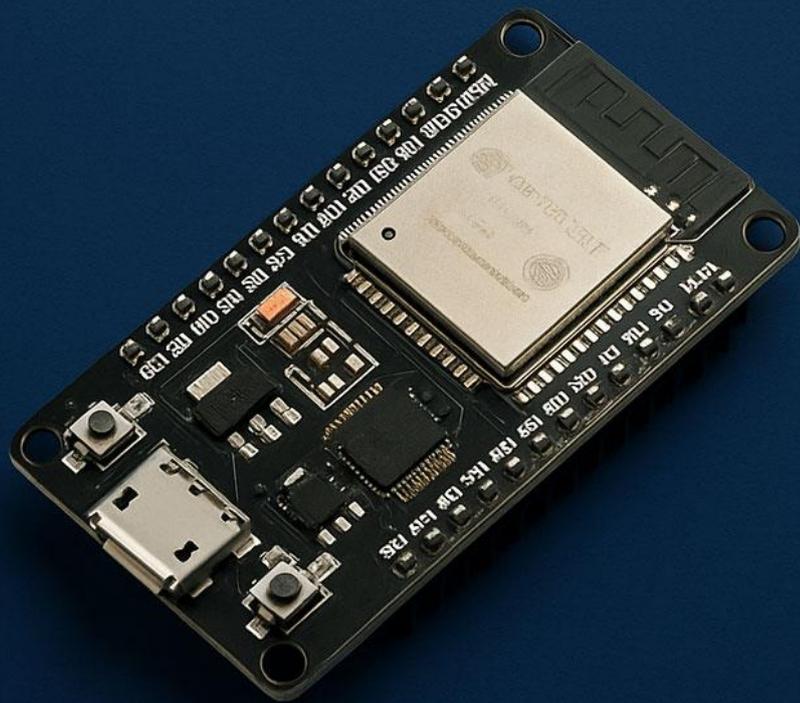




ESP32 in Arduino

einrichten & programmieren



Allgemeine Sicherheitshinweise.....	1
ESP32 in der Arduino IDE einrichten.....	2
ESP32 ins WLAN einbinden.....	5
Weitere Beispielcodes.....	6

Allgemeine Sicherheitshinweise

1. Stromversorgung

- Verwende nur die empfohlene Spannung (5V über USB oder 3.3V über geregelte Quelle).
- Vermeide direkte Versorgung über 5V-Pins – dies kann den ESP32 beschädigen.
- Niemals gleichzeitig über USB und externe Spannungsquelle versorgen, wenn nicht korrekt getrennt.

2. Kurzschlussgefahr

- Achte darauf, dass keine losen Kabel oder Bauteile auf metallischen Flächen liegen.
- Vermeide Brücken zwischen Pins oder Lötstellen ohne ausreichende Isolation.

3. Umgang mit Spannung

- Der ESP32 arbeitet mit **3,3 V-Logik** – höhere Eingangsspannungen an GPIOs (z. B. 5 V) können ihn zerstören.
- Verwende bei Bedarf Spannungsteiler oder Pegelwandler.

4. Überhitzung vermeiden

- Baue Schaltungen nicht auf wärmeempfindlichen Oberflächen auf.
- Wenn Bauteile ungewöhnlich heiß werden: **Strom sofort trennen!**

5. Arbeiten am Gerät

- Änderungen an der Verkabelung **nur bei getrennter Stromversorgung** durchführen.
- Beim Hochladen von Code keine anderen Geräte an die GPIOs angeschlossen lassen, die Störungen verursachen könnten.

6. Experimentieren mit Bedacht

- Teste neue Schaltungen zuerst mit geringer Spannung und Strom.
- Lies das Datenblatt deines ESP32-Modells und angeschlossener Komponenten sorgfältig.

7. WLAN und Netzwerksicherheit

- Verwende sichere WLAN-Passwörter.
- Veröffentliche keine sensiblen Daten im Klartext über das Netzwerk.
- Achte darauf, dass dein ESP32 nicht unbeabsichtigt offene Netzwerke bereitstellt (Access Point-Modus).

8. Ordnung & Sauberkeit

- Halte deinen Arbeitsplatz trocken, sauber und gut beleuchtet.
- Achte auf antistatische Maßnahmen bei empfindlicher Elektronik.

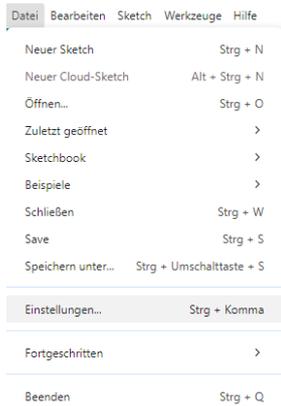
ESP32 in der Arduino IDE einrichten

Voraussetzungen

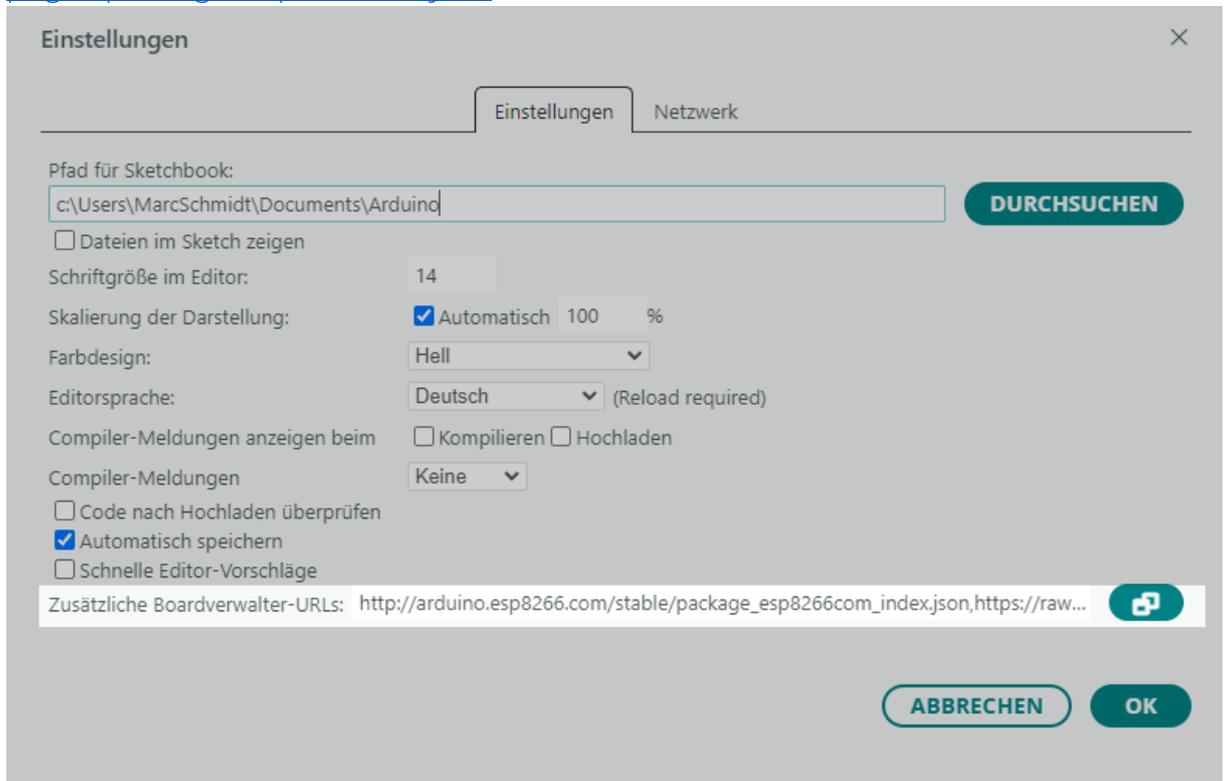
- Arduino IDE (Arduino IDE 2.x)
- USB-Kabel zum Anschluss des ESP32
- Ein ESP32-Board (z. B. ESP32 DevKit v1)

Schritt 1: Boardverwalter-URL hinzufügen

1. Öffne die Arduino IDE.
2. Gehe zu **Datei > Einstellungen**.



3. Füge unter **Zusätzliche Boardverwalter-URLs** Folgendes ein:
https://raw.githubusercontent.com/espressif/arduino-esp32/gh-pages/package_esp32_index.json

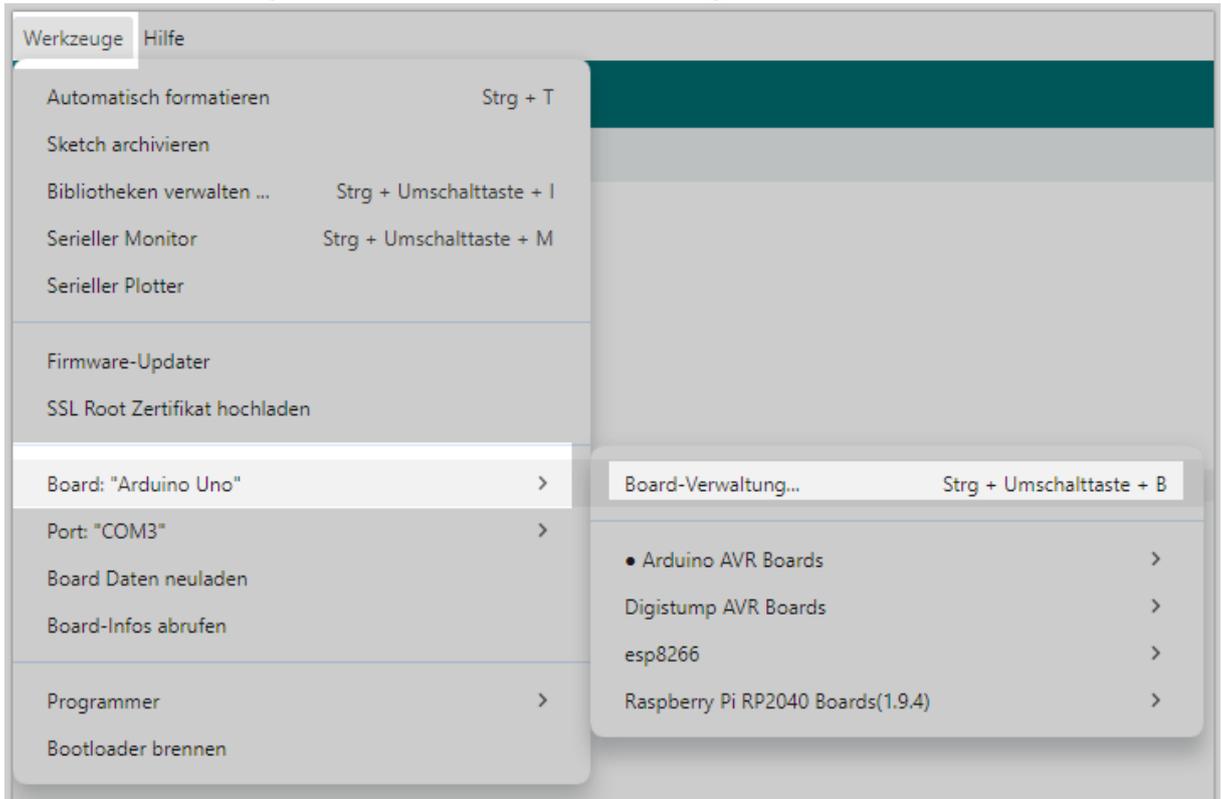


Mehrere URLs können einfach mit Komma getrennt werden.

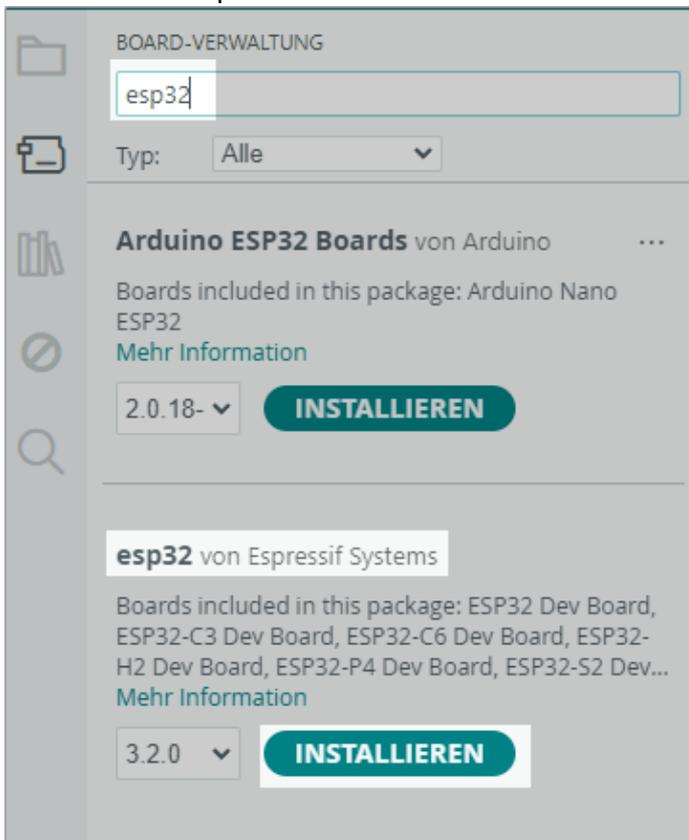
4. Klicke auf **OK**.

Schritt 2: ESP32-Board installieren

1. Gehe zu **Werkzeuge > Board > Board-Verwaltung...**



2. Suche nach **esp32**.



3. Wähle "esp32 by Espressif Systems" aus und klicke auf **Installieren**.

Schritt 3: ESP32 anschließen

1. Schließe deinen ESP32 per USB an den PC an.
2. Warte, bis er erkannt wird (ggf. Treiber installieren, z. B. CP210x oder CH340).
3. In der IDE: Gehe zu **Werkzeuge > Port** und wähle den passenden COM-Port aus.

Schritt 4: Board und Port wählen

1. Gehe zu **Werkzeuge > Board** und wähle das passende Board aus z. B. "ESP32 Dev Module".
2. Wähle den **richtigen Port** aus (z. B. COM3 oder /dev/ttyUSB0).

Schritt 5: Erstes Programm hochladen

1. Öffne das Beispiel: **Datei > Beispiele > 01.Basics > Blink**
2. Passe ggf. die LED-Pin-Nummer an (LED_BUILTIN → meist GPIO 2 bei ESP32):

```
void setup() {  
  pinMode(2, OUTPUT); // GPIO 2  
}  
  
void loop() {  
  digitalWrite(2, HIGH);  
  delay(1000);  
  digitalWrite(2, LOW);  
  delay(1000);  
}
```

3. Klicke auf den **Haken** (Überprüfen) und dann auf den **Pfeil** (Hochladen).

 **Hinweis:** Wenn der Upload hängt, drücke während des "Connecting..." den Boot-Knopf am ESP32, bis der Upload startet.

Fertig!

Die LED auf dem ESP32 sollte jetzt im Sekundentakt blinken – dein ESP32 läuft!

ESP32 ins WLAN einbinden

Voraussetzungen

- WLAN-Name (SSID)
- WLAN-Passwort

Beispielcode um WLAN-Verbindung herzustellen

```
#include <WiFi.h> // Bibliothek für ESP32 WLAN

const char* ssid = "DEIN_SSID"; // <-- Dein WLAN-Name
const char* password = "DEIN_PASSWORT"; // <-- Dein WLAN-Passwort

void setup() {
  Serial.begin(115200);
  delay(1000);

  Serial.println("Verbinde mit WLAN...");
  WiFi.begin(ssid, password);

  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }

  Serial.println("");
  Serial.println("WLAN verbunden!");
  Serial.print("IP-Adresse: ");
  Serial.println(WiFi.localIP());
}

void loop() {
  // Hier kannst du z. B. Sensorwerte senden oder einen Webserver starten
}
```

So verwendest du den Code

4. Öffne die Arduino IDE.
5. Ersetze DEIN_SSID und DEIN_PASSWORT im Sketch mit deinen WLAN-Daten.
6. Hochladen auf den ESP32.
7. Öffne den **seriellen Monitor** (Baudrate 115200), um den Verbindungsstatus und die IP-Adresse zu sehen.

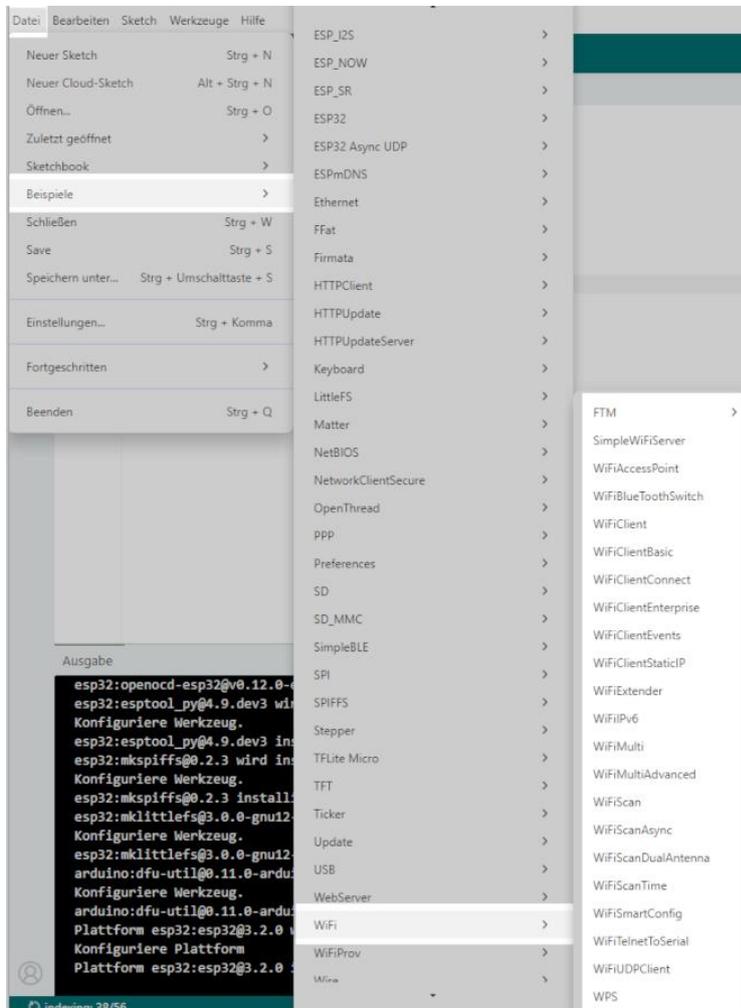
Was kannst du mit dem Code machen?

- Einen **Webserver** auf dem ESP32 starten
- Daten an einen **MQTT-Broker** senden
- Einen **IoT-Sensor** ins Heimnetz einbinden

Weitere Beispielcodes

Wenn das Board ESP32 eingestellt ist, findest du unter „Datei – Beispiele „ jede Menge praktische Codes zum Ausprobieren.

Besonders praktisch sind die Codes unter „Datei – Beispiele - WIFI“ wie z. B. der Wifiscan. Der WIFI-Scanner startet das ESP32 Modul im Scan-Modus und sucht nach verfügbaren WLANs, zeigt deren Name, Signalstärke und Verschlüsselung an.



In der Liste ganz runter bis WiFi scrollen