Poseidon Quick Facts:

Operation:	infuse & withdraw
Total Cost:	$500
Print time:	20hr
Build time:	20min
Syringe:	BD Plastic 1-60mL
Dist Units:	mm, mL, μL
Time units:	sec, min, hr
Pos. Tol:	± 0.01 mm
Pos. Range:	0-150mm
Speed Tol:	± 0.01 mm/sec
Speed Range:	0-∞ mm/sec
Max Rate:	0 mL/sec
Min Rate:	0 mL/sec
Microscope:	8x

Experiments are easy:

Assuming components are connected and powered:

1. Launch the GUI by typing `python gui.pu`
2. Load `testing.txt` by pressing `File -> Load Settings`
3. Select the arduino port
4. Send the settings to the controller by pressing `send to controller`
5. In the run tab, enable the pumps you want to run, and enter their displacements
6. Press `run` on the run tab to start your experiment
7. Pause or stop at any time by pressing `pause` or `stop`

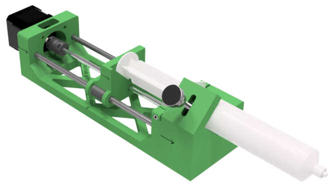
Get started:

Check out the project page which has all of the files as well as detailed instructions on how to build (and hack) the system! `https://pachterlab.github.io/poseidon/`

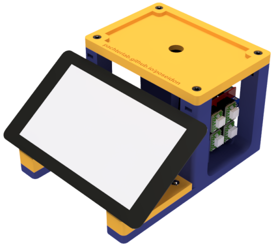
On the development of the Poseidon System

The Poseidon system was developed as a highly modifiable and cheap alternative to currently available syringe pump and microscope systems. At 10x cheaper than current systems, and built using the *Principles of Open Hardware Design*, Poseidon allows users to simply setup, modify, and run microfluidics experiments.

> *Principles of Open Hardware Design include: principle one, principle two, etc*



The Poseidon system includes a syringe pump and a microscope. These are two important components in any microfluidic experiment. Current pumping and vision systems include Harvard Apparatus, **insert other here**, Nikon desktop optical microscope, and **insert other here**. While these devices are used widly and have many unique features, they are expensive and many lack the ability to modify the operation of the device.



The Poseidon system was developed using entirely 3-D printed parts and components that are all easily sourced from Amazon. The syringe pump uses an arduino, compatible motor drivers, and Nema 17 stepper motor to drive a 0.8mm/step lead screw which in turn moves a sled that is mounted on linear bearings. The displacement of the sled moves the syringe forward or backward allowing the user to dispel or intake liquid.

The microscope was built using a raspberry pi computer and USB microscope. The raspberry pi acts as a computer that controls the syringe pumps by sending move commands via the arduino microcontroller.

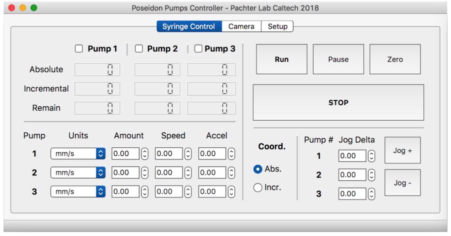
On the hackability of the Poseidon System

Every user made component of the Poseidon system is hackable and modifiable. These include

- CAD (computer aided design) file of the 3-D printed components
- GUI (graphical user interface) and its layout
- Arduino sketch that runs the motors
- Controller code that connects commands from the GUI to the arduino

The 3-D printed components were designed using Fusion 360, a CAD software available free to students. To modify any 3-D printed part the user downloads Fusion 360, loads the files, and implements changes. These files can then be 3-D printed using available slicing software and printer.

The GUI was created using Qt designer, a drag and drop application for organizing buttons. The user can load the uiFileName.ui file and modify any buttons and inputs she desires. This new GUI can then be converted to a python file using `pyuic5 uiFileName.ui > uiFileName.py`.



Poseidon Pumps Controller - Pachter Lab Caltech 2018

Pump	Units	Amount	Speed	Accel
1	mm/s	0.00	0.00	0.00
2	mm/s	0.00	0.00	0.00
3	mm/s	0.00	0.00	0.00

Coord. ● Abs. ○ Incr.

Pump #	Jog Delta
1	0.00
2	0.00
3	0.00

With the new ui python file, the user can configure which commands are sent to the controller when different elements of the GUI are activated (via a button press or checkbox toggle etc). These commands are sent to, and interpreted by the custom Arduino sketch.

`<mode,setting,motor,value,dir,opt1,opt2,opt3>`

Variable	Argument (sent as a string)
mode	["RUN", "SETTING", "JOG", "STOP"]
setting	["ALL"", "FEW", "ONE", "ACCEL", "SPEED", "DELTA"]
motorID	[1, 2, 3, 12, 23, 13, 123]
value	any positive floating number
direction	[’F’, ’B’]
opt1	any floating number
opt2	any floating number
opt3	any floating number

The Arduino interprets the custom commands that are sent by python, through the serial port (USB port), and executes the associated function on the board which in turn results in stepper motor movement. Certain commands require value inputs, such as distance to move, and in this case the Arduino is capable of grabbing those values from the python command and executing a local function with that value. The user can take advantage of this protocol by developing custom movement patterns using the Arduino functions.

On the licensing of the Poseidon System

This is still TBD but Eduardo is adamant about a certain license.

User Quotes

> *I loved how easy the system was to setup!*
>
> — L. Pachter (P.I.)

> *Screw big corporations! I want to start my own big corporation using these pumps for profit. MWUAHAHA*
>
> — E. D. V. Beltrame (Graduate Student)

> *Finally a reliable, inexpensive, and hackable system that I can use for my Drop Seq experiments.*
>
> — J. Gehring (Post Doc)

Tech Specs

Motor Type:	Nema 17
Motor Steps:	200/rev
Screw pitch:	0.8 mm/rev
Microstep:	$^1/_2$ $^1/_8$ $^1/_{16}$ $^1/_{32}$
Dist per step:	4 μm
Arduino Power:	12V DC @ 2A
Raspi Power:	5V DC @ 1A

Contact Us:
Pachter Lab, California Institute of Technology Pasadena, CA.
pachterlab@caltech.edu
pachterlab.github.io/poseidon