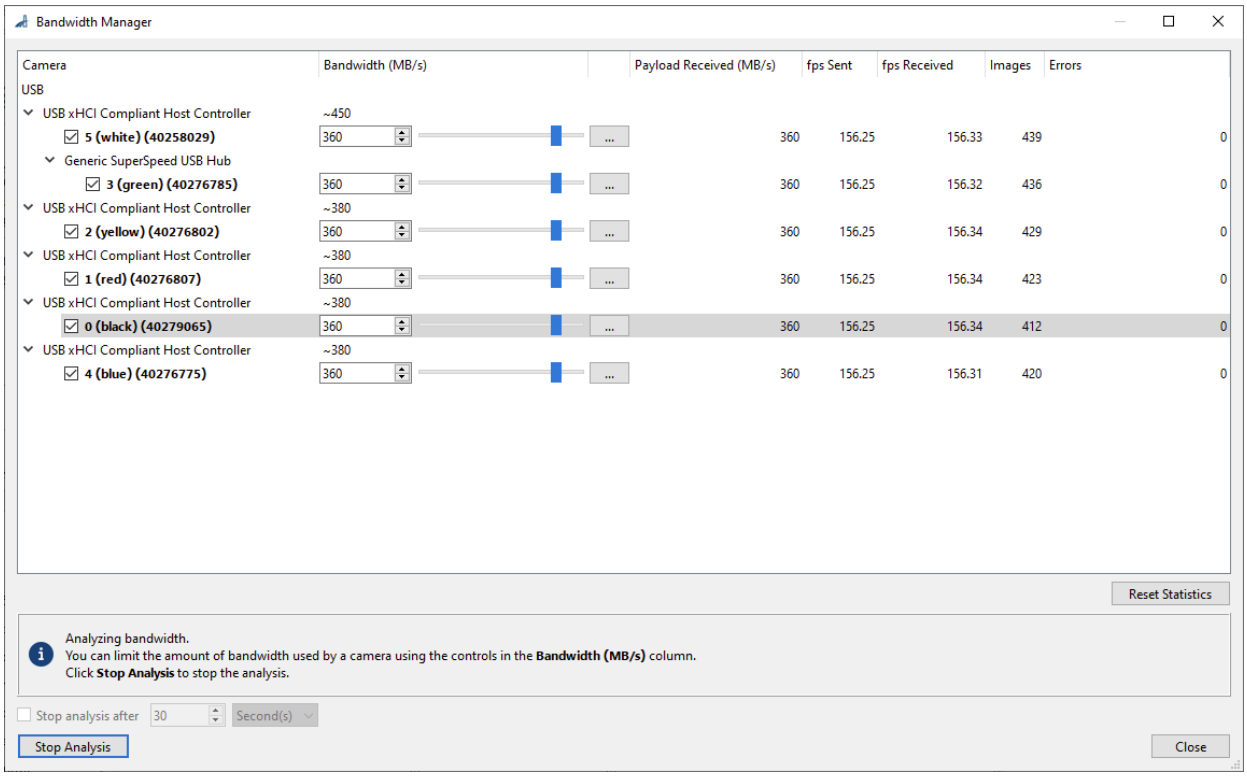


Figure: Pylon Bandwidth Manager has shows bandwidth for all USB cameras.

Configure Features in Pylon Viewer

- Open pylon viewer
- Double click on a camera in the Devices tab to "open" it
- Reset all camera feature by loading the Default User Set
    - Go to category "User Set Control"
    - "User Set Selector" pick "Default User Set" (should be default option)
    - For User Set Load, click "Execute"
    - ⚠️**WARNING: Don't take any picture until you have disabled auto-white balance! If click the camera icon to take a shot, and you haven't changed auto white balance, the camera will adjust the color of the shot for the content. You can re-run the above steps to remove the white balance adjustment.**

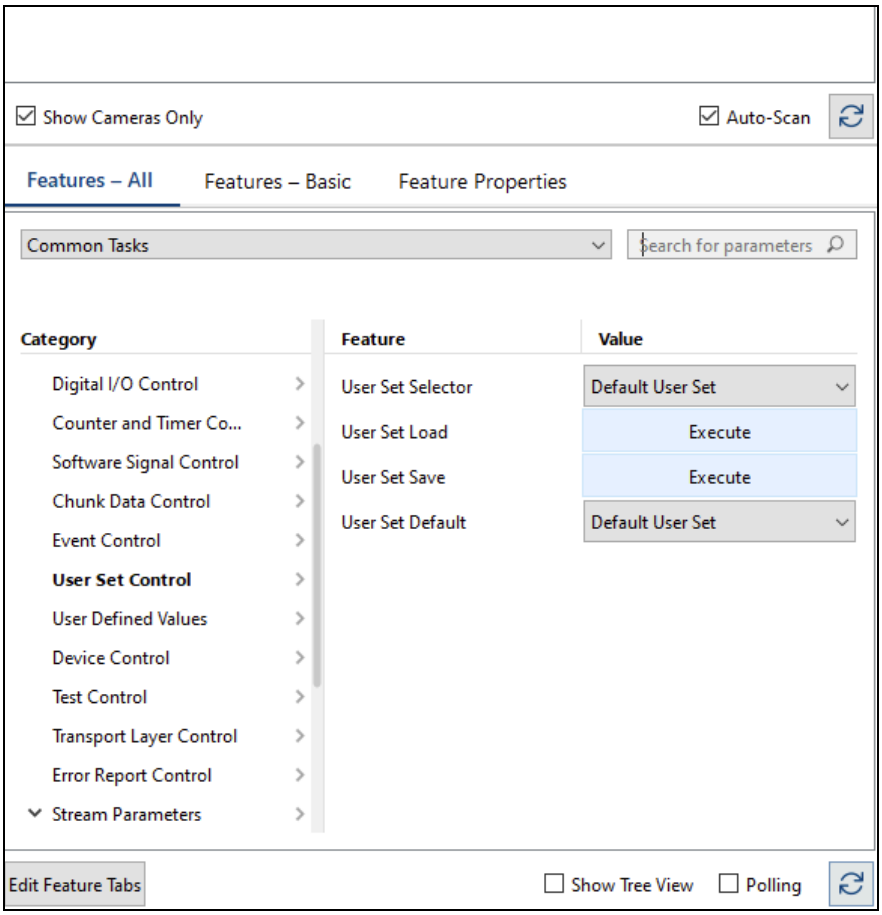


Figure: Loading default set

- Set the following features:
  - **Disable auto white balance** by going to Feature Category: Image Processing Control -> White Balance Auto = "Off". Now you can click the camera icon to preview your features in the view finder.
  - **Change the pixel format** by going to Feature Catoegory: "Image Format Control" → "Pixel Format" and changing from "Bayer RG8" to "RGB8"

Adjust the following features as necessary:

- Image Processing Control -> Light Source Preset
  - This depends on your light sources, but the goal is to make items which appear white in real life look white on the computer. I usually use Daylight 5000K.
- Image Format Control: width, height, offset X, & offset Y
  - Adjust as necessary to remove extra pixels on the top/bottom or left/right of the screen. I don't change these values.
- Acquisition Control: Exposure Time
  - Can change to increase brightness of the image

    - Ideally, keep this value low (e.g. below 5000 microseconds) to reduce motion blur.

NOTE: You do not need to change FPS settings, because the hardware trigger (arduino) will determine the FPS.

Change the following features to enable hardware triggering.

⚠️**NOTE: you will not be able to click the camera icon in Pylon Viewer to obtain preview images after you change this (but you can always change it back temporarily)!**

1. Acquisition Control -> Trigger Mode = "On"
2. Acquisition Control -> Trigger Source = "Line 2"

Name cameras [optional]

This step is optional, but it will make it easier to remember which camera corresponds to which color tape.

Go to the feature: "Features All" -> Device control -> Device User ID

- Enter a name here. I use the format **Number (color)**. E.g. "0 (black)" or "4 (blue)".
    - The number corresponds to the camera index (0 - 5) and the color corresponds to the camera/cable tape.
- You may have to unplug & replug the USB's from the computer and/or restart the computer for this change to take effect. You can tell it worked if the names in the "Devices" pane have changed to your names.
- The camera name in the "Devices" pane will still be followed by the serial number in parentheses

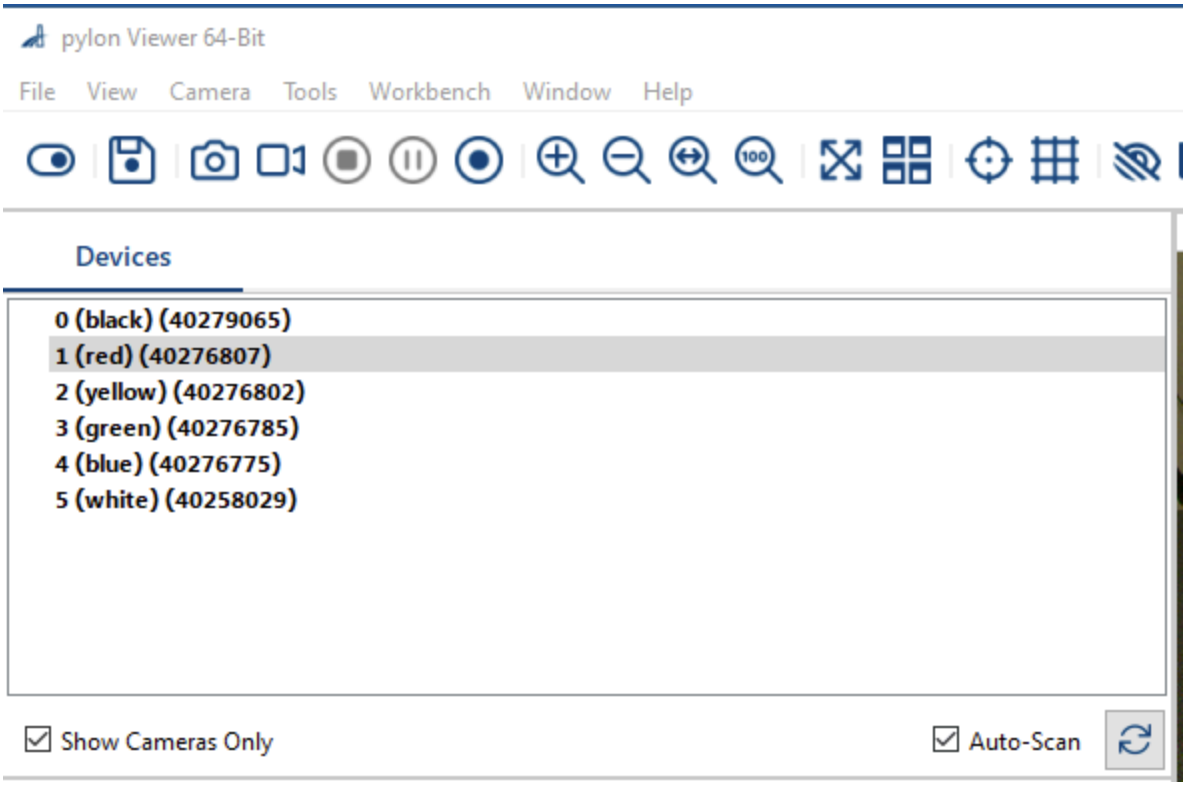


Figure: Devices pane with cameras renamed

Export features to a .pfs file

This file will be used by campy to configure the camera setting when starting acquisition.

You will not need to re-configure the cameras in pylon after creating this file.

HOWEVER, the .pfs file does not overwrite white-balance adjustments, so you may have to go through the procedure to load "Default User Set" & execute if someone else has used the cameras and enabled auto-white balance.

You only need to create one .pfs file for all 6 cameras.

Steps:

- Configure camera features described in prior section
- Double-click the first camera in the list to "open" it
- In the menu bar, click "Camera" -> "Save Features" and select a file location and a reasonable name (e.g. "2024-06-20_ace2_basic_trigger").

Close all cameras in Pylon Viewer

- Right click on each camera in the devices menu and click "Close Device"
- **If you do not close each device, you will not be able to do recording with campy!**
- When the camera is "closed" in pylon viewer, the name will turn from bold to normal font weight.

Set up Microcontroller Trigger

Wiring the Arduino

- Connect the Arduino the the computer with a usb cable
- Connect one of the ground pins on the arduino (labeled GND) to the ground of the splitter circuit
- Connect digital pin #9 on the arduino (labeled 9) to the other pin of the splitter circuit.

Upload the triggering script to the arduino:

- Open the Arduino IDE
- Click "File" -> "Open", and select the following file from the campy repo:
    - "campy/trigger/trigger.ino"
- Select the arduino from the USB dropdown menu:

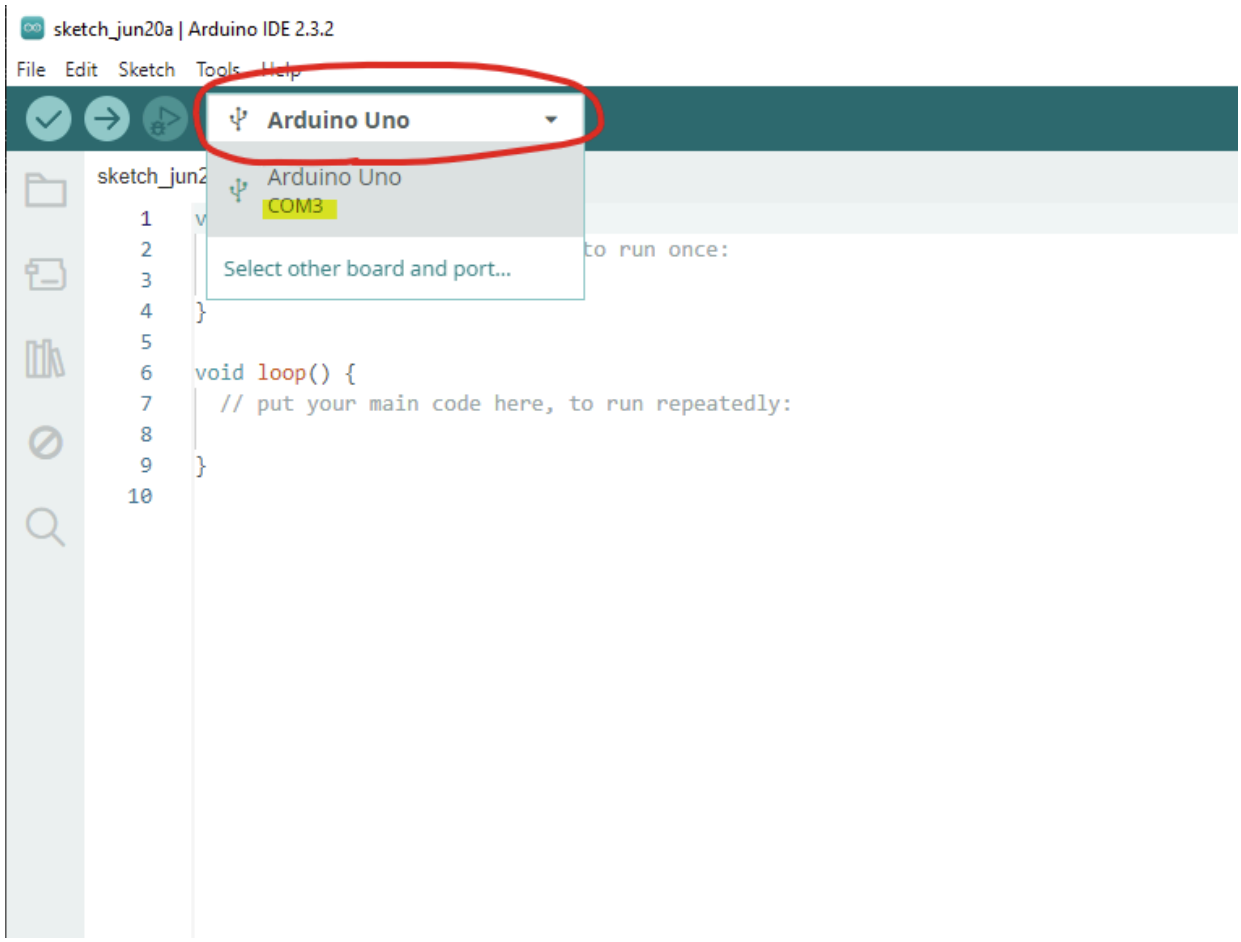


Figure: Selecting arduino from drop-down. Note port name ("COM3") is highlighted.

- Note the port name (e.g. "COM3") in the dropdown - you will need this for the campy config file.
- Click the right-arrow icon in the top menu. This should compile the script and upload it to the arduino. Once you get a notification that the script is "Done Uploading" you can close the Arduino IDE
    - This should take about 5-30 seconds

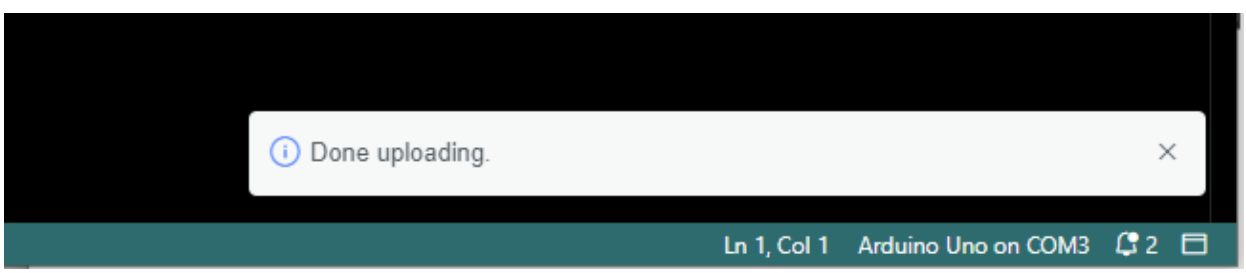


Figure: Done uploading on arduino - OK to close the Arduino IDE but ***do not unplug the arduino!***

STEPS FOR SETTING UP RIG

TODO : add steps for calibration GUI

FAQ

1. **Can I install Linux on the acquisition workstation?**

A: Not recommended. There are versions of Pylon Viewer for Linux; however, the camera's USB communication does not appear to work as well and you may have trouble recording with multiple cameras.

You can use Mac/Linux/Windows for other steps of the DANNCE labeling/training. I personally use windows for the acquisition computer but do most of my development on Mac.

2. **Why do we use a wide aperture?**

A: There is a trade-off between depth of field and speed of the shot which is determined by the camera's aperture. We want to take fast shots (note: this is the amount of time the shutter is open, per frame; and different from the frame-rate of recording) to reduce motion blur (ideally <5ms); but we also want the subject to be well-lit and in focus.

In my experience, at the lowest aperture on our lenses, the entire arena is mostly in focus when the focal plane is in the middle of the arena; and this lets us take the fastest shots with a given amount of light. If you find you need a larger depth of field, you can reduce the aperture slightly (use a higher f-number); just note this will either make the shot darker or slower, depending on how you compensate.

If you need more light/faster shots there are a couple of options but each option has a trade-off:

a. Increase lighting in the arena - you can always add more studio light or overhead lighting, but this may stress out the subject as rats prefer the dark, and change their behavior.
b. Increase the gain - you can increase the sensor gain (similar to ISO in photography). This will increase the sensitivity of the sensor, but it comes at the cost of additional graininess and poorer image quality.

c. Use a larger or more sensitive camera sensor - You can choose a different camera with a newer sensor or a larger sensor format. This can increase the amount of light the sensor receives. The Basler Ace2 Basic cameras we use have use the IMX392 sensor with a diagonal that is 1/2.3" (0.435") diagonal. Some industrial cameras use 2/3" (0.667") sensors and some professional digital cameras use full-frame 1.7" sensors. Larger sensor cameras are typically significantly more expensive and the selection & features may be more limited. This tutorial only covers Basler ace2 cameras, and campy only currently supports Basler and Flir cameras.

3. Does camera resolution matter?

A: For DANNCE, high resolution cameras may not be beneficial, as we already downscale the image before processing it. We have used 2.3MP successfully and found it to be a nice mix of good resolution and cost-effective. I would not recommend less than 2MP however, since you may have trouble with calibration and loss of quality.

That said, recording data is the slowest and most person-intensive part of the DANNCE process, and newer methods may be able to make use of higher-resolution data, so there is no reason not to use higher resolution cameras aside from cost.

Troubleshooting

ERROR MESSAGE:

Launching Pylon Viewer: **VCruntime140_1.dll** not found.

SOLUTION:

Install Microsoft VC Runtime from [this page on microsoft.com](#).

—
ERROR MESSAGE:

"Error while writing node: 'BslSerialBaudRate' not found" –

SOLUTION:

This means the features file (.pfs) is for a different version of the camera. E.g. the Pro version instead of the Basic version. To fix: create a new camera features stream file.

—

ERROR MESSAGE:

Some variation on "Unable to open device" or "Device is in use by another application"

```
ERROR:root:Caught exception at camera/unicam.py LoadSystems. Check cameraMake: Failed to open device '\\?\us
b#vid_2676&pid_ba05&mi_00#8&558a96a&0&0000#{79bfeeca-947f-11e1-bc58-001b2163ef96}' for XML file download. Er
ror: 'Device is exclusively opened by another client.' : RuntimeException thrown (file 'pylonusbtl.cpp', lin
e 497)
```

Figure: error device is opened by another client

POSSIBLE SOLUTIONS

- A camera is likely open in pylon viewer; navigate to the pylon viewer application, right click any bolded names in the "Devices" panel, and click "Close Device".
- It is also possible another program is using the camera. Try closing other programs, including instances of campy.
- If nothing else works, try restarting your computer.

—

GENERAL ISSUE:

Arduino not triggering camera frames

POSSIBLE SOLUTIONS:

- Note the **PINK** wire out of the basler IO cables is **ground**, and the **black wire** is **GPIO Line 2**. **THIS IS UNINTUITIVE**. Make sure you connect the pink wire to the ground line on the splitter board (IO cable pink to splitter board black); and the black wire to the signal line (IO cable black to splitter board red)
- Make sure you've uploaded the sketch to the arduino using the arduino IDE, and the Arduino is powered (e.g. plugged into the computer)
- Make sure all wire connections are secure on the splitter board. You can test for connection with a multimeter.
- Make sure the campy **config.yaml** file has the following entries:
    - startArduino: True
    - serialPort: "COM3"
        - This depends on what port your computer assigns to the arduino. You can check it in the Arduino IDE.
    - digitalPins: [9]

        - This should match the pin number your are using on the arduino. You can alternatively use multiple digital pins and no splitter. E.g. digitalPins: [6,7,8,9,10,11].
- Ensure all GPIO lines are connected.
- Make sure the Arduino IDE is not running a serial monitor. This can prevent other applications from communicating with the Arduino. Try closing the Arduino IDE (you should not need it after uploading the script during initial setup).
- Most camera models have a LED on the back of the camera which turns green if the camera is on. If the cameras do not have this light, it is possible they are not receiving power from the USB, or the camera is broken.

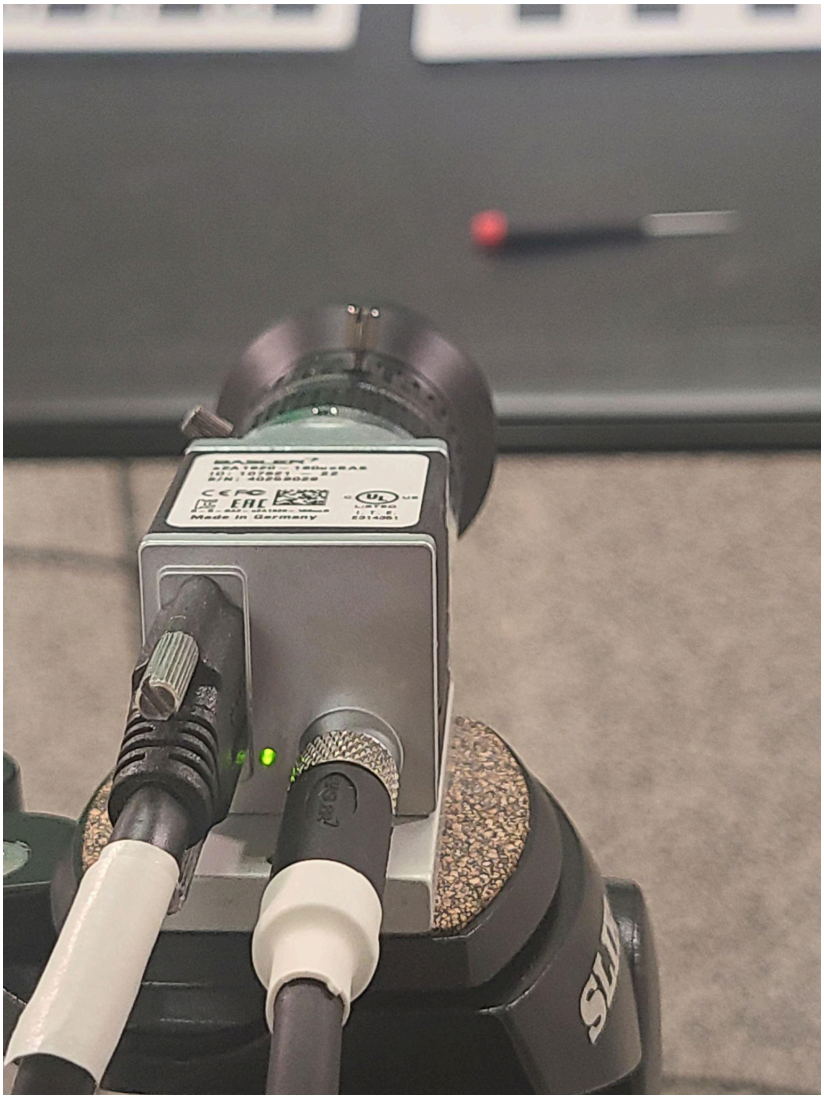


Figure: green LED on camera means it is receiving power

—

ISSUE:

Campy displays monochrome video while recording, even though RGB is enabled. Also, the recorded video files appear corrupted, and show a 3x3 grid of camera views.

E.g .

Figure: corrupted recording with 3x3 grid of monochrome views

SOLUTION:

This means there is a pixel format mismatch between ffmpeg and the cameras. To fix there are two options:

1. Create a new feature stream file (.pfs) with the following option:

   Category "Image Format Control" -> Feature "Pixel Format" set to "RGB 8".

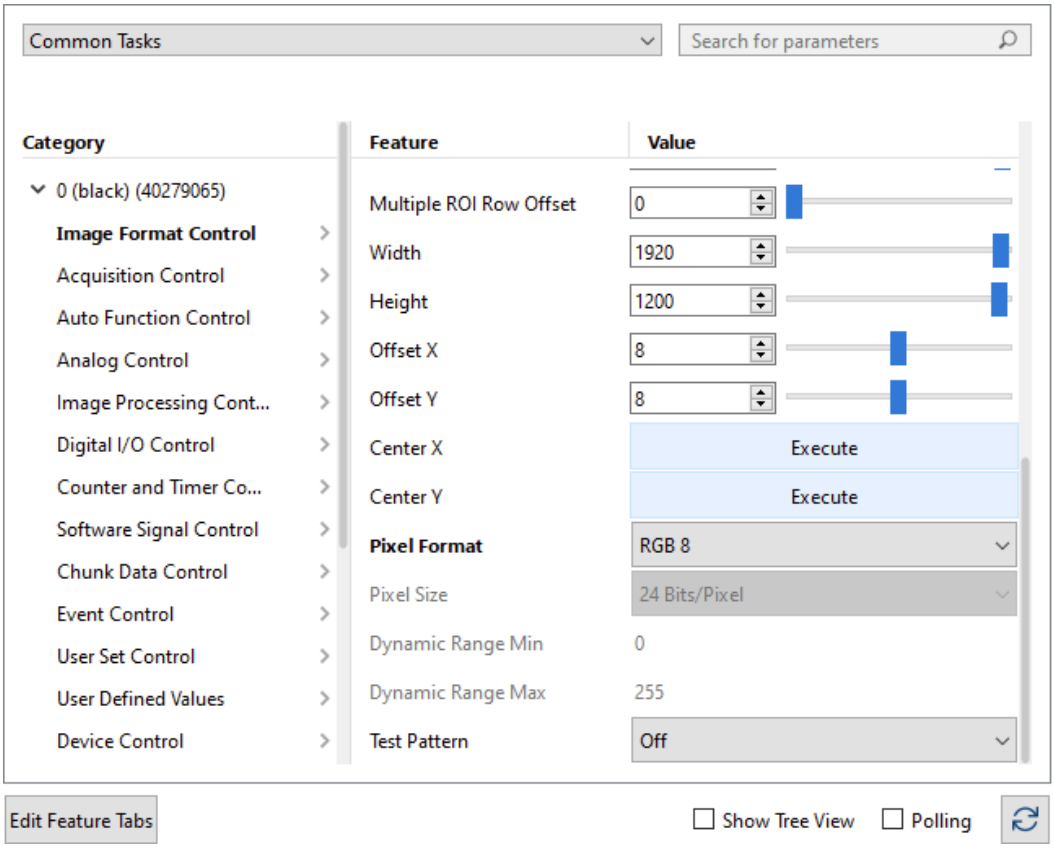


Figure: update pixel format to "RGB 8" in Pylon Viewer

2. Edit the following line in the campy config.yaml:

   pixelFormatInput: "rgb24" → pixelFormatInput: "bayer_bggr8"

The first option is preferred since converting the image on-camera will give a higher quality output, but it may reduce FPS if you are capturing near the camera's max framerate since RGB8 takes more data per pixel.