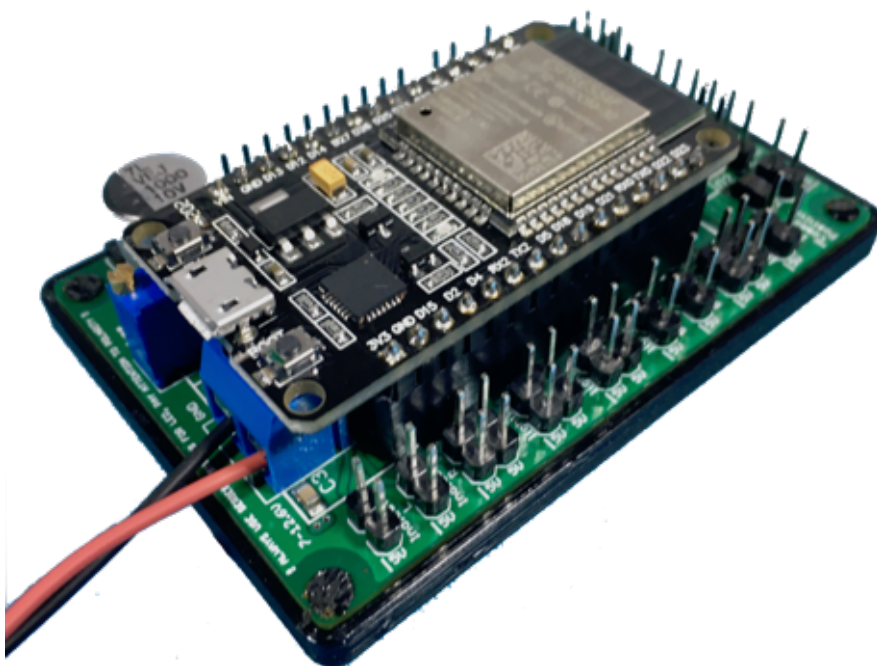




ESP32 RC Sound- und Lichtsteuerung

Schnellstartanleitung



Inhaltsverzeichnis

<u>SICHERHEITSBESTIMMUNGEN</u>	<u>2</u>
<u>TECHNISCHE DATEN</u>	<u>3</u>
<u>PRODUKTBESCHREIBUNG</u>	<u>4</u>
<u>SYSTEMÜBERSICHT</u>	<u>5</u>
<u>MINIMUMVERKABELUNG (SBUS-MODUS)</u>	<u>7</u>
<u>LED-Verkabelung</u>	<u>8</u>
<u>KANALZUWEISUNG (VERSION IM STILVOR-MODUS „MICRO RC“)</u>	<u>9</u>
<u>SOFTWAREKONFIGURATION</u>	<u>10</u>
<u>LINKS</u>	<u>11</u>



ESP32 RC Sound- und Lichtsteuerung

Schnellstartanleitung

Sicherheitsvorschriften



- **Dieses Produkt darf erst verwendet werden, nachdem diese Bedienungsanleitung vollständig gelesen und verstanden wurde. Ebenso müssen Sie allen in diesem Kapitel genannten Einschränkungen zustimmen.**Die
- Dieses Open-Source-Produkt unterliegt der alleinigen Verantwortung des Nutzers hinsichtlich Nutzung, Weiterentwicklung und Anpassung. TheDIYGuy999 übernimmt keinerlei Haftung für Schäden oder Folgeschäden, die durch die Nutzung dieses Systems entstehen.
- Dieses Produkt ist für große und gefährliche Modelle nicht geeignet, da keine Redundanz vorhanden ist.
- Dieses Produkt darf niemals auf öffentlichen Straßen verwendet werden.
- Dieses Produkt wurde nicht auf FCC- oder CE-Konformität geprüft.
- Schließen Sie niemals Kabel an oder trennen Sie sie, während das Produkt an eine Batterie angeschlossen ist.
- Trennen Sie den Akku immer ab, wenn dieses Produkt nicht verwendet wird. Es verbraucht stets einige Milliampere, selbst wenn der Regler ausgeschaltet ist, und entlädt den Akku andernfalls.
- Dieses Produkt ist nicht gegen Verpolung geschützt! Überprüfen Sie die Verkabelung unbedingt sorgfältig, bevor Sie die Batterie anschließen. Verpolung zerstört das Produkt sofort.
- Isolieren Sie die Fahrzeugverkabelung stets ordnungsgemäß mit Schrumpfschlauch, um Kurzschlüsse zu vermeiden. Montieren Sie die blanke Leiterplatte niemals direkt auf einer Metallplatte. Dies führt zu Kurzschlüssen.
- Dieses Produkt darf nur in trockenen Umgebungen verwendet werden.
- Dieses Produkt ist nicht für Kinder unter 14 Jahren geeignet.



ESP32 RC Sound- und Lichtsteuerung

Schnellstartanleitung

Technische Spezifikationen

Allgemein:	
Open Source:	Ja (Software & Hardware): https://github.com/TheDIYGuy999/Rc_Engine_Sound_ESP32
Größe (SMD-Version):	74 x 50 x 25 mm (einschließlich Anschlüsse und Bodengehäuse)
Spannungsbereiche:	
Batteriespannung:	Der Eingangsspannungsbereich beträgt 7 – 12,6 VDC (2S oder 3S LiPo).
Maximales „Sig“ <small>Stromspannung:</small>	3,3 VDC (Die meisten modernen Empfänger arbeiten mit dieser Logikspannung, bei sehr alten Empfängern ist jedoch Vorsicht geboten)
Maximale „+V“-Spannung von ESC	6,5 VDC, abhängig auch von den Einschränkungen der anderen Geräte, die an die „+V“-Schiene angeschlossen sind.
Ausgangsspannung:	5 VDC für LED, Shaker und Verstärker, maximal 1 A insgesamt
Eingangssignaltypen:	
PWM:	6 Kanäle, Anschlüsse CH1 – CH6
PPM:	8 Kanäle, Anschluss RX
SBUS:	13 Kanäle, Anschluss RX
IBUS:	13 Kanäle, Anschluss RX
Ausgaben:	
Lautsprecherausgänge:	Es können 1 oder 2 Lautsprecher mit 4 – 8 Ohm angeschlossen werden.
5VDC-Ausgänge für LEDs:	11 Kanäle, alle mit PWM-Helligkeitssteuerung, gemeinsamer Pluspol, immer mit Serienwiderstand verwenden, max. 100 mA pro Kanal
5VDC-Motorvibrationen	1 Ausgang, max. 300 mA
Simulationsschüttler aus:	
ESC:	Der Regler (Crawler-Ausführung mit Direktbremse) muss an den „ESC“-Ausgang angeschlossen werden. Dadurch kann die „virtuelle Fahrzeugträgheit“ genutzt werden.
Kompatible Fernbedienungen:	
Arduino „Micro RC“:	Für die volle Funktionalität ist ein bestimmter Fahrzeugtyp erforderlich.
Flysky FS-I6x:	SBUS-Modus empfohlen, Dual-Rate-Umschaltung erforderlich 100 % / 75 %
Andere:	Definieren Sie Ihr eigenes Profil auf der Registerkarte „2_adjustmentsRemote“.
Geräusche:	
Audiodateityp:	.h-Dateien, 8 Bit, 22050 Hz (variable Abtastrate für die Engine, generiert aus WAV-Dateien mit dem mitgelieferten Tool „Audio2Header.html“)
Klangkategorien:	Starten, Leerlauf, Gasgeben, Zündklopfen, Druckluftbremse, Feststellbremse, Motorbremse, Hupe, Turbolader, Blow-off-Ventil, Hupe, Sirene, Geräusch 1
Übertragungen:	
Schaltgetriebe:	TAMIYA 3-Gang, virtueller 3-Gang (geschaltet durch 3-Positionen-Schalter)
Automatisch	Virtuelles Automatikgetriebe mit Drehmomentwandler, virtuelles
Übertragungen:	Doppelkupplungsgetriebe, jeweils mit 3, 4 oder 6 Gängen
Vordefinierte Fahrzeuge:	
Viele Fahrzeuge sind bereits konfiguriert:	Lkw, Panzer, Pkw, Geländewagen, Traktoren, Motorräder, Flugzeuge, Lokomotiven aus der EU, Russland und den USA... Selbstverständlich können Sie auch Ihre eigenen Fahrzeugkonfigurationen erstellen.



ESP32 RC Sound- und Lichtsteuerung

Schnellstartanleitung

Konfiguration:

Erforderliches USB-Kabel:	Micro-USB
Hochladen und Konfigurieren:	Die Anpassungen werden in der Arduino IDE, Mindestversion 1.7.4, vorgenommen.
Eingangskanal automatisch	Ja, nach dem Einschalten, falls die Option im Tab „2_adjustmentsRemote“ aktiviert ist.
Nullpunktkalibrierung:	
Kanalumkehr	Ja, wenn im Tab „2_adjustmentsRemote“ aktiviert.
Flexibler Kanal	Ja, gemäß den Einstellungen im Tab „2_adjustmentsRemote“.
Abbildung:	

Produktbeschreibung

Das **Open-Source-RC-Sound- und Lichtsteuerung** Es ist hauptsächlich für ferngesteuerte Trucks im Maßstab 1:14 wie den TAMIYA King Hauler gedacht. Natürlich kann es auch für alle anderen Fahrzeuge verwendet werden, sofern diese hineinpassen. Es ist ein guter Ersatz für den TAMIYA MFC-01 oder MFC-03 und bietet deutlich mehr Realismus.

Viele Fahrzeuges sind vorkonfiguriert und können **leicht auswählbar** Selbstverständlich können Sie auch Ihre eigenen Konfigurationen erstellen, ohne den Arduino-Hauptcode zu verändern.

Es kann zur Steuerung von Fahrzeuggeräuschen wie Motorengeräuschen, Hupen, Sirenen, Bremsgeräuschen usw. verwendet werden. **Mehrere Klänge** können gleichzeitig abgespielt werden. Der Motorensound ist **spontan vermischt**, wobei verschiedene Geräusche wie Leerlauf, Gasgeben, Turbo, Diesellopfen usw. verwendet werden. Die Lautstärke des Motorgeräuschs beträgt **last- und drehzahlabhängig** Dadurch klingt der Motor sehr realistisch.

Es bietet außerdem einzigartige Funktionen wie **virtuelle Fahrzeugträgheit**, synchronisiertes TAMIYA 3-Gang-Getriebe (in Software), **virtuelle manuelle und automatische Getriebe** All diese Eigenschaften sorgen für ein sehr geschmeidiges und realistisches Fahrverhalten.

Selbstverständlich ist es auch in der Lage, das gesamte System zu kontrollieren. **Fahrzeug- und Anhängerbeleuchtung** Die Helligkeit aller LEDs kann mittels Pulsweitenmodulation (PWM) variiert werden. Dies ermöglicht die Simulation des enormen Stromverbrauchs beim Anlassen des Motors. **sanftes Schalten der Glühbirne** Blinker, Fern- und Abblendlicht, **Xenon** Zündblitz usw.

A TAMIYA **Anhängerbeleuchtung** Satz 56502 im Lagerzustand kann angeschlossen werden.

A **Schüttelausgang** ist ebenfalls enthalten. Es dient zum Antrieb eines Rüttelmotors mit einem Exzentergewicht zur Simulation von Motorschwingungen.

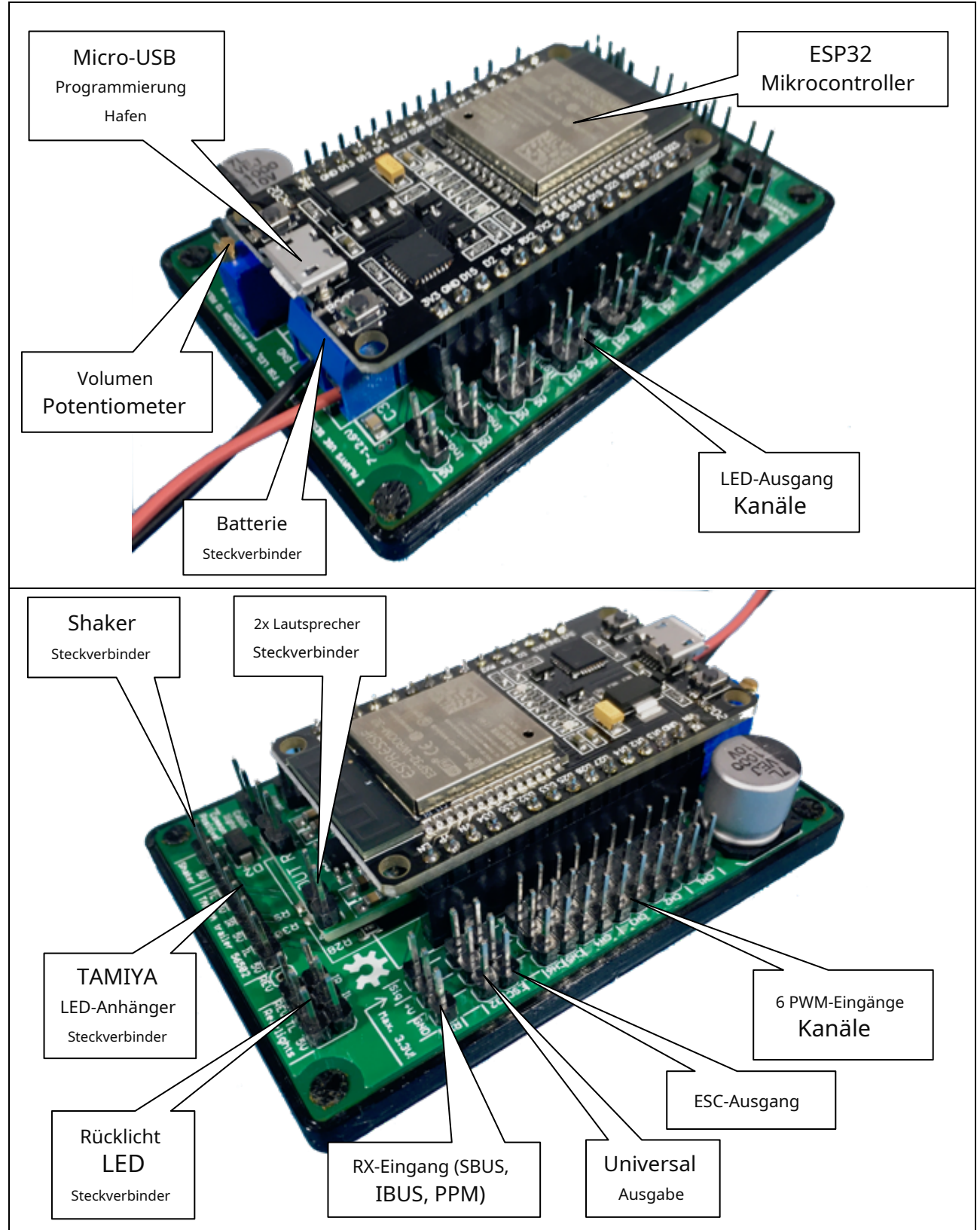
Das Modul ist **kompatibel** bei den meisten Fernsystemen, die verwenden **PWM, PPM, SBUS oder IBUS** Kommunikation. Es ist eine flexible Kanaluordnung implementiert. Details finden Sie in der beigefügten Datei „adjustmentsRemote.xlsx“.



ESP32 RC Sound- und Lichtsteuerung

Schnellstartanleitung

Systemübersicht





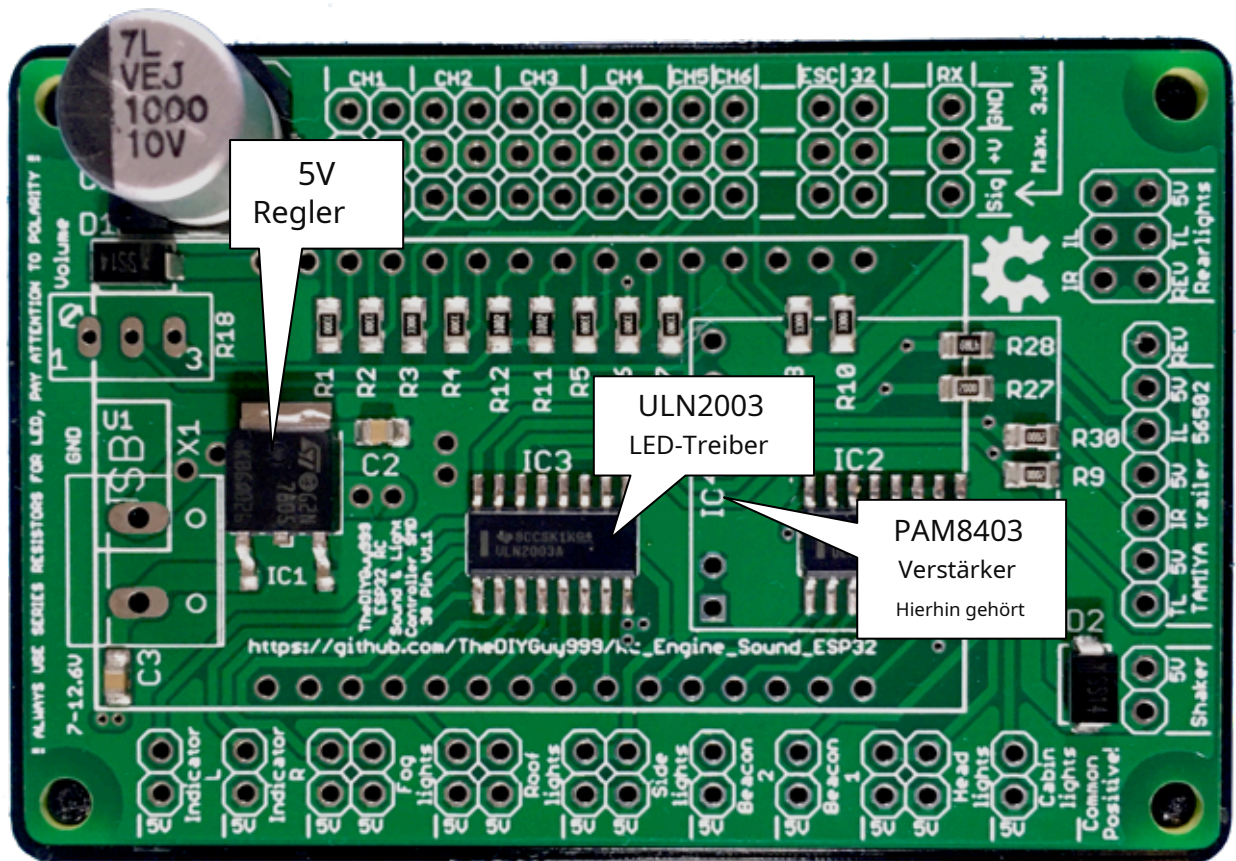
ESP32 RC Sound- und Lichtsteuerung

Schnellstartanleitung

Diese Ansicht zeigt eine unfertige Platine, wie sie von PCBWay.com geliefert wird. So bestellen Sie sie: https://github.com/TheDIYGuy999/Rc_Engine_Sound_ESP32/tree/master/Eagle_PCB/SM

Nur die SMD-Bauteile sind bestückt, die restlichen bedrahteten Bauteile müssen noch bestückt werden.
Selbst löten. Anleitungsvideo ansehen: <https://www.youtube.com/watch?v=csQgTfxRd8Y&t=1s>

Es zeigt auch die Steckerbelegung im Detail. CH1 – 4 sind Steckerpaare, die eliminieren die Notwendigkeit von Y-Kabeln.



Beachten Sie, dass die LEDs verkabelt werden müssen. **häufig positiv** Das bedeutet, dass die langen LED-Beine Die positive Seite kann mit einem Kabel verbunden werden. **gemeinsame 5V** Potenzial. Die negative Seite jeder LED (der kurze Draht) muss an einen **geeigneter Strombegrenzungswiderstand** Und dann zum LED-Ausgangspin.

Wenn ein "TAMIYA 56502 Anhängerbeleuchtungsset" Ist das Bauteil mit dem entsprechenden Anschluss verbunden, sind keine zusätzlichen Strombegrenzungswiderstände erforderlich. Diese sind bereits auf der Leiterplatte integriert.

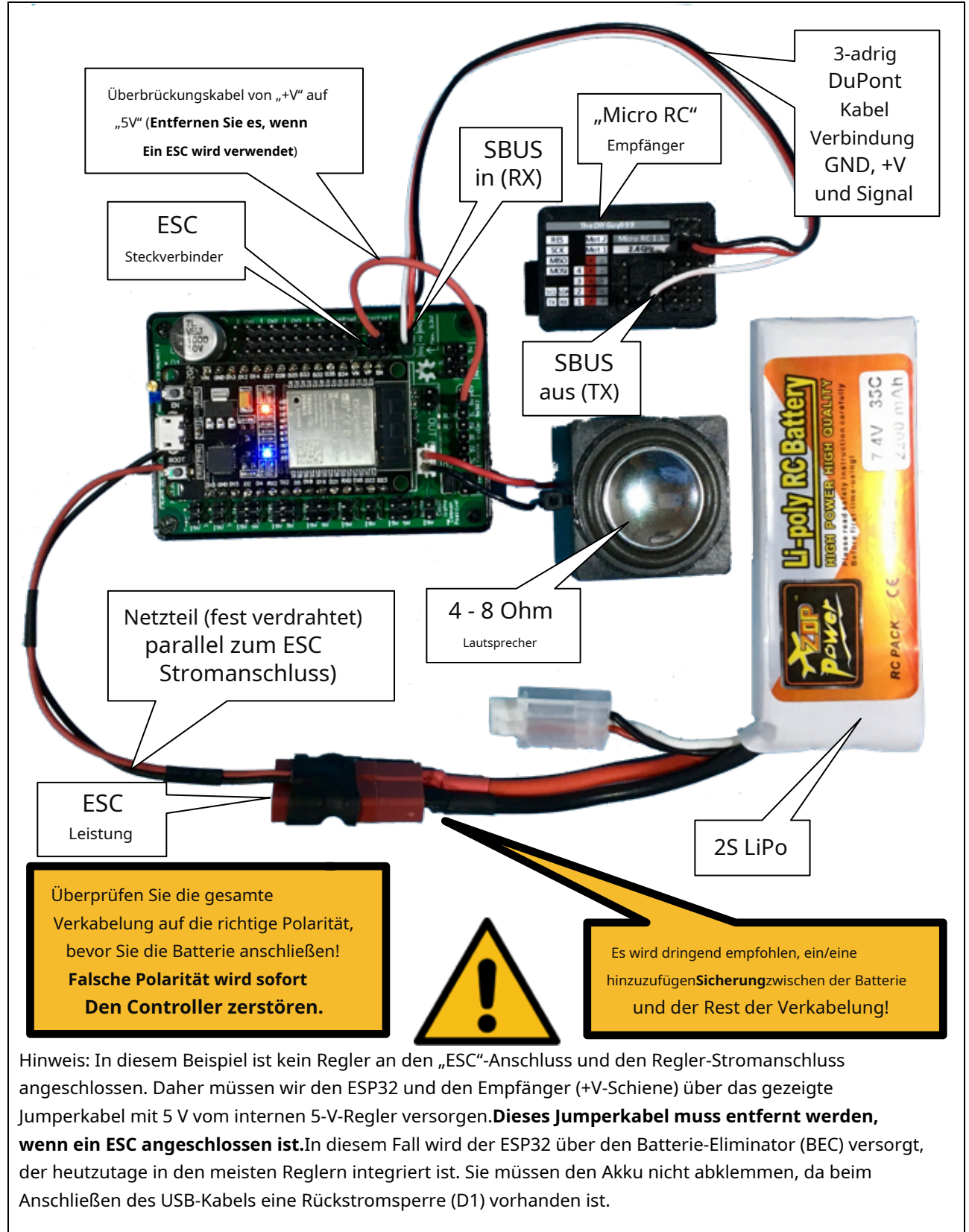
Weitere Einzelheiten finden Sie im Kapitel „LED-Verkabelung,“



ESP32 RC Sound- und Lichtsteuerung

Schnellstartanleitung

Minimale Verkabelung (SBUS-Modus)





ESP32 RC Sound- und Lichtsteuerung

Schnellstartanleitung

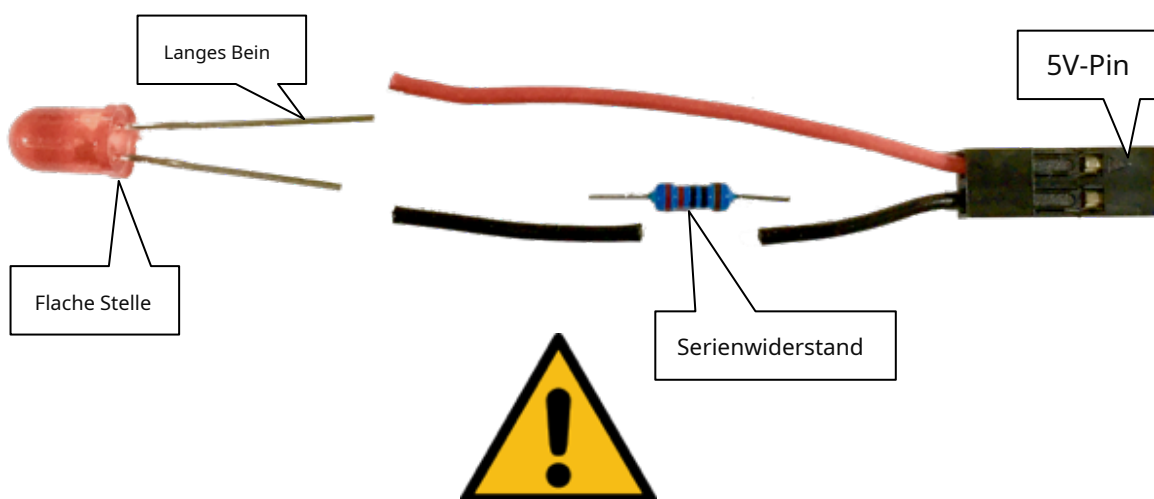
LED-Verkabelung

Wie bereits erwähnt, beträgt die LED-Versorgungsspannung **5V**, das vom internen Regler kommt. Ein üblicher maximaler LED-Strom beträgt **15 mA**. Die

Die folgende Tabelle enthält die erforderlichen Mindestwerte der Serienwiderstände:

LED-Farbe	Durchlassspannung [Volt]	Minimaler Widerstand [Ohm]
Rot	1.8	220
orange	1.9	220
Grün	2	200
Weiß	3	150
Blau	3	150

Sie können auch einen Taschenrechner verwenden, um den korrekten Wert zu ermitteln: <https://www.digikey.de/en/resources/conversion-calculators/conversion-calculator-ledseries-resistor>



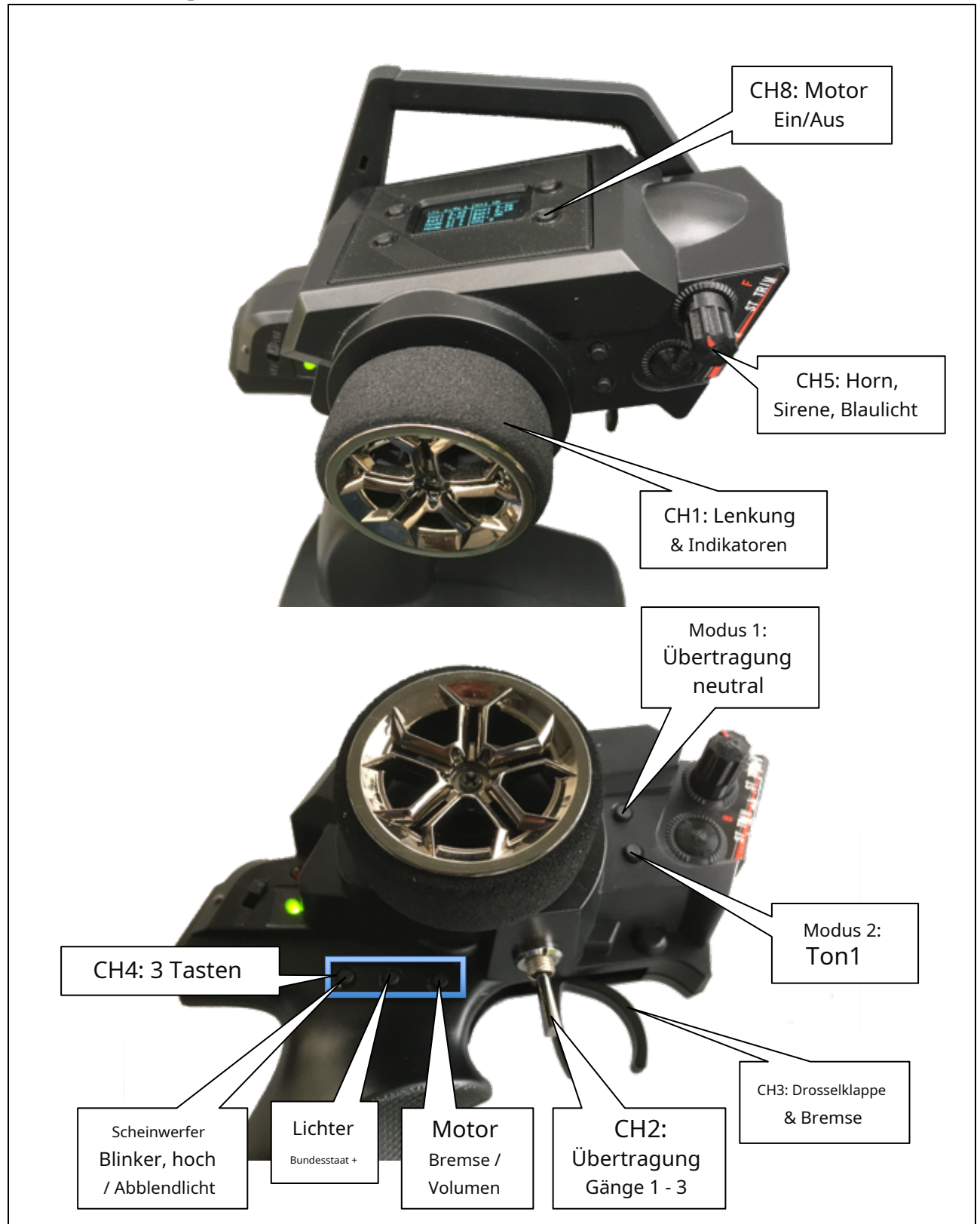
- Die LED wird **nicht aufleuchten**, wenn verbunden **Falscher Weg** um.
- **Nie verbinden** LED oder andere Geräte **ohne** ein geeignetes **Serienwiderstand** an die LED-Ausgangspins (außer dem TAMIYA-Anhängerstecker). Dies wird **dauerhafte Schäden** die integrierten ULN2003 LED-Treiber.
- Wenn mehr als eine LED parallel an einen LED-Ausgangspin angeschlossen ist, benötigt jede ihren eigenen Serienwiderstand.
- Zwei rote oder orange LEDs können in Reihe geschaltet werden. Dadurch wird der Stromkreis hergestellt. **effizienter** In diesem Fall müssen Sie die beiden Serienspannungen für die Berechnung des Serienwiderstands addieren.
- Der **maximale durchschnittliche LED-Ausgangsstrom pro Kanal beträgt 100 mA**. Es wird jedoch nicht empfohlen, jeden Kanal mit diesem Strom zu belasten. Andernfalls könnte der ULN2003-LED-Treiber beschädigt werden. **überhitzen**. Stets **Überprüfe es** Bei übermäßiger Temperaturerhöhung sollten Sie nach dem Anschließen eines neuen Fahrzeugs mindestens 10 Minuten warten. Wenn die Temperatur zu hoch ist, um sie zu berühren, verwenden Sie einen externen Treibertransistor für die LED.
- Beachten Sie außerdem, dass **maximale Summe Strom** für die gesamte 5V-Schiene, die vom Bordregler versorgt wird, ist **1A**. Die



ESP32 RC Sound- und Lichtsteuerung

Schnellstartanleitung

Kanalzuordnung (Version im Stil von „Micro RC“-Autos)





ESP32 RC Sound- und Lichtsteuerung

Schnellstartanleitung

Softwarekonfiguration

Für weitere Details siehe:

https://github.com/TheDIYGuy999/Rc_Engine_Sound_ESP32/blob/master/README.md

Nehmen Sie die Anpassungen in den Registerkarten „Anpassungen“ vor. **Ändern Sie den Code im Haupt-Tab nicht.** Die

The screenshot shows the Arduino IDE with the project 'Rc_Engine_Sound_ESP32' open. The main code editor displays the following code:

```

1 // RC engine sound & LED controller for Arduino ESP32. Written by TheDIYGuy999
2 // Based on the code for ATmega 328: https://github.com/TheDIYGuy999/Rc_Engine_Sound
3
4 // ***** ESP32 CPU frequency must be set to 240MHz! *****
5
6 // Sound files converted with: https://github.com/TheDIYGuy999/Rc_Engine_Sound_ESP32/blob/master/Audio2Header.html
7 // Original converter code by bitluni (send him a high five, if you like the code)
8
9 // Parts of automatic transmission code from Wombii's fork: https://github.com/Wombii/Rc_Engine_Sound_ESP32
10
11 //
12 const float codeVersion = 6.2; // Software revision.
13
14 //
15 // =====
16 // !! IMPORTANT !! SETTINGS (ADJUST THEM BEFORE CODE UPLOAD), REQUIRED ESP32 BOARD DEFINITION
17 // =====
18 //
19 // All the required user settings are done in the following .h files:
20
21 #include "1_adjustmentsVehicle.h" // <----- Select the vehicle you want to simulate
22 #include "2_adjustmentsRemote.h" // <----- Remote control system related adjustments
23 #include "3_adjustmentsESC.h" // <----- ESC related adjustments
24 #include "4_adjustmentsTransmission.h" // <----- Transmission related adjustments
25 #include "5_adjustmentsShaker.h" // <----- Shaker related adjustments
26
27 // Install ESP32 board according to: https://randomnerdtutorials.com/installing-the-esp32-board-in-arduino-ide-windows-instruction/
28 // Adjust board settings according to: https://github.com/TheDIYGuy999/Rc_Engine_Sound_ESP32/blob/master/Board%20settings.png
29
30 // Make sure to remove -master from your sketch folder name
31
32 // DEBUG options can slow down the playback loop! Only uncomment them for debugging, may slow down your simulation
33 // #define CHANNEL_DEBUG // uncomment it for input signal debugging info
34 // #define ESC_DEBUG // uncomment it to debug the ESC
35 // #define AUTO_TRANS_DEBUG // uncomment it to debug the automatic transmission
36 // #define MANUAL_TRANS_DEBUG // uncomment it to debug the manual transmission
37 // #define TRACKED_DEBUG // debugging tracked vehicle mode
38
39 // TODO = Things to clean up!
40
41 //
42 // =====
43 // LIBRARIES & HEADER FILES
44 // =====
45 //
46 // Header files
47 #include "headers/curves.h" // Nonlinear throttle curve arrays
48
49 // Libraries (you have to install all of them in the "Arduino sketchbook"/libraries folder)
50 #include <statusLED.h> // https://github.com/TheDIYGuy999/statusLED <----- Install the newest version!
51 #include <SBUS.h> // https://github.com/TheDIYGuy999/SBUS you need to install my fork of this library!
52 #include <rcTrigger.h> // https://github.com/TheDIYGuy999/rcTrigger <----- v4.7: This one is now required as well
53 #include <IBusBM.h> // https://github.com/bmellink/IBusBM required for IBUS interface
54
55

```

Callouts and annotations:

- Hauptregisterkarte**: Points to the main code editor tab.
- Fahrzeug**: Points to the '1_adjustmentsVehicle.h' tab.
- Fernbedienung**: Points to the '2_adjustmentsRemote.h' tab.
- ESC**: Points to the '3_adjustmentsESC.h' tab.
- Übertragung**: Points to the '4_adjustmentsTransmission.h' tab.
- Shaker**: Points to the '5_adjustmentsShaker.h' tab.
- Softwareversion**: Points to the line `const float codeVersion = 6.2; // Software revision.`
- Die Anpassungen**: Points to the include statements for the adjustment files. Sub-note: **Registerkarten oben** (Tabs above).
- Erforderlich Board-Einstellungen**: Points to the board definition section. Sub-note: **Platine installieren Definition** (Install board definition).
- Installieren Sie diese Bibliotheken**: Points to the library include statements.



ESP32 RC Sound- und Lichtsteuerung

Schnellstartanleitung

Links

Software- und Hardware-Downloads: https://github.com/TheDIYGuy999/Rc_Engine_Sound_ESP32

Readme:

https://github.com/TheDIYGuy999/Rc_Engine_Sound_ESP32/blob/master/README.md

Videos:

<https://www.youtube.com/channel/UCqWO3PNCsJHmYiACDMLr23w>

Forum:

<https://www.rc-modellbau-portal.de/index.php?threads/esp32-arduino-rc-sound-und-lichtcontroller.7183/>



Entworfen in der Schweiz von TheDIYGuy999