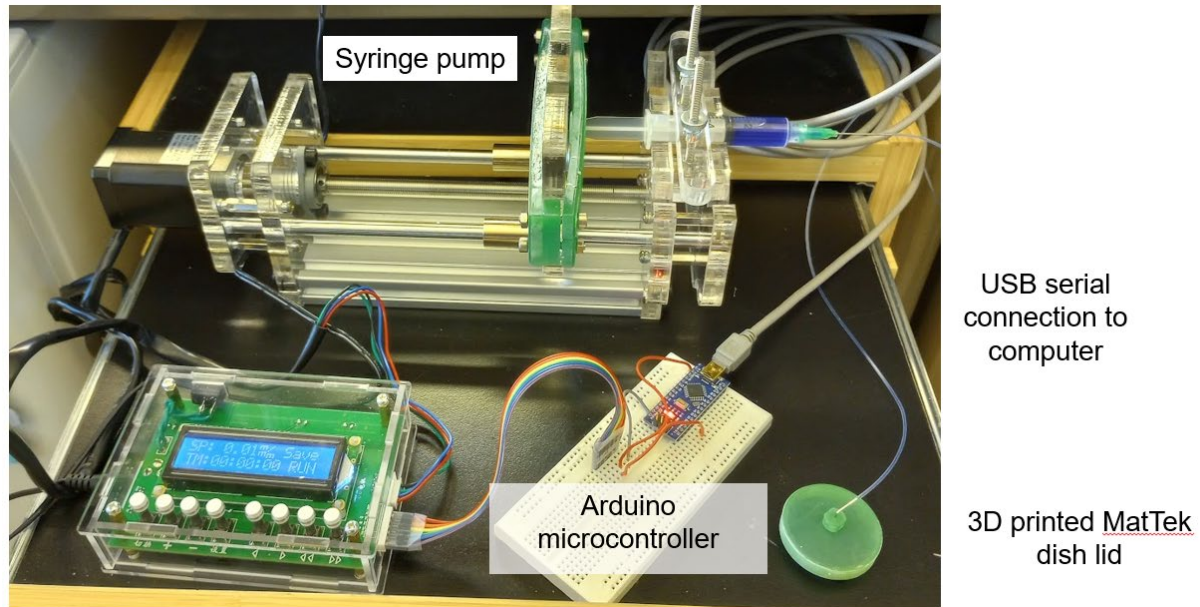


Auto-injection system

Thomas Graham, UC Berkeley, Updated 5-27-24



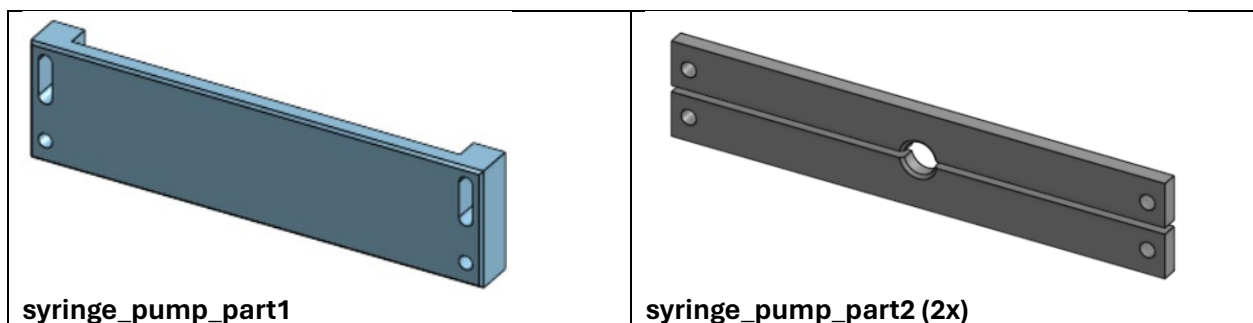
The solution is delivered to the sample through PE60 tubing (e.g., Intramedic 427416) connected to the syringe via a 21-gauge needle (McMaster-Carr 75165A681) and to the MatTek dish via a bent 21-gauge needle (McMaster-Carr 75165A85) inserted in a custom lid.

The syringe plunger is held in place using custom 3D-printed brackets.

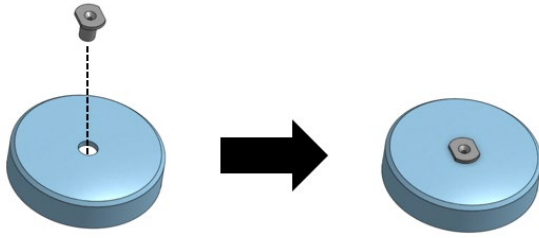
Operation of the syringe pump is controlled using Python code via a serial connection to an Arduino microcontroller.

Syringe pump

Print one copy of **syringe_pump_part1** and two copies of **syringe_pump_part2**. Connect these three parts using four M4 screws to clamp the syringe plunger to the pump.



35 mm dish lid



Print and assemble **MatTek_perfusion_lid** and **MatTek_perfusion_lid_insert** as shown. Cut or break off a 90° angle 21-gauge needle, and place in the insert. Glue the components together. UV-curable 3D printing resin can be used for this purpose.

Arduino code

Load **triggered_backandforth.ino** into an Arduino microcontroller. When the Arduino receives the serial input “1”, it will cause the syringe pump to inject the contents of the syringe and move back and forth to mix five times. We connected it to the fast-motion pins of the syringe pump to move the plunger of the syringe back and forth as fast as possible.

The time delay needs to be calibrated to dispense the appropriate volume.

```
int val = 0;

void setup() {
  pinMode(13, OUTPUT); // connect to fast backward movement input of syringe pump
  pinMode(2, OUTPUT);  // connect to fast forward movement input of syringe pump
  digitalWrite(13, HIGH); // HIGH setting stops movement
  digitalWrite(2, HIGH);
  Serial.begin(9600);
  Serial.println('a');
}

void loop() {
  if (Serial.available() > 0){
    val = Serial.read();
    if (val == '1') // inject and mix
    {
      for (int i = 0; i < 5; i++) {
        digitalWrite(13, HIGH);
        delay(100);
        digitalWrite(2, LOW); // LOW setting turns movement on
        delay(7000); // change this depending on the volume you want to dispense
        digitalWrite(2, HIGH);
        delay(100);
        digitalWrite(13, LOW);
        delay(7000); // change this depending on the volume you want to dispense
      }
      digitalWrite(13, HIGH); // important to turn the movement off at the end!
    }
  }
}
```

Python code

Running the short script **syringe_inject.py** opens a serial connection to the Arduino and sends the character “1”, triggering the autoinjection. The name of the serial port needs to be set to whatever corresponds to the Arduino on your computer. We found that the `time.sleep(2)` step was necessary to allow time for the serial connection to be established.

```
import serial
import time

ser = serial.Serial('COM7', baudrate=9600, timeout=1)
time.sleep(2)

ser.write(b'1')

ser.close()
```