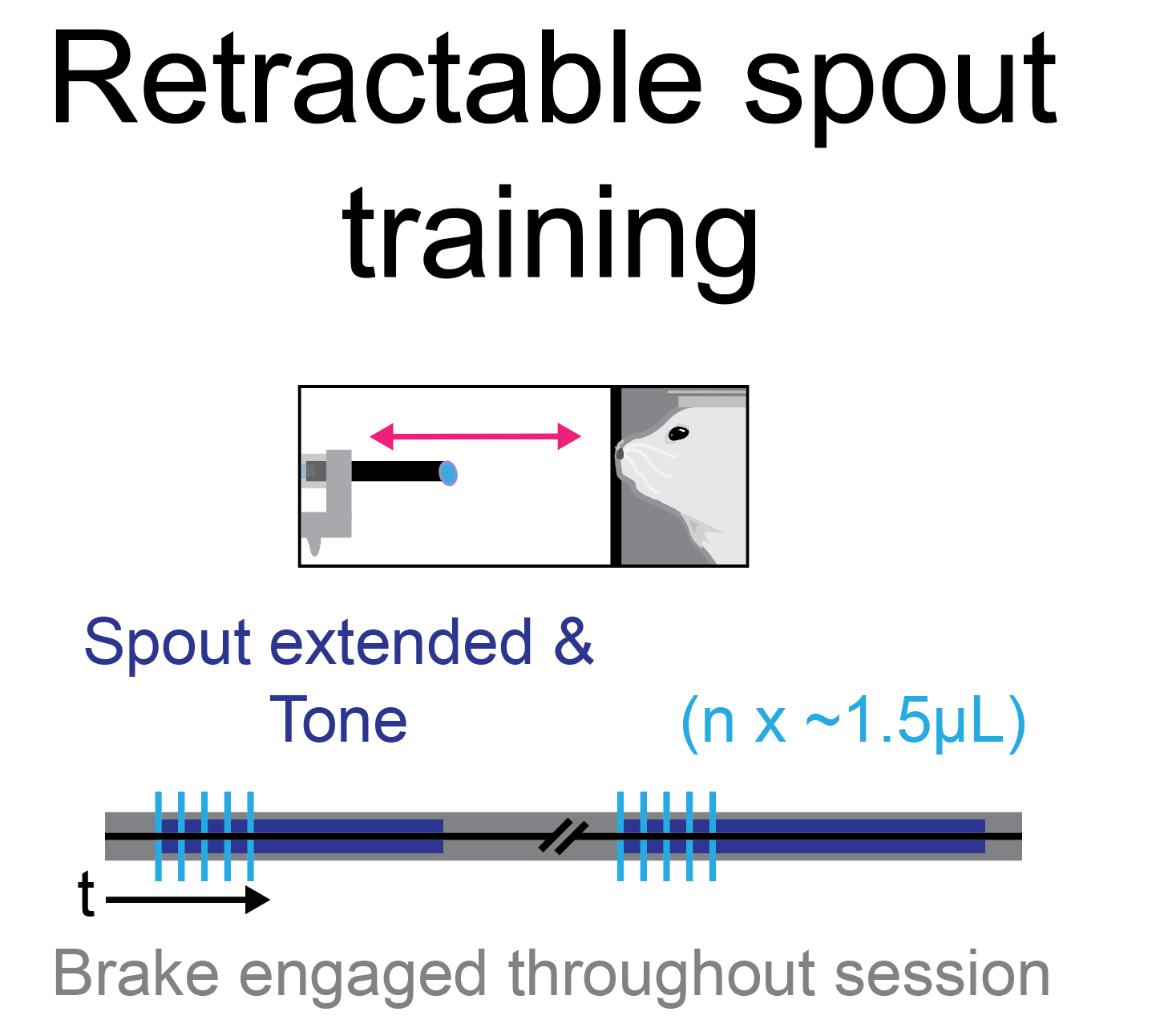
**Protocol for beh\_retractablespout.ino**



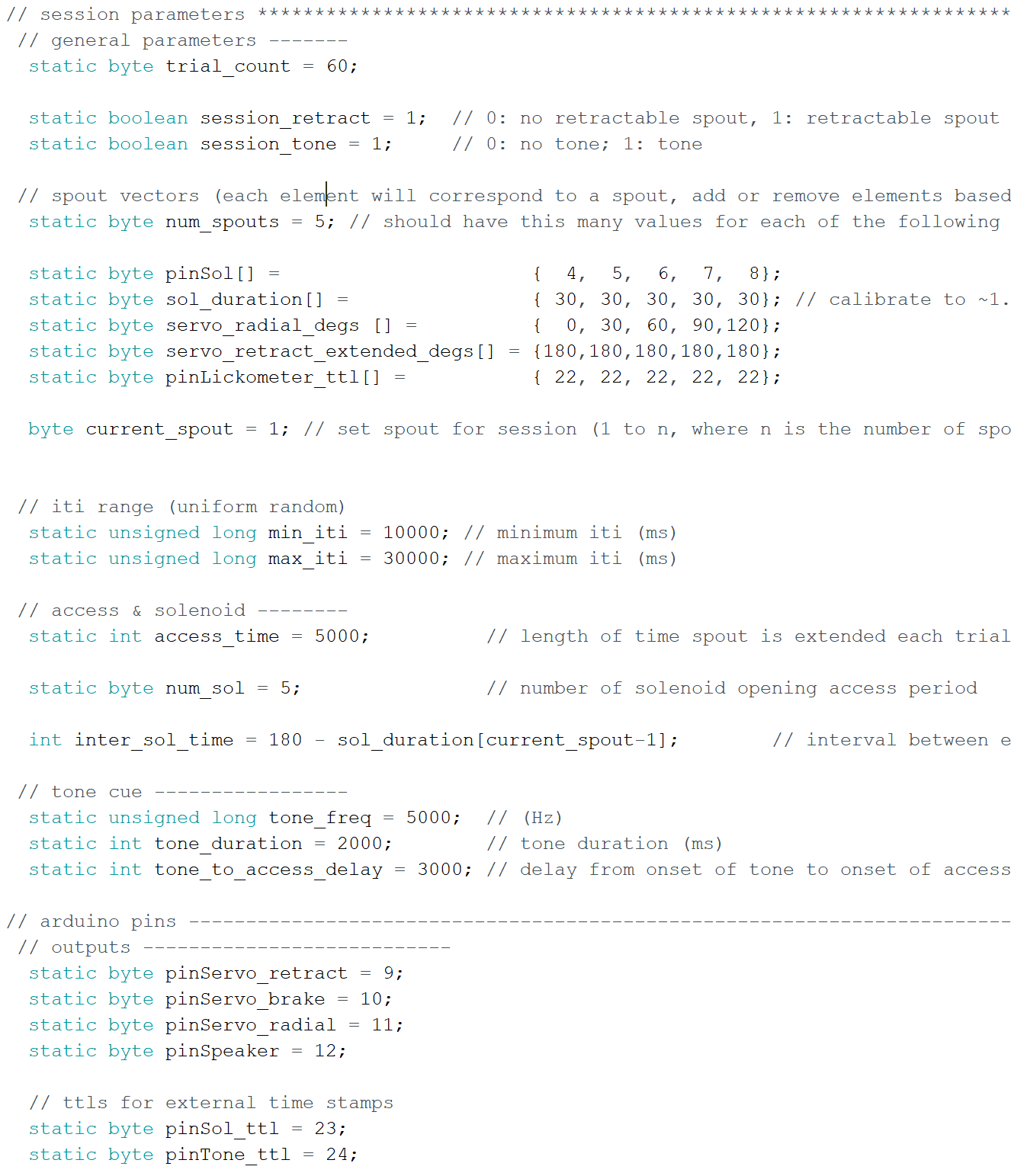
Program purpose: delivery of a fixed volume of solution with options for retractable spout training and Pavlovian Conditioning.

**Preparation Instructions:**

1. Around once per week, calibrate the solenoid open duration by following *protocol\_helper\_calibratesolenoid*. Save the solenoid open duration(s) somewhere convenient in a vector format with 1 value per solenoid (e.g. 3 values for 3 spouts: each vector would look like {*value*, *value*, *value*}).
2. Fill the solution line(s) by following *protocol\_helper\_opensol*.
3. *Optional: connect BNC or TTL outputs from arduino console to external hardware*

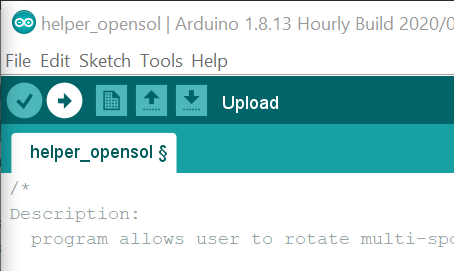
**Arduino Setup Instructions:**

1. Open the program
2. Set the parameters and pins (see Arduino script for all parameters / comments)



* Use the tone\_duration and tone\_to\_access\_delay to set up the timing for tones and access period / liquid delivery.
* For spout / sol pins & parameters, you will need 1 value for each of the spout / sol that your system has. This way you can set current\_spout to perform free-access licking on any spout (e.g. current\_spout 1 uses the first value for each vector, current\_spout 2 uses the second value, etc.).
  + Note: If your system only has 1 spout, then change the vectors to be a length of 1 (e.g. pinSol[] = {4}; sol\_duration[] ={30}; etc.).
* It is fine if you do not have a radial servo, so long as nothing is plugged into the corresponding pin.
* pinLickometer\_ttl is used as an external timestamp for onset of licks on each spout.
* pinSol\_ttl is used as an external timestamp for onset of any solenoid opening.
* pinTone\_ttl is used as an external timestamp for the tone (high for the duration of the tone).

1. Upload script to arduino by clicking the “Upload” arrow button on the top left corner

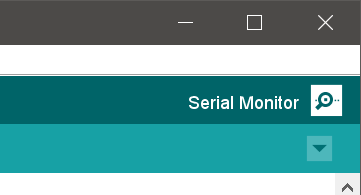


* The arduino will now have the script running on it and it will wait until a start command is sent over serial.

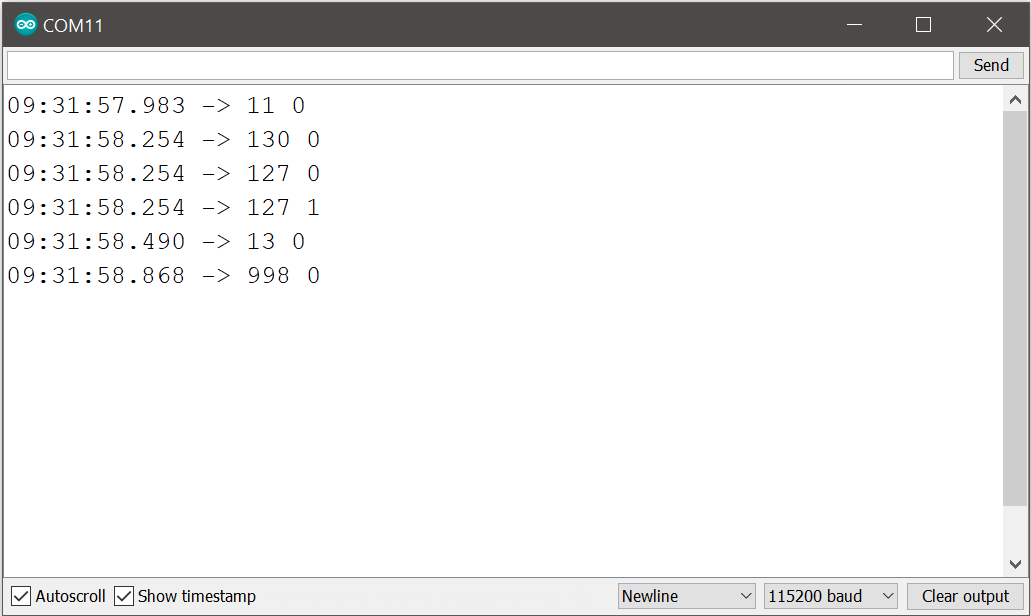
**Arduino Test Instructions:**

Test the program at the start of each day to ensure all hardware is working and events are being recorded correctly.

1. To test the program, open serial monitor by clicking the “Serial Monitor” button on the top right corner

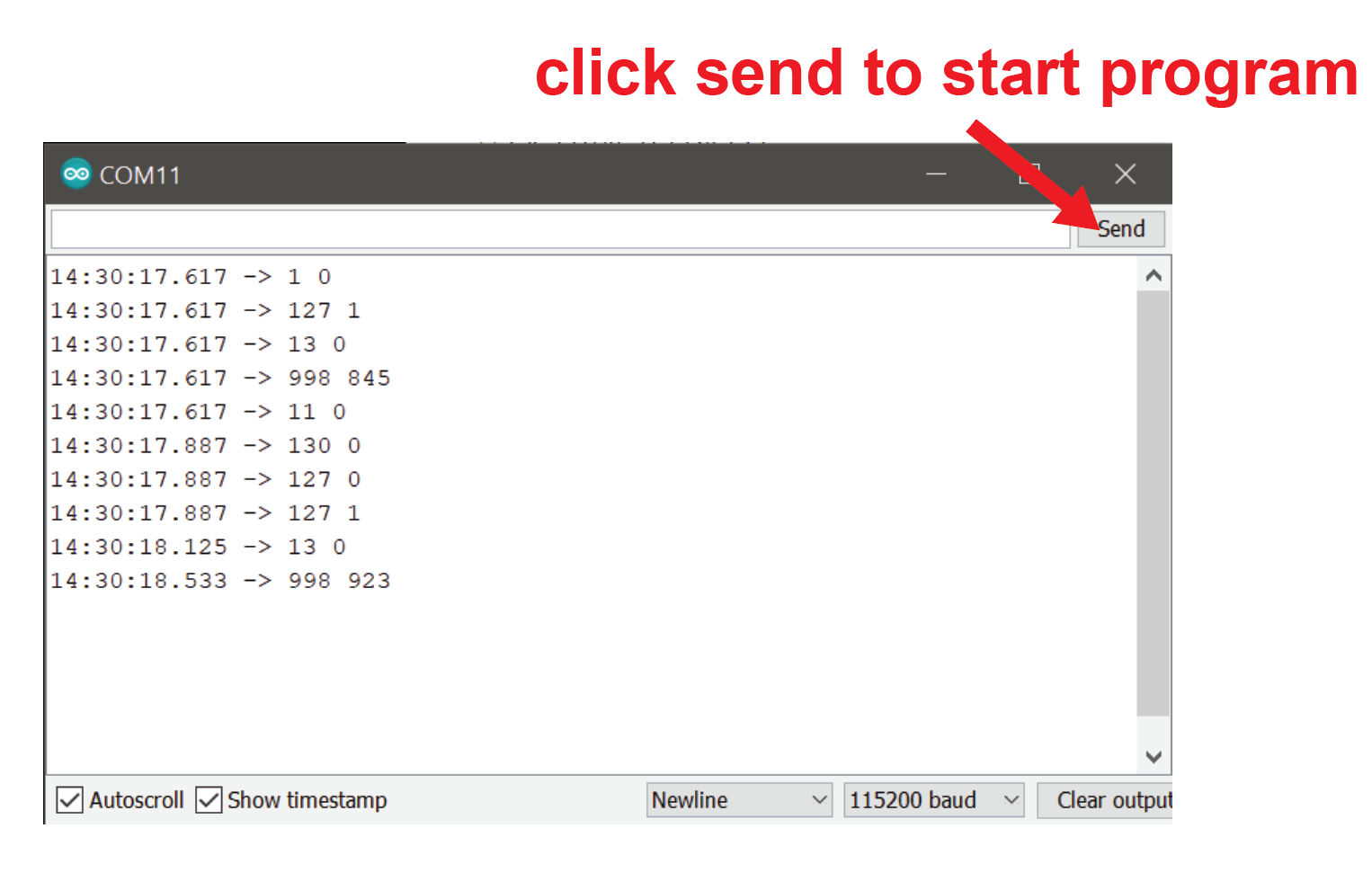


You should then be presented with this screen:



*Note: If you do not see this text printed in the serial monitor, see troubleshooting arduino software document.*

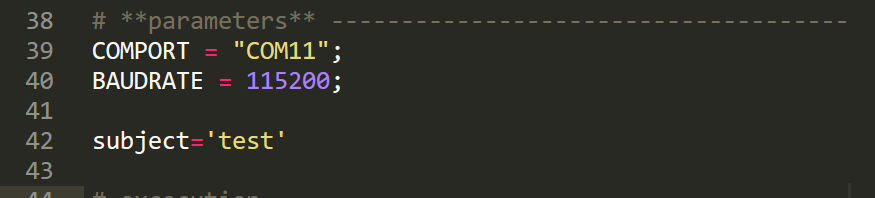
1. Start the arduino program by click the button to send an empty string over serial



1. Wait for the trial to begin (tone indicated with code 51, spout extension with code 13, solenoid opening with 18) and check to ensure that liquid is coming out of the tip of the spout. Next, manually touch the spout gently with the tip of an ungloved finger several times to ensure that licks are being detected (indicated with code 30 + current\_spout).
   1. For multi-spout systems, if the intended spout is not detecting touch, check other spouts to see if they are detecting touch. If you find that a spout other than the one rotated towards the mouse is detecting touch, then check the servo\_retract\_radial\_degs vector to ensure the values are correct for your Multi-Spout Head, and check to ensure the servo is working properly.
   2. If liquid is emerging from the wrong spout, then check the pinSol vector to ensure that the correct pins are included for your system.
2. Reset the arduino by closing and reopening the Serial Monitor, and then close Serial Monitor prior to the session.

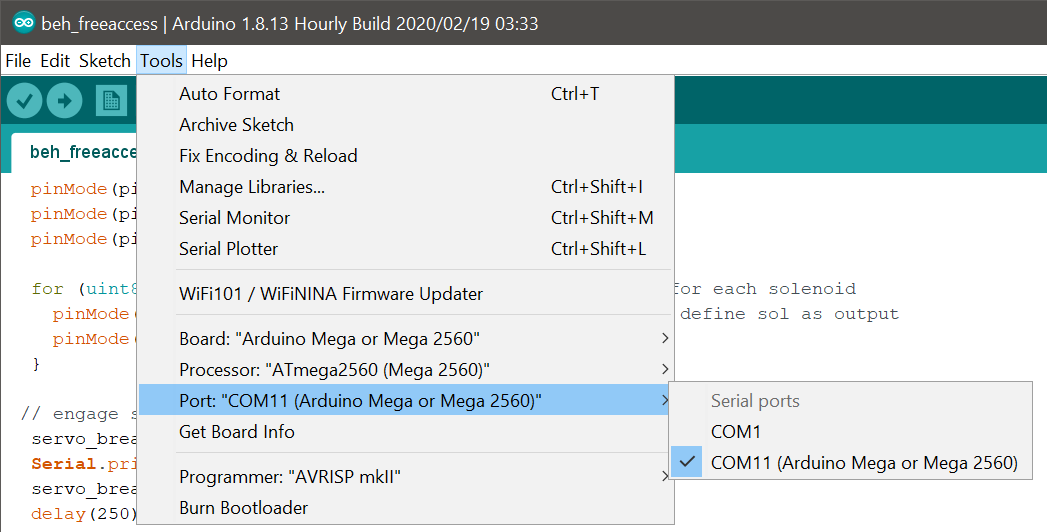
**Arduino Run Instructions:**

1. Open the *write\_serial.py* program (shown here using Sublime Text 3)

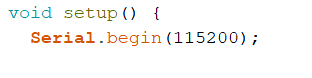


- Set the COMPORT and BAUDRATE to match the arduino script

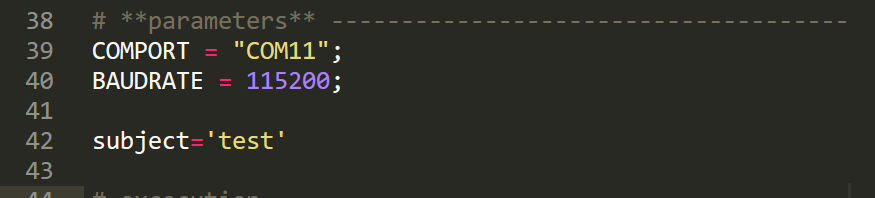
Arduino comport is shown in the tools drop down menu



Arduino baudrate is set with “Serial.begin” call

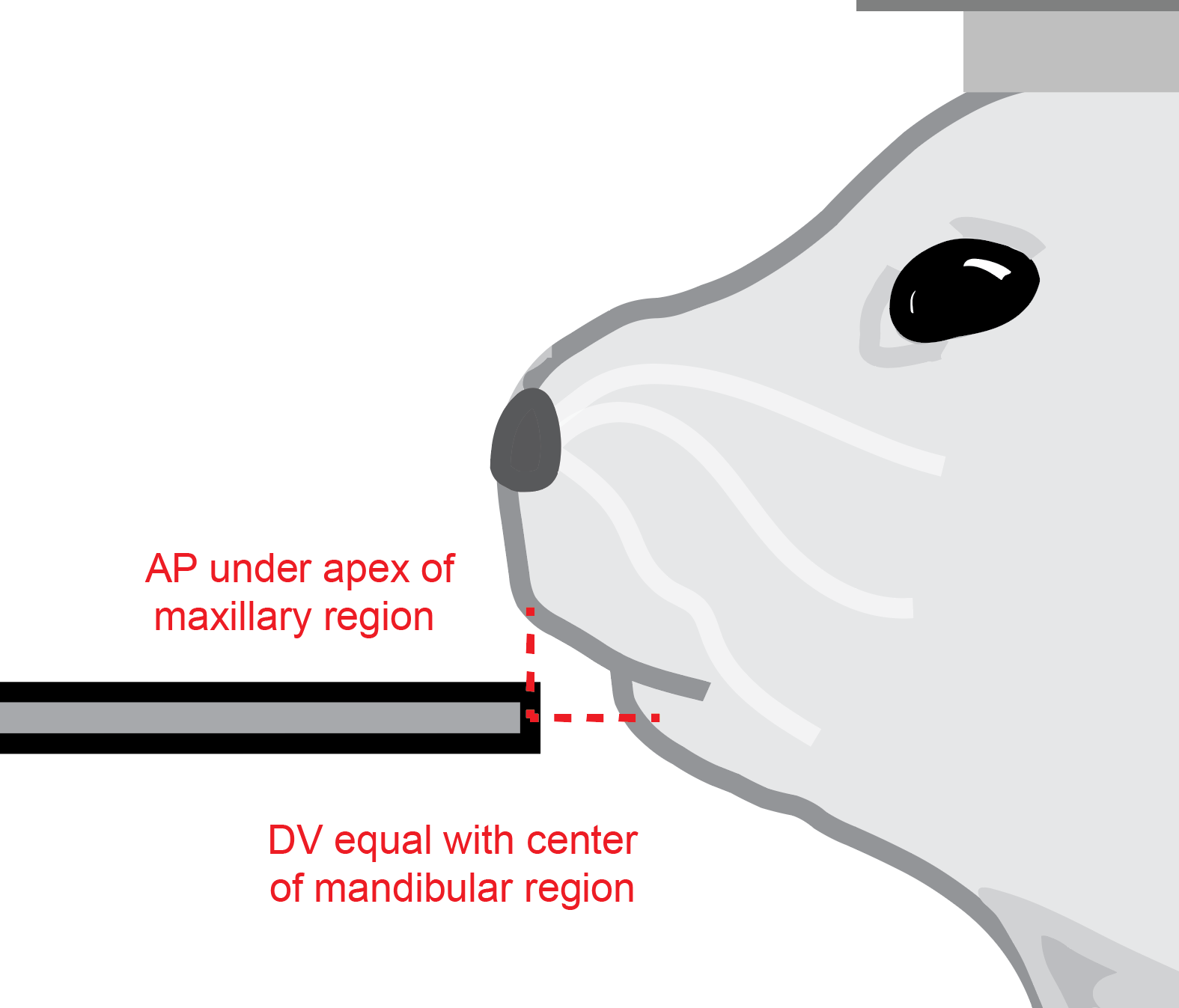


* Set the subject name for the session within the *write\_serial.py*



*Note: See general arduino software document for more details about the functionality of the write\_serial.py program*

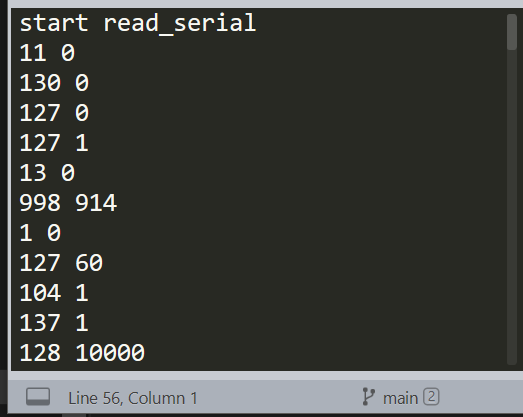
1. After the program and hardware have been tested, head fix the animal.
2. Set the spout position as shown in this diagram:



*Note: See* [*link*](https://www.facebase.org/mouseanatomy/) *for examples of mouse face anatomy*

1. Run the python program (Cntrl + B for Sublime Text 3).

* Values should begin to print with the start of the session, and will continue to print during recorded events.

****

*Note: If you see an error, refer to the general arduino software document for troubleshooting tips.*

1. At the end of the session, the spout will retract (if using a retractable spout) and the python script will print “stop read\_serial”).
2. Remove subject from head-fixation.
3. Restart arduino prior to subsequent subjects by opening / closing the arduino Serial Monitor. Otherwise, the spout will extend for the next subject and could result in poking the subject.

