

Quick-Start

LED Streifen



Inhaltsverzeichnis

Einführung	4
Spezifikation	5
Stromversorgung der LEDs	5
Auswahl eines geeigneten Netzteils	5
Verkabelung und Verbindungen	5
Schritte zur Verbindung:	6
Wichtige Hinweise:	6
Spannungsabfall und Leistung Injektion	6
Testen Ihrer Einrichtung	7
Fehlersuche bei allgemeinen Problemen	8
Verwenden Sie die LED-Streifen mit einem	9
Arduino-kompatiblen Board	9
Dinge, die Sie im Auge behalten sollten	9
Anschluss Diagramm	11
Beispiel code	12
Einbeziehung der Bibliothek:	15
Definitionen der Pins und LED-Zahlen:	15
Erstellen des NeoPixel-Objekts:	15
Funktion einrichten:	15
Schleifenfunktion:	15
Verwendung der LED-Streifen mit ESP32/ESP8266 (WLED)	16
Herunterladen der WLED Firmware	17
Flashen der WLED Firmware	17
Anschluss Diagramm	18
Verwendung der LED-Streifen mit einem Raspberry Pi	19
Anschluss Diagramm	20
Einrichten der virtuellen Python-Umgebung	21
Installieren Sie die benötigten Bibliotheken	21
Beispiel Code	22

Einführung

Willkommen bei der Schnellstartanleitung für HC2812B LED-Streifen! Egal, ob Sie Bastler, Macher oder Profi sind, diese Anleitung wird Ihnen helfen, das volle Potenzial dieser vielseitigen und lebendigen LED-Streifen zu erschließen. HC2812B LEDs sind einzeln adressierbare RGB-LEDs, mit denen Sie mühelos atemberaubende visuelle Effekte, Animationen und dynamische Beleuchtungsprojekte erstellen können.

In diesem Leitfaden erfahren Sie alles, was Sie wissen müssen, um loszulegen:

- **Spezifikationen:** Verstehen Sie die wichtigsten Merkmale und technischen Details Ihrer LED-Streifen.
- **Die Stromversorgung der LEDs:** Erfahren Sie, wie Sie Ihre LED-Streifen sicher und effizient mit Strom versorgen, um eine optimale Leistung zu gewährleisten.
- **Verwenden Sie die LED-Streifen mit einem Arduino:** Schritt-für-Schritt-Anleitung zur Steuerung Ihrer LEDs mit einem Arduino-Mikrocontroller.
- **Verwendung der LED-Streifen mit ESP32/ESP8266 (WLED):** Erfahren Sie, wie Sie die WLED-Firmware auf ESP32/ESP8266-Boards für erweiterte Lichteffekte und Wi-Fi-Steuerung verwenden können.
- **Verwendung der LED-Streifen mit einem Raspberry Pi:** Entdecken Sie, wie Sie Ihre LED-Streifen für komplexere Projekte mit einem Raspberry Pi verbinden können.
- **Zusätzliche Tipps:** Erhalten Sie Einblicke und Best Practices, um Ihre Erfahrung mit LED-Streifen zu verbessern.

Am Ende dieses Leitfadens werden Sie mit dem Wissen und dem Selbstvertrauen ausgestattet sein, Ihre kreativen Beleuchtungsideen zum Leben zu erwecken. Lassen Sie uns Ihre Projekte beleuchten und tauchen Sie ein in die spannende Welt der HC2812B LED-Streifen!

Spezifikation

Modell	Typ	IP-Klasse	Länge	Leistung	Lichtleistung	Spannung	Current	Lebensdauer
SL-5050-60HC2812B	60LEDs/M	20	1m	12W/m	80lm	5V	40mA/LED	50000h
SL-5050-144HC2812B	144LEDs/M	20	1m	28.8W/m	80lm	5V	40mA/LED	50000h
SL-5050-60HC2812B	60LEDs/M	65	1m	12W/m	80lm	5V	40mA/LED	50000h
SL-5050-60HC2812B	60LEDs/M	66	1m	12W/m	80lm	5V	40mA/LED	50000h
SL-5050-30HC2812B	30LEDs/M	20	1m	6W/m	80lm	5V	40mA/LED	50000h
SL-5050-60HC2812B	60LEDs/M	20	5m	12W/m	80lm	5V	40mA/LED	50000h
SL-5050-144HC2812B	144LEDs/M	20	5m	28.8W/m	80lm	5V	40mA/LED	50000h
SL-5050-30HC2812B	30LEDs/M	20	5m	6W/m	80lm	5V	40mA/LED	50000h
SL-5050-60HC2812B	60LEDs/M	65	5m	12W/m	80lm	5V	40mA/LED	50000h
SL-5050-60HC2812B	60LEDs/M	66	5m	12W/m	80lm	5V	40mA/LED	50000h

Stromversorgung der LEDs

Der erste Schritt bei der Stromversorgung Ihrer LED-Streifen besteht darin, deren Stromverbrauch zu verstehen:

- **Stromverbrauch:** Unsere Variante mit 30 LEDs/m verbraucht **6 Watt pro Meter**.
- **Erforderliche Spannung:** Die HC2812B LEDs arbeiten mit **5 Volt DC**.
- **Stromaufnahme:** Da Leistung (W) = Spannung (V) × Strom (A), zieht jedes Messgerät: Strom = Leistung/Spannung = $6\text{W}/5\text{V} = 1,2\text{A}$

Berechnungsbeispiel: Wenn Sie einen 5-Meter-Streifen haben:

- **Gesamtleistung:** $6\text{ W/m} \times 5\text{ m} = 30\text{ W}$
- **Gesamtstrom:** $1,2\text{ A/m} \times 5\text{ m} = 6\text{ A}$

Auswahl eines geeigneten Netzteils

Bei der Auswahl eines Netzteils sollten Sie Folgendes beachten:

- **Nennspannung:** Muss **5V DC** betragen, um den Anforderungen des LED-Streifens zu entsprechen.
- **Stromstärke:** Sollte gleich oder größer sein als die Gesamtstromaufnahme Ihres LED-Streifens.
- **Reserven:** Es ist ratsam, ein Netzteil zu wählen, dessen **Kapazität** mindestens **20% über** der berechneten Stromaufnahme **liegt**, um eine Überlastung zu vermeiden.

Beispiel: Für einen Streifen, der 6A benötigt, wählen Sie ein Netzteil, das mindestens für **7.2A**.

Verkabelung und Verbindungen

Benötigte Materialien:

- **Stromversorgung:** Passend zu Ihren Anforderungen an Spannung und Stromstärke.

- **Drähte:** Ausreichende Stärke für den Strom (z.B. 18 AWG für bis zu 5A).
- **Steckverbinder:** Optional, für einfaches Trennen und Wartung.
- **Sicherung:** Zur Sicherheit empfohlen, um Überstromsituationen zu vermeiden.

Schritte zur Verbindung:

1. Bereiten Sie die Drähte vor:

- **Positiv (+):** Normalerweise rot.
- **Negativ (-):** Normalerweise schwarz oder weiß.

2. Schließen Sie den LED-Streifen an:

- **Positives Kabel (+5V):** Verbinden Sie es mit dem 5V-Pad auf dem Streifen.
- **Negatives Kabel (GND):** Verbinden Sie es mit dem GND-Pad auf dem Streifen.

3. Schließen Sie das Gerät an das Stromnetz an:

- **Positives Kabel (+5V):** Schließen Sie es an den positiven Anschluss an.
- **Negatives Kabel (GND):** Schließen Sie es an den Minuspol an.

4. Sichere Verbindungen:

- Verwenden Sie Lötstellen oder geeignete Stecker, um sichere Verbindungen zu gewährleisten.
- Isolieren Sie freiliegende Drähte mit Schrumpfschlauch oder Isolierband.

Wichtige Hinweise:

- **Gemeinsame Masse:** Stellen Sie sicher, dass die GND des LED-Streifens und des Mikrocontrollers mit der gleichen Masse verbunden sind, um Signalprobleme zu vermeiden.
- **Die Polarität ist wichtig:** Wenn Sie die Spannung vertauschen, können die LEDs beschädigt werden. Überprüfen Sie Ihre Anschlüsse doppelt.

Spannungsabfall und Leistung Injektion

Bei längeren Streifen kann es zu einem Spannungsabfall kommen, der dazu führt, dass die weiter von der Stromversorgung entfernten LEDs schwächer

erscheinen oder ihre Farbe nicht richtig wechseln.

Lösungen:

- **Stromeinspeisung:** Versorgen Sie mehrere Punkte entlang des Streifens mit Strom.
- **Stärkere Drähte:** Verwenden Sie dickere Drähte, um den Widerstand zu verringern.
- **Kürzere Segmente:** Teilen Sie Ihren LED-Streifen in kürzere Segmente auf, die jeweils über eine eigene Stromversorgung verfügen.

Power Injection Beispiel:

- **Alle 5 Meter:** Bei Streifen, die länger als 5 Meter sind, sollten Sie alle 5 Meter Strom einspeisen.
- **Beide Enden:** Bei Streifen mittlerer Länge schließen Sie das Netzteil an beide Enden des Streifens an.

Sicherheit Vorsichtsmaßnahmen

- **Vermeiden Sie Überlastung:** Überschreiten Sie nicht die maximale Stromstärke des Netzteils.
- **Wärmeableitung:** Hohe Ströme können Wärme erzeugen. Sorgen Sie für ausreichende Belüftung.
- **Sicherungen und Schutz:** Integrieren Sie Sicherungen zum Schutz vor Kurzschlüssen und Überstrom.
- **Verwendung in Innenräumen:** Sofern nicht anders angegeben, sind die meisten LED-Leisten und Netzteile für die Verwendung in Innenräumen vorgesehen.

Testen Ihrer Einrichtung

Bevor Sie Ihr komplettes Beleuchtungsprogramm ausführen:

1. Erster Test:

- Schalten Sie das System mit einer einfachen Farbe ein (z.B. alle LEDs rot), um die Einheitlichkeit zu testen.

2. Prüfen Sie auf Probleme:

- Achten Sie auf flackernde, uneinheitliche Farben oder schwache LEDs.

3. Temperatur überwachen:

- Prüfen Sie nach einer Weile des Betriebs, ob irgendwelche Komponenten überhitzt sind.

Fehlersuche bei allgemeinen Problemen

- **LEDs leuchten nicht auf:**
 - Überprüfen Sie alle Stromanschlüsse und stellen Sie sicher, dass die Stromversorgung funktioniert.
- **Farbverschiebungen oder Flackern:**
 - Kann auf einen Spannungsabfall hindeuten; erwägen Sie eine Stromeinspeisung.
- **Mikrocontroller-Resets:**
 - Könnte an Problemen mit der Stromversorgung liegen. Stellen Sie sicher, dass der Mikrocontroller richtig mit Strom versorgt und geerdet ist.

Verwenden Sie die LED-Streifen mit einem Arduino-kompatiblen Board

In diesem Abschnitt erfahren Sie, wie Sie Ihren LED-Streifen HC2812B an einen Arduino-kompatiblen Mikrocontroller anschließen und erhalten einen Beispielcode für den Einstieg. Sie werden lernen, wie man:

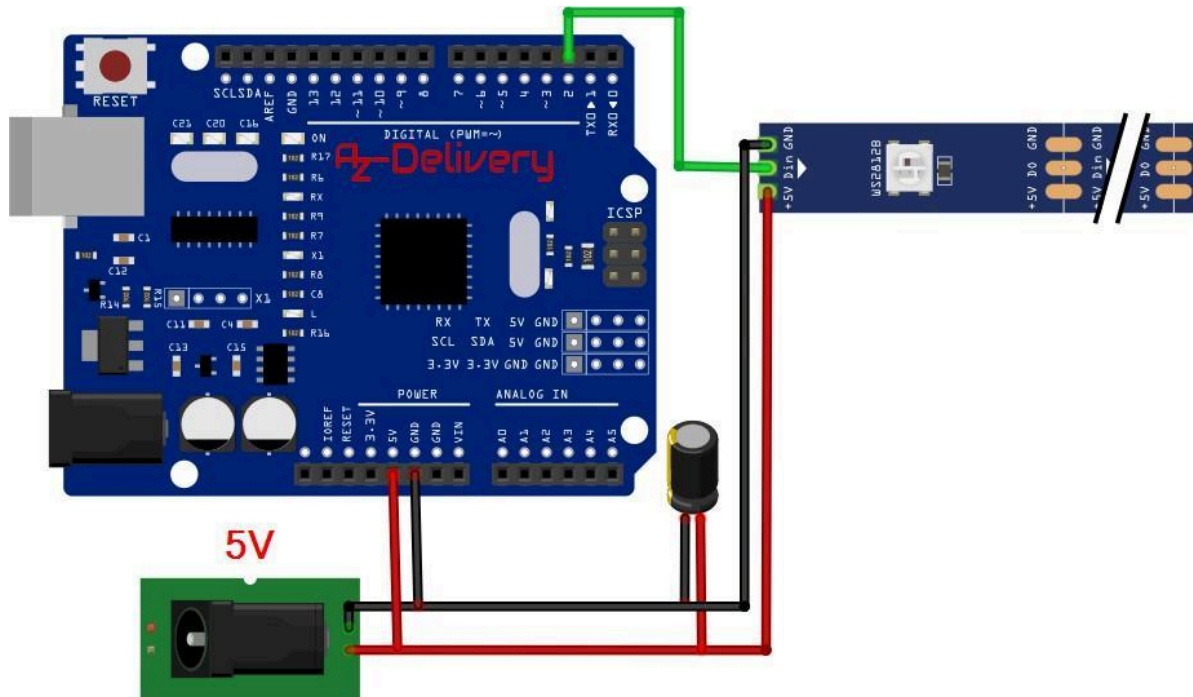
- **Richten Sie die Hardware ein:** Verkabeln Sie den LED-Streifen ordnungsgemäß mit dem Mikrocontroller, um einen sicheren und zuverlässigen Betrieb zu gewährleisten.
- **Installieren Sie die erforderlichen Bibliotheken:** Verwenden Sie Bibliotheken wie **Adafruit NeoPixel**, um die Codierung zu vereinfachen.
- **Code schreiben und hochladen:** Steuern Sie die LEDs mit einfachen Skripts, um verschiedene Lichteffekte zu erzeugen.

Dinge, die Sie im Auge behalten sollten

- **Überlegungen zur Stromversorgung:**
 - **Betreiben Sie die LEDs nicht über den Mikrocontroller:** Der 5V-Pin des Mikrocontrollers kann den für den LED-Streifen benötigten Strom nicht liefern. Verwenden Sie immer eine externe 5V-Stromversorgung für die LEDs.
 - **Gemeinsame Masse:** Verbinden Sie die Masse (GND) des Mikrocontrollers und die Stromversorgung des LED-Streifens, um eine ordnungsgemäße Signalübertragung zu gewährleisten.
- **Komponenten zur Rauschunterdrückung:**
 - **Entkopplungskondensator:** Legen Sie einen großen Kondensator (z.B. 1000 μ F, 6,3 V oder höher) über die Stromversorgungsleitungen des LED-Streifens, um Spannungsschwankungen auszugleichen.
- **Programmiertipps:**
 - **Verwenden Sie etablierte Bibliotheken:** Bibliotheken wie Adafruit NeoPixel oder FastLED erfüllen die komplexen Timing-Anforderungen von HC2812B LEDs.
 - **Achten Sie auf das Timing der Daten:** HC2812B LEDs

erfordern ein präzises Timing. Vermeiden Sie die übermäßige Verwendung von Funktionen wie "delay()", da diese die Aktualisierung der LEDs beeinträchtigen können.

Anschluss Diagramm



Beispiel code

```
#include <Adafruit_NeoPixel.h>

#define LED_PIN      2      // Digital IO pin connected to
the NeoPixels.
#define LED_COUNT    30     // Number of NeoPixels.

Adafruit_NeoPixel strip(LED_COUNT, LED_PIN, NEO_GRB +
NEO_KHZ800);

void setup() {
    strip.begin();           // INITIALIZE NeoPixel strip
object (REQUIRED).
    strip.show();           // Turn OFF all pixels ASAP.
    strip.setBrightness(50); // Set BRIGHTNESS to about 20%
(0 to 255).
}

void loop() {
    colorWipe(strip.Color(255,  0,  0), 50); // Red
    colorWipe(strip.Color(  0, 255,  0), 50); // Green
    colorWipe(strip.Color(  0,  0, 255), 50); // Blue
    theaterChase(strip.Color(127, 127, 127), 50); // White
    rainbow(10);
    rainbowCycle(10);
    theaterChaseRainbow(50);
}

// Fill the dots one after the other with a color.
void colorWipe(uint32_t color, int wait) {
    for(int i=0; i<strip.numPixels(); i++) {
        strip.setPixelColor(i, color);
    }
}
```

```
        strip.show();
        delay(wait);
    }
}

// Theater-style crawling lights.
void theaterChase(uint32_t color, int wait) {
    for(int j=0; j<10; j++) { // Repeat 10 times.
        for(int q=0; q<3; q++) {
            for(int i=0; i<strip.numPixels(); i+=3) {
                strip.setPixelColor(i+q, color); // Turn every
third pixel on.
            }
            strip.show();
            delay(wait);
            for(int i=0; i<strip.numPixels(); i+=3) {
                strip.setPixelColor(i+q, 0); // Turn every
third pixel off.
            }
        }
    }
}

// Rainbow cycle along the whole strip.
void rainbow(int wait) {
    for(long firstPixelHue = 0; firstPixelHue < 5*65536;
firstPixelHue += 256) {
        strip.rainbow(firstPixelHue);
        strip.show();
        delay(wait);
    }
}

// Slightly different, this makes the rainbow equally
distributed throughout.
```

```
void rainbowCycle(int wait) {
    for(long firstPixelHue = 0; firstPixelHue < 5*65536;
firstPixelHue += 256) {
        for(int i=0; i<strip.numPixels(); i++) {
            int pixelHue = firstPixelHue + (i * 65536L /
strip.numPixels());
            strip.setPixelColor(i,
strip.gamma32(strip.ColorHSV(pixelHue)));
        }
        strip.show();
        delay(wait);
    }
}

// Theatre-style crawling lights with rainbow effect.
void theaterChaseRainbow(int wait) {
    for(int j=0; j<256; j++) {        // Cycle all 256 colors
in the wheel.
        for(int q=0; q<3; q++) {
            for(int i=0; i<strip.numPixels(); i+=3) {
                int pixelHue = (i * 65536L / strip.numPixels()) +
(j * 256);
                strip.setPixelColor(i+q,
strip.gamma32(strip.ColorHSV(pixelHue)));
            }
            strip.show();
            delay(wait);
            for(int i=0; i<strip.numPixels(); i+=3) {
                strip.setPixelColor(i+q, 0);        // Turn every
third pixel off.
            }
        }
    }
}
```

Einbeziehung der Bibliothek:

- `#include <Adafruit_NeoPixel.h>` enthält die notwendige Bibliothek zur Steuerung der NeoPixels.

Definitionen der Pins und LED-Zahlen:

- `LED_PIN` gibt den Arduino-Pin an, der mit der Datenleitung des LED-Streifens verbunden ist.
- `LED_COUNT` ist die Anzahl der LEDs auf Ihrem Streifen.

Erstellen des NeoPixel-Objekts:

- `Adafruit_NeoPixel strip(LED_COUNT, LED_PIN, NEO_GRB + NEO_KHZ800);` initialisiert den Streifen.
 - `NEO_GRB` gibt die Farbreihenfolge an; einige Streifen verwenden möglicherweise `NEO_RGB`.
 - `NEO_KHZ800` spezifiziert die Kommunikationsgeschwindigkeit für HC2812B LEDs.

Funktion einrichten:

- `strip.begin();` initialisiert die Bibliothek.
- `strip.show();` löscht alle Daten und schaltet alle LEDs aus.
- `strip.setBrightness(50);` passt die Helligkeitsstufe an (0-255).

Schleifenfunktion:

- Ruft verschiedene Funktionen auf, um verschiedene Lichteffekte anzuzeigen:
 - `colorWipe()` füllt den Streifen Farbe für Farbe.
 - `theaterChase()` erzeugt einen bewegten Lichteffekt.
 - `rainbow()`, `rainbowCycle()`, und `theaterChaseRainbow()` zeigen Regenbogenmuster an.

Verwendung der LED-Streifen mit ESP32/ESP8266 (WLED)

Die drahtlose Steuerung von HC2812B LED-Streifen eröffnet Ihnen neue Möglichkeiten für Ihre Beleuchtungsprojekte. Hier kommt **WLED** ins Spiel - eine leistungsstarke und benutzerfreundliche Firmware, die speziell für ESP32 und ESP8266 Mikrocontroller entwickelt wurde. WLED verwandelt diese erschwinglichen Wi-Fi-fähigen Boards in funktionsreiche LED-Controller.

Was ist WLED?

WLED ist eine Open-Source-Firmware, die entwickelt wurde, um adressierbare LEDs wie HC2812B über Wi-Fi zu steuern. Sie verwandelt ein ESP32- oder ESP8266-Board in einen vielseitigen LED-Controller, der über ein Webinterface, eine Smartphone-App oder sogar Sprachassistenten wie Amazon Alexa und Google Assistant gesteuert werden kann.

Hauptmerkmale von WLED

- **Benutzerfreundliches Interface:** Steuern Sie Ihre LEDs über einen Webbrowser oder die WLED-App mit Schiebereglern, Farbwählern und Effektwählern.
- **Anpassbare Effekte:** Über 100 vordefinierte Effekte und die Möglichkeit, eigene Sequenzen zu erstellen.
- **Segment-Steuerung:** Unterteilen Sie Ihren LED-Streifen in mehrere Segmente, jedes mit unabhängigen Einstellungen.
- **Zeitplanung und Timer:** Automatisieren Sie Ihre Lichteffekte zeit- oder ereignisabhängig.
- **OTA-Updates:** Aktualisieren Sie die Firmware ganz einfach Over-the-Air, ohne physischen Zugriff auf das Gerät.
- **Energie-Effizienz:** Enthält Optionen zur Begrenzung der Helligkeit und des Stromverbrauchs, um die Lebensdauer Ihrer LEDs zu verlängern.

Herunterladen der WLED Firmware

Rufen Sie die Seite <https://github.com/Aircoookie/WLED/releases> auf, um die neueste Firmware-Version herunterzuladen. Sie müssen die richtige Firmware-Variante auswählen.

Für ESP8266-Boards (wie NodeMCU, D1 Mini) laden Sie "WLED_0.x.x_ESP8266.bin" herunter.

Für ESP32-Boards laden Sie "WLED_0.x.x_ESP32.bin" herunter.

Stellen Sie sicher, dass Sie die Firmware auswählen, die zu Ihrem Board passt, um Kompatibilitätsprobleme zu vermeiden.

ESPHome Flasher herunterladen:

Für ein einfaches Hochladen der Firmware empfehlen wir die Verwendung des ESPHome Flasher.

<https://github.com/esphome/esphome-flasher/releases>

Laden Sie die neueste Version herunter, die zu Ihrem Betriebssystem passt. Abhängig von Ihrem Board benötigen Sie möglicherweise USB-Treiber wie CH340 oder CP2102.

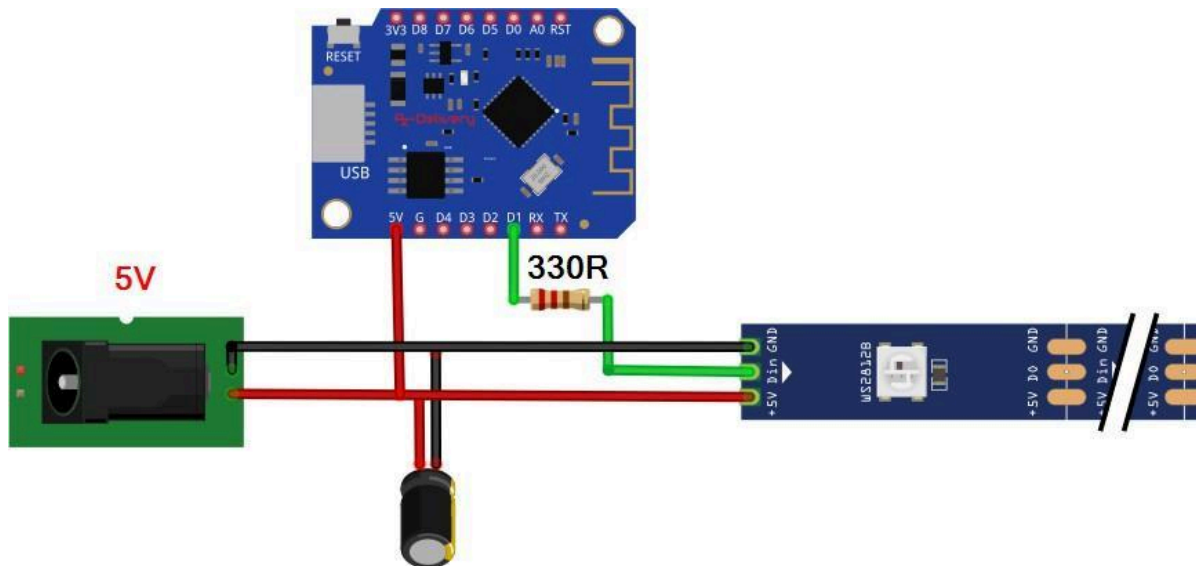
Flashen der WLED Firmware

Verbinden Sie das Board über das USB-Kabel mit Ihrem PC.

Starten Sie ESPHome flasher und wählen Sie den Port im Dropdown-Menü "Serieller Anschluss".

Klicken Sie auf "Durchsuchen" und wählen Sie die heruntergeladene WLED .bin-Datei aus und klicken Sie schließlich auf "Flash ESP".

Anschluss Diagramm



Nach dem Einschalten öffnet der esp einen WiFi-Zugangspunkt namens "WLED-AP", mit dem Sie sich mit Ihrem Smartphone verbinden können. Das Passwort lautet "wled1234".

Auf der Willkommensseite können Sie "WIFI EINSTELLUNGEN" wählen, um ihn in Ihr Heimnetzwerk zu integrieren, aber Sie können ihn auch im AP-Modus lassen und die SSID und das Passwort ändern.

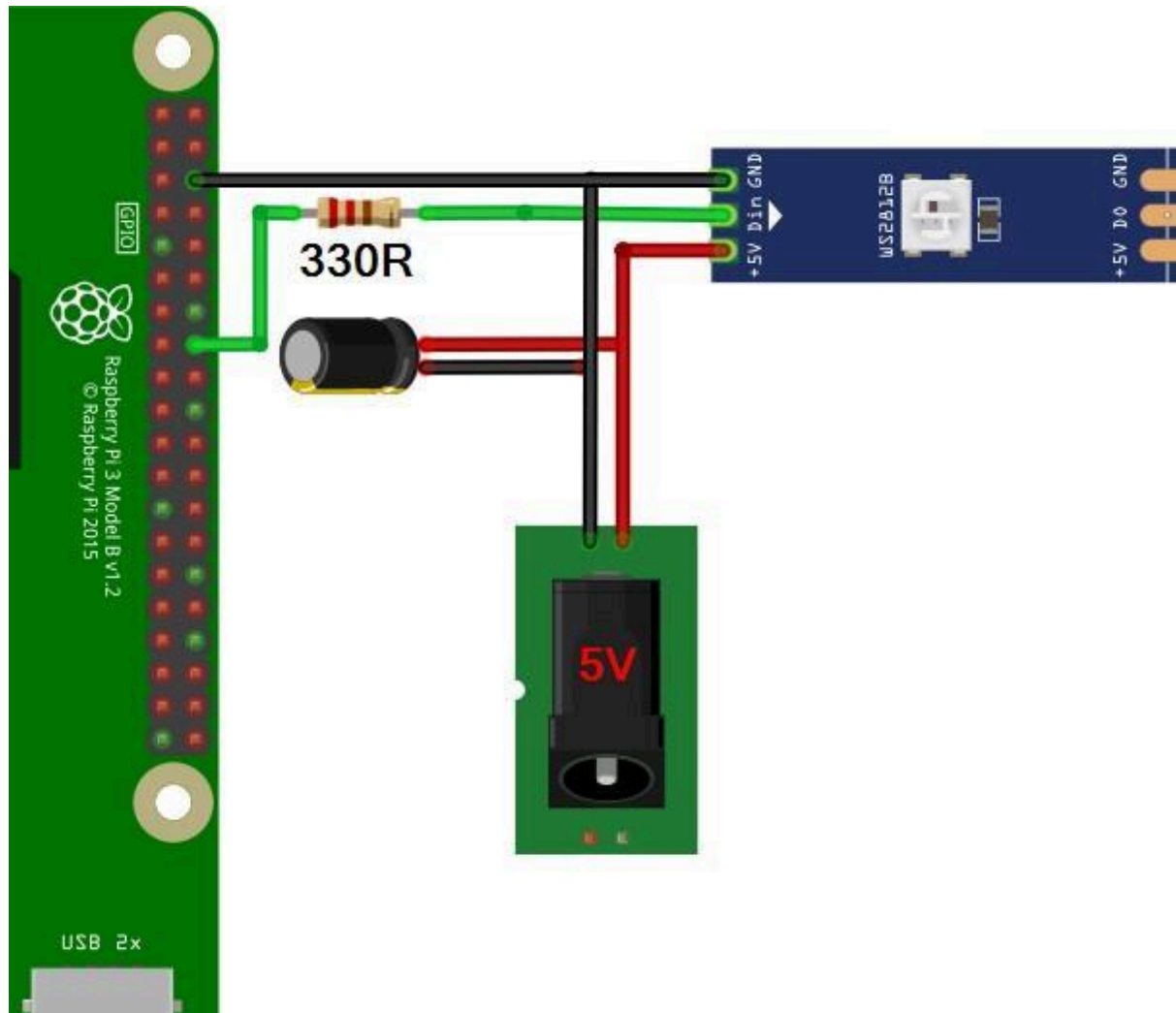
Um mit der Einrichtung fortzufahren, klicken Sie auf "ZU DEN KONTROLLEN" und gehen Sie zu "Konfig". Wählen Sie "LED-Einstellungen" und scrollen Sie nach unten zu "Hardware-Einstellungen". Hier können Sie die Länge Ihres Streifens (z.B. 30) und den verwendeten GPIO-Pin eingeben. Wir haben D1 auf dem D1 Mini verwendet, das ist GPIO5. Klicken Sie auf "Speichern", wenn Sie fertig sind. Jetzt leuchtet der Streifen gelb. Jetzt können Sie mit verschiedenen Farben, Segmentierungen und einer Menge von Effekten und Animationen spielen.

Verwendung der LED-Streifen mit einem Raspberry Pi

Der Raspberry Pi ist ein vielseitiger, kreditkartengroßer Computer, auf dem ein vollständiges Linux-Betriebssystem läuft. Mit seinen GPIO-Pins (General Purpose Input/Output) und den Möglichkeiten der Python-Programmierung können Sie komplizierte Lichteffekte erzeugen, LEDs mit Musik synchronisieren oder ihn sogar in größere Projekte wie Hausautomatisierungssysteme integrieren.

Wir empfehlen die Verwendung des offiziellen Raspberry Pi OS.

Anschluss Diagramm



Vermeiden Sie es, die LEDs über den 5V-Pin des Raspberry Pi zu betreiben:

Der Raspberry Pi kann den für den LED-Streifen erforderlichen Strom nicht liefern.

Verwenden Sie ein separates Netzteil für die LEDs: Stellen Sie sicher, dass es die gesamte Stromaufnahme bewältigen kann.

Schließen Sie zuerst die Erdung an: Schließen Sie immer zuerst alle Erdungskabel an, bevor Sie den Strom anschließen, um Schäden zu vermeiden.

Einrichten der virtuellen Python-Umgebung

Bei den neuesten Versionen des Raspberry Pi-Betriebssystems empfiehlt es sich, virtuelle Python-Umgebungen (venv) zu verwenden, um Abhängigkeiten zu verwalten und Konflikte zwischen System- und Projektpaketen zu vermeiden.

Öffnen Sie ein Terminal und führen Sie aus:

```
sudo apt update && sudo apt upgrade -y
```

```
sudo apt install python3 python3-venv python3-pip -y
```

Navigieren Sie zu Ihrem Home-Verzeichnis und erstellen Sie einen neuen Ordner:

```
cd ~  
mkdir led_strip_project  
cd led_strip_project
```

Erstellen Sie eine virtuelle Umgebung:

```
python3 -m venv venv
```

Aktivieren Sie die virtuelle Umgebung:

```
source venv/bin/activate
```

Installieren Sie die benötigten Bibliotheken

Upgrade pip:

```
pip install --upgrade pip
```

Installieren Sie die rpi_ws281x Bibliothek:

```
pip install rpi_ws281x adafruit-circuitpython-neopixel
```

Stellen Sie sicher, dass SPI und PWM aktiviert sind:

Führen Sie `sudo raspi-config`

aus Navigieren Sie zu

Schnittstellenoptionen

Aktivieren Sie SPI und PWM

Starten Sie den Raspberry Pi neu, wenn Sie dazu aufgefordert werden.

Beispiel Code

Nun, da alles eingerichtet ist, schreiben wir ein Python-Skript zur Steuerung der LED-Leiste.

Erstellen Sie in Ihrem Projektverzeichnis eine neue Datei namens "led_control.py":

```
nano led_control.py
```

Fügen Sie den folgenden Code ein:

```
import board
import neopixel
import time

# LED strip configuration:
LED_COUNT = 30          # Number of LED pixels.
LED_PIN = board.D23     # GPIO pin connected to the pixels
                        # (must support PWM).
LED_BRIGHTNESS = 0.5   # Float from 0.0 (min) to 1.0 (max)
LED_ORDER = neopixel.GRB # Order of LED colors
```

```
# Create the NeoPixel object
led_strip = neopixel.NeoPixel(
    LED_PIN, LED_COUNT, brightness=LED_BRIGHTNESS,
    auto_write=False, pixel_order=LED_ORDER
)

def colorWipe(color, wait_ms=50):
    """Wipe color across display a pixel at a time."""
    for i in range(LED_COUNT):
        led_strip[i] = color
        led_strip.show()
        time.sleep(wait_ms / 1000.0)

def theaterChase(color, wait_ms=50, iterations=10):
    """Movie theater light style chaser animation."""
    for j in range(iterations):
        for q in range(3):
            for i in range(0, LED_COUNT, 3):
                led_strip[i+q] = color
                led_strip.show()
                time.sleep(wait_ms / 1000.0)
            for i in range(0, LED_COUNT, 3):
                led_strip[i+q] = (0, 0, 0)

def rainbowCycle(wait_ms=20, iterations=5):
    """Draw rainbow that uniformly distributes itself
    across all pixels."""
    for j in range(255 * iterations):
        for i in range(LED_COUNT):
            pixel_index = (i * 256 // LED_COUNT) + j
            led_strip[i] = wheel(pixel_index & 255)
        led_strip.show()
        time.sleep(wait_ms / 1000.0)

def wheel(pos):
```

```
"""Generate rainbow colors across 0-255 positions."""
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))

if __name__ == '__main__':
    try:
        while True:
            colorWipe((255, 0, 0)) # Red wipe
            colorWipe((0, 255, 0)) # Green wipe
            colorWipe((0, 0, 255)) # Blue wipe
            theaterChase((127, 127, 127)) # White
    theater chase
        rainbowCycle()
    except KeyboardInterrupt:
        led_strip.fill((0, 0, 0))
        led_strip.show()
```

Speichern und Beenden.

Führen Sie das Skript mit aus:

```
sudo python3 led_control.py
```


Jetzt ist es an der Zeit, zu lernen und Ihre eigenen Projekte zu erstellen. Das können Sie mit Hilfe vieler Beispielskripte und anderer Anleitungen tun, die Sie im Internet finden können.

Wenn Sie auf der Suche nach hochwertigen Produkten für Arduino und Raspberry Pi sind, sind Sie bei der AZ-Delivery Vertriebs GmbH an der richtigen Adresse. Sie erhalten zahlreiche Anwendungsbeispiele, vollständige Installationsanleitungen, eBooks, Bibliotheken und Unterstützung durch unsere technischen Experten.

<https://az-delivery.de> Viel

Spaß!

Impressum

<https://az-delivery.de/pages/about-us>