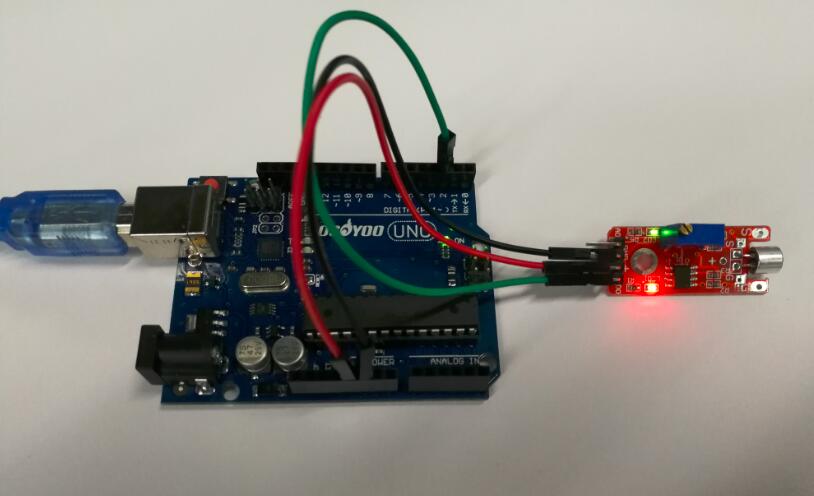
# Arduino lesson – Sound Detection Sensor

## Introduction

The Sound Detection Sensor is a small board that combines a microphone and some processing circuitry, it has the ability to detect different sizes of sound. This sensor can be used to for a variety of uses from industrial to simple hobby or playing around.

In this lesson we will guide you through hooking up and using the Sound Detector. It will examine how the circuit works, explain some details about getting the best performance from the Sound Sensor, then present some projects that demonstrate how to use it.

[](Arduino%20lesson%20–%20Sound%20Detection%20Sensor%20«%20osoyoo.com_files/sound-sensor-circuit.jpg)

Preparations

Hardware

Osoyoo UNO Board (Fully compatible with Arduino UNO rev.3)x 1

Sound Detection Sensor x 1

Breadboard x 1

Jumpers

USB Cable x 1

PC x 1

Software

Arduino IDE (version 1.6.4+)

About Sound Detection Sensor

Overview

The Sound Detection sensor module has a built-in capacitive electret microphone which is highly sensitive to sound. Sound waves cause the thin film of the electret to vibrate and then the capacitance changes, thus producing the corresponding changed voltage, so it can detect the sound intensity in ambient environment. Since the change is extremely weak, it needs to be amplified. We use a LM393 as the power amplifier here. You can adjust the sensitivity with by adjusting the Potentiometer. When the sound level exceeds the set point, an LED on the sensor module is illuminated and the output is sent low.

**Note:** This sound sensor is used to detect whether there’s sound surround or not, it cannot recognize the frequence or volum, please don’t use the module to collect sound signal.

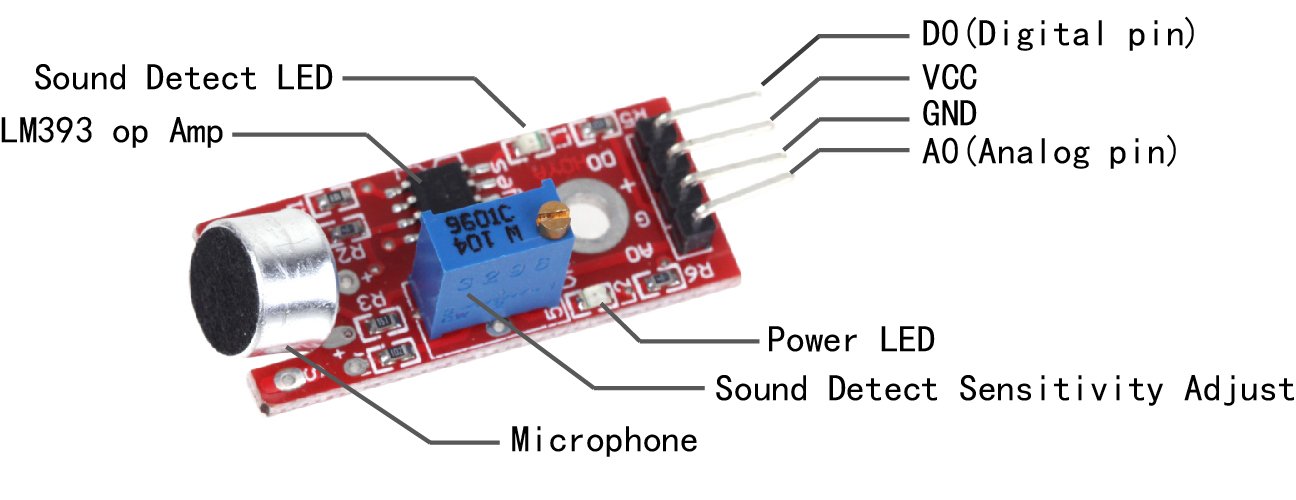
Arduino Sound Detection Sensor Pin Outs

The image and table below detail the controls, pin outs, and other key components.

When referring sensititivity, well. I mean:

When less sensitive, it takes more sound to trigger the device

When more sensitive, it takes less sound to trigger the device

[](Arduino%20lesson%20–%20Sound%20Detection%20Sensor%20«%20osoyoo.com_files/soundsensor-pinoutput-1.jpg)

|  |  |
| --- | --- |
| Parameter | Value |
| + | 5 V DC from your Arduino |
| G | GND from your Arduino |
| D0 | Connect to Digital Input Pin |
| A0 | Connect to Analog Input Pin |
| Power LED | Illuminates when power is applied |
| Sound Detect LED | Illuminates when sound is detected |
| Potentiometer | CW = More Sensitive  CCW = Less Sensitive |

It has **four pins** that needs to be connected to your Arduino. The top one(if you look at the image above), is **AO**. This should be connected to the analog input 0 on the Arduino(A0). The one beside that is **GND**, which is connected to ground, the **VCC** is connected to +5V, and the last one is **DO** – which is the digital output of the module, and should be connected to digital pin 2 on the Arduino.

On the top of the sound sensor is a little flathead screw you can turn to adjust the sensitivity and analog output of the sound sensor. To calibrate the sound sensor you can make some noise and keep turning it until you start seeing the sensor-LED on the module starts blinking with the rhythm.

Uses for the Arduino Sound Detector

Given that this device measures whether or not sound has exceeded a threshold, you’re basically left with determining what it is you want to do. What I mean by this is that you can do something when it is quiet and/or you can do something when it is loud. For example:

You could detect whether or not a motor is running.

You could set a threshold on pump sound so that you know whether or not there is cavitation.

In the presence of no sound, you might want to create an ambiance by turning on music.

In the presence of no sound and no motion, you may go into an energy savings mode and turn off the lights.

Examples

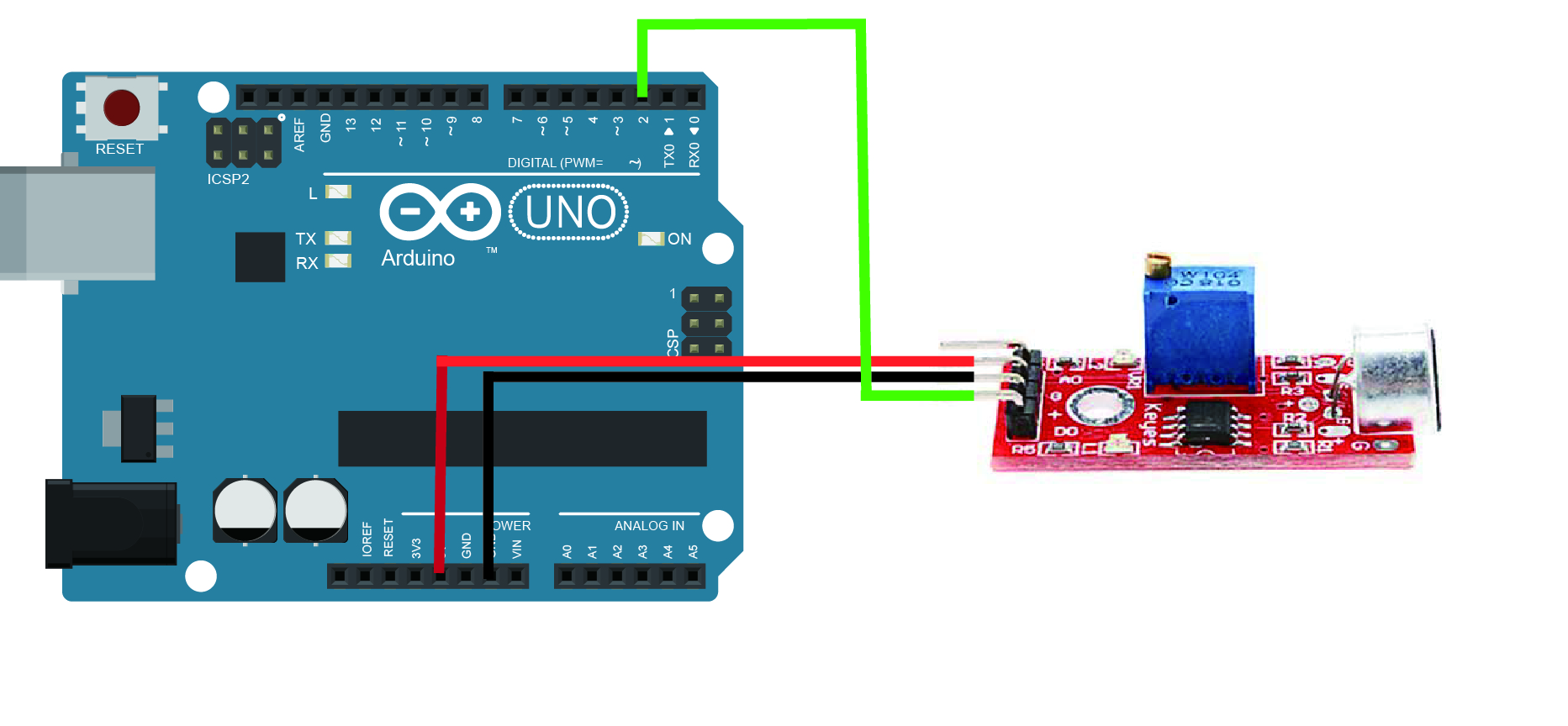
Digital Detect Sound Sensitive Lights

In this example, we are going to connect the Sound Detection sensor module to the Arduino digital pin to control the on-board LED, so that the LED will light up every time the sensor detects sound.

**Note**: The sensitivity of the Sound Detection sensor is adjustable – you may adjust it by the potentiometer.

### Connection

Overhere we use the D2 as the digital pin to connect with the sound sensor, build the circuit as below:

[](Arduino%20lesson%20–%20Sound%20Detection%20Sensor%20«%20osoyoo.com_files/circuit-sound-sensor.jpg)

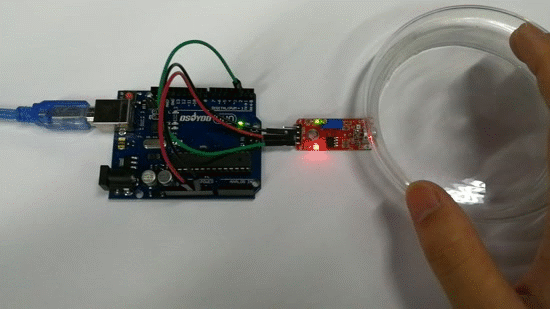
### Code Program

After above operations are completed, connect the Arduino board to your computer using the USB cable. The green power LED (labelled **PWR**)should go on.Open the Arduino IDE and choose corresponding board type and port type for you project. Then load up [the following sketch](https://raw.githubusercontent.com/osoyoo/Osoyoo-development-kits/master/Osoyoo%20lessons%20for%20Arduino/sound_sensor_digital.zip) onto your Arduino.

int digit\_sensor = 2;//select the input pin for the potentiometer int ledPin = 13;//select the pin for the LED int digitValue;//value from the digit input pin void setup (){ pinMode (ledPin, OUTPUT); pinMode (digit\_sensor, INPUT); **Serial**.begin (9600); } void loop (){ digitValue=digitalRead(digit\_sensor); if (digitValue==LOW){ digitalWrite (ledPin, HIGH); delay(50); }else{ digitalWrite (ledPin, LOW); delay(10); } }

### Running Result

A few seconds after the upload finishes, you can clap or knock something next to the sensor, when the volume reaches to a certain value, the LED attached to pin 13 on the Osoyoo Uno board will light up.

[](Arduino%20lesson%20–%20Sound%20Detection%20Sensor%20«%20osoyoo.com_files/soundsensor.gif)

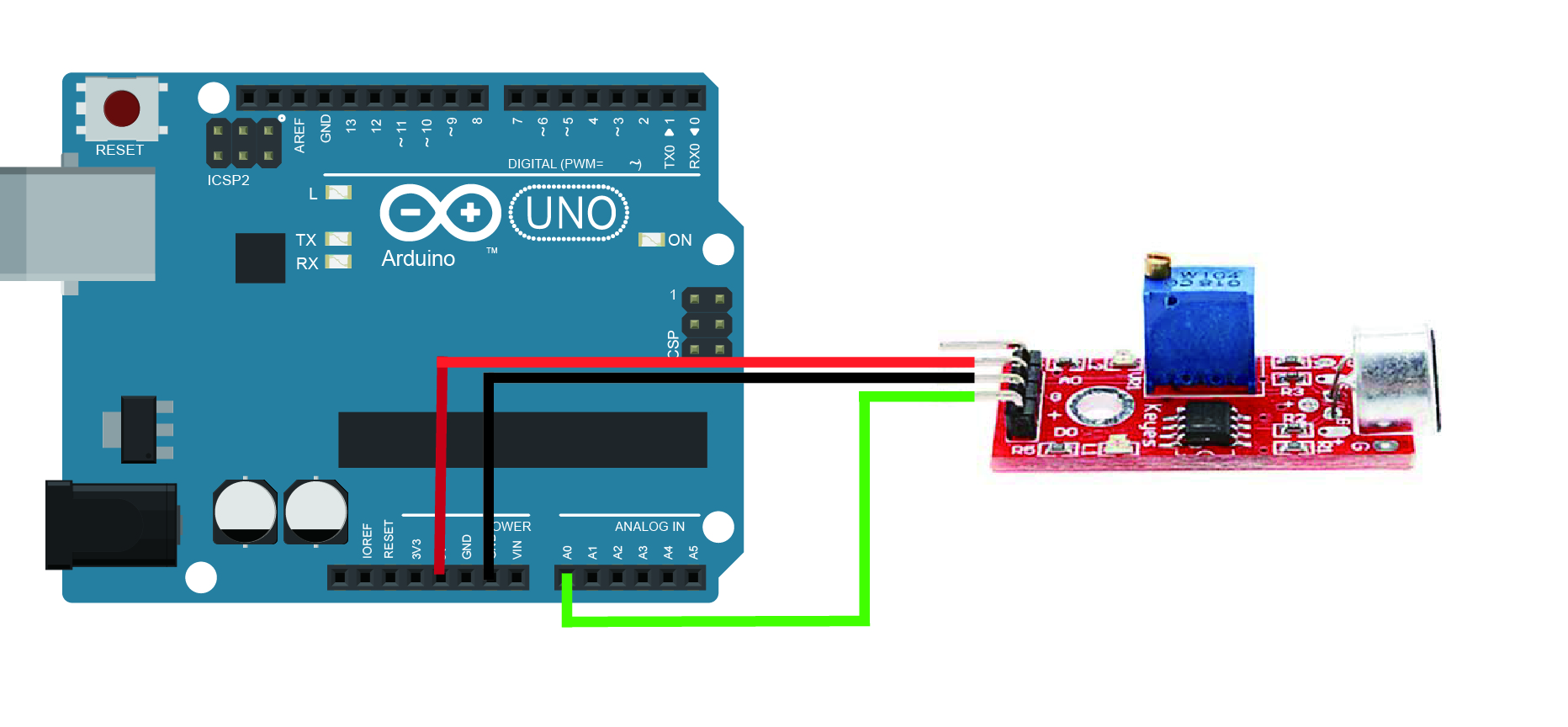
**Note:** If the LED is not lighting up or the LED is always bright, you need to change the sensor sensitivity by rotating the potentiometer.

Analog Detect Sound Sensitive Lights

In this example, we will show how to use the analog pin to detect the sound. The microphone sensor will detect the sound intensity of your surroundings and will light up an LED if the sound intensity is above a certain threshold.

### Connection

Overhere we use the A0 as the analog pin to connect with the sound sensor, build the circuit as below:

[](Arduino%20lesson%20–%20Sound%20Detection%20Sensor%20«%20osoyoo.com_files/circuit-sound-sensor-analog.jpg)

### Code Program

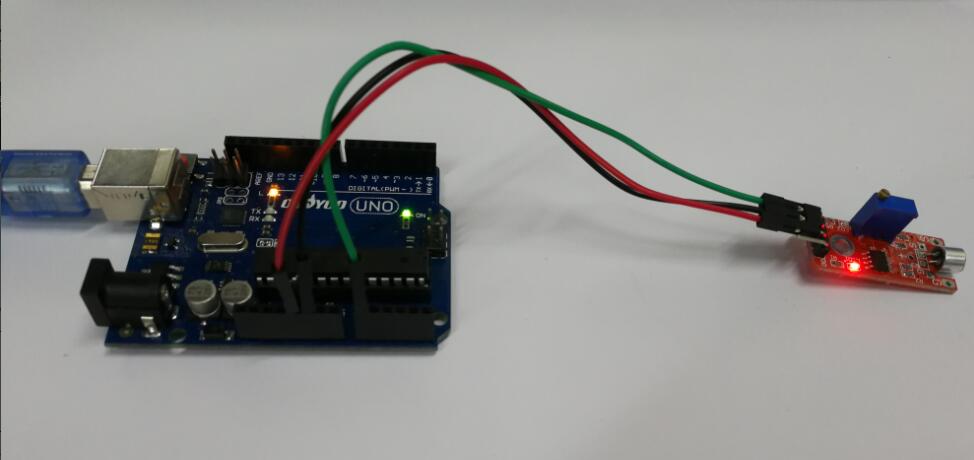
After above operations are completed, connect the Arduino board to your computer using the USB cable. The green power LED (labelled **PWR**)should go on.Open the Arduino IDE and choose corresponding board type and port type for you project. Then load up [the following sketch](https://raw.githubusercontent.com/osoyoo/Osoyoo-development-kits/master/Osoyoo%20lessons%20for%20Arduino/sound_sensor_analog.zip) onto your Arduino.

const int ledPin = 13;//pin 13 built-in led const int soundPin = A0;//sound sensor attach to A0 int threshold = 600;//Set minimum threshold for LED lit void setup(){ pinMode(ledPin,OUTPUT);//set pin13 as OUTPUT **Serial**.begin(9600);//initialize serial } void loop(){ int value = analogRead(soundPin);//read the value of A0 **Serial**.println(value);//print the value if(value > threshold)//if the value is greater than 600{ digitalWrite(ledPin,HIGH);//turn on the led delay(200);//delay 200ms }else{ digitalWrite(ledPin,LOW);//turn off the led } delay(1000); }

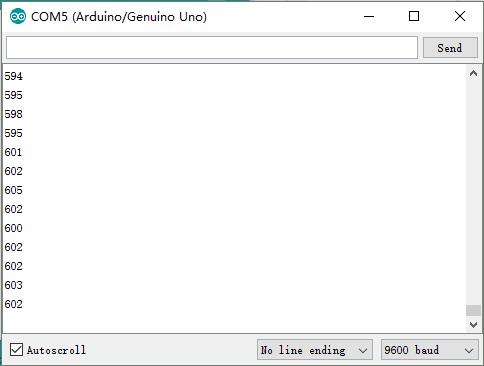
This is the code to make a LED blink with sound. You have to set the threshold so it' sensible enough to make the led blink, you can also see the value of sound intensity on Serial Monitor.

### Running Result

After copying and uploading this code, when the volume reaches to a certain value, the LED attached to pin 13 on the Uno board will light up. If the sound does not sense very well, try changing the threshold value or changing the sensor sensitivity by rotating the potentiometer.

[](Arduino%20lesson%20–%20Sound%20Detection%20Sensor%20«%20osoyoo.com_files/analog.jpg)

You can open the Serial Monitor by going to Tools > Serial Monitor or pressing the magnifying glass-button in the Arduino software window.

[](Arduino%20lesson%20–%20Sound%20Detection%20Sensor%20«%20osoyoo.com_files/serial-monitor.jpg)

What prints out is the analog and digital values of from the sound sensor module. The analog value should spike up when a noise occurs and stabilize when it gets quiet again. Now in the code there is an "int threshold = 600;" line that needs to be changed to something very close but higher than the value you get from the Serial Monitor when it is quiet around you. For instance if you see an analog value of 500, then threshold should be changed to perhaps 503 or 505. When a sound occurs, the analog value will rise and go above the threshold value. When that happens your LEDs will turn on. When it gets quiet again the analog value will go back to 503 and the LEDs go dark again.

Sound Sensor Module Project

Post Time: 2015-03-20 23:03:51 Category: [Arduino Parts & Projects](http://osoyoo.com/category/diy-parts/)

This project is to test Sound Sensor module. The sensor A0 pin will output analog sound value. the D0 pin will output HIGH voltage (1)if value over the threshold, otherwise export LOW (0). the threshold can be adjust by screwdriver.

**Step 1:** connect the sensor to Arduino as per following graph:

**Step 2:** [Download sample sketch code](http://osoyoo.com/wp-content/uploads/samplecode/soundsensor.txt) and load it into Arduino IDE.

/\* sound sensor test project

\* sensor project tutorial url: http://osoyoo.com/?p=508

\*/

int analog\_sensor = A0;//select the input pin for the potentiometer

int digit\_sensor = 12;//select the input pin for the potentiometer

int ledPin = 3;//select the pin for the LED

int analogValue;//value from the analog input pin

int digitValue;//value from the digit input pin

void setup (){

pinMode (ledPin, OUTPUT);

pinMode (digit\_sensor, INPUT);

Serial.begin (9600);

}

void loop (){

analogValue = analogRead (analog\_sensor);

digitValue=digitalRead(digit\_sensor);

Serial.println (analogValue, DEC);

if (digitValue==HIGH)

digitalWrite (ledPin, HIGH);

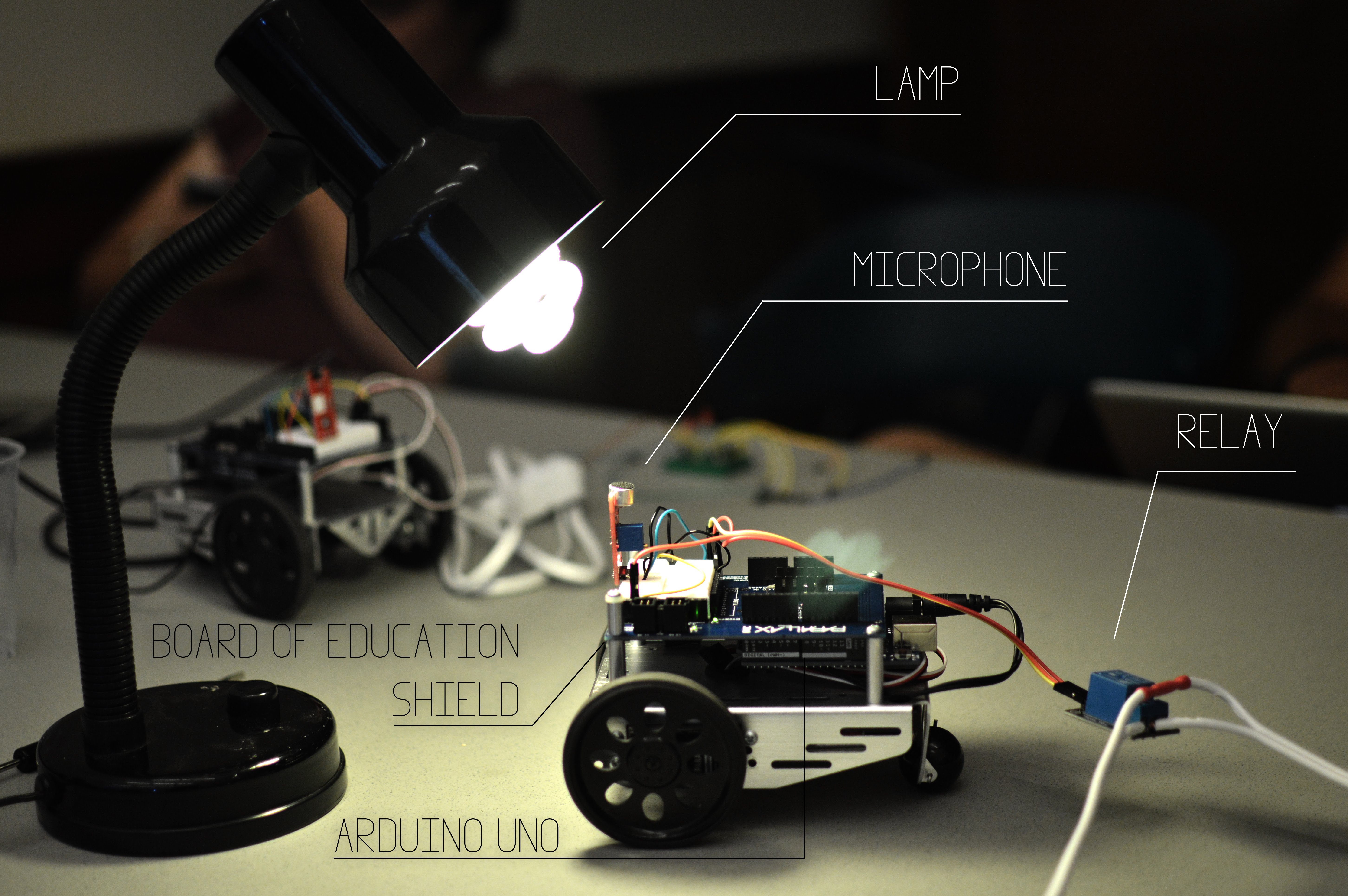
else digitalWrite (ledPin, LOW);

}

**Step 3:** click the serial monitor window, the monitor will show sound value as following:

Microphone Sensor (KY-038)

By Luan Mateus Vick

[](https://sites.google.com/site/summerfuelrobots/arduino-sensor-tutorials/microphones-with-arduino/Arduino.jpg?attredirects=0)

Introduction

When microphones were created in 1877, they were primarily used only to transmit sound in speakers or to record music. 137 years later, it is possible to notice that they are more important than ever in our society. They can be found in cellphones, cars, video game consoles, computers and even televisions, and are used for different purposes in each situations.

However, one of the most recent trends in the uses for microphones is **voice control**, were users are able to use their voices to control a computer or other electronic device without having to physically interact with it.

Many "smart homes" use microphone-enabled systems to allow the user to control, for example, a lamp. Inspired by this idea, this tutorial aims to teach how to make a clap-controlled lamp with an Arduino, demonstrating the use of a microphone sensor.

Technical Details

Microphone specifications

Frequency range:100 ~ 10,000 Hz.

Sensitivity: - 46 ± 2.0, (0 dB = 1V / Pa)at 1K Hz.

Power supply: 5V maximum

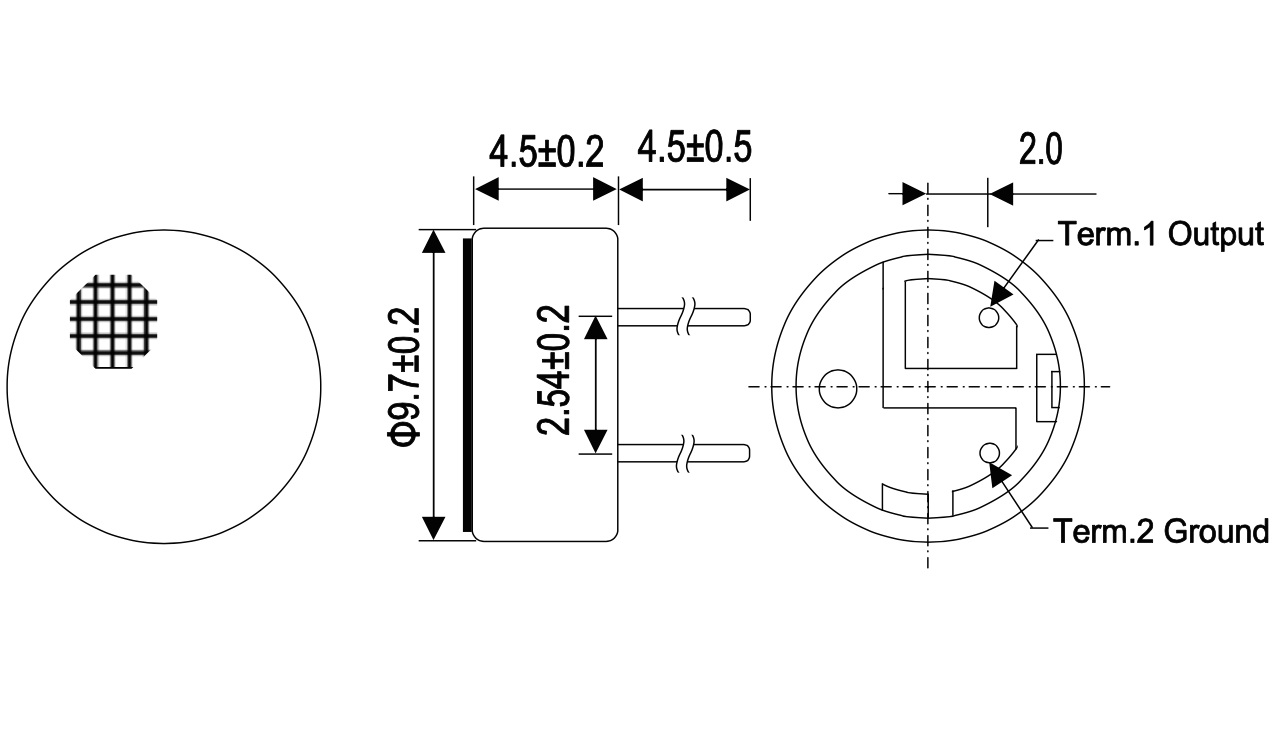
Minimum Sensitivity to Noise Ratio: 58dB

Microphone dimensions\*

Diameter: 9.7mm

Height: 4.5mm

Weight: 0.7 grams

[](https://sites.google.com/site/summerfuelrobots/arduino-sensor-tutorials/microphones-with-arduino/Microphone%20Specs.png?attredirects=0)

\*Dimensions for the microphone only, not the complete microphone module.

Relay specifications

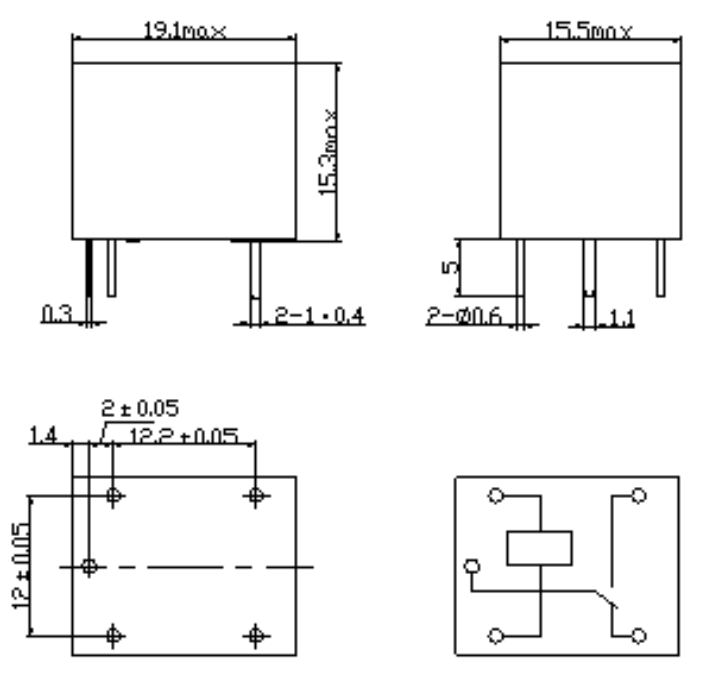
10A 250VAC

10A 125VAC

10A 30VDC

10A 28VDC

Relay dimensions (in millimeters)

[](https://sites.google.com/site/summerfuelrobots/arduino-sensor-tutorials/microphones-with-arduino/Relay%20dimentions.png?attredirects=0)

Similar Modules

There are many different models of microphones available for purchase, with different sensitivities, precision, frequency ranges and more.

Example of Use

The Arduino board is able to analyze certain levels of sound being detected by the microphone, and, if the level is high enough, it sends a signal to the relay, allowing power to flow to the lamp.

Materials / Equipment

Arduino UNO Board

A Board of Education Shield (optional)

8 wires

A KY-038 microphone (included in the [37-in-1 Sensor Module Kit for Arduino from dx.com](http://www.dx.com/p/arduno-37-in-1-sensor-module-kit-black-142834#.U8_rv4BdVqc))

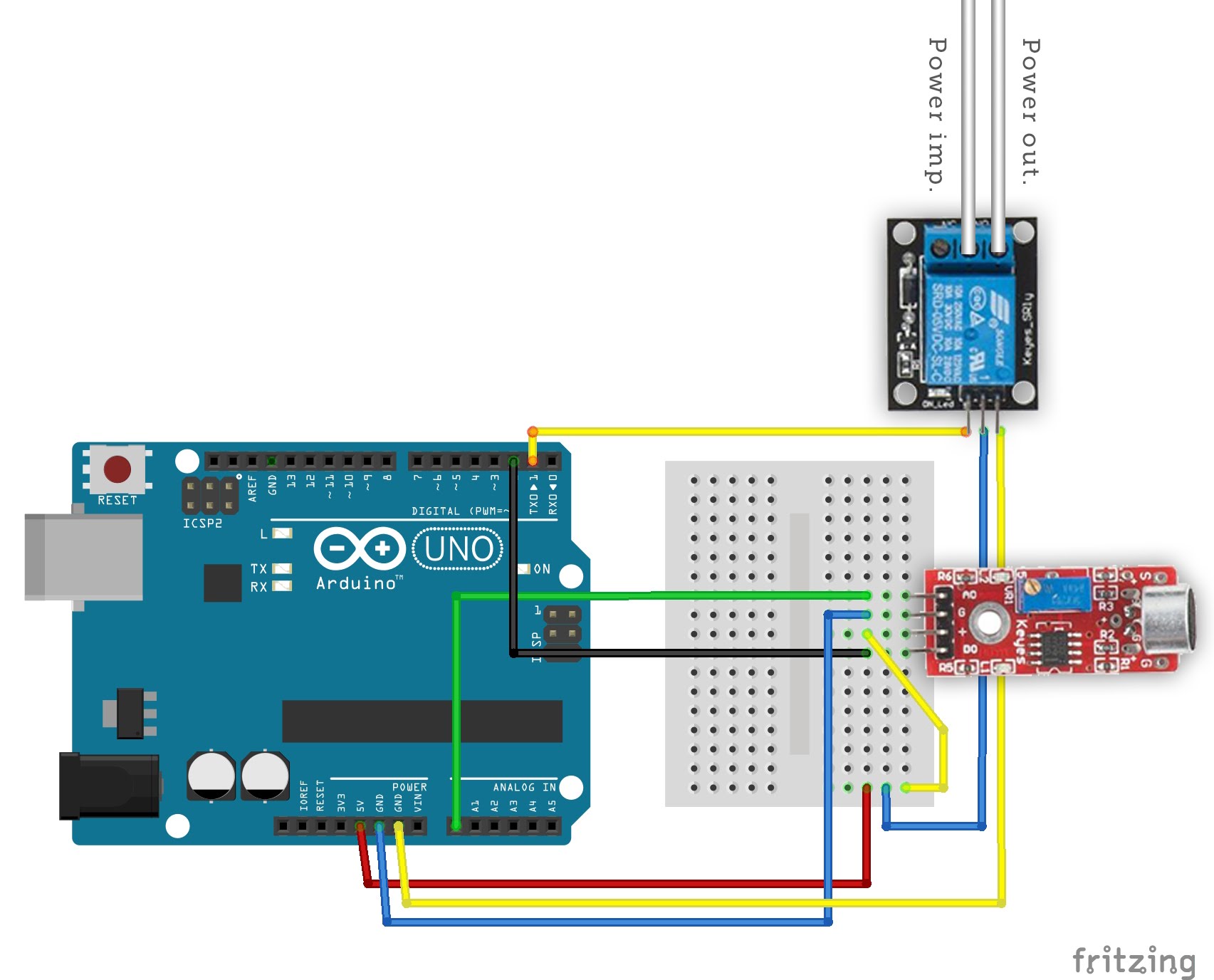
A Songle relay

A lamp

A extension cable

Wiring Instructions

On the microphone, wire the pin named AG to a analog input header on the Arduino board (in this case, A0). After this, wire the pin named G to a ground header on the board, and wire the pin named "+" to a 5V power output header in the Arduino board. Finally, wire the DO pin to a digital output header on the Arduino board (in this case, header 2). On the relay, wire the S pin to a digital header on the Arduino board (in this case, header 1). The "+" pin should be wired to a 5V power output, while the "-" pin should be wired to a ground header.

[](https://sites.google.com/site/summerfuelrobots/arduino-sensor-tutorials/microphones-with-arduino/Arduino%20Sketch.jpg?attredirects=0)

Sketch Instructions

After wiring everything up, upload this code to your Arduino board.

Sketch Code

\\ Sound-activated lamp with Arduino (Analog mode)

const int buttonPin = 0;

const int ledPin = 1;

int buttonstate = 16;

int ledstate = 0;

void setup(){

pinMode(ledPin, OUTPUT);

pinMode(buttonPin, INPUT);

}

void loop(){

buttonstate = analogRead(buttonPin);

ledstate = digitalRead(ledPin);

if (ledstate == HIGH && buttonstate > 17){

delay(250);

digitalWrite(ledPin, LOW);

delay(250);

}

if (ledstate == LOW && buttonstate > 17){

delay(250);

digitalWrite(ledPin, HIGH);

delay(250);

}

}

In case the microphone is not detecting your sounds properly, try changing its **sensitivity**, either in the hardware (by rotating a small screw on the blue resistor on the microphone)or in the software (by changing the number **17** in ledstate == HIGH && buttonstate > **17** and on ledstate == LOW && buttonstate > **17**to a different value).

**Try also...**

Instead of using the microphone in analog input, try testing the digital mode. To do this, try this code:

Sketch Code

\\ Sound-activated lamp with Arduino (Digital mode)

const int buttonPin = 2;

const int ledPin = 1;

int buttonstate = 16;

int ledstate = 0;

void setup(){

pinMode(ledPin, OUTPUT);

pinMode(buttonPin, INPUT);

}

void loop(){

buttonstate = digitalRead(buttonPin);

ledstate = digitalRead(ledPin);

if (ledstate == HIGH && buttonstate > 17){

delay(250);

digitalWrite(ledPin, LOW);

delay(250);

}

if (ledstate == LOW && buttonstate > 17){

delay(250);

digitalWrite(ledPin, HIGH);

delay(250);

}

}