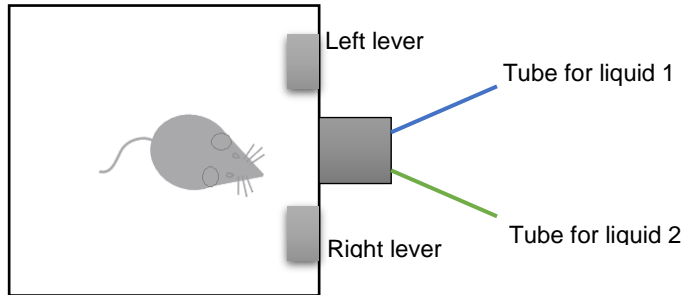


Arduino for Neuroscientists

Session 2 problem set

I. Am Lerner wants to conduct a task in which a mouse can obtain 10% sucrose solution by pressing a lever or sweetened milk by pressing the other lever in the chamber. To deliver the liquid rewards we will need to use two solenoids or an H-bridge like the one used in the OM2 boards. On each lever, there is an IR beam such that beam breaks can be used to track the lever presses. The port also uses an IR beam to track entries into the port.



- a) In Arduino you can do quite a lot without creating functions. However, they can provide an elegant way to execute a set of lines that has multiple uses. To learn how a function is written in IDE read the tutorial linked here. (<https://www.arduino.cc/en/Reference/FunctionDeclaration>). In the skeleton code for this problem, you will find a function called `trackBeamBreak()`. Read the description to see how it is used and then use it for the three IR beams in the program (left, center, right).
- b) I. Am wants to counterbalance the pairings of the levers and rewards such that half of her rats get (liquid 1 – left lever) and (liquid 2 – right lever), and the other half get (liquid 1 – right lever) and (liquid 2 – left lever). Of course, she could do this manually by switching the wires on the board but she would rather have code that allows her to change 1 variable in order to switch the pairings. In the skeleton code, you'll find a function called `openSolenoid()`. To make sure that you understand how it works, use it by itself in the code (comment out these tests before submitting your work). To enable the desired control, you will need a set of "if" statements that calls `openSolenoid()` and a variable that determines which condition to execute. Note that the variable you use for this problem is one that I. Am would change (or not) before uploading the script to the Arduino and it wouldn't change while the animal is performing the task.
- c) Optional: The current code implements what is called continuous reinforcement (CRF) because each press leads to a reward. Another schedule that is common in operant conditioning is the fixed ratio (FR) schedule. For example, FR5 means that after every 5 actions (e.g. lever presses) the reward is delivered. Modify the code so that one of the levers delivers on an FR schedule. You will want to declare a variable that counts the number of lever presses. Also, you may want to borrow some ideas/code from this tutorial (<https://www.arduino.cc/en/Tutorial/StateChangeDetection>).

Submission instructions:

For general info about Arduino IDE commands and syntax, check out <https://www.arduino.cc/en/Reference/HomePage>

Create a zip folder that contains the ino folder(s). Name the zip folder with your first initial, last name, and pset2. For example, "FPena_pset2". Email the zip to me (fxp2102@columbia.edu) by the beginning of the next class.

Tips for troubleshooting:

- Serial.println() is your best friend.
- Seriously, when I first write an Arduino program I hardly use any hardware. Why? There could be issues with hardware that have nothing to do with my program such as a clogged solenoid or a ground wire that became accidentally unplugged or connection on the PCB that was not properly soldered. Those issues are important but they are completely independent of whether the logic of your code accomplishes what you set out to do.
- For this p-set, the code does require input (i.e. IR beam breaks) in order to test whether it works properly, so you do need at least 1 beam break. But you don't need any solenoids. You can just take the prints in the serial monitor as your indication that the appropriate commands were executed under the appropriate conditions (e.g. a solenoid was opened because of a beam break and not for some other reason).

For general info about Arduino IDE commands and syntax, check out <https://www.arduino.cc/en/Reference/HomePage>