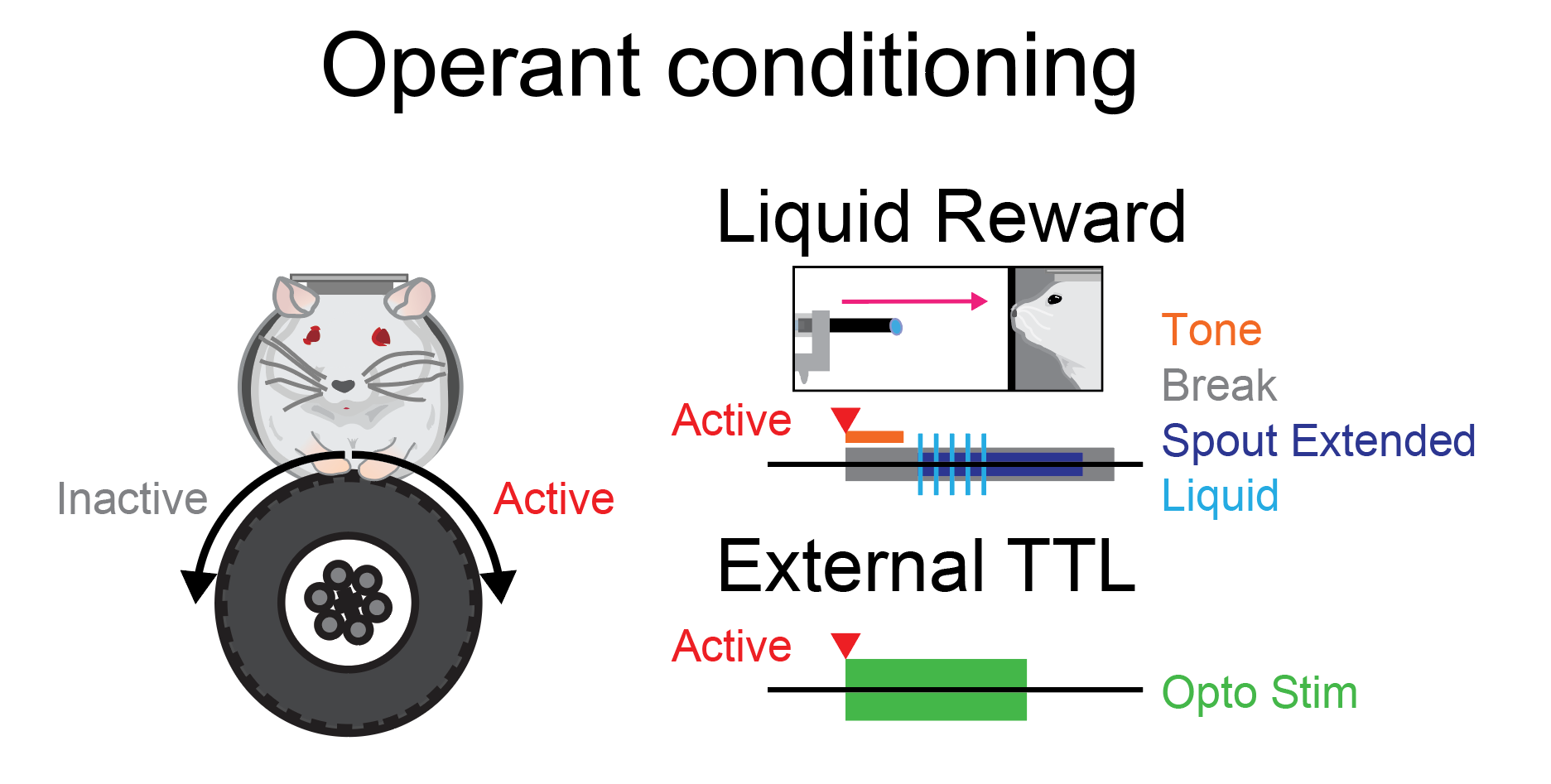
**Protocol for beh\_operant.ino**



Program purpose: Operant conditioning for liquid reward or TTL for using external hardware (e.g. lasers for optogenetic stimulation) with wheel rotation as an operant response. Optional use of a wheel-brake, retractable spout, and tones for both tasks.

**Preparation Instructions:**

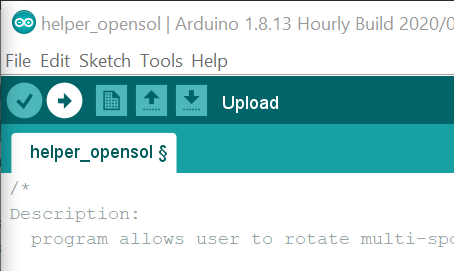
1. Around once per week, calibrate the solenoid open duration by following *protocol\_helper\_calibratesolenoid*. Save the solenoid open duration(s) somewhere convenient in a vector format with 1 value per solenoid (e.g. 3 for 3 spouts: each vector would look like {*value*, *value*, *value*}).
2. Fill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

**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.

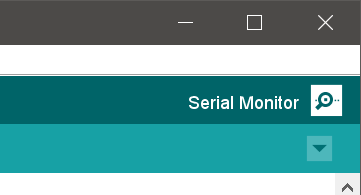


* The arduino will now have the script running on it and it will wait until a start command is sent over serial.

**Arduino Test Instructions:**

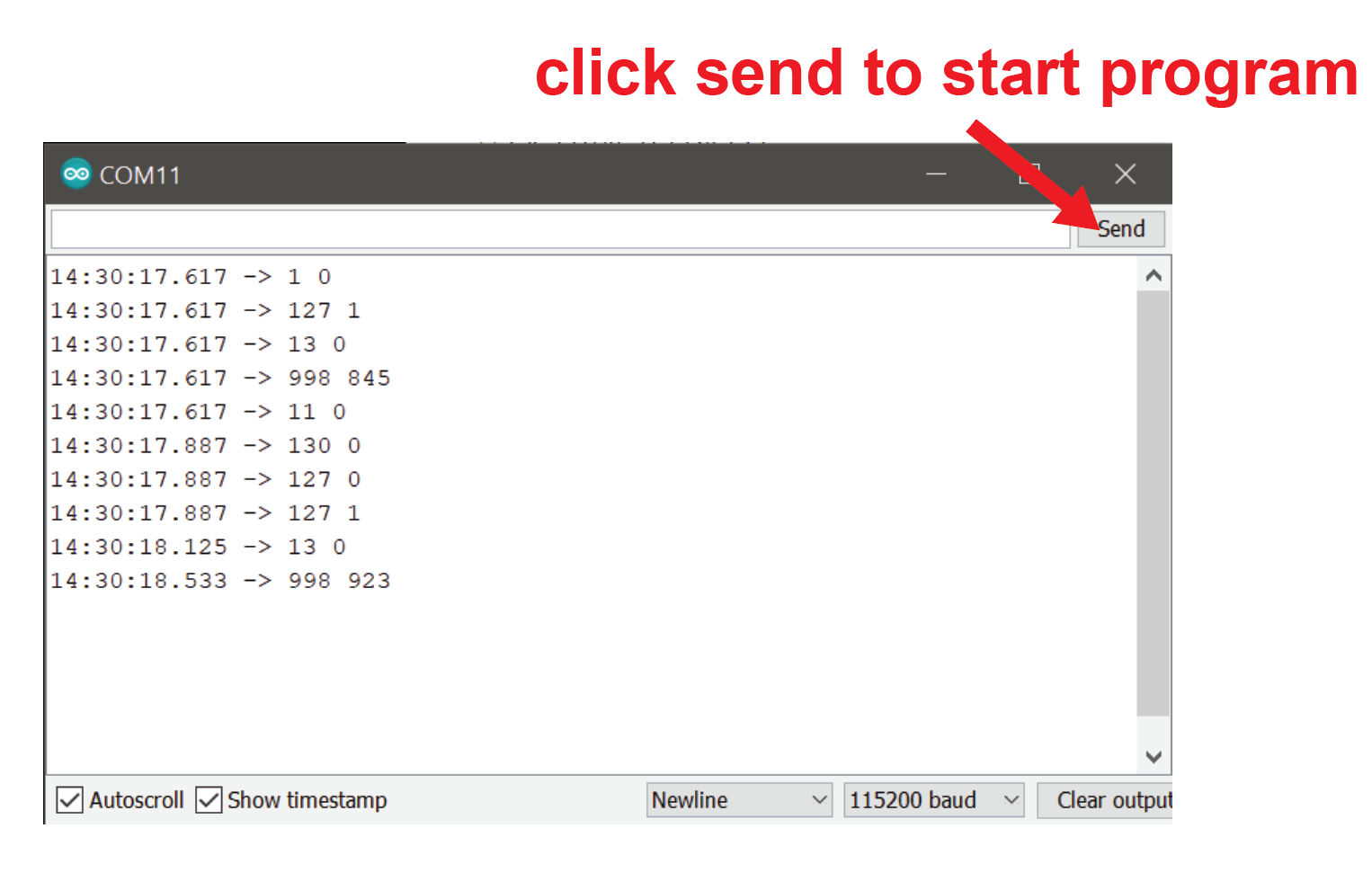
Test the program at the start of each day to ensure all hardware is working and events are being recorded correctly.

1. To test the program, open the 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*

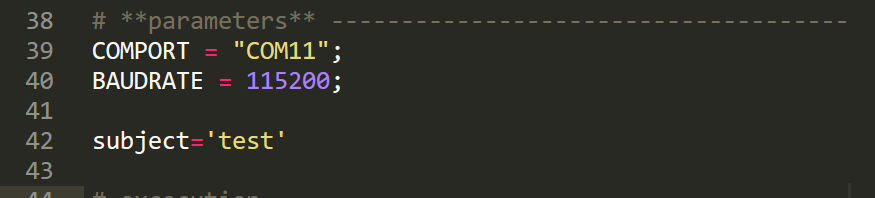
1. Start the arduino program by clicking the button to send an empty string over serial.



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.
   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 multi-spout 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.

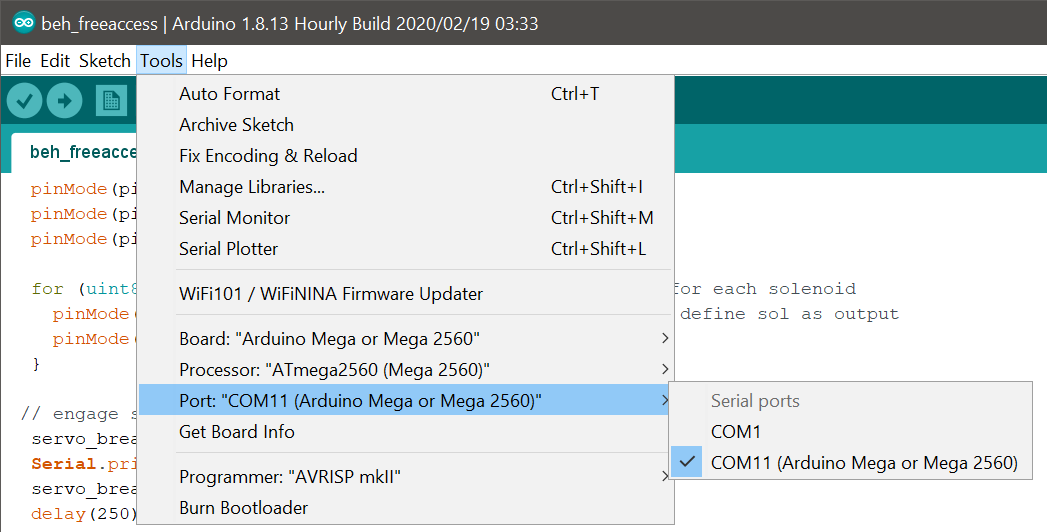
**Arduino Run Instructions:**

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

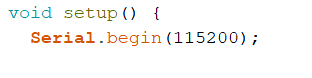


- Set the COMPORT and BAUDRATE to match the arduino script

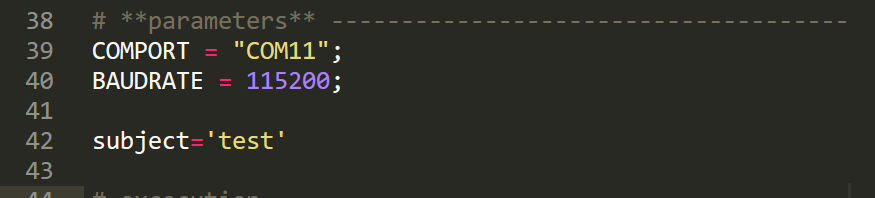
Arduino comport is shown in the tools drop down menu.



Arduino baudrate is set with “Serial.begin” call

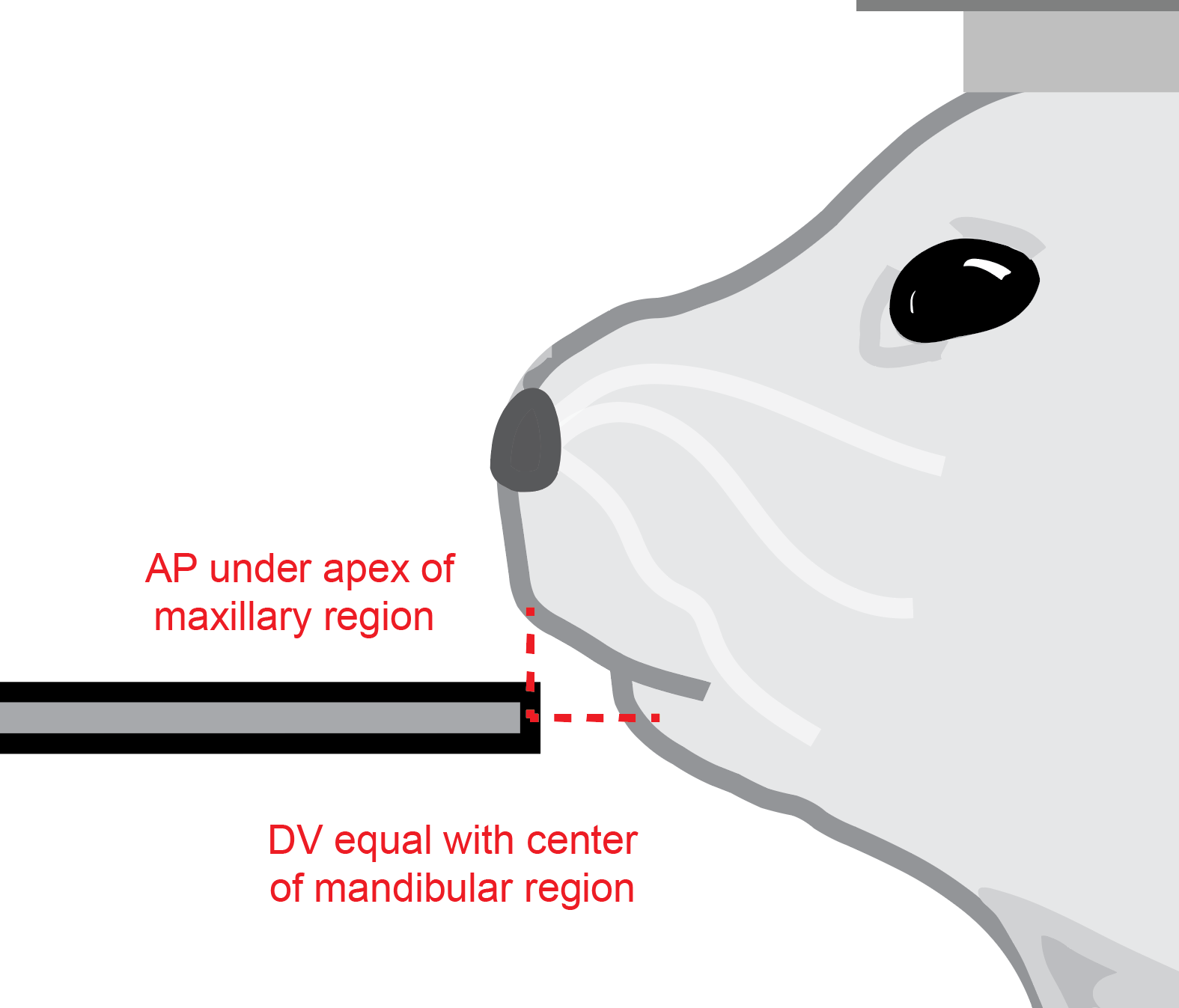


- Set the subject name for the session in the python program



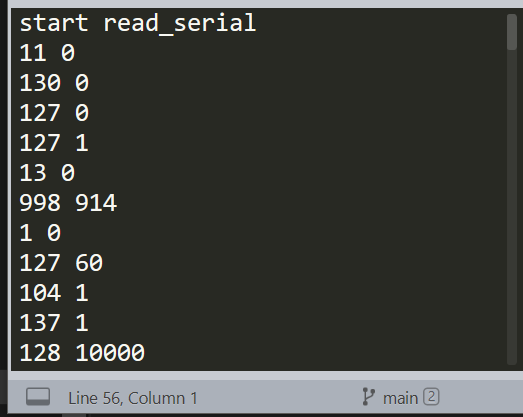
*Note: See general arduino software document for more details about the functionality of the write\_serial.py program.*

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



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 break 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.

