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 Acess Point and Webserver**
- **No proprietary elements, construction time 1 weekend, material cost ca. 25 €**

PiedPiperS Version 244, J. Ruppert 2021-08-26, GNU GPL v3 GitHub: https://github.com/jorail/PiedPiperS

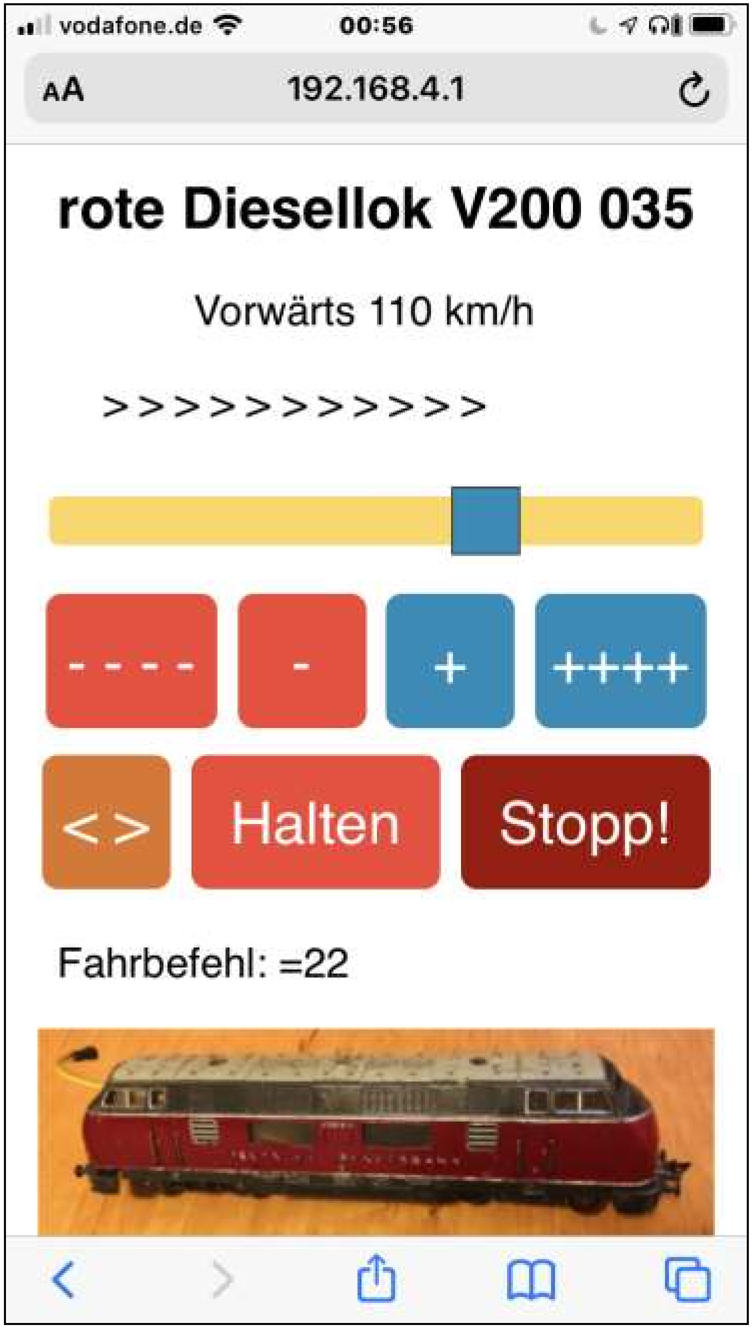
Train control website on smartphone

- Compatible with every smartphone with WIFI
- ESP32 WIFI access point mounted on train
- Browser access to control panel via Webserver
- Stable remote controle 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 protaction

- Option for online power measurement and data display in tables and charts

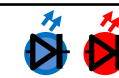
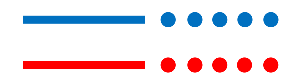
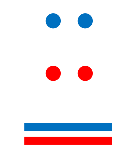
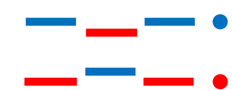
- Option for true speed measurement by IR sensor detection of railway sleepers



PiedPiperS touch commands

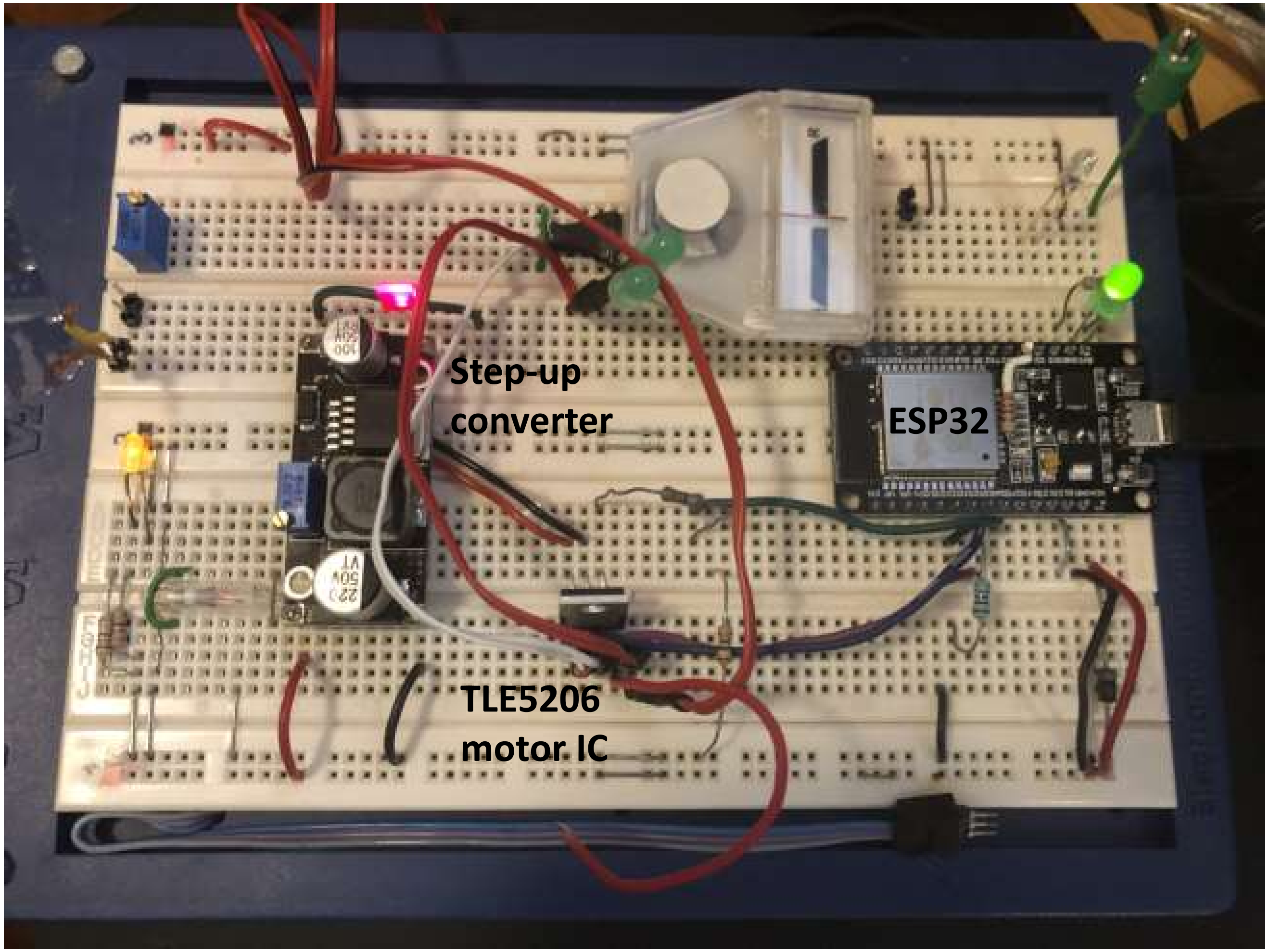
Touch code	Command for speed level	Meaning
•	--	decrease speed
••	--------	slow down
•••	--------------------------------	break to halt
••••	0	fast brake and stop
•••••	00<	fast stop, reverse direction
• —	++	increase speed
• — —	++++++++	speed up
• — — —	++++++++++++++++	go fast
—	0	fast brake and stop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

PiedPiper LED indicators

	LEDs 
Information on motor direction and speedlevel: • forward = 1 = green LED + 0 to 16 short LED flashes • backward = 0 = red LED + 0 to 16 short LED flashes	
Adjustment of motor speed: • 2 green LED flash = speed level + 2; • 2 red LED flash = speed level - 2; • green & red LEDs long flash = fast brake, speed level = 0	
Change of motor direction: • green/red/green + green LED flash = forward = 1 • red/green/red + red LED flash = backward = 0	
Program running and main loop frequency: • orange flash after 5,000 main loop cycles, 200 times monitoring for input activity and up to 5 speed adjustments	

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, 2nd for replacement when 1st 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 KΩ resistor from pin
  (or alternatively LEDs: 1 blue, 1 red, 3 mm diameter plus 2 x 1.5 KΩ resistors)
- 2.7 kΩ resistor, protection for digital input
- 1 kΩ 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

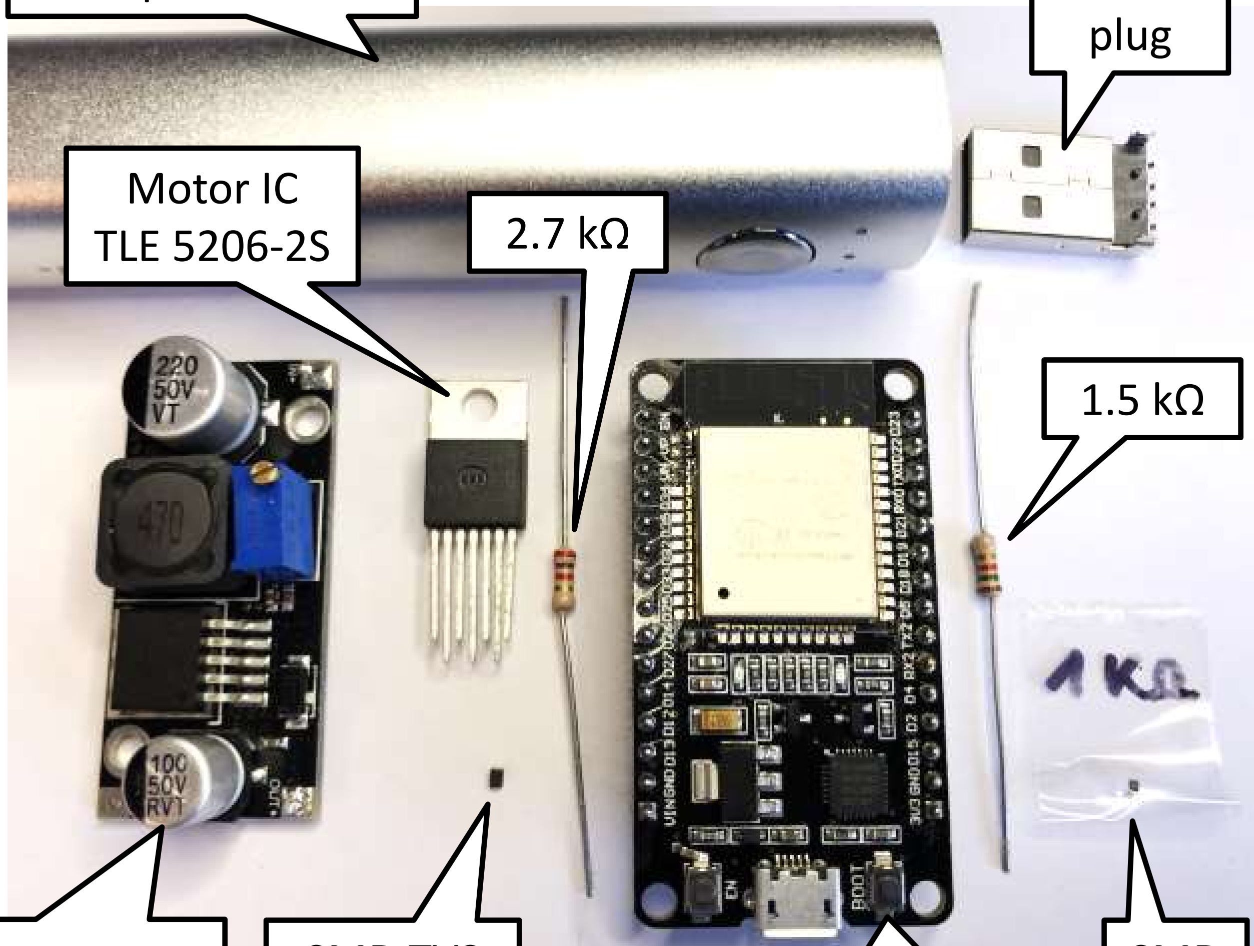
- ESP32 DevKit V1
  - 2 analog inputs for voltage and current readings
- Measuring resistors 1 Ω, 1 kΩ, 10 kΩ, tolerance 1%
- 1 kΩ resistor, protection for analog input

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

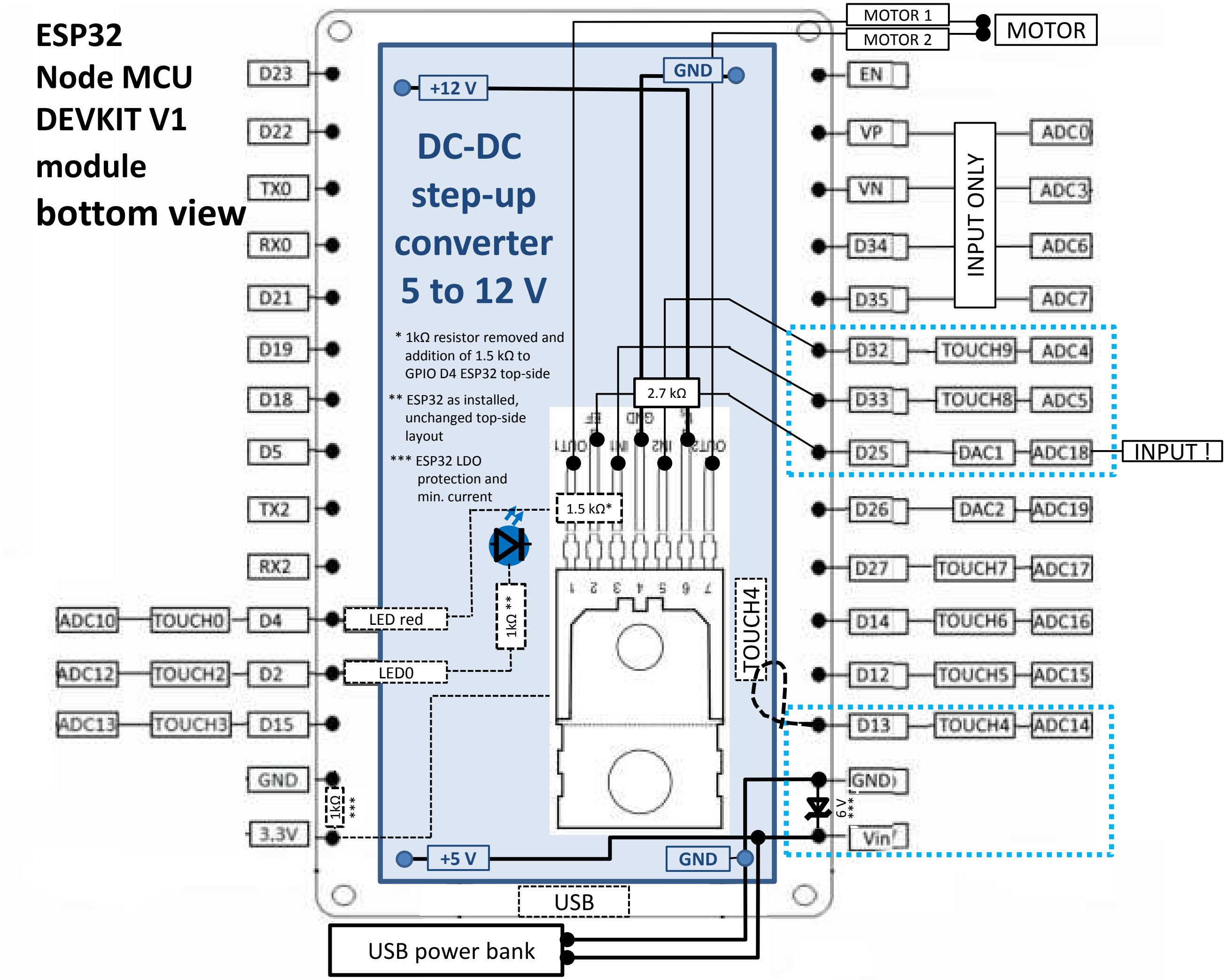
Estimated material cost ca. 25 € excl. other material, tools and optional additions

Main electronic parts



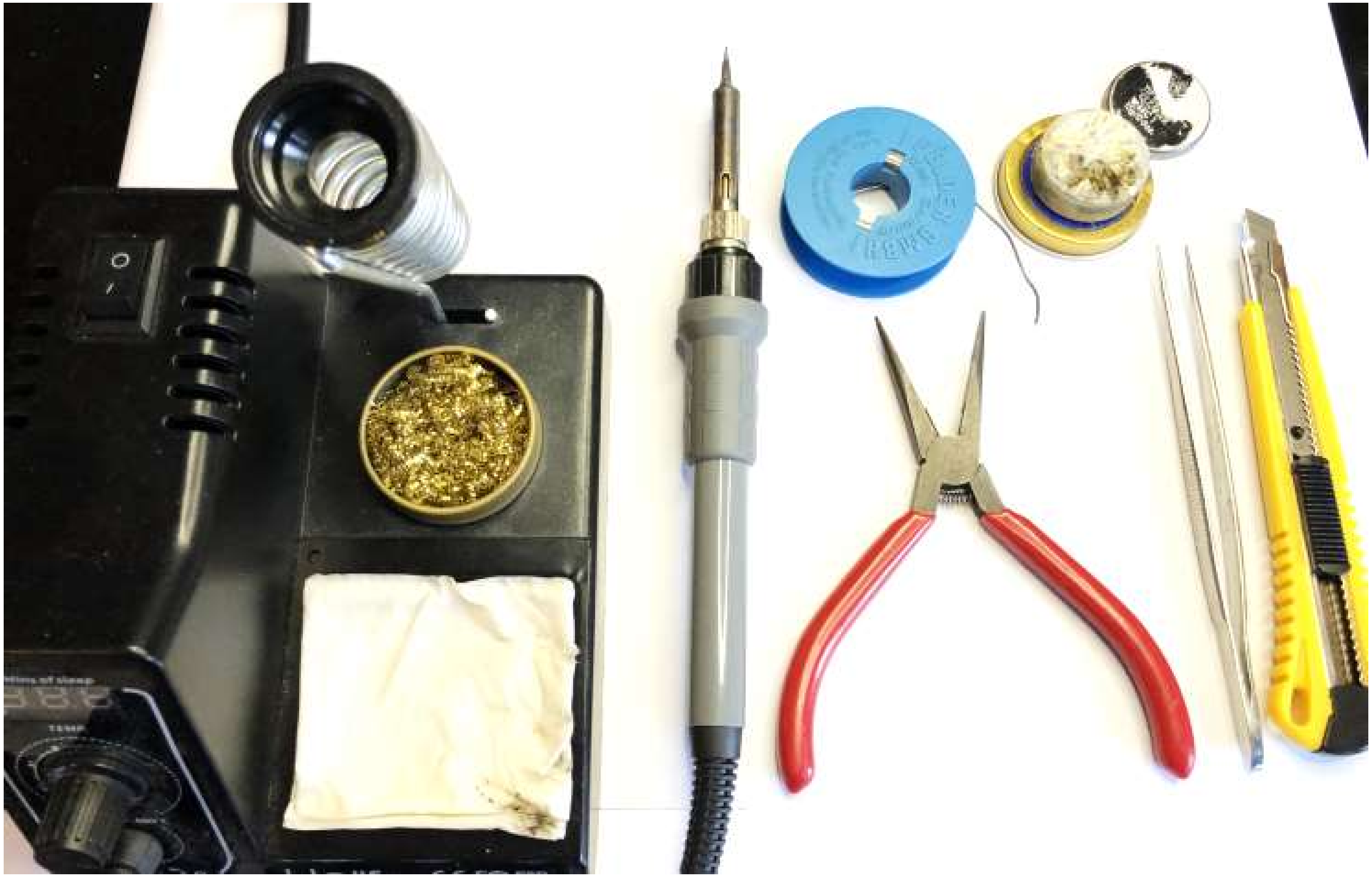
ESP32 & TLE5206-2S

integrated mounting and wiring with LDO protection, v100



PiedPiperS version 095, J. Ruppert 2021-03-21, GNU GPL v3 https://github.com/jorail/PiedPiperS

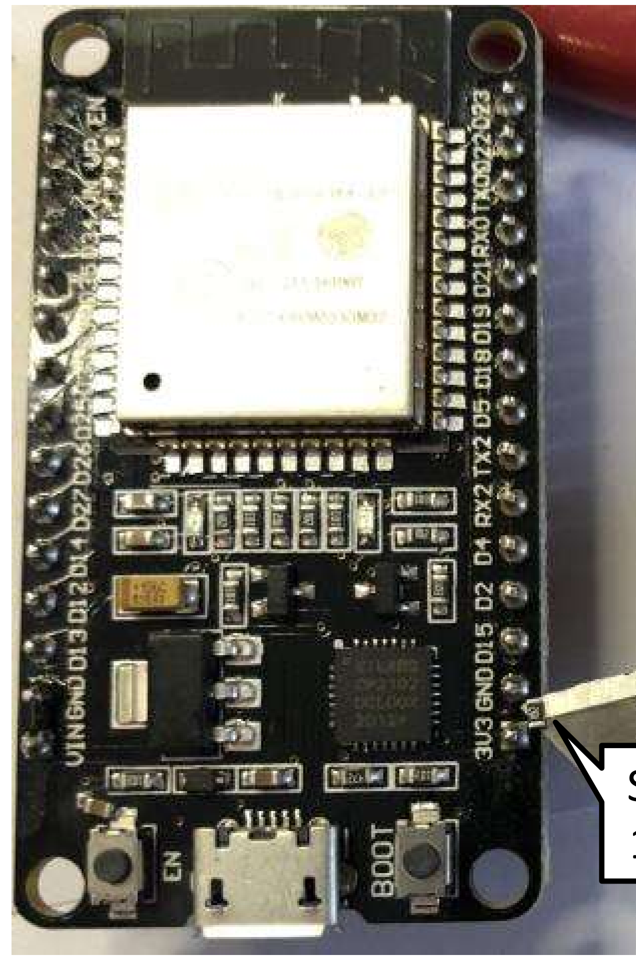
Required tools



Material for ESP32 topside adjustment

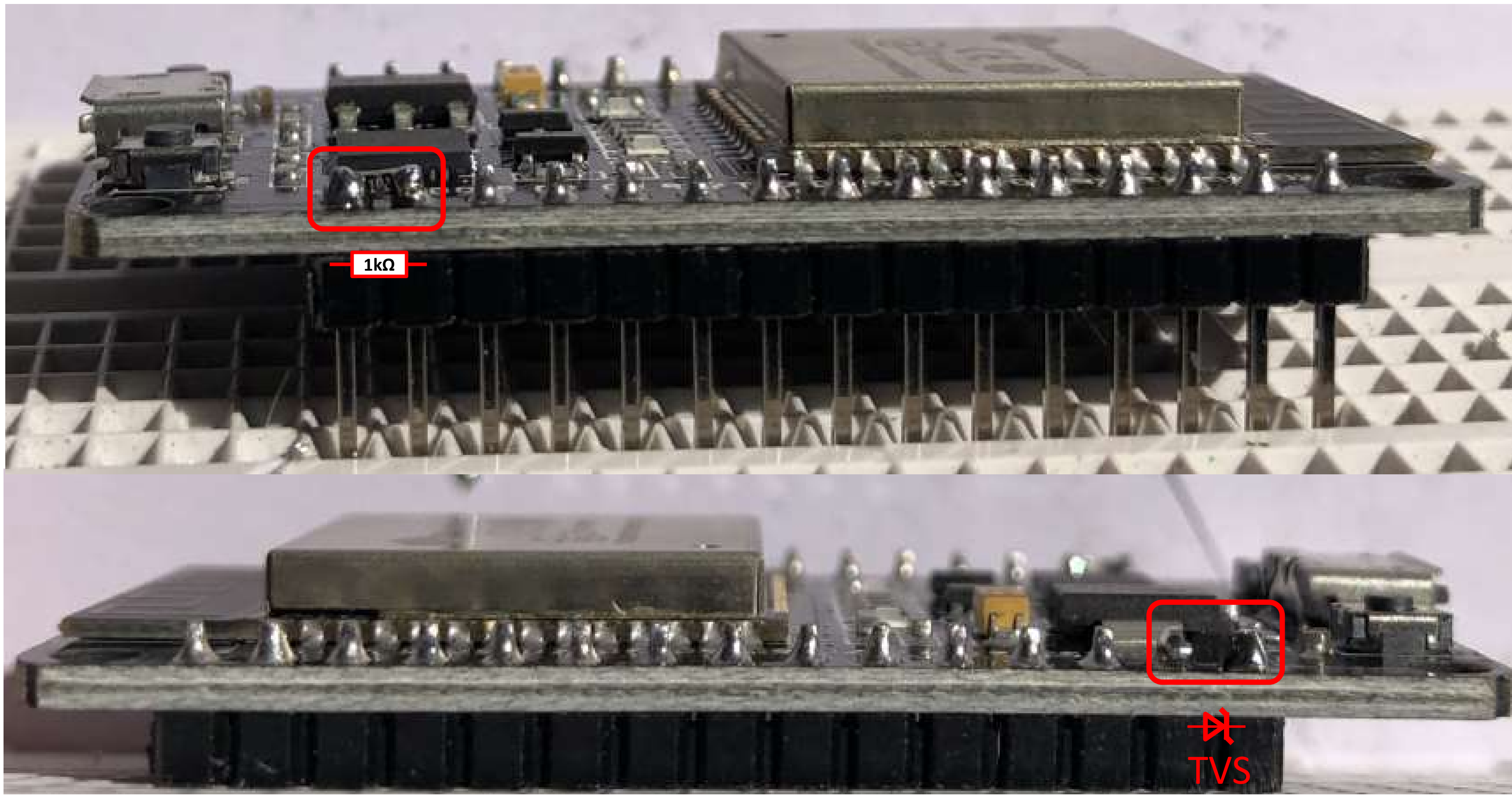
Solder SMD elements at 5 V input and 3.3 V output

SMD TVS diode 5 V

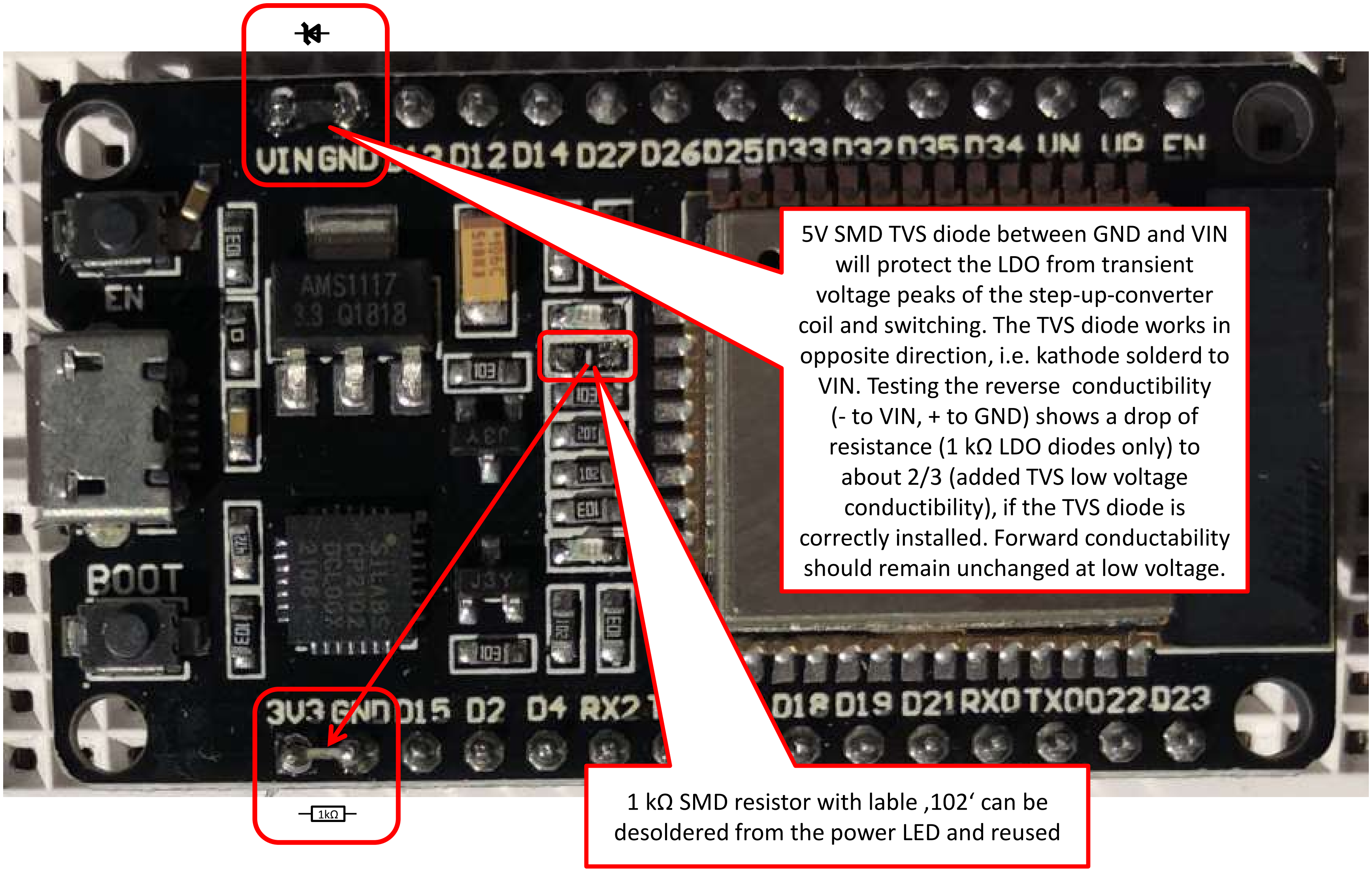


SMD 1 kΩ

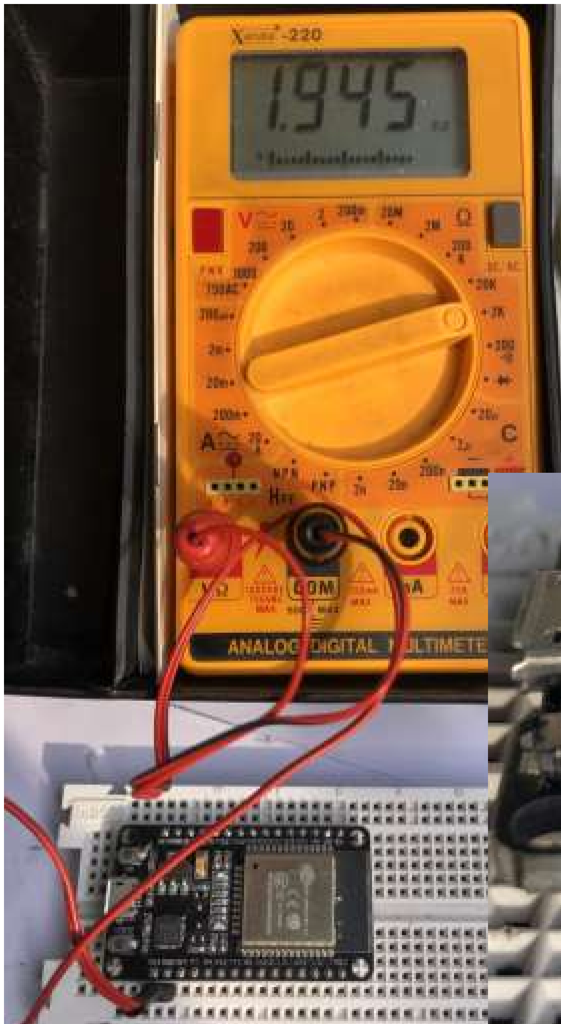
SMD resistor and TVS diode at power pins



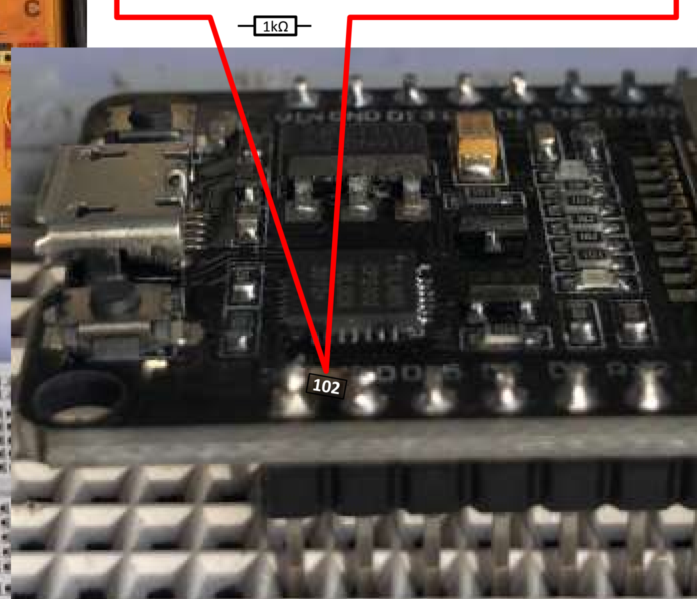
Protecting ESP32 for power supply from USB power bank linked to step-up-converter



Soldering 1 kΩ resistor assuring LDO load and protection

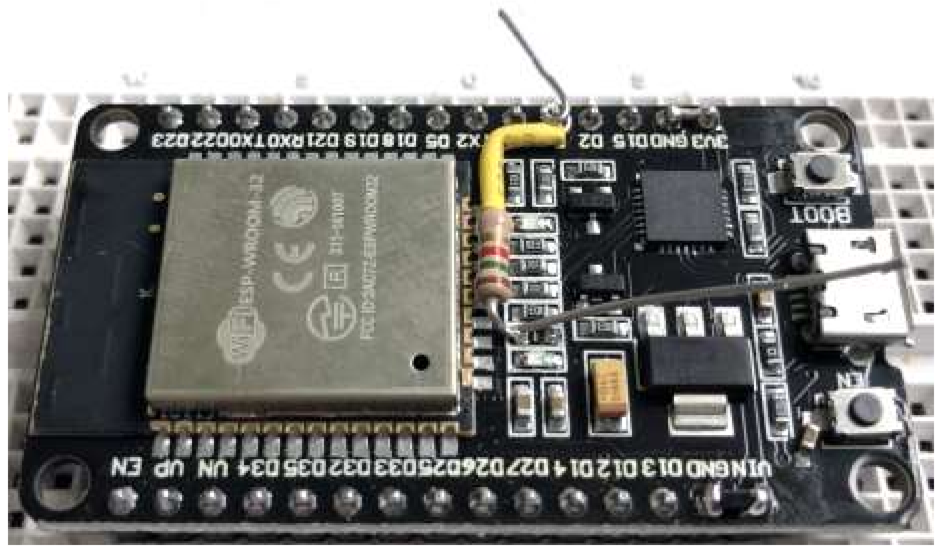


1 kΩ SMD resistor between 3.3V and GND assures a minimum current and load behind the low-dropout regulator (LDO) and its ability to regulate and protect the 3.3 V level from input voltage peaks in case of brownout. The initial total resistance of 2 kΩ between the 3.3 V and GND pins will drop to about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

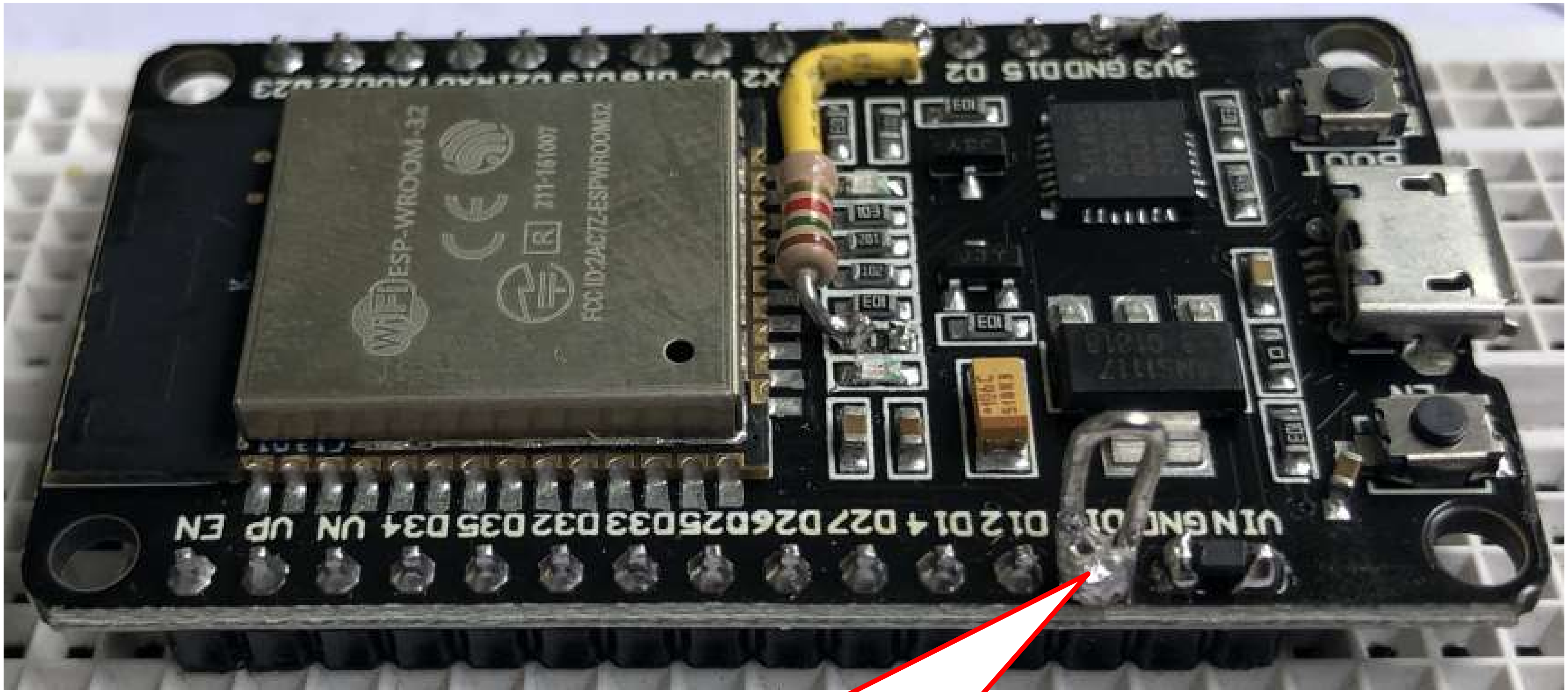


Connecting red LED to GPIO pin D4

1.5 kΩ resistor connects GPIO pin D4 to the de-solderd 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.

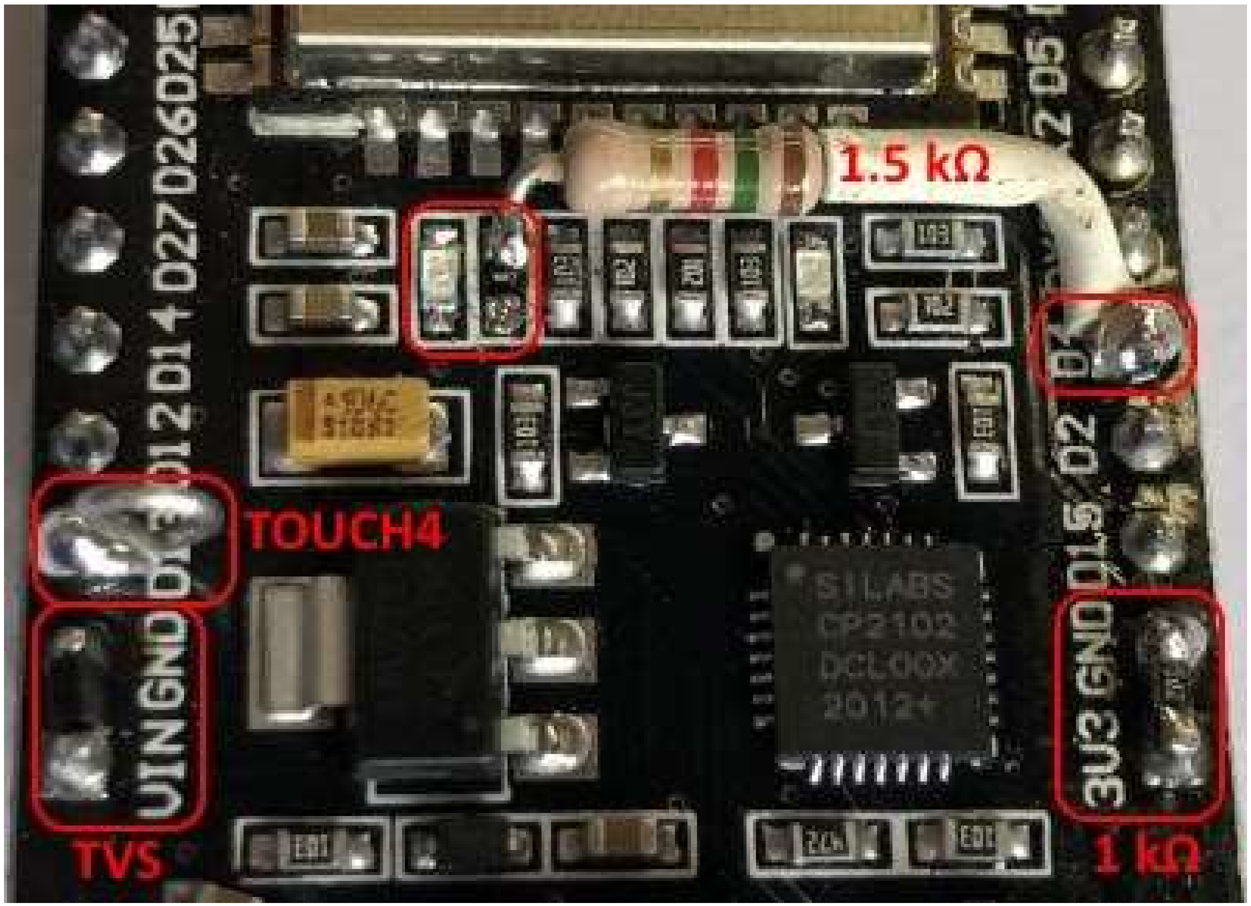


Installing a manual Touch input



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)

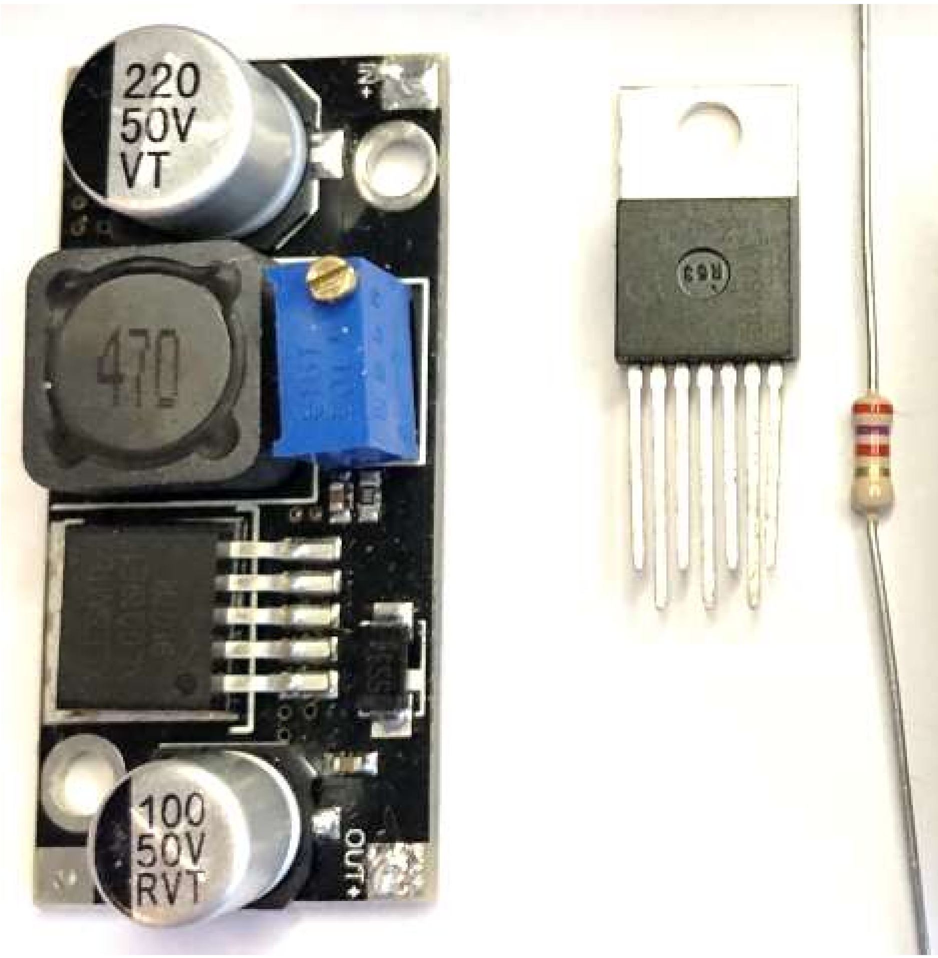
ESP32 DEVKIT V1 top view



- 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

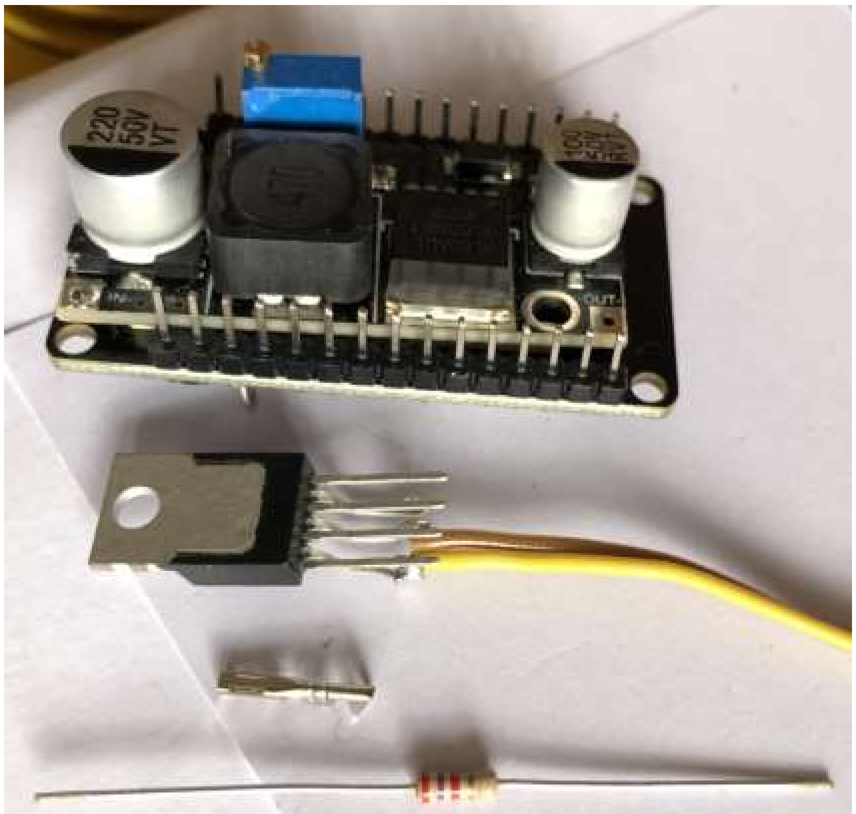
Materials for motor control

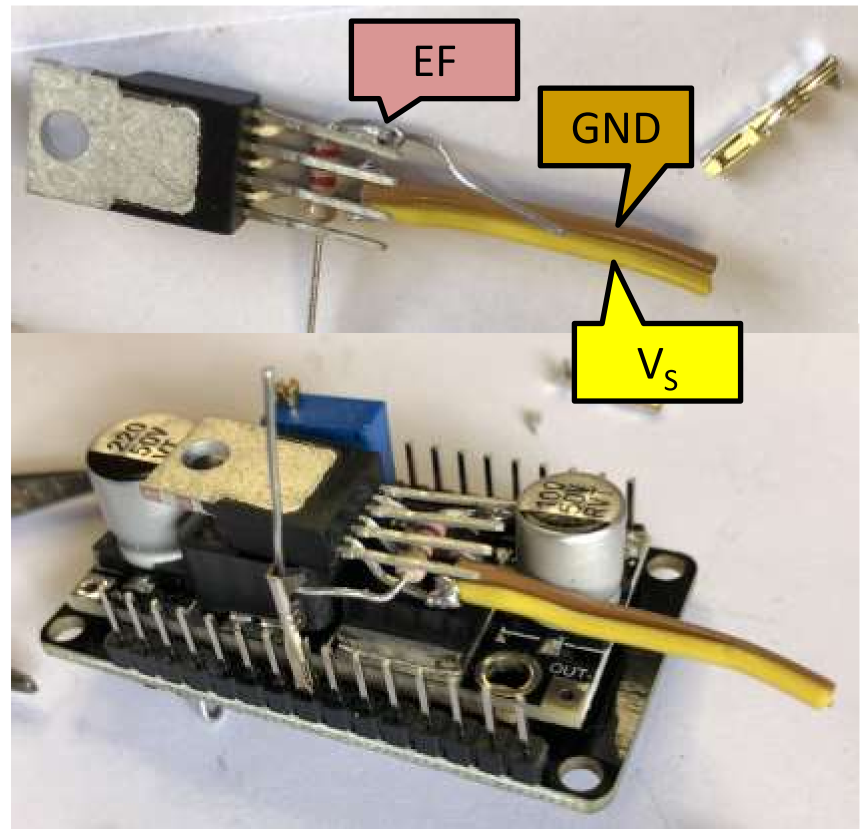
step-up-converter and motor IC fitting at ESP32 bottom

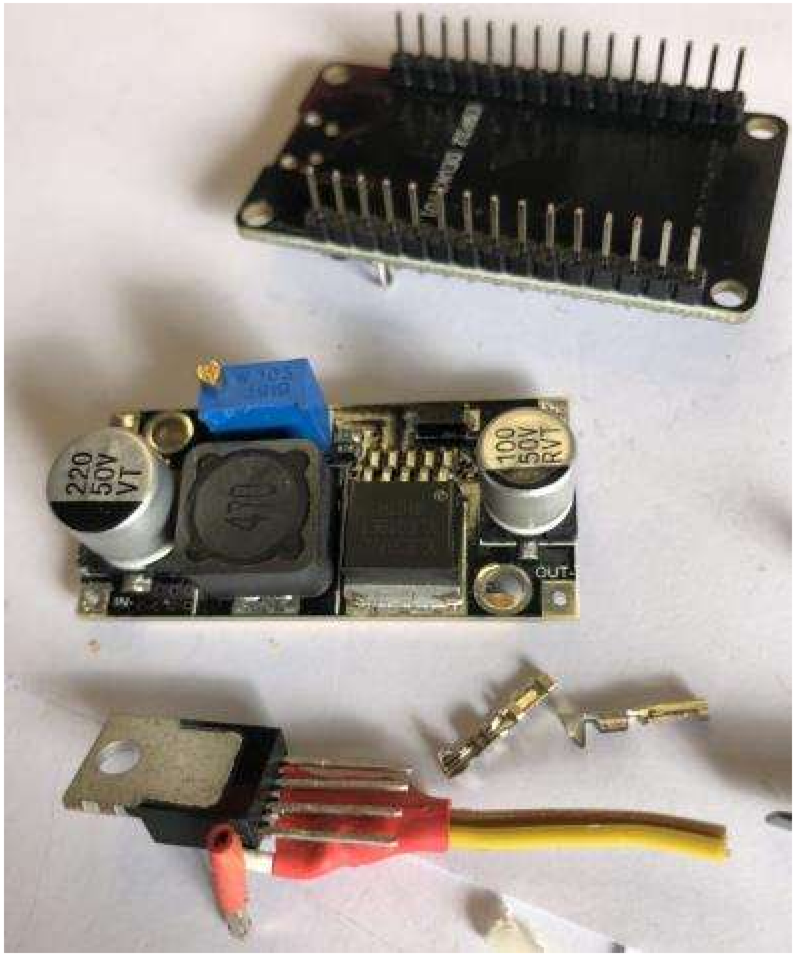


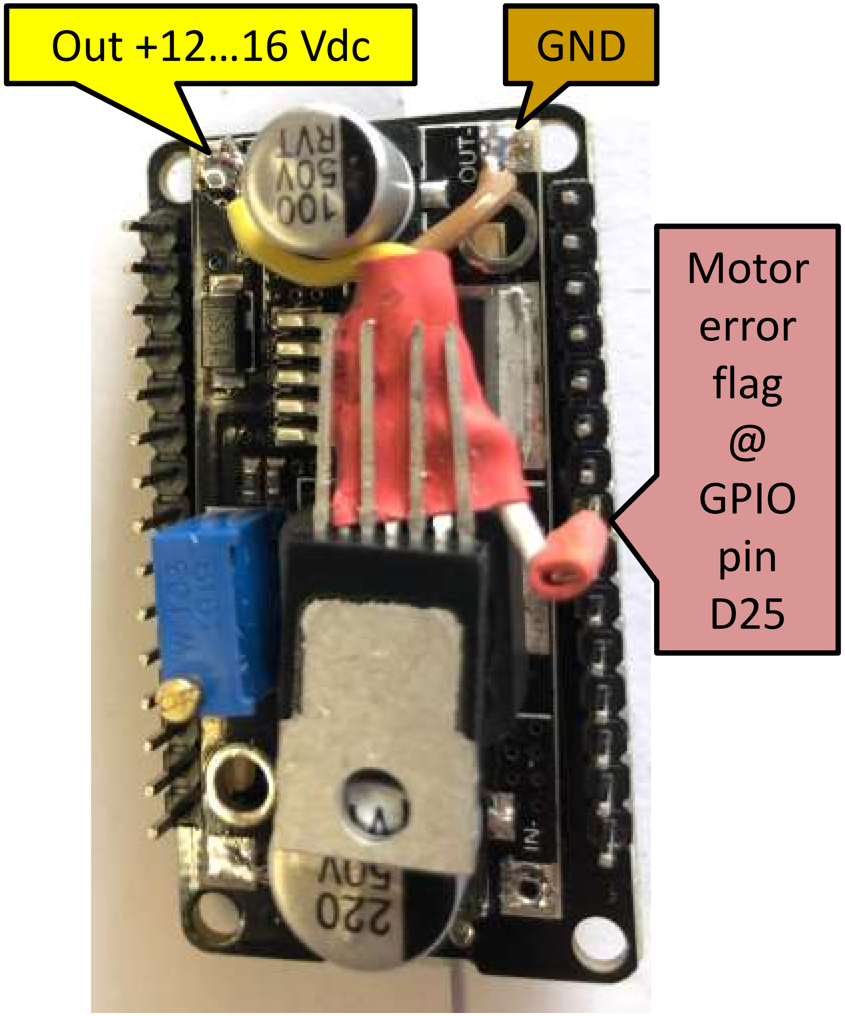
Glue two matches on bottom of step-up-converter for fitting with some distance to ESP32 bottom side

Mounting motor power control IC



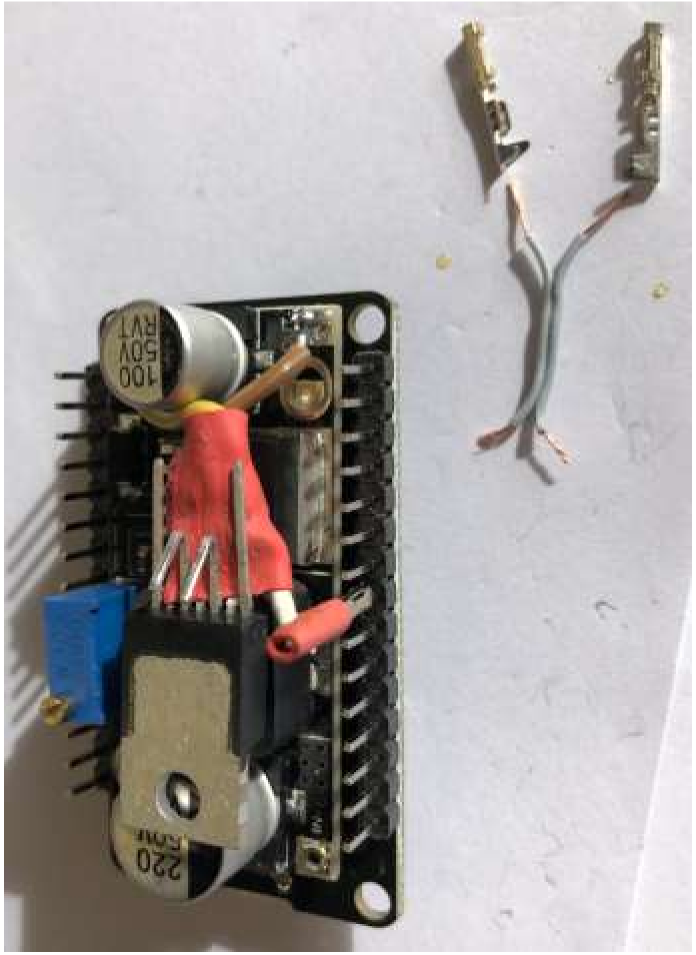


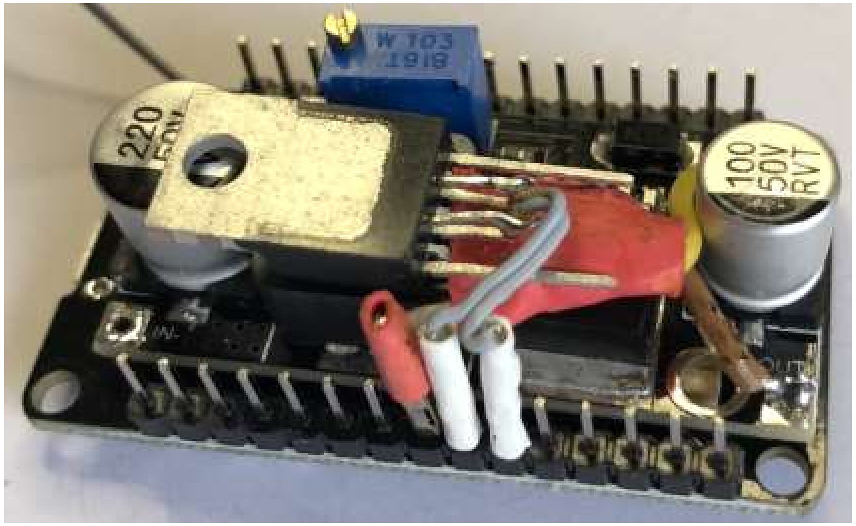




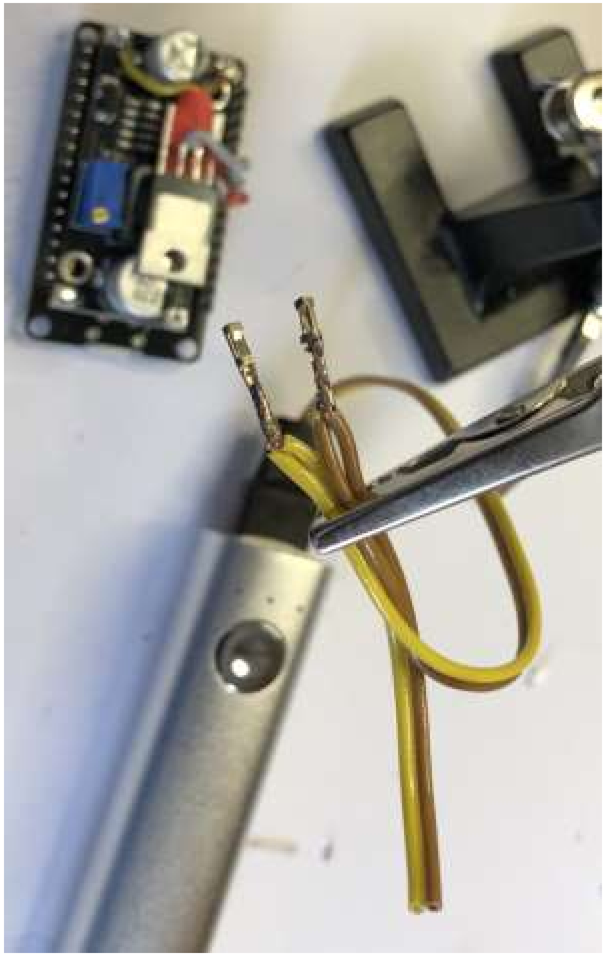
Out +12…16 Vdc
GND
Motor error flag @ GPIO pin D25

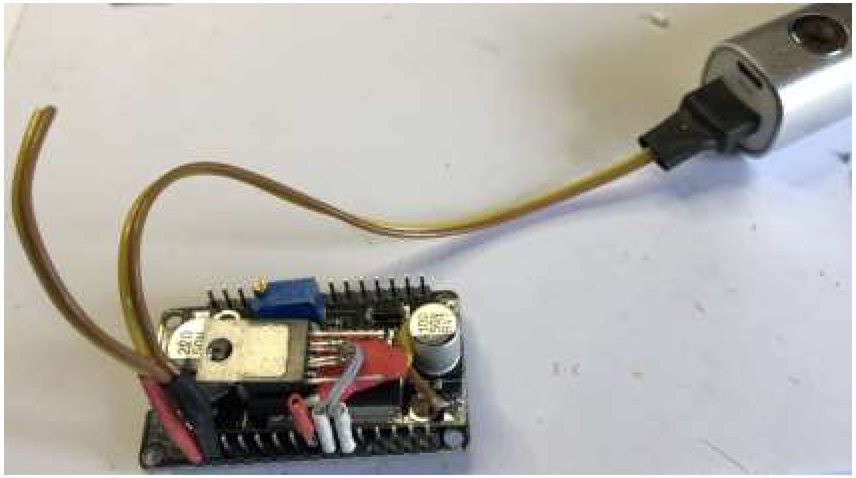
Connect ESP32 GPIO pins D33 and D32 to motor IC IN1 and IN2

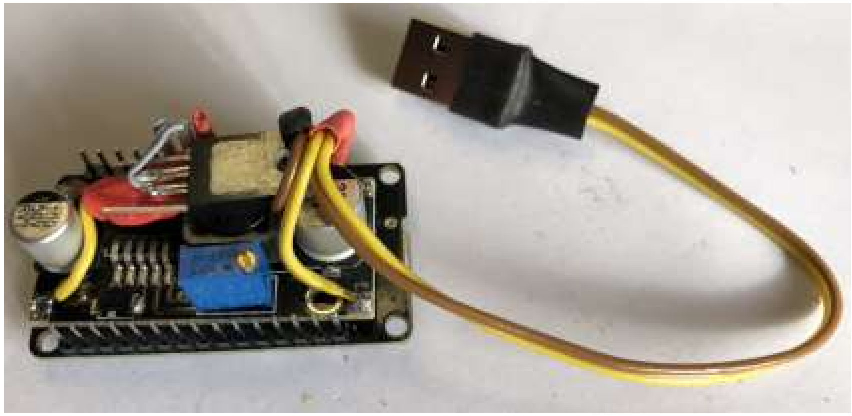




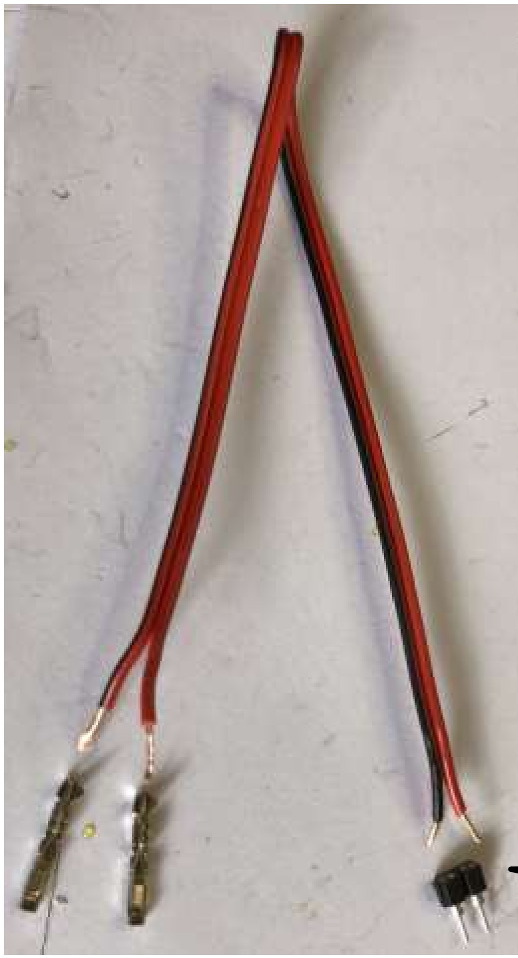
Connect USB power bank to
ESP32 VDD, GND and In+, In- of Step-up-converter

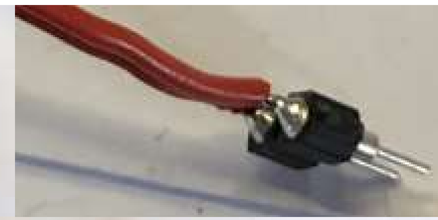






Cable with self-made mini plug for connection to locomotive motor

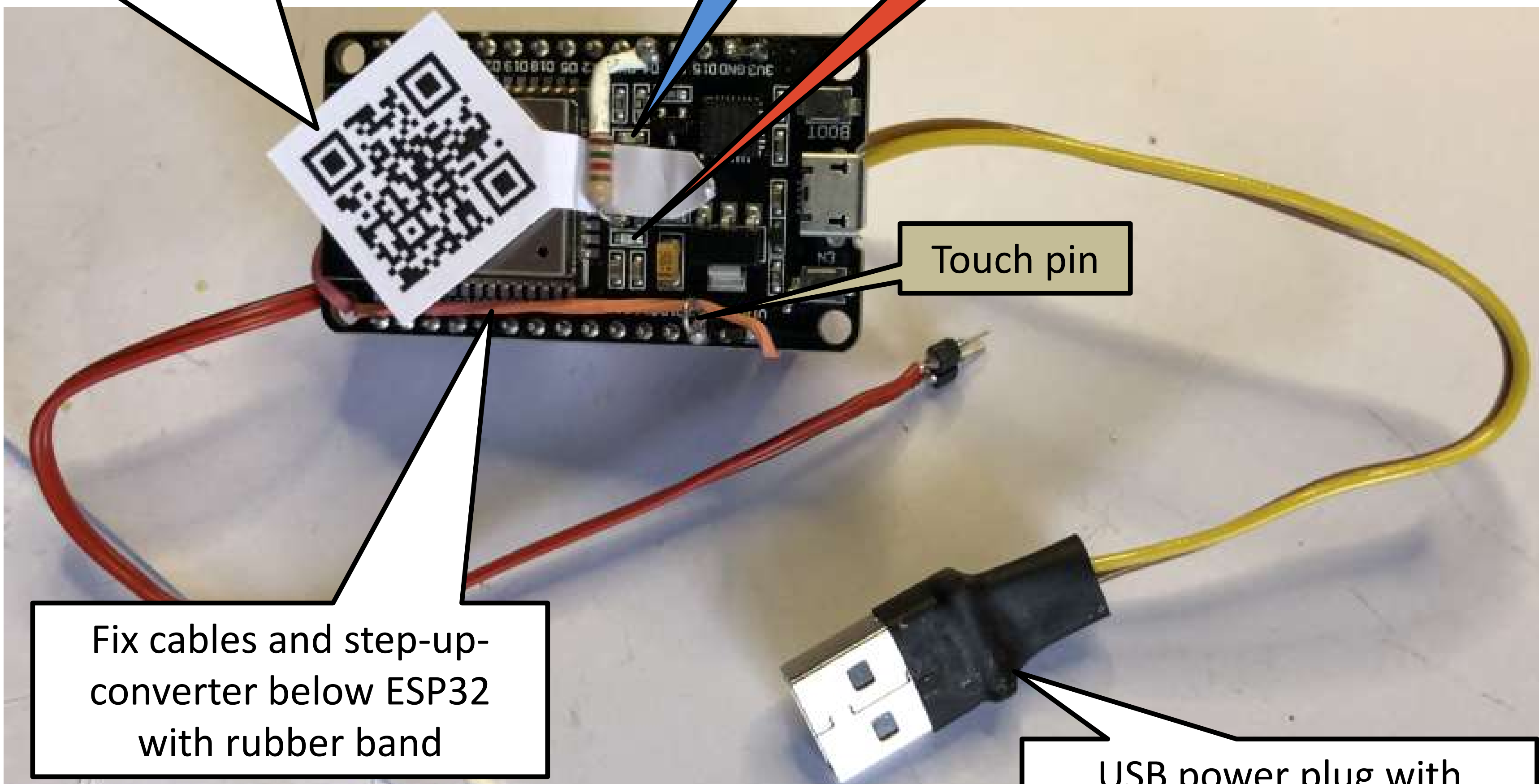




Insulate and connect motor cable to motor IC OUT1 and OUT2

Self made male mini plug for connection to locomotive motor

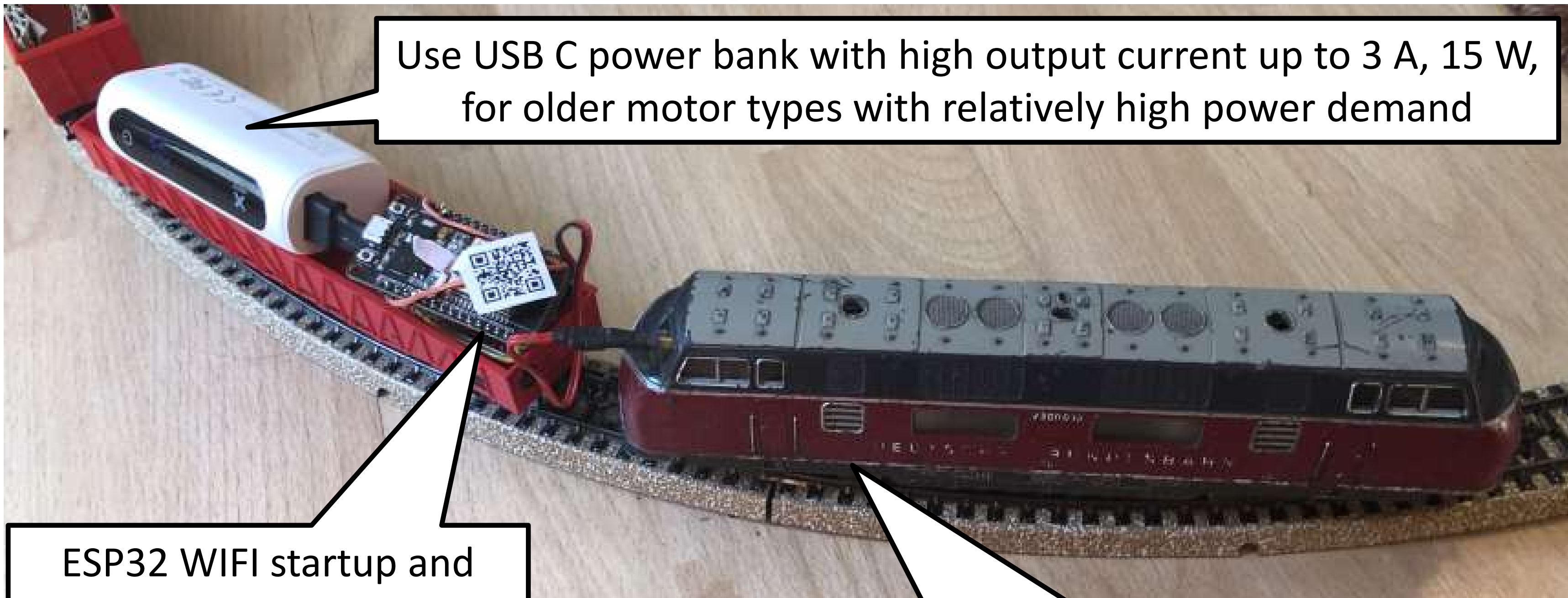
Completing ESP32 motor control module



Upload the PiedPiperS sketch and test the motor control

PiedPiperS software for ESP32
https://github.com/jorail/PiedPiperS

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.

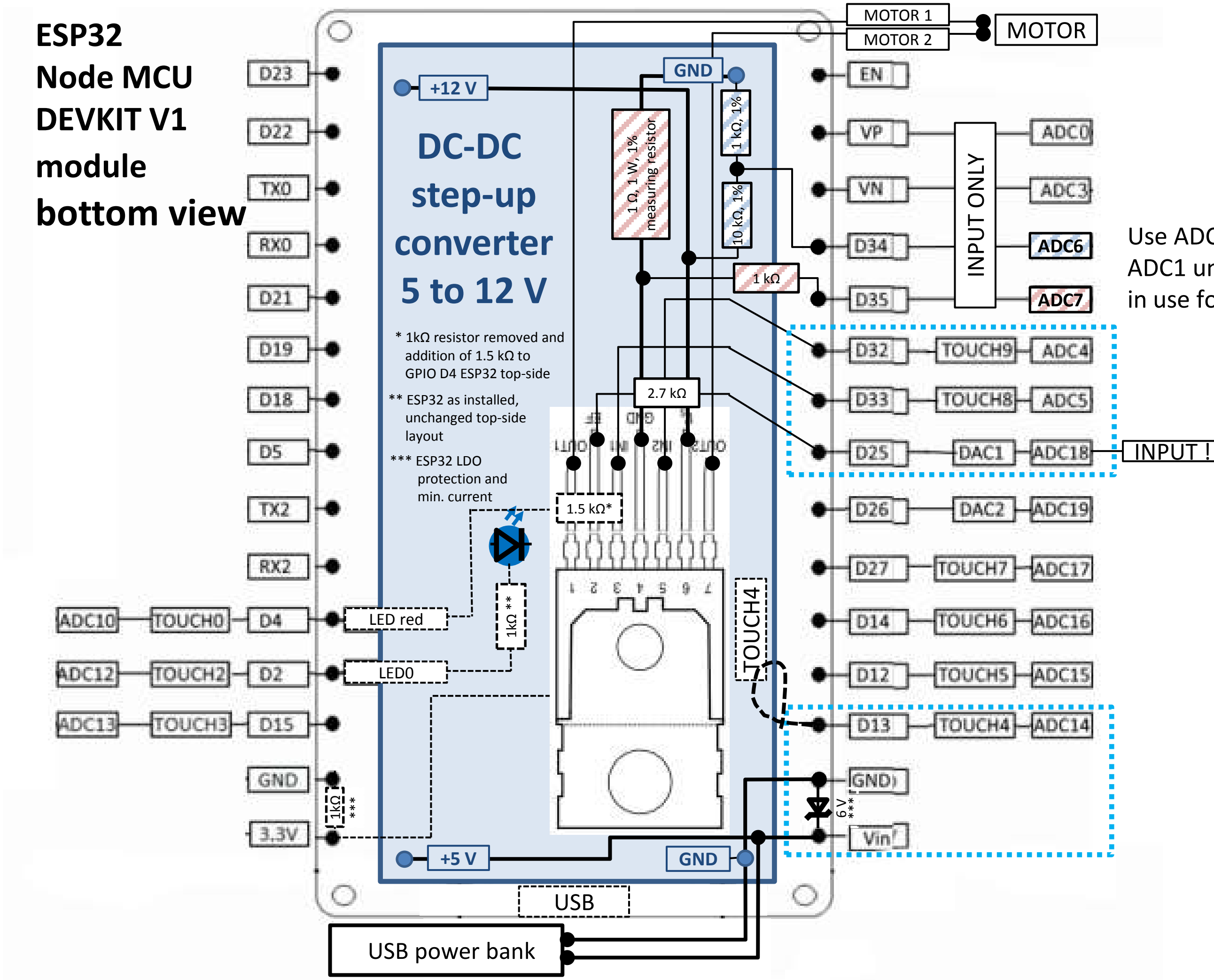
If more than one train is running, cut internal connection to rail power.

Option 1: Power sampling

by reading current and voltage on measuring resistors

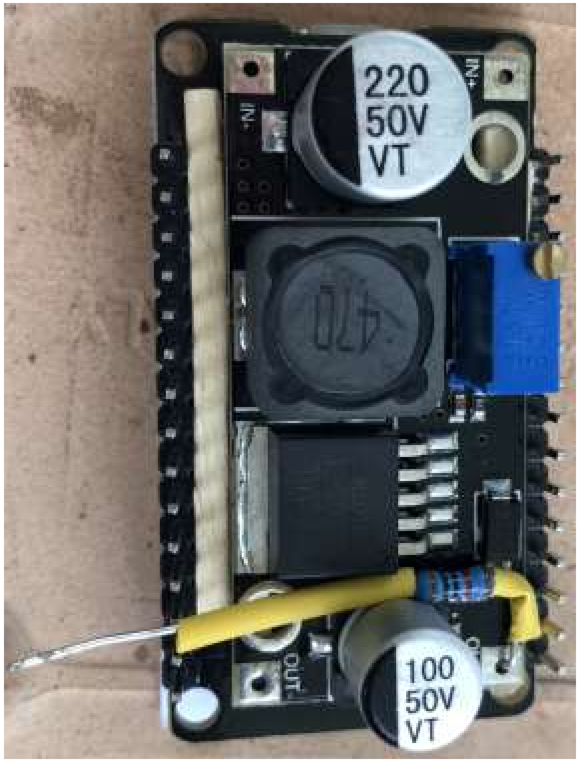
ESP32 & TLE5206-2S

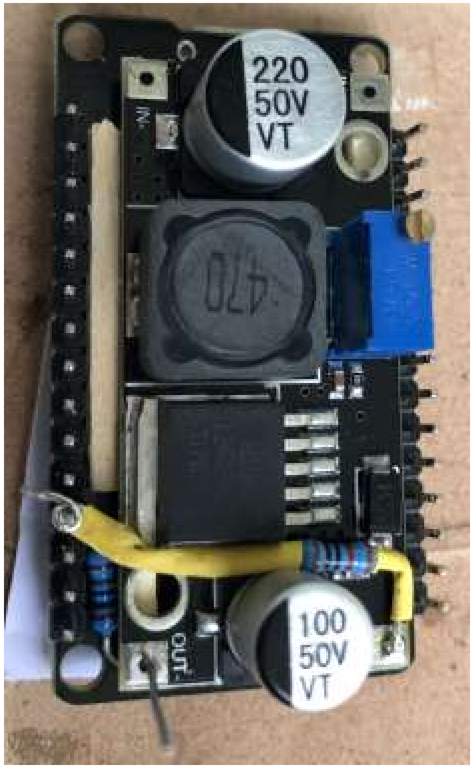
integrated mounting and wiring with LDO protection,
v173 including optional measuring resistors for power data reading



Use ADC0 unit channels only
ADC1 unit channels are already
in use for wifi operation!

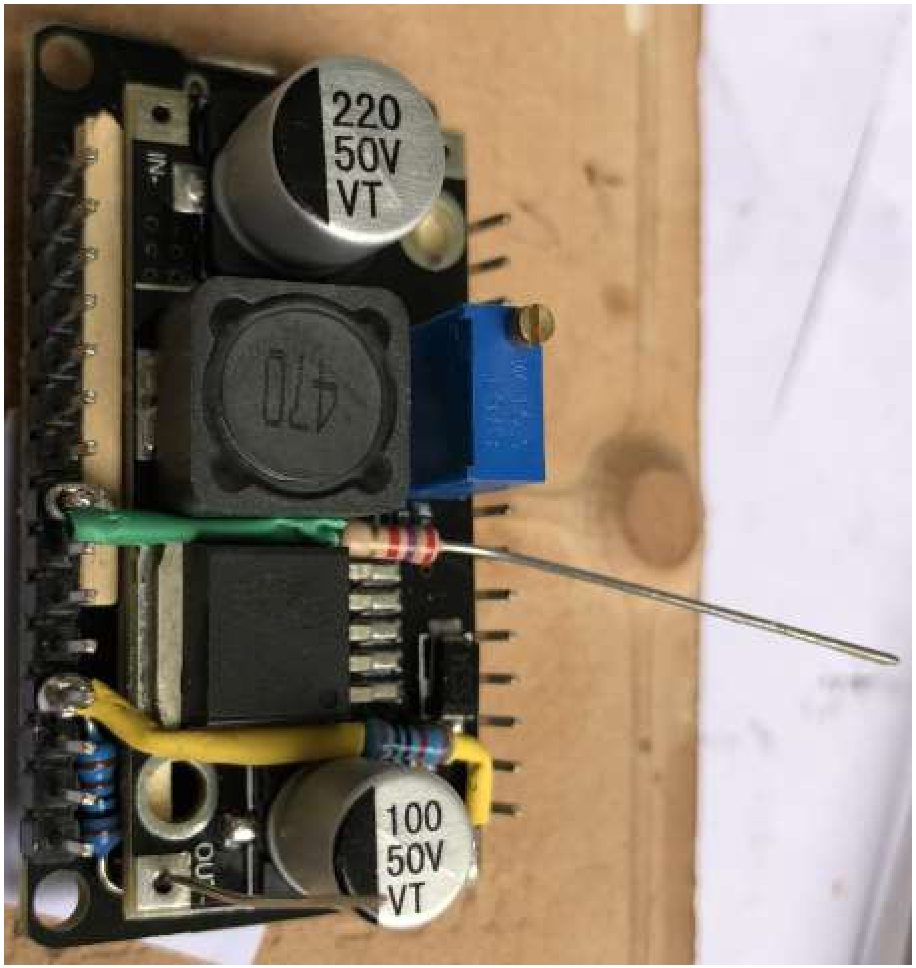
Voltage measruement from devider between OUT+ and OUT- of step-up-converter consisting of 10.0 kΩ and 1.00 kΩ measuring resistors (1% tolorance), connected to GPIO pin D34

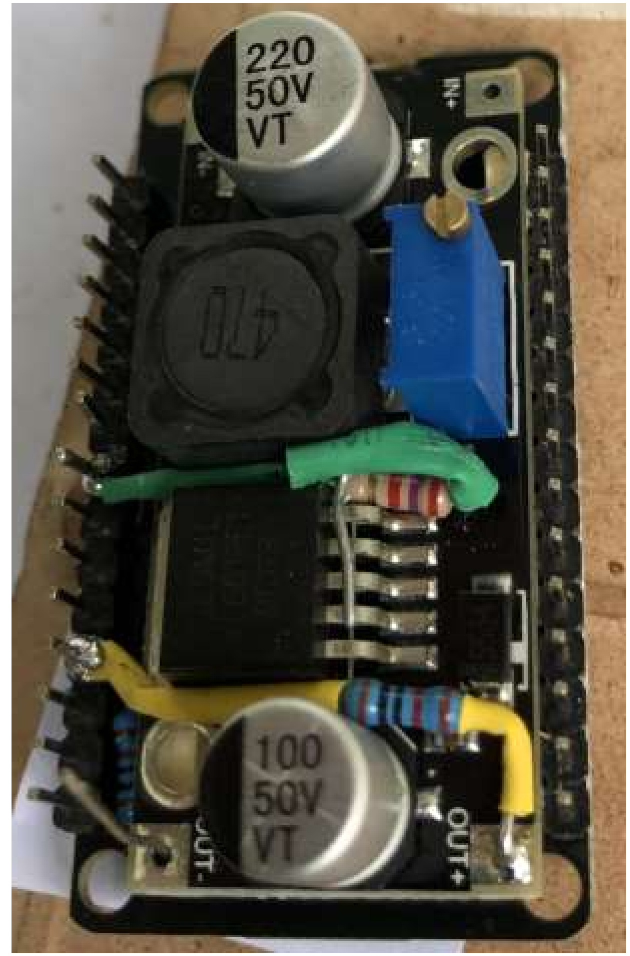




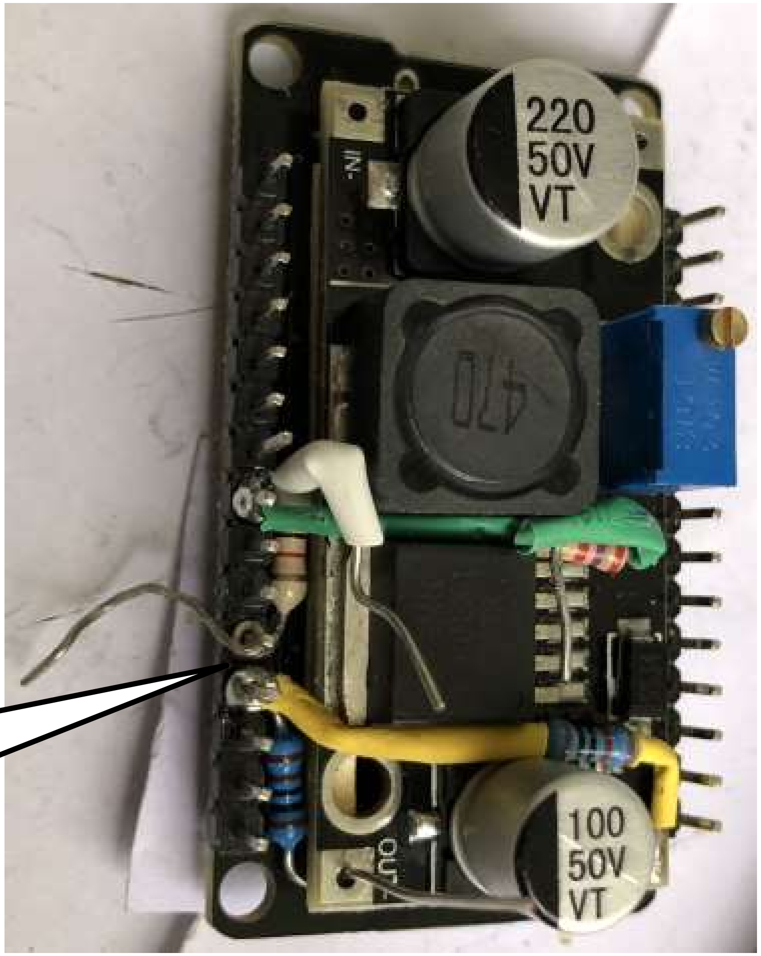
Isolation with 1 mm shrinking tube

Mounting 2.7 kΩ resistor for motor IC error flag EF as input to GPIO pin D25



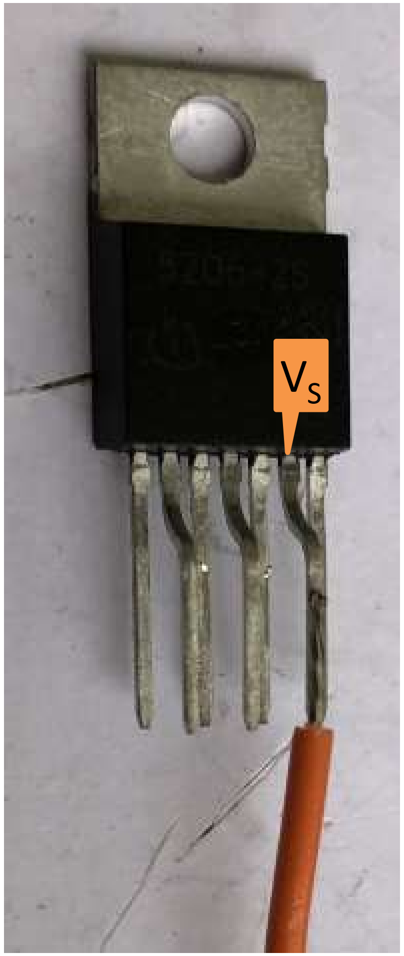


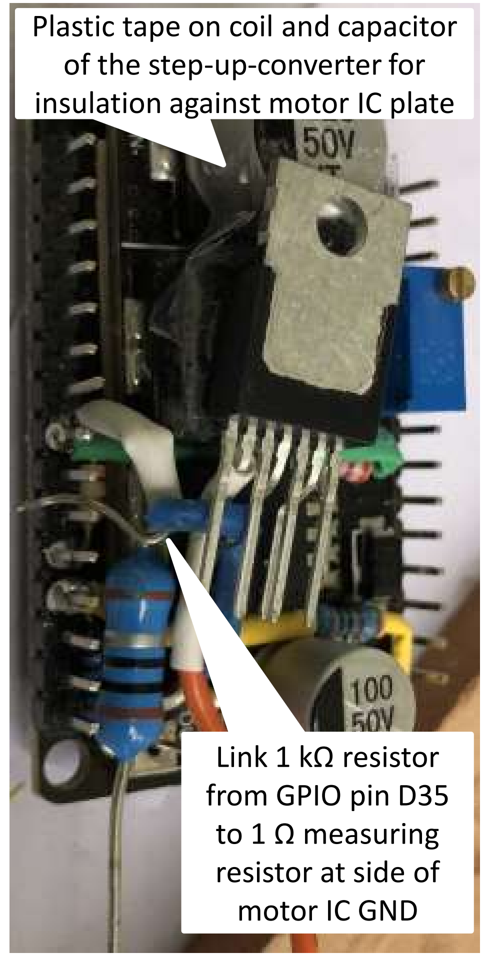
1 kΩ resistor for transfer of motor current measurment to GPIO pin D35



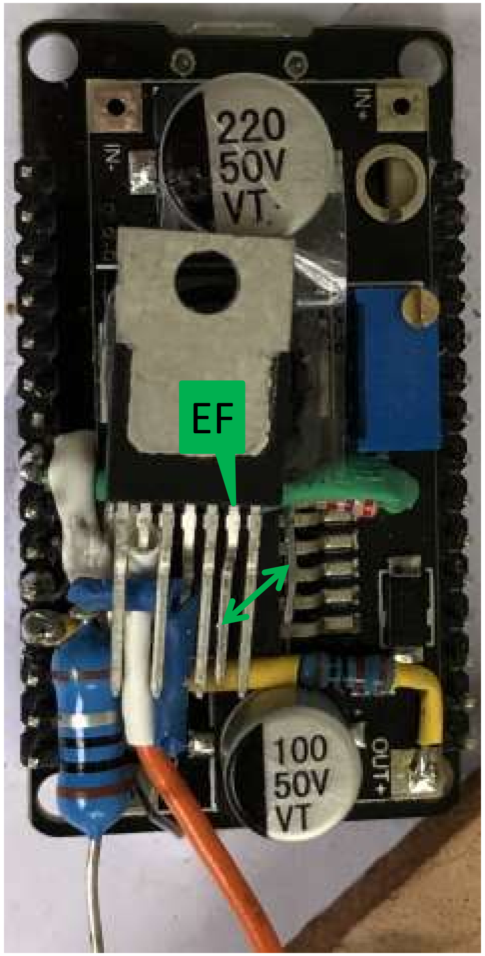
Assure sufficient distance of wires at ESP32 pins before soldering

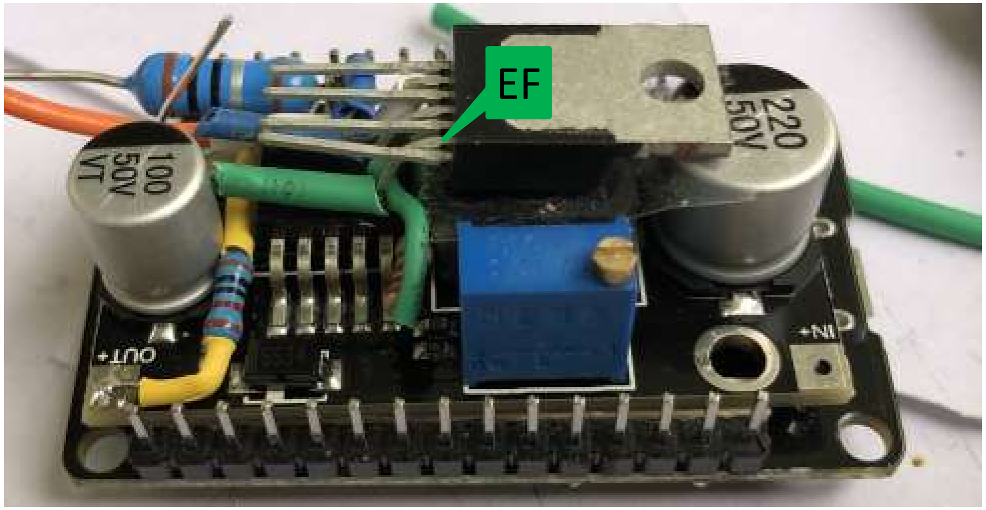
Motror IC power connection
V_S to +12…16 Vdc at OUT+ and
GND via 1 Ω current measuring resistor to OUT-





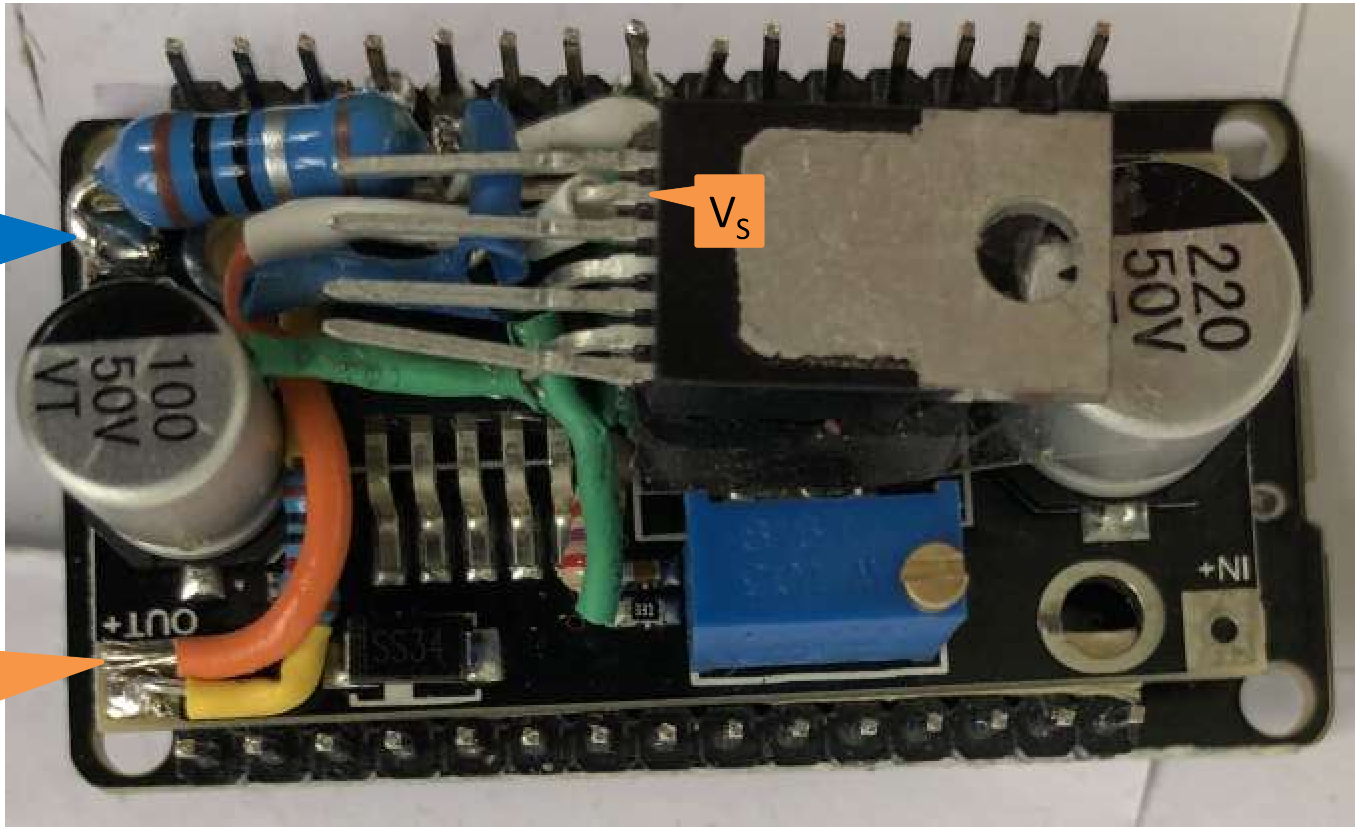
Connect 2.7 k resistor to motor IC EF, isolate the soldered link with shrink tube





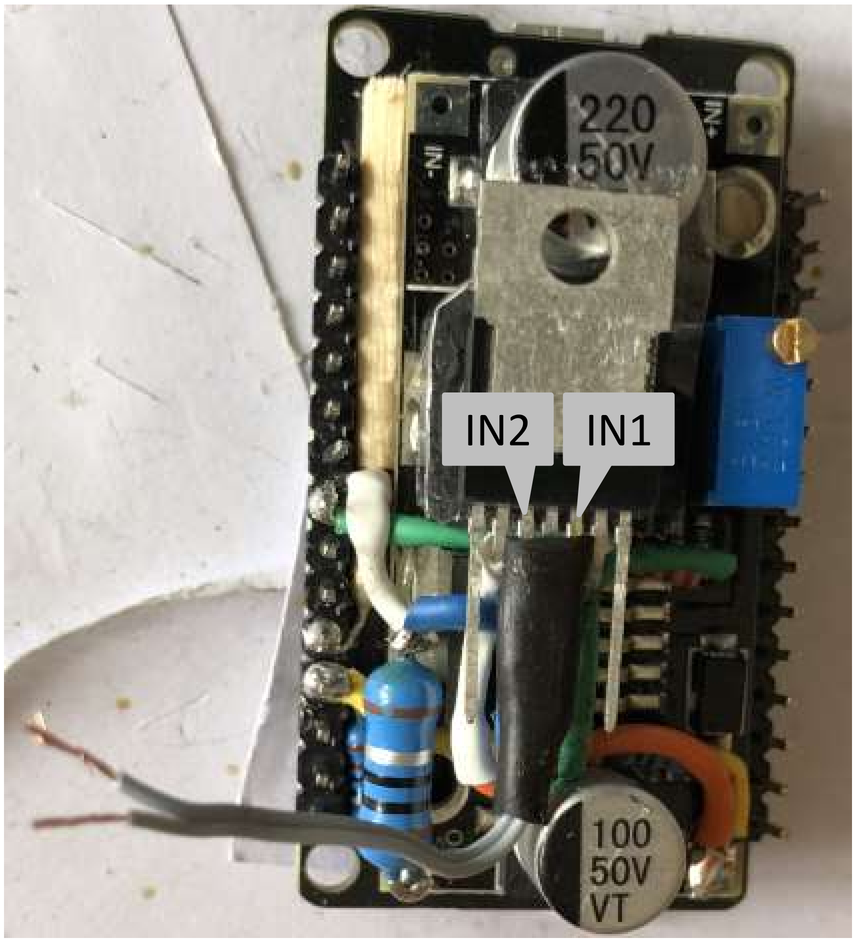
Connect motor IC to step-up converter output

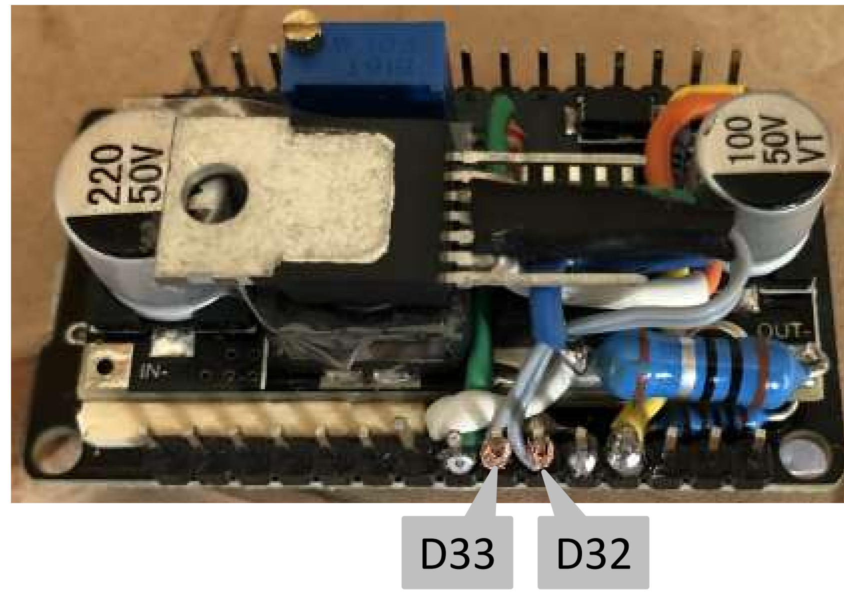
Solder other end of 1 Ω resistor to GND = OUT-



Solder V_S cable to 12…16 Vdc OUT+

Connect IN1 and IN2 to GPIO pins D33 and D32



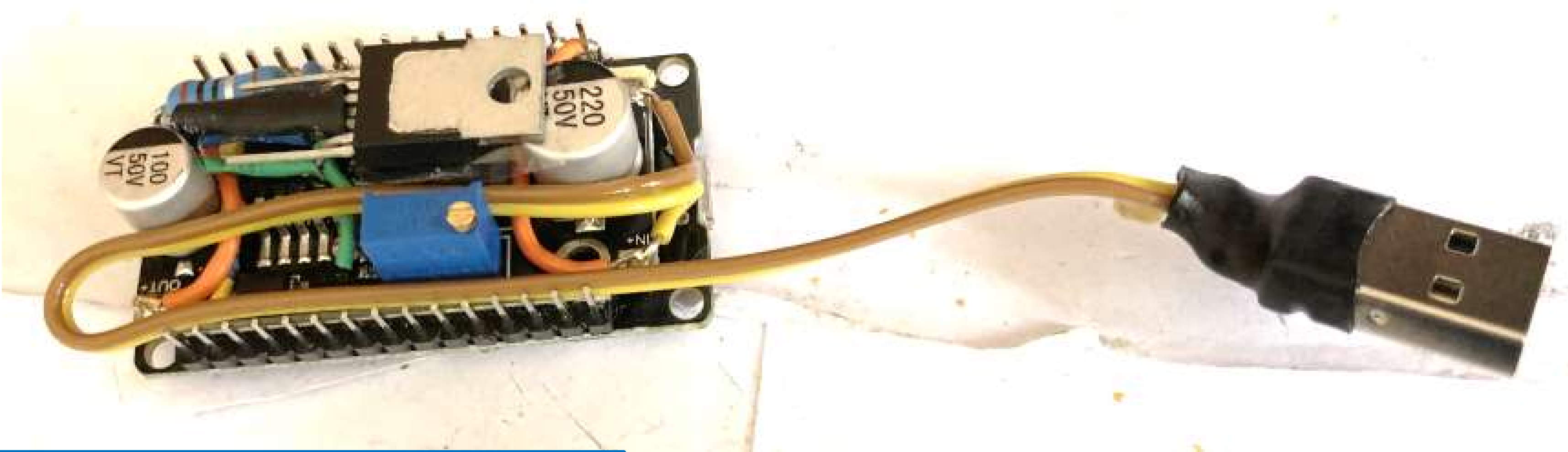


Add appropriate insulation for motor IC input pins with shrink tube

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

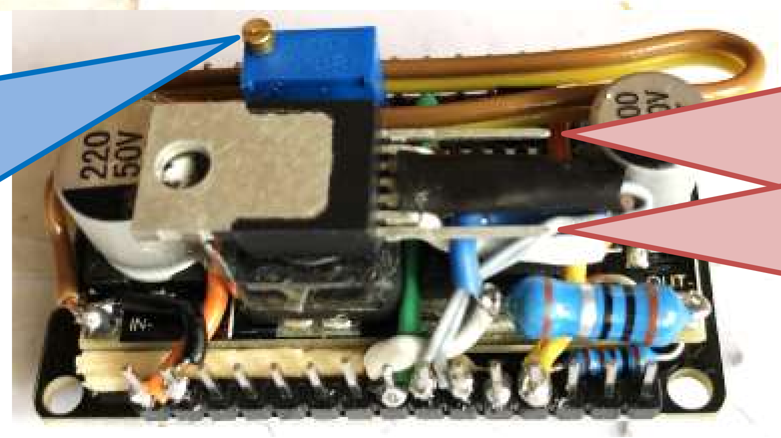
Use at least 0.25 mm^2 cable cross-section for limiting power loss before step-up converter

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

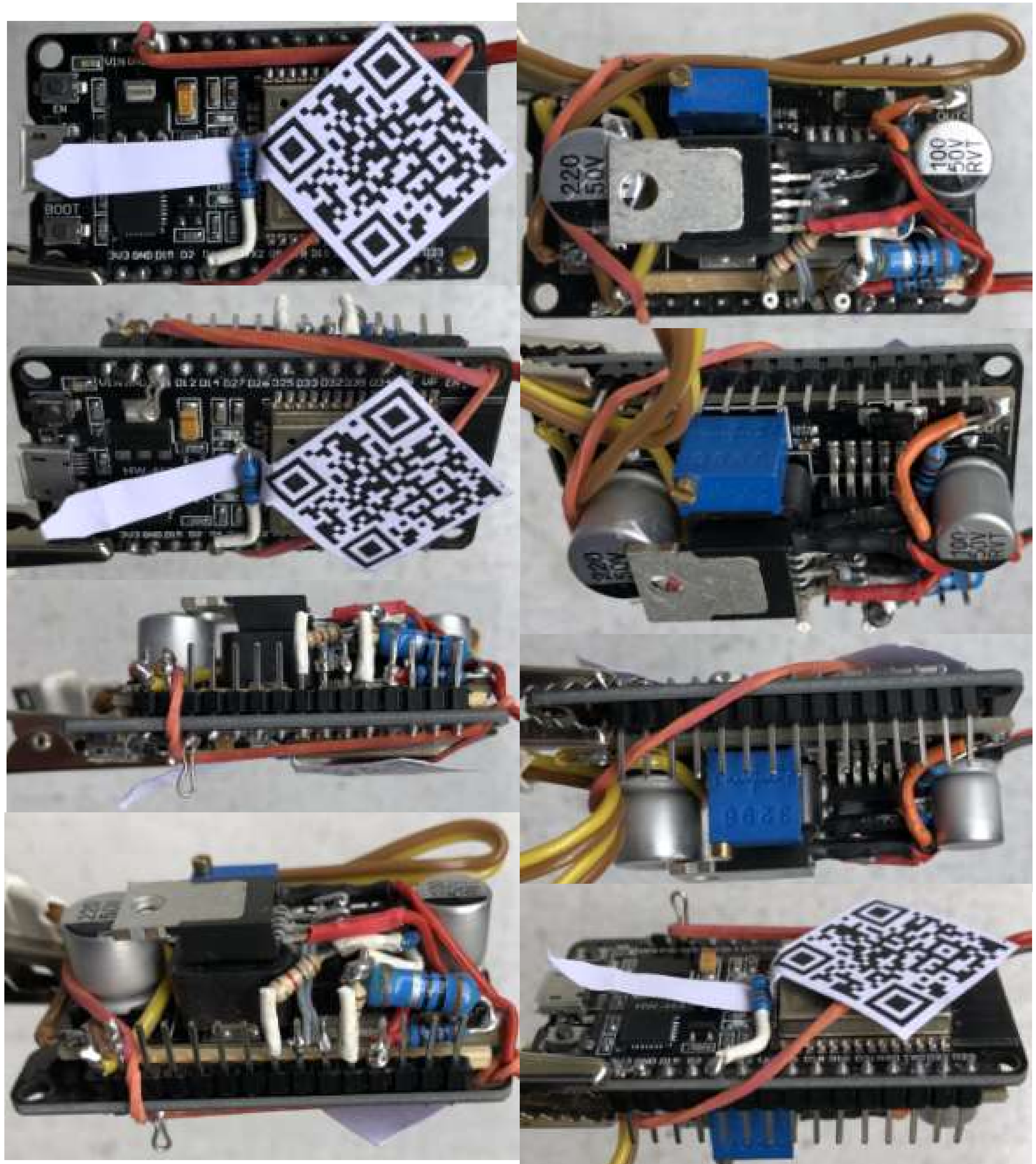


The step-up converter output voltage 12 … 16 Vdc can be adjusted and fine tuned at the potentiometer screw.

Measure the output voltage with mulitmeter. 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.


EN
BOOT
220
50V
100
50V
RVT

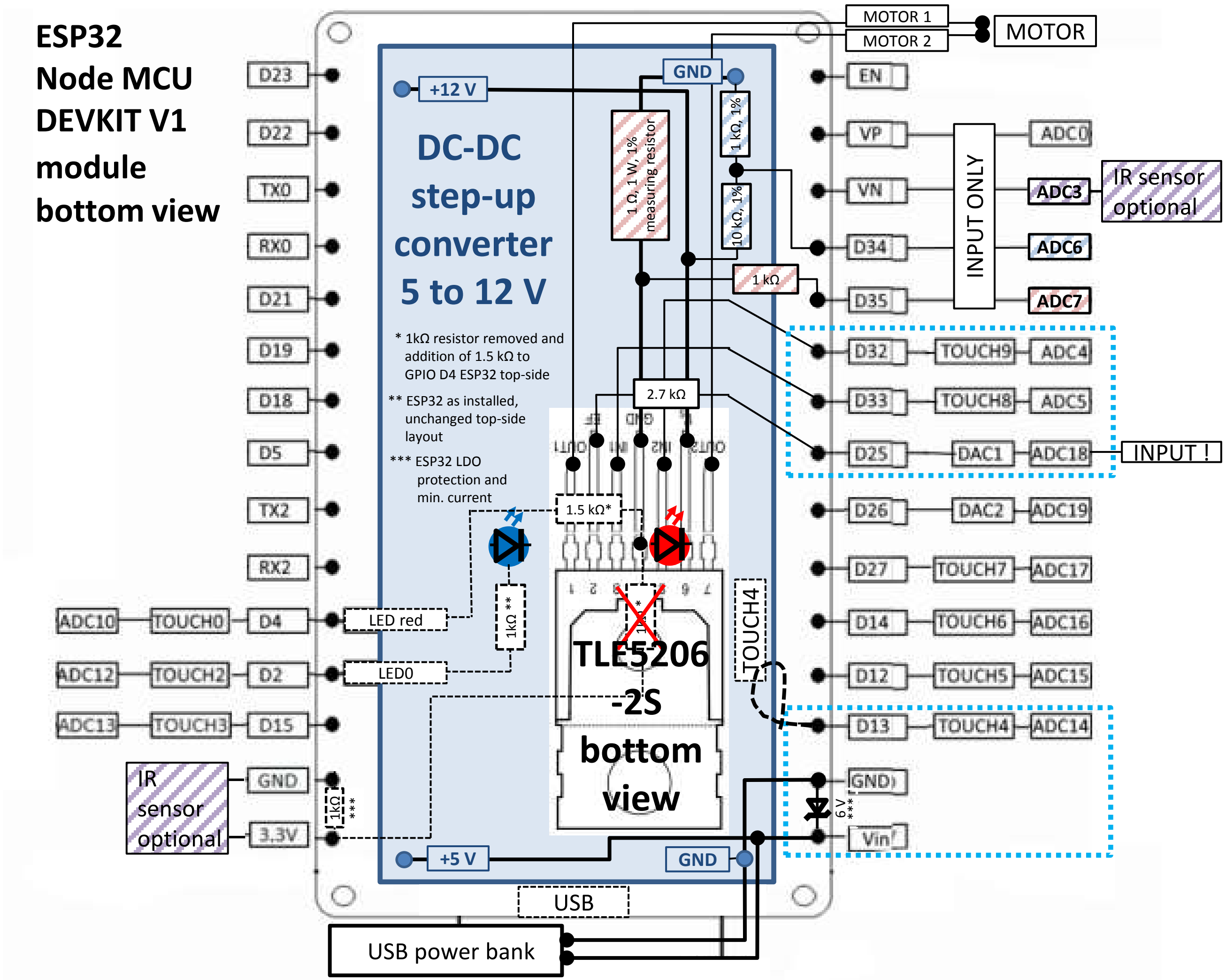
Fixing cable and electrical conection to locomotive motor

Option 2: Speedo, speed sampling

by counting railway sleepers with reflective IR sensor

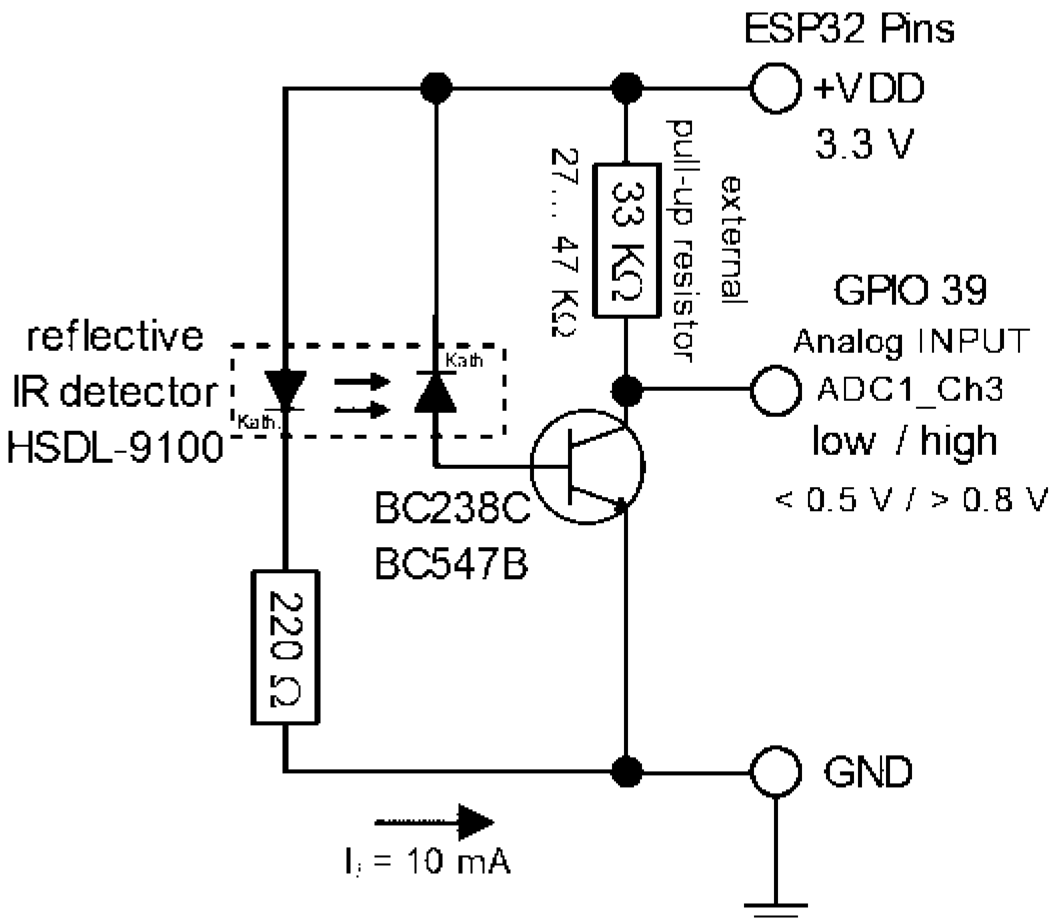
ESP32 & TLE5206-2S

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

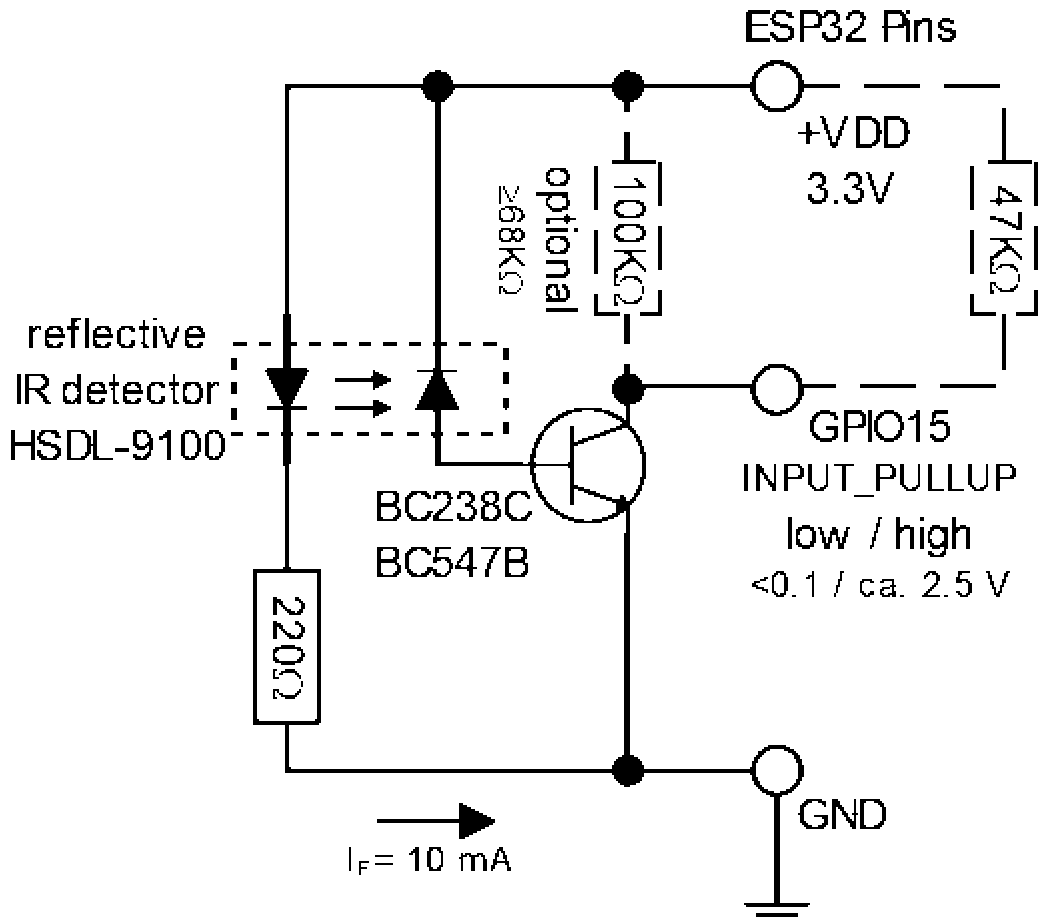


PiedPiperS version 191, J. Ruppert 2021-05-13, [GNU GPL v3](#) https://github.com/jorail/PiedPiperS

Reflective IR sensor



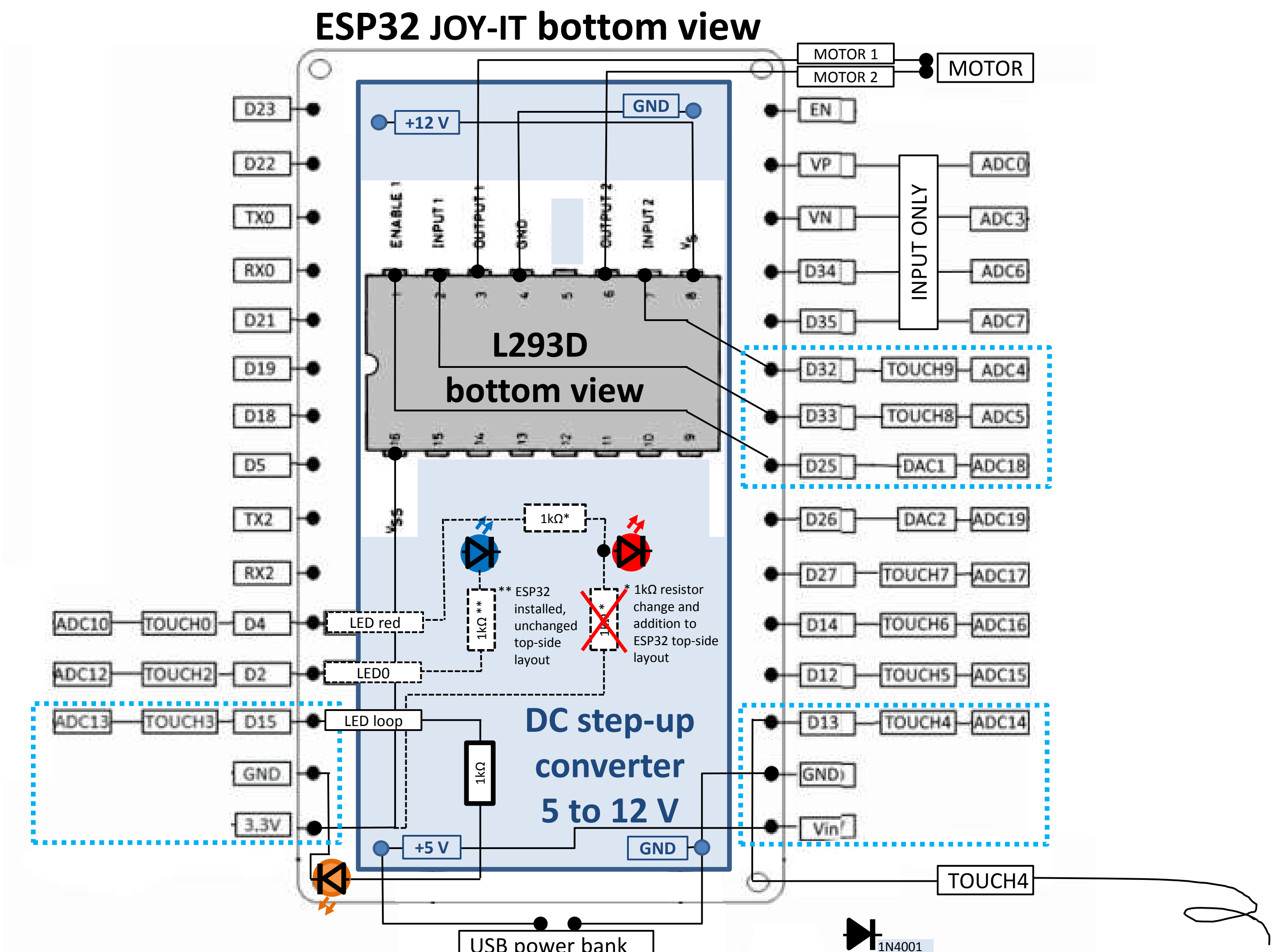
Reflective IR detector,
alternative ESP32 using internal pullup resistor



Alternative implementation with motor IC L293D

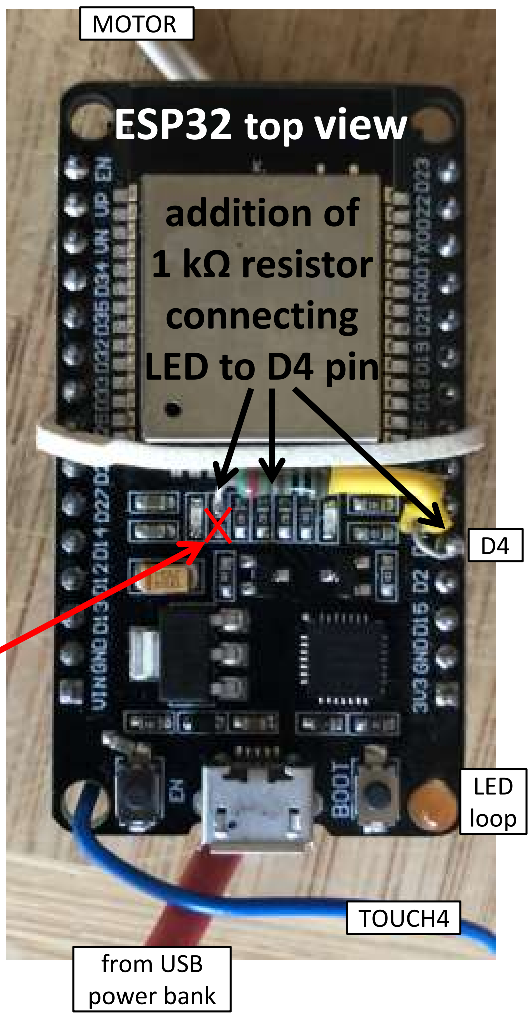
ESP32 & L293D

integrated mounting and wiring



PiedPiper Version 043, J. Ruppert 2020-12-12, [GNU GPL v3](GNU GPL v3)

ESP32 & L293D
integrated mounting and wiring



de-solder
1 kΩ SMD
resistor
marked
by 102
next to
red LED