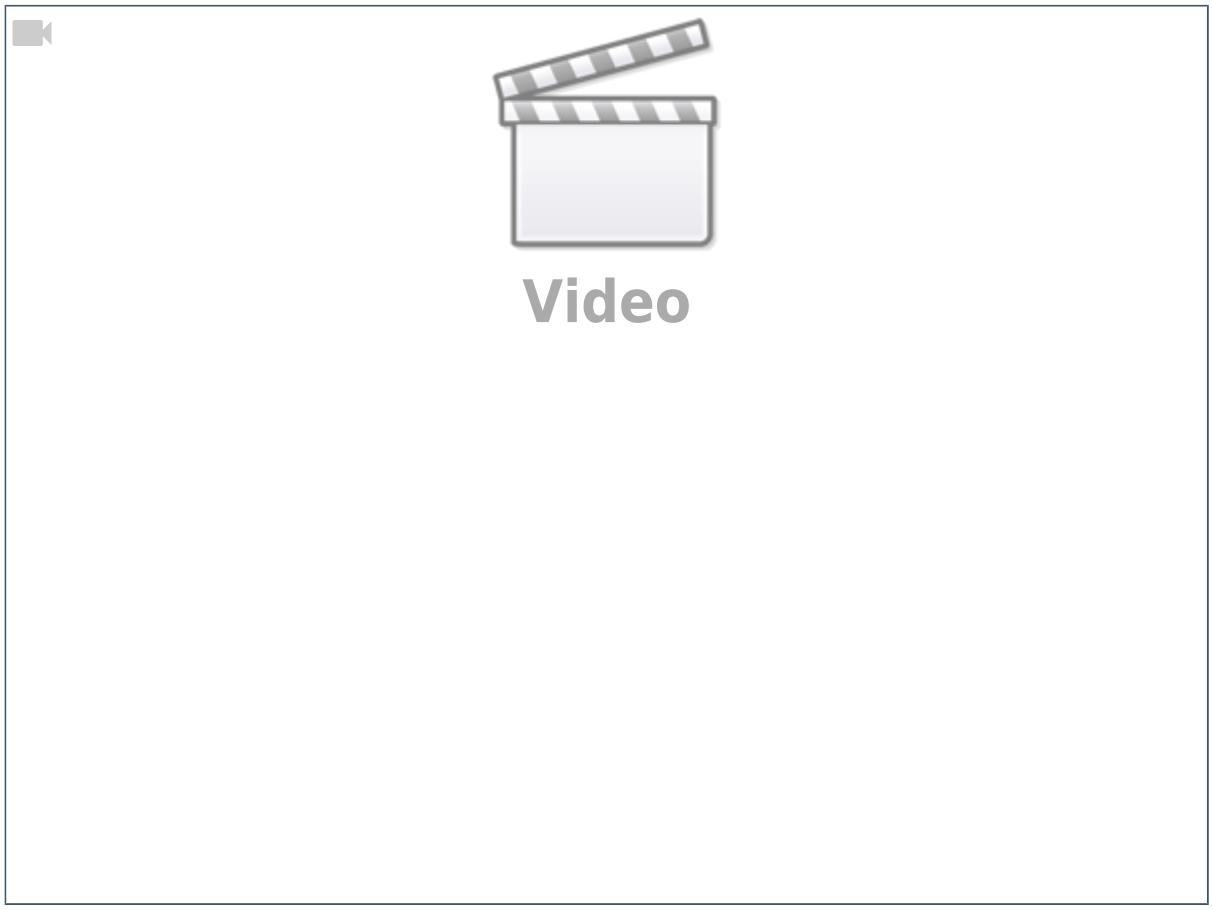
# Villa mit UPS-Zustellung

Grundlage für die Gestaltung dieser Szenerie sind die Modelle „Bürgerhaus mit Atelier in Bonn“ (Kibri 39103) und dem UPS-Fahrzeug (Herpa 097 123). Beide Objekte sind entsprechend modifiziert und mit einer Eigenkonstruktion zu einer animierten Einheit zusammengefügt. Das Haus wurde mit 14 Zimmern, zwei Hausfluren sowie einem Laden auf der Rückseite ausgestattet. Alle Zimmer auf der Vorderseite und der Laden bekamen eine detaillierte Inneneinrichtung. Hinzu kommt noch ein Kamin, zwei separate TV-Geräte, zwei Stehlampen, eine Schaufensterpuppe mit beleuchtetem Podest und eine Servo-gesteuerte Haustür. Auch das UPS-Auto wurde mit Abblend- und Rücklicht sowie je drei Blinkern auf jeder Seite ausgestattet. Des Weiteren kann die seitliche Tür mechanisch geöffnet werden und ein dazugehöriger UPS-Mann liefert die Post bis an die Haustür. Dort wird sie dann von der Hausdame in Empfang genommen. Auch eine entspreche Konversation zwischen beiden Personen wurde mit eingebunden.

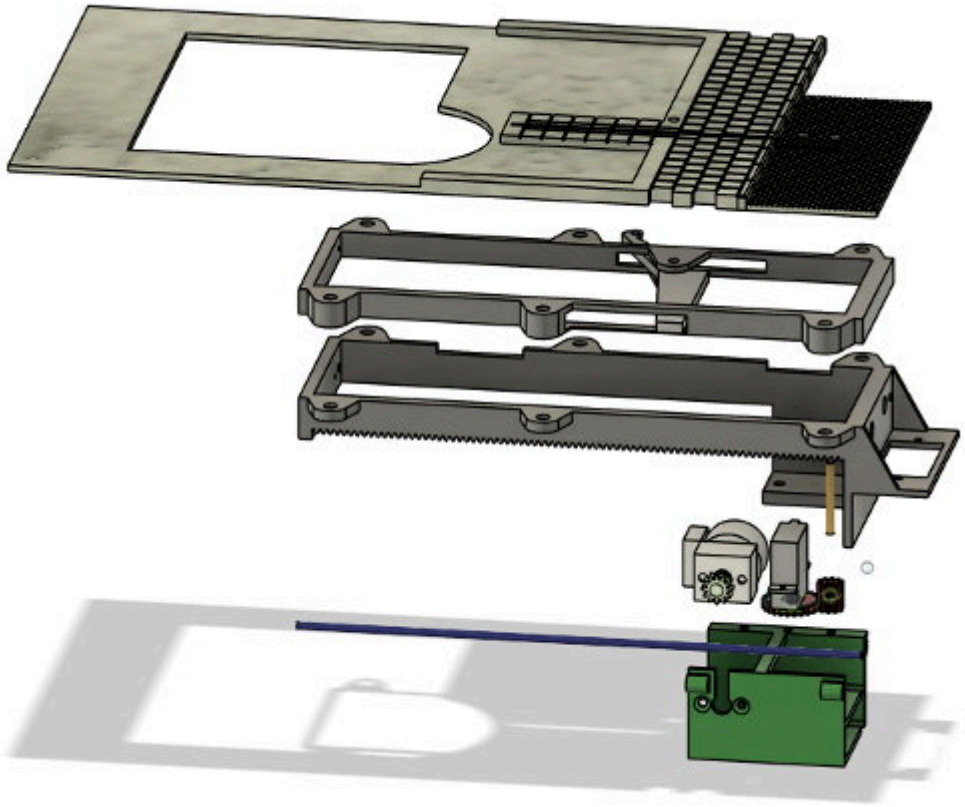


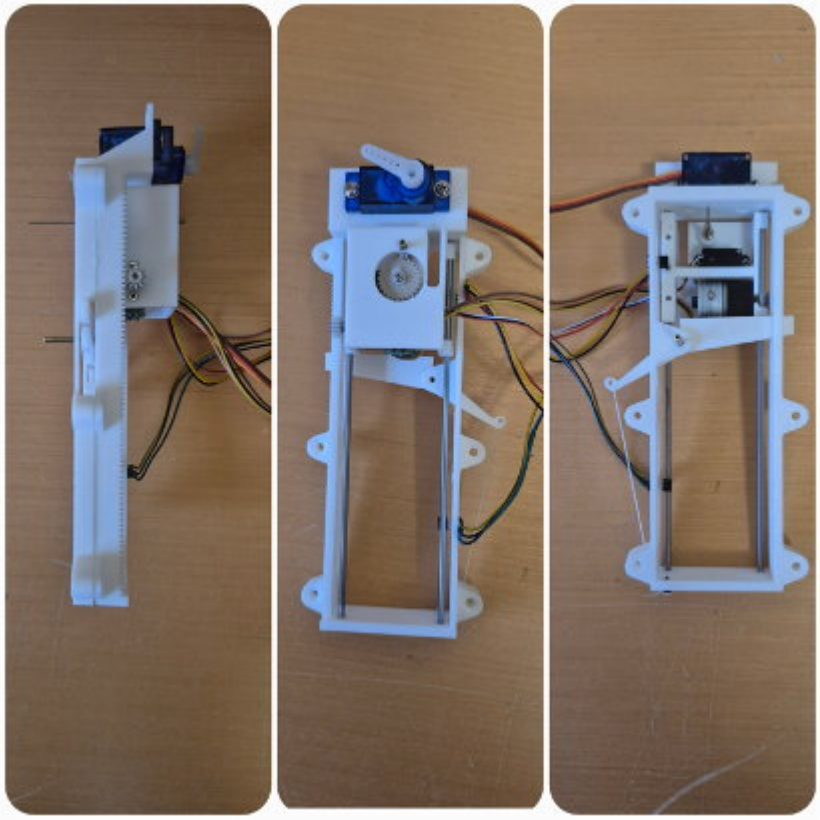
Alle Elemente werden über Grundbausteine der MobaLedLib angesteuert und programmiert. Die Animation kann über einen Taster am Anlagenrand ausgelöst werden.




# Aufbau und Steuerung des UPS-Manns

### Unterkonstruktion





## 3D-Druckdateien:

Eignung für 3D-Drucker: FFF / FDM ★★★★ SLA / STL ★★☆☆

Die Hauptunterkonstruktion besteht aus UK1 und Schlitten. Die UK2 mit Schwenkarm ist eine zusätzliche Komponente. Sie kann mit UK1 zusammen gefügt werden, wenn sich die Gartentür bewegen soll. Die Rückstellung des Schwenkarmes erfolgt dabei über ein Gummi (Stick-Elastic). Die beiden Zahnräder 1 +2 sind für die Bewegung des UPS-Mannes erforderlich. Die Hausgrundplatte besteht (drucktechnisch bedingt) aus zwei Teilen (HGP1 und HGP2). HGP2 ist für die genaue Positionierung der Unterkonstruktion vorgesehen und wird daher mit HGP1 verklebt.

Alle Teile wurden mit einem Bambu Lab P1S geduckt. Damit der Micro-Servo wegen Schwergängigkeit keinen Schaden erleidet, ist es unbedingt erforderlich, die Zahnräder mit einer 0,2mm-Düse zu drucken.

Die Konstruktion zum Betätigen der UPS-Tür wurde dem Türantrieb moderner Bustüren nachempfunden. Sie wird über einen Servo angetrieben und bildet eine separate Einheit (An einer perfekten Ausführung wird noch getüftelt).

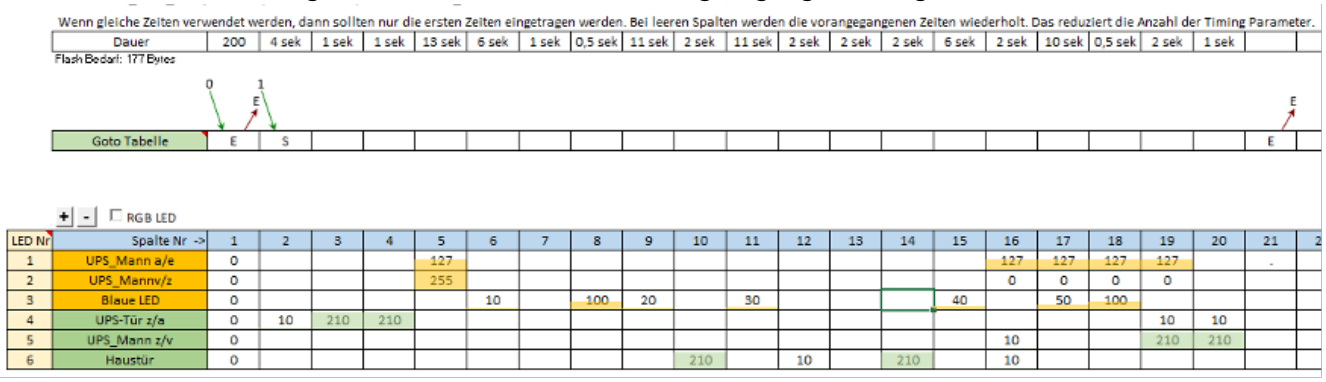
Die Druckdaten sind hier zu finden: https://github.com/Hardi-St/MobaLedLib_Docu/tree/master/3D_Daten_fuer_die_MobaLedLib/Gehaeuse-102/Beta2/

# Ablauf:

Die Bewegung des „UPS-Mann“ wird über einen im Schlitten montierten Steppermotor mit Getriebe angetrieben. Er zieht den Schlitten durch die in der Unterkonstruktion integrierte Zahnstange an die einzelnen Endpositionen. Diese sind durch magnetische Positionssensoren fest fixiert. Als Steuerung kommt die Stepperleiterplatte (550) zum Einsatz. Sie steuert den Stepper und wertet die Informationen der Sensoren aus. Mit Hilfe des Pattern-Configurators wird der zeitliche Ablauf (gelb dargestellt) festgelegt. Die Werte im LED-Kanal 1 und 2 steuern die Bewegungen und Richtung des Steppers. Mit Hilfe von Kanal 3 (blaue LED) werden dem Programm-Generator Werte zu Verfügung gestellt, die eine Beeinflussung des Sounds und der Hausflurbeleuchtung ermöglichen.

An den jeweiligen Endpositionen wird über ein im Schlitten integriertes Micro-Servo die Gehrichtung des UPS-Mannes je nach notwendiger Laufrichtung geändert. Damit die notwendige 180° Kehrtwende ordentlich ausgeführt werden kann, ist die kleine Zahnradkombination notwendig.

Die verbauten Servos werden über die Sevoplatine (510) angesteuert. Der zeitliche Ablauf wird ebenfalls im Pattern-Configurator im LED-Kanal 4-6 festgelegt (grün dargestellt).



Wenn gleiche Zeiten verwendet werden, dann sollten nur die ersten Zeiten eingetragen werden. Bei leeren Spalten werden die vorangegangenen Zeiten wiederholt. Das reduziert die Anzahl der Timing Parameter.

| Dauer | 200 | 4 sek | 1 sek | 1 sek | 13 sek | 6 sek | 1 sek | 0,5 sek | 11 sek | 2 sek | 11 sek | 2 sek | 2 sek | 2 sek | 6 sek | 2 sek | 10 sek | 0,5 sek | 2 sek | 1 sek | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Goto Tabelle | E | S | | | | | | | | | | | | | | | | | | | E | |

Flash Bedarf: 177 Bytes

RGB LED

| LED Nr | Spalte Nr -> | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 2 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | UPS_Mann a/e | 0 | | | | 127 | | | | | | | | | | | 127 | 127 | 127 | 127 | | . | |
| 2 | UPS_Mannv/z | 0 | | | | 255 | | | | | | | | | | | 0 | 0 | 0 | 0 | | | |
| 3 | Blaue LED | 0 | | | | | 10 | | 100 | 20 | | 30 | | | | 40 | | 50 | 100 | | | | |
| 4 | UPS-Tür z/a | 0 | 10 | 210 | 210 | | | | | | | | | | | | | | | 10 | 10 | | |
| 5 | UPS_Mann z/v | 0 | | | | | | | | | | | | | | | 10 | | | 210 | 210 | | |
| 6 | Haustür | 0 | | | | | | | | | 210 | | 10 | | 210 | | 10 | | | | | | |

(Stand 28.02.2025)

Das benötigte Soundmodul wird über eine Servoplatine (510), aber mit entsprechendem Sound-ATTiny bestückt, angesteuert.

Programm-Generator (Stand 28.02.2025):

| Aktiv | Filter | Adresse oder Name | Typ | Start wert | Beschreibung | Verteiler Nummer | Stecker Nummer | Icon | Name | Beleuchtung, Sound, oder andere Effekte | Anz LedNr | LEDs | InCh | Last InCh | Last Sound Kanal |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| ✓ | AE | | | | LED auf dem Mainboard | | | | Heartbeat LED | RGB_Heartbeat(#LED) | 0 | 1 | 0 | 0 | 0 |
| ✓ | AE | | | | | | | | | | | | | | |
| ✓ | AE | SwitchD1 | | | Tageszeilen Tag/Nacht | | | | Taster unbeleuchtet, 2 Funktionen | PushButton_0_2(#InCh, TZ0, 1, 0, 1, 0, 0) | | | 1 | 0 | |
| ✓ | AE | | | | | | | | | | | | | | |
| ✓ | UPS | TZ2 | | | Licht Rückseite | | 1_1 | | Belebtes Haus | House7(#LED, #InCh, 2, 4, 5, 10, NEON_LIGHT, ROOM_WARM_W, RO | 1 | 6 | 1 | 0 | 0 |
| ✓ | UPS | TZ2 | | | Licht Vorderseite | | | | Belebtes Haus | House7(#LED, #InCh, 5, 9, 1, 5, SKIP_ROOM, FIRED, ROOM_TV0, : | 7 | 12 | 1 | 0 | 0 |

Im „Belebten Haus“ der Vorderseite wurde der 1.Raum (Hausflur) als SKIP_ROOM erfasst und somit in der Ansteuerungskette übersprungen. Er wird nachfolgend separat mit Parametern aus UPS_Auto (pc) angesteuert.

| | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| ✓ | UPS | | | | Rücksprung für Licht Eingang Vorderseite | | | | LED Nummer manipulieren | // Next_LED(-12) | 19 | -12 | 0 | 0 | 0 |
| ✓ | UPS | UPS_EG | | | | | | | Toggle Flip-Flop | T_FlipFlopReset(UPS_EGtff, #InCh, SI_0) | | | 1 | 0 | |
| ✓ | UPS | UPS_EGtf | | | Licht Eingang Vorderseite | | | | RGB LED einstellbar | ConstRGB(#LED, #InCh, 0, 0, 0, 127, 127, 127) | 7 | 1 | 1 | 0 | 0 |
| ✓ | UPS | | | | Vorsprung in die laufende Kette | | | | LED Nummer manipulieren | // Next_LED(11) | 8 | 11 | 0 | 0 | 0 |

Damit aber die Hausflur-LED angesteuert werden kann, springt man in der Ansteuerungskette auf diese zurück, führt die notwendige Funktion aus und springt dann wieder vor in die laufende Ansteuerungskette. (die Funktion hier: in Abhängigkeit von Kanal 3 (blaue LED) von UPS_Auto (pc) wird die Hausflur-LED ein.- oder ausgeschaltet.)

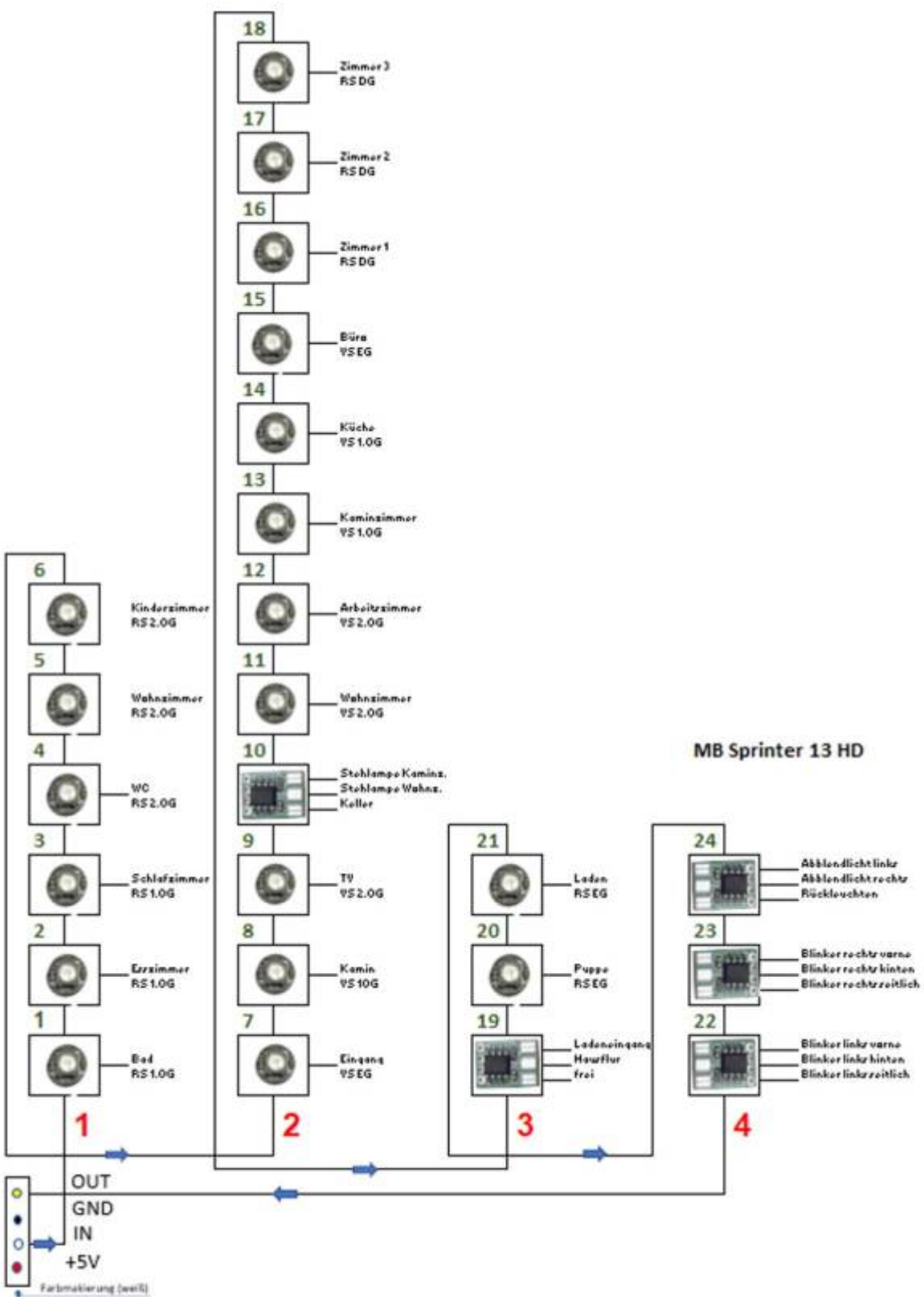
| | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| ✓ | UPS | TZ1 | | | Licht EG Rückseite (Ladeneingang) | | 1_3 | | LED einstellbar | Const(#LED, C1, #InCh, 0, 127) | 19 | C1-1 | 1 | 0 | 0 |
| ✓ | UPS | | | | Licht Rückseite (Hausflur EG) | | 1_3 | | Zufallsschaltung 1 Ausgang | Random(UPS_EG_HF, #InCh, RM_NORMAL, 10 Sek, 50 Sek, 15 Sek, | | | 1 | 0 | |
| ✓ | UPS | UPS_EG_HF | | | Licht Rückseite (Hausflur EG) | | 1_3 | | LED einstellbar | Const(#LED, C2, #InCh, 0, 50) | 19 | C2-2 | 1 | 0 | 0 |
| ✓ | UPS | | | | Licht Rückseite EG (nicht belegt) | | 1_3 | | | | 19 | C3 3 | 1 | 0 | 0 |
| ✓ | UPS | TZ1 | | | Licht Rückseite (Puppe im Laden) | | 1_3 | | RGB-LED einstellbar | ConstRGB(#LED, #InCh, 5, 5, 0, 0, 10, 10) | 20 | 1 | 1 | 0 | 0 |
| ✓ | UPS | TZ1 | | | Licht Rückseite (Laden) | | 1_3 | | RGB-LED einstellbar | ConstRGB(#LED, #InCh, 0, 0, 0, 80, 80, 80) | 21 | 1 | 1 | 0 | 0 |
| ✓ | UPS | | | | | | | | | | | | | | |
| ✓ | UPS | SwitchD3 | | | UPS-Auto-Aktion | | 1_4 | | Taster unbeleuchtet, 1 Funktion | PushButton_0_1(#InCh, UPS0, 1, 0, 1, 0, 80 Sek) | | | 1 | 0 | |
| ✓ | UPS | TZ2 | | | UPS-Auto (Abblendlicht Rückleuchten) | | 1_4 | | LED einstellbar | Const(#LED, C_ALL, #InCh, 0, 127) | 22 | 1 | 1 | 0 | 0 |
| ✓ | UPS | UPS1 | | | UPS-Auto (Blinker rechts) | | 1_4 | | Blinker (Frequenz und Helligkeit) | Blink2(#LED, C_ALL, #InCh, 0.5 Sek, 0.5 Sek, 0, 60) | 23 | 1 | 1 | 0 | 0 |
| ✓ | UPS | UPS1 | | | UPS_Auto (Blinker links) | | 1_4 | | Blinker (Frequenz und Helligkeit) | Blink2(#LED, C_ALL, #InCh, 0.5 Sek, 0.5 Sek, 0, 60) | 24 | 1 | 1 | 0 | 0 |
| ✓ | UPS | | | | | | | | | | | | | | |
| ✓ | UPS | UPS1 | | | UPS_Auto (pc) automatik | | 2+3 | | Muster Pattern Configurator | // Activation: N_Buttons!InCh_to_TmpVar1(#InCh, 1)PatternT20 | 25 | 2 | 1 | 0 | 0 |
| ✓ | UPS | | | | Auswertung wann Flurlicht an und aus geht | | | | LED-Werte als Variable | LED_to_Var(UPS_EG, 2, =, 100) | | | 0 | 0 | |
| ✓ | UPS | | | | Auswertung wann Sound 1 abgespielt wird | | | | LED-Werte als Variable | LED_to_Var(UPS_SO1, 2, =, 10) | | | 0 | 0 | |
| ✓ | UPS | | | | Auswertung wann Sound 2 abgespielt wird | | | | LED-Werte als Variable | LED_to_Var(UPS_SO2, 2, =, 20) | | | 0 | 0 | |
| ✓ | UPS | | | | Auswertung wann Sound 3 abgespielt wird | | | | LED-Werte als Variable | LED_to_Var(UPS_SO3, 2, =, 30) | | | 0 | 0 | |
| ✓ | UPS | | | | Auswertung wann Sound 4 abgespielt wird | | | | LED-Werte als Variable | LED_to_Var(UPS_SO4, 2, =, 40) | | | 0 | 0 | |
| ✓ | UPS | | | | Auswertung wann Sound 5 abgespielt wird | | | | LED-Werte als Variable | LED_to_Var(UPS_SO5, 2, =, 50) | | | 0 | 0 | |

Mit Hilfe der Funktion „LED_to_Var“ wird der Wert des 3. Kanals (blaue LED) der ersten LED im Patter-Generator UPS_Auto (pc) abgefragt und für weitere Auswertungen zwischengespeichert.

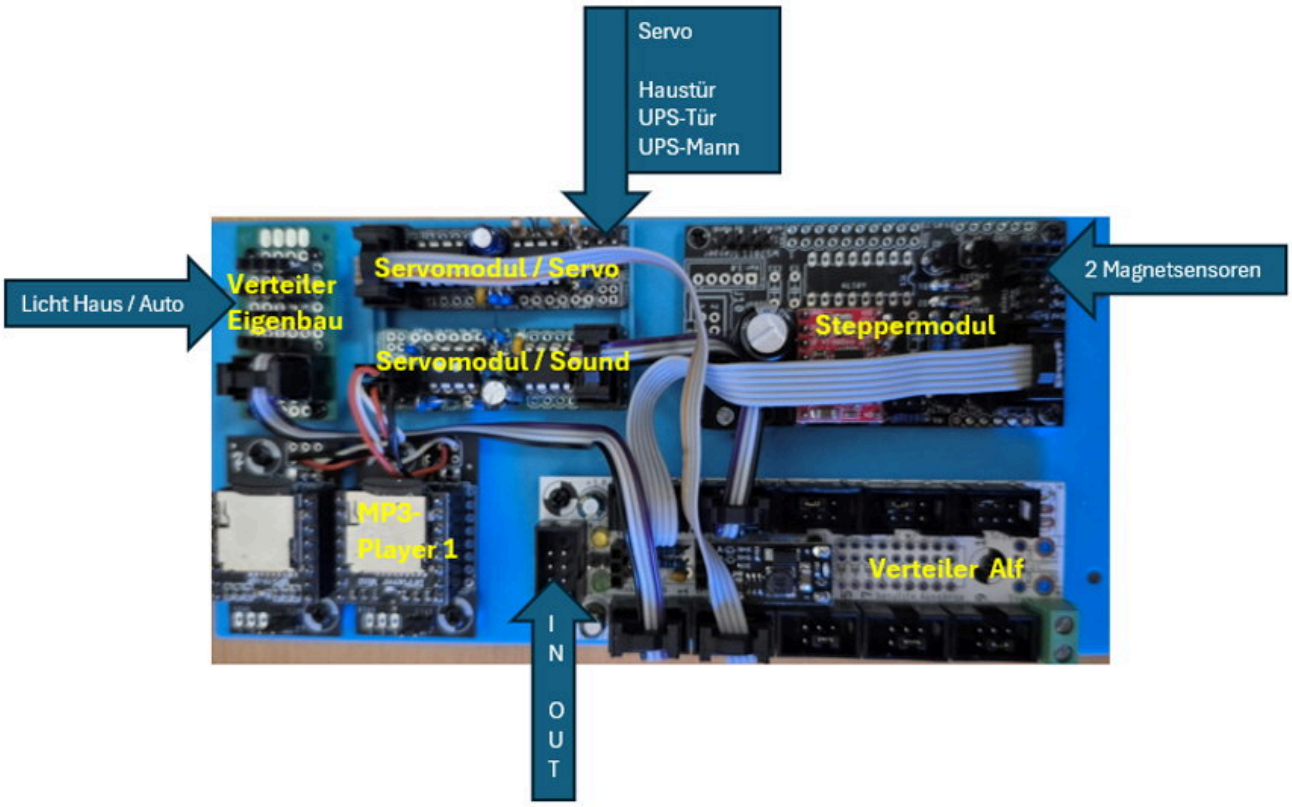
| | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| ✓ | UPS | UPS_SO1 | | | Sound 1 (Klingel / Hund) | | 4 | | Mono-Flop | MonoFlop(UPS_SO1mf, #InCh, 350) | | | 1 | 0 | |
| ✓ | UPS | UPS_SO1mf | | | | | 4 | | Titel # aus Hauptverzeichnis absp | MP3_PLAY_TRACK_ON(#LED, #InCh, 1, 1) | 27 | C1-1 | 1 | 0 | 0 |
| ✓ | UPS | UPS_SO2 | | | Sound 2 (Teppe / Schlüssel) | | 4 | | Mono-Flop | MonoFlop(UPS_SO2mf, #InCh, 350) | | | 1 | 0 | |
| ✓ | UPS | UPS_SO2mf | | | | | 4 | | Titel # aus Hauptverzeichnis absp | MP3_PLAY_TRACK_ON(#LED, #InCh, 1, 2) | 27 | ^C1-1 | 1 | 0 | 0 |
| ✓ | UPS | UPS_SO3 | | | Sound 3 (Begrüßung) | | 4 | | Mono-Flop | MonoFlop(UPS_SO3mf, #InCh, 350) | | | 1 | 0 | |
| ✓ | UPS | UPS_SO3mf | | | | | 4 | | Titel # aus Hauptverzeichnis absp | MP3_PLAY_TRACK_ON(#LED, #InCh, 1, 3) | 27 | ^C1-1 | 1 | 0 | 0 |
| ✓ | UPS | UPS_SO4 | | | Sound 3 (Verabschiedung) | | 4 | | Mono-Flop | MonoFlop(UPS_SO4mf, #InCh, 350) | | | 1 | 0 | |
| ✓ | UPS | UPS_SO4mf | | | | | 4 | | Titel # aus Hauptverzeichnis absp | MP3_PLAY_TRACK_ON(#LED, #InCh, 1, 4) | 27 | ^C1-1 | 1 | 0 | 0 |
| ✓ | UPS | UPS_SO5 | | | Sound 5 (Schlüssel) | | 4 | | Mono-Flop | MonoFlop(UPS_SO5mf, #InCh, 350) | | | 1 | 0 | |
| ✓ | UPS | UPS_SO5mf | | | | | 4 | | Titel # aus Hauptverzeichnis absp | MP3_PLAY_TRACK_ON(#LED, #InCh, 1, 5) | 27 | ^C1-1 | 1 | 0 | 0 |
| ✓ | AE | | | | | | | | | | | | | | |
| ✓ | AE | | | | LED auf dem Mainboard | | | | Heartbeat LED | RGB_Heartbeat(#LED) | 28 | 1 | 0 | 0 | 0 |

Der erforderliche Sound wird in Abhängigkeit von dem gespeicherten Wert aus „LED_to_Var“ nach setzen eins Mono-Flops abgespielt.

# Verdrahtung im Haus:




# Verwendete MobaLedLib Baugruppen und Teile:

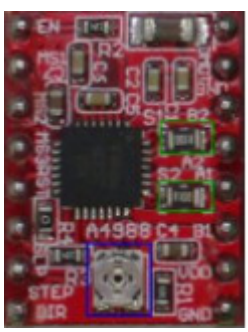



## Stepper mit Getriebe 1:63 1260 Impulse für 360° (Aliexpress)




- Zahnrad 7mm
- 12 Zähne Modul 0,5
- Motorschrittwinkel 18° (20 Steps)
- Spulenwiderstand 47,8 ohm
- maximaler Strom 0,124 A (6V)
- Die Montageplatte wurde demontiert

## Treiber A4988-Modul



## Servo SG90




- Torque: 1.5kg/cm(4.8V)
- Geschwindigkeit 0.3sec/60 Grad(4.8v)
- Betriebsspannung 4.8V
- Größe: 23 x 12.2 x 29mm
- Gewicht: 9g

## Micro-Servo

- Spannungsbereich: 3,5...6 V-
- Stellkraft: 0,25 kg/cm²
- Geschwindigkeit: 0,15 s/60°
- Gewicht: 1,8 g
- Maße (LxBxH): 16,6×13,5×6 mm

From:
https://wiki.mobaledlib.de/ - **MobaLedLib Wiki**

Permanent link:
**https://wiki.mobaledlib.de/anleitungen/spezial/codevorlagen/ups-zustellung?rev=1741976988**

Last update: **2025/03/14 18:29**