Matthew Kim

PY371 Final Project Proposal

Obelisk

Abstract

My project uses a few parts and an Arduino to create an “obelisk” of lights that reacts dynamically to music. The MAX9814 microphone component receives sound (typically music) and feeds it into the Arduino, which processes the sound. The entire system is powered by an external AC/DC power supply which is plugged into the wall. Then, the Arduino displays the data on a meter-length strip of individually addressable RGBW LEDs, the SK6812. This results in a vibrant display of colors, which can either be static or dynamic; the static modes are invariant to sound, while the dynamic modes react directly to the sound, according to its volume and frequency. The user can toggle through the various modes with a pushbutton. Some modes include: rainbow fading, pulsing to music, bouncing LEDs, and a lava lamp.

Concepts/Technique

The MAX9814 has a nifty auto-adjustable gain control microphone, which is handy for any type of external speaker that is liable to move in position relative to the microphone. It receives sounds and amplifies it with the op amp housed in its chip. Then, it feeds this into the Arduino where the analog signals are converted to digital signals. The Arduino sends the now-digital data into the SK6812 strip in a manner specific to each function, which is then translated into the LEDs lighting up different colors. A resistor and blocking capacitor are included in the system to make the whole process work; the blocking capacitor ensures the first pixel is not overrun with

current. The pushbutton interrupts the signal passing between the Arduino and LEDs to give us our desired mode changes. Ideally, everything would be soldered, but due to the circumstances everything will have to be housed on a breadboard, which raises some concerns about the function of the system.

Steps

1) Wire the Arduino, SK6182, external power supply, capacitor, and resistor together. The external power supply powers the Arduino and LED strip with +5V; the power supply and SK6182 ground goes to the Arduino ground. The capacitor goes between the power supply and the Arduino and SK6182. The Arduino digital pin goes into the Din pin of the SK6182 and there is a 470 Ohm resistor placed in series with this connection.

2) Wire the Arduino, MAX9184, and pushbutton. The MAX9184 has 5 pins, but 3 are the necessary ones to connect. The MAX9184's Vdd is connected to the Arduino's +3.3V, the Out pin is connected to any analog pin, and the GND pin is connected to Arduino ground. The pushbutton is connected to an Arduino digital pin and the Arduino ground.

3) Now, my project is all wired together. I have to begin the code through the Arduino IDE. It involves taking inputs from the MAX9184 and sending them as bits through the Arduino into the LED, where each pixel will strip off a section of the bit string before sending it to the rest of the pixels. The result will be a sound-responsive display of lights on the LED strip.

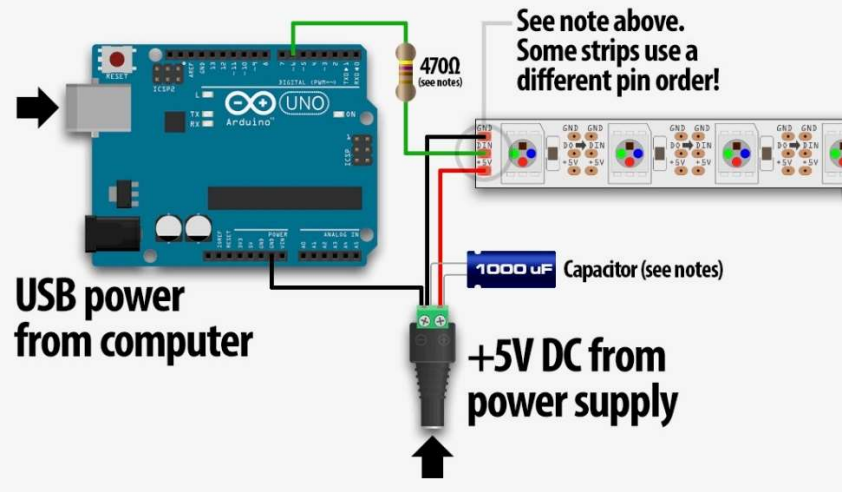
4) Test my project and make sure it works!

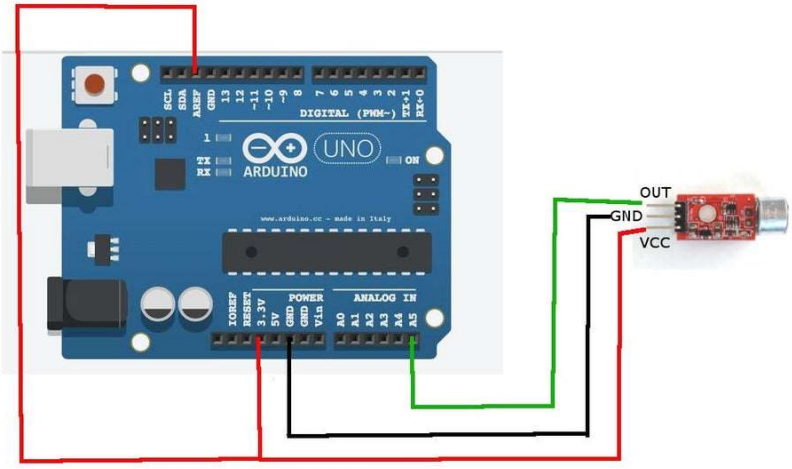
Optional Steps

5) Enclose the LED strip in a diffusing material for aesthetic effect

6) Add more functions to the LED display (static and dynamic both)

Schematics/Diagrams





(The pinout for MAX9184 is the same, while the two extra pins are for the Attack/Release ratio and for different dB gains. The Out, Vcc, and GND pinouts are the same)

Parts Required

- Arduino Uno
- IRF520 MOSFET
- 1N4001 Diode
- 1000 uF Capacitor
- (2) 470 Ohm Resistor
- MAX9814 [Electret Microphone Amplifier]
    - https://www.adafruit.com/product/1713
- SK6812 [RGBW]
    - https://www.btf-lighting.com/products/1-sk6812-rgbw-4-in-1-pixels-individual-addressable-led-strip-dc5v?variant=25757793419364
- AC to DC Converter and Power Jack
    - https://www.amazon.com/ALITOVE-Converter-5-5x2-1mm-100V-240V-Security/dp/B078RT3ZPS/ref=sr_1_3?dchild=1&keywords=5v+4+amp+power+supply&qid=1586589525&sr=8-3
- Pushbutton
- (15) Assorted AWG Wires