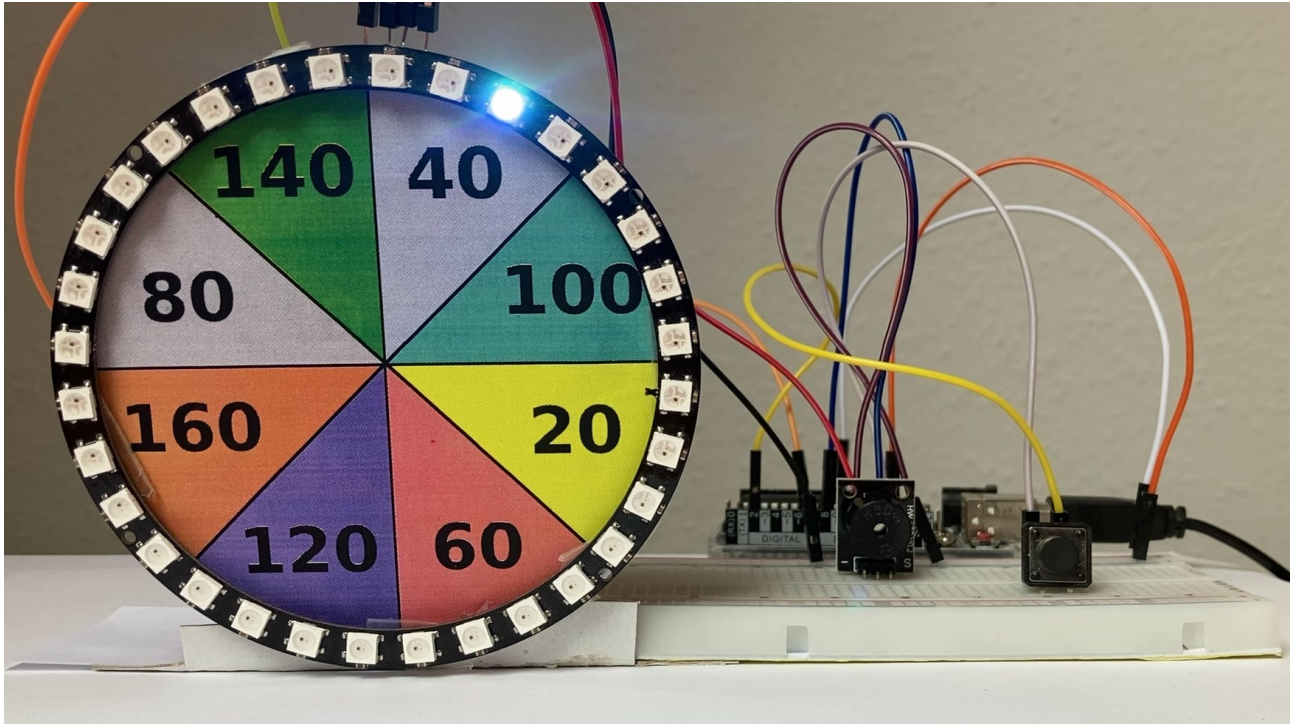


Ziel des Projekts

Ein Neopixel soll als Glücksrad funktionieren. Wird der Taster gedrückt, läuft das Glücksrad, ein erneuter Tasterdruck stoppt es.



Die Hardware



Der NeoPixel-Ring besteht aus mehreren miteinander verbundenen RGB-LEDs. Jede besitzt einen eigenen Controller und kann einzeln angesteuert werden. Er benötigt nur einen digitalen Eingang.

Der NeoPixel-Ring ist in verschiedenen Bauformen zwischen 12 und 60 LEDs erhältlich.

Die Programmierung unterscheidet sich nicht.

RGB ist eine Mischung der Farben Rot, Grün und Blau. Jede Farbe kann von 0 bis 255 gesetzt werden, die Werte werden durch Kommata getrennt.

Beispiele:

163, 0, 93

226, 176, 50

255, 255, 0

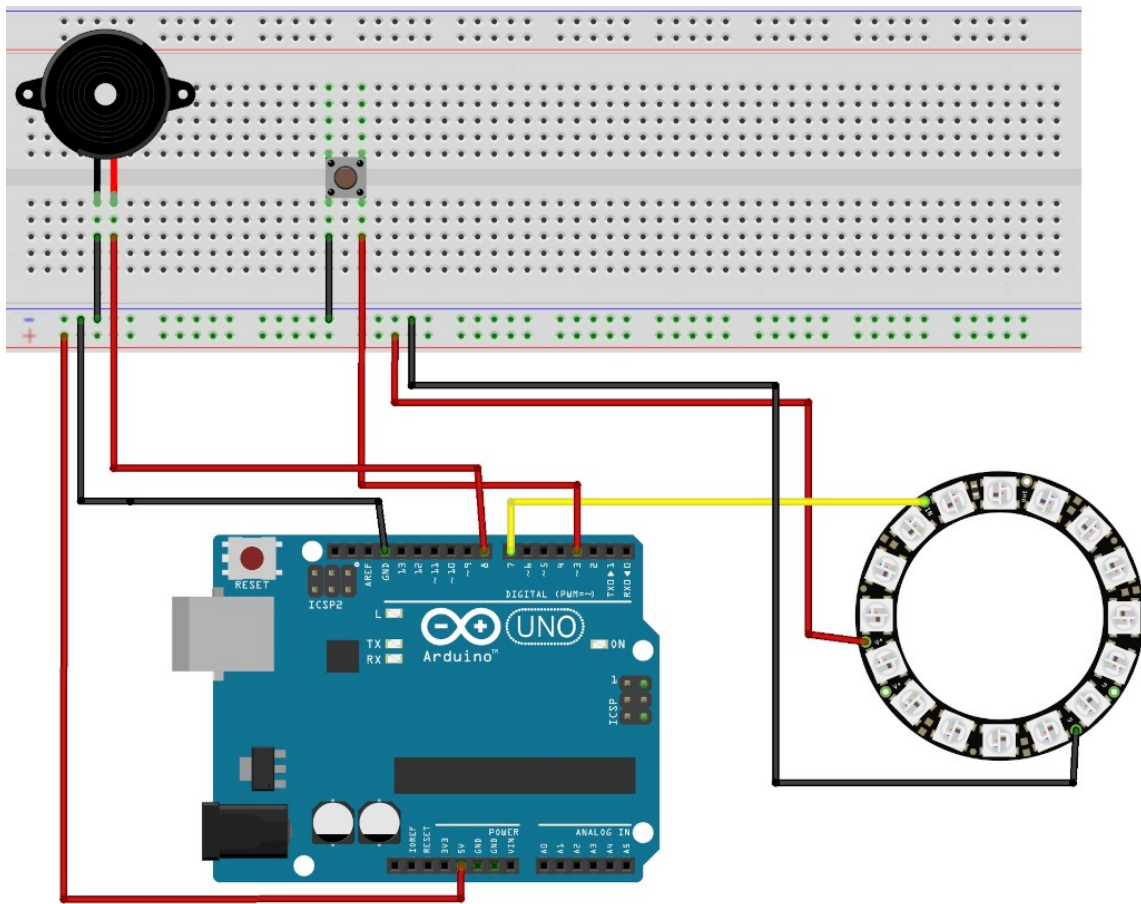
255, 228, 225

Benötigte Bauteile

- ➡ NeoPixel-Ring
- ➡ Taster
- ➡ Lautsprecher
- ➡ Leitungsdrähte



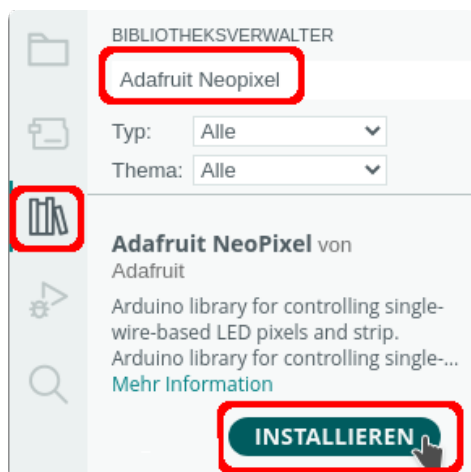
Der Schaltplan



fritzing

Benötigte Bibliothek

Sketch → Bibliothek einbinden → Bibliotheken verwalten



**Übersicht über die Funktionen der Bibliothek Adafruit_NeoPixel (Auswahl)**

Schlüsselwort	Aktion
begin()	LED-Ring starten
numPixels()	Anzahl der LEDs lesen
show()	LED-Ring einschalten
clear()	LED-Ring ausschalten
setPixelColor(Nummer, rot, grün, blau)	Farbe einer LED setzen Nummer → Nummer der LED rot -> 0 - 255 grün -> 0 - 255 blau -> 0 - 255
setBrightness()	Helligkeit setzen (0-255)
Color(rot, grün, blau)	Farbe für alle LEDs setzen rot -> 0 - 255 grün -> 0 - 255 blau -> 0 - 255 Beispiel blau: int Farbe = LEDRing.Color(0, 0, 255);
fill(Farbe, Start, Ende)	Farbe für die mit Start und Ende bezeichneten Pixel setzen



Das Programm

Beispiele

Farbwechsel

```
#include "Adafruit_NeoPixel.h"

#define RING 7

// Anzahl der LEDs -> muss angepasst werden
#define AnzahlLED 32

// LEDRing -> Name des LED-Rings
Adafruit_NeoPixel LEDRing = Adafruit_NeoPixel(AnzahlLED, RING, NEO_GRB + NEO_KHZ800);

void setup()
{
    // setBrightness(0..255)
    LEDRing.setBrightness(200);

    // NeoPixel Bibliothek initialisieren
    LEDRing.begin();
}

void loop()
{
    // Variable für die Farbe festlegen
    int Farbe;

    // rot
    Farbe = LEDRing.Color(255, 0, 0);
    LEDRing.fill(Farbe, 0, AnzahlLED);
    LEDRing.show();
    delay(1000);

    // grün
    Farbe = LEDRing.Color(0, 255, 0);
    LEDRing.fill(Farbe, 0, AnzahlLED);
    LEDRing.show();
    delay(1000);

    // blau
    Farbe = LEDRing.Color(0, 0, 255);
    LEDRing.fill(Farbe, 0, AnzahlLED);
    LEDRing.show();
    delay(1000);

    // gelb
    Farbe = LEDRing.Color(255, 255, 0);
    LEDRing.fill(Farbe, 0, AnzahlLED);
    LEDRing.show();
    delay(1000);
}
```



```
// pink
Farbe = LEDRing.Color(255, 20, 147);
LEDRing.fill(Farbe, 0, AnzahlLED);
LEDRing.show();
delay(1000);

// Pause
LEDRing.clear();
LEDRing.show();
delay(2000);
}
```

Lauflicht

```
#include "Adafruit_NeoPixel.h"

#define RING 7

// Anzahl der LEDs → muss angepasst werden
#define AnzahlLED 32

// LED-Ring -> Name des LED-Rings
Adafruit_NeoPixel LEDRing = Adafruit_NeoPixel(AnzahlLED, RING, NEO_GRB + NEO_KHZ800);

void setup()
{
    // NeoPixel Bibliothek initialisieren
    LEDRing.begin();

    // setBrightness(0..255)
    LEDRing.setBrightness(200);
}

void loop()
{
    // vorwärts mit blau
    for (int LEDNummer = 0; LEDNummer < LEDRing.numPixels(); LEDNummer++)
    {
        LEDRing.setPixelColor(LEDNummer, LEDRing.Color(0, 0, 255));
        LEDRing.show();
        delay(100);
        LEDRing.setPixelColor(LEDNummer, LEDRing.Color(0, 0, 0));
        LEDRing.show();
    }

    // rückwärts mit gelb
    for (int LEDNummer = LEDRing.numPixels(); LEDNummer >= 0 ; LEDNummer--)
    {
        LEDRing.setPixelColor(LEDNummer, LEDRing.Color(255, 255, 0));
        LEDRing.show();
        delay(100);
        LEDRing.setPixelColor(LEDNummer, LEDRing.Color(0, 0, 0));
        LEDRing.show();
    }
}
```



Glücksrad

Bibliothek einbinden und Variablen definieren

```
#include "Adafruit_NeoPixel.h"

#define RING 7

// Anzahl der LEDs -> muss an den verwendeten Neopixel-Ring angepasst werden
#define AnzahlLED 32

// Pin des Lautsprechers
#define LAUTSPRECHER 8

// LED-Ring -> Name des LED-Rings
Adafruit_NeoPixel LEDRing = Adafruit_NeoPixel(AnzahlLED, RING, NEO_GRB +
NEO_KHZ800);

// Status entscheidet, ob das Glücksrad läuft oder gestoppt wird
// false -> Glücksrad läuft beim Start
volatile bool Status = false;

// true -> Lautsprecher ein
// false -> Lautsprecher aus
bool Toene = true;

// true -> zufällige Farbe nach jeder Runde wechseln
// false -> Farbwechsel nach jedem Pixel
bool FarbWechselRunde = false;

// Definition der Farben
int ROT, GRUEN, BLAU;

// Taster am Interrupt-Pin definieren
#define TASTER 3

// Variablen für die mit millis() ermittelte Zeit
// damit nur jeweils ein Tasterdruck erkannt wird
static unsigned long GesicherteStartZeit = 0;
unsigned long Startzeit;

// Zeit zwischen den Bewegungen der LED
int WarteZeit = 50;
```



Der setup-Teil

```
void setup()
{
  // NeoPixel Bibliothek initialisieren
  LEDRing.begin();

  // setBrightness(0..255)
  LEDRing.setBrightness(250);

  // pinMode des Tasters mit eingeschaltetem Vorwiderstand
  pinMode(TASTER, INPUT_PULLUP);

  // Interrupt dem Taster zuordnen und die Methode LEDSchalten aufrufen
  attachInterrupt(digitalPinToInterrupt(TASTER), LEDSchalten, FALLING);

  // Zufallsgenerator starten
  randomSeed(analogRead(0));
}
```

Der loop-Teil. Beachte die Kommentare.

```
void loop()
{
  // LED-Ring ausschalten
  LEDRing.clear();

  // zufällige Farbwerte Wechsel nach jeder 360° Drehung
  // -> FarbWechselRunde true
  if (FarbWechselRunde)
  {
    ROT = ZufallsFarbe();
    GRUEN = ZufallsFarbe();
    BLAU = ZufallsFarbe();
  }

  // wenn Status false -> Glücksrad läuft
  if (!Status)
  {
    if (Toene) tone(LAUTSPRECHER, 1000, 5);

    // Start der Drehung mit LED 0
    // Ende mit Anzahl der LEDs
    for (int i = 0; i < LEDRing.numPixels(); i++)
    {
      // zufällige Farbwerte: Wechsel nach jeder LED
      // -> FarbWechselRunde false
      if (!FarbWechselRunde)
      {
        ROT = ZufallsFarbe();
        GRUEN = ZufallsFarbe();
        BLAU = ZufallsFarbe();
      }
      // Farbe für die aktuelle LED
      LEDRing.setPixelColor(i, ROT, GRUEN, BLAU);
    }
  }
}
```



```
// LEDs einschalten
LEDRing.show();

/*
  Taster gedrückt: Status = true -> Glücksrad stoppen
  break -> for-Schleife verlassen
*/
if (Status)
{
  if (Toene) tone(LAUTSPRECHER, 1000, 500);
  break;
}

// Wartezeit bis zur nächsten LED
delay(WarteZeit);

// aktuell eingeschaltete LEDs ausschalten
LEDRing.setPixelColor(i, 0);
}
}
```

Jetzt fehlen noch die Methoden `LEDSchalten()` und `ZufallsFarbe()`.

```
void LEDSchalten()
{
  // aktuelle Zeit sichern
  Startzeit = millis();

  /*
    bei mehr als 250 Millisekunden Differenz
    zwischen der aktuellen und der gesicherten Zeit
    -> Status umdrehen
    aus false wird true -> Glücksrad stoppen
    aus true wird false -> Glücksrad starten
  */
  if (Startzeit - GesicherteStartZeit > 250)
  {
    Status = !Status;
  }

  GesicherteStartZeit = Startzeit;
}

int ZufallsFarbe()
{
  int Farbe = random(0, 255);
  return Farbe;
}
```