

Piezo Elements and Your Rube Goldberg Machine

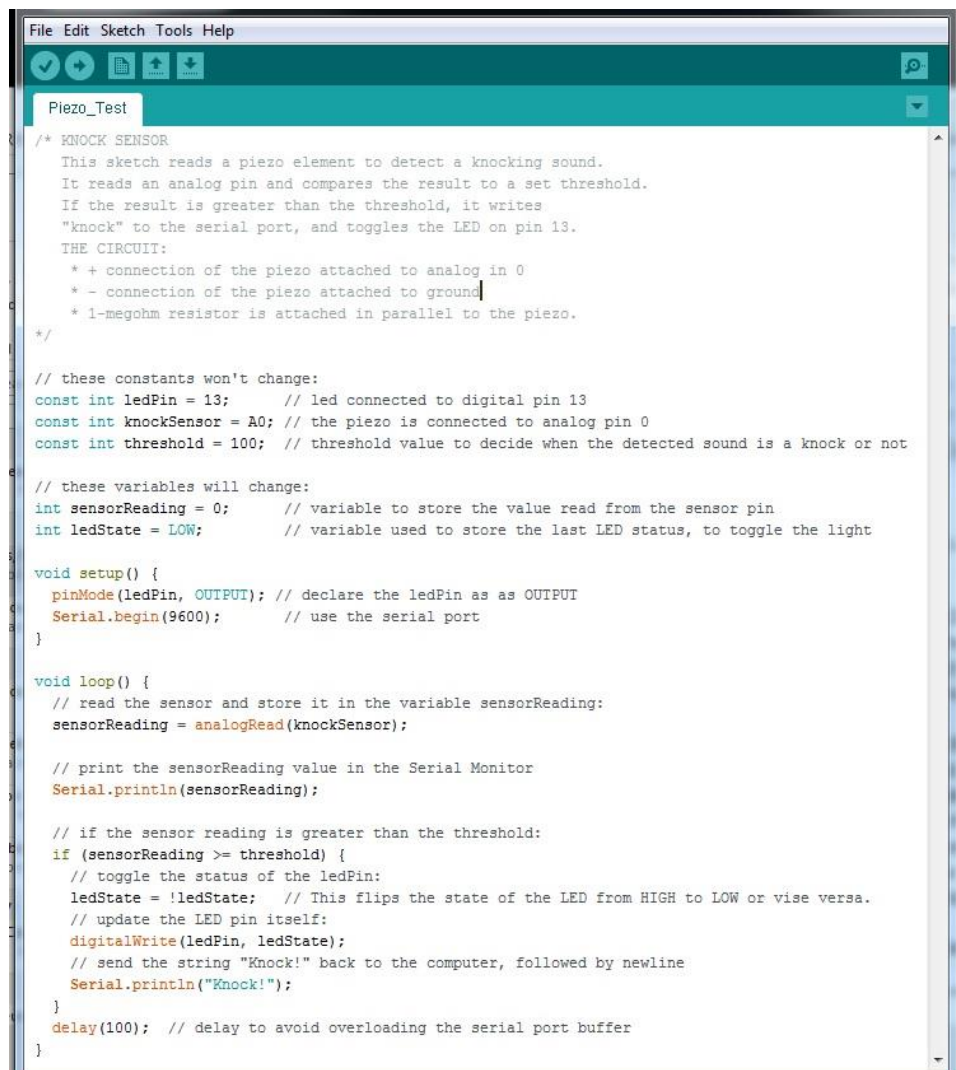
Making Your RGB Do Something

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 is to be used to cause something to occur that the Arduino controls.

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> and 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.

In the example program to the right, which is the sketch you just looked at with some modifications, two things occur when the piezo element senses a knock. First, when the piezo measures a knock that is higher than the threshold value (100 in this example), the Arduino will toggle the LED on DigitalPin 13 on/off, and it prints the value being registered on AnalogPin 0, the pin that the piezo is connected to, to the serial monitor. Also note that 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 allowed 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.

A screenshot of the Arduino IDE interface. The title bar shows 'File Edit Sketch Tools Help'. Below the menu bar is a toolbar with icons for opening, saving, compiling, and uploading. The main text area contains a C++ sketch titled 'Piezo_Test'. The sketch includes comments explaining its purpose: to detect a knocking sound using a piezo element connected to analog pin 0, and to toggle an LED on digital pin 13 if the sound exceeds a threshold of 100. The code defines constants for the LED pin, knock sensor pin, and threshold. It also defines variables for sensor readings and LED state. The setup function initializes the LED pin as an output and starts the serial port at 9600 baud. The loop function reads the sensor value, prints it to the serial monitor, checks if it exceeds the threshold, and toggles the LED state if it does. A 100ms delay is used to prevent serial buffer overflow.

```
File Edit Sketch Tools Help

Piezo_Test

/* 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 connected to digital pin 13
const int knockSensor = A0; // the piezo is connected to analog pin 0
const int threshold = 100;  // 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 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 as OUTPUT
  Serial.begin(9600);      // use the serial port
}

void loop() {
  // read the sensor and store it in the variable sensorReading:
  sensorReading = analogRead(knockSensor);

  // print the sensorReading value in the Serial Monitor
  Serial.println(sensorReading);

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

What You Need To Do:

So now . . . you have a task. As all good programmers do, you are going to start with a program that has been created by someone else (the above sketch). You are to change it so that it does the following:

1. Responds to an impact to a piezo element. This might require adjusting the threshold value. It might also require you to use alligator clips and jumper wires so that the piezo can be located at the point of your RGB where it is to be struck.
2. The impact must be gentle enough to avoid damage to the piezo. Your instructors **MUST** approve your method for striking the piezo.
3. You must have your Arduino do something other than merely toggling on and off an LED – **WE HAVE ALREADY SHOWN YOU HOW TO DO THIS ABOVE!** We are leaving this up to you, but you must run your ideas by your instructors to receive approval first. We may or may not have materials to support what you want to do.

Other Criteria:

- When you are using your Arduino with your RGM, you will need to be powering your Arduino with a 110V power adaptor. We only have a limited number of these so you will have to share. In this mode, the Arduino receives power so that it can function from the power adaptor – it will **NOT** be hooked to a computer. Of course, you will have needed to load your program onto your Arduino prior to taking it to your RGM.
- Your RGM must have a place to hold your Arduino and breadboard **SO THEY WILL NOT GET DAMAGED.**
- Your piezo element must be hooked up in parallel with a 1.0 M Ω resistor. It will be difficult to get good readings if you don't do this.