Supplementary Material

(Revised Version for the v1.3 Printed Circuit Board), *by MJY Zimmermann*

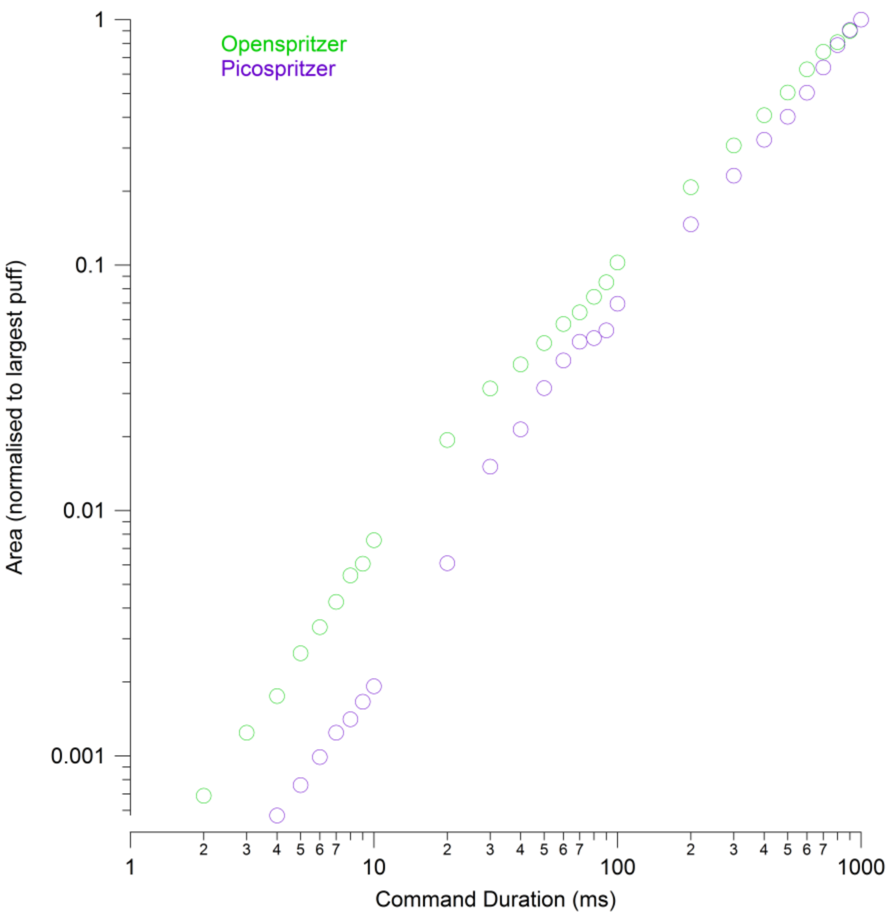
Openspritzer: an open hardware pressure ejection system for reliably delivering picolitre volumes

Forman, C.J.[1,4§], Tomes, H.[2], Mbobo, B.[2], Baden, T[1,3,4§]., Raimondo, J.V.[2,4§]

Affiliations:
1) School of Life Sciences, University of Sussex, United Kingdom
2) Division of Physiology, Department of Human Biology, Neuroscience Institute and Institute of Infectious Disease and Molecular Medicine, Faculty of Health Sciences, University of Cape Town, Cape Town, South Africa
3) Institute for Ophthalmic Research, University of Tuebingen, Germany
4) TReND in Africa gUG (www.TReNDinAfrica.org)

§Correspondence: C.F.: chrisforman@cantab.net, T.B.: t.baden@sussex.ac.uk, J.V.R.: joseph.raimondo@uct.ac.za

Command duration is correlated to dosage

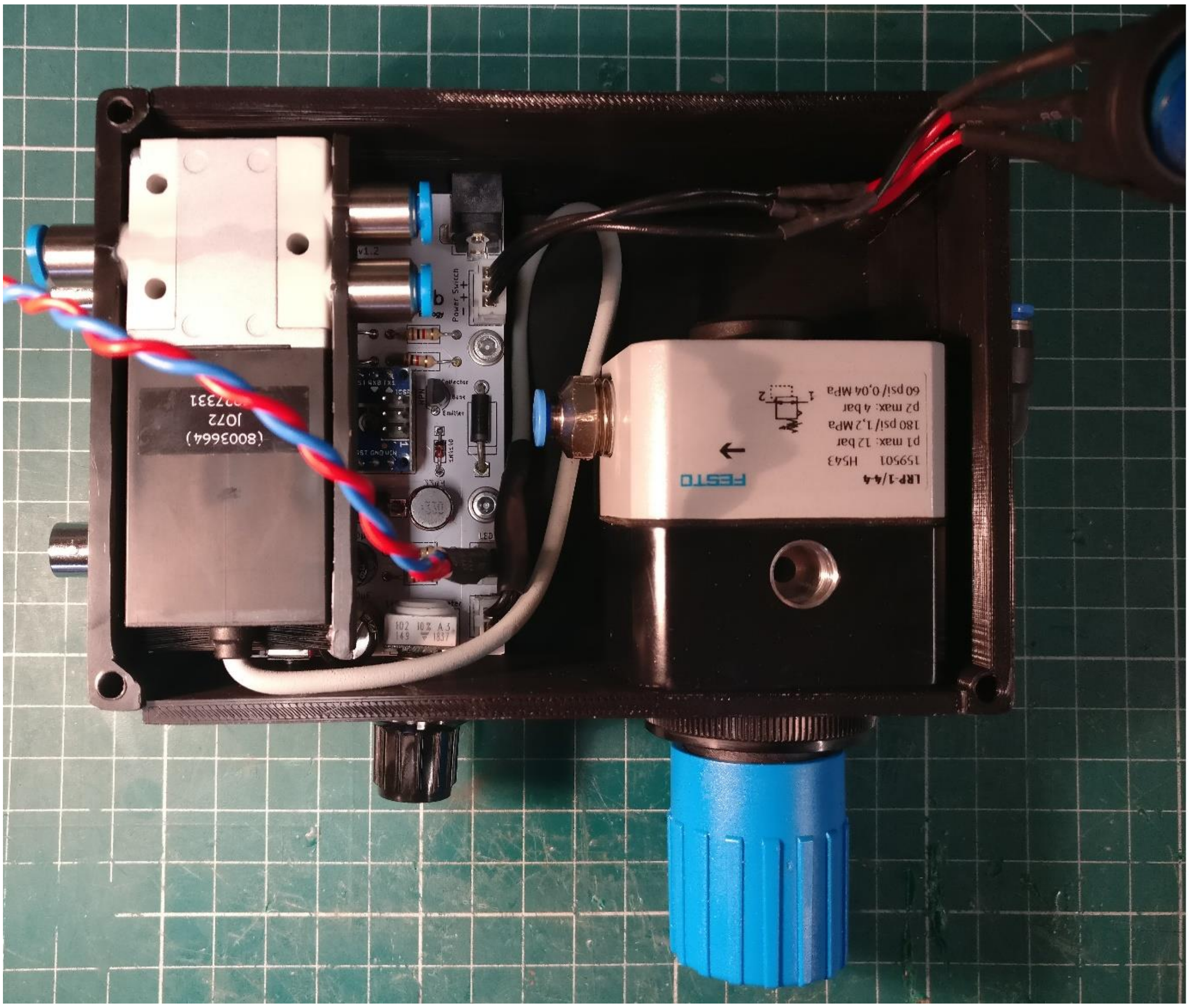


Supplementary Figure S1: The integrated area for each puff plotted against the command duration. The area under the puff profile is proportional to the dosage. Command duration and dosage are strongly correlated, regardless of the details of the puff profile over time.

Overview

This document contains detailed assembly instructions and an operation manual for Openspritzer and also includes a parts list. The Arduino code and 3D printing files for the exterior casing are provided and may be downloaded at https://github.com/BadenLab/Openspritzer. The purpose of the device is to regulate the pressure and duration of a puff of compressed air. Typically, the output port is connected to a glass puffer pipette which has been drawn into a sharp point with a narrow (<2-3 µm) diameter pore.

The device consists of a circuit board, a solenoid valve, a pressure regulator with a gauge and various interface components.



Supplementary Fig. S2: The device is shown without its covering lid. The pressure regulator is mounted on the right and attached to the front panel *via* a retaining threaded ring that comes with the regulator. The printed circuit board (PCB) is mounted onto attachment points on the bottom of the box. There is a platform supports on which the solenoid can rest above the circuit board and a LED can be attached to a mount-point on the lid.

Parts List

Supplementary Table 1 has an exhaustive list of parts, suppliers and prices, allowing calculation of the total cost of parts, which is between £320 and £360 at submission, depending on how much of the full system one chooses to include. Much of the circuitry e.g. the Arduino, footswitch and toggle switch is purely for convenience. A core functional system consists of the regulator, solenoid and a much simpler control circuit under TTL pulse control. The simpler minimal circuit is indicated in the construction instructions below and consists of the branch of the circuit containing the transistor.

In addition, prices are highly variable depending on local availability of components. For example, the single most expensive component when we assembled ours was the regulator which cost £180 at the time. However, we subsequently found the same regulator for $43 in a different country. If one is prepared to shop around, we estimate that a functional Openspritzer without add-ons could be constructed for as little as £175. If one has all the components it takes approximately 1 hours to construct a full Openspritzer depending on experience and excluding the print time for the case and lid.

Frequently components only come in bags of 10 or 50 so if one only intends to fabricate a single device there may be an additional overhead of extra parts, depending on the flexibility of your supplier. For example, ours cost £360 for the parts actually included in the device, but we paid £495 in total and now have some spare components for additional units.

Name	Manufacturer	Order ID	Retailer	Quantity	Observation 1	Observation 2
Solenoid (either 2 ms or 7 ms model)	FESTO	MHE2-M1SH-3/2G-M7-K-RR	Kiowa	1	Pneumatics	2 ms response + cable
Precision pressure regulator	FESTO	LRP-1/4-4	RS	1	Pneumatics	0.05 to 4.0 bar
Gauge	FESTO	MAP-40-4-1/8-EN	Kiowa	1	Pneumatics	0.0 to 4.0 bar, 1/8 inch thread
Push-in (6 mm) to thread (1/4")	FESTO	QSL-1/4-6	RS	1	Pneumatics	6 mm to 1/4 inch
Push-in (4 mm) to thread (1/4")	FESTO	QS-1/4-4	RS	1	Pneumatics	4 mm to 1/4 inch
Push-in (4 mm) to thread (M7)	FESTO	QSM-M7-4-I	RS	2	Pneumatics	4 mm to M7 thread
4mm OD plastic tubing	FESTO	PUN-4x0.75	RS	12 cm	Pneumatics	4 mm OD and 10 bar
PicoNozzle Kit	World Precision Instrument	5430 - x	WPI	1	Pneumatics	for 1.0 / 1.2 / 1.5 / 2.0 mm glass pipettes
Arduino Nano	Arduino		RS	1	Electronics	nano
PCB socket	Samtec	SSW-116-02-G-S	RS	2	Electronics	1 row, 15 pin
Darlington pair NPN transistor	Fairchild	KSP 13BU	Farnell	1	Electronics	Ic = 500 mA, Vces = 30 V
1 k? variable resistor	Vishay	249FGJSPXB25102KA	RS	1	Electronics	panel mount
Knob for variable resistor	RS Online	783-0192	TE Connectivity	1	Electronics	6.35mm shaft radius
DPDT switch	Multicomp	2MD3T2B2M6RE	Farnell	1	Electronics	ON-(OFF)-ON panel mount
SPST footswitch	RS Online	335-319	RS	1	Electronics	push-to-contact

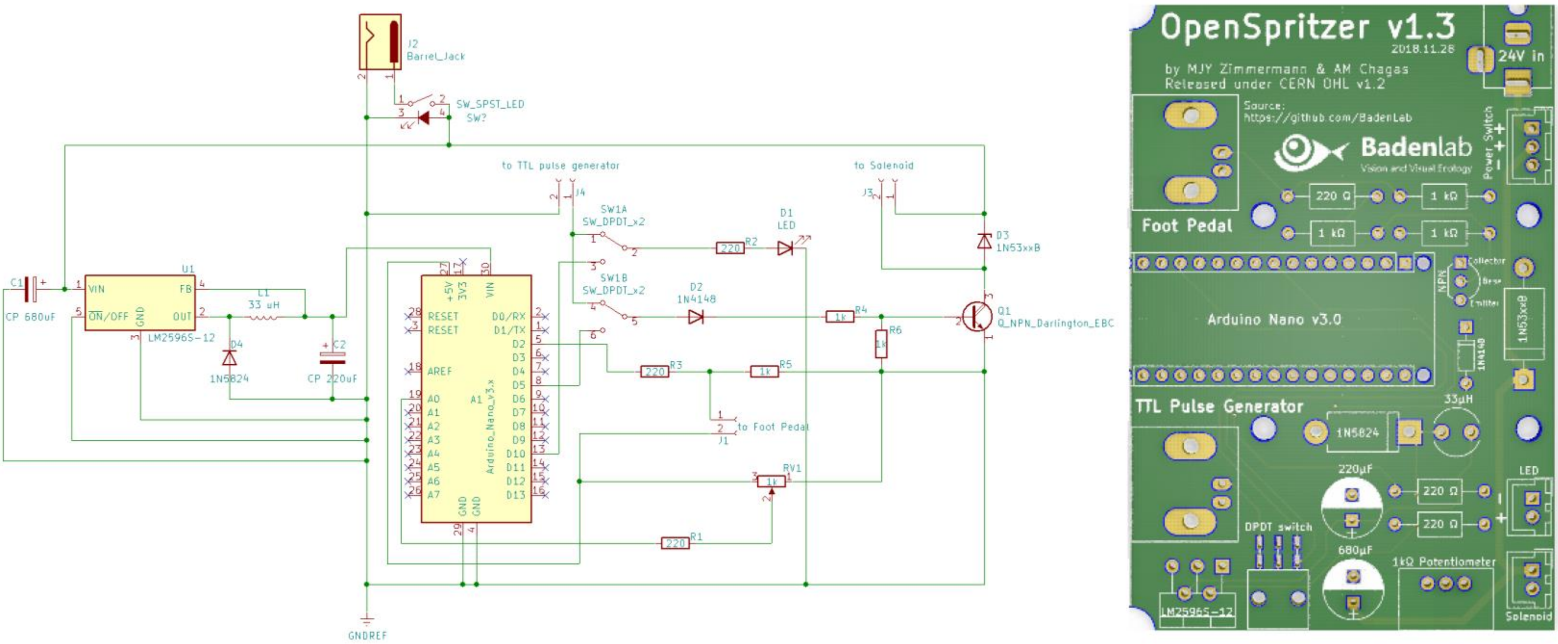
24V DC desktop PSU 2.1 mm jack	RS Online	436-058	RS	1	Electronics	24 V dc 1 A
2.1 mm right angle DC socket	RS Online	884-0957	RS	1	Electronics	2.1 mm right angle
24 V Zener Diode	Semiconductor	1N5359BG	Farnell	1	Electronics	24V
1N4148 signal diode	Fairchild	1n4148	Farnell	1	Electronics	24V
1 k? resistor	Digikey	CF14JT1K00CT-ND	Digikey	3	Electronics	1 k? through hole resistor
220 ? resistor	Digikey	CF14JT220RCT-ND	Digikey	3	Electronics	220 ? through hole resistor
Voltage step-down regulator	RS Online	533-3743	RS	1	Electronics	5V, 3A
Capacitor 680 µF	RS Online	526-1569	RS	1	Electronics	24V minimum
Capacitor 220 µF	RS Online	526-1480	RS	1	Electronics	24V minimum
Diode 1N5824	RS Online	714-6819	RS	1	Electronics	24V minimum
Inductor 33 µH	Bourns	RLB0914-330KL	Farnell	1	Electronics	
3 mm LED	Digikey	350-1557-ND	Digikey	1	Electronics	3mm round, throughhole, grn
Mount for 3 mm LED	Digikey	67-1330-ND	Digikey	1	Electronics	3mm panel mount
BNC coaxial connectors	TE Connectivity	1-1337542-0	Farnell	2	Electronics	Right angle jacks, 50ohm
Cable Mount BNC Connector	RS Online	546-3721	RS	1	Electronics	Plug Crimp Termination
2 way JST Female Connector Housing	RS Online	820-1611	JST	2	Electronics	
3 way JST Female Connector Housing	RS Online	820-1614	JST	1	Electronics	
2 way JST PCB header	RS Online	820-1554	JST	2	Electronics	
3 way JST PCB header	RS Online	820-1557	JST	1	Electronics	
2 distinc colour wires	RS Online	331-862	RS	20 cm	Electronics	20 AWG
Cable shrink	RS Online	170-6377	RS	5 cm	Electronics	Shrunk diameter 1.2 mm
Rocker LED switch			Ebay	1	Electronics	Round Switch / 3 pins
M3 x 10mm screw	RS Online	660-4636	RS	4	Mechanical	M3 x 10 Hex socket cap
M3 nut	RS Online	122-4400	RS	4	Mechanical	M3 nuts
M4 x 50mm Hex Socket Cap Screws	RS Online	304-4592	RS	4	Mechanical	M4 x 50
M4 nut	RS Online	189-579	RS	4	Mechanical	M4 nuts

Supplementary Table T1: Parts schedule for Openspritzer

Assembling the Stimulator

<u>1 – Obtaining the custom-designed PCB</u>

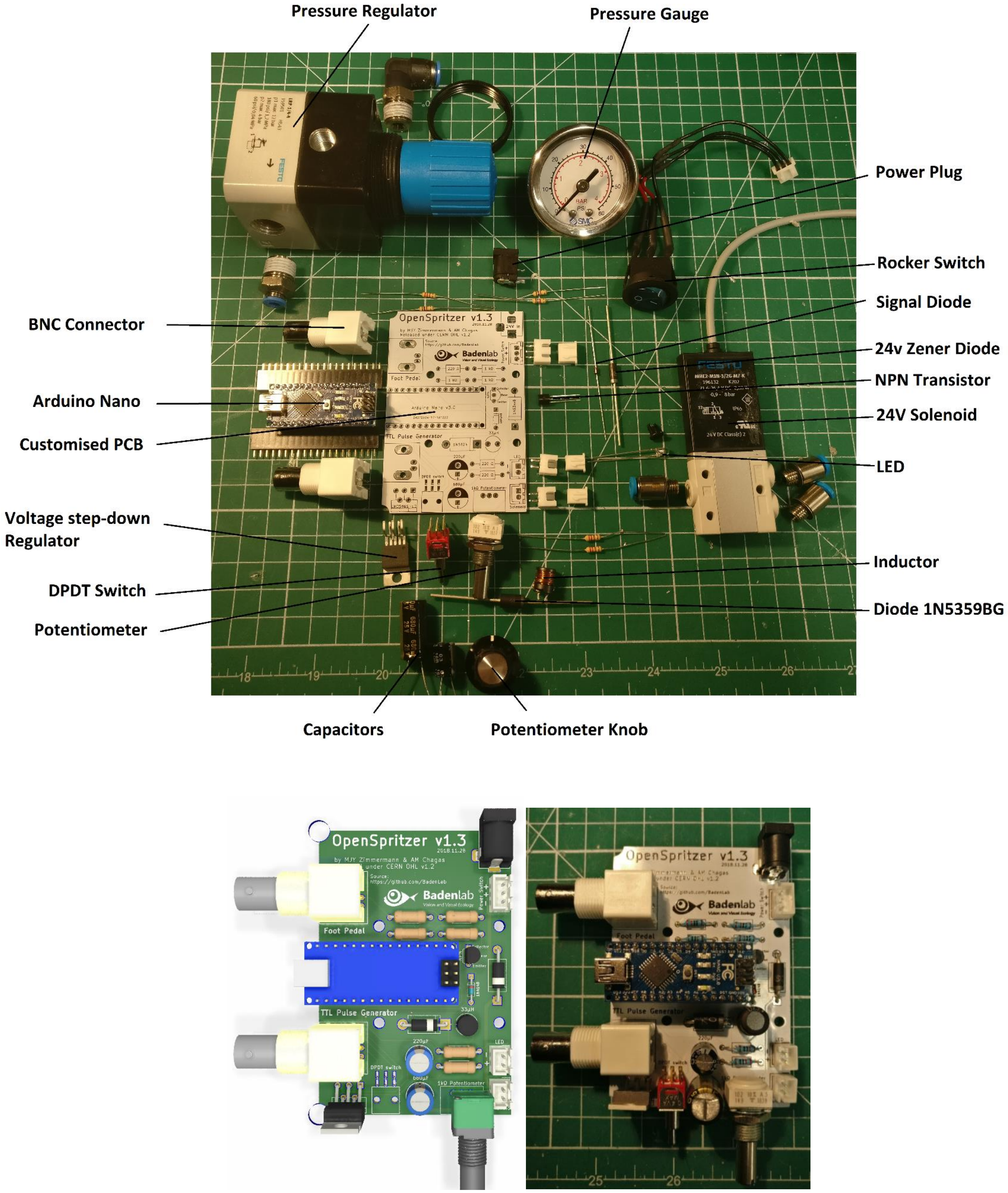
From the GitHub repository (https://github.com/BadenLab/Openspritzer/PCB), one can find the <u>gerber.zip</u> folder needed to order the PCB to any manufacturer company (i.e. https://jlcpcb.com). Ordering a minimum of 5 units should not cost more than £10. Gerber files were designed with KiCad 5.0. Schematics and PCB footprint can be downloaded and modified from the same repository if need be (Fig. S3).



Supplementary Fig. S3: Schematics representing the device circuitry. The 24V power (J2) supply directly the solenoid valve which is controlled by an Arduino nano (A1), itself powered by a voltage regulator (U1). A potentiometer (RV1) value is read by the Arduino and sends a pulse in turn to the NPN transistor (Q1) which then connects the solenoid to the ground, activating it. A DPDT switch (SW1) allows the user to switch from a manual to a TTL control mode (left). Custom made PCB designed to fit the v1.3 custom box (right).

2 – Soldering the custom-designed PCB

The board is self-explanatory and should not take more than an hour to solder.



Supplementary Figure S4: Components and PCB assembly

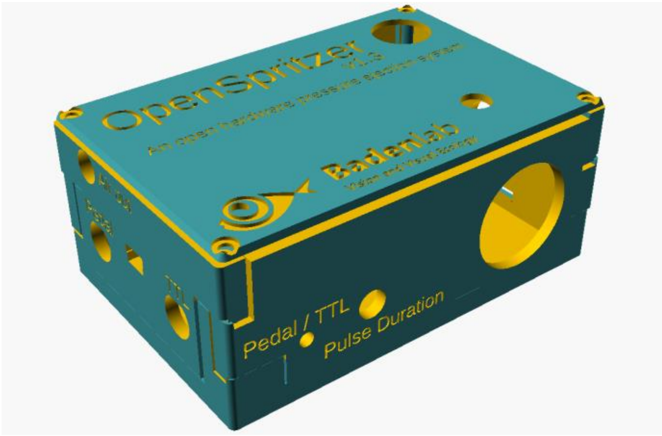
3 – Printing the Stimulator box

STL files can be found on the GitHub repository and print directly. However, SCAD files are also available and easily adjustable for personalised design.

We used OpenSCAD (freely available at www.openscad.com) to design the stimulator box. The tolerance of the printer can be adjusted in the “USER Parameters” section of the script (tol =0.1; by default, this value is used for Prusa MK3 and Ultimaker 2). Each component can be displayed/design individually in the “Switches” section. Variables can be adjusted in the “Component Parameters” section.

The PCB is screwed to the “Bottom” part of the box by using M3 screws and nuts.

All part should fit tightly together and are maintained together by 4 M3*50mm socket screws.



Supplementary Figure S5: Rendering of the custom design 3D printing box that fits exactly the custom PCB and the Spritzer components.

4 – Connecting the Spritzer

Connect an air compressor to the “Air In” port via the L-shaped adaptor.

The output airhose must be connected to the “Air Out” port.

A good pressure for sharp crisp pulses, which helps the solenoid to close again, is 1.4 bar (20 psi). **Important**: When first connecting high-pressure air to Openspritzer, e.g. from a compressor or air-tank, make sure that the Openspritzer gauge is near zero. A too-high line pressure (>>1.4 bar) risks irreversibly damaging the solenoid.

The stimulator must be powered with a 24v power supply. The Arduino Nano is powered through an internal voltage step-down regulator (24 to 5v). And does not need to be connected to be connected through a mini-USB cable once code has been uploaded.

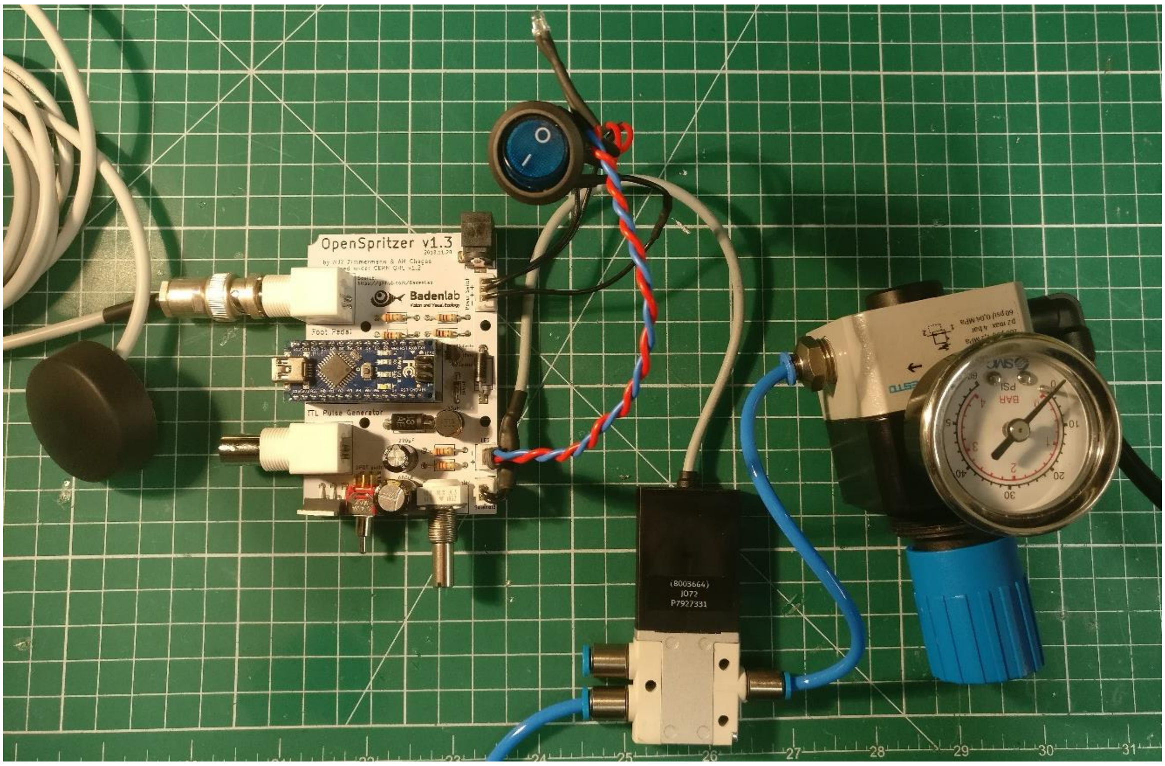
For convenience in darkened circumstances the LED will also illuminate whenever the 5v line is high.

The foot pedal and/or the TTL cable(s), once soldered to a BNC male connector should be connected on the left side of the box to their appropriate BNC female connector. To operate with a footswitch the DPDT must be placed in the "Pedal" position. It is not necessary to use the Arduino to operate the solenoid if you have a convenient method for producing 5v TTL pulses, such as PulseQ electrophysiology package (Funetics) running on Igor Pro (Wavemetrics) in conjunction with the ITC 1600. To use TTL pulses ensure the DPDT switch is in the “TTL” position. Whenever the 5v line is high the solenoid will open and release compressed air from the output valve.

Attaching a computer at operation time allows for dynamic reprogramming of the Arduino during experimentation, which could be very useful. E.g. resetting the base time unit.

To operate the unit, press the foot-button and after 400 milliseconds a puff of a predefined duration will occur. The LED will illuminate during the puff to give visual confirmation that the puff has occurred. The duration of the puff can be changed by adjusting the potentiometer.

When the potentiometer is changed Openspritzer goes into a monitoring mode during which time the LED is lit and indicate the new time duration.



Supplementary Figure S5: The above picture shows the connection and arrangement between the custom PCB and all components.

<u>5 - Programming the Arduino on board Openspritzer</u>

Once Openspritzer is assembled the Arduino must be programmed: Download and install Arduino programming environment on your computer (www.Arduino.org).

Open the ***Openspritzer.ino*** code which is included in the Supplementary Materials.

Compile and upload code to the Openspritzer by connecting the USB port of Openspritzer to the USB port of your computer and by clicking on the "Upload" icon in the top left corner of the screen.

Code can be modified and uploaded as many times as you like. The way the Openspritzer code has been constructed allows simple modification of key parameters such as the base unit of time and are listed in supporting table 2.

IMPORTANT: The first time that the Arduino is used the **POT_LOW** and **POT_HIGH** parameters should be measured. A Serial.println statement is included in the code to allow monitoring of this value over the serial port. To measure these values, open the serial port monitoring window in the Arduino programming environment (magnifying lens, top right corner) and then sweep the potentiometer from low to high. The two parameters **POT_HIGH** and **POT_LOW** should be initialised to the highest values and lowest values observed in the monitoring window output (usually those should be 0 and 1023). This is used in the calculations of the pulse duration and will vary from Openspritzer to Openspritzer depending on the precise value of the potentiometer resistance.

The monitoring algorithm relies on the **PotNoise** parameter to decide whether or not the potentiometer has settled. To set **PotNoise**, monitor the serial port output during the analog monitoring procedure (described above) and find the **PotValue** measured by the analogRead command. **PotNoise** should be larger than any fluctuations in the **PotValue** when it is not being altered. During testing, depending on which power supply was used, it was found that a reasonable value for PotNoise was 5.

Next the user assigns a time to the minimum and maximum potentiometer value by modifying **PULSE_LOW** and **PULSE_HIGH**. Those value are in milliseconds. **PULSE_LOW** will correspond to the minimum air puff duration when the potentiometer is turn all the way down, and reciprocally for **PULSE_HIGH**. By default those values are 10 and 1000 ms.

Finally, the **PULSE_Delay** is to be defined. This value allows some time for the user to release the button before applying the next pulse. This value can be adjusted to give rapid or spread pulses, however we found that for the average user, a value of 500ms is sufficient.

The user is obviously free to write any code they wish to monitor the foot button, read the potentiometer, blink the LED or operate the solenoid.