

Bottle Based Lickometer to Trigger TTL Output or MedPC Events

From: Janak Lab, Johns Hopkins University

Design by Kurt Fraser

The purpose of this device is to record licks from a small animal (e.g rat or mouse) during the oral consumption of rewards from either a metallic port or bottle with a metallic spout. The capacitive lickometer is run by an Arduino and can be configured to trigger outputs based on licks (e.g. to write licks in as an input for MedAssociates Behavioral Chambers) or to activate a TTL output (e.g. for closed loop optogenetic laser stimulation via a Master9 Stimulus Controller). The Arduino can trigger outputs and the sensor is able to simultaneously record from up to 12 different sources. If the goal is to just record licks, then this is a very cost effective manner of achieving this. The lickometer is high fidelity, flexible, and easy to construct that makes it a superior option compared to commercially available systems, which can be expensive and require use of proprietary software and are inflexible to setup (e.g.

<https://www.noldus.com/phenotyper/lickometer> or <https://www.stoeltingco.com/stoelting-lickometer-system-6792.html>), and simpler than other published custom-made lickometers (<https://www.sciencedirect.com/science/article/pii/S2468067218300105> or <http://brianisett.com/2016/03/26/diy-lickometer/>).

It should be noted that because the lickometer is capacitive that you should design your setup to prevent the recording of non-lick events and limit the possibility of the sensor detecting or being triggered by the animal touching or holding onto the recording spout.

Our setup records all data onto a mounted SD Card that will write each drinking session at its completion. If you do not intend to save data through other behavioral equipment (e.g. through MedAssociates) you will not need a computer to be actively running the Arduino. It is critical to note that files are constructed without naming. As a result, the SD card will need to be removed, the files uploaded and appropriately named on a PC, and the SD card replaced between every drinking session.

1) Parts List – everything is easily acquired from Amazon

Expected Parts Cost Per Lickometer: \$75~\$100

1x Arduino Uno R3

Cost: \$23

Link: <https://www.amazon.com/Arduino-A000066-ARDUINO-UNO-R3/dp/B008GRTSV6>

1x Adafruit MPR121 12-Key capacitive sensor breakout board

Cost: \$12

Link: <https://www.amazon.com/Adafruit-12-Key-Capacitive-Sensor-Breakout/dp/B00SK8PVNA>

1x SD Card Shield compatible with Arduino Uno R3

Cost: \$14

Link: <https://www.amazon.com/Seeed-sd-card-shield-Arduino/dp/B00KAE24PA>

1x 9V AC/DC Power Source**Cost:** \$9**Link:** <https://www.amazon.com/Adapter-Arduino-Exercise-Elliptical-Recumbent/dp/B06Y1LF8T5>**1x 16-32 GB SD Card****Cost:** \$8**Link:** <https://www.amazon.com/SanDisk-Ultra-Class-Memory-SDSDUNC-032G-GN6IN/dp/B0143RT8OY>**Jumper Cables****Cost:** ~ \$8 for more than you need**Link:** <https://www.amazon.com/Multicolored-Breadboard-Dupont-Jumper-Wires/dp/B073X7P6N2>**Breadboard****Cost:** ~ \$8 for 3**Link:** <https://www.amazon.com/gp/product/B01EV640I6/>**Female BNC adaptors w/screw terminal block (if planning to use a Master9 or other BNC-Based Stimulus controller for optogenetics)****Cost:** ~ \$12 for 5**Link:** <https://www.amazon.com/gp/product/B00P5AR94A/>**Soldering Iron****Solder****BNC Cables****Bottles/Cage/Setup for actual use (we use Allentown cages and bottles)****Electrical Tape****2) Assembling the lickometer***Estimated Assembly Time: 1-2 hours*

1) Connect the SD Card Shield to the Arduino Uno R3 by simply inserting the pins of the shield directly into the Arduino UNO. Please note there are jumpers on the Arduino UNO ICSP pins (the sharp pins sticking up between A5 and digital 0 opposite the USB and DC power connectors) that need to be removed and placed on top of the pins on the SD shield that the Arduino ICSP pins connect to. You can also insert your SD Card at this time.

2) To assemble the MPR121 we will need to solder the pins first. Insert an appropriately cut and insert the correct number of pins (7) into the breadboard. For the wiring of the sensors, the amount of pins you will solder depends on the number of sensors this chip will control. Insert

those into the breadboard at this time as well as one pin into the ground portion of the sensor. The shorter side of the pins will be soldered into the chip and the longer length will be inserted into the breadboard. (There is a great overview of the assembly for the MPR121 from Adafruit <https://learn.adafruit.com/adafruit-mp121-12-key-capacitive-touch-sensor-breakout-tutorial/overview>)

3) Solder the pins carefully so that they are well connected. (Apply small amount of solder to solder tip, heat joint of pin to board for 2-3 seconds, then introduce solder. See this excellent solder guide <https://learn.adafruit.com/adafruit-guide-excellent-soldering/making-a-good-solder-joint>)

4) Connect the MPR121 to the Arduino following the wiring guide from Adafruit. The VIN wire will be connected to the 5V, SCL to A5, SDA to A4, and GND to GND on the Arduino along its Digital side.

5) Prepare the BNC connector at this time if you intend to use the Arduino for optogenetic approaches. Strip two ends of jumper cable and screw into the + and – (+ is the power and – is the ground) of the BNC block. Ensure that these are tight. Connect the – wire to the ground of the breadboard and the + to the pin on the Arduino we intend to have used as the dedicated TTL generator (we chose digital pin 8 for this purpose).

6) Attach your sensor cables to their intended sensing source. For our purposes we stripped jumper cables and wound the wire ends around a metal sipper bottle top and then well insulated the bottle top with electric tape so that only the tip was exposed. Make sure that these are conductive sources and, once connected, insulate appropriately.

3) Testing the lickometer

1) The code to operate the sensor, write data to an SD card, and produce a TTL pulse are located at <https://github.com/kmfras/lickometer>. Make sure that the MPR121 library is installed on the PC you will be using to upload to the Arduino. We find that first testing with the included MPR121 test code works best to confirm proper wiring as it can show when sources are touched and released.

2) Open the Arduino IDE, open the lickometer code, and upload. If all was successful when you open the terminal on the computer you will see the Arduino boot up and then with every lick/contact you will see a brief flash on the MPR121 sensor. Most common problems we encounter came from bad soldering on the MPR121 sensor or issues with the insulation on the sensors.

3) For some situations the default properties of the MPR121 may cause a high rate of false positives or even not detect. This is because of the nature of the capacitive sensor. You can alter the current applied and the minimum amount of voltage change to count as a response in the MPR121 library CPP file titled Adafruit_MPR121. The code, setThresholds defines the amount of change to call a sensor on, and the amount of change from on to become off. The current can be altered in the writeRegister(MPR121_Config,) code. See documentation on the sensor for more information on how to alter these properties. Save the changes to the CPP file and reupload the code to the Arduino to implement these changes.

4) Lickometer Operation

1) Every time the Arduino is plugged in it will create a file on the SD Card with a random title, but no information of the date or time that it was run. It is important, then, to remove, rename, and store the data after every subject/operation of the Arduino.

2) Other than that stipulation all you'll need to do is connect your sensors, warm up your lasers, set up your stimulation parameters on you Master9, plug in the Arduino, and get ready for the data to come in!

5) Lickometer Data

1) Data from the lickometer is stored as a text file with every instance of an event recorded with the Arduino local time in the style TIME, EVENTNAME, . This is a CSV format, but will need to be extracted and converted to the format you choose for analysis.

2) We have provided code that allows for the extraction of lick events and the sorting of lick events into bursts and clusters for microstructural analysis using MATLAB. Code can be found at <https://github.com/kmfras/lickometer>