BoSL FAL pump: A small, low-cost, easily constructed, 3D-printed peristaltic pump for sampling of waters

David T. McCarthy *, Baiqian Shi, Miao Wang, Stephen Catsamas

BoSL Water Monitoring and Control, Department of Civil Engineering, Monash University, Victoria 3800, Australia

ABSTRACT

Water sampling is an essential undertaking for water utilities and agencies to protect and enhance our natural resources. The high variability in water quality, however, often necessitates a spatially distributed sampling program which is impeded by high-cost and large sampling devices. This paper presents the BoSL FAL Pump - a low-cost, easily constructed, 3D-printed peristaltic pump which can be made from commonly available components and is sized to suit even the most space constrained installations. The pump is 38 mm in height and 28 mm in diameter, its components cost $19 AUD and the construction time is just 12 min (excluding 3D printing times). The pump is driven by a direct current motor which is commonly available, cheap and allows for flexibility in the energy supply (5–12 V). Optionally, the pump has a Hall effect sensor and magnet to detect rotation rates and pumping volumes to improve the accuracy of pumping rates/volumes. The pump can be easily controlled by commonly available microcontrollers, as demonstrated by this paper which implements the ATmega328P on the Arduino Uno R3. This paper validates the pump for long-term deployments at flow rates of up to 13 mL per minute in 0.14 mL volume increments at accuracy levels of greater than 99%. The pump itself is scalable, allowing for a wider range of pumping rates when, for example, large volume samples are required for pathogen and micropollutant detection.

© 2021 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 | BoSL FAL-Pump |
|---------------|--------------|
| Subject area  | Engineering and Material Science  |
|               | Chemistry and Biochemistry        |
|               | Medical (e.g. Pharmaceutical Science) |
|               | Biological Sciences (e.g. Microbiology and Biochemistry) |
|               | Environmental, Planetary and Agricultural Sciences |
| Hardware type | Biological sample handling and preparation |
|               | Field measurements and sensors     |
|               | Electrical engineering and computer science |
| Open Source License | CC-BY-4                   |
| Cost of Hardware | $28 AUD; $19 AUD without an Arduino Uno R3 |
| Source File Repository | https://doi.org/10.17632/prb2wzr77y.1 |

* Corresponding author.  
E-mail address: david.mccarthy@monash.edu (D.T. McCarthy).  

https://doi.org/10.1016/j.ohx.2021.e00214  
2468-0672/© 2021 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

Monitoring of water quality is essential for the protection and enhancement of human and ecosystem health [1]. For example, disease causing microbes (i.e. pathogens) are routinely monitored in water distribution systems [2], water recycling plants [3,4] and recreational waterways [5] to ensure their levels are below required levels to protect public health risks. Monitoring water quality is challenging because pollutant concentrations can vary significantly [6], meaning our methods must be spatially distributed (i.e. monitor multiple points in a system) and temporally representative (i.e. continuously monitor with time).

Significant improvements have been made to online and real time sensors that sit within the waterway and measure water quality [7,8]. The advantage is that they can continuously and rapidly monitor for water quality changes, ensuring a very high temporal resolution. Furthermore, these sensors are becoming cheaper and easier to maintain, allowing for higher spatial disaggregation of our water systems. However, these sensors are unable to reliably and accurately detect all types of water quality pollutants, especially pathogens and micropollutants [9]. As such, the physical process of taking water samples and then analyzing these either in the field or back in the laboratory is still common practice.

The design of water monitoring regimes has received a significant amount of research attention, especially regarding the number and frequency of samples required to adequately characterize a pollutant at an individual site [10]. Indeed, Harmel et al. [11] show that frequent sampling of stormwater over rainfall events is required to adequately characterize pollutant levels at any given site, while Ort et al. [12] have shown that similarly frequent samples are required for characterization of wastewaters. As multiple samples are often required for such characterization, auto-sampling equipment is commonly installed to take water from the system at regular intervals, either spaced by time or flow volumes [11].

Research has also demonstrated the benefits of monitoring multiple locations in a system and that Near Source Tracking (NST [13]) can detect, understand and remedy pollution sources. For example, wastewater-based epidemiology (WBE [14]) has been applied at sewage treatment plants to help understand SARS-CoV-2 infections in the community, yet the real advantages of WBE only comes when sampling is done at a scale that is suited to targeted public health agency responses (i.e. sampling at allotment or building scales) [13,15].

Sampling at high temporal and spatial resolution is commonly done with automatic sampling devices that sit on the surface of the catchment and sample water using a single pump which either distributes the samples among multiple individual bottles or automatically creates one time or flow weighted composite sample [16]. These autosamplers are usually costly and are bulky (e.g. [17]). While some are waterproof, many agencies are reluctant to install these in open environments in case they are vandalized or stolen (especially in remote communities), therefore they also require expensive monitoring huts. Furthermore, as demonstrated by the recent COVID-19 pandemic, the stock availability of automatic sampling equipment can sometimes be scarce with many agencies being unable to access this equipment [18]. These points demonstrate the need for more easily available equipment that can be readily made and assembled by water managers and utilities around the globe.

To enable monitoring at high spatial and temporal resolution, cheap, easy to maintain and readily available sampling methods are required. A key part of delivering such methods is the development of smaller, robust, commonly available and cheap pumping systems that are capable of sampling waters. Peristaltic pumps are commonly used for environmental sampling [11] as they avoid contact with the water [19] and they do not macerate the sample as much as other pumping solutions. While there are many commercially available pumps and autosamplers on the market [17], there are very few cheap, open source and open hardware options specifically developed for sampling of waters in the environments. The majority of open source pumping solutions either focus on multichannel laboratory requirements [20], are large /bulky [19,21], cannot accurately measure pumping rates or peristaltic pump rotations and/or have not had their pumping rates properly tested (e.g. see [22]). We believe there is a significant gap in the literature and in online opensource repositories for a well calibrated and tested, low cost and easy to construct peristaltic pumping system.

As such, this paper presents an open hardware, open source, peristaltic pumping system that is small, cheap and quick to construct, and only relies on easily available equipment and consumables. Once waterproofed, the pump itself is small enough to be installed directly into creeks, stormwater drains and wastewater sewers, enabling widespread and remote deployment without the need for expensive equipment to be installed, maintained and protected from vandalism or damage.

2. Hardware description

The BoSL FAL Pump was designed for the sampling of waters. It can be applied to sample from drinking water systems, stormwater drains, wastewater sewers and open water environments including creeks, rivers, bays and oceans. This pump offers many advantages over traditional water sampling systems; it is extremely low cost, easily constructed from readily available parts and can be installed without the need for expensive equipment to be installed, maintained and protected from vandalism or damage.

The BoSL FAL Pump was designed for the sampling of waters. It can be applied to sample from drinking water systems, stormwater drains, wastewater sewers and open water environments including creeks, rivers, bays and oceans. This pump offers many advantages over traditional water sampling systems; it is extremely low cost, easily constructed from readily available parts and can be installed without the need for expensive equipment to be installed, maintained and protected from vandalism or damage.
available materials, has a small footprint and future users can easily scale the system to suit a range of pumping requirements. Furthermore, we believe the pump could be used in other applications, including:

- laboratory sampling and pumping systems, such as being used to filter water through membranes or transfer water from one container to the next;
- chemical dosing systems for a range of practical applications (like pH balancing systems or chlorine dosing);
- dosing of annular reactor devices to study biofilms; and,
- feeding microfluidic devices with chemicals or water matrices.

The BoSL FAL Pump (Fig. 1, Fig. 2) is comprised of a 3D-printed casing and rotating wheel, a geared motor, peristaltic tubing and a fuse. Optional components can include a Hall effect sensor and a magnet to count the rotations and pumping rates.

---

**Fig. 1.** Components of the BoSL FAL Pump: (a) 3D-printed outer casing and lid, (b) 3D-printed rotating wheel with three rollers held in place by 1.8 mm diameter 13 mm long metal rods, (c) 5-12v direct current geared motor, (d) Hall effect sensor and magnet to count pump rotations and estimate pumping rates and volumes (optional), (e) 2 mm internal diameter, 4 mm external diameter, 100 mm long peristaltic tubing, (f) 50 mL centrifuge tube, or equivalent, used to house the pump and hold the tubing in place, and (g) an electrical control system that could either be: (1) an Arduino Uno R3 with an appropriate power supply that includes a suitable MOSFET to control the pump OR (2) another alternative that has a suitable microcontroller, switch, relay or MOSFET OR (3) a continuous power supply connected directly to the motor, combined with an appropriate 500 mA fuse and holder.

**Fig. 2.** Assembly of the pump: rotating wheel including pins and rollers (a, b, c), installation of the optional magnet (d), placement of motor in lid (e), connection of rotating wheel and motor (f), insertion of the motor and wheel assembly into outer casing (g), installation of the peristaltic tubing and the optional Hall effect sensor (see insert) (h) and installation of the pump into a 50 mL centrifuge container to hold tubes in place (i).
Fig. 3. Wiring diagram to power and operate the pump using an Arduino Uno R3. Left: graphical illustration. Centre: component pinouts. Right: schematic diagram. A0 on the Arduino is used to measure magnetic field changes via the Hall effect sensor (if included), D6 on the Arduino is used to power this sensor, D9 (grounded via 1 k Ohm resistor) is used to switch the MOSFET which then powers the fused DC motor.

Fig. 4. Setup of the validation tests. Top-left: setup for Test 1, showing the direct connection of the motor to a voltage supply. Top-right: setup for Test 2, showing the connection of the motor to the control circuitry (i.e. the MOSFET and then the Arduino Uno R3). Bottom: setup for Tests 3 and 4, showing the connection of the motor as per Test 2 but with the addition of the magnet and Hall effect sensor to count pump rotations.

Fig. 5. Pumping rates (mL/min) for each voltage tested.
The pump requires either a power supply to continuously operate the pump, or a microcontroller and relays/MOSFET to operate the pump either based on time or other inputs. This article focuses on using an Arduino Uno R3 to operate and control the pump. The size of the main 3D-printed casing is 28 mm in diameter and 38 mm high. There is no requirement for glue or screws unless the user wishes to improve the design for longer term deployments; indeed, all components clip and clamp together easily, hence not requiring any tools.

3. Design files

| Design file name                  | File type   | Open source license | Location of the file                        |
|-----------------------------------|-------------|---------------------|---------------------------------------------|
| Fig. 1                            | figure      | Public domain       | available with the article                  |
| Fig. 2                            | figure      | Public domain       | available with the article                  |
| Fig. 3                            | figure      | Public domain       | available with the article                  |
| Fig. 4                            | figure      | Public domain       | available with the article                  |
| Fig. 5                            | figure      | Public domain       | available with the article                  |
| Fig. 6                            | figure      | Public domain       | available with the article                  |
| BoSL_FAL_Pump.stl                  | STL         | CC BY 4.0           | https://doi.org/10.17632/prb2wzr77y.1       |
| BoSL_FAL_Pump.skp                  | Sketchup file| CC BY 4.0           | https://doi.org/10.17632/prb2wzr77y.1       |
| BoSL_FAL_Pump_1.ino                | Arduino IDE code | CC BY 4.0       | https://doi.org/10.17632/prb2wzr77y.1       |
| BoSL_FAL_Pump_Dataset.xlsx         | Microsoft Excel | CC BY 4.0        | https://doi.org/10.17632/prb2wzr77y.1       |

Fig. 1. A figure showing the components of the pump.
Fig. 2. A figure showing the method of assembly.
Fig. 3. A figure showing the wiring diagram for the pump.
Fig. 4. A figure showing the setup of the validation experiments for the BoSL FAL Pump.
Fig. 5. A figure showing pumping rates (mL/min) for each voltage tested.
Fig. 6. A figure showing the relationship between number of pump rotations and the measured flow rate from the pump.

BoSL_FAL_Pump.stl. A Standard Tessellation Language (STL) ready to print component file for the pump.
BoSL_FAL_Pump.skp. An original Sketchup file used to create the STL file.
BoSL_FAL_Pump.ino. An Arduino IDE file displaying the code used to control the pump.
BoSL_FAL_Pump_Dataset.xlsx. A spreadsheet containing validation data.
## 4. Bill of materials

| Designator                              | Component Description                                                                 | Number | Cost per unit AUD | Total cost AUD | Source of materials                                      | Material type        |
|-----------------------------------------|---------------------------------------------------------------------------------------|--------|-------------------|----------------|---------------------------------------------------------|----------------------|
| Outer casing, lid, rotating wheel, rollers | 1.76 mm 3D printing filament                                                          | 11.91 g | $0.04 AUD         | $0.48 AUD      | [https://www.bilby3d.com.au/DispProd.asp?ProdID=PLA175Purple1](https://www.bilby3d.com.au/DispProd.asp?ProdID=PLA175Purple1) | PLA                  |
| Motor                                   | 1.8 mm diameter, 13 mm high, pins for rollers                                         | 3      | $0.13 AUD         | $0.39 AUD      | [https://www.amazon.ca/Steel-Round-Stock-Lathe-Tools/dp/B012T6E710](https://www.amazon.ca/Steel-Round-Stock-Lathe-Tools/dp/B012T6E710) | Stainless steel or equivalent |
| Motor                                   | Direct current, geared (298:1), 5v-12v motor                                          | 1      | $6.79 AUD         | $6.79 AUD      | [https://www.aliexpress.com/item/32991622456.html](https://www.aliexpress.com/item/32991622456.html) | Direct current motor  |
| Control system                          | Arduino Uno R3(or equivalent)                                                         | 1      | $9.51 AUD         | $9.51 AUD      | [https://www.aliexpress.com/item/4000587244657.html](https://www.aliexpress.com/item/4000587244657.html) | Circuit board        |
| Control system                          | MOSFET – IRF520N                                                                      | 1      | $0.33 AUD         | $0.33 AUD      | [https://www.aliexpress.com/item/32832806692.html](https://www.aliexpress.com/item/32832806692.html) | IC                   |
| Optional: Hall effect sensor            | Hall effect sensor, A1309KUA-9-T                                                       | 1      | $3.05 AUD         | $3.05 AUD      | [https://www.digikey.ca/en/products/detail/allegromicrosystems/A1309KUA-9-T/6821587](https://www.digikey.ca/en/products/detail/allegromicrosystems/A1309KUA-9-T/6821587) | Sensor               |
| Optional: Hall effect sensor            | Magnet, 3 mm diameter, 2 mm high                                                       | 1      | $1.03 AUD         | $1.03 AUD      | [https://www.digikey.ca/en/products/detail/comusinternational/m1219-2/11562464](https://www.digikey.ca/en/products/detail/comusinternational/m1219-2/11562464) | Neodymium Iron Boron |
| Peristaltic tubing                      | Silicone tubing, 2 mm ID, 4 mm OD                                                     | 100 mm | $0.0037 AUD       | $0.37 AUD      | [https://www.amazon.ca/Gikfun-Silicone-Flexible-Transparent-Peristaltic/dp/B08H1ZD5VZ?ref=sr_1_6](https://www.amazon.ca/Gikfun-Silicone-Flexible-Transparent-Peristaltic/dp/B08H1ZD5VZ?ref=sr_1_6) | Silicone             |
| Fuse                                    | 500 mA 2AG                                                                            | 1      | $0.16 AUD         | $0.16 AUD      | [https://www.digikey.com/short/bwfyqbdp](https://www.digikey.com/short/bwfyqbdp) | Fuse                 |
| Fuse Holder                             | 2AG In-Line Fuse Holder                                                               | 1      | $2.46 AUD         | $2.46 AUD      | [https://www.digikey.com/short/m7584z0j](https://www.digikey.com/short/m7584z0j) | Brass and Plastic Polypropylene |
| Container                               | 50 mL centrifuge tube, 28 mm internal diameter                                        | 1      | $0.72 AUD         | $0.72 AUD      | [www.amazon.ca/Plastic-Centrifuge-Membrane-Solutions-Pyrogenic/dp/B07FDN43XV/](www.amazon.ca/Plastic-Centrifuge-Membrane-Solutions-Pyrogenic/dp/B07FDN43XV/) |                     |

### 4.1. Build instructions

**3D printing.** The design file (BoSL_FAL_Pump.skp) is exported to a Standard Tessellation Language file (BoSL_FAL_Pump.stl) which is then loaded into the printer software (FlashPrint 4.0.0, FlashForge). Polylactic acid (PLA) plastic 1.75 mm in diameter can be used, but we have also had success with other printing materials including polyethylene terephthalate glycol (PET-G) and Acrylonitrile butadiene styrene (ABS). We use a FlashForge CreatorPro with FlashPrint 4.0.0 software and for ABS the printing parameters are: 100% fill density, hexagon pattern, combining every 2 layers, first layer height 0.2 mm,
3 perimeter shells, 4 top solid layers, 4 bottom solid layers, layer height 0.12 mm, 50 mm/s print speed, 70 mm/s travel speed, extruder temperature 210 °C, bed temperature 55 °C and no wall or raft was used. Under these conditions, the print takes 2 h and 15 min. After printing, all parts are cleaned from excess printing material using a sharp edge (e.g. scissors). Most importantly, the rollers (Fig. 1b) and the internal section of the outer casing are thoroughly cleaned (Fig. 1a).

Rotating wheel. The stainless-steel rods fit into the holes on the bottom and top of the rotating wheel and they should enter easily, without force (clean the holes if force is required or check the rods’ diameters). The rods are placed into either the top or bottom of the wheel (e.g. Fig. 2a) and then the rollers are slid over each bar (Fig. 2b). The top and bottom of the wheel are carefully aligned and then slowly squeezed together (Fig. 2c); once joined, the rods are not visible, and the rollers turn with ease. If rotation or pumping rates are to be measured, insert the magnet into the hole provided and ensure it is well seated as shown in Fig. 2d.

Pump assembly. The motor slides into the lid (Fig. 2e) and the motor shaft then slides tightly into the hole of the rotating wheel as shown in Fig. 2f. The motor and rotating wheel assembly should slide easily into the case and if not remove and ensure all 3D printed material is smooth. The lid clips securely onto the outer casing (Fig. 2g). The peristaltic tubing is wrapped around the rotating wheel (Fig. 2h); to do this either (1) pull the lid off the outer casing and tightly wrap the tubing around the rollers and re-connect the lid and outer casing or (2) wet the end of the tubing and push the tubing into the pump as the roller is slowly rotated using your thumb. To count rotations or pumping rates, insert the Hall effect sensor into the hole provided in the lid and ensure it is well seated as shown in Fig. 2h.

Pump container. To hold the peristaltic tubing in place during operation, a small force must be exerted on the outside of the intake tubing and no force exerted on the outlet tubing. This force can be achieved by using tape to hold the tubing in place. However, variability in the pressure of the tape can change the functioning of the pump, and hence to standardize the pressure on the tubing we sized the outer casing of the pump to apply the right amount of pressure by sliding it into a 50 mL centrifuge tube (28 mm internal diameter). The case was also carefully designed to have one restricted opening for the inlet and a wide opening for the tubing outlet (Fig. 2i). The inlet tubing feeds from the right-hand side of the pump and, if included, the hall effect sensor according to the wiring diagram in Fig. 3. In brief, the motor is powered by the 5v output from the Arduino Uno R3 and is grounded using the specified MOSFET which is switched using digital pin 9 on the Arduino. A one kilohm resistor pulls digital pin 9 to ground. The Hall effect sensor (if included) is powered using digital pin 6 on the Arduino, is grounded and has its output recorded on analog pin 0 of the Arduino. To ensure our paper can be applied by a wide range of enthusiasts, we demonstrate the functionality of the BoSL FAL pump using jumper-based wiring. However, we also provide enough detail for those who would like to make more formal control components, including more elegant and robust circuit boards.

5. Operation instructions

The operation of the BoSL FAL Pump is done using an Arduino Uno R3, programmed using the Arduino IDE. It is assumed the reader is familiar with the Arduino IDE and the uploading of code to the board. For further instructions on how this is done, please refer to https://www.arduino.cc/en/Guide. In this current setup, the pump only operates in the clockwise direction, but it could be adapted to have a H bridge or multiple MOSFETs to allow switching of the pump’s direction. This could be useful, for example, when purging of lines is required. For this case-study, if the pump operates in an anti-clockwise direction, simply switch polarity of the motor.

To pump continuously, the user sets digital pin 9 on the Arduino to high. This can be done through the Arduino IDE and uploading the following code to the correct port. This continuous pumping mode does not need the Hall effect sensor and can also be run independently from the Arduino if the user has an appropriate power supply (between 3 and 12 V).

```
Void setup() digitalWrite(9,HIGH);
Void loop();
```

We have created well commented code for more advanced pumping regimes: BoSL_FAL_Pump.ino located in the Source file repository (https://doi.org/10.17632/prb2wzr77y.1). The code is flexible to whether the hall effect sensor and magnet are included. If they are included, then the pump’s operation is based on the number of pump rotations that occur in user-defined intervals, else it is based on a duration of pumping every user-defined interval. For the remainder of this example operation, it is assumed that the Hall effect sensor and magnet is installed as this is the more complex use-case.

At the start of BoSL_FAL_Pump.ino the user defines three variables: (1) PumpEveryXMins, the number of minutes to wait between pump operation, (2) NumberOfSpins, the number of rotations the pump makes each time the pump operates, and (3) DurationOfRun, the number of minutes that the pump repeats the sequence. For example, if the user wishes to make the
pump rotate 10 times every 30 min for an entire day, then the following should be defined: PumpEveryXMins = 30; NumberOfSpins = 10; DurationOfRun = 1440.

The program then uses these variables to operate the pump. It begins by calibrating the Hall effect sensor and magnet by running the function “GetMinsMaxs(“) to determine the characteristics of each individual pump and the background conditions for each pump installation. The function “GetMinsMaxs(“) rotates the pump at least three times while the Hall effect sensor continuously reads magnetic field strength; these values are then used to set minimum and maximum magnetic field thresholds which are used to measure pump rotations. Assuming the tubing remains constant in its internal area, each rotation is equivalent to pumping a certain volume of water (see Validation below). Once calibrated, the pump then repeats (via the “loop(“) function a sequence of pump operation (via the “SpinMe(“) function, where the desired number of rotations occur) and resting periods (to ensure the desired wait time elapses) until the desired operational run time is completed.

7. Validation and characterization

The operation of the BoSL FAL Pump was characterized and validated using four tests (Fig. 4 shows the setup for all four tests): (1) continuous pumping at different directly-supplied voltages, (2) intermittent timed-based pumping with timed pumping periods using the Arduino Uno R3, (3) pulse-based pumping where the rotation rate of the pump is measured using sensors and the Arduino Uno R3 and (4) intermittent pulse-based pumping where the number of rotations is measured and pumping is intermittent. Prior to each test, the pump’s tubing was primed with water and all tests were carried out with room temperature drinking water. All raw data and analyses for the four tests are included in the Source Repository.

Test 1. Continuous pumping. In this test, the pump was directly powered, without an Arduino, by either an adjustable external power supply (PA-30150 W-ZMX, Zozo, China) (Fig. 1, top-left) or directly from an Arduino Uno R3. Four voltages were tested: 5 V (from either the external power supply or the Arduino Uno), and 7.5 V, 9 V and 12 V from the external power supply. All voltages and currents were checked at the start and end of each test using a multimeter (FSK-830D+, Q-MING, China). The pump was operated for a period of 60 s at 5 and 7.5 V and 20 s at 9 and 12 V. During each pumping period, the volume of water pumped was measured by a 0.2 mL graduated cylinder (10 mL Oral Syringe, Terumo, Philippines). The process was repeated three times at each voltage, with these datapoints used to estimate mean, 5th and 95th percentiles.

Using the external power supply, the voltage was measured to be roughly in line with the manufacturer’s specifications, with only a slightly higher voltage being measured (0.2 to 0.5 V). The current drawn from the supply varied approximately linearly depending on the supply voltage, ranging from 105 mA at 5 V to 140 mA at 12 V. The supply voltage from the Arduino Uno R3 was measured to be between 4.9 and 5.0 V and the current drawn ranged from 80 to 120 mA.

Under continuous operation, the BoSL FAL Pump was able to sample water at a mean rate of 4.97 mL/min (5th percentile = 4.79 mL/min; 95th percentile = 5.15 mL/min) when powered by 5 V from the external supply and at a mean rate of 13.8 mL/min (5th percentile = 13.40 mL/min; 95th percentile = 13.80 mL/min) when supplied with 12 V (Fig. 5). Continuous operation of the BoSL FAL Pump over an entire day would result in collecting between 7 and 20 L of water when operating at 5 and 12 V respectively. This could be beneficial when large sample volumes are required for sample assay (e.g. pathogen assays).

The relationship between the applied voltage and the pumping rate was close to linear over the range tested in this paper (Fig. 5). However, a noticeable drop in the pumping rate was observed when switching from the external power supply 5 V to that of the Arduino Uno R3, even though the measured supply voltage was just 0.29 V lower. This could be caused by the limitation in the Arduino Uno R3 to supply high loading. Indeed, the current draw was seen to increase to over 180 mA as the pump rotates when all three rollers apply pressure simultaneously (this only occurs in about 15% of the rotation period). It is also very likely that the motor is operating close to its maximum torque at this voltage, and so any slight decrease in voltage will result in a significant reduction in motor rotation. As such, we suggest that the BoSL FAL Pump is always operated at or above 5 V capable of delivering a current well in excess of 180 mA.

Test 2. Intermittent timed-based pumping. The pump without the Hall effect sensor or magnet was operated by the Arduino Uno R3 (Fig. 4, top-right) which was programmed with the BoSL_FAL_Pump.ino code provided in the repository. The code was modified for this test as follows: Line 12 was commented out, Line 17 was uncommented, on Line 17 the DurationOfPumping was set to 10 (10 s) and on Line 21 the PumpEveryXMins was set to 1 (1 min). The program ran for 10 min. During each 10 s pumping period, the period of pumping was timed and then the volume of pumped water was measured. The duration of resting was also measured. This process was repeated three times, resulting in 30 individual measurements of pumping duration, pumping volume and resting duration. This data was then used to estimate the relative uncertainty in the measured pumping rates and the timing intervals following that of previous studies [23].

The measured durations of pumping and resting were slightly different than what was programmed (30 s of pumping and 30 s of resting). Indeed, the mean pumping and resting period was measured to be 29.97 s, with a 5th percentile of 29.91 and a 95th percentile of 30.02 s. While much of this variation is likely caused by human error while operating the stopwatch, it should also be noted that the Arduino Uno R3 used in this example does not have a real time clock configured, meaning that the timing kept by the device is not accurate. This will be further explored in subsequent tests. As a result of these errors, each test lasted for less than 10 min in duration: 9 min 59.33 s, 9 min 58.98 s and 9 min 59.66 s. While these deviations seem small, they can accumulate over long term operation; for example, extrapolating these values to a full day would result in
each test ending between 47 and 147 s early. Using this data, we estimate that the relative uncertainty of the clock is ± 0.2% of the required time period. If high precision timing is required for your applications, it is strongly encouraged to install a real time clock for your microcontroller.

The measured volume of water pumped in the 10 s pumping period ranged from 1.6 mL to 1.8 mL with an average of 1.75 mL across the three tests and 10 measurement increments. This equates to an average pumping rate of 3.50 mL/min, with a 5th percentile of 3.46 and a 95th percentile of 3.54 mL/min. This results in a relative uncertainty in the pumping volume/rate of ± 1.2%.

**Test 3. Pulse-based pumping.** The pump with the Hall effect sensor and magnet was operated by the Arduino Uno R3 (Fig. 4, bottom) which was programmed with the BoSL_FAL_Pump.ino code provided in the repository. To evaluate the volume of water pumped during each rotation, a series of tests were conducted where **NumberOfSpins** was varied (on Line 12): 10, 20, 40. The volume of water pumped was measured as per previous tests. This process was repeated three times at each value of **NumberOfSpins**. A long-term test was also conducted after the above tests, where the tubing was replaced and then run continuously for 24 h at 5 V. On Day 2, after this extended run time, the volume of water pumped was again measured for 10 rotations (repeated 10 times).

The volume of water pumped was directly proportional to the number of pump rotations, with 10 rotations pumping exactly 1.4 mL, 20 rotations pumping exactly 2.8 mL, and 40 rotations pumping exactly 5.6 mL (Fig. 6). This provides confidence that each rotation of the pump samples 0.14 mL of water, meaning that the lowest reliable sample volume increment is 0.14 mL. In this case, specific volumes of water can be transferred by the pump; if 1.4L needs to be moved, then the **NumberOfSpins** needs to be set to 10,000. The mean rotation rate of the pump is 2.2 s, meaning this transfer would take approximately 6 h. However, a higher supply voltage would reduce this significantly (noting that here we validate the Pulse-based pumping only at 5v, but we have also seen positive results when operating it at 12v).

The results of the test conducted on Day 2 (i.e. after the long-term test) confirmed the above results with a mean pumping volume of 1.38 mL for 10 rotations, equating to a volume per rotation of 0.14 mL (like the above). This test also provides confidence that the results are repeatable (i.e. the pump was rebuilt with new tubing).

**Test 4. Intermittent pulse-based pumping.** The pump with the Hall effect sensor and magnet was operated by the Arduino Uno R3 (Fig. 4, bottom) which was programmed with the BoSL_FAL_Pump.ino code provided in the repository. In this test, two changes were made to the code: the **NumberOfSpins** was set to 10 on and **PumpEveryXmins** was set to 1 (1 min). During each pumping period, the period of pumping was timed, the number of rotations was counted and then the volume of pumped water was measured. The duration of resting was also measured. Each test was 10 min in duration. The entire process was repeated three times, resulting in 30 individual measurements of pumping duration, pumping volume and resting duration. This data was then used to estimate the relative uncertainty in the measured pumping rates and the uncertainty in timing intervals following that of previous studies (CITE ME). A long-term test was also conducted after the above tests, where the tubing was replaced and then run continuously for 24 h at 5 V. On Day 2, after this extended run time, the above test was repeated.

The duration of each test was slightly less than 10 min, again likely because of the limited time-keeping capability of the Arduino UNO R3. The uncertainty in the timing is estimated to be 0.2% of the time period required, which concurs well with that found in Test 2.

During the initial runs (i.e. on Day 1), the mean volume of water pumped was 1.39 mL (5th percentile = 1.38 mL, 95th percentile = 1.40 mL), resulting in a relative uncertainty in the pumping volume of ± 0.7%. This is almost half of the uncertainty measured when the pump was operated only on a timed basis, demonstrating the extra accuracy provided by counting the number of pump rotations. On average, the pumping took 21.52 s to conduct the 10 rotations and the pump had rest periods that were on average 37.82 s long. On Day 2, after the pump’s rebuild and long-term test, the mean volume of water pumped was 1.38 mL (5th percentile = 1.36, 95th percentile = 1.40 mL).

**Summary.** We have characterized and validated the BoSL FAL Pump for the sampling of waters: (1) continuously, (2) intermittently based on timed pumping periods and (3) intermittently based on pulsed pumping.

- **Continuous pumping using external power supply (i.e. plug in wall).** Under the design and conditions provided in this paper, we estimate that the minimum voltage required to operate the pump is 5 V and will draw approximately 100 mA, resulting in a power consumption of 0.5mAh during its operation. The start-up current is slightly higher, noting at around 160 mA. At a higher voltage, the pump can reach almost 14 mL/min, resulting in almost 20 L of water being sampled across an entire day of continuous operation.
  - Minimum supply voltage: 5v
  - Current draw at minimum: 100 mA
  - Pumping rate at minimum (5V supply): 4.97 mL /min
  - Maximum supply voltage: 12v (N.B. continuous operation at this voltage may overheat the motor)
  - Current draw at maximum: 147 mA
  - Pumping rate at maximum (12V supply): 13.6 mL/min

- **Timed pumping with intermittent operation using Arduino UNO R3.** While continuous operation could be a feasible use case, often sampling is done intermittently using either time or volume based spacing. In this paper, we explored the potential for the BoSL FAL Pump to be operated intermittently using an Arduino UNO R3. Under the test conditions pro-
vided, we estimate that the pumping rate was a mean 3.5 mL per minute of operation and that the relative uncertainty in pumping volumes was ± 1.2%. Furthermore, the timing kept by the Arduino Uno R3 was found to have an uncertainty of ± 0.2% of the period, meaning that an hour of operation could run for 1 h and 6 s or 59 min and 54 s.

- **Pulsed pumping using Arduino Uno R3, a Hall effect sensor and a magnet.** While pumping for set durations resulted in moderate uncertainties (see above – 1.2%), more accurate volumes could be pumped when counting the pump’s rotations using the optional Hall effect sensor and magnet. Indeed, under the tested conditions, we estimate that the uncertainty in the pumping volume is just 0.7% when using the Hall effect sensor and magnet to count rotations, almost half of that when we used a timed operation. Furthermore, we estimate that this pump could deliver accurate volume increments of just 0.14 mL, a value which was maintained even after rebuilding the pump and tubing replacement. However, while extremely accurate pumping volumes were achieved, the errors in timing introduced by the time keeping ability (±0.2%) of the Arduino Uno R3 should also be considered when conducting long-term sampling.

- Mean pumping rate of 3.5 mL/min
- Uncertainty in pumping rate of ± 1.2%
- Uncertainty in timing was ± 0.2% of period (e.g. an hour operation will result in up to 6 s error, a day operation will result in up to 2.4 min in error)

### Human and animal rights

NA.

### 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

This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

### References

[1] S. Liu, D. Ryu, J. Webb, A. Lintern, D. Waters, D. Guo, A. Western, Characterisation of spatial variability in water quality in the Great Barrier Reef catchments using multivariate statistical analysis, Mar. Pollut. Bull. 137 (2018) 137–151.
[2] H. Murphy, M. Thomas, P. Schmidt, D. Medeiros, S. McFadyen, K. Pintar, Estimating the burden of acute gastrointestinal illness due to Giardia, Cryptosporidium, Campylobacter, E. coli O157 and norovirus associated with private wells and small water systems in Canada, Epidemiol. Infect. 144 (7) (2016) 1355–1370.
[3] A.J. Hamilton, V.L. Versace, F. Stagnitti, P. Li, W. Yin, P. Maher, K. Hermon, R.R. Premier, D. Jerodiaconou, Balancing environmental impacts and benefits of wastewater reuse, WSEAS Trans. Environ. Dev. 2 (2) (2006) 117–129.
[4] H.M. Murphy, Z. Meng, R. Henry, A. Deletic, D.T. McCarthy, Current stormwater harvesting guidelines are inadequate for mitigating risk from Campylobacter during nonpotable reuse activities, Environ. Sci. Technol. 51 (21) (2017) 12498–12507.
[5] C. Schang, R. Henry, P.A. Kolotelo, T. Prosser, N. Crosbie, T. Grant, D. Cottam, P. O’Brien, S. Coutts, A. Deletic, Evaluation of techniques for measuring microbial hazards in bathing waters: a comparative study, PLoS One 11 (5) (2016) e0155848.
[6] B. Shi, P.M. Bach, A. Lintern, K. Zhang, R.A. Coleman, L. Metzelting, D.T. McCarthy, A. Deletic, Understanding spatiotemporal variability of in-stream water quality in urban environments–a case study of Melbourne, Australia, J. Environ. Manage. 246 (2019) 203–213.
[7] J.-B. Burnet, É. Sylvestre, J. Balbert, S. Imsbert, P. Servais, M. Prévost, S. Dorner, Tracking the contribution of multiple raw and treated wastewater discharges at an urban drinking water supply using near real-time monitoring of β-d-glucuronidase activity, Water Res. 164 (2019) 114869.
[8] S.A. Jaywant, K.M. Arif, A comprehensive review of microfluidic water quality monitoring sensors, Sensors 19 (21) (2019) 4781.
[9] M. Hassard, F. Lundy, L. Singer, A.C., Grimsley, J. Di Cesare, M., Innovation in wastewater near-source tracking for rapid identification of COVID-19 in schools. The Lancet Microbe.
[10] C. Ort, M.G. Lawrence, J. Reungoat, J.F. Mueller, Sampling for PPCPs in wastewater systems: comparison of different sampling modes and optimization strategies, Environ. Sci. Technol. 44 (16) (2010) 6289–6296.
[11] R. Harmel, K. King, R. Slade, Automated storm water sampling on small watersheds, Appl. Eng. Agric. 19 (6) (2003) 667.
[12] S.A. Jaywant, K.M. Arif, A comprehensive review of microfluidic water quality monitoring sensors, Sensors 19 (21) (2019) 4781.
[13] H.M. Murphy, Z. Meng, R. Henry, A. Deletic, D.T. McCarthy, Current stormwater harvesting guidelines are inadequate for mitigating risk from Campylobacter during nonpotable reuse activities, Environ. Sci. Technol. 51 (21) (2017) 12498–12507.
[14] C. Schang, C., Crosbie, N.D., Nolan, M., Poon, R., Wang, M., Jex, A., Scales, P., Schmidt, J., Thorley, B.R., Henry, R., Kolotelo, P., Langeveld, J., Schilperoort, R., Shi, B., Einsiedel, S., Thomas, M., Black, J., Wilson, S., McCarthy, D.T., 2020, Passive sampling of viruses for wastewater-based epidemiology: a case-study of SARS-CoV-2, ResearchGate Preprint Server, DOI:10.13140/RG.2.2.241338.39367.
[15] D. McCarthy, D. Harmel, D., Quality assurance/quality control in stormwater sampling, in Quality Assurance & Quality Control of Environmental Field Sampling. p. 98-127.
[16] HACH. AS950 Peristaltic Samplers. 2020 [cited 2021 6th January 2021]; Available from: https://www.hach.com/as950-peristaltic-samplers/family?productCategoryid=35547137069.
[18] Foley, K.E., So many communities are testing poop for Covid-19, equipment is running out, in QUARTZ. 2020, QUARTZ: https://qz.com/1910540/wastewater-testing-for-covid-19-is-straining-supply-chains/.
[19] D.A. Mucciarone, H.B. DeJong, R.B. Dunbar, Autonomous underwater pumping system, HardwareX 8 (2020) e00154.
[20] A. Jonsson, A. Toppi, M. Dufva, The FAST Pump, a low-cost, easy to fabricate, SLA-3D-printed peristaltic pump for multi-channel systems in any lab, HardwareX 8 (2020) e00115.
[21] D.A. Mucciarone, R.B. Dunbar, Automated multiport flow-through water pumping and sampling system, HardwareX 8 (2020) e00147.
[22] thingiverse.com. Digital designs for physical objects. 2021 [cited 2021; Search term: peristaltic pumps]. Available from: https://www.thingiverse.com/search?q=peristaltic+pumps&type=things&sort=relevant.
[23] D.T. McCarthy, A. Deletic, V.G. Mitchell, T.D. Fletcher, C. Diaper, Uncertainties in stormwater E. coli levels, Water Res. 42 (6-7) (2008) 1812–1824.

David McCarthy My research area is integrated water management, with specialised skills in urban hydrology, stormwater management, green water treatment technologies, city scale modelling, sensor networks and water quality processes (especially regarding pollutants of concern to human health). Much of my work focusses on the harvesting and reuse of urban stormwaters, greywaters and wastewaters.