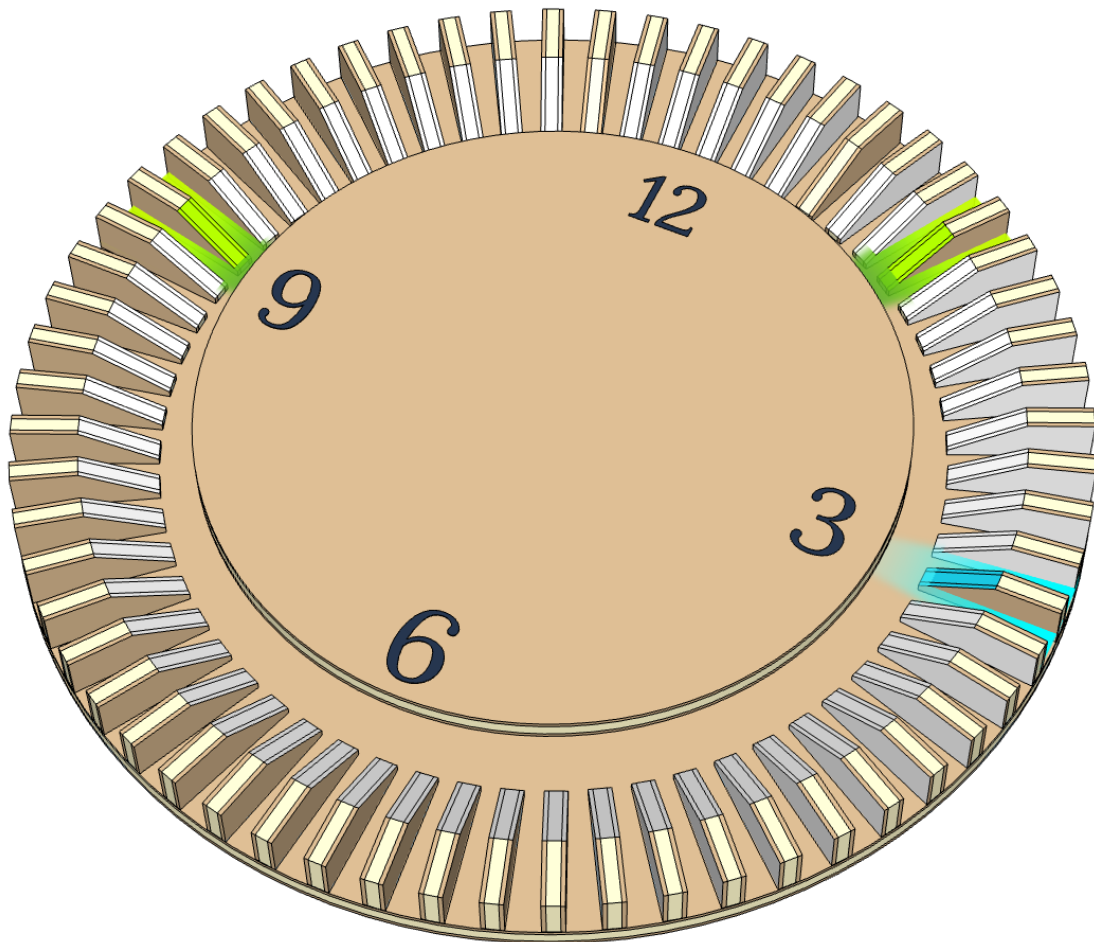


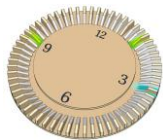
ECHTZEIT-UHR MIT MODUL DS1307 UND NEOPIXELN



Stand 13.9.2021

Diese Anleitung zeigt, wie du mit dem Echtzeitmodul DS1307, sowie einem PGLU-Controller oder einem Arduino Board, eine einfache Uhr bauen kannst.

Wenn du einen Neopixelstripe mit 60 Pixeln zur Zeitanzeige verwendest, öffnen sich dir viele tolle Möglichkeiten zum Bau einer einfachen Uhr mit schönen Lichteffekten. Suchst du eine ganz besondere Challenge? Dann lasse dir die Zeit mit einem LCD-Display anzeigen, und zwar in Worten.



INHALT

Echtzeit-Uhr mit Modul DS1307 und Neopixeln	1
1. Materialliste	2
2. Bauen und Gestalten einer Neopixel Uhr (zu Sketch 6.2)	3
3. Konstruktionsvorschläge Neopixel Uhr	4
4. Designvorschläge Neopixel Uhr	7
5. Anschlussschemen	9
5.1. Anschlussschema Neopixel Stripe und USB-A Kabel	9
5.2. Anschlussschema Zeitmodul und Potentiometer	10
6. Sketches	11
6.1. Grundfunktion des RTC-Moduls verstehen	11
6.2. Neopixel-Uhr Grundprogramm zum Verändern	12
6.3. Extra I: Zeit mit einem LCD-Display anzeigen	13
6.4. Extra II: Wort-Uhr «Schwiizertütsch» mit LCD-Display	13
7. Die Zeit der Uhr richtig einstellen (zu Sketch 6.2)	14
8. Arduino Libraries für das Arbeiten ohne Teacher's Box	14
9. Fehler beheben	14
9.1. Uhr	14
9.2. Teacher's Box	14

1. MATERIALLISTE

Hardware aus dem PGLU Shop:

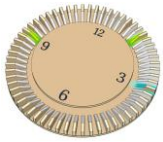
- > [Mikrocontroller PIXEL](#) oder Arduino Nano/Uno
- > [Neopixel Stripe mit 60 Pixeln zum Anzeigen der Zeit](#)
- > [Potentiometer 10 kΩ](#)
- > [Litzen](#)

Hardware aus dem Internet:

- > RTC Timer-Modul DS1307: [RTC GROVE DS1307](#)
- > [Knopfzelle 3V CR1225](#)
- > [USB-A Stecker mit offenem Kabelende für Stromversorgung](#)
- > Alternative: alte USB-A Kabel von Zuhause mitbringen lassen
- > Extra I+II: [LCD Display zum Anzeigen der Zeit](#)

Baumaterial:

- > Holz-Rondell mit $\varnothing=132$ mm, Dicke 12 mm
- > weiteres Holz für Zifferblatt, Dicke ca. 5 mm



2. BAUEN UND GESTALTEN EINER NEOPIXEL UHR (ZU SKETCH 6.2)

Die Neopixel Uhr besteht im Wesentlichen aus einer inneren Holzscheibe, auf deren Kante der Neopixel Streifen aufgeklebt wird. Der Durchmesser dieser Scheibe entspricht: **Länge Pixelstreifen $\div \pi$** .

Dies ergibt folgende Masse: z.B. Sperrholz Pappel **D = 12mm, Ø 132mm**

Hinweis: Anleitung zum Löten der Neopixel: PGLU.CH > BAU > NEOPIXEL LICHT & MUSIK

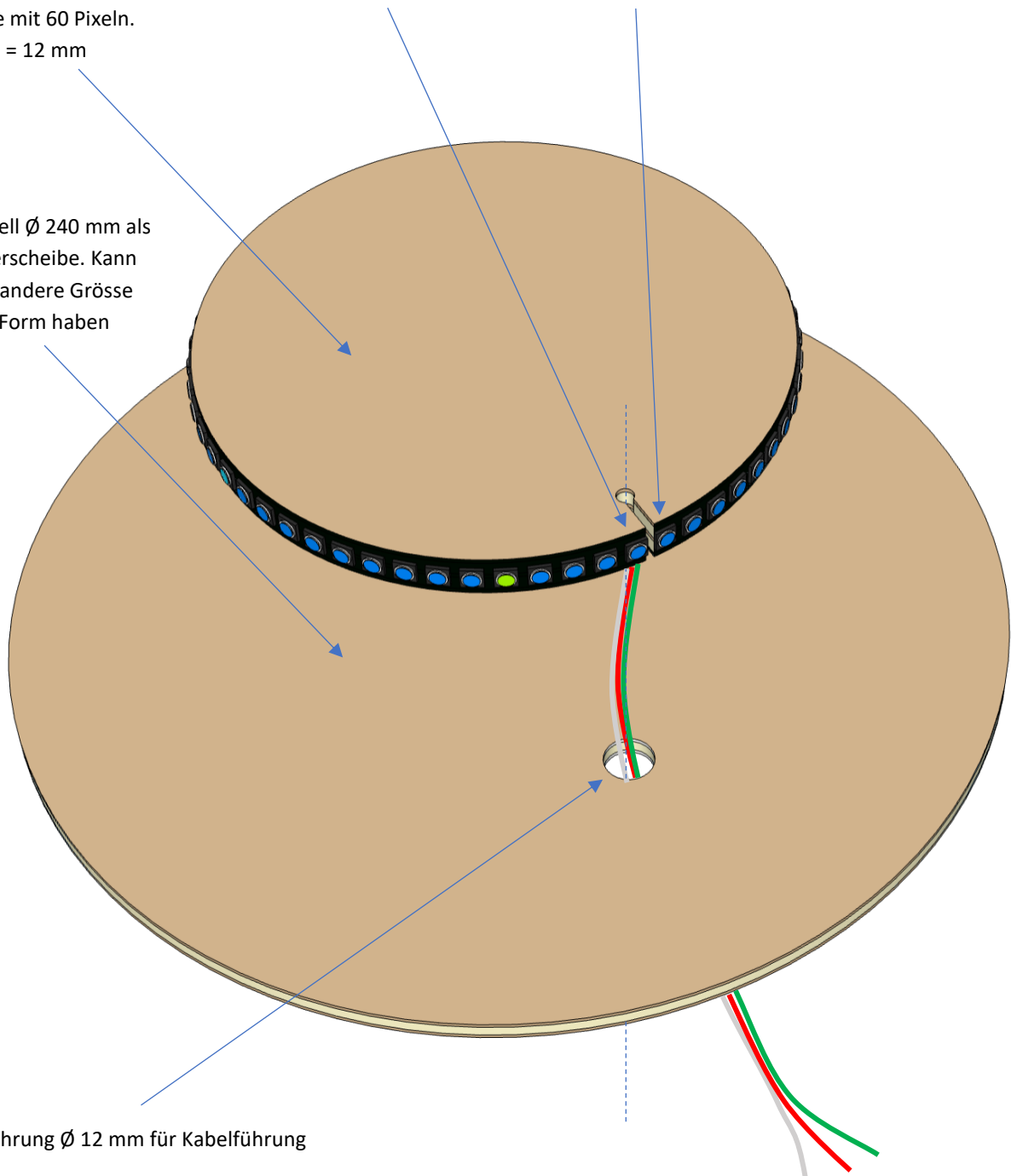
Rondell Ø 132 mm für
Stripe mit 60 Pixeln.
Dicke = 12 mm

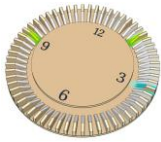
Pixel Nr. 1 = 12:01 Uhr

Pixel Nr. 60 = 12:00 Uhr

Rondell Ø 240 mm als
Trägerscheibe. Kann
auch andere Grösse
oder Form haben

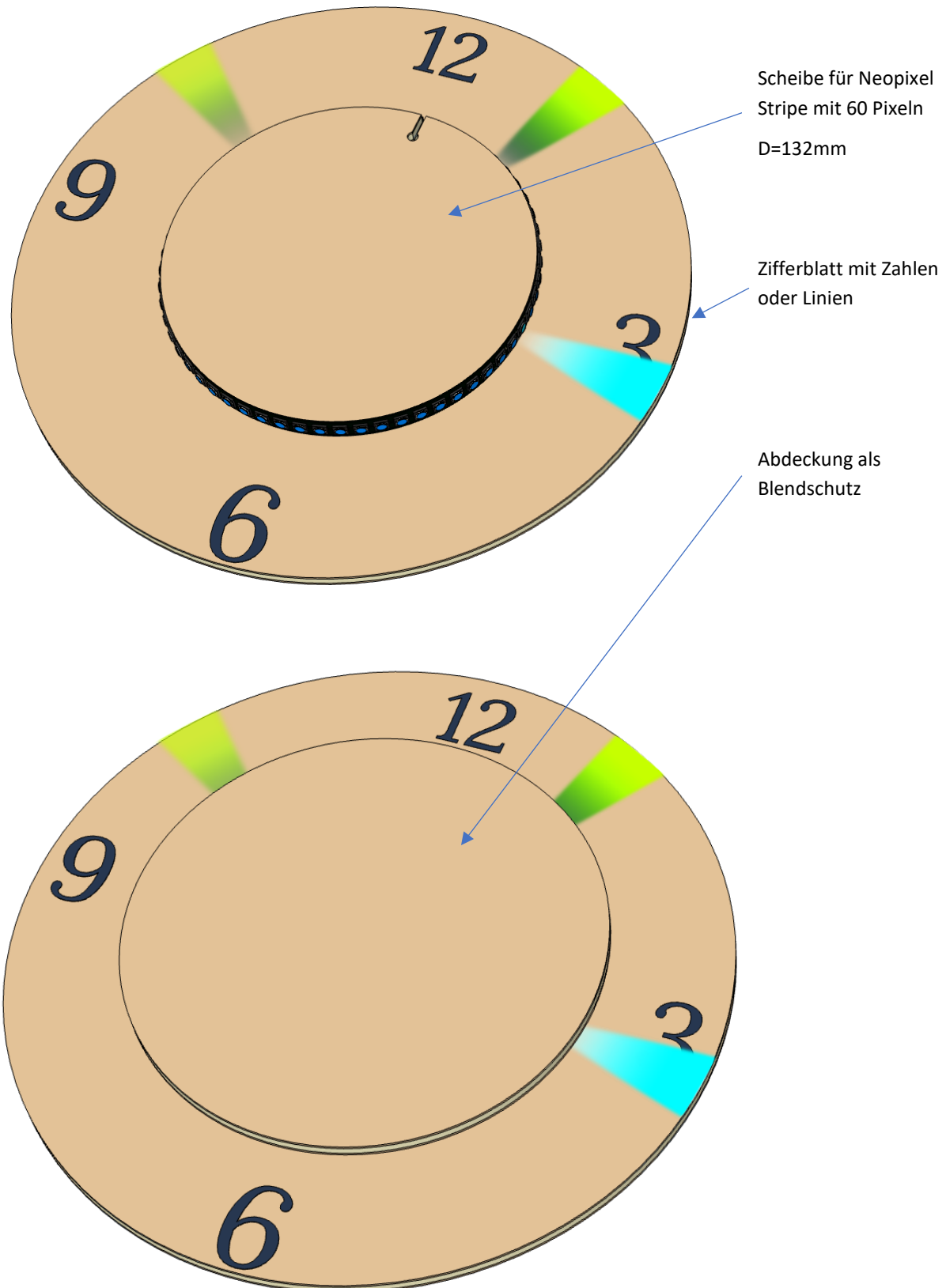
Bohrung Ø 12 mm für Kabelführung

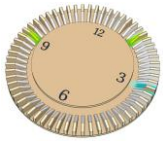




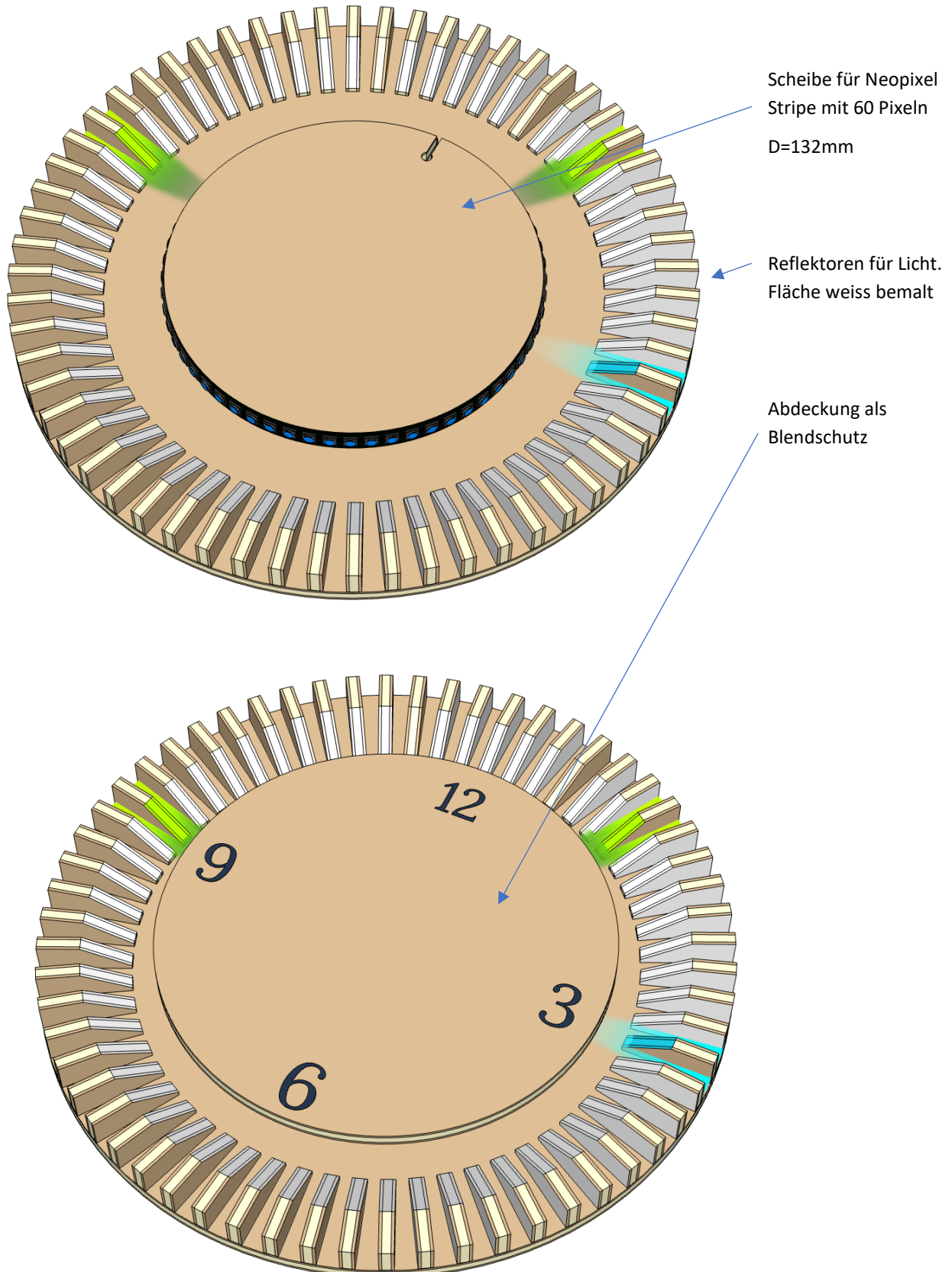
3. KONSTRUKTIONSVORSCHLÄGE NEOPIXEL UHR

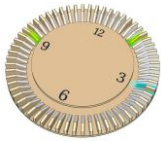
Beispiel: Simple Version mit gezeichnetem Zifferblatt auf Grundplatte und mit Blendschutz





Beispiel: Variante mit Lichtreflektoren und Blendschutz





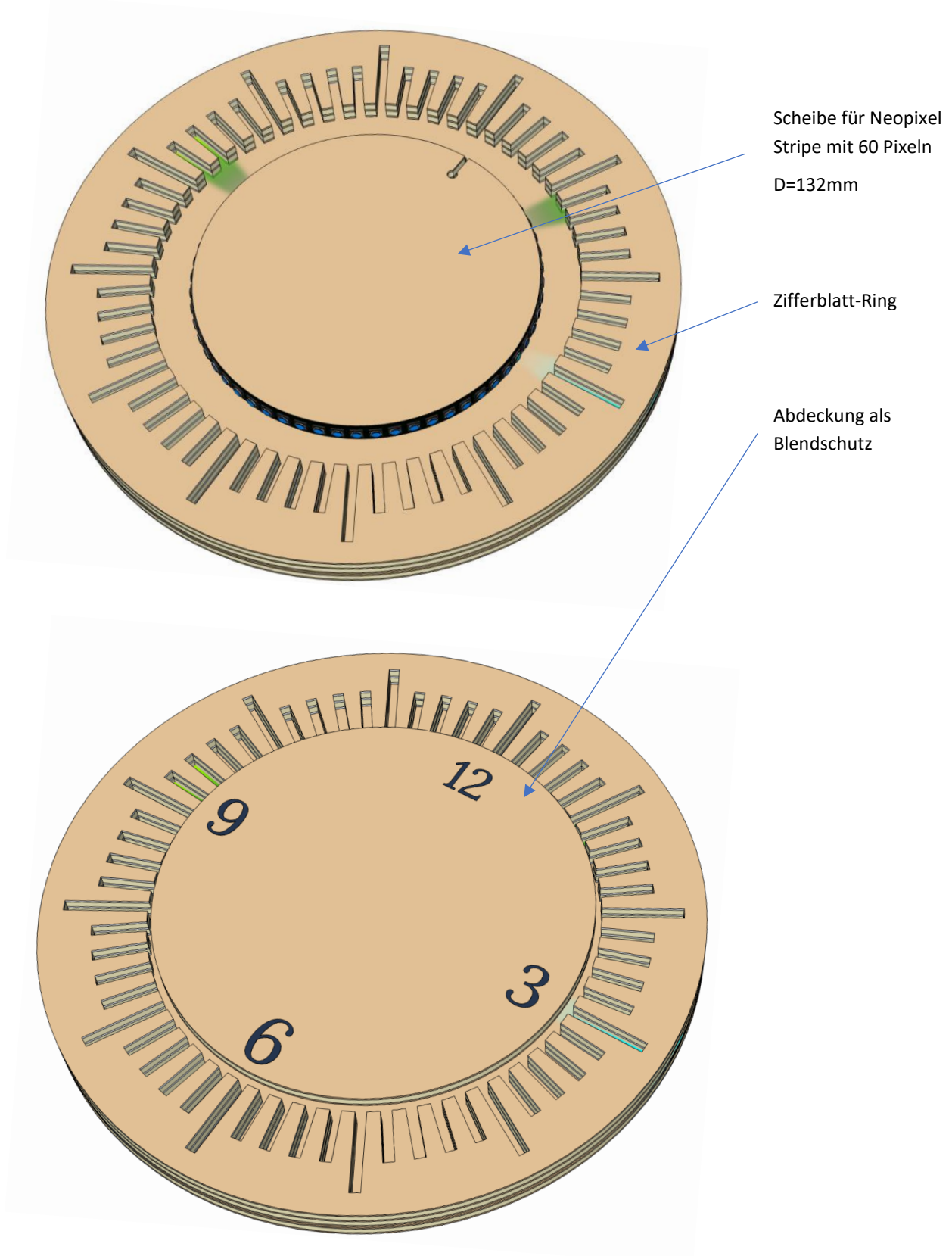
Beispiel: Variante mit Zifferblatt-Ring, siehe diese Videos:

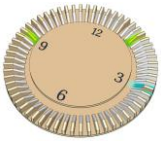
[vimeo.com/594 729 206](https://vimeo.com/594729206)

[vimeo.com/592 318 408](https://vimeo.com/592318408)

Schnittvorlagen für Lasercutting:

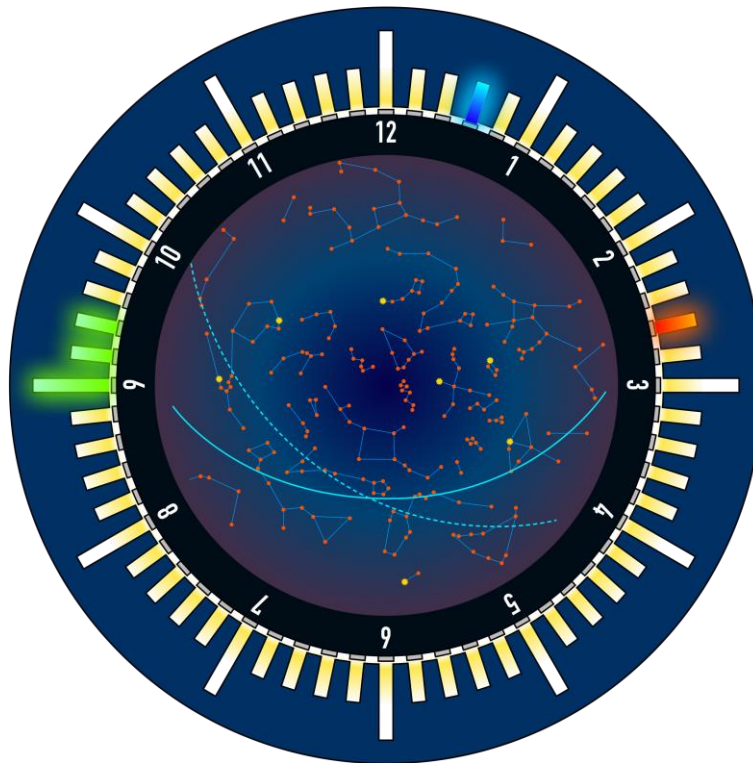
[PGLU.CH](https://pglu.ch) > ANLEITUNG > HILFREICHE DOKUMENTE > 3D-DRUCK UND LASERCUTTING



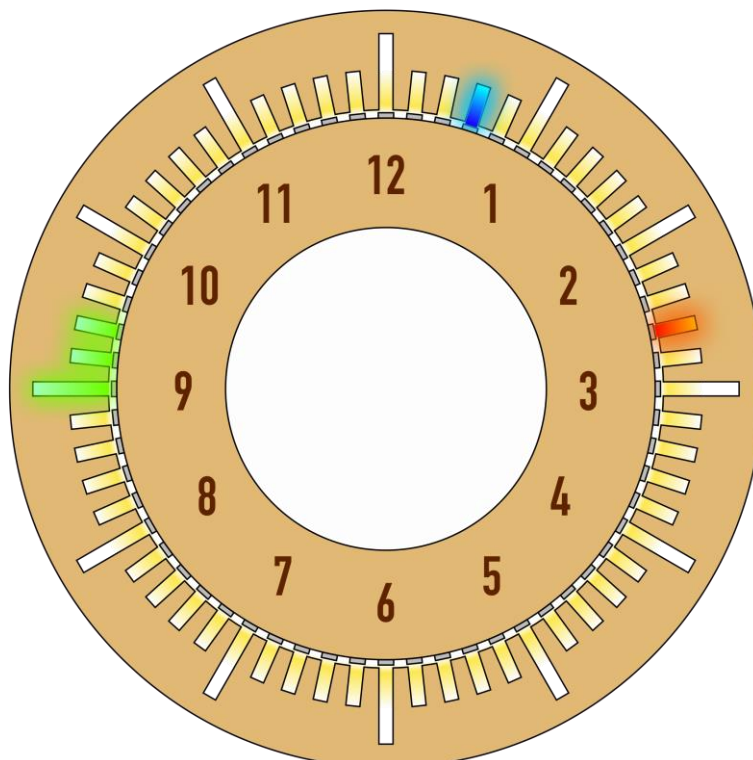


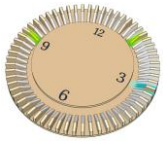
4. DESIGNVORSCHLÄGE NEOPIXEL UHR

Beispiel "Sternkarte"

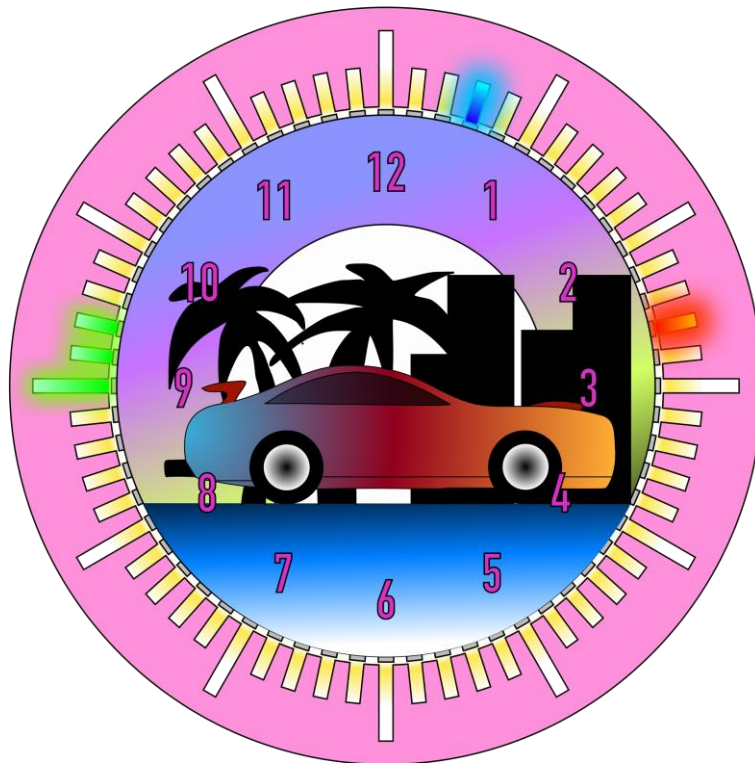


Beispiel "Klassisch"





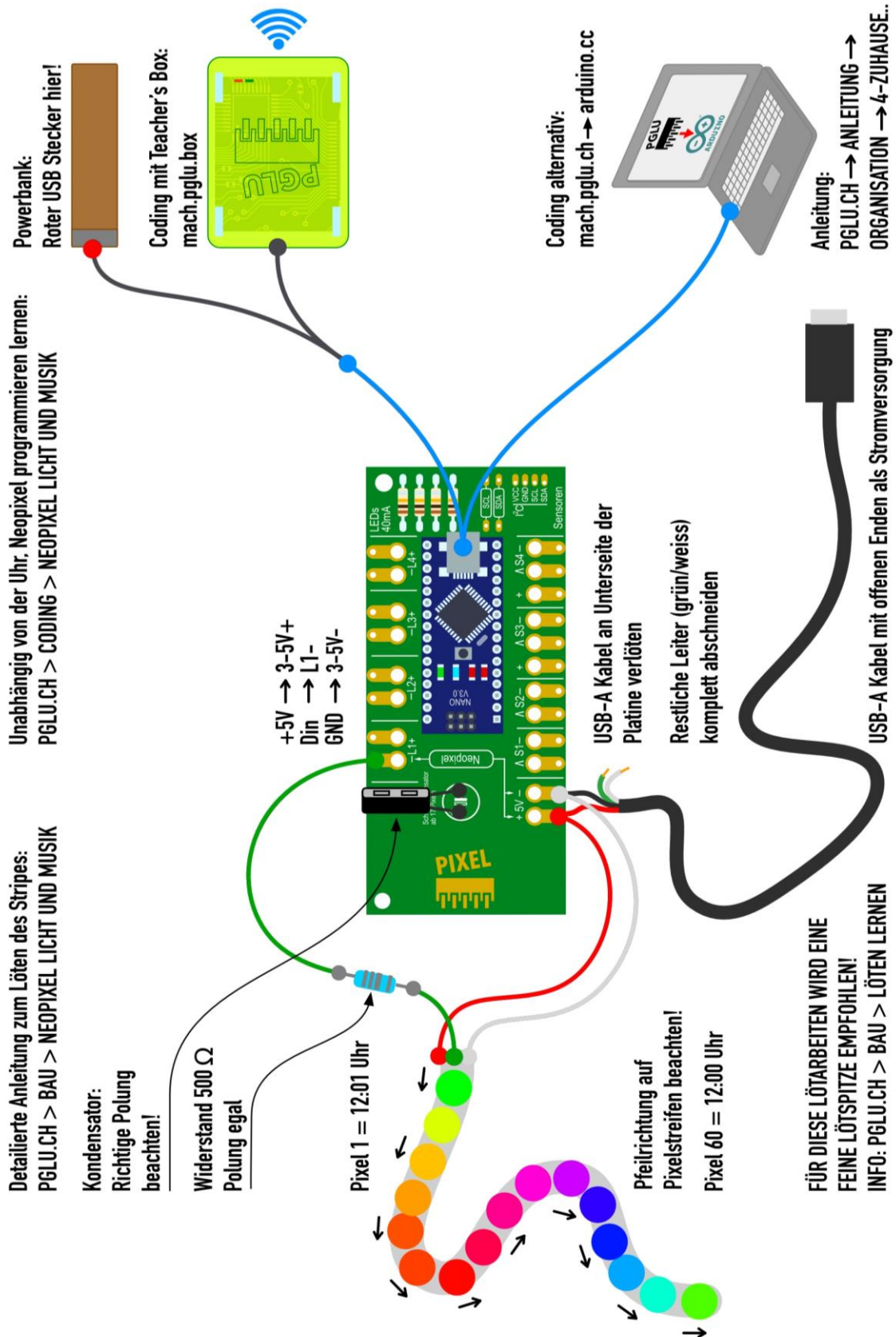
Beispiel "Las Vegas"

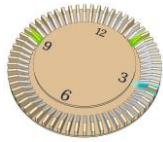




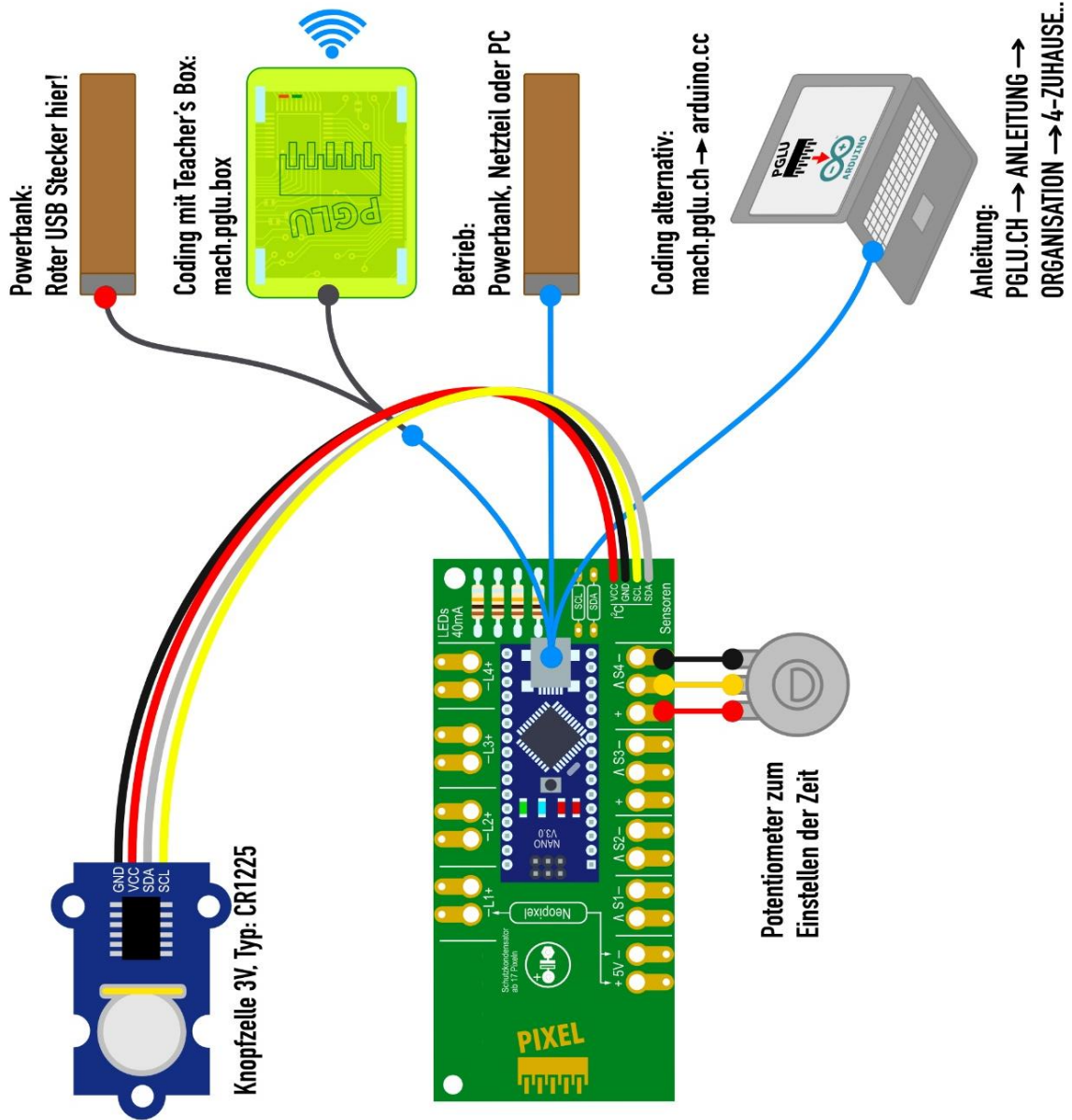
5. ANSCHLUSSSCHEMEN

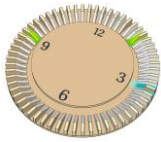
5.1. ANSCHLUSSSCHEMA NEOPIXEL STRIPE UND USB-A KABEL





5.2. ANSCHLUSSSCHEMA ZEITMODUL UND POTENTIOMETER





6. SKETCHES

6.1. GRUNDFUNKTION DES RTC-MODULS VERSTEHEN

Für das einmalige Einstellen der aktuellen Zeit, diese Blöcke einsetzen und Sketch hochladen. Danach Blöcke wieder entfernen.

Vor Hauptloop: 1x

- `#include <TimeLib.h>`
- `#include <Wire.h>`
- `#include <DS1307RTC.h>`
- `setSyncProvider(RTC.get());`
- `setSyncInterval(120);`
- `pausiere 3000 ms`
- `setTime(15,11,00,6,7,2021);`
- `RTC.set(now());`

Hauptloop: 100'000 mal pro Sekunde

- `second()`
- `hour()`
- `minute()`

Verwende diese Blöcke an beliebigen Stellen in deinem Sketch

Bild: Grundfunktionen zum Einstellen oder Abrufen der Zeit

Beschreibung: Dieses Sketch zeigt die Blöcke, mit denen du Zeit und Datum in unterschiedlichen Formaten in deinen Projekten nutzen kannst. Die Library [<DS1307.h>](#) stellt dabei die Kommunikation mit dem Echtzeitmodul her, während [<TimeLib.h>](#) die Signale des Moduls in lesbare Zeit- und Datumsangaben übersetzt.

Die Zeit muss nur einmal eingestellt werden, solange die Batterie im Modul eingesetzt ist. Setze dazu die Blöcke `setTime(Std,Min,Sek,Tag,Mon,Jahr)` und `RTC.set(now())` in «vor Hauptloop 1x» ein (siehe Screenshot oben). Den Hauptloop kannst du leerlassen. Lade das Sketch genau zur angegebenen Zeit (im Beispiel 15:11 Uhr, am 6.7.2021) auf den Mikrocontroller und entferne die Blöcke für das einmalige Einstellen der Zeit wieder.

Eine etwas elegantere Art, die Zeit der Uhr jederzeit mit einem Poti einzustellen, findest du in Beispiel 2 (Neopixel-Uhr).

>Code mit 3-fach-Klick kopieren und in [Editor](#) einfügen

```
const int ledPin = 13; // LED pin
const int potPin = A0; // Potentiometer pin
const int buzzerPin = 8; // Buzzer pin

// RTC module pins
const int rtcDataPin = 2; // DS1307 data pin
const int rtcVccPin = 5; // DS1307 VCC pin
const int rtcGndPin = GND; // DS1307 GND pin

// TimeLib.h and DS1307RTC.h
#include <TimeLib.h>
#include <DS1307RTC.h>

// RTC module
RTC_DS1307 rtc;

// Time variables
Time_t nowTime;
Time_t lastTime;

// Time setting variables
int year = 2021;
int month = 7;
int day = 6;
int hour = 15;
int min = 11;
int sec = 00;

// Time reading variables
int yearRead;
int monthRead;
int dayRead;
int hourRead;
int minRead;
int secRead;

// Time setting function
void setTime() {
  rtc.setTime(year, month, day, hour, min, sec);
}

// Time reading function
void readTime() {
  rtc.getTime(yearRead, monthRead, dayRead, hourRead, minRead, secRead);
}

// Time setting and reading function
void setTimeAndRead() {
  setTime();
  readTime();
}

// Time setting and reading function
void setTimeAndRead() {
  setTime();
  readTime();
}
```



Beschreibung: Dieses Sketch zeigt, wie du mit einfachen Neopixel-Blöcken die Zeit auf einem Stripe mit 60 Pixeln anzeigen kannst. Experimentiere mit der Farbe und Anzahl der Pixel für die einzelnen Zeiger.

bestimmen, welcher Zeiger, welchen überdecken soll. Der Pixelblock, der im Hauptloop als letzter eingefügt ist, wird auf dem Stripe «zuoberst» angezeigt. Experimentiere!

[illegible]

A circular LED display with a black center and a white outer ring. The center features a white silhouette of a horse in profile, facing left. The outer ring is composed of numerous small, round LEDs. Most of these LEDs are illuminated with a warm orange-red light. A few LEDs are illuminated with a bright blue light, positioned at approximately the 10 o'clock and 7 o'clock positions. A black power cable is connected to the back of the display. The device is resting on a light-colored wooden surface. In the background, a blue-handled screwdriver and a black circular object with a white pattern are visible.

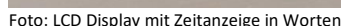
Foto: Neopixel Clock mit Papier-Cover

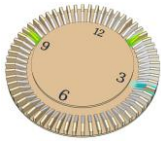


Beschreibung: Dieses Sketch zeigt, wie du Zeit und Datum als klassische Angaben auf einem LCD-Display darstellen kannst. Das [hier verwendete Display](#) wird in [Challenge 7 genauer beschrieben](#) und erlaubt die Darstellung der Zeit in unterschiedlichen Farben, welche auch im Verlauf der Zeit angepasst werden können. Finde heraus, wie das geht!

[illegible]

<https://www.arduino.cc/reference/de/language/variables/data-types/string/>

[illegible]



7. DIE ZEIT DER UHR RICHTIG EINSTELLEN (ZU SKETCH 6.2)

Falls du das Sketch 6.2 für deine Uhr verwendest, lässt sich die Zeit so einstellen:

- > Controller mit Strom verbinden oder **Reset** drücken (schwarze Taste auf Nano)
- > **12.00 Uhr leuchtet orange**: drehe sofort kurz am Poti
- > Stundenzeiger-Modus: **der Zeiger leuchtet für 8 Sekunden pink**: stelle die Stunde ein
- > Minutenzeiger-Modus: **der Zeiger leuchtet für 8 Sekunden cyan**: stelle die Minute ein
- > **12.00 Uhr leuchtet kurz grün**: die Zeit wurde erfolgreich eingestellt
- > **12.00 Uhr leuchtet kurz rot**: die Zeit wurde nicht eingestellt, die Uhr läuft ab 12.00 Uhr

8. ARDUINO LIBRARIES FÜR DAS ARBEITEN OHNE TEACHER'S BOX

Falls du ohne Teacher's Box arbeitest, musst du diese Libraries in deiner Arduino IDE installieren:

- > [DS1307RTC.h](#) für den Betrieb des Moduls
- > [TimeLib.h](#) für die Übersetzung des Signals in Sekunden, Minuten, Stunden, Tage, Monate, Jahre
- > [Wire.h](#) für den I2C Anschluss
- > [FastLED](#) für die Neopixel

Alle Libraries sind in der Teacher's Box ab V4.2 enthalten.

9. FEHLER BEHEBEN

9.1. UHR

Möglicher Fehler:

- > Nach dem Aufstarten der Uhr leuchtet 12 Uhr zuerst **orange**, dann **rot**
- > Danach läuft die Uhr ab 12.00 Uhr anstelle in der richtigen Zeit

Mögliche Ursachen und Behebung:

- > Die Batterie des RTC-Moduls ist leer oder falsch eingesetzt
- > Das RTC-Modul ist nicht richtig angeschlossen. Anschlussschema und Lötstellen überprüfen
- > Die Pixel verbrauchen zu viel Strom > Helligkeiten der Hintergrundpixel reduzieren!
- > Die Uhr wird über die Teacher's Box mit Strom versorgt, was zu Störungen führen kann. Uhr über Handyladegerät oder Powerbank mit Strom versorgen

9.2. TEACHER'S BOX

Möglicher Fehler:

- > Programmname wird nach Speichern **rot** angezeigt

Mögliche Behebung:

- > Kopie des Sketches speichern und damit weiterarbeiten
- > Uhr von Teacher's Box trennen, wenn das Sketch gespeichert wird. Dann Uhr wieder anschliessen und Sketch hochladen
- > Schutzkondensator und Schutzwiderstand anbringen/kontrollieren (siehe Anschlussschema 5.1)
- > Mit Arduino IDE arbeiten ([Anleitung](#))