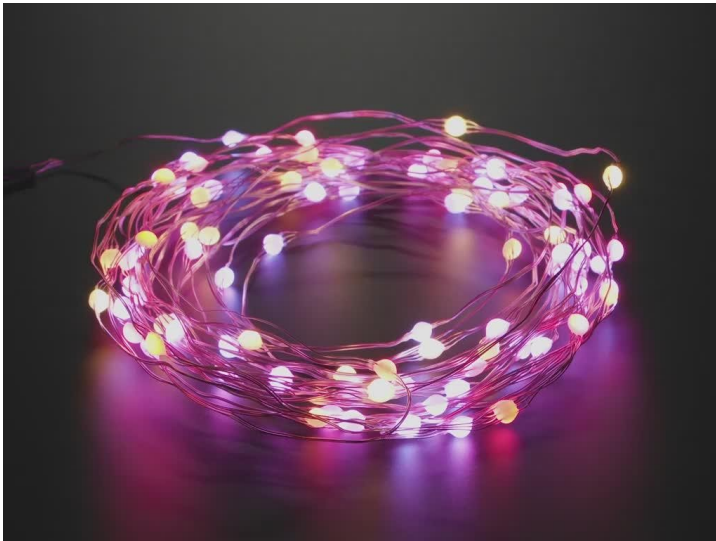




## Adafruit Fixed Address NeoPixel LED-Lichterkette - 100 verdrahtete LEDs



Artikel-Nr.:	ADA4917
Hersteller:	Adafruit
Herkunftsland:	China
Zolltarifnummer:	85423111
Gewicht:	0.049 kg

Diese sehen unseren NeoPixel-Lichterketten hier sehr ähnlich, haben aber einen funky Twist. Tatsächlich sind wir ein wenig zögerlich, sie NeoPixel zu nennen, denn **während sie durch "NeoPixel"-Code gesteuert werden können, funktionieren sie nicht wie NeoPixel**. Technisch gesehen ist ein NeoPixel eine LED, die auch als Schieberegister fungiert. Am Eingangspin kommen 24 oder 32 Bits an Helligkeitsdaten an, die LED entfernt diese und gibt den Rest weiter. Auf diese Weise kann man beliebig viele Pixel aneinanderreihen: jedes Pixel schnappt sich einfach die ersten Farbdaten und gibt den Rest wie eine Eimerkette weiter

Diese LEDs sind **nicht** Schieberegister. Tatsächlich gibt es, wenn man genau hinsieht, keinen Eingangs- und Ausgangspin, sondern nur eine Datenleitung. Das liegt daran, dass jede LED in diesem Strang von 0 bis 100 voradressiert ist. Immer wenn sie die NeoPixel-Daten empfängt, wählt sie die n-ten Farbdaten aus (passend zu ihrer Voradresse) und zeigt diese an. Sie können die Voradressierung nicht ändern, sie ist fest eingestellt - oder zumindest haben wir keine Ahnung, wie man sie umadressieren kann.

Sie können diese also mit jeder NeoPixel-Bibliothek verwenden, aber Sie können nicht zwei davon verketteten und erwarten, dass sich der Streifen verlängert. Auch wenn man diese in zwei Hälften schneidet und die Datenleitungen der beiden Hälften mit dem Datenpin des Mikrocontrollers verbindet, wird es sich immer noch so verhalten, als wäre es ein langer Strang.

Das ist zwar etwas anders als bei den meisten NeoPixeln, kann aber für einige Situationen nützlich sein - wenn man mehrere "Verzweigungen" von NeoPixeln haben möchte, ohne die Kette im Zickzack zu führen. Oder wenn Sie mehrere Stränge haben wollen, die dasselbe anzeigen.

Jede LED ist in einem milchig-weißen Epoxy-Blob eingeschlossen, wir haben versucht, eine aufzuschneiden, aber sie ist nahezu unzerstörbar. Der verwendete Draht ist Solid-Core Einzellitze, flexibel, aber hält seine Form, der Draht ist Magnet-Draht, mit einer nicht-leitenden Beschichtung so lange, wie die Beschichtung nicht abkratzen, können Sie die Drähte berühren haben. Es gibt einen beschichteten Pixel alle ~100mm (~4 Zoll). Es kommt nicht mit wetterfester Ummantelung, so dass wir nicht empfehlen, diese zu waschen oder tauchen sie unter Wasser. Sie können die Litze recht einfach mit einer Drahtschere schneiden. Achten Sie darauf, Abrieb oder wiederholtes Biegen der Anschlusskabel zu vermeiden.

Das Anschließen kann ein wenig knifflig sein, weil es keine Strom/Daten-Markierungen gibt (es gibt keine Platine!) Der rote Draht (der kupferigere rote Draht) ist **Strom**. Der mittlere Draht ist **Daten**. Der restliche Draht ist Masse. Kratzen oder brennen Sie die Beschichtung des Magnetdrahtes ab, um ihn mit einem Bügeleisen anzulöten. Natürlich gibt es bei diesen Pixeln keine Ein-/Ausgangsseite, da sie nur eine Datenleitung haben.

Die Litzen werden mit vormontierten Steckern geliefert, aber Sie können auch vorsichtig an die flexiblen Drähte löten. Sie können zu einem 3-poligen JST-Set greifen, um daraus eine schnelle Verbindung zu machen. Natürlich können Sie auch Stränge miteinander verbinden, um sie länger zu machen, achten Sie nur darauf, wie viel Strom Sie benötigen! Wir haben eine 5V/4A-Versorgung, die in der Lage sein sollte, viele Meter zu treiben (je nach Verwendung) **Sie müssen ein 5V DC-Netzteil verwenden, um diese Streifen zu**



**betreiben; verwenden Sie nicht mehr als 6V, oder Sie können den gesamten Streifen zerstören.**

Unser detaillierter [NeoPixel Überguide](#) hat alles, was Sie brauchen, um NeoPixels in jeder Form und Größe zu verwenden. Inklusive fertiger Bibliothek & Beispielcode für Arduino-kompatible Geräte. Sie können diese Streifen auch mit MakeCode oder CircuitPython verwenden, wenn Sie die NeoPixel-Softwareunterstützung nutzen.

## Weitere Bilder:

