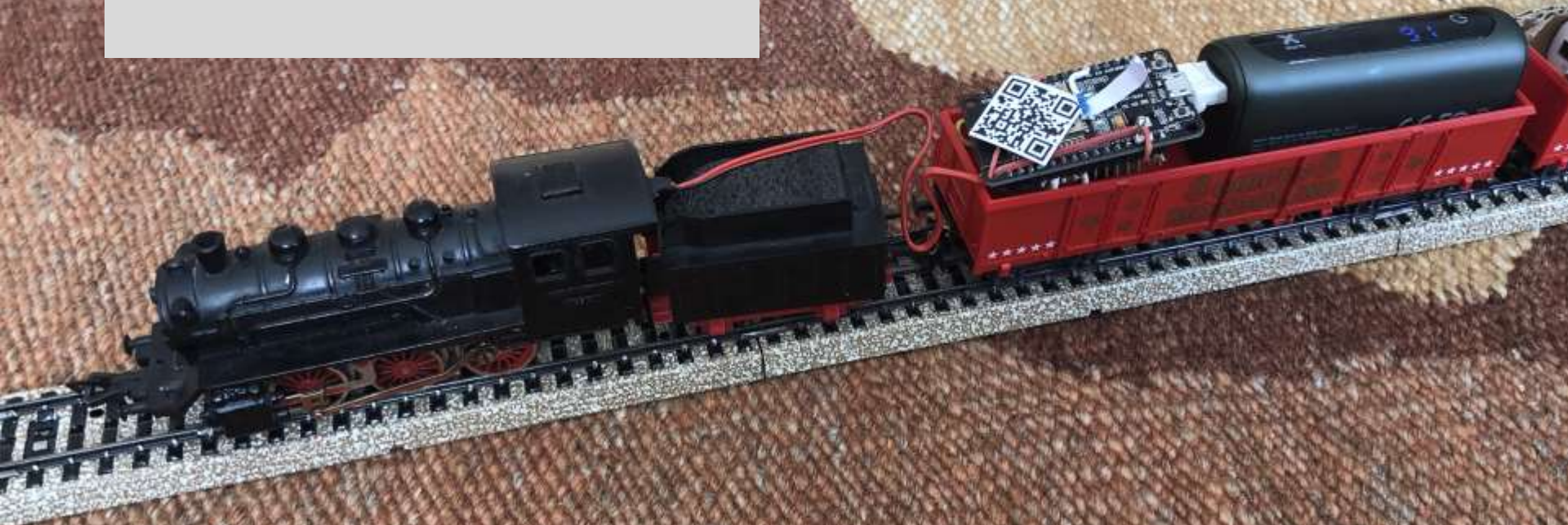


# PiedPiperS

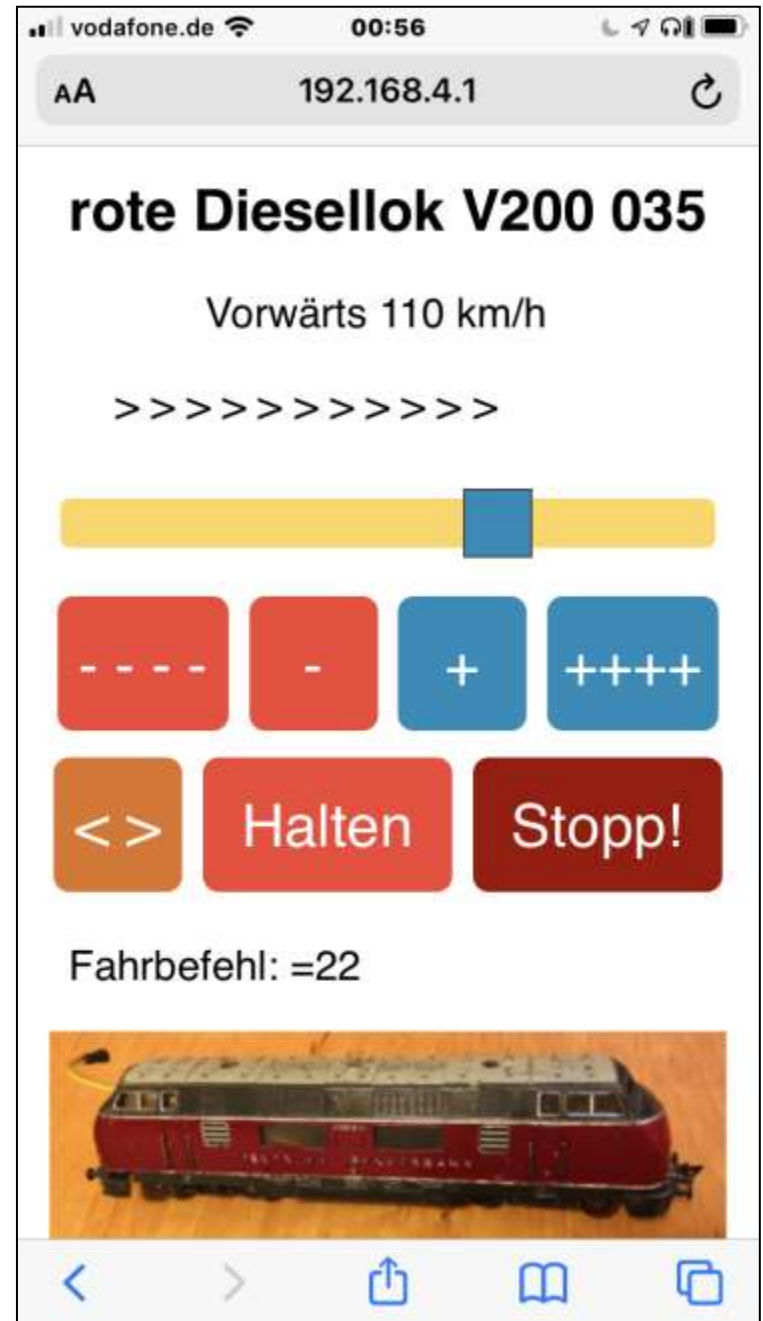
smartphone speed control  
for model trains



- Outdoor compatible, on-board USB power supply independent of track current
- ESP32 microprocessor, C++ Arduino software, with WIFI control, access point and webserver
- No proprietary elements, construction time 1 weekend, material cost ca. 25 €

# Train control website on smartphone








- Compatible with every smartphone with WIFI
- ESP32 WIFI access point mounted on train
- Browser access to control panel via webserver
- Stable WIFI remote control for up to two users
- Additional manual control via touch pin
- Feedback from train operation on two LEDs
- Motor IC error handling, short cut protection
- ESP32 voltage spike and brownout protection
- Option for online power measurement and data display in tables and charts
- Option for true speed measurement by IR sensor detection of railway sleepers



# PiedPiperS touch commands

Touch code	Command for speed level	Meaning
•	--	decrease speed
••	-----	slow down
•••	-----	break to halt
••••	0	fast brake and stop
•••••	00<	fast stop, reverse direction
• —	++	increase speed
• — —	+++++++	speed up
• — — —	+++++	go fast
—	0	fast brake and stop
— —	00	fast brake and stop
————	00	fast brake and stop
— — —	?	info?

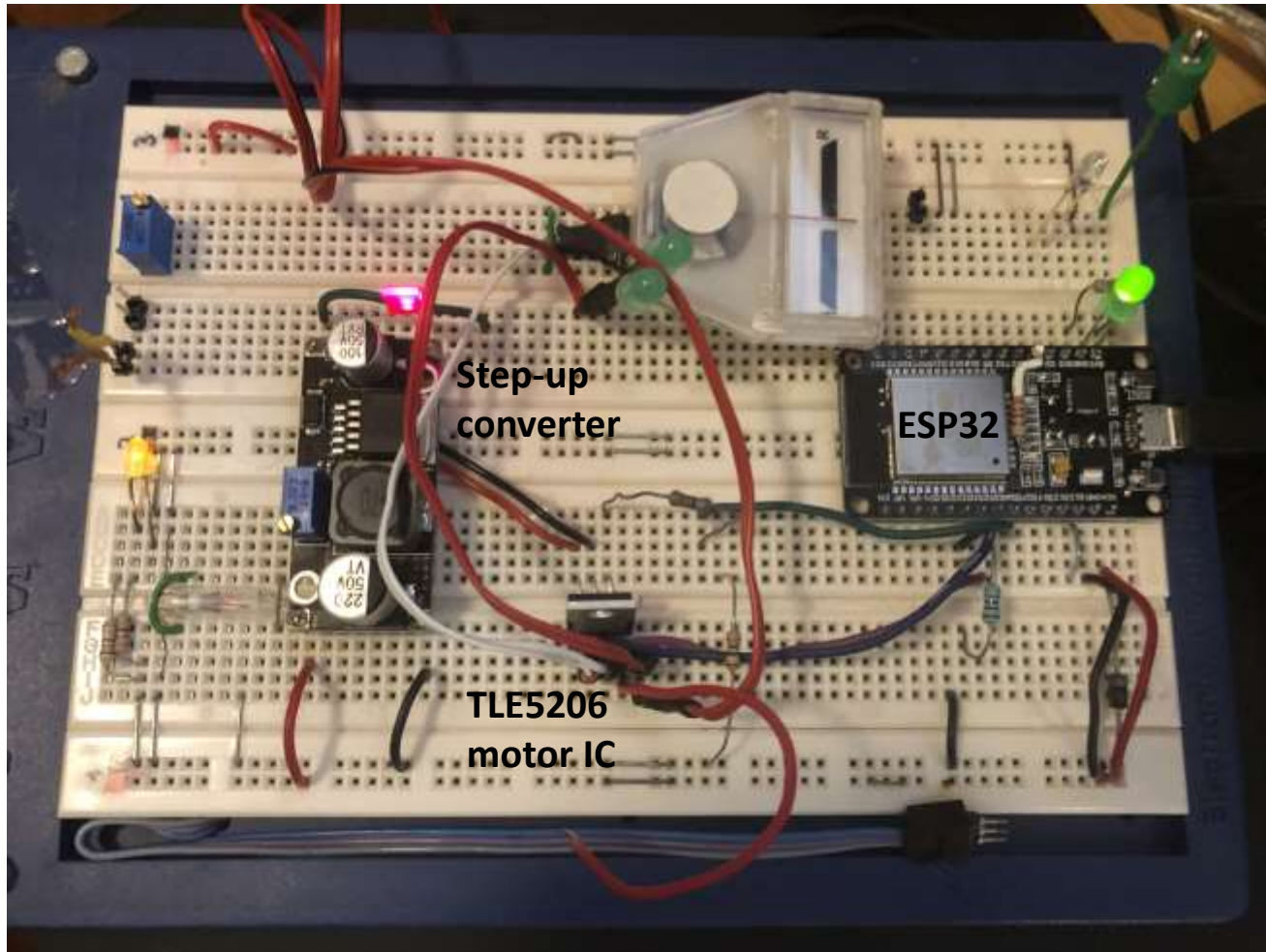
# PiedPiper LED indicators

	LEDs  
Information on motor direction and speedlevel 0 to 32: <ul style="list-style-type: none"> <li>• forward = 1 = blue LED + 0 to 16 short LED flashes</li> <li>• backward = 0 = red LED + 0 to 16 short LED flashes</li> </ul>	 
Adjustment of motor speed: <ul style="list-style-type: none"> <li>• 2 blue LED flash = speed level + 2;</li> <li>• 2 red LED flash = speed level - 2;</li> <li>• green &amp; red LEDs long flash = fast brake, speed level = 0</li> </ul>	
Change of motor direction: <ul style="list-style-type: none"> <li>• blue/red/blue + blue LED flash = forward = 1</li> <li>• red/blue/red + red LED flash = backward = 0</li> </ul>	
Program running and main loop frequency: <ul style="list-style-type: none"> <li>• orange flash after 5,000 main loop cycles, 200 times monitoring for input activity and up to 5 speed adjustments</li> </ul>	



# PiedPiperS project development

<https://github.com/jorail/PiedPiperS>



# Material demand

## for 1 train with WiFi control

### Electronics

- Nano size microcontroller with WiFi  
e.g. ESP32 DevKit V1, etc.
  - 2 PWM outputs for motor control
  - 2 digital outputs for LED indicators
  - 1 digital input for touch switch
  - 1 digital input for motor IC error flag
- Step-up DC/DC converter module, 5 V to 12V ... 16 V
- H-Bridge DC motor control IC, e.g. TLE5206-2S
- Small size USB power bank as 5 V DC power supply  
it is good to have two, 2<sup>nd</sup> for replacement when 1<sup>st</sup> empty
- Shrink tube with diameter for holding two IC pin contacts
- Thick USB cable, old and used, but reliable for power connection with small voltage drop
- Reuse of 2 on-board LEDs, by soldering 1.5 K $\Omega$  resistor from pin  
(or alternatively LEDs: 1 blue, 1 red, 3 mm diameter plus 2 x 1.5 K $\Omega$  resistors)
- 2.7 k $\Omega$  resistor, protection for digital input
- 1 k $\Omega$  SMD resistor as artificial load after LDO
- 1 SMD TVS-diode, 5 V reverse working (standoff) voltage as protection of 5 V input from peak voltages of the Step-up converter
- 2 pole cable, thin and flexible, ca. 40 cm, for connection to motor
- 1 pole wires for breadboard and step up converter connections, ca. 20 cm in total

### Other material

- Model train locomotive with 12V ... 16 V DC motor
- Flat wagon for electronics equipment
- Eaos high board open wagon for USB power bank
- Some paper cardboard
- Single and double sided adhesive tape

### Tools

- Very fine tip solder iron with equipment
- Cutter, fine pincers
- Multi-meter tool

### Option 1 addition

- ESP32 DevKit V1
  - 2 analog inputs for voltage and current readings
- Measuring resistors 1  $\Omega$ , 1 k $\Omega$ , 10 k $\Omega$ , tolerance 1%
- 1 k $\Omega$  resistor, protection for analog input

### Option 2 addition

- ESP32 DevKit V1
  - 1 analog input for amplified IR sensor voltage reading
- Suitable reflective IR sensor
- IR LED resistor
- Signal amplifier with small transistor and pullup resistor

Estimated material cost ca. 25 € excl. other material, tools and optional additions

More detailed information on electronic parts: <https://github.com/jorail/PiedPiperS/blob/main/PiedPiperS/data/parts.html>

# PiedPiperS-Bauteile 1/3

## Materialbedarf für einen Zug mit WLAN-Steuerung

- **Mikrocontroller mit WLAN**, z.B. ESP32 Node MCU DevKit V1, oder anderer
  - 2 PWM-Ausgänge für die Motorsteuerung mit TLE5206-2S und ESP32: GPIO 32 und GPIO 33
  - 2 digital Ausgänge für die LED-Anzeigen, ESP32: GPIO 2 und GPIO 4
  - 1 digitaler Eingang als Morse-Schalter, ESP32: Berührungssensor GPIO 13=TOUCH 4
  - 1 digitaler Eingang für Fehleranzeige des Motor-ICs, ESP32: GPIO 25
- **Step-up DC/DC-Konverter-Modul**, 5 V auf 8 V bis 16 V
- **IC mit H-Brücke zur Steuerung eines Gleichstrommotors**, z.B. TLE5206-2S
- Kleine **USB-Powerbank** als 5 V DC Stromquelle. Es ist gut, 2 davon zu haben, die 2. als Ersatz, wenn die 1. leer ist. Der WLAN-Betrieb des ESP32 Mikrocontrollers benötigt beim Start und zeitweise bis zu 600 mA Stromversorgung. 5 W Leistung aus einer klassischen kleinen USB-Powerbank (siehe Abbildung) sind für kleinere Lokomotiven und effiziente Gleichstrommotoren geeignet. Bei großen H0-Modellbahnlokomotiven oder bei Modellen mit älteren Motoren sind diese meist nicht ausreichend. Eine etwas teurere, kompakte USB-C-Powerbank für die Schnellladung mit ca. 15 W Leistung und 2 bis 3 A Stromquelle hat sich für diese Lokomotiven als ausreichend und im langen Fahrbetrieb als sehr gut geeignet erwiesen.

# PiedPiperS-Bauteile 2/3

## Materialbedarf für einen Zug mit WLAN-Steuerung

### Elektronik-Bauteile für ESP32-Modifikation

- TVS-Diode mit 5 V-Arbeitsspannung (reverse standoff voltage) zum Schutz des 5 V Eingangs des Mikroprozessors vor Spannungsspitzen aus dem Step-up-Konverter
- 1 k $\Omega$  SMD-Widerstand als künstliche Last nach dem Spannungsregler, kann nach sehr vorsichtigem Auslöten eines Widerstands aus dem ESP32 Node MCU DevKit V1 wieder verwendet werden
- Nutzung von 2 on-board SMD-LEDs durch einlöten eines 1,5 k $\Omega$  Widerstands von GPIO 4 (alternativ zwei 3 mm LEDs: 1 blau, 1 rot, plus 2 x 1,5 k $\Omega$  Widerstände)

Die **Bauanleitung mit Fotos** für das PiedPiperS-Projekt liegt im Internet unter <https://github.com/jorail>



# PiedPiperS-Bauteile 3/3

## Materialbedarf für einen Zug mit WLAN-Steuerung

### Elektronik-Bauteile

- 2,7 oder 1,0 k $\Omega$  Widerstand für die Übertragung des Fehlersignals vom Motro-IC
- 2-poliges Kabel, dick, ca. 20 cm, 0,25 mm<sup>2</sup> Litze, für die Stromzufuhr mit bis zu 2,5 A von der USB-Powerbank mit wenig Spannungs- und Leistungsverlust der 5 V-Spannungsquelle  
Alternativ: USB-Kabel, ggf. alt und gebraucht, aber mit dicken Leitern (American wire gauge, AWG 22, 23 oder 24) und zuverlässig für die Stromverbindung mit geringem Spannungsverlust
- USB-2.0 Typ A Stecker, möglichst kurz und mit Lötfahnen
- 2-poliges Kabel, dünn und flexibel, ca. 30 cm, für die Verbindung zum Motor. Bei einer Versorgungsspannung für den Lokmotor von ca. 12 V sind Leiterquerschnitte einer 0,14 mm<sup>2</sup> Litze ausreichend.
- 2 x 2-Pol-IC-Kontakte zum Selbstbau einer einfachen und sehr kleinen Steckverbindung in der Zuleitung zum Motor, kann einerseits als Stecker und andererseits als Buchse in die Doppellitze eingelötet werden und passt zur Stromstärke bis 1 A
- Schrumpfschlauch, verschiedene Größen, u.a. mit einem Durchmesser, um 2-Pol-IC-Kontakte aufzunehmen

# PiedPiperS - Weitere Materialien

Materialbedarf für einen Zug mit WLAN-Steuerung

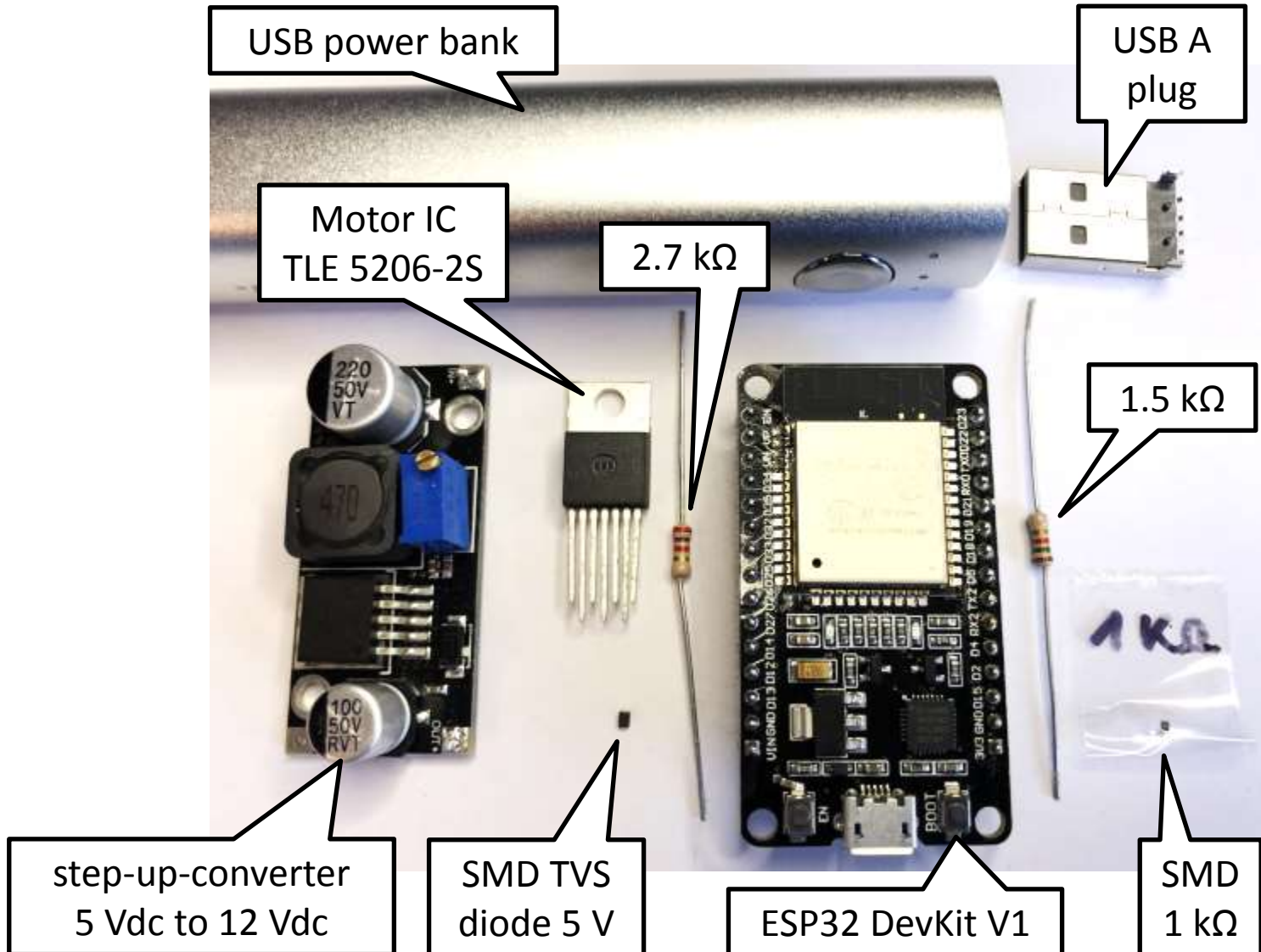
- Modellbahn-Lokomotive mit 10 V bis 24 V Gleichstrommotor
- Flachwagen für die elektronischen Bauteile
- Eaos Hochbord offener Güterwagen für USB-Powerbank
- Etwas weiße Pappe für den QR-Code
- Einfaches und doppelseitiges Klebeband
- Etwas Klebstoff
- Ein Haushaltsgummi
- USB-Datenkabel für die Programmierung des ESP32 Mikroprozessors

# PiedPiperS - Werkzeuge

## Materialbedarf für einen Zug mit WLAN-Steuerung

- Lötkolben mit sehr feiner Spitze, feiner Lötdraht und Zubehör
- Schneidmesser
- Seitenschneider
- Werkzeug zum Abisolieren
- Feine Flachzange
- Feine Spitzzange
- Spitze Pinzette für SMD-Lötarbeiten
- Lupe für SMD-Lötarbeiten
- Elektronik-Steckbrett für einen Test-Aufbau oder für eine Umsetzung der Schaltung ohne Löten. Ein kleines Steckbrett mit 170 Kontakten passt auf einen HO Modellbahnwagen. Siehe hierzu das [PiedPiper-Projekt](#)
- Multimeter

# Main electronic parts



integrated mounting and wiring with LDO protection, v100



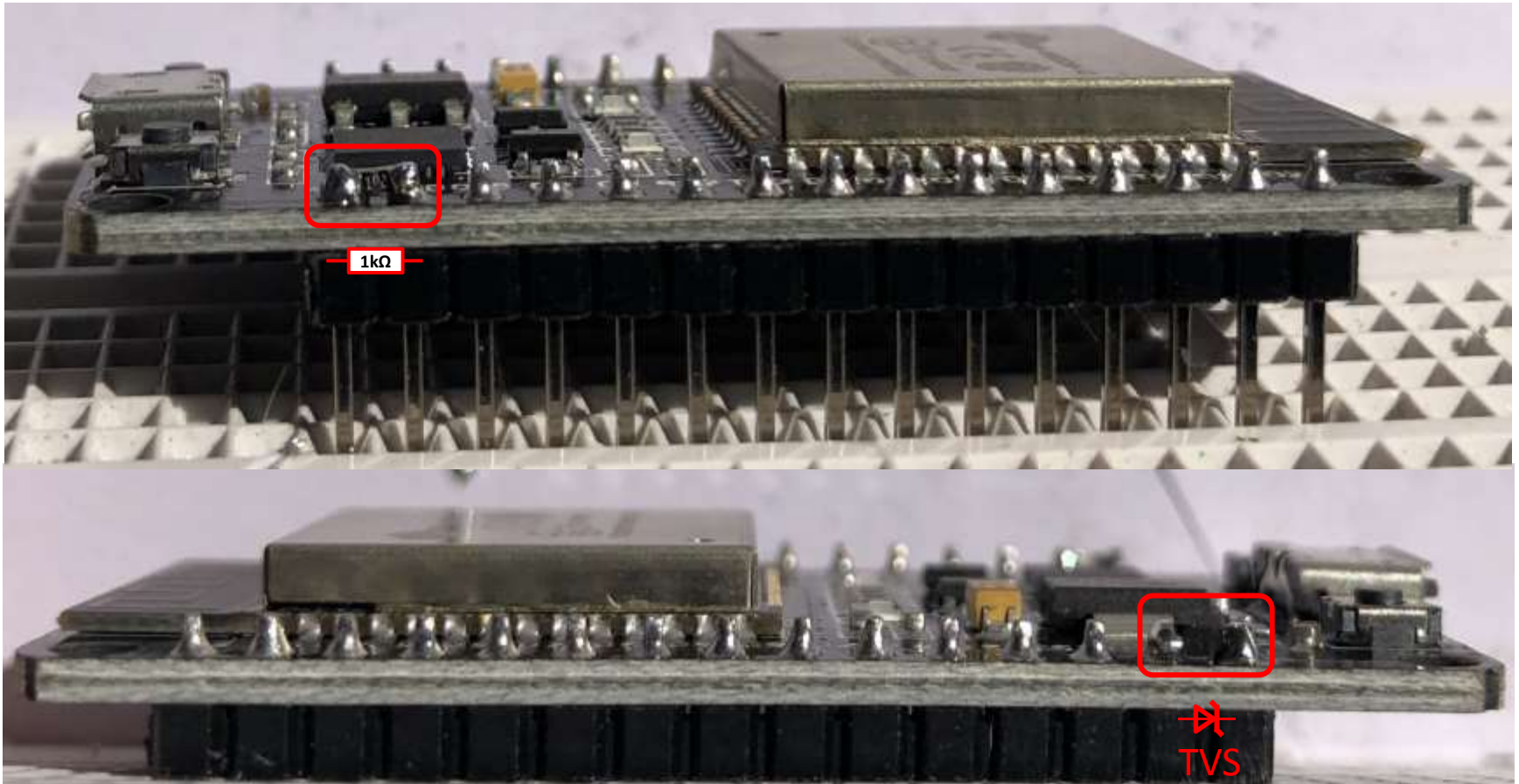
# Required tools



# Material for ESP32 topside adjustment

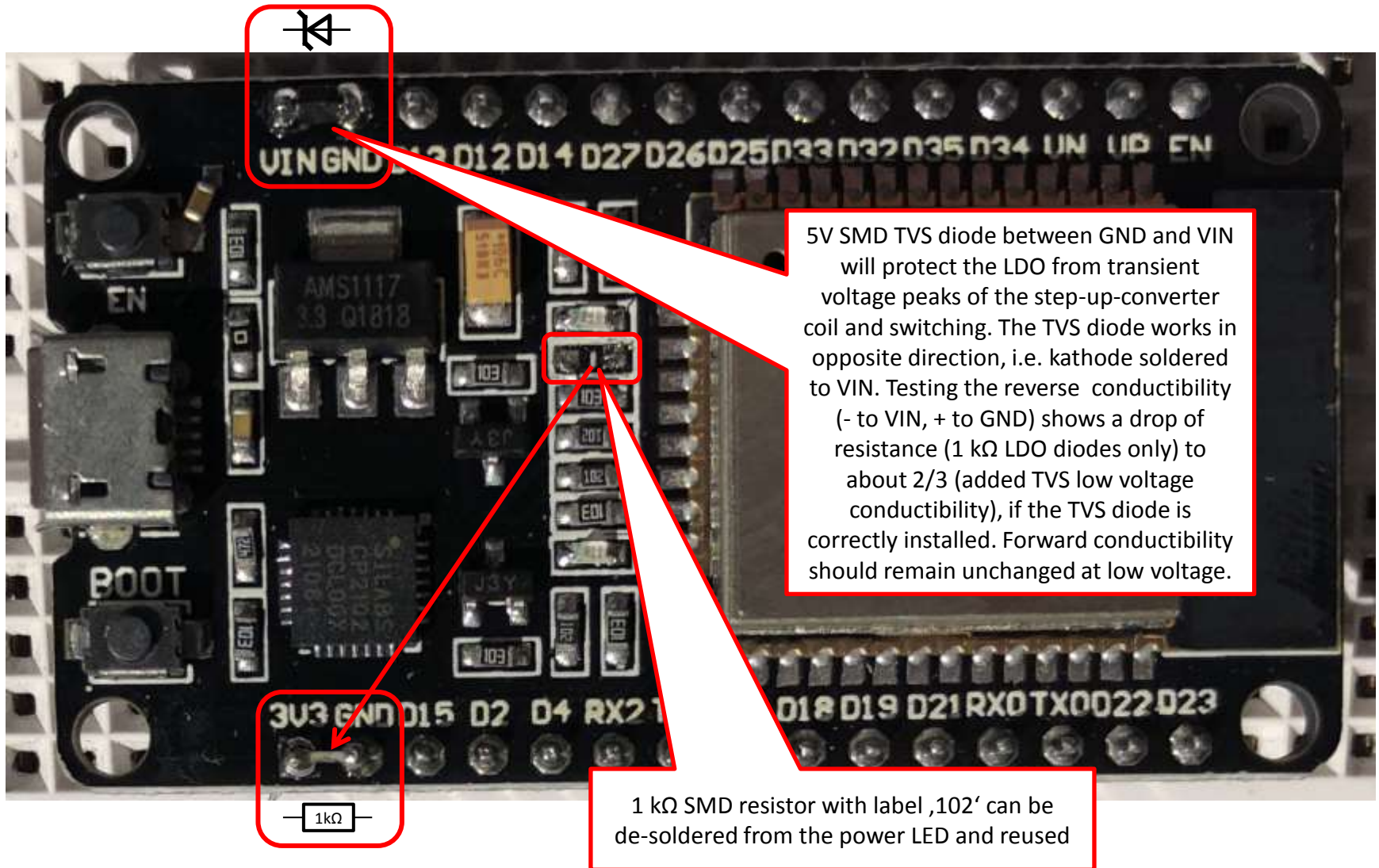


# SMD resistor and TVS diode at power pins

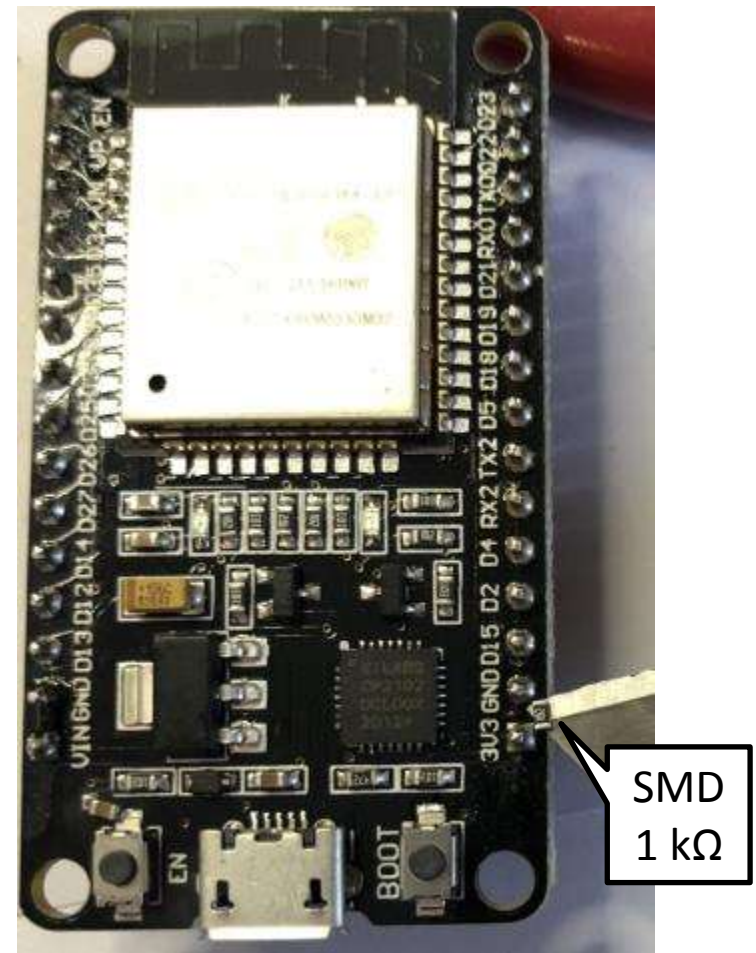
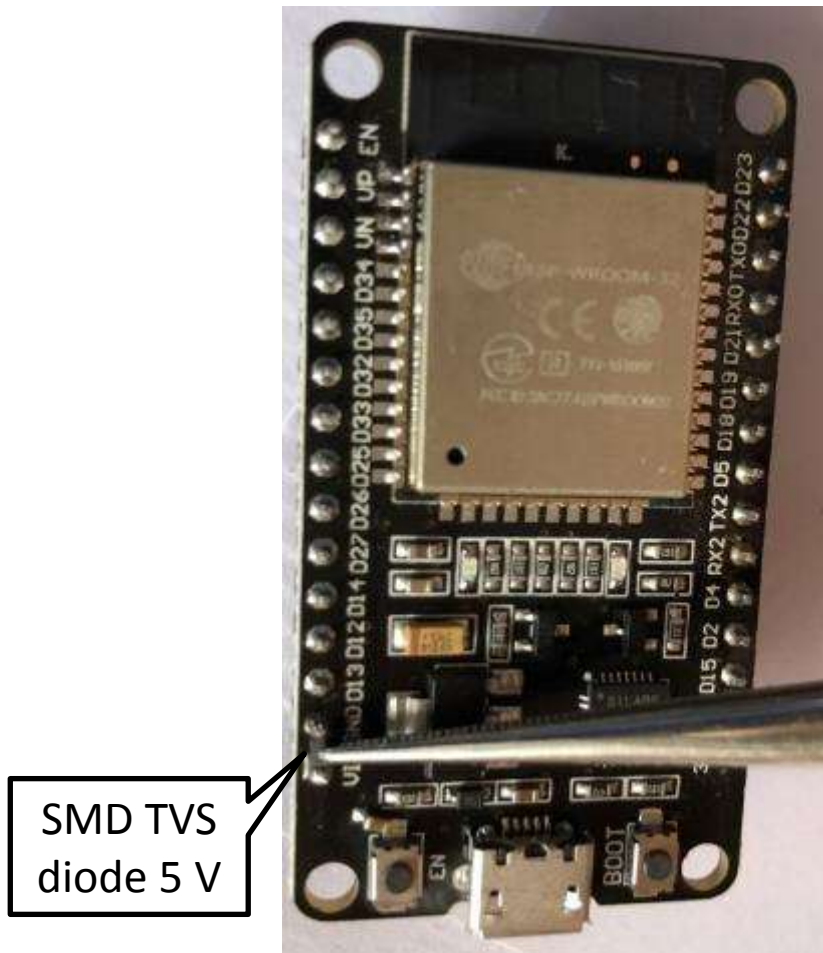




# Protecting ESP32 for power supply from USB power bank linked to step-up-converter



# Solder SMD elements at 5 V input and 3.3 V output



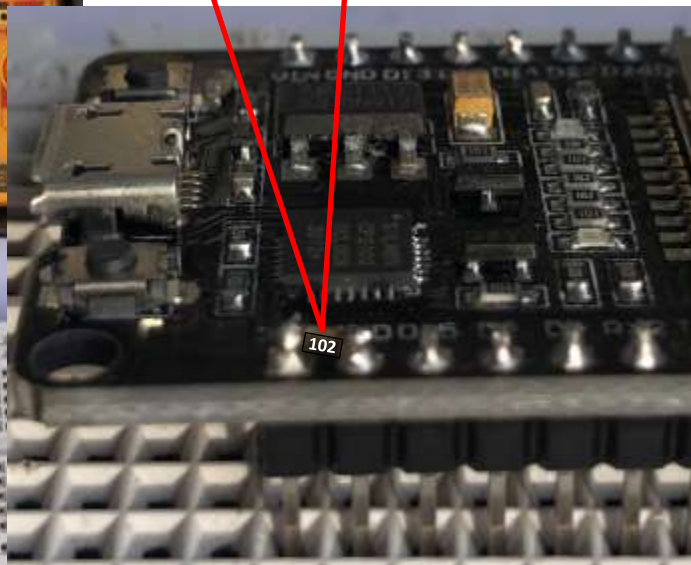


# Soldering 1 k $\Omega$ resistor assuring LDO load and protection



1 k $\Omega$  SMD resistor between 3.3V and GND assures a minimum current and load behind the low-dropout regulator (LDO) and its ability to regulate and protect the 3.3 V level from input voltage peaks in case of brownout. The initial total resistance of 2 k $\Omega$  between the 3.3 V and GND pins will drop to about 650  $\Omega$  after successful installation of the 1 k $\Omega$  resistor.

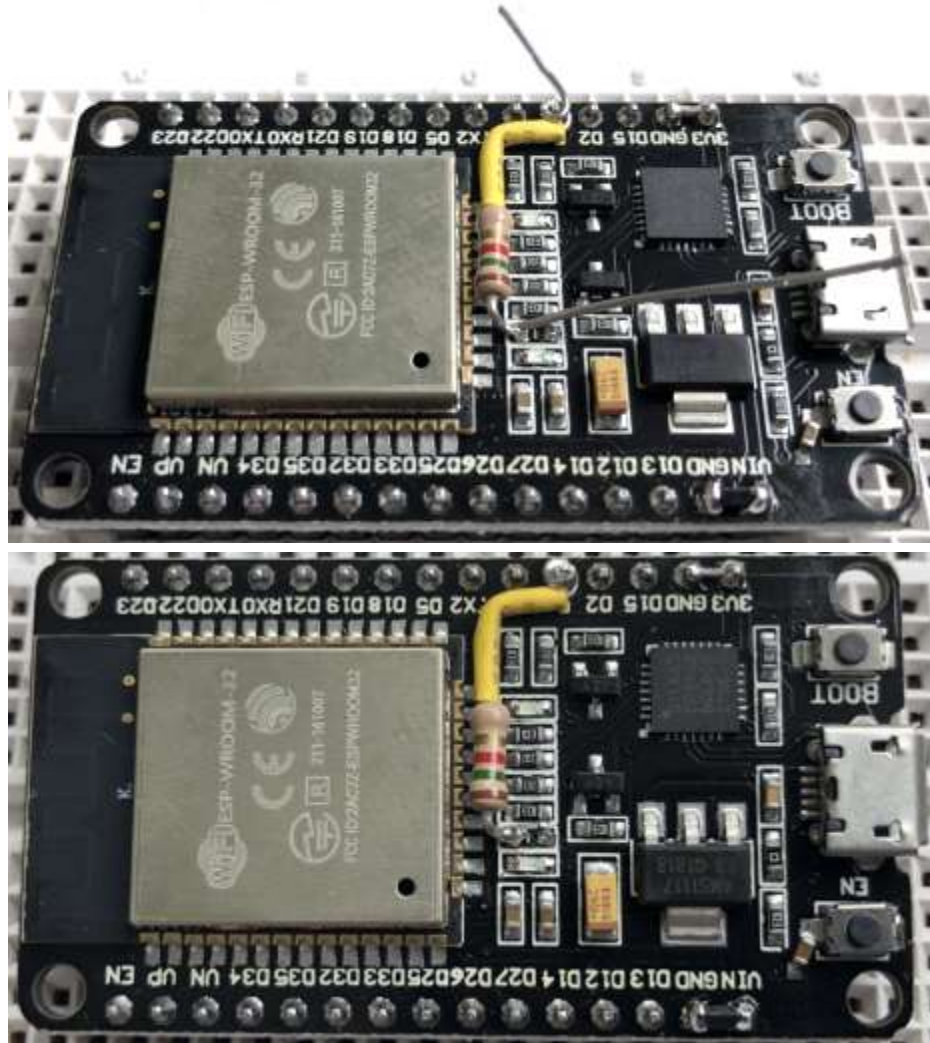
1k $\Omega$



# Connecting red LED to GPIO pin D4



1.5 k $\Omega$  resistor connects GPIO pin D4 to the de-soldered pad connecting to the red power LED. Use some small shrink tube to protect the wire from unintended contacts. Do not cover the blue LED corresponding to D2.



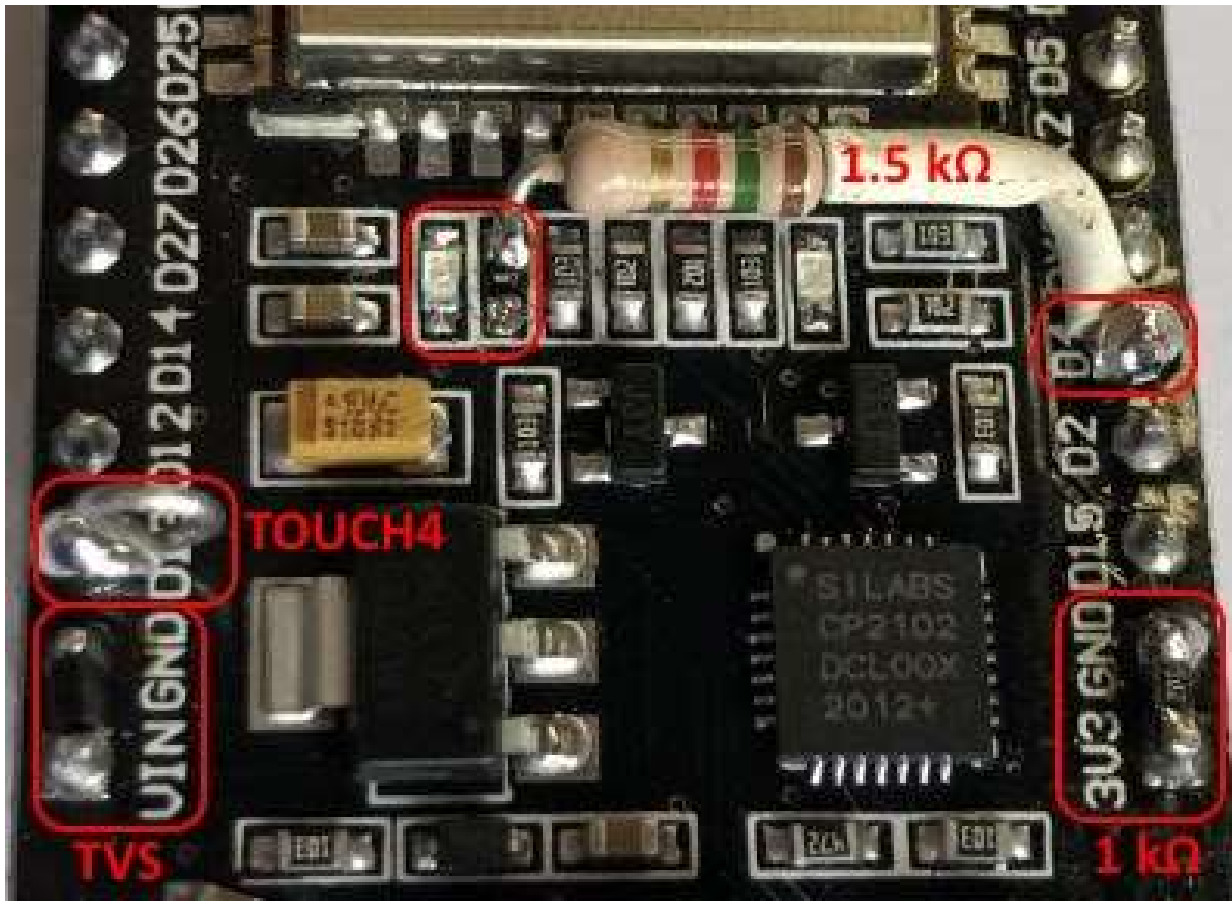
# Installing a manual Touch input



Solder a very small loop of strong wire to GPIO pin D13 as TOUCH4 input for manual commands to the model train (instead of a manual switch)



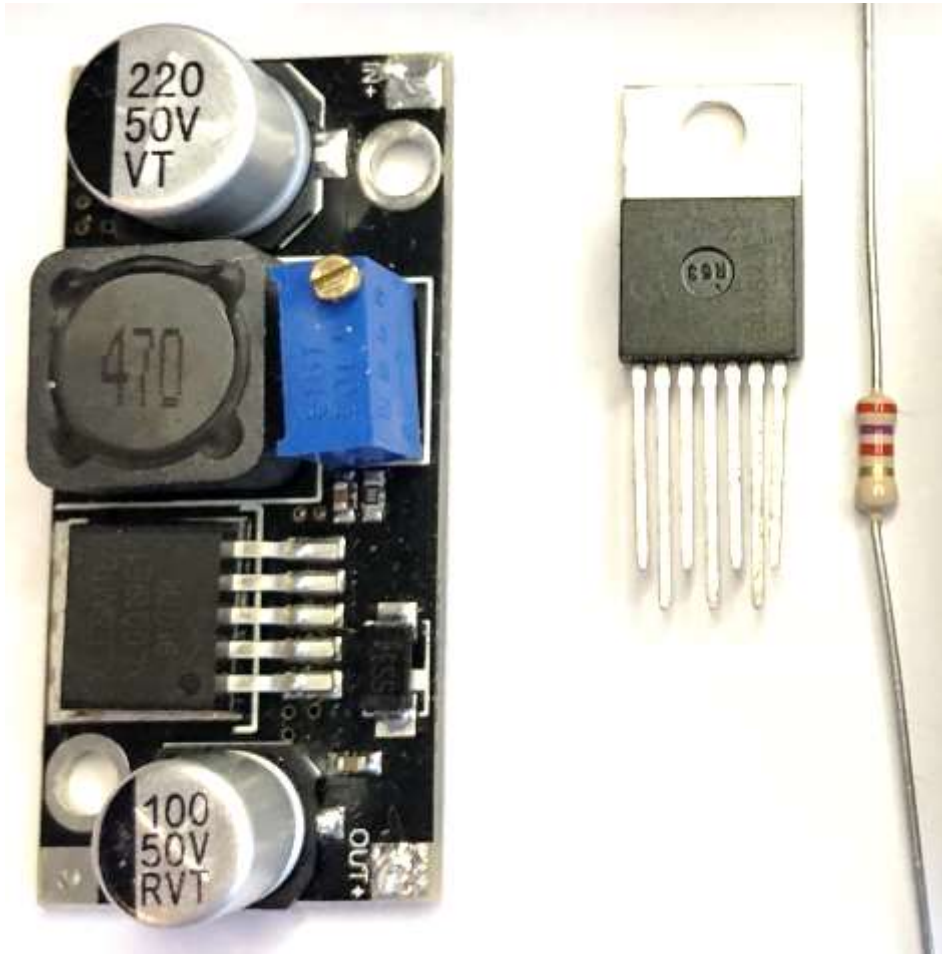
# ESP32 DEVKIT V1 top view



- power LED modification by 1.5 kΩ resistor to D4 pin
- TOUCH4 pin wire loop
- TVS diode 5 V and load by 1 kΩ resistor at 3.3 V for LDO protection

# Materials for motor control

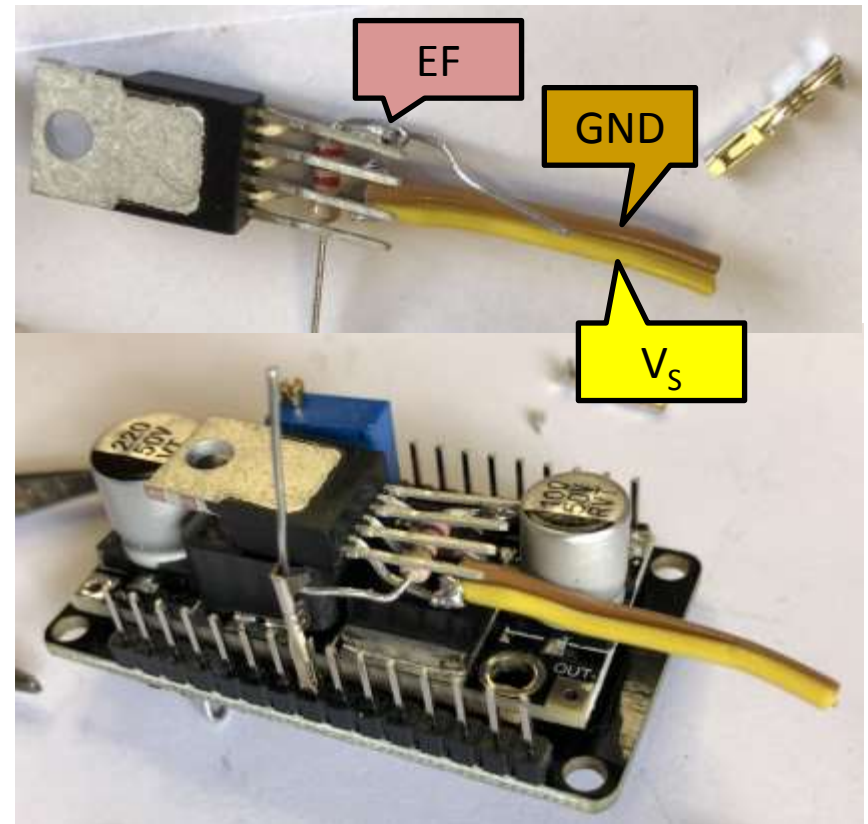
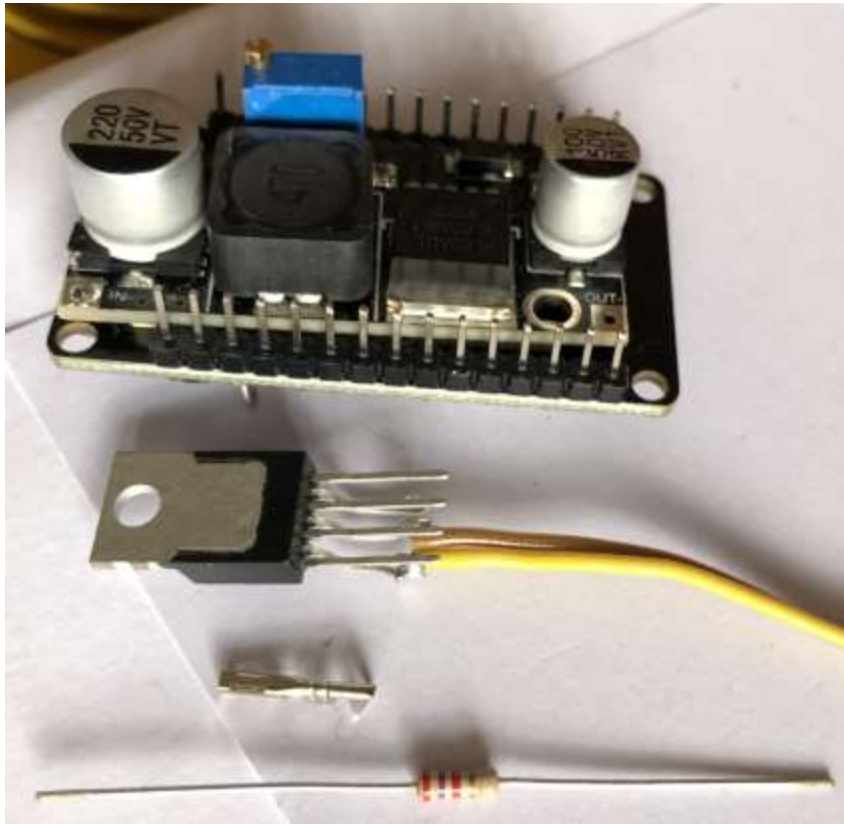
step-up-converter and motor IC fitting at ESP32 bottom

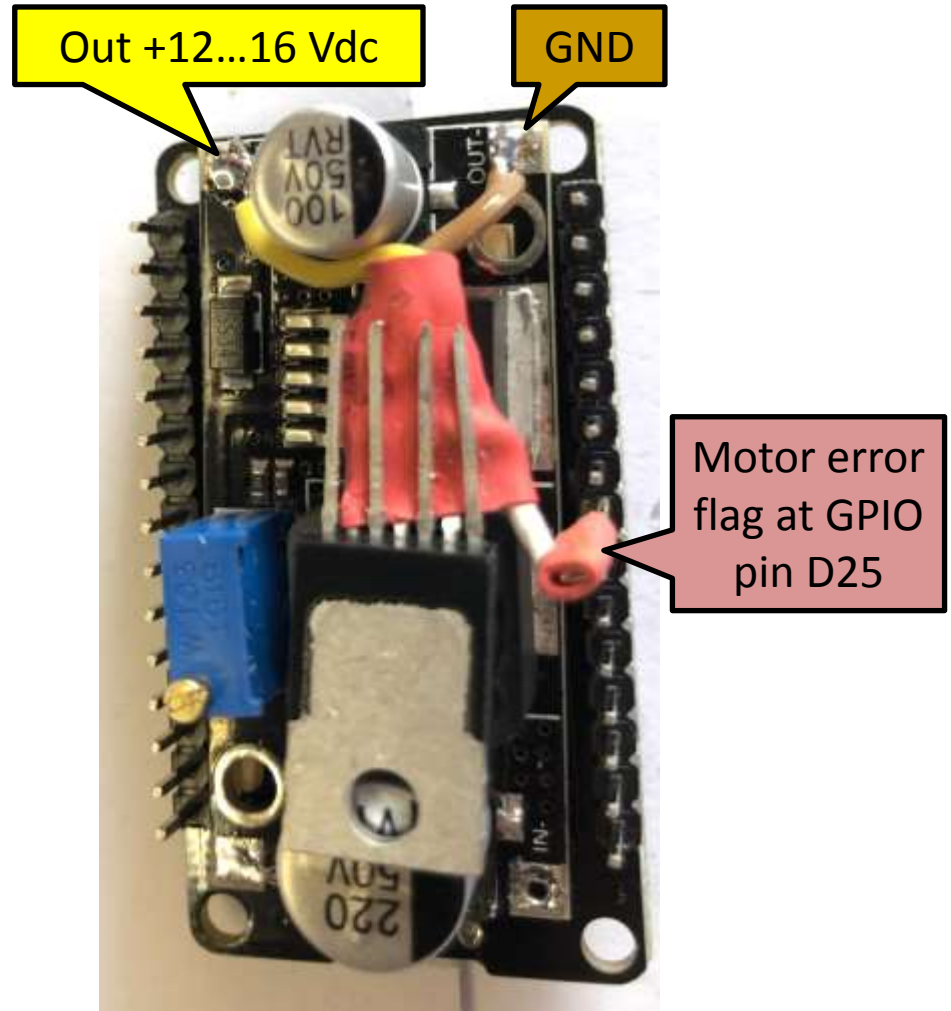
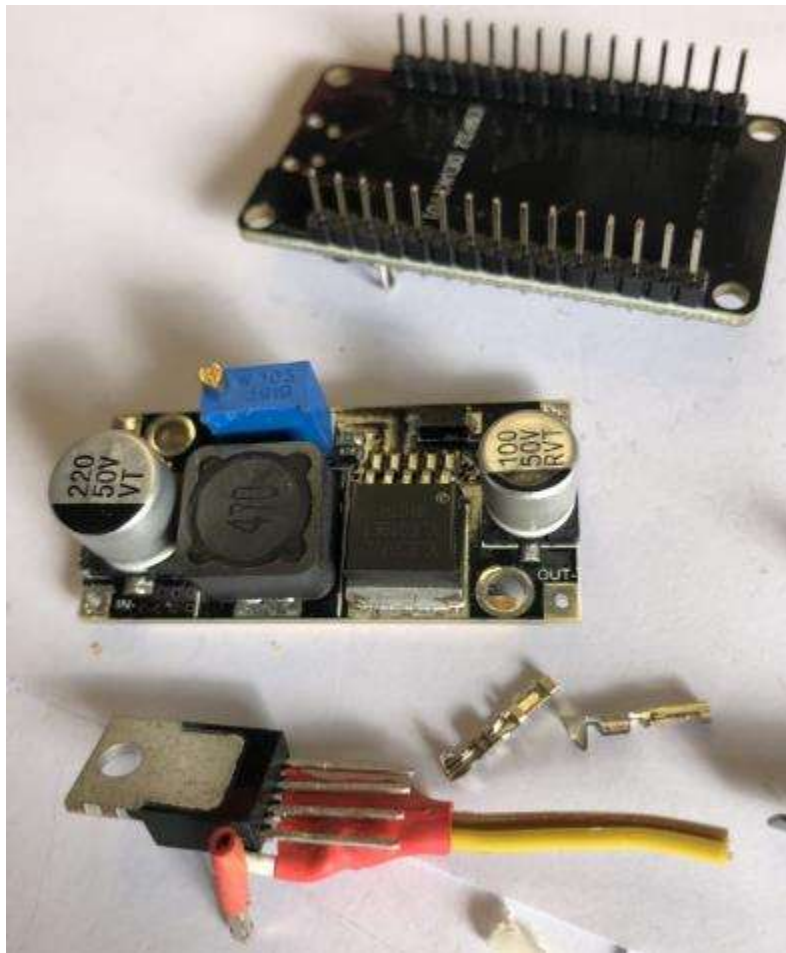


Glue two matches on bottom of step-up-converter for fitting with some distance to ESP32 bottom side



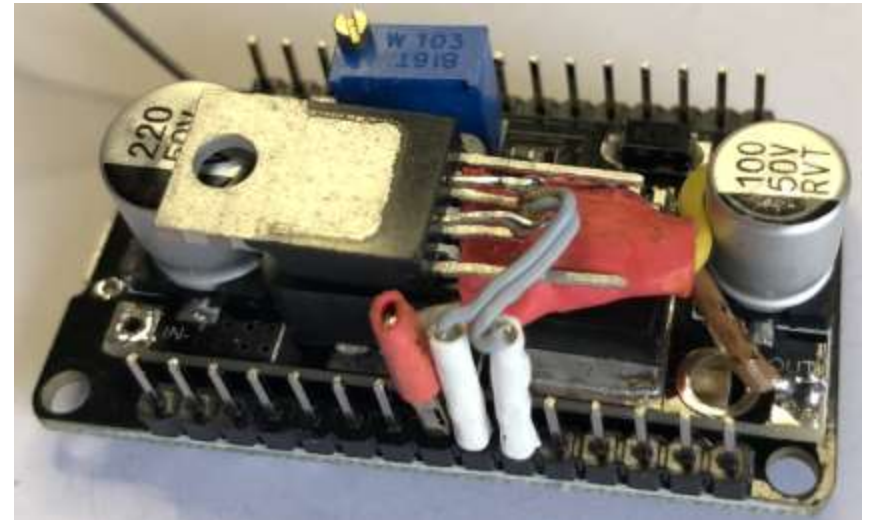
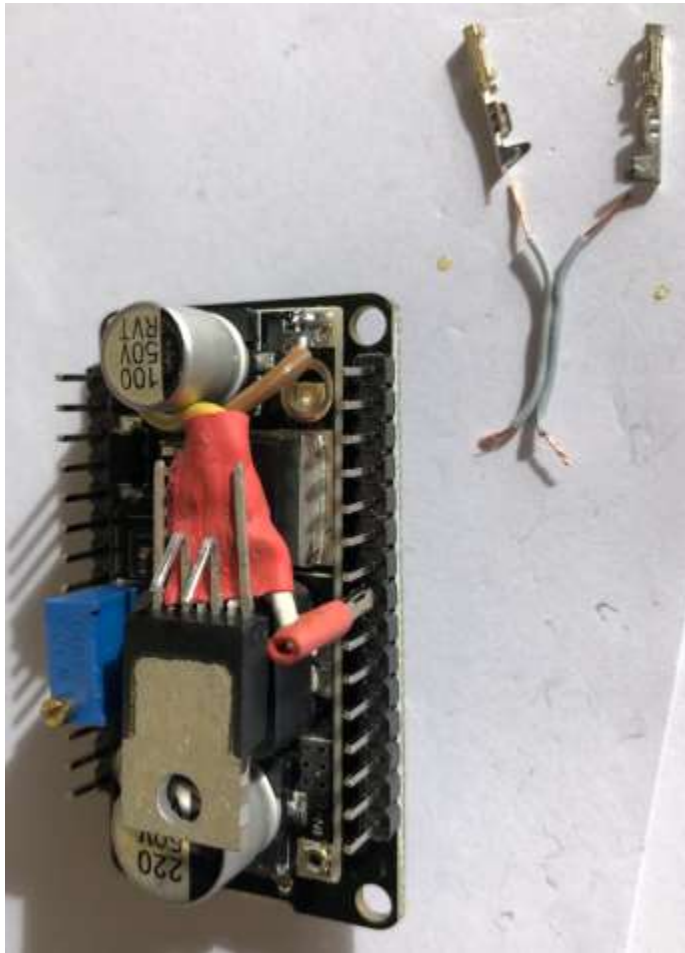
# Mounting motor power control IC





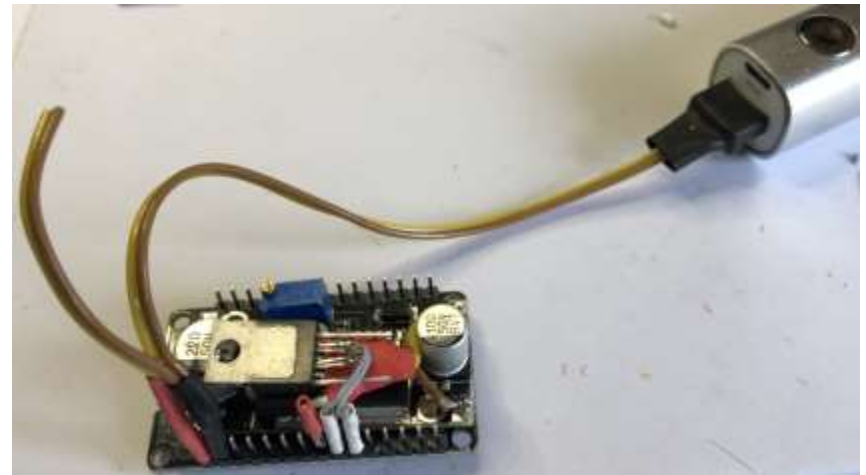
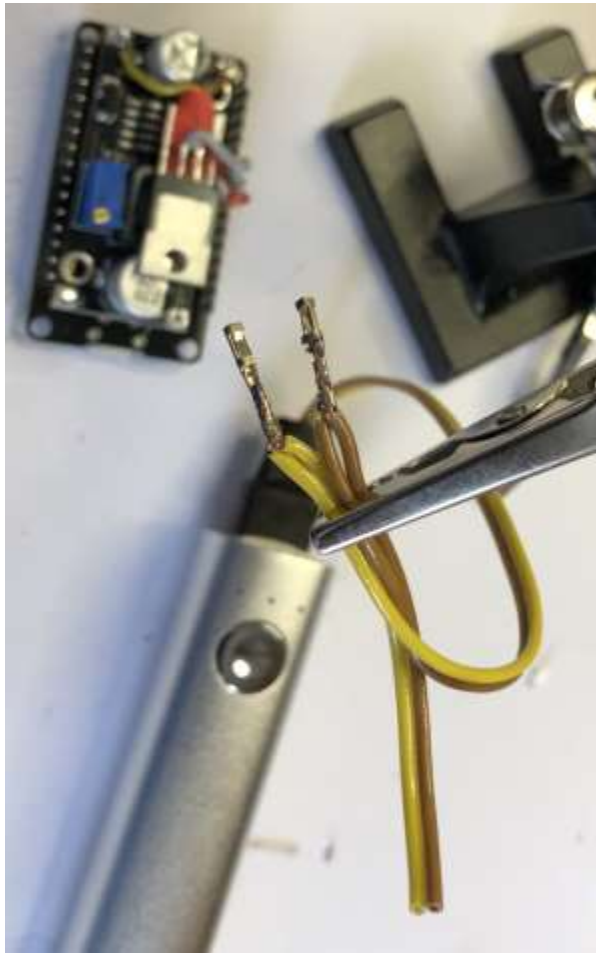
Isolate the soldered pins with heat shrink tubing.

# Connect ESP32 GPIO pins D33 and D32 to motor IC IN1 and IN2



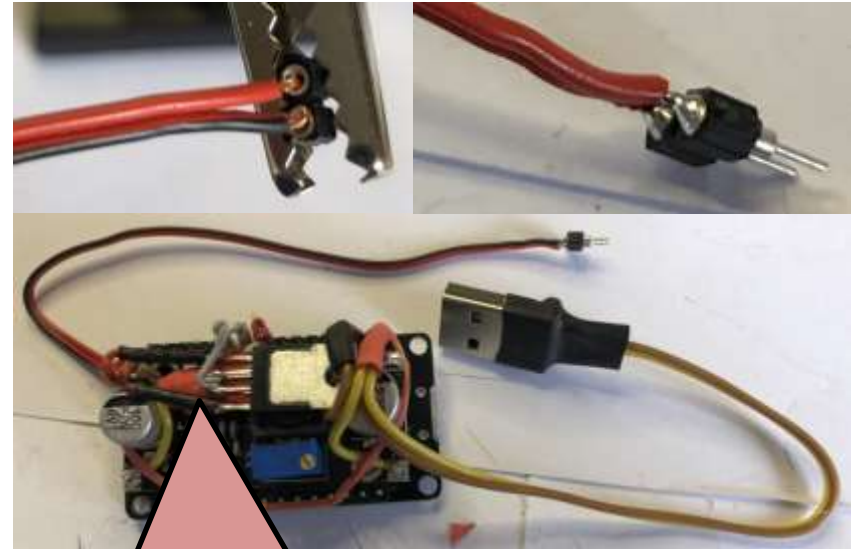
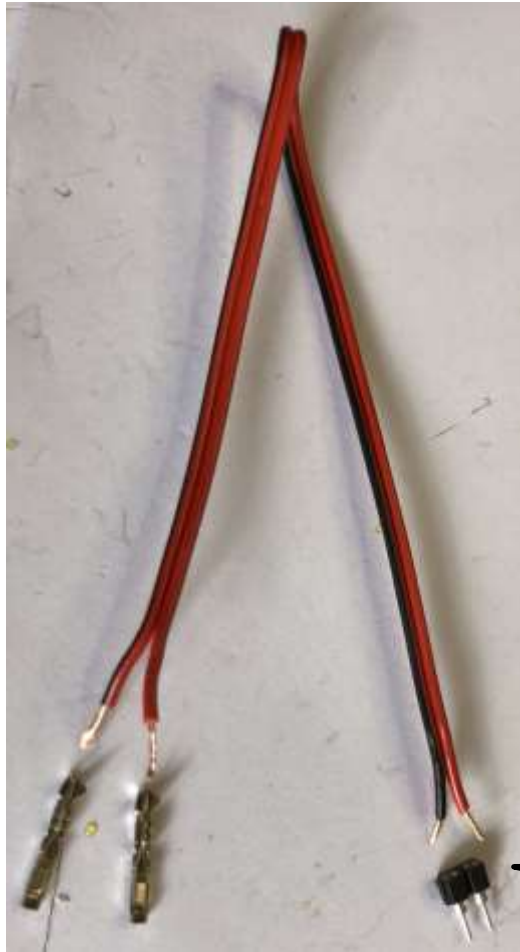


# Connect USB power bank to ESP32 VDD, GND and In+, In- of Step-up-converter



Use at least 0.25 mm<sup>2</sup> cable cross-section for  
limiting power loss before step-up converter

# Cable with self-made mini plug for connection to locomotive motor



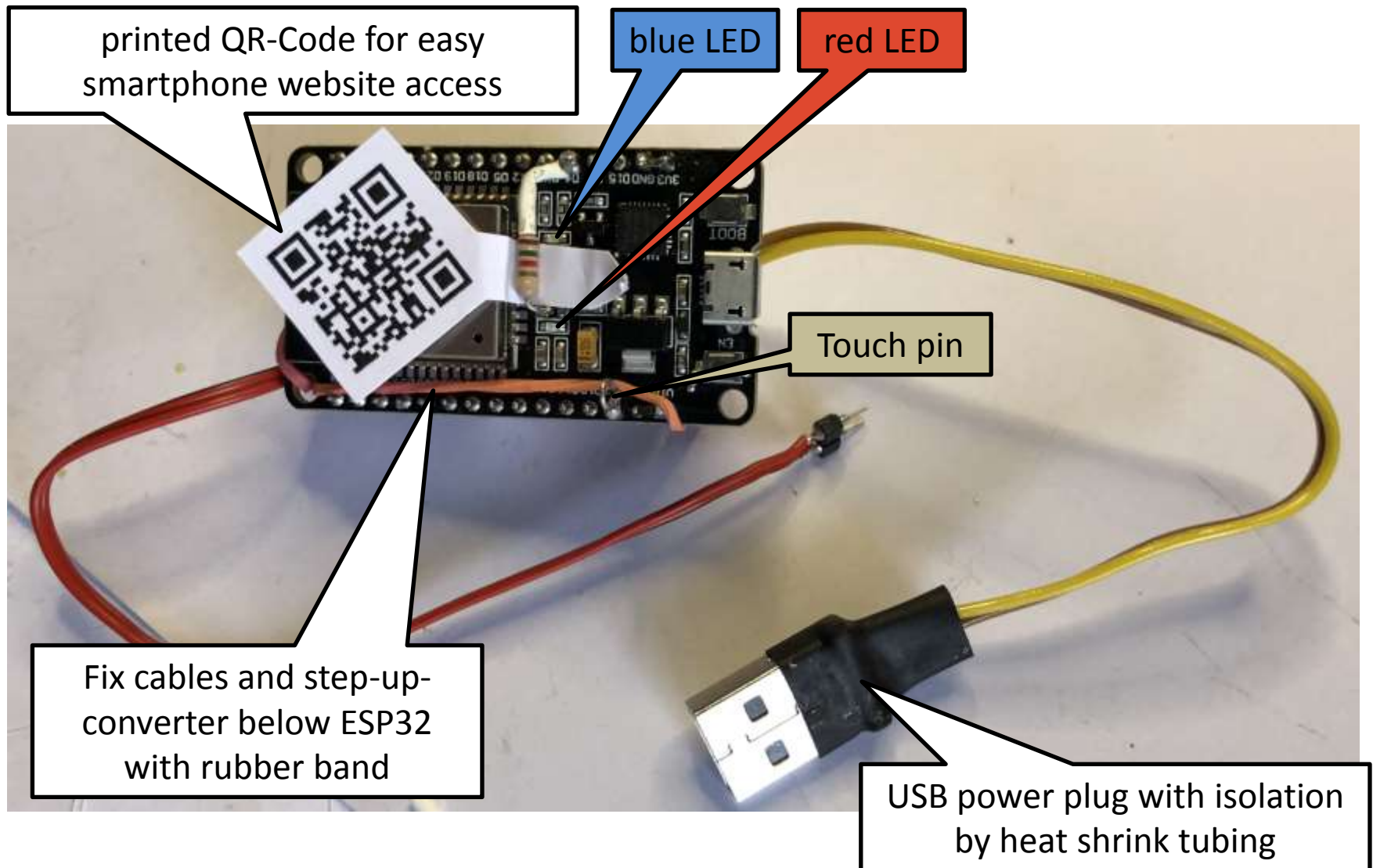
Insulate and connect motor cable to motor IC OUT1 and OUT2

Self made male mini plug for connection to locomotive motor





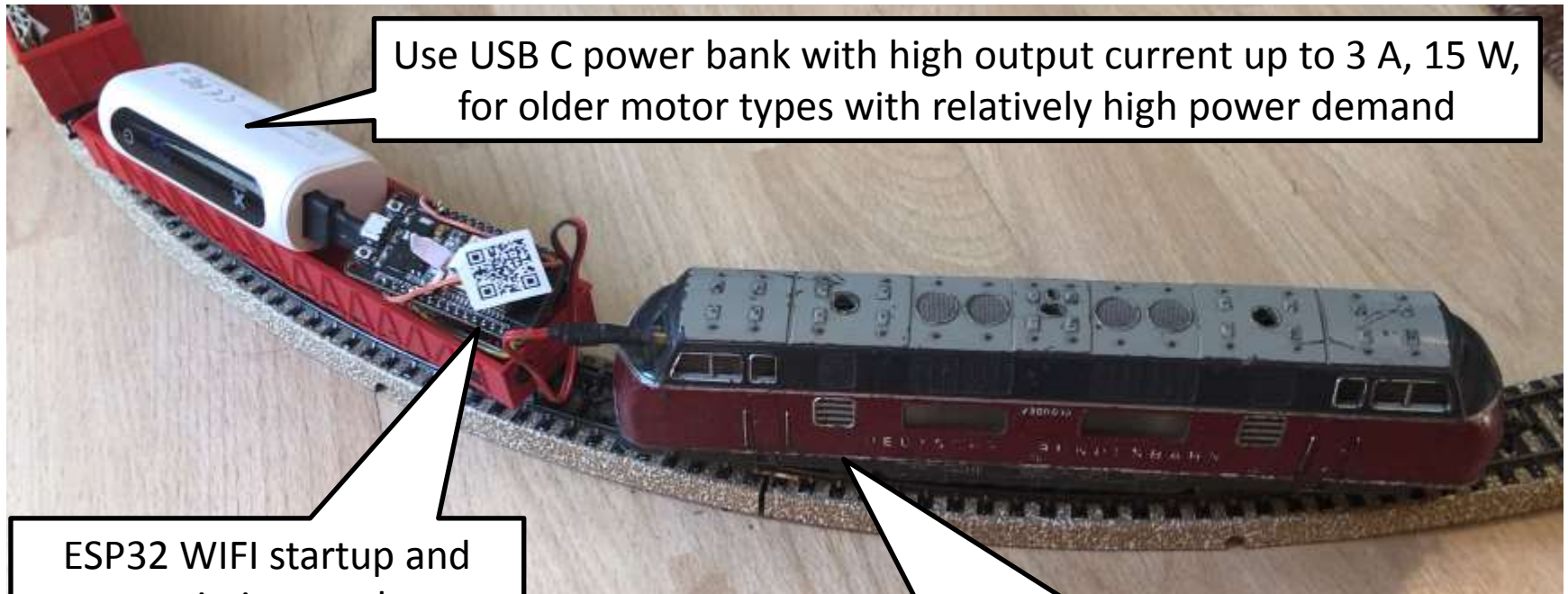
# Completing ESP32 motor control module



# Upload the PiedPiperS sketch and test the motor control



# Complete mounting on first wagon behind locomotive



Use USB C power bank with high output current up to 3 A, 15 W, for older motor types with relatively high power demand

ESP32 WIFI startup and transmission can have significant power demand, reported up to 500 mA

Train can run completely independent of rail power supply.

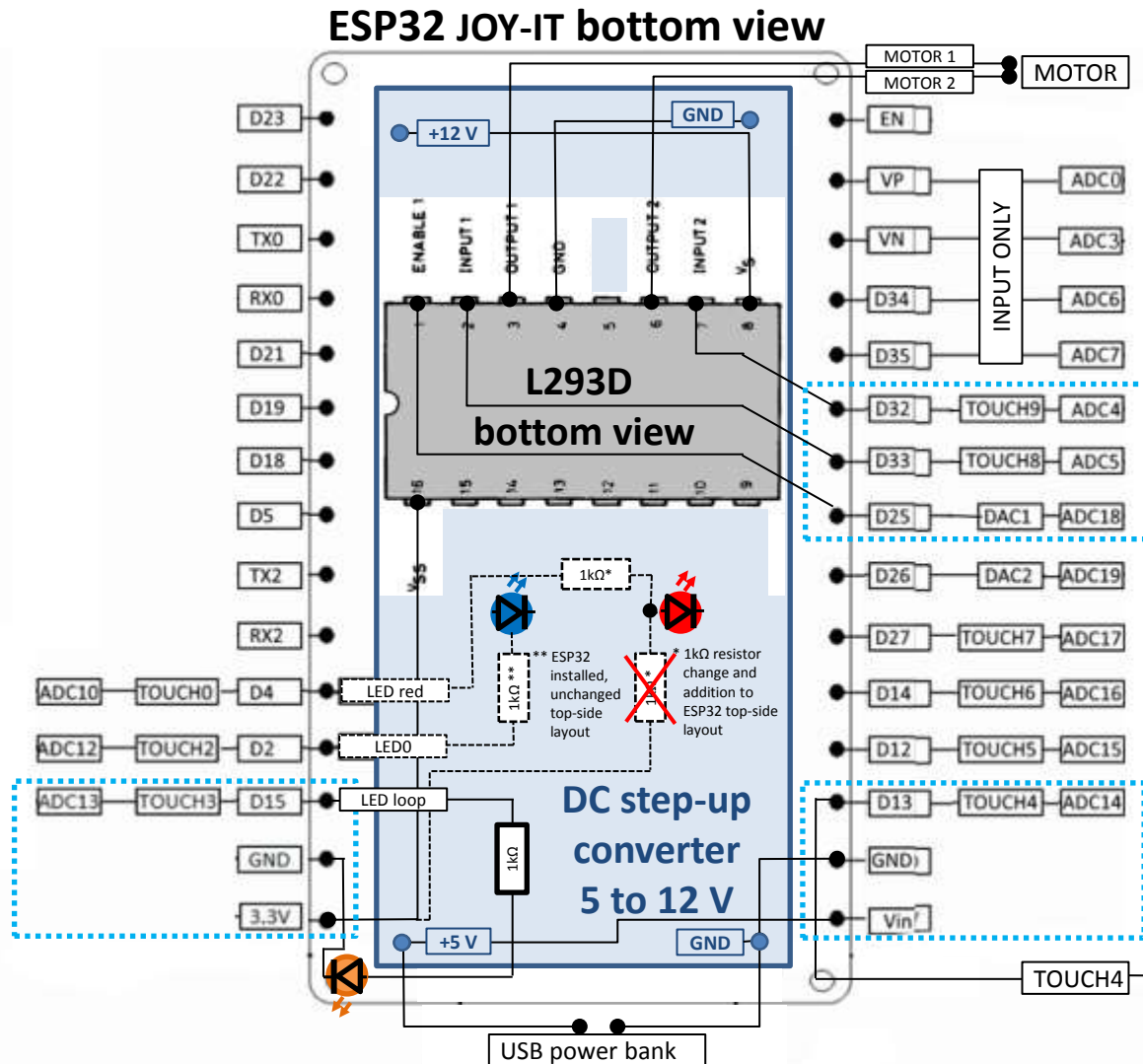
If more than one train is running, cut internal connection to rail power.

Alternative implementation with  
motor IC L293D



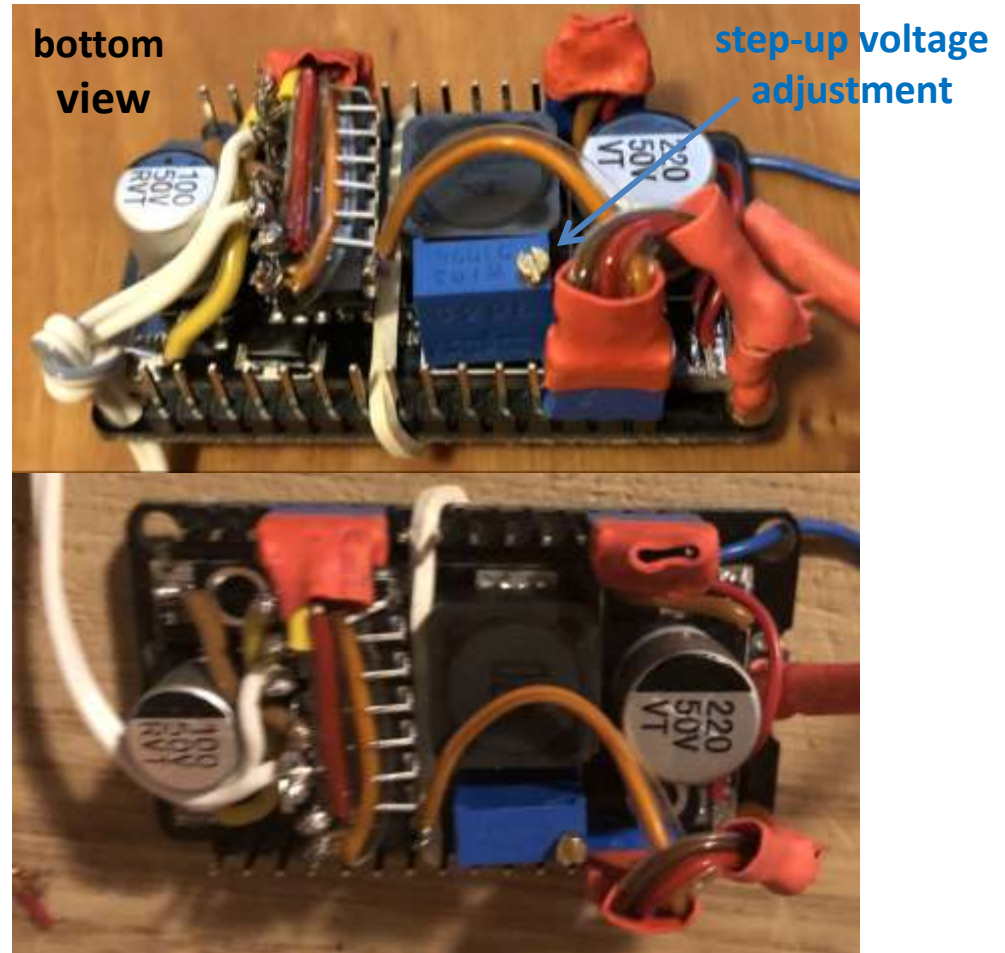
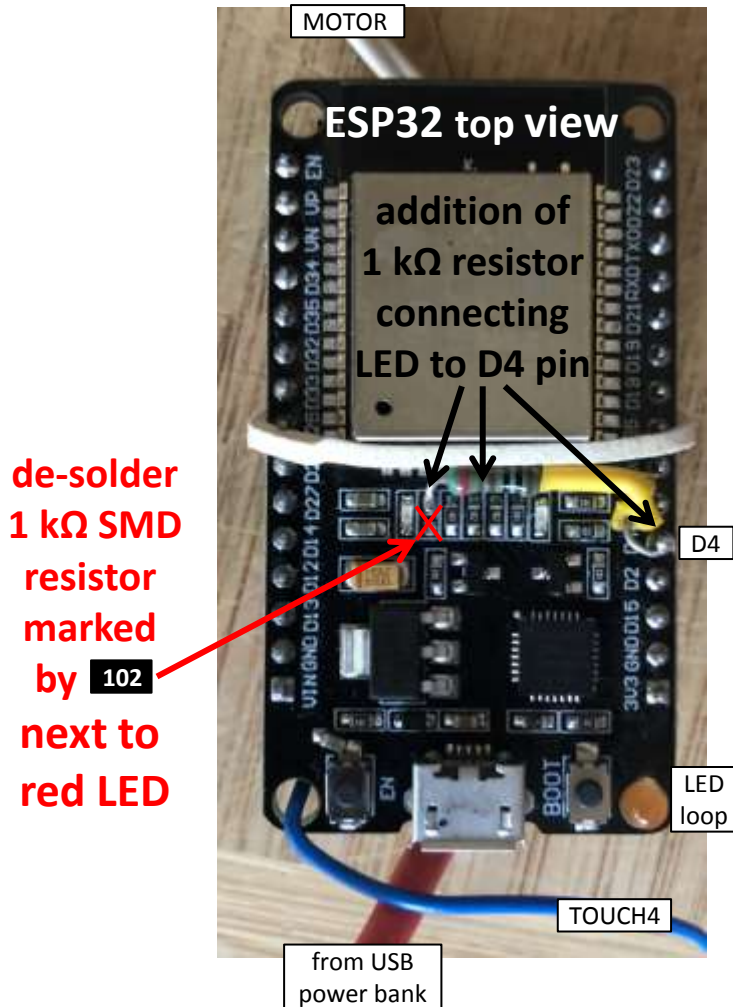
# ESP32 & L293D

## integrated mounting and wiring



# ESP32 & L293D

## integrated mounting and wiring





# Option 1: Power measurement

by reading current and voltage on measuring resistors

# PiedPiperS-Bauteile (Option 1)

## Materialbedarf für einen Zug mit WLAN-Steuerung

### Messwiderstände

Zur Beobachtung der Motorleistung können der effektive Motorstrom und die Versorgungsspannung an zwei Analogen Eingängen des Mikrocontrollers während der Fahrt gemessen werden. Die Messwiderstände sollten dafür eine niedrige Toleranz aufweisen, z.B. Metallschichtwiderstände. Der  $1,00\ \Omega$  Messwiderstand für den Motorstrom muss hinsichtlich seiner Leistung (in Watt) an die maximale Stromstärke (in Ampere) angepasst werden.

- $1,00\ \Omega$  Messwiderstand für die Strommessung, 1 % Toleranz, 1 W ist geeignet für übliche H0-Modellbahnlokomotiven mit weniger als 1 A maximaler Stromaufnahme

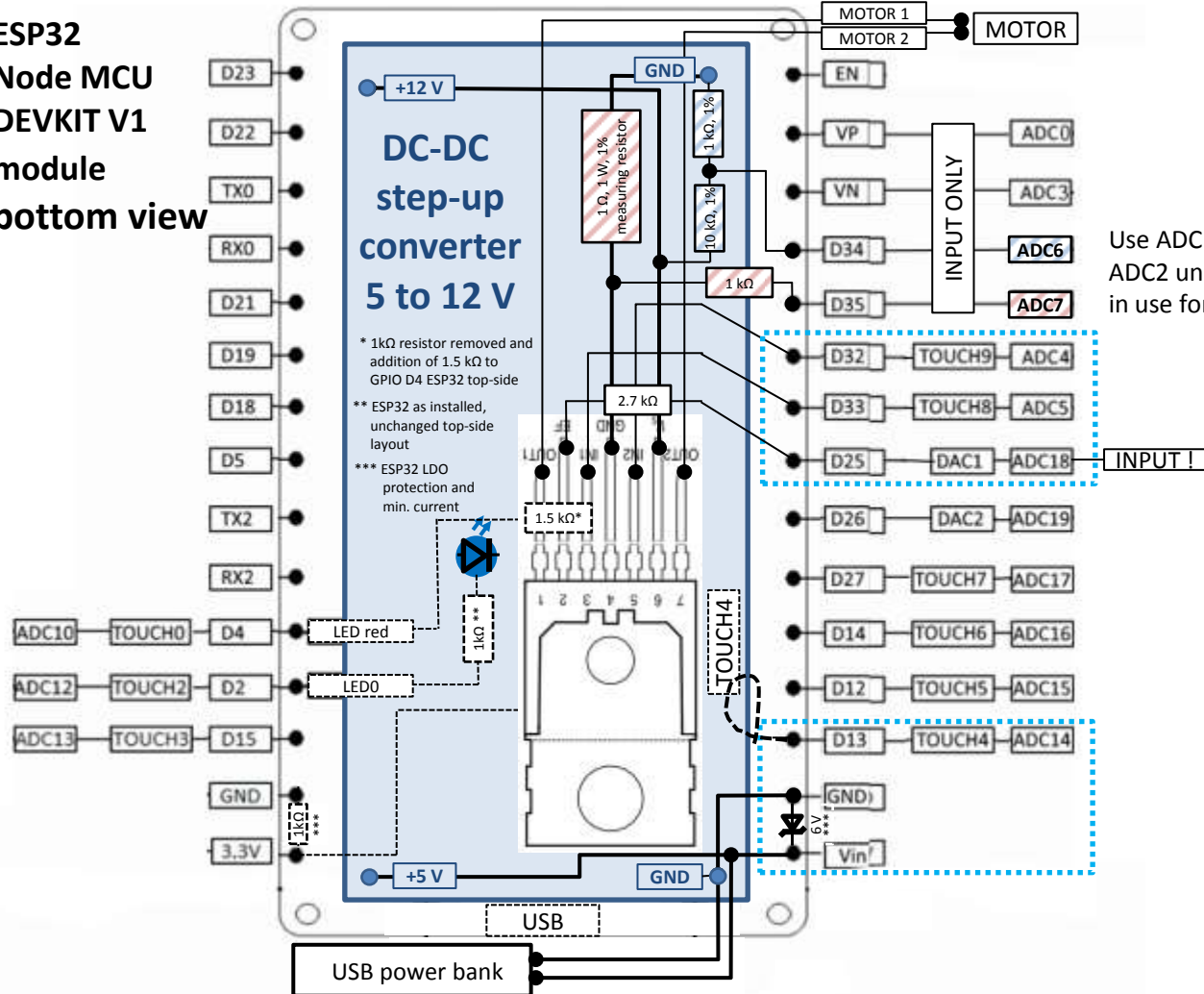
### Für 1:11 Spannungsteiler

- $1,00\ \text{k}\Omega$  Messwiderstand, 1 % Toleranz, 0,5 W
- $10,0\ \text{k}\Omega$  Messwiderstand, 1 % Toleranz, 0,5 W
- $1\ \text{k}\Omega$  Widerstand, 0,25 W, zum Schutz des analogen Messeingangs des Mikrocontrollers vor Spannungsspitzen aus der Strommessung im Leistungsteil der Motorsteuerung
- Schrumpfschlauch, sehr dünn, zur teilweisen Isolierung der Anschlussdrähte der Widerstände

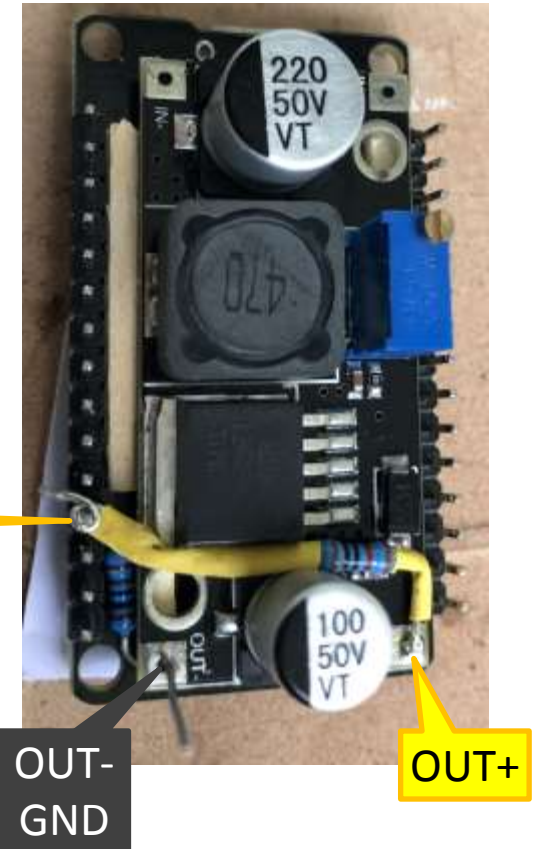
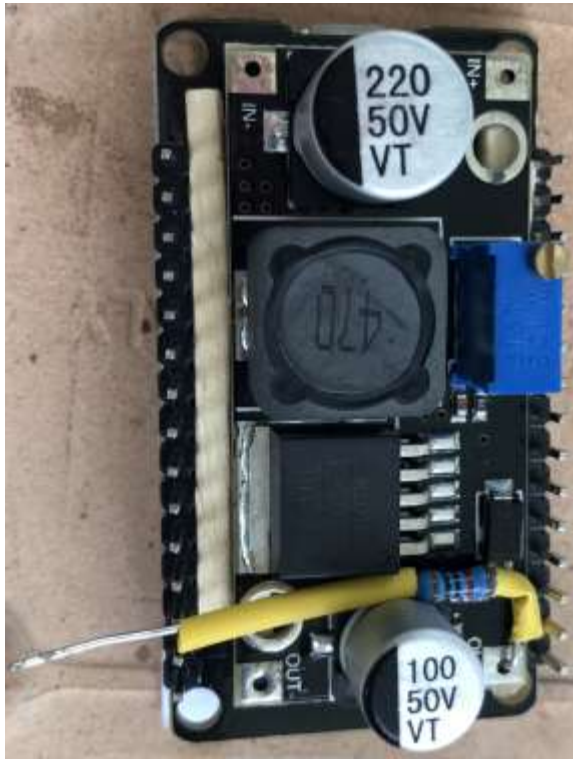
# ESP32 & TLE5206-2S

integrated mounting and wiring with LDO protection,  
v173 including optional measuring resistors for power data reading

ESP32  
Node MCU  
DEVKIT V1  
module  
bottom view



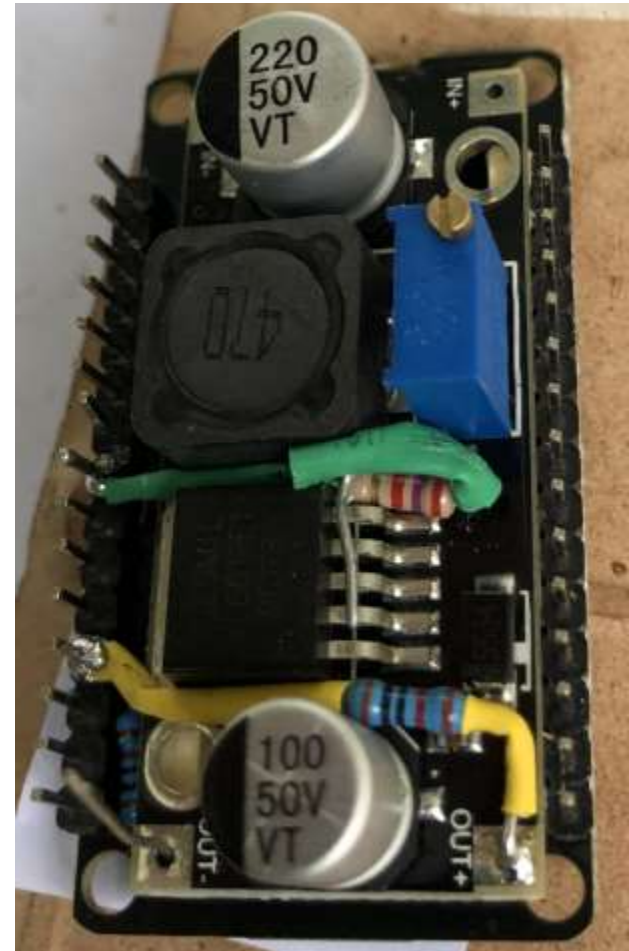
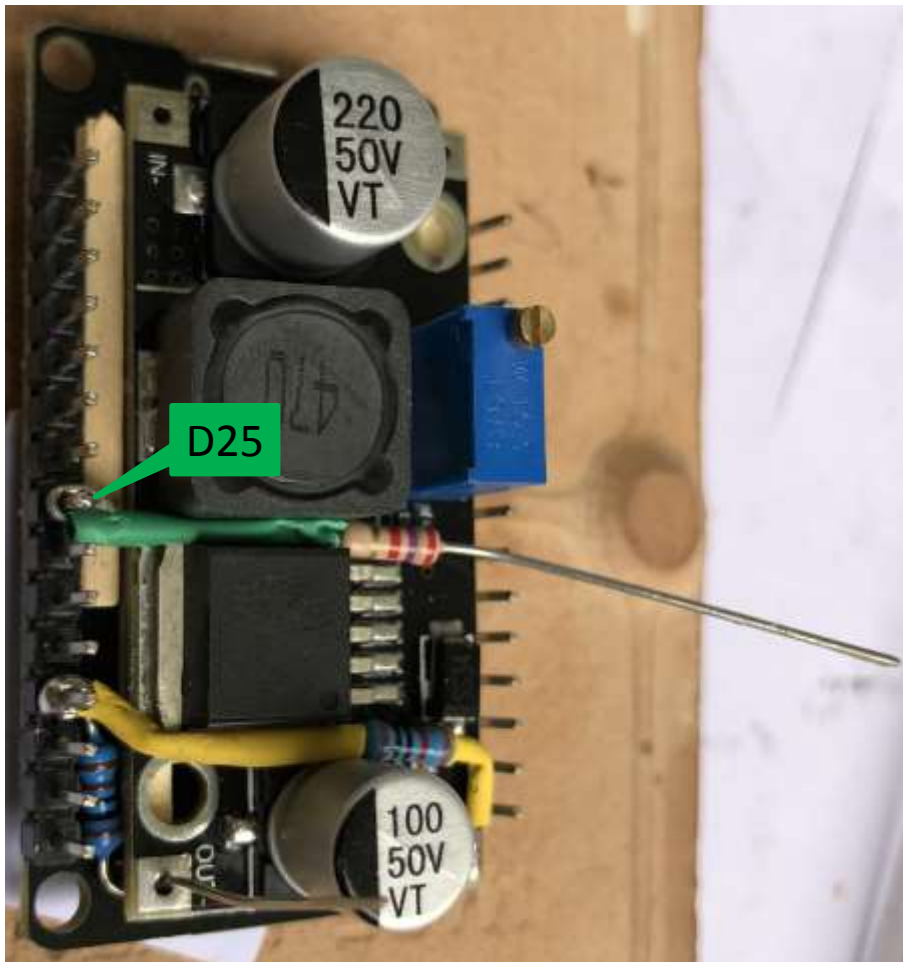
Voltage measurement from divider between OUT+ and OUT- of step-up-converter consisting of 10.0 k $\Omega$  and 1.00 k $\Omega$  measuring resistors (1% tolerance), connected to GPIO pin D34



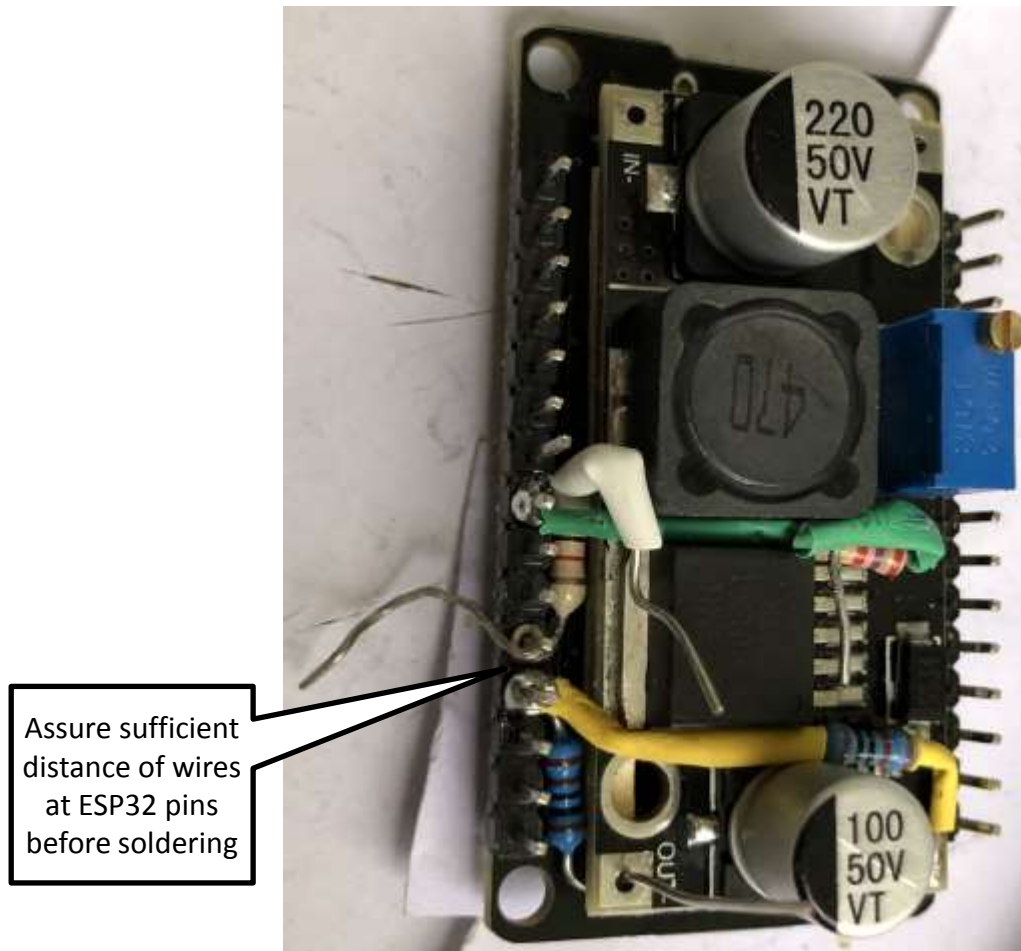
Isolation with 1 mm heat shrink tubing



# Mounting 2.7 k $\Omega$ resistor for motor IC error flag EF as input to GPIO pin D25

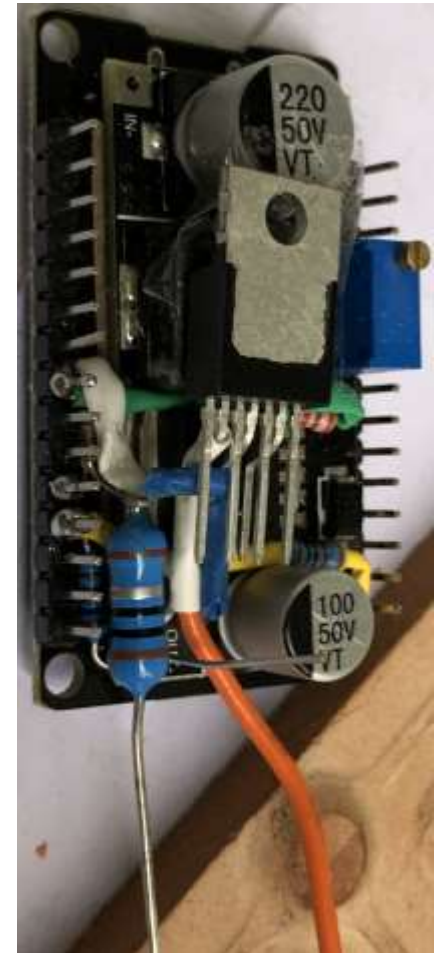
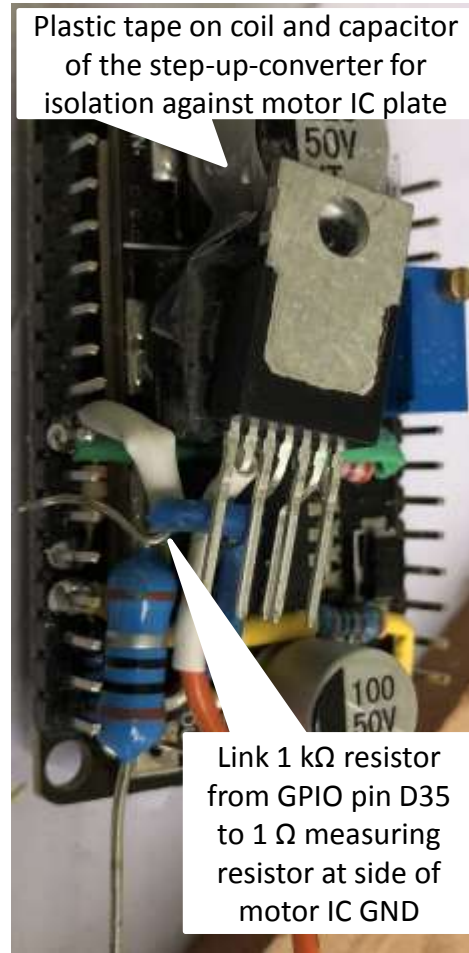
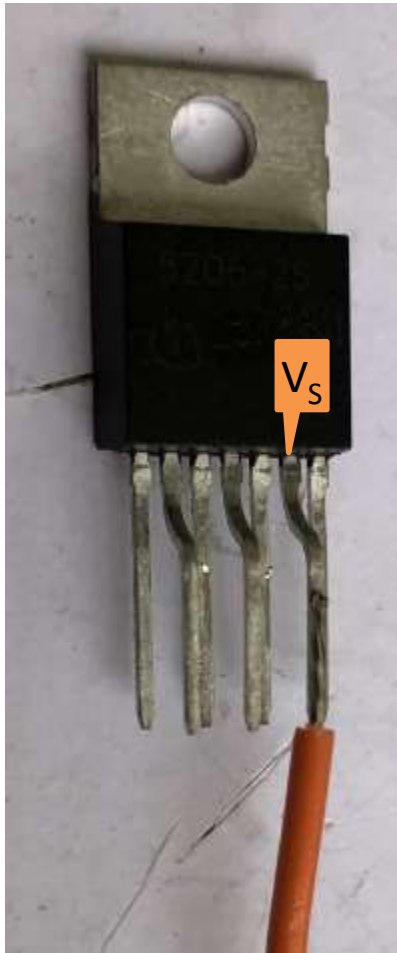


# 1 k $\Omega$ resistor for transfer of motor current measurement to GPIO pin D35

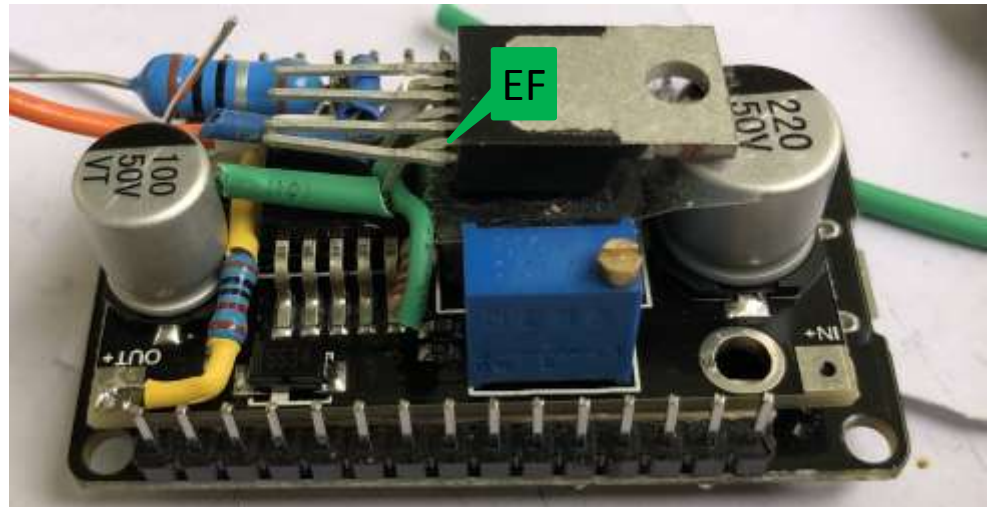
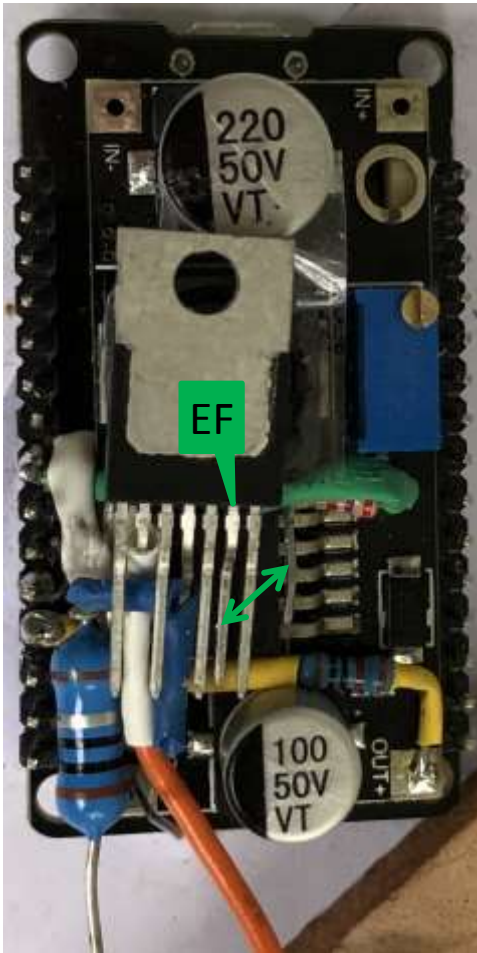


# Motor IC power connection

$V_S$  to +12...16 Vdc at OUT+ and  
GND via 1  $\Omega$ , 1 W current measuring resistor to OUT-



Connect 2.7 k resistor to motor IC EF



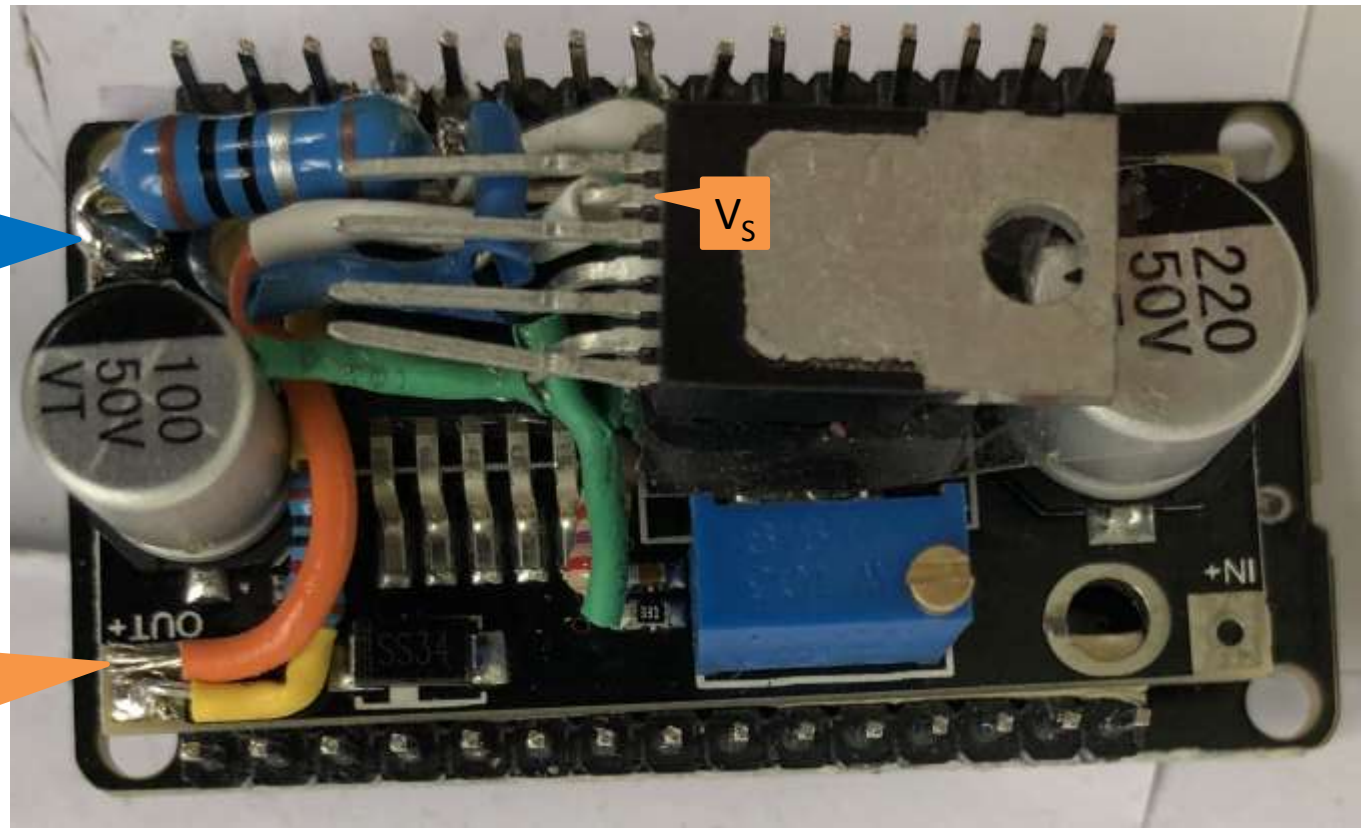
Isolate the soldered link with heat shrink tubing



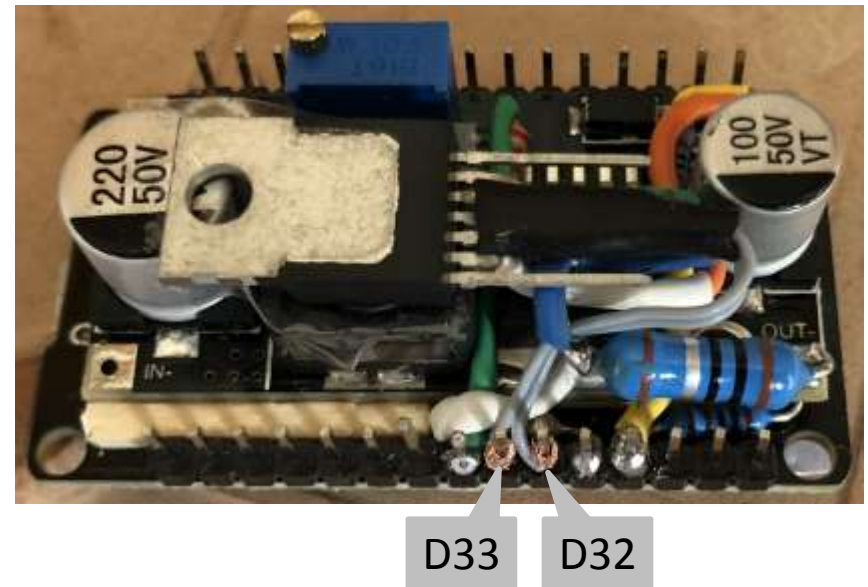
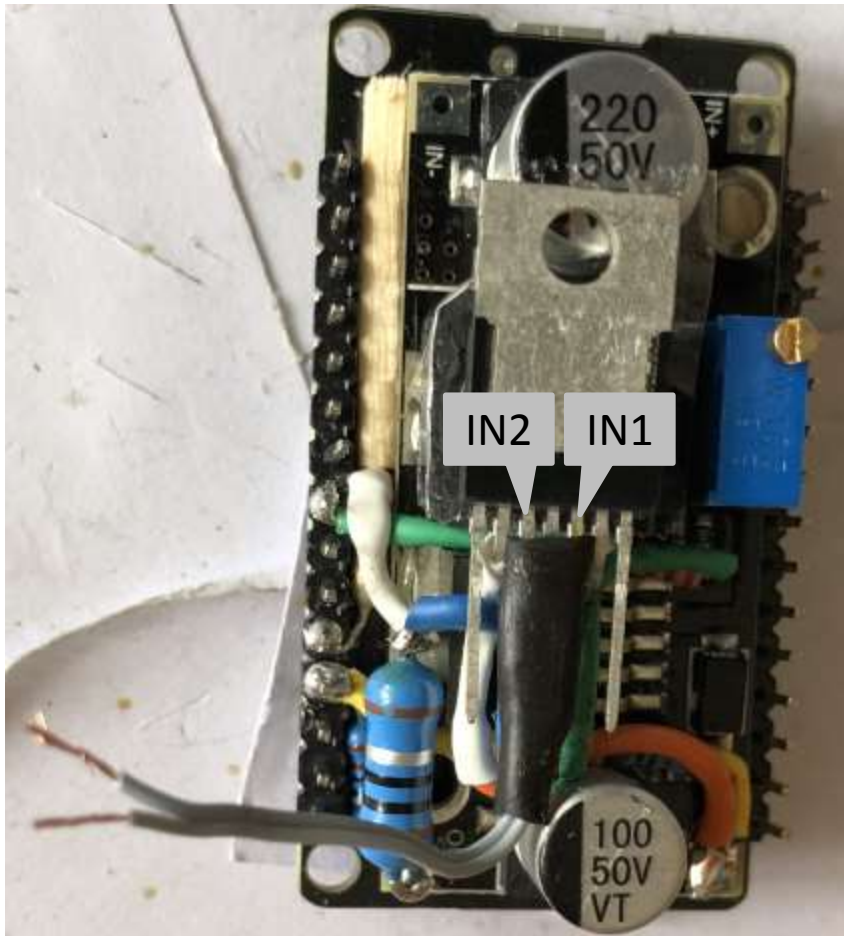
# Connect motor IC to step-up converter output

Solder other end of  $1\ \Omega$  resistor to GND = OUT-

Solder  $V_S$  cable to 12...16 Vdc OUT+

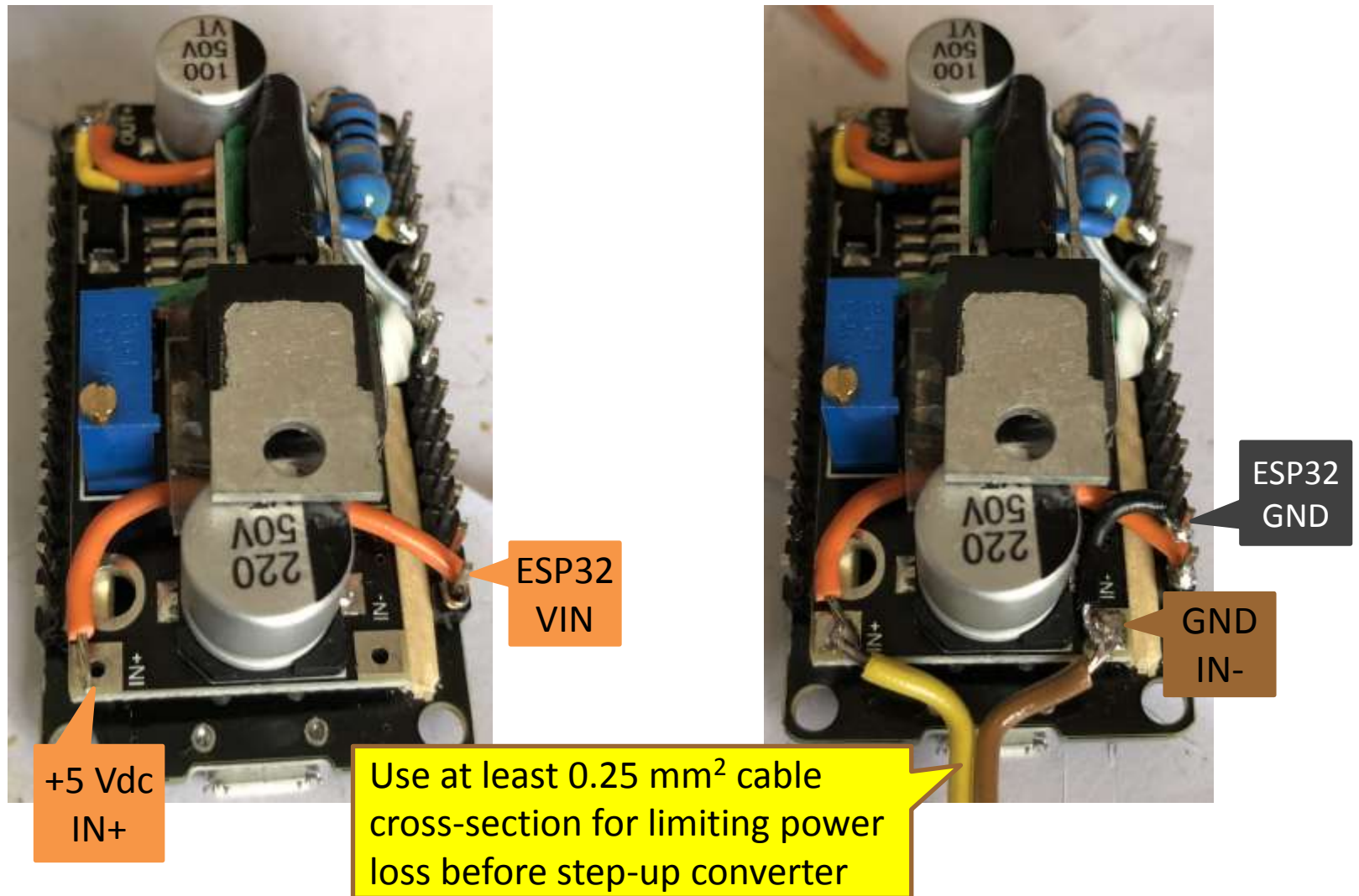


# Connect IN1 and IN2 to GPIO pins D33 and D32



Add appropriate insulation for motor IC input pins with shrink tubing

Connect 5 Vdc power input of ESP32 and step-up converter to short USB power supply cable.



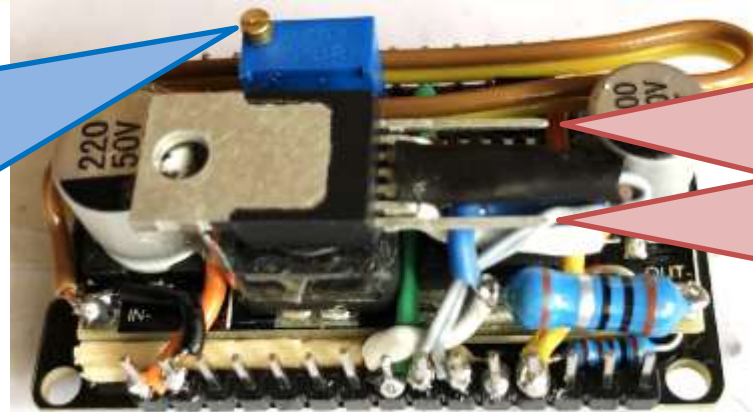


# Install short USB A plug with insulation for connection to USB power bank



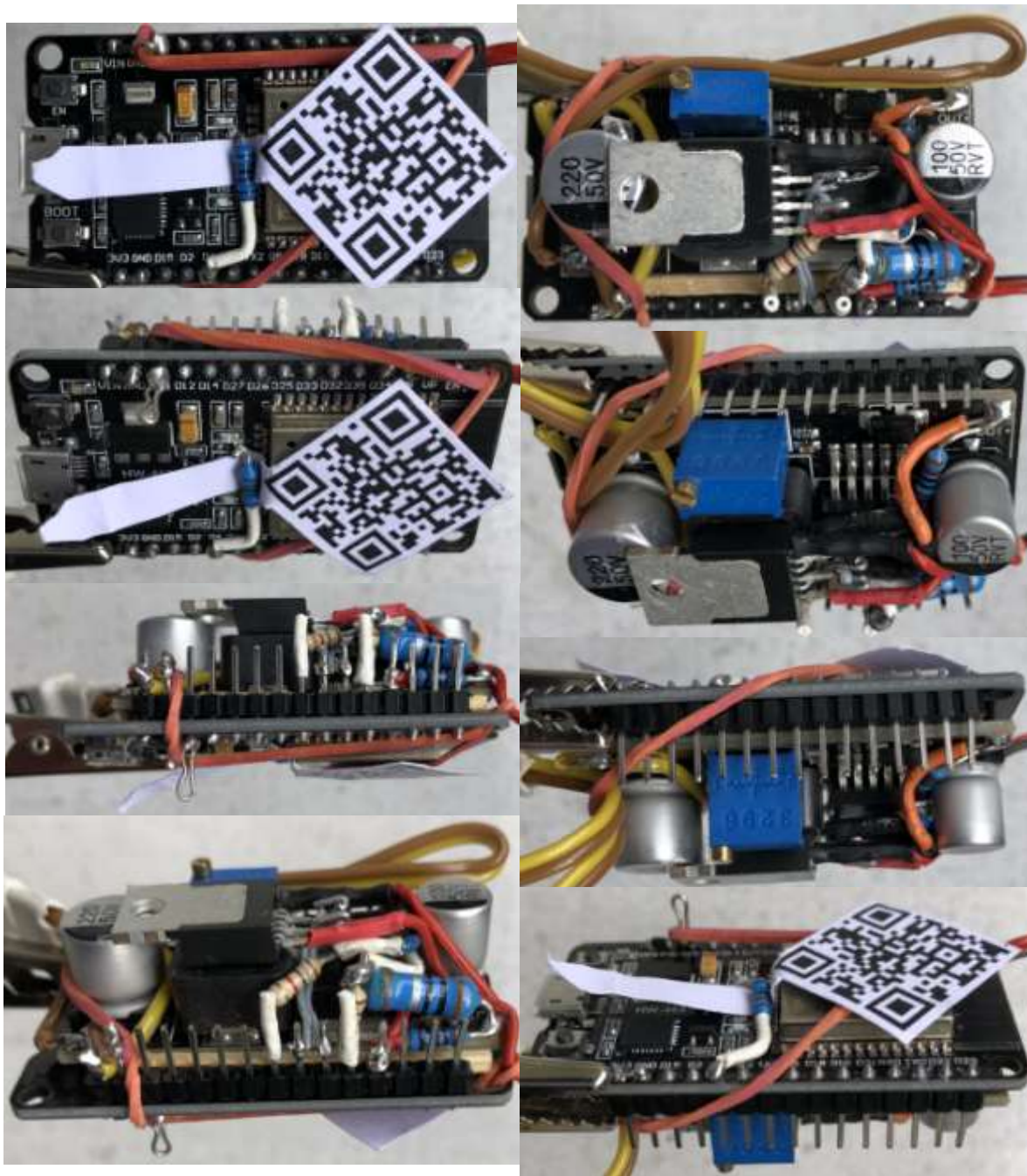
The step-up converter output voltage 12 ... 16 Vdc can be adjusted and fine tuned at the potentiometer screw.

Measure the output voltage with multimeter. Correspondingly, set the voltage in the lok.ini file before USB data upload to the ESP32.



Attach thin power cable with mini plug connector from OUT1 and OUT2 of motor IC to locomotive.

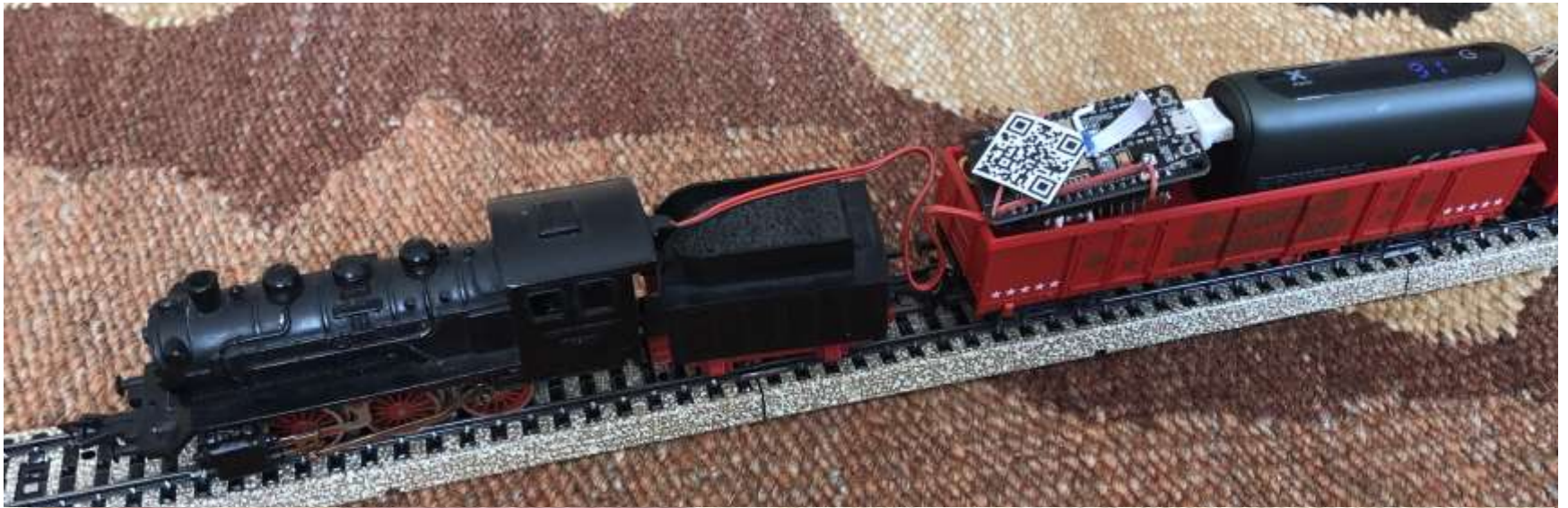




# Fixing cable and electrical connection to locomotive motor







# Option 2: Speed measurement

by counting railway sleepers with reflective IR sensor

Developed in [PiedPiperS branch](#)  
['speedo', version 191 to 244](#)



# Geschwindigkeitsmessung (Option 2)

Für eine Geschwindigkeitsmessung vom fahrenden Zug aus (d.h. kontinuierlich und online) werden die schwarzen oder dunklen Schienenschwellen der Modellbahnschienen beim Überfahren durch einen Infrarot-Sensor erfasst. Die Schienenschwellen haben gleichmäßige Abstände. Aus der Zeit, die der Zug bis zur nächsten Schwelle benötigt, und der Frequenz des Überfahrens kann dann die **wahre Geschwindigkeit der Fahrt über die Schiene in Meter pro Sekunde (m/s)** bestimmt werden. Die Geschwindigkeit wird regelmäßig gemessen, **in km/h entsprechend dem Modellmaßstab** umgerechnet und in der Loksteuerung online angezeigt. Außerdem können die Ergebnisse der Geschwindigkeitsmessung in einem Diagramm auf der zusätzlichen Webseite [speed.html](http://speed.html) verfolgt werden. Die Daten werden hierzu jede Sekunde über *get /speeddata* vom ESP32-Webserver angefordert.

# Geschwindigkeitsmessung (Option 2)

## Elektronik-Bauteile

Für diese Geschwindigkeitsmessung werden wenige zusätzliche Bauteile und eine sehr einfache elektrische Schaltung gebraucht. Eine **Infrarot-(IR)-Reflexlichtschranke dient als Sensor zur Erkennung der Schienenschwellen** unter dem Waagen. In der elektrischen Schaltung wird die Signalspannung für den Microprozessor verstärkt. Die unten abgebildete Schaltung ist in den [Projektdokumenten](#) beschrieben und besteht aus folgenden Bauteilen:

- Infrarot-(IR)-Reflexlichtschranke und Näherungssensor für ca. 5 mm Abstand, z.B. Typ HSDL-9100, mit IR-LED als Sender und IR-Diode als Empfänger
- 220  $\Omega$  Vorwiderstand für die Einstellung des Stroms durch die IR-LED  $I_F$  bei 10 mA
- Signalverstärker aus einfachem NPN-Kleintransistor mit hoher Verstärkung, z.B. BC 238C, BC 548C, BC 547B, BC 848C, BC 846B
- 33 k $\Omega$  Pull-up-Widerstand für die Erzeugung des analogen Eingangssignals. Der Widerstand kann zwischen 27 k $\Omega$  und 47 k $\Omega$  so gewählt werden, so dass sich eine gute Erkennung der dunklen Schwellen ergibt.

Geschätzte zusätzliche Materialkosten für Elektronik ca. 3 €. Die **Bauanleitung mit Fotos** für das PiedPiperS-Projekt liegt im Internet unter <https://github.com/jorail>

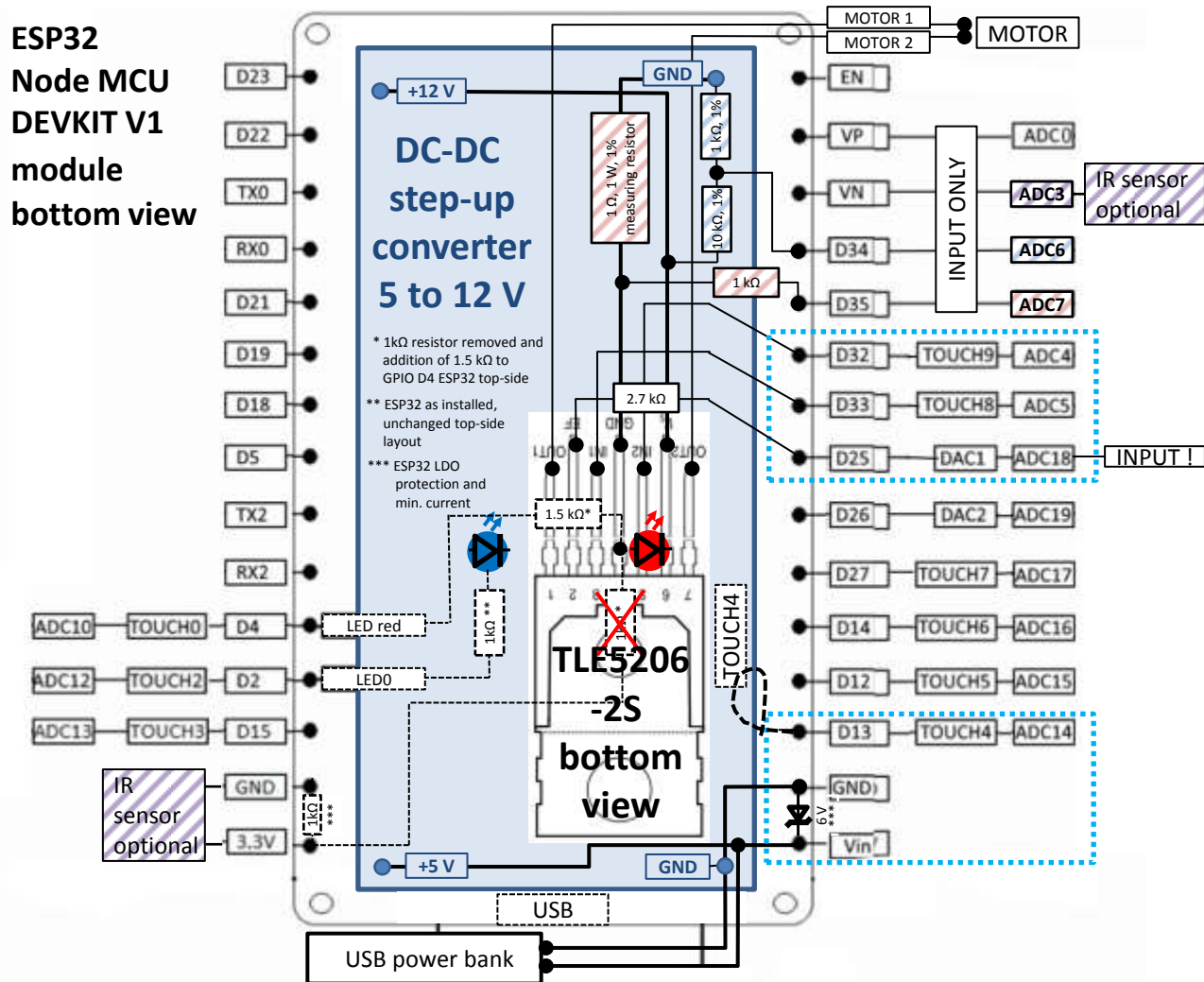
# Geschwindigkeitsmessung (Option 2)

Die verstärkte Signalspannung wird über den analogen Eingang am Pin GPIO 39 und den Analog-Digital-Wandler ADC1\_CH3 des Microprozessors erfasst. Die laufende Erkennung der Schienenschwellen im PiedPiperS-Programm kann unmittelbar durch die blaue LED auf dem Microprozessor angezeigt werden.

Um eine Mehrfachzählung einzelner Schwellen bei kontinuierlich zu- und abnehmender Signalspannung zu verhindern, ist zwischen dem niedrigen Spannungsniveau ( $< 0,5\text{ V}$ , heller Grund, reflektierend, keine Bahnschwelle) und dem hohen Spannungsniveau (ca.  $1,5\text{ V}$ ,  $> 0,8\text{ V}$ , dunkle Bahnschwelle, Transistor sperrt) ein ausreichend großes Totband (hier z.B.  $0,3\text{ V}$  zwischen dem niedrigen und hohen Spannungsniveau) als Verzögerung (Hysterese) erforderlich. Letztlich müssen die Erkennung der Bahnschwellen in der Praxis entsprechend der Eigenschaften des Modellbahngleises und IR-Sensors erprobt und der Pull-up-Widerstand und die Grenzwerte für die Signalerkennung dazu passend eingestellt werden. Die Grenzwerte des niedrigen und hohen Spannungsniveaus können bei Bedarf ähnlich wie die Motorspannung in der [lok.ini-Datei](#) angepasst werden.

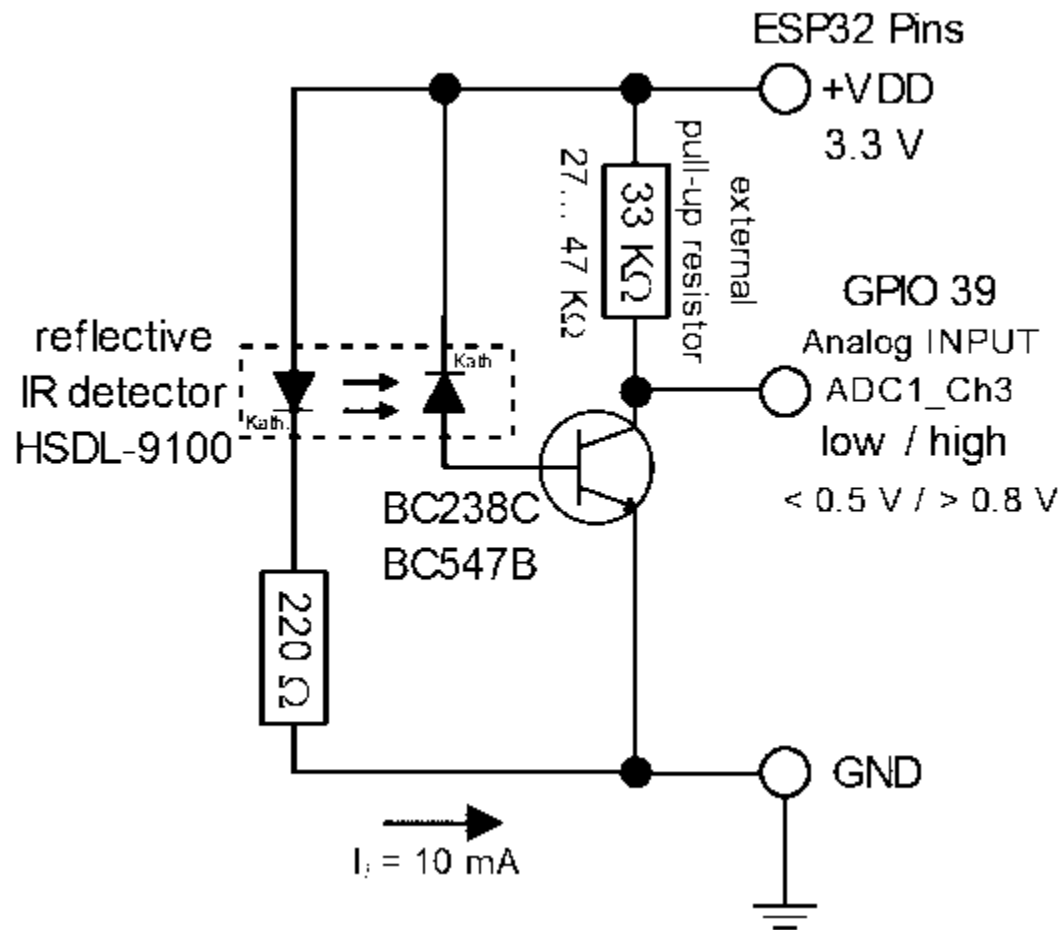
# ESP32 & TLE5206-2S

integrated mounting and wiring with LDO protection, incl. options  
for power data reading & speed measurement by reflective IR sensor



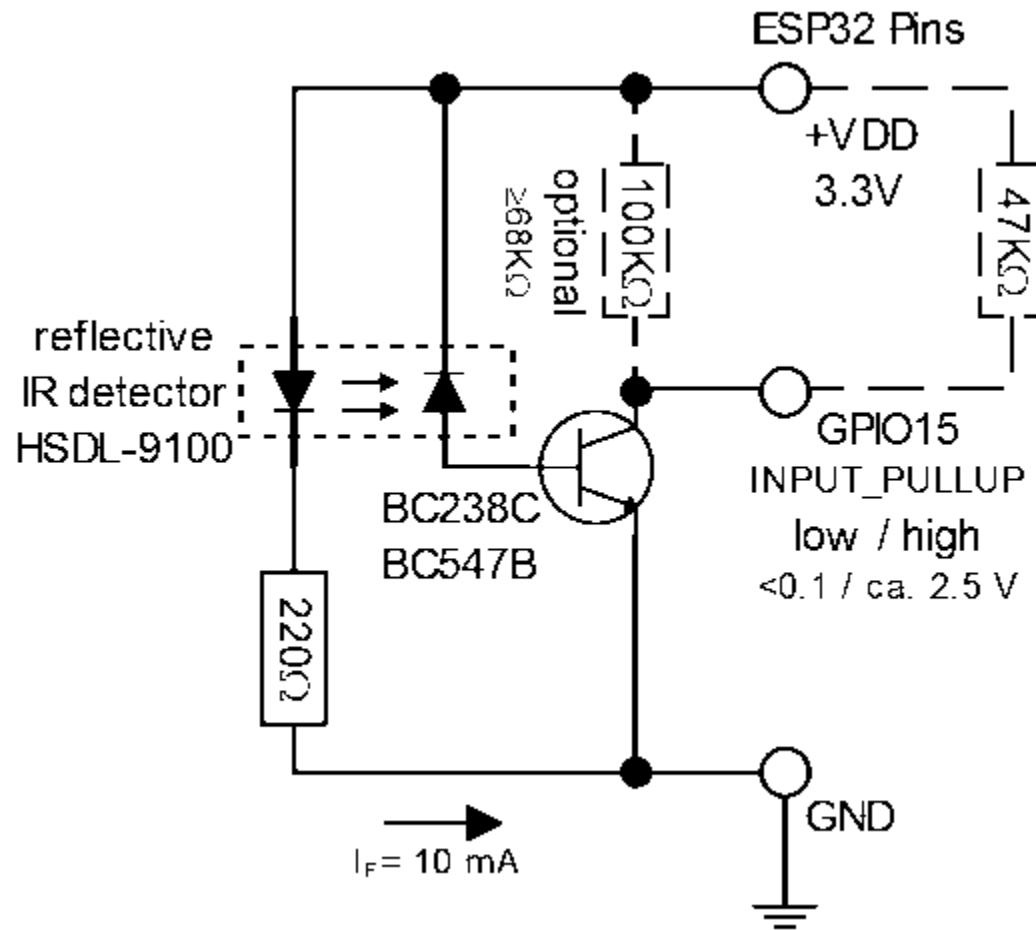


# Reflective IR sensor



# Reflective IR sensor

## alternative ESP32 using internal pullup resistor



# Reflective IR sensor mounting and preliminary circuit setup on breadboard

