Electronic Supplementary Material (ESI) for RSC Advances.

Supporting Information

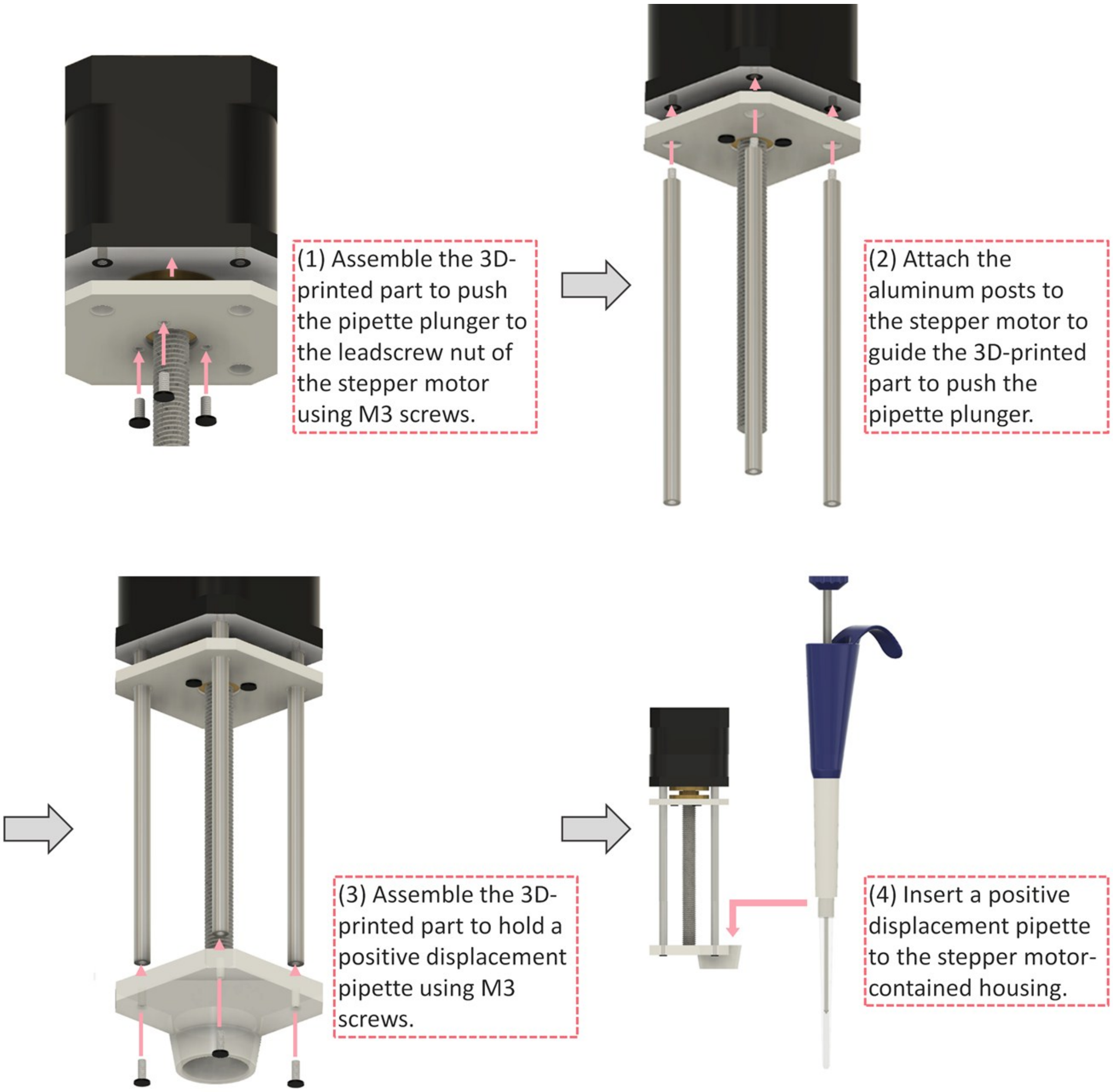
An open-source programmable smart pipette for portable cell separation and counting

Eunjung Lee,* Byeongyeon Kim,* and Sungyoung Choi**

Department of Biomedical Engineering, Hanyang University, Seoul 04763, Republic of Korea

Contents:

1. Supplementary Figures



(1) Assemble the 3D-printed part to push the pipette plunger to the leadscrew nut of the stepper motor using M3 screws.

(2) Attach the aluminum posts to the stepper motor to guide the 3D-printed part to push the pipette plunger.

(3) Assemble the 3D-printed part to hold a positive displacement pipette using M3 screws.

(4) Insert a positive displacement pipette to the stepper motor-contained housing.

Figure S1. Assembly process of the smart pipette.

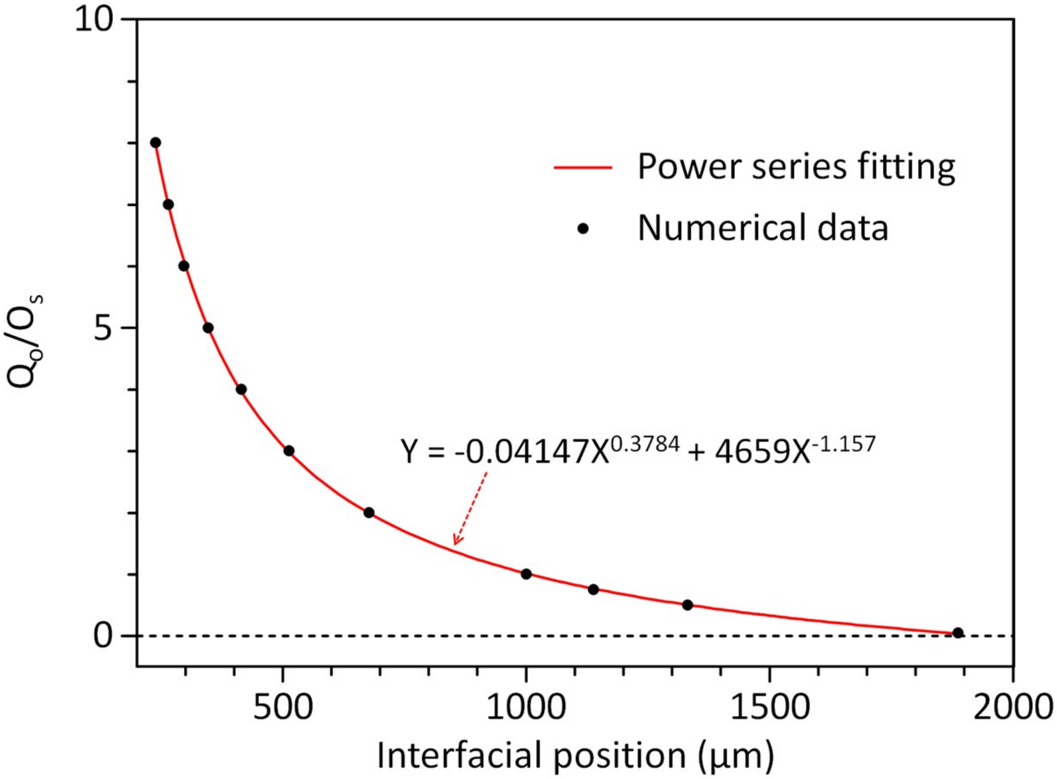


Figure S2. Nonlinear relationship between the flow rate ratio of Q_o to Q_s and the interfacial position. The Q_o value was calculated by measuring the interfacial position in the microfluidic comparator and obtaining the ratio, Q_o/Q_s from the power series fitting.

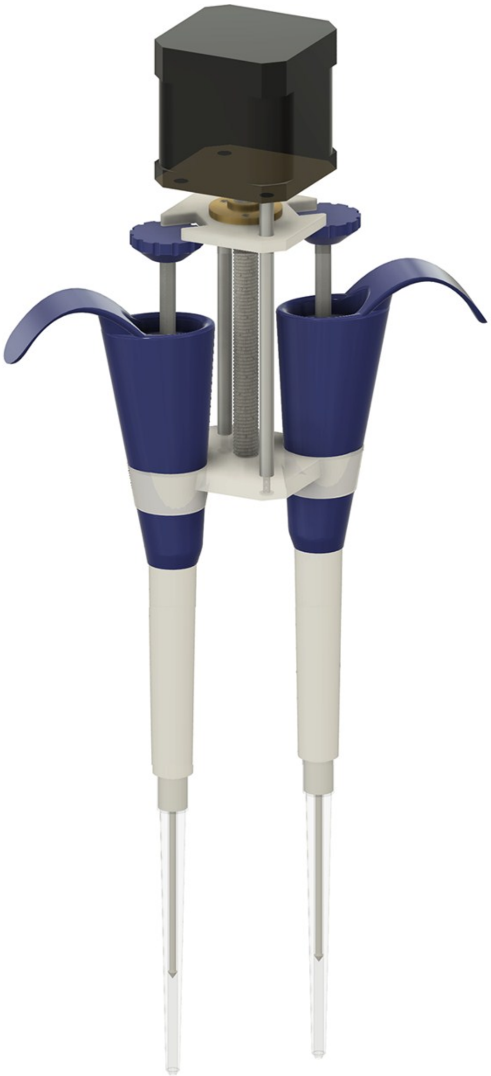


Figure S3. Potential design of the smart pipette which can hold and control two positive-displacement pipettes.

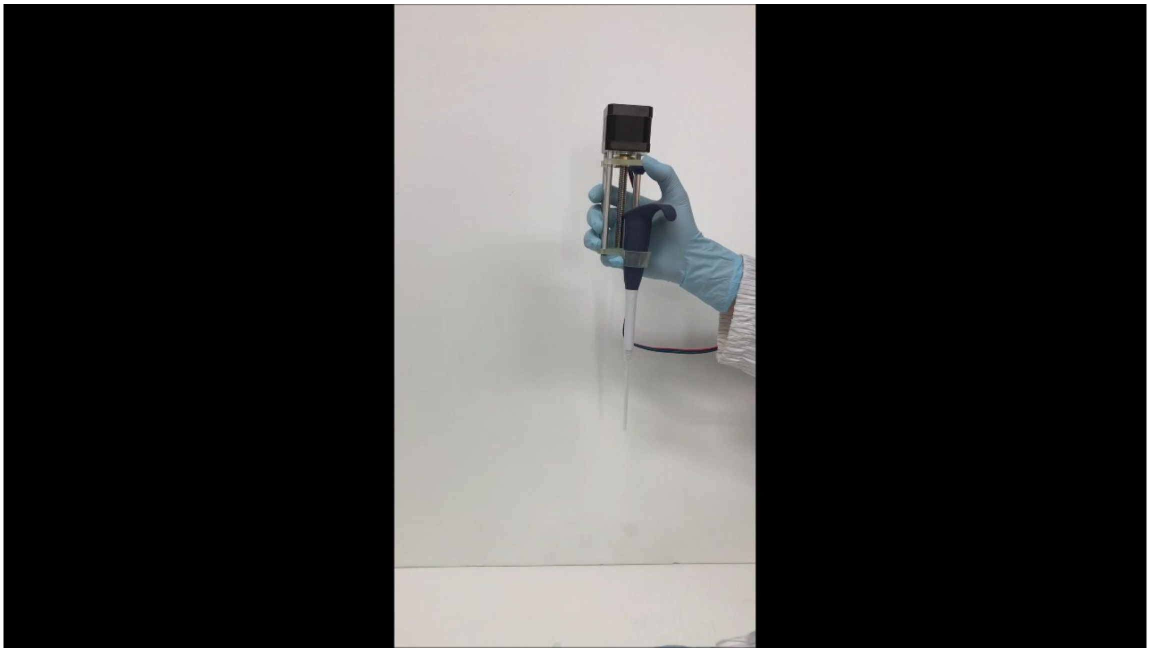
3. Supplementary Table

Table S1. List of parts for the smart pipette assembly.

Part name	Product number	Manufacturer	Cost ($)
Arduino	Arduino Uno R3	SparkFun Corp, United States	22
Stepper motor driver	A4988	Allegro MicroSystems, LLC, United States	0.8
Stepper motor (containing a lead screw and a lead screw nut)	SL42STH40-1684A-300	Changzhou Fulling Motor Co., Ltd., China	21
Positive displacement pipette tip	Prstrl 180/3 C-250	METTLER TOLEDO Corp, United States	0.4
Positive displacement pipette	Positive-Displacement Pipette MR-250	METTLER TOLEDO Corp, United States	410
Aluminum posts	Custom-made products	Doo Kyoung Corp., Korea	30 (each post)
3D-printed holder			4
3D-printed push block			3

3. Supplementary Videos

Movie S1. Flow-rate response of the OS pipette to pulse commands for a set flow rate of 97.4 μL/min. The set pulse widths were 0.5, 0.9, 2, 4, 8 and 12 seconds in order. The dotted red line denotes the ROI for interfacial position measurement.



Movie S2. Handheld operation of the microfluidic plasma separator using the OS pipette.

4. Supplementary Arduino Codes

4.1. Constant flow-rate generation

```
const int enable=11;
const int stepper=9;
const int direc=10;

void setup() {
pinMode(enable,OUTPUT);
pinMode(stepper, OUTPUT);
pinMode(direc, OUTPUT);

digitalWrite(enable, LOW);// Disable enable pin
Serial.begin(9600);// put your setup code here, to run once:
}

void loop() {
int index;

delay(100);
   digitalWrite(direc,LOW);//Set up direction


   for(index=0;index<14800;index++)
  {
    digitalWrite(stepper, HIGH);
    delay(25);
    digitalWrite(stepper, LOW);
    delay(25);//Set up stepper motor velocity 19.5ul/min
  }

  delay(10000);
```

```
 }
```

4.2. Pulsed flow-rate generation

```
const int enable=11;
const int stepper=9;
const int direc=10;

void setup() {
pinMode(enable,OUTPUT);
pinMode(stepper, OUTPUT);
pinMode(direc, OUTPUT);

digitalWrite(enable, LOW);// Disable enable pin
Serial.begin(9600);// put your setup code here, to run once:
}

void loop() {
int index;

delay(100);
  digitalWrite(direc,LOW);//Set up direction

 for(index=0;index<300;index++)
 {
   digitalWrite(stepper, HIGH);
   delay(25);
   digitalWrite(stepper, LOW);
   delay(25);
 }//Set up stepper motor velocity 19.5ul/min
 for(index=0;index<53;index++)
 {
  digitalWrite(stepper, HIGH);
  delayMicroseconds(5000);
  digitalWrite(stepper, LOW);
  delayMicroseconds(5000);
 }//Set up stepper motor velocity 100ul/min for 0.5s
 for(index=0;index<212;index++)
 {
  digitalWrite(stepper, HIGH);
  delay(25);
  digitalWrite(stepper, LOW);
  delay(25);
 }//Set up stepper motor velocity 19.5ul/min for 10s
for(index=0;index<95;index++)
 {
  digitalWrite(stepper, HIGH);
  delayMicroseconds(5000);
  digitalWrite(stepper, LOW);
  delayMicroseconds(5000);
 }  //Set up stepper motor velocity 100ul/min for 0.9s
 for(index=0;index<212;index++)
 {
  digitalWrite(stepper, HIGH);
  delay(25);
  digitalWrite(stepper, LOW);
  delay(25);
 }
for(index=0;index<211;index++)
 {
  digitalWrite(stepper, HIGH);
  delayMicroseconds(5000);
  digitalWrite(stepper, LOW);
  delayMicroseconds(5000);
 }  //Set up stepper motor velocity 100ul/min for 2s
```

```
  for(index=0;index<212;index++)
  {
    digitalWrite(stepper, HIGH);
    delay(25);
    digitalWrite(stepper, LOW);
    delay(25);
  }
for(index=0;index<422;index++)
  {
    digitalWrite(stepper, HIGH);
    delayMicroseconds(5000);
    digitalWrite(stepper, LOW);
    delayMicroseconds(5000);
  }  //Set up stepper motor velocity 100ul/min for 4s
for(index=0;index<212;index++)
  {
    digitalWrite(stepper, HIGH);
    delay(25);
    digitalWrite(stepper, LOW);
    delay(25);
  }
    for(index=0;index<844;index++)
  {
    digitalWrite(stepper, HIGH);
    delayMicroseconds(5000);
    digitalWrite(stepper, LOW);
    delayMicroseconds(5000);
  } //Set up stepper motor velocity 100ul/min for 8s
  for(index=0;index<212;index++)
  {
    digitalWrite(stepper, HIGH);
    delay(25);
    digitalWrite(stepper, LOW);
    delay(25);
  }
  for(index=0;index<1266;index++)
  {
    digitalWrite(stepper, HIGH);
    delayMicroseconds(5000);
    digitalWrite(stepper, LOW);
    delayMicroseconds(5000);
  }  //Set up stepper motor velocity 100ul/min for 12s
    for(index=0;index<500;index++)
  {
    digitalWrite(stepper, HIGH);
    delay(25);
    digitalWrite(stepper, LOW);
    delay(25);
  }
  delay(10000);
  }
```

4.3. Stepped flow-rate generation

```
const int enable=11;
const int stepper=9;
const int direc=10;

void setup() {
pinMode(enable,OUTPUT);
pinMode(stepper, OUTPUT);
pinMode(direc, OUTPUT);

digitalWrite(enable, LOW);
Serial.begin(9600);// put your setup code here, to run once:
}
```

```
void loop() {
int index;

delay(100);
   digitalWrite(direc,LOW);

  for(index=0;index<250;index++)
  {
    digitalWrite(stepper, HIGH);
    delay(25);
    digitalWrite(stepper, LOW);
    delay(25);
  }
   for(index=0;index<338;index++)
  {
    digitalWrite(stepper, HIGH);
    delayMicroseconds(12500);
    digitalWrite(stepper, LOW);
    delayMicroseconds(12500);
  }//Set up stepper motor velocity 39.0ul/min for 8s
  for(index=0;index<506;index++)
  {
    digitalWrite(stepper, HIGH);
    delayMicroseconds(8333);
    digitalWrite(stepper, LOW);
    delayMicroseconds(8333);
  }//Set up stepper motor velocity 57.8ul/min for 8s
  for(index=0;index<666;index++)
  {
    digitalWrite(stepper, HIGH);
    delayMicroseconds(6250);
    digitalWrite(stepper, LOW);
    delayMicroseconds(6250);
  }//Set up stepper motor velocity 78.0ul/min for 8s
  for(index=0;index<844;index++)
  {
    digitalWrite(stepper, HIGH);
    delayMicroseconds(5000);
    digitalWrite(stepper, LOW);
    delayMicroseconds(5000);
  }//Set up stepper motor velocity 97.4ul/min for 8s
 for(index=0;index<666;index++)
  {
    digitalWrite(stepper, HIGH);
    delayMicroseconds(6250);
    digitalWrite(stepper, LOW);
    delayMicroseconds(6250);
  }//Set up stepper motor velocity 78.0ul/min for 8s
   for(index=0;index<506;index++)
  {
    digitalWrite(stepper, HIGH);
    delayMicroseconds(8333);
    digitalWrite(stepper, LOW);
    delayMicroseconds(8333);
  }//Set up stepper motor velocity 57.8ul/min for 8s
  for(index=0;index<338;index++)
  {
    digitalWrite(stepper, HIGH);
    delayMicroseconds(12500);
    digitalWrite(stepper, LOW);
    delayMicroseconds(12500);
  }//Set up stepper motor velocity 39.0ul/min for 8s
  for(index=0;index<500;index++)
  {
    digitalWrite(stepper, HIGH);
    delay(25);
    digitalWrite(stepper, LOW);
    delay(25);
  }

  delay(10000);
  }
```

5. Supplementary Matlab Codes

5.1. MATLAB code for interfacial position measurement

```
vid = VideoReader('50-3.avi');%Read video
numFrames = vid.NumberOfFrames;
A=zeros(1800,1);
 for w=1:1800

     image=read(vid,w);%Read frame
     n=0;
     for k=1:1:300

         B=impixel(J, 300, k);
         C=B(1,1); %Get image intesity information
         if C>60
             n=n+1;
         end % Measure interfacial position
         A(w,1)=n;
     end
 end
xlswrite('50-3',A);
```

5.2. MATLAB code for analyzing fluorescene bead images

```
  vid = VideoReader('10-3.avi');%Read video
  numFrames = vid.NumberOfFrames;
   n=0;
   for w=1:numFrames
       image=read(vid,w); % Read frame
       I=rgb2gray(image);
       for m=1:1:112
```

```
        B=impixel(I, m+199, 400);
        C=B(1,1);
        A(w,m)=C;% Get image intensity information

    end
 end
xlswrite('10-3',A);
```