

Programmieren mit Arduino

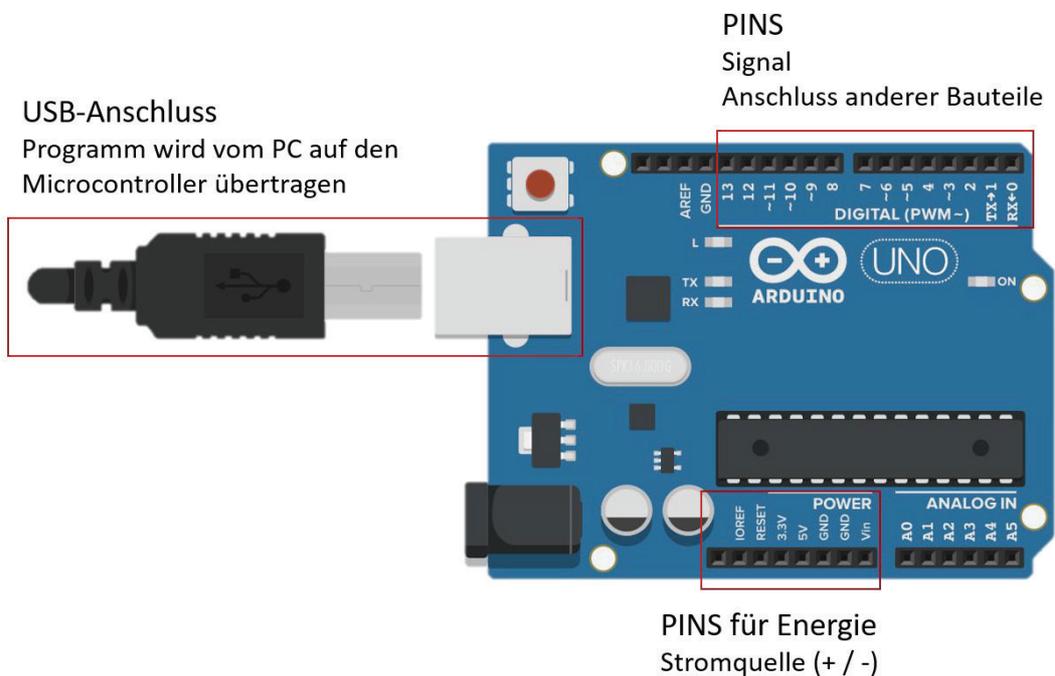
Mit Arduino kannst du einfach Mikrocontroller programmieren und damit zum Beispiel großartige bunte Lichteffekte erzeugen. Was du dafür brauchst, was ein Mikrocontroller genau macht und wie du ihn programmierst erfährst du im folgenden Tutorial.

Material

- 1 Mikrocontroller Arduino Uno
- 1 USB-A auf USB-B Kabel
- 1 Adafruit NeoPixel Ring mit 12 LEDs und angelöteten PINS
- 3 Dupont Kabel männlich auf weiblich

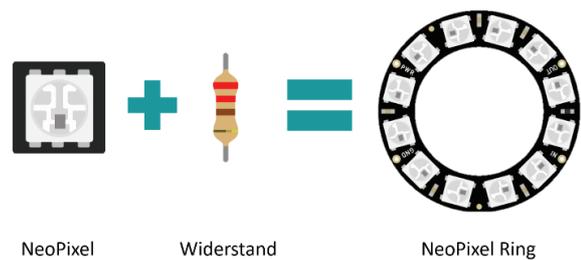
Was ist ein Mikrocontroller?

Ein Mikrocontroller ist eine Art Mini-Computer. Er hat jedoch kein Betriebssystem wie du es von deinem Computer kennst. Daher programmierst du auf deinem PC und überträgst dann einfach das fertige Programm mit dem **USB-Anschluss** auf den Mikrocontroller. An den **Pins** kannst du Komponenten wie LEDs, Schalter oder sogar Motoren anschließen und sie über dein Programm steuern. Es gibt zusätzlich **Pins zur Energieversorgung** deiner Komponenten.



Welche weiteren Bauteile werden genutzt?

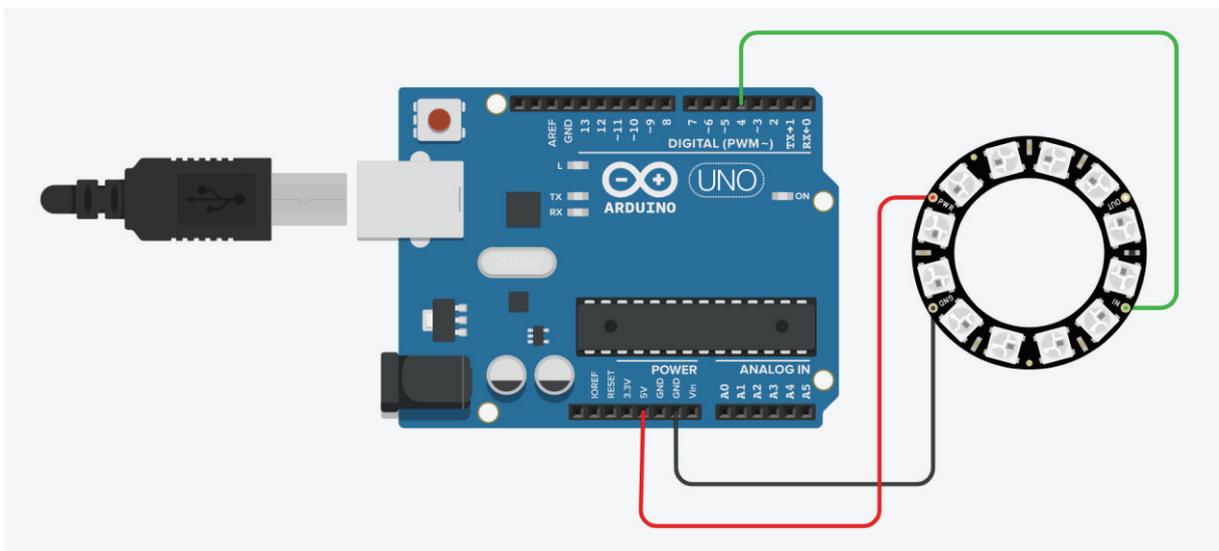
Ein **NeoPixel** ist eine LED die aus drei bis vier kleinen LEDs besteht. Diese haben die Farben Rot, Grün, Blau und manchmal auch zusätzlich Weiß. Durch das Mischen der drei Farben, kann ein **NeoPixel** in allen Farbtönen leuchten.



Ein **NeoPixel** ist ein empfindliches Bauteil, durch das nicht zu viel Strom fließen darf. Daher wird durch den Einbau eines **Widerstands** der Stromfluss begrenzt.

Auf einem **NeoPixel Ring** ist der Widerstand schon angelötet, danach werden mehrere LEDs in Reihe geschaltet. Dadurch kann man einen **NeoPixel Ring** einfach an einen Arduino angeschlossen und programmiert werden, ohne extra Widerstände davor zu schalten.

Wie wird der LED-Ring an den Arduino angeschlossen?



Verbinde folgende Anschlüsse

- PIN 4 -> Data Input
- 5 V -> 5V DC Power
- GND -> GND

Wie wird der LED-Ring programmiert?

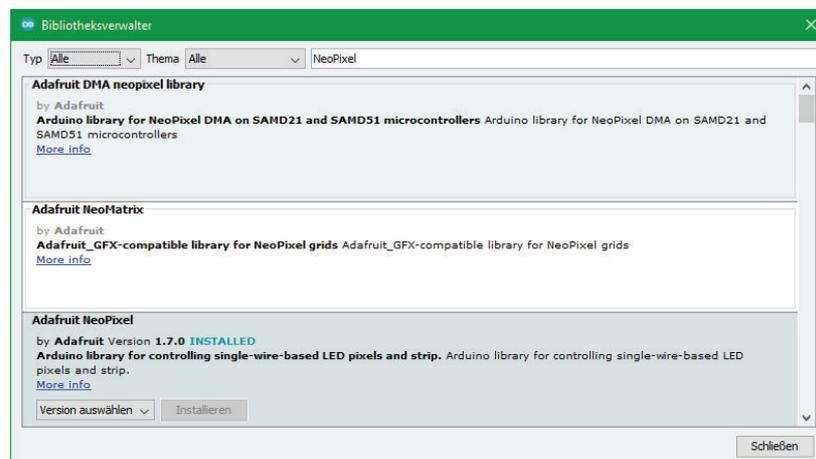
Öffne die Arduino IDE auf deinem PC.

1. Bibliothek verwalten

Eine Bibliothek ist eine Sammlung von fertigem Code, den du ganz einfach zum Programmieren verwenden kannst.

Bibliothek herunterladen

Klicke auf **Sketch -> Bibliothek einbinden -> Bibliothek verwalten**. Ein neues Fenster öffnet sich. Gebe in das Suchfeld oben rechts „NeoPixel“ ein. Wähle dann darunter die Bibliothek „Adafruit NeoPixel“ aus.



Bibliothek einbinden

Klicke auf **Sketch -> Bibliothek einbinden -> Adafruit NeoPixel**. Es erscheint dann eine Codezeile ganz oben auf deinem Bildschirm.

```

Datei Bearbeiten Sketch Werkzeuge Hilfe
sketch_jan07a $
#include <Adafruit_NeoPixel.h>

void setup() {
  // put your setup code here, to run once:
}

void loop() {
  // put your main code here, to run repeatedly:
}
```

Hinweis: Im Programmcode werden Kommentare mit zwei Schrägstrichen „//“ gekennzeichnet. Der PC ignoriert Kommentare beim Ausführen des Codes. Daher kannst du in Kommentaren den Code erklären oder dir z.B. eine Erinnerung schreiben, was du noch in den Code einfügen möchtest.

2. NeoPixel Ring anlegen

LED Pin und Anzahl der LEDs festlegen

Ab jetzt startest du mit dem eigentlichen Programmieren. Zuerst musst du den Pin, an den die Datenleitung angeschlossen ist, festlegen. Auf der obigen Zeichnung ist das die grüne Leitung, die an PIN 4 des Mikrokontrollers angeschlossen ist. In deinem Code legst du den PIN folgendermaßen fest:

```
#define PIXEL_PIN 4
```

Anschließend musst du auch noch die Anzahl der NeoPixel festlegen. Unser NeoPixel Ring hat 12 NeoPixel. In deinem Code legst du die Anzahl der NeoPixel folgendermaßen fest:

```
#define PIXEL_COUNT 12
```

Füge die beiden Zeilen an die im Bild gezeigte Position ein.



```
LED_Ring$
#include <Adafruit_NeoPixel.h>

#define PIXEL_PIN 4 // Festlegen des Daten Pins
#define PIXEL_COUNT 12 // Festlegen der Anzahl an NeoPixel

void setup() {

}

void loop() {

}
```

NeoPixel Ring im Code anlegen

Als nächstes wird der NeoPixel Ring im Code angelegt. Dies ist notwendig, damit du ihn später ansteuern kannst. Für das Anlegen des Rings nutzt du die beiden gerade festgelegten Werte PIXEL_PIN und PIXEL_COUNT. In deinem Code legst du den NeoPixel Ring folgendermaßen an:

```
Adafruit_NeoPixel circle(PIXEL_COUNT, PIXEL_PIN, NEO_GRB + NEO_KHZ800);
```



```
LED_Ring$
#include <Adafruit_NeoPixel.h>

#define PIXEL_PIN 4 // Festlegen des Daten Pins
#define PIXEL_COUNT 12 // Festlegen der Anzahl an NeoPixel

Adafruit_NeoPixel circle(PIXEL_COUNT, PIXEL_PIN, NEO_GRB + NEO_KHZ800); // Anlegen des NeoPixel-Rings

void setup() {

}

void loop() {

}
```

NeoPixel Ring startbereit machen

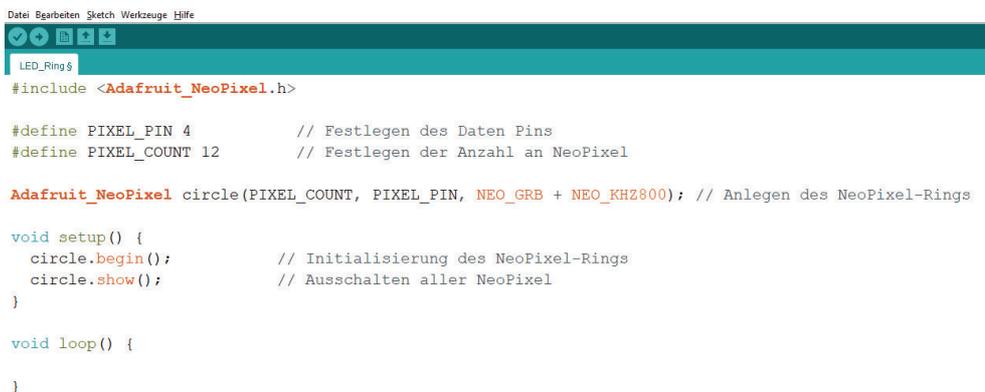
Um den NeoPixel Ring gleich ansteuern zu können, müssen wir ihn erst starten. Dies geschieht in der Funktion `void setup()`, die bereits in deinem Code vorhanden ist. Diese Funktion wird einmal beim Start des Programmes ausgeführt. Schreibe den folgenden Code daher zwischen die geschweiften Klammern der `setup()`-Funktion (du siehst die Position auch im Bild).

Zuerst wird der NeoPixel-Ring gestartet, dies funktioniert mir folgendem Code:

```
circle.begin();
```

Anschließend werden mit dem folgenden Code alle NeoPixel ausgeschaltet:

```
circle.show();
```



```
LED_Ring$
#include <Adafruit_NeoPixel.h>

#define PIXEL_PIN 4           // Festlegen des Daten Pins
#define PIXEL_COUNT 12       // Festlegen der Anzahl an NeoPixel

Adafruit_NeoPixel circle(PIXEL_COUNT, PIXEL_PIN, NEO_GRB + NEO_KHZ800); // Anlegen des NeoPixel-Rings

void setup() {
  circle.begin();           // Initialisierung des NeoPixel-Rings
  circle.show();           // Ausschalten aller NeoPixel
}

void loop() {
}
```

3. NeoPixel programmieren

Jetzt lernst du, wie du einzelne NeoPixel in einer bestimmten Farbe einschalten kannst.

Um die Farbe der LEDs festzulegen, wird sie als RGB-Wert angegeben. In diesem Farbraum wird die Intensität der Farben Rot, Grün und Blau einzeln mit einem Wert zwischen 0 und 255 angegeben. Haben mehrere Farben einen Wert über 0, werden sie gemischt. Die Farbe Rot hat den RGB-Wert (255,0,0), Türkis (0,200,200).

Die NeoPixel werden in der `loop`-Funktion programmiert. Diese wird nach dem Start des Programms immer wieder ausgeführt. Dies geschieht so lange, bis dein Programm gestoppt wird. Dies ermöglicht, dass du später deine NeoPixel animierst, also zum Beispiel blinken lässt.

Um eine NeoPixel in einer bestimmten Farbe einzuschalten, ersetzt du die grauen Werte der folgenden Codezeile einfach mit deinen eigenen Werten:

```
circle.setPixelColor(Nummer der LED, Rot-Wert, Grün-Wert, Blau-Wert);
```

Achtung! In der Informatik beginnt man beim Zählen immer mit 0. Wenn du den ersten NeoPixel einschalten willst, musst du als Nummer des NeoPixels daher 0 angeben. Der letzte NeoPixel ist die Nummer 11.

Die folgende Codezeile würde zum Beispiel den 3. NeoPixel in Blau leuchten lassen:

```
circle.setPixelColor(2, 0, 0, 255);
```

Du kannst die Codezeile beliebig oft wiederholen, um die Farben verschiedener LEDs festzulegen. Im Beispiel wird jede zweite LED in einer jeweils anderen Farbe eingeschaltet. Wenn du alle LED-Farben fertig festgelegt hast, musst du noch den Befehl geben, dass die LEDs eingeschaltet werden. Den Code dafür kennst du schon, füge ihn als letzte Zeile der loop-Funktion ein:

```
circle.show();
```

```
LED_Ring$
#include <Adafruit_NeoPixel.h>

#define PIXEL_PIN 4           // Festlegen des Daten Pins
#define PIXEL_COUNT 12       // Festlegen der Anzahl an NeoPixel

Adafruit_NeoPixel circle(PIXEL_COUNT, PIXEL_PIN, NEO_GRB + NEO_KHZ800); // Anlegen des NeoPixel-Rings

void setup() {
  circle.begin();           // Initialisierung des NeoPixel-Rings
  circle.show();           // Ausschalten aller NeoPixel
}

void loop() {
  circle.setPixelColor(0, 255, 0, 0); // 1. NeoPixel: Rot
  circle.setPixelColor(2, 255, 255, 0); // 3. NeoPixel: Gelb
  circle.setPixelColor(4, 0, 255, 0); // 5. NeoPixel: Grün
  circle.setPixelColor(6, 0, 255, 255); // 7. NeoPixel: Türkies
  circle.setPixelColor(8, 0, 0, 255); // 9. NeoPixel: Blau
  circle.setPixelColor(10, 255, 0, 255); // 11. NeoPixel: Magenta

  circle.show();           // Ausschalten aller vorher festgelegten NeoPixel
}
```

4. Programm auf den Mikrokontroller übertragen

Mikrocontroller mit dem PC verbinden

Verbinde den Mikrocontroller durch das USB-Kabel mit deinem Computer. Danach kannst du in der Arduino IDE den Typ des verbundenen Mikrocontrollers auswählen. Klicke dafür auf **Werkzeuge** -> **Board: „** -> **Arduino Uno**.

Danach musst du noch auswählen, an welchem Port deines PCs der Mikrocontroller angeschlossen ist. Klicke dafür auf Werkzeuge -> COM... . Oft wird hier nur ein Port angezeigt, falls es mehrere gibt probiere die folgenden Schritte mit verschiedenen Ports aus, bis das Hochladen funktioniert.

Programm hochladen

Klicke jetzt auf den Button, du findest ihn auf deinem Display oben links. Der Code wird jetzt auf den Mikrocontroller übertragen, dies kann einen Moment dauern. Wenn alles funktioniert, sollen dann die von dir festgelegten NeoPixel aufleuchten. Falls du noch einen Fehler im Code hast, wird die Zeile mit dem Fehler rot hervorgehoben. Verbessere den Fehler und klicke dann nochmal auf den Hochladen-Button.



Beispielcode

Regenbogen

Das folgende Programm erzeugt einen Farbverlauf über alle NeoPixel.

```

Datei Bearbeiten Sketch Werkzeuge Hilfe
LED_Ring $
#include <Adafruit_NeoPixel.h>

#define PIXEL_PIN 4           // Festlegen des Daten Pins
#define PIXEL_COUNT 12       // Festlegen der Anzahl an NeoPixel

Adafruit_NeoPixel circle(PIXEL_COUNT, PIXEL_PIN, NEO_GRB + NEO_KHZ800); // Anlegen des NeoPixel-Rings

void setup() {
  circle.begin();           // Initialisierung des NeoPixel-Rings
  circle.show();            // Ausschalten aller NeoPixel
}

void loop() {
  circle.setPixelColor(0, 255, 0, 0); // 1. NeoPixel: Rot
  circle.setPixelColor(1, 255, 128, 0); // 2. NeoPixel: Orange
  circle.setPixelColor(2, 255, 255, 0); // 3. NeoPixel: Gelb
  circle.setPixelColor(3, 128, 255, 0); // 4. NeoPixel: Hellgrün
  circle.setPixelColor(4, 0, 255, 0); // 5. NeoPixel: Grün
  circle.setPixelColor(5, 0, 255, 128); // 6. NeoPixel: Blaugrün
  circle.setPixelColor(6, 0, 255, 255); // 7. NeoPixel: Türkies
  circle.setPixelColor(7, 0, 128, 255); // 8. NeoPixel: Hellblau
  circle.setPixelColor(8, 0, 0, 255); // 9. NeoPixel: Blau
  circle.setPixelColor(9, 128, 0, 255); // 10. NeoPixel: Violett
  circle.setPixelColor(10, 255, 0, 255); // 11. NeoPixel: Magenta
  circle.setPixelColor(11, 255, 0, 128); // 12. NeoPixel: Pink
}

```



Blinken

Durch den Aufruf der Funktion `delay(500);` wartet der PC die in Klammern angegebene Anzahl an Millisekunden, bis er den Code weiter ausführt. Im Beispiel würde er also 500 ms, also eine halbe Sekunde warten. Delays ermöglichen das Programmieren einfacher Animationen.

Im folgenden Programm leuchten alle NeoPixel abwechselnd in Rot und in Blau auf.

```

Datei Bearbeiten Sketch Werkzeuge Hilfe
LED_Ring$
#include <Adafruit_NeoPixel.h>

#define PIXEL_PIN 4           // Festlegen des Daten Pins
#define PIXEL_COUNT 12       // Festlegen der Anzahl an NeoPixel

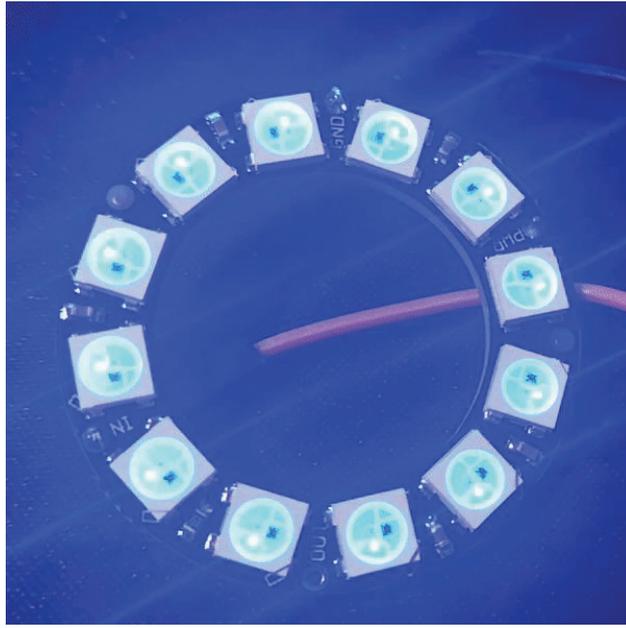
Adafruit_NeoPixel circle(PIXEL_COUNT, PIXEL_PIN, NEO_GRB + NEO_KHZ800); // Anlegen des NeoPixel-Rings

void setup() {
  circle.begin();           // Initialisierung des NeoPixel-Rings
  circle.show();           // Ausschalten aller NeoPixel
}

void loop() {
  circle.setPixelColor(0, 0, 0, 255); // 1. NeoPixel: Blau
  circle.setPixelColor(1, 0, 0, 255); // 2. NeoPixel: Blau
  circle.setPixelColor(2, 0, 0, 255); // 3. NeoPixel: Blau
  circle.setPixelColor(3, 0, 0, 255); // 4. NeoPixel: Blau
  circle.setPixelColor(4, 0, 0, 255); // 5. NeoPixel: Blau
  circle.setPixelColor(5, 0, 0, 255); // 6. NeoPixel: Blau
  circle.setPixelColor(6, 0, 0, 255); // 7. NeoPixel: Blau
  circle.setPixelColor(7, 0, 0, 255); // 8. NeoPixel: Blau
  circle.setPixelColor(8, 0, 0, 255); // 9. NeoPixel: Blau
  circle.setPixelColor(9, 0, 0, 255); // 10. NeoPixel: Blau
  circle.setPixelColor(10, 0, 0, 255); // 11. NeoPixel: Blau
  circle.setPixelColor(11, 0, 0, 255); // 12. NeoPixel: Blau
  circle.show();           // Ein-/Ausschalten aller vorher festgelegten NeoPixel
  delay(500);              // Pause für 500 ms

  circle.setPixelColor(0, 255, 0, 0); // 1. NeoPixel: Rot
  circle.setPixelColor(1, 255, 0, 0); // 2. NeoPixel: Rot
  circle.setPixelColor(2, 255, 0, 0); // 3. NeoPixel: Rot
  circle.setPixelColor(3, 255, 0, 0); // 4. NeoPixel: Rot
  circle.setPixelColor(4, 255, 0, 0); // 5. NeoPixel: Rot
  circle.setPixelColor(5, 255, 0, 0); // 6. NeoPixel: Rot
  circle.setPixelColor(6, 255, 0, 0); // 7. NeoPixel: Rot
  circle.setPixelColor(7, 255, 0, 0); // 8. NeoPixel: Rot
  circle.setPixelColor(8, 255, 0, 0); // 9. NeoPixel: Rot
  circle.setPixelColor(9, 255, 0, 0); // 10. NeoPixel: Rot
  circle.setPixelColor(10, 255, 0, 0); // 11. NeoPixel: Rot
  circle.setPixelColor(11, 255, 0, 0); // 12. NeoPixel: Rot
  circle.show();           // Ein-/Ausschalten aller vorher festgelegten NeoPixel
  delay(500);              // Pause für 500 ms
}

```



Lauflicht Halbkreis

Im folgenden Programm werden die NeoPixel so programmiert, dass ein Licht sich scheinbar über mehrere LEDs bewegt.

```

Datei Bearbeiten Sketch Werkzeuge Hilfe
LED_Ring
#include <Adafruit_NeoPixel.h>

#define PIXEL_PIN 4           // Festlegen des Daten Pins
#define PIXEL_COUNT 12       // Festlegen der Anzahl an NeoPixel

Adafruit_NeoPixel circle(PIXEL_COUNT, PIXEL_PIN, NEO_GRB + NEO_KHZ800); // Anlegen des NeoPixel-Rings

void setup() {
  circle.begin();           // Initialisierung des NeoPixel-Rings
  circle.show();            // Ausschalten aller NeoPixel
}

void loop() {
  circle.setPixelColor(5, 0, 0, 0); // 6. NeoPixel: Schwarz (Aus)
  circle.setPixelColor(0, 255, 0, 0); // 1. NeoPixel: Rot
  delay(300); // Pause für 300 ms
  circle.show(); // Ein-/Ausschalten der festgelegten NeoPixel

  circle.setPixelColor(0, 0, 0, 0); // 1. NeoPixel: Schwarz (Aus)
  circle.setPixelColor(1, 255, 0, 0); // 2. NeoPixel: Rot
  delay(300); // Pause für 300 ms
  circle.show(); // Ein-/Ausschalten der festgelegten NeoPixel

  circle.setPixelColor(1, 0, 0, 0); // 2. NeoPixel: Schwarz (Aus)
  circle.setPixelColor(2, 255, 0, 0); // 3. NeoPixel: Rot
  delay(300); // Pause für 300 ms
  circle.show(); // Ein-/Ausschalten der festgelegten NeoPixel

  circle.setPixelColor(2, 0, 0, 0); // 3. NeoPixel: Schwarz (Aus)
  circle.setPixelColor(3, 255, 0, 0); // 4. NeoPixel: Rot
  delay(300); // Pause für 300 ms
  circle.show(); // Ein-/Ausschalten der festgelegten NeoPixel

  circle.setPixelColor(3, 0, 0, 0); // 4. NeoPixel: Schwarz (Aus)
  circle.setPixelColor(4, 255, 0, 0); // 5. NeoPixel: Rot
  delay(300); // Pause für 300 ms
  circle.show(); // Ein-/Ausschalten der festgelegten NeoPixel

  circle.setPixelColor(4, 0, 0, 0); // 5. NeoPixel: Schwarz (Aus)
  circle.setPixelColor(5, 255, 0, 0); // 6. NeoPixel: Rot
  delay(300); // Pause für 300 ms
  circle.show(); // Ein-/Ausschalten der festgelegten NeoPixel
}

```

