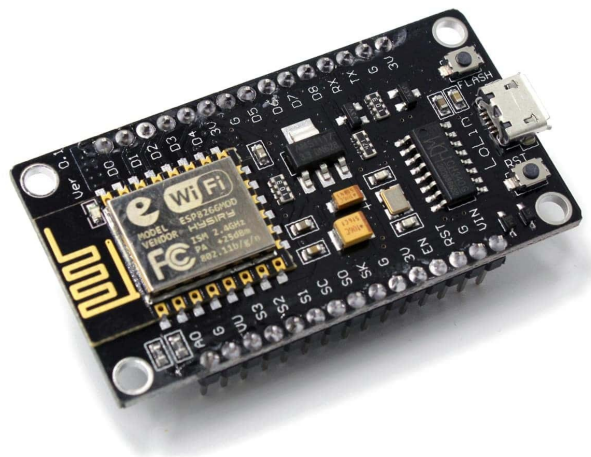


NodeMCU ESP8266 mit Arduino-IDE ins WLAN bringen

NodeMCU V3.4 ESP8266 ESP-12E Dev Kit mit CH340

Der erste Test verbindet das Board mit deinem WLAN und gibt die lokale IP-Adresse im seriellen Monitor aus.



Produkt bei Makershop:

<https://www.makershop.de/plattformen/esp8266/nodemcu-esp8266-dev-kit/>

SKU 7779 | Stand 26.07.2026

In wenigen Minuten die erste WLAN-Meldung sehen

Der erste Test verbindet das Board mit deinem WLAN und gibt die lokale IP-Adresse im seriellen Monitor aus.

Boardprofil

NodeMCU 1.0 (ESP-12E Module) im Arduino-ESP8266-Core

Treiber	CH340-Treiber installieren, falls kein serieller Port erscheint
Upload	Micro-USB verbinden, Port wählen und Sketch übertragen
Serieller Monitor	115200 Baud für Status und IP-Adresse
WLAN-Daten	SSID und Passwort im Beispielsketch eintragen
GPIO-Test	LED oder Ausgang erst nach erfolgreichem Upload anschließen

Schritt für Schritt

1. Schritt 1

Arduino-IDE öffnen und den ESP8266-Boardverwalter installieren.

2. Schritt 2

NodeMCU per Micro-USB verbinden und den passenden seriellen Port auswählen.

3. Schritt 3

Im Sketch deine WLAN-SSID und dein Passwort eintragen.

4. Schritt 4

Sketch hochladen und den seriellen Monitor mit 115200 Baud öffnen.

5. Schritt 5

Die angezeigte IP-Adresse im Netzwerk weiterverwenden, zum Beispiel für Webserver- oder MQTT-Projekte.

Beispiel

```
#include <ESP8266WiFi.h>

const char* ssid = "DEIN_WLAN";
const char* password = "DEIN_PASSWORT";

void setup() {
  Serial.begin(115200);
  pinMode(LED_BUILTIN, OUTPUT);
  WiFi.mode(WIFI_STA);
  WiFi.begin(ssid, password);

  while (WiFi.status() != WL_CONNECTED) {
    digitalWrite(LED_BUILTIN, !digitalRead(LED_BUILTIN));
    delay(250);
    Serial.print(".");
  }

  digitalWrite(LED_BUILTIN, HIGH);
  Serial.println();
  Serial.print("NodeMCU online: " );
  Serial.println(WiFi.localIP());
}

void loop() {
  delay(1000);
}
```

Vertiefung

Boardpaket installieren

In der Arduino-IDE den ESP8266-Boardverwalter einrichten und anschließend ein NodeMCU-kompatibles Boardprofil auswählen.

USB-Verbindung prüfen

Das Board per Micro-USB anschließen, den CH340-Port auswählen und den seriellen Monitor auf 115200 Baud setzen.

WLAN-Test hochladen

SSID und Passwort eintragen, Sketch übertragen und prüfen, ob eine IP-Adresse ausgegeben wird.

GPIO-Projekt erweitern

Nach dem erfolgreichen Netzwerkstart Sensoren, LEDs oder Treiberstufen anschließen und die Ausgabe per Webserver oder MQTT ausbauen.

Stabile Versorgung planen

Bei Relais, Motoren oder mehreren Sensoren eine passende externe Versorgung und gemeinsame Masse einplanen.

Häufige Fragen

Welches Board wähle ich in der Arduino-IDE?

Für diese Bauform ist in der Praxis meist „NodeMCU 1.0 (ESP-12E Module)“ der passende Startpunkt im Arduino-ESP8266-Core.

Warum erscheint kein serieller Port?

Dann fehlt häufig der CH340-Treiber oder das USB-Kabel ist nur ein Ladekabel. Ein Datenkabel und der richtige Treiber lösen das meist sofort.

Sind die GPIOs 5-V-tolerant?

Nein. Die ESP8266-Pins sind für 3,3-V-Logik gedacht. 5-V-Signale sollten über Pegelwandler oder geeignete Schaltungen angepasst werden.

Kann ich damit einen kleinen Webserver bauen?

Ja. Der ESP8266 kann sich ins WLAN einbuchen und einfache Webseiten, Messwerte oder Schaltflächen im lokalen Netzwerk bereitstellen.

Wofür sind D0 bis D8 gut?

Das sind NodeMCU-Pinnamen für ESP8266-GPIOs. Einige Pins beeinflussen den Bootvorgang, deshalb sollte man sie bewusst auswählen.

Verlässliche Quellen

Espressif ESP8266EX Datasheet

https://www.espressif.com/sites/default/files/documentation/0a-esp8266ex_datasheet_en.pdf

Arduino Core für ESP8266

<https://github.com/esp8266/Arduino>

NodeMCU DevKit v1.0 Hardware

<https://github.com/nodemcu/nodemcu-devkit-v1.0>

Mehr Wissen für dein Projekt

Aktuelle Produktinformationen, weitere Dokumente und passende Komponenten findest du direkt bei Makershop.

<https://www.makershop.de/plattformen/esp8266/nodemcu-esp8266-dev-kit/>