

ESP8266 Deep Sleep – So legst du deinen ESP schlafen



Der ESP8266 ist ein vielseitiger und preiswerter Mikrocontroller mit integrierter WiFi-Funktionalität. **Eine seiner bemerkenswertesten Funktionen ist jedoch der Deep-Sleep-Modus, der es ermöglicht, den Stromverbrauch drastisch zu senken, wenn der Mikrocontroller nicht aktiv ist.** Dies ist besonders nützlich in batteriebetriebenen IoT-Projekten, in denen die Energieeffizienz entscheidend ist.

So kann dein ESP zum Beispiel alle 20 Minuten aufwachen, einen Sensorwert übermitteln und sich dann wieder schlafen legen. In diesem Tutorial erfährst du, wie du dabei vorgehst und wie den Deep-Sleep-Modus programmierst.

Benötigte Materialien

Für einen ersten Test ohne Sensoren benötigst du nicht viel:

- Ein ESP8266 Modul (z.B. NodeMCU oder Wemos D1 Mini)
- Ein USB-Kabel zum Programmieren
- Arduino IDE

Übrigens – falls du es noch nicht weißt – bei uns erfährst du, wie du deinen [ESP8266 mit der Arduino IDE programmierst](#).

Verkabelung für den ESP8266 Deep Sleep

Die gute Nachricht, du benötigst nur ein einziges Kabel: Um den Deep-Sleep-Modus zu nutzen, muss der RESET-Pin des ESP8266 mit dem D0-Pin verbunden werden. Entweder verwendest du hier ein Female-Female-Kabel, das du direkt an die Pins des ESP steckst – oder du verwendest ein Breadboard. Mehr musst du für den Schlafmodus nicht tun.

Der Sketch

Lege in der Arduino IDE einen neuen Sketch an und füge den folgenden Code ein und lade ihn auf deinen ESP8266.

```
void setup() {
  // Beginne die serielle Kommunikation mit einer Baudrate von
  9600
  Serial.begin(9600);

  // Ausgabe einer Nachricht, wenn der ESP8266 aufwacht
  Serial.println("Der ESP8266 ist aufgewacht!");
}

void loop() {
  // Warte ein wenig, bevor der ESP8266 in den Deep-Sleep-
  Modus wechselt
  delay(2000);

  // Ausgabe einer Nachricht, bevor der ESP8266 in den Deep-
  Sleep-Modus wechselt
  Serial.println("Der ESP8266 geht in den Deep-Sleep-
  Modus...");

  // Gib dem ESP8266 Zeit, die Nachricht zu senden, bevor er
  in den Deep-Sleep-Modus geht
  delay(100);

  // ESP8266 in den Deep-Sleep-Modus versetzen
```

```
// Die Zeit ist in Mikrosekunden angegeben
// 1 Sekunde = 1.000.000 Mikrosekunden
// In diesem Fall wird der ESP8266 alle 10 Sekunden
aufwachen
ESP.deepSleep(10e6);
}
```

Nun sollte dein ESP8266 alle 10 Sekunden für 2 Sekunden aufwachen.