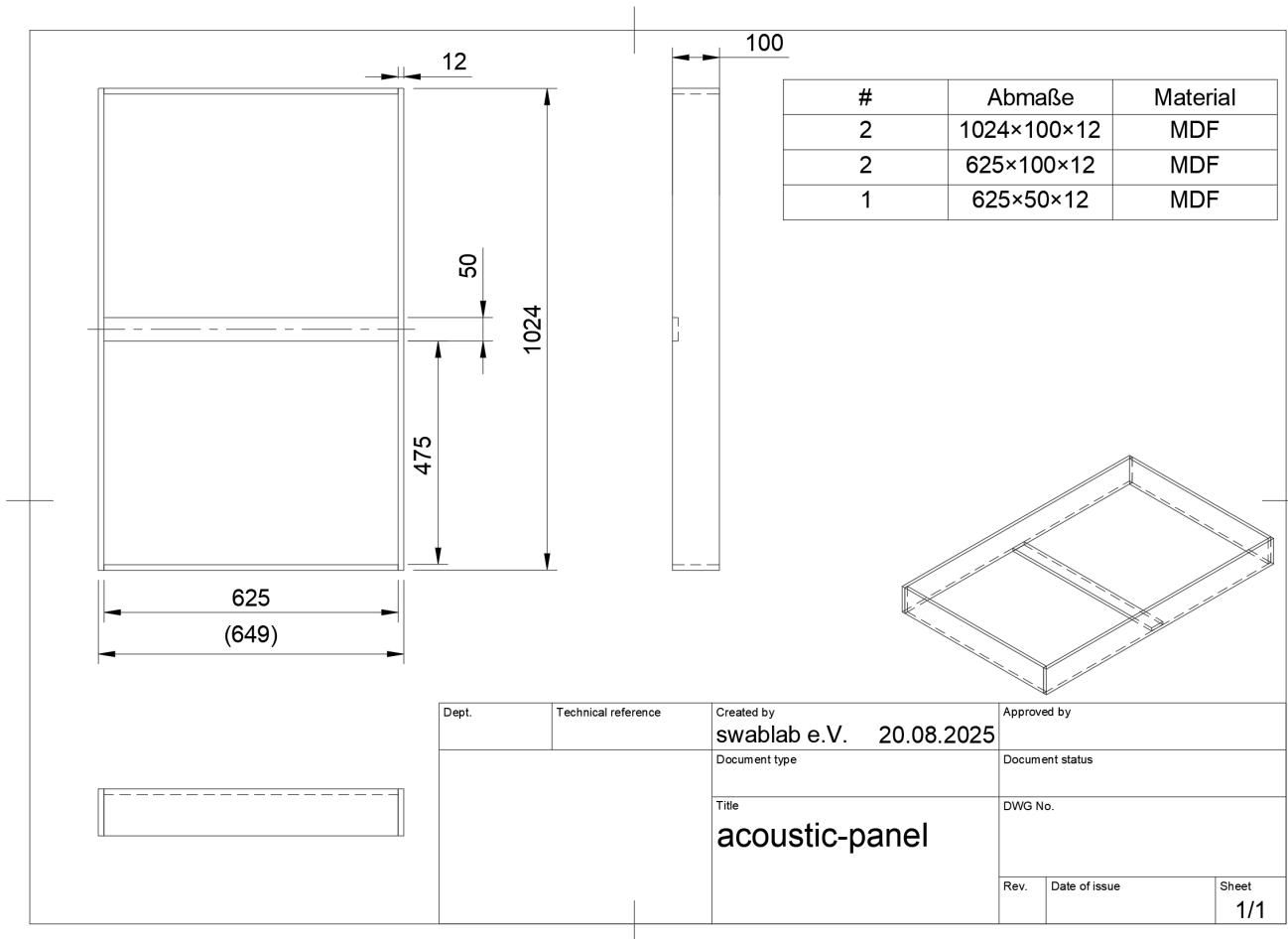


# Aktive Projekte

- [Decke OG](#)
- [Audio Gästebuch](#)
- [Klapphocker](#)
- [Klappstuhl](#)
- [Disco-Kugel](#)
- [lightpickrally](#)

# Decke OG

## Akustik-Paneele



## LED Beleuchtung

- 2 Balkenreihen á 2 separate Balken, mit direkter Beleuchtung nach unten und indirekter Beleuchtung nach oben, getrennte Kreise
- Netzteil kann max. 75W, 4 LED Module müssten an einem Netzteil dran gewesen sein --> 18,75W pro Modul
- Gesamt-Beleuchtung sollte nicht >300W verbrauchen, ideal ~200W

Fragen:

- geben die Netzteile immer 75W aus? -> nein. die verbrauchen soviel wie verbraucht wird. die effizienz geht in keller je weniger verbraucht wird. -> wir können uns überlegen wie wir die leds verteilen, brauchen aber immer zwei netzteile pro balken
- Abwärme? -> testen



# Audio Gästebuch

Idee ist ein mobiles Audio-Gästebuch für beispielsweise Hochzeiten, bei denen Gäste eine Nachricht einsprechen können die dann gespeichert wird.

Inspiriert wurden wir von folgendem Projekt: <https://github.com/playfultechnology/audio-guestbook>

Initiales Brainstorming:

- Software (zu finden unter <https://git.swablab.de/projects/audio-guestbook>)
  - Audio abspielen
  - Audio aufnehmen
  - Abspeichern auf SD Karte
  - (optional) Upload in die Cloud
  - (optional) Hardware Interrupts um Energie zu sparen
- Hardware
  - ESP Board kaufen
  - Kabellos mit Batterie oder USB C
  - Soll in das alte Telefon passen
  - Telefon restaurieren
  - Hörer verbinden
  - Taster zum Auflegen verbinden

Mit dem esp32a1s hatten wir Probleme bei der Aufzeichnung. Das Board hat einen Bug und die OnBoard Mikrofone lassen sich nicht deaktivieren. Aus diesem Grund haben wir noch ein teensy4.0 <https://www.pjrc.com/store/teensy40.html> mit Audio Shield gekauft und den ersten Prototypen mit diesem Board erstellt.

Der Code in unserem Git lässt sich wahlweise mit dem Teensy Board oder dem esp32a1 kompilieren.

Für den esp32a1s müssen jedoch die OnBoard-Mikrofone zugeklebt oder entfernt werden und die aktuellste Version ist nicht getestet.

## Hardware

Wir haben uns für das Teensy Board entschieden.

Schematic, Rev D [schematic\\_audio4.png](#)

<https://www.pjrc.com/store/teensy41.html>

# Ablauf

## Wav to MP3

```
find . -name '*.wav' -print0 | xargs -0 -I{} sh -c 'base=$(basename "{}" .wav); ffmpeg -i "{}" -codec:a libmp3lame -vn -q:a 0 -loglevel error "${base}.mp3"'
```

# Klapphocker

## Projektidee

Ziel dieses Projekts war es, einen transportablen und funktionalen Klapphocker aus einfach zu bearbeitbarem Holz zu gestalten. Dieser sollte möglichst ein kleines Packmaß haben.

## Verwendete Materialien

- 210cm Glattkantbrett 60x18mm: [Link Bauhaus](#) (Sitzfläche (groß/klein) / Verstrebung (kurz/lang))
  - 340cm Rahmenholz 48x24mm: [Link Bauhaus](#) (Füße (innen/außen) / Sitzflächenhalter (innen/außen))
  - 11,8cm Rundstab D35: [Link Bauhaus](#)
  - 2x Holzschraube (in der Mitte auf den Rundstab): 4,5x80mm
  - 2x Holzschraube (für oberes Gelenk) 3,5x40mm
  - 2x M4x45 Senkschraube (Gelenk in der Mitte)
  - 2x Sicherungsmutter M4
  - 2x Unterlegscheibe M4
  - 28x Holzschraube 3x30mm
- 

## Benötigtes Werkzeug

- Kappsäge mit Winkelverstellung
- Tischkreissäge
- Schleifpapier
- Akkuschauber (oder Standbohrmaschine)
  - Holzbohrer (Ø: 4,5mm; 5mm; 12mm)
  - Senker
  - Bitset

## Arbeitsschritte

### 1. grober Zuschnitt

Zunächst werden die einzelnen Bretter grob zugeschnitten:

- Glattkantbrett 60x18mm:  
5x 31cm  
1x 26,2cm  
1x 21,4cm  
=202,6cm (plus Sägeblattbreite)
- Rahmenholz 48x24mm:  
4x 34cm  
4x 49,5cm  
=334cm (plus Sägeblattbreite)
- Rundstab D35:  
1x 11,8cm

## 2. fein Zuschnitt

Nach dem groben Zuschnitt müssen noch die Winkel an den Enden der Hölzer gesägt werden. Hierfür am besten die einzelnen Zeichnungen (B\_Fuss\_breit und B\_Fuss\_schmal) verwenden.

Außerdem muss ein 31 cm langes Glattkantbrett der Länge nach in der Mitte geteilt werden, dass zwei etwa gleich große 28,5mm breite Stücke entstehen. Am besten geht das auf einer Tischkreissäge

**\*\*Wichtig:\*\*** Hier müssen zwingend Schiebestöcke verwendet werden, da dieser Schritt sehr gefährlich ist.

## 3. Bohrungen

Anschließend werden alle notwendigen Bohrungen durchgeführt. Hierfür am besten die einzelnen Zeichnungen (B\_Fuss\_breit; B\_Fuss\_schmal; B\_Sitzflaeche\_schmal und B\_Sitzflaeche\_breit) verwenden. Hier sollte darauf geachtet werden, dass die Bohrungen senkrecht sind, da der Zusammenbau und die spätere Funktion des Klapphockers sonst nicht gegeben ist.

## 4. Schleifen

Im Anschluss des Sägens und Bohrens müssen alle Bretter geschliffen werden. Vor allem die fünf 31cm langen Glattkantbretter sind wichtig, da diese später die Sitzfläche bilden.

**\*\*Tipp:\*\*** Je sauberer hier gearbeitet wird, desto bequemer ist der Klapphocker anschließend.

## 5. Montage

Ich wünsche viel Freude beim Nachbauen und kreative Stunden in der Werkstatt!



# Klappstuhl

## Materialliste

### Sitzfläche

- 2x 62.5 x 4.5 x 2 cm
- 4x 37.5 x 4.5 x 2 cm
- 16x 4x45 Spax

### Rückenlehne

- 2x 80 x4.5 x 2 cm
- 3x 37.5 x4.5 x 2 cm
- 12x 4x45 Spax
- 1x Stofftasche 50 x 50 cm

5,5 m Holzlatten/Stuhl

ca 110m Holzlatten für 20 Stühle

28 Spax/Stuhl

560 spax für 20 Stühle

## Zusammenbau

### Schritt 1: Holzlatten zusammenstellen

Alle in der Materialliste aufgeführten Holzlatten an bereitgestellter Materialausgabe abholen und nach den Größen sortiert an den gewünschten Arbeitsplatz legen.

### Schritt 2: Holzlatten abschleifen

Alle Kanten und Flächen des Baumaterials abschleifen, bis keine scharfen Kanten oder größere Spreißen mehr zu sehen sind. Das hierfür benötigte Schleifpapier ist ebenfalls bei der Materialausgabe zu finden.

# Disco-Kugel

Wir bauen eine Disco-Kugel, welche auf Geräusche und Musik reagieren kann und dadurch ihre Farbe ändert. Wir haben diese Disco-Kugel als ein Sommerferienprogramm entworfen, um den Kindern das Löten näher zu bringen.

Nachfolgend findet ihr eine Anleitung, sowie alle benötigten Teile. Viel Spaß beim Nachbauen!

## Bauteile

- Arduino (<https://amzn.to/3AigJXC>\*)
- Mikrofon (\*)
- Lochrasterplatine (\*)
- LED-Band (\*)
- (Welcher Durchmesser) Kabel (\*)
- Holzgehäuse
- Diffusor + Zwischenrahmen ()

## Software

## Hardware

## 3D Druck

## Holzgehäuse

Um die Elektronik aufzubewahren verwenden wir ein selbstgebautes Holzgehäuse.

## Zusammenbau

---

Bei den mit "\*" markierten Links handelt es sich um Affiliate-Links.

Wenn ihr etwas über diese Links kauft, erhalten wir eine kleine Provision. Für euch ändert sich dabei nichts.

Dadurch könnt ihr uns ohne Aufwand unterstützen!  
Wo ihr eure Sachen kauft, bleibt aber natürlich euch überlassen!

# lightpickrally

Dieses Projekt soll eine Kombination aus dem Open LED Race und einem Regalsystem mit Pick by Light werden.

Die LED-Streifen werden unterhalb unserer Regale geklebt und es wird ein Arduino Nano zur Ansteuerung der LEDs verwendet.

Der Arduino hört auf einer HTTP-Schnittstelle auf Anforderungen um einzelne LEDs aufleuchten zu lassen.

Sobald ein Knopf des LED-Races gedrückt wird, schaltet die Software in den Race Modus. Nach einem Ablauf von einer bestimmten Zeit wechselt die Software wieder in den Pick by Light Modus.

Die LEDs werden im Directus zu den Inventaren gepflegt und der HTTP Post Aufruf erfolgt über eine separate Web Anwendung.

Code für den Arduino Nano: <https://git.swablab.de/projects/lightpickrally>

## Display-Pi

- Installieren von Alpine armv7
- usercfg.txt (neben config.txt) folgenden Inhalt: `dtoverlay=vc4-kms-v3d`
- Display per HDMI anschließen
- `setup-alpine` durchlaufen und user `pi` anlegen
- [Autologin einrichten](#)
  - `nano /usr/sbin/autologin` mit folgendem Inhalt:

```
#!/bin/sh
exec login -f pi
```
  - `chmod +x /usr/sbin/autologin`
  - `/etc/inittab` bearbeiten:

```
tty1::respawn:/sbin/getty 38400 tty1 -n -l /usr/sbin/autologin
```
- [Weston installieren](#)
  - `setup-wayland-base`
  - `setup-devd udev`
  - `apk add weston weston-backend-drm weston-backend-wayland weston-shell-desktop seatd`
  - `rc-update add seatd`
  - `rc-service seatd start`
  - `adduser pi seat`
- User einrichten (`su -l pi`)
  - `mkdir ~/xdg`
  - `.profile` anlegen:

```
while true
```

```
XDG_RUNTIME_DIR=/home/pi/xdg_weston  
done
```

- `.config/weston.ini` anlegen:

```
[core]
```

```
mode=kiosk
```

```
idle-time=0
```

```
[output]
```

```
name=HDMI-A-1
```

```
transform=rotate-270
```

```
[shell]
```

```
locking=false
```

```
[autolaunch]
```

```
path=/home/pi/lightpickrally
```

- Persistieren (wieder als root)

- `lbu add /home/pi /usr/sbin/autologin`

- `lbu ci`