

Das Mysterium der (adressierbaren) fairylight-LEDs

- [Worum geht's?](#)
- [FairyLight-LED-string](#)
- [Nun kommt das Mysterium](#)
- [Die Erklärung](#)
- [Vor- und Nachteile](#)
- [Workaround](#)



Worum geht's?

Normalerweise haben die WS2812-LEDs auf zB einem Streifen einen Data-In-Eingang und einen Data-Out-Ausgang.

Erfolgt nun die Programmierung dieser LED-Streifen, werden die gewünschten Farbwerte der einzelnen LEDs als Binärdaten auf der Datenleitung mit einer fest vorgegeben Frequenz eingespeist / "gemorst".

Die erste LED nimmt die ersten 24 eingehenden Bits und setzt ihre entsprechende Farbe. Alle weiteren eingehenden Daten werden auf dem Data-Out durchgereicht zur nächsten LED. So geht das weiter, bis auch die letzte LED auf dem Streifen ihren Farbwert hat.

Diese Daten werden relativ schnell gesendet, mit 400 kHz bzw. 800 kHz.

FairyLight-LED-string

Nun gibt es zur Zeit einen neuen hot-stuff - die adressierbaren fairylight-LED-Strings-Lights - in 5, 10 und 20 Metern mit 50, 100 bzw. 200 LEDs.

Die LEDs sind in einer Kunststoffblase eingegossen und dadurch wasserdicht. Die Verbindungen sind jeweils drei Drähte, also keine Litzen.

Protokolltechnisch sind sie kompatibel zu den WS2812b-LEDs, also zB programmierbar mit Arduino, ESP8266, ESP32, atTiny usw.

Nun kommt das Mysterium

Auf Twitter berichten Nutzer, dass sie den gesamten String wie gewohnt Led für Led ansteuern konnten. Dann trennten sie die Kette, indem sie einen Teil der LEDs abgeschnitten und in veränderter Reihenfolge wieder zusammengelötet haben - und jetzt kommt's - die LEDs wissen noch ihre ursprüngliche Reihenfolge und Position in der Kette. Hm... very spooky 😬

Auch bei unseren ersten Tests stellten wir folgendes fest: eine LED (LED 15) hatte nicht den ihr zugewiesenen Farbwert, sondern den Farbwert der ersten LED. Wie kann das sein? Theoretisch bei WS2812 bekommt nach der ersten LED keine der folgende LEDs mehr die Farbwerte der ersten LED. Wie kann also die 15. LED diesen Farbwert "erahnen"?

Die Erklärung

Diese LED-strings haben nicht, wie bei WS2812 üblich und oben beschrieben, jeweils einen Daten-Eingang und Daten-Ausgang pro LED, sondern haben nur **einen** Daten-PIN und dieser ist durchgeschliffen von der ersten bis zur letzten LED. Sozusagen sind alle LEDs parallel auf den 3 Leitungen angelötet und alle LEDs teilen sich **eine** gemeinsame Datenleitung. Vermutlich wird in der Produktion während des Löt-Vorgangs das jeweilige kleine LED-Platinchen mit einer festen Adresse beschrieben ("Du bist LED Nr. 15") dann verlötet und eingegossen.

Wenn die Kontaktierung / Adressierung nicht funktioniert hat, ist vermutlich die default-Adresse "0" eingestellt, also die der ersten LED auf dem string. So ließe sich erklären, warum Nr. 15 bei uns eine Nr. 0 ist.

Feste Adressierung der einzelnen LEDs? Sowas hatten wir bisher noch nie gesehen. 😊

Draufgekommen sind wir bei der Fehlersuche nach unserem LED-15-issue. Ein Test mit einem Durchgangsmesser hat uns bestätigt, dass die Datenleitung von vorne bis hinten EIN Stück ist.

Vor- und Nachteile

Vorteil ist, sollte einmal eine LED ausfallen, leuchten die restlichen LEDs trotzdem wie vorgesehen weiter. Ist natürlich perfekt für schlecht-zugängliche Installationen, zB im Außenbereich an Gebäuden usw..

Nachteil ist, man kann solche strings nicht einfach durch Austauschen einzelner LEDs reparieren, da man die Adresse ja nicht selber schreiben / umstellen kann. Die Daten auf dem Daten-Bus werden nicht "signaltechnisch aufgefrischt", sondern je weiter die LED vom Sender entfernt ist, umso "schlechter" wird das Signal. Die WS2812b-LEDs hingegen machen ein internes signal-shaping und frisken somit das Signal zwischen den LEDs immer wieder auf. Reststücke können natürlich auch nicht so einfach wiederverwertet, also zusammengestückelt werden, da man die Adressen weder auslesen noch ändern kann (bisheriger Kenntnisstand 😊)

Workaround

Um unseren LED-15-issue zu lösen, werden wir die Software um ein "ausmapp-array" erweitern, in das man zu überspringende LED-Nummern konfiguriert. Dann kann man die LED-15 physikalisch rauswickeln und trotzdem bleibt die Reihenfolge des strings konsistent.

Error rendering macro 'widget'

```
java.lang.IllegalArgumentException: The provided url- https://publish.twitter.com/oembed?url=https%3A%2F%2Ftwitter.com%2FFabLabMuc%2Fstatus%2F1354144732100882433%3Fs%3D20&maxwidth=400&lang=en is not included in the whitelist!.Please add the url to whitelist to allow access
```

zB bei:

<https://de.aliexpress.com/item/1005001670784973.html>