

Elektro_{xy}

D1 Mini

WLAN-Entwicklungsboard mit ESP8266EX, Version 3

Kurzanleitung

ElektroXY · Bach IT Service UG (haftungsbeschränkt)

Lindenstr. 7 · 53506 Kesseling · www.elektroxy.de

Worum es geht

Der D1 Mini ist ein WLAN-fähiger Mikrocontroller im Streichholzschachtel-Format. Für die meisten Aufgaben rund um Hausautomatisierung und Messwerterfassung ist er die praktischere Wahl als ein größeres Board: Er kostet wenig, braucht kaum Platz und hat über die aufsteckbaren Shields ein Zubehörsystem, das ohne Löten auskommt.

Verglichen mit dem größeren ESP32 hat er einen Kern statt zwei, kein Bluetooth und weniger Anschlüsse. Dafür reicht er für den typischen Einsatz — Sensor auslesen, Wert ins Netz melden, schlafen legen — vollkommen aus und verbraucht dabei weniger.

Ein Eigenheit sollten Sie kennen, bevor Sie loslegen: Die Beschriftung der Anschlüsse auf der Platine stimmt nicht mit den Nummern überein, die Sie im Programm verwenden. Der nächste Abschnitt klärt das.

Technische Daten

Eigenschaft	Wert
Prozessor	ESP8266EX, ein Kern, 80 MHz (auf 160 MHz umschaltbar)
Programmspeicher	4 MB Flash
WLAN	802.11 b/g/n, 2,4 GHz
Antenne	Keramikantenne an Bord, zusätzlich Anschluss für externe Antenne
USB-Anschluss	Micro-USB über CH340-Wandler
Digitale Anschlüsse	11, beschriftet mit D0 bis D8
Analogeingang	einer, beschriftet mit A0
Betriebsspannung der Anschlüsse	3,3 V
Versorgung	5 V über USB oder den 5V-Anschluss
Abmessungen	ca. 35 × 26 mm

Die Beschriftung der Anschlüsse

Auf der Platine stehen D0 bis D8. Im Programm sprechen Sie die Anschlüsse aber über ihre GPIO-Nummer an — und die ist eine andere. Diese Tabelle ist der wichtigste Teil dieser Anleitung; ohne sie verdrahten Sie richtig und wundern sich, warum nichts passiert:

Aufdruck	GPIO	Besonderheit
D0	GPIO 16	kein Interrupt, kein PWM; weckt aus dem Tiefschlaf
D1	GPIO 5	Standardbelegung für I ² C SCL
D2	GPIO 4	Standardbelegung für I ² C SDA
D3	GPIO 0	muss beim Einschalten HIGH sein
D4	GPIO 2	muss beim Einschalten HIGH sein; Onboard-LED
D5	GPIO 14	SPI SCK
D6	GPIO 12	SPI MISO
D7	GPIO 13	SPI MOSI
D8	GPIO 15	muss beim Einschalten LOW sein; SPI CS

Aufdruck	GPIO	Besonderheit
A0	—	Analogeingang, 0 bis 3,2 V
TX / RX	GPIO 1 / 3	belegt durch die USB-Verbindung

In der Arduino-IDE sind die Namen D0 bis D8 als Konstanten hinterlegt. Sie dürfen also `digitalWrite(D4, HIGH)` schreiben — gemeint ist dann GPIO 2. Beim Lesen fremder Beispiele lohnt der Blick, welche der beiden Zählweisen dort verwendet wird.

Anschlüsse mit Startbedingung

D3, D4 und D8 entscheiden beim Einschalten darüber, ob der Chip normal startet. Hängt dort etwas, das die Leitung in die falsche Richtung zieht, bleibt das Board stumm. Die Regel:

- An D3 und D4 nichts anschließen, das beim Einschalten nach Masse zieht — also keine LED gegen GND, keinen gedrückten Taster.
- An D8 nichts anschließen, das beim Einschalten nach 3,3 V zieht.
- Nach dem Start sind alle drei ganz normal verwendbar.

Der Analogeingang

Der ESP8266 selbst misst nur bis 1,0 V. Auf dem D1 Mini sitzt ein Spannungsteiler davor, deshalb verträgt A0 hier bis 3,2 V. Mehr nicht — 5 V zerstören den Eingang.

`analogRead(A0)` liefert Werte von 0 bis 1023. Für eine Spannung in Volt rechnen Sie den Messwert mal 3,2 geteilt durch 1023.

Arduino-IDE einrichten

1. Treiber

Die Version 3 des Boards nutzt den USB-Wandler CH340. Erscheint beim Anstecken kein neuer COM-Port, installieren Sie den CH340-Treiber. Unter Linux und macOS ist in der Regel keiner nötig.

2. Boardpaket hinzufügen

In den Einstellungen unter „Zusätzliche Boardverwalter-URLs“ eintragen:

http://arduino.esp8266.com/stable/package_esp8266com_index.json

Danach im Boardverwalter nach „esp8266“ suchen und das Paket installieren.

3. Board auswählen

Einstellung	Wert
Board	LOLIN(WEMOS) D1 R2 & mini
Flash Size	4MB (FS:2MB OTA:~1019KB)
Upload Speed	921600 (bei Problemen 115200)
Port	der COM-Port, der beim Anstecken neu erscheint

Erstes Programm

Die Onboard-LED hängt an D4 und arbeitet verkehrt herum: LOW schaltet sie ein, HIGH aus. Das verwirrt beim ersten Mal, ist aber so gewollt.

```
void setup() {
  Serial.begin(115200);
  pinMode(LED_BUILTIN, OUTPUT); // entspricht D4 bzw. GPIO 2
  Serial.println();
  Serial.println("D1 Mini gestartet.");
  Serial.print("Chip-ID: ");
  Serial.println(ESP.getChipId());
  Serial.print("Flash-Groesse: ");
  Serial.print(ESP.getFlashChipRealSize() / 1024 / 1024);
  Serial.println(" MB");
}

void loop() {
  digitalWrite(LED_BUILTIN, LOW); // LOW schaltet die LED EIN
  delay(200);
  digitalWrite(LED_BUILTIN, HIGH); // HIGH schaltet sie AUS
  delay(1800);
}
```

Serieller Monitor auf 115200 Baud. Beim Start gibt der ESP8266 zusätzlich eine Zeile mit wirren Zeichen aus — das ist die Startmeldung des Bootloaders mit einer anderen Baudrate und kein Fehler.

Ins WLAN bringen

```
#include <ESP8266WiFi.h>

const char* WLAN_NAME      = "MeinNetzwerk";
const char* WLAN_PASSWORT = "geheim";

void setup() {
  Serial.begin(115200);
  delay(500);

  WiFi.mode(WIFI_STA);
  WiFi.begin(WLAN_NAME, WLAN_PASSWORT);

  Serial.print("Verbinde");
  unsigned long start = millis();
  while (WiFi.status() != WL_CONNECTED) {
    if (millis() - start > 20000) {
      Serial.println(" - keine Verbindung. Neustart.");
      ESP.restart();
    }
    Serial.print(".");
    delay(500);
  }

  Serial.println();
}
```

```
Serial.print("IP-Adresse: ");
Serial.println(WiFi.localIP());
Serial.print("Signalstaerke: ");
Serial.print(WiFi.RSSI());
Serial.println(" dBm");
}
```

```
void loop() {
}
```

Der ESP8266 funkt nur auf 2,4 GHz. Ein reines 5-GHz-Netz findet er nicht.

Tiefschlaf für Batteriebetrieb

Hier hat der D1 Mini eine Besonderheit, die viele übersehen: Damit er aus dem Tiefschlaf wieder aufwacht, muss D0 mit RST verbunden werden — mit einem Draht oder einem Widerstand von etwa 470 Ohm. Ohne diese Brücke schläft das Board ein und wacht nie wieder auf.

Die Brücke stört allerdings das Übertragen neuer Programme. Nehmen Sie sie zum Programmieren ab und setzen Sie sie danach wieder.

```
#define SCHLAFDAUER_SEKUNDEN 300

void setup() {
  Serial.begin(115200);
  Serial.println("Wach. Messung laeuft ...");

  // hier messen und den Wert verschicken

  Serial.println("Gehe schlafen.");
  Serial.flush();
  ESP.deepSleep(SCHLAFDAUER_SEKUNDEN * 1000000ULL);
}

void loop() {
  // wird nie erreicht
}
```

Die längste Schlafdauer liegt bauartbedingt bei etwa drei Stunden. Wer längere Pausen braucht, weckt das Board mehrfach und zählt die Durchläufe mit.

Shields

Für den D1 Mini gibt es aufsteckbare Erweiterungen in derselben Grundfläche — Displays, Speicherkarten-Leser, Relais, Akkuladeregler, Prototypenplatinen. Sie werden schlicht aufgesteckt, mehrere lassen sich stapeln.

Beim Stapeln lohnt ein Blick auf die belegten Anschlüsse: Zwei Shields, die beide dieselbe Leitung verwenden, vertragen sich nicht. Bei I²C-Geräten ist das unkritisch, solange die Adressen verschieden sind — beide hängen ohnehin gemeinsam an D1 und D2.

Fehlersuche

Beobachtung	Wahrscheinliche Ursache	Abhilfe
Kein COM-Port sichtbar	CH340-Treiber fehlt oder reines Ladekabel	Treiber installieren, anderes USB-Kabel probieren
Board startet nicht, bleibt stumm	D3, D4 oder D8 beim Einschalten falsch beschaltet	Diese Anschlüsse beim Start freilassen
Angeschlossenes Bauteil reagiert nicht	Aufdruck mit GPIO-Nummer verwechselt	Zuordnungstabelle prüfen, D4 ist GPIO 2
Board wacht aus dem Tiefschlaf nicht auf	Brücke von D0 nach RST fehlt	Draht oder 470 Ohm zwischen D0 und RST setzen
Übertragung schlägt fehl, obwohl Port da ist	Brücke D0 – RST noch gesteckt	Zum Programmieren abnehmen
Analogwert bleibt am Anschlag	Mehr als 3,2 V an A0	Spannungsteiler vorschalten
Serieller Monitor zeigt wirre Zeichen	Falsche Baudrate	Auf 115200 stellen
WLAN wird nicht gefunden	Netz funkt nur auf 5 GHz	2,4-GHz-Band am Router aktivieren
Board wird warm und startet neu	Zu viel Strom über 3V3 entnommen	Sensoren mit eigener Quelle versorgen

Angaben ohne Gewähr. Alle Werte beziehen sich auf den Baustein ESP8266EX und die übliche Beschaltung der D1-Mini-Boards; die Bestückung kann je nach Ausführung abweichen.