Supplementary Information (SI) for Digital Discovery.

Supplementary Information:

Archerfish: A Retrofitted 3D Printer for General High-throughput Combinatorial Experimentation

Alexander E. Siemenn[1*], Basita Das[1], Eunice Aissi[1], Fang Sheng[1], Lleyton Elliott[1], Blake Hudspeth[1], Marilyn Meyers[1], James Serdy[1], Tonio Buonassisi[1*]

[1]Department of Mechanical Engineering, Massachusetts Institute of Technology, Cambridge, MA 02139, USA

[*]Corresponding authors: {asiemenn@mit.edu, buonassisi@mit.edu}

Schematics and Materials

Block Diagram

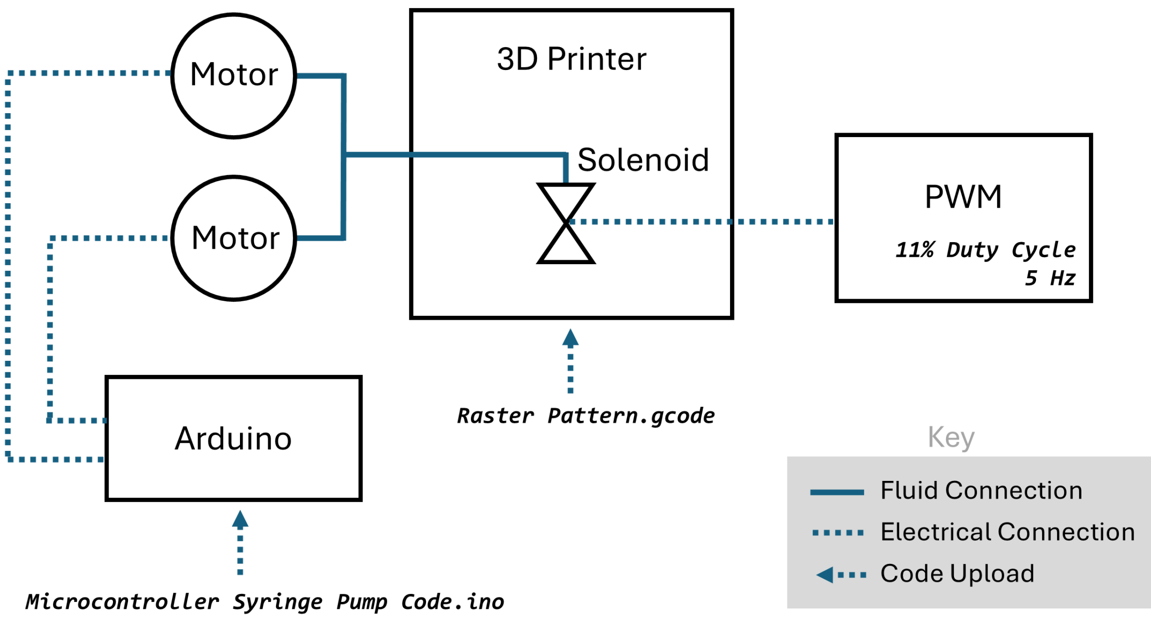


Fig. S-1 Block diagram of the primary components of Archerfish and their fluid and electrical connections as well as code upload steps.

Figure S-1 illustrates the system diagram for an Archerfish unit. The Arduino microcontroller is electrically connected to each stepper motor to send commands and drive motor motion. To send these commands, the `Microcontroller Syringe Pump Code.ino` Arduino script is uploaded through the Arduino IDE to the board. Each motor is connected to a syringe pump, which drives fluid into the solenoid. The solenoid is controlled using a pulse width modulator (PWM). The settings to drive this PWM are programmed directly to the on-board interface of the PWM itself, *e.g.*, 11% duty cycle and 5 Hz actuation frequency. To raster the XY motion of the solenoid and dispense droplets onto a substrate, the solenoid and fluid-combining junction are mounted to a 3D printer chassis. This design workflow is agnositic to the specific 3D printer used to raster the print head, an appropriate mount should be constructed to attach the solenoid to the print head. In our embodiment, we use a Monoprice Mini Select V2 and upload the `Raster Pattern.gcode` file *via* microSD card to the printer to drive its motion. All parts used to mount the solenoid to the print head of this 3D printer are included in the Archerfish GitHub repository. To extend utility to various 3D printers, an additional script is also included in the Archerfish GitHub repository, enabling users to modify the raster pattern G-code to fit the print bed of any 3D printer. All code to control these components can be found on the public GitHub repository: https://github.com/PV-Lab/Archerfish.

Component Designs and Parts

To ensure the reproducibility of the Archerfish system, we make all custom-designed parts available freely to the public in Solidworks and 3D-printable STL file formats, including all required control code. These designs and code can be found in the public GitHub repository: https://github.com/PV-Lab/Archerfish. All parts required to construct a minimum working Archerfish unit for a 3D printer are illustrated in Figure S-2. In our embodiment of the Archerfish system, there are two custom-designed components: the syringe pump and the droplet generator.

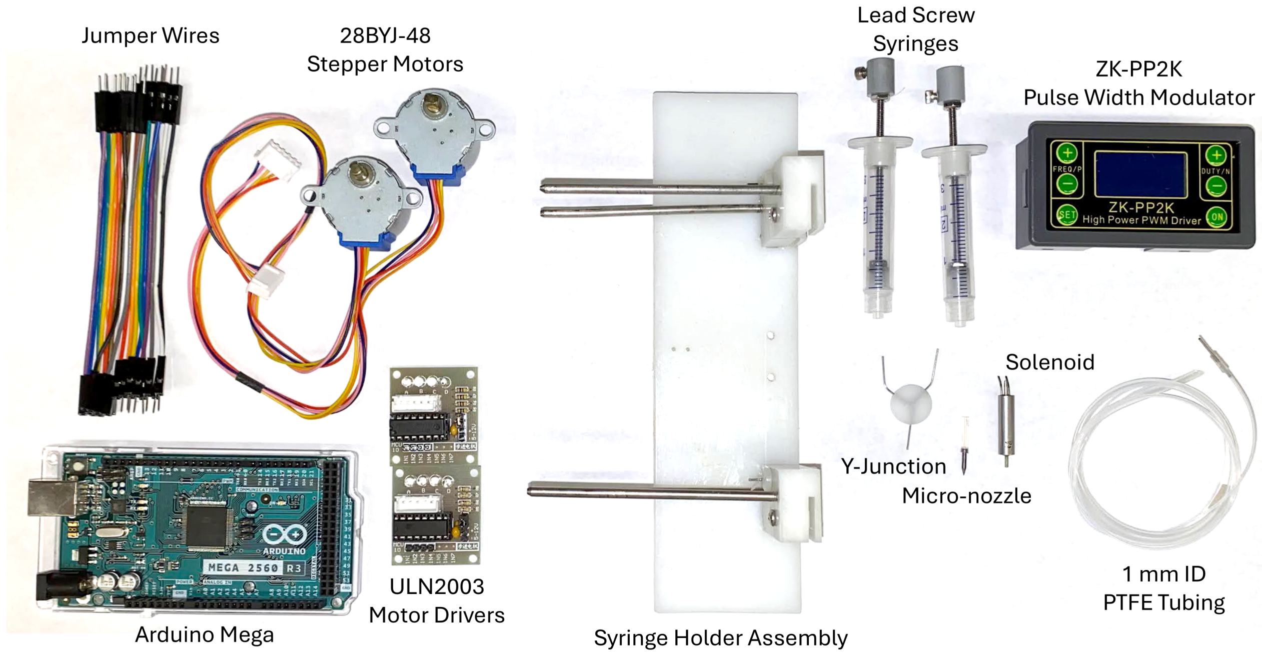


Fig. S-2 Parts used to construct an Archerfish unit.

Part Name	Vendor	Part Number	Cost of Materials Used (USD)
Power Button	Digikey	SW-PB7-1Z-A-LRRG-H-3-G2A	7.56
Motor Switch	Digikey	RF1-1A-DC-2-R-1	1.26
PWM	Amazon	ZK-PP2K 24V	10.57
3D Printer	Monopice	MP Select Mini V2	175.99
Stainless steel inlet	McMaster-Carr	5560K63	1.96
POM Y-Junction Material	McMaster-Carr	8497K311	6.26
Silicone Tubing	Amazon	0.5 mm ID x 2 mm OD	0.06
Solenoid Valve	The Lee Company	INKA2424212H VHS Series P-P	183.58
Micro-nozzle	Small Precision Tools	1551-120-437P 200 (10-11D-20)	50.00
PTFE Tubing	McMaster-Carr	7130N273	8.76
Barbed Luer Lock	McMaster-Carr	51525K291	1.24
3 mL Syringes	McMaster-Carr	7510A42	2.44
Syringe Drive Shaft	McMaster-Carr	95412A410	2.28
Coupling Material	McMaster-Carr	8497K171	0.08
Motor Rails	McMaster-Carr	8974K19	0.31
Stepper Motors and Drivers	Amazon	28BYJ-48 5V and ULN2003	5.90
Arduino Mega	Amazon		48.90
Total			**507.15**

Table S-1 Bill of materials to construct an Archerfish unit.

The syringe pump assembly (Figure 2) has four custom parts to enable its functionality: a coupling between the pump drive and the syringe, a coupling between the pump drive and the stepper motor, a syringe holder, and a syringe holder plate. To assemble this pump drive for the syringe pump, the 4-40 threaded 75 mm rod is inserted into one end of the drive-motor coupling, and the stepper shaft is inserted into the other end of the coupling. Set screws can then be threaded into the remaining holes of the coupling to fix them in place, or epoxy can be used as an alternative. Then, the drive-syringe coupling is threaded to the middle of the 4-40 threaded 75 mm rod. Then, the plunger of a 3 mL syringe should be removed, and the cap can be taken off and then tapped or epoxied to fix the remaining end of the 4-40 threaded 75 mm rod. The completed drive can now be inserted into the syringe, ensuring the drive-syringe coupling press fits into the opening of the syringe. The 90 mm long aluminum rods can be press-fit into the holes of the syringe holder, and the motor can be slid on while the syringe is slotted into the holder spacing. Finally, the syringe holder can be fixed to the syringe holder's back plate using screws to complete the assembly.

The droplet generator assembly (Figure 3) has only one custom part: the flow-combining Y-junction. This junction can be machined out of polyoxyethylene (POM) or 3D printed using resin stereolithography (SLA) printing and press fit with 304 stainless steel tubes to connect the plumbing lines. The design of the mount is not important to the functionality of Archerfish and should be determined based on the type of 3D printer used to construct an Archerfish unit. We have provided the designs for the droplet generator mount used for our embodiment of Archerfish using a Monoprice Mini V2 3D printer.

Component Unit Tests

To determine the minimum working capabilities of the primary components of Archerfish, we perform unit tests. The primary components of Archerfish include the Arduino microcontroller (Figure 1(1)), the syringe pumps (Figure 1(4)), and the droplet generator (Figure 1(5)).

Microcontroller Code

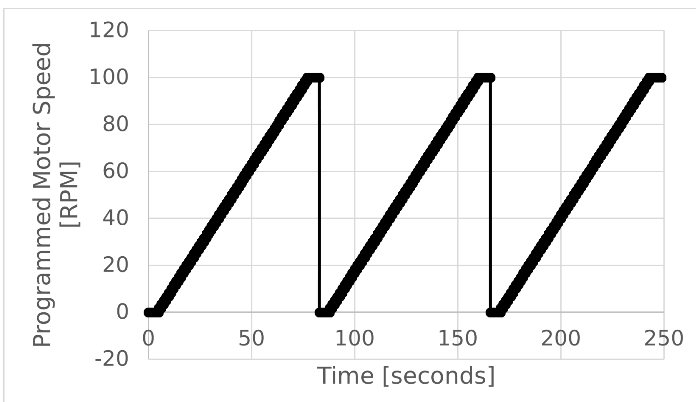


Fig. S-3 Programmed motor speed of a syringe pump using an Arduino microcontroller.

The Arduino Mega microcontroller supplies an electrical signal to drive each syringe pump motor at different speeds, generating a gradient of compositions using continuous printing. By updating each syringe pump's motor speed incrementally within a loop, we create a linearly increasing flow pattern for one motor and a linearly decreasing flow pattern for another motor. This motor speed update happens in the microcontroller code using the function `stepper.setSpeed`(x), where x is the desired motor speed of the syringe pump in rotations per minute (RPM). This code can be found in the public GitHub repository: https://github.com/PV-Lab/Archerfish.

Overlaying these flow patterns for each Arduino-controlled motor generates a gradient of fluids at an outlet. In Figure S-3, we unit-test the programmed RPM of one of the syringe pump motors by plotting the motor speed over time, as output by the Arduino, for three cycles. We see the motor starting from rest at 0 RPM, and then it increases linearly up to 100 RPM before resetting back to 0 RPM to start the next cycle. This unit test helps validate that the speeds we are programming our motor to run are correct.

Syringe Pump Flow

Controlling the motors on the syringe pumps directly determines the flow rate out of the syringe and into the mixing junction to be dispensed. We expect that as the programmed motor speed increases, the volumetric flow out of the syringe pump increases proportionally. In Figure S-4, we unit test this by connecting a flow meter in line with the syringe pump outlet to measure the volumetric flow. As the motor speed increases, the volumetric flow rate increases with the same periodicity. This unit test validates that the programmed motor speed from the Arduino is correctly coupled to the flow rate cycles of the fluid.

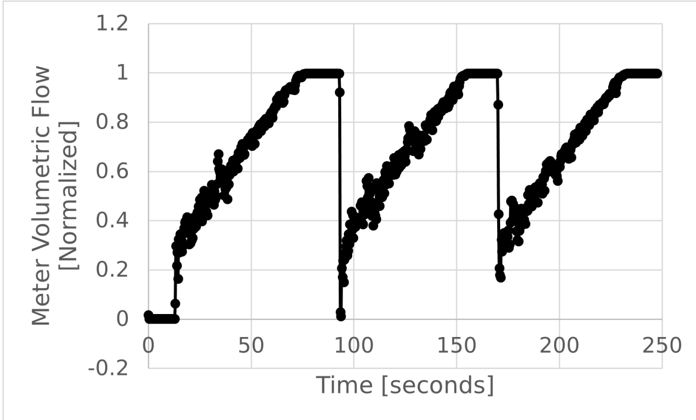


Fig. S-4 Measured volumetric flow rate pumped by the programmed motor.

Droplet Generation

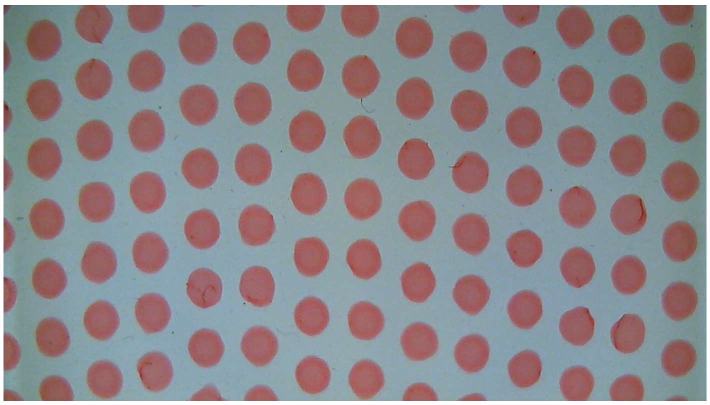
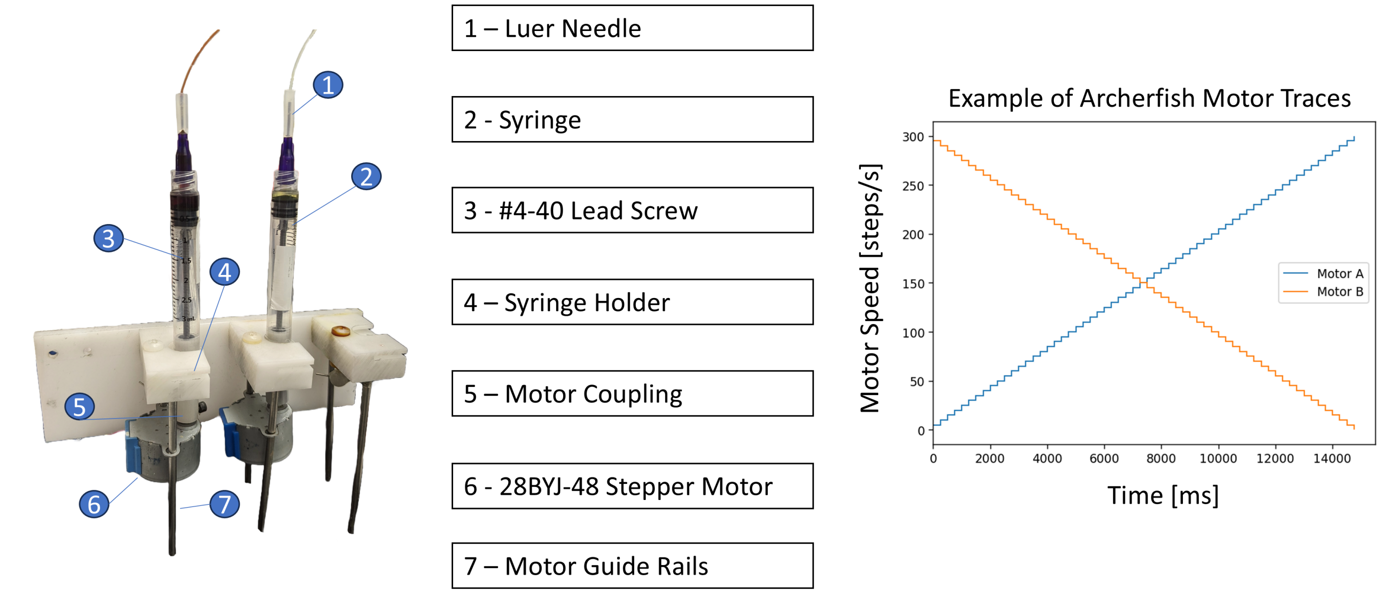


Fig. S-5 Red dyed water droplets generated using the droplet generator and PWM driver with settings of 5 Hz and 11% duty cycle.

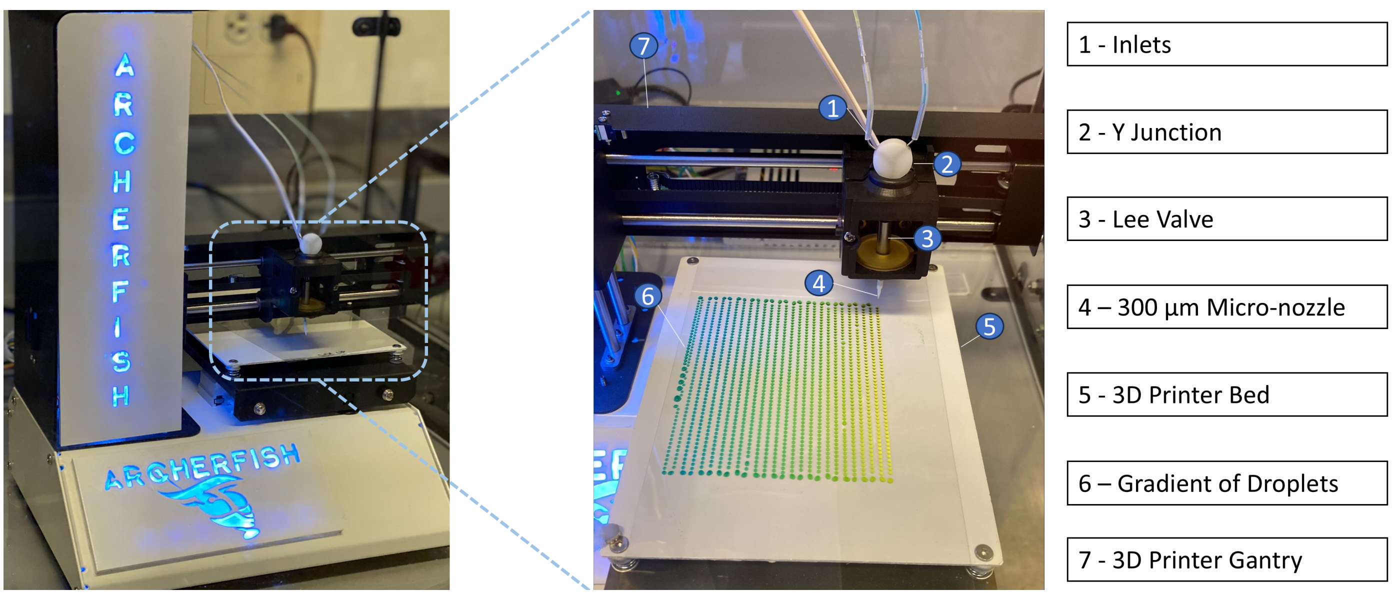
As the fluid flows to the outlet nozzle, it is jetted through the air and deposited onto a substrate. The PWM is the primary driver for droplet formation, and tuning its parameters is important to forming optimal droplets. In Figure S-5, we unit test the droplet-generating parameters of the PWM by visually inspecting images of the deposited droplet patterns. Here, we use water with red dye deposited onto a polyester sheet to easily visualize the output droplets. With a constant flow rate, we set the PWM to 5 Hz and 11% duty cycle to obtain this droplet pattern. This unit test instills confidence in the system's droplet-generating capabilities.

Validation Experiment

After succeeding the prior three unit tests, we can put all the pieces together to run a validation experiment and print a gradient of droplets using the validated microcontroller code, validated syringe pumps, and the validate droplet generator. In Figure S-6(a) we show the programmed motor speed traces for one gradient print cycle. Figure S-6(b) illustrates a gradient of droplets generated by Archerfish by mixing blue and yellow dye.



(a) Dual syringe pump motor traces



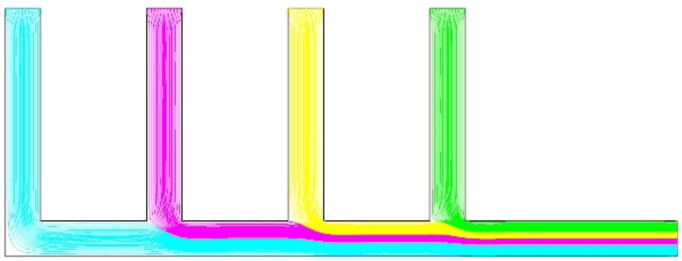
(b) Continuous printing validation

Fig. S-6 Continuous printing validation experiment using dual syringe pumps to generate a gradient of materials.

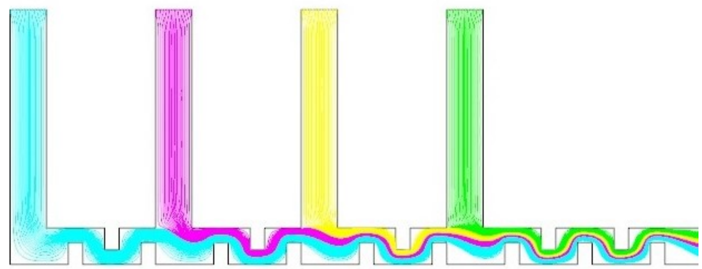
Comparative Analysis

Mixer Design Comparison

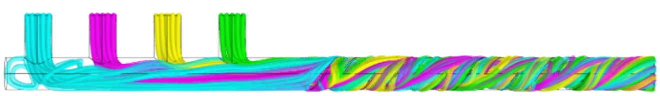
Figure S-7 compares the mixing potential of three different static mixer designs using COMSOL Multiphysics fluid simulation. The fluid flows are assumed non-viscous and = incompressible within 2 mm diameter pipes. Figure S-7(a) shows the simulation result for the design of four inlet flows into a uniform channel. Under the laminar flow regime, these four fluid flows only layer onto of each other, hence, mixing would only occur through diffusion over long pipe lengths with this design. To induce mixing in microfluidics systems using static or passive designs, Ward & Fan[62] implement ridged channels. In Figure S-7(b), we simulate this design, however, we note that after four ridges, complete mixing is not achieved. These results from Ward & Fan[62] likely do not translate from the micro-scale flow to milli-scale flow due to differences in pipe diameter and flow rates, although, increasing the number of baffled channels may improve mixing potential. Figure S-7(c) illustrates that by introducing baffled vanes into the channel, thorough mixing is induced over a short length of pipe. Figure S-7(d) extends the results of this baffled channel design to visualize the cross-sectional mixing potential of a two-inlet flow. Although strong mixing potential over short lengths is demonstrated with the implementation of a baffled channel, manufacturing such a channel at the millimeter scale is a challenge. Therefore, a simpler design solution of using active mixing *via* pulsation of a solenoid valve, as shown experimentally in Figure 3, achieves uniform mixing over short lengths at the millimeter scale.



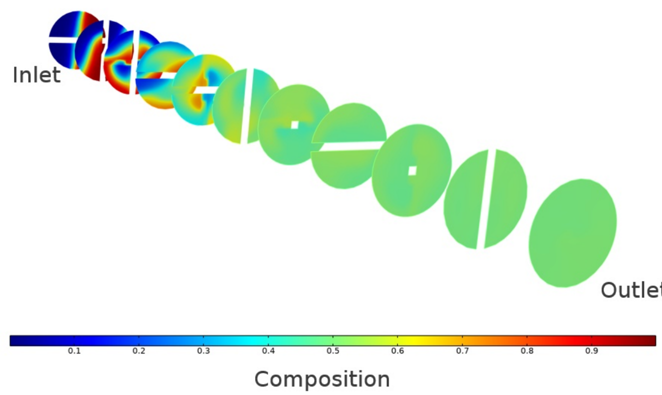
(a) Uniform Channel Laminar Flow



(b) Ridged Channel Laminar Flow



(c) Baffled Channel Laminar Flow

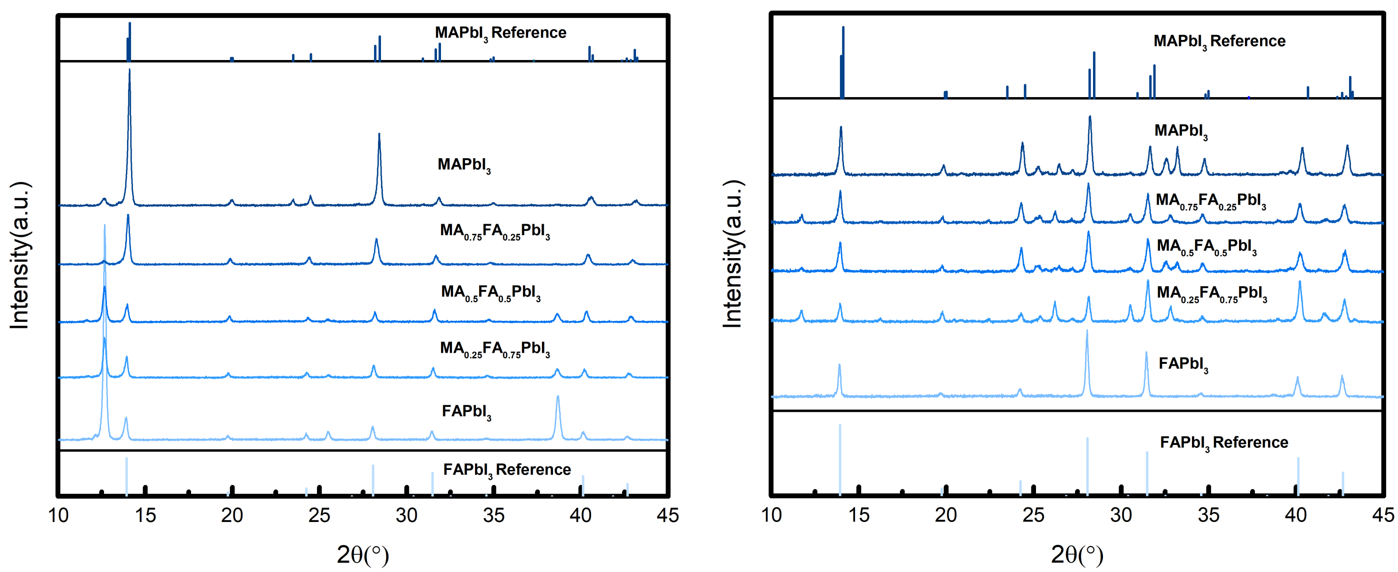


(d) Baffled Channel Cross-section View

Fig. S-7 Comparison of static mixer designs for combinatorial material printing. Laminar flow streamlines are shown for a (a) uniform channel mixer, (b) ridged channel mixer, and (c) baffled channel mixer. Each of these mixers has four unique inlet flows and one outlet of 2 mm diameter. Without active mixing, such as pulsation-induced mixing from a solenoid, implementing a baffled channel can induce mixing. (d) Cross-sectional view of mixing from two inlet flows using a baffled channel design is demonstrated. White regions of the cross-section represent the cross-section of the baffled vanes. The first flow is visualized as blue, assigned a value of 0.0, and the second flow is visualized as red, assigned a value of 1.0. A fully mixed flow is shown to occur after passing approximately three baffled vanes, visualized as green, assigned a value of 0.5.

X-ray Diffraction (XRD) Comparison

To benchmark the phase formation of functional materials, such as perovskite semiconductors, deposited using Archerfish, we compare the XRD traces of traditional spun-coat samples with Archerfish samples. Figure S-8(a) illustrates the XRD traces of five spun-coat formamidinium (FA) and methylammonium (MA) mixed cation perovskites, $MA_xFA_{1-x}PbI_3$. Figure S-8(b) illustrates the XRD traces of the same five $MA_xFA_{1-x}PbI_3$ compositions but deposited using Archerfish printing instead of traditional spin coating. It is shown that the major peaks of both the spun-coat and Archerfish perovskite samples align with the $MAPbI_3$ and $FAPbI_3$ reference peaks, however, two deviations should be noted. First, FA-rich spun-coat samples have a strong peak appearing at a 2θ angle of 12.5°, implying degradation of $FAPbI_3$ into PbI_2. This degradation was likely captured in the spun-coat samples because of their thinness compared to the Archerfish samples and the tendency of $FAPbI_3$ to rapidly degrade in humid ambient air. The second deviation is of the Archerfish samples containing more minor peaks in the XRD traces than the spun-coat samples and the references. Although most of the major peaks of the Archerfish samples align well with the spun-coat samples and the references, these extra XRD peaks imply lower phase purity in the crystal structures. However, it is noted that the Archerfsh samples do not exhibit any detected degradation to PbI_2 at the 12.5° 2θ peak, unlike the spun-coat samples, hence, implying higher ambient air stability.



(a) Spun-coat Samples XRD

(b) Archerfish Samples XRD

Fig. S-8 XRD traces of five formamidinium (FA) and methylammonium (MA) mixed cation perovskites, $MA_xFA_{1-x}PbI_3$. (a) XRD traces of five uniquely spun-coat $MA_xFA_{1-x}PbI_3$ compositions. (b) XRD traces of five Archerfish droplets printed in a gradient of $MA_xFA_{1-x}PbI_3$ compositions. The reference peaks and magnitudes of pure $MAPbI_3$ and $FAPbI_3$ phases are illustrated at the tops and bottoms of the figure panels.

References

1 Kevin Ward and Z. Hugh Fan. Mixing in microfluidic devices and enhancement methods. *Journal of Micromechanics and Microengineering*, 25:1–15, 2015.