

SOUND SENSOR

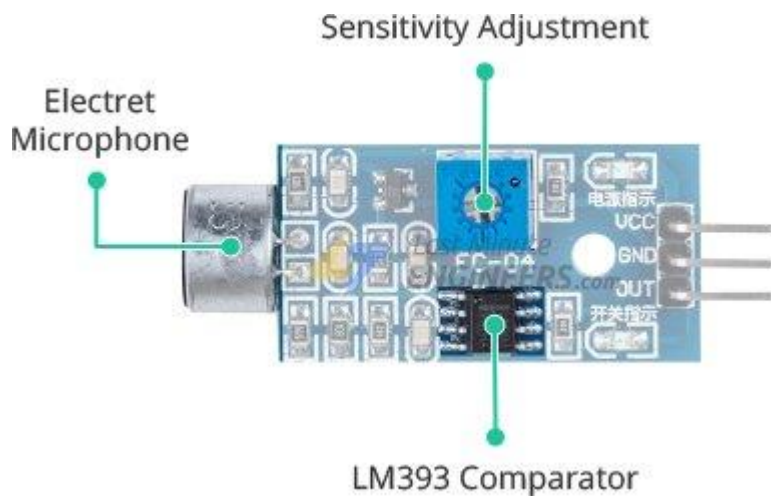
Ever wanted to control your electronics with just a clap or snap? With a sound sensor module, your Arduino can start “listening” to the world around it!

These inexpensive and easy-to-use sensors allow you to build sound-reactive projects—like clap-activated lights, noise-sensitive alarms, or even fun interactive gadgets.

In this tutorial, we'll walk you through how to hook up a sound sensor to an Arduino, understand how it works, and use it to trigger actions based on sound. Let's get started and give your project an ear!

HARDWARE OVERVIEW

The sound sensor module is a small circuit board that includes a microphone, an LM393 High Precision Comparator chip, and some additional electronic components that help detect the sound signals.



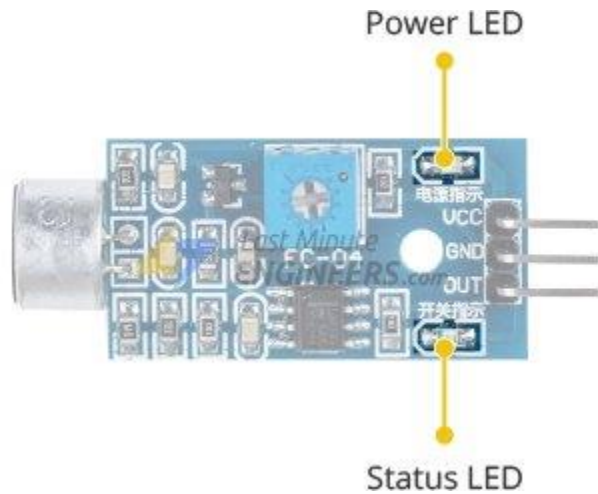
The microphone picks up sound waves from the environment, typically within a frequency range between 50Hz and 10kHz. It then converts these sound waves into analog voltage signals.

On the module, there's a potentiometer (a small knob) that allows you to adjust how sensitive the sensor is. In simpler terms, this lets you set a threshold or limit for detecting sound. When you turn the knob counterclockwise, the sensor becomes more sensitive and will detect quieter sounds. If you turn it clockwise, it becomes less sensitive and will only detect louder sounds.

The LM393 comparator chip plays an important role. It compares the analog signals from the microphone to the threshold level you have set with the potentiometer. If the

incoming sound is louder than your threshold, the sensor's output will switch to LOW. If the incoming sound is quieter than the threshold, the output stays HIGH.

This feature allows you to configure the sensor to respond only to sounds that exceed a certain loudness. This capability is particularly useful in projects where you want specific actions triggered by sound. For example, you can set the sensor to activate a relay and turn on a light whenever someone claps loudly.

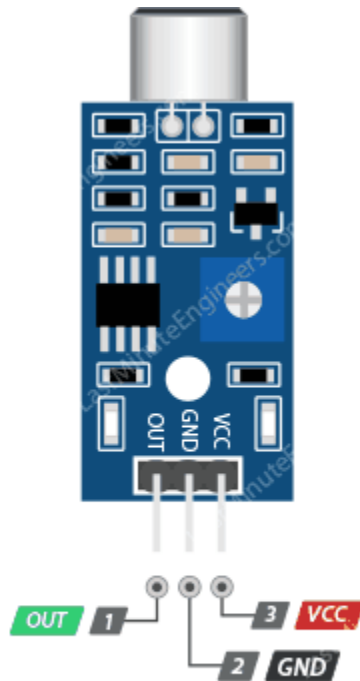


Additionally, the module has two small LEDs:

- The Power LED turns on whenever the module is powered up.
- The Status LED lights up whenever the detected sound exceeds your set threshold.

SOUND SENSOR PINOUT

The sound sensor module has only three pins:



OUT is the output pin. When the sound level is louder than the threshold you set using the potentiometer on the board, the OUT pin sends a LOW signal. When sound is quieter than the threshold, it sends a HIGH signal. You can connect this pin to any digital pin on an Arduino or directly to a 5V relay module to control other devices.

GND is the ground pin.

VCC is used to supply power to the module. You should connect it to a voltage source between 3.3V and 5V.

WIRING A SOUND SENSOR TO AN ARDUINO

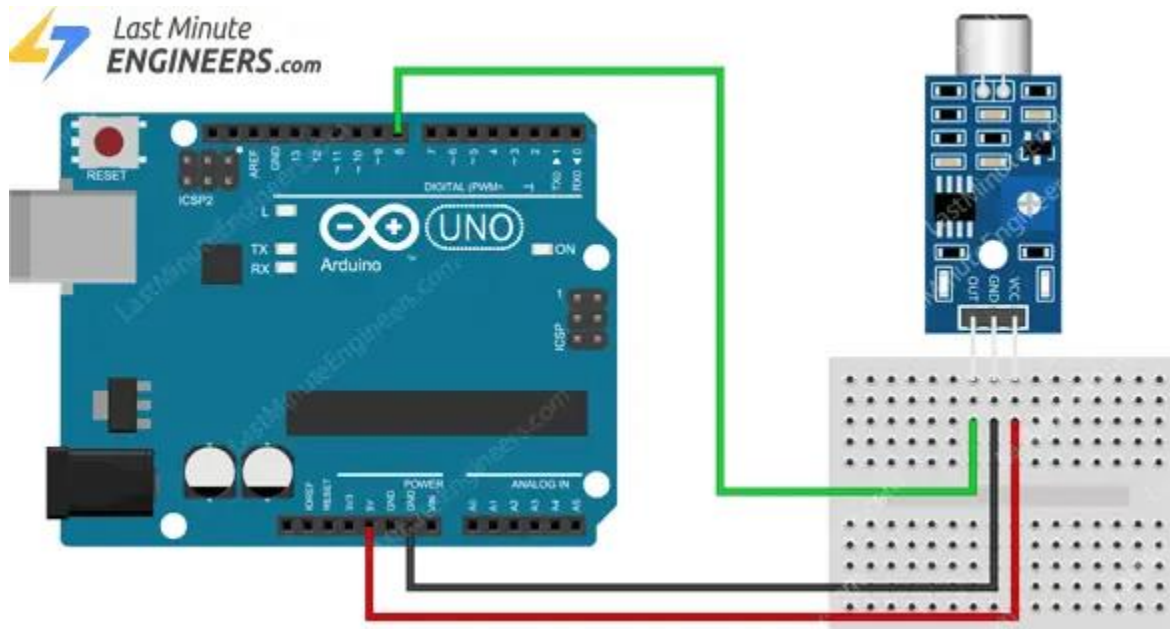
Let's connect the sound sensor to an Arduino. The wiring is quite simple and doesn't take much time.

First, connect the VCC pin on the sound sensor to the 5V pin on the Arduino to supply power. Next, connect the GND pin on the sensor to one of the Arduino's ground (GND) pins. Finally, connect the OUT pin from the sensor to digital pin #8 on the Arduino.

The following table lists the pin connections:

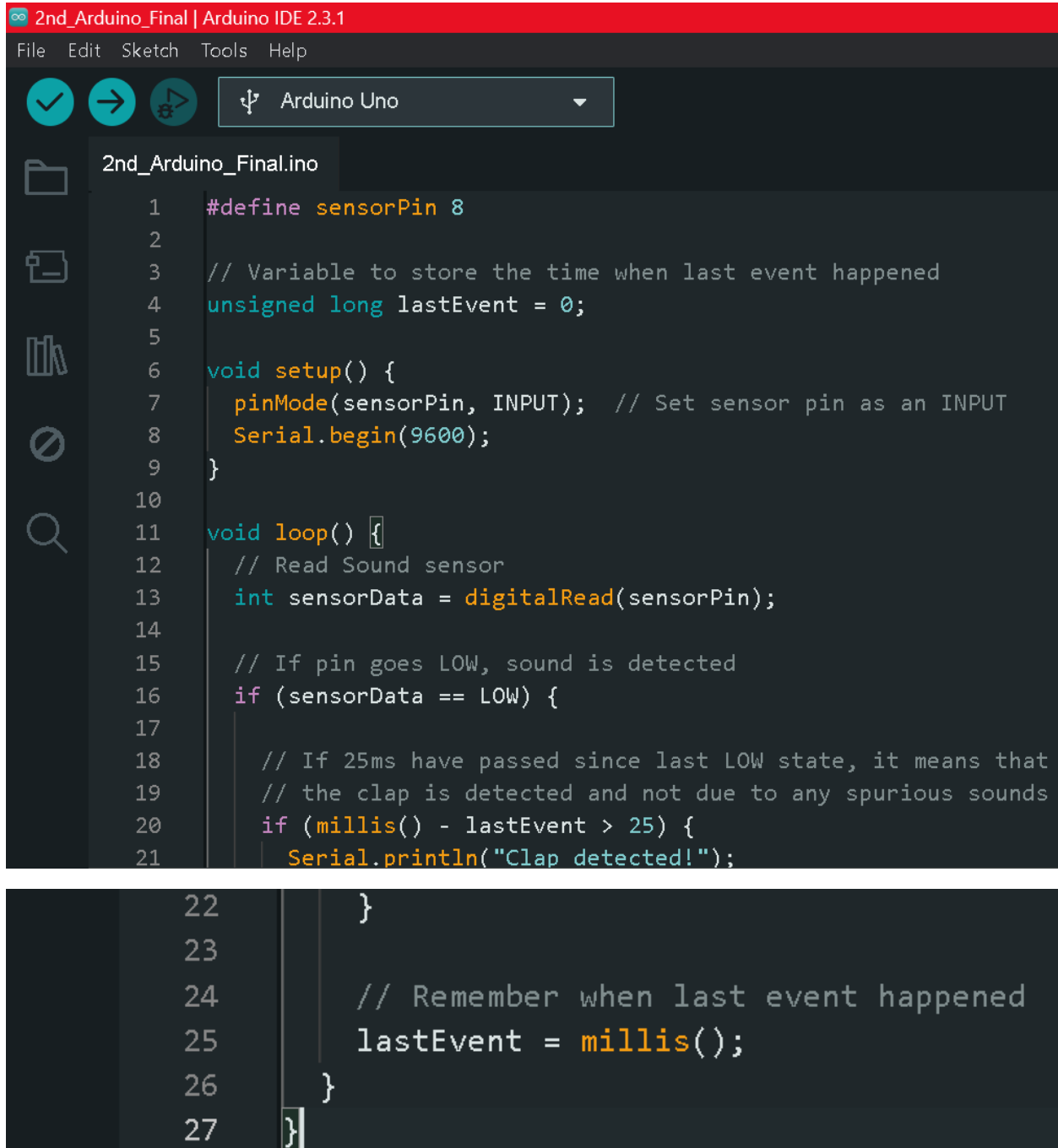
Sound Sensor	Arduino
VCC	5V
GND	GND
OUT	8

The wiring is shown in the image below.



BASIC SOUND DETECTION

In this first example, we'll create a simple project that uses the sound sensor to detect claps or snaps and then display a message on the serial monitor. Give it a try; we'll go over how the code works in a moment.



```
2nd_Arduino_Final | Arduino IDE 2.3.1
File Edit Sketch Tools Help

[Checkmark] [Next] [Previous] [USB] Arduino Uno

2nd_Arduino_Final.ino
1  #define sensorPin 8
2
3  // Variable to store the time when last event happened
4  unsigned long lastEvent = 0;
5
6  void setup() {
7      pinMode(sensorPin, INPUT); // Set sensor pin as an INPUT
8      Serial.begin(9600);
9  }
10
11 void loop() {
12     // Read Sound sensor
13     int sensorData = digitalRead(sensorPin);
14
15     // If pin goes LOW, sound is detected
16     if (sensorData == LOW) {
17
18         // If 25ms have passed since last LOW state, it means that
19         // the clap is detected and not due to any spurious sounds
20         if (millis() - lastEvent > 25) {
21             Serial.println("Clap detected!");
22         }
23
24         // Remember when last event happened
25         lastEvent = millis();
26     }
27 }
```

REFERENCE:

<https://lastminuteengineers.com/sound-sensor-arduino-tutorial/>