Stimulus calibration

Overview

Principle

- **What are we calibrating** : **Power of LED emitted light** to make the **photoisomerization rate of mouse M and S opsin** to **green** and **blue light stimulus** respectively be **the same**
- **How do we calibrate** :
  - Measure the **input power** and the **light intensity** (`I/O`) of LEDs
  - Calculate **the relationships between the input power and the photo-isomerization rate** of M and S opsins for green and blue LEDs from the LED `I/O`
  - **Calibrate the input power** accordingly
- See Calculating R star and the **ipython notebook** `stimulator_calibration_v4.ipynb` in the documentation folder for more explanation.

Calibration in 4 steps:

- Collect the information of the **spectrum** of **LED, LED filter, dichroic beam splitter**
- Assemble the hardware for LED `I/O` measurement → see Hardware
- Upload the **ipython notebook** and **Arduino sketches** for LED measurement and plotting the relationships between the input power and the photoisomerization rate → see Software
- Run the **ipython notebook** and get the plots, and adjust the input power accordingly → see Run Calibration Measurement
- Usually the calibration can be done within 5-10 mins, read FAQ if you got some unexpected problems

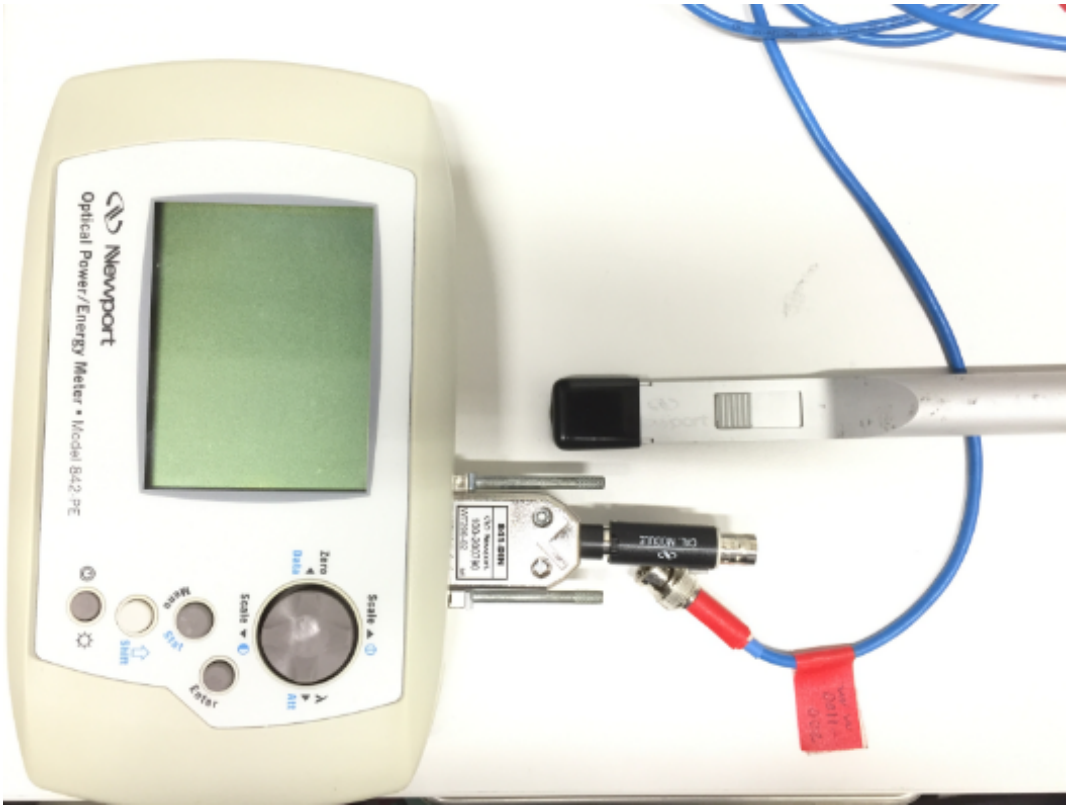
> **Note**: I have only tried the calibration on Setup 1. The procedures **might need to be done differently for the other setups are highlighted**.

Hardware

Photometer

- Power Cable for photometer **(Label `1`)**:

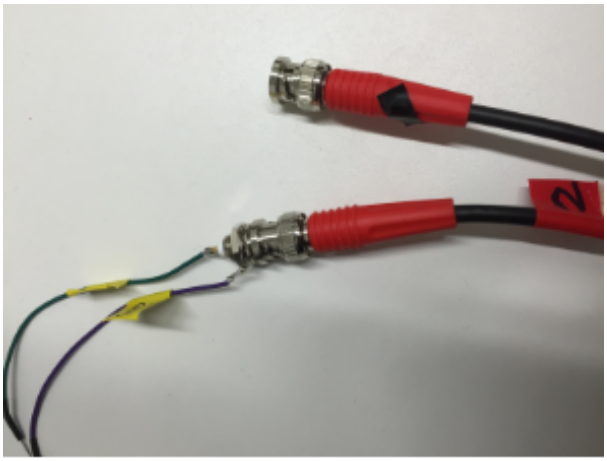
- Others:

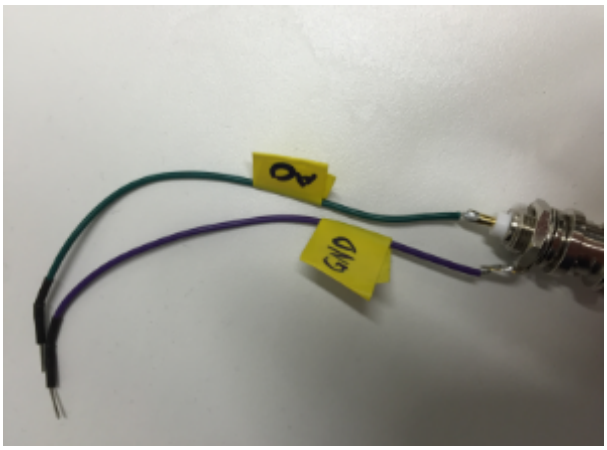


Arduino UNO Broad assembling

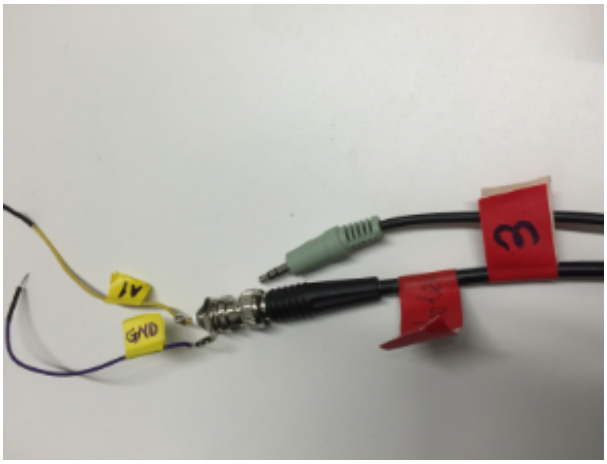
- `UNO` Broad:

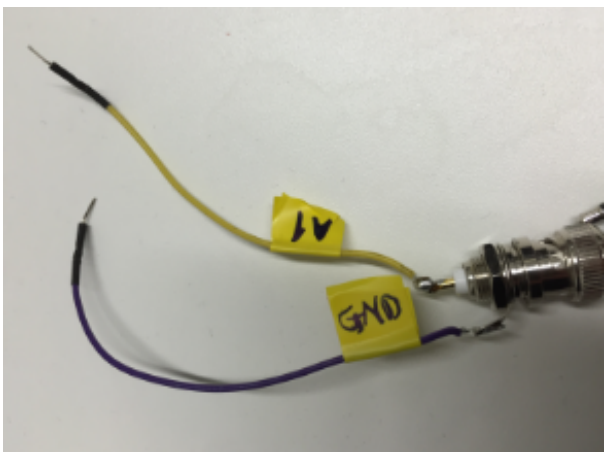
- Cable `[2]` (for trigger input) with two thin cables `A0` and `GND`:





- Cable `[3]` for photometer input with two thin cables `A1` and `GND`:

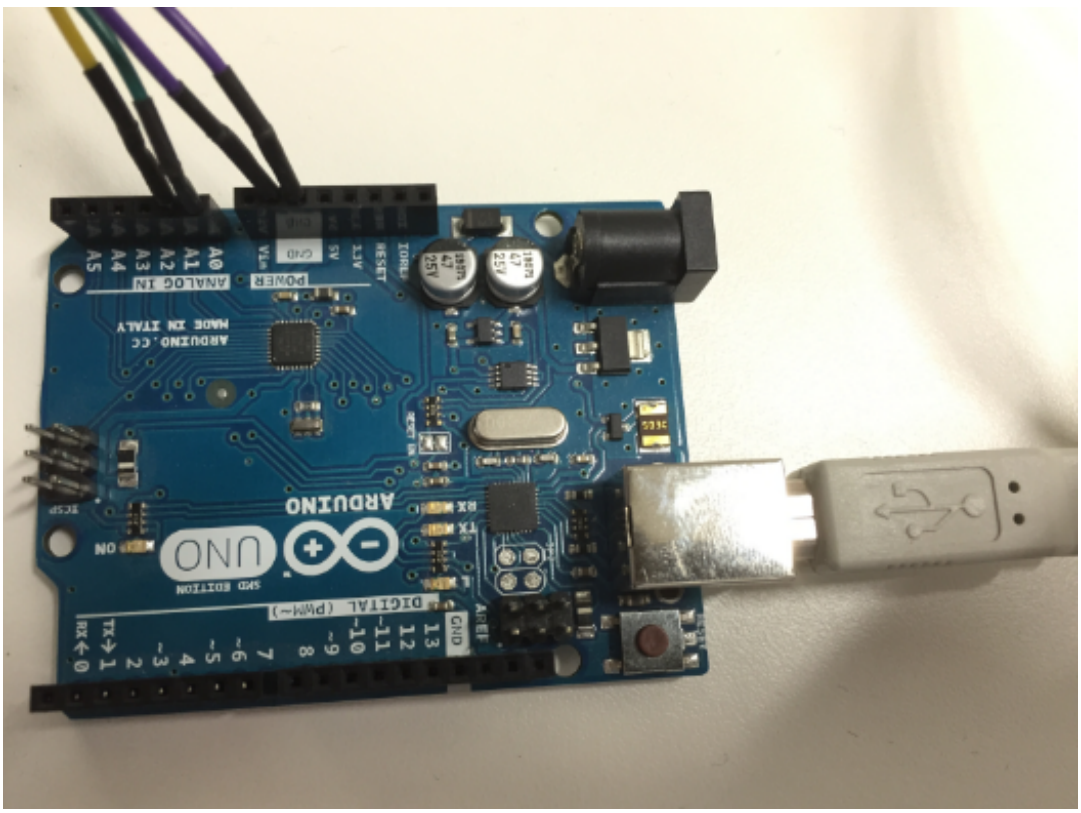




- `USB` cable:

Assembling

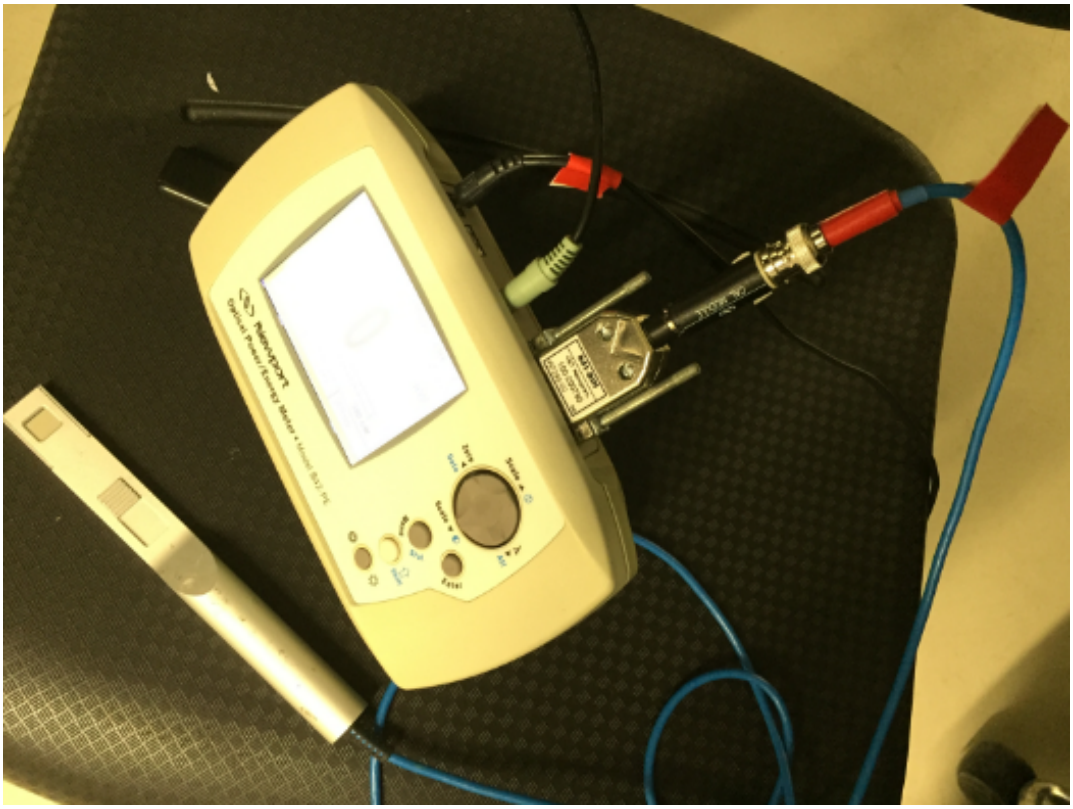
- Connect `USB` to the **port on Arduino broad** and **on PC**
- Connect the thin cables with `GND` label to the GND ports on `UNO` (*GND: Ground*)
- Connect the thin cables `A0` (on Cable [2]) and `A1` (on Cable [3]) to the A0 and A1 ports on `UNO` respectively:



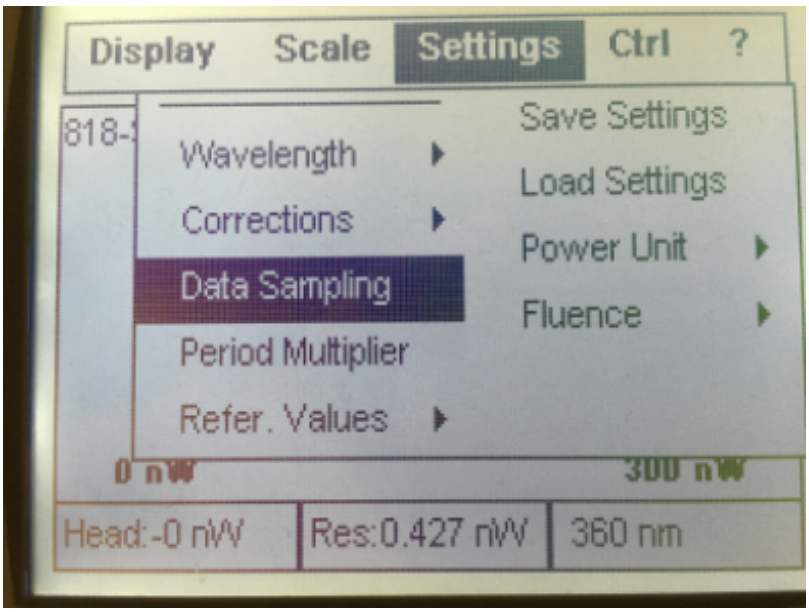
- Connect the other end of Cable `[2]` to **the port for trigger channel**

> **I have only labeled the trigger channel port for Setup 1**, you need to find out the right port for the other setups

- Assemble the photometer: Plug in Cable `[1]` and `[3]`:



- Switch on the photometer, check the **data sampling rate** (set to 20 pts/sec, can be higher):

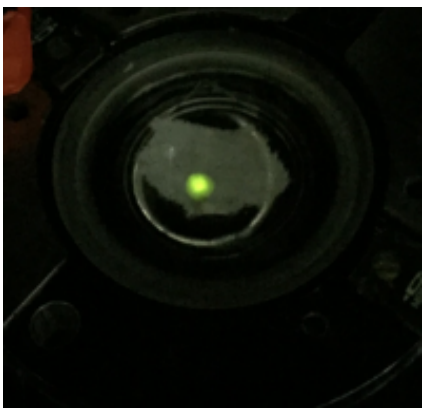


- Turn on the **analog output**:

- Turn on green and blue LED, **turn off infrared LED** to avoid any interference :

Might be different in the other setup

- **Adjust the condenser** (below the microscope chamber) so the photometer can get max light input:



Might be different in the other setup

- **Expose the photometer sensor** and **place the photometer sensor area in the centre of the chamber**. **Measure in dark** to avoid the interference of other light sources:

Software

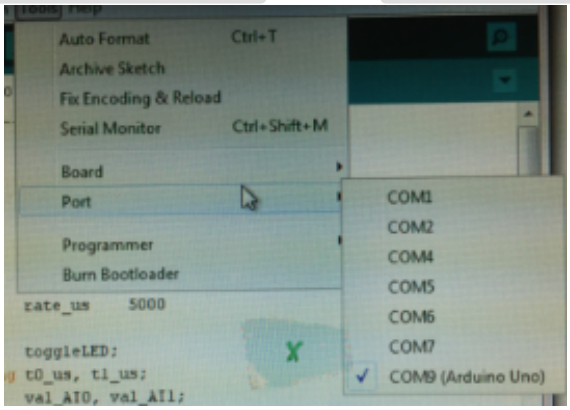
- General: **Igor, jupyter notebook, python, pyserial python module**
- The iPython notebook `stimulator_calibration_v4.ipynb` in the "softwares" folder
- `SimpleAIRecorder.ino` in the "arduino" folder
- `Calibration.ino` in the "calibration" folder (if your LED is controlled by Arduino broad)

Preparation for calibration

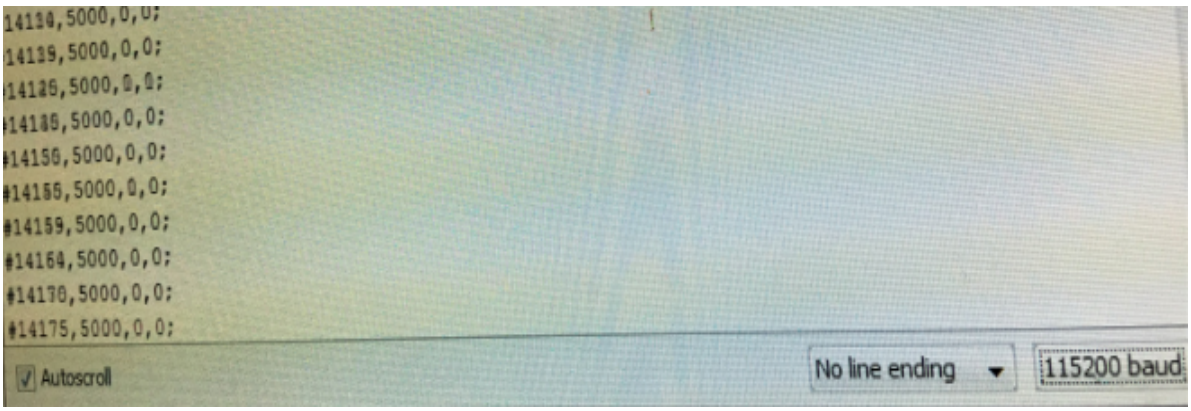
Part A: SimpleAIRecorder.ino

- This sketch file returns the trigger time and light intensity data to PC so the iPython notebook can access the data for calibration.
- Open and upload the `SimpleAIRecorder.ino` sketch to `UNO` broad in Arduino IDE, **before you upload**, make sure:

  Board type: `UNO`

> **Port**: `COM... (Arduino Uno)` (usually `COM9 (Arduino Uno)` but can be changed sometimes)
>
> 
>

- Open the `serial monitor` (the rightmost icon on the image above) and switch baud to `115200 baud`, you should see something like this:

> 
>
>
> - For each line from left to right: **time (ms)**, **measuring rate (µs)**, **A0 (trigger) input**, **A1 (photometer) input**
> - Switch on the photometer and see if the A1 input returns nonzero number, if not, check if you assembled everything correctly.

Part B. Calibration.ino Might be different in the other setup

- This sketch file contains **the calibration stimuli for green and blue LEDs** which allow to measure the power for a series of light levels applied to the different LEDs in sequence
- Switch on the **microcontroller** and open **Igor**, set a configuration and start to scan (otherwise the LEDs won't response)
- Upload the `Calibration.ino` sketch to **LED controlling Arduino broad**, again **check the broad type and the port**

> In **Setup 1** the broad type is `Duemilanove` and the port name is `COM6`.

- Open the serial monitor for the `Calibration.ino` sketch, set **green** or **blue LEDs to maximum** (commands = `2` and `4` respectively) in turns to **adjust the measuring range of the photometer to a reasonable range**:

> - Make sure you change the **measuring wavelength of the photometer** (`λ` on the photometer panel) to the corresponding green and blue light wavelength and **'Zero'** the photometer before every measurement.

    - **Make sure you set the LED you are not measuring to minimum level** (`1` for green LED, `3` for blue)

- Play a white flash stimuli (command = `8`) and see if the change of the **A0** and **A1** values returned in the **serial monitor for SimpleAIRecorder sketch** match the stimuli.

Part C. iPython notebook

- Type `jupyter notebook` in Windows Powershell, which should open the **Home page** of the notebook. Upload the iPython notebook and open it in a new tab.
- The notebook contains **the script for calibration** and also **detailed explanation of the reasons and principles of the calibration** which I would highly recommend to read through before your first calibration.
- The following parameters are required by the script and you need to know them all before the calibration:
    - **LEDs and LED filters spectrum**

      For Setup 1, a narrow filter is used therefore only the peak and the bandwidth of the LED/filter spectrum)

      For the other setup, .txt files of the spectrum can be found in the data folder (e.g. "F73-063_z400-580-890.txt")

    - **Spectrum of dichroic beam splitter**:

      For setup 1 not needed, because the spectrum of the beam splitter is mostly overlapped with the spectrum of the two LEDs

    - **Mouse M and S Opsin spectrum**: "mouse_cone_opsins.txt" in the software folder
- The paths: `script_path` and `pathData` In the **2nd and 3rd python code block** of the notebook are the path where your **ipython notebook, mouse opsin spectrum and LED/filters spectrum files** are located respectively. `pathData` is also the path**to save all the calibration measurements**
- In the **10th code block** (**Read calibration data**), change the `comPortName` if it does not match with the port name of the `UNO` **Arduino broad**.
- The `calFileName` is the **file name for the next calibration measurement**, change it every time before you start a new measurement, otherwise no data will be saved.

Run Calibration Measurement

1. Run the **11th code block** in the notebook, and you should see the following:

```
Illuminated area is 1.963 mm2
Prepare to measure LED `green` and press Enter to continue...
```

2. **Set the measuring wavelength of the photometer to the peak of the corresponding LED spectrum** (e.g. in setup 1, measuring wavelength for green LED should be set to 578)
3. In **the serial monitor window for the** `Calibration.ino` **sketch**, type `g` which is the command for green LED calibration measurement.
4. Wait until seen this:

```
Illuminated area is 1.963 mm2
Prepare to measure LED `green` and press Enter to continue...
Opened serial port COM6 at 115200 baud
Sampling rate is 5.001 ms
56.000 s duration = 11198 samples
100% done
SUCCESS
11198 data points recorded
Rate = 5.000 +/- 0.001 ms
Closed serial port COM6
Prepare to measure LED `blue` and press Enter to continue...
```

5. Repeat **Step 2 and 3** for the other LED (type `b` for blue LED calibration measurement).
6. Run the rest of the script until you get **the plot of photoisomerization rates for each LED/filter vs. photoreceptor combination**

FAQ

Q: The readout of photometer is not stable or too small

> a. Make sure the cables are not broken.
>
> b. examine **the connection between the photometer calibration module and the photometer device** (where your photometer sensor connect to the device)
>
> c. Adjust the condenser and the position of the photometer sensor to make sure the sensor can receive the maximum output
>
> d. Adjust the scale of the photometer to a reasonable range
>
> e. Make sure you **take off the cap and pull down the filter to really expose the photometer sensor**
>
> f. Check if the hardwares are wired correctly (e.g. `A0`, `A1`, `GND` really connect to the right ports). Make sure you don't shunt anything

Q: The photometer's readout is out of range

> a. Make sure the sensor is placed in dark so **no other light source** can interfere the photometer measurement.
>
> b. Adjust the scale of the photometer to a reasonable range
>
> c. Check the wiring and make sure you don't shunt anything
>
> d. Check if **the IR-LED has been turned off**

Q: Photometer turned itself off..

> Plug in the power cable ... (one stupid mistake I repeatedly made)

Q: Takes forever to upload the Arduino sketches

> a. Check if you select the **correct broad type** and **ports** on the Arduino IDE
>
> b. Check your **firewall settings** and open and upload with the administrator permission
>
> b. Check if you can successfully compile your sketches

Q: Arduino returns the error "access is denied...." when uploading the sketches to the port

> > The Arduino broad has been occupied by some other processes (usually the ipython notebook and the python functions it called), you must release it before uploading the sketch.
>
> a. **Restart your ipython notebook**
>
> b. **End the python processes**
>
> c. Reboot your Arduino broad by re-plugging in the USB cable
>
> d. Restart the PC

Q: Random weird symbols instead of numbers displayed on the SimpleAIRecorder serial monitor

> Switch baud to `115200 baud`

Q: A1 inputs shown in the serial monitor are out of range (e.g. A1 input = 1023) or stay at zero:

> > **A1 inputs should be in the range from 0 to 999.**
>
> a. Check if the `A1` / `A0` cables are **mistakenly plugged into the wrong port** on the `UNO` Arduino broad.
>
> b. Check if the **analog output** of the photometer has not been turned on
>
> c. Make sure Cable `[2]` is connected to the photometer device.
>
> d. Make sure you have turned on the LEDs (**please don't directly stared at the blue and UV LEDs**)
>
> d. **Turn on the microcontroller, open Igor and start recording**

Q: Error: "No module named 'simple_ai_recorder'" when running the 2nd code block

> Check if you can find **simple_ai_recorder.py** under the `script_path` you defined

Q: Error: "No module named 'serial'" when running the 2nd code block

> Open **Windows powershell**, type `pip install pyserial` and return, wait until the installation finished

Q: Error: "[Errno 2] No such file or directory'" when running the 6/7/13th code block

> Check if the **"mouse_cone_opsins.txt"** and your **spectrum files** is under the `script_path` and the `pathdata` you defined

Q: No messages or plots returned after starting the calibration measurement (11th code block)

> This can be due to many reasons, here I picked the most common ones:
>
> a. `comPortName` does not match with the port name of the `UNO` **Arduino broad**.
>
> b. The current python kernel is occupied by some other processes, **restart the kernel**.

> c. Check the `A0` / `A1` / `GND` connection and make sure you don't shunt anything

Q: 'Error: Link not open' when running the 11th block

> Usually caused by the connection problems between the `Uno` broad, the photometer and the PC. Check if you assemble the hardware correctly and you don't shunt anything

Q: 'Error: list index out of range' when running the 12th block, no errors returned in the 11th block

> Check if the `A1` / `A0` cables are **mistakenly plugged into the wrong port** on the `UNO` Arduino broad.

Q: 'Error: Less power levels recorded ...' in 12th block

> Check **all the parameters in the 10th code block**, e.g. `nLevels` if it's smaller or greater than the number of levels you measured