12-8-2020

Silicon Oil Drop Experiment

Jacob Lutz
Western Michigan University, mr.jake_l@yahoo.com

Western Michigan University

College of Engineering and Applied Science

ECE 4820: Senior Design II

Automated Silicone Oil Droplet Generator

Myles McPherson Jacob Lutz Rowan Sheehan

Advisor: Prof. Durbin

WESTERN MICHIGAN UNIVERSITY

COLLEGE OF ENGINEERING AND APPLIED SCIENCES

DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING

KALAMAZOO, MICHIGAN 49008

SENIOR DESIGN PROJECT REPORT RELEASE FORM

In accordance with the "Policy on Patents and Release of Reports Resulting from Senior Design Projects" as adopted by the Executive Committee of the College of Engineering and Applied Sciences on Feb. 9, 1989, permission is hereby granted by the individuals listed below to release copies of the final report written for the Senior Design Project entitled:

PROJECT TITLE: **Automated Silicone Oil Droplet Generator**

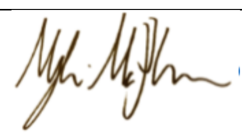
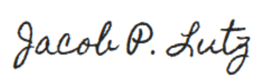
PROJECT SPONSOR* Did this project have a sponsor? **No**

Contact person and email address &/or telephone ______Dr. Steve Durbin___ steven.durbin@wmich.edu___

Company Name__________________________NA____

Design team has requested sponsor to verify in writing to course coordinator that all promised deliverables have been received. **YES____NO_X__** (please check)

TEAM MEMBERS NAMES:

Names	Signature	Date
Rowan Sheehan		12/08/2020
Myles McPhearson		12/08/2020
Jacob Lutz		12/08/2020

* Those teams with a sponsor must have sponsor provide the **course coordinator** with written evidence that they have provided the sponsor with a copy of the final project report as well as with other items that the team has promised to the sponsor. The evidence could be a short note via email, fax or US mail from the sponsor indicating receipt of a copy of the report and all promised deliverables

Table of Contents

I. Summary

The Automated Silicone Oil Droplet Generator can provide an easily obtainable way to recreate famous quantum-based experiments on a macro scale. By having a self-contained system that can run continuously, the large number of trials required to recreate the experiments can be achieved.

The Droplet Generator consists of three sub systems: the mechanical system, the electrical system, and the computer system. The mechanical system includes the main oil droplet generator device, an electromagnetic shaker that holds the experimental forms, and an oil system that converges on a centralized reservoir. The electrical system consists of variable DC power source that drives the oil pump and delivers the voltage to the h-bridge, the function generator that provides the sinusoidal function to the electronic shaker, the audio amplifier to driver electronic shaker, the Arduino that controls the h-bridge, and the piezoelectric disc that pushes the droplet into the experimental form. The computer system tracks the droplets in real time and triggers the electrical system for droplet generation. It also automatically records data for future analysis within a CSV file.

II. Introduction

The use of automation has no dauntedly revolutionized every aspect of life from production, to medicine, and most importantly, scientific research. The Pilot Wave theory has been a notable, but often overlooked contender with the Copenhagen interpretation of quantum mechanics. When designing models for quantum systems, researchers are often limited to computer models to develop a large pool of data. This has always been the case until the discovery was made that bouncing droplets of silicone oil can potentially be a representation of quantum mechanical particles.

There is a need to develop an automated system of the bouncing droplets that will allow for a large pool of data to be collected of the without the need for people to reset the system. The automation would allow for time between trials to be diminished. For a system to have a significant impact on research, it must be able to run continuously for many trials without the need for human interaction and collect the data created by the movement of the oil droplets.

This document is a project report for the Automated Silicone Oil Droplet Generator. Dr. Steve Durbin of Western Michigan University’s department of Electrical and Computer Engineering will be advising this project.

III. Overview

A. Related Work

Harris, et al. [1] created a design for a mechanical droplet generator in 2015. The system was composed of a fluid reservoir, pump, transitional stage, piezoelectric buzzer, a fluid chamber, and a nozzle. The droplet generator uses the nozzle to create the optimal droplet diameter. This creates a small amount of variance in the droplets produced which can be reduced to less than five percent. The vibration of the silicone oil can be caused by a small subwoofer. Litwhiler, et al., [2] created a system to power and operate a small subwoofer using a power supply, a function generator, and a power amplifier. The system also uses a refence accelerometer to provide visual feedback with the use of an oscilloscope.

The Quantum Particle Modeling System will consist of the subsystems of the block diagrams shown in figures one through three. The first subsystem will create the droplets being observed. The droplet generation subsystem will be based on the Harris and team's design with modifications to allow for automation. The second subsystem will be mechanical vibrator using the subwoofer to vibrate the pool of silicone oil. The third subsystem will be the camera tracking system that is responsible for the processing of the droplet locations. This will use the image processing from visual basic to collect the droplets path and location. Once the droplet has completed the path, the program will send a signal to allow the production of another droplet.

B. Block Diagram

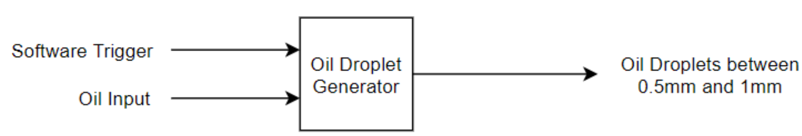


Figure 1: Tank vibration driving system overview

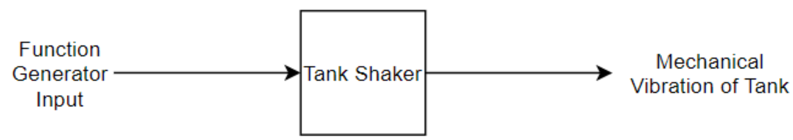


Figure 2: Oil droplet generator system overview

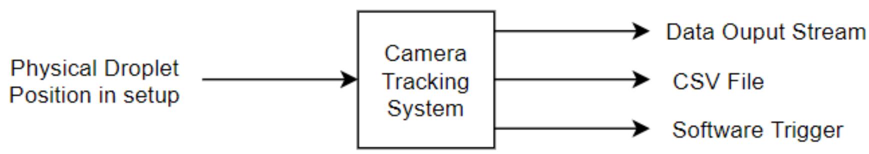


Figure 3: Camera tracking system overview

C. *Specifications*

1. General

1.1 Create a device that can create oil droplets on demand. Device must have the ability to replenish oil used during operation. Device must be driven electrically.

1.2 Create an experiment form that allows oil to be replenished without altering the parameters of the experiment.

1.3 Create a software program that can identify droplets, track droplets, record needed data points, and communicate with the rest of the system to initiate another droplet.

1.4 Create an electrical system to drive the mechanical device that creates oil droplets.

D. *Deliverables*

The system will need to perform the following specifications:

1. To detect when a droplet has completed its trial to create the next droplet
2. To process the video feed to track and record the position data of the droplet

3. Computations of the position data for analysis dependent on the experiment

E. Acceptance Testing

The project will be considered acceptable if:

1. The setup can run 2500 trials autonomously
2. Requires no replenishing of additional oil, i.e., system is all-in-one
3. System can track and record droplet position with statistical processing in real time

F. Parts List

The bill of materials is a list of all primary components used in the creation of the device. Software packages used to model/code are provided free from either the university or their respective manufacturer. The full bill of materials is listed in Appendix C. The experiment main experimental parts are listed below.

Experiment Parts:

1. Droplet Mechanism
2. Computer/Raspberry Pi
3. Webcam/Pi Camera
4. Silicon Oil
5. Speaker
6. Amplifier
7. 3D Printed Bases
8. Oil Reservoir

G. Relevant Coursework to Project Technologies

1. IEE 2610
2. ECE 3200
3. ECE 3100
4. CS 1200

IV. Design and Simulation

A. Oil Droplet System

All components of the project are contained in a 16x18x24” structure. The main structure consists of six plywood pieces bolted into two symmetrical “C” structures. More ½” plywood paneling was added to the top and back of the structure to allow mounting and suspension points for the electronics, reservoir, and droplet generator. A 6” wide shelf was added to the back of the project to allow room for extra weight and storage of experiment forms.

The main droplet generator device, figure 4, is based on the 3D printed setup created by Harris’s team with modifications to allow for droplets to be produced and recycled continuously [1]. The design utilizes a concept from inkjet printing where the mechanical movement from a piezoelectric component drives a droplet from a fluid chamber by changing the chamber’s volume. Our device was constructed using AutoCAD fusion360 files that were provided by Dr. Bush from MIT and were fabricated in Battle Creek by a local machine shop, Springfield Machine and Plumbing. The device is suspended directly over the experiment form on four ¼-20 threaded rods to allow for direct dropping of the silicone oil without the aid of an additional transition stage. After setup, droplets are created in the fluid chamber by the actuation of a piezo electric disc that is centered on the top part of the chamber. Oil droplet diameter is controlled by the diameter of the hole at the tip of the nozzle. Most testing was completed with a 0.6mm nozzle.

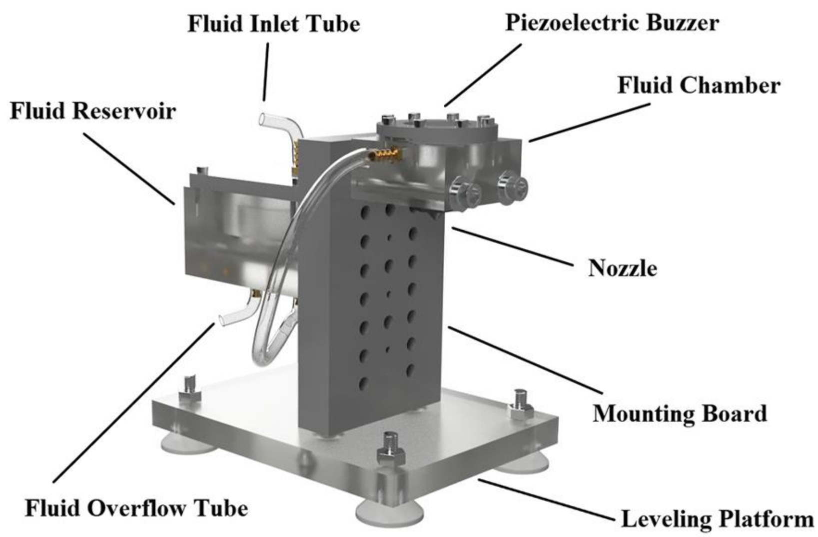


Figure 4: Droplet creation set up [1]

The electromagnetic shaker consists of an 8” subwoofer installed within the 10x10x5” plywood box. This box sits on top of 2” of foam which helps to dampen the mechanical oscillations from the subwoofer to the rest of the experiment structure. By not being attached to the main structure, the shaker can be moved to a desired position for both the dropper and imaging system. A holder for experimental forms is adhered to the top of the subwoofer allowing users to mount various experimental forms with the assistance of two small spring clamps. Additionally, the shaker connects to the amplifier by an Anderson connector mounted to the right side of the experimental structure, thus opening the possibilities of using other sized shakers with different form holder styles.

In order to meet the continuous operation requirement, two modifications were added to the experiment forms to allow the piezoelectric device to directly drop into the form, as well as to drain excess oil that would otherwise overfill the experiment area. For the first modification, an area was reserved on the experiment form that would allow the piezoelectric device to sit over the form without physically blocking the imaging system. The second modification was the addition of a weir to control the oil level as well as a drain hole to allow any overflow to move back into the systems main reservoir. The weir is positioned such that a fixed level of oil always remains in the experimental area of the form. All changes were made using the open-source 3D modeling software FreeCAD.

B. Electronics

The circuit created for this device is relatively simple when compared to the mechanics of said device. In short, the circuit needs to be able to provide a ±30V signal when triggered by the imaging software from the raspberry pi. The 30V pulse is delivered by the SN754410 Quad Half-H Driver to the piezoelectric chip causing actuation of a drop from the fluid chamber. The length of the pulse is determined by the resonant frequency of the piezoelectric disc, in our case 2.8 kHz. The driver IC is controlled by an Arduino Nano which provides the driver with both a 5V logic voltage supply, as well as the digital control voltages.

The 30V is supplied to the Half-H driver through a GW Instek 2-channel adjustable DC power supply. This same power supply is also responsible for precision control of the DC motor voltage for the peristaltic pump. The DC power supply connects via banana connection points on the outside of the control electronics box.

A custom PCB was designed and created using KiCAD. Due to the lack of online resources, libraries and footprint components had to be made for both the Nano and Half-H Driver. Beyond the complementary components pre-installed on the nano, complementary 100µF capacitors were added to the VCC supplies for the Half-H Driver along with pull down 1kΩ resistors on the digital inputs. To make future soldering easier, all components were chosen to be through hole components (THT).

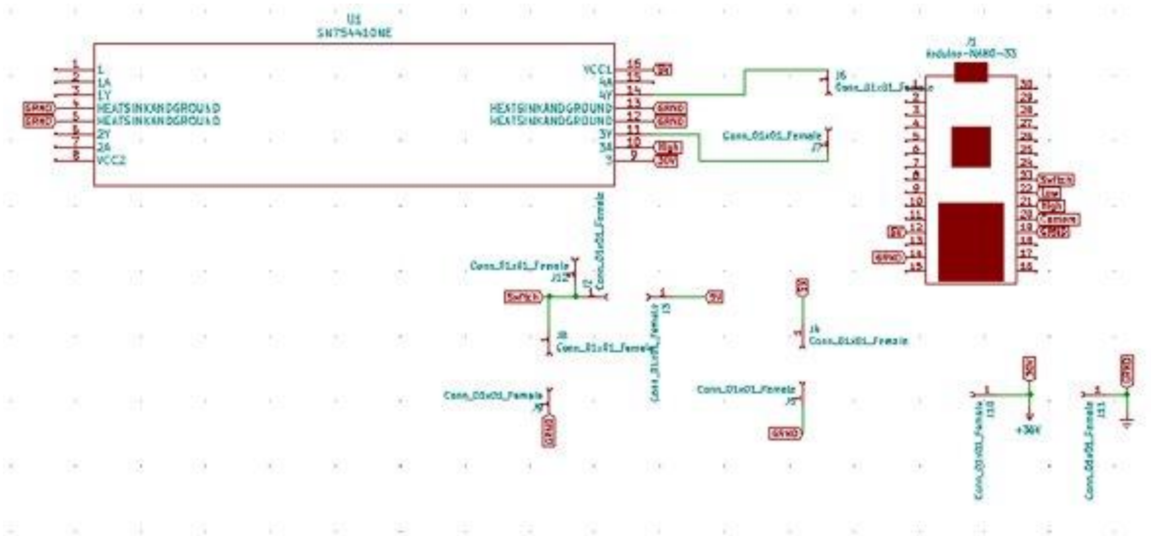


Figure 5: Circuit Diagram

Power is provided to the Arduino and Raspberry Pi via an Anker USB block capable of supplying up to 2.8A to either device. The USB hub is tied into the 120V AC terminal connections on the 12V DC power supply powering the electromagnetic shaker's amplifier. This helps to reduce the amount of 120V plugs the system uses. The 12V DC power supply was originally provided with the project and supplies the subwoofer amplifier. All components are mounted to the back of the main plywood structure.

Droplet Tracking System

The machine vision and tracking system is implemented on a Raspberry Pi 4b using the Raspberry Pi Camera Module. This was chosen due to its ease of implementation and overall processing capability. The tracking program was created using OpenCV and python on the Mu programming IDE that comes with the Raspberry Pi. A few different tracking algorithms were tested, including different motion detection and object detection functions. The Hough Circle Transform was chosen, as it allows for easy function tuning, it was a relatively quick program, and the droplets being monitored are circular.

Once the tracking algorithm was selected, some extra features were added for easier tuning and data collection. On code startup, a CSV writer is created, titled after the date and time, and then is prepared to log each time the droplet passes a certain plane (that being the back wall of the mold). Next, the master loop begins and images from the camera module are taken and transformed to grayscale for tracking. This is then put through the Hough Transform to identify circles.

Motion based tracking is more processing intensive because it must compare the current image with the previous image and run something like a Kalman filter to find moving objects. The Hough Transform (as set up in this instance) instead finds the best circle within the image.

Once the droplet is found, it is then displayed (a circle in the window that shows the live camera image and the centroid in the blank window to show the droplet through time). The coordinate of the circle is then kept comparing against the next circle in the next image. If the line between those two coordinated crosses the plane as mentioned above, the coordinate is logged in the open CSV file for future evaluation and the loop continues again.

While this loop repeats itself, the FPS is calculated for diagnostic and tuning. This is very helpful. The Hough Transform is tuned to maximize FPS, because having more frames of the circle throughout its lifetime allows for better tracking granularity.

Once the data is collected and the program is stopped, the CSV file is automatically saved. This file can then be opened in Excel or in the excel-like program installed on the Raspberry Pi. It can then easily be graphed as a histogram to see which coordinates were most common.

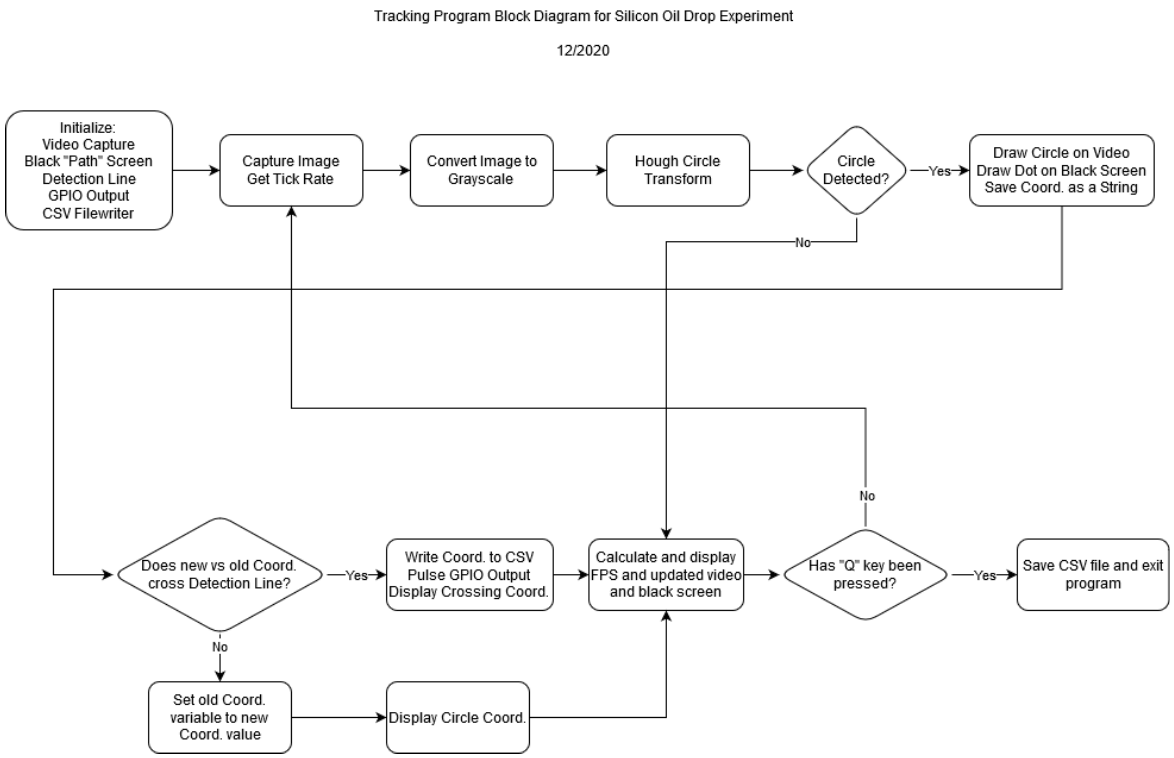


Figure 6: Raspberry Pi Program Block Diagram

C. Constraints

This project did not come with any constraints other than the ones laid out for each senior design project. These were to source any funds over $100, finish the project within two semesters, and present and report on the project at the end of the allotted time. The events of 2020 did have their effect on this project, restricting group access to materials and forcing much of the planning to go online. However, these constraints were overcome, and the project was completed.

V. Implementation and Validation

A. Performance Testing

The setup has the ability to run 2500 trials autonomously

This system has the ability to run this many trials. The two limiting factors would be the silicon oil level and the raspberry pi system memory. The overflow design of the 3D printed form allows the oil to runoff back into the reservoir, making the oil a fully closed loop system. The memory of the raspberry pi chosen is more than adequate of keeping track of the coordinates in a single CSV file that can be kept open without filling up all the system memory. With these factors in mind, this goal has been met.

Requires no replenishing of additional oil, i.e., system is all-in-one

As previously mentioned, the overflow design of the form allows the oil to be recirculated into the system. This performance metric has also been met.

System can track and record droplet position with statistical processing in real time

The raspberry pi program can track droplets throughout their journey as well as record the point at which they reach the end barrier. This is done in real time and is recorded in a data file for complex analysis later. Opensource software allows this process to be altered easily for different tracking metrics and tuning. This goal has been successfully met.

B. Performance Results

A series of small tests were completed to test the different system components. The largest of these consisted of 83 droplets through a double slit form. Although this isn't near enough data to determine if an interference pattern is present, it was enough to test the camera tracking functionality of the system. These results can be seen in the histogram in Figure 7. Note that the unusually high number of trials in the lower bins are most likely due to a drift across the oil surface causing them to pool towards that side.

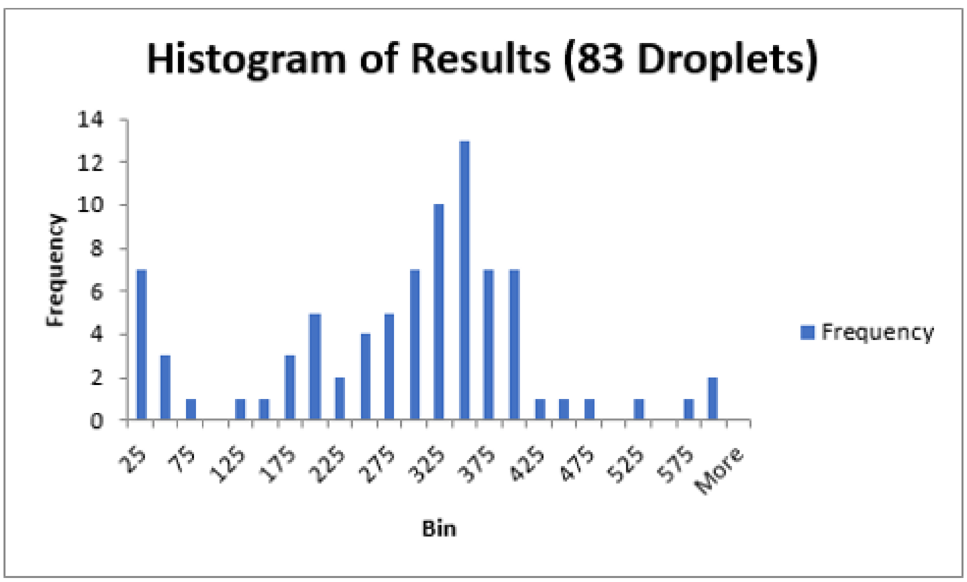


Figure 7: Histogram of Results from Largest Trial

C. Specification Met

The specifications mentioned earlier have been completed. A device to create oil droplets on demand was created. It is electrically driven and is automatically refilled by the reservoir and the 3D printed form. The electrical system designed serves as a means for either the user or the raspberry pi to signal the mechanical system. The droplet tracking software on the raspberry pi can track the droplet's movements, record data points, and store the data for later analysis.

VI. Recommendations

- The piezo electric device shows signs of being influenced by the electromagnetic shaker. Without appropriate dampening of the setup, the meniscus will oscillate and overflow outside of the nozzle tip, causing wetting and the end of the droplet generation cycle.
- Changing the fluid chamber and fluid reservoir in figure 4 to a clear acrylic would help in troubleshooting issues such as overflow state, and air bubble detection.
- The use of a USB camera system would allow for more options in terms of camera selection as the current system is fairly limited by the current camera being utilized.

VII. Conclusion

This project was successfully completed. All specifications were met as mentioned above. In all, some different approaches could have been made to complete this project. For example, the droplet generator could have been constructed out of a less expensive material, or the tracking software could have used a full-sized computer with a more powerful camera. However, given the constraints to both time and budget, the project was completed in an efficient manner and met all validation metrics.

VIII. References

[1] Harris, D. M., Liu, T., & Bush, J. W. M. (2015). *A low-cost, precise piezoelectric droplet-on-demand generator*. Cambridge, MA: Department of Mathematics, MIT

[2] Litwhiler, D. H. (2011). A Custom Vibration Test Fixture Using A Subwoofer. International Journal of Modern Engineering, 11(2), 68–78.

[3] Perrard, S., Labousse, M., Miskin, M., Fort, E., & Couder, Y. (2014). Self-organization into quantized eigenstates of a classical wave-driven particle. Nature Communications, 5(1). doi: 10.1038/ncomms4219

[4] Brady, R., & Anderson, R. (2014). Why bouncing droplets are a pretty good model of quantum mechanics. ArXiv, 1–20. Retrieved from https://arxiv.org/abs/1401.4356

[5] Bush, J. W. M. (2010). Quantum mechanics writ large. Proceedings of the National Academy of Sciences, 107(41), 17455–17456. doi: 10.1073/pnas.1012399107

[6] Couder, Y., Protière, S., Fort, E., & Boudaoud, A. (2005). Walking and orbiting droplets. Nature, 437(7056), 208–208. doi: 10.1038/437208a

[7] Fort, E., Eddi, A., Boudaoud, A., Moukhtar, J., & Couder, Y. (2010). Path-memory induced quantization of classical orbits. Proceedings of the National Academy of Sciences, 107(41), 17515–17520. doi: 10.1073/pnas.1007386107

Appendix A – Raspberry Pi Code

Raspberry Pi Code:

```
import cv2

import numpy as np

import csv

from datetime import datetime

import RPi.GPIO as GPIO

import time


frameWidth = 640    #640

frameHeight = 480   #480

cap = cv2.VideoCapture(0)

cap.set(3, frameWidth)

cap.set(4, frameHeight)


#detection line

detLine = 400

j=620

k=620

m=0
```

```
#setup blank img to track path
path_img = np.zeros((frameHeight+1, frameWidth, 3), np.uint8)
#cv2.line(path_img, (detLine,0),(detLine,480),(255,0,0),1)
cv2.line(path_img, (0,detLine),(640,detLine),(255,0,0),1)
#cv2.line(path_img, (220,0),(220,480),(255,0,0),1)
#cv2.line(path_img, (420,0),(420,480),(255,0,0),1)

#GPIO pin setup
pin_out = 21
GPIO.setwarnings(False)
GPIO.setmode(GPIO.BCM)
GPIO.setup(pin_out, GPIO.OUT)

#create and bein writtin to csv file
now = datetime.now()
filename = "tracking" + now.strftime("%m:%d:%Y, %H:%M:%S") + ".csv"
with open(filename, 'w') as csvfile:
    filewriter = csv.writer(csvfile, delimiter=',')#, quotechar='|', quoting=csv.QUOTE_MINIMAL)

#being while loop
    while True:
        #tick rate for fps calc
        timer = cv2.getTickCount()
```

```
        #capture frame as img
        success, img = cap.read()

        #convert to gray
        gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)

        #blur using 3 * 3 kernal
#        gray_blurred = cv2.blur(gray, (3,3))

    #hough transform on blurred image
        detected_circles = cv2.HoughCircles(gray, cv2.HOUGH_GRADIENT, 1, 1000, param1 =
50, param2 = 15, minRadius = 1, maxRadius = 5)
    #input frame
    #detection method
    #matrix resolution ratio           1
    #min dist between circles          20
    #var for ede detection             50
    #higher num => only better circles  30
```

```
#min circle radius                1

#max "    "                       40


#draw detected circles

    if detected_circles is not None:

    #convert circle parameters a, b, r to integers

        detected_circles = np.uint16(np.around(detected_circles))


        for pt in detected_circles[0, :]:

            a, b, r = pt[0], pt[1], pt[2]


        #draw circumference of circle

            cv2.circle(img, (a, b), r, (0, 255, 0), 2)

        #save y var

            j = b


        #draw a dot at the center of the circle

            path_img[int(b-1),int(a-1)] = [0,0,255]

        #save the location as a string

            coord = str(int(a)) + ", " + str(int(b))


#           #write the cordinate to the csv file

#               row = [str(int(a)),str(int(b))]
```

```
#                   filewriter.writerow(row)
            #compare
                if j>detLine and k<detLine:
                    filewriter.writerow([str(int(a))])
                    l=str(int(a))
                    cv2.putText(img,l,(75,100),cv2.FONT_HERSHEY_SIMPLEX,0.7,(0,0,255),2)
                    GPIO.output(pin_out, GPIO.HIGH) # Turn LED on
                    time.sleep(2)                   # Delay for 3 second
                    GPIO.output(pin_out, GPIO.LOW)  # Turn LED off
                    time.sleep(1)                   # Delay for 1 second

                k = j
                #jk = str(int(j)) + ", " + str(int(k))

            #display the cordinate on screen
                cv2.putText(img,coord,(75,75),cv2.FONT_HERSHEY_SIMPLEX,0.7,(0,0,255),2)
        else:
            m=m+1

    #calc fps
        fps = cv2.getTickFrequency()/(cv2.getTickCount()-timer)
    #show fps
        cv2.putText(img,str(int(fps)),(75,50),cv2.FONT_HERSHEY_SIMPLEX,0.7,(0,0,255),2)
```

```
    #display frame with circle
        cv2.imshow("Detected Circle", img)
    #display circle path
        cv2.imshow("Path", path_img)
#       cv2.imshow("gray", gray_blurred)

    #press q to stop, ELSE THE CSV WON'T SAVE
        if cv2.waitKey(1) & 0xFF == ord('q'):
            Break
```

Appendix B – Arduino Code

```
/*
 * Senior Design II, Piezo-electric silicone drop generator.
 *
 * Rowan M. Sheehan, Jacob P. Lutz, Myles R. McPhearson
 *
 * Code currently set for Arduino Nano and was made in Arduino 1.8.11
 *
 * The purpose of this software is to run the piezo-electric dropper using an SN754410 Quad-Half H Driver.
 * After recieving a 5V logic High Pulse from the rasberry PI, the droplet generator will create a single droplet
 * from the generator. Using the constants below, a user can define a pulsewidth time (microseconds) between
 * pulses.
 *
 * The code is reletively simple for running the device which may allow for future periferals on the device.
 */



//define the two direction logic pins
const int DIR_A = 4; //pin 5 on Arduino Nano board
 //pin 4 on Arduino Nano board
const int DIR_B = 3;
//pin 2 on Arduino Uno board (used to trigger camera if used)
const int cameraTrigger = 2;


//pulse width for which the piezoelectric actuator is activated in us
```

```
const int pulsewidth = 1000;

int triggerState = LOW;
bool dropletDelay;



void setup() {
  //set all output pins
  pinMode(DIR_A, OUTPUT);
  pinMode(DIR_B, OUTPUT);

  //set all input pins
  pinMode(cameraTrigger, INPUT);

  //initial state of piezo disk setup to apply +30V
  digitalWrite(DIR_A, HIGH);
  digitalWrite(DIR_B, LOW);
}

void loop() {
  //constantly look for state change between droplets
  triggerState = digitalRead(cameraTrigger);



  //if camera triggers for silicone droplet
  if (triggerState == HIGH) {
  //create droplet sequence
```

```
    //turn of interrupts for more accurate counting
    noInterrupts();

    //Apply a -30V across piezo electric disk causing drop actuation
    digitalWrite(DIR_A, LOW);
    digitalWrite(DIR_B, HIGH);

    //pulse width for which piezoelectic actuator is activated inmicroseconds
    delayMicroseconds(pulsewidth);

    //set voltage back to +30V, completing actuation
    digitalWrite(DIR_A, HIGH);
    digitalWrite(DIR_B, LOW);

    //re-enable intrrupts for other functions
    interrupts();

    //set flag for droplet sequence.
    dropletDelay = true;
  }
  //setup delay between droplets
  if (dropletDelay == true) {

    delay(5000);
    dropletDelay = false;
  }
}
```

Appendix C – Bill of Material

Bill of Materials				
Item #	Description	Supplier	Quantity	Cost
Mechanical Parts				
Custom Fabrication Pieces				
1	Aluminium Droplet Device	Springfield Machine	1	$ 750.00
2	3D Printed Nozzles (0.6mm to 1.4mm)	Proto Labs	6	$ 150.00
Framing Pieces				
3	2'x4'x1/2" Plywood Sheet	Menards	1	$ 15.00
4	3/4" wood screws	Menards	50	$ 4.00
5	Aluminium L bracket	Menards	2	$ 5.00
6	1/4-20 6' Threaded Rod	Menards	1	$ 5.00
Hardware				
1	Abrasion-Resistant PVC Plastic Clear Tubing for Food and Beverage, 1/8" ID, 3/16" OD, 25 ft. Length	McMaster-Carr	25ft	$ 5.50
2	Brass Threaded Fitting, Low-Pressure, 1/8" Tube ID x 10-32 UNF Male	McMaster-Carr	10	$ 14.00
3	18-8 Stainless Steel Socket Head Screw, 1/4"-20 Thread Size, 2-1/2" Long, Partially Threaded, Packs of 25	McMaster-Carr	25	$ 14.35
4	18-8 Stainless Steel Socket Head	McMaster-Carr	100	$ 6.30

	Screw, M3 x 0.5 mm Thread, 14 mm Long, Packs of 100			
5	18-8 Stainless Steel Socket Head Screw, 1/4"-20 Thread Size, 3-1/2" Long, Packs of 5	McMaster-Carr	5	$ 7.75
6	18-8 Stainless Steel Socket Head Screw, 1/4"-20 Thread Size, 1/2" Long, Packs of 50	McMaster-Carr	50	$ 9.60
7	Light Duty Leveling Mount, 1" Long 1/4"-20 Threaded Stud, Packs of 4	McMaster-Carr	4	$ 1.90
8	18-8 Stainless Steel Hex Nut, 1/4"-20 Thread Size, Packs of 100	McMaster-Carr	100	$ 4.25
9	18-8 Stainless Steel Washer for 1/4" Screw Size, 0.281" ID, 0.625" OD, Packs of 100	McMaster-Carr	100	$ 3.50
10	Dowel Pin, 316 Stainless Steel, 1/8" Diameter, 1/2" Long, Packs of 10	McMaster-Carr	10	$ 7.25
11	Oil-Resistant Buna-N O-Ring, 1/16 Fractional Width, Dash Number 026, Packs of 100	McMaster-Carr	100	$ 8.50
12	Oil-Resistant Buna-N O-Ring, 1/16 Fractional Width, Dash Number 014, Packs of 100	McMaster-Carr	100	$ 4.65

13	Rubber Washer for 1/4 screw size	Menards	4	$ 5.00
14	1/4-20 Zinc plated wing nut	Menards	4	$ 3.38
Chemicals				
Fluids				
1	20 cSt Silicone Oil	ESCO	1 qt	$ 95.00
Pump				
2	INTLLAB 12V Parastaltic Pump	Amazon	1	$ 10.00
Reservoir				
3	3/4" PVC Tubing	Menards	6ft	$ 5.40
4	3/4" PVC End Cap	Menards	1	$ 2.40
5	3/4" PVC threaded end cap	Menards	1	$ 3.20
6	PVC Threaded Cap	Menards	1	$ 1.75
Electronics				
Computers/Micro controllers				
1	Arduino Nano, 3rd party	Elegoo	1	$ 15.50
2	Rasberry Pi 4b	Rasberry Pi Foundation	1	$ 35.00
3	Rasberry Pi 5 Megapixel camera add-on	Digikey	1	$ 25.00
Electromagnetic Shaker				
1	Mean Well NES-350-12 DC Power Supply	Digikey	1	$ 65.00
2	Pioneer SW2002D shallow 8" subwoofer	Pioneer	1	$ 140.00
3	Planet Audio AC600-2 600W Audio Amplifier	Planet Audio	1	$ 70.00
Electronic Components				
1	Hilitchi Electrolytic Capacitor Kit	Amazon	1	$ 10.75
2	1k OHM 1/4W 5% Yageo Axial Resisotr	Digikey	3	$ 0.30

3	Elegoo Breadboard	Amazon	1	$ 5.50
4	SN754410 Half H Bridge	Digikey	1	2.51
Misc				
1	USB C Cord	Amazon	1	$ 3.00
2	USB A Mini	Amazon	1	$ 3.00
3	22 GA wire Kit	Amazon	1	$ 10.00
Notes				
	Provided by Project Advisor			
	Provided by 3rd party donations / product samples (free)			
		Total		
		In-Kind		$ 1,316.55
		Actual Expenses		$ 188.94
		Project Total		**$ 1,528.24**