



Contents lists available at [ScienceDirect](#)

HardwareX

journal homepage: [www.elsevier.com/locate/ohx](http://www.elsevier.com/locate/ohx)



## Hardware Article

# $\mu$ Pump: An open-source pressure pump for precision fluid handling in microfluidics



Run Ze Gao<sup>1</sup>, Marie Hébert<sup>1</sup>, Jan Huissoon, Carolyn L. Ren<sup>\*</sup>

*Mechanical and Mechatronics Engineering, University of Waterloo, 200 University Avenue West, Waterloo N2L 3G1, Canada*

## ARTICLE INFO

### Article history:

Received 6 September 2019

Received in revised form 19 November 2019

Accepted 6 January 2020

### Keywords:

Open-source hardware

Microfluidics

Pneumatics

Laboratory automation

Fluid handling

Lab on a chip

Micro total analysis systems ( $\mu$ TAS)

Biochip

BioMEMS

Pressure-driven flow

Active control droplet microfluidics

## ABSTRACT

An open-source precision pressure pump system and control software is presented, primarily designed for the experimental microfluidics community, although others may find additional uses for this precision pressure source. This mechatronic system is coined ' $\mu$ Pump,' and its performance rivals that of commercially available systems, at a fraction of the cost. The pressure accuracy, stability, and resolution are 0.09%, 0.02%, and 0.02% of the full span, respectively. The settling time to reach 2 bar from zero and stabilize is less than 2 s. Material for building a four-channel  $\mu$ Pump (approx. \$3000 USD) or an eight-channel  $\mu$ Pump (approx. \$5000 USD) is approximately a quarter, or a third of the cost of buying a high-end commercial system, respectively. The design rationale is presented, together with documented design details and software, so that the system may be replicated or customized to particular applications.  $\mu$ Pump can be used for two-phase droplet microfluidics, single-phase microfluidics, gaseous flow microfluidics and any other applications requiring precise fluid handling.  $\mu$ Pump provides researchers, students, and startups with a cost-effective solution for precise fluid control.

© 2020 The Authors. Published by Elsevier Ltd. This is an open access article under the CC BY-NC-ND license (<http://creativecommons.org/licenses/by-nc-nd/4.0/>).

## Specifications table

Hardware name	Open source precision pneumatic pump ( $\mu$ Pump)
Subject area	<ul style="list-style-type: none"><li>• Microfluidics</li><li>• Engineering and Material Science</li><li>• Chemistry and Biochemistry</li><li>• Medical (e.g. Pharmaceutical Science)</li><li>• Biological Sciences (e.g. Microbiology and Biochemistry)</li><li>• Educational Tools and Open Source Alternatives to Existing Infrastructure</li><li>• General</li></ul>
Hardware type	<ul style="list-style-type: none"><li>• Precision fluidic handling</li></ul>
Open Source License	GNU General Public License (GPL)
Cost of Hardware	~\$5,000USD for a full 8-channel $\mu$ Pump
Source File Repository	All source files and design files are uploaded at the time of submission

\* Corresponding author.

E-mail address: [c3ren@uwaterloo.ca](mailto:c3ren@uwaterloo.ca) (C.L. Ren).

<sup>1</sup> These authors contributed equally to this work.

<https://doi.org/10.1016/j.ohx.2020.e00096>

2468-0672/© 2020 The Authors. Published by Elsevier Ltd.

This is an open access article under the CC BY-NC-ND license (<http://creativecommons.org/licenses/by-nc-nd/4.0/>).

## 1. Hardware in context

Microfluidics deals with the study of fluids at the micron scale. The small scale provides key advantages, including lower reagent consumption, shorter reaction time, and lower cost. Applications range across a variety of fields, including material synthesis [1], biological assays such as PCR [2] and drug development [3], organs-on-a-chip [4] and fuel cells [5]. Most microfluidics platforms use either a syringe pump or a pressure pump to drive the fluids.

In the context of droplet microfluidics, syringe pumps are commonly used when a pressurized air source is not available. However, their long and persistent transient is undesirable for high-performance applications [6,7]. Harvard Apparatus is one of the common providers of commercial syringe pumps. The syringe pump technology is relatively simple and relies on a stepper motor driving the piston of a syringe linearly; hence, building such a prototype is fairly straight forward and has been documented in the literature [8].

Pressure pumps reduce both long and short term pressure oscillations. Commercial systems typically cost more than syringe pumps and are made by companies such as Elveflow and Fluigent. Although some commercial systems have software development kits (SDKs) available, the pump is effectively a black box, and customization is limited.

Open-source pressure-driven options are quite limited. Information for the so-called Quake's valves is available to build a system to control multi-layer microfluidic chips [9,10], but the operating principle is inherently different from a pressure pump system such as the MFCS-EZ sold by Fluigent. Pressure pumps are microcontroller-based systems that can essentially generate a continuously variable pressure signal (albeit in tiny discrete increments); multi-layer devices, on the other hand, are controlled by solenoid valves that switch port pressures on or off, and are thus discrete (or binary) as opposed to continuous. Recent work has also produced low cost open-source microfluidic control system that focuses on miniaturization and control of a large array of solenoid valves while incorporating a mini air compressor within the build [11]. However, such a system trades precision and accuracy for portability.  $\mu$ Pump, on the other hand, is the opposite, it is a high precision benchtop fluidic control system that has the performance on par with much more expensive commercial systems.

An open-source pressure pump system offers not only the higher performance in terms of short response time and improved stability, but also the ability to customize and tailor the design to individual requirements, and at a significantly lower cost than commercially available systems. The lower cost should help microfluidics become more accessible to researchers in a broad range of fields, including chemistry, biology, material science, etc. As an open-source system with full access to, and understanding of, the detailed construction and operation, it also shows promise in less developed areas of microfluidics such as active control of individual droplets [12,13].

## 2. Hardware description

### 2.1. Design overview

The design of  $\mu$ Pump is based around a commercial electronic pressure actuator from ControlAir Inc. This pressure actuator is an electro-mechanical device that controls the air pressure at the output port in response to an input voltage signal, via an internal closed-loop pressure feedback system and a piezoceramic actuator. This pressure actuator is integrated into a mechatronic system, as shown (for a single channel) in Fig. 1.

Referring to Fig. 1, the PC runs the  $\mu$ Pump Graphic User Interface (GUI) that communicates with  $\mu$ Pump via a serial USB connection. At the  $\mu$ Pump end, an Arduino Mega microcontroller provides the USB interface with the PC and communicates with a Digital to Analog Converter (DAC) to set the desired output pressure, as well as with an Analog to Digital Converter (ADC) to read the actual pressure at the outlet port. The communications between the microcontroller and the DAC and ADC are via a SPI serial interface.

The overall design is modular in the sense that the printed circuit board (PCB) on which the ADC, DAC and support circuitry are placed is designed to accommodate components for up to eight individually controlled ports, although the board needs only be populated with as many channels as required for a particular application. Likewise, only as many pressure actuators as required need to be included. This helps reduce costs in applications where only 1 or 2 ports are required, while providing the potential to control up to 8 ports using the same PCB and control software. The pressure actuator can also be changed to any OEM (original equipment manufacturer), as long as the input signal is 0–5 V.

#### 2.1.1. Required performance

#### 2.1.2. Active pressure actuator characterization

The active pressure actuator selection is informed by the performance criteria (Table 1) previously specified to ensure suitable pressure output characteristics. The performance of  $\mu$ Pump is primarily determined by the active pressure actuator. As such, the dynamic characteristics of the pressure actuator, ControlAir model T900-CIM [14], were determined from the manufacturer's performance specifications (shown in Table 2) and by investigating the unit's step and frequency response

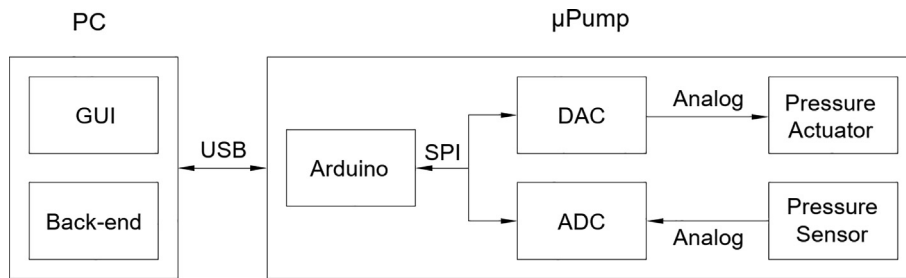


Fig. 1. System overview showing the communication pathways between components.

Table 1

The minimum criteria that must be met in the context of microfluidics applications correspond to the performance of a commercial Fluigent pump.

Minimum Performance Criteria	Values
Pressure output accuracy	$\pm 2$ mbar
Pressure reading resolution	$\pm 1$ mbar
Sampling rate	10 Hz ( $\sim 63$ rad/s)

Table 2

Control Air T900-CIM performance specifications.

Specifications	Values
Flow capacity	7.6 m <sup>3</sup> /hr
Accuracy, hysteresis and repeatability	$\pm 0.10\%$ of span
Deadband	0.02% of span

in outlet pressure. For a span of 2 bars, the corresponding error band of  $\pm 0.10\%$  is  $\pm 2$  mbar, which satisfied the first criterion from Table 1.

The outlet pressure is measured using an MPX4250DP linear pressure sensor produced by NXP Semiconductor [15]. This is an integrated piezoresistive transducer with on-chip signal conditioning and temperature compensation. The sensor output voltage is given as:

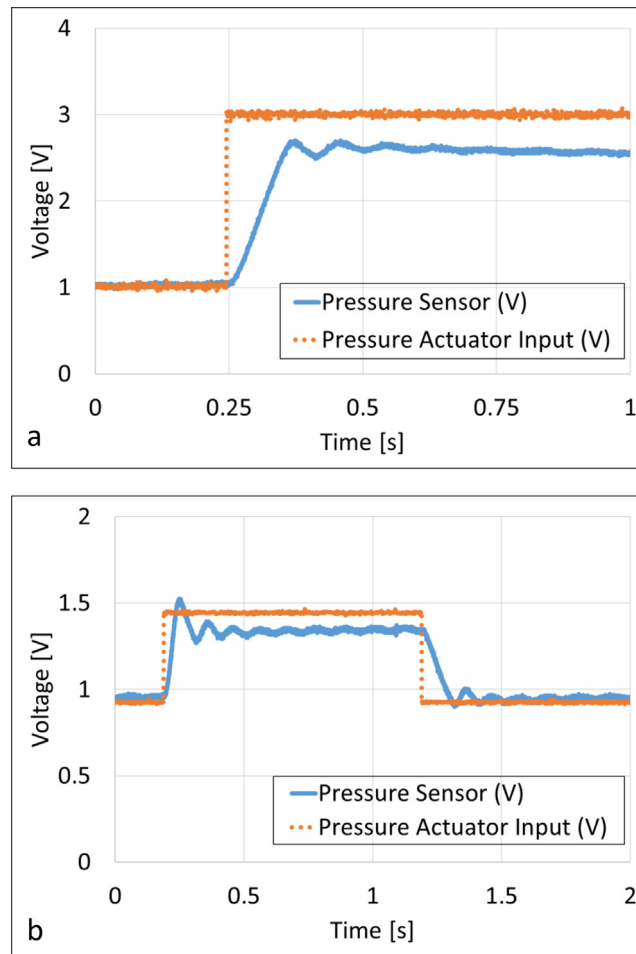
$$V_{out} = V_{cc} \cdot (P \cdot 0.00369 - 0.04) \pm V_{err} \quad (1)$$

where  $V_{cc}$  is nominally 5 V,  $P$  is between 0 and 250 kPa, and  $V_{err}$  is typically  $\pm 0.04$  V.

The output voltage ( $V_{out}$ ) range is linear between 0.2 V and 4.86 V, and a 12-bit ADC with a 5 V input range provides a least significant bit (LSB) resolution of 0.5 mbar. The resolution is suitable with respect to the criterion established in Table 1. It is important that the sensor  $V_{cc}$  is stable and accurate since the output voltage is directly proportional to this. As such, a MAX6250 precision voltage reference capable of supplying the nominal 7 mA sensor supply current was chosen.

Since we aim to be able to both read and set the outlet pressure to be within 1 mbar, a 12-bit DAC was chosen to provide the input voltage to the ControlAir pressure actuator. A MCP4922 dual 12-bit converter was selected since most applications were envisioned as requiring at least two controlled output ports.

The outlet pressure response of the ControlAir model T900-CIM to a control voltage step input from 1 V to 3 V is shown in Fig. 2a. It is evident that the response is underdamped with a settling time of approximately 0.2 s, and with a maximum slew rate of 8 bar/s. The response to a 0.5 V square wave is shown in Fig. 2b, from which it is evident that the positive and negative change in pressure response is somewhat different, although the response and settling times are similar. To estimate the natural frequency, the amplitude ratio and phase response at select frequencies were measured, as shown in Fig. 3. The natural frequency was estimated to be about 40.8 rad/s or 6.5 Hz, and the low-frequency in-phase response and 180° high-frequency phase lag indicate a second-order system. The performance characteristics of the ControlAir pressure actuator are thus suited to applications where command pressure updates are issued at up to about 62.8 rad/s or 10 Hz as specified in Table 1. The frequency response analysis better informs the upper command update limit for continuously varying pressures; the settling time depends on the step and volume but can generally be guaranteed to be less than 2 s.



**Fig. 2.** (a) Pressure regulator (ControlAir T900-CIM) response to a step from 1V to 3V; (b) Pressure regulator (ControlAir T900-CIM) response to a 0.5V square wave.

## 2.2. Detailed description

The  $\mu$ Pump design is modular in the sense that the number of independent pressure channels can be easily changed to accommodate the number of ports required. The control software can also be easily configured to accommodate up to 8 channels, and can readily be embedded within third-party software.  $\mu$ Pump can be used as standalone equipment or can be integrated into a more complex system. All hardware components used to build  $\mu$ Pump can be readily purchased from online vendors, and local suppliers can often provide many, if not all, the components needed. The graphical user interface (GUI) gives the user full control of  $\mu$ Pump, allowing for quick parameter changes and seamlessly connecting and disconnecting with  $\mu$ Pump, as well as reading and recording the pressure information for each fluidic channel.

The paper is divided into the following sections: **3. Design files**, **4. Bill of material**, **5. Build instructions**, and **6. Operation instructions**, which provide a complete set of information on making and using  $\mu$ Pump.  $\mu$ Pump (Fig. 4) consists of three assemblies: a pneumatics assembly, an electronics assembly, and the housing. The working fluid, typically dry air filtered to 1  $\mu$ m (although a variety of non-corrosive gasses may be used) connects to the unit via a 1/4" NPT inlet port at the back panel.

### 2.2.1. Pneumatics assembly

The pneumatics assembly provides fluidic pressure to each channel. Shown in Fig. 5 is a 3D CAD image of an 8-channel pneumatic actuator system. The three-port two-way (3/2) solenoid valve controls flow from the inlet port to the distribution manifold. The pneumatic manifold distributes the working fluid to each pneumatic actuator. An inlet pressure sensor (not shown in Fig. 5) connected to the pneumatic manifold provides real-time inlet pressure data as well as an input signal to the software-controlled emergency shutoff via the 3/2 solenoid valve. This provides a layer of protection against out-of-range inlet pressure.

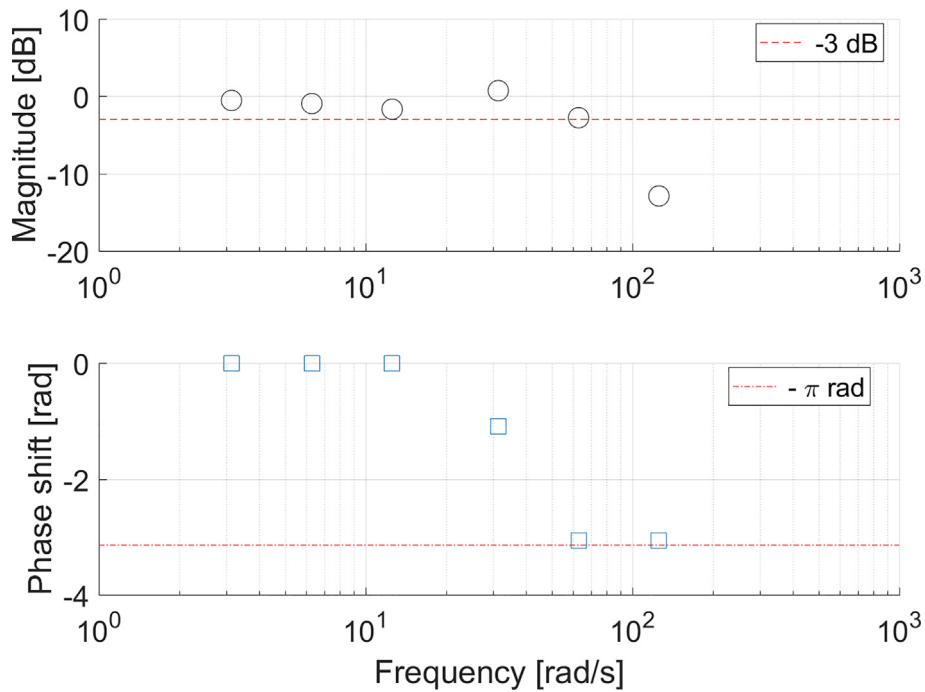


Fig. 3. Pressure regulator (ControlAir T900-CIM) frequency response (Bode plot).

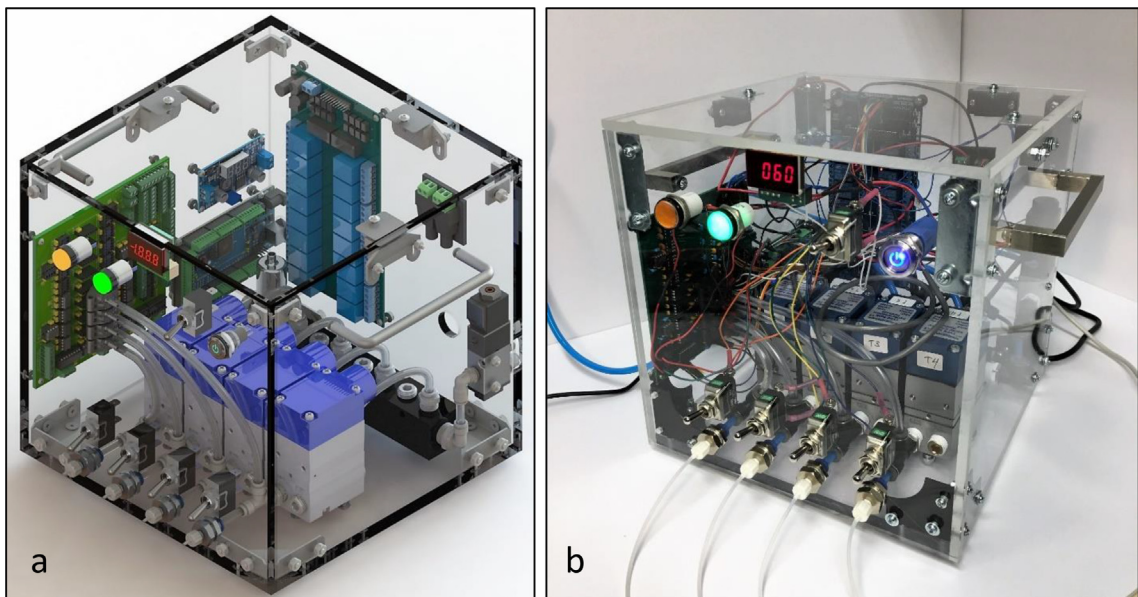


Fig. 4. (a) 3D rendering of a pneumatically controlled microfluidic pump ( $\mu$ Pump) with four independently controlled fluidic channels, (and can be upgraded to a max of 8 channels). (b) Photograph of a completed pneumatically controlled microfluidic pump ( $\mu$ Pump) with four independently controlled fluidic channels.

A pneumatic actuator controls the pressure at each of the 8 independent outlet ports. A solid-state pressure sensor provides real-time data of the actual regulated pressure at each outlet port. Panel mounted outlet connectors provide a secure connection means via silicone or Tygon tubing to reagent reservoirs. All internal tubing is  $\frac{1}{4}$ " o.d. with  $\frac{1}{16}$ " wall thickness. All external pneumatic connections are either  $\frac{1}{4}$ " or  $\frac{3}{8}$ " NPT (National Pipe Thread) or  $\frac{1}{4}$ " push-to-connect fittings, with the exception of the outlet ports which use a #10-32 UNF thread and a Luer lock connection.

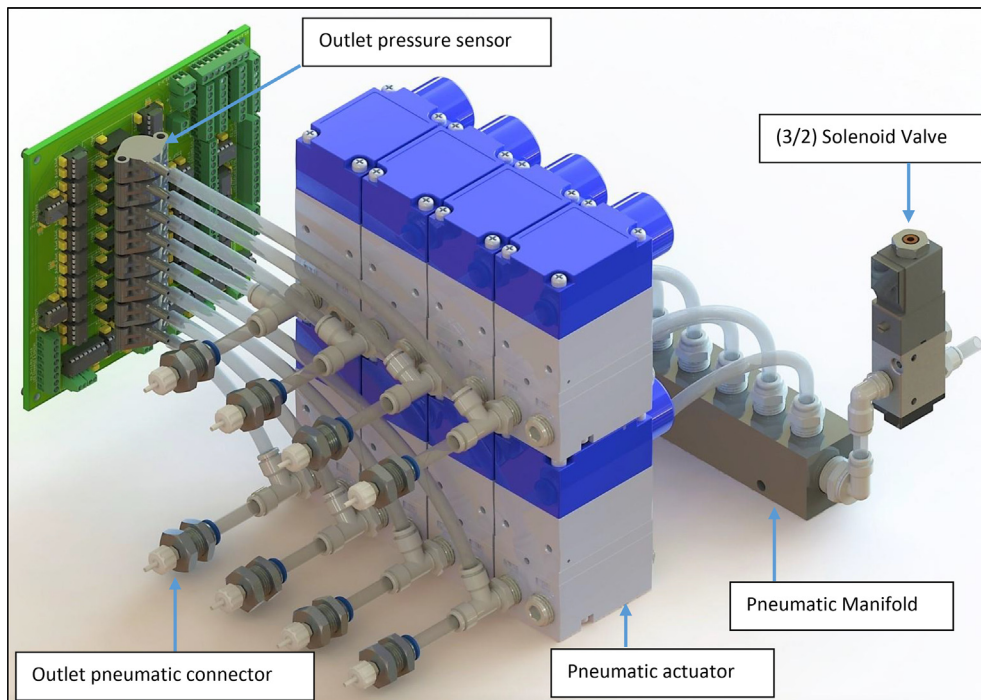


Fig. 5. 3D rendering of an 8-channel pneumatic assembly.

### 2.2.2. Electronics assembly

The electronics assembly provides the power and signal communication to the pneumatic components as well as the interface with the GUI. Shown in Fig. 6 is a 3D CAD image of the electronics assembly for an 8 channel  $\mu$ Pump. The microcontroller, an Arduino Mega, provides a USB interface between  $\mu$ Pump and an external PC; it establishes the communication between the PC and the pneumatic peripherals such as the pressure sensors and pressure actuators through the printed circuit board assembly (PCBA). The user sets the control parameters for the microcontroller via a graphical user interface (GUI) on the PC.

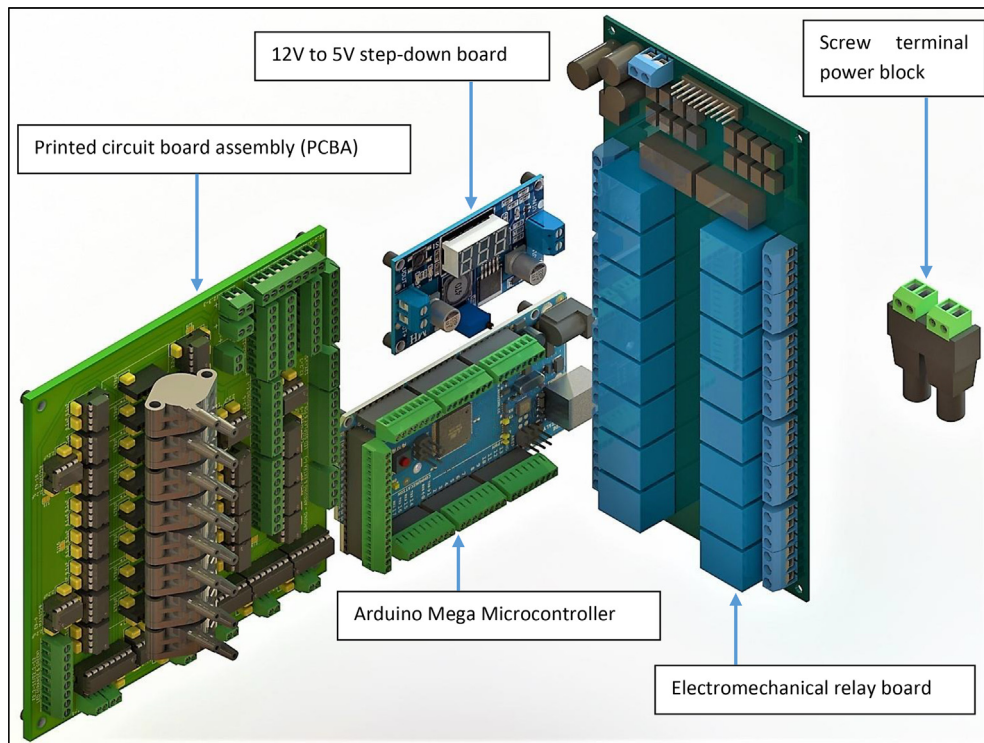
PCBA interfaces the pressure sensors and actuators to the microcontroller. It includes 8 active low-pass filters, an 8-channel 12-bit ADC, four 2-channel 12-bit DACs, and precision voltage references. Electrical power to the active components is also provided on this PCB. The 12 V to 5 V step-down board provides power to the inlet pressure sensor and inlet pressure display. The electromechanical relay board interfaces the inlet solenoid valve and pneumatic actuators to outputs on the Arduino Mega; an electromechanical relay board was chosen for convenience and provides a simple safety disconnect rather than design a solid-state power circuit interface.  $\mu$ Pump requires a 12 V, 2 A external power supply, such as a laptop power adapter. The connector is panel mounted on the rear panel, and the various circuit boards and microcontroller are internally connected via a screw terminal power block. All electrical connections between the various boards are made with stranded tinned wire (22GA for signal, 16GA for power) and screw terminal blocks. All electronic components are off-the-shelf except for PCBA which is custom-made but can be easily outsourced for fabrication based on the provided design files.

### 2.2.3. Housing assembly

The housing provides an enclosure and mounting points for the internal components, as well as the switches and panel connectors. Shown in Fig. 7 is a 3D CAD image of a housing assembly for a 4-channel  $\mu$ Pump.

The six sides of the housing assembly are made of transparent acrylic. The four side walls and top are 1/4" thick, and the base is 3/8" thick. Eight corner brackets hold the sides, top and base walls together using 8–32 threaded fasteners. Four rubber feet are adhered to the base to provide damping and to avoid sliding on benchtop surfaces. Two handles are installed on the left and right walls to facilitate handling and transportation.

Two LED indicators on the front panel indicate the status of  $\mu$ Pump. Toggle switches (for manually turning on and off the inlet solenoid valve and each pneumatic actuator) are mounted on the front panel. The toggle switches act as a secondary safety control, enabling the user to easily and quickly disable any output port. The power push button is a latching switch for turning  $\mu$ Pump on/off. The Arduino Mega and the yellow LED indicator are on as long as a USB cable is connected to a powered-on PC. The 3-digit LED panel meter displays the inlet air pressure in psi.



**Fig. 6.** 3D rendering of the electronics assembly.

Two screw terminal power blocks are secured to the rear panel. The power cable from the AC adapter and the USB cable between the PC and Arduino Mega enter the enclosure through two holes with strain relief fittings. The input tubing and exhaust of the inlet solenoid valve pass through a slot cut in the rear panel.

### 2.3. Software and graphical user interface

The open-source software is a key feature of this project, and enables users and developers to (a) use the software as-is to control  $\mu$ Pump, (b) use the source code to integrate  $\mu$ Pump within other custom software (e.g. [12]), (c) use the software structure as a basis to develop other open-source microfluidic tools.

The open-source nature of the project dictates that open-source libraries are used to build the software. Hence, the graphical user interface (GUI) is developed using the Qt library, and the controller used is the ubiquitous Arduino. The availability of the source code not only allows the user to customize the software superficially by changing the size and format of buttons, window sizes and color scheme, but also allows the user to tailor the back-end functions such as serial communication, pressure update logic, additional connections to pumps, etc., to meet their specific needs.

The software is designed in a modular fashion by leveraging the object-oriented nature of C++. Thus, the different building blocks of the software are encapsulated and more or less self-contained. The modularity allows a plug-and-play approach to be combined with other open-source software or the software to be further developed for user-specific requirements. Furthermore, the configuration class definition can be modified to change other features such as the pin numbers, sensor type, and actuator type, as described in the software architecture overview document provided in the supplementary information.

### 2.4. Types of application

$\mu$ Pump is designed to provide pressure-driven flow for applications within the realm of microfluidics. However, it could also be used in a variety of fields involving precise pressure control, such as:

- Droplet microfluidics
- Single-phase microfluidics
- Soft robotics
- Pneumatics
- Pressurized chambers

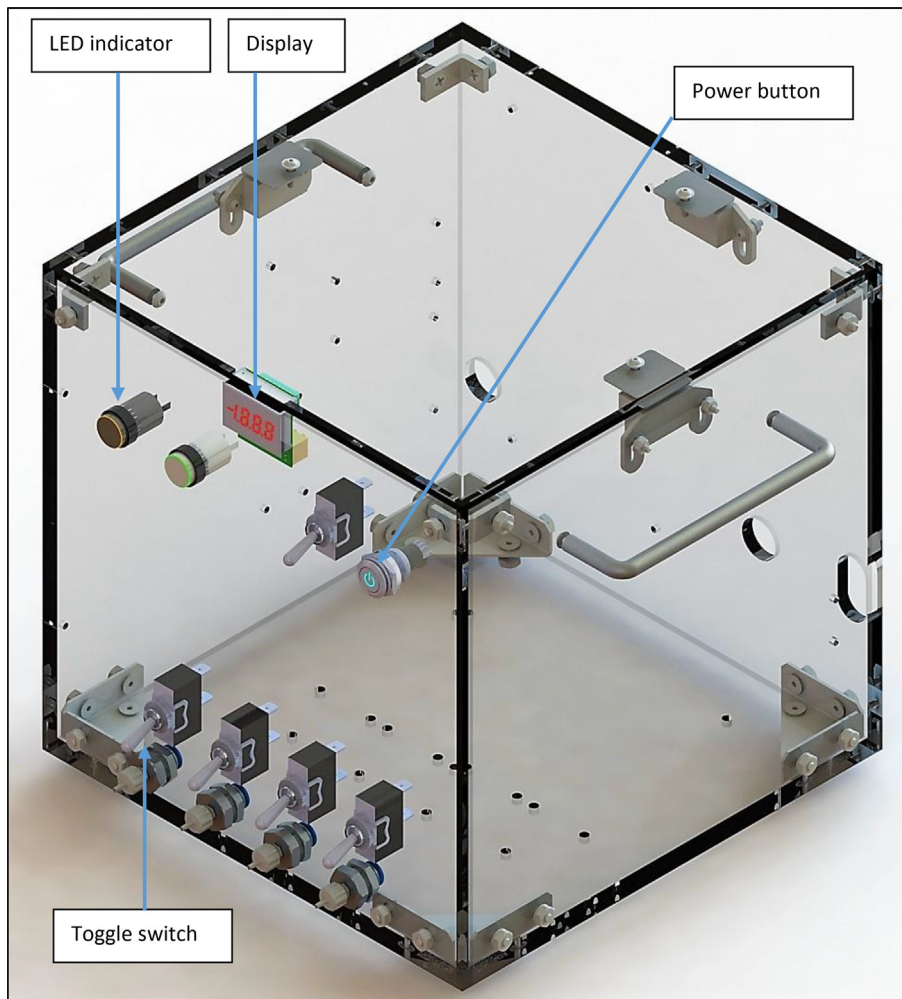


Fig. 7. 3D rendering of the housing assembly.

### 2.5. Open source advantages and cautions

$\mu$ Pump has four main advantages over commercial systems such as Fluigent, Elveflow, and Dolomite:

- Comparable performance but at much lower cost
- The ability to edit and upgrade the GUI and the control software provides the user with capabilities that traditional software development kit (SDK) provided by commercial systems does not
- Components can be added or substituted to tailor performance or cost to the application
- Readily integrated with other systems to make custom devices

The cost of an eight-channel commercial system is typically about three times that of an eight-channel  $\mu$ Pump. Similarly, the cost of a four-channel commercial system is typically four times that of a four-channel  $\mu$ Pump. The cost of a one-channel  $\mu$ Pump is approximately \$1,500 USD, and each additional channel costs approximately \$500 USD more (i.e. an eight-channel  $\mu$ Pump costs approximately \$5,000 USD). Table 3 shows a comparison of the two most important performance characteristics; details of this comparison are provided in Section 7. **Validation and characterization.** For microfluidics applications, the small differences in pressure accuracy and precision between  $\mu$ Pump and high-end commercial systems are negligible in practice as the fluid inside microchannels damps out the variations; for further information, please refer to section 7. **Validation and characterization.**

$\mu$ Pump's software and GUI are programmed in C++ and give the user the ability to tailor the operation and interface of the system. Commercial systems' SDKs typically only give the user relatively high-level function and user instructions with very limited information about the pump details, and minimal flexibility for the user to make changes. Active feedback-controlled



**Table 3**Performance benchmark between  $\mu$ Pump and high-end commercial system.

	$\mu$ Pump (0–2 bar)	Fluigent MFCS-EZ (0–2 bar)	Description
Bias (% of full span)	0.09	0.08	1 standard deviation from set pressure.
Precision (% of full span)	0.02	0.01	1 standard deviation from mean steady state pressure

droplet microfluidics benefits greatly from  $\mu$ Pump's C++ coding due to its versatility, optimization and fine-tuning capabilities.

The heart of  $\mu$ Pump is the electronic pressure actuator (EPA). While there are several manufacturers of such devices, we used a ControlAir T900-CIM in the prototype  $\mu$ Pump. EPAs typically accept either a voltage (0–5 V, 0–10 V) or current (4–20 mA) input, and internally regulate the output pressure as a linear function of the input signal. The prototype  $\mu$ Pump control board provides a nominally 0–5 V signal for each EPA, although this could readily be converted to a different range or current using an auxiliary circuit board.

Although many suppliers of pressure pump system within the microfluidic research market offer “original equipment manufacturer” (OEM) options for their pressure pump, they are still limited by the “black box” problem, meaning that much if not all of the software must be programmed through the suppliers' SDK. Whereas  $\mu$ Pump is programmed in C++ and under open source license; hence, the user can integrate  $\mu$ Pump with their device or equipment with no restrictions. For instance, AES Life Sciences, a maker of microfluidics enabled protein separation instrument, is currently replacing their syringe pump with  $\mu$ Pump for lower cost, higher precision and stability.

## 2.6. Hardware summary

- $\mu$ Pump is made up of three sub-assemblies: the pneumatics assembly, the electronics assembly, and the housing assembly.
- The software is C++ based. The GUI is coded using Qt.
- $\mu$ Pump is intended for use in droplet and single-phase microfluidics; however, it can also be used outside of the field of microfluidics, for instance, soft robotics and pneumatics.
- $\mu$ Pump's open-source advantages over commercial systems are lower cost with comparable performance, offering flexibility to the user to make changes (not a black box), true modularity, and potential for being integrated with any system.
- The user must be ready and prepared to make a  $\mu$ Pump as it does take time, and must be prepared to maintain it and fix it if necessary.

## 3. Design files

Table 4 provides a summary of the design files, all of which are available with the article and under the GNU General Public License (GPL) for open-source distribution and usage.

### 3.1. Design files summary

**Table 4**

Design file summary.

Design filename	File type	Location of the file
uPump pneumatics assembly build guide.pdf	Portable Document Format (PDF)	available with the article
uPump electronics assembly build guide.pdf	Portable Document Format (PDF)	available with the article
uPump housing assembly build guide.pdf	Portable Document Format (PDF)	available with the article
uPump bill of materials.xlsx	Excel Workbook	available with the article
uPump SOP.pdf	Portable Document Format (PDF)	available with the article
uPump model.SLDASM	SolidWorks assembly file	available with the article
uPump pneumatics.SLDASM	SolidWorks assembly file	available with the article
uPump electrical.brd	Autodesk EAGLE PCB board file	available with the article
uPump electrical.sch	Autodesk EAGLE PCB schematic file	available with the article
uPump_PC.exe	Executable file	available with the article
uPump_Arduino.ino	Arduino IDE file	available with the article
uPump_Arduino_testMode.ino	Arduino IDE file	available with the article
uPump software	Folder	available with the article
Software architecture overview.pdf	Portable Document Format (PDF)	available with the article
uPump PCB fabrication	Folder	available with the article
uPump housing wall fabrication	Folder	available with the article

### 3.2. Description of the design files

Below are the descriptions of each design file outlining its purpose and relevance to  $\mu$ Pump. For the pneumatics and electronics assemblies, the files are prepared for building an 8-channel  $\mu$ Pump, which would be helpful to the users with limited experiences in pneumatics and electronics. For the housing assembly, the files are prepared for building a 4-channel  $\mu$ Pump, which is the one physically built and tested in this study, as shown in Fig. 4b. It can be easily expanded to accommodate an 8-channel  $\mu$ Pump with slight modifications such as increasing the size of the housing and doubling the number of holes for hosting the increased number of switches.

#### 3.2.1. Build instructions

**uPump pneumatics assembly build guide.pdf** documents the step-by-step build guide for an 8-channel  $\mu$ Pump pneumatic assembly.

**uPump electronics assembly build guide.pdf** documents the step-by-step build guide for an 8-channel  $\mu$ Pump electronics assembly.

**uPump housing assembly build guide.pdf** documents the step-by-step build guide for a 4-channel  $\mu$ Pump housing assembly.

**uPump bill of materials.xlsx** is an interactive bill of material that shows all the components and associated information such as costs, quantities, and vendors for making various versions of  $\mu$ Pump.

#### 3.2.2. Drawings, schematics and 3D CAD models

**uPump model.SLDASM** is the 3D CAD model of a complete 4-channel  $\mu$ Pump with housing. **uPump model.SLDASM** can be found inside the **uPump model.zip** folder and accessed once all the files are extracted from **uPump model.zip**. It must be noted that all the files must be contained in one folder for **uPump model.SLDASM** to open correctly. The model contains all the components from the pneumatic actuators to the nuts and bolts. It can be used for multiple purposes such as a reference for measuring wire and cable lengths, a visual reference while assembling  $\mu$ Pump, and for modifications to achieve a different  $\mu$ Pump variation (e.g. an 8-channel system by stacking four more pneumatic actuators on top of the existing actuators), or making the box bigger or smaller for accommodating various number or a different model of pneumatic actuators. It must be noted that the CAD model can only be opened by SolidWorks 2018 or later.

**uPump pneumatics.SLDASM** is the 3D CAD model of an 8-channel  $\mu$ Pump's pneumatics assembly. **uPump pneumatics.SLDASM** can be found inside **uPump pneumatics.zip** and accessed once all the files are extracted from **uPump pneumatics.zip**. It must be noted that all the files must be contained in one folder for **uPump pneumatics.SLDASM** to open correctly. The pneumatics assembly 3D model can be used while assembling the pneumatics system as a visual reference, and modifications to achieve a different  $\mu$ Pump variation. It must be noted that the model can only be opened by SolidWorks 2018 or later.

**uPump electrical.brd** and **uPump electrical.sch** are both Autodesk EAGLE files and are linked, meaning that changes made to one are automatically updated on the other. **uPump electrical.brd** is the 2D CAD design file of  $\mu$ Pump's printed circuit board (PCB). **uPump electrical.sch** is the electrical schematic design file of  $\mu$ Pump's printed circuit board. The primary function of these files is for the user to modify and design their tailored PCB. The EAGLE files can export all the files required for standard PCB fabrication. **uPump electrical.brd** and **uPump electrical.sch** are located in the folder named **uPump electrical**.

Please note that SolidWorks and Autodesk Eagle are not free and are not open source software, although student and education licenses are available for both. There are also various online tutorials that may be helpful.

SolidWorks: <https://www.solidworks.com/>

Autodesk Eagle: <https://www.autodesk.com/products/eagle/overview>

#### 3.2.3. Software and GUI

**uPump\_PC.exe** opens the graphical user interface (GUI) on the user's PC. The GUI allows the user to connect to  $\mu$ Pump via a USB cable, change the parameters of  $\mu$ Pump, control each channel of  $\mu$ Pump, save the outlet pressure information, and view the outlet pressure information in real-time via digital seven-segment display or plot. Under the same folder for **uPump\_PC.exe**, a few other files exist, which are the dynamic-link libraries that must be in the same folder as **uPump\_PC.exe** to function correctly.

**uPump\_Arduino.ino** is the control software code that is directly uploaded onto Arduino Mega and saved in its memory. **uPump\_Arduino.ino** allows the Arduino Mega to control and interact with all  $\mu$ Pump's active components, as well as communicate with the GUI. **uPump\_Arduino.ino** is programmed in C++ with many Arduino specific libraries that simplify programming for Arduino microcontrollers and compiled in the Arduino IDE, which is a free and open-source software. The folder which contains **uPump\_Arduino.ino** also contains other files that are the libraries for **uPump\_Arduino.ino** and must be included within the same folder as **uPump\_Arduino.ino** to function correctly.

**uPump\_Arduino\_testMode.ino** allows the user to test the Arduino connected to the PC without using the actual system.

**uPump software** folder contains the files that can create and modify **uPump\_PC.exe**, which is coded in C++. One of the files is **uPump\_PC.sln**, a solution file that opens in Microsoft Visual Studio. By opening **uPump\_PC.sln**, all the subfiles within the **uPump software** folder also open in Microsoft Visual Studio. Moreover, Qt, a UI creating software framework, must be

added to Microsoft Visual Studio. Please refer to **Software architecture overview.pdf** for the complete information on how **uPump\_PC.sln** works to produce **uPump\_PC.exe**. Note that Microsoft Visual Studio is not free and must be purchased. Qt has an open-source option, which is under the (L)GPL v3 License. There are also various online tutorials that may be helpful.

Arduino IDE: <https://www.arduino.cc/en/main/software>

Microsoft Visual Studio: <https://visualstudio.microsoft.com/vs/>

Qt add-on for Microsoft Visual Studio: <https://doc.qt.io/qtvstools/qtvstools-getting-started.html>

Qt Framework: <https://www.qt.io/download>

### 3.2.4. Operation instructions

**uPump SOP.pdf** documents a complete set of step-by-step instructions on how to use  $\mu$ Pump, including the instructions to download all the required software and drivers, pre-usage testing to verify that all connections, both electrical and pneumatics, are correctly made, maintenance and safe operation of  $\mu$ Pump, as well as standard operating procedures.

### 3.2.5. Outsourced fabrication

**uPump PCB fabrication** folder contains the files required to produce the printed circuit board (PCB) for  $\mu$ Pump.

### 3.2.6. Machined parts

**uPump housing wall fabrication** folder contains the drawing files needed to create the housing walls for  $\mu$ Pump.

## 4. Bill of materials

The bill of materials for  $\mu$ Pump is interactive; below is a demonstration of what it can do and how to use it. Upon opening **uPump bill of materials.xlsx**, the user will be greeted by the bill of materials right away and be presented with a few options. Please refer to Fig. 8 while going through the list below, each number corresponds to the number in Fig. 8, and the red box beside shows the item being discussed:

1. The user can click the “1” button on the top left corner to minimize each subsection (i.e. electronics Assembly, Pneumatics Assembly, Housing Assembly, and Assembly Tools), or click the “2” button to expand all the subsections.
2. The section for the main parameters allows the user to choose different parameters. For example, the number of channels and the number of maximum 1 bar pressure channels. The number of maximum 2 bar pressure channels will update automatically. The vendor for the pneumatic fittings and the choices for having the housing and assembly tools can also be made. The drop-down lists for the parameters are made with the Data Validation function under the Data tab in Excel. The options for each parameter can be found and changed from cell P14 to V23; for instance, the currency conversion rate can be changed to account for fluctuations.
3. The current total cost, as well as the breakdown of each subsection, are displayed in US dollars; however, the user is allowed to choose a particular currency. All cost values are automatically generated based on the exchange rate, which can be updated in parameter selection.
4. This subsection shows what each column is, and the arrow beside each cell opens up a window that allows the user to hide/show any value for that particular column. This is particularly useful for filtering out specific vendors, and any components that are zero for a specific build of a  $\mu$ Pump.
5. This is the subsections for each assembly and assembly tools. The button with +/- sign to the left of each subsection can expand or minimize each subsection.
6. This is an example of the expanded subsection. The component column displays the name of each component. The vendor column shows the online seller for each component. The supplier column shows the maker of each component. The vendor number is the number of each component that is displayed on their respective vendor's website, which can also be used to search for each component. Also, the vendor numbers are the links that can be clicked to go directly to the webpage of each component. The quantities of each component are automatically generated based on the parameters. The extra column allows the user to put down a few extras for particular components. It is recommended that the user always purchase a few spare parts for the cheaper large quantity components such as the capacitors. The cost/unit (USD) column displays the cost per unit in U.S. dollars for each component. The column to the right of cost/unit (USD) column shows the per-unit-cost of each component in the currency selected in the parameters section. The subtotal column displays the cost of the total quantity, including the extras for each component. The Universal Part Number (UPN) column displays the unique part number given to every component that is used to identify that part in all documents. The description column shows a concise description of each component. Please note that all underlying calculations for each cell can be modified based on the user's needs.

The image shows a spreadsheet interface for an interactive bill of materials (BOM). The main spreadsheet is divided into several sections:

- Parameters (Rows 2-4):** A table with columns for various parameters. Row 4 shows values: # of Channels? (8), # of 1 Bar Channels? (4), # of 2 Bar Channels? (4), Pneumatic Fitting Vendor (MettleAir), Housing? (Yes), Assembly Tools? (Yes), Currency? (USD \$), and Conversion Rate (1).
- Summary (Rows 6-8):** A table with columns: Grand Total, Electrical Assembly, Pneumatics Assembly, Housing, and Assembly Tools. Row 8 shows values: 5032, 1086, 3741, 164, and 41.
- Table Headers (Row 10):** Component, Vendor, Supplier, Vendor Number, Quantity, Extra, Cost/Unit (USD), USD \$, Subtotal, Universal Part Number, and Description.
- Table Body (Rows 11-22):** A list of components including Arduino Mega 2560, USB 2.0 A-B Male Cable, Nylon Spacers, Screw - Button Head, Nut - Hex, Washer, Board Terminal Block, Motherboard PCB, and Screw - Button Head - UNC.

Annotations and interactions are shown:

- Annotation 1:** A red box highlights the top-left corner of the spreadsheet.
- Annotation 2:** A red box highlights the Parameters section.
- Annotation 3:** A red box highlights the Summary section.
- Annotation 4:** A red box highlights the table headers.
- Annotation 5:** A red box highlights the table body.
- Annotation 6:** A red box highlights the bottom portion of the table body.
- Annotation 7:** A blue box highlights a zoomed-in view of the "# of Channels?" parameter, showing a list of values from 1 to 8.
- Annotation 8:** A dropdown menu for the "Vendor" column is open, showing a list of suppliers with checkboxes: (Select All), Amazon, CPI Automation, DigiKey Electronics, ELECROW, McMaster-Carr, MettleAir, and (Blanks).

Fig. 8. Tutorial image for interactive bill of materials.

The way universal part numbers is given to each component are listed below:

- Every component begins with “E,” “P,” “H,” or “T.” The “E” stands for electronics, and is given to every component in the Electronics Assembly, the “P” stands for pneumatics and is given to every component in the Pneumatics Assembly, the “H” stands for housing, and is given to every component in the Housing Assembly, and the “T” stands for tools and is given to every component in the Assembly Tools.
- The letter is then followed by a number, and is based on the top-down sequence in the bill of materials.
- Then, it can be either a dot “.”, an underscore “\_”, a hyphen “-”, or nothing. The dot is used when there is an equivalent replacement available. The underscore is used if there are subcomponents to the main component. The hyphen is used when more than one unit is required. A passive component or tool will not be followed by any number.
- If a number is followed by a dot, then it represents a specific option for that particular component. If a number is followed by an underscore, it represents a subcomponent. If a number is followed by a hyphen, then it represents a specific part of a multiple quantity component.
- Following this number, a hyphen and a number could exist to represent a particular connection or port for a specific component.

**Examples.** E1-A0: Arduino Mega port A0;

E2\_2-1-3: the third connection of the first 4-position board terminal block for the motherboard PCB;

E4.1: 1 bar output pressure sensor, the first option for output pressure sensors;

220 Ohm: 220 Ohm resistor;

H1\_3: washers to secure H1, which is the bottom corner brackets.

## 5. Build instructions

This  $\mu$ Pump consists of three assemblies, namely pneumatics, electronics, and housing. Please refer to ***uPump pneumatics assembly build guide.pdf***, ***uPump electronics assembly build guide.pdf***, and ***uPump housing assembly build guide.pdf***, for a step by step build guide for all three assemblies, respectively. Please refer to ***Build Guide Tutorial.pdf*** in the supplementary for an in-depth description of how to use each of the assembly build guides.

### 5.1. Build sequence

1. It is recommended to solder the required components onto the PCB board first. Please refer to the PCBA build guide section of ***uPump electronics assembly build guide.pdf***.
2. The next step is to build the pneumatics assembly, as it will help with positioning all the components. Please refer to ***uPump pneumatics assembly build guide.pdf***.
3. The following step is to build the electronics assembly; there is no need to disassemble the pneumatics assembly, merely add the electronics assembly onto the previously completed pneumatic assembly. Please refer to the electrical connections build guide section of ***uPump electronics assembly build guide.pdf***. The user can use the 3D CAD models to approximate the length of the wires. It is recommended that the user cut the wires longer than the approximations, and trim or cable tie the wires once the housing assembly is complete.
4. It is recommended that the user test the completed pneumatic and electronics assemblies with the GUI and software before building the housing assembly.
5. The last step is to build the housing assembly. There is no need to disassemble the pneumatics and electronics assemblies. Only disconnect the wires and tubing as needed to facilitate building the housing assembly. Please note that the walls of the housing assembly can be machined, milled or laser cut.

### 5.2. Safety during the build

Sharp tools such as scissors, wire cutters and wire strippers will be used. Therefore, precaution must be taken to avoid injury due to sharp tools. Also, appropriate safety precautions must be taken when fabricating the walls of the housing assembly, whether that be machining, milling, or laser cutting. For PCBA soldering, please practice using the various online tutorials or allow a person who is experienced to conduct the soldering. Safety glasses, laboratory gloves as well as closed-toe shoes are recommended for building a  $\mu$ Pump.

## 6. Operation instructions

Operating  $\mu$ Pump is intuitive and straightforward. Please refer to ***uPump operating instructions.pdf*** for a complete guide on testing, using, and maintaining it. Please also refer to the vendor or manufacturer’s websites for specification sheets via the link provided in the bill of materials for more detailed information on individual components.

### 6.1. Hardware operation overview

Please refer to Fig. 9 while reading the following descriptions. The back wall of the housing assembly has three ports, one for electrical power from the AC/DC adapter, one for air pressure from an external air pressure regulator, and one for USB connection from the user's PC. The user has the option to leave all three ports connected if  $\mu$ Pump is in a location for an extended period, or the user can connect/disconnect all three ports before/after using  $\mu$ Pump. The rocker switch on the back is the master switch for power, it can be left on at all times, but must be turned off during transportation and maintenance of  $\mu$ Pump as well as any time the top wall is removed.

The front wall of the housing assembly consists of a latching button for power, a toggle switch for turning on and off the inlet pneumatic solenoid valve, a series of toggle switches matching the number of pneumatic actuators, two LED indicators and a digital process meter that indicates the inlet pressure.

It is recommended to always turn on the software GUI and connect to the PC before turning on the hardware. After  $\mu$ Pump establishes a connection with the PC through the GUI, the user can click the power button to turn on the power. The built-in LED light in the power button will light up; the inlet process meter should also turn on, and after stabilizing, it should display "0" or very near "0" and the yellow LED is turned on and becomes solid. Then it is time to toggle on the solenoid valve and stabilize the inlet pressure to anywhere between 41 and 69 PSI, which is the required input pressure for the pneumatic actuators to function correctly. Other pneumatic actuator models may need different input pressure. Please note that both LED indicators will flash back and forth after the inlet pressure is above 10 PSI, but not within the required pressure range. Once the green LED indicator turns on and becomes stable, the system is ready to go (i.e. the input pressure is correct, the software and hardware function normally). At this point, the user can toggle on the actuators to begin using  $\mu$ Pump. A solid yellow LED indicator means that the system is powered off, or the inlet pressure is lower than 10 PSI. Please make sure the outlet tubing is connected and secure before increasing the inlet pressure via the GUI.

### 6.2. Software operation overview

Please refer to Fig. 10 while reading the following descriptions. Please copy and paste the entire folder containing **uPump\_PC.exe** to the desired location on the user's PC. Please do not remove any files, which are critical for the proper operation of **uPump\_PC.exe**, from the folder. Double right-clicking **uPump\_PC.exe** will open the GUI main window; the command window prompt is not required for operation and provides additional debugging feedback. The command window can be minimized to keep out of view.

In the main window, clicking "Setup" brings up a window where the user can select the COM port, the number of pump outputs, and maximum pressure [1 to 10 000 (mbar)]. The "Initialize" button will input all the parameters into the software, hide the setup window and open the control window. In the control window, the user can connect to  $\mu$ Pump, activate/pause communication with  $\mu$ Pump, view the outlet pressure supplied to each fluidic channel in real-time, independently control the outlet pressure for each fluidic channel, see the outlet pressure and record the outlet pressure for every channel in ".csv" file format. Left-clicking the "plotter" button in the main window opens a real-time plot window that shows the set pressure and actual pressure in each selected channel.

It is recommended to first zero all the outputs before closing the system. The user can close the control window directly or by pressing the "Disconnect" button. The program can be safely exited by using the "exit" button of the main window or the "x" of the main window bar.

### 6.3. Operation safety and maintenance

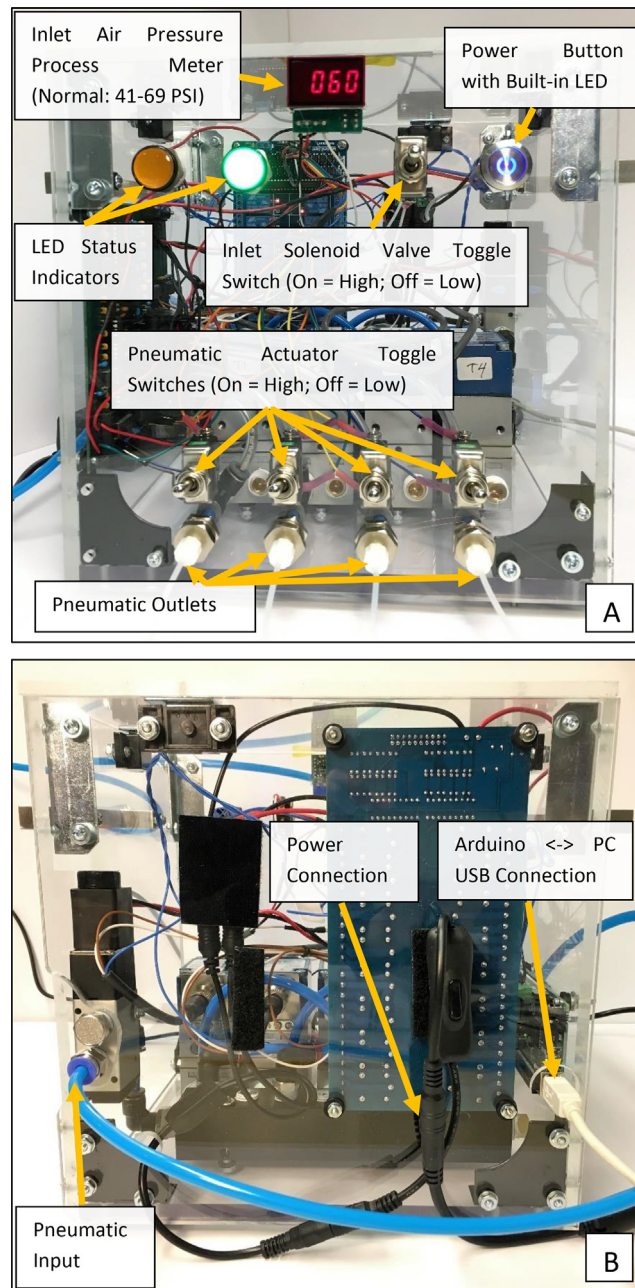
When using  $\mu$ Pump, it is critical to fully enclose the internal components, meaning that the top wall that is magnetically attached to the rest of the housing assembly must be on. This precaution is to ensure that the user is protected from the electrical components. Also, the user must make sure that all the tubing, both internal and external, are securely attached. This precaution is to protect the user as well as the internal equipment from pressure leakages and sudden pressure jumps.

The maintenance of  $\mu$ Pump can only be conducted with the main power on the back wall of the housing assembly being switched off. To test the Arduino to PC connections, the user simply connects the USB cable and uses **uPump\_Arduino\_test-Mode.ino**. Over time, some connections, especially the pin electrical connections, might become loose due to transportation and vibration. Therefore, it is recommended to check all the electrical connections if something goes wrong.

Over time, the digital process meter that displays the inlet air pressure, as well as the pneumatic actuators, might become less accurate due to longterm signal drift. Therefore, it is recommended that the user calibrate the digital process meter as well as the pneumatic actuators if accuracy becomes a problem. Lastly, individual malfunctioning components can be replaced if necessary.

## 7. Validation and characterization

A 4-channel  $\mu$ Pump built for University of Waterloo Microfluidic Laboratory is being used for various applications and studies. In one application it is being used as a standalone system for a study in continuous flow electrophoresis using



**Fig. 9.** Hardware operations image for (A) front wall and (B) back wall.

microchannels and pressure-driven flow for control. Another fundamental study in active-controlled droplet microfluidics uses  $\mu$ Pump as an embedded system as part of a feedback control system [13]. Lastly, AES Life Sciences, a company focused on protein characterization and analysis solutions is using a two-channel  $\mu$ Pump as an embedded system for various tests and applications.

As seen by the various applications mentioned above, the performance of  $\mu$ Pump is vital for many microfluidics applications. Below is a full performance characterization of the 4-channel  $\mu$ Pump using a high-end commercial system as a benchmark.

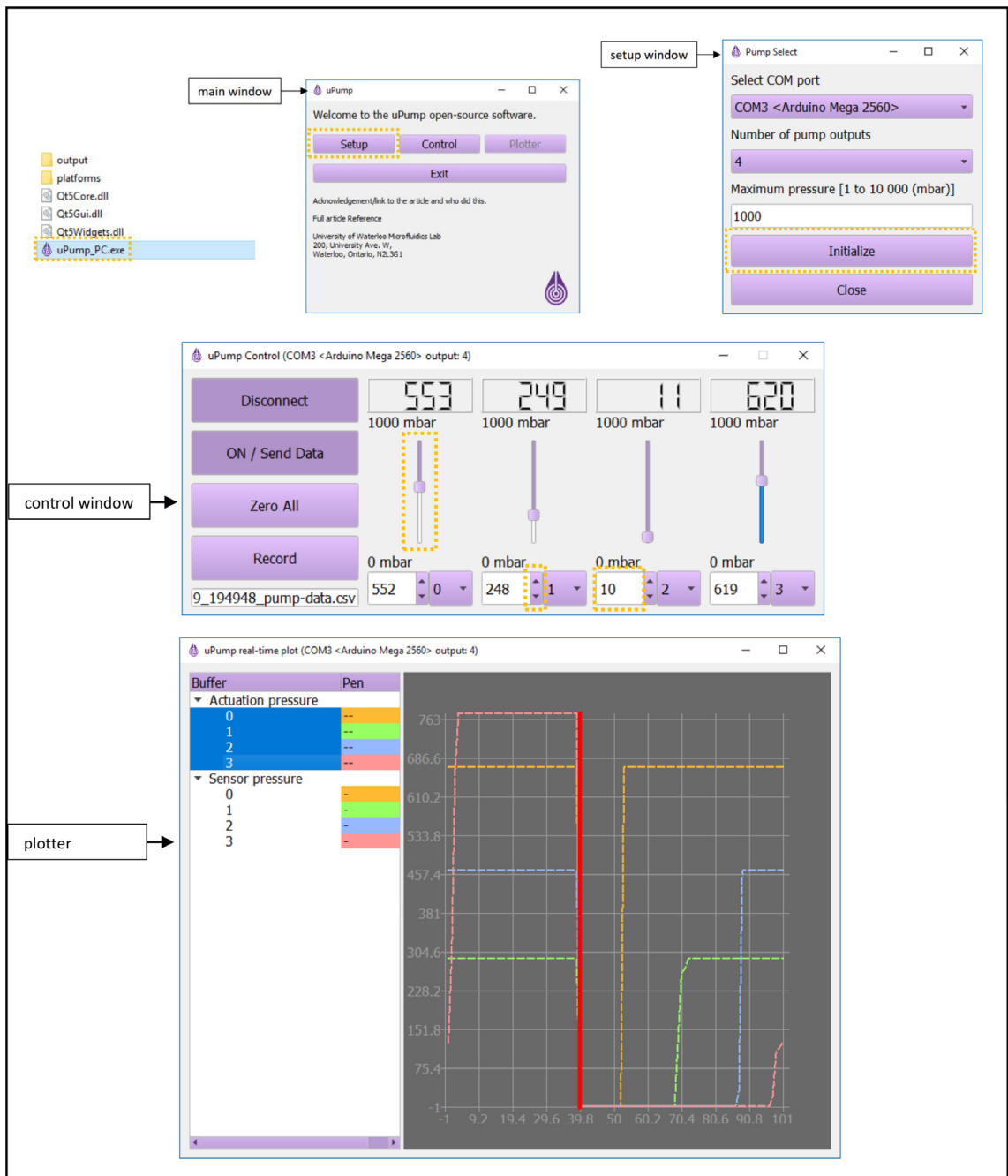


Fig. 10. Software operations image.

### 7.1. Methodology

Two tests were performed. The first test is to benchmark  $\mu$ Pump against a high-end commercial pump, Fluigent MFCS-EZ (0–2 bar), regarding performance under load, meaning that a real microfluidic chip and laboratory experimental setup is used. The second test is to validate  $\mu$ Pump performance in droplet microfluidic applications. All testing is done in an indoor laboratory environment with the temperature at a constant 22 °C and a constant humidity of 45% for the entire duration of the testing period.



### 7.1.1. Performance characterization setup

Three sets of identical tests conducted on different days are used to compare  $\mu$ Pump and Fluigent MFCS-EZ pump. Each set of the test consisted of testing the response time, hysteresis and long term stability for both pneumatic pressure and pressure-driven flow. The testing setup is shown in Fig. 11. An external high precision pneumatic pressure sensor (TE Connectivity U5300) and an external high precision liquid flow sensor (Sensirion SLG1430) are used to measure the pneumatic pressure and pressure-driven flow rate. The high precision liquid flow sensor is calibrated for 20 cSt silicone oil using a high precision syringe pump (Harvard Apparatus Model 33) with a 1 ml syringe (BD 309628).

A simple T junction microfluidic chip made out of Polydimethylsiloxane (PDMS) via soft lithography technology (Fig. 12) is used to test the pressure-driven flow performance. Silicone oil with a viscosity of 20 cSt (25 °C) (Sigma-Aldrich 63148-62-9) is used as the single-phase fluid for testing. The PDMS is thermally treated on a 95 °C hotplate for three days before testing, which renders its surface to be hydrophobic for wetting the silicon oil. A single microfluidic chip is used for all the performance tests. After each day of testing, the microfluidic chip is cleaned by flushing the channels with isopropyl alcohol, then blow-dried, and placed on a 95 °C hotplate for at least 12 hours before the next set of tests.

During testing,  $\mu$ Pump or Fluigent MFCS-EZ pump pressurizes the liquid reservoir containing the silicone oil via an air tubing. Under pressure, the silicone oil traverses a plastic tubing to reach the high precision liquid flow sensor, then into the microfluidic chip and out into the output container. The high precision pneumatic pressure sensor is connected with the liquid reservoir on the air tubing side. All the external sensors and the two pneumatic pumps interface with the PC during the entire testing period.

### 7.1.2. Droplet microfluidic validation setup

Three sets of tests, each on a different day, are used to quantify the quality of droplet formation in the squeezing regime where the capillary number, a dimensionless ratio of viscous force versus interfacial tension force, is less than 0.002 ( $Ca < 0.002$ ). For more detailed information about droplet formation regimes, readers are referred to an excellent article by Christopher *et al.* [16]. The droplet volume is a good indication for droplet microfluidic system performance and consistency. Therefore, the goal of this validation test is to determine whether the  $\mu$ Pump can achieve similar performance for droplet volume consistency compared to that achieved by the Fluigent MFCS-EZ pump under the same settings. The secondary objective is to compare with the previously published dataset [17] to further validate the consistency of pressure-driven flow using two different models. Two chips of the same design (Fig. 13) are used, one used for two sets of tests, while the other is used for one set of tests. The cross-flow configuration is used for droplet generation, as shown in Fig. 13.

Stable droplet formation requires the carrier fluid to preferably wet the PDMS chips. For water-in-oil droplet formation, PDMS chips should be superhydrophobic to wet the carrier fluid, oil. PDMS is hydrophobic by nature; however, it requires surface treatment to be superhydrophobic. A surface treatment technique has been developed in-house to ensure the contact angle to be as high as 170°. Briefly, the microfluidic chip is baked on a hot plate at 95 °C for three days to make it highly hydrophobic before testing. During the testing, no wetting issues were observed for making water-in-oil droplets. Before testing, the whole chip is primed with 20 cSt silicone oil for 1 h. Two high precision liquid flow sensors are used: one for measuring the carrier fluid (oil) and the other for measuring the droplet fluid (water). Please refer to Fig. 14 for an image of the setup. A microscope, Nikon Ti-E coupled with Zyla 5.5 sCMOS camera from Andor, is used to visualize the droplets and capture images; image processing determines the top-down area and the tip-to-tip length of the droplets to calculate droplet volume. Please refer to **Image processing.pdf** in the supplementary for an in-depth description of the image processing steps.

Moreover, the hydraulic resistance is approximated for the entire system including the inlet liquid tubing, liquid flow sensor, microfluidic chip, and the outlet liquid tubing by using the gauge pressure difference between the output pressure sensor in  $\mu$ Pump and atmosphere, and the reading on the high precision liquid flow sensor. Seven different pressure settings (100 mbar to 700 mbar with 100 mbar increments) are used to induce various flow rates while satisfying the squeezing regime criterion ( $Ca < 0.002$ ). The inlet and outlet liquid tubing and flow sensors' internal capillary all have known circular cross-section and length dimensions; their hydraulic resistance is calculated using Eq. (3) [18], then subtracted from the total hydraulic resistance calculated from Eq. (2) to get the microchannel hydraulic resistance by itself. The average microchannel hydraulic resistance is then used to back-calculate the channel height based on Eq. (2). Lastly, the droplet volume is calculated using Eq. (4) [19].

$$R_{hyd} = \frac{P}{Q} \quad (2)$$

where  $R_{hyd}$  is the hydraulic resistance [ $\frac{kg}{m^4 \cdot s}$ ],  $P$  is the pressure [ $\frac{kg}{m \cdot s^2}$ ], and  $Q$  is the flow rate [ $\frac{m^3}{s}$ ].

$$R_{hyd} = \frac{12\mu L}{wh^3} \left[ 1 - \frac{h}{w} \left( \frac{192}{\pi^5} \sum_{n=1,3,5}^{\infty} \frac{1}{n^5} \left( \frac{n\pi w}{2h} \right) \right) \right] \quad (3)$$

where  $R_{hyd}$  is the hydraulic resistance [ $\frac{kg}{m^4 \cdot s}$ ],  $\mu$  is the dynamic viscosity [ $\frac{kg}{m \cdot s}$ ],  $L$  is the length of the rectangular fluid channel [ $m$ ],  $w$  is the height of the rectangular fluid channel [ $m$ ], and  $h$  is the height of the rectangular fluid channel [ $m$ ].

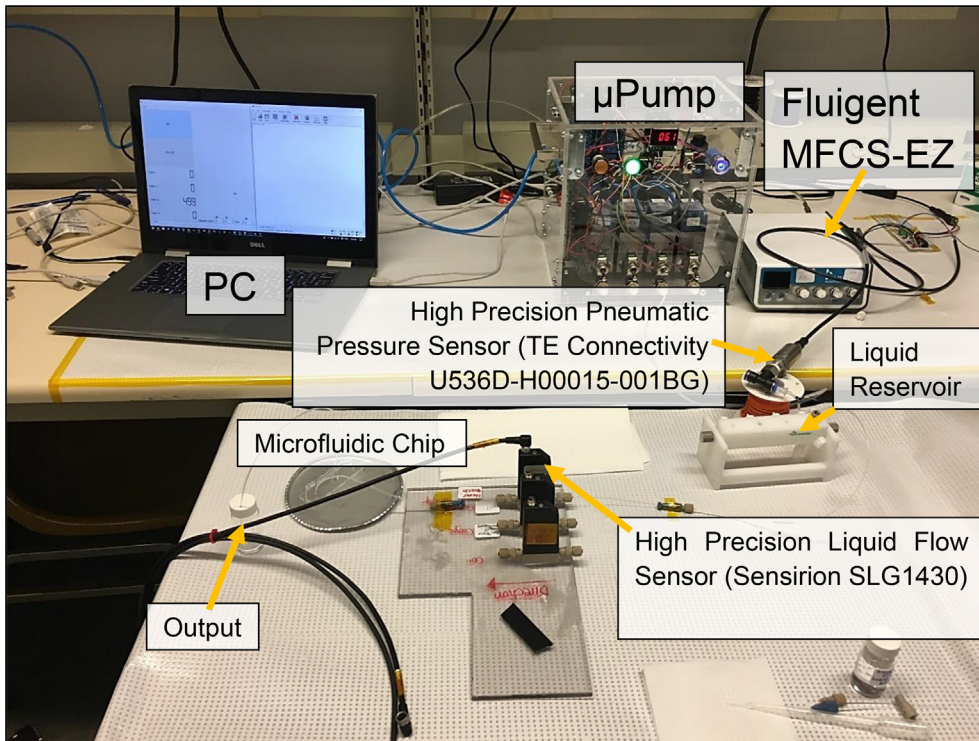


Fig. 11. Performance characterization setup.

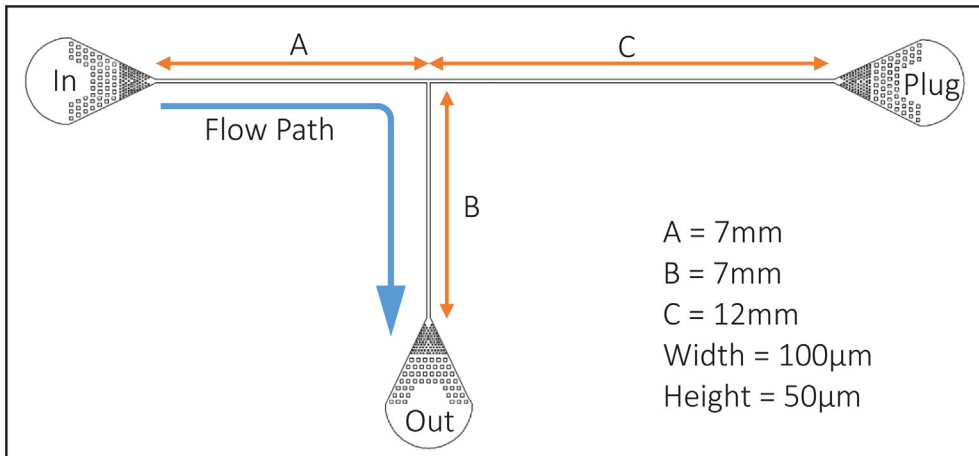


Fig. 12. Schematic of the microfluidic chip used for performance characterization.

$$V_d = h_m A_d - \frac{h_m^2}{2} \left(1 - \frac{\pi}{4}\right) l_d \quad (4)$$

where  $V_d$  is the droplet volume [ $\text{m}^3$ ],  $h_m$  is the height of the droplet [ $\text{m}$ ],  $A_d$  is the 2-D top-view area of the droplet [ $\text{m}^2$ ], and  $l_d$  is the 2-D top-view perimeter of the droplet [ $\text{m}$ ].

## 7.2. Pressure-driven flow performance

Pressure response, flow response, hysteresis and stability are the parameters used to compare the performance of  $\mu\text{Pump}$  and Fluigent MFCS-EZ pump, which are both a four output channel system. Only the results for one channel is shown below, as each channel exhibits nearly identical performances. The pressure and flow step responses for  $\mu\text{Pump}$  exhibit more oscil-

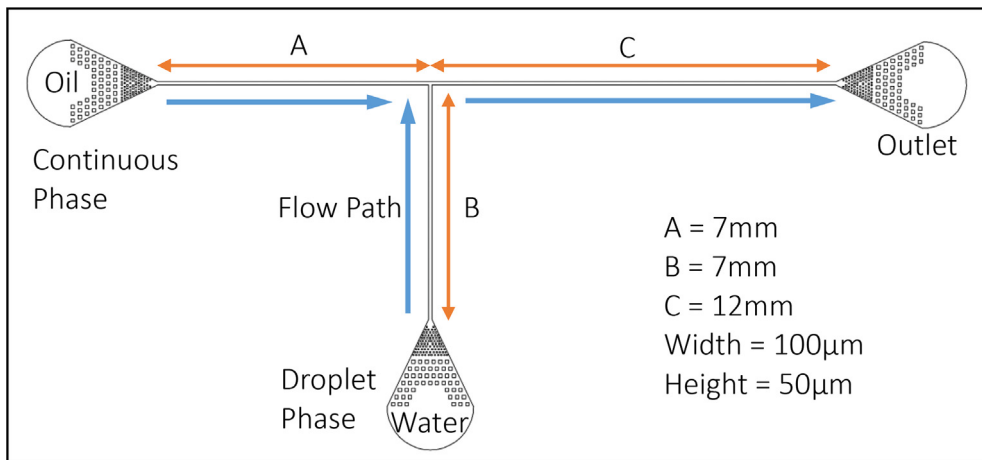


Fig. 13. Schematic of the chip used for droplet microfluidics validation.

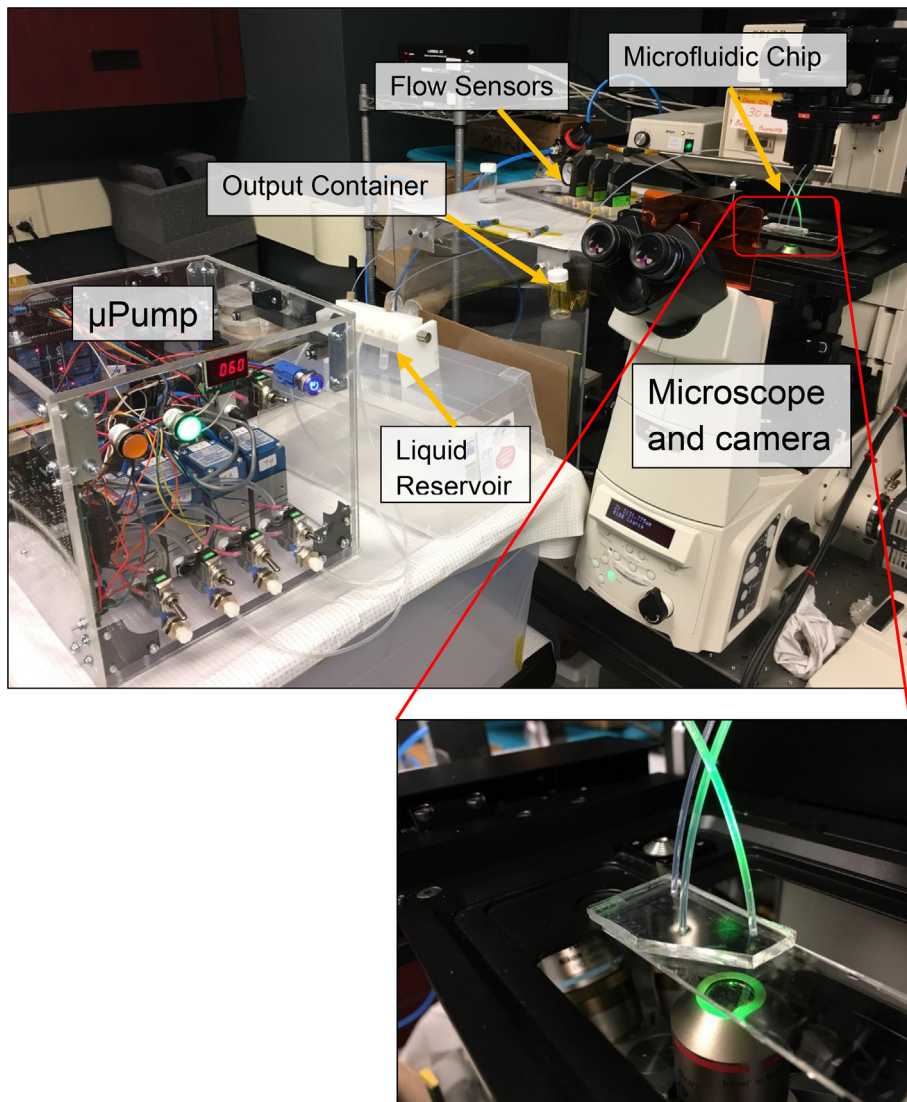
lations and overshoot than Fluigent MFCS-EZ pump, as seen in Fig. 15. The pressure and flow hysteresis for  $\mu$ Pump are better than that of Fluigent MFCS-EZ pump as shown in Fig. 16. The pressure stability for  $\mu$ Pump is worse than that of Fluigent MFCS-EZ pump as seen in Fig. 17a. However, the flow stability as seen in Fig. 17b, when the liquid (20 cSt silicone oil) is pushed inside the microchannels is practically identical. The input pressure to  $\mu$ Pump is set to a constant 60 psi, which is in the accepted range of 40–70 psi for the pressure actuators, while the input pressure to Fluigent MFCS-EZ pump is set to a constant 2.3 bar per manufacturer specification.

The control algorithm of the pneumatic actuator and the quality of its electrical input signal are potential reasons for the pressure oscillations, overshoot, and instability exhibited by  $\mu$ Pump. In the context of electrical signals, “quality” is defined as the signal to noise ratio (SNR) of the analog signal. Sources of low- and high-frequency noise include but are not limited to: electromagnetic interferences, wire and connection imperfections, and component cross-talk. Nevertheless, the liquid flow stability is more representative of the real-world performance, and  $\mu$ Pump performs nearly identically to Fluigent MFCS-EZ. The difference between the fluid flow and pressure output performance attributes to the viscous damping effects occurring both in the deformable air tubing and the microchannels. Hence, the performance criteria for the open-source system are satisfied as the performance is on par with the premium commercial system. Potential improvements to increase the SNR to reduce oscillations and increase stability include: shielding the PCBA from electromagnetic interference, finding better analog filter cutoff frequencies to further attenuate the noises while preserving signals, and redesigning the PCBA to better separate the digital IC chips and the analog IC chips along with better insulation of analog electrical signal traces from the digital electrical signal traces.

### 7.3. Droplet validation

The performance of a pressure pump can be characterized by the consistency of droplet formation in a microfluidic chip where the fluids are driven by the pressure pump. It has been found that under certain operating conditions, the generated droplet volume has a linear relationship with the flow rate ratio between the droplet fluid and carrier fluid, as seen in Fig. 18. The y-axis is the dimensionless droplet volume ( $V_{d,exp}^*$ ) defined as the volume of a droplet over the height of the channel times the square of the channel width of the carrier fluid, and the x-axis is the flow rate ratio ( $\phi_{exp}$ ). Fig. 18a shows the published experimental data [17], where the Fluigent MFCS-EZ pump was used, and Fig. 18b shows the experimental data obtained using both the Fluigent MFCS-EZ pump and the 4-channel  $\mu$ Pump presented here. A prior published study is chosen for comparison for the droplet validation to prove the repeatability of generating droplets using a high precision fluid control system. Fig. 18a is chosen due to the fact that it is in the same droplet squeezing regime ( $Ca < 0.002$ ), using the same carrier fluid (20 cSt silicone oil), the same droplet fluid (water), the same type of T-junction microfluidic chip (expanded upon below) and material (PDMS), and conducted using the same Fluigent MFCS-EZ pump as the current testing setup, allowing for standardized comparison and benchmark across droplet microfluidic system for the purpose of real-world droplet microfluidics performance comparison between  $\mu$ Pump and Fluigent MFCs-EZ.

It is noted that the data represented by the squares, ( $\square$ ), in Fig. 18a are comparable to that presented in Fig. 18b because of their similar geometric conditions. Specifically, the aspect ratio (channel height to width) and width ratio (width ratio of the droplet fluid channel to carrier fluid channel) are 0.4 and 1, respectively, in Fig. 18a, and 0.5 and 1, respectively, in Fig. 18b. Very similar linear relationships between the droplet volume and the flow rate ratio are observed in Fig. 18a and b, which validates the overall performance of  $\mu$ Pump. There are day-to-day as well as chip-to-chip variations as shown in Fig. 18b. This is mainly due to the inevitable, slight differences between different batches of chip fabrication and varying



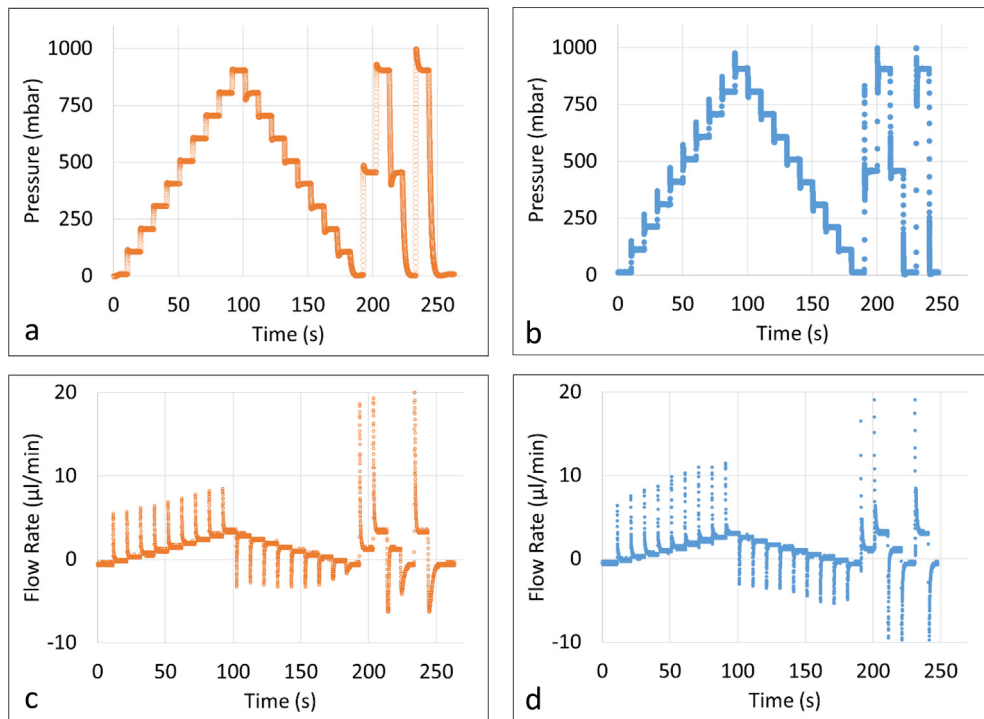
**Fig. 14.** Droplet microfluidics validation setup photo image.

extent of oil absorption by PDMS chips. Also, slight differences are often observed between different PDMS chips for their reaction to the heating treatment that is developed to ensure surface wettability. Stable droplet formation requires the drop fluid and carrier fluid to preferably wet the PDMS chip with a contact angle above  $135^\circ$ . A heating technique has been developed for surface treatment, which however often results in slightly different wettability due to environmental conditions such as room temperature and humidity [20]. The differences between Fluigent MFCS-EZ pump and our  $\mu$ Pump are minimal. Fig. 19 shows an example of the droplets produced by our  $\mu$ Pump. Please refer to Table 5 for  $\mu$ Pump's specifications.

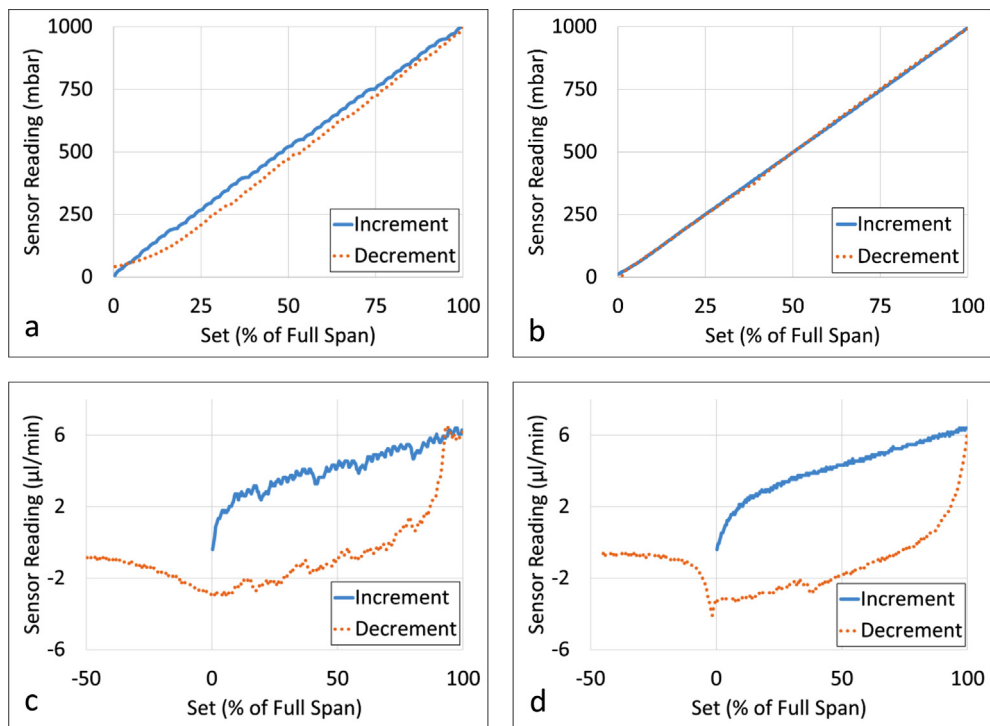
#### 7.4. Specification and limitations

Below is a list of limitations of  $\mu$ Pump:

- The pneumatic actuators have an integrated feedback controller; hence, the user does not have control over the control algorithm.
- The pressure stability is slightly worse than the benchmarked high-end commercial system, Fluigent MFCS-EZ, which could be more significant for microchannel gaseous flow applications.
- The user needs to become familiar with the system and build it. Afterwards, they must personally maintain  $\mu$ Pump.
- Changes to the GUI or any part of the software require coding skills.

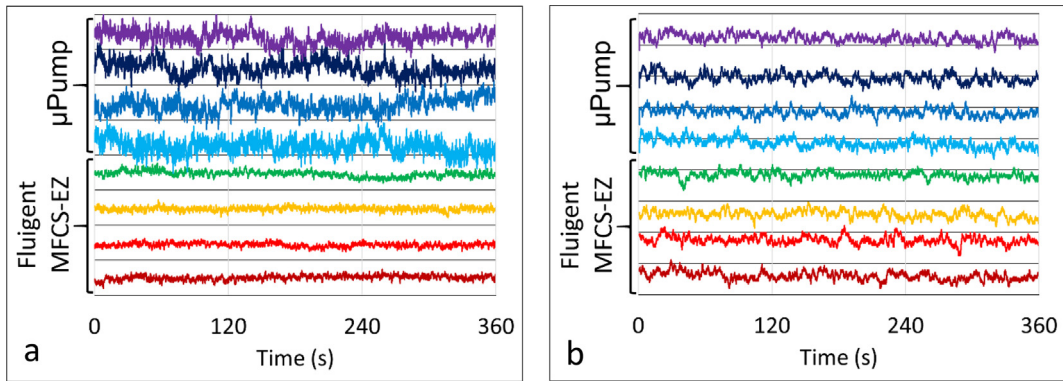


**Fig. 15.** Pressure (a) and flow (c) step response under load for Fluigent MFCS-EZ, and pressure (b) and flow (d) step response under load for  $\mu\text{Pump}$ .

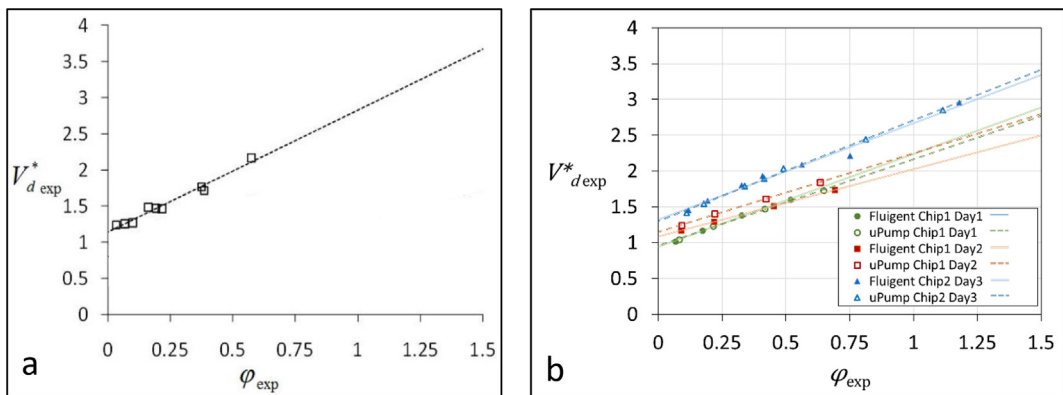


**Fig. 16.** Pressure (a) and flow (c) hysteresis under load for Fluigent MFCS-EZ, and pressure (b) and flow (d) hysteresis under load for  $\mu\text{Pump}$ .

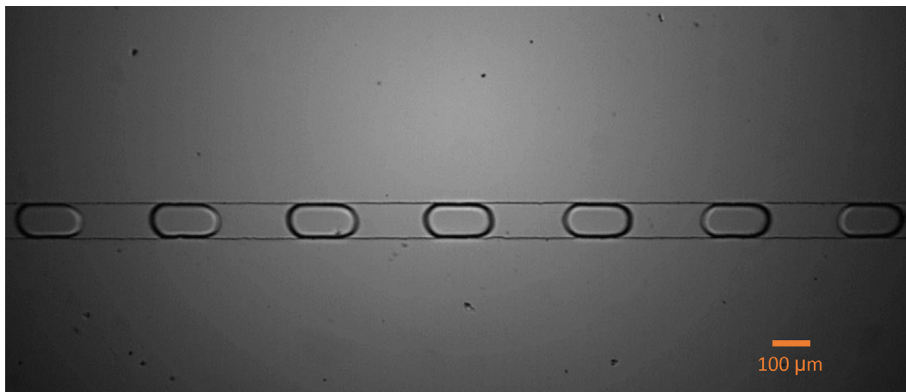
If the user desires to have control over the control algorithm, the authors suggest changing the pressure actuator from the ControlAir to another option that allows tuning of the control algorithm, for example, this open-source pressure regulator [11]. For microchannel gaseous flow applications, a flow damper could be implemented to increase pressure stability, which



**Fig. 17.** Pressure (a) and flow (b) stability under load for  $\mu$ Pump and Fluigent MFCs-EZ. The pressure stability (a) plot has a y-axis scale of 1 mbar per grid line and the flow stability (b) plot has a y-axis scale of 0.1  $\mu$ l/min per grid line.



**Fig. 18.** Dimensionless droplet volume vs. flow rate ratio of published data (a) [15] and  $\mu$ Pump and Fluigent MFCs-EZ (b).



**Fig. 19.** Photo image of droplets produced by  $\mu$ Pump.

**Table 5**  
 $\mu$ Pump specifications.

Parameter	Value	Description
Pressure accuracy (% of full span)	0.09	1 standard deviation
Pressure stability (% of full span)	0.02	1 standard deviation
Pressure resolution (% of full span)	0.02	12 bit DAC gives a resolution of 4096 levels
Settling time (seconds)	<2	Time to reach max pressure from zero and stabilize. Output volume dependent.

would somewhat restrict flow rate depending on the size of the flow damper, but the flow rate provided by the pressure actuators should be sufficient for most microchannel gaseous flow applications. Maintaining  $\mu$ Pump is fairly straight forward; the most common problem that occurs is an electrical connection or pneumatic fitting getting loose during transportation of  $\mu$ Pump, which should be readily fixable. If the user desires to change the GUI or any part of the open-source software, please refer to the numerous online resources that teach coding in Qt and Arduino, which are the framework for the GUI, and Arduino microcontroller, respectively.

### Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

### Acknowledgements

The authors would like to declare that Run Ze Gao and Marie Hébert have contributed equally to the development of  $\mu$ Pump.

This work was carried out thanks to grants from the Natural Sciences and Engineering Research Council of Canada (NSERC) and Ontario Centres of Excellence (OCE) provided to Prof. Carolyn L. Ren. A special thanks go to AES Life Sciences for their contribution to the OCE grant.

The authors acknowledge NSERC and the University of Waterloo for scholarship funding in the form of Canada Graduate Scholarships – Masters and Doctoral (CGS-M and CGS-D). The authors would also like to thank the Waterloo Institute of Nanotechnology (WIN) for providing Nanofellowship scholarships.

The authors would like to thank Terry Huang for helping with the characterization of the  $\mu$ Pump. The authors would also like to thank Anna Nguyen for helping in the laboratory as well as Will Baxter for helping on  $\mu$ Pump's software

### Appendix A. Supplementary data

Supplementary data to this article can be found online at <https://doi.org/10.1016/j.ohx.2020.e00096>.

### References

- [1] S. Sevim et al, Self-assembled materials and supramolecular chemistry within microfluidic environments: From common thermodynamic states to non-equilibrium structures, *Chem. Soc. Rev.* 47 (11) (2018) 3788–3803.
- [2] S.C. Kim, I.C. Clark, P. Shahi, A.R. Abate, Single-Cell RT-PCR in microfluidic droplets with integrated chemical lysis, *Anal. Chem.* 90 (2) (2018) 1273–1279.
- [3] M. Shang, R.H. Soon, C.T. Lim, B.L. Khoo, J. Han, Microfluidic modelling of the tumor microenvironment for anti-cancer drug development, *Lab Chip* 19 (3) (2019) 369–386.
- [4] A. Przekwas, M.R. Somayaji, D.E. Ingber, R. Prantil-Baun, D. Das, R. Novak, Physiologically Based Pharmacokinetic and Pharmacodynamic Analysis Enabled by Microfluidically Linked Organs-on-Chips, *Annu. Rev. Pharmacol. Toxicol.* 58 (1) (2018) 37–64.
- [5] Y.H. Kwok et al, A dual fuel microfluidic fuel cell utilizing solar energy and methanol, *J. Power Sources* 409 (2019) 58–65.
- [6] P.M. Korczyk, O. Cybulski, S. Makulska, P. Garstecki, Effects of unsteadiness of the rates of flow on the dynamics of formation of droplets in microfluidic systems, *Lab Chip* 11 (1) (2011) 173–175.
- [7] C. Kieffer, S. Ritty, T. Boudot, N. Petit, J. Weber, A. Le Nel, A high-precision fluid handling system based on pressure actuation: multi-inlets flow rate control, in: 3rd European Conference on Microfluidics, 2012, pp. 1–4.
- [8] J.R. Lake, K.C. Heyde, W.C. Ruder, Low-cost feedback-controlled syringe pressure pumps for microfluidics applications, *PLoS One* 12 (4) (2017) 1–12.
- [9] K. Brower et al, An open-source, programmable pneumatic setup for operation and automated control of single- and multi-layer microfluidic devices, *HardwareX* 3 (2018) 117–134.
- [10] J.A. White, A.M. Streets, Controller for microfluidic large-scale integration, *HardwareX* 3 (2018) 135–145.
- [11] C. Watson, S. Senyo, All-in-one automated microfluidics control system, *HardwareX* 6 (2019) 1–10.
- [12] D. Wong, C.L. Ren, Microfluidic droplet trapping, splitting and merging with feedback controls and state space modelling, *Lab Chip* 16 (17) (2016) 3317–3329.
- [13] M. Hébert, M. Courtney, C.L. Ren, Semi-automated on-demand control of individual droplets with a sample application to a drug screening assay, *Lab Chip* 19 (8) (2019) 1490–1501.
- [14] ControlAir Inc., "Type 900X Miniature I/P, E/P Transducer Installation, Operation, and Maintenance Instructions," 2016.
- [15] NXP, "MPX4250D Data Sheet," 2017.
- [16] G.F. Christopher, N.N. Noharuddin, J.A. Taylor, S.L. Anna, Experimental observations of the squeezing-to-dripping transition in T-shaped microfluidic junctions, *Phys. Rev. E – Stat. Nonlinear, Soft Matter Phys.* 78 (3) (2008) 1–12.
- [17] T. Glawdel, "Droplet production and transport in microfluidic networks with pressure driven flow control," 2012.
- [18] H. Bruus, *Theoretical Microfluidics*. 2008.
- [19] V. Van Steijn, C.R. Kleijn, M.T. Kreutzer, Predictive model for the size of bubbles and droplets created in microfluidic T-junctions, *Lab Chip* 10 (19) (2010) 2513–2518.
- [20] T. Glawdel, C. Elbuku, C.L. Ren, Droplet formation in microfluidic T-junction generators operating in the transitional regime. I. Experimental observations, *Phys. Rev. E – Stat. Nonlinear, Soft Matter Phys.* 85 (1) (2012) 1–9.