Elizabeth Brown

Carter

Final Write-up

29 April 2019

Arduino Neopixel Project

My project is a Neopixel LED lamp controlled by an Arduino UNO. The lamp I made has two functions. First, it can cycle through different colors and patterns. The code I used, originally by Matt Nupen, creates a soft rainbow gradient. Secondly, my lamp is connected to a sound sensor. I found and altered a code by PrinceTronics that makes the lights react to music; different sound frequencies show different colors. My goal with this project was to become more acquainted with Arduino, coding, and electronics. In order to create the lamp, I had to do extensive research into creating the circuit, knowing what components to buy, and understanding C++ coding syntax.

Arduino is an open source electronics and software platform. In class, I was introduced to them with the Arduino UNO, a microcontroller board. In my project, I used the UNO to control the colors and patterns displayed on the LED strip. The LED strip utilizes special LED's called Neopixels. Neopixel LED strips are special because each RGB LED is individually addressable. This means that you can have several colors going at a time, whereas typical RGB LED's can only display one color at a time because they are connected in a series.

During the research process, I accessed several Arduino related blogs and YouTube tutorials. No project was exactly like mine, so I knew I would have to gather a lot of information. These sources helped me understand what components I needed, how to use them, and how to code for the Neopixels and sound sensor.

In order to complete this project, I had several problems to solve. At first, I was not sure what kind of LED strip to order. I just knew I wanted to control RGB LED's with sound. It was only when I accidently ordered a 12-volt Neopixel LED strip, that I decided to use them. Originally, I was going to use a 5-volt simple LED strip, so ordering this part incorrectly posed new issues. Not only were the MOSFET's I ordered useless, but I needed to somehow connect a 12-volt power source to the Nonpixel strip without blowing up the 5-volt Arduino board. This also made coding a more difficult task. Neopixel coding is way more complex than the typical LED strip.

To address these problems, I found a 12-volt car battery charger in my garage and used it as a power source. With jumper wires, the GND and positive wires of my LED strip were connected to the power source. I had to include a 1000-uf capacitor, so that the circuit was not overloaded with energy. With my Arduino, I connected the data wire of my LED to pin 5, with the addition of a 550-ohm resistor. Lastly, I had to connect the GND's of the Neopixels and Arduino. The data line, which controls color and patterns, only requires 5-volts. This let me control the LED's with the Arduino, but light them with the power source.

After I finished making the circuit, I had to worry about coding. I used the Arduino IDE for this. From my research, I discovered that with Neopixels you must include one of two libraries: FastLED or Adafruit Neopixel. I experimented with code from each library. Personally, I prefer Adafruit Neopixel because the syntax is slightly simpler. I was not able to completely

write the code by myself due to its complexity, so I played around with Matt Nupen's transitioning rainbow code. I changed the pin numbers, LED's brightness, and color pattern of his code.

Another issue I had was incorporating the sound sensor into my project. Attaching the sensor to the Arduino was relatively simple. However, coding for it proved difficult. I was able to find a code by PrinceTronics, which made the Neopixels react to different sound frequencies, but it required heavy alterations. The original code did not work well with my sound sensor because it was too sensitive to background noises. The original brightness was also an eye sore. I had to change the delay, slope of sound samples, number of LED's, and brightness. One last thing I wanted to do was make the lights clap on and off. With some code splicing, I was able to make the lights pause with a clap, but not turn off. This was quite frustrating until I did more research. I discovered that clap switches require a relay, which I did not have.

Schematics:

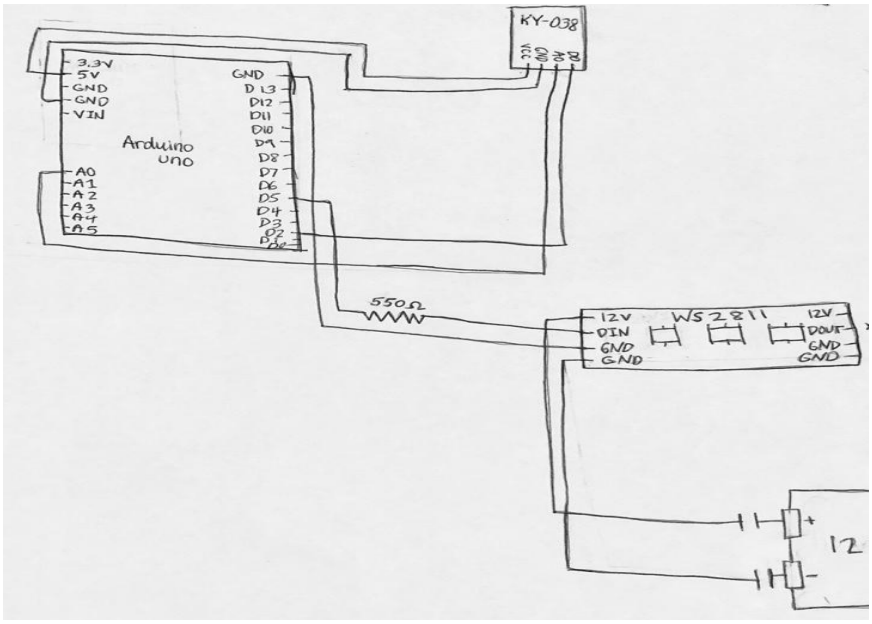

FIGURE 1 HAND DRAWN CIRCUIT SCHEMATICS BY ME

Progress Photos:

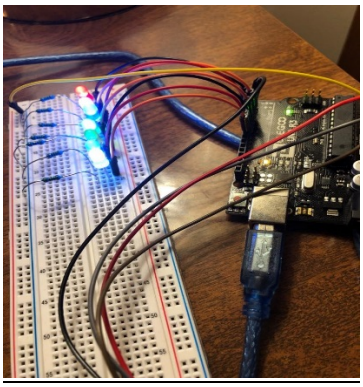

FIGURE 2 LED CLAP CIRCUIT

FIGURE 3 GETTING NEOPIXELS TO LIGHT UP

FIGURE 4 MAKING LAM

Code Examples:

sound_sensor_sketch

```
#include <Adafruit_NeoPixel.h>

#define N_PIXELS  50  // Number of pixels you are using
#define MIC_PIN   A0  // Microphone is attached to Trinket GPIO #2/Gemma D2 (A1)
#define LED_PIN    5  // NeoPixel LED strand is connected to GPIO #0 / D0
#define DC_OFFSET  0  // DC offset in mic signal - if unusure, leave 0
#define NOISE     100  // Noise/hum/interference in mic signal
#define SAMPLES   80  // Length of buffer for dynamic level adjustment
#define TOP       (N_PIXELS +1) // Allow dot to go slightly off scale

byte
  peak      = 0,      // Used for falling dot
  dotCount  = 0,      // Frame counter for delaying dot-falling speed
  volCount  = 0;      // Frame counter for storing past volume data

int
  vol[SAMPLES],       // Collection of prior volume samples
  lvl       = 10,     // Current "dampened" audio level
  minLvlAvg = 0,      // For dynamic adjustment of graph low & high
  maxLvlAvg = 512;

Adafruit_NeoPixel  strip = Adafruit_NeoPixel(N_PIXELS, LED_PIN, NEO_GRB + NEO_KHZ800);

void setup() {
  memset(vol, 0, sizeof(vol));
  strip.begin();
}
void loop() {
  uint8_t  i;
  uint16_t minLvl, maxLvl;
  int      n, height;
  n   = analogRead(MIC_PIN);                 // Raw reading from mic
  n   = abs(n - 512 - DC_OFFSET);            // Center on zero
  n   = (n <= NOISE) ? 0 : (n - NOISE);      // Remove noise/hum
  lvl = ((lvl * 7) + n) >> 3;    // "Dampened" reading (else looks twitchy)
```

FIGURE 5 EXAMPLE OF PRINCETRONICS NEOPIXEL SOUND SENSOR CODE EDITED BY ME

Rainbow_Sketch

```
#include <Adafruit_NeoPixel.h>

// constants won't change. They're used here to
// set pin numbers:
const int ledPin = 5;      // the number of the neopixel strip
const int numLeds = 50;


//Adafruit_NeoPixel pixels = Adafruit_NeoPixel(50, ledPin);
Adafruit_NeoPixel strip = Adafruit_NeoPixel(numLeds, ledPin, NEO_GRB + NEO_KHZ800);


void setup() {
  strip.begin();
  strip.setBrightness(80); // 1/3 brightness

}

void loop()

{
    rainbow(30);

  delay(10);
}


void rainbow(uint8_t wait) {
  uint16_t i, j;
```

FIGURE 6 EXAMPLE OF MATT NUPEN'S NEOPIXEL RAINBOW CODE EDITED BY ME

Final Prototype:

FIGURE 7 FINAL PROTOTYPE MUSICAL LAMP FUNCTION

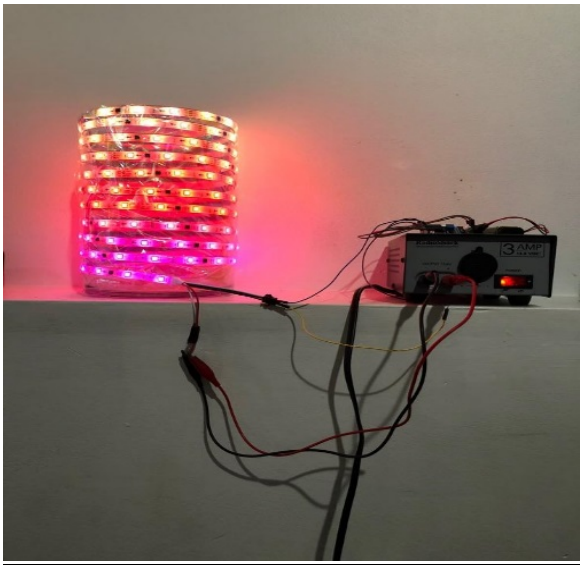

FIGURE 8 FINAL PROTOTYPE MOOD LAMP FUNCTION

Throughout this project, I gained tons of information about electronics, Arduino's, and coding. I am proud of the lamp that I created. However, there is room for improvement. Going forward, I want the next prototype to be more aesthetically pleasing. The car battery charger is bulky, and the display could benefit from a lamp shade. There are also two features I want to add: A clap switch and changing color patterns to sound. Due to time constraints and lack of knowledge, I had to use source code. With more time, I hope to eventually write my own code for this project.

Works Cited

Adafruit. *Adafruit_NeoPixel.h*. *GitHub*, github.com/adafruit/Adafruit_NeoPixel.

Anouskadg. "Musical Neopixels (WS2812B)." *Instructables*, AutoDesk, www.instructables.com/id/Musical-Neopixels-WS2812B/. Accessed 23 Apr. 2019.

Barela, Mike. *Trinket Sound-Reactive Color Organ*. *Adafruit*, learn.adafruit.com/trinket-sound-reactive-led-color-organ/code. Accessed 25 Apr. 2019.

Garcia, Daniel, and Mark Kriegsman. *FastLED.h*. *FastLED Animation Library*, 2010, fastled.io/. Accessed 23 Apr. 2019.

"How to Use WS2812B RGB LEDs with Arduino." *Youtube*, Google, 13 Sept. 2017, www.youtube.com/watch?v=9hJyyUTflXA&t=222s. Accessed 23 Apr. 2019.

Luijten, Hans. "Arduino – Controlling a WS2812 LED strand with NeoPixel or FastLED." *Tweak4All*, 4 Jan. 2014, www.tweaking4all.com/hardware/arduino/arduino-ws2812-led/. Accessed 23 Apr. 2019.

Nupen, Matt. *Neopixel Rainbow*. *Code Bender*, DISQUS, codebender.cc/sketch:80438#Neopixel%20Rainbow.ino. Accessed 23 Apr. 2019.

Prince. "Sound Sensitive Lights w/ Sound Sensor & Arduino." *PrinceTronics*, www.princetronics.com/sound-sensitive-lights-w-sound-sensor-arduino/. Accessed 31 Jan. 2015.