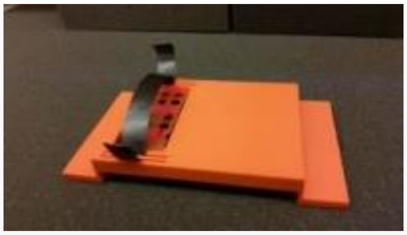
## Module 1: Measurement, Mechanism, and Modeling

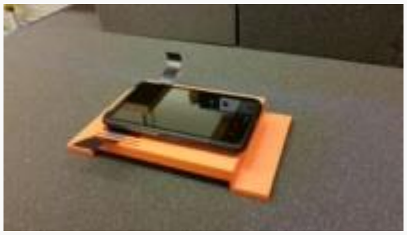
**Description:** Introduction to the intersection of physics and biology including information on the tools (3D printing, microscopy, electronics, MATLAB, etc.) and methods used in studying the physics of living systems.

**Protocols:**

# Smartphone microscope (phonescope)

**Goal:** Turn your smartphone into a microscope by designing and 3D printing a stage (or clip) that can be used with ball lenses of varying diameter (inspired by the work of Anton van Leeuwenhoek).





## Materials

- 3D printer
- calipers
- rubber sheet
- ball lenses:
    - dia. = 1.0mm
    - dia. = 0.7mm
- Solidworks
- smartphone (with front & back camera)
- LEDs/phone flashlight
- light diffuser (filter paper, plastic, etc.)

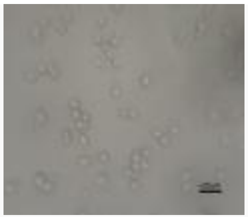
## Procedure

1. Design your own smartphone microscope in Solidworks that includes the following features:
    - slide holder
    - lens holder
    - aperture (light window)
    - adjustable field of view (ability to explore sample on slide)
    - able to accommodate multiple lenses (with different diameters)

2. Double check all measurements with the calipers and then print your design.
3. Using a small needle, poke a hole in a strip of the rubber sheet and insert a ball lens.
4. Add the rubber sheet to your printed "phonescope."

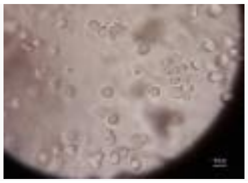
## Image Gallery

All photos and videos are taken with the smartphone microscope (magnification ~100X) using a 0.7mm diameter glass bead.



Unicellular budding yeast

"Snowflake" yeast



Mouse macrophages

Rotifers (gif)

Female rotifer with an egg (gif)

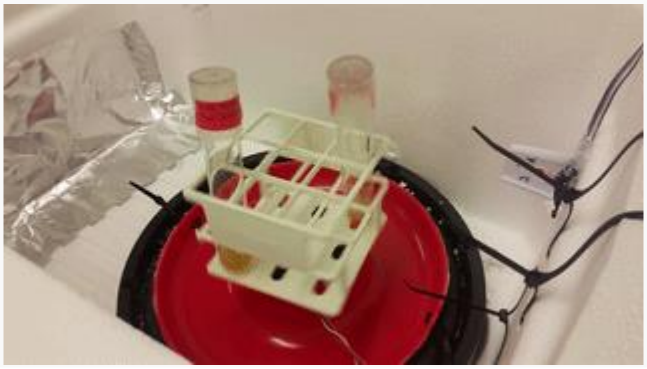
C. elegans on an agar pad (gif)

## Resources

- Anton van Leeuwenhoek
- Understanding Ball Lenses (Edmund Optics)

# Shaking incubator

**Goal:** Construct a laboratory shaking incubator that can provide vigorous shaking for culture tubes (25mm X 150mm test tubes) and maintains a constant, uniform temperature (30 deg C OR 37 deg C).



# Materials

- 12" bass speaker/ subwoofer
- audio mono amplifier (each amplifier should drive 2 speakers)
- function generator:
  - for testing: smartphone signal generator app
  - for culturing: Arduino – tone() command
- breadboard
- 3.3-5V wall wart/battery (for Arduino)
- power supply for amplifier (ADC comes with amplifier)
- base & test tube holder
- hot glue gun & glue
- alligator clips/lead wires
- Arduino Uno & 6' USB plug
- servo motor
- large Styrofoam (or other insulating material) box
- 40W light bulb
- socket & plug for light bulb
- TMP36 (temperature sensor)
- solid state relay (SSR - switch)
- extension cord
- microSD card (optional)
- transflash microSD breakout board (optional)

# Procedure

1. First, gather materials and ensure that speaker fits inside the insulating box with sufficient clearance and spare room for test tube holder (cut box and reseal, if needed).
2. Connect extension cord to SSR output terminals and solder wires to the input terminals.
3. Connect TMP36 to Arduino (first with breadboard, then by soldering once it's working).
4. Copy (and/or improve) the example code included into Arduino program and upload to board.
5. Calibrate your code such that the temperature sensor is used to control whether the light bulb (heat source) is on or off (bang-bang control).
6. Download a signal generator app and set the Frequency Range Minimum to 0 [Hz] and the Frequency Range Maximum to (at least) 20K [Hz].
7. Connect your speaker(s) to the output of the audio amplifier and then connect the amplifier to the power supply.
8. Using the signal generator app on your phone, supply a signal to the amplifier and test the output of the speaker (disconnect power supply when done).
9. Now, use the tone() command to send a signal to the speaker through the Arduino (see example code below) and figure out the best frequency for shaking (40 Hz?, 45 Hz?, 200 Hz?)
10. Once the shaking level is sufficient, hot glue your printed test tube rack & plate to the speaker.
11. Finally, test if incubator can maintain a constant temperature while shaking overnight.

## Example code

*For Arduino program:*

```
#include <Servo.h>

//TMP36 Pin Variables
int sensorPin = 0; //the analog pin the TMP36's Vout (sense) pin is connected to
                        //the resolution is 10 mV / degree centigrade with a
                        //500 mV offset to allow for negative temperatures

// light bulb
int RELAY1 = 9;

// speaker
int shake1 = 10;
int shake2 = 11;

// servo
int servoPin = 6;
int angle = 0;   // servo position in degrees
Servo servo;



/*
 * setup() - this function runs once when you turn your Arduino on
 * We initialize the serial connection with the computer
 */
void setup()
{
  Serial.begin(9600);  //Start the serial connection with the computer
                       //to view the result open the serial monitor
  pinMode(RELAY1, OUTPUT);
  pinMode(shake1, OUTPUT);
  pinMode(shake2, OUTPUT);

servo.attach(servoPin);

}
```

```
void loop()                     // run over and over again
{
 //getting the voltage reading from the temperature sensor
 int reading = analogRead(sensorPin);

 // converting that reading to voltage, for 3.3v arduino use 3.3
 float voltage = reading * 5.0;
 voltage /= 1024.0;

 // print out the voltage
 Serial.print(voltage); Serial.println(" volts");

 // now print out the temperature
 float temperatureC = (voltage - 0.5) * 100 ;  //converting from 10 mv per
degree wit 500 mV offset
                                               //to degrees ((voltage -
500mV) times 100)
 Serial.print(temperatureC); Serial.println(" degrees C");

 //cells = 37   // incubator ideal temp
 //yeast  = 30  incubator ideal temp

//while(1){
  //digitalWrite(RELAY1,LOW);
 // delay(1000);
 // digitalWrite(RELAY1,HIGH);
 // delay(1000);
//}

  if (temperatureC < 30) {
    digitalWrite(RELAY1,HIGH);
    Serial.print("T < 30 degrees C -- LIGHT ON");
    Serial.print("\n");
   } else {
   digitalWrite(RELAY1,LOW);
   Serial.print("T > 30 degrees C -- LIGHT OFF");
    Serial.print("\n");
  }
 tone(shake1,45);
 tone(shake2,45);


  // scan from 0 to 40 degrees
  for(angle = 0; angle < 40; angle=angle+20)
  {
    servo.write(angle);
    delay(300);
  }
  // now scan back from 40 to 0 degrees
  for(angle = 40; angle > 0; angle=angle-20)
  {
    servo.write(angle);
    delay(300);

 delay(10000);      // wait 10 seconds



 }
```

*For Processing program:*

```
import java.util.zip.GZIPOutputStream;
import java.io.ObjectOutputStream;
import java.io.FileOutputStream;
import processing.serial.*;

Serial mySerial;       // serial port
String value1,value2;  // to read in temperature data
String[] data;
int counter = -1;      // to increment the text file



void setup () {
  println(Serial.list());  // print the serial ports
  frameRate =30;

  mySerial = new Serial(this, Serial.list()[2], 9600);  // choose serial port
for arduino
```

```
  mySerial.bufferUntil('\n');
  data = new String[10000000];                          // arbitrarily large
array
                                                        // (size determined
after running
                                                        // shaker overnight)
}

void draw() {

  // test printing
  //println("counter");
  //println(counter);
  //println("value 1");
  //println(value1);
  //println("value 2");
  //println(value2);
  //println(value2==value1);

  // check data from serial monitor
  if (value2!=value1){
  counter++;

  // store data in large array
  data[counter] = value1;

  // save approx. every second
  if (counter%30==0){
  saveStrings("temps.txt", data);
  }
}
  value2=value1;
  }

void serialEvent(Serial mySerial){
 String val = mySerial.readString();      // read in data from serial monitor

   if (val != null){
     value1 = val.substring(0,(val.length()-2));   // chop off extra
whitespace/new                                               // line
```

```
   }
}
```

## Save to a text file on a microSD card

*For Arduino program:*

```
#include <SPI.h>
#include <SD.h>
#include <Servo.h>

//TMP36 Pin Variables
int sensorPin = 0; //the analog pin the TMP36's Vout (sense) pin is connected
to

                        //the resolution is 10 mV / degree centigrade with a
                        //500 mV offset to allow for negative temperatures

int servoPin = 6;
int RELAY1 = 9;
int shake1 = 10;




const int chipSelect = 8;

Servo servo;

//servo position (degrees)
int angle = 0;

/*
 * setup() - this function runs once when you turn your Arduino on
 * We initialize the serial connection with the computer
 */
void setup()
{
  Serial.begin(9600);  //Start the serial connection with the computer
                       //to view the result open the serial monitor
  pinMode(RELAY1, OUTPUT);
  pinMode(shake1, OUTPUT);
  //pinMode(shake2, OUTPUT);

  servo.attach(servoPin);

   Serial.print("Initializing SD card...");
   // check for SD card
   if (!SD.begin(chipSelect)){
    Serial.println("SD card failed or not present");
    return;   }
    Serial.println("SD card initialized");
    }

void loop()                     // run over and over again
{

 //getting the voltage reading from the temperature sensor
 int reading = analogRead(sensorPin);

 // converting that reading to voltage, for 3.3v arduino use 3.3
 float voltage = reading * 5.0;
 voltage /= 1024.0;

 // print out the voltage
 Serial.print(voltage); Serial.println(" volts");

 // now print out the temperature
 float temperatureC = (voltage - 0.5) * 100 ;  //converting from 10 mv per
degree wit 500 mV offset
                                               //to degrees ((voltage -
500mV) times 100)
 //Serial.print(temperatureC); Serial.println(" degrees C");


  if (temperatureC < 27) {
    digitalWrite(RELAY1,HIGH);
    //Serial.print("T < 25 degrees C -- LIGHT ON");
    //Serial.print("\n");
   } else {
   digitalWrite(RELAY1,LOW);
   //Serial.print("T > 30 degrees C -- LIGHT OFF");
    //Serial.print("\n");
  }

  tone(shake1,45);


//  scan from 0 to 180 degrees
  for(angle = 0; angle < 40; angle=angle+20)
  {
    servo.write(angle);
    delay(300);
  }
//  now scan back from 180 to 0 degrees
  for(angle = 40; angle > 0; angle=angle-20)
  {
    servo.write(angle);
    delay(300);
  }

 //delay(1000);      // wait 10 seconds

 // start string for datalogging
  String tempdata = "";


  // read in data from TMP36
tempdata += String(temperatureC);


// open temperature sensor data file
  File dataFile0 = SD.open("temp.txt", FILE_WRITE);

//   write to file
  if (dataFile0) {
    dataFile0.println(tempdata);
    dataFile0.println("\n");
    dataFile0.close();
    Serial.print(tempdata);
    Serial.println("\n");
  }

//   if the file isn't open, pop up an error:
  else {
    Serial.println("error opening temp.txt");
  }



  delay(300);
 }
```

## Resources

- Continuous rotation servo

**Other DIY incubators**

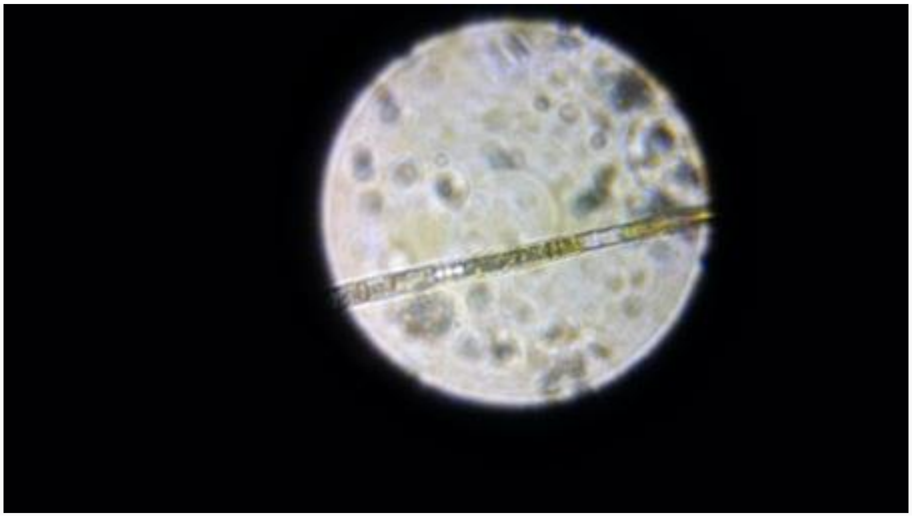
- one
- two
- three
- four

# Pond scum microscopy

**Goal:** To achieve a deeper understand of the amount of life present in a drop of water students will use their 3D printed microscopes and smartphones to capture an image of an organism in pond scum. Based on the images and observations, they must also identify the organism.

## Examples

**Smartphone microscope (phonescope) [~100-150X]:**



***Oedogonium* 100X magnification:**

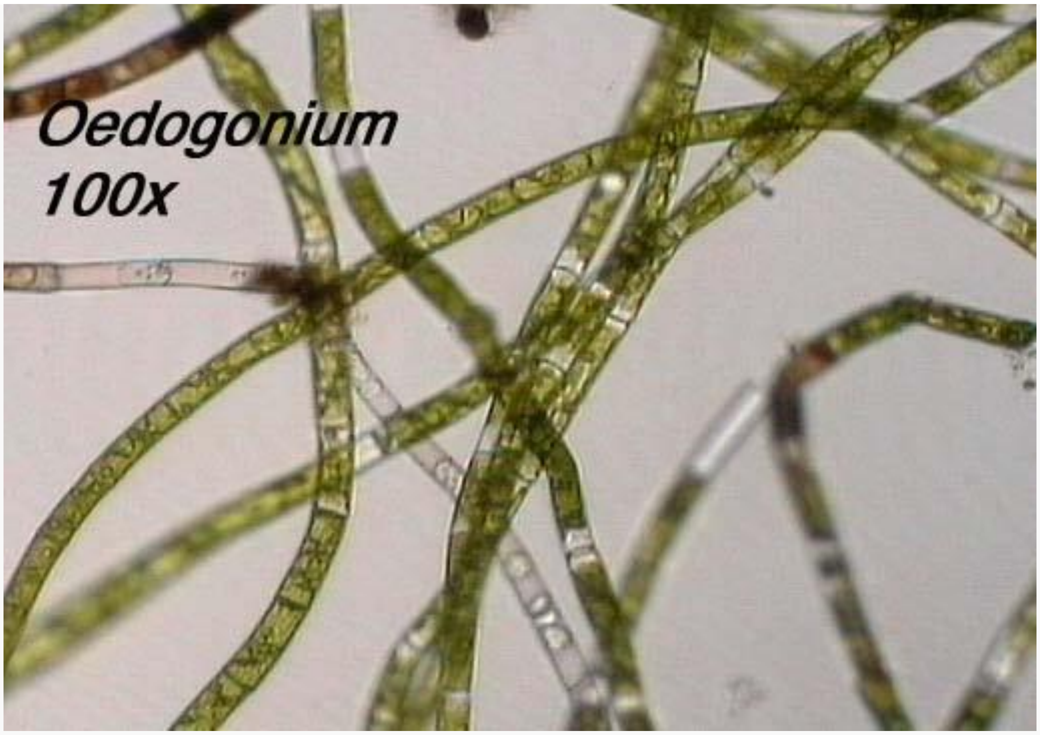
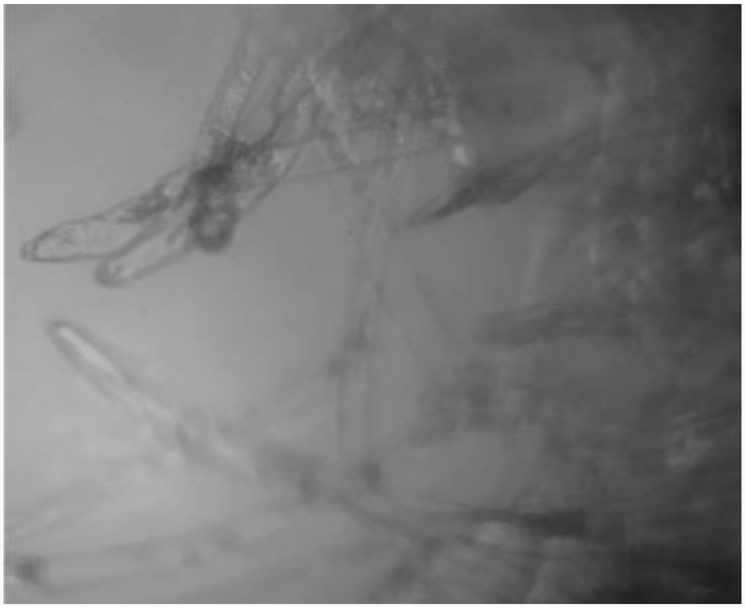


photo credit: WISC

**Smartphone microscope (phonescope) [~100-150X]:**



***C. sphaerosperum:***

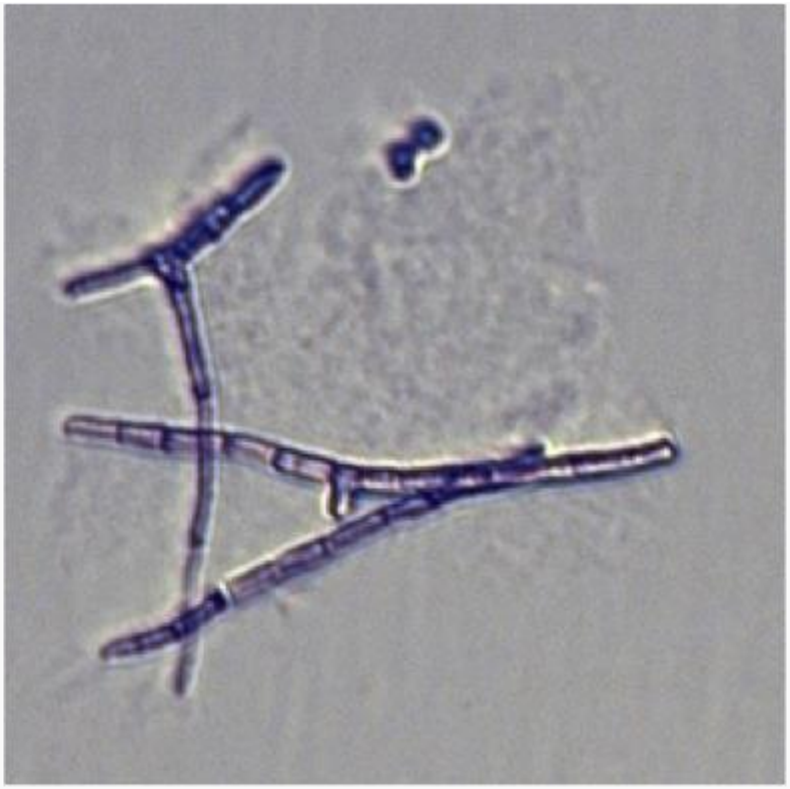


photo credit: *unknown*

# Resources

- Pond scum
- Waterbear