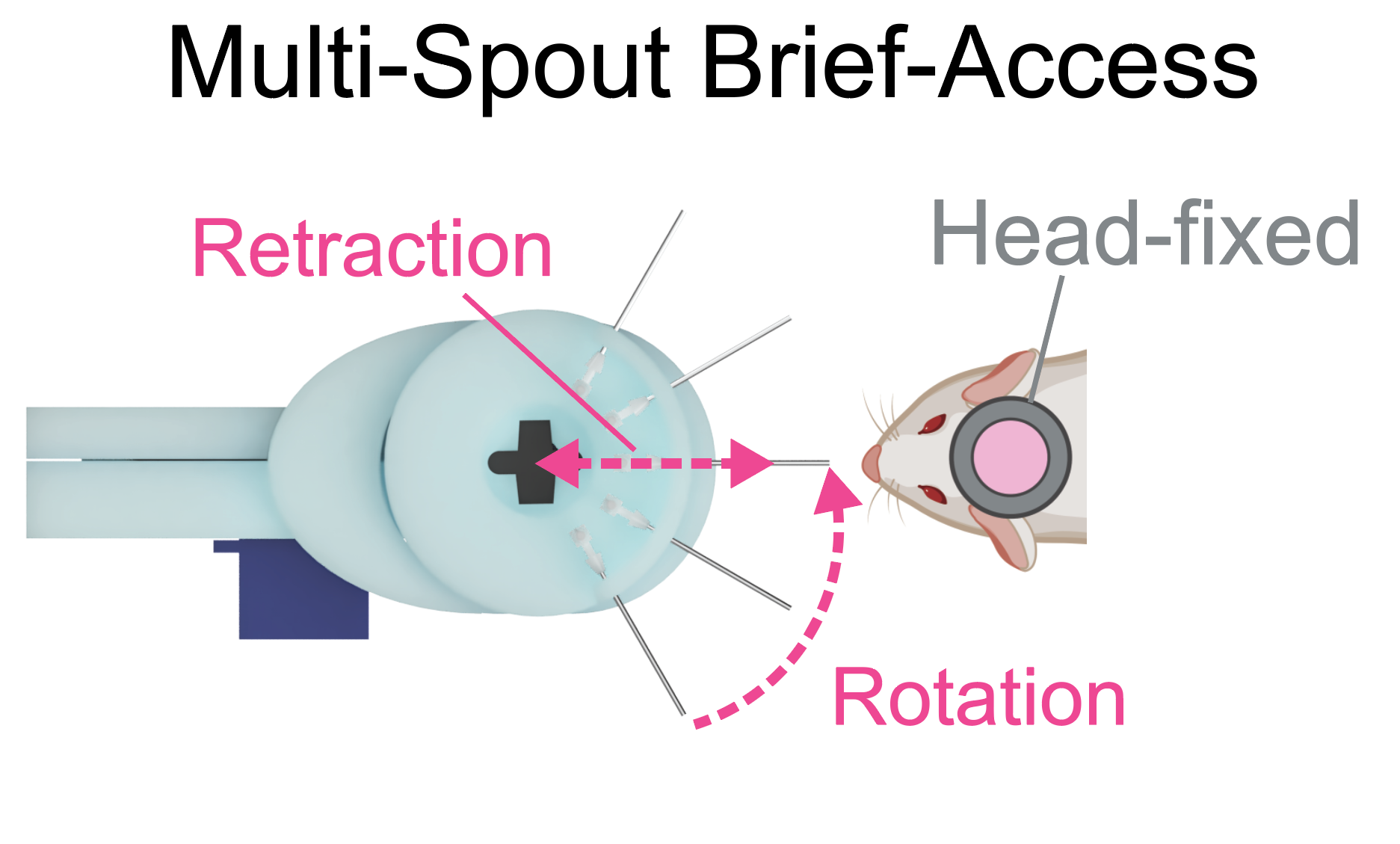
**Protocol for beh\_multispout\_briefaccess.ino**



Program purpose: allows subject fixed windows of access time to one of multiple spouts.

Access windows can include free-access consumption or fixed volume delivery.

**Preparation Instructions:**

1. Around once per week, calibrate the solenoid open duration for each solenoid 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 elements 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 the arduino console to the external hardware.*

**Initial Calibration of Rotation and Extension**

After assembling the hardware, it is useful to perform an initial calibration to determine estimates for the rotation and extension of each spout. These values will serve as a starting point but may be changed slightly at the start of each day.

Graphical user interface

Description automatically generated

1. Use a length of wire and wrap it over a head stage such that the end of the wire sits roughly where the nose of a mouse would be located.
2. Upload *manual\_servo.ino* and open the Serial Monitor.
3. First, fully extend the retractable spout.
   1. Enter the pin for the servo used for retraction and send over serial (default = 9).
   2. Enter 180 for a fully extended starting position.
4. Next, move the servo to determine the rotation angle for each spout.
   1. Enter the pin for the servo used for the radial multi-spout head and send over serial (default = 11).
   2. Rotate the head to the first spout (right most spout from the mouse’s perspective) so it is at a right angle relative to the head-stage. Start with sending a value of 0 over the serial, then progressively increase until the spout is at a right angle. Record this value.
   3. Use the micro-positioner to move the spout so it is in line with the piece of wire and almost touching.

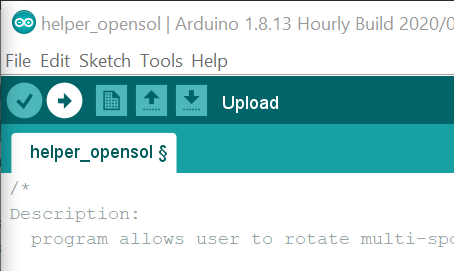
1. For the remaining spouts, send new serial values to change the rotation of the head to align each spout with the tip of the wire and record the angle for each spout (you should have one angle for each spout). With default hardware, the spouts should be roughly 30 degrees apart.
2. To calibrate the extension for each spout
   1. Open the beh\_multispout\_briefaccess.ino program.
   2. Enter the servo\_radial\_degs values obtained from step 4 (format: {#,#,#};, with one # for each spout).
   3. Enter default values for servo\_retract\_extended\_degs ({180,180,180}; with one value for each spout).

**Arduino Setup Instructions:**

1. Open the program
2. Set the parameters and input / output pins

* Carefully read the comments next to each variable

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

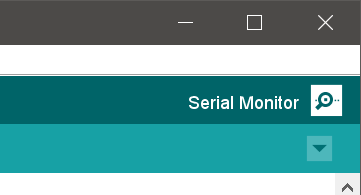


* The arduino will now have the script running on it and it will wait until a start command is sent over serial.

**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

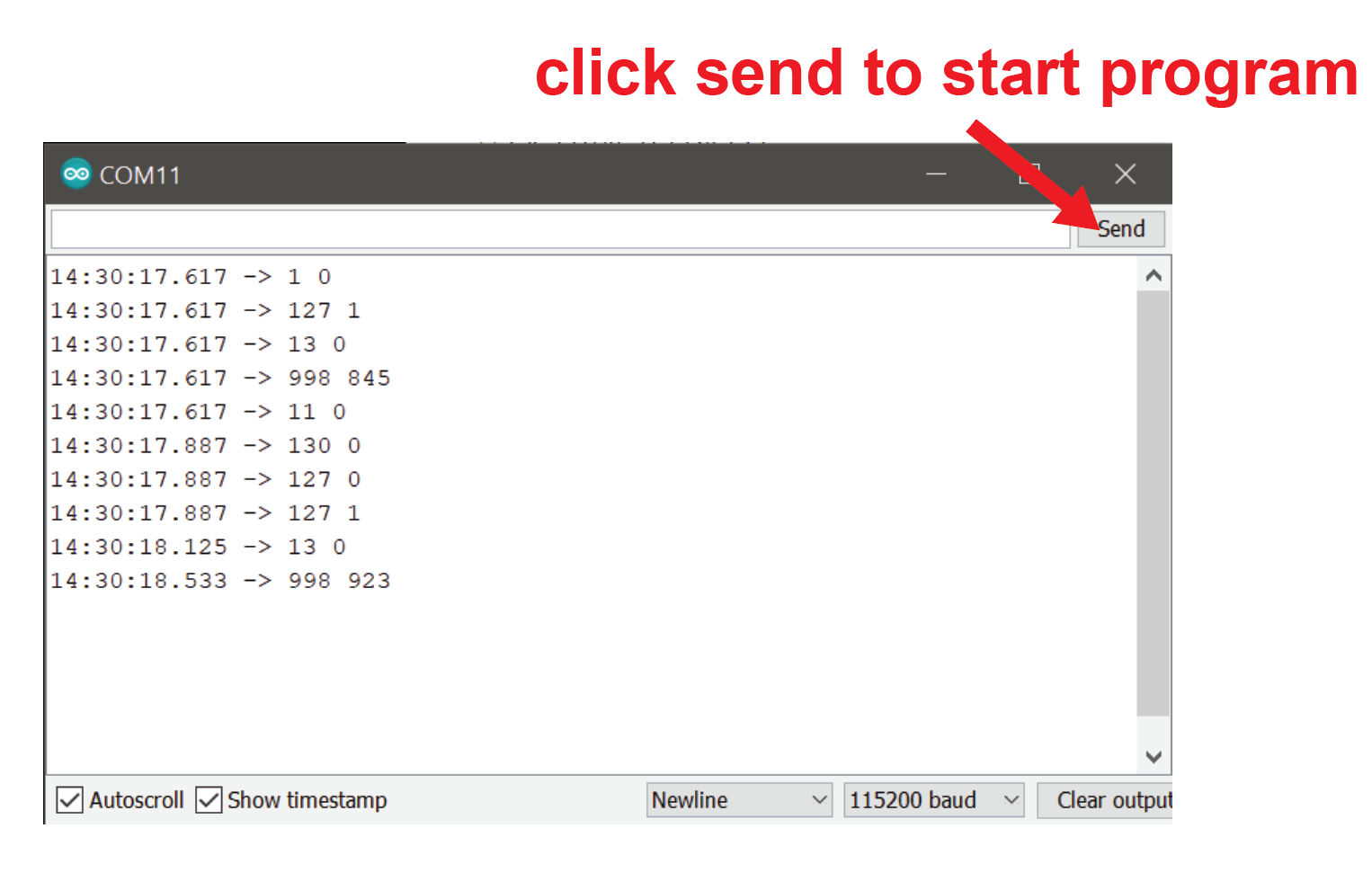


*Note: If you do not see this text printed in the serial monitor, see troubleshooting arduino software document.*

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

### **Note on lick detection with high-molar NaCl**

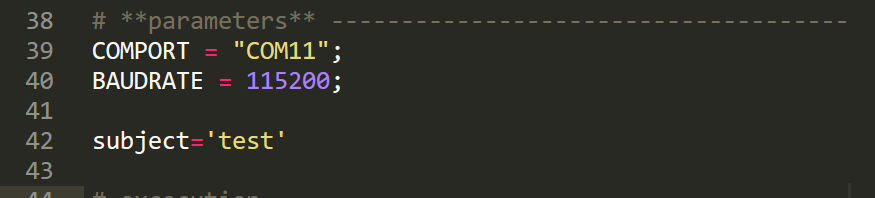
Special care needs to be taken to ensure that the capacitive touch sensor is accurately detecting licks when using high-molar NaCl solutions. Some systems have produced poor detection on specific spouts during mouse licking. After confirming that all hardware works with finger touches, watch subject licking carefully while monitoring lick detection. If there is a problem with lick detection on a particular spout, do not use this spout for high molar NaCl or build a new multi-spout head. Future improvements to the design will be posted in the accompanying GitHub.



1. Watching the Serial Monitor while interacting with the hardware, check to ensure that…
   1. Rotating the wheel in the intended active direction produces code 81.
   2. Rotating the wheel in the intended inactive direction produces code 71.
   3. Touching the spout produces code 30 + current spout (e.g. 31 for spout 1, 32 for spout 2, etc.).
   4. Rotation of the wheel in the active direction that reaches the fixed ratio triggers the intended sequence of events (break, tone, spout extension, liquid delivery or external TTL signal).
   5. If performing progressive ratio, check to ensure that the cost of the reinforcer increases as intended.
   6. Rotation in the inactive direction that reaches the fixed ratio triggers the intended sequence.
2. 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 head, and check to ensure the servo is working properly.
   1. If liquid is emerging from the wrong spout, then check the pinSol vector to ensure that the correct pins are included for your system.
3. Reset the arduino by closing and reopening the Serial Monitor, and then close Serial Monitor prior to the session.

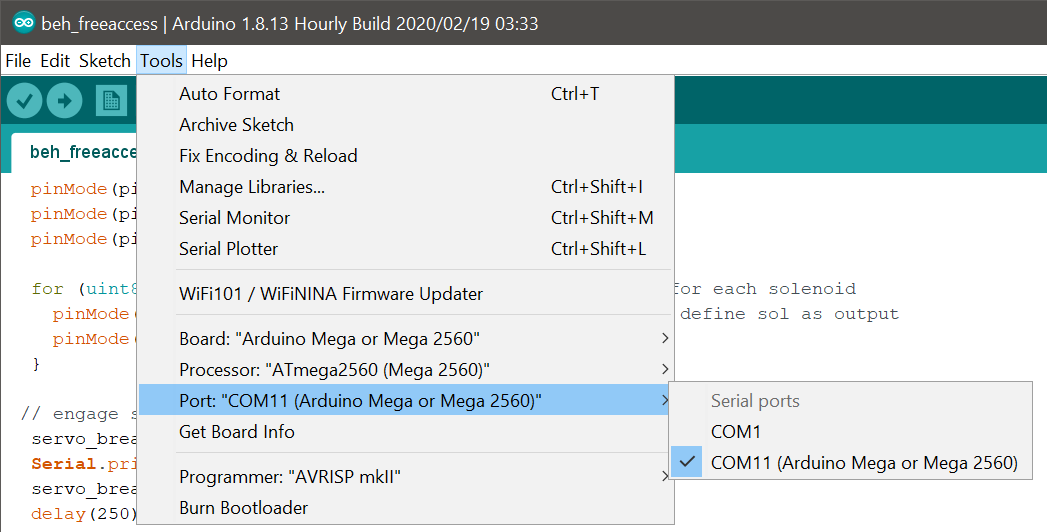
**Arduino Run Instructions:**

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

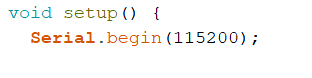


- Set the COMPORT and BAUDRATE to match the arduino script

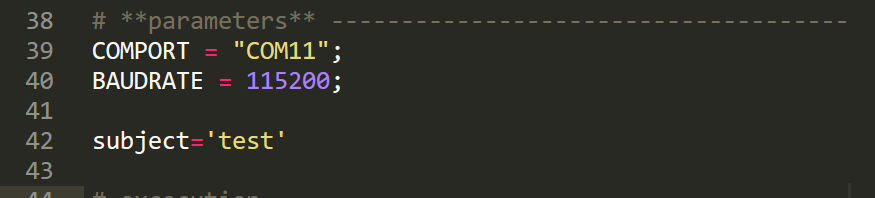
Arduino comport is shown in the tools drop down menu



Arduino baudrate is set with “Serial.begin” call

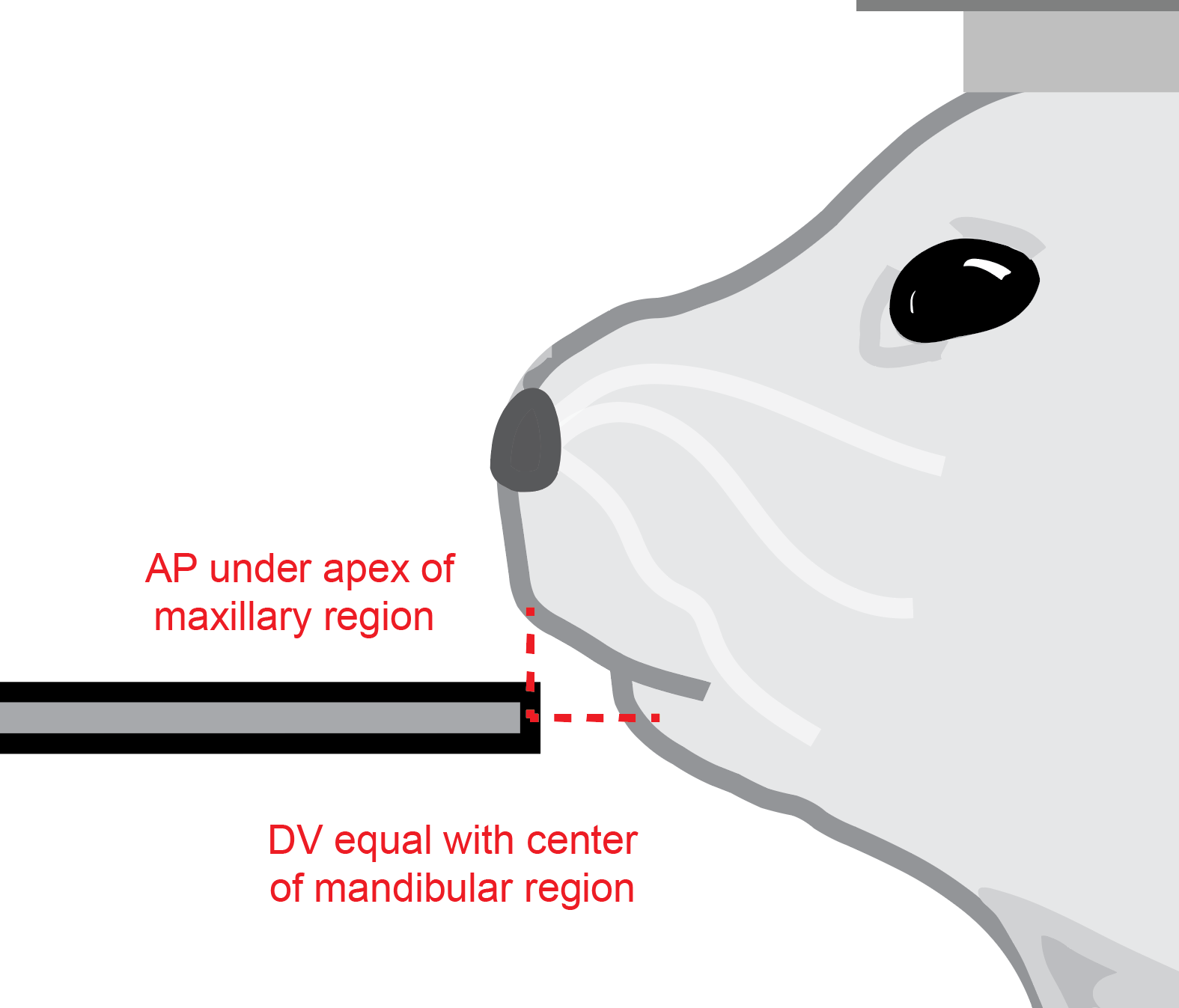


- Set the subject name for the session in the python program



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

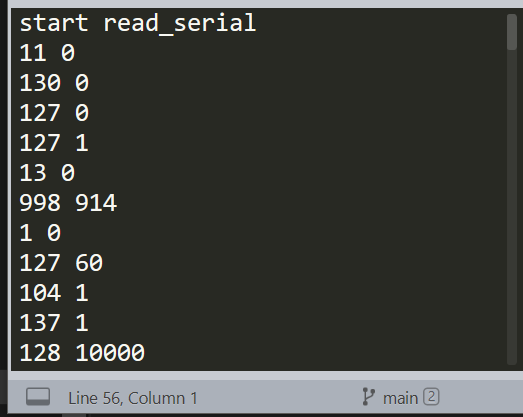
1. After the program and hardware have been tested, head-fix the animal.
2. Set the spout position as shown in this diagram:



*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), the brake will engage, 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.

