# Piezo Elements and Your Rube Goldberg Machine
# Making Your RGB Do Something

Your objective is simple. Your Rube Goldberg Machine (RGM) must activate a piezo element that is controlled by an Arduino controller that you have programmed. The analog input of the piezo element that is generated when the preceding group's marble strikes your piezo element will cause a servo motor to turn and allow your marble to start rolling down your RGM.

The following materials provide you with some guidance, hints, and instructions for making a successful Arduino program. READ EVERYTHING CAREFULLY . . . if you expect this to help you.

Piezo elements are sometimes known as knock sensors. Please go to this link https://www.arduino.cc/en/Tutorial/KnockSensor , read the first four paragraphs, and look over the first sketch to begin to understand how piezo elements work, and how to incorporate them in the Arduino universe. Given what you read at this link, answer the following questions (using either MS Word or Google Docs):

1. Previously when lighting up LED's, you were using digital pins. Using a piezo to detect a knock requires using analog pins. Where are these located on the Arduino relative to the digital pins, and how are they labelled?

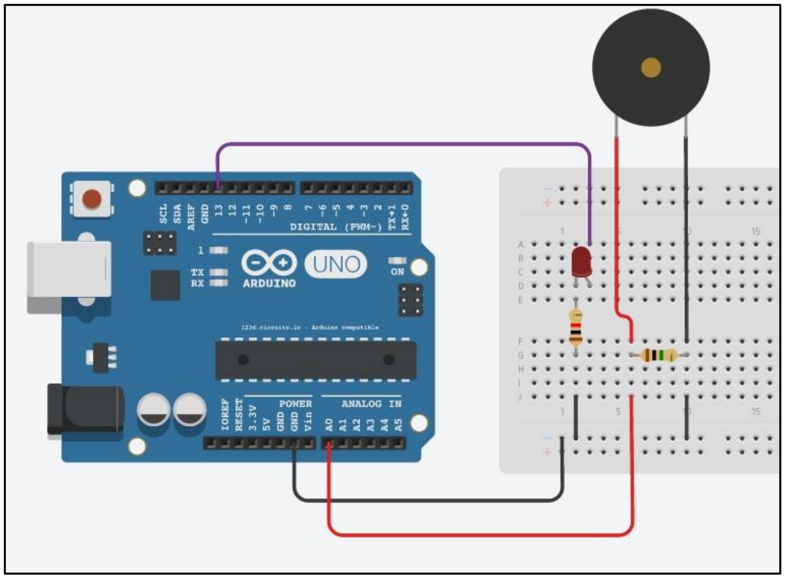
2. Digital pins usually utilize only one of two values: either HIGH (5 Volts), or LOW (0 Volts). What range of values does an analog pin read (0 to 5 Volts is translated into a number between what and what)?

3. Which of the two wires on the piezo (red or black) must be connected to ground?

4. When using a piezo, it is best to use it in conjunction with a resistor. How should this resistor be connected to the piezo, and what size should the resistor be? (Hint: elements in circuits are usually connected in either series or parallel relative to one another, and resistances are measured in Ohms.)

The following code (which is the sketch you just looked at with some modifications) will allow you to read the input from a piezo element that senses a knock. Two things occur when the piezo element senses a knock higher than the threshold value (100 in this example). First, when the piezo measures a knock exceeding the threshold value, the Arduino toggles the LED on DigitalPin 13 on/off. Secondly, the Arduino prints the value being registered from the piezo on AnalogPin 0 to the serial monitor. The serial monitor allows us to interact with the Arduino while the Arduino is running a sketch. We can send information to the running Arduino, and the Arduino can send information that we can see and interpret. In this sketch, we utilize the latter function and use the serial monitor to see what the piezo is sensing regardless of its value. Every 100 milliseconds the value of the piezo sensor is recorded on the serial monitor. Printing to the serial monitor is a part of this program because it provided a quick means of checking the effectiveness of the threshold value, and allows additional troubleshooting for the sketch. It allows this because you can continually see what the sensor is measuring, and what reading is on the AnalogPin. Therefore, you know when the LED should come on, for instance. You can also relate the size of the impact on the piezo element to the value that is sensed. This will be really important when calibrating your RGM for operation with an incoming marble.

Using a breadboard, a piezo, a resistor, a 1 MΩ resistor, a 1 kΩ resistor, and an LED, wire up the following circuit as shown here.



Next, copy this code into the Arduino IDE. You can cut and paste, but the formatting will not be correct. Spend some time realigning comments and statements within the IDE window so that your sketch is neat. If you do this now, the code will be ready to quickly employ later in the class and will require minimal modification.

```
/*      KNOCK SENSOR
        This sketch reads a piezo element to detect a knocking sound.
        It reads an analog pin and compares the result to a set threshold.
        If the result is greater than the threshold, it writes
        "knock" to the serial port, and toggles the LED on pin 13.
        THE CIRCUIT:
        * + connection of the piezo attached to analog in 0
        * - connection of the piezo attached to ground
        * 1-megohm resistor is attached in parallel to the piezo.
*/

// these constants won't change:
const int ledPin = 13;              // led is connected to digital pin 13
const int knockSensor = A0;         // the piezo is connected to analog pin 0
const int threshold =100;           // the threshold value to decide when the detected sound is a knock or not

// these variables will change:
int sensorReading = 0;              // variable to store the value read from the piezo sensor pin
int ledState = LOW;                 // variable used to store the last LED status, to toggle the light

void setup() {
  pinMode(ledPin, OUTPUT);          // declare the ledPin as an OUTPUT
  Serial.begin(9600);               // this statement sets up the serial port so that the serial monitor can be used
}

void loop() {
  sensorReading = analogRead(knockSensor);    // read the sensor and store it in the variable sensorReading:
  Serial.println(sensorReading);              // send the string "Knock!" back to the computer, followed by a newline
  if (sensorReading >= threshold) {           // if the sensor reading is greater than the threshold:
    // toggle the status of the ledPin:
    ledState = !ledState;                     // This flips the state of the LED from HIGH to LOW or vise versa.
    digitalWrite(ledPin, ledState);           // update the LED pin itself
  }
  delay(100);                                 // delay here to avoid overloading the serial port buffer
}
```

Upload and run your code. You will need to open the serial monitor (in the top right of the IDE window) to see the output to the serial monitor. Adjust the threshold value on the code so that a gentle tap on the piezo is all that is required to change the state of the LED. Answer the following questions.

5. What final threshold number was required in order to cause the LED to toggle on/off with a gentle tap?

6. Try lowering the threshold number to 10 and upload your code to the Arduino. Fully describe the impact this has on the operation of the Arduino and the circuit.

7. Now, try raising the threshold number to 600 and upload your code to the Arduino. Again, fully describe the impact this has on the operation of the Arduino and the circuit.

8. What is the number that is being reported on the serial monitor each time a knock is sensed representing? In other words, what is this number measuring?

9. Use online resources (this site might be helpful https://www.arduino.cc/ ) to determine what the analogRead() function is doing in this program. What purpose does it serve?

10. Digital pins on the Arduino can be referenced by a simple number (i.e. 13) within a sketch. How must analog pins be referenced in a sketch? (Hint: look at the code).