

1a. Steuerung und Spannungsversorgung - ESP32

Diese Anleitung beschreibt die Anbindung der Wetterstation über einen ESP32. Wir verwenden hier einen NodeMCU-ESP32 (Dev Kit C V2/V4). Er ist kostengünstig, klein, bringt bereits eine WLAN-Anbindung sowie Analogeingänge mit und der Stromverbrauch ist sehr gering.

Selbstverständlich könnt ihr auch jeden anderen ESP verwenden, auch ein ESP8266 reicht theoretisch aus. Da ihr hier aber ein zusätzliches Modul für Analogeingänge anschaffen müsst, macht sich das preislich kaum etwas aus. Ich empfehle daher auf einen ESP32 zu gehen.

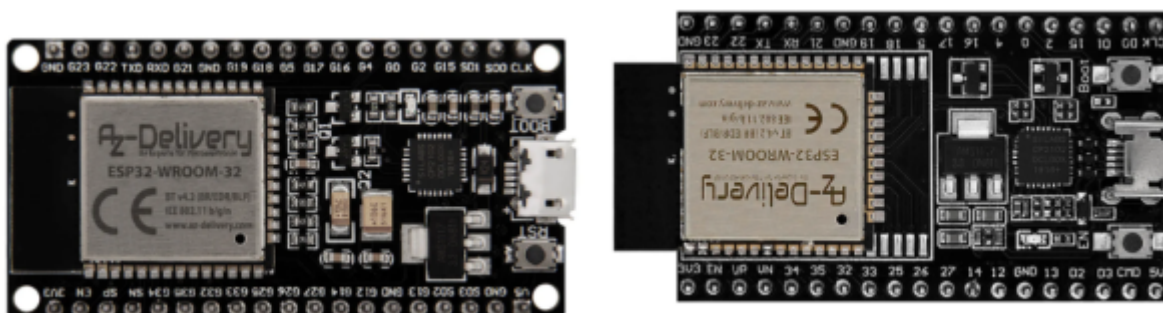
Einkaufsliste

Komponente	Preis	Bezugsquelle (Beispiel)
NodeMCU-ESP32 Dev Kit C V2/V4	5 EUR	Amazon
Netzteilmodul 24V → 5V MP1584	1,50 EUR	Amazon
Netzteil 12V / 1A	7 EUR	Amazon
Streifenraster-Platine 160 x 100 mm, RM 2,54mm	2 EUR	Amazon
Buchsen-/Steckerleiste 1-polig, RM 2,54mm (Set)	(7 EUR)	Amazon
PCB Schraubklemmen RM 2.54mm, verschiedene Größen	(1 EUR)	Amazon
Buchsenleiste, 2x20 Stifte 2,54 mm	1 EUR	Amazon
GESAMT	16 EUR	

NodeMCU-ESP32 Dev Kit C V2/V4

Der ESP32 kommt gesockelt auf eine kleine Streifenraster-Platine. Die Platine nimmt den ESP32 sowie die Spannungsversorgung auf und "konvertiert" die I/Os des ESP32 auf das Format des Raspberrys. Damit passt dann später [der Prototyping Head](#) des Raspberrys auch auf den ESP32.

Es gibt verschiedene Modelle des NodeMCU-ESP32 - oft auch als Dev Kit C bezeichnet. Am häufigsten findet man die Version 2 und Version 4. Letztendlich funktionieren beide und unterscheiden sich nur im Platindesign (und sicherlich in einigen Optimierungen). Ich würde versuchen eine Version 4 zu finden.

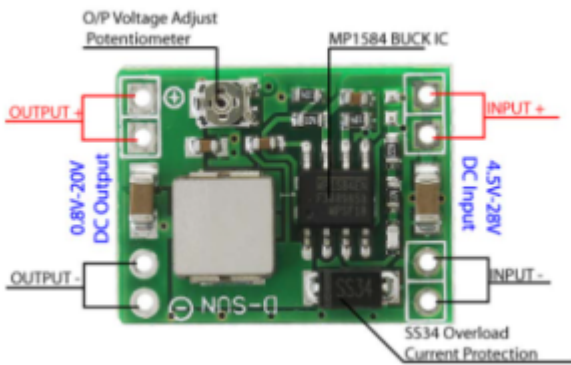


Links: [ESP-32 Dev Kit C V2](#) - Rechts: [ESP-32 Dev Kit C V4](#) / Beide Abbildungen © <https://www.az-delivery.de/>

Stromversorgung

Wir verwenden für das Projekt kein Standard-5V-Netzteil für den ESP32. Der Grund: Einige Sensoren benötigen eine Spannungsversorgung von 12V-30V. Daher nutzen wir ein 12V (oder alternativ 24V) Netzteil und steppen die 5V Spannung, die der ESP32 benötigt, entsprechend herunter.

Der ESP32 ist dabei sehr anspruchslos und verbraucht nur extrem wenig Energie. Die Sensoren benötigen etwas zusätzliche Energie. Daher können wir zur Erzeugung der notwendigen 5V-Spannung für den ESP32 einen Mini DC-DC Wandler (Typ MP1584) verwenden. Auch das Netzteil kann ein kleines 1A-Netzteil sein.



Mini MP1584 DC-DC 3A Adjustable Buck Module, © <https://components101.com>

Zusammenbau der Platine

Die Platine ist simpel aufgebaut und die Lötarbeit ist schnell erledigt. Achtet darauf, dass ihr das korrekte Layout für euren ESP32 verwendet! Ich habe hier Version 2 und Version 4 des Layouts bereit gestellt. Wenn ihr einen anderen ESP32 verwendet, müsst ihr die entsprechenden PINs passend verkabeln. Besorgt Euch dazu das PIN-Layout eures ESP und "konvertiert" es entsprechend auf die GPIO-PINs des Raspberrys.

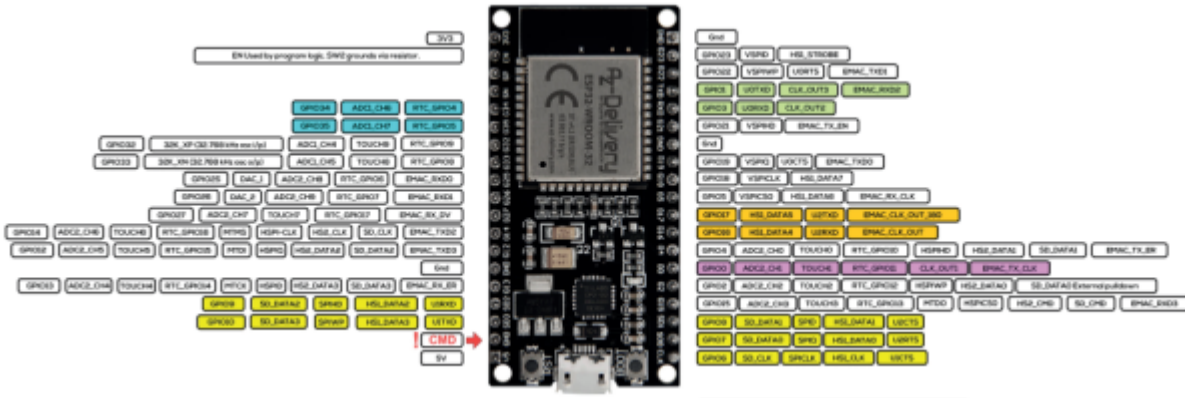
	Raspberry GPIO	Raspberry PIN	ESP32
I2C SDA	GPIO 2	3	GPIO 21
I2C SCL	GPIO 3	5	GPIO 22
DI Regenmenge	GPIO 17	11	GPIO27, ADC2_CH7, TOUCH_CH7
DI Windspeed	GPIO 27	13	GPIO26, ADC2_CH9, DAC_2
DI Regen (ja/nein)	GPIO 22	15	GPIO25, ADC2_CH8, DAC_1
DI Blitzsensor	GPIO 18	12	GPIO33, ADC1_CH5, TOUCH_CH8, XTAL_32K_N
AI Dämmerungssensor	(ADS1115 P0)		GPIO34, ADC1_CH6, VDET_1
AI UV-Sensor	(ADS1115 P1)		GPIO 39, ADC1_CH3, S_VN
AI Windrichtung	(ADS1115 P2)		GPIO 36, ADC1_CH0, S_VP

PIN-Out des Raspberrys ©

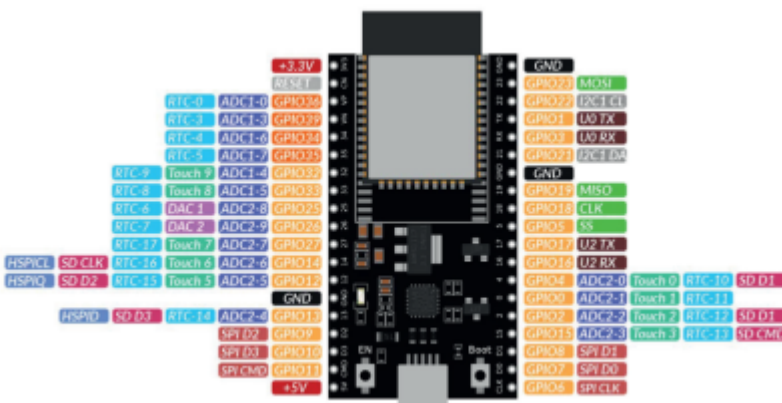
<https://www.raspberrypi-spy.co.uk/2012/06/simple-guide-to-the-rpi-gpio-header-and-pins/>



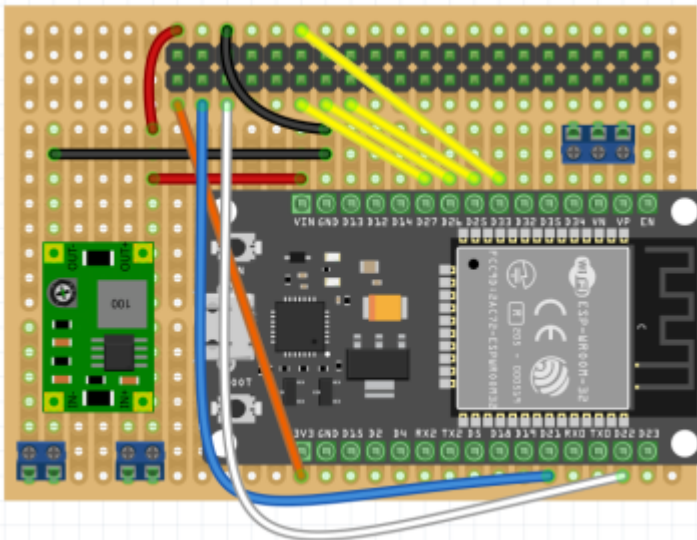
PIN-Out des ESP-32 Dev Kit C **V2** © <https://www.az-delivery.de/>



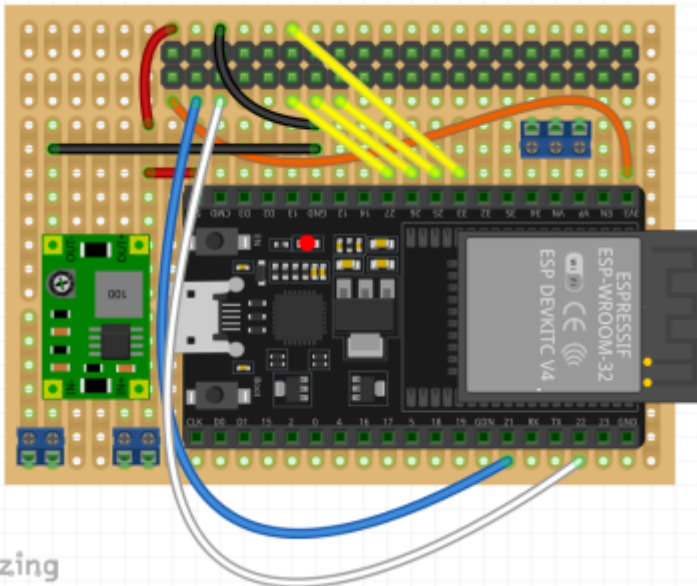
PIN-Out des ESP-32 Dev Kit C **V4** © <https://www.az-delivery.de/>



Die fertige Platine sieht dann so aus:



fritzing



fritzing

Links: **ESP-32 Dev Kit C V2** - Rechts: **ESP-32 Dev Kit C V4**

Grundaufbau

Ihr benötigt:

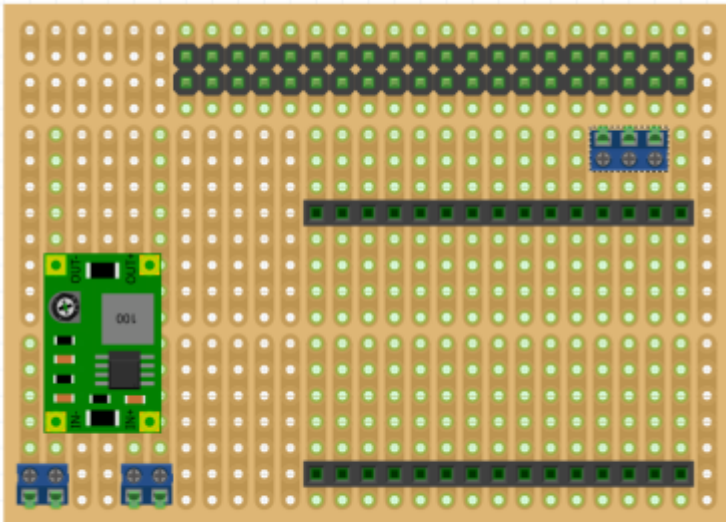
- Streifenrasterplatine 19 (H) x 27 (B) Bohrungen
- 2x PCB Schraubklemmen RM 2.54mm: 2-polig
- 1x PCB Schraubklemmen RM 2.54mm: 3-polig
- 2x Buchsenleiste 1-polig, RM 2,54mm: 15 Beine (ESP-32 Dev Kit C V2)
- 2x Buchsenleiste 1-polig, RM 2,54mm: 19 Beine (ESP-32 Dev Kit C V4)
- 2x Steckerleiste 1-polig, RM 2,54mm: 20 Beine
- 4x Steckerleiste 1-polig, RM 2,54mm: 2 Beine
- 1x Buchsenleiste, 2x20 Stifte



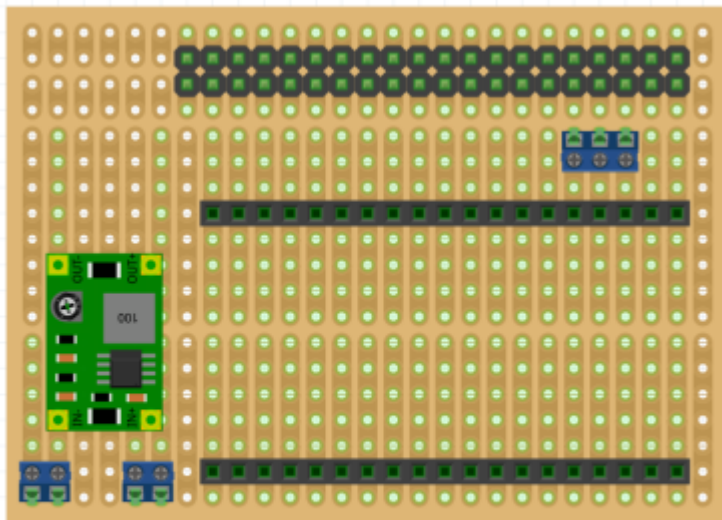


Achtet beim Layout auf die 3 durchgehenden Trennstellen der Streifen! Die Trennstellen sind durchgehend von links nach rechts! Am Besten macht man so etwas mit einem Dremel oder alternativ mit einem Cuttermesser.

Nachdem ich die Stecker- bzw. Buchsenleisten und Schraubklemmen angelötet habt, setzt ihr mit den Steckerleisten mit 2 Beinchen noch den DC-Stepdown-Wandler auf die Platine und verlötet diesen. Anschließend schließt er 12V an den Spannungswandler an **und justiert den Drehpoti auf dem Spannungswandler so, das am Ausgang ca. 5,2V anliegen**. Ich habe den Drehpoti anschließend mit einem Tropfen Sekundenkleber fixiert.



fritzing



fritzing

Links: ESP-32 Dev Kit C **V2** - Rechts: ESP-32 Dev Kit C **V4**

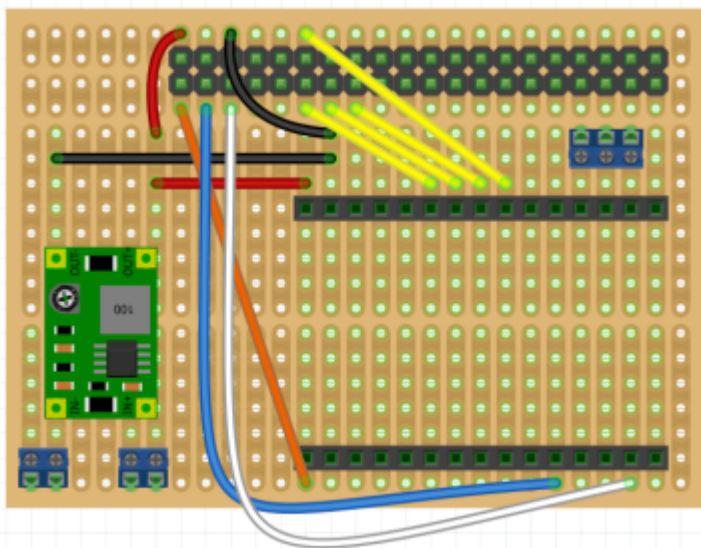
Die Buchsenleiste mit 2x20 Stiften steckt ihr nun noch auf den oberen 2x20 Stift-Header. Sie dient als Abstandshalter für den späteren Prototyping Hat. Ohne passt der ESP32 nicht unter die Platine.

Verkabelung

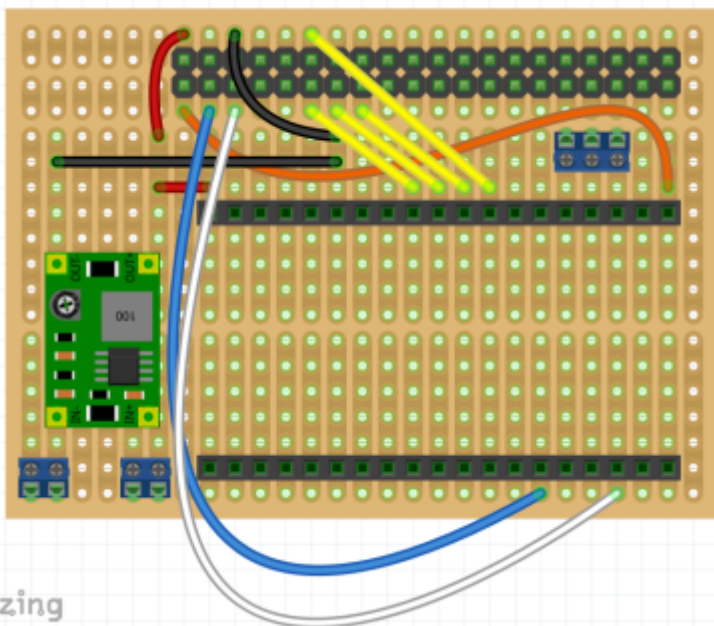
Ihr benötigt:

- Kabel Blau
- Kabel Weiß
- Kabel Rot
- Kabel Schwarz
- Kabel Orange
- Kabel Gelb

Nun werden die I/O-Pins des ESP32 passend mit den PINs der Steckerleiste für den Raspberry verbunden.



fritzing



fritzing

Links: ESP-32 Dev Kit C **V2** - Rechts: ESP-32 Dev Kit C **V4**

From:

<https://wiki.loxberry.de/> - **LoxBerry Wiki - BEYOND THE LIMITS**

Permanent link:

https://wiki.loxberry.de/howtos_knowledge_base/loxberry_wetterstation/1_steuerung/esp32?rev=1735535359

Last update: **2024/12/30 06:09**