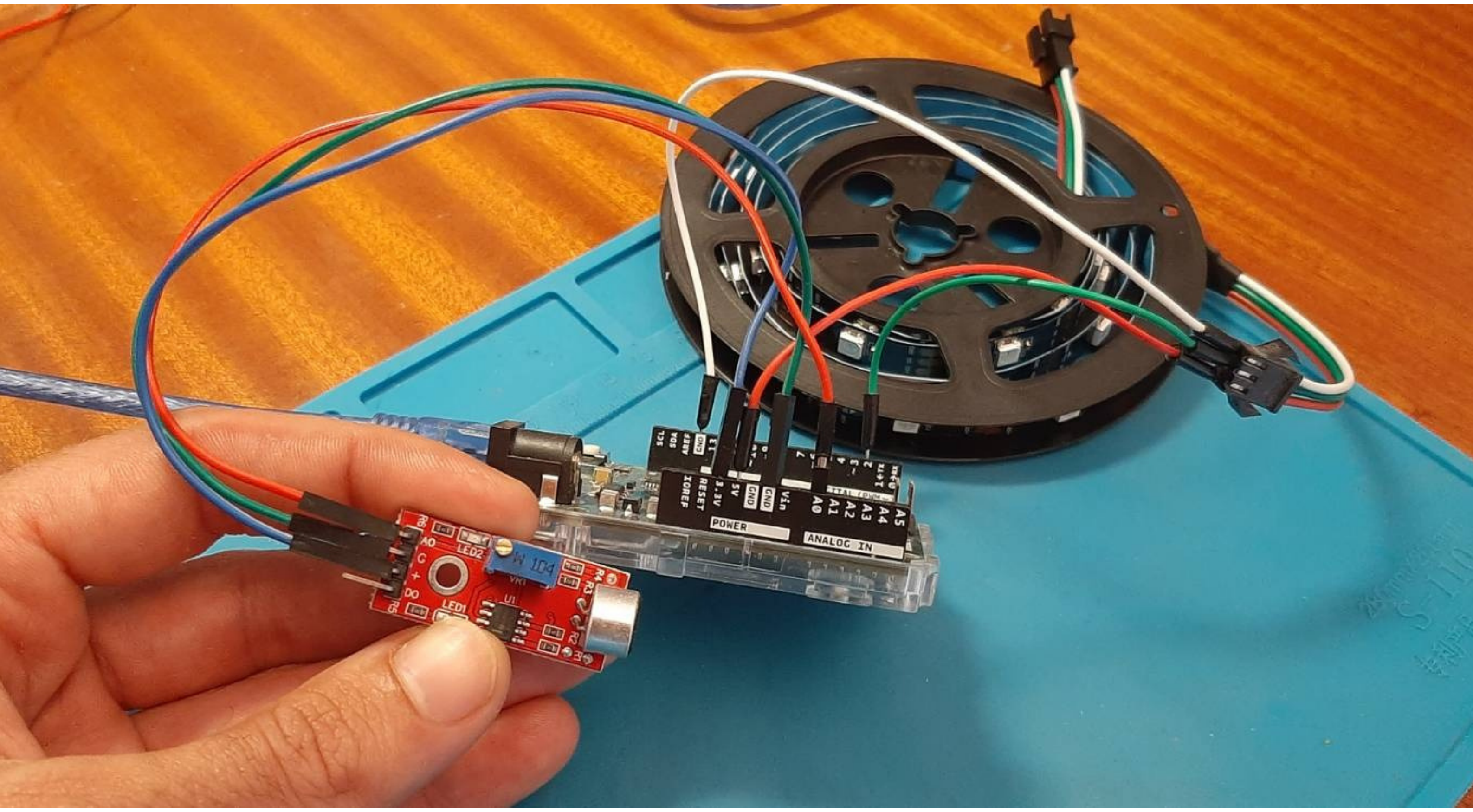
DIGIJEUNES

Dancing LEDS Stripe Tutorial

01 LIST OF COMPONENTS



YOUR TASK

The aim is to build a small circuit with Arduino and make some LEDS Stripes react to a music speaker.

We will use the library FastLED to interact with the stripe and a microphone as input to make the LEDS interact with the music.

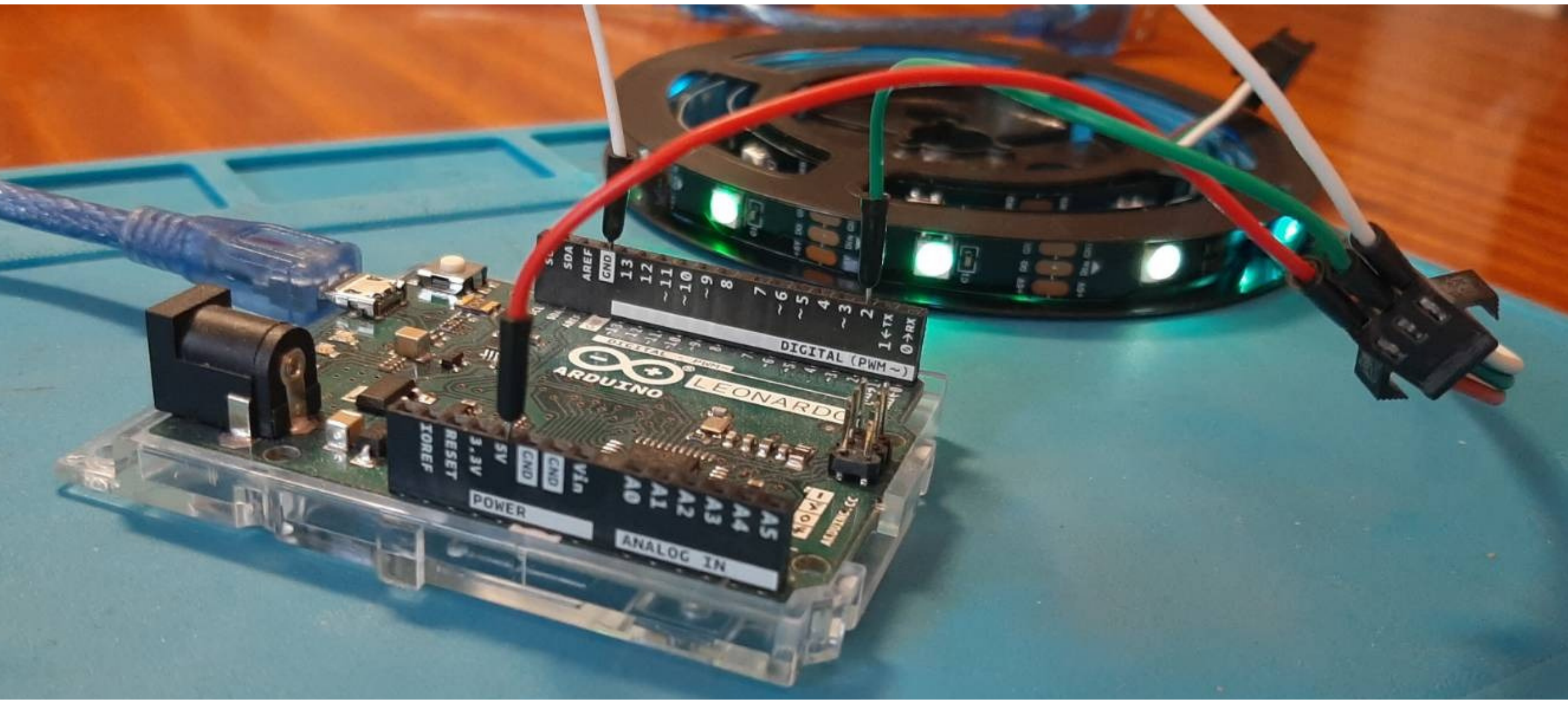
- LEDS Stripes WS2812
- Microphone Sound Sensor
- Arduino Uno
- Speaker with some music
- Jumpers wires

Computer

- Arduino IDE with FastLED library

02 PLUG THE LEDS

Connect the LEDS to the Arduino by connecting ground to ground, 5V to 5V and the middle cable to the digital port 2.

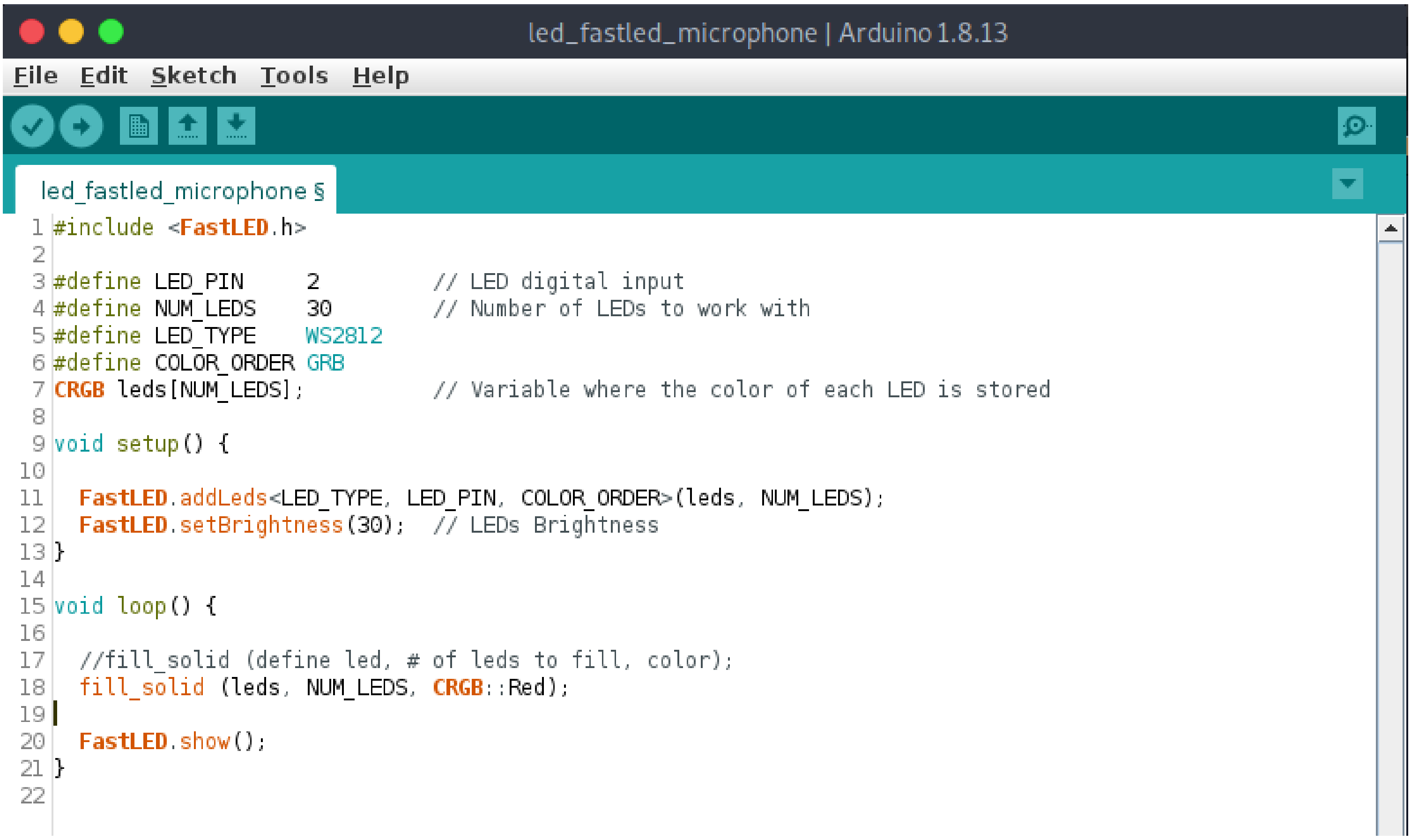


Connect the Arduino to your Computer and open Arduino IDE. You need to install the library FastLED.

In the top menu of Arduino click on **Sketch => Include Library => Manage Libraries**

03 LIGHT UP THE LEDS

Copy this code In Arduino IDE and upload it. Code here



```
 1 #include <FastLED.h>
 2 
 3 #define LED_PIN     2         // LED digital input
 4 #define NUM_LEDS    30        // Number of LEDs to work with
 5 #define LED_TYPE    WS2812
 6 #define COLOR_ORDER GRB
 7 CRGB leds[NUM_LEDS];          // Variable where the color of each LED is stored
 8 
 9 void setup() {
10 
11   FastLED.addLeds<LED_TYPE, LED_PIN, COLOR_ORDER>(leds, NUM_LEDS);
12   FastLED.setBrightness(30);  // LEDs Brightness
13 }
14 
15 void loop() {
16 
17   //fill_solid (define led, # of leds to fill, color);
18   fill_solid (leds, NUM_LEDS, CRGB::Red);
19 
20   FastLED.show();
21 }
22 
```

This code should light up 30 of your LEDs in red. Lets make some modifications to it to understand how it works.

At the **line 18** you can try to change **CRGB::Red** by **Blue** or **Green** and see for yourself what happens by uploading it.

fill_solid() is a function that will give each of the **NUM_LEDS** a solid color (Red in our exemple).

Now write **leds[4] = CRGB::Purple;** after the line 18. You should see the fifth led turn to blue.

As you may have understood, **FastLED.show();** is the function that will give order to your LEDs to light up. It will use what is inside the variable **leds** so everything written after line 18 will override it.

04 USE THE MAGICALS FUNCTIONS

It exist some functions to generate some effect like a rainbow.

Inside the loop, **remove** the code **until FastLED.show();** and put this one:
fill_rainbow (leds, NUM_LEDS, 0, 255/NUM_LEDS);

Let's try some palettes. FastLED have predefined color for you to use.
Write these two following lines at line 8.

#define currentPalette OceanColors_p
#define currentBlending LINEARBLEND

You can use these different palettes :
RainbowColors_p, RainbowStripeColors_p, OceanColors_p,CloudColors_p, LavaColors_p, ForestColors_p, PartyColors_p

Changing **LINEARBLEND** to **NOBLEND** which mix or separate the colors.

MAKING A GRADIENT

Before fastLED.show() write this line:
fill_gradient (leds, 0, CHSV(0, 255, 255), NUM_LEDS, CHSV(255, 255, 255), FORWARD_HUES);

Play with the value of **CHSV(0, 255, 255)** and **CHSV(255, 255, 255)** to change the gradients colors.

Change **FORWARD_HUE** with these value to change the gradient
FORWARD_HUES: hue always goes forwards
BACKWARD_HUES: hue always goes backwards
SHORTEST_HUES: hue goes whichever way is shortest
LONGEST_HUES: hue goes whichever way is longest

05 MAKE THE LIGHT MOVE

Remove the content of the loop() and write this one

```
for (int i=0; i<255; i=i+1) {
    fill_solid (leds, NUM_LEDS, CHSV(i, 255, 255));
    FastLED.show();
    delay(3);
}
```

fill_solid() will be called every 3 milliseconds and will change the color according the variable **i** of the **for**. Going from 0 to 255 and then restart the general loop.

Instead of the **fill_solid()** function let's try to replace it with **fill_palette()**

```
fill_palette (leds, NUM_LEDS, i, 20, currentPalette, 255, currentBlending);
```

Don't forget that you can change the palette for different color effects.

LET'S PUT SOME BEATS

beatsin16(BPM, low value, high value) is a function that will give some ups and downs to a value. It is used to give a pulsing effect. Add this line before fill_palette and change **i** by beat_sin.

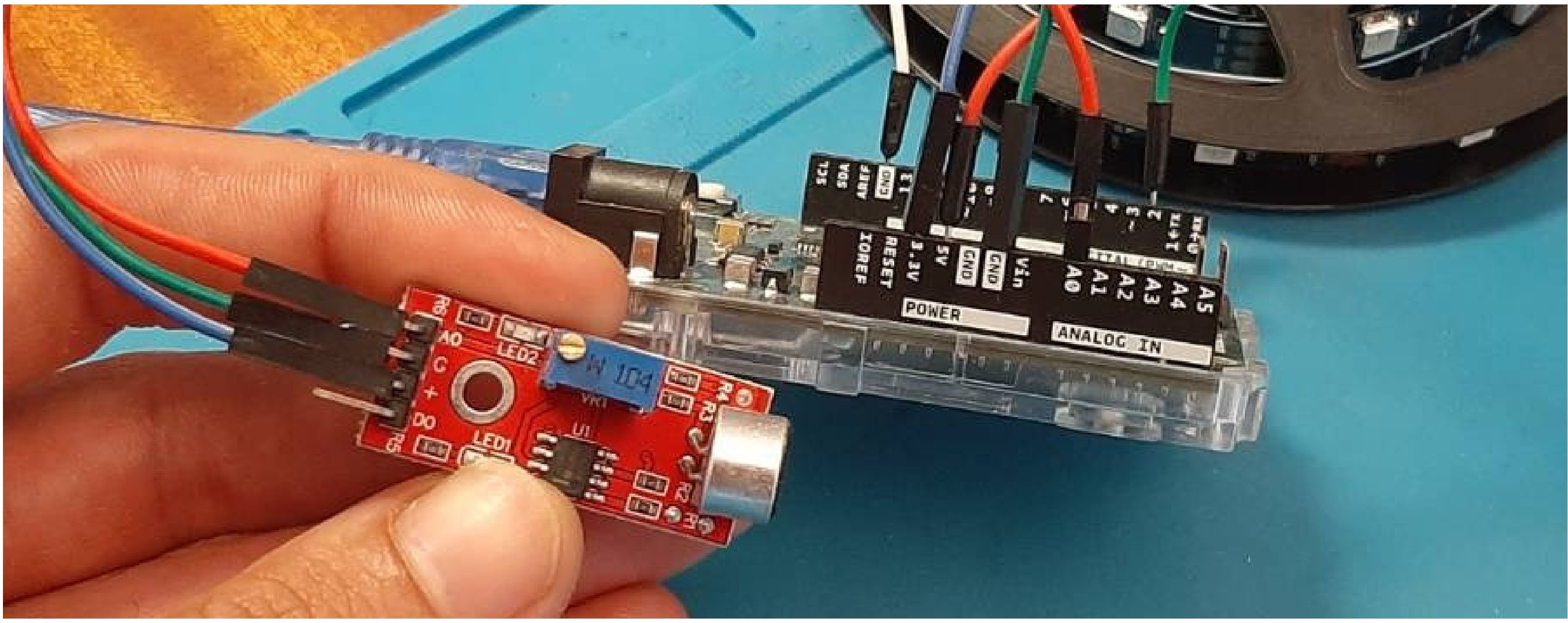
```
int beat_sin = beatsin16(60, 0, 255);
```

Now your leds should beat at 60 BPM right and left !

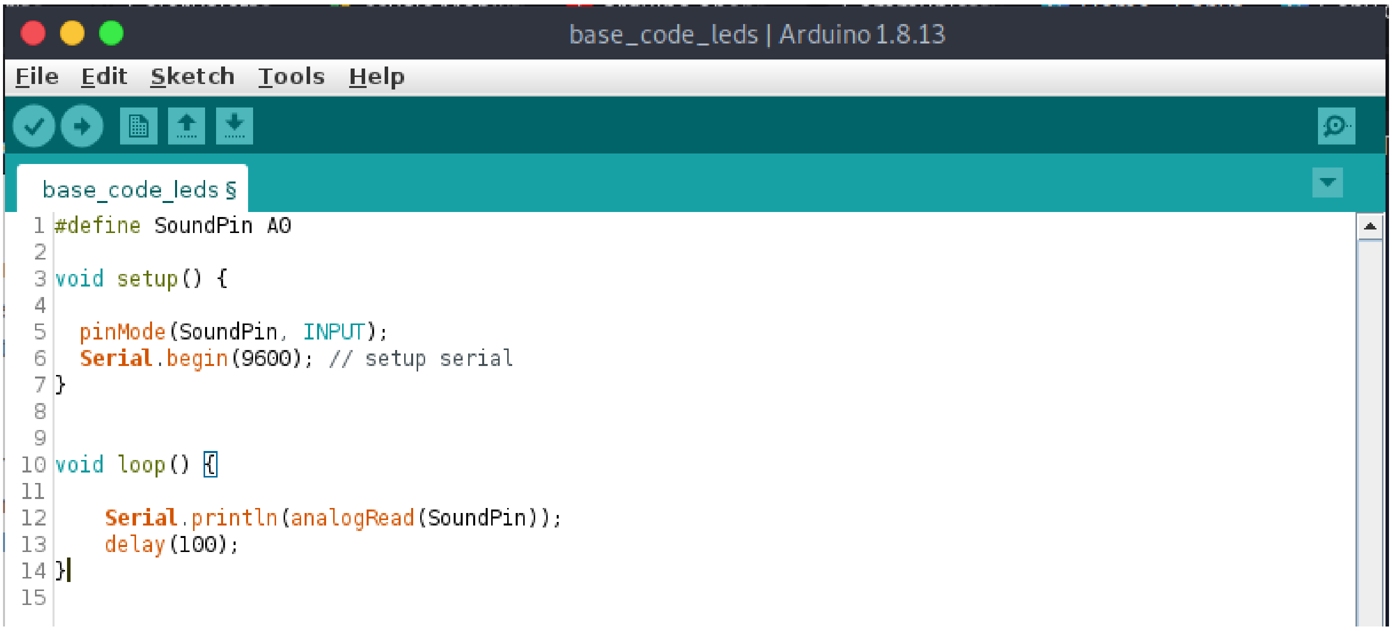
You can find more functions here http://fastled.io/docs/3.1/modules.html

It exist a generator that you can also play with :
https://wokwi.com/arduino/libraries/FastLED/ColorPalette

06 PLUG THE MIC



Place a jumper wire on the **A0** port of your microphone and connect it to **A0** of Arduino. Connect the **G** port to the **ground** port of Arduino. Connect the **+** port to the **5V** of the Arduino.

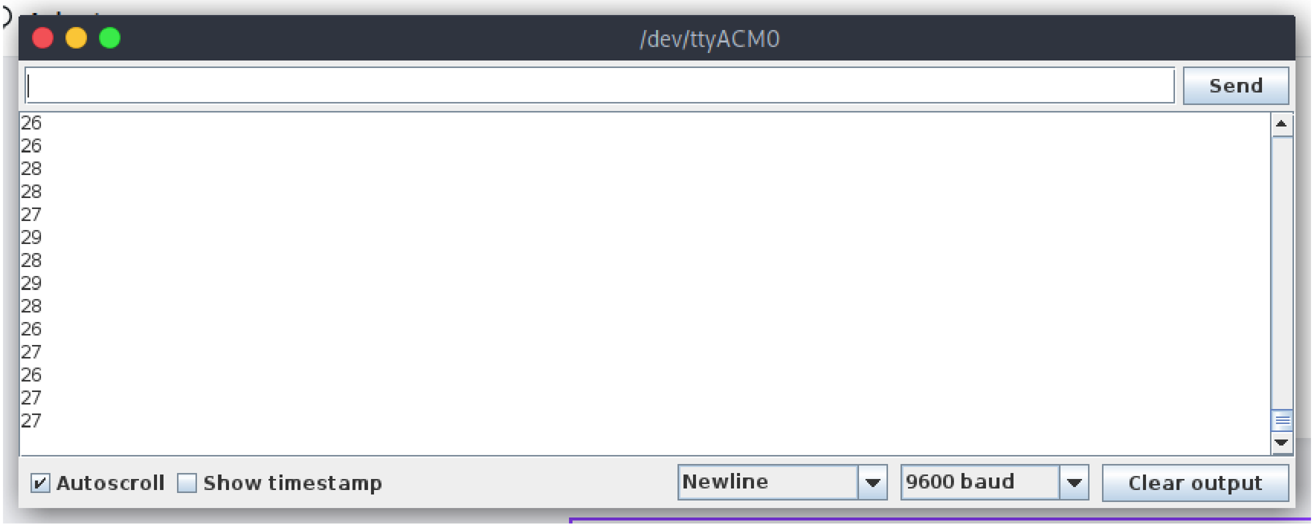


```
#define SoundPin A0

void setup() {

  pinMode(SoundPin, INPUT);
  Serial.begin(9600); // setup serial
}


void loop() {

    Serial.println(analogRead(SoundPin));
    delay(100);
}

```

Replace your code by this code. [You cand find it here.]()

07 USE THE MIC'S OUTPUT

We need to find the values returned by your microphone **in the calm**, to do this upload the previous code and open the **Serial Monitor** in your **tools menu.**



```
26
26
28
28
27
29
28
29
28
26
27
26
27
27
```

I have 29 as strongest value in my calm so I'll use this as my base. Do the same with your own results.

Use the code of the file leds_mic_loud_on_beat **here**.
Line 16 change **LowSoundValue** with your low value you just found.

Start your speaker very close to the microphone and watch it react to your music!

You can play with the variables a little bit but their is not too much flexibility here.

You can now try the code of this file **here** and tweak the variables with what you learned to make the effect of your dreams.

DIGIJEUNES