Term project on

Sound sensing RGB music sync light

Course: Operating System Design (CSE323)

Faculty: Saeed Mahmud Ullah (Smu1)

Submitted By:

Name: Chowdhury Nafis Faiyaz

ID:1931841642

Section: 5

Group:4

Group mates:

Sujana Rahman Khan

Md Shadman Zarif

Shajeda Parvin

Date: 25th August 2022

Sound sensing RGB music sync light

Project overview:-

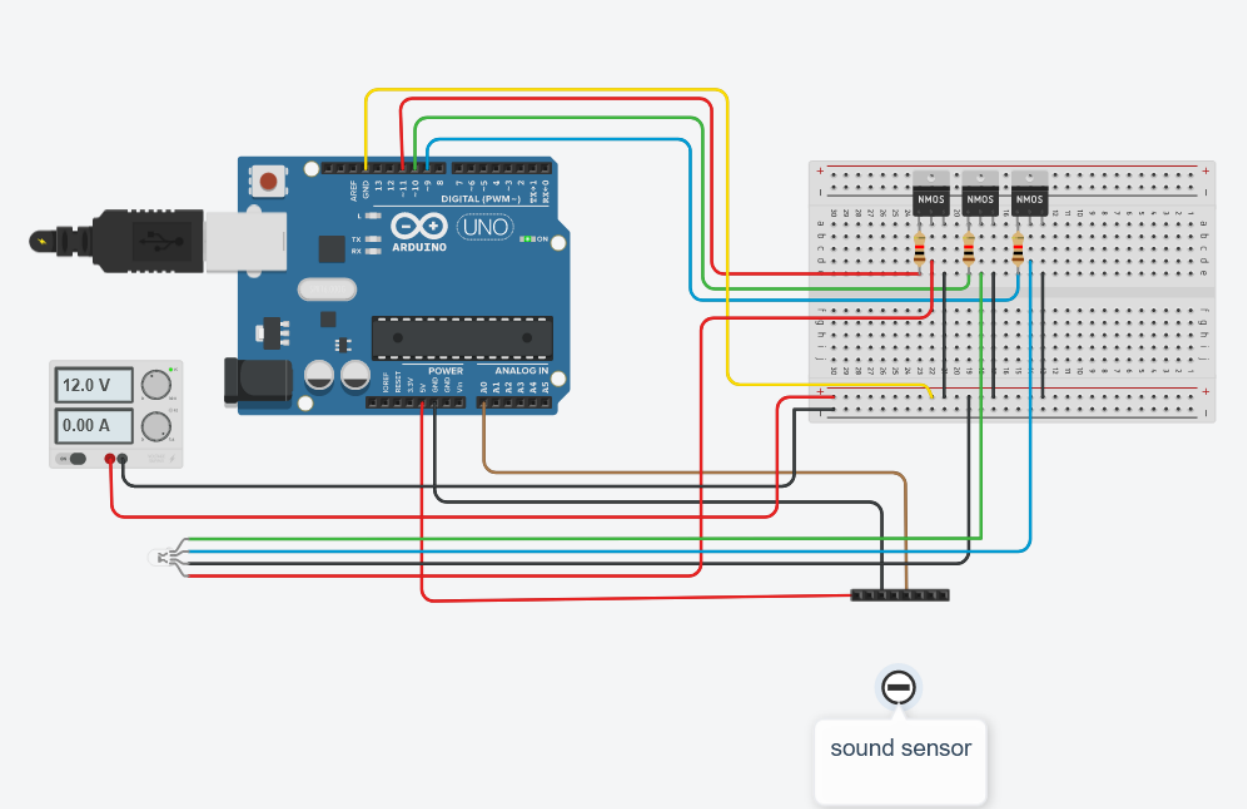
In this project we have tried to make a sound sensing RGB music sync light wihich is controlled by a Arduino Uno microcontroller. The main objective of the project Is that it will take analogue sound signals via the microphone of the sound sensor. The signals will be fed into the microcontroller chip. As the sound travels as wave and has certain frequencies, the signals will have varying frequencies, and the color of the RGB strip light will change accordingly. The change in the color of the RGB light is controlled by a set of conditional statement. The micro controller computes the frequency of sound, compares it with the given code and changes the color of light accordingly. Hence we can say that the micro processor is interrupted by the I/O input which is the different range of frequencies. And after computing the input frequency value it changes the color of the light accordingly. This is run in a loop until the varying sound source stops. Our code contains a number of functions which does the following work:

1. FilterSignal(float sensorSignal)- This function is used to change the value of the signalFiltered variable.
2. CompareSignalFiltered(float signalFiltered)- This function defines the color of the LEDS by comparing
3. RGBColor(int redColor, int greenColor, int blueColor)- This is the function which is actually switching the color of the leds through arduino pin
4. MainFunction()- contains a variable named MicValue which converts the analogue mic frequency value to a voltage value ranging from 0V-5V. Then the FilterSignal method is called which is used to change the value of the singnal filtered variable(micValue). And in the end the CompareSignalFiltered function is called to compare the input signal and display the necessary color. The main function is run in a loop in the arduino IDE.

Apparatus used for this project:-

1. Arduino UNO
2. TIP41C transistors x3
3. 1kohm resistors x3
4. 12v RGB stripe light
5. Sound sensor
6. 12v AC adapter power source

Working setup diagram:-



Problems faced during the project:-

During the initial setup, the arduino uno was not being detected by the computer, hence it was not accessible. In the windows device manager it showed that the device is not recognized. This problem arised due to a driver error. The driver that was needed to run the Arduino UNO microcontroller wasn not present in the computer. To solve this problem a “CH340” driver was installed.

The sound sensor was not accutate detecting sounds form a distance. Hence it is needed to make sure that the sound source is held close to the mic of the sound sensor. As the sound sensor is not highly sensitive to sound change hence sometimes it lags to server the change in frequency timely to the microprocessor

Code:-

```
/*
CSE323 Project
Section: 5
Group: 4
Summer 2022
Project Name: Music Controlled Lighting System
*/
#define redPin 11
#define greenPin 10
#define bluePin 9
#define ledDelay 3
#define micPin A0

float micValue = 0, signalFiltered = 0, signalFilteredValues[] = {3.4, 3.1, 2.7, 2.4,
2.1, 1.7, 1.3, 0.9, 0.4};

void setup()
{
  Serial.begin(9600); // This starts the arduinos communication
}
void loop()
{
  MainFunction();
}
void MainFunction()
{
  micValue = (float)analogRead(micPin) * (5.0 / 1024.0); // micPin value is 0 to 1023
& then we convert the value to volts ranging 0 to 5.

  FilterSignal(micValue); // This function is used to change the value of the
signalFiltered variable

  CompareSignalFiltered(signalFiltered); // This function compares the singal value
in order to choose the colour of the light
}

void FilterSignal(float sensorSignal) // This function is used to change the value of
the signalFiltered variable
{
  signalFiltered = (0.945 * signalFiltered) + (0.0549 * sensorSignal);
}

void CompareSignalFiltered(float signalFiltered) // This function defines the color
of the LEDS by comparing
{

  if (signalFiltered > signalFilteredValues[0])
  {
    RGBColor(0, 0, 255); // Blue
  }
  else if (signalFiltered <= signalFilteredValues[0] && signalFiltered >
signalFilteredValues[1])
  {
```

```
    RGBColor(0, 255, 255); // Azure
  }
  else if (signalFiltered <= signalFilteredValues[1] && signalFiltered >
signalFilteredValues[2])
  {
    RGBColor(0, 127, 255); // Cyan
  }
  else if (signalFiltered <= signalFilteredValues[2] && signalFiltered >
signalFilteredValues[3])
  {
    RGBColor(0, 255, 127); // Aqua Marine
  }
  else if (signalFiltered <= signalFilteredValues[3] && signalFiltered >
signalFilteredValues[4])
  {
    RGBColor(0, 255, 0); // Green
  }
  else if (signalFiltered <= signalFilteredValues[4] && signalFiltered >
signalFilteredValues[5])
  {
    RGBColor(255, 255, 0); // Yellow
  }
  else if (signalFiltered <= signalFilteredValues[5] && signalFiltered >
signalFilteredValues[6])
  {
    RGBColor(255, 0, 255); // Magenta
  }
  else if (signalFiltered <= signalFilteredValues[6] && signalFiltered >
signalFilteredValues[7])
  {
    RGBColor(255, 0, 127); // Rose
  }
  else if (signalFiltered <= signalFilteredValues[7] && signalFiltered >
signalFilteredValues[8])
  {
    RGBColor(255, 127, 0); // Orange
  }
  else if (signalFiltered <= signalFilteredValues[8])
  {
    RGBColor(255, 0, 0); // Red
  }
  else
  {
    RGBColor(0, 0, 255); // Default:Blue
  }
}

void RGBColor(int redColor, int greenColor, int blueColor) // This is the function
which is actually switching the color of the leds through arduino pin
{
  analogWrite(redPin, redColor);
  analogWrite(greenPin, greenColor);
  analogWrite(bluePin, blueColor);
  delay(ledDelay); // Delays the light
}
```