ESP32 RC Sound & Light Controller

Quick Start Manual

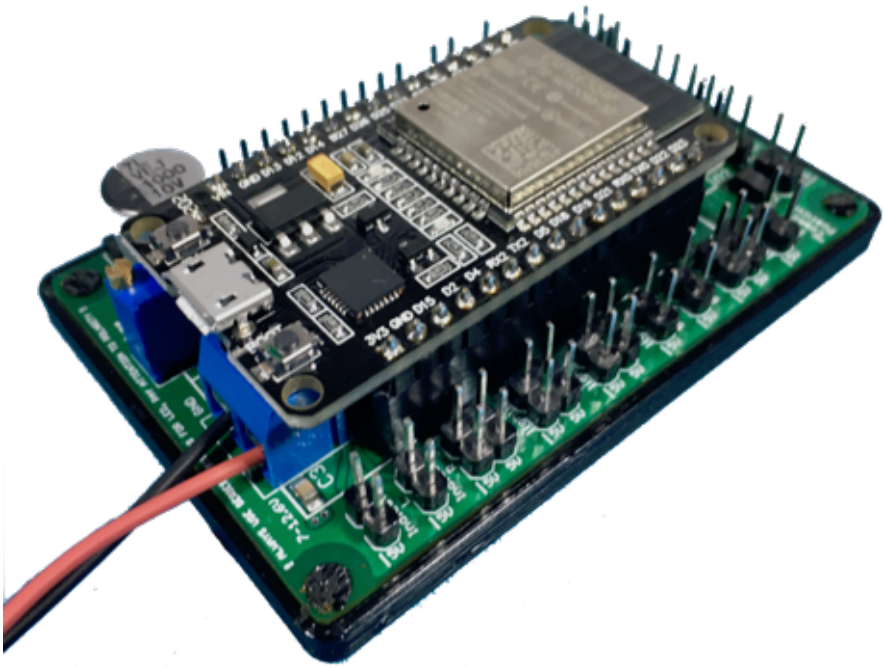


Table of Contents

ESP32 RC Sound & Light Controller

Quick Start Manual

Safety regulations

- **This product must not be used until this manual has been fully read and understood. Likewise, you must agree with all the restrictions mentioned in this chapter**.
- This open source product is left to the user's own responsibility for use, free development and customization. TheDIYGuy999 assumes no liability whatsoever for any damage or consequential damage resulting from the use of this system.
- This product is not suitable for large and dangerous models, because there is no redundancy.
- Never use this product on public roads.
- This product has not been tested for FCC or CE compliance.
- Never connect or disconnect wires while the product is connected to a battery.
- Always disconnect the battery, if this product is not in use. It always draws a couple of Milliamps, even if the ESC is switched off and will drain the battery otherwise.
- This product is not protected against wrong polarity! Always double check your wiring before connecting the battery. Wrong polarity will immediately destroy the product.
- Always insulate the vehicle wiring properly, using heat shrinks in order to prevent short circuits. Never mount the bare PCB directly on top of a metal plate. This will cause short circuits.
- This product may only be used in dry locations.
- This product is not suitable for children below 14 years.

ESP32 RC Sound & Light Controller

Quick Start Manual

Technical specifications

General:	
Open Source:	Yes (Software & Hardware): https://github.com/TheDIYGuy999/Rc_Engine_Sound_ESP32
Size (SMD version):	74x50x25mm (including connectors and bottom shell)
Voltage ranges:	
Battery supply voltage:	The input supply voltage range is 7 – 12.6VDC (2S or 3S LiPo)
Maximum "Sig" voltage:	3.3VDC (most modern receivers are working with this logic voltage, but you have to be careful with very old receivers)
Maximum "+V" voltage coming from ESC	6.5VDC, also depending on the limitations of the other devices, which are connected to the "+V" rail
Output voltage:	5VDC for LED, shaker and amplifier, **max. 1A in total**
Input signal types:	
PWM:	6 channels, connectors CH1 – CH6
PPM:	8 channels, connector RX
SBUS:	13 channels, connector RX
IBUS:	13 channels, connector RX
Outputs:	
Speaker outputs:	1 or 2 speakers with 4 – 8 Ohms can be connected
5VDC outputs for LED:	11 channels, all with PWM brightness control, common positive, always use a series resistor, **max. 100mA** per channel
5VDC engine vibration simulation shaker out:	1 output, **max. 300mA**
ESC:	The ESC (crawler style with direct brake) needs to be connected to the "ESC" output. This allows to use the "virtual vehicle inertia"
Compatible remotes:	(predefined profiles in the "2_adjustmentsRemote" tab)
Arduino "Micro RC":	Car style type required for full functionality
Flysky FS-I6x:	SBUS mode recommended, **dual rate switching required** 100% / 75%
Others:	Define your own profile in the "2_adjustmentsRemote" tab
Sounds:	
Sound file type:	.h files, 8bit, 22050Hz (variable sampling rate for engine, generated out of wav files using the included tool "Audio2Header.html")
Sound categories:	Start, idle, revving, ignition knock, air brake, parking brake, engine brake, horn, turbo, blow-off valve, horn, siren, sound 1
Transmissions:	
Manual transmissions:	TAMIYA 3 speed, virtual 3 speed (shifted by 3 pos. switch)
Automatic transmissions:	Virtual automatic with torque converter, virtual double clutch, each with 3, 4 or 6 gears
Predefined vehicles:	(select them in the "1_adjustmentsVehicle" tab)
A lot of vehicles are already configured:	EU, Russian & US trucks, tanks, EU & US cars, EU & US SUV, tractors, motorcycles, planes, locomotives… Of course, you can also make your own vehicle configurations

ESP32 RC Sound & Light Controller

Quick Start Manual

Configuration:	
Required USB lead:	Micro USB
Upload and config:	Adjustments are done in Arduino IDE, min. version 1.7.4
Input channel auto zero calibration:	Yes, after power-on, if enabled in "2_adjustmentsRemote" tab
Channel reversing	Yes, if enabled in "2_adjustmentsRemote" tab
Flexible channel mapping:	Yes, according to settings in "2_adjustmentsRemote" tab

Product description

This **open source RC sound and light controller** is mainly intended for 1/14 scale RC trucks like the TAMIYA King Hauler. Of course, it can also be used for all kinds of other vehicles, as long as it fits inside. It is a good replacement for the TAMIYA MFC-01 or MFC-03 and offers way more realism.

A lot of vehicles are pre-configured and can **easily be selected**. Of course, you can also make your own configurations without changing the Arduino main code.

It can be used to control the vehicle sounds like engine sounds, horns, sirens, brake sounds etc. **Multiple sounds** can be played at the same time. The engine sound is **mixed together on the fly**, using multiple sounds like idle, revving, turbo, Diesel knock etc. The engine sound volume is **load- and RPM-dependent**. This makes the engine sound very realistic.

It also offers unique features like **virtual vehicle inertia**, synchronized TAMIYA 3 speed transmission (in software), **virtual manual and automatic transmissions**. All these features make the vehicle drive behavior very smooth and realistic.

Of course, it also is able to control the entire **vehicle and trailer lighting**. All the LED can be varied in brightness, using PWM (Pulse Width Modulation). This allows to simulate the huge current draw while cranking the engine, **smooth switching incandescent bulb** indicators, high & low beam, **Xenon** ignition flash etc.

A TAMIYA **trailer light** set 56502 in stock condition can be connected.

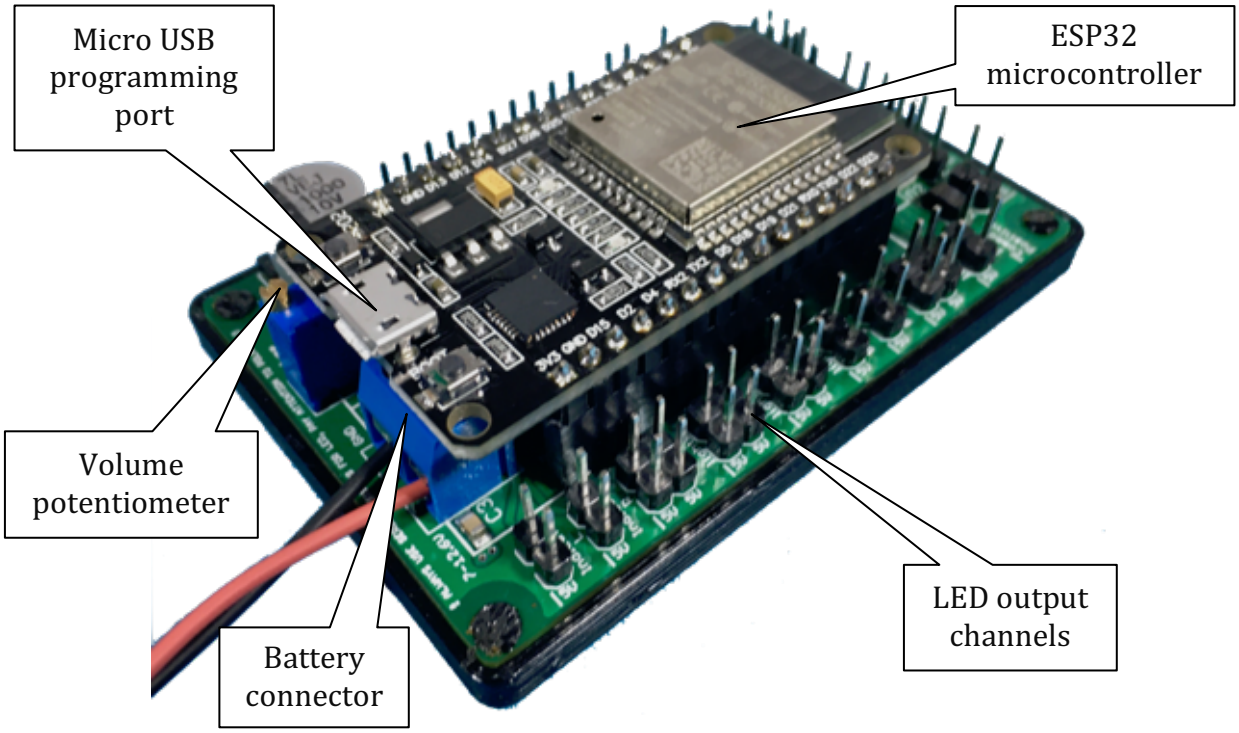
A **shaker output** is included as well. It is used to drive a shaker motor with an eccentric weight to simulate engine vibrations.

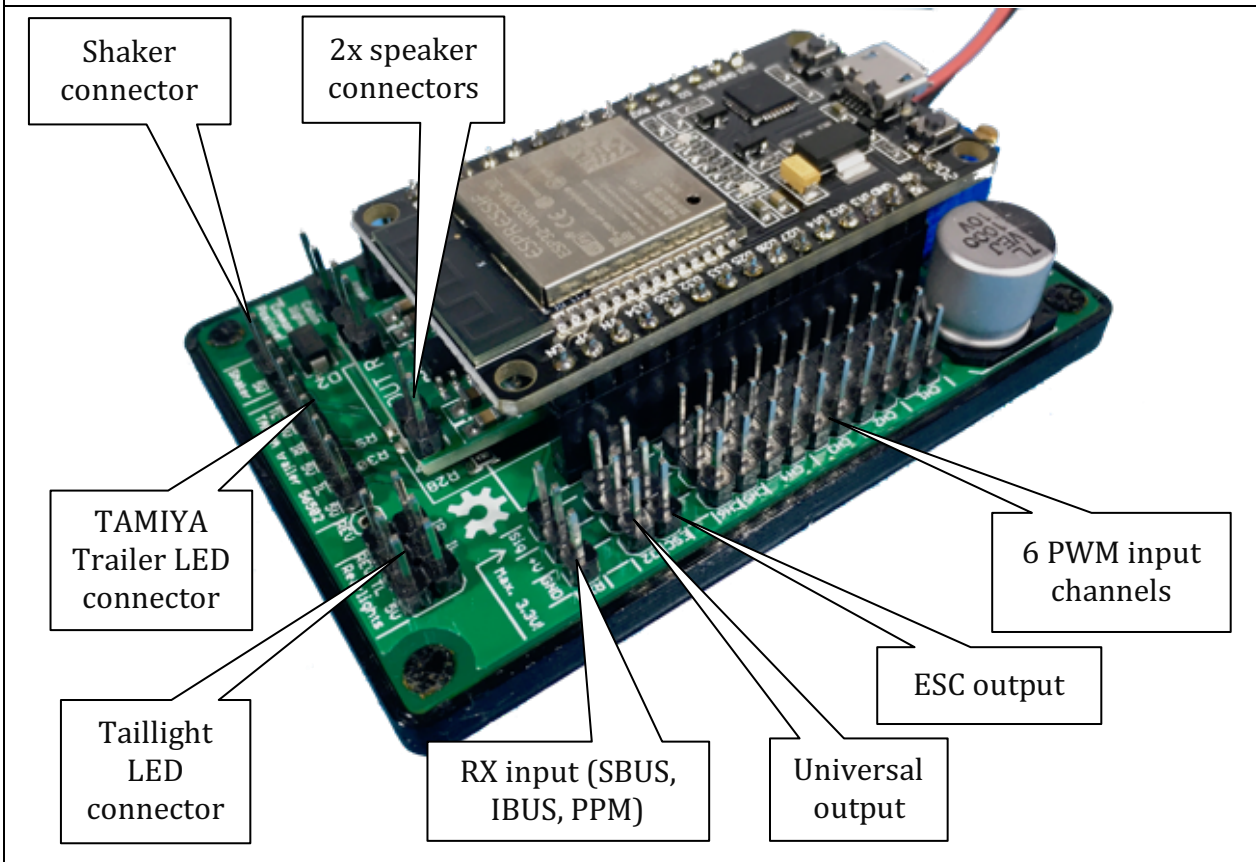
The module is **compatible** with most remote systems, which use **PWM, PPM, SBUS or IBUS** communication. A flexible channel mapping is implemented. For details refer to the included "adjustmentsRemote.xlsx"

ESP32 RC Sound & Light Controller

Quick Start Manual

System overview





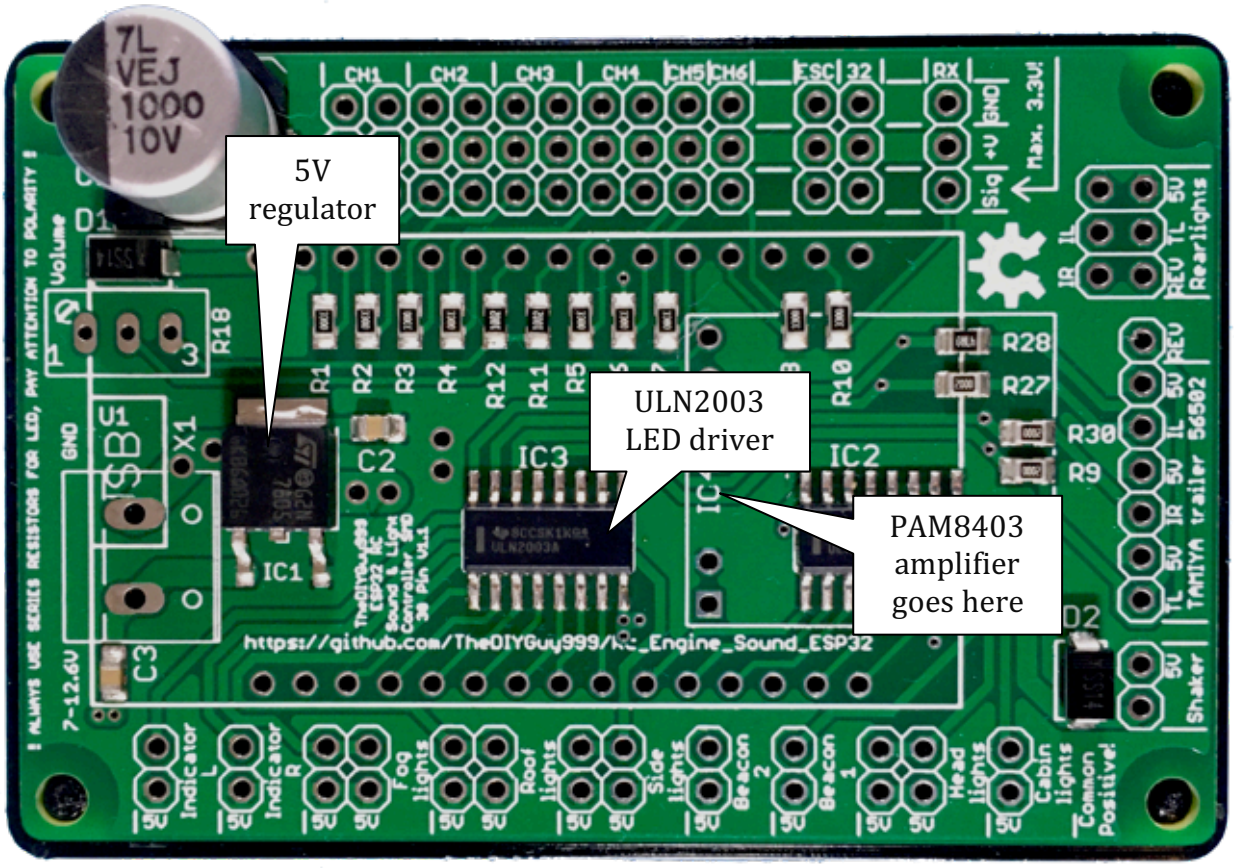
ESP32 RC Sound & Light Controller

Quick Start Manual

This view shows an unfinished board as it is delivered by PCBWay.com. How to order it: https://github.com/TheDIYGuy999/Rc_Engine_Sound_ESP32/tree/master/Eagle_PCB/SMD

Only the SMD components are populated, the remaining through hole components need to be soldered by yourself. See instructions video: https://www.youtube.com/watch?v=csQgTfxRd8Y&t=1s

It also shows the connector layout in detail. CH1 – 4 are connector pairs, which eliminate the need for y-cables.



Note, that the LED need to be wired "**common positive**". This means, that the long LED legs (the + side) can be wired together, using a **shared 5V** potential. The negative side of each LED (the short wire) needs to be wired to an **appropriate current limiting resistor** and then to the LED output pin.

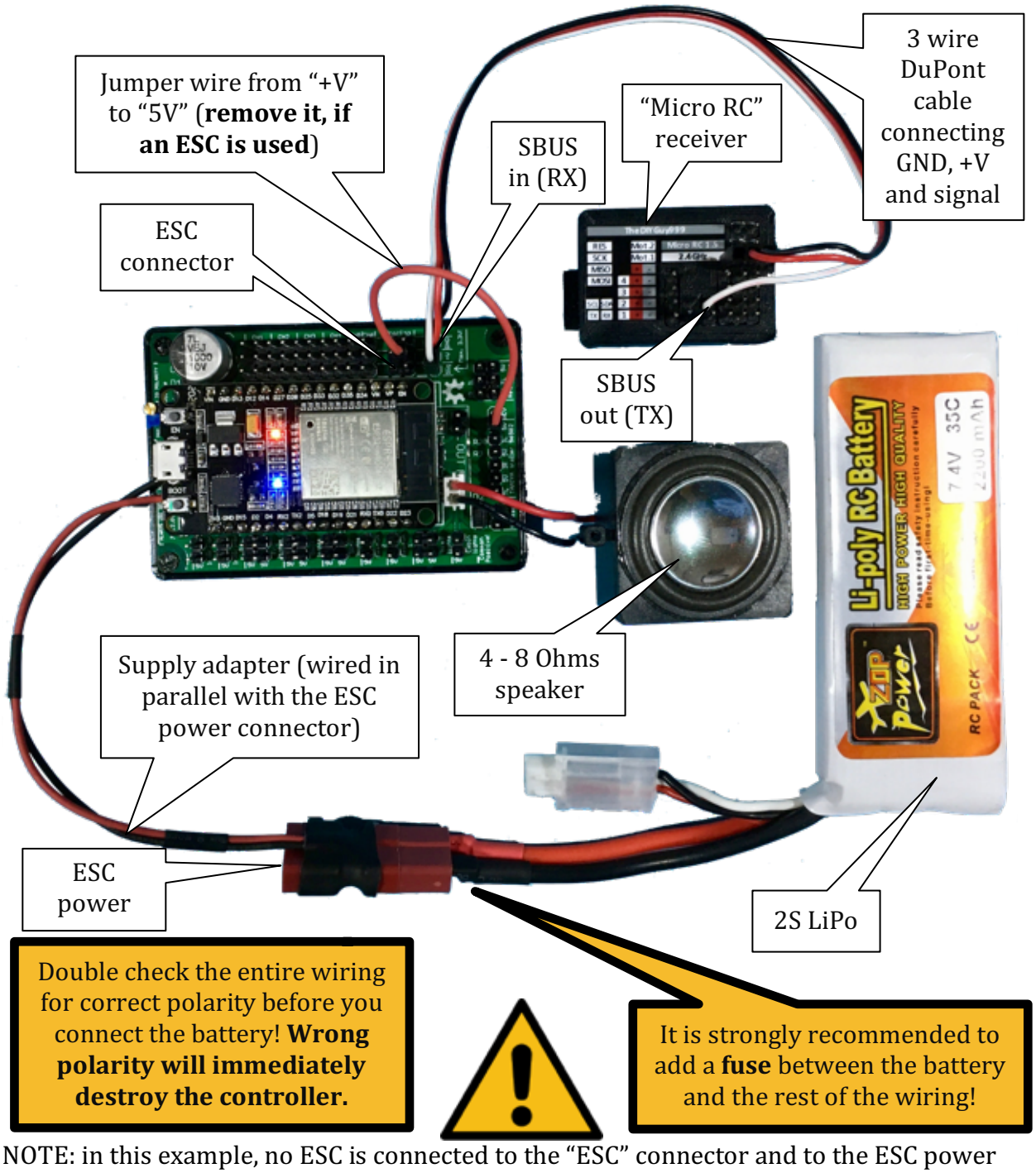
If a "**TAMIYA 56502 trailer light set** is connected to the corresponding connector, no additional current limiting resistors are required. They are already included on the PCB.

For more details, refer to the chapter "LED wiring"

ESP32 RC Sound & Light Controller

Quick Start Manual

Minimum wiring (SBUS mode)



NOTE: in this example, no ESC is connected to the "ESC" connector and to the ESC power connector. So, we have to supply the ESP32 and the receiver (+V rail) with 5V from the internal 5V regulator, via the shown jumper wire. **This jumper wire needs to be removed, if an ESC is connected**. In this case, the ESP32 is supplied by the battery eliminator circuit (BEC), which is integrated in most ESC these days. You don't need to disconnect the battery, if you connect the USB cable, there is an anti backfeed diode (D1).

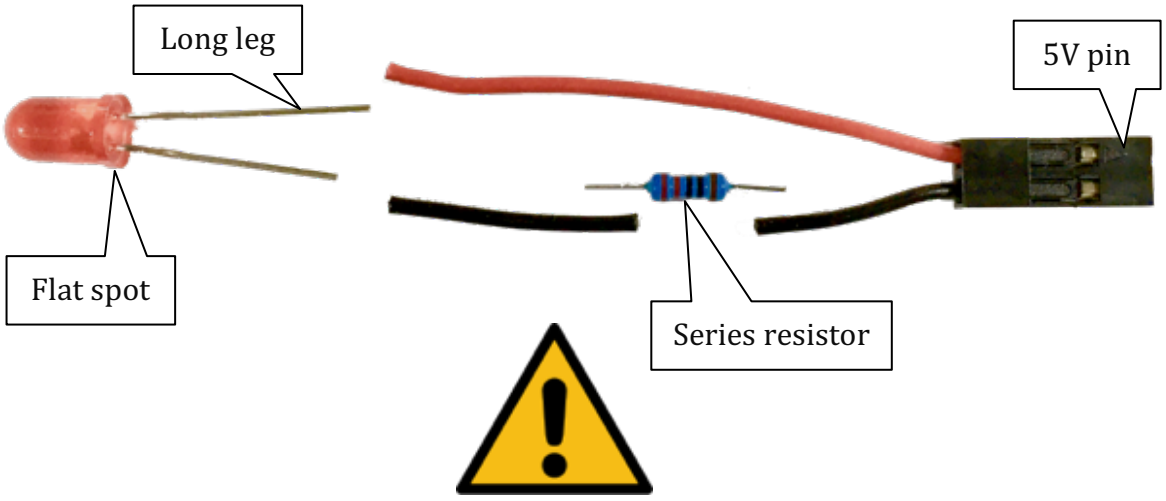
LED wiring

As already mentioned, the LED supply voltage is **5V**, coming from the internal regulator.
A common LED maximum current is **15mA**.

The following chart provides the required minimum series resistor values:

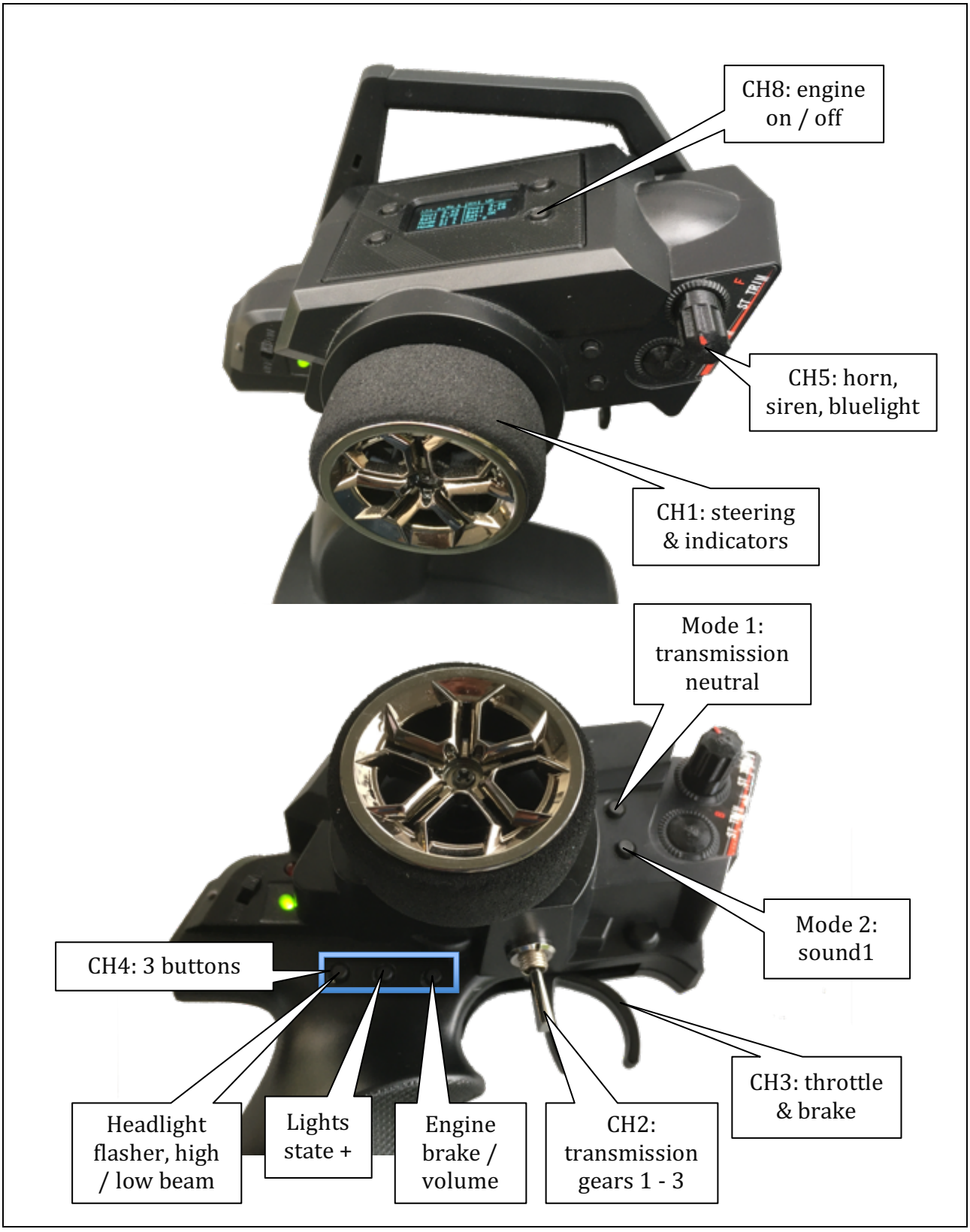
LED color	Forward voltage [Volts]	Minimum resistor [Ohms]
red	1.8	220
orange	1.9	220
green	2	200
white	3	150
blue	3	150

You can also use a calculator in order to determine the correct value:
https://www.digikey.de/en/resources/conversion-calculators/conversion-calculator-led-series-resistor



- The LED will **not light up**, if connected the **wrong way** around.
- **Never connect** LED or other devices **without** a suitable **series resistor** to the LED output pins (except the TAMIYA trailer connector). This will **permanently damage** the integrated ULN2003 LED drivers.
- If more than one LED is wired in parallel to an LED output pin, each one needs its own series resistor.
- Two red or orange LED can be wired in series. This makes the circuit **more efficient**. In this case you need to add both series voltages together for the series resistor calculation.
- The **maximum average LED output current per channel is 100mA**. However, it is not recommended to load each channel with this current. Otherwise, the ULN2003 LED driver may **overheat**. Always **check it** for excessive temperature raise for at least 10 minutes after you wired a new vehicle. If the temperature is too high to the touch, you should use an external driver transistor for the LED.
- Also note, that the **maximum sum current** for the entire 5V rail, which is supplied by the on-board regulator is **1A**.

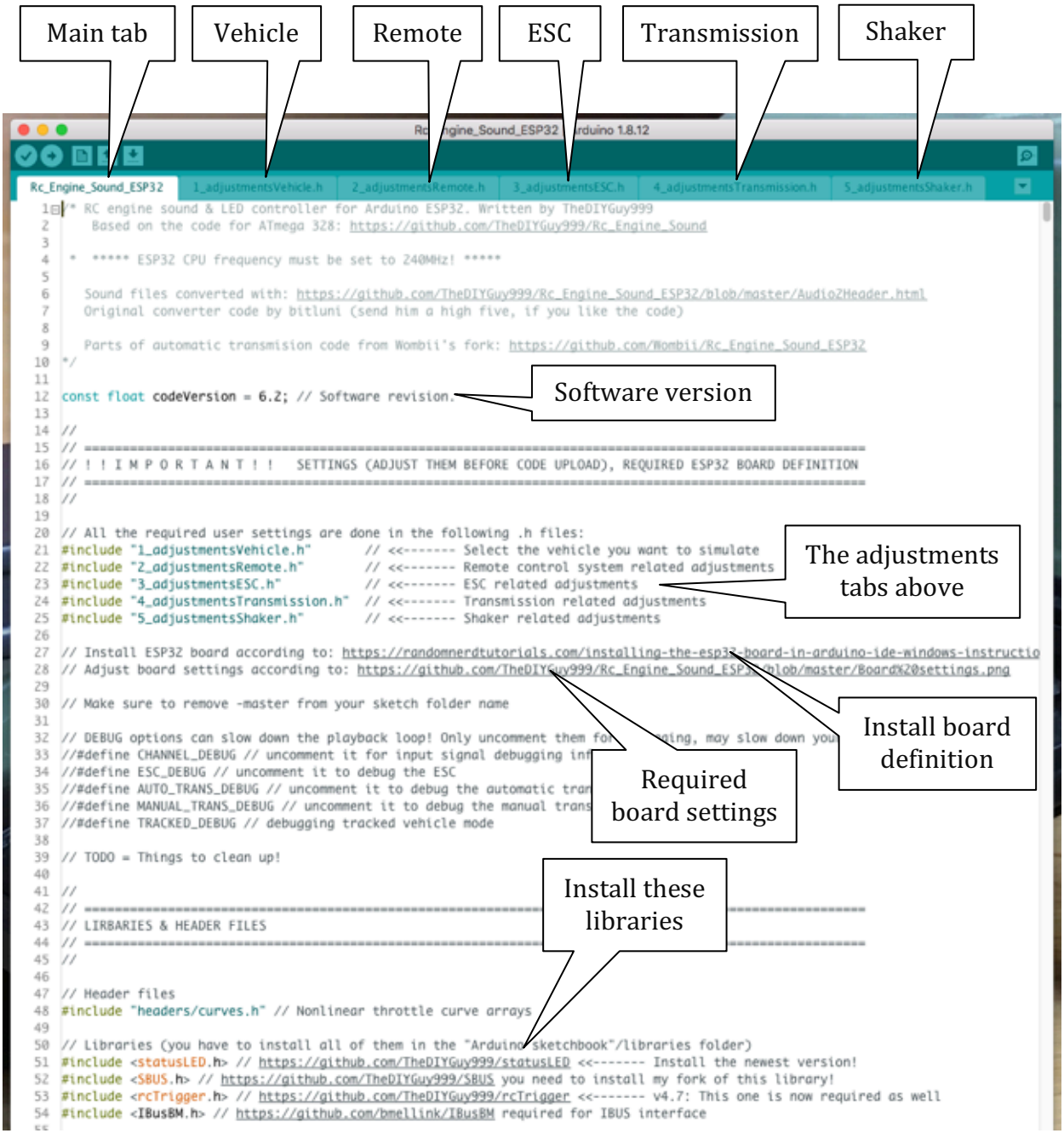
Channel assignment ("Micro RC" car style version)



Software configuration

For details refer to:
https://github.com/TheDIYGuy999/Rc_Engine_Sound_ESP32/blob/master/README.md

Do the adjustments in the “adjustments” tabs. **Don’t change the code in the main tab**.



Links

Software and 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-licht-controller.7183/

Designed in Switzerland by TheDIYGuy999