

## **CS 209-210 Mini Project**

Name- **Biswajit Sarkar**

Roll- **2101AI11**

### **Model:-**

Music Reactive RGB LED Strip using arduino nano

### **OVERVIEW:-**

Light up your new year parties and impress all your friends with these amazing music reactive multicolor LED lights that response and change their colors on every loud beat. These lights are nothing but simple RGB LED strips connected to arduino - the brain of this project. LED strips mean that you can mount them anywhere in your home and even outdoors. The main purpose of this project was to use it in parties but you can also use it for daily purpose for making your music more interesting! Whether you mount in on your entrance door, around your sofa, your LED TV, on your computer desk or walls, that's up to you. The uses of this thing are endless! The only condition is that you have to have an audio output device near your lights to feed them with audio signals which can be processed and reacted upon.

### **OBJECTIVE:-**

In this project I will design a simple cheap sound or music sensitive RGB led strip using simple components. The project is based on Arduino and we have used Arduino Nano for the project.

### **Components Required:-**

1.Sound Sensor



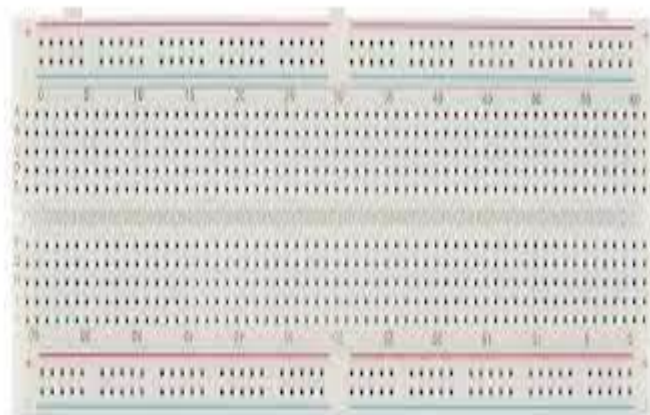
## 2. WS2812B Addressable LED Strip(5V)



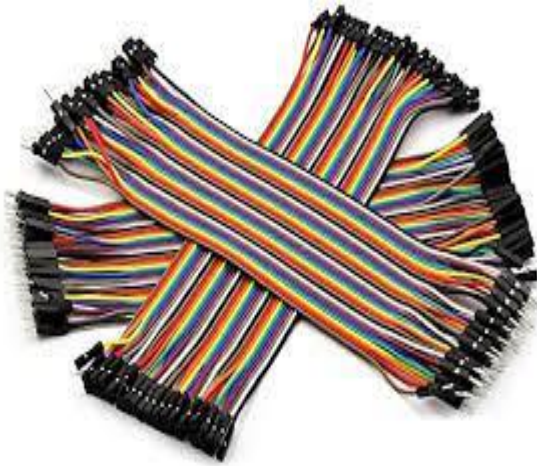
## 3. Arduino Nano



## 4. Bread Board



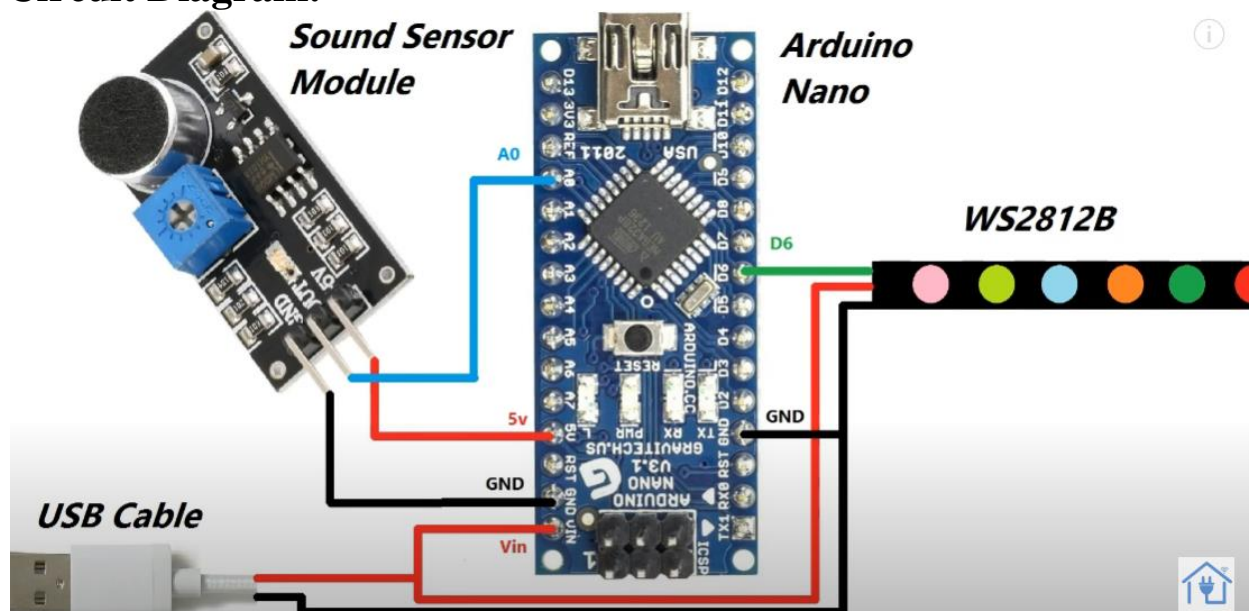
## 5. Jumper Wires



## 6. Power Bank



## Circuit Diagram:-



## Working Model:-

As told above, this project uses an sound sensor for taking the analog audio input and then it convert into digital , and it is connected to arduino which processing it and then giving an output through it's digital pins to the LED strip controller circuit which then drives the strips. It uses a 5v power supply to power for the strip .

As soon as the program is uploaded, the sound sensor start receiving audio signals and then it convert into digital signals then as per the code was uploaded the led lights blinks as per the music beats or sound....

## CODE ANALYSIS:-

**////code starts here////**

```
#include "FastLED.h"
```

```
#define NUM_LEDS 60          // How many leds in your strip?
```

```
#define updateLEDS 8        // How many do you want to update every millisecond?
```

```
#define COLOR_SHIFT 180000  // Time for colours to shift to a new spectrum  
                             In (ms)
```

```
CRGB leds[NUM_LEDS];        // Define the array of leds
```

```
// Define the digital I/O PINS..
```

```
#define DATA_PIN 6          // led data transfer
```

```
#define PITCH_PIN 0          // pitch input from frequency to voltage converter
```

```
#define BRIGHT_PIN 4         // brightness input from amplified audio signal
```

```
// Don't touch these, internal color variation variables
```

```
unsigned long setTime = COLOR_SHIFT;
```

```
int shiftC = 0;
```

```
int mulC = 2;
```

```
// Define color structure for rgb
```

```
struct color {
```

```
    int r;
```

```
    int g;
```

```
    int b;
```

```
};
```

```

typedef struct color Color;

void setup() {
    Serial.begin(9600);
    FastLED.addLeds<NEOPIXEL, DATA_PIN>(leds, NUM_LEDS);
    pinMode(A0, INPUT);
    pinMode(A4, INPUT);

    for(int i = 0; i < NUM_LEDS ; i++) {
        leds[i] = CRGB(0,0,0);
    }
    FastLED.show();
}

void loop() {
    unsigned long time = millis();

    // Shift the color spectrum by 200 on set intervals (setTime)

    if(time / (double)setTime >= 1) {
        setTime = time + COLOR_SHIFT;
        Serial.println(setTime);
        shiftC += 200;
        mulC++;
        if(shiftC >= 600) {
            shiftC = 0;
        }
        if(mulC > 3) {
            mulC = 2;
        }
    }

    // Shift all LEDs to the right by updateLEDS number each time

    for(int i = NUM_LEDS - 1; i >= updateLEDS; i--) {
        leds[i] = leds[i - updateLEDS];
    }

    // Get the pitch and brightness to compute the new color

    int newPitch = (analogRead(PITCH_PIN)*2) + shiftC;
    Color nc = pitchConv(newPitch, analogRead(BRIGHT_PIN));

    // Set the left most updateLEDS with the new color

```

```

for(int i = 0; i < updateLEDS; i++) {
    leds[i] = CRGB(nc.r, nc.g, nc.b);
}
FastLED.show();

//printColor(nc);
delay(1);
}

/**
 * Converts the analog brightness reading into a percentage
 * 100% brightness is 614.. about 3 volts based on frequency to voltage converter circuit
 * The resulting percentage can simply be multiplied on the rgb values when setting our colors,
 * for example black is (0,0,0) so when volume is off we get 0v and all colors are black (leds are
 off)
 */

double convBrightness(int b) {
    double c = b / 614.0000;
    if( c < 0.2 ) {
        c = 0;
    }
    else if(c > 1) {
        c = 1.00;
    }
    return c;
}

/**
 * Creates a new color from pitch and brightness readings
 * int p    analogRead(pitch) representing the voltage between 0 and 5 volts
 * double b  analogRead(brightness) representing volume of music for LED brightness
 * returns Color structure with rgb values, which appear synced to the music
 */

Color pitchConv(int p, int b) {
    Color c;
    double bright = convBrightness(b);

    if(p < 40) {
        setColor(&c, 255, 0, 0);
    }
    else if(p >= 40 && p <= 77) {

```

```

    int b = (p - 40) * (255/37.0000);
    setColor(&c, 255, 0, b);
}
else if(p > 77 && p <= 205) {
    int r = 255 - ((p - 78) * 2);
    setColor(&c, r, 0, 255);
}
else if(p >= 206 && p <= 238) {
    int g = (p - 206) * (255/32.0000);
    setColor(&c, 0, g, 255);
}
else if(p <= 239 && p <= 250) {
    int r = (p - 239) * (255/11.0000);
    setColor(&c, r, 255, 255);
}
else if(p >= 251 && p <= 270) {
    setColor(&c, 255, 255, 255);
}
else if(p >= 271 && p <= 398) {
    int rb = 255-((p-271)*2);
    setColor(&c, rb, 255, rb);
}
else if(p >= 398 && p <= 653) {
    setColor(&c, 0, 255-(p-398), (p-398));
}
else {
    setColor(&c, 255, 0, 0);
}
setColor(&c, c.r * bright, c.g * bright, c.b * bright);
return c;
}

```

```

void setColor(Color *c, int r, int g, int b) {
    c->r = r;
    c->g = g;
    c->b = b;
}

```

// Prints color structure data

```

void printColor(Color c) {
    Serial.print("(");
    Serial.print(c.r);
    Serial.print(", ");

```

```
Serial.print(c.g);  
Serial.print(", ");  
Serial.print(c.b);  
Serial.println(" ");  
}
```

**//code ends here**

## **Result-**

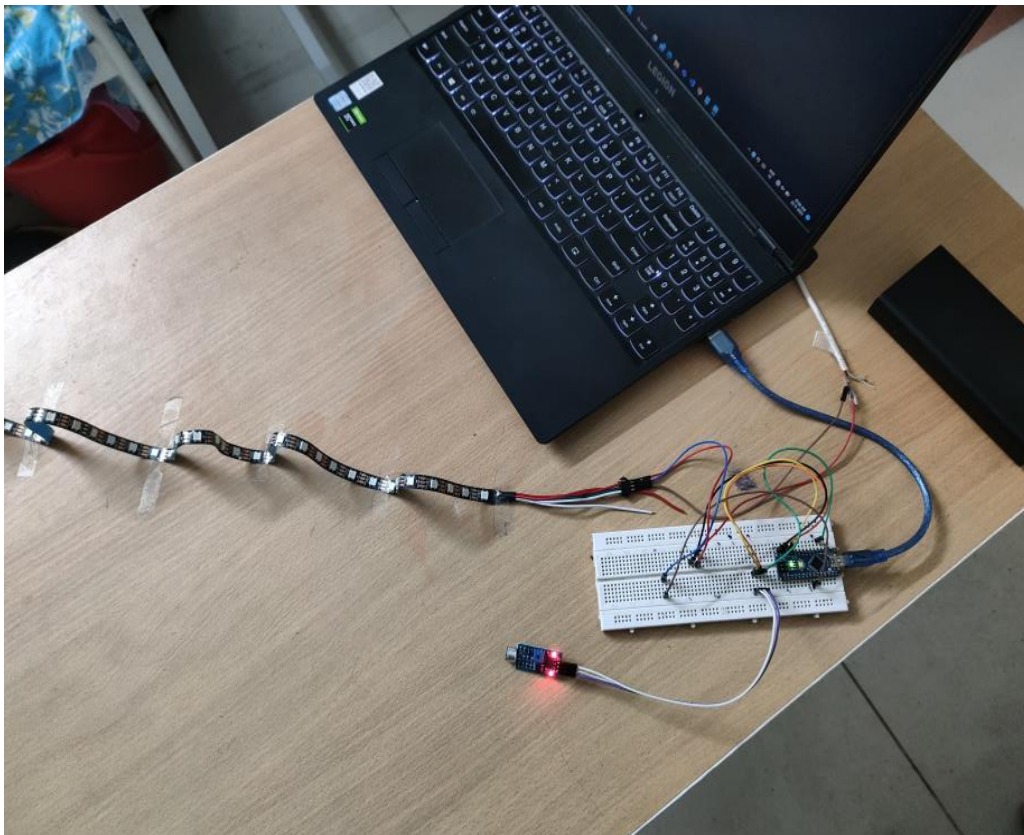
The video of working of sound reactive led strip is included in the zip file.

**Project Video Demonstration Link :**

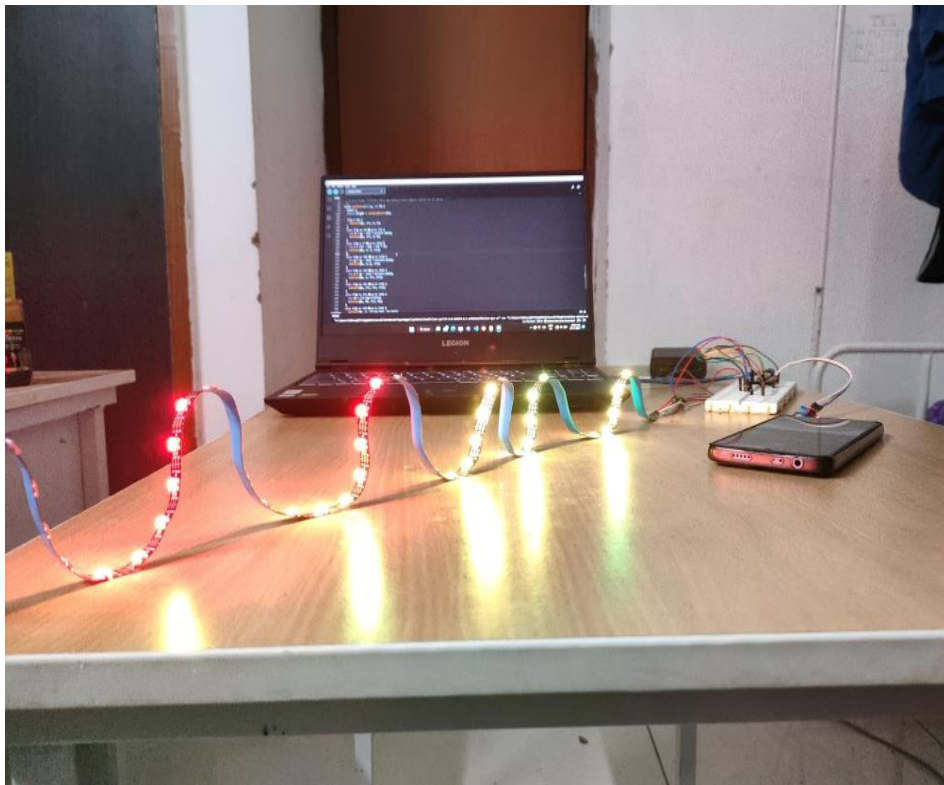
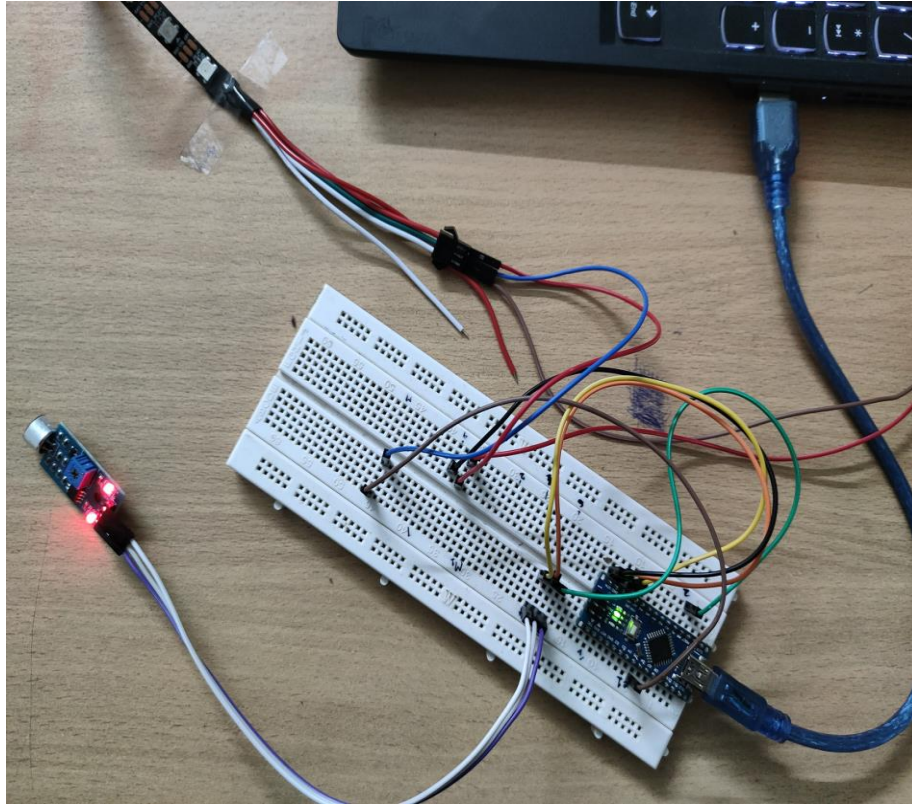
YouTube link:- <https://www.youtube.com/watch?v=jkW4Po9quac>

Drive link:- <https://drive.google.com/file/d/18jFxK21WJlYzyPJUEdKLglHybYyDcwto/view?usp=sharing>

## **Project Images:-**







## **References:-**

- <https://create.arduino.cc/projecthub/MaheshBHISHNOI/how-to-make-diy-music-reactive-rgb-led-strip-ws2812b-0dbaf4>
- <https://www.instructables.com/Sound-Reactive-LED-strip/#:~:text=This%20module%20has%20a%20dial,average%20sounds%20loud%20claps%2C%20shouts%2C>
- <https://www.youtube.com/watch?v=NeTVxjG42tk&t=157s>
- [https://create.arduino.cc/projecthub/diyprojectslab/diy-music-relative-rgb-strip-light-5ca700?ref=similar&ref\\_id=376065&offset=1](https://create.arduino.cc/projecthub/diyprojectslab/diy-music-relative-rgb-strip-light-5ca700?ref=similar&ref_id=376065&offset=1)

# THANK YOU!!