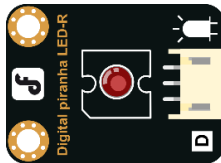


## Project 4: Sound Activated LED

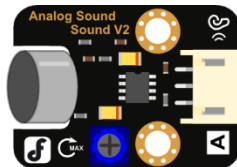
In this session we will make a sound activated LED. You will be able to make an LED flash by clapping your hands. To make it work, we need a sound sensor. We can use a sound sensor to make various interactive and sound control devices besides this experiment.

### COMPONENT LIST:

1× Digital Piranha LED-R



1× Analog Sound Sensor



### HARDWARE CONNECTIONS

Connect the analog sound sensor to analog pin 0

Connect the Digital Piranha LED-R to digital pin 13

Use the diagram below for reference.

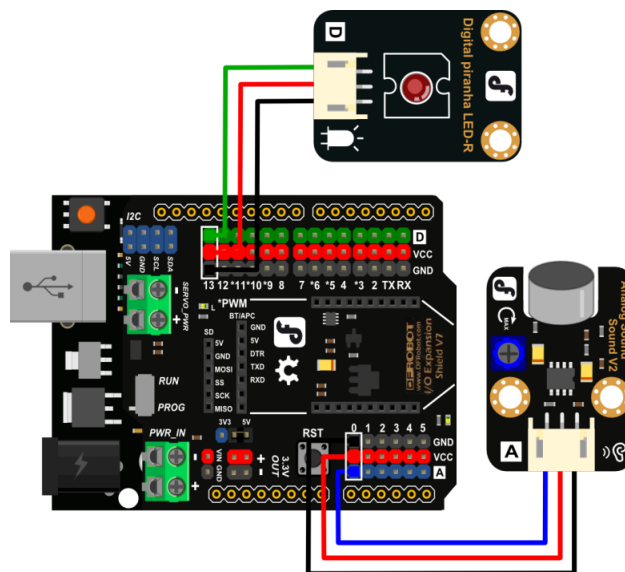


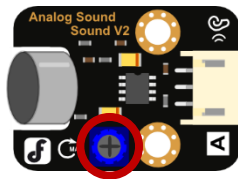
Chart 4-1 Sound Control Light

Be sure that your power, ground and signal connections are correct or you risk damaging your components.

When you have connected the components and checked your connections, plug the Arduino in to your PC with the USB cable so you can upload a program.

## INSTRUCTION

Sometimes the device may not work well in the first beginning, this is because the sensitivity is either set too high or too low, which makes the LED stays on or never lights up. In this case, you shall play around the sensitivity by adjusting the on-board potentiometer till your device works well.



## CODE INPUT

Sample Code 4-1:

```
// Item Four-Sound Control Light

int soundPin = 0; // Analog sound sensor is to be attached to analog pin 0
int ledPin = 13; // Digital Piranha LED-R is to be attached to pin 13.

void setup() {
    pinMode(ledPin, OUTPUT);
    // Serial.begin(9600); // For monitoring
}

void loop(){
    int soundState = analogRead(soundPin); // Read sound sensor's value
    // Serial.println(soundState); // serial port print the sound sensor's value
    // if the sound sensor's value is greater than 10, the light will be on for
    10s. Otherwise, the light will be turned off
    if (soundState > 10) {
        digitalWrite(ledPin, HIGH);
        delay(10000);
    }else{
        digitalWrite(ledPin, LOW);
    }
}
```

Use this sample code to implement the behavior we want.

You can copy and paste it in to the Arduino IDE, but if you want to develop your skills we recommend typing it out.

When you have finished, click “Verify” to check the code for syntax errors. If the code verifies successfully, you can upload it to your Arduino.

Once the code has uploaded, try clapping your hands. The LED should flash on.

When there is no sound, the LED should remain off.

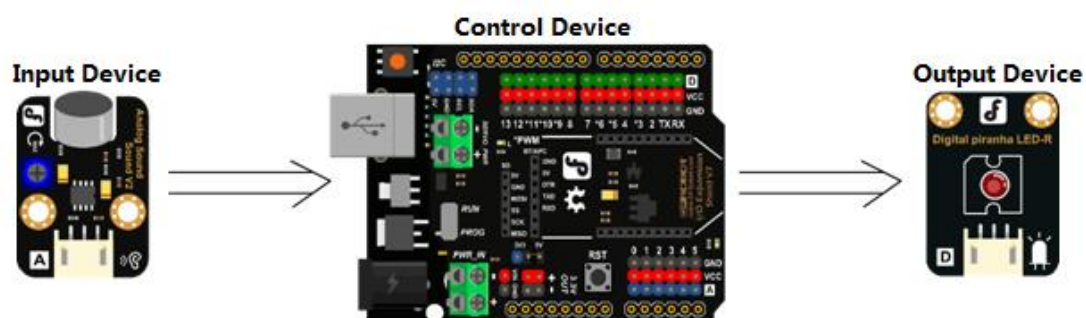
## HARDWARE ANALYSIS (ANALOG INPUT – DIGITAL OUTPUT)

In the previous sessions we have talked about analog and digital signals. The Analog sound sensor is, as the name suggests, an analog sensor.

Digital signals have only two values: 0 and 1

Analog signals have many values, ranging between 0 and 1023

More detailed information can be found in Session 2.



## CODE REVIEW

The `setup()` function defines LED to be in the mode of OUTPUT. Why isn't the sound sensor set up in the mode of INPUT? Because all analog pins are in INPUT mode and we don't need to set it.

The sound sensor is the input unit, and it should read from analog pin 0. The function for this execution is as follows.

```
analogRead(pin)
```

This function is used to read the analog pin. Arduino's analog pin is attached to an A/D converter, so voltage between 0 and 5V will be mapped to a number between 0 and 1023. Each

number collected represents a voltage. For example, 512 = 2.5V

Finally, an "if" statement checks if the result exceeds a certain value.

```
if (soundState > 10) {  
    ...  
}else{  
    ...  
}
```

You can open the serial monitor and test out a threshold value that best represents the sound volume you need. Then you can adjust the value in the code accordingly.