Trevor Shannon
6.163 Strobe Lab Final Report

A Method and Apparatus for Easily Creating Liquid Splash Collisions

Project Summary

Previous attempts to collide droplets of liquid with splashes from previous droplets has been successful, but the methods employed have often been sensitive to minor perturbations in the apparatus, difficult to set up, or limited in their ability to be customized. We present an apparatus and method for reliably and consistently producing collisions between a falling droplet and the splash made from a previous droplet. Additionally, we provide suggestions and recommendations for photographing liquid collisions with both a high speed video camera and with strobe photography techniques. The information, and schematics in this report are free for personal use. The goal of this project was not only to develop an easy method for producing droplet collisions, but also to share the knowledge gained in this exercise with others pursuing a study of liquid collisions.

Previous Work

Martin Wahl has long been regarded as the expert on droplet collisions. His website, liquidsculpture.com displays a number of his stunning strobe photographs. Martin demonstrated his technique and apparatus on season 1 of Discovery Channel's Time Warp in episode 8. Martin's amazing ability to produce beautiful splash photos is surpassed only by his ability to set up his drop collision apparatus. The apparatus he uses, while sound and functional, is tuned entirely by hand, using years of experience and gained intuition. The goal of this project was to create results similar to Martin's, while creating a collision apparatus that an inexperienced user could operate and quickly master.

Mumford Microsystems provides an easier solution for creating splash collisions with the Time Machine controller. The Time Machine can be used in conjunction with an aftermarket water droplet system available from Mumford to create individual droplets at precise time intervals. This is done via the Drop mode of the Time Machine, and makes droplet collisions much more accessible to the general public (more info). Many individuals make use of this controller to create splash collision photos, and the results obtained are often quite beautiful. However, because the Time Machine has a plethora of different functions, the user interaction and interface is often non-intuitive for a particular mode of operation. A series of menus must be navigated before the user is able to change a simple parameter such as the time between two droplets. Additionally, the Time Machine only allows user control over a small set of parameters. Finally, the droplet system used with the Time Machine is is driven from the pressure created by an elevated tank of liquid. Thus, as the liquid level decreases over time, the pressure behind the droplets changes, leading to a slow shift in droplet properties and decreasing repeatability.

Description of the Droplet Collider Apparatus

Apparatus Requirements

We feel an apparatus for creating droplet collisions needs to meet the following general requirements. The droplet collider should:

- Have the ability to produce a single droplet on command

- Produce consistent droplets (e.g. initial velocity, droplet diameter)
- Operate independent of source container liquid volume
- Be resistant leaks, air accumulation, and other transient behaviors
- Be easily customizable (e.g. nozzles, liquids)
- Provide an easy-to-use interface
- Have the ability to self-calibrate to enable inexperienced users to produce collisions

Apparatus Overview

With the above requirements in mind, we developed a system for producing splash collisions. The apparatus consists of a solenoid pump to produce droplets, a power source for the aforementioned pump, a droplet controller for timing the release of droplets, an air release valve to bleed any air that gets introduced into the system, an infrared breakbeam sensor for calibration and strobe photos, and two independent basins of liquid: a source basin and a splash basin. Also, if strobe photos are to be taken, a strobe, camera, and strobe delay controller are required. Note that the strobe and camera are not considered to be part of the droplet collider apparatus, as these are easily interchangeable with no effect on the physical collisions. Figure 1 shows a photograph of the apparatus without the droplet controller, strobe controller, or power source. Additional images of the apparatus can be found in Appendix A.

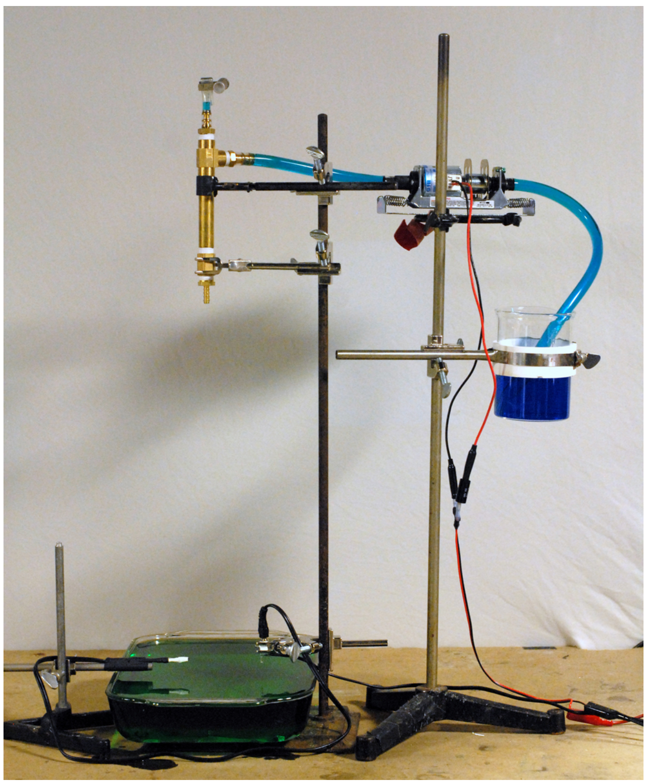


Figure 1. Apparatus developed for reliably producing splash collisions. The brass nozzle is on the left with an air bleed valve at the top of the nozzle. A solenoid pump for producing droplets is suspended on the right ring stand. An infrared breabeam sensor is positioned above the splash basin. Note the droplet controller, strobe controller, and power supply are not pictured.

Figure 2 shows a schematic representation of the apparatus pictured in Figure 1, with each component labeled. The positioning of elements in the schematic of Figure 2 closely matches the positioning of the elements pictured in Figure 1.

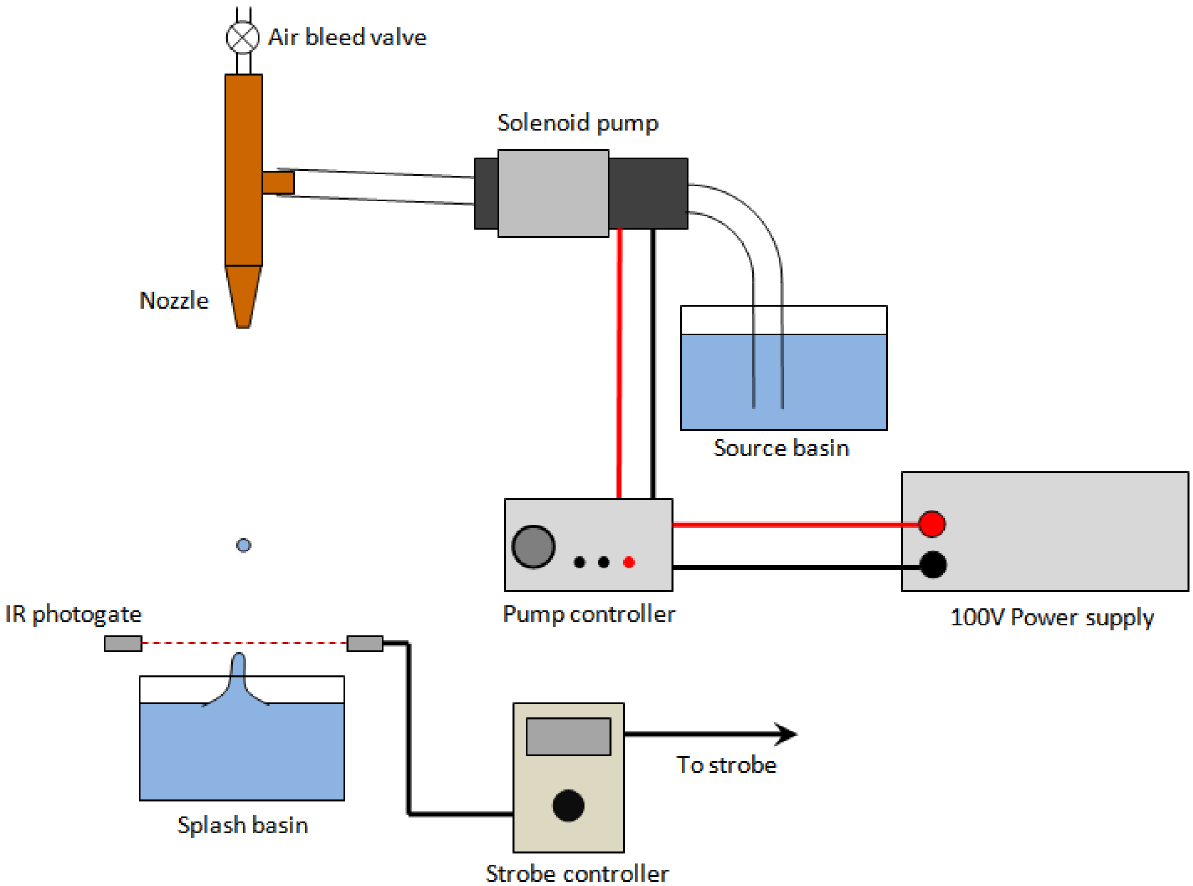


Figure 2. Schematic of the apparatus developed for creating splash collisions. Refer to Figure 1 for a photograph and detailed description of the apparatus. Note this figure shows a setup including a strobe controller, which is only necessary when taking strobe photographs. The strobe controller is simply an adjustable delay unit

While the exact components used likely have little effect on the apparatus on the whole, a detailed description of this particular apparatus follows, highlighting the important aspects of the apparatus critical to success.

Apparatus Detail - Nozzle

The nozzle plays an important role in drop formation. A nozzle with a very small hole will most likely produce a stream of liquid rather than a droplet. As the nozzle diameter decreases, the mass flow through the nozzle needs to also decrease to preserve clean droplet formation. This translates to either very small pump motions or very rapid pump motions, both of which are difficult to obtain. A nozzle with a very large hole may produce an oblong, oscillating droplet with many smaller droplets following. Additionally, if the nozzle is large enough, the liquid will not stay suspended inside the nozzle and simply leak out when the nozzle is in the vertical orientation.

The nozzle we used is a brass fitting with an 0.180" inner diameter cylindrical bore. The nozzle fits into a short segment of brass piping with a T-fitting to create an inlet. At the end opposite the nozzle, there is a short segment of flexible tubing. This tubing can be folded over and clamped to act as a valve that can be opened to release any air that makes its way into the system. Figure 3 shows the nozzle and bleed valve assembly in detail.

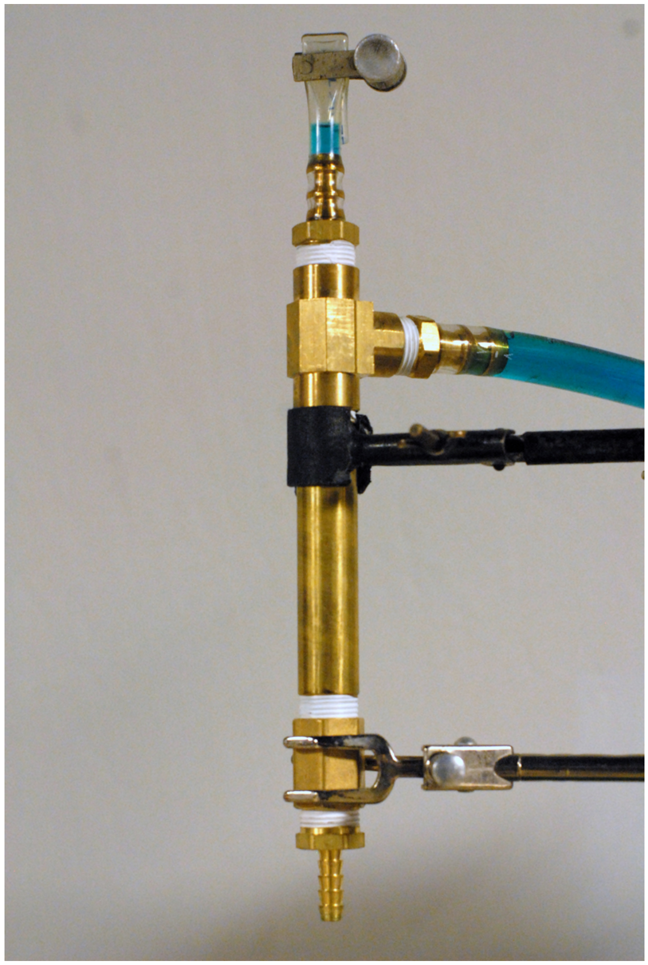


Figure 3. Droplet collider nozzle assembly. The nozzle is a brass fitting with and 0.180 inch cylindrical bore connected to the liquid inlet via a short length of brass piping and a T-junction. Vertically above the nozzle is a valve consisting of a short piece of flexible tubing and tube clamp that can be opened to release air or change liquids.

Note that when the bleed valve is opened to release air, the nozzle must be stopped to keep the liquid from flowing out. When in the vertical configuration, this valve is aligned vertically above the nozzle because any air bubbles will rise vertically and therefore become concentrated at the air release valve. The removal of air is critical to dropper performance, as air in the system reduces the response of the dropper to a pump's input. Additionally, air can limit the flow of fluid through the tubing, and these effects vary with the volume of air in the system. Thus, the only way to reliably create a droplet is to ensure there is no air present in the collider apparatus. Ideally, once the system is filled with liquid, air would not enter. However, we found that there are always small bubbles stuck to the sides of the tubes or in the pump, so the air bleed valve provides a convenient outlet for these bubbles once they are gathered at the proper location (whether by tapping the tubes or due to the buoyancy force over time). Finally, the air bleed valve simplifies the removal of one liquid and the subsequent refilling of the system with a different liquid. By opening the air valve with the nozzle unblocked, the nozzle is quickly purged of liquid, and a new liquid can be pumped in to replace it.

The nozzle is easily removable and can be replaced with nozzles of different sizes or shapes to produce the desired effect. Also, being made of brass, it is resistant to corrosion and very rigid. The rigidity of the nozzle assembly makes it quite easy to hold and position properly.

Apparatus Detail - Solenoid Pump

A solenoid pump consists of an impeller coupled to a simple solenoid. When the current is passed through the solenoid, it actuates the impeller, pushing water through the pump outlet. When solenoid is switched off, a restoring spring brings the piston back to the resting position. A check valve is used to ensure flow in only one direction. Additionally, a diode is often included as part of the solenoid pump to rectify a source voltage that changes sign. When pumping large volumes of water, a solenoid pump will commonly be used as an oscillating pump, such as in Doc Edgerton's Piddler setups. Solenoid pumps are useful for creating droplets because they can be quickly turned on and off to provide an impulsive force to the water, which creates a very small, fast volume flow. This forces a small amount of liquid out of the nozzle, which is then released as a droplet when the pump returns to the resting position. In fact, even when used in a continuous, oscillating mode, the outlet of the pump is still producing discrete droplets of water--one for each period of oscillation--which is the basis for Edgerton's Piddlers. Solenoid pumps are also useful because they can be controlled beyond on/off. By controlling the amount of current flowing through the solenoid, the pumping action can be modulated. For example, a triangle wave being sent to the solenoid pump would produce a slightly different motion when compared to a square wave. The particular solenoid pump used in this apparatus is a model 14825-601 oscillating pump from Gorman-Rupp Industries. The tubing used in conjunction with this pump is 7/16" inner diameter Tygon tubing.

The pump is placed close to the nozzle to minimize tubing between the two components. This helps decrease the response time of the system, as there is less fluid inertance and fluid resistance with a shorter tube. Also, placing the pump only a few centimeters below the nozzle inlet minimizes the mechanical work the pump needs to do to move water and create droplets. Note that the source basin connected to the inlet of the pump should also be at nearly the same height as the pump for the same reason.

The apparatus used in this investigation uses an impulsive (i.e. very short) square wave to produce individual droplets. Figure 4 shows a representation of the voltage across the pump as a function of time. The maximum voltage depends on the power supply being used, but as the Gorman-Rupp pump was designed for 115V AC, we supplied it with nominally 100V (closer to 90V) with an HP 62058 Dual DC Power Supply for the best results. This particular supply has two 50V outputs, and we put these in series to get a single 100V output.

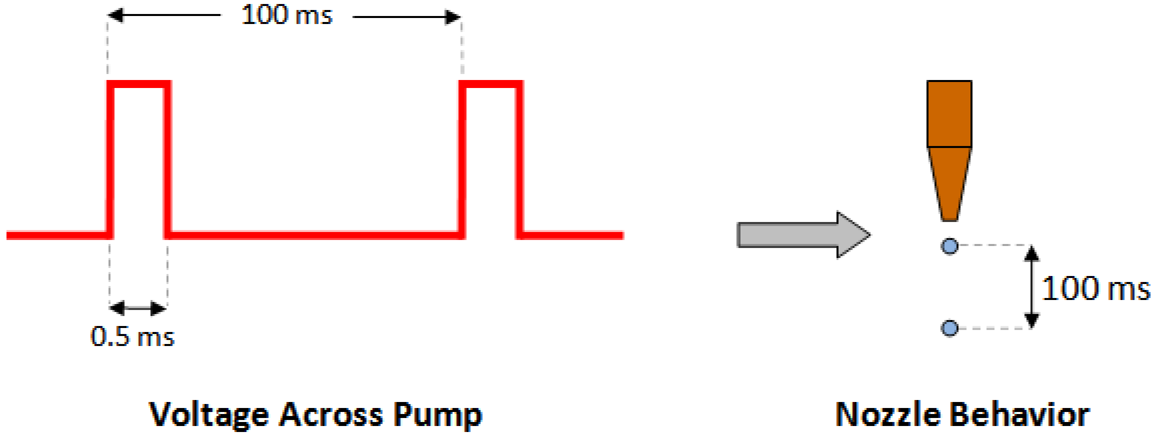


Figure 4. Diagram of the signal used to produce multiple discrete droplets with a solenoid pump. The red line shows the voltage across the pump as a function of time. The maximum voltage was nominally 100V, and the minimum was zero. Note the time intervals listed are subject to change or adjustment depending on the droplet parameters required.

The short time for which the pump is turned on (when the voltage across the pump is high) creates one droplet (droplet formation time). The time for which the pump is on roughly determines the droplet size. This is due to the fact that the pump and fluid system has a finite response time to a step input. Thus, if the step is short enough, the solenoid pump will not travel it's maximum throw, as the current flow ceases before the pump can reach maximum position. Similarly, if the droplet formation time interval is long, the pump will complete a full pumping motion and come to a stop in that position until the current flow once again stops. Thus, longer droplet formation times correspond to larger droplets. We found that long droplet formation times lead to a high number of "follower droplets" that pull away from the nozzle as the main, large droplet is released. A shorter time can lead to a reduction of these follower droplets, though there is a lower limit on the droplet formation time. At this limit, liquid will be forced out of the pump, but then get pulled back in without detaching from the nozzle on the pump's return to the resting position. By repeating this droplet formation signal periodically, a stream of discrete droplets can be formed. This is convenient for droplet collisions, as two droplets need to be formed: one to make a splash, and one to collide with that splash. As discussed in the Overview of Droplet Collision Procedure section, the droplet formation times used were generally in the range of 5000 microseconds to 5200 microseconds, and the droplet delay times were on the order of 100 milliseconds.

Apparatus Detail - Droplet Controller

At the heart of the droplet controller is a switching circuit that connects the solenoid pump to the 100V power supply on command. This circuit takes a TTL level (e.g. 0 or 5 volts from a microcontroller or other IC) signal as input. When the input is high, it connects the solenoid pump to the power supply, turning the pump on. When the input is low, the power supply is disconnected and no current flows through the solenoid pump, forcing it to return to the resting position. The circuit is based on a N-channel power MOSFET acting as a switch. The MOSFET used is an IRF520N HEXFET from International Rectifier, though any similar MOSFET should work fine. The MOSFET is rated to 100V, which is just at the limit of our operating conditions. The circuit also employs a protection diode across the solenoid pump. A solenoid is simply a coil of wire, so it has a large inductance. If the voltage across an inductor is rapidly switched to zero (as is the case with the droplet signal shown in Figure

4), the inductor will act like a voltage source for a short time to keep current flowing. This can damage the MOSFET, so the diode allows the current created by the pump to circulate in the diode-pump loop without ever reaching the MOSFET. Figure 5 shows a schematic of the switching circuit used in this apparatus.

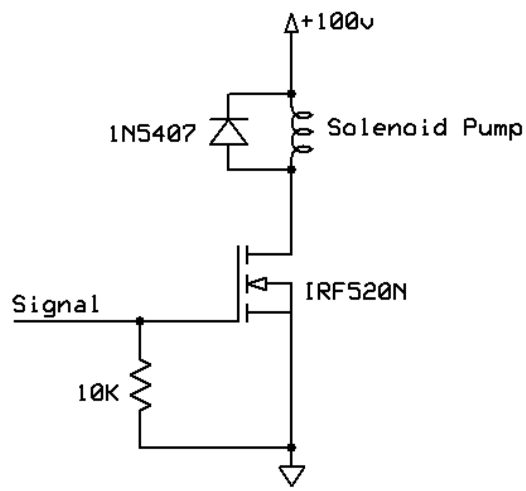


Figure 5. Schematic of the switching circuit used to control the solenoid pump in the droplet colliding apparatus. The circuit uses a MOSFET as the switching component and incorporates a protection diode across the solenoid. Note the signal is a TTL level signal, and the MOSFET conducts when this signal is high (5 volts).

The switching circuit is driven with a signal identical to the one shown in Figure 4, except the high voltage is only 5 volts, with the low voltage still at zero. This, via the MOSFET, creates the voltage profile necessary to create droplets with the pump.

The droplet controller also includes a microcontroller for handling droplet timing and providing a convenient user interface. The Arduino (specifically, an Arduino Duemilanove) microcontroller was chosen for it's simplicity and ability to support rapid circuit prototyping. Arduino boards are based on the ATmega chipset, with the Duemilanove built on an ATmega328 chip.

The enclosure and user interface for the droplet controller was custom-built for this project. Figure 6 shows the droplet collider unit and user interface.

Figure 6. Droplet controller for timing droplets, performing pump maintenance, and calibrating the droplet collider. The connections to the right are for the 100V power supply, and the connections at the back of the device (top of the photo) are for the solenoid pump.

With this interface, a user can make droplet delay timing adjustments (Freq. Adjust), create single droplets (Single), run the pump in an oscillatory continuous mode (Cont.), calibrate the droplet collider, and create droplet collisions (Go). The 100V pump power supply is connected to the jacks on the right, while the pump itself is connected to the jacks at the back of the controller. Note that the protection diode is not included within this enclosure; it must be connected externally to the pump. There is an additional jack at the back of the controller (not shown in Figure 6) for a trigger. In this case, an infrared photogate sensor is used. The green LED partially hidden by the drop frequency adjustment knob is a trigger indicator to display when the trigger is active. The switch on the right of the controller puts the system into calibration mode (indicated by the red LED labeled Calib.), which is described in the Overview of Droplet Collision Procedure section. Additional photographs of the droplet controller appear in Appendix A. There is a USB port not shown in Figure 6 on the left side of the droplet controller. This is used for communication with a computer for reprogramming, but also to read serial data being sent from the microcontroller. This data contains information about the current state of the controller, as well as providing the numerical value of the current droplet delay time in milliseconds.

The circuitry making up the droplet controller is fairly simple, as most everything is handled by the microcontroller. A circuit schematic of the complete controller is shown in Figure 7 below. This schematic includes the switching circuit shown in Figure 5. There is also an NPN transistor used to amplify the trigger signal, which ensures high sensitivity. Two of the switches include RC low pass filters for switch debouncing. This can be handled with software, but a simple RC circuit works equally well. All referenced pin numbers are Arduino board pin numbers, and the Arduino code is available in Appendix B.

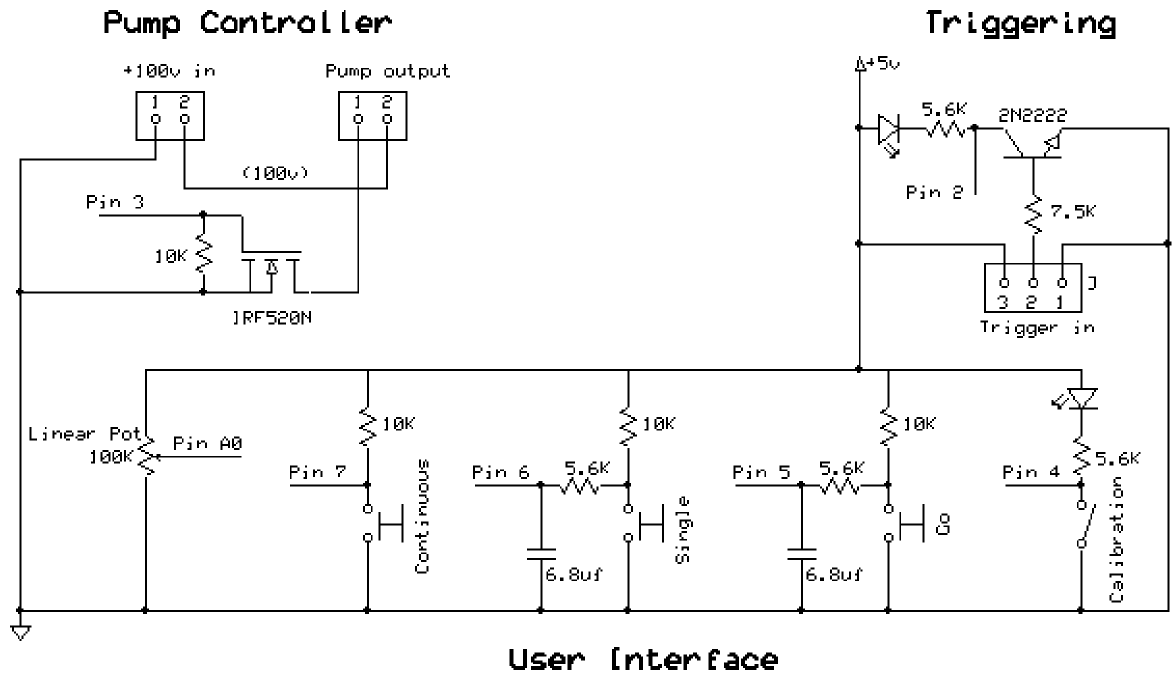


Figure 7. Circuit schematic of the droplet controller. This schematic includes the MOSFET switching circuit shown in Figure 5 (Pump Controller), as well as an NPN transistor for the trigger signal (Triggering). Pin numbers correspond to the Arduino pins, and the necessary Arduino code is given in Appendix B.

Apparatus Detail - Infrared Breakbeam Sensor

The trigger used in this apparatus is an infrared breakbeam (also known as a photogate) sensor. It consists of an infrared light source and infrared phototransistor, and it can sense when an object is obstructing the line-of-sight between the two components. This is perfect for sensing a falling droplet of liquid. This particular IR photogate is made up of a OPV382 infrared VCSEL diode and a PNA1401L NPN phototransistor. A VCSEL is a type of laser diode that is driven like an LED. It outputs a high intensity beam, but with a fairly large divergence angle of approximately 6 degrees. A piece of exposed and developed 35mm photographic negative is used over the phototransistor to attenuate visible light while allowing infrared light to pass. It should be noted that an infrared sensor was chosen over a visible laser system, as the laser light would have been visible in the resulting photographic images. Digital imaging sensors are sensitive to infrared light, but the IR-blocking filter put in place by the manufacturer attenuates IR light enough to make the light from the breakbeam sensor essentially invisible to the camera. While not rigorously tested, this photogate seems to remain functional with a emitter-detector separation distance of about 6 inches. Figure 8 shows the breakbeam sensor as it is positioned in the droplet collider apparatus. The sensor is placed 1-2 inches above the liquid surface such that it can capture both falling droplets and the resulting splash jet traveling upwards. This is critical for system calibration, as discussed in the Overview of Droplet Collision Procedure section.

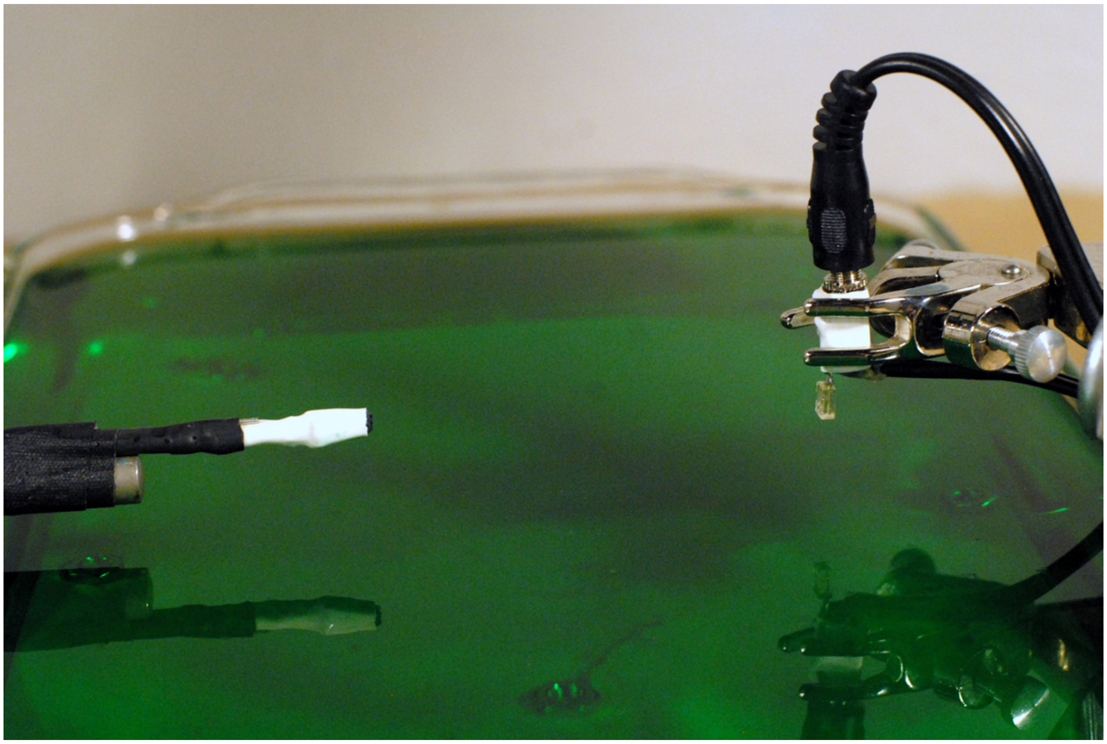


Figure 8. Infrared breakbeam sensor positioned over the splash basin. The phototransistor (detector) is on the left, and the infrared VCSEL (emitter) is on the right. A falling droplet occludes the detector, which can be read as a TTL level change on the sensor output.

A circuit schematic of the infrared breakbeam sensor is shown in Figure 9. Note the 2N2222 transistor within the droplet controller (Figure 7) is not considered part of the sensor circuit.

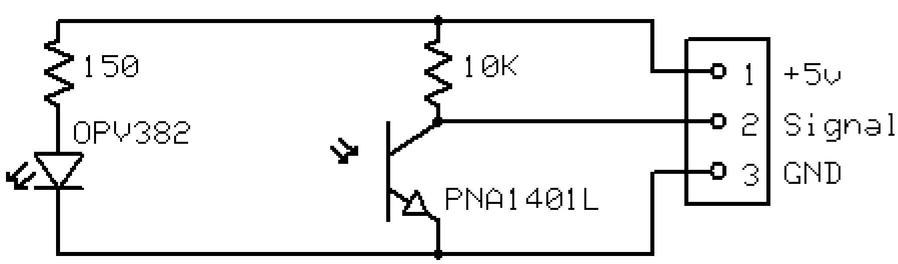


Figure 9. Circuit schematic of the infrared breakbeam sensor. The connector at right is a 3.5mm 3-conductor headphone-style plug.

The sensor terminates with a standard 3-conductor 3.5mm headphone-style plug. The pinout and plug type are identical to and therefore compatible with the Mumford Time Machine camera/strobe controller. The Mumford sensor pinout is as follows:

Table 1. Mumford Time Machine Sensor pinout for a 3 conductor 3.5mm plug

Conductor	Function

Sleeve (base)	Ground
Ring (middle)	TTL Signal
Tip (end)	+5V

The signal is low when the sensor is inactive, but goes high (5V) when active. This corresponds to a high output when a droplet is breaking the infrared beam. This is fully compatible with the Mumford system. However, note that when the sensor is plugged into the droplet controller, the microcontroller is actually looking for an active low signal, as the 2N2222 transistor circuit shown in Figure 7 inverts the signal (i.e. pin 2 goes low when the sensor input goes high).

The photogate is designed in such a way that the emitter can be detached from the detector. This way, an extension cable can be used if a longer emitter-detector separation distance is required. This is also done with 3.5mm jacks and plugs for consistency. The pinouts of these connectors are identical to the one listed in Table 1.

Apparatus Detail - Strobe Controller

The strobe controller used for taking strobe photographs of splash collisions is simply the Mumford Time Machine. This device is a multipurpose strobe and camera controller, available from Mumford Microsystems. In fact the Time Machine (with a few add-ons) can act as a droplet controller. However, as discussed in the Prior Work section, it has limited functionality, and is not easy to use. So, despite the high functionality of the device, it is used as a simple delay unit in our custom droplet collider apparatus. The strobe controller is fully independent of the the droplet controller. As discussed in the Future Work section, we hope to include the strobe controller as a part of the droplet controller, using the microcontroller to handle the strobe delay. However, our present apparatus still relies on the Time Machine, and will therefore be discussed here.

The Time Machine takes the custom infrared breakbeam sensor into the Sensor input, and the strobe is controlled via the Flash output. The particular strobe being used is not critical, though we used a SPOT 500ns strobe from Prism Science Works. Note that both the Time Machine and the droplet controller have a sensor input. Currently, the droplet controller sensor input is used only for initial calibration (see Overview of Droplet Collision Procedure section), and otherwise the sensor is used with the Time Machine strobe controller. The Time Machine is put into Shoot mode with the Flash Lag at 0 and the Flash Timeout at 3 seconds. The Trigger Delay can be adjusted to capture the desired moment in a splash collision. The sensitivity is also freely adjustable to whichever setting gives the best results. For this apparatus, the sensitivity was generally placed about halfway between the minimum and maximum settings.

Figure 10 shows the physical strobe controller, along with the droplet controller and power supply. Additional images of the complete apparatus appear in Appendix A.

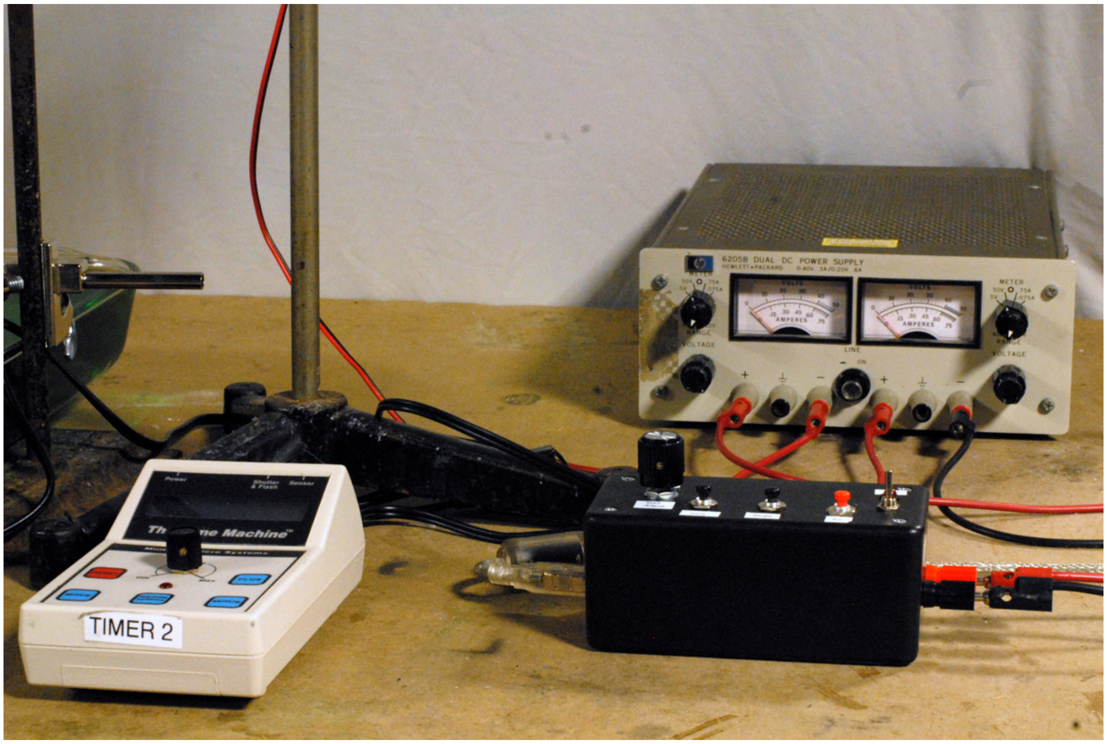


Figure 10. Overview of the apparatus control systems. The strobe controller is shown on the left, the droplet controller is at the front right, and the pump power supply is at the rear right.

Overview of Droplet Collision Procedure

Setup of Apparatus

The droplet apparatus is a relatively self-contained device, but does require some extra setup before photos can be taken. The basic setup procedure is described below.

Making Connections

First, the power supply, pump, and pump controller need to be connected. The basic connection scheme is shown in Figure 2. Two banana plugs connect the power supply to the pump controller at the banana jacks labeled "100V In" and two more cables connect the pump to the controller at the jacks labeled "Pump In". The pump controller is powered via the USB connection, so the controller must be plugged into a computer as well. The breakbeam sensor will eventually communicate with the strobe controller, but it is best to plug it into the pump controller for initial setup and calibration. This goes in the 3.5mm headphone-style jack labeled "Trigger." This is also a good time to begin the a USB serial monitor via the Arduino programming environment. The serial monitor allows for communication between the droplet controller and the user.

Preparing the Pump

Liquids should be set up with any coloring or additives desired. The pump can be primed with the "continuous" button on the pump controller. At this point, it is important to ensure there is no air in the system. This can be done by opening the air bleed valve, sealing the droplet nozzle opening, and running the pump until liquid flows out of the air bleed valve. The air bleed valve should be closed before the droplet nozzle is unsealed, otherwise the brass nozzle tubing will purge the liquid. A few test droplets can confirm that the pump is prepared correctly.

Aligning the Sensor

The breakbeam sensor is not strictly necessary for high speed video, especially for those with experience in splash collisions. However, it can be useful to set up the breakbeam sensor for more reproducible results or for calibration purposes. If strobe photos are going to be taken, the breakbeam sensor is critical to timing the pictures. Many things must be properly aligned for this to work, which is not easy with the current apparatus. First, the infrared detector must be pointed directly at the infrared emitter. Additionally, the droplet nozzle must be aligned such that the droplets fall directly through the light path from emitter to detector. This can sometimes take a lot of tweaking to get just right, but it is generally easiest aligning the emitter/detector pair first, and then moving the nozzle to the location prescribed by the emitter/detector pair. Due to the invisibility of infrared light to humans, any digital camera with a live view (such as a camera phone or point-and-shoot camera) can be used to view the infrared light being emitted by the VCSEL. This is due to the fact that digital imaging sensors are inherently sensitive in the infrared spectrum. Despite manufacturers' efforts to filter out infrared radiation, some infrared light still reaches the imaging sensor and can be seen as a purplish glow. In order to be useful for calibration, the breakbeam sensor should be placed as close as possible to the apex of a splash jet. However, locations below the apex are acceptable while locations above the apex will not allow a proper calibration.

If a strobe is being used for photographs, it is convenient to check the sensor alignment by triggering the strobe off the sensor output directly with no delay. Thus, if everything is aligned properly, the user will see a single droplet hanging in midair just between the breakbeam sensor shortly after a droplet is created. Sometimes, the alignment will be off just enough that a falling droplet will not trigger the sensor, but instead the resulting jet will trigger the sensor. This creates the illusion that the sensor is aligned because the sensor triggers after each droplet, but it is actually incorrect. This offset in timing can make collisions difficult. If strobe flashes are revealing jets rather than hanging drops, the alignment should be checked.

Calibrating the Droplet Collider

For those inexperienced in droplet collisions, the droplet controller provides an easy-to-use calibration feature. The most critical time interval for producing collisions is the time between droplets. With experience, a user can set these based on previous results to get the desired photo. However, without this experience, it can be hard to determine a starting point.

The system must be fully set up before calibrating. Ideally one would re-calibrate after every setup change, but often this is not necessary. Additionally, even if an external strobe controller (such as a Time Machine) is being used, the sensor should be plugged into the droplet controller for the duration of the calibration. This is because the droplet controller uses feedback from the breakbeam sensor to determine a ballpark droplet timing interval. For calibration, the droplet controller must be switched into calibration mode. This is done with the "Calib" switch. When the controller is in calibration mode, the red indicator LED will light. To actually begin the calibration procedure, first drop a few single droplets with the "Single" button. This will prime the droplet system and ensure the water is level with the outlet of the nozzle. Then, press the "Go" button to initiate the calibration routine. The controller will automatically calibrate itself while giving the user information via the computer serial connection.

A calibration cycle consists of five independent droplets released with 3 seconds between them to allow the splash basin surface to settle. For each droplet, the first and second sensor beam breaks are recorded. The first trigger corresponds to the droplet free-falling through a prescribed point in space (i.e. the line of infrared light between the beambreak components). The second trigger corresponds to the splash jet subsequently occupying the same point in space. The time delay between these two events gives a good estimate for the time needed between droplets, as that ensures a second drop is occupying the same point in space and time as the previous splash jet.

Collision Procedure

Photographic Procedure

Example Photographs

The following images were obtained using the outlined apparatus and procedure. Note that the lighting used was not always identical to that depicted in Figure ??. The camera settings, as well as liquid and timing details are listed beneath each image. More images are available at this gallery.

Figure ??. SPOT strobe, Nikon D200, Tamron 90mm macro lens, f/14, 400ISO, colored water and glycerin. 105 ms flash delay, 110 ms droplet delay, 5100 us droplet formation time.

Figure ??. SPOT strobe, Nikon D200, Tamron 90mm macro lens, f/14, 400ISO, 91% isopropyl alcohol. Timing information not recorded.

Figure ??. SPOT strobe, Nikon D200, Tamron 90mm macro lens, f/20, 400ISO, water**.** 145 ms flash delay, 140 ms droplet delay, 5263 us droplet formation time.

Figure ??. SPOT strobe, Nikon D200, Tamron 90mm macro lens, f/18, 400ISO, water**.** 96 ms flash delay, 100 ms droplet delay, 5000 us droplet formation time.

Figure ??. SPOT strobe, Nikon D200, Tamron 90mm macro lens, f/25, 800ISO, colored water and glycerin**.** 173 ms flash delay, 179 ms droplet delay, 5260 us droplet formation time.

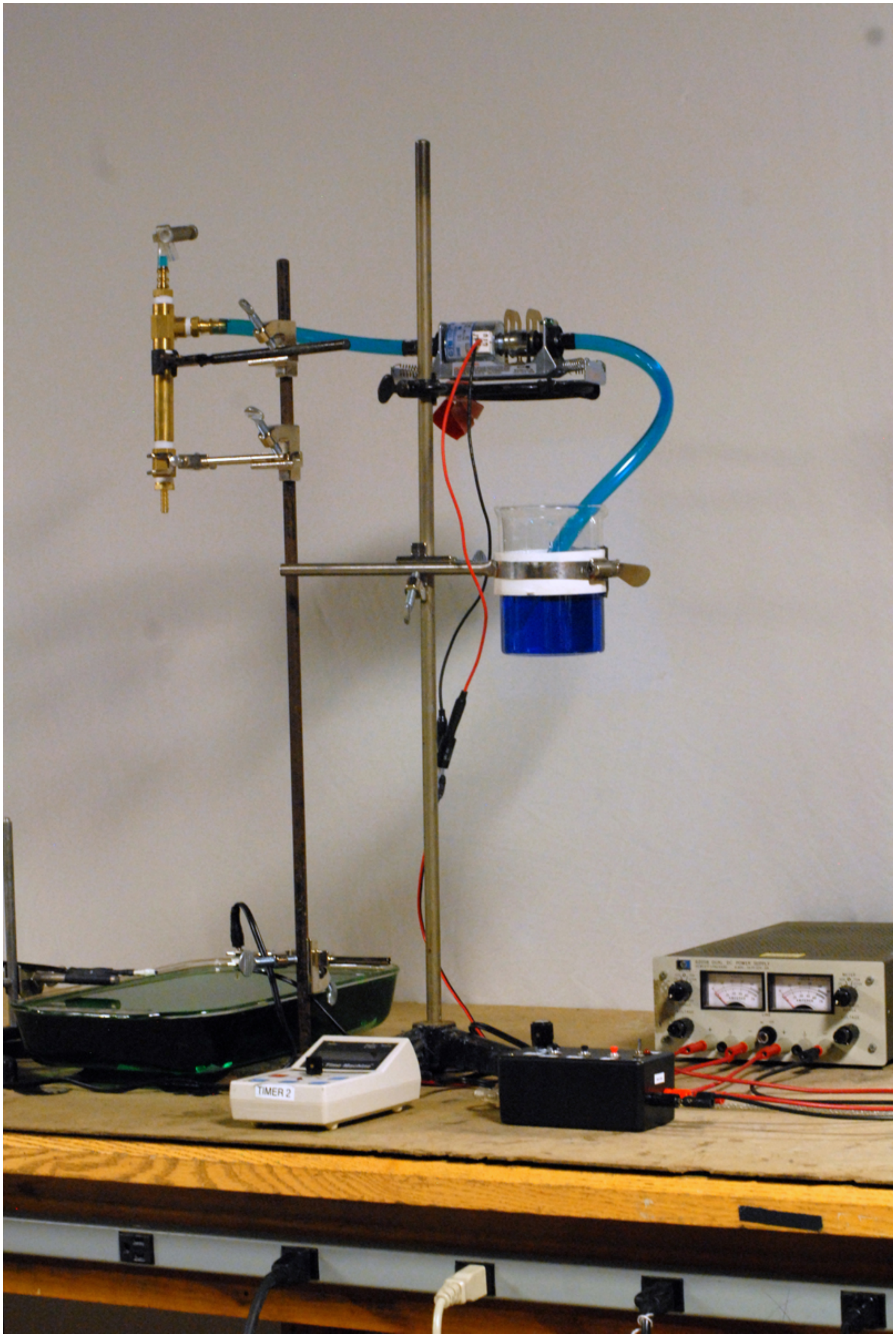
Problems

Future Work

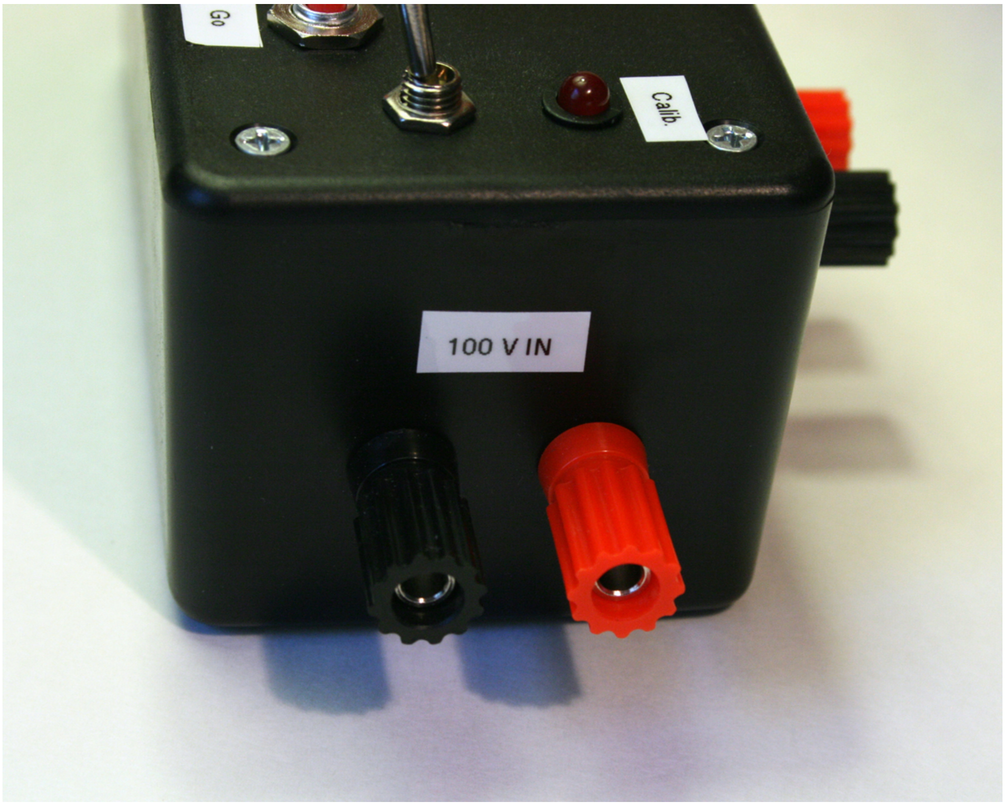
Appendix A - Additional Apparatus Images

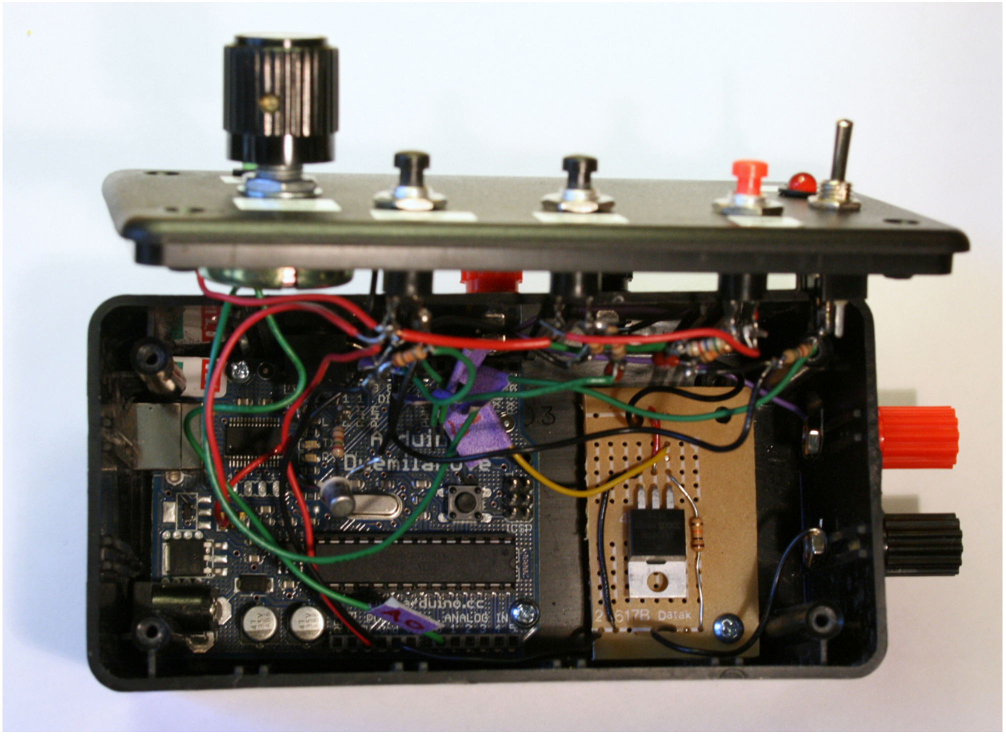
This appendix provides additional images of the apparatus and photographic setup for reference purposes.

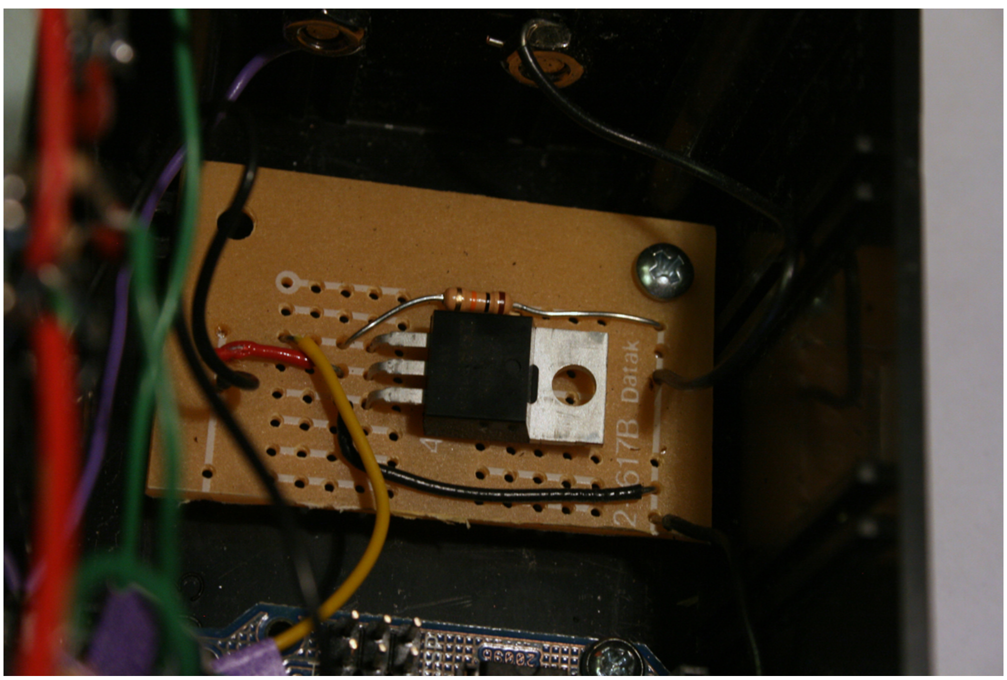
Complete Droplet Collider Apparatus


TIMER 2

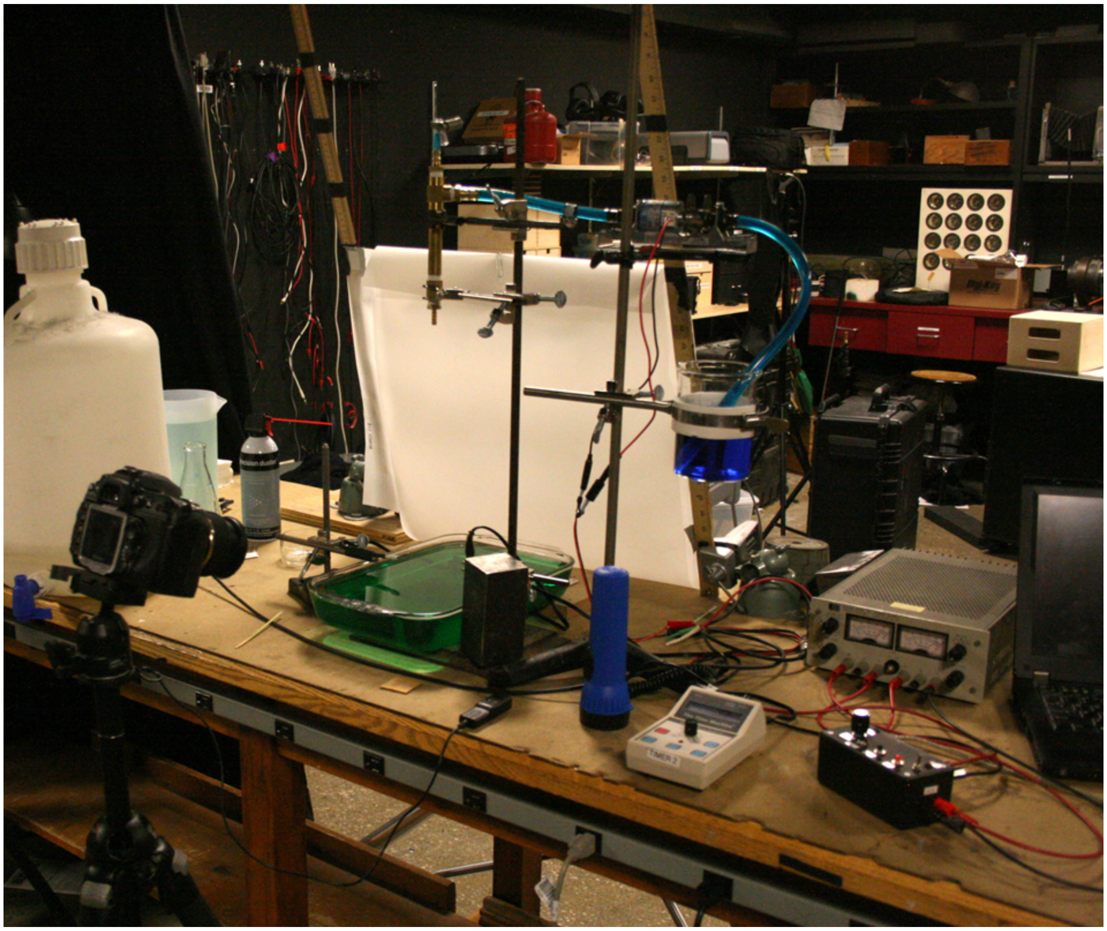
Droplet Controller

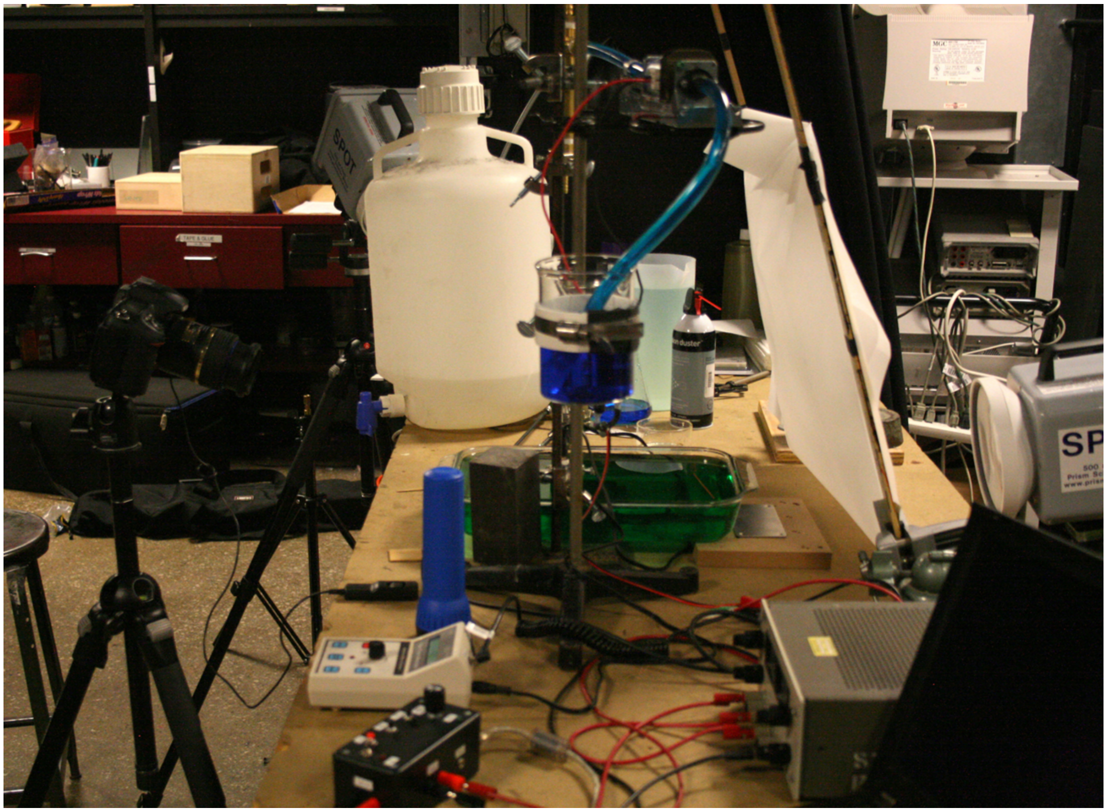


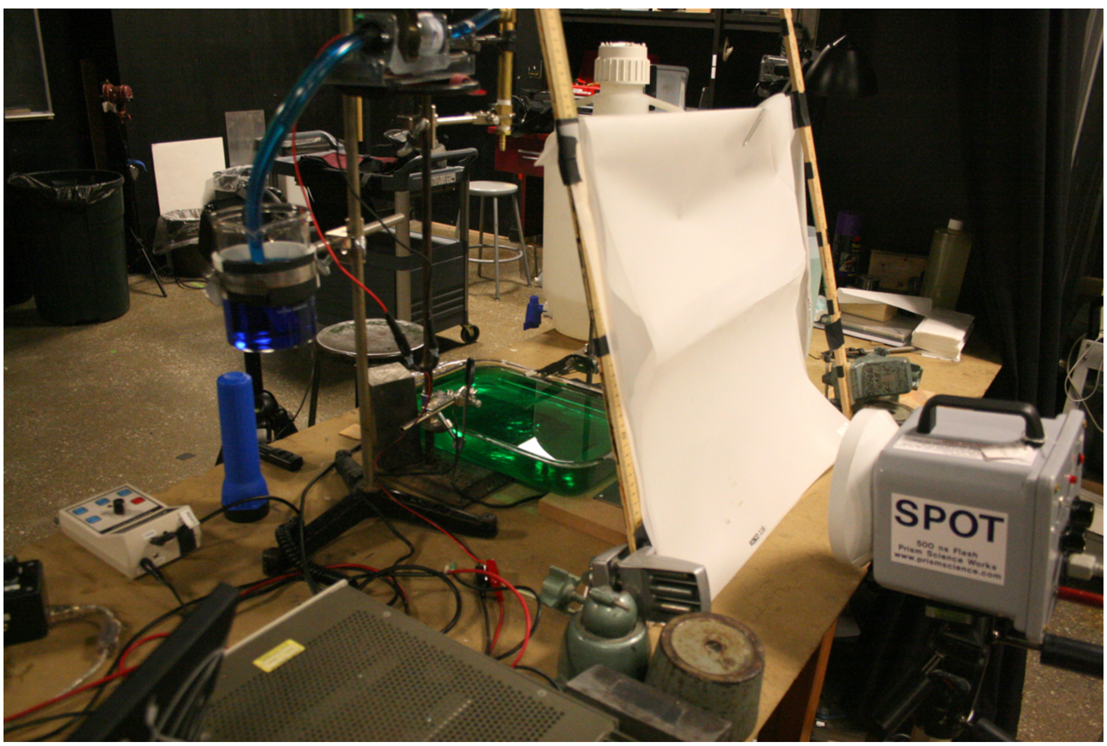




Photographic Setup







Appendix B - Arduino Code

This appendix provides the Arduino microcontroller code used to run the droplet controller. Note that most of the critical portions are commented with explanations, though there is likely a large fraction of the code that is unexplained. Additionally, we don't claim that this code is highly efficient, though it seems to work for our purposes. There is a section of the code related to strobe control and timing, but it is currently commented out. Once the strobe controller is implemented as part of the droplet controller, that part of the code will be made active.

waterCollider

```
/* waterCollider by Trevor Shannon 2010.
Used to create water droplets with a solenoid pump and time said droplets such
that a second drop collides with the splash of the first.  Also includes a
calibration function to establish a time interval between droplets to ensure a
collision.
*/
//digital pins
int sensorPin = 2;
int dropletPin = 3;
int calibrationPin = 4;
int goPin = 5;
int singlePin = 6;
int runPumpPin = 7;
int strobePin = 8;
//analog pins
int freqAdjustPin = 0;
//int strobeAdjustPin = 1;

int dropFormationFreq = 5260; //us, 5000 = 100 hz, 5102 = 98 hz, 5263 = 95 hz
int dropFreq = 125; //ms
int calibDropFreq;
int calibDropFreqList[5] = {0,0,0,0,0};

int safetyDelay = 6000; //ms delay to protect the strobe from firing too often

int breakCount = 0; //keeps track of number of beam breaks for calibration
int calibCounter = 0; //allows for multiple sequential calibration runs
long calibStart;  //records when calibration began
long collisionStart; //records when a collision began

int sensorReading = HIGH;
int previousSensorReading = HIGH;

unsigned long breakTime[2] = {0, 0}; //stores times when the photogate is broken

int delayAdjustZero = 0;
int adjustScaleFactor = 5; //converts rotary change into milliseconds change.
increase for more precision

int go = LOW;  //whether or not a drop collision sequence should start
int previousGo = LOW; //record previous state of the go button
int single = LOW; //for the manual droplet button
int previousSingle = LOW;

boolean calibration = false; //whether or not calibration mode is active

```

```
int sum = 0;
double percentError = 0.3;

void setup(){
pinMode(sensorPin, INPUT);
pinMode(calibrationPin, INPUT);
pinMode(goPin, INPUT);
pinMode(singlePin, INPUT);
pinMode(dropletPin, OUTPUT);

Serial.begin(9600);

  delayAdjustZero = analogRead(freqAdjustPin);  //initialize delay dial setting
  calibDropFreq = dropFreq;
}

void loop(){

  calibration = digitalRead(calibrationPin);
  go = digitalRead(goPin);
  single = digitalRead(singlePin);

////calibration////
  if (calibration == true){
if (go == true && previousGo == false){ //go button has been pressed
      Serial.println("calibrating...");
      calibCounter = 0;
while (calibCounter < 5 && digitalRead(calibrationPin) == HIGH){
      breakCount = 0; //reset break count
      calibStart = millis();
///create one droplet///
      digitalWrite(dropletPin, HIGH);//note can't use makeDroplet() because of
the delay at the end of that funtion.
      delayMicroseconds(dropFormationFreq);
digitalWrite(dropletPin, LOW);
////////////////////////
      while (breakCount < 2 && digitalRead(calibrationPin) == HIGH){
        sensorReading = digitalRead(sensorPin); //query the photogate
        if (sensorReading == LOW && previousSensorReading == HIGH){ //the
photogate is triggered (low -> high)
          breakTime[breakCount] = millis(); //record time of beam break
          breakCount ++; //add one to the break counter
          delay(10); //simple debouncing
        }
if (breakCount == 2){ //droplet has fallen and jet has passed
          calibDropFreqList[calibCounter] = breakTime[1] - breakTime[0];
//calculate drop frequency from recorded calibration times
          Serial.print("calibration ");
Serial.print(calibCounter);
Serial.print(": ");
Serial.println(calibDropFreqList[calibCounter]);
          sum = 0;
for (int j=0; j<calibCounter; j++){
            sum += calibDropFreqList[j];
          }
if (calibCounter == 0){
            calibCounter ++;
          }
```

```
else{
if (calibDropFreqList[calibCounter] < (1+percentError)*(sum/calibCounter) &&
calibDropFreqList[calibCounter] > (1-percentError)*(sum/calibCounter)){
              calibCounter ++;
            }
else{
Serial.println("bad calibration.  retrying.");
            }
          }
delay(3000);
        }
if (millis() - calibStart > 5000){//if calibration isn't finished in 5 seconds,
something went wrong.
          Serial.println("calbration error: timeout.");
          breakCount = 2; //fake a calibration
          //calibration == false;
        }
        previousSensorReading = sensorReading;
      }
    }
    sum = 0;
for (int j=0; j<5; j++){
      sum += calibDropFreqList[j];
    }
if (calibCounter == 5){
      calibDropFreq = sum/5;
      dropFreq = calibDropFreq;
Serial.print("calculated drop frequency: ");
Serial.println(dropFreq);
    }
    delayAdjustZero = analogRead(freqAdjustPin); //set current dial position to
the zero point
    }
  }
////end calibration////

else{ //calibration mode is off
    ///run the pump continuously///
    if (digitalRead(runPumpPin) == HIGH){
Serial.println("pump running");
      runPump();
    }

    dropFreq = calibDropFreq + (analogRead(freqAdjustPin) - delayAdjustZero)/
adjustScaleFactor;
//strobeDelay = calibDropFreq + (analogRead(strobeAdjustPin) -
strobeAdjustZero)/adjustScaleFactor;
    Serial.println(dropFreq);
///make collision///
    if (go == HIGH && previousGo == LOW){ //go button is pressed
      collisionStart = millis();
Serial.println("collision initiated.");
for (int i=0; i<2; i++){ //runs the makeDroplet() code twice.
        makeDroplet();
      }
/*while (true){
if (digitalRead(sensorPin) == LOW){
delay(dropFreq); //should be adjustable.
```

```
digitalWrite(strobePin, HIGH);
delay(20);
digitalWrite(strobePin, LOW);
Serial.println("lockout period...");
delay(safetyDelay);
break;
        }
if ((millis() - collisionStart) > 5000){ //prevents an infinite loop if the
sensor fails.
Serial.println("strobe error: timeout");
break;
        }
      }*/
    }

///make a single droplet manually///
    if (single == HIGH && previousSingle == LOW){ //single droplet button is
pressed
      Serial.println("single droplet");
      makeDroplet();
    }
  }
  previousGo = go;
  previousSingle = single;
}

void runPump(){
digitalWrite(dropletPin, HIGH);
delay(10);
digitalWrite(dropletPin, LOW);
delay(100);
}

void makeDroplet(){
digitalWrite(dropletPin, HIGH);
delayMicroseconds(dropFormationFreq);
digitalWrite(dropletPin, LOW);
delay(dropFreq-(dropFormationFreq/1000));
}
```

Appendix C - Hyperlinked URLs

This appendix contains a list of the URLs for all the linked websites in this document. The URLs are listed in order of their appearance.

- **Martin Waugh's website:** liquidsculpture.com/
- **Mumford Microsystems Time Machine:** http://www.bmumford.com/photo/camctlr.html
- **Time Machine Drip Kit:** www.bmumford.com/photo/waterdrops/dripkit.html
- **Time Machine splash collision gallery:** www.flickr.com/groups/timemachine/
- **Gorman-Rupp 14825-601 pump:** www.gripumps.com/series.asp?action=display&id=50
- **IRF520N MOSFET datasheet:** www.irf.com/product-info/datasheets/data/irf520n.pdf
- **Arduino Duemilanove:** arduino.cc/en/Main/ArduinoBoardDuemilanove
- **OPV382 VCSEL datasheet:** pdf1.alldatasheet.com/datasheet-pdf/view/151469/OPTEK/OPV382.html

- **PNA1401L phototransistor datasheet:** www.datasheetcatalog.org/datasheet/panasonic/SHE00001AED.pdf
- **Photographic results gallery:** picasaweb.google.com/trevor.j.shannon/6163StrobeLabWaterCollisions#