Working with NeoPixels

PRESENTED BY RICHARD GOWEN (@alt_bier)

Created for BSidesDFW 2020 HHV

This Slide Deck Is Available at https://altbier.us/neopixels/

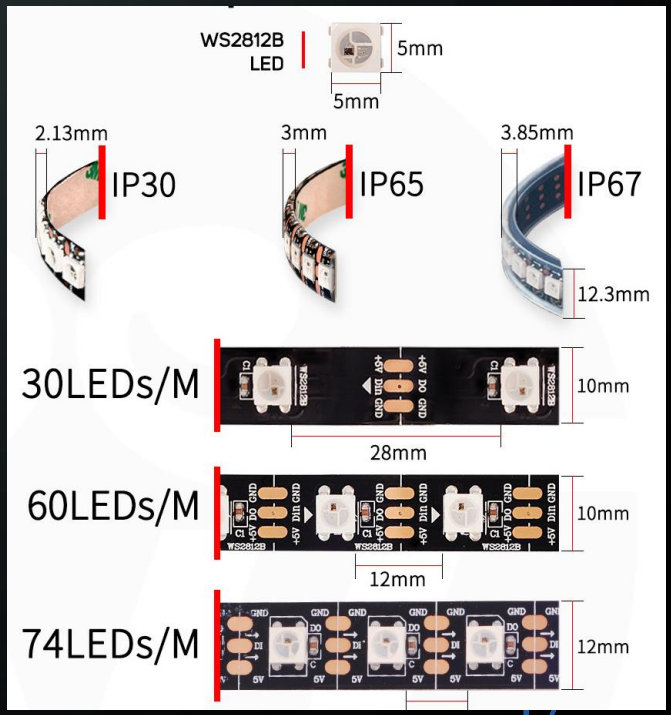
What is a NeoPixel?

- NeoPixel is the term coined by Adafruit to refer to their brand of individually addressable RGB color pixels and strips based on the WS2812 (and other) integrated light source packages using a single wire control protocol.
- The term NeoPixel has become the defacto standard name for addressable single wire controlled LEDs regardless of manufacturer.
- Unlike a standard LED, even the RGB type, NeoPixels cannot be lit with a simple circuit. They require a microcontroller to send the control signals that tell it how to light up.

Form Factors

NeoPixels come in many different form factors from individual LEDs to strips to matrixes and more. Here are a few popular examples:

- Through-Hole NeoPixels – These look like RGB LEDs as they have 4 pins, but they have a tiny NeoPixel embedded and can be chained together in a circuit using a single control wire.
- Surface Mount NeoPixels – Available a 5050 and other popular packages these SMD components are used in other NeoPixel form factors like the strips.
- NeoPixel Strips – These flexible LED strips are very popular as they can be cut to length to fit into any project. These come in several pixel densities such as 30 LEDs per meter, 60 LEDs per meter, and more.



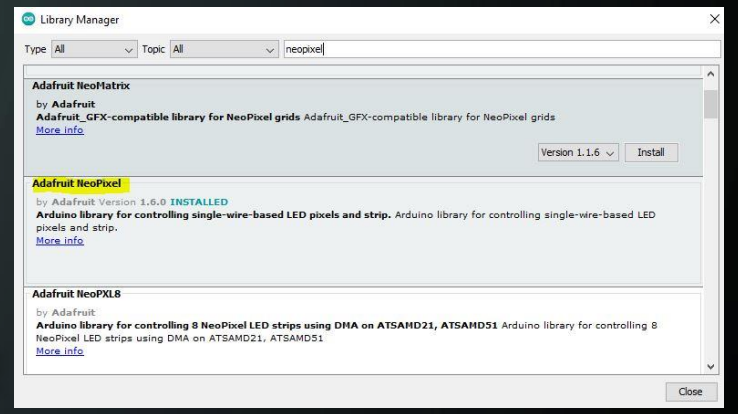
Adafruit NeoPixel Resources

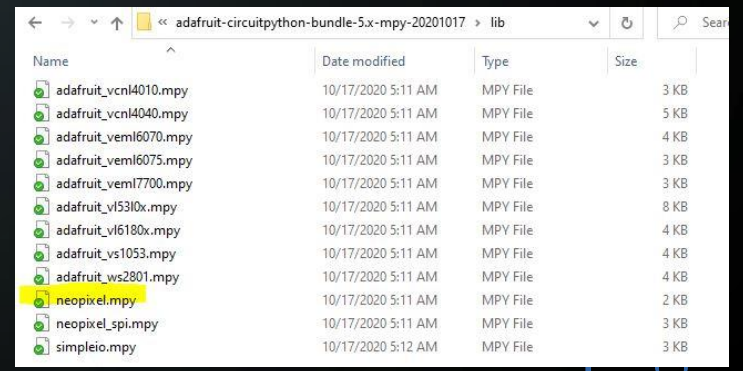
- The Adafruit website https://www.adafruit.com/ has a wide range of NeoPixel products and microcontrollers that can control them. They also have devices that contain microcontrollers and NeoPixels and other cool things all in a single form factor.
- You can find an awesome detailed guide to NeoPixels here: https://learn.adafruit.com/adafruit-neopixel-uberguide/

NeoPixel Libraries

While it is possible to write your own NeoPixel interface library, Adafruit has already created libraries for Arduino and CircuitPython.

- The Adafruit Arduino library is called Adafruit_NeoPixel
  - It can be installed via the library manager in the Arduino IDE
  - To use it you will add `#include <Adafruit_NeoPixel.h>` in your code
- The Adafruit CircuitPython library is named neopixel.mpy
  - This library file is part of the CircuitPython library bundle
  - To use it you just copy the file to the lib directory on your device and add `import neopixel` in your code





Connecting NeoPixels

To connect Neopixels for use you must provide power (usually +5V) and ground and a control wire that will connect a microcontroller to the Data-In on the pixel.

Individual pixels will have at least four connectors: VIN, GND, Data-In, Data-Out.

In a strip or other multi-pixel form the voltage and ground are run through the pixels like power rails and each pixels Data-Out is chained to the next Data-In.

While a small number of pixels can be powered by a microcontrollers power output, this is not recommended for larger strips of pixels as the power draw will quickly exceed the microcontrollers limits (each pixel can draw a max current of 60mA).

A large capacitor (1000uF) should be used on the power source for the pixels to prevent problems with initial on-rush current or voltage drops.

The control data line should contain a 470 Ohm resistor between the microcontroller and the NeoPixel Data-In to prevent spikes that could damage the pixels.

NeoPixel Addresses

- NeoPixels are individually addressable.
- What this means is that you can identify each pixel in a chain to control it even though they are all using the same control pin on the microcontroller.
- When assigning an object to a NeoPixel control pin it will operate like an array with each element referring to a pixel in the chain.
- Assigning objects and referencing individual pixel addresses varies slightly between Arduino and CircuitPython

Working with NeoPixels on Arduino

- You need to do the following for NeoPixels on Arduino
- Include the library using `#include <Adafruit_NeoPixel.h>`
- Define an object to reference the NeoPixels using the following syntax:

```
Adafruit_NeoPixel object = Adafruit_NeoPixel(<num_pixels>, <pin>, <type_flag> + <type_flag>);
```

  - The object name can be assigned following typical Arduino variable naming rules
  - The first parameter is the number of pixels; the second parameter is the Arduino pin that is controlling it
  - The third parameter is made up of flags that identify the type of NeoPixel. This typically includes the color layout with values such as NEO_GRB or NEO_RGB and the bitstream rate with values such as NEO_KHZ800 or NEO_KHZ400

```
Adafruit_NeoPixel strip = Adafruit_NeoPixel(5, A1, NEO_GRB + NEO_KHZ800);
```

- Prime your NeoPixels in setup() using `object.begin();` and `object.show();` like this:

```
void setup() { strip.begin(); strip.show(); }
```

- Within your main loop assign the NeoPixels colors using the following syntax:

```
object.setPixelColor(<pixel>, <Red[0-255]>, <Green[0-255]>, <Blue[0-255]>);
```

  - To set the 3rd pixel in your strip to bright red you would use this: `strip.setPixelColor(2, 255, 0, 0);`
- After assigning colors use `object.show();` to send the assignments to the NeoPixels: `strip.show();`
- Beyond that you would add delays and other code like you would working with any other LED.

Working with NeoPixels on CircuitPython

- You need to do the following for NeoPixels with CircuitPython
- Import the library using `import neopixel`
- Define an object to reference the NeoPixels using the following syntax:

  `object = neopixel.NeoPixel(<pin>, <num_pixels>, <flag>=<flag_value>, <flag>=<flag_value>...)`

  - The object name can be assigned following typical variable naming rules
  - The first parameter is the pin that is controlling it; the second parameter is the number of pixels
  - The third parameter is made up of flags that identify the type of NeoPixel and other settings. Here are some flags:
  - `pixel_order` typical values: neopixel.GRB or neopixel.RGB – pixel color Layout
  - `auto_write` values:. True or False – send to pixels upon assignment without a show()
  - `brightness` values: 0.0 to 1.0 – brightness percentage

  ```
  strip = neopixel.NeoPixxel(board.D1, 5, pixel_order=neopixel.GRB, brightness=0.7);
  ```

- Flags can be set outside object definition using `object.<flag> = <flag_value>` like this: `strip.auto_write = False`
- Within your main loop assign the NeoPixels colors using the following syntax:

  `object[pixel] = (<Red[0-255]>, <Green[0-255]>, <Blue[0-255]>)`

  - To set the 3rd pixel in your strip to bright red you would use this: `strip[2] = (255, 0, 0)`
- After assigning colors use `object.show()` to send the assignments to the NeoPixels: `strip.show()`
- Beyond that you would add delays and other code like you would working with any other LED.

NEOPIXEL PROJECTS

This next section will outline some NeoPixel projects using the Arduino and Trinket.

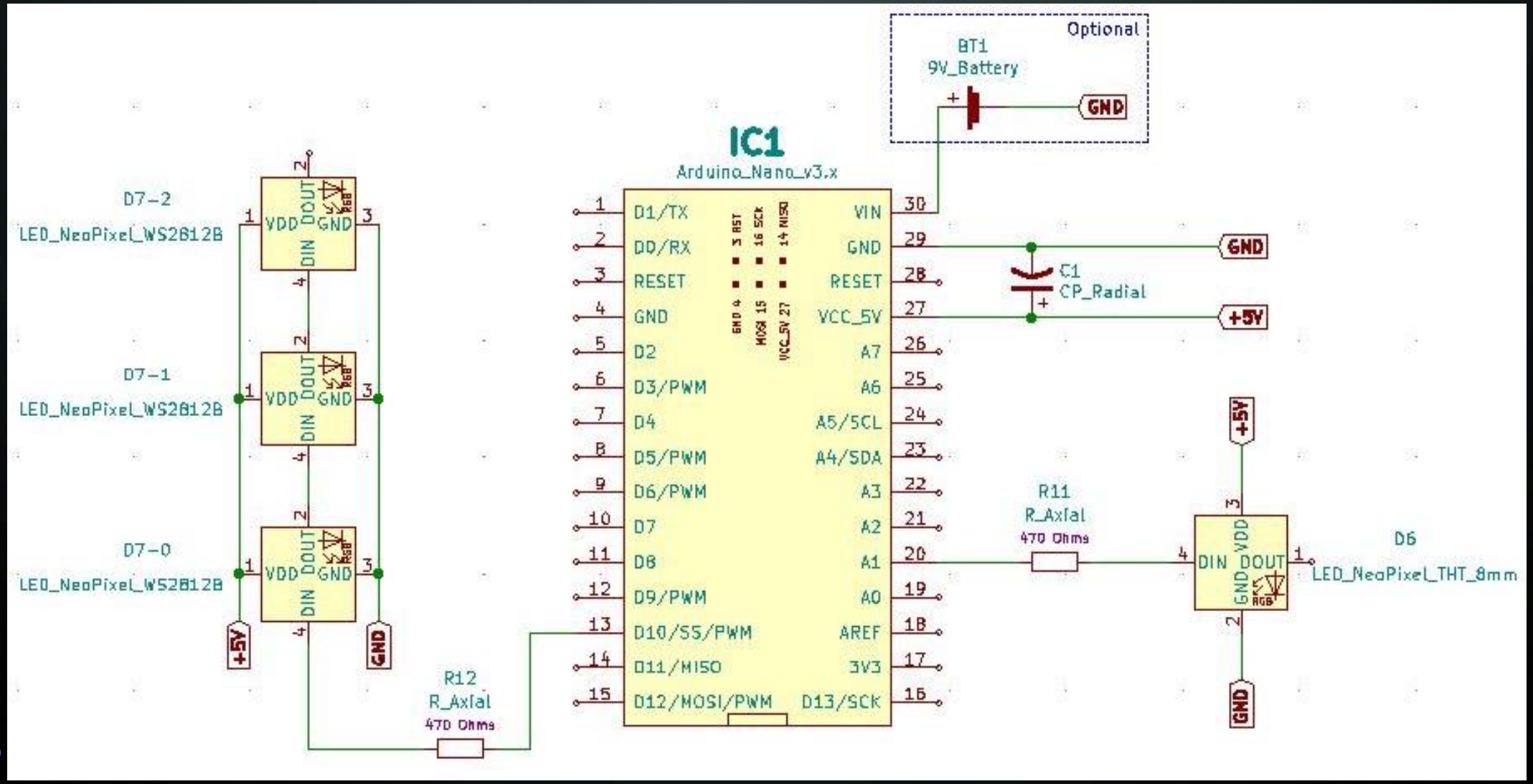
These projects will be centered around two different physical circuit layouts with several Arduino or CircuitPython code blocks for each.

- Arduino Nano NeoPixel Control – Lab HHV2020_06
  - Blink a single 8mm NeoPixel
  - Blink a strip of NeoPixels
  - Color cycle a strip of NeoPixels
- Adafruit Trinket CircuitPython NeoPixel Control – Lab HHV2020_07
  - Blink a single 8mm NeoPixel
  - Blink a strip of NeoPixels
  - Color cycle a strip of NeoPixels

The Lab reference numbers refer to the BSidesDFW Hardware Hacking Village Videos which can be accessed here: **https://altbier.us/bsidesdfwHHV2020/**

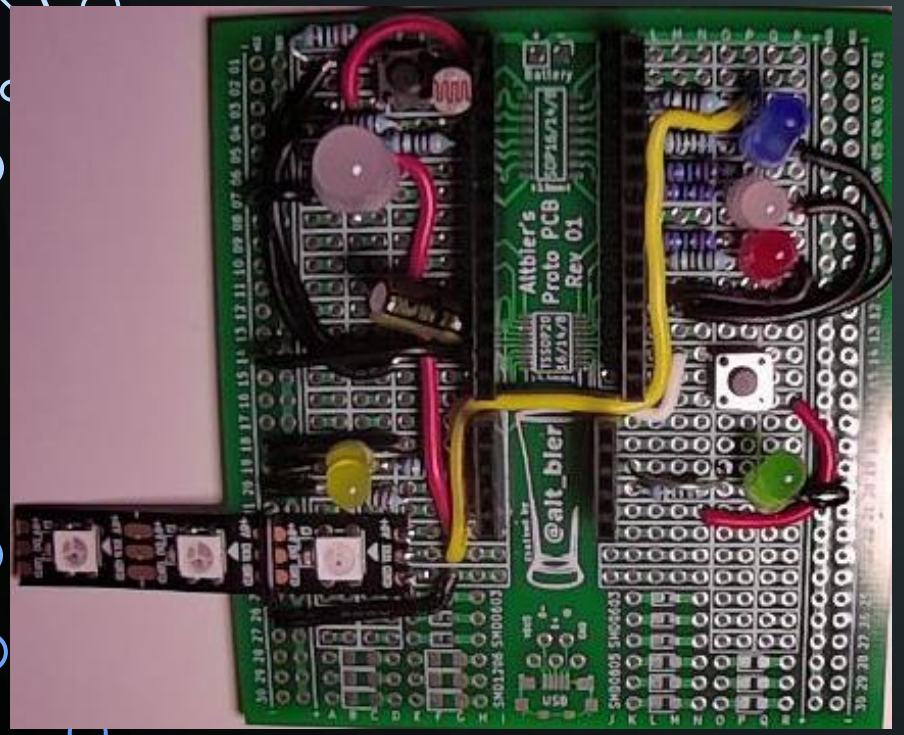
ARDUINO NANO NEOPIXEL CONTROL

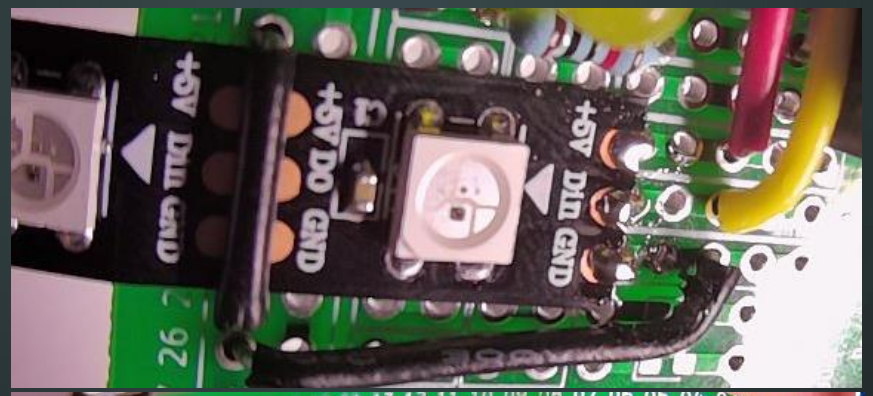
Schematic

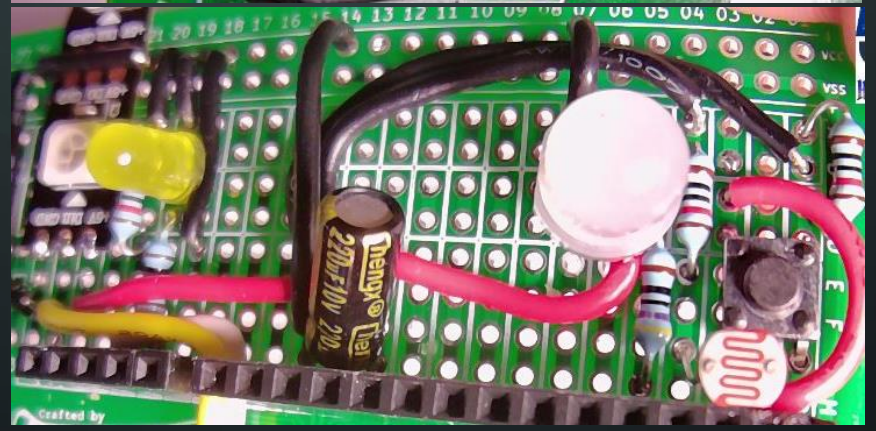


ARDUINO NANO NEOPIXEL CONTROL

Physical Layout







Wire up a circuit as shown in the schematic and physical layout.

Components:

- 2x Resistor 470 Ohm
- 1x Capacitor 220 uF
- 1x NeoPixel 8mm THT
- 1x NeoPixel Strip

Strip Board Connection Details

- Nano – I1-15 and K1-15
- Resistor 470 Ohm – L3 and O3
- Resistor 470 Ohm – D5 and H5
- Capacitor 220 uF (Polarized)
  - Anode – H12
  - Cathode – H14
- NeoPixel 8mm THT APA106
  - Pin 1 (DIN) – C5
  - Pin 2 (VIN) – C6
  - Pin 3 (GND) – B7
  - Pin 4 (DOUT) – C8
- NeoPixel Strip WS2812B
  - Pin 1 (VIN) – E22
  - Pin 2 (DIN) – E23
  - Pin 3 (GND) – E24
- Wire – G14 and VCC14
- Wire – A7 and VCC7
- Wire – G24 and VCC27
- Wire – D6 and E12
- Wire – F12 and G22
- Wire – G23 and P3
- Wire – VCC21 and VCC25

ARDUINO NANO NEOPIXEL CONTROL

Blink a single 8mm NeoPixel

This code will blink an external 8mm NeoPixel with the colors Red then Green then Blue.

Let's examine this code.

We include the Adafruit_NeoPixel library.

We then create a NeoPixel object named LED6 using pin A1 with 1 pixel that is an RGB 800Khz type.

We prime it in the setup() and then in the main loop() we set each color and show it for ½ sec.

Arduino_Nano_Neopixel_Single_.ino

```ino
/* Include the Adafruit NeoPixel library */
#include <Adafruit_NeoPixel.h>

/* DEFINE APA106 NEOPIXEL */
#define LED6_PIN A1
#define LED6_NUM 1

// Create NeoPixel object
Adafruit_NeoPixel LED6 = Adafruit_NeoPixel(LED6_NUM, LED6_PIN, NEO_RGB + NEO_KHZ800);

void setup() {
  // start the single neopixel and blank it out
  LED6.begin();
  LED6.show();
}

// MAIN LOOP
void loop() {
  // set pixel color and show with delay
  LED6.setPixelColor(0, 255, 0, 0);
  LED6.show();
  delay(500);
  // set pixel color and show with delay
  LED6.setPixelColor(0, 0, 255, 0);
  LED6.show();
  delay(500);
  // set pixel color and show with delay
  LED6.setPixelColor(0, 0, 0, 255);
  LED6.show();
  delay(500);
}
```

This Code Is Available Here: https://github.com/gowenrw/BSidesDFW_2020_HHV/

ARDUINO NANO NEOPIXEL CONTROL

Blink a strip of NeoPixels

This code will expand on the previous code to also blink an external strip of 3 NeoPixels with each showing the colors Red, Green or Blue.

Let's examine this code.

We include the Adafruit_NeoPixel library.

We then create a NeoPixel object named LED7 using pin D10 with 3 pixels that is an GRB 800Khz type.

We prime it in the setup() and then in the main loop() we set each color and show it for ½ sec.

```
#include <Adafruit_NeoPixel.h>

/* DEFINE APA106 NEOPIXEL */
#define LED6_PIN A1
#define LED6_NUM 1

/* DEFINE WS2812B NEOPIXEL STRIP */
#define LED7_PIN 10
#define LED7_NUM 3

// Create NeoPixel objects
Adafruit_NeoPixel LED6 = Adafruit_NeoPixel(LED6_NUM, LED6_PIN, NEO_RGB + NEO_KHZ800);
Adafruit_NeoPixel LED7 = Adafruit_NeoPixel(LED7_NUM, LED7_PIN, NEO_GRB + NEO_KHZ800);

void setup() {
  // start the single neopixel and blank it out
  LED6.begin();
  LED6.show();
  // start the neopixel strip and blank it out
  LED7.begin();
  LED7.show();
}

// MAIN LOOP
void loop() {
  // set pixel color and show with delay
  LED6.setPixelColor(0, 255, 0, 0);
  LED6.show();
  LED7.setPixelColor(0, 255, 0, 0);
  LED7.setPixelColor(1, 0, 255, 0);
  LED7.setPixelColor(2, 0, 0, 255);
  LED7.show();
  delay(500);
  // set pixel color and show with delay
  LED6.setPixelColor(0, 0, 255, 0);
  LED6.show();
  LED7.setPixelColor(0, 0, 255, 0);
  LED7.setPixelColor(1, 0, 0, 255);
  LED7.setPixelColor(2, 255, 0, 0);
  LED7.show();
  delay(500);
  // set pixel color and show with delay
  LED6.setPixelColor(0, 0, 0, 255);
  LED6.show();
  LED7.setPixelColor(0, 0, 0, 255);
  LED7.setPixelColor(1, 255, 0, 0);
  LED7.setPixelColor(2, 0, 255, 0);
  LED7.show();
  delay(500);
}
```

This Code Is Available Here: https://github.com/gowenrw/BSidesDFW_2020_HHV/

ARDUINO NANO NEOPIXEL CONTROL

Color cycle a strip of NeoPixels

This code expands on the previous code changing the main loop from blinking colors to cycling through colors.

Let's examine this new main loop.

We use a for loop running 256 times and use some simple math on the iteration to set different RGB color values for the NeoPixels.

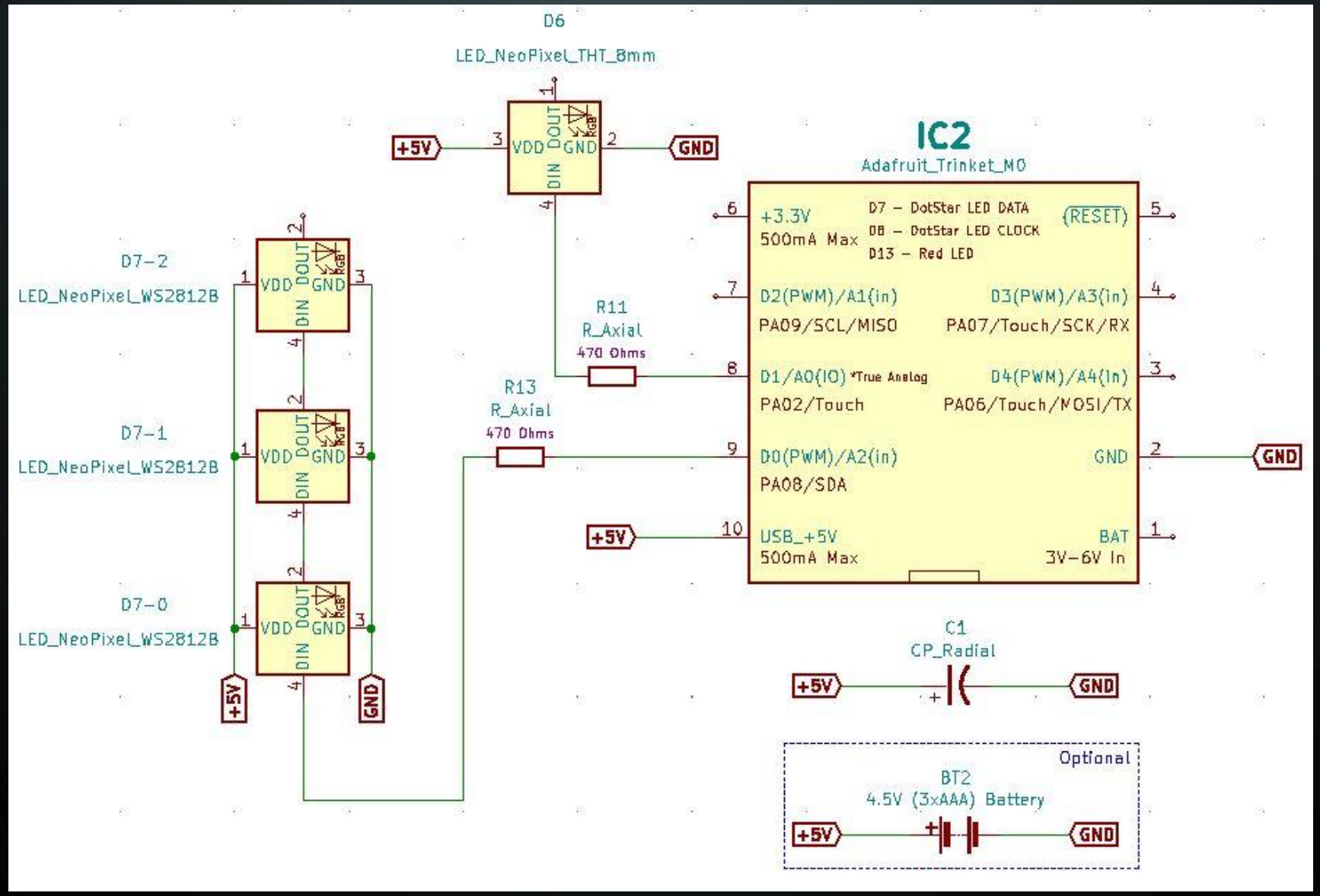
Within each loop iteration we show the given color set for 50ms before we return to the top of the loop to set the next color.

```
41  void loop() {
42
43    // Loop 0-255 to set pixel colors
44    for(int i=0; i<256; i++) {
45      int pos = i;
46      if (pos < 85) {
47        // Cycle Red & Green based on itiration number
48        LED6.setPixelColor(0, int(pos * 3), int(255 - (pos*3)), 0);
49        LED7.setPixelColor(0, int(pos * 3), int(255 - (pos*3)), 0);
50        LED7.setPixelColor(1, int(pos * 3), int(255 - (pos*3)), 0);
51        LED7.setPixelColor(2, int(pos * 3), int(255 - (pos*3)), 0);
52      } else if (pos < 170) {
53        pos = pos - 85;
54        // Cycle Red & Blue based on itiration number
55        LED6.setPixelColor(0, int(255 - pos*3), 0, int(pos*3));
56        LED7.setPixelColor(0, int(255 - pos*3), 0, int(pos*3));
57        LED7.setPixelColor(1, int(255 - pos*3), 0, int(pos*3));
58        LED7.setPixelColor(2, int(255 - pos*3), 0, int(pos*3));
59      } else {
60        pos = pos -170;
61        // Cycle Green & Blue based on itiration number
62        LED6.setPixelColor(0, 0, int(pos*3), int(255 - pos*3));
63        LED7.setPixelColor(0, 0, int(pos*3), int(255 - pos*3));
64        LED7.setPixelColor(1, 0, int(pos*3), int(255 - pos*3));
65        LED7.setPixelColor(2, 0, int(pos*3), int(255 - pos*3));
66      }
67      LED6.show();
68      LED7.show();
69      delay(50);
70    }
71
72  }
```

This Code Is Available Here: https://github.com/gowenrw/BSidesDFW_2020_HHV/

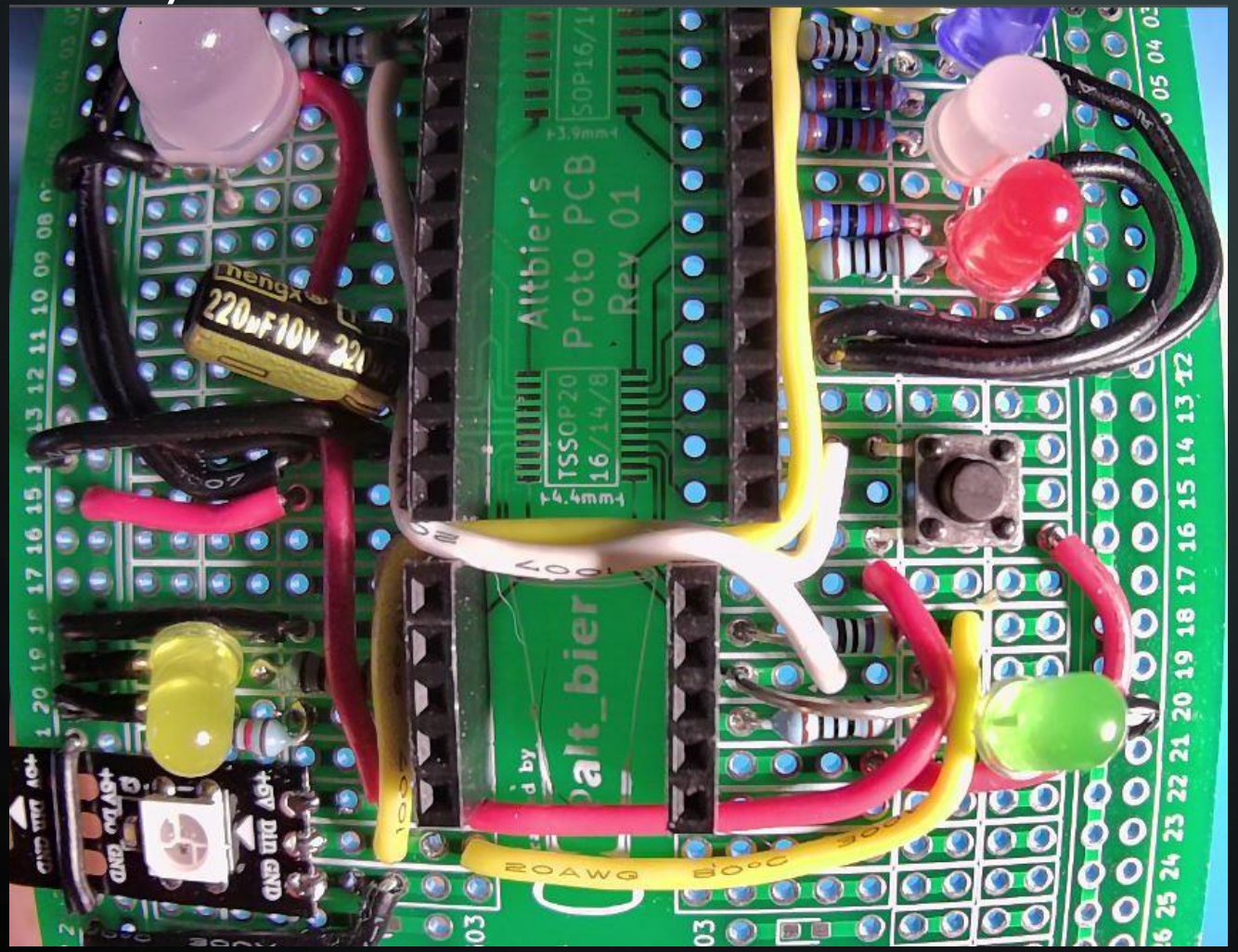
ADAFRUIT TRINKET CIRCUITPYTHON NEOPIXEL CONTROL

Schematic



ADAFRUIT TRINKET CIRCUITPYTHON NEOPIXEL CONTROL

Physical Layout



Wire up a circuit as shown in the schematic and physical layout.

Strip Board Connection Details

- Trinket – I17-21 and J17-21
- Resistor 470 Ohm – K18 and O18
- Wire – P18 and I23
- Wire – N17 and I22
- Wire – M19 and G5

Note: Most of the components and connections needed for this schematic are being shared with the Arduino schematic and were added in the previous lab

Components (not added in previous lab):

- 1x Resistor 470 Ohm

ADAFRUIT TRINKET CIRCUITPYTHON NEOPIXEL CONTROL

CircuitPython Library File Needed

The code used in the following examples will require a library file that is not built into CircuitPython.

You can find the library file bundle packages for recent versions of CircuitPython here: `https://circuitpython.org/libraries`

Download the bundle for the version you are working with (5.x) and extract it in a directory on your PC.

Locate this library file and copy it to the lib directory on the CIRCUITPY drive:

- **neopixel.mpy**

ADAFRUIT TRINKET CIRCUITPYTHON NEOPIXEL CONTROL

Blink a single 8mm NeoPixel

This code will blink an external 8mm NeoPixel with the color red.

Let's examine this code.

We import the neopixel library.

We define the pin and color-order constants

We then create a neopixel object named LED6 using the pin and color-order and with a type flag of auto_write=False (telling it to wait for a show command to send to the pixels).

We set our color and show it then sleep for ½ sec before turning it off for ½ sec.

main.py

```
import board, time
import neopixel

# Define Neopixels
LED6_PIN = board.D1  # pin that the NeoPixel is connected to
# Most Neopixels have a color order of GRB or GRBW some use RGB
LED6_ORDER = neopixel.RGB  # pixel color channel order
# Create NeoPixel object
LED6 = neopixel.NeoPixel(
    LED6_PIN, 1, pixel_order=LED6_ORDER, auto_write=False
)

### MAIN LOOP ###
while True:
    LED6[0] = (255, 0, 0)
    LED6.show()
    time.sleep(0.5)
    LED6[0] = (0, 0, 0)
    LED6.show()
    time.sleep(0.5)
```

This Code Is Available Here: https://github.com/gowenrw/BSidesDFW_2020_HHV/

ADAFRUIT TRINKET CIRCUITPYTHON NEOPIXEL CONTROL

Blink a strip of NeoPixels

This code expands on the previous code to also blink an external strip of 3 NeoPixels.

Let's examine this code.

We import the neopixel library.

We define the pin and color-order constants (note that the strip has a different order than the 8mm pixel)

We then create neopixel objects named LED6 and LED7 using the pin and color-order and with a type flag of auto_write=False (telling it to wait for a show command to send to the pixels).

We set an additional type flag on LED7 of brightness=0.3 since the strip is so bright we want to turn down its intensity.

We set our colors and show it then sleep for ½ sec before turning them off for ½ sec.

```
import board, time
import neopixel

# Define Neopixels
LED6_PIN = board.D1  # pin that the NeoPixel is connected to
LED7_PIN = board.D0  # pin that the NeoPixel is connected to
# Most Neopixels have a color order of GRB or GRBW some use RGB
LED6_ORDER = neopixel.RGB  # pixel color channel order
LED7_ORDER = neopixel.GRB  # pixel color channel order
# Create NeoPixel object
LED6 = neopixel.NeoPixel(
    LED6_PIN, 1, pixel_order=LED6_ORDER, auto_write=False
)
LED7 = neopixel.NeoPixel(
    LED7_PIN, 3, pixel_order=LED7_ORDER, auto_write=False
)
# Turn down strip brightness to 30%
LED7.brightness = 0.3

### MAIN LOOP ###
while True:
    LED6[0] = (255, 0, 0)
    LED6.show()
    LED7[0] = (255, 0, 0)
    LED7[1] = (0, 255, 0)
    LED7[2] = (0, 0, 255)
    LED7.show()
    time.sleep(0.5)
    LED6[0] = (0, 0, 0)
    LED6.show()
    LED7[0] = (0, 0, 0)
    LED7[1] = (0, 0, 0)
    LED7[2] = (0, 0, 0)
    LED7.show()
    time.sleep(0.5)
```

This Code Is Available Here: https://github.com/gowenrw/BSidesDFW_2020_HHV/

ADAFRUIT TRINKET CIRCUITPYTHON NEOPIXEL CONTROL

Color cycle a strip of NeoPixels

This code color cycles the NeoPixel strip.

Let's examine this code.

We import the neopixel library and define pin and color-order constants for LED7 (we use direct values in LED6 object)

We then create neopixel objects named LED6 and LED7. Note that the auto_write flag is not set meaning it will default to True.

We set the LED7 brightness=0.3

We create a function to return RGB color values based on an input from 0 – 255.

We set our set the colors for LED7 pixels based on our function and iterate a number from 0-255 to use in that function. Note we are not using show() since it is not needed with auto_write=True

```python
import board, time
import neopixel

# Define Neopixels
LED7_PIN = board.D0  # pin that the NeoPixel is connected to
# Most Neopixels have a color order of GRB or GRBW some use RGB
LED7_ORDER = neopixel.GRB  # pixel color channel order
# Create NeoPixel object
LED6 = neopixel.NeoPixel(board.D1, 1, pixel_order=neopixel.RGB)
LED7 = neopixel.NeoPixel(LED7_PIN, 3, pixel_order=LED7_ORDER)
# Turn down brightness to 30%
LED7.brightness = 0.3
# Function to color cycle NeoPixels
def wheel(pos):
    if (pos < 0) or (pos > 255):
        return (0, 0, 0)
    if (pos < 85):
        return (int(pos * 3), int(255 - (pos*3)), 0)
    elif (pos < 170):
        pos -= 85
        return (int(255 - pos*3), 0, int(pos*3))
    else:
        pos -= 170
        return (0, int(pos*3), int(255 - pos*3))
# Iteration Var
i = 0

### MAIN LOOP ###
while True:
    LED6[0] = (0, 0, 0) # turn off 8mm to focus on strip
    LED7[0] = wheel(i & 255)
    LED7[1] = wheel(i & 255)
    LED7[2] = wheel(i & 255)
    time.sleep(0.05)
    i = (i+1) % 256  # run from 0 to 255
```

This Code Is Available Here: https://github.com/gowenrw/BSidesDFW_2020_HHV/

THANK YOU

I hope you enjoyed this presentation and learned something from it.

-- @alt_bier

This Slide Deck – https://altbier.us/neopixels/

Code – https://github.com/gowenrw/BSidesDFW_2020_HHV/