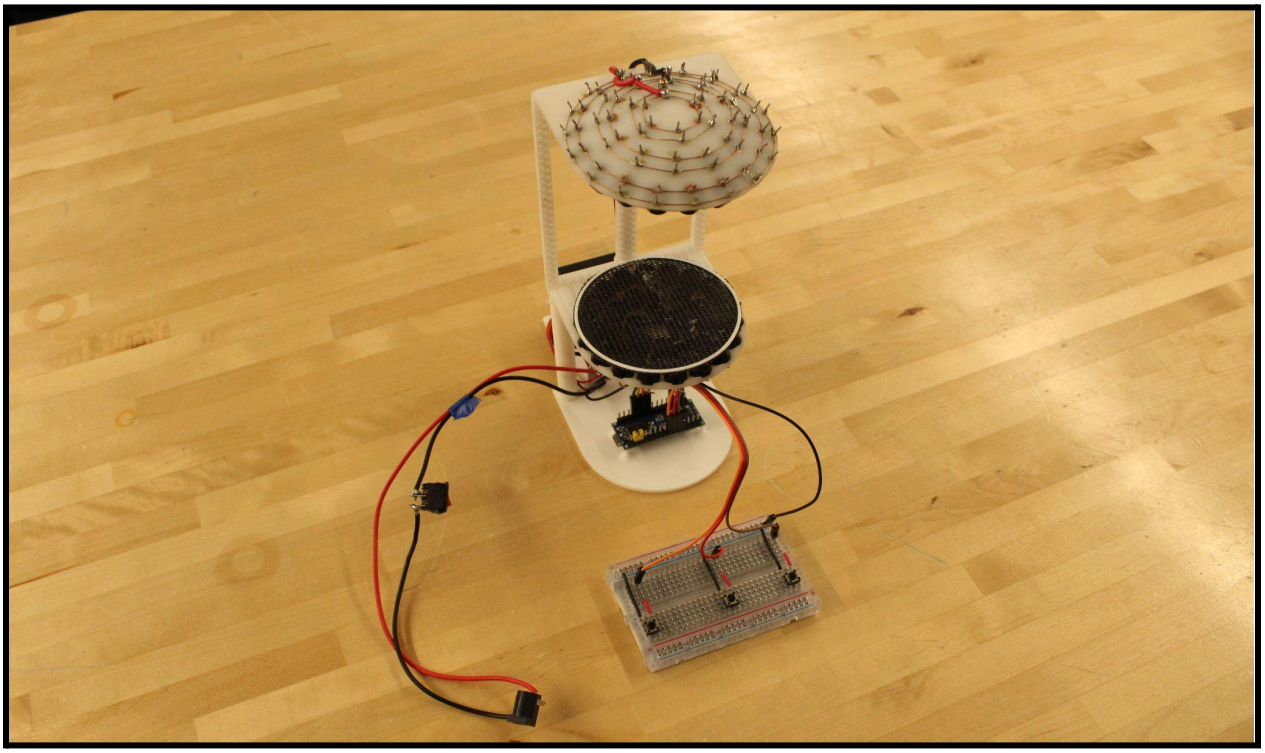
Acoustic Levitation

https://www.che.utah.edu/outreach/



Summary:

While at first glance it doesn't seem like sound waves would affect our world in a physical way other than to carry sound, it is important to understand what they are. In this module, you will build an acoustic levitator with an arduino kit in order to further study how sound waves work and can be utilized to interact with the world.

General Information:

Main Curriculum Tie:	Sound and Pressure Waves
Additional Curriculum Ties:	Electronics, Mass Transfer
Career Connections:	Many engineers use sound waves to speed up chemical processes and improve industrial processes such as oil extraction.
Typical Time Frame:	30 min instructions 60 min to solder 15 min to wire 10 min to assemble 30 min to test
Typical Group Size	Groups of 2 to 3
Student Prior Knowledge:	Some experience with electronics and soldering

Essential Questions:

- What is sound?
- What exactly is a wave?
- How do waves interact with each other?
- How can we apply sound waves to the physical world?
- What is important to consider when designing a circuit?
- Can this novel levitator be used for scientific purposes?
- How can we analyze the acoustic levitation of liquid droplets?

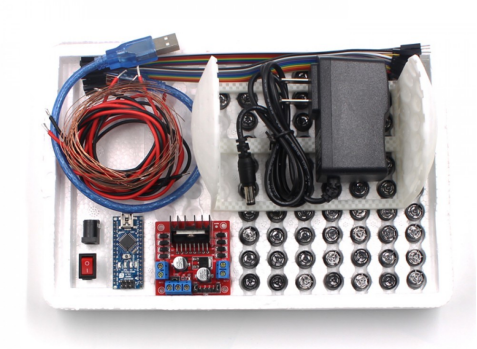
Materials:

Tools Used:

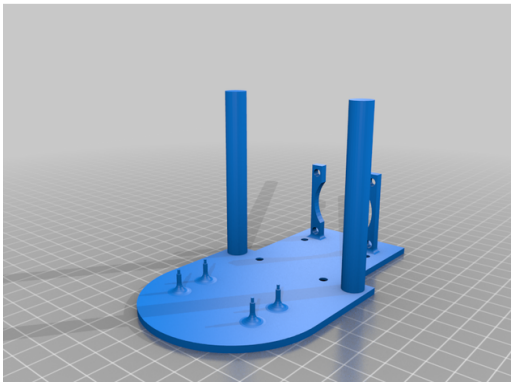
- 3D printer - Printing the electronics and levitator mount design.
- Computer - Programming the provided arduino.
- Soldering iron, tin, flux, and hot glue gun - Assembly of the levitator array.
- Oscilloscope - Checking polarity of transducers.
- Wire Stripper - Tidying wire provided to proper lengths.
- Phillips Head Screwdriver - Mounting electronics to the housing.
- Needle Nose Pliers - Assembly work.

Components:

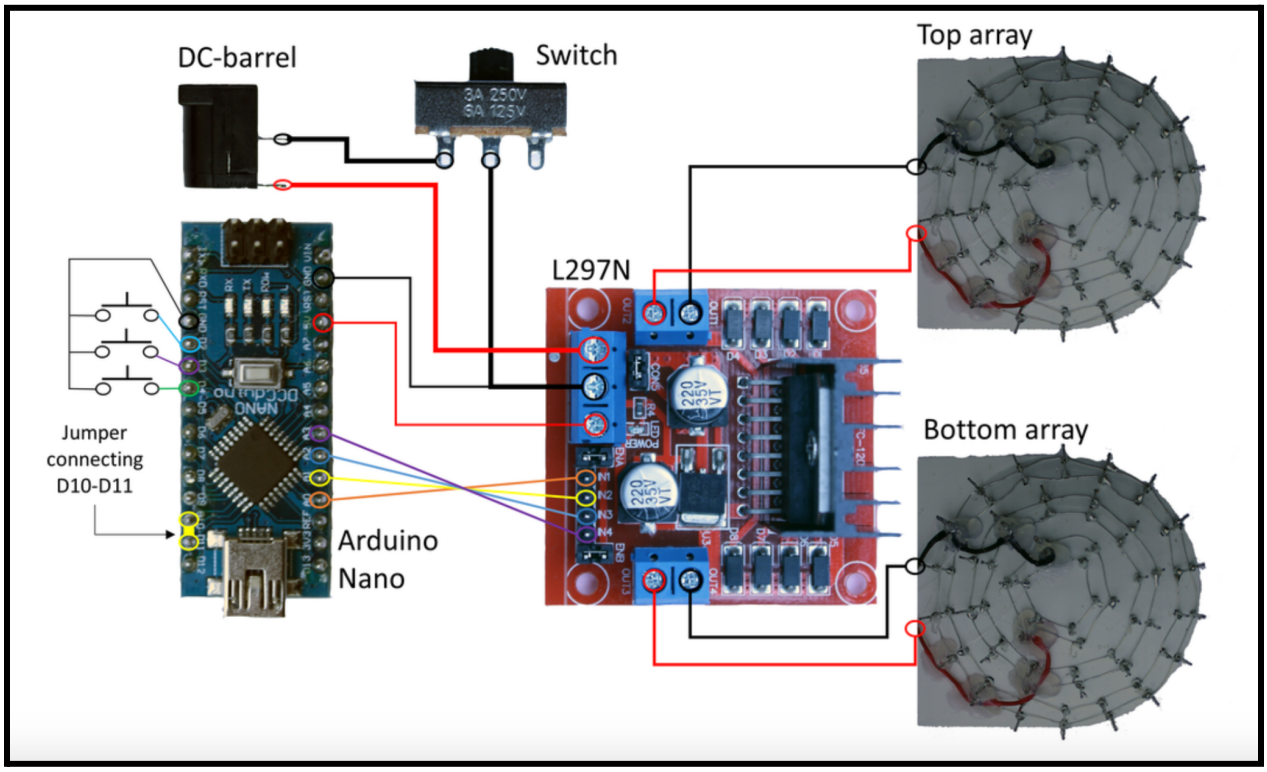
- A TinyLev kit. This kit can be purchased as a whole from makerfabs, robotshop, or tindie. If you would like to purchase parts individually, a more exhaustive list with links can be found in this instructables which provides the instructions we followed for assembly.
    - 72x 10mm 40kHz transducers. Manorshi provides MSO-P1040H07T at a very good price, minimum order is 500 but they will ship with less at a higher price. Also Ningbo has good ones FBULS1007P-T
    - 1x 3D-printed TinyLev support.
    - 1x Arduino Nano
    - 1x L298N Dual Motor Drive Board
    - 1x 130x90mm sheet (wood or acrylic) for the base of the driver board.
    - 1x power switch
    - DC adaptor variable between 7V and 12V
    - DC female connector
    - Jumper wires
    - 12AWG black and red wire
    - 24AWG black and red wire
    - 24AWG exposed wire
    - Some Expanded Polystyrene beads to levitate (between 1mm and 3mm diameter)
    - An acoustically transparent material: A metallic grid, very thin fabric or teabag paper.



- Our 3D-printed mounting platform to hold both the electronics and mount the levitator onto. This was designed to be minimal and we encourage customization to practice 3D modeling and design principles. Our model can be found here.



Electronics:



- The above electronics circuit diagrams the setup for the levitator. Once again, more complete instructions are provided at this instructables. On this arduino, the following code must be uploaded using a computer:

```
#include <avr/sleep.h>
#include <avr/power.h>

#define N_PORTS 1
#define N_DIVS 24

#define WAIT_LOT(a) __asm__ __volatile__ ("nop"); __asm__ __volatile__ ("nop"); __asm__ __volatile__
("nop");  __asm__ __volatile__ ("nop"); __asm__ __volatile__ ("nop");__asm__ __volatile__ ("nop");
__asm__ __volatile__ ("nop");__asm__ __volatile__ ("nop"); __asm__ __volatile__ ("nop");  __asm__
__volatile__ ("nop"); __asm__ __volatile__ ("nop");__asm__ __volatile__ ("nop");  __asm__ __volatile__
("nop");  __asm__ __volatile__ ("nop")
#define WAIT_MID(a) __asm__ __volatile__ ("nop"); __asm__ __volatile__ ("nop"); __asm__ __volatile__
("nop");  __asm__ __volatile__ ("nop"); __asm__ __volatile__ ("nop");__asm__ __volatile__ ("nop");
__asm__ __volatile__ ("nop");__asm__ __volatile__ ("nop"); __asm__ __volatile__ ("nop"); __asm__
__volatile__ ("nop"); __asm__ __volatile__ ("nop"); __asm__ __volatile__ ("nop"); __asm__ __volatile__
("nop")
#define WAIT_LIT(a) __asm__ __volatile__ ("nop"); __asm__ __volatile__ ("nop"); __asm__ __volatile__
("nop");  __asm__ __volatile__ ("nop"); __asm__ __volatile__ ("nop");  __asm__ __volatile__ ("nop");
__asm__ __volatile__ ("nop");  __asm__ __volatile__ ("nop");  __asm__ __volatile__ ("nop")


#define OUTPUT_WAVE(pointer, d)  PORTC = pointer[d*N_PORTS + 0]
```

```
#define N_BUTTONS 6
//half a second
#define STEP_SIZE 1
#define BUTTON_SENS 2500
#define N_FRAMES 24

static byte frame = 0;
static byte animation[N_FRAMES][N_DIVS] =
{{0x5,0x5,0x5,0x5,0x5,0x5,0x5,0x5,0x5,0x5,0x5,0x5,0xa,0xa,0xa,0xa,0xa,0xa,0xa,0xa,0xa,0xa,0xa,0xa},
{0x9,0x5,0x5,0x5,0x5,0x5,0x5,0x5,0x5,0x5,0x5,0x6,0xa,0xa,0xa,0xa,0xa,0xa,0xa,0xa,0xa,0xa,0xa,0xa},
{0x9,0x9,0x5,0x5,0x5,0x5,0x5,0x5,0x5,0x5,0x5,0x6,0x6,0xa,0xa,0xa,0xa,0xa,0xa,0xa,0xa,0xa,0xa,0xa},
{0x9,0x9,0x9,0x5,0x5,0x5,0x5,0x5,0x5,0x5,0x5,0x6,0x6,0x6,0xa,0xa,0xa,0xa,0xa,0xa,0xa,0xa,0xa,0xa},
{0x9,0x9,0x9,0x9,0x5,0x5,0x5,0x5,0x5,0x5,0x5,0x6,0x6,0x6,0x6,0xa,0xa,0xa,0xa,0xa,0xa,0xa,0xa,0xa},
{0x9,0x9,0x9,0x9,0x9,0x5,0x5,0x5,0x5,0x5,0x5,0x6,0x6,0x6,0x6,0x6,0xa,0xa,0xa,0xa,0xa,0xa,0xa,0xa},
{0x9,0x9,0x9,0x9,0x9,0x9,0x5,0x5,0x5,0x5,0x5,0x6,0x6,0x6,0x6,0x6,0x6,0xa,0xa,0xa,0xa,0xa,0xa,0xa},
{0x9,0x9,0x9,0x9,0x9,0x9,0x9,0x5,0x5,0x5,0x5,0x6,0x6,0x6,0x6,0x6,0x6,0x6,0xa,0xa,0xa,0xa,0xa,0xa},
{0x9,0x9,0x9,0x9,0x9,0x9,0x9,0x9,0x5,0x5,0x5,0x6,0x6,0x6,0x6,0x6,0x6,0x6,0x6,0xa,0xa,0xa,0xa,0xa},
{0x9,0x9,0x9,0x9,0x9,0x9,0x9,0x9,0x9,0x5,0x5,0x6,0x6,0x6,0x6,0x6,0x6,0x6,0x6,0x6,0xa,0xa,0xa,0xa},
{0x9,0x9,0x9,0x9,0x9,0x9,0x9,0x9,0x9,0x9,0x5,0x5,0x6,0x6,0x6,0x6,0x6,0x6,0x6,0x6,0x6,0xa,0xa,0xa},
{0x9,0x9,0x9,0x9,0x9,0x9,0x9,0x9,0x9,0x9,0x9,0x5,0x5,0x6,0x6,0x6,0x6,0x6,0x6,0x6,0x6,0x6,0xa,0xa},
{0x9,0x9,0x9,0x9,0x9,0x9,0x9,0x9,0x9,0x9,0x9,0x9,0x5,0x5,0x6,0x6,0x6,0x6,0x6,0x6,0x6,0x6,0x6,0xa},
{0x9,0x9,0x9,0x9,0x9,0x9,0x9,0x9,0x9,0x9,0x9,0x9,0x9,0x6,0x6,0x6,0x6,0x6,0x6,0x6,0x6,0x6,0x6,0x6},
{0x5,0x9,0x9,0x9,0x9,0x9,0x9,0x9,0x9,0x9,0x9,0x9,0xa,0x6,0x6,0x6,0x6,0x6,0x6,0x6,0x6,0x6,0x6,0x6},
{0x5,0x5,0x9,0x9,0x9,0x9,0x9,0x9,0x9,0x9,0x9,0x9,0xa,0xa,0x6,0x6,0x6,0x6,0x6,0x6,0x6,0x6,0x6,0x6},
{0x5,0x5,0x5,0x9,0x9,0x9,0x9,0x9,0x9,0x9,0x9,0x9,0xa,0xa,0xa,0x6,0x6,0x6,0x6,0x6,0x6,0x6,0x6,0x6},
{0x5,0x5,0x5,0x5,0x9,0x9,0x9,0x9,0x9,0x9,0x9,0x9,0xa,0xa,0xa,0xa,0x6,0x6,0x6,0x6,0x6,0x6,0x6,0x6},
{0x5,0x5,0x5,0x5,0x5,0x9,0x9,0x9,0x9,0x9,0x9,0x9,0xa,0xa,0xa,0xa,0xa,0x6,0x6,0x6,0x6,0x6,0x6,0x6},
{0x5,0x5,0x5,0x5,0x5,0x5,0x9,0x9,0x9,0x9,0x9,0x9,0xa,0xa,0xa,0xa,0xa,0xa,0x6,0x6,0x6,0x6,0x6,0x6},
{0x5,0x5,0x5,0x5,0x5,0x5,0x5,0x9,0x9,0x9,0x9,0x9,0xa,0xa,0xa,0xa,0xa,0xa,0xa,0x6,0x6,0x6,0x6,0x6},
{0x5,0x5,0x5,0x5,0x5,0x5,0x5,0x5,0x9,0x9,0x9,0x9,0xa,0xa,0xa,0xa,0xa,0xa,0xa,0xa,0x6,0x6,0x6,0x6},
{0x5,0x5,0x5,0x5,0x5,0x5,0x5,0x5,0x5,0x9,0x9,0x9,0xa,0xa,0xa,0xa,0xa,0xa,0xa,0xa,0xa,0x6,0x6,0x6},
{0x5,0x5,0x5,0x5,0x5,0x5,0x5,0x5,0x5,0x5,0x9,0x9,0xa,0xa,0xa,0xa,0xa,0xa,0xa,0xa,0xa,0xa,0x6,0x6},
{0x5,0x5,0x5,0x5,0x5,0x5,0x5,0x5,0x5,0x5,0x5,0x9,0xa,0xa,0xa,0xa,0xa,0xa,0xa,0xa,0xa,0xa,0xa,0x6}};


void setup()
{

/*
 for (int i = 0; i < (N_PORTS*N_DIVS); ++i){
        animation[frame][i] =  0;
  }

  for (int i = 0; i < (N_PORTS*N_DIVS/2); ++i){
        animation[frame][i] = 0b11111111;
  }

  for(int i = 0; i < N_DIVS; ++i){
        if (i % 2 == 0){
        animation[frame][i * N_PORTS] |= 0b00000001;
        }else{
        animation[frame][i * N_PORTS] &= 0b11111110;
        }
  }
*/
   DDRC = 0b00001111; //A0 to A3 are the signal outputs
   PORTC = 0b00000000;

   pinMode(10, OUTPUT); //pin 10 (B2) will generate a 40kHz signal to sync
   pinMode(11, INPUT_PULLUP); //pin 11 (B3) is the sync in
```

```
   //please connect pin 10 to pin 11

   for (int i = 2; i < 8; ++i){ //pin 2 to 7 (D2 to D7) are inputs for the buttons
        pinMode(i, INPUT_PULLUP);
   }

  // generate a sync signal of 40khz in pin 10
  noInterrupts();       // disable all interrupts
  TCCR1A = bit (WGM10) | bit (WGM11) | bit (COM1B1); // fast PWM, clear OC1B on compare
  TCCR1B = bit (WGM12) | bit (WGM13) | bit (CS10);   // fast PWM, no prescaler
  OCR1A =  (F_CPU / 40000L) - 1;
  OCR1B = (F_CPU / 40000L) / 2;
  interrupts();         // enable all interrupts

  // disable everything that we do not need
  ADCSRA = 0;  // ADC
  power_adc_disable ();
  power_spi_disable();
  power_twi_disable();
  power_timer0_disable();
  //power_usart0_disable();
  Serial.begin(115200);

 byte* emittingPointer = &animation[frame][0];
 byte buttonsPort = 0;

 bool anyButtonPressed;
 bool buttonPressed[N_BUTTONS];
 short buttonCounter = 0;

  LOOP:
        while(PINB & 0b00001000); //wait for pin 11 (B3) to go low

        OUTPUT_WAVE(emittingPointer, 0); buttonsPort = PIND; WAIT_LIT();
        OUTPUT_WAVE(emittingPointer, 1); anyButtonPressed = (buttonsPort & 0b11111100) != 0b11111100;
WAIT_MID();
        OUTPUT_WAVE(emittingPointer, 2); buttonPressed[0] = buttonsPort & 0b00000100; WAIT_MID();
        OUTPUT_WAVE(emittingPointer, 3); buttonPressed[1] = buttonsPort & 0b00001000; WAIT_MID();
        OUTPUT_WAVE(emittingPointer, 4); buttonPressed[2] = buttonsPort & 0b00010000; WAIT_MID();
        OUTPUT_WAVE(emittingPointer, 5); buttonPressed[3] = buttonsPort & 0b00100000; WAIT_MID();
        OUTPUT_WAVE(emittingPointer, 6); buttonPressed[4] = buttonsPort & 0b01000000; WAIT_MID();
        OUTPUT_WAVE(emittingPointer, 7); buttonPressed[5] = buttonsPort & 0b10000000; WAIT_MID();
        OUTPUT_WAVE(emittingPointer, 8); WAIT_LOT();
        OUTPUT_WAVE(emittingPointer, 9); WAIT_LOT();
        OUTPUT_WAVE(emittingPointer, 10); WAIT_LOT();
        OUTPUT_WAVE(emittingPointer, 11); WAIT_LOT();
        OUTPUT_WAVE(emittingPointer, 12); WAIT_LOT();
        OUTPUT_WAVE(emittingPointer, 13); WAIT_LOT();
        OUTPUT_WAVE(emittingPointer, 14); WAIT_LOT();
        OUTPUT_WAVE(emittingPointer, 15); WAIT_LOT();
        OUTPUT_WAVE(emittingPointer, 16); WAIT_LOT();
        OUTPUT_WAVE(emittingPointer, 17); WAIT_LOT();
        OUTPUT_WAVE(emittingPointer, 18); WAIT_LOT();
        OUTPUT_WAVE(emittingPointer, 19); WAIT_LOT();
        OUTPUT_WAVE(emittingPointer, 20); WAIT_LOT();
        OUTPUT_WAVE(emittingPointer, 21); WAIT_LOT();
        OUTPUT_WAVE(emittingPointer, 22); WAIT_LOT();
        OUTPUT_WAVE(emittingPointer, 23);
```

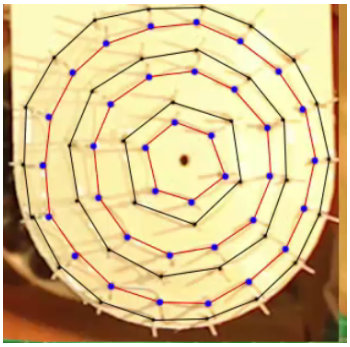
```
        if( anyButtonPressed ){
        ++buttonCounter;
        if (buttonCounter > BUTTON_SENS){
        buttonCounter = 0;

        if (! buttonPressed[0] ) {
        if( frame < STEP_SIZE ) {
        frame = N_FRAMES-1;
        }else{
        frame-=STEP_SIZE;
        }
        }
        else if (! buttonPressed[1] ) {
        if ( frame >= N_FRAMES-STEP_SIZE ) {
        frame = 0;
        }else {
        frame+=STEP_SIZE;
        }
        }else if (! buttonPressed[2] ) {
        frame = 0;
        }
        emittingPointer = & animation[frame][0];
        }
        }else {
        buttonCounter = 0;
        }

  goto LOOP;

}

void loop(){}
```

Instructional Procedures:

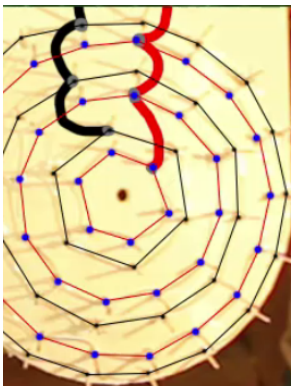
The following procedure will be based around the concept of creating an acoustic levitator, and using it as a teaching device to instruct a class on soundwaves with additional information targeted at mass transfer analysis. This can be adapted for groups to assemble their own levitators, but in general it seems that this would be too time consuming for most classroom settings to do. For the assembly process, supplemental instructions can be found here.

Assembly:

1. First we will assemble the acoustic levitator. Begin by purchasing or gathering all required components to assemble the levitator.
2. Now, we need to measure and mark the polarity of each transducer. While they are labeled from the factory, this has been found to be an inconsistent metric to rely on. To do this, connect a multimeter to each transducer leg. Watch as you connect the second leg to the multimeter, if the voltage reads negative right away, mark the side connected to the multimeter ground, and vice versa.



3. Now place a drop of hot glue underneath each transducer as you place it into each slot of the 3D printed transducer array mount. Make sure to have all the positive legs on the inside of each socket so that we can wire them easily.
4. Once all the transducer's have been placed into the array, we can begin to wire them together. For this all positives should be wired together and all negatives likewise. Essentially you will wrap the copper wires in a circle around each set of legs as is seen in the image on the right.



5. The next step is to go through and place solder onto each connection between a pin and a copper wire. This is necessary to ensure connectivity, and to make sure that the device will stay together.
6. Now using the wires provided in the kit, bridge every other circle of wires together starting with positive on the innermost circle. These wires will go off to be connected to the stepper board that comes in the kit.
7. Now you should take the arduino, with the code provided above uploaded to it, and connect its pins to those shown in the wiring diagram.
8. Additionally you will now solder together the DC barrel port, and the switch as is shown in the diagram as well.
9. Now the arduino and the power should both be connected to the driver board, and we can next test to ensure the driver is working. For this an oscilloscope will allow the ports that are shown to connect to the arrays to be probed and the resulting wave should be observed to be 40 kHz with an amplitude 2X the power being delivered to it.
10. Using the multimeter make sure to test the arrays by connecting the black and red wires you've soldered to them to the multimeter and making sure no short is measured.
11. FInally, these arrays can be wired into their respective slots on the driver board as one again can be seen in the circuitry diagram.
12. Now, with the circuit plugged in and turned on, we can test to make sure each transducer is operating.To do this a transducer can be placed on either probe of an oscilloscope and each can be touched to a corresponding transducer in the array. Comparing the two frequencies measured will ensure that all are emitting at the same frequency and functioning properly. If one isn't working it will need to be replaced to ensure the integrity of the net pressure field generated by the array.
13. Finally, all boards can be placed into the mounting model provided, and the wires can be cut down to length to fit into the finished acoustic levitator footprint. The levitator glues to the tall large standoffs, the arduino sits on the rounded short standoffs, and the driver board can be mounted on standard standoffs with M6 hardware screws that fit the countersunk holes on the bottom. This should complete the levitator.
    a. For our design we chose to add an optional fan so if chosen the extra horizontal screw holes on the mounting board can be used for that, although the addition is not necessary unless planning to have the levitator running for excessively long periods of time. If a fan is added, we recommend additionally blocking the backside of the levitator with something such as paper as airstreams coming off

the fan caused serious interference with stabilizing objects in our levitator when we introduced a fan.

b. We also chose to add a mesh over the bottom array of transducers to keep liquid out as we were experimenting. This is recommended to place fine mesh which water droplets won't go through if you plan on levitating a liquid inside of yours as well such as we describe in the mass transfer section later on.

In class:

1. Give the background information on sound waves and how the acoustic levitator works.
    a. This slideshow may be useful:
2. Show the students how the buttons on the acoustic levitator work, so that they can move the objects vertically.
    a. The buttons are used to shift the frequencies at which the transducer arrays are generating. In doing this, the node points also move up and down. As such, the right button shifts the nodes up, the middle down, and the left resets the transducers to their strongest frequencies.
3. Have the kids experiment by placing different types of objects in the acoustic levitator (for easy examples: styrofoam pellets, water, juice).
    a. Selection of the levitation medium can allow for some interesting results. For example, some things we tried levitating included water, isopropanol, methanol, and even things like dry ice. The dry ice for example tended to form small clouds of water particles that would form around the nodes and allow the pressure field to be visualized well. The isopropanol test we did allowed for the nodes to be spotted as its low surface tension caused it to break apart and in doing so small droplets found their way to the nodes.
        i. Using other fluids or items can allow for students to try to use intuition based around the core principles to infer behavior. One example of a medium that could have interesting results while being safe for a classroom may be a non-newtonian fluid (cornstarch and water i.e.). We did not have the time to experiment with this ourselves, but suspect that it may feature interesting behavior when levitated.
4. Ask the kids to give potential explanations as to how acoustic levitation could be possible based on their new knowledge and experiences.
    a. One of the most important skills a prospective STEM student can learn is to take their scientific findings and use them to craft world views and theorems. This is the perfect opportunity to hone this skill.

Background for Teachers:

The question of if acoustic levitation is possible has been rounding the scientific community for more than a century. Acoustic levitation is a method to suspend matter in the air against the force of gravity using ultrasound waves that radiate with a higher frequency than is perceptible to the human ear. For a long time, acoustic levitation was limited to small objects but nowadays it can be expanded to objects bigger than the waves, and it can also go further away than suspending to rotating and translating solid objects in more than one plane. Nevertheless,

the study of acoustically levitated fluids has been getting a lot of visibility in different engineering fields.

To understand the behavior of liquids in acoustic levitation is vital to understand the apparatus that allowed this process to occur. The most known acoustic levitator arrangement at the time is the single-axis arrangement; among this arrangement there are two main prototypes. The first one is a resonant levitator (Figure a.) that consists of one set of transducers with a reflector positioned, at a certain distance and angle, meant to act as a resonant cavity. The second prototype is a nonresonant levitator (Figure b.) that consists of two opposite and separate sets of emitters. Both of these prototypes have differences as far as how efficient they are under unsteady scenarios. Nevertheless, resonant and nonresonant levitators are guided by sinusoidal excitation signals, meaning that this force has one single excitation at a time and the excitation swings from one frequency to another one, giving the chance to the structure to interact through a harmonic vibration.

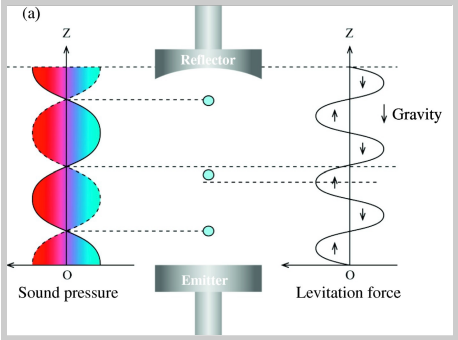


Figure a.

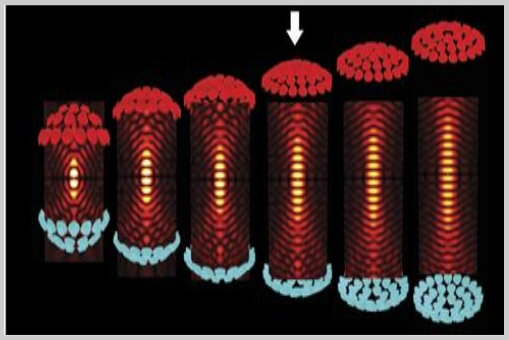


Figure b.

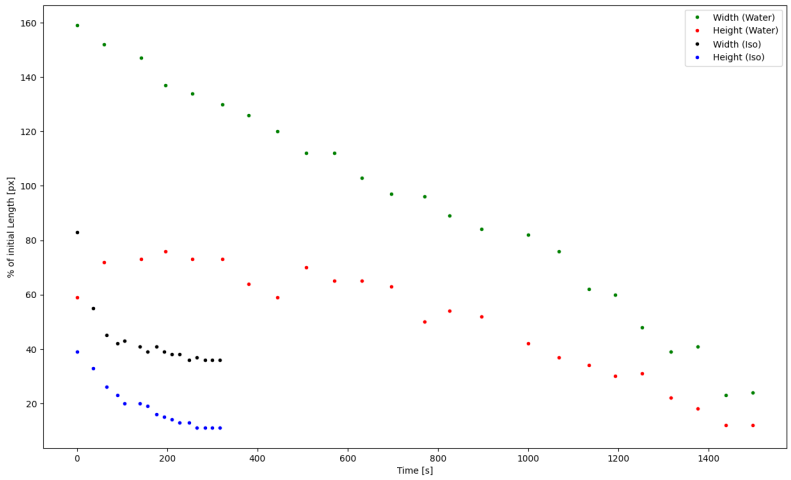
Because of its lack of solid surfaces, the levitation of liquid droplets has made huge contributions to the study of fluids in different areas such as physics, nucleations, biological and chemical processes. When levitation occurs the gravity of the droplet is balanced by the radiation force exerted on its surface, resulting in the surface tension added to the internal pressure of the droplet, leading to radial oscillation as a consequence of the confrontation between both pressures. Nevertheless, acoustic levitation could potentially help one understand the instability behavior field because upon being suspended, the droplet displays many different types of instability behaviors such as atomization and buckling stability. Considering that mechanical contact is completely obviate in the levitation process it is an ideal way to also examine phase transition. Levitated droplets could cover a wide range of substances with different chemical properties and that will help us study various physical processes such as evaporation and crystallization.

Mass Transfer Addendum:

(Code that was used for image analysis and for modeling purposes can be found here.)

Acoustic levitation provides a great demonstration to aid in the understanding of mass transfer. While our team was unable to accomplish a full data analysis portion project, we have been able to collect data and note some trends, as well as finish a rudimentary analysis. For more information on how to take the data and fully analyze the mass transfer portion, these two sources could be very useful: Source 1 and Source 2. Our data analysis will follow.

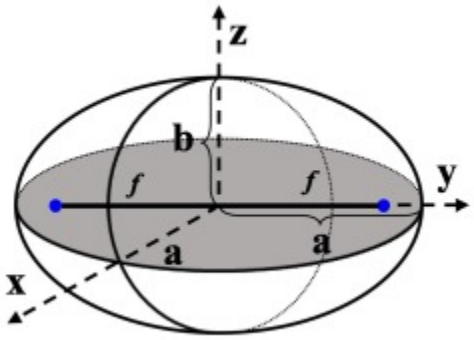
First we used image analysis to collect the width and height of both water and isopropanol beads levitated over time. The data for each is shown in this graph:



We then utilized this data to find the area and volume of each sphere over time using the equations determined in Source 2:

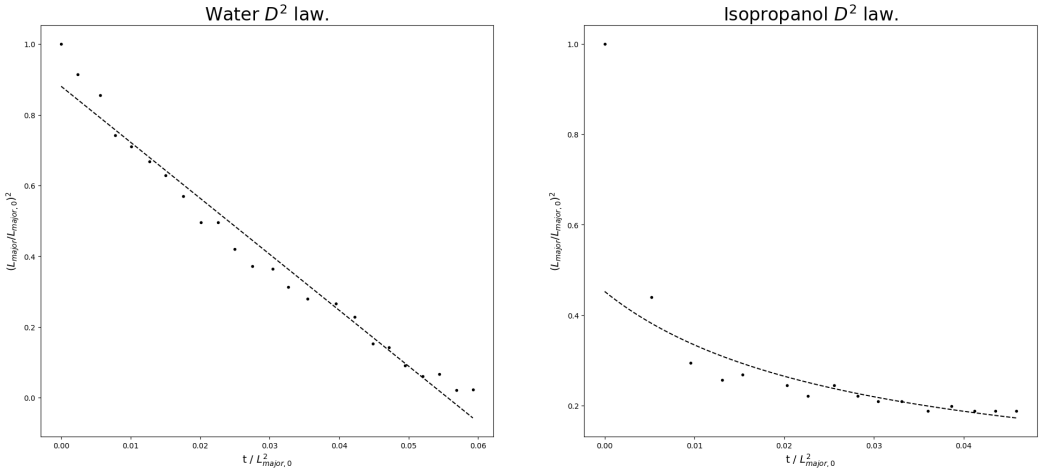
$$A_{os} = 2\pi a^2 \left(1 + \frac{1-e^2}{2e} \ln\left(\frac{1+e}{1-e}\right)\right)$$

$$V_{os} = \frac{1}{6}\pi \cdot A_R \cdot L_{major}^3$$

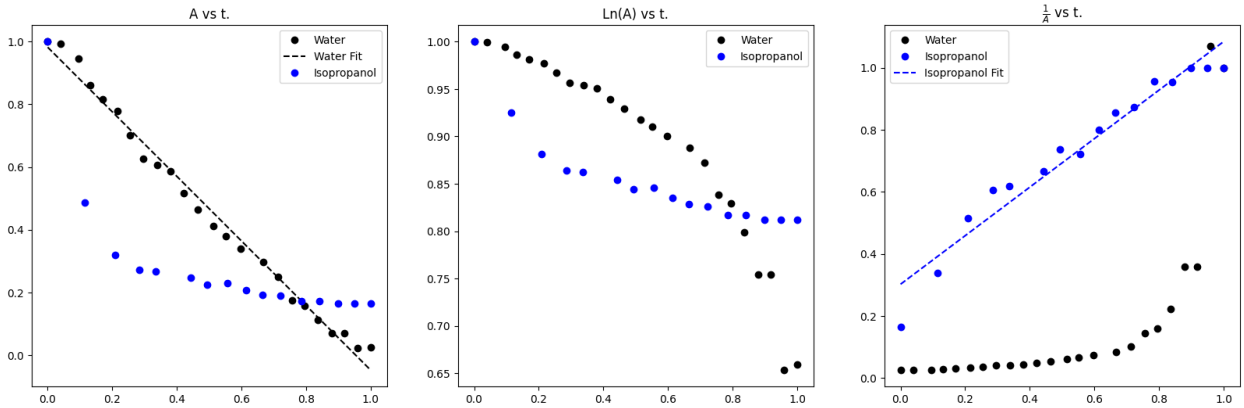


Where *e* is the oblate spheroid eccentricity. A_R is the aspect ratio of the spheroid, $A_R = b/a$. For an oblate spheroid, with a and b being half the width and height from the horizontally captured image respectively.

Next, we plotted the squared major axis of the oblate droplet over the initial major axis versus dimensionless time.



This graph is essential, as it should be linear. For our water data this is clearly true, and thus can be modeled using extensive calculations. For our isopropanol data, however, there seems to be some anomaly causing it to act like a second order set of data when it should be zeroth order. To make this more clear we then added a line of best fit to the data and determined that it did follow both those previously mentioned fits using the below figures.



To understand better why the data is expected to appear linearly would require a deep dive into [film-theory of mass transfer], but to sum it simply: the droplet evaporation is expected to follow what is colloquially known as the d^2 law. This law comes from the mass transfer equation that governs droplet evaporation.

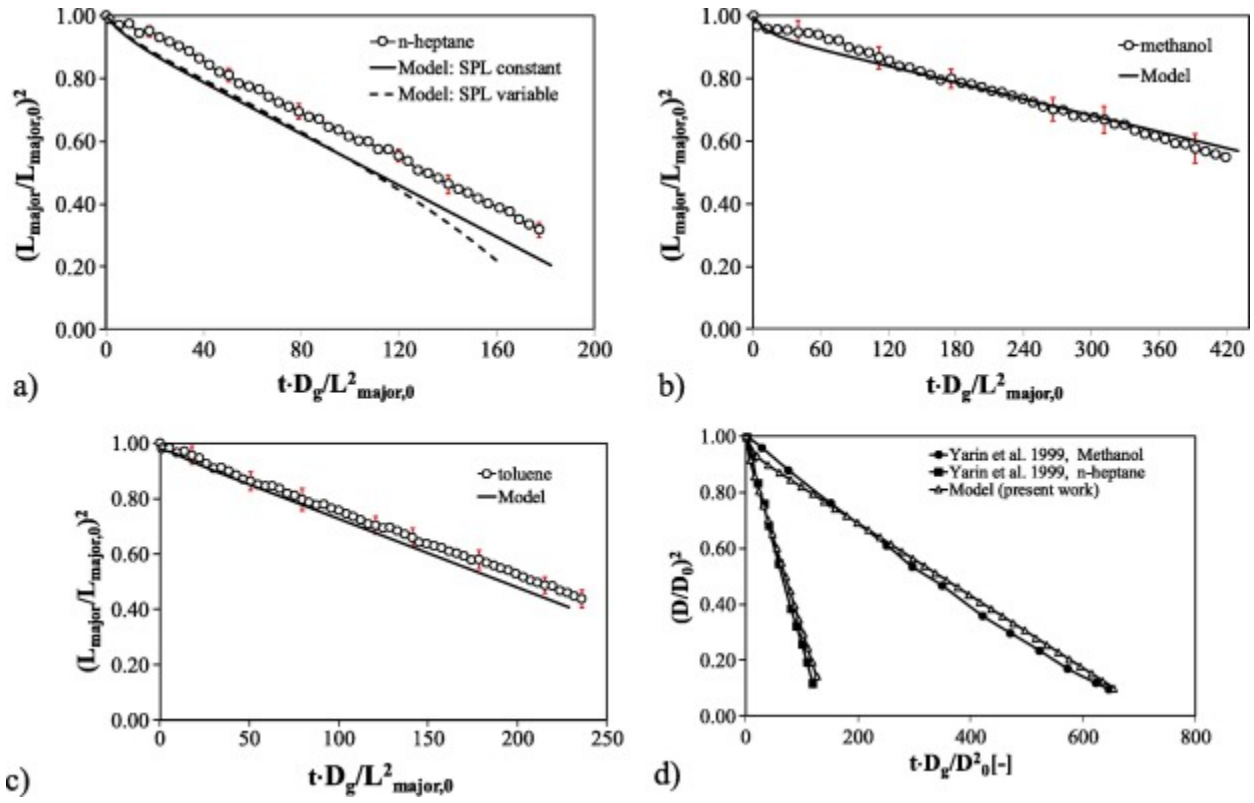
$$\dot{m}(\theta)_{eva} = \rho_g \mathfrak{D}_g \frac{A_{os}}{L_c} \cdot Sh(\theta) \cdot \frac{y_{vap,s}(\theta) - y_{\mathrm{vap},\infty}}{1 - y_{vap,s}(\theta)}$$

This equation is given and then modified in such a way to obtain an understanding that there should be a linear relation between (Lmajor / Lmajor0)^2, and t / Lmajor0^2.

Based on this understanding of the expectation it can be said that something unknown is occurring in our isopropanol data. One guess could be that because our acoustic field is much less pure in nature than that of the paper we are comparing to, there could be an introduced

factor in surface vibrations that are causing evaporation to occur more abnormally in a naturally low surface tension liquid like the isopropanol.

To conclude, while we were able to reasonably map our data collected for water to the d^2 law, it seems something strange is at play in our isopropanol data. With the data we have there are simply too many factors to pinpoint what it is, but we believe that while in the time of this project we were unable to isolate certain parameters, this topic of mass transfer could be taken and adopted in an undergraduate classroom setting to have students determine and model the mass transfer of these oblate spheroid droplets. In Source 2, they (for example) were able to obtain a clear model prediction of droplet evaporation:



The general concepts explored here are excellent starting points however for further scientific use of something that may have appeared novel at first glance. Other researchers have taken acoustic levitation and used it to simulate zero gravity for bacterial growth, and some have used it to create volumetric displays. We hope that this specific example of mass transfer can serve as a springboard into coming up with a unique way to utilize acoustic levitation as a teaching tool in your classroom!

Bibliography:

- Marzo, A., Barnes, A. C., & Drinkwater, B. W. (2017). TinyLev: A multi-emitter single-axis acoustic levitator. *Review of Scientific Instruments*, *88*(8), 085105. https://doi.org/10.1063/1.4989995.
- Zang, D., Yu, Y., Chen, Z., Li, X., Wu, H., & Geng, X. (2017). Acoustic levitation of liquid drops: Dynamics, manipulation and phase transitions. *Advances in colloid and interface science*, *243*, 77-85. Acoustic levitation of liquid drops: Dynamics, manipulation and phase transitions - PubMed (nih.gov)
- Belal Al Zaitone, Oblate spheroidal droplet evaporation in an acoustic levitator, International Journal of Heat and Mass Transfer, Volume 126, Part B, 2018, Pages 164-172, ISSN 0017-9310, https://doi.org/10.1016/j.ijheatmasstransfer.2018.06.029.

Acknowledgments:

Supported by funding from the University of Utah's Chemical Engineering Department. Supervision and advice given by Dr. Anthony Butterfield, Dr. Stacy Firth, and Dr. Swomitra Mohanty.

Module developed by Juliana Ortiz-Castillo, Tyler Owens, and John Peterson.