Lick-activated water dispenser

Clara Hartmann and Mahesh M. Karnani

Department of Integrative Neurophysiology
Center for Neurogenomics and Cognitive Research
Vrije Universiteit Amsterdam
De Boelelaan 1085
1081 HV Amsterdam
The Netherlands

General Description

The following build instructions are for a lick-activated water dispenser using cost-effective off-the-shelf components, useful for behavioural neuroscience experiments on rodents (Figure 1). It is controlled by a Raspberry Pi computer that logs licking and regulates water availability according to a user-defined Python script. Our example code saves data in a small .csv file containing a list of time stamps (at 5 μs precision using the PiGPIO library[1]) for water drop availability, the ensuing first lick contact and total licks between consecutive drop availability moments. A user with beginner skills in Python programming can easily modify the code. The water dispenser can be easily implemented in any home cage or behavioural setup, as only a 4 mm diameter hole is needed to pass the licking spout into the animal enclosure.

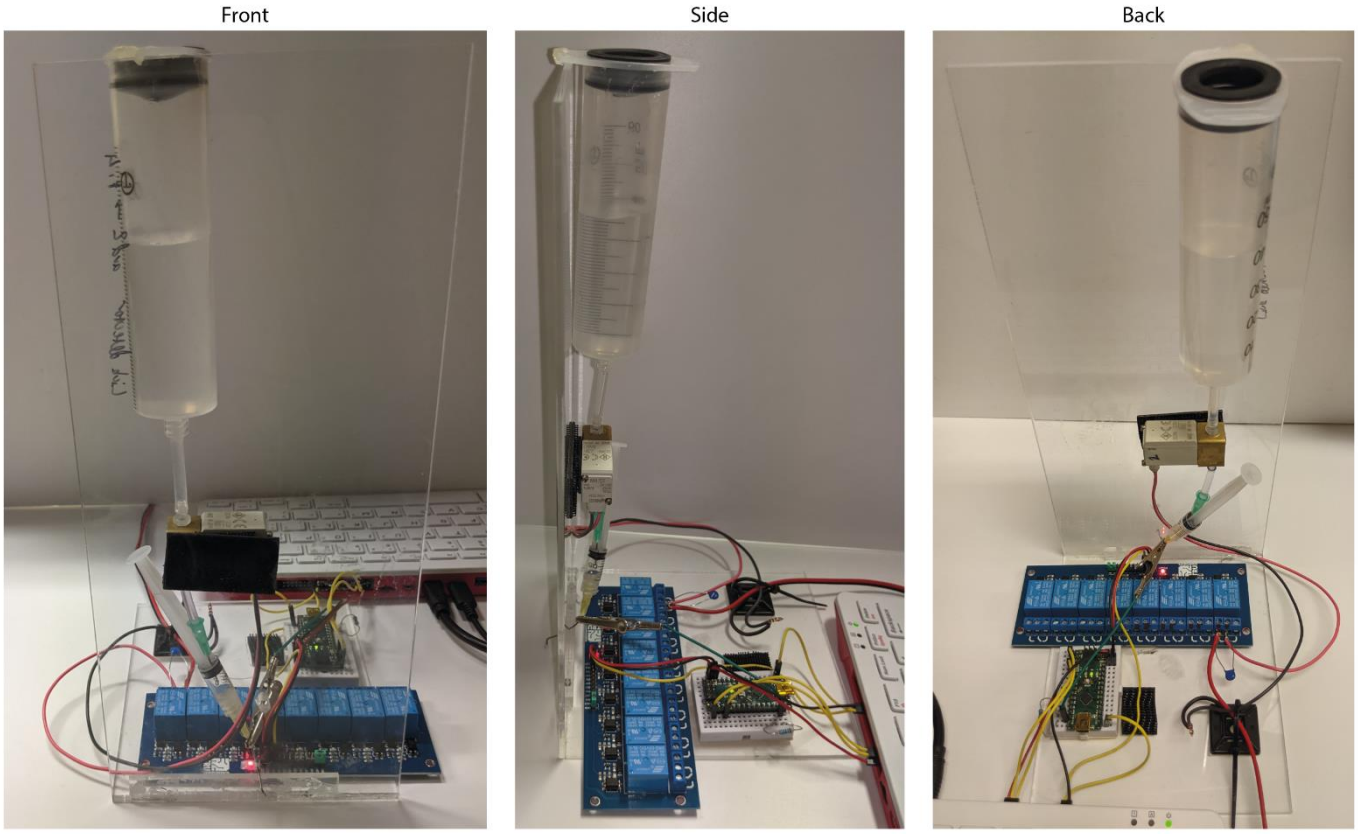

Figure 1, Front, side and back views of a lick activated-water dispenser built according to the instructions here. Exceptionally in this build, we used one channel in an 8-channel relay module (blue) instead of the single channel component listed below.

Materials

Table 1, Bill of materials for the lick-activated water dispenser.

Part	Count or length	Manufacturer	Manufacturer serial # or *.stl file	Approximate cost, EUR
Arduino nano	1	Arduino	A000005	19.9
Raspberry Pi	1	Raspberry Pi	KW-2646	45 – 95.0
Relay module	1	TRU Components	TC-9927156	5.0
Varistor 25VAC	1	TDK	B72205S0250K101	0.26
Resistor 10MOhm	1	Vishay	594-CBB0207001003GCT	0.6
Power source 24V	1	RS-PRO	175-3306	10
Solenoid valve 24V	1	SMC	VDW12GA	17.2
50 ml syringe	1	Terumo	SS 50L1	5.4
2.5 ml syringe	1	Terumo	SS 02LE1	0.5
21 G hypodermic needle	2	Terumo	AN*2150R1	0.1
3 mm (ID)/ 5mm (OD) silicone tubing	100 mm	Saint Gobain	760210	1.5
M5 threaded nylon barbed tube fitting	2	McMaster-Carr	5463K557	1.2
Alligator clip	1	RS Pro	738-5856	2.8
wire	500 mm	RS Pro	196-4225	0.2
Female to female jumper wire connectors	9	MikroElectronika	MIKROE-511	4.3
Male to male jumper wire connectors	8	MikroElectronika	MIKROE-513	4.3
Breadboard	1	Bud industries	BB-32650-B	3.30
			Approx. total cost: 122-172 EUR	

Useful tools: pliers, wire cutters, metal file, soldering iron, drill. The components can be mounted e.g., using Velcro tape.

Mechanical assembly

The water reservoir - a 50 ml plastic syringe - is connected, using 3 mm ID, 5mm OD silicone tubing and a M5 threaded nylon barbed tube fitting (McMaster-Carr 5463K557), to a solenoid valve. From the valve, 30 mm of the tubing is connected to a 21 G needle (forcing the 5mm OD tube inside the 4mm ID luer lock forms a strong junction) which is inserted through the plunger and rubber stopper of a shortened 2.5 ml syringe so that upon valve opening water will drop into an isolated small reservoir in the syringe (Figure 2). This smaller syringe acts to isolate the large reservoir of water from the water inside the lick spout which is sensed during capacitive measurement. The 2.5 ml syringe itself is fitted with a blunted, bent 21 G needle, which functions as the licking spout. An alligator clip connects to the licking spout to sense capacitance changes during licking. To ensure that there is no contact between the spout and the maze itself, the middle of the needle can be insulated with 20 mm

of rubber tubing, leaving the tip of the spout exposed. The spout is bent downward so hanging drops do not spread along the spout changing its capacitance.

The Lick sensor/water spout can be mounted to the outside of a behavioural enclosure wall and the spout is fed through a 4 mm hole into the enclosure. The rubber stopper of the 50 ml syringe can be used as a dust cap if a small hole is made in it to prevent pressure build up.

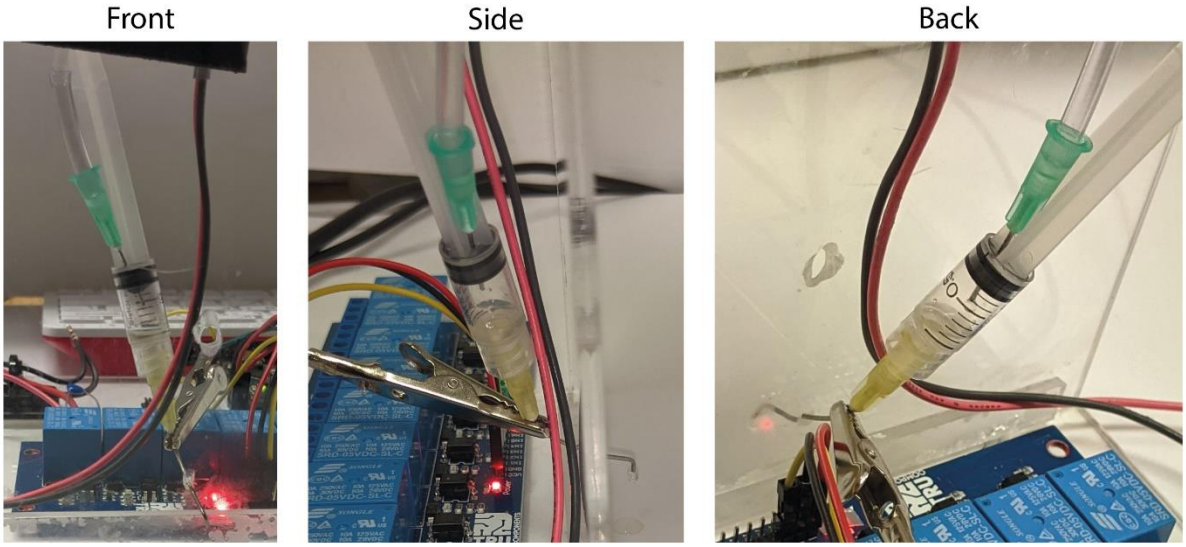

Figure 2, Front, side and back views of the isolated small reservoir sensed by capacitive measurement.

Electronics

The 24 V solenoid valve (SMC VDW12GA) is attached to a 24 V power source through a relay module (TRU components TC-9927156) and a varistor (S05K25), such that the varistor and the switch are in parallel on the positive lead and the negative lead is attached to the valve. The relay module is wired to a TTL input from a Raspberry Pi.

Capacitive sensing is done using an Arduino Nano board, a 10 MΩ resistor connected across pins D10 and D11 and a cable connecting with alligator clip to the lick spout on D11. The Arduino is programmed with a script based on the capacitive sensing library[2]. The Arduino sends a TTL to the Raspberry Pi which runs a Python script that logs licks and sends TTL pulses to the relay module to dispense water.

Connections are made according to Figure 3.

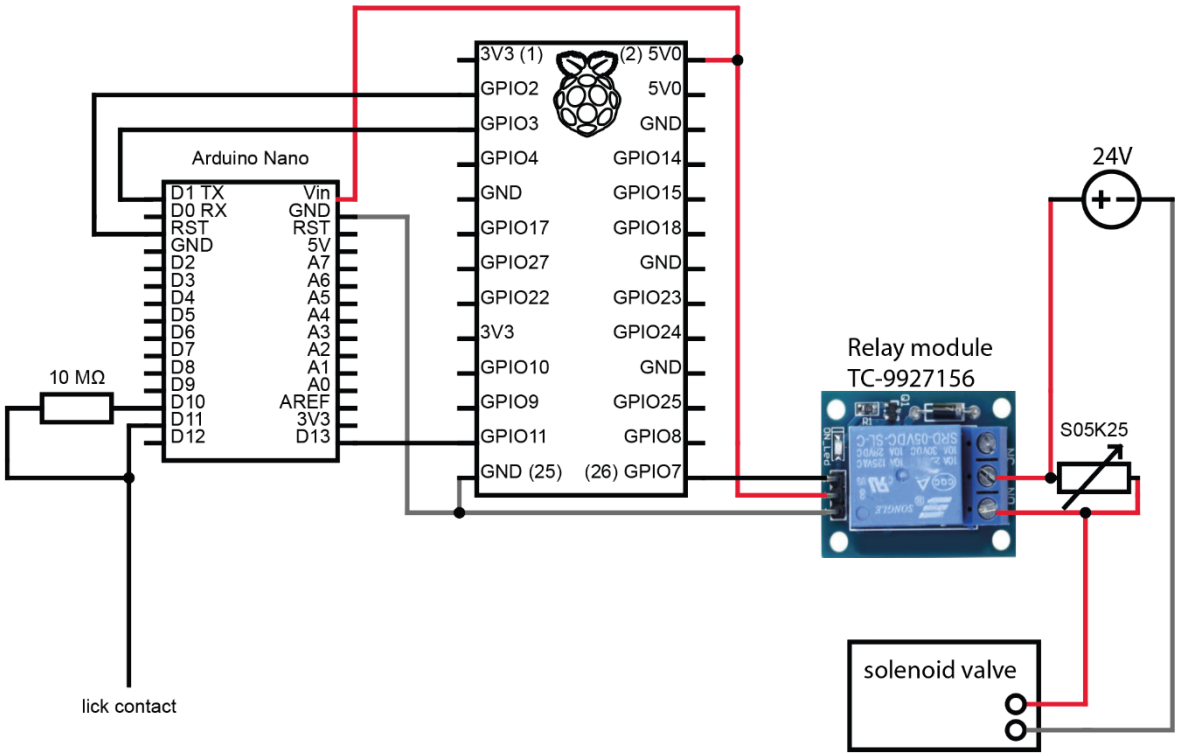

Figure 3, Wiring diagram.

Code and software set up

An Arduino board is set up for capacitive sensing and a Raspberry Pi to log licks and dispense water when desired. Software and code is set up following these steps:

1) On the Raspberry Pi (we have used models 400 and 4b) with the operating system installed (Raspberry Pi OS, released 21-02-2023), Arduino IDE (version 1.8.19) is used to install the CapacitiveSensor library [2].
2) The Arduino code *generic_lick_sensor_arduino_nano.ino* is downloaded from [3] or supporting files of this document and flashed to the Arduino Nano. Arduino Uno and Mega boards also work.
3) For setting up Python (Python 3.9.2 is preinstalled on the Raspberry Pi OS), Pandas and Numpy libraries are first installed via the terminal (e.g. `sudo pip3 install numpy` and `sudo pip3 install pandas`). The *requirements.pip* file is downloaded from [4] or supporting files of this document. After navigating in terminal to the folder containing the *requirements.pip* file, all listed requirements are installed by typing `pip3 install -r requirements.pip`.
4) The helper script *Lick_sensor_functions.py* and the main script *Generic_lick_sensor.py* are downloaded from [4] or supporting files of this document and placed in the same folder.
5) The capacitance threshold is determined empirically by enabling the serial monitor in Arduino IDE and reading out the values while touching the licking spout with a small metal object like a pair of tweezers to simulate licking contact. A reasonable threshold is determined and added to the Arduino code *generic_lick_sensor_arduino_nano.ino* on line 16 as *threshold_increment*. The code is then flashed onto the Arduino again.
6) The PiGPIO daemon is launched from the terminal (`sudo pigpiod`).
7) Now the main script can be run via the terminal or an IDE (we use Thonny Python IDE).

Adjustments and troubleshooting

The user can change parameters like how long the valve should remain open, how long water is unavailable and where to store data in *Lick_sensor_functions.py*. The size of the dispensed amount of water can be changed from the *water_time* variable on line 15 or by creating a tight bottle-neck in the water line, e.g., by introducing indentations along the drinking spout using a pair of pliers.

As the length of cable connecting to the drinking spout and the noise in each system will vary, the capacitance sensing typically needs some trouble-shooting in the beginning. One should focus on electrically isolating the sensing cable, alligator clip, drinking spout and the small water reservoir within the small syringe; ensuring the tubes are connected properly, do not leak and are dry on the outside; changing the sensitivity value on lines 23 and 32 and/or threshold on line 16 of *generic_lick_sensor_arduino_nano.ino* and/or changing the 10 MΩ resistor to one with another value.

References

1. joan2937/pigpio. https://github.com/joan2937/pigpio.
2. Stoffregen, P. *CapacitiveSensor Library*. (2023).
3. MaheshKarnani. SensingWaterDispenser_arduino. https://github.com/MaheshKarnani/Switch_maze/tree/main/Modules_SM/SensingWaterDispenser/generic_lick_sensor_arduino_nano (2023).
4. MaheshKarnani. SensingWaterDispenser. https://github.com/MaheshKarnani/Switch_maze/tree/main/Modules_SM/SensingWaterDispenser (2023).