Arduino with Neopixel Ring Workshop

SLQ Wiki Fabrication Lab 2026/01/18 08:28

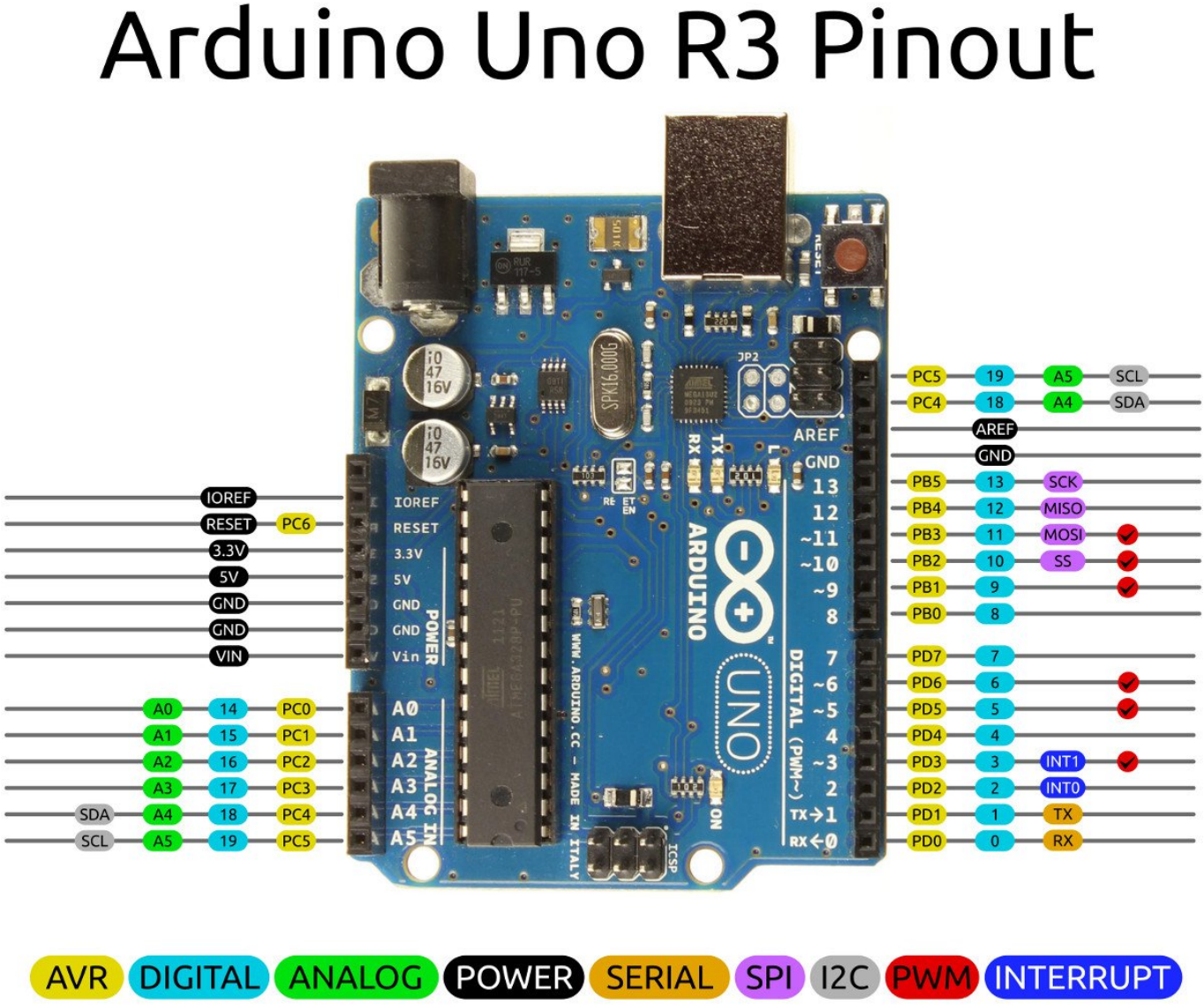
Arduino with Neopixel Ring Workshop

Introduction and Setup

Goal: Recap Arduino basics and set up the workspace.

Activities:

- **Brief review of Arduino Uno setup:** We will begin by familiarising ourselves with the Arduino Uno. It's similar to the Nano but has more digital input/output pins. For us this isn't super relevant as we'll be using 3 pins.
- **Installing the Arduino IDE and Neopixel library:** Ensure the Arduino IDE is installed on your computer. We will also go through the process of adding the Neopixel library via the Library Manager. Search for “Adafruit Neopixel” and install the latest version.
- **Setting up hardware:** Connect the Arduino Uno to your breadboard, then connect the Neopixel ring. Ensure that you have connected the data input pin of the Neopixel to one of the PWM output pins on the Arduino (usually pin 6 for simplicity).



Understanding Arduino Uno Hardware Design

Datasheet for the LED controller

Understanding the Neopixel Library

Goal: Understand how the Neopixel library controls the LED ring.

Activities:

- **Discussing Neopixel library functions and parameters:** The Neopixel library provides a set

of functions that allow you to control the LEDs in terms of color and brightness. Functions such as `setPixelColor()` and `show()` are essential for basic operations.
- **Exploring the documentation:** Navigate the online documentation to identify key functions and how they're used.

Challenge:

1. Understand how to change the color and intensity of individual LEDs using `setPixelColor()` where color is a combination of red, green, and blue values. This function allows for precise control over each LED's color on the ring.

Potentially useful functions (fill in the blanks!):

- `begin()` - Initializes the pin and sets up the Neopixel strip.
- `show()` - Sends the updated color data to the Neopixel ring.
- `setPixelColor(index, color)` - Sets the color of a specific LED in the ring.
- `fill(color, firstLED, count)` - Fills a number of LEDs with a specified color starting from a particular index.
- `setBrightness()`
- `clear()`
- `fill()`

Adafruit Neopixel Library on GitHub Adafruit Neopixel Library Documentation Adafruit Neopixel Code Snippets

Dissecting Example Code

Goal: Apply understanding by analyzing and modifying provided example code.

Activities:

- **Walk through a basic example:** Analyze a simple sketch where the Neopixel ring is set to display a single color across all LEDs.
- **Modifying the example:** Change parameters in the `setPixelColor()` function to adjust colors and experiment with `setBrightness()` to see the effect on the ring's luminosity.
- **Pattern creation:** Try turning on LEDs in a specific sequence or pattern to create visual effects like a running light or a color wave.

```
// NeoPixel Ring simple sketch (c) 2013 Shae Erisson
// Released under the GPLv3 license to match the rest of the
// Adafruit NeoPixel library

#include <Adafruit_NeoPixel.h>
#ifdef __AVR__
 #include <avr/power.h> // Required for 16 MHz Adafruit Trinket
#endif

// Why do we have these #define instructions and how do they differ from
```

```
regular variables?

#define PIN        6  // On Trinket or Gemma, suggest changing this to 1
#define NUMPIXELS 16  // Popular NeoPixel ring size

// Declare our NeoPixel strip object:
// Argument 1 = Number of LEDs in NeoPixel ring
// Argument 2 = Arduino pin number (most are valid)
// Argument 3 = Pixel type flags, add together as needed:
//   NEO_KHZ800  800 KHz bitstream (most NeoPixel products w/WS2812 LEDs)
//   NEO_KHZ400  400 KHz (classic 'v1' (not v2) FLORA pixels, WS2811
drivers)
//   NEO_GRB     Pixels are wired for GRB bitstream (most NeoPixel products)
//   NEO_RGB     Pixels are wired for RGB bitstream (v1 FLORA pixels, not
v2)
//   NEO_RGBW    Pixels are wired for RGBW bitstream (NeoPixel RGBW
products)

Adafruit_NeoPixel pixels(NUMPIXELS, PIN, NEO_GRB + NEO_KHZ800);

#define DELAYVAL 500  // Time (in milliseconds) to pause between pixels

void setup() {
  pixels.begin();  // INITIALIZE NeoPixel strip object (REQUIRED)
}

void loop() {
  pixels.clear();  // Set all pixel colors to 'off'

  for(int i = 0; i < NUMPIXELS; i++) {  // For each pixel...
    pixels.setPixelColor(i, pixels.Color(0, 150, 0));
    pixels.show();   // Send the updated pixel colors to the hardware.
    delay(DELAYVAL); // Pause before next pass through loop
  }
}
```

Programming Challenge: Create Patterns

Goal: Develop custom light patterns and sequences.

Activities:

- **Pattern design:** Design some unique patterns using loops and conditional statements.
- **Dynamic effects:** Think about techniques for creating gradients and other dynamic effects by manipulating delays and colors within the loop.

Challenges:

1. Create a gradient effect that transitions across the Neopixel ring.
2. Develop a “chase” sequence where a single lit LED moves around the ring.