

M E S P I W O R D F  
B A C L O C K D A N  
E A N A L L E E F T  
M A N U A L N J L K

# Anleitung

Für die Multilayout-ESP-Wordclock

---

# Inhaltsverzeichnis

<b>1</b>	<b>Einleitung</b>	<b>1</b>
1.1	Hauptfunktionen . . . . .	1
1.2	Uhren Varianten . . . . .	2
1.3	Voraussetzungen . . . . .	2
<b>2</b>	<b>Installation der Wortuhr Software</b>	<b>3</b>
2.1	Erstmaliges Aufspielen mittels Binary . . . . .	3
2.2	Windows . . . . .	3
2.3	MacOS . . . . .	4
2.4	Linux . . . . .	4
2.5	Update der Firmware über das Web-Frontend . . . . .	4
<b>3</b>	<b>Hardware</b>	<b>5</b>
3.1	Pegelanpassung . . . . .	6
3.2	Anschlussbeispiel . . . . .	6
<b>4</b>	<b>Funktionen der Uhrensoftware auf dem ESP</b>	<b>8</b>
4.1	Verbinden der Uhr mit dem eigenen WLAN . . . . .	8
4.2	Zeitserver (NTP Server) . . . . .	8
4.3	Automatische Helligkeitsregelung . . . . .	8

# 1 Einleitung

Diese Anleitung soll helfen, eine mehrsprachige Wortuhr zu bauen, die auf einem ESP8266 Mikrocontroller und einer programmierbaren LED-Leiste (WS2812 oder SK6812) basiert. Dieses Projekt wurde über mehrere Jahre von einer Vielzahl von Mitwirkenden auf Github entwickelt und frei zur Verfügung gestellt.

Eine Wortuhr (auch als Wordclock bekannt) ist ein schönes DIY-Projekt für Anfänger, das Technologie und Design kombiniert, um eine funktionelle und ästhetisch ansprechende Uhr zu schaffen. Egal, ob Sie Anfänger oder erfahrener Bastler sind, dieses Projekt ist eine großartige Gelegenheit, Ihre Fähigkeiten unter Beweis zu stellen und etwas wirklich Besonderes zu schaffen. Die Gestaltungsmöglichkeiten liegen vor allem im Rahmen und in der Wahl der Oberfläche der Uhr.

## 1.1 Hauptfunktionen

Die bereitgestellte Software zum Betreiben der Wortuhr hat mehrere Funktionen:

- Mehrsprachig (Deutsch, Englisch, Niederländisch, Italienisch, Spanisch, Ungarisch, Französisch, Römänisch, Schweizer Deutsch, Russisch und Schwedisch)
- Unterstützung für mehrere Matrizengrößen und LED-Abstände
- Farbwechsel der Anzeigefarbe möglich (RGB oder RGBW)
- Digitale Uhranzeige
- Regenbogenfarbwechsel
- Umgebungslicht (als Sekundenzeiger ausgeführt oder Ambientlight)
- Automatische Helligkeitsregelung (optional über Sensor)
- Offlinebetrieb (optional über entsprechende RTC)

- Auswahl an dialektspezifischen Anzeigen
- Home-Assistant-Einbindung mit Autodiscovery

## 1.2 Uhren Varianten

## 1.3 Voraussetzungen

Die folgende Hardware/Software wird für dieses Projekt benötigt:

### Hardware

- NodeMCU oder vergleichbares Board mit einem dem ESP8266 oder ESP8285 Chip
- WS2812B RGB-LED-Streifen oder SK6812 RGBW-Streifen
- Stromversorgung 5V 2A
- Optional: LDR, 10 KOhm-Widerstand
- Optional: BH1750 (digitaler Helligkeitssensor, anstatt LDR)
- Optional: DS3231 (Echtzeituhr, für den offline Betrieb)

### Software

- PlatformIO Core oder IDE (bspw. Visual Studio Code)
- Node.js
- Git

## 2 Installation der Wortuhr Software

### 2.1 Erstmaliges Aufspielen mittels Binary

Für das erstmalige Aufspielen liegen auf der Githubseite Rechts «Releases» zur Verfügung. Dabei handelt es sich um fertig compilierte Versionen der Software. Programme zum aufspielen der binary findet man z.B. unter «<https://github.com/nodemcu/nodemcu-flasher>»

### 2.2 Windows

1. Installieren Sie die aktuellsten Versionen von PlatformIO, Node.js und Git manuell
2. Dadurch wird Visual Studio Code installiert, mit einem PlatformIO-Symbol (Ameisenkopf/Alien) in der Seitenleiste
3. Gehen Sie zu «Quick Access / Miscellaneous» und geben Sie den Befehl «Clone Git Project» ein, und geben Sie <https://github.com/ESPWortuhr/Multilayout-ESP-Wordclock> als URL ein
4. Gehen Sie dann zu «Projekte», fügen Sie das neue Projekt mit «Vorhandenes hinzufügen» zur Liste hinzu und klicken Sie auf «Öffnen».
5. In der PlatformIO-Seitenleiste erscheint nun «Project Tasks». Wählen Sie den Befehl «General / Upload» (dauert ein paar Minuten, die Software wird zuerst erstellt).
6. Schließen Sie den ESP8266 über USB an. Wenn die Wortuhr-Software erstellt ist, wird sie auf dem ESP installiert.

## 2.3 MacOS

Der einfachste Weg wäre mit «homebrew»:

```
1    brew install platformio
2    brew install node
3    git clone https://github.com/ESPWortuhr/Wortuhr
4    cd Wortuhr
5    pio run -t upload
```

Dabei ist der ESP8266 über USB angeschlossen. Die Wortuhr-Software wird dabei mittels Plattformio erstellt und auf den ESP übertragen.

## 2.4 Linux

```
1    python3 -c "$(curl -fsSL https://raw.githubusercontent.com/...
      platformio/platformio/master/scripts/get-platformio.py) "
2    sudo apt install npm
3    git clone https://github.com/ESPWortuhr/Wortuhr
4    cd Wortuhr
5    pio run -t upload
```

Dabei ist der ESP8266 über USB angeschlossen. Die Wortuhr-Software wird dabei mittels Plattformio erstellt und auf den ESP übertragen.

## 2.5 Update der Firmware über das Web-Frontend

Mit der Software ist es möglich, die Firmware **OTA** (over the air), über den in der Uhr eingebauten Webserver zu aktualisieren. Dies ist in den meisten Fällen einfacher als das direkte Flashen. Ausschlaggebend für diese Vorgehensweise ist, dass die Uhr über WLAN erreichbar ist. Die Update-Schnittstelle ist dann über die Adresse `http://<IP-Adresse der Uhr>:81/update` erreichbar (Beispiel: `http://192.168.4.1:81/update`). Die Adresse der Uhr wird in der Grundeinstellung beim Start als Laufschrift ausgegeben.

## 3 Hardware

Für den Aufbau der Wortuhr wird folgende elektronische Hardware benötigt:

- NodeMCU oder vergleichbares Board mit ESP8266 oder ESP8285 Chip
- RGB-LED-Streifen WS2812B oder RGBW-Streifen SK6812
- Stromversorgung 5V 2A
- Optional: LDR, 10 KOhm Widerstand
- Optional: BH1750 (digitaler Helligkeitssensor, ersetzt LDR)
- Optional: DS3231 (Echtzeituhr, für Offline-Betrieb)

Ich möchte dich gerne darüber informieren, was du noch alles für den Bau benötigst. Da es verschiedene Arten gibt, eine Uhr zu bauen, werde ich dir hier nur die grundlegenden Dinge aufzählen, die ich oft verwende:

- Bilderrahmen (Holz oder Metall mit mindestens 14mm Rahmenhöhe)
- Diffusorfolie
- Buchstabenmaske
- Abstandsgitter (Holz oder Kunststoffschaum)

## 3.1 Pegelanpassung

Der ESP arbeitet intern mit einer Betriebsspannung von 3,3V. Die WS2812(B) Streifen arbeiten dagegen mit 5V. Dies erfordert eine Pegelanpassung, damit die digitalen Signale fehlerfrei zu den Stripes übertragen werden können. Ohne Anpassung kann es auch funktionieren, muss es aber nicht. Es gilt: "Der minimale High-Pegel des Ausgangs muss höher sein als der minimale High-Pegel des Eingangs" Was heißt das im Klartext? Laut Datenblatt des Stripes sind folgende Pegel notwendig, um ein sauberes HIGH und LOW Signal zu bekommen. Um ein HIGH Signal am Eingang des WS2812B Stripes zu erzeugen, muss laut Datenblatt (Bild 1) mindestens ein Pegel von 4,3V erzeugt werden. Da der ESP aber maximal 3,3V erzeugen kann, wird hier kein sicheres HIGH Signal erzeugt.

Nun kommt der Trick mit der Diode: Die erste LED wird nun mit einer Diode in der Plus Leitung betrieben. Die Durchlassspannung der Diode beträgt 0,7V. Das heißt, dass die erste Diode mit 4,3V betrieben wird. Das ist laut Datenblatt auch zulässig. Bei 4,3V beträgt der benötigte HIGH Pegel nun noch 3,6V. Da die Hersteller noch Toleranzen mit einberechnen funktioniert das ganze Konstrukt wieder. Nun gibt die erste LED den Pegel mit 4,3V an die nächste LED weiter (diese wird wieder mit 5V betrieben und der HIGH Pegel liegt mit 4,3V wieder im Datenblatt in der Toleranz). Die zweite LED verstärkt sozusagen das Signal wieder auf die volle 5V. Das heißt, die erste LED wird sozusagen als Pegelanpassung missbraucht.

## 3.2 Anschlussbeispiel

Der Datenpin des LED Stripes muss zwingend an dem Port «Rx» angeschlossen werden. Grund für den festen Pin ist die verwendete Library. Der Pin kann nicht verändert werden (auch nicht in der Config). Wie auf dem muss eine Diode in die +5V Leitung der 1. LED. Die +5V Leitung zwischen der 1. und 2. LED muss durchtrennt werden (aber nur diese, nicht die Daten und GND!). Am 2. Streifen wird dann zusätzlich vom Netzteil die +5V eingespeist. Die LED's vom 1. Stripe werden dann sozusagen "rückwärts" mit den nötigen +5V versorgt.



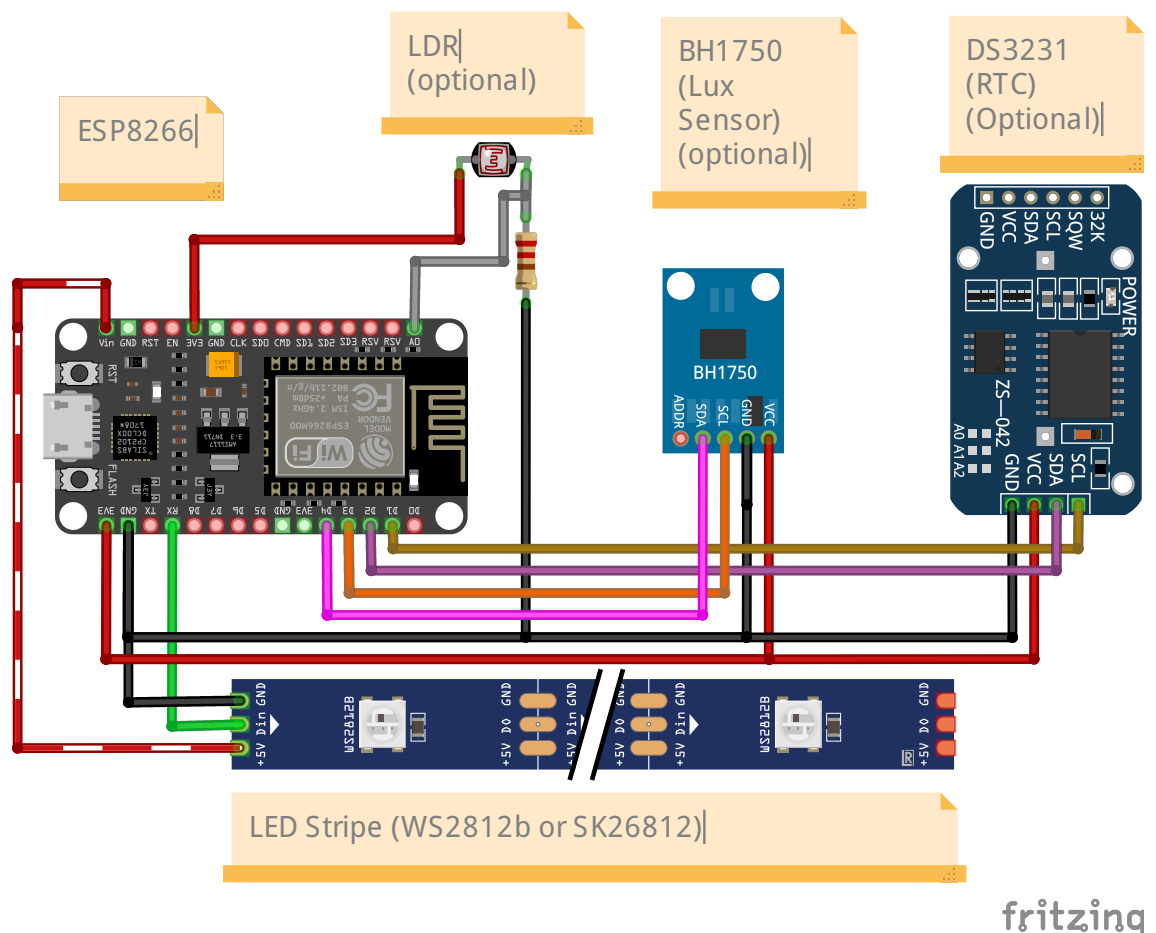


Abbildung 3.1: Anschlussbeispiel

## **4 Funktionen der Uhrensoftware auf dem ESP**

### **4.1 Verbinden der Uhr mit dem eigenen WLAN**

Nach dem erfolgreichen Verbinden kann die Uhr nach Eingabe der angezeigten IP-Adresse in einem Browser konfiguriert werden. Dabei kann die Uhr auch mit dem eigenen WLAN Netz verbunden werden.

### **4.2 Zeitserver (NTP Server)**

Falls die Uhr an das eigene WLAN mit Internet angebunden wird, kann ein Zeitserver hinterlegt werden. Dieser sorgt dafür, dass die Uhrzeit zyklisch sekundengenau gestellt wird.

### **4.3 Automatische Helligkeitsregelung**

Falls der optionale LDR an der Uhr angeschlossen wurde, ist es möglich, die Helligkeit automatisch, abhängig der Umgebungshelligkeit, regeln zu lassen. Dazu muss folgende Einstellung aktiviert werden. Durch das Feld Kalibrierung kann man die Grundhelligkeit an seine eigenen Wünsche anpassen.