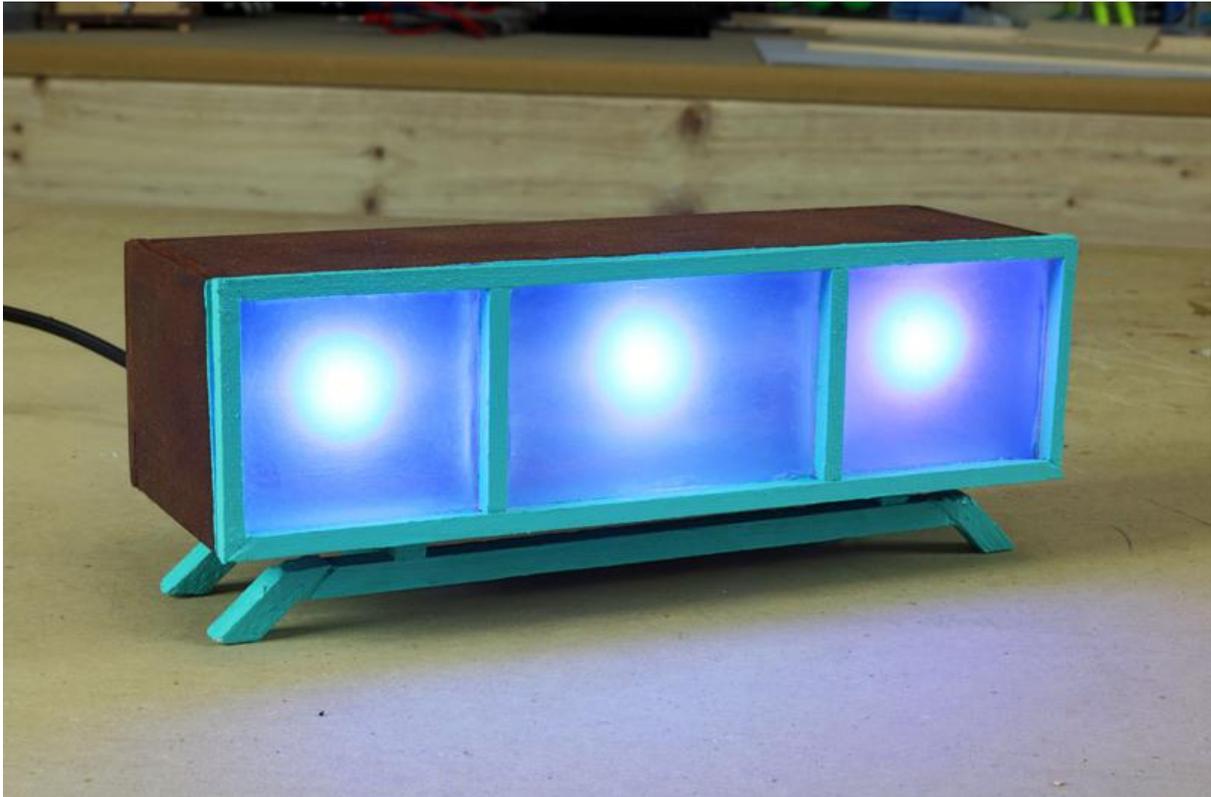




CheerLights

Created by Marc de Vinck



<https://learn.adafruit.com/cheerlights>

Last updated on 2022-12-01 02:44:04 PM EST

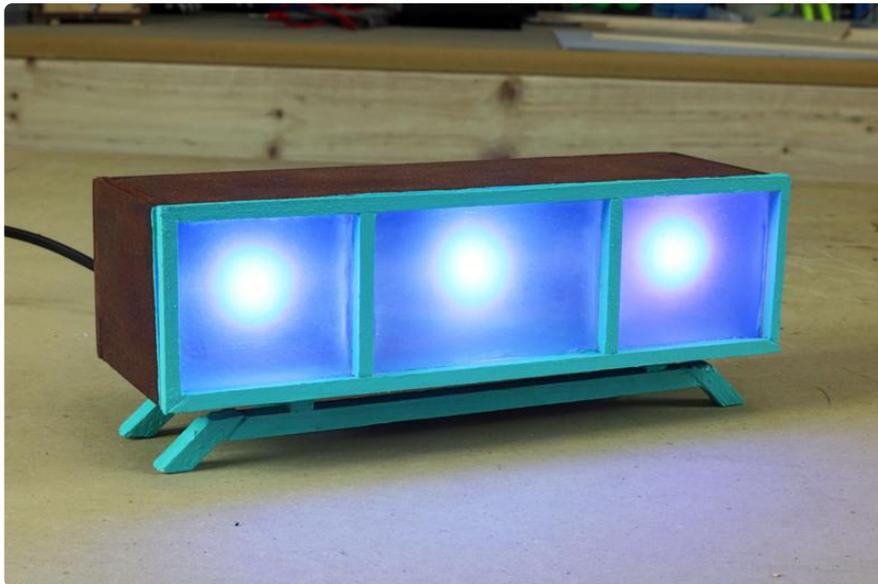
Table of Contents

Video	3
Overview	3
The Circuit	4
Libraries & Code	5
Enclosure	8

Video

Have some fun with the Adafruit Feather Huzzah and the CheerLight project. Originally created by Hans Scharler, CheerLights allows people's lights all over the world to synchronize to one color set by a specific Twitter feed. In this interpretation, Marc de Vinck creates a WiFi enable CheerLight using the Adafruit Feather Huzzah, Neopixels, and some clever code.

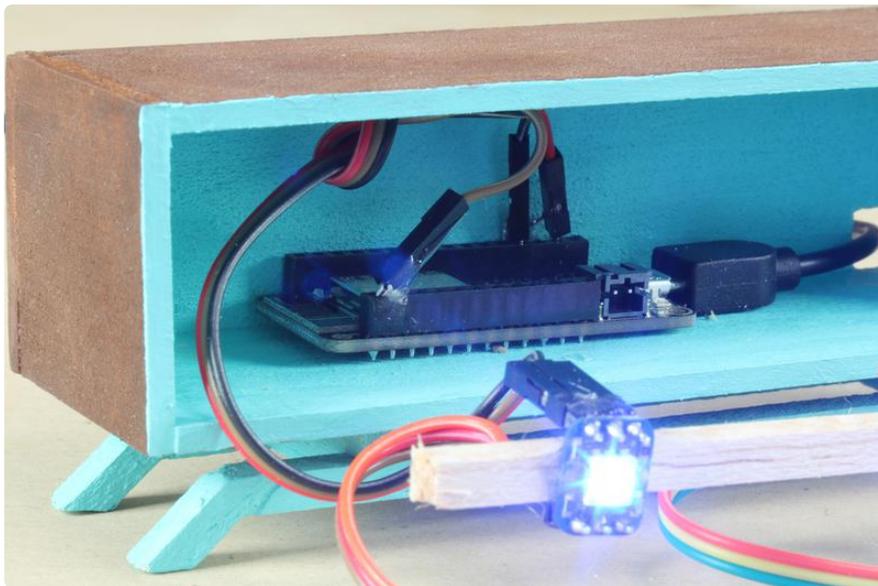
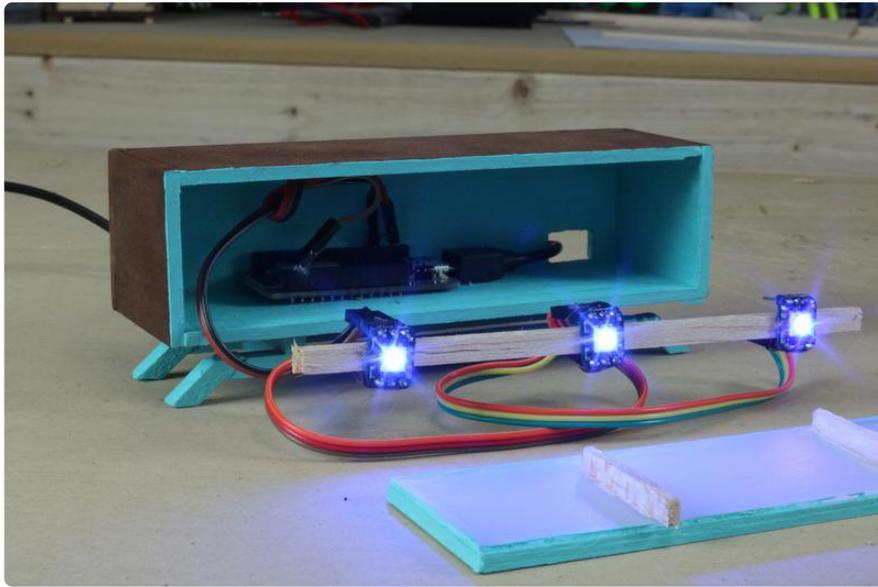
Overview



Connecting the virtual and physical worlds is always a popular DIY project. In this build I modify the the Cheerlight project code by Hans Scharler to work via a Feather Huzzah. Why the need for modifications? Good question! The Feather Huzzah uses a ESP8266 for the wireless connection, and although you can enable PWM on any GPIO pin, except Pin(16), to control the color mixing of an RGB LED, I found it much cleaner and easier to use Neopixels.

You can read all about the Feather Huzzah, and it's pinouts [here](#) (). And if you want to add a bunch of PWM to your Feather, you can always add an [8-Channel PWM or Servo FeatherWing](http://adafru.it/2928) (<http://adafru.it/2928>) to your project. There are also a lot of other [Feather projects](#) () in the Adafruit Learning System.

The Circuit



Wiring up this project is really simple. All you need to do is connect the following:

Feather HUZZAH pin -> Neopixel pin:

- HUZAZH GND -> Neopixel #1 (G)
- HUZAZH 5V -> Neopixel #1 (+)
- HUZAZH Pin (4) - Neopixel #1 (In)

After you wire up the Feather Huzzah to the first of your Neopixels just connect all subsequent Neopixels together: (Just make sure out from #1 goes to In of #2)

- Neopixel #1 (G) -> Neopixel #2 (G)
- Neopixel #1 (+) -> Neopixel #2 (+)
- Neopixel #1 (Out) -> Neopixel #2 (In)

And then add a few more if needed: (I added a 3rd Neopixel)

- Neopixel #2 (G) -> Neopixel #3 (G)
- Neopixel #2 (+) -> Neopixel #3 (+)
- Neopixel #2 (Out) -> Neopixel #3 (In)

You can solder them all together, or you could use female to female jumper wires, which is what I chose to use. Why? Because I realise that I'm the kind of maker that likes to constantly modify and reuse parts so I wanted it all to come apart easily for a future project.

Libraries & Code

You'll need several Arduino plugins and libraries for this project to work, they include:

- ESP8266 core board support library ([Learn more & download/install here \(\)](#))
- ThingSpeak Library ([Learn more & download here \(\)](#)) or load from within the Arduino IDE
- Adafruit Neopixel Library ([Learn more & download here \(\)](#)) or load from within the Arduino IDE

You can learn more about how to [install libraries here \(\)](#) if you're new to Arduino Libraries!

Once installed, the code is very straight forward. Sure, you'll need a basic understanding of Arduino, but it's nothing too complex! I've commented it fairly well, and wrote it so it can be easily modified. Once you made the simple circuit and installed the libraries, all you have to do is download the code below and make the following changes:

- Change the line of code "my_wifi_name" to your WiFi network name
- Change the line of code "my_wifi_password" to your WiFi password

That's it! Really! It's that simple to get up and running.

```
/*
  Adafruit Feather HUZZAH & Neopixel Cheerlights by Marc de Vinck
  CheerLights originally created by Hans Scharler

  Lots of code borrowed from CheerLights & Adafruit

  Adafruit HUZZAH - https://www.adafruit.com/products/2821
  Adafruit Neopixels - https://www.adafruit.com/products/1312
  Cheerlights - http://www.cheerlights.com
  ThingSpeak - https://www.thingspeak.com

  Project link - https://blog.adafruit.com/2016/05/06/feather-huzzah-neopixel-
  cheerlighs/
*/

#include <ESP8266WiFi.h> // Include the ESP8266 Library
#include "ThingSpeak.h" // Include the ThingSpeak library
#include <Adafruit_NeoPixel.h> // Include the adafruit Neopixel Library

#define PIN          4 // What pin is the data being sent to the Neopixels
#define NUMPIXELS    3 // How many Neopixels are you using?

Adafruit_NeoPixel pixels = Adafruit_NeoPixel(NUMPIXELS, PIN, NEO_GRB +
NEO_KHZ800); // Set up the Neopixels
int delayval = 500; // delay for half a second

WiFiClient  client;

const char* ssid      = "my_wifi_name"; // Enter your WiFi network name
const char* password  = "my_wifi_password"; // Enter your WiFi password
const char* host      = "api.thingspeak.com";
unsigned long cheerLightsChannelNumber = 1417; // The channel of Cheerlights

void setup() {
  pixels.begin(); // Start up Neopixels

  Serial.begin(115200); // Get ready for serial communications and display the
  connection status
  delay(100);

  Serial.println();
  Serial.println();
  Serial.print("Connecting to WiFi network - ");
  Serial.println(ssid);

  WiFi.begin(ssid, password);

  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }

  Serial.println("");
  Serial.println("WiFi connected");
  Serial.println("IP address: ");
  Serial.println(WiFi.localIP());

  ThingSpeak.begin(client);
}

String colorName[] = {"none", "red", "pink", "green", "blue", "cyan", "white",
```

```

"warmwhite", "oldlace", "purple", "magenta", "yellow", "orange", "1999"}; // List
of CheerLights color names

int colorRGB[][3] = {    0,  0,  0,  // "none"  // Map of RGB values for each of
the Cheerlight color names
                        255,  0,  0,  // "red"
                        255, 192, 203, // "pink"
                        0, 255,  0,  // "green"
                        0,  0, 255,  // "blue"
                        0, 255, 255,  // "cyan",
                        255, 255, 255,  // "white",
                        255, 223, 223,  // "warmwhite",
                        255, 223, 223,  // "oldlace",
                        128,  0, 128,  // "purple",
                        255,  0, 255,  // "magenta",
                        255, 255,  0,  // "yellow",
                        255,165,  0}; // "orange"};
                        128,  0, 128}; // "1999 Prince purple"

void loop() {

  Serial.print("connecting to ");
  Serial.println(host);

  // Use WiFiClient class to create TCP connections
  WiFiClient client;
  const int httpPort = 80;
  if (!client.connect(host, httpPort)) {
    Serial.println("connection failed");
    return;
  }

  // We now create a URI for the request
  String url = "/channels/1417/field/1/last.json";
  Serial.print("Requesting URL: ");
  Serial.println(url);

  // This will send the request to the server
  client.print(String("GET ") + url + " HTTP/1.1\r\n" +
              "Host: " + host + "\r\n" +
              "Connection: close\r\n\r\n");
  delay(500); // Just a quick pause to let everything settle
  String color = ThingSpeak.readStringField(cheerLightsChannelNumber, 1); //Read
the Cheerlight color and print to the serial monitor
  Serial.println(color);
  setColor(color);
  delay(18000);
}

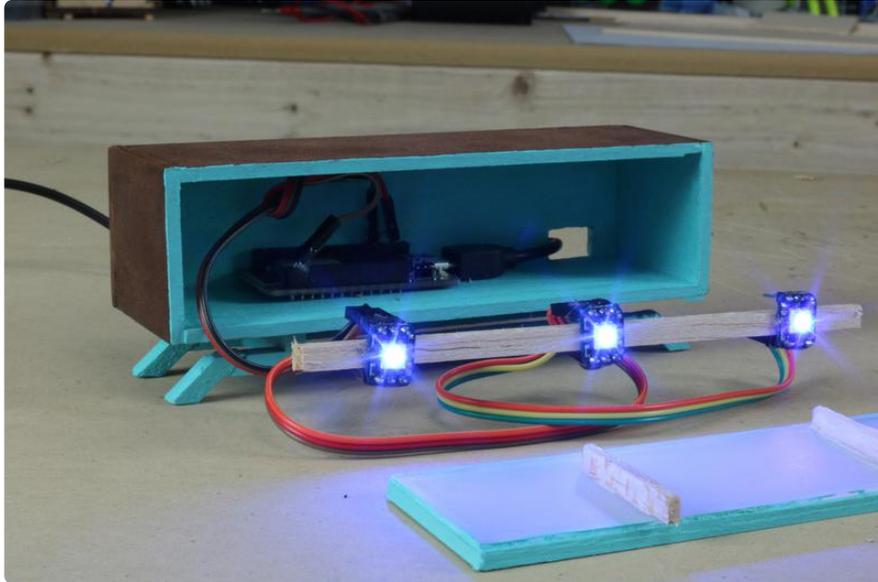
void setColor(String color)
{
  // Look through the list of colors to find the one that was requested
  for (int iColor = 0; iColor &lt;= 13; iColor++)
  {
    if (color == colorName[iColor])
    {
      pixels.setPixelColor(0, pixels.Color (colorRGB[iColor][0], colorRGB[iColor]
[1], colorRGB[iColor][2])); // set pixel 1 color (kept them separate for easy
hacking & understanding)
      pixels.setPixelColor(1, pixels.Color (colorRGB[iColor][0], colorRGB[iColor]
[1], colorRGB[iColor][2])); // set pixel 2 color (kept them separate for easy
hacking & understanding)
      pixels.setPixelColor(2, pixels.Color (colorRGB[iColor][0], colorRGB[iColor]
[1], colorRGB[iColor][2])); // set pixel 3 color (kept them separate for easy
hacking & understanding)
      pixels.show(); // This sends the updated pixel color to the hardware.

      delay(delayval); // Delay for a period of time (in milliseconds).
    }
  }
}

```

```
}  
}  
}
```

Enclosure



I'm not going to cover how I built the enclosure. I took some scrap pieces of wood that I had lying around, and cut them to build my little midcentury modern inspired enclosure.

There are a few things that I hope you learn from my build.

1. Enclosures are a great finishing touch. It makes any pile of electronics look infinitely better. OK, I like bare electronics too, but an enclosure is a good way to ensure it lasts and is safe.
2. You don't need a laser cutter or special tools. (gasp!) I used a standard box cutter and some quick drying glue and it looks fine. More importantly, it works fine!
3. A great resource for enclosure ideas is a simple Google image search based on a furniture or architectural design period. Examples: Midcentury Modern, Contemporary, Bauhaus, Art Deco, etc.

Almost anything can be an enclosure. I find some of the best things at the dollar store, garage sales, and even in the recycling bin. Be creative and have fun. Just remember if your enclosure has any exposed metal parts to cover them, or the electronics to prevent shorts!

If you are interested in how to design a laser cut enclosure for your project, [check out our guide \(\)](#).