



# Wohlfühl-Licht

## Tannenzapfen-Lampe mit RGB-Leuchte



Der dänisch/norwegische Begriff „Hygge“ ist nicht zuletzt wegen der Covid-19-Pandemie ein fester Bestandteil von Lifestyle-Magazinen und Einrichtungsberatungen geworden. Hygge bedeutet „Behaglichkeit“, und ein ganz wesentlicher Teil dieser Philosophie ist es, eine gemütliche und herzliche Atmosphäre zu schaffen, in der Beleuchtung und Kerzen eine wesentliche Rolle spielen. Ziel dieses Leserwettbewerb-Projekts ist es, eine stylische Stehlampe in Form eines Tannenzapfens mit einer selbst-gemachten LED-Leuchte zu versehen, die verschiedenste Lichtstimmungen erzeugen kann.

**Vielen Dank!**



*\*siehe Seite 112*

**Dr. Peter Tschulik**

hat für seinen Beitrag zum Leserwettbewerb einen Gutscheincode\* über 200,- Euro erhalten!

### Zwei Varianten

Zum Einsatz kommen hier leicht erhältliche WS2812-Neopixel-Streifen. Das Projekt ist eine spannende Verbindung von Hard- und Software, Webdesign, 3D-Druck und Programmierung und kann in zwei Varianten umgesetzt werden, die im Folgenden beschrieben werden:

**Variante 1:** Steuerung durch den ELV Bausatz Lichtorgel LED-WS2812 [1]

**Variante 2:** Steuerung durch einen eigenen ESP32 [2] mit umfangreichen und komfortablen Konfigurationsmöglichkeiten über einen eigenen WLAN Access Point

Am Anfang steht der 3D-Druck: Zuerst einmal drucke ich die vier Teile der Tannenzapfen-Lampe nach dem Download von Thingiverse [3] mit der Bezeichnung

PineCone\_Lamp-cap  
PineCone\_Lamp-base  
PineCone\_Lamp-middle  
PineCone\_Lamp-top

aus, was mehrere Stunden dauert. Als Filament verwende ich PLA von Renkforce in der Farbe „Natural“, mit dem ich gute Erfahrungen ge-

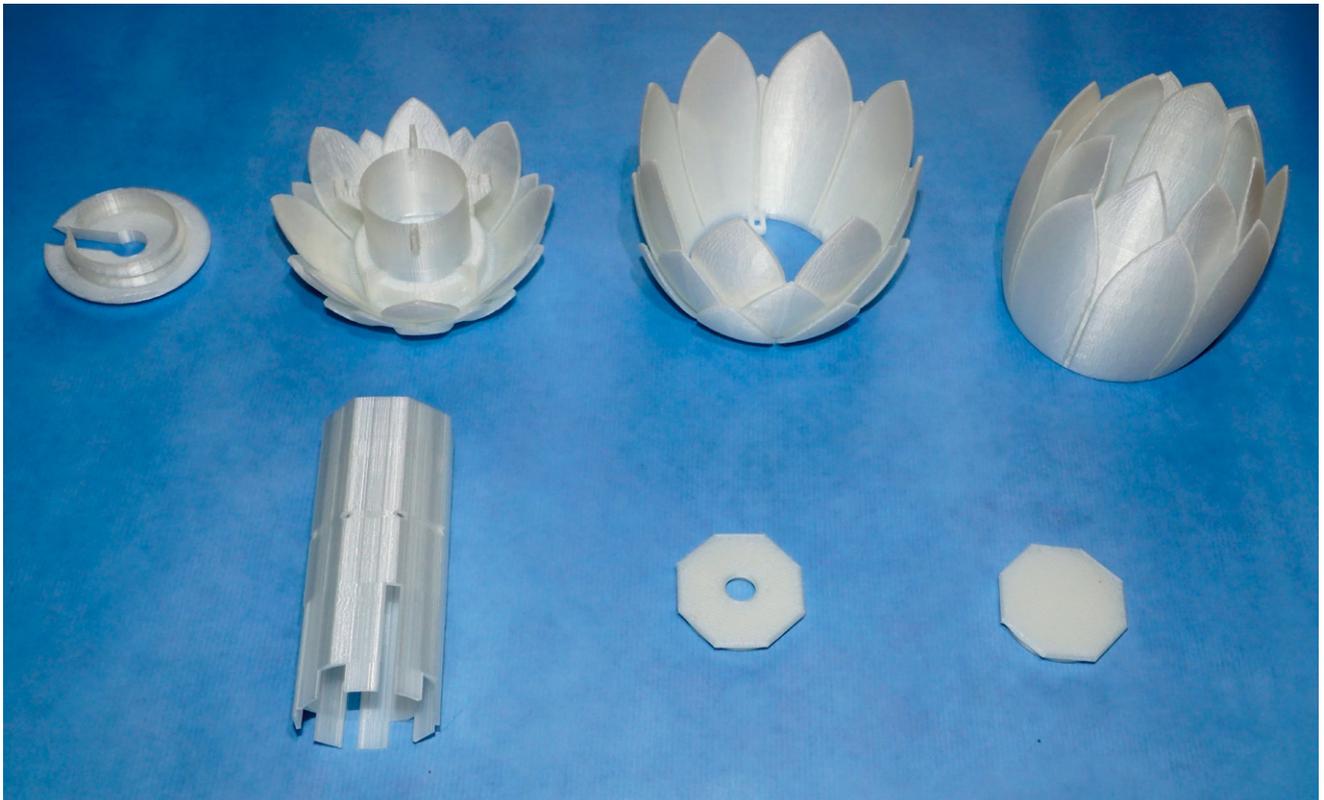


Bild 1: Die sieben per 3D-Druck erstellten Teile für die Leuchte

macht habe. Als Druckparameter wähle ich einen Infill von 50 %, Supports und ein Raft zur besseren Haftung auf dem Druckbett.

Die RGB-Leuchte habe ich mit dem Open-Source-Programm OpenSCAD entwickelt [4, 5], mit dem auch komplexe Formen schnell und einfach entworfen werden können. Die Originaldateien können unter [6] heruntergeladen und nach Bedarf angepasst werden.

Der Hauptteil trägt den Namen „Neopixel bulb Main“ und wird mit den Seitenschlitzen nach oben mit einem Infill von 60 %, Supports und einem Raft ausgedruckt. Mit dem gleichen Parameter drucke ich wahlweise „Neopixel bulb Top 1“ für Variante 1 für die Steuerung mittels der ELV Lichtorgel oder „Neopixel bulb Top 2“ für Variante 2 für die Steuerung durch den ESP32 aus. Letzterer verfügt über eine Bohrung für einen Ein-/Aus-Taster. Bild 1 zeigt alle sieben gedruckten Teile.

Damit ist der zeitintensivste Teil abgeschlossen!

### Zusammenbau der RGB-Leuchte

Für die RGB-Leuchte schneide ich acht Stücke zu je sieben LEDs von einem WS2812-Strip mit 60 LEDs pro Meter ab. Als Nächstes konfektioniere ich acht rote und acht schwarze Kabel aus 0,14-mm<sup>2</sup>-Schaltlitze mit einer Länge von ca. 9 cm. Die eine Seite wird auf 2–3 mm abisoliert und verzinnt, die andere Seite auf 8–10 mm abisoliert und nicht verzinnt.

Weitere sieben Kabel einer anderen Farbe (hier gelb) mit einer Länge von 3 cm werden zugeschnitten, auf beiden Seiten auf einer Länge von 3–4 mm abisoliert und verzinnt. Ein letztes Kabel der gleichen Farbe mit einer Länge von 19 cm wird zugeschnitten, auf beiden Seiten auf einer Länge von 2–3 mm abisoliert und verzinnt. Bild 2 zeigt die acht WS2812-Streifen und jeweils eines der unterschiedlichen Kabel.

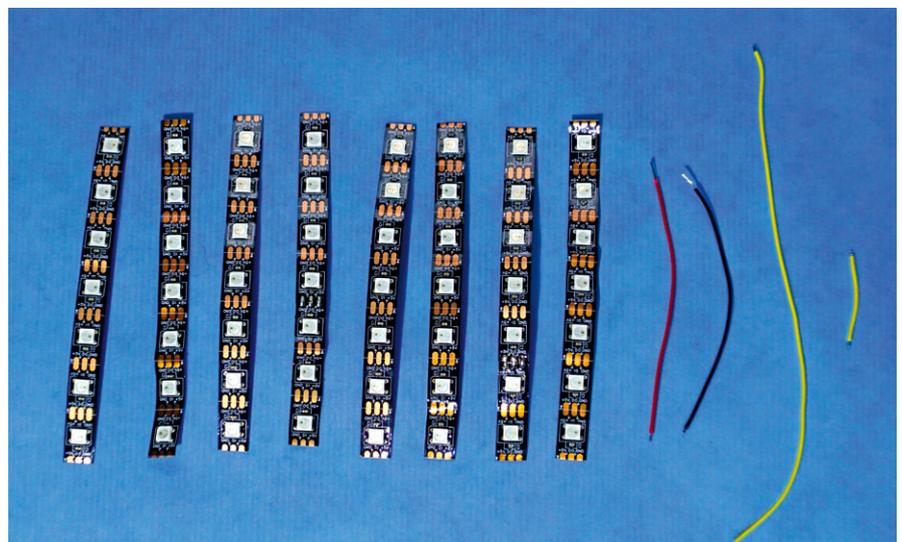


Bild 2: Die vorkonfektionierten LED-Streifen und Kabel

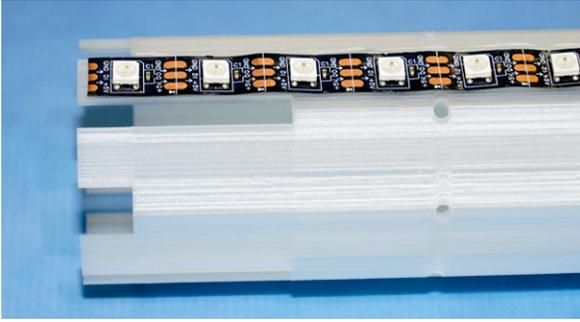


Bild 3: Der erste Streifen ist verklebt.

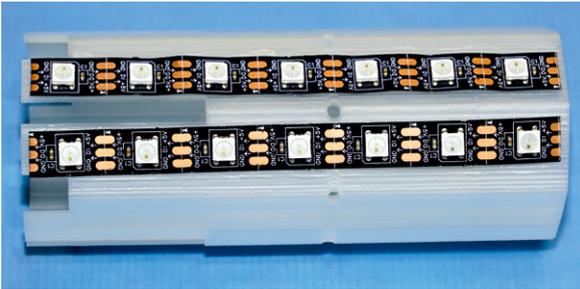


Bild 4: Position des zweiten Streifens

Ab nun ist Konzentration gefragt. Der erste Streifen wird genau wie im [Bild 3](#) gezeigt zwischen einem länglichen und einem kurzen Schlitz montiert – mit dem DI-Pin in Richtung der Schlitz (Unterseite). Da das Doppelklebeband auf dem PLA-Material nicht nachhaltig haftet, verwende ich zusätzlich einen transparenten Kraftkleber.

Beim nächsten Streifen, der von der geschlitzten Seite aus im Uhrzeigersinn angebracht wird, muss darauf geachtet werden, dass der DI-Eingang des nächsten Streifens neben dem DO-Ausgang des ersten Streifens zu liegen kommt (Mäanderform) ([Bild 4](#)).

Im Anschluss werden nach dem gleichen Prinzip die restlichen Streifen verklebt.

Danach wird der LötKolben angeheizt, um mit der Verkabelung zu beginnen: Ausgehend vom ersten Streifen wird der DO-Ausgang mit den kurzen Kabelstücken mit dem DI-Eingang des zweiten Streifens verbunden. Dies wiederholt sich bis zum letzten Streifen, wo der DO-Ausgang unbeschaltet bleibt. Dabei ist genau darauf zu achten, keine Kurzschlüsse durch Lötbrücken zu produzieren, die einzelne LEDs später zerstören könnten.

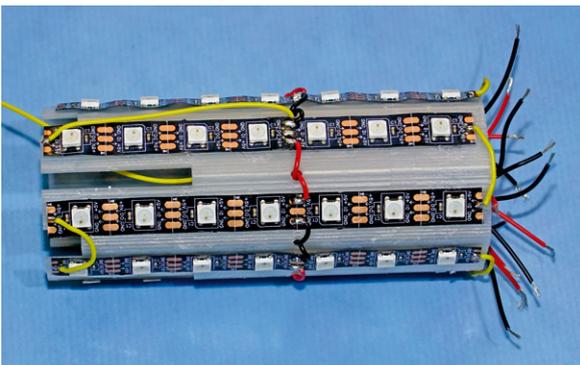


Bild 5: Verkabelung von DI/DO und Spannungsversorgung an den Streifen

Anschließend werden je nach der Polarität abwechselnd zwei rote und zwei schwarze Drähte durch die mittleren Bohrungen geführt und auf die Oberseite der Lampe (jene ohne die Schlitz) gezogen und danach mit den korrekten Versorgungspins neben den Löchern an den Strips verlötet. Das verbleibende lange Kabel wird an den DI-Pin des ersten Strips gelötet und durch eine Mittelbohrung nach unten gefädelt. Dabei ist wieder darauf zu achten, keine Kurzschlüsse durch Lötbrücken zu produzieren. [Bild 5](#) zeigt den abgeschlossenen Schritt.

Nun werden jeweils vier rote und vier schwarze Adern miteinander verdreht und verlötet ([Bild 6](#)). Falls Variante 2 mit dem ESP32 aufgebaut werden soll, ist darauf zu achten, dass die jeweils zusammengefassten Drähte sich nicht mit den Führungsschienen im Inneren der RGB-Leuchte überlappen. Kabelbinder sorgen für zusätzliche Ordnung.

Zwei rote und zwei schwarze, ca. 14 cm lange Kabel werden an die vier Enden angelötet und danach die Lötstellen mit einem Schrumpfschlauch isoliert. Schlussendlich werden die vier Kabel durch den Lampenkörper nach unten durchgefädelt und alle Nahtstellen im Hohlraum der Lampe verstaut.

Nun beginnt der Zusammenbau der Lampe. Zuerst einmal wird der Mittelteil auf das Unterteil aufgesetzt und verklebt. Danach wird die RGB-Leuchte auf den unteren Teil aufgesetzt, und die Kabel werden durch den Hohlraum geführt.

Achtung: Die RGB-Leuchte erst nach dem Funktionstest verkleben, Oberteil noch nicht aufsetzen und unteren Schraubteil noch nicht montieren!

#### Fertigstellung für Variante 1 – Steuerung durch den ELV Bausatz Lichtorgel LED-WS2812

Bei dieser Variante gilt es nur mehr, den Anschluss zur Lichtorgel herzustellen. Ich verwende dazu den Originalsteckverbinder der LED-Streifen, der verpolungssicher ausgeführt ist. Die beiden roten Kabel werden untereinander verbunden, ebenso die beiden weißen Kabel mit den schwarzen und das mittlere, grüne Kabel mit dem DI-Eingang. [Bild 7](#) zeigt die endgültige Verdrahtung.

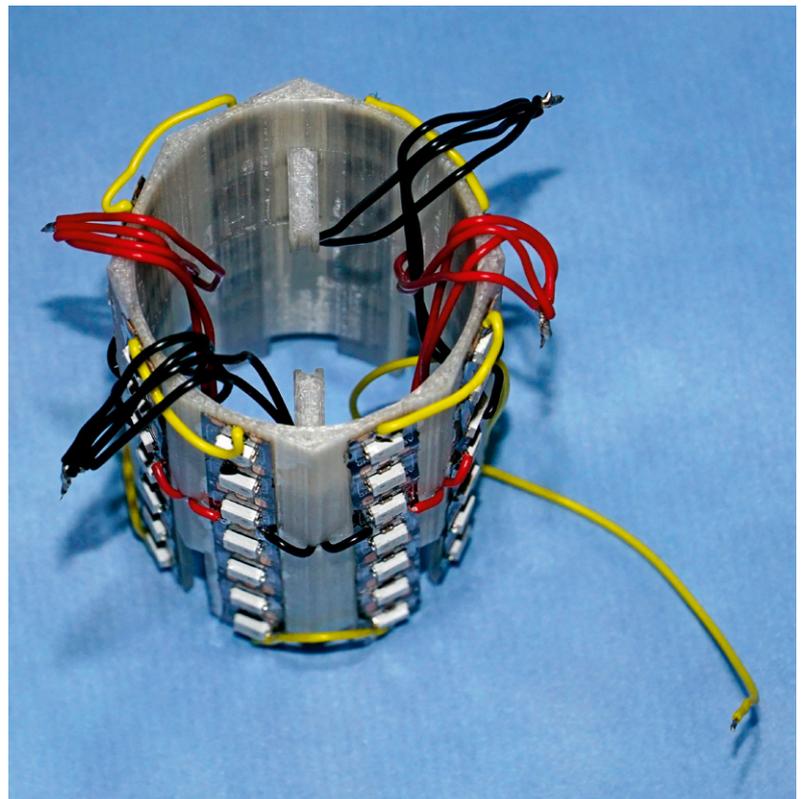


Bild 6: Fertigstellen der Verkabelung im Inneren

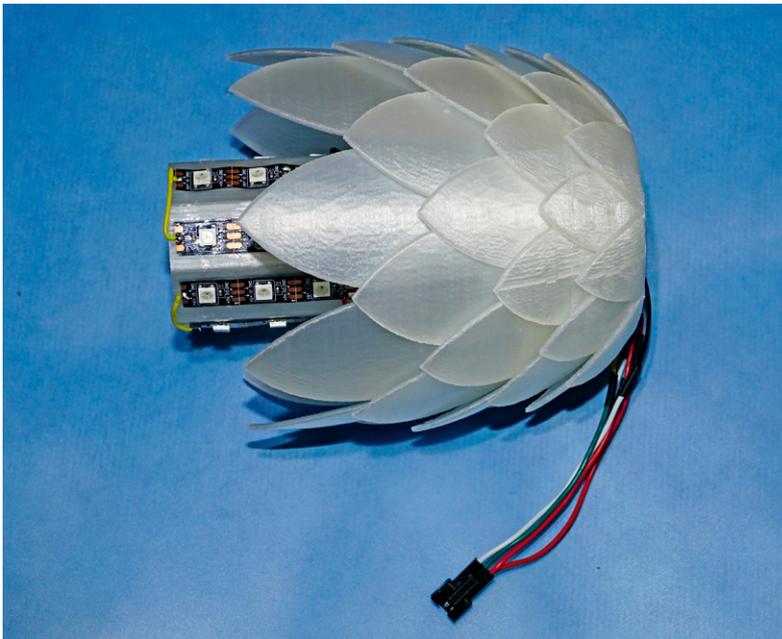


Bild 7: Verdrahtung für Variante 1

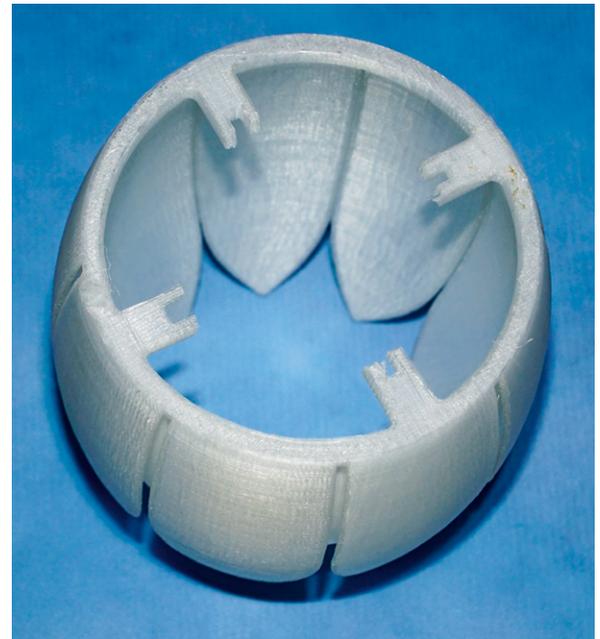


Bild 8: Vorbearbeitung des Oberteils

Nach einem Funktionstest werden die Kabel im Lampenkörper verstaut, der Schraubteil wird befestigt und der Stecker kann in der Aussparung an der Unterseite mit Heißkleber befestigt werden. Der Gegenstecker wird wie in der Anleitung zum ELV Bausatz Lichtorgel LED-WS2812 [1] beschrieben angeschlossen. Danach wird die Lichtorgel mit Strom versorgt. Der maximale Strom der RGB-Leuchte beträgt  $56 \times 60 \text{ mA} = 3,36 \text{ A}$ .

Wie in der Anleitung beschrieben wird die Taste „Ein/Aus“ für mindestens 3 s gedrückt, damit man in das Set-up-Menü gelangt. Im Menüpunkt 1 wird die Anzahl der angeschlossenen LEDs mit der Hinauf- und Hinuntertaste auf 56 eingestellt. Im Menüpunkt 2 wird die Option 3 „90° gedreht + Mäander“ ausgewählt, im Menüpunkt 3 wird für die Anzahl der Spalten „8“ eingetragen.

Funktioniert alles, wird die Kappe (Variante ohne Loch) auf die Oberseite der RGB-Leuchte aufgesetzt und gegebenenfalls verklebt. Ebenso wird die RGB-Leuchte mit dem Unterteil verklebt.

Das Oberteil der Lampe muss noch etwas modifiziert werden: Die vier Schnappnasen müssen wie in Bild 8 gezeigt mit einer Minirennscheibe oder einem scharfen Messer gekürzt werden. Dann wird auch das Oberteil vorsichtig über die RGB-Leuchte geschoben und verklebt.

### Fertigstellung Variante 2 – Steuerung durch einen ESP32

Ich beginne mit der Montage des Ein-/Aus-Tasters [7]. Dieser wird mittels der beiliegenden Mutter verschraubt, und jeweils ein schwarzer und ein z. B. grüner Draht mit ca. 14 cm Länge werden angelötet und isoliert (Bild 9).

Danach folgt die Verdrahtung des ESP32-Pico-Kits, die von der Unterseite der Lampe aus erfolgt.

Der grüne Draht für den Taster wird auf IO15 (entsprechend GPIO13) angelötet. Der schwarze Draht kommt an einen GND-Pin des ESP32-Pico-Kits. Der DI-Pin wird über einen 150-Ω-Widerstand an IO2 (entsprechend GPIO 12) angelötet. Der 150-Ω-Widerstand dient als Schutz, wenn nur das ESP32-Pico-Kit ohne die WS2812-LEDs versorgt wird.

Die Anschlussbeine werden ein wenig nach innen gebogen, um den ESP32 später leichter ins Gehäuse zu schieben.

Für den Anschluss des ESP32-Pico-Kits schneide ich ca. 10 cm von einem passenden USB-Kabel ab. Die Versorgungsadern löte ich gemeinsam mit den zwei roten und zwei schwarzen Versorgungsleitungen der WS2812-LEDs an einen USB-Anschlussadapter an. Dies dient dazu, das ESP32-Pico-Kit ohne angeschlossene Last der LEDs (> 3 A) vom PC aus programmieren zu können.

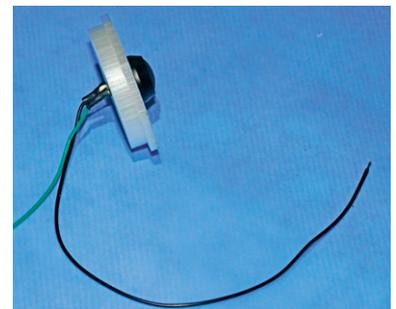


Bild 9: Vormontage des Ein-/Aus-Tasters

Bild 10 zeigt die Verdrahtung vor dem Einbau, wobei der Schumpfschlauch noch nicht über den Widerstand gezogen ist. Da mich die rote Power-Led am ESP32 stört, die immer leuchtet, sobald die Schaltung mit Strom versorgt wird, habe ich diese vor dem Einbau ausgelötet.

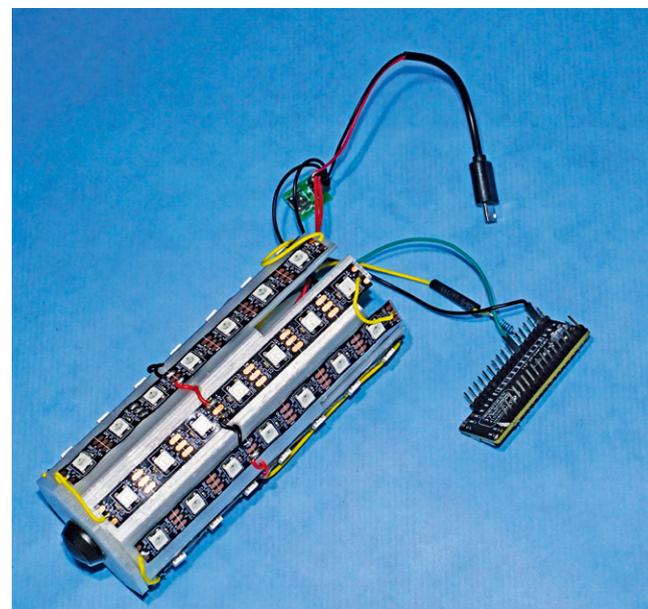


Bild 10: Verdrahtung vor dem Einbau

## Software

Vor dem Einbau der Komponenten erfolgt noch das Aufspielen der Software [6]. Diese wurde in der Arduino IDE erstellt. Eine genaue Beschreibung der umfangreichen Software würde diesen Artikel sprengen, allerdings ist die Software ausführlich dokumentiert und in folgende Blöcke geteilt: „Tannenzapfenlampe.ino“ enthält neben den Definitionen die Setup- und Main-Routine. „Basis.ino“ enthält Basisroutinen wie beispielsweise das Ein- und Ausschalten über den Taster, die Erstinitialisierung des Flash-Speichers, das Einlesen des Init.txt-Files, die Anzeige des Flash-Inhalts im Debugging-Modus sowie die Neopixel-Routinen, die in einem eigenen Task auf einem Core laufen. „Led.ino“ enthält die einzelnen LED-Programme wie beispielsweise statische Farben, Programme, das Auslesen und Beschreiben von fünf Speichern sowie eine Testroutine, die weiter unten bei der Bedienung beschrieben ist. „Web.ino“ schließlich beinhaltet die Behandlung und den Aufruf der einzelnen Webseiten, die über die Bibliothek „ESP Async Web Server“ erfolgt.

Folgende Vorbereitungsarbeiten sind vor dem Übertragen der Software notwendig:

- Drei Libraries „ESP Async Web Server“, „Async TCP“ und „Fastled“ müssen installiert werden. Die Links zu den Libraries finden sich in Zeile 14–16 von „Tannenzapfenlampe.ino“.
- In Zeile 205 kann der Befehl „#define \_DEBUG\_“ durch Entfernung der beiden // am Beginn der Zeile aktiviert werden. Dann werden viele zusätzliche Informationen über den Serial-Monitor der Arduino IDE mit 115.200 Baud ausgegeben. Das ist sehr hilfreich, wenn etwas nicht funktioniert oder der Code erweitert bzw. geändert werden soll.



Bild 11: USB-Anschluss an der Unterseite

Bevor jedoch der Code auf den ESP32 übertragen wird, müssen noch die Daten in Unterverzeichnis „data“ angepasst und übertragen werden. Die Datei „Init.txt“ enthält die vier Konfigurationsparameter. In der ersten Zeile (Default: NEOPIXEL\_BULB“) wird der Namen des WLAN-Netztes angegeben, das von der Tannenzapfen-Lampe aufgespannt wird. Die zweite Zeile gibt das Passwort an (Default: Default), das eingegeben werden muss, um sich mit dem WLAN der Tannenzapfen-Lampe zu verbinden. In der dritten Zeile wird die IP-Adresse (Default: 192.168.122.1) angegeben, unter der die Tannenzapfen-Lampe erreichbar ist. In der vierten und letzten Zeile wird die Subnet-Maske (Default: 255.255.255.0) angegeben. Die einzelnen Zeilen sind mit einem CRLF (Enter) geteilt, nach der letzten Zeile darf kein Zeilenumbruch mehr stehen! Im Unterverzeichnis „data“ sind die Styledatei „Style.css“ und die fünf Webseiten abgelegt. Bei Bedarf können auch diese geändert werden.

## Übertragung der Daten und des Programms

Nun geht es endlich darum, die Daten und das Programm zu übertragen. Das ESP32-Pico-Kit wird mit dem PC direkt (also ohne Versorgung der WS2812-LEDs) verbunden. In der Arduino IDE muss sowohl die ESP32-Erweiterung als auch der ESP32-Sketch-Data-Uploader (siehe Link in Tannenzapfen.ino Zeile 20–22) installiert sein.

Als Board wird das ESP-Pico-Kit ausgewählt, danach die korrekte Schnittstelle. Zuerst werden die Daten unter Tools → ESP32 Sketch Data Upload übertragen und dann der Sketch mittels Sketch → Upload. Hat das funktioniert, so geht es an einen ersten Funktionstest.

Der ESP32 wird über das USB-Kabel an den USB-Kabeladapter angeschlossen und die gesamte Schaltung mit einem genügend starken USB-5-V-Schaltenteil versorgt. Alle LEDs sollten kurz weiß aufleuchten und dann dunkel werden. Mit dem Taster wird die Lampe ein- und ausgeschaltet.

Hat der Schalttest funktioniert, wird die Schaltung in der RGB-Leuchte verstaut: Das ESP32-Pico-Kit wird in die Halterung in der Birne eingeschoben und eventuell verklebt, wobei das Abnehmen des Tasteroberteils beim Einsetzen hilfreich ist. Eine lange Spitzzange leistet gute Dienste. Den USB-Kabeladapter habe ich in die halbkreisförmige Aussparung an der Unterseite mit Heißkleber befestigt. Nachdem alle Kabel verstaut sind, wird das USB-Kabel an den ESP32 angeschlossen. Filzgleiter an der Unterseite verhindern Kratzspuren an Möbelstücken. Bild 11 zeigt die fertig montierte Tannenzapfen-Lampe.

## Steuerung der Lampe

Kommen wir zur Beschreibung der umfangreichen Steuerungsfunktionen über eine Web-Oberfläche. Um diese zu nutzen, muss man sich nach dem Einschalten der Lampe zuerst einmal mit dem WLAN (Default NEOPIXEL\_BULB) verbinden und das Passwort (Default: Default) eingeben. Bei manchen mobilen Endgeräten wie Smartphones muss man die mobile Datenverbindung abschalten, um auf die Lampe zugreifen zu kön-

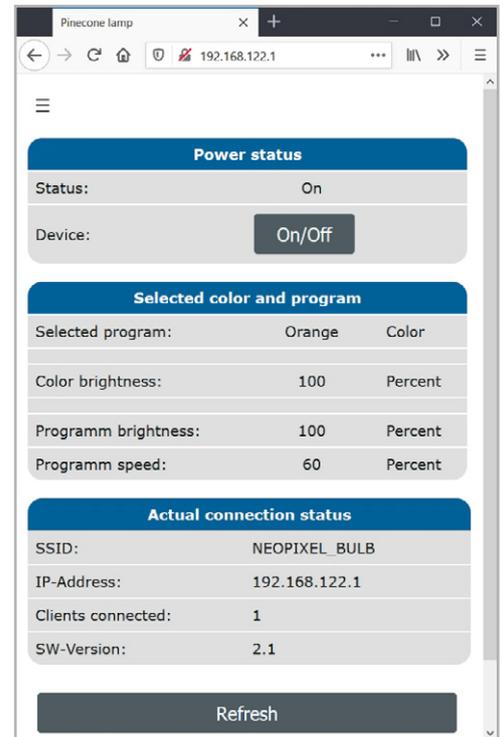
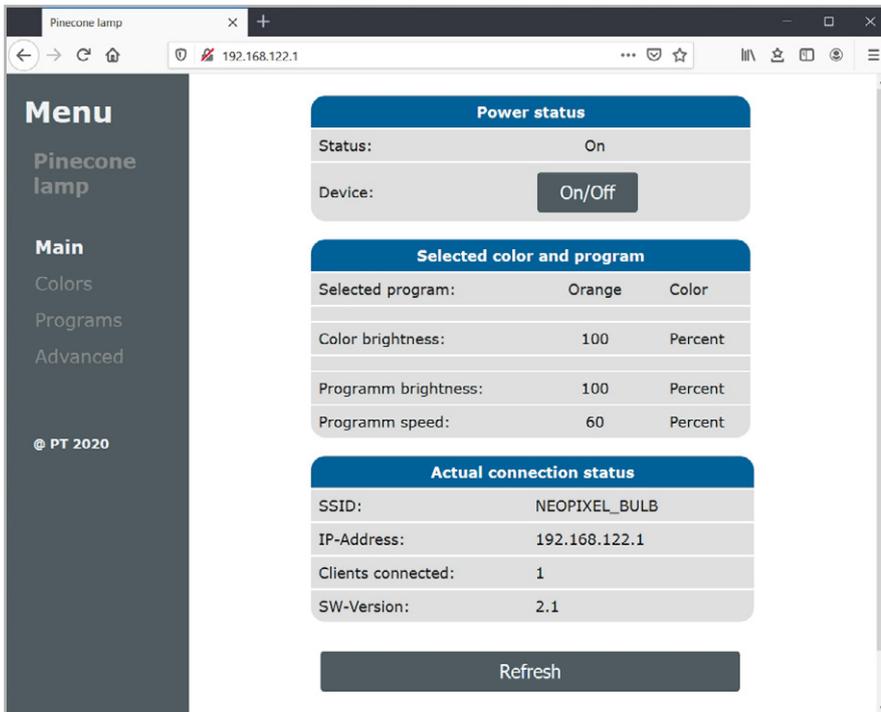


Bild 12: Darstellung der Hauptseite – links auf einem Tablet/Laptop, rechts auf einem Mobiltelefon

nen. In einem Browser gibt man die IP-Adresse der Lampe ein (Default: 192.168.122.1), und schon sollte die Hauptseite der Lampe angezeigt werden. Links im Bild 12 die Variante für größere Geräte wie Laptops und rechts die Variante für kleinere Geräte wie Mobiltelefone.

In Bild 12 sieht man das Hauptmenü, das entweder permanent angezeigt wird (Tablet/Laptop) oder sich ausklappen lässt (Mobiltelefon). Im Menü „Colors“ können festgelegte Farben eingestellt werden, im Menü „Programs“ Animationen und im Menü „Advanced“ können fünf Speicherplätze abgerufen oder programmiert werden bzw. beliebige Farben eingestellt werden.

Doch zurück zum Menü „Main“. Ganz oben wird der aktuelle Einschaltstatus angezeigt, und mit der Taste darunter kann die Lampe ein- bzw. ausgeschaltet werden. Im nächsten Abschnitt werden die aktuell eingestellten Werte angezeigt, in diesem Fall ist die Farbe Orange mit voller Helligkeit ausgewählt. Für die Programme ist auch eine volle Helligkeit gespeichert sowie eine Ablaufgeschwindigkeit von 60 Prozent. Der untere Abschnitt zeigt die aktuellen Verbindungsparameter an, den WLAN-Namen, die IP-Adresse, die Anzahl der verbundenen Clients sowie die SW-Version. Wichtig ist zu wissen, dass Parameter im Flash jeweils beim

Ausschalten der Lampe permanent gespeichert werden und daher beim neuerlichen Einschalten die letzte Einstellung wiederhergestellt wird.

Gehen wir zum Menü „Colors“ (Bild 13). Im Drop-down-Menü „Color“ können die Farben Kaltweiß, Warmweiß, Gelb, Orange, Rot, Dunkelviolet, Violett, Hellviolett, Dunkelblau, Blau, Cyan, Hellgrün und Grün ausgewählt werden. Im Drop-down-Menü „Color brightness“ kann die Helligkeit zwischen 10 % und 100 % in einigen Schritten eingestellt werden. Der Button „Save“ wendet die neue Einstellung an.

Noch Interessanteres findet sich im Menüpunkt „Programs“ (Bild 14). In diesem können fünf verschiedene Farbwechselprogramme, ein Programm namens „Steps“ mit an- und abschwellenden Stufen, zwei verschiedene Programme mit Blinkeffekten, eine Kamin-simulation sowie ein Unwetterprogramm ausgewählt werden. Im Drop-down-Menü „Programm brightness“ kann die Helligkeit in vier Stufen gedimmt werden.

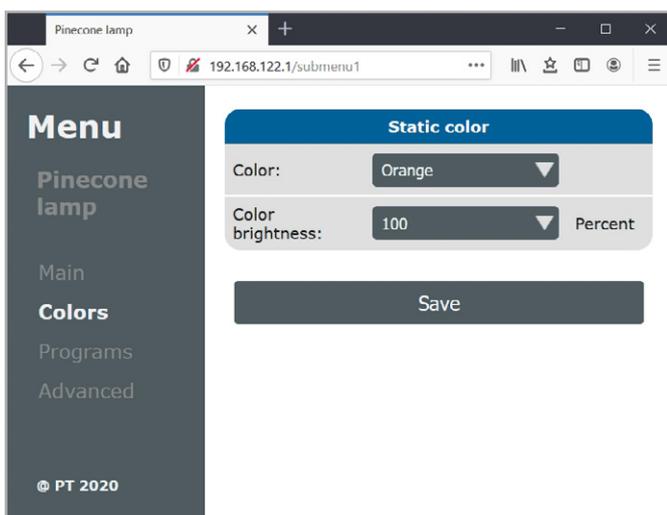


Bild 13: Einstellung der Farben

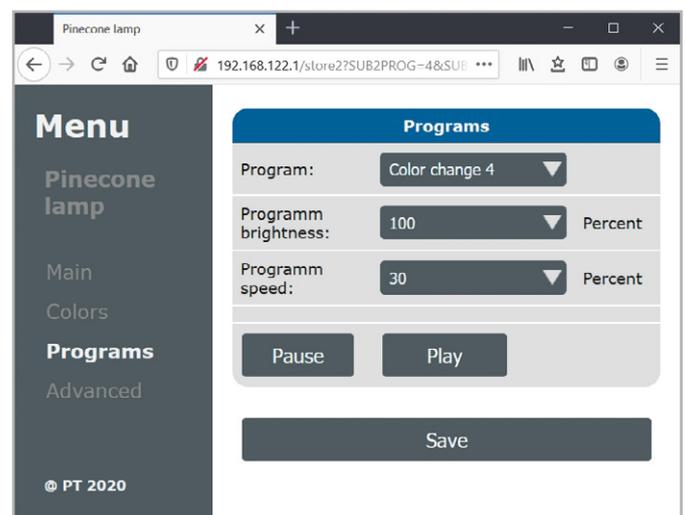


Bild 14: Einstellung von Farbwechseln

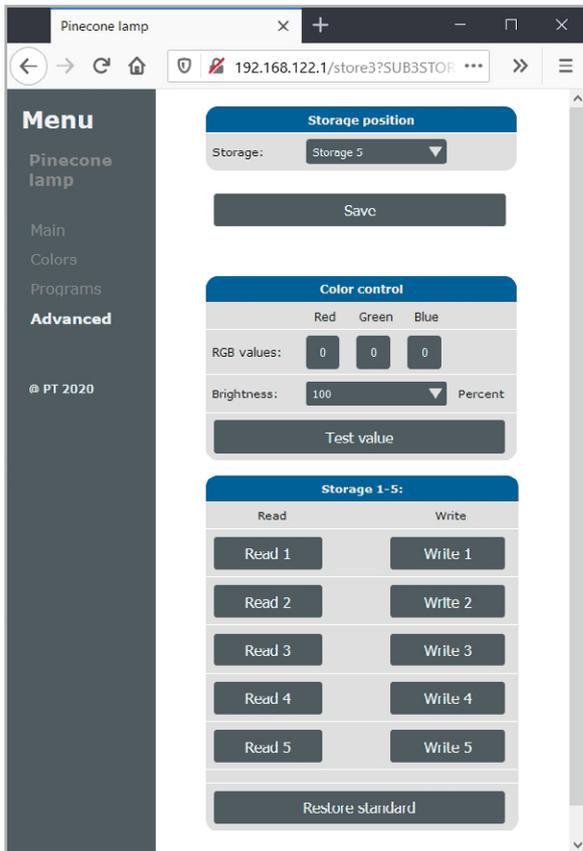


Bild 15: Maximale Flexibilität im Menü „Advanced“

Im Drop-down-Menü „Program speed“ kann die Geschwindigkeit des Programms in 10 Stufen eingestellt werden. Der Button „Pause“ hält das Programm an der aktuellen Position an, „Play“ setzt es an der aktuellen Position fort. Der Button „Save“ wendet die neue Einstellung an.

Die maximale Flexibilität bietet der letzte Menüpunkt „Advanced“ (Bild 15).

Im Abschnitt „Storage position“ kann eine von fünf Speicherpositionen ausgewählt und mit dem Button „Save“ angewendet werden.

Im Abschnitt „Color control“ kann jede beliebige Farbe erzeugt und gedimmt und mit dem Druck auf die Taste „Test value“ ausprobiert werden. Dieser Modus funktioniert nicht, solange ein Programm ausgeführt wird.

Im Abschnitt „Storage 1–5“ können fünf Speicher mit „Read 1“ bis „Read 5“ ausgelesen und im Abschnitt „Color control“ angezeigt werden. Mit „Write 1“ bis „Write 5“ werden die Einstellungen, die gerade in „Color control“ angezeigt werden, in den jeweiligen Speicher übernommen.

- Beispiel 1:

Speicherposition 2 mit neuer Farbe belegen: Zuerst unter „Color control“ die Farbanteile Rot, Grün und Blau auswählen, ebenso die Helligkeit und mit „Test value“ testen. Wenn die Farbauswahl gefällt, kann man die Schaltfläche „Write 2“ betätigen.

- Beispiel 2:

Speicherposition 3 ändern: Zuerst mit „Read 3“ den Speicherplatz abrufen und anzeigen, dann unter „Color control“ die Farben bzw. Helligkeit ändern und dann mit „Write 3“ abspeichern.

Die Schaltfläche „Restore standard“ setzt alle fünf Speicherplätze auf die im Programm hinterlegten Standardwerte zurück. Eine entsprechende Sicherheitsabfrage ist eingebaut.

Viel Spaß bei diesem Projekt, das Hard- und Software, Webdesign, 3D-Druck und Programmierung verbindet!

ELV



## Weitere Infos:

[1] ELV Bausatz Lichtorgel LED-WS2812: Artikel-Nr. 151851

[2] ESP32 Pico Kit: <https://docs.espressif.com/projects/esp-idf/en/latest/esp32/hw-reference/esp32/get-started-pico-kit.html#get-started-pico-kit-v4-board-front>

[3] Tannenzapfen-Lampe auf Thingiverse: <https://www.thingiverse.com/thing:3322642>

[4] OpenSCAD: <https://www.openscad.org/>

[5] ELVjournal 6/2019 – Von der Idee zum Objekt – Einstieg in das Computer-Aided Design (CAD), Teil 3: Artikel-Nr. 251078

[6] Download der Projekt-Dateien: Artikel-Nr. 252169

[7] Tastertyp: <https://www.segor.de/#Q=GQ12IP65-sw&M=1>

Alle Links finden Sie auch online unter: [de.elv.com/elvjournal-links](http://de.elv.com/elvjournal-links)

## Ihr Feedback zählt!

Das ELVjournal steht seit mehr als 40 Jahren für selbst entwickelte, qualitativ hochwertige Bausätze und Hintergrundartikel zu verschiedenen Technik-Themen. Aus den Elektronik-Entwicklungen des ELVjournals sind auch viele Geräte aus dem Smart Home Bereich hervorgegangen.

Wir möchten uns für Sie, liebe Leser, ständig weiterentwickeln und benötigen daher Ihre Rückmeldung: Was gefällt Ihnen besonders gut am ELVjournal? Welche Themen lesen Sie gerne? Welche Wünsche bezüglich Bausätzen und Technik-Wissen haben Sie? Was können wir in Zukunft für Sie besser machen?

Senden Sie Ihr Feedback per E-Mail an: [redaktion@elvjournal.com](mailto:redaktion@elvjournal.com)

oder per Post an: ELV Elektronik AG, Redaktion ELVjournal, Maiburger Str. 29–36, 26789 Leer, Deutschland

Vorab schon einmal vielen Dank vom Team des ELVjournals.

