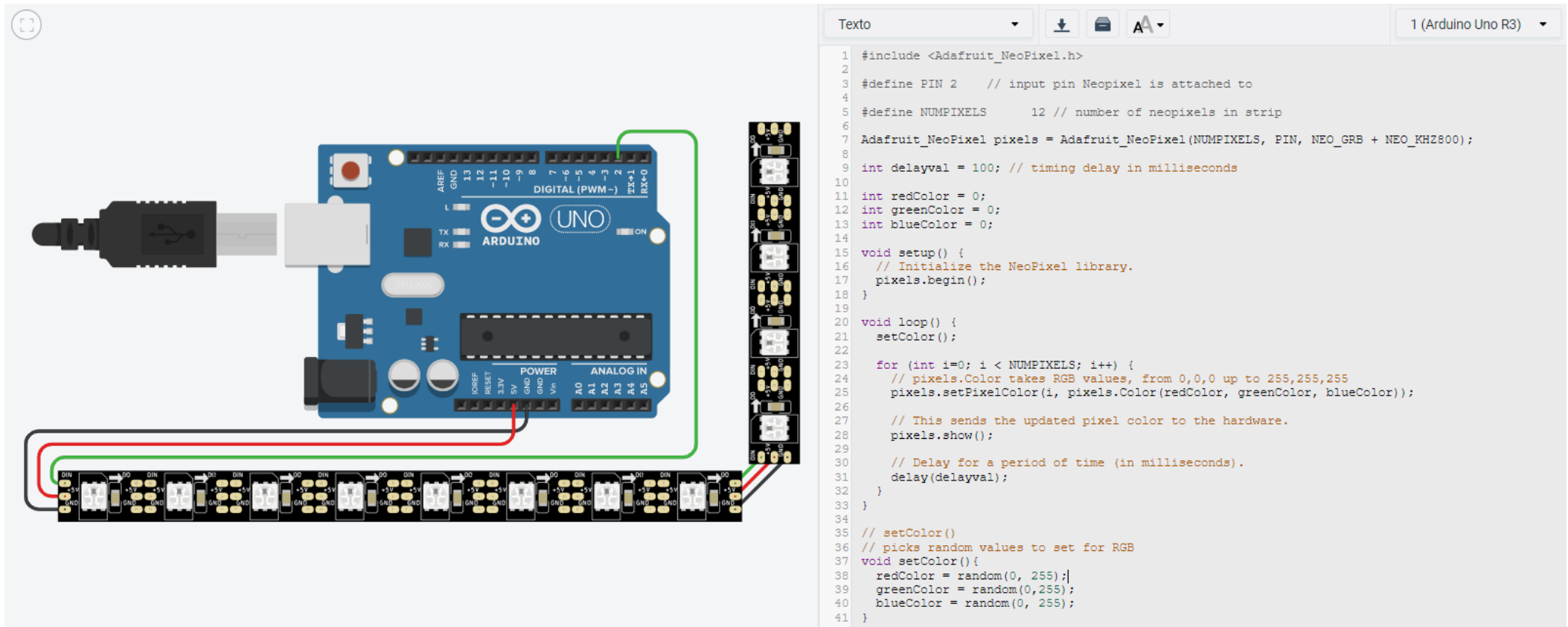


NeoPixel



`#include <Adafruit_NeoPixel.h>` => está inserindo a biblioteca do NeoPixel.

`#define PIN 2` => pino de entrada dos NeoPixels.

`#define NUMPIXELS 12` => é o número de NeoPixels que estão ligados no Arduino.

Adafruit_NeoPixel pixels = Adafruit_NeoPixel(NUMPIXELS, PIN, NEO_GRB + NEO_KHZ800); => está declarando o objeto “pixels” da “faixa” onde estão os NeoPixels. Há parâmetros para inserir nesse método (Adafruit_NeoPixel):

O primeiro é o número de pixels na “faixa” (NUMPIXELS), o segundo é o número do pino que ele está conectado (PIN) e o terceiro é tipo de pixel (NEO_GRB) mais o bitstream (NEO_KHZ800).

- NEO_GRB => Os pixels estão conectados para bitstream GRB
- NEO_KHZ800 => Fluxo de bits de 800 kHz

int delayval = 100; => Delay em milissegundos.

pixels.begin(); => método que inicia a biblioteca do NeoPixel.

setColor(); => método que vai pegar valores “aleatórios” para aparecer no RGB (o setColor é apenas o nome do método, já que os valores aleatórios serão sorteados pelo “random”).

pixels.setPixelColor(i, pixels.Color(redColor, greenColor, blueColor)); => o método “setPixelColor” está pegando o objeto “pixels” e definindo sua cor. O “i” seria o número ao longo da “faixa”; e o método pixels.Color vai pegar valores de 0 a 255 para cada cor no RGB (redColor, greenColor, blueColor).

pixels.show(); => método que envia a cor atualizada do pixel para o hardware.

delay(delayval); => o método de delay que vai pegar o delayval e colocar aqui como parâmetro.